



TEaS Lab 4

Port E Hardware

TM4C123 80 MHz

SW +3.3 V

PE0 PE1

LED

Port E Registers

DATA: 0x02 PUR: 0x00 LOCK: 0x01

DIR: 0x02 PDR: 0x00 CR: 0xFF

DEN: 0x03 RCGCGPIO: 0x00000039 Clock enabled

Grading Controls

Grade Score: 0

Memory 1

Address: 0x20000030

0x20000030:	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001
0x20000054:	00000001	00000001	00000001	00000001	00000001	00000010	00000011	00000010	00000011
0x20000078:	00000010	00000011	00000010	00000011	00000010	00000011	00000010	00000011	00000010
0x2000009C:	00000011	00000010	00000011	00000010	00000011	00000010	00000011	00000001	00000001
0x200000C0:	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001
0x200000E4:	00000001	00000001	00000001	00000001	00000001	00FFFFFF	00B45052	0068A0E5	001CF178
0x20000108:	00D1420B	0085929E	0039E331	00EE33C4	00A28457	0056D4EA	000B257D	00BF7610	0073C6A3
0x2000012C:	00281736	00DC67CB	0090B860	004508F5	00F9598A	00ADAA1F	0061FAB4	00164B49	00CA9BDE
0x20000150:	007EEC73	00333D08	00E78D9D	009BDE32	00502EC7	00047F5C	00B8CFF1	006D2086	0021711B
0x20000174:	00D5C1B0	008A1245	003E62DA	00F2B36D	00A70400	005B5493	000FA526	00C3F5B9	0078464C

BreakSet BreakAccess COVERAGE DEFINE DIR

Call Stack + Locals Memory 1

Simulation t1: 3.06956019 sec L:248 C:1 CAP NUM SCRL OVR R/W

```
;***** main.s *****
; Program written by: Michael Griffin and Rohan Kondetimmanahalli
; Date Created: 1/24/2015
; Last Modified: 2/27/2015
; Section 1-2pm   TA: Wooseok Lee
; Lab number: 4
; Brief description of the program
; If the switch is presses, the LED toggles at 8 Hz
; Hardware connections
; PE0 is switch input (1 means pressed, 0 means not pressed)
; PE1 is LED output (1 activates external LED on protoboard)
; Overall functionality of this system is the similar to Lab 3, with four changes:
;1- activate the PLL to run at 80 MHz (12.5ns bus cycle time)
;2- initialize SysTick with RELOAD 0x00FFFFFF
;3- add a heartbeat to PF2 that toggles every time through loop
;4- add debugging dump of input, output, and time
; Operation
;   1) Make PE1 an output and make PE0 an input.
;   2) The system starts with the LED on (make PE1 =1).
;   3) Wait about 62 ms
;   4) If the switch is pressed (PE0 is 1), then toggle the LED once, else turn the LED on.
;   5) Steps 3 and 4 are repeated over and over
```

```
SWITCH      EQU 0x40024004 ;PE0
LED          EQU 0x40024008 ;PE1
SYSCTL_RCGCGPIO_R    EQU 0x400FE608
SYSCTL_RCGC2_GPIOE   EQU 0x00000010 ; port E Clock Gating Control
SYSCTL_RCGC2_GPIOF   EQU 0x00000020 ; port F Clock Gating Control
GPIO_PORTE_DATA_R     EQU 0x400243FC
GPIO_PORTE_DIR_R      EQU 0x40024400
GPIO_PORTE_AFSEL_R    EQU 0x40024420
GPIO_PORTE_PUR_R      EQU 0x40024510
GPIO_PORTE_DEN_R      EQU 0x4002451C
GPIO_PORTF_DATA_R     EQU 0x400253FC
GPIO_PORTF_DIR_R      EQU 0x40025400
GPIO_PORTF_AFSEL_R    EQU 0x40025420
GPIO_PORTF_DEN_R      EQU 0x4002551C
NVIC_ST_CTRL_R        EQU 0xE000E010
NVIC_ST_RELOAD_R      EQU 0xE000E014
NVIC_ST_CURRENT_R     EQU 0xE000E018
```

```
THUMB
```

```
AREA DATA, ALIGN=4
```

```
SIZE EQU 50
```

```
;You MUST use these two buffers and two variables
```

```
;You MUST not change their names
```

```
;These names MUST be exported
```

```
EXPORT DataBuffer
```

```
EXPORT TimeBuffer
```

```
EXPORT DataPt [DATA,SIZE=4]
```

```
EXPORT TimePt [DATA,SIZE=4]
```

```
DataBuffer SPACE SIZE*4
```

```
TimeBuffer SPACE SIZE*4
```

DataPt SPACE 4
TimePt SPACE 4

ALIGN
AREA |.text|, CODE, READONLY, ALIGN=2
THUMB
EXPORT Start
IMPORT TExaS_Init

Start BL TExaS_Init ; running at 80 MHz, scope voltmeter on PD3
BL PortE_Init; initialize Port E
BL PortF_Init; initialize Port F
BL Debug_Init; initialize debugging dump, including SysTick
MOV R0, #0x02
BL PortE_Output ;turn LED on

CPSIE I ; TExaS voltmeter, scope runs on interrupts
loop BL Debug_Capture ;675ns execution time, .062s between calls, .0011% overhead
BL Heartbeat ;toggle board led
BL delay
BL PortE_Input
CMP R0, #0x01
BEQ pressed ;if switch pressed go to pressed, else continue
MOV R0, #0x02
BL PortE_Output ;turn LED on
B loop

pressed
LDR R0, =GPIO_PORTE_DATA_R
LDR R1, [R0] ;R1 = GPIO_PORTE_DATA_R
EOR R1, #0x02 ;toggle PE1
STR R1, [R0]
B loop

;-----Debug_Init-----
; Initializes the debugging instrument
; Input: none
; Output: none
; Modifies: none
; Note: push/pop an even number of registers so C compiler is happy

Debug_Init
MOV R1, #0xFFFFFFFF ;R1 is all 1's
LDR R0, =DataBuffer ;R0 points to first address of databuffer
MOV R2, #0 ;R2 = 0
loop1 STR R1, [R0] ;change value of databuffer at current address to 1's
ADD R0, #4 ;increment temporary pointer
ADD R2, #1 ;increment counter
CMP R2, #50
BNE loop1 ;if counter == 50, continue
LDR R0, =TimeBuffer ;R0 points to first address of timebuffer
MOV R2, #0 ;R2 = 0

```

loop2 STR    R1, [R0]           ;change value of timebuffer at current address to 1's
      ADD    R0, #4             ;increment temporary pointer
      ADD    R2, #1             ;increment counter
      CMP    R2, #50
      BNE    loop2             ;if counter == 50, continue
      LDR    R1, =DataBuffer    ;R1 points to first address of databuffer
      LDR    R0, =DataPt       ;R0 points to address of datapointer
      STR    R1, [R0]          ;datapointer now has address of first index in databuffer
      LDR    R1, =TimeBuffer    ;R1 points to first address of timebuffer
      LDR    R0, =TimePt       ;R0 points to address of timepointer
      STR    R1, [R0]          ;timepointer now has address of first index in timebuffer
      PUSH   {R0, LR}          ;save LR
      BL SysTick_Init          ;initialize timer
      POP   {R0, PC}           ;restore LR to PC
; init SysTick

```

```

BX    LR

```

```

;-----Debug_Capture-----

```

```

; Dump Port E and time into buffers

```

```

; Input: none

```

```

; Output: none

```

```

; Modifies: none

```

```

; Note: push/pop an even number of registers so C compiler is happy

```

```

Debug_Capture

```

```

      PUSH   {R0-R4, LR}       ;save values
      LDR    R0, =DataBuffer    ;R0 points to first address of databuffer
      ADD    R0, #196           ;R0 points to last address of databuffer
      LDR    R1, =DataPt       ;R1 points to address of datapointer
      LDR    R2, [R1]          ;R2 has data pointer address
      CMP    R2, R0
      BHI    leave             ;if pointer address is higher than last address, stop subroutine
      LDR    R0, =GPIO_PORTE_DATA_R
      LDR    R3, [R0]          ;R3 = GPIO_PORTE_DATA_R
      AND    R2, R3, #0x02      ;isolate bit 1 and put in R2
      LSR    R2, #1            ;shift bit 1 to bit 0
      AND    R3, #0x01         ;isolate bit 0
      LSL    R3, #4            ;shift bit 0 to bit 4
      ORR    R3, R2            ;combine two values
      LDR    R0, [R1]          ;R0 = pointer address
      STR    R3, [R0]          ;combined values go to pointer address
      ADD    R0, #4            ;increment data pointer
      STR    R0, [R1]
      LDR    R1, =TimePt       ;R1 points to address of timepointer
      LDR    R0, [R1]          ;R0 has time pointer address
      LDR    R2, =NVIC_ST_CURRENT_R
      LDR    R3, [R2]          ;R3 has current time
      STR    R3, [R0]          ;time buffer gets current time at time pointer address
      ADD    R0, #4            ;increment time pointer
      STR    R0, [R1]
leave POP   {R0-R4, PC}       ;restore values and leave

```

```

;-----PortE_Init-----

```

```

; Initialize GPIO Port E for positive logic switch on

```

```

; PE1 as the Launchpad is wired. Make the PE0 output.
; Input: none
; Output: none
; Modifies: R0, R1, R2
PortE_Init
    LDR R1, =SYSCTL_RCGCGPIO_R    ; activate clock for Port E
    LDR R0, [R1]
    ORR R0, R0, #0x10              ; set bit 5 to turn on clock
    STR R0, [R1]
    NOP
    NOP                            ; allow time for clock to finish
    LDR R1, =GPIO_PORTE_DIR_R      ; set direction register
    MOV R0, #0x02                  ; PE0 input, PE1 output
    STR R0, [R1]
    LDR R1, =GPIO_PORTE_DEN_R      ; enable Port E digital port
    MOV R0, #0x03                  ; 1 means enable digital I/O, PE0 and PE1
    STR R0, [R1]
    BX LR

```

```

;-----PortF_Init-----
; Initialize GPIO Port F for heartbeat on
; PF2 as the Launchpad is wired.
; Input: none
; Output: none
; Modifies: R0, R1, R2

```

```

PortF_Init
    LDR R1, =SYSCTL_RCGCGPIO_R    ; activate clock for Port F
    LDR R0, [R1]
    ORR R0, R0, #0x20              ; set bit 5 to turn on clock
    STR R0, [R1]
    NOP
    NOP                            ; allow time for clock to finish
    LDR R1, =GPIO_PORTF_DIR_R      ; set direction register
    MOV R0, #0x04                  ; PF2 output
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DEN_R      ; enable Port F digital port
    MOV R0, #0x04                  ; 1 means enable digital I/O, PF2
    STR R0, [R1]
    BX LR

```

```

;-----PortE_Input-----
; Read and return the status of the switches.
; Input: none
; Output: R0 0x01 if Switch is pressed
;         R0 0x00 if Switch isn't pressed
; Modifies: R1

```

```

PortE_Input
    LDR R1, =SWITCH                ; pointer to Port E data
    LDR R0, [R1]                  ; read all of Port E
    AND R0, R0, #0x01              ; just the input pin PE0
    BX LR                          ; return R0 with inputs

```

```

;-----PortE_Output-----
; Set the output state of PE1.
; Input: R0 new state of PE
; Output: none

```

```

; Modifies: R1
PortE_Output
    LDR R1, =LED                ; pointer to Port E data
    STR R0, [R1]                ; write to PF1
    BX LR

;-----SysTick_Init-----
; Initialize SysTick timer
; Input: none
; Output: none
; Modifies: R0, R1, R2
SysTick_Init
    LDR    R1, =NVIC_ST_CTRL_R
    MOV    R0, #0                ;disable SysTick during setup
    STR R0, [R1]
    LDR    R1, =NVIC_ST_RELOAD_R    ;R1 = &NVIC_ST_RELOAD_R
    LDR    R0, =0x00FFFFFF        ;max reload value
    STR R0, [R1]                  ;[R1] = R0 = NVIC_ST_RELOAD_M
    LDR    R1, =NVIC_ST_CURRENT_R    ;R1 = &NVIC_ST_CURRENT_R
    MOV    R0, #0                ;any writes to current clears it
    STR R0, [R1]                  ;clear counter
    LDR    R1, =NVIC_ST_CTRL_R        ;enable SysTick with core clock
    MOV    R0, #0x05
    STR R0, [R1]                  ;ENABLE and CLK_SRC bits set
    BX LR

;-----delay-----
; Long delay function for testing, which delays about 3*count cycles.
; Input: R0 count
; Output: none
; Modifies: R1
delay
    LDR    R0, =1240000            ; approximately 62 ms delay at ~80 MHz clock

back2    SUBS    R0, #1                ; R0 = R0 - 1 (count = count - 1)

        BNE back2                ; if count (R0) != 0, skip to 'delay'
        BX LR                    ; return

;-----Heartbeat-----
; Toggles PF2 for debugging
; Input: none
; Output: none
; Modifies: none
Heartbeat
    PUSH {R0, R1}                ;save R0 and R1
    LDR    R1, =GPIO_PORTF_DATA_R
    LDR    R0, [R1]                ;R0 = GPIO_PORTF_DATA_R
    EOR    R0, #0x04                ;not pin 2
    STR    R0, [R1]                ;store R0 back to GPIO_PORTF_DATA_R
    POP {R0, R1}                  ;restore R0 and R1
    BXL R
    ALIGN                ; make sure the end of this section is aligned
    END                    ; end of file

```

01000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
010000000100000001000000011000000
100000001100000001000000011000000
100000001100000001000000011000000
100000001100000001000000011000000
100000001100000001000000011000000
100000001100000001000000011000000
01000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
0100000001000000BFFFFFF005250B400
E5A0680078F11C000B42D1009E928500
31E33900C433EE005784A200EAD45600
7D250B001076BF00A3C6730036172800
CB67DC0060B89000F50845008A59F900
1FAAAD00B4FA6100494B1600DE9BCA00
73EC7E00083D33009D8DE70032DE9B00
C72E50005C7F0400F1CFB80086206D00
1B712100B0C1D50045128A00DA623E00
6DB3F2000004A70093545B0026A50F00
B9F5C3004C467800DF962C0072E7E000
05389500988849002BD9FD00BE29B200
517A6600E4CA1A00771BCF000A6C8300

$(00FFFFBF-00B45052) * 12.5 \text{ ns} = 4960109 * 12.5e-9 = 62.001 \text{ ms cycle period}$
 $2 * \text{cycle period} = \text{time period}$
time period = 124.002 ms