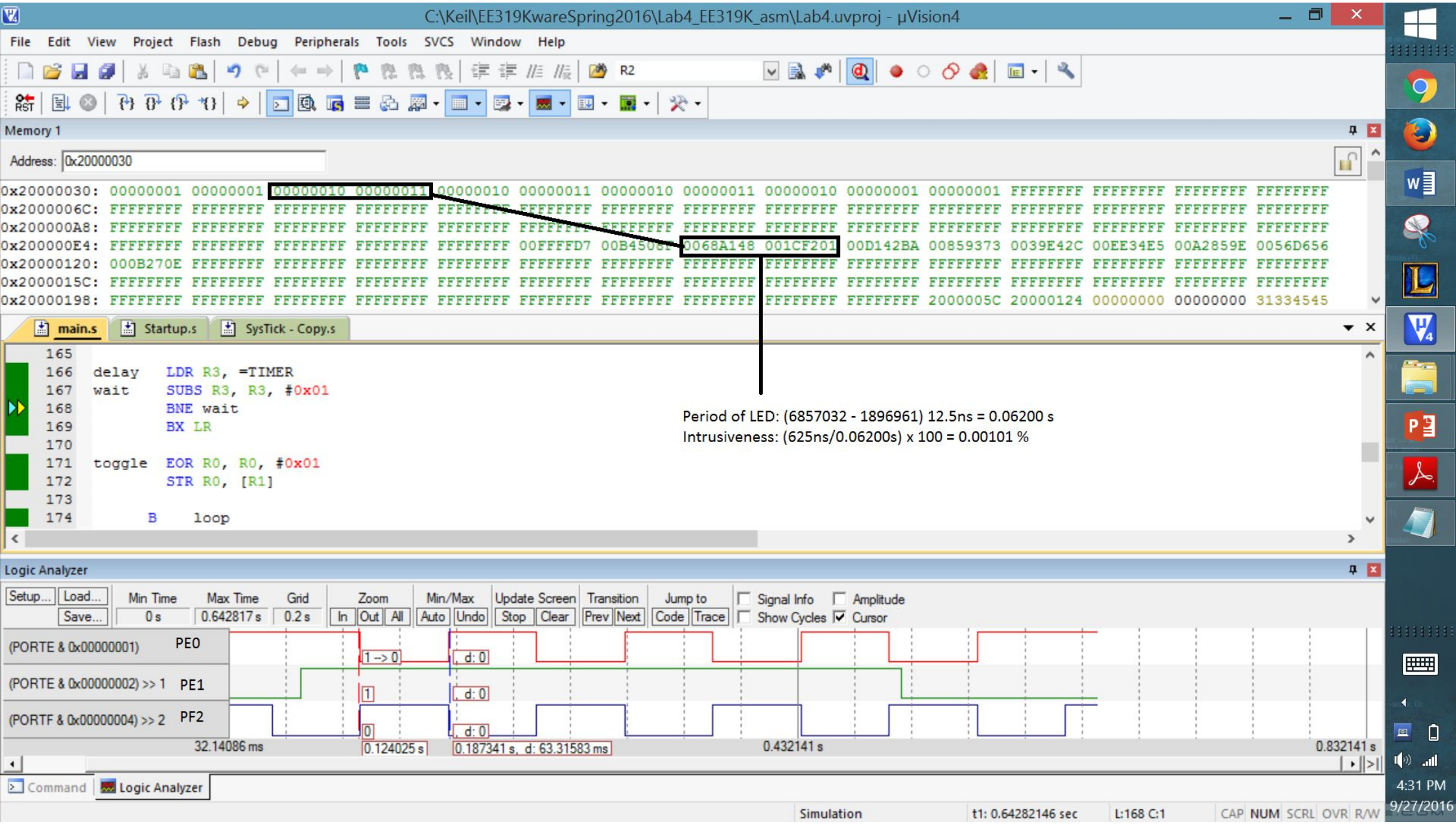
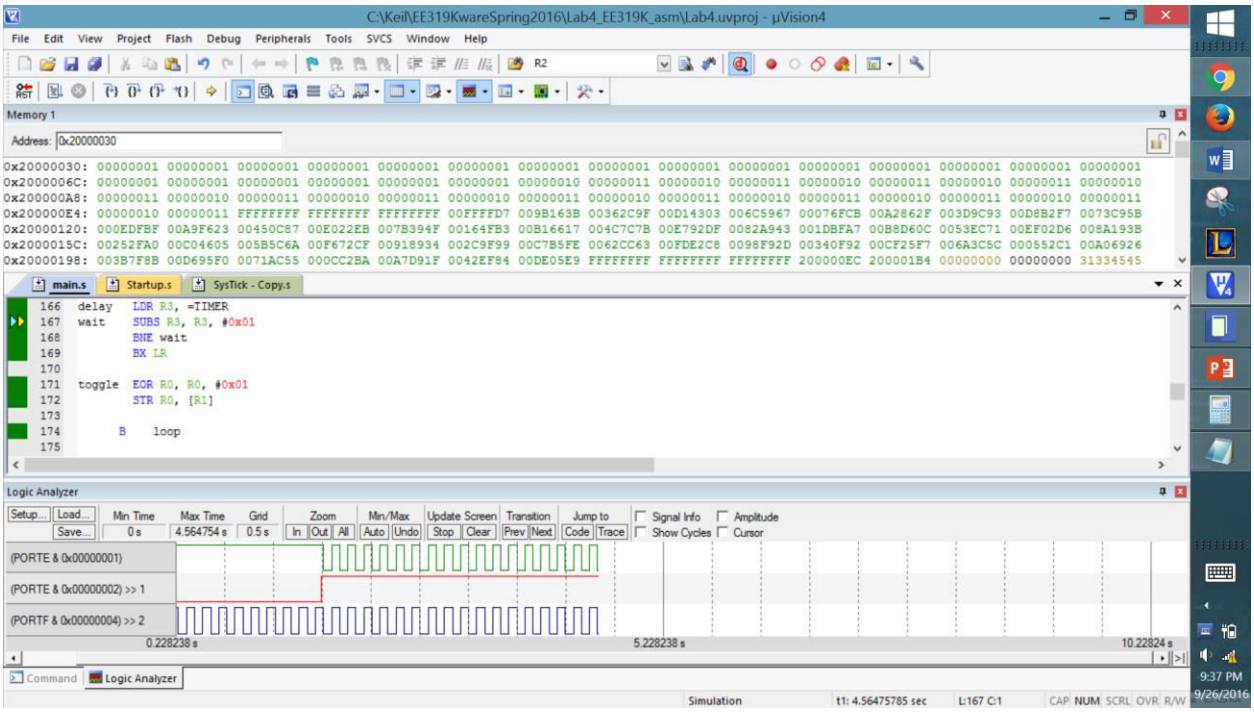

C:\Keil\EE319KwareSpring2016\Lab4_EE319K_asm\Lab4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Memory 1
Address: 0x20000030
0x20000030: 00000001 00000001 00000010 00000011 00000010 00000011 00000010 00000011 00000010 00000001 00000001 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x2000006C: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x200000A8: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x200000E4: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00FFFFD7 00B45084 0068A148 001CF201 00D142BA 00859373 0039E42C 00EE34E5 00A2859E 0056D656
0x20000120: 000B270E FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x2000015C: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
0x20000198: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 2000005C 20000124 00000000 00000000 31334545
main.s Startup.s SysTick - Copy.s
165
166 delay LDR R3, =TIMER
167 wait SUBS R3, R3, #0x01
168 BNE wait
169 BX LR
170
171 toggle EOR R0, R0, #0x01
172 STR R0, [R1]
173
174 B loop
Period of LED: (6857032 - 1896961) 12.5ns = 0.06200 s
Intrusiveness: (625ns/0.06200s) x 100 = 0.00101 %
Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 0.642817 s
Grid 0.2 s
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info Show Cycles Amplitude Cursor
(PORTE & 0x00000001) PE0
(PORTE & 0x00000002) >> 1 PE1
(PORTF & 0x00000004) >> 2 PF2
1 --> 0
d: 0
32.14086 ms
0.124025 s
0.187341 s, d: 63.31583 ms
0.432141 s
0.832141 s
Command Logic Analyzer
Simulation t1: 0.64282146 sec L:168 C:1 CAP NUM SCRL OVR R/W
4:31 PM
9/27/2016

```
:020000042000DA
:1000300001000000010000000100000001000000BC
:1000400001000000010000000100000001000000AC
:1000500001000000010000000100000001000000 9C
:1000600001000000010000000100000001000000 8C
:1000700001000000010000000100000001000000 6D
:1000800011000000010000000110000001000000 2E
:1000900011000000010000000100000001000000 4C
:1000A000010000000100000001000000FFFFFFFF51
:1000B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:1000C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF40
:1000D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:1000E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF20
:1000F000FFFFFFFFFFFFFFFFD7FFFF003B169B0047
:100100009F2C36000343D10067596C00CB6F07006A
:100110002F86A200939C3D00F7B2D8005BC9730004
:10012000BFDF0E0023F6A900870C4500EB22E0009C
:100130004F397B00B34F16001766B1007B7C4C0033
:10014000DF92E70044A98200A9BF1D000ED6B800C7
:1001500073EC5300D802EF003D198A00A12F25004F
:100160000546C000695C5B00CD72F60031899100E4
:10017000959F2C00FFFFFFFFFFFFFFFFFFFFFFFF2B
:10018000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
:10019000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6F
:1001A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5F
:1001B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4F
:0101C000AC92
:00000001FF
```



;****************** main.s ***************

; Program written by: ***Elvin Galarza and Ankith Kandikonda***

; Date Created: 1/22/2016

; Last Modified: 1/22/2016

; Section ***Tuesday 3-4***

; Instructor: ***Vijay Janapa Reddi***

; 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
datacmp                                         EQU 0x200000E8
TIMER                                       EQU 1653333
SWITCH              EQU 0x40024008  ;PE0
LED                 EQU 0x40024004  ;PE1
SYSCTL_RCGCGPIO_R       EQU 0x400FE608
SYSCTL_RCGC2_GPIOE      EQU 0x00000010   ; port E Clock Gating Control
SYSCTL_RCGC2_GPIOF      EQU 0x00000020   ; port F Clock Gating Control
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
        THUMB
        AREA    DATA, ALIGN=4
SIZE      EQU    50
;You MUST use these two buffers and two variables
;You MUST not change their names
```

```
;These names MUST be exported
        EXPORT DataBuffer
        EXPORT TimeBuffer
        EXPORT DataPt [DATA,SIZE=4]
        EXPORT TimePt [DATA,SIZE=4]
DataBuffer SPACE  SIZE*4
TimeBuffer SPACE  SIZE*4
DataPt    SPACE  4
TimePt    SPACE  4


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


Start BL   TExaS_Init  ; running at 80 MHz, scope voltmeter on PD3
; initialize Port E
          LDR R1, =SYSCTL_RCGCGPIO_R
          LDR R0, [R1]
          ORR R0, R0, #0x10
          STR R0, [R1]
          NOP
          NOP

          LDR R1, =GPIO_PORTE_DEN_R
          LDR R0, [R1]
```

```
        ORR R0, R0, #0x03
        STR R0, [R1]

        LDR R1, =GPIO_PORTE_DIR_R   ;LED is high
        LDR R0, [R1]
        ORR R0, R0, #0x01
        STR R0, [R1]

        LDR R1, =GPIO_PORTE_DIR_R    ;SW1 is low
        LDR R0, [R1]
        AND R0, R0, #0xFD
        STR R0, [R1]

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

        LDR R1, =GPIO_PORTE_DATA_R                    ;makes PE0 on, originally
        LDR R0, [R1]
        ORR R0, R0, #0x01
        STR R0, [R1]

; initialize Port F
        LDR R1, =SYSCTL_RCGCGPIO_R
        LDR R0, [R1]
        ORR R0, R0, #0x20
        STR R0, [R1]
        NOP
```

```
        NOP

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

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

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

; initialize debugging dump, including SysTick


    CPSIE  I    ; TExaS voltmeter, scope runs on interrupts

        BL         Debug_Init

loop

        BL   Debug_Capture

                ;// HEARTBEAT = toggles onboard LED on/off to signify that the code is running
```

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

;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;||||||||                                                  |||||||||
;||||||||              Lab 3 code                  |||||||||
;||||||||                                                  |||||||||
;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;|||||||||||||||||||||||||||||||||||||||||||||||||||||
;delay
;input PE1 test output PE0


        BL delay
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]
        LSLS R2, R0, #30                ;puts PE1 in most sig bit
        BMI toggle                      ;if switch is pressed, go to
toggle

        LDR R1, =GPIO_PORTE_DATA_R      ;if not then fall through, keep LED on
        LDR R0, [R1]
        ORR R0, R0, #0x01
```

```
                STR R0, [R1]
                B loop


delay   LDR R3, =TIMER
wait    SUBS R3, R3, #0x01
                BNE wait
                BX LR

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

          B   loop


;------------Debug_Init------------
; Initializes the debugging instrument
; Input: none
; Output: none
; Modifies: none
; Note: push/pop an even number of registers so C compiler is happy

Debug_Init
    ; // R0-R3 (parameters)
          ; // THIS IS JUST CODE TO INITIALIZE THE DEBUGGING SEQUENCE
          ; // THE SAME CONCEPT APPLIES TO PUTTING THE REAL DATA IN

NoDataFF        EQU 0xFFFFFFFF
```

```
        ; // Data Buffer Stack Initialization
                LDR R0, =DataBuffer
                LDR R1, =DataPt
                LDR R2, =SIZE
                LDR R3, =NoDataFF
                STR R0, [R1]
DLoop
                SUBS R2, #1                     ;counts down, where SIZE = size of array
(number of elements) = counter (8bytes)
                BMI  TArray                     ;go to 2nd Array (time array) when done with
I/O array (data buffer)
                STR  R3, [R0]           ;stores NoData into element space
                ADD R0, R0, #4          ;goes to next valid space to store elements
                B DLoop

        ;//Time Buffer Stack Initialization
TArray
                LDR R0, =TimeBuffer
                LDR R1, =TimePt
                LDR R2, =SIZE
                LDR R3, =NoDataFF
                STR R0, [R1]
TLoop
                SUBS R2, #1
                BMI      SysTick
                STR R3, [R0]
                ADD R0, R0, #4
                B TLoop
```

```
                ;// SysTick Initialization

SysTick

        IMPORT SysTick_Init
                MOV R2, R14                     ;saves return address
                BL SysTick_Init         ;changes Link Register value... need to save Link Register
                                                ;and load it back after calling SysTick
                MOV R14, R2
                BX LR


;------------Debug_Capture------------
; Dump Port E and time into buffers
; Input: none
; Output: none
; Modifies: none
; Note: push/pop an even number of registers so C compiler is happy
Debug_Capture


                LDR R2, =DataPt
                LDR R1, =datacmp
                LDR R0, [R2]
                CMP R1, R0
                BMI done
```

```
                LDR R3, [R2]
                LDR R1, =GPIO_PORTE_DATA_R
                LDR R0, [R1]
                AND R0, R0, #0x03
                LSLS R1, R0, #30                ; checking if shifting is needed
                BPL next                        ; checks to see if switch is not pressed,
if not pressed br next
                ADD R0, R0, #0x02               ; shifting bit 1 to bit 5
                ADD R0, R0, #0x04               ;  0000,00XY becomes..
                ADD R0, R0, #0x08               ;  000X,000Y
next    STR R0, [R3]
                ADD R0, R3, #4                  ; increment pointer ***
                STR R0, [R2]
                LDR R3, =TimePt
                LDR R1, =NVIC_ST_CURRENT_R
                LDR R0, [R1]
                LDR R2, [R3]
                STR R0, [R2]
                ADD R2, R2, #4
                STR R2, [R3]


done    BX LR


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