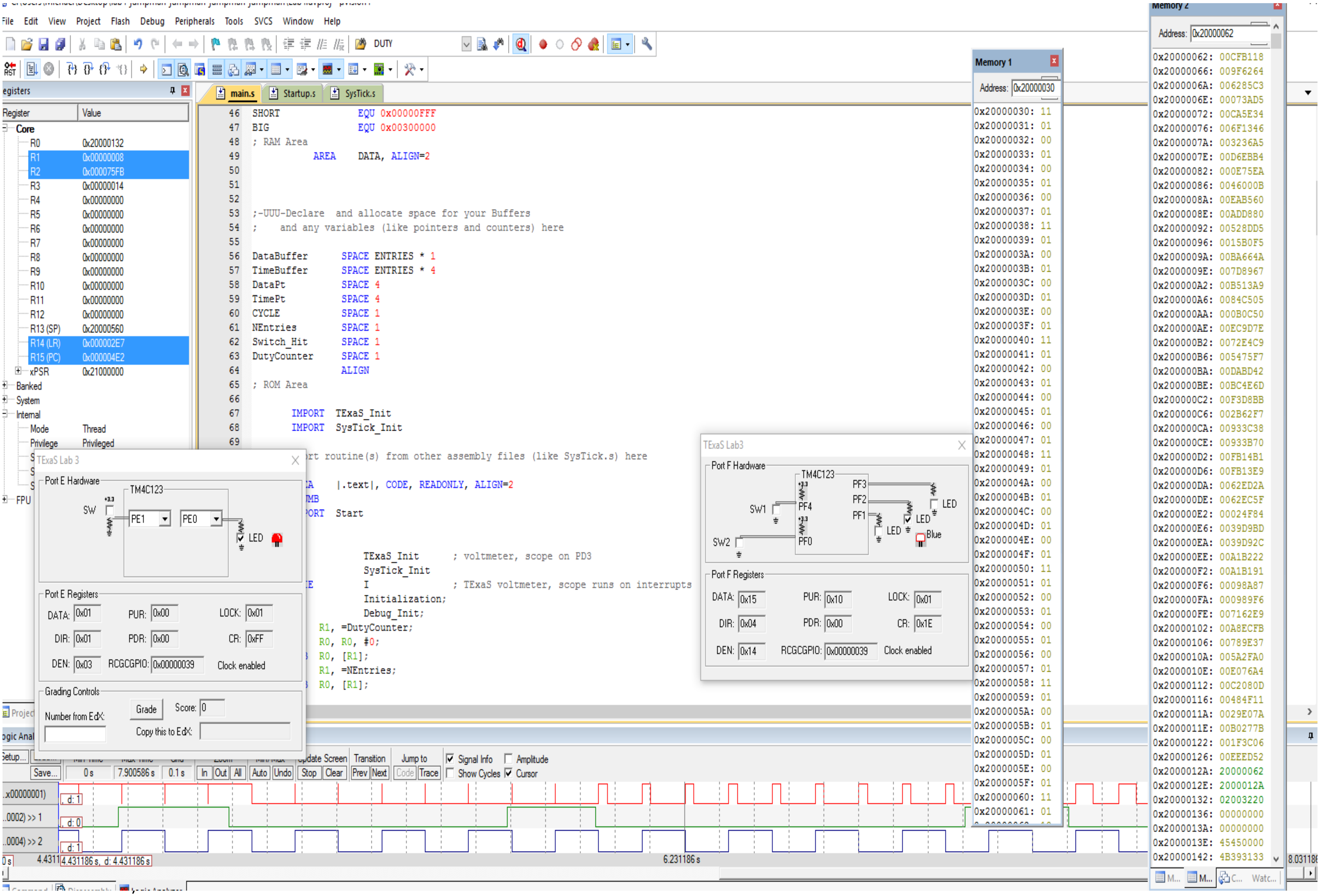


Register	Value
Core	
R0	0x20000132
R1	0x00000008
R2	0x000075FB
R3	0x00000014
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000560
R14 (LR)	0x000002E7
R15 (PC)	0x000004E2
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
FPU	

```
SHORT             EQU 0x00000FFF
BIG               EQU 0x00300000
; RAM Area
         AREA    DATA, ALIGN=2


;-UUU-Declare  and allocate space for your Buffers
;    and any variables (like pointers and counters) here

DataBuffer      SPACE ENTRIES * 1
TimeBuffer      SPACE ENTRIES * 4
DataPt          SPACE 4
TimePt          SPACE 4
CYCLE           SPACE 1
NEntries        SPACE 1
Switch_Hit      SPACE 1
DutyCounter     SPACE 1
                ALIGN
; ROM Area

       IMPORT  TExaS_Init
       IMPORT  SysTick_Init

rt routine(s) from other assembly files (like SysTick.s) here

A    |.text|, CODE, READONLY, ALIGN=2
MB
ORT  Start


                TExaS_Init      ; voltmeter, scope on PD3
                SysTick_Init
E               I               ; TExaS voltmeter, scope runs on interrupts
                Initialization;
                Debug_Init;
       R1, =DutyCounter;
       R0, R0, #0;
       R0, [R1];
       R1, =NEntries;
       R0, [R1];
```

TExaS Lab 3

Port E Hardware

Port E Registers

DATA:	0x01	PUR:	0x00	LOCK:	0x01
DIR:	0x01	PDR:	0x00	CR:	0xFF
DEN:	0x03	RCGCGPIO:	0x00000039	Clock enabled	

Grading Controls

Number from EdX: Grade Score: 0

Copy this to EdX:

TExaS Lab3

Port F Hardware

Port F Registers

DATA:	0x15	PUR:	0x10	LOCK:	0x01
DIR:	0x04	PDR:	0x00	CR:	0x1E
DEN:	0x14	RCGCGPIO:	0x00000039	Clock enabled	

Memory 1

Address: 0x20000030

```
0x20000030: 11
0x20000031: 01
0x20000032: 00
0x20000033: 01
0x20000034: 00
0x20000035: 01
0x20000036: 00
0x20000037: 01
0x20000038: 11
0x20000039: 01
0x2000003A: 00
0x2000003B: 01
0x2000003C: 00
0x2000003D: 01
0x2000003E: 00
0x2000003F: 01
0x20000040: 11
0x20000041: 01
0x20000042: 00
0x20000043: 01
0x20000044: 00
0x20000045: 01
0x20000046: 00
0x20000047: 01
0x20000048: 11
0x20000049: 01
0x2000004A: 00
0x2000004B: 01
0x2000004C: 00
0x2000004D: 01
0x2000004E: 00
0x2000004F: 01
0x20000050: 11
0x20000051: 01
0x20000052: 00
0x20000053: 01
0x20000054: 00
0x20000055: 01
0x20000056: 00
0x20000057: 01
0x20000058: 11
0x20000059: 01
0x2000005A: 00
0x2000005B: 01
0x2000005C: 00
0x2000005D: 01
0x2000005E: 00
0x2000005F: 01
0x20000060: 11
0x20000061: 01
```

Memory 2

Address: 0x20000062

```
0x20000062: 00CFB118
0x20000066: 009F6264
0x2000006A: 006285C3
0x2000006E: 00073AD5
0x20000072: 00CA5E34
0x20000076: 006F1346
0x2000007A: 003236A5
0x2000007E: 00D6EBB4
0x20000082: 000E75EA
0x20000086: 0046000B
0x2000008A: 00EAB560
0x2000008E: 00ADD880
0x20000092: 00528DD5
0x20000096: 0015B0F5
0x2000009A: 00BA664A
0x2000009E: 007D8967
0x200000A2: 00B513A9
0x200000A6: 0084C505
0x200000AA: 000B0C50
0x200000AE: 00EC9D7E
0x200000B2: 0072E4C9
0x200000B6: 005475F7
0x200000BA: 00DABD42
0x200000BE: 00BC4E6D
0x200000C2: 00F3D8BB
0x200000C6: 002B62F7
0x200000CA: 00933C38
0x200000CE: 00933B70
0x200000D2: 00FB14B1
0x200000D6: 00FB13E9
0x200000DA: 0062ED2A
0x200000DE: 0062EC5F
0x200000E2: 00024F84
0x200000E6: 0039D9BD
0x200000EA: 0039D92C
0x200000EE: 00A1B222
0x200000F2: 00A1B191
0x200000F6: 00098A87
0x200000FA: 000989F6
0x200000FE: 007162E9
0x20000102: 00A8ECFB
0x20000106: 00789E37
0x2000010A: 005A2FA0
0x2000010E: 00E076A4
0x20000112: 00C2080D
0x20000116: 00484F11
0x2000011A: 0029E07A
0x2000011E: 00B0277B
0x20000122: 001F3C06
0x20000126: 00EEED52
0x2000012A: 20000062
0x2000012E: 2000012A
0x20000132: 02003220
0x20000136: 00000000
0x2000013A: 00000000
0x2000013E: 45450000
0x20000142: 4B393133
```

2.) It is the image above.

*0x20000132 =   02 00 32 20

Cycle = 0x20

NEntries = 0x32

Switch_Hit = 0x00

DutyCounter = 0x02

3.) main code below

```
;****************** main.s ***************
; Program written by: ***Michael Hernandez and Arkan Abuyazid
; Date Created: 2/14/2017
; Last Modified: 3/01/2017
; Brief description of the program
;   The LED toggles at 8 Hz and a varying duty-cycle
;   Repeat the functionality from Lab2-3 but now we want you to
;   insert debugging instruments which gather data (state and timing)
;   to verify that the system is functioning as expected.
; Hardware connections (External: One button and one LED)
;  PE1 is Button input  (1 means pressed, 0 means not pressed)
;  PE0 is LED output (1 activates external LED on protoboard)

;  PF2 is Blue LED on Launchpad used as a heartbeat

; Instrumentation data to be gathered is as follows:

; After Button(PE1) press collect one state and time entry.

; After Buttin(PE1) release, collect 7 state and

; time entries on each change in state of the LED(PE0):

; An entry is one 8-bit entry in the Data Buffer and one

; 32-bit entry in the Time Buffer

;  The Data Buffer entry (byte) content has:

;    Lower nibble is state of LED (PE0)

;    Higher nibble is state of Button (PE1)

;  The Time Buffer entry (32-bit) has:

;    24-bit value of the SysTick's Current register (NVIC_ST_CURRENT_R)

; Note: The size of both buffers is 50 entries. Once you fill these

;       entries you should stop collecting data

; The heartbeat is an indicator of the running of the program.

; On each iteration of the main loop of your program toggle the

; LED to indicate that your code(system) is live (not stuck or dead).


GPIO_PORTE_DATA_R  EQU 0x400243FC

GPIO_PORTE_DIR_R   EQU 0x40024400
```

```
GPIO_PORTE_AFSEL_R EQU 0x40024420

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_PUR_R   EQU 0x40025510

GPIO_PORTF_DEN_R   EQU 0x4002551C

SYSCTL_RCGCGPIO_R  EQU 0x400FE608

ENTRIES                         EQU 50

DATA_RESET                      EQU 0xFF

TIME_RESET                      EQU 0xFFFFFFFF

TERM            EQU 0x00030B00 ; one millisecond 2867C (hardware lab 2-3),  30B00 (software),  28A9C (hardware for lab 4)

SHORT           EQU 0x00000FFF

BIG             EQU 0x00300000

; RAM Area

        AREA    DATA, ALIGN=2




;-UUU-Declare  and allocate space for your Buffers
```

```
;    and any variables (like pointers and counters) here


DataBuffer       SPACE ENTRIES * 1

TimeBuffer       SPACE ENTRIES * 4

DataPt                SPACE 4

TimePt                SPACE 4

CYCLE                 SPACE 1

NEntries              SPACE 1

Switch_Hit   SPACE 1

DutyCounter   SPACE 1

                        ALIGN

; ROM Area


    IMPORT  TExaS_Init

         IMPORT  SysTick_Init


;-UUU-Import routine(s) from other assembly files (like SysTick.s) here


    AREA    |.text|, CODE, READONLY, ALIGN=2

    THUMB

    EXPORT  Start
```

```
Start
   BL                  TExaS_Init            ; voltmeter, scope on PD3
        BL      SysTick_Init
   CPSIE               I                     ; TExaS voltmeter, scope runs on interrupts
        BL                         Initialization;
        BL             Debug_Init;
        LDR   R1, =DutyCounter;
        AND    R0, R0, #0;
        STRB  R0, [R1];
        LDR     R1, =NEntries;
   STRB  R0, [R1];
loop
   BL                  HIT;
   BL                  LED_ON;
        BL                         Debug_Capture;
        BL                         DELAY_ON;
        BL                         LED_OFF;
        BL                         Debug_Capture;
        BL                         DELAY_OFF;
        BL                         HeartBeat;
```

```
     B                          loop;








;------------Debug_Init------------

; Resets both buffers, initializes the points

; of said buffers, and activates SysTick Timer

; Input: None

; Output: None


Debug_Init


DPointer RN 0

DElement RN 1

Count RN 2

            LDR             DPointer, =DataPt                           ; Stores address of first data element

            LDR             R1, =DataBuffer                             ; to DataPt

            STR             R1, [DPointer]
```

```
        LDR             DPointer, [DPointer]

        MOV             DElement, #0xFFF

        MOV   Count, #0

Data_Clear

        STRB  DElement, [DPointer, Count]

        ADD             Count, Count, #1

        CMP   Count, #ENTRIES

        BHS             Time_Init

        B                       Data_Clear

Time_Init

TPointer RN 0

TElement RN 1

        LDR             TPointer, =TimePt

        LDR             R1, =TimeBuffer

        STR             R1, [TPointer]

        LDR             TPointer, [TPointer]

        LDR             TElement, =0xFFFFFFFF

        MOV             Count, #0

Time_Clear

        STR             TElement, [TPointer, Count]

        ADD             Count, Count, #4
```

```
          CMP    Count, #ENTRIES*4
          BHS           Owari
     B                  Time_Clear
Owari
     PUSH {R0, LR}                                          ; ADDED THIS PUSH AND POP BECAUSE CODE BECAME STUCK IN ENDLESS LOOP
          BL            SysTick_Init                  ; Initializes SysTick
          POP  {R0, LR};
          BX            LR



Initialization
     LDR R1, =SYSCTL_RCGCGPIO_R                ; TExaS_Init sets bus clock at 80 MHz
     LDR R0, [R1]                              ;Initialization
     ORR          R0,  R0, #0x30;
     STR  R0, [R1];
     NOP;
     NOP;
     NOP;
     NOP;
```

```
LDR R1, =GPIO_PORTE_DIR_R;
LDR R0, [R1];
AND           R0,  R0, #0xFD;
ORR           R0,  R0, #0x01                         ; PE1 input and PE0 output
STR  R0, [R1];
LDR R1, =GPIO_PORTE_AFSEL_R;
LDR R0, [R1];
AND           R0,  R0, #0x00;
STR  R0, [R1];
LDR R1, =GPIO_PORTE_DEN_R;
LDR R0, [R1];
ORR           R0,  R0, #0x03                         ; enables just PE0 and PE1
STR  R0, [R1];
LDR R1, =GPIO_PORTF_DIR_R;
LDR R0, [R1];
      AND   R0,  R0, #0x00                           ; PF4 INPUT
      ORR   R0,  R0, #0x04                           ; PF2 OUTPUT
STR  R0, [R1];
LDR R1, =GPIO_PORTF_AFSEL_R;
LDR R0, [R1];
AND           R0,  R0, #0x0000;
```

```
STR  R0, [R1];

LDR  R1, =GPIO_PORTF_PUR_R;

LDR  R0, [R1];

ORR           R0,  R0, #0x10;

STR  R0, [R1];

LDR  R1, =GPIO_PORTF_DEN_R;

LDR  R0, [R1];

ORR           R0,  R0, #0x14;

STR  R0, [R1]                                  ;END of initialization

                                                                                  ; TExaS_Init sets bus clock at 80 MHz


AND           R0, #0x0000;

ADD           R0,  R0, #16;

LDR  R1, =CYCLE;

STRB          R0, [R1]                                  ;code to store 20% to duty cycle before running loop

      LDR   R0, =Switch_Hit;

      AND   R1, R1, #0x00;

      STRB  R1, [R0];

      BX    LR;
```

```
        ;;------------HIT------------
; Updates duty cycle
;
; Input: None
; Output: None

HIT                                                             ; implement debug_Capture here
        PUSH   {R0-R2, R3};
        LDR    R0, =Switch_Hit;
        LDRB   R1, [R0];
        CMP    R1, #0;
        BNE                    Still_Pressed;
        LDR      R0, =GPIO_PORTE_DATA_R               ; check for switch release
   LDRSB  R1, [R0];
        AND    R1, R1, #0x02;
        CMP     R1, #0x02;
        BEQ                    Store;
        POP     {R0-R2, R3};
        BX                      LR;
Store
   LDR  R0, =Switch_Hit
```

```
            AND    R1, R1, #0;
            ADD    R1, R1, #1;
            STRB  R1, [R0];
            POP  {R0-R2, R3};
            BX                              LR;
Still_Pressed
     LDR  R0, =GPIO_PORTE_DATA_R;
            LDR     R1, [R0];
            AND    R1,  R1, #0x02;
            CMP    R1, #0x02;
            BNE                                         Update;
            POP     {R0-R2, R3};
            BX                              LR;
Update
     LDR   R0, =Switch_Hit;
            AND   R1, R1, #0x00;
            STRB  R1, [R0];
            LDR     R0, =CYCLE                                    ; Duty Cycle subroutine
     LDRB           R1, [R0];
     ADDS           R1, #16                                       ; Duty cycle to next level
     CMP            R1, #80                                       ; 0X60
```

```
    BNE                           Valid;


    STRB            R1, [R0];

    SUBS            R1, #80;

    B                                    DONE;

Valid

    CMP             R1, #96;

    BEQ                           RESET;

    STRB            R1, [R0];

    B                                    DONE;           ; GOES TO DELAY SUBROUTINE

RESET

    SUBS            R1, #96;

    STRB            R1, [R0];

DONE

    POP {R0-R2, R3};

    BX              LR;


           ;------------Debug_Capture------------

; Updates buffers and  pointers

;

; Input: None
; Output: None
Debug_Capture
    PUSH {R0-R11, R12, LR}                                                  ; %= 0.00196 calculated as  ((100) (.000002450))/.125
    LDR   R1,      =NEntries;
           LDRB  R0, [R1];
           CMP   R0,        #50;
           BEQ                     Full;
           LDR   R11, =Switch_Hit;
           LDRB  R10, [R11];
           CMP   R10, #0;
           BNE                     Next_Case;
           LDR   R1, =DutyCounter;
           LDRB  R0, [R1];
           CMP   R0, #0;
           BEQ                     skipped;
           CMP   R0, #7;
           BNE                     Run
           AND   R0, R0, #0x0000                  ; this line may cause problems
           STR    R0, [R1];
           B    Clear;
```

```
Run
   ADD   R0, R0, #1;
         STRB  R0, [R1];
Clear
         LDR     R1,        =GPIO_PORTE_DATA_R;
         LDR     R0, [R1];
         LDR     R3,        =0xE000E018 ;
         LDR     R2, [R3];
         AND   R4,  R0, #0x02                       ; Port E bit 1
         AND   R5,  R0, #0X01                 ; Port E bit 0
         LSL   R4,  R4, #3;
         ORR   R4,  R4, R5;
         LDR   R6,          =DataPt;
         LDR   R7,  [R6];
         STRB  R4,  [R7];
         ADD   R7,   R7, #1;
         STR   R7,  [R6];
         LDR   R8,          =TimePt;
         LDR   R9,  [R8];
         STR   R2,  [R9];
         ADD   R9,   R9, #4;
```

```
            STR   R9,  [R8]                              ; Assumes NVIC_ST_CURRENT_R is the time information


            LDR   R1,   =NEntries;

            LDRB  R0, [R1];

            ADD   R0, R0, #1

            STRB  R0,  [R1];

            POP  {R0-R11, R12, PC}

Full

    POP  {R0-R11, R12, PC}

Next_Case

            LDR R1, =DutyCounter;

            LDRB  R0, [R1];

            CMP   R0, #0;

    BEQ   Run

skipped

    POP   {R0-R11, R12, PC}


           ;------------HeartBeat------------

; Toggles PF2 Blue LED on and  off

;

; Input: PF2
; Output: PF2

HeartBeat

   PUSH { R0-R3}

         LDR   R1,          =GPIO_PORTF_DATA_R ;

         LDR     R0, [R1];

         EOR     R0,  R0, #0x04;

         STR     R0, [R1];

         POP     {R0-R3};

         BX    LR;




;-----------LED_ON----------

; Turns on LED

;

; Input: none

; Output: None

LED_ON

         PUSH  {R0, R3}

         LDR     R0, =GPIO_PORTE_DATA_R        ;Turn on LED

         LDR     R3, [R0];
```

```
        ORR     R3, #0X01;
        STR     R3, [R0];
   POP  {R0, R3}
        BX      LR
;-----------LED_OFF----------
; Turns off LED
;
; Input: none
; Output: None
LED_OFF
        PUSH  {R0, R3}
        LDR     R0, =GPIO_PORTE_DATA_R        ;Turn on LED
        LDR     R3, [R0];
        AND     R3, #0XFE;
        STR     R3, [R0];
        POP   {R0, R3}
        BX      LR
```

```
;---------DELAY_ON-------
;
;
DELAY_ON
    PUSH {R0-R3}
    LDR  R0, =CYCLE                              ;Delay subroutine
    LDRSB R1, [R0];
    CMP         R1, #0;
    BEQ   LEAVE;
REPEAT2
    LDR  R2, =TERM;
REPEAT

    SUBS  R2, #1                                 ; Millisecond load
    CMP         R2, #0;
    BNE         REPEAT;
    SUBS        R1, #8;
    CMP         R1, #0                                  ; on for duty cycle
    BNE         REPEAT2;

LEAVE
```

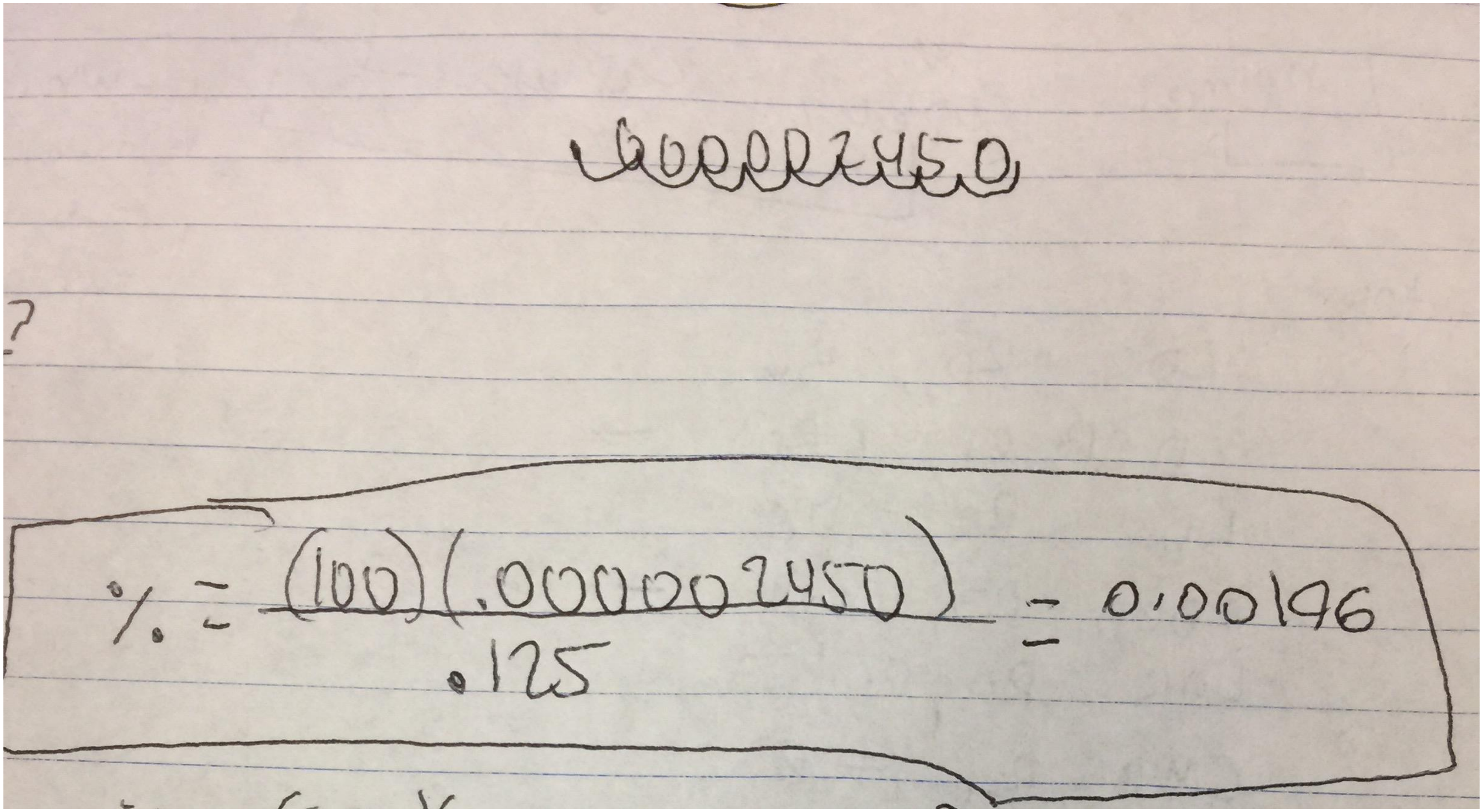
```
    POP {R0-R3}

    BX LR




;----------DELAY_OFF-------

;

;

;

DELAY_OFF

    PUSH {R0-R3}

    LDR  R0, =CYCLE;

    LDRB            R1, [R0];

    RSBS            R1,  R1, #80;

    BEQ               LEAVE3;




REPEAT4

    LDR  R2, =TERM;
```

```
REPEAT5

    SUBS        R2, #1                                                                     ; off for (duty cycle)!
    CMP         R2, #0;
    BNE         REPEAT5;
    CMP         R1, #0;
    BEQ                 LEAVE3;
    SUBS        R1, #8;
    CMP         R1, #0;
    BNE REPEAT4;
LEAVE3
    POP {R0-R3}
          BX    LR



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

Systick.s is below here

```
; SysTick.s
; Module written by: Arkan Abuyazid & Michael Hernandez
; Date Created: 2/14/2017
; Last Modified: 3/01/2017
; Brief Description: Initializes SysTick

NVIC_ST_CTRL_R        EQU 0xE000E010
NVIC_ST_RELOAD_R      EQU 0xE000E014
NVIC_ST_CURRENT_R     EQU 0xE000E018
TICKS                                    EQU 0x00FFFFFF

       AREA    |.text|, CODE, READONLY, ALIGN=2
       THUMB
; ;-UUU-Export routine(s) from SysTick.s to callers
                      EXPORT SysTick_Init
;------------SysTick_Init------------
; ;-UUU-Complete this subroutine
; Initialize SysTick with busy wait running at bus clock.
; Input: none
; Output: none
; Modifies: ??

SysTick_Init
                                                                        ; Set Enable bit to 0
     LDR          R0,=NVIC_ST_CTRL_R;
     LDR          R1, [R0];
     AND          R1, R1, #0xFE;
     STR          R1, [R0];
                                                                        ; 10 million ticks is .125
ms in 80MHz
     LDR          R0,=NVIC_ST_RELOAD_R;
     LDR          R1, =TICKS
     STR          R1, [R0];


                                                                        ; Reset Current Register
     LDR          R0,=NVIC_ST_CURRENT_R;
     STR          R1,[R0]                                     ; Attempting to store a number in
CURRENT_R
                                                                        ; Resets it automatically


                                                                        ; Turn on counter
     LDR           R0,=NVIC_ST_CTRL_R

     LDR           R1, [R0]

     ORR                  R1, R1, #0x00000005                                   ; Enable bit = 1, Clock bit = 1

     AND           R1, R1, #0xFFFFFFFD                                   ; Interrupt bit = 0;

     STR                  R1, [R0]


BX  LR                                             ; return



ALIGN                                           ; make sure the end of this section is aligned

END                                                    ; end of file
```

4.)

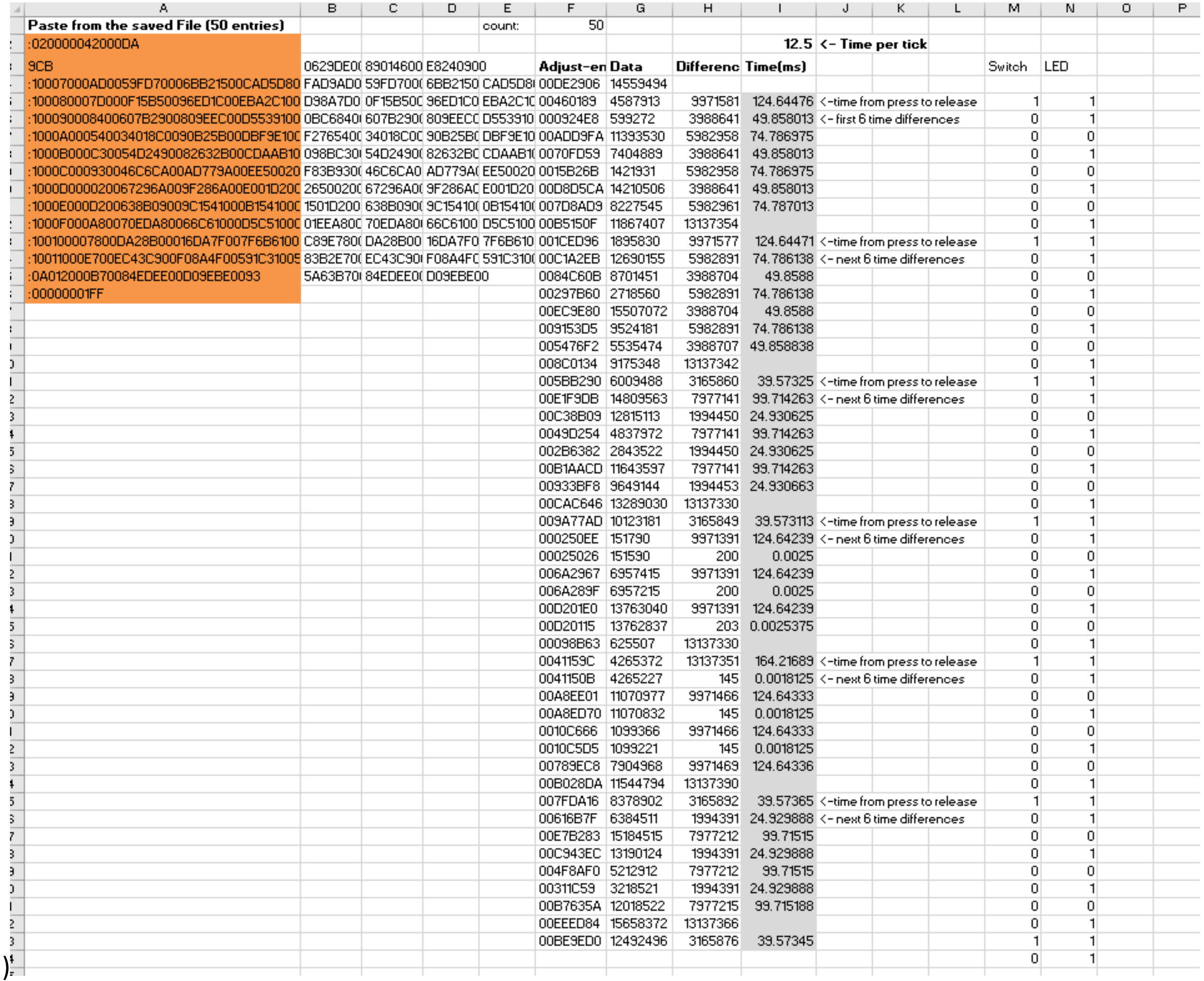


00002450

?

$$\boxed{\% = \frac{(100)(.000002450)}{.125} = 0.00196}$$

5.)



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Paste from the saved File (50 entries)				count:	50										
:020000042000DA								**12.5**	**<- Time per tick**						
9CB	0629DE0(	89014600	E8240900		**Adjust-en**	**Data**	**Differenc**	**Time(ms)**				Switch	LED		
:10007000AD0059FD70006BB21500CAD5D80	FAD9AD0	59FD700(	6BB2150	CAD5D8(	00DE2906	14559494									
:100080007D000F15B50096ED1C00EBA2C100	D98A7D0	0F15B50[	96ED1C0	EBA2C1[	00460189	4587913	9971581	124.64476	<-time from press to release			1	1		
:100090008400607B2900809EEC00D5539100	0BC6840(	607B290(	809EEC[	D553910	000924E8	599272	3988641	49.858013	<- first 6 time differences			0	1		
:1000A000540034018C0090B25B00DBF9E10[	F276540[	34018C0[	90B25B[	DBF9E10	00ADD9FA	11393530	5982958	74.786975				0	0		
:1000B000C30054D2490082632B00CDAAB10	098BC30(	54D2490(	82632B[	CDAAB1(	0070FD59	7404889	3988641	49.858013				0	1		
:1000C000930046C6CA00AD779A00EE50020	F83B930(	46C6CA0	AD779A(	EE50020	0015B26B	1421931	5982958	74.786975				0	0		
:1000D000020067296A009F286A00E001D20[	2650020[	67296A0(	9F286A[	E001D20	00D8D5CA	14210506	3988641	49.858013				0	1		
:1000E000D200638B09009C1541000B154100[	1501D200	638B090(	9C15410(	0B15410(	007D8AD9	8227545	5982961	74.787013				0	0		
:1000F000A80070EDA80066C61000D5C5100[	01EEA80[	70EDA80(	66C6100	D5C5100	00B5150F	11867407	13137354					0	1		
:100100007800DA28B00016DA7F007F6B6100	C89E780(	DA28B00	16DA7F0	7F6B610	001CED96	1895830	9971577	124.64471	<-time from press to release			1	1		
:10011000E700EC43C900F08A4F00591C31005	83B2E70(	EC43C90(	F08A4F[	591C310(	00C1A2EB	12690155	5982891	74.786138	<- next 6 time differences			0	1		
:0A012000B70084EDEE00D09EBE0093	5A63B70(	84EDEE0(	D09EBE00		0084C60B	8701451	3988704	49.8588				0	0		
:00000001FF					00297B60	2718560	5982891	74.786138				0	1		
					00EC9E80	15507072	3988704	49.8588				0	0		
					009153D5	9524181	5982891	74.786138				0	1		
					005476F2	5535474	3988707	49.858838				0	0		
					008C0134	9175348	13137342					0	1		
					005BB290	6009488	3165860	39.57325	<-time from press to release			1	1		
					00E1F9DB	14809563	7977141	99.714263	<- next 6 time differences			0	1		
					00C38B09	12815113	1994450	24.930625				0	0		
					0049D254	4837972	7977141	99.714263				0	1		
					002B6382	2843522	1994450	24.930625				0	0		
					00B1AACD	11643597	7977141	99.714263				0	1		
					00933BF8	9649144	1994453	24.930663				0	0		
					00CAC646	13289030	13137330					0	1		
					009A77AD	10123181	3165849	39.573113	<-time from press to release			1	1		
					000250EE	151790	9971391	124.64239	<- next 6 time differences			0	1		
					00025026	151590	200	0.0025				0	0		
					006A2967	6957415	9971391	124.64239				0	1		
					006A289F	6957215	200	0.0025				0	0		
					00D201E0	13763040	9971391	124.64239				0	1		
					00D20115	13762837	203	0.0025375				0	0		
					00098B63	625507	13137330					0	1		
					0041159C	4265372	13137351	164.21689	<-time from press to release			1	1		
					0041150B	4265227	145	0.0018125	<- next 6 time differences			0	1		
					00A8EE01	11070977	9971466	124.64333				0	0		
					00A8ED70	11070832	145	0.0018125				0	1		
					0010C666	1099366	9971466	124.64333				0	0		
					0010C5D5	1099221	145	0.0018125				0	1		
					00789EC8	7904968	9971469	124.64336				0	0		
					00B028DA	11544794	13137390					0	1		
					007FDA16	8378902	3165892	39.57365	<-time from press to release			1	1		
					00616B7F	6384511	1994391	24.929888	<- next 6 time differences			0	1		
					00E7B283	15184515	7977212	99.71515				0	0		
					00C943EC	13190124	1994391	24.929888				0	1		
					004F8AF0	5212912	7977212	99.71515				0	0		
					00311C59	3218521	1994391	24.929888				0	1		
					00B7635A	12018522	7977215	99.715188				0	0		
					00EEED84	15658372	13137366					0	1		
					00BE9ED0	12492496	3165876	39.57345				1	1		
												0	1		