

MA4832 Microprocessor Systems

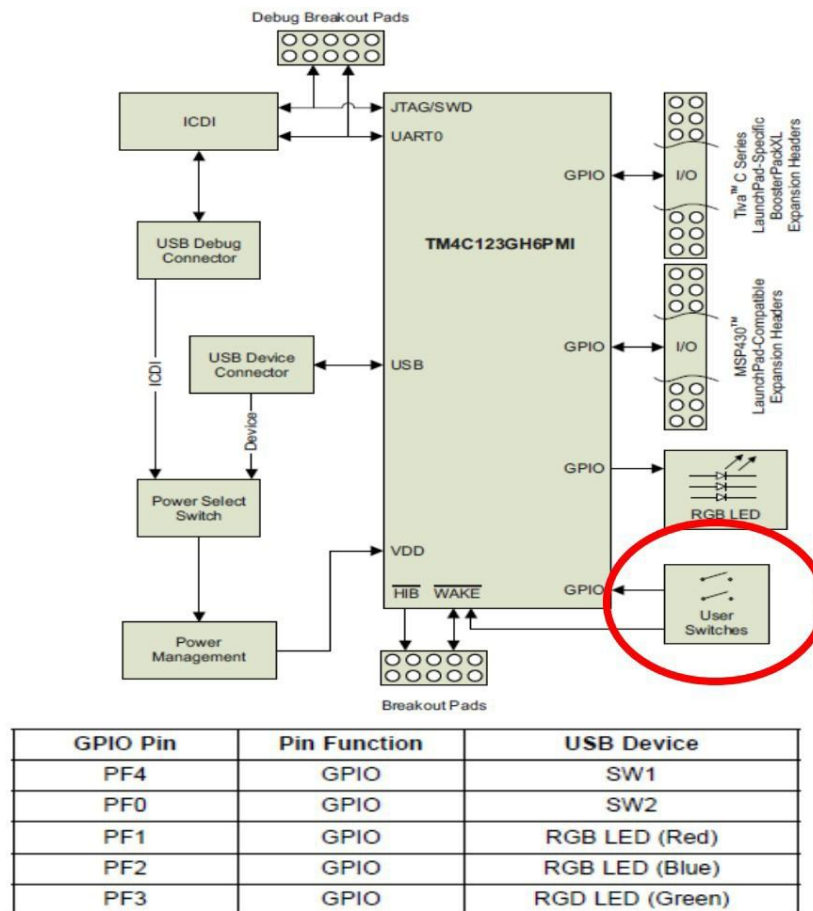
Lab 2 Exercise – Hardware – Pushbutton SW 1 and RGB LED

You will learn in this session:

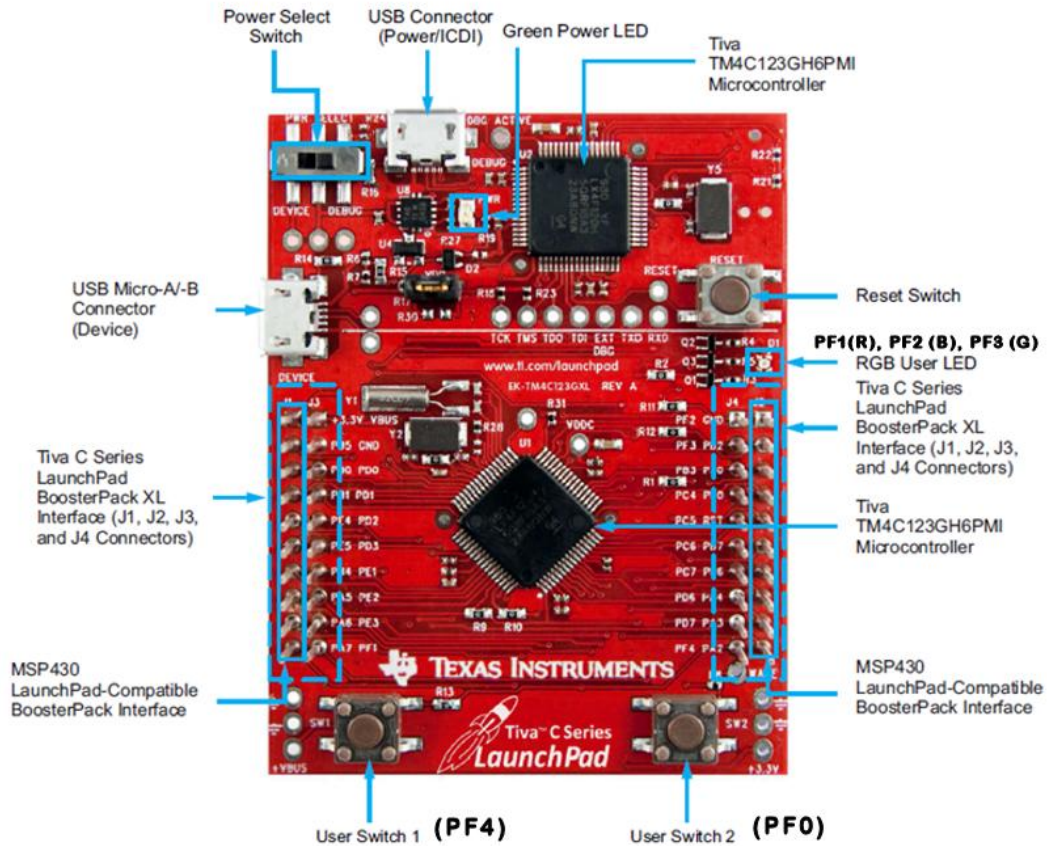
- How to read digital signal from input device through GPIO port F (read the Status of SW1 on PF4)
- How to send digital signal to output device through GPIO port F (turn on LED, Red (PF1), Blue (PF2) and Green (PF3))

Hardware Connection:

Please follow the diagram below to do the hardware connection among TM4C123GH6P, three LEDs and two switches. Once done, please approach the lab Technician for verification.



Tiva C Series TM4C123G LaunchPad



1. Program to read the status of SW1 and turn on and off the LED (white)

- Program: main.s

```
GPIO_PORTF_DATA_R EQU 0x400253FC      ;      set bit to 1 for Bits 9:2
GPIO_PORTF_DIR_R   EQU 0x40025400
GPIO_PORTF_AFSEL_R EQU 0x40025420
GPIO_PORTF_PUR_R   EQU 0x40025510
GPIO_PORTF_DEN_R   EQU 0x4002551C
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R  EQU 0x4002552C
PF0                EQU 0x40025004      ;      SW2 - negative logic
PF1                EQU 0x40025008      ;      RED LED
PF2                EQU 0x40025010      ;      BLUE LED - ORIG
PF3                EQU 0x40025020      ;      GREEN LED
PF4                EQU 0x40025040      ;      SW1 - ORIG -negative logic
PFA                EQU 0x40025038      ;      All 3 colours (RGB)- white
SYSCTL_RCGCGPIO_R EQU 0x400FE608      ;      Register to enable port F .p340

        AREA    |.text|, CODE, READONLY, ALIGN=2
        THUMB
        EXPORT  Start

Start

; initialize PF 1-3 output, PF4 an input,
; enable digital I/O, ensure alt. functions off.
; Input: none, Output: none, Modifies: R0, R1

        ; activate clock for Port F
        LDR R1, =SYSCTL_RCGCGPIO_R
        LDR R0, [R1]
        ORR R0, R0, #0x20                ; set bit 5 to turn on clock
        STR R0, [R1]
        NOP                               ; allow time for clock to finish
        NOP
        NOP

; no need to unlock PF2

        ; disable analog functionality
        LDR R1, =GPIO_PORTF_AMSEL_R
        LDR R0, [R1]
        BIC R0, #0x0E                    ; 0 means analog is off
        STR R0, [R1]

        ;configure as GPIO
        LDR R1, =GPIO_PORTF_PCTL_R
        LDR R0, [R1]
        BIC R0, R0, #0x0000FF0          ; Clears bit 1 & 2 (to ensure default GPIO func selected)
        BIC R0, R0, #0x000FF000         ; Clears bit 3 & 4 (to ensure default GPIO func selected)
        STR R0, [R1]

        ;set direction register
        LDR R1, =GPIO_PORTF_DIR_R
        LDR R0, [R1]
        ORR R0, R0, #0x0E                ; PF 1,2,3 output (1 in output)
        BIC R0, R0, #0x10                ; Make PF4 built-in button input (0 is output)
        STR R0, [R1]

        ; regular port function
        LDR R1, =GPIO_PORTF_AFSEL_R
        LDR R0, [R1]
        BIC R0, R0, #0x1E                ; 0 means disable alternate function
        STR R0, [R1]

        ; pull-up resistors on switch pins
```

```

    LDR R1, =GPIO_PORTF_PUR_R      ; R1 = &GPIO_PORTF_PUR_R
    LDR R0, [R1]                   ; R0 = [R1]
    ORR R0, R0, #0x10               ; R0 = R0|0x10 (enable pull-up on PF4)
    STR R0, [R1]                   ; [R1] = R0

    ; enable digital port
    LDR R1, =GPIO_PORTF_DEN_R      ; 7) enable Port F digital port
    LDR R0, [R1]
    ORR R0, #0x0E                   ; 1 means enable digital I/O
    ORR R0, R0, #0x10               ; R0 = R0|0x10 (enable digital I/O on PF4)
    STR R0, [R1]

    LDR R4, =PF4                   ; R4 = &PF4

loop                                ; in this loop, the appliance (PF2) toggles when the switch
                                   ; is released

    BL SSR_On
waitforpress1                       ; proceed only when the button is pressed
    LDR R0, [R4]                   ; R0 = [R4] (read status of PF4)
    CMP R0, #0x10                  ; R0 == 0x10?
    BEQ waitforpress1              ; if so, spin
waitforrelease1                     ; proceed only when the button is released
    LDR R0, [R4]                   ; R0 = [R4] (read status of PF4)
    CMP R0, #0x10                  ; R0 != 0x10?
    BNE waitforrelease1            ; if so, spin
    BL SSR_Off
waitforpress2                       ; proceed only when the button is pressed
    LDR R0, [R4]                   ; R0 = [R4] (read status of PF4)
    CMP R0, #0x10                  ; R0 == 0x10?
    BEQ waitforpress2              ; if so, spin
waitforrelease2                     ; proceed only when the button is released
    LDR R0, [R4]                   ; R0 = [R4] (read status of PF4)
    CMP R0, #0x10                  ; R0 != 0x10?
    BNE waitforrelease2            ; if so, spin
    B    loop

;-----SSR_On-----
; Make PFA high.
; Input: none
; Output: none
; Modifies: R0, R1
SSR_On
    LDR R1, =PFA                   ; R1 = &PFA
    MOV R0, #0x0E                  ; R0 = 0x0E (turn on the appliance)
    STR R0, [R1]                   ; [R1] = R0, write to PFA
    BX LR                           ; return

;-----SSR_Off-----
; Make PFA low.
; Input: none
; Output: none
; Modifies: R0, R1
SSR_Off
    LDR R1, =PFA                   ; R1 = &PFA
    MOV R0, #0x00                  ; R0 = 0x00 (turn off the appliance)
    STR R0, [R1]                   ; [R1] = R0, write to PFA
    BX LR                           ; return

ALIGN                               ; make sure the end of this section is aligned
END                                 ; end of file

```

-
- Load and run the program.
 - Study the code and determine;
 - (i) How status of SW1 (PF4) is read when it is pressed and released.
 - (ii) How the LED (White) is generated and then turned off.

Exercise:

Modify the code such that with each successive press and release of switch SW1

- The TM4C123G LED will perform the following operation.

Press SW1 - Red Led

Press SW1 – Off

Press SW1 – Blue Led

Press SW1 – Off

Press SW1 - Green Led

Press SW1 – Off

Press SW1 - White Led

Press SW1 – Off

Repeat.