Guoyue Wang (....7301)

COVER SHEET FOR ASSESSED COURSEWORK

Students must complete this coversheet to accompany each piece of assessed coursework.

PROGRAMME: Electronic and Electrical Engineering (ELUB10) YEAR/PART: C

MODULE: Real Time Software Engineering (14ELC018)

ASSESSMENT TITLE: C++ device drivers

STAFF MEMBER RESPONSIBLE: David Mulvaney

Student Declaration (please sign and date)

I certify that the attached work is my own work and that anything taken from or based upon the work of others has its source clearly and explicitly cited. I certify that each and every quotation, diagram or other piece of exposition which is copied from or based upon the work of others has its source clearly acknowledged in the text. All field work and/or laboratory work has been carried out by me (or my group) with no more assistance from the members of the department/school than has been specified. Any additional assistance which has been received is indicated and referenced in my work.

I have abided by the department/school policy on plagiarism and the Coursework Code of Practice as specified in the Student Handbook (see Learn).

Submitted By:

Group: Group 5

Guoyue Wang (....7301) Date: 06/02/2015 Signature: ....

Devon Kerai (....8203) Signature (if applicable): ....

Simon Alexander Tate (....4127) Signature (if applicable): ....

Specific Items: Written documents(1)

Note:
Late-Hand-In: Approval for an extension to the deadline must be sought from the Responsible Examiner prior to the deadline. For any submission handed in after an agreed extension you must ensure that you have completed an Impaired Performance Form for your mark to be considered by the Module Board.

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.