

# CS7004 - Lecture Theatre Lighting Control System

Wesley Fung

12304003

School of Computer Science and Engineering  
Trinity College Dublin

fungw@tcd.ie

This document is concerned with the implementation of the project “Lecture Theatre Lighting Control System” for CS7004 - Embedded Systems 2016.

## Introduction

Lecture Theatre Lighting Control System simulated on the LPC2468 micro-controller. The board features 4 I<sup>2</sup>C connected LEDs. Listed below are the core functionalities specified by the project brief which has been implemented.

- master on/off switch
- independent on/off/dimmer control of each of the four zones
- two stored, reconfigurable presets
- PIR motion sensor inputs that automatically turn on and keep on the lights when motion is detected or turn off the lights when no motion is detected for more than a preset time

On top of this, additional functionality has been implemented to expand the functions that the micro-controller can undertake. Listed below are the additional features implemented.

- Fire alarm triggered by inputting an activation code and deactivated through a deactivation code.
- “Clap” function, simulation of a clapping motion in front of a PIR motion sensor to turn ON/OFF the corresponding light. Triggered by double clicking on any one of the four push buttons.
- On the user interface, hold preset button to save configuration, double tap preset button to force it into preset state.
- Motion detection has been altered, press any push buttons to simulate motion detection on push buttons. Hold after timeout to revert to previous known state.

The project itself features extensive use of FreeRTOS software timers and queues to control the interactions between the user interface and the I<sup>2</sup>C push buttons. Extensive code coverage in the form of comments is also included in the source code.

## Requirements

Certain configurations must be set on KEIL  $\mu$ Vision prior to flashing the software onto the micro-controller. Listed below are the configurations that must be made.

Submitted to:  
Jonathan Dukes

- Optimisation must be set to Level 3 (-O3) within the C/C++ tab of target configurations.
- Thumb-mode instead of ARM-mode set within the Target tab of target configurations; this is due to the project exceeding the optimisation size of previous point.
- FreeRTOS library in the core directory of the project.

Please note the following as LCD touch register events varies. Listed below are some noticeable touchscreen offsets.

- Minus buttons registers as presses a few pixels to the left of the actual button.
- Similarly with the plus buttons. Button registers to the right of the actual button.
- Preset buttons should be held as close to the bottom of the screen to register.

## 1 Design

### 1.1 Controller View Design

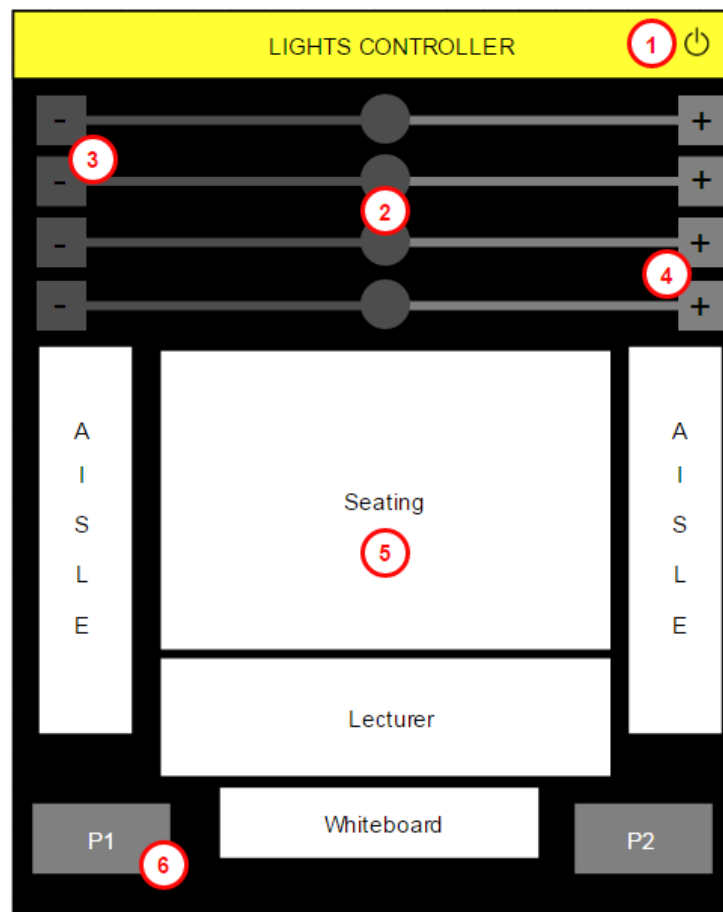


Figure 1: Controller View

## Reference

1. Master power ON/OFF switch
2. Slider indicators
3. Minus button to control specific slider button
4. Plus button to control specific slider button
5. Theatre lighting buttons
6. Preset buttons, reconfigurable

## Interaction

1. On press, turns ON all lights when all lights are OFF. Turns OFF all lights when any lights are ON.
2. Indicator for each of the four corresponding lighting areas. Indicates what light setting its corresponding light is on. Not interactable directly, controlled through the plus/minus buttons connected to corresponding slider.<sup>1</sup>
3. On press, moves the slider from the state that it is at to the previous set state.<sup>1</sup>
4. On press, moves the slider from the state that it is at to the next set state.<sup>1</sup>
5. On press, will turn the corresponding lecture area ON (GREEN) if OFF. Corresponding LED and slider indicator will reflect new state. On press, will turn the corresponding lecture area OFF (WHITE) if ON. Corresponding LED and slider will reflect new state.
6. On hold, will save the current state of the lecture theatre to held preset button. On double click, will force the state of the preset configuration. LED, slider and theatre button will reflect all necessary changes.

## 1.2 Slider Specification

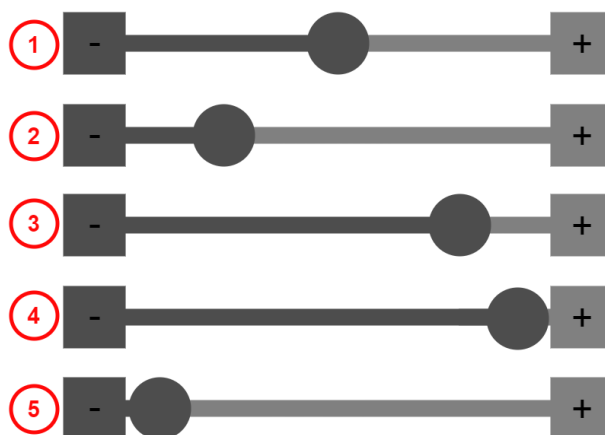


Figure 2: Slider Specification

<sup>1</sup>Refer to figure 2

## Reference

1. State 3; set OFF; locked
2. State 2; read from PWM1
3. State 4; read from PWM0
4. State 5; set ON; unlocked
5. State 1; set OFF; unlocked

Every time plus/minus buttons pressed, the corresponding slider will go from its current state to +1/-1, next/previous states.

### 1.3 Push Buttons Specification

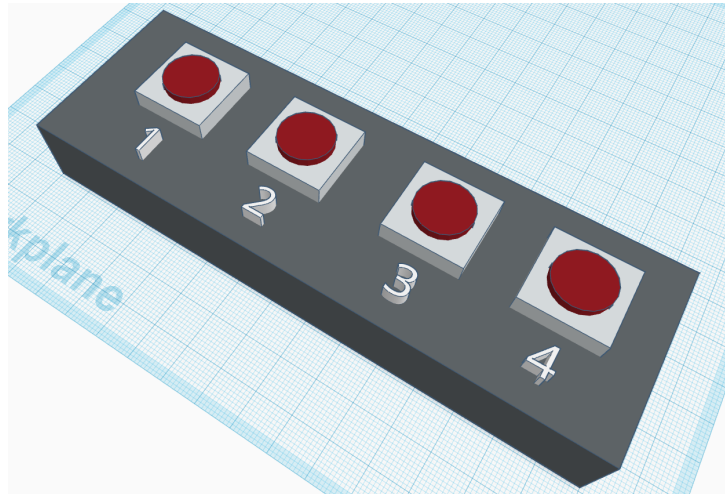


Figure 3: Push Buttons Specification

## Reference

Each button corresponds to one of the four lecture theatre lighting areas.

1. Whiteboard
2. Lecturer
3. Seating
4. Aisles (Both)

## Interface Interaction

- Input fire activation code.<sup>2</sup> Default code is: 3211. Code corresponds to pressing button #3, #2, #1, #1 in that order to trigger fire alarm.

---

<sup>2</sup>Global variable@sensors.c <FIRE\_ACTIVATE\_CODE >

- Input fire deactivation code.<sup>3</sup> Default code is 2141. Code corresponds to pressing button #2, #1, #4, #1 in that order to turn off fire alarm.
- Double click any one of the four push buttons to turn its corresponding light ON/OFF.
- Hold or press any one of the four push buttons to simulate motion detection; resets timer. Motion sensors keep on all lights. If motion is detected after motion sensor timeout expires, state is retained and theatre is re-initiated in state prior to timeout.

## 2 Implementation

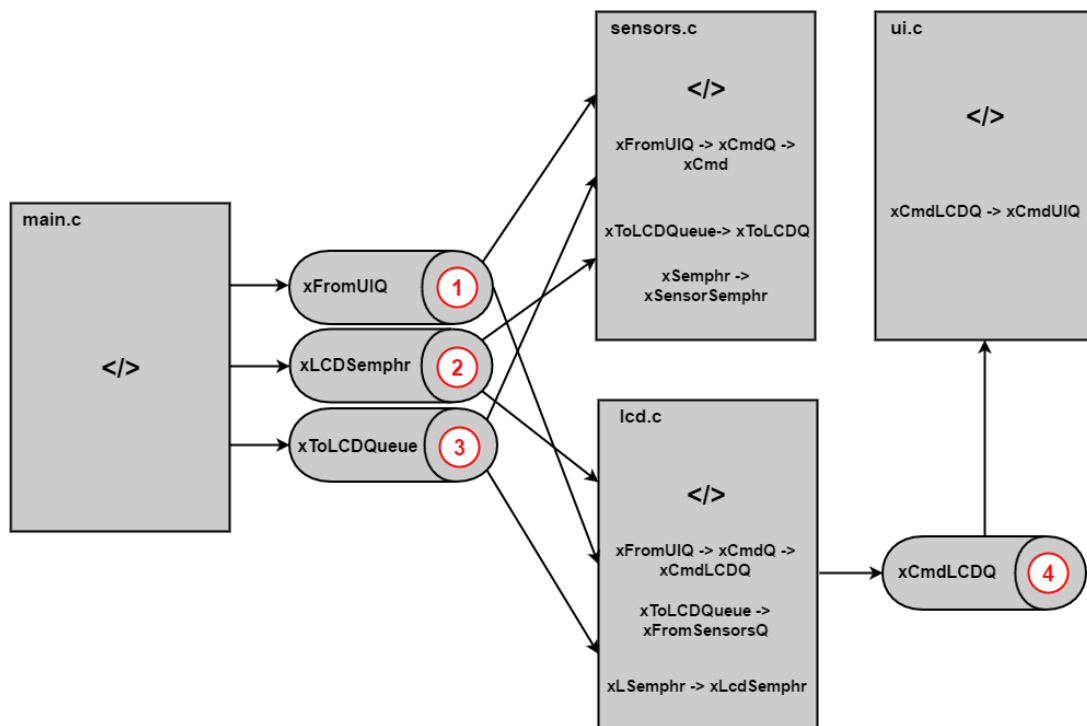


Figure 4: Queues Architecture

Most of the backbone of the project revolves around the use of queues. Two queues are instantiated in the main.c and passed as parameters into the sensors and LCD tasks respectively.

### 2.1 Queues

#### 2.1.1 xFromUIQ

- Passed onto ui.c via the lcd.c task. Double click of presets timer within UI, when double click on preset is detected, UI pops relevant command onto queue so that sensors file will reflect the saved configuration.
- Sensors task receives commands from LCD on this queue.

<sup>3</sup>Global variable@sensors.c <FIRE\_DEACTIVATE\_CODE >

### 2.1.2 xLCDSemphr

- LCD TS semaphore extracted from lcd.c. Created in main.c, passed into both sensors.c and lcd.c. This is so that an action that occurs on the sensors side may be reflected on the ui by giving a semaphore to the semaphore that blocks on lcd side.

### 2.1.3 xToLCDQueue

- LCD task receives commands from sensors on this queue.

### 2.1.4 xCmdLCDQ

- Same queue as xFromUIQ. UI is allowed to send commands on this queue. Sensors task is on the receivers end.

## 2.2 Double Click

- Used for turning on the lights via the I<sup>2</sup>C buttons. Double click the I<sup>2</sup>C buttons to turn corresponding light ON/OFF.
- Also used for forcing the state of a preset. Double tap the preset buttons to force the state of the reconfigurable setting.

### 2.2.1 Clap function

- Forces an individual LED to turn ON/OFF when any I<sup>2</sup>C push button is double clicked.
- Button states only saved on ui side. Sensors side have no state holding property to determine whether or not the LED should be turned ON or OFF. Implementation involves querying the ui side for the state of the button and to then construct commands for both sensor and LCD to reflect the changes.
- FreeRTOS software timer is used. Default time is 500ms. If two clicks are registered within 500ms, then a double click event is triggered.

### 2.2.2 Preset function

- Forces the entire systems state into one of two reconfigurable states.
- UI handles the double tap functionality of preset buttons. Similar implementation to that of clap function.
- Use of FreeRTOS software timers. Timer starts on click of preset buttons. When timer is up, checks whether or not two clicks have been registered within the default time of 750ms.

## 2.3 Fire Alarm

- Activation code triggers the fire alarm.
- Deactivation code turns the fire alarm OFF.
- Activation and deactivation codes are stored in array. At each I<sup>2</sup>C button registered, checks whether the correct combination of code has been entered.

- Fire alarm suspension state (alternating LED lights and alternating BLUE and RED buttons) is implemented with a FreeRTOS software timer.
- When in suspended state, the LEDs are forced into a pre-determined alternating fire state<sup>4</sup>. Changes are sent to LCD queue so that the alternating state may be changed on UI side. Timer is reset at every interval and simple boolean logic upholds the timeout alternation.

## 2.4 Motion Sensors

- Motion sensors keep the LEDs ON when there are LEDs in the ON state.
- After motion sensors times out, hold any I<sup>2</sup>C button to revert to previous state before timeout.
- Use of FreeRTOS software timer. Default time for timeout is 30s.
- Every time an I<sup>2</sup>C button is interacted with, the timer is reset.
- When timeout occurs, LED lights turn OFF, and UI is asked to shutdown appropriately.

## 2.5 Preset Buttons Interaction

- Double tap functionality explained in previous section.
- Holding the preset buttons saves the current instance of the configuration of the LED lighting system.
- Implemented through the use of FreeRTOS software timers.
- Default timer is 2s for saving and corresponding “Configuration Saved” message pop up occurs for 3s.

## 2.6 LEDs

- PWM0 and PWM1 are set at 75% ( $255 * 75\%$ ) and 25% ( $255 * 25\%$ ) respectively.
- Bit manipulation is how the change of states of the I<sup>2</sup>C LEDs are changed. Generation of mask to clear the relevant bit is done first to ease the bit manipulation process straight after.
- Bits set accordingly depending on which state they wish to point to (**ON**(01) / **OFF**(00) / **PWM0**(10) / **PWM1**(11))

---

<sup>4</sup>Global variable@sensors.c <STATE\_FIRE1 >& <STATE\_FIRE2 >

### 3 Functional Testing

#### 3.1 User Interface

Listed below are most of the cases possible within the touchscreen user interface. All test cases have been tested and verified.

Test Case	Expected Outcome	Actual Outcome	Pass
<b>Power</b>			
Press POWER when all lights OFF	All BUTTONS GREEN, SLIDER (state 5) <sup>5</sup>	All BUTTONS GREEN, SLIDER (state 5) <sup>5</sup>	✓
Press POWER when one or more lights ON	All BUTTONS WHITE, SLIDER (state 3) <sup>5</sup>	All BUTTONS WHITE, SLIDER (state 3) <sup>5</sup>	✓
<b>Theatre Buttons</b>			
Press any BUTTON when BUTTON state is OFF	BUTTON turns GREEN, SLIDER (state 5) <sup>5</sup>	BUTTON turns GREEN, SLIDER (state 5) <sup>5</sup>	✓
Press any BUTTON when state is ON	BUTTON turns WHITE, SLIDER (state 3) <sup>5</sup>	BUTTON turns WHITE, SLIDER (state 3) <sup>5</sup>	✓
<b>Plus/Minus</b>			
Press PLUS or MINUS when all lights OFF	SLIDER locked (state 3) <sup>5</sup> , nothing happens	Nothing happens	✓
Press PLUS when state is BUTTON is ON	SLIDER goes +1 state <sup>5</sup> , turns BUTTON GREEN	SLIDER goes +1 state <sup>5</sup> , BUTTON turns GREEN	✓
Press MINUS when state is BUTTON is ON	SLIDER goes -1 state <sup>5</sup> , turns BUTTON GREEN	SLIDER goes -1 state <sup>5</sup> , BUTTON turns GREEN	✓
<b>Presets</b>			
Hold any preset; default (2s)	Configuration saved message pops up	Configuration saved message pops up (Shows for default 3s)	✓
Double tap any preset	BUTTONS reflect saved state, SLIDERS reflect saved state	BUTTONS and SLIDERS reflect saved state	✓

---

<sup>5</sup>Refer to figure 2



<b>Fire Alarm</b>			
All lights OFF, ACTIVATION code entered	BUTTONS alternating BLUE and RED	BUTTONS alternate BLUE and RED	✓
One or more lights ON, ACTIVATION code entered	SHUTDOWN, SLIDERS (state 3) <sup>6</sup> , BUTTONS WHITE. Then BUTTONS alternating BLUE and RED	SHUTS DOWN, SLIDERS (state 3) <sup>6</sup> , BUTTON WHITE. Alternating BLUE and RED	✓
Fire alarm state, ACTIVATION code entered	Nothing	Nothing happens	✓
Fire alarm state, DEACTIVATION code entered	Turns off fire alarm	Fire alarm turned off	✓
Not in fire alarm state, DEACTIVATION code entered	Nothing happens	Nothing happens	✓
<b>Motion Detection</b>			
Any light ON, press any I <sup>2</sup> C BUTTON	Keeps SLIDER and BUTTON in suspended state	Kept SLIDER and BUTTON in suspended state	✓
No lights ON, press any I <sup>2</sup> C BUTTON	Nothing happens	Nothing happens	✓
HOLD any I <sup>2</sup> C BUTTON after TIMEOUT	Reverts to state prior to timeout	SLIDER and BUTTON returns to state before timeout	✓
Wait for motion timeout	SHUTDOWN, BUTTON turns WHITE, slider (state 3) <sup>6</sup>	SHUTDOWN, BUTTON turns WHITE, slider (state 3) <sup>6</sup>	✓
<b>Clap Clap</b>			
Double click I <sup>2</sup> C BUTTON, BUTTON state is OFF	BUTTON turns GREEN, SLIDER (state 5) <sup>6</sup>	BUTTON turns GREEN, SLIDER (state 5) <sup>6</sup>	✓
Double click I <sup>2</sup> C BUTTON, BUTTON state is ON	BUTTON turns WHITE, SLIDER (state 3) <sup>6</sup>	BUTTON turns WHITE, SLIDER (state 3) <sup>6</sup>	✓
Double click I <sup>2</sup> C BUTTON, BUTTON state is DIMMED	BUTTON turns WHITE, SLIDER (state 3) <sup>6</sup>	BUTTON turns WHITE, SLIDER (state 3) <sup>6</sup>	✓

---

<sup>6</sup>Refer to figure 2

### 3.2 Sensors

Listed below are the interactable test cases tested and verified for the 4 push buttons on the micro-controller.

Test Case	Expected Outcome	Actual Outcome	Pass
<b>Fire Alarm</b>			
All lights OFF, ACTIVATION code entered	Alternating fire LED state	LEDs alternating in fire state	✓
One or more lights ON, ACTIVATION code entered	Alternating fire LED state	LEDs alternating in fire state	✓
Fire alarm state, ACTIVATION code entered	Nothing happens	Nothing happens	✓
Fire alarm state, DEACTIVATION code entered	All LEDs turn OFF	All LEDs turned OFF	✓
Not in fire alarm state, DEACTIVATION code entered	Nothing happens	Nothing happens	✓
<b>Motion Detection</b>			
Any LEDs ON, press any I <sup>2</sup> C BUTTON	Keeps LEDs ON	Kept all LEDs in suspended state	✓
No LEDs ON, press any I <sup>2</sup> C BUTTON	Nothing happens	Nothing happens	✓
HOLD any I <sup>2</sup> C BUTTON after TIMEOUT	Reverts LEDs to state before timeout	Reverts LEDs to state before timeout	✓
Wait for motion timeout	LED turns OFF	LED turned OFF	✓
<b>Clap Clap</b>			
Double click I <sup>2</sup> C BUTTON, LED is OFF	Corresponding LED turns ON	Corresponding LED turned ON	✓
Double click I <sup>2</sup> C BUTTON, LED is ON	Corresponding LED turns OFF	Corresponding LED turned OFF	✓
Double click I <sup>2</sup> C BUTTON, LED is DIMMED	Corresponding LED turns OFF	Corresponding LED turned ON	✓