COMP9032 Project Report

Author: Gary (Xinhui) Wang ZID: z5318369

Project Development

Project definition and hardware implementation

This project aims to design the software that controls the opacity level of the windows on an airplane for both passengers and flight crew. The passengers should be able to adjust their windows through two keys and the flight crew should be able to adopt central control to overwrite the passenger's control and enter emergency state.

Due to the limitation of the resources, we use LED light bar to simulate the window and the brightness of the LEDs represents the opacity. The keypad is used to simulate the input request to control the windows states for both passengers and flight crew. The push button is used to enter emergency state. The LCD screen should be able to display the current state of the windows.

The group members' contribution is roughly the same for all members.

Project Design

Hardware and interfacing

1. LED light bar

   Due to the lack of resources, the LED light bar is used to simulate the window. We use two led light bulb to represent a window. The brightness of the LED is used to indicate the opacity level of the window. The bottom four led light will flash when the push button PB0 is pushed which simulate the emergency happening.

2. LCD screen

   The LCD screen is used to display the current status of the windows we are controlling. An example display is shown below:

   ```
   State:          W1          W2
                   0           0
   ```

   W1 and W2 represents window 1 and window 2 respectively. The table 1 below indicates the opacity of the window and the corresponding brightness of the LED bar.

	Opacity level
0	Clear – maximum brightness
1	Light opaque – lightly dimmer
2	Medium opaque – medium brightness
3	Dark – LED off

Table 1 Brightness level and corresponding opacity level

There are four states for the simulation:

- *Initial state (S)*

  This is the beginning of the simulation and the windows will be set to clear.
- *Local control (L)*

  Local control means the window can be adjusted individually by passengers to any value
- *Central control (C)*

  Central control means the flight crew member can now control all the windows and passengers cannot control. In this state, the window can only be set to either clear (0) or dark (3)
- *Emergency (!!!)*

  All the windows will be set to clear (0) and the bottom 4 LED light will be flashing. Neither passengers nor the crew member can adjust the window.

3. Keypad and pushbutton

   The keypad and push button are used to control the opacity of different windows. Table 2 below shows the function of the keys and push button.

Key	Function
‘1’ and ‘4’	Window one local control ‘1’ – increase the opacity level ‘4’ – decrease the opacity level
‘2’ and ‘5’	Window two local control ‘2’ – increase the opacity level ‘5’ – decrease the opacity level
‘3’ and ‘6’	Central Control ‘3’ – set opacity level to dark ‘6’ –set opacity level to clear
‘A’	Return from central control to local
Push button (PB0)	Enter emergency state

Table 2 Input request simulation by keypad and push button

Software code and execution flow

The software structure is roughly shown in Fig 1. The code can be divided into four sections.

1. *Initialization*

   The initialization part includes defining **interrupt** and **timer** and initializing **ports**, **registers**, and **constants**. We use interrupt **EXT_INT0** to handle the press of the push button. We use the 8-bit **Timer0** to control the flash of LED light every half second when entering the emergency state. We also use the **Time5** to generate the PWM signal so we can control the brightness of the LED light. We also initialize the LCD screen in this stage before using it.

At the first stage, all the windows are set to clear, and the four LED are lighting up with maximum brightness. Then, we enter the **main** section to start the keypad scan.

2. *Keypad scan*

The keypad scanning process is inside the **main** and it will continue in a forever **loop**. Every time the passenger or a crew member pressed the keys, the corresponding function will be serviced.

If the central control related key is pressed, the code will start a lock to prevent the program from operating the local control related key function. Similarly, before the central control, it will check the emergency state lock to maintain the priority of different states.

3. *LED and LCD adjusting*

When a key or a push button is pressed, it will enter its corresponding section to adjust the LCD display and generate the PWM signals based on the value of OCR5A and OCR5B. The PWM signal controlled by OCR5A will change the brightness of window 1 and OCR5B will affect window 2. When this process is finished, it will return to the key scan section and wait for the next key press.

4. *Interrupt*

The emergency state can happen at any time during the code. Therefore, we need an interrupt to handle the situation. When entering the interrupt, we will first check the register used to mark emergency state.

If we have already entered the emergency state, we can deactivate it by turning off the flashing LED, disabling the Timer0 used to flash the LED and releasing the lock so that central control and local control can perform.

If we are not in emergency state, we activate it. We will start the overflow bit of the Timer0 so that we can flash the bottom 4-LED every half second (The timer interrupt will trigger every time it counts to maximum). We then adjusting the brightness and LCD screen accordingly and set a lock to prevent central and local control.

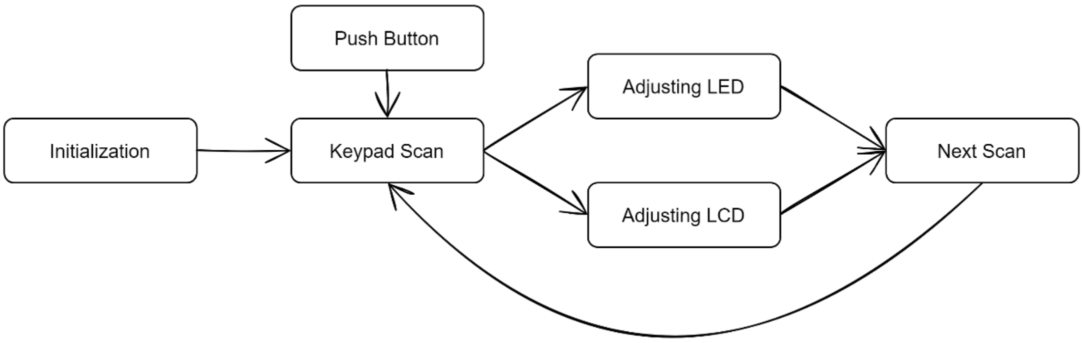

Fig 1 Software structures flow diagram

Hardware and Software interaction

1. LED light and PWM signal generation

The LED light bar is connected to port B 3-0 and port K 8-11. Port B controls the LED that represents the windows, and the Port K controls the 4 LEDs that flash when emergency. The PWM signal OC5A and OC5B is output from Port L pin 3 and pin 4 to Port G pin 1 and pin 3.

Based on the value receive in port G, we set the value of corresponding pins in port B so that the PWM signal generated is transmitted to the LED port. The flashing of the LED is so quick that the human eyes can only get an average of the brightness. By changing the OCR5A and OCR5B value, we can change the duty cycle of the PWM signal.

2. Push Button and Keypad

The keypad is connected to port C and the port direction is set as 1111 0000 and the key press is recognized through the column and row mask to see which column and which row is pressed. Then, based on the column value and row value, we can decide which key is pressed.

We use Push Button to start the interrupt. The interrupt will be run on falling edge detection since when the push button is pressed, the output from PB0 is low and therefore trigger the falling edge detection.

We need to deal with the button bouncing problem, for keypad, we add delays. The push button is done using both delay and an extra register 'indicator'. If the interrupt is triggered, the register will be checked to see if it is triggered by button bouncing. The register will be flipped when the interrupt finishes.

3. LCD screen

The LCD data pins are connected to Port F and the LCD control pins are connected to Port A. We use macro "`do_lcd_command`" to adjust the LCD such as clear display or position of the display. We use "`do_lcd_value`" to control what content we want to display on the screen.

Conclusion and evaluation

Overall, the software is able to deliver all the requirements given in the project specification except one. The program fails to take multiple command at the same time. If two passengers want to adjust the window at the same time or the passengers and flight crew wants to adjust the window at the same time, the software will fail to accomplish the input given in parallel.

The solution to improve that would be modifications to adjust the keypad scan process. Right now, the program will service one input request as soon as it is recognized in the keypad scan process and therefore, before the request is fulfilled, other requests won't be recognized. Therefore, the keypad scan process should scan all column and all rows to collect all the request and then start responding to them.