ENGR 270 LAB #3 – EDbot Micro Introduction

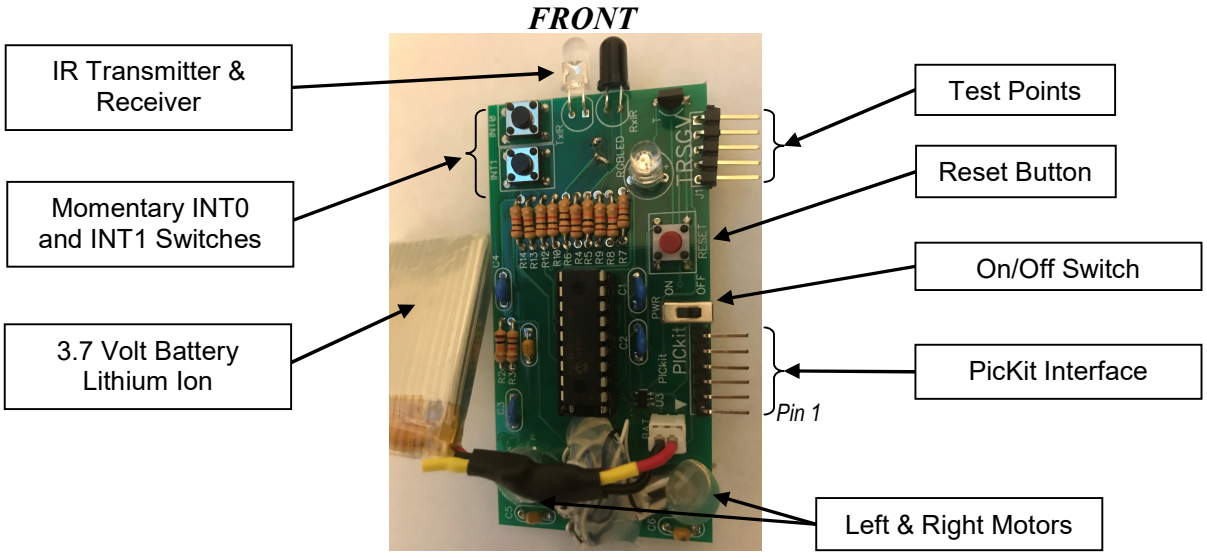
Objective
Introduce the EDbot Micro platform and use of broader range of assembly instructions and constructs.

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

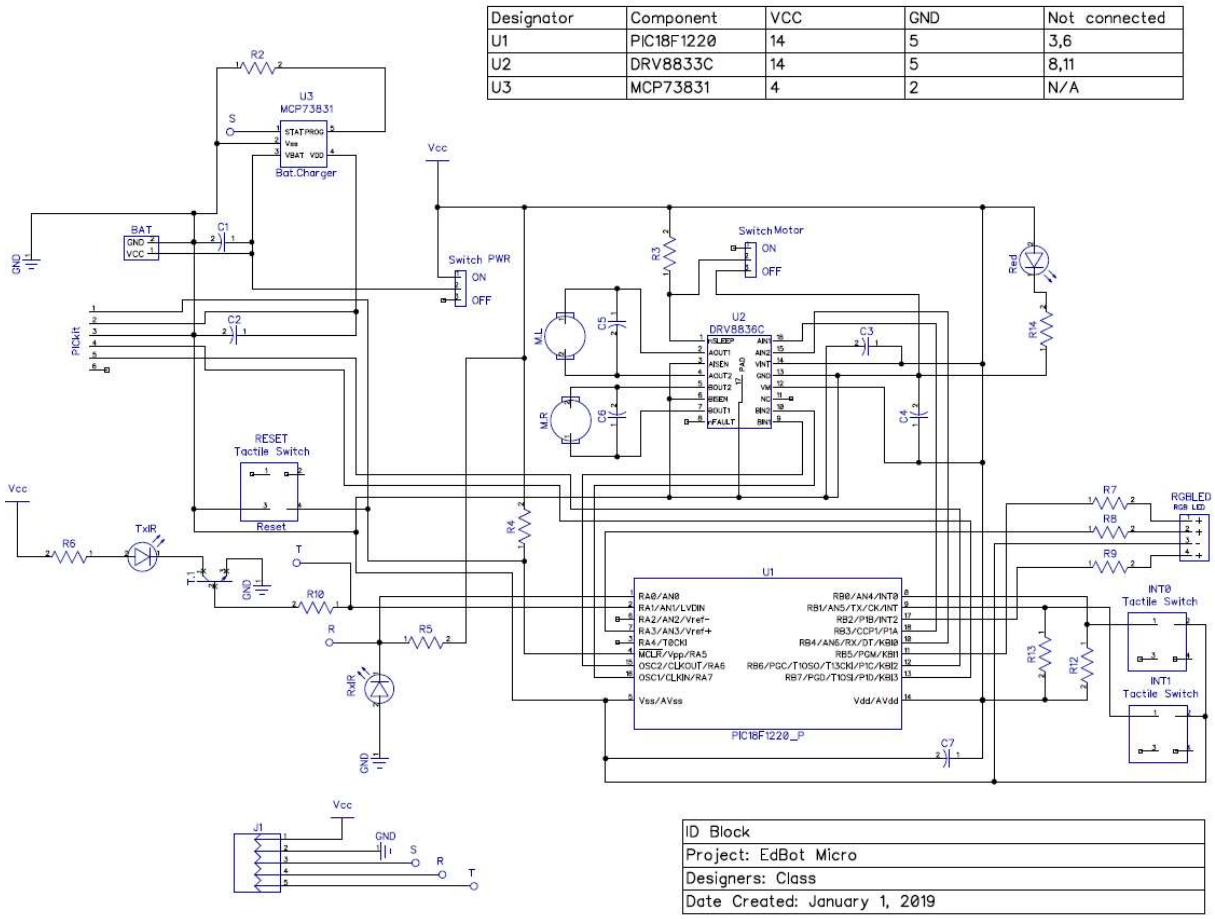
Equipment
- Windows-based PC with MPLAB Simulation Solutions Software
- USB hard disk or other removable drives
- Microchip PICKit programmer
- EDbot Micro V12 Platform

Supplies
- None

Preparation/Background
EDbot Micro was designed and implement by past students based on the learning from this course. The design is similar to the work done in labs 1 and 2. The following diagram outlines the physical design and labels major components of EDbot Micro:



EDbot Micro uses PIC18F1220 as the microcontroller with two independent DC motors and One sets of infrared transmitors (IR333-A) and receivers (BPV10NF) which makes EDbot Micro a highly flexible robotic platform. EDbot Micro V10.0 schematic follows:



EDbot Micro has specific assignment for all the PICmicro I/O Pins as follows:

Registers <bit #>	Pin # - Name -Type	Function
Port A <0>	1 – RA0 - input	Infrared (IR) Receiver
Port A <1>	2 – RA1 - output	Infrared (IR) Transmitter
Port A <3>	7 – RA3 – output	RGB LED- Green
Port B <5>	11 – RB5 - output	RGB LED - Blue
Port B <2>	17 – RB2 - output	RGB LED - Red
Port A <6>	15 – RA6 - output	Motor – right (RA6, RA7 direction)
Port A <7>	16 – RA7 - output	(1,0 forward) (0,1 backward) (0,0 or 1,1 stop)
Port B <3>	18 – RB3 - output	Motor – left (RB4, RB3 direction)
Port B <4>	10 – RB4 - output	(1,0 forward) (0,1 backward) (0,0 or 1,1 stop)
Port B <0>	8 – RB0 - input	Interrupt 0 Button (INT0)
Port B <1>	9 – RB1 - input	Interrupt 1 Button (INT1
Port A <5>	4 – RA5 – input	PIC Reset switch (MCLR')
VCC	14	Vcc "3.7-4.2 V"
GND	5	Ground
Port A <2>	6 – RA2 - Input	Not Used
Port A <4>	3 – RA4 - Input	Not Used
Port B <6>	12 - RB6 - Input	Not Used
Port B <7>	13 – RB7 - Input	Not Used

Experiment #1

Create a new MPLAB project using the code provided on the next page. Build the project and program EDbot Micro. Write a summary of EDbot Micro operation based on your review of the code, schematics and observation of EDbot Micro while executing the code.

Notes:

1) During the Programming and execution EDbot Micro motors will rotate. Be careful not to drop it!

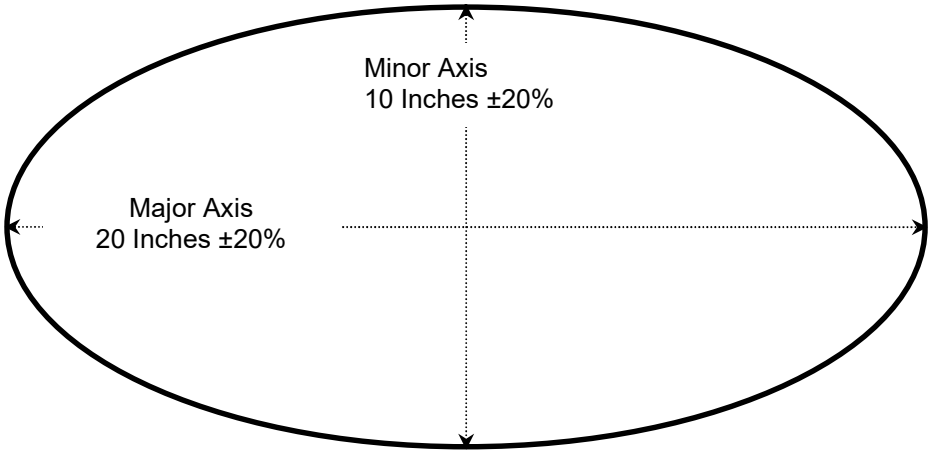
2) Microchip header file supplied with MPLAB® IDE contains the definition for all the SFR register addresses and bit names in addition to commonly used constant values listed in the course text appendix. You can use SFR register names and bit names by adding the following statement in your code to include the header file;

```
# include p18f1220.inc
```

3) In project property, select Pickit and change power setting to “power target” and “4.25 v”.

Experiment #2

Modify the code in experiment 1 in order for EDbot Micro to drive an ellipse (per specifications shown below) pattern once clockwise while RGB LED is green. Upon completion, reverse direction and complete the pattern once counter clockwise while RGB LED is blue.



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.

Team Members: • • • •	LAB3 Demo Instructor Approval Signature & Date:

```
;-------------------------------------------------------------------
; Desc: Use Int0 Switch to click through all the functionality:
;           1) LED Blue is on
;           2) LED Green is on
;           3) LED Blue is on
;           4) Right motor on forward
;           5) Left motor on forward
;           6) IR sensor test
;              LED is Red when IR is facing an obstacle
;              LED is not green when no obstacle
;
; Last Update: January 16, 2019
;-------------------------------------------------------------------

   list    p=18F1220          ;processor type
   radix   hex                ;default radix for data
   config  WDT=OFF,LVP=OFF,OSC=INTIO2

#include    p18F1220.inc              ;header file

#define     dCount          0x80
#define     dCountInner     0x81
#define     LastValue   0x82

   org      0x000 ; Set program origin to absolute 0x000
   ; initialize all I/O ports
   CLRF    PORTA           ; Initialize PORTA
   CLRF    PORTB           ; Initialize PORTB
   MOVLW   0x7F
   MOVWF   ADCON1      ; Configure PortA<0:7> Digital and PortA<> Analog
   MOVLW   0x35
   MOVWF   TRISA           ; Set Port A direction Per EDbot Micro Spec.
   MOVLW   0xC3
   MOVWF   TRISB           ; Set Port A direction Per EDbot Micro Spec.

   ; Use INT0 switch to click through all functionality
Main:
   BSF     PORTA,3         ; Step 1 - Turn on green LED
   CALL    Int0Press
   BCF     PORTA,3         ; Turn off green LED
   BSF     PORTB,5         ; Step 2 - Turn on blue LED
   CALL    Int0Press
   BCF     PORTB,5         ; Turn off Blue LED
   BSF     PORTB,2         ; Step 3 -Turn on red LED
   CALL    Int0Press
   BCF     PORTB,2         ;Turn off LED
   BSF     PORTA,7         ; Step 4 - Forward Right Motor
   BCF     PORTA,6
   CALL    Int0Press
   BCF     PORTA,7         ; Stop Right Motor
   BSF     PORTB,3         ; Step 5 - Forward Left Motor
   BCF     PORTB,4
   CALL    Int0Press
   BCF     PORTB,3         ; Stop Left Motor

SensorT:                   ; Step 6 - Sensor Test
   BSF     PORTA,1         ; Turn On IR Transmitter
   BTFSC   PORTA,0         ; check IR Receiver
   BRA     NotBlocked
Blocked:
   BCF     PORTA,1         ; IR Transmitter off
   BCF     PORTA,3     ; Green LE off
   BSF     PORTB,2         ; Red LED on
   BRA     SensorT
NotBlocked:
   BCF     PORTA,1         ; IR Transmitter off
```

```
   BSF      PORTA,3    ; Green LED on
   BCF      PORTB,2       ; Red LED off
   BRA      SensorT

MainDone:
   BRA      MainDone       ; Done


; Delay function will delay by (Wreg/10) seconds
Delay:
   MOVWF   dCount
DelayLoop:
   CALL    DelayOnce
   DECF    dCount
   BNZ      DelayLoop
   RETURN
DelayOnce:
   CLRF    dCountInner     ; Internal delay loop
DelayOnceLoop:
   NOP
   INCF    dCountInner
   BNZ      DelayOnceLoop
   RETURN  ; Delay Function

; Wait until INT0 Button is pressed (include SW Debounce)
Int0Press:
   MOVLW   .1
   CALL    Delay
   MOVF    PORTB,0
   ANDLW   0x01
   BZ       Int0Press ; wait for button to be released
Int0PressZ:
   MOVLW   .1
   CALL    Delay
   MOVF    PORTB,0
   ANDLW   0x01
   BNZ      Int0PressZ ; wait for buton to be pressed
   RETURN  ; Int1Press Function

 end      ; code end
```

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.