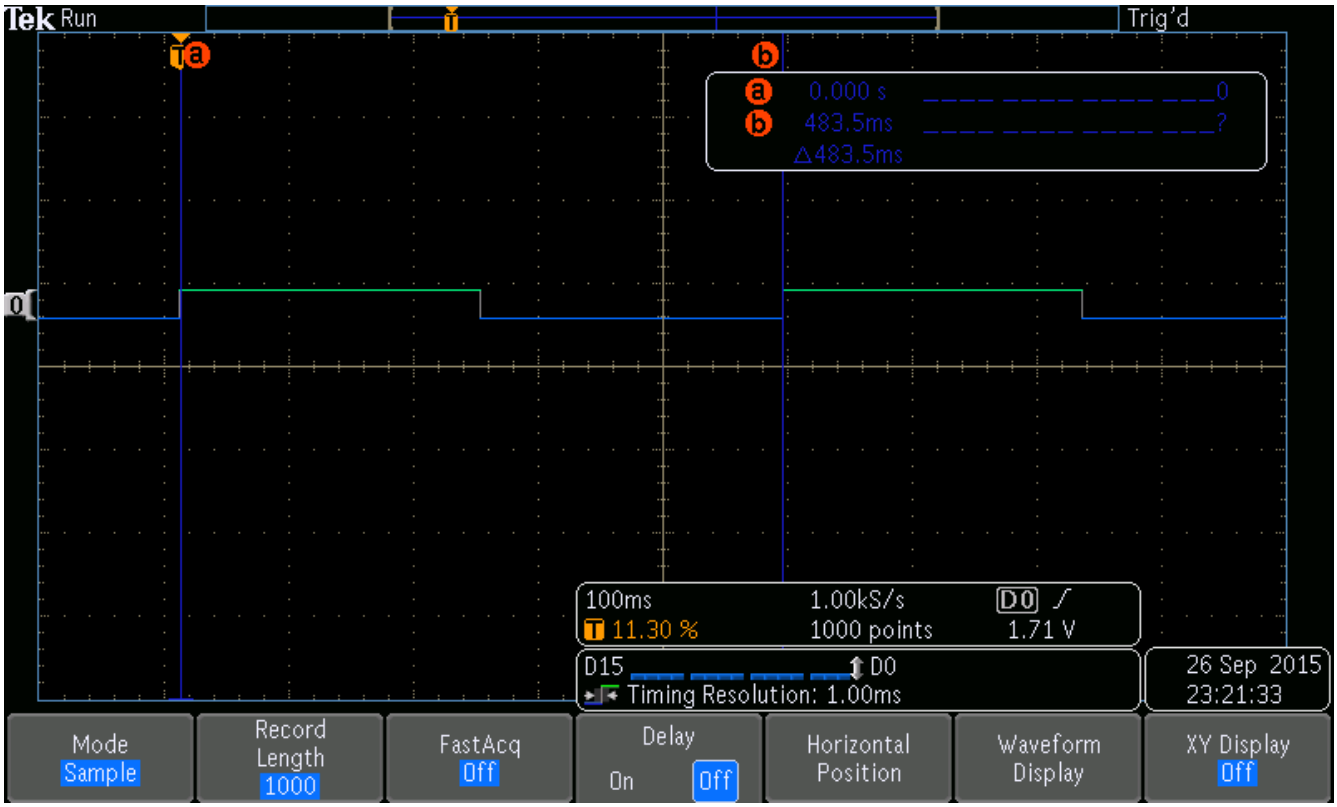
LAB 2
ECE 3710
Dr. Ryan Gerdes
By: Jonathan Tousley

Goal:
Design a system such that a 10 pin LED light will count in binary beginning at 0. Count to the max value and loop. Do nothing until start is pressed, and then begin counting. Design an interrupt stop that will pause the counter and a reset interrupt that will return the counter to 0 and wait for start. The rate of increase should be 2 counts per second, plus or minus 5%.

Solution:
Connect the 10 pins to the LED to the microcontroller in Pull-Down configuration so the microcontroller does not need to supply the power. Connect the third switch to the microcontroller in Pull-Down configuration and add some resistance to reduce current sink. Because of the active-low position configuration of the LEDs, set the counter to start from the max value and count down to zero. The pin configuration was as follows:
LED[0:5] = PA[2:7] (lower six values of the LED reflecting lower binary values. OUTPUT)
LED[6:9] = PB[0:3] (higher four values of the LED. OUTPUT)
SW3 = PB4 (switch 3, configured to be RESET. INPUT)
SW2 = PF0 (switch 2, configured to be STOP. INPUT)
SW1 = PF4 (switch 1, configured to be START. INPUT)

Schematics drawn in other PDF file.

Verification of timing to be 500ms +- 25ms:



Code for microcontroller:

```
PA EQU 0x40004000
PB EQU 0x40005000
PF EQU 0x40025000
RCGC2 EQU 0x400FE608

Start

; ***************CONFIGURATION / SETUP ***********************
  ; 1. activate clock
  LDR R1,=RCGC2                 ;clock enable register
  MOV R0,#0x23                  ; enable PA and PB and PF
  STR R0,[R1]
  NOP                           ;system clock takes a while...
  NOP                           ;to get going...

  ; 2. disable alt. function
  LDR R1,=PA
  MOV R0,#0x0
  STR R0,[R1,#0x420]
  LDR R1, =PB
  STR R0, [R1, #0x420]
  LDR R1, =PF
  STR R0, [R1, #0x420]
  MOV32 R0, #0x4C4F434B             ; GPIO Unlock code.
  STR R0, [R1,#0x520]               ; unlock GPIOF_LOCK

  ; 3. set port pins as OUTPUT
  LDR R1, =PA
  MOV R0,#0xFF                  ; PA[0:7] output
  STR R0,[R1,#0x400]        ;
  LDR R1, =PB
  MOV R0, #0x0F                 ; PB[0:3] output, [4:7] INPUT
  STR R0, [R1, #0x400]      ;
  LDR R1, =PF
  MOV R0, #0x00                 ; set PF as input
  STR R0, [R1, #0x400]

  ; 4. additional settings
  LDR R1, =PA
  MOV R0, #0x0                  ;disable GPIOPCTL PA
  STR R0, [R1, #0x52C]

  LDR R1, =PB
  STR R0, [R1, #0x52C]      ;disable GPIOPCTL PB
  MOV R0, #0x10                 ; set pull down for PB4
  STR R0, [R1, #0x514]
```

```
  LDR R1, =PF
  MOV R0, #0x01
  STR R0, [R1,#0x524]           ; GPIOCR unlock pin PF0
;  MOV32 R0, #0x4C4F434B              ; GPIO Unlock code. Unlock GPIOF_LOCK after write to
GPIOCR (pg681)
;  STR R0, [R1,#0x520]           ;
  MOV R0, #0x11                 ; set pull up for PF0 and PF4
  STR R0, [R1, #0x510]

  ; 5. enable port
  LDR R1, =PA
  MOV R0,#0xFF
  STR R0,[R1,#0x51C]            ; enable 8 pins on PA
  LDR R1, =PB
  STR R0, [R1, #0x51C]          ;enable 8 pins on PB
  MOV R0, #0x1F                 ; enable PF4 (sw1) and PF0 (sw2)
  LDR R1, =PF
  STR R0, [R1, #0x51C]

;********************* CONFIGURATION END ***************************

begin
        MOV R0, #0xFF                   ; initialize zero
        LDR R1, =PA
        LDR R3, =PB
        LDR R5, =PF
        STR R0, [R1, #0x3FC]
        STR R0, [R3, #0x3FC]
        MOV R12, #0x3FF                 ; value to count
        MOV R10, #0x0                   ; value to compare


stop
        ADDS R0, R12, #0xFFFFFFFF       ; set carry bit
        LDR R0, [R5, #0x3FC]            ; PF DATA
        LSR R0, R0, #4                  ; take 2nd bit
        AND R0, #0x1                    ; mask
        CMP R10, R0                     ; check if 0
        BEQ count

        LDR R0, [R3, #0x3FC]
        LSR R0, #4
        AND R0, #0x1
        CMP R10, R0
        BNE begin

        B stop

count
```

```
        SUBS R12, #0x1                  ;count--
        MOV32 R8, #0x562AD              ; delay counter

        LSR R0, R12, #6                 ; take 4 MShB of count
        STR R0, [R3, #0x3FC]            ; write to PB[0:3]
        LSL R0, R12, #2                 ; take 6 LShB of count
        STR R0, [R1, #0x3FC]            ; write to PA[2:7]
delay
        ; stop counting
        LDR R0, [R5, #0x3FC]
        AND R0, #0x01
        CMP R10, R0
        BEQ stop

        ; reset
        LDR R0, [R3, #0x3FC]
        LSR R0, #4
        AND R0, #0x1
        CMP R10, R0
        BNE begin

        SUBS R8, #0x1
        BNE delay
        ;keep counting!
        B count
```