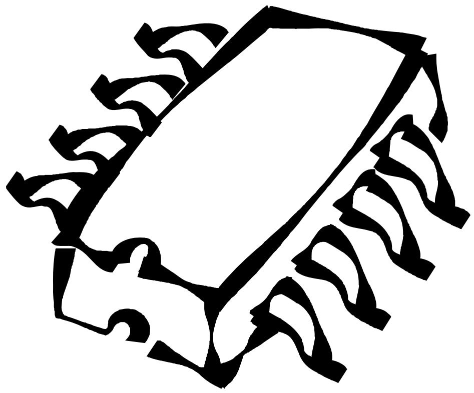


IEEE/CETA Robotics Workshop

Lab Manual
(Day 4)
How to Program the Inputs and Outputs

Dennis Cecic, P. Eng.
TISP Coordinator, IEEE Toronto Section
(d.cecic@ieee.org)

Brad North
Computer Engineering Teacher
Rick Hansen Secondary School
(brad.north@peelsb.com)

Disclaimer

The information in this workshop is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor IEEE shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Introduction – IEEE/CETA Robotics Workshop

Robot design and fabrication is an extremely rewarding and exciting activity for students of all ages.

Targeting high-school computer engineering technology teachers, this hands-on workshop uncovers the basic principles involved in construction and programming of a simple autonomous line following robot. The activity satisfies many aspects of the Ontario's TEJ/ICS curricula, and it is hoped that the teacher will be equipped to integrate this activity into his/her TEJ/ICS curriculum activities and also to field a team to compete in the annual CETA (Computer Engineering Teachers Association – Dufferin-Peel) Regional Robotics Competition.

The robot platform is based on a modern 16-bit microcontroller (PIC24), and is programmed using standard tools and languages **used in industry**. The robot platform provides a cost effective system for exploration of a variety of topics in TEJ/ICS curricula, such as analog/digital interfacing, motor control, computer networking, embedded control & mathematical algorithms, as well as both C and assembly language programming.

Training Objectives:

After the workshop the attendee shall,

- be able to develop embedded control C programs for the PIC24 MCU,
- be able to fabricate the robot platform used in the CETA competition,
- be able to configure the MCU inputs and outputs required for the robot,
- be able to implement a basic line-following control algorithm used in the CETA competition

The training consists of 5, 1-day modules, each of which consist of a PowerPoint presentation, with a corresponding instructional lab manual and lab projects. All materials are provided on the student CDROM.

Workshop Outline (Summary):

Day 1. How to Program in C – Day 1
Day 2. How to Program in C – Day 2
Day 3. How to Build the Robot Platform
Day 4. How to Program the Inputs and Outputs
Day 5. How to Control the Robot

Prerequisites:

You should ideally have some basic experience in electronics as well as microcontroller programming, and be aware of basic safety rules when working with electronics.

Objective (Day 4)

The objective of this module is to cover the fabrication and basic programming of the electronic controls for the robot.

Training Objectives:

After this module, the attendee shall,

- understand the PIC24 Architecture and Programmer's Model, and how to create embedded C programs for it, using the 16-bit Microstick development board,
- be able to fabricate the basic electronic control circuit for the robot,
- be able to write a program to drive a small DC motor,
- be able to modify the program to accept optical sensor input to control the DC motor

Detailed Outline:

Development Platform
Lab 1 – Wiring the Board
Very Short Review of the PIC24 Architecture
MPLAB C for PIC24 Compiler Toolset Overview
Lab 2 – Creating PIC24 C Projects in MPLAB
Data Input and Output: Working with Digital I/O Ports
Basics of DC Motors and Their Control
Timer & Pulse-Width Modulation Peripherals
Lab 3 – Motor Control using the PWM Peripheral
Introduction to Sensors
Optical Sensors used in Line Following Robots
Analog-to-Digital Converter Peripheral
Lab 4 – Calibration of Line Detection Using ADC

Release Notes

v001 Summary (11 Feb 2011):

- Initial release of Day 4 material (outline 004). Targeted for ACSE PIC24 workshop in Toronto on February 11, 2011. Coverage limited to Labs1-3.
- Schematic and bill of materials for this release are provided in Appendix A & B respectively.

Initial Installation/Set-Up

Purpose:

Set-up your PC to run the labs. Build/Run a simple program on your PIC24 Microstick.

Equipment & Software:

1. (**Required**) PC running Windows 7 or Windows XP Professional with Service Pack2 (SP2) or higher.
2. (**Required**) An **externally powered USB Hub**, connected to the PC. Although we will add a protection fuse to protect the USB's power supply, you may wish to safeguard your PC's USB port by connecting the Microstick to an external hub.
3. (**Required**) Adobe Acrobat PDF Reader and Microsoft Word. Microsoft's Word reader may be found here: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=3657ce88-7cfa-457a-9aec-f4f827f20cac&displaylang=en
4. Line Follower Robot "Electronic Control" Component Kit (see **APPENDIX B**), including the 16-Bit Microstick development board.
5. Day 4 Student CDROM
6. (**Required**) MPLAB IDE v8.53 (www.microchip.com/mplab) & MPLAB C for PIC24 Compiler v3.24 "Lite" version (www.microchip.com/mplabc) - Both are available on the Student CDROM in the "/Development Tools" sub-folder
7. (Recommended) Basic multimeter (dc voltage/current + frequency counter function)
8. (Recommended) Oscilloscope.
9. (**Required**) Pliers & wirecutter/stripper
10. (**Required**) Solder Station (suggest Weller WD1001)

Software Installation Procedure:

1. Using Windows' standard program uninstallation procedure, uninstall any existing MPLAB IDE and/or MPLAB C for PIC24 compiler versions.

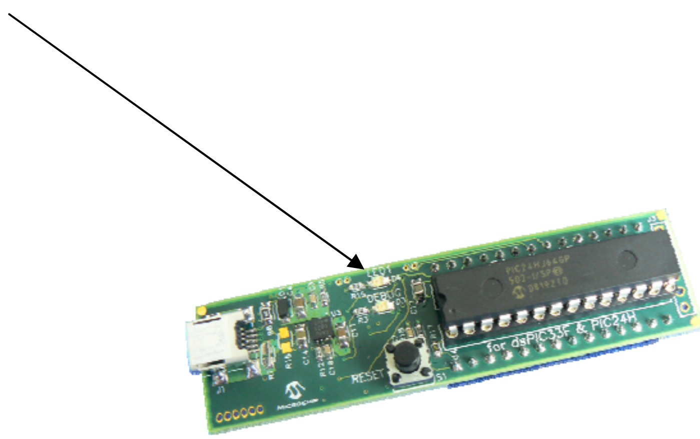
2. Copy all the files/folders from the student CDROM to **C:\IEEE\IEEECETA_WS**

3. Install the MPLAB IDE and MPLAB C for PIC24 Compiler (use the default paths). Double-click on the self-extracting .exes found in \Development Tools

4. Test the basic software installation by connecting the Microstick to your PC, then opening/building/running Lab 2's solution project/workspace :

    a. Connect the Microstick to an available USB Port
    b. Start MPLAB IDE
    c. Open the Lab 2 solution workspace (File → Open Workspace) "C:\IEEE\IEEECETA_WS\Day 4\Lab 2\Solution\Lab 2.mcw"
    d. Select the Microstick as the Debugger (Debugger → Select Tool → Starter Kits)
    e. Build the project (Project → Build All)
    f. Program the board (Debugger → Program)
    g. Run the program (Debugger → Run)

LED1 should blink!



Class Folder Structure

Root Folder: "C:\IEEE\IEEECETA_WS"

"C:\IEEE\IEEECETA_WS\Day 4\Lab1..Lab4"

Contains the Class Labs – Solution projects/workspaces are provided in the **\Solution** sub-folder

"C:\IEEE\IEEECETA_WS\Day 4\Presentation & Handouts"

Contains .pdf copies of the slides and lab manual.

"C:\IEEE\IEEECETA_WS \Users Guides & Data Sheets"

Repository for helpful documents while doing the labs, such as MCU data sheets and development board schematics

"C:\IEEE\IEEECETA_WS \Development Tools"

Contains the MPLAB IDE and PIC24 C-Compiler, plus any other useful tools used in the training.

"C:\IEEE\IEEECETA_WS \Help"

Contains useful help information

Lab 1 – Wiring the Board

Purpose:

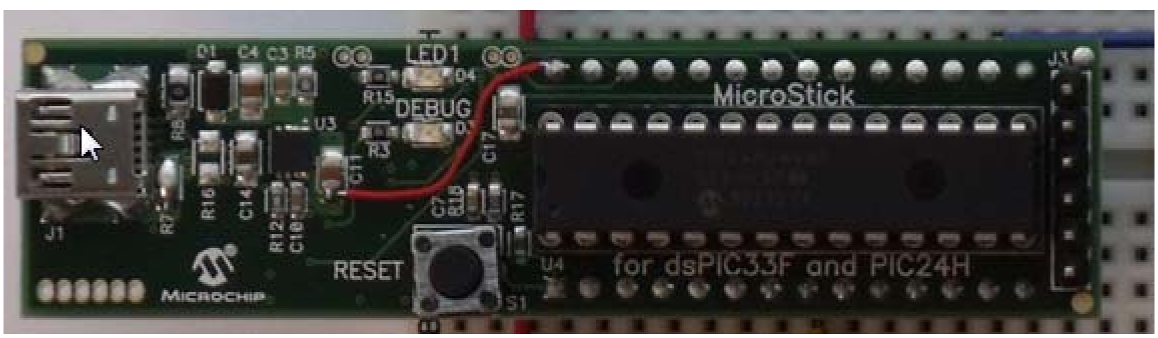
To understand the basic fabrication procedure of the line follower robot's control circuitry using the PIC24 Microstick.

First, you will make the necessary modification to the 16-bit Microstick board to route the 3.3v supply onto pin 28 of the target socket. Next, you will insert the appropriate components into the solderless breadboard as indicated in the schematic and pictures. Finally, you will insert the Microstick, Motor battery pack, and motor into the board, and program working demo code into the PIC24 device to test your completed circuit.

You will then confirm the operation of your circuit by controlling the motor's speed using the potentiometer.

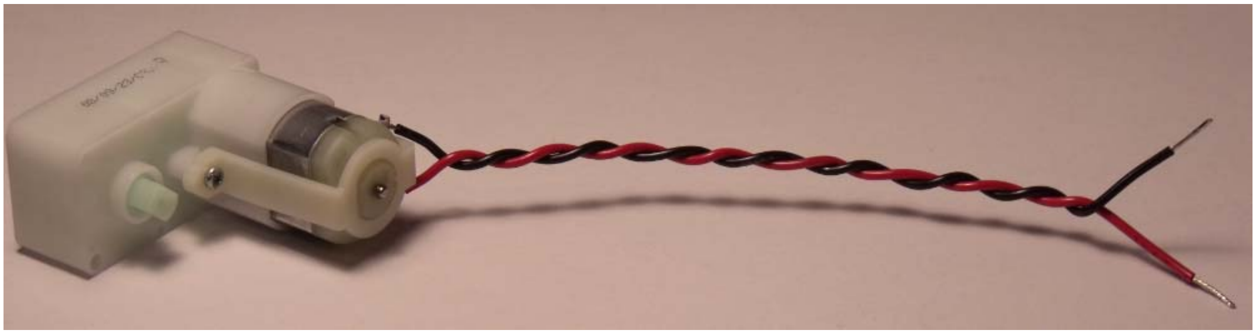
Procedure:

1. Using a small (~2-3cm) piece of #30 AWG wire, make a soldered connection between the Microstick's 3.3v power supply and pin 28 of the target header as shown here:

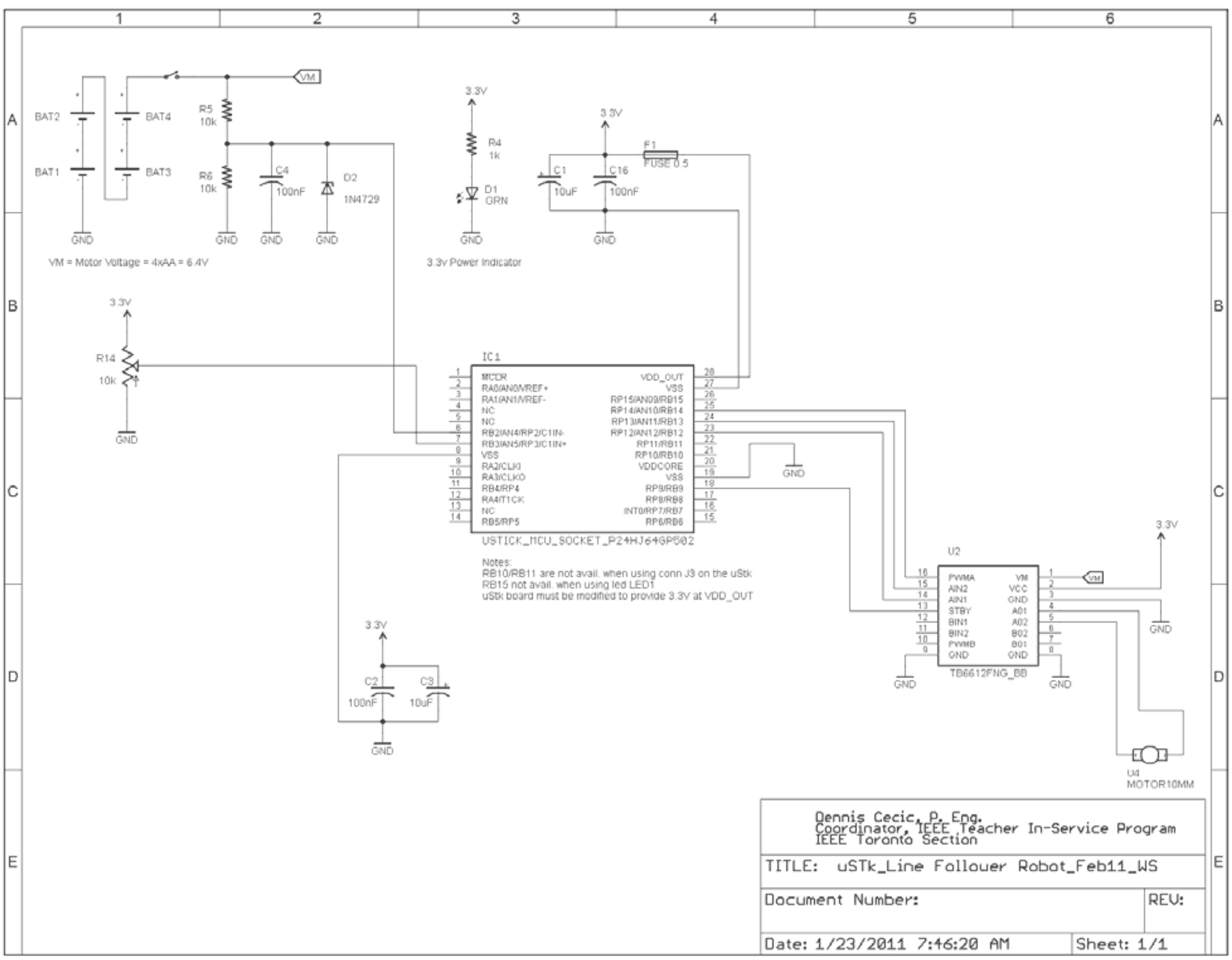


2. Break Off 2, 8-pin sections from the 36-pin header provided in the kit and solder them onto the TB6612 motor driver board as shown here:

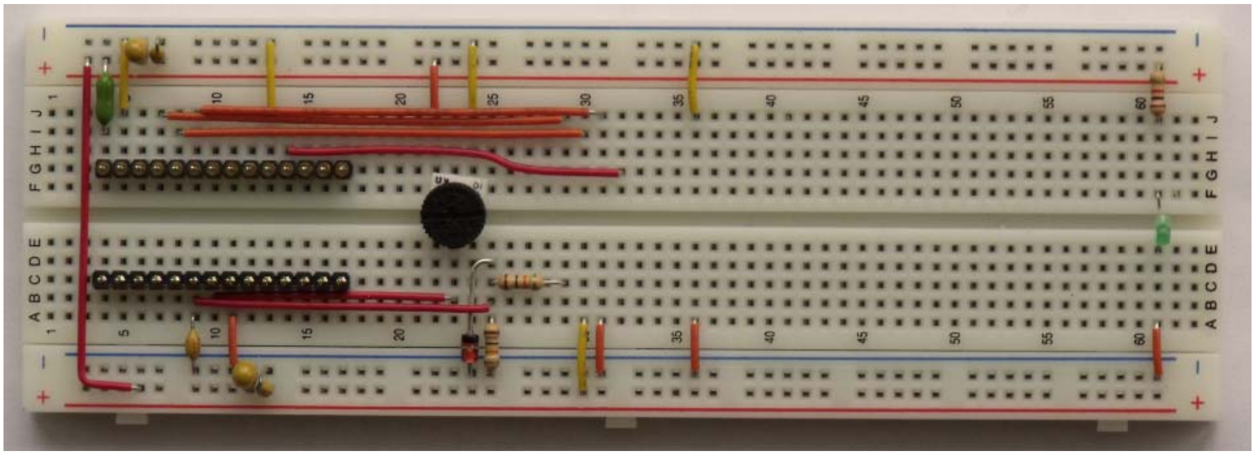
3. Using wire strippers, strip off about 1cm from each end of the Red and Black #22AWG wires. Tin the ends, twist the wires together and solder them to the DC motor as shown here (do not install the wheel):



4. Assemble the circuit shown below. The mating connectors for the target MCU are inside the Microstick package. See the next page for the assembly steps:



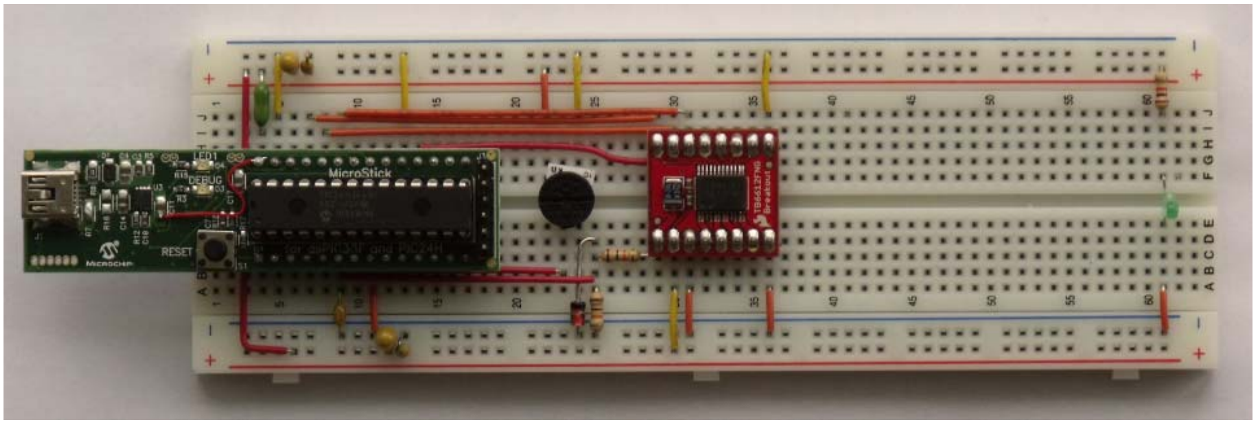
5. Insert the basic wiring and circuitry as shown:



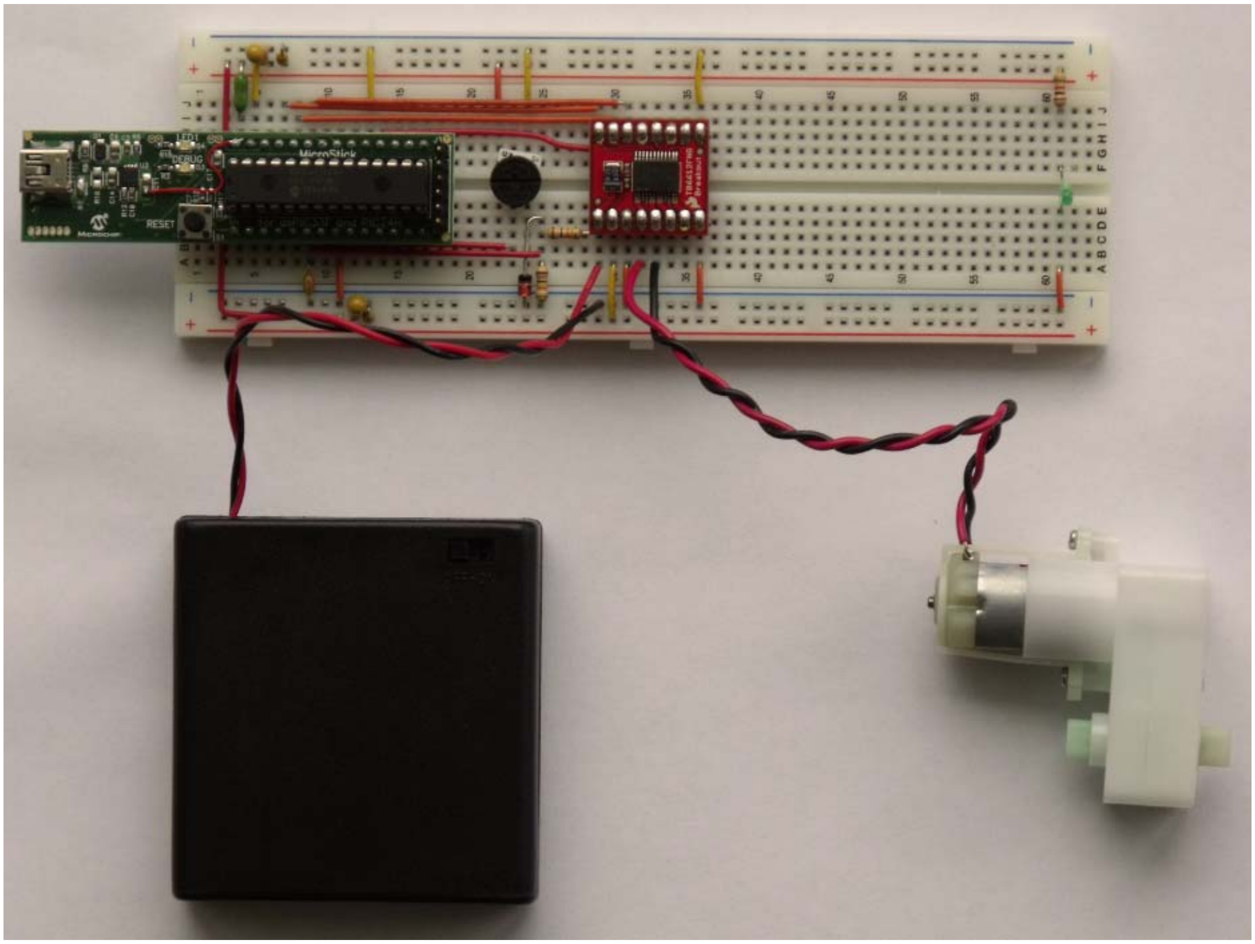
Design & Assembly Notes/Tips (Refer to schematic previous page):

C1 & **C3**: These caps are polarized (i.e. there is a "+" end – this is to be connected to the +3.3v rail).
LED D1: The anode-side of this component is the longer lead.
Zener Diode D2: This component (along with R5, R6, C4) is used to signal the firmware when the battery pack has been switched on, in order to enable the sending of control signals to the motor chip. The design of the circuit limits the voltage produced on MCU analog input pin to ~2v maximum, with up to 15VDC on VM.
500mA Fuse F1: Keep a set of spare fuses handy. If your power led (**D1**) fails to light after prototyping your circuit, this is a likely cause.

6. Next, insert the Microstick and the Motor driver modules as shown:



7. Finally, connect the motor and motor supply battery pack (twist the battery pack leads as shown and **make sure the power switch is OFF**):



Design & Assembly Notes/Tips:

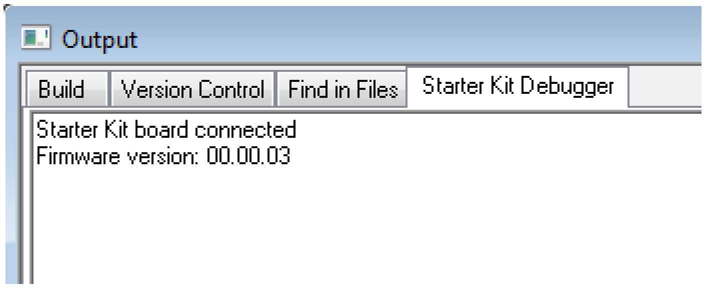
Why twist the leads? Wires are components! They have resistance, capacitance and inductance. The leads from the battery pack to the circuit and from the motor driver to the motor can carry up to 1A, and they require this current to change up/down rather rapidly. Twisting the wires reduces the inductance of these leads, making it easier for the current to change.

8. Test the completed assembly by importing/programming a .hex file into the MCU:

a. Open MPLAB, and set the processor to PIC24HJ64GP502 (Configure → Select Device → PIC24HJ64GP502)

b. Select the Microstick Debugger as Debugger (Debugger → Select Tool → Starter Kits)

c. Successful connection to target indicated in output window:



2. Import the .hex file (File→Import→ C:\IEEE\IEEECETA_WS\Day 4\Lab 3\Solution\Lab 3.hex)

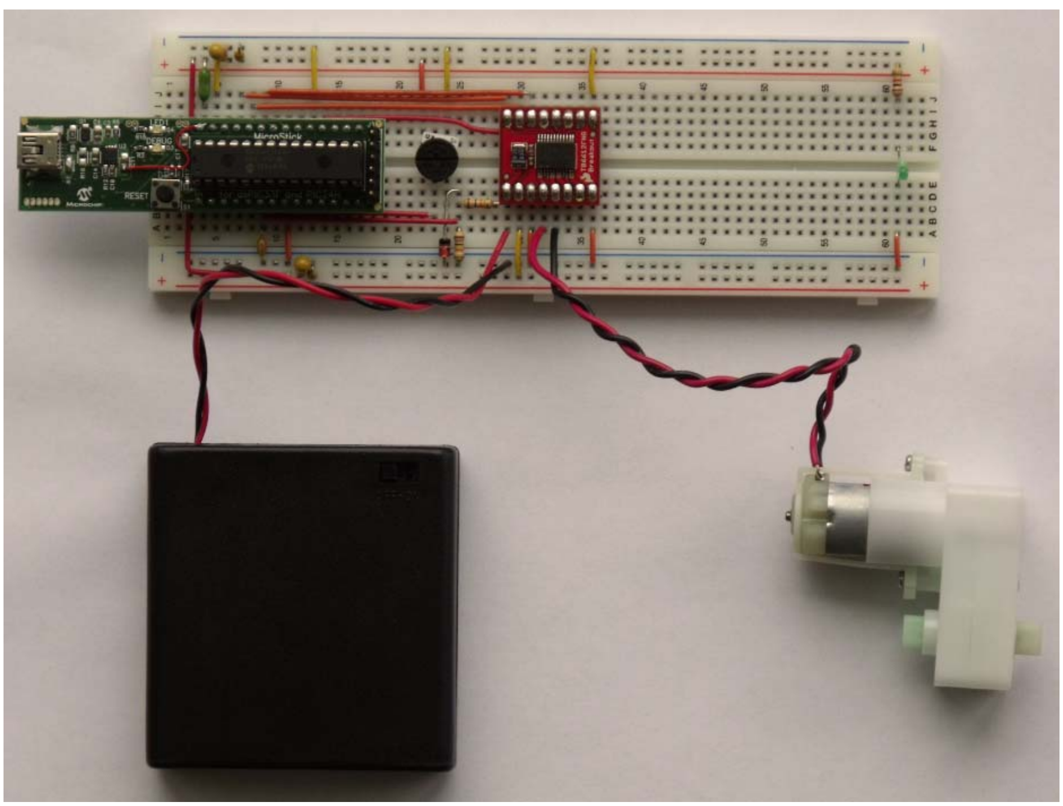
3. Program the device (Debugger → Program)

4. Run the program (Debugger → Run)

5. Turn on the battery pack's power switch.

6. The motor should start to spin!

7. Using a screwdriver, rotate the potentiometer to control the speed of the motor. To stop the motor, turn off the battery pack power switch. To halt the MCU press Debugger → Halt



Lab 2 – Creating PIC24 C Projects in MPLAB

Purpose:

To learn how to create a C-based project from scratch for PIC24 using the MPLAB C Compiler for PIC24. This project will blink the Microstick on-board led, LED1.

A solution workspace is provided: C:\IEEE\IEEECETA_WS\Day 4\Lab 2\Solution\Lab2.mcw

Procedure:

1. MPLAB IDE and C30 must be installed in their default directories. The hardware for this lab should already be constructed (see Lab 1). The student CDROM must be copied into C:\IEEE\IEEECETA_WS

2. Using Windows Explorer, copy header file "p24HJ64GP502.h" from C:\Program Files\Microchip\MPLAB C30\support\PIC24H\h to C:\IEEE\IEEECETA_WS\Day 4\Lab 2

3. Using Windows Explorer, copy linker script file “p24HJ64GP502.gld” from C:\Program Files\Microchip\MPLAB C30\support\PIC24H\gld to C:\IEEE\IEEECETA_WS\Day 4\Lab 2

4. Start MPLAB IDE, and create a new c-source code file for the main() function:

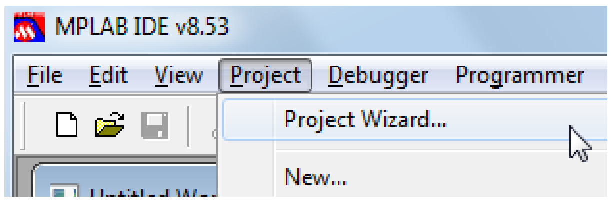
    a. File→New

    b. Type the following source code into the new file (leave out the comments to save time):

```
/*** #include files ******************************************************/

#include "p24hj64gp502.h"

/*** Symbolic Constants used by main() ***********************************/

#define DELAY_VAL 30000

/*** Local Function Prototypes *******************************************/

void delay(void);       // simple delay

/*** Configuration Bit Macros ********************************************/

_FOSCSEL(FNOSC_FRC & IESO_OFF)
_FOSC(POSCMD_NONE)
_FWDT(FWDTEN_OFF)
_FICD(JTAGEN_OFF & ICS_PGD1)

/*** Global Variable Declarations ****************************************/

int count;

/*** main() **************************************************************/

int main(void)
{
      TRISBbits.TRISB15 = 0;  // make the port pin an output

      while(1)
      {
            count++;
            LATBbits.LATB15 = 1;
            delay();
            LATBbits.LATB15 = 0;
            delay();
      }
}

/*** delay(void) *********************************************************/

void delay(void)
{
      // initialize the delay counter
      int delay_var = DELAY_VAL;

      // implement the delay loop
      do
      {
            delay_var = delay_var - 1;
      }
      while(delay_var < DELAY_VAL);

}
```

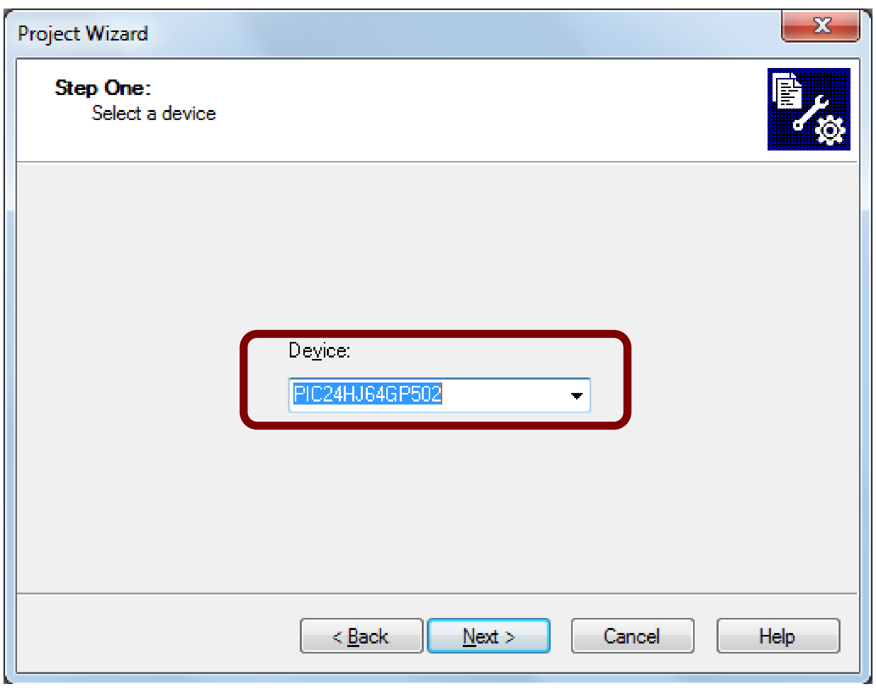
c. Save the new file: "File→Save As" main.c in C:\IEEE\IEEECETA_WS\Day 4\Lab 2

5. Start the "Project Wizard" which will guide you through the remaining steps to create an MPLAB IDE project and workspace (select Project→Project Wizard)



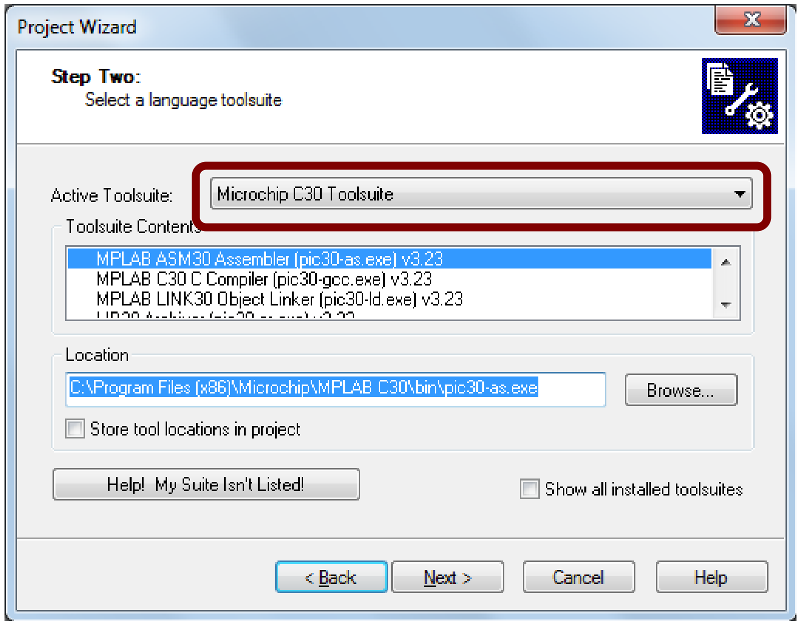
After the project wizard opens, click "Next" to continue

6. In the "Device" selection box, confirm that **PIC24HJ64GP502** is selected as device



After you do this, click "Next" to continue

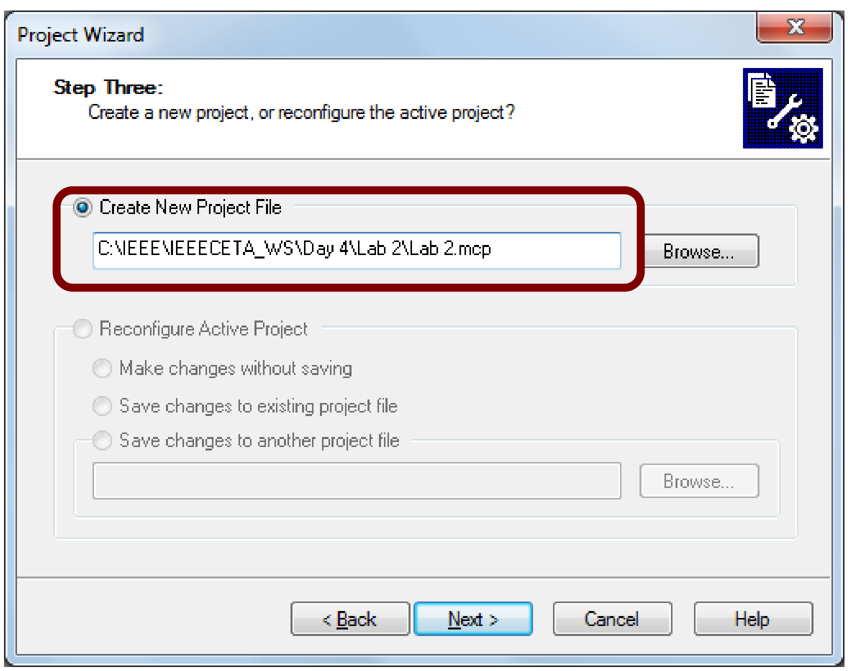
7. In the “Active Toolsuite” selection box, select **Microchip MPLAB C30** as the active tool suite.



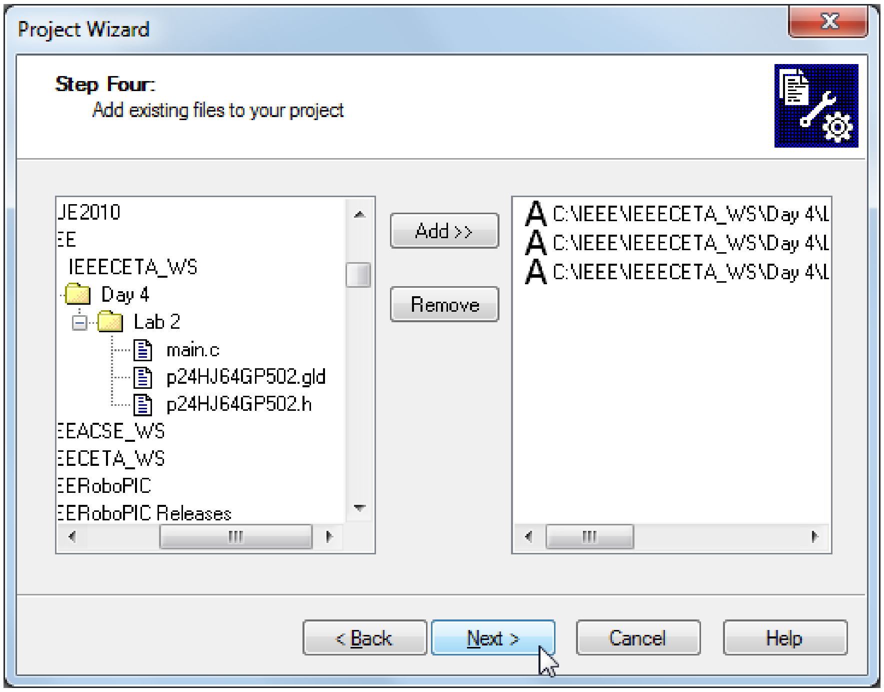
If there are red X’s beside any of the toolsuite files (compiler, linker etc), you will need to select the tool and “Browse” to the location where you instructed the MPLAB Compiler to install the files.

Click “Next” to continue

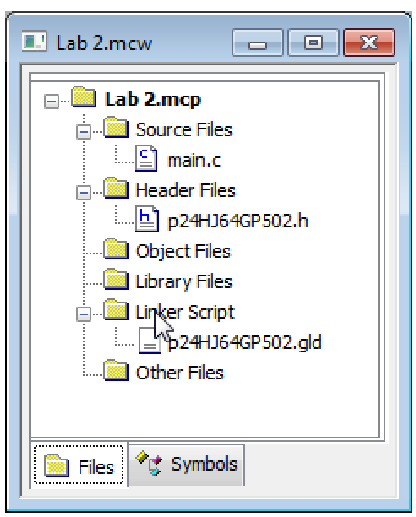
8. Enter the following project name in the box: **C:\IEEE\IEEECETA_WS\Day 4\Lab 2\Lab 2.mcp**



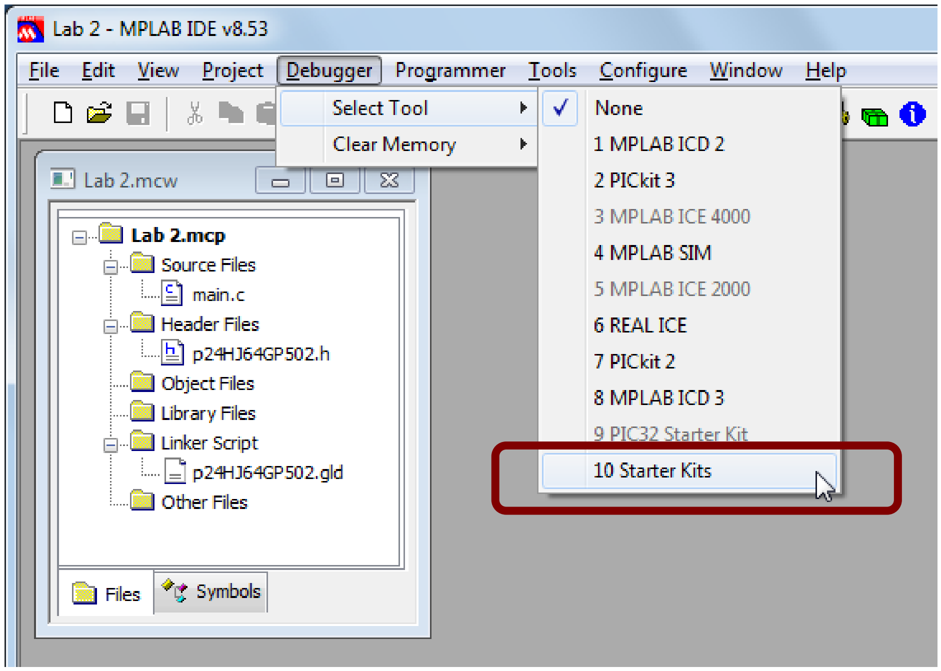
9. In the left pane, navigate to **C:\IEEE\IEEECETA_WS\Day 4\Lab 2**. Select each one of the 3 displayed files and press “Add” to add them to the project (main.c, p24HJ64GP502.h, p24HJ64GP502.gld):



10. Click “Finish”. If the project tree isn’t visible, select from the menu (View→Project). You should see the three files used in this project.

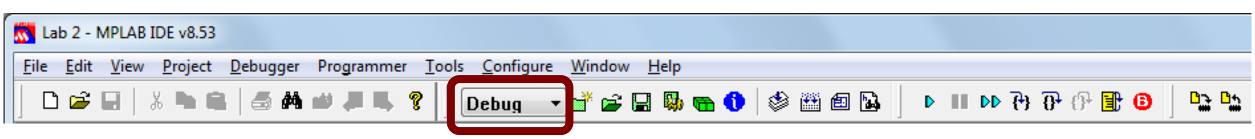


11. If not done already, connect the Microstick's USB cable to the PC. Enable the Microstick's on-board debugger circuit by selecting from the menu (Debugger→Select Tool→Starter Kits)

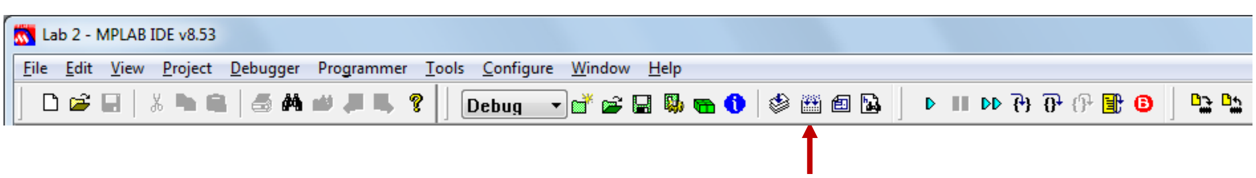


The output window (View→Output) should show a successful connection to the Microstick debugger circuit.

12. Select/Enable a "Debug" code build:

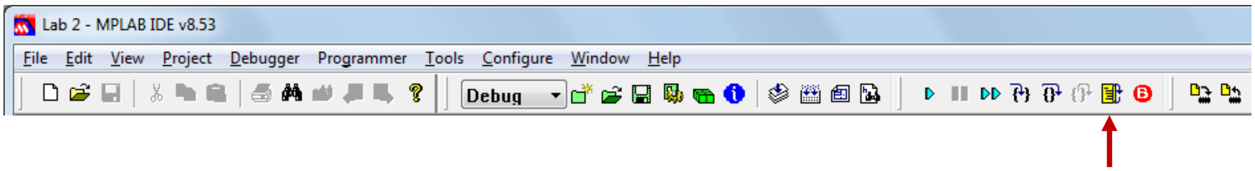


13. Click the **Build All** button 

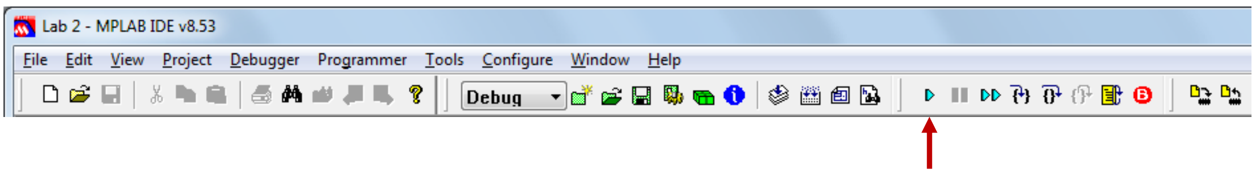


14. If no build errors are reported, program the PIC24 by clicking the **Program** button 

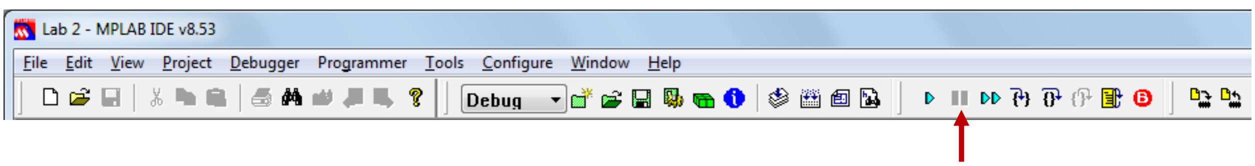
15. When programming completes, click the **Reset** button to reset the PIC24



16. To run the program, click the **Run** button



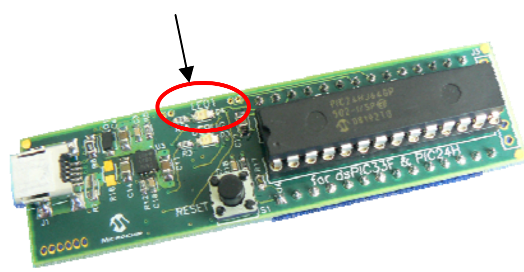
17. To stop the program, click the **Halt** button



To startup again, click **Reset**, followed by **Run**.

Results:

LED1 Blinks!



To learn more about debugging in the MPLAB IDE, please refer to the document "16-Bit Language Tools Getting Started (DS70094E)" guide available in C:\Program Files\Microchip\MPLAB C30\docs

Lab 3 – Motor Control using the PWM Peripheral

Purpose:

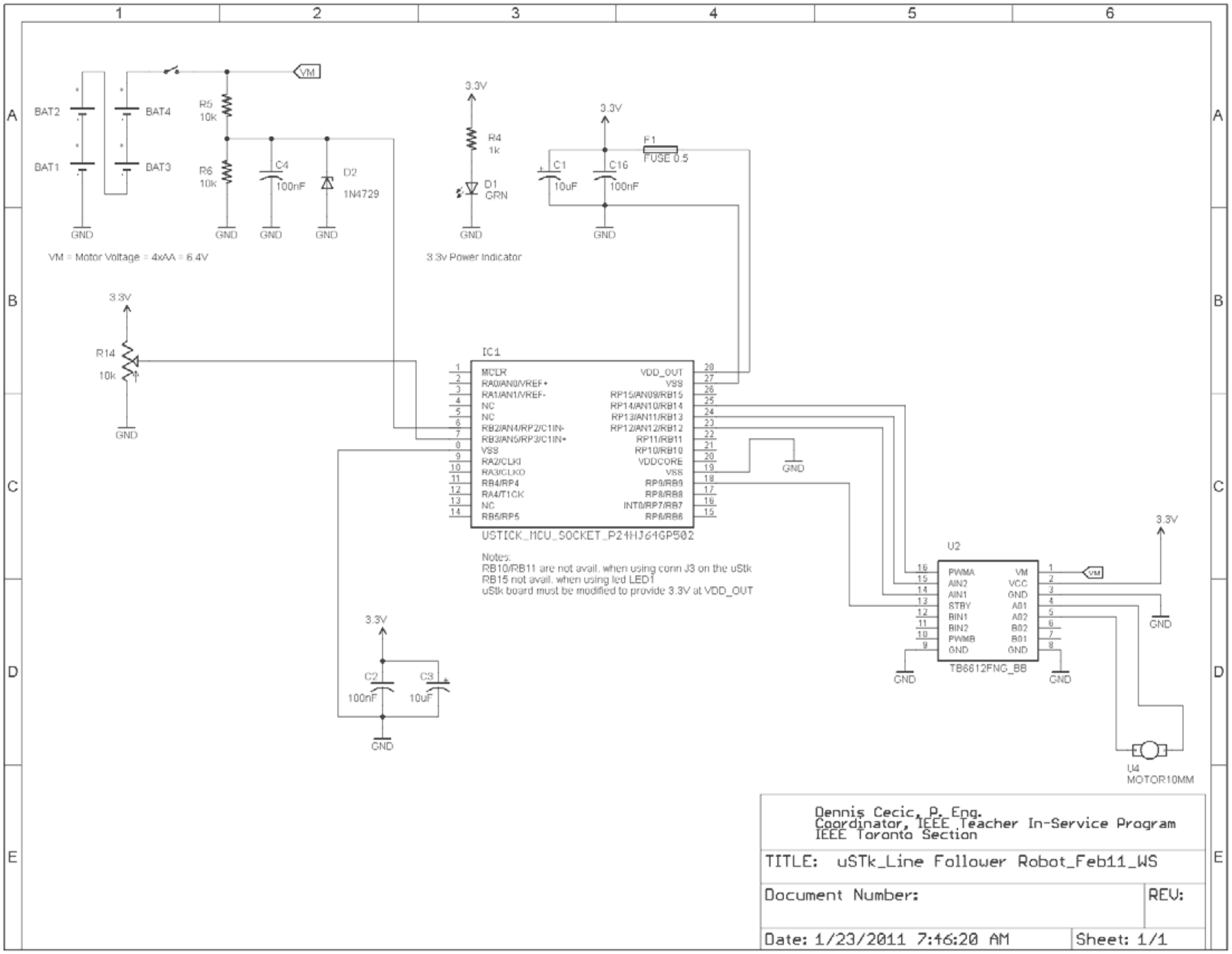
Experiment (modify, rebuild, program) the solution firmware code to demonstrate the 2 main operating modes of the motor, CW and CCW. The operating modes of the TB6612 are reproduced here:

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

Procedure:

1. Start MPLAB IDE
2. Open the solution workspace C:\IEEE\IEEECETA_WS\Day 4\Lab 3\Solution\Lab 3.mcw
3. Build/Program and Run the project. Observe the rotation direction of the motor.
4. Open main.c and find the initialization function Initialize(). Find the RB12 & RB13 port pin initialization code which controls the direction of the Motor (CW or CCW). Change the setting to reverse the direction of the motor.
5. Build/Program and Run the project. Observe the rotation direction of the motor. It should be reversed.

Appendix A – Line Follower Robot Electronic Control Circuit Schematic (Feb 11 ACSE_WS)



Appendix B – Electronic Control Circuit Component Kit (Feb 11 ACSE_WS)

Component Description	MFR Part Number	Digikey Part Number	Qty.	Notes
MICROSTICK DSPIC33F/PIC24H BOARD	DM330013	DM330013-ND	1	Academic pricing
BREADBOARD 2.13x6.496 SLDLESS	TW-E40-1020	438-1045-ND	1	
WIRE SET 140PC FOR BOARD	TW-E012-000	438-1049-ND	1	
CAP .1UF 50V 10% CER AXIAL	C412C104K5R5TA7200	399-4484-1-ND	4	
CAP TANTALUM 10UF 16V 10% RAD	TAP106K016SCS	478-1839-ND	2	
FUSE PICO FAST .500A 125V AXIAL	0251.500MXL	F2311-ND	2	
DIODE ZENER 3.6V 1W DO-41	1N4729A-T	1N4729ADICT-ND	1	
HOLDER BATT W/COVR 4AA ON/OFF SW	SBH-341AS	SBH-341AS-ND	1	
POT 10K OHM THUMBWHEEL CERM ST	3352E-1-103LF	3352E-103LF-ND	1	
BATTERY IND ALKALINE AA SIZE	EN91	N107-ND	4	Motor's VBUS
SPARKFUN ROB-09457 MOTOR DRVE 1A DUAL TB6612FNG	ROB-09457	SOLARBOTICS#50646	1	Vendor: Solarbotics (1-866-276-2687)
143:1 GEAR MOTOR + WHEEL	GM8 + GMPW-B	GM8PW Deal	1	Vendor: Solarbotics (1-866-276-2687)
CONN HEADER 36POS .100 VERT GOLD	22-28-4363	WM6536-ND	1	
RES 10K OHM 1/4W 5% CARBON FILM	CFR-25JB-10K	10KQBK-ND	5	
RES 1.0K OHM 1/4W 5% CARBON FILM	CFR-25JB-1K0	1.0KQBK-ND	5	
LED 3.1X2MM 563NM GREEN DIFFUSED	SLR-322MG3F	511-1226-ND	1	
#22AWG Black Stranded (6")	3051 BK005	A2016B-100-ND	1	100' Price = $41.23
#22AWG Red Stranded (6")	3051 RD005	A2016R-100-ND	1	100' Price = $41.23
#30AWG Wire Wrap (1")	R-30R-0050	K232-ND	1	50' Price = 9.97