Theme Report - Observe

Junbo Wang - wangj430 - 400249823

In the 2DX4 course, it is organized around three themes: observe, reason, and act. For the first three weeks, the theme is called "observe," and we are making observations about the intelligent system. We learnt about the digital signals, analog to digital which correspond to voltage and code, and system clock and timing. We use these in decoding the assembly code in order to operate the hardware device. I will show some examples from labs to demonstrate this theme.

In lab 2 milestone 1, I built an active low circuit and connected it to the microcontroller's input buttons PM0, PM1, and PM2. On the microcontroller attached, the load button is on PM3. To decode the combinational lock, I used the combinational code 0001. When I executed the code, LED D1 turns on; when we hit the buttons pin 1, pin 2, and pin 3 simultaneously, LED D1 turns off, and LED D2 turns on. **[1]**

For the debugging, I started by changing the addresses of Port M and Port N. Then, I set up the RCGCGPIO, DIR, and DEN of Port N to 0x1000, 0x3 and 0x3 and Port M to 0x800, 0x0, and 0xF. Due to Port N being the input, I set DEN and DIR points to both LEDs. Also, I set the initial state to locked which points to D1. Finally, I created a loop and used a masking method to AND my combinational code 0001 with 1111. This could check whether the result is locked or unlocked after the AND operation. **[2]**

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;ADDRESS DEFINTIONS

;The EQU directive gives a symbolic name to a numeric constant, a register-relative value or a program-relative value


SYSCTL_RCGCGPIO_R            EQU 0x400FE608  ;General-Purpose Input/Output Run Mode Clock Gating Control Register (RCGCGPIO Register)
GPIO_PORTN_DIR_R             EQU 0x40064400  ;GPIO Port N Direction Register address
GPIO_PORTN_DEN_R             EQU 0x4006451C  ;GPIO Port N Digital Enable Register address
GPIO_PORTN_DATA_R            EQU 0x400643FC  ;GPIO Port N Data Register address
```

```
GPIO_PORTM_DIR_R             EQU 0x40063400  ;GPIO Port M Direction Register
Address (Fill in these addresses)
GPIO_PORTM_DEN_R             EQU 0x4006351C  ;GPIO Port M Direction Register
Address (Fill in these addresses)
GPIO_PORTM_DATA_R            EQU 0x400633FC  ;GPIO Port M Data Register Address
(Fill in these addresses)

COMBINATION EQU 2_0001  ;password
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;Do not alter this section

        AREA    |.text|, CODE, READONLY, ALIGN=2 ;code in flash ROM
        THUMB                                     ;specifies using Thumb
instructions
        EXPORT Start

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;Function PortN_Init
PortN_Init
    ;STEP 1
    LDR R1, =SYSCTL_RCGCGPIO_R
    LDR R0, [R1]
    ORR R0,R0, #0x1000
    STR R0, [R1]
    NOP
    NOP

    ;STEP 5
    LDR R1, =GPIO_PORTN_DIR_R
    LDR R0, [R1]
    ORR R0,R0, #0x3
    STR R0, [R1]

    ;STEP 7
    LDR R1, =GPIO_PORTN_DEN_R
    LDR R0, [R1]
    ORR R0, R0, #0x3
    STR R0, [R1]
    BX  LR

PortM_Init
    ;STEP 1
    LDR R1, =SYSCTL_RCGCGPIO_R
    LDR R0, [R1]
    ORR R0,R0, #0x800
    STR R0, [R1]
    NOP
    NOP

    ;STEP 5
    LDR R1, =GPIO_PORTM_DIR_R   ;/*direction*/
    LDR R0, [R1]
```

```
    AND R0,R0, #0x00
    STR R0, [R1]

    ;STEP 7
    LDR R1, =GPIO_PORTM_DEN_R
    LDR R0, [R1]
    ORR R0, R0, #0xF

    STR R0, [R1]
    BX  LR


State_Init LDR R5,=GPIO_PORTN_DATA_R  ;Locked is the Initial State
           MOV R4,#2_00000010
           STR R4,[R5]
           BX LR

Start
    BL  PortN_Init
    BL  PortM_Init
    BL  State_Init
    LDR R0, =GPIO_PORTM_DATA_R  ; Inputs set pointer to the input
    LDR R3, =COMBINATION         ;R3 stores our combination

Loop
            LDR R1,[R0]
            AND R2,R1,#2_00001111
            CMP R2,R3
            BEQ Unlocked_State
            BNE Locked_State
Locked_State
    LDR R5,=GPIO_PORTN_DATA_R
    MOV R4,#2_00000010
    STR R4,[R5]
    B Loop

Unlocked_State
    LDR R5, =GPIO_PORTN_DATA_R
    MOV R4,#2_00000001
    STR R4, [R5]
    B Loop
    ALIGN

END
[3]
```

The theme for the first three weeks is called "observe", I think that this theme lets me observe different outputs through changing the inputs. In this milestone, I choose the digital lock and use the knowledge from analog signal conditioning and introduction of combinational and sequential lock to complete the task. After I finish the milestone, I observe that the output of the LED is controlled by input buttons and assembly codes. With different inputs, the output on the board will perform differently.

Through this milestone, I understand this theme, observe. The observations on the output let me know how the inputs and assembly codes operate. With this understanding of each theme, I know how the intelligent system and microcontroller work and can perform the operations in future projects.

References:

[1] Junbo Wang & Yichen Lu. “Lab_02_wangj430_luy191”. Observation and conclusion, pp3, Feb. 2022

[2] Junbo Wang & Yichen Lu. “Lab_02_wangj430_luy191”. Method, pp2, Feb. 2022

[3] Junbo Wang & Yichen Lu. “Lab_02_wangj430_luy191”. Code Appendix, pp4-6, Feb. 2022