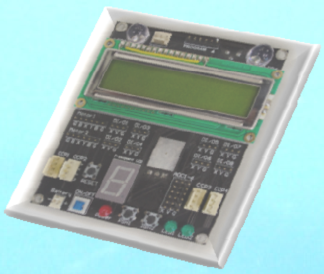
Advanced Robotics & Intelligent Control Centre
SINGAPORE POLYTECHNIC



CsBoard-03 Controller User Guide

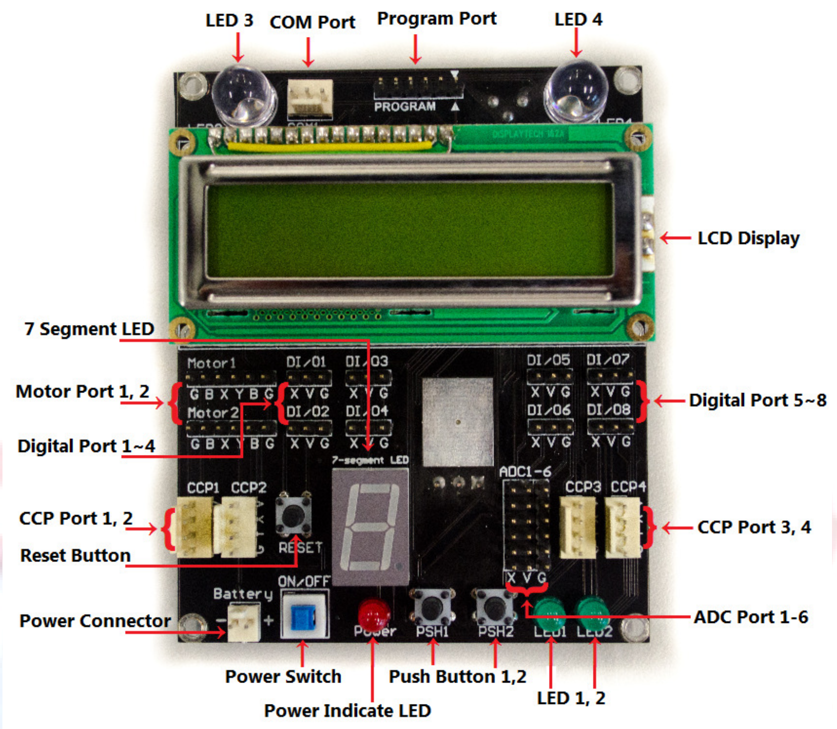
Advanced Robotics & Intelligent Control Centre
SINGAPORE POLYTECHNIC

http://www.robocupsingapore.org/cospace/

Email: CoSpace@robocupsingapore.org

1: Hardware Overview

The CsBoard-03 is an educational controller board used for motor control and sensor interface.



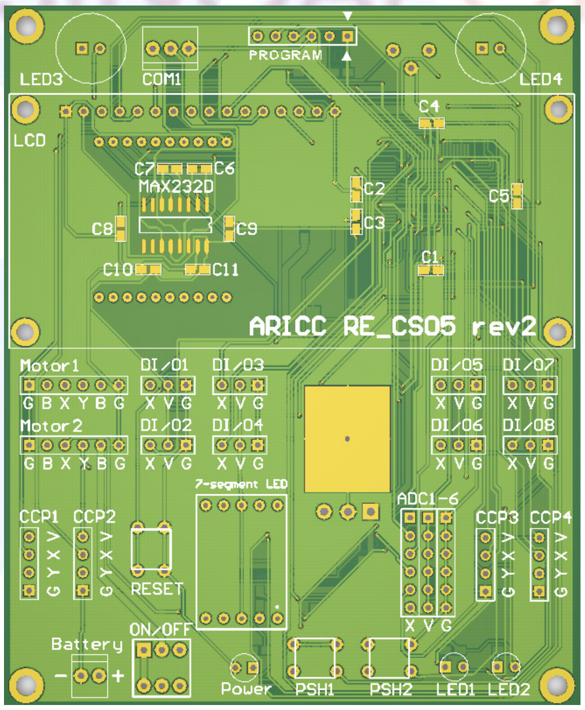


Fig. 1: CsBoard-03 controller board

1.1: Circuit Board Features

- **CPU Specification**
  - dsPIC30F6014A
  - Manufacturer: Microchip Technology Inc.
  - 16-bit Digital Signal Controller
  - Program Memory: 144 Kbytes
  - RAM: 8192
  - EEPROM Data Memory: 4096 Kbytes
- **Power Supply**
  - This board can be powered using 5-12V voltage source. The power supplied to the CsBoard-03 will be directly supplied to motors as well, hence requiring only a single power supply. The user can decide on the power supply voltage level according to the motor specification.
- **PC Interface**
  - Connected to a PC via two UART ports using serial cable or ZigBee module
- **Boot Loader**
  - The user can program code and data using PICkit 2 Programmer
- **PAN**
  - This board provides wireless communication with devices that support Zigbee technology via a PAN (ZigBee) (Optional)
- **LED**
  - 4 on-board LEDs are available for the board condition and data status
- **Push Button**
  - One reset button is available for hardware/firmware reset.
  - Two programmable push buttons are available.

- **ADC Port**
  - There are 6 12-bit Analog-to-Digital Converter Ports.

    These ports allow conversion of an analog input signal to a 12-bit digital number. These ports can be used for various infrared distance sensors, colour sensors etc.
- **CCP Port**
  - There are 4 CCP ports for input pulse capture. The capture interrupt mode can be set for the rising or falling edge of the pulse.
- **Motor Port**
  - These motor ports can be configured to control 3-pin or 4-pin motors.
- **Digital Port**
  - These programmable digital ports can be configured as input or output ports. They can be used to read digital sensor data or output digital signal to trigger external devices.
- **Common pin labels**
  - V: Vcc, 5V
  - G: Ground, 0V
  - B: Battery
  - X,Y: signal pins

2: Peripheral Summary

- **Motor 1, 2**
  - These ports can be configured to work with 3-pin or 4-pin DC motors. Each motor connector can also be configured as 2 independent pulse-width-controlled servos.
- **CCP 1, 2, 3, 4**
  - These ports can be connected to CCP sensors. Most often used for interfacing with ultrasonic sensors.
- **DI/O1**
  - This port is reserved for compass
- **ADC1-6**
  - These are standard ADC port for sensors such as infrared sensor, distance sensor, photodiode, etc.
- **DI/O2-DI/O8**
  - These are programmable digital pins. These can be used for read/write digital signal.
- **RESET**
  - Push button to reset motherboard program to the initial state
- **PSH1, PSH2**
  - Programmable push buttons
- **LED 1, 2, 3, 4**
  - Programmable LEDs
- **COM1**
  - Standard RS232 serial port that could connects directly to a PC or other compatible devices.

3: Software Installation Guide

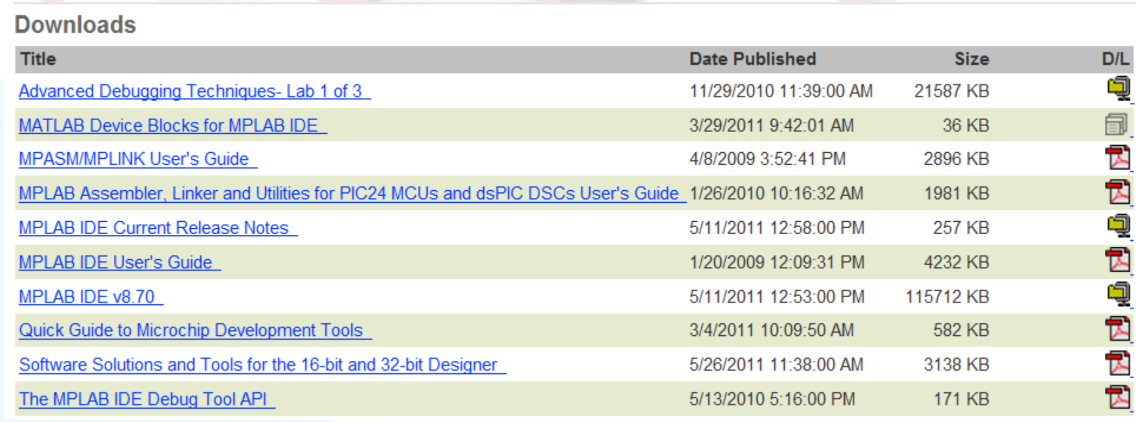
3.1. Install MPLAB IDE

What is MPLAB IDE

MPLAB Integrated Development Environment (IDE) is a free, integrated toolset for the development of embedded applications employing Microchip's PIC® and dsPIC® microcontrollers. MPLAB IDE runs as a 32-bit application on MS Windows®, is easy to use and includes a host of free software components for fast application development and super-charged debugging. MPLAB IDE also serves as a single, unified graphical user interface for additional Microchip and third party software and hardware development tools. Moving between tools is a snap, and upgrading from the free software simulator to hardware debug and programming tools is done in a flash because MPLAB IDE has the same user interface for all tools.

Download Link:

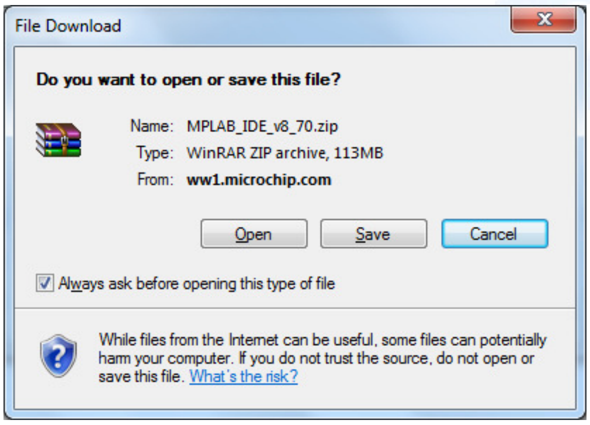
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002



Downloads

Title	Date Published	Size	D/L
Advanced Debugging Techniques- Lab 1 of 3	11/29/2010 11:39:00 AM	21587 KB	📦
MATLAB Device Blocks for MPLAB IDE	3/29/2011 9:42:01 AM	36 KB	▤
MPASM/MPLINK User's Guide	4/8/2009 3:52:41 PM	2896 KB	📄
MPLAB Assembler, Linker and Utilities for PIC24 MCUs and dsPIC DSCs User's Guide	1/26/2010 10:16:32 AM	1981 KB	📄
MPLAB IDE Current Release Notes	5/11/2011 12:58:00 PM	257 KB	📦
MPLAB IDE User's Guide	1/20/2009 12:09:31 PM	4232 KB	📄
MPLAB IDE v8.70	5/11/2011 12:53:00 PM	115712 KB	📦
Quick Guide to Microchip Development Tools	3/4/2011 10:09:50 AM	582 KB	📄
Software Solutions and Tools for the 16-bit and 32-bit Designer	5/26/2011 11:38:00 AM	3138 KB	📄
The MPLAB IDE Debug Tool API	5/13/2010 5:16:00 PM	171 KB	📄

Download MPLAB IDE v8.70


File Download

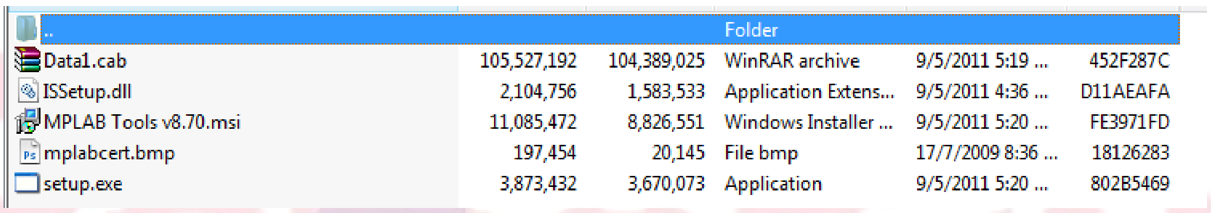
Do you want to open or save this file?

Name:	MPLAB_IDE_v8_70.zip
Type:	WinRAR ZIP archive, 113MB
From:	**ww1.microchip.com**

[✓] Always ask before opening this type of file

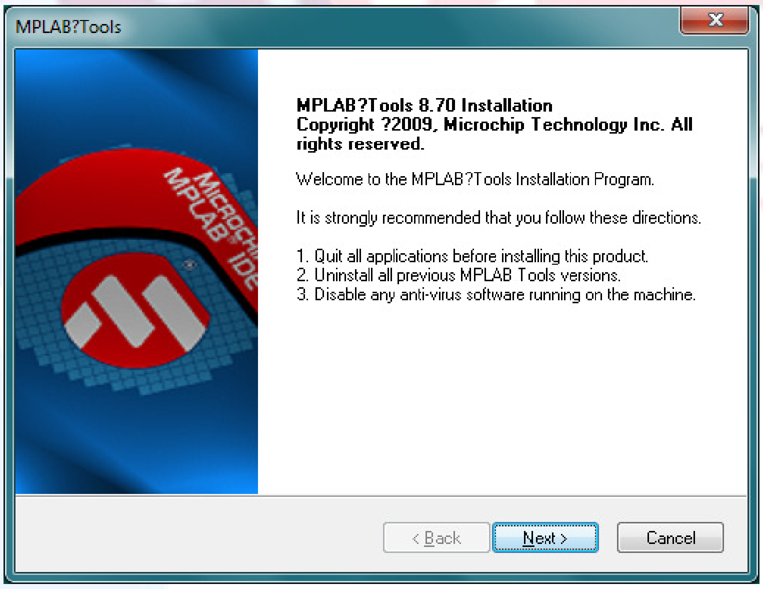
While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

Open the MPLAB_IDE_v8_70.zip file after finishing Downloading, and then double click the setup.exe file to start the installing.



..			Folder		
Data1.cab	105,527,192	104,389,025	WinRAR archive	9/5/2011 5:19 ...	452F287C
ISSetup.dll	2,104,756	1,583,533	Application Extens...	9/5/2011 4:36 ...	D11AEAFA
MPLAB Tools v8.70.msi	11,085,472	8,826,551	Windows Installer ...	9/5/2011 5:20 ...	FE3971FD
mplabcert.bmp	197,454	20,145	File bmp	17/7/2009 8:36 ...	18126283
setup.exe	3,873,432	3,670,073	Application	9/5/2011 5:20 ...	802B5469

Simply follow the following steps for the installing of MPLAB_IDE.



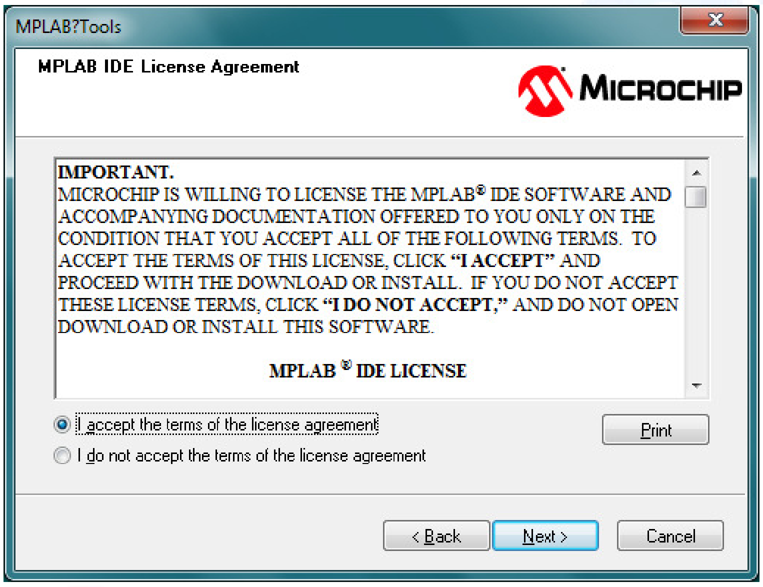
MPLAB?Tools 8.70 Installation
Copyright ?2009, Microchip Technology Inc. All rights reserved.

Welcome to the MPLAB?Tools Installation Program.

It is strongly recommended that you follow these directions.

1. Quit all applications before installing this product.
2. Uninstall all previous MPLAB Tools versions.
3. Disable any anti-virus software running on the machine.

Click the "Next >" button


MPLAB?Tools

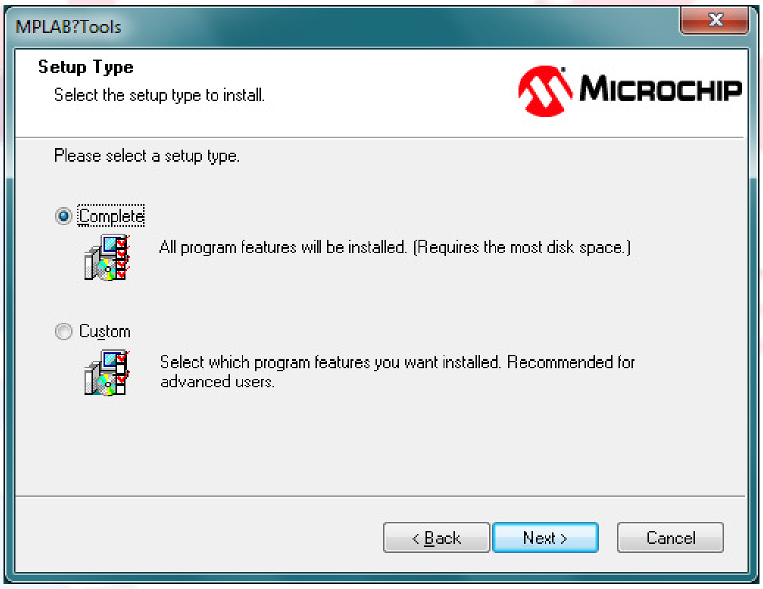
MPLAB IDE License Agreement

IMPORTANT.
MICROCHIP IS WILLING TO LICENSE THE MPLAB® IDE SOFTWARE AND ACCOMPANYING DOCUMENTATION OFFERED TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE FOLLOWING TERMS. TO ACCEPT THE TERMS OF THIS LICENSE, CLICK **"I ACCEPT"** AND PROCEED WITH THE DOWNLOAD OR INSTALL. IF YOU DO NOT ACCEPT THESE LICENSE TERMS, CLICK **"I DO NOT ACCEPT,"** AND DO NOT OPEN DOWNLOAD OR INSTALL THIS SOFTWARE.

MPLAB® IDE LICENSE

[✓ I accept the terms of the license agreement]
I do not accept the terms of the license agreement

Choose the option "I accept the terms of the license agreement" and click the "Next >" button.


MPLAB?Tools

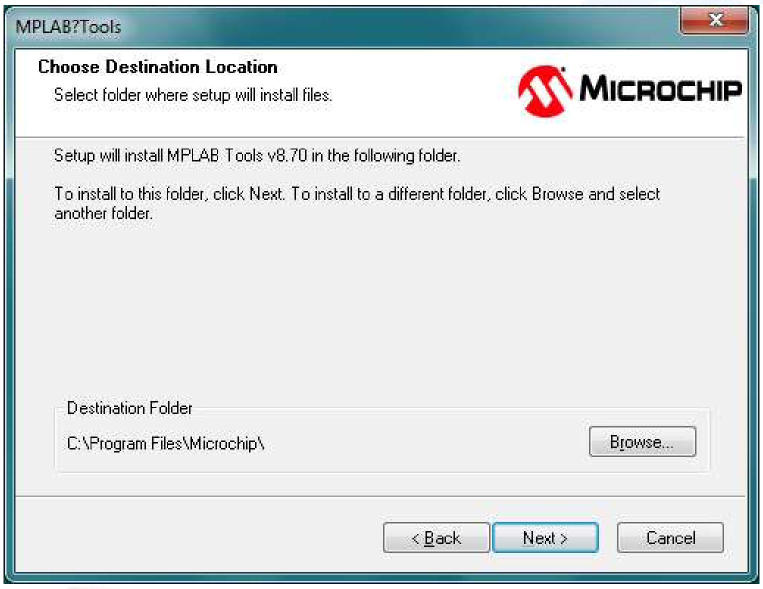
Setup Type
Select the setup type to install.

Please select a setup type.

[✓ Complete]
All program features will be installed. (Requires the most disk space.)

Custom
Select which program features you want installed. Recommended for advanced users.

Choose "Complete" option and then click the "Next>" button.


MPLAB?Tools

Choose Destination Location

Select folder where setup will install files.

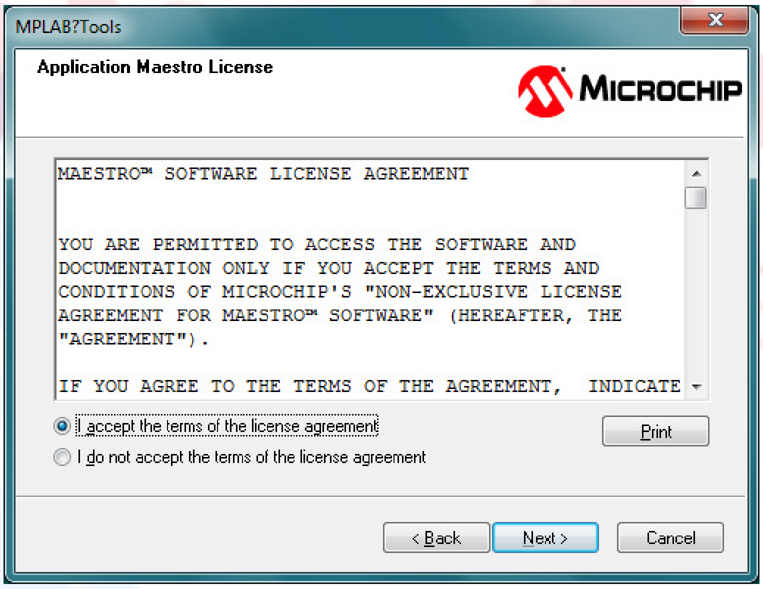
Setup will install MPLAB Tools v8.70 in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\Microchip\

Choose the Destination Folder you want to install the MPLAB_IDE, and then click the "Next >" button.


MPLAB?Tools

Application Maestro License

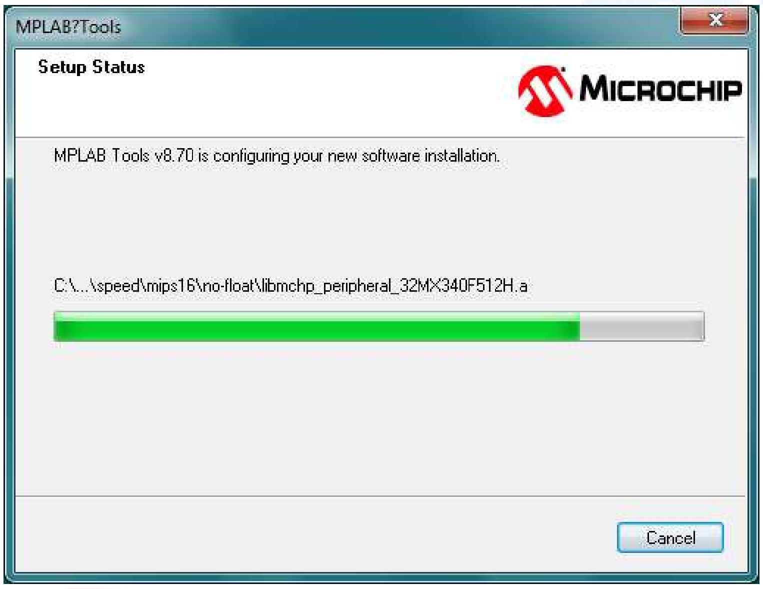
```
MAESTRO™ SOFTWARE LICENSE AGREEMENT


YOU ARE PERMITTED TO ACCESS THE SOFTWARE AND
DOCUMENTATION ONLY IF YOU ACCEPT THE TERMS AND
CONDITIONS OF MICROCHIP'S "NON-EXCLUSIVE LICENSE
AGREEMENT FOR MAESTRO™ SOFTWARE" (HEREAFTER, THE
"AGREEMENT").

IF YOU AGREE TO THE TERMS OF THE AGREEMENT,  INDICATE
```

[✓ I accept the terms of the license agreement]

I do not accept the terms of the license agreement

Choose the option "I accept the terms of the license agreement", and then click the "Next >" button.

And click the "Next >" button again then the following installing screen will show:

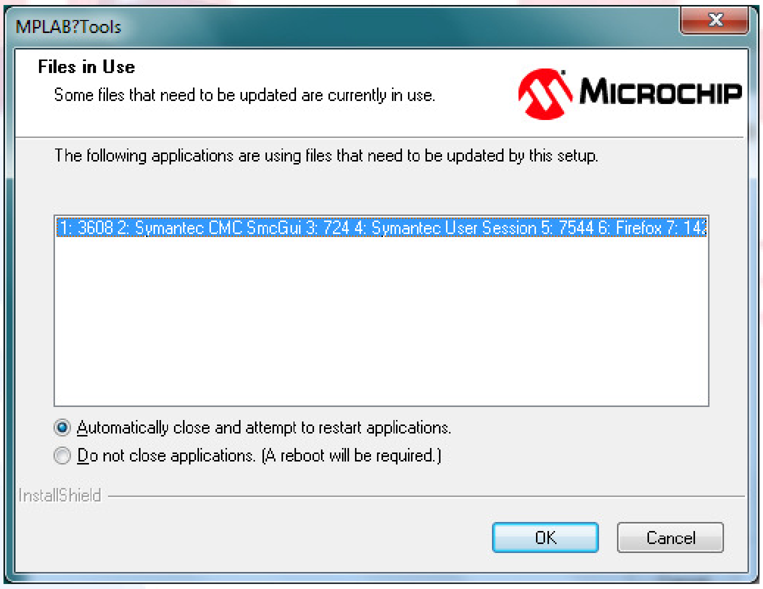

MPLAB?Tools

Setup Status

MPLAB Tools v8.70 is configuring your new software installation.

C:\...\speed\mips16\no-float\libmchp_peripheral_32MX340F512H.a

During the setup, you may counter the following situation:


MPLAB?Tools

Files in Use

Some files that need to be updated are currently in use.

The following applications are using files that need to be updated by this setup.

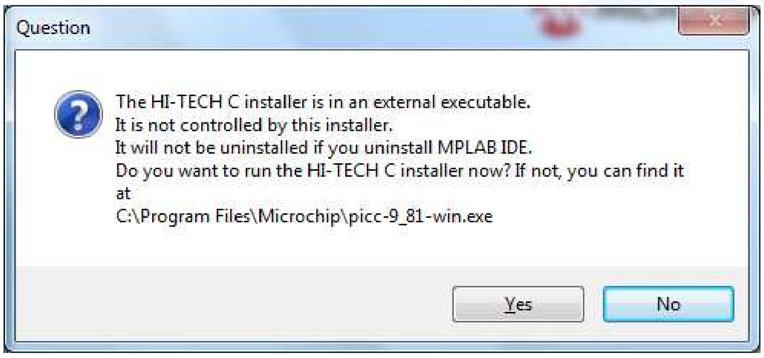
1: 3608 2: Symantec CMC SmcGui 3: 724 4: Symantec User Session 5: 7544 6: Firefox 7: 142

[✓ Automatically close and attempt to restart applications.]

Do not close applications. (A reboot will be required.)

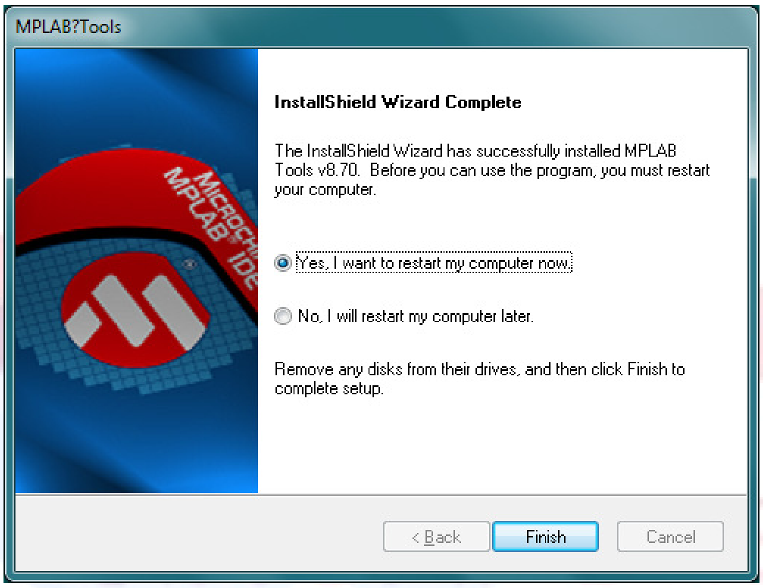
You need to close the application mentioned in the text book, for example, the IE browser or Firefox browser. Then click the “OK” button to continue.

Or you can choose the option “Do not close applications.(A reboot will be required.)”, and click the “OK “button to continue installing, make sure you reboot you computer after finish the setup.


Question

The HI-TECH C installer is in an external executable.
It is not controlled by this installer.
It will not be uninstalled if you uninstall MPLAB IDE.
Do you want to run the HI-TECH C installer now? If not, you can find it at
C:\Program Files\Microchip\picc-9_81-win.exe

If you encounter the above window, just click the No button, because you will not need the HI-TECH C compiler in this project.


MPLAB?Tools

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed MPLAB Tools v8.70. Before you can use the program, you must restart your computer.

[✓ Yes, I want to restart my computer now.]

No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Click the “Finish” button and restart you computer to finish install the MPLAB IDE.

3.2 Install MPLAB C Compiler for PIC24 MCU and dsPIC DSCs

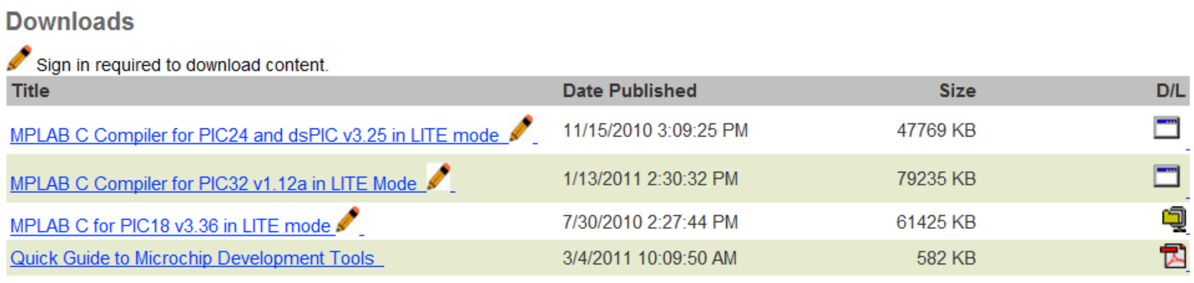
<u>What is MPLAB C Compiler?</u>

The MPLAB C Compiler for Academic Use are the LITE versions of the fully ANSI compliant products with standard libraries for Microchip’s PIC18, PIC24, dsPIC DSC, and PIC32 families. They take advantage of the PIC MCU and dsPIC DSC architectures to provide highly efficient software code generation. The MPLAB C compilers provide extensions for in-depth support such as interrupts and peripherals and special function registers. They are fully integrated with the MPLAB IDE with a full-featured programmer’s editor, a graphical project manager and high level, source debugging. These compilers come complete with assembler, linker and librarian for mixed mode C and assembly programs.

There is a special free version for academic use. You need to register before installation.

Download link:

https://www.microchip.com/wwwregister/default.aspx?ReturnURL=http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en536656

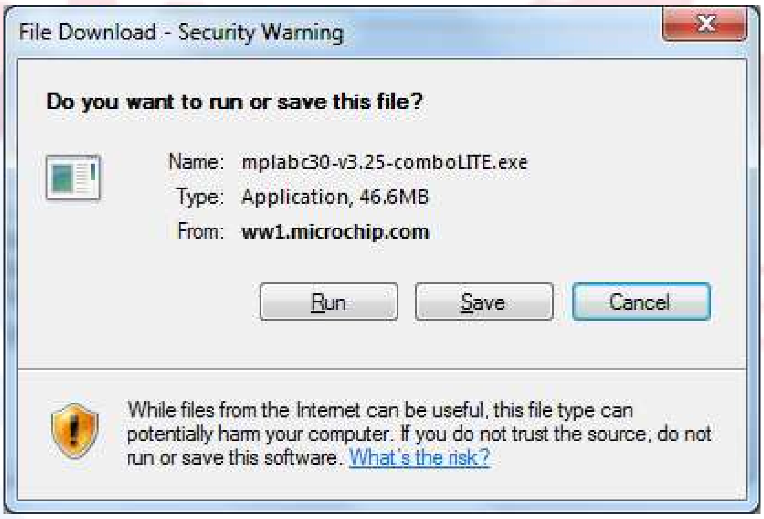


Downloads

Sign in required to download content.

Title	Date Published	Size	D/L
MPLAB C Compiler for PIC24 and dsPIC v3.25 in LITE mode	11/15/2010 3:09:25 PM	47769 KB	
MPLAB C Compiler for PIC32 v1.12a in LITE Mode	1/13/2011 2:30:32 PM	79235 KB	
MPLAB C for PIC18 v3.36 in LITE mode	7/30/2010 2:27:44 PM	61425 KB	
Quick Guide to Microchip Development Tools	3/4/2011 10:09:50 AM	582 KB	

Download "MPLAB C Compiler for PIC24 and dsPIC v3.25 in LITE mode"



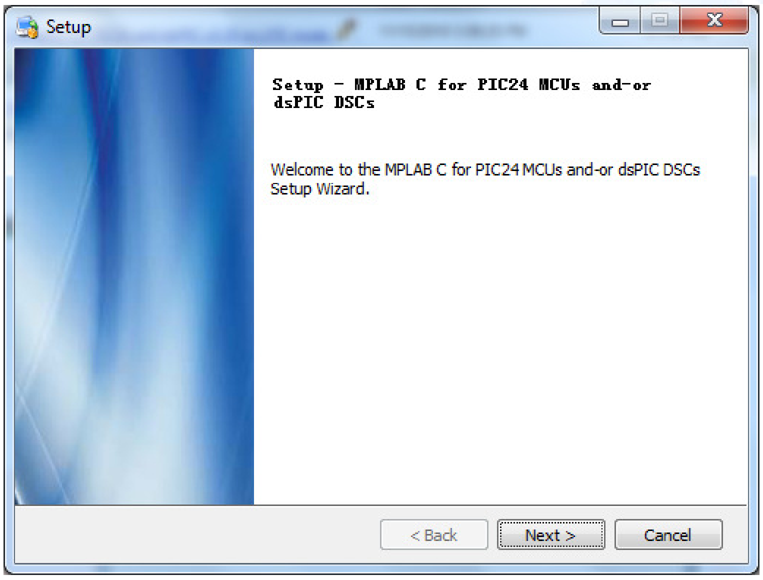
File Download - Security Warning

Do you want to run or save this file?

Name: mplabc30-v3.25-comboLITE.exe
Type: Application, 46.6MB
From: ww1.microchip.com

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

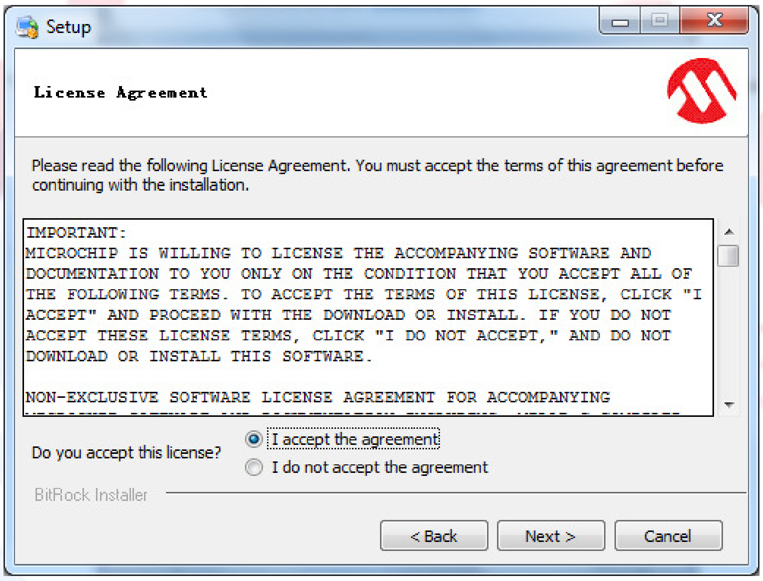
Click the "mplabc30-v3.25-comboLITE.exe" file to start the installation after finishing downloading.


Setup

Setup - MPLAB C for PIC24 MCUs and-or dsPIC DSCs

Welcome to the MPLAB C for PIC24 MCUs and-or dsPIC DSCs Setup Wizard.

Click the “Next >” button to proceed with the installation.


Setup

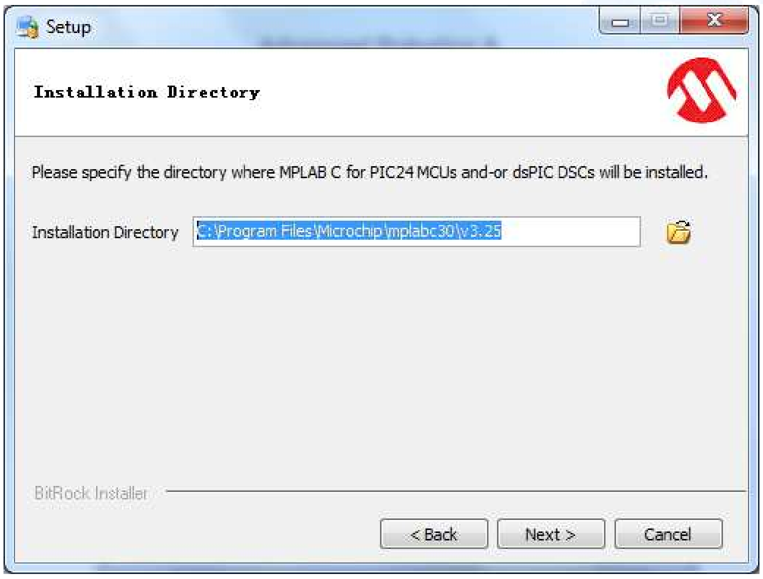
License Agreement

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

```
IMPORTANT:
MICROCHIP IS WILLING TO LICENSE THE ACCOMPANYING SOFTWARE AND
DOCUMENTATION TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF
THE FOLLOWING TERMS. TO ACCEPT THE TERMS OF THIS LICENSE, CLICK "I
ACCEPT" AND PROCEED WITH THE DOWNLOAD OR INSTALL. IF YOU DO NOT
ACCEPT THESE LICENSE TERMS, CLICK "I DO NOT ACCEPT," AND DO NOT
DOWNLOAD OR INSTALL THIS SOFTWARE.

NON-EXCLUSIVE SOFTWARE LICENSE AGREEMENT FOR ACCOMPANYING
```

Do you accept this license? [✓ I accept the agreement] I do not accept the agreement

Choose the option “I accept the agreement” and then click the “Next >” button.

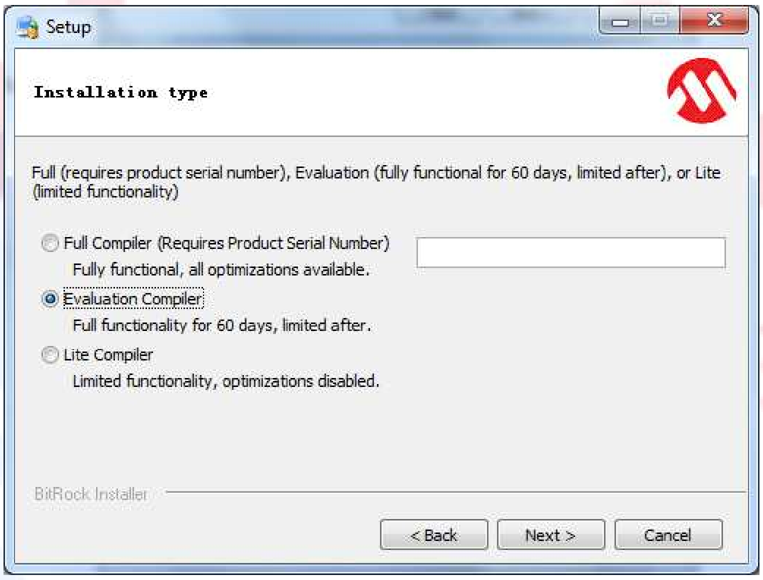

Setup

Installation Directory

Please specify the directory where MPLAB C for PIC24 MCUs and-or dsPIC DSCs will be installed.

Installation Directory: C:\Program Files\Microchip\mplabc30\v3.25

Choose the Installation Directory and then click the “Next >” button.


Setup

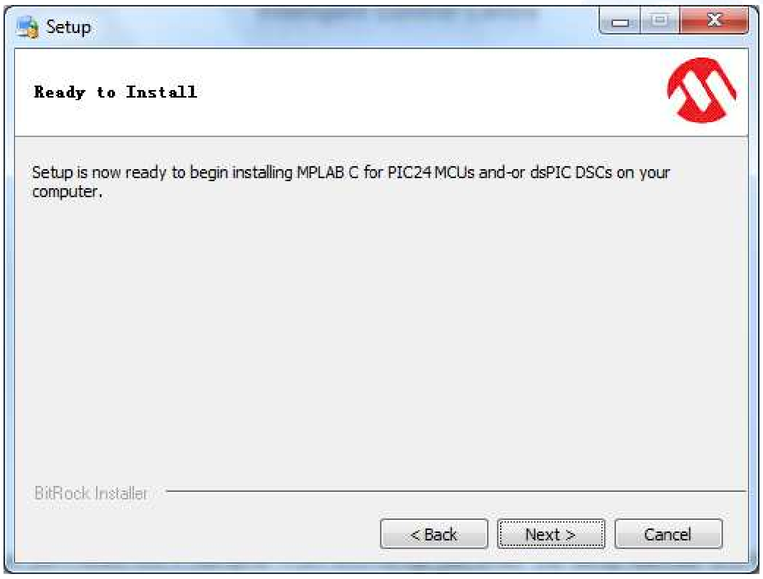
Installation type

Full (requires product serial number), Evaluation (fully functional for 60 days, limited after), or Lite (limited functionality)

- ( ) Full Compiler (Requires Product Serial Number)
  Fully functional, all optimizations available.
- [✓ Evaluation Compiler]
  Full functionality for 60 days, limited after.
- ( ) Lite Compiler
  Limited functionality, optimizations disabled.

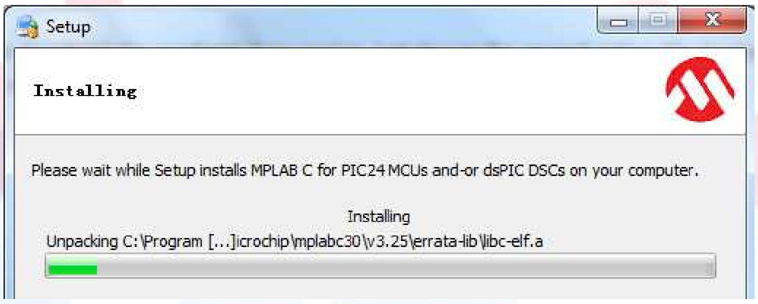
If you have purchased this C compiler from Microchip, just enter the Serial Number and then click the “Next >” button to continue.

If you just download the academic free version, just choose the second option “Evaluation Compiler” and then click the “Next >” button to continue.



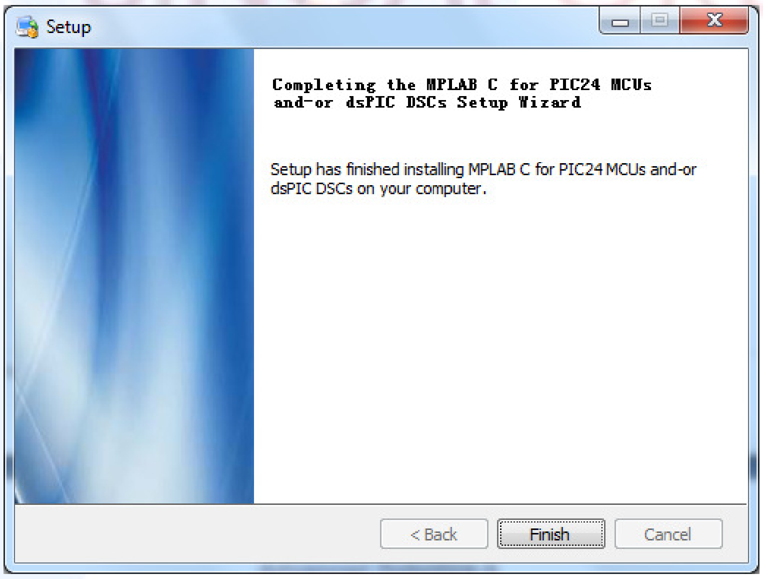
Setup is now ready to begin installing MPLAB C for PIC24 MCUs and-or dsPIC DSCs on your computer.

Click the “Next >” button to continue the installation.



Please wait while Setup installs MPLAB C for PIC24 MCUs and-or dsPIC DSCs on your computer.

Installing
Unpacking C:\Program [...]icrochip\mplabc30\v3.25\errata-lib\libc-elf.a



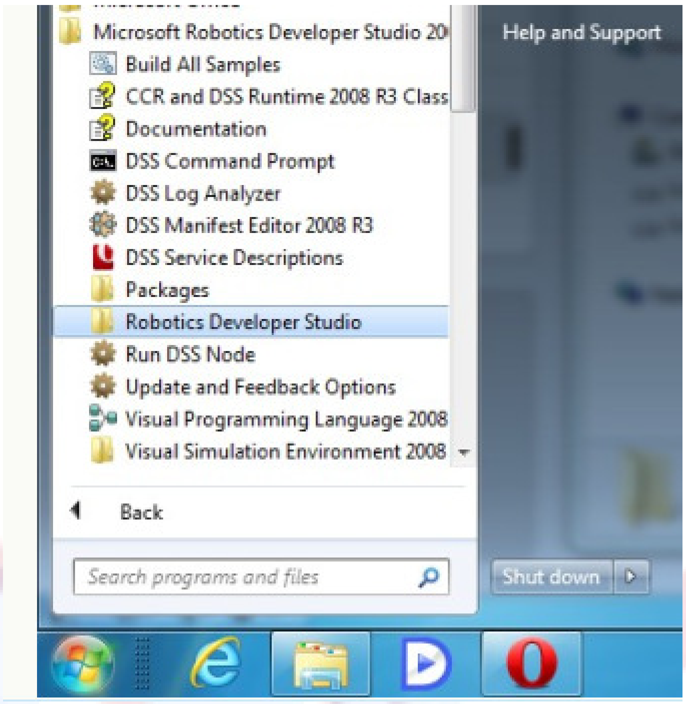
Completing the MPLAB C for PIC24 MCUs and-or dsPIC DSCs Setup Wizard

Setup has finished installing MPLAB C for PIC24 MCUs and-or dsPIC DSCs on your computer.

Click the “Finish” button to finish the installation.

3.3 Build the project file in the MPLAB IDE

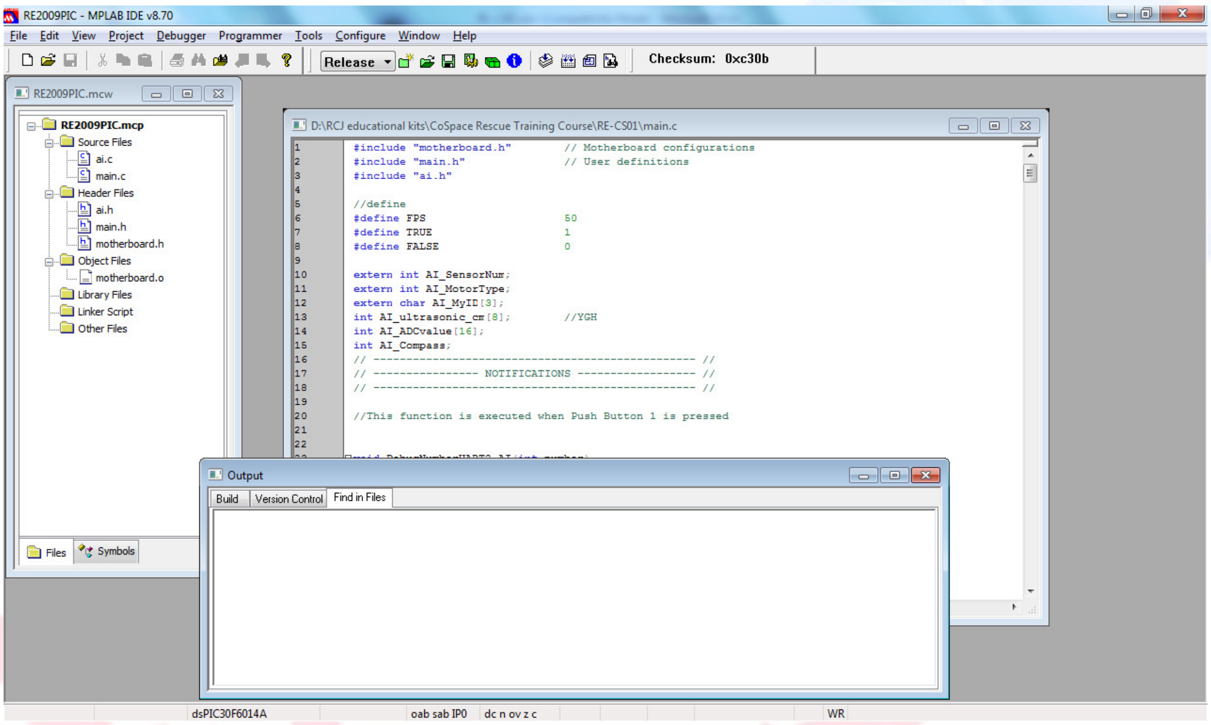
After you have successfully installed MPLAB IDE, C Compiler, Microsoft Robotics Develop Studio and CsBot Simulator, click the Robotics Developer Studio folder as shown below


Microsoft Robotics Developer Studio 20
Build All Samples
CCR and DSS Runtime 2008 R3 Class
Documentation
DSS Command Prompt
DSS Log Analyzer
DSS Manifest Editor 2008 R3
DSS Service Descriptions
Packages
Robotics Developer Studio
Run DSS Node
Update and Feedback Options
Visual Programming Language 2008
Visual Simulation Environment 2008
Back
Search programs and files
Help and Support
Shut down

Then go to the following directory:

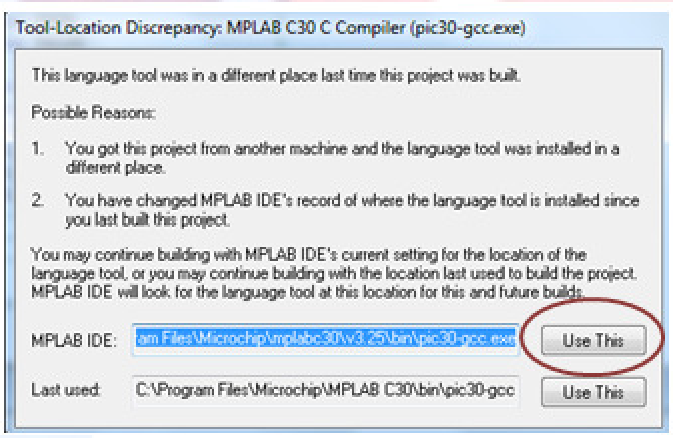
Microsoft Robotics Dev Studio 2008 R3\CoSpace\DIYWheel\RealAI\CsBoard-03

Open the RE2009PIC Project file in MPLAB IDE. You will see some window like below:



```
#include "motherboard.h"        // Motherboard configurations
#include "main.h"               // User definitions
#include "ai.h"

//define
#define FPS                     50
#define TRUE                    1
#define FALSE                   0

extern int AI_SensorNum;
extern int AI_MotorType;
extern char AI_MyID[3];
int AI_ultrasonic_cm[8];        //YGH
int AI_ADCvalue[16];
int AI_Compass;
// ------------------------------------------------ //
// --------------- NOTIFICATIONS ------------------ //
// ------------------------------------------------ //

//This function is executed when Push Button 1 is pressed
```

Choose from menu: “Project → Build All” or press “Ctrl + F10” to build the project.


Tool-Location Discrepancy: MPLAB C30 C Compiler (pic30-gcc.exe)

This language tool was in a different place last time this project was built.

Possible Reasons:

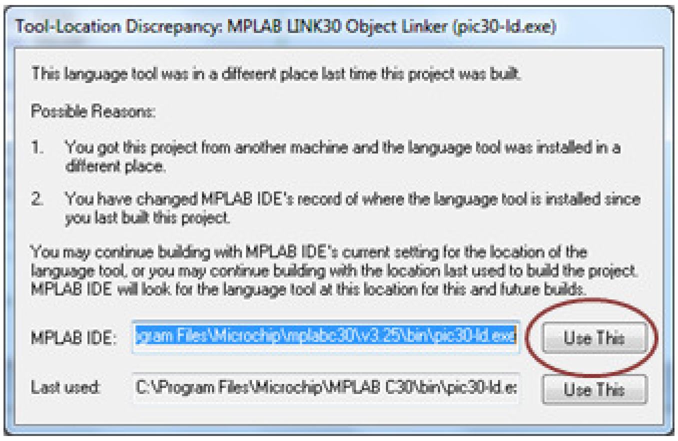
1. You got this project from another machine and the language tool was installed in a different place.
2. You have changed MPLAB IDE's record of where the language tool is installed since you last built this project.

You may continue building with MPLAB IDE's current setting for the location of the language tool, or you may continue building with the location last used to build the project. MPLAB IDE will look for the language tool at this location for this and future builds.

MPLAB IDE: am Files\Microchip\mplabc30\v3.25\bin\pic30-gcc.exe [✓ Use This]

Last used: C:\Program Files\Microchip\MPLAB C30\bin\pic30-gcc Use This

Make sure you choose the TOP “Use This” when you account the above window.



Tool-Location Discrepancy: MPLAB LINK30 Object Linker (pic30-ld.exe)

This language tool was in a different place last time this project was built.

Possible Reasons:

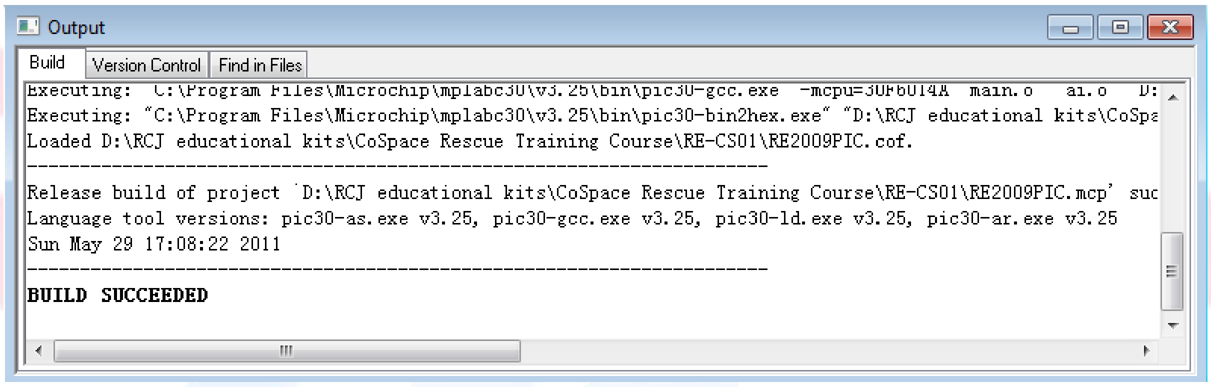
1. You got this project from another machine and the language tool was installed in a different place.
2. You have changed MPLAB IDE's record of where the language tool is installed since you last built this project.

You may continue building with MPLAB IDE's current setting for the location of the language tool, or you may continue building with the location last used to build the project. MPLAB IDE will look for the language tool at this location for this and future builds.

MPLAB IDE: gram Files\Microchip\mplabc30\v3.25\bin\pic30-ld.exe [✓ Use This]

Last used: C:\Program Files\Microchip\MPLAB C30\bin\pic30-ld.e: Use This

Make sure you choose the TOP “Use This” when you account the above window as well.



```
Executing: C:\Program Files\Microchip\mplabc30\v3.25\bin\pic30-gcc.exe  -mcpu=30F6014A  main.o   ai.o   D:
Executing: "C:\Program Files\Microchip\mplabc30\v3.25\bin\pic30-bin2hex.exe" "D:\RCJ educational kits\CoSpa
Loaded D:\RCJ educational kits\CoSpace Rescue Training Course\RE-CS01\RE2009PIC.cof.
----------------------------------------------------------------------
Release build of project `D:\RCJ educational kits\CoSpace Rescue Training Course\RE-CS01\RE2009PIC.mcp' suc
Language tool versions: pic30-as.exe v3.25, pic30-gcc.exe v3.25, pic30-ld.exe v3.25, pic30-ar.exe v3.25
Sun May 29 17:08:22 2011
----------------------------------------------------------------------
BUILD SUCCEEDED
```

And make sure it shows “BUILD SUCCEEDED” in the output window.

3.4: Install PICKIT 2

What is PICKIT 2?

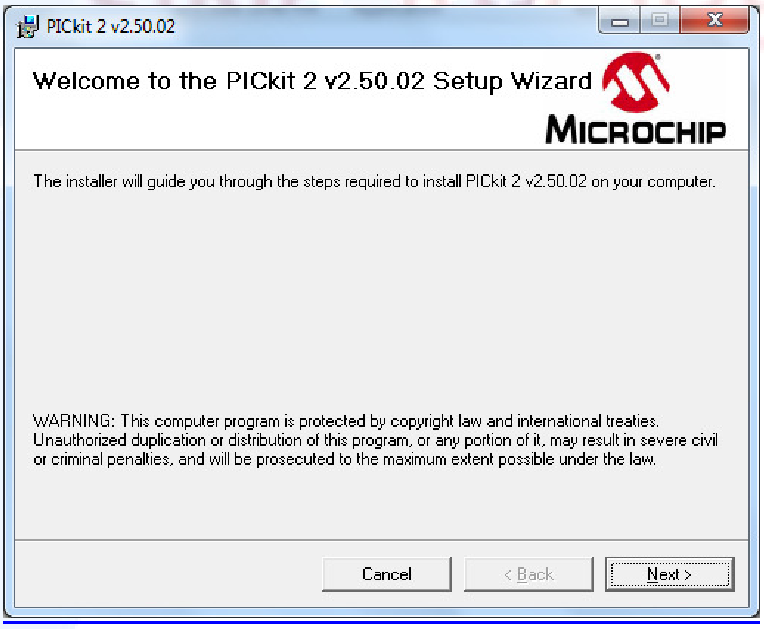
The PICkit™ 2 Development Programmer/Debugger (PG164120) is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

You need to purchase the PICKit 2. You can purchase it directly from Microchips official website:

Link:

http://www.microchipdirect.com/ProductSearch.aspx?Keywords=DV164121

Click the "Setup.exe" file to start the installation after finishing downloading.

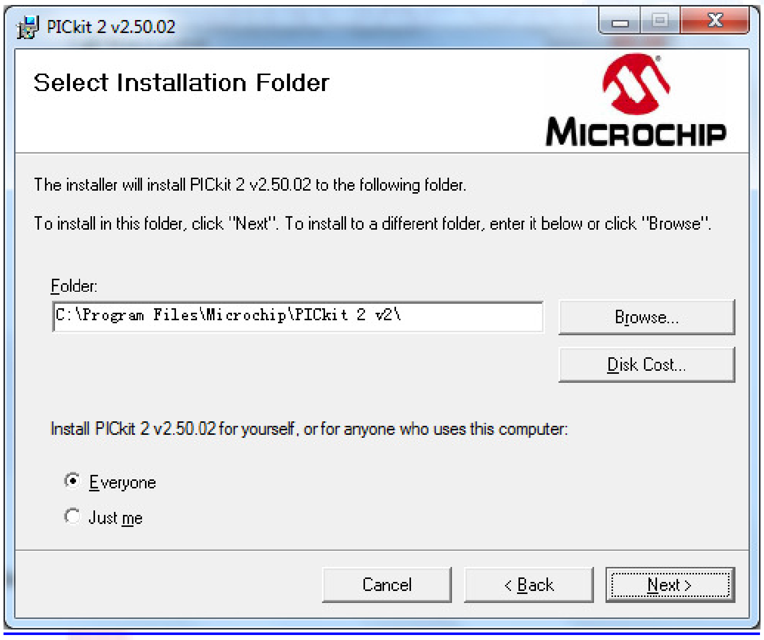

PICkit 2 v2.50.02

Welcome to the PICkit 2 v2.50.02 Setup Wizard

The installer will guide you through the steps required to install PICkit 2 v2.50.02 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Click the "Next >" button to start the installation.


PICkit 2 v2.50.02

Select Installation Folder

The installer will install PICkit 2 v2.50.02 to the following folder.

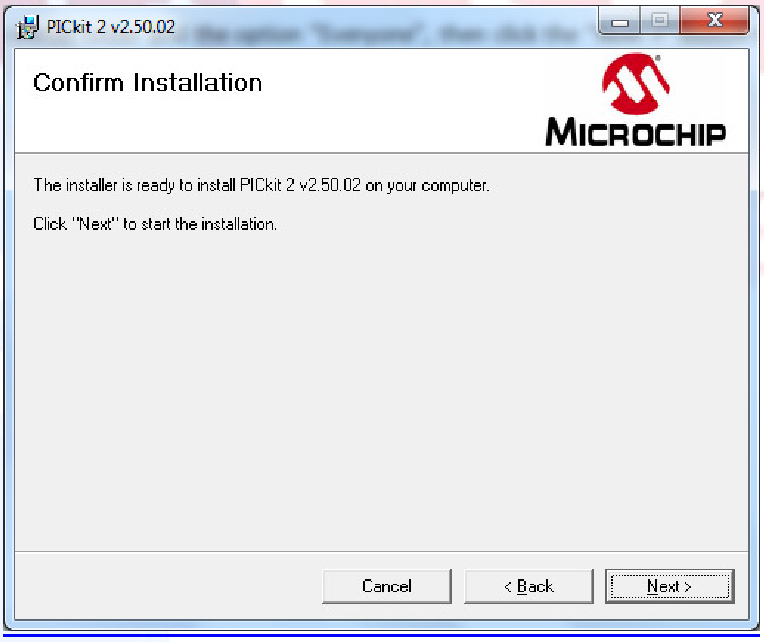
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files\Microchip\PICkit 2 v2\

Install PICkit 2 v2.50.02 for yourself, or for anyone who uses this computer:

(●) Everyone
( ) Just me

Choose the install folder and the option “Everyone”, then click the “Next >” button to continue.

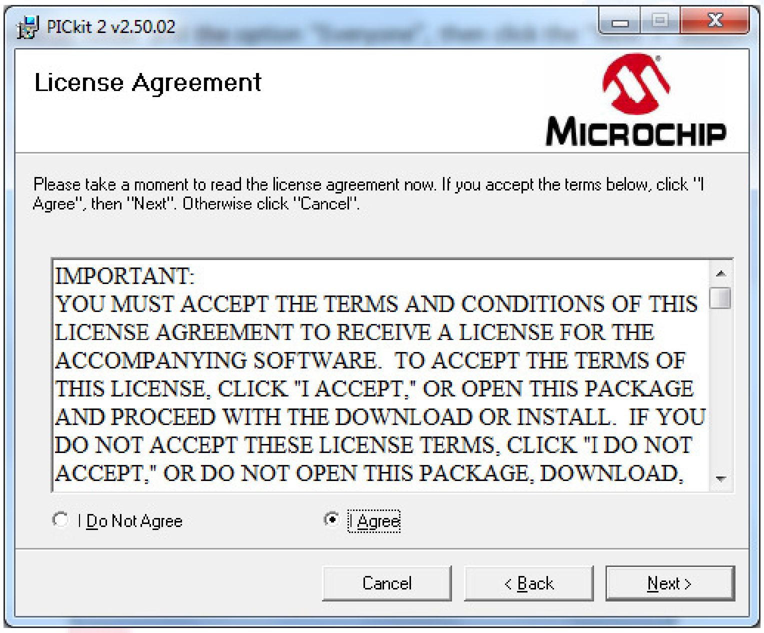

PICkit 2 v2.50.02

Confirm Installation

The installer is ready to install PICkit 2 v2.50.02 on your computer.

Click "Next" to start the installation.

Click the “Next >” button to continue.


PICkit 2 v2.50.02

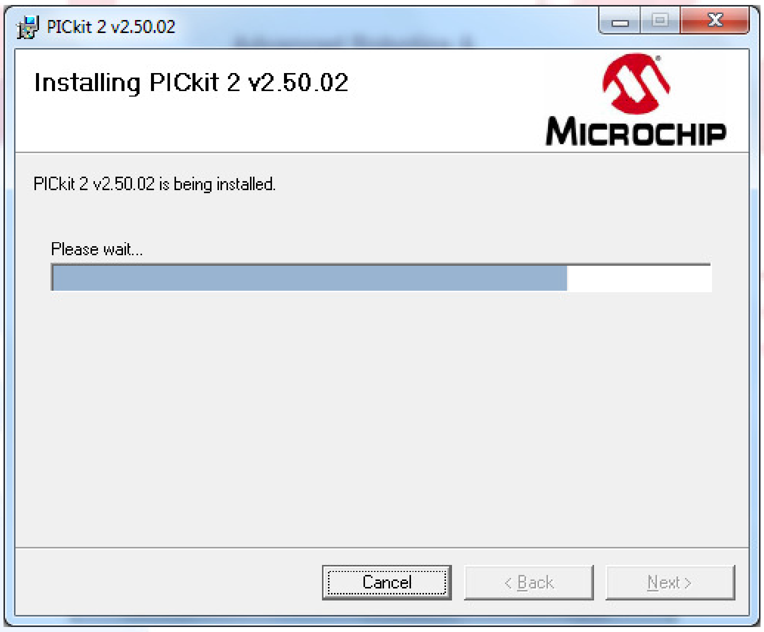
License Agreement

Please take a moment to read the license agreement now. If you accept the terms below, click "I Agree", then "Next". Otherwise click "Cancel".

IMPORTANT:
YOU MUST ACCEPT THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT TO RECEIVE A LICENSE FOR THE ACCOMPANYING SOFTWARE. TO ACCEPT THE TERMS OF THIS LICENSE, CLICK "I ACCEPT," OR OPEN THIS PACKAGE AND PROCEED WITH THE DOWNLOAD OR INSTALL. IF YOU DO NOT ACCEPT THESE LICENSE TERMS, CLICK "I DO NOT ACCEPT," OR DO NOT OPEN THIS PACKAGE, DOWNLOAD,

I Do Not Agree [✓ I Agree]

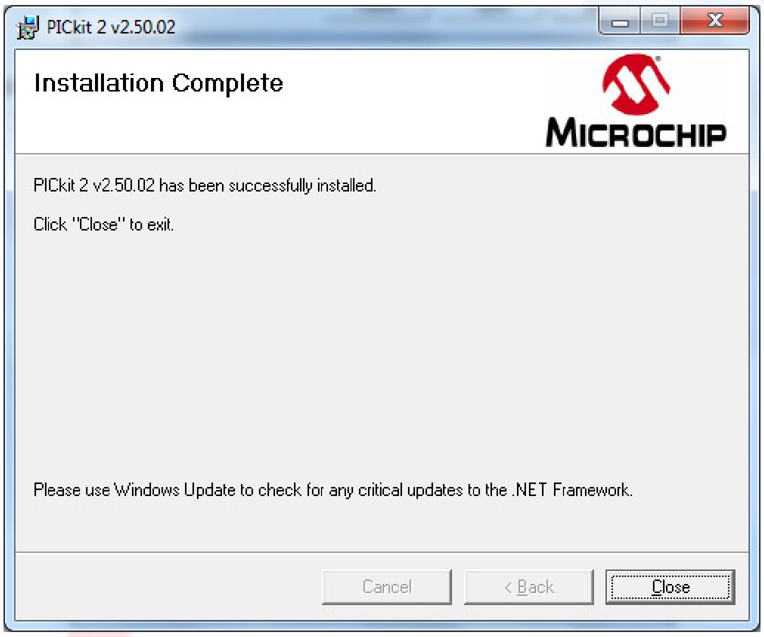
Choose “I Agree” Option and click the “Next >” button to continue.


PICkit 2 v2.50.02

Installing PICkit 2 v2.50.02

PICkit 2 v2.50.02 is being installed.

Please wait...



PICkit 2 v2.50.02 has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Click the “Close” button to finish the installation.

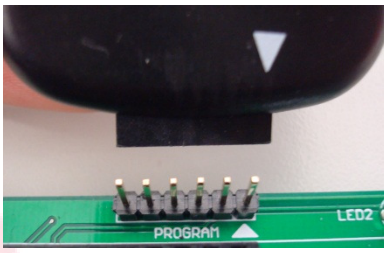
3.5. Downloading Program

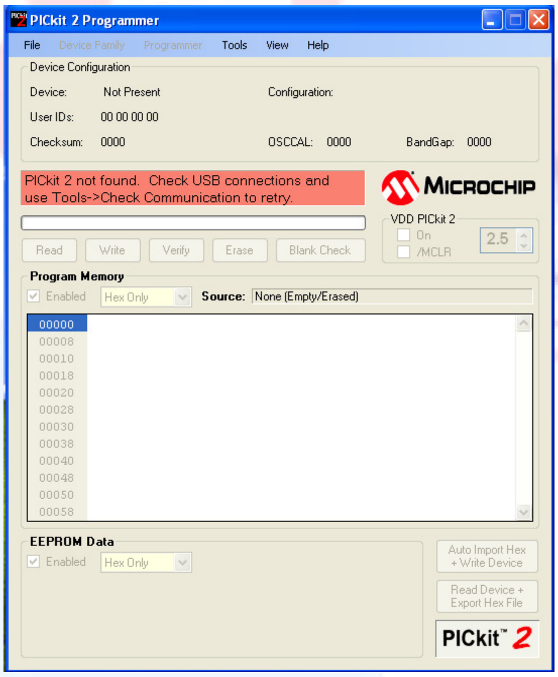
Step 1: Click onto this icon and launch the programmer. PICkit 2 v2.50

Step 2: The PICkit 2 programmer window will appear.

Make sure that the USB connection and the programmer pin are well connected.

"Tools-> Check communication"





Device Configuration

Device: Not Present
Configuration:
User IDs: 00 00 00 00
Checksum: 0000
OSCCAL: 0000
BandGap: 0000

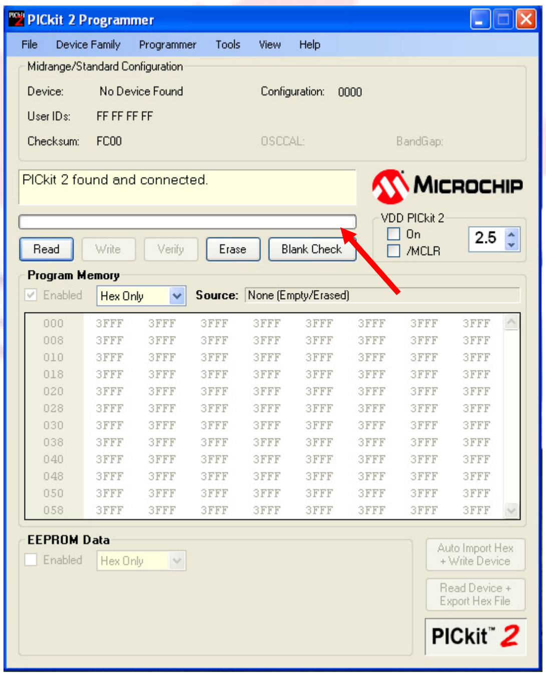
PICkit 2 not found. Check USB connections and use Tools->Check Communication to retry.

VDD PICkit 2: On ☐ /MCLR ☐ 2.5

Program Memory
Enabled ✓ Hex Only Source: None (Empty/Erased)

```
00000
00008
00010
00018
00020
00028
00030
00038
00040
00048
00050
00058
```

EEPROM Data
Enabled ✓ Hex Only



Midrange/Standard Configuration

Device: No Device Found
Configuration: 0000
User IDs: FF FF FF FF
Checksum: FC00
OSCCAL:
BandGap:

PICkit 2 found and connected.

VDD PICkit 2: On ☐ /MCLR ☐ 2.5

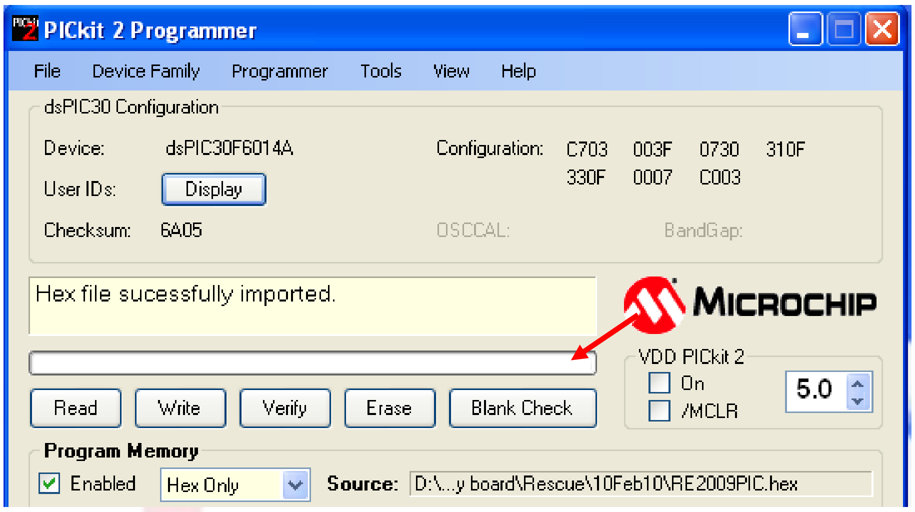
Program Memory
Enabled ✓ Hex Only Source: None (Empty/Erased)

```
000  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
008  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
010  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
018  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
020  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
028  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
030  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
038  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
040  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
048  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
050  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
058  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

EEPROM Data
Enabled ☐ Hex Only

Step 3: Import the hex file to be written into the controller board. "File-> Import Hex"

Step 4: Write the hex file into the controller board by clicking onto the "Write" button.

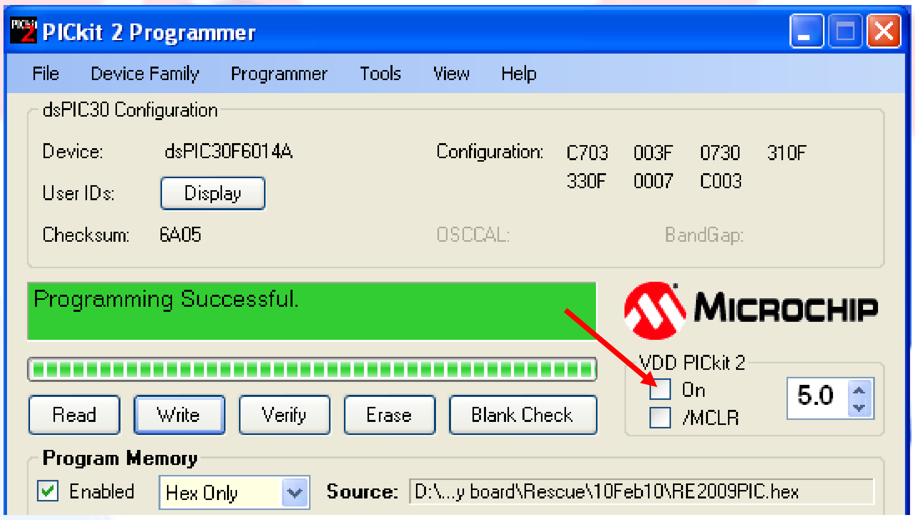


dsPIC30 Configuration

Device: dsPIC30F6014A
User IDs:
Checksum: 6A05
Configuration: C703 003F 0730 310F 330F 0007 C003
OSCCAL:
BandGap:

Hex file sucessfully imported.

Program Memory
Enabled
Source: D:\...y board\Rescue\10Feb10\RE2009PIC.hex



dsPIC30 Configuration

Device: dsPIC30F6014A
User IDs:
Checksum: 6A05
Configuration: C703 003F 0730 310F 330F 0007 C003
OSCCAL:
BandGap:

Programming Successful.

Program Memory
Enabled
Source: D:\...y board\Rescue\10Feb10\RE2009PIC.hex

Step 5: You can turn on the controller board by clicking on the "On" button to test the controller board. However, the voltage provided by the PC is not strong enough to power the motor.

4: Program Guide

4.1: Start-up and Configure

Power-on:

When powered on, all peripherals on the boards are initialized to default state. Initialize Program function is to be executed after all the peripherals and ports are initialized. Further configuration of the motherboard can be done here. For example, types of motor, sensors can be selected here.

Normal program:

Sensors connected to the boards are updated every 20 milliseconds (50 times per second) after which a special function named ExecuteProgram is called (every 20 milliseconds). This function is the main entry point for users to customize the motherboard for various projects. There are easy to use functions available for users to retrieve current sensor values, battery voltage, display texts on LED, etc.

SetMotorControl(SINGLE)	Set all motor port to use 3-pin type motors
SetMotorControl(DUO)	Set all motor port to use 4-pin type motors

4.2 Special Events

An event occurs when there is a change to the push buttons or when a new character is received by communication ports.

void PushButton1Pressed(void)	Executed when Push Button 1 is pressed
void PushButton1Released(void)	Executed when Push Button 1 is released
void PushButton2Pressed(void)	Executed when Push Button 2 is pressed
void PushButton1Released(void)	Executed when Push Button 2 is released
GetButton1PressedTime()	To be used in button-release event to obtain the duration (in milliseconds) in which Push Button 1 was pressed
GetButton2PressedTime()	To be used in button-release event to obtain the duration (in milliseconds) in which Push Button 2 was pressed
void COM1ReceiveCharacter(char newChar)	Executed when a new character is received at COM1
void COM2ReceiveCharacter(char newChar)	Executed when a new character is received by Zigbee

4.3 Functions

TurnOnLED, TurnOffLED, ToggleLED

Description: Turn on/off or toggle current state of LED

Prototype:

```
void TurnOnLED(int number);
void TurnOffLED(int number);
void ToggleLED(int number);
```

Arguments: Integer value of LED number 1~4

Return value: None

Remarks: Nothing happens if the argument is out of range.

Code example:

```
TurnOnLED(2);        //Turn on LED2 with integer value of 2
```

StartMotherboard, StopMotherboard

Description: Start/stop execution of ExecuteProgram function (every 20 milliseconds)

Prototype:

```
void StartMotherboard(void);
void StopMotherboard(void);
```

Arguments: None

Return value: None

Remarks: Nothing happens if the argument is out of range.

Code example:

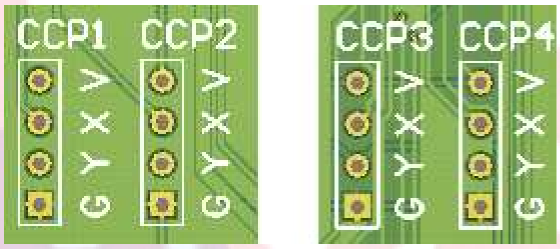
```
StartMotherboard();   //Start execute program
StopMotherboard();    //Stop execute program
```

GetUltrasonic

Description:	Obtain ultrasonic sensor value
Prototype:	float GetUltrasonic(int index);
Arguments:	Integer value of CCP sensor number 1~4
Return value:	Floating value in millimeter
Remarks:	Nothing happens if the argument is out of range.

Code example:

```
GetUltrasonic(2);       //Get ultrasonic2 value in millimeter
```

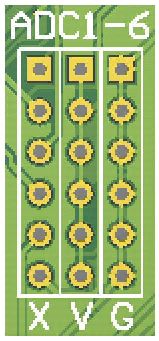


GetADCSensor

Description:	Obtain ADC sensor value
Prototype:	unsigned int GetADCSensor(int index);
Arguments:	Integer value of ADC sensor number 1~6
Return value:	Unsigned integer
Remarks:	Nothing happens if the argument is out of range.

Code example:

```
GetADCSensor(2);     //Get ADC sensor 2 value
```



GetIOSensor

Description: Obtain IO sensor value or status

Prototype: int GetIOSensor(int index);

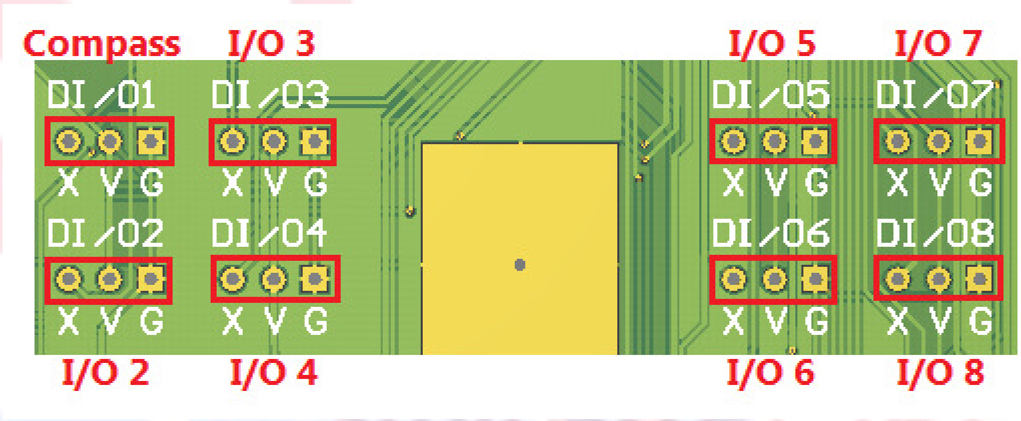
Arguments: Integer value of IO sensor number 2~8 (IO sensor 1 is reserved for Compass)

Return value: Integer value

Remarks: Nothing happens if the argument is out of range.

Code example:

```
GetIOSensor(2);        //Get IO sensor 2 value
```



GetCompass

Description: Obtain compass sensor value

Prototype: float GetCompass(void);

Arguments: None

Return value: Floating value for 0 to 360 degree

Remarks: Nothing happens if the argument is out of range.

Code example:

```
GetCompass();          //Get compass value in degree
```

SetMotorControl

Description: Set number of motor control pin

Prototype: void SetMotorControl(int MotorPin);

Arguments: SINGLE for one control pin or DUO for two control pins

Return value: None

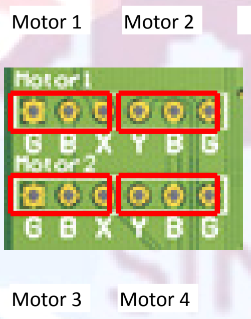
Remarks: SINGLE motor control pin can control up to 4 motors

DUO motor control pins can control up to 2 motors

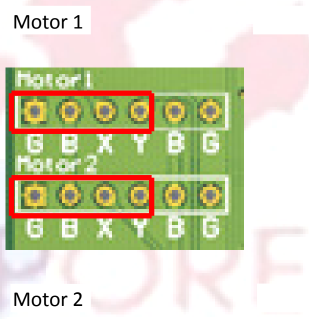
Code example:

```
SetMotorControl(SINGLE);   //Set  one motor control pin
```

<u>SINGLE motor control</u>



<u>DUO motor control</u>



SetWheelOnePercentage, SetWheelTwoPercentage,

SetWheelThreePercentage.SetWheelFourPercentage

Description:	Set wheel velocity
Prototype:	void SetWheelOnePercentage(float percentage);
	void SetWheelTwoPercentage(float percentage);
	void SetWheelThreePercentage(float percentage);
	void SetWheelFourPercentage(float percentage);
Arguments:	Floating value between 0.0% to 100.0% or 0.0% to -100.0%
Return value:	None
Remarks:	Max value is 100.0% and min value is -100.0% if the argument is out of range.

Code example:

```
SetWheelThreePercentage(50.0f);     //Set wheel3 to 50% speed
```

DebugNumberUART1

Description:	Send out 5 characters to represent a number through UART1 (COM1)
Prototype:	void DebugNumberUART1(int number);
Arguments:	Integer value with maximum 5 digit
Return value:	None
Remarks:	Nothing happens if the argument is out of range.

Code example:

```
DebugNumberUART1(123)             //Send a string “00123” through UART1
```

DebugNumberUART2

Description: Send out 5 characters to represent a number through UART2 (ZigBee)

Prototype: void DebugNumberUART2(int number);

Arguments: Integer value with maximum 5 digit

Return value: None

Remarks: Nothing happens if the argument is out of range.

Code example:

```
DebugNumberUART2(12)            //Send a string "00012" through UART1
```

waitms

Description: Delay for a specific period in unit of milliseconds

Prototype: void waitms(float ms);

Arguments: Floating value in milliseconds

Return value: None

Remarks: Nothing happens if the argument is out of range.

Code example:

```
waitms(17.0f)                   //Delay for 17 milliseconds
```

wait100ms

Description: Delay for a specific period in unit of 100 milliseocnds

Prototype: void wait100ms(int ms100);

Arguments: Integer value in units of 100 milliseconds

Return value: None

Remarks: Nothing happens if the argument is out of range.

Code example:

```
wait100ms(20)         //Delay for 2000 milliseconds
```

4.4 Configure ZigBee Channel

Both the transmitting and receiving ZigBee have to be configured in the same frequency channel in order to communicate. The following are the steps to configure ZigBee frequency channel in HyperTerminal:

- Connect the USB Zigbee transmitter to computer.
- Open HyperTerminal and select the appropriate COM port.
- If the ZigBee module is a brand new one, select baud rate 9600. Otherwise, use baud rate of 115200. Leave other settings as default.
- Wait for 1 second without typing anything
- Type +++ then wait for "OK" response. If "OK" is not shown, check the COM port & baud rate settings in previous steps
- Execute the following command; wait for "OK" response after each command. If "ERROR" is receive, check for any typo mistake and execute the command again.
    - Type "ATPL4" and enter (Set maximum transmission power)
    - Type "ATDLFFFF" and enter (Set to broadcast to all in the same channel)
    - Type "ATMY0" and enter (Disable MAC address checking, ie. Receive from anyone)
    - Type "ATDB7" and enter (Set baud rate to 115200)
    - Type "ATCHxx" and enter (Set the channel, xx is a **hexadecimal** number between **C** and **17**)
    - Type "ATCH" and enter (Check current channel)
    - Type "ATWR" and enter (Save changes to memory)
    - Type "ATCN" and enter (Exit from command mode)
- The first few commands before ATCHxx are only required for configuring a brand new ZigBee module. Subsequent configurations do not need to include them.