```
;***************** main.s ***************
; Program written by: ***Your Names**update this***
; Date Created: 2/26/2017
; Last Modified: 2/14/2017
; 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
; 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_LOCK_R  EQU 0x40025520
GPIO_PORTF_CR_R    EQU 0x40025524
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_LOCK_KEY      EQU 0x4C4F434B  ; Unlocks the GPIO_CR register
SYSCTL_RCGCGPIO_R  EQU 0x400FE608
;
NVIC_ST_CURRENT_R  EQU 0xE000E018
       ;RAM AREA
          AREA DATA, ALIGN =1
Databuffer SPACE 54                             ;DATA ARRAY
Timebuffer SPACE 200                   ;TIME ARRAY
Tpt SPACE 4                            ;TIME ARRAY POINTER
Dpt SPACE 4                            ;DATA ARRAY POINTER

         IMPORT  TExaS_Init
```

```
        IMPORT  SysTick_Init
    AREA     |.text|, CODE, READONLY, ALIGN=2
    THUMB
     ;EXPORT  SysTick_Init
    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 Port F/E Clock
    LDR R0, [R1]
    ORR R0, R0, #0x30
    STR R0, [R1]                 ;Clock Initialize Time
    NOP
    NOP
    LDR R1, =GPIO_PORTF_LOCK_R      ;Unlock PortF Register
    LDR R0, =0x4C4F434B           ;MAYBE YES OR NO NOT SURE WHY
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_CR_R
    MOV R0, #0xFF
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AMSEL_R     ;Disable Analog
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DIR_R       ;Set Direction Register
    MOV R0, #0x04                ;PF2 Output
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AFSEL_R     ;Initialize PortF
    MOV R0, #0
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_AFSEL_R    ;Initialize PortE
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PUR_R       ;Pull Up Resistors
    MOV R0, #0x10
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DEN_R       ;Enable PortF
    MOV R0, #0xFF
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_DEN_R       ;Enable PortE
    MOV R0, #0xFF
    STR R0, [R1]
      LDR R1, =GPIO_PORTE_DIR_R       ;Set Direction Register
    MOV R0,#0x01                 ;0 Output, 1 Input
    STR R0, [R1]
      ;
      MOV R3, #1                              ; R3 CONTAINS COUNT FOR
DUTY MULTIPLIER
      MOV R4, #4                              ; R4 CONTAINS COUNT FOR
LEDOFF DUTY MULTIPLIER
      MOV R5, #25                             ; PUTS MULITPLE OF 25 IN R5
      MOV R12, #0
      ;R6 WILL CONTAIN 25*X IN LEDON
      ;R7 WILL CONTAIN 25*Y IN LEDOFF
```

```
      ;R0 AND R1 RESERVED FOR LOADING OF ADDRESS AND DATA
      ;
      ;
      ;
      BL Debug_Init
mainloop;-------------------------------------------------------------
-

      BL buttoncheck                        ;CHECKS PORTE1(BUTTON)
checkpoint
      BL LEDON                              ;TOGGLE LED ON
      BL dataEntry
      BL LEDOFF                             ;TOGGLE LED OFF
      BL dataEntry
      B mainloop                            ;REPEAT
      ;
      ;
      ;
LEDON;-----------------------------------------------------------------
--
      CMP R3, #0
      BEQ next
      LDR R1, =GPIO_PORTE_DATA_R   ;Load PortE Data Address
      LDR R0, [R1]                               ;Loading PortE Data
      MOV R0, #0x01                             ;Toggling LED PE0
      STR R0, [R1]                              ;Storing Result back in
PortE
      LDR R1, =GPIO_PORTF_DATA_R        ; LOAD PORTE DATA ADDRESS
      LDR R0, [R1]
      MOV R0, #0x04                             ; ON
      STR R0, [R1]
      MUL R6, R3, R5                            ; NUMBER OF MILISECS, R5
HAS 25, R3 HAS NUMBER OF TIMES TO MULTIPLY BY, PUT IN R6
LOOP
      MOV R2, #19990                           ; COUNT
delay
      SUBS R2, R2, #1                          ; R2 CONTAINS A NUMBER TO
GET UP TO 1MS (16000)
      BNE delay
      SUBS R6, R6, #1                          ; R6 CONTAINS NUMBER OF MS
COUNT
      BNE LOOP
next
      BX LR
      ;
      ;
      ;
LEDOFF;-------------------------------------------------------------
      CMP R4, #0
      BEQ next1
      LDR R1, =GPIO_PORTE_DATA_R        ; LOAD PORTE DATA ADDRESS
      LDR R0, [R1]
      MOV R0, #0x00                            ; OFF
      STR R0, [R1]
```

```
      LDR R1, =GPIO_PORTF_DATA_R          ; LOAD PORTF DATA ADDRESS
      LDR R0, [R1]
      MOV R0, #0x00                       ; OFF
      STR R0, [R1]
      MUL R7, R4, R5                      ;# OF MILISECS, R5 HAS 25
AND, R4 HAS NUMBER OF TIMES TO MULTIPLY BY, PUT IN R7
LOOP1
      MOV R2, #20000                      ; COUNT
delay1
      SUBS R2, R2, #1                     ; R2 CONTAINS A NUMBER TO
GET UP TO 1MS (16000)
      BNE delay1
      SUBS R7, R7, #1                     ; R4 CONTAINS NUMBER OF OFF
MS COUNT
      BNE LOOP1
next1
      BX LR
      ;
      ;
      ;
buttoncheck;------------------------------------------------------------
      LDR R1, =GPIO_PORTE_DATA_R
      LDR R0, [R1]
      LSR R0, #1
      EOR R0, #0x01
      CMP R0, #0                          ;CHECKS IF PE1 IS PRESSED
      BEQ check
      BX LR                               ;BACK TO MAINLOOP
check
      PUSH {R12, LR}                      ;WHEN BUTTON IS PRESSED
INCREASES R3 COUNT AND DECREASES R4
      MOV R12, #1                         ;Setting dataEntry to run
once after button release
      BL dataEntry
here
      LDR R1, =GPIO_PORTE_DATA_R
      LDR R0, [R1]
      LSR R0, #1
      EOR R0, #0x01
      CMP R0, #0
      BNE done
      BL LEDON
      BL LEDOFF
      B  here
done
      POP {R12, LR}
      ADD R3, #1
      CMP R3, #6                          ;SEES IF BUTTON HAS BEEN
PRESSED 5 TIMES (6TH TIME SHOULD SET TO
      BNE change2
      AND R3, #0
change2
      MOV R12, #7                         ;Setting dataEntry to run 7
times after button release
```

```
      SUBS R4, R4 , #1
      CMP  R4, #0
      BPL back                                ;IF POSITIVE OR ZERO JUMP,
IF NEGATIVE WANT TO ADD 5 TO GET TO 4(RESET)
      ADD R4, #6
back
      BX LR                                   ;TO MAIN LOOP
      ;
      ;
      ;
Debug_Init;-------------------------------------------------------------
---
      PUSH {R12,LR}
      LDR R1, =Databuffer
      MOV R0, #0xFF                           ; NO STATE SAVED YET
      STR R0, [R1]
      LDR R1, =Timebuffer
      MOV R0, #0xFFFFFFFF                      ; NO TIMING SAVED YET
      STR R0, [R1]
      LDR R1, =Databuffer                     ;INITIALIZATION Data Array
Pointer
      LDR R0, =Dpt
      STR R1, [R0]
      LDR R1, =Timebuffer                     ;INITIALIZATION Time Array
Pointer
      LDR R0, =Tpt
      STR R1, [R0]
      BL SysTick_Init                         ;TO SYSTIC SR
      MOV R6, #50
      MOV R2, #0xFFFFFFFF
      LDR R1,=Tpt
      LDR R0, [R1]
L1
      CMP R6, #0
      BEQ fin1
      STR R2, [R0]
      ADD R0, #4
      SUBS R6, #1
      B L1
fin1
      MOV R6, #50
      MOV R2, #0xFF
      LDR R1, =Dpt
      LDR R0, [R1]
L2
      CMP R6, #0
      BEQ fin2
      STRB R2, [R0]
      ADD R0, #1
      SUBS R6, #1
      B L2
fin2
      POP {R12, LR}
      BX LR                                   ;TO MAIN LOOP
```

```
        ;
        ;
        ;
dataEntry ;DEBUG capture----------------------------------
      CMP R12, #0                              ;R12 = amount of times
dataEntry should run w/out reset (7 or 1)
      BEQ back
      SUBS R12, #1
      CMP R11, #50                            ;R11 = index counter (if 50
then done)
      BEQ back
      ADD R11, #1
      LDR R1, =GPIO_PORTE_DATA_R   ;Loading PortE Data
      LDR R0, [R1]
      MOV R2, R0                              ;Data Manipulation
      AND R2, #0x02
      LSL R2, #3
      AND R0, #0x01
      ORR R2, R2, R0                          ;Final Value
      LDR R1, =Dpt                            ;Storing in the Array
R1=DPOINTER
      LDR R0, [R1]                            ;ADDRESS IN R0
      STR R2, [R0]                            ;VALUE IN R2
      ADD R0, #1                              ;1 BECAUSE 8BIT (should it
be 4 because register address is 32bit)
      STR R0, [R1]                            ;
      LDR R1, =NVIC_ST_CURRENT_R        ;systick timer register
      LDR R2, [R1]                            ;value into R2
      LDR R1, =Tpt                            ;Storing in the Array
R1=TPOINTER
      LDR R0, [R1]                            ;ADDRESS IN R0
      STR R2, [R0]                            ;VALUE IN R2
      ADD R0, #4                              ;4 BECAUSE 32BIT
      STR R0, [R1]                            ;Fin
      BX LR                                   ;TO CHECK LOOP IN DATAENTRY

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

Memory 1

Address	Value
0x20000030:	10
0x20000031:	01
0x20000032:	00
0x20000033:	01
0x20000034:	00
0x20000035:	01
0x20000036:	00
0x20000037:	01
0x20000038:	10
0x20000039:	01
0x2000003A:	00
0x2000003B:	01
0x2000003C:	00
0x2000003D:	01
0x2000003E:	00
0x2000003F:	01
0x20000040:	10
0x20000041:	01
0x20000042:	00
0x20000043:	01
0x20000044:	00
0x20000045:	01
0x20000046:	00
0x20000047:	01
0x20000048:	10
0x20000049:	01
0x2000004A:	01
0x2000004B:	01
0x2000004C:	01
0x2000004D:	01
0x2000004E:	01
0x2000004F:	01
0x20000050:	11
0x20000051:	01
0x20000052:	00
0x20000053:	00
0x20000054:	00
0x20000055:	00
0x20000056:	00
0x20000057:	00
0x20000058:	10
0x20000059:	01
0x2000005A:	00
0x2000005B:	01
0x2000005C:	00
0x2000005D:	01
0x2000005E:	00
0x2000005F:	01
0x20000060:	10
0x20000061:	01

Memory 1

Address	Value
0x20000030:	10
0x20000031:	01
0x20000032:	00
0x20000033:	01
0x20000034:	00
0x20000035:	01
0x20000036:	00
0x20000037:	01
0x20000038:	10
0x20000039:	01
0x2000003A:	00
0x2000003B:	01
0x2000003C:	00
0x2000003D:	01
0x2000003E:	00
0x2000003F:	01
0x20000040:	00
0x20000041:	00
0x20000042:	00
0x20000043:	FF
0x20000044:	FF
0x20000045:	FF
0x20000046:	FF
0x20000047:	FF
0x20000048:	FF
0x20000049:	FF
0x2000004A:	FF
0x2000004B:	FF
0x2000004C:	FF
0x2000004D:	FF
0x2000004E:	FF
0x2000004F:	FF
0x20000050:	FF
0x20000051:	FF
0x20000052:	FF
0x20000053:	FF
0x20000054:	FF
0x20000055:	FF
0x20000056:	FF
0x20000057:	FF
0x20000058:	FF
0x20000059:	FF
0x2000005A:	FF
0x2000005B:	FF
0x2000005C:	FF
0x2000005D:	FF
0x2000005E:	FF
0x2000005F:	FF
0x20000060:	FF
0x20000061:	FF

Memory 2 (left panel)

Address: 0x20000066

```
0x20000066:  0010332800
0x2000006A:  0009890700
0x2000006E:  0003890405
0x20000072:  0016669383
0x20000076:  0010669089
0x2000007A:  0006670851
0x2000007E:  0000670556
0x20000082:  0013449534
0x20000086:  0001009826
0x2000008A:  0005350439
0x2000008E:  0001350219
0x20000092:  0012130122
0x20000096:  0008129903
0x2000009A:  0002132590
0x2000009E:  0014909586
0x200000A2:  0008912273
0x200000A6:  4294967295
0x200000AA:  4294967295
0x200000AE:  4294967295
0x200000B2:  4294967295
0x200000B6:  4294967295
0x200000BA:  4294967295
0x200000BE:  4294967295
0x200000C2:  4294967295
0x200000C6:  4294967295
0x200000CA:  4294967295
0x200000CE:  4294967295
0x200000D2:  4294967295
0x200000D6:  4294967295
0x200000DA:  4294967295
0x200000DE:  4294967295
0x200000E2:  4294967295
0x200000E6:  4294967295
0x200000EA:  4294967295
0x200000EE:  4294967295
0x200000F2:  4294967295
0x200000F6:  4294967295
0x200000FA:  4294967295
0x200000FE:  4294967295
0x20000102:  4294967295
0x20000106:  4294967295
0x2000010A:  4294967295
0x2000010E:  4294967295
0x20000112:  4294967295
0x20000116:  4294967295
0x2000011A:  4294967295
0x2000011E:  4294967295
0x20000122:  4294967295
0x20000126:  4294967295
0x2000012A:  4294967295
```

UART #1 | Memor... | Memor...

Memory 2 (right panel)

Address: 0x20000066

```
0x20000066:  0007111108
0x2000006A:  0009890648
0x2000006E:  0003890353
0x20000072:  0016669331
0x20000076:  0010669037
0x2000007A:  0006670799
0x2000007E:  0000670504
0x20000082:  0013449482
0x20000086:  0007449158
0x2000008A:  0005010983
0x2000008E:  0001010763
0x20000092:  0011790666
0x20000096:  0007790447
0x2000009A:  0001793134
0x2000009E:  0014570130
0x200000A2:  0008572817
0x200000A6:  0004572568
0x200000AA:  0000138318
0x200000AE:  0014915389
0x200000B2:  0006919001
0x200000B6:  0004918857
0x200000BA:  0013699685
0x200000BE:  0011699540
0x200000C2:  0003703152
0x200000C6:  0012398188
0x200000CA:  0015964291
0x200000CE:  0015964232
0x200000D2:  0005968769
0x200000D6:  0005968711
0x200000DA:  0012750464
0x200000DE:  0012750405
0x200000E2:  0002754942
0x200000E6:  0016318400
0x200000EA:  0013104697
0x200000EE:  0003104252
0x200000F2:  0003104175
0x200000F6:  0009880947
0x200000FA:  0009880870
0x200000FE:  0016657641
0x20000102:  0016657564
0x20000106:  0003762853
0x2000010A:  0005316861
0x2000010E:  0014093707
0x20000112:  0012094544
0x20000116:  0004094175
0x2000011A:  0002095012
0x2000011E:  0010871858
0x20000122:  0008872695
0x20000126:  0014427820
0x2000012A:  0007542440
```
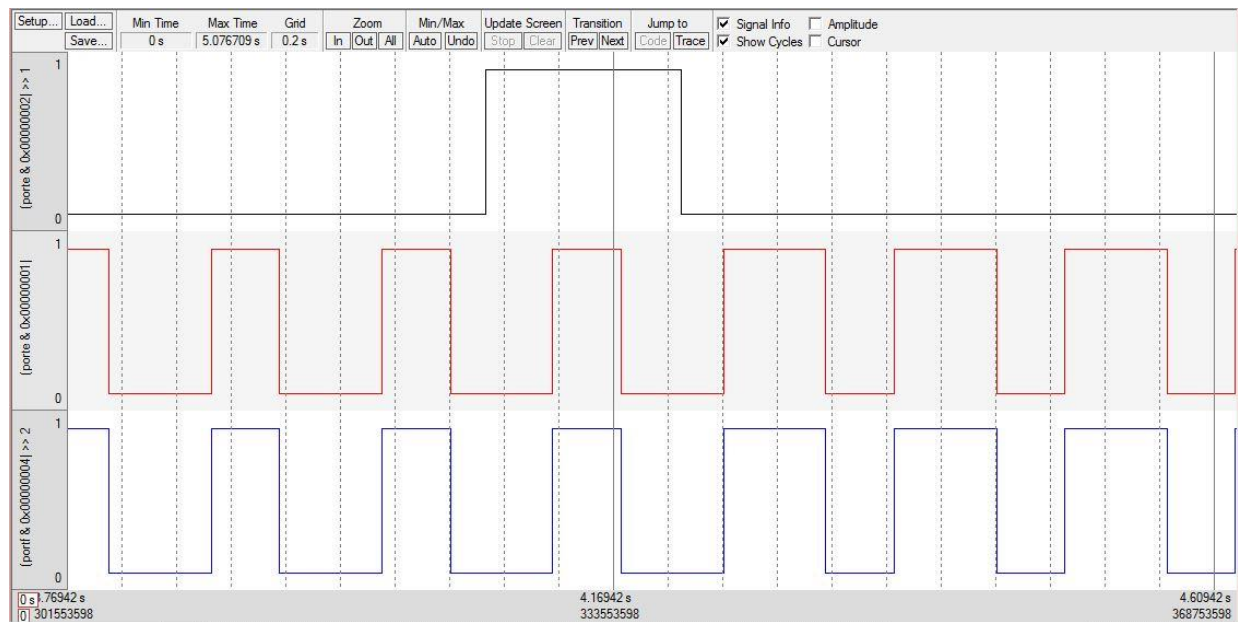
Call Sta... | Watch 1 | Memory 2

Setup... | Load... | Min Time | Max Time | Grid | Zoom | Min/Max | Update Screen | Transition | Jump to | ☑ Signal Info | ☐ Amplitude
Save... | 0 s | 5.076709 s | 0.2 s | In Out All | Auto Undo | Stop Clear | Prev Next | Code Trace | ☑ Show Cycles | ☐ Cursor

[porte & 0x00000002] >> 1

[porte & 0x00000001]

[portf & 0x00000004] >> 2

0 s | .76942 s
0 | 301553598

4.16942 s
333553598

4.60942 s
368753598

$$\{[(27 cycles*2)* 125ns]*200\}/.125 = .00108\%\ intrusiveness.$$

Paste from the saved File (50 entries)   count:

:020000042000DA
:0A006600C4816C0058EB9600B15CF9
:100070003B00935AFE00EDCBA200CFC96500283BA0
:100080000A000A39CD0046AA710027764C004B6C55
:100090000F004AE9B3006FDF76006E5C1B009252DE
:1000A000DE0091CF820098C545004E1C02003D97AE
:1000B000E30059936900490E4B00650AD10054854D
:1000C000B200708138006C2EBD008398F300489810
:1000D000F30081135B0047135B00808EC200458EE6
:1000E000C2007E092A00C0FFF80039F6C700FC5D97
:1000F0002F00AF5D2F0073C5960026C59600E92C32
:10010000FE009C2CFE00A56A3900FD2051008B0DDD
:10011000D700508CB800DF783E00A4F71F0032E40F
:0E012000A500F7628700AC26DC00A81673006D

C4816C00	58EB9600	B15CF9
	EDCBA20	CFC9650
3B00	935AFE00	0
	0A39CD0	46AA710	27764C0
283B0A00	0	0	0
	4AE9B30	6FDF760	6E5C1B0
4B6C0F00	0	0	0
		98C5450	4E1C020
9252DE00	91CF8200	0	0
		490E4B0	650AD10
3D97E300	59936900	0	0
		6C2EBD0	8398F30
5485B200	70813800	0	0
	81135B0	47135B0	808EC20
4898F300	0	0	0
	7E092A0		39F6C70
458EC200	0	C0FFF800	0
	AF5D2F0	73C5960	26C5960
FC5D2F00	0	0	0
	A56A390	FD20510
E92CFE00	9C2CFE00	0	0
8B0DD70	508CB80	DF783E0	A4F71F0
0	0	0	0
32E4A500	F7628700	AC26DC00

:00000001FF

| | | | 12.5 | <- Time per tick |
Adjust-endian	Data	Differences	Time(ms)	
006C81C4	7111108			
0096EB58	9890648	13997676	174.97095	<-time from press to release
F95CB1	16342193	10325671	129.0708875	<- first 6 time differences
003B	59	16342134	204.276675	
00FE5A93	16669331	107944	1.3493	
00A2CBED	10669037	6000294	75.003675	
0065C9CF	6670799	3998238	49.977975	
000A3B28	670504	6000295	75.0036875	
00CD390A	13449482	3998238		
0071AA46	7449158	6000324	75.00405	<-time from press to release
004C7627	5010983	2438175	30.4771875	<- next 6 time differences
000F6C4B	1010763	4000220	50.00275	
00B3E94A	11790666	5997313	74.9664125	
0076DF6F	7790447	4000219	50.0027375	
001B5C6E	1793134	5997313	74.9664125	
00DE5292	14570130	4000220	50.00275	
0082CF91	8572817	5997313		
0045C598	4572568	4000249	50.0031125	<-time from press to release
00021C4E	138318	4434250	55.428125	<- next 6 time differences
00E3973D	14915389	2000145	25.0018125	
00699359	6919001	7996388	99.95485	
004B0E49	4918857	2000144	25.0018	
00D10A65	13699685	7996388	99.95485	
00B28554	11699540	2000145	25.0018125	
00388170	3703152	7996388		
00BD2E6C	12398188	8082180	101.02725	<-time from press to release
00F39883	15964291	13211113	165.1389125	<- next 6 time differences
00F39848	15964232	59	0.0007375	
005B1381	5968769	9995463	124.9432875	

Address	Value 1	Value 2	Time	Note
005B1347	5968711	58	0.000725	
00C28E80	12750464	9995463	124.9432875	
00C28E45	12750405	59	0.0007375	
002A097E	2754942	9995463		
00F8FFC0	16318400	3213758	40.171975	<-time from press to release
00C7F639	13104697	3213703	40.1712875	<- next 6 time differences
002F5DFC	3104252	10000445	125.0055625	
002F5DAF	3104175	77	0.0009625	
0096C573	9880947	10000444	125.00555	
0096C526	9880870	77	0.0009625	
00FE2CE9	16657641	10000445	125.0055625	
00FE2C9C	16657564	77		
00396AA5	3762853	12894711	161.1838875	<-time from press to release
005120FD	5316861	15223208	190.2901	<- next 6 time differences
00D70D8B	14093707	8000370	100.004625	
00B88C50	12094544	1999163	24.9895375	
003E78DF	4094175	8000369	100.0046125	
001FF7A4	2095012	1999163	24.9895375	
00A5E432	10871858	8000370	100.004625	
008762F7	8872695	1999163		
00DC26AC	14427820	11222091	140.2761375	