COMP9032 Project Design Manual

Smart Airplane Window Controller

Contents

1. Project Definition

This project simulates a window shade control system, where having 4 Windows and each window has 4 opaque levels, representing the clear (0), light opaque (1), medium opaque (2), and dark (3) respectively. Besides, the 4 kinds of window control modes are represented on the LCD display.

There are 2 representations of opaque levels, one is LCD to display the value of the level and another is LEDs dimming to indicate the opaque levels.

The pressing of keypads and push button are able to change the control modes among Initial (S), Local (L), Central (C) and Emergency (!!).

In summary, there are three kinds of input as below:

(1) The combination of two keys represent for one local window control, one for increasing the opaque level and one for decreasing and there are 4 groups of combination for four windows.

(2) Two keys to represent a central control, one for setting all windows to clear and one for setting all to dark.

(3) Two push buttons are used to trigger the emergency mode, i.e. setting all windows to clear.

And there are two kinds of output as the following:

(1) Four LEDs pairs are used to indicate the opaque level for the four windows.

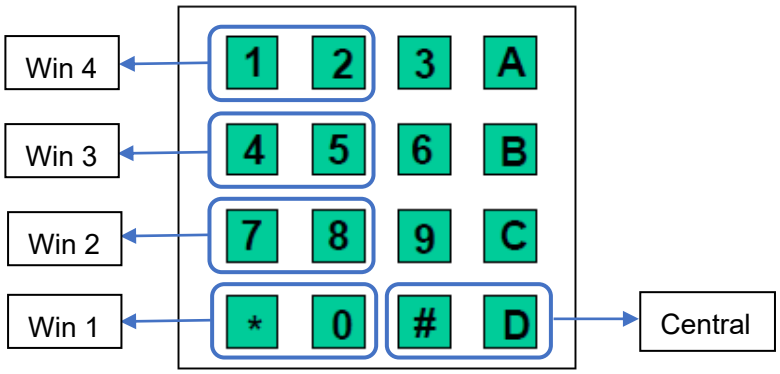
(2) Apart from using the LED indicator, LCD is used to provide textual information. It consists of two parts: the left part shows the mode of simulation and the right part shows the opaque level of each window.

2. Project Specification

2.1 Operating Requirement

2.1.1 Inputs

(1) The keypad layout and 10 keys function on it as below:



(2) Apart from the keypad, two push buttons are used for emergency control.

The table for all inputs is shown as below:

Key	Function	Key	Function
1	Win4 Up	2	Win4 Down
4	Win3 Up	5	Win3 Down
7	Win2 Up	8	Win2 Down
*	Win1 Up	0	Win1 Down
#	Central Dark	D	Central Clear
PB0	Emergency Clear	PB1	Emergency Clear

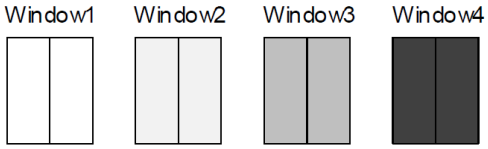
Key pairs (1, 2), (4, 5), (7, 8), (*, 0) are for window 4, 3, 2, 1 to increase or decrease opaque level.

Key pair (#, D) is for central control to set all windows dark or clear.

PB0 and PB1 are both for emergency to set all windows clear.

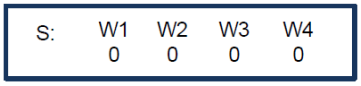
2.1.2 Outputs

(1) LED pairs are used to represent windows in the simulation, the dimming level of LEDs represents the opaque level of the windows. The figure shown as below.

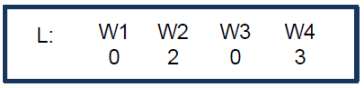


(2) LCD is used to display the control mode and the opaque level of each window. The figure shown as below.

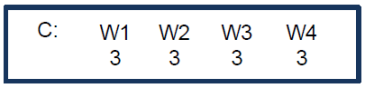
(a) Initial state, all windows are set to clear.



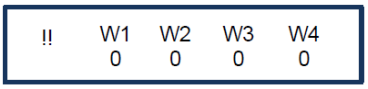
(b) Local control state, where W2 and W4 opaque levels are set to “2” and “3” respectively.



(c) Central control state, where all windows are set to dark.



(d) Emergency state, all windows are set to clear.



2.2 Hardware Design

Hardware: AVR lab board (ATmega2560) powered by 5V USB charger or DC power supply.

To connect I/O interfaces, jumper wires are used as following Pinout table, all pins are labeled on the AVR lab board.

Port A for LCD control and PORT C for LCD data, PORT D for push buttons INT0/1, PORT E & H for LEDs dimming control and PORT L for keypad inputs.

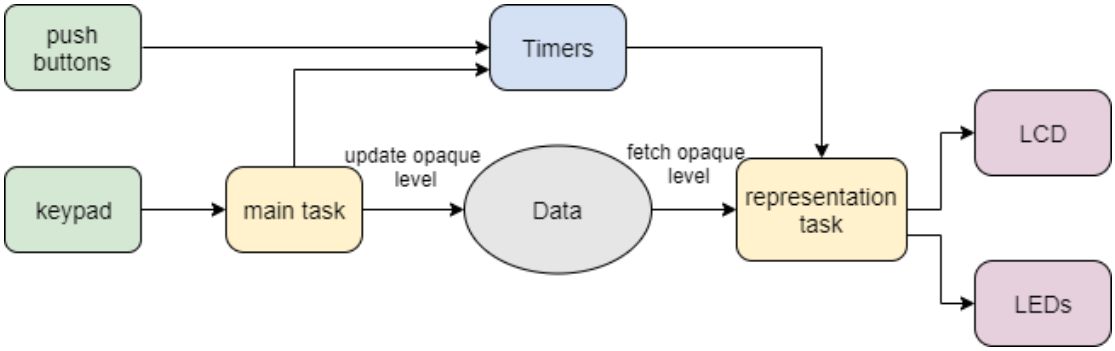
Port Group	Pin	Port Group	Pin	Function
PORT A	PA4	LCD CTRL	BE	LCD busy flag
PORT A	PA5	LCD CTRL	RW	LCD Read/Write
PORT A	PA6	LCD CTRL	E	LCD Enable
PORT A	PA7	LCD CTRL	RS	LCD Command/Data
PORT C	PC0-PC7	LCD DATA	D0-D7	LCD data out
PORT D	PD0-PD1	INPUTS	PB0-PB1	Push buttons for INT0/1
PORT E	PE3	LED BAR	LED0-LED1	PWM control LEDs dimming
PORT E	PE4	LED BAR	LED2-LED3	PWM control LEDs dimming
PORT E	PE5	LED BAR	LED4-LED5	PWM control LEDs dimming
PORT H	PH5	LED BAR	LED6-LED7	PWM control LEDs dimming
PORT L	PL0-PL3	KEYPAD	R0-R3	Keypad control for row 0-3
PORT L	PL4-PL7	KEYPAD	C0-C3	Keypad control for col 0-3

2.3 Software Design

IDE: Atmel Studio 7.0, Arduino. Atmel studio to build the machine code (hex file) and Arduino to download the program (hex file) into flash.

2.3.1 Overview Block Diagram

Below diagram shows the design of input/output interaction through the communication between main task and representation task and data storing/fetching to achieve the connection between the two tasks.

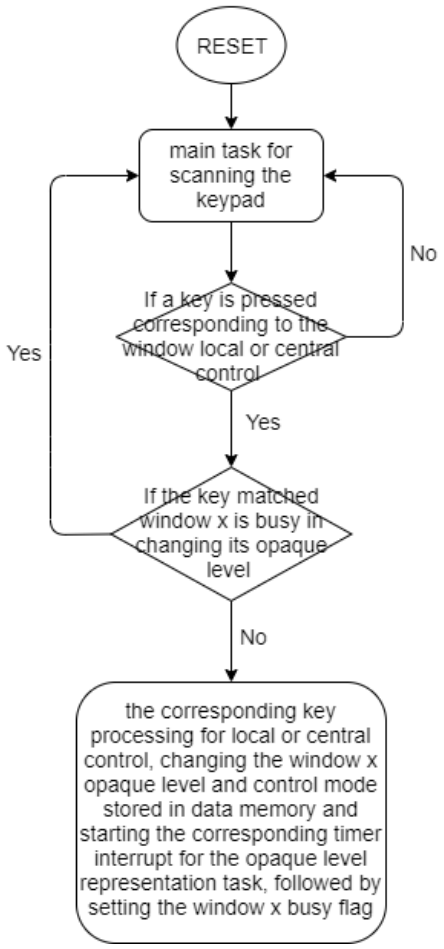


2.3.2 Flow Chart

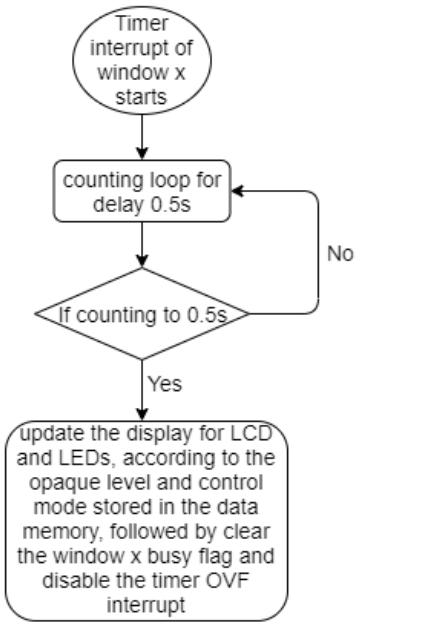
The flow chart for the window controller system divided into two parts, one for background task scanning the keypad pressing, one for timing task to simulate 0.5 second delay for updating the window opaque level.

The task for simulating delay of updating the opaque level of window x will only start when the corresponding key is pressed, and the window x busy flag is clear. The design allows multiple keys pressing to start multiple tasks for updating the opaque level of each window and its control mode. In order to achieve this, there are 4 individual tasks for updating the representation of each window. The flow chart for main task and the task for update each window opaque level and control mode representation is shown as below. The representation task may be further divided into 4 individual tasks for 4 different windows.

main task for scanning the keypad

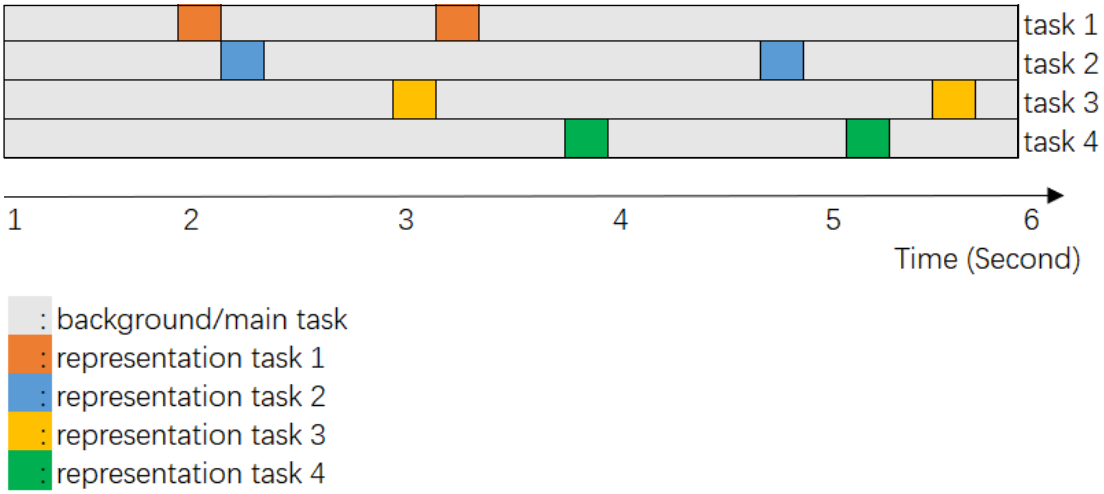


task for window x opaque level representation



2.3.3 Timing Schedule

The timing schedule is figured as below, background task represents for the main task for scanning the keypad and representation task 1, 2, 3, 4 is for updating the display of window 1, 2, 3, 4 opaque level and control mode respectively. For example, multiple inputs for window 1 and 2 may occur at time slot [1, 2] since the representation task for window 2 is followed by window 1 during the next time slot. Task 3 and 4 may be for the multiple inputs during time slot [2, 3] and followed by performing task 4. Time slot [4, 5] may only have task 2 being performed and during [5, 6] control for window 3 is done after window 4.



2.3.4 Data Structure

The data memory is used for storing/fetching the data, the structure of data model divide into three parts:

(1) 4 bytes unsigned integer array for storing the opaque level of each window.

(2) 2 bytes for storing the control mode.

(3) 2 bytes counters for 4 timers counting 0.5s delay to perform the representation tasks.

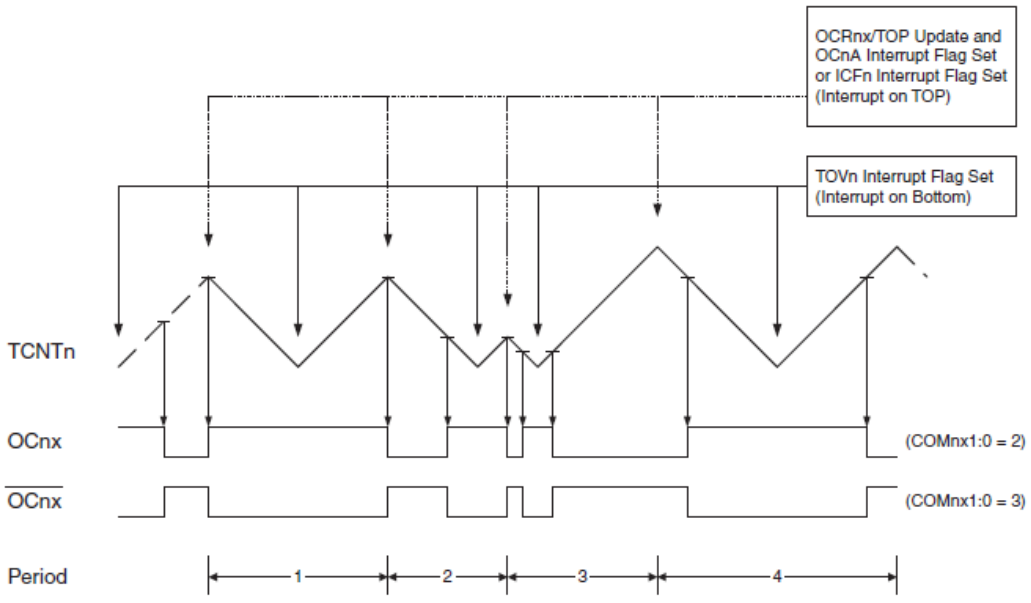
The data structure defined in the data segment shown as below:

```
; Define variables
.dseg
.org 0x200
Windows:
    .byte WIN_NUM
Mode:
    .byte 2
Counter1:
    .byte 2
Counter2:
    .byte 2
Counter3:
    .byte 2
Counter4:
    .byte 2
```

2.3.5 PWM for LEDs dimming control

This section is to illustrate technic specification for generating the PWM waveform adjusting the LEDs dimming. There are 4 groups of LEDs need dimming adjustment. Therefore, timer 3 and 4 are set to Phase correct PWM mode with compare mode in COMxn1, i.e. clear when up-counting match, set when down-counting match. OC3A, OC3B, OC3C and OC4C are outputs for PWM generation. The timing diagram in Phase Correct PWM mode is shown as below. The timer is either used for setting the duty cycle to adjust the LEDs dimming or generating timer OVF interrupt for fixed timing schedule. In this project, timer 1, 3, 4, 5 are used for timing schedule for 4 windows respectively and timer 3 and 4 are also used for LEDs dimming control (PWM).

Figure 17-8. Phase Correct PWM Mode, Timing Diagram



2.3.6 Program Structure

The program is structured as below:

; description

; include files

; define constants, variables and macros

; set up interrupt vector table

; reset and other interrupt subroutines

; INT0 / INT1:

; Timer1, 3, 4, 5 OVF:

; main

; end of main

; normal functions

3. Project testing / verification

In the end, according to the project specifications and requirements to test the functions, result for testing: program runs stable without critical bugs and all functions work as specified with supporting multiple inputs to control different windows. Inputs (keypad and push buttons) work as normal; outputs (LCD and LEDs display as normal); data representation correct and the delay time for representation is accurate.