

ArmsysApi

0.12.18

Generated by Doxygen 1.8.14



# Contents

- 1 Main Page** **1**
  - 1.1 Introduction . . . . . 1
  - 1.2 Overview . . . . . 1
  - 1.3 Examples . . . . . 1
  
- 2 Hierarchical Index** **3**
  - 2.1 Class Hierarchy . . . . . 3
  
- 3 Class Index** **7**
  - 3.1 Class List . . . . . 7
  
- 4 File Index** **9**
  - 4.1 File List . . . . . 9
  
- 5 Class Documentation** **13**
  - 5.1 BAdc Class Reference . . . . . 13
    - 5.1.1 Constructor & Destructor Documentation . . . . . 13
      - 5.1.1.1 BAdc() . . . . . 13
    - 5.1.2 Member Function Documentation . . . . . 14
      - 5.1.2.1 enableBatteryInput() . . . . . 14
      - 5.1.2.2 enableTempAndRefInput() . . . . . 14
      - 5.1.2.3 getValue() . . . . . 14
      - 5.1.2.4 init() . . . . . 14
      - 5.1.2.5 initChannel() . . . . . 14
    - 5.1.3 Member Data Documentation . . . . . 15

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.1.3.1  | oadc                                   | 15 |
| 5.1.3.2  | odev                                   | 15 |
| 5.2      | BArray< T > Class Template Reference   | 15 |
| 5.2.1    | Constructor & Destructor Documentation | 16 |
| 5.2.1.1  | BArray() [1/3]                         | 16 |
| 5.2.1.2  | BArray() [2/3]                         | 17 |
| 5.2.1.3  | BArray() [3/3]                         | 17 |
| 5.2.1.4  | ~BArray()                              | 17 |
| 5.2.2    | Member Function Documentation          | 17 |
| 5.2.2.1  | allocate()                             | 17 |
| 5.2.2.2  | append()                               | 17 |
| 5.2.2.3  | clear()                                | 18 |
| 5.2.2.4  | data()                                 | 18 |
| 5.2.2.5  | dataSize()                             | 18 |
| 5.2.2.6  | del()                                  | 18 |
| 5.2.2.7  | get() [1/2]                            | 18 |
| 5.2.2.8  | get() [2/2]                            | 19 |
| 5.2.2.9  | insert()                               | 19 |
| 5.2.2.10 | number()                               | 19 |
| 5.2.2.11 | operator T*()                          | 19 |
| 5.2.2.12 | operator=()                            | 19 |
| 5.2.2.13 | operator[]() [1/2]                     | 20 |
| 5.2.2.14 | operator[]() [2/2]                     | 20 |
| 5.2.2.15 | relocate()                             | 20 |
| 5.2.2.16 | resize()                               | 20 |
| 5.2.2.17 | size()                                 | 20 |
| 5.2.3    | Member Data Documentation              | 20 |
| 5.2.3.1  | chunkSize                              | 21 |
| 5.2.3.2  | odata                                  | 21 |
| 5.2.3.3  | odataSize                              | 21 |

---

|         |                                               |    |
|---------|-----------------------------------------------|----|
| 5.2.3.4 | osize                                         | 21 |
| 5.3     | BBuffer Class Reference                       | 21 |
| 5.3.1   | Constructor & Destructor Documentation        | 22 |
| 5.3.1.1 | BBuffer()                                     | 22 |
| 5.3.1.2 | ~BBuffer()                                    | 22 |
| 5.3.2   | Member Function Documentation                 | 22 |
| 5.3.2.1 | data()                                        | 22 |
| 5.3.2.2 | len()                                         | 23 |
| 5.3.2.3 | operator[]()                                  | 23 |
| 5.3.2.4 | setSize()                                     | 23 |
| 5.3.2.5 | size()                                        | 23 |
| 5.3.3   | Member Data Documentation                     | 23 |
| 5.3.3.1 | odata                                         | 23 |
| 5.3.3.2 | odataSize                                     | 23 |
| 5.3.3.3 | osize                                         | 24 |
| 5.4     | BBufferFixed< Size > Class Template Reference | 24 |
| 5.4.1   | Constructor & Destructor Documentation        | 24 |
| 5.4.1.1 | BBufferFixed()                                | 24 |
| 5.4.2   | Member Data Documentation                     | 24 |
| 5.4.2.1 | odataFixed                                    | 25 |
| 5.5     | BBufferStore Class Reference                  | 25 |
| 5.5.1   | Constructor & Destructor Documentation        | 26 |
| 5.5.1.1 | BBufferStore()                                | 26 |
| 5.5.1.2 | ~BBufferStore()                               | 26 |
| 5.5.2   | Member Function Documentation                 | 26 |
| 5.5.2.1 | getHexString()                                | 26 |
| 5.5.2.2 | getPos()                                      | 26 |
| 5.5.2.3 | pop() [1/14]                                  | 26 |
| 5.5.2.4 | pop() [2/14]                                  | 27 |
| 5.5.2.5 | pop() [3/14]                                  | 27 |

---

|          |                           |    |
|----------|---------------------------|----|
| 5.5.2.6  | pop() [4/14]              | 27 |
| 5.5.2.7  | pop() [5/14]              | 27 |
| 5.5.2.8  | pop() [6/14]              | 27 |
| 5.5.2.9  | pop() [7/14]              | 27 |
| 5.5.2.10 | pop() [8/14]              | 27 |
| 5.5.2.11 | pop() [9/14]              | 28 |
| 5.5.2.12 | pop() [10/14]             | 28 |
| 5.5.2.13 | pop() [11/14]             | 28 |
| 5.5.2.14 | pop() [12/14]             | 28 |
| 5.5.2.15 | pop() [13/14]             | 28 |
| 5.5.2.16 | pop() [14/14]             | 28 |
| 5.5.2.17 | push() [1/14]             | 28 |
| 5.5.2.18 | push() [2/14]             | 29 |
| 5.5.2.19 | push() [3/14]             | 29 |
| 5.5.2.20 | push() [4/14]             | 29 |
| 5.5.2.21 | push() [5/14]             | 29 |
| 5.5.2.22 | push() [6/14]             | 29 |
| 5.5.2.23 | push() [7/14]             | 29 |
| 5.5.2.24 | push() [8/14]             | 29 |
| 5.5.2.25 | push() [9/14]             | 30 |
| 5.5.2.26 | push() [10/14]            | 30 |
| 5.5.2.27 | push() [11/14]            | 30 |
| 5.5.2.28 | push() [12/14]            | 30 |
| 5.5.2.29 | push() [13/14]            | 30 |
| 5.5.2.30 | push() [14/14]            | 30 |
| 5.5.2.31 | setHexString()            | 30 |
| 5.5.2.32 | setPos()                  | 31 |
| 5.5.3    | Member Data Documentation | 31 |
| 5.5.3.1  | opos                      | 31 |
| 5.5.3.2  | oswapBytes                | 31 |

---

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.6      | BCan Class Reference                   | 31 |
| 5.6.1    | Constructor & Destructor Documentation | 32 |
| 5.6.1.1  | BCan()                                 | 32 |
| 5.6.2    | Member Function Documentation          | 32 |
| 5.6.2.1  | debugPrint()                           | 32 |
| 5.6.2.2  | init()                                 | 32 |
| 5.6.2.3  | interrupt()                            | 32 |
| 5.6.2.4  | packetMode()                           | 33 |
| 5.6.2.5  | read()                                 | 33 |
| 5.6.2.6  | readAvailable()                        | 33 |
| 5.6.2.7  | recvCanPacket()                        | 33 |
| 5.6.2.8  | sendCanPacket()                        | 33 |
| 5.6.2.9  | write()                                | 34 |
| 5.6.2.10 | writeAvailable()                       | 34 |
| 5.6.3    | Member Data Documentation              | 34 |
| 5.6.3.1  | ocan                                   | 34 |
| 5.6.3.2  | odev                                   | 34 |
| 5.6.3.3  | orxFifoCmd                             | 34 |
| 5.6.3.4  | orxFifoReply                           | 35 |
| 5.7      | BCanPacketFifo Class Reference         | 35 |
| 5.7.1    | Constructor & Destructor Documentation | 35 |
| 5.7.1.1  | BCanPacketFifo()                       | 35 |
| 5.7.2    | Member Data Documentation              | 35 |
| 5.7.2.1  | inPacket                               | 36 |
| 5.7.2.2  | num                                    | 36 |
| 5.7.2.3  | numFifo                                | 36 |
| 5.8      | BClocks Class Reference                | 36 |
| 5.8.1    | Member Data Documentation              | 36 |
| 5.8.1.1  | ahbClock                               | 36 |
| 5.8.1.2  | apb1Clock                              | 37 |

---

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.8.1.3  | apb2Clock                              | 37 |
| 5.8.1.4  | sysClock                               | 37 |
| 5.9      | BComms Class Reference                 | 37 |
| 5.9.1    | Member Enumeration Documentation       | 38 |
| 5.9.1.1  | Flush                                  | 38 |
| 5.9.2    | Constructor & Destructor Documentation | 39 |
| 5.9.2.1  | BComms()                               | 39 |
| 5.9.2.2  | ~BComms()                              | 39 |
| 5.9.3    | Member Function Documentation          | 39 |
| 5.9.3.1  | connect()                              | 39 |
| 5.9.3.2  | disconnect()                           | 39 |
| 5.9.3.3  | eventQueue()                           | 39 |
| 5.9.3.4  | flush()                                | 40 |
| 5.9.3.5  | init()                                 | 40 |
| 5.9.3.6  | isConnected()                          | 40 |
| 5.9.3.7  | packetMode()                           | 40 |
| 5.9.3.8  | read()                                 | 40 |
| 5.9.3.9  | readAvailable()                        | 40 |
| 5.9.3.10 | setPacketMode()                        | 41 |
| 5.9.3.11 | setTimeout()                           | 41 |
| 5.9.3.12 | wait()                                 | 41 |
| 5.9.3.13 | write()                                | 41 |
| 5.9.3.14 | writeAvailable()                       | 41 |
| 5.9.3.15 | writeChunks()                          | 42 |
| 5.9.4    | Member Data Documentation              | 42 |
| 5.9.4.1  | oconnected                             | 42 |
| 5.9.4.2  | oevent                                 | 42 |
| 5.9.4.3  | oeventNum                              | 42 |
| 5.9.4.4  | oeventQueue                            | 42 |
| 5.9.4.5  | opacketMode                            | 42 |

---

|           |                                        |    |
|-----------|----------------------------------------|----|
| 5.9.4.6   | otimeout                               | 43 |
| 5.10      | BCondInt Class Reference               | 43 |
| 5.10.1    | Detailed Description                   | 44 |
| 5.10.2    | Constructor & Destructor Documentation | 44 |
| 5.10.2.1  | BCondInt()                             | 44 |
| 5.10.2.2  | ~BCondInt()                            | 44 |
| 5.10.3    | Member Function Documentation          | 44 |
| 5.10.3.1  | decrement()                            | 44 |
| 5.10.3.2  | increment()                            | 44 |
| 5.10.3.3  | operator++()                           | 44 |
| 5.10.3.4  | operator+=()                           | 45 |
| 5.10.3.5  | operator--()                           | 45 |
| 5.10.3.6  | operator-=()                           | 45 |
| 5.10.3.7  | setValue()                             | 45 |
| 5.10.3.8  | value()                                | 45 |
| 5.10.3.9  | waitLessThan()                         | 45 |
| 5.10.3.10 | waitLessThanOrEqual()                  | 46 |
| 5.10.3.11 | waitMoreThanOrEqual()                  | 46 |
| 5.10.4    | Member Data Documentation              | 46 |
| 5.10.4.1  | omutex                                 | 46 |
| 5.10.4.2  | osema                                  | 46 |
| 5.10.4.3  | ovalue                                 | 46 |
| 5.11      | BCritical Class Reference              | 47 |
| 5.11.1    | Detailed Description                   | 47 |
| 5.11.2    | Member Function Documentation          | 47 |
| 5.11.2.1  | lock()                                 | 47 |
| 5.11.2.2  | unlock()                               | 47 |
| 5.11.3    | Member Data Documentation              | 47 |
| 5.11.3.1  | ostate                                 | 48 |
| 5.12      | BCriticalLock Class Reference          | 48 |

---

|                                                         |    |
|---------------------------------------------------------|----|
| 5.12.1 Detailed Description . . . . .                   | 48 |
| 5.12.2 Constructor & Destructor Documentation . . . . . | 48 |
| 5.12.2.1 BCriticalLock() . . . . .                      | 48 |
| 5.12.2.2 ~BCriticalLock() . . . . .                     | 48 |
| 5.12.3 Member Function Documentation . . . . .          | 49 |
| 5.12.3.1 lock() . . . . .                               | 49 |
| 5.12.4 Member Data Documentation . . . . .              | 49 |
| 5.12.4.1 olock . . . . .                                | 49 |
| 5.13 BDataChunk Class Reference . . . . .               | 49 |
| 5.13.1 Constructor & Destructor Documentation . . . . . | 49 |
| 5.13.1.1 BDataChunk() . . . . .                         | 49 |
| 5.13.2 Member Data Documentation . . . . .              | 50 |
| 5.13.2.1 data . . . . .                                 | 50 |
| 5.13.2.2 size . . . . .                                 | 50 |
| 5.14 BDevices Class Reference . . . . .                 | 50 |
| 5.14.1 Constructor & Destructor Documentation . . . . . | 50 |
| 5.14.1.1 BDevices() . . . . .                           | 50 |
| 5.14.2 Member Function Documentation . . . . .          | 50 |
| 5.14.2.1 enable() . . . . .                             | 51 |
| 5.14.2.2 getClocks() . . . . .                          | 51 |
| 5.14.2.3 init() . . . . .                               | 51 |
| 5.15 BDict< Type > Class Template Reference . . . . .   | 51 |
| 5.15.1 Member Typedef Documentation . . . . .           | 52 |
| 5.15.1.1 iterator . . . . .                             | 52 |
| 5.15.2 Constructor & Destructor Documentation . . . . . | 52 |
| 5.15.2.1 BDict() [1/2] . . . . .                        | 53 |
| 5.15.2.2 BDict() [2/2] . . . . .                        | 53 |
| 5.15.3 Member Function Documentation . . . . .          | 53 |
| 5.15.3.1 append() [1/2] . . . . .                       | 53 |
| 5.15.3.2 append() [2/2] . . . . .                       | 53 |

---

|           |                                                                  |    |
|-----------|------------------------------------------------------------------|----|
| 5.15.3.3  | <a href="#">clear()</a>                                          | 53 |
| 5.15.3.4  | <a href="#">del() [1/2]</a>                                      | 54 |
| 5.15.3.5  | <a href="#">del() [2/2]</a>                                      | 54 |
| 5.15.3.6  | <a href="#">find()</a>                                           | 54 |
| 5.15.3.7  | <a href="#">hashAdd()</a>                                        | 54 |
| 5.15.3.8  | <a href="#">hashDelete()</a>                                     | 54 |
| 5.15.3.9  | <a href="#">hashFind()</a>                                       | 55 |
| 5.15.3.10 | <a href="#">hashPrint()</a>                                      | 55 |
| 5.15.3.11 | <a href="#">hasKey()</a>                                         | 55 |
| 5.15.3.12 | <a href="#">insert()</a>                                         | 55 |
| 5.15.3.13 | <a href="#">key()</a>                                            | 55 |
| 5.15.3.14 | <a href="#">operator+()</a>                                      | 55 |
| 5.15.3.15 | <a href="#">operator=()</a>                                      | 56 |
| 5.15.3.16 | <a href="#">operator[]() [1/3]</a>                               | 56 |
| 5.15.3.17 | <a href="#">operator[]() [2/3]</a>                               | 56 |
| 5.15.3.18 | <a href="#">operator[]() [3/3]</a>                               | 56 |
| 5.15.4    | <a href="#">Member Data Documentation</a>                        | 56 |
| 5.15.4.1  | <a href="#">ohashLists</a>                                       | 56 |
| 5.15.4.2  | <a href="#">ohashSize</a>                                        | 56 |
| 5.16      | <a href="#">BDictItem&lt; Type &gt; Class Template Reference</a> | 57 |
| 5.16.1    | <a href="#">Detailed Description</a>                             | 57 |
| 5.16.2    | <a href="#">Constructor &amp; Destructor Documentation</a>       | 57 |
| 5.16.2.1  | <a href="#">BDictItem()</a>                                      | 57 |
| 5.16.3    | <a href="#">Member Data Documentation</a>                        | 57 |
| 5.16.3.1  | <a href="#">key</a>                                              | 57 |
| 5.16.3.2  | <a href="#">value</a>                                            | 58 |
| 5.17      | <a href="#">BDma Class Reference</a>                             | 58 |
| 5.17.1    | <a href="#">Member Enumeration Documentation</a>                 | 59 |
| 5.17.1.1  | <a href="#">Burst</a>                                            | 59 |
| 5.17.1.2  | <a href="#">Dir</a>                                              | 59 |

---

|           |                                        |    |
|-----------|----------------------------------------|----|
| 5.17.1.3  | DmaType                                | 59 |
| 5.17.1.4  | Fifo                                   | 60 |
| 5.17.1.5  | Size                                   | 60 |
| 5.17.1.6  | Status                                 | 60 |
| 5.17.2    | Constructor & Destructor Documentation | 61 |
| 5.17.2.1  | BDma()                                 | 61 |
| 5.17.3    | Member Function Documentation          | 61 |
| 5.17.3.1  | config()                               | 61 |
| 5.17.3.2  | configFlow()                           | 61 |
| 5.17.3.3  | doDma()                                | 61 |
| 5.17.3.4  | init()                                 | 62 |
| 5.17.3.5  | interrupt()                            | 62 |
| 5.17.3.6  | numLeft()                              | 62 |
| 5.17.3.7  | printStatus()                          | 62 |
| 5.17.3.8  | setCompleteCallback()                  | 62 |
| 5.17.3.9  | setNextBuffer()                        | 62 |
| 5.17.3.10 | statusClear()                          | 63 |
| 5.17.3.11 | statusGet()                            | 63 |
| 5.17.3.12 | stop()                                 | 63 |
| 5.17.3.13 | wait()                                 | 63 |
| 5.17.3.14 | waitStatus()                           | 63 |
| 5.17.4    | Member Data Documentation              | 63 |
| 5.17.4.1  | ochannel                               | 63 |
| 5.17.4.2  | ocompleteCallback                      | 64 |
| 5.17.4.3  | ocontroller                            | 64 |
| 5.17.4.4  | odma                                   | 64 |
| 5.17.4.5  | odmaStream                             | 64 |
| 5.17.4.6  | odmaType                               | 64 |
| 5.17.4.7  | oerr                                   | 64 |
| 5.17.4.8  | ostatus                                | 64 |

---

|           |                                        |    |
|-----------|----------------------------------------|----|
| 5.17.4.9  | ostream                                | 65 |
| 5.17.4.10 | otranSize                              | 65 |
| 5.17.4.11 | owait                                  | 65 |
| 5.18      | BError Class Reference                 | 65 |
| 5.18.1    | Constructor & Destructor Documentation | 65 |
| 5.18.1.1  | BError()                               | 66 |
| 5.18.2    | Member Function Documentation          | 66 |
| 5.18.2.1  | num()                                  | 66 |
| 5.18.2.2  | operator int()                         | 66 |
| 5.18.2.3  | set()                                  | 66 |
| 5.18.2.4  | str()                                  | 66 |
| 5.18.3    | Member Data Documentation              | 66 |
| 5.18.3.1  | nullString                             | 66 |
| 5.18.3.2  | oerr                                   | 67 |
| 5.18.3.3  | ostr                                   | 67 |
| 5.19      | BEvent Class Reference                 | 67 |
| 5.19.1    | Constructor & Destructor Documentation | 67 |
| 5.19.1.1  | BEvent()                               | 67 |
| 5.19.2    | Member Function Documentation          | 67 |
| 5.19.2.1  | arg()                                  | 68 |
| 5.19.2.2  | type()                                 | 68 |
| 5.19.3    | Member Data Documentation              | 68 |
| 5.19.3.1  | oarg                                   | 68 |
| 5.19.3.2  | otype                                  | 68 |
| 5.20      | BFifo< Type > Class Template Reference | 68 |
| 5.20.1    | Constructor & Destructor Documentation | 70 |
| 5.20.1.1  | BFifo()                                | 70 |
| 5.20.1.2  | ~BFifo()                               | 70 |
| 5.20.2    | Member Function Documentation          | 70 |
| 5.20.2.1  | clear()                                | 70 |

---

|           |                                        |    |
|-----------|----------------------------------------|----|
| 5.20.2.2  | <code>operator[]()</code>              | 70 |
| 5.20.2.3  | <code>read()</code> [1/2]              | 71 |
| 5.20.2.4  | <code>read()</code> [2/2]              | 71 |
| 5.20.2.5  | <code>readAvailable()</code>           | 71 |
| 5.20.2.6  | <code>readAvailableChunk()</code>      | 71 |
| 5.20.2.7  | <code>readData()</code> [1/2]          | 71 |
| 5.20.2.8  | <code>readData()</code> [2/2]          | 72 |
| 5.20.2.9  | <code>readDone()</code>                | 72 |
| 5.20.2.10 | <code>readPeak()</code>                | 72 |
| 5.20.2.11 | <code>readPos()</code>                 | 72 |
| 5.20.2.12 | <code>rebase()</code>                  | 72 |
| 5.20.2.13 | <code>resize()</code>                  | 73 |
| 5.20.2.14 | <code>size()</code>                    | 73 |
| 5.20.2.15 | <code>write()</code> [1/2]             | 73 |
| 5.20.2.16 | <code>write()</code> [2/2]             | 73 |
| 5.20.2.17 | <code>writeAvailable()</code>          | 73 |
| 5.20.2.18 | <code>writeAvailableChunk()</code>     | 74 |
| 5.20.2.19 | <code>writeBackup()</code>             | 74 |
| 5.20.2.20 | <code>writeData()</code> [1/2]         | 74 |
| 5.20.2.21 | <code>writeData()</code> [2/2]         | 74 |
| 5.20.2.22 | <code>writeDone()</code>               | 74 |
| 5.20.2.23 | <code>writePos()</code>                | 75 |
| 5.20.3    | Member Data Documentation              | 75 |
| 5.20.3.1  | <code>odata</code>                     | 75 |
| 5.20.3.2  | <code>oreadPos</code>                  | 75 |
| 5.20.3.3  | <code>osize</code>                     | 75 |
| 5.20.3.4  | <code>owritePos</code>                 | 75 |
| 5.21      | BFifoChar Class Reference              | 76 |
| 5.21.1    | Constructor & Destructor Documentation | 76 |
| 5.21.1.1  | <code>BFifoChar()</code>               | 76 |

|           |                                        |    |
|-----------|----------------------------------------|----|
| 5.21.2    | Member Function Documentation          | 76 |
| 5.21.2.1  | debugPrint()                           | 76 |
| 5.21.2.2  | find()                                 | 77 |
| 5.21.2.3  | getBUInt32()                           | 77 |
| 5.21.2.4  | putBUInt32()                           | 77 |
| 5.21.2.5  | readInt()                              | 77 |
| 5.22      | BFirmwareInfo Struct Reference         | 77 |
| 5.22.1    | Member Data Documentation              | 78 |
| 5.22.1.1  | checksum                               | 78 |
| 5.22.1.2  | length                                 | 78 |
| 5.22.1.3  | magic                                  | 78 |
| 5.22.1.4  | type                                   | 78 |
| 5.22.1.5  | ver0                                   | 78 |
| 5.22.1.6  | ver1                                   | 78 |
| 5.22.1.7  | ver2                                   | 78 |
| 5.23      | BFlash Class Reference                 | 79 |
| 5.23.1    | Constructor & Destructor Documentation | 79 |
| 5.23.1.1  | BFlash()                               | 79 |
| 5.23.2    | Member Function Documentation          | 80 |
| 5.23.2.1  | disableAccess()                        | 80 |
| 5.23.2.2  | enableAccess()                         | 80 |
| 5.23.2.3  | getWriteProtect()                      | 80 |
| 5.23.2.4  | init()                                 | 80 |
| 5.23.2.5  | page()                                 | 80 |
| 5.23.2.6  | pageAddress()                          | 81 |
| 5.23.2.7  | programEnable()                        | 81 |
| 5.23.2.8  | programErase()                         | 81 |
| 5.23.2.9  | programWord() [1/2]                    | 81 |
| 5.23.2.10 | programWord() [2/2]                    | 81 |
| 5.23.2.11 | waitUntillReady()                      | 82 |

---

|                                               |    |
|-----------------------------------------------|----|
| 5.23.2.12 writeProtect()                      | 82 |
| 5.23.3 Member Data Documentation              | 82 |
| 5.23.3.1 odummy                               | 82 |
| 5.24 BGPIO Class Reference                    | 82 |
| 5.24.1 Member Enumeration Documentation       | 83 |
| 5.24.1.1 Mode                                 | 83 |
| 5.24.1.2 OutType                              | 83 |
| 5.24.1.3 Pull                                 | 84 |
| 5.24.1.4 Source                               | 84 |
| 5.24.1.5 Speed                                | 85 |
| 5.24.2 Constructor & Destructor Documentation | 85 |
| 5.24.2.1 BGPIO()                              | 85 |
| 5.24.3 Member Function Documentation          | 85 |
| 5.24.3.1 getPin()                             | 85 |
| 5.24.3.2 init()                               | 86 |
| 5.24.3.3 initPin()                            | 86 |
| 5.24.3.4 setAlternateSource()                 | 86 |
| 5.24.3.5 setPin()                             | 86 |
| 5.24.3.6 togglePin()                          | 86 |
| 5.25 BHeap Class Reference                    | 87 |
| 5.25.1 Constructor & Destructor Documentation | 87 |
| 5.25.1.1 BHeap()                              | 87 |
| 5.25.2 Member Function Documentation          | 87 |
| 5.25.2.1 allocate()                           | 87 |
| 5.25.2.2 init()                               | 87 |
| 5.25.3 Member Data Documentation              | 87 |
| 5.25.3.1 obase                                | 88 |
| 5.25.3.2 onext                                | 88 |
| 5.25.3.3 otop                                 | 88 |
| 5.26 BHTML Class Reference                    | 88 |

---

---

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.26.1   | Constructor & Destructor Documentation | 88 |
| 5.26.1.1 | BHtml()                                | 89 |
| 5.26.1.2 | ~BHtml()                               | 89 |
| 5.26.2   | Member Function Documentation          | 89 |
| 5.26.2.1 | append() [1/2]                         | 89 |
| 5.26.2.2 | append() [2/2]                         | 89 |
| 5.26.2.3 | appendFormField()                      | 89 |
| 5.26.2.4 | appendFormItem()                       | 90 |
| 5.26.2.5 | appendText()                           | 90 |
| 5.26.2.6 | render() [1/2]                         | 90 |
| 5.26.2.7 | render() [2/2]                         | 90 |
| 5.26.3   | Member Data Documentation              | 90 |
| 5.26.3.1 | nodes                                  | 90 |
| 5.26.3.2 | oargs                                  | 90 |
| 5.26.3.3 | otag                                   | 91 |
| 5.27     | BHttpRequest Class Reference           | 91 |
| 5.27.1   | Member Enumeration Documentation       | 92 |
| 5.27.1.1 | ReadType                               | 92 |
| 5.27.2   | Constructor & Destructor Documentation | 93 |
| 5.27.2.1 | BHttpRequest()                         | 93 |
| 5.27.3   | Member Function Documentation          | 93 |
| 5.27.3.1 | cookieGet()                            | 93 |
| 5.27.3.2 | cookieSet()                            | 93 |
| 5.27.3.3 | devInit()                              | 93 |
| 5.27.3.4 | devProcess()                           | 93 |
| 5.27.3.5 | devRun()                               | 93 |
| 5.27.3.6 | devWrite()                             | 94 |
| 5.27.3.7 | function()                             | 94 |
| 5.27.3.8 | init()                                 | 94 |
| 5.27.3.9 | pageSend()                             | 94 |

---

|           |                           |    |
|-----------|---------------------------|----|
| 5.27.3.10 | pageSendError()           | 94 |
| 5.27.3.11 | pageSendHead()            | 94 |
| 5.27.3.12 | pageSendTail()            | 94 |
| 5.27.3.13 | processHome()             | 95 |
| 5.27.3.14 | processRequest()          | 95 |
| 5.27.3.15 | processUrl()              | 95 |
| 5.27.3.16 | readData()                | 95 |
| 5.27.3.17 | run()                     | 95 |
| 5.27.3.18 | setHeader()               | 95 |
| 5.27.3.19 | writeChunk()              | 96 |
| 5.27.3.20 | writeFlush()              | 96 |
| 5.27.3.21 | writeHead()               | 96 |
| 5.27.3.22 | writeString()             | 96 |
| 5.27.3.23 | writeTableInput() [1/2]   | 96 |
| 5.27.3.24 | writeTableInput() [2/2]   | 96 |
| 5.27.3.25 | writeTableRow()           | 97 |
| 5.27.3.26 | writeTail()               | 97 |
| 5.27.4    | Member Data Documentation | 97 |
| 5.27.4.1  | oauthorisation            | 97 |
| 5.27.4.2  | oboundary                 | 97 |
| 5.27.4.3  | obuffer                   | 97 |
| 5.27.4.4  | obufferNext               | 97 |
| 5.27.4.5  | ocmd                      | 98 |
| 5.27.4.6  | ocontentLen               | 98 |
| 5.27.4.7  | ocontentTypeRecv          | 98 |
| 5.27.4.8  | ocontentTypeSend          | 98 |
| 5.27.4.9  | ocookies                  | 98 |
| 5.27.4.10 | ocookiesSend              | 98 |
| 5.27.4.11 | oget                      | 98 |
| 5.27.4.12 | ohed_cache                | 98 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.27.4.13 | ahead_chunked                          | 99  |
| 5.27.4.14 | ahead_contentType                      | 99  |
| 5.27.4.15 | ahead_extra                            | 99  |
| 5.27.4.16 | ahead_size                             | 99  |
| 5.27.4.17 | ahead_status                           | 99  |
| 5.27.4.18 | orxFifo                                | 99  |
| 5.27.4.19 | osocket                                | 99  |
| 5.27.4.20 | osocketListen                          | 99  |
| 5.27.4.21 | otxFifo                                | 100 |
| 5.27.4.22 | ourl                                   | 100 |
| 5.28      | BI2c Class Reference                   | 100 |
| 5.28.1    | Constructor & Destructor Documentation | 101 |
| 5.28.1.1  | BI2c()                                 | 101 |
| 5.28.2    | Member Function Documentation          | 101 |
| 5.28.2.1  | getStatus()                            | 101 |
| 5.28.2.2  | hasStatus()                            | 101 |
| 5.28.2.3  | init() [1/2]                           | 101 |
| 5.28.2.4  | init() [2/2]                           | 102 |
| 5.28.2.5  | interrupt()                            | 102 |
| 5.28.2.6  | printStatus()                          | 102 |
| 5.28.2.7  | read()                                 | 102 |
| 5.28.2.8  | readAddress16()                        | 102 |
| 5.28.2.9  | readAddress8()                         | 102 |
| 5.28.2.10 | readReg8()                             | 103 |
| 5.28.2.11 | waitFor()                              | 103 |
| 5.28.2.12 | write()                                | 103 |
| 5.28.2.13 | writeAddress16()                       | 103 |
| 5.28.2.14 | writeAddress8()                        | 103 |
| 5.28.2.15 | writeReg8()                            | 104 |
| 5.28.3    | Member Data Documentation              | 104 |

---

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.28.3.1 | odmaRx                                 | 104 |
| 5.28.3.2 | odmaTx                                 | 104 |
| 5.28.3.3 | oi2c                                   | 104 |
| 5.28.3.4 | oint                                   | 104 |
| 5.28.3.5 | olock                                  | 104 |
| 5.28.3.6 | ostatus                                | 105 |
| 5.28.3.7 | otimeout                               | 105 |
| 5.28.3.8 | ouseDma                                | 105 |
| 5.29     | BInterrupt Class Reference             | 105 |
| 5.29.1   | Member Enumeration Documentation       | 105 |
| 5.29.1.1 | anonymous enum                         | 105 |
| 5.29.2   | Constructor & Destructor Documentation | 106 |
| 5.29.2.1 | BInterrupt()                           | 106 |
| 5.29.3   | Member Function Documentation          | 106 |
| 5.29.3.1 | config()                               | 106 |
| 5.29.3.2 | enable()                               | 106 |
| 5.29.3.3 | extConfig()                            | 106 |
| 5.29.3.4 | init()                                 | 107 |
| 5.29.3.5 | print()                                | 107 |
| 5.29.3.6 | setPriority()                          | 107 |
| 5.30     | BIter Class Reference                  | 107 |
| 5.30.1   | Detailed Description                   | 108 |
| 5.30.2   | Constructor & Destructor Documentation | 108 |
| 5.30.2.1 | BIter()                                | 108 |
| 5.30.3   | Member Function Documentation          | 108 |
| 5.30.3.1 | operator BNode *()                     | 108 |
| 5.30.3.2 | operator==(())                         | 108 |
| 5.30.3.3 | valid()                                | 108 |
| 5.30.4   | Member Data Documentation              | 108 |
| 5.30.4.1 | oi                                     | 108 |

---

|                                                         |     |
|---------------------------------------------------------|-----|
| 5.31 BList< T > Class Template Reference . . . . .      | 109 |
| 5.31.1 Detailed Description . . . . .                   | 111 |
| 5.31.2 Member Typedef Documentation . . . . .           | 111 |
| 5.31.2.1 SortFunc . . . . .                             | 111 |
| 5.31.3 Constructor & Destructor Documentation . . . . . | 111 |
| 5.31.3.1 BList() [1/2] . . . . .                        | 111 |
| 5.31.3.2 BList() [2/2] . . . . .                        | 111 |
| 5.31.3.3 ~BList() . . . . .                             | 111 |
| 5.31.4 Member Function Documentation . . . . .          | 112 |
| 5.31.4.1 append() [1/2] . . . . .                       | 112 |
| 5.31.4.2 append() [2/2] . . . . .                       | 112 |
| 5.31.4.3 begin() . . . . .                              | 112 |
| 5.31.4.4 clear() . . . . .                              | 112 |
| 5.31.4.5 del() . . . . .                                | 113 |
| 5.31.4.6 deleteFirst() . . . . .                        | 113 |
| 5.31.4.7 deleteLast() . . . . .                         | 113 |
| 5.31.4.8 end() [1/2] . . . . .                          | 113 |
| 5.31.4.9 end() [2/2] . . . . .                          | 113 |
| 5.31.4.10 front() . . . . .                             | 114 |
| 5.31.4.11 get() [1/2] . . . . .                         | 114 |
| 5.31.4.12 get() [2/2] . . . . .                         | 114 |
| 5.31.4.13 goTo() . . . . .                              | 114 |
| 5.31.4.14 has() . . . . .                               | 114 |
| 5.31.4.15 insert() . . . . .                            | 115 |
| 5.31.4.16 insertAfter() . . . . .                       | 115 |
| 5.31.4.17 isEnd() . . . . .                             | 115 |
| 5.31.4.18 next() . . . . .                              | 115 |
| 5.31.4.19 nodeCreate() [1/2] . . . . .                  | 115 |
| 5.31.4.20 nodeCreate() [2/2] . . . . .                  | 116 |
| 5.31.4.21 nodeGet() [1/2] . . . . .                     | 116 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.31.4.22 | nodeGet() [2/2]                        | 116 |
| 5.31.4.23 | number()                               | 116 |
| 5.31.4.24 | operator+()                            | 116 |
| 5.31.4.25 | operator=()                            | 116 |
| 5.31.4.26 | operator[]() [1/4]                     | 117 |
| 5.31.4.27 | operator[]() [2/4]                     | 117 |
| 5.31.4.28 | operator[]() [3/4]                     | 117 |
| 5.31.4.29 | operator[]() [4/4]                     | 117 |
| 5.31.4.30 | pop()                                  | 117 |
| 5.31.4.31 | position()                             | 117 |
| 5.31.4.32 | prev()                                 | 118 |
| 5.31.4.33 | push()                                 | 118 |
| 5.31.4.34 | queueAdd()                             | 118 |
| 5.31.4.35 | queueGet()                             | 118 |
| 5.31.4.36 | rear()                                 | 118 |
| 5.31.4.37 | size()                                 | 119 |
| 5.31.4.38 | sort() [1/2]                           | 119 |
| 5.31.4.39 | sort() [2/2]                           | 119 |
| 5.31.4.40 | start()                                | 119 |
| 5.31.4.41 | swap()                                 | 119 |
| 5.31.5    | Member Data Documentation              | 119 |
| 5.31.5.1  | olength                                | 120 |
| 5.31.5.2  | onodes                                 | 120 |
| 5.32      | BMutex Class Reference                 | 120 |
| 5.32.1    | Detailed Description                   | 120 |
| 5.32.2    | Constructor & Destructor Documentation | 121 |
| 5.32.2.1  | BMutex() [1/2]                         | 121 |
| 5.32.2.2  | BMutex() [2/2]                         | 121 |
| 5.32.2.3  | ~BMutex()                              | 121 |
| 5.32.3    | Member Function Documentation          | 121 |

---

---

|          |                                              |     |
|----------|----------------------------------------------|-----|
| 5.32.3.1 | lock()                                       | 121 |
| 5.32.3.2 | operator=()                                  | 121 |
| 5.32.3.3 | tryLock()                                    | 121 |
| 5.32.3.4 | unlock()                                     | 122 |
| 5.32.4   | Member Data Documentation                    | 122 |
| 5.32.4.1 | omutex                                       | 122 |
| 5.33     | BMutexLock Class Reference                   | 122 |
| 5.33.1   | Constructor & Destructor Documentation       | 122 |
| 5.33.1.1 | BMutexLock()                                 | 122 |
| 5.33.1.2 | ~BMutexLock()                                | 123 |
| 5.33.2   | Member Function Documentation                | 123 |
| 5.33.2.1 | lock()                                       | 123 |
| 5.33.2.2 | unlock()                                     | 123 |
| 5.33.3   | Member Data Documentation                    | 123 |
| 5.33.3.1 | olock                                        | 123 |
| 5.34     | BNameValue< T > Class Template Reference     | 123 |
| 5.34.1   | Constructor & Destructor Documentation       | 124 |
| 5.34.1.1 | BNameValue() [1/2]                           | 124 |
| 5.34.1.2 | BNameValue() [2/2]                           | 124 |
| 5.34.2   | Member Function Documentation                | 124 |
| 5.34.2.1 | getName()                                    | 124 |
| 5.34.2.2 | getValue()                                   | 124 |
| 5.34.3   | Member Data Documentation                    | 124 |
| 5.34.3.1 | oname                                        | 125 |
| 5.34.3.2 | ovalue                                       | 125 |
| 5.35     | BNameValueList< T > Class Template Reference | 125 |
| 5.35.1   | Member Function Documentation                | 125 |
| 5.35.1.1 | find()                                       | 125 |
| 5.35.1.2 | findPos()                                    | 126 |
| 5.36     | BNetwork Class Reference                     | 126 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.36.1    | Member Enumeration Documentation       | 127 |
| 5.36.1.1  | PhyType                                | 127 |
| 5.36.2    | Constructor & Destructor Documentation | 128 |
| 5.36.2.1  | BNetwork()                             | 128 |
| 5.36.2.2  | ~BNetwork()                            | 128 |
| 5.36.3    | Member Function Documentation          | 128 |
| 5.36.3.1  | config()                               | 128 |
| 5.36.3.2  | etherInit()                            | 128 |
| 5.36.3.3  | etherPacketRecv()                      | 129 |
| 5.36.3.4  | etherPacketSend()                      | 129 |
| 5.36.3.5  | getIpAddress()                         | 129 |
| 5.36.3.6  | getMac()                               | 129 |
| 5.36.3.7  | getStats()                             | 129 |
| 5.36.3.8  | init()                                 | 129 |
| 5.36.3.9  | interrupt()                            | 130 |
| 5.36.3.10 | linkChanged()                          | 130 |
| 5.36.3.11 | phyInit()                              | 130 |
| 5.36.3.12 | phyPrint()                             | 130 |
| 5.36.3.13 | phyProcess()                           | 130 |
| 5.36.3.14 | phySet()                               | 130 |
| 5.36.3.15 | phySetup()                             | 131 |
| 5.36.3.16 | phyUpdate()                            | 131 |
| 5.36.3.17 | process()                              | 131 |
| 5.36.3.18 | process_dhcp()                         | 131 |
| 5.36.3.19 | start()                                | 131 |
| 5.36.4    | Member Data Documentation              | 131 |
| 5.36.4.1  | odhcp                                  | 132 |
| 5.36.4.2  | odhcpState                             | 132 |
| 5.36.4.3  | odhcpTask                              | 132 |
| 5.36.4.4  | oether                                 | 132 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.36.4.5  | oeventQueue                            | 132 |
| 5.36.4.6  | ogateway                               | 132 |
| 5.36.4.7  | oipAddress                             | 133 |
| 5.36.4.8  | omacAddress                            | 133 |
| 5.36.4.9  | onetif                                 | 133 |
| 5.36.4.10 | onetmask                               | 133 |
| 5.36.4.11 | ophyAutoNeg                            | 133 |
| 5.36.4.12 | ophyFullDuplex                         | 133 |
| 5.36.4.13 | ophyNum                                | 134 |
| 5.36.4.14 | ophySpeed                              | 134 |
| 5.36.4.15 | ophyType                               | 134 |
| 5.36.4.16 | oprocessTask                           | 134 |
| 5.36.4.17 | ostats                                 | 134 |
| 5.37      | BNetworkDhcpTask Class Reference       | 135 |
| 5.37.1    | Constructor & Destructor Documentation | 135 |
| 5.37.1.1  | BNetworkDhcpTask()                     | 135 |
| 5.37.2    | Member Function Documentation          | 135 |
| 5.37.2.1  | run()                                  | 135 |
| 5.37.3    | Member Data Documentation              | 136 |
| 5.37.3.1  | onetwork                               | 136 |
| 5.38      | BNetworkProcessTask Class Reference    | 136 |
| 5.38.1    | Constructor & Destructor Documentation | 136 |
| 5.38.1.1  | BNetworkProcessTask()                  | 136 |
| 5.38.2    | Member Function Documentation          | 137 |
| 5.38.2.1  | run()                                  | 137 |
| 5.38.3    | Member Data Documentation              | 137 |
| 5.38.3.1  | onetwork                               | 137 |
| 5.39      | BNetworkStats Class Reference          | 137 |
| 5.39.1    | Constructor & Destructor Documentation | 138 |
| 5.39.1.1  | BNetworkStats()                        | 138 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.39.2   | Member Data Documentation              | 138 |
| 5.39.2.1 | rxDroppedPackets                       | 138 |
| 5.39.2.2 | rxPackets                              | 138 |
| 5.39.2.3 | txDroppedPackets                       | 138 |
| 5.39.2.4 | txPackets                              | 138 |
| 5.40     | BNode Class Reference                  | 139 |
| 5.40.1   | Constructor & Destructor Documentation | 139 |
| 5.40.1.1 | BNode()                                | 139 |
| 5.40.2   | Member Data Documentation              | 139 |
| 5.40.2.1 | next                                   | 139 |
| 5.40.2.2 | prev                                   | 139 |
| 5.41     | BNtpClient Class Reference             | 140 |
| 5.41.1   | Constructor & Destructor Documentation | 140 |
| 5.41.1.1 | BNtpClient()                           | 140 |
| 5.41.2   | Member Function Documentation          | 140 |
| 5.41.2.1 | init()                                 | 140 |
| 5.41.2.2 | run()                                  | 140 |
| 5.41.2.3 | setServerAddress()                     | 140 |
| 5.41.2.4 | setSystemTime()                        | 141 |
| 5.42     | BoapMc1Comms Class Reference           | 141 |
| 5.42.1   | Constructor & Destructor Documentation | 142 |
| 5.42.1.1 | BoapMc1Comms()                         | 142 |
| 5.42.1.2 | ~BoapMc1Comms()                        | 142 |
| 5.42.2   | Member Function Documentation          | 143 |
| 5.42.2.1 | getApiVersion()                        | 143 |
| 5.42.2.2 | packetRx()                             | 143 |
| 5.42.2.3 | packetRxData()                         | 143 |
| 5.42.2.4 | packetRxEnd()                          | 143 |
| 5.42.2.5 | packetTx()                             | 143 |
| 5.42.2.6 | processRequest()                       | 143 |

---

|           |                               |     |
|-----------|-------------------------------|-----|
| 5.42.2.7  | processRequests()             | 144 |
| 5.42.2.8  | processRx()                   | 144 |
| 5.42.2.9  | setAddress()                  | 144 |
| 5.42.2.10 | setComms() [1/2]              | 144 |
| 5.42.2.11 | setComms() [2/2]              | 144 |
| 5.42.2.12 | setCommsMode()                | 145 |
| 5.42.2.13 | setTimeout()                  | 145 |
| 5.42.3    | Member Data Documentation     | 145 |
| 5.42.3.1  | oaddressFrom                  | 145 |
| 5.42.3.2  | oaddressTo                    | 145 |
| 5.42.3.3  | oapiVersion                   | 145 |
| 5.42.3.4  | ocomms                        | 145 |
| 5.42.3.5  | oerror                        | 146 |
| 5.42.3.6  | ohalfDuplex                   | 146 |
| 5.42.3.7  | olockCall                     | 146 |
| 5.42.3.8  | olockTx                       | 146 |
| 5.42.3.9  | opacketRpcCmd                 | 146 |
| 5.42.3.10 | opacketRpcDoneSema            | 146 |
| 5.42.3.11 | opacketRpcSema                | 147 |
| 5.42.3.12 | opacketRx                     | 147 |
| 5.42.3.13 | opacketRxBase                 | 147 |
| 5.42.3.14 | opacketTx                     | 147 |
| 5.42.3.15 | opacketTxBase                 | 147 |
| 5.42.3.16 | oreqSize                      | 147 |
| 5.42.3.17 | othreaded                     | 148 |
| 5.42.3.18 | otimeout                      | 148 |
| 5.43      | BoapMc1Error Struct Reference | 148 |
| 5.43.1    | Member Data Documentation     | 148 |
| 5.43.1.1  | number                        | 148 |
| 5.43.1.2  | string                        | 149 |

|                                                         |     |
|---------------------------------------------------------|-----|
| 5.44 BoapMc1Packet Class Reference . . . . .            | 149 |
| 5.44.1 Member Data Documentation . . . . .              | 149 |
| 5.44.1.1 data . . . . .                                 | 149 |
| 5.44.1.2 head . . . . .                                 | 149 |
| 5.45 BoapMc1PacketHead Struct Reference . . . . .       | 149 |
| 5.45.1 Member Data Documentation . . . . .              | 150 |
| 5.45.1.1 addressFrom . . . . .                          | 150 |
| 5.45.1.2 addressTo . . . . .                            | 150 |
| 5.45.1.3 checksum . . . . .                             | 150 |
| 5.45.1.4 cmd . . . . .                                  | 150 |
| 5.45.1.5 error . . . . .                                | 151 |
| 5.45.1.6 length . . . . .                               | 151 |
| 5.45.1.7 magic . . . . .                                | 151 |
| 5.46 BoapMcClientObject Class Reference . . . . .       | 151 |
| 5.46.1 Constructor & Destructor Documentation . . . . . | 152 |
| 5.46.1.1 BoapMcClientObject() . . . . .                 | 152 |
| 5.46.1.2 ~BoapMcClientObject() . . . . .                | 152 |
| 5.46.2 Member Function Documentation . . . . .          | 152 |
| 5.46.2.1 getApiVersion() . . . . .                      | 152 |
| 5.46.2.2 performCall() . . . . .                        | 152 |
| 5.46.2.3 performRecv() . . . . .                        | 153 |
| 5.46.2.4 performSend() . . . . .                        | 153 |
| 5.46.2.5 setAddress() . . . . .                         | 153 |
| 5.46.3 Member Data Documentation . . . . .              | 153 |
| 5.46.3.1 oaddressFrom . . . . .                         | 153 |
| 5.46.3.2 oaddressTo . . . . .                           | 153 |
| 5.46.3.3 oapiVersion . . . . .                          | 153 |
| 5.46.3.4 ocomms . . . . .                               | 154 |
| 5.46.3.5 opacket . . . . .                              | 154 |
| 5.47 BoapMcComms Class Reference . . . . .              | 154 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.47.1    | Constructor & Destructor Documentation | 155 |
| 5.47.1.1  | BoapMcComms()                          | 156 |
| 5.47.1.2  | ~BoapMcComms()                         | 156 |
| 5.47.2    | Member Function Documentation          | 156 |
| 5.47.2.1  | getApiVersion()                        | 156 |
| 5.47.2.2  | packetRecv()                           | 156 |
| 5.47.2.3  | packetSend()                           | 156 |
| 5.47.2.4  | performCall()                          | 156 |
| 5.47.2.5  | performSend()                          | 157 |
| 5.47.2.6  | processPacket()                        | 157 |
| 5.47.2.7  | processRequest()                       | 157 |
| 5.47.2.8  | processRequests()                      | 157 |
| 5.47.2.9  | processRx()                            | 157 |
| 5.47.2.10 | setAddress()                           | 158 |
| 5.47.2.11 | setComms() [1/2]                       | 158 |
| 5.47.2.12 | setComms() [2/2]                       | 158 |
| 5.47.2.13 | setCommsMode()                         | 158 |
| 5.47.2.14 | setTimeout()                           | 158 |
| 5.47.3    | Member Data Documentation              | 158 |
| 5.47.3.1  | oaddressFrom                           | 159 |
| 5.47.3.2  | oaddressTo                             | 159 |
| 5.47.3.3  | oapiVersion                            | 159 |
| 5.47.3.4  | ocomms                                 | 159 |
| 5.47.3.5  | olockCall                              | 159 |
| 5.47.3.6  | olockTx                                | 159 |
| 5.47.3.7  | opacket                                | 159 |
| 5.47.3.8  | opacketReqQueue                        | 160 |
| 5.47.3.9  | opacketReqRx                           | 160 |
| 5.47.3.10 | opacketReqTx                           | 160 |
| 5.47.3.11 | opacketRx                              | 160 |

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.47.3.12 | opacketRxSema                          | 160 |
| 5.47.3.13 | opacketTx                              | 160 |
| 5.47.3.14 | opacketTxQueue                         | 161 |
| 5.47.3.15 | opacketTxQueueWriteNum                 | 161 |
| 5.47.3.16 | opacketTxSema                          | 161 |
| 5.47.3.17 | oslave                                 | 161 |
| 5.47.3.18 | othreaded                              | 161 |
| 5.47.3.19 | otimeout                               | 161 |
| 5.48      | BoapMcPacket Class Reference           | 162 |
| 5.48.1    | Member Data Documentation              | 162 |
| 5.48.1.1  | data                                   | 162 |
| 5.48.1.2  | head                                   | 162 |
| 5.49      | BoapMcPacketHead Struct Reference      | 162 |
| 5.49.1    | Member Data Documentation              | 162 |
| 5.49.1.1  | addressFrom                            | 163 |
| 5.49.1.2  | addressTo                              | 163 |
| 5.49.1.3  | checksum                               | 163 |
| 5.49.1.4  | cmd                                    | 163 |
| 5.49.1.5  | error                                  | 163 |
| 5.49.1.6  | length                                 | 163 |
| 5.50      | BoapMcServiceObject Class Reference    | 163 |
| 5.50.1    | Constructor & Destructor Documentation | 164 |
| 5.50.1.1  | BoapMcServiceObject()                  | 164 |
| 5.50.1.2  | ~BoapMcServiceObject()                 | 164 |
| 5.50.2    | Member Function Documentation          | 164 |
| 5.50.2.1  | process()                              | 164 |
| 5.50.2.2  | processEvent()                         | 164 |
| 5.50.2.3  | sendEvent()                            | 165 |
| 5.50.3    | Member Data Documentation              | 165 |
| 5.50.3.1  | oapiVersion                            | 165 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.51     | BoapMcSignalObject Class Reference     | 165 |
| 5.51.1   | Constructor & Destructor Documentation | 165 |
| 5.51.1.1 | BoapMcSignalObject()                   | 165 |
| 5.51.2   | Member Function Documentation          | 166 |
| 5.51.2.1 | performSend()                          | 166 |
| 5.51.3   | Member Data Documentation              | 166 |
| 5.51.3.1 | ocomms                                 | 166 |
| 5.52     | BObj Class Reference                   | 166 |
| 5.52.1   | Constructor & Destructor Documentation | 166 |
| 5.52.1.1 | BObj()                                 | 166 |
| 5.52.1.2 | ~BObj()                                | 167 |
| 5.52.2   | Member Function Documentation          | 167 |
| 5.52.2.1 | getMembers()                           | 167 |
| 5.52.2.2 | getType()                              | 167 |
| 5.52.2.3 | membersPrint()                         | 167 |
| 5.53     | BObjMember Struct Reference            | 167 |
| 5.53.1   | Member Data Documentation              | 168 |
| 5.53.1.1 | dataOffset                             | 168 |
| 5.53.1.2 | name                                   | 168 |
| 5.53.1.3 | size                                   | 168 |
| 5.53.1.4 | type                                   | 168 |
| 5.53.1.5 | typeComp                               | 168 |
| 5.53.1.6 | typeName                               | 168 |
| 5.54     | BPwm Class Reference                   | 168 |
| 5.54.1   | Constructor & Destructor Documentation | 169 |
| 5.54.1.1 | BPwm()                                 | 169 |
| 5.54.2   | Member Function Documentation          | 169 |
| 5.54.2.1 | init()                                 | 169 |
| 5.54.2.2 | initChannel()                          | 169 |
| 5.54.2.3 | set()                                  | 169 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.54.3   | Member Data Documentation              | 170 |
| 5.54.3.1 | operiod                                | 170 |
| 5.54.3.2 | otim                                   | 170 |
| 5.55     | BQueue< T > Class Template Reference   | 170 |
| 5.55.1   | Detailed Description                   | 171 |
| 5.55.2   | Constructor & Destructor Documentation | 171 |
| 5.55.2.1 | BQueue()                               | 171 |
| 5.55.2.2 | ~BQueue()                              | 171 |
| 5.55.3   | Member Function Documentation          | 171 |
| 5.55.3.1 | clear()                                | 171 |
| 5.55.3.2 | read()                                 | 171 |
| 5.55.3.3 | readAvailable()                        | 172 |
| 5.55.3.4 | readAvailableFromIsr()                 | 172 |
| 5.55.3.5 | readFromIsr()                          | 172 |
| 5.55.3.6 | write()                                | 172 |
| 5.55.3.7 | writeAvailable()                       | 172 |
| 5.55.3.8 | writeFromIsr()                         | 172 |
| 5.55.4   | Member Data Documentation              | 173 |
| 5.55.4.1 | oqueue                                 | 173 |
| 5.56     | BRandom Class Reference                | 173 |
| 5.56.1   | Constructor & Destructor Documentation | 173 |
| 5.56.1.1 | BRandom()                              | 173 |
| 5.56.2   | Member Function Documentation          | 173 |
| 5.56.2.1 | init()                                 | 173 |
| 5.56.2.2 | value()                                | 174 |
| 5.57     | BRefData Struct Reference              | 174 |
| 5.57.1   | Member Data Documentation              | 174 |
| 5.57.1.1 | data                                   | 174 |
| 5.57.1.2 | len                                    | 174 |
| 5.57.1.3 | refCount                               | 175 |

|                                               |     |
|-----------------------------------------------|-----|
| 5.58 BRefString Class Reference               | 175 |
| 5.58.1 Constructor & Destructor Documentation | 175 |
| 5.58.1.1 BRefString() [1/3]                   | 176 |
| 5.58.1.2 BRefString() [2/3]                   | 176 |
| 5.58.1.3 BRefString() [3/3]                   | 176 |
| 5.58.1.4 ~BRefString()                        | 176 |
| 5.58.2 Member Function Documentation          | 176 |
| 5.58.2.1 clear()                              | 176 |
| 5.58.2.2 debugPrint()                         | 176 |
| 5.58.2.3 len()                                | 177 |
| 5.58.2.4 operator=()                          | 177 |
| 5.58.2.5 refDelete()                          | 177 |
| 5.58.2.6 refMakeWriteable()                   | 177 |
| 5.58.2.7 refNew()                             | 177 |
| 5.58.2.8 refResize()                          | 177 |
| 5.58.2.9 refWrite()                           | 177 |
| 5.58.2.10 str() [1/2]                         | 178 |
| 5.58.2.11 str() [2/2]                         | 178 |
| 5.58.3 Member Data Documentation              | 178 |
| 5.58.3.1 alwaysCopy                           | 178 |
| 5.58.3.2 oref                                 | 178 |
| 5.58.3.3 ostr                                 | 178 |
| 5.59 BRtc Class Reference                     | 178 |
| 5.59.1 Constructor & Destructor Documentation | 179 |
| 5.59.1.1 BRtc()                               | 179 |
| 5.59.2 Member Function Documentation          | 179 |
| 5.59.2.1 enterInitialisation()                | 179 |
| 5.59.2.2 getBackupRegister()                  | 180 |
| 5.59.2.3 getTime()                            | 180 |
| 5.59.2.4 init()                               | 180 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.59.2.5  | setBackupRegister()                    | 180 |
| 5.59.2.6  | setTime()                              | 180 |
| 5.59.2.7  | time()                                 | 180 |
| 5.59.2.8  | waitForSync()                          | 181 |
| 5.59.3    | Member Data Documentation              | 181 |
| 5.59.3.1  | oLock                                  | 181 |
| 5.60      | BSdio Class Reference                  | 181 |
| 5.60.1    | Member Enumeration Documentation       | 182 |
| 5.60.1.1  | CardType                               | 182 |
| 5.60.1.2  | ResponseType                           | 182 |
| 5.60.2    | Constructor & Destructor Documentation | 183 |
| 5.60.2.1  | BSdio()                                | 183 |
| 5.60.3    | Member Function Documentation          | 183 |
| 5.60.3.1  | backgroundEnable()                     | 183 |
| 5.60.3.2  | cardFullSpeed()                        | 183 |
| 5.60.3.3  | cardInit()                             | 183 |
| 5.60.3.4  | erase()                                | 183 |
| 5.60.3.5  | getCardType()                          | 184 |
| 5.60.3.6  | getStatus()                            | 184 |
| 5.60.3.7  | init()                                 | 184 |
| 5.60.3.8  | interrupt()                            | 184 |
| 5.60.3.9  | numBlocks()                            | 184 |
| 5.60.3.10 | readBlocks()                           | 184 |
| 5.60.3.11 | sendCmd()                              | 185 |
| 5.60.3.12 | sync()                                 | 185 |
| 5.60.3.13 | trim()                                 | 185 |
| 5.60.3.14 | writeBlocks()                          | 185 |
| 5.60.4    | Member Data Documentation              | 185 |
| 5.60.4.1  | oCardType                              | 185 |
| 5.60.4.2  | odma                                   | 186 |

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.60.4.3  | oend                                   | 186 |
| 5.60.4.4  | oint                                   | 186 |
| 5.60.4.5  | olock                                  | 186 |
| 5.60.4.6  | onumBlocks                             | 186 |
| 5.60.4.7  | orca                                   | 186 |
| 5.60.4.8  | osdmmc                                 | 187 |
| 5.60.4.9  | ospeed                                 | 187 |
| 5.60.4.10 | ostatus                                | 187 |
| 5.61      | BSemaphore Class Reference             | 187 |
| 5.61.1    | Detailed Description                   | 188 |
| 5.61.2    | Constructor & Destructor Documentation | 188 |
| 5.61.2.1  | BSemaphore() [1/2]                     | 188 |
| 5.61.2.2  | BSemaphore() [2/2]                     | 188 |
| 5.61.2.3  | ~BSemaphore()                          | 188 |
| 5.61.3    | Member Function Documentation          | 188 |
| 5.61.3.1  | operator=()                            | 188 |
| 5.61.3.2  | set()                                  | 188 |
| 5.61.3.3  | setFromIsr()                           | 189 |
| 5.61.3.4  | wait()                                 | 189 |
| 5.61.4    | Member Data Documentation              | 189 |
| 5.61.4.1  | osemaphore                             | 189 |
| 5.62      | BSemaphoreBool Class Reference         | 189 |
| 5.62.1    | Constructor & Destructor Documentation | 190 |
| 5.62.1.1  | BSemaphoreBool() [1/2]                 | 190 |
| 5.62.1.2  | BSemaphoreBool() [2/2]                 | 190 |
| 5.62.1.3  | ~BSemaphoreBool()                      | 190 |
| 5.62.2    | Member Function Documentation          | 190 |
| 5.62.2.1  | clear()                                | 190 |
| 5.62.2.2  | operator int()                         | 190 |
| 5.62.2.3  | operator=()                            | 191 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.62.2.4 | operator==( )                          | 191 |
| 5.62.2.5 | set( )                                 | 191 |
| 5.62.2.6 | value( )                               | 191 |
| 5.62.2.7 | wait( )                                | 191 |
| 5.62.3   | Member Data Documentation              | 191 |
| 5.62.3.1 | osema                                  | 191 |
| 5.62.3.2 | ovalue                                 | 192 |
| 5.63     | BSemaphoreCount Class Reference        | 192 |
| 5.63.1   | Constructor & Destructor Documentation | 192 |
| 5.63.1.1 | BSemaphoreCount( ) [1/2]               | 192 |
| 5.63.1.2 | BSemaphoreCount( ) [2/2]               | 193 |
| 5.63.1.3 | ~BSemaphoreCount( )                    | 193 |
| 5.63.2   | Member Function Documentation          | 193 |
| 5.63.2.1 | add( )                                 | 193 |
| 5.63.2.2 | addFromIsr( )                          | 193 |
| 5.63.2.3 | operator=( )                           | 193 |
| 5.63.2.4 | setValue( )                            | 193 |
| 5.63.2.5 | take( )                                | 194 |
| 5.63.2.6 | value( )                               | 194 |
| 5.63.2.7 | wait( )                                | 194 |
| 5.63.3   | Member Data Documentation              | 194 |
| 5.63.3.1 | osema                                  | 194 |
| 5.63.3.2 | ovalue                                 | 194 |
| 5.64     | BSnmpServer Class Reference            | 194 |
| 5.64.1   | Constructor & Destructor Documentation | 195 |
| 5.64.1.1 | BSnmpServer( )                         | 195 |
| 5.64.2   | Member Function Documentation          | 195 |
| 5.64.2.1 | init( )                                | 195 |
| 5.64.2.2 | run( )                                 | 195 |
| 5.65     | BSpi Class Reference                   | 195 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.65.1   | Member Enumeration Documentation       | 196 |
| 5.65.1.1 | Mode                                   | 196 |
| 5.65.2   | Constructor & Destructor Documentation | 196 |
| 5.65.2.1 | BSpi()                                 | 197 |
| 5.65.3   | Member Function Documentation          | 197 |
| 5.65.3.1 | init() [1/2]                           | 197 |
| 5.65.3.2 | init() [2/2]                           | 197 |
| 5.65.3.3 | initDev()                              | 197 |
| 5.65.3.4 | sendRecv()                             | 197 |
| 5.65.3.5 | setMode()                              | 198 |
| 5.65.3.6 | transact()                             | 198 |
| 5.65.3.7 | transactFull()                         | 198 |
| 5.65.4   | Member Data Documentation              | 198 |
| 5.65.4.1 | obus                                   | 198 |
| 5.65.4.2 | ocsActives                             | 198 |
| 5.65.4.3 | odmaRx                                 | 199 |
| 5.65.4.4 | odmaTx                                 | 199 |
| 5.65.4.5 | omodes                                 | 199 |
| 5.65.4.6 | opinCs                                 | 199 |
| 5.65.4.7 | ospi                                   | 199 |
| 5.65.4.8 | ouseDma                                | 199 |
| 5.66     | BString Class Reference                | 199 |
| 5.66.1   | Constructor & Destructor Documentation | 202 |
| 5.66.1.1 | BString() [1/4]                        | 202 |
| 5.66.1.2 | BString() [2/4]                        | 202 |
| 5.66.1.3 | BString() [3/4]                        | 202 |
| 5.66.1.4 | BString() [4/4]                        | 202 |
| 5.66.1.5 | ~BString()                             | 203 |
| 5.66.2   | Member Function Documentation          | 203 |
| 5.66.2.1 | add()                                  | 203 |

---

|           |                                      |     |
|-----------|--------------------------------------|-----|
| 5.66.2.2  | <code>append()</code>                | 203 |
| 5.66.2.3  | <code>clear()</code>                 | 203 |
| 5.66.2.4  | <code>compare()</code> [1/2]         | 203 |
| 5.66.2.5  | <code>compare()</code> [2/2]         | 204 |
| 5.66.2.6  | <code>convert()</code> [1/10]        | 204 |
| 5.66.2.7  | <code>convert()</code> [2/10]        | 204 |
| 5.66.2.8  | <code>convert()</code> [3/10]        | 204 |
| 5.66.2.9  | <code>convert()</code> [4/10]        | 204 |
| 5.66.2.10 | <code>convert()</code> [5/10]        | 204 |
| 5.66.2.11 | <code>convert()</code> [6/10]        | 205 |
| 5.66.2.12 | <code>convert()</code> [7/10]        | 205 |
| 5.66.2.13 | <code>convert()</code> [8/10]        | 205 |
| 5.66.2.14 | <code>convert()</code> [9/10]        | 205 |
| 5.66.2.15 | <code>convert()</code> [10/10]       | 205 |
| 5.66.2.16 | <code>convertHex()</code> [1/2]      | 206 |
| 5.66.2.17 | <code>convertHex()</code> [2/2]      | 206 |
| 5.66.2.18 | <code>copy()</code>                  | 206 |
| 5.66.2.19 | <code>debugPrint()</code>            | 206 |
| 5.66.2.20 | <code>del()</code>                   | 206 |
| 5.66.2.21 | <code>find()</code> [1/2]            | 206 |
| 5.66.2.22 | <code>find()</code> [2/2]            | 207 |
| 5.66.2.23 | <code>findReverse()</code>           | 207 |
| 5.66.2.24 | <code>hash()</code>                  | 207 |
| 5.66.2.25 | <code>insert()</code>                | 207 |
| 5.66.2.26 | <code>len()</code>                   | 207 |
| 5.66.2.27 | operator <code>const char *()</code> | 207 |
| 5.66.2.28 | operator <code>!=(())</code> [1/2]   | 208 |
| 5.66.2.29 | operator <code>!=(())</code> [2/2]   | 208 |
| 5.66.2.30 | operator <code>+(())</code> [1/2]    | 208 |
| 5.66.2.31 | operator <code>+(())</code> [2/2]    | 208 |

|                                  |     |
|----------------------------------|-----|
| 5.66.2.32 operator+={()          | 208 |
| 5.66.2.33 operator<() [1/2]      | 208 |
| 5.66.2.34 operator<() [2/2]      | 208 |
| 5.66.2.35 operator<=() [1/2]     | 209 |
| 5.66.2.36 operator<=() [2/2]     | 209 |
| 5.66.2.37 operator=()            | 209 |
| 5.66.2.38 operator==(1/2]        | 209 |
| 5.66.2.39 operator==(2/2]        | 209 |
| 5.66.2.40 operator>() [1/2]      | 209 |
| 5.66.2.41 operator>() [2/2]      | 209 |
| 5.66.2.42 operator>=() [1/2]     | 210 |
| 5.66.2.43 operator>=() [2/2]     | 210 |
| 5.66.2.44 operator[]()           | 210 |
| 5.66.2.45 pad()                  | 210 |
| 5.66.2.46 printf()               | 210 |
| 5.66.2.47 pullToken()            | 210 |
| 5.66.2.48 removeSeparators()     | 211 |
| 5.66.2.49 retFloat64()           | 211 |
| 5.66.2.50 retInt()               | 211 |
| 5.66.2.51 retStr()               | 211 |
| 5.66.2.52 retUInt()              | 211 |
| 5.66.2.53 reverse()              | 211 |
| 5.66.2.54 split()                | 212 |
| 5.66.2.55 str()                  | 212 |
| 5.66.2.56 subString()            | 212 |
| 5.66.2.57 toLower()              | 212 |
| 5.66.2.58 toUpper()              | 212 |
| 5.66.2.59 truncate()             | 212 |
| 5.66.3 Member Data Documentation | 213 |
| 5.66.3.1 orefs                   | 213 |

|                                                         |     |
|---------------------------------------------------------|-----|
| 5.67 BSys Class Reference . . . . .                     | 213 |
| 5.67.1 Constructor & Destructor Documentation . . . . . | 214 |
| 5.67.1.1 BSys() . . . . .                               | 214 |
| 5.67.2 Member Function Documentation . . . . .          | 214 |
| 5.67.2.1 init() . . . . .                               | 214 |
| 5.67.2.2 reset() . . . . .                              | 214 |
| 5.67.2.3 run() . . . . .                                | 214 |
| 5.67.2.4 setTaskHeap() . . . . .                        | 215 |
| 5.67.2.5 start() . . . . .                              | 215 |
| 5.67.2.6 systemStatePrint() . . . . .                   | 215 |
| 5.67.2.7 taskStatsPrint() . . . . .                     | 215 |
| 5.67.2.8 timeUs() . . . . .                             | 215 |
| 5.67.2.9 timeUsReset() . . . . .                        | 215 |
| 5.67.3 Member Data Documentation . . . . .              | 216 |
| 5.67.3.1 devices . . . . .                              | 216 |
| 5.67.3.2 interrupts . . . . .                           | 216 |
| 5.67.3.3 otimer . . . . .                               | 216 |
| 5.67.3.4 rtosRunning . . . . .                          | 216 |
| 5.67.3.5 taskHeap . . . . .                             | 216 |
| 5.68 BTask Class Reference . . . . .                    | 217 |
| 5.68.1 Constructor & Destructor Documentation . . . . . | 217 |
| 5.68.1.1 BTask() . . . . .                              | 218 |
| 5.68.1.2 ~BTask() . . . . .                             | 218 |
| 5.68.2 Member Function Documentation . . . . .          | 218 |
| 5.68.2.1 delayMs() . . . . .                            | 218 |
| 5.68.2.2 init() . . . . .                               | 218 |
| 5.68.2.3 run() . . . . .                                | 218 |
| 5.68.2.4 runTasks() . . . . .                           | 219 |
| 5.68.2.5 setPriority() . . . . .                        | 219 |
| 5.68.2.6 start() . . . . .                              | 219 |

---

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.68.2.7 | stop()                                 | 219 |
| 5.68.3   | Member Data Documentation              | 219 |
| 5.68.3.1 | oname                                  | 219 |
| 5.68.3.2 | opriority                              | 219 |
| 5.68.3.3 | orunning                               | 220 |
| 5.68.3.4 | ostackSize                             | 220 |
| 5.68.3.5 | otask                                  | 220 |
| 5.69     | BTim Class Reference                   | 220 |
| 5.69.1   | Member Enumeration Documentation       | 220 |
| 5.69.1.1 | Mode                                   | 220 |
| 5.69.1.2 | OcMode                                 | 221 |
| 5.69.2   | Constructor & Destructor Documentation | 221 |
| 5.69.2.1 | BTim()                                 | 221 |
| 5.69.3   | Member Function Documentation          | 221 |
| 5.69.3.1 | clear()                                | 221 |
| 5.69.3.2 | getCount()                             | 221 |
| 5.69.3.3 | init()                                 | 222 |
| 5.69.3.4 | ocInit()                               | 222 |
| 5.69.3.5 | ocSetCompare()                         | 222 |
| 5.69.3.6 | start()                                | 222 |
| 5.69.4   | Member Data Documentation              | 222 |
| 5.69.4.1 | otim                                   | 222 |
| 5.70     | BTime Class Reference                  | 223 |
| 5.70.1   | Constructor & Destructor Documentation | 223 |
| 5.70.1.1 | BTime()                                | 224 |
| 5.70.2   | Member Function Documentation          | 224 |
| 5.70.2.1 | addSeconds()                           | 224 |
| 5.70.2.2 | getDate()                              | 224 |
| 5.70.2.3 | getSeconds()                           | 224 |
| 5.70.2.4 | getString()                            | 224 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.70.2.5  | getTime()                              | 225 |
| 5.70.2.6  | isLeapYear()                           | 225 |
| 5.70.2.7  | isSet()                                | 225 |
| 5.70.2.8  | operator!=()                           | 225 |
| 5.70.2.9  | operator+()                            | 225 |
| 5.70.2.10 | operator+=()                           | 225 |
| 5.70.2.11 | operator<()                            | 226 |
| 5.70.2.12 | operator<=()                           | 226 |
| 5.70.2.13 | operator==()                           | 226 |
| 5.70.2.14 | operator>()                            | 226 |
| 5.70.2.15 | operator>=()                           | 226 |
| 5.70.2.16 | set() [1/2]                            | 226 |
| 5.70.2.17 | set() [2/2]                            | 227 |
| 5.70.2.18 | setString()                            | 227 |
| 5.70.2.19 | setYearDay()                           | 227 |
| 5.70.3    | Member Data Documentation              | 227 |
| 5.70.3.1  | otime                                  | 227 |
| 5.71      | BTimeUs Class Reference                | 228 |
| 5.71.1    | Constructor & Destructor Documentation | 229 |
| 5.71.1.1  | BTimeUs()                              | 229 |
| 5.71.2    | Member Function Documentation          | 229 |
| 5.71.2.1  | addMicroSeconds()                      | 229 |
| 5.71.2.2  | addSeconds()                           | 229 |
| 5.71.2.3  | getDate()                              | 229 |
| 5.71.2.4  | getMicroSeconds()                      | 229 |
| 5.71.2.5  | getSeconds()                           | 230 |
| 5.71.2.6  | getString()                            | 230 |
| 5.71.2.7  | getTime()                              | 230 |
| 5.71.2.8  | isLeapYear()                           | 230 |
| 5.71.2.9  | isSet()                                | 230 |

|                                               |     |
|-----------------------------------------------|-----|
| 5.71.2.10 operator!=(())                      | 230 |
| 5.71.2.11 operator+()                         | 231 |
| 5.71.2.12 operator+=()                        | 231 |
| 5.71.2.13 operator<()                         | 231 |
| 5.71.2.14 operator<=()                        | 231 |
| 5.71.2.15 operator==(())                      | 231 |
| 5.71.2.16 operator>()                         | 231 |
| 5.71.2.17 operator>=()                        | 231 |
| 5.71.2.18 set() [1/2]                         | 232 |
| 5.71.2.19 set() [2/2]                         | 232 |
| 5.71.2.20 setString()                         | 232 |
| 5.71.2.21 setYearDay()                        | 232 |
| 5.71.3 Member Data Documentation              | 232 |
| 5.71.3.1 otime                                | 233 |
| 5.72 BUart Class Reference                    | 233 |
| 5.72.1 Constructor & Destructor Documentation | 234 |
| 5.72.1.1 BUart()                              | 234 |
| 5.72.2 Member Function Documentation          | 234 |
| 5.72.2.1 flushRx()                            | 234 |
| 5.72.2.2 hold()                               | 234 |
| 5.72.2.3 init() [1/2]                         | 235 |
| 5.72.2.4 init() [2/2]                         | 235 |
| 5.72.2.5 interrupt()                          | 235 |
| 5.72.2.6 read()                               | 235 |
| 5.72.2.7 readAvailable()                      | 235 |
| 5.72.2.8 readLine()                           | 236 |
| 5.72.2.9 setMode485()                         | 236 |
| 5.72.2.10 setRtsCts()                         | 236 |
| 5.72.2.11 wait()                              | 236 |
| 5.72.2.12 write()                             | 236 |

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.72.2.13 | writeAvailable()                       | 237 |
| 5.72.3    | Member Data Documentation              | 237 |
| 5.72.3.1  | devs                                   | 237 |
| 5.72.3.2  | odev                                   | 237 |
| 5.72.3.3  | olastRxTime                            | 237 |
| 5.72.3.4  | opinCts                                | 237 |
| 5.72.3.5  | opinRts                                | 237 |
| 5.72.3.6  | orx                                    | 237 |
| 5.72.3.7  | orxDisable                             | 238 |
| 5.72.3.8  | orxFifo                                | 238 |
| 5.72.3.9  | otxEnablePin                           | 238 |
| 5.72.3.10 | otxFifo                                | 238 |
| 5.72.3.11 | ouart                                  | 238 |
| 5.72.3.12 | owait                                  | 238 |
| 5.72.3.13 | owaitReadNum                           | 238 |
| 5.73      | BUartBasic Class Reference             | 239 |
| 5.73.1    | Constructor & Destructor Documentation | 239 |
| 5.73.1.1  | BUartBasic()                           | 239 |
| 5.73.2    | Member Function Documentation          | 240 |
| 5.73.2.1  | init() [1/2]                           | 240 |
| 5.73.2.2  | init() [2/2]                           | 240 |
| 5.73.2.3  | printf()                               | 240 |
| 5.73.2.4  | read()                                 | 240 |
| 5.73.2.5  | readAvailable()                        | 241 |
| 5.73.2.6  | setMode485()                           | 241 |
| 5.73.2.7  | write()                                | 241 |
| 5.73.2.8  | writeAvailable()                       | 241 |
| 5.73.3    | Member Data Documentation              | 241 |
| 5.73.3.1  | orxDisable                             | 241 |
| 5.73.3.2  | otxEnablePin                           | 242 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.73.3.3  | uart                                   | 242 |
| 5.74      | BUsbConfig Struct Reference            | 242 |
| 5.74.1    | Member Data Documentation              | 242 |
| 5.74.1.1  | devClass                               | 242 |
| 5.74.1.2  | devProtocol                            | 242 |
| 5.74.1.3  | devSubClass                            | 243 |
| 5.74.1.4  | manufacturer                           | 243 |
| 5.74.1.5  | product                                | 243 |
| 5.74.1.6  | productId                              | 243 |
| 5.74.1.7  | serialNumber                           | 243 |
| 5.74.1.8  | vendorId                               | 243 |
| 5.75      | BUsbSerial Class Reference             | 243 |
| 5.75.1    | Constructor & Destructor Documentation | 245 |
| 5.75.1.1  | BUsbSerial()                           | 245 |
| 5.75.2    | Member Function Documentation          | 245 |
| 5.75.2.1  | close()                                | 245 |
| 5.75.2.2  | init()                                 | 246 |
| 5.75.2.3  | printf()                               | 246 |
| 5.75.2.4  | printfv()                              | 246 |
| 5.75.2.5  | read()                                 | 246 |
| 5.75.2.6  | readAvailable()                        | 246 |
| 5.75.2.7  | setBlocking()                          | 247 |
| 5.75.2.8  | setMaximumUsbChunk()                   | 247 |
| 5.75.2.9  | setSync()                              | 247 |
| 5.75.2.10 | start()                                | 247 |
| 5.75.2.11 | usbDelInit()                           | 247 |
| 5.75.2.12 | usbdInit()                             | 247 |
| 5.75.2.13 | usbdInitHardware()                     | 248 |
| 5.75.2.14 | usbdInitUsbDev()                       | 248 |
| 5.75.2.15 | usbInterrupt()                         | 248 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 5.75.2.16 | usbDRx()                               | 248 |
| 5.75.2.17 | usbDSetup()                            | 248 |
| 5.75.2.18 | usbDStarted()                          | 248 |
| 5.75.2.19 | usbDTx()                               | 249 |
| 5.75.2.20 | usbDTxComplete()                       | 249 |
| 5.75.2.21 | wait()                                 | 249 |
| 5.75.2.22 | write()                                | 249 |
| 5.75.2.23 | writeAvailable()                       | 249 |
| 5.75.3    | Member Data Documentation              | 250 |
| 5.75.3.1  | oblocking                              | 250 |
| 5.75.3.2  | oconnected                             | 250 |
| 5.75.3.3  | ohighSpeed                             | 250 |
| 5.75.3.4  | ohpcd                                  | 250 |
| 5.75.3.5  | oinitialised                           | 250 |
| 5.75.3.6  | opacketSize                            | 251 |
| 5.75.3.7  | orxBuffer                              | 251 |
| 5.75.3.8  | orxFifo                                | 251 |
| 5.75.3.9  | orxStalled                             | 251 |
| 5.75.3.10 | osync                                  | 251 |
| 5.75.3.11 | otxFifo                                | 251 |
| 5.75.3.12 | otxSending                             | 252 |
| 5.75.3.13 | otxSendLen                             | 252 |
| 5.75.3.14 | ousbChunkSize                          | 252 |
| 5.75.3.15 | ousbDev                                | 252 |
| 5.75.3.16 | owait                                  | 252 |
| 5.75.3.17 | owaitReadNum                           | 252 |
| 5.76      | BUsbSerialLock Class Reference         | 253 |
| 5.76.1    | Constructor & Destructor Documentation | 253 |
| 5.76.1.1  | BUsbSerialLock()                       | 253 |
| 5.76.2    | Member Function Documentation          | 253 |

|          |                                        |     |
|----------|----------------------------------------|-----|
| 5.76.2.1 | printfv()                              | 254 |
| 5.76.2.2 | read()                                 | 254 |
| 5.76.2.3 | readAvailable()                        | 254 |
| 5.76.2.4 | write()                                | 254 |
| 5.76.2.5 | writeAvailable()                       | 254 |
| 5.76.3   | Member Data Documentation              | 255 |
| 5.76.3.1 | oLock                                  | 255 |
| 5.77     | BWatchdog Class Reference              | 255 |
| 5.77.1   | Constructor & Destructor Documentation | 255 |
| 5.77.1.1 | BWatchdog()                            | 255 |
| 5.77.2   | Member Function Documentation          | 255 |
| 5.77.2.1 | reset()                                | 255 |
| 5.77.2.2 | start()                                | 256 |
| 5.78     | HeapTestState Struct Reference         | 256 |
| 5.78.1   | Member Data Documentation              | 256 |
| 5.78.1.1 | caller                                 | 256 |
| 5.78.1.2 | pos                                    | 256 |
| 5.78.1.3 | size                                   | 256 |
| 5.78.1.4 | type                                   | 257 |
| 5.79     | IpAddress Class Reference              | 257 |
| 5.79.1   | Constructor & Destructor Documentation | 257 |
| 5.79.1.1 | IpAddress() [1/3]                      | 257 |
| 5.79.1.2 | IpAddress() [2/3]                      | 257 |
| 5.79.1.3 | IpAddress() [3/3]                      | 258 |
| 5.79.2   | Member Function Documentation          | 258 |
| 5.79.2.1 | getString()                            | 258 |
| 5.79.2.2 | getUInt()                              | 258 |
| 5.79.2.3 | operator ip_addr_t()                   | 258 |
| 5.79.2.4 | setString()                            | 258 |
| 5.79.3   | Member Data Documentation              | 258 |
| 5.79.3.1 | oaddress                               | 258 |
| 5.80     | BList< T >::Node Class Reference       | 259 |
| 5.80.1   | Constructor & Destructor Documentation | 259 |
| 5.80.1.1 | Node()                                 | 259 |
| 5.80.2   | Member Data Documentation              | 259 |
| 5.80.2.1 | item                                   | 259 |

|                                           |            |
|-------------------------------------------|------------|
| <b>6 File Documentation</b>               | <b>261</b> |
| 6.1 BAdc.cpp File Reference               | 261        |
| 6.1.1 Macro Definition Documentation      | 262        |
| 6.1.1.1 DualMode_AlterTrig                | 262        |
| 6.1.1.2 DualMode_InjecSimult              | 262        |
| 6.1.1.3 DualMode_Interl                   | 262        |
| 6.1.1.4 DualMode_RegSimult                | 262        |
| 6.1.1.5 DualMode_RegSimult_AlterTrig      | 262        |
| 6.1.1.6 DualMode_RegSimult_InjecSimult    | 263        |
| 6.1.1.7 Mode_Independent                  | 263        |
| 6.1.1.8 Prescaler_Div2                    | 263        |
| 6.1.1.9 Prescaler_Div4                    | 263        |
| 6.1.1.10 Prescaler_Div6                   | 263        |
| 6.1.1.11 Prescaler_Div8                   | 263        |
| 6.1.1.12 SampleTime_112Cycles             | 263        |
| 6.1.1.13 SampleTime_144Cycles             | 263        |
| 6.1.1.14 SampleTime_15Cycles              | 264        |
| 6.1.1.15 SampleTime_28Cycles              | 264        |
| 6.1.1.16 SampleTime_3Cycles               | 264        |
| 6.1.1.17 SampleTime_480Cycles             | 264        |
| 6.1.1.18 SampleTime_56Cycles              | 264        |
| 6.1.1.19 SampleTime_84Cycles              | 264        |
| 6.1.1.20 TripleMode_AlterTrig             | 264        |
| 6.1.1.21 TripleMode_InjecSimult           | 264        |
| 6.1.1.22 TripleMode_Interl                | 265        |
| 6.1.1.23 TripleMode_RegSimult             | 265        |
| 6.1.1.24 TripleMode_RegSimult_AlterTrig   | 265        |
| 6.1.1.25 TripleMode_RegSimult_InjecSimult | 265        |
| 6.1.1.26 TwoSamplingDelay_10Cycles        | 265        |
| 6.1.1.27 TwoSamplingDelay_11Cycles        | 265        |

---

|          |                                |     |
|----------|--------------------------------|-----|
| 6.1.1.28 | TwoSamplingDelay_12Cycles      | 265 |
| 6.1.1.29 | TwoSamplingDelay_13Cycles      | 265 |
| 6.1.1.30 | TwoSamplingDelay_14Cycles      | 266 |
| 6.1.1.31 | TwoSamplingDelay_15Cycles      | 266 |
| 6.1.1.32 | TwoSamplingDelay_16Cycles      | 266 |
| 6.1.1.33 | TwoSamplingDelay_17Cycles      | 266 |
| 6.1.1.34 | TwoSamplingDelay_18Cycles      | 266 |
| 6.1.1.35 | TwoSamplingDelay_19Cycles      | 266 |
| 6.1.1.36 | TwoSamplingDelay_20Cycles      | 266 |
| 6.1.1.37 | TwoSamplingDelay_5Cycles       | 266 |
| 6.1.1.38 | TwoSamplingDelay_6Cycles       | 267 |
| 6.1.1.39 | TwoSamplingDelay_7Cycles       | 267 |
| 6.1.1.40 | TwoSamplingDelay_8Cycles       | 267 |
| 6.1.1.41 | TwoSamplingDelay_9Cycles       | 267 |
| 6.2      | BAdc.h File Reference          | 267 |
| 6.3      | BArray.h File Reference        | 267 |
| 6.3.1    | Typedef Documentation          | 268 |
| 6.3.1.1  | BArrayFloat32                  | 268 |
| 6.3.1.2  | BArrayFloat64                  | 268 |
| 6.3.1.3  | BArrayInt32                    | 268 |
| 6.4      | BArray_imp.h File Reference    | 268 |
| 6.5      | BBuffer.cpp File Reference     | 268 |
| 6.5.1    | Variable Documentation         | 269 |
| 6.5.1.1  | roundSize                      | 269 |
| 6.6      | BBuffer.h File Reference       | 269 |
| 6.7      | BCan.cpp File Reference        | 269 |
| 6.7.1    | Macro Definition Documentation | 270 |
| 6.7.1.1  | USE_INTERRUPTS                 | 270 |
| 6.7.2    | Function Documentation         | 270 |
| 6.7.2.1  | CAN1_RX0_IRQHandler()          | 270 |

---

|          |                             |     |
|----------|-----------------------------|-----|
| 6.7.2.2  | CAN2_RX0_IRQHandler()       | 270 |
| 6.7.3    | Variable Documentation      | 270 |
| 6.7.3.1  | devs                        | 270 |
| 6.8      | BCan.h File Reference       | 270 |
| 6.9      | BComms.cpp File Reference   | 271 |
| 6.10     | BComms.h File Reference     | 271 |
| 6.11     | BComplex.h File Reference   | 271 |
| 6.11.1   | Typedef Documentation       | 271 |
| 6.11.1.1 | BComplex                    | 271 |
| 6.11.1.2 | BComplex32                  | 271 |
| 6.11.1.3 | BComplex64                  | 272 |
| 6.12     | BCondInt.cpp File Reference | 272 |
| 6.13     | BCondInt.h File Reference   | 272 |
| 6.14     | BConfig.h File Reference    | 272 |
| 6.15     | BConvert.cpp File Reference | 272 |
| 6.15.1   | Function Documentation      | 273 |
| 6.15.1.1 | base64Decode()              | 273 |
| 6.15.2   | Variable Documentation      | 273 |
| 6.15.2.1 | base64_decode_table         | 273 |
| 6.16     | BConvert.h File Reference   | 273 |
| 6.16.1   | Function Documentation      | 273 |
| 6.16.1.1 | base64Decode()              | 274 |
| 6.17     | BCrc16.cpp File Reference   | 274 |
| 6.17.1   | Function Documentation      | 274 |
| 6.17.1.1 | bcrc16()                    | 274 |
| 6.17.2   | Variable Documentation      | 274 |
| 6.17.2.1 | table_crc_hi                | 275 |
| 6.17.2.2 | table_crc_lo                | 275 |
| 6.18     | BCrc16.h File Reference     | 276 |
| 6.18.1   | Function Documentation      | 276 |

---

|          |                                |     |
|----------|--------------------------------|-----|
| 6.18.1.1 | bcrc16()                       | 276 |
| 6.19     | BCrc32.cpp File Reference      | 276 |
| 6.19.1   | Function Documentation         | 276 |
| 6.19.1.1 | bcrc32()                       | 276 |
| 6.19.2   | Variable Documentation         | 277 |
| 6.19.2.1 | crc32_tab                      | 277 |
| 6.20     | BCrc32.h File Reference        | 277 |
| 6.20.1   | Function Documentation         | 277 |
| 6.20.1.1 | bcrc32()                       | 277 |
| 6.21     | BCritical.h File Reference     | 277 |
| 6.22     | BCritical.inc File Reference   | 278 |
| 6.23     | BDebug.cpp File Reference      | 278 |
| 6.23.1   | Function Documentation         | 278 |
| 6.23.1.1 | bhd32()                        | 278 |
| 6.23.1.2 | bhd8()                         | 278 |
| 6.23.1.3 | bhd8a()                        | 278 |
| 6.23.1.4 | bhda32()                       | 279 |
| 6.23.1.5 | bhda8()                        | 279 |
| 6.24     | BDebug.h File Reference        | 279 |
| 6.24.1   | Macro Definition Documentation | 279 |
| 6.24.1.1 | dl1printf                      | 279 |
| 6.24.1.2 | dl2printf                      | 280 |
| 6.24.1.3 | dl3printf                      | 280 |
| 6.24.1.4 | dl4printf                      | 280 |
| 6.24.1.5 | dprintf                        | 280 |
| 6.24.2   | Function Documentation         | 280 |
| 6.24.2.1 | bhd32()                        | 280 |
| 6.24.2.2 | bhd8()                         | 281 |
| 6.24.2.3 | bhd8a()                        | 281 |
| 6.24.2.4 | bhda8()                        | 281 |

---

---

|          |                                |     |
|----------|--------------------------------|-----|
| 6.24.2.5 | bhds32()                       | 281 |
| 6.25     | BDevices.cpp File Reference    | 281 |
| 6.25.1   | Macro Definition Documentation | 282 |
| 6.25.1.1 | dlprintf                       | 282 |
| 6.25.1.2 | LDEBUG                         | 282 |
| 6.25.2   | Variable Documentation         | 282 |
| 6.25.2.1 | HSI_CLOCK                      | 282 |
| 6.26     | BDevices.h File Reference      | 282 |
| 6.26.1   | Enumeration Type Documentation | 283 |
| 6.26.1.1 | Device                         | 283 |
| 6.27     | BDict.cpp File Reference       | 285 |
| 6.27.1   | Function Documentation         | 285 |
| 6.27.1.1 | bdictStringToString()          | 285 |
| 6.27.1.2 | fromBString()                  | 286 |
| 6.27.1.3 | toBString()                    | 286 |
| 6.28     | BDict.h File Reference         | 286 |
| 6.28.1   | Typedef Documentation          | 286 |
| 6.28.1.1 | BDictString                    | 286 |
| 6.28.2   | Function Documentation         | 287 |
| 6.28.2.1 | bdictStringToString()          | 287 |
| 6.28.2.2 | fromBString()                  | 287 |
| 6.28.2.3 | toBString()                    | 287 |
| 6.29     | BDma.cpp File Reference        | 287 |
| 6.29.1   | Macro Definition Documentation | 288 |
| 6.29.1.1 | dlprintf                       | 288 |
| 6.29.1.2 | dlprintStatus                  | 288 |
| 6.29.1.3 | LDEBUG                         | 288 |
| 6.29.2   | Function Documentation         | 288 |
| 6.29.2.1 | DMA1_Stream0_IRQHandler()      | 289 |
| 6.29.2.2 | DMA1_Stream1_IRQHandler()      | 289 |

---

---

|           |                            |     |
|-----------|----------------------------|-----|
| 6.29.2.3  | DMA1_Stream2_IRQHandler()  | 289 |
| 6.29.2.4  | DMA1_Stream3_IRQHandler()  | 289 |
| 6.29.2.5  | DMA1_Stream4_IRQHandler()  | 289 |
| 6.29.2.6  | DMA1_Stream5_IRQHandler()  | 289 |
| 6.29.2.7  | DMA1_Stream6_IRQHandler()  | 289 |
| 6.29.2.8  | DMA1_Stream7_IRQHandler()  | 289 |
| 6.29.2.9  | DMA2_Stream0_IRQHandler()  | 290 |
| 6.29.2.10 | DMA2_Stream1_IRQHandler()  | 290 |
| 6.29.2.11 | DMA2_Stream2_IRQHandler()  | 290 |
| 6.29.2.12 | DMA2_Stream3_IRQHandler()  | 290 |
| 6.29.2.13 | DMA2_Stream4_IRQHandler()  | 290 |
| 6.29.2.14 | DMA2_Stream5_IRQHandler()  | 290 |
| 6.29.2.15 | DMA2_Stream6_IRQHandler()  | 290 |
| 6.29.2.16 | DMA2_Stream7_IRQHandler()  | 290 |
| 6.29.3    | Variable Documentation     | 291 |
| 6.29.3.1  | dmaStreams                 | 291 |
| 6.30      | BDma.h File Reference      | 291 |
| 6.30.1    | Typedef Documentation      | 291 |
| 6.30.1.1  | BDmaCompleteCallback       | 291 |
| 6.31      | BEndian.cpp File Reference | 291 |
| 6.31.1    | Function Documentation     | 292 |
| 6.31.1.1  | bswap_copy()               | 292 |
| 6.32      | BEndian.h File Reference   | 292 |
| 6.32.1    | Function Documentation     | 292 |
| 6.32.1.1  | bswap_copy()               | 292 |
| 6.32.1.2  | bswap_p16()                | 292 |
| 6.32.1.3  | bswap_p32()                | 293 |
| 6.32.1.4  | bswap_p64()                | 293 |
| 6.32.1.5  | bswap_p8()                 | 293 |
| 6.33      | BError.cpp File Reference  | 293 |

---

- 6.34 BError.h File Reference . . . . . 293
  - 6.34.1 Enumeration Type Documentation . . . . . 293
    - 6.34.1.1 BErrorNum . . . . . 293
- 6.35 BEvent.cpp File Reference . . . . . 294
- 6.36 BEvent.h File Reference . . . . . 294
  - 6.36.1 Typedef Documentation . . . . . 295
    - 6.36.1.1 BEventQueue . . . . . 295
- 6.37 BFifo.h File Reference . . . . . 295
- 6.38 BFifo.inc File Reference . . . . . 295
- 6.39 BFirmware.cpp File Reference . . . . . 295
  - 6.39.1 Macro Definition Documentation . . . . . 295
    - 6.39.1.1 BDEBUGL1 . . . . . 296
  - 6.39.2 Function Documentation . . . . . 296
    - 6.39.2.1 bfirmwareBoot() . . . . . 296
    - 6.39.2.2 bfirmwareValid() . . . . . 296
- 6.40 BFirmware.h File Reference . . . . . 296
  - 6.40.1 Function Documentation . . . . . 296
    - 6.40.1.1 bfirmwareBoot() . . . . . 297
    - 6.40.1.2 bfirmwareValid() . . . . . 297
  - 6.40.2 Variable Documentation . . . . . 297
    - 6.40.2.1 BFirmwareInfoEncrypt1 . . . . . 297
    - 6.40.2.2 BFirmwareInfoMagic . . . . . 297
- 6.41 BFlash.cpp File Reference . . . . . 297
  - 6.41.1 Macro Definition Documentation . . . . . 298
    - 6.41.1.1 dlprintf . . . . . 298
    - 6.41.1.2 FLASH\_KEY1 . . . . . 298
    - 6.41.1.3 FLASH\_KEY2 . . . . . 298
    - 6.41.1.4 FLASH\_OPT\_KEY1 . . . . . 298
    - 6.41.1.5 FLASH\_OPT\_KEY2 . . . . . 298
    - 6.41.1.6 LDEBUG . . . . . 298

|          |                                |     |
|----------|--------------------------------|-----|
| 6.41.1.7 | RDP_KEY                        | 298 |
| 6.42     | BFlash.h File Reference        | 299 |
| 6.43     | BGpio.cpp File Reference       | 299 |
| 6.43.1   | Function Documentation         | 299 |
| 6.43.1.1 | bank()                         | 299 |
| 6.44     | BGpio.h File Reference         | 299 |
| 6.44.1   | Enumeration Type Documentation | 301 |
| 6.44.1.1 | Pin                            | 301 |
| 6.44.1.2 | PinState                       | 304 |
| 6.44.2   | Function Documentation         | 304 |
| 6.44.2.1 | bitsSet() [1/3]                | 304 |
| 6.44.2.2 | bitsSet() [2/3]                | 305 |
| 6.44.2.3 | bitsSet() [3/3]                | 305 |
| 6.44.3   | Variable Documentation         | 305 |
| 6.44.3.1 | gpio                           | 305 |
| 6.45     | BHeap.cpp File Reference       | 305 |
| 6.46     | BHeap.h File Reference         | 305 |
| 6.47     | BHeapTest.cpp File Reference   | 306 |
| 6.47.1   | Function Documentation         | 306 |
| 6.47.1.1 | _calloc_r()                    | 306 |
| 6.47.1.2 | _free_r()                      | 306 |
| 6.47.1.3 | _malloc_r()                    | 307 |
| 6.47.1.4 | _realloc_r()                   | 307 |
| 6.47.1.5 | heapTestAlloc()                | 307 |
| 6.47.1.6 | heapTestPrint()                | 307 |
| 6.47.2   | Variable Documentation         | 307 |
| 6.47.2.1 | heapTestMem                    | 307 |
| 6.47.2.2 | heapTestMemPos                 | 307 |
| 6.47.2.3 | heapTestMemSize                | 308 |
| 6.47.2.4 | heapTestState                  | 308 |

|          |                                          |     |
|----------|------------------------------------------|-----|
| 6.47.2.5 | heapTestStateNext . . . . .              | 308 |
| 6.48     | BHeapTest.h File Reference . . . . .     | 308 |
| 6.48.1   | Function Documentation . . . . .         | 308 |
| 6.48.1.1 | heapTestPrint() . . . . .                | 308 |
| 6.48.2   | Variable Documentation . . . . .         | 308 |
| 6.48.2.1 | heapTestMem . . . . .                    | 309 |
| 6.48.2.2 | heapTestMemSize . . . . .                | 309 |
| 6.49     | BHtml.cpp File Reference . . . . .       | 309 |
| 6.49.1   | Function Documentation . . . . .         | 309 |
| 6.49.1.1 | htmlDecode() . . . . .                   | 309 |
| 6.49.1.2 | htmlEncode() . . . . .                   | 309 |
| 6.49.1.3 | urlDecode() . . . . .                    | 309 |
| 6.49.1.4 | urlEncode() . . . . .                    | 310 |
| 6.49.1.5 | urlEncodeComponent() . . . . .           | 310 |
| 6.50     | BHtml.h File Reference . . . . .         | 310 |
| 6.50.1   | Typedef Documentation . . . . .          | 310 |
| 6.50.1.1 | HtmlWriteFunc . . . . .                  | 310 |
| 6.50.2   | Function Documentation . . . . .         | 310 |
| 6.50.2.1 | htmlDecode() . . . . .                   | 311 |
| 6.50.2.2 | htmlEncode() . . . . .                   | 311 |
| 6.50.2.3 | urlDecode() . . . . .                    | 311 |
| 6.50.2.4 | urlEncode() . . . . .                    | 311 |
| 6.50.2.5 | urlEncodeComponent() . . . . .           | 311 |
| 6.51     | BHttpServer.cpp File Reference . . . . . | 311 |
| 6.51.1   | Macro Definition Documentation . . . . . | 312 |
| 6.51.1.1 | BDEBUGL1 . . . . .                       | 312 |
| 6.51.1.2 | BDEBUGL2 . . . . .                       | 312 |
| 6.51.1.3 | BDEBUGL3 . . . . .                       | 312 |
| 6.51.1.4 | BDEBUGL4 . . . . .                       | 312 |
| 6.51.2   | Function Documentation . . . . .         | 312 |

---

|           |                                               |     |
|-----------|-----------------------------------------------|-----|
| 6.51.2.1  | findCRNL()                                    | 312 |
| 6.51.3    | Variable Documentation                        | 313 |
| 6.51.3.1  | Timeout                                       | 313 |
| 6.52      | BHttpServer.h File Reference                  | 313 |
| 6.53      | BI2c.cpp File Reference                       | 313 |
| 6.53.1    | Macro Definition Documentation                | 314 |
| 6.53.1.1  | EVENT_MASTER_BYTE_RECEIVED                    | 314 |
| 6.53.1.2  | EVENT_MASTER_BYTE_TRANSMITTED                 | 314 |
| 6.53.1.3  | EVENT_MASTER_BYTE_TRANSMITTING                | 314 |
| 6.53.1.4  | EVENT_MASTER_MODE_ADDRESS10                   | 314 |
| 6.53.1.5  | EVENT_MASTER_MODE_SELECT                      | 314 |
| 6.53.1.6  | EVENT_MASTER_RECEIVER_MODE_SELECTED           | 314 |
| 6.53.1.7  | EVENT_MASTER_TRANSMITTER_MODE_SELECTED        | 315 |
| 6.53.1.8  | EVENT_SLAVE_GENERALCALLADDRESS_MATCHED        | 315 |
| 6.53.1.9  | EVENT_SLAVE_RECEIVER_ADDRESS_MATCHED          | 315 |
| 6.53.1.10 | EVENT_SLAVE_RECEIVER_SECONDADDRESS_MATCHED    | 315 |
| 6.53.1.11 | EVENT_SLAVE_TRANSMITTER_ADDRESS_MATCHED       | 315 |
| 6.53.1.12 | EVENT_SLAVE_TRANSMITTER_SECONDADDRESS_MATCHED | 315 |
| 6.53.1.13 | I2CTYPE_0                                     | 315 |
| 6.53.2    | Variable Documentation                        | 315 |
| 6.53.2.1  | gdevs                                         | 316 |
| 6.54      | BI2c.h File Reference                         | 316 |
| 6.55      | BIInterrupt.cpp File Reference                | 316 |
| 6.55.1    | Macro Definition Documentation                | 316 |
| 6.55.1.1  | dlprintf                                      | 316 |
| 6.55.1.2  | LDEBUG                                        | 317 |
| 6.56      | BIInterrupt.h File Reference                  | 317 |
| 6.57      | BList.h File Reference                        | 317 |
| 6.57.1    | Macro Definition Documentation                | 317 |
| 6.57.1.1  | BListLoop                                     | 317 |

---

|                                                 |     |
|-------------------------------------------------|-----|
| 6.58 BList_func.h File Reference . . . . .      | 318 |
| 6.59 BMutex.cpp File Reference . . . . .        | 318 |
| 6.60 BMutex.h File Reference . . . . .          | 318 |
| 6.61 BNameValue.h File Reference . . . . .      | 318 |
| 6.62 BNetwork.cpp File Reference . . . . .      | 319 |
| 6.62.1 Macro Definition Documentation . . . . . | 319 |
| 6.62.1.1 dlprintf . . . . .                     | 320 |
| 6.62.1.2 LDEBUG . . . . .                       | 320 |
| 6.62.1.3 TX_ZERO_COPY . . . . .                 | 320 |
| 6.62.2 Function Documentation . . . . .         | 320 |
| 6.62.2.1 ETH_IRQHandler() . . . . .             | 320 |
| 6.62.2.2 ether_init() . . . . .                 | 320 |
| 6.62.2.3 ether_link_changed() . . . . .         | 320 |
| 6.62.2.4 ether_send() . . . . .                 | 321 |
| 6.62.2.5 HAL_ETH_RxCpltCallback() . . . . .     | 321 |
| 6.62.3 Variable Documentation . . . . .         | 321 |
| 6.62.3.1 bnetwork . . . . .                     | 321 |
| 6.62.3.2 DhcpTries . . . . .                    | 321 |
| 6.62.3.3 dmaRxDscrTab . . . . .                 | 321 |
| 6.62.3.4 dmaTxDscrTab . . . . .                 | 321 |
| 6.62.3.5 ethPhySettings . . . . .               | 321 |
| 6.62.3.6 EventEthernet . . . . .                | 322 |
| 6.62.3.7 EventLinkChanged . . . . .             | 322 |
| 6.62.3.8 PhyDP83848_CR . . . . .                | 322 |
| 6.62.3.9 PhyDP83848_LEDCCR . . . . .            | 322 |
| 6.62.3.10 ProcessTaskPriority . . . . .         | 322 |
| 6.62.3.11 rxBuffer . . . . .                    | 322 |
| 6.62.3.12 txBuffer . . . . .                    | 322 |
| 6.63 BNetwork.h File Reference . . . . .        | 323 |
| 6.63.1 Typedef Documentation . . . . .          | 323 |

|          |                                |     |
|----------|--------------------------------|-----|
| 6.63.1.1 | IpPacket                       | 323 |
| 6.63.2   | Enumeration Type Documentation | 323 |
| 6.63.2.1 | DhcpState                      | 323 |
| 6.64     | BNtpClient.cpp File Reference  | 324 |
| 6.64.1   | Macro Definition Documentation | 324 |
| 6.64.1.1 | BDEBUGL1                       | 324 |
| 6.64.2   | Function Documentation         | 324 |
| 6.64.2.1 | sntp_set_system_time()         | 325 |
| 6.64.3   | Variable Documentation         | 325 |
| 6.64.3.1 | ontpClient                     | 325 |
| 6.65     | BNtpClient.h File Reference    | 325 |
| 6.66     | BoapMc.cpp File Reference      | 325 |
| 6.66.1   | Macro Definition Documentation | 325 |
| 6.66.1.1 | DEBUG_LOCAL                    | 326 |
| 6.66.1.2 | DEBUG_LOCAL1                   | 326 |
| 6.66.1.3 | dl1printf                      | 326 |
| 6.66.1.4 | dlprintf                       | 326 |
| 6.67     | BoapMc.h File Reference        | 326 |
| 6.67.1   | Enumeration Type Documentation | 327 |
| 6.67.1.1 | BoapMcType                     | 327 |
| 6.67.2   | Function Documentation         | 327 |
| 6.67.2.1 | __attribute__()                | 327 |
| 6.67.3   | Variable Documentation         | 327 |
| 6.67.3.1 | __attribute__                  | 327 |
| 6.67.3.2 | addressFrom                    | 328 |
| 6.67.3.3 | addressTo                      | 328 |
| 6.67.3.4 | checksum                       | 328 |
| 6.67.3.5 | cmd                            | 328 |
| 6.67.3.6 | error                          | 328 |
| 6.67.3.7 | length                         | 328 |

|                                          |     |
|------------------------------------------|-----|
| 6.68 BoapMc1.cpp File Reference          | 328 |
| 6.68.1 Macro Definition Documentation    | 329 |
| 6.68.1.1 BDEBUGL1                        | 329 |
| 6.68.1.2 BDEBUGL2                        | 329 |
| 6.69 BoapMc1.h File Reference            | 329 |
| 6.69.1 Enumeration Type Documentation    | 330 |
| 6.69.1.1 BoapMc1Type                     | 330 |
| 6.69.2 Function Documentation            | 330 |
| 6.69.2.1 __attribute__()                 | 330 |
| 6.69.2.2 boapMc1CommsRoundupLen()        | 331 |
| 6.69.3 Variable Documentation            | 331 |
| 6.69.3.1 __attribute__                   | 331 |
| 6.69.3.2 addressFrom                     | 331 |
| 6.69.3.3 addressTo                       | 331 |
| 6.69.3.4 BoapMc1Magic                    | 331 |
| 6.69.3.5 checksum                        | 331 |
| 6.69.3.6 cmd                             | 332 |
| 6.69.3.7 data                            | 332 |
| 6.69.3.8 error                           | 332 |
| 6.69.3.9 head                            | 332 |
| 6.69.3.10 length                         | 332 |
| 6.69.3.11 magic                          | 332 |
| 6.69.3.12 number                         | 332 |
| 6.69.3.13 string                         | 333 |
| 6.70 BObj.cpp File Reference             | 333 |
| 6.71 BObj.h File Reference               | 333 |
| 6.72 BObjStringFormat.cpp File Reference | 333 |
| 6.72.1 Function Documentation            | 334 |
| 6.72.1.1 toBDictStringFromJson()         | 334 |
| 6.72.1.2 toBString() [1/18]              | 334 |

---

|           |                                        |     |
|-----------|----------------------------------------|-----|
| 6.72.1.3  | <a href="#">toString() [2/18]</a>      | 334 |
| 6.72.1.4  | <a href="#">toString() [3/18]</a>      | 335 |
| 6.72.1.5  | <a href="#">toString() [4/18]</a>      | 335 |
| 6.72.1.6  | <a href="#">toString() [5/18]</a>      | 335 |
| 6.72.1.7  | <a href="#">toString() [6/18]</a>      | 335 |
| 6.72.1.8  | <a href="#">toString() [7/18]</a>      | 335 |
| 6.72.1.9  | <a href="#">toString() [8/18]</a>      | 335 |
| 6.72.1.10 | <a href="#">toString() [9/18]</a>      | 336 |
| 6.72.1.11 | <a href="#">toString() [10/18]</a>     | 336 |
| 6.72.1.12 | <a href="#">toString() [11/18]</a>     | 336 |
| 6.72.1.13 | <a href="#">toString() [12/18]</a>     | 336 |
| 6.72.1.14 | <a href="#">toString() [13/18]</a>     | 336 |
| 6.72.1.15 | <a href="#">toString() [14/18]</a>     | 336 |
| 6.72.1.16 | <a href="#">toString() [15/18]</a>     | 337 |
| 6.72.1.17 | <a href="#">toString() [16/18]</a>     | 337 |
| 6.72.1.18 | <a href="#">toString() [17/18]</a>     | 337 |
| 6.72.1.19 | <a href="#">toString() [18/18]</a>     | 337 |
| 6.72.1.20 | <a href="#">toStringJson() [1/18]</a>  | 337 |
| 6.72.1.21 | <a href="#">toStringJson() [2/18]</a>  | 337 |
| 6.72.1.22 | <a href="#">toStringJson() [3/18]</a>  | 338 |
| 6.72.1.23 | <a href="#">toStringJson() [4/18]</a>  | 338 |
| 6.72.1.24 | <a href="#">toStringJson() [5/18]</a>  | 338 |
| 6.72.1.25 | <a href="#">toStringJson() [6/18]</a>  | 338 |
| 6.72.1.26 | <a href="#">toStringJson() [7/18]</a>  | 338 |
| 6.72.1.27 | <a href="#">toStringJson() [8/18]</a>  | 338 |
| 6.72.1.28 | <a href="#">toStringJson() [9/18]</a>  | 339 |
| 6.72.1.29 | <a href="#">toStringJson() [10/18]</a> | 339 |
| 6.72.1.30 | <a href="#">toStringJson() [11/18]</a> | 339 |
| 6.72.1.31 | <a href="#">toStringJson() [12/18]</a> | 339 |
| 6.72.1.32 | <a href="#">toStringJson() [13/18]</a> | 339 |

|                                        |     |
|----------------------------------------|-----|
| 6.72.1.33 toBStringJson() [14/18]      | 339 |
| 6.72.1.34 toBStringJson() [15/18]      | 340 |
| 6.72.1.35 toBStringJson() [16/18]      | 340 |
| 6.72.1.36 toBStringJson() [17/18]      | 340 |
| 6.72.1.37 toBStringJson() [18/18]      | 340 |
| 6.73 BObjStringFormat.h File Reference | 340 |
| 6.73.1 Function Documentation          | 341 |
| 6.73.1.1 base64_decode()               | 341 |
| 6.73.1.2 base64_encode()               | 342 |
| 6.73.1.3 toBDictStringFromJson()       | 342 |
| 6.73.1.4 toBString() [1/18]            | 342 |
| 6.73.1.5 toBString() [2/18]            | 342 |
| 6.73.1.6 toBString() [3/18]            | 342 |
| 6.73.1.7 toBString() [4/18]            | 342 |
| 6.73.1.8 toBString() [5/18]            | 343 |
| 6.73.1.9 toBString() [6/18]            | 343 |
| 6.73.1.10 toBString() [7/18]           | 343 |
| 6.73.1.11 toBString() [8/18]           | 343 |
| 6.73.1.12 toBString() [9/18]           | 343 |
| 6.73.1.13 toBString() [10/18]          | 343 |
| 6.73.1.14 toBString() [11/18]          | 344 |
| 6.73.1.15 toBString() [12/18]          | 344 |
| 6.73.1.16 toBString() [13/18]          | 344 |
| 6.73.1.17 toBString() [14/18]          | 344 |
| 6.73.1.18 toBString() [15/18]          | 344 |
| 6.73.1.19 toBString() [16/18]          | 344 |
| 6.73.1.20 toBString() [17/18]          | 345 |
| 6.73.1.21 toBString() [18/18]          | 345 |
| 6.73.1.22 toBStringJson() [1/18]       | 345 |
| 6.73.1.23 toBStringJson() [2/18]       | 345 |

|                                       |     |
|---------------------------------------|-----|
| 6.73.1.24 toBStringJson() [3/18]      | 345 |
| 6.73.1.25 toBStringJson() [4/18]      | 345 |
| 6.73.1.26 toBStringJson() [5/18]      | 346 |
| 6.73.1.27 toBStringJson() [6/18]      | 346 |
| 6.73.1.28 toBStringJson() [7/18]      | 346 |
| 6.73.1.29 toBStringJson() [8/18]      | 346 |
| 6.73.1.30 toBStringJson() [9/18]      | 346 |
| 6.73.1.31 toBStringJson() [10/18]     | 346 |
| 6.73.1.32 toBStringJson() [11/18]     | 347 |
| 6.73.1.33 toBStringJson() [12/18]     | 347 |
| 6.73.1.34 toBStringJson() [13/18]     | 347 |
| 6.73.1.35 toBStringJson() [14/18]     | 347 |
| 6.73.1.36 toBStringJson() [15/18]     | 347 |
| 6.73.1.37 toBStringJson() [16/18]     | 347 |
| 6.73.1.38 toBStringJson() [17/18]     | 348 |
| 6.73.1.39 toBStringJson() [18/18]     | 348 |
| 6.74 BPwm.cpp File Reference          | 348 |
| 6.75 BPwm.h File Reference            | 348 |
| 6.76 BQueue.cpp File Reference        | 348 |
| 6.77 BQueue.h File Reference          | 348 |
| 6.77.1 Typedef Documentation          | 349 |
| 6.77.1.1 BQueueInt                    | 349 |
| 6.78 BRandom.cpp File Reference       | 349 |
| 6.79 BRandom.h File Reference         | 349 |
| 6.80 BRtc.cpp File Reference          | 349 |
| 6.80.1 Macro Definition Documentation | 350 |
| 6.80.1.1 dlprintf                     | 350 |
| 6.80.1.2 LDEBUG                       | 350 |
| 6.80.1.3 USE_SYNC                     | 350 |
| 6.80.2 Function Documentation         | 350 |

---

|           |                                |     |
|-----------|--------------------------------|-----|
| 6.80.2.1  | fromBcd()                      | 350 |
| 6.80.2.2  | toBcd()                        | 351 |
| 6.81      | BRtc.h File Reference          | 351 |
| 6.82      | BSdio.cpp File Reference       | 351 |
| 6.82.1    | Macro Definition Documentation | 353 |
| 6.82.1.1  | BDEBUGL1                       | 353 |
| 6.82.1.2  | BDEBUGL2                       | 353 |
| 6.82.1.3  | SD_CMD_ALL_SEND_CID            | 353 |
| 6.82.1.4  | SD_CMD_APP_CMD                 | 353 |
| 6.82.1.5  | SD_CMD_APP_SD_SET_BUSWIDTH     | 353 |
| 6.82.1.6  | SD_CMD_CLR_WRITE_PROT          | 353 |
| 6.82.1.7  | SD_CMD_ERASE                   | 353 |
| 6.82.1.8  | SD_CMD_ERASE_GRP_END           | 354 |
| 6.82.1.9  | SD_CMD_ERASE_GRP_START         | 354 |
| 6.82.1.10 | SD_CMD_FAST_IO                 | 354 |
| 6.82.1.11 | SD_CMD_GEN_CMD                 | 354 |
| 6.82.1.12 | SD_CMD_GO_IDLE_STATE           | 354 |
| 6.82.1.13 | SD_CMD_GO_INACTIVE_STATE       | 354 |
| 6.82.1.14 | SD_CMD_GO_IRQ_STATE            | 354 |
| 6.82.1.15 | SD_CMD_HS_BUSTEST_READ         | 354 |
| 6.82.1.16 | SD_CMD_HS_BUSTEST_WRITE        | 355 |
| 6.82.1.17 | SD_CMD_HS_SEND_EXT_CSD         | 355 |
| 6.82.1.18 | SD_CMD_HS_SWITCH               | 355 |
| 6.82.1.19 | SD_CMD_LOCK_UNLOCK             | 355 |
| 6.82.1.20 | SD_CMD_NO_CMD                  | 355 |
| 6.82.1.21 | SD_CMD_PROG_CID                | 355 |
| 6.82.1.22 | SD_CMD_PROG_CSD                | 355 |
| 6.82.1.23 | SD_CMD_READ_DAT_UNTIL_STOP     | 355 |
| 6.82.1.24 | SD_CMD_READ_MULT_BLOCK         | 356 |
| 6.82.1.25 | SD_CMD_READ_SINGLE_BLOCK       | 356 |

---

|                                                         |     |
|---------------------------------------------------------|-----|
| 6.82.1.26 SD_CMD_SD_APP_OP_COND . . . . .               | 356 |
| 6.82.1.27 SD_CMD_SD_APP_SEND_NUM_WRITE_BLOCKS . . . . . | 356 |
| 6.82.1.28 SD_CMD_SD_APP_SEND_SCR . . . . .              | 356 |
| 6.82.1.29 SD_CMD_SD_APP_SET_CLR_CARD_DETECT . . . . .   | 356 |
| 6.82.1.30 SD_CMD_SD_APP_STAUS . . . . .                 | 356 |
| 6.82.1.31 SD_CMD_SD_ERASE_GRP_END . . . . .             | 356 |
| 6.82.1.32 SD_CMD_SD_ERASE_GRP_START . . . . .           | 357 |
| 6.82.1.33 SD_CMD_SDIO_RW_DIRECT . . . . .               | 357 |
| 6.82.1.34 SD_CMD_SDIO_RW_EXTENDED . . . . .             | 357 |
| 6.82.1.35 SD_CMD_SDIO_SEN_OP_COND . . . . .             | 357 |
| 6.82.1.36 SD_CMD_SEL_DESEL_CARD . . . . .               | 357 |
| 6.82.1.37 SD_CMD_SEND_CID . . . . .                     | 357 |
| 6.82.1.38 SD_CMD_SEND_CSD . . . . .                     | 357 |
| 6.82.1.39 SD_CMD_SEND_OP_COND . . . . .                 | 357 |
| 6.82.1.40 SD_CMD_SEND_STATUS . . . . .                  | 358 |
| 6.82.1.41 SD_CMD_SEND_WRITE_PROT . . . . .              | 358 |
| 6.82.1.42 SD_CMD_SET_BLOCK_COUNT . . . . .              | 358 |
| 6.82.1.43 SD_CMD_SET_BLOCKLEN . . . . .                 | 358 |
| 6.82.1.44 SD_CMD_SET_DSR . . . . .                      | 358 |
| 6.82.1.45 SD_CMD_SET_REL_ADDR . . . . .                 | 358 |
| 6.82.1.46 SD_CMD_SET_WRITE_PROT . . . . .               | 358 |
| 6.82.1.47 SD_CMD_STOP_TRANSMISSION . . . . .            | 358 |
| 6.82.1.48 SD_CMD_WRITE_DAT_UNTIL_STOP . . . . .         | 359 |
| 6.82.1.49 SD_CMD_WRITE_MULT_BLOCK . . . . .             | 359 |
| 6.82.1.50 SD_CMD_WRITE_SINGLE_BLOCK . . . . .           | 359 |
| 6.82.1.51 STATUS_ERR_BITS . . . . .                     | 359 |
| 6.82.1.52 STATUS_READY . . . . .                        | 359 |
| 6.82.1.53 USE_DMA . . . . .                             | 359 |
| 6.82.2 Enumeration Type Documentation . . . . .         | 359 |
| 6.82.2.1 SDCardState . . . . .                          | 359 |

---

|          |                                |     |
|----------|--------------------------------|-----|
| 6.82.3   | Function Documentation         | 360 |
| 6.82.3.1 | SDIO_IRQHandler()              | 360 |
| 6.82.3.2 | SDMMC2_IRQHandler()            | 360 |
| 6.82.4   | Variable Documentation         | 360 |
| 6.82.4.1 | dev                            | 360 |
| 6.83     | BSdio.h File Reference         | 360 |
| 6.84     | BSemaphore.cpp File Reference  | 361 |
| 6.85     | BSemaphore.h File Reference    | 361 |
| 6.86     | BSnmpServer.cpp File Reference | 361 |
| 6.86.1   | Macro Definition Documentation | 361 |
| 6.86.1.1 | BDEBUGL1                       | 362 |
| 6.86.2   | Variable Documentation         | 362 |
| 6.86.2.1 | osnmpServer                    | 362 |
| 6.87     | BSnmpServer.h File Reference   | 362 |
| 6.88     | BSpi.cpp File Reference        | 362 |
| 6.88.1   | Function Documentation         | 362 |
| 6.88.1.1 | log2()                         | 362 |
| 6.89     | BSpi.h File Reference          | 363 |
| 6.90     | BString.cpp File Reference     | 363 |
| 6.90.1   | Function Documentation         | 363 |
| 6.90.1.1 | blistToString()                | 363 |
| 6.90.1.2 | bstringListinList()            | 364 |
| 6.90.1.3 | bstringToList()                | 364 |
| 6.90.1.4 | bstrncpy()                     | 364 |
| 6.90.1.5 | bstrtrim()                     | 364 |
| 6.90.1.6 | floatToString()                | 364 |
| 6.90.1.7 | intToString()                  | 364 |
| 6.91     | BString.h File Reference       | 365 |
| 6.91.1   | Typedef Documentation          | 365 |
| 6.91.1.1 | BStringList                    | 365 |

|          |                                 |     |
|----------|---------------------------------|-----|
| 6.91.2   | Function Documentation          | 365 |
| 6.91.2.1 | blistToString()                 | 366 |
| 6.91.2.2 | bstringListinList()             | 366 |
| 6.91.2.3 | bstringToList()                 | 366 |
| 6.91.2.4 | bstrncpy()                      | 366 |
| 6.91.2.5 | bstrtrim()                      | 366 |
| 6.91.2.6 | floatToString()                 | 366 |
| 6.91.2.7 | from_hex()                      | 367 |
| 6.91.2.8 | intToString()                   | 367 |
| 6.91.2.9 | to_hex()                        | 367 |
| 6.92     | BSys.cpp File Reference         | 367 |
| 6.92.1   | Function Documentation          | 368 |
| 6.92.1.1 | _version()                      | 368 |
| 6.92.1.2 | delayMs()                       | 368 |
| 6.92.1.3 | delayUs()                       | 368 |
| 6.92.1.4 | delayUsClock()                  | 368 |
| 6.92.1.5 | delayUsLoop()                   | 369 |
| 6.92.1.6 | delayUsTask()                   | 369 |
| 6.92.1.7 | timeUs()                        | 369 |
| 6.92.1.8 | vApplicationGetIdleTaskMemory() | 369 |
| 6.92.1.9 | xPortSysTickHandler()           | 369 |
| 6.92.2   | Variable Documentation          | 369 |
| 6.92.2.1 | _ebss                           | 369 |
| 6.92.2.2 | _edata                          | 370 |
| 6.92.2.3 | _eheap                          | 370 |
| 6.92.2.4 | _estack                         | 370 |
| 6.92.2.5 | _etext                          | 370 |
| 6.92.2.6 | _sbss                           | 370 |
| 6.92.2.7 | _sdata                          | 370 |
| 6.92.2.8 | _stext                          | 370 |

---

|           |                                       |     |
|-----------|---------------------------------------|-----|
| 6.92.2.9  | <code>binterruptYield</code>          | 370 |
| 6.92.2.10 | <code>delayFunc</code>                | 371 |
| 6.93      | <code>BSys.h</code> File Reference    | 371 |
| 6.93.1    | Macro Definition Documentation        | 372 |
| 6.93.1.1  | <code>binterruptEnd</code>            | 372 |
| 6.93.1.2  | <code>binterruptStart</code>          | 372 |
| 6.93.2    | Typedef Documentation                 | 372 |
| 6.93.2.1  | <code>FuncDelay</code>                | 372 |
| 6.93.2.2  | <code>FuncRun</code>                  | 372 |
| 6.93.2.3  | <code>FuncTask</code>                 | 372 |
| 6.93.3    | Function Documentation                | 373 |
| 6.93.3.1  | <code>binterruptDisable()</code>      | 373 |
| 6.93.3.2  | <code>binterruptEnable()</code>       | 373 |
| 6.93.3.3  | <code>binterruptInside()</code>       | 373 |
| 6.93.3.4  | <code>bsysTaskYield()</code>          | 373 |
| 6.93.3.5  | <code>delayMs()</code>                | 373 |
| 6.93.3.6  | <code>delayUs()</code>                | 373 |
| 6.93.3.7  | <code>delayUsClock()</code>           | 374 |
| 6.93.3.8  | <code>delayUsLoop()</code>            | 374 |
| 6.93.3.9  | <code>delayUsTask()</code>            | 374 |
| 6.93.3.10 | <code>timeUs()</code>                 | 374 |
| 6.93.4    | Variable Documentation                | 374 |
| 6.93.4.1  | <code>binterruptYield</code>          | 374 |
| 6.93.4.2  | <code>BSysTimer</code>                | 374 |
| 6.93.4.3  | <code>delayFunc</code>                | 375 |
| 6.93.4.4  | <code>sys</code>                      | 375 |
| 6.94      | <code>BTask.cpp</code> File Reference | 375 |
| 6.94.1    | Function Documentation                | 375 |
| 6.94.1.1  | <code>__attribute__()</code>          | 375 |
| 6.94.1.2  | <code>makeFreeRtosPriority()</code>   | 375 |

---

---

|           |                                |     |
|-----------|--------------------------------|-----|
| 6.94.1.3  | osThreadCreate()               | 376 |
| 6.94.1.4  | taskDelayUs()                  | 376 |
| 6.94.1.5  | taskFunc()                     | 376 |
| 6.94.1.6  | vApplicationIdleHook()         | 376 |
| 6.95      | BTask.h File Reference         | 376 |
| 6.96      | BTim.cpp File Reference        | 376 |
| 6.96.1    | Variable Documentation         | 377 |
| 6.96.1.1  | TIM_CCER_CCE                   | 377 |
| 6.96.1.2  | TIM_CCMR1_OCCE                 | 377 |
| 6.97      | BTim.h File Reference          | 377 |
| 6.98      | BTime.cpp File Reference       | 377 |
| 6.98.1    | Function Documentation         | 378 |
| 6.98.1.1  | yearDays()                     | 378 |
| 6.98.1.2  | yearIsLeap()                   | 378 |
| 6.98.2    | Variable Documentation         | 378 |
| 6.98.2.1  | monDays                        | 378 |
| 6.99      | BTime.h File Reference         | 378 |
| 6.100     | BTimeUs.cpp File Reference     | 378 |
| 6.100.1   | Function Documentation         | 379 |
| 6.100.1.1 | yearDays()                     | 379 |
| 6.100.1.2 | yearIsLeap()                   | 379 |
| 6.100.2   | Variable Documentation         | 379 |
| 6.100.2.1 | monDays                        | 379 |
| 6.101     | BTimeUs.h File Reference       | 379 |
| 6.102     | BTypes.h File Reference        | 380 |
| 6.102.1   | Macro Definition Documentation | 381 |
| 6.102.1.1 | BBigEndian                     | 381 |
| 6.102.1.2 | M_PI                           | 381 |
| 6.102.2   | Typedef Documentation          | 381 |
| 6.102.2.1 | BChar                          | 382 |

---

|                    |     |
|--------------------|-----|
| 6.102.2.2 BDouble  | 382 |
| 6.102.2.3 BFloat   | 382 |
| 6.102.2.4 BFloat32 | 382 |
| 6.102.2.5 BFloat64 | 382 |
| 6.102.2.6 Bint     | 382 |
| 6.102.2.7 Bint16   | 382 |
| 6.102.2.8 Bint32   | 382 |
| 6.102.2.9 Bint64   | 383 |
| 6.102.2.10BInt8    | 383 |
| 6.102.2.11BBool    | 383 |
| 6.102.2.12BTimeout | 383 |
| 6.102.2.13BUInt    | 383 |
| 6.102.2.14BUInt16  | 383 |
| 6.102.2.15BUInt32  | 383 |
| 6.102.2.16BUInt64  | 383 |
| 6.102.2.17BUInt8   | 384 |
| 6.102.2.18BDouble  | 384 |
| 6.102.2.19BFloat   | 384 |
| 6.102.2.20BFloat32 | 384 |
| 6.102.2.21BFloat64 | 384 |
| 6.102.2.22BInt     | 384 |
| 6.102.2.23BInt16   | 384 |
| 6.102.2.24BInt32   | 384 |
| 6.102.2.25BInt64   | 385 |
| 6.102.2.26BInt8    | 385 |
| 6.102.2.27BUInt    | 385 |
| 6.102.2.28BUInt16  | 385 |
| 6.102.2.29BUInt32  | 385 |
| 6.102.2.30BUInt64  | 385 |
| 6.102.2.31BUInt8   | 385 |

---

|                                                  |     |
|--------------------------------------------------|-----|
| 6.102.3 Enumeration Type Documentation . . . . . | 385 |
| 6.102.3.1 BEventType . . . . .                   | 385 |
| 6.102.3.2 BEventWaitSet . . . . .                | 386 |
| 6.102.3.3 BType . . . . .                        | 386 |
| 6.102.3.4 BTypeComp . . . . .                    | 387 |
| 6.102.3.5 BUsbType . . . . .                     | 387 |
| 6.102.4 Function Documentation . . . . .         | 387 |
| 6.102.4.1 bswap_16() . . . . .                   | 387 |
| 6.102.4.2 bswap_32() . . . . .                   | 387 |
| 6.102.4.3 bswap_copy() . . . . .                 | 388 |
| 6.102.4.4 bswap_p64() . . . . .                  | 388 |
| 6.102.4.5 timeoutTicks() . . . . .               | 388 |
| 6.102.5 Variable Documentation . . . . .         | 388 |
| 6.102.5.1 BTimeoutForever . . . . .              | 388 |
| 6.103BUart.cpp File Reference . . . . .          | 388 |
| 6.103.1 Function Documentation . . . . .         | 389 |
| 6.103.1.1 UART4_IRQHandler() . . . . .           | 389 |
| 6.103.1.2 UART5_IRQHandler() . . . . .           | 389 |
| 6.103.1.3 USART1_IRQHandler() . . . . .          | 389 |
| 6.103.1.4 USART2_IRQHandler() . . . . .          | 389 |
| 6.103.1.5 USART3_IRQHandler() . . . . .          | 389 |
| 6.103.1.6 USART6_IRQHandler() . . . . .          | 389 |
| 6.104BUart.h File Reference . . . . .            | 389 |
| 6.105BUartBasic.cpp File Reference . . . . .     | 390 |
| 6.105.1 Function Documentation . . . . .         | 390 |
| 6.105.1.1 uartWrite() . . . . .                  | 390 |
| 6.105.2 Variable Documentation . . . . .         | 390 |
| 6.105.2.1 uart . . . . .                         | 390 |
| 6.106BUartBasic.h File Reference . . . . .       | 390 |
| 6.107BUsbSerial.cpp File Reference . . . . .     | 391 |

|                                              |     |
|----------------------------------------------|-----|
| 6.107.1 Macro Definition Documentation       | 393 |
| 6.107.1.1 dl1printf                          | 393 |
| 6.107.1.2 dl2printf                          | 393 |
| 6.107.1.3 dl3printf                          | 393 |
| 6.107.1.4 L1DEBUG                            | 393 |
| 6.107.1.5 L2DEBUG                            | 393 |
| 6.107.1.6 L3DEBUG                            | 393 |
| 6.107.1.7 USBD_CONFIGURATION_FS_STRING       | 394 |
| 6.107.1.8 USBD_CONFIGURATION_HS_STRING       | 394 |
| 6.107.1.9 USBD_INTERFACE_FS_STRING           | 394 |
| 6.107.1.10USBD_INTERFACE_HS_STRING           | 394 |
| 6.107.1.11USBD_LANGID_STRING                 | 394 |
| 6.107.1.12USBD_MANUFACTURER_STRING           | 394 |
| 6.107.1.13USBD_PID                           | 394 |
| 6.107.1.14USBD_PRODUCT_FS_STRING             | 394 |
| 6.107.1.15USBD_PRODUCT_HS_STRING             | 395 |
| 6.107.1.16USBD_SERIALNUMBER_FS_STRING        | 395 |
| 6.107.1.17USBD_SERIALNUMBER_HS_STRING        | 395 |
| 6.107.1.18USBD_VID                           | 395 |
| 6.107.2 Function Documentation               | 395 |
| 6.107.2.1 HAL_PCD_ConnectCallback()          | 395 |
| 6.107.2.2 HAL_PCD_DataInStageCallback()      | 395 |
| 6.107.2.3 HAL_PCD_DataOutStageCallback()     | 395 |
| 6.107.2.4 HAL_PCD_DisconnectCallback()       | 396 |
| 6.107.2.5 HAL_PCD_ISOINIncompleteCallback()  | 396 |
| 6.107.2.6 HAL_PCD_ISOOUTIncompleteCallback() | 396 |
| 6.107.2.7 HAL_PCD_ResetCallback()            | 396 |
| 6.107.2.8 HAL_PCD_ResumeCallback()           | 396 |
| 6.107.2.9 HAL_PCD_SetupStageCallback()       | 396 |
| 6.107.2.10HAL_PCD_SOFCallback()              | 396 |

|                                                |     |
|------------------------------------------------|-----|
| 6.107.2.11HAL_PCD_SuspendCallback()            | 397 |
| 6.107.2.12OTG_FS_IRQHandler()                  | 397 |
| 6.107.2.13OTG_HS_IRQHandler()                  | 397 |
| 6.107.2.14USBD_LL_ClearStallEP()               | 397 |
| 6.107.2.15USBD_LL_CloseEP()                    | 397 |
| 6.107.2.16USBD_LL_DeInit()                     | 397 |
| 6.107.2.17USBD_LL_Delay()                      | 397 |
| 6.107.2.18USBD_LL_FlushEP()                    | 398 |
| 6.107.2.19USBD_LL_GetRxDataSize()              | 398 |
| 6.107.2.20USBD_LL_Init()                       | 398 |
| 6.107.2.21USBD_LL_IsStallEP()                  | 398 |
| 6.107.2.22USBD_LL_OpenEP()                     | 398 |
| 6.107.2.23USBD_LL_PrepareReceive()             | 398 |
| 6.107.2.24USBD_LL_SetUSBAddress()              | 399 |
| 6.107.2.25USBD_LL_StallEP()                    | 399 |
| 6.107.2.26USBD_LL_Start()                      | 399 |
| 6.107.2.27USBD_LL_Stop()                       | 399 |
| 6.107.2.28USBD_LL_Transmit()                   | 399 |
| 6.107.2.29USBD_VCP_ConfigStrDescriptor()       | 399 |
| 6.107.2.30USBD_VCP_DeviceDescriptor()          | 400 |
| 6.107.2.31USBD_VCP_InterfaceStrDescriptor()    | 400 |
| 6.107.2.32USBD_VCP_LangIDStrDescriptor()       | 400 |
| 6.107.2.33USBD_VCP_ManufacturerStrDescriptor() | 400 |
| 6.107.2.34USBD_VCP_ProductStrDescriptor()      | 400 |
| 6.107.2.35USBD_VCP_SerialStrDescriptor()       | 400 |
| 6.107.2.36usbDelInit()                         | 401 |
| 6.107.2.37usbGetDeviceQualifierDescriptor()    | 401 |
| 6.107.2.38usbGetFSCfgDesc()                    | 401 |
| 6.107.2.39usbGetHSCfgDesc()                    | 401 |
| 6.107.2.40usbGetOtherSpeedCfgDesc()            | 401 |

|                                       |     |
|---------------------------------------|-----|
| 6.107.2.41usbInit()                   | 401 |
| 6.107.2.42usbRx()                     | 402 |
| 6.107.2.43usbRx0()                    | 402 |
| 6.107.2.44usbSetup()                  | 402 |
| 6.107.2.45usbTxSent()                 | 402 |
| 6.107.2.46usbWrite()                  | 402 |
| 6.107.3 Variable Documentation        | 402 |
| 6.107.3.1 __ALIGN_END                 | 403 |
| 6.107.3.2 busbConfig                  | 403 |
| 6.107.3.3 ConfigDescriptorSize        | 403 |
| 6.107.3.4 EndPointCmd                 | 403 |
| 6.107.3.5 EndPointRx                  | 404 |
| 6.107.3.6 EndPointTx                  | 404 |
| 6.107.3.7 PacketSizeCmd               | 404 |
| 6.107.3.8 PacketSizeFullSpeed         | 404 |
| 6.107.3.9 PacketSizeHighSpeed         | 404 |
| 6.107.3.10usbClass                    | 404 |
| 6.107.3.11usbSerial                   | 405 |
| 6.107.3.12VCP_Desc                    | 405 |
| 6.108BUbSerial.h File Reference       | 405 |
| 6.108.1 Typedef Documentation         | 405 |
| 6.108.1.1 BUbConfig                   | 405 |
| 6.109BUbSerialLock.cpp File Reference | 406 |
| 6.110BUbSerialLock.h File Reference   | 406 |
| 6.111BWatchdog.cpp File Reference     | 406 |
| 6.112BWatchdog.h File Reference       | 406 |
| 6.113overview.dox File Reference      | 406 |

# Chapter 1

## Main Page

### Author

Dr Terry Barnaby

### Version

0.12.18

### Date

2019-06-15

## 1.1 Introduction

This document covers the Armsys real-time embedded system API. The Armsys system provides a real-time system for embedded ARM based hardware platforms using a C++ object orientated API.

## 1.2 Overview

## 1.3 Examples

There are some examples of simple Armsys applications in the **examples** directory of the installation. Some simple client examples are listed below:

```
/******  
*   Main.cpp           ArmSys example  
*                       T.Barnaby,      Beam Ltd,      2012-11-12  
*   Copyright (c) 2013 All Right Reserved, Beam Ltd, http://www.beam.ltd.uk  
*****  
* Simple test of flashing an LED.  
*/  
#include <BConfig.h>  
  
// Board configuration  
BUInt32    armSysCrystal = CrystalFreq; // The system crystal frequency  
BUInt32    armSysPinLed0 = PinLed0;     // The first system LED
```

```
BUInt32      armSysPinLed1  = PinLed1;          // The second system LED

BSystem      sys;          // The system
BGpio        gpio;        // The GPIO (pins) interface

// Initialise the system
void init(){
    // Initialise the system
    sys.init();

    // Pin configuration
    gpio.initPin(PinLed0, BGpio::ModeOutput);
    gpio.initPin(PinLed1, BGpio::ModeOutput);
}

// The main program loop
void run(){
    while(1){
        // Flash a LED
        gpio.setPin(PinLed0, On);
        delayMs(1000);
        gpio.setPin(PinLed0, Off);
        delayMs(1000);
    }
}

int main(){
    init();
    sys.start();
    sys.run(run);

    return 0;
}
```

## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

|                          |     |
|--------------------------|-----|
| BAdc                     | 13  |
| BArray< T >              | 15  |
| BArray< BList< BIter > > | 15  |
| BBuffer                  | 21  |
| BBufferFixed< Size >     | 24  |
| BBufferStore             | 25  |
| BClocks                  | 36  |
| BComms                   | 37  |
| BCan                     | 31  |
| BUart                    | 233 |
| BUartBasic               | 239 |
| BUsbSerial               | 243 |
| BUsbSerialLock           | 253 |
| BCondInt                 | 43  |
| BCritical                | 47  |
| BCriticalLock            | 48  |
| BDataChunk               | 49  |
| BDevices                 | 50  |
| BDictItem< Type >        | 57  |
| BDma                     | 58  |
| BError                   | 65  |
| BEvent                   | 67  |
| BFifo< Type >            | 68  |
| BFifo< BoapMcPacket >    | 68  |
| BFifo< BUInt16 >         | 68  |
| BFifo< char >            | 68  |
| BCanPacketFifo           | 35  |
| BFifoChar                | 76  |
| BFirmwareInfo            | 77  |
| BFlash                   | 79  |
| BGpio                    | 82  |
| BHeap                    | 87  |
| BHtml                    | 88  |
| BI2c                     | 100 |

|                               |     |
|-------------------------------|-----|
| BInterrupt                    | 105 |
| BIter                         | 107 |
| BList< T >                    | 109 |
| BList< BDictItem< BString > > | 109 |
| BDict< BString >              | 51  |
| BList< BDictItem< Type > >    | 109 |
| BDict< Type >                 | 51  |
| BList< BHtml >                | 109 |
| BList< BIter >                | 109 |
| BList< BNameValue< T > >      | 109 |
| BNameValueList< T >           | 125 |
| BList< BString >              | 109 |
| BMutex                        | 120 |
| BMutexLock                    | 122 |
| BNameValue< T >               | 123 |
| BNetwork                      | 126 |
| BNetworkStats                 | 137 |
| BNode                         | 139 |
| BList< T >::Node              | 259 |
| BNtpClient                    | 140 |
| BoapMc1Comms                  | 141 |
| BoapMc1Error                  | 148 |
| BoapMc1Packet                 | 149 |
| BoapMc1PacketHead             | 149 |
| BoapMcClientObject            | 151 |
| BoapMcComms                   | 154 |
| BoapMcPacket                  | 162 |
| BoapMcPacketHead              | 162 |
| BoapMcServiceObject           | 163 |
| BoapMcSignalObject            | 165 |
| BObj                          | 166 |
| BObjMember                    | 167 |
| BPwm                          | 168 |
| BQueue< T >                   | 170 |
| BQueue< BoapMcPacket >        | 170 |
| BQueue< BUInt >               | 170 |
| BRandom                       | 173 |
| BRefData                      | 174 |
| BRefString                    | 175 |
| BRtc                          | 178 |
| BSdio                         | 181 |
| BSemaphore                    | 187 |
| BSemaphoreBool                | 189 |
| BSemaphoreCount               | 192 |
| BSnmpServer                   | 194 |
| BSpi                          | 195 |
| BString                       | 199 |
| BSys                          | 213 |
| BTask                         | 217 |
| BHttpServer                   | 91  |
| BNetworkDhcpTask              | 135 |
| BNetworkProcessTask           | 136 |
| BTim                          | 220 |
| BTime                         | 223 |
| BTimeUs                       | 228 |
| BUsbConfig                    | 242 |
| BWatchdog                     | 255 |

---

|                         |     |
|-------------------------|-----|
| HeapTestState . . . . . | 256 |
| IpAddress . . . . .     | 257 |



# Chapter 3

## Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

|                                                      |     |
|------------------------------------------------------|-----|
| <a href="#">BAdc</a>                                 | 13  |
| <a href="#">BArray&lt; T &gt;</a>                    | 15  |
| <a href="#">BBuffer</a>                              | 21  |
| <a href="#">BBufferFixed&lt; Size &gt;</a>           | 24  |
| <a href="#">BBufferStore</a>                         | 25  |
| <a href="#">BCan</a>                                 | 31  |
| <a href="#">BCanPacketFifo</a>                       | 35  |
| <a href="#">BClocks</a>                              | 36  |
| <a href="#">BComms</a>                               | 37  |
| <a href="#">BCondInt</a>                             |     |
| <a href="#">BCondInt class</a>                       | 43  |
| <a href="#">BCritical</a>                            |     |
| Critical region class                                | 47  |
| <a href="#">BCriticalLock</a>                        |     |
| Critical region class with unlock on object deletion | 48  |
| <a href="#">BDataChunk</a>                           | 49  |
| <a href="#">BDevices</a>                             | 50  |
| <a href="#">BDict&lt; Type &gt;</a>                  | 51  |
| <a href="#">BDictItem&lt; Type &gt;</a>              |     |
| Template based Dictionary class                      | 57  |
| <a href="#">BDma</a>                                 | 58  |
| <a href="#">BError</a>                               | 65  |
| <a href="#">BEvent</a>                               | 67  |
| <a href="#">BFifo&lt; Type &gt;</a>                  | 68  |
| <a href="#">BFifoChar</a>                            | 76  |
| <a href="#">BFirmwareInfo</a>                        | 77  |
| <a href="#">BFlash</a>                               | 79  |
| <a href="#">BGpio</a>                                | 82  |
| <a href="#">BHeap</a>                                | 87  |
| <a href="#">BHtml</a>                                | 88  |
| <a href="#">BHttpServer</a>                          | 91  |
| <a href="#">BI2c</a>                                 | 100 |
| <a href="#">BInterrupt</a>                           | 105 |
| <a href="#">BIter</a>                                |     |
| Iterator for <a href="#">BList</a>                   | 107 |

|                           |     |
|---------------------------|-----|
| BList< T >                | 109 |
| Template based list class | 109 |
| BMutex                    | 120 |
| Mutex class               | 120 |
| BMutexLock                | 122 |
| BNameValue< T >           | 123 |
| BNameValueList< T >       | 125 |
| BNetwork                  | 126 |
| BNetworkDhcpTask          | 135 |
| BNetworkProcessTask       | 136 |
| BNetworkStats             | 137 |
| BNode                     | 139 |
| BNtpClient                | 140 |
| BoapMc1Comms              | 141 |
| BoapMc1Error              | 148 |
| BoapMc1Packet             | 149 |
| BoapMc1PacketHead         | 149 |
| BoapMcClientObject        | 151 |
| BoapMcComms               | 154 |
| BoapMcPacket              | 162 |
| BoapMcPacketHead          | 162 |
| BoapMcServiceObject       | 163 |
| BoapMcSignalObject        | 165 |
| BObj                      | 166 |
| BObjMember                | 167 |
| BPwm                      | 168 |
| BQueue< T >               | 170 |
| Queue class               | 170 |
| BRandom                   | 173 |
| BRefData                  | 174 |
| BRefString                | 175 |
| BRtc                      | 178 |
| BSdio                     | 181 |
| BSemaphore                | 187 |
| Semaphore class           | 187 |
| BSemaphoreBool            | 189 |
| BSemaphoreCount           | 192 |
| BSnmpServer               | 194 |
| BSpi                      | 195 |
| BString                   | 199 |
| BSys                      | 213 |
| BTask                     | 217 |
| BTim                      | 220 |
| BTime                     | 223 |
| BTimeUs                   | 228 |
| BUart                     | 233 |
| BUartBasic                | 239 |
| BUsbConfig                | 242 |
| BUsbSerial                | 243 |
| BUsbSerialLock            | 253 |
| BWatchdog                 | 255 |
| HeapTestState             | 256 |
| IpAddress                 | 257 |
| BList< T >::Node          | 259 |

# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

|                               |     |
|-------------------------------|-----|
| <a href="#">BAdc.cpp</a>      | 261 |
| <a href="#">BAdc.h</a>        | 267 |
| <a href="#">BArray.h</a>      | 267 |
| <a href="#">BArray_imp.h</a>  | 268 |
| <a href="#">BBuffer.cpp</a>   | 268 |
| <a href="#">BBuffer.h</a>     | 269 |
| <a href="#">BCan.cpp</a>      | 269 |
| <a href="#">BCan.h</a>        | 270 |
| <a href="#">BComms.cpp</a>    | 271 |
| <a href="#">BComms.h</a>      | 271 |
| <a href="#">BComplex.h</a>    | 271 |
| <a href="#">BCondInt.cpp</a>  | 272 |
| <a href="#">BCondInt.h</a>    | 272 |
| <a href="#">BConfig.h</a>     | 272 |
| <a href="#">BConvert.cpp</a>  | 272 |
| <a href="#">BConvert.h</a>    | 273 |
| <a href="#">BCrc16.cpp</a>    | 274 |
| <a href="#">BCrc16.h</a>      | 276 |
| <a href="#">BCrc32.cpp</a>    | 276 |
| <a href="#">BCrc32.h</a>      | 277 |
| <a href="#">BCritical.h</a>   | 277 |
| <a href="#">BCritical.inc</a> | 278 |
| <a href="#">BDebug.cpp</a>    | 278 |
| <a href="#">BDebug.h</a>      | 279 |
| <a href="#">BDevices.cpp</a>  | 281 |
| <a href="#">BDevices.h</a>    | 282 |
| <a href="#">BDict.cpp</a>     | 285 |
| <a href="#">BDict.h</a>       | 286 |
| <a href="#">BDma.cpp</a>      | 287 |
| <a href="#">BDma.h</a>        | 291 |
| <a href="#">BEndian.cpp</a>   | 291 |
| <a href="#">BEndian.h</a>     | 292 |
| <a href="#">BError.cpp</a>    | 293 |
| <a href="#">BError.h</a>      | 293 |
| <a href="#">BEvent.cpp</a>    | 294 |

|                      |     |
|----------------------|-----|
| BEvent.h             | 294 |
| BFifo.h              | 295 |
| BFifo.inc            | 295 |
| BFirmware.cpp        | 295 |
| BFirmware.h          | 296 |
| BFlash.cpp           | 297 |
| BFlash.h             | 299 |
| BGpio.cpp            | 299 |
| BGpio.h              | 299 |
| BHeap.cpp            | 305 |
| BHeap.h              | 305 |
| BHeapTest.cpp        | 306 |
| BHeapTest.h          | 308 |
| BHtml.cpp            | 309 |
| BHtml.h              | 310 |
| BHttpServer.cpp      | 311 |
| BHttpServer.h        | 313 |
| BI2c.cpp             | 313 |
| BI2c.h               | 316 |
| BInterrupt.cpp       | 316 |
| BInterrupt.h         | 317 |
| BList.h              | 317 |
| BList_func.h         | 318 |
| BMutex.cpp           | 318 |
| BMutex.h             | 318 |
| BNameValue.h         | 318 |
| BNetwork.cpp         | 319 |
| BNetwork.h           | 323 |
| BNtpClient.cpp       | 324 |
| BNtpClient.h         | 325 |
| BoapMc.cpp           | 325 |
| BoapMc.h             | 326 |
| BoapMc1.cpp          | 328 |
| BoapMc1.h            | 329 |
| BObj.cpp             | 333 |
| BObj.h               | 333 |
| BObjStringFormat.cpp | 333 |
| BObjStringFormat.h   | 340 |
| BPwm.cpp             | 348 |
| BPwm.h               | 348 |
| BQueue.cpp           | 348 |
| BQueue.h             | 348 |
| BRandom.cpp          | 349 |
| BRandom.h            | 349 |
| BRtc.cpp             | 349 |
| BRtc.h               | 351 |
| BSdio.cpp            | 351 |
| BSdio.h              | 360 |
| BSemaphore.cpp       | 361 |
| BSemaphore.h         | 361 |
| BSnmpServer.cpp      | 361 |
| BSnmpServer.h        | 362 |
| BSpi.cpp             | 362 |
| BSpi.h               | 363 |
| BString.cpp          | 363 |
| BString.h            | 365 |
| BSys.cpp             | 367 |
| BSys.h               | 371 |

---

|                    |     |
|--------------------|-----|
| BTask.cpp          | 375 |
| BTask.h            | 376 |
| BTim.cpp           | 376 |
| BTim.h             | 377 |
| BTime.cpp          | 377 |
| BTime.h            | 378 |
| BTimeUs.cpp        | 378 |
| BTimeUs.h          | 379 |
| BTypes.h           | 380 |
| BUart.cpp          | 388 |
| BUart.h            | 389 |
| BUartBasic.cpp     | 390 |
| BUartBasic.h       | 390 |
| BUsbSerial.cpp     | 391 |
| BUsbSerial.h       | 405 |
| BUsbSerialLock.cpp | 406 |
| BUsbSerialLock.h   | 406 |
| BWatchdog.cpp      | 406 |
| BWatchdog.h        | 406 |



# Chapter 5

## Class Documentation

### 5.1 BAdc Class Reference

```
#include <BAdc.h>
```

#### Public Member Functions

- [BAdc](#) ()
- [BError init](#) (UInt8 dev)  
*Initialises the ADC. Devices 1 to 3.*
- [BError initChannel](#) (UInt8 channel)  
*Initialise a channel and pin for use.*
- [UInt16 getValue](#) (UInt8 channel)  
*Sample the data, returns a value 0 - 4095.*
- void [enableTempAndRefInput](#) (Bool on)  
*Enable temperatue sensor (chan 16) and voltage ref sensor (chan 17)*
- void [enableBatteryInput](#) (Bool on)  
*Enable RTC battery input (chan 18)*

#### Protected Attributes

- int [odev](#)
- ADC\_TypeDef \* [oadc](#)

#### 5.1.1 Constructor & Destructor Documentation

##### 5.1.1.1 BAdc()

```
BAdc::BAdc ( )
```

## 5.1.2 Member Function Documentation

### 5.1.2.1 enableBatteryInput()

```
void BAdc::enableBatteryInput (
    Bool on )
```

Enable RTC battery input (chan 18)

### 5.1.2.2 enableTempAndRefInput()

```
void BAdc::enableTempAndRefInput (
    Bool on )
```

Enable temperature sensor (chan 16) and voltage ref sensor (chan 17)

### 5.1.2.3 getValue()

```
UInt16 BAdc::getValue (
    UInt8 channel )
```

Sample the data, returns a value 0 - 4095.

### 5.1.2.4 init()

```
BError BAdc::init (
    UInt8 dev )
```

Initialises the ADC. Devices 1 to 3.

### 5.1.2.5 initChannel()

```
BError BAdc::initChannel (
    UInt8 channel )
```

Initialise a channel and pin for use.

### 5.1.3 Member Data Documentation

#### 5.1.3.1 oadc

ADC\_TypeDef\* BAdc::oadc [protected]

#### 5.1.3.2 odev

int BAdc::odev [protected]

The documentation for this class was generated from the following files:

- [BAdc.h](#)
- [BAdc.cpp](#)

## 5.2 BArray< T > Class Template Reference

```
#include <BArray.h>
```

### Public Member Functions

- [BArray](#) ()
- [BArray](#) (const [BArray](#)< T > &a)
- [BArray](#) (BUInt size, T \*data=0)
- [~BArray](#) ()
- [UInt](#) size ()
  - returns the number of elements in the array*
- [UInt](#) number ()
  - returns the number of elements in the array*
- T & [get](#) (BUInt i)
  - Access one of the arrays elements.*
- const T & [get](#) (BUInt i) const
  - Access one of the arrays elements.*
- void [clear](#) ()
  - Clear the array.*
- void [resize](#) (BUInt size)
  - Sets the arrays size.*
- void [append](#) (const T &value)
  - Appemd an element to the end of the array.*
- void [insert](#) (BUInt i, const T &value)
  - Insert an element before the position given.*
- void [del](#) (BUInt i, BUInt num=1)
  - Delete the given number of elements from the arrays starting at position given.*

- `BUInt dataSize ()`  
*Size of raw data in bytes.*
- `T * data ()`  
*Access to raw data.*
- `BArray< T > & operator= (const BArray< T > &a)`  
*Array copy.*
- `T & operator[] (int i)`  
*Access one of the arrays elements.*
- `const T & operator[] (int i) const`  
*Access one of the arrays elements.*
- `operator T* ()`  
*Access raw data.*

### Private Member Functions

- void `allocate (BUInt size)`
- void `relocate (BUInt size)`  
*Reallocate the arrays memory to a new size.*

### Private Attributes

- `BUInt osize`  
*The number of elements in the array.*
- `BUInt odataSize`  
*The actual size in elements of the array.*
- `T * odata`  
*The array data itself.*

### Static Private Attributes

- static `BUInt chunkSize = 8`  
*The data allocation chunk size.*

## 5.2.1 Constructor & Destructor Documentation

### 5.2.1.1 BArray() [1/3]

```
template<class T >
BArray< T >::BArray ( )
```

### 5.2.1.2 BArray() [2/3]

```
template<class T>
BArray< T >::BArray (
    const BArray< T > & a )
```

### 5.2.1.3 BArray() [3/3]

```
template<class T>
BArray< T >::BArray (
    BUInt size,
    T * data = 0 )
```

### 5.2.1.4 ~BArray()

```
template<class T >
BArray< T >::~~BArray ( )
```

## 5.2 Member Function Documentation

### 5.2.2.1 allocate()

```
template<class T >
void BArray< T >::allocate (
    BUInt size ) [private]
```

### 5.2.2.2 append()

```
template<class T>
void BArray< T >::append (
    const T & value )
```

Append an element to the end of the array.

### 5.2.2.3 clear()

```
template<class T >
void BArray< T >::clear ( )
```

Clear the array.

### 5.2.2.4 data()

```
template<class T >
T * BArray< T >::data ( )
```

Access to raw data.

### 5.2.2.5 dataSize()

```
template<class T >
BUInt BArray< T >::dataSize ( )
```

Size of raw data in bytes.

### 5.2.2.6 del()

```
template<class T >
void BArray< T >::del (
    BUInt i,
    BUInt num = 1 )
```

Delete the given number of elements from the arrays starting at position given.

### 5.2.2.7 get() [1/2]

```
template<class T >
T & BArray< T >::get (
    BUInt i )
```

Access one of the arrays elements.

### 5.2.2.8 get() [2/2]

```
template<class T >
const T & BArray< T >::get (
    BUInt i ) const
```

Access one of the arrays elements.

### 5.2.2.9 insert()

```
template<class T>
void BArray< T >::insert (
    BUInt i,
    const T & value )
```

Insert an element before the position given.

### 5.2.2.10 number()

```
template<class T >
UInt BArray< T >::number ( )
```

returns the number of elements in the array

### 5.2.2.11 operator T\*()

```
template<class T >
BArray< T >::operator T* ( )
```

Access raw data.

### 5.2.2.12 operator=()

```
template<class T>
BArray< T > & BArray< T >::operator= (
    const BArray< T > & a )
```

Array copy.

### 5.2.2.13 operator[]() [1/2]

```
template<class T >
T & BArray< T >::operator[] (
    int i )
```

Access one of the arrays elements.

### 5.2.2.14 operator[]() [2/2]

```
template<class T >
const T & BArray< T >::operator[] (
    int i ) const
```

Access one of the arrays elements.

### 5.2.2.15 relocate()

```
template<class T >
void BArray< T >::relocate (
    BUInt size ) [private]
```

Reallocate the arrays memory to a new size.

### 5.2.2.16 resize()

```
template<class T >
void BArray< T >::resize (
    BUInt size )
```

Sets the arrays size.

### 5.2.2.17 size()

```
template<class T >
UInt BArray< T >::size ( )
```

returns the number of elements in the array

## 5.2.3 Member Data Documentation

### 5.2.3.1 chunkSize

```
template<class T>
BUInt BArray< T >::chunkSize = 8 [static], [private]
```

The data allocation chunk size.

### 5.2.3.2 odata

```
template<class T>
T* BArray< T >::odata [private]
```

The array data itself.

### 5.2.3.3 odataSize

```
template<class T>
BUInt BArray< T >::odataSize [private]
```

The actual size in elements of the array.

### 5.2.3.4 osize

```
template<class T>
BUInt BArray< T >::osize [private]
```

The number of elements in the array.

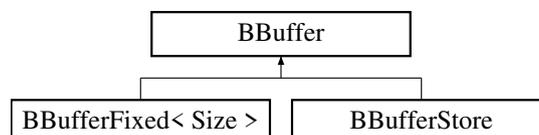
The documentation for this class was generated from the following files:

- [BArray.h](#)
- [BArray\\_imp.h](#)

## 5.3 BBuffer Class Reference

```
#include <BBuffer.h>
```

Inheritance diagram for BBuffer:



## Public Member Functions

- [BBuffer](#) (char \*data=0, BUInt len=0)
- [~BBuffer](#) ()
- [BError setSize](#) (BUInt size)  
*Sets the bufer size.*
- char \* [data](#) ()
- [UInt len](#) ()
- [UInt size](#) ()
- char & [operator\[\]](#) (int pos)

## Protected Attributes

- char \* [odata](#)
- [BUInt odataSize](#)
- [BUInt osize](#)

## 5.3.1 Constructor & Destructor Documentation

### 5.3.1.1 BBuffer()

```
BBuffer::BBuffer (
    char * data = 0,
    BUInt len = 0 )
```

### 5.3.1.2 ~BBuffer()

```
BBuffer::~~BBuffer ( )
```

## 5.3.2 Member Function Documentation

### 5.3.2.1 data()

```
char* BBuffer::data ( ) [inline]
```

### 5.3.2.2 len()

```
UInt BBuffer::len ( ) [inline]
```

### 5.3.2.3 operator[]()

```
char& BBuffer::operator[] (
    int pos ) [inline]
```

### 5.3.2.4 setSize()

```
BError BBuffer::setSize (
    BUInt size )
```

Sets the bufer size.

### 5.3.2.5 size()

```
UInt BBuffer::size ( ) [inline]
```

## 5.3.3 Member Data Documentation

### 5.3.3.1 odata

```
char* BBuffer::odata [protected]
```

### 5.3.3.2 odataSize

```
BUInt BBuffer::odataSize [protected]
```

### 5.3.3.3 osize

```
BUInt BBuffer::osize [protected]
```

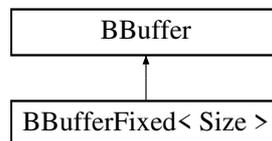
The documentation for this class was generated from the following files:

- [BBuffer.h](#)
- [BBuffer.cpp](#)

## 5.4 BBufferFixed< Size > Class Template Reference

```
#include <BBuffer.h>
```

Inheritance diagram for BBufferFixed< Size >:



### Public Member Functions

- [BBufferFixed\(\)](#)

### Private Attributes

- char [odataFixed](#) [Size]

### Additional Inherited Members

#### 5.4.1 Constructor & Destructor Documentation

##### 5.4.1.1 BBufferFixed()

```
template<int Size>
BBufferFixed< Size >::BBufferFixed ( ) [inline]
```

#### 5.4.2 Member Data Documentation

## 5.4.2.1 odataFixed

```
template<int Size>
char BBufferFixed< Size >::odataFixed[Size] [private]
```

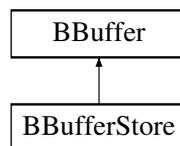
The documentation for this class was generated from the following file:

- [BBuffer.h](#)

## 5.5 BBufferStore Class Reference

```
#include <BBuffer.h>
```

Inheritance diagram for BBufferStore:



## Public Member Functions

- [BBufferStore](#) ([BUInt](#) size=0, [int](#) swapBytes=[BBigEndian](#))
- [~BBufferStore](#) ()
- [BUInt32](#) [getPos](#) ()
- void [setPos](#) ([BUInt32](#) pos)
- [BString](#) [getHexString](#) ()
- void [setHexString](#) ([BString](#) s)
- int [push](#) ([BInt8](#) v)
- int [push](#) ([BUInt8](#) v)
- int [push](#) ([BInt16](#) v)
- int [push](#) ([BUInt16](#) v)
- int [push](#) ([BInt32](#) v)
- int [push](#) ([BUInt32](#) v)
- int [push](#) ([BInt64](#) v)
- int [push](#) ([BUInt64](#) v)
- int [push](#) ([BFloat32](#) v)
- int [push](#) ([BFloat64](#) v)
- int [push](#) (const [BString](#) &v)
- int [push](#) (const [BError](#) &v)
- int [push](#) (const [BComplex](#) &v)
- int [push](#) ([BUInt32](#) nBytes, const void \*data, const char \*swapType="1")
- int [pop](#) ([BInt8](#) &v)
- int [pop](#) ([BUInt8](#) &v)
- int [pop](#) ([BInt16](#) &v)
- int [pop](#) ([BUInt16](#) &v)
- int [pop](#) ([BInt32](#) &v)
- int [pop](#) ([BUInt32](#) &v)
- int [pop](#) ([BInt64](#) &v)
- int [pop](#) ([BUInt64](#) &v)
- int [pop](#) ([BFloat32](#) &v)
- int [pop](#) ([BFloat64](#) &v)
- int [pop](#) ([BString](#) &v)
- int [pop](#) ([BError](#) &v)
- int [pop](#) ([BComplex](#) &v)
- int [pop](#) ([BUInt32](#) nBytes, void \*data, const char \*swapType="1")

## Protected Attributes

- BUInt32 opos
- int oswapBytes

## 5.5.1 Constructor & Destructor Documentation

### 5.5.1.1 BBufferStore()

```
BBufferStore::BBufferStore (
    BUInt size = 0,
    int swapBytes = BBigEndian )
```

### 5.5.1.2 ~BBufferStore()

```
BBufferStore::~~BBufferStore ( )
```

## 5.5.2 Member Function Documentation

### 5.5.2.1 getHexString()

```
BString BBufferStore::getHexString ( )
```

### 5.5.2.2 getPos()

```
BUInt32 BBufferStore::getPos ( )
```

### 5.5.2.3 pop() [1/14]

```
int BBufferStore::pop (
    BInt8 & v )
```

**5.5.2.4 pop()** [2/14]

```
int BBufferStore::pop (
    BUInt8 & v )
```

**5.5.2.5 pop()** [3/14]

```
int BBufferStore::pop (
    BInt16 & v )
```

**5.5.2.6 pop()** [4/14]

```
int BBufferStore::pop (
    BUInt16 & v )
```

**5.5.2.7 pop()** [5/14]

```
int BBufferStore::pop (
    BInt32 & v )
```

**5.5.2.8 pop()** [6/14]

```
int BBufferStore::pop (
    BUInt32 & v )
```

**5.5.2.9 pop()** [7/14]

```
int BBufferStore::pop (
    BInt64 & v )
```

**5.5.2.10 pop()** [8/14]

```
int BBufferStore::pop (
    BUInt64 & v )
```

**5.5.2.11 pop()** [9/14]

```
int BBufferStore::pop (
    BFloat32 & v )
```

**5.5.2.12 pop()** [10/14]

```
int BBufferStore::pop (
    BFloat64 & v )
```

**5.5.2.13 pop()** [11/14]

```
int BBufferStore::pop (
    BString & v )
```

**5.5.2.14 pop()** [12/14]

```
int BBufferStore::pop (
    BError & v )
```

**5.5.2.15 pop()** [13/14]

```
int BBufferStore::pop (
    BComplex & v )
```

**5.5.2.16 pop()** [14/14]

```
int BBufferStore::pop (
    BUInt32 nBytes,
    void * data,
    const char * swapType = "1" )
```

**5.5.2.17 push()** [1/14]

```
int BBufferStore::push (
    BInt8 v )
```

**5.5.2.18 push()** [2/14]

```
int BBufferStore::push (  
    BUInt8 v )
```

**5.5.2.19 push()** [3/14]

```
int BBufferStore::push (  
    BInt16 v )
```

**5.5.2.20 push()** [4/14]

```
int BBufferStore::push (  
    BUInt16 v )
```

**5.5.2.21 push()** [5/14]

```
int BBufferStore::push (  
    BInt32 v )
```

**5.5.2.22 push()** [6/14]

```
int BBufferStore::push (  
    BUInt32 v )
```

**5.5.2.23 push()** [7/14]

```
int BBufferStore::push (  
    BInt64 v )
```

**5.5.2.24 push()** [8/14]

```
int BBufferStore::push (  
    BUInt64 v )
```

**5.5.2.25 push()** [9/14]

```
int BBufferStore::push (
    BFloat32 v )
```

**5.5.2.26 push()** [10/14]

```
int BBufferStore::push (
    BFloat64 v )
```

**5.5.2.27 push()** [11/14]

```
int BBufferStore::push (
    const BString & v )
```

**5.5.2.28 push()** [12/14]

```
int BBufferStore::push (
    const BError & v )
```

**5.5.2.29 push()** [13/14]

```
int BBufferStore::push (
    const BComplex & v )
```

**5.5.2.30 push()** [14/14]

```
int BBufferStore::push (
    BUInt32 nBytes,
    const void * data,
    const char * swapType = "1" )
```

**5.5.2.31 setHexString()**

```
void BBufferStore::setHexString (
    BString s )
```

## 5.5.2.32 setPos()

```
void BBufferStore::setPos (
    BUInt32 pos )
```

## 5.5.3 Member Data Documentation

## 5.5.3.1 opos

```
BUInt32 BBufferStore::opos [protected]
```

## 5.5.3.2 oswapBytes

```
int BBufferStore::oswapBytes [protected]
```

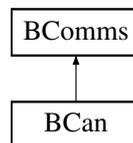
The documentation for this class was generated from the following files:

- [BBuffer.h](#)
- [BBuffer.cpp](#)

## 5.6 BCan Class Reference

```
#include <BCan.h>
```

Inheritance diagram for BCan:



## Public Member Functions

- [BCan](#) (BUInt fifoSize=64)
- [BError](#) [init](#) (UInt8 dev, BUInt speed=125000, bool loopback=0)  
*Initialise device: 1 to 2.*
- [Bool](#) [packetMode](#) ()  
*Device is in packet mode.*
- [BUInt](#) [writeAvailable](#) ()  
*Check if can write, returns 0 or 1.*
- [BError](#) [write](#) (const void \*data, UInt32 nBytes, UInt32 &nTrans)  
*Write nBytes, will wait for hardware.*
- [BUInt](#) [readAvailable](#) ()  
*Check if data is available, returns 0 or 1.*
- [BError](#) [read](#) (void \*data, UInt32 nBytes, UInt32 &nTrans)  
*Read nBytes, will wait for hardware.*
- [BError](#) [sendCanPacket](#) (Bool extId, BUInt id, const BUInt8 \*data, BUInt nBytes)
- [BError](#) [recvCanPacket](#) (Bool &extId, BUInt &id, BUInt8 \*data, BUInt &nBytes)
- void [interrupt](#) ()
- void [debugPrint](#) ()

## Protected Attributes

- [BCanPacketFifo orxFifoCmd](#)
- [BCanPacketFifo orxFifoReply](#)
- int `odev`
- CAN\_TypeDef \* `ocan`

## Additional Inherited Members

### 5.6.1 Constructor & Destructor Documentation

#### 5.6.1.1 BCan()

```
BCan::BCan (
    BUInt fifoSize = 64 )
```

### 5.6.2 Member Function Documentation

#### 5.6.2.1 debugPrint()

```
void BCan::debugPrint ( )
```

#### 5.6.2.2 init()

```
BError BCan::init (
    UInt8 dev,
    BUInt speed = 125000,
    bool loopback = 0 )
```

Initialise device: 1 to 2.

#### 5.6.2.3 interrupt()

```
void BCan::interrupt ( )
```

#### 5.6.2.4 packetMode()

```
Bool BCan::packetMode ( ) [virtual]
```

Device is in packet mode.

Reimplemented from [BComms](#).

#### 5.6.2.5 read()

```
BError BCan::read (
    void * data,
    UInt32 nBytes,
    UInt32 & nTrans ) [virtual]
```

Read nBytes, will wait for hardware.

Implements [BComms](#).

#### 5.6.2.6 readAvailable()

```
BUInt BCan::readAvailable ( ) [virtual]
```

Check if data is available, returns 0 or 1.

Reimplemented from [BComms](#).

#### 5.6.2.7 recvCanPacket()

```
BError BCan::recvCanPacket (
    Bool & extId,
    BUInt & id,
    BUInt8 * data,
    BUInt & nBytes )
```

#### 5.6.2.8 sendCanPacket()

```
BError BCan::sendCanPacket (
    Bool extId,
    BUInt id,
    const BUInt8 * data,
    BUInt nBytes )
```

### 5.6.2.9 write()

```
BError BCan::write (
    const void * data,
    UInt32 nBytes,
    UInt32 & nTrans ) [virtual]
```

Write nBytes, will wait for hardware.

Implements [BComms](#).

### 5.6.2.10 writeAvailable()

```
BUInt BCan::writeAvailable ( ) [virtual]
```

Check if can write, returns 0 or 1.

Reimplemented from [BComms](#).

## 5.6.3 Member Data Documentation

### 5.6.3.1 ocan

```
CAN_TypeDef* BCan::ocan [protected]
```

### 5.6.3.2 odev

```
int BCan::odev [protected]
```

### 5.6.3.3 orxFifoCmd

```
BCanPacketFifo BCan::orxFifoCmd [protected]
```

### 5.6.3.4 orxFifoReply

`BCanPacketFifo` `BCan::orxFifoReply` [protected]

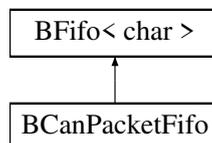
The documentation for this class was generated from the following files:

- [BCan.h](#)
- [BCan.cpp](#)

## 5.7 BCanPacketFifo Class Reference

```
#include <BCan.h>
```

Inheritance diagram for BCanPacketFifo:



### Public Member Functions

- [BCanPacketFifo](#) (`BUInt` fifoSize)

### Public Attributes

- `Bool` inPacket
- `BUInt16` num
- `BFifo< BUInt16 >` numFifo

### Additional Inherited Members

#### 5.7.1 Constructor & Destructor Documentation

##### 5.7.1.1 BCanPacketFifo()

```
BCanPacketFifo::BCanPacketFifo (  
    BUInt fifoSize )
```

#### 5.7.2 Member Data Documentation

### 5.7.2.1 inPacket

`Bool BCanPacketFifo::inPacket`

### 5.7.2.2 num

`BUInt16 BCanPacketFifo::num`

### 5.7.2.3 numFifo

`BFifo<BUInt16> BCanPacketFifo::numFifo`

The documentation for this class was generated from the following files:

- [BCan.h](#)
- [BCan.cpp](#)

## 5.8 BClocks Class Reference

```
#include <BDevices.h>
```

### Public Attributes

- [BUInt sysClock](#)
- [BUInt ahbClock](#)
- [BUInt apb1Clock](#)
- [BUInt apb2Clock](#)

### 5.8.1 Member Data Documentation

#### 5.8.1.1 ahbClock

`BUInt BClocks::ahbClock`

### 5.8.1.2 apb1Clock

```
BUInt BComms::apb1Clock
```

### 5.8.1.3 apb2Clock

```
BUInt BComms::apb2Clock
```

### 5.8.1.4 sysClock

```
BUInt BComms::sysClock
```

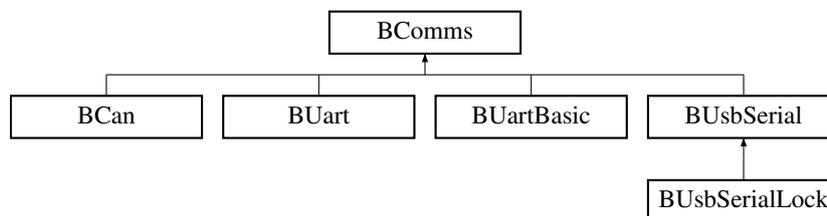
The documentation for this class was generated from the following file:

- [BDevices.h](#)

## 5.9 BComms Class Reference

```
#include <BComms.h>
```

Inheritance diagram for BComms:



### Public Types

- enum [Flush](#) { [FlushRead](#), [FlushWrite](#), [FlushReadWrite](#) }

## Public Member Functions

- [BComms](#) ()
- virtual [~BComms](#) ()
- virtual [BError](#) [init](#) ()
- virtual [BError](#) [setPacketMode](#) ([Bool](#) packetMode)
  - Set packet mode.*
- virtual [Bool](#) [packetMode](#) ()
  - Device is in packet mode.*
- virtual [BError](#) [setTimeout](#) ([BTimeout](#) timeoutUs)
  - Set communication timeout.*
- virtual [BError](#) [connect](#) (const char \*resource)
  - Create a connection.*
- virtual [Bool](#) [isConnected](#) ()
- virtual [BError](#) [disconnect](#) ()
  - Disconnect.*
- virtual void [flush](#) ([Flush](#) flush)
- virtual [BUInt](#) [writeAvailable](#) ()
- virtual [BError](#) [write](#) (const void \*data, [BUInt32](#) nBytes, [BUInt32](#) &nTrans)=0
- virtual [BError](#) [writeChunks](#) (const [BDataChunk](#) \*chunks, [BUInt](#) nChunks, [BUInt32](#) &nTrans)
- virtual [BUInt](#) [readAvailable](#) ()
- virtual [BError](#) [read](#) (void \*data, [BUInt32](#) num, [BUInt32](#) &nTrans)=0
- virtual [BError](#) [wait](#) ([BEventWaitSet](#) events, [BTimeout](#) timeoutUs=[BTimeoutForever](#), [BUInt32](#) num=1)
- virtual void [eventQueue](#) ([BEventQueue](#) \*eventQueue, [BUInt32](#) event, [BUInt](#) num=1)

## Protected Attributes

- [Bool](#) oconnected
- [Bool](#) opacketMode
- [BTimeout](#) otimeout
- [BEventQueue](#) \* oeventQueue
- [BUInt32](#) oevent
- [BUInt](#) oeventNum

## 5.9.1 Member Enumeration Documentation

### 5.9.1.1 Flush

```
enum BComms::Flush
```

#### Enumerator

|                                |  |
|--------------------------------|--|
| <a href="#">FlushRead</a>      |  |
| <a href="#">FlushWrite</a>     |  |
| <a href="#">FlushReadWrite</a> |  |

## 5.9.2 Constructor & Destructor Documentation

### 5.9.2.1 BComms()

```
BComms::BComms ( )
```

### 5.9.2.2 ~BComms()

```
BComms::~~BComms ( ) [virtual]
```

## 5.9.3 Member Function Documentation

### 5.9.3.1 connect()

```
BError BComms::connect (
    const char * resource ) [virtual]
```

Create a connection.

### 5.9.3.2 disconnect()

```
BError BComms::disconnect ( ) [virtual]
```

Disconnect.

### 5.9.3.3 eventQueue()

```
void BComms::eventQueue (
    BEventQueue * eventQueue,
    BInt32 event,
    BUInt num = 1 ) [virtual]
```

#### 5.9.3.4 flush()

```
void BComms::flush (
    Flush flush ) [virtual]
```

#### 5.9.3.5 init()

```
BError BComms::init ( ) [virtual]
```

#### 5.9.3.6 isConnected()

```
Bool BComms::isConnected ( ) [virtual]
```

#### 5.9.3.7 packetMode()

```
Bool BComms::packetMode ( ) [virtual]
```

Device is in packet mode.

Reimplemented in [BCan](#).

#### 5.9.3.8 read()

```
virtual BError BComms::read (
    void * data,
    BUInt32 num,
    BUInt32 & nTrans ) [pure virtual]
```

Implemented in [BUsbSerial](#), [BCan](#), [BUart](#), [BUartBasic](#), and [BUsbSerialLock](#).

#### 5.9.3.9 readAvailable()

```
BUInt BComms::readAvailable ( ) [virtual]
```

Reimplemented in [BUsbSerial](#), [BCan](#), [BUart](#), [BUartBasic](#), and [BUsbSerialLock](#).

#### 5.9.3.10 setPacketMode()

```
BError BComms::setPacketMode (
    Bool packetMode ) [virtual]
```

Set packet mode.

#### 5.9.3.11 setTimeout()

```
BError BComms::setTimeout (
    BTimeout timeoutUs ) [virtual]
```

Set communication timeout.

#### 5.9.3.12 wait()

```
BError BComms::wait (
    BEventWaitSet events,
    BTimeout timeoutUs = BTimeoutForever,
    BUInt32 num = 1 ) [virtual]
```

#### 5.9.3.13 write()

```
virtual BError BComms::write (
    const void * data,
    BUInt32 nBytes,
    BUInt32 & nTrans ) [pure virtual]
```

Implemented in [BUbSerial](#), [BCan](#), [BUart](#), [BUartBasic](#), and [BUbSerialLock](#).

#### 5.9.3.14 writeAvailable()

```
BUInt BComms::writeAvailable ( ) [virtual]
```

Reimplemented in [BUbSerial](#), [BCan](#), [BUart](#), [BUartBasic](#), and [BUbSerialLock](#).

### 5.9.3.15 writeChunks()

```
BError BComms::writeChunks (  
    const BDataChunk * chunks,  
    BUInt nChunks,  
    BUInt32 & nTrans ) [virtual]
```

## 5.9.4 Member Data Documentation

### 5.9.4.1 oconnected

```
Bool BComms::oconnected [protected]
```

### 5.9.4.2 oevent

```
BInt32 BComms::oevent [protected]
```

### 5.9.4.3 oeventNum

```
BUInt BComms::oeventNum [protected]
```

### 5.9.4.4 oeventQueue

```
BEventQueue* BComms::oeventQueue [protected]
```

### 5.9.4.5 opacketMode

```
Bool BComms::opacketMode [protected]
```

## 5.9.4.6 otimeout

```
BTimeout BComms::otimeout [protected]
```

The documentation for this class was generated from the following files:

- [BComms.h](#)
- [BComms.cpp](#)

## 5.10 BCondInt Class Reference

[BCondInt](#) class.

```
#include <BCondInt.h>
```

## Public Member Functions

- [BCondInt](#) ()
- [~BCondInt](#) ()
- void [setValue](#) ([BInt](#) value)  
*Set the value. Wakes waiting.*
- [BInt](#) [value](#) ()  
*Current value.*
- [BInt](#) [increment](#) ([BInt](#) v=1)  
*Increment. Wakes waiting.*
- [BInt](#) [decrement](#) ([BInt](#) v=1)  
*Decrement. Wakes waiting.*
- [Bool](#) [waitMoreThanOrEqual](#) ([BInt](#) v, [Bool](#) decrement=0, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait until value is at least the value given.*
- [Bool](#) [waitLessThanOrEqual](#) ([BInt](#) v, [Bool](#) increment=0, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait until value is equal to or below the value given.*
- [Bool](#) [waitLessThan](#) ([BInt](#) v, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait until value is equal to or below the value given.*
- void [operator+=](#) ([BInt](#) v)  
*Add to value. Wakes waiting.*
- void [operator-=](#) ([BInt](#) v)  
*Subtract from value. Wakes waiting.*
- void [operator++](#) (int)  
*Increment value. Wakes waiting.*
- void [operator--](#) (int)  
*Decrement value. Wakes waiting.*

## Private Attributes

- [BMutex](#) omutex
- [BSemaphore](#) osema
- [BInt](#) ovalue

## 5.10.1 Detailed Description

[BCondInt](#) class.

## 5.10.2 Constructor & Destructor Documentation

### 5.10.2.1 BCondInt()

```
BCondInt::BCondInt ( )
```

### 5.10.2.2 ~BCondInt()

```
BCondInt::~~BCondInt ( )
```

## 5.10.3 Member Function Documentation

### 5.10.3.1 decrement()

```
BInt BCondInt::decrement (
    Bint v = 1 )
```

Decrement. Wakes waiting.

### 5.10.3.2 increment()

```
BInt BCondInt::increment (
    Bint v = 1 )
```

Increment. Wakes waiting.

### 5.10.3.3 operator++()

```
void BCondInt::operator++ (
    int ) [inline]
```

Increment value. Wakes waiting.

#### 5.10.3.4 operator+=()

```
void BCondInt::operator+= (
    BInt v ) [inline]
```

Add to value. Wakes waiting.

#### 5.10.3.5 operator--()

```
void BCondInt::operator-- (
    int ) [inline]
```

Decrement value. Wakes waiting.

#### 5.10.3.6 operator-=()

```
void BCondInt::operator-= (
    BInt v ) [inline]
```

Subtract from value. Wakes waiting.

#### 5.10.3.7 setValue()

```
void BCondInt::setValue (
    BInt value )
```

Set the value. Wakes waiting.

#### 5.10.3.8 value()

```
BInt BCondInt::value ( )
```

Current value.

#### 5.10.3.9 waitLessThan()

```
Bool BCondInt::waitLessThan (
    BInt v,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait until value is equal to or below the value given.

#### 5.10.3.10 waitLessThanOrEqual()

```
Bool BCondInt::waitLessThanOrEqual (
    BInt v,
    Bool increment = 0,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait until value is equal to or below the value given.

#### 5.10.3.11 waitMoreThanOrEqual()

```
Bool BCondInt::waitMoreThanOrEqual (
    BInt v,
    Bool decrement = 0,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait until value is at least the value given.

### 5.10.4 Member Data Documentation

#### 5.10.4.1 omutex

```
BMutex BCondInt::omutex [private]
```

#### 5.10.4.2 osema

```
BSemaphore BCondInt::osema [private]
```

#### 5.10.4.3 ovalue

```
BInt BCondInt::ovalue [private]
```

The documentation for this class was generated from the following files:

- [BCondInt.h](#)
- [BCondInt.cpp](#)

## 5.11 BCritical Class Reference

Critical region class.

```
#include <BCritical.h>
```

### Public Member Functions

- void [lock](#) ()  
*Enter critical region.*
- void [unlock](#) ()  
*Leave critical region.*

### Private Attributes

- UBaseType\_t [ostate](#)

#### 5.11.1 Detailed Description

Critical region class.

#### 5.11.2 Member Function Documentation

##### 5.11.2.1 lock()

```
void BCritical::lock ( )
```

Enter critical region.

##### 5.11.2.2 unlock()

```
void BCritical::unlock ( )
```

Leave critical region.

#### 5.11.3 Member Data Documentation

### 5.11.3.1 ostate

```
UBaseType_t BCritical::ostate [private]
```

The documentation for this class was generated from the following file:

- [BCritical.h](#)

## 5.12 BCriticalLock Class Reference

Critical region class with unlock on object deletion.

```
#include <BCritical.h>
```

### Public Member Functions

- [BCriticalLock](#) (int doLock=0)
- [~BCriticalLock](#) ()
- void [lock](#) ()

### Private Attributes

- [BCritical](#) olock

### 5.12.1 Detailed Description

Critical region class with unlock on object deletion.

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 BCriticalLock()

```
BCriticalLock::BCriticalLock (  
    int doLock = 0 )
```

#### 5.12.2.2 ~BCriticalLock()

```
BCriticalLock::~BCriticalLock ( )
```

### 5.12.3 Member Function Documentation

#### 5.12.3.1 lock()

```
void BCriticalLock::lock ( )
```

### 5.12.4 Member Data Documentation

#### 5.12.4.1 olock

```
BCritical BCriticalLock::olock [private]
```

The documentation for this class was generated from the following file:

- [BCritical.h](#)

## 5.13 BDataChunk Class Reference

```
#include <BTypes.h>
```

### Public Member Functions

- [BDataChunk](#) (void \*data=0, BUInt size=0)

### Public Attributes

- void \* [data](#)
- [BUInt](#) [size](#)

### 5.13.1 Constructor & Destructor Documentation

#### 5.13.1.1 BDataChunk()

```
BDataChunk::BDataChunk (
    void * data = 0,
    BUInt size = 0 ) [inline]
```

## 5.13.2 Member Data Documentation

### 5.13.2.1 data

`void* BDataChunk::data`

### 5.13.2.2 size

`BUInt BDataChunk::size`

The documentation for this class was generated from the following file:

- [BTypes.h](#)

## 5.14 BDevices Class Reference

```
#include <BDevices.h>
```

### Public Member Functions

- [BDevices](#) ()
- [BError init](#) ()  
*Initialise.*
- void [enable](#) ([Device](#) device, [Bool](#) on)  
*Enable the given device.*
- void [getClocks](#) ([BClocks](#) &clocks)  
*return clock information*

### 5.14.1 Constructor & Destructor Documentation

#### 5.14.1.1 BDevices()

```
BDevices::BDevices ( )
```

### 5.14.2 Member Function Documentation

#### 5.14.2.1 enable()

```
void BDevices::enable (
    Device device,
    Bool on )
```

Enable the given device.

#### 5.14.2.2 getClocks()

```
void BDevices::getClocks (
    BClocks & clocks )
```

return clock information

#### 5.14.2.3 init()

```
BError BDevices::init ( )
```

Initialise.

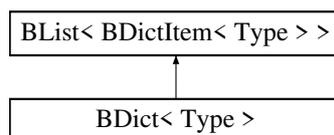
The documentation for this class was generated from the following files:

- [BDevices.h](#)
- [BDevices.cpp](#)

## 5.15 BDict< Type > Class Template Reference

```
#include <BDict.h>
```

Inheritance diagram for BDict< Type >:



### Public Types

- typedef [Blter](#) iterator

## Public Member Functions

- [BDict](#) (int hashSize=100)
- [BDict](#) (const [BDict](#)< Type > &dict)
- int [hasKey](#) (const [BString](#) &k) const
- [BString](#) [key](#) (const [Blter](#) &i) const
- void [clear](#) ()
  - Clear the list.*
- void [insert](#) ([Blter](#) &i, const [BDictItem](#)< Type > &item)
  - Insert item before item.*
- void [append](#) (const [BDictItem](#)< Type > &item)
- void [append](#) (const [BDict](#)< Type > &dict)
- void [del](#) (const [BString](#) &k)
- void [del](#) ([Blter](#) &i)
  - Delete specified item.*
- [Blter](#) [find](#) (const [BString](#) &k) const
- Type & [operator\[\]](#) (const [BString](#) &i)
- Type & [operator\[\]](#) (const [Blter](#) &i)
- const Type & [operator\[\]](#) (const [Blter](#) &i) const
- [BDict](#)< Type > [operator+](#) (const [BDict](#)< Type > &dict) const
- [BDict](#)< Type > & [operator=](#) (const [BDict](#)< Type > &dict)
- void [hashPrint](#) ()

## Private Member Functions

- void [hashAdd](#) (const [BString](#) &k, [Blter](#) iter)
- void [hashDelete](#) (const [BString](#) &k, [Blter](#) iter)
- int [hashFind](#) (const [BString](#) &k, [Blter](#) &iter) const

## Private Attributes

- int [ohashSize](#)
- [BArray](#)< [BList](#)< [Blter](#) > > [ohashLists](#)

## Additional Inherited Members

### 5.15.1 Member Typedef Documentation

#### 5.15.1.1 iterator

```
template<class Type>
typedef Blter BDict< Type >::iterator
```

### 5.15.2 Constructor & Destructor Documentation

**5.15.2.1 BDict()** [1/2]

```
template<class Type >
BDict< Type >::BDict (
    int hashSize = 100 )
```

**5.15.2.2 BDict()** [2/2]

```
template<class Type>
BDict< Type >::BDict (
    const BDict< Type > & dict )
```

**5.15.3 Member Function Documentation****5.15.3.1 append()** [1/2]

```
template<class Type>
void BDict< Type >::append (
    const BDictItem< Type > & item )
```

**5.15.3.2 append()** [2/2]

```
template<class Type>
void BDict< Type >::append (
    const BDict< Type > & dict )
```

**5.15.3.3 clear()**

```
template<class Type >
void BDict< Type >::clear ( ) [virtual]
```

Clear the list.

Reimplemented from [BList< BDictItem< Type > >](#).

#### 5.15.3.4 del() [1/2]

```
template<class Type >
void BDict< Type >::del (
    const BString & k )
```

#### 5.15.3.5 del() [2/2]

```
template<class Type >
void BDict< Type >::del (
    BIter & i ) [virtual]
```

Delete specified item.

Reimplemented from [BList< BDictItem< Type > >](#).

#### 5.15.3.6 find()

```
template<class Type >
BIter BDict< Type >::find (
    const BString & k ) const
```

#### 5.15.3.7 hashAdd()

```
template<class Type >
void BDict< Type >::hashAdd (
    const BString & k,
    BIter iter ) [private]
```

#### 5.15.3.8 hashDelete()

```
template<class Type >
void BDict< Type >::hashDelete (
    const BString & k,
    BIter iter ) [private]
```

### 5.15.3.9 hashFind()

```
template<class Type >
int BDict< Type >::hashFind (
    const BString & k,
    BIter & iter ) const [private]
```

### 5.15.3.10 hashPrint()

```
template<class Type >
void BDict< Type >::hashPrint ( )
```

### 5.15.3.11 hasKey()

```
template<class Type >
int BDict< Type >::hasKey (
    const BString & k ) const
```

### 5.15.3.12 insert()

```
template<class Type>
void BDict< Type >::insert (
    BIter & i,
    const BDictItem< Type > & item ) [virtual]
```

Insert item before item.

Reimplemented from [BList< BDictItem< Type > >](#).

### 5.15.3.13 key()

```
template<class Type >
BString BDict< Type >::key (
    const BIter & i ) const
```

### 5.15.3.14 operator+()

```
template<class Type>
BDict< Type > BDict< Type >::operator+ (
    const BDict< Type > & dict ) const
```

### 5.15.3.15 operator=()

```
template<class Type>
BDict< Type > & BDict< Type >::operator= (
    const BDict< Type > & dict )
```

### 5.15.3.16 operator[]() [1/3]

```
template<class Type >
Type & BDict< Type >::operator[] (
    const BString & i )
```

### 5.15.3.17 operator[]() [2/3]

```
template<class Type >
Type & BDict< Type >::operator[] (
    const BIter & i )
```

### 5.15.3.18 operator[]() [3/3]

```
template<class Type >
const Type & BDict< Type >::operator[] (
    const BIter & i ) const
```

## 5.15.4 Member Data Documentation

### 5.15.4.1 ohashLists

```
template<class Type>
BArray<BList<BIter> > BDict< Type >::ohashLists [private]
```

### 5.15.4.2 ohashSize

```
template<class Type>
int BDict< Type >::ohashSize [private]
```

The documentation for this class was generated from the following file:

- [BDict.h](#)

## 5.16 BDictItem< Type > Class Template Reference

Template based Dictionary class.

```
#include <BDict.h>
```

### Public Member Functions

- [BDictItem](#) (BString k="", Type v=Type())

### Public Attributes

- [BString](#) key
- [Type](#) value

### 5.16.1 Detailed Description

```
template<class Type>  
class BDictItem< Type >
```

Template based Dictionary class.

### 5.16.2 Constructor & Destructor Documentation

#### 5.16.2.1 BDictItem()

```
template<class Type >  
BDictItem< Type >::BDictItem (  
    BString k = "",  
    Type v = Type() ) [inline]
```

### 5.16.3 Member Data Documentation

#### 5.16.3.1 key

```
template<class Type >  
BString BDictItem< Type >::key
```

### 5.16.3.2 value

```
template<class Type >
Type BDictItem< Type >::value
```

The documentation for this class was generated from the following file:

- [BDict.h](#)

## 5.17 BDma Class Reference

```
#include <BDma.h>
```

### Public Types

- enum [Dir](#) { [DirDeviceToMemory](#) = 0, [DirMemoryToDevice](#) = 1, [DirMemoryToMemory](#) = 2 }
- enum [DmaType](#) { [DmaTypeNormal](#), [DmaTypeCircular](#), [DmaTypeDouble](#) }
- enum [Size](#) { [Size8](#) = 0, [Size16](#) = 1, [Size32](#) = 2 }
- enum [Burst](#) { [Burst1](#) = 0, [Burst4](#) = 1, [Burst8](#) = 2, [Burst16](#) = 3 }
- enum [Fifo](#) { [FifoNone](#), [FifoQuarter](#), [FifoHalf](#), [FifoThreeQuarter](#), [FifoFull](#) }
- enum [Status](#) { [StatusNone](#) = 0x0, [StatusComplete](#) = 0x01, [StatusHalfComplete](#) = 0x02, [StatusError](#) = 0x04 }

### Public Member Functions

- [BDma](#) ()
- [BError](#) [init](#) ([BUInt8](#) controller, [BUInt8](#) stream, [BUInt8](#) channel)  
*Initialise the device.*
- void [setCompleteCallback](#) ([BDmaCompleteCallback](#) callBack)
- [BError](#) [stop](#) ([Bool](#) waitForCompletion=1)  
*Stops the DMA and waits until it has stopped if wait is set.*
- [BError](#) [config](#) ([Size](#) memSize, [Bool](#) memInc, [Size](#) deviceSize, [Bool](#) deviceInc, [DmaType](#) dmaType, [BUInt8](#) priority)
- [BError](#) [configFlow](#) ([Bool](#) flowDevice, [Burst](#) memBurst, [Burst](#) deviceBurst, [Fifo](#) fifo)
- [BError](#) [doDma](#) ([Dir](#) dir, void \*fromAddress, void \*toAddress, [BUInt32](#) size)
- void [setNextBuffer](#) (void \*address)  
*In double buffer mode set next buffer address.*
- [BError](#) [wait](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait for completion.*
- [BError](#) [waitStatus](#) ([BUInt8](#) &status, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait for completion.*
- [BUInt32](#) [numLeft](#) ()  
*The number of DMA items to transfer.*
- void [interrupt](#) ()
- void [printStatus](#) ()
- [BUInt8](#) [statusGet](#) ()
- void [statusClear](#) ()

## Public Attributes

- [BUInt8 ocontroller](#)
- [BUInt8 ochannel](#)
- [BUInt8 ostream](#)
- [Size otranSize](#)
- [DmaType odmaType](#)
- [BDmaCompleteCallback ocompleteCallback](#)
- [DMA\\_TypeDef \\* odma](#)
- [DMA\\_Stream\\_TypeDef \\* odmaStream](#)
- [BSemaphore owait](#)
- [BUInt8 ostatus](#)
- [BUInt oerr](#)

*Error flagged.*

## 5.17.1 Member Enumeration Documentation

### 5.17.1.1 Burst

enum [BDma::Burst](#)

#### Enumerator

|         |  |
|---------|--|
| Burst1  |  |
| Burst4  |  |
| Burst8  |  |
| Burst16 |  |

### 5.17.1.2 Dir

enum [BDma::Dir](#)

#### Enumerator

|                   |  |
|-------------------|--|
| DirDeviceToMemory |  |
| DirMemoryToDevice |  |
| DirMemoryToMemory |  |

### 5.17.1.3 DmaType

enum [BDma::DmaType](#)

**Enumerator**

|                 |  |
|-----------------|--|
| DmaTypeNormal   |  |
| DmaTypeCircular |  |
| DmaTypeDouble   |  |

**5.17.1.4 Fifo**

enum `BDma::Fifo`

**Enumerator**

|                  |  |
|------------------|--|
| FifoNone         |  |
| FifoQuarter      |  |
| FifoHalf         |  |
| FifoThreeQuarter |  |
| FifoFull         |  |

**5.17.1.5 Size**

enum `BDma::Size`

**Enumerator**

|        |  |
|--------|--|
| Size8  |  |
| Size16 |  |
| Size32 |  |

**5.17.1.6 Status**

enum `BDma::Status`

**Enumerator**

|                    |  |
|--------------------|--|
| StatusNone         |  |
| StatusComplete     |  |
| StatusHalfComplete |  |
| StatusError        |  |

## 5.17.2 Constructor & Destructor Documentation

### 5.17.2.1 BDma()

```
BDma::BDma ( )
```

## 5.17.3 Member Function Documentation

### 5.17.3.1 config()

```
BError BDma::config (
    Size memSize,
    Bool memInc,
    Size deviceSize,
    Bool deviceInc,
    DmaType dmaType,
    BUInt8 priority )
```

### 5.17.3.2 configFlow()

```
BError BDma::configFlow (
    Bool flowDevice,
    Burst memBurst,
    Burst deviceBurst,
    Fifo fifo )
```

### 5.17.3.3 doDma()

```
BError BDma::doDma (
    Dir dir,
    void * fromAddress,
    void * toAddress,
    BUInt32 size )
```

#### 5.17.3.4 `init()`

```
BError BDma::init (
    BUInt8 controller,
    BUInt8 stream,
    BUInt8 channel )
```

Initialise the device.

#### 5.17.3.5 `interrupt()`

```
void BDma::interrupt ( )
```

#### 5.17.3.6 `numLeft()`

```
BUInt32 BDma::numLeft ( )
```

The number of DMA items to transfer.

#### 5.17.3.7 `printStatus()`

```
void BDma::printStatus ( )
```

#### 5.17.3.8 `setCompleteCallback()`

```
void BDma::setCompleteCallback (
    BDmaCompleteCallback callback )
```

#### 5.17.3.9 `setNextBuffer()`

```
void BDma::setNextBuffer (
    void * address )
```

In double buffer mode set next buffer address.

### 5.17.3.10 statusClear()

```
void BDma::statusClear ( )
```

### 5.17.3.11 statusGet()

```
BUInt8 BDma::statusGet ( )
```

### 5.17.3.12 stop()

```
BError BDma::stop (
    Bool waitForCompletion = 1 )
```

Stops the DMA and waits until it has stopped if wait is set.

### 5.17.3.13 wait()

```
BError BDma::wait (
    BTimeout timeoutUs = BTimeoutForever )
```

Wait for completion.

### 5.17.3.14 waitStatus()

```
BError BDma::waitStatus (
    BUInt8 & status,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait for completion.

## 5.17.4 Member Data Documentation

### 5.17.4.1 ochannel

```
BUInt8 BDma::ochannel
```

#### 5.17.4.2 ocompleteCallback

`BDmaCompleteCallback` `BDma::ocompleteCallback`

#### 5.17.4.3 ocontroller

`BUInt8` `BDma::ocontroller`

#### 5.17.4.4 odma

`DMA_TypeDef*` `BDma::odma`

#### 5.17.4.5 odmaStream

`DMA_Stream_TypeDef*` `BDma::odmaStream`

#### 5.17.4.6 odmaType

`DmaType` `BDma::odmaType`

#### 5.17.4.7 oerr

`BUInt` `BDma::oerr`

Error flagged.

#### 5.17.4.8 ostatus

`BUInt8` `BDma::ostatus`

#### 5.17.4.9 ostream

[BUInt8](#) `BDma::ostream`

#### 5.17.4.10 otranSize

[Size](#) `BDma::otranSize`

#### 5.17.4.11 owait

[BSemaphore](#) `BDma::owait`

The documentation for this class was generated from the following files:

- [BDma.h](#)
- [BDma.cpp](#)

## 5.18 BError Class Reference

```
#include <BError.h>
```

### Public Member Functions

- [BError](#) ([BInt](#) err=[ErrorOk](#), const char \*str=[nullString](#))
- [BError](#) & [set](#) ([BInt](#) err, const char \*str=[nullString](#))
- [BInt](#) num () const
- const char \* [str](#) () const
- [operator int](#) () const

### Static Public Attributes

- static char [nullString](#) [1] = { 0 }

### Private Attributes

- [BInt](#) oerr
- const char \* [ostr](#)

#### 5.18.1 Constructor & Destructor Documentation

### 5.18.1.1 BError()

```
BError::BError (
    BInt err = ErrorOk,
    const char * str = nullString ) [inline]
```

## 5.18.2 Member Function Documentation

### 5.18.2.1 num()

```
BInt BError::num ( ) const [inline]
```

### 5.18.2.2 operator int()

```
BError::operator int ( ) const [inline]
```

### 5.18.2.3 set()

```
BError& BError::set (
    BInt err,
    const char * str = nullString ) [inline]
```

### 5.18.2.4 str()

```
const char* BError::str ( ) const [inline]
```

## 5.18.3 Member Data Documentation

### 5.18.3.1 nullString

```
char BError::nullString = { 0 } [static]
```

### 5.18.3.2 oerr

```
BInt BError::oerr [private]
```

### 5.18.3.3 ostr

```
const char* BError::ostr [private]
```

The documentation for this class was generated from the following files:

- [BError.h](#)
- [BError.cpp](#)

## 5.19 BEvent Class Reference

```
#include <BEvent.h>
```

### Public Member Functions

- [BEvent](#) (BUInt32 type=BEventTypeNone, BUInt32 arg=0)
- [BUInt32 type](#) ()
- [BUInt32 arg](#) ()

### Private Attributes

- [BUInt32 otype](#)  
*The events type.*
- [BUInt32 oarg](#)  
*The events argument.*

## 5.19.1 Constructor & Destructor Documentation

### 5.19.1.1 BEvent()

```
BEvent::BEvent (
    BUInt32 type = BEventTypeNone,
    BUInt32 arg = 0 )
```

## 5.19.2 Member Function Documentation

### 5.19.2.1 arg()

```
BUInt32 BEvent::arg ( )
```

### 5.19.2.2 type()

```
BUInt32 BEvent::type ( )
```

## 5.19.3 Member Data Documentation

### 5.19.3.1 oarg

```
BUInt32 BEvent::oarg [private]
```

The events argument.

### 5.19.3.2 otype

```
BUInt32 BEvent::otype [private]
```

The events type.

The documentation for this class was generated from the following files:

- [BEvent.h](#)
- [BEvent.cpp](#)

## 5.20 BFifo< Type > Class Template Reference

```
#include <BFifo.h>
```

## Public Member Functions

- [BFifo \(BUInt size\)](#)
- [~BFifo \(\)](#)
- [void clear \(\)](#)
- [BUInt size \(\)](#)  
*Returns fifo size.*
- [BError resize \(BUInt size\)](#)  
*Resize FIFO, clears it as well.*
- [BError rebase \(\)](#)  
*Rebases fifo so read pointer is at zero moving memory as needed.*
- [BUInt writeAvailable \(\)](#)  
*How many items that can be written.*
- [BUInt writeAvailableChunk \(\)](#)  
*How many items that can be written in a chunk.*
- [BError write \(const Type v\)](#)  
*Write a single item.*
- [BError write \(const Type \\*data, BUInt num\)](#)  
*Write a set of items. Can only write a maximum of [writeAvailableChunk\(\)](#) to save going beyond end of FIFO buffer.*
- [Type \\* writeData \(\)](#)  
*Returns a pointer to the data.*
- [Type \\* writeData \(BUInt &num\)](#)  
*Returns a pointer to the data and how many can be written in a chunk.*
- [void writeDone \(BUInt num\)](#)  
*Indicates when write is complete.*
- [void writeBackup \(BUInt num\)](#)  
*Backup, remove num items at end of fifo. Careful, make sure read is not already happening.*
- [BUInt readAvailable \(\)](#)  
*How many items are available to read.*
- [BUInt readAvailableChunk \(\)](#)  
*How many items are available to read in a chunk.*
- [Type read \(\)](#)  
*Read one item.*
- [BError read \(Type \\*data, BUInt num\)](#)  
*Read a set of items.*
- [BError readPeak \(Type \\*data, BUInt num\)](#)  
*Read a set of items leaving them in the buffer.*
- [Type \\* readData \(\)](#)  
*Returns a pointer to the data.*
- [Type \\* readData \(BUInt &num\)](#)  
*Returns a pointer to the data and how many can be read in a chunk.*
- [void readDone \(BUInt num\)](#)
- [Type readPos \(BUInt pos\)](#)  
*Read item at given offset from current read position.*
- [void writePos \(BUInt pos, const Type &v\)](#)  
*Write item at given offset from current read position.*
- [Type & operator\[\] \(int pos\)](#)  
*Direct access to read samples in buffer.*

## Protected Attributes

- [BUInt osize](#)  
*The size of the FIFO.*
- `Type * odata`  
*FIFO memory buffer.*
- `volatile BUInt owritePos`  
*The write pointer.*
- `volatile BUInt oreadPos`  
*The read pointer.*

## 5.20.1 Constructor & Destructor Documentation

### 5.20.1.1 Bfifo()

```
template<class Type>
Bfifo< Type >::Bfifo (
    BUInt size )
```

### 5.20.1.2 ~Bfifo()

```
template<class Type>
Bfifo< Type >::~~Bfifo ( )
```

## 5.20.2 Member Function Documentation

### 5.20.2.1 clear()

```
template<class Type>
void Bfifo< Type >::clear ( )
```

### 5.20.2.2 operator[]()

```
template<class Type>
Type& Bfifo< Type >::operator[] (
    int pos )
```

Direct access to read samples in buffer.

### 5.20.2.3 read() [1/2]

```
template<class Type>
Type BFifo< Type >::read ( )
```

Read one item.

### 5.20.2.4 read() [2/2]

```
template<class Type>
BError BFifo< Type >::read (
    Type * data,
    BUInt num )
```

Read a set of items.

### 5.20.2.5 readAvailable()

```
template<class Type>
BUInt BFifo< Type >::readAvailable ( )
```

How many items are available to read.

### 5.20.2.6 readAvailableChunk()

```
template<class Type>
BUInt BFifo< Type >::readAvailableChunk ( )
```

How many items are available to read in a chunk.

### 5.20.2.7 readData() [1/2]

```
template<class Type>
Type* BFifo< Type >::readData ( )
```

Returns a pointer to the data.

#### 5.20.2.8 readData() [2/2]

```
template<class Type>
Type* BFifo< Type >::readData (
    BUInt & num )
```

Returns a pointer to the data and how many can be read in a chunk.

#### 5.20.2.9 readDone()

```
template<class Type>
void BFifo< Type >::readDone (
    BUInt num )
```

#### 5.20.2.10 readPeak()

```
template<class Type>
BError BFifo< Type >::readPeak (
    Type * data,
    BUInt num )
```

Read a set of items leaving them in the buffer.

#### 5.20.2.11 readPos()

```
template<class Type>
Type BFifo< Type >::readPos (
    BUInt pos )
```

Read item at given offset from current read position.

#### 5.20.2.12 rebase()

```
template<class Type>
BError BFifo< Type >::rebase ( )
```

Rebases fifo so read pointer is at zero moving memory as needed.

#### 5.20.2.13 `resize()`

```
template<class Type>
BError BFifo< Type >::resize (
    BUInt size )
```

Resize FIFO, clears it as well.

#### 5.20.2.14 `size()`

```
template<class Type>
BUInt BFifo< Type >::size ( )
```

Returns fifo size.

#### 5.20.2.15 `write()` [1/2]

```
template<class Type>
BError BFifo< Type >::write (
    const Type v )
```

Write a single item.

#### 5.20.2.16 `write()` [2/2]

```
template<class Type>
BError BFifo< Type >::write (
    const Type * data,
    BUInt num )
```

Write a set of items. Can only write a maximum of `writeAvailableChunk()` to save going beyond end of FIFO buffer.

#### 5.20.2.17 `writeAvailable()`

```
template<class Type>
BUInt BFifo< Type >::writeAvailable ( )
```

How many items that can be written.

#### 5.20.2.18 writeAvailableChunk()

```
template<class Type>
BUInt BFifo< Type >::writeAvailableChunk ( )
```

How many items that can be written in a chunk.

#### 5.20.2.19 writeBackup()

```
template<class Type>
void BFifo< Type >::writeBackup (
    BUInt num )
```

Backup, remove num items at end of fifo. Careful, make sure read is not already happening.

#### 5.20.2.20 writeData() [1/2]

```
template<class Type>
Type* BFifo< Type >::writeData ( )
```

Returns a pointer to the data.

#### 5.20.2.21 writeData() [2/2]

```
template<class Type>
Type* BFifo< Type >::writeData (
    BUInt & num )
```

Returns a pointer to the data and how many can be written in a chunk.

#### 5.20.2.22 writeDone()

```
template<class Type>
void BFifo< Type >::writeDone (
    BUInt num )
```

Indicates when write is complete.

### 5.20.2.23 writePos()

```
template<class Type>
void BFifo< Type >::writePos (
    BUInt pos,
    const Type & v )
```

Write item at given offset from current read position.

## 5.20.3 Member Data Documentation

### 5.20.3.1 odata

```
template<class Type>
Type* BFifo< Type >::odata [protected]
```

FIFO memory buffer.

### 5.20.3.2 oreadPos

```
template<class Type>
volatile BUInt BFifo< Type >::oreadPos [protected]
```

The read pointer.

### 5.20.3.3 osize

```
template<class Type>
BUInt BFifo< Type >::osize [protected]
```

The size of the FIFO.

### 5.20.3.4 owritePos

```
template<class Type>
volatile BUInt BFifo< Type >::owritePos [protected]
```

The write pointer.

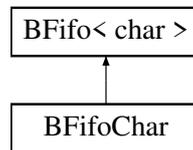
The documentation for this class was generated from the following file:

- [BFifo.h](#)

## 5.21 BFifoChar Class Reference

```
#include <BHttpServer.h>
```

Inheritance diagram for BFifoChar:



### Public Member Functions

- [BFifoChar](#) ([BUInt](#) fifoSize)
- [Bool](#) [find](#) (const char \*pattern, [BUInt](#) &pos)  
*Search fifo for string.*
- [Bool](#) [readInt](#) (int &value)
- [BUInt32](#) [getBUInt32](#) ([BUInt](#) pos)
- void [putBUInt32](#) ([BUInt](#) pos, [BUInt32](#) v)
- void [debugPrint](#) ([BUInt](#) n=0)

### Additional Inherited Members

#### 5.21.1 Constructor & Destructor Documentation

##### 5.21.1.1 BFifoChar()

```
BFifoChar::BFifoChar (
    BUInt fifoSize ) [inline]
```

#### 5.21.2 Member Function Documentation

##### 5.21.2.1 debugPrint()

```
void BFifoChar::debugPrint (
    BUInt n = 0 )
```

### 5.21.2.2 find()

```
Bool BFifoChar::find (
    const char * pattern,
    BUInt & pos ) [inline]
```

Search fifo for string.

### 5.21.2.3 getBUInt32()

```
BUInt32 BFifoChar::getBUInt32 (
    BUInt pos ) [inline]
```

### 5.21.2.4 putBUInt32()

```
void BFifoChar::putBUInt32 (
    BUInt pos,
    BUInt32 v ) [inline]
```

### 5.21.2.5 readInt()

```
Bool BFifoChar::readInt (
    int & value ) [inline]
```

The documentation for this class was generated from the following file:

- [BHttpServer.h](#)

## 5.22 BFirmwareInfo Struct Reference

```
#include <BFirmware.h>
```

### Public Attributes

- [BUInt32 magic](#)
- [BUInt32 length](#)
- [BUInt32 checksum](#)
- [BUInt8 type](#)
- [BUInt8 ver0](#)
- [BUInt8 ver1](#)
- [BUInt8 ver2](#)

## 5.22.1 Member Data Documentation

### 5.22.1.1 checksum

`BUInt32` `BFirmwareInfo::checksum`

### 5.22.1.2 length

`BUInt32` `BFirmwareInfo::length`

### 5.22.1.3 magic

`BUInt32` `BFirmwareInfo::magic`

### 5.22.1.4 type

`BUInt8` `BFirmwareInfo::type`

### 5.22.1.5 ver0

`BUInt8` `BFirmwareInfo::ver0`

### 5.22.1.6 ver1

`BUInt8` `BFirmwareInfo::ver1`

### 5.22.1.7 ver2

`BUInt8` `BFirmwareInfo::ver2`

The documentation for this struct was generated from the following file:

- [BFirmware.h](#)

## 5.23 BFlash Class Reference

```
#include <BFlash.h>
```

### Public Member Functions

- [BFlash](#) ()
- [BError init](#) ()  
*Initialise.*
- [BError programEnable](#) (Bool on)  
*Enable/disable programming.*
- [BError programErase](#) (BUInt32 page)  
*Erase the page.*
- [BError programWord](#) (BUInt32 address, BUInt32 data)  
*Program a 32bit wide memory location,.*
- [BError programWord](#) (BUInt32 address, BUInt16 data)  
*Program a 16bit wide memory location,.*
- [BError disableAccess](#) ()  
*Disable access to FLASH to protect against program theft.*
- [BError enableAccess](#) ()  
*Enable access to FLASH.*
- void [writeProtect](#) (BUInt32 page, Bool on)  
*Write protect the given page.*
- Bool [getWriteProtect](#) (BUInt32 page)  
*Gets write protect status for the given page.*
- [BUInt32 page](#) (BUInt32 address)  
*Return page address is in.*
- [BUInt32 pageAddress](#) (BUInt32 page)  
*Return start address of page.*

### Protected Member Functions

- [BError waitUntilReady](#) (Bool quick=0)  
*Waits while FLASH operation is on-going.*

### Private Attributes

- [BUInt32 odummy](#)  
*To pad object to 32bit boundary.*

### 5.23.1 Constructor & Destructor Documentation

#### 5.23.1.1 BFlash()

```
BFlash::BFlash ( )
```

## 5.23.2 Member Function Documentation

### 5.23.2.1 disableAccess()

```
BError BFlash::disableAccess ( )
```

Disable access to FLASH to protect against program theft.

### 5.23.2.2 enableAccess()

```
BError BFlash::enableAccess ( )
```

Enable access to FLASH.

### 5.23.2.3 getWriteProtect()

```
Bool BFlash::getWriteProtect (
    BUInt32 page )
```

Gets write protect status for the given page.

### 5.23.2.4 init()

```
BError BFlash::init ( )
```

Initialise.

### 5.23.2.5 page()

```
BInt32 BFlash::page (
    BUInt32 address )
```

Return page address is in.

#### 5.23.2.6 pageAddress()

```
BUInt32 BFlash::pageAddress (
    BUInt32 page )
```

Return start address of page.

#### 5.23.2.7 programEnable()

```
BError BFlash::programEnable (
    Bool on )
```

Enable/disable programming.

#### 5.23.2.8 programErase()

```
BError BFlash::programErase (
    BUInt page )
```

Erase the page.

#### 5.23.2.9 programWord() [1/2]

```
BError BFlash::programWord (
    BUInt32 address,
    BUInt32 data )
```

Program a 32bit wide memory location,.

#### 5.23.2.10 programWord() [2/2]

```
BError BFlash::programWord (
    BUInt32 address,
    BUInt16 data )
```

Program a 16bit wide memory location,.

### 5.23.2.11 waitUntillReady()

```
BError BFlash::waitUntillReady (
    Bool quick = 0 ) [protected]
```

Waits while FLASH operation is on-going.

### 5.23.2.12 writeProtect()

```
void BFlash::writeProtect (
    BUInt32 page,
    Bool on )
```

Write protect the given page.

## 5.23.3 Member Data Documentation

### 5.23.3.1 odummy

```
BUInt32 BFlash::odummy [private]
```

To pad object to 32bit boundary.

The documentation for this class was generated from the following files:

- [BFlash.h](#)
- [BFlash.cpp](#)

## 5.24 BGPIO Class Reference

```
#include <BGpio.h>
```

### Public Types

- enum [Mode](#) { [ModeInput](#) = 0x00, [ModeOutput](#) = 0x01, [ModeAlternate](#) = 0x02, [ModeAnalog](#) = 0x03 }
- enum [OutType](#) { [OutTypePushPull](#) = 0x00, [OutTypeOpenDrain](#) = 0x01 }
- enum [Pull](#) { [PullNone](#) = 0x00, [PullUp](#) = 0x01, [PullDown](#) = 0x02 }
- enum [Speed](#) { [Speed1MHz](#) = 0x00, [Speed25MHz](#) = 0x01, [Speed50MHz](#) = 0x02, [Speed100MHz](#) = 0x03 }
- enum [Source](#) {
  - [Source0](#) = 0x00, [Source1](#) = 0x01, [Source2](#) = 0x02, [Source3](#) = 0x03,
  - [Source4](#) = 0x04, [Source5](#) = 0x05, [Source6](#) = 0x06, [Source7](#) = 0x07,
  - [Source8](#) = 0x08, [Source9](#) = 0x09, [Source10](#) = 0x0a, [Source11](#) = 0x0b,
  - [Source12](#) = 0x0c, [Source13](#) = 0x0d, [Source14](#) = 0x0e, [Source15](#) = 0x0f,
  - [SourceTim1](#) = 0x01, [SourceTim2](#) = 0x01, [SourceTim3](#) = 0x02, [SourceTim4](#) = 0x02,
  - [SourceTim5](#) = 0x02, [SourceTim8](#) = 0x03, [SourceTim9](#) = 0x03, [SourceTim10](#) = 0x03,
  - [SourceTim11](#) = 0x03, [SourceTim12](#) = 0x09, [SourceTim13](#) = 0x09, [SourceTim14](#) = 0x09,
  - [SourceSpi1](#) = 0x05, [SourceSpi2](#) = 0x05, [SourceSpi3](#) = 0x06, [SourceI2c1](#) = 0x04,
  - [SourceI2c2](#) = 0x04, [SourceI2c3](#) = 0x04, [SourceUart123](#) = 0x07, [SourceUart456](#) = 0x08,
  - [SourceCan](#) = 0x09, [SourceFsmc](#) = 0x0c, [SourceSdio](#) = 0x0c }

## Public Member Functions

- [BGpio](#) ()
- void [init](#) ()
  - Initialise the gpio system.*
- void [initPin](#) ([Pin](#) pin, [Mode](#) mode, [OutType](#) outType=[OutTypePushPull](#), [Pull](#) pull=[PullNone](#), [Speed](#) speed=[Speed100MHz](#), [Source](#) source=[Source0](#))
  - Set the pins mode.*
- void [setAlternateSource](#) ([Pin](#) pin, [Source](#) source)
  - Set the pins alternate source.*
- void [setPin](#) ([Pin](#) pin, [UInt8](#) state)
  - Set the ping to the given state, Off = 0, On = 1.*
- void [togglePin](#) ([Pin](#) pin)
  - Toggle the pins state.*
- [UInt8](#) [getPin](#) ([Pin](#) pin)
  - Get the pins state.*

### 5.24.1 Member Enumeration Documentation

#### 5.24.1.1 Mode

```
enum BGpio::Mode
```

##### Enumerator

|                               |  |
|-------------------------------|--|
| <a href="#">ModeInput</a>     |  |
| <a href="#">ModeOutput</a>    |  |
| <a href="#">ModeAlternate</a> |  |
| <a href="#">ModeAnalog</a>    |  |

#### 5.24.1.2 OutType

```
enum BGpio::OutType
```

##### Enumerator

|                                  |  |
|----------------------------------|--|
| <a href="#">OutTypePushPull</a>  |  |
| <a href="#">OutTypeOpenDrain</a> |  |

## 5.24.1.3 Pull

```
enum BGpio::Pull
```

## Enumerator

|          |  |
|----------|--|
| PullNone |  |
| PullUp   |  |
| PullDown |  |

## 5.24.1.4 Source

```
enum BGpio::Source
```

## Enumerator

|             |  |
|-------------|--|
| Source0     |  |
| Source1     |  |
| Source2     |  |
| Source3     |  |
| Source4     |  |
| Source5     |  |
| Source6     |  |
| Source7     |  |
| Source8     |  |
| Source9     |  |
| Source10    |  |
| Source11    |  |
| Source12    |  |
| Source13    |  |
| Source14    |  |
| Source15    |  |
| SourceTim1  |  |
| SourceTim2  |  |
| SourceTim3  |  |
| SourceTim4  |  |
| SourceTim5  |  |
| SourceTim8  |  |
| SourceTim9  |  |
| SourceTim10 |  |
| SourceTim11 |  |
| SourceTim12 |  |
| SourceTim13 |  |
| SourceTim14 |  |
| SourceSpi1  |  |
| SourceSpi2  |  |
| SourceSpi3  |  |
| SourceI2c1  |  |
| SourceI2c2  |  |

### Enumerator

|               |  |
|---------------|--|
| SourceI2c3    |  |
| SourceUart123 |  |
| SourceUart456 |  |
| SourceCan     |  |
| SourceFsmc    |  |
| SourceSdio    |  |

### 5.24.1.5 Speed

```
enum BGpio::Speed
```

### Enumerator

|             |  |
|-------------|--|
| Speed1MHz   |  |
| Speed25MHz  |  |
| Speed50MHz  |  |
| Speed100MHz |  |

## 5.24.2 Constructor & Destructor Documentation

### 5.24.2.1 BGpio()

```
BGpio::BGpio ( )
```

## 5.24.3 Member Function Documentation

### 5.24.3.1 getPin()

```
UInt8 BGpio::getPin (
    Pin pin )
```

Get the pins state.

### 5.24.3.2 `init()`

```
void BGpio::init ( )
```

Initialise the gpio system.

### 5.24.3.3 `initPin()`

```
void BGpio::initPin (
    Pin pin,
    Mode mode,
    OutType outType = OutTypePushPull,
    Pull pull = PullNone,
    Speed speed = Speed100MHz,
    Source source = Source0 )
```

Set the pins mode.

### 5.24.3.4 `setAlternateSource()`

```
void BGpio::setAlternateSource (
    Pin pin,
    Source source )
```

Set the pins alternate source.

### 5.24.3.5 `setPin()`

```
void BGpio::setPin (
    Pin pin,
    UInt8 state )
```

Set the ping to the given state, Off = 0, On = 1.

### 5.24.3.6 `togglePin()`

```
void BGpio::togglePin (
    Pin pin )
```

Toggle the pins state.

The documentation for this class was generated from the following files:

- [BGpio.h](#)
- [BGpio.cpp](#)

## 5.25 BHeap Class Reference

```
#include <BHeap.h>
```

### Public Member Functions

- [BHeap](#) ()
- void [init](#) (void \*base, size\_t length)
- void \* [allocate](#) (size\_t length)

### Public Attributes

- char \* [obase](#)
- char \* [otop](#)
- char \* [onext](#)

### 5.25.1 Constructor & Destructor Documentation

#### 5.25.1.1 BHeap()

```
BHeap::BHeap ( )
```

### 5.25.2 Member Function Documentation

#### 5.25.2.1 allocate()

```
void * BHeap::allocate (
    size_t length )
```

#### 5.25.2.2 init()

```
void BHeap::init (
    void * base,
    size_t length )
```

### 5.25.3 Member Data Documentation

### 5.25.3.1 obase

```
char* BHeap::obase
```

### 5.25.3.2 onext

```
char* BHeap::onext
```

### 5.25.3.3 otop

```
char* BHeap::otop
```

The documentation for this class was generated from the following files:

- [BHeap.h](#)
- [BHeap.cpp](#)

## 5.26 BHtml Class Reference

```
#include <BHtml.h>
```

### Public Member Functions

- [BHtml](#) (const [BString](#) &tag="", const [BString](#) &args="", const [BString](#) &text="")
- [~BHtml](#) ()
- [BHtml](#) \* [append](#) (const [BHtml](#) &node)
- [BHtml](#) \* [append](#) (const [BString](#) &tag="", const [BString](#) &args="", const [BString](#) &text="")
- [BHtml](#) \* [appendText](#) (const [BString](#) &text)
- [BHtml](#) \* [appendFormField](#) ([BString](#) name, [BString](#) title, [BString](#) type, [BString](#) param, [BString](#) value, [BString](#) arg="", int submit=0)
- void [appendFormItem](#) ([BString](#) name, [BString](#) title, [BString](#) type, [BString](#) param, [BString](#) value)
- void [render](#) ([HtmlWriteFunc](#) writeFunc)
- [BString](#) [render](#) ()

### Private Attributes

- [BString](#) otag
- [BString](#) oargs
- [BList](#)< [BHtml](#) > nodes

### 5.26.1 Constructor & Destructor Documentation

### 5.26.1.1 BHtml()

```
BHtml::BHtml (
    const BString & tag = "",
    const BString & args = "",
    const BString & text = "" )
```

### 5.26.1.2 ~BHtml()

```
BHtml::~~BHtml ( )
```

## 5.26.2 Member Function Documentation

### 5.26.2.1 append() [1/2]

```
BHtml * BHtml::append (
    const BHtml & node )
```

### 5.26.2.2 append() [2/2]

```
BHtml * BHtml::append (
    const BString & tag = "",
    const BString & args = "",
    const BString & text = "" )
```

### 5.26.2.3 appendFormField()

```
BHtml * BHtml::appendFormField (
    BString name,
    BString title,
    BString type,
    BString param,
    BString value,
    BString arg = "",
    int submit = 0 )
```

#### 5.26.2.4 appendFormItem()

```
void BHtml::appendFormItem (
    BString name,
    BString title,
    BString type,
    BString param,
    BString value )
```

#### 5.26.2.5 appendText()

```
BHtml * BHtml::appendText (
    const BString & text )
```

#### 5.26.2.6 render() [1/2]

```
void BHtml::render (
    HtmlWriteFunc writeFunc )
```

#### 5.26.2.7 render() [2/2]

```
BString BHtml::render ( )
```

### 5.26.3 Member Data Documentation

#### 5.26.3.1 nodes

```
BList<BHtml> BHtml::nodes [private]
```

#### 5.26.3.2 oargs

```
BString BHtml::oargs [private]
```

## 5.26.3.3 otag

```
BString BHtml::otag [private]
```

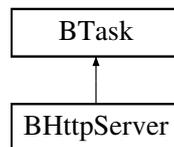
The documentation for this class was generated from the following files:

- [BHtml.h](#)
- [BHtml.cpp](#)

## 5.27 BHttpRequest Class Reference

```
#include <BHttpRequest.h>
```

Inheritance diagram for BHttpRequest:



### Public Types

- enum [ReadType](#) { [ReadTypeData](#), [ReadTypeLine](#), [ReadTypeDataLine](#) }

### Public Member Functions

- [BHttpRequest](#) ([BUInt](#) rxFifoSize=1500, [BUInt](#) txFifoSize=1500)
- [BError](#) [init](#) ()
- void [run](#) ()
  - The tasks main run loop.*
- virtual [BError](#) [processUrl](#) ()
- virtual [BError](#) [processHome](#) ()
- virtual [BError](#) [pageSend](#) (const char \*title, const char \*str)
- virtual [BError](#) [pageSendError](#) (const char \*str, const char \*status="403 Error")
- virtual [BError](#) [pageSendHead](#) (const char \*title)
- virtual [BError](#) [pageSendTail](#) ()
- void [cookieSet](#) ([BString](#) name, [BString](#) value)
- [BString](#) [cookieGet](#) ([BString](#) name)
- void [writeTableRow](#) (const char \*name, const char \*title, const char \*value, const char \*units=0, int err← Num=0)
- void [writeTableInput](#) (const char \*name, const char \*title, const char \*type, const char \*value, const char \*\*args=0)
- void [writeTableInput](#) (const char \*name, const char \*title, const char \*type, const char \*value, [BString](#) str)
- [BError](#) [writeHead](#) (const char \*status, const char \*contentType, [BUInt](#) size=0, [Bool](#) cache=0, [Bool](#) chunked=0, [BStringList](#) extra=[BStringList](#)())
- [BError](#) [writeString](#) (const char \*str)
- [BError](#) [writeChunk](#) (const void \*data, int len)
- [BError](#) [writeFlush](#) ()
- [BError](#) [writeTail](#) ()
- virtual [BError](#) [setHeader](#) (const char \*status, const char \*contentType, [BUInt](#) size=0, [Bool](#) cache=0, [Bool](#) chunked=0, [BStringList](#) extra=[BStringList](#)())
- virtual [BError](#) [processRequest](#) ()
- [BError](#) [readData](#) ([BUInt](#) len, [BUInt32](#) &nt, [ReadType](#) readType)

## Protected Member Functions

- void \* [function](#) ()
- [BError devInit](#) ()
- void [devRun](#) ()
- [BError devWrite](#) (const void \*[data](#), [BUInt](#) [len](#), [BUInt32](#) &[nt](#))
- [BError devProcess](#) ([Bool](#) [wait](#))

## Protected Attributes

- [BFifoChar orxFifo](#)
- [BFifoChar otxFifo](#)
- char \* [obuffer](#)
- [BUInt obufferNext](#)
- [BString ocmd](#)
- [BString ourl](#)
- [BString oget](#)
- [BUInt ocontentLen](#)
- [BString ocontentTypeRecv](#)
- [BString ocontentTypeSend](#)
- [BString oboundary](#)
- [BString oauthorisation](#)
- [BDictString ocookies](#)
- [BDictString ocookiesSend](#)
- const char \* [ohead\\_status](#)
- const char \* [ohead\\_contentType](#)
- [BUInt ohead\\_size](#)
- [Bool ohead\\_cache](#)
- [Bool ohead\\_chunked](#)
- [BStringList ohead\\_extra](#)
- struct netconn \* [osocketListen](#)
- struct netconn \* [osocket](#)

## Additional Inherited Members

### 5.27.1 Member Enumeration Documentation

#### 5.27.1.1 ReadType

enum [BHttpServer::ReadType](#)

#### Enumerator

|                                  |  |
|----------------------------------|--|
| <a href="#">ReadTypeData</a>     |  |
| <a href="#">ReadTypeLine</a>     |  |
| <a href="#">ReadTypeDataLine</a> |  |

## 5.27.2 Constructor & Destructor Documentation

### 5.27.2.1 BHttpRequest()

```
BHttpRequest::BHttpRequest (
    BUInt rxFifoSize = 1500,
    BUInt txFifoSize = 1500 )
```

## 5.27.3 Member Function Documentation

### 5.27.3.1 cookieGet()

```
BString BHttpRequest::cookieGet (
    BString name )
```

### 5.27.3.2 cookieSet()

```
void BHttpRequest::cookieSet (
    BString name,
    BString value )
```

### 5.27.3.3 devInit()

```
BError BHttpRequest::devInit ( ) [protected]
```

### 5.27.3.4 devProcess()

```
BError BHttpRequest::devProcess (
    Bool wait ) [protected]
```

### 5.27.3.5 devRun()

```
void BHttpRequest::devRun ( ) [protected]
```

### 5.27.3.6 devWrite()

```
BError BHttpServer::devWrite (
    const void * data,
    BUInt len,
    BUInt32 & nt ) [protected]
```

### 5.27.3.7 function()

```
void * BHttpServer::function ( ) [protected]
```

### 5.27.3.8 init()

```
BError BHttpServer::init ( )
```

### 5.27.3.9 pageSend()

```
BError BHttpServer::pageSend (
    const char * title,
    const char * str ) [virtual]
```

### 5.27.3.10 pageSendError()

```
BError BHttpServer::pageSendError (
    const char * str,
    const char * status = "403 Error" ) [virtual]
```

### 5.27.3.11 pageSendHead()

```
BError BHttpServer::pageSendHead (
    const char * title ) [virtual]
```

### 5.27.3.12 pageSendTail()

```
BError BHttpServer::pageSendTail ( ) [virtual]
```

### 5.27.3.13 processHome()

```
BError BHttpRequest::processHome ( ) [virtual]
```

### 5.27.3.14 processRequest()

```
BError BHttpRequest::processRequest ( ) [virtual]
```

### 5.27.3.15 processUrl()

```
BError BHttpRequest::processUrl ( ) [virtual]
```

### 5.27.3.16 readData()

```
BError BHttpRequest::readData (
    BUInt len,
    BUInt32 & nt,
    ReadType readType )
```

### 5.27.3.17 run()

```
void BHttpRequest::run ( ) [virtual]
```

The tasks main run loop.

Reimplemented from [BTask](#).

### 5.27.3.18 setHeader()

```
BError BHttpRequest::setHeader (
    const char * status,
    const char * contentType,
    BUInt size = 0,
    Bool cache = 0,
    Bool chunked = 0,
    BStringList extra = BStringList() ) [virtual]
```

### 5.27.3.19 writeChunk()

```
BError BHttpServer::writeChunk (
    const void * data,
    int len )
```

### 5.27.3.20 writeFlush()

```
BError BHttpServer::writeFlush ( )
```

### 5.27.3.21 writeHead()

```
BError BHttpServer::writeHead (
    const char * status,
    const char * contentType,
    BUInt size = 0,
    Bool cache = 0,
    Bool chunked = 0,
    BStringList extra = BStringList() )
```

### 5.27.3.22 writeString()

```
BError BHttpServer::writeString (
    const char * str )
```

### 5.27.3.23 writeTableInput() [1/2]

```
void BHttpServer::writeTableInput (
    const char * name,
    const char * title,
    const char * type,
    const char * value,
    const char ** args = 0 )
```

### 5.27.3.24 writeTableInput() [2/2]

```
void BHttpServer::writeTableInput (
    const char * name,
    const char * title,
    const char * type,
    const char * value,
    BString str )
```

#### 5.27.3.25 writeTableRow()

```
void BHttpRequest::writeTableRow (
    const char * name,
    const char * title,
    const char * value,
    const char * units = 0,
    int errNum = 0 )
```

#### 5.27.3.26 writeTail()

```
BError BHttpRequest::writeTail ( )
```

### 5.27.4 Member Data Documentation

#### 5.27.4.1 oauthorisation

```
BString BHttpRequest::oauthorisation [protected]
```

#### 5.27.4.2 oboundary

```
BString BHttpRequest::oboundary [protected]
```

#### 5.27.4.3 obuffer

```
char* BHttpRequest::obuffer [protected]
```

#### 5.27.4.4 obufferNext

```
BUInt BHttpRequest::obufferNext [protected]
```

#### 5.27.4.5 ocmd

`BString` `BHttpServer::ocmd` [protected]

#### 5.27.4.6 ocontentLen

`BUInt` `BHttpServer::ocontentLen` [protected]

#### 5.27.4.7 ocontentTypeRecv

`BString` `BHttpServer::ocontentTypeRecv` [protected]

#### 5.27.4.8 ocontentTypeSend

`BString` `BHttpServer::ocontentTypeSend` [protected]

#### 5.27.4.9 ocookies

`BDictString` `BHttpServer::ocookies` [protected]

#### 5.27.4.10 ocookiesSend

`BDictString` `BHttpServer::ocookiesSend` [protected]

#### 5.27.4.11 oget

`BString` `BHttpServer::oget` [protected]

#### 5.27.4.12 ohead\_cache

`Bool` `BHttpServer::ohead_cache` [protected]

#### 5.27.4.13 `ohead_chunked`

`Bool` BHttpRequest::ohead\_chunked [protected]

#### 5.27.4.14 `ohead_contentType`

`const char*` BHttpRequest::ohead\_contentType [protected]

#### 5.27.4.15 `ohead_extra`

`BStringList` BHttpRequest::ohead\_extra [protected]

#### 5.27.4.16 `ohead_size`

`BUInt` BHttpRequest::ohead\_size [protected]

#### 5.27.4.17 `ohead_status`

`const char*` BHttpRequest::ohead\_status [protected]

#### 5.27.4.18 `orxFifo`

`BFifoChar` BHttpRequest::orxFifo [protected]

#### 5.27.4.19 `osocket`

`struct netconn*` BHttpRequest::osocket [protected]

#### 5.27.4.20 `osocketListen`

`struct netconn*` BHttpRequest::osocketListen [protected]

#### 5.27.4.21 otxFifo

`BFifoChar` BHttpServer::otxFifo [protected]

#### 5.27.4.22 ourl

`BString` BHttpServer::ourl [protected]

The documentation for this class was generated from the following files:

- [BHttpServer.h](#)
- [BHttpServer.cpp](#)

## 5.28 BI2c Class Reference

```
#include <BI2c.h>
```

### Public Member Functions

- [BI2c](#) ([Bool](#) useDma=0)
- [BError](#) [init](#) ([BUInt](#) bus, [Pin](#) pinClk, [Pin](#) pinData, [BUInt](#) speed=100000)
- [BError](#) [init](#) ([BUInt](#) b, [BUInt](#) speed=100000)  
*Old style init.*
- [BError](#) [writeReg8](#) ([UInt8](#) dev, [UInt8](#) address, [UInt8](#) data)
- [BError](#) [readReg8](#) ([UInt8](#) dev, [UInt8](#) address, [UInt8](#) &data)
- [BError](#) [writeAddress8](#) ([UInt8](#) dev, [UInt8](#) address, void \*data, [UInt](#) len)
- [BError](#) [readAddress8](#) ([UInt8](#) dev, [UInt8](#) address, void \*data, [UInt](#) len)
- [BError](#) [writeAddress16](#) ([UInt8](#) dev, [UInt16](#) address, void \*data, [UInt](#) len)
- [BError](#) [readAddress16](#) ([UInt8](#) dev, [UInt16](#) address, void \*data, [UInt](#) len)
- [BError](#) [write](#) ([UInt8](#) dev, void \*address, [UInt](#) addressLen, void \*data, [UInt](#) len)
- [BError](#) [read](#) ([UInt8](#) dev, void \*address, [UInt](#) addressLen, void \*data, [UInt](#) len)
- void [interrupt](#) ()
- [BError](#) [waitFor](#) ([BUInt32](#) status)
- void [printStatus](#) ()

### Protected Member Functions

- [BUInt32](#) [getStatus](#) ()
- [Bool](#) [hasStatus](#) ([BUInt32](#) status)

## Protected Attributes

- [BMutex olock](#)
- [Bool ouseDma](#)
- [I2C\\_TypeDef \\* oi2c](#)
- [UInt otimeout](#)
- [BSemaphore oint](#)
  - Interrupt.*
- [BUInt ostatus](#)
- [BDma odmaTx](#)
- [BDma odmaRx](#)

## 5.28.1 Constructor & Destructor Documentation

### 5.28.1.1 BI2c()

```
BI2c::BI2c (
    Bool useDma = 0 )
```

## 5.28.2 Member Function Documentation

### 5.28.2.1 getStatus()

```
BUInt32 BI2c::getStatus ( ) [protected]
```

### 5.28.2.2 hasStatus()

```
Bool BI2c::hasStatus (
    BUInt32 status ) [protected]
```

### 5.28.2.3 init() [1/2]

```
BError BI2c::init (
    BUInt bus,
    Pin pinClk,
    Pin pinData,
    BUInt speed = 10000 )
```

#### 5.28.2.4 `init()` [2/2]

```
BError BI2c::init (
    BUInt b,
    BUInt speed = 10000 )
```

Old style init.

#### 5.28.2.5 `interrupt()`

```
void BI2c::interrupt ( )
```

#### 5.28.2.6 `printStatus()`

```
void BI2c::printStatus ( )
```

#### 5.28.2.7 `read()`

```
BError BI2c::read (
    UInt8 dev,
    void * address,
    UInt addressLen,
    void * data,
    UInt len )
```

< While the bus is busy

#### 5.28.2.8 `readAddress16()`

```
BError BI2c::readAddress16 (
    UInt8 dev,
    UInt16 address,
    void * data,
    UInt len )
```

#### 5.28.2.9 `readAddress8()`

```
BError BI2c::readAddress8 (
    UInt8 dev,
    UInt8 address,
    void * data,
    UInt len )
```

#### 5.28.2.10 readReg8()

```
BError BI2c::readReg8 (
    UInt8 dev,
    UInt8 address,
    UInt8 & data )
```

#### 5.28.2.11 waitFor()

```
BError BI2c::waitFor (
    BUInt32 status )
```

#### 5.28.2.12 write()

```
BError BI2c::write (
    UInt8 dev,
    void * address,
    UInt addressLen,
    void * data,
    UInt len )
```

#### 5.28.2.13 writeAddress16()

```
BError BI2c::writeAddress16 (
    UInt8 dev,
    UInt16 address,
    void * data,
    UInt len )
```

#### 5.28.2.14 writeAddress8()

```
BError BI2c::writeAddress8 (
    UInt8 dev,
    UInt8 address,
    void * data,
    UInt len )
```

#### 5.28.2.15 writeReg8()

```
BError BI2c::writeReg8 (
    UInt8 dev,
    UInt8 address,
    UInt8 data )
```

### 5.28.3 Member Data Documentation

#### 5.28.3.1 odmaRx

```
BDma BI2c::odmaRx [protected]
```

#### 5.28.3.2 odmaTx

```
BDma BI2c::odmaTx [protected]
```

#### 5.28.3.3 oi2c

```
I2C_TypeDef* BI2c::oi2c [protected]
```

#### 5.28.3.4 oint

```
BSemaphore BI2c::oint [protected]
```

Interrupt.

#### 5.28.3.5 olock

```
BMutex BI2c::olock [protected]
```

## 5.28.3.6 ostatus

```
BUInt BI2c::ostatus [protected]
```

## 5.28.3.7 otimeout

```
UInt BI2c::otimeout [protected]
```

## 5.28.3.8 ouseDma

```
Bool BI2c::ouseDma [protected]
```

The documentation for this class was generated from the following files:

- [BI2c.h](#)
- [BI2c.cpp](#)

## 5.29 BInterrupt Class Reference

```
#include <BInterrupt.h>
```

### Public Types

- enum { [PriorityBase](#) = 5, [PriorityRtosMax](#) = 10 }

### Public Member Functions

- [BInterrupt](#) ()
- [BError](#) init ()  
*Initialise the interrupts.*
- void [config](#) ([BUInt](#) channel, [BUInt8](#) priority, [Bool](#) on)  
*Configure an interrupt.*
- void [extConfig](#) ([Pin](#) pin, [Bool](#) event, [Bool](#) rising, [Bool](#) falling, [Bool](#) on)  
*Configure external interrupt.*
- void [setPriority](#) ([BUInt](#) channel, [BUInt8](#) priority)  
*Sets an interrupt's priority.*
- [Bool](#) [enable](#) ([BUInt](#) channel, [Bool](#) on)  
*Enable/disable the interrupt.*
- void [print](#) ()  
*Print out enabled interrupts with priorities.*

### 5.29.1 Member Enumeration Documentation

#### 5.29.1.1 anonymous enum

```
anonymous enum
```

## Enumerator

|                 |  |
|-----------------|--|
| PriorityBase    |  |
| PriorityRtosMax |  |

## 5.29.2 Constructor & Destructor Documentation

### 5.29.2.1 BInterrupt()

```
BInterrupt::BInterrupt ( )
```

## 5.29.3 Member Function Documentation

### 5.29.3.1 config()

```
void BInterrupt::config (
    BUInt channel,
    BUInt8 priority,
    Bool on )
```

Configure an interrupt.

### 5.29.3.2 enable()

```
Bool BInterrupt::enable (
    BUInt channel,
    Bool on )
```

Enable/disable the interrupt.

### 5.29.3.3 extConfig()

```
void BInterrupt::extConfig (
    Pin pin,
    Bool event,
    Bool rising,
    Bool falling,
    Bool on )
```

Configure external interrupt.

#### 5.29.3.4 init()

```
BError BInterrupt::init ( )
```

Initialise the interrupts.

#### 5.29.3.5 print()

```
void BInterrupt::print ( )
```

Print out enabled interrupts with priorities.

#### 5.29.3.6 setPriority()

```
void BInterrupt::setPriority (
    BUInt channel,
    BUInt8 priority )
```

Sets an interrupt's priority.

The documentation for this class was generated from the following files:

- [BInterrupt.h](#)
- [BInterrupt.cpp](#)

## 5.30 Blter Class Reference

Iterator for [BList](#).

```
#include <BList.h>
```

### Public Member Functions

- [Blter](#) ([BNode](#) \*i=0)
- [operator BNode \\*](#) ()
- [int operator==](#) (const [Blter](#) &i)
- [int valid](#) ()

### Private Attributes

- [BNode](#) \* oi

### 5.30.1 Detailed Description

Iterator for [BList](#).

### 5.30.2 Constructor & Destructor Documentation

#### 5.30.2.1 BIter()

```
BIter::BIter (
    BNode * i = 0 ) [inline]
```

### 5.30.3 Member Function Documentation

#### 5.30.3.1 operator BNode \*()

```
BIter::operator BNode * ( ) [inline]
```

#### 5.30.3.2 operator==( )

```
int BIter::operator==(
    const BIter & i ) [inline]
```

#### 5.30.3.3 valid()

```
int BIter::valid ( ) [inline]
```

### 5.30.4 Member Data Documentation

#### 5.30.4.1 oi

```
BNode* BIter::oi [private]
```

The documentation for this class was generated from the following file:

- [BList.h](#)

## 5.31 BList< T > Class Template Reference

Template based list class.

```
#include <BList.h>
```

### Classes

- class [Node](#)

### Public Types

- typedef int(\* [SortFunc](#)) (T &a, T &b)  
*Prototype for sorting function.*

### Public Member Functions

- [BList](#) ()
- [BList](#) (const [BList](#)< T > &l)
- virtual [~BList](#) ()
- void [start](#) ([Blter](#) &i) const  
*Iterator to start of list.*
- [Blter begin](#) () const  
*Iterator for start of list.*
- [Blter end](#) () const  
*Iterator for end of list.*
- [Blter end](#) ([Blter](#) &i) const  
*Iterator for end of list.*
- void [next](#) ([Blter](#) &i) const  
*Iterator for next item in list.*
- void [prev](#) ([Blter](#) &i)  
*Iterator for previous item in list.*
- [Blter goTo](#) (int pos) const  
*Iterator for pos item in list.*
- int [position](#) ([Blter](#) i)  
*Postition in list item with iterator i.*
- unsigned int [number](#) () const  
*Number of items in list.*
- unsigned int [size](#) () const  
*Number of items in list.*
- int [isEnd](#) ([Blter](#) &i) const  
*True if iterator refers to last item.*
- T & [front](#) ()  
*Get first item in list.*
- T & [rear](#) ()  
*Get last item in list.*
- T & [get](#) ([Blter](#) i)  
*Get item specified by iterator in list.*
- const T & [get](#) ([Blter](#) i) const

- Get item specified by iterator in list.*
- void [append](#) (const T &item)
- Append item to list.*
- virtual void [insert](#) ([BIter](#) &i, const T &item)
- Insert item before item.*
- void [insertAfter](#) ([BIter](#) &i, const T &item)
- Insert item after item.*
- virtual void [clear](#) ()
- Clear the list.*
- virtual void [del](#) ([BIter](#) &i)
- Delete specified item.*
- void [deleteLast](#) ()
- Delete last item.*
- void [deleteFirst](#) ()
- Delete first item.*
- void [push](#) (const T &i)
- Push item onto list.*
- T [pop](#) ()
- Pop item from list deleting item.*
- void [queueAdd](#) (const T &i)
- Add item to end of list.*
- T [queueGet](#) ()
- Get item from front of list deleting item.*
- void [append](#) (const [BList](#)< T > &l)
- Append list to list.*
- int [has](#) (const T &i) const
- Checks if the item is in the list.*
- void [swap](#) ([BIter](#) i1, [BIter](#) i2)
- Swap two items in list.*
- void [sort](#) ()
- Sort list based on get(i) values.*
- void [sort](#) ([SortFunc](#) func)
- Sort list based on Sort func.*
- [BList](#)< T > & [operator=](#) (const [BList](#)< T > &l)
- T & [operator\[\]](#) (int i)
- const T & [operator\[\]](#) (int i) const
- T & [operator\[\]](#) ([BIter](#) i)
- const T & [operator\[\]](#) (const [BIter](#) &i) const
- [BList](#)< T > [operator+](#) (const [BList](#)< T > &l) const

### Protected Member Functions

- virtual [Node](#) \* [nodeGet](#) ([BIter](#) i)
- virtual const [Node](#) \* [nodeGet](#) ([BIter](#) i) const
- virtual [Node](#) \* [nodeCreate](#) (const T &item)

### Protected Attributes

- [Node](#) \* [onodes](#)
- unsigned int [olength](#)

## Private Member Functions

- virtual `Node * nodeCreate ()`

### 5.31.1 Detailed Description

```
template<class T>  
class BList< T >
```

Template based list class.

### 5.31.2 Member Typedef Documentation

#### 5.31.2.1 SortFunc

```
template<class T>  
typedef int (* BList< T >::SortFunc) (T &a, T &b)
```

Prototype for sorting function.

### 5.31.3 Constructor & Destructor Documentation

#### 5.31.3.1 BList() [1/2]

```
template<class T >  
BList< T >::BList ( )
```

#### 5.31.3.2 BList() [2/2]

```
template<class T>  
BList< T >::BList (   
    const BList< T > & l )
```

#### 5.31.3.3 ~BList()

```
template<class T >  
BList< T >::~~BList ( ) [virtual]
```

## 5.31.4 Member Function Documentation

### 5.31.4.1 `append()` [1/2]

```
template<class T>
void BList< T >::append (
    const T & item )
```

Append item to list.

### 5.31.4.2 `append()` [2/2]

```
template<class T>
void BList< T >::append (
    const BList< T > & l )
```

Append list to list.

### 5.31.4.3 `begin()`

```
template<class T >
BIter BList< T >::begin ( ) const
```

Iterator for start of list.

### 5.31.4.4 `clear()`

```
template<class T >
void BList< T >::clear ( ) [virtual]
```

Clear the list.

Reimplemented in [BDict< Type >](#), and [BDict< BString >](#).

#### 5.31.4.5 del()

```
template<class T >
void BList< T >::del (
    BIter & i ) [virtual]
```

Delete specified item.

Reimplemented in [BDict< Type >](#), and [BDict< BString >](#).

#### 5.31.4.6 deleteFirst()

```
template<class T >
void BList< T >::deleteFirst ( )
```

Delete first item.

#### 5.31.4.7 deleteLast()

```
template<class T >
void BList< T >::deleteLast ( )
```

Delete last item.

#### 5.31.4.8 end() [1/2]

```
template<class T >
BIter BList< T >::end ( ) const
```

Iterator for end of list.

#### 5.31.4.9 end() [2/2]

```
template<class T >
BIter BList< T >::end (
    BIter & i ) const
```

Iterator for end of list.

#### 5.31.4.10 front()

```
template<class T >
T & BList< T >::front ( )
```

Get first item in list.

#### 5.31.4.11 get() [1/2]

```
template<class T >
T & BList< T >::get (
    BIter i )
```

Get item specified by iterator in list.

#### 5.31.4.12 get() [2/2]

```
template<class T >
const T & BList< T >::get (
    BIter i ) const
```

Get item specified by iterator in list.

#### 5.31.4.13 goTo()

```
template<class T >
BIter BList< T >::goTo (
    int pos ) const
```

Iterator for pos item in list.

#### 5.31.4.14 has()

```
template<class T>
int BList< T >::has (
    const T & i ) const
```

Checks if the item is in the list.

#### 5.31.4.15 insert()

```
template<class T>
void BList< T >::insert (
    BIter & i,
    const T & item ) [virtual]
```

Insert item before item.

Reimplemented in [BDict< Type >](#), and [BDict< BString >](#).

#### 5.31.4.16 insertAfter()

```
template<class T>
void BList< T >::insertAfter (
    BIter & i,
    const T & item )
```

Insert item after item.

#### 5.31.4.17 isEnd()

```
template<class T >
int BList< T >::isEnd (
    BIter & i ) const
```

True if iterator refers to last item.

#### 5.31.4.18 next()

```
template<class T >
void BList< T >::next (
    BIter & i ) const
```

Iterator for next item in list.

#### 5.31.4.19 nodeCreate() [1/2]

```
template<class T>
BList< T >::Node * BList< T >::nodeCreate (
    const T & item ) [protected], [virtual]
```

**5.31.4.20 nodeCreate()** [2/2]

```
template<class T>
BList< T >::Node * BList< T >::nodeCreate ( ) [private], [virtual]
```

**5.31.4.21 nodeGet()** [1/2]

```
template<class T >
BList< T >::Node * BList< T >::nodeGet (
    BIter i ) [protected], [virtual]
```

**5.31.4.22 nodeGet()** [2/2]

```
template<class T >
const BList< T >::Node * BList< T >::nodeGet (
    BIter i ) const [protected], [virtual]
```

**5.31.4.23 number()**

```
template<class T >
unsigned int BList< T >::number ( ) const
```

Number of items in list.

**5.31.4.24 operator+()**

```
template<class T>
BList< T > BList< T >::operator+ (
    const BList< T > & l ) const
```

**5.31.4.25 operator=()**

```
template<class T>
BList< T > & BList< T >::operator= (
    const BList< T > & l )
```

**5.31.4.26 operator[]()** [1/4]

```
template<class T >
T & BList< T >::operator[] (
    int i )
```

**5.31.4.27 operator[]()** [2/4]

```
template<class T >
const T & BList< T >::operator[] (
    int i ) const
```

**5.31.4.28 operator[]()** [3/4]

```
template<class T >
T & BList< T >::operator[] (
    BIter i )
```

**5.31.4.29 operator[]()** [4/4]

```
template<class T >
const T & BList< T >::operator[] (
    const BIter & i ) const
```

**5.31.4.30 pop()**

```
template<class T >
T BList< T >::pop ( )
```

Pop item from list deleting item.

**5.31.4.31 position()**

```
template<class T >
int BList< T >::position (
    BIter i )
```

Position in list item with iterator i.

#### 5.31.4.32 prev()

```
template<class T >
void BList< T >::prev (
    BIter & i )
```

Iterator for previous item in list.

#### 5.31.4.33 push()

```
template<class T>
void BList< T >::push (
    const T & i )
```

Push item onto list.

#### 5.31.4.34 queueAdd()

```
template<class T>
void BList< T >::queueAdd (
    const T & i )
```

Add item to end of list.

#### 5.31.4.35 queueGet()

```
template<class T >
T BList< T >::queueGet ( )
```

Get item from front of list deleting item.

#### 5.31.4.36 rear()

```
template<class T >
T & BList< T >::rear ( )
```

Get last item in list.

#### 5.31.4.37 size()

```
template<class T >
unsigned int BList< T >::size ( ) const
```

Number of items in list.

#### 5.31.4.38 sort() [1/2]

```
template<class T >
void BList< T >::sort ( )
```

Sort list based on get(i) values.

#### 5.31.4.39 sort() [2/2]

```
template<class T >
void BList< T >::sort (
    SortFunc func )
```

Sort list based on Sort func.

#### 5.31.4.40 start()

```
template<class T >
void BList< T >::start (
    BIter & i ) const
```

Iterator to start of list.

#### 5.31.4.41 swap()

```
template<class T >
void BList< T >::swap (
    BIter i1,
    BIter i2 )
```

Swap two items in list.

### 5.31.5 Member Data Documentation

### 5.31.5.1 olength

```
template<class T>
unsigned int BList< T >::olength [protected]
```

### 5.31.5.2 onodes

```
template<class T>
Node* BList< T >::onodes [protected]
```

The documentation for this class was generated from the following files:

- [BList.h](#)
- [BList\\_func.h](#)

## 5.32 BMutex Class Reference

Mutex class.

```
#include <BMutex.h>
```

### Public Member Functions

- [BMutex](#) ()
- [BMutex](#) (const [BMutex](#) &mutex)
- [~BMutex](#) ()
- void [lock](#) ()  
*Set lock, wait as necessary.*
- void [unlock](#) ()  
*Unlock the lock.*
- int [tryLock](#) ()  
*Test the lock.*
- [BMutex](#) & [operator=](#) (const [BMutex](#) &mutex)

### Private Attributes

- SemaphoreHandle\_t [omutex](#)

### 5.32.1 Detailed Description

Mutex class.

## 5.32.2 Constructor & Destructor Documentation

### 5.32.2.1 BMutex() [1/2]

```
BMutex::BMutex ( )
```

### 5.32.2.2 BMutex() [2/2]

```
BMutex::BMutex (
    const BMutex & mutex )
```

### 5.32.2.3 ~BMutex()

```
BMutex::~~BMutex ( )
```

## 5.32.3 Member Function Documentation

### 5.32.3.1 lock()

```
void BMutex::lock ( )
```

Set lock, wait as necessary.

### 5.32.3.2 operator=()

```
BMutex & BMutex::operator= (
    const BMutex & mutex )
```

### 5.32.3.3 tryLock()

```
int BMutex::tryLock ( )
```

Test the lock.

#### 5.32.3.4 unlock()

```
void BMutex::unlock ( )
```

Unlock the lock.

### 5.32.4 Member Data Documentation

#### 5.32.4.1 omutex

```
SemaphoreHandle_t BMutex::omutex [private]
```

The documentation for this class was generated from the following files:

- [BMutex.h](#)
- [BMutex.cpp](#)

## 5.33 BMutexLock Class Reference

```
#include <BMutex.h>
```

### Public Member Functions

- [BMutexLock](#) ([BMutex](#) &[lock](#), int doLock=0)
- [~BMutexLock](#) ()
- void [lock](#) ()
- void [unlock](#) ()

### Private Attributes

- [BMutex](#) & [olock](#)

### 5.33.1 Constructor & Destructor Documentation

#### 5.33.1.1 BMutexLock()

```
BMutexLock::BMutexLock (
    BMutex & lock,
    int doLock = 0 ) [inline]
```

### 5.33.1.2 ~BMutexLock()

```
BMutexLock::~BMutexLock ( ) [inline]
```

## 5.33.2 Member Function Documentation

### 5.33.2.1 lock()

```
void BMutexLock::lock ( ) [inline]
```

### 5.33.2.2 unlock()

```
void BMutexLock::unlock ( ) [inline]
```

## 5.33.3 Member Data Documentation

### 5.33.3.1 olock

```
BMutex& BMutexLock::olock [private]
```

The documentation for this class was generated from the following file:

- [BMutex.h](#)

## 5.34 BNameValue< T > Class Template Reference

```
#include <BNameValue.h>
```

### Public Member Functions

- [BNameValue](#) ( )
- [BNameValue](#) (BString name, const T &value)
- [BString](#) [getName](#) ( )
- T & [getValue](#) ( )

## Private Attributes

- [BString oname](#)
- [T ovalue](#)

## 5.34.1 Constructor & Destructor Documentation

### 5.34.1.1 BNameValue() [1/2]

```
template<class T >  
BNameValue< T >::BNameValue ( ) [inline]
```

### 5.34.1.2 BNameValue() [2/2]

```
template<class T >  
BNameValue< T >::BNameValue (   
    BString name,  
    const T & value ) [inline]
```

## 5.34.2 Member Function Documentation

### 5.34.2.1 getName()

```
template<class T >  
BString BNameValue< T >::getName ( ) [inline]
```

### 5.34.2.2 getValue()

```
template<class T >  
T& BNameValue< T >::getValue ( ) [inline]
```

## 5.34.3 Member Data Documentation

## 5.34.3.1 oname

```
template<class T >
BString BNameValue< T >::oname [private]
```

## 5.34.3.2 ovalue

```
template<class T >
T BNameValue< T >::ovalue [private]
```

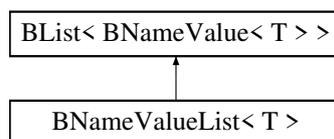
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

## 5.35 BNameValueList&lt; T &gt; Class Template Reference

```
#include <BNameValue.h>
```

Inheritance diagram for BNameValueList< T >:



## Public Member Functions

- T \* [find](#) (BString name)
- [Blter findPos](#) (BString name)

## Additional Inherited Members

## 5.35.1 Member Function Documentation

## 5.35.1.1 find()

```
template<class T >
T* BNameValueList< T >::find (
    BString name ) [inline]
```

### 5.35.1.2 findPos()

```
template<class T >
BIter BNameValueList< T >::findPos (
    BString name ) [inline]
```

The documentation for this class was generated from the following file:

- [BNameValue.h](#)

## 5.36 BNetwork Class Reference

```
#include <BNetwork.h>
```

### Public Types

- enum [PhyType](#) { [PhyTypeDP83848](#), [PhyTypeLAN8742A](#) }

### Public Member Functions

- [BNetwork](#) ()
- virtual [~BNetwork](#) ()
- [BError](#) [init](#) (const [BUInt8](#) \*macAddress, [BUInt8](#) phyNum, [PhyType](#) phyType, const [Pin](#) \*pins=0)  
*Initialise the Ethernet.*
- [BError](#) [start](#) ([Bool](#) dhcp, [IpAddress](#) ipAddress=0, [IpAddress](#) netmask=0, [IpAddress](#) gateway=0)  
*Configure the Ethernet.*
- [BError](#) [config](#) ([Bool](#) dhcp, [IpAddress](#) ipAddress=0, [IpAddress](#) netmask=0, [IpAddress](#) gateway=0)  
*Configure the Ethernet.*
- void [getMac](#) ([BUInt8](#) \*mac)
- void [getIpAddress](#) ([IpAddress](#) &ipAddress)
- void [getStats](#) ([BNetworkStats](#) &stats)
- void [interrupt](#) (int event)
- void [process](#) ()  
*Processes RX packets and other events.*
- void [process\\_dhcp](#) ()  
*Process DHCP.*
- void [linkChanged](#) ()  
*The Ethernet link has changed.*
- [err\\_t](#) [etherInit](#) ()  
*Initialise low level ethernet.*
- [err\\_t](#) [etherPacketSend](#) ([IpPacket](#) \*packet)  
*Send an Ethernet packet.*
- [Bool](#) [etherPacketRecv](#) ()  
*Process an Ethernet input packet.*
- void [phyInit](#) ()  
*Initialise PHY information.*
- [BError](#) [phySet](#) ([Bool](#) autoNeg, [Bool](#) fullDuplex, [BUInt32](#) speed)  
*Set phy mode.*

- void [phySetup](#) ()  
*Setup PHY.*
- void [phyProcess](#) ()  
*Process PHY changes on interrupt from PHY.*
- void [phyUpdate](#) ()  
*Update MAC information from PHY.*
- void [phyPrint](#) ()  
*Print PHY settings.*

### Public Attributes

- [ETH\\_HandleTypeDef](#) [oether](#)  
*The ethernet low level driver.*

### Protected Attributes

- [BUInt8](#) [omacAddress](#) [6]  
*The hardware MAC address.*
- [BUInt8](#) [ophyNum](#)  
*The PHY number.*
- [PhyType](#) [ophyType](#)  
*The PHY type.*
- [Bool](#) [ophyAutoNeg](#)  
*PHY auto negotiation.*
- [Bool](#) [ophyFullDuplex](#)  
*PHY full duplex.*
- [BUInt32](#) [ophySpeed](#)  
*PHY speed.*
- [ip\\_addr\\_t](#) [oipAddress](#)
- [ip\\_addr\\_t](#) [onetmask](#)
- [ip\\_addr\\_t](#) [ogateway](#)
- [struct netif](#) [onetif](#)  
*The LwIP network interface.*
- [Bool](#) [odhcp](#)  
*DHCP enabled.*
- [DhcpState](#) [odhcpState](#)  
*DHCP state.*
- [BNetworkProcessTask](#) [oprocessTask](#)  
*The Ethernet processing task.*
- [BNetworkDhcpTask](#) [odhcpTask](#)  
*The Dhcp processing task.*
- [BQueue](#)< [BUInt](#) > [oeventQueue](#)  
*Event processing queue.*
- [BNetworkStats](#) [ostatats](#)  
*Statistics.*

## 5.36.1 Member Enumeration Documentation

### 5.36.1.1 PhyType

enum [BNetwork::PhyType](#)

## Enumerator

|                 |  |
|-----------------|--|
| PhyTypeDP83848  |  |
| PhyTypeLAN8742A |  |

## 5.36.2 Constructor & Destructor Documentation

### 5.36.2.1 BNetwork()

```
BNetwork::BNetwork ( )
```

### 5.36.2.2 ~BNetwork()

```
BNetwork::~~BNetwork ( ) [virtual]
```

## 5.36.3 Member Function Documentation

### 5.36.3.1 config()

```
BError BNetwork::config (
    Bool dhcp,
    IPAddress ipAddress = 0,
    IPAddress netmask = 0,
    IPAddress gateway = 0 )
```

Configure the Ethernet.

### 5.36.3.2 etherInit()

```
err_t BNetwork::etherInit ( )
```

Initialise low level ethernet.

### 5.36.3.3 etherPacketRecv()

```
Bool BNetwork::etherPacketRecv ( )
```

Process an Ethernet input packet.

### 5.36.3.4 etherPacketSend()

```
err_t BNetwork::etherPacketSend (
    IpPacket * packet )
```

Send an Ethernet packet.

### 5.36.3.5 getIpAddress()

```
void BNetwork::getIpAddress (
    IpAddress & ipAddress )
```

### 5.36.3.6 getMac()

```
void BNetwork::getMac (
    BUInt8 * mac )
```

### 5.36.3.7 getStats()

```
void BNetwork::getStats (
    BNetworkStats & stats )
```

### 5.36.3.8 init()

```
BError BNetwork::init (
    const BUInt8 * macAddress,
    BUInt8 phyNum,
    PhyType phyType,
    const Pin * pins = 0 )
```

Initialise the Ethernet.

[BNetwork::init](#) pins define CPU pins used.

#### 5.36.3.9 interrupt()

```
void BNetwork::interrupt (
    int event )
```

#### 5.36.3.10 linkChanged()

```
void BNetwork::linkChanged ( )
```

The Ethernet link has changed.

#### 5.36.3.11 phyInit()

```
void BNetwork::phyInit ( )
```

Initialise PHY information.

#### 5.36.3.12 phyPrint()

```
void BNetwork::phyPrint ( )
```

Print PHY settings.

#### 5.36.3.13 phyProcess()

```
void BNetwork::phyProcess ( )
```

Process PHY changes on interrupt from PHY.

Process PHY changes on interrupt from PHY or poll.

#### 5.36.3.14 phySet()

```
BError BNetwork::phySet (
    Bool autoNeg,
    Bool fullDuplex,
    BUInt32 speed )
```

Set phy mode.

#### 5.36.3.15 phySetup()

```
void BNetwork::phySetup ( )
```

Setup PHY.

#### 5.36.3.16 phyUpdate()

```
void BNetwork::phyUpdate ( )
```

Update MAC information from PHY.

#### 5.36.3.17 process()

```
void BNetwork::process ( )
```

Processes RX packets and other events.

#### 5.36.3.18 process\_dhcp()

```
void BNetwork::process_dhcp ( )
```

Process DHCP.

#### 5.36.3.19 start()

```
BError BNetwork::start (
    Bool dhcp,
    IPAddress ipAddress = 0,
    IPAddress netmask = 0,
    IPAddress gateway = 0 )
```

Configure the Ethernet.

### 5.36.4 Member Data Documentation

#### 5.36.4.1 odhcp

`Bool BNetwork::odhcp` [protected]

DHCP enabled.

#### 5.36.4.2 odhcpState

`DhcpState BNetwork::odhcpState` [protected]

DHCP state.

#### 5.36.4.3 odhcpTask

`BNetworkDhcpTask BNetwork::odhcpTask` [protected]

The Dhcp processing task.

#### 5.36.4.4 oether

`ETH_HandleTypeDef BNetwork::oether`

The ethernet low level driver.

#### 5.36.4.5 oeventQueue

`BQueue<BUInt> BNetwork::oeventQueue` [protected]

Event processing queue.

#### 5.36.4.6 ogateway

`ip_addr_t BNetwork::ogateway` [protected]

#### 5.36.4.7 oipAddress

```
ip_addr_t BNetwork::oipAddress [protected]
```

#### 5.36.4.8 omacAddress

```
BUInt8 BNetwork::omacAddress[6] [protected]
```

The hardware MAC address.

#### 5.36.4.9 onetif

```
struct netif BNetwork::onetif [protected]
```

The LwIP network interface.

#### 5.36.4.10 onetmask

```
ip_addr_t BNetwork::onetmask [protected]
```

#### 5.36.4.11 ophyAutoNeg

```
Bool BNetwork::ophyAutoNeg [protected]
```

PHY auto negotiation.

#### 5.36.4.12 ophyFullDuplex

```
Bool BNetwork::ophyFullDuplex [protected]
```

PHY full duplex.

#### 5.36.4.13 ophyNum

`BUInt8 BNetwork::ophyNum` [protected]

The PHY number.

#### 5.36.4.14 ophySpeed

`BUInt32 BNetwork::ophySpeed` [protected]

PHY speed.

#### 5.36.4.15 ophyType

`PhyType BNetwork::ophyType` [protected]

The PHY type.

#### 5.36.4.16 oprocessTask

`BNetworkProcessTask BNetwork::oprocessTask` [protected]

The Ethernet processing task.

#### 5.36.4.17 ostats

`BNetworkStats BNetwork::ostats` [protected]

Statistics.

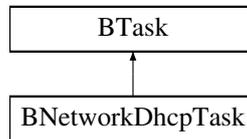
The documentation for this class was generated from the following files:

- [BNetwork.h](#)
- [BNetwork.cpp](#)

## 5.37 BNetworkDhcpTask Class Reference

```
#include <BNetwork.h>
```

Inheritance diagram for BNetworkDhcpTask:



### Public Member Functions

- [BNetworkDhcpTask \(BNetwork &BNetwork\)](#)
- void [run \(\)](#)

*The tasks main run loop.*

### Private Attributes

- [BNetwork & onetwork](#)

### Additional Inherited Members

#### 5.37.1 Constructor & Destructor Documentation

##### 5.37.1.1 BNetworkDhcpTask()

```
BNetworkDhcpTask::BNetworkDhcpTask (  
    BNetwork & BNetwork )
```

#### 5.37.2 Member Function Documentation

##### 5.37.2.1 run()

```
void BNetworkDhcpTask::run ( ) [virtual]
```

The tasks main run loop.

Reimplemented from [BTask](#).

### 5.37.3 Member Data Documentation

#### 5.37.3.1 onetwork

```
BNetwork& BNetworkDhcpTask::onetwork [private]
```

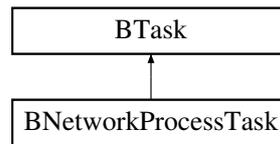
The documentation for this class was generated from the following files:

- [BNetwork.h](#)
- [BNetwork.cpp](#)

## 5.38 BNetworkProcessTask Class Reference

```
#include <BNetwork.h>
```

Inheritance diagram for BNetworkProcessTask:



### Public Member Functions

- [BNetworkProcessTask \(BNetwork &BNetwork\)](#)
- void [run \(\)](#)

*The tasks main run loop.*

### Private Attributes

- [BNetwork](#) & [onetwork](#)

### Additional Inherited Members

#### 5.38.1 Constructor & Destructor Documentation

##### 5.38.1.1 BNetworkProcessTask()

```
BNetworkProcessTask::BNetworkProcessTask (
    BNetwork & BNetwork )
```

## 5.38.2 Member Function Documentation

### 5.38.2.1 run()

```
void BNetworkProcessTask::run ( ) [virtual]
```

The tasks main run loop.

Reimplemented from [BTask](#).

## 5.38.3 Member Data Documentation

### 5.38.3.1 onetwork

```
BNetwork& BNetworkProcessTask::onetwork [private]
```

The documentation for this class was generated from the following files:

- [BNetwork.h](#)
- [BNetwork.cpp](#)

## 5.39 BNetworkStats Class Reference

```
#include <BNetwork.h>
```

### Public Member Functions

- [BNetworkStats](#) ()

### Public Attributes

- [BUInt32 txPackets](#)  
*Number of TX packets.*
- [BUInt32 txDroppedPackets](#)  
*Number of dropped TX packets.*
- [BUInt32 rxPackets](#)  
*Number of RX packets.*
- [BUInt32 rxDroppedPackets](#)  
*Number of dropped RX packets.*

## 5.39.1 Constructor & Destructor Documentation

### 5.39.1.1 BNetworkStats()

```
BNetworkStats::BNetworkStats ( ) [inline]
```

## 5.39.2 Member Data Documentation

### 5.39.2.1 rxDroppedPackets

```
BUInt32 BNetworkStats::rxDroppedPackets
```

Number of dropped RX packets.

### 5.39.2.2 rxPackets

```
BUInt32 BNetworkStats::rxPackets
```

Number of RX packets.

### 5.39.2.3 txDroppedPackets

```
BUInt32 BNetworkStats::txDroppedPackets
```

Number of dropped TX packets.

### 5.39.2.4 txPackets

```
BUInt32 BNetworkStats::txPackets
```

Number of TX packets.

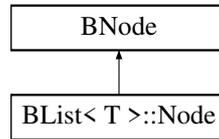
The documentation for this class was generated from the following file:

- [BNetwork.h](#)

## 5.40 BNode Class Reference

```
#include <BList.h>
```

Inheritance diagram for BNode:



### Public Member Functions

- [BNode\(\)](#)

### Public Attributes

- [BNode \\* next](#)
- [BNode \\* prev](#)

### 5.40.1 Constructor & Destructor Documentation

#### 5.40.1.1 BNode()

```
BNode::BNode ( ) [inline]
```

### 5.40.2 Member Data Documentation

#### 5.40.2.1 next

```
BNode\* BNode::next
```

#### 5.40.2.2 prev

```
BNode\* BNode::prev
```

The documentation for this class was generated from the following file:

- [BList.h](#)

## 5.41 BNtpClient Class Reference

```
#include <BNtpClient.h>
```

### Public Member Functions

- [BNtpClient](#) ()
- [BError](#) [init](#) ()
- void [run](#) ([Bool](#) run)
- [BError](#) [setServerAddress](#) ([Bool](#) dhcp, [IpAddress](#) timeServer)
- virtual void [setSystemTime](#) (u32\_t sec, u32\_t usec)

### 5.41.1 Constructor & Destructor Documentation

#### 5.41.1.1 BNtpClient()

```
BNtpClient::BNtpClient ( )
```

### 5.41.2 Member Function Documentation

#### 5.41.2.1 init()

```
BError BNtpClient::init ( )
```

#### 5.41.2.2 run()

```
void BNtpClient::run (
    Bool run )
```

#### 5.41.2.3 setServerAddress()

```
BError BNtpClient::setServerAddress (
    Bool dhcp,
    IpAddress timeServer )
```

## 5.41.2.4 setSystemTime()

```
void BNtpClient::setSystemTime (
    u32_t sec,
    u32_t usec ) [virtual]
```

The documentation for this class was generated from the following files:

- [BNtpClient.h](#)
- [BNtpClient.cpp](#)

## 5.42 BoapMc1Comms Class Reference

```
#include <BoapMc1.h>
```

## Public Member Functions

- [BoapMc1Comms](#) ([Bool](#) threaded=0, [BUInt](#) reqSize=512)
- virtual [~BoapMc1Comms](#) ()
- void [setCommsMode](#) ([Bool](#) halfDuplex)
  - Sets half duplex mode.*
- void [setComms](#) ([BComms](#) &comms)
  - Sets the communications interface to use.*
- void [setComms](#) ([BComms](#) \*comms)
  - Sets the communications interface to use.*
- void [setAddress](#) ([BUInt16](#) addressTo, [BUInt16](#) addressFrom)
  - Sets the to and from addresses.*
- [BUInt32](#) [getApiVersion](#) ()
  - Returns the API version.*
- [BUInt32](#) [setTimeout](#) ([BUInt32](#) timeoutUs)
  - Sets the call timeout returning the current value.*
- [BoapMc1Packet](#) \* [packetRx](#) ()
  - returns a reference to the current RX packet*
- virtual [BError](#) [processRx](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Process any RX packets queuing them as needed.*

## Protected Member Functions

- virtual [BError](#) [processRequests](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Check and process any requests.*
- virtual [BError](#) [processRequest](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Check and process any request.*
- [BError](#) [packetTx](#) ([BDataChunk](#) \*chunks, [BUInt](#) nChunks, [BUInt16](#) waitCmdReply)
- [BError](#) [packetRxData](#) (void \*data, [BUInt](#) nBytes)
- [BError](#) [packetRxEnd](#) ()

## Protected Attributes

- [Bool othreaded](#)  
*Threaded operation.*
- [BUInt32 oreqSize](#)  
*The maximum request size.*
- [BMutex olockCall](#)  
*Lock for RPC calls. Only one at a time.*
- [BMutex olockTx](#)  
*Lock for TX.*
- [BComms \\* ocomms](#)
- [BUInt32 oapiVersion](#)
- [Bool ohalfDuplex](#)  
*Half duplex mode.*
- [BUInt32 otimeout](#)  
*The timeout in us for calls.*
- [BUInt16 oaddressTo](#)
- [BUInt16 oaddressFrom](#)
- [BoapMc1Packet opacketRxBase](#)
- [BoapMc1Packet \\* opacketRx](#)  
*The RX packet.*
- [BoapMc1Packet opacketTxBase](#)
- [BoapMc1Packet \\* opacketTx](#)  
*The TX packet.*
- [BUInt opacketRpcCmd](#)  
*Waiting for RPC reply to cmd.*
- [BSemaphore opacketRpcSema](#)  
*Wait RPC reply semaphore.*
- [BSemaphore opacketRpcDoneSema](#)  
*Wait RPC complete semaphore.*
- [BoapMc1Error oerror](#)  
*The call return error;.*

## 5.42.1 Constructor & Destructor Documentation

### 5.42.1.1 BoapMc1Comms()

```
BoapMc1Comms::BoapMc1Comms (
    Bool threaded = 0,
    BUInt reqSize = 512 )
```

### 5.42.1.2 ~BoapMc1Comms()

```
BoapMc1Comms::~~BoapMc1Comms ( ) [virtual]
```

## 5.42.2 Member Function Documentation

### 5.42.2.1 getApiVersion()

```
BUInt32 BoapMc1Comms::getApiVersion ( )
```

Returns the API version.

### 5.42.2.2 packetRx()

```
BoapMc1Packet * BoapMc1Comms::packetRx ( )
```

returns a reference to the current RX packet

### 5.42.2.3 packetRxData()

```
BError BoapMc1Comms::packetRxData (
    void * data,
    BUInt nBytes ) [protected]
```

### 5.42.2.4 packetRxEnd()

```
BError BoapMc1Comms::packetRxEnd ( ) [protected]
```

### 5.42.2.5 packetTx()

```
BError BoapMc1Comms::packetTx (
    BDataChunk * chunks,
    BUInt nChunks,
    BUInt16 waitCmdReply ) [protected]
```

### 5.42.2.6 processRequest()

```
BError BoapMc1Comms::processRequest (
    BTimeout timeoutUs = BTimeoutForever ) [protected], [virtual]
```

Check and process any request.

#### 5.42.2.7 processRequests()

```
BError BoapMclComms::processRequests (  
    BTimeout timeoutUs = BTimeoutForever ) [protected], [virtual]
```

Check and process any requests.

#### 5.42.2.8 processRx()

```
BError BoapMclComms::processRx (  
    BTimeout timeoutUs = BTimeoutForever ) [virtual]
```

Process any RX packets queuing them as needed.

#### 5.42.2.9 setAddress()

```
void BoapMclComms::setAddress (  
    BUInt16 addressTo,  
    BUInt16 addressFrom )
```

Sets the to and from addresses.

#### 5.42.2.10 setComms() [1/2]

```
void BoapMclComms::setComms (  
    BComms & comms )
```

Sets the communications interface to use.

#### 5.42.2.11 setComms() [2/2]

```
void BoapMclComms::setComms (  
    BComms * comms )
```

Sets the communications interface to use.

#### 5.42.2.12 setCommsMode()

```
void BoapMc1Comms::setCommsMode (
    Bool halfDuplex )
```

Sets half duplex mode.

#### 5.42.2.13 setTimeout()

```
BUInt32 BoapMc1Comms::setTimeout (
    BUInt32 timeoutUs )
```

Sets the call timeout returning the current value.

### 5.42.3 Member Data Documentation

#### 5.42.3.1 oaddressFrom

```
BUInt16 BoapMc1Comms::oaddressFrom [protected]
```

#### 5.42.3.2 oaddressTo

```
BUInt16 BoapMc1Comms::oaddressTo [protected]
```

#### 5.42.3.3 oapiVersion

```
BUInt32 BoapMc1Comms::oapiVersion [protected]
```

#### 5.42.3.4 ocomms

```
BComms* BoapMc1Comms::ocomms [protected]
```

#### 5.42.3.5 oerror

`BoapMclError` `BoapMclComms::oerror` [protected]

The call return error;.

#### 5.42.3.6 ohalfDuplex

`Bool` `BoapMclComms::ohalfDuplex` [protected]

Half duplex mode.

#### 5.42.3.7 olockCall

`BMutex` `BoapMclComms::olockCall` [protected]

Lock for RPC calls. Only one at a time.

#### 5.42.3.8 olockTx

`BMutex` `BoapMclComms::olockTx` [protected]

Lock for TX.

#### 5.42.3.9 opacketRpcCmd

`BUInt` `BoapMclComms::opacketRpcCmd` [protected]

Waiting for RPC reply to cmd.

#### 5.42.3.10 opacketRpcDoneSema

`BSemaphore` `BoapMclComms::opacketRpcDoneSema` [protected]

Wait RPC complete semaphore.

#### 5.42.3.11 opacketRpcSema

[BSemaphore](#) BoapMc1Comms::opacketRpcSema [protected]

Wait RPC reply semaphore.

#### 5.42.3.12 opacketRx

[BoapMc1Packet\\*](#) BoapMc1Comms::opacketRx [protected]

The RX packet.

#### 5.42.3.13 opacketRxBase

[BoapMc1Packet](#) BoapMc1Comms::opacketRxBase [protected]

#### 5.42.3.14 opacketTx

[BoapMc1Packet\\*](#) BoapMc1Comms::opacketTx [protected]

The TX packet.

#### 5.42.3.15 opacketTxBase

[BoapMc1Packet](#) BoapMc1Comms::opacketTxBase [protected]

#### 5.42.3.16 oreqSize

[BUInt32](#) BoapMc1Comms::oreqSize [protected]

The maximum request size.

#### 5.42.3.17 othreaded

`Bool` BoapMc1Comms::othreaded [protected]

Threaded operation.

#### 5.42.3.18 otimeout

`BUInt32` BoapMc1Comms::otimeout [protected]

The timeout in us for calls.

The documentation for this class was generated from the following files:

- [BoapMc1.h](#)
- [BoapMc1.cpp](#)

## 5.43 BoapMc1Error Struct Reference

```
#include <BoapMc1.h>
```

### Public Attributes

- `BInt16` number  
*The error number.*
- char `string` [32]  
*The error string.*

### 5.43.1 Member Data Documentation

#### 5.43.1.1 number

`BInt16` BoapMc1Error::number

The error number.

### 5.43.1.2 string

```
char BoapMc1Error::string[32]
```

The error string.

The documentation for this struct was generated from the following file:

- [BoapMc1.h](#)

## 5.44 BoapMc1Packet Class Reference

```
#include <BoapMc1.h>
```

### Public Attributes

- [BoapMc1PacketHead](#) head
- char [data](#) [8]

### 5.44.1 Member Data Documentation

#### 5.44.1.1 data

```
char BoapMc1Packet::data[8]
```

#### 5.44.1.2 head

```
BoapMc1PacketHead BoapMc1Packet::head
```

The documentation for this class was generated from the following file:

- [BoapMc1.h](#)

## 5.45 BoapMc1PacketHead Struct Reference

```
#include <BoapMc1.h>
```

## Public Attributes

- BUInt16 magic
- BUInt16 length
- BUInt16 addressTo

*Do we want addresses ?*

- BUInt16 addressFrom
- BUInt16 cmd
- BInt16 error
- BUInt32 checksum

*Not used. Maybe return a [BError](#) object ?*

*Not used. Not sure we want this here or at the very end of the packet.*

### 5.45.1 Member Data Documentation

#### 5.45.1.1 addressFrom

[BUInt16](#) BoapMc1PacketHead::addressFrom

#### 5.45.1.2 addressTo

[BUInt16](#) BoapMc1PacketHead::addressTo

Do we want addresses ?

#### 5.45.1.3 checksum

[BUInt32](#) BoapMc1PacketHead::checksum

Not used. Not sure we want this here or at the very end of the packet.

#### 5.45.1.4 cmd

[BUInt16](#) BoapMc1PacketHead::cmd

#### 5.45.1.5 error

`BInt16 BoapMc1PacketHead::error`

Not used. Maybe return a `BError` object ?

#### 5.45.1.6 length

`BUInt16 BoapMc1PacketHead::length`

#### 5.45.1.7 magic

`BUInt16 BoapMc1PacketHead::magic`

The documentation for this struct was generated from the following file:

- [BoapMc1.h](#)

## 5.46 BoapMcClientObject Class Reference

```
#include <BoapMc.h>
```

### Public Member Functions

- [BoapMcClientObject](#) (`BComms &comms`)
- virtual `~BoapMcClientObject` ()
- void [setAddress](#) (`BUInt8 addressTo`, `BUInt8 addressFrom`)
- `BUInt32` [getApiVersion](#) ()  
*Returns the API version.*

### Protected Member Functions

- `BError` [performCall](#) ()  
*Performs a RPC call to the named service.*
- `BError` [performSend](#) ()  
*Performs a send to the named service.*
- `BError` [performRecv](#) ()  
*Performs a receive.*

## Protected Attributes

- [BUInt32](#) `oapiVersion`
- [BComms](#) & `ocomms`
- [BUInt8](#) `oaddressTo`
- [BUInt8](#) `oaddressFrom`
- [BoapMcPacket](#) `opacket`

## 5.46.1 Constructor & Destructor Documentation

### 5.46.1.1 BoapMcClientObject()

```
BoapMcClientObject::BoapMcClientObject (
    BComms & comms )
```

### 5.46.1.2 ~BoapMcClientObject()

```
BoapMcClientObject::~~BoapMcClientObject ( ) [virtual]
```

## 5.46.2 Member Function Documentation

### 5.46.2.1 getApiVersion()

```
BUInt32 BoapMcClientObject::getApiVersion ( )
```

Returns the API version.

### 5.46.2.2 performCall()

```
BError BoapMcClientObject::performCall ( ) [protected]
```

Performs a RPC call to the named service.

### 5.46.2.3 performRecv()

`BError` BoapMcClientObject::performRecv ( ) [protected]

Performs a receive.

### 5.46.2.4 performSend()

`BError` BoapMcClientObject::performSend ( ) [protected]

Performs a send to the named service.

### 5.46.2.5 setAddress()

```
void BoapMcClientObject::setAddress (
    BUInt8 addressTo,
    BUInt8 addressFrom )
```

## 5.46.3 Member Data Documentation

### 5.46.3.1 oaddressFrom

`BUInt8` BoapMcClientObject::oaddressFrom [protected]

### 5.46.3.2 oaddressTo

`BUInt8` BoapMcClientObject::oaddressTo [protected]

### 5.46.3.3 oapiVersion

`BUInt32` BoapMcClientObject::oapiVersion [protected]

#### 5.46.3.4 ocomms

[BComms](#)& BoapMcClientObject::ocomms [protected]

#### 5.46.3.5 opacket

[BoapMcPacket](#) BoapMcClientObject::opacket [protected]

The documentation for this class was generated from the following files:

- [BoapMc.h](#)
- [BoapMc.cpp](#)

## 5.47 BoapMcComms Class Reference

```
#include <BoapMc.h>
```

### Public Member Functions

- [BoapMcComms](#) ([Bool](#) threaded=0, [BUInt](#) rxQueueSize=4)
- virtual [~BoapMcComms](#) ()
- void [setCommsMode](#) ([Bool](#) slave, [BUInt](#) txQueueSize)
  - Sets slave mode.*
- void [setComms](#) ([BComms](#) &comms)
  - Sets the communications interface to use.*
- void [setComms](#) ([BComms](#) \*comms)
  - Sets the communications interface to use.*
- void [setAddress](#) ([BUInt8](#) addressTo, [BUInt8](#) addressFrom)
  - Sets the to and from addresses.*
- [BUInt32](#) [getApiVersion](#) ()
  - Returns the API version.*
- [BUInt32](#) [setTimeout](#) ([BUInt32](#) timeoutUs)
  - Sets the call timeout returning the current value.*
- virtual [BError](#) [processRx](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Process any RX packets queuing them as needed.*
- virtual [BError](#) [processRequests](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Check and process all requests.*
- virtual [BError](#) [processRequest](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Check and process any request.*
- virtual [BError](#) [processPacket](#) ([BoapMcPacket](#) &rx, [BoapMcPacket](#) &tx)
  - Process a recieved packet.*

## Protected Member Functions

- [BError performCall \(\)](#)  
*Performs a RPC call to the remote side.*
- [BError performSend \(\)](#)  
*Performs a RPC send to the remote side.*
- [BError packetSend \(BoapMcPacket &packet\)](#)  
*Receives a packet.*
- [BError packetRecv \(BoapMcPacket &packet\)](#)  
*Receives a packet.*

## Protected Attributes

- [Bool othreaded](#)
- [BMutex olockCall](#)  
*Lock for RPC calls. Only one at a time.*
- [BMutex olockTx](#)  
*Lock for TX.*
- [BComms \\* ocomms](#)
- [BUInt32 oapiVersion](#)
- [Bool oslave](#)  
*Set slave mode.*
- [BUInt32 otimeout](#)  
*The timeout in us for calls.*
- [BUInt8 oaddressTo](#)
- [BUInt8 oaddressFrom](#)
- [BoapMcPacket opacket](#)  
*Packet RX buffer.*
- [BoapMcPacket opacketTx](#)  
*Packet TX buffer for calls.*
- [BoapMcPacket opacketRx](#)  
*Packet RX buffer for calls.*
- [BSemaphore opacketRxSema](#)  
*Wait RX semaphore.*
- [BoapMcPacket opacketReqTx](#)  
*Packet TX buffer for requests.*
- [BoapMcPacket opacketReqRx](#)  
*Packet RX buffer for requests.*
- [BQueue< BoapMcPacket > opacketReqQueue](#)  
*Packet RX buffer queue for requests.*
- [BFifo< BoapMcPacket > opacketTxQueue](#)  
*Packet TX Queue.*
- [BSemaphoreCount opacketTxQueueWriteNum](#)  
*Packet TX Queue number.*
- [BSemaphore opacketTxSema](#)  
*Wait for TX semaphore.*

### 5.47.1 Constructor & Destructor Documentation

### 5.47.1.1 BoapMcComms()

```
BoapMcComms::BoapMcComms (
    Bool threaded = 0,
    BUInt rxQueueSize = 4 )
```

### 5.47.1.2 ~BoapMcComms()

```
BoapMcComms::~~BoapMcComms ( ) [virtual]
```

## 5.47.2 Member Function Documentation

### 5.47.2.1 getApiVersion()

```
BUInt32 BoapMcComms::getApiVersion ( )
```

Returns the API version.

### 5.47.2.2 packetRecv()

```
BError BoapMcComms::packetRecv (
    BoapMcPacket & packet ) [protected]
```

Receives a packet.

### 5.47.2.3 packetSend()

```
BError BoapMcComms::packetSend (
    BoapMcPacket & packet ) [protected]
```

Receives a packet.

### 5.47.2.4 performCall()

```
BError BoapMcComms::performCall ( ) [protected]
```

Performs a RPC call to the remote side.

#### 5.47.2.5 performSend()

```
BError BoapMcComms::performSend ( ) [protected]
```

Performs a RPC send to the remote side.

#### 5.47.2.6 processPacket()

```
BError BoapMcComms::processPacket (
    BoapMcPacket & rx,
    BoapMcPacket & tx ) [virtual]
```

Process a recieved packet.

#### 5.47.2.7 processRequest()

```
BError BoapMcComms::processRequest (
    BTimeout timeoutUs = BTimeoutForever ) [virtual]
```

Check and process any request.

#### 5.47.2.8 processRequests()

```
BError BoapMcComms::processRequests (
    BTimeout timeoutUs = BTimeoutForever ) [virtual]
```

Check and process all requests.

#### 5.47.2.9 processRx()

```
BError BoapMcComms::processRx (
    BTimeout timeoutUs = BTimeoutForever ) [virtual]
```

Process any RX packets queuing them as needed.

!!! This should wait on comms for timeoutUs !!!

#### 5.47.2.10 setAddress()

```
void BoapMcComms::setAddress (
    BUInt8 addressTo,
    BUInt8 addressFrom )
```

Sets the to and from addresses.

#### 5.47.2.11 setComms() [1/2]

```
void BoapMcComms::setComms (
    BComms & comms )
```

Sets the communications interface to use.

#### 5.47.2.12 setComms() [2/2]

```
void BoapMcComms::setComms (
    BComms * comms )
```

Sets the communications interface to use.

#### 5.47.2.13 setCommsMode()

```
void BoapMcComms::setCommsMode (
    Bool slave,
    BUInt txQueueSize )
```

Sets slave mode.

#### 5.47.2.14 setTimeout()

```
BUInt32 BoapMcComms::setTimeout (
    BUInt32 timeoutUs )
```

Sets the call timeout returning the current value.

### 5.47.3 Member Data Documentation

#### 5.47.3.1 oaddressFrom

`BUInt8` BoapMcComms::oaddressFrom [protected]

#### 5.47.3.2 oaddressTo

`BUInt8` BoapMcComms::oaddressTo [protected]

#### 5.47.3.3 oapiVersion

`BUInt32` BoapMcComms::oapiVersion [protected]

#### 5.47.3.4 ocomms

`BComms*` BoapMcComms::ocomms [protected]

#### 5.47.3.5 olockCall

`BMutex` BoapMcComms::olockCall [protected]

Lock for RPC calls. Only one at a time.

#### 5.47.3.6 olockTx

`BMutex` BoapMcComms::olockTx [protected]

Lock for TX.

#### 5.47.3.7 opacket

`BoapMcPacket` BoapMcComms::opacket [protected]

Packet RX buffer.

#### 5.47.3.8 opacketReqQueue

[BQueue<BoapMcPacket>](#) BoapMcComms::opacketReqQueue [protected]

Packet RX buffer queue for requests.

#### 5.47.3.9 opacketReqRx

[BoapMcPacket](#) BoapMcComms::opacketReqRx [protected]

Packet RX buffer for requests.

#### 5.47.3.10 opacketReqTx

[BoapMcPacket](#) BoapMcComms::opacketReqTx [protected]

Packet TX buffer for requests.

#### 5.47.3.11 opacketRx

[BoapMcPacket](#) BoapMcComms::opacketRx [protected]

Packet RX buffer for calls.

#### 5.47.3.12 opacketRxSema

[BSemaphore](#) BoapMcComms::opacketRxSema [protected]

Wait RX semaphore.

#### 5.47.3.13 opacketTx

[BoapMcPacket](#) BoapMcComms::opacketTx [protected]

Packet TX buffer for calls.

#### 5.47.3.14 opacketTxQueue

[BFifo](#)<[BoapMcPacket](#)> BoapMcComms::opacketTxQueue [protected]

Packet TX Queue.

#### 5.47.3.15 opacketTxQueueWriteNum

[BSemaphoreCount](#) BoapMcComms::opacketTxQueueWriteNum [protected]

Packet TX Queue number.

#### 5.47.3.16 opacketTxSema

[BSemaphore](#) BoapMcComms::opacketTxSema [protected]

Wait for TX semaphore.

#### 5.47.3.17 oslave

[Bool](#) BoapMcComms::oslave [protected]

Set slave mode.

#### 5.47.3.18 otheaded

[Bool](#) BoapMcComms::othreaded [protected]

#### 5.47.3.19 otimeout

[BUInt32](#) BoapMcComms::otimeout [protected]

The timeout in us for calls.

The documentation for this class was generated from the following files:

- [BoapMc.h](#)
- [BoapMc.cpp](#)

## 5.48 BoapMcPacket Class Reference

```
#include <BoapMc.h>
```

### Public Attributes

- [BoapMcPacketHead](#) head
- char [data](#) [256 - sizeof([BoapMcPacketHead](#))]

### 5.48.1 Member Data Documentation

#### 5.48.1.1 data

```
char BoapMcPacket::data[256 - sizeof(BoapMcPacketHead)]
```

#### 5.48.1.2 head

```
BoapMcPacketHead BoapMcPacket::head
```

The documentation for this class was generated from the following file:

- [BoapMc.h](#)

## 5.49 BoapMcPacketHead Struct Reference

```
#include <BoapMc.h>
```

### Public Attributes

- [BUInt8](#) length
- [BUInt8](#) addressTo
- [BUInt8](#) addressFrom
- [BUInt8](#) cmd
- [BUInt16](#) error
- [BUInt16](#) checksum

### 5.49.1 Member Data Documentation

#### 5.49.1.1 addressFrom

`BUInt8` BoapMcPacketHead::addressFrom

#### 5.49.1.2 addressTo

`BUInt8` BoapMcPacketHead::addressTo

#### 5.49.1.3 checksum

`BUInt16` BoapMcPacketHead::checksum

#### 5.49.1.4 cmd

`BUInt8` BoapMcPacketHead::cmd

#### 5.49.1.5 error

`BUInt16` BoapMcPacketHead::error

#### 5.49.1.6 length

`BUInt8` BoapMcPacketHead::length

The documentation for this struct was generated from the following file:

- [BoapMc.h](#)

## 5.50 BoapMcServiceObject Class Reference

```
#include <BoapMc.h>
```

## Public Member Functions

- [BoapMcServiceObject \(\)](#)
- virtual [~BoapMcServiceObject \(\)](#)
- virtual [BError process \(BoapMcPacket &rx, BoapMcPacket &tx\)](#)
- virtual [BError processEvent \(BoapMcPacket &rx\)](#)

## Protected Member Functions

- [BError sendEvent \(BoapMcPacket &tx\)](#)

## Protected Attributes

- [BUInt32 oapiVersion](#)

## 5.50.1 Constructor & Destructor Documentation

### 5.50.1.1 BoapMcServiceObject()

```
BoapMcServiceObject::BoapMcServiceObject ( )
```

### 5.50.1.2 ~BoapMcServiceObject()

```
BoapMcServiceObject::~~BoapMcServiceObject ( ) [virtual]
```

## 5.50.2 Member Function Documentation

### 5.50.2.1 process()

```
BError BoapMcServiceObject::process (
    BoapMcPacket & rx,
    BoapMcPacket & tx ) [virtual]
```

### 5.50.2.2 processEvent()

```
BError BoapMcServiceObject::processEvent (
    BoapMcPacket & rx ) [virtual]
```

### 5.50.2.3 sendEvent()

```
BError BoapMcServiceObject::sendEvent (
    BoapMcPacket & tx ) [protected]
```

## 5.50.3 Member Data Documentation

### 5.50.3.1 oapiVersion

```
BUInt32 BoapMcServiceObject::oapiVersion [protected]
```

The documentation for this class was generated from the following files:

- [BoapMc.h](#)
- [BoapMc.cpp](#)

## 5.51 BoapMcSignalObject Class Reference

```
#include <BoapMc.h>
```

### Public Member Functions

- [BoapMcSignalObject](#) ([BComms](#) &comms)

### Protected Member Functions

- [BError performSend](#) ([BoapMcPacket](#) &tx)

### Protected Attributes

- [BComms](#) & [ocomms](#)

## 5.51.1 Constructor & Destructor Documentation

### 5.51.1.1 BoapMcSignalObject()

```
BoapMcSignalObject::BoapMcSignalObject (
    BComms & comms )
```

## 5.51.2 Member Function Documentation

### 5.51.2.1 performSend()

```
BError BoapMcSignalObject::performSend (  
    BoapMcPacket & tx ) [protected]
```

## 5.51.3 Member Data Documentation

### 5.51.3.1 ocomms

```
BComms& BoapMcSignalObject::ocomms [protected]
```

The documentation for this class was generated from the following files:

- [BoapMc.h](#)
- [BoapMc.cpp](#)

## 5.52 BObj Class Reference

```
#include <BObj.h>
```

### Public Member Functions

- [BObj](#) ()
- virtual [~BObj](#) ()
- virtual const char \* [getType](#) () const
- virtual const [BObjMember](#) \* [getMembers](#) () const
- virtual void [membersPrint](#) () const

*Prints out members.*

### 5.52.1 Constructor & Destructor Documentation

#### 5.52.1.1 BObj()

```
BObj::BObj ( )
```

### 5.52.1.2 ~BObj()

```
BObj::~~BObj ( ) [virtual]
```

## 5.52.2 Member Function Documentation

### 5.52.2.1 getMembers()

```
const BObjMember * BObj::getMembers ( ) const [virtual]
```

### 5.52.2.2 getType()

```
const char * BObj::getType ( ) const [virtual]
```

### 5.52.2.3 membersPrint()

```
void BObj::membersPrint ( ) const [virtual]
```

Prints out members.

The documentation for this class was generated from the following files:

- [BObj.h](#)
- [BObj.cpp](#)

## 5.53 BObjMember Struct Reference

```
#include <BTypes.h>
```

### Public Attributes

- [BType type](#)
- [BTypeComp typeComp](#)
- [BUInt16 dataOffset](#)
- [BUInt16 size](#)
- const char \* [typeName](#)
- const char \* [name](#)

## 5.53.1 Member Data Documentation

### 5.53.1.1 dataOffset

`BUInt16` `BObjMember::dataOffset`

### 5.53.1.2 name

`const char*` `BObjMember::name`

### 5.53.1.3 size

`BUInt16` `BObjMember::size`

### 5.53.1.4 type

`BType` `BObjMember::type`

### 5.53.1.5 typeComp

`BTypeComp` `BObjMember::typeComp`

### 5.53.1.6 typeName

`const char*` `BObjMember::typeName`

The documentation for this struct was generated from the following file:

- [BTypes.h](#)

## 5.54 BPwm Class Reference

```
#include <BPwm.h>
```

## Public Member Functions

- [BPwm \(\)](#)
- [BError init \(UInt8 dev, BUInt32 freq\)](#)
- [BError initChannel \(UInt8 chan\)](#)
- [BError set \(UInt8 chan, UInt16 duty\)](#)

## Protected Attributes

- [BTim otim](#)
- [BUInt32 operiod](#)

### 5.54.1 Constructor & Destructor Documentation

#### 5.54.1.1 BPwm()

```
BPwm::BPwm ( )
```

### 5.54.2 Member Function Documentation

#### 5.54.2.1 init()

```
BError BPwm::init (
    UInt8 dev,
    BUInt32 freq )
```

#### 5.54.2.2 initChannel()

```
BError BPwm::initChannel (
    UInt8 chan )
```

#### 5.54.2.3 set()

```
BError BPwm::set (
    UInt8 chan,
    UInt16 duty )
```

### 5.54.3 Member Data Documentation

#### 5.54.3.1 operiod

`BUInt32 BPwm::operiod` [protected]

#### 5.54.3.2 otim

`BTim BPwm::otim` [protected]

The documentation for this class was generated from the following files:

- [BPwm.h](#)
- [BPwm.cpp](#)

## 5.55 BQueue< T > Class Template Reference

Queue class.

```
#include <BQueue.h>
```

### Public Member Functions

- [BQueue](#) ([BUInt](#) size)
- [~BQueue](#) ()
- void [clear](#) ()  
*Clears a queue.*
- [BUInt](#) [writeAvailable](#) ()
- [BUInt](#) [readAvailable](#) ()
- [BUInt](#) [readAvailableFromIsr](#) ()
- [BError](#) [write](#) (const T &v, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Append an item onto the queue.*
- [BError](#) [read](#) (T &v, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Get an item from the queue.*
- [BError](#) [writeFromIsr](#) (const T &v)  
*Append an item onto the queue.*
- [BError](#) [readFromIsr](#) (T &v)  
*Get an item from the queue.*

### Protected Attributes

- QueueHandle\_t [oqueue](#)

### 5.55.1 Detailed Description

```
template<class T>
class BQueue< T >
```

Queue class.

### 5.55.2 Constructor & Destructor Documentation

#### 5.55.2.1 BQueue()

```
template<class T >
BQueue< T >::BQueue (
    BUInt size )
```

#### 5.55.2.2 ~BQueue()

```
template<class T >
BQueue< T >::~~BQueue ( )
```

### 5.55.3 Member Function Documentation

#### 5.55.3.1 clear()

```
template<class T >
void BQueue< T >::clear ( )
```

Clears a queue.

#### 5.55.3.2 read()

```
template<class T>
BError BQueue< T >::read (
    T & v,
    BTimeout timeoutUs = BTimeoutForever )
```

Get an item from the queue.

### 5.55.3.3 readAvailable()

```
template<class T >
BUInt BQueue< T >::readAvailable ( )
```

### 5.55.3.4 readAvailableFromIsr()

```
template<class T >
BUInt BQueue< T >::readAvailableFromIsr ( )
```

### 5.55.3.5 readFromIsr()

```
template<class T>
BError BQueue< T >::readFromIsr (
    T & v )
```

Get an item from the queue.

### 5.55.3.6 write()

```
template<class T>
BError BQueue< T >::write (
    const T & v,
    BTimeout timeoutUs = BTimeoutForever )
```

Append an item onto the queue.

### 5.55.3.7 writeAvailable()

```
template<class T >
BUInt BQueue< T >::writeAvailable ( )
```

### 5.55.3.8 writeFromIsr()

```
template<class T>
BError BQueue< T >::writeFromIsr (
    const T & v )
```

Append an item onto the queue.

## 5.55.4 Member Data Documentation

### 5.55.4.1 oqueue

```
template<class T>  
QueueHandle_t BQueue< T >::oqueue [protected]
```

The documentation for this class was generated from the following file:

- [BQueue.h](#)

## 5.56 BRandom Class Reference

```
#include <BRandom.h>
```

### Public Member Functions

- [BRandom](#) ()
- void [init](#) ()  
*Initialise.*
- [BUInt32 value](#) ()  
*Return a random value.*

## 5.56.1 Constructor & Destructor Documentation

### 5.56.1.1 BRandom()

```
BRandom::BRandom ( )
```

## 5.56.2 Member Function Documentation

### 5.56.2.1 init()

```
void BRandom::init ( )
```

Initialise.

### 5.56.2.2 value()

```
BUInt32 BRandom::value ( )
```

Return a random value.

The documentation for this class was generated from the following file:

- [BRandom.h](#)

## 5.57 BRefData Struct Reference

```
#include <BString.h>
```

### Public Attributes

- [BUInt refCount](#)  
*The reference count, how many users.*
- [BUInt len](#)  
*The actual length of data in oData.*
- char [data](#) [1]  
*The data.*

### 5.57.1 Member Data Documentation

#### 5.57.1.1 data

```
char BRefData::data[1]
```

The data.

#### 5.57.1.2 len

```
BUInt BRefData::len
```

The actual length of data in oData.

### 5.57.1.3 refCount

```
BUInt BRefData::refCount
```

The reference count, how many users.

The documentation for this struct was generated from the following file:

- [BString.h](#)

## 5.58 BRefString Class Reference

```
#include <BString.h>
```

### Public Member Functions

- [BRefString](#) (const char \*str=0, int len=-1, Bool copy=0)
- [BRefString](#) (const [BRefString](#) &str)
- [BRefString](#) (BUInt len)
- [~BRefString](#) ()
- [BUInt len](#) () const
- const char \* [str](#) () const
- char \* [str](#) ()
- void [refNew](#) (BUInt len)
- void [refDelete](#) ()
- void [refMakeWriteable](#) ()
- void [refWrite](#) (const char \*str, BUInt len)
- void [refResize](#) (BUInt len)
- void [clear](#) ()
- [BRefString](#) & [operator=](#) (const [BRefString](#) &ref)
- void [debugPrint](#) ()

### Static Public Attributes

- static Bool [alwaysCopy](#) = 0

### Private Attributes

- const char \* [ostr](#)
- [BRefData](#) \* [oref](#)

### 5.58.1 Constructor & Destructor Documentation

### 5.58.1.1 BRefString() [1/3]

```
BRefString::BRefString (
    const char * str = 0,
    int len = -1,
    Bool copy = 0 ) [inline]
```

### 5.58.1.2 BRefString() [2/3]

```
BRefString::BRefString (
    const BRefString & str ) [inline]
```

### 5.58.1.3 BRefString() [3/3]

```
BRefString::BRefString (
    BUInt len ) [inline]
```

### 5.58.1.4 ~BRefString()

```
BRefString::~BRefString ( ) [inline]
```

## 5.58.2 Member Function Documentation

### 5.58.2.1 clear()

```
void BRefString::clear ( ) [inline]
```

### 5.58.2.2 debugPrint()

```
void BRefString::debugPrint ( ) [inline]
```

### 5.58.2.3 len()

```
BUInt BRefString::len ( ) const [inline]
```

### 5.58.2.4 operator=()

```
BRefString & BRefString::operator= (
    const BRefString & ref ) [inline]
```

### 5.58.2.5 refDelete()

```
void BRefString::refDelete ( ) [inline]
```

### 5.58.2.6 refMakeWriteable()

```
void BRefString::refMakeWriteable ( ) [inline]
```

### 5.58.2.7 refNew()

```
void BRefString::refNew (
    BUInt len ) [inline]
```

### 5.58.2.8 refResize()

```
void BRefString::refResize (
    BUInt len ) [inline]
```

### 5.58.2.9 refWrite()

```
void BRefString::refWrite (
    const char * str,
    BUInt len ) [inline]
```

#### 5.58.2.10 str() [1/2]

```
const char * BRefString::str ( ) const [inline]
```

#### 5.58.2.11 str() [2/2]

```
char * BRefString::str ( ) [inline]
```

### 5.58.3 Member Data Documentation

#### 5.58.3.1 alwaysCopy

```
Bool BRefString::alwaysCopy = 0 [static]
```

#### 5.58.3.2 oref

```
BRefData* BRefString::oref [private]
```

#### 5.58.3.3 ostr

```
const char* BRefString::ostr [private]
```

The documentation for this class was generated from the following files:

- [BString.h](#)
- [BString.cpp](#)

## 5.59 BRtc Class Reference

```
#include <BRtc.h>
```

## Public Member Functions

- [BRtc](#) ()
- [BError](#) `init` (`Bool` reset=0)  
*Initialise.*
- [BError](#) `setTime` (`BTime` time)  
*Sets the current datetime.*
- [BError](#) `getTime` (`BTime` &time)  
*Returns the current datetime.*
- [BTime](#) `time` ()
- `void` [setBackupRegister](#) (`BUInt` num, `BUInt32` value)  
*Set a backup registers vaue.*
- [BUInt32](#) `getBackupRegister` (`BUInt` num)  
*Get a backup registers value.*

## Protected Member Functions

- [BError](#) `enterInitialisation` ()  
*Enters initialisation mode.*
- [BError](#) `waitForSync` ()  
*Waits for RTC sync.*

## Protected Attributes

- [BMutex](#) `oClock`  
*Allow multiple threads to access the clock.*

### 5.59.1 Constructor & Destructor Documentation

#### 5.59.1.1 BRtc()

```
BRtc::BRtc ( )
```

### 5.59.2 Member Function Documentation

#### 5.59.2.1 enterInitialisation()

```
BError BRtc::enterInitialisation ( ) [protected]
```

Enters initialisation mode.

### 5.59.2.2 getBackupRegister()

```
BUInt32 BRtc::getBackupRegister (
    BUInt num )
```

Get a backup registers value.

### 5.59.2.3 getTime()

```
BError BRtc::getTime (
    BTime & time )
```

Returns the current datetime.

### 5.59.2.4 init()

```
BError BRtc::init (
    Bool reset = 0 )
```

Initialise.

### 5.59.2.5 setBackupRegister()

```
void BRtc::setBackupRegister (
    BUInt num,
    BUInt32 value )
```

Set a backup registers vaue.

### 5.59.2.6 setTime()

```
BError BRtc::setTime (
    BTime time )
```

Sets the current datetime.

### 5.59.2.7 time()

```
BTime BRtc::time ( )
```

## 5.59.2.8 waitForSync()

`BError` `BRtc::waitForSync ( )` [protected]

Waits for RTC sync.

## 5.59.3 Member Data Documentation

## 5.59.3.1 olock

`BMutex` `BRtc::olock` [protected]

Allow multiple threads to access the clock.

The documentation for this class was generated from the following files:

- [BRtc.h](#)
- [BRtc.cpp](#)

## 5.60 BSdio Class Reference

```
#include <BSdio.h>
```

## Public Types

- enum `CardType` { `CardTypeNone`, `CardTypeSD`, `CardTypeSDHC`, `CardTypeEMMC` }
- enum `ResponseType` { `ResponseNone` = 0x00, `ResponseShort` = 0x40, `ResponseLong` = 0xC0 }

## Public Member Functions

- `BSdio` ( )
- `BError` `init` (`BUInt8` dev, `BUInt` speedKHz=400)
- `CardType` `getCardType` ( )
- `BUInt32` `numBlocks` ( )
- `BError` `getStatus` (`BUInt32` &status)
- `BError` `readBlocks` (`BUInt32` blockNumber, void \*data, `BUInt` num)  
*Read data from card.*
- `BError` `writeBlocks` (`BUInt32` blockNumber, void \*data, `BUInt` num)  
*Write data to card.*
- `BError` `trim` (`UInt32` blockNumber, `UInt32` num)  
*Mark blocks as free.*
- `BError` `erase` (`UInt32` blockNumber, `UInt32` num, `Bool` now)  
*Erase blocks.*
- void `backgroundEnable` (`Bool` on)
- void `sync` ( )
- `BError` `sendCmd` (`BUInt` cmd, `BUInt32` arg, `ResponseType` rType, `Bool` cpsm, `Bool` wait, `BUInt32` \*response)
- void `interrupt` ( )

## Protected Member Functions

- [BError cardInit \(\)](#)
- [BError cardFullSpeed \(\)](#)

## Protected Attributes

- [BMutex olock](#)  
*Lock for multi-threaded access.*
- [SDIO\\_TypeDef \\* osdmmc](#)  
*The SDMMC device.*
- [BUInt ospeed](#)  
*Clockrate of interface.*
- [CardType ocardType](#)  
*The card type.*
- [BUInt32 onumBlocks](#)  
*The number of 512 byte blocks.*
- [BUInt orca](#)  
*Cards relative address.*
- [BDma odma](#)  
*The DMAB.*
- [BUInt32 ostatus](#)  
*Interrupt status.*
- [BSemaphore oint](#)  
*Interrupt.*
- [BSemaphore oend](#)  
*End of transaction semaphore.*

## 5.60.1 Member Enumeration Documentation

### 5.60.1.1 CardType

enum [BSdio::CardType](#)

#### Enumerator

|              |  |
|--------------|--|
| CardTypeNone |  |
| CardTypeSD   |  |
| CardTypeSDHC |  |
| CardTypeEMMC |  |

### 5.60.1.2 ResponseType

enum [BSdio::ResponseType](#)

## Enumerator

|               |  |
|---------------|--|
| ResponseNone  |  |
| ResponseShort |  |
| ResponseLong  |  |

## 5.60.2 Constructor & Destructor Documentation

### 5.60.2.1 BSdio()

```
BSdio::BSdio ( )
```

## 5.60.3 Member Function Documentation

### 5.60.3.1 backgroundEnable()

```
void BSdio::backgroundEnable (
    Bool on )
```

### 5.60.3.2 cardFullSpeed()

```
BError BSdio::cardFullSpeed ( ) [protected]
```

### 5.60.3.3 cardInit()

```
BError BSdio::cardInit ( ) [protected]
```

### 5.60.3.4 erase()

```
BError BSdio::erase (
    UInt32 blockNumber,
    UInt32 num,
    Bool now )
```

Erase blocks.

### 5.60.3.5 getCardType()

```
BSdio::CardType BSdio::getCardType ( )
```

### 5.60.3.6 getStatus()

```
BError BSdio::getStatus (
    BUInt32 & status )
```

### 5.60.3.7 init()

```
BError BSdio::init (
    BUInt8 dev,
    BUInt speedKHz = 400 )
```

### 5.60.3.8 interrupt()

```
void BSdio::interrupt ( )
```

### 5.60.3.9 numBlocks()

```
BUInt32 BSdio::numBlocks ( )
```

### 5.60.3.10 readBlocks()

```
BError BSdio::readBlocks (
    BUInt32 blockNumber,
    void * data,
    BUInt num )
```

Read data from card.

### 5.60.3.11 sendCmd()

```
BError BSdio::sendCmd (
    BUInt cmd,
    BUInt32 arg,
    ResponseType rType,
    Bool cpsm,
    Bool wait,
    BUInt32 * response )
```

### 5.60.3.12 sync()

```
void BSdio::sync ( )
```

### 5.60.3.13 trim()

```
BError BSdio::trim (
    UInt32 blockNumber,
    UInt32 num )
```

Mark blocks as free.

### 5.60.3.14 writeBlocks()

```
BError BSdio::writeBlocks (
    BUInt32 blockNumber,
    void * data,
    BUInt num )
```

Write data to card.

## 5.60.4 Member Data Documentation

### 5.60.4.1 ocardType

```
CardType BSdio::ocardType [protected]
```

The card type.

#### 5.60.4.2 odma

`BDma` `BSdio::odma` [protected]

The DMAB.

#### 5.60.4.3 oend

`BSemaphore` `BSdio::oend` [protected]

End of transaction semaphore.

#### 5.60.4.4 oint

`BSemaphore` `BSdio::oint` [protected]

Interrupt.

#### 5.60.4.5 olock

`BMutex` `BSdio::olock` [protected]

Lock for multi-threaded access.

#### 5.60.4.6 onumBlocks

`BUInt32` `BSdio::onumBlocks` [protected]

The number of 512 byte blocks.

#### 5.60.4.7 orca

`BUInt` `BSdio::orca` [protected]

Cards relative address.

## 5.60.4.8 osdmmc

```
SDIO_TypeDef* BSdio::osdmmc [protected]
```

The SDMMC device.

## 5.60.4.9 ospeed

```
BUInt BSdio::ospeed [protected]
```

Clockrate of interface.

## 5.60.4.10 ostatus

```
BUInt32 BSdio::ostatus [protected]
```

Interrupt status.

The documentation for this class was generated from the following files:

- [BSdio.h](#)
- [BSdio.cpp](#)

## 5.61 BSemaphore Class Reference

Semaphore class.

```
#include <BSemaphore.h>
```

### Public Member Functions

- [BSemaphore](#) ()
- [BSemaphore](#) (const [BSemaphore](#) &semaphore)
- [~BSemaphore](#) ()
- [Bool](#) wait ([BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait for the semaphore.*
- [Bool](#) set ()  
*Set the semaphore, returns 1 if another task needs to run.*
- [Bool](#) setFromIsr ()  
*Set the semaphore from an IRQ handler, returns 1 if another task needs to run.*
- [BSemaphore](#) & operator= (const [BSemaphore](#) &semaphore)

## Private Attributes

- SemaphoreHandle\_t [osemaphore](#)

### 5.61.1 Detailed Description

Semaphore class.

### 5.61.2 Constructor & Destructor Documentation

#### 5.61.2.1 BSemaphore() [1/2]

```
BSemaphore::BSemaphore ( )
```

#### 5.61.2.2 BSemaphore() [2/2]

```
BSemaphore::BSemaphore (
    const BSemaphore & semaphore )
```

#### 5.61.2.3 ~BSemaphore()

```
BSemaphore::~~BSemaphore ( )
```

### 5.61.3 Member Function Documentation

#### 5.61.3.1 operator=()

```
BSemaphore & BSemaphore::operator= (
    const BSemaphore & semaphore )
```

#### 5.61.3.2 set()

```
Bool BSemaphore::set ( )
```

Set the semaphore, returns 1 if another task needs to run.

## 5.61.3.3 setFromIsr()

```
Bool BSemaphore::setFromIsr ( )
```

Set the semaphore from an IRQ handler, returns 1 if another task needs to run.

## 5.61.3.4 wait()

```
Bool BSemaphore::wait (
    BTimeout timeoutUs = BTimeoutForever )
```

Wait for the semaphore.

## 5.61.4 Member Data Documentation

## 5.61.4.1 osemaphore

```
SemaphoreHandle_t BSemaphore::osemaphore [private]
```

The documentation for this class was generated from the following files:

- [BSemaphore.h](#)
- [BSemaphore.cpp](#)

## 5.62 BSemaphoreBool Class Reference

```
#include <BSemaphore.h>
```

## Public Member Functions

- [BSemaphoreBool](#) ()
- [BSemaphoreBool](#) (const [BSemaphoreBool](#) &semaphore)
- [~BSemaphoreBool](#) ()
- [Bool set](#) ([Bool](#) on=1)
- [Bool clear](#) ()
- [Bool wait](#) ([Bool](#) v=1, [BTimeout](#) timeoutUs=[BTimeoutForever](#))  
*Wait for the semaphore.*
- [Bool value](#) ()
- [operator int](#) ()
- [int operator==](#) ([Bool](#) on)
- [BSemaphoreBool & operator=](#) ([Bool](#) on)

## Private Attributes

- [BSemaphore](#) osema
- volatile [Bool](#) ovalue

## 5.62.1 Constructor & Destructor Documentation

### 5.62.1.1 BSemaphoreBool() [1/2]

```
BSemaphoreBool::BSemaphoreBool ( )
```

### 5.62.1.2 BSemaphoreBool() [2/2]

```
BSemaphoreBool::BSemaphoreBool (
    const BSemaphoreBool & semaphore )
```

### 5.62.1.3 ~BSemaphoreBool()

```
BSemaphoreBool::~~BSemaphoreBool ( )
```

## 5.62.2 Member Function Documentation

### 5.62.2.1 clear()

```
Bool BSemaphoreBool::clear ( )
```

### 5.62.2.2 operator int()

```
BSemaphoreBool::operator int ( )
```

### 5.62.2.3 operator=()

```
BSemaphoreBool & BSemaphoreBool::operator= (
    Bool on )
```

### 5.62.2.4 operator==( )

```
int BSemaphoreBool::operator== (
    Bool on )
```

### 5.62.2.5 set()

```
Bool BSemaphoreBool::set (
    Bool on = 1 )
```

### 5.62.2.6 value()

```
Bool BSemaphoreBool::value ( )
```

### 5.62.2.7 wait()

```
Bool BSemaphoreBool::wait (
    Bool v = 1,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait for the semaphore.

## 5.62.3 Member Data Documentation

### 5.62.3.1 osema

```
BSemaphore BSemaphoreBool::osema [private]
```

### 5.62.3.2 ovalue

```
volatile Bool BSemaphoreBool::ovalue [private]
```

The documentation for this class was generated from the following files:

- [BSemaphore.h](#)
- [BSemaphore.cpp](#)

## 5.63 BSemaphoreCount Class Reference

```
#include <BSemaphore.h>
```

### Public Member Functions

- [BSemaphoreCount](#) ()
- [BSemaphoreCount](#) (const [BSemaphoreCount](#) &semaphore)
- [~BSemaphoreCount](#) ()
- [Bool setValue](#) ([BUInt](#) v)
- [Bool add](#) (int v=1)
  - Set the semaphore.*
- [Bool addFromIsrc](#) (int v=1)
  - Set the semaphore from an IRQ handler, returns 1 if another task needs to run.*
- [Bool wait](#) ([BUInt](#) v=1, [BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Wait for the semaphore.*
- [Bool take](#) ([BUInt](#) v=1, [BTimeout](#) timeoutUs=[BTimeoutForever](#))
  - Take for the semaphore.*
- [BUInt value](#) ()
- [BSemaphoreCount](#) & [operator=](#) (const [BSemaphoreCount](#) &semaphore)

### Private Attributes

- [BSemaphore](#) osema
- volatile [BUInt](#) ovalue

### 5.63.1 Constructor & Destructor Documentation

#### 5.63.1.1 BSemaphoreCount() [1/2]

```
BSemaphoreCount::BSemaphoreCount ( )
```

### 5.63.1.2 BSemaphoreCount() [2/2]

```
BSemaphoreCount::BSemaphoreCount (
    const BSemaphoreCount & semaphore )
```

### 5.63.1.3 ~BSemaphoreCount()

```
BSemaphoreCount::~~BSemaphoreCount ( )
```

## 5.63.2 Member Function Documentation

### 5.63.2.1 add()

```
Bool BSemaphoreCount::add (
    int v = 1 )
```

Set the semaphore.

### 5.63.2.2 addFromIsr()

```
Bool BSemaphoreCount::addFromIsr (
    int v = 1 )
```

Set the semaphore from an IRQ handler, returns 1 if another task needs to run.

### 5.63.2.3 operator=()

```
BSemaphoreCount & BSemaphoreCount::operator= (
    const BSemaphoreCount & semaphore )
```

### 5.63.2.4 setValue()

```
Bool BSemaphoreCount::setValue (
    BUInt v )
```

### 5.63.2.5 take()

```
Bool BSemaphoreCount::take (
    BUInt v = 1,
    BTimeout timeoutUs = BTimeoutForever )
```

Take for the semaphore.

### 5.63.2.6 value()

```
BUInt BSemaphoreCount::value ( )
```

### 5.63.2.7 wait()

```
Bool BSemaphoreCount::wait (
    BUInt v = 1,
    BTimeout timeoutUs = BTimeoutForever )
```

Wait for the semaphore.

## 5.63.3 Member Data Documentation

### 5.63.3.1 osema

```
BSemaphore BSemaphoreCount::osema [private]
```

### 5.63.3.2 ovalue

```
volatile BUInt BSemaphoreCount::ovalue [private]
```

The documentation for this class was generated from the following files:

- [BSemaphore.h](#)
- [BSemaphore.cpp](#)

## 5.64 BSnmppServer Class Reference

```
#include <BSnmppServer.h>
```

## Public Member Functions

- [BSnmpServer](#) ()
- [BError](#) `init` (const char \*systemName=0)
- void `run` ([Bool](#) run)

## 5.64.1 Constructor & Destructor Documentation

### 5.64.1.1 BSnmpServer()

```
BSnmpServer::BSnmpServer ( )
```

## 5.64.2 Member Function Documentation

### 5.64.2.1 `init()`

```
BError BSnmpServer::init (
    const char * systemName = 0 )
```

### 5.64.2.2 `run()`

```
void BSnmpServer::run (
    Bool run )
```

The documentation for this class was generated from the following files:

- [BSnmpServer.h](#)
- [BSnmpServer.cpp](#)

## 5.65 BSpi Class Reference

```
#include <BSpi.h>
```

## Public Types

- enum [Mode](#) { [Mode0](#) = 0, [Mode1](#) = 1, [Mode2](#) = 2, [Mode3](#) = 3 }

## Public Member Functions

- [BSpi](#) ([Bool](#) useDma=0)
- [BError](#) [init](#) ([UInt8](#) bus, [UInt](#) speed=1000000)  
*Initialise the SPI bus.*
- [BError](#) [init](#) ([UInt8](#) bus, [Pin](#) pinClk, [Pin](#) pinMosi, [Pin](#) pinMiso, [UInt](#) speed=1000000)  
*Initialise the SPI bus.*
- [BError](#) [initDev](#) ([UInt8](#) dev, [Pin](#) pinCs, [Mode](#) mode, [Bool](#) csActive=0)  
*Initialise an SPI device with appropriate CS. dev 0 - 7, mode 0 - 3.*
- [BError](#) [transact](#) ([UInt8](#) dev, void \*txBuf, int txLen, int pad, void \*rxBuf, int rxLen)
- [BError](#) [transactFull](#) ([UInt8](#) dev, void \*txBuf, int txLen, int rxIgnore, void \*rxBuf, int rxLen)
- void [setMode](#) ([Mode](#) mode)  
*Sets the SPI mode to 0,1,2,3.*
- [UInt8](#) [sendRecv](#) ([UInt8](#) txData, [UInt8](#) &rxData)

## Protected Attributes

- int [ouseDma](#)
- int [obus](#)
- [Pin](#) [opinCs](#) [4]
- [Mode](#) [omodes](#) [4]
- [Bool](#) [ocsActives](#) [4]
- [SPI\\_TypeDef](#) \* [osp](#)
- [BDma](#) [odmaTx](#)
- [BDma](#) [odmaRx](#)

## 5.65.1 Member Enumeration Documentation

### 5.65.1.1 Mode

```
enum BSpi::Mode
```

#### Enumerator

|       |  |
|-------|--|
| Mode0 |  |
| Mode1 |  |
| Mode2 |  |
| Mode3 |  |

## 5.65.2 Constructor & Destructor Documentation

### 5.65.2.1 BSpI()

```
BSpI::BSpI (
    Bool useDma = 0 )
```

## 5.65.3 Member Function Documentation

### 5.65.3.1 init() [1/2]

```
BError BSpI::init (
    UInt8 bus,
    UInt speed = 1000000 )
```

Initialise the SPI bus.

### 5.65.3.2 init() [2/2]

```
BError BSpI::init (
    UInt8 bus,
    Pin pinClk,
    Pin pinMosi,
    Pin pinMiso,
    UInt speed = 1000000 )
```

Initialise the SPI bus.

### 5.65.3.3 initDev()

```
BError BSpI::initDev (
    UInt8 dev,
    Pin pinCs,
    Mode mode,
    Bool csActive = 0 )
```

Initialise an SPI device with appropriate CS. dev 0 - 7, mode 0 - 3.

### 5.65.3.4 sendRecv()

```
UInt8 BSpI::sendRecv (
    UInt8 txData,
    UInt8 & rxData )
```

#### 5.65.3.5 setMode()

```
void BSpi::setMode (
    Mode mode )
```

Sets the SPI mode to 0,1,2,3.

#### 5.65.3.6 transact()

```
BError BSpi::transact (
    UInt8 dev,
    void * txBuf,
    int txLen,
    int pad,
    void * rxBuf,
    int rxLen )
```

#### 5.65.3.7 transactFull()

```
BError BSpi::transactFull (
    UInt8 dev,
    void * txBuf,
    int txLen,
    int rxIgnore,
    void * rxBuf,
    int rxLen )
```

### 5.65.4 Member Data Documentation

#### 5.65.4.1 obus

```
int BSpi::obus [protected]
```

#### 5.65.4.2 ocsActives

```
Bool BSpi::ocsActives[4] [protected]
```

#### 5.65.4.3 odmaRx

`BDma` `BSpi::odmaRx` [protected]

#### 5.65.4.4 odmaTx

`BDma` `BSpi::odmaTx` [protected]

#### 5.65.4.5 omodes

`Mode` `BSpi::omodes[4]` [protected]

#### 5.65.4.6 opinCs

`Pin` `BSpi::opinCs[4]` [protected]

#### 5.65.4.7 ospi

`SPI_TypeDef*` `BSpi::ospi` [protected]

#### 5.65.4.8 ouseDma

`int` `BSpi::ouseDma` [protected]

The documentation for this class was generated from the following files:

- [BSpi.h](#)
- [BSpi.cpp](#)

## 5.66 BString Class Reference

```
#include <BString.h>
```

## Public Member Functions

- **BString** (const char \*str=0, int len=-1, Bool copy=0)
- **BString** (const BString &string)
- **BString** (BChar c)
- **BString** (BInt v)
- **~BString** ()
- **BString copy** () const  
*Return an independant copy.*
- **BUInt len** () const  
*Length of string.*
- const char \* **str** () const  
*Ptr to char\* representation.*
- const char \* **retStr** () const  
*Ptr to char\* representation.*
- **BInt retInt** () const  
*Return string as a int.*
- **BUInt retUInt** () const  
*Return string as a int.*
- **BFloat64 retFloat64** () const  
*Return string as a double.*
- int **compare** (const BString &string) const  
*Compare strings.*
- int **compare** (const char \*string) const  
*Compare strings.*
- int **find** (char ch) const  
*Find ch in string searching forwards.*
- int **find** (BString str) const  
*Find string in string searching forwards.*
- int **findReverse** (char ch) const  
*Find ch in string searching backwards.*
- void **clear** ()  
*Clear the string.*
- **BString add** (const BString &str) const  
*Add strings returning result.*
- **BString & truncate** (BUInt len)  
*Truncate to length len.*
- **BString & pad** (BUInt len)  
*Pad to length len.*
- **BString & printf** (const char \*fmt,...)  
*Formatted print into the string.*
- **BString & toUpper** ()  
*Convert to uppercase.*
- **BString & toLower** ()  
*Convert to lowercase.*
- **BString subString** (BUInt start, int len) const  
*Returns substring.*
- **BString reverse** () const  
*Reverse character order.*
- void **del** (BUInt start, int len)  
*Delete substring.*

- void `insert` (`BUInt` start, const `BString` &str)
  - Insert substring.*
- void `append` (const `BString` &str)
  - Append a string.*
- `BUInt32` `hash` () const
  - return a hash value*
- `BString` `pullToken` (const `BString` &terminators)
  - Pull token from start of string.*
- `BString` `removeSeparators` (`BString` separators) const
  - Remove any char from sepatators from string.*
- `BList`< `BString` > `split` (char splitChar)
  - Split string into an array based on the character separator.*
- `BString` & `operator=` (const `BString` &string)
- char & `operator[]` (int pos)
- `operator` const char \* () const
- int `operator==` (const `BString` &s) const
- int `operator==` (const char \*s) const
- int `operator!=` (const `BString` &s) const
- int `operator!=` (const char \*s) const
- int `operator>` (const `BString` &s) const
- int `operator>` (const char \*s) const
- int `operator<` (const `BString` &s) const
- int `operator<` (const char \*s) const
- int `operator>=` (const `BString` &s) const
- int `operator>=` (const char \*s) const
- int `operator<=` (const `BString` &s) const
- int `operator<=` (const char \*s) const
- `BString` `operator+` (const `BString` &s) const
- `BString` `operator+=` (const `BString` &s)
- `BString` `operator+` (`BChar` ch) const
- void `debugPrint` ()

### Static Public Member Functions

- static `BString` `convert` (`Bool` v)
  - Converts Bool to string.*
- static `BString` `convert` (`BInt8` v)
  - Converts Int8 to string.*
- static `BString` `convert` (`BUInt8` v)
  - Converts UInt8 to string.*
- static `BString` `convert` (`BInt16` v)
  - Converts Int16 to string.*
- static `BString` `convert` (`BUInt16` v)
  - Converts UInt16 to string.*
- static `BString` `convert` (`BInt32` value)
  - Converts int to string.*
- static `BString` `convert` (`BUInt32` value)
  - Converts uint to string.*
- static `BString` `convert` (double value, int eFormat=0)
  - Converts double to string.*
- static `BString` `convert` (`BUInt64` value)

- Converts long long to string.*
- static [BString convert \(BChar value\)](#)  
*Converts BChar to string.*
- static [BString convertHex \(BInt value\)](#)  
*Converts int to string as hex value.*
- static [BString convertHex \(BUInt value\)](#)  
*Converts uint to string as hex value.*

## Protected Attributes

- [BRefString orefs](#)

## 5.66.1 Constructor & Destructor Documentation

### 5.66.1.1 BString() [1/4]

```
BString::BString (  
    const char * str = 0,  
    int len = -1,  
    Bool copy = 0 )
```

### 5.66.1.2 BString() [2/4]

```
BString::BString (  
    const BString & string )
```

### 5.66.1.3 BString() [3/4]

```
BString::BString (  
    BChar c )
```

### 5.66.1.4 BString() [4/4]

```
BString::BString (  
    BInt v )
```

### 5.66.1.5 ~BString()

```
BString::~~BString ( )
```

## 5.66.2 Member Function Documentation

### 5.66.2.1 add()

```
BString BString::add (
    const BString & str ) const
```

Add strings returning result.

### 5.66.2.2 append()

```
void BString::append (
    const BString & str )
```

Append a string.

### 5.66.2.3 clear()

```
void BString::clear ( )
```

Clear the string.

### 5.66.2.4 compare() [1/2]

```
int BString::compare (
    const BString & string ) const
```

Compare strings.

#### 5.66.2.5 compare() [2/2]

```
int BString::compare (
    const char * string ) const
```

Compare strings.

#### 5.66.2.6 convert() [1/10]

```
BString BString::convert (
    Bool v ) [static]
```

Converts Bool to string.

#### 5.66.2.7 convert() [2/10]

```
BString BString::convert (
    BInt8 v ) [static]
```

Converts Int8 to string.

#### 5.66.2.8 convert() [3/10]

```
BString BString::convert (
    BUInt8 v ) [static]
```

Converts UInt8 to string.

#### 5.66.2.9 convert() [4/10]

```
BString BString::convert (
    BInt16 v ) [static]
```

Converts Int16 to string.

#### 5.66.2.10 convert() [5/10]

```
BString BString::convert (
    BUInt16 v ) [static]
```

Converts UInt16 to string.

**5.66.2.11 convert()** [6/10]

```
BString BString::convert (
    BInt32 value ) [static]
```

Converts int to string.

**5.66.2.12 convert()** [7/10]

```
BString BString::convert (
    BUInt32 value ) [static]
```

Converts uint to string.

**5.66.2.13 convert()** [8/10]

```
BString BString::convert (
    double value,
    int eFormat = 0 ) [static]
```

Converts double to string.

**5.66.2.14 convert()** [9/10]

```
BString BString::convert (
    BUInt64 value ) [static]
```

Converts long long to string.

**5.66.2.15 convert()** [10/10]

```
BString BString::convert (
    BChar value ) [static]
```

Converts BChar to string.

**5.66.2.16 convertHex()** [1/2]

```
BString BString::convertHex (
    BInt value ) [static]
```

Converts int to string as hex value.

**5.66.2.17 convertHex()** [2/2]

```
BString BString::convertHex (
    BUInt value ) [static]
```

Converts uint to string as hex value.

**5.66.2.18 copy()**

```
BString BString::copy ( ) const
```

Return an independant copy.

**5.66.2.19 debugPrint()**

```
void BString::debugPrint ( )
```

**5.66.2.20 del()**

```
void BString::del (
    BUInt start,
    int len )
```

Delete substring.

**5.66.2.21 find()** [1/2]

```
int BString::find (
    char ch ) const
```

Find ch in string searching forwards.

#### 5.66.2.22 find() [2/2]

```
int BString::find (
    BString str ) const
```

Find string in string searching forwards.

#### 5.66.2.23 findReverse()

```
int BString::findReverse (
    char ch ) const
```

Find ch in string searching backwards.

#### 5.66.2.24 hash()

```
BUInt32 BString::hash ( ) const
```

return a hash value

#### 5.66.2.25 insert()

```
void BString::insert (
    BUInt start,
    const BString & str )
```

Insert substring.

#### 5.66.2.26 len()

```
BUInt BString::len ( ) const
```

Length of string.

#### 5.66.2.27 operator const char \*()

```
BString::operator const char * ( ) const [inline]
```

**5.66.2.28 operator!=( ) [1/2]**

```
int BString::operator!=(  
    const BString & s ) const [inline]
```

**5.66.2.29 operator!=( ) [2/2]**

```
int BString::operator!=(  
    const char * s ) const [inline]
```

**5.66.2.30 operator+( ) [1/2]**

```
BString BString::operator+ (  
    const BString & s ) const [inline]
```

**5.66.2.31 operator+( ) [2/2]**

```
BString BString::operator+ (  
    BChar ch ) const [inline]
```

**5.66.2.32 operator+=( )**

```
BString BString::operator+=(  
    const BString & s ) [inline]
```

**5.66.2.33 operator<( ) [1/2]**

```
int BString::operator< (  
    const BString & s ) const [inline]
```

**5.66.2.34 operator<( ) [2/2]**

```
int BString::operator< (  
    const char * s ) const [inline]
```

**5.66.2.35 operator<=()** [1/2]

```
int BString::operator<= (
    const BString & s ) const [inline]
```

**5.66.2.36 operator<=()** [2/2]

```
int BString::operator<= (
    const char * s ) const [inline]
```

**5.66.2.37 operator=()**

```
BString & BString::operator= (
    const BString & string )
```

**5.66.2.38 operator==(** [1/2]

```
int BString::operator==(
    const BString & s ) const [inline]
```

**5.66.2.39 operator==(** [2/2]

```
int BString::operator==(
    const char * s ) const [inline]
```

**5.66.2.40 operator>()** [1/2]

```
int BString::operator> (
    const BString & s ) const [inline]
```

**5.66.2.41 operator>()** [2/2]

```
int BString::operator> (
    const char * s ) const [inline]
```

#### 5.66.2.42 operator>=() [1/2]

```
int BString::operator>= (
    const BString & s ) const [inline]
```

#### 5.66.2.43 operator>=() [2/2]

```
int BString::operator>= (
    const char * s ) const [inline]
```

#### 5.66.2.44 operator[]()

```
char & BString::operator[] (
    int pos )
```

#### 5.66.2.45 pad()

```
BString & BString::pad (
    BUInt len )
```

Pad to length len.

#### 5.66.2.46 printf()

```
BString & BString::printf (
    const char * fmt,
    ... )
```

Formatted print into the string.

#### 5.66.2.47 pullToken()

```
BString BString::pullToken (
    const BString & terminators )
```

Pull token from start of string.

#### 5.66.2.48 removeSeparators()

```
BString BString::removeSeparators (
    BString separators ) const
```

Remove any char from sepatators from string.

#### 5.66.2.49 retFloat64()

```
BFloat64 BString::retFloat64 ( ) const
```

Return string as a double.

#### 5.66.2.50 retInt()

```
BInt BString::retInt ( ) const
```

Return string as a int.

#### 5.66.2.51 retStr()

```
const char * BString::retStr ( ) const
```

Ptr to char\* representation.

#### 5.66.2.52 retUInt()

```
BUInt BString::retUInt ( ) const
```

Return string as a int.

#### 5.66.2.53 reverse()

```
BString BString::reverse ( ) const
```

Reverse character order.

#### 5.66.2.54 split()

```
BList< BString > BString::split (
    char splitChar )
```

Split string into an array based on the character separator.

#### 5.66.2.55 str()

```
const char * BString::str ( ) const
```

Ptr to char\* representation.

#### 5.66.2.56 subString()

```
BString BString::subString (
    BUInt start,
    int len ) const
```

Returns substring.

#### 5.66.2.57 toLower()

```
BString & BString::toLower ( )
```

Convert to lowercase.

#### 5.66.2.58 toUpper()

```
BString & BString::toUpper ( )
```

Convert to uppercase.

#### 5.66.2.59 truncate()

```
BString & BString::truncate (
    BUInt len )
```

Truncate to length len.

### 5.66.3 Member Data Documentation

#### 5.66.3.1 orefs

```
BRefString BString::orefs [protected]
```

The documentation for this class was generated from the following files:

- [BString.h](#)
- [BString.cpp](#)

## 5.67 BSys Class Reference

```
#include <BSys.h>
```

### Public Member Functions

- [BSys](#) ()
- void [init](#) ()  
*Initialise system with optional system tick.*
- void [setTaskHeap](#) (void \*base, size\_t length)  
*Optionaly set the tasks heap and stack.*
- void [reset](#) (Bool bootLoader=0)  
*Reset the system.*
- void [start](#) ()  
*Start timers, interrupts etc.*
- void [run](#) (FuncRun func=0, BUInt stackSize=1024)  
*Run RTOS system by starting scheduler.*
- [UInt32](#) [timeUs](#) ()  
*Return the time in microseconds.*
- void [timeUsReset](#) ()  
*Reset system time.*
- void [systemStatePrint](#) ()  
*Print the system state.*
- void [taskStatsPrint](#) (Bool fromPrev=0, Bool noPrint=0)  
*Print information on the tasks running.*

### Public Attributes

- [BDevices](#) [devices](#)  
*Device power and clock control.*
- [BInterrupt](#) [interrupts](#)  
*Interrupts.*
- [BHeap](#) [taskHeap](#)  
*Optional heap for task data and stacks.*

## Static Public Attributes

- static `Bool rtosRunning = 0`  
*The RTOS is running.*

## Private Attributes

- `BTimer timer`

## 5.67.1 Constructor & Destructor Documentation

### 5.67.1.1 BSys()

```
BSys::BSys ( )
```

## 5.67.2 Member Function Documentation

### 5.67.2.1 init()

```
void BSys::init ( )
```

Initialise system with optional system tick.

### 5.67.2.2 reset()

```
void BSys::reset (
    Bool bootLoader = 0 )
```

Reset the system.

### 5.67.2.3 run()

```
void BSys::run (
    FuncRun func = 0,
    BUInt stackSize = 1024 )
```

Run RTOS system by starting scheduler.

#### 5.67.2.4 setTaskHeap()

```
void BSys::setTaskHeap (
    void * base,
    size_t length )
```

Optionally set the tasks heap and stack.

#### 5.67.2.5 start()

```
void BSys::start ( )
```

Start timers, interrupts etc.

#### 5.67.2.6 systemStatePrint()

```
void BSys::systemStatePrint ( )
```

Print the system state.

#### 5.67.2.7 taskStatsPrint()

```
void BSys::taskStatsPrint (
    Bool fromPrev = 0,
    Bool noPrint = 0 )
```

Print information on the tasks running.

#### 5.67.2.8 timeUs()

```
UInt32 BSys::timeUs ( )
```

Return the time in microseconds.

#### 5.67.2.9 timeUsReset()

```
void BSys::timeUsReset ( )
```

Reset system time.

### 5.67.3 Member Data Documentation

#### 5.67.3.1 devices

`BDevices` `BSys::devices`

Device power and clock control.

#### 5.67.3.2 interrupts

`BInterrupt` `BSys::interrupts`

Interrupts.

#### 5.67.3.3 otimer

`BTim` `BSys::otimer` `[private]`

#### 5.67.3.4 rtosRunning

`Bool` `BSys::rtosRunning = 0` `[static]`

The RTOS is running.

#### 5.67.3.5 taskHeap

`BHeap` `BSys::taskHeap`

Optional heap for task data and stacks.

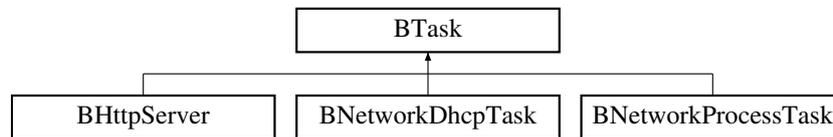
The documentation for this class was generated from the following files:

- [BSys.h](#)
- [BSys.cpp](#)

## 5.68 BTask Class Reference

```
#include <BTask.h>
```

Inheritance diagram for BTask:



### Public Member Functions

- `BTask` (const char \*name="", UInt stackSize=1024, UInt priority=1)
- `~BTask` ()
- void `init` (const char \*name, UInt stackSize=1024, UInt priority=1)
- `BError start` ()
  - Starts the task running.*
- void `stop` ()
  - Stops the task.*
- void `setPriority` (UInt priority)
  - Set the priority of the task: 0 upwards.*
- void `delayMs` (UInt ms)
  - Delay for a time in ms.*
- virtual void `run` ()
  - The tasks main run loop.*

### Static Public Member Functions

- static void `runTasks` ()
  - Main run tasks loop.*

### Protected Attributes

- const char \* `oname`
- UInt `ostackSize`
- UInt `opriority`
- TaskHandle\_t `otask`
- Bool `orunning`

#### 5.68.1 Constructor & Destructor Documentation

### 5.68.1.1 BTask()

```
BTask::BTask (
    const char * name = "",
    UInt stackSize = 1024,
    UInt priority = 1 )
```

### 5.68.1.2 ~BTask()

```
BTask::~BTask ( )
```

## 5.68.2 Member Function Documentation

### 5.68.2.1 delayMs()

```
void BTask::delayMs (
    UInt ms )
```

Delay for a time in ms.

### 5.68.2.2 init()

```
void BTask::init (
    const char * name,
    UInt stackSize = 1024,
    UInt priority = 1 )
```

### 5.68.2.3 run()

```
void BTask::run ( ) [virtual]
```

The tasks main run loop.

Reimplemented in [BHttpServer](#), [BNetworkDhcpTask](#), and [BNetworkProcessTask](#).

#### 5.68.2.4 runTasks()

```
void BTask::runTasks ( ) [static]
```

Main run tasks loop.

#### 5.68.2.5 setPriority()

```
void BTask::setPriority (
    UInt priority )
```

Set the priority of the task: 0 upwards.

#### 5.68.2.6 start()

```
BError BTask::start ( )
```

Starts the task running.

#### 5.68.2.7 stop()

```
void BTask::stop ( )
```

Stops the task.

### 5.68.3 Member Data Documentation

#### 5.68.3.1 oname

```
const char* BTask::oname [protected]
```

#### 5.68.3.2 opriority

```
UInt BTask::opriority [protected]
```

### 5.68.3.3 orunning

```
Bool BTask::orunning [protected]
```

### 5.68.3.4 ostackSize

```
UInt BTask::ostackSize [protected]
```

### 5.68.3.5 otask

```
TaskHandle_t BTask::otask [protected]
```

The documentation for this class was generated from the following files:

- [BTask.h](#)
- [BTask.cpp](#)

## 5.69 BTim Class Reference

```
#include <BTim.h>
```

### Public Types

- enum [Mode](#) { [ModeCountUp](#), [ModeCountDown](#) }
- enum [OcMode](#) { [OcModeOff](#) = 0x0, [OcModePwm1](#) = 0x6, [OcModePwm2](#) = 0x7 }

### Public Member Functions

- [BTim](#) ()
- [BError](#) [init](#) ([BUInt](#) timer, [Mode](#) mode, [BUInt](#) prescale, [BUInt](#) period)
- void [start](#) ([Bool](#) on)
- [BUInt32](#) [getCount](#) ()
- void [clear](#) ()
- [BError](#) [oclnit](#) ([BUInt](#) chan, [OcMode](#) ocMode, [Bool](#) enable)
- void [ocSetCompare](#) ([BUInt](#) chan, [BUInt](#) value)

### Protected Attributes

- [TIM\\_TypeDef](#) \* [otim](#)

## 5.69.1 Member Enumeration Documentation

### 5.69.1.1 Mode

```
enum BTim::Mode
```

### Enumerator

|               |  |
|---------------|--|
| ModeCountUp   |  |
| ModeCountDown |  |

#### 5.69.1.2 OcMode

```
enum BTim::OcMode
```

### Enumerator

|            |  |
|------------|--|
| OcModeOff  |  |
| OcModePwm1 |  |
| OcModePwm2 |  |

## 5.69.2 Constructor & Destructor Documentation

### 5.69.2.1 BTim()

```
BTim::BTim ( )
```

## 5.69.3 Member Function Documentation

### 5.69.3.1 clear()

```
void BTim::clear ( )
```

### 5.69.3.2 getCount()

```
BUInt32 BTim::getCount ( )
```

### 5.69.3.3 `init()`

```
BError BTim::init (
    BUInt timer,
    Mode mode,
    BUInt prescale,
    BUInt period )
```

### 5.69.3.4 `ocInit()`

```
BError BTim::ocInit (
    BUInt chan,
    OcMode ocMode,
    Bool enable )
```

### 5.69.3.5 `ocSetCompare()`

```
void BTim::ocSetCompare (
    BUInt chan,
    BUInt value )
```

### 5.69.3.6 `start()`

```
void BTim::start (
    Bool on )
```

## 5.69.4 Member Data Documentation

### 5.69.4.1 `otim`

```
TIM_TypeDef* BTim::otim [protected]
```

The documentation for this class was generated from the following files:

- [BTim.h](#)
- [BTim.cpp](#)

## 5.70 BTime Class Reference

```
#include <BTime.h>
```

### Public Member Functions

- [BTime](#) ([BUInt32](#) t=0)
- void [set](#) ([BUInt32](#) seconds)  
*Set the date and time.*
- void [set](#) ([BUInt](#) year, [BUInt](#) month, [BUInt](#) day, [BUInt](#) hour=0, [BUInt](#) minute=0, [BUInt](#) second=0)  
*Set the date and time.*
- void [setYearDay](#) ([BUInt](#) year, [BUInt](#) yearDay, [BUInt](#) hour=0, [BUInt](#) minute=0, [BUInt](#) second=0)  
*Set the date and time.*
- void [getDate](#) ([BUInt](#) &year, [BUInt](#) &month, [BUInt](#) &day) const  
*Return the date information.*
- void [getTime](#) ([BUInt](#) &hour, [BUInt](#) &minute, [BUInt](#) &second) const  
*Return the time information.*
- [BUInt32](#) [getSeconds](#) () const  
*Return the number of seconds.*
- int [isSet](#) () const  
*Check if set.*
- int [isLeapYear](#) ()  
*Returns if a leap year.*
- void [addSeconds](#) (int seconds)  
*Add the given number of seconds.*
- [BString](#) [getString](#) ([BString](#) format="isoT") const  
*Gets the date/time in string format.*
- [BError](#) [setString](#) (const [BString](#) dateTime)  
*Sets the date/time from string format.*
- int [operator==](#) (const [BTime](#) &time) const
- int [operator!=](#) (const [BTime](#) &time) const
- int [operator>](#) (const [BTime](#) &time) const
- int [operator>=](#) (const [BTime](#) &time) const
- int [operator<](#) (const [BTime](#) &time) const
- int [operator<=](#) (const [BTime](#) &time) const
- [BTime](#) [operator+](#) (int seconds) const
- [BTime](#) & [operator+=](#) (int seconds)

### Private Attributes

- [BUInt32](#) [otime](#)  
*Time in seconds since 1970. range 1970-01-02 to 2106-02-07.*

#### 5.70.1 Constructor & Destructor Documentation

### 5.70.1.1 BTime()

```
BTime::BTime (  
    BUInt32 t = 0 )
```

## 5.70.2 Member Function Documentation

### 5.70.2.1 addSeconds()

```
void BTime::addSeconds (  
    int seconds )
```

Add the given number of seconds.

### 5.70.2.2 getDate()

```
void BTime::getDate (  
    BUInt & year,  
    BUInt & month,  
    BUInt & day ) const
```

Return the date information.

### 5.70.2.3 getSeconds()

```
BUInt32 BTime::getSeconds ( ) const
```

Return the number of seconds.

### 5.70.2.4 getString()

```
BString BTime::getString (  
    BString format = "isoT" ) const
```

Gets the date/time in string format.

### 5.70.2.5 getTime()

```
void BTime::getTime (
    BUInt & hour,
    BUInt & minute,
    BUInt & second ) const
```

Return the time information.

### 5.70.2.6 isLeapYear()

```
int BTime::isLeapYear ( )
```

Returns if a leap year.

### 5.70.2.7 isSet()

```
int BTime::isSet ( ) const [inline]
```

Check if set.

### 5.70.2.8 operator!=(=)

```
int BTime::operator!=(= (
    const BTime & time ) const [inline]
```

### 5.70.2.9 operator+()

```
BTime BTime::operator+ (
    int seconds ) const [inline]
```

### 5.70.2.10 operator+=(=)

```
BTime& BTime::operator+=(= (
    int seconds ) [inline]
```

### 5.70.2.11 operator<()

```
int BTime::operator< (
    const BTime & time ) const [inline]
```

### 5.70.2.12 operator<=()

```
int BTime::operator<= (
    const BTime & time ) const [inline]
```

### 5.70.2.13 operator==(())

```
int BTime::operator==(
    const BTime & time ) const [inline]
```

### 5.70.2.14 operator>()

```
int BTime::operator> (
    const BTime & time ) const [inline]
```

### 5.70.2.15 operator>=()

```
int BTime::operator>= (
    const BTime & time ) const [inline]
```

### 5.70.2.16 set() [1/2]

```
void BTime::set (
    BUInt32 seconds )
```

Set the date and time.

### 5.70.2.17 set() [2/2]

```
void BTime::set (
    BUInt year,
    BUInt month,
    BUInt day,
    BUInt hour = 0,
    BUInt minute = 0,
    BUInt second = 0 )
```

Set the date and time.

### 5.70.2.18 setString()

```
BError BTime::setString (
    const BString dateTime )
```

Sets the date/time from string format.

### 5.70.2.19 setYearDay()

```
void BTime::setYearDay (
    BUInt year,
    BUInt yearDay,
    BUInt hour = 0,
    BUInt minute = 0,
    BUInt second = 0 )
```

Set the date and time.

## 5.70.3 Member Data Documentation

### 5.70.3.1 otime

```
BUInt32 BTime::otime [private]
```

Time in seconds since 1970. range 1970-01-02 to 2106-02-07.

The documentation for this class was generated from the following files:

- [BTime.h](#)
- [BTime.cpp](#)



## 5.71.1 Constructor & Destructor Documentation

### 5.71.1.1 BTimeUs()

```
BTimeUs::BTimeUs (
    BUInt64 t = 0 )
```

## 5.71.2 Member Function Documentation

### 5.71.2.1 addMicroSeconds()

```
void BTimeUs::addMicroSeconds (
    BUInt64 microseconds )
```

Add the given number of seconds.

### 5.71.2.2 addSeconds()

```
void BTimeUs::addSeconds (
    BUInt64 seconds )
```

Add the given number of seconds.

### 5.71.2.3 getDate()

```
void BTimeUs::getDate (
    BUInt & year,
    BUInt & month,
    BUInt & day ) const
```

Return the date information UTC.

### 5.71.2.4 getMicroSeconds()

```
BUInt64 BTimeUs::getMicroSeconds ( ) const
```

Return the number of micro seconds TAI.

#### 5.71.2.5 getSeconds()

```
BUInt64 BTimeUs::getSeconds ( ) const
```

Return the number of seconds TAI.

#### 5.71.2.6 getString()

```
BString BTimeUs::getString (
    BString format = "isoT" ) const
```

Gets the date/time in string format.

#### 5.71.2.7 getTime()

```
void BTimeUs::getTime (
    BUInt & hour,
    BUInt & minute,
    BUInt & second ) const
```

Return the time information UTC.

#### 5.71.2.8 isLeapYear()

```
int BTimeUs::isLeapYear ( )
```

Returns if a leap year.

#### 5.71.2.9 isSet()

```
int BTimeUs::isSet ( ) const [inline]
```

Check if set.

#### 5.71.2.10 operator!=(=)

```
int BTimeUs::operator!=(= (
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.11 operator+()

```
BTimeUs BTimeUs::operator+ (
    BInt64 microSeconds ) const [inline]
```

#### 5.71.2.12 operator+=()

```
BTimeUs& BTimeUs::operator+= (
    BInt64 microSeconds ) [inline]
```

#### 5.71.2.13 operator<()

```
int BTimeUs::operator< (
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.14 operator<=()

```
int BTimeUs::operator<= (
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.15 operator==(())

```
int BTimeUs::operator==(
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.16 operator>()

```
int BTimeUs::operator> (
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.17 operator>=()

```
int BTimeUs::operator>= (
    const BTimeUs & time ) const [inline]
```

#### 5.71.2.18 set() [1/2]

```
void BTimeUs::set (
    BUInt64 microseconds )
```

Set the time to TAI us.

#### 5.71.2.19 set() [2/2]

```
void BTimeUs::set (
    BUInt year,
    BUInt month,
    BUInt day,
    BUInt hour = 0,
    BUInt minute = 0,
    BUInt second = 0,
    BUInt microSecond = 0 )
```

Set the date and time from UTC.

#### 5.71.2.20 setString()

```
BError BTimeUs::setString (
    const BString dateTime )
```

Sets the date/time from string format.

#### 5.71.2.21 setYearDay()

```
void BTimeUs::setYearDay (
    BUInt year,
    BUInt yearDay,
    BUInt hour = 0,
    BUInt minute = 0,
    BUInt second = 0,
    BUInt microSecond = 0 )
```

Set the date and time from UTC.

### 5.71.3 Member Data Documentation

## 5.71.3.1 otime

```
BUInt64 BTimeUs::otime [private]
```

Time in microseconds since 1970-01-01 to TAI standard.

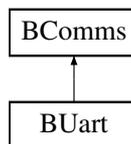
The documentation for this class was generated from the following files:

- [BTimeUs.h](#)
- [BTimeUs.cpp](#)

## 5.72 BUart Class Reference

```
#include <BUart.h>
```

Inheritance diagram for BUart:



## Public Member Functions

- [BUart](#) ([BUInt](#) fifoSize=64)
- [BError init](#) ([UInt8](#) dev, [Pin](#) pinTx, [Pin](#) pinRx, int baudRate=9600, int bits=8, int stopBits=2)  
*Initialise device: 1 to 3.*
- [BError init](#) ([UInt8](#) dev, int baudRate=9600, int bits=8, int stopBits=2)  
*Initialise device: 1 to 3.*
- [BError setMode485](#) ([Pin](#) txEnablePin, [Bool](#) rxDisable=1)  
*Sets the UART to 485 mode with a TX enable pin.*
- [BError setRtsCts](#) ([Pin](#) pinRts, [Pin](#) pinCts, [Bool](#) soft)  
*Sets the UART to use the RTS and CTS lines in hardware or software.*
- void [hold](#) ([Bool](#) hold)  
*Stop input via CTS.*
- [BUInt writeAvailable](#) ()  
*Check if can write, returns 0 or 1.*
- [BError write](#) (const void \*data, [UInt32](#) nBytes, [UInt32](#) &nWritten)  
*Write nBytes, will wait for hardware.*
- [BUInt readAvailable](#) ()  
*Check if data is available, returns 0 or 1.*
- [BError read](#) (void \*data, [UInt32](#) nBytes, [UInt32](#) &nRead)  
*Read nBytes, will wait for hardware.*
- [BError readLine](#) (void \*data, [UInt32](#) nBytes, [UInt32](#) &nRead)  
*Read a line (to NL char) or nBytes, will wait for hardware.*
- [BError wait](#) ([BUInt8](#) events, [BTimeout](#) timeoutUs=[BTimeoutForever](#), [BUInt32](#) num=1)
- void [flushRx](#) ()
- void [interrupt](#) ()

## Static Public Attributes

- static BUart \* devs []

## Protected Attributes

- BFifo< char > otxFifo
- BFifo< char > orxFifo
- int odev
- USART\_TypeDef \* ouart
- Pin otxEnablePin
- Bool orxDisable
- BSemaphore owait
- BUInt owaitReadNum
- Bool orx
- BUInt32 olastRxTime
- Pin opinCts
- Pin opinRts

## Additional Inherited Members

### 5.72.1 Constructor & Destructor Documentation

#### 5.72.1.1 BUart()

```
BUart::BUart (
    BUInt fifoSize = 64 )
```

### 5.72.2 Member Function Documentation

#### 5.72.2.1 flushRx()

```
void BUart::flushRx ( )
```

#### 5.72.2.2 hold()

```
void BUart::hold (
    Bool hold )
```

Stop input via CTS.

### 5.72.2.3 `init()` [1/2]

```
BError BUart::init (
    UInt8 dev,
    Pin pinTx,
    Pin pinRx,
    int baudRate = 9600,
    int bits = 8,
    int stopBits = 2 )
```

Initialise device: 1 to 3.

### 5.72.2.4 `init()` [2/2]

```
BError BUart::init (
    UInt8 dev,
    int baudRate = 9600,
    int bits = 8,
    int stopBits = 2 )
```

Initialise device: 1 to 3.

### 5.72.2.5 `interrupt()`

```
void BUart::interrupt ( )
```

### 5.72.2.6 `read()`

```
BError BUart::read (
    void * data,
    UInt32 nBytes,
    UInt32 & nRead ) [virtual]
```

Read nBytes, will wait for hardware.

Implements [BComms](#).

### 5.72.2.7 `readAvailable()`

```
BUInt BUart::readAvailable ( ) [virtual]
```

Check if data is available, returns 0 or 1.

Reimplemented from [BComms](#).

### 5.72.2.8 readLine()

```
BError BUart::readLine (
    void * data,
    UInt32 nBytes,
    UInt32 & nRead )
```

Read a line (to NL char) or nBytes, will wait for hardware.

### 5.72.2.9 setMode485()

```
BError BUart::setMode485 (
    Pin txEnablePin,
    Bool rxDisable = 1 )
```

Sets the UART to 485 mode with a TX enable pin.

### 5.72.2.10 setRtsCts()

```
BError BUart::setRtsCts (
    Pin pinRts,
    Pin pinCts,
    Bool soft )
```

Sets the UART to use the RTS and CTS lines in hardware or software.

### 5.72.2.11 wait()

```
BError BUart::wait (
    BUInt8 events,
    BTimeout timeoutUs = BTimeoutForever,
    BUInt32 num = 1 )
```

### 5.72.2.12 write()

```
BError BUart::write (
    const void * data,
    UInt32 nBytes,
    UInt32 & nWritten ) [virtual]
```

Write nBytes, will wait for hardware.

Implements [BComms](#).

### 5.72.2.13 writeAvailable()

```
BUInt BUart::writeAvailable ( ) [virtual]
```

Check if can write, returns 0 or 1.

Reimplemented from [BComms](#).

## 5.72.3 Member Data Documentation

### 5.72.3.1 devs

```
BUart * BUart::devs [static]
```

### 5.72.3.2 odev

```
int BUart::odev [protected]
```

### 5.72.3.3 olastRxTime

```
BUInt32 BUart::olastRxTime [protected]
```

### 5.72.3.4 opinCts

```
Pin BUart::opinCts [protected]
```

### 5.72.3.5 opinRts

```
Pin BUart::opinRts [protected]
```

### 5.72.3.6 orx

```
Bool BUart::orx [protected]
```

### 5.72.3.7 orxDisable

`Bool BUart::orxDisable [protected]`

### 5.72.3.8 orxFifo

`BFifo<char> BUart::orxFifo [protected]`

### 5.72.3.9 otxEnablePin

`Pin BUart::otxEnablePin [protected]`

### 5.72.3.10 otxFifo

`BFifo<char> BUart::otxFifo [protected]`

### 5.72.3.11 ouart

`USART_TypeDef* BUart::ouart [protected]`

### 5.72.3.12 owait

`BSemaphore BUart::owait [protected]`

### 5.72.3.13 owaitReadNum

`BUInt BUart::owaitReadNum [protected]`

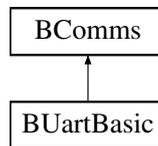
The documentation for this class was generated from the following files:

- [BUart.h](#)
- [BUart.cpp](#)

## 5.73 BUartBasic Class Reference

```
#include <BUartBasic.h>
```

Inheritance diagram for BUartBasic:



### Public Member Functions

- [BUartBasic](#) ()
- [BError](#) [init](#) ([UInt8](#) dev, [Pin](#) pinTx, [Pin](#) pinRx, int baudRate=9600, int bits=8, int stopBits=2)  
*Initialise device: 1 to 3.*
- [BError](#) [init](#) ([UInt8](#) dev, int baudRate=9600, int bits=8, int stopBits=2)  
*Initialise device: 1 to 3.*
- [BError](#) [setMode485](#) ([Pin](#) txEnablePin, [Bool](#) rxDisable=1)  
*Sets the UART to 485 mode with a TX enable pin.*
- [BUInt](#) [writeAvailable](#) ()  
*Check if can write, returns 0 or 1.*
- [BError](#) [write](#) (const void \*data, [UInt32](#) nBytes, [UInt32](#) &nWritten)  
*Write nBytes, will wait for hardware.*
- [BUInt](#) [readAvailable](#) ()  
*Check if data is available, returns 0 or 1.*
- [BError](#) [read](#) (void \*data, [UInt32](#) nBytes, [UInt32](#) &nRead)  
*Read nBytes, will wait for hardware.*
- int [printf](#) (const char \*fmt,...)

### Protected Attributes

- [USART\\_TypeDef](#) \* ouart
- [Pin](#) otxEnablePin
- [Bool](#) orxDisable

### Additional Inherited Members

#### 5.73.1 Constructor & Destructor Documentation

##### 5.73.1.1 BUartBasic()

```
BUartBasic::BUartBasic ( )
```

## 5.73.2 Member Function Documentation

### 5.73.2.1 `init()` [1/2]

```
BError BUartBasic::init (
    UInt8 dev,
    Pin pinTx,
    Pin pinRx,
    int baudRate = 9600,
    int bits = 8,
    int stopBits = 2 )
```

Initialise device: 1 to 3.

### 5.73.2.2 `init()` [2/2]

```
BError BUartBasic::init (
    UInt8 dev,
    int baudRate = 9600,
    int bits = 8,
    int stopBits = 2 )
```

Initialise device: 1 to 3.

### 5.73.2.3 `printf()`

```
int BUartBasic::printf (
    const char * fmt,
    ... )
```

### 5.73.2.4 `read()`

```
BError BUartBasic::read (
    void * data,
    UInt32 nBytes,
    UInt32 & nRead ) [virtual]
```

Read nBytes, will wait for hardware.

Implements [BComms](#).

#### 5.73.2.5 readAvailable()

```
BUInt BUartBasic::readAvailable ( ) [virtual]
```

Check if data is available, returns 0 or 1.

Reimplemented from [BComms](#).

#### 5.73.2.6 setMode485()

```
BError BUartBasic::setMode485 (
    Pin txEnablePin,
    Bool rxDisable = 1 )
```

Sets the UART to 485 mode with a TX enable pin.

#### 5.73.2.7 write()

```
BError BUartBasic::write (
    const void * data,
    UInt32 nBytes,
    UInt32 & nWritten ) [virtual]
```

Write nBytes, will wait for hardware.

Implements [BComms](#).

#### 5.73.2.8 writeAvailable()

```
BUInt BUartBasic::writeAvailable ( ) [virtual]
```

Check if can write, returns 0 or 1.

Reimplemented from [BComms](#).

### 5.73.3 Member Data Documentation

#### 5.73.3.1 orxDisable

```
Bool BUartBasic::orxDisable [protected]
```

### 5.73.3.2 otxEnablePin

```
Pin BUartBasic::otxEnablePin [protected]
```

### 5.73.3.3 ouart

```
USART_TypeDef* BUartBasic::ouart [protected]
```

The documentation for this class was generated from the following files:

- [BUartBasic.h](#)
- [BUartBasic.cpp](#)

## 5.74 BUsbConfig Struct Reference

```
#include <BUsbSerial.h>
```

### Public Attributes

- [BUInt16 vendorId](#)
- [BUInt16 productId](#)
- [BUInt8 devClass](#)
- [BUInt8 devSubClass](#)
- [BUInt8 devProtocol](#)
- const char \* [manufacturer](#)
- const char \* [product](#)
- const char \* [serialNumber](#)

### 5.74.1 Member Data Documentation

#### 5.74.1.1 devClass

```
BUInt8 BUsbConfig::devClass
```

#### 5.74.1.2 devProtocol

```
BUInt8 BUsbConfig::devProtocol
```

### 5.74.1.3 devSubClass

`BUInt8 BUsbConfig::devSubClass`

### 5.74.1.4 manufacturer

`const char* BUsbConfig::manufacturer`

### 5.74.1.5 product

`const char* BUsbConfig::product`

### 5.74.1.6 productId

`BUInt16 BUsbConfig::productId`

### 5.74.1.7 serialNumber

`const char* BUsbConfig::serialNumber`

### 5.74.1.8 vendorId

`BUInt16 BUsbConfig::vendorId`

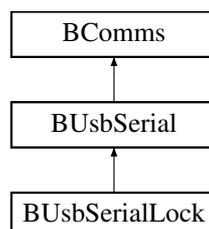
The documentation for this struct was generated from the following file:

- [BUsbSerial.h](#)

## 5.75 BUsbSerial Class Reference

```
#include <BUsbSerial.h>
```

Inheritance diagram for BUsbSerial:



## Public Member Functions

- [BUbSerial](#) ([BUInt](#) rxFifoSize=512, [BUInt](#) txFifoSize=512)
- [BError](#) [init](#) ([BUbType](#) type=[BUbTypeFullSpeed](#), [BUbConfig](#) \*usbConfig=0)
  - Initialise with optional config params.*
- void [setMaximumUsbChunk](#) ([BUInt](#) usbChunkSize)
  - Sets the maximum USB transmitted chunk size.*
- void [close](#) ()
  - Close down the USB interface.*
- void [setBlocking](#) ([Bool](#) on)
  - Enable blocking mode.*
- void [setSync](#) ([Bool](#) on)
  - Set synchronous mode for writes.*
- void [start](#) ()
  - Start processing.*
- int [printf](#) (const char \*fmt,...)
- int [printfv](#) (const char \*fmt, va\_list args)
- [BUInt](#) [writeAvailable](#) ()
  - How much write space is available.*
- [BError](#) [write](#) (const void \*data, [UInt32](#) nBytes, [UInt32](#) &nWritten)
  - Write a set of data.*
- [BUInt](#) [readAvailable](#) ()
  - How many bytes of read data is available.*
- [BError](#) [read](#) (void \*data, [UInt32](#) nBytes, [UInt32](#) &nRead)
  - Read the data.*
- [BError](#) [wait](#) ([BUInt8](#) events, [BTimeout](#) timeoutUs=[BTimeoutForever](#), [BUInt32](#) num=1)
- [BError](#) [usbInitHardware](#) ()
- [BError](#) [usbInitUsbDev](#) ([USBHandleTypeDef](#) \*pdev)
- [BUInt8](#) [usbInit](#) ([USBHandleTypeDef](#) \*pdev, [BUInt8](#) cfgidx)
- void [usbStarted](#) ()
  - The USB device is connected and running.*
- [BUInt8](#) [usbDeInit](#) ([USBHandleTypeDef](#) \*pdev, [BUInt8](#) cfgidx)
- [BUInt8](#) [usbSetup](#) ([USBHandleTypeDef](#) \*pdev, [USBSetupReqTypedef](#) \*req)
- void [usbTx](#) ()
  - Data to USB stack.*
- [BUInt8](#) [usbTxComplete](#) ()
  - Data has been sent.*
- int [usbRx](#) ([USBHandleTypeDef](#) \*pdev, [BUInt](#) len)
  - Data from USB stack.*
- void [usbInterrupt](#) ()

## Private Attributes

- [BFifo](#)< char > [otxFifo](#)
  - The transmit fifo.*
- [BFifo](#)< char > [orxFifo](#)
  - The receive fifo.*
- [Bool](#) [ohighSpeed](#)
  - High speed USB.*
- [BUInt](#) [opacketSize](#)
  - The maximum packet size.*

- [Bool oinitialised](#)  
*The USB device has initialised.*
- [Bool oconnected](#)  
*The USB is connected.*
- [BSemaphore owait](#)  
*Read wait semaphore.*
- [BUInt owaitReadNum](#)  
*The number of bytes to wait for.*
- [Bool oblocking](#)  
*Device set to block on writes.*
- [Bool osync](#)  
*Device set to wait for writes.*
- [USB\\_D\\_HandleTypeDef ousbDev](#)  
*The USB device.*
- [PCD\\_HandleTypeDef ohpcc](#)  
*The USB PCD layer.*
- [Bool otxSending](#)  
*USB is transmitting.*
- [int otxSendLen](#)  
*The length of data sent.*
- [char orxBuffer \[512\]](#)  
*Receive packet buffer.*
- [Bool orxStalled](#)  
*Receive has stalled as no room in buffer.*
- [BUInt ousbChunkSize](#)  
*The maximum USB chunk size.*

## Additional Inherited Members

### 5.75.1 Constructor & Destructor Documentation

#### 5.75.1.1 BUsbSerial()

```
BUsbSerial::BUsbSerial (
    BUInt rxFifoSize = 512,
    BUInt txFifoSize = 512 )
```

### 5.75.2 Member Function Documentation

#### 5.75.2.1 close()

```
void BUsbSerial::close ( )
```

Close down the USB interface.

### 5.75.2.2 `init()`

```
BError BUsbSerial::init (
    BUsbType type = BUsbTypeFullSpeed,
    BUsbConfig * usbConfig = 0 )
```

Initialise with optional config params.

### 5.75.2.3 `printf()`

```
int BUsbSerial::printf (
    const char * fmt,
    ... )
```

### 5.75.2.4 `printfv()`

```
int BUsbSerial::printfv (
    const char * fmt,
    va_list args )
```

### 5.75.2.5 `read()`

```
BError BUsbSerial::read (
    void * data,
    UInt32 nBytes,
    UInt32 & nRead ) [virtual]
```

Read the data.

Implements [BComms](#).

Reimplemented in [BUsbSerialLock](#).

### 5.75.2.6 `readAvailable()`

```
BUInt BUsbSerial::readAvailable ( ) [virtual]
```

How many bytes of read data is available.

Reimplemented from [BComms](#).

Reimplemented in [BUsbSerialLock](#).

#### 5.75.2.7 setBlocking()

```
void BUsbSerial::setBlocking (
    Bool on )
```

Enable blocking mode.

#### 5.75.2.8 setMaximumUsbChunk()

```
void BUsbSerial::setMaximumUsbChunk (
    BUInt usbChunkSize )
```

Sets the maximum USB transmitted chunk size.

#### 5.75.2.9 setSync()

```
void BUsbSerial::setSync (
    Bool on )
```

Set synchronous mode for writes.

#### 5.75.2.10 start()

```
void BUsbSerial::start ( )
```

Start processing.

#### 5.75.2.11 usbDeInit()

```
BUInt8 BUsbSerial::usbDeInit (
    USBD_HandleTypeDef * pdev,
    BUInt8 cfgidx )
```

#### 5.75.2.12 usbInit()

```
BUInt8 BUsbSerial::usbInit (
    USBD_HandleTypeDef * pdev,
    BUInt8 cfgidx )
```

### 5.75.2.13 usbdInitHardware()

```
BError BUsbSerial::usbdInitHardware ( )
```

### 5.75.2.14 usbdInitUsbDev()

```
BError BUsbSerial::usbdInitUsbDev (
    USBD_HandleTypeDef * pdev )
```

### 5.75.2.15 usbdInterrupt()

```
void BUsbSerial::usbdInterrupt (
    void )
```

### 5.75.2.16 usbdRx()

```
int BUsbSerial::usbdRx (
    USBD_HandleTypeDef * pdev,
    BUInt len )
```

Data from USB stack.

### 5.75.2.17 usbdSetup()

```
BUInt8 BUsbSerial::usbdSetup (
    USBD_HandleTypeDef * pdev,
    USBD_SetupReqTypedef * req )
```

### 5.75.2.18 usbdStarted()

```
void BUsbSerial::usbdStarted ( )
```

The USB device is connected and running.

#### 5.75.2.19 usbdTx()

```
void BUsbSerial::usbdTx ( )
```

Data to USB stack.

#### 5.75.2.20 usbdTxComplete()

```
BUInt8 BUsbSerial::usbdTxComplete ( )
```

Data has been sent.

#### 5.75.2.21 wait()

```
BError BUsbSerial::wait (
    BUInt8 events,
    BTimeout timeoutUs = BTimeoutForever,
    BUInt32 num = 1 )
```

#### 5.75.2.22 write()

```
BError BUsbSerial::write (
    const void * data,
    UInt32 nBytes,
    UInt32 & nWritten ) [virtual]
```

Write a set of data.

Implements [BComms](#).

Reimplemented in [BUsbSerialLock](#).

#### 5.75.2.23 writeAvailable()

```
BUInt BUsbSerial::writeAvailable ( ) [virtual]
```

How much write space is available.

Reimplemented from [BComms](#).

Reimplemented in [BUsbSerialLock](#).

### 5.75.3 Member Data Documentation

#### 5.75.3.1 oblocking

```
Bool BUsbSerial::oblocking [private]
```

Device set to block on writes.

#### 5.75.3.2 oconnected

```
Bool BUsbSerial::oconnected [private]
```

The USB is connected.

#### 5.75.3.3 ohighSpeed

```
Bool BUsbSerial::ohighSpeed [private]
```

High speed USB.

#### 5.75.3.4 ohpcd

```
PCD_HandleTypeDef BUsbSerial::ohpcd [private]
```

The USB PCD layer.

#### 5.75.3.5 oinitialised

```
Bool BUsbSerial::oinitialised [private]
```

The USB device has initialised.

### 5.75.3.6 opacketSize

```
BUInt BUsbSerial::opacketSize [private]
```

The maximum packet size.

### 5.75.3.7 orxBuffer

```
char BUsbSerial::orxBuffer[512] [private]
```

Receive packet buffer.

### 5.75.3.8 orxFifo

```
BFifo<char> BUsbSerial::orxFifo [private]
```

The receive fifo.

### 5.75.3.9 orxStalled

```
Bool BUsbSerial::orxStalled [private]
```

Receive has stalled as no room in buffer.

### 5.75.3.10 osync

```
Bool BUsbSerial::osync [private]
```

Device set to wait for writes.

### 5.75.3.11 otxFifo

```
BFifo<char> BUsbSerial::otxFifo [private]
```

The transmit fifo.

#### 5.75.3.12 otxSending

```
Bool BUsbSerial::otxSending [private]
```

USB is transmitting.

#### 5.75.3.13 otxSendLen

```
int BUsbSerial::otxSendLen [private]
```

The length of data sent.

#### 5.75.3.14 ousbChunkSize

```
BUInt BUsbSerial::ousbChunkSize [private]
```

The maximum USB chunk size.

#### 5.75.3.15 ousbDev

```
USBD_HandleTypeDef BUsbSerial::ousbDev [private]
```

The USB device.

#### 5.75.3.16 owait

```
BSemaphore BUsbSerial::owait [private]
```

Read wait semaphore.

#### 5.75.3.17 owaitReadNum

```
BUInt BUsbSerial::owaitReadNum [private]
```

The number of bytes to wait for.

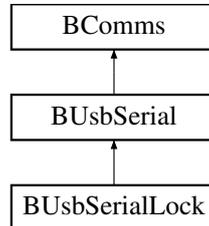
The documentation for this class was generated from the following files:

- [BUsbSerial.h](#)
- [BUsbSerial.cpp](#)

## 5.76 BUsbSerialLock Class Reference

```
#include <BUsbSerialLock.h>
```

Inheritance diagram for BUsbSerialLock:



### Public Member Functions

- [BUsbSerialLock](#) ([BUInt](#) rxFifoSize=512, [BUInt](#) txFifoSize=512)
- [int printfv](#) (const char \*fmt, va\_list args)
- [BUInt writeAvailable](#) ()  
*How much write space is available.*
- [BError write](#) (const void \*data, [BUInt32](#) nBytes, [UInt32](#) &nWritten)  
*Write a set of data.*
- [BUInt readAvailable](#) ()  
*How manu bytes of read data is available.*
- [BError read](#) (void \*data, [BUInt32](#) nBytes, [UInt32](#) &nRead)  
*Read the data.*

### Private Attributes

- [BMutex olock](#)

### Additional Inherited Members

#### 5.76.1 Constructor & Destructor Documentation

##### 5.76.1.1 BUsbSerialLock()

```
BUsbSerialLock::BUsbSerialLock (  
    BUInt rxFifoSize = 512,  
    BUInt txFifoSize = 512 )
```

#### 5.76.2 Member Function Documentation

### 5.76.2.1 printfv()

```
int BUsbSerialLock::printfv (
    const char * fmt,
    va_list args )
```

### 5.76.2.2 read()

```
BError BUsbSerialLock::read (
    void * data,
    BUInt32 nBytes,
    UInt32 & nRead ) [virtual]
```

Read the data.

Reimplemented from [BUsbSerial](#).

### 5.76.2.3 readAvailable()

```
BUInt BUsbSerialLock::readAvailable ( ) [virtual]
```

How manu bytes of read data is available.

Reimplemented from [BUsbSerial](#).

### 5.76.2.4 write()

```
BError BUsbSerialLock::write (
    const void * data,
    BUInt32 nBytes,
    UInt32 & nWritten ) [virtual]
```

Write a set of data.

Reimplemented from [BUsbSerial](#).

### 5.76.2.5 writeAvailable()

```
BUInt BUsbSerialLock::writeAvailable ( ) [virtual]
```

How much write space is available.

Reimplemented from [BUsbSerial](#).

### 5.76.3 Member Data Documentation

#### 5.76.3.1 olock

```
BMutex BUsbSerialLock::olock [private]
```

The documentation for this class was generated from the following files:

- [BUsbSerialLock.h](#)
- [BUsbSerialLock.cpp](#)

## 5.77 BWatchdog Class Reference

```
#include <BWatchdog.h>
```

### Public Member Functions

- [BWatchdog](#) ()
- void [start](#) ()  
*Start the watchdog timer.*
- void [reset](#) ()  
*Reset the timer.*

### 5.77.1 Constructor & Destructor Documentation

#### 5.77.1.1 BWatchdog()

```
BWatchdog::BWatchdog ( )
```

### 5.77.2 Member Function Documentation

#### 5.77.2.1 reset()

```
void BWatchdog::reset ( )
```

Reset the timer.

### 5.77.2.2 start()

```
void BWatchdog::start ( )
```

Start the watchdog timer.

The documentation for this class was generated from the following files:

- [BWatchdog.h](#)
- [BWatchdog.cpp](#)

## 5.78 HeapTestState Struct Reference

### Public Attributes

- [BUInt type](#)
- void \* [pos](#)
- [size\\_t size](#)
- void \* [caller](#)

### 5.78.1 Member Data Documentation

#### 5.78.1.1 caller

```
void* HeapTestState::caller
```

#### 5.78.1.2 pos

```
void* HeapTestState::pos
```

#### 5.78.1.3 size

```
size_t HeapTestState::size
```

## 5.78.1.4 type

```
BUInt HeapTestState::type
```

The documentation for this struct was generated from the following file:

- [BHeapTest.cpp](#)

## 5.79 IPAddress Class Reference

```
#include <BNetwork.h>
```

### Public Member Functions

- [IPAddress](#) (BUInt32 address=0)
- [IPAddress](#) (ip\_addr\_t address)
- [IPAddress](#) (BUInt8 a, UInt8 b, UInt8 c, UInt8 d)
- [BUInt32](#) getUInt ()
- [BString](#) getString ()
- void [setString](#) (BString address)
- [operator ip\\_addr\\_t](#) ()

### Private Attributes

- ip\_addr\_t [oaddress](#)

### 5.79.1 Constructor & Destructor Documentation

#### 5.79.1.1 [IPAddress\(\)](#) [1/3]

```
IPAddress::IPAddress (
    BUInt32 address = 0 )
```

#### 5.79.1.2 [IPAddress\(\)](#) [2/3]

```
IPAddress::IPAddress (
    ip_addr_t address )
```

### 5.79.1.3 IPAddress() [3/3]

```
IPAddress::IPAddress (
    BUInt8 a,
    UInt8 b,
    UInt8 c,
    UInt8 d )
```

## 5.79.2 Member Function Documentation

### 5.79.2.1 getString()

```
BString IPAddress::getString ( )
```

### 5.79.2.2 getUInt()

```
BUInt32 IPAddress::getUInt ( )
```

### 5.79.2.3 operator ip\_addr\_t()

```
IPAddress::operator ip_addr_t ( )
```

### 5.79.2.4 setString()

```
void IPAddress::setString (
    BString address )
```

## 5.79.3 Member Data Documentation

### 5.79.3.1 oaddress

```
ip_addr_t IPAddress::oaddress [private]
```

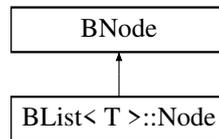
The documentation for this class was generated from the following files:

- [BNetwork.h](#)
- [BNetwork.cpp](#)

## 5.80 BList< T >::Node Class Reference

```
#include <BList.h>
```

Inheritance diagram for BList< T >::Node:



### Public Member Functions

- [Node](#) (const T &i)

### Public Attributes

- [T item](#)

### 5.80.1 Constructor & Destructor Documentation

#### 5.80.1.1 Node()

```
template<class T>  
BList< T >::Node::Node (  
    const T & i ) [inline]
```

### 5.80.2 Member Data Documentation

#### 5.80.2.1 item

```
template<class T>  
T BList< T >::Node::item
```

The documentation for this class was generated from the following file:

- [BList.h](#)



# Chapter 6

## File Documentation

### 6.1 BAdc.cpp File Reference

```
#include <BAdc.h>
#include <BSys.h>
#include <BGpio.h>
```

#### Macros

- #define [Mode\\_Independent](#) 0x00000000
- #define [DualMode\\_RegSimult\\_InjecSimult](#) 0x00000001
- #define [DualMode\\_RegSimult\\_AlterTrig](#) 0x00000002
- #define [DualMode\\_InjecSimult](#) 0x00000005
- #define [DualMode\\_RegSimult](#) 0x00000006
- #define [DualMode\\_Interl](#) 0x00000007
- #define [DualMode\\_AlterTrig](#) 0x00000009
- #define [TripleMode\\_RegSimult\\_InjecSimult](#) 0x00000011
- #define [TripleMode\\_RegSimult\\_AlterTrig](#) 0x00000012
- #define [TripleMode\\_InjecSimult](#) 0x00000015
- #define [TripleMode\\_RegSimult](#) 0x00000016
- #define [TripleMode\\_Interl](#) 0x00000017
- #define [TripleMode\\_AlterTrig](#) 0x00000019
- #define [Prescaler\\_Div2](#) 0x00000000
- #define [Prescaler\\_Div4](#) 0x00010000
- #define [Prescaler\\_Div6](#) 0x00020000
- #define [Prescaler\\_Div8](#) 0x00030000
- #define [SampleTime\\_3Cycles](#) 0x00
- #define [SampleTime\\_15Cycles](#) 0x01
- #define [SampleTime\\_28Cycles](#) 0x02
- #define [SampleTime\\_56Cycles](#) 0x03
- #define [SampleTime\\_84Cycles](#) 0x04
- #define [SampleTime\\_112Cycles](#) 0x05
- #define [SampleTime\\_144Cycles](#) 0x06
- #define [SampleTime\\_480Cycles](#) 0x07
- #define [TwoSamplingDelay\\_5Cycles](#) 0x00000000
- #define [TwoSamplingDelay\\_6Cycles](#) 0x00000100

- `#define TwoSamplingDelay_7Cycles 0x00000200`
- `#define TwoSamplingDelay_8Cycles 0x00000300`
- `#define TwoSamplingDelay_9Cycles 0x00000400`
- `#define TwoSamplingDelay_10Cycles 0x00000500`
- `#define TwoSamplingDelay_11Cycles 0x00000600`
- `#define TwoSamplingDelay_12Cycles 0x00000700`
- `#define TwoSamplingDelay_13Cycles 0x00000800`
- `#define TwoSamplingDelay_14Cycles 0x00000900`
- `#define TwoSamplingDelay_15Cycles 0x00000A00`
- `#define TwoSamplingDelay_16Cycles 0x00000B00`
- `#define TwoSamplingDelay_17Cycles 0x00000C00`
- `#define TwoSamplingDelay_18Cycles 0x00000D00`
- `#define TwoSamplingDelay_19Cycles 0x00000E00`
- `#define TwoSamplingDelay_20Cycles 0x00000F00`

## 6.1.1 Macro Definition Documentation

### 6.1.1.1 DualMode\_AlterTrig

```
#define DualMode_AlterTrig 0x00000009
```

### 6.1.1.2 DualMode\_InjecSimult

```
#define DualMode_InjecSimult 0x00000005
```

### 6.1.1.3 DualMode\_Interl

```
#define DualMode_Interl 0x00000007
```

### 6.1.1.4 DualMode\_RegSimult

```
#define DualMode_RegSimult 0x00000006
```

### 6.1.1.5 DualMode\_RegSimult\_AlterTrig

```
#define DualMode_RegSimult_AlterTrig 0x00000002
```

#### 6.1.1.6 DualMode\_RegSimult\_InjecSimult

```
#define DualMode_RegSimult_InjecSimult 0x00000001
```

#### 6.1.1.7 Mode\_Independent

```
#define Mode_Independent 0x00000000
```

#### 6.1.1.8 Prescaler\_Div2

```
#define Prescaler_Div2 0x00000000
```

#### 6.1.1.9 Prescaler\_Div4

```
#define Prescaler_Div4 0x00010000
```

#### 6.1.1.10 Prescaler\_Div6

```
#define Prescaler_Div6 0x00020000
```

#### 6.1.1.11 Prescaler\_Div8

```
#define Prescaler_Div8 0x00030000
```

#### 6.1.1.12 SampleTime\_112Cycles

```
#define SampleTime_112Cycles 0x05
```

#### 6.1.1.13 SampleTime\_144Cycles

```
#define SampleTime_144Cycles 0x06
```

**6.1.1.14 SampleTime\_15Cycles**

```
#define SampleTime_15Cycles 0x01
```

**6.1.1.15 SampleTime\_28Cycles**

```
#define SampleTime_28Cycles 0x02
```

**6.1.1.16 SampleTime\_3Cycles**

```
#define SampleTime_3Cycles 0x00
```

**6.1.1.17 SampleTime\_480Cycles**

```
#define SampleTime_480Cycles 0x07
```

**6.1.1.18 SampleTime\_56Cycles**

```
#define SampleTime_56Cycles 0x03
```

**6.1.1.19 SampleTime\_84Cycles**

```
#define SampleTime_84Cycles 0x04
```

**6.1.1.20 TripleMode\_AlterTrig**

```
#define TripleMode_AlterTrig 0x00000019
```

**6.1.1.21 TripleMode\_InjecSimult**

```
#define TripleMode_InjecSimult 0x00000015
```

### 6.1.1.22 TripleMode\_Interl

```
#define TripleMode_Interl 0x00000017
```

### 6.1.1.23 TripleMode\_RegSimult

```
#define TripleMode_RegSimult 0x00000016
```

### 6.1.1.24 TripleMode\_RegSimult\_AlterTrig

```
#define TripleMode_RegSimult_AlterTrig 0x00000012
```

### 6.1.1.25 TripleMode\_RegSimult\_InjecSimult

```
#define TripleMode_RegSimult_InjecSimult 0x00000011
```

### 6.1.1.26 TwoSamplingDelay\_10Cycles

```
#define TwoSamplingDelay_10Cycles 0x00000500
```

### 6.1.1.27 TwoSamplingDelay\_11Cycles

```
#define TwoSamplingDelay_11Cycles 0x00000600
```

### 6.1.1.28 TwoSamplingDelay\_12Cycles

```
#define TwoSamplingDelay_12Cycles 0x00000700
```

### 6.1.1.29 TwoSamplingDelay\_13Cycles

```
#define TwoSamplingDelay_13Cycles 0x00000800
```

**6.1.1.30 TwoSamplingDelay\_14Cycles**

```
#define TwoSamplingDelay_14Cycles 0x00000900
```

**6.1.1.31 TwoSamplingDelay\_15Cycles**

```
#define TwoSamplingDelay_15Cycles 0x00000A00
```

**6.1.1.32 TwoSamplingDelay\_16Cycles**

```
#define TwoSamplingDelay_16Cycles 0x00000B00
```

**6.1.1.33 TwoSamplingDelay\_17Cycles**

```
#define TwoSamplingDelay_17Cycles 0x00000C00
```

**6.1.1.34 TwoSamplingDelay\_18Cycles**

```
#define TwoSamplingDelay_18Cycles 0x00000D00
```

**6.1.1.35 TwoSamplingDelay\_19Cycles**

```
#define TwoSamplingDelay_19Cycles 0x00000E00
```

**6.1.1.36 TwoSamplingDelay\_20Cycles**

```
#define TwoSamplingDelay_20Cycles 0x00000F00
```

**6.1.1.37 TwoSamplingDelay\_5Cycles**

```
#define TwoSamplingDelay_5Cycles 0x00000000
```

### 6.1.1.38 TwoSamplingDelay\_6Cycles

```
#define TwoSamplingDelay_6Cycles 0x00000100
```

### 6.1.1.39 TwoSamplingDelay\_7Cycles

```
#define TwoSamplingDelay_7Cycles 0x00000200
```

### 6.1.1.40 TwoSamplingDelay\_8Cycles

```
#define TwoSamplingDelay_8Cycles 0x00000300
```

### 6.1.1.41 TwoSamplingDelay\_9Cycles

```
#define TwoSamplingDelay_9Cycles 0x00000400
```

## 6.2 BAdc.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <ArmSystem.h>
```

### Classes

- class [BAdc](#)

## 6.3 BArray.h File Reference

```
#include <BTypes.h>  
#include <BArray_imp.h>
```

### Classes

- class [BArray< T >](#)

## Typedefs

- typedef [BArray](#)< [Int32](#) > [BArrayInt32](#)
- typedef [BArray](#)< [BFloat32](#) > [BArrayFloat32](#)
- typedef [BArray](#)< [BFloat64](#) > [BArrayFloat64](#)

### 6.3.1 Typedef Documentation

#### 6.3.1.1 BArrayFloat32

```
typedef BArray<BFloat32> BArrayFloat32
```

#### 6.3.1.2 BArrayFloat64

```
typedef BArray<BFloat64> BArrayFloat64
```

#### 6.3.1.3 BArrayInt32

```
typedef BArray<Int32> BArrayInt32
```

## 6.4 BArray\_imp.h File Reference

```
#include <BArray.h>  
#include <string.h>
```

## 6.5 BBuffer.cpp File Reference

```
#include <stdlib.h>  
#include <string.h>  
#include <BBuffer.h>  
#include <BComplex.h>
```

## Variables

- const int [roundSize](#) = 256

## 6.5.1 Variable Documentation

### 6.5.1.1 roundSize

```
const int roundSize = 256
```

## 6.6 BBuffer.h File Reference

```
#include <BTypes.h>  
#include <BString.h>  
#include <BError.h>  
#include <BComplex.h>
```

### Classes

- class [BBuffer](#)
- class [BBufferFixed< Size >](#)
- class [BBufferStore](#)

## 6.7 BCan.cpp File Reference

```
#include <BCan.h>  
#include <BSys.h>  
#include <string.h>  
#include <BDebug.h>
```

### Macros

- `#define` [USE\\_INTERRUPTS](#) 1

### Functions

- void [CAN1\\_RX0\\_IRQHandler](#) ()
- void [CAN2\\_RX0\\_IRQHandler](#) ()

### Variables

- static [BCan](#) \* [devs](#) [2]

## 6.7.1 Macro Definition Documentation

### 6.7.1.1 USE\_INTERRUPTS

```
#define USE_INTERRUPTS 1
```

## 6.7.2 Function Documentation

### 6.7.2.1 CAN1\_RX0\_IRQHandler()

```
void CAN1_RX0_IRQHandler ( )
```

### 6.7.2.2 CAN2\_RX0\_IRQHandler()

```
void CAN2_RX0_IRQHandler ( )
```

## 6.7.3 Variable Documentation

### 6.7.3.1 devs

```
BCan* devs[2] [static]
```

## 6.8 BCan.h File Reference

```
#include <BComms.h>  
#include <BGpio.h>  
#include <BFifo.h>  
#include <ArmSystem.h>
```

### Classes

- class [BCanPacketFifo](#)
- class [BCan](#)

## 6.9 BComms.cpp File Reference

```
#include <BComms.h>
```

## 6.10 BComms.h File Reference

```
#include <BTypes.h>  
#include <BEvent.h>  
#include <BError.h>
```

### Classes

- class [BComms](#)

## 6.11 BComplex.h File Reference

```
#include <BTypes.h>  
#include <complex>  
#include <algorithm>
```

### Typedefs

- typedef std::complex< double > [BComplex](#)
- typedef std::complex< float > [BComplex32](#)
- typedef std::complex< double > [BComplex64](#)

### 6.11.1 Typedef Documentation

#### 6.11.1.1 BComplex

```
typedef std::complex<double> BComplex
```

This is based on the Standard C++ library complex class and has all of the functionality of that class.

#### 6.11.1.2 BComplex32

```
typedef std::complex<float> BComplex32
```

### 6.11.1.3 BComplex64

```
typedef std::complex<double> BComplex64
```

## 6.12 BCondInt.cpp File Reference

```
#include <BCondInt.h>
```

## 6.13 BCondInt.h File Reference

```
#include <BTypes.h>  
#include <BMutex.h>  
#include <BSemaphore.h>
```

### Classes

- class [BCondInt](#)  
*BCondInt* class.

## 6.14 BConfig.h File Reference

```
#include <BSys.h>  
#include <BGpio.h>
```

## 6.15 BConvert.cpp File Reference

```
#include <BConvert.h>
```

### Functions

- [BError base64Decode](#) (const [BString](#) strIn, [BString](#) &strOut)

### Variables

- static const [BUInt8 base64\\_decode\\_table](#) []



### 6.16.1.1 base64Decode()

```
BError base64Decode (
    const BString strIn,
    BString & strOut )
```

## 6.17 BCrc16.cpp File Reference

```
#include <BCrc16.h>
```

### Functions

- [BUInt16 bcrc16](#) (void \*buf, [BUInt16](#) len)

### Variables

- static const [BUInt8 table\\_crc\\_hi](#) []
- static const [BUInt8 table\\_crc\\_lo](#) []

## 6.17.1 Function Documentation

### 6.17.1.1 bcrc16()

```
BUInt16 bcrc16 (
    void * buf,
    BUInt16 len )
```

## 6.17.2 Variable Documentation

## 6.17.2.1 table\_crc\_hi

```
const BUInt8 table_crc_hi[] [static]
```

**Initial value:**

```
= {
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
}
```

## 6.17.2.2 table\_crc\_lo

```
const BUInt8 table_crc_lo[] [static]
```

**Initial value:**

```
= {
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
    0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
    0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
    0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
    0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
    0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
    0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
    0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
    0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
    0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
    0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
    0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
    0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
    0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
    0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
    0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
    0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
    0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
    0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x9A, 0x99, 0x5B, 0x5B,
    0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
    0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
    0x43, 0x83, 0x41, 0x81, 0x80, 0x40
}
```

## 6.18 BCrc16.h File Reference

```
#include <BTypes.h>
```

### Functions

- [BUInt16 bcrc16](#) (void \*buf, [BUInt16](#) len)

### 6.18.1 Function Documentation

#### 6.18.1.1 bcrc16()

```
BUInt16 bcrc16 (  
    void * buf,  
    BUInt16 len )
```

## 6.19 BCrc32.cpp File Reference

```
#include <BCrc32.h>
```

### Functions

- [BUInt32 bcrc32](#) ([BUInt32](#) crc, const void \*buf, [BUInt32](#) len)

### Variables

- static [BUInt32 crc32\\_tab](#) []

### 6.19.1 Function Documentation

#### 6.19.1.1 bcrc32()

```
BUInt32 bcrc32 (  
    BUInt32 crc,  
    const void * buf,  
    BUInt32 len )
```

## 6.19.2 Variable Documentation

### 6.19.2.1 crc32\_tab

```
BUInt32 crc32_tab[] [static]
```

## 6.20 BCrc32.h File Reference

```
#include <BTypes.h>
```

### Functions

- [BUInt32 bcrc32](#) ([BUInt32](#) crc, const void \*buf, [BUInt32](#) len)

### 6.20.1 Function Documentation

#### 6.20.1.1 bcrc32()

```
BUInt32 bcrc32 (  
    BUInt32 crc,  
    const void * buf,  
    BUInt32 len )
```

## 6.21 BCritical.h File Reference

```
#include <BSys.h>  
#include <BCritical.inc>
```

### Classes

- class [BCritical](#)  
*Critical region class.*
- class [BCriticalLock](#)  
*Critical region class with unlock on object deletion.*

## 6.22 BCritical.inc File Reference

## 6.23 BDebug.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <BDebug.h>
```

### Functions

- void [bhd8](#) (const void \**data*, unsigned int *n*)
- void [bhd8a](#) (const void \**data*, unsigned int *n*)
- void [bhda8](#) (const void \**data*, unsigned int *n*)
- void [bhd32](#) (const void \**data*, unsigned int *n*)
- void [bhda32](#) (const void \**data*, unsigned int *n*)

### 6.23.1 Function Documentation

#### 6.23.1.1 bhd32()

```
void bhd32 (
    const void * data,
    unsigned int n )
```

#### 6.23.1.2 bhd8()

```
void bhd8 (
    const void * data,
    unsigned int n )
```

#### 6.23.1.3 bhd8a()

```
void bhd8a (
    const void * data,
    unsigned int n )
```

#### 6.23.1.4 bhda32()

```
void bhda32 (
    const void * data,
    unsigned int n )
```

#### 6.23.1.5 bhda8()

```
void bhda8 (
    const void * data,
    unsigned int n )
```

## 6.24 BDebug.h File Reference

```
#include <stdio.h>
```

### Macros

- #define [dprintf](#)(fmt, a...)  
*General debug functions.*
- #define [dl1printf](#)(fmt, a...)
- #define [dl2printf](#)(fmt, a...)
- #define [dl3printf](#)(fmt, a...)
- #define [dl4printf](#)(fmt, a...)

### Functions

- void [bhd8](#) (const void \*[data](#), unsigned int n)
- void [bhd8a](#) (const void \*[data](#), unsigned int n)
- void [bhda8](#) (const void \*[data](#), unsigned int n)
- void [bhd32](#) (const void \*[data](#), unsigned int n)
- void [bhds32](#) (const void \*[data](#), unsigned int n)

### 6.24.1 Macro Definition Documentation

#### 6.24.1.1 dl1printf

```
#define dl1printf(  
    fmt,  
    a... )
```

### 6.24.1.2 dl2printf

```
#define dl2printf(  
    fmt,  
    a... )
```

### 6.24.1.3 dl3printf

```
#define dl3printf(  
    fmt,  
    a... )
```

### 6.24.1.4 dl4printf

```
#define dl4printf(  
    fmt,  
    a... )
```

### 6.24.1.5 dprintf

```
#define dprintf(  
    fmt,  
    a... )
```

General debug functions.

## 6.24.2 Function Documentation

### 6.24.2.1 bhd32()

```
void bhd32 (  
    const void * data,  
    unsigned int n )
```

#### 6.24.2.2 bhd8()

```
void bhd8 (
    const void * data,
    unsigned int n )
```

#### 6.24.2.3 bhd8a()

```
void bhd8a (
    const void * data,
    unsigned int n )
```

#### 6.24.2.4 bhda8()

```
void bhda8 (
    const void * data,
    unsigned int n )
```

#### 6.24.2.5 bhds32()

```
void bhds32 (
    const void * data,
    unsigned int n )
```

## 6.25 BDevices.cpp File Reference

```
#include <BDevices.h>
#include <BDebug.h>
#include <ArmSystem.h>
```

### Macros

- #define [LDEBUG](#) 0
- #define [dlprintf](#)(fmt, a...)

### Variables

- const [BUInt](#) [HSI\\_CLOCK](#) = 16000000

## 6.25.1 Macro Definition Documentation

### 6.25.1.1 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

### 6.25.1.2 LDEBUG

```
#define LDEBUG 0
```

## 6.25.2 Variable Documentation

### 6.25.2.1 HSI\_CLOCK

```
const BUInt HSI_CLOCK = 16000000
```

## 6.26 BDevices.h File Reference

```
#include <BTypes.h>  
#include <BError.h>
```

### Classes

- class [BClocks](#)
- class [BDevices](#)

## Enumerations

- enum `Device` {
  - `DeviceGPIOA`, `DeviceGPIOB`, `DeviceGPIOC`, `DeviceGPIOD`,
  - `DeviceGPIOE`, `DeviceGPIOF`, `DeviceGPIOG`, `DeviceGPIOH`,
  - `DeviceGPIOI`, `DeviceGPIOJ`, `DeviceCRC`, `DeviceFLITF`,
  - `DeviceSRAM1`, `DeviceSRAM2`, `DeviceBKPSRAM`, `DeviceCCMDATARAM`,
  - `DeviceDMA1`, `DeviceDMA2`, `DeviceETH_MAC`, `DeviceETH_MAC_Tx`,
  - `DeviceETH_MAC_Rx`, `DeviceETH_MAC_PTP`, `DeviceOTG_HS`, `DeviceOTG_HS_ULPI`,
  - `DeviceDCMI`, `DeviceCRYP`, `DeviceHASH`, `DeviceRNG`,
  - `DeviceOTG_FS`, `DeviceFSMC`, `DeviceTIM2`, `DeviceTIM3`,
  - `DeviceTIM4`, `DeviceTIM5`, `DeviceTIM6`, `DeviceTIM7`,
  - `DeviceTIM12`, `DeviceTIM13`, `DeviceTIM14`, `DeviceTIM15`,
  - `DeviceTIM16`, `DeviceTIM17`, `DeviceTIM18`, `DeviceWWDG`,
  - `DeviceSPI2`, `DeviceSPI3`, `DeviceUSART2`, `DeviceUSART3`,
  - `DeviceUART4`, `DeviceUART5`, `DeviceI2C1`, `DeviceI2C2`,
  - `DeviceI2C3`, `DeviceCAN1`, `DeviceCAN2`, `DevicePWR`,
  - `DeviceDAC`, `DeviceDAC1`, `DeviceDAC2`, `DeviceTIM1`,
  - `DeviceTIM8`, `DeviceUSART1`, `DeviceUSART6`, `DeviceADC1`,
  - `DeviceADC2`, `DeviceADC3`, `DeviceSDIO`, `DeviceSDMMC1`,
  - `DeviceSDMMC2`, `DeviceSPI1`, `DeviceSYSCFG`, `DeviceTIM9`,
  - `DeviceTIM10`, `DeviceTIM11`, `DeviceSPI4`, `DeviceSPI5`,
  - `DeviceSPI6`, `DeviceSDADC1`, `DeviceSDADC2`, `DeviceSDADC3`,
  - `DeviceSAI1`, `DeviceSAI2`, `DeviceQSPI1`, `DeviceTSC1`,
  - `DeviceRTC` }

### 6.26.1 Enumeration Type Documentation

#### 6.26.1.1 Device

enum `Device`

##### Enumerator

|                               |  |
|-------------------------------|--|
| <code>DeviceGPIOA</code>      |  |
| <code>DeviceGPIOB</code>      |  |
| <code>DeviceGPIOC</code>      |  |
| <code>DeviceGPIOD</code>      |  |
| <code>DeviceGPIOE</code>      |  |
| <code>DeviceGPIOF</code>      |  |
| <code>DeviceGPIOG</code>      |  |
| <code>DeviceGPIOH</code>      |  |
| <code>DeviceGPIOI</code>      |  |
| <code>DeviceGPIOJ</code>      |  |
| <code>DeviceCRC</code>        |  |
| <code>DeviceFLITF</code>      |  |
| <code>DeviceSRAM1</code>      |  |
| <code>DeviceSRAM2</code>      |  |
| <code>DeviceBKPSRAM</code>    |  |
| <code>DeviceCCMDATARAM</code> |  |
| <code>DeviceDMA1</code>       |  |

## Enumerator

|                   |  |
|-------------------|--|
| DeviceDMA2        |  |
| DeviceETH_MAC     |  |
| DeviceETH_MAC_Tx  |  |
| DeviceETH_MAC_Rx  |  |
| DeviceETH_MAC_PTP |  |
| DeviceOTG_HS      |  |
| DeviceOTG_HS_ULPI |  |
| DeviceDCMI        |  |
| DeviceCRYP        |  |
| DeviceHASH        |  |
| DeviceRNG         |  |
| DeviceOTG_FS      |  |
| DeviceFSMC        |  |
| DeviceTIM2        |  |
| DeviceTIM3        |  |
| DeviceTIM4        |  |
| DeviceTIM5        |  |
| DeviceTIM6        |  |
| DeviceTIM7        |  |
| DeviceTIM12       |  |
| DeviceTIM13       |  |
| DeviceTIM14       |  |
| DeviceTIM15       |  |
| DeviceTIM16       |  |
| DeviceTIM17       |  |
| DeviceTIM18       |  |
| DeviceWWDG        |  |
| DeviceSPI2        |  |
| DeviceSPI3        |  |
| DeviceUSART2      |  |
| DeviceUSART3      |  |
| DeviceUART4       |  |
| DeviceUART5       |  |
| DeviceI2C1        |  |
| DeviceI2C2        |  |
| DeviceI2C3        |  |
| DeviceCAN1        |  |
| DeviceCAN2        |  |
| DevicePWR         |  |
| DeviceDAC         |  |
| DeviceDAC1        |  |
| DeviceDAC2        |  |
| DeviceTIM1        |  |
| DeviceTIM8        |  |
| DeviceUSART1      |  |
| DeviceUSART6      |  |
| DeviceADC1        |  |
| DeviceADC2        |  |
| DeviceADC3        |  |
| DeviceSDIO        |  |

## Enumerator

|              |  |
|--------------|--|
| DeviceSDMMC1 |  |
| DeviceSDMMC2 |  |
| DeviceSPI1   |  |
| DeviceSYSCFG |  |
| DeviceTIM9   |  |
| DeviceTIM10  |  |
| DeviceTIM11  |  |
| DeviceSPI4   |  |
| DeviceSPI5   |  |
| DeviceSPI6   |  |
| DeviceSDADC1 |  |
| DeviceSDADC2 |  |
| DeviceSDADC3 |  |
| DeviceSAI1   |  |
| DeviceSAI2   |  |
| DeviceQSPI1  |  |
| DeviceTSC1   |  |
| DeviceRTC    |  |

## 6.27 BDict.cpp File Reference

```
#include <BDict.h>
```

### Functions

- void [toBString](#) (const [BDictString](#) &v, [BString](#) &s)
- void [fromBString](#) (const [BString](#) &str, [BDictString](#) &v)
- [BString](#) [bdictStringToString](#) (const [BDictString](#) &dict)

### 6.27.1 Function Documentation

#### 6.27.1.1 [bdictStringToString\(\)](#)

```
BString bdictStringToString (  
    const BDictString & dict )
```

### 6.27.1.2 fromBString()

```
void fromBString (
    const BString & str,
    BDictString & v )
```

### 6.27.1.3 toBString()

```
void toBString (
    const BDictString & v,
    BString & s )
```

## 6.28 BDict.h File Reference

```
#include <BNameValue.h>
#include <BList.h>
#include <BArray.h>
```

### Classes

- class [BDictItem< Type >](#)  
*Template based Dictionary class.*
- class [BDict< Type >](#)

### Typedefs

- typedef [BDict< BString >](#) [BDictString](#)

### Functions

- void [toBString](#) (const [BDictString](#) &v, [BString](#) &s)
- void [fromBString](#) (const [BString](#) &s, [BDictString](#) &v)
- [BString](#) [bdictStringToString](#) (const [BDictString](#) &dict)

## 6.28.1 Typedef Documentation

### 6.28.1.1 BDictString

```
typedef BDict<BString> BDictString
```

## 6.28.2 Function Documentation

### 6.28.2.1 bdictStringToString()

```
BString bdictStringToString (  
    const BDictString & dict )
```

### 6.28.2.2 fromBString()

```
void fromBString (  
    const BString & s,  
    BDictString & v )
```

### 6.28.2.3 toBString()

```
void toBString (  
    const BDictString & v,  
    BString & s )
```

## 6.29 BDma.cpp File Reference

```
#include <BDma.h>  
#include <BSys.h>  
#include <BDebug.h>
```

### Macros

- #define `LDEBUG` 0
- #define `dlprintf`(fmt, a...)
- #define `dlprintStatus`()

## Functions

- void [DMA1\\_Stream0\\_IRQHandler](#) ()
- void [DMA1\\_Stream1\\_IRQHandler](#) ()
- void [DMA1\\_Stream2\\_IRQHandler](#) ()
- void [DMA1\\_Stream3\\_IRQHandler](#) ()
- void [DMA1\\_Stream4\\_IRQHandler](#) ()
- void [DMA1\\_Stream5\\_IRQHandler](#) ()
- void [DMA1\\_Stream6\\_IRQHandler](#) ()
- void [DMA1\\_Stream7\\_IRQHandler](#) ()
- void [DMA2\\_Stream0\\_IRQHandler](#) ()
- void [DMA2\\_Stream1\\_IRQHandler](#) ()
- void [DMA2\\_Stream2\\_IRQHandler](#) ()
- void [DMA2\\_Stream3\\_IRQHandler](#) ()
- void [DMA2\\_Stream4\\_IRQHandler](#) ()
- void [DMA2\\_Stream5\\_IRQHandler](#) ()
- void [DMA2\\_Stream6\\_IRQHandler](#) ()
- void [DMA2\\_Stream7\\_IRQHandler](#) ()

## Variables

- static [BDma](#) \* [dmaStreams](#) [16]

## 6.29.1 Macro Definition Documentation

### 6.29.1.1 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

### 6.29.1.2 dlprintStatus

```
#define dlprintStatus( )
```

### 6.29.1.3 LDEBUG

```
#define LDEBUG 0
```

## 6.29.2 Function Documentation

**6.29.2.1 DMA1\_Stream0\_IRQHandler()**

```
void DMA1_Stream0_IRQHandler ( )
```

**6.29.2.2 DMA1\_Stream1\_IRQHandler()**

```
void DMA1_Stream1_IRQHandler ( )
```

**6.29.2.3 DMA1\_Stream2\_IRQHandler()**

```
void DMA1_Stream2_IRQHandler ( )
```

**6.29.2.4 DMA1\_Stream3\_IRQHandler()**

```
void DMA1_Stream3_IRQHandler ( )
```

**6.29.2.5 DMA1\_Stream4\_IRQHandler()**

```
void DMA1_Stream4_IRQHandler ( )
```

**6.29.2.6 DMA1\_Stream5\_IRQHandler()**

```
void DMA1_Stream5_IRQHandler ( )
```

**6.29.2.7 DMA1\_Stream6\_IRQHandler()**

```
void DMA1_Stream6_IRQHandler ( )
```

**6.29.2.8 DMA1\_Stream7\_IRQHandler()**

```
void DMA1_Stream7_IRQHandler ( )
```

**6.29.2.9 DMA2\_Stream0\_IRQHandler()**

```
void DMA2_Stream0_IRQHandler ( )
```

**6.29.2.10 DMA2\_Stream1\_IRQHandler()**

```
void DMA2_Stream1_IRQHandler ( )
```

**6.29.2.11 DMA2\_Stream2\_IRQHandler()**

```
void DMA2_Stream2_IRQHandler ( )
```

**6.29.2.12 DMA2\_Stream3\_IRQHandler()**

```
void DMA2_Stream3_IRQHandler ( )
```

**6.29.2.13 DMA2\_Stream4\_IRQHandler()**

```
void DMA2_Stream4_IRQHandler ( )
```

**6.29.2.14 DMA2\_Stream5\_IRQHandler()**

```
void DMA2_Stream5_IRQHandler ( )
```

**6.29.2.15 DMA2\_Stream6\_IRQHandler()**

```
void DMA2_Stream6_IRQHandler ( )
```

**6.29.2.16 DMA2\_Stream7\_IRQHandler()**

```
void DMA2_Stream7_IRQHandler ( )
```

## 6.29.3 Variable Documentation

### 6.29.3.1 dmaStreams

```
BDma* dmaStreams[16] [static]
```

## 6.30 BDma.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BSemaphore.h>  
#include <ArmSystem.h>
```

### Classes

- class [BDma](#)

### Typedefs

- typedef void(\* [BDmaCompleteCallback](#)) ()

## 6.30.1 Typedef Documentation

### 6.30.1.1 BDmaCompleteCallback

```
typedef void(* BDmaCompleteCallback) ()
```

## 6.31 BEndian.cpp File Reference

```
#include <BEndian.h>  
#include <string.h>
```

### Functions

- void [bswap\\_copy](#) (int swap, const void \*src, void \*dst, [BUInt32](#) nBytes, const char \*swapType)

## 6.31.1 Function Documentation

### 6.31.1.1 bswap\_copy()

```
void bswap_copy (
    int swap,
    const void * src,
    void * dst,
    BUInt32 nBytes,
    const char * swapType )
```

## 6.32 BEndian.h File Reference

```
#include <BTypes.h>
```

### Functions

- void [bswap\\_p8](#) (const void \*s, void \*d)
- void [bswap\\_p16](#) (const void \*s, void \*d)
- void [bswap\\_p32](#) (const void \*s, void \*d)
- void [bswap\\_p64](#) (const void \*s, void \*d)
- void [bswap\\_copy](#) (int swap, const void \*src, void \*dst, BUInt32 nBytes, const char \*swapType)

## 6.32.1 Function Documentation

### 6.32.1.1 bswap\_copy()

```
void bswap_copy (
    int swap,
    const void * src,
    void * dst,
    BUInt32 nBytes,
    const char * swapType )
```

### 6.32.1.2 bswap\_p16()

```
void bswap_p16 (
    const void * s,
    void * d ) [inline]
```

### 6.32.1.3 bswap\_p32()

```
void bswap_p32 (
    const void * s,
    void * d ) [inline]
```

### 6.32.1.4 bswap\_p64()

```
void bswap_p64 (
    const void * s,
    void * d ) [inline]
```

### 6.32.1.5 bswap\_p8()

```
void bswap_p8 (
    const void * s,
    void * d ) [inline]
```

## 6.33 BError.cpp File Reference

```
#include <BError.h>
```

## 6.34 BError.h File Reference

```
#include <BTypes.h>
```

### Classes

- class [BError](#)

### Enumerations

- enum [BErrorNum](#) {  
[ErrorOk](#) = 0, [ErrorMisc](#) = 1, [ErrorWarning](#) = 2, [ErrorParam](#) = 3,  
[ErrorTimeout](#) = 4, [ErrorNotAvailable](#) = 5, [ErrorData](#) = 6, [ErrorChecksum](#) = 7,  
[ErrorOverrun](#) = 8, [ErrorUnderrun](#) = 9, [ErrorInit](#) = 10, [ErrorConfig](#) = 11,  
[ErrorNotImplemented](#) = 12, [ErrorResourceLimit](#) = 13, [ErrorEndOfFile](#) = 14, [ErrorFile](#) = 15,  
[ErrorFormat](#) = 16, [ErrorComms](#) = 17, [ErrorAccessDenied](#) = 18, [ErrorNoData](#) = 19,  
[ErrorEndOfData](#) = 20, [ErrorDataPresent](#) = 21, [ErrorDataTruncated](#) = 22, [ErrorAppBase](#) = 64 }

### 6.34.1 Enumeration Type Documentation

#### 6.34.1.1 BErrorNum

```
enum BErrorNum
```

### Enumerator

|                     |  |
|---------------------|--|
| ErrorOk             |  |
| ErrorMisc           |  |
| ErrorWarning        |  |
| ErrorParam          |  |
| ErrorTimeout        |  |
| ErrorNotAvailable   |  |
| ErrorData           |  |
| ErrorChecksum       |  |
| ErrorOverrun        |  |
| ErrorUnderrun       |  |
| ErrorInit           |  |
| ErrorConfig         |  |
| ErrorNotImplemented |  |
| ErrorResourceLimit  |  |
| ErrorEndOfFile      |  |
| ErrorFile           |  |
| ErrorFormat         |  |
| ErrorComms          |  |
| ErrorAccessDenied   |  |
| ErrorNoData         |  |
| ErrorEndOfData      |  |
| ErrorDataPresent    |  |
| ErrorDataTruncated  |  |
| ErrorAppBase        |  |

## 6.35 BEvent.cpp File Reference

```
#include <BEvent.h>
```

## 6.36 BEvent.h File Reference

```
#include <BTypes.h>  
#include <BQueue.h>
```

### Classes

- class [BEvent](#)

### Typedefs

- typedef [BQueue](#)< [BEvent](#) > [BEventQueue](#)

## 6.36.1 Typedef Documentation

### 6.36.1.1 BEventQueue

```
typedef BQueue<BEvent> BEventQueue
```

## 6.37 BFifo.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BFifo.inc>
```

### Classes

- class [BFifo< Type >](#)

## 6.38 BFifo.inc File Reference

## 6.39 BFirmware.cpp File Reference

```
#include <BFirmware.h>  
#include <BString.h>  
#include <BCrc32.h>  
#include <BSys.h>
```

### Macros

- #define [BDEBUGL1](#) 0

### Functions

- int [bfirmwareValid](#) (BUInt32 baseAddress, BUInt type, Bool checkChecksum, char \*version)
- void [bfirmwareBoot](#) (BUInt32 baseAddress)

## 6.39.1 Macro Definition Documentation

### 6.39.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

## 6.39.2 Function Documentation

### 6.39.2.1 bfirmwareBoot()

```
void bfirmwareBoot (  
    BUInt32 baseAddress )
```

### 6.39.2.2 bfirmwareValid()

```
int bfirmwareValid (  
    BUInt32 baseAddress,  
    BUInt type,  
    Bool checkChecksum,  
    char * version )
```

## 6.40 BFirmware.h File Reference

```
#include <BTypes.h>
```

### Classes

- struct [BFirmwareInfo](#)

### Functions

- int [bfirmwareValid](#) (BUInt32 baseAddress, BUInt type, Bool checkChecksum, char \*version=0)
- void [bfirmwareBoot](#) (BUInt32 baseAddress)

### Variables

- const BUInt32 [BFirmwareInfoMagic](#) = 0xBBEEAA00
- const BUInt8 [BFirmwareInfoEncrypt1](#) = 0x40

### 6.40.1 Function Documentation

### 6.40.1.1 bfirmwareBoot()

```
void bfirmwareBoot (
    BUInt32 baseAddress )
```

### 6.40.1.2 bfirmwareValid()

```
int bfirmwareValid (
    BUInt32 baseAddress,
    BUInt type,
    Bool checkChecksum,
    char * version = 0 )
```

## 6.40.2 Variable Documentation

### 6.40.2.1 BFirmwareInfoEncrypt1

```
const BUInt8 BFirmwareInfoEncrypt1 = 0x40
```

### 6.40.2.2 BFirmwareInfoMagic

```
const BUInt32 BFirmwareInfoMagic = 0xBBEEAA00
```

## 6.41 BFlash.cpp File Reference

```
#include <BFlash.h>
#include <BSys.h>
#include <ArmSystem.h>
#include <BDebug.h>
```

### Macros

- #define LDEBUG 0
- #define dlprintf(fmt, a...)
- #define RDP\_KEY ((uint16\_t)0x00A5)
- #define FLASH\_KEY1 ((uint32\_t)0x45670123)
- #define FLASH\_KEY2 ((uint32\_t)0xCDEF89AB)
- #define FLASH\_OPT\_KEY1 ((uint32\_t)0x08192A3B)
- #define FLASH\_OPT\_KEY2 ((uint32\_t)0x4C5D6E7F)

## 6.41.1 Macro Definition Documentation

### 6.41.1.1 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

### 6.41.1.2 FLASH\_KEY1

```
#define FLASH_KEY1 ((uint32_t)0x45670123)
```

### 6.41.1.3 FLASH\_KEY2

```
#define FLASH_KEY2 ((uint32_t)0xCDEF89AB)
```

### 6.41.1.4 FLASH\_OPT\_KEY1

```
#define FLASH_OPT_KEY1 ((uint32_t)0x08192A3B)
```

### 6.41.1.5 FLASH\_OPT\_KEY2

```
#define FLASH_OPT_KEY2 ((uint32_t)0x4C5D6E7F)
```

### 6.41.1.6 LDEBUG

```
#define LDEBUG 0
```

### 6.41.1.7 RDP\_KEY

```
#define RDP_KEY ((uint16_t)0x00A5)
```

## 6.42 BFlash.h File Reference

```
#include <BTypes.h>
#include <BError.h>
```

### Classes

- class [BFlash](#)

## 6.43 BGpio.cpp File Reference

```
#include <BGpio.h>
#include <BSys.h>
#include <stdio.h>
#include <ArmSystem.h>
```

### Functions

- static GPIO\_TypeDef \* [bank](#) ([Pin](#) pin)

### 6.43.1 Function Documentation

#### 6.43.1.1 [bank\(\)](#)

```
static GPIO_TypeDef* bank (
    Pin pin ) [inline], [static]
```

## 6.44 BGpio.h File Reference

```
#include <BTypes.h>
```

### Classes

- class [BGpio](#)

## Enumerations

- enum `Pin` {
  - `PinNone` = 0, `PinA0` =0x00001, `PinA1` =0x00002, `PinA2` =0x00004,
  - `PinA3` =0x00008, `PinA4` =0x00010, `PinA5` =0x00020, `PinA6` =0x00040,
  - `PinA7` =0x00080, `PinA8` =0x00100, `PinA9` =0x00200, `PinA10` =0x00400,
  - `PinA11` =0x00800, `PinA12` =0x01000, `PinA13` =0x02000, `PinA14` =0x04000,
  - `PinA15` =0x08000, `PinB0` =0x10001, `PinB1` =0x10002, `PinB2` =0x10004,
  - `PinB3` =0x10008, `PinB4` =0x10010, `PinB5` =0x10020, `PinB6` =0x10040,
  - `PinB7` =0x10080, `PinB8` =0x10100, `PinB9` =0x10200, `PinB10` =0x10400,
  - `PinB11` =0x10800, `PinB12` =0x11000, `PinB13` =0x12000, `PinB14` =0x14000,
  - `PinB15` =0x18000, `PinC0` =0x20001, `PinC1` =0x20002, `PinC2` =0x20004,
  - `PinC3` =0x20008, `PinC4` =0x20010, `PinC5` =0x20020, `PinC6` =0x20040,
  - `PinC7` =0x20080, `PinC8` =0x20100, `PinC9` =0x20200, `PinC10` =0x20400,
  - `PinC11` =0x20800, `PinC12` =0x21000, `PinC13` =0x22000, `PinC14` =0x24000,
  - `PinC15` =0x28000, `PinD0` =0x30001, `PinD1` =0x30002, `PinD2` =0x30004,
  - `PinD3` =0x30008, `PinD4` =0x30010, `PinD5` =0x30020, `PinD6` =0x30040,
  - `PinD7` =0x30080, `PinD8` =0x30100, `PinD9` =0x30200, `PinD10` =0x30400,
  - `PinD11` =0x30800, `PinD12` =0x31000, `PinD13` =0x32000, `PinD14` =0x34000,
  - `PinD15` =0x38000, `PinE0` =0x40001, `PinE1` =0x40002, `PinE2` =0x40004,
  - `PinE3` =0x40008, `PinE4` =0x40010, `PinE5` =0x40020, `PinE6` =0x40040,
  - `PinE7` =0x40080, `PinE8` =0x40100, `PinE9` =0x40200, `PinE10` =0x40400,
  - `PinE11` =0x40800, `PinE12` =0x41000, `PinE13` =0x42000, `PinE14` =0x44000,
  - `PinE15` =0x48000, `PinF0` =0x50001, `PinF1` =0x50002, `PinF2` =0x50004,
  - `PinF3` =0x50008, `PinF4` =0x50010, `PinF5` =0x50020, `PinF6` =0x50040,
  - `PinF7` =0x50080, `PinF8` =0x50100, `PinF9` =0x50200, `PinF10` =0x50400,
  - `PinF11` =0x50800, `PinF12` =0x51000, `PinF13` =0x52000, `PinF14` =0x54000,
  - `PinF15` =0x58000, `PinG0` =0x60001, `PinG1` =0x60002, `PinG2` =0x60004,
  - `PinG3` =0x60008, `PinG4` =0x60010, `PinG5` =0x60020, `PinG6` =0x60040,
  - `PinG7` =0x60080, `PinG8` =0x60100, `PinG9` =0x60200, `PinG10` =0x60400,
  - `PinG11` =0x60800, `PinG12` =0x61000, `PinG13` =0x62000, `PinG14` =0x64000,
  - `PinG15` =0x68000, `PinH0` =0x70001, `PinH1` =0x70002, `PinH2` =0x70004,
  - `PinH3` =0x70008, `PinH4` =0x70010, `PinH5` =0x70020, `PinH6` =0x70040,
  - `PinH7` =0x70080, `PinH8` =0x70100, `PinH9` =0x70200, `PinH10` =0x70400,
  - `PinH11` =0x70800, `PinH12` =0x71000, `PinH13` =0x72000, `PinH14` =0x74000,
  - `PinH15` =0x78000, `PinI0` =0x80001, `PinI1` =0x80002, `PinI2` =0x80004,
  - `PinI3` =0x80008, `PinI4` =0x80010, `PinI5` =0x80020, `PinI6` =0x80040,
  - `PinI7` =0x80080, `PinI8` =0x80100, `PinI9` =0x80200, `PinI10` =0x80400,
  - `PinI11` =0x80800, `PinI12` =0x81000, `PinI13` =0x82000, `PinI14` =0x84000,
  - `PinI15` =0x88000, `PinJ0` =0x90001, `PinJ1` =0x90002, `PinJ2` =0x90004,
  - `PinJ3` =0x90008, `PinJ4` =0x90010, `PinJ5` =0x90020, `PinJ6` =0x90040,
  - `PinJ7` =0x90080, `PinJ8` =0x90100, `PinJ9` =0x90200, `PinJ10` =0x90400,
  - `PinJ11` =0x90800, `PinJ12` =0x91000, `PinJ13` =0x92000, `PinJ14` =0x94000,
  - `PinJ15` =0x98000 }
- enum `PinState` { `Off` = 0, `On` = 1 }

## Functions

- void `bitsSet` (volatile `BUInt8` &reg, `BUInt32` bitPos, `BUInt32` bitMask, `BUInt32` value)
- void `bitsSet` (volatile `BUInt16` &reg, `BUInt32` bitPos, `BUInt32` bitMask, `BUInt32` value)
- void `bitsSet` (volatile `BUInt32` &reg, `BUInt32` bitPos, `BUInt32` bitMask, `BUInt32` value)

## Variables

- `BGpio gpio`

## 6.44.1 Enumeration Type Documentation

### 6.44.1.1 Pin

enum `Pin`

#### Enumerator

|         |  |
|---------|--|
| PinNone |  |
| PinA0   |  |
| PinA1   |  |
| PinA2   |  |
| PinA3   |  |
| PinA4   |  |
| PinA5   |  |
| PinA6   |  |
| PinA7   |  |
| PinA8   |  |
| PinA9   |  |
| PinA10  |  |
| PinA11  |  |
| PinA12  |  |
| PinA13  |  |
| PinA14  |  |
| PinA15  |  |
| PinB0   |  |
| PinB1   |  |
| PinB2   |  |
| PinB3   |  |
| PinB4   |  |
| PinB5   |  |
| PinB6   |  |
| PinB7   |  |
| PinB8   |  |
| PinB9   |  |
| PinB10  |  |
| PinB11  |  |
| PinB12  |  |
| PinB13  |  |
| PinB14  |  |
| PinB15  |  |
| PinC0   |  |
| PinC1   |  |
| PinC2   |  |
| PinC3   |  |
| PinC4   |  |
| PinC5   |  |
| PinC6   |  |
| PinC7   |  |
| PinC8   |  |

## Enumerator

|        |  |
|--------|--|
| PinC9  |  |
| PinC10 |  |
| PinC11 |  |
| PinC12 |  |
| PinC13 |  |
| PinC14 |  |
| PinC15 |  |
| PinD0  |  |
| PinD1  |  |
| PinD2  |  |
| PinD3  |  |
| PinD4  |  |
| PinD5  |  |
| PinD6  |  |
| PinD7  |  |
| PinD8  |  |
| PinD9  |  |
| PinD10 |  |
| PinD11 |  |
| PinD12 |  |
| PinD13 |  |
| PinD14 |  |
| PinD15 |  |
| PinE0  |  |
| PinE1  |  |
| PinE2  |  |
| PinE3  |  |
| PinE4  |  |
| PinE5  |  |
| PinE6  |  |
| PinE7  |  |
| PinE8  |  |
| PinE9  |  |
| PinE10 |  |
| PinE11 |  |
| PinE12 |  |
| PinE13 |  |
| PinE14 |  |
| PinE15 |  |
| PinF0  |  |
| PinF1  |  |
| PinF2  |  |
| PinF3  |  |
| PinF4  |  |
| PinF5  |  |
| PinF6  |  |
| PinF7  |  |
| PinF8  |  |
| PinF9  |  |
| PinF10 |  |
| PinF11 |  |

## Enumerator

|        |  |
|--------|--|
| PinF12 |  |
| PinF13 |  |
| PinF14 |  |
| PinF15 |  |
| PinG0  |  |
| PinG1  |  |
| PinG2  |  |
| PinG3  |  |
| PinG4  |  |
| PinG5  |  |
| PinG6  |  |
| PinG7  |  |
| PinG8  |  |
| PinG9  |  |
| PinG10 |  |
| PinG11 |  |
| PinG12 |  |
| PinG13 |  |
| PinG14 |  |
| PinG15 |  |
| PinH0  |  |
| PinH1  |  |
| PinH2  |  |
| PinH3  |  |
| PinH4  |  |
| PinH5  |  |
| PinH6  |  |
| PinH7  |  |
| PinH8  |  |
| PinH9  |  |
| PinH10 |  |
| PinH11 |  |
| PinH12 |  |
| PinH13 |  |
| PinH14 |  |
| PinH15 |  |
| PinI0  |  |
| PinI1  |  |
| PinI2  |  |
| PinI3  |  |
| PinI4  |  |
| PinI5  |  |
| PinI6  |  |
| PinI7  |  |
| PinI8  |  |
| PinI9  |  |
| PinI10 |  |
| PinI11 |  |
| PinI12 |  |
| PinI13 |  |
| PinI14 |  |

## Enumerator

|        |  |
|--------|--|
| PinI15 |  |
| PinJ0  |  |
| PinJ1  |  |
| PinJ2  |  |
| PinJ3  |  |
| PinJ4  |  |
| PinJ5  |  |
| PinJ6  |  |
| PinJ7  |  |
| PinJ8  |  |
| PinJ9  |  |
| PinJ10 |  |
| PinJ11 |  |
| PinJ12 |  |
| PinJ13 |  |
| PinJ14 |  |
| PinJ15 |  |

## 6.44.1.2 PinState

```
enum PinState
```

## Enumerator

|     |  |
|-----|--|
| Off |  |
| On  |  |

## 6.44.2 Function Documentation

## 6.44.2.1 bitsSet() [1/3]

```
void bitsSet (
    volatile BUInt8 & reg,
    BUInt32 bitPos,
    BUInt32 bitMask,
    BUInt32 value ) [inline]
```

### 6.44.2.2 bitsSet() [2/3]

```
void bitsSet (  
    volatile BUInt16 & reg,  
    BUInt32 bitPos,  
    BUInt32 bitMask,  
    BUInt32 value ) [inline]
```

### 6.44.2.3 bitsSet() [3/3]

```
void bitsSet (  
    volatile BUInt32 & reg,  
    BUInt32 bitPos,  
    BUInt32 bitMask,  
    BUInt32 value ) [inline]
```

## 6.44.3 Variable Documentation

### 6.44.3.1 gpio

BGpio gpio

## 6.45 BHeap.cpp File Reference

```
#include <BHeap.h>  
#include <stdlib.h>  
#include <FreeRTOS.h>  
#include <task.h>
```

## 6.46 BHeap.h File Reference

```
#include <stdlib.h>
```

### Classes

- class [BHeap](#)

## 6.47 BHeapTest.cpp File Reference

```
#include <BHeapTest.h>
#include <stdlib.h>
#include <malloc.h>
#include <stdio.h>
```

### Classes

- struct [HeapTestState](#)

### Functions

- void \* [heapTestAlloc](#) (size\_t size)
- void [heapTestPrint](#) ()
- void \* [\\_malloc\\_r](#) (\_reent \*r, size\_t size)
- void \* [\\_realloc\\_r](#) (\_reent \*r, void \*pOld, size\_t size)
- void \* [\\_calloc\\_r](#) (\_reent \*r, size\_t size1, size\_t size2)
- void [\\_free\\_r](#) (\_reent \*r, void \*p)

### Variables

- char \* [heapTestMem](#)
- BUInt [heapTestMemSize](#)
- char \* [heapTestMemPos](#) = 0
- [HeapTestState](#) [heapTestState](#) [64]
- BUInt [heapTestStateNext](#)

### 6.47.1 Function Documentation

#### 6.47.1.1 [\\_calloc\\_r\(\)](#)

```
void* _calloc_r (
    _reent * r,
    size_t size1,
    size_t size2 )
```

#### 6.47.1.2 [\\_free\\_r\(\)](#)

```
void _free_r (
    _reent * r,
    void * p )
```

### 6.47.1.3 `_malloc_r()`

```
void* _malloc_r (
    _reent * r,
    size_t size )
```

### 6.47.1.4 `_realloc_r()`

```
void* _realloc_r (
    _reent * r,
    void * pOld,
    size_t size )
```

### 6.47.1.5 `heapTestAlloc()`

```
void* heapTestAlloc (
    size_t size )
```

### 6.47.1.6 `heapTestPrint()`

```
void heapTestPrint ( )
```

## 6.47.2 Variable Documentation

### 6.47.2.1 `heapTestMem`

```
char* heapTestMem
```

### 6.47.2.2 `heapTestMemPos`

```
char* heapTestMemPos = 0
```

### 6.47.2.3 heapTestMemSize

`BUInt` heapTestMemSize

### 6.47.2.4 heapTestState

`HeapTestState` heapTestState[64]

### 6.47.2.5 heapTestStateNext

`BUInt` heapTestStateNext

## 6.48 BHeapTest.h File Reference

```
#include <BTypes.h>
```

### Functions

- void `heapTestPrint` ()

### Variables

- char \* `heapTestMem`
- `BUInt` `heapTestMemSize`

## 6.48.1 Function Documentation

### 6.48.1.1 heapTestPrint()

```
void heapTestPrint ( )
```

## 6.48.2 Variable Documentation

### 6.48.2.1 heapTestMem

```
char* heapTestMem
```

### 6.48.2.2 heapTestMemSize

```
BUInt heapTestMemSize
```

## 6.49 BHtml.cpp File Reference

```
#include <BHtml.h>
```

### Functions

- [BString urlEncode](#) ([BString](#) str)
- [BString urlEncodeComponent](#) ([BString](#) str)
- [BString urlDecode](#) ([BString](#) str)
- void [htmlDecode](#) (char \*str)
- void [htmlEncode](#) (char \*str)

### 6.49.1 Function Documentation

#### 6.49.1.1 htmlDecode()

```
void htmlDecode (  
    char * str )
```

#### 6.49.1.2 htmlEncode()

```
void htmlEncode (  
    char * str )
```

#### 6.49.1.3 urlDecode()

```
BString urlDecode (  
    BString str )
```

#### 6.49.1.4 `urlEncode()`

```
BString urlEncode (
    BString str )
```

#### 6.49.1.5 `urlEncodeComponent()`

```
BString urlEncodeComponent (
    BString str )
```

## 6.50 BHtml.h File Reference

```
#include <BList.h>
#include <BString.h>
```

### Classes

- class [BHtml](#)

### Typedefs

- typedef void(\* [HtmlWriteFunc](#)) (const [BString](#) &str)

### Functions

- [BString urlEncode](#) ([BString](#) str)
- [BString urlEncodeComponent](#) ([BString](#) str)
- [BString urlDecode](#) ([BString](#) str)
- void [htmlDecode](#) (char \*str)
- void [htmlEncode](#) (char \*str)

### 6.50.1 Typedef Documentation

#### 6.50.1.1 `HtmlWriteFunc`

```
typedef void(* HtmlWriteFunc) (const BString &str)
```

### 6.50.2 Function Documentation

### 6.50.2.1 htmlDecode()

```
void htmlDecode (  
    char * str )
```

### 6.50.2.2 htmlEncode()

```
void htmlEncode (  
    char * str )
```

### 6.50.2.3 urlDecode()

```
BString urlDecode (  
    BString str )
```

### 6.50.2.4 urlEncode()

```
BString urlEncode (  
    BString str )
```

### 6.50.2.5 urlEncodeComponent()

```
BString urlEncodeComponent (  
    BString str )
```

## 6.51 BHttpRequest.cpp File Reference

```
#include <BHttpRequest.h>  
#include <string.h>  
#include <stdio.h>  
#include <ctype.h>  
#include <limits.h>  
#include <BDebug.h>
```

### Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [BDEBUGL3](#) 0
- #define [BDEBUGL4](#) 0

## Functions

- static int `findCRNL` (const char \*buf)

## Variables

- const `BUInt Timeout` = 10

## 6.51.1 Macro Definition Documentation

### 6.51.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 6.51.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

### 6.51.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

### 6.51.1.4 BDEBUGL4

```
#define BDEBUGL4 0
```

## 6.51.2 Function Documentation

### 6.51.2.1 findCRNL()

```
static int findCRNL (  
    const char * buf ) [static]
```

### 6.51.3 Variable Documentation

#### 6.51.3.1 Timeout

```
const BUInt Timeout = 10
```

## 6.52 BHttpRequest.h File Reference

```
#include <BTask.h>
#include <BNetwork.h>
#include <BFifo.h>
#include <BString.h>
#include <BObjStringFormat.h>
#include <BDict.h>
#include <BError.h>
#include <string.h>
#include <stdio.h>
#include <ctype.h>
```

### Classes

- class [BFifoChar](#)
- class [BHttpRequest](#)

## 6.53 BI2c.cpp File Reference

```
#include <BI2c.h>
#include <BSys.h>
#include <BGpio.h>
#include <stdio.h>
```

### Macros

- #define [I2CTYPE\\_0](#) 1
- #define [EVENT\\_MASTER\\_MODE\\_SELECT](#) 0x00030001
- #define [EVENT\\_MASTER\\_TRANSMITTER\\_MODE\\_SELECTED](#) 0x00070082
- #define [EVENT\\_MASTER\\_RECEIVER\\_MODE\\_SELECTED](#) 0x00030002
- #define [EVENT\\_MASTER\\_MODE\\_ADDRESS10](#) 0x00030008
- #define [EVENT\\_MASTER\\_BYTE\\_RECEIVED](#) 0x00030040
- #define [EVENT\\_MASTER\\_BYTE\\_TRANSMITTING](#) 0x00070080
- #define [EVENT\\_MASTER\\_BYTE\\_TRANSMITTED](#) 0x00070084
- #define [EVENT\\_SLAVE\\_RECEIVER\\_ADDRESS\\_MATCHED](#) 0x00020002
- #define [EVENT\\_SLAVE\\_TRANSMITTER\\_ADDRESS\\_MATCHED](#) 0x00060082
- #define [EVENT\\_SLAVE\\_RECEIVER\\_SECONDADDRESS\\_MATCHED](#) 0x00820000
- #define [EVENT\\_SLAVE\\_TRANSMITTER\\_SECONDADDRESS\\_MATCHED](#) 0x00860080
- #define [EVENT\\_SLAVE\\_GENERALCALLADDRESS\\_MATCHED](#) 0x00120000

## Variables

- static `Bl2c * gdevs [3]`

### 6.53.1 Macro Definition Documentation

#### 6.53.1.1 EVENT\_MASTER\_BYTE\_RECEIVED

```
#define EVENT_MASTER_BYTE_RECEIVED 0x00030040
```

#### 6.53.1.2 EVENT\_MASTER\_BYTE\_TRANSMITTED

```
#define EVENT_MASTER_BYTE_TRANSMITTED 0x00070084
```

#### 6.53.1.3 EVENT\_MASTER\_BYTE\_TRANSMITTING

```
#define EVENT_MASTER_BYTE_TRANSMITTING 0x00070080
```

#### 6.53.1.4 EVENT\_MASTER\_MODE\_ADDRESS10

```
#define EVENT_MASTER_MODE_ADDRESS10 0x00030008
```

#### 6.53.1.5 EVENT\_MASTER\_MODE\_SELECT

```
#define EVENT_MASTER_MODE_SELECT 0x00030001
```

#### 6.53.1.6 EVENT\_MASTER\_RECEIVER\_MODE\_SELECTED

```
#define EVENT_MASTER_RECEIVER_MODE_SELECTED 0x00030002
```

#### 6.53.1.7 EVENT\_MASTER\_TRANSMITTER\_MODE\_SELECTED

```
#define EVENT_MASTER_TRANSMITTER_MODE_SELECTED 0x00070082
```

#### 6.53.1.8 EVENT\_SLAVE\_GENERALCALLADDRESS\_MATCHED

```
#define EVENT_SLAVE_GENERALCALLADDRESS_MATCHED 0x00120000
```

#### 6.53.1.9 EVENT\_SLAVE\_RECEIVER\_ADDRESS\_MATCHED

```
#define EVENT_SLAVE_RECEIVER_ADDRESS_MATCHED 0x00020002
```

#### 6.53.1.10 EVENT\_SLAVE\_RECEIVER\_SECONDADDRESS\_MATCHED

```
#define EVENT_SLAVE_RECEIVER_SECONDADDRESS_MATCHED 0x00820000
```

#### 6.53.1.11 EVENT\_SLAVE\_TRANSMITTER\_ADDRESS\_MATCHED

```
#define EVENT_SLAVE_TRANSMITTER_ADDRESS_MATCHED 0x00060082
```

#### 6.53.1.12 EVENT\_SLAVE\_TRANSMITTER\_SECONDADDRESS\_MATCHED

```
#define EVENT_SLAVE_TRANSMITTER_SECONDADDRESS_MATCHED 0x00860080
```

#### 6.53.1.13 I2CTYPE\_0

```
#define I2CTYPE_0 1
```

### 6.53.2 Variable Documentation

### 6.53.2.1 gdevs

```
BI2c* gdevs[3] [static]
```

## 6.54 BI2c.h File Reference

```
#include <BTypes.h>
#include <BGpio.h>
#include <BError.h>
#include <BMutex.h>
#include <BDma.h>
#include <ArmSystem.h>
```

### Classes

- class [BI2c](#)

## 6.55 BInterrupt.cpp File Reference

```
#include <BInterrupt.h>
#include <BSys.h>
#include <BDebug.h>
#include <ArmSystem.h>
```

### Macros

- #define [LDEBUG](#) 0
- #define [dlprintf](#)(fmt, a...)

### 6.55.1 Macro Definition Documentation

#### 6.55.1.1 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

### 6.55.1.2 LDEBUG

```
#define LDEBUG 0
```

## 6.56 BInterrupt.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BGpio.h>
```

### Classes

- class [BInterrupt](#)

## 6.57 BList.h File Reference

```
#include <BList_func.h>
```

### Classes

- class [BNode](#)
- class [BIter](#)  
*Iterator for BList.*
- class [BList< T >](#)  
*Template based list class.*
- class [BList< T >::Node](#)

### Macros

- `#define BListLoop(list, i) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))`

### 6.57.1 Macro Definition Documentation

#### 6.57.1.1 BListLoop

```
#define BListLoop(  
    list,  
    i ) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))
```

## 6.58 BList\_func.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

## 6.59 BMutex.cpp File Reference

```
#include <BMutex.h>
#include <BSys.h>
```

## 6.60 BMutex.h File Reference

```
#include <BTypes.h>
#include <FreeRTOS.h>
#include <semphr.h>
```

### Classes

- class [BMutex](#)  
*Mutex class.*
- class [BMutexLock](#)

## 6.61 BNameValue.h File Reference

```
#include <BList.h>
#include <BString.h>
```

### Classes

- class [BNameValue< T >](#)
- class [BNameValueList< T >](#)

## 6.62 BNetwork.cpp File Reference

```
#include <BNetwork.h>
#include <BSys.h>
#include <string.h>
#include <lwip/opt.h>
#include <lwip/init.h>
#include <lwip/netif.h>
#include <lwip/tcpip.h>
#include <lwip/stats.h>
#include <lwip/snmp.h>
#include <netif/etharp.h>
#include <BDebug.h>
#include <lwip/dhcp.h>
```

### Macros

- #define [LDEBUG](#) 0
- #define [TX\\_ZERO\\_COPY](#) 0
- #define [dlprintf](#)(fmt, a...)

### Functions

- void [ETH\\_IRQHandler](#) (void)
- void [HAL\\_ETH\\_RxCpltCallback](#) (ETH\_HandleTypeDef \*eth)
- err\_t [ether\\_init](#) (struct netif \*netif)
- static err\_t [ether\\_send](#) (struct netif \*netif, struct pbuf \*p)
- void [ether\\_link\\_changed](#) (struct netif \*netif)

### Variables

- const int [PhyDP83848\\_LEDCR](#) = 0x18
- const int [PhyDP83848\\_CR](#) = 0x19
- const int [DhcpTries](#) = 4
- const int [ProcessTaskPriority](#) = 3
- const int [EventEthernet](#) = 1
- const int [EventLinkChanged](#) = 2
- EthPhySettings [ethPhySettings](#)
- static [BNetwork](#) \* [bnetwork](#)
- static ETH\_DMADescTypeDef [dmaRxDscrTab](#) [ETH\_RXBUFNB]
- static ETH\_DMADescTypeDef [dmaTxDscrTab](#) [ETH\_TXBUFNB]
- static uint8\_t [rxBuffer](#) [ETH\_RXBUFNB][ETH\_RX\_BUF\_SIZE]
- static uint8\_t [txBuffer](#) [ETH\_TXBUFNB][ETH\_TX\_BUF\_SIZE]

#### 6.62.1 Macro Definition Documentation

### 6.62.1.1 dprintf

```
#define dprintf(  
    fmt,  
    a... )
```

### 6.62.1.2 LDEBUG

```
#define LDEBUG 0
```

### 6.62.1.3 TX\_ZERO\_COPY

```
#define TX_ZERO_COPY 0
```

## 6.62.2 Function Documentation

### 6.62.2.1 ETH\_IRQHandler()

```
void ETH_IRQHandler (  
    void )
```

### 6.62.2.2 ether\_init()

```
err_t ether_init (  
    struct netif * netif )
```

### 6.62.2.3 ether\_link\_changed()

```
void ether_link_changed (  
    struct netif * netif )
```

#### 6.62.2.4 ether\_send()

```
static err_t ether_send (  
    struct netif * netif,  
    struct pbuf * p ) [static]
```

#### 6.62.2.5 HAL\_ETH\_RxCpltCallback()

```
void HAL_ETH_RxCpltCallback (  
    ETH_HandleTypeDef * eth )
```

### 6.62.3 Variable Documentation

#### 6.62.3.1 bnetwork

```
BNetwork* bnetwork [static]
```

#### 6.62.3.2 DhcpTries

```
const int DhcpTries = 4
```

#### 6.62.3.3 dmaRxDscrTab

```
ETH_DMADescTypeDef dmaRxDscrTab[ETH_RXBUFNB] [static]
```

#### 6.62.3.4 dmaTxDscrTab

```
ETH_DMADescTypeDef dmaTxDscrTab[ETH_TXBUFNB] [static]
```

#### 6.62.3.5 ethPhySettings

```
EthPhySettings ethPhySettings
```

### 6.62.3.6 EventEthernet

```
const int EventEthernet = 1
```

### 6.62.3.7 EventLinkChanged

```
const int EventLinkChanged = 2
```

### 6.62.3.8 PhyDP83848\_CR

```
const int PhyDP83848_CR = 0x19
```

### 6.62.3.9 PhyDP83848\_LED\_CR

```
const int PhyDP83848_LED_CR = 0x18
```

### 6.62.3.10 ProcessTaskPriority

```
const int ProcessTaskPriority = 3
```

### 6.62.3.11 rxBuffer

```
uint8_t rxBuffer[ETH_RXBUFNB][ETH_RX_BUF_SIZE] [static]
```

### 6.62.3.12 txBuffer

```
uint8_t txBuffer[ETH_TXBUFNB][ETH_TX_BUF_SIZE] [static]
```

## 6.63 BNetwork.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BString.h>
#include <BTask.h>
#include <BQueue.h>
#include <ArmSystem.h>
#include <lwip/netif.h>
#include <lwip/api.h>
```

### Classes

- class [IpAddress](#)
- class [BNetworkProcessTask](#)
- class [BNetworkDhcpTask](#)
- class [BNetworkStats](#)
- class [BNetwork](#)

### Typedefs

- typedef pbuf [IpPacket](#)

### Enumerations

- enum [DhcpState](#) {  
    [DhcpStateOff](#), [DhcpStateStart](#), [DhcpStateWait](#), [DhcpStateAssigned](#),  
    [DhcpStateTimeout](#), [DhcpStateDown](#) }

#### 6.63.1 Typedef Documentation

##### 6.63.1.1 IpPacket

```
typedef pbuf IpPacket
```

#### 6.63.2 Enumeration Type Documentation

##### 6.63.2.1 DhcpState

```
enum DhcpState
```

### Enumerator

|                   |  |
|-------------------|--|
| DhcpStateOff      |  |
| DhcpStateStart    |  |
| DhcpStateWait     |  |
| DhcpStateAssigned |  |
| DhcpStateTimeout  |  |
| DhcpStateDown     |  |

## 6.64 BNtpClient.cpp File Reference

```
#include <BNtpClient.h>
#include <BDebug.h>
#include <time.h>
#include <lwip/apps/sntp.h>
```

### Macros

- `#define BDEBUGL1 0`

### Functions

- void `sntp_set_system_time` (u32\_t sec, u32\_t usec)

### Variables

- static `BNtpClient * ontpClient`

## 6.64.1 Macro Definition Documentation

### 6.64.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

## 6.64.2 Function Documentation

### 6.64.2.1 sntp\_set\_system\_time()

```
void sntp_set_system_time (
    u32_t sec,
    u32_t usec )
```

### 6.64.3 Variable Documentation

#### 6.64.3.1 ontpClient

```
BNtpClient* ontpClient [static]
```

## 6.65 BNtpClient.h File Reference

```
#include <BNetwork.h>
```

### Classes

- class [BNtpClient](#)

## 6.66 BoapMc.cpp File Reference

```
#include <BoapMc.h>
#include <BCrc16.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

### Macros

- #define [DEBUG\\_LOCAL](#) 0
- #define [DEBUG\\_LOCAL1](#) 0
- #define [dlprintf](#)(fmt, a...)
- #define [dl1printf](#)(fmt, a...)

### 6.66.1 Macro Definition Documentation

### 6.66.1.1 DEBUG\_LOCAL

```
#define DEBUG_LOCAL 0
```

### 6.66.1.2 DEBUG\_LOCAL1

```
#define DEBUG_LOCAL1 0
```

### 6.66.1.3 dl1printf

```
#define dl1printf(  
    fmt,  
    a... )
```

### 6.66.1.4 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

## 6.67 BoapMc.h File Reference

```
#include <BTypes.h>  
#include <BMutex.h>  
#include <BSemaphore.h>  
#include <BQueue.h>  
#include <BFifo.h>  
#include <BComms.h>
```

### Classes

- struct [BoapMcPacketHead](#)
- class [BoapMcPacket](#)
- class [BoapMcClientObject](#)
- class [BoapMcSignalObject](#)
- class [BoapMcServiceObject](#)
- class [BoapMcComms](#)

### Enumerations

- enum [BoapMcType](#) { [BoapMcTypeRequest](#) = 0x00, [BoapMcTypeReply](#) = 0x80 }

## Functions

- struct [BoapMcPacketHead](#) `__attribute__((aligned(8), packed))`

## Variables

- [BUInt8](#) `length`
- [BUInt8](#) `addressTo`
- [BUInt8](#) `addressFrom`
- [BUInt8](#) `cmd`
- [BUInt16](#) `error`
- [BUInt16](#) `checksum`
- class [BoapMcPacket](#) `__attribute__((aligned(8), packed))`

## 6.67.1 Enumeration Type Documentation

### 6.67.1.1 BoapMcType

```
enum BoapMcType
```

#### Enumerator

|                   |  |
|-------------------|--|
| BoapMcTypeRequest |  |
| BoapMcTypeReply   |  |

## 6.67.2 Function Documentation

### 6.67.2.1 \_\_attribute\_\_((aligned(8), packed))

```
struct BoapMcPacketHead __attribute__((aligned(8), packed))
```

## 6.67.3 Variable Documentation

### 6.67.3.1 \_\_attribute\_\_((aligned(8), packed))

```
struct BoapMcError __attribute__((aligned(8), packed))
```

### 6.67.3.2 addressFrom

`BUInt8` addressFrom

### 6.67.3.3 addressTo

`BUInt8` addressTo

### 6.67.3.4 checksum

`BUInt16` checksum

### 6.67.3.5 cmd

`BUInt8` cmd

### 6.67.3.6 error

`BUInt16` error

### 6.67.3.7 length

`BUInt8` length

## 6.68 BoapMc1.cpp File Reference

```
#include <BoapMc1.h>
#include <BSys.h>
#include <BCrc32.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <BDebug.h>
```

## Macros

- `#define BDEBUGL1 0`
- `#define BDEBUGL2 0`

### 6.68.1 Macro Definition Documentation

#### 6.68.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 6.68.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 6.69 BoapMc1.h File Reference

```
#include <BTypes.h>
#include <BMutex.h>
#include <BSemaphore.h>
#include <BQueue.h>
#include <BFifo.h>
#include <BComms.h>
```

## Classes

- struct [BoapMc1PacketHead](#)
- class [BoapMc1Packet](#)
- struct [BoapMc1Error](#)
- class [BoapMc1Comms](#)

## Enumerations

- enum [BoapMc1Type](#) { [BoapMc1TypeRequest](#) = 0x0000, [BoapMc1TypeReply](#) = 0x8000 }

## Functions

- struct [BoapMc1PacketHead](#) `__attribute__((aligned(8), packed))`
- [BUInt32 boapMc1CommsRoundupLen](#) ([BUInt32](#) len)

## Variables

- const `BUInt16 BoapMc1Magic` = 0x5542
- `BUInt16 magic`
- `BUInt16 length`
- `BUInt16 addressTo`
  - Do we want addresses ?*
- `BUInt16 addressFrom`
- `BUInt16 cmd`
- `BInt16 error`
  - Not used. Maybe return a `BError` object ?*
- `BUInt32 checksum`
  - Not used. Not sure we want this here or at the very end of the packet.*
- `BoapMc1PacketHead head`
- char `data` [8]
- `BInt16 number`
  - The error number.*
- char `string` [32]
  - The error string.*
- class `BoapMc1Comms __attribute__`

## 6.69.1 Enumeration Type Documentation

### 6.69.1.1 BoapMc1Type

```
enum BoapMc1Type
```

#### Enumerator

|                    |  |
|--------------------|--|
| BoapMc1TypeRequest |  |
| BoapMc1TypeReply   |  |

## 6.69.2 Function Documentation

### 6.69.2.1 \_\_attribute\_\_()

```
struct BoapMc1PacketHead __attribute__ (
    (aligned(8), packed) )
```

### 6.69.2.2 boapMc1CommsRoundupLen()

```
BUInt32 boapMc1CommsRoundupLen (
    BUInt32 len ) [inline]
```

## 6.69.3 Variable Documentation

### 6.69.3.1 \_\_attribute\_\_

```
class BoapMc1Comms __attribute__
```

### 6.69.3.2 addressFrom

```
BUInt16 addressFrom
```

### 6.69.3.3 addressTo

```
BUInt16 addressTo
```

Do we want addresses ?

### 6.69.3.4 BoapMc1Magic

```
const BUInt16 BoapMc1Magic = 0x5542
```

### 6.69.3.5 checksum

```
BUInt32 checksum
```

Not used. Not sure we want this here or at the very end of the packet.

**6.69.3.6 cmd**

`BUInt16` cmd

**6.69.3.7 data**

`char data[8]`

**6.69.3.8 error**

`BInt16` error

Not used. Maybe return a `BError` object ?

**6.69.3.9 head**

`BoapMclPacketHead` head

**6.69.3.10 length**

`BUInt16` length

**6.69.3.11 magic**

`BUInt16` magic

**6.69.3.12 number**

`BInt16` number

The error number.

### 6.69.3.13 string

```
char string[32]
```

The error string.

## 6.70 BObj.cpp File Reference

```
#include <BObj.h>
#include <stdio.h>
```

## 6.71 BObj.h File Reference

```
#include <BTypes.h>
```

### Classes

- class [BObj](#)

## 6.72 BObjStringFormat.cpp File Reference

```
#include <BObjStringFormat.h>
#include <BTime.h>
#include <math.h>
```

### Functions

- [BString toBString](#) (BString n, Bool v)
- [BString toBString](#) (BString n, BInt8 v)
- [BString toBString](#) (BString n, BUInt8 v)
- [BString toBString](#) (BString n, BInt16 v)
- [BString toBString](#) (BString n, BUInt16 v)
- [BString toBString](#) (BString n, BInt32 v)
- [BString toBString](#) (BString n, BUInt32 v)
- [BString toBString](#) (BString n, BInt64 v)
- [BString toBString](#) (BString n, BUInt64 v)
- [BString toBString](#) (BString n, BFloat32 v)
- [BString toBString](#) (BString n, BFloat64 v)
- [BString toBString](#) (BString n, BChar v)
- [BString toBString](#) (BString n, const BChar \*v)
- [BString toBString](#) (BString n, BString v)
- [BString toBString](#) (BString n, BError v)
- [BString toBString](#) (BString n, BTime v)

- [BString toBString](#) (BString name, const [BObjMember](#) \*m, const void \*obj, [BStringList](#) ignoreFields)
- [BString toBString](#) (BString n, [BObj](#) &obj)
- [BString toBStringJson](#) (BString n, [Bool](#) v)
- [BString toBStringJson](#) (BString n, [BInt8](#) v)
- [BString toBStringJson](#) (BString n, [BUInt8](#) v)
- [BString toBStringJson](#) (BString n, [BInt16](#) v)
- [BString toBStringJson](#) (BString n, [BUInt16](#) v)
- [BString toBStringJson](#) (BString n, [BInt32](#) v)
- [BString toBStringJson](#) (BString n, [BUInt32](#) v)
- [BString toBStringJson](#) (BString n, [BInt64](#) v)
- [BString toBStringJson](#) (BString n, [BUInt64](#) v)
- [BString toBStringJson](#) (BString n, [BFloat32](#) v)
- [BString toBStringJson](#) (BString n, [BFloat64](#) v)
- [BString toBStringJson](#) (BString n, [BChar](#) v)
- [BString toBStringJson](#) (BString n, const [BChar](#) \*v)
- [BString toBStringJson](#) (BString n, [BString](#) v)
- [BString toBStringJson](#) (BString n, [BError](#) v)
- [BString toBStringJson](#) (BString n, [BTime](#) v)
- [BString toBStringJson](#) (BString n, const [BObjMember](#) \*m, const void \*obj, [BStringList](#) ignoreFields)
- [BString toBStringJson](#) (BString n, [BObj](#) &obj)
- [BError toBDictStringFromJson](#) (BString json, [BDictString](#) &ds)

## 6.72.1 Function Documentation

### 6.72.1.1 toBDictStringFromJson()

```
BError toBDictStringFromJson (
    BString json,
    BDictString & ds )
```

### 6.72.1.2 toBString() [1/18]

```
BString toBString (
    BString n,
    Bool v )
```

### 6.72.1.3 toBString() [2/18]

```
BString toBString (
    BString n,
    BInt8 v )
```

**6.72.1.4 toBString()** [3/18]

```
BString toBString (  
    BString n,  
    BUInt8 v )
```

**6.72.1.5 toBString()** [4/18]

```
BString toBString (  
    BString n,  
    BInt16 v )
```

**6.72.1.6 toBString()** [5/18]

```
BString toBString (  
    BString n,  
    BUInt16 v )
```

**6.72.1.7 toBString()** [6/18]

```
BString toBString (  
    BString n,  
    BInt32 v )
```

**6.72.1.8 toBString()** [7/18]

```
BString toBString (  
    BString n,  
    BUInt32 v )
```

**6.72.1.9 toBString()** [8/18]

```
BString toBString (  
    BString n,  
    BInt64 v )
```

**6.72.1.10 toBString()** [9/18]

```
BString toBString (
    BString n,
    BUInt64 v )
```

**6.72.1.11 toBString()** [10/18]

```
BString toBString (
    BString n,
    BFloat32 v )
```

**6.72.1.12 toBString()** [11/18]

```
BString toBString (
    BString n,
    BFloat64 v )
```

**6.72.1.13 toBString()** [12/18]

```
BString toBString (
    BString n,
    BChar v )
```

**6.72.1.14 toBString()** [13/18]

```
BString toBString (
    BString n,
    const BChar * v )
```

**6.72.1.15 toBString()** [14/18]

```
BString toBString (
    BString n,
    BString v )
```

**6.72.1.16 toBString()** [15/18]

```
BString toBString (
    BString n,
    BError v )
```

**6.72.1.17 toBString()** [16/18]

```
BString toBString (
    BString n,
    BTime v )
```

**6.72.1.18 toBString()** [17/18]

```
BString toBString (
    BString name,
    const BObjMember * m,
    const void * obj,
    BStringList ignoreFields )
```

**6.72.1.19 toBString()** [18/18]

```
BString toBString (
    BString n,
    BObj & obj )
```

**6.72.1.20 toBStringJson()** [1/18]

```
BString toBStringJson (
    BString n,
    Bool v )
```

**6.72.1.21 toBStringJson()** [2/18]

```
BString toBStringJson (
    BString n,
    BInt8 v )
```

**6.72.1.22 toBStringJson()** [3/18]

```
BString toBStringJson (
    BString n,
    BUInt8 v )
```

**6.72.1.23 toBStringJson()** [4/18]

```
BString toBStringJson (
    BString n,
    BInt16 v )
```

**6.72.1.24 toBStringJson()** [5/18]

```
BString toBStringJson (
    BString n,
    BUInt16 v )
```

**6.72.1.25 toBStringJson()** [6/18]

```
BString toBStringJson (
    BString n,
    BInt32 v )
```

**6.72.1.26 toBStringJson()** [7/18]

```
BString toBStringJson (
    BString n,
    BUInt32 v )
```

**6.72.1.27 toBStringJson()** [8/18]

```
BString toBStringJson (
    BString n,
    BInt64 v )
```

**6.72.1.28 toBStringJson()** [9/18]

```
BString toBStringJson (
    BString n,
    BUInt64 v )
```

**6.72.1.29 toBStringJson()** [10/18]

```
BString toBStringJson (
    BString n,
    BFloat32 v )
```

**6.72.1.30 toBStringJson()** [11/18]

```
BString toBStringJson (
    BString n,
    BFloat64 v )
```

**6.72.1.31 toBStringJson()** [12/18]

```
BString toBStringJson (
    BString n,
    BChar v )
```

**6.72.1.32 toBStringJson()** [13/18]

```
BString toBStringJson (
    BString n,
    const BChar * v )
```

**6.72.1.33 toBStringJson()** [14/18]

```
BString toBStringJson (
    BString n,
    BString v )
```

**6.72.1.34 toBStringJson()** [15/18]

```
BString toBStringJson (
    BString n,
    BError v )
```

**6.72.1.35 toBStringJson()** [16/18]

```
BString toBStringJson (
    BString n,
    BTime v )
```

**6.72.1.36 toBStringJson()** [17/18]

```
BString toBStringJson (
    BString n,
    const BObjMember * m,
    const void * obj,
    BStringList ignoreFields )
```

**6.72.1.37 toBStringJson()** [18/18]

```
BString toBStringJson (
    BString n,
    BObj & obj )
```

## 6.73 BObjStringFormat.h File Reference

```
#include <BObj.h>
#include <BString.h>
#include <BDict.h>
#include <BTime.h>
```

## Functions

- [BString toBString](#) (BString name, [Bool](#) value)
- [BString toBString](#) (BString name, [BInt8](#) value)
- [BString toBString](#) (BString name, [BUInt8](#) value)
- [BString toBString](#) (BString name, [BInt16](#) value)
- [BString toBString](#) (BString name, [BUInt16](#) value)
- [BString toBString](#) (BString name, [BInt32](#) value)
- [BString toBString](#) (BString name, [BUInt32](#) value)
- [BString toBString](#) (BString name, [BInt64](#) value)
- [BString toBString](#) (BString name, [BUInt64](#) value)
- [BString toBString](#) (BString name, [BFloat32](#) value)
- [BString toBString](#) (BString name, [BFloat64](#) value)
- [BString toBString](#) (BString name, [BChar](#) value)
- [BString toBString](#) (BString name, const [BChar](#) \*value)
- [BString toBString](#) (BString name, [BString](#) value)
- [BString toBString](#) (BString name, [BError](#) value)
- [BString toBString](#) (BString name, [BTime](#) time)
- [BString toBString](#) (BString name, const [BObjMember](#) \*members, const void \*obj, [BStringList](#) ignore←  
Fields=[BStringList](#)())
- [BString toBString](#) (BString name, [BObj](#) &obj)
- [BString toBStringJson](#) (BString name, [Bool](#) value)
- [BString toBStringJson](#) (BString name, [BInt8](#) value)
- [BString toBStringJson](#) (BString name, [BUInt8](#) value)
- [BString toBStringJson](#) (BString name, [BInt16](#) value)
- [BString toBStringJson](#) (BString name, [BUInt16](#) value)
- [BString toBStringJson](#) (BString name, [BInt32](#) value)
- [BString toBStringJson](#) (BString name, [BUInt32](#) value)
- [BString toBStringJson](#) (BString name, [BInt64](#) value)
- [BString toBStringJson](#) (BString name, [BUInt64](#) value)
- [BString toBStringJson](#) (BString name, [BFloat32](#) value)
- [BString toBStringJson](#) (BString name, [BFloat64](#) value)
- [BString toBStringJson](#) (BString name, [BChar](#) value)
- [BString toBStringJson](#) (BString name, const [BChar](#) \*value)
- [BString toBStringJson](#) (BString name, [BString](#) value)
- [BString toBStringJson](#) (BString name, [BError](#) value)
- [BString toBStringJson](#) (BString name, [BTime](#) time)
- [BString toBStringJson](#) (BString name, const [BObjMember](#) \*members, const void \*obj, [BStringList](#) ignore←  
Fields=[BStringList](#)())
- [BString toBStringJson](#) (BString name, [BObj](#) &obj)
- [BError toBDictStringFromJson](#) (BString json, [BDictString](#) &ds)
- [BString base64\\_encode](#) (void \*data, [BUInt](#) len)
- [BError base64\\_decode](#) (BString strIn, [BString](#) &strOut)

### 6.73.1 Function Documentation

#### 6.73.1.1 base64\_decode()

```
BError base64_decode (
    BString strIn,
    BString & strOut )
```

### 6.73.1.2 base64\_encode()

```
BString base64_encode (
    void * data,
    BUInt len )
```

### 6.73.1.3 toBDictStringFromJson()

```
BError toBDictStringFromJson (
    BString json,
    BDictString & ds )
```

### 6.73.1.4 toBString() [1/18]

```
BString toBString (
    BString name,
    Bool value )
```

### 6.73.1.5 toBString() [2/18]

```
BString toBString (
    BString name,
    BInt8 value )
```

### 6.73.1.6 toBString() [3/18]

```
BString toBString (
    BString name,
    BUInt8 value )
```

### 6.73.1.7 toBString() [4/18]

```
BString toBString (
    BString name,
    BInt16 value )
```

**6.73.1.8 toBString()** [5/18]

```
BString toBString (  
    BString name,  
    BUInt16 value )
```

**6.73.1.9 toBString()** [6/18]

```
BString toBString (  
    BString name,  
    BInt32 value )
```

**6.73.1.10 toBString()** [7/18]

```
BString toBString (  
    BString name,  
    BUInt32 value )
```

**6.73.1.11 toBString()** [8/18]

```
BString toBString (  
    BString name,  
    BInt64 value )
```

**6.73.1.12 toBString()** [9/18]

```
BString toBString (  
    BString name,  
    BUInt64 value )
```

**6.73.1.13 toBString()** [10/18]

```
BString toBString (  
    BString name,  
    BFloat32 value )
```

**6.73.1.14 toBString()** [11/18]

```
BString toBString (
    BString name,
    BFloat64 value )
```

**6.73.1.15 toBString()** [12/18]

```
BString toBString (
    BString name,
    BChar value )
```

**6.73.1.16 toBString()** [13/18]

```
BString toBString (
    BString name,
    const BChar * value )
```

**6.73.1.17 toBString()** [14/18]

```
BString toBString (
    BString name,
    BString value )
```

**6.73.1.18 toBString()** [15/18]

```
BString toBString (
    BString name,
    BError value )
```

**6.73.1.19 toBString()** [16/18]

```
BString toBString (
    BString name,
    BTime time )
```

**6.73.1.20 toBString()** [17/18]

```
BString toBString (
    BString name,
    const BObjMember * members,
    const void * obj,
    BStringList ignoreFields = BStringList() )
```

**6.73.1.21 toBString()** [18/18]

```
BString toBString (
    BString name,
    BObj & obj )
```

**6.73.1.22 toBStringJson()** [1/18]

```
BString toBStringJson (
    BString name,
    Bool value )
```

**6.73.1.23 toBStringJson()** [2/18]

```
BString toBStringJson (
    BString name,
    BInt8 value )
```

**6.73.1.24 toBStringJson()** [3/18]

```
BString toBStringJson (
    BString name,
    BUInt8 value )
```

**6.73.1.25 toBStringJson()** [4/18]

```
BString toBStringJson (
    BString name,
    BInt16 value )
```

**6.73.1.26 toBStringJson()** [5/18]

```
BString toBStringJson (  
    BString name,  
    BUInt16 value )
```

**6.73.1.27 toBStringJson()** [6/18]

```
BString toBStringJson (  
    BString name,  
    BInt32 value )
```

**6.73.1.28 toBStringJson()** [7/18]

```
BString toBStringJson (  
    BString name,  
    BUInt32 value )
```

**6.73.1.29 toBStringJson()** [8/18]

```
BString toBStringJson (  
    BString name,  
    BInt64 value )
```

**6.73.1.30 toBStringJson()** [9/18]

```
BString toBStringJson (  
    BString name,  
    BUInt64 value )
```

**6.73.1.31 toBStringJson()** [10/18]

```
BString toBStringJson (  
    BString name,  
    BFloat32 value )
```

**6.73.1.32 toBStringJson()** [11/18]

```
BString toBStringJson (  
    BString name,  
    BFloat64 value )
```

**6.73.1.33 toBStringJson()** [12/18]

```
BString toBStringJson (  
    BString name,  
    BChar value )
```

**6.73.1.34 toBStringJson()** [13/18]

```
BString toBStringJson (  
    BString name,  
    const BChar * value )
```

**6.73.1.35 toBStringJson()** [14/18]

```
BString toBStringJson (  
    BString name,  
    BString value )
```

**6.73.1.36 toBStringJson()** [15/18]

```
BString toBStringJson (  
    BString name,  
    BError value )
```

**6.73.1.37 toBStringJson()** [16/18]

```
BString toBStringJson (  
    BString name,  
    BTime time )
```

#### 6.73.1.38 toBStringJson() [17/18]

```
BString toBStringJson (  
    BString name,  
    const BObjMember * members,  
    const void * obj,  
    BStringList ignoreFields = BStringList() )
```

#### 6.73.1.39 toBStringJson() [18/18]

```
BString toBStringJson (  
    BString name,  
    BObj & obj )
```

### 6.74 BPwm.cpp File Reference

```
#include <BPwm.h>  
#include <BSys.h>
```

### 6.75 BPwm.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BTim.h>
```

#### Classes

- class [BPwm](#)

### 6.76 BQueue.cpp File Reference

```
#include <BQueue.h>  
#include <BSys.h>
```

### 6.77 BQueue.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BSys.h>  
#include <FreeRTOS.h>  
#include <queue.h>
```

## Classes

- class [BQueue< T >](#)  
*Queue class.*

## Typedefs

- typedef [BQueue< Bint32 >](#) [BQueueInt](#)

### 6.77.1 Typedef Documentation

#### 6.77.1.1 BQueueInt

```
typedef BQueue<BInt32> BQueueInt
```

## 6.78 BRandom.cpp File Reference

```
#include <BRandom.h>  
#include <BSys.h>  
#include <ArmSystem.h>
```

## 6.79 BRandom.h File Reference

```
#include <BTypes.h>
```

## Classes

- class [BRandom](#)

## 6.80 BRtc.cpp File Reference

```
#include <BRtc.h>  
#include <BSys.h>  
#include <BGpio.h>  
#include <BDebug.h>
```

## Macros

- #define `USE_SYNC` 1
- #define `LDEBUG` 0
- #define `dlprintf`(fmt, a...)

## Functions

- static `BUInt8 toBcd` (`BUInt v`)
- static `BUInt8 fromBcd` (`BUInt v`)

### 6.80.1 Macro Definition Documentation

#### 6.80.1.1 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

#### 6.80.1.2 LDEBUG

```
#define LDEBUG 0
```

#### 6.80.1.3 USE\_SYNC

```
#define USE_SYNC 1
```

### 6.80.2 Function Documentation

#### 6.80.2.1 fromBcd()

```
static BUInt8 fromBcd (  
    BUInt v ) [static]
```

### 6.80.2.2 toBcd()

```
static BUInt8 toBcd (  
    BUInt v ) [static]
```

## 6.81 BRtc.h File Reference

```
#include <BError.h>  
#include <BTime.h>  
#include <BMutex.h>  
#include <ArmSystem.h>
```

### Classes

- class [BRtc](#)

## 6.82 BSdio.cpp File Reference

```
#include <BSdio.h>  
#include <BSys.h>  
#include <BGpio.h>  
#include <BDebug.h>
```

### Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [USE\\_DMA](#) 1
- #define [SD\\_CMD\\_GO\\_IDLE\\_STATE](#) 0
- #define [SD\\_CMD\\_SEND\\_OP\\_COND](#) 1
- #define [SD\\_CMD\\_ALL\\_SEND\\_CID](#) 2
- #define [SD\\_CMD\\_SET\\_REL\\_ADDR](#) 3
- #define [SD\\_CMD\\_SET\\_DSR](#) 4
- #define [SD\\_CMD\\_SDIO\\_SEN\\_OP\\_COND](#) 5
- #define [SD\\_CMD\\_HS\\_SWITCH](#) 6
- #define [SD\\_CMD\\_SEL\\_DESEL\\_CARD](#) 7
- #define [SD\\_CMD\\_HS\\_SEND\\_EXT\\_CSD](#) 8
- #define [SD\\_CMD\\_SEND\\_CSD](#) 9
- #define [SD\\_CMD\\_SEND\\_CID](#) 10
- #define [SD\\_CMD\\_READ\\_DAT\\_UNTIL\\_STOP](#) 11
- #define [SD\\_CMD\\_STOP\\_TRANSMISSION](#) 12
- #define [SD\\_CMD\\_SEND\\_STATUS](#) 13
- #define [SD\\_CMD\\_HS\\_BUSTEST\\_READ](#) 14
- #define [SD\\_CMD\\_GO\\_INACTIVE\\_STATE](#) 15
- #define [SD\\_CMD\\_SET\\_BLOCKLEN](#) 16
- #define [SD\\_CMD\\_READ\\_SINGLE\\_BLOCK](#) 17

- #define `SD_CMD_READ_MULT_BLOCK` 18
- #define `SD_CMD_HS_BUSTEST_WRITE` 19
- #define `SD_CMD_WRITE_DAT_UNTIL_STOP` 20
- #define `SD_CMD_SET_BLOCK_COUNT` 23
- #define `SD_CMD_WRITE_SINGLE_BLOCK` 24
- #define `SD_CMD_WRITE_MULT_BLOCK` 25
- #define `SD_CMD_PROG_CID` 26
- #define `SD_CMD_PROG_CSD` 27
- #define `SD_CMD_SET_WRITE_PROT` 28
- #define `SD_CMD_CLR_WRITE_PROT` 29
- #define `SD_CMD_SEND_WRITE_PROT` 30
- #define `SD_CMD_SD_ERASE_GRP_START` 32
- #define `SD_CMD_SD_ERASE_GRP_END` 33
- #define `SD_CMD_ERASE_GRP_START` 35
- #define `SD_CMD_ERASE_GRP_END` 36
- #define `SD_CMD_ERASE` 38
- #define `SD_CMD_FAST_IO` 39
- #define `SD_CMD_GO_IRQ_STATE` 40
- #define `SD_CMD_LOCK_UNLOCK` 42
- #define `SD_CMD_APP_CMD` 55
- #define `SD_CMD_GEN_CMD` 56
- #define `SD_CMD_NO_CMD` 64
- #define `SD_CMD_APP_SD_SET_BUSWIDTH` 6
- #define `SD_CMD_SD_APP_STAUS` 13
- #define `SD_CMD_SD_APP_SEND_NUM_WRITE_BLOCKS` 22
- #define `SD_CMD_SD_APP_OP_COND` 41
- #define `SD_CMD_SD_APP_SET_CLR_CARD_DETECT` 42
- #define `SD_CMD_SD_APP_SEND_SCR` 51
- #define `SD_CMD_SDIO_RW_DIRECT` 52
- #define `SD_CMD_SDIO_RW_EXTENDED` 53
- #define `STATUS_READY` 0x0100
- #define `STATUS_ERR_BITS` (SDIO\_STA\_CTIMEOUT|SDIO\_STA\_CCRCFAIL|SDIO\_STA\_DTIMEO←  
UT|SDIO\_STA\_DCRCFAIL|SDIO\_STA\_STBITERR|SDIO\_STA\_RXOVERR|SDIO\_STA\_TXUNDERR)

## Enumerations

- enum `SDCardState` {  
`SD_CARD_READY` = 0x00000001, `SD_CARD_IDENTIFICATION` = 0x00000002, `SD_CARD_STANDBY` =  
0x00000003, `SD_CARD_TRANSFER` = 0x00000004,  
`SD_CARD_SENDING` = 0x00000005, `SD_CARD_RECEIVING` = 0x00000006, `SD_CARD_PROGRAMMING`  
= 0x00000007, `SD_CARD_DISCONNECTED` = 0x00000008,  
`SD_CARD_ERROR` = 0x000000FF }

## Functions

- void `SDIO_IRQHandler` ()
- void `SDMMC2_IRQHandler` ()

## Variables

- static `BSdio * dev`

## 6.82.1 Macro Definition Documentation

### 6.82.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 6.82.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

### 6.82.1.3 SD\_CMD\_ALL\_SEND\_CID

```
#define SD_CMD_ALL_SEND_CID 2
```

### 6.82.1.4 SD\_CMD\_APP\_CMD

```
#define SD_CMD_APP_CMD 55
```

### 6.82.1.5 SD\_CMD\_APP\_SD\_SET\_BUSWIDTH

```
#define SD_CMD_APP_SD_SET_BUSWIDTH 6
```

### 6.82.1.6 SD\_CMD\_CLR\_WRITE\_PROT

```
#define SD_CMD_CLR_WRITE_PROT 29
```

### 6.82.1.7 SD\_CMD\_ERASE

```
#define SD_CMD_ERASE 38
```

**6.82.1.8 SD\_CMD\_ERASE\_GRP\_END**

```
#define SD_CMD_ERASE_GRP_END 36
```

**6.82.1.9 SD\_CMD\_ERASE\_GRP\_START**

```
#define SD_CMD_ERASE_GRP_START 35
```

**6.82.1.10 SD\_CMD\_FAST\_IO**

```
#define SD_CMD_FAST_IO 39
```

**6.82.1.11 SD\_CMD\_GEN\_CMD**

```
#define SD_CMD_GEN_CMD 56
```

**6.82.1.12 SD\_CMD\_GO\_IDLE\_STATE**

```
#define SD_CMD_GO_IDLE_STATE 0
```

**6.82.1.13 SD\_CMD\_GO\_INACTIVE\_STATE**

```
#define SD_CMD_GO_INACTIVE_STATE 15
```

**6.82.1.14 SD\_CMD\_GO\_IRQ\_STATE**

```
#define SD_CMD_GO_IRQ_STATE 40
```

**6.82.1.15 SD\_CMD\_HS\_BUSTEST\_READ**

```
#define SD_CMD_HS_BUSTEST_READ 14
```

**6.82.1.16 SD\_CMD\_HS\_BUSTEST\_WRITE**

```
#define SD_CMD_HS_BUSTEST_WRITE 19
```

**6.82.1.17 SD\_CMD\_HS\_SEND\_EXT\_CSD**

```
#define SD_CMD_HS_SEND_EXT_CSD 8
```

**6.82.1.18 SD\_CMD\_HS\_SWITCH**

```
#define SD_CMD_HS_SWITCH 6
```

**6.82.1.19 SD\_CMD\_LOCK\_UNLOCK**

```
#define SD_CMD_LOCK_UNLOCK 42
```

**6.82.1.20 SD\_CMD\_NO\_CMD**

```
#define SD_CMD_NO_CMD 64
```

**6.82.1.21 SD\_CMD\_PROG\_CID**

```
#define SD_CMD_PROG_CID 26
```

**6.82.1.22 SD\_CMD\_PROG\_CSD**

```
#define SD_CMD_PROG_CSD 27
```

**6.82.1.23 SD\_CMD\_READ\_DAT\_UNTIL\_STOP**

```
#define SD_CMD_READ_DAT_UNTIL_STOP 11
```

**6.82.1.24 SD\_CMD\_READ\_MULT\_BLOCK**

```
#define SD_CMD_READ_MULT_BLOCK 18
```

**6.82.1.25 SD\_CMD\_READ\_SINGLE\_BLOCK**

```
#define SD_CMD_READ_SINGLE_BLOCK 17
```

**6.82.1.26 SD\_CMD\_SD\_APP\_OP\_COND**

```
#define SD_CMD_SD_APP_OP_COND 41
```

**6.82.1.27 SD\_CMD\_SD\_APP\_SEND\_NUM\_WRITE\_BLOCKS**

```
#define SD_CMD_SD_APP_SEND_NUM_WRITE_BLOCKS 22
```

**6.82.1.28 SD\_CMD\_SD\_APP\_SEND\_SCR**

```
#define SD_CMD_SD_APP_SEND_SCR 51
```

**6.82.1.29 SD\_CMD\_SD\_APP\_SET\_CLR\_CARD\_DETECT**

```
#define SD_CMD_SD_APP_SET_CLR_CARD_DETECT 42
```

**6.82.1.30 SD\_CMD\_SD\_APP\_STAUS**

```
#define SD_CMD_SD_APP_STAUS 13
```

**6.82.1.31 SD\_CMD\_SD\_ERASE\_GRP\_END**

```
#define SD_CMD_SD_ERASE_GRP_END 33
```

**6.82.1.32 SD\_CMD\_SD\_ERASE\_GRP\_START**

```
#define SD_CMD_SD_ERASE_GRP_START 32
```

**6.82.1.33 SD\_CMD\_SDIO\_RW\_DIRECT**

```
#define SD_CMD_SDIO_RW_DIRECT 52
```

**6.82.1.34 SD\_CMD\_SDIO\_RW\_EXTENDED**

```
#define SD_CMD_SDIO_RW_EXTENDED 53
```

**6.82.1.35 SD\_CMD\_SDIO\_SEN\_OP\_COND**

```
#define SD_CMD_SDIO_SEN_OP_COND 5
```

**6.82.1.36 SD\_CMD\_SEL\_DESEL\_CARD**

```
#define SD_CMD_SEL_DESEL_CARD 7
```

**6.82.1.37 SD\_CMD\_SEND\_CID**

```
#define SD_CMD_SEND_CID 10
```

**6.82.1.38 SD\_CMD\_SEND\_CSD**

```
#define SD_CMD_SEND_CSD 9
```

**6.82.1.39 SD\_CMD\_SEND\_OP\_COND**

```
#define SD_CMD_SEND_OP_COND 1
```

**6.82.1.40 SD\_CMD\_SEND\_STATUS**

```
#define SD_CMD_SEND_STATUS 13
```

**6.82.1.41 SD\_CMD\_SEND\_WRITE\_PROT**

```
#define SD_CMD_SEND_WRITE_PROT 30
```

**6.82.1.42 SD\_CMD\_SET\_BLOCK\_COUNT**

```
#define SD_CMD_SET_BLOCK_COUNT 23
```

**6.82.1.43 SD\_CMD\_SET\_BLOCKLEN**

```
#define SD_CMD_SET_BLOCKLEN 16
```

**6.82.1.44 SD\_CMD\_SET\_DSR**

```
#define SD_CMD_SET_DSR 4
```

**6.82.1.45 SD\_CMD\_SET\_REL\_ADDR**

```
#define SD_CMD_SET_REL_ADDR 3
```

**6.82.1.46 SD\_CMD\_SET\_WRITE\_PROT**

```
#define SD_CMD_SET_WRITE_PROT 28
```

**6.82.1.47 SD\_CMD\_STOP\_TRANSMISSION**

```
#define SD_CMD_STOP_TRANSMISSION 12
```

#### 6.82.1.48 SD\_CMD\_WRITE\_DAT\_UNTIL\_STOP

```
#define SD_CMD_WRITE_DAT_UNTIL_STOP 20
```

#### 6.82.1.49 SD\_CMD\_WRITE\_MULT\_BLOCK

```
#define SD_CMD_WRITE_MULT_BLOCK 25
```

#### 6.82.1.50 SD\_CMD\_WRITE\_SINGLE\_BLOCK

```
#define SD_CMD_WRITE_SINGLE_BLOCK 24
```

#### 6.82.1.51 STATUS\_ERR\_BITS

```
#define STATUS_ERR_BITS (SDIO_STA_CTIMEOUT|SDIO_STA_CCRCFAIL|SDIO_STA_DTIMEOUT|SDIO_STA_DCRCFAL↵  
IL|SDIO_STA_STBITERR|SDIO_STA_RXOVERR|SDIO_STA_TXUNDERR)
```

#### 6.82.1.52 STATUS\_READY

```
#define STATUS_READY 0x0100
```

#### 6.82.1.53 USE\_DMA

```
#define USE_DMA 1
```

## 6.82.2 Enumeration Type Documentation

### 6.82.2.1 SDCardState

```
enum SDCardState
```

### Enumerator

|                        |  |
|------------------------|--|
| SD_CARD_READY          |  |
| SD_CARD_IDENTIFICATION |  |
| SD_CARD_STANDBY        |  |
| SD_CARD_TRANSFER       |  |
| SD_CARD_SENDING        |  |
| SD_CARD_RECEIVING      |  |
| SD_CARD_PROGRAMMING    |  |
| SD_CARD_DISCONNECTED   |  |
| SD_CARD_ERROR          |  |

## 6.82.3 Function Documentation

### 6.82.3.1 SDIO\_IRQHandler()

```
void SDIO_IRQHandler ( )
```

### 6.82.3.2 SDMMC2\_IRQHandler()

```
void SDMMC2_IRQHandler ( )
```

## 6.82.4 Variable Documentation

### 6.82.4.1 dev

```
BSdio* dev [static]
```

## 6.83 BSdio.h File Reference

```
#include <BTypes.h>  
#include <BGpio.h>  
#include <BError.h>  
#include <BDma.h>  
#include <BMutex.h>  
#include <BSemaphore.h>
```

## Classes

- class [BSdio](#)

## 6.84 BSemaphore.cpp File Reference

```
#include <BSemaphore.h>
#include <BSys.h>
#include <BCritical.h>
```

## 6.85 BSemaphore.h File Reference

```
#include <BTypes.h>
#include <FreeRTOS.h>
#include <semphr.h>
```

## Classes

- class [BSemaphore](#)  
*Semaphore class.*
- class [BSemaphoreBool](#)
- class [BSemaphoreCount](#)

## 6.86 BSnmppServer.cpp File Reference

```
#include <BSnmppServer.h>
#include <BDebug.h>
#include <lwip/apps/snmp.h>
#include <lwip/apps/snmp_mib2.h>
```

## Macros

- `#define BDEBUGL1 0`

## Variables

- static [BSnmppServer](#) \* `osnmppServer`

### 6.86.1 Macro Definition Documentation

### 6.86.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

## 6.86.2 Variable Documentation

### 6.86.2.1 osnmpServer

```
BSnmpServer* osnmpServer [static]
```

## 6.87 BSnmpServer.h File Reference

```
#include <BNetwork.h>
```

### Classes

- class [BSnmpServer](#)

## 6.88 BSpi.cpp File Reference

```
#include <BSpi.h>  
#include <BSys.h>  
#include <BDebug.h>
```

### Functions

- static [BUInt log2](#) ([BUInt v](#))

### 6.88.1 Function Documentation

#### 6.88.1.1 log2()

```
static BUInt log2 (  
    BUInt v ) [static]
```

## 6.89 BSpi.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BGpio.h>
#include <BDma.h>
#include <ArmSystem.h>
```

### Classes

- class [BSpi](#)

## 6.90 BString.cpp File Reference

```
#include <BString.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include <ctype.h>
```

### Functions

- int [bstringListInList](#) ([BStringList](#) &list, [BString](#) s)
- [BString](#) [blistToString](#) (const [BStringList](#) &list)  
*Convert a string list to a comma separated string.*
- [BStringList](#) [bstringToList](#) ([BString](#) str, int stripSpaces)  
*Convert a comma separated string to a string list.*
- const char \* [intToString](#) (char \*str, [BUInt](#) strLen, int value, int base)
- const char \* [floatToString](#) (char \*str, [BUInt](#) strLen, [BFloat32](#) f, [BUInt](#) precision)
- char \* [bstrncpy](#) (char \*dest, const char \*src, size\_t n)
- char \* [bstrtrim](#) (char \*str)

### 6.90.1 Function Documentation

#### 6.90.1.1 [blistToString\(\)](#)

```
BString blistToString (  
    const BStringList & list )
```

Convert a string list to a comma separated string.

### 6.90.1.2 bstringListinList()

```
int bstringListinList (
    BStringList & list,
    BString s )
```

### 6.90.1.3 bstringToList()

```
BStringList bstringToList (
    BString str,
    int stripSpaces )
```

Convert a comma separated string to a string list.

### 6.90.1.4 bstrncpy()

```
char* bstrncpy (
    char * dest,
    const char * src,
    size_t n )
```

### 6.90.1.5 bstrtrim()

```
char* bstrtrim (
    char * str )
```

### 6.90.1.6 floatToString()

```
const char* floatToString (
    char * str,
    BUInt strlen,
    BFloat32 f,
    BUInt precision )
```

### 6.90.1.7 intToString()

```
const char* intToString (
    char * str,
    BUInt strlen,
    int value,
    int base )
```

## 6.91 BString.h File Reference

```
#include <BTypes.h>
#include <BList.h>
#include <ctype.h>
```

### Classes

- struct [BRefData](#)
- class [BRefString](#)
- class [BString](#)

### Typedefs

- typedef [BList](#)< [BString](#) > [BStringList](#)

### Functions

- int [bstringListinList](#) ([BStringList](#) &l, [BString](#) s)
- [BString](#) [blistToString](#) (const [BStringList](#) &list)  
*Convert a string list to a comma separated string.*
- [BStringList](#) [bstringToList](#) ([BString](#) str, int stripSpaces=0)  
*Convert a comma separated string to a string list.*
- char [from\\_hex](#) (char ch)
- char [to\\_hex](#) (char code)
- char \* [bstrncpy](#) (char \*dest, const char \*src, size\_t n)
- char \* [bstrtrim](#) (char \*str)
- const char \* [intToString](#) (char \*str, [BUInt](#) strLen, int value, int base=10)
- const char \* [floatToString](#) (char \*str, [BUInt](#) strLen, [BFloat32](#) f, [BUInt](#) precision)

### 6.91.1 Typedef Documentation

#### 6.91.1.1 BStringList

```
typedef BList<BString> BStringList
```

### 6.91.2 Function Documentation

### 6.91.2.1 blistToString()

```
BString blistToString (
    const BStringList & list )
```

Convert a string list to a comma separated string.

### 6.91.2.2 bstringListinList()

```
int bstringListinList (
    BStringList & l,
    BString s )
```

### 6.91.2.3 bstringToList()

```
BStringList bstringToList (
    BString str,
    int stripSpaces = 0 )
```

Convert a comma separated string to a string list.

### 6.91.2.4 bstrncpy()

```
char* bstrncpy (
    char * dest,
    const char * src,
    size_t n )
```

### 6.91.2.5 bstrtrim()

```
char* bstrtrim (
    char * str )
```

### 6.91.2.6 floatToString()

```
const char* floatToString (
    char * str,
    BUInt strlen,
    BFloat32 f,
    BUInt precision )
```

## 6.91.2.7 from\_hex()

```
char from_hex (
    char ch ) [inline]
```

## 6.91.2.8 intToString()

```
const char* intToString (
    char * str,
    BUInt strlen,
    int value,
    int base = 10 )
```

## 6.91.2.9 to\_hex()

```
char to_hex (
    char code ) [inline]
```

## 6.92 BSys.cpp File Reference

```
#include <BSys.h>
#include <BTask.h>
#include <ArmSystem.h>
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
```

## Functions

- [int \\_version \(\)](#)
- [void xPortSysTickHandler \(\)](#)
- [uint32\\_t timeUs \(\)](#)
- [void delayUsClock \(UInt us\)](#)  
*Hard loop delay to system clock.*
- [void delayUsLoop \(UInt us\)](#)  
*Hard loop delay.*
- [void delayUsTask \(UInt us\)](#)  
*RTOS task delay.*
- [void delayUs \(UInt us\)](#)  
*Will delay for given time in us, if tasks running task will sleep.*
- [void delayMs \(UInt ms\)](#)  
*Will delay for given time in ms, if tasks running task will sleep.*
- [void vApplicationGetIdleTaskMemory \(StaticTask\\_t \\*\\*taskData, StackType\\_t \\*\\*taskStack, uint32\\_t \\*taskStackSize\)](#)

## Variables

- char [\\_stext](#)
- char [\\_etext](#)
- char [\\_sdata](#)
- char [\\_edata](#)
- char [\\_sbss](#)
- char [\\_ebss](#)
- char \* [\\_eheap](#)
- char [\\_estack](#)
- FuncDelay [delayFunc](#) = [delayUsLoop](#)
  - Pointer to function to use for delays, can be set to tasking delay function.*
- BaseType\_t [binterruptYield](#)

## 6.92.1 Function Documentation

### 6.92.1.1 [\\_version\(\)](#)

```
int _version ( )
```

### 6.92.1.2 [delayMs\(\)](#)

```
void delayMs (
    UInt ms )
```

Will delay for given time in ms, if tasks running task will sleep.

### 6.92.1.3 [delayUs\(\)](#)

```
void delayUs (
    UInt us )
```

Will delay for given time in us, if tasks running task will sleep.

### 6.92.1.4 [delayUsClock\(\)](#)

```
void delayUsClock (
    UInt us )
```

Hard loop delay to system clock.

### 6.92.1.5 delayUsLoop()

```
void delayUsLoop (
    UInt us )
```

Hard loop delay.

### 6.92.1.6 delayUsTask()

```
void delayUsTask (
    UInt us )
```

RTOS task delay.

### 6.92.1.7 timeUs()

```
uint32_t timeUs ( )
```

### 6.92.1.8 vApplicationGetIdleTaskMemory()

```
void vApplicationGetIdleTaskMemory (
    StaticTask_t ** taskData,
    StackType_t ** taskStack,
    uint32_t * taskStackSize )
```

### 6.92.1.9 xPortSysTickHandler()

```
void xPortSysTickHandler ( )
```

## 6.92.2 Variable Documentation

### 6.92.2.1 \_ebss

```
char _ebss
```

**6.92.2.2** `_edata``char _edata`**6.92.2.3** `_eheap``char* _eheap`**6.92.2.4** `_estack``char _estack`**6.92.2.5** `_etext``char _etext`**6.92.2.6** `_sbss``char _sbss`**6.92.2.7** `_sdata``char _sdata`**6.92.2.8** `_stext``char _stext`**6.92.2.9** `binterruptYield``BaseType_t binterruptYield`

## 6.92.2.10 delayFunc

```
FuncDelay delayFunc = delayUsLoop
```

Pointer to function to use for delays, can be set to tasking delay function.

## 6.93 BSys.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BDevices.h>
#include <BInterrupt.h>
#include <BTim.h>
#include <BHeap.h>
#include <FreeRTOS.h>
#include <task.h>
```

## Classes

- class [BSys](#)

## Macros

- #define [binterruptStart\(\)](#) `Bool oldYield = binterruptYield; binterruptYield = 0;`
- #define [binterruptEnd\(\)](#) `if(binterruptYield){ binterruptYield = oldYield; portYIELD_FROM_ISR(1); } else { binterruptYield = oldYield; }`

## Typedefs

- typedef void(\* [FuncRun](#)) ()
- typedef void(\* [FuncTask](#)) (void \*arg)
- typedef void(\* [FuncDelay](#)) (UInt us)

## Functions

- `uint32_t` [timeUs](#) ()
- void [delayUs](#) (UInt us)  
*Will delay for given time in us, if tasks running task will sleep.*
- void [delayMs](#) (UInt ms)  
*Will delay for given time in ms, if tasks running task will sleep.*
- void [delayUsTask](#) (UInt us)  
*RTOS task delay.*
- void [delayUsClock](#) (UInt us)  
*Hard loop delay to system clock.*
- void [delayUsLoop](#) (UInt us)  
*Hard loop delay.*
- void [binterruptEnable](#) ()
- void [binterruptDisable](#) ()
- `Bool` [binterruptInside](#) ()
- void [bsysTaskYield](#) ()

## Variables

- const int [BSystemTimer](#) = 2
- [BSystem](#) sys
- [FuncDelay](#) delayFunc  
*Pointer to function to use for delays, can be set to tasking delay function.*
- BaseType\_t [bInterruptYield](#)

## 6.93.1 Macro Definition Documentation

### 6.93.1.1 bInterruptEnd

```
#define bInterruptEnd( ) if(bInterruptYield){ bInterruptYield = oldYield; portYIELD_FROM_ISR(1); } else { bInterruptYield = oldYield; }
```

### 6.93.1.2 bInterruptStart

```
#define bInterruptStart( ) Bool oldYield = bInterruptYield; bInterruptYield = 0;
```

## 6.93.2 Typedef Documentation

### 6.93.2.1 FuncDelay

```
typedef void(* FuncDelay) (UInt us)
```

### 6.93.2.2 FuncRun

```
typedef void(* FuncRun) ()
```

### 6.93.2.3 FuncTask

```
typedef void(* FuncTask) (void *arg)
```

### 6.93.3 Function Documentation

#### 6.93.3.1 binterruptDisable()

```
void binterruptDisable ( ) [inline]
```

#### 6.93.3.2 binterruptEnable()

```
void binterruptEnable ( ) [inline]
```

#### 6.93.3.3 binterruptInside()

```
Bool binterruptInside ( ) [inline]
```

#### 6.93.3.4 bsysTaskYield()

```
void bsysTaskYield ( ) [inline]
```

#### 6.93.3.5 delayMs()

```
void delayMs (
    UInt ms )
```

Will delay for given time in ms, if tasks running task will sleep.

#### 6.93.3.6 delayUs()

```
void delayUs (
    UInt us )
```

Will delay for given time in us, if tasks running task will sleep.

#### 6.93.3.7 delayUsClock()

```
void delayUsClock (
    UInt us )
```

Hard loop delay to system clock.

#### 6.93.3.8 delayUsLoop()

```
void delayUsLoop (
    UInt us )
```

Hard loop delay.

#### 6.93.3.9 delayUsTask()

```
void delayUsTask (
    UInt us )
```

RTOS task delay.

#### 6.93.3.10 timeUs()

```
uint32_t timeUs ( )
```

### 6.93.4 Variable Documentation

#### 6.93.4.1 binterruptYield

```
BaseType_t binterruptYield
```

#### 6.93.4.2 BSysTimer

```
const int BSysTimer = 2
```

### 6.93.4.3 delayFunc

[FuncDelay](#) delayFunc

Pointer to function to use for delays, can be set to tasking delay function.

### 6.93.4.4 sys

[BSys](#) sys

## 6.94 BTask.cpp File Reference

```
#include <BTask.h>
#include <BSys.h>
#include <cmsis_os.h>
```

### Functions

- const int [\\_\\_attribute\\_\\_](#) ((used)) uxTopUsedPriority
- static void [taskDelayUs](#) (UInt us)
- static void [taskFunc](#) (void \*arg)
- void [vApplicationIdleHook](#) (void)
- static unsigned portBASE\_TYPE [makeFreeRtosPriority](#) (osPriority priority)
- osThreadId [osThreadCreate](#) (const osThreadDef\_t \*thread\_def, void \*argument)

### 6.94.1 Function Documentation

#### 6.94.1.1 [\\_\\_attribute\\_\\_](#)()

```
const int \_\_attribute\_\_ (
    (used) )
```

#### 6.94.1.2 [makeFreeRtosPriority](#)()

```
static unsigned portBASE_TYPE makeFreeRtosPriority (
    osPriority priority ) [static]
```

### 6.94.1.3 osThreadCreate()

```
osThreadId osThreadCreate (
    const osThreadDef_t * thread_def,
    void * argument )
```

### 6.94.1.4 taskDelayUs()

```
static void taskDelayUs (
    UInt us ) [static]
```

### 6.94.1.5 taskFunc()

```
static void taskFunc (
    void * arg ) [static]
```

### 6.94.1.6 vApplicationIdleHook()

```
void vApplicationIdleHook (
    void )
```

## 6.95 BTask.h File Reference

```
#include <BError.h>
#include <FreeRTOS.h>
#include <task.h>
```

### Classes

- class [BTask](#)

## 6.96 BTim.cpp File Reference

```
#include <BTim.h>
#include <BSys.h>
#include <BGpio.h>
#include <BDebug.h>
```

## Variables

- const BUInt TIM\_CCER\_CCE = 0x01
- const BUInt TIM\_CCMR1\_OCCE = 0x80

### 6.96.1 Variable Documentation

#### 6.96.1.1 TIM\_CCER\_CCE

```
const BUInt TIM_CCER_CCE = 0x01
```

#### 6.96.1.2 TIM\_CCMR1\_OCCE

```
const BUInt TIM_CCMR1_OCCE = 0x80
```

## 6.97 BTim.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <ArmSystem.h>
```

## Classes

- class BTim

## 6.98 BTime.cpp File Reference

```
#include <BTime.h>  
#include <stdio.h>
```

## Functions

- static bool `yearIsLeap` (BUInt16 year)
- static BUInt16 `yearDays` (BUInt16 year)

## Variables

- static BUInt16 `monDays` [2][13]

## 6.98.1 Function Documentation

### 6.98.1.1 yearDays()

```
static BUInt16 yearDays (  
    BUInt16 year ) [inline], [static]
```

### 6.98.1.2 yearIsLeap()

```
static bool yearIsLeap (  
    BUInt16 year ) [inline], [static]
```

## 6.98.2 Variable Documentation

### 6.98.2.1 monDays

```
BUInt16 monDays[2][13] [static]
```

#### Initial value:

```
= {  
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },  
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }  
}
```

## 6.99 BTime.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BString.h>
```

### Classes

- class [BTime](#)

## 6.100 BTimeUs.cpp File Reference

```
#include <BTimeUs.h>  
#include <stdio.h>
```

## Functions

- static bool `yearIsLeap` (`BUInt16` year)
- static `BUInt16` `yearDays` (`BUInt16` year)

## Variables

- static `BUInt16` `monDays` [2][13]

### 6.100.1 Function Documentation

#### 6.100.1.1 `yearDays()`

```
static BUInt16 yearDays (  
    BUInt16 year ) [inline], [static]
```

#### 6.100.1.2 `yearIsLeap()`

```
static bool yearIsLeap (  
    BUInt16 year ) [inline], [static]
```

### 6.100.2 Variable Documentation

#### 6.100.2.1 `monDays`

```
BUInt16 monDays[2][13] [static]
```

#### Initial value:

```
= {  
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },  
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }  
}
```

## 6.101 BTimeUs.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BString.h>
```

## Classes

- class [BTimeUs](#)

## 6.102 BTypes.h File Reference

```
#include <stdint.h>
```

## Classes

- class [BDataChunk](#)
- struct [BObjMember](#)

## Macros

- #define [BBigEndian](#) 0
- #define [M\\_PI](#) 3.14159265358979323846

## Typedefs

- typedef bool [Bool](#)
- typedef int8\_t [BInt8](#)
- typedef uint8\_t [BUInt8](#)
- typedef int16\_t [BInt16](#)
- typedef uint16\_t [BUInt16](#)
- typedef int32\_t [BInt32](#)
- typedef uint32\_t [BUInt32](#)
- typedef int64\_t [BInt64](#)
- typedef uint64\_t [BUInt64](#)
- typedef float [BFloat32](#)
- typedef double [BFloat64](#)
- typedef char [BChar](#)
- typedef [BInt32](#) [BInt](#)
- typedef [BUInt32](#) [BUInt](#)
- typedef [BFloat32](#) [BFloat](#)
- typedef [BFloat64](#) [BDouble](#)
- typedef int8\_t [Int8](#)
- typedef uint8\_t [UInt8](#)
- typedef int16\_t [Int16](#)
- typedef uint16\_t [UInt16](#)
- typedef int32\_t [Int32](#)
- typedef uint32\_t [UInt32](#)
- typedef int64\_t [Int64](#)
- typedef uint64\_t [UInt64](#)
- typedef float [Float32](#)
- typedef double [Float64](#)
- typedef [Int32](#) [Int](#)
- typedef [UInt32](#) [UInt](#)
- typedef [Float32](#) [Float](#)
- typedef [Float64](#) [Double](#)
- typedef [BUInt32](#) [BTimeout](#)

## Enumerations

- enum `BEventType` {  
`BEventTypeNone`, `BEventTypeError`, `BEventTypeRead`, `BEventTypeWrite`,  
`BEventTypeDisconnect` }
- enum `BEventWaitSet` {  
`BEventWaitNone` = 0x00, `BEventWaitError` = 0x01, `BEventWaitRead` = 0x02, `BEventWaitWrite` = 0x04,  
`BEventWaitDisconnect` = 0x08, `BEventWaitAny` = 0xFFFFFFFF }
- enum `BUsbType` { `BUsbTypeNone`, `BUsbTypeFullSpeed`, `BUsbTypeHighSpeed` }
- enum `BType` {  
`BTypeNone`, `BTypeBool`, `BTypeInt8`, `BTypeUInt8`,  
`BTypeInt16`, `BTypeUInt16`, `BTypeInt32`, `BTypeUInt32`,  
`BTypeInt64`, `BTypeUInt64`, `BTypeFloat32`, `BTypeFloat64`,  
`BTypeChar`, `BTypeString`, `BTypeError`, `BTypeTime`,  
`BTypeObj` = 100 }
- enum `BTypeComp` {  
`BTypeCompSingle`, `BTypeCompArray`, `BTypeCompArrayFixed`, `BTypeCompList`,  
`BTypeCompDict` }

## Functions

- `BTimeout timeoutTicks` (`BTimeout timeoutUs`)
- `BUInt16 bswap_16` (`BUInt16 v`)
- `BUInt32 bswap_32` (`BUInt32 v`)
- `BUInt64 bswap_p64` (`BUInt64 v`)
- void `bswap_copy` (int swap, const void \*src, void \*dst, `BUInt32` nBytes, const char \*swapType)

## Variables

- const `BTimeout BTimeoutForever` = 0xFFFFFFFF

### 6.102.1 Macro Definition Documentation

#### 6.102.1.1 BBigEndian

```
#define BBigEndian 0
```

#### 6.102.1.2 M\_PI

```
#define M_PI 3.14159265358979323846
```

### 6.102.2 Typedef Documentation

#### 6.102.2.1 BChar

```
typedef char BChar
```

#### 6.102.2.2 BDouble

```
typedef BFloat64 BDouble
```

#### 6.102.2.3 BFloat

```
typedef BFloat32 BFloat
```

#### 6.102.2.4 BFloat32

```
typedef float BFloat32
```

#### 6.102.2.5 BFloat64

```
typedef double BFloat64
```

#### 6.102.2.6 BInt

```
typedef BInt32 BInt
```

#### 6.102.2.7 BInt16

```
typedef int16_t BInt16
```

#### 6.102.2.8 BInt32

```
typedef int32_t BInt32
```

### 6.102.2.9 BInt64

```
typedef int64_t BInt64
```

### 6.102.2.10 BInt8

```
typedef int8_t BInt8
```

### 6.102.2.11 Bool

```
typedef bool Bool
```

### 6.102.2.12 BTimeout

```
typedef BUInt32 BTimeout
```

### 6.102.2.13 BUInt

```
typedef BUInt32 BUInt
```

### 6.102.2.14 BUInt16

```
typedef uint16_t BUInt16
```

### 6.102.2.15 BUInt32

```
typedef uint32_t BUInt32
```

### 6.102.2.16 BUInt64

```
typedef uint64_t BUInt64
```

**6.102.2.17 BUInt8**

```
typedef uint8_t BUInt8
```

**6.102.2.18 Double**

```
typedef Float64 Double
```

**6.102.2.19 Float**

```
typedef Float32 Float
```

**6.102.2.20 Float32**

```
typedef float Float32
```

**6.102.2.21 Float64**

```
typedef double Float64
```

**6.102.2.22 Int**

```
typedef Int32 Int
```

**6.102.2.23 Int16**

```
typedef int16_t Int16
```

**6.102.2.24 Int32**

```
typedef int32_t Int32
```

#### 6.102.2.25 Int64

```
typedef int64_t Int64
```

#### 6.102.2.26 Int8

```
typedef int8_t Int8
```

#### 6.102.2.27 UInt

```
typedef UInt32 UInt
```

#### 6.102.2.28 UInt16

```
typedef uint16_t UInt16
```

#### 6.102.2.29 UInt32

```
typedef uint32_t UInt32
```

#### 6.102.2.30 UInt64

```
typedef uint64_t UInt64
```

#### 6.102.2.31 UInt8

```
typedef uint8_t UInt8
```

### 6.102.3 Enumeration Type Documentation

#### 6.102.3.1 BEventType

```
enum BEventType
```

## Enumerator

|                      |  |
|----------------------|--|
| BEventTypeNone       |  |
| BEventTypeError      |  |
| BEventTypeRead       |  |
| BEventTypeWrite      |  |
| BEventTypeDisconnect |  |

## 6.102.3.2 BEventWaitSet

```
enum BEventWaitSet
```

## Enumerator

|                      |  |
|----------------------|--|
| BEventWaitNone       |  |
| BEventWaitError      |  |
| BEventWaitRead       |  |
| BEventWaitWrite      |  |
| BEventWaitDisconnect |  |
| BEventWaitAny        |  |

## 6.102.3.3 BType

```
enum BType
```

## Enumerator

|              |  |
|--------------|--|
| BTypeNone    |  |
| BTypeBool    |  |
| BTypeInt8    |  |
| BTypeUInt8   |  |
| BTypeInt16   |  |
| BTypeUInt16  |  |
| BTypeInt32   |  |
| BTypeUInt32  |  |
| BTypeInt64   |  |
| BTypeUInt64  |  |
| BTypeFloat32 |  |
| BTypeFloat64 |  |
| BTypeChar    |  |
| BTypeString  |  |
| BTypeError   |  |
| BTypeTime    |  |
| BTypeObj     |  |

### 6.102.3.4 BTypeComp

enum [BTypeComp](#)

#### Enumerator

|                     |  |
|---------------------|--|
| BTypeCompSingle     |  |
| BTypeCompArray      |  |
| BTypeCompArrayFixed |  |
| BTypeCompList       |  |
| BTypeCompDict       |  |

### 6.102.3.5 BUsbType

enum [BUsbType](#)

#### Enumerator

|                   |  |
|-------------------|--|
| BUsbTypeNone      |  |
| BUsbTypeFullSpeed |  |
| BUsbTypeHighSpeed |  |

## 6.102.4 Function Documentation

### 6.102.4.1 bswap\_16()

```
BUInt16 bswap_16 (  
    BUInt16 v ) [inline]
```

### 6.102.4.2 bswap\_32()

```
BUInt32 bswap_32 (  
    BUInt32 v ) [inline]
```

#### 6.102.4.3 bswap\_copy()

```
void bswap_copy (
    int swap,
    const void * src,
    void * dst,
    BUInt32 nBytes,
    const char * swapType )
```

#### 6.102.4.4 bswap\_p64()

```
BUInt64 bswap_p64 (
    BUInt64 v ) [inline]
```

#### 6.102.4.5 timeoutTicks()

```
BTimeout timeoutTicks (
    BTimeout timeoutUs ) [inline]
```

### 6.102.5 Variable Documentation

#### 6.102.5.1 BTimeoutForever

```
const BTimeout BTimeoutForever = 0xFFFFFFFF
```

## 6.103 BUart.cpp File Reference

```
#include <BUart.h>
#include <BSys.h>
#include <stdio.h>
```

### Functions

- void [USART1\\_IRQHandler](#) ()
- void [USART2\\_IRQHandler](#) ()
- void [USART3\\_IRQHandler](#) ()
- void [UART4\\_IRQHandler](#) ()
- void [UART5\\_IRQHandler](#) ()
- void [USART6\\_IRQHandler](#) ()

### 6.103.1 Function Documentation

#### 6.103.1.1 UART4\_IRQHandler()

```
void UART4_IRQHandler ( )
```

#### 6.103.1.2 UART5\_IRQHandler()

```
void UART5_IRQHandler ( )
```

#### 6.103.1.3 USART1\_IRQHandler()

```
void USART1_IRQHandler ( )
```

#### 6.103.1.4 USART2\_IRQHandler()

```
void USART2_IRQHandler ( )
```

#### 6.103.1.5 USART3\_IRQHandler()

```
void USART3_IRQHandler ( )
```

#### 6.103.1.6 USART6\_IRQHandler()

```
void USART6_IRQHandler ( )
```

## 6.104 BUart.h File Reference

```
#include <BComms.h>  
#include <BFifo.h>  
#include <BGpio.h>  
#include <BSemaphore.h>  
#include <ArmSystem.h>
```

## Classes

- class [BUart](#)

## 6.105 BUartBasic.cpp File Reference

```
#include <BUartBasic.h>
#include <BSys.h>
#include <bstdio.h>
```

## Functions

- static int [uartWrite](#) (const char \**data*, int *nBytes*)

## Variables

- static [BUartBasic](#) \* *uart*

### 6.105.1 Function Documentation

#### 6.105.1.1 [uartWrite\(\)](#)

```
static int uartWrite (
    const char * data,
    int nBytes ) [static]
```

### 6.105.2 Variable Documentation

#### 6.105.2.1 [uart](#)

```
BUartBasic* uart [static]
```

## 6.106 BUartBasic.h File Reference

```
#include <BComms.h>
#include <BGpio.h>
#include <ArmSystem.h>
```

## Classes

- class [BUartBasic](#)

## 6.107 BUsbSerial.cpp File Reference

```
#include <BUsbSerial.h>
#include <BSys.h>
#include <bstdio.h>
#include <BDebug.h>
#include <usbd_conf.h>
#include <usbd_def.h>
#include <usbd_ioreq.h>
```

## Macros

- #define [L1DEBUG](#) 0
- #define [L2DEBUG](#) 0
- #define [L3DEBUG](#) 0
- #define [dl1printf](#)(fmt, a...)
- #define [dl2printf](#)(fmt, a...)
- #define [dl3printf](#)(fmt, a...)
- #define [USB\\_D\\_VID](#) busbConfig.vendorId
- #define [USB\\_D\\_PID](#) busbConfig.productId
- #define [USB\\_D\\_LANGID\\_STRING](#) 0x409
- #define [USB\\_D\\_MANUFACTURER\\_STRING](#) busbConfig.manufacturer
- #define [USB\\_D\\_PRODUCT\\_HS\\_STRING](#) busbConfig.product
- #define [USB\\_D\\_SERIALNUMBER\\_HS\\_STRING](#) busbConfig.serialNumber
- #define [USB\\_D\\_PRODUCT\\_FS\\_STRING](#) busbConfig.product
- #define [USB\\_D\\_SERIALNUMBER\\_FS\\_STRING](#) busbConfig.serialNumber
- #define [USB\\_D\\_CONFIGURATION\\_HS\\_STRING](#) "VCP Config"
- #define [USB\\_D\\_INTERFACE\\_HS\\_STRING](#) "VCP Interface"
- #define [USB\\_D\\_CONFIGURATION\\_FS\\_STRING](#) "VCP Config"
- #define [USB\\_D\\_INTERFACE\\_FS\\_STRING](#) "VCP Interface"

## Functions

- [BUInt8 \\* USB\\_D\\_VCP\\_DeviceDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_LangIDStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_ProductStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_ManufacturerStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_SerialStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_ConfigStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- [BUInt8 \\* USB\\_D\\_VCP\\_InterfaceStrDescriptor](#) (USB\_D\_SpeedTypeDef speed, uint16\_t \*length)
- void [HAL\\_PCD\\_SetupStageCallback](#) (PCD\_HandleTypeDef \*hpcd)
- void [HAL\\_PCD\\_DataOutStageCallback](#) (PCD\_HandleTypeDef \*hpcd, [BUInt8](#) epnum)
- void [HAL\\_PCD\\_DataInStageCallback](#) (PCD\_HandleTypeDef \*hpcd, [BUInt8](#) epnum)
- void [HAL\\_PCD\\_SOFCallback](#) (PCD\_HandleTypeDef \*hpcd)
- void [HAL\\_PCD\\_ResetCallback](#) (PCD\_HandleTypeDef \*hpcd)
- void [HAL\\_PCD\\_SuspendCallback](#) (PCD\_HandleTypeDef \*hpcd)

- void [HAL\\_PCD\\_ResumeCallback](#) (PCD\_HandleTypeDef \*hpcd)
- void [HAL\\_PCD\\_ISOOUTIncompleteCallback](#) (PCD\_HandleTypeDef \*hpcd, [BUInt8](#) epnum)
- void [HAL\\_PCD\\_ISOINIncompleteCallback](#) (PCD\_HandleTypeDef \*hpcd, [BUInt8](#) epnum)
- void [HAL\\_PCD\\_ConnectCallback](#) (PCD\_HandleTypeDef \*hpcd)
- void [HAL\\_PCD\\_DisconnectCallback](#) (PCD\_HandleTypeDef \*hpcd)
- USBD\_StatusTypeDef [USB\\_LL\\_DeInit](#) (USB\_HandleTypeDef \*pdev)
- USBD\_StatusTypeDef [USB\\_LL\\_Start](#) (USB\_HandleTypeDef \*pdev)
- USBD\_StatusTypeDef [USB\\_LL\\_Stop](#) (USB\_HandleTypeDef \*pdev)
- USBD\_StatusTypeDef [USB\\_LL\\_OpenEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr, [BUInt8](#) ep\_type, uint16\_t ep\_mps)
- USBD\_StatusTypeDef [USB\\_LL\\_CloseEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- USBD\_StatusTypeDef [USB\\_LL\\_FlushEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- USBD\_StatusTypeDef [USB\\_LL\\_StallEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- USBD\_StatusTypeDef [USB\\_LL\\_ClearStallEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- [BUInt8](#) [USB\\_LL\\_IsStallEP](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- USBD\_StatusTypeDef [USB\\_LL\\_SetUSBAddress](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) dev\_addr)
- USBD\_StatusTypeDef [USB\\_LL\\_Transmit](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr, [BUInt8](#) \*pbuf, uint16\_t size)
- USBD\_StatusTypeDef [USB\\_LL\\_PrepareReceive](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr, [BUInt8](#) \*pbuf, uint16\_t size)
- uint32\_t [USB\\_LL\\_GetRxDataSize](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) ep\_addr)
- void [USB\\_LL\\_Delay](#) (uint32\_t Delay)
- USBD\_StatusTypeDef [USB\\_LL\\_Init](#) (USB\_HandleTypeDef \*pdev)
- void [OTG\\_FS\\_IRQHandler](#) (void)
- void [OTG\\_HS\\_IRQHandler](#) (void)
- static [BUInt8](#) [usbInit](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) cfgidx)
- static [BUInt8](#) [usbDeInit](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) cfgidx)
- static [BUInt8](#) [usbSetup](#) (USB\_HandleTypeDef \*pdev, USB\_SetupReqTypedef \*req)
- static [BUInt8](#) [usbRx0](#) (USB\_HandleTypeDef \*pdev)
- static [BUInt8](#) [usbRx](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) epnum)
- static [BUInt8](#) [usbTxSent](#) (USB\_HandleTypeDef \*pdev, [BUInt8](#) epnum)
- static [BUInt8](#) \* [usbGetHSCfgDesc](#) (uint16\_t \*length)
- static [BUInt8](#) \* [usbGetFSCfgDesc](#) (uint16\_t \*length)
- static [BUInt8](#) \* [usbGetOtherSpeedCfgDesc](#) (uint16\_t \*length)
- static [BUInt8](#) \* [usbGetDeviceQualifierDescriptor](#) (uint16\_t \*length)
- static int [usbWrite](#) (const char \*data, int nBytes)

## Variables

- const int [PacketSizeHighSpeed](#) = 512
- const int [PacketSizeFullSpeed](#) = 64
- const int [PacketSizeCmd](#) = 8
- const int [ConfigDescriptorSize](#) = 67
- const int [EndPointCmd](#) = 0x82
- const int [EndPointTx](#) = 0x81
- const int [EndPointRx](#) = 0x01
- static [BUsbConfig](#) [busbConfig](#)
- [\\_\\_ALIGN\\_BEGIN](#) [BUInt8](#) [hUSBDDeviceDesc](#) [[USB\\_LEN\\_DEV\\_DESC](#)] [\\_\\_ALIGN\\_END](#)
- USBD\_DescriptorsTypeDef [VCP\\_Desc](#)
- static [BUsbSerial](#) \* [usbSerial](#)
- USBD\_ClassTypeDef [usbClass](#)

## 6.107.1 Macro Definition Documentation

### 6.107.1.1 dl1printf

```
#define dl1printf(  
    fmt,  
    a... )
```

### 6.107.1.2 dl2printf

```
#define dl2printf(  
    fmt,  
    a... )
```

### 6.107.1.3 dl3printf

```
#define dl3printf(  
    fmt,  
    a... )
```

### 6.107.1.4 L1DEBUG

```
#define L1DEBUG 0
```

### 6.107.1.5 L2DEBUG

```
#define L2DEBUG 0
```

### 6.107.1.6 L3DEBUG

```
#define L3DEBUG 0
```

**6.107.1.7 USBD\_CONFIGURATION\_FS\_STRING**

```
#define USBD_CONFIGURATION_FS_STRING "VCP Config"
```

**6.107.1.8 USBD\_CONFIGURATION\_HS\_STRING**

```
#define USBD_CONFIGURATION_HS_STRING "VCP Config"
```

**6.107.1.9 USBD\_INTERFACE\_FS\_STRING**

```
#define USBD_INTERFACE_FS_STRING "VCP Interface"
```

**6.107.1.10 USBD\_INTERFACE\_HS\_STRING**

```
#define USBD_INTERFACE_HS_STRING "VCP Interface"
```

**6.107.1.11 USBD\_LANGID\_STRING**

```
#define USBD_LANGID_STRING 0x409
```

**6.107.1.12 USBD\_MANUFACTURER\_STRING**

```
#define USBD_MANUFACTURER_STRING busbConfig.manufacturer
```

**6.107.1.13 USBD\_PID**

```
#define USBD_PID busbConfig.productId
```

**6.107.1.14 USBD\_PRODUCT\_FS\_STRING**

```
#define USBD_PRODUCT_FS_STRING busbConfig.product
```

#### 6.107.1.15 USBD\_PRODUCT\_HS\_STRING

```
#define USBD_PRODUCT_HS_STRING busbConfig.product
```

#### 6.107.1.16 USBD\_SERIALNUMBER\_FS\_STRING

```
#define USBD_SERIALNUMBER_FS_STRING busbConfig.serialNumber
```

#### 6.107.1.17 USBD\_SERIALNUMBER\_HS\_STRING

```
#define USBD_SERIALNUMBER_HS_STRING busbConfig.serialNumber
```

#### 6.107.1.18 USBD\_VID

```
#define USBD_VID busbConfig.vendorId
```

### 6.107.2 Function Documentation

#### 6.107.2.1 HAL\_PCD\_ConnectCallback()

```
void HAL_PCD_ConnectCallback (  
    PCD_HandleTypeDef * hpcd )
```

#### 6.107.2.2 HAL\_PCD\_DataInStageCallback()

```
void HAL_PCD_DataInStageCallback (  
    PCD_HandleTypeDef * hpcd,  
    BUInt8 epnum )
```

#### 6.107.2.3 HAL\_PCD\_DataOutStageCallback()

```
void HAL_PCD_DataOutStageCallback (  
    PCD_HandleTypeDef * hpcd,  
    BUInt8 epnum )
```

#### 6.107.2.4 HAL\_PCD\_DisconnectCallback()

```
void HAL_PCD_DisconnectCallback (
    PCD_HandleTypeDef * hpcd )
```

#### 6.107.2.5 HAL\_PCD\_ISOINIncompleteCallback()

```
void HAL_PCD_ISOINIncompleteCallback (
    PCD_HandleTypeDef * hpcd,
    BUInt8 epnum )
```

#### 6.107.2.6 HAL\_PCD\_ISOOUTIncompleteCallback()

```
void HAL_PCD_ISOOUTIncompleteCallback (
    PCD_HandleTypeDef * hpcd,
    BUInt8 epnum )
```

#### 6.107.2.7 HAL\_PCD\_ResetCallback()

```
void HAL_PCD_ResetCallback (
    PCD_HandleTypeDef * hpcd )
```

#### 6.107.2.8 HAL\_PCD\_ResumeCallback()

```
void HAL_PCD_ResumeCallback (
    PCD_HandleTypeDef * hpcd )
```

#### 6.107.2.9 HAL\_PCD\_SetupStageCallback()

```
void HAL_PCD_SetupStageCallback (
    PCD_HandleTypeDef * hpcd )
```

#### 6.107.2.10 HAL\_PCD\_SOFCallback()

```
void HAL_PCD_SOFCallback (
    PCD_HandleTypeDef * hpcd )
```

**6.107.2.11 HAL\_PCD\_SuspendCallback()**

```
void HAL_PCD_SuspendCallback (
    PCD_HandleTypeDef * hpcd )
```

**6.107.2.12 OTG\_FS\_IRQHandler()**

```
void OTG_FS_IRQHandler (
    void )
```

**6.107.2.13 OTG\_HS\_IRQHandler()**

```
void OTG_HS_IRQHandler (
    void )
```

**6.107.2.14 USBD\_LL\_ClearStallEP()**

```
USBD_StatusTypeDef USBD_LL_ClearStallEP (
    USBD_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

**6.107.2.15 USBD\_LL\_CloseEP()**

```
USBD_StatusTypeDef USBD_LL_CloseEP (
    USBD_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

**6.107.2.16 USBD\_LL\_DeInit()**

```
USBD_StatusTypeDef USBD_LL_DeInit (
    USBD_HandleTypeDef * pdev )
```

**6.107.2.17 USBD\_LL\_Delay()**

```
void USBD_LL_Delay (
    uint32_t Delay )
```

#### 6.107.2.18 USBD\_LL\_FlushEP()

```
USB_StatusTypeDef USBD_LL_FlushEP (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

#### 6.107.2.19 USBD\_LL\_GetRxDataSize()

```
uint32_t USBD_LL_GetRxDataSize (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

#### 6.107.2.20 USBD\_LL\_Init()

```
USB_StatusTypeDef USBD_LL_Init (
    USB_HandleTypeDef * pdev )
```

#### 6.107.2.21 USBD\_LL\_IsStallEP()

```
BUInt8 USBD_LL_IsStallEP (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

#### 6.107.2.22 USBD\_LL\_OpenEP()

```
USB_StatusTypeDef USBD_LL_OpenEP (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr,
    BUInt8 ep_type,
    uint16_t ep_mps )
```

#### 6.107.2.23 USBD\_LL\_PrepareReceive()

```
USB_StatusTypeDef USBD_LL_PrepareReceive (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr,
    BUInt8 * pbuf,
    uint16_t size )
```

**6.107.2.24 USBD\_LL\_SetUSBAddress()**

```
USB_StatusTypeDef USBD_LL_SetUSBAddress (
    USB_HandleTypeDef * pdev,
    BUInt8 dev_addr )
```

**6.107.2.25 USBD\_LL\_StallEP()**

```
USB_StatusTypeDef USBD_LL_StallEP (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr )
```

**6.107.2.26 USBD\_LL\_Start()**

```
USB_StatusTypeDef USBD_LL_Start (
    USB_HandleTypeDef * pdev )
```

**6.107.2.27 USBD\_LL\_Stop()**

```
USB_StatusTypeDef USBD_LL_Stop (
    USB_HandleTypeDef * pdev )
```

**6.107.2.28 USBD\_LL\_Transmit()**

```
USB_StatusTypeDef USBD_LL_Transmit (
    USB_HandleTypeDef * pdev,
    BUInt8 ep_addr,
    BUInt8 * pbuf,
    uint16_t size )
```

**6.107.2.29 USBD\_VCP\_ConfigStrDescriptor()**

```
BUInt8* USBD_VCP_ConfigStrDescriptor (
    USB_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.30 USBD\_VCP\_DeviceDescriptor()**

```
BUInt8* USBD_VCP_DeviceDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.31 USBD\_VCP\_InterfaceStrDescriptor()**

```
BUInt8* USBD_VCP_InterfaceStrDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.32 USBD\_VCP\_LangIDStrDescriptor()**

```
BUInt8* USBD_VCP_LangIDStrDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.33 USBD\_VCP\_ManufacturerStrDescriptor()**

```
BUInt8* USBD_VCP_ManufacturerStrDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.34 USBD\_VCP\_ProductStrDescriptor()**

```
BUInt8* USBD_VCP_ProductStrDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.35 USBD\_VCP\_SerialStrDescriptor()**

```
BUInt8* USBD_VCP_SerialStrDescriptor (
    USBD_SpeedTypeDef speed,
    uint16_t * length )
```

**6.107.2.36 usbdDelInit()**

```
static BUInt8 usbdDeInit (
    USBD_HandleTypeDef * pdev,
    BUInt8 cfgidx ) [static]
```

**6.107.2.37 usbdGetDeviceQualifierDescriptor()**

```
static BUInt8* usbdGetDeviceQualifierDescriptor (
    uint16_t * length ) [static]
```

**6.107.2.38 usbdGetFSCfgDesc()**

```
static BUInt8* usbdGetFSCfgDesc (
    uint16_t * length ) [static]
```

**6.107.2.39 usbdGetHSCfgDesc()**

```
static BUInt8* usbdGetHSCfgDesc (
    uint16_t * length ) [static]
```

**6.107.2.40 usbdGetOtherSpeedCfgDesc()**

```
static BUInt8* usbdGetOtherSpeedCfgDesc (
    uint16_t * length ) [static]
```

**6.107.2.41 usbdInit()**

```
static BUInt8 usbdInit (
    USBD_HandleTypeDef * pdev,
    BUInt8 cfgidx ) [static]
```

#### 6.107.2.42 usbdRx()

```
static BUInt8 usbdRx (  
    USBD_HandleTypeDef * pdev,  
    BUInt8 epnum ) [static]
```

#### 6.107.2.43 usbdRx0()

```
static BUInt8 usbdRx0 (  
    USBD_HandleTypeDef * pdev ) [static]
```

#### 6.107.2.44 usbdSetup()

```
static BUInt8 usbdSetup (  
    USBD_HandleTypeDef * pdev,  
    USBD_SetupReqTypedef * req ) [static]
```

#### 6.107.2.45 usbdTxSent()

```
static BUInt8 usbdTxSent (  
    USBD_HandleTypeDef * pdev,  
    BUInt8 epnum ) [static]
```

#### 6.107.2.46 usbWrite()

```
static int usbWrite (  
    const char * data,  
    int nBytes ) [static]
```

### 6.107.3 Variable Documentation

6.107.3.1 `__ALIGN_END`

```
__ALIGN_BEGIN BUInt8 usbdDeviceQualifierDesc [USB_LEN_DEV_QUALIFIER_DESC] __ALIGN_END
```

**Initial value:**

```
= {
    0x12,
    USB_DESC_TYPE_DEVICE,
    0x00,
    0x02,
    busbConfig.devClass,
    busbConfig.devSubClass,
    busbConfig.devProtocol,
    USB_MAX_EP0_SIZE,
    LOBYTE(USB_D_VID),
    HIBYTE(USB_D_VID),
    LOBYTE(USB_D_PID),
    HIBYTE(USB_D_PID),
    0x00,
    0x02,
    USB_D_IDX_MFC_STR,
    USB_D_IDX_PRODUCT_STR,
    USB_D_IDX_SERIAL_STR,
    USB_D_MAX_NUM_CONFIGURATION
}
```

6.107.3.2 `busbConfig`

```
BUsbConfig busbConfig [static]
```

**Initial value:**

```
= {
    0x0483,
    0x5740,
    0x00,
    0x00,
    0x00,
    0x00,
    "Beam",
    "Armsys",
    "0",
}
```

6.107.3.3 `ConfigDescriptorSize`

```
const int ConfigDescriptorSize = 67
```

6.107.3.4 `EndPointCmd`

```
const int EndPointCmd = 0x82
```

### 6.107.3.5 EndPointRx

```
const int EndPointRx = 0x01
```

### 6.107.3.6 EndPointTx

```
const int EndPointTx = 0x81
```

### 6.107.3.7 PacketSizeCmd

```
const int PacketSizeCmd = 8
```

### 6.107.3.8 PacketSizeFullSpeed

```
const int PacketSizeFullSpeed = 64
```

### 6.107.3.9 PacketSizeHighSpeed

```
const int PacketSizeHighSpeed = 512
```

### 6.107.3.10 usbdClass

```
USBD_ClassTypeDef usbdClass
```

#### Initial value:

```
= {  
    usbdInit,  
    usbdDeInit,  
    usbdSetup,  
    NULL,  
    usbdRx0,  
    usbdTxSent,  
    usbdRx,  
    NULL,  
    NULL,  
    NULL,  
    usbdGetHSCfgDesc,  
    usbdGetFSCfgDesc,  
    usbdGetOtherSpeedCfgDesc,  
    usbdGetDeviceQualifierDescriptor,  
}
```

### 6.107.3.11 usbSerial

```
BUsbSerial* usbSerial [static]
```

### 6.107.3.12 VCP\_Desc

```
USBDescriptorTypeDef VCP_Desc
```

#### Initial value:

```
= {  
    USBDescriptor_VCP_DeviceDescriptor,  
    USBDescriptor_VCP_LangIDStrDescriptor,  
    USBDescriptor_VCP_ManufacturerStrDescriptor,  
    USBDescriptor_VCP_ProductStrDescriptor,  
    USBDescriptor_VCP_SerialStrDescriptor,  
    USBDescriptor_VCP_ConfigStrDescriptor,  
    USBDescriptor_VCP_InterfaceStrDescriptor,  
}
```

## 6.108 BUsbSerial.h File Reference

```
#include <BComms.h>  
#include <BFifo.h>  
#include <BSemaphore.h>  
#include <stdarg.h>  
#include <usbd_def.h>
```

### Classes

- struct [BUsbConfig](#)
- class [BUsbSerial](#)

### Typedefs

- typedef struct [BUsbConfig](#) [BUsbConfig](#)

## 6.108.1 Typedef Documentation

### 6.108.1.1 BUsbConfig

```
typedef struct BUsbConfig BUsbConfig
```

## 6.109 BUsbSerialLock.cpp File Reference

```
#include <BUsbSerialLock.h>
```

## 6.110 BUsbSerialLock.h File Reference

```
#include <BMutex.h>  
#include <BUsbSerial.h>
```

### Classes

- class [BUsbSerialLock](#)

## 6.111 BWatchdog.cpp File Reference

```
#include <BWatchdog.h>  
#include <ArmSystem.h>
```

## 6.112 BWatchdog.h File Reference

### Classes

- class [BWatchdog](#)

## 6.113 overview.dox File Reference

# Index

- `__ALIGN_END`
    - `BUsbSerial.cpp`, 402
  - `__attribute__`
    - `BTask.cpp`, 375
    - `BoapMc.h`, 327
    - `BoapMc1.h`, 330, 331
  - `__calloc_r`
    - `BHeapTest.cpp`, 306
  - `__ebss`
    - `BSys.cpp`, 369
  - `__edata`
    - `BSys.cpp`, 369
  - `__eheap`
    - `BSys.cpp`, 370
  - `__estack`
    - `BSys.cpp`, 370
  - `__etext`
    - `BSys.cpp`, 370
  - `__free_r`
    - `BHeapTest.cpp`, 306
  - `__malloc_r`
    - `BHeapTest.cpp`, 306
  - `__realloc_r`
    - `BHeapTest.cpp`, 307
  - `__sbss`
    - `BSys.cpp`, 370
  - `__sdata`
    - `BSys.cpp`, 370
  - `__stext`
    - `BSys.cpp`, 370
  - `__version`
    - `BSys.cpp`, 368
  - `~BArray`
    - `BArray`, 17
  - `~BBuffer`
    - `BBuffer`, 22
  - `~BBufferStore`
    - `BBufferStore`, 26
  - `~BComms`
    - `BComms`, 39
  - `~BCondInt`
    - `BCondInt`, 44
  - `~BCriticalLock`
    - `BCriticalLock`, 48
  - `~BFifo`
    - `BFifo`, 70
  - `~BHtml`
    - `BHtml`, 89
  - `~BList`
    - `BList`, 111
  - `~BMutex`
    - `BMutex`, 121
  - `~BMutexLock`
    - `BMutexLock`, 122
  - `~BNetwork`
    - `BNetwork`, 128
  - `~BObj`
    - `BObj`, 166
  - `~BQueue`
    - `BQueue`, 171
  - `~BRefString`
    - `BRefString`, 176
  - `~BSemaphore`
    - `BSemaphore`, 188
  - `~BSemaphoreBool`
    - `BSemaphoreBool`, 190
  - `~BSemaphoreCount`
    - `BSemaphoreCount`, 193
  - `~BString`
    - `BString`, 202
  - `~BTask`
    - `BTask`, 218
  - `~BoapMc1Comms`
    - `BoapMc1Comms`, 142
  - `~BoapMcClientObject`
    - `BoapMcClientObject`, 152
  - `~BoapMcComms`
    - `BoapMcComms`, 156
  - `~BoapMcServiceObject`
    - `BoapMcServiceObject`, 164
- `add`
- `BSemaphoreCount`, 193
  - `BString`, 203
- `addFromIsr`
- `BSemaphoreCount`, 193
- `addMicroSeconds`
- `BTimeUs`, 229
- `addSeconds`
- `BTime`, 224
  - `BTimeUs`, 229
- `addressFrom`
- `BoapMc.h`, 327
  - `BoapMc1.h`, 331
  - `BoapMc1PacketHead`, 150
  - `BoapMcPacketHead`, 162
- `addressTo`
- `BoapMc.h`, 328
  - `BoapMc1.h`, 331

- BoapMc1PacketHead, [150](#)
- BoapMcPacketHead, [163](#)
- ahbClock
  - BClocks, [36](#)
- allocate
  - BArray, [17](#)
  - BHeap, [87](#)
- alwaysCopy
  - BRefString, [178](#)
- apb1Clock
  - BClocks, [36](#)
- apb2Clock
  - BClocks, [37](#)
- append
  - BArray, [17](#)
  - BDict, [53](#)
  - BHtml, [89](#)
  - BList, [112](#)
  - BString, [203](#)
- appendFormField
  - BHtml, [89](#)
- appendFormItem
  - BHtml, [89](#)
- appendText
  - BHtml, [90](#)
- arg
  - BEvent, [67](#)
- BAdc, [13](#)
  - BAdc, [13](#)
  - enableBatteryInput, [14](#)
  - enableTempAndRefInput, [14](#)
  - getValue, [14](#)
  - init, [14](#)
  - initChannel, [14](#)
  - oadc, [15](#)
  - odev, [15](#)
- BAdc.cpp, [261](#)
  - DualMode\_AlterTrig, [262](#)
  - DualMode\_InjecSimult, [262](#)
  - DualMode\_Interl, [262](#)
  - DualMode\_RegSimult, [262](#)
  - DualMode\_RegSimult\_AlterTrig, [262](#)
  - DualMode\_RegSimult\_InjecSimult, [262](#)
  - Mode\_Independent, [263](#)
  - Prescaler\_Div2, [263](#)
  - Prescaler\_Div4, [263](#)
  - Prescaler\_Div6, [263](#)
  - Prescaler\_Div8, [263](#)
  - SampleTime\_112Cycles, [263](#)
  - SampleTime\_144Cycles, [263](#)
  - SampleTime\_15Cycles, [263](#)
  - SampleTime\_28Cycles, [264](#)
  - SampleTime\_3Cycles, [264](#)
  - SampleTime\_480Cycles, [264](#)
  - SampleTime\_56Cycles, [264](#)
  - SampleTime\_84Cycles, [264](#)
  - TripleMode\_AlterTrig, [264](#)
  - TripleMode\_InjecSimult, [264](#)
  - TripleMode\_Interl, [264](#)
  - TripleMode\_RegSimult, [265](#)
  - TripleMode\_RegSimult\_AlterTrig, [265](#)
  - TripleMode\_RegSimult\_InjecSimult, [265](#)
  - TwoSamplingDelay\_10Cycles, [265](#)
  - TwoSamplingDelay\_11Cycles, [265](#)
  - TwoSamplingDelay\_12Cycles, [265](#)
  - TwoSamplingDelay\_13Cycles, [265](#)
  - TwoSamplingDelay\_14Cycles, [265](#)
  - TwoSamplingDelay\_15Cycles, [266](#)
  - TwoSamplingDelay\_16Cycles, [266](#)
  - TwoSamplingDelay\_17Cycles, [266](#)
  - TwoSamplingDelay\_18Cycles, [266](#)
  - TwoSamplingDelay\_19Cycles, [266](#)
  - TwoSamplingDelay\_20Cycles, [266](#)
  - TwoSamplingDelay\_5Cycles, [266](#)
  - TwoSamplingDelay\_6Cycles, [266](#)
  - TwoSamplingDelay\_7Cycles, [267](#)
  - TwoSamplingDelay\_8Cycles, [267](#)
  - TwoSamplingDelay\_9Cycles, [267](#)
- BAdc.h, [267](#)
- BArray
  - ~BArray, [17](#)
  - allocate, [17](#)
  - append, [17](#)
  - BArray, [16, 17](#)
  - chunkSize, [20](#)
  - clear, [17](#)
  - data, [18](#)
  - dataSize, [18](#)
  - del, [18](#)
  - get, [18](#)
  - insert, [19](#)
  - number, [19](#)
  - odata, [21](#)
  - odataSize, [21](#)
  - operator T\*, [19](#)
  - operator=, [19](#)
  - operator[], [19, 20](#)
  - osize, [21](#)
  - relocate, [20](#)
  - resize, [20](#)
  - size, [20](#)
- BArray< T >, [15](#)
- BArray.h, [267](#)
  - BArrayFloat32, [268](#)
  - BArrayFloat64, [268](#)
  - BArrayInt32, [268](#)
- BArray\_imp.h, [268](#)
- BArrayFloat32
  - BArray.h, [268](#)
- BArrayFloat64
  - BArray.h, [268](#)
- BArrayInt32
  - BArray.h, [268](#)
- BBigEndian
  - BTypes.h, [381](#)
- BBuffer, [21](#)

- ~BBuffer, 22
- BBuffer, 22
- data, 22
- len, 22
- odata, 23
- odataSize, 23
- operator[], 23
- osize, 23
- setSize, 23
- size, 23
- BBuffer.cpp, 268
  - roundSize, 269
- BBuffer.h, 269
- BBufferFixed
  - BBufferFixed, 24
  - odataFixed, 24
- BBufferFixed< Size >, 24
- BBufferStore, 25
  - ~BBufferStore, 26
  - BBufferStore, 26
  - getHexString, 26
  - getPos, 26
  - opos, 31
  - oswapBytes, 31
  - pop, 26–28
  - push, 28–30
  - setHexString, 30
  - setPos, 30
- BCan, 31
  - BCan, 32
  - debugPrint, 32
  - init, 32
  - interrupt, 32
  - ocan, 34
  - odev, 34
  - orxFifoCmd, 34
  - orxFifoReply, 34
  - packetMode, 32
  - read, 33
  - readAvailable, 33
  - recvCanPacket, 33
  - sendCanPacket, 33
  - write, 33
  - writeAvailable, 34
- BCan.cpp, 269
  - CAN1\_RX0\_IRQHandler, 270
  - CAN2\_RX0\_IRQHandler, 270
  - devs, 270
  - USE\_INTERRUPTS, 270
- BCan.h, 270
- BCanPacketFifo, 35
  - BCanPacketFifo, 35
  - inPacket, 35
  - num, 36
  - numFifo, 36
- BChar
  - BTypes.h, 381
- BClocks, 36
  - ahbClock, 36
  - apb1Clock, 36
  - apb2Clock, 37
  - sysClock, 37
- BComms, 37
  - ~BComms, 39
  - BComms, 39
  - connect, 39
  - disconnect, 39
  - eventQueue, 39
  - Flush, 38
  - flush, 39
  - init, 40
  - isConnected, 40
  - oconnected, 42
  - oevent, 42
  - oeventNum, 42
  - oeventQueue, 42
  - opacketMode, 42
  - otimeout, 42
  - packetMode, 40
  - read, 40
  - readAvailable, 40
  - setPacketMode, 40
  - setTimeout, 41
  - wait, 41
  - write, 41
  - writeAvailable, 41
  - writeChunks, 41
- BComms.cpp, 271
- BComms.h, 271
- BComplex
  - BComplex.h, 271
- BComplex.h, 271
  - BComplex, 271
  - BComplex32, 271
  - BComplex64, 271
- BComplex32
  - BComplex.h, 271
- BComplex64
  - BComplex.h, 271
- BCondInt, 43
  - ~BCondInt, 44
  - BCondInt, 44
  - decrement, 44
  - increment, 44
  - omutex, 46
  - operator++, 44
  - operator+=", 44
  - operator--, 45
  - operator-=, 45
  - osema, 46
  - ovalue, 46
  - setValue, 45
  - value, 45
  - waitLessThan, 45
  - waitLessThanOrEqual, 45
  - waitMoreThanOrEqual, 46

- BCondInt.cpp, 272
- BCondInt.h, 272
- BConfig.h, 272
- BConvert.cpp, 272
  - base64\_decode\_table, 273
  - base64Decode, 273
- BConvert.h, 273
  - base64Decode, 273
- BCrc16.cpp, 274
  - bcrc16, 274
  - table\_crc\_hi, 274
  - table\_crc\_lo, 275
- BCrc16.h, 276
  - bcrc16, 276
- BCrc32.cpp, 276
  - bcrc32, 276
  - crc32\_tab, 277
- BCrc32.h, 277
  - bcrc32, 277
- BCritical, 47
  - lock, 47
  - ostate, 47
  - unlock, 47
- BCritical.h, 277
- BCritical.inc, 278
- BCriticalLock, 48
  - ~BCriticalLock, 48
  - BCriticalLock, 48
  - lock, 49
  - olock, 49
- BDEBUGL1
  - BFirmware.cpp, 295
  - BHttpServer.cpp, 312
  - BNtpClient.cpp, 324
  - BSdio.cpp, 353
  - BSnmpServer.cpp, 361
  - BoapMc1.cpp, 329
- BDEBUGL2
  - BHttpServer.cpp, 312
  - BSdio.cpp, 353
  - BoapMc1.cpp, 329
- BDEBUGL3
  - BHttpServer.cpp, 312
- BDEBUGL4
  - BHttpServer.cpp, 312
- BDataChunk, 49
  - BDataChunk, 49
  - data, 50
  - size, 50
- BDebug.cpp, 278
  - bhd32, 278
  - bhd8, 278
  - bhd8a, 278
  - bhda32, 278
  - bhda8, 279
- BDebug.h, 279
  - bhd32, 280
  - bhd8, 280
  - bhd8a, 281
  - bhda8, 281
  - bhds32, 281
  - dl1printf, 279
  - dl2printf, 279
  - dl3printf, 280
  - dl4printf, 280
  - dprintf, 280
- BDevices, 50
  - BDevices, 50
  - enable, 50
  - getClocks, 51
  - init, 51
- BDevices.cpp, 281
  - dlprintf, 282
  - HSI\_CLOCK, 282
  - LDEBUG, 282
- BDevices.h, 282
  - Device, 283
- BDict
  - append, 53
  - BDict, 52, 53
  - clear, 53
  - del, 53, 54
  - find, 54
  - hasKey, 55
  - hashAdd, 54
  - hashDelete, 54
  - hashFind, 54
  - hashPrint, 55
  - insert, 55
  - iterator, 52
  - key, 55
  - ohashLists, 56
  - ohashSize, 56
  - operator+, 55
  - operator=, 55
  - operator[], 56
- BDict< Type >, 51
- BDict.cpp, 285
  - bdictStringToString, 285
  - fromBString, 285
  - toBString, 286
- BDict.h, 286
  - BDictString, 286
  - bdictStringToString, 287
  - fromBString, 287
  - toBString, 287
- BDictItem
  - BDictItem, 57
  - key, 57
  - value, 57
- BDictItem< Type >, 57
- BDictString
  - BDict.h, 286
- BDma, 58
  - BDma, 61
  - Burst, 59

- config, [61](#)
- configFlow, [61](#)
- Dir, [59](#)
- DmaType, [59](#)
- doDma, [61](#)
- Fifo, [60](#)
- init, [61](#)
- interrupt, [62](#)
- numLeft, [62](#)
- ochannel, [63](#)
- ocompleteCallback, [63](#)
- ocontroller, [64](#)
- odma, [64](#)
- odmaStream, [64](#)
- odmaType, [64](#)
- oerr, [64](#)
- ostatus, [64](#)
- ostream, [64](#)
- otranSize, [65](#)
- owait, [65](#)
- printStatus, [62](#)
- setCompleteCallback, [62](#)
- setNextBuffer, [62](#)
- Size, [60](#)
- Status, [60](#)
- statusClear, [62](#)
- statusGet, [63](#)
- stop, [63](#)
- wait, [63](#)
- waitStatus, [63](#)
- BDma.cpp, [287](#)
- DMA1\_Stream0\_IRQHandler, [288](#)
- DMA1\_Stream1\_IRQHandler, [289](#)
- DMA1\_Stream2\_IRQHandler, [289](#)
- DMA1\_Stream3\_IRQHandler, [289](#)
- DMA1\_Stream4\_IRQHandler, [289](#)
- DMA1\_Stream5\_IRQHandler, [289](#)
- DMA1\_Stream6\_IRQHandler, [289](#)
- DMA1\_Stream7\_IRQHandler, [289](#)
- DMA2\_Stream0\_IRQHandler, [289](#)
- DMA2\_Stream1\_IRQHandler, [290](#)
- DMA2\_Stream2\_IRQHandler, [290](#)
- DMA2\_Stream3\_IRQHandler, [290](#)
- DMA2\_Stream4\_IRQHandler, [290](#)
- DMA2\_Stream5\_IRQHandler, [290](#)
- DMA2\_Stream6\_IRQHandler, [290](#)
- DMA2\_Stream7\_IRQHandler, [290](#)
- dlprintStatus, [288](#)
- dlprintf, [288](#)
- dmaStreams, [291](#)
- LDEBUG, [288](#)
- BDma.h, [291](#)
- BDmaCompleteCallback, [291](#)
- BDmaCompleteCallback
  - BDma.h, [291](#)
- BDouble
  - BTypes.h, [382](#)
- BEndian.cpp, [291](#)
- bswap\_copy, [292](#)
- BEndian.h, [292](#)
- bswap\_copy, [292](#)
- bswap\_p16, [292](#)
- bswap\_p32, [292](#)
- bswap\_p64, [293](#)
- bswap\_p8, [293](#)
- BError, [65](#)
- BError, [65](#)
- nullString, [66](#)
- num, [66](#)
- oerr, [66](#)
- operator int, [66](#)
- ostr, [67](#)
- set, [66](#)
- str, [66](#)
- BError.cpp, [293](#)
- BError.h, [293](#)
- BErrorNum, [293](#)
- BErrorNum
  - BError.h, [293](#)
- BEvent, [67](#)
- arg, [67](#)
- BEvent, [67](#)
- oarg, [68](#)
- otype, [68](#)
- type, [68](#)
- BEvent.cpp, [294](#)
- BEvent.h, [294](#)
- BEventQueue, [295](#)
- BEventQueue
  - BEvent.h, [295](#)
- BEventType
  - BTypes.h, [385](#)
- BEventWaitSet
  - BTypes.h, [386](#)
- BFifo
  - ~BFifo, [70](#)
  - BFifo, [70](#)
  - clear, [70](#)
  - odata, [75](#)
  - operator[], [70](#)
  - oreadPos, [75](#)
  - osize, [75](#)
  - owritePos, [75](#)
  - read, [70, 71](#)
  - readAvailable, [71](#)
  - readAvailableChunk, [71](#)
  - readData, [71](#)
  - readDone, [72](#)
  - readPeak, [72](#)
  - readPos, [72](#)
  - rebase, [72](#)
  - resize, [72](#)
  - size, [73](#)
  - write, [73](#)
  - writeAvailable, [73](#)
  - writeAvailableChunk, [73](#)

- writeBackup, 74
- writeData, 74
- writeDone, 74
- writePos, 74
- BFifo< Type >, 68
- BFifo.h, 295
- BFifo.inc, 295
- BFifoChar, 76
  - BFifoChar, 76
  - debugPrint, 76
  - find, 76
  - getBUInt32, 77
  - putBUInt32, 77
  - readInt, 77
- BFirmware.cpp, 295
  - BDEBUGL1, 295
  - bfirmwareBoot, 296
  - bfirmwareValid, 296
- BFirmware.h, 296
  - BFirmwareInfoEncrypt1, 297
  - BFirmwareInfoMagic, 297
  - bfirmwareBoot, 296
  - bfirmwareValid, 297
- BFirmwareInfo, 77
  - checksum, 78
  - length, 78
  - magic, 78
  - type, 78
  - ver0, 78
  - ver1, 78
  - ver2, 78
- BFirmwareInfoEncrypt1
  - BFirmware.h, 297
- BFirmwareInfoMagic
  - BFirmware.h, 297
- BFlash, 79
  - BFlash, 79
  - disableAccess, 80
  - enableAccess, 80
  - getWriteProtect, 80
  - init, 80
  - odummy, 82
  - page, 80
  - pageAddress, 80
  - programEnable, 81
  - programErase, 81
  - programWord, 81
  - waitUntillReady, 81
  - writeProtect, 82
- BFlash.cpp, 297
  - dlprintf, 298
  - FLASH\_KEY1, 298
  - FLASH\_KEY2, 298
  - FLASH\_OPT\_KEY1, 298
  - FLASH\_OPT\_KEY2, 298
  - LDEBUG, 298
  - RDP\_KEY, 298
- BFlash.h, 299
- BFloat
  - BTypes.h, 382
- BFloat32
  - BTypes.h, 382
- BFloat64
  - BTypes.h, 382
- BGpio, 82
  - BGpio, 85
  - getPin, 85
  - init, 85
  - initPin, 86
  - Mode, 83
  - OutType, 83
  - Pull, 83
  - setAlternateSource, 86
  - setPin, 86
  - Source, 84
  - Speed, 85
  - togglePin, 86
- BGpio.cpp, 299
  - bank, 299
- BGpio.h, 299
  - bitsSet, 304, 305
  - gpio, 305
  - Pin, 301
  - PinState, 304
- BHeap, 87
  - allocate, 87
  - BHeap, 87
  - init, 87
  - obase, 87
  - onext, 88
  - otop, 88
- BHeap.cpp, 305
- BHeap.h, 305
- BHeapTest.cpp, 306
  - \_calloc\_r, 306
  - \_free\_r, 306
  - \_malloc\_r, 306
  - \_realloc\_r, 307
  - heapTestAlloc, 307
  - heapTestMem, 307
  - heapTestMemPos, 307
  - heapTestMemSize, 307
  - heapTestPrint, 307
  - heapTestState, 308
  - heapTestStateNext, 308
- BHeapTest.h, 308
  - heapTestMem, 308
  - heapTestMemSize, 309
  - heapTestPrint, 308
- BHtml, 88
  - ~BHtml, 89
  - append, 89
  - appendFormField, 89
  - appendFormItem, 89
  - appendText, 90
  - BHtml, 88

- nodes, 90
- oargs, 90
- otag, 90
- render, 90
- BHtml.cpp, 309
  - htmlDecode, 309
  - htmlEncode, 309
  - urlDecode, 309
  - urlEncode, 309
  - urlEncodeComponent, 310
- BHtml.h, 310
  - htmlDecode, 310
  - htmlEncode, 311
  - HtmlWriteFunc, 310
  - urlDecode, 311
  - urlEncode, 311
  - urlEncodeComponent, 311
- BHttpServer, 91
  - BHttpServer, 93
  - cookieGet, 93
  - cookieSet, 93
  - devInit, 93
  - devProcess, 93
  - devRun, 93
  - devWrite, 93
  - function, 94
  - init, 94
  - oauthorisation, 97
  - oboundary, 97
  - obuffer, 97
  - obufferNext, 97
  - ocmd, 97
  - ocontentLen, 98
  - ocontentTypeRecv, 98
  - ocontentTypeSend, 98
  - ocookies, 98
  - ocookiesSend, 98
  - oget, 98
  - ohead\_cache, 98
  - ohead\_chunked, 98
  - ohead\_contentType, 99
  - ohead\_extra, 99
  - ohead\_size, 99
  - ohead\_status, 99
  - orxFifo, 99
  - osocket, 99
  - osocketListen, 99
  - otxFifo, 99
  - ourl, 100
  - pageSend, 94
  - pageSendError, 94
  - pageSendHead, 94
  - pageSendTail, 94
  - processHome, 94
  - processRequest, 95
  - processUrl, 95
  - readData, 95
  - ReadType, 92
  - run, 95
  - setHeader, 95
  - writeChunk, 95
  - writeFlush, 96
  - writeHead, 96
  - writeString, 96
  - writeTableInput, 96
  - writeTableRow, 96
  - writeTail, 97
- BHttpServer.cpp, 311
  - BDEBUGL1, 312
  - BDEBUGL2, 312
  - BDEBUGL3, 312
  - BDEBUGL4, 312
  - findCRNL, 312
  - Timeout, 313
- BHttpServer.h, 313
- BI2c, 100
  - BI2c, 101
  - getStatus, 101
  - hasStatus, 101
  - init, 101
  - interrupt, 102
  - odmaRx, 104
  - odmaTx, 104
  - oi2c, 104
  - oint, 104
  - olock, 104
  - ostatus, 104
  - otimeout, 105
  - ouseDma, 105
  - printStatus, 102
  - read, 102
  - readAddress16, 102
  - readAddress8, 102
  - readReg8, 102
  - waitFor, 103
  - write, 103
  - writeAddress16, 103
  - writeAddress8, 103
  - writeReg8, 103
- BI2c.cpp, 313
  - EVENT\_MASTER\_BYTE\_RECEIVED, 314
  - EVENT\_MASTER\_BYTE\_TRANSMITTED, 314
  - EVENT\_MASTER\_BYTE\_TRANSMITTING, 314
  - EVENT\_MASTER\_MODE\_ADDRESS10, 314
  - EVENT\_MASTER\_MODE\_SELECT, 314
  - EVENT\_MASTER\_RECEIVER\_MODE\_SELECTED, 314
  - EVENT\_MASTER\_TRANSMITTER\_MODE\_SELECTED, 314
  - EVENT\_SLAVE\_GENERALCALLADDRESS\_MATCHED, 315
  - EVENT\_SLAVE\_RECEIVER\_ADDRESS\_MATCHED, 315
  - EVENT\_SLAVE\_RECEIVER\_SECONDADDRESS\_MATCHED, 315

- EVENT\_SLAVE\_TRANSMITTER\_ADDRESS\_↔  
MATCHED, 315
- EVENT\_SLAVE\_TRANSMITTER\_SECONDDAD↔  
DRESS\_MATCHED, 315
- gdevs, 315
- I2CTYPE\_0, 315
- BI2c.h, 316
- BInt
  - BTypes.h, 382
- BInt16
  - BTypes.h, 382
- BInt32
  - BTypes.h, 382
- BInt64
  - BTypes.h, 382
- BInt8
  - BTypes.h, 383
- BInterrupt, 105
  - BInterrupt, 106
  - config, 106
  - enable, 106
  - extConfig, 106
  - init, 106
  - print, 107
  - setPriority, 107
- BInterrupt.cpp, 316
  - dprintf, 316
  - LDEBUG, 316
- BInterrupt.h, 317
- BIter, 107
  - BIter, 108
  - oi, 108
  - operator BNode \*, 108
  - operator==, 108
  - valid, 108
- BList
  - ~BList, 111
  - append, 112
  - BList, 111
  - begin, 112
  - clear, 112
  - del, 112
  - deleteFirst, 113
  - deleteLast, 113
  - end, 113
  - front, 113
  - get, 114
  - goTo, 114
  - has, 114
  - insert, 114
  - insertAfter, 115
  - isEnd, 115
  - next, 115
  - nodeCreate, 115
  - nodeGet, 116
  - number, 116
  - olength, 119
  - onodes, 120
  - operator+, 116
  - operator=, 116
  - operator[], 116, 117
  - pop, 117
  - position, 117
  - prev, 117
  - push, 118
  - queueAdd, 118
  - queueGet, 118
  - rear, 118
  - size, 118
  - sort, 119
  - SortFunc, 111
  - start, 119
  - swap, 119
- BList< T >, 109
- BList< T >::Node, 259
- BList.h, 317
  - BListLoop, 317
- BList::Node
  - item, 259
  - Node, 259
- BList\_func.h, 318
- BListLoop
  - BList.h, 317
- BMutex, 120
  - ~BMutex, 121
  - BMutex, 121
  - lock, 121
  - omutex, 122
  - operator=, 121
  - tryLock, 121
  - unlock, 121
- BMutex.cpp, 318
- BMutex.h, 318
- BMutexLock, 122
  - ~BMutexLock, 122
  - BMutexLock, 122
  - lock, 123
  - olock, 123
  - unlock, 123
- BNameValue
  - BNameValue, 124
  - getName, 124
  - getValue, 124
  - oname, 124
  - ovalue, 125
- BNameValue< T >, 123
- BNameValue.h, 318
- BNameValueList
  - find, 125
  - findPos, 125
- BNameValueList< T >, 125
- BNetwork, 126
  - ~BNetwork, 128
  - BNetwork, 128
  - config, 128
  - etherInit, 128

- etherPacketRecv, 128
- etherPacketSend, 129
- getIpAddress, 129
- getMac, 129
- getStats, 129
- init, 129
- interrupt, 129
- linkChanged, 130
- odhcp, 131
- odhcpState, 132
- odhcpTask, 132
- oether, 132
- oeventQueue, 132
- ogateway, 132
- oipAddress, 132
- omacAddress, 133
- onetif, 133
- onetmask, 133
- ophyAutoNeg, 133
- ophyFullDuplex, 133
- ophyNum, 133
- ophySpeed, 134
- ophyType, 134
- oprocessTask, 134
- ostats, 134
- phyInit, 130
- phyPrint, 130
- phyProcess, 130
- phySet, 130
- phySetup, 130
- PhyType, 127
- phyUpdate, 131
- process, 131
- process\_dhcp, 131
- start, 131
- BNetwork.cpp, 319
  - bnetwork, 321
  - DhcpTries, 321
  - dprintf, 319
  - dmaRxDscrTab, 321
  - dmaTxDscrTab, 321
  - ETH\_IRQHandler, 320
  - ethPhySettings, 321
  - ether\_init, 320
  - ether\_link\_changed, 320
  - ether\_send, 320
  - EventEthernet, 321
  - EventLinkChanged, 322
  - HAL\_ETH\_RxCpltCallback, 321
  - LDEBUG, 320
  - PhyDP83848\_CR, 322
  - PhyDP83848\_LEDCR, 322
  - ProcessTaskPriority, 322
  - rxBuffer, 322
  - TX\_ZERO\_COPY, 320
  - txBuffer, 322
- BNetwork.h, 323
  - DhcpState, 323
  - IpPacket, 323
  - BNetworkDhcpTask, 135
    - BNetworkDhcpTask, 135
    - onetwork, 136
    - run, 135
  - BNetworkProcessTask, 136
    - BNetworkProcessTask, 136
    - onetwork, 137
    - run, 137
  - BNetworkStats, 137
    - BNetworkStats, 138
    - rxDroppedPackets, 138
    - rxPackets, 138
    - txDroppedPackets, 138
    - txPackets, 138
  - BNode, 139
    - BNode, 139
    - next, 139
    - prev, 139
  - BNtpClient, 140
    - BNtpClient, 140
    - init, 140
    - run, 140
    - setServerAddress, 140
    - setSystemTime, 140
  - BNtpClient.cpp, 324
    - BDEBUGL1, 324
    - ontpClient, 325
    - sntp\_set\_system\_time, 324
  - BNtpClient.h, 325
- BObj, 166
  - ~BObj, 166
  - BObj, 166
  - getMembers, 167
  - getType, 167
  - membersPrint, 167
- BObj.cpp, 333
- BObj.h, 333
- BObjMember, 167
  - dataOffset, 168
  - name, 168
  - size, 168
  - type, 168
  - typeComp, 168
  - typeName, 168
- BObjStringFormat.cpp, 333
  - toBDictStringFromJson, 334
  - toBString, 334–337
  - toBStringJson, 337–340
- BObjStringFormat.h, 340
  - base64\_decode, 341
  - base64\_encode, 341
  - toBDictStringFromJson, 342
  - toBString, 342–345
  - toBStringJson, 345–348
- BPwm, 168
  - BPwm, 169
  - init, 169

- initChannel, 169
- operiod, 170
- otim, 170
- set, 169
- BPwm.cpp, 348
- BPwm.h, 348
- BQueue
  - ~BQueue, 171
  - BQueue, 171
  - clear, 171
  - oqueue, 173
  - read, 171
  - readAvailable, 171
  - readAvailableFromlSr, 172
  - readFromlSr, 172
  - write, 172
  - writeAvailable, 172
  - writeFromlSr, 172
- BQueue< T >, 170
- BQueue.cpp, 348
- BQueue.h, 348
  - BQueueInt, 349
- BQueueInt
  - BQueue.h, 349
- BRandom, 173
  - BRandom, 173
  - init, 173
  - value, 173
- BRandom.cpp, 349
- BRandom.h, 349
- BRefData, 174
  - data, 174
  - len, 174
  - refCount, 174
- BRefString, 175
  - ~BRefString, 176
  - alwaysCopy, 178
  - BRefString, 175, 176
  - clear, 176
  - debugPrint, 176
  - len, 176
  - operator=, 177
  - oref, 178
  - ostr, 178
  - refDelete, 177
  - refMakeWriteable, 177
  - refNew, 177
  - refResize, 177
  - refWrite, 177
  - str, 177, 178
- BRtc, 178
  - BRtc, 179
  - enterInitialisation, 179
  - getBackupRegister, 179
  - getTime, 180
  - init, 180
  - oClock, 181
  - setBackupRegister, 180
  - setTime, 180
  - time, 180
  - waitForSync, 180
- BRtc.cpp, 349
  - dlprintf, 350
  - fromBcd, 350
  - LDEBUG, 350
  - toBcd, 350
  - USE\_SYNC, 350
- BRtc.h, 351
- BSdio, 181
  - BSdio, 183
  - backgroundEnable, 183
  - cardFullSpeed, 183
  - cardInit, 183
  - CardType, 182
  - erase, 183
  - getCardType, 183
  - getStatus, 184
  - init, 184
  - interrupt, 184
  - numBlocks, 184
  - oCardType, 185
  - odma, 185
  - oend, 186
  - oInt, 186
  - oClock, 186
  - oNumBlocks, 186
  - orca, 186
  - osdmmc, 186
  - ospeed, 187
  - oStatus, 187
  - readBlocks, 184
  - ResponseType, 182
  - sendCmd, 184
  - sync, 185
  - trim, 185
  - writeBlocks, 185
- BSdio.cpp, 351
  - BDEBUGL1, 353
  - BDEBUGL2, 353
  - dev, 360
  - SD\_CMD\_ALL\_SEND\_CID, 353
  - SD\_CMD\_APP\_CMD, 353
  - SD\_CMD\_APP\_SD\_SET\_BUSWIDTH, 353
  - SD\_CMD\_CLR\_WRITE\_PROT, 353
  - SD\_CMD\_ERASE\_GRP\_END, 353
  - SD\_CMD\_ERASE\_GRP\_START, 354
  - SD\_CMD\_ERASE, 353
  - SD\_CMD\_FAST\_IO, 354
  - SD\_CMD\_GEN\_CMD, 354
  - SD\_CMD\_GO\_IDLE\_STATE, 354
  - SD\_CMD\_GO\_INACTIVE\_STATE, 354
  - SD\_CMD\_GO\_IRQ\_STATE, 354
  - SD\_CMD\_HS\_BUSTEST\_READ, 354
  - SD\_CMD\_HS\_BUSTEST\_WRITE, 354
  - SD\_CMD\_HS\_SEND\_EXT\_CSD, 355
  - SD\_CMD\_HS\_SWITCH, 355

- SD\_CMD\_LOCK\_UNLOCK, 355
- SD\_CMD\_NO\_CMD, 355
- SD\_CMD\_PROG\_CID, 355
- SD\_CMD\_PROG\_CSD, 355
- SD\_CMD\_READ\_DAT\_UNTIL\_STOP, 355
- SD\_CMD\_READ\_MULT\_BLOCK, 355
- SD\_CMD\_READ\_SINGLE\_BLOCK, 356
- SD\_CMD\_SD\_APP\_OP\_COND, 356
- SD\_CMD\_SD\_APP\_SEND\_NUM\_WRITE\_BLOCKS, 356
- SD\_CMD\_SD\_APP\_SEND\_SCR, 356
- SD\_CMD\_SD\_APP\_SET\_CLR\_CARD\_DETECT, 356
- SD\_CMD\_SD\_APP\_STAUS, 356
- SD\_CMD\_SD\_ERASE\_GRP\_END, 356
- SD\_CMD\_SD\_ERASE\_GRP\_START, 356
- SD\_CMD\_SDIO\_RW\_DIRECT, 357
- SD\_CMD\_SDIO\_RW\_EXTENDED, 357
- SD\_CMD\_SDIO\_SEN\_OP\_COND, 357
- SD\_CMD\_SEL\_DESEL\_CARD, 357
- SD\_CMD\_SEND\_CID, 357
- SD\_CMD\_SEND\_CSD, 357
- SD\_CMD\_SEND\_OP\_COND, 357
- SD\_CMD\_SEND\_STATUS, 357
- SD\_CMD\_SEND\_WRITE\_PROT, 358
- SD\_CMD\_SET\_BLOCK\_COUNT, 358
- SD\_CMD\_SET\_BLOCKLEN, 358
- SD\_CMD\_SET\_DSR, 358
- SD\_CMD\_SET\_REL\_ADDR, 358
- SD\_CMD\_SET\_WRITE\_PROT, 358
- SD\_CMD\_STOP\_TRANSMISSION, 358
- SD\_CMD\_WRITE\_DAT\_UNTIL\_STOP, 358
- SD\_CMD\_WRITE\_MULT\_BLOCK, 359
- SD\_CMD\_WRITE\_SINGLE\_BLOCK, 359
- SDCardState, 359
- SDIO\_IRQHandler, 360
- SDMMC2\_IRQHandler, 360
- STATUS\_ERR\_BITS, 359
- STATUS\_READY, 359
- USE\_DMA, 359
- BSdio.h, 360
- BSemaphore, 187
  - ~BSemaphore, 188
  - BSemaphore, 188
  - operator=, 188
  - osemaphore, 189
  - set, 188
  - setFromIsr, 188
  - wait, 189
- BSemaphore.cpp, 361
- BSemaphore.h, 361
- BSemaphoreBool, 189
  - ~BSemaphoreBool, 190
  - BSemaphoreBool, 190
  - clear, 190
  - operator int, 190
  - operator=, 190
  - operator==, 191
- osema, 191
- ovalue, 191
- set, 191
- value, 191
- wait, 191
- BSemaphoreCount, 192
  - ~BSemaphoreCount, 193
  - add, 193
  - addFromIsr, 193
  - BSemaphoreCount, 192
  - operator=, 193
  - osema, 194
  - ovalue, 194
  - setValue, 193
  - take, 193
  - value, 194
  - wait, 194
- BSnmpServer, 194
  - BSnmpServer, 195
  - init, 195
  - run, 195
- BSnmpServer.cpp, 361
  - BDEBUGL1, 361
  - osnmpServer, 362
- BSnmpServer.h, 362
- BSpi, 195
  - BSpi, 196
  - init, 197
  - initDev, 197
  - Mode, 196
  - obus, 198
  - ocsActives, 198
  - odmaRx, 198
  - odmaTx, 199
  - omodes, 199
  - opinCs, 199
  - ospi, 199
  - ouseDma, 199
  - sendRecv, 197
  - setMode, 197
  - transact, 198
  - transactFull, 198
- BSpi.cpp, 362
  - log2, 362
- BSpi.h, 363
- BString, 199
  - ~BString, 202
  - add, 203
  - append, 203
  - BString, 202
  - clear, 203
  - compare, 203
  - convert, 204, 205
  - convertHex, 205, 206
  - copy, 206
  - debugPrint, 206
  - del, 206
  - find, 206

- findReverse, 207
- hash, 207
- insert, 207
- len, 207
- operator const char \*, 207
- operator!=, 207, 208
- operator<, 208
- operator<=, 208, 209
- operator>, 209
- operator>=, 209, 210
- operator+, 208
- operator+=", 208
- operator=, 209
- operator==, 209
- operator[], 210
- orefs, 213
- pad, 210
- printf, 210
- pullToken, 210
- removeSeparators, 210
- retFloat64, 211
- retInt, 211
- retStr, 211
- retUInt, 211
- reverse, 211
- split, 211
- str, 212
- subString, 212
- toLower, 212
- toUpper, 212
- truncate, 212
- BString.cpp, 363
  - blistToString, 363
  - bstringListinList, 363
  - bstringToList, 364
  - bstrncpy, 364
  - bstrtrim, 364
  - floatToString, 364
  - intToString, 364
- BString.h, 365
  - BStringList, 365
  - blistToString, 365
  - bstringListinList, 366
  - bstringToList, 366
  - bstrncpy, 366
  - bstrtrim, 366
  - floatToString, 366
  - from\_hex, 366
  - intToString, 367
  - to\_hex, 367
- BStringList
  - BString.h, 365
- BSys, 213
  - BSys, 214
  - devices, 216
  - init, 214
  - interrupts, 216
  - otimer, 216
  - reset, 214
  - rtosRunning, 216
  - run, 214
  - setTaskHeap, 214
  - start, 215
  - systemStatePrint, 215
  - taskHeap, 216
  - taskStatsPrint, 215
  - timeUs, 215
  - timeUsReset, 215
- BSys.cpp, 367
  - \_ebss, 369
  - \_edata, 369
  - \_eheap, 370
  - \_estack, 370
  - \_etext, 370
  - \_sbss, 370
  - \_sdata, 370
  - \_stext, 370
  - \_version, 368
  - binterruptYield, 370
  - delayFunc, 370
  - delayMs, 368
  - delayUs, 368
  - delayUsClock, 368
  - delayUsLoop, 368
  - delayUsTask, 369
  - timeUs, 369
  - vApplicationGetIdleTaskMemory, 369
  - xPortSysTickHandler, 369
- BSys.h, 371
  - BSysTimer, 374
  - binterruptDisable, 373
  - binterruptEnable, 373
  - binterruptEnd, 372
  - binterruptInside, 373
  - binterruptStart, 372
  - binterruptYield, 374
  - bsysTaskYield, 373
  - delayFunc, 374
  - delayMs, 373
  - delayUs, 373
  - delayUsClock, 373
  - delayUsLoop, 374
  - delayUsTask, 374
  - FuncDelay, 372
  - FuncRun, 372
  - FuncTask, 372
  - sys, 375
  - timeUs, 374
- BSysTimer
  - BSys.h, 374
- BTask, 217
  - ~BTask, 218
  - BTask, 217
  - delayMs, 218
  - init, 218
  - oname, 219

- opriority, 219
- orunning, 219
- ostackSize, 220
- otask, 220
- run, 218
- runTasks, 218
- setPriority, 219
- start, 219
- stop, 219
- BTask.cpp, 375
  - \_\_attribute\_\_, 375
  - makeFreeRtosPriority, 375
  - osThreadCreate, 375
  - taskDelayUs, 376
  - taskFunc, 376
  - vApplicationIdleHook, 376
- BTask.h, 376
- BTim, 220
  - BTim, 221
  - clear, 221
  - getCount, 221
  - init, 221
  - Mode, 220
  - oclnit, 222
  - OcMode, 221
  - ocSetCompare, 222
  - otim, 222
  - start, 222
- BTim.cpp, 376
  - TIM\_CCER\_CCE, 377
  - TIM\_CCMR1\_OCCE, 377
- BTim.h, 377
- BTime, 223
  - addSeconds, 224
  - BTime, 223
  - getDate, 224
  - getSeconds, 224
  - getString, 224
  - getTime, 224
  - isLeapYear, 225
  - isSet, 225
  - operator!=, 225
  - operator<, 225
  - operator<=, 226
  - operator>, 226
  - operator>=, 226
  - operator+, 225
  - operator+=", 225
  - operator==, 226
  - otime, 227
  - set, 226
  - setString, 227
  - setYearDay, 227
- BTime.cpp, 377
  - monDays, 378
  - yearDays, 378
  - yearIsLeap, 378
- BTime.h, 378
- BTimeUs, 228
  - addMicroSeconds, 229
  - addSeconds, 229
  - BTimeUs, 229
  - getDate, 229
  - getMicroSeconds, 229
  - getSeconds, 229
  - getString, 230
  - getTime, 230
  - isLeapYear, 230
  - isSet, 230
  - operator!=, 230
  - operator<, 231
  - operator<=, 231
  - operator>, 231
  - operator>=, 231
  - operator+, 230
  - operator+=", 231
  - operator==, 231
  - otime, 232
  - set, 231, 232
  - setString, 232
  - setYearDay, 232
- BTimeUs.cpp, 378
  - monDays, 379
  - yearDays, 379
  - yearIsLeap, 379
- BTimeUs.h, 379
- BTimeout
  - BTypes.h, 383
- BTimeoutForever
  - BTypes.h, 388
- BType
  - BTypes.h, 386
- BTypeComp
  - BTypes.h, 387
- BTypes.h, 380
  - BBigEndian, 381
  - BChar, 381
  - BDouble, 382
  - BEventType, 385
  - BEventWaitSet, 386
  - BFloat, 382
  - BFloat32, 382
  - BFloat64, 382
  - BInt, 382
  - BInt16, 382
  - BInt32, 382
  - BInt64, 382
  - BInt8, 383
  - BTimeout, 383
  - BTimeoutForever, 388
  - BType, 386
  - BTypeComp, 387
  - BUInt, 383
  - BUInt16, 383
  - BUInt32, 383
  - BUInt64, 383

- BUInt8, [383](#)
- BUsbType, [387](#)
- Bool, [383](#)
- bswap\_16, [387](#)
- bswap\_32, [387](#)
- bswap\_copy, [387](#)
- bswap\_p64, [388](#)
- Double, [384](#)
- Float, [384](#)
- Float32, [384](#)
- Float64, [384](#)
- Int, [384](#)
- Int16, [384](#)
- Int32, [384](#)
- Int64, [384](#)
- Int8, [385](#)
- M\_PI, [381](#)
- timeoutTicks, [388](#)
- UInt, [385](#)
- UInt16, [385](#)
- UInt32, [385](#)
- UInt64, [385](#)
- UInt8, [385](#)
- BUInt
  - BTypes.h, [383](#)
- BUInt16
  - BTypes.h, [383](#)
- BUInt32
  - BTypes.h, [383](#)
- BUInt64
  - BTypes.h, [383](#)
- BUInt8
  - BTypes.h, [383](#)
- BUart, [233](#)
  - BUart, [234](#)
  - devs, [237](#)
  - flushRx, [234](#)
  - hold, [234](#)
  - init, [234](#), [235](#)
  - interrupt, [235](#)
  - odev, [237](#)
  - olastRxTime, [237](#)
  - opinCts, [237](#)
  - opinRts, [237](#)
  - orx, [237](#)
  - orxDisable, [237](#)
  - orxFifo, [238](#)
  - otxEnablePin, [238](#)
  - otxFifo, [238](#)
  - ouart, [238](#)
  - owait, [238](#)
  - owaitReadNum, [238](#)
  - read, [235](#)
  - readAvailable, [235](#)
  - readLine, [235](#)
  - setMode485, [236](#)
  - setRtsCts, [236](#)
  - wait, [236](#)
  - write, [236](#)
  - writeAvailable, [236](#)
- BUart.cpp, [388](#)
  - UART4\_IRQHandler, [389](#)
  - UART5\_IRQHandler, [389](#)
  - USART1\_IRQHandler, [389](#)
  - USART2\_IRQHandler, [389](#)
  - USART3\_IRQHandler, [389](#)
  - USART6\_IRQHandler, [389](#)
- BUart.h, [389](#)
- BUartBasic, [239](#)
  - BUartBasic, [239](#)
  - init, [240](#)
  - orxDisable, [241](#)
  - otxEnablePin, [241](#)
  - ouart, [242](#)
  - printf, [240](#)
  - read, [240](#)
  - readAvailable, [240](#)
  - setMode485, [241](#)
  - write, [241](#)
  - writeAvailable, [241](#)
- BUartBasic.cpp, [390](#)
  - uart, [390](#)
  - uartWrite, [390](#)
- BUartBasic.h, [390](#)
- BUsbConfig, [242](#)
  - BUsbSerial.h, [405](#)
  - devClass, [242](#)
  - devProtocol, [242](#)
  - devSubClass, [242](#)
  - manufacturer, [243](#)
  - product, [243](#)
  - productId, [243](#)
  - serialNumber, [243](#)
  - vendorId, [243](#)
- BUsbSerial, [243](#)
  - BUsbSerial, [245](#)
  - close, [245](#)
  - init, [245](#)
  - oblocking, [250](#)
  - oconnected, [250](#)
  - ohighSpeed, [250](#)
  - ohpcd, [250](#)
  - oinitialised, [250](#)
  - opacketSize, [250](#)
  - orxBuffer, [251](#)
  - orxFifo, [251](#)
  - orxStalled, [251](#)
  - osync, [251](#)
  - otxFifo, [251](#)
  - otxSendLen, [252](#)
  - otxSending, [251](#)
  - ousbChunkSize, [252](#)
  - ousbDev, [252](#)
  - owait, [252](#)
  - owaitReadNum, [252](#)
  - printf, [246](#)

- printfv, 246
- read, 246
- readAvailable, 246
- setBlocking, 246
- setMaximumUsbChunk, 247
- setSync, 247
- start, 247
- usbDelInit, 247
- usbInit, 247
- usbInitHardware, 247
- usbInitUsbDev, 248
- usbInterrupt, 248
- usbRx, 248
- usbSetup, 248
- usbStarted, 248
- usbTx, 248
- usbTxComplete, 249
- wait, 249
- write, 249
- writeAvailable, 249
- BUsbSerial.cpp, 391
  - \_\_ALIGN\_END, 402
  - busbConfig, 403
  - ConfigDescriptorSize, 403
  - dl1printf, 393
  - dl2printf, 393
  - dl3printf, 393
  - EndPointCmd, 403
  - EndPointRx, 403
  - EndPointTx, 404
  - HAL\_PCD\_ConnectCallback, 395
  - HAL\_PCD\_DataInStageCallback, 395
  - HAL\_PCD\_DataOutStageCallback, 395
  - HAL\_PCD\_DisconnectCallback, 395
  - HAL\_PCD\_ISOINIncompleteCallback, 396
  - HAL\_PCD\_ISOOUTIncompleteCallback, 396
  - HAL\_PCD\_ResetCallback, 396
  - HAL\_PCD\_ResumeCallback, 396
  - HAL\_PCD\_SOFCallback, 396
  - HAL\_PCD\_SetupStageCallback, 396
  - HAL\_PCD\_SuspendCallback, 396
  - L1DEBUG, 393
  - L2DEBUG, 393
  - L3DEBUG, 393
  - OTG\_FS\_IRQHandler, 397
  - OTG\_HS\_IRQHandler, 397
  - PacketSizeCmd, 404
  - PacketSizeFullSpeed, 404
  - PacketSizeHighSpeed, 404
  - USB\_CONFIGURATION\_FS\_STRING, 393
  - USB\_CONFIGURATION\_HS\_STRING, 394
  - USB\_INTERFACE\_FS\_STRING, 394
  - USB\_INTERFACE\_HS\_STRING, 394
  - USB\_LANGID\_STRING, 394
  - USB\_LL\_ClearStallEP, 397
  - USB\_LL\_CloseEP, 397
  - USB\_LL\_DelInit, 397
  - USB\_LL\_Delay, 397
  - USB\_LL\_FlushEP, 397
  - USB\_LL\_GetRxDataSize, 398
  - USB\_LL\_Init, 398
  - USB\_LL\_IsStallEP, 398
  - USB\_LL\_OpenEP, 398
  - USB\_LL\_PrepareReceive, 398
  - USB\_LL\_SetUSBAddress, 398
  - USB\_LL\_StallEP, 399
  - USB\_LL\_Start, 399
  - USB\_LL\_Stop, 399
  - USB\_LL\_Transmit, 399
  - USB\_MANUFACTURER\_STRING, 394
  - USB\_PID, 394
  - USB\_PRODUCT\_FS\_STRING, 394
  - USB\_PRODUCT\_HS\_STRING, 394
  - USB\_SERIALNUMBER\_FS\_STRING, 395
  - USB\_SERIALNUMBER\_HS\_STRING, 395
  - USB\_VCP\_ConfigStrDescriptor, 399
  - USB\_VCP\_DeviceDescriptor, 399
  - USB\_VCP\_InterfaceStrDescriptor, 400
  - USB\_VCP\_LangIDStrDescriptor, 400
  - USB\_VCP\_ManufacturerStrDescriptor, 400
  - USB\_VCP\_ProductStrDescriptor, 400
  - USB\_VCP\_SerialStrDescriptor, 400
  - USB\_VID, 395
  - usbSerial, 404
  - usbWrite, 402
  - usbClass, 404
  - usbDelInit, 400
  - usbGetDeviceQualifierDescriptor, 401
  - usbGetFSCfgDesc, 401
  - usbGetHSCfgDesc, 401
  - usbGetOtherSpeedCfgDesc, 401
  - usbInit, 401
  - usbRx, 401
  - usbRx0, 402
  - usbSetup, 402
  - usbTxSent, 402
  - VCP\_Desc, 405
- BUsbSerial.h, 405
  - BUsbConfig, 405
- BUsbSerialLock, 253
  - BUsbSerialLock, 253
  - oLock, 255
  - printfv, 253
  - read, 254
  - readAvailable, 254
  - write, 254
  - writeAvailable, 254
- BUsbSerialLock.cpp, 406
- BUsbSerialLock.h, 406
- BUsbType
  - BTypes.h, 387
- BWatchdog, 255
  - BWatchdog, 255
  - reset, 255
  - start, 255
- BWatchdog.cpp, 406

- BWatchdog.h, 406
- backgroundEnable
  - BSdio, 183
- bank
  - BGpio.cpp, 299
- base64\_decode
  - BObjStringFormat.h, 341
- base64\_decode\_table
  - BConvert.cpp, 273
- base64\_encode
  - BObjStringFormat.h, 341
- base64Decode
  - BConvert.cpp, 273
  - BConvert.h, 273
- bcrc16
  - BCrc16.cpp, 274
  - BCrc16.h, 276
- bcrc32
  - BCrc32.cpp, 276
  - BCrc32.h, 277
- bdictStringToString
  - BDict.cpp, 285
  - BDict.h, 287
- begin
  - BList, 112
- bfirmwareBoot
  - BFirmware.cpp, 296
  - BFirmware.h, 296
- bfirmwareValid
  - BFirmware.cpp, 296
  - BFirmware.h, 297
- bhd32
  - BDebug.cpp, 278
  - BDebug.h, 280
- bhd8
  - BDebug.cpp, 278
  - BDebug.h, 280
- bhd8a
  - BDebug.cpp, 278
  - BDebug.h, 281
- bhda32
  - BDebug.cpp, 278
- bhda8
  - BDebug.cpp, 279
  - BDebug.h, 281
- bhds32
  - BDebug.h, 281
- bininterruptDisable
  - BSys.h, 373
- bininterruptEnable
  - BSys.h, 373
- bininterruptEnd
  - BSys.h, 372
- bininterruptInside
  - BSys.h, 373
- bininterruptStart
  - BSys.h, 372
- bininterruptYield
  - BSys.cpp, 370
  - BSys.h, 374
- bitsSet
  - BGpio.h, 304, 305
- blistToString
  - BString.cpp, 363
  - BString.h, 365
- bnetwork
  - BNetwork.cpp, 321
- BoapMc.cpp, 325
  - DEBUG\_LOCAL1, 326
  - DEBUG\_LOCAL, 325
  - dl1printf, 326
  - dlprintf, 326
- BoapMc.h, 326
  - \_\_attribute\_\_, 327
  - addressFrom, 327
  - addressTo, 328
  - BoapMcType, 327
  - checksum, 328
  - cmd, 328
  - error, 328
  - length, 328
- BoapMc1.cpp, 328
  - BDEBUGL1, 329
  - BDEBUGL2, 329
- BoapMc1.h, 329
  - \_\_attribute\_\_, 330, 331
  - addressFrom, 331
  - addressTo, 331
  - boapMc1CommsRoundupLen, 330
  - BoapMc1Magic, 331
  - BoapMc1Type, 330
  - checksum, 331
  - cmd, 331
  - data, 332
  - error, 332
  - head, 332
  - length, 332
  - magic, 332
  - number, 332
  - string, 332
- BoapMc1Comms, 141
  - ~BoapMc1Comms, 142
  - BoapMc1Comms, 142
  - getApiVersion, 143
  - oaddressFrom, 145
  - oaddressTo, 145
  - oapiVersion, 145
  - ocomms, 145
  - oerror, 145
  - ohalfDuplex, 146
  - oclockCall, 146
  - oclockTx, 146
  - opacketRpcCmd, 146
  - opacketRpcDoneSema, 146
  - opacketRpcSema, 146
  - opacketRx, 147

- opacketRxBase, 147
- opacketTx, 147
- opacketTxBase, 147
- oreqSize, 147
- othreaded, 147
- otimeout, 148
- packetRx, 143
- packetRxData, 143
- packetRxEnd, 143
- packetTx, 143
- processRequest, 143
- processRequests, 143
- processRx, 144
- setAddress, 144
- setComms, 144
- setCommsMode, 144
- setTimeout, 145
- boapMc1CommsRoundupLen
  - BoapMc1.h, 330
- BoapMc1Error, 148
  - number, 148
  - string, 148
- BoapMc1Magic
  - BoapMc1.h, 331
- BoapMc1Packet, 149
  - data, 149
  - head, 149
- BoapMc1PacketHead, 149
  - addressFrom, 150
  - addressTo, 150
  - checksum, 150
  - cmd, 150
  - error, 150
  - length, 151
  - magic, 151
- BoapMc1Type
  - BoapMc1.h, 330
- BoapMcClientObject, 151
  - ~BoapMcClientObject, 152
  - BoapMcClientObject, 152
  - getApiVersion, 152
  - oaddressFrom, 153
  - oaddressTo, 153
  - oapiVersion, 153
  - ocomms, 153
  - opacket, 154
  - performCall, 152
  - performRecv, 152
  - performSend, 153
  - setAddress, 153
- BoapMcComms, 154
  - ~BoapMcComms, 156
  - BoapMcComms, 155
  - getApiVersion, 156
  - oaddressFrom, 158
  - oaddressTo, 159
  - oapiVersion, 159
  - ocomms, 159
- oLockCall, 159
- oLockTx, 159
- opacket, 159
- opacketReqQueue, 159
- opacketReqRx, 160
- opacketReqTx, 160
- opacketRx, 160
- opacketRxSema, 160
- opacketTx, 160
- opacketTxQueue, 160
- opacketTxQueueWriteNum, 161
- opacketTxSema, 161
- oslave, 161
- othreaded, 161
- otimeout, 161
- packetRecv, 156
- packetSend, 156
- performCall, 156
- performSend, 156
- processPacket, 157
- processRequest, 157
- processRequests, 157
- processRx, 157
- setAddress, 157
- setComms, 158
- setCommsMode, 158
- setTimeout, 158
- BoapMcPacket, 162
  - data, 162
  - head, 162
- BoapMcPacketHead, 162
  - addressFrom, 162
  - addressTo, 163
  - checksum, 163
  - cmd, 163
  - error, 163
  - length, 163
- BoapMcServiceObject, 163
  - ~BoapMcServiceObject, 164
  - BoapMcServiceObject, 164
  - oapiVersion, 165
  - process, 164
  - processEvent, 164
  - sendEvent, 164
- BoapMcSignalObject, 165
  - BoapMcSignalObject, 165
  - ocomms, 166
  - performSend, 166
- BoapMcType
  - BoapMc.h, 327
- Bool
  - BTypes.h, 383
- bstringListInList
  - BString.cpp, 363
  - BString.h, 366
- bstringToList
  - BString.cpp, 364
  - BString.h, 366

- bstrncpy
  - BString.cpp, 364
  - BString.h, 366
- bstrtrim
  - BString.cpp, 364
  - BString.h, 366
- bswap\_16
  - BTypes.h, 387
- bswap\_32
  - BTypes.h, 387
- bswap\_copy
  - BEndian.cpp, 292
  - BEndian.h, 292
  - BTypes.h, 387
- bswap\_p16
  - BEndian.h, 292
- bswap\_p32
  - BEndian.h, 292
- bswap\_p64
  - BEndian.h, 293
  - BTypes.h, 388
- bswap\_p8
  - BEndian.h, 293
- bsysTaskYield
  - BSys.h, 373
- Burst
  - BDma, 59
- busbConfig
  - BUsbSerial.cpp, 403
- CAN1\_RX0\_IRQHandler
  - BCan.cpp, 270
- CAN2\_RX0\_IRQHandler
  - BCan.cpp, 270
- caller
  - HeapTestState, 256
- cardFullSpeed
  - BSdio, 183
- cardInit
  - BSdio, 183
- CardType
  - BSdio, 182
- checksum
  - BFirmwareInfo, 78
  - BoapMc.h, 328
  - BoapMc1.h, 331
  - BoapMc1PacketHead, 150
  - BoapMcPacketHead, 163
- chunkSize
  - BArray, 20
- clear
  - BArray, 17
  - BDict, 53
  - BFifo, 70
  - BList, 112
  - BQueue, 171
  - BRefString, 176
  - BSemaphoreBool, 190
  - BString, 203
  - BTim, 221
- close
  - BUsbSerial, 245
- cmd
  - BoapMc.h, 328
  - BoapMc1.h, 331
  - BoapMc1PacketHead, 150
  - BoapMcPacketHead, 163
- compare
  - BString, 203
- config
  - BDma, 61
  - BInterrupt, 106
  - BNetwork, 128
  - ConfigDescriptorSize
    - BUsbSerial.cpp, 403
- configFlow
  - BDma, 61
- connect
  - BComms, 39
- convert
  - BString, 204, 205
- convertHex
  - BString, 205, 206
- cookieGet
  - BHttpServer, 93
- cookieSet
  - BHttpServer, 93
- copy
  - BString, 206
- crc32\_tab
  - BCrc32.cpp, 277
- DEBUG\_LOCAL1
  - BoapMc.cpp, 326
- DEBUG\_LOCAL
  - BoapMc.cpp, 325
- DMA1\_Stream0\_IRQHandler
  - BDma.cpp, 288
- DMA1\_Stream1\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream2\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream3\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream4\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream5\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream6\_IRQHandler
  - BDma.cpp, 289
- DMA1\_Stream7\_IRQHandler
  - BDma.cpp, 289
- DMA2\_Stream0\_IRQHandler
  - BDma.cpp, 289
- DMA2\_Stream1\_IRQHandler
  - BDma.cpp, 290
- DMA2\_Stream2\_IRQHandler
  - BDma.cpp, 290

- DMA2\_Stream3\_IRQHandler
  - BDma.cpp, 290
- DMA2\_Stream4\_IRQHandler
  - BDma.cpp, 290
- DMA2\_Stream5\_IRQHandler
  - BDma.cpp, 290
- DMA2\_Stream6\_IRQHandler
  - BDma.cpp, 290
- DMA2\_Stream7\_IRQHandler
  - BDma.cpp, 290
- data
  - BArray, 18
  - BBuffer, 22
  - BDataChunk, 50
  - BRefData, 174
  - BoapMc1.h, 332
  - BoapMc1Packet, 149
  - BoapMcPacket, 162
- dataOffset
  - BObjMember, 168
- dataSize
  - BArray, 18
- debugPrint
  - BCan, 32
  - BFifoChar, 76
  - BRefString, 176
  - BString, 206
- decrement
  - BCondInt, 44
- del
  - BArray, 18
  - BDict, 53, 54
  - BList, 112
  - BString, 206
- delayFunc
  - BSys.cpp, 370
  - BSys.h, 374
- delayMs
  - BSys.cpp, 368
  - BSys.h, 373
  - BTask, 218
- delayUs
  - BSys.cpp, 368
  - BSys.h, 373
- delayUsClock
  - BSys.cpp, 368
  - BSys.h, 373
- delayUsLoop
  - BSys.cpp, 368
  - BSys.h, 374
- delayUsTask
  - BSys.cpp, 369
  - BSys.h, 374
- deleteFirst
  - BList, 113
- deleteLast
  - BList, 113
- dev
  - BSdio.cpp, 360
- devClass
  - BUsbConfig, 242
- devInit
  - BHttpServer, 93
- devProcess
  - BHttpServer, 93
- devProtocol
  - BUsbConfig, 242
- devRun
  - BHttpServer, 93
- devSubClass
  - BUsbConfig, 242
- devWrite
  - BHttpServer, 93
- Device
  - BDevices.h, 283
- devices
  - BSys, 216
- devs
  - BCan.cpp, 270
  - BUart, 237
- DhcpState
  - BNetwork.h, 323
- DhcpTries
  - BNetwork.cpp, 321
- Dir
  - BDma, 59
- disableAccess
  - BFlash, 80
- disconnect
  - BComms, 39
- dl1printf
  - BDebug.h, 279
  - BUsbSerial.cpp, 393
  - BoapMc.cpp, 326
- dl2printf
  - BDebug.h, 279
  - BUsbSerial.cpp, 393
- dl3printf
  - BDebug.h, 280
  - BUsbSerial.cpp, 393
- dl4printf
  - BDebug.h, 280
- dlprintStatus
  - BDma.cpp, 288
- dlprintf
  - BDevices.cpp, 282
  - BDma.cpp, 288
  - BFlash.cpp, 298
  - BInterrupt.cpp, 316
  - BNetwork.cpp, 319
  - BRtc.cpp, 350
  - BoapMc.cpp, 326
- dmaRxDscrTab
  - BNetwork.cpp, 321
- dmaStreams
  - BDma.cpp, 291

- dmaTxDescTab
  - BNetwork.cpp, 321
- DmaType
  - BDma, 59
- doDma
  - BDma, 61
- Double
  - BTypes.h, 384
- dprintf
  - BDebug.h, 280
- DualMode\_AlterTrig
  - BAdc.cpp, 262
- DualMode\_InjecSimult
  - BAdc.cpp, 262
- DualMode\_Interl
  - BAdc.cpp, 262
- DualMode\_RegSimult
  - BAdc.cpp, 262
- DualMode\_RegSimult\_AlterTrig
  - BAdc.cpp, 262
- DualMode\_RegSimult\_InjecSimult
  - BAdc.cpp, 262
  
- ETH\_IRQHandler
  - BNetwork.cpp, 320
- EVENT\_MASTER\_BYTE\_RECEIVED
  - BI2c.cpp, 314
- EVENT\_MASTER\_BYTE\_TRANSMITTED
  - BI2c.cpp, 314
- EVENT\_MASTER\_BYTE\_TRANSMITTING
  - BI2c.cpp, 314
- EVENT\_MASTER\_MODE\_ADDRESS10
  - BI2c.cpp, 314
- EVENT\_MASTER\_MODE\_SELECT
  - BI2c.cpp, 314
- EVENT\_MASTER\_RECEIVER\_MODE\_SELECTED
  - BI2c.cpp, 314
- EVENT\_MASTER\_TRANSMITTER\_MODE\_SELECTED↔
  - ED
  - BI2c.cpp, 314
- EVENT\_SLAVE\_GENERALCALLADDRESS\_MATC↔
  - HED
  - BI2c.cpp, 315
- EVENT\_SLAVE\_RECEIVER\_ADDRESS\_MATCHED
  - BI2c.cpp, 315
- EVENT\_SLAVE\_RECEIVER\_SECONDADDRESS\_↔
  - MATCHED
  - BI2c.cpp, 315
- EVENT\_SLAVE\_TRANSMITTER\_ADDRESS\_MATC↔
  - HED
  - BI2c.cpp, 315
- EVENT\_SLAVE\_TRANSMITTER\_SECONDADDRE↔
  - SS\_MATCHED
  - BI2c.cpp, 315
- enable
  - BDevices, 50
  - BInterrupt, 106
- enableAccess
  - BFlash, 80
- enableBatteryInput
  - BAdc, 14
- enableTempAndRefInput
  - BAdc, 14
- end
  - BList, 113
- EndPointCmd
  - BUsbSerial.cpp, 403
- EndPointRx
  - BUsbSerial.cpp, 403
- EndPointTx
  - BUsbSerial.cpp, 404
- enterInitialisation
  - BRtc, 179
- erase
  - BSdio, 183
- error
  - BoapMc.h, 328
  - BoapMc1.h, 332
  - BoapMc1PacketHead, 150
  - BoapMcPacketHead, 163
- ethPhySettings
  - BNetwork.cpp, 321
- ether\_init
  - BNetwork.cpp, 320
- ether\_link\_changed
  - BNetwork.cpp, 320
- ether\_send
  - BNetwork.cpp, 320
- etherInit
  - BNetwork, 128
- etherPacketRecv
  - BNetwork, 128
- etherPacketSend
  - BNetwork, 129
- EventEthernet
  - BNetwork.cpp, 321
- EventLinkChanged
  - BNetwork.cpp, 322
- eventQueue
  - BComms, 39
- extConfig
  - BInterrupt, 106
- FLASH\_KEY1
  - BFlash.cpp, 298
- FLASH\_KEY2
  - BFlash.cpp, 298
- FLASH\_OPT\_KEY1
  - BFlash.cpp, 298
- FLASH\_OPT\_KEY2
  - BFlash.cpp, 298
- Fifo
  - BDma, 60
- find
  - BDict, 54
  - BFifoChar, 76
  - BNameValueList, 125
  - BString, 206

- findCRNL
  - BHttpServer.cpp, 312
- findPos
  - BNameValueList, 125
- findReverse
  - BString, 207
- Float
  - BTypes.h, 384
- Float32
  - BTypes.h, 384
- Float64
  - BTypes.h, 384
- floatToString
  - BString.cpp, 364
  - BString.h, 366
- Flush
  - BComms, 38
- flush
  - BComms, 39
- flushRx
  - BUart, 234
- from\_hex
  - BString.h, 366
- fromBString
  - BDict.cpp, 285
  - BDict.h, 287
- fromBcd
  - BRtc.cpp, 350
- front
  - BList, 113
- FuncDelay
  - BSys.h, 372
- FuncRun
  - BSys.h, 372
- FuncTask
  - BSys.h, 372
- function
  - BHttpServer, 94
- gdevs
  - BI2c.cpp, 315
- get
  - BArray, 18
  - BList, 114
- getApiVersion
  - BoapMc1Comms, 143
  - BoapMcClientObject, 152
  - BoapMcComms, 156
- getBUInt32
  - BFifoChar, 77
- getBackupRegister
  - BRtc, 179
- getCardType
  - BSdio, 183
- getClocks
  - BDevices, 51
- getCount
  - BTim, 221
- getDate
  - BTime, 224
  - BTimeUs, 229
- getHexString
  - BBufferStore, 26
- getIpAddress
  - BNetwork, 129
- getMac
  - BNetwork, 129
- getMembers
  - BObj, 167
- getMicroSeconds
  - BTimeUs, 229
- getName
  - BNameValue, 124
- getPin
  - BGpio, 85
- getPos
  - BBufferStore, 26
- getSeconds
  - BTime, 224
  - BTimeUs, 229
- getStats
  - BNetwork, 129
- getStatus
  - BI2c, 101
  - BSdio, 184
- getString
  - BTime, 224
  - BTimeUs, 230
  - IpAddress, 258
- getTime
  - BRtc, 180
  - BTime, 224
  - BTimeUs, 230
- getType
  - BObj, 167
- getUInt
  - IpAddress, 258
- getValue
  - BAdc, 14
  - BNameValue, 124
- getWriteProtect
  - BFlash, 80
- goTo
  - BList, 114
- gpio
  - BGpio.h, 305
- HAL\_ETH\_RxCpltCallback
  - BNetwork.cpp, 321
- HAL\_PCD\_ConnectCallback
  - BUsbSerial.cpp, 395
- HAL\_PCD\_DataInStageCallback
  - BUsbSerial.cpp, 395
- HAL\_PCD\_DataOutStageCallback
  - BUsbSerial.cpp, 395
- HAL\_PCD\_DisconnectCallback
  - BUsbSerial.cpp, 395
- HAL\_PCD\_ISOINIncompleteCallback

- BUsbSerial.cpp, 396
- HAL\_PCD\_ISOOUTIncompleteCallback
  - BUsbSerial.cpp, 396
- HAL\_PCD\_ResetCallback
  - BUsbSerial.cpp, 396
- HAL\_PCD\_ResumeCallback
  - BUsbSerial.cpp, 396
- HAL\_PCD\_SOFCallback
  - BUsbSerial.cpp, 396
- HAL\_PCD\_SetupStageCallback
  - BUsbSerial.cpp, 396
- HAL\_PCD\_SuspendCallback
  - BUsbSerial.cpp, 396
- HSI\_CLOCK
  - BDevices.cpp, 282
- has
  - BList, 114
- hasKey
  - BDict, 55
- hasStatus
  - BI2c, 101
- hash
  - BString, 207
- hashAdd
  - BDict, 54
- hashDelete
  - BDict, 54
- hashFind
  - BDict, 54
- hashPrint
  - BDict, 55
- head
  - BoapMc1.h, 332
  - BoapMc1Packet, 149
  - BoapMcPacket, 162
- heapTestAlloc
  - BHeapTest.cpp, 307
- heapTestMem
  - BHeapTest.cpp, 307
  - BHeapTest.h, 308
- heapTestMemPos
  - BHeapTest.cpp, 307
- heapTestMemSize
  - BHeapTest.cpp, 307
  - BHeapTest.h, 309
- heapTestPrint
  - BHeapTest.cpp, 307
  - BHeapTest.h, 308
- HeapTestState, 256
  - caller, 256
  - pos, 256
  - size, 256
  - type, 256
- heapTestState
  - BHeapTest.cpp, 308
- heapTestStateNext
  - BHeapTest.cpp, 308
- hold
  - BUart, 234
- htmlDecode
  - BHtml.cpp, 309
  - BHtml.h, 310
- htmlEncode
  - BHtml.cpp, 309
  - BHtml.h, 311
- HtmlWriteFunc
  - BHtml.h, 310
- I2CTYPE\_0
  - BI2c.cpp, 315
- inPacket
  - BCanPacketFifo, 35
- increment
  - BCondInt, 44
- init
  - BAdc, 14
  - BCan, 32
  - BComms, 40
  - BDevices, 51
  - BDma, 61
  - BFlash, 80
  - BGpio, 85
  - BHeap, 87
  - BHttpServer, 94
  - BI2c, 101
  - BInterrupt, 106
  - BNetwork, 129
  - BNtpClient, 140
  - BPwm, 169
  - BRandom, 173
  - BRtc, 180
  - BSdio, 184
  - BSnmpServer, 195
  - BSpi, 197
  - BSys, 214
  - BTask, 218
  - BTim, 221
  - BUart, 234, 235
  - BUartBasic, 240
  - BUsbSerial, 245
- initChannel
  - BAdc, 14
  - BPwm, 169
- initDev
  - BSpi, 197
- initPin
  - BGpio, 86
- insert
  - BArray, 19
  - BDict, 55
  - BList, 114
  - BString, 207
- insertAfter
  - BList, 115
- Int
  - BTypes.h, 384
- Int16

- BTypes.h, [384](#)
- Int32
  - BTypes.h, [384](#)
- Int64
  - BTypes.h, [384](#)
- Int8
  - BTypes.h, [385](#)
- intToString
  - BString.cpp, [364](#)
  - BString.h, [367](#)
- interrupt
  - BCan, [32](#)
  - BDma, [62](#)
  - BI2c, [102](#)
  - BNetwork, [129](#)
  - BSdio, [184](#)
  - BUart, [235](#)
- interrupts
  - BSys, [216](#)
- IpAddress, [257](#)
  - getString, [258](#)
  - getUInt, [258](#)
  - IpAddress, [257](#)
  - oaddress, [258](#)
  - operator ip\_addr\_t, [258](#)
  - setString, [258](#)
- IpPacket
  - BNetwork.h, [323](#)
- isConnected
  - BComms, [40](#)
- isEnd
  - BList, [115](#)
- isLeapYear
  - BTime, [225](#)
  - BTimeUs, [230](#)
- isSet
  - BTime, [225](#)
  - BTimeUs, [230](#)
- item
  - BList::Node, [259](#)
- iterator
  - BDict, [52](#)
- key
  - BDict, [55](#)
  - BDictItem, [57](#)
- L1DEBUG
  - BUsbSerial.cpp, [393](#)
- L2DEBUG
  - BUsbSerial.cpp, [393](#)
- L3DEBUG
  - BUsbSerial.cpp, [393](#)
- LDEBUG
  - BDevices.cpp, [282](#)
  - BDma.cpp, [288](#)
  - BFlash.cpp, [298](#)
  - BInterrupt.cpp, [316](#)
  - BNetwork.cpp, [320](#)
  - BRtc.cpp, [350](#)
- len
  - BBuffer, [22](#)
  - BRefData, [174](#)
  - BRefString, [176](#)
  - BString, [207](#)
- length
  - BFirmwareInfo, [78](#)
  - BoapMc.h, [328](#)
  - BoapMc1.h, [332](#)
  - BoapMc1PacketHead, [151](#)
  - BoapMcPacketHead, [163](#)
- linkChanged
  - BNetwork, [130](#)
- lock
  - BCritical, [47](#)
  - BCriticalLock, [49](#)
  - BMutex, [121](#)
  - BMutexLock, [123](#)
- log2
  - BSpi.cpp, [362](#)
- M\_PI
  - BTypes.h, [381](#)
- magic
  - BFirmwareInfo, [78](#)
  - BoapMc1.h, [332](#)
  - BoapMc1PacketHead, [151](#)
- makeFreeRtosPriority
  - BTask.cpp, [375](#)
- manufacturer
  - BUsbConfig, [243](#)
- membersPrint
  - BObj, [167](#)
- Mode
  - BGpio, [83](#)
  - BSpi, [196](#)
  - BTim, [220](#)
- Mode\_Independent
  - BAdc.cpp, [263](#)
- monDays
  - BTime.cpp, [378](#)
  - BTimeUs.cpp, [379](#)
- name
  - BObjMember, [168](#)
- next
  - BList, [115](#)
  - BNode, [139](#)
- Node
  - BList::Node, [259](#)
- nodeCreate
  - BList, [115](#)
- nodeGet
  - BList, [116](#)
- nodes
  - BHtml, [90](#)
- nullString
  - BError, [66](#)

- num
  - BCanPacketFifo, 36
  - BError, 66
- numBlocks
  - BSdio, 184
- numFifo
  - BCanPacketFifo, 36
- numLeft
  - BDma, 62
- number
  - BArray, 19
  - BList, 116
  - BoapMc1.h, 332
  - BoapMc1Error, 148
- OTG\_FS\_IRQHandler
  - BUsbSerial.cpp, 397
- OTG\_HS\_IRQHandler
  - BUsbSerial.cpp, 397
- oadc
  - BAdc, 15
- oaddress
  - IpAddress, 258
- oaddressFrom
  - BoapMc1Comms, 145
  - BoapMcClientObject, 153
  - BoapMcComms, 158
- oaddressTo
  - BoapMc1Comms, 145
  - BoapMcClientObject, 153
  - BoapMcComms, 159
- oapiVersion
  - BoapMc1Comms, 145
  - BoapMcClientObject, 153
  - BoapMcComms, 159
  - BoapMcServiceObject, 165
- oarg
  - BEvent, 68
- oargs
  - BHtml, 90
- oauthorisation
  - BHttpServer, 97
- obase
  - BHeap, 87
- oblocking
  - BUsbSerial, 250
- oboundary
  - BHttpServer, 97
- obuffer
  - BHttpServer, 97
- obufferNext
  - BHttpServer, 97
- obus
  - BSpi, 198
- oclnit
  - BTim, 222
- OcMode
  - BTim, 221
- ocSetCompare
  - BTim, 222
- ocan
  - BCan, 34
- ocardType
  - BSdio, 185
- ochannel
  - BDma, 63
- ocmd
  - BHttpServer, 97
- ocomms
  - BoapMc1Comms, 145
  - BoapMcClientObject, 153
  - BoapMcComms, 159
  - BoapMcSignalObject, 166
- ocompleteCallback
  - BDma, 63
- oconnected
  - BComms, 42
  - BUsbSerial, 250
- ocontentLen
  - BHttpServer, 98
- ocontentTypeRecv
  - BHttpServer, 98
- ocontentTypeSend
  - BHttpServer, 98
- ocontroller
  - BDma, 64
- ocookies
  - BHttpServer, 98
- ocookiesSend
  - BHttpServer, 98
- ocsActives
  - BSpi, 198
- odata
  - BArray, 21
  - BBuffer, 23
  - BFifo, 75
- odataFixed
  - BBufferFixed, 24
- odataSize
  - BArray, 21
  - BBuffer, 23
- odev
  - BAdc, 15
  - BCan, 34
  - BUart, 237
- odhcp
  - BNetwork, 131
- odhcpState
  - BNetwork, 132
- odhcpTask
  - BNetwork, 132
- odma
  - BDma, 64
  - BSdio, 185
- odmaRx
  - BI2c, 104
  - BSpi, 198

- odmaStream
  - BDma, [64](#)
- odmaTx
  - BI2c, [104](#)
  - BSpi, [199](#)
- odmaType
  - BDma, [64](#)
- odummy
  - BFlash, [82](#)
- oend
  - BSdio, [186](#)
- oerr
  - BDma, [64](#)
  - BError, [66](#)
- oerror
  - BoapMc1Comms, [145](#)
- oether
  - BNetwork, [132](#)
- oevent
  - BComms, [42](#)
- oeventNum
  - BComms, [42](#)
- oeventQueue
  - BComms, [42](#)
  - BNetwork, [132](#)
- ogateway
  - BNetwork, [132](#)
- oget
  - BHttpServer, [98](#)
- ohalfDuplex
  - BoapMc1Comms, [146](#)
- ohashLists
  - BDict, [56](#)
- ohashSize
  - BDict, [56](#)
- ohead\_cache
  - BHttpServer, [98](#)
- ohead\_chunked
  - BHttpServer, [98](#)
- ohead\_contentType
  - BHttpServer, [99](#)
- ohead\_extra
  - BHttpServer, [99](#)
- ohead\_size
  - BHttpServer, [99](#)
- ohead\_status
  - BHttpServer, [99](#)
- ohighSpeed
  - BUsbSerial, [250](#)
- ohpcd
  - BUsbSerial, [250](#)
- oi
  - Blter, [108](#)
- oi2c
  - BI2c, [104](#)
- oinitialised
  - BUsbSerial, [250](#)
- oint
  - BI2c, [104](#)
  - BSdio, [186](#)
- oipAddress
  - BNetwork, [132](#)
- olastRxTime
  - BUart, [237](#)
- olength
  - BList, [119](#)
- olock
  - BCriticalLock, [49](#)
  - BI2c, [104](#)
  - BMutexLock, [123](#)
  - BRtc, [181](#)
  - BSdio, [186](#)
  - BUsbSerialLock, [255](#)
- olockCall
  - BoapMc1Comms, [146](#)
  - BoapMcComms, [159](#)
- olockTx
  - BoapMc1Comms, [146](#)
  - BoapMcComms, [159](#)
- omacAddress
  - BNetwork, [133](#)
- omodes
  - BSpi, [199](#)
- omutex
  - BCondInt, [46](#)
  - BMutex, [122](#)
- oname
  - BNameValue, [124](#)
  - BTask, [219](#)
- onetif
  - BNetwork, [133](#)
- onetmask
  - BNetwork, [133](#)
- onetwork
  - BNetworkDhcpTask, [136](#)
  - BNetworkProcessTask, [137](#)
- onext
  - BHeap, [88](#)
- onodes
  - BList, [120](#)
- onntpClient
  - BNtpClient.cpp, [325](#)
- onumBlocks
  - BSdio, [186](#)
- opacket
  - BoapMcClientObject, [154](#)
  - BoapMcComms, [159](#)
- opacketMode
  - BComms, [42](#)
- opacketReqQueue
  - BoapMcComms, [159](#)
- opacketReqRx
  - BoapMcComms, [160](#)
- opacketReqTx
  - BoapMcComms, [160](#)
- opacketRpcCmd

- BoapMc1Comms, 146
- opacketRpcDoneSema
  - BoapMc1Comms, 146
- opacketRpcSema
  - BoapMc1Comms, 146
- opacketRx
  - BoapMc1Comms, 147
  - BoapMcComms, 160
- opacketRxBase
  - BoapMc1Comms, 147
- opacketRxSema
  - BoapMcComms, 160
- opacketSize
  - BUsbSerial, 250
- opacketTx
  - BoapMc1Comms, 147
  - BoapMcComms, 160
- opacketTxBase
  - BoapMc1Comms, 147
- opacketTxQueue
  - BoapMcComms, 160
- opacketTxQueueWriteNum
  - BoapMcComms, 161
- opacketTxSema
  - BoapMcComms, 161
- operator BNode \*
  - BIter, 108
- operator const char \*
  - BString, 207
- operator int
  - BError, 66
  - BSemaphoreBool, 190
- operator ip\_addr\_t
  - IpAddress, 258
- operator T\*
  - BArray, 19
- operator!=
  - BString, 207, 208
  - BTime, 225
  - BTimeUs, 230
- operator<
  - BString, 208
  - BTime, 225
  - BTimeUs, 231
- operator<=
  - BString, 208, 209
  - BTime, 226
  - BTimeUs, 231
- operator>
  - BString, 209
  - BTime, 226
  - BTimeUs, 231
- operator>=
  - BString, 209, 210
  - BTime, 226
  - BTimeUs, 231
- operator+
  - BDict, 55
- BList, 116
- BString, 208
- BTime, 225
- BTimeUs, 230
- operator++
  - BCondInt, 44
- operator+=
  - BCondInt, 44
  - BString, 208
  - BTime, 225
  - BTimeUs, 231
- operator--
  - BCondInt, 45
- operator-=
  - BCondInt, 45
- operator=
  - BArray, 19
  - BDict, 55
  - BList, 116
  - BMutex, 121
  - BRefString, 177
  - BSemaphore, 188
  - BSemaphoreBool, 190
  - BSemaphoreCount, 193
  - BString, 209
- operator==
  - BIter, 108
  - BSemaphoreBool, 191
  - BString, 209
  - BTime, 226
  - BTimeUs, 231
- operator[]
  - BArray, 19, 20
  - BBuffer, 23
  - BDict, 56
  - BFifo, 70
  - BList, 116, 117
  - BString, 210
- operiod
  - BPwm, 170
- ophyAutoNeg
  - BNetwork, 133
- ophyFullDuplex
  - BNetwork, 133
- ophyNum
  - BNetwork, 133
- ophySpeed
  - BNetwork, 134
- ophyType
  - BNetwork, 134
- opinCs
  - BSpi, 199
- opinCts
  - BUart, 237
- opinRts
  - BUart, 237
- opos
  - BBufferStore, 31

- opriority
  - BTask, 219
- oprocessTask
  - BNetwork, 134
- oqueue
  - BQueue, 173
- orca
  - BSdio, 186
- oreadPos
  - BFifo, 75
- oref
  - BRefString, 178
- orefs
  - BString, 213
- oreqSize
  - BoapMc1Comms, 147
- orunning
  - BTask, 219
- orx
  - BUart, 237
- orxBuffer
  - BUsbSerial, 251
- orxDisable
  - BUart, 237
  - BUartBasic, 241
- orxFifo
  - BHttpServer, 99
  - BUart, 238
  - BUsbSerial, 251
- orxFifoCmd
  - BCan, 34
- orxFifoReply
  - BCan, 34
- orxStalled
  - BUsbSerial, 251
- osThreadCreate
  - BTask.cpp, 375
- osdmmc
  - BSdio, 186
- osema
  - BCondInt, 46
  - BSemaphoreBool, 191
  - BSemaphoreCount, 194
- osemaphore
  - BSemaphore, 189
- osize
  - BArray, 21
  - BBuffer, 23
  - BFifo, 75
- oslave
  - BoapMcComms, 161
- osnmpServer
  - BSnmpServer.cpp, 362
- osocket
  - BHttpServer, 99
- osocketListen
  - BHttpServer, 99
- ospeed
  - BSdio, 187
- ospi
  - BSpi, 199
- ostackSize
  - BTask, 220
- ostate
  - BCritical, 47
- ostats
  - BNetwork, 134
- ostatus
  - BDma, 64
  - BI2c, 104
  - BSdio, 187
- ostr
  - BError, 67
  - BRefString, 178
- ostream
  - BDma, 64
- oswapBytes
  - BBufferStore, 31
- osync
  - BUsbSerial, 251
- otag
  - BHtml, 90
- otask
  - BTask, 220
- othreaded
  - BoapMc1Comms, 147
  - BoapMcComms, 161
- otim
  - BPwm, 170
  - BTim, 222
- otime
  - BTime, 227
  - BTimeUs, 232
- otimeout
  - BComms, 42
  - BI2c, 105
  - BoapMc1Comms, 148
  - BoapMcComms, 161
- otimer
  - BSys, 216
- otop
  - BHeap, 88
- otranSize
  - BDma, 65
- otxEnablePin
  - BUart, 238
  - BUartBasic, 241
- otxFifo
  - BHttpServer, 99
  - BUart, 238
  - BUsbSerial, 251
- otxSendLen
  - BUsbSerial, 252
- otxSending
  - BUsbSerial, 251
- otype

- BEvent, 68
- ouart
  - BUart, 238
  - BUartBasic, 242
- ourl
  - BHttpServer, 100
- ousbChunkSize
  - BUsbSerial, 252
- ousbDev
  - BUsbSerial, 252
- ouseDma
  - BI2c, 105
  - BSpi, 199
- OutType
  - BGpio, 83
- ovalue
  - BCondInt, 46
  - BNameValue, 125
  - BSemaphoreBool, 191
  - BSemaphoreCount, 194
- overview.dox, 406
- owait
  - BDma, 65
  - BUart, 238
  - BUsbSerial, 252
- owaitReadNum
  - BUart, 238
  - BUsbSerial, 252
- owritePos
  - BFifo, 75
- packetMode
  - BCan, 32
  - BComms, 40
- packetRecv
  - BoapMcComms, 156
- packetRx
  - BoapMc1Comms, 143
- packetRxData
  - BoapMc1Comms, 143
- packetRxEnd
  - BoapMc1Comms, 143
- packetSend
  - BoapMcComms, 156
- PacketSizeCmd
  - BUsbSerial.cpp, 404
- PacketSizeFullSpeed
  - BUsbSerial.cpp, 404
- PacketSizeHighSpeed
  - BUsbSerial.cpp, 404
- packetTx
  - BoapMc1Comms, 143
- pad
  - BString, 210
- page
  - BFlash, 80
- pageAddress
  - BFlash, 80
- pageSend
  - BHttpServer, 94
- pageSendError
  - BHttpServer, 94
- pageSendHead
  - BHttpServer, 94
- pageSendTail
  - BHttpServer, 94
- performCall
  - BoapMcClientObject, 152
  - BoapMcComms, 156
- performRecv
  - BoapMcClientObject, 152
- performSend
  - BoapMcClientObject, 153
  - BoapMcComms, 156
  - BoapMcSignalObject, 166
- PhyDP83848\_CR
  - BNetwork.cpp, 322
- PhyDP83848\_LEDCR
  - BNetwork.cpp, 322
- phyInit
  - BNetwork, 130
- phyPrint
  - BNetwork, 130
- phyProcess
  - BNetwork, 130
- phySet
  - BNetwork, 130
- phySetup
  - BNetwork, 130
- PhyType
  - BNetwork, 127
- phyUpdate
  - BNetwork, 131
- Pin
  - BGpio.h, 301
- PinState
  - BGpio.h, 304
- pop
  - BBufferStore, 26–28
  - BList, 117
- pos
  - HeapTestState, 256
- position
  - BList, 117
- Prescaler\_Div2
  - BAdc.cpp, 263
- Prescaler\_Div4
  - BAdc.cpp, 263
- Prescaler\_Div6
  - BAdc.cpp, 263
- Prescaler\_Div8
  - BAdc.cpp, 263
- prev
  - BList, 117
  - BNode, 139
- print
  - BInterrupt, 107

- printStatus
  - BDma, [62](#)
  - BI2c, [102](#)
- printf
  - BString, [210](#)
  - BUartBasic, [240](#)
  - BUsbSerial, [246](#)
- printfv
  - BUsbSerial, [246](#)
  - BUsbSerialLock, [253](#)
- process
  - BNetwork, [131](#)
  - BoapMcServiceObject, [164](#)
- process\_dhcp
  - BNetwork, [131](#)
- processEvent
  - BoapMcServiceObject, [164](#)
- processHome
  - BHttpServer, [94](#)
- processPacket
  - BoapMcComms, [157](#)
- processRequest
  - BHttpServer, [95](#)
  - BoapMc1Comms, [143](#)
  - BoapMcComms, [157](#)
- processRequests
  - BoapMc1Comms, [143](#)
  - BoapMcComms, [157](#)
- processRx
  - BoapMc1Comms, [144](#)
  - BoapMcComms, [157](#)
- ProcessTaskPriority
  - BNetwork.cpp, [322](#)
- processUrl
  - BHttpServer, [95](#)
- product
  - BUsbConfig, [243](#)
- productId
  - BUsbConfig, [243](#)
- programEnable
  - BFlash, [81](#)
- programErase
  - BFlash, [81](#)
- programWord
  - BFlash, [81](#)
- Pull
  - BGpio, [83](#)
- pullToken
  - BString, [210](#)
- push
  - BBufferStore, [28–30](#)
  - BList, [118](#)
- putBUInt32
  - BFifoChar, [77](#)
- queueAdd
  - BList, [118](#)
- queueGet
  - BList, [118](#)
- RDP\_KEY
  - BFlash.cpp, [298](#)
- read
  - BCan, [33](#)
  - BComms, [40](#)
  - BFifo, [70, 71](#)
  - BI2c, [102](#)
  - BQueue, [171](#)
  - BUart, [235](#)
  - BUartBasic, [240](#)
  - BUsbSerial, [246](#)
  - BUsbSerialLock, [254](#)
- readAddress16
  - BI2c, [102](#)
- readAddress8
  - BI2c, [102](#)
- readAvailable
  - BCan, [33](#)
  - BComms, [40](#)
  - BFifo, [71](#)
  - BQueue, [171](#)
  - BUart, [235](#)
  - BUartBasic, [240](#)
  - BUsbSerial, [246](#)
  - BUsbSerialLock, [254](#)
- readAvailableChunk
  - BFifo, [71](#)
- readAvailableFromIsr
  - BQueue, [172](#)
- readBlocks
  - BSdio, [184](#)
- readData
  - BFifo, [71](#)
  - BHttpServer, [95](#)
- readDone
  - BFifo, [72](#)
- readFromIsr
  - BQueue, [172](#)
- readInt
  - BFifoChar, [77](#)
- readLine
  - BUart, [235](#)
- readPeak
  - BFifo, [72](#)
- readPos
  - BFifo, [72](#)
- readReg8
  - BI2c, [102](#)
- ReadType
  - BHttpServer, [92](#)
- rear
  - BList, [118](#)
- rebase
  - BFifo, [72](#)
- recvCanPacket
  - BCan, [33](#)
- refCount
  - BRefData, [174](#)

- refDelete
  - BRefString, [177](#)
- refMakeWriteable
  - BRefString, [177](#)
- refNew
  - BRefString, [177](#)
- refResize
  - BRefString, [177](#)
- refWrite
  - BRefString, [177](#)
- relocate
  - BArray, [20](#)
- removeSeparators
  - BString, [210](#)
- render
  - BHtml, [90](#)
- reset
  - BSys, [214](#)
  - BWatchdog, [255](#)
- resize
  - BArray, [20](#)
  - BFifo, [72](#)
- ResponseType
  - BSdio, [182](#)
- retFloat64
  - BString, [211](#)
- retInt
  - BString, [211](#)
- retStr
  - BString, [211](#)
- retUInt
  - BString, [211](#)
- reverse
  - BString, [211](#)
- roundSize
  - BBuffer.cpp, [269](#)
- rtosRunning
  - BSys, [216](#)
- run
  - BHttpServer, [95](#)
  - BNetworkDhcpTask, [135](#)
  - BNetworkProcessTask, [137](#)
  - BNtpClient, [140](#)
  - BSnmpServer, [195](#)
  - BSys, [214](#)
  - BTask, [218](#)
- runTasks
  - BTask, [218](#)
- rxBuffer
  - BNetwork.cpp, [322](#)
- rxDroppedPackets
  - BNetworkStats, [138](#)
- rxPackets
  - BNetworkStats, [138](#)
- SD\_CMD\_ALL\_SEND\_CID
  - BSdio.cpp, [353](#)
- SD\_CMD\_APP\_CMD
  - BSdio.cpp, [353](#)
- SD\_CMD\_APP\_SD\_SET\_BUSWIDTH
  - BSdio.cpp, [353](#)
- SD\_CMD\_CLR\_WRITE\_PROT
  - BSdio.cpp, [353](#)
- SD\_CMD\_ERASE\_GRP\_END
  - BSdio.cpp, [353](#)
- SD\_CMD\_ERASE\_GRP\_START
  - BSdio.cpp, [354](#)
- SD\_CMD\_ERASE
  - BSdio.cpp, [353](#)
- SD\_CMD\_FAST\_IO
  - BSdio.cpp, [354](#)
- SD\_CMD\_GEN\_CMD
  - BSdio.cpp, [354](#)
- SD\_CMD\_GO\_IDLE\_STATE
  - BSdio.cpp, [354](#)
- SD\_CMD\_GO\_INACTIVE\_STATE
  - BSdio.cpp, [354](#)
- SD\_CMD\_GO\_IRQ\_STATE
  - BSdio.cpp, [354](#)
- SD\_CMD\_HS\_BUSTEST\_READ
  - BSdio.cpp, [354](#)
- SD\_CMD\_HS\_BUSTEST\_WRITE
  - BSdio.cpp, [354](#)
- SD\_CMD\_HS\_SEND\_EXT\_CSD
  - BSdio.cpp, [355](#)
- SD\_CMD\_HS\_SWITCH
  - BSdio.cpp, [355](#)
- SD\_CMD\_LOCK\_UNLOCK
  - BSdio.cpp, [355](#)
- SD\_CMD\_NO\_CMD
  - BSdio.cpp, [355](#)
- SD\_CMD\_PROG\_CID
  - BSdio.cpp, [355](#)
- SD\_CMD\_PROG\_CSD
  - BSdio.cpp, [355](#)
- SD\_CMD\_READ\_DAT\_UNTIL\_STOP
  - BSdio.cpp, [355](#)
- SD\_CMD\_READ\_MULT\_BLOCK
  - BSdio.cpp, [355](#)
- SD\_CMD\_READ\_SINGLE\_BLOCK
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_APP\_OP\_COND
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_APP\_SEND\_NUM\_WRITE\_BLOCKS
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_APP\_SEND\_SCR
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_APP\_SET\_CLR\_CARD\_DETECT
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_APP\_STAUS
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_ERASE\_GRP\_END
  - BSdio.cpp, [356](#)
- SD\_CMD\_SD\_ERASE\_GRP\_START
  - BSdio.cpp, [356](#)
- SD\_CMD\_SDIO\_RW\_DIRECT
  - BSdio.cpp, [357](#)

SD\_CMD\_SDIO\_RW\_EXTENDED  
BSdio.cpp, 357

SD\_CMD\_SDIO\_SEN\_OP\_COND  
BSdio.cpp, 357

SD\_CMD\_SEL\_DESEL\_CARD  
BSdio.cpp, 357

SD\_CMD\_SEND\_CID  
BSdio.cpp, 357

SD\_CMD\_SEND\_CSD  
BSdio.cpp, 357

SD\_CMD\_SEND\_OP\_COND  
BSdio.cpp, 357

SD\_CMD\_SEND\_STATUS  
BSdio.cpp, 357

SD\_CMD\_SEND\_WRITE\_PROT  
BSdio.cpp, 358

SD\_CMD\_SET\_BLOCK\_COUNT  
BSdio.cpp, 358

SD\_CMD\_SET\_BLOCKLEN  
BSdio.cpp, 358

SD\_CMD\_SET\_DSR  
BSdio.cpp, 358

SD\_CMD\_SET\_REL\_ADDR  
BSdio.cpp, 358

SD\_CMD\_SET\_WRITE\_PROT  
BSdio.cpp, 358

SD\_CMD\_STOP\_TRANSMISSION  
BSdio.cpp, 358

SD\_CMD\_WRITE\_DAT\_UNTIL\_STOP  
BSdio.cpp, 358

SD\_CMD\_WRITE\_MULT\_BLOCK  
BSdio.cpp, 359

SD\_CMD\_WRITE\_SINGLE\_BLOCK  
BSdio.cpp, 359

SDCardState  
BSdio.cpp, 359

SDIO\_IRQHandler  
BSdio.cpp, 360

SDMMC2\_IRQHandler  
BSdio.cpp, 360

STATUS\_ERR\_BITS  
BSdio.cpp, 359

STATUS\_READY  
BSdio.cpp, 359

SampleTime\_112Cycles  
BAdc.cpp, 263

SampleTime\_144Cycles  
BAdc.cpp, 263

SampleTime\_15Cycles  
BAdc.cpp, 263

SampleTime\_28Cycles  
BAdc.cpp, 264

SampleTime\_3Cycles  
BAdc.cpp, 264

SampleTime\_480Cycles  
BAdc.cpp, 264

SampleTime\_56Cycles  
BAdc.cpp, 264

SampleTime\_84Cycles  
BAdc.cpp, 264

sendCanPacket  
BCan, 33

sendCmd  
BSdio, 184

sendEvent  
BoapMcServiceObject, 164

sendRecv  
BSpi, 197

serialNumber  
BUsbConfig, 243

set  
BError, 66  
BPwm, 169  
BSemaphore, 188  
BSemaphoreBool, 191  
BTime, 226  
BTimeUs, 231, 232

setAddress  
BoapMc1Comms, 144  
BoapMcClientObject, 153  
BoapMcComms, 157

setAlternateSource  
BGpio, 86

setBackupRegister  
BRtc, 180

setBlocking  
BUsbSerial, 246

setComms  
BoapMc1Comms, 144  
BoapMcComms, 158

setCommsMode  
BoapMc1Comms, 144  
BoapMcComms, 158

setCompleteCallback  
BDma, 62

setFromIsr  
BSemaphore, 188

setHeader  
BHttpServer, 95

setHexString  
BBufferStore, 30

setMaximumUsbChunk  
BUsbSerial, 247

setMode  
BSpi, 197

setMode485  
BUart, 236  
BUartBasic, 241

setNextBuffer  
BDma, 62

setPacketMode  
BComms, 40

setPin  
BGpio, 86

setPos  
BBufferStore, 30

setPriority  
     BInterrupt, 107  
     BTask, 219  
 setRtsCts  
     BUart, 236  
 setServerAddress  
     BNtpClient, 140  
 setSize  
     BBuffer, 23  
 setString  
     BTime, 227  
     BTimeUs, 232  
     IpAddress, 258  
 setSync  
     BUsbSerial, 247  
 setSystemTime  
     BNtpClient, 140  
 setTaskHeap  
     BSys, 214  
 setTime  
     BRtc, 180  
 setTimeout  
     BComms, 41  
     BoapMc1Comms, 145  
     BoapMcComms, 158  
 setValue  
     BCondInt, 45  
     BSemaphoreCount, 193  
 setYearDay  
     BTime, 227  
     BTimeUs, 232  
 Size  
     BDma, 60  
 size  
     BArray, 20  
     BBuffer, 23  
     BDataChunk, 50  
     BFifo, 73  
     BList, 118  
     BObjMember, 168  
     HeapTestState, 256  
 sntp\_set\_system\_time  
     BNtpClient.cpp, 324  
 sort  
     BList, 119  
 SortFunc  
     BList, 111  
 Source  
     BGpio, 84  
 Speed  
     BGpio, 85  
 split  
     BString, 211  
 start  
     BList, 119  
     BNetwork, 131  
     BSys, 215  
     BTask, 219  
     BTim, 222  
     BUsbSerial, 247  
     BWatchdog, 255  
 Status  
     BDma, 60  
 statusClear  
     BDma, 62  
 statusGet  
     BDma, 63  
 stop  
     BDma, 63  
     BTask, 219  
 str  
     BError, 66  
     BRefString, 177, 178  
     BString, 212  
 string  
     BoapMc1.h, 332  
     BoapMc1Error, 148  
 subString  
     BString, 212  
 swap  
     BList, 119  
 sync  
     BSdio, 185  
 sys  
     BSys.h, 375  
 sysClock  
     BClocks, 37  
 systemStatePrint  
     BSys, 215  
 TIM\_CCER\_CCE  
     BTim.cpp, 377  
 TIM\_CCMR1\_OCCE  
     BTim.cpp, 377  
 TX\_ZERO\_COPY  
     BNetwork.cpp, 320  
 table\_crc\_hi  
     BCrc16.cpp, 274  
 table\_crc\_lo  
     BCrc16.cpp, 275  
 take  
     BSemaphoreCount, 193  
 taskDelayUs  
     BTask.cpp, 376  
 taskFunc  
     BTask.cpp, 376  
 taskHeap  
     BSys, 216  
 taskStatsPrint  
     BSys, 215  
 time  
     BRtc, 180  
 timeUs  
     BSys, 215  
     BSys.cpp, 369  
     BSys.h, 374  
 timeUsReset

- BSys, 215
- Timeout
  - BHttpServer.cpp, 313
- timeoutTicks
  - BTypes.h, 388
- to\_hex
  - BString.h, 367
- toBDictStringFromJson
  - BObjStringFormat.cpp, 334
  - BObjStringFormat.h, 342
- toBString
  - BDict.cpp, 286
  - BDict.h, 287
  - BObjStringFormat.cpp, 334–337
  - BObjStringFormat.h, 342–345
- toBStringJson
  - BObjStringFormat.cpp, 337–340
  - BObjStringFormat.h, 345–348
- toBcd
  - BRtc.cpp, 350
- toLower
  - BString, 212
- toUpper
  - BString, 212
- togglePin
  - BGpio, 86
- transact
  - BSpi, 198
- transactFull
  - BSpi, 198
- trim
  - BSdio, 185
- TripleMode\_AlterTrig
  - BAdc.cpp, 264
- TripleMode\_InjecSimult
  - BAdc.cpp, 264
- TripleMode\_Interl
  - BAdc.cpp, 264
- TripleMode\_RegSimult
  - BAdc.cpp, 265
- TripleMode\_RegSimult\_AlterTrig
  - BAdc.cpp, 265
- TripleMode\_RegSimult\_InjecSimult
  - BAdc.cpp, 265
- truncate
  - BString, 212
- tryLock
  - BMutex, 121
- TwoSamplingDelay\_10Cycles
  - BAdc.cpp, 265
- TwoSamplingDelay\_11Cycles
  - BAdc.cpp, 265
- TwoSamplingDelay\_12Cycles
  - BAdc.cpp, 265
- TwoSamplingDelay\_13Cycles
  - BAdc.cpp, 265
- TwoSamplingDelay\_14Cycles
  - BAdc.cpp, 265
- TwoSamplingDelay\_15Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_16Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_17Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_18Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_19Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_20Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_5Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_6Cycles
  - BAdc.cpp, 266
- TwoSamplingDelay\_7Cycles
  - BAdc.cpp, 267
- TwoSamplingDelay\_8Cycles
  - BAdc.cpp, 267
- TwoSamplingDelay\_9Cycles
  - BAdc.cpp, 267
- txBuffer
  - BNetwork.cpp, 322
- txDroppedPackets
  - BNetworkStats, 138
- txPackets
  - BNetworkStats, 138
- type
  - BEvent, 68
  - BFirmwareInfo, 78
  - BObjMember, 168
  - HeapTestState, 256
- typeComp
  - BObjMember, 168
- typeName
  - BObjMember, 168
- UART4\_IRQHandler
  - BUart.cpp, 389
- UART5\_IRQHandler
  - BUart.cpp, 389
- UInt
  - BTypes.h, 385
- UInt16
  - BTypes.h, 385
- UInt32
  - BTypes.h, 385
- UInt64
  - BTypes.h, 385
- UInt8
  - BTypes.h, 385
- USART1\_IRQHandler
  - BUart.cpp, 389
- USART2\_IRQHandler
  - BUart.cpp, 389
- USART3\_IRQHandler
  - BUart.cpp, 389
- USART6\_IRQHandler

- BUart.cpp, 389
- USB\_CONFIGURATION\_FS\_STRING
  - BUsbSerial.cpp, 393
- USB\_CONFIGURATION\_HS\_STRING
  - BUsbSerial.cpp, 394
- USB\_INTERFACE\_FS\_STRING
  - BUsbSerial.cpp, 394
- USB\_INTERFACE\_HS\_STRING
  - BUsbSerial.cpp, 394
- USB\_LANGID\_STRING
  - BUsbSerial.cpp, 394
- USB\_LL\_ClearStallEP
  - BUsbSerial.cpp, 397
- USB\_LL\_CloseEP
  - BUsbSerial.cpp, 397
- USB\_LL\_DeInit
  - BUsbSerial.cpp, 397
- USB\_LL\_Delay
  - BUsbSerial.cpp, 397
- USB\_LL\_FlushEP
  - BUsbSerial.cpp, 397
- USB\_LL\_GetRxDataSize
  - BUsbSerial.cpp, 398
- USB\_LL\_Init
  - BUsbSerial.cpp, 398
- USB\_LL\_IsStallEP
  - BUsbSerial.cpp, 398
- USB\_LL\_OpenEP
  - BUsbSerial.cpp, 398
- USB\_LL\_PrepareReceive
  - BUsbSerial.cpp, 398
- USB\_LL\_SetUSBAddress
  - BUsbSerial.cpp, 398
- USB\_LL\_StallEP
  - BUsbSerial.cpp, 399
- USB\_LL\_Start
  - BUsbSerial.cpp, 399
- USB\_LL\_Stop
  - BUsbSerial.cpp, 399
- USB\_LL\_Transmit
  - BUsbSerial.cpp, 399
- USB\_MANUFACTURER\_STRING
  - BUsbSerial.cpp, 394
- USB\_PID
  - BUsbSerial.cpp, 394
- USB\_PRODUCT\_FS\_STRING
  - BUsbSerial.cpp, 394
- USB\_PRODUCT\_HS\_STRING
  - BUsbSerial.cpp, 394
- USB\_SERIALNUMBER\_FS\_STRING
  - BUsbSerial.cpp, 395
- USB\_SERIALNUMBER\_HS\_STRING
  - BUsbSerial.cpp, 395
- USB\_VCP\_ConfigStrDescriptor
  - BUsbSerial.cpp, 399
- USB\_VCP\_DeviceDescriptor
  - BUsbSerial.cpp, 399
- USB\_VCP\_InterfaceStrDescriptor
  - BUsbSerial.cpp, 400
- USB\_VCP\_LangIDStrDescriptor
  - BUsbSerial.cpp, 400
- USB\_VCP\_ManufacturerStrDescriptor
  - BUsbSerial.cpp, 400
- USB\_VCP\_ProductStrDescriptor
  - BUsbSerial.cpp, 400
- USB\_VCP\_SerialStrDescriptor
  - BUsbSerial.cpp, 400
- USB\_VID
  - BUsbSerial.cpp, 395
- USE\_DMA
  - BSdio.cpp, 359
- USE\_INTERRUPTS
  - BCan.cpp, 270
- USE\_SYNC
  - BRtc.cpp, 350
- uart
  - BUartBasic.cpp, 390
- uartWrite
  - BUartBasic.cpp, 390
- unlock
  - BCritical, 47
  - BMutex, 121
  - BMutexLock, 123
- urlDecode
  - BHtml.cpp, 309
  - BHtml.h, 311
- urlEncode
  - BHtml.cpp, 309
  - BHtml.h, 311
- urlEncodeComponent
  - BHtml.cpp, 310
  - BHtml.h, 311
- usbSerial
  - BUsbSerial.cpp, 404
- usbWrite
  - BUsbSerial.cpp, 402
- usbClass
  - BUsbSerial.cpp, 404
- usbDelInit
  - BUsbSerial, 247
  - BUsbSerial.cpp, 400
- usbGetDeviceQualifierDescriptor
  - BUsbSerial.cpp, 401
- usbGetFSCfgDesc
  - BUsbSerial.cpp, 401
- usbGetHSCfgDesc
  - BUsbSerial.cpp, 401
- usbGetOtherSpeedCfgDesc
  - BUsbSerial.cpp, 401
- usbInit
  - BUsbSerial, 247
  - BUsbSerial.cpp, 401
- usbInitHardware
  - BUsbSerial, 247
- usbInitUsbDev
  - BUsbSerial, 248

- usbInterrupt
  - BUsbSerial, 248
- usbRx
  - BUsbSerial, 248
  - BUsbSerial.cpp, 401
- usbRx0
  - BUsbSerial.cpp, 402
- usbSetup
  - BUsbSerial, 248
  - BUsbSerial.cpp, 402
- usbStarted
  - BUsbSerial, 248
- usbTx
  - BUsbSerial, 248
- usbTxComplete
  - BUsbSerial, 249
- usbTxSent
  - BUsbSerial.cpp, 402
- vApplicationGetIdleTaskMemory
  - BSystem.cpp, 369
- vApplicationIdleHook
  - BTask.cpp, 376
- VCP\_Desc
  - BUsbSerial.cpp, 405
- valid
  - BIter, 108
- value
  - BCondInt, 45
  - BDictItem, 57
  - BRandom, 173
  - BSemaphoreBool, 191
  - BSemaphoreCount, 194
- vendorId
  - BUsbConfig, 243
- ver0
  - BFirmwareInfo, 78
- ver1
  - BFirmwareInfo, 78
- ver2
  - BFirmwareInfo, 78
- wait
  - BComms, 41
  - BDma, 63
  - BSemaphore, 189
  - BSemaphoreBool, 191
  - BSemaphoreCount, 194
  - BUart, 236
  - BUsbSerial, 249
- waitFor
  - BI2c, 103
- waitForSync
  - BRtc, 180
- waitLessThan
  - BCondInt, 45
- waitLessThanOrEqual
  - BCondInt, 45
- waitMoreThanOrEqual
  - BCondInt, 46
- waitStatus
  - BDma, 63
- waitUntilReady
  - BFlash, 81
- write
  - BCan, 33
  - BComms, 41
  - BFifo, 73
  - BI2c, 103
  - BQueue, 172
  - BUart, 236
  - BUartBasic, 241
  - BUsbSerial, 249
  - BUsbSerialLock, 254
- writeAddress16
  - BI2c, 103
- writeAddress8
  - BI2c, 103
- writeAvailable
  - BCan, 34
  - BComms, 41
  - BFifo, 73
  - BQueue, 172
  - BUart, 236
  - BUartBasic, 241
  - BUsbSerial, 249
  - BUsbSerialLock, 254
- writeAvailableChunk
  - BFifo, 73
- writeBackup
  - BFifo, 74
- writeBlocks
  - BSdio, 185
- writeChunk
  - BHttpServer, 95
- writeChunks
  - BComms, 41
- writeData
  - BFifo, 74
- writeDone
  - BFifo, 74
- writeFlush
  - BHttpServer, 96
- writeFromIsr
  - BQueue, 172
- writeHead
  - BHttpServer, 96
- writePos
  - BFifo, 74
- writeProtect
  - BFlash, 82
- writeReg8
  - BI2c, 103
- writeString
  - BHttpServer, 96
- writeTableInput
  - BHttpServer, 96

writeTableRow  
  BHttpRequest, [96](#)

writeTail  
  BHttpRequest, [97](#)

xPortSysTickHandler  
  BSystem.cpp, [369](#)

yearDays  
  BTime.cpp, [378](#)  
  BTimeUs.cpp, [379](#)

yearIsLeap  
  BTime.cpp, [378](#)  
  BTimeUs.cpp, [379](#)