Embedded Systems Development Group Report

This report will discuss the design and development of a burglar alarm as a part of the Electronics and Embedded Hardware module. The module was constructed on a foundation of groupwork and therefore the report will reflect the individual contributions of two authors. For the purpose of this assignment, the contributors will be referred to as Author 1 and Author 2. An analysis of the technologies used to create a fully functioning burglar alarm using the QL-200 Development Board, will be presented, and will incorporate an examination of the driver code employed to control the devices. A critical evaluation will highlight the individual contributions and challenges faced. A reflection of these challenges and resolutions will then be discussed.

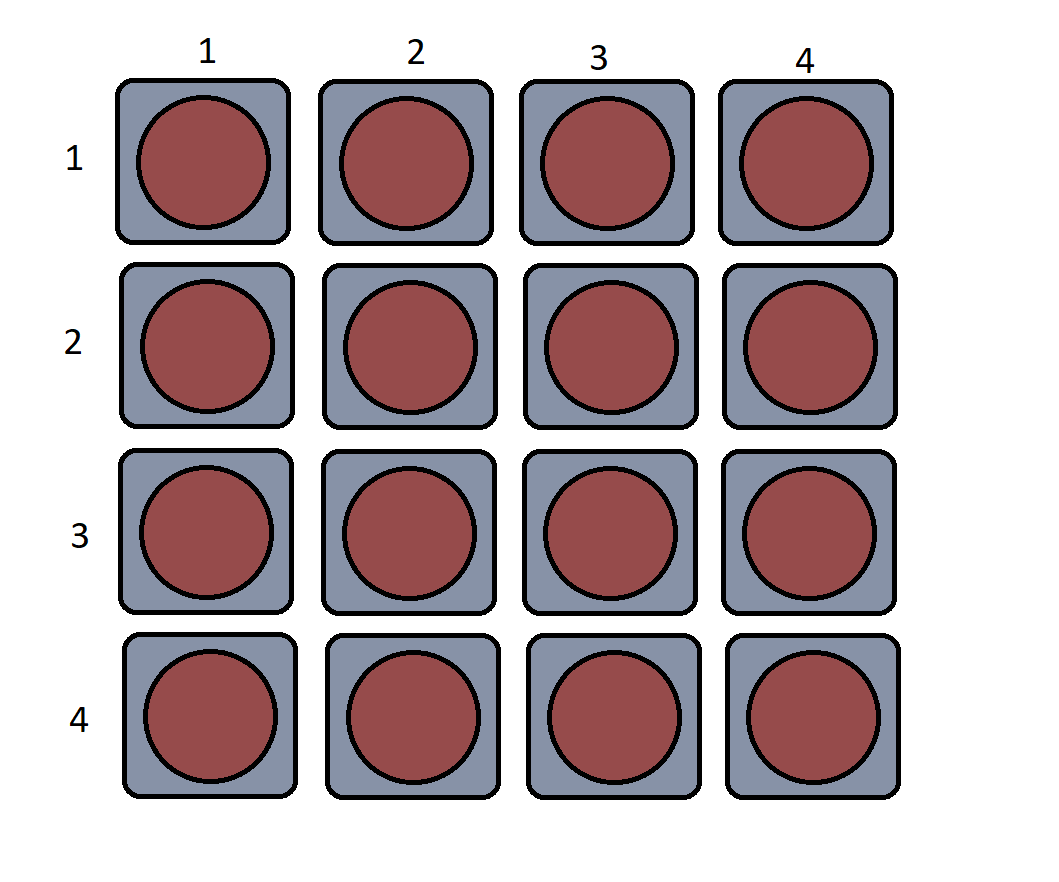
The overall project aim was to utilise a QL-200 development board and PIC16F877A microcontroller to create a burglar alarm which allowed for user input. The device monitors four separate security zones which can be activated or deactivated manually. Zone four contains a temperature based sensor which activates the alarm if the temperature rises above a threshold for a predetermined amount of time. The temperature threshold, activation time and alert time can be set using the 4x4 matrix buttons when the devices is first powered. An LCD panel would be employed to display information pertinent to the user such as the temperature of Zone 0, the current date and time and the activation status of each zone.

LCD Display

The LCD display located towards the upper left of the board will automatically be turned on. When a button is pressed for a function, the LCD display will show a message related to the function; this will either be an instruction or a statement. For example, pressing the alarm on button with cause the message “Alarm on” to appear, followed by the message “Select a function” and pressing the alarm off button will cause the message “Alarm off” to display.

Buttons

The buttons are the main way for a user to interact with the alarm system and will be used to access the features of the alarm system that will be useful to the user. Most buttons will give the user an output to tell which function has been pressed. The system features a layered, context-sensitive button system in which some buttons will have different functions depending on what mode the system is set to, which is determined by certain buttons. Below is a diagram of the button layout, with coordinates.



Below is a list of functions (see further down for information on each function) and its button allocation (layered button functions will have corresponding number (n) to their respective mode button **(n)**):

*(The button layout is row x column)*

* 1x1: Zone check mode **(1)**/Increase hours (4)/Increase Year (5)
* 1x2: Check zone 0 (1)/Activate zone 0 (2)/Temperature change mode **(3)**/Increase minutes (4)/Increase month (5)/Increase Alarm Delay (6)
* 1x3: Deactivate zone 1 (2)/Increase trigger temperature (3)/Increase seconds (4)/Increase day (5)
* 1x4: Return
* 2x1: Zone activate mode **(2)**/Decrease hours (4)/Decrease year (5)
* 2x2: Check zone 1 (1)/Activate zone 1 (2)/Decrease minutes (4)/Decrease month (5)/Decrease Alarm Delay (6)
* 2x3: Deactivate zone 1 (2)/Decrease trigger temperature (3)/Decrease seconds (4)/Decrease day (5)
* 2x4: Show current temperature
* 3x1: Change delay mode **(6)**
* 3x2: Check zone 2 (1)/Activate zone 2 (2)/Show current date
* 3x3: Deactivate zone 1 (2)/Time change mode (4)
* 3x4: Alarm off
* 4x1:
* 4x2: Check zone 3 (1)/Activate zone 3 (2)/Show current time
* 4x3: Deactivate zone 1 (2)/Date change mode (5)
* 4x4: Alarm on

It is important to note the 8 small switches next to the 16 buttons must be set to on, otherwise the buttons will not work.

The DS18B20 temperature sensor was used in the development of the alarm to provide an accurate temperature measurement in degrees Celsius. Displaying the current temperature requires the user to select the designated button specified in the driver code, this will output the temperature onto the 1602 LCD to one decimal place. As this continuously calls the update temperature method located in the temperature class, the temperature value shown on the LCD will always be the update value and show the correct temperature for the room. The alarm must also determine whether the current temperature of the room is above a specific threshold set by the user. When the alarm is powered on, the trigger temperature value must be set. To set the trigger temperature, the user must first enter change temp mode by selecting a pre-defined button. When in this mode, there are three buttons which can be used; button one increments the trigger value one degree at a time. Button one only has to be pressed once as the value will continue to increase until the value is saved. Saving the desired trigger value can be achieved by selecting the ‘save’ button. When the value has risen to the desired temperature threshold, holding the save button down for two seconds saves the value. Decreasing the trigger value is achieved by the user selecting the designated button. Selecting the button will decrease the threshold one degree at a time and can be stopped by pressing the save button.

Altering the trigger temperature can be achieved at any time while the alarm is in operation. The user simply needs to enter into check temperature mode and increase or decrease the value to their desired preference.

The real-time clock was used to provide time based features for the burglar alarm. Setting the time was achieved by storing the individual time values in an array which is then output onto the 1602 LCD. To view the current time, the user must select the designated show time button located on row four of column two. This will allow the user to view the time in a 24-hour format. The date can be viewed by pressing the show date button located on row three of column two which is printed out in the following format dd/mm/yy.

The alarm should only detect intrusions and sound the alarm inside a specific set of pre-defined hours. The alarm will detect instructions between the hours of 8:00 in the morning and 7:30 at night, outside of these hours, the alarm will be deactivated and will not sound. These values are pre-set meaning they cannot be change by the user.

The buzzer is a simple method which is called when the alarm detects an intrusion. The user must first enable the alarm by selecting the ‘alarm on’ button located on row four column four, the LCD should then Display ‘Alarm On’ indicated the armed status. If an intruder is then detected, the buzzer will sound. In the case of the temperature sensor, the buzzer will remain active until the room temperature has dropped below the trigger value set by the user.

Functions

Each function of the system will be given below with a description and instructions.

Alarm on – Press the alarm on button to turn on the alarm. The message “Alarm on” will show, “select a function” will then follow.

Alarm off – Press the alarm off button to turn off the alarm. The message “Alarm off” will show. Note that the other functions of the system can only be accessed if the alarm is on.

Zone check – Press the zone check mode button, the message “Check” will show; the system will now be in Zone check mode. Now press one of the zone check buttons, each one will show the current status of a zone.

Zone activate/deactivate – Press the zone activate mode button, the message “Activate” will show; the system will now be in Zone activate mode. Now press one of the zone activation buttons to activate a zone, or press one of the zone deactivation buttons to deactivate a zone. When pressed an appropriate message will be displayed.

Trigger temperature – Press the trigger temperature mode button; then press either the increase or decrease temperature buttons, the trigger temperature will show on the LCD and will increase or decrease by one each second, when it has reached the desired value, press the return button.

Time change – Press the time change mode button. Then press the increase or decrease buttons for the seconds, minutes or hours, depending on what needs to be changed. The value will show on the LCD and increase or decrease by one each second, when it has reached the desired value, press the return button.

Date change – Press the date change mode button. Then press the increase or decrease buttons for the days, months or years, depending on what needs to be changed. The value will show on the LCD and increase or decrease by one each second, when it has reached the desired value, press the return button.

Alarm delay length – Press the alarm delay button, then press the increase or decrease buttons. The value will show on the LCD and increase or decrease by one each second, when it has reached the desired value, press the return button. This will then determine how long the alarm will wait before alerting, with the delay value equalling the number of seconds.

Show time – Press the show time button to see the current time.

Show date – Press the show date button to see the current date.

Return – To back out of any mode, press the return button. Note that in order to enter a different mode, the return button must be pressed first.

Temperature trigger – Once the temperature detected by the thermometer is greater or equal to the set trigger temperature, the alarm will sound after a given delay and Zone 0 will show an alert message. The current temperature must be lower than the trigger temperature for the alarm to stop.

Critical Evaluation

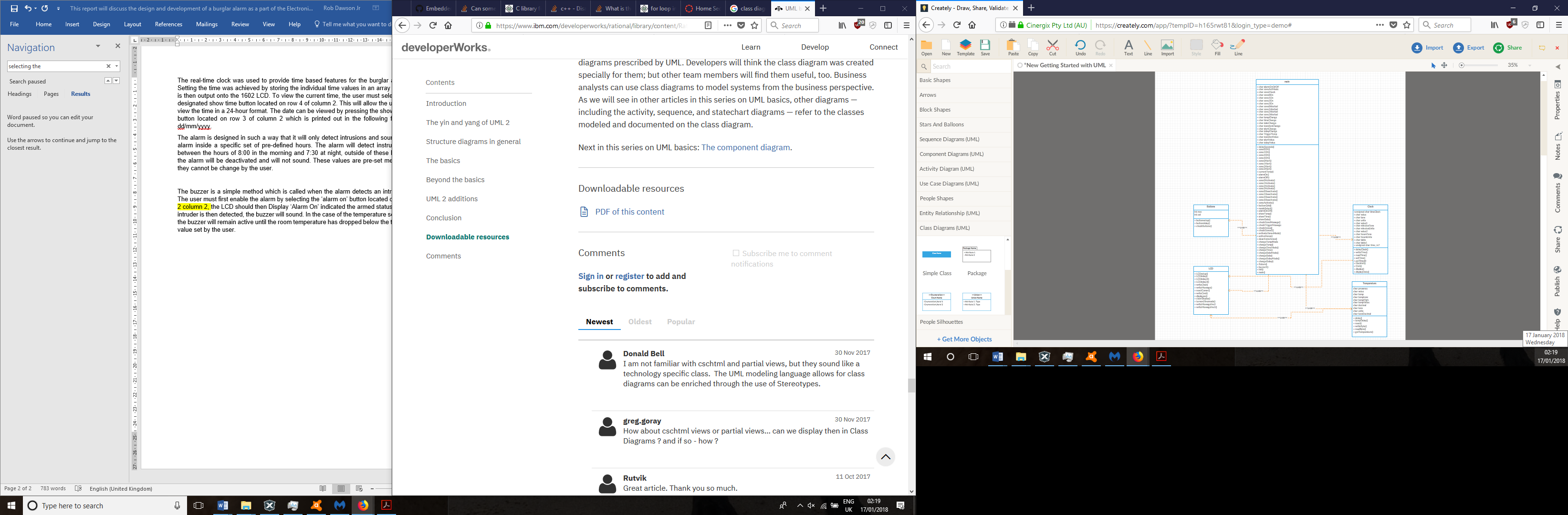
Overall, the development of the alarm system was successful in part. While some parts of the system worked as expected, other functions did not. While the drivers were successfully programmed and worked as expected, the implementation of the drivers in the main source file was not entirely successful. Some functions worked as expected such as detecting button inputs, reading and displaying the temperature and displaying the time and date. However, other methods did not work as expected, such as the date and time not being able to be properly changed, or certain variables could be altered, but not in the way that was intended. Overall this was partly down to time management issues, not enough time was allocated to combining the code written by both authors and fix any bugs and issues that writing the complete system would bring. Time was also not set aside for uncontrollable delays, such as board malfunctions which did occur and slow development down. This meant that not every function was fully implemented in the alarm system. Also, another factor was memory optimisation. Due to how the main source file was written, including how each function was programmed most of the memory was used. This meant that code has to be altered to make sure there was enough space for more code, this was partly why every function could not be implemented. Communication between partners was frequent, meaning that both partner had a good idea of what was being done. Roles for the project were decided, and both partners were able to communicate and collaborate remotely via online messaging and online code repositories.

In the future, better time management between group members must be developed to complete projects more efficiently. Setting aside extra time for unforeseen circumstances and problems should be done so that there is time to keep the project on track. A better understanding of memory management is also needed, learning how to make code more robust in order to have enough space to fit the entire system will be important for future projects. Communication is an important part of the project development; the project would have been less successful had it not been for the good communication between partners and continuing this will be useful for the future.

Individual Contribution

Author 1 created the thermometer driver and the clock driver. Author 1 also created the temperature and clock output methods. Author 2 created the button and LCD driver. Author 2 also created the LCD message output methods. Both authors worked together to create the functions of the alarm system in the main source file.

**Appendix A**



A class diagram of the system.