Project Manual

Author: Haowen Guo

zID: z5251113

Project Specification

The Hardware is:

Arduino mega2560

Software:

Windows 10 1903

Atmel Studio 7.0

Arduino 1.0.6

Operating specification:

There are three controlling situations: individual control, central control, and automatic emergency response. Among the three controlling situations, the emergency response has a highest priority, followed by the central control; the individual control can be overwritten by any of the other two control operations.

Specifically,on emergency, all windows will be immediately set to clear; otherwise o if there is a central operation, all windows will be either set to clear or dark; else o an individual window can be adjusted anytime by the local passenger to one of the four states and the local controls are allowed to be done in parallel.

Input:

The local and central controls can be done by using keys on the keypad.

- For each window, two keys are used, one for increasing the window opaque level and one for decreasing the level.

Local control	**window 1**	**window 2**	**window 3**	**window 4**
increasing	key 2	key 4	key 6	key 8
decreasing	key 3	key 5	key 7	key 9

- The central control only uses two keys, one for setting all windows to clear and one for setting all windows to dark.

Central control	**all clear**	**all dark**
Operation	key A	key B

- Emergency is simulated by using a push button. When the button is pushed, an emergency happens and all windows are set to clear.

Emergency control	all clear
Operation	PUSHBUTTON 0

Output:

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

Apart from using the LED indicator, LCD is used to provide textual information about the simulation. The LCD display consists of two parts: the left part shows the state of the simulation and the right part shows the opaque level of each window. The simulation can have four states:

- Initial state (S:), simulation starts and all windows are set to clear
- Local control (L:), windows are individually controlled
- Central control (C:), all windows can only be set either to clear or to dark.
- Emergency (!!), all windows are set to clear

Hardware design and implementation

- Using LCD to display the status of the brightness of each windows and current controlling status.
- Using LED to display the brightness of each windows
- Inputs: Encoding input keys of the keyboard and push button 0 to represent all possible input requests

Software design

This project is in four sections:

1. Receive input from keypad and pushbutton
2. Check if there is a input triggering interrupt
3. Displays on LEDS via PWM controlling
4. Displays on LCD about the information of current status

Section 1

Receiving keypad input function can receive all the input from keypad, so in project, it will only process the pre-defined input. Pre-defined input as shown above.

Configuration of keypad, using port K to get input

Double loop for getting input.

```
RESET:
    ldi temp1,PORTKDIR      ; columns are outputs, rows are inputs
    sts DDRK, temp1
    ...
    ...
main:
    ldi cmask, INITCOLMASK  ; initial column mask
    clr col                 ; initial column
    ...
    ...
```

Following codes are just used for getting input.

Section 2

According to the Specification, Emergency has the highest priority and Central control has the second highest priority, so interrupt can be used here.

When pushing button 0, program will jump into a interrupt 0 subroutine.

When pushing key A or B, program will jump into a interrupt 1 subroutine.

Variables which are used by the interrupt routines and need to be carefully accessed outside of the interrupt routines.

```
;address in memory
    jmp RESET
.org INT0addr
    jmp EXT_INT0
.org INT1addr
    jmp EXT_INT1

RESET:
    ...
    ...
    ldi temp,(2<<ISC00)|(2<<ISC10)      ;set INTO and INT1 as falling edge
triggered interrupts
    sts EICRA, temp

    in temp,EIMSK
    ori temp,(1<<INT0)|(1<<INT1)
    out EIMSK,temp

    sei                                 ; set global interrupt
    jmp main
```

External Interrupt subroutines:

```
EXT_INT0:
    push temp           ;save register
    in temp, SREG       ;save SREG
    push temp
    ...                 ;set all clear
    ...
    pop temp            ;restore SREG
    out SREG,temp
    pop temp            ;restore register
    reti

EXT_INT1:
    push temp
    in temp, SREG
    push temp
    do_lcd_command 0b00000001 ; clear display
    cpi temp2,1
    breq cdark
    jmp cclear

cdark:
```

```
    ;set all windows dark
    ...                 ; set all dark
    ...
    jmp int1end

cclear:
    ;set all windows clear
    ...                 ;set all clear
    ...

int1end:
    sbi PORTD,1         ;set bit for INT1
    pop temp
    out SREG,temp
    pop temp
    reti
```

Section 3:

Because this project will adjust the brightness of LEDS, so PWM will be used to implement this function.

A `Macro` called `displayLocal` is used to change the PWM signal.

```
.macro displayLocal
    cpi @0,0
    breq play0
    cpi @0,1
    breq play1
    cpi @0,2
    breq play2
    cpi @0,3
    breq play3
    jmp main
play0:
    ldi temp1,0x00      ; brightness level 0 clear
    rjmp endm
play1:
    ldi temp1,0x55      ; brightness level 1
    rjmp endm
play2:
    ldi temp1,0xAA      ; brightness level 2
    rjmp endm
play3:
    ldi temp1,0xFF      ; brightness level 3
    rjmp endm
endm:
.endmacro
```

Configuration of PWM:

```
    ldi temp1, 0b00111000
    sts DDRL,temp1          ;port L bit3 4 5 are used to output PWM
    ldi temp1, 0b00001000
    out DDRE, temp1         ;port E bit 3 is used to output PWM

    clr temp1               ;at first, initializing as all clear
```

```
    sts OCR5AH,temp1
    sts OCR5BH,temp1
    sts OCR5CH,temp1
    sts OCR3AH,temp1

    ldi temp1,0x00
    sts OCR5AL,temp1
    sts OCR5BL,temp1
    sts OCR5CL,temp1
    sts OCR3AL,temp1

    ldi temp1,(1<<CS50)     ;configuration of OCR5A, OCR5B, OCR5C and OCR3A
    sts TCCR5B,temp1
    ldi temp1,(1<<CS30)
    sts TCCR3B,temp1
    ldi temp1,(1<<WGM50)|(1<<COM5A1)|(1<<COM5B1)|(1<<COM5C1)    ;
    sts TCCR5A,temp1

    ldi temp1,(1<<WGM30)|(1<<COM3A1)
    sts TCCR3A,temp1
```

Section 4:

Displays on LCD about the brightness and status information of windows.

Configuration of LCD is as normal as the lab before, so here only mention one part.

Because the displays on LCD are two line and each column needs to match each other, here using

`do_lcd_command 0b10101011`

to move cursor to the third bit on the second line.

One typical example used in Emergency control:

```
    do_lcd_command 0b00000001 ; clear display

    ldi temp1,0x00          ; set all clear
    sts OCR5AL,temp1
    sts OCR5BL,temp1
    sts OCR5CL,temp1
    sts OCR3AL,temp1
    do_lcd_data '!'
    do_lcd_data '!'
    do_lcd_data ' '
    do_lcd_data 'W'
    do_lcd_data '1'
    do_lcd_data ' '
    do_lcd_data 'W'
    do_lcd_data '2'
    do_lcd_data ' '
    do_lcd_data 'W'
    do_lcd_data '3'
    do_lcd_data ' '
    do_lcd_data 'W'
    do_lcd_data '4'
    do_lcd_command 0b10101011   ;move cursor to the third bit on the second line
    do_lcd_data '0'
```

```
    do_lcd_data ' '
    do_lcd_data ' '
    do_lcd_data '0'
    do_lcd_data ' '
    do_lcd_data ' '
    do_lcd_data '0'
    do_lcd_data ' '
    do_lcd_data ' '
    do_lcd_data '0'
```