ENGR 270 LAB #6 – Autonomous Robot

Objective
Utilize the resources of EDbot and your knowledge of PICmicro Assembly language to build an autonomous robot that moves forward without running into objects.

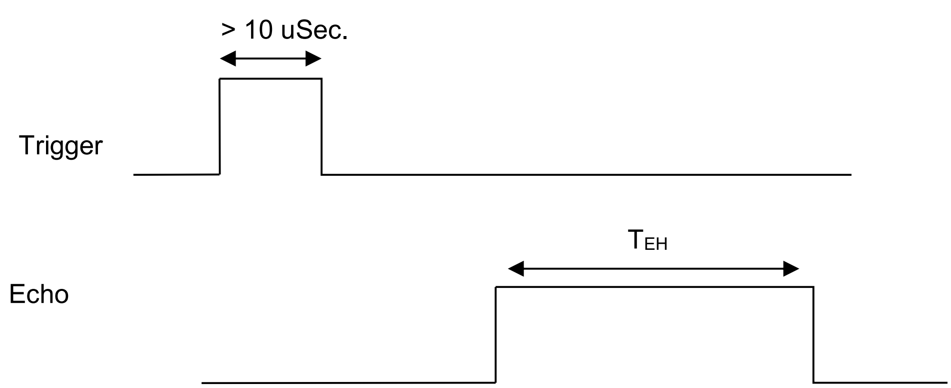
Objective
Application of interrupts, timers and other EDbot resources to solve a more complex problem.

Related Principles
- Computer Organization and Design
- Microprocessors
- Hardware and Software Interface
- Digital Design
- Assembly language

Equipment
- Windows-based PC with MPLAB Simulation Solutions Software
- USB hard disk or other removable drives
- Microchip PICKit programmer
- EDbot V7.0 Platform

Preparation/Background
EDbot includes two HC-SR04 Ultrasonic Ranging Modules, which can be used to estimate distance from objects. By sending a trigger pulse that is at least 10 micro Second to the module and then measuring the duration of echo pulse as shown by the following equation:

Distance (Inches) =(Echo Pulse high, T_{HE} in uSec)/148
Note: Detection angle is 15 degrees and distance range is from 1 to 150 inches.



Below is an example code that sets EDbot's PICmicro oscillator speed to 4 MHz (T_{OSC} = 0.25 uSec) and measures the distance from any objects using only left sensor. The LED will lit up when the object is within 5 inches of the left sensor.

```
;-----------------------------------------------------------------------------------------------------
; Demonstrate use of Sensors to detect distance from objects
; LAST UPDATE: 6/15/2016
; AUTH: Class
; DEVICE: PICmicro (PIC18F1220)
;;-----------------------------------------------------------------------------------------------------

    list        p=18F1220                           ; processor type
    radix       hex                                 ; default radix for data
    ; Disable Watchdog timer,  Low V. Prog, and RA6 as a clock
    config      WDT=OFF, LVP=OFF, OSC = INTIO2

#include    p18f1220.inc

#define     lastL       0x80                ; Last L Sensor Value
#define     loopCount   0x81                ; Timer Loop Count
#define      countL     0x82                ; Count the cycles we have had echoL on
#define      countOD    0x83                ; Count for outer delay loop
#define      countID    0x84                ; Count for inner delay loop

;these are shortcuts, string replacements
#define     _TrigL      PORTA,RA1
#define     _TrigR      PORTA,RA4
#define     _EchoL      PORTA,RA0

    org         0x000       ; Executes after reset, equivalent to org
    GOTO        StartL

    org 0x008   ; Executes after high priority interrupt
    GOTO        HPRIO

    org 0x020                   ; Start of the code

HPRIO:
    BTFSC       PIR1, TMR2IF ; high priority loop
    BRA         iLoop
    RETFIE              ; return from interrupt

iLoop:
    INCF         loopCount
    MOVLW       .120
    CPFSLT       loopCount
    BRA         doTrigger                   ; trigger every 30,000 uSec.

    MOVLW       .1
    CPFSGT      loopCount
    BRA         stopTrigger

    ; we didn't trigger so update
    BRA         updateSensor

doTrigger:
    CLRF        loopCount
    BRA         doTriggerL

doTriggerL:
    MOVFF       countL,lastL
    ; we should check to see if echo is high and kill trigger if that's the case.
    BTFSC       _EchoL
    BRA         killL

continueL:
    BSF          _TrigL         ; Set Left trigger on
    CLRF        countL           ; clear count of eccho
    BRA          loopDone

killL:          ; Sensors is known to hang whne when no object is found within its
                ; Measurement range - Noise is known to reset the sensor.
                ; So here, we are using the left sensor to reset right sensor.
                ; Sensors work best with 4.5-5.5 v supply voltage.
    BSF             _TrigR       ; start trigger or right sensor
    MOVLW   .1                  ; 1 millisecond
    CALL    Delay
    BCF             _TrigR      ; Clear right trigger on
    MOVLW   .1                  ; 1 millisecond
```

```
    CALL    Delay

    ; If Echo is not cleared then try to reset it again
    BTFSS   _EchoL
    BRA     continueL
    BRA     killL

stopTrigger:
    BCF         _TrigL          ; Set Left trigger off
    BRA         loopDone

updateSensor:
    ;increment count for each cycle echo is on
    btfsc       _EchoL
    incf        countL
    bra         loopDone

loopDone:
    bcf         PIR1, TMR2IF ; Clear Timer 2 interrupt Flag
    bra         HPRIO         ; Go to start and service any pending Interrupt

StartL:
     ; Initialize all I/O ports per EDbot Specifications
    MOVLW       0x7F
    MOVWF       ADCON1          ; Set all Port A Pins as digital
    CLRF        PORTA           ; Initialize PORTA
    CLRF        PORTB           ; Initialize PORTB
    MOVLW       0x0D
    MOVWF       TRISA           ; Set Port A direction
    MOVLW       0xC7
    MOVWF       TRISB           ; Set Port B direction

    MOVLW       0x60
    IORWF       OSCCON        ; Set to 4mhz

    ; Clear Sensor related counter
    CLRF        lastL
    CLRF        loopCount

    BSF         INTCON, PEIE                    ; enable peripheral interrupts

    ; Enable Timer2 Interrupat as high priority
    BSF         PIE1, TMR2IE
    BSF         IPR1, TMR2IP

    CLRF        TMR2
    CLRF        T2CON           ; Timer 2 is set to 8-bit with no scaling
    MOVLW       0xFA            ; Timer 2 is set to interrupt in 250 uSec.
    MOVWF       PR2

    BSF         T2CON,TMR2ON                    ; enable TMR2
    BSF         INTCON, GIE                     ; enable interrupts globally

Mloop:
    BCF         PORTB,RB5       ; turn off LED
    MOVLW       .2              ; this is the distance we are checking for
    CPFSGT      lastL           ; skip if LastL > wreg
    BSF         PORTB,RB5       ; turn on LED
    BRA         Mloop

;Function to delay for Wreg miliseconds
Delay:
    MOVWF       countOD
DelayOL:                        ; delay Outer loop
    CLRF        countID
DelayIL:                        ; Delay Inner Loop
    NOP
    INCF        countID
    BNZ         DelayIL
    DECF        countOD
    BNZ         DelayOL
    RETURN                      ; end delay function

   end                          ; end of code
```

Experiment #1

Use the sensor sample code provided earlier to develop an EDbot code that would performs the following steps:

1) Move forward until an object is detected within 10 inches
2) Moves straight back for 0.5 seconds
3) Turns 30 degrees
4) Go to step 1

Experiment #2

Write an assembly code for EDbot that would drives EDbot forward (not circular) for a minimum of 20 seconds without hitting any object in its path.

This experiment requires that you review your high level design (flow chart or pseudo code) and demonstrate your system to the instructor upon completion. Include the approval signature in your report.

Report Requirements

All reports must be computer printed (formulas and diagrams may be hand drawn) and at minimum include:

For each experiment:

a) Clear problem statement; specify items given and to be found.
b) Specific responses to each question asked in the experiment.
c) Documentation of resulting high level design, disassembled code, system diagram, schematics and any other supporting material.

For the report as a whole

a) Cover sheet with your name, course, lab title, date of completion and your teammates' name.
b) Lessons learned from this lab.
c) A new experiment and expected results which provide additional opportunity to practice the concepts in this lab.