

Ga-wun Kim

ENGR 270

Scott Koss

Lab Partners: Rebecca, Kantas

Computer and Organization & Microprocessors Lab #3

Introduction

In this experiment, the EDbot platform is introduced; we are required to figure it out, and, with the base of the prior experiments, we ask to modify the given code and to build the codes.

Experiment 1

We inserted the provided code in to the EDbot; we could observe the EDbot to move forward, back, and stop. Also, we were careful not to drop it.

Experiment 2

We modified the code in the experiment 1 in order for EDbot to drive an ellipse; pattern took between 5 to 15 seconds to complete. The code that we inserted into the EDbot is into the below box.

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

#include p18F1220.inc ; This header file includes address and bit definitions for all SFRs
#define dCount 0x80
#define dCountInner 0x81
org 0x000 ; Set the program origin (start) to absolute 0x000
; Initialize all I/O ports
CLRF PORTA ; Initialize PORTA
CLRF PORTB ; Initialize PORTB
MOVLW 0x7F ; Set all A/D Converter Pins as
MOVWF ADCON1 ; digital I/O pins
MOVLW 0x0D ; Value used to initialize data direction
MOVWF TRISA ; Set Port A direction
MOVLW 0xC7 ; Value used to initialize data direction
MOVWF TRISB ; Set Port B direction
MOVLW 0x00 ; clear Wreg
; Toggle Portb,5, direction, and delay.
; start by going forward for first delay cycle

Turnhalf:
BSF PORTB,4 ;Enable Right motor
BSF PORTA,6 ;Forward Right
BSF PORTB,3 ;Enable Left Motor
BSF PORTA,7 ;Backward Left
MOVLW .5 ;Move forward
CALL Delay

BCF PORTB,4 ;1st turn
MOVLW .1
CALL Delay

BCF PORTB,4 ; forward
MOVLW .1
CALL Delay

;half ellipse

BSF PORTB,4 ;Forward
MOVLW .1
```

```

CALL Delay

BCF PORTB,4 ;2nd turn
MOVLW .2
CALL Delay

BSF PORTB,4 ;Forward
MOVLW .3
CALL Delay

BCF PORTB,4 ;1st turn
MOVLW .2
CALL Delay

BSF PORTB,4
MOVLW .5 ; Move forward
CALL Delay

BRA Turnhalf

Delay:
MOVWF dCount
DelayLoop:
CALL DelayOnce
DECF dCount
BNZ DelayLoop
RETURN

DelayOnce:
CLRF dCountInner ;Internal delay loop
DelayOnceLoop:
NOP
INCF dCountInner
BNZ DelayOnceLoop
RETURN
end ; code end

```

After complete inserting the code, we separated the EDbot and USB; and confirmed this object motion. It drove an ellipse shape as its track; however, this shape was very similar with the direction (actually, it was not perfect ellipse that we expected), and occasionally, it leaved away from its own track.

Learn from this lab

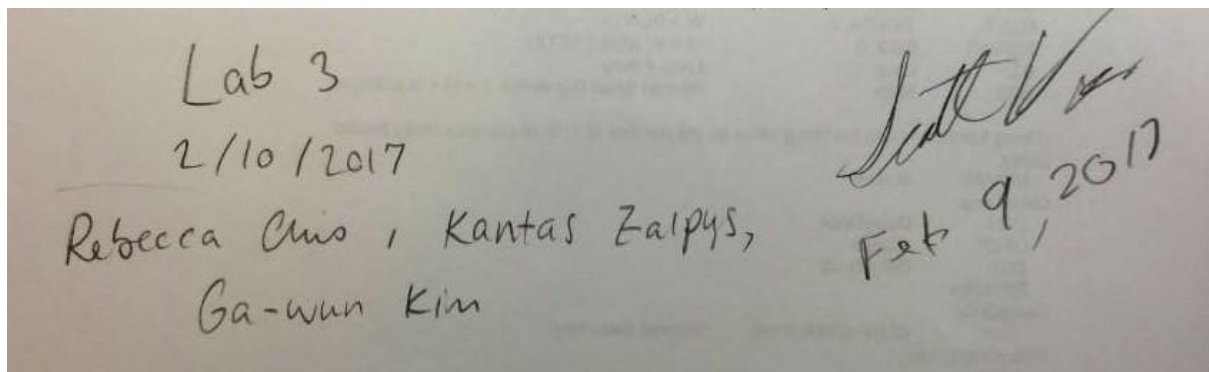
- How to design the code to make the object move the expected track shape by using time delaying (Specifically, “CALL delay”).
- What is EDbot, the structure of this object in an aspect of using PIC18F1220 chip, and how to use it.

Conclusion (New experiment)

When we were about to start this lab, we had no idea because we had never used this object before. However, I could figure it out after read all directions and I could implement this code. However, ironically, it did not take too much of time to finish this lab. I guess it was based on

the PIC18F1220 chip and, from the prior labs, we got used to using this chip so that we could save much of time. It was interesting experience for me because, before this lab, we only used the protoboard that I designed; but that was not expected to move itself. However, in this lab, by using the code, I was glad to make the robot move itself by the code.

Signature for certificate.



Lab 3
2/10/2017
Rebecca Chis, Kantas Zalpyis,
Ga-wun Kim
Feb 9, 2017