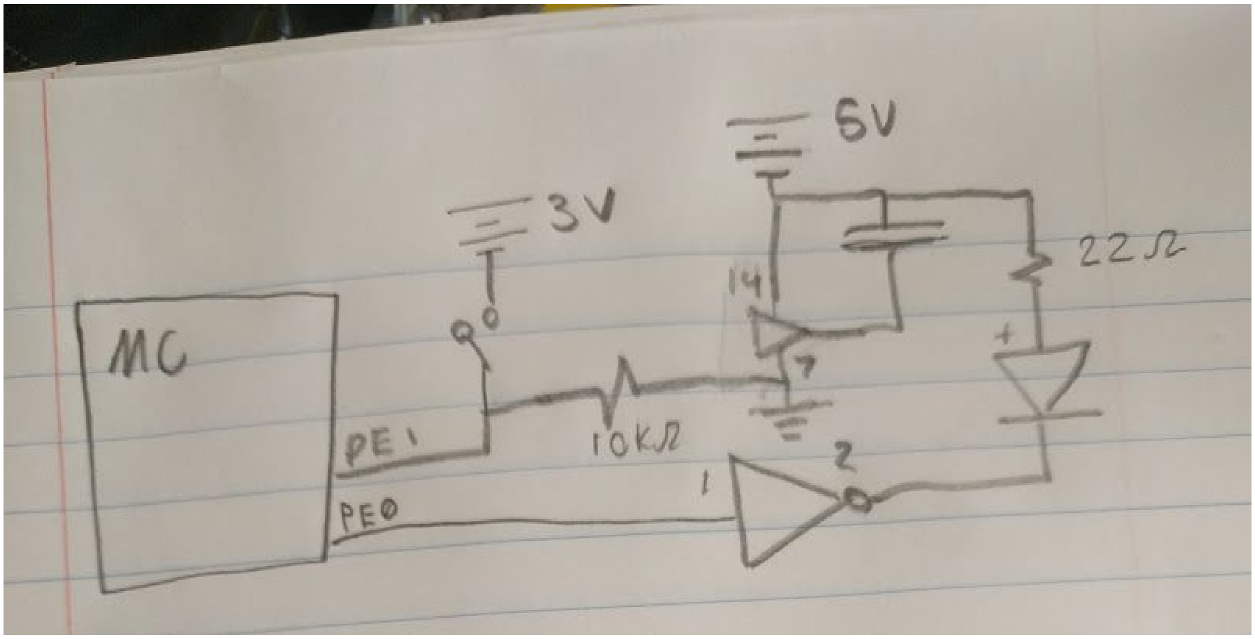
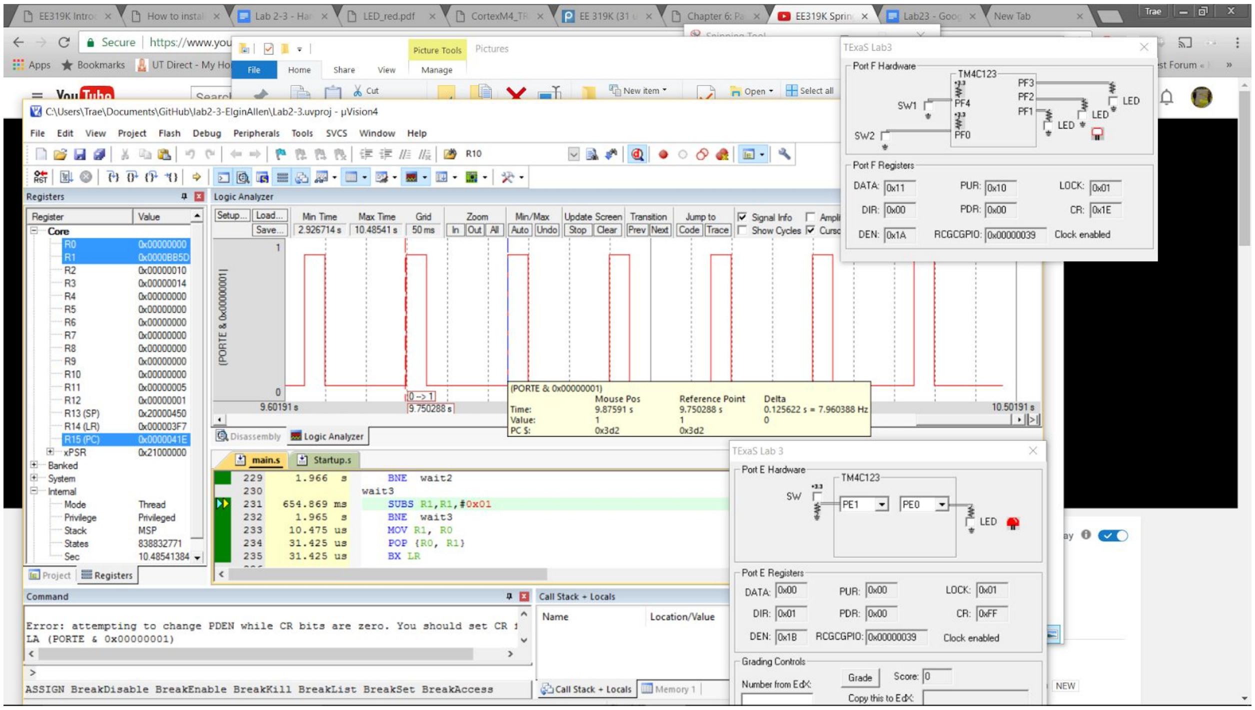
Zach Sisti and Trae Allen Deliverables

1. Circuit Diagram



2. Screenshots



Register	Value
Core	
R0	0x00000000
R1	0x0000BB5D
R2	0x00000010
R3	0x00000014
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000005
R12	0x00000001
R13 (SP)	0x20000450
R14 (LR)	0x000003F7
R15 (PC)	0x0000041E
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	838832771
Sec	10.48541384

```
(PORTE & 0x00000001)
        Mouse Pos     Reference Point   Delta
Time:   9.87591 s     9.750288 s        0.125622 s = 7.960388 Hz
Value:  1             1                 0
PC $:   0x3d2         0x3d2
```

```
229     1.966  s        BNE   wait2
230                 wait3
231   654.869 ms        SUBS R1,R1,#0x01
232     1.965  s        BNE   wait3
233    10.475 us        MOV R1, R0
234    31.425 us        POP {R0, R1}
235    31.425 us        BX LR
```

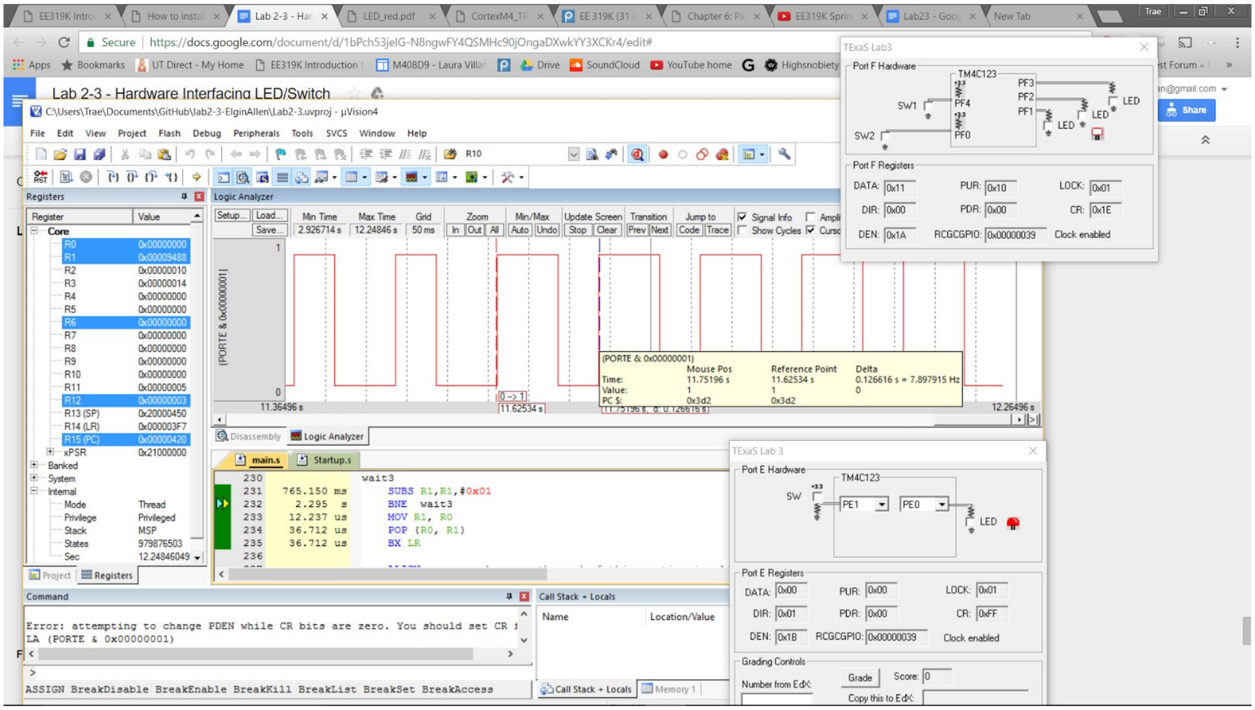
```
Error: attempting to change PDEN while CR bits are zero. You should set CR 1
LA (PORTE & 0x00000001)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Port F Registers

DATA:	0x11	PUR:	0x10	LOCK:	0x01
DIR:	0x00	PDR:	0x00	CR:	0x1E
DEN:	0x1A	RCGCGPIO:	0x00000039	Clock enabled	

Port E Registers

DATA:	0x00	PUR:	0x00	LOCK:	0x01
DIR:	0x01	PDR:	0x00	CR:	0xFF
DEN:	0x1B	RCGCGPIO:	0x00000039	Clock enabled	



Register	Value
Core	
R0	0x00000000
R1	0x00009488
R2	0x00000010
R3	0x00000014
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000005
R12	0x00000003
R13 (SP)	0x20000450
R14 (LR)	0x000003F7
R15 (PC)	0x00000420
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	979876503
Sec	12.24846049

```
(PORTE & 0x00000001)
        Mouse Pos     Reference Point   Delta
Time:   11.75196 s    11.62534 s        0.126616 s = 7.897915 Hz
Value:  1             1                 0
PC $:   0x3d2         0x3d2
```

```
230                 wait3
231   765.150 ms        SUBS R1,R1,#0x01
232     2.295  s        BNE   wait3
233    12.237 us        MOV R1, R0
234    36.712 us        POP {R0, R1}
235    36.712 us        BX LR
236
```

```
Error: attempting to change PDEN while CR bits are zero. You should set CR 1
LA (PORTE & 0x00000001)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Port F Registers

DATA:	0x11	PUR:	0x10	LOCK:	0x01
DIR:	0x00	PDR:	0x00	CR:	0x1E
DEN:	0x1A	RCGCGPIO:	0x00000039	Clock enabled	

Port E Registers

DATA:	0x00	PUR:	0x00	LOCK:	0x01
DIR:	0x01	PDR:	0x00	CR:	0xFF
DEN:	0x1B	RCGCGPIO:	0x00000039	Clock enabled	

3. Switch Measurements

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9760	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	6.4	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	4.8	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	.49 .46	mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	0	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

4. LED Measurements

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	220	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	10.3	volts	(measured with voltmeter relative to ground, *notice that the +5V power is not exactly +5 volts*)
3	TM4C123 Output, V_{PE0} input to 7406	0	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
4	7406 Output, V_{k-} LED k-	5.2	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
5	LED a+, V_{a+} Bottom side of R19	10.3	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
6	LED voltage	5.1	volts	calculated as V_{a+} - V_{k-}
7	LED current	0 0	mA	calculated as $(V_{+5} - V_{a+})$/R19 and measured with an ammeter
8	TM4C123 Output, V_{PE0} input to 7406	0	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
9	7406 Output, V_{k-} LED k-	0	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
10	LED a+, V_{a+} Bottom side of R19	10.3	volts	with **PE0** = 1 (measured with voltmeter relative to ground)

11	LED voltage	10.3	volts	calculated as V_{a+} - V_{k-}
12	LED current	0 0	mA	calculated as $(V_{+5}$ - $V_{a+})$/R19 and measured with an ammeter

5. Assembly Code

```
GPIO_PORTE_DATA_R  EQU 0x400243FC
GPIO_PORTE_DIR_R   EQU 0x40024400
GPIO_PORTE_AFSEL_R EQU 0x40024420
GPIO_PORTE_DEN_R   EQU 0x4002451C
; PortF device registers
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
      IMPORT  TExaS_Init
      AREA    |.text|, CODE, READONLY, ALIGN=2
      THUMB
      EXPORT  Start

Start
 ; TExaS_Init sets bus clock at 80 MHz
     BL  TExaS_Init ; voltmeter, scope on PD3
     CPSIE  I    ; TExaS voltmeter, scope runs on interrupts
          LDR R1, =SYSCTL_RCGCGPIO_R ;1) activate clock for Port F and E
          LDR R0, [R1]
          ORR R0, R0, #0x30 ;set bit 4,5 to turn on clock
          STR R0, [R1]
          NOP
          NOP ;allow time for clock to finish
          LDR R1, =GPIO_PORTF_DIR_R ;5) set direction register
          LDR R0, [R1]
          MOV R0, #0x00 ;set all pins as inputs
          STR R0, [R1]
          LDR R1, =GPIO_PORTF_DEN_R ;7) enable Port F digital port
          LDR R0, [R1]
          MOV R0, #0x1B ;1 means enable digital I/O
          STR R0, [R1]
          LDR R1, =GPIO_PORTF_AFSEL_R
          LDR R1, [R1]
          AND R0, R0, #0x00
          STR R0, [R1]
          LDR R1, =GPIO_PORTF_PUR_R ;pull-up resistors for PF4
          MOV R0, #0x10 ;enable weak pull-up on PF4
          STR R0, [R1]
```

```
        LDR R1, =GPIO_PORTE_DIR_R ;5) set direction register
        LDR R0, [R1]
        MOV R0, #0x1 ;set pins 3-1 input, pin 0 as output
        STR R0, [R1]
        LDR R1, =GPIO_PORTE_DEN_R ;7) enable Port E digital port
        LDR R0, [R1]
        MOV R0, #0x1B ;1 means enable digital I/O
        STR R0, [R1]
        LDR R1, =GPIO_PORTE_AFSEL_R
        LDR R1, [R1]
        AND R0, R0, #0x00
        STR R0, [R1]

        MOV R11, #5  ;PERIOD
        MOV R12, #1  ;DUTY CYCLE

loop
        LDR R1, =GPIO_PORTF_DATA_R
        LDR R2, [R1]
        AND R2, R2, #0x10
        AND R0, #0                  ;CLEAR R0
        CMP R2, R0
        BEQ breathe
        CMP R9, R0
        BGT noduty
        BL toggleLightOn
noduty
        ADD R0, R12, R0             ;R0=DUTY
        BL delay
        BL delay
        SUB R0, R11, R12      ;R0=PERIOD-DUTY
        AND R1, R1, #0
        CMP R0, R1
        BEQ fullcycle
        BL toggleLightOff
fullcycle
        BL delay
        BL delay
        LDR R1, =GPIO_PORTE_DATA_R   ;CHECKING SWITCH
        LDR R0, [R1]
        AND R2, R0, #0x02
        MOV R0, #0                                      ;CREATING CONSTANT FOR
COMPARING
        CMP R2, R0                                      ;CHECKING SWITCH STATE
        BEQ notPressed
        AND R9, R9, #0
        AND R6, R6, #0                                  ;CLEAR R6
        ADD R6, R6, #1                          ;INDICATES SWITCH HAS BEEN PRESSED
        B loop
notPressed
        AND R5, R5, #0                                  ;CLEAR R5
        CMP R6, R5                                      ;CHECK IF SWITCH HAS BEEN PRESSED
        BEQ loop
        CMP R11, R12                                    ;CHECKS IF DUTY=PERIOD
        BEQ reset
        ADD R12, R12, #1                        ;IF NOT INCREASE DUTY (20%)
        AND R6, R6, #0
        B loop
reset
        AND R12, R12, #0                        ;RESETS DUTY TO 0%
        AND R6, R6, #0
        AND R9, R9, #0
        ADD R9, R9, #1
        B loop

breathe
        PUSH {R12, R7, R0, R11}
        MOV R11, #10
        MOV R7, #0                      ;SET COUNTER TO 0
        MOV R10, #0                     ;UP/DOWN INDICATOR (0 UP, 1 DOWN)
        MOV R1, #0                      ;0 CONSTANT FOR COMPARING
breathe1
        BL toggleLightOn
        ADD R0, R12, R0                 ;R0=DUTY
        BL delay
        BL toggleLightOff
        SUB R0, R11, R12        ;R0=PERIOD-DUTY
        BL delay
        CMP R10, R1                     ;increment or decrement (up/down indicator)
        BEQ increment                   ;if R10==0
        ADD R12, R12, #-1       ;decrement
        B skp
increment
        ADD R12, R12, #1
```

```
skp
        CMP R12, R11                    ;are we at top?
        BEQ top
        CMP R12, R1                     ;are we at bottom?
        BEQ bottom
        B fin
top
        MOV R10, #1                     ;set up/down indicator to down direction (1)
        B fin
bottom
        MOV R10, #0                     ;set up/down indicator to up direction (0)
        B fin
fin
        LDR R1, =GPIO_PORTF_DATA_R                              ;check if button is still pressed
        LDR R2, [R1]
        AND R2, R2, #0x10
        AND R0, #0                      ;CLEAR R0
        CMP R2, R0
        BEQ breathe1
        POP {R12, R7, R0, R11}
                B    loop



toggleLightOn
        PUSH {R0, R1, R2, LR}
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]
        ORR R0, R0, #0x01                               ;SETTING PE0 TO 1
        STR R0, [R1]                                    ;STORES NEW DATAREG
        POP {R0, R1, R2, LR}
        BX LR

toggleLightOff
        PUSH {R0, R1, R2, LR}
        LDR R1, =GPIO_PORTE_DATA_R
        LDR R0, [R1]
        AND R0, R0, #0xFE                               ;SETTING PE0 TO 0
        STR R0, [R1]                                    ;STORES NEW DATAREG
        POP {R0, R1, R2, LR}
        BX LR

delay
```

```
        PUSH {LR, R1, R0, R2}
        ;MOV R2, #40
        ;MUL R0, R0, R2
while
        MOV R1, #0                                   ;CLEAR R1
        BL sub1
        ADD R0, #-1                                  ;DECREMTENTS WHILE LOOP
INDICATOR
        CMP R0, R1
        BGT while                                    ;REPEATS EACH 1/5 OF THE
DUTY CYCLE
        POP {LR, R1, R0, R2}
        BX LR




sub1                                                 ;DELAYS FOR 1/80th OF PERIOD
        PUSH {R0,R1}
        MOV R0,#0xF424                               ;SETS R0 TO ARBIRTRARY
CONSTANT WHICH WE FOUND WITH TRIAL AND ERROR
        MOV R1, R0                                   ;SETS R1 TO R0
wait
        SUBS R0,R0,#0x01                     ;DELAYS FOR A VERY SMALL AMOUNT
OF TIME
        BNE  wait
wait1
        SUBS R1,R1,#0x01
        BNE  wait1
        MOV R0,#0xF424
        MOV R1, R0
wait2
                SUBS R0,R0,#0x01
        BNE  wait2
wait3
        SUBS R1,R1,#0x01
        BNE  wait3
        MOV R1, R0
        POP {R0, R1}
        BX LR

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