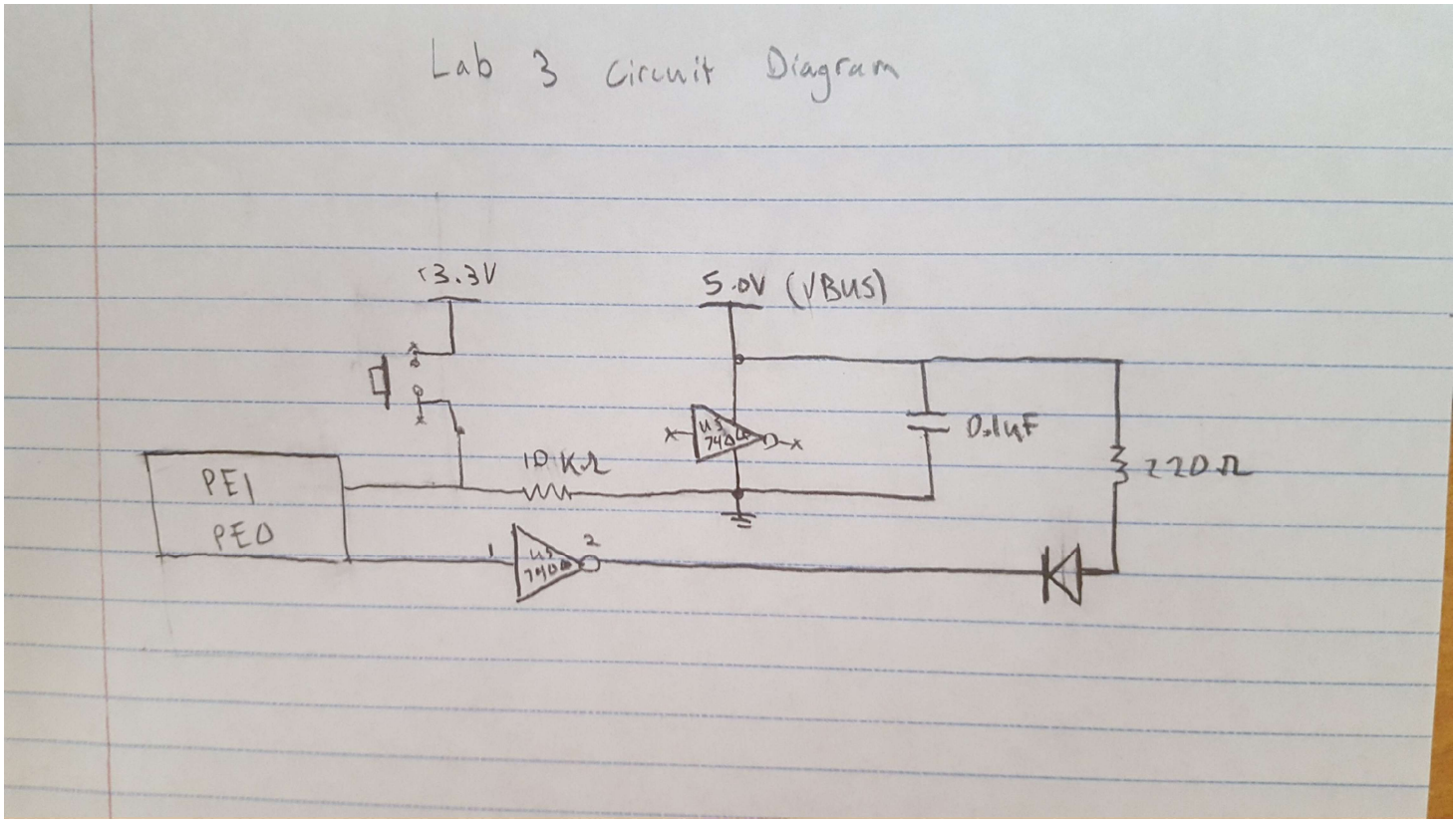
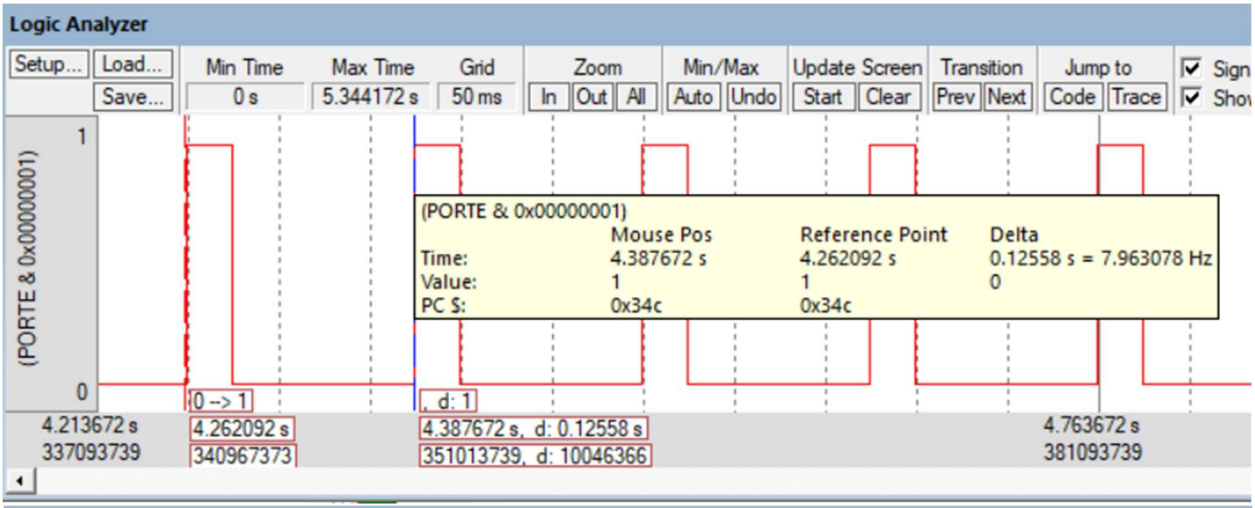
```
1       ;****************** main.s ***************
2       ; Program written by: Matteus Oosterlaken & Ryan Taylor
3       ; Date Created: 2/4/2017
4       ; Last Modified: 2/14/2018
5       ; Brief description of the program
6       ;   The LED toggles at 8 Hz and a varying duty-cycle
7       ; Hardware connections (External: One button and one LED)
8       ;  PE1 is Button input  (1 means pressed, 0 means not pressed)
9       ;  PE0 is LED output (1 activates external LED on protoboard)
10      ;  PF4 is builtin button SW1 on Launchpad (Internal)
11      ;       Negative Logic (0 means pressed, 1 means not pressed)
12      ; Overall functionality of this system is to operate like this
13      ;   1) Make PE0 an output and make PE1 and PF4 inputs.
14      ;   2) The system starts with the the LED toggling at 8Hz, 15   ;      which is 8 times per second with a duty-cycle of 20%. 16   ;      Therefore, the LED is
        ON for (0.2*1/8)th of a second
17      ;      and OFF for (0.8*1/8)th of a second.
18      ;   3) When the button on (PE1) is pressed-and-released increase
19      ;      the duty cycle by 20% (modulo 100%). Therefore for each
20      ;      press-and-release the duty cycle changes from 20% to 40% to 60%
21      ;      to 80% to 100%(ON) to 0%(Off) to 20% to 40% so on
22      ;   4) Implement a "breathing LED" when SW1 (PF4) on the Launchpad is pressed:
23      ;      a) Be creative and play around with what "breathing" means.
24      ;         An example of "breathing" is most computers power LED in sleep mode
25      ;         (e.g., https://www.youtube.com/watch?v=ZT6siXyIjvQ).
26      ;      b) When (PF4) is released while in breathing mode, resume blinking at 8Hz. 27   ;         The duty cycle can either match the most recent duty-
28      ;         cycle or reset to 20%.
29      ;      TIP: debugging the breathing LED algorithm and feel on the simulator is impossible. 30   ; PortE device registers
31      GPIO_PORTE_DATA_R EQU 0x400243FC
32      GPIO_PORTE_DIR_R        EQU 0x40024400
33      GPIO_PORTE_AFSEL_R EQU 0x40024420
34      GPIO_PORTE_DEN_R        EQU 0x4002451C
35      ; PortF device registers
36      GPIO_PORTF_DATA_R EQU 0x400253FC
37      GPIO_PORTF_DIR_R          EQU 0x40025400
38      GPIO_PORTF_AFSEL_R EQU 0x40025420
39      GPIO_PORTF_PUR_R          EQU 0x40025510
40      GPIO_PORTF_DEN_R          EQU 0x4002551C
41      GPIO_PORTF_LOCK_R EQU 0x40025520
42      GPIO_PORTF_CR_REQU 0x40025524
43      GPIO_LOCK_KEY   EQU 0x4C4F434B                  ; Unlocks the GPIO_CR register
44      SYSCTL_RCGCGPIO_R EQU 0x400FE608
45  EIGTH_SECOND            EQU 2985833 ;2500000 for simulator
46
47
48
49                  IMPORT TExaS_Init
50      THUMB 51          AREA    DATA, ALIGN=2
52
53  ;global variables go here
54
55      AREA    |.text|, CODE, READONLY, ALIGN=2 56          THUMB
57                  EXPORT Start
58
59
60  Start
61
62          BL TExaS_Init           ; voltmeter, scope on PD3
63          BL LED_Init   ; initializes clock, DEN, DIR
64
65                      Increment RN 4          ; A constant value of delay that will be used to increment
                        the
        duty, its 20% duty and this 1/40th of a second
```

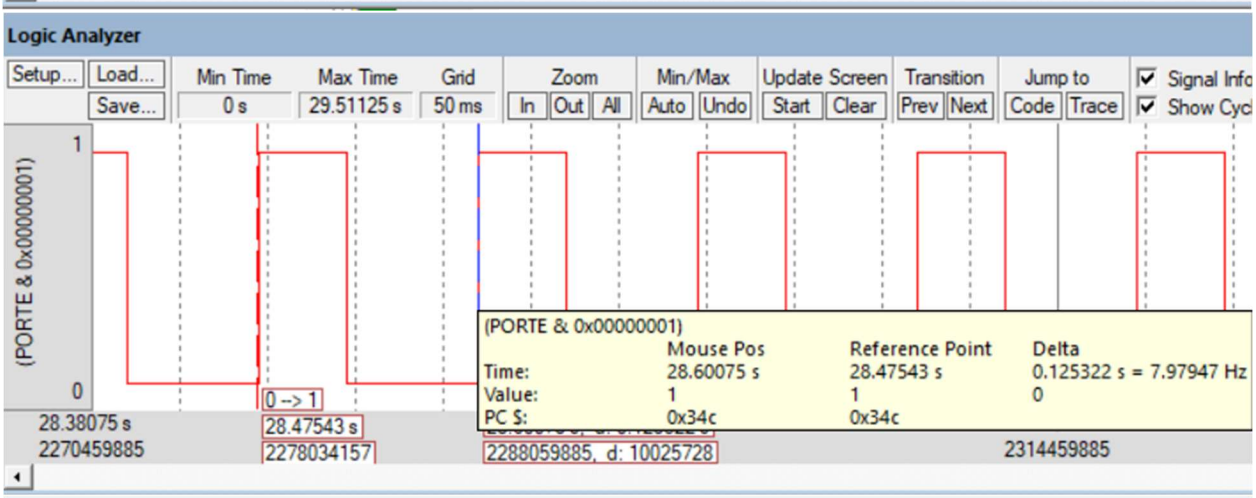
```
66                      Duty        RN 5      ; keeps track of the percent duty
67                      offDuty RN 6          ; keeps track of the opposite perccent duty.
68                      High RN 7
69                      Low RN 8
70                      LDR High,= 10000
71                      LDR Low,= 90000
72                      LDR Duty,= 0x00
73                      LDR offDuty,= EIGTH_SECOND
74                      MOV R0, #0x05
75                      SDIV Increment, offDuty, R0
76
77
78          loop ; the Main loop
79          BL Change_Duty          ; checks if the switch PE1 has been pressed and increments
        accordingly
80          BL Breathing  ; checks if Switch PF0 has been pressed and causes LED to
        breathe
81          BL Check_Duty           ; checks if the Duty is more than 100 or 0 and acts accordingly 82      MOV R1, #0x01
83          BL LED_Out  ; Turn on LED
84          CMP Duty, R1 85         BEQ loop
86          MOV R0, Duty
87          BL Delay        ; activates a delay of the duty
88          MOV R1, #0x00
89          BL LED_Out  ; Turn off LED
90          MOV R0, offDuty
91          BL Delay        ; activates a delay of the off duty
92          B loop
93
94  Breathing       ; led "breathes" with a period of .0125 seconds   95            MOV R10, LR
96              LDR R0, = GPIO_PORTF_DATA_R
97              LDR R1, [R0]
98              ANDS R1, #0x10        ; check if switch 4 is pressed
99              BNE Not_Breathing     ; proceed with main loop if not pressed
100             LDR R2,= 752
101             ADD High, High, R2    ; increment between duties at 8hz
102              LDR R3,= 100000
103         CMP High, R3              ; High 100000 = 100%duty
104         BLO Breathing_up          ; branch if 100% duty is reached
105         Breathing_b
106         LDR R2,= 752
107           LDR R3,= 100000
108     ADD Low, Low, R2              ; checks if to switch from breathing down to breathing up 109          CMP Low, R3
110         BLO Breathing_down
111         B Breathing_up
112         Breathing_down            ;light dims
113         SUB High, R3, Low
114         MOV R1, #0x01
115         BL LED_Out
116         MOV R0, High
117         BL Delay
118         MOV R1, #0x00
119         BL LED_Out
120         MOV R0, Low
121         BL Delay
122         B Breathing_b
123         Breathing_up  ; light brightens
124         SUB Low, R3, High
125         MOV R1, #0x01
126         BL LED_Out
127         MOV R0, High
128         BL Delay
129         MOV R1, #0x00
```

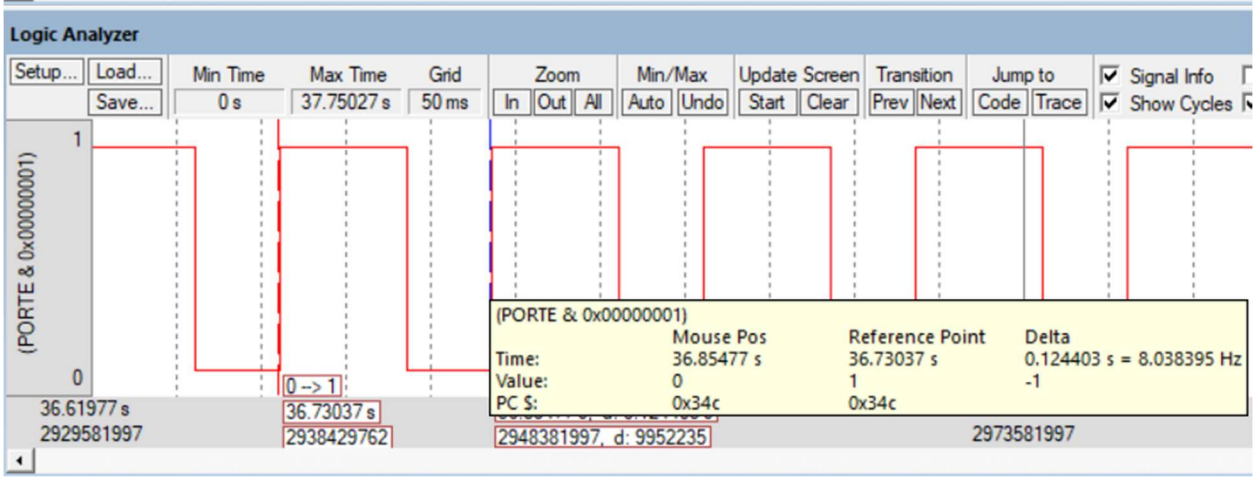
```
130             BL LED_Out
131             MOV R0, Low 132         BL Delay
133         MOV LR, R10 134
B Breathing
135  Not_Breathing
136              MOV LR, R10
137             BX LR
138
139             LED_Out       ; Subroutine: changes the value of the LED according to the
        value of R1 (Uses R1, Clears R0, R1, R2)
140             LDR R0, =GPIO_PORTE_DATA_R          ; toggles PE0
141             LDR R2, [R0]
142             BIC R2, #0x01
143             ORR R2, R1
144             STR R2, [R0]
145             LDR R1,= EIGTH_SECOND
146             BX LR
147
148             Delay           ; Subroutine: delay of the value loaded into R0 (Uses R0,
        Clears R0)
149             SUBS R0, R0, #0x01
150             BPL Delay
151             BX LR
152
153             Change_Duty  ; Subroutine: checks if the switch PE1 has been pressed and
        increments accordingly (Clears R0, R1, R2)
154             LDR R0, = GPIO_PORTE_DATA_R
155             LDR R1, [R0]
156             ANDS R1, #0x02
157             BEQ no_change
158             Switch_Release            ; checks for switch release
159             LDR R0, = GPIO_PORTE_DATA_R
160             LDR R1, [R0]
161             ANDS R1, #0x02
162             BNE Switch_Release
163             MOV R1, Duty
164             MOV R0, Increment
165             MOV R2, offDuty
166             ADD R1, R0   ; Duty gets incremented by 20%
167             SUBS R2, R0  ; offDuty gets decremented by 20%
168             MOV Duty, R1
169             MOV offDuty, R2
170             no_change
171             BX LR
172
173             Check_Duty    ; Subroutine: checks if the Duty is more than 100 or 0 and acts
        accordingly (Clears R1)
174             CMP Duty, #0x00         ; if duty = 0 it leaves it off
175             BEQ loop
176             LDR R1,= EIGTH_SECOND
177             CMP R1, Duty
178             BGE notnegative          ; if duty is greater than 100% then reset duty and offduty 179             MOV Duty, #0x00
180             LDR offDuty, =EIGTH_SECOND
181             notnegative
182             BX LR
183
184             LED_Init       ; Subroutine: initializes clock, DEN, DIR (Clears R0, R1)
185             LDR R0, =SYSCTL_RCGCGPIO_R          ; initialize clock
186             LDR R1, [R0]
187             ORR R1, R1, #0x30
188             STR R1, [R0]
189             NOP
```

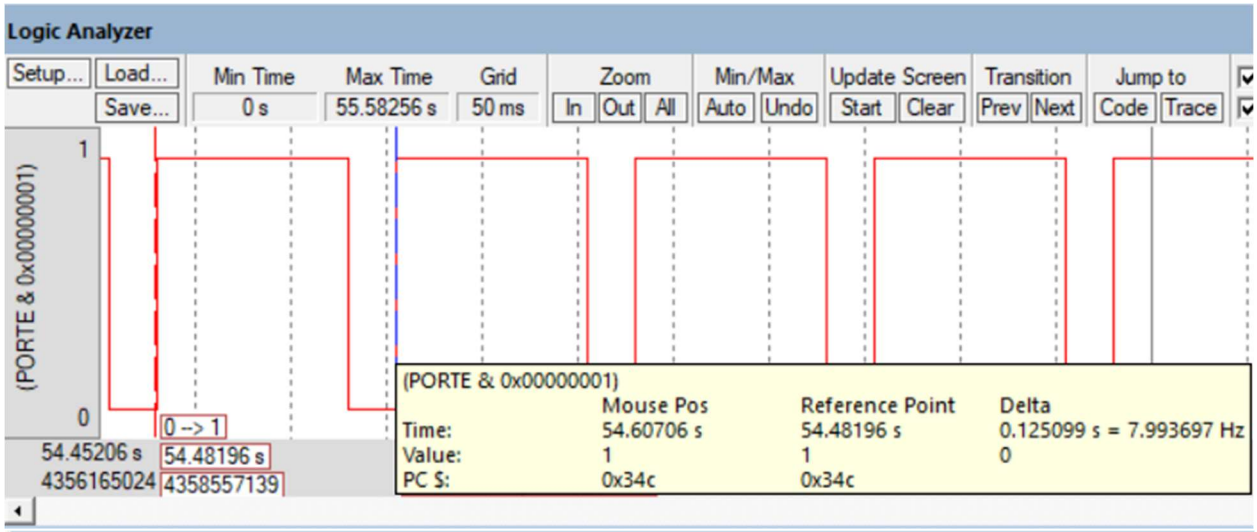
```
190         NOP
191         LDR R0, =GPIO_PORTE_DIR_R ; allow Output direction for port PE0
192         LDR R1, [R0]
193         ORR R1, R1, #0x01
194         STR R1, [R0]
195         LDR R0, =GPIO_PORTE_DEN_R       ; Digital enable for PE0 and PE1
196         LDR R1, [R0]
197         ORR R1, R1, #0x03
198         STR R1, [R0]
199         LDR R0, =GPIO_PORTF_LOCK_R      ; unlock
200           LDR R1, =0x4C4F434B
201         STR R1, [R0]
202         LDR R0, =GPIO_PORTF_CR_R
203         ORR R1, #0x1F
204         STR R1, [R0]
205         LDR R0, =GPIO_PORTF_PUR_R
206         ORR R1, #0x10
207         STR R1, [R0]
208         LDR R0, =GPIO_PORTF_DEN_R       ; Digital enable for PF4 209   LDR R1, [R0]
210         ORR R1, R1, #0xFF
211         STR R1, [R0]
212         CPSIE I         ; TExaS voltmeter, scope runs on interrupts
213         BX LR
214
215
216           ALIGN         ; make sure the end of this section is aligned
217           END           ; end of file
218
219
220
221
```

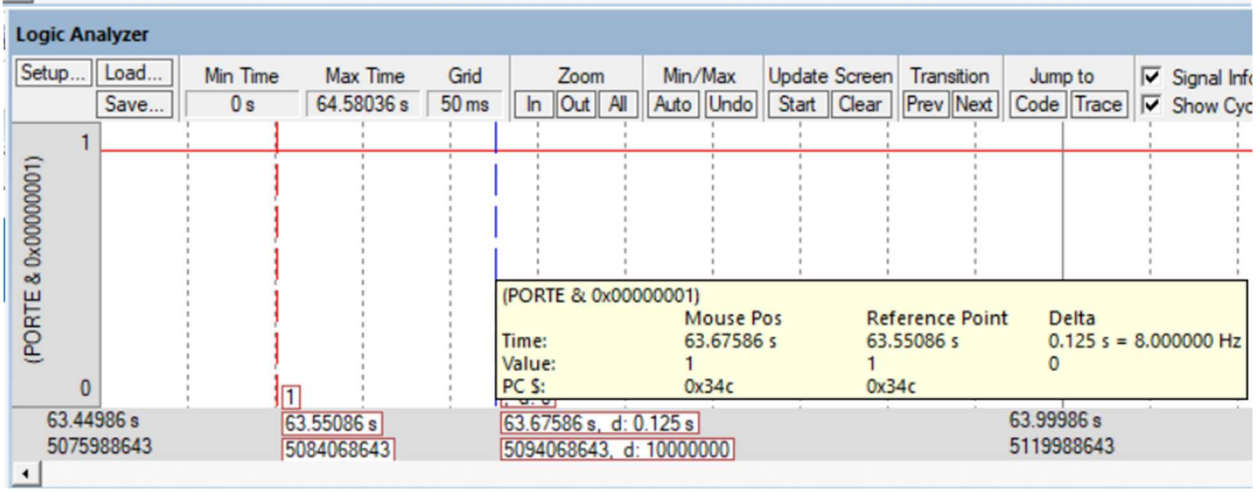



Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 5.344172 s
Grid 50 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Start Clear
Transition Prev Next
Jump to Code Trace
(PORTE & 0x00000001)
Mouse Pos Reference Point Delta
Time: 4.387672 s 4.262092 s 0.12558 s = 7.963078 Hz
Value: 1 1 0
PC $: 0x34c 0x34c
0 --> 1
, d: 1
4.213672 s
4.262092 s
4.387672 s, d: 0.12558 s
4.763672 s
337093739
340967373
351013739, d: 10046366
381093739


Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 29.51125 s
Grid 50 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Start Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Show Cycles
(PORTE & 0x00000001)
Mouse Pos Reference Point Delta
Time: 28.60075 s 28.47543 s 0.125322 s = 7.97947 Hz
Value: 1 1 0
PC $: 0x34c 0x34c
0 --> 1
28.38075 s
28.47543 s
2270459885
2278034157
2288059885, d: 10025728
2314459885


Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 37.75027 s
Grid 50 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Start Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Show Cycles
(PORTE & 0x00000001)
Mouse Pos Reference Point Delta
Time: 36.85477 s 36.73037 s 0.124403 s = 8.038395 Hz
Value: 0 1 -1
PC $: 0x34c 0x34c
0 --> 1
36.61977 s
36.73037 s
2929581997
2938429762
2948381997, d: 9952235
2973581997


Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
55.58256 s
Grid
50 ms
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Start
Clear
Transition
Prev
Next
Jump to
Code
Trace
(PORTE & 0x00000001)
1
0
0 --> 1
54.45206 s
54.48196 s
4356165024
4358557139
(PORTE & 0x00000001)
Mouse Pos
Reference Point
Delta
Time:
54.60706 s
54.48196 s
0.125099 s = 7.993697 Hz
Value:
1
1
0
PC $:
0x34c
0x34c


Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
64.58036 s
Grid
50 ms
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Start
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Inf
Show Cyc
(PORTE & 0x00000001)
1
0
1
63.44986 s
63.55086 s
63.67586 s, d: 0.125 s
63.99986 s
5075988643
5084068643
5094068643, d: 10000000
5119988643
(PORTE & 0x00000001)
Mouse Pos
Reference Point
Delta
Time:
63.67586 s
63.55086 s
0.125 s = 8.000000 Hz
Value:
1
1
0
PC $:
0x34c
0x34c

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1	9.79k	ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.29	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	0	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current	0	mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3.28	volts	Powered and with switch pressed (measured with voltmeter)
Resistor current	.329	mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R19	215	ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}	5.01	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	.09	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
4	7406 Output, V_{k-} LED k-	3.37	volts	with **PE0** = 0 (measured with voltmeter relative to ground)

5	LED a+, V_{a+} Bottom side of R19 (anode side of LED)	4.98	volts	with **PE0** = 0 (measured with voltmeter relative to ground)
6	LED voltage	1.61	volts	calculated as V_{a+} - V_{k-}
7	LED current	-7.3	mA	calculated as (V_{+5} - V_{a+})/R19 and measured with an ammeter
8	TM4C123 Output, V_{PE0} input to 7406	3.29	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
9	7406 Output, V_{k-} LED k-	.13	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
10	LED a+, V_{a+} Bottom side of R19 (anode side of LED)	2.25	volts	with **PE0** = 1 (measured with voltmeter relative to ground)
11	LED voltage	1.61	volts	calculated as V_{a+} - V_{k-}
12	LED current	-7.3 -11	mA	calculated as (V_{+5} - V_{a+})/R19 and measured with an ammeter