A3: Using the ADC system

Group members:
Priya Wilkhu
Elizabeth Sotomayor
Matthew Sharp
Bekhzodbek Moydinboyer

Project description:

Using the ADC System involves creating a system where an external LED blinks at a frequency controlled by a potentiometer. This assignment requires setting up the LED on a breadboard, connecting a potentiometer as an analog input, and utilizing a microcontroller equipped with an ADC to achieve the desired functionality. The blinking frequency of the LED will be determined by the analog voltage output from the potentiometer. Adjusting the potentiometer will vary the blinking rate of the LED accordingly, allowing for real-time control of the output frequency. We must showcase the system in operation, displaying how the LED blinks at different rates based on the potentiometer's adjustment, and provide clear visuals of the setup to accompany the demonstration.

Pseudocode:

```
LED 1          = pin 1.0   (output)
Potentiometer  = pin 1.2   (input)
Potentiometer is used to control blinking frequency, higher resistance results in longer delays between
LED toggles. As the knob is turned the resistance across the potentiometer changes.

// main loop here
init:
   Setup LED1 as output
   Turn on I/O

   Setup Timer B0 so it counts from 0 to value X
   Enable interrupts so whenever timer reaches X, it causes an interrupt

   Setup ADC for potentiometer input (P1.2 analog input)

convert:
   Start ADC conversion and wait for it to complete
   If conversion not complete, wait
   If complete, move ADC result to a register
   Add offset (1000) to ADC result
   // Amplify rate of change by multiplying it by 8
```

Update Timer B0 CCR0 with new value so that blinking frequency is now dependant on the value from potentiometer

Make a delay so it gives enough time for the LED blinking to occur before the CPU polls for the next ADC conversion result

Repeat ADC conversion

// Interrupt Service Routines
Toggle LED1 whenever an interrupt occurs from timer
Clear CCR0 interrupt flag
Return from interrupt

Code

```
;-------------------------------------------------------------------------------
; MSP430 Assembler Code Template for use with TI Code Composer Studio
;
;
;-------------------------------------------------------------------------------
        .cdecls C,LIST,"msp430.h"       ; Include device header file

;-------------------------------------------------------------------------------
        .def    RESET                   ; Export program entry-point to
                                        ; make it known to linker.
;-------------------------------------------------------------------------------
        .text                           ; Assemble into program memory.
        .retain                         ; Override ELF conditional linking
                                        ; and retain current section.
        .retainrefs                     ; And retain any sections that have
                                        ; references to current section.

;-------------------------------------------------------------------------------
RESET       mov.w   #__STACK_END,SP         ; Initialize stackpointer
StopWDT     mov.w   #WDTPW|WDTHOLD,&WDTCTL  ; Stop watchdog timer


;-------------------------------------------------------------------------------
; Main loop here
;-------------------------------------------------------------------------------

init:
                                ;-- Setup LED1 (Configure P1.0 as output and turn off LED1)
                        bis.b   #BIT0, &P1DIR           ; Set P1.0 as output
                        bic.b   #BIT0, &P1OUT           ; Turn off P1.0 (LED1)
```

```
                bic.w#0001h, &PM5CTL0        ; turn on I/O

                ;-- Setup Timer B0
                bis.w   #TBCLR, &TB0CTL         ; Clear Timer B0
                bis.w   #TBSSEL__ACLK, &TB0CTL  ; Set ACLK as the timer source
                bis.w   #MC__UP, &TB0CTL        ; Set Timer B0 to UP mode
                mov.w   #16384, &TB0CCR0        ; Set CCR0 value for Timer B0. Counting up to 0.5
seconds as default
                bis.w   #CCIE, &TB0CCTL0        ; Enable CCR0 interrupt
                bic.w   #CCIFG, &TB0CCTL0       ; Clear the interrupt flag

                nop
                bis.w   #GIE, SR                ; Enable global interrupts
                nop

                ;-- Setup ADC for potentiometer
                bis.b     #BIT2, &P1SEL0          ; Set P1.2 for ADC input (function select)
                bis.b     #BIT2, &P1SEL1          ; Set P1.2 for ADC input (function select)


                mov.w   #0210h, &ADCCTL0        ; 16 cycles, ADC on
                mov.w   #0220h, &ADCCTL1        ; sampling timer, ADC clock
                mov.w   #0020h, &ADCCTL2        ; 12 bit resolution
                mov.w   #0002h, &ADCMCTL0       ; vref, channel A2

convert:
                mov.w   #0213h, &ADCCTL0        ; turn on ADCENC & ADCSC
simultaneously
                bit.w     #BIT0, &ADCIFG          ; Check if conversion is complete
                jz                convert               ; Wait for ADC conversion
                mov.w   ADCMEM0, R12            ; Move ADC result to R12

                ; Offset the ADC result. If potentiometer is zero,
                ; offset is necessary before doing amplifying/multiplication
                add.w   #1000, R12

                ; Multiply by 8 to make the rate of change change in potentiometer more
noticeable
                rla.w   R12
                rla.w   R12
                rla.w   R12
                mov.w   R12, &TB0CCR0           ; Update Timer B0 CCR0 with new value
                call      #Delay                ; Introduce a delay
                jmp       convert               ; Repeat ADC conversion
```

```
                nop


; delay gives enough time for the LED blinking to occur
; before the CPU polls for the next ADC conversion result
Delay:
                mov.w  #0FFFFh, R10             ; Load delay counter
Delay_Loop:
                dec.w   R10             ; Decrement counter
                jnz             Delay_Loop          ; Continue loop until counter reaches 0
                ret

;-------------------------------------------------------------------------------
; Interrupt Service Routines
;-------------------------------------------------------------------------------

ISR_TB0_CCR0:
                xor.b   #BIT0, &P1OUT       ; Toggle LED1
                bic.w   #CCIFG, &TB0CCTL0   ; Clear CCR0 interrupt flag
                reti                ; Return from interrupt

;-------------------------------------------------------------------------------
; Stack Pointer definition
;-------------------------------------------------------------------------------
        .global __STACK_END
        .sect   .stack

;-------------------------------------------------------------------------------
; Interrupt Vectors
;-------------------------------------------------------------------------------
        .sect   ".reset"            ; MSP430 RESET Vector
        .short  RESET

                .sect   ".int43"                    ; Timer B0 CCR0 Interrupt Vector
                .short  ISR_TB0_CCR0
```

Wiring Diagram:

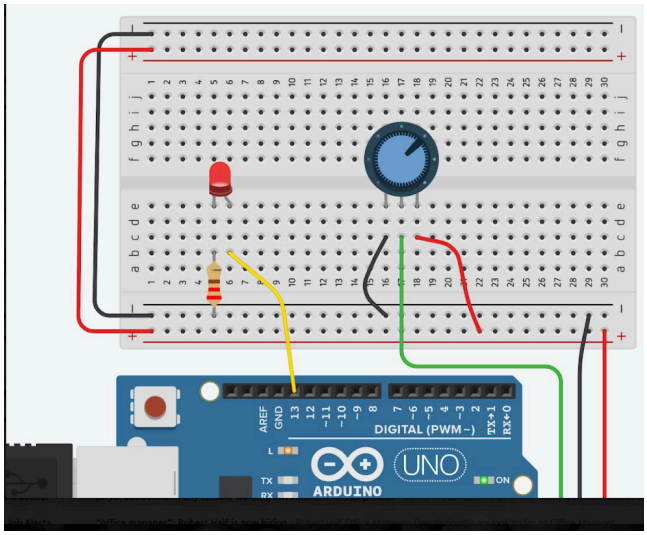
1. **LED Setup:**
    - Connect the anode (longer leg) of the LED to a digital I/O pin on the MSP430FR2355 (P1.0).
    - Connect the cathode (shorter leg) of the LED to one end of the 220Ω resistor.
    - Connect the other end of the resistor to the ground (GND) on the breadboard.
2. **Potentiometer Setup:**
    - Connect the middle pin (wiper) of the potentiometer to an analog input pin on the MSP430FR2355 (P1.2/A2).

- Connect one of the outer pins of the potentiometer to the 5v power rail on the breadboard.
- Connect the other outer pin of the potentiometer to the ground (GND) on the breadboard.

3. **Power and Ground:**
   - Connect the 5v and GND pins of the MSP430FR2355 to the corresponding power and ground rails on the breadboard.

Circuit Diagram:

1. **MSP430FR2355 Connections:**
   - P1.0 → Anode of LED
   - GND → Cathode of LED through 220Ω resistor
   - P1.2/A2 → Middle pin of potentiometer
   - 5V → One outer pin of potentiometer
   - GND → Other outer pin of potentiometer
2. **Breadboard Connections:**
   - Power rail (5v) connected to5V pin of MSP430FR2355
   - Ground rail connected to GND pin of MSP430FR2355



-yellow wire attached to pin 1.0

-green wire attached to p1.2

Video:

https://youtu.be/airkn85PK7Y