

Pseudo Code

Initialize PortF

Load PortF Data Address

Load PortF Data

Isolate Input Pins

Isolate SW1

Move SW1 bit to common position

Isolate SW2

Move SW2 bit to common position

Check if either is on

Move Result to LED

Insert Result

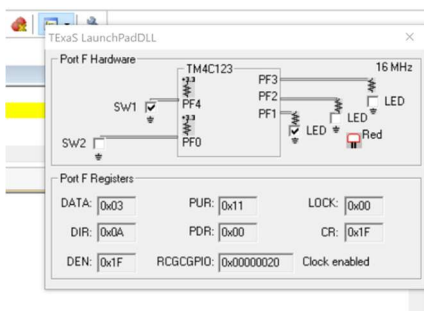
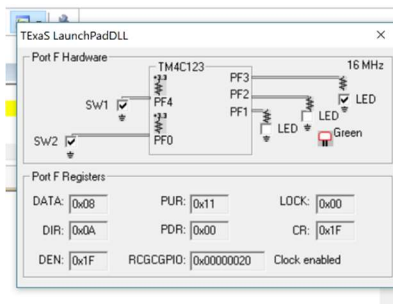
Move to other LED

Insert Result

Reload Data Register Address

Toggle LED's accordingly

Loop Back



```

1  ;***** main.s *****
2  ; Program written by: Dylan Cauwels
3  ; Date Created: 1/20/2017
4  ; Last Modified: 1/24/2017
5  ; Brief description of the program
6  ; The objective of this system is to implement a Car door signal system
7  ; Hardware connections: Inputs are negative logic; output is positive logic
8  ; PF0 is right-door input sensor (1 means door is open, 0 means door is closed)
9  ; PF4 is left-door input sensor (1 means door is open, 0 means door is closed)
10 ; PF3 is Safe (Green) LED signal - ON when both doors are closed, otherwise OFF
11 ; PF1 is Unsafe (Red) LED signal - ON when either (or both) doors are open, otherwise OFF
12 ; The specific operation of this system
13 ; Turn Unsafe LED signal ON if any or both doors are open, otherwise turn the Safe LED signal ON
14 ; Only one of the two LEDs must be ON at any time.
15 ; NOTE: Do not use any conditional branches in your solution.
16 ; We want you to think of the solution in terms of logical and shift operations
17
18 GPIO_PORTF_DATA_R EQU 0x400253FC
19 GPIO_PORTF_DIR_R EQU 0x40025400
20 GPIO_PORTF_AFSEL_R EQU 0x40025420
21 GPIO_PORTF_PUR_R EQU 0x40025510
22 GPIO_PORTF_DEN_R EQU 0x4002551C
23 GPIO_PORTF_LOCK_R EQU 0x40025520
24 GPIO_PORTF_CR_R EQU 0x40025524
25 GPIO_PORTF_AMSEL_R EQU 0x40025528
26 GPIO_PORTF_PCTL_R EQU 0x4002552C
27 GPIO_LOCK_KEY EQU 0x4C4F434B ; Unlocks the GPIO_CR register
28 SYSCTL_RCGCGPIO_R EQU 0x400FE608
29 THUMB
30 AREA DATA, ALIGN=2
31 ;global variables go here
32 ALIGN
33 AREA |.text|, CODE, READONLY, ALIGN=2
34 EXPORT Start
35
36 Start
37 LDR R1, =SYSCTL_RCGCGPIO_R ;Activate Port F Clock
38 LDR R0, [R1]
39 ORR R0, R0, #0x20
40 STR R0, [R1] ;Clock Initialize Time
41 NOP
42 NOP
43 LDR R1, =GPIO_PORTF_LOCK_R ;Unlock PortF Register
44 LDR R0, =0x4C4F434B
45 STR R0, [R1]
46 LDR R1, =GPIO_PORTF_CR_R
47 MOV R0, #0xFF
48 STR R0, [R1]
49 LDR R1, =GPIO_PORTF_AMSEL_R ;Disable Analog
50 MOV R0, #0
51 STR R0, [R1]
52 LDR R1, =GPIO_PORTF_PCTL_R ;Start GPIO
53 MOV R0, #0x00000000
54 STR R0, [R1]
55 LDR R1, =GPIO_PORTF_DIR_R ;Set Direction Register
56 MOV R0, #0x0A ;0 & 4 Input, 1-3 Output
57 STR R0, [R1]
58 LDR R1, =GPIO_PORTF_AFSEL_R ;Initialize PortF
59 MOV R0, #0
60 STR R0, [R1]
61 LDR R1, =GPIO_PORTF_PUR_R ;Pull Up Resistors
62 MOV R0, #0x11
63 STR R0, [R1]
64 LDR R1, =GPIO_PORTF_DEN_R ;Enable PortF
65 MOV R0, #0xFF
66 STR R0, [R1]
67
68 loop ;PortF_Input
69 LDR R1, =GPIO_PORTF_DATA_R ;Load PortF Data Address
70 LDR R0, [R1] ;Load PortF Data
71 AND R0, R0, #0x11 ;Isolate Input Pins
72 AND R1, R0, #0x01 ;Isolate SW1

```

```
73     LSL R1, R1, #1           ;Move SW1 bit to common position
74     AND R2, R0, #0x10       ;Isolate SW2
75     LSR R2, R2, #3           ;Move SW2 bit to common position
76     ORR R3, R2, R1           ;Check if either is on
77     MOV R4, #2               ;Move Result to LED
78     EOR R5, R4, R3           ;Insert Result
79     LSL R5, R5, #2           ;Move to other LED
80     ADD R5, R5, R3           ;Insert Result
81     LDR R1, =GPIO_PORTF_DATA_R ;Reload Data Register Address
82     STR R5, [R1]             ;Toggle LED's accordingly
83
84     B     loop
85
86     ALIGN                    ; make sure the end of this section is aligned
87     END                      ; end of file
```

Initialize Machine



Load Port F Data



Isolate Switch Pins & Clear Them



Move Switch Bits to same position



Test if either is open (ORR)



move resulting bit to Red LED port



toggle the bit again & move it to green
LED port



Insert new values into Port F Data



Store _{new} Port F Data into Port F



loop