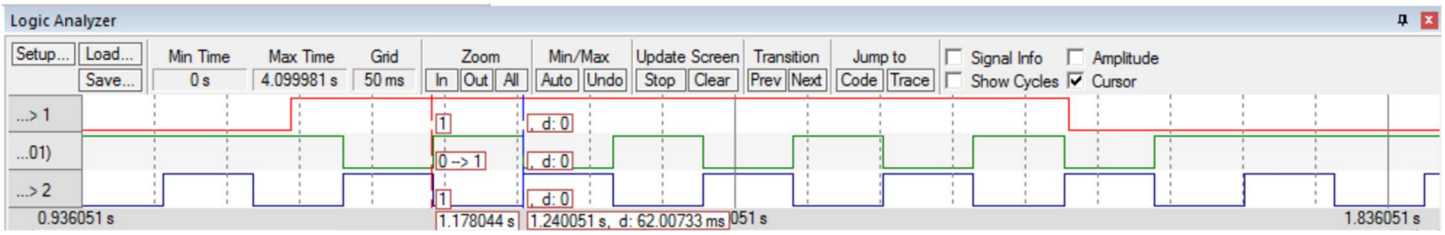
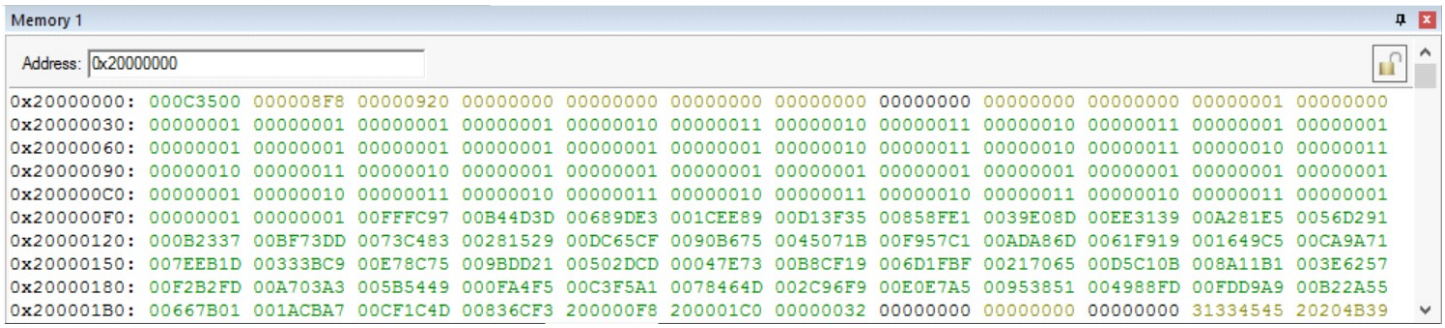
Screenshots





Assembly Code

```
;****************** main.s ***************
; Program written by: ***Your Names**update this***
; Date Created: 1/22/2016
; Last Modified: 1/22/2016
; Section ***Tuesday 1-2***update this***
; Instructor: ***Ramesh Yerraballi**update this***
; Lab number: 4
; Brief description of the program
;   If the switch is presses, the LED toggles at 8 Hz
; Hardware connections
;  PE1 is switch input  (1 means pressed, 0 means not pressed)
;  PE0 is LED output (1 activates external LED on protoboard)
;Overall functionality of this system is the similar to Lab 3, with three changes:
;1-  initialize SysTick with RELOAD 0x00FFFFFF
;2-  add a heartbeat to PF2 that toggles every time through loop
;3-  add debugging dump of input, output, and time
; Operation
;          1) Make PE0 an output and make PE1 an input.
;          2) The system starts with the LED on (make PE0 =1).
;   3) Wait about 62 ms
;   4) If the switch is pressed (PE1 is 1), then toggle the LED once, else turn the LED on.
;   5) Steps 3 and 4 are repeated over and over


SWITCH                  EQU 0x40024004  ;PE0
LED                     EQU 0x40024008          ;PE1
SYSCTL_RCGCGPIO_R       EQU 0x400FE608
SYSCTL_RCGC2_GPIOE      EQU 0x00000010
```

```
SYSCTL_RCGC2_GPIOF      EQU 0x00000020
GPIO_PORTE_DATA_R       EQU 0x400243FC
GPIO_PORTE_DIR_R        EQU 0x40024400;
GPIO_PORTE_AFSEL_R      EQU 0x40024420;
GPIO_PORTE_PUR_R        EQU 0x40024510
GPIO_PORTE_DEN_R        EQU 0x4002451C;
GPIO_PORTF_DATA_R       EQU 0x400253FC
GPIO_PORTF_DIR_R        EQU 0x40025400;
GPIO_PORTF_AFSEL_R      EQU 0x40025420;
GPIO_PORTF_DEN_R        EQU 0x4002551C;
NVIC_ST_CTRL_R          EQU 0xE000E010
NVIC_ST_RELOAD_R        EQU 0xE000E014
NVIC_ST_CURRENT_R       EQU 0xE000E018
SOFT_COUNT              EQU      1240000
HARD_COUNT              EQU 1653333
        THUMB
        AREA    DATA, ALIGN=4
        EXPORT DataBuffer
        EXPORT TimeBuffer
SIZE      EQU    50
        EXPORT DataPt [DATA,SIZE=4]
        EXPORT TimePt [DATA,SIZE=4]
DataBuffer SPACE  SIZE*4
TimeBuffer SPACE  SIZE*4
DataPt    SPACE  4
TimePt    SPACE  4
COUNT     SPACE  1


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

Start     BL TExaS_Init

          LDR R1, =SYSCTL_RCGCGPIO_R
          LDR R0, [R1]
          ORR R0, #0x30
          STR R0, [R1]
          NOP
          NOP


          LDR R1, =GPIO_PORTE_DIR_R
          LDR R0, [R1]
          AND R0, #0xFC

          ORR R0, #0x01
          STR R0, [R1]
```

```
        LDR R1, =GPIO_PORTE_AFSEL_R
        LDR R0, [R1]
        AND R0, #0xFC
        STR R0, [R1]

        LDR R1, =GPIO_PORTE_DEN_R
        LDR R0, [R1]
        ORR R0, #0x03
        STR R0, [R1]

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

        LDR R1, =GPIO_PORTF_DIR_R
        LDR R0, [R1]
        AND R0, #0xFB
        ORR R0, #0x04
        STR R0, [R1]

        LDR R1, =GPIO_PORTF_DEN_R
        LDR R0, [R1]
        ORR R0, #0x04
        STR R0, [R1]


        CPSIE  I
        BL Debug_Init
         B set_LED

loop
        LDR R1, =COUNT
        LDR R0, [R1]
        CMP R0, #50

        BEQ Skip
        BL Debug_Capture
;heartbeat
Skip
        BL Heartbeat
; Delay
        BL delay
;input PE1 test output PE0 2480009

        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]

        AND R2, #0
        AND R2, R0, #0x02
        LSR R2, #1
        CMP R2, #1
```

```
        BNE set_LED

        EOR R3, R0, #0x01
        STR R3, [R1]

        B loop


Debug_Init
        PUSH{R0, LR}
        BL SysTick_Init

        LDR R1, =COUNT
        MOV R0, #0
        STR R0, [R1]

        LDR R0, =DataBuffer
        LDR R1, =DataPt
        STR R0, [R1]

        LDR R0, =TimeBuffer
        LDR R1, =TimePt
        STR R0, [R1]

        MOV R0, #0xFFFFFFFF
        LDR R1, =DataBuffer
        BL Set_Buffer

        MOV R0, #0xFFFFFFFF
        LDR R1, =TimeBuffer
        BL Set_Buffer

        POP{R0, LR}
        BX LR

Debug_Capture
        LDR R1, =TimePt
        LDR R0, [R1]
        LDR R2, =NVIC_ST_CURRENT_R
        LDR R2, [R2]
        STR R2, [R0]
        ADD R0, #4
        STR R0, [R1]

        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]
        AND R2, R0, #0x01
        MOV R3, R2
        AND R2, R0, #0x02
        LSL R2, #3
        ORR R3, R2
        LDR R1, =DataPt
        LDR R0, [R1]
```

```
        STR R3, [R0]
        ADD R0, #4
        STR R0, [R1]

        LDR R1, =COUNT
        LDR R0, [R1]
        ADD R0, #1
        STR R0, [R1]

        BX LR

Heartbeat
        LDR R1, =GPIO_PORTF_DATA_R
        LDR R0, [R1]
        EOR R0, #0x04
        STR R0, [R1]

        BX LR

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

set_LED
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]
        ORR R0, #0x01
        STR R0, [R1]

        B loop

Set_Buffer
        MOV R2, #0
B_Loop
        STR R0, [R1]
        ADD R1, #4
        ADD R2, #1
        CMP R2, #50
        BNE B_Loop
        BX LR

        ALIGN
        END
```

Execution Time

Debugger_Capture takes 24 cycles to complete
Number of cycles between each Debugger_Capture is 2480022 cycles

Time for Debugger_Capture is 24*2*12.5*10^-9 = 6*10^-7 seconds
Time between each Debugger_Capture is 2480022*2*12.5*10^-9 = 0.06200055 seconds

Percentage Overhead is 100*(6*10^-7 seconds)/(0.06200055 seconds) = 0.000967%

Debugging Results

01000000010000000100000001000000
01000000100000001100000010000000
11000000100000001100000010000000
11000000100000001100000010000000
01000000010000000100000001000000
01000000010000000100000001000000
10000000110000001000000011000000
10000000110000001000000011000000
10000000110000001000000011000000
10000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
1000000011000000

97FCFF003D4DB400
E39D680089EE1C002F3FD100DB8F8500
87E039003331EE00DF81A2008BD25600
37230B00E373BF008FC473003B152800
E765DC0093B6900039074500DF57F900
85A8AD002BF96100D1491600779ACA00
1DEB7E00C33B33006F8CE7001BDD9B00
C72D5000737E04001FCFB800CB1F6D00
7770210023C1D500CF118A007B623E00
27B3F200D303A7007F545B0025A50F00
CBF5C3007146780017972C00BDE7E000
6338950009894900AFD9FD00552AB200
FB7A6600A1CB1A004D1CCF00F96C8300

Difference between two Time Buffer values is 00FFFC97 - 00B44D3D = 4BAF5A or 4960090

Time between two Time Buffer values is 4960090*12.5*10^-9 = 0.062001125 or about 62ms