

**Real Time Software Engineering Task 2**

**Written by:**

**Devon Kerai**

**Simon Tate**

**Guoyue Wang**

**Introduction**

The task was to write a set of device drivers in C++ for the individual units on the washing machine. The device drivers needed to be independent as much as possible and it also had to be written in such a way so that it was simple to re-map the drivers to an alternative target platform. Three files were created to split the code into different layers: The application layer, the device layer and the target hardware layer. The top application layer is called *main.cpp*, the device layer is called *functions.cpp* and the target hardware layer is called *classes.h*.

**Main.cpp**

*Main.cpp*, the application layer, contains all the functions to interact with the individual units on the washing machine simulator. This allows someone who is unfamiliar with the code to test the full range of the operations of the device drivers.

To operate a unit on the washing machine, you must create an object of its class:

*Motor myMotor;*

To execute functions within that class, call them using the dot operator:

*myMotor.motorOn;*

*myMotor.turnMotorClockwise();*

**Functions.cpp**

*Functions.cpp*, the device layer, contains all the function definitions of each class. These functions are called and used from within the application layer (*main.cpp*).

**Classes.h**

*Classes.h*, the target hardware layer, contains all the prototypes for the classes and defines the addresses of the ports. Eleven classes were built: *Motor*, *Buzzer*, *Seven Seg Display*, *Door*, *Accept Button*, *Cancel Button*, *Programme 1*, *Programme 2*, *Programme 3*, *Ports*, and *Timer*.

This file is the only location which assigns the ports to different identifier. If our code was to be used on a different platform, only this file will need to be modified.

**Shortcomings**

If the user wanted to use a different board with different or additional hardware, *classes.h* would have to be changed. Ports must be configured manually in *main.cpp* and *functions.cpp*, making the application and device layer not completely independent from the target hardware layer.