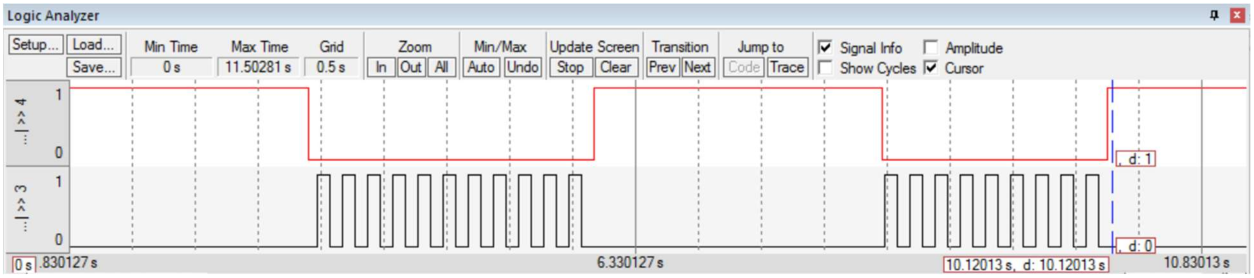
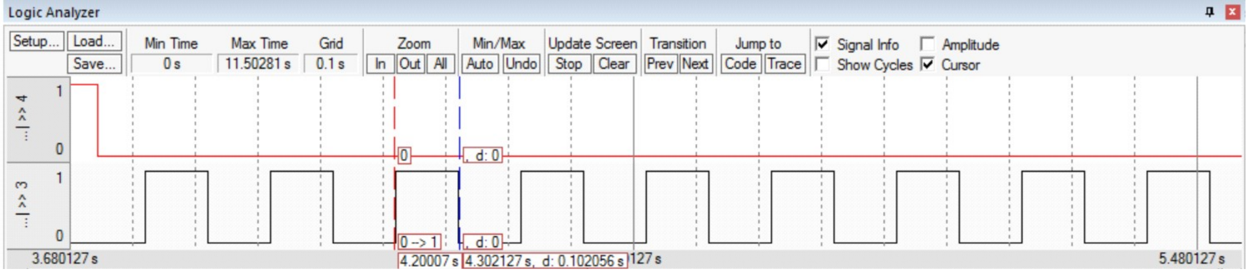
Lab 2

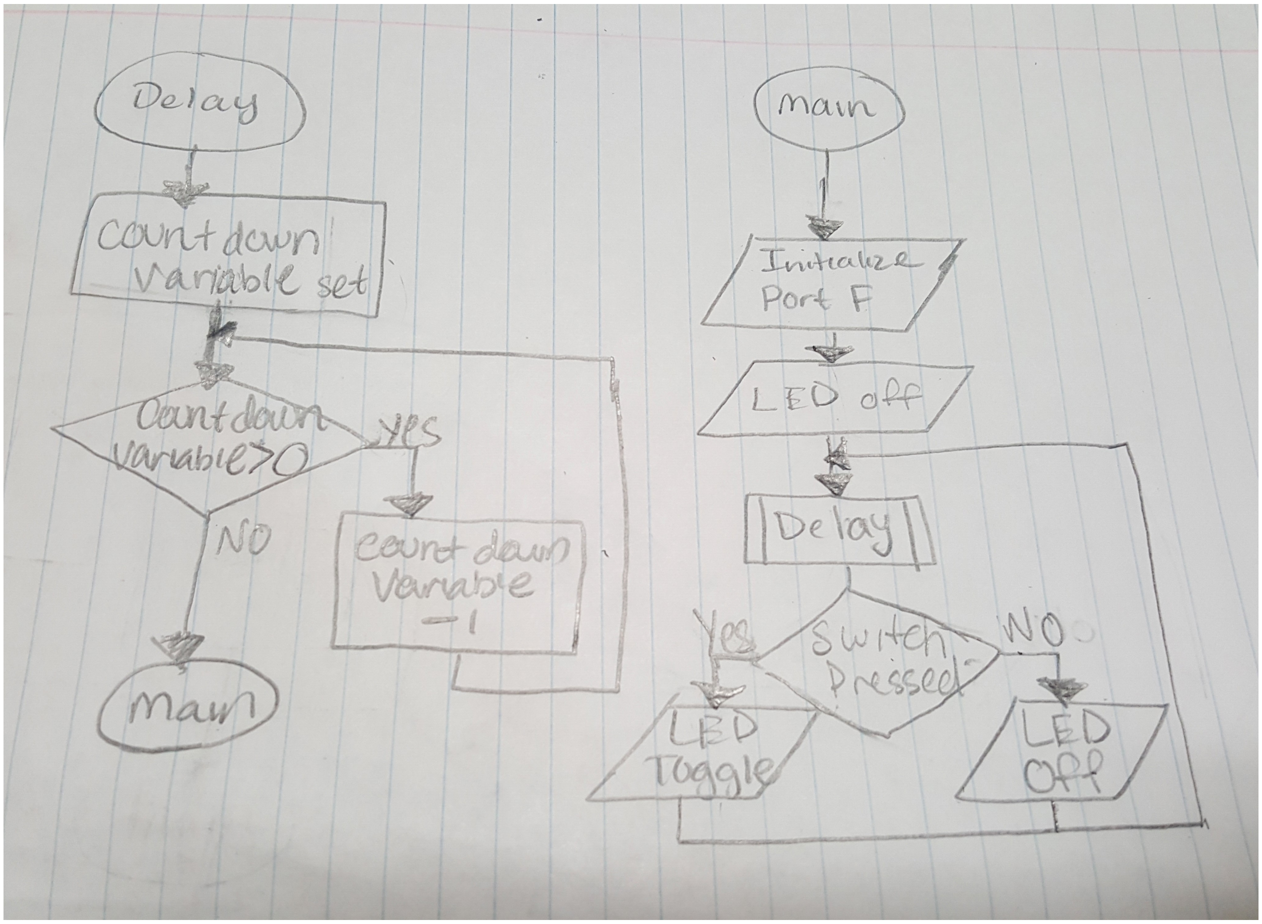
Chimezie Iwuanyanwu
Allen Pan

Screenshot





Flowchart



Pseudocode

```
start   Turn on the clock for Port F
            Set the Port F direction register so
                PF4 is an input and
                PF3 is an output
            Enable the PF4 and PF3 bits in the Port F DEN register
            Set bit 4 in Port F PUR register so it will run on the real board
        Set PF3 so the LED is OFF
loop    go to delay
        Read the switch and test if the switch is pressed
        If PF4=0 (the switch is pressed),
                toggle PF3 (flip bit from 0 to 1, or from 1 to 0)
        If PF4=1 (the switch is not pressed),
                clear PF3 so LED is OFF
        Go to loop

delay           Save register used for a delay
                Set delay register to for a 100ms delay
delay_loop      Return to main if delay register = 0
                Subtract delay register by 1
                Go to delay_loop
```

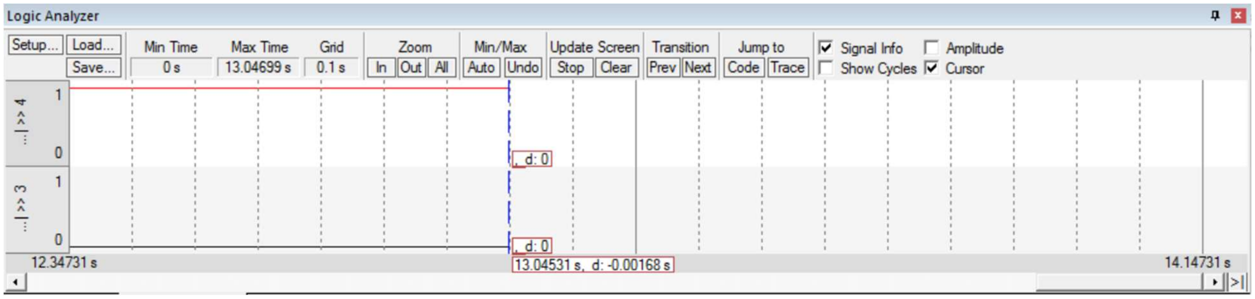
Assembly Code

```
;****************** main.s ***************
; Program written by: Chimezie Iwuanyanwu & Allen Pan
; Date Created: 1/22/2016
; Last Modified: 2/14/2016
; Section: Wednesday 5-6
; Instructor: Ramesh Yerraballi
; Lab number: 2
; Brief description of the program
; The overall objective of this system an interactive alarm
; Hardware connections
;  PF4 is switch input  (1 means SW1 is not pressed, 0 means SW1 is pressed)
;  PF3 is LED output (1 activates green LED)
; The specific operation of this system
;    1) Make PF3 an output and make PF4 an input (enable PUR for PF4).
;    2) The system starts with the LED OFF (make PF3 =0).
;    3) Delay for about 100 ms
;    4) If the switch is pressed (PF4 is 0), then toggle the LED once, else turn the LED OFF.
;    5) Repeat steps 3 and 4 over and over

GPIO_PORTF_DATA_R       EQU   0x400253FC
GPIO_PORTF_DIR_R        EQU   0x40025400;X
GPIO_PORTF_AFSEL_R      EQU   0x40025420;X
GPIO_PORTF_PUR_R        EQU   0x40025510;X
GPIO_PORTF_DEN_R        EQU   0x4002551C;X
GPIO_PORTF_AMSEL_R      EQU   0x40025528;X
GPIO_PORTF_PCTL_R       EQU   0x4002552C;X
SYSCTL_RCGCGPIO_R       EQU   0x400FE608;X
```

```
COUNT EQU 400000

     AREA    |.text|, CODE, READONLY, ALIGN=2
     THUMB
     EXPORT  Start
Start
              LDR R1, =SYSCTL_RCGCGPIO_R
              LDR R0, [R1]
              ORR R0, #0x20
              STR R0, [R1]
              NOP
              NOP

              LDR R1, =GPIO_PORTF_AMSEL_R
              LDR R0, [R1]
              AND R0, #0xE7
              STR R0, [R1]

              LDR R1, =GPIO_PORTF_PCTL_R
              LDR R0, [R1]
              AND R0, #0xE7
              STR R0, [R1]

              LDR R1, =GPIO_PORTF_DIR_R
              LDR R0, [R1]
              AND R0, #0xE7
              RR R0, #0x08
              STR R0, [R1]

              LDR R1, =GPIO_PORTF_AFSEL_R
              LDR R0, [R1]
              AND R0, #0xE7
              STR R0, [R1]

              LDR R1, =GPIO_PORTF_DEN_R
              LDR R0, [R1]
              ORR R0, #0x18
              STR R0, [R1]

              LDR R1, =GPIO_PORTF_PUR_R
              LDR R0, [R1]
              ORR R0, #0x10
              STR R0, [R1]

              B clear
loop
              BL delay

              LDR R1, =GPIO_PORTF_DATA_R
              LDR R0, [R1]

              AND R2, #0
              AND R2, R0, #0x10
```

```
        CMP R2, #0

        BNE clear

        EOR R3, R0, #0x08
        STR R3, [R1]

        B loop

clear
        LDR R1, =GPIO_PORTF_DATA_R
        LDR R0, [R1]
        AND R0, #0xF7
        STR R0, [R1]

        B loop

delay
        LDR R1, =COUNT
delay_loop
        SUBS R1, #1
        BNE delay_loop
        BX LR

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

Microcontroller vs. Real time


Logic Analyzer

Min Time 0 s
Max Time 13.04699 s
Grid 0.1 s

☑ Signal Info □ Amplitude
□ Show Cycles ☑ Cursor

12.34731 s
13.04531 s, d: -0.00168 s
14.14731 s

For every 10 secs in real time there is about 13 seconds in simulated time