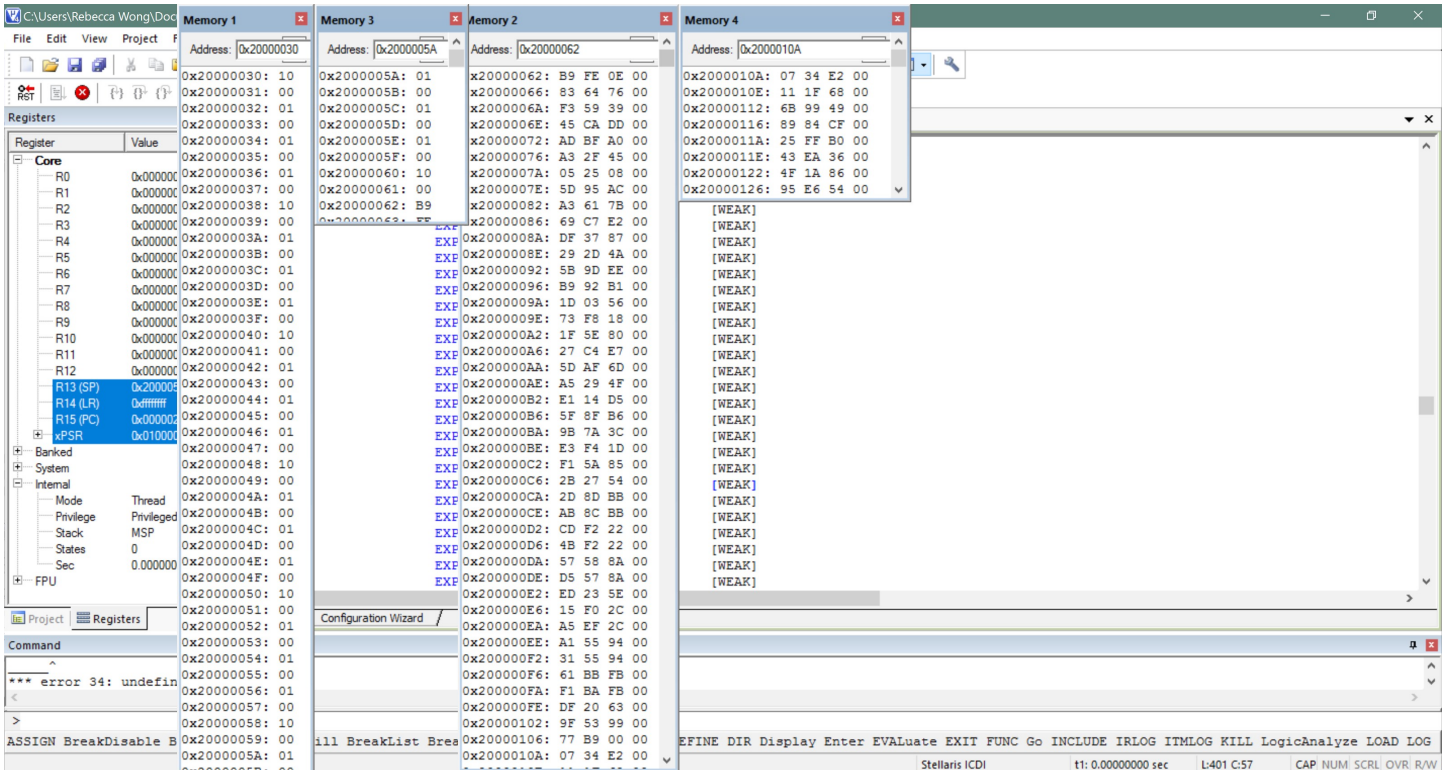
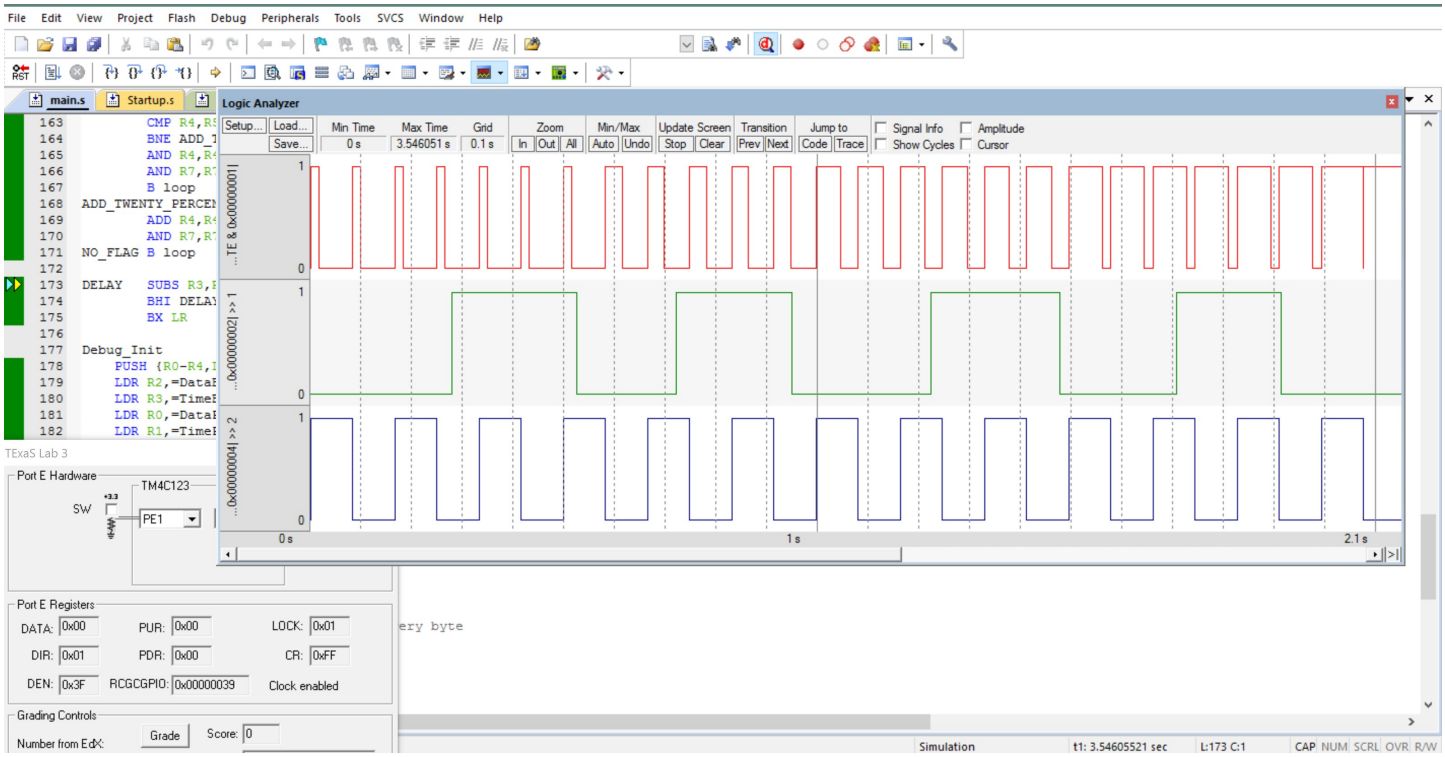

Memory 1
Address: 0x20000030
0x20000030: 10
0x20000031: 00
0x20000032: 01
0x20000033: 00
0x20000034: 01
0x20000035: 00
0x20000036: 01
0x20000037: 00
0x20000038: 10
0x20000039: 00
0x2000003A: 01
0x2000003B: 00
0x2000003C: 01
0x2000003D: 00
0x2000003E: 01
0x2000003F: 00
0x20000040: 10
0x20000041: 00
0x20000042: 01
0x20000043: 00
0x20000044: 01
0x20000045: 00
0x20000046: 01
0x20000047: 00
0x20000048: 10
0x20000049: 00
0x2000004A: 01
0x2000004B: 00
0x2000004C: 01
0x2000004D: 00
0x2000004E: 01
0x2000004F: 00
0x20000050: 10
0x20000051: 00
0x20000052: 01
0x20000053: 00
0x20000054: 01
0x20000055: 00
0x20000056: 01
0x20000057: 00
0x20000058: 10
0x20000059: 00
0x2000005A: 01
Memory 3
Address: 0x2000005A
0x2000005A: 01
0x2000005B: 00
0x2000005C: 01
0x2000005D: 00
0x2000005E: 01
0x2000005F: 00
0x20000060: 10
0x20000061: 00
0x20000062: B9
Memory 2
Address: 0x20000062
0x20000062: B9 FE 0E 00
0x20000066: 83 64 76 00
0x2000006A: F3 59 39 00
0x2000006E: 45 CA DD 00
0x20000072: AD BF A0 00
0x20000076: A3 2F 45 00
0x2000007A: 05 25 08 00
0x2000007E: 5D 95 AC 00
0x20000082: A3 61 7B 00
0x20000086: 69 C7 E2 00
0x2000008A: DF 37 87 00
0x2000008E: 29 2D 4A 00
0x20000092: 5B 9D EE 00
0x20000096: B9 92 B1 00
0x2000009A: 1D 03 56 00
0x2000009E: 73 F8 18 00
0x200000A2: 1F 5E 80 00
0x200000A6: 27 C4 E7 00
0x200000AA: 5D AF 6D 00
0x200000AE: A5 29 4F 00
0x200000B2: E1 14 D5 00
0x200000B6: 5F 8F B6 00
0x200000BA: 9B 7A 3C 00
0x200000BE: E3 F4 1D 00
0x200000C2: F1 5A 85 00
0x200000C6: 2B 27 54 00
0x200000CA: 2D 8D BB 00
0x200000CE: AB 8C BB 00
0x200000D2: CD F2 22 00
0x200000D6: 4B F2 22 00
0x200000DA: 57 58 8A 00
0x200000DE: D5 57 8A 00
0x200000E2: ED 23 5E 00
0x200000E6: 15 F0 2C 00
0x200000EA: A5 EF 2C 00
0x200000EE: A1 55 94 00
0x200000F2: 31 55 94 00
0x200000F6: 61 BB FB 00
0x200000FA: F1 BA FB 00
0x200000FE: DF 20 63 00
0x20000102: 9F 53 99 00
0x20000106: 77 B9 00 00
0x2000010A: 07 34 E2 00
Memory 4
Address: 0x2000010A
0x2000010A: 07 34 E2 00
0x2000010E: 11 1F 68 00
0x20000112: 6B 99 49 00
0x20000116: 89 84 CF 00
0x2000011A: 25 FF B0 00
0x2000011E: 43 EA 36 00
0x20000122: 4F 1A 86 00
0x20000126: 95 E6 54 00
Registers
Register
Value
Core
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13 (SP)
R14 (LR)
R15 (PC)
xPSR
Banked
System
Internal
Mode
Thread
Privilege
Privileged
Stack
MSP
States
0
Sec
FPU
Project
Registers
Configuration Wizard
Command
*** error 34: undefin
ASSIGN BreakDisable B
ill BreakList Brea
EFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE IRLOG ITMLOG KILL LogicAnalyze LOAD LOG
[WEAK]
Stellaris ICDI
t1: 0.00000000 sec
L:401 C:57
CAP NUM SCRL OVR R/W


File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
main.s
Startup.s
Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 3.546051 s
Grid 0.1 s
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Show Cycles
Amplitude
Cursor
...TE & 0x00000001
...0x00000002] >> 1
...0x00000004] >> 2
0 s
1 s
2.1 s
TExaS Lab 3
Port E Hardware
TM4C123
SW
PE1
Port E Registers
DATA: 0x00
PUR: 0x00
LOCK: 0x01
DIR: 0x01
PDR: 0x00
CR: 0xFF
DEN: 0x3F
RCGCGPIO: 0x00000039
Clock enabled
Grading Controls
Number from EdX:
Grade
Score: 0
Simulation
t1: 3.54605521 sec
L:173 C:1
CAP NUM SCRL OVR R/W

```
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_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R  EQU 0x4002552C
GPIO_PORTF_LOCK_R  EQU 0x40025520
GPIO_PORTF_CR_R    EQU 0x40025524
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

COUNT         EQU 332000
ENTIRE_COUNT EQU 1660000
LED_ON EQU1
LED_OFF EQU 0
SWITCH EQU 2
ONE EQU 1

; RAM Area
          AREA    DATA, ALIGN=2
;-UUU-Declare  and allocate space for your Buffers
;    and any variables (like pointers and counters) here
DataBuffer SPACE 50
TimeBuffer SPACE 200 ; 50 elements 4 bytes each
DataPt    SPACE 4
```

```
TimePt            SPACE 4
NEntries       SPACE 1

; ROM Area
      IMPORT  TExaS_Init
        IMPORT SysTick_Init
        IMPORT NVIC_ST_CURRENT_R
;-UUU-Import routine(s) from other assembly files (like SysTick.s) here
      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 clock for Port F
   LDR R0, [R1]
   ORR R0, R0, #0x30              ; set bits 4 & 5 to turn on clock
   STR R0, [R1]
   NOP
   NOP
        LDR R1, =GPIO_PORTF_LOCK_R      ; 2) unlock the lock register
   LDR R0, =0x4C4F434B            ; unlock GPIO Port F Commit Register
   STR R0, [R1]
   LDR R1, =GPIO_PORTF_CR_R        ; enable commit for Port F
   MOV R0, #0xFF                  ; 1 means allow access
   STR R0, [R1]
   LDR R1, =GPIO_PORTF_AMSEL_R     ; 3) disable analog functionality
```

```
    MOV R0, #0                  ; 0 means analog is off
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PCTL_R      ; 4) configure as GPIO
    MOV R0, #0x00000000             ; 0 means configure Port F as GPIO
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DIR_R       ; 5) set direction register
    MOV R0,#0x04                    ;PF2 is output
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AFSEL_R     ; 6) regular port function
    MOV R0, #0                  ; 0 means disable alternate function
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DEN_R       ; 7) enable Port F digital port
    MOV R0, #0xFF                ; 1 means enable digital I/O
    STR R0, [R1]
    LDR R1, =GPIO_PORTE_DIR_R       ; set direction register
    MOV R0,#0x01                ; PE1 INPUT, PE0 output
    STR R0, [R1]
    LDR R1, =GPIO_PORTE_AFSEL_R     ; regular port function
    MOV R0, #0                  ; 0 means disable alternate function
    STR R0, [R1]
    LDR R1, =GPIO_PORTE_DEN_R       ; enable Port E digital port
    MOV R0, #0xFF                ; 1 means enable digital I/O
    STR R0, [R1]
    LDR R0, [R1]
            BL      Debug_Init
            LDR R4,=COUNT                                   ;R4 HAS 20% OF DELAY
            LDR R9,=ONE
            LDR R5,=ENTIRE_COUNT
            AND R7,R7,#0                                    ;R7 WILL HOLD FLAG
THAT WILL BE ASSERTED IF SWITCH HAS BEEN PRESSED
```

```
        LDR R0,=GPIO_PORTE_DATA_R
        MOV   R11,#0
        LDR R10,=GPIO_PORTF_DATA_R      ; THIS SECTION TOGGLES
loop    LDR R12,[R10]                   ;
        AND R12,R12,#0x04                       ; THE HEARTBEAT
        EOR R12,#0x04                           ;
        STR R12,[R10]                           ; LED ON PF2
        CMP   R11,#0
        BEQ   FLASH
        BL      Debug_Capture
        SUB   R11,R11,#1
FLASH
        LDR R1,[R0]                             ;STARTING FLASHING
        LDR R2,=LED_ON                          ;TURNING
        ORR R1,R1,R2                            ;ON
        STR R1,[R0]                             ;LED
        MOV R3,R4                               ;R3 WILL BE USED BY
DELAY FUNCTION & DECREMENTED
        BL DELAY
        CMP   R11,#0
        BEQ   FLASH2
        BL      Debug_Capture
        SUB   R11,R11,#1
FLASH2
        LDR R1,[R0]                             ;TURNING
        LDR R2,=LED_OFF
        AND R1,R1,R2                            ;LED
        STR R1,[R0]                             ;OFF
        MOV R3,R4               ;REPLENISHING R3 (BECAUSE IT WAS
DECREMENTED TO 0 IN DELAY)
```

```
        RSB R3,R3,R5                                    ;SUBTRACTING DUTY
CYCLE PERCENTATGE FROM 100% TO GET OFF DUTY CYCLE

        BL DELAY

        LDR R1,[R0]

        LDR R6,=SWITCH

        AND R1,R1,R6                                    ;BIT MASKING SWITCH

        CMP R1, R6

        BNE NOT_CURRENTLY_PRESSED                               ;CHECKING
IF SWITCH  IS PRESSED CURRENTLY

        CMP R7,R9
    ;SWITCH HAS ALREADY BEEN ASSERTED  BUT SWITCH IS STILL CURRENTLY
PRESSED

        BEQ NO_FLAG

        MOV R11,#1

        LDR R7,=ONE

        B NO_FLAG

NOT_CURRENTLY_PRESSED CMP R7,R9                         ;COMPARING TO
ONE TO SEE IF FLAG IS ASSERTED

        BNE NO_FLAG

        MOV R11,#7

        CMP R4,R5                                       ;SWITCH HAS BEEN
ASSERTED AND RELEASED, CHANGING DUTY CYCLE

        BNE ADD_TWENTY_PERCENT                  ;CHECKING IF THE DUTY
CYCLE IS AT 100%

        AND R4,R4,#0

        AND R7,R7,#0

        B loop                                          ;SETTING IT TO 0
FOR THE NEXT ROUND

ADD_TWENTY_PERCENT LDR R8,=COUNT

        ADD R4,R4,R8                    ;ADDING 20% TO THE DUTY CYCLE

        AND R7,R7,#0

NO_FLAG B loop
```

```
DELAYSUBS R3,R3,#1
        BHI DELAY
        BX LR


Debug_Init
    PUSH {R0-R4,LR}
    LDR R2,=DataBuffer
    LDR R3,=TimeBuffer
    LDR R0,=DataPt
    LDR R1,=TimePt
    STR R2,[R0]
    STR R3,[R1]
    LDR R0,[R0]  ; R0 and R1 now have the actual pointers R0 DATA R1 TIME
    LDR R1,[R1]
    MOV R2,#50 ;R2 will keep track of how many array values we have traversed
through
    MOV R3,#0xFF
Buffer_Init             ;loop that stores 0xFF into all buffer locations
    STRB R3,[R0]        ;STORE 0xFF into data buffer location
    STRB R3,[R1]
    STRB R3,[R1,#1]
    STRB R3,[R1,#2]
    STRB R3,[R1,#3] ;STORE 0xFF into every byte
    ADD R0,R0,#1
    ADD R1,R1,#4
    SUBS R2,R2,#1
    BNE Buffer_Init
    BL      SysTick_Init
    LDR R0,=NEntries
    MOV R1,#50
```

```
        STRB R1,[R0]
        POP {R0-R4,PC}     ;return from subroutine


Debug_Capture
        PUSH {R0-R10,LR}
        LDR R0,=NEntries                    ;Checking if we have stored 50 entries
yet
        LDRB R1,[R0]
        CMP R1,#0
        BEQ DONE
        LDR R2,=GPIO_PORTE_DATA_R
        LDR R3,[R2]                         ;Loading in port E data
        LDR R7,=NVIC_ST_CURRENT_R
        LDR R8,[R7]                         ;Loading in SysTick currrent
time
        AND R4,R3,#0x02                     ;bit masking switch
        LSL R4,R4,#3                        ;shifting it over to next nibble
        AND R3,R3,#0x01                     ;combining Data to have PEO in
first nibble and
        ORR R3,R3,R4                        ; PE1 in next nibble
        LDR R5,=DataPt
        LDR R6,[R5]
        STRB R3,[R6]                        ;storing data in data buffer
array
        ADD R6,R6,#1
        STR R6,[R5]                         ;incrememnted dataBuffer
pointer for next time around
        LDR R9,=TimePt
        LDR R10,[R9]                        ;R10 has next availabl address
of time buffer array
        STR R8,[R10]
```

```
        ADD R10,R10,#4                          ;Incrementing time buffer
pointer
        STR R10,[R9]
        SUB    R1,R1,#1                         ;decrementing index counter
        STRB R1,[R0]
DONE
        POP {R0-R10,PC}


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


NVIC_ST_CTRL_R        EQU 0xE000E010
NVIC_ST_RELOAD_R      EQU 0xE000E014
NVIC_ST_CURRENT_R     EQU 0xE000E018
```

```
RELOAD_VALUE                EQU 0xFFFFFFFF
        AREA    |.text|, CODE, READONLY, ALIGN=2
        THUMB
; ;-UUU-Export routine(s) from SysTick.s to callers
          EXPORT        SysTick_Init
          EXPORT        NVIC_ST_CURRENT_R


;------------SysTick_Init------------
; ;-UUU-Complete this subroutine
; Initialize SysTick with busy wait running at bus clock.
; Input: none
; Output: none
; Modifies: ??
SysTick_Init
          PUSH {R0-R3}
          LDR R0,=NVIC_ST_CTRL_R
          LDR R1,[R0]
          BIC R1,R1,#0x01
          STR R1,[R0]
          LDR R2,=NVIC_ST_RELOAD_R
          LDR R3,=RELOAD_VALUE
          STR R3,[R2]
          LDR R2,=NVIC_ST_CTRL_R
          MOV R1,#0
          STR R1,[R2]
          LDR R1,[R0]
          ORR R1,R1,#0x05
          BIC R1,R1,#0x02
          STR R1,[R0]
          POP {R0-R3}
```

```
BX  LR                  ; return

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

Paste from the saved File (50 entries)

:020000042000DA		
:0E006200D11E030011EBD10073E09400CB50CF	D11E0300	11EBD100
:1000700039002D46FC004FB6A000ABAB6300E91B76	CB503900	2D46FC00
:100080000800B5816F00F54D3E0065BEE200AFB3DC	E91B0800	B5816F00
:10009000A500DD234A0033190D009789B100ED7EDC	AFB3A500	DD234A00
:1000A0007400B9161C0083152600BF00AC00077B46	ED7E7400	B9161C00
:1000B0008D0037661300C1E0F400F1CB7A003946B9	077B8D00	37661300
:1000C0005C008B0D5D003971E20035D74900B3D675	39465C00	8B0D5D00
:1000D0004900F53CB100733CB10079A21800F7A1CA	B3D64900	F53CB100
:1000E0001800C3068F003D069400CD059400C36B35	F7A11800	C3068F00
:1000F000FB00536BFB0083D1620013D16200DB363F	C36BFB00	536BFB00
:10010000CA008F35D9008DCE7600E7485800F1330C	DB36CA00	8F35D900
:10011000DE0081AEBF00AB9945000514270023FF28	F133DE00	81AEBF00
:0A012000AC00C9FDB600B59654000E	23FFAC00	C9FDB600
:00000001FF		

count: 50

					12.5	**<- Time p**
73E09400		**Adjust-endia**	**Data**	**Differences**	**Time(ms)**	
4FB6A000	ABAB6300	00031ED1	204497			
F54D3E00	65BEE200	00D1EB11	13757201	3224512	40.3064	<-time fror
33190D00	9789B100	0094E073	9756787	4000414	50.005175	<- first 6 ti
83152600	BF00AC00	003950CB	3756235	6000552	75.0069	
C1E0F400	F1CB7A00	00FC462D	16533037	4000414	50.005175	
3971E200	35D74900	00A0B64F	10532431	6000606	75.007575	
733CB100	79A21800	0063ABAB	6532011	4000420	50.00525	
3D069400	CD059400	00081BE9	531433	6000578	75.007225	
83D16200	13D16200	006F81B5	7307701	10000948		
8DCE7600	E7485800	003E4DF5	4083189	3224512	40.3064	<-time fror
AB994500	05142700	00E2BE65	14859877	6000528	75.0066	<- next 6 t
B5965400		00A5B3AF	10859439	4000438	50.005475	
		004A23DD	4858845	6000594	75.007425	
		000D1933	858419	4000426	50.005325	
		00B18997	11635095	6000540	75.00675	
		00747EED	7634669	4000426	50.005325	
		001C16B9	1840825	5793844		
		00261583	2495875	16122166	201.527075	<-time fror
		00AC00BF	11272383	8000708	100.00885	<- next 6 t
		008D7B07	9272071	2000312	25.0039	
		00136637	1271351	8000720	100.009	
		00F4E0C1	16048321	2000246	25.003075	
		007ACBF1	8047601	8000720	100.009	
		005C4639	6047289	2000312	25.0039	
		005D0D8B	6098315	16726190		
		00E27139	14840121	8035410	100.442625	<-time fror
		0049D735	4839221	10000900	125.01125	<- next 6 t
		0049D6B3	4839091	130	0.001625	
		00B13CF5	11615477	10000830	125.010375	
		00B13C73	11615347	130	0.001625	
		0018A279	1614457	10000890	125.011125	
		0018A1F7	1614327	130	0.001625	
		008F06C3	9373379	9018164		
		0094063D	9700925	16449670	205.620875	<-time fror
		009405CD	9700813	112	0.0014	<- next 6 t
		00FB6BC3	16477123	10000906	125.011325	
		00FB6B53	16477011	112	0.0014	
		0062D183	6476163	10000848	125.0106	
		0062D113	6476051	112	0.0014	
		00CA36DB	13252315	10000952	125.0119	
		00D9358F	14235023	15794508		
		0076CE8D	7786125	6448898	80.611225	<-time fror
		005848E7	5785831	2000294	25.003675	<- next 6 t
		00DE33F1	14562289	8000758	100.009475	

00BFAE81	12562049	2000240	25.003
004599AB	4561323	8000726	100.009075
00271405	2561029	2000294	25.003675
00ACFF23	11337507	8000738	100.009225
00B6FDC9	11992521	16122202	
005496B5	5543605	6448916	80.61145

er tick

n press to release
me differences

n press to release
ime differences

n press to release
ime differences

n press to release
ime differences

n press to release
ime differences

n press to release
ime differences

```
:020000042000DA
:10003000100001000100010010000100010001009A
:10004000100001000100010010000100010001008A
:10005000100001000100010010000100010001007A
:0200600010008E
:00000001FF
```

Estimation of Intrusiveness

$$26 \text{ instructions} \times \frac{2 \text{ cycles}}{\text{instruction}} = 52 \text{ cycles} \cdot \frac{12.5 \text{ ns}}{1 \text{ cycle}} =$$

Δt ↓ (called twice)

$$\frac{(650 \text{ ns})(2)}{.125} \cdot 100 =$$

.00104 % intrusive