**Design of project**

Group3, 21T3, COMP9032

**Hardware components**

LEDs:

* Window1: LED0-LED1
  + Pins connection
    - PL3(OC5A, 16bits) to LED0
    - PL4(OC5B, 16bits) to LED1
* Window2: LED2-LED3
  + Pins connection
    - PE3(OC3A, 16bits) to LED0
    - PE5(OC3C, 16bits) to LED1
* Emergency flashes: LED6-LED9
  + Pins connection
    - PG0-PG4 to LED6-LED9

Keypad

* Individual control
  + Key 1: window 1 decreasing
  + Key 4: window 1 increasing
  + Key 2: window 2 decreasing
  + Key 5: window 2 increasing
    - decreasing: window brighter = LED darker
    - increasing: window darker = LED brighter
* Central control
  + A: set to 0-clear
  + B: set to 3-dark
  + C: exit central control. Window back to local state
* Pins connection
  + PC0 To C3
  + PC1 To C2
  + PC2 To C1
  + PC3 To C0
  + PC4 To R3
  + PC5 To R2
  + PC6 To R1
  + PC7 To R0

Push button

* PB1 for Emergency control
* Use interrupt
* Pins connection
  + PD0(INT0) to PB1

LCD

* Display state information
* Pins connection
  + PF0-PF7 to D0-D7
  + PA5 to RW
  + PA6 to E
  + PA7 to RS

**Software components**

All software components communicate with each other through 3 context registers:

|  |  |  |
| --- | --- | --- |
| **Register** | **Explanation** | **Values** |
| control\_state = r0 | the state of simulation | 0: Initial state(S)  1: Local control (L)  2: Central control(C)  3: Emergency (!!!) |
| win1\_level = r2 | window 1 opacity level | 0: clear  1: light opaque  2: medium opaque  3: dark |
| win2\_level = r3 | window 2 opacity level | 0: clear  1: light opaque  2: medium opaque  3: dark |

**Window brightness control component**

Requirement:

Set all windows (LEDs) brightness through those 3 context variables. Notice the priority should be implemented through register control\_state.

Interface definition:

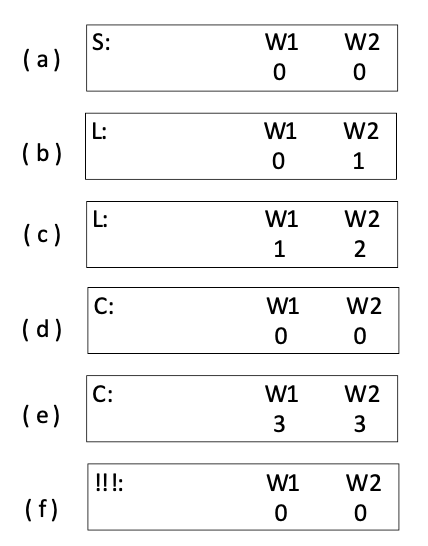
; set the brightness of all windows  
; Please flow the priority requirement through control\_state. 3:Emergency > 2:Central control > 1:Local control  
; input:  
 ; control\_state ; 0:Initial state(S); 1:Local control (L); 2:Central control(C); 3:Emergency (!!!)  
 ; win1\_level ; window 1 opacity level. levels: 0-clear; 1-light opaque; 2-medium opaque; 3-dark  
 ; win2\_level ; window 2 opacity level.

set\_windows\_brightness:  
  
 ret

**LCD display simulation states component**

Requirement:

LCD display current simulation states through those 3 context registers.

Example:  


Interface definition:  
  
; LCD display states. Display current control\_state, win1\_level, and win2\_level on LCD  
; input:  
 ; control\_state ; 0:Initial state(S); 1:Local control (L); 2:Central control(C); 3:Emergency (!!!)  
 ; win1\_level ; window 1 opacity level. levels: 0-clear; 1-light opaque; 2-medium opaque; 3-dark  
 ; win2\_level ; window 2 opacity level.

LCD\_display\_states:  
  
 ret

**Keypad component**

Requirement:

* All key push actions should follow the priority requirement.
* Key 1:
  + condition: control\_state = 0 or 1
  + decreasing the window1 opacity level
  + decreasing win1\_level, bottom to 0
  + set control\_state to 1
* Key 4:
  + condition: control\_state = 0 or 1
  + increasing the window1 opacity level
  + increasing win1\_level, up to 3
  + set control\_state to 1
* Key 2:
  + condition: control\_state = 0 or 1
  + decreasing the window2 opacity level
  + decreasing win2\_level, bottom to 0
  + set control\_state to 1
* Key 5:
  + condition: control\_state = 0 or 1
  + increasing the window2 opacity level
  + increasing win2\_level, up to 3
  + set control\_state to 1
* Key A:
  + condition: control\_state = 0 or 1 or 2
  + setting all windows to clear
    - set win1\_level and win2\_level to 0
  + set control\_state to Central control (2)
* Key B:
  + condition: control\_state = 0 or 1 or 2
  + setting all windows to dark
    - set win1\_level and win2\_level to 3
  + set control\_state to Central control (2)
* Key C:
  + condition: control\_state = 2
  + Exit central control
  + set control\_state to Local control (1)
* After setting the context registers should invoke set\_windows\_brightness and LCD\_display\_states

**Emergency button component**

Requirement:

* Pushing the button will trigger the emergency state
  + set control\_state to 3
* Pushing the button again will dismiss the emergency and all windows will be **set back to their previous state**.

Logic:

If (control\_state != 3)

control\_state = 3

call set\_windows\_brightness

call LCD\_display\_states

call LED\_flash

Else

control\_state = 0

call set\_windows\_brightness

call LCD\_display\_states

Notice within LED\_flash is an infinite loop, so the push button should through interrupt to set control\_state back to 0 and the LED\_flash will check the control\_state to break out the loop.

**Convention**

Please use registers defined in Project-variable-convention.asm. You can take variable\_a and variable\_b as temp variables. If you want to use additional registers, please let Xiaohang Hu knows.

You can find some common functions which has been used in lab3 in Project-base.asm. Such as lcd\_init, sleep\_n\_ms, etc.