djs 3967

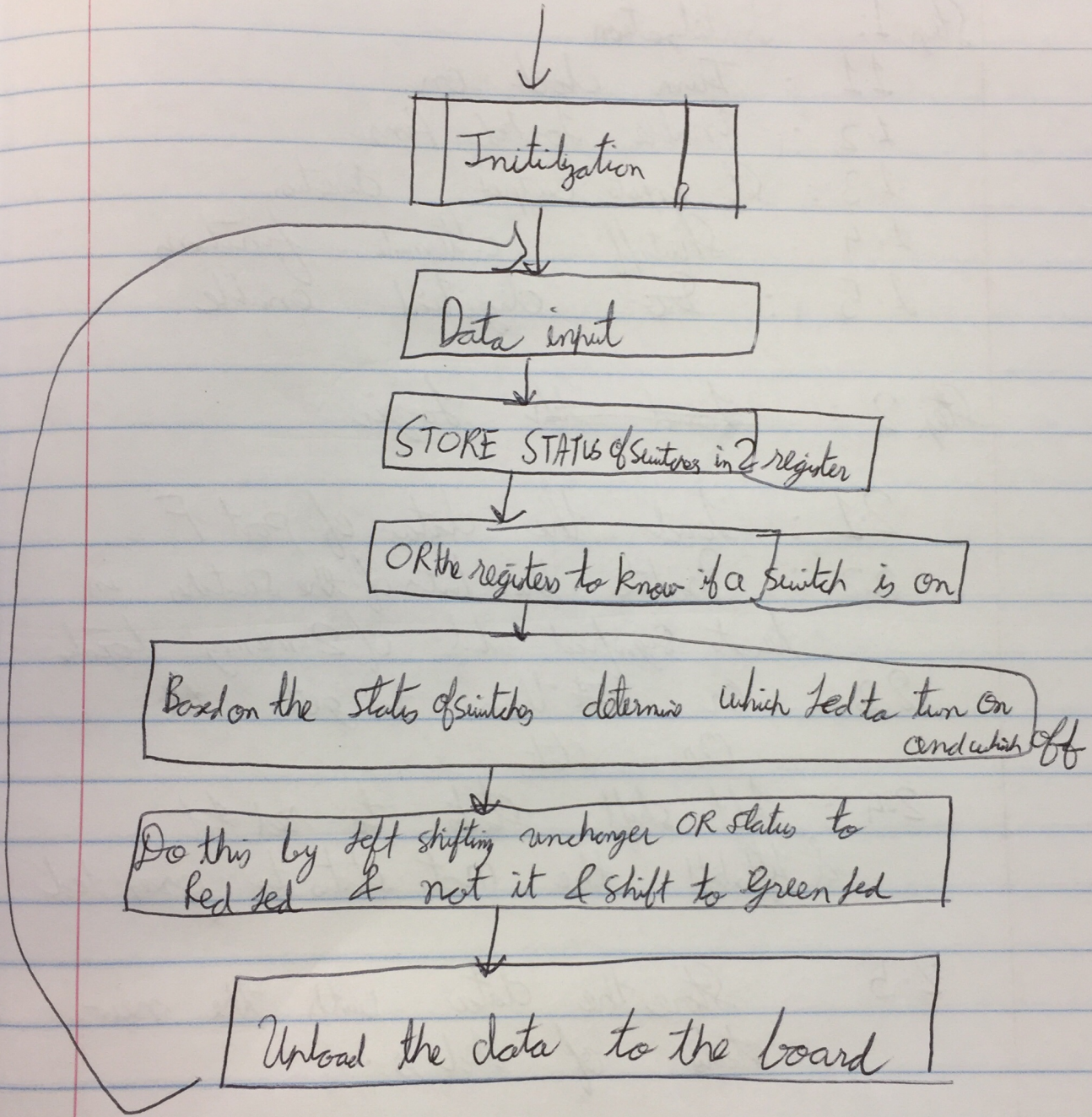
EE319K

1/24/16

Dhruv Sondeva

Lab 0

Flow Chart



Pseudo Code

Step 1: Initilization

1.1 : Turn clock on

1.2 : Enable locked pins

1.3 : Set input output direction

1.4 : Shutoff alternate functions

1.5 : Do digitial enable

Step 2 : ~~load th~~ Logic

2.1 : load the data of port F

2.2 : Isolate the bits of the Switches in least significat bit of 2 memory locats

2.3 : Or them to get the Or stats

2.4 : left shift stats to red led & leftshift the not stats to green led

2.5 : Store the data with the new stats of led's.

```
;****************** main.s ***************
; Program written by: Dhruv Sandesara
; Date Created: 1/15/2017
; Last Modified: 1/23/2017
; Brief description of the program
; The objective of this system is to implement a Car door signal system
; Hardware connections: Inputs are negative logic; output is positive logic
;  PF0 is right-door input sensor (1 means door is open, 0 means door is closed)
;  PF4 is left-door input sensor (1 means door is open, 0 means door is closed)
;  PF3 is Safe (Green) LED signal - ON when both doors are closed, otherwise OFF
;  PF1 is Unsafe (Red) LED signal - ON when either (or both) doors are open, otherwise OFF
; The specific operation of this system
;   Turn Unsafe LED signal ON if any or both doors are open, otherwise turn the Safe LED signal ON
;   Only one of the two LEDs must be ON at any time.
; NOTE: Do not use any conditional branches in your solution.
;       We want you to think of the solution in terms of logical and shift operations

GPIO_PORTF_DATA_R  EQU 0x400253FC
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_LOCK_R  EQU 0x40025520
GPIO_PORTF_CR_R    EQU 0x40025524
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R  EQU 0x4002552C
GPIO_LOCK_KEY      EQU 0x4C4F434B  ; Unlocks the GPIO_CR register
SYSCTL_RCGCGPIO_R  EQU 0x400FE608
      THUMB
      AREA    DATA, ALIGN=2
;global variables go here
      ALIGN
      AREA    |.text|, CODE, READONLY, ALIGN=2
      EXPORT  Start
Start
    LDR R1,=SYSCTL_RCGCGPIO_R; TURN CLOCK ON
    LDR R0,[R1]
    ORR R0,R0,#0X20
    STR R0,[R1]

    NOP
    NOP

    LDR R1,=GPIO_PORTF_LOCK_R; ENABLE THE LOCKED PINS
    LDR R0,=0X4C4F434B
    STR R0,[R1]
    LDR R1, =GPIO_PORTF_CR_R
    MOV R0, #0XFF
    STR R0,[R1];

    LDR R1,=GPIO_PORTF_DIR_R; DIRECTIONS OF SWITCHES
    LDR R0,[R1];
    AND R0,#0X0E;
    ORR R0,#0XA;
    STR R0,[R1];


    LDR R1,=GPIO_PORTF_AFSEL_R; ALTERNATE FUNCTION SHUTOFF
    LDR R0,[R1];
    AND R0,#0XE4;
    STR R0,[R1];

    LDR R1,=GPIO_PORTF_DEN_R; DIGITAL ENABLE
    LDR R0,[R1]
    ORR R0,#0X1B
    STR R0,[R1]

loop
    LDR R1,=GPIO_PORTF_DATA_R;
    LDR R0, [R1];

    AND R2,R2,#0;
```

```
73        ADD R2,R2,R0; R2 IS FOR THE RIGHT SWITCH
74        AND R2,R2,#1;
75
76        AND R3,R3,#0;
77        ADD R3,R3,R0; R3 IS FOR THE LEFT SHIFT
78        AND R3,R3,#0X10;
79        LSR R3,R3,#4;
80
81        AND R4,R4,#0; R4 IS FOR THE FINAL VALUE WETHER ANY OF THE SWITCHES ARE ON
82        ORR R4,R2,R3;  1 MEANS A DOOR IS ON AND 0 IS A DOOR IS OFF
83
84        AND R5,R5,#0;
85        LSL R5,R4,#1; FOR THE RED LED
86
87        EOR R4,R4,#1; NOT
88        LSL R4,R4,#3; STATUS OF GREEN LED
89
90        ADD R5,R5,R4; COMBINING THE BOTH VALUES
91        AND R0,R0,#0X4
92
93        ADD R0,R0,R5
94        STR R0,[R1];
95
96
97
98
99
100
101
102
103       B   loop
104
105         ALIGN        ; make sure the end of this section is aligned
106         END          ; end of file
```

C:\Keil\EE319Kware\Lab0_EE319K\Lab0.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Core	
R0	0x00000012
R1	0x400253FC
R2	0x00000012
R3	0x00000001
R4	0x00000000
R5	0x00000002
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x000002CC
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	286578276
Sec	17.91114225
FPU	

Project Registers

Disassembly

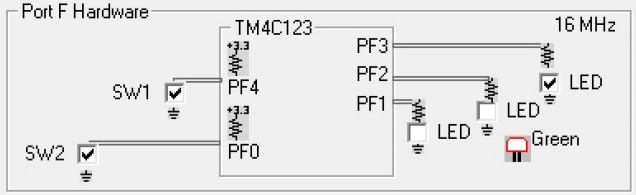
```
    74:         AND R2,R2,#1;
    75:
0x000002CC F0020201  AND          r2,r2,#0x01
    76:         AND R3,R3,#0;
```

main.s Startup.s

```
67
68  loop
69      LDR R1,=GPIO_PORTF_DATA_R;
70      LDR R0, [R1];
71
72      AND R2,R2,#0;
73      ADD R2,R2,R0; R2 IS FOR THE RIGHT SWITCH
74      AND R2,R2,#1;
75
76      AND R3,R3,#0;
77      ADD R3,R3,R0; R3 IS FOR THE LEFT SHIFT
78      AND R3,R3,#0X10;
79      LSR R3,R3,#4;
80
81      AND R4,R4,#0; R4 IS FOR THE FINAL VALUE WETHER ANY OF THE SWITCH
82      ORR R4,R2,R3;  1 MEANS A DOOR IS ON AND 0 IS A DOOR IS OFF
83
84      AND R5,R5,#0;
85      LSL R5,R4,#1; FOR THE RED LED
86
87      EOR R4,R4,#1; NOT
88      LSL R4,R4,#3; STATUS OF GREEN LED
89
```

TExaS LaunchPadDLL

Port F Hardware



Port F Registers

DATA:	0x08	PUR:	0x00	LOCK:	0x00
DIR:	0x0A	PDR:	0x00	CR:	0x1F
DEN:	0x1B	RCGCGPIO:	0x00000020	Clock enabled	

Command

```
*** Currently used: 804 Bytes (2%)
```

>

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess

Call Stack + Locals

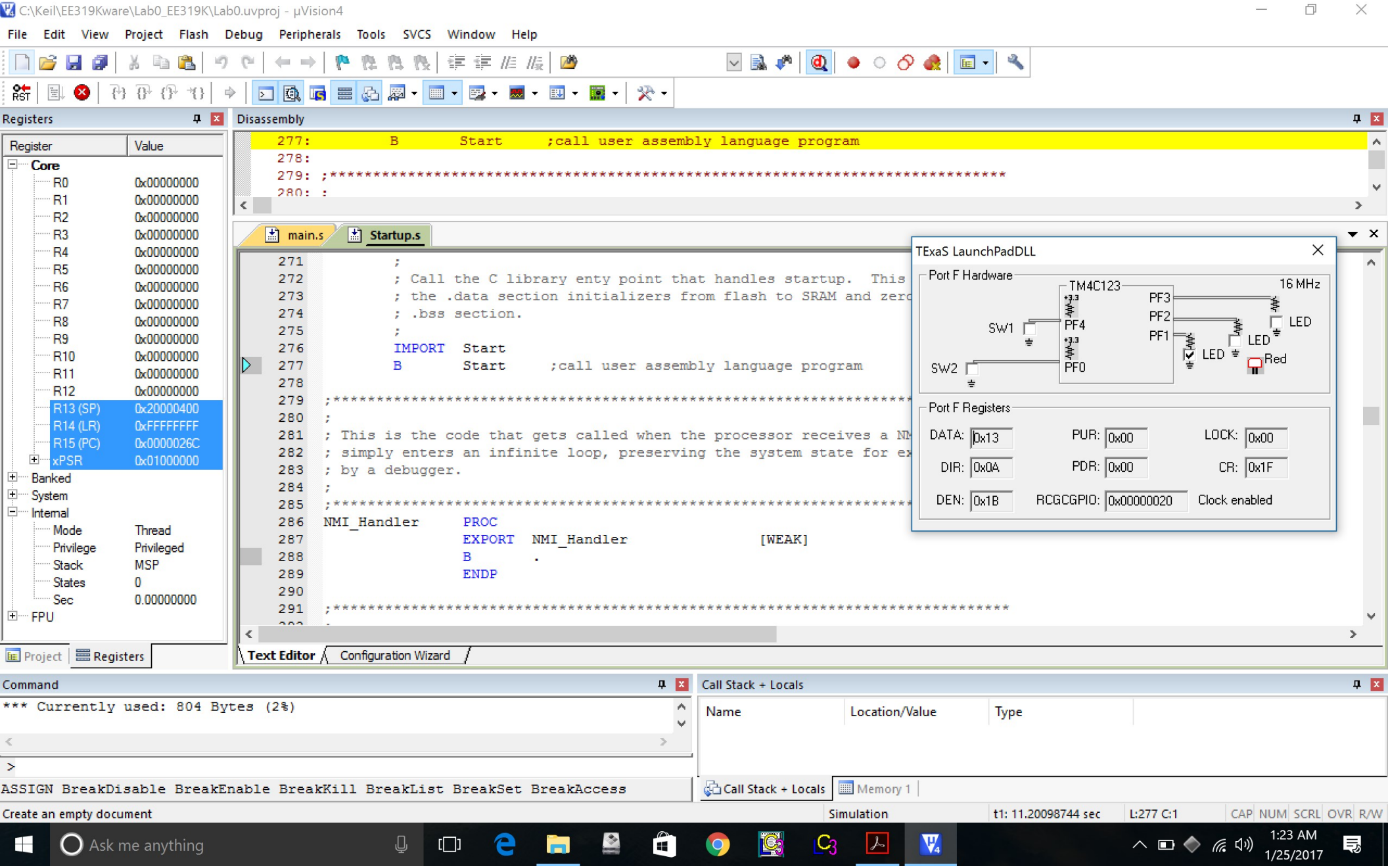
Name	Location/Value	Type

Call Stack + Locals | Memory 1

Simulation | t1: 20.17256556 sec | L:74 C:1 | CAP NUM SCRL OVR R/W

Ask me anything

1:24 AM
1/25/2017


C:\Keil\EE319Kware\Lab0_EE319K\Lab0.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Core
R0 0x00000000
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x20000400
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x0000026C
xPSR 0x01000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 0
Sec 0.00000000
FPU
Project Registers
Disassembly
277: B Start ;call user assembly language program
278:
279: ;*****
main.s Startup.s
271 ;
272 ; Call the C library enty point that handles startup. This
273 ; the .data section initializers from flash to SRAM and zero
274 ; .bss section.
275 ;
276 IMPORT Start
277 B Start ;call user assembly language program
278
279 ;*****
280 ;
281 ; This is the code that gets called when the processor receives a NM
282 ; simply enters an infinite loop, preserving the system state for ex
283 ; by a debugger.
284 ;
285 ;*****
286 NMI_Handler PROC
287 EXPORT NMI_Handler [WEAK]
288 B .
289 ENDP
290
291 ;*****
Text Editor Configuration Wizard
TExaS LaunchPadDLL
Port F Hardware
TM4C123
16 MHz
SW1
SW2
PF4
PF0
PF3
PF2
PF1
LED
LED
LED
Red
Port F Registers
DATA: 0x13
PUR: 0x00
LOCK: 0x00
DIR: 0x0A
PDR: 0x00
CR: 0x1F
DEN: 0x1B
RCGCGPIO: 0x00000020
Clock enabled
Command
*** Currently used: 804 Bytes (2%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Name Location/Value Type
Call Stack + Locals Memory 1
Create an empty document
Simulation
t1: 11.20098744 sec
L:277 C:1
CAP NUM SCRL OVR R/W
Ask me anything
1:23 AM
1/25/2017