

Ga-wun Kim

ENGR 270

Scott Koss

Lab Partners: Rebecca, Kantas

Computer and Organization & Microprocessors Lab #6

Introduction

In this lab, we were required to utilize the resources of EDbot and my knowledge of PICmicro Assembly language to build an autonomous robot that moved forward without running into objects. We mainly concerned about application of interrupts, timers and other EDbot resources. Based on the background,

Experiment 1

In the experiment1, we used the sensor sample code provided earlier to develop an EDbot code that would perform 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

The below codes are what we did.

Configured Code:

```
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 PortA 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 Interrupt 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, 5 ; turn off LED
BSF PORTB, 4 ; Enable Right motor
BSF PORTA, 6 ; Forward Right
BSF PORTB, 3 ; Enable Left Motor
BSF PORTA, 7 ; Forward Left
WaitBlock:
MOVLW .5 ; this is the distance we are checking for
CPFSGT lastL ; skip if LastL > wreg
GOTO Backwards
BRA WaitBlock
;Function to delay for Wreg miliseconds
Backwards:
BSF PORTB, 5 ; turn on LED
BSF PORTB, 4 ; Enable Right motor
BCF PORTA, 6 ; Backward Right
BSF PORTB, 3 ; Enable Left Motor
BCF PORTA, 7 ; Backward Left
MOVLW .162 ; about .25 seconds
CALL Delay ; (((1+1+1+1+2)*256+1+1+2)*4*(1/4mhz) = .249483 approx.
MOVLW .162 ; so do it twice for .5 seconds
CALL Delay
TURN:
BSF PORTA, 6 ; Forward Right
BCF PORTA, 7 ; Backward Left
MOVLW .100 ; roughly 30 degree turn
CALL Delay
wait2:
BCF PORTB, 4 ; turn on LED
BCF PORTB, 3 ; Enable Left Motor
MOVLW .5 ; this is the distance we are checking for
CPFSGT lastL ; skip if LastL > wreg
GOTO Backwards
BRA Mloop
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 2

We were required to write an assembly code for EDbot that would drive EDbot forward for a minimum of 20 seconds without hitting any object in its path.

```

Code:
list p=18F1220 ; processor type
radix hex ; default radix for data
; Disable Watchdog timer, Low V. Prog, and RA6 as aclock
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 HPRI0
org 0x020 ; Start of the code
HPRI0:
BTFSC PIR1, TMR2IF ; high priority
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 echo

BRA loopDone

killL: ; Sensors is known to hang 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 on 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

bra HPRIO ; Go to start and service any

StartL:

; Initialize all I/O ports per EDbot Specifications

MOVLW 0x7F

MOVWF ADCON1 ; Set all Port

CLRF PORTA ; Initialize PORTA

CLRF PORTB ; Initialize PORTB

MOVLW 0x0D

MOVWF TRISA ; Set Port A

MOVLW 0xC7

MOVWF TRISB ; Set Port B

MOVLW 0x60

IORWF OSCCON ; Set to 4mhz

; Clear Sensor related counter

CLRF lastL

CLRF loopCount

BSF INTCON, PEIE ; enable

; Enable Timer2 Interrupt 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
BSF INTCON, GIE ; enable interrupts
Mloop: ;enable wheels, turn off LED
BCF PORTB,5 ; turn off LED
BSF PORTB,4 ;Enable Right motor
BSF PORTA,6 ;Forward Right
BSF PORTB,3 ;Enable Left Motor
BSF PORTA,7 ;Forward Left
WaitBlock:
MOVLW .5 ; this is the distance we are checking for, approx 10 in.
CPFSGT lastL ; skip if LastL > wreg; or about 10 inches.
GOTO Backwards
BRA WaitBlock
Backwards: ;it was < 10 inches, so it goes here.
BSF PORTB,5 ; turn on LED
BSF PORTB,4 ;Enable Right motor.
BCF PORTA,6 ;Backward Right
BSF PORTB,3 ;Enable Left motor
BCF PORTA,7 ;Backward Left
MOVLW .250 ;0.38 seconds.
CALL Delay
TURN:
BSF PORTA,6 ;Forward Right
BCF PORTA,7 ;Backward Left
MOVLW .100 ;0.15 seconds. Both wheels turn so it turns in place to not hit a
CALL Delay ; a wall. Both wheels turning is a really fast turn. .15 is fine.
wait2:
BCF PORTB,4 ; turn on LED
BCF PORTB,3 ;Enable Left Motor
MOVLW .5 ; this is the distance we are checking for
CPFSGT lastL ; checks again to see if it is next to a wall
GOTO Backwards ;if it is it goes backwards and turns again
BRA Mloop ;If not it goes forward.
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

```

What we learned from this lab.

- How to make the robot move and change itself, automatically without any shape, by using sensor.

- How to arrange the echo pulse high for the recognition of the distance in C language.

Conclusion (New experiment)

This lab was pretty interesting experiment because we used the sensor for the first time in EDbot. Last year summer quarter, I saw some people who were trying to arrange the sensor of EDbot. At that time, I was curious how did it work and what code were needed. After this lab, finally I could realize the principal of that robot.

Actually, this lab was the last one during the 270 class. We spent much of time to do this lab; when we were done this lab, we could not get the signature from the lab tutor because, at that time, there was no person to be able to give a signature. So I could not attach the signature of the tutor.