EE4207 Real Time Control Experiment Using Microcontroller

1. Objective

In this lab session, you will experiment the use of a microcontroller board to test some features of a microcontroller for real time application. Specifically, you will use the PWM, ADC and timer features. After going through this hands-on session, you should gain a good understanding the process of compiling and linking a real time C program, as well as programming the code onto a microcontroller.

2. Equipment & Software

2.1 Equipment

- Microchip dsPIC30F MC1 Motor Control Development Board (MCDB) with a microcontroller dsPIC30F6010A (Fig. 1)
- Microchip MPLAB In-Circuit Debugger (ICD2) (Fig. 2)

2.2 Software

- MPLAB V8.60
- MPLAB C30 Compiler

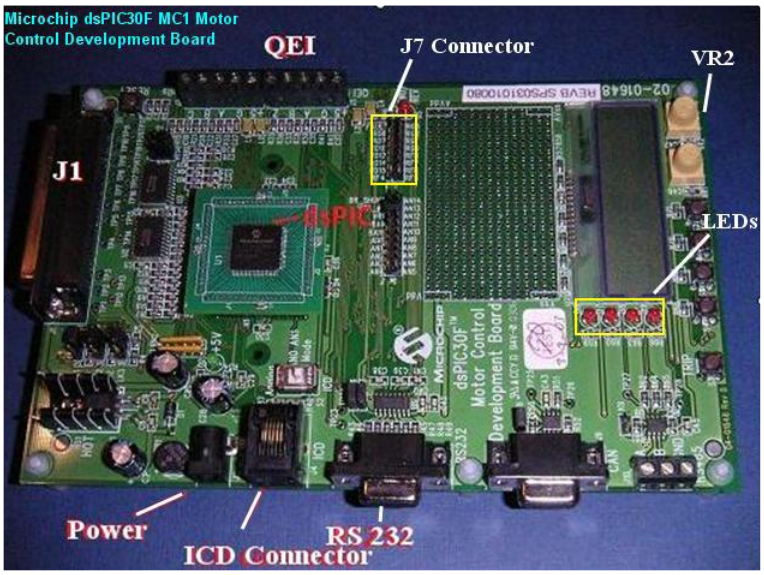


Fig. 1 Motor Control Development Board (MCDB)

3. Experimental Setup

3.1 Connect ICD2 (Fig. 2) to the PC

Use the USB cable to connect the ICD2 to the USB port of PC. ("Power" LED light on ICD2 will light up).

3.2 Power-up sequence

a. Power-up the ICD2.
b. Start the MPLAB program to communicate with the ICD 2.
c. Power-up the MC1.

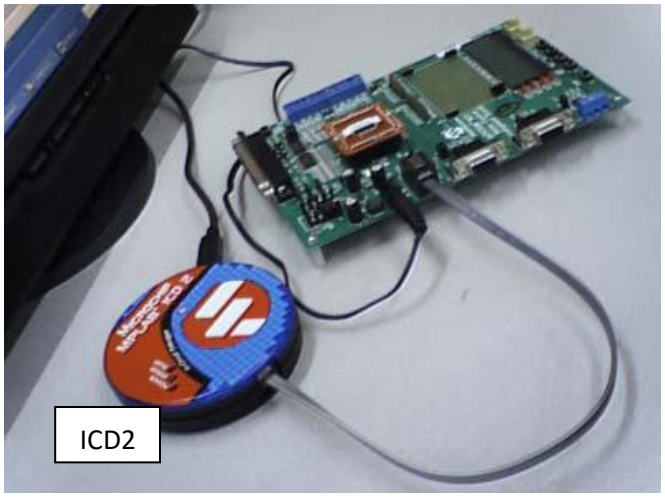


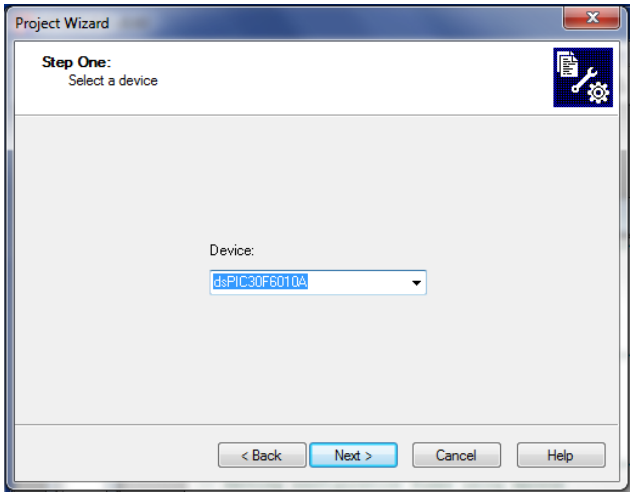
Fig. 2 Experimental Setup

4 . Using Sample C Program File

a. Copy folder at **E:\EE4207 Design Class-Microchip** to **OS(C): Users\Username\My Documents\EEE4207 Design Class-Microchip.**
b. From the PC desktop, launch the **MPLAB IDE V8.60** program.
c. Inside the program, select ***File > Open***, this will invoke the MPLAB editor.
d. From the Microsoft Window explorer, choose **My Documents** > **EE4207 Design Class-Microchip** > **student-example** > **DemoLab1.c** and click **Open**.

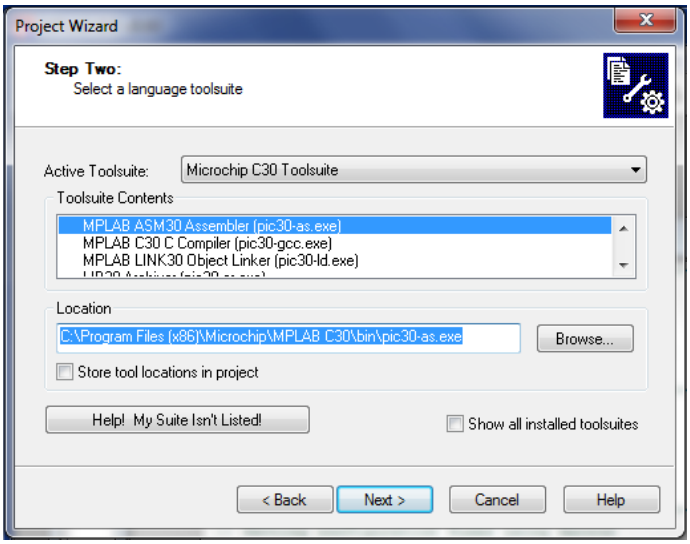
5 . Creating Project

a. Select ***Project > Project Wizard.***
b. Click **Next >** to continue.
c. In "**Step One: Select a device**", use the pull-down menu to select **dsPIC30F6010A** and click **Next>** to continue.



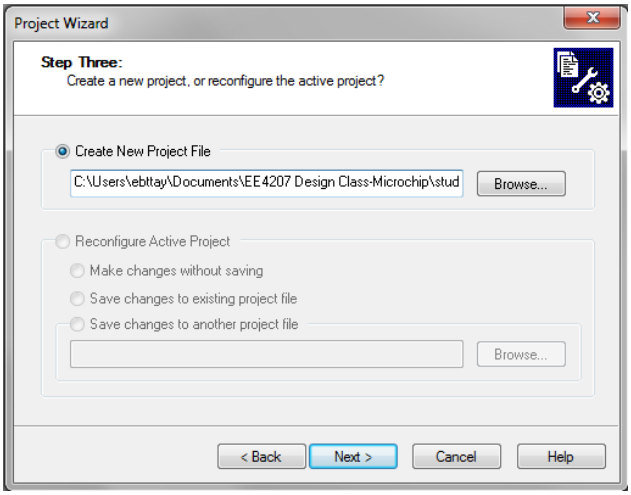
d. In "**Step Two: Select a language toolsuite**", choose "**Microchip C30 Toolsuite**" as the "Active Toolsuite", and click on each language tool under "Toolsuite Contents" and set up its associated executable location as follows:

MPLAB ASM30 Assembler
C:\Program Files(x86)\Microchip\MPLAB C30\bin\pic30-as.exe
MPLAB C30 C Compiler
C:\Program Files(x86)\\Microchip\MPLAB C30\bin\pic30-gcc.exe
MPLAB LINK30 Object Linker
C:\Program Files(x86)\\Microchip\MPLAB C30\bin\pic30-ld.exe
LIB30 Archiver
C:\Program Files(x86)\\Microchip\MPLAB C30\bin\pic30-ar.exe



Click Next> to continue.

e. In **"Step Three: Create a new project, or reconfigure the active project?"**, click on the **Browse** to select the folder **student-example\DemoLab1.c** where the project (e.g. DemoLab1) will be saved.



- Click **Next>** to continue.

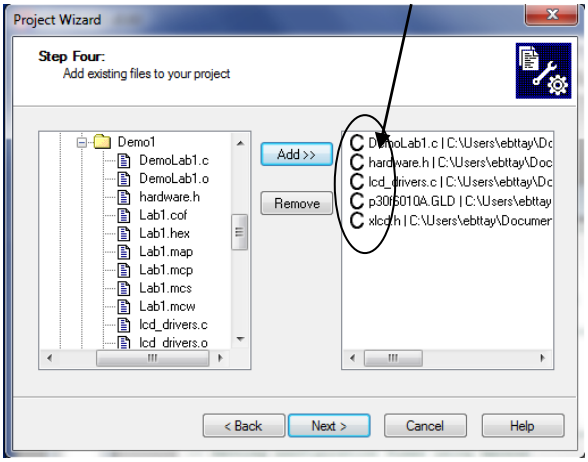
f. In "**Step Four: Add any existing files to your project**", select list of files to be added for this project.

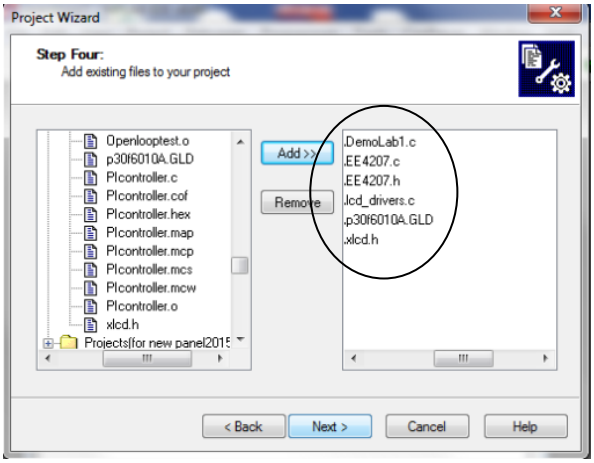
i. Select the source file **DemoLab1.c** and press **Add >>**

ii. Continue to select and add the following files to your DemoLab.

Demo Lab 1	Demo Lab 2	Demo Lab 3	Demo Lab 4
• DemoLab1.c	• DemoLab2.c	• DemoLab3.c	• DemoLab4.c
• EE4207.c	• EE4207.c	• EE4207.c	• EE4207.c
• EE4207.h	• EE4207.h	• EE4207.h	• EE4207.h
• lcd_drivers.c	• lcd_drivers.c	• lcd_drivers.c	• lcd_drivers.c
• P30f6010A.GLD	• P30f6010A.GLD	• P30f6010A.GLD	• P30f6010A.GLD
• xlcd.h	• xlcd.h	• xlcd.h	• xlcd.h

iii. Click on the "A" until it changes from ("A", "U", "S") to "C"





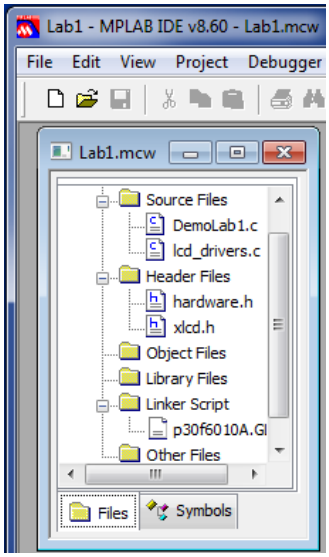
iv. Click **Next>** to continue.

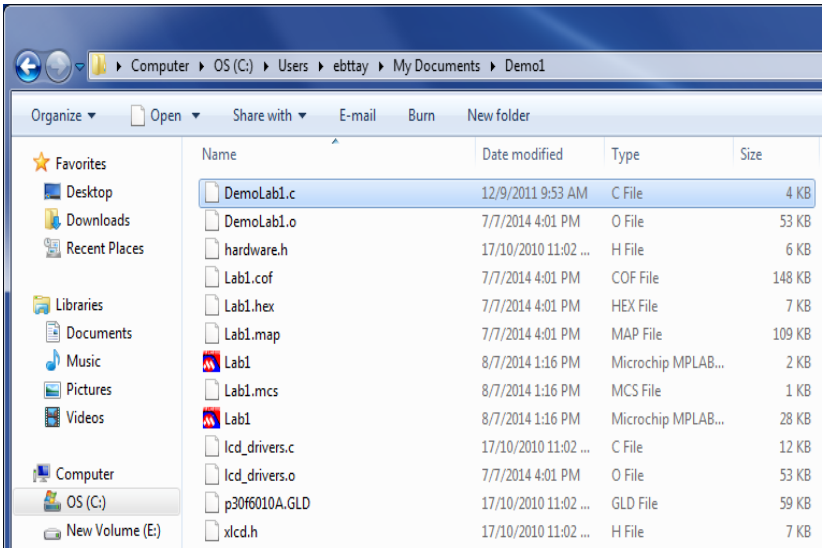
v. Click **Finish** to create the new project and workspace.

vi. If you open the project directory, you may notice there are many different type of files. In general, there are organized into 3 main groups:

- **Workspace:** The .mcw file that holds the project plus the various development files.
- **Project:** The .mcp file that holds the documents.
- **Program Files:** The source files (.c), header files (.h), linker files (.lkr), etc.

Now, open the **Source File >** DemoLab1.c



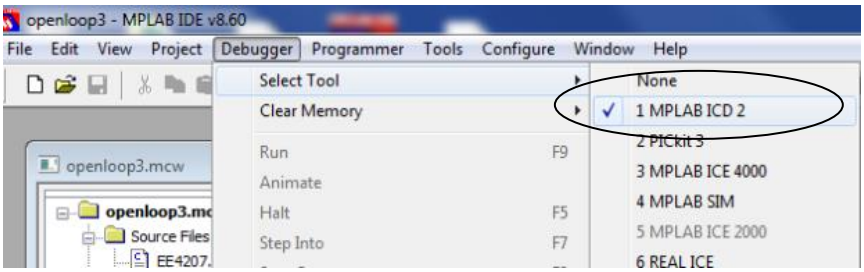


6. Build Project – compile, assemble and link the programs

Make sure the **MICRO-CONTROLLER BOARD** is powered ON **(GREEN LED +5V LIGHTED UP)** and connected to the **ICD2**.

a. To compile and create the downloadable file, perform the followings:

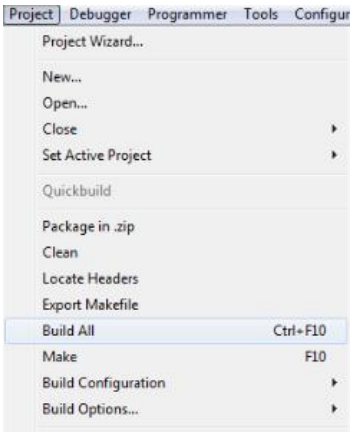
- Go to ***Debugger* > Select *Tool*,** then select **MPLAB ICD 2**.



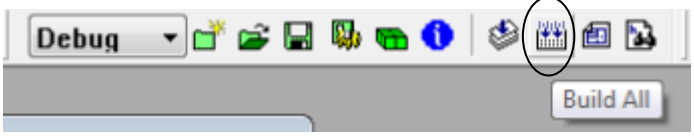
Set **Com Port** to **USB** and **Baud Rate** to **19200.**

- Click **Next >** to continue.
  Uncheck **MPLAB IDE automatically connects to the MPLAB ICD2**.
- Click **Next >** to continue.
  Check **MPLAB ICD2 automatically downloads the required operating system**.
- Click **Next >** to continue.
- Click **Finish** to accept MPLAB ICD2 settings.

b. To compile, assemble and link the project. Go to ***Project > Build All***

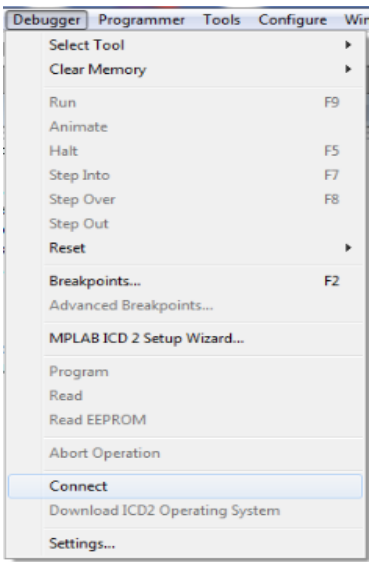


Or you may click the icon as shown below:



Output window should display either "BUILD SUCCEEDED" or error and warning message.
If "**BUILD SUCCEEDED**", proceed to program your board.

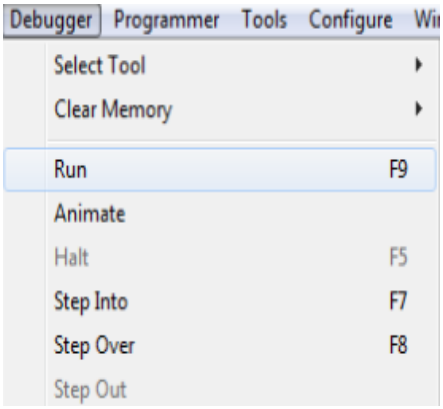
- Go to ***Debugger > Connect***



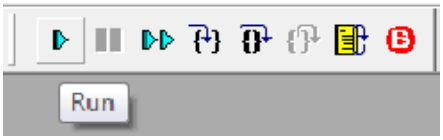
- Go to ***Debugger > Program***

c. After you see the message "MPLAB ICD 2 Ready", proceed to run the program on the board as follows:

- Go to **Debugger > Run**



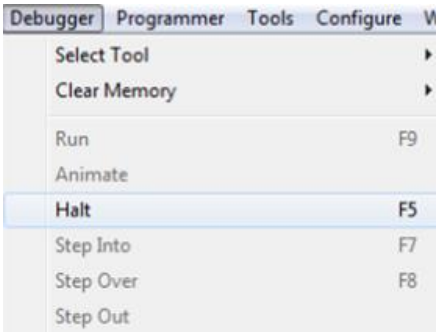
Or click the icon:



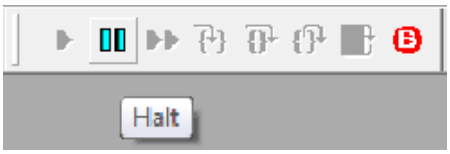
- Vary the trimmer VR2 on the microcontroller board to control the lighting sequence of the 4 bits red LEDs on the board.

d. After you have experimented, halt the program as follow:

- Go to ***Debugger > Halt***



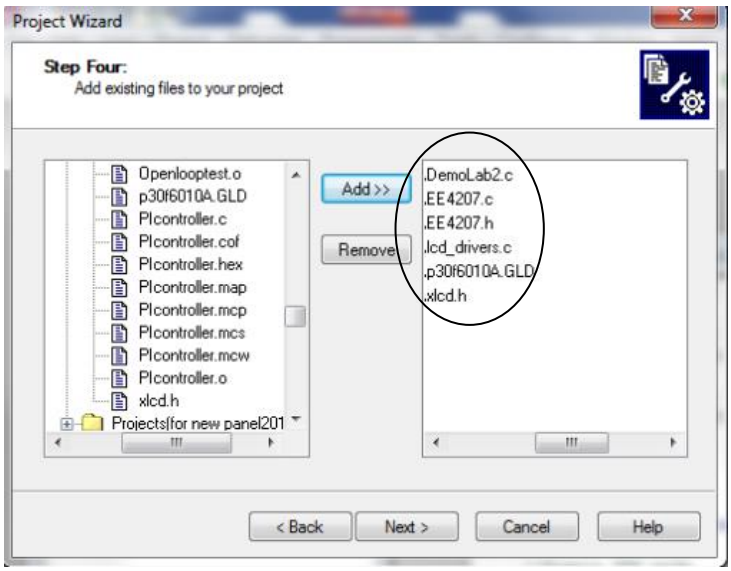
Or click the icon:



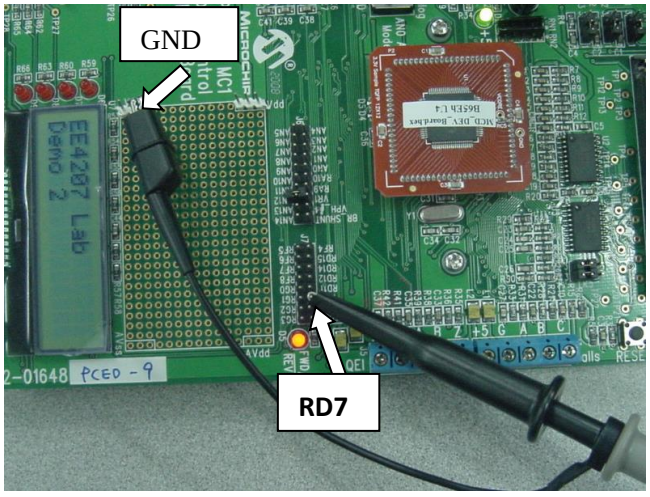
e. Next, quit the project as follow:
   - Go to ***Project > Close***

7. Testing PWM Demo Program

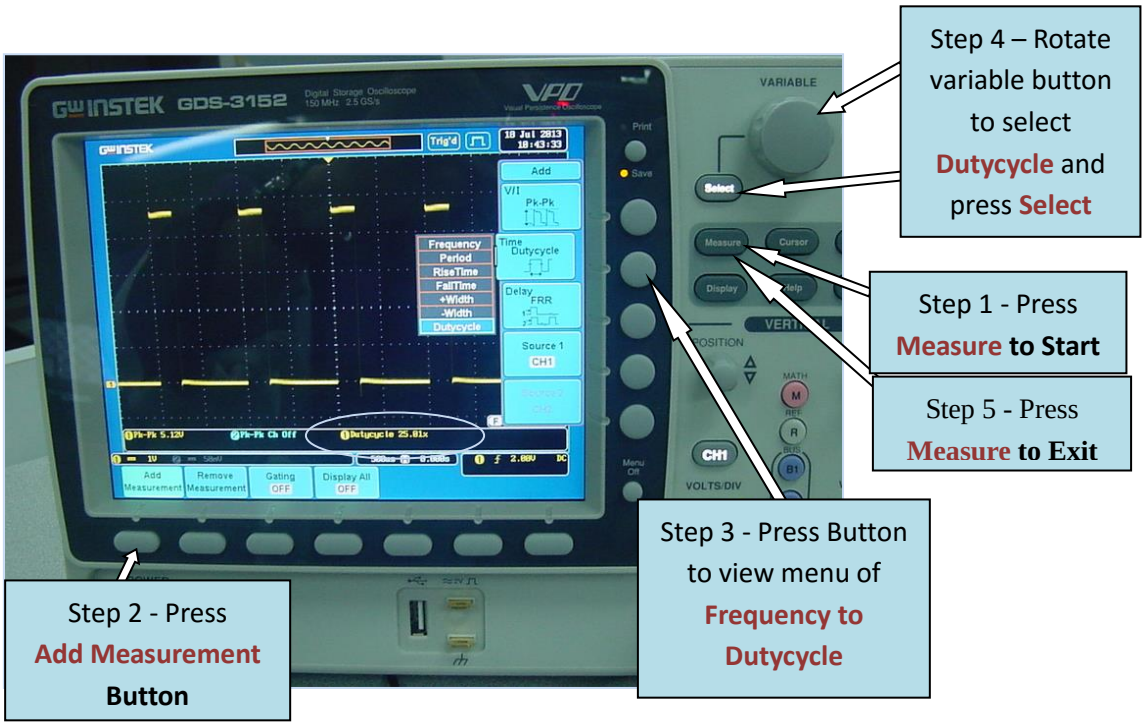
a. Next, you may test the PWM demo program (i.e. **DemoLab2.c**) by repeating the procedures from **4c** to **6** (For DemoLab2). Make sure that you add the files listed as follows:



b. After you have managed to run the program on the microcontroller board, observe the PWM output using an oscilloscope. Connect the oscilloscope probe to the **RD7 pin** of connector J7. Also, connect the **GROUND pin** of probe to **Vss** that is located near the LCD.



- Adjust the settings of the oscilloscope to view the **Duty Cycle (method 1)**.



c. After you have experimented the PWM waveform, halt the program and click to open the **DemoLab2.c** program from the project window.

d. Study this C program and change the PWM output to a waveform of 1kHz frequency with a **duty cycle = 25%**.

e. After you have finished experimenting, halt the program and exit the project.

- Adjust the settings of the oscilloscope to view the **Duty Cycle (method 2)**.

i. Press the button "Cursor" so that the screen will display "H cursor" in the lower left corner.

ii. Press the button under "H Cursor" several times so as to switch the moving line between the solid line and dotted line. The solid line is the moving line.

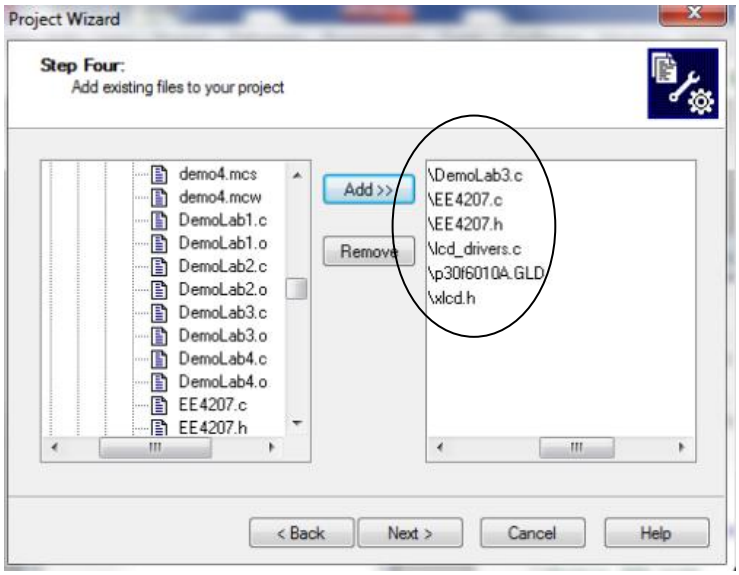
iii. Measurement the period of high level signal (green arrows) and the cycle of signal (red

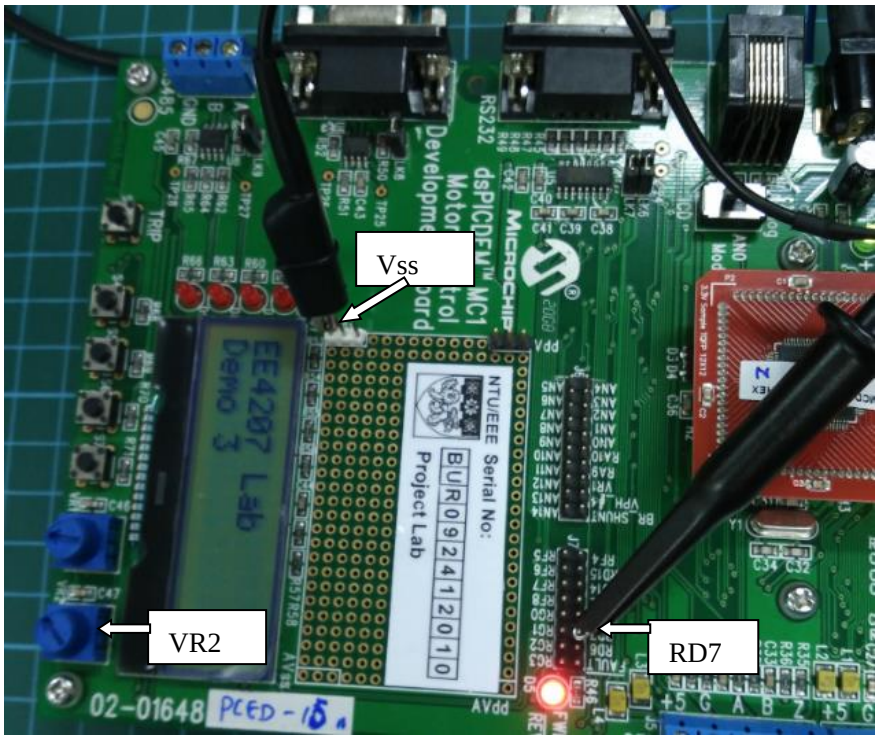
arrows).

iv. The duty cycle is (green arrows/ red arrows)x100%.

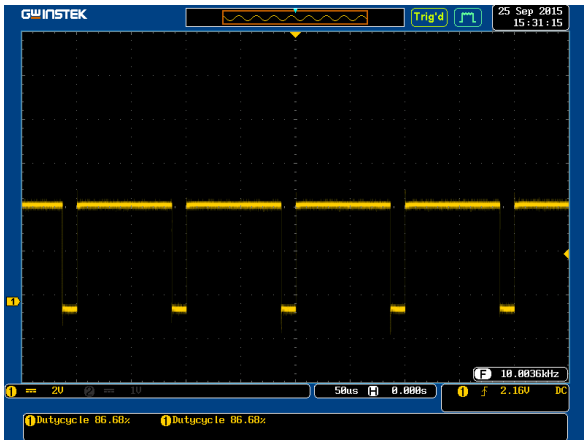
8. Testing ADC Demo Program

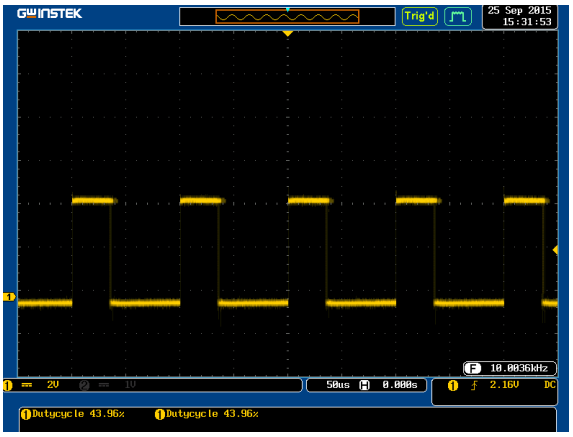
a. Next, you may test the ADC demo program (i.e. **DemoLab3.c**) by repeating the procedures from **4c** to **6** (For DemoLab3). Add the files shown below and keep the same connection of the scope to the microcontroller board.





b. After you have managed to run the program on the microcontroller board, observe the PWM output by varying the trimmer **VR2**. You may see that the duty cycle is changed.

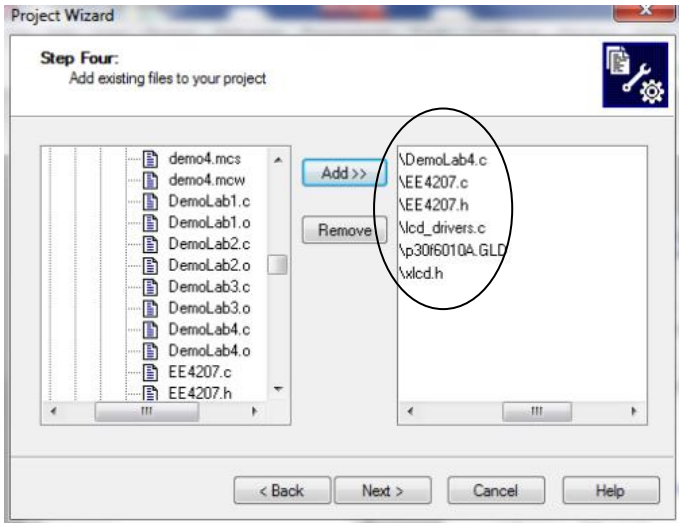


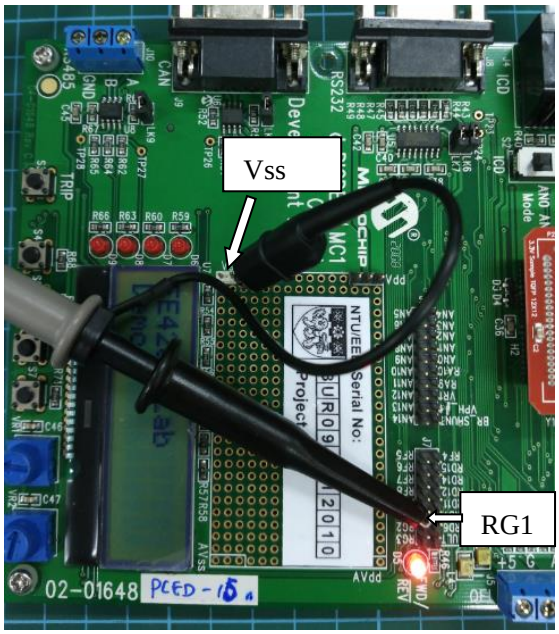


c. After you have finished experimenting, halt the program and exit the project.

9. Running time of while loop

a. Next, you may test the running time of while loop (i.e. **DemoLab4.c**) by repeating the procedures from **4c** to **6** (For DemoLab4). Add the files shown below and keep the same connection of the scope to the microcontroller board.





In this demo you need to test the running time with the following delay value:

Delay value	Running time
10,000	
30,000	
50,000	
70,000	
90,000	
135,000	

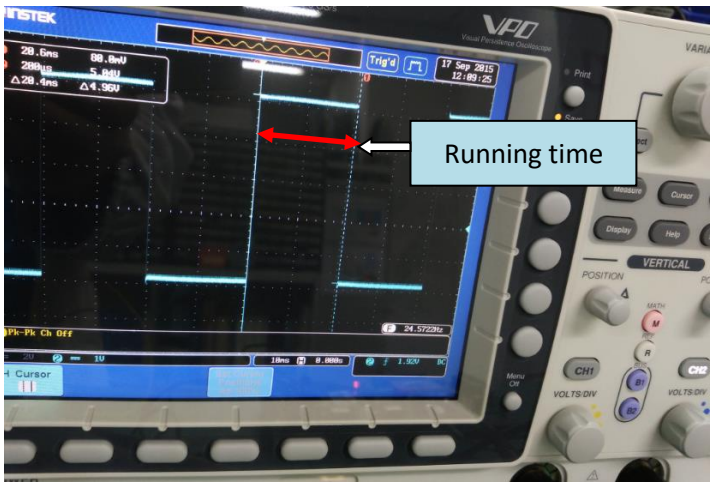
You may change the delay value in file “DemoLab4.c”. For example, put in the delay value “30000” as shown in next figure, then click “Build all”, “Program” and “Run” one by one and then measure the period of high level signal. The measurement is the running time. Re-enter the new delay value given in the above table and record the new running time.

C:\...\DemoLab4.c*

```
int main( void )
{
    Setup();

    //LCD display
    OpenXLCD(FOUR_BIT&LINES_5X7);
    Delay(50000);
    while(BusyXLCD());
    SetDDRamAddr(0);
    putsXLCD("EE4207 Lab");
    while(BusyXLCD());
    SetDDRamAddr(0x40);
    putsXLCD("Demo 4");

    while(1)
    {
        LATG= LATG & 0xFFFD;  // Set RG1 = 0
        Delay(30000);
        LATG= LATG | 0x0002; // Set RG1 = 1
        Delay(30000);
    }           // end while(1)
}               // end main
```

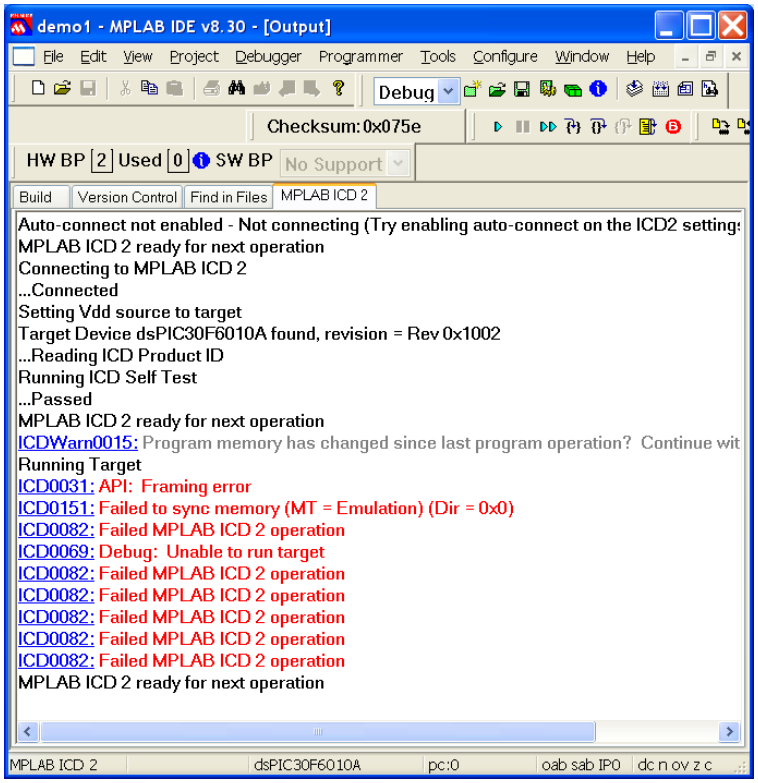


10. Errors Encountered and Solutions

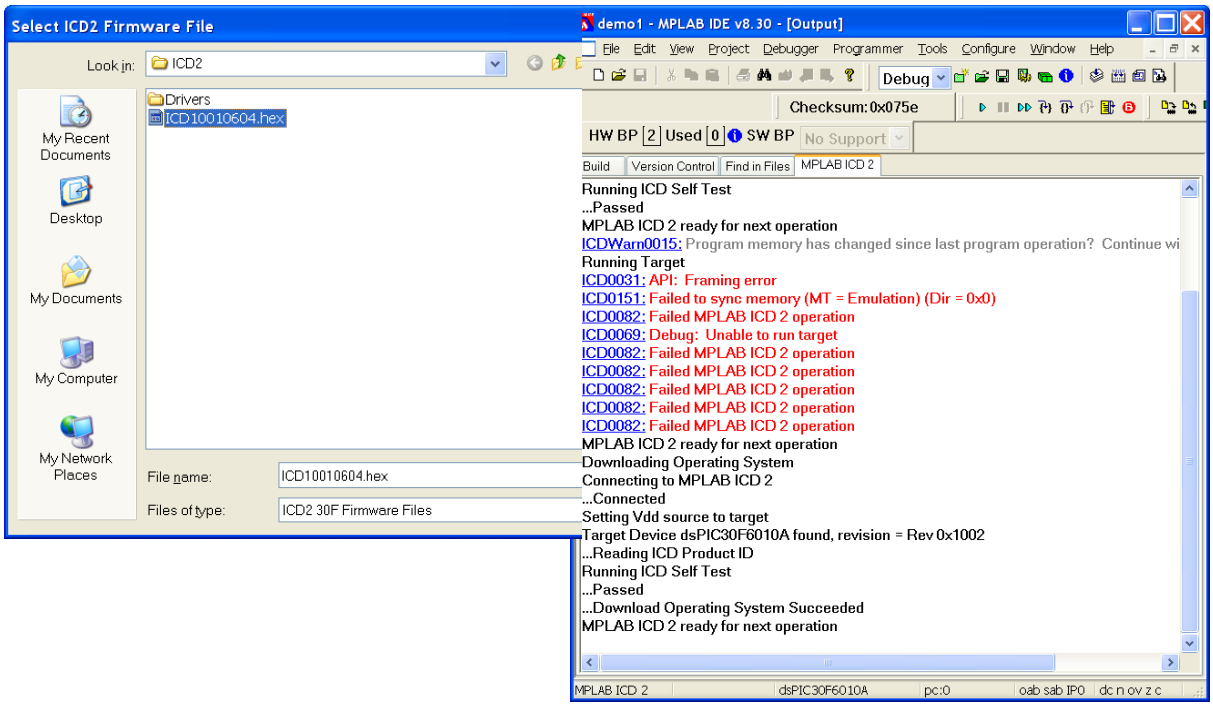
10.1 Framing Error

a. ***Debugger > Run***

ICD0031: API: Framing error



b. Solution : ***Debugger > Download ICD2 OS > ICD10010604.hex***



10.2. Heap Required (when building DemoLab3.c)

c:\program files\microchip\mplab c30\bin\pic30-coff-ld.exe Error: A heap is required, but has not been specified.
Link step failed.

Solution : Set heap to 1024

Project > Build Options > project > MPLAB Link30