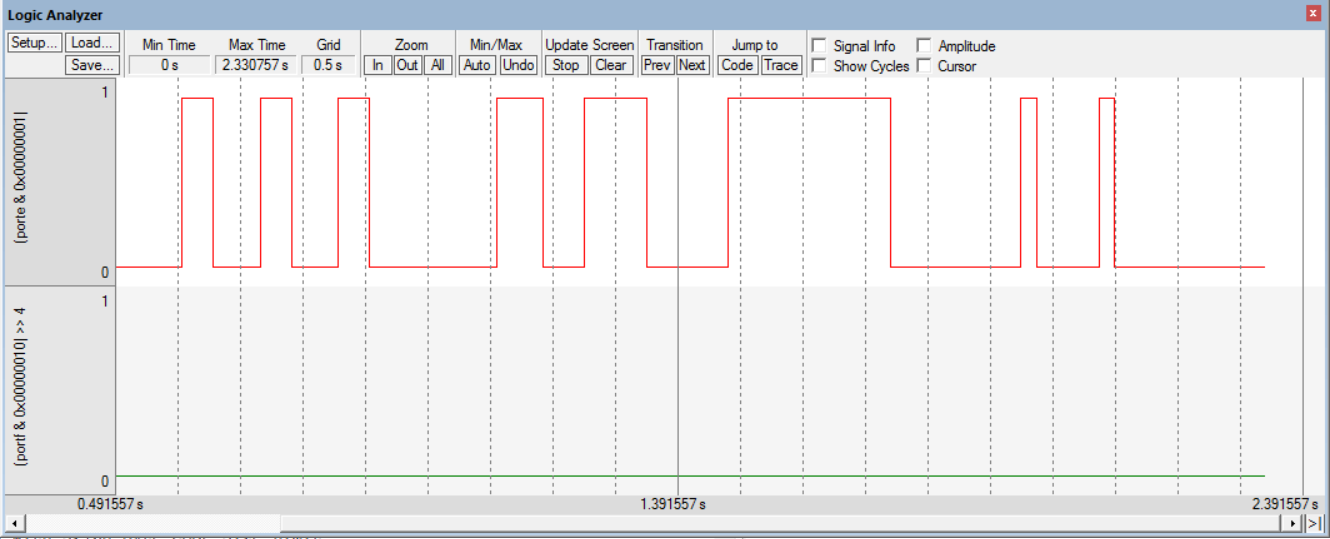
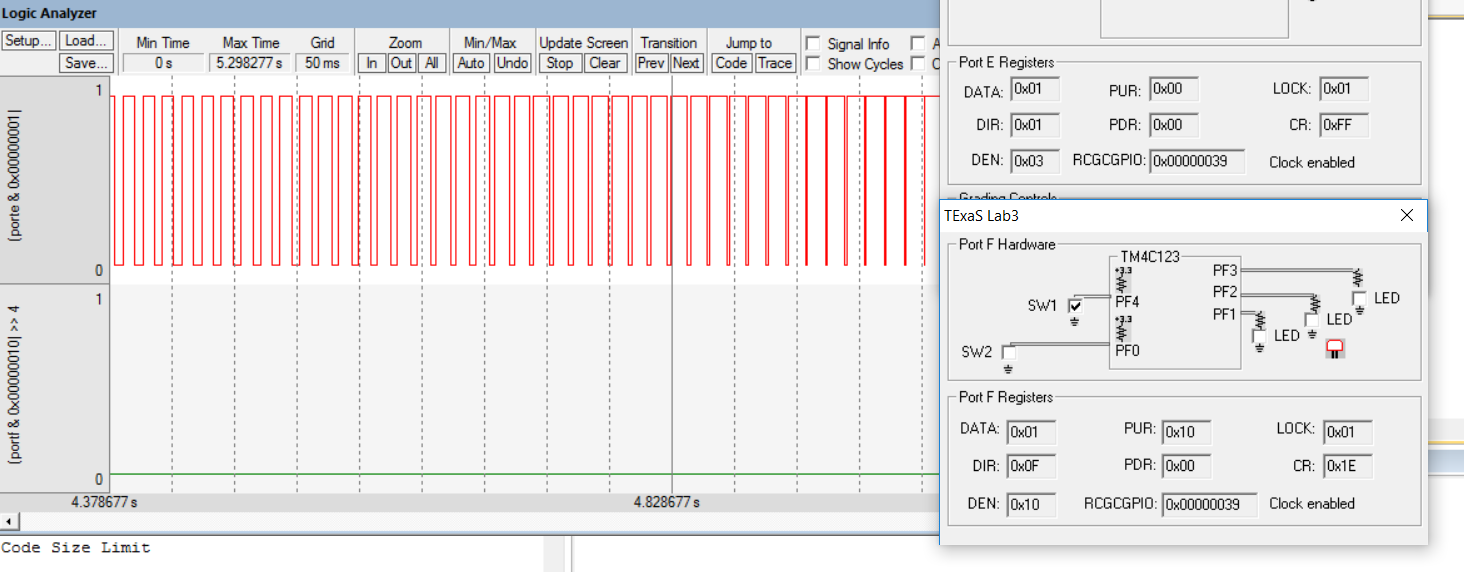
Daniel Canterino

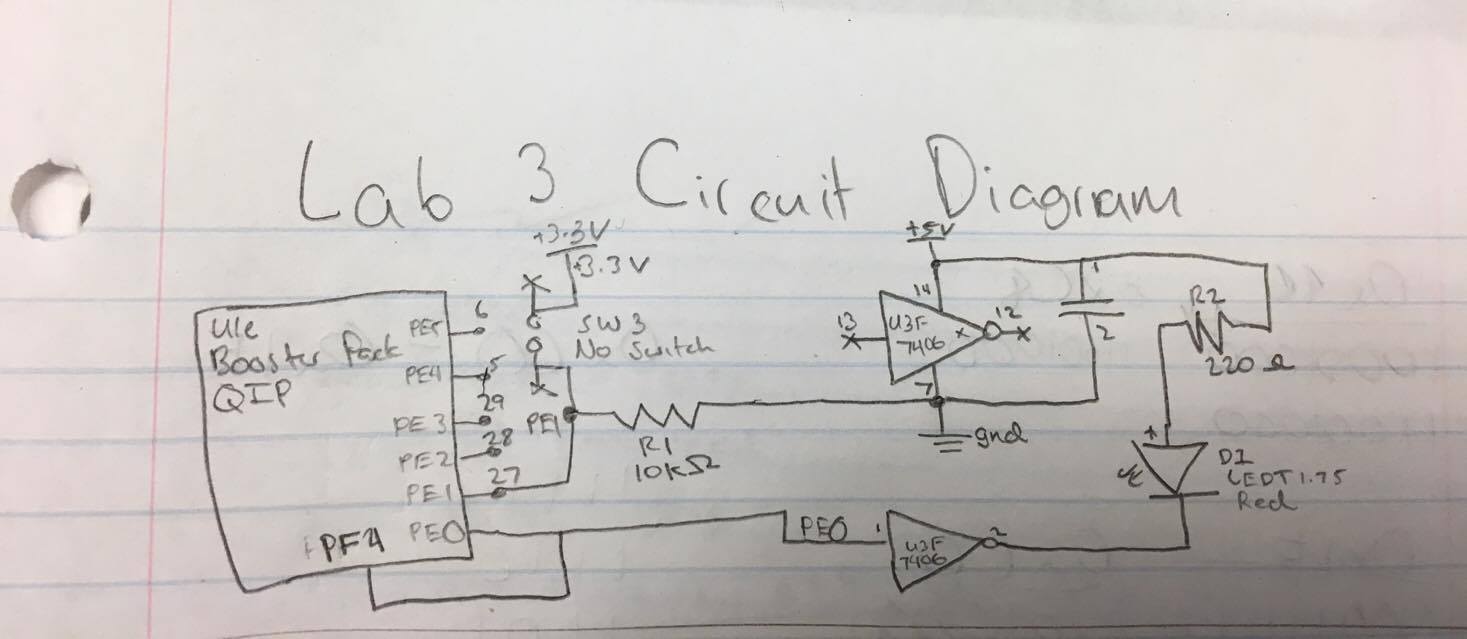
Pranav Padmanabha

This image shows the duty cycle increasing by 20% each time the button is pressed



This image shows the gradual increase/decrease that the led undergoes when breathing





**Switch Measurements**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value | Units | Conditions |
| Resistance of the    10kΩ resistor, R1 | 9.92k | ohms | with power off and    disconnected from circuit  (measured with ohmmeter) |
| Supply Voltage, V+3.3 | 3.24 | volts | Powered  (measured with voltmeter) |
| Input Voltage, VPE1 | 0 | volts | Powered, but  with switch not pressed  (measured with voltmeter) |
| Resistor current | 0  0.01 | mA | Powered, but switch not pressed      I=VPE1/R1 (calculated and  measured with an ammeter) |
| Input Voltage, VPE1 | 3.27 | volts | Powered and  with switch pressed  (measured with voltmeter) |
| Resistor current | 0.327  0.319 | mA | Powered and switch pressed      I=VPE1/R1 (calculated and  measured with an ammeter) |

**LED Measurements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Row | Parameter | Value | Units | Conditions |
| 1 | Resistance of the    220Ω resistor, R19 | 233 | ohms | with power off and    disconnected from circuit  (measured with ohmmeter) |
| 2 | +5 V power supply  *V+5* | 5.001 | volts | (measured with voltmeter relative to ground, *notice that the +5V power is not exactly +5 volts*) |
| 3 | TM4C123 Output, *VPE0*    input to 7406 | 1.2 | volts | with **PE0** = 0  (measured with voltmeter relative to ground) |
| s  4 | 7406 Output, *Vk-*    LED k- | 0.174 | volts | with **PE0** = 0  (measured with voltmeter relative to ground) |
| 5 | LED a+, *Va+*    Bottom side of R19 | 2.128 | volts | with **PE0** = 0  (measured with voltmeter relative to ground) |
| 6 | LED voltage | 1.95 | volts | calculated as *Va+*- *Vk-* |
| 7 | LED current | 0.0219  0.04 | mA | calculated as (*V+5*- *Va+*)/R19  and  measured with an ammeter |
| 8 | TM4C123 Output, *VPE0*    input to 7406 | 3.28 | volts | with **PE0** = 1  (measured with voltmeter relative to ground) |
| 9 | 7406 Output, *Vk-*    LED k- | 2.98 | volts | with **PE0** = 1  (measured with voltmeter relative to ground) |
| 10 | LED a+, *Va+*    Bottom side of R19 | 4.2(max) | volts | with **PE0** = 1  (measured with voltmeter relative to ground) |
| 11 | LED voltage | 1.22 | volts | calculated as *Va+*- *Vk-* |
| 12 | LED current | 3.6 | mA | calculated as (*V+5*- *Va+*)/R19  and  measured with an ammeter |
| 3.4 |

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

; Program written by: Daniel Canterino and Pranav Padmanabha

; Date Created: 2/4/2017

; Last Modified: 2/4/2017

; Brief description of the program

; The LED toggles at 8 Hz and a varying duty-cycle

; 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 external9 LED on protoboard)

; PF4 is builtin button SW1 on Launchpad (Internal)

; Negative Logic (0 means pressed, 1 means not pressed)

; Overall functionality of this system is to operate like this

; 1) Make PE0 an output and make PE1 and PF4 inputs.

; 2) The system starts with the the LED toggling at 8Hz,

; which is 8 times per second with a duty-cycle of 20%.

; Therefore, the LED is ON for (0.2\*1/8)th of a second

; and OFF for (0.8\*1/8)th of a second.

; 3) When the button on (PE1) is pressed-and-released increase

; the duty cycle by 20% (modulo 100%). Therefore for each

; press-and-release the duty cycle changes from 20% to 40% to 60%

; to 80% to 100%(ON) to 0%(Off) to 20% to 40% so on

; 4) Implement a "breathing LED" when SW1 (PF4) on the Launchpad is pressed:

; a) Be creative and play around with what "breathing" means.

; An example of "breathing" is most computers power LED in sleep mode

; (e.g., https://www.youtube.com/watch?v=ZT6siXyIjvQ).

; b) When (PF4) is released while in breathing mode, resume blinking at 8Hz.

; The duty cycle can either match the most recent duty-

; cycle or reset to 20%.

; TIP: debugging the breathing LED algorithm and feel on the simulator is impossible.

; PortE device registers

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

GPIO\_PORTF\_CR\_R EQU 0x40025524;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

SYSCTL\_RCGCGPIO\_R EQU 0x400FE608

PRESERVE8;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

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 ;ACTIVATE THE CLOCK FOR PORT F AND PORT E

LDR R0, [R1]

ORR R0, R0, #0x30 ;SET BIT 4 AND 5 TO TURN ON CLOCK

STR R0, [R1]

NOP

NOP

LDR R1, =GPIO\_PORTF\_DIR\_R ;SET PORT F PIN 4 AS INPUT

MOV R0, #0x00

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DIR\_R ;SET PORT E PIN 0 AS OUTPUT AND PIN 1 AS INPUT

MOV R0, #0x01

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_PUR\_R ;ENABLE PULL UP FOR PIN 4

MOV R0, #0x10

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_DEN\_R ;ENABLE PORT F DIGITAL PORT

MOV R0, #0x10

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DEN\_R ;ENABLE PORT E DIGITAL PORT

MOV R0, #0x03

STR R0, [R1]

; LDR R1, =GPIO\_PORTF\_CR\_R

; MOV R0, #0xFF

; STR R0, [R1]

MOV R5, #2

loop

CHECK MOV R2, #0

LDR R1, =GPIO\_PORTF\_DATA\_R ;checks to see if the switch on the board has been pressed

LDR R0, [R1]

MOV R2, R0

AND R2, R2, #0x10

CMP R2, #0

BEQ BREATH ;if pressed go to the breathing function

;;;;;;;;;;;;;;;CHECK PORT F;;;;;;;;;;;;;;;;;;

LDR R1, =GPIO\_PORTE\_DATA\_R ;check to see if the button has been pressed

LDR R0, [R1]

AND R2, R0, #0x02

CMP R2, #0x02

BEQ RELEASE ;if it has go to wait till its been released

BNE DET\_STATE ;if not pressed, execute the current state

B loop

BREATH BL BREATH\_WORK ;do the breath function

B CHECK ;once done, go back and check to see if new buttons have been pressed

RELEASE MOV R2, #0 ;stay here till button has been released

LDR R1, =GPIO\_PORTE\_DATA\_R

LDR R0, [R1]

AND R2, R0, #0x02

CMP R2, #0x02

BEQ RELEASE

BL ADD\_STATE ;once released, go to the next state

ADD\_STATE ADD R5, R5, #2 ;sets next state

DET\_STATE CMP R5, #12 ;this function will check to see what state should be executed. once the state is identified, it is executed

BEQ ZERO

CMP R5, #0

BEQ ZERO

CMP R5, #2

BEQ TWENTY

CMP R5, #4

BEQ FOURTY

CMP R5, #6

BEQ SIXTY

CMP R5, #8

BEQ EIGHTY

BL HUNDY

ZERO MOV R5, #0 ;zero percent duty cycle

BL TURN\_OFF

BL CHECK

;;;;;;;;;;;; R6 WILL BE COUNTER FOR ON R7 WILL BE INVERSE;;;;;;;;;;;;;;R7=100-R6

TWENTY MOV R5, #2 ;twenty percent duty cycle

BL TURN\_ON

MOV R6, #20

MOV R7, #80

TWENTY\_D BL DELAY

SUB R6, R6, #1

CMP R6, #0

BNE TWENTY\_D

BL TURN\_OFF

TWENTY\_O BL DELAY

SUB R7, R7, #1

CMP R7, #0

BNE TWENTY\_O

BL CHECK

FOURTY MOV R5, #4 ;fourty percent duty cycle

BL TURN\_ON

MOV R6, #40

MOV R7, #60

FOURTY\_D BL DELAY

SUB R6, R6, #1

CMP R6, #0

BNE FOURTY\_D

BL TURN\_OFF

FOURTY\_O BL DELAY

SUB R7, R7, #1

CMP R7, #0

BNE FOURTY\_O

BL CHECK

SIXTY MOV R5, #6 ;sixty percent duty cycle

BL TURN\_ON

MOV R6, #60

MOV R7, #40

SIXTY\_D BL DELAY

SUB R6, R6, #1

CMP R6, #0

BNE SIXTY\_D

BL TURN\_OFF

SIXTY\_O BL DELAY

SUB R7, R7, #1

CMP R7, #0

BNE SIXTY\_O

BL CHECK

EIGHTY MOV R5, #8 ;eighty percent duty cycle

BL TURN\_ON

MOV R6, #80

MOV R7, #20

EIGHTY\_D BL DELAY

SUB R6, R6, #1

CMP R6, #0

BNE EIGHTY\_D

BL TURN\_OFF

EIGHTY\_O BL DELAY

SUB R7, R7, #1

CMP R7, #0

BNE EIGHTY\_O

BL CHECK

HUNDY MOV R5, #10 ;hundred percent duty cycle

BL TURN\_ON

BL CHECK

TURN\_ON LDR R1, =GPIO\_PORTE\_DATA\_R ;turns on LED

LDR R0, [R1]

ORR R0, #0x01

STR R0, [R1]

BX LR

TURN\_OFF LDR R1, =GPIO\_PORTE\_DATA\_R ;turns off LED

LDR R0, [R1]

BIC R0, #0x01

STR R0, [R1]

BX LR

DELAY MOV R0, #20000 ;ASSUMING THAT THIS DELAY IS WILL BE FOR 1.25 MILLI SECOND = 1 percent of an 1/8th of a second

WAIT SUBS R0, R0, #1

CMP R0, #0

BNE WAIT

BX LR

;;;;;;;;;;;;;;;;; R6 HAS THE OFF DELAY TIME;;;;;;;;; R7 HAS THE ON DELAY TIME

;;;;;This subroutine will increase the duty cycle everytime providing a increased glow until 100 percent duty cycle. Then it does the reverse.

BREATH\_WORK PUSH{LR}

MOV R6, #100

MOV R7, #0

BACK MOV R8, R6;;;;;;R8 NOW HAS VALUE EQUIVALENT TO R6

MOV R9, R7;;;;;;R9 NOW HAS VALUE EQUIVALENT TO R7

BL TURN\_OFF

BREATH\_OFF BL DELAY\_B

SUB R8, R8, #1

CMP R8, #0

BNE BREATH\_OFF

BL TURN\_ON

BREATH\_ON BL DELAY\_B

SUB R9, R9, #1

CMP R9, #-1

BNE BREATH\_ON

SUB R6, R6, #1

ADD R7, R7, #1

CMP R6, #0

BNE BACK

MOV R6, #100

MOV R7, #0

BACK\_2 MOV R8, R6

MOV R9, R7

BL TURN\_ON

BREATH\_ON\_2 BL DELAY\_B

SUB R8, R8, #1

CMP R8, #0

BNE BREATH\_ON\_2

BL TURN\_OFF

BREATH\_OF\_2 BL DELAY\_B

SUB R9, R9, #1

CMP R9, #-1

BNE BREATH\_OF\_2

SUB R6, R6, #1

ADD R7, R7, #1

CMP R6, #0

BNE BACK\_2

POP {LR}

BX LR

DELAY\_B MOV R0, #2500 ;this is decreased delay for the breathing to make the transitions smoother.

WAIT\_B SUBS R0, R0, #1

CMP R0, #0

BNE WAIT\_B

BX LR

ALIGN ; make sure the end of this section is aligned

END ; end of file