```cpp
#include "BBB.h"

namespace Hardware
{
    /*! Constructor*/
    BBB::BBB()
    {
        threadRunning = false;
        callbackFunction = NULL;
        debounceTime = 0;
        thread = (pthread_t)NULL;
    }

    /*! De-constructor*/
    BBB::~BBB() {   }

    /*! Reads the first line from a file
    \param path constant string pointing towards the file
    \returns this first line
    */
    string BBB::Read(const string &path)
    {
        ifstream fs;
        fs.open(path.c_str());
        if (!fs.is_open())  { throw Exception::GPIOReadException(("Can't open: " + path).c_str());  }
        string input;
        getline(fs, input);
        fs.close();
        return input;
    }

    /*! Writes a value to a file
    \param path a constant string pointing towards the file
    \param value a constant string which should be written in the file
    */
    void BBB::Write(const string &path, const string &value)
    {
        ofstream fs;
        fs.open(path.c_str());
        if (!fs.is_open()){ throw Exception::GPIOReadException(("Can't open: " + path).c_str()); }
        fs << value;
```

```
42          fs.close();
43      }
44
45      /*! Checks if a directory exist
46      \returns true if the directory exists and false if not
47      */
48      bool BBB::DirectoryExist(const string &path)
49      {
50          struct stat st;
51          if (stat((char *)path.c_str(), &st) != 0) { return false; }
52          return true;
53      }
54
55      /*! Checks if a cape is loaded in the file /sys/devices/bone_capemgr.9/slots
56      \param shield a const search string which is a (part) of the shield name
57      \return true if the search string is found otherwise false
58      */
59      bool BBB::CapeLoaded(const string &shield)
60      {
61          bool shieldFound = false;
62
63          ifstream fs;
64          fs.open(SLOTS);
65          if (!fs.is_open()) { throw Exception::GPIOReadException("Can't open SLOTS"); }
66
67          string line;
68          while (getline(fs, line))
69          {
70              if (line.find(shield) != string::npos) { shieldFound = true; break; }
71          }
72          fs.close();
73          return shieldFound;
74      }
75  }
```