

BEGV606M

USER MANUAL

- LCD Embedded System,
- Microchip PIC24FJ64GA002 MCU
- 64KB program flash/ 8KB SRAM
- I²C-EEPROM 64KB(128KB option)
- Character 20*2 I²C-STN LCD w/ white LED backlight
- 1* RS232 + (shared RS232/RS422/RS485)
- 3 GPIOs
- Operating at 35 mA
- Free SDK support



Table of Content

Chapter 1 Introduction	6
1.1 Features	7
1.2 Board Layout.....	7
1.3 Block Diagram	8
1.4 Mechanical Dimensions	9
1.5 Board Specifications	10
1-6 Ordering Information	10
Chapter 2 Installation	11
2.1 Connectors.....	12
2.1.1 Connector CN1 illustration	12
2.1.2 Connectors & Pin Definition of CN2	13
2.2 Pin vs. Function Diagram	15
2.2.1 Power/LCD/Backlight.....	15
2.2.2 In-System Programming	15
2.2.3 RS-232	16
2.2.4 I ² C-EEPROM	16
2.2.5 RS-422	17
2.2.6 RS-485	17
Chapter 3 MCU port mapping	18
3.1 MCU Pin Configuration	19
3.2 MCU Port Mapping	20
3.2.1 LCD Controller.....	20
3.2.2 RS-232/RS-422/RS-485	20
3.2.3 Backlight PWM.....	20
3.2.4 I ² C-EEPROM	20
3.2.5 General Purpose I/O	20
Chapter 4 Software Development Tool & Utility.....	21
4.1 Install CCSC 4.093	22
4.2 Install MPLAB ver 8.36.....	27
4.3 Install Bolymin Demo Code and library (SDK)	34
4.4 Compile source code using MPLAB IDE ver 8.36	36
4.5 Programming HEX code using ICD3	42
4.6 Running Hyper Terminal	54

Chapter 5 Software Reference Manual	68
5.1 Setup development environment	69
5.2 Introduction of library	71
5.2.1 Summary of modules of library	71
5.2.2 How to use library in application program	72
5.3 Function Primitives	74
5.3 Function Primitives	75
5.3.1 UART function	75
5.3.2 I ² C eeprom function	77
5.3.3 LCD control function	78
5.3.4 Backlight PWM control function	79
Appendix A - Simple operation guide of BTImg2LCD program	80
Appendix B - Important Notice	82
Appendix C - Notes when running under Windows 7	83

Precaution

FCC



This device is designed to meet the requirement in part 15 of the FCC rules. Operation is subject to conditions ruled under FCC part 15.

Please check packing content upon receiving BEGV606M parcel, make sure that all materials and options are packed inside parcel according to your order.

Packing Contents Check-List

- ☐ BEGV606M LCD Embedded module
- ☐ Software Utility Disc
- ☐ ICD3 and ISP Cable (option)
- ☐ MGI02-0\$ ISP converter board (option)



Chapter 1 Introduction

Abstract

This chapter is to offer you basic information regarding BEGV606M, to help you incorporate BEGV606M into your system.

Contents include:

- 1-1 Features
- 1-2 Board Layout
- 1-3 Block Diagram
- 1-4 Mechanical Dimension
- 1-5 Board Specifications
- 1-6 Ordering information

1.1 Features

BEGV606M is designed based on PIC24FJ64GA002 microprocessor and ST7036i LCD controller, which requires no operating system to run on. Together with a 20x2 character monochrome STN LCD with white LED backlight built-in, this all-in-one LCD embedded system BEGV606M help designer enhance a compact design with cost saving, space saving, and design phase saving.

Armed with RS232, RS422/485, I²C-EEPROM and 3 GPIOs, BEGV606M communicates many devices and peripherals. The BEGV606M is therefore suitable to sit as a industrial control panel for factory automation equipment, electronics instrument, HMI (human-machine interface), office automation equipment, medical equipment, parking system, ticketing system.. and so on.

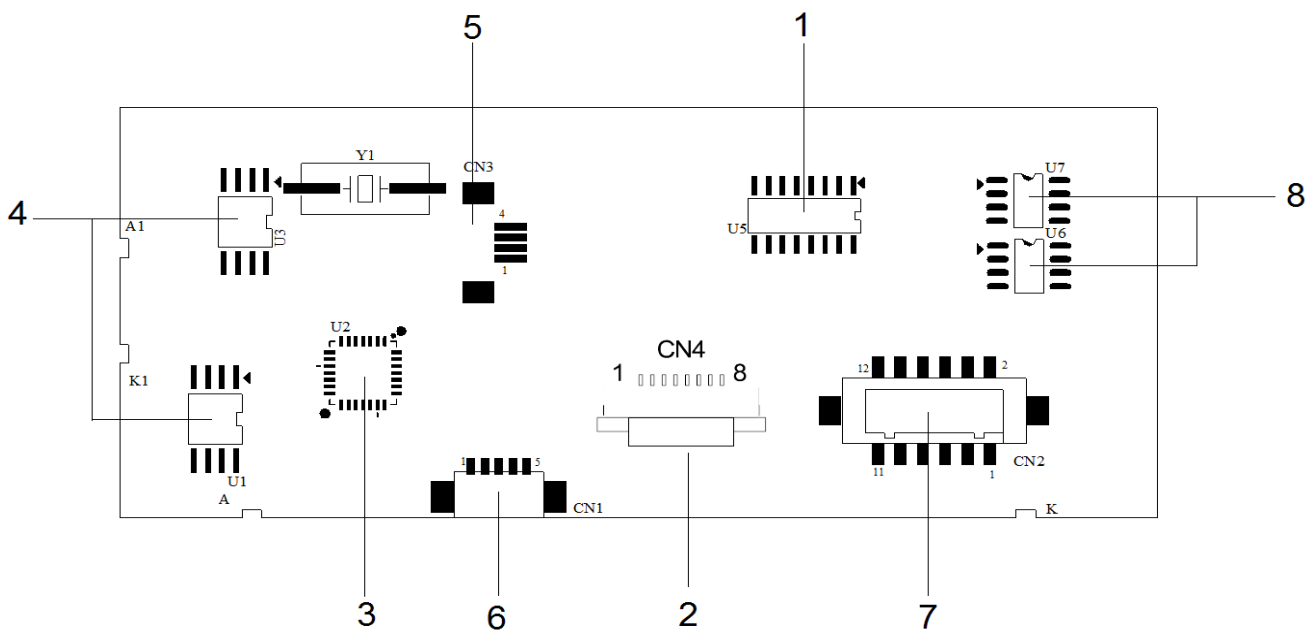
64KB in-system programmable(ISP) Flash, additional 64KB extendable, offers sufficient ROM size for designer to develop software and operate i2c-controlled character LCD and 3 PIOs(IOA,IOB,IOC)

BEGV606M is more than simply a Microchip development board, furthermore, it integrates display and I/O so that developers may start her application without the hassle of hardware integration. Henceforth, a quick time to market for customers' innovative product is ensured..

1.2 Board Layout

This layout shows the location of each important IC, connector and jumper. Please refer to chapter 2 for further information on jumper and connector.

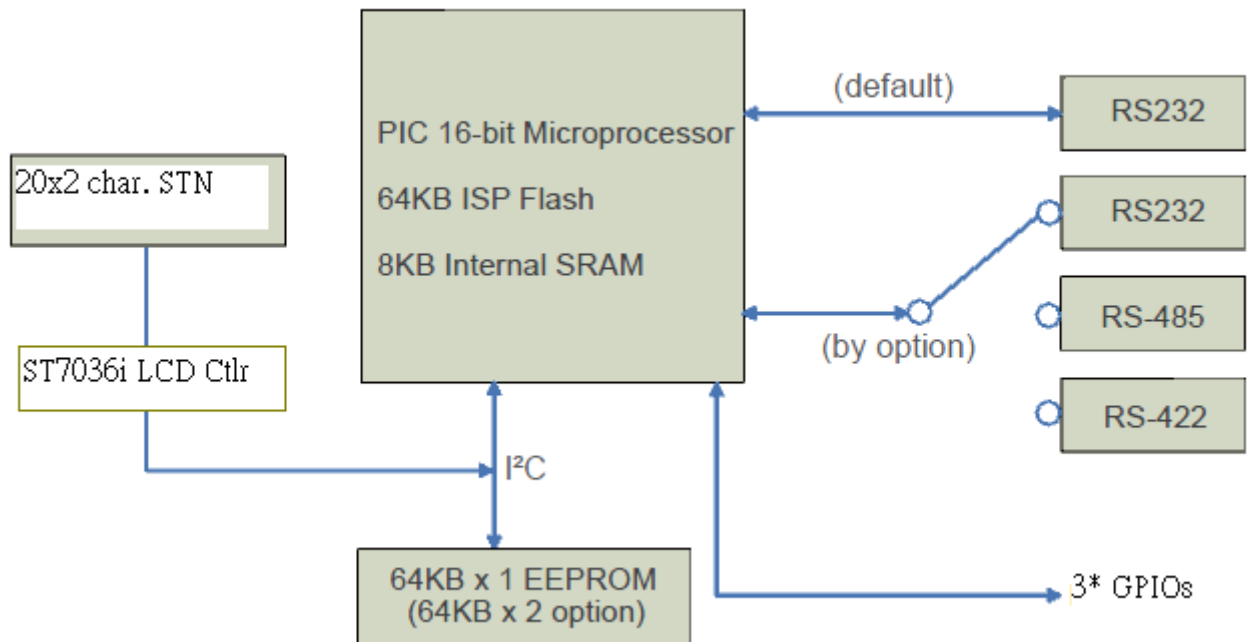
(Drawing 1.2)



- | | |
|---------------------------|---------------------|
| 1 :RS232 | 7 : CN2 |
| 2 :CN4 | 8 :RS422/485 |
| 3 : Microprocessor | |
| 4 :EEPROM | |
| 5 :CN3 | |
| 6 :CN1 | |

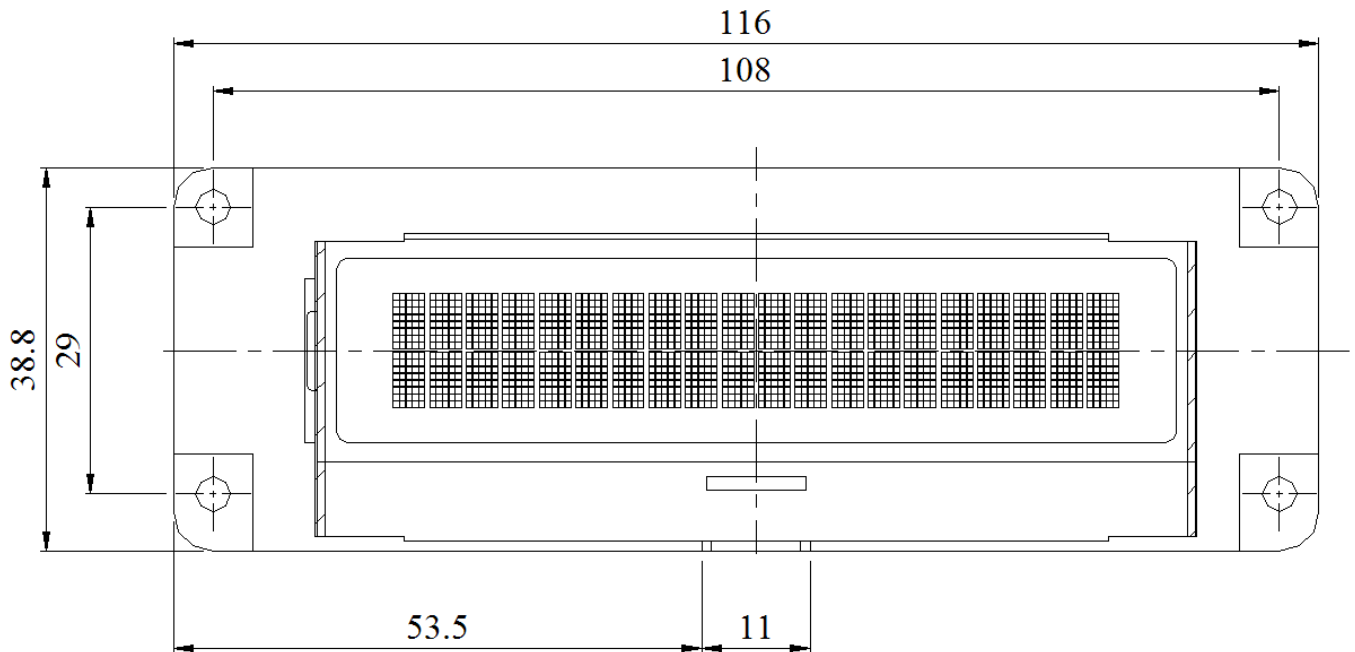
1.3 Block Diagram

(Drawing 1.3)

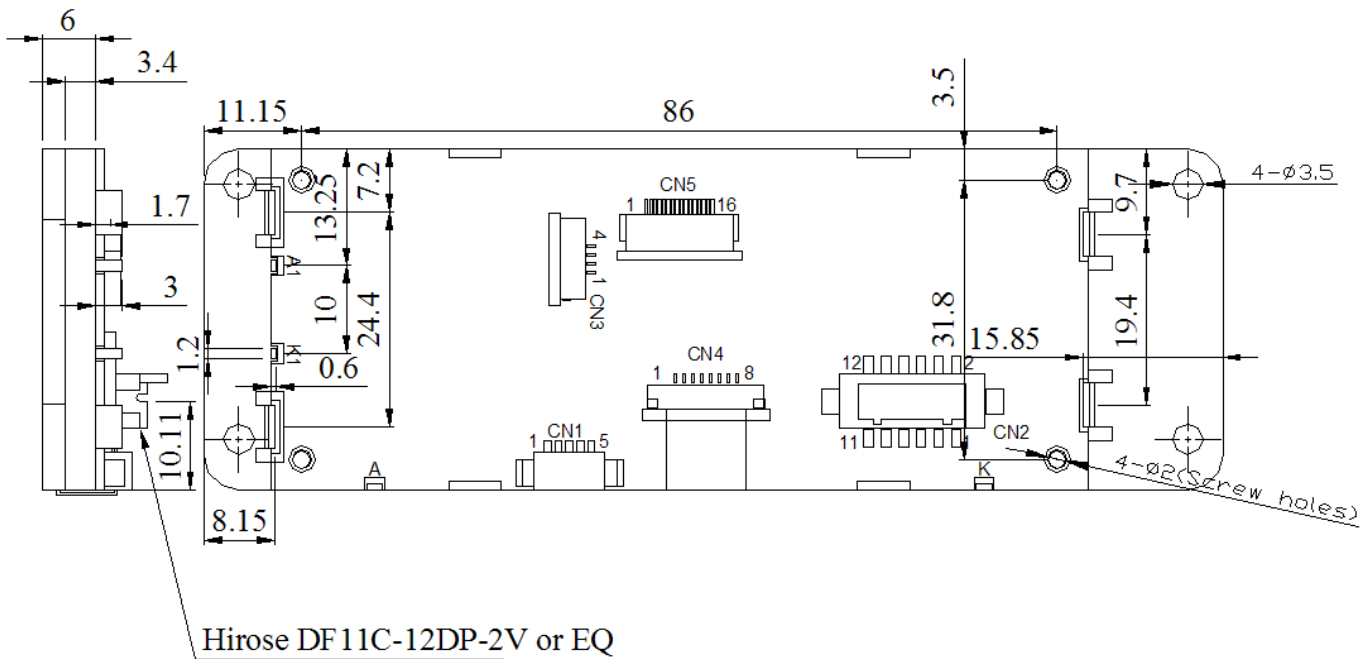


1.4 Mechanical Dimensions

(Drawing 1.4)



(side and back view)



The non-specified tolerance of dimension is $\pm 0.2\text{mm}$.

1.5 Board Specifications

(Table 1.5)

MCU	High-performance, Low-power PIC® 16-bit microprocessor MICROCHIP PIC24FJ64A002
Memory	64K Bytes In-System Programmable Flash 8K Bytes Internal SRAM 64Kx1 Bytes External I2C EEPROM(additional 64KB optional)
Display	Support I2C 20x2 character monochrome STN LCD(edge LED white backlight); use ST7036i for LCD controller.
Serial Ports	Support 1 x RS232 port, and 1 x RS232/RS422/RS485 shared port

1-6 Ordering Information

(Table 1.6)

Part No.	Description	RS232-A	RS232-B	RS422	RS485
BEGV606M	Dual RS232	☆	☆		
BEGV606M1	One RS232	☆			
BEGV606M2	One RS232/One RS422	☆		☆	
BEGV606M3	One RS232/One RS485	☆			☆

Display: FSTN/Gray LCD,LED/White Backlight(Default)

Chapter 2 Installation

Abstract

This chapter is to offer designer fundamental information of BEGV606M jumpers and connectors, in order to help designer configure correct setting and connection between BEGV606M and system application.

2.1 Connectors

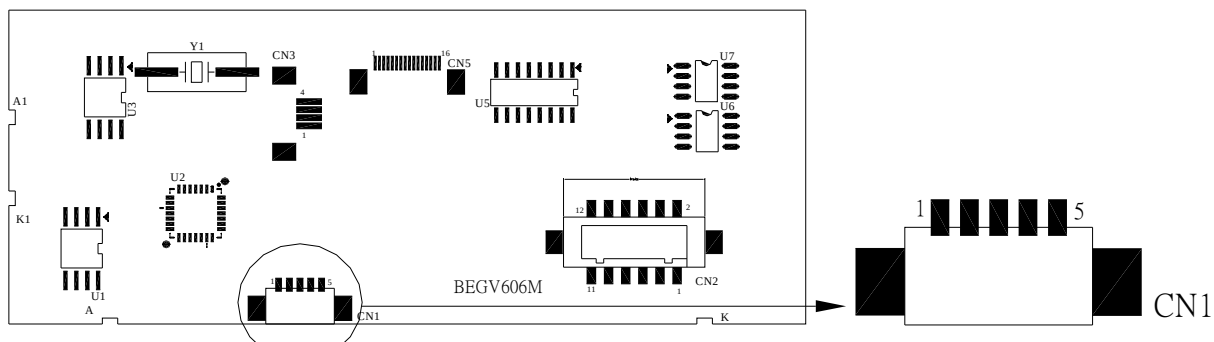
Connectors are the key link between BEGV606M and external devices. Detail locations and functions of available connectors are tabled and illustrated below.

Connectors: (Table 2.1)

Label	Pin No.	Function	Item No.
CN1	5	In-System Programming (ISP)	WFI-RT05-10Q
CN2	12	Connector for GPIO, serial IO, /Reset, and Power. Connect CN2 of 606M to J2 of MGI02-0\$ for powering up target board(606M) and for Hyper terminal operation.	DF11C-12DP-2V(57)

2.1.1 Connector CN1 illustration

(Drawing 2.1CN1)



● CN1 Pin Definition (Table 2.1.1a)

Pin No.	Signal	Description
1	/Reset	
2	A	MPLAB controllable voltage – power target device from ICD3 (ranges from 3 to 3.5 volt, see page 48)
3	GND	
4	PGD	Connect PIC24FJ64GA002 port RP0/PGD1,#1
5	PGC	Connect PIC24FJ64GA002 port RP1/PGC1,#2

● ISP Power & Ground (Table 2.1.1b)

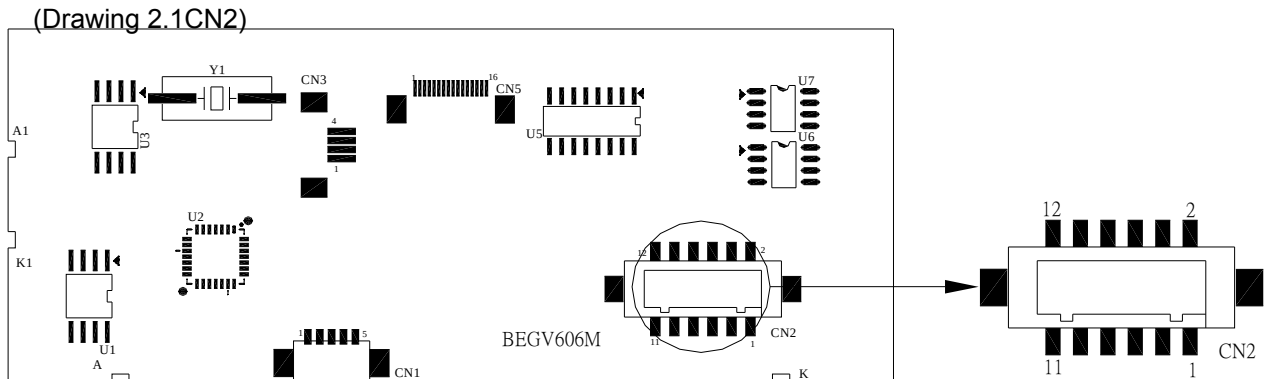
Signal	Type	Pin No.	Description
VCC	P	2	Logic power supply (+3.3V)
GND	P	3	Logic power supply (ground)

● ISP Power & Ground (Table 2.1.1c)

Signal	Type	Pin No.	Description
/Reset	I	1	Auxiliary moment reset for external input
PGD	Bi	4	ISP Programming Data
PGC	Bi	5	ISP Programming Clock

2.1.2 Connectors & Pin Definition of CN2

Note the variance in pin definition by ordering option is as highlighted.



- **CN2 Pin Definition of BEGV606M - Dual RS232 (Table 2.1.2a)**

Signal	Pin No	Pin No.	Signal
GND	2	1	VDD
TX0	4	3	NC
RX0	6	5	NC
IOA	8	7	TX1
IOB	10	9	RX1
IOC	12	11	/RST

- **CN2 Pin Definition of BEGV606M1 - One RS232 (Table 2.1.2b)**

Signal	Pin No	Pin No.	Signal
GND	2	1	VDD
TX0	4	3	NC
RX0	6	5	NC
IOA	8	7	NC
IOB	10	9	NC
IOC	12	11	/RST

- **CN2 Pin Definition of BEGV606M2 - One RS232/One RS422 (Table 2.1.2c)**

Signal	Pin No	Pin No.	Signal
GND	2	1	VDD
TX0	4	3	422RP
RX0	6	5	422RN
IOA	8	7	422TP
IOB	10	9	422TN
IOC	12	11	/RST

- **CN2 Pin Definition of BEGV606M3 - One RS232/One RS485 (Table 2.1.2d)**

Signal	Pin No	Pin No.	Signal
GND	2	1	VDD
TX0	4	3	NC
RX0	6	5	NC
IOA	8	7	485P
IOB	10	9	485N
IOC	12	11	/RST

Here summarize all **CN2** pins applicable on 606M embedded system categorized by signal type.

Pin Types	
I	=Input
O	=Output
Bi	=Input / Output (Bi-Directional)
U	=User defined
P	=Power

● **Power & Ground** (Table 2.1.2e)

Signal	Type	Pin No.	Description
VDD	P	1	Logic power supply (+5V)
GND	P	2	Logic power supply (ground)

● **Serial I/O** (Table 2.1.2f)

Signal	Type	Pin No.	Description
RX0	I	6	Receiver of first RS232 with driver
TX0	O	4	Transmitter of first RS232 with driver
422RP	I	3	no inverting receiver of RS422
422RN	I	5	inverting receiver of RS422
422TP/485P/TX1	Bi	7	When is configured as RS422 it act as no inverting transmitter, When is configured as RS 485 it acts as positive differential IO. When is configured as RS232 Transmitter of second RS232.
422TN/485N/RX1	Bi	9	When is configured as RS422 it acts as inverting transmitter, When is configured as RS 485 it acts as negative differential IO. When is configured as RS232 Receiver of second RS232.

● **General Purpose I/O** (Table 2.1.2g)

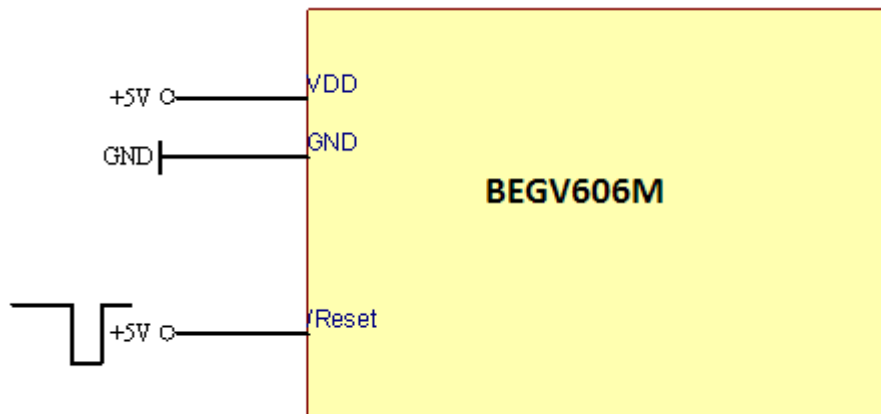
Signal	Type	Pin No.	Description
/Reset	I	11	Auxiliary moment reset for external input
IOA	U	8	I/O port (PIC24FJ64GA002 port RP4,#8)
IOB	U	10	I/O port (PIC24FJ64GA002 port RP7,#13)
IOC	U	12	I/O port (PIC24FJ64GA002 port RP11,#19)

Note that there is a series resistor of 100 ohm on the 3 GPIOs, also with ESD protection circuit. The RS-232 voltage is elevated from CMOS 3.3v to TTL 5 volt level through ICL3232.

2.2 Pin vs. Function Diagram

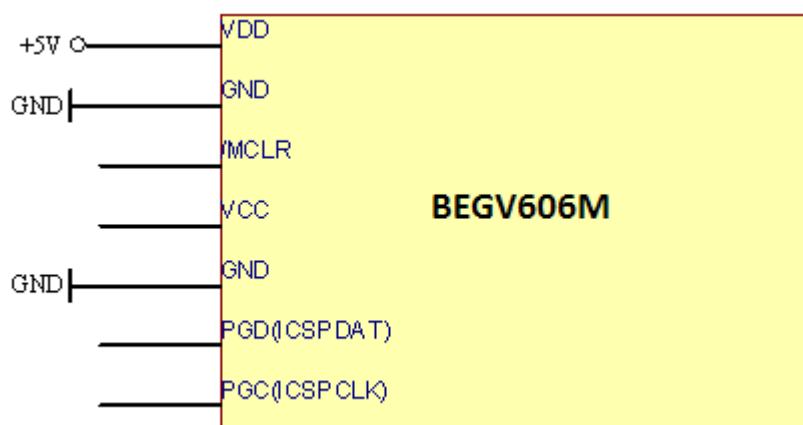
2.2.1 Power/LCD/Backlight

Diagram of system power supply.



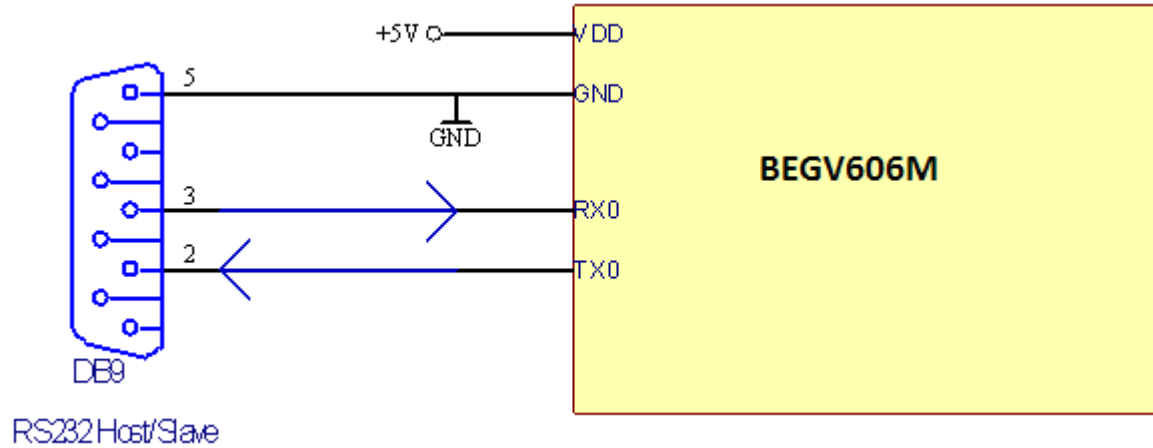
2.2.2 In-System Programming

Customer may contact Bolymin sales for purchase of Microchip ICD3 and proprietary converter board **MG102-\$0**.



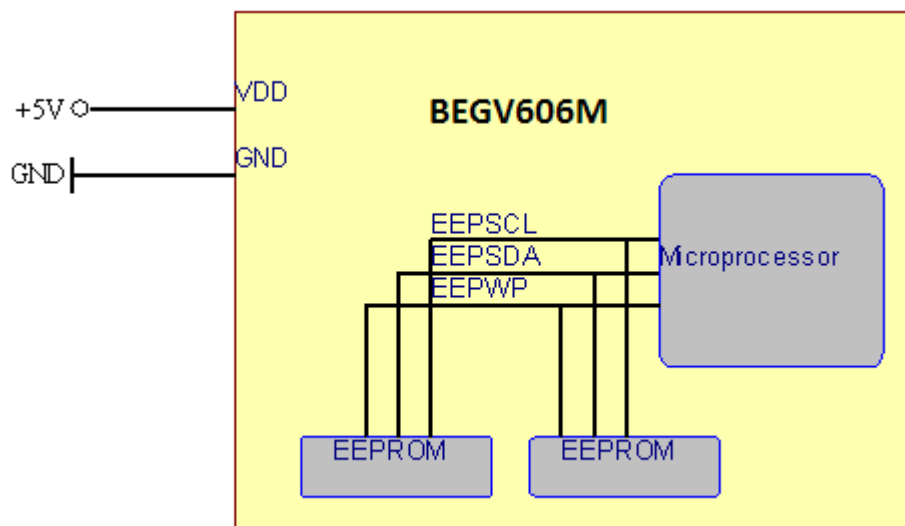
2.2.3 RS-232

BEGV606M offers RS-232 port to connect PC or other RS232 device directly without ICL232.



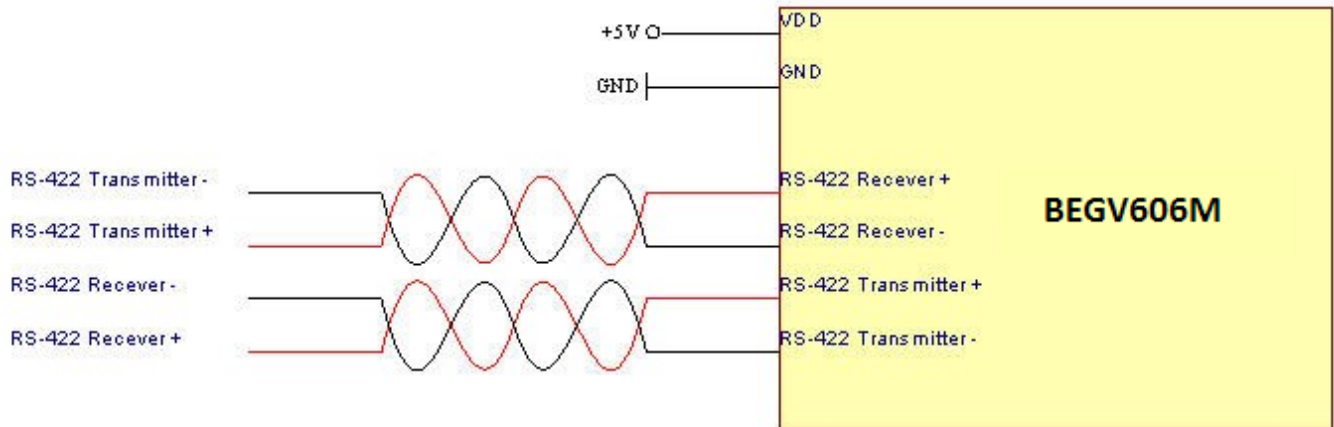
2.2.4 I²C-EEPROM

BEGV606M supports I²C bus. to control 64Kbytes x 2 in-system EEPROM.



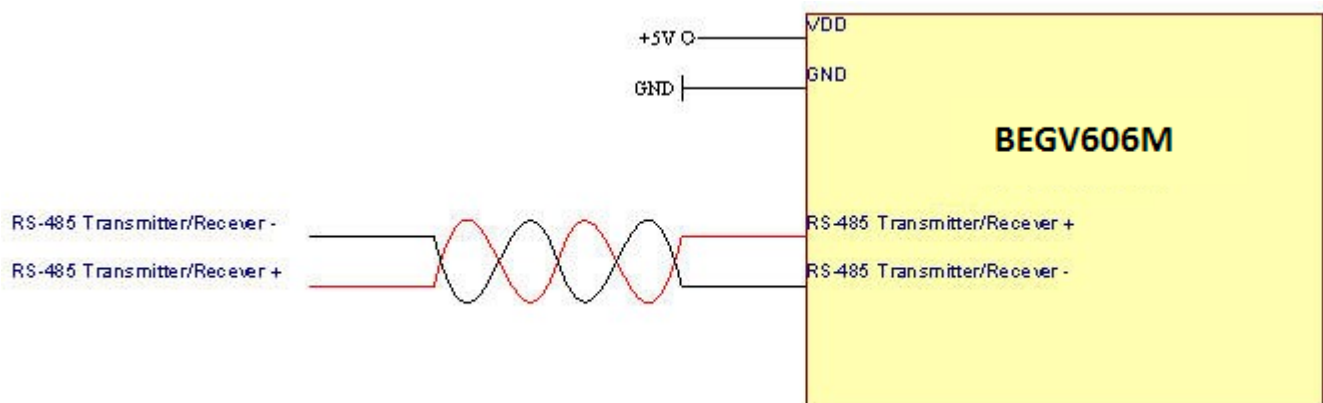
2.2.5 RS-422

BEGV606A offers 1 x RS-422(isolated) port.



2.2.6 RS-485

BEGV606A offers 1 x RS-485(isolated) port.



Chapter 3 MCU port mapping

Abstract

This chapter explains PIC24F MCU pin configuration and port mapping toward key elements such as LCD, Touch Panel, RS-232, RS-422, RS-485, LED Backlight, I²C-EEPROM, 2-wire serial port, and 3 general purpose I/O.

3.1 MCU Pin Configuration

(Drawing 3.1, PIC24FJ64GA002 MCU)



3.2 MCU Port Mapping

3.2.1 LCD Controller

(Table 3.2a)

MCU PIC24FJ64GA002	LCD Controller
RP2/SDA2, #3	LSDA
RP3/SCL2, #4	LSCL
TDO/SDA1/RP9, #15	LCD RST

3.2.2 RS-232/RS-422/RS-485

(Table 3.2b)

MCU PIC24FJ64GA002	RS-232/422/485
RP14, #22	RX0 (CN2#6)
RP15, #23	TX0 (CN2#4)
RP6/PGC3, #12	422TN/485N/RX1 (CN2#9)
RP5/PGD3, #11	422TP/485N/TX1 (CN2#7)

(Table 3.2c)

MCU PIC24FJ64GA002	RS-485
RA4/T1CK, #9	Enable RS-485

3.2.3 Backlight PWM

(Table 3.2d)

MCU PIC24FJ64GA002	LED backlight
RP10, #18	Backlight PWM

3.2.4 I²C-EEPROM

(Table 3.2e)

MCU PIC24FJ64GA002	EEPROM/ I ² C
RP2/SDA2, #3	EEPROM SDA
RP3/SCL2, #4	EEPROM SCL
RP8/TCK, #14	EEPROM WP

3.2.5 General Purpose I/O

(Table 3.2f)

MCU PIC24FJ64GA002	General Purpose I/O
RP4, #8	IOA(CN2#8)
RP7, #13	IOB(CN2#10)
RP11, #19	IOC(CN2#12)

* GPIO pins are pulled high with 3.3K ohm resistor.

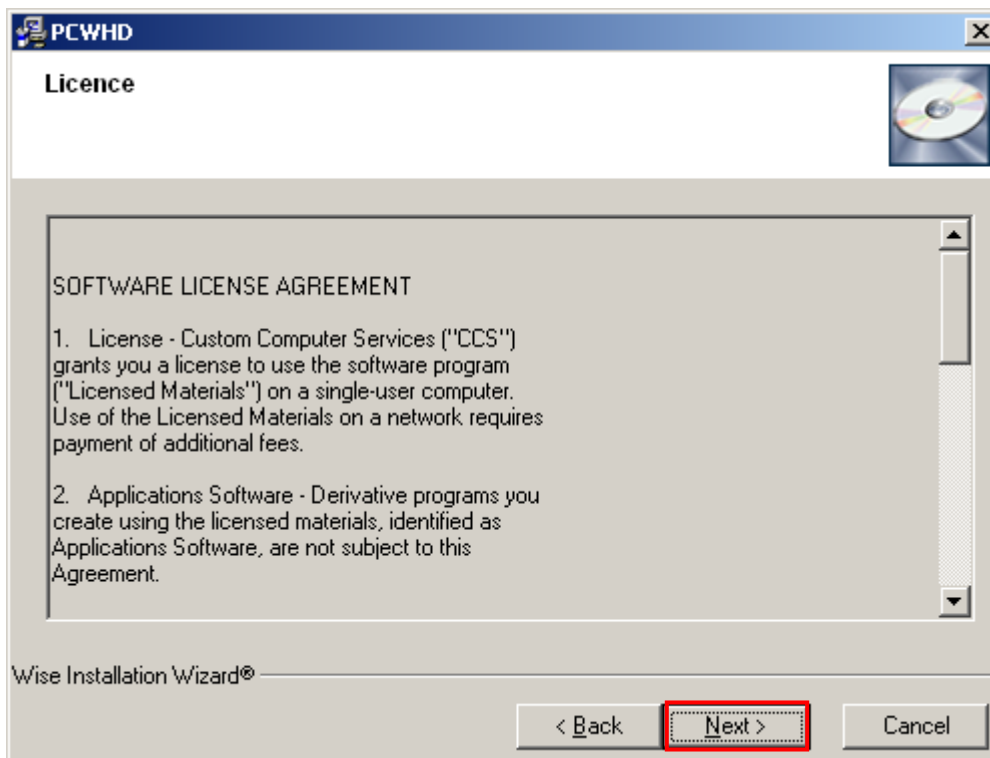
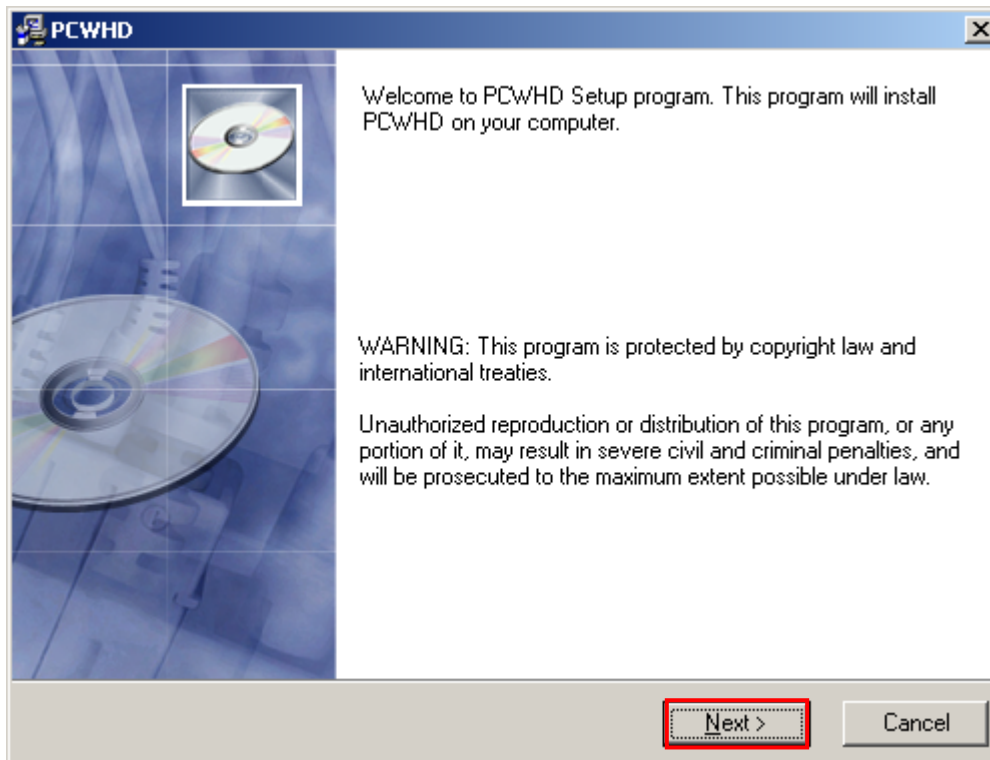
Chapter 4 Software Development Tool & Utility

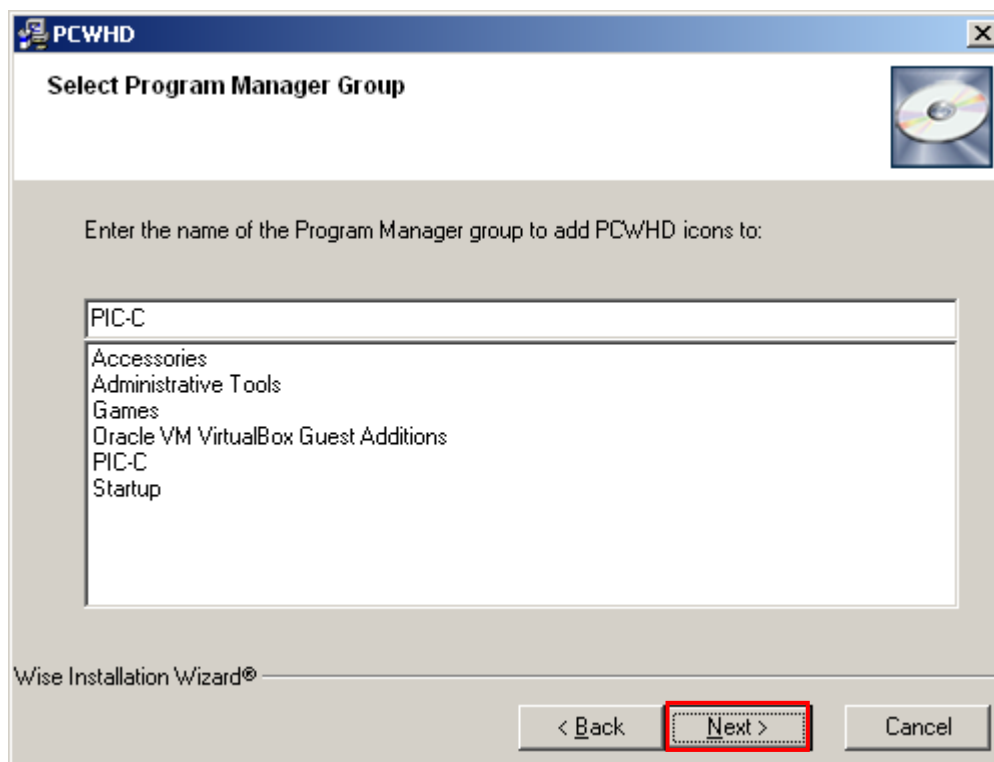
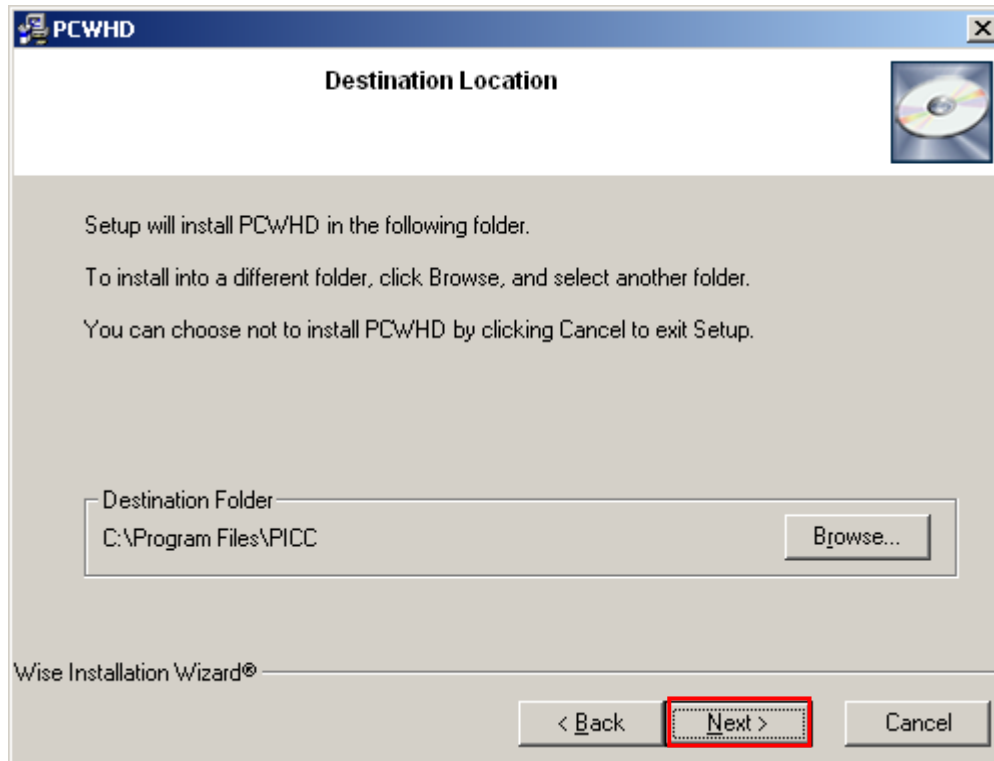
Abstract

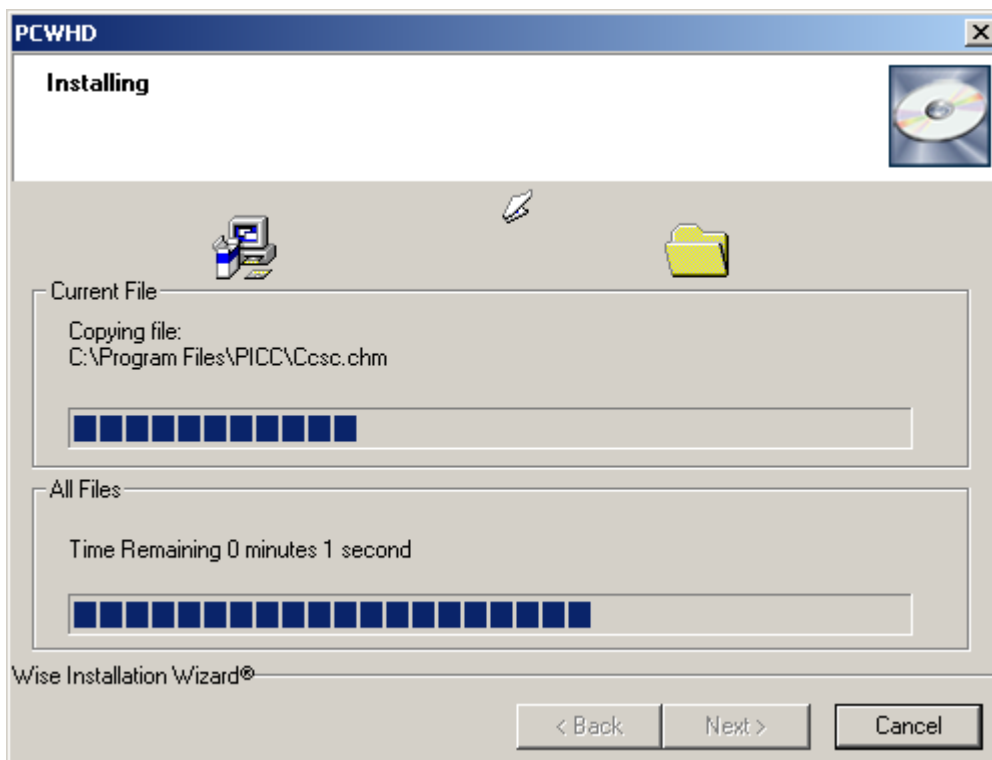
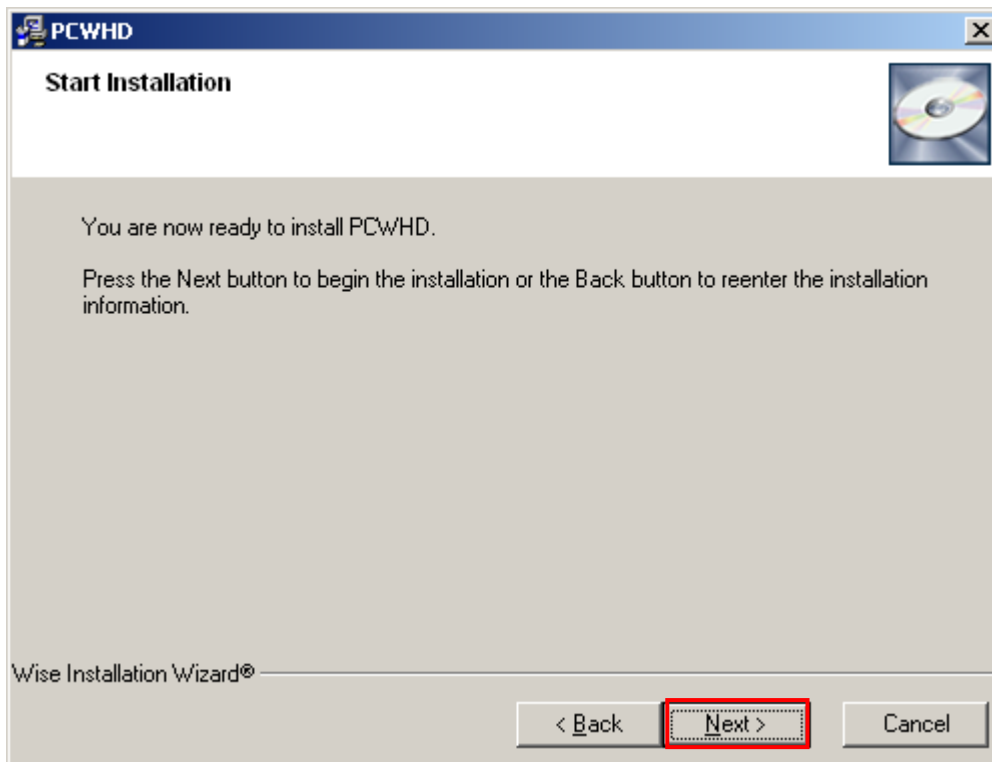
This chapter explains PIC24FJ64GA002 MCU software development tool Microchip ICD3 and Bolymin free software utilities.

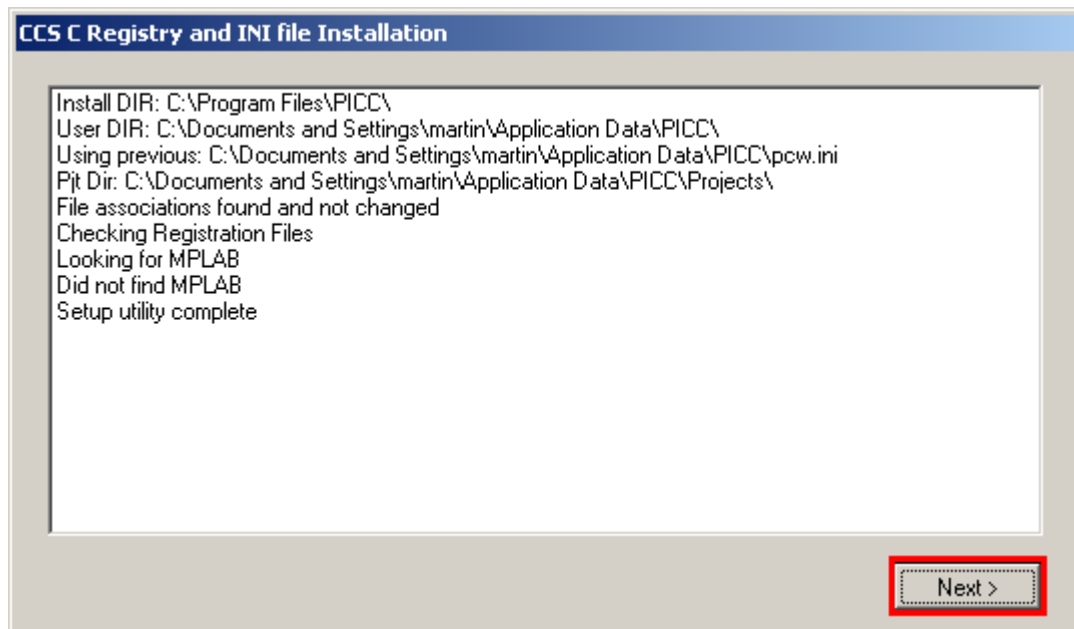
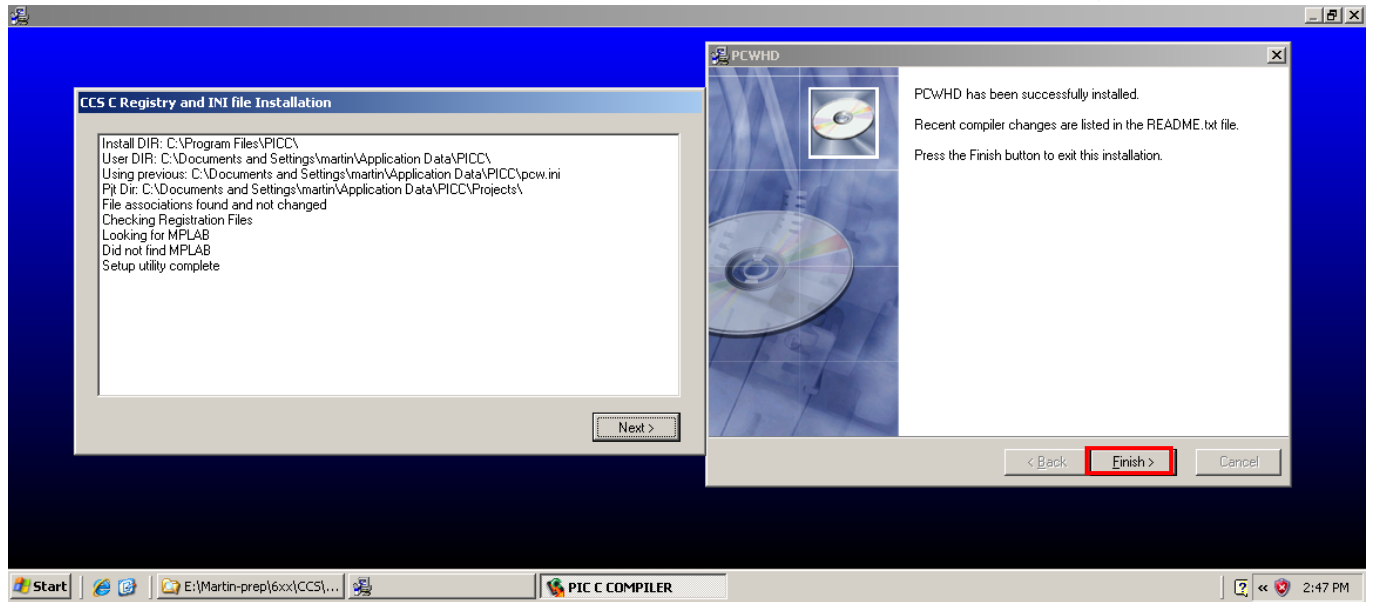
Please follow the step guide to set up Microchip PIC24FJ64GA002 Software Development Tools.

4.1 Install CCSC 4.093



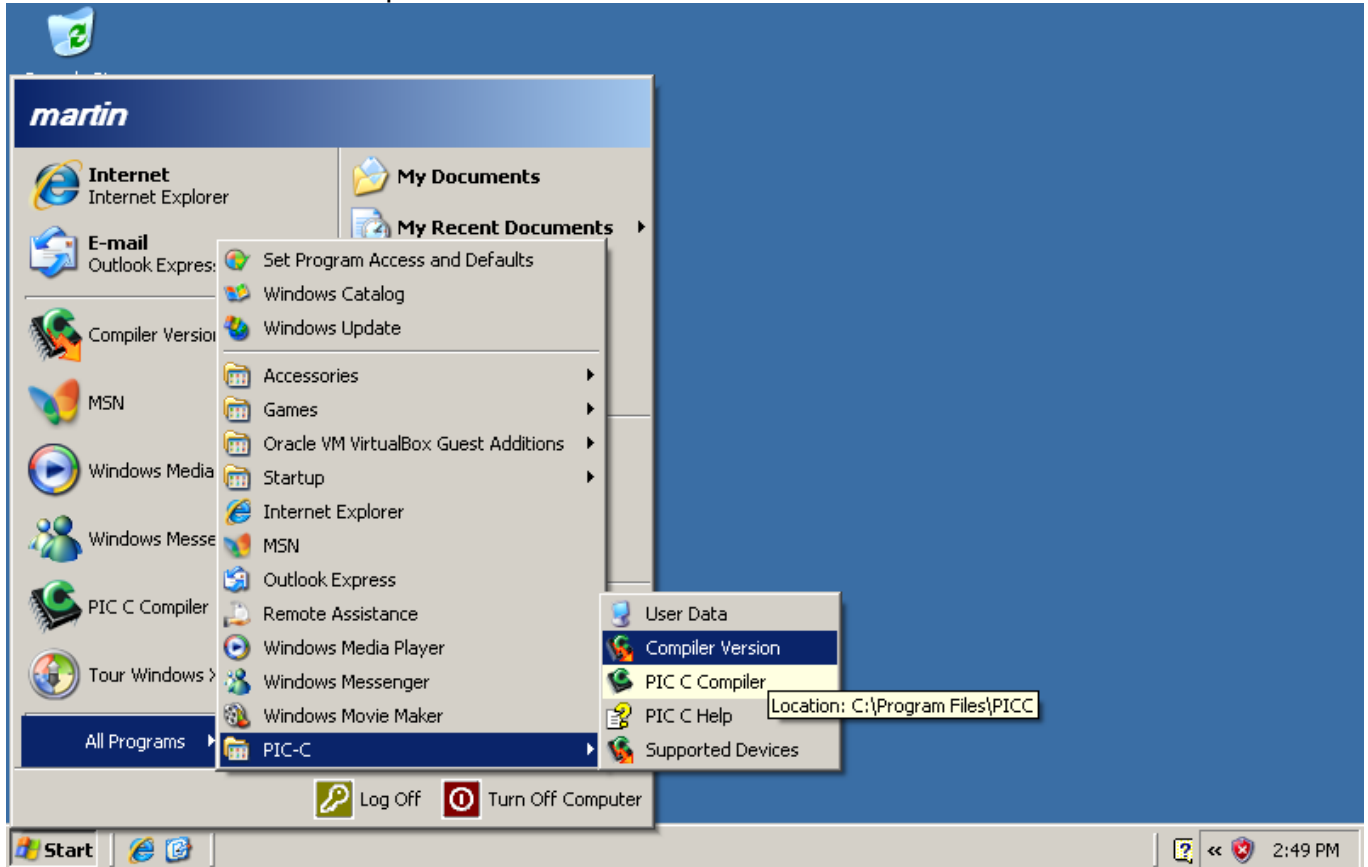




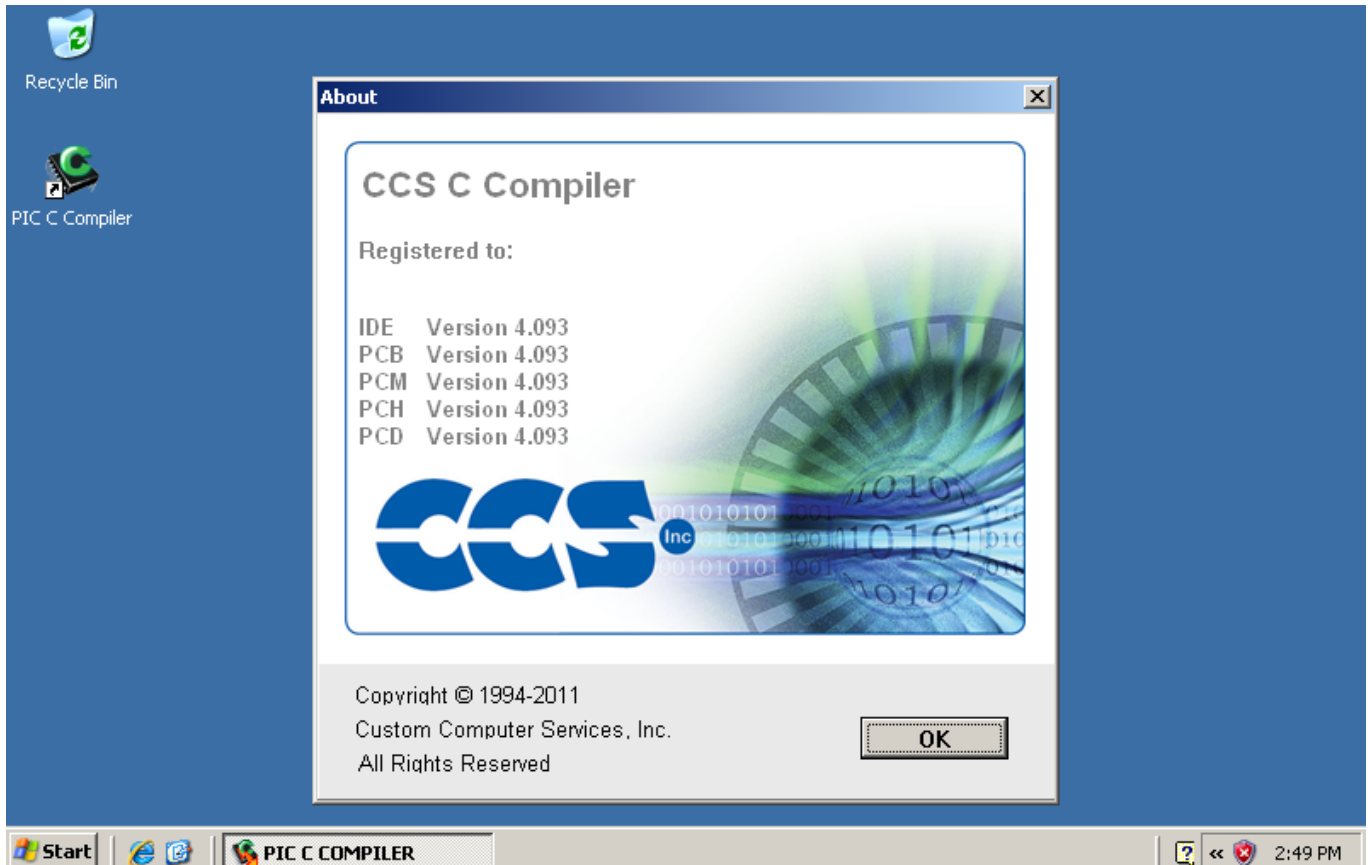


Click **Next** to complete installation of CCSC 4.093 .

Now check the CCSC compiler version



CCSC version is 4.093.



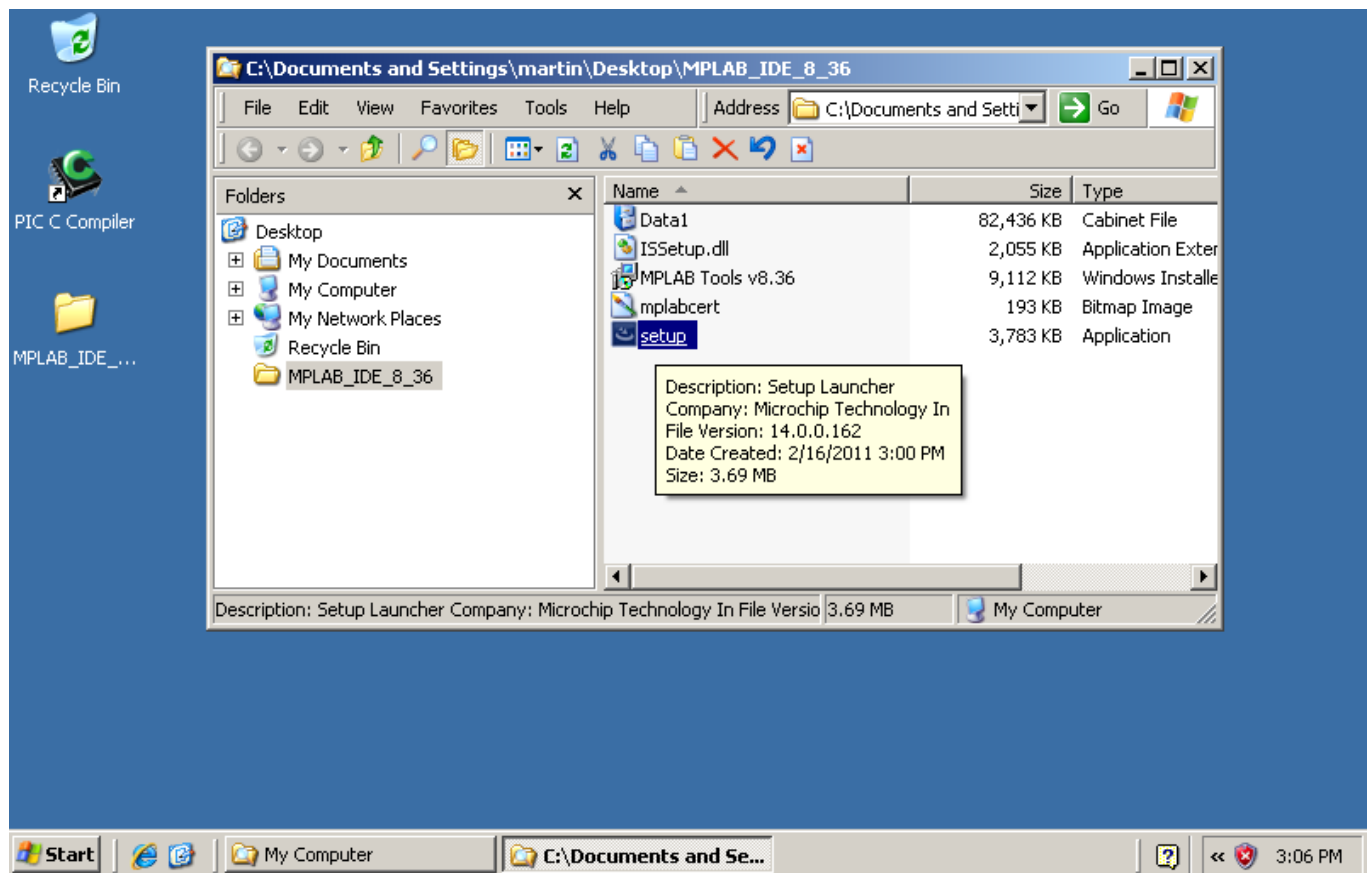
4.2 Install MPLAB ver 8.36

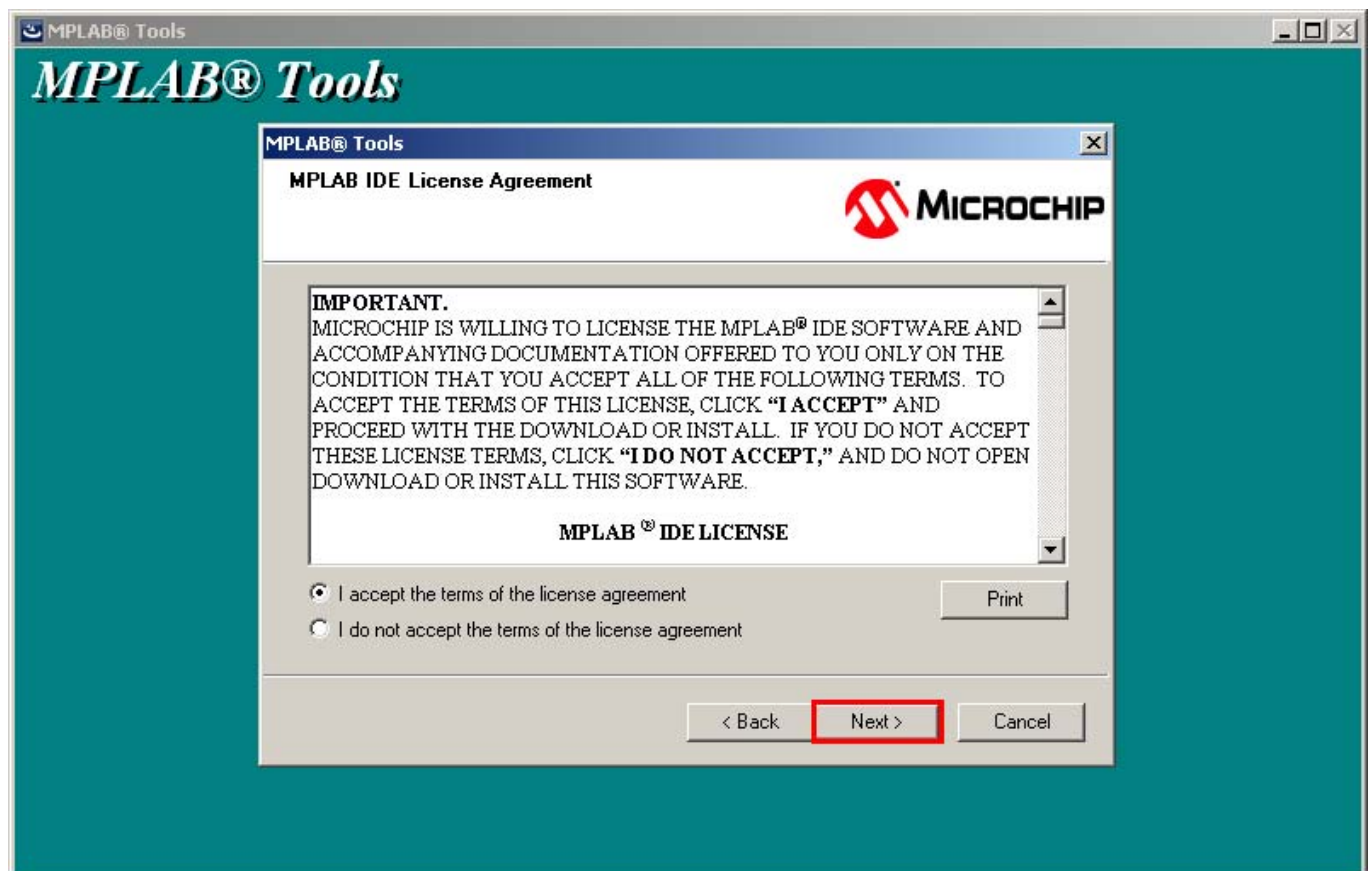
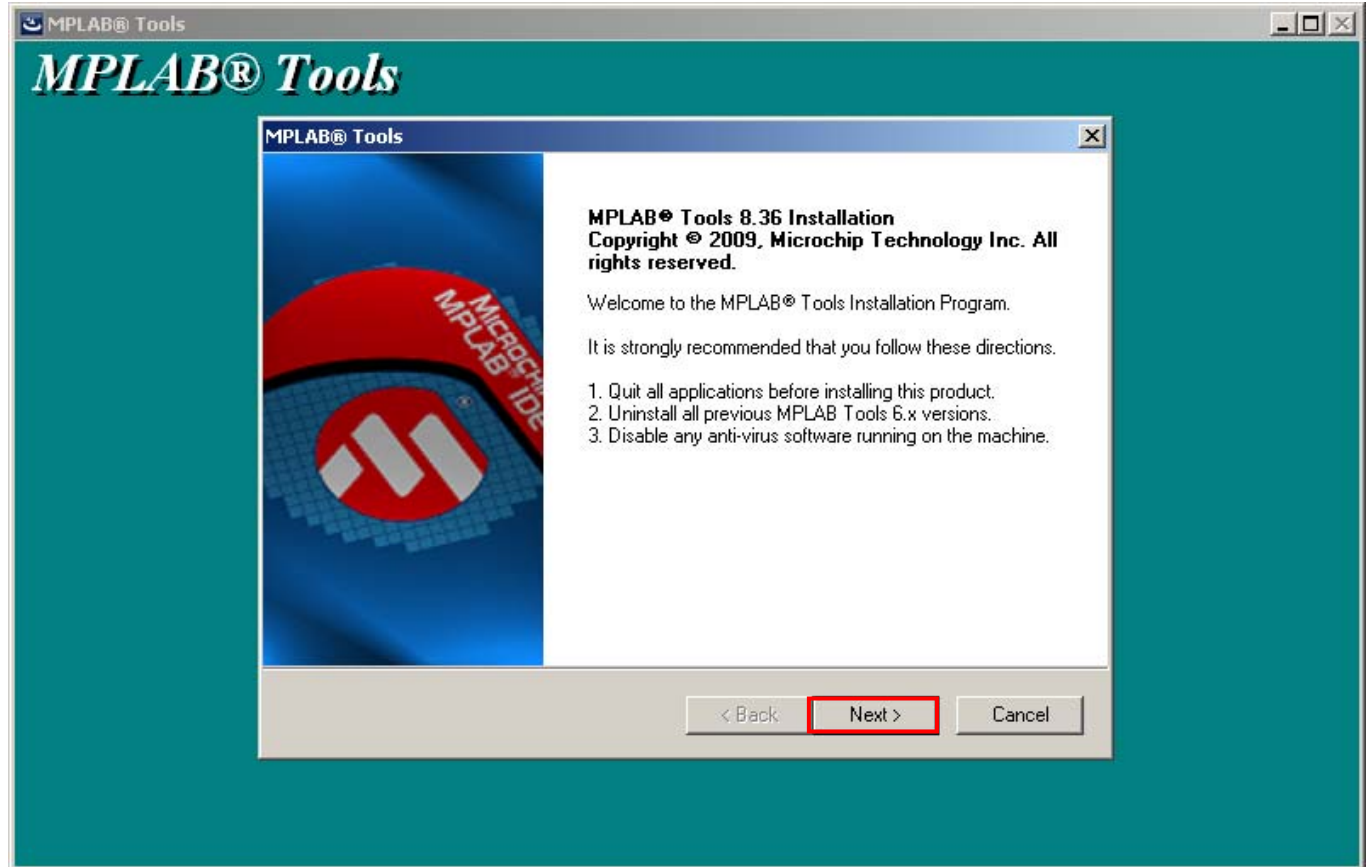
Designers may download software development tool MPLAB ver 8.36 from the following URL:

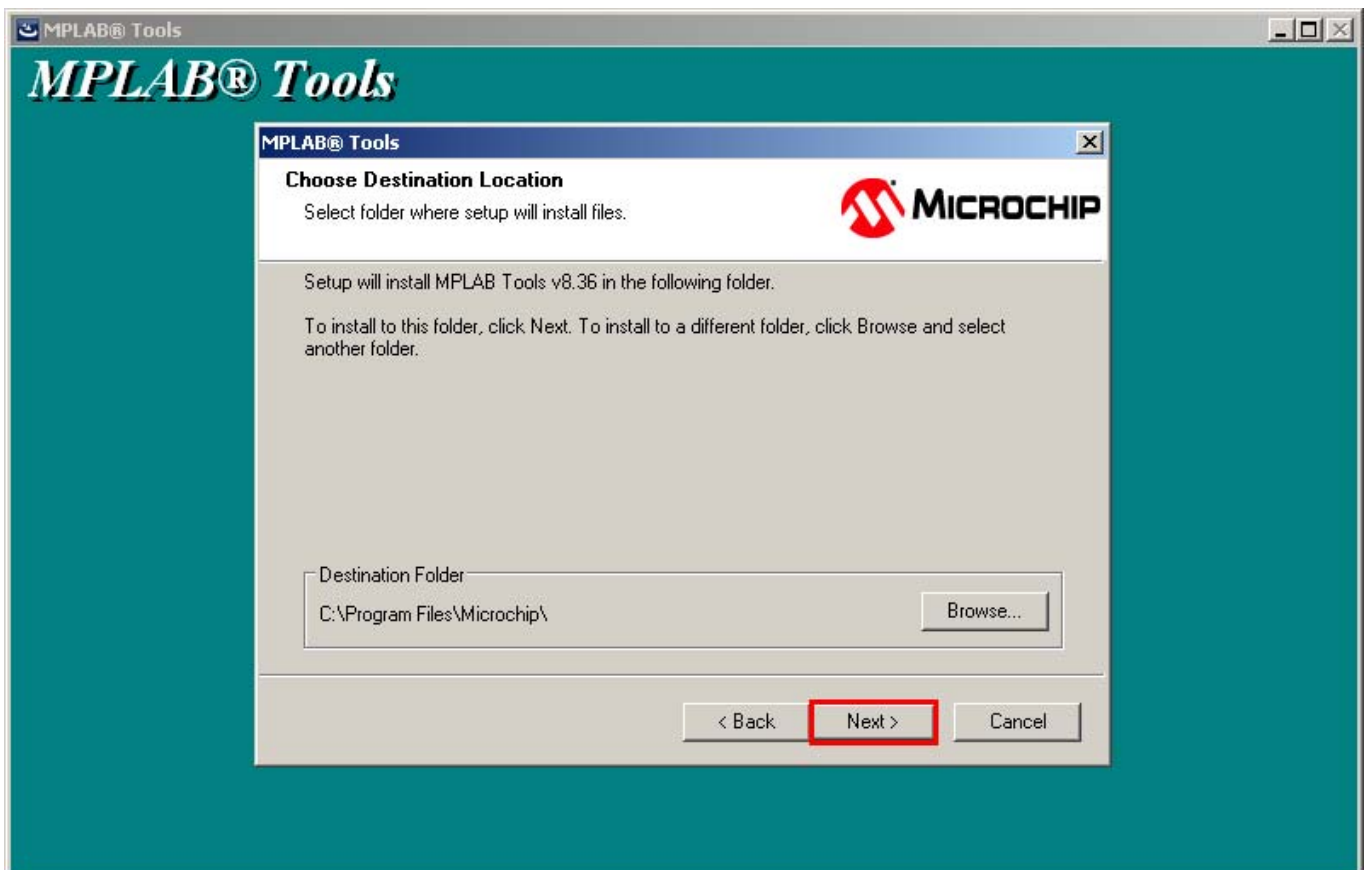
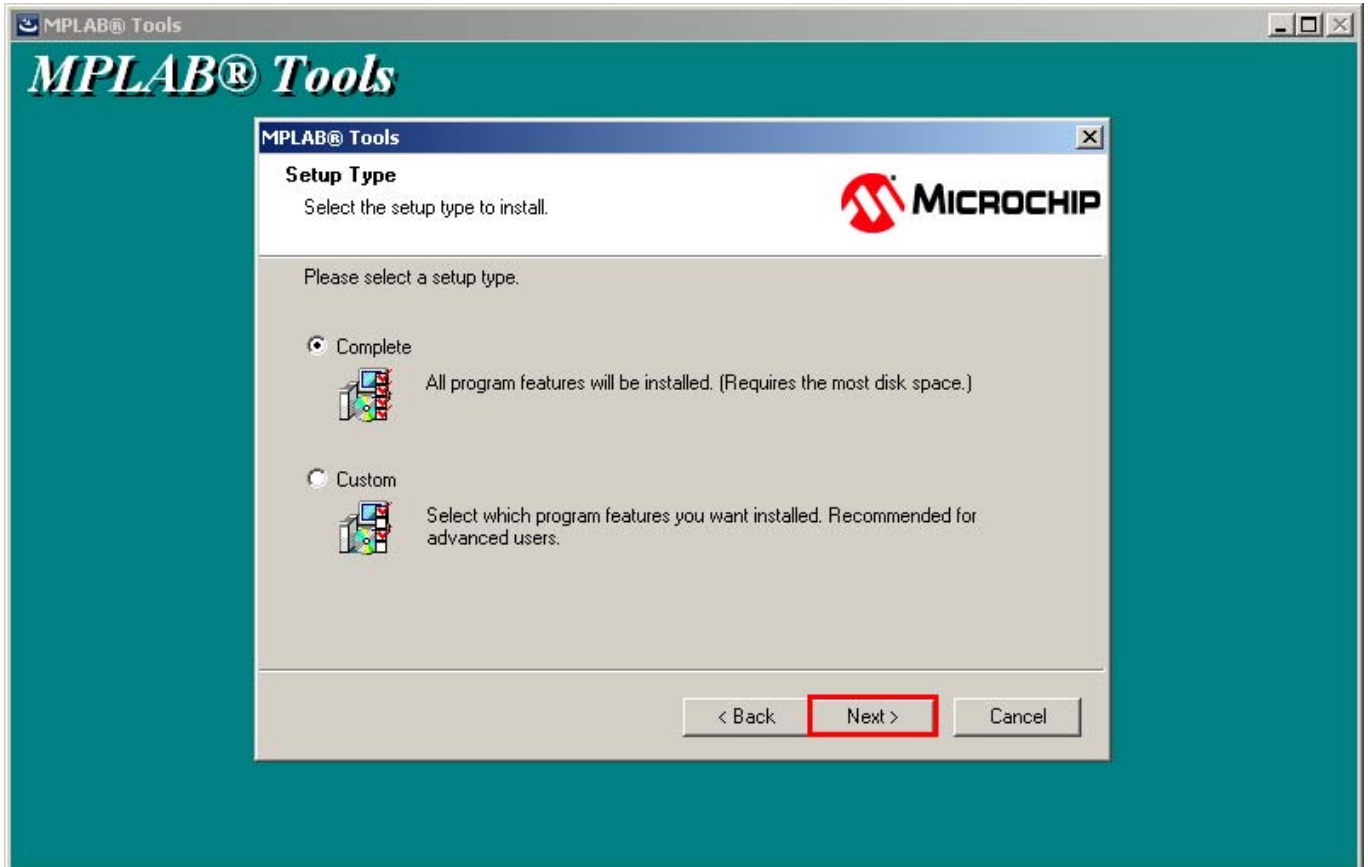
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&docName=en023073

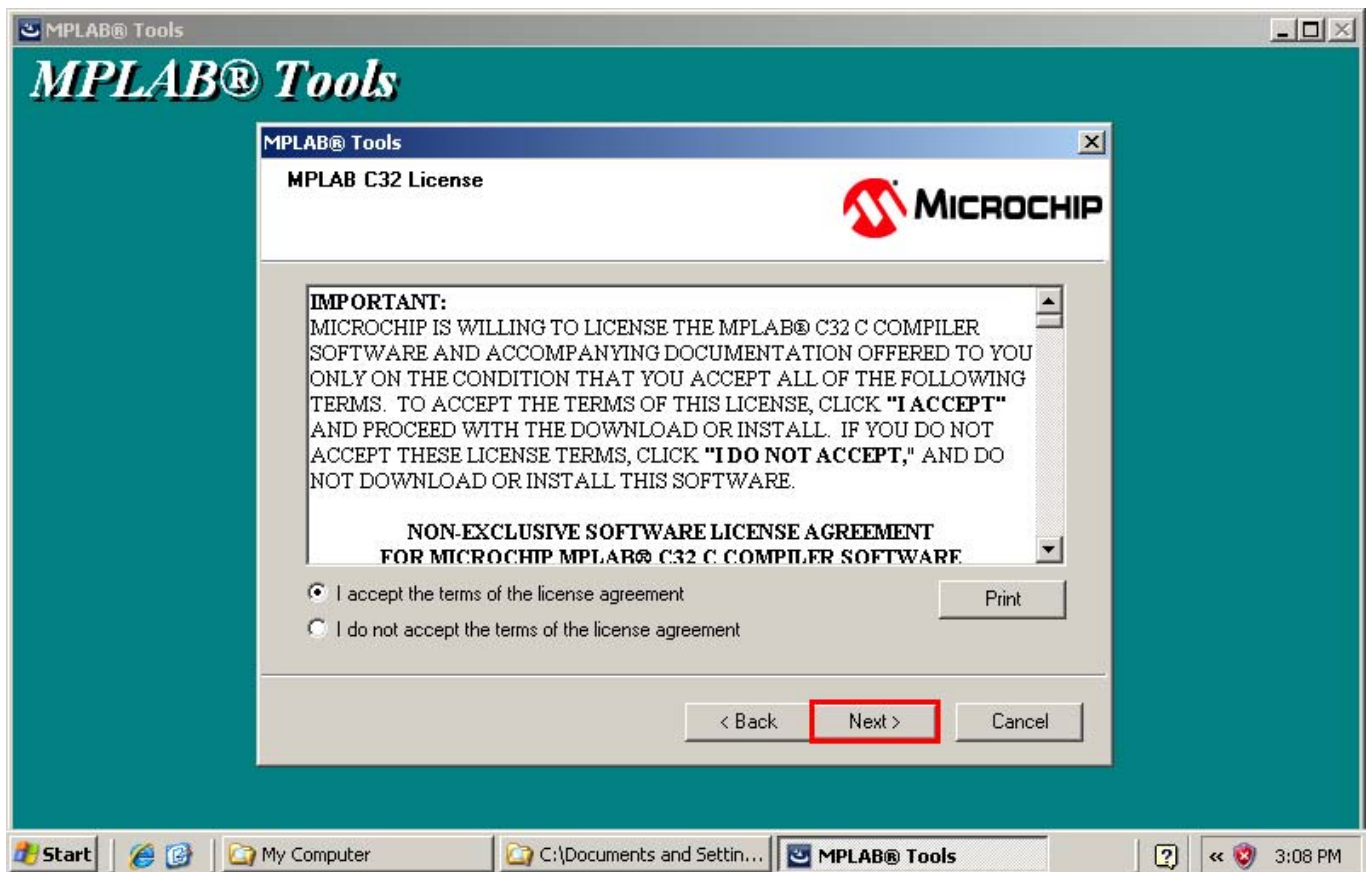
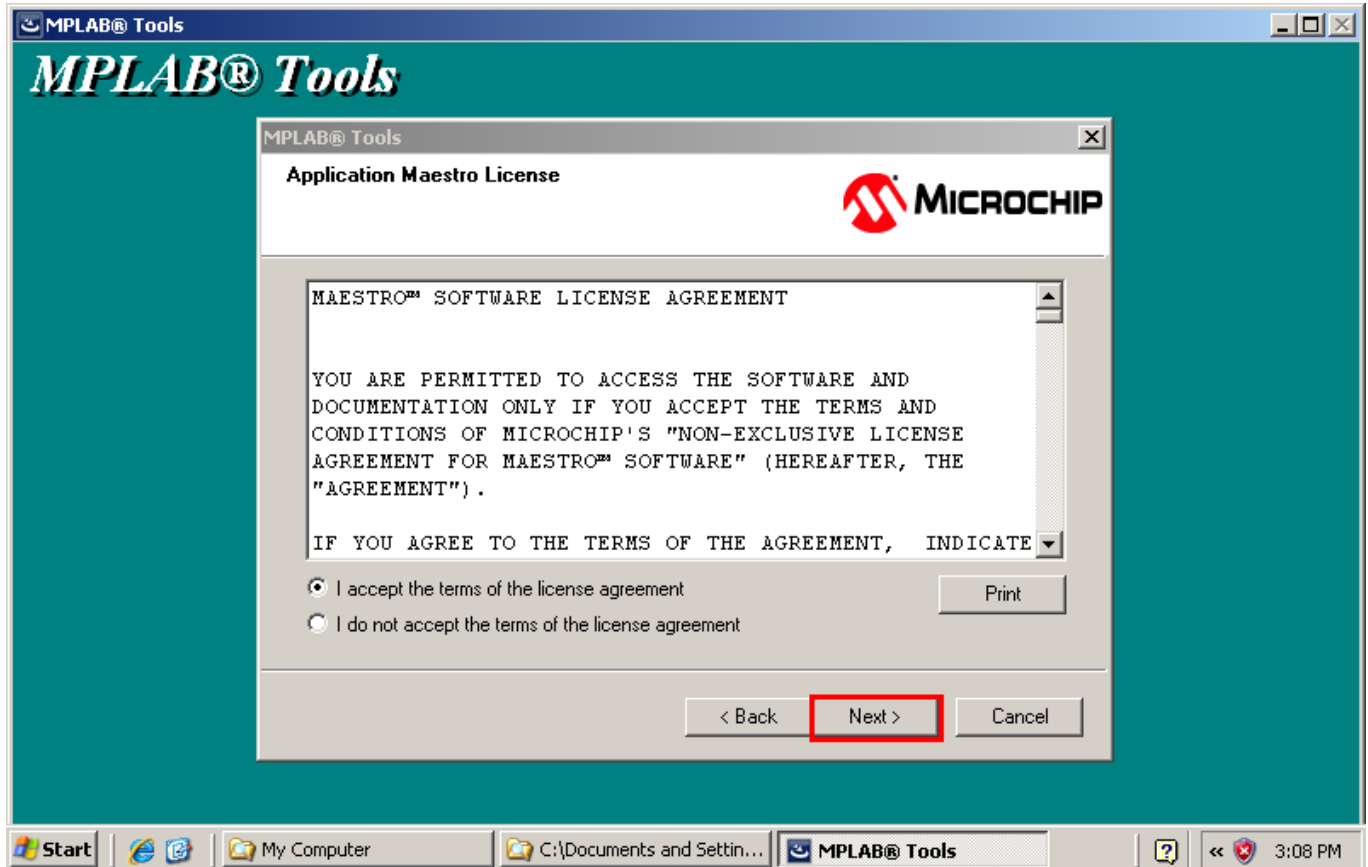
http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_IDE_8_36.zip

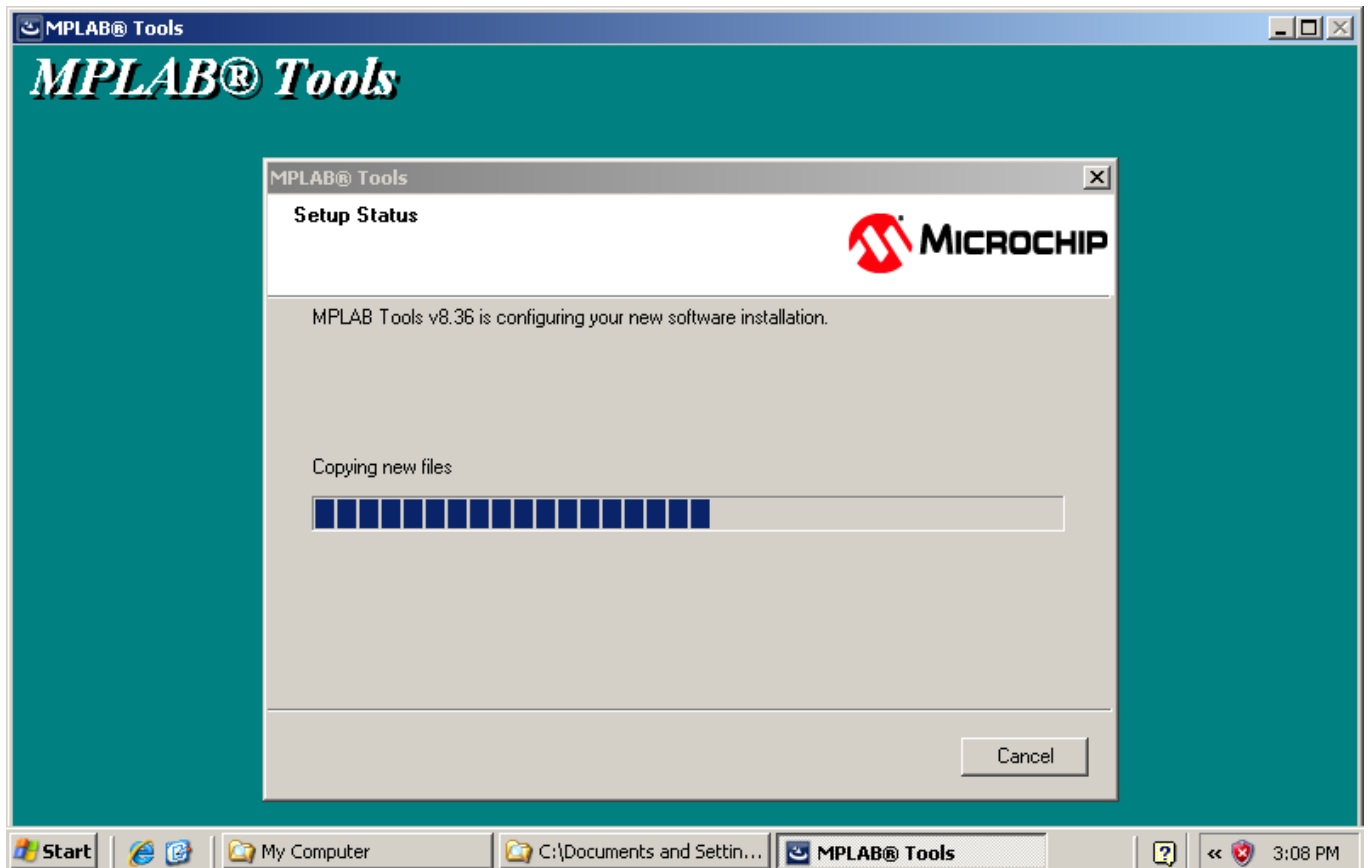
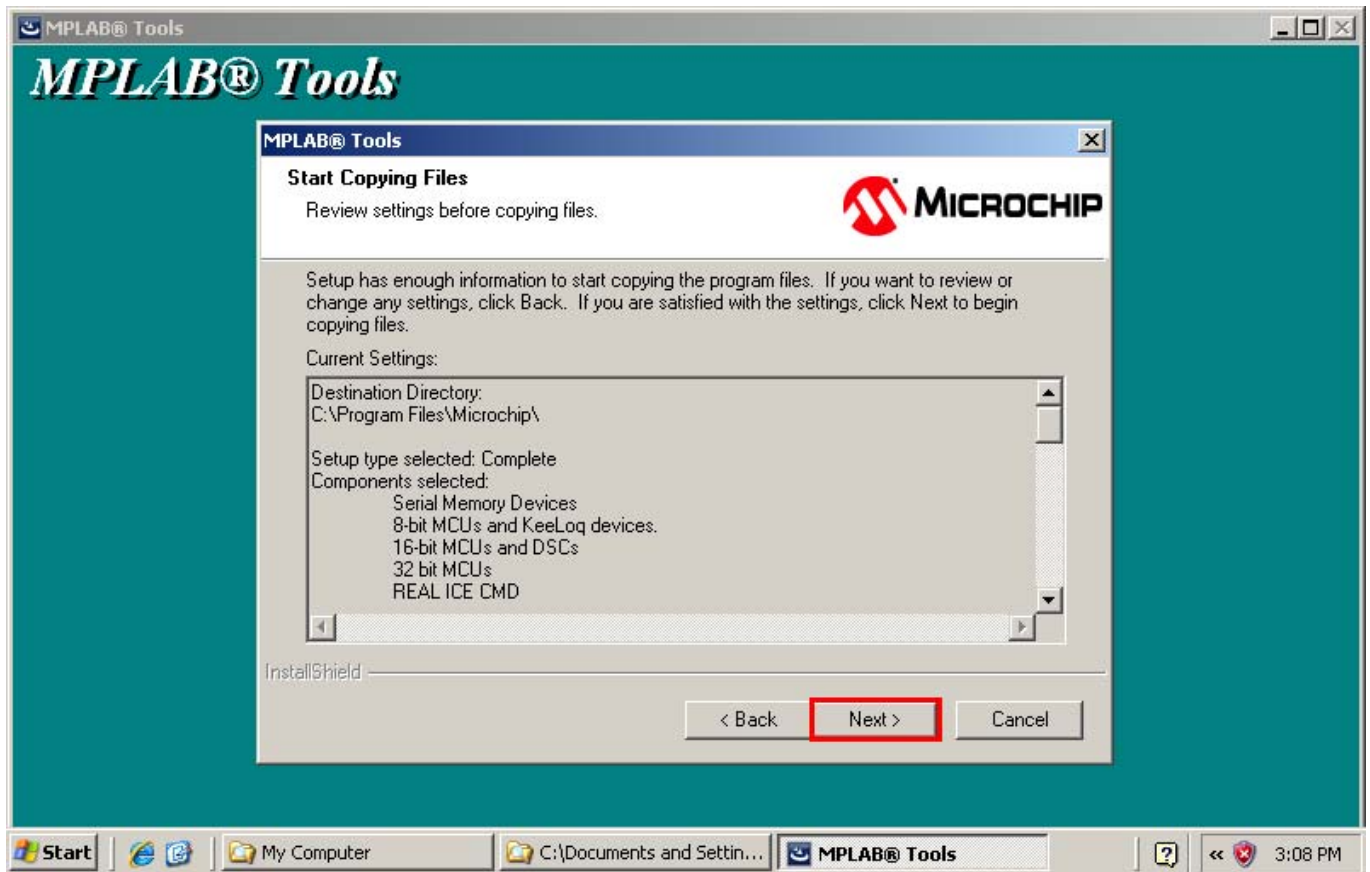
or from BOLYMIN utility disk.

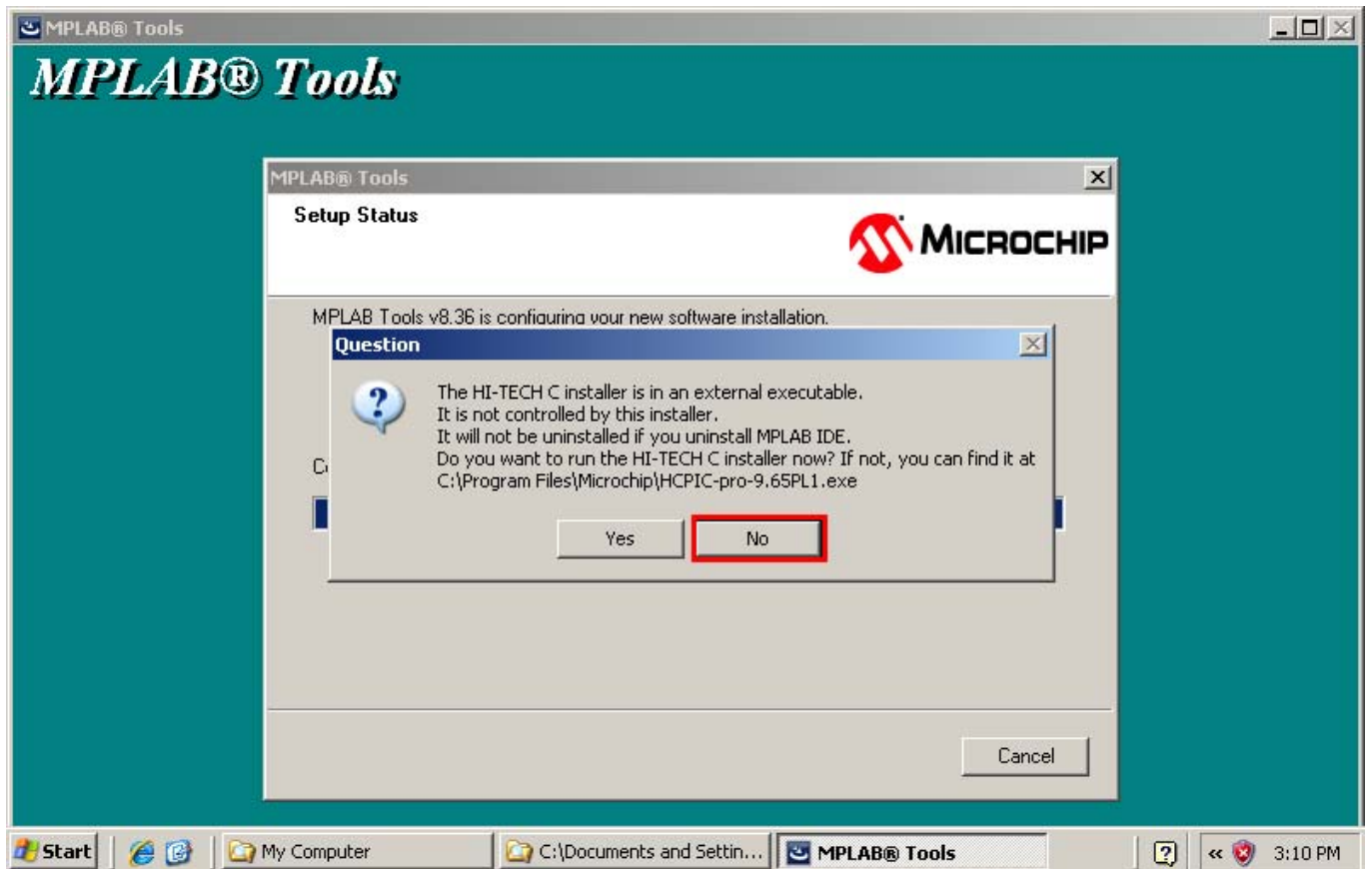




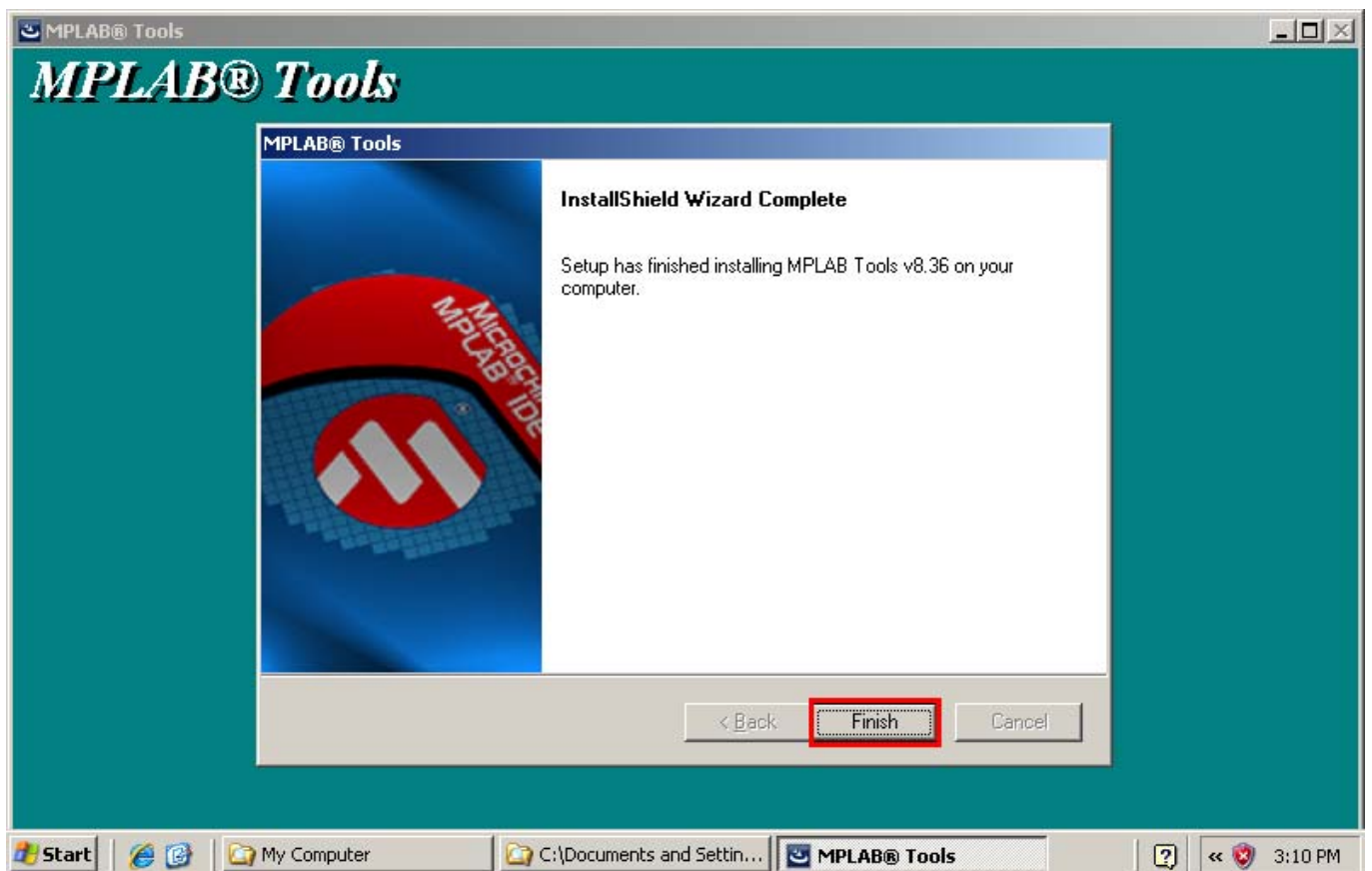


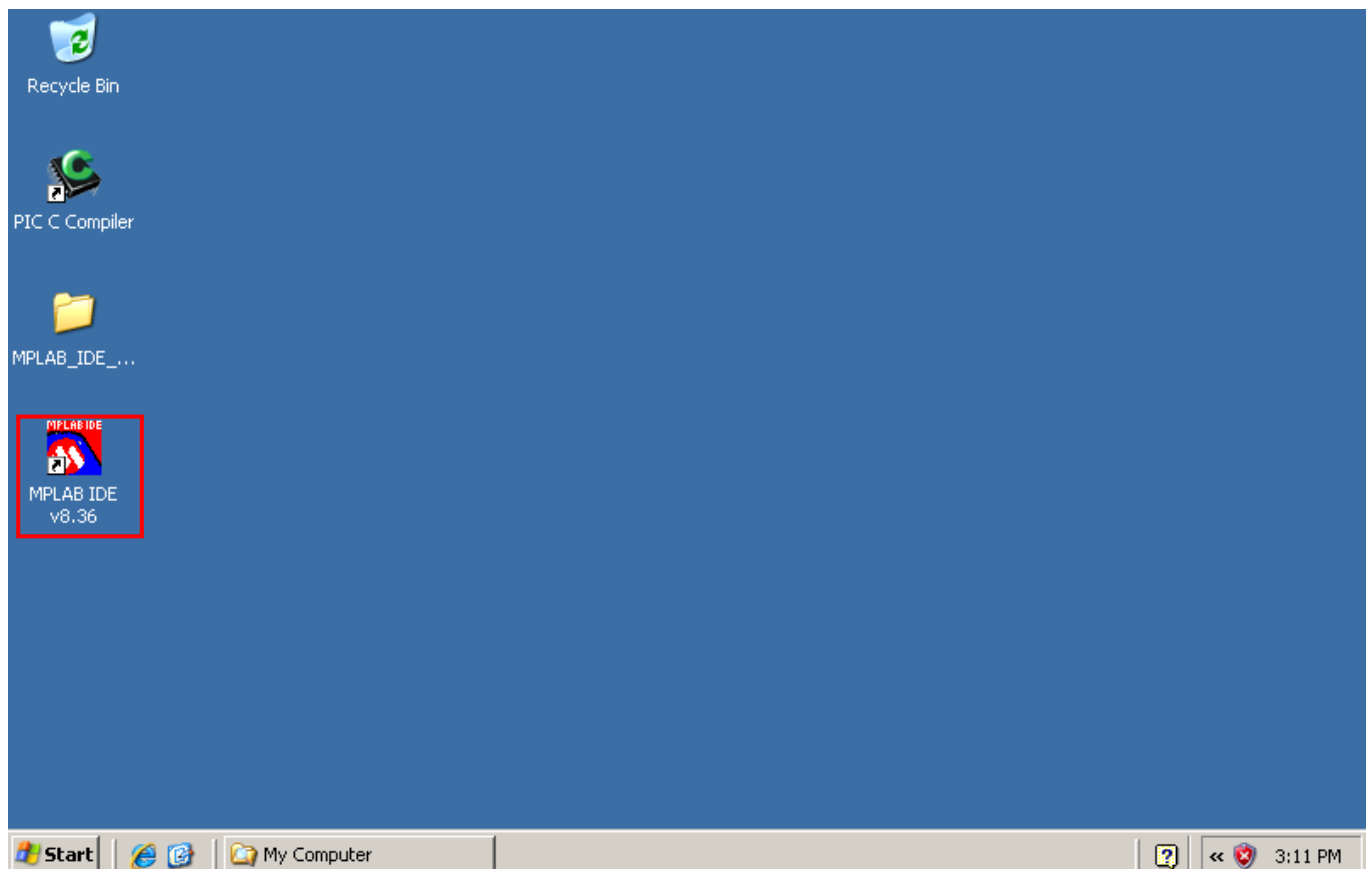
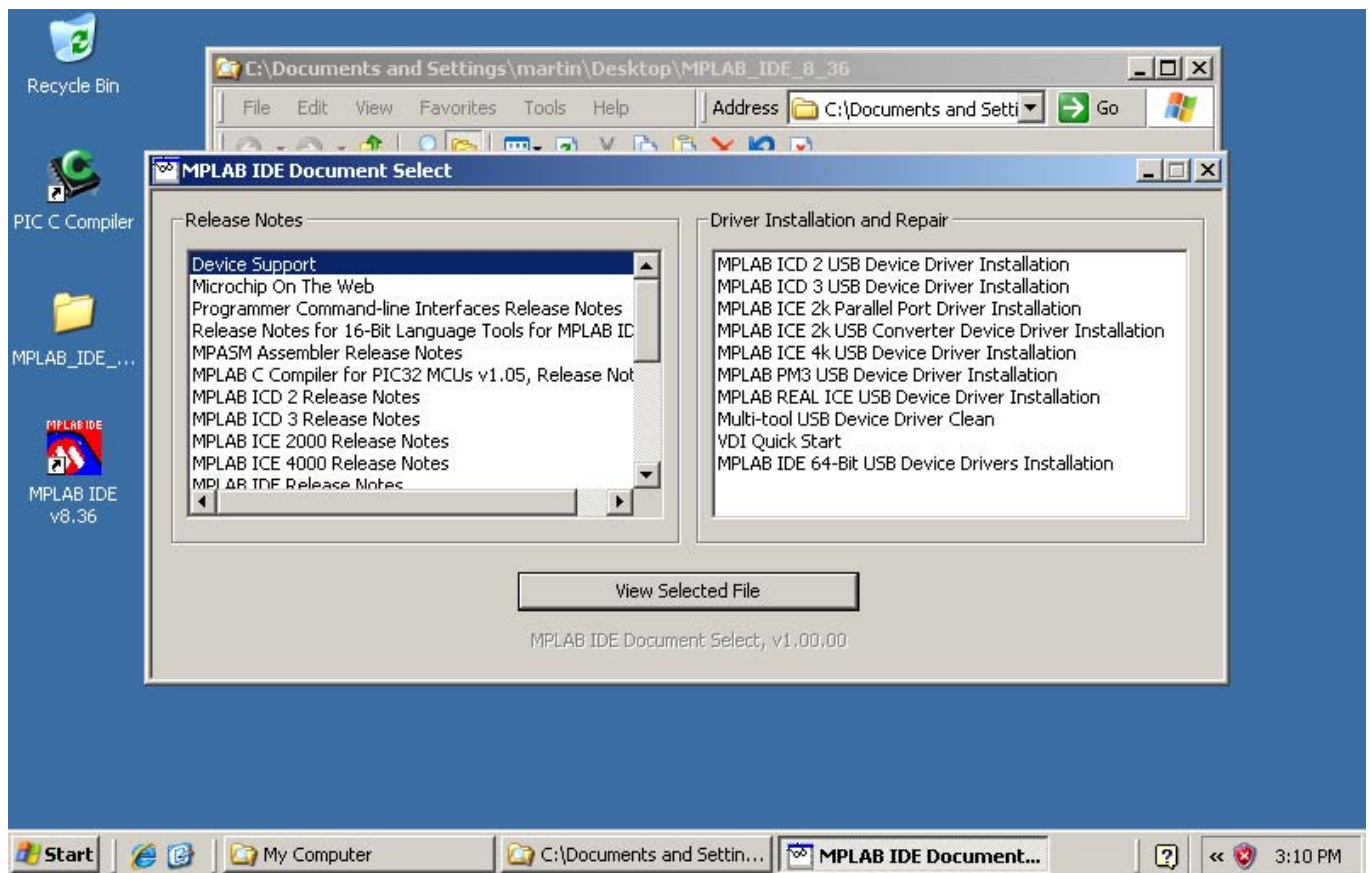






Click **No**. Bolymin driver supports only CCSC v4.093 binary object file and henceforth user have to apply the same compiler such that binary objects can be linkable.





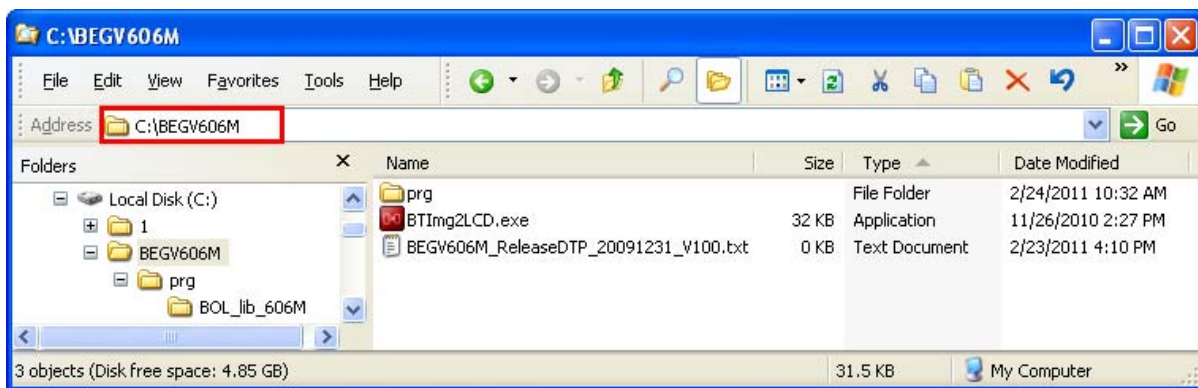
Note the **MPLAB IDE v8.36** is added on the desktop.

4.3 Install Bolymin Demo Code and library (SDK)

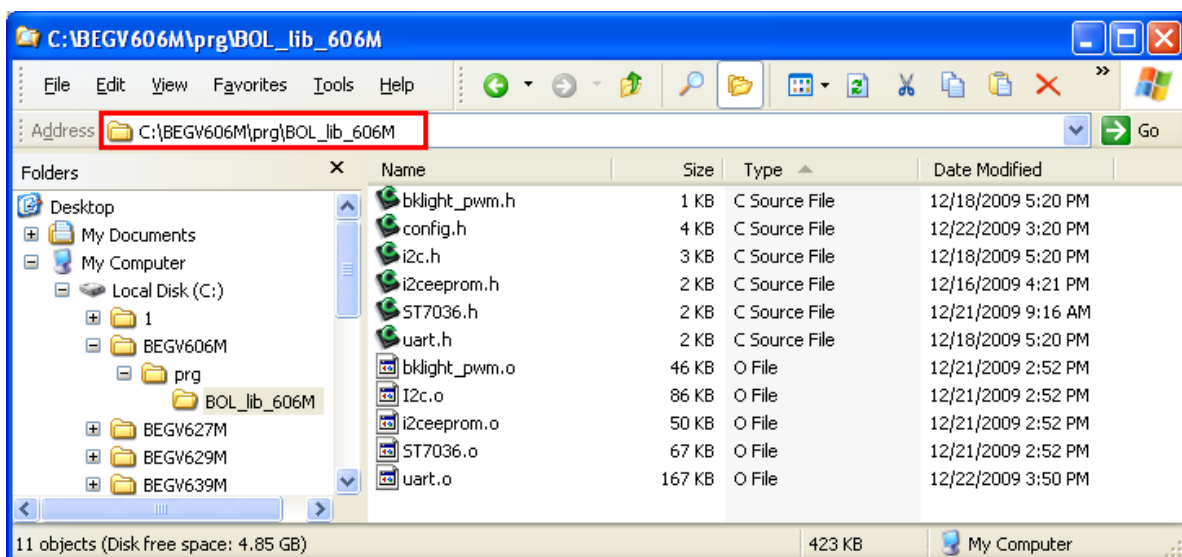
Here is the URL to download latest version of source code and libraries with pre-configured compile environment. <http://www.bolymin.com.tw/embedded/BEGV606M.rar>

Decompress the rar file into c:\BEGV606M and here is the file listing –

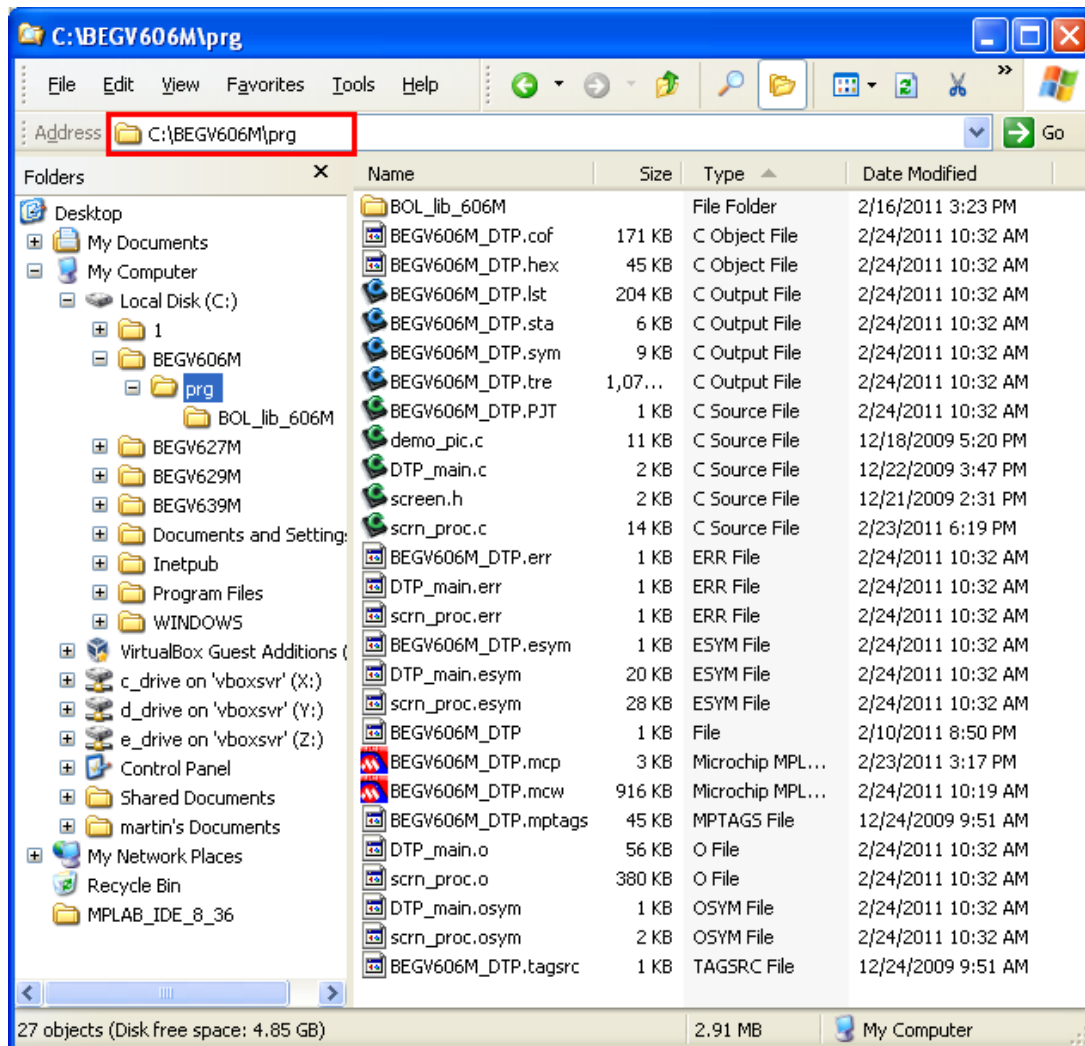
- **C:\BEGV606M** folder contains version info README file and BTImg2LCD, a bitmap font conversion utility program. Please refer to last appendix for step guide.



- Header file (*.h) and linkable object files (*.o) – use default folder as follows :



- Working folder contains all demo source code , intermediate files, and HEX codes

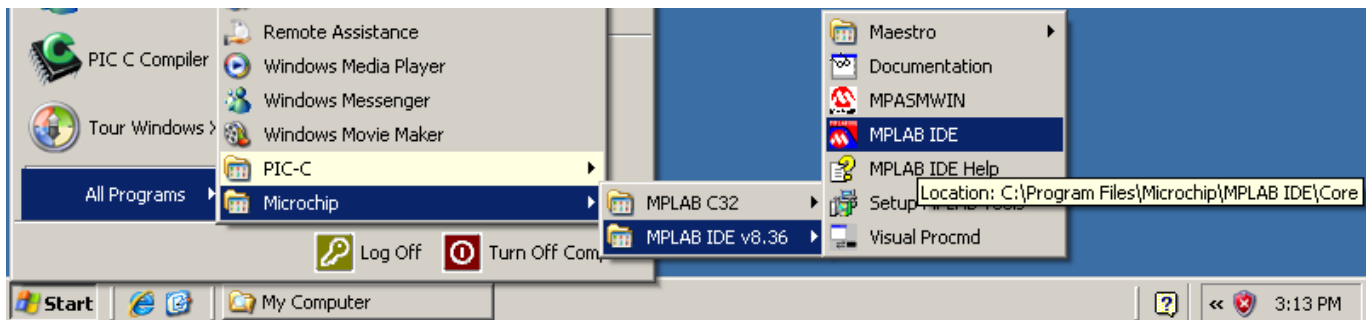


4.4 Compile source code using MPLAB IDE ver 8.36

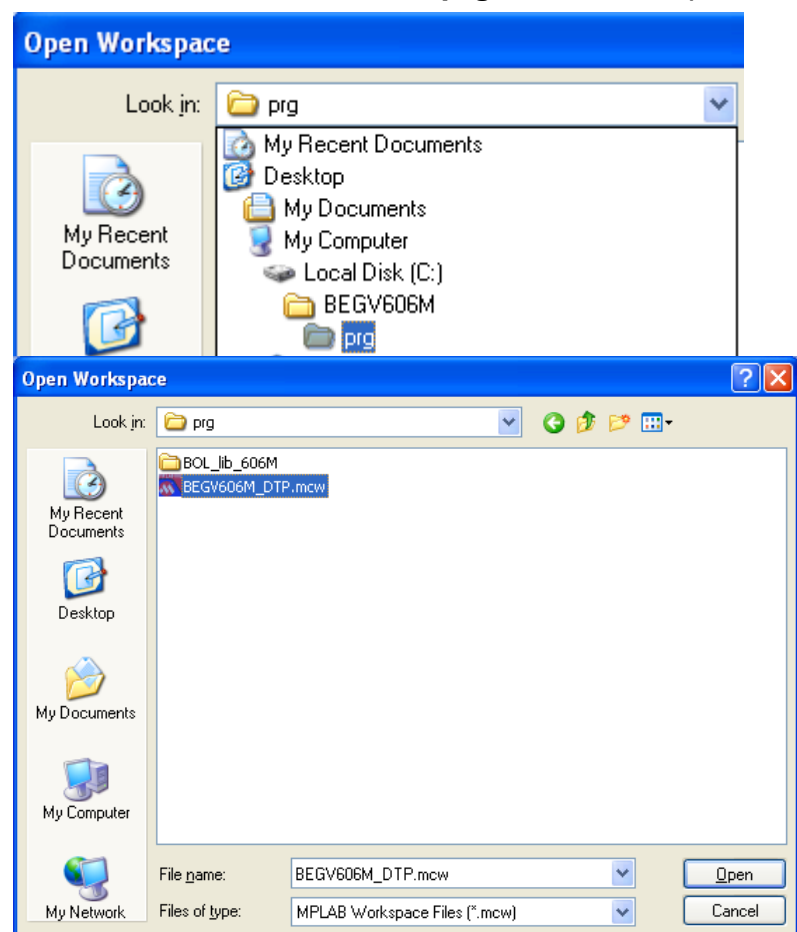
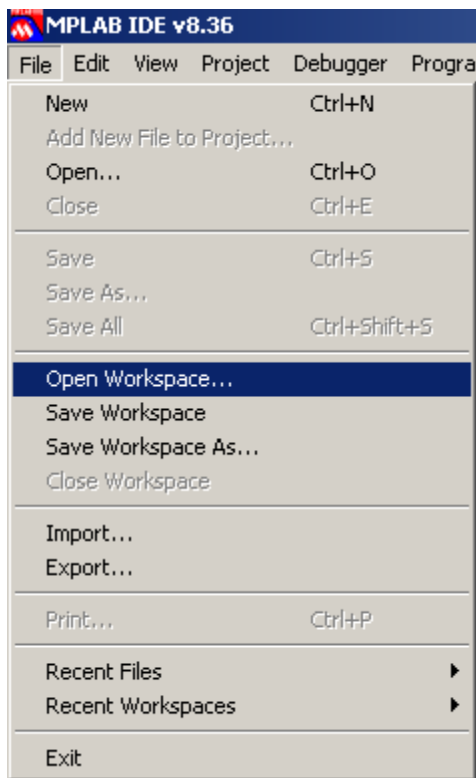
Please follow the step guide to make your 1st successful compilation.



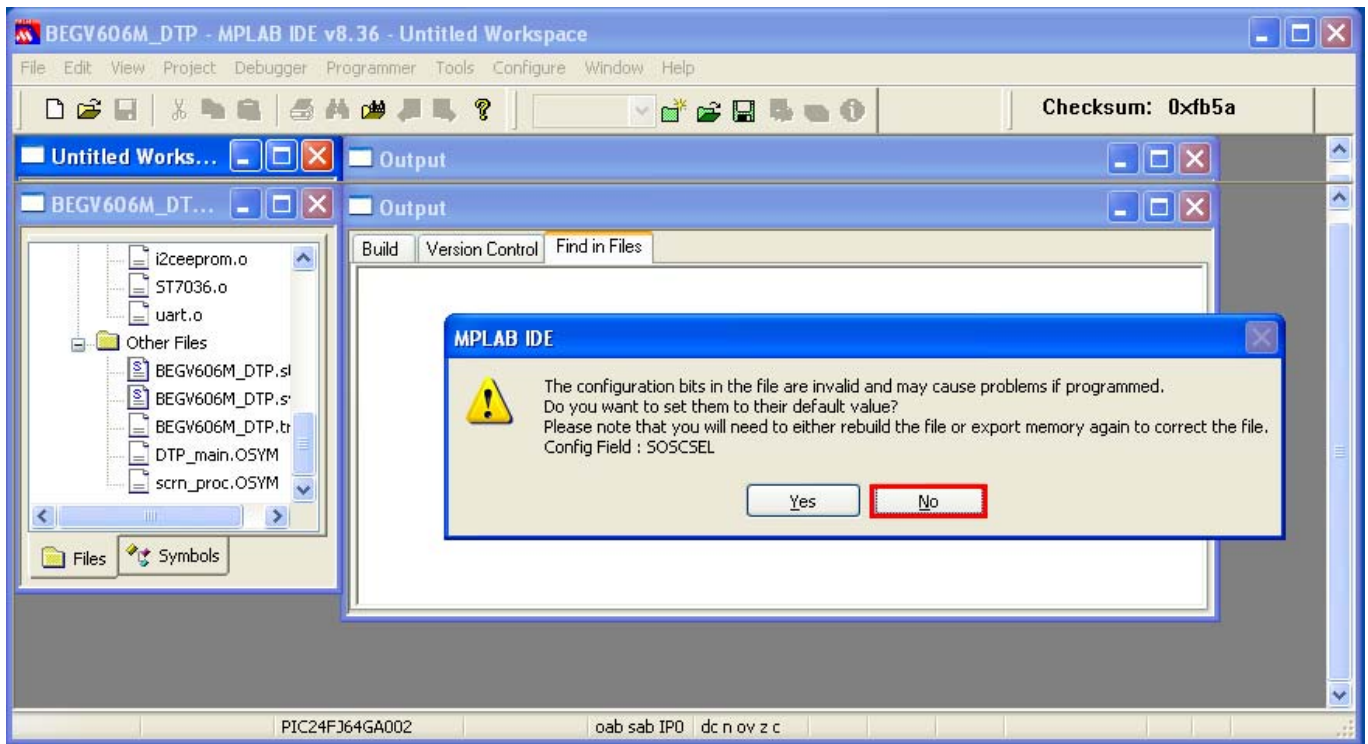
Step 1 - click left icon on desktop or **Start**→**All programs**→**Microchip**→**MPLAB IDE**



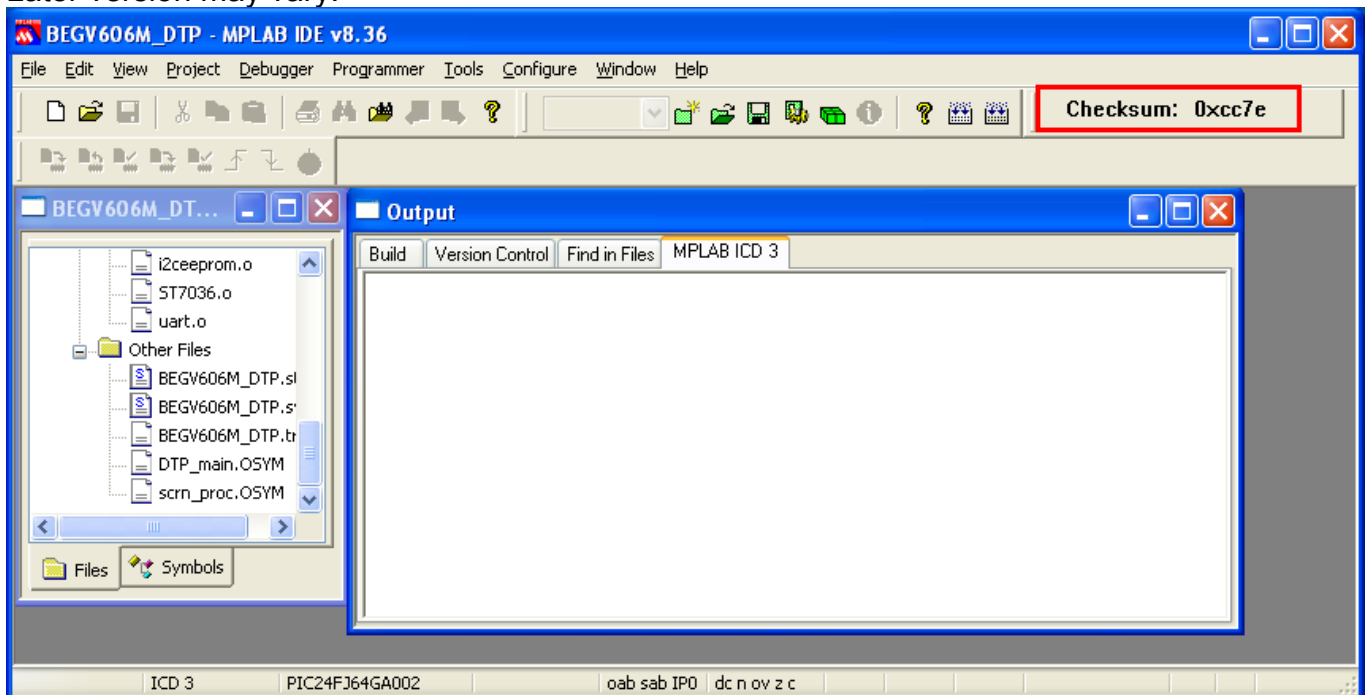
Step 2 – Click **File-Open Workspace** – browse into **c:\BEGV606M\prg** folder , and open **BEGV606_DTP.mcw**



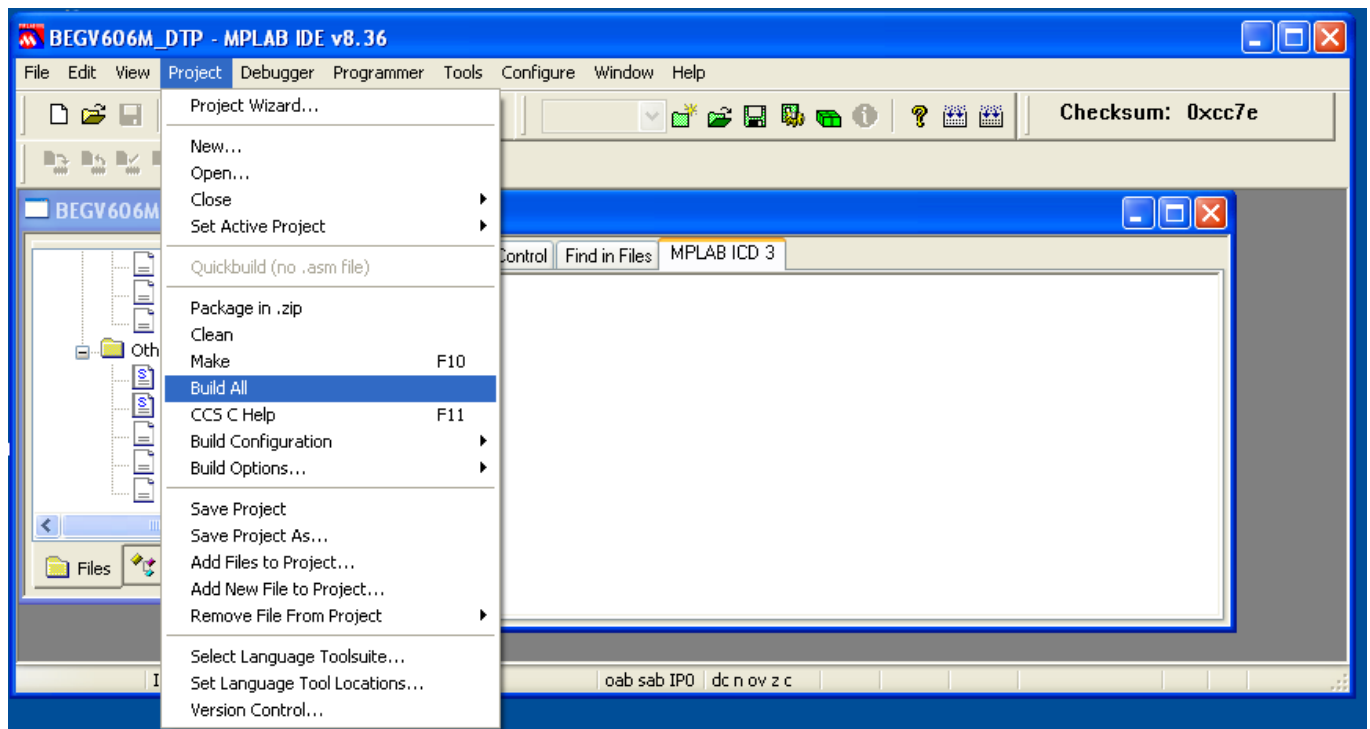
Step 3 – Click **No**. The configuration bits setting is written in C source code already and delay the warning message..



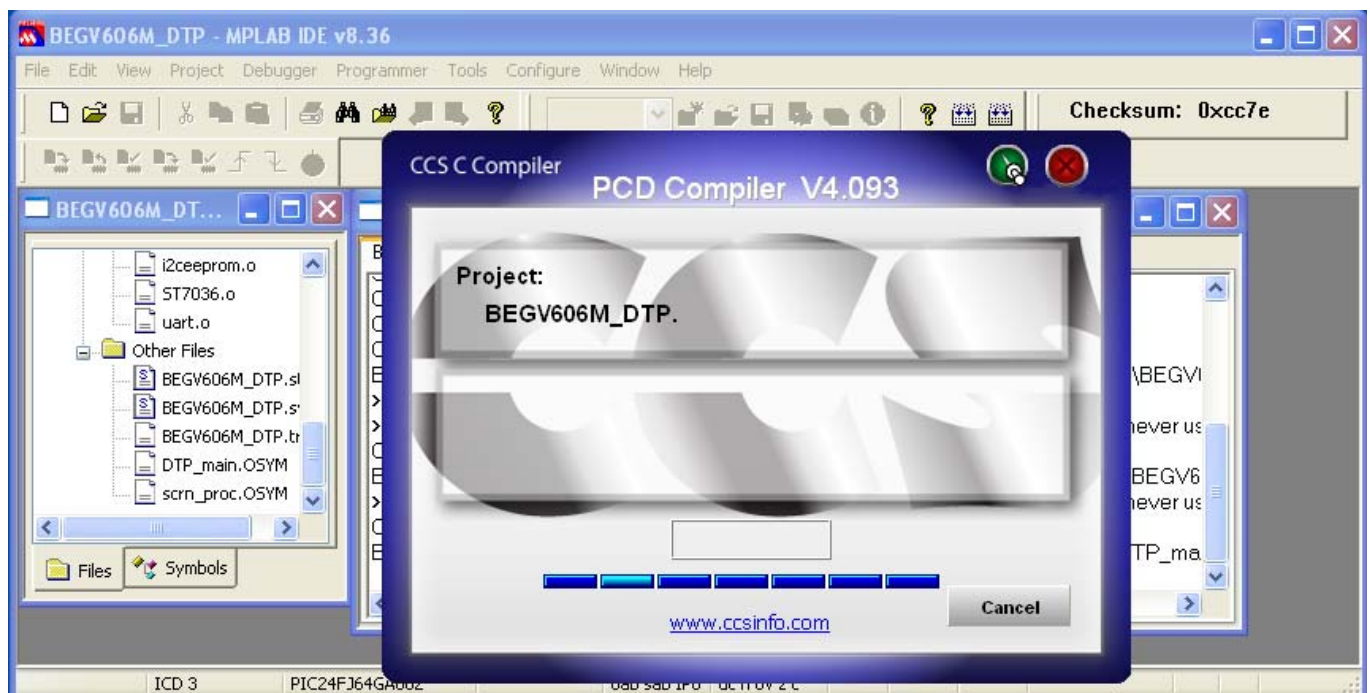
Step 4 – Your screen should be like this. Mind the checksum for version ver 1.00 source code. Later version may vary.



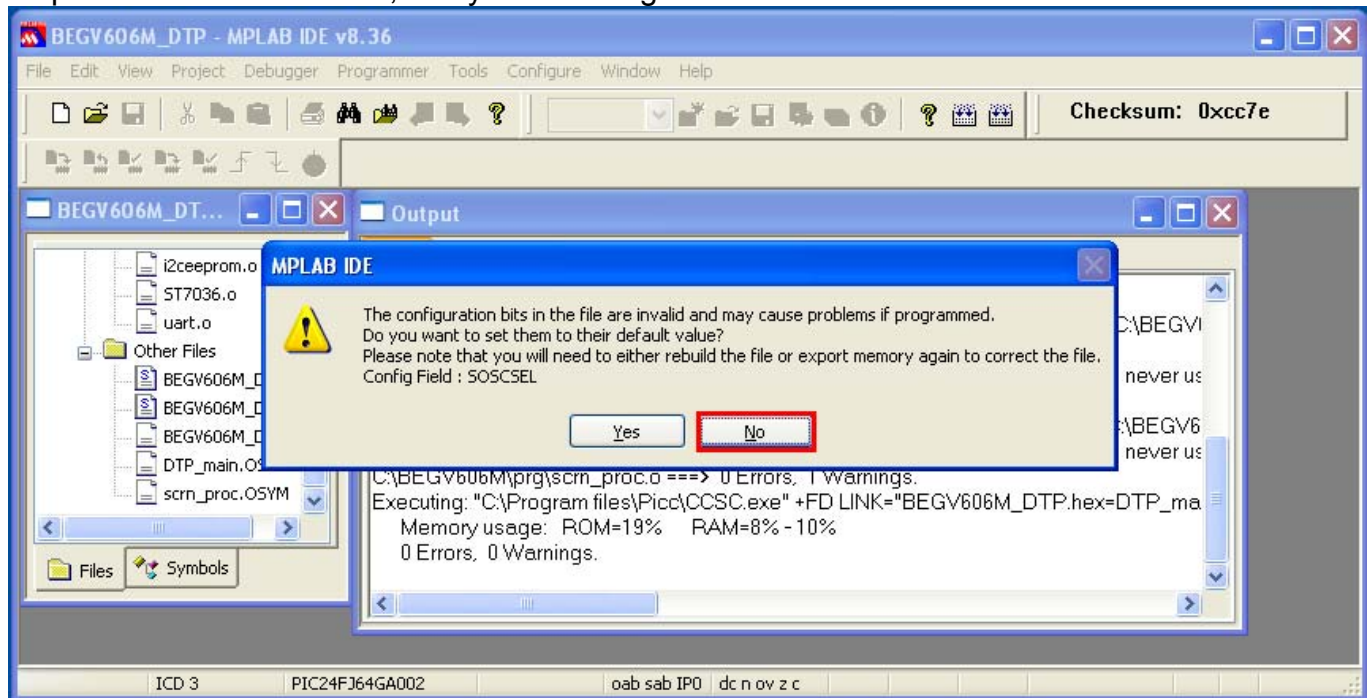
Step 5 – Now you are all set for 1st compilation. Click **Project-Build all**.



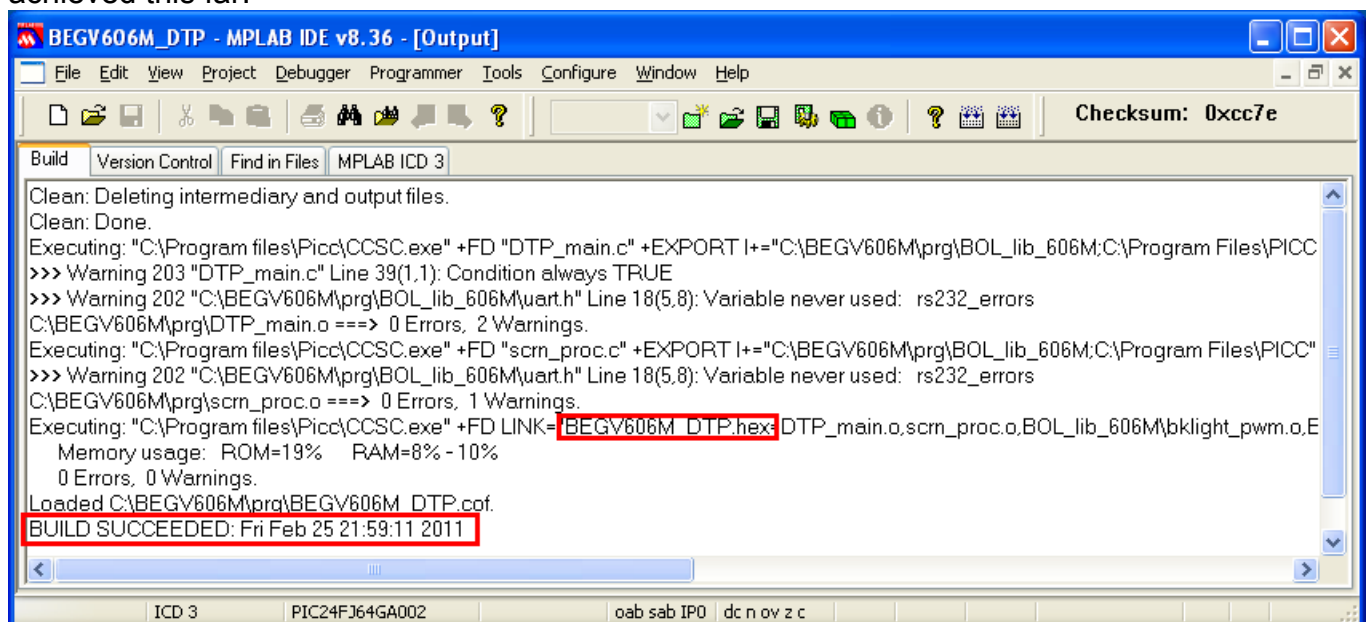
You will see CCSC compiling in progress as illustrated.



Step 6 – Here comes the same old warning message complaining about **configuration bit** as in step 2. For the same token, belay the message and click **No**.

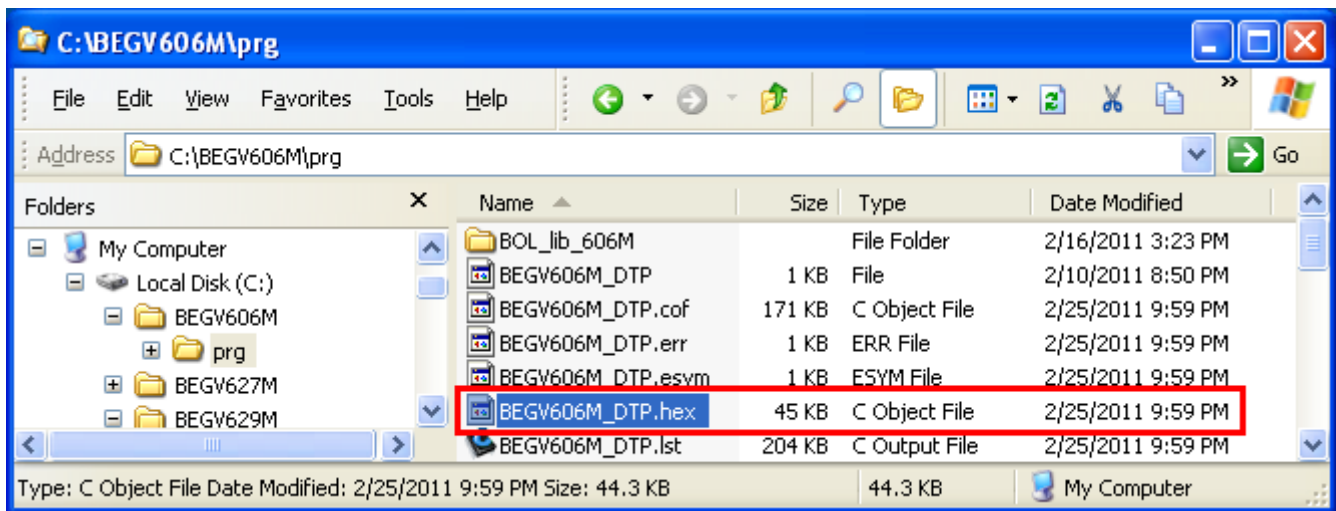


Step 7 – check the compilation result – go to output window by clicking **Windows-Output**, maximize it, and read the last line – **BUILD SUCCEEDED**. Congratulations for what you have achieved this far.

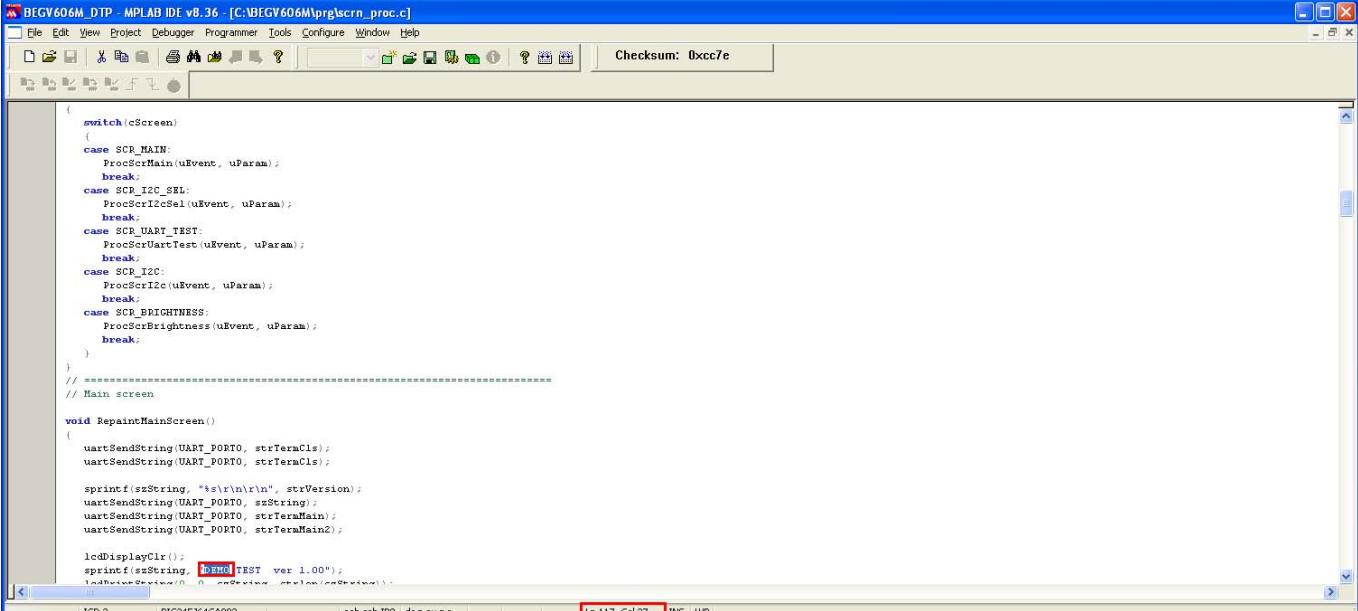


Also note that a new **BEGV606M_DTP.hex** file, the program HEX code for EEPROM programming, is generated. In the next section, we will introduce you the steps for writing EEPROM HEX code to target device using ICD3.

Step 8 - The generated HEX file locates in the working folder as follows. Double check the time stamp will be identical to the build time at the last line of above screen.



Advanced users may start playing with source code, change “Demo” into “Mars” for instance. Then recompile to generate new program HEX and write it into EEPROM. You will see the change takes effect as expected.



```

BEGV606M_DTP - MPLAB IDE v8.36 - [C:\BEGV606M\prg\scrn_proc.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xcc7e

{
  switch (cScreen)
  {
    case SCR_MAIN:
      ProcScrMain(uEvent, uParam);
      break;
    case SCR_I2C_SEL:
      ProcScrI2cSel(uEvent, uParam);
      break;
    case SCR_UART_TEST:
      ProcScrUartTest(uEvent, uParam);
      break;
    case SCR_I2C:
      ProcScrI2c(uEvent, uParam);
      break;
    case SCR_BRIGHTNESS:
      ProcScrBrightness(uEvent, uParam);
      break;
  }
}
// Main screen
void RepaintMainScreen()
{
  uartSendString(UART_PORT0, strTeraCls);
  uartSendString(UART_PORT0, strTeraCls);

  sprintf(ssString, "%s\r\n\r\n", strVersion);
  uartSendString(UART_PORT0, ssString);
  uartSendString(UART_PORT0, strTeraMain);
  uartSendString(UART_PORT0, strTeraMain2);

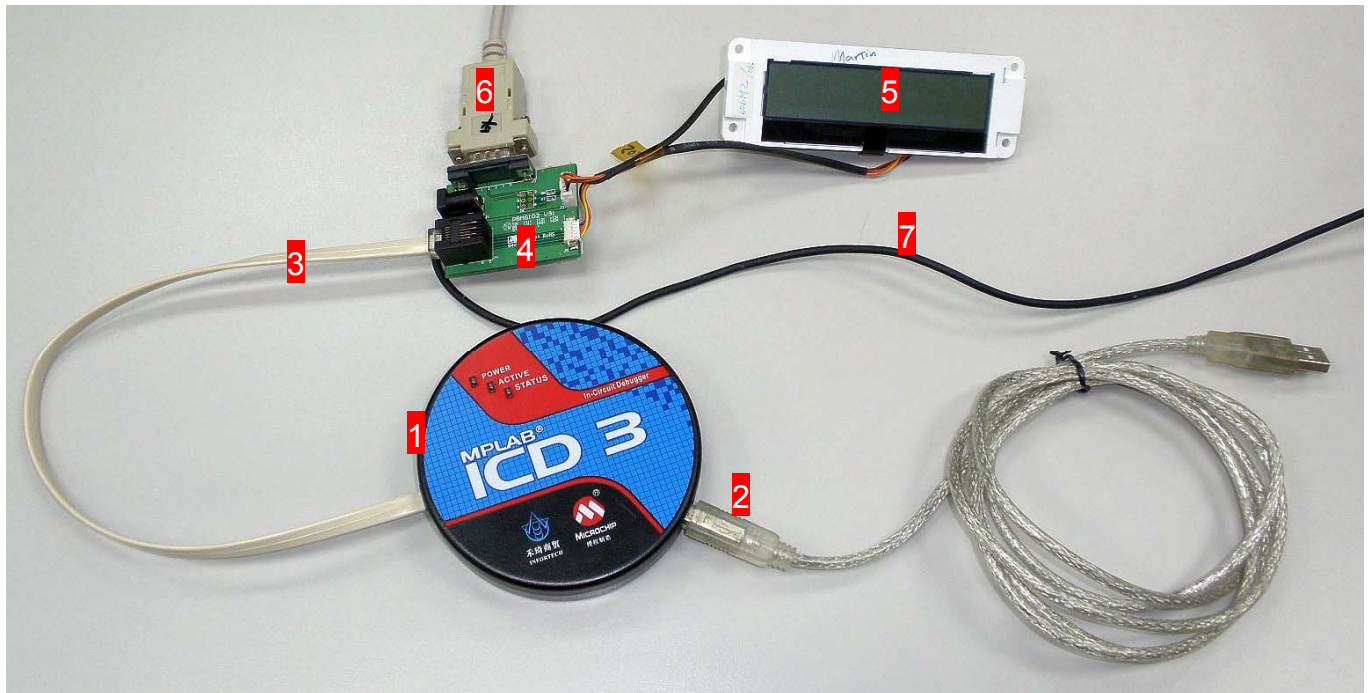
  lcdDisplayClr();
  printf(ssString, "Demo TEST ver 1.00");
  lcdWriteScreen(0, 0, 16, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}

```

ICD 3 PIC24FJ64GA002 oab sab IPO dc n ov z c Ln 117, Col 27 INS WR



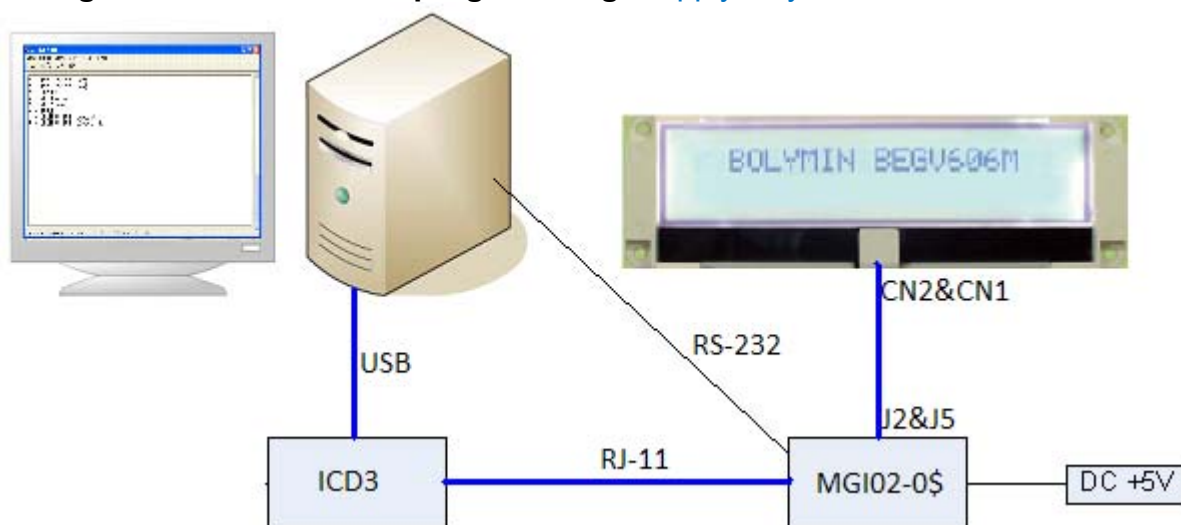
4.5 Programming HEX code using ICD3



** Inside ICD3(1), designer will get DVD, ICD3 device(2), RJ-45 cable(3) and USB cable(4). Please purchase MGI02-0\$ (4) , a Bolymin proprietary converter board to interface between ICD3 and target device, BEGV606M (5). You need a RS-232 straight-through cable (pin 2,3,5 used) for hyper terminal operation by PC (Windows XP recommended). Prepare a DC 5v/500mA adapter to connect to MGI02-0\$ ((4) to supply power to target device. User may custom may consider using 5volt from PC-USB(7) as power source.

ICD3 is always self-powered by PC-USB cable(2). **Note to remove 5 volt power from target board before ICD3 is connected to MGI02-0\$.**

Configuration for EEPROM programming – Apply only connection marked with blue.



Here are requirements for HEX code EEPROM programming-

Hardware Requirements

#	HW Item	Qty	Description
1	PC with COM port	1	Requires RS-232 port
2	Programmer-ICD3	1	
3	Converter board	1	MGI02-0\$
4	USB cable	1	Type A male←---→ Type B male
5	RS-232 ext. cable	1	Straight through RS-232 extension cable DB-9 male-DB-9 male; or use a USB-serial cable

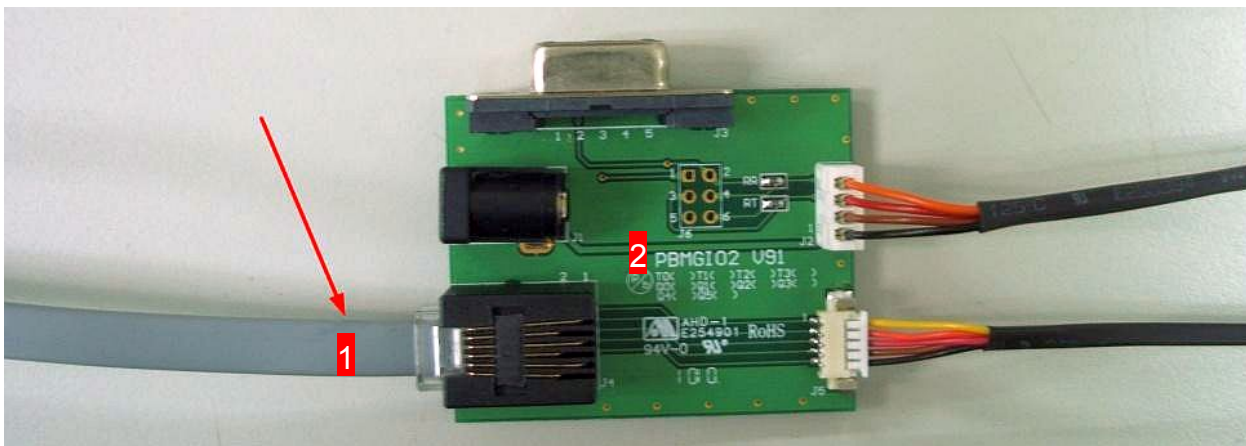
Software Requirements

#	SW item	Qty	Description
1	Programming IDE	1	MPLAB IDE 4.83 by Microchip
2	Terminal Emulation	1	Hyper Terminal (XP build in)
3	Bolymin DEMO program	1	Demo source code, DLL, and program HEX
4	English Fonts	1	For default English fonts
5	Japanese Fonts	1	For default Japanese fonts

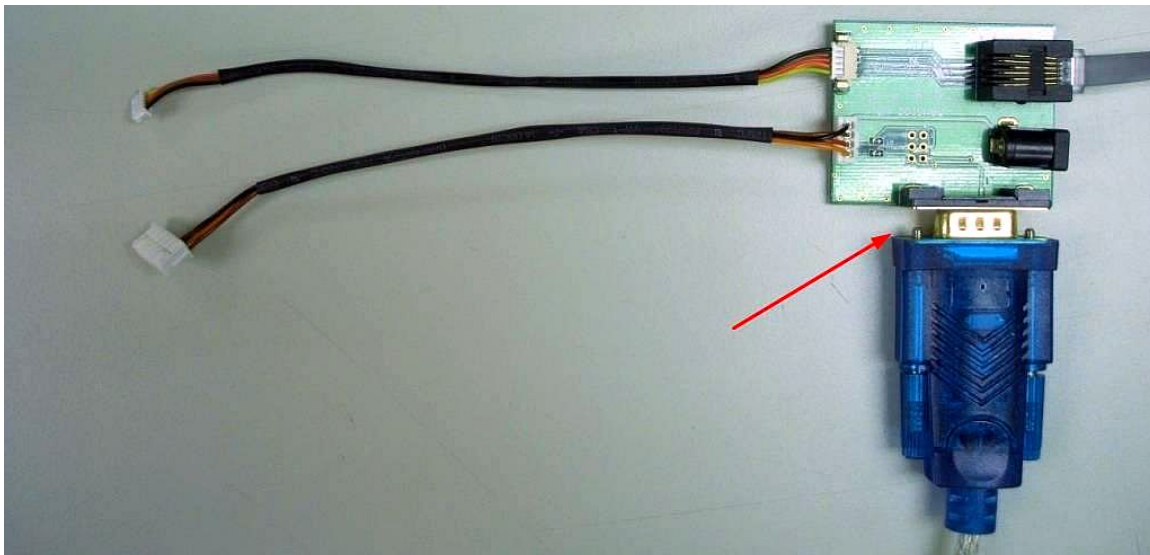
Step 1 - Connect 1 RJ-11 and 2 USB cable to 3 ICD3



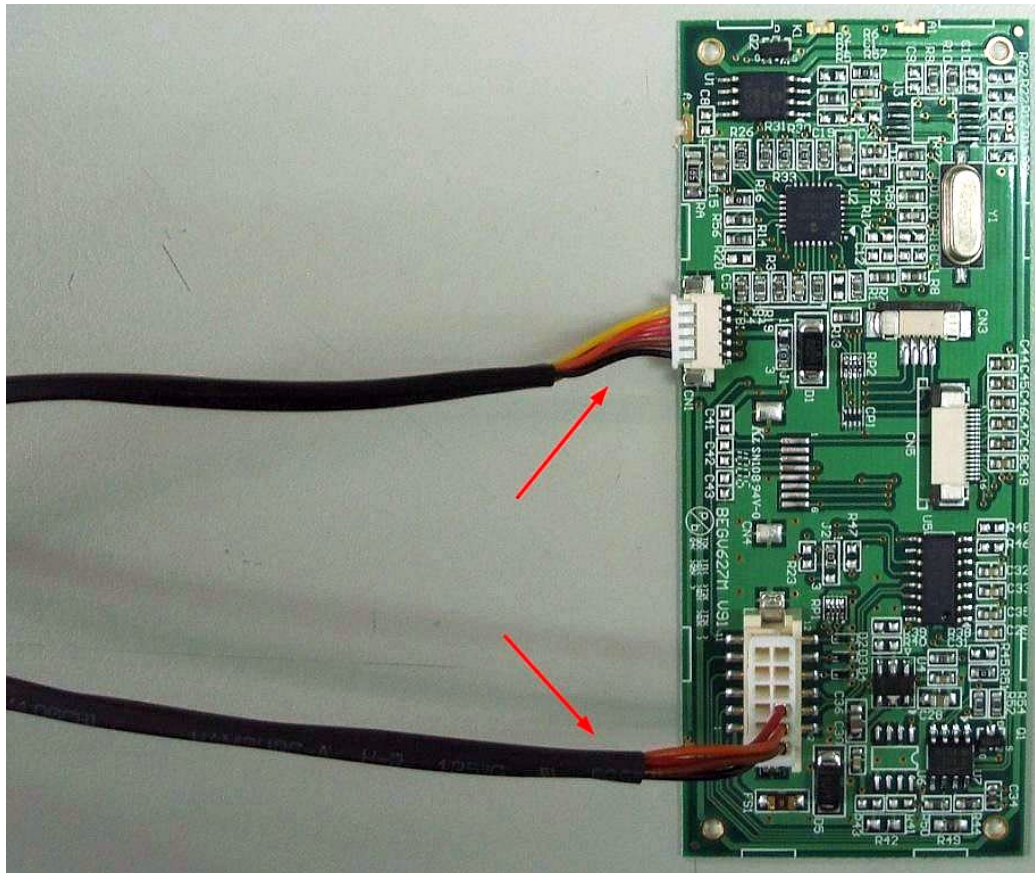
Step 2 - Connect 1 RJ-11 and cables to 2 MGI02-0\$



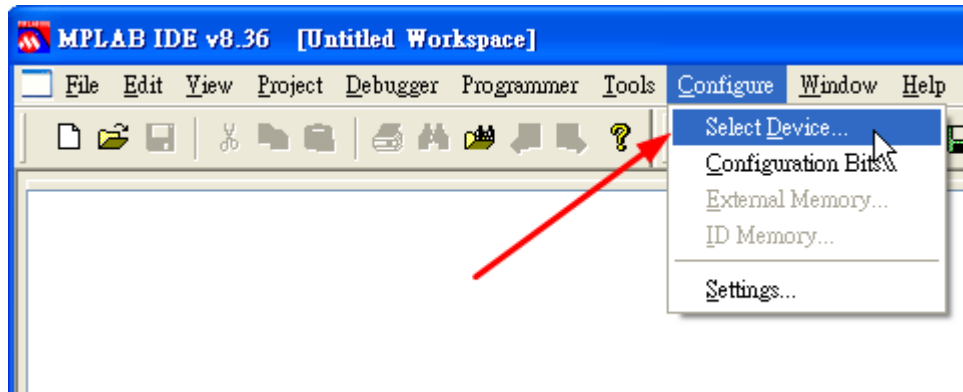
Step 3 - Connect MGI02-0\$ to PC using a RS-232 cable



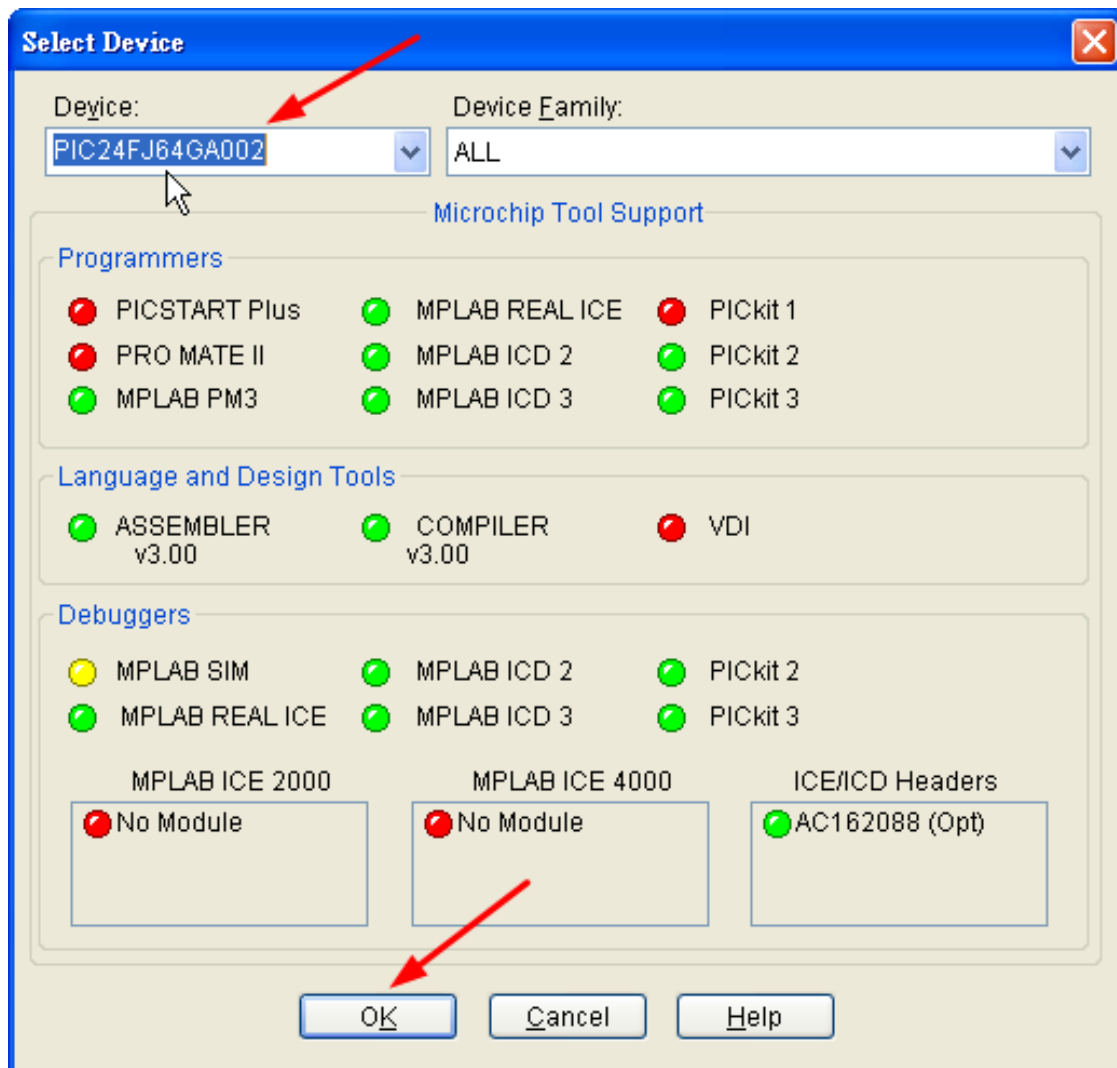
Step 4 - Connect MGI02-0\$ to BEGV606M.



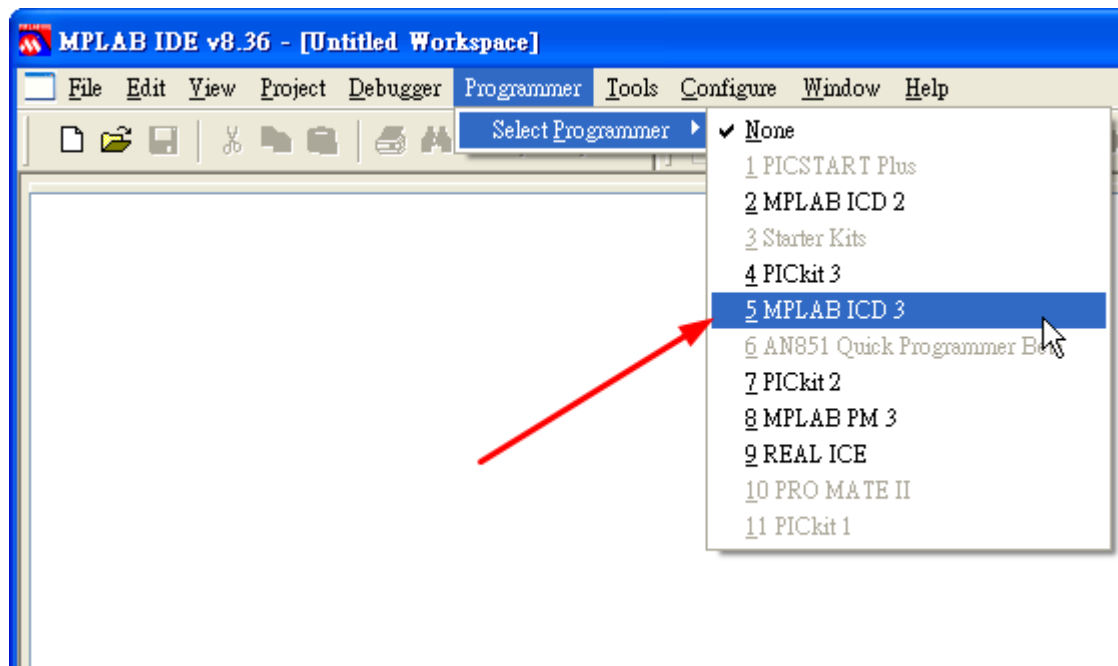
Step 5 - Open MPLAB IDE v8.36 , Click on Configure-Select Device



Step 6 - Select PIC24FJ64GA002 as Device, then click OK.

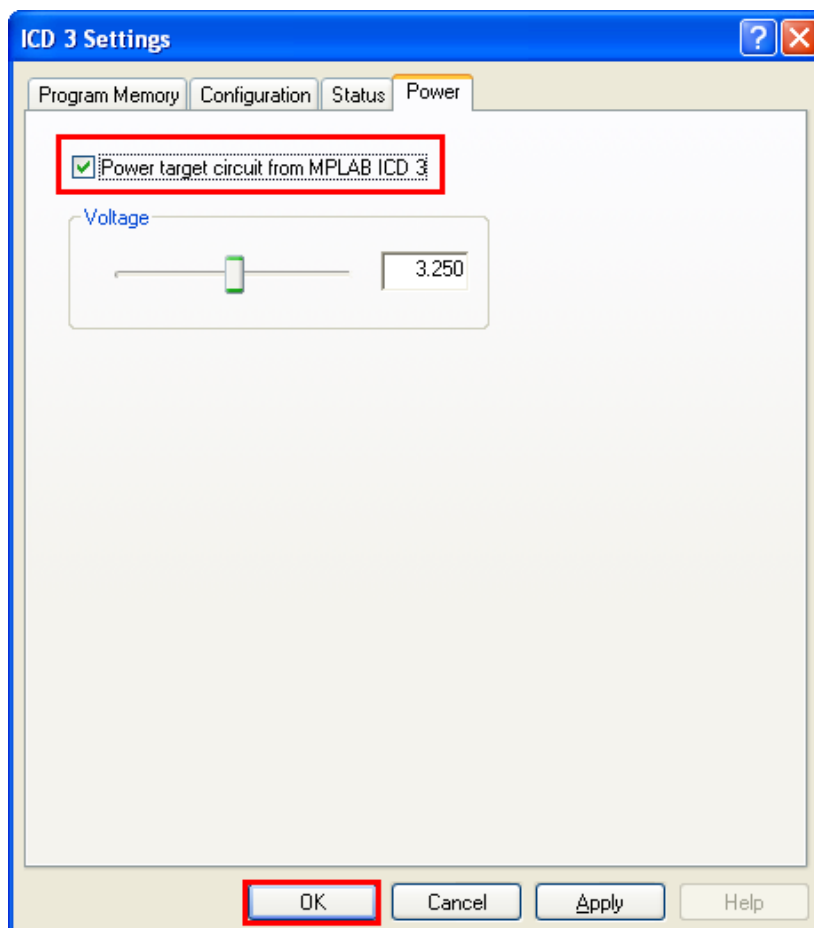
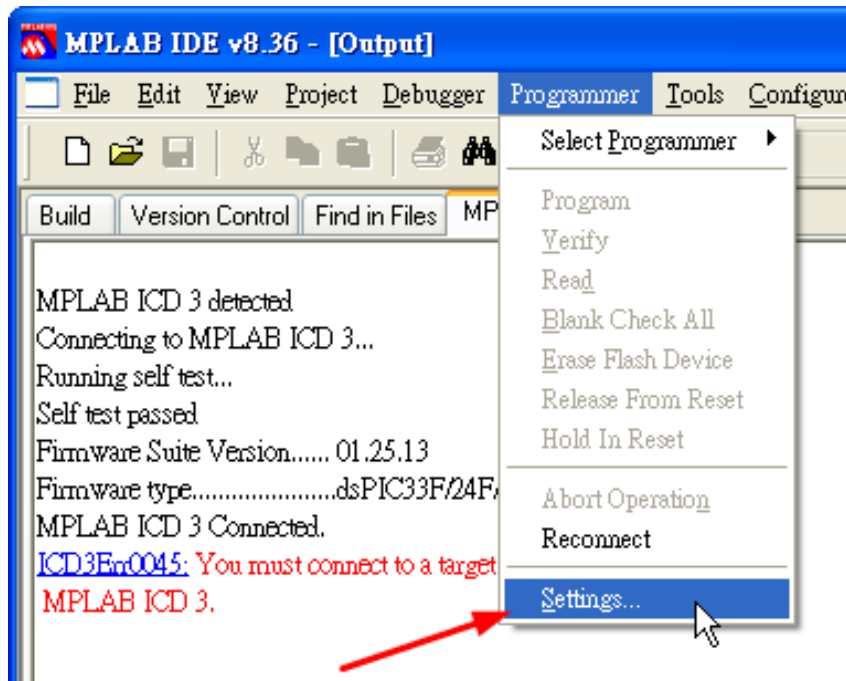


Step 7 – click Programmer-Select Programmer - 5 MPLAB ICD3.



Note target device is not powered as yet. Therefore there is no display on LCD.

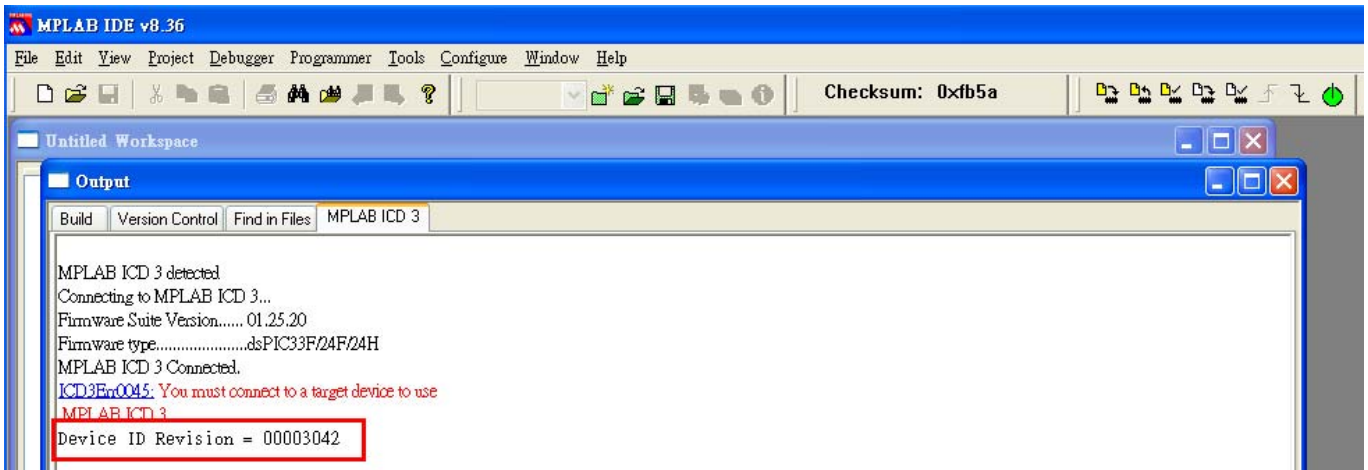
Step 8 - Change programmer settings to use ICD3 as power supply to target device. Click **Programmer-Settings**, select **Power** tab, check on **Power target circuit from MPLAB ICD3**. Then click OK.



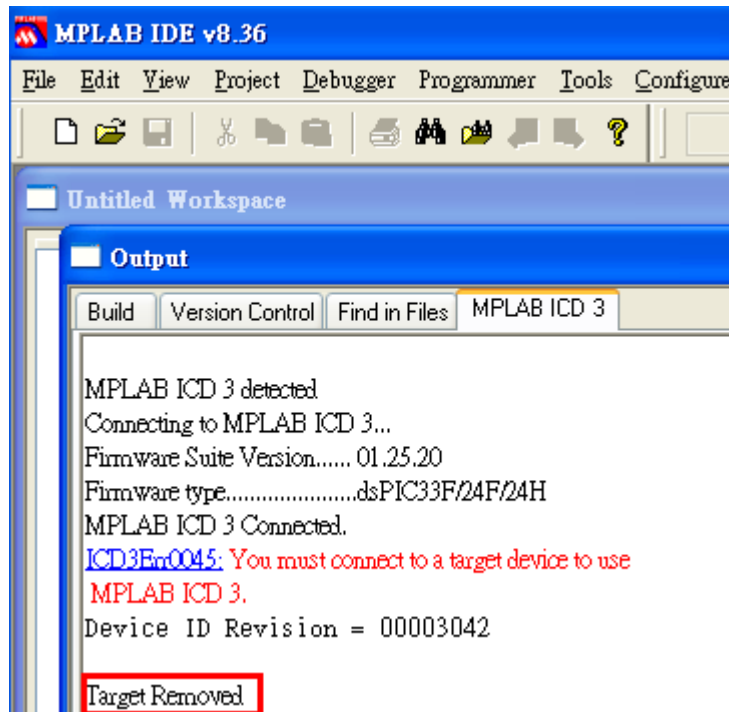
Step 9 - As soon as the power setting is done, note target device is now powered by ICD3 and LCD will display the default demo program as illustrated. Also on ICD3, observe the blue Active LED is on.



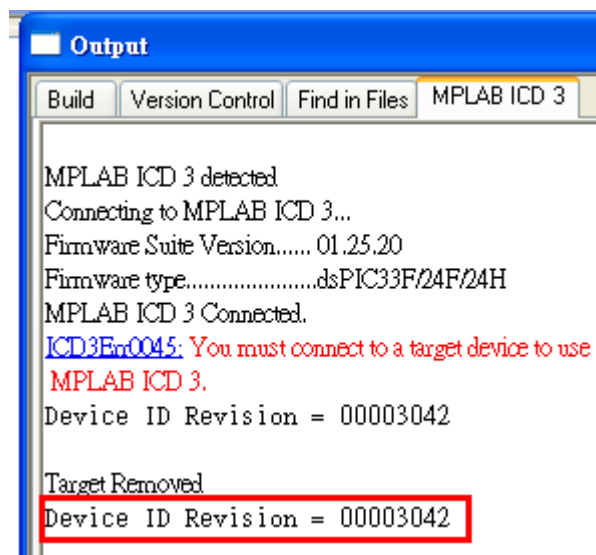
Notice a **Device ID Revision** will display on the last line of output window.



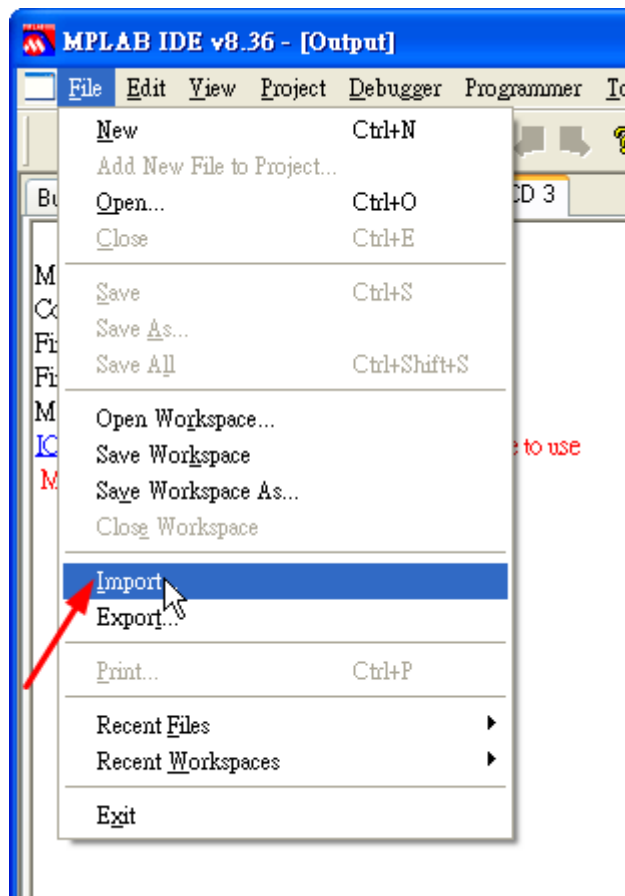
Step 10 - User may try toggle the power target device from ICD3 and observe the output windows message as follows, blue LED on ICD3 is off – 606M LCD display is off.



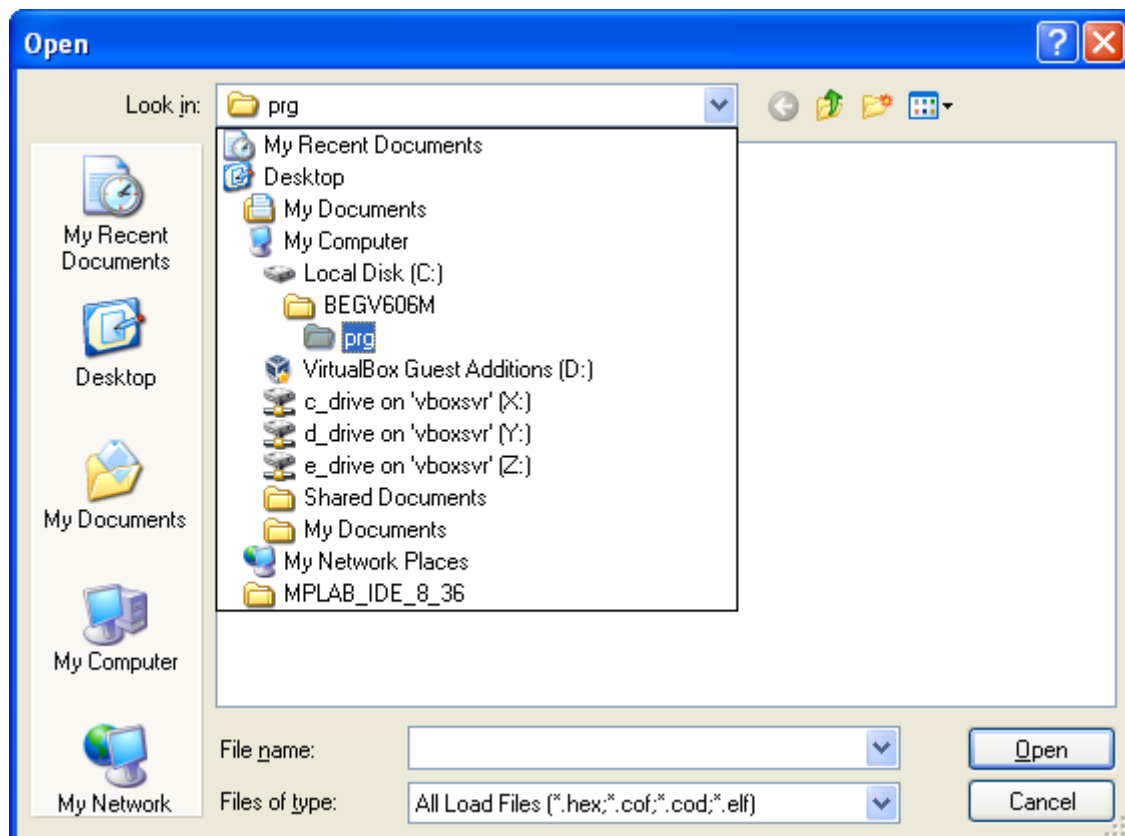
Now switch back to powered by ICD3.



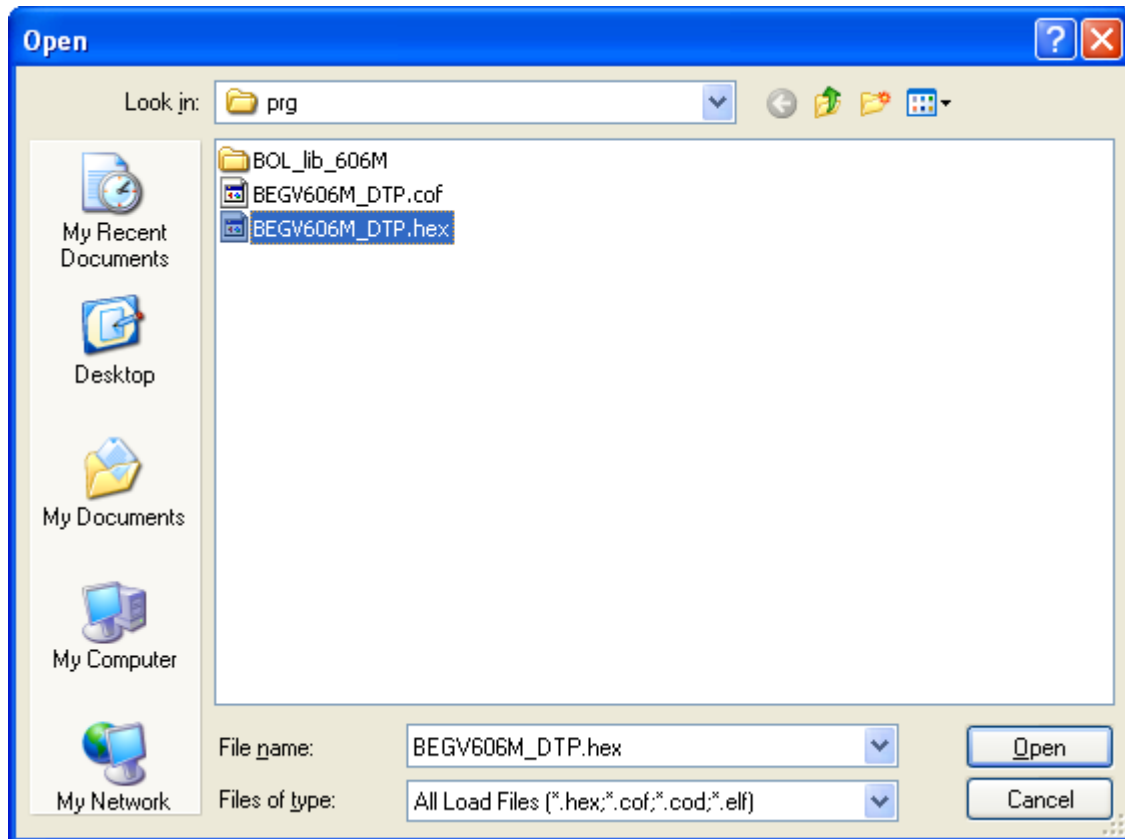
Step 11 - Select the Hex code for programming.



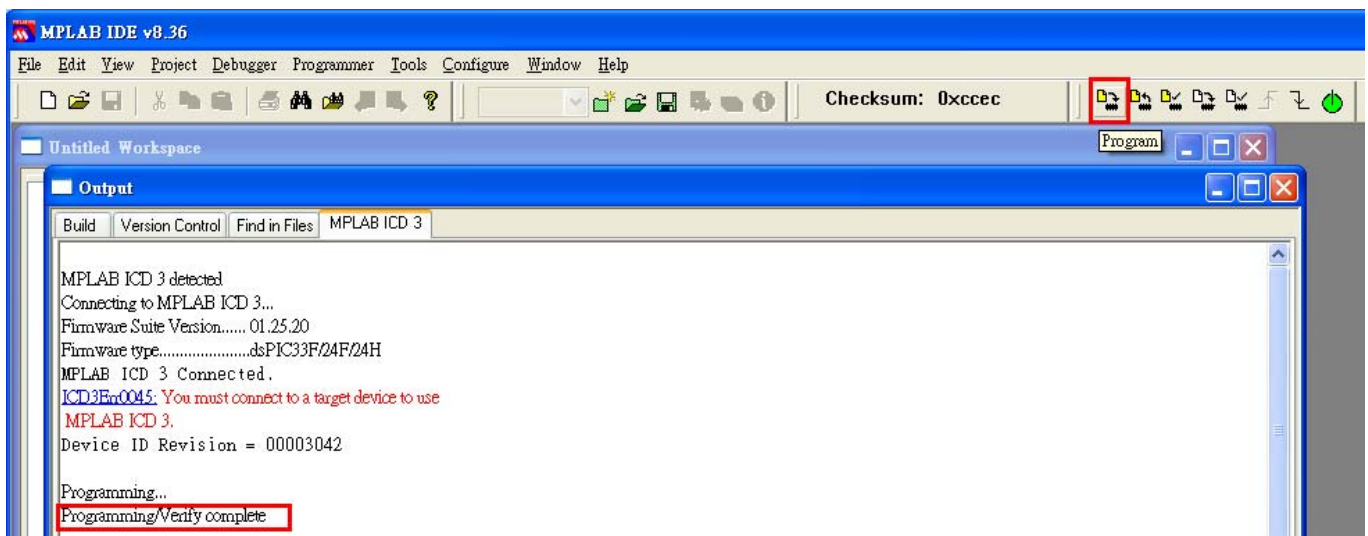
Browse into **c:\BEGV606M\prg** folder



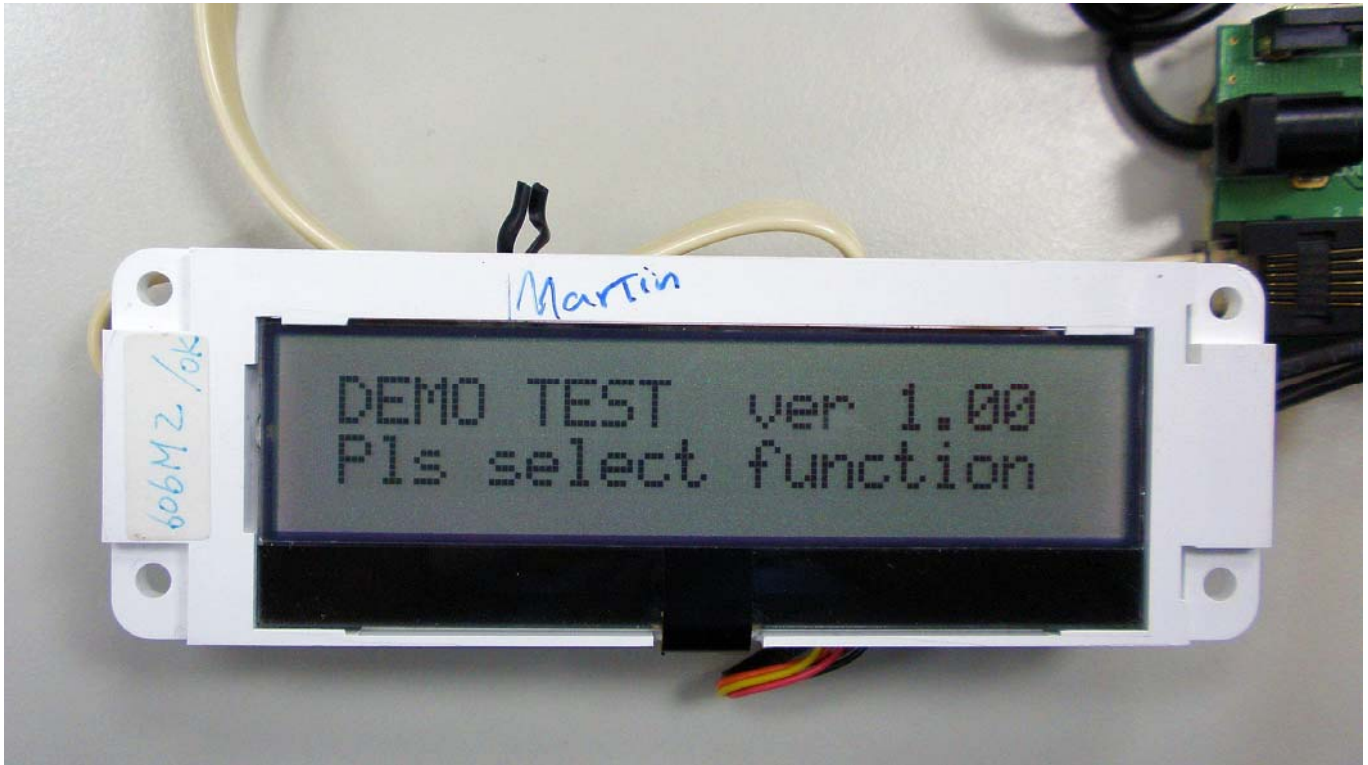
And select the HEX file



Step 12 - Click on  icon to program the program HEX code. Observe the output windows should display **Programming/Verify complete** upon completion of a successful HEX programming.



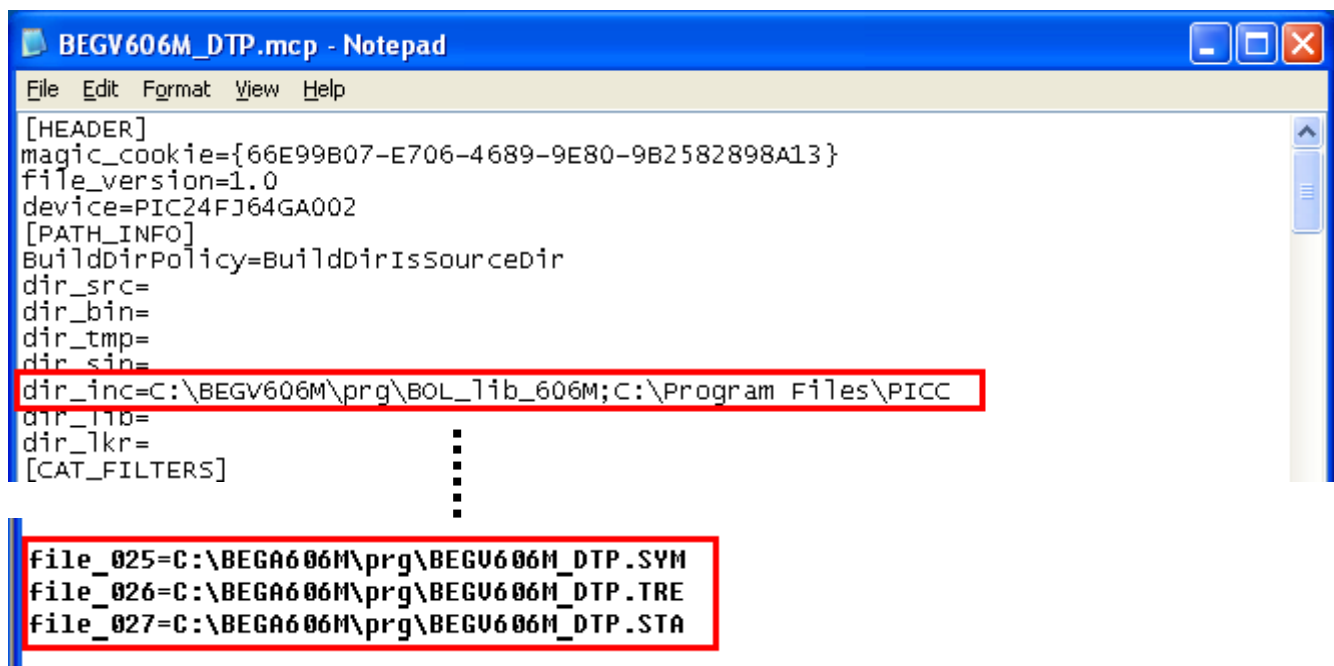
Step 13 - After HEX code programming, target device is reset and BEGV606M LCD should display as follows:



For more detail operation about ICD3, please also refer to Microchip URL:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en537580&redirects=icd3

In case user wanted to change to folder structure, remember to change .mcp file with appropriate folder name. Traverse through the .mcp and modify target folder as appropriate.



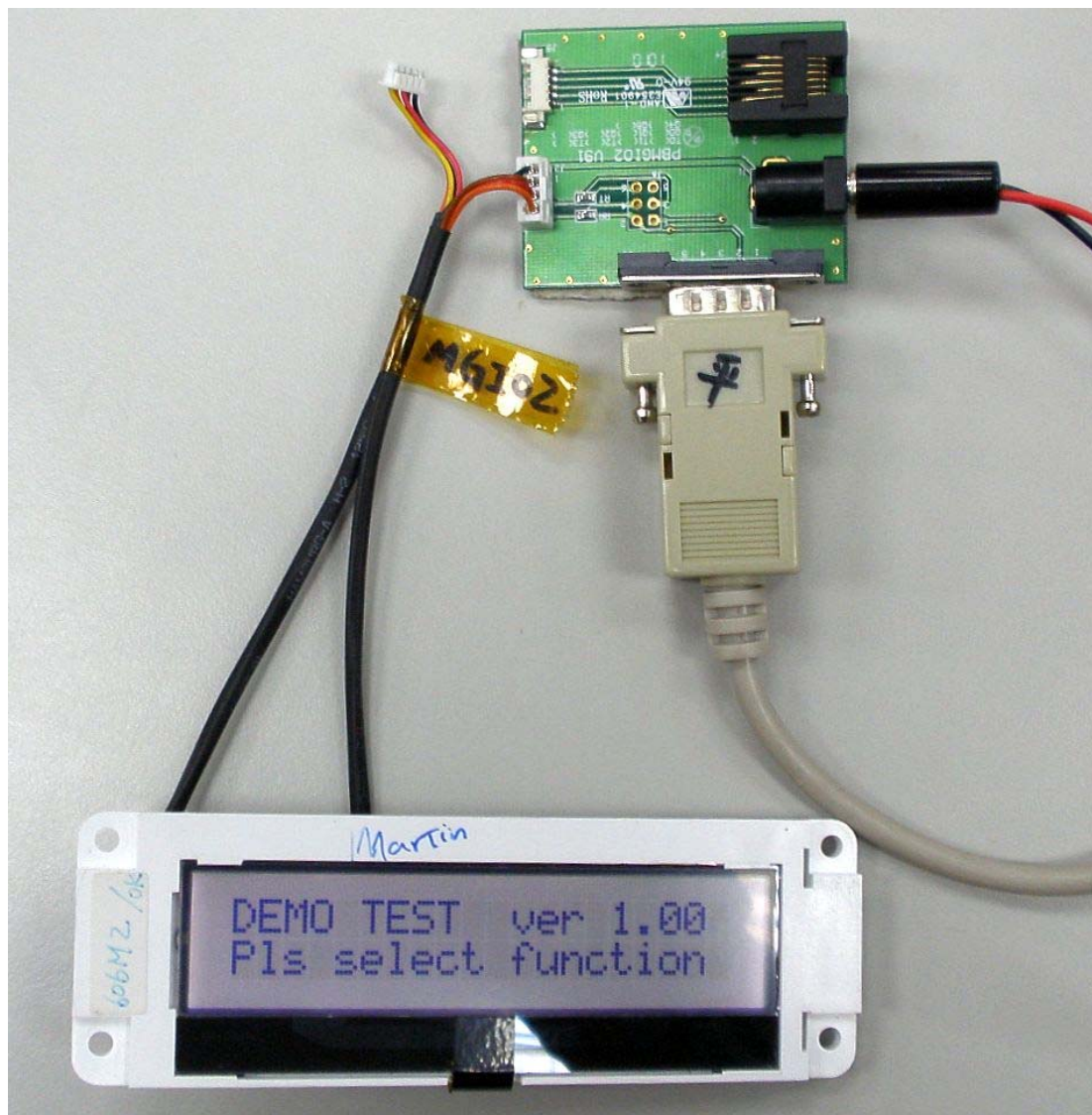
```

BEGV606M_DTP.mcp - Notepad
File Edit Format View Help
[HEADER]
magic_cookie={66E99B07-E706-4689-9E80-9B2582898A13}
file_version=1.0
device=PIC24FJ64GA002
[PATH_INFO]
BuildDirPolicy=BuildDirIsSourceDir
dir_src=
dir_bin=
dir_tmp=
dir_sin=
dir_inc=C:\BEGV606M\prg\BOL_lib_606M;C:\Program Files\PICC
dtp_110=
dir_1kr=
[CAT_FILTERS]
.
.
.
file_025=C:\BEGA606M\prg\BEGV606M_DTP.SYM
file_026=C:\BEGA606M\prg\BEGV606M_DTP.TRE
file_027=C:\BEGA606M\prg\BEGV606M_DTP.STA

```

4.6 Running Hyper Terminal

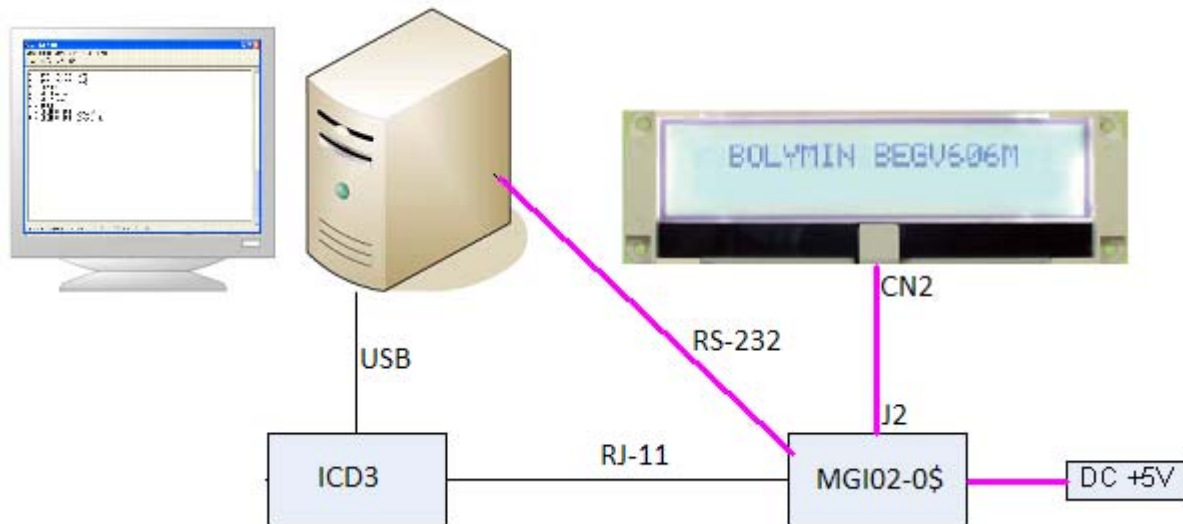
After EEPROM programming is done, disconnect ICD3, and power target device using 5Vdc. Now connect RS-232 from convert board to XP/Win7 PC. (Photo applies configuration #1, see next page)



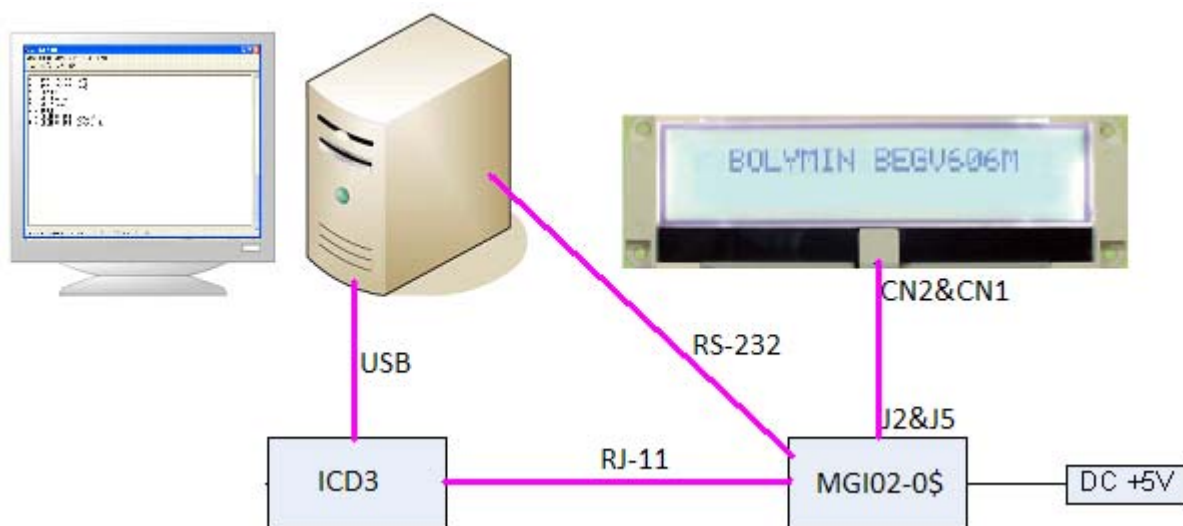
. Connect **CN2** of 606M to **J2** of MGI\$02 converter board. At least 4 pins are required as marked.

J2#	CN2 Signal	CN2 #	CN2 #	CN2 Signal	J2#
2	GND	2	1	VDD	1
4	TX0	4	3	NC	
3	RX0	6	5	NC	
	IOA	8	7	TX1	
	IOB	10	9	RX1	
	IOC	12	11	/RST	

Configuration #1 for hyper terminal operation – As highlighted pink, apply External DC 5v , only J2-CN2 connection needed, and RS-232 to support hyper terminal operation from XP PC end.



Configuration #2 for hyper terminal operation – Alternative connection is have target device draw power directly from ICD3. Note external 5Vdc should NOT be connected. Remember to change ICD3 setting using MPLAB as stated at step 10 of Chapter 4-5.

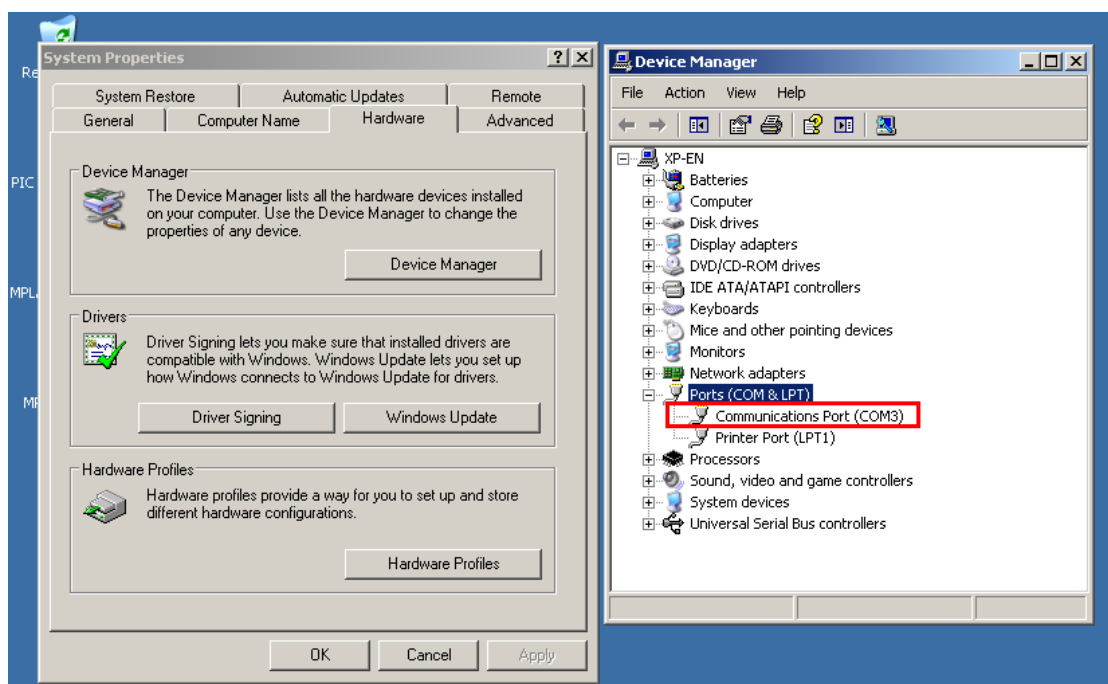


Either configuration works, configuration #1 might be easier if there is no source code change involved. Also configuration #1 provides a 5 volt rather than 3.3volt on target device.

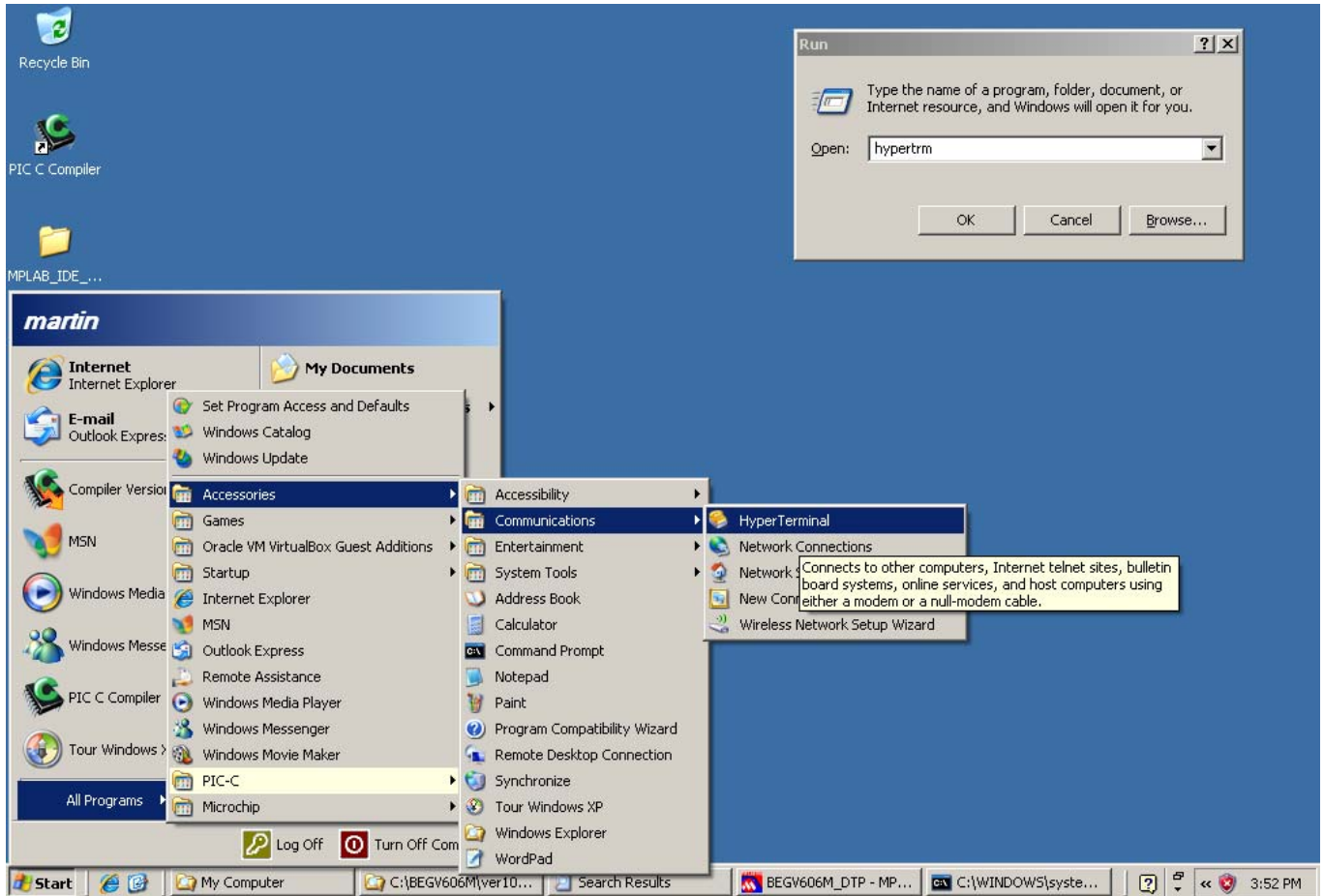
Use **Hyper terminal** to operate the demo program

Since there is no touch panel on 606M, please use hyper terminal for terminal emulation and kernel debugging. The PC keyboard can emulate as a input device to 606M. Here is the step guide -

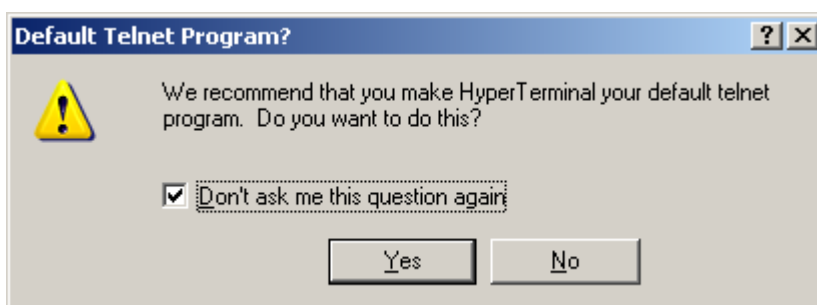
Step 1 - Make sure you have at least one **RS-232 serial port** available. The following example shows a COM3 is available.



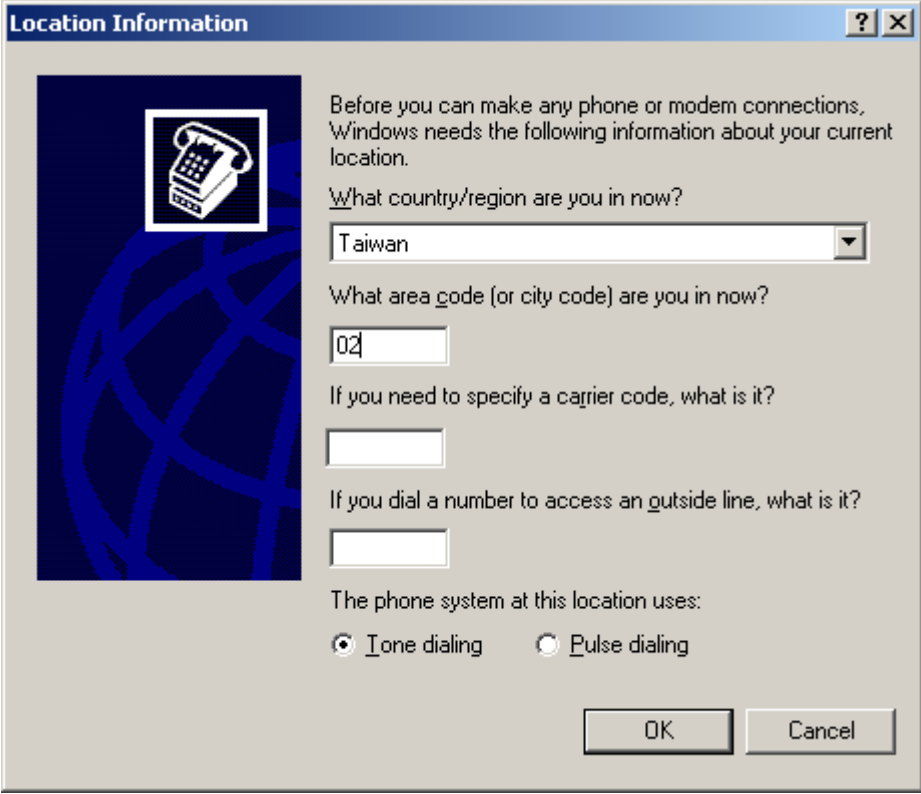
Step 2 - On XP PC: Start→All programs→Telecommunication→HyperTerminal (or Windows+R, hypertrm)



Step 3 - Click No if you do not use hyper terminal to telnet to UNIX host.



Step 4 – enter **country** and **area code** as appropriate.



Location Information

Before you can make any phone or modem connections, Windows needs the following information about your current location.

What country/region are you in now?

Taiwan

What area code (or city code) are you in now?

02

If you need to specify a carrier code, what is it?

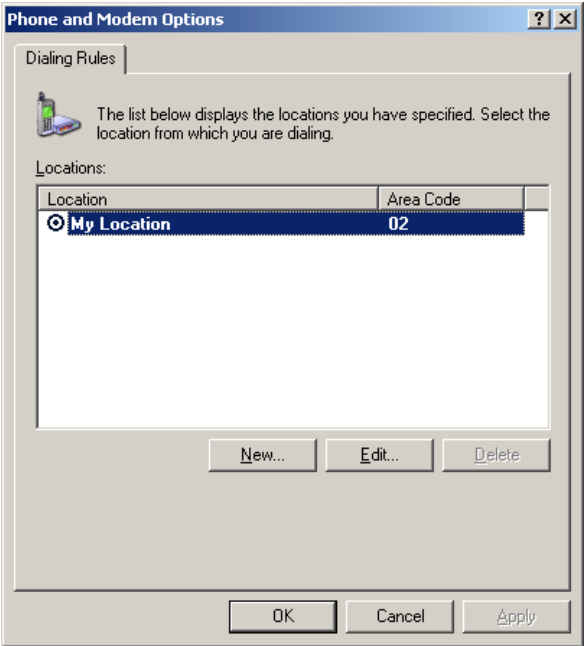
If you dial a number to access an outside line, what is it?

The phone system at this location uses:

☒ T one dialing ☐ Pulse dialing

OK Cancel

Step 5 – Click **OK**.



Phone and Modem Options

Dialing Rules

The list below displays the locations you have specified. Select the location from which you are dialing.

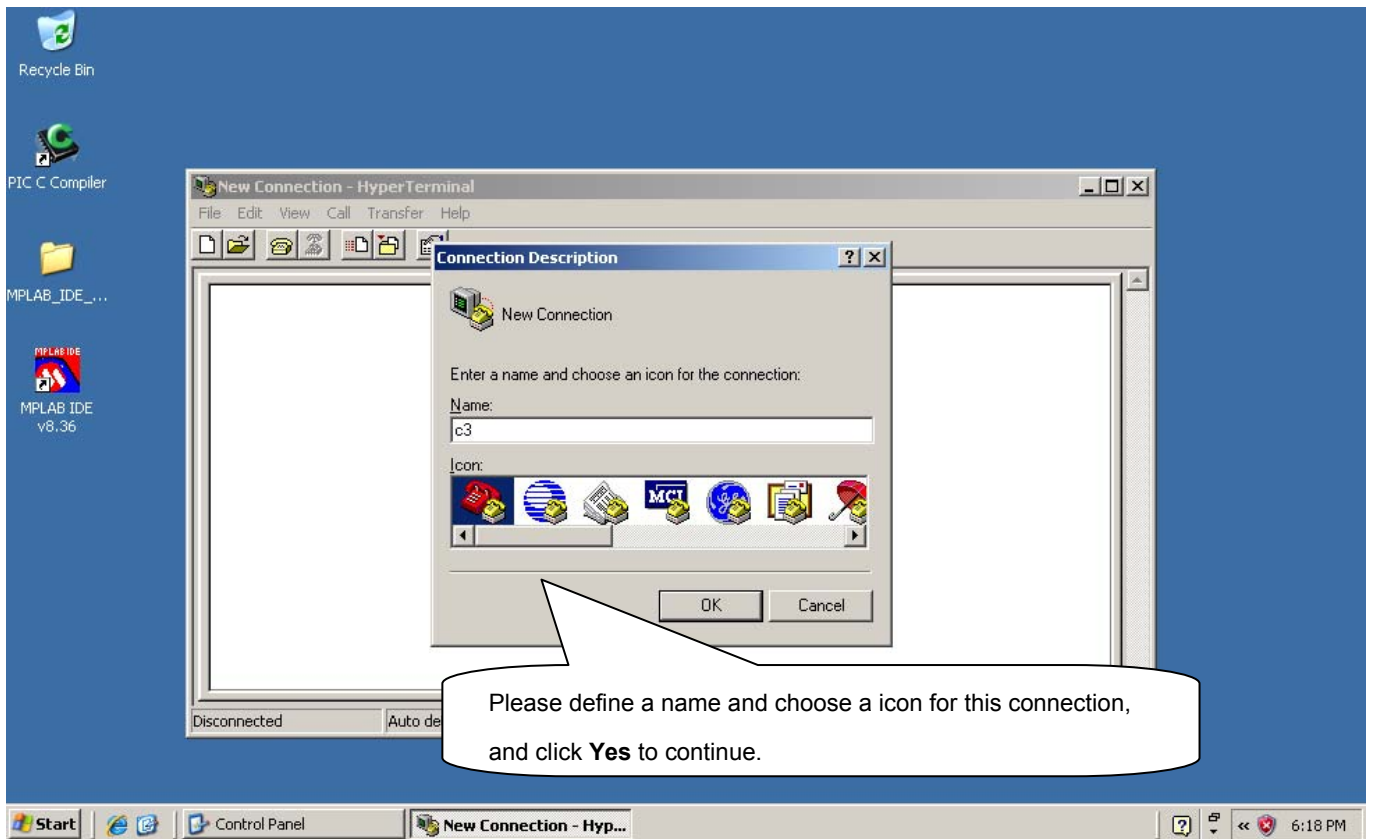
Locations:

Location	Area Code
My Location	02

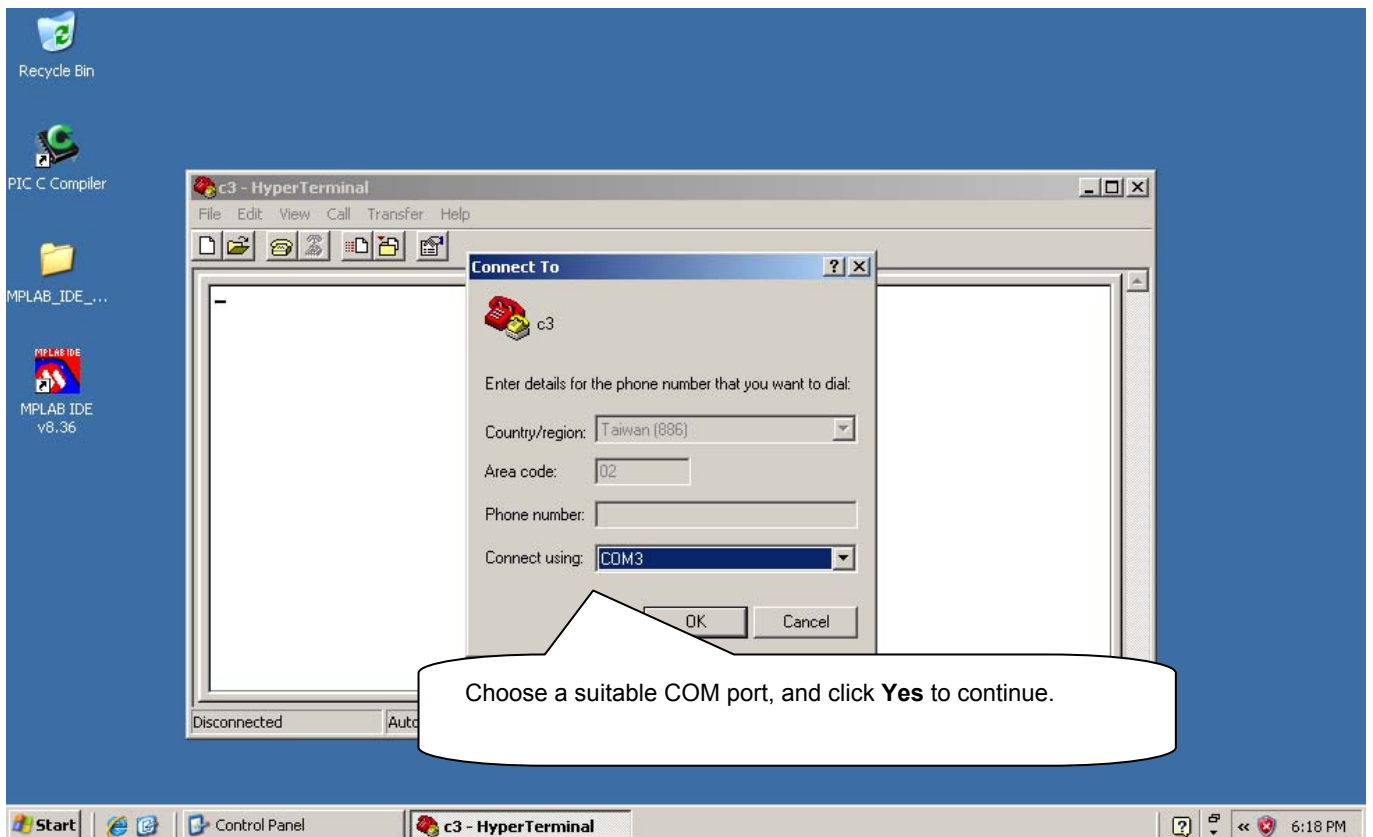
New... Edit... Delete

OK Cancel Apply

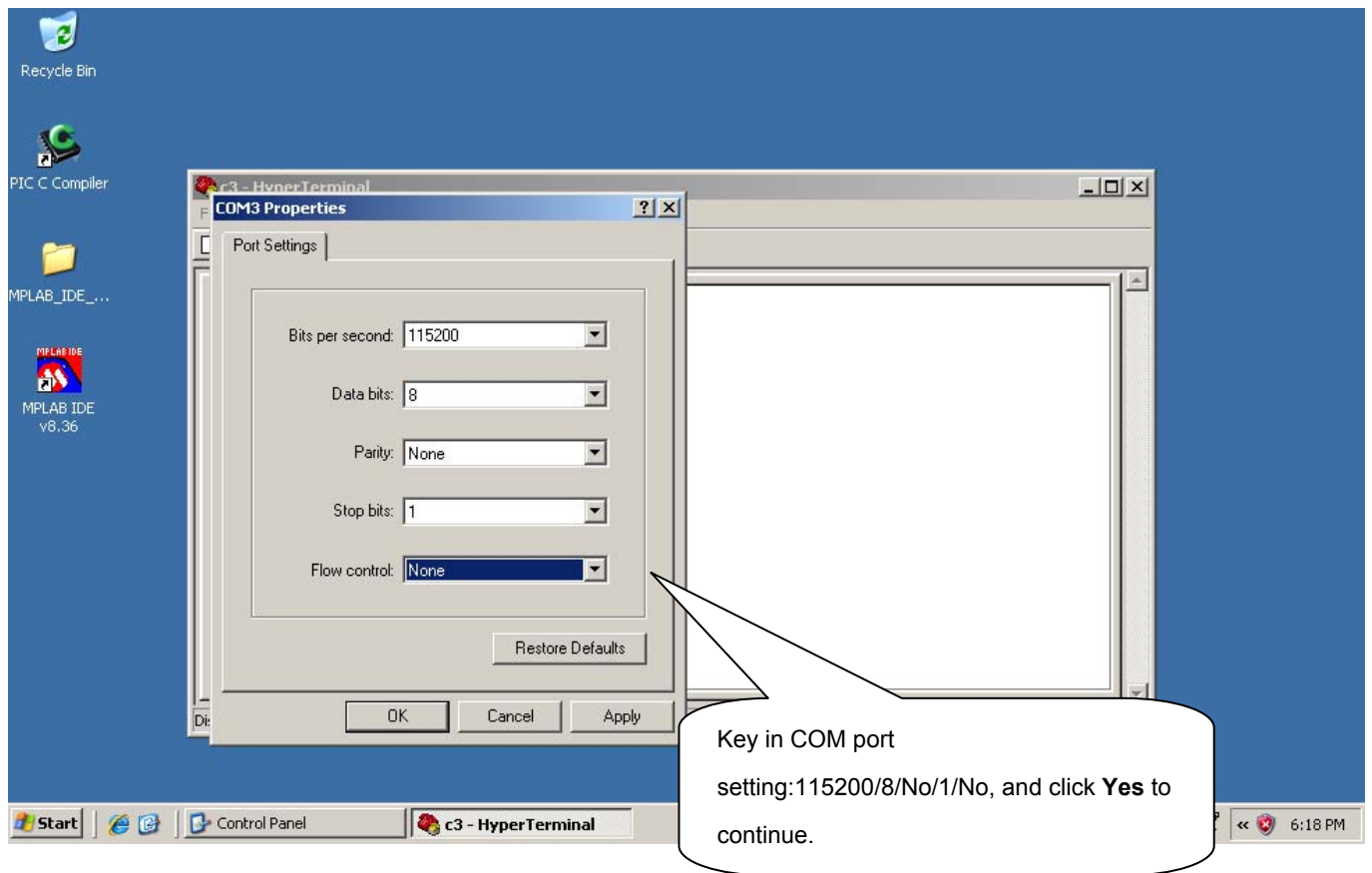
Step 6 - Enter the file name to store the hyper terminal settings. System will auto add a .ht extension name.



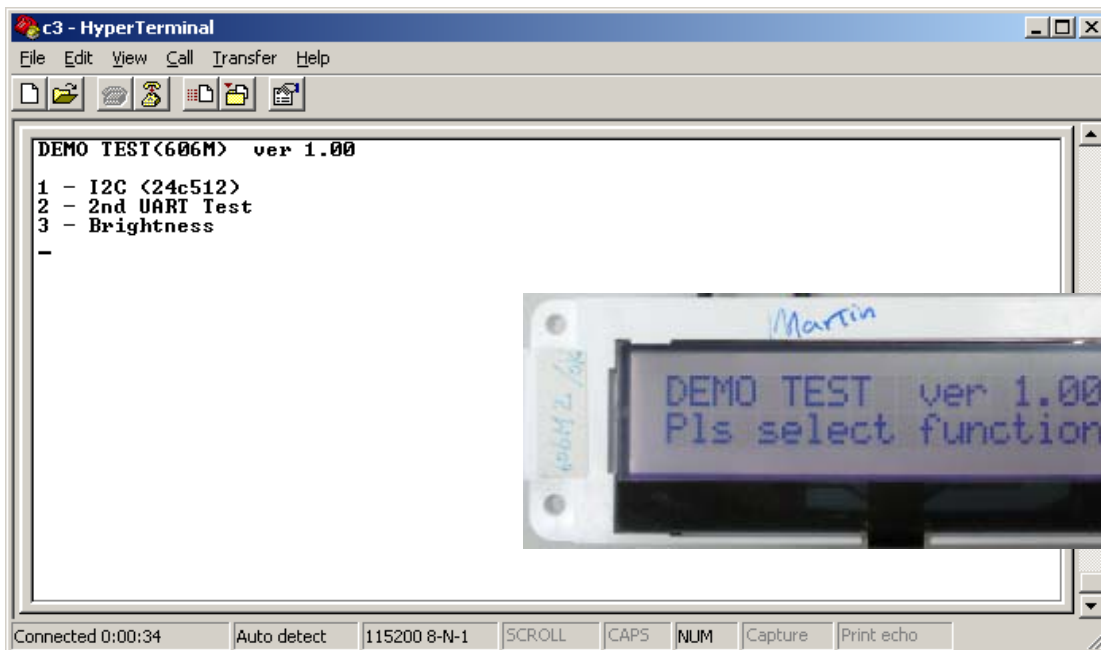
Step 7 -Select **COM** port as appropriate. Hyper terminal will pull down only valid COM ports.



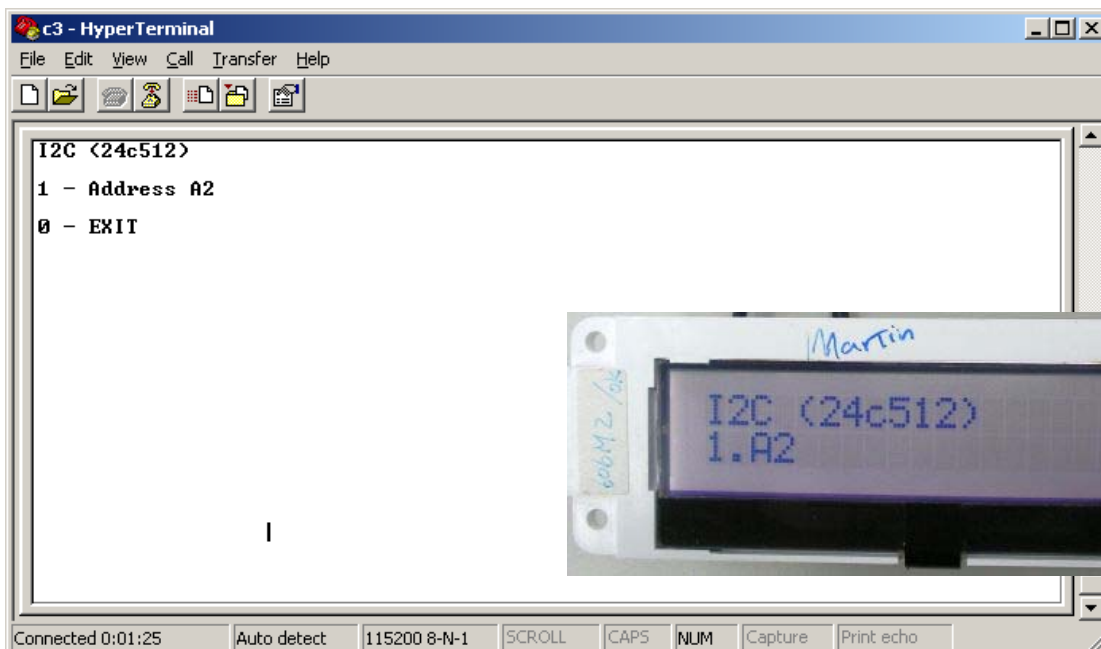
Step 8 - It is required to set the serial communication as follows – 115200 bps, N/8/1/None.



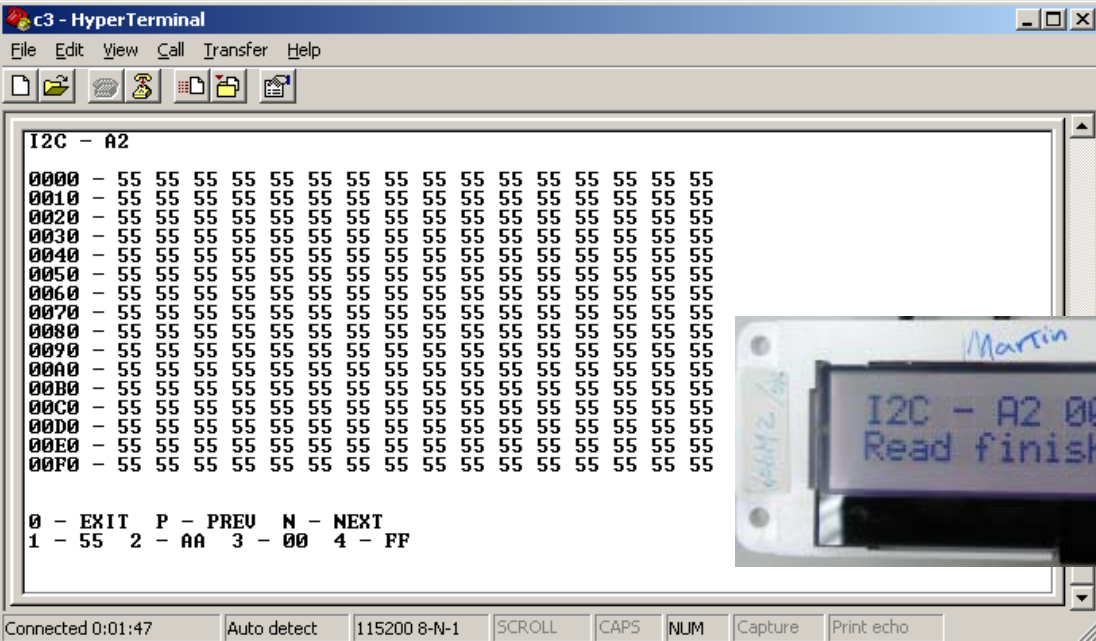
Step 9 - Press **Enter** to bring up top menu.



Step 10 - Press **1** to get to I2C (24c512) sub menu



Step 11 - Press **1** to see memory dump of 1st page of i2c-eprom with a i2c hardware address of \$A2. One page consists of 0xff or 256 bytes.



```

c3 - HyperTerminal
File Edit View Call Transfer Help

I2C - A2
0000 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0010 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0020 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0030 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0040 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0050 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0060 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0070 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0080 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0090 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00A0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00B0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00C0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00D0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00E0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
00F0 - 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55

0 - EXIT  P - PREV  N - NEXT
1 - 55    2 - AA    3 - 00    4 - FF

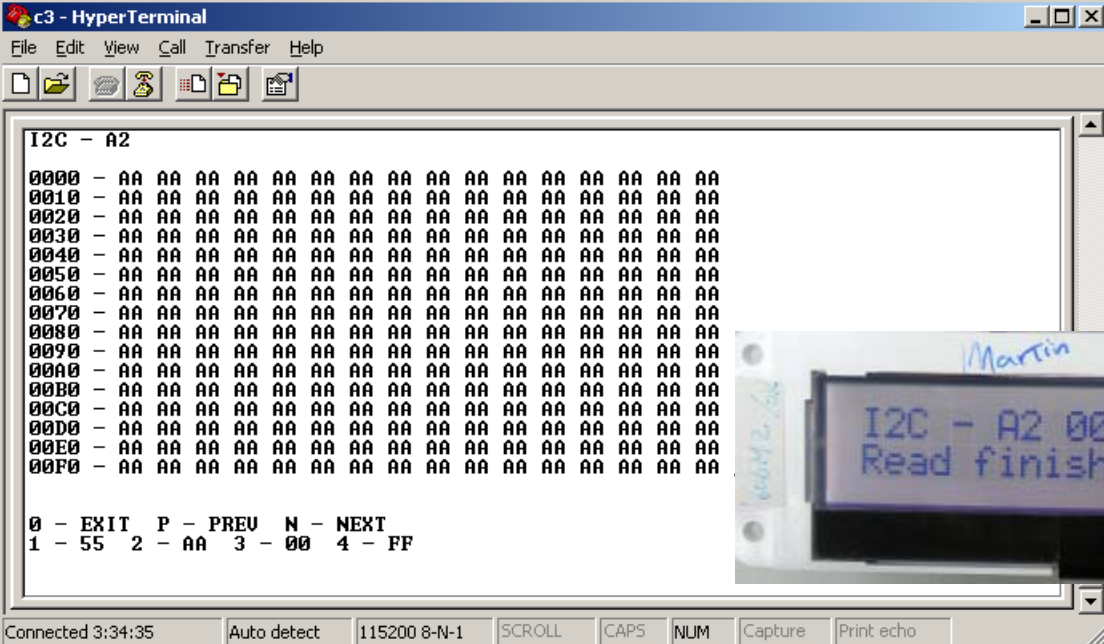
Connected 0:01:47  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo

```



Step 12 - Press **2** to write 0xAA into the whole page. Observe the 2nd line on LCD display will show in the sequence of **writing data**, **reading data**, and **Read finished**.

Press **N** for next page, **P** for previous page to confirm the change made.



```

c3 - HyperTerminal
File Edit View Call Transfer Help

I2C - A2
0000 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0010 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0020 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0030 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0040 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0050 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0060 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0070 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0080 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
0090 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00A0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00B0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00C0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00D0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00E0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
00F0 - AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA

0 - EXIT  P - PREV  N - NEXT
1 - 55    2 - AA    3 - 00    4 - FF

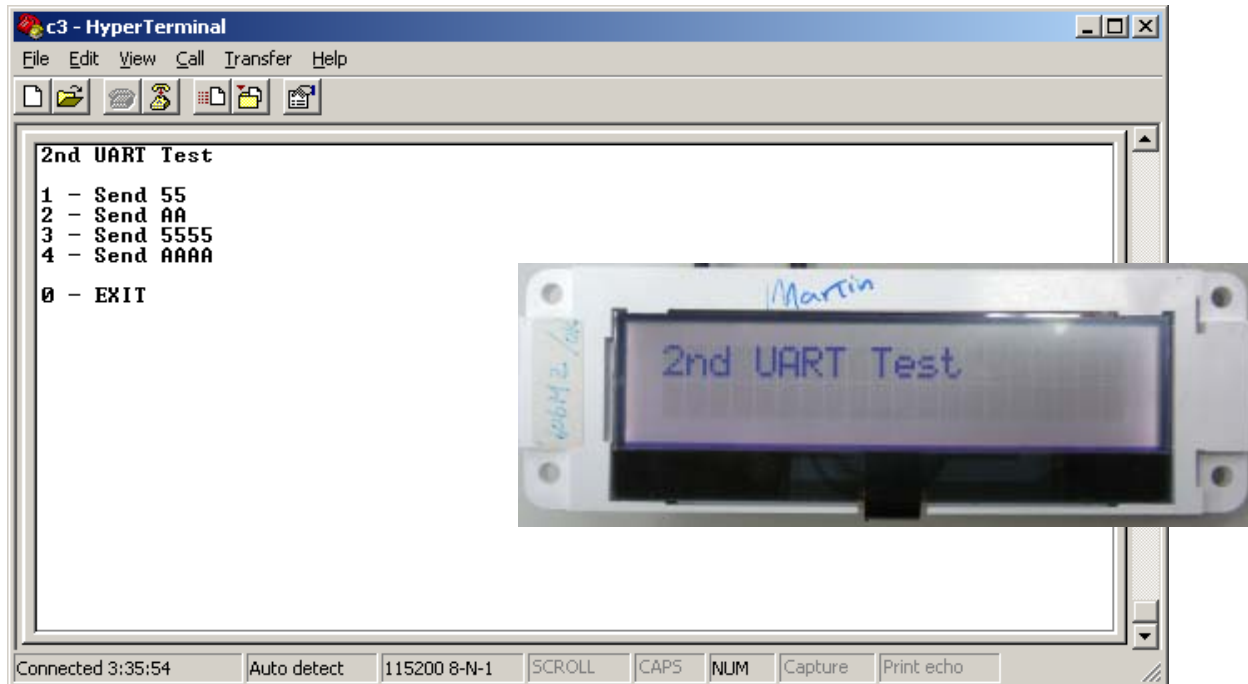
Connected 3:34:35  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo

```

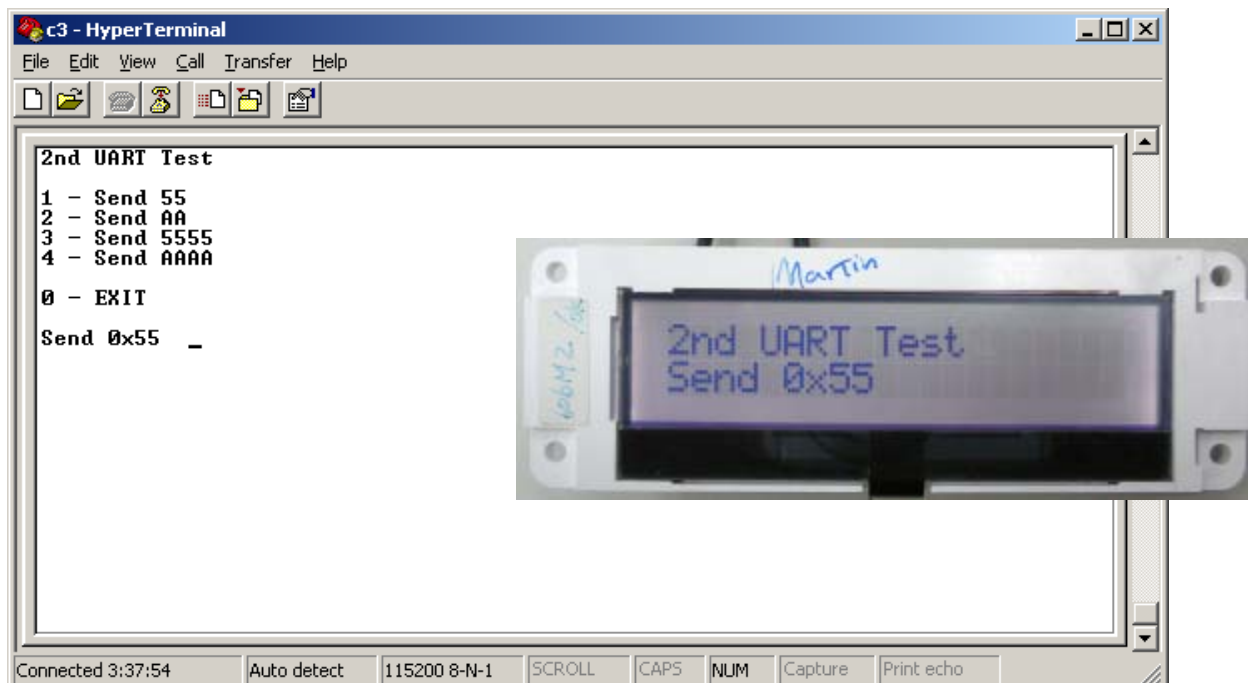


Press **0 0** to return to top menu.

Step 13 - Press **2** to enter 2nd UART Test.



Step 14 - Press **1** to send 1 byte (0x55) via 2nd UART.

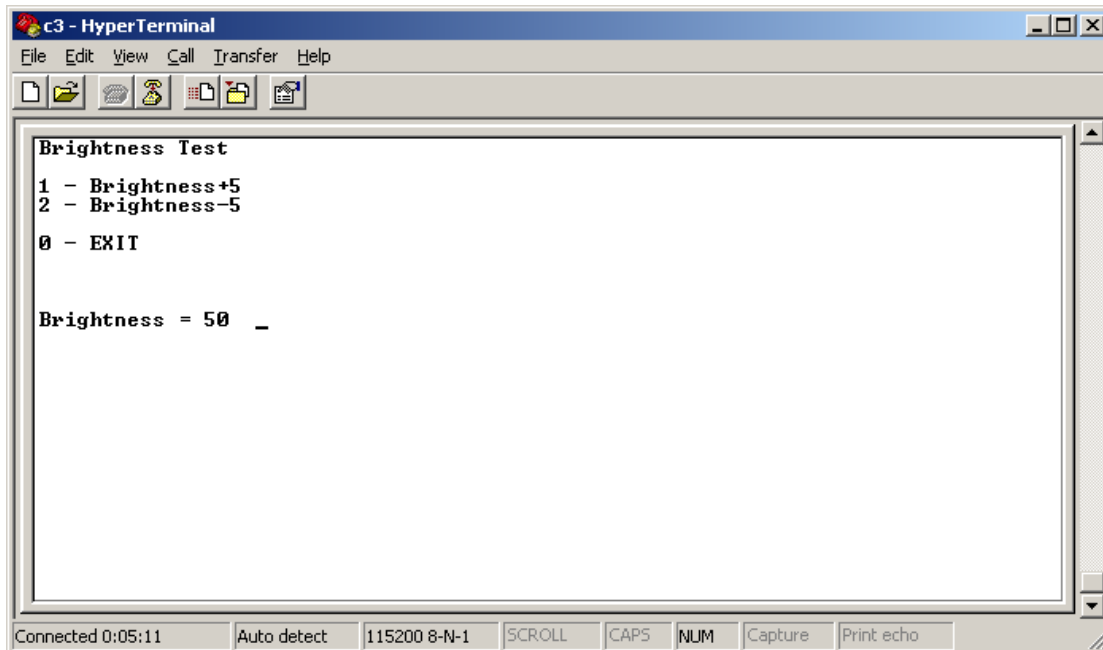


Step 15 - Press **4** to send 2 bytes (0xAAAA) via 2nd UART.



Press **0** to return to top menu.

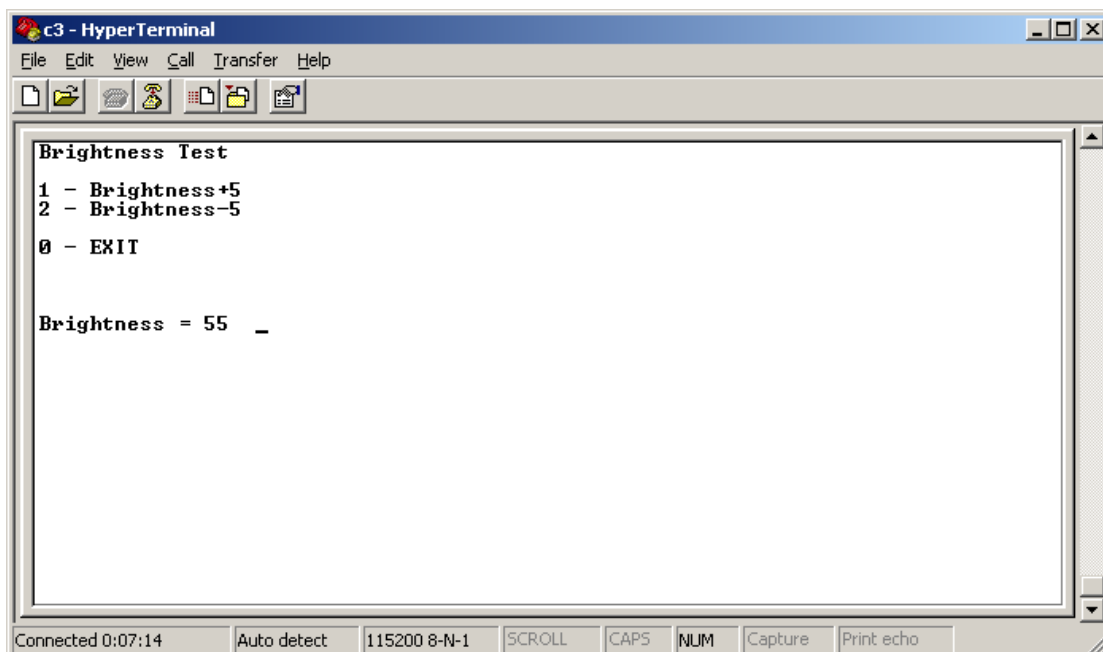
Step 16 - Press **3** to adjust brightness level. Default brightness is set to 50 and each increment/decrement step will +/- 5 to brightness level with a span of 0 to 100.



```
c3 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
Brightness Test
1 - Brightness+5
2 - Brightness-5
0 - EXIT

Brightness = 50 _
Connected 0:05:11  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

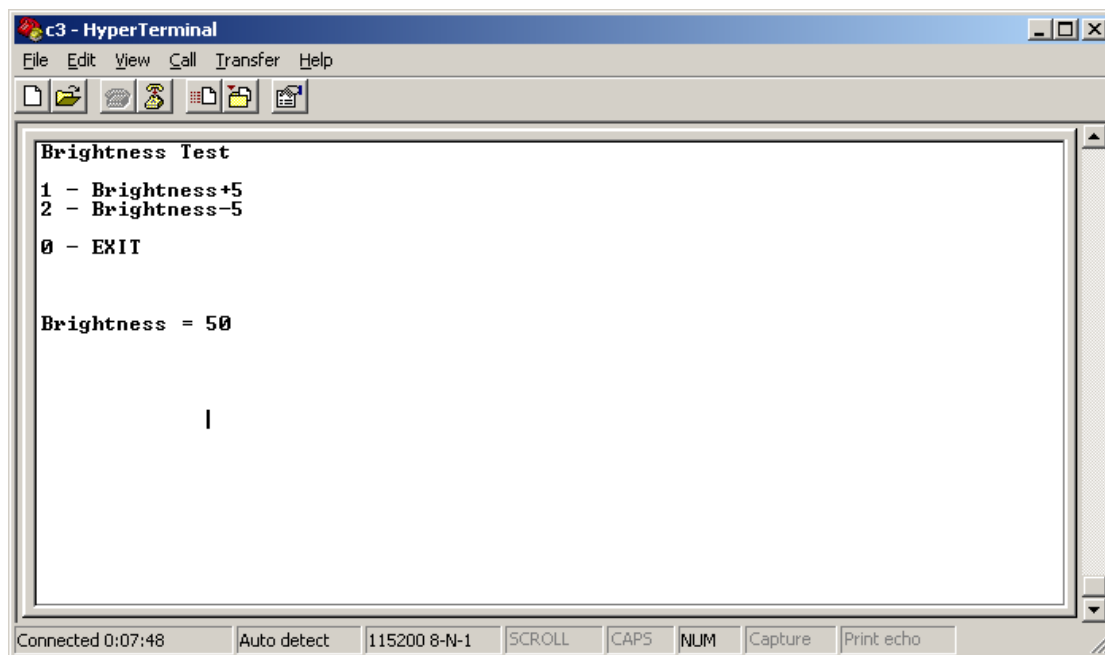
For example, press **1** to increment contrast level by 5 per step..



```
c3 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
Brightness Test
1 - Brightness+5
2 - Brightness-5
0 - EXIT

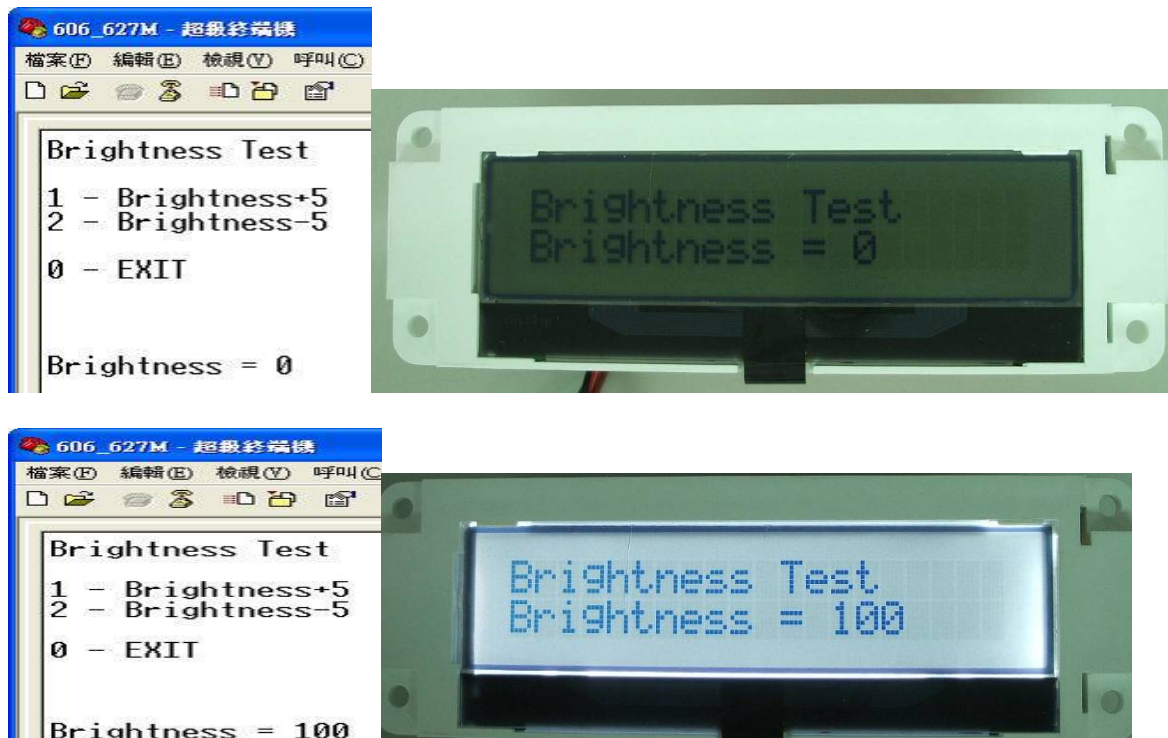
Brightness = 55 _
Connected 0:07:14  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Press **2** to reduce to brightness level.



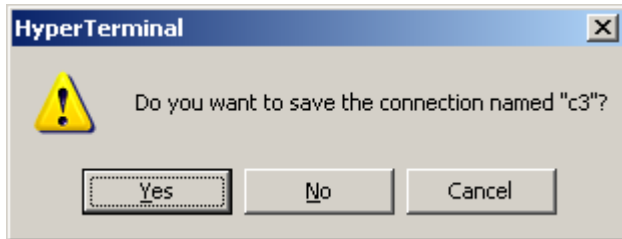
Press **0** to return to top menu.

Here is the result of bright level of 0 and 100.



We just complete all functions of demo programs.

Step 17 – Close hyper terminal .



Click **yes** to save into **c3.ht** the serial communication protocol (com3,115200bps, N/8/1). Next time open the hyperterm, simply click **file-open** and select **c3.ht**.

Chapter 5 Software Reference Manual

Abstract

This chapter introduces all functions of device driver library for BEGV606M device and describes major issues about application program development.

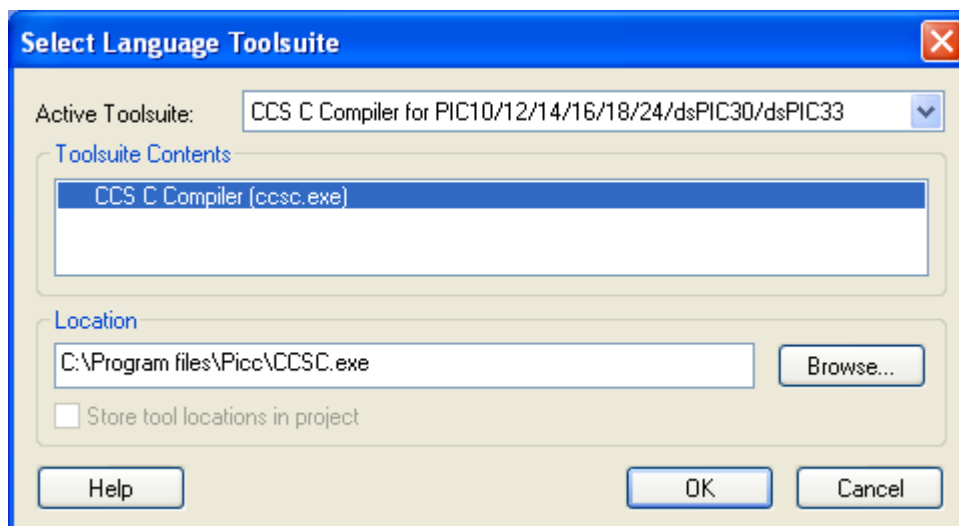
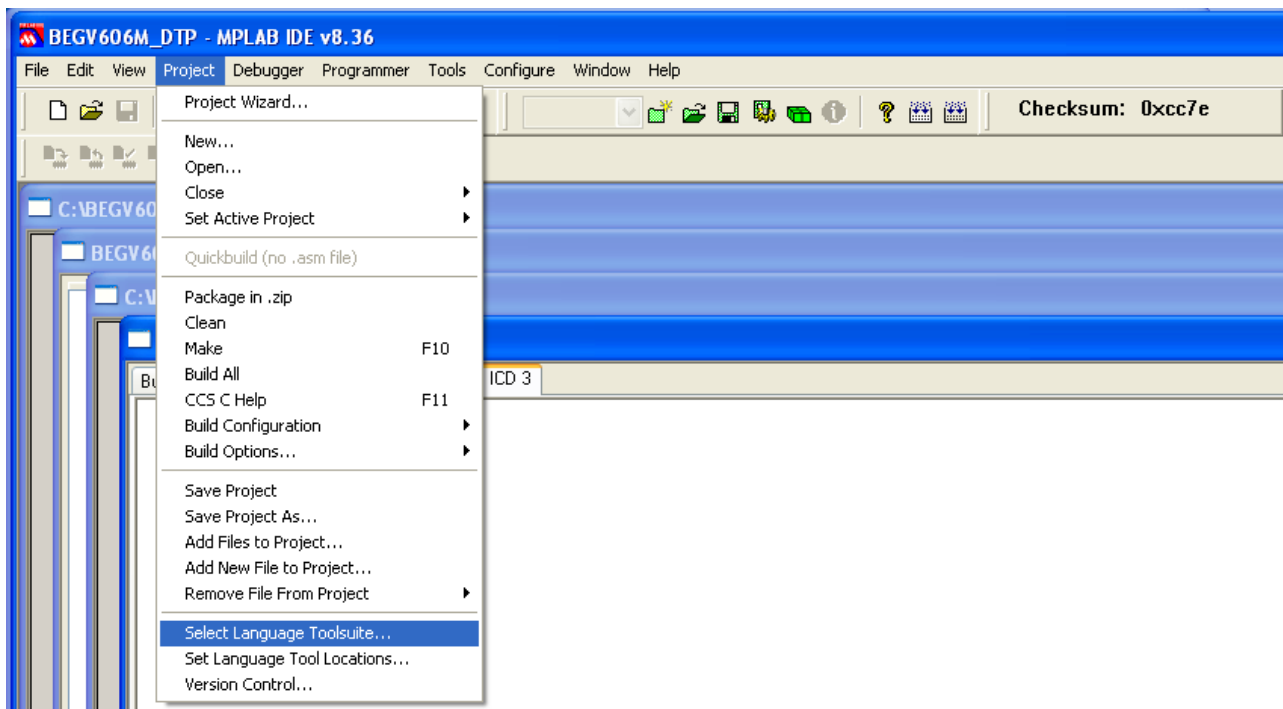
5.1 Setup development environment

Bolymin recommends to use the Integrated Development Environment (IDE) provided by Microchip, MPLAB IDE v8.36, and CCS C compiler v4.093 to develop new application software. Here is the step guide to set up the development environment:

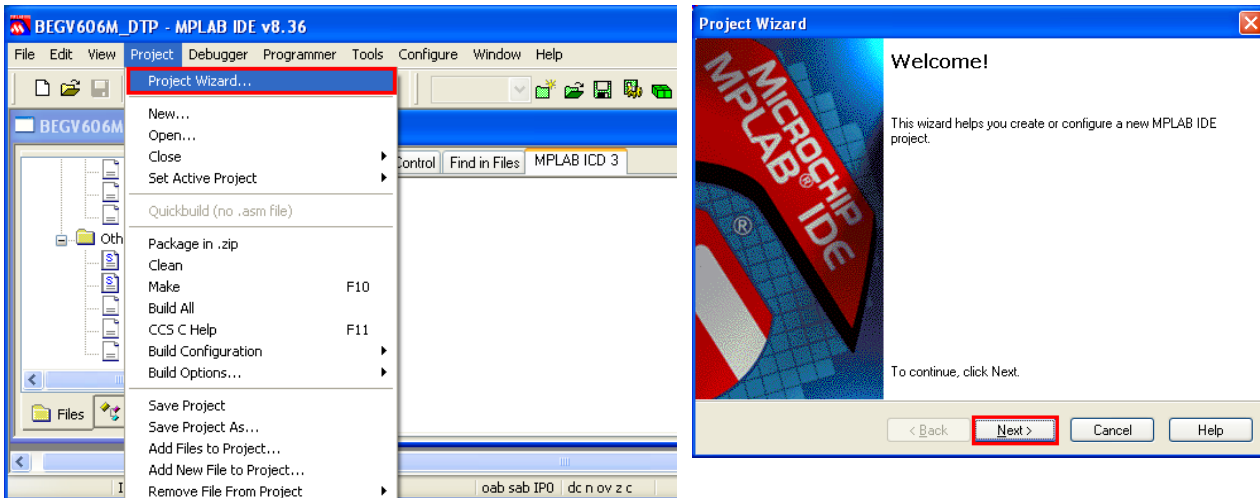
Step 1 - Install CCS C compiler v4.093 – refer to 4-1

Step 2 - Install MPLAB IDE v8.36 – refer to 4-2

Step 3 - Open MPLAB 8.36, then select **Project-Set Language Tool Locations...** to confirm default compiler is **CCS C v4.093** compiler. User may not need to change the setting if she follows chapter 4.1 and 4.2 installation procedure since installation program will automatically configure accordingly.



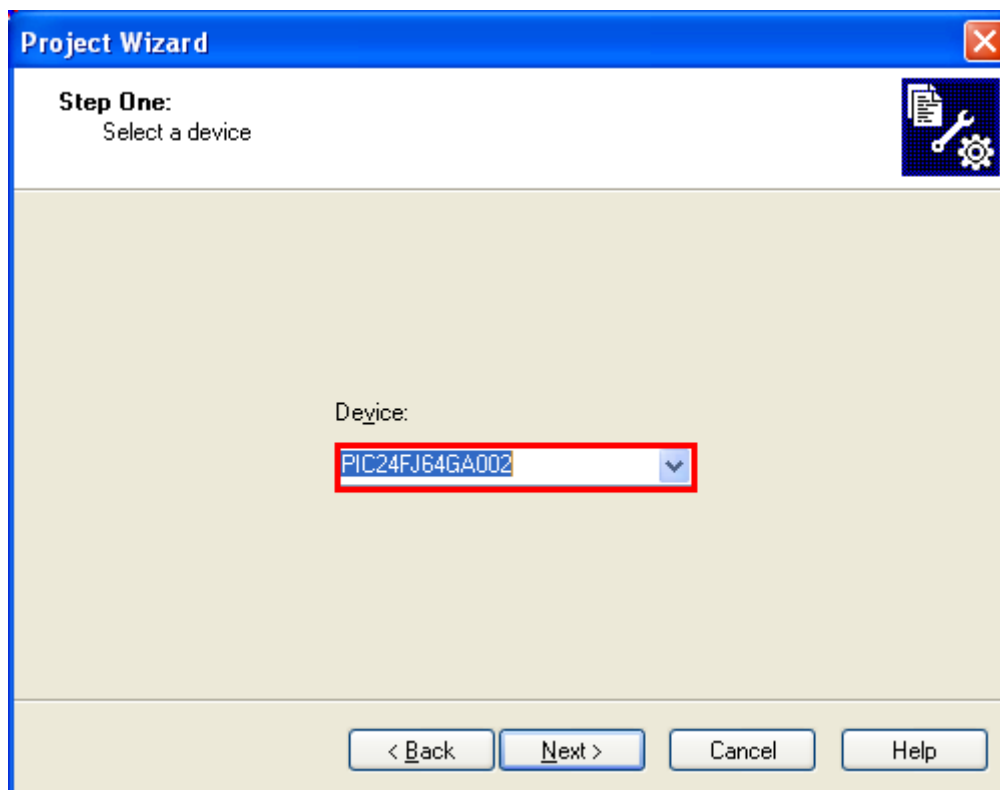
Step 4 - Select Project-Project Wizard... to create a new project and start development. Click **Next** to skip the Welcome screen.



NOTE:

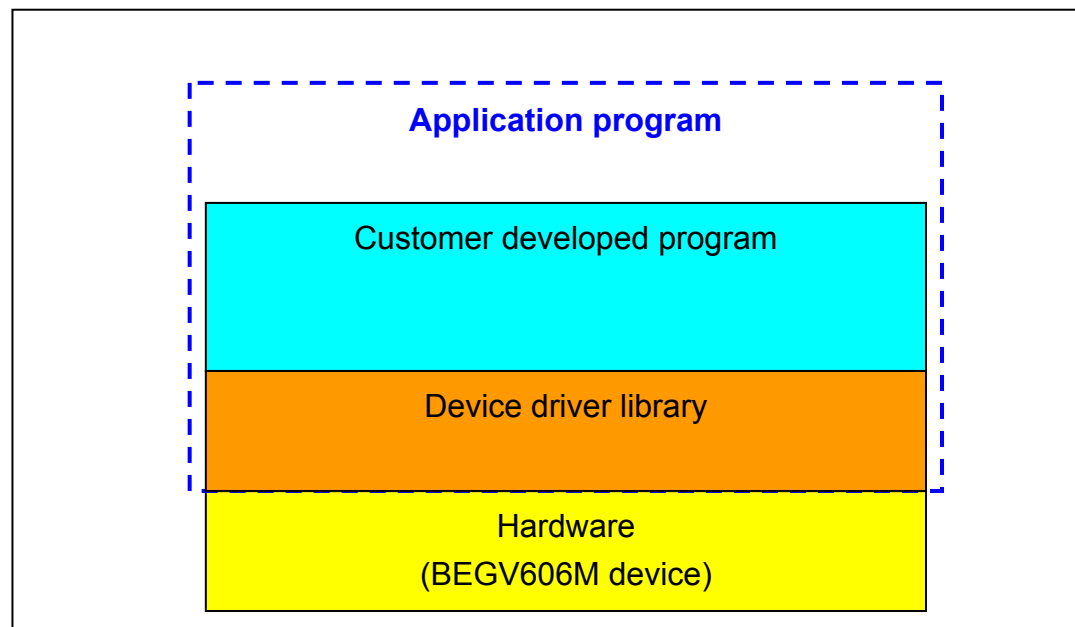
1. After creating a project, user need to create an empty file by the name of project in the directory of project so that the HEX file could be created successfully. For example, Project DTP_606M needs a file named **DTP_606M** in the same directory.
2. Please include **config.h** for all *.C file.
3. We need to select device type of MCU while creating new project at the screen shown as follows:

Step 5 - For BEGV606M , select PIC24FJ64GA002 as device.



5.2 Introduction of library

Bolymin supports device driver library for application development of BEGV606M. Refer to the following figure for program hierarchy.



5.2.1 Summary of modules of library

BEGV606M driver library consists of 5 modules. There are several functions for each module. Following table is a summary of modules in library:

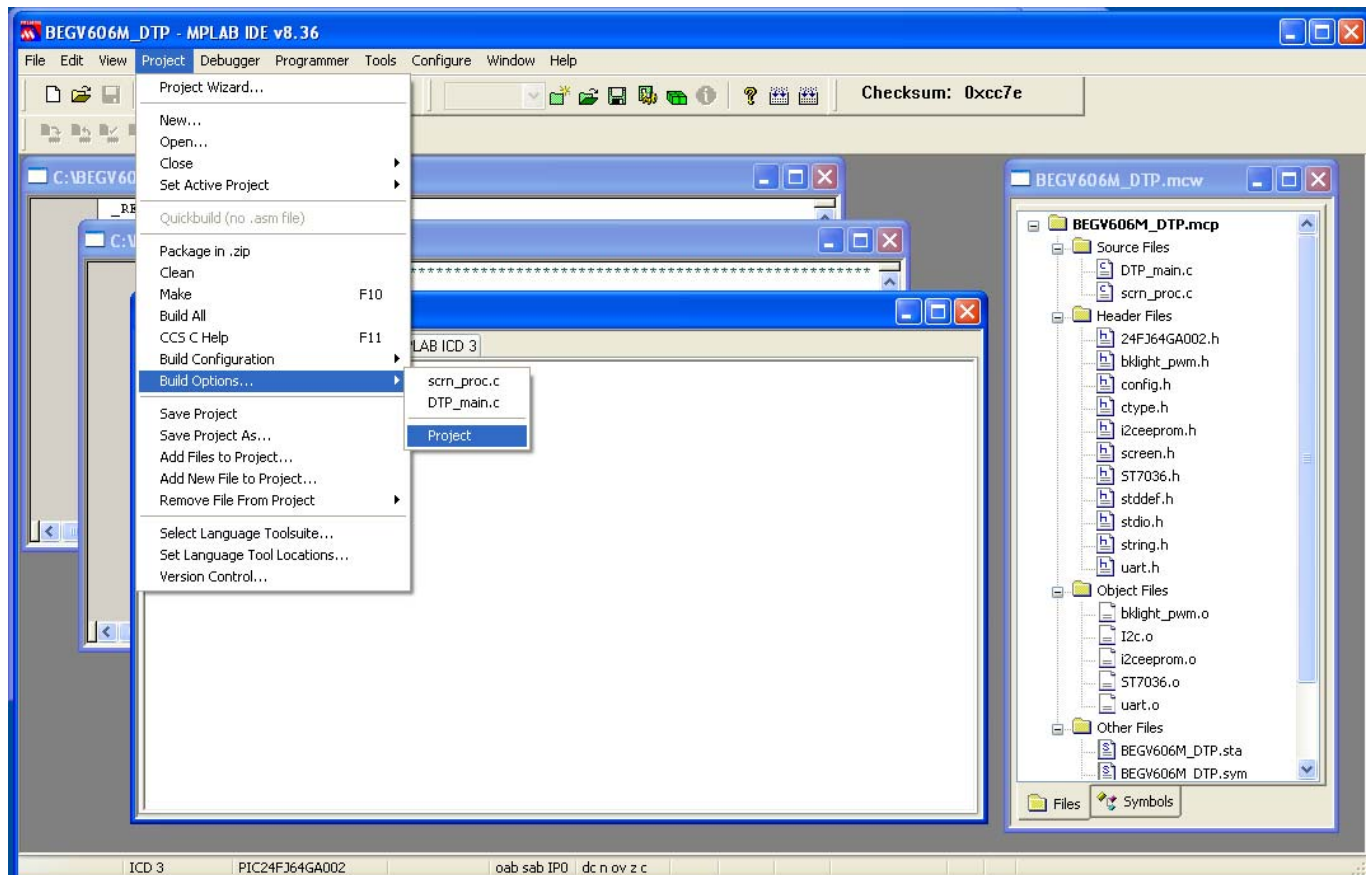
Module name	Description	Header file and compiled object
UART	Include all functions about operation of UART port.	uart.h uart.o
I ² C eeprom	Read and write functions for I ² C eeprom.	I2c.h I2c.o I2ceeprom.h I2ceeprom.o
LCD display	LCD display functions	St7036.h ST7036.o
PWM for backlight	Set or get current PWM value to adjust backlight of LCD.	bklight_pwm.o bklight_pwm.h

5.2.2 How to use library in application program

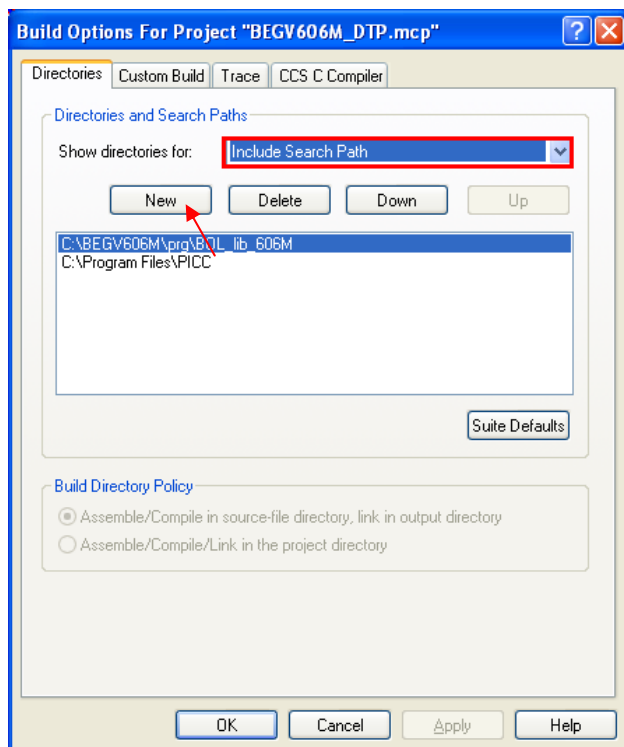
Please follow below steps to use device library provided by Bolymin:

Step1 –Refer to 4.2 and 4.3 for Bolymin library and demo code installation. Follow 4.4 to step 4.

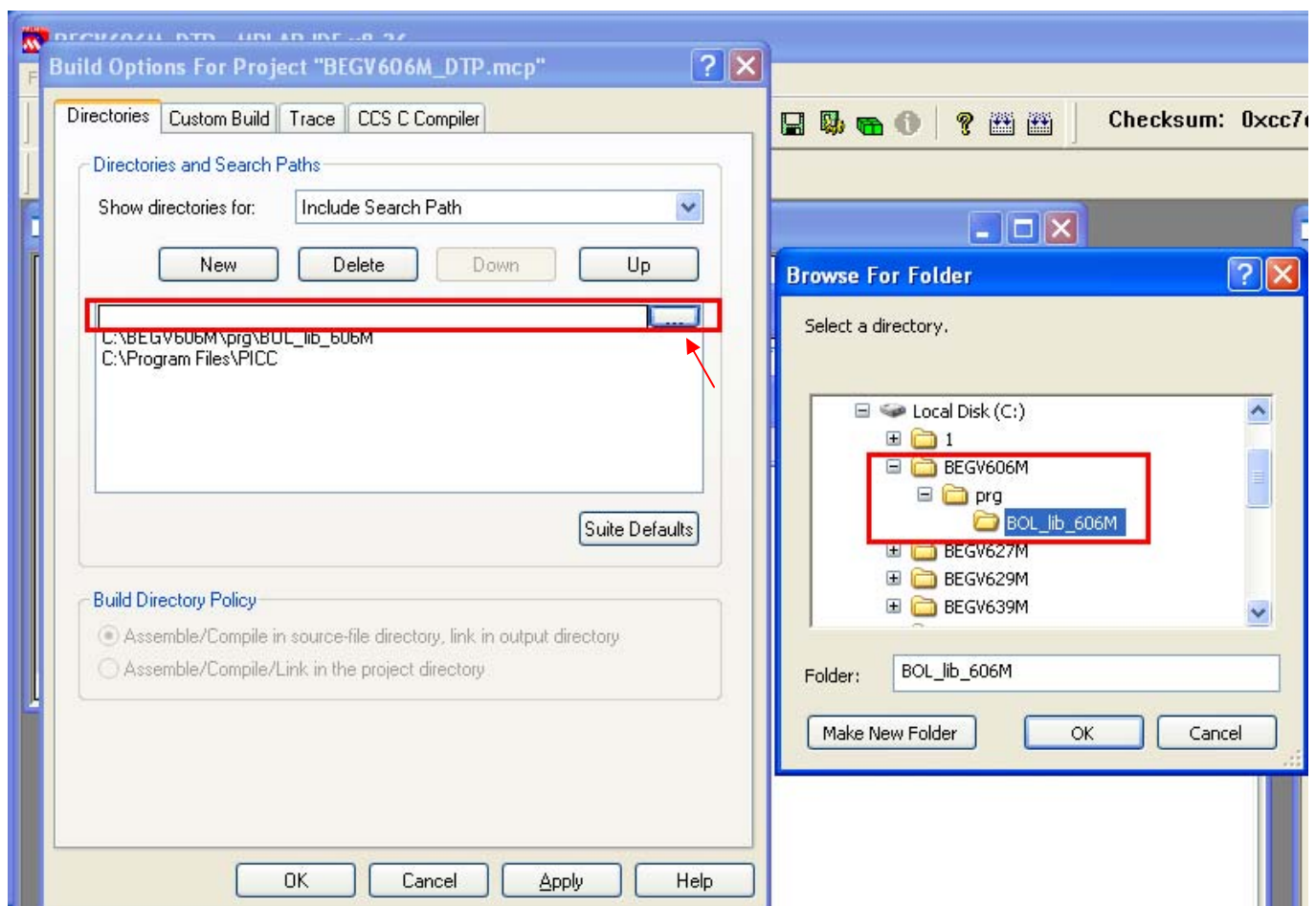
Step 2.- Select Project-Build Options...-Project.



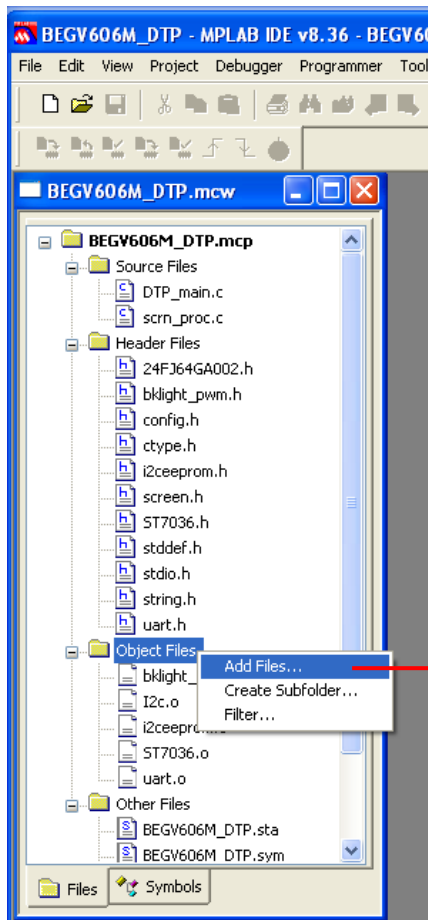
Step 3 – Set Include Search Path to the path of header files of library provided by Bolymin.



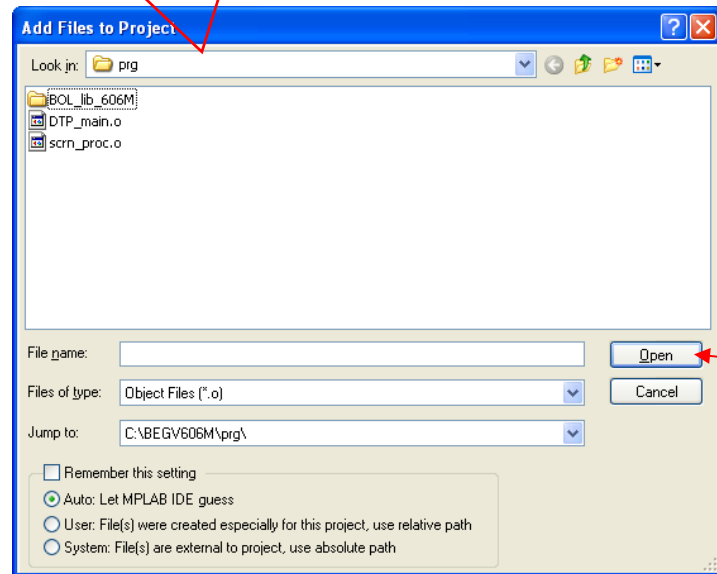
Click on **New** to bring up a new row and a folder browse button. Click on browse button to bring up a **Browse For Folder** dialog windows to add it to include search path.



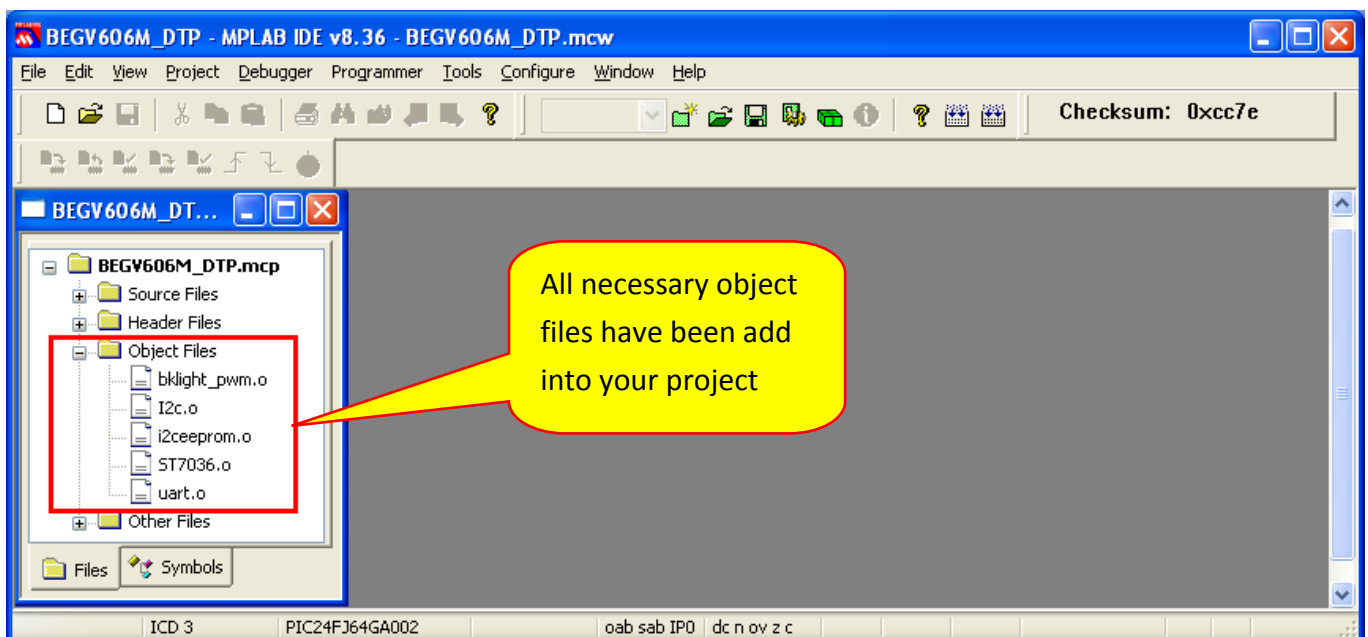
Step 4 - Click **Object File** of mcw window shown below by right-clicking mouse and select **Add Files...** to bring up **Add Files to Project** windows. Click on the .o object file and click **open** button to add necessary object files into your project.



Move into the directory of object files of device library and select necessary object files.



Step 5 - After clicking **Open** button of **Add files to Project** dialog, you can see all necessary object files have been add into your project, like below picture. Now you can start to develop your application program.



5.3 Function Primitives

5.3.1 UART function

Header file : uart.h

Object file : uart.o

uartInit Function: Initial UART.

Syntax	<pre>void uartInit(uint8_t byPort, uint32_t uBaudrate, uint8_t byParity, uint8_t uDatabit, uint8_t uStopbit, uint8_t nTxMode);</pre>	
Parameters	byPort nBaudrat byParity uDatabit uStopbit nTxMode	UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port Baud rate, ex: 9600. Parity Check, 'N' – None, 'E' – EVEN, 'O' - ODD. data bit , 5 ~ 8. data bit , 1 ~ 2. Transmission Mode. 0 or FALSE – RS232. 1 or TRUE – RS485 or RS422.
Return value	None.	

uartSetBaudRate Function: Set up baud rate for assigned UART port

Syntax	<pre>void uartSetBaudRate(uint8_t byPort, uint32_t uBaudrate);</pre>	
Parameters	byPort nBaudrate	UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port Baud Rate, ex: 9600.
Return value	None.	

uartSendByte Function: Send 1 byte from assigned UART port

Syntax	<pre>void uartSendByte(uint8_t byPort, uint8_t txData);</pre>	
Parameters	byPort txData	UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port Byte to be sent.
Return value	None.	

uartSendString Function: Send string from assigned UART port

Syntax	void uartSendString(uint8_t byPort uint8_t* str);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port str Memory pointer of string to be sent, ending with 0.
Return value	None.

uartSendBuffer Function: Send buffer from assigned UART port

Syntax	void uartSendBuffer(uint8_t byPort, uint8_t* buffer, uint16_t nBytes);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port buffer Memory pointer of data to be sent nBytes Bytes of data to be sent
Return value	None.

uartReceiveByte Function: Read 1 byte data from assigned UART port

Syntax	uint8_t uartReceiveByte(uint8_t byPort, uint8_t* rxData);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port rxData buffer used to put received data
Return value	TRUE – UART port receive data and put in rxData FALSE – There is no data in assigned UART port.

uartReceiveBufferIsEmpty Function: Check if the receive buffer of assigned UART port is empty or not

Syntax	void uartReceiveBufferIsEmpty(uint8_t byPort);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port
Return value	TRUE – The receive buffer of assigned UART port is empty. FALSE – There is data in receive buffer of assigned UART port

uartFlushReceiveBuffer Function: Clear receive buffer of assigned UART port

Syntax	void uartFlushReceiveBuffer(uint8_t byPort);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port
Return value	None.

5.3.2 I²C eeprom function

Header file : i2ceeprom.h
object file : i2c.o, i2ceeprom.o

i2cInitial Function: Initial I²C functions. User should call this function before using I²C functions.

Syntax	void i2cInitial();
Parameters	None.
Return value	None.

i2cReadByte Function: Read 1 byte data from I²C eeprom.

Syntax	uint8_t i2cReadByte(uint8_t uDevAddr, int16 nAddr);
Parameters	uDevAddr I ² C device address. (address of eeprom on board are: A2 _{hex} , A4 _{hex} . nAddr Address will be read.
Return value	Data reading from I ² C eeprom.

i2cWriteByte Function: Write 1 byte data to I²C eeprom.

Syntax	void i2cWriteByte(uint8_t uDevAddr, int16 nAddr, uint8_t byData);
Parameters	uDevAddr I ² C device address.(address of eeprom on board are: A2 _{hex} , A4 _{hex} . nAddr Address to write in . byData Data to be written.
Return value	None.

i2cIsEEPROMExist Function: Check if the specified address eeprom is available or not.

Syntax	uint8_t i2cIsEEPROMExist (uint8_t uDevAddr,);
Parameters	uDevAddr I ² C device address.(address of eeprom on board are: A2 _{hex} , A4 _{hex} .
Return value	TRUE: Eeprom with uDevAddr address is available. FALSE: Eeprom with uDevAddr address is not available.

5.3.3 LCD control function

Device	BEGV606M
Header file	ST7036.h
Object file	ST7036.o

LcdInit Function: Initialize all parameters of LCD display. User should call this function before use functions of LCD display.

Syntax	void LcdInit ();
Parameters	None.
Return value	None.

LcdDisplayClr Function: Clear screen.

Syntax	LcdDisplayClr();
Parameters	None.
Return value	None.

LcdSetCursorPos Function: Set the position of cursor. (UNIT=character=8*8 pixel)

Syntax	void LcdSetCursorPos(uint8_t x, uint8_t y);
Parameters	x X coordinate of cursor y Y coordinate of cursor
Return value	None.

LcdPrintString Function: Print input string to specified location of text layer.

Syntax	void LcdPrintString (uint8_t x_start, uint8_t y_start, char * string, uint8_t str_count);
Parameters	x_start X coordinate of start location that input string will be printed. (UNIT= character=8*8 pixel) y_start Y coordinate of start location that input string will be printed. (UNIT= character=8*8 pixel) string string will be printed to LCD str_count character count of input string
Return value	None.

5.3.4 Backlight PWM control function

Header file : bklight_pwm.h
object file : bklight_pwm.o

bklPWM_Init Function: Initialize all parameters of backlight PWM control function. User should call this function before use backlight PWM control functions.

Syntax	void bklPWM_Init ();
Parameters	None.
Return value	None.

bklSetBrightness Function: Set current brightness value of backlight.

Syntax	void bklSetBrightness (int8_t brightness);
Parameters	Brightness New brightness value 0 – OFF backlight 1~100 – Control the brightness of backlight
Return value	None.

bklGetBrightness Function: Get current brightness value of backlight

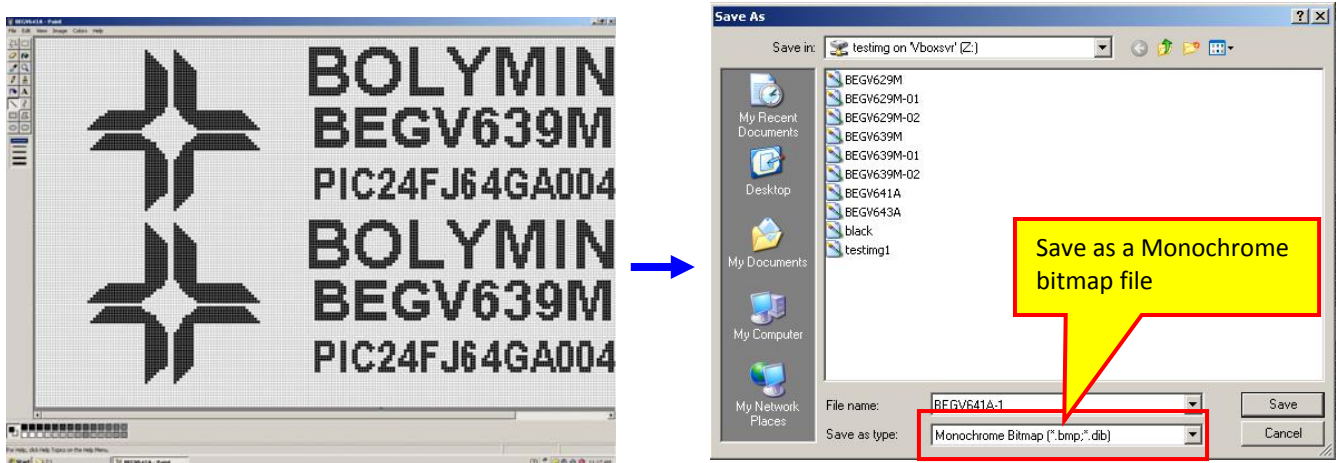
Syntax	uint8_t bklGetBrightness ();
Parameters	None.
Return value	Current brightness value of backlight. (0 ~ 100)

Appendix A - Simple operation guide of BTImg2LCD program

Requires Windows XP or Win7 with .Net framework 3.5 or later

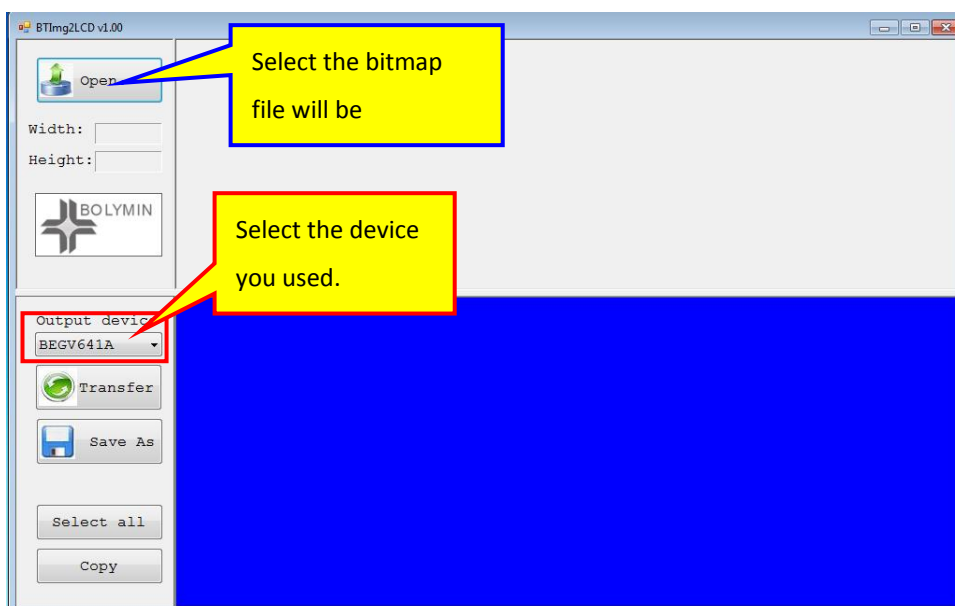
BTImg2LCD is a tool program provided by Bolymin to transfer a bitmap file to source code which could be used in user's program. This document will describe the operation steps of BTImg2LCD program.

Step 1 - User needs to create a monochrome bitmap file (no grayscale) by drawing software, such as **msPaint** before running BTImg2LCD program. **It is recommended that the width and height of the picture to be a multiple of 8.**



Hint to turn grid on: On msPaint, click **view-zoom-larger size**, then click **view-zoom-show grid** or press Ctrl-G to toggle grid.

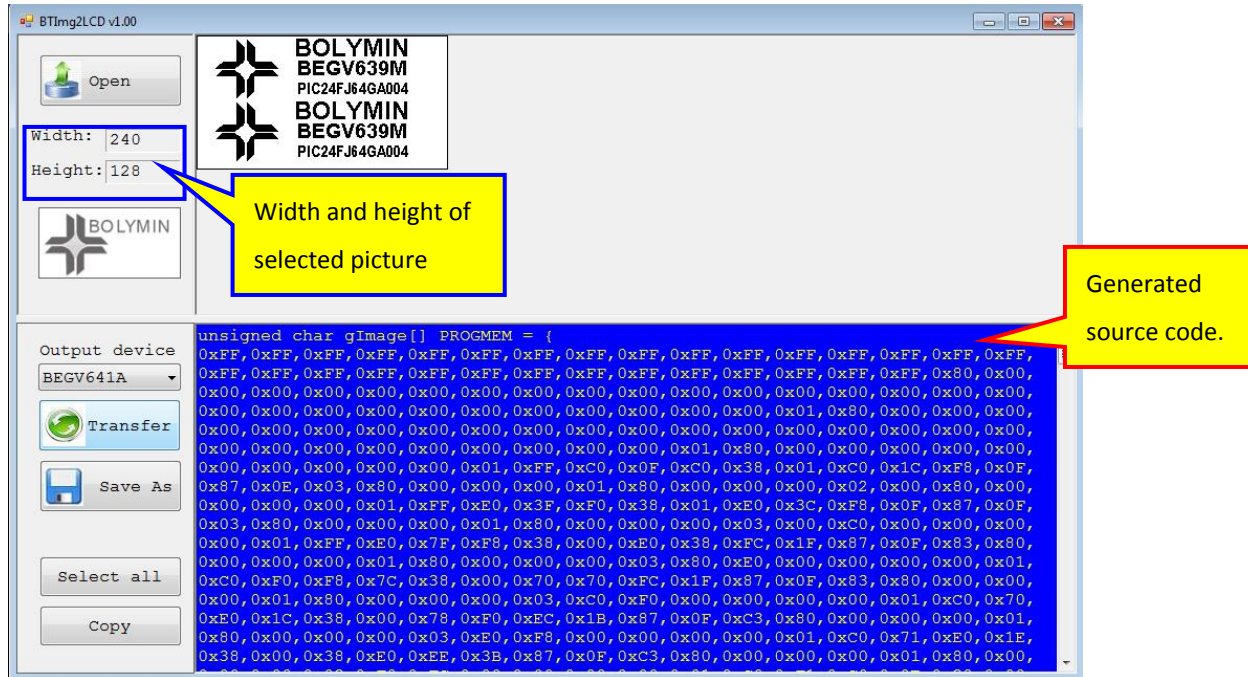
Step 2 - Double click  to run **BTImg2LCD.exe**. Here is the main screen .:



Step 3 - Click  button to select the bitmap file you created.

Step 4 - Select the device you used from the “output device” list.

Step 5 - Click  button to transfer the selected bitmap file to source code. All generated source code will be added into the editor as following picture shows.



Step 6 - User may change the generated source code in the editor as appropriate.

There are 2 methods to use the generated source code in your program:

Method 1: Save the source code as a C file by  button then include this C file by “#include” statement in the source code of your program. For example: #include “demo_pic.c”.

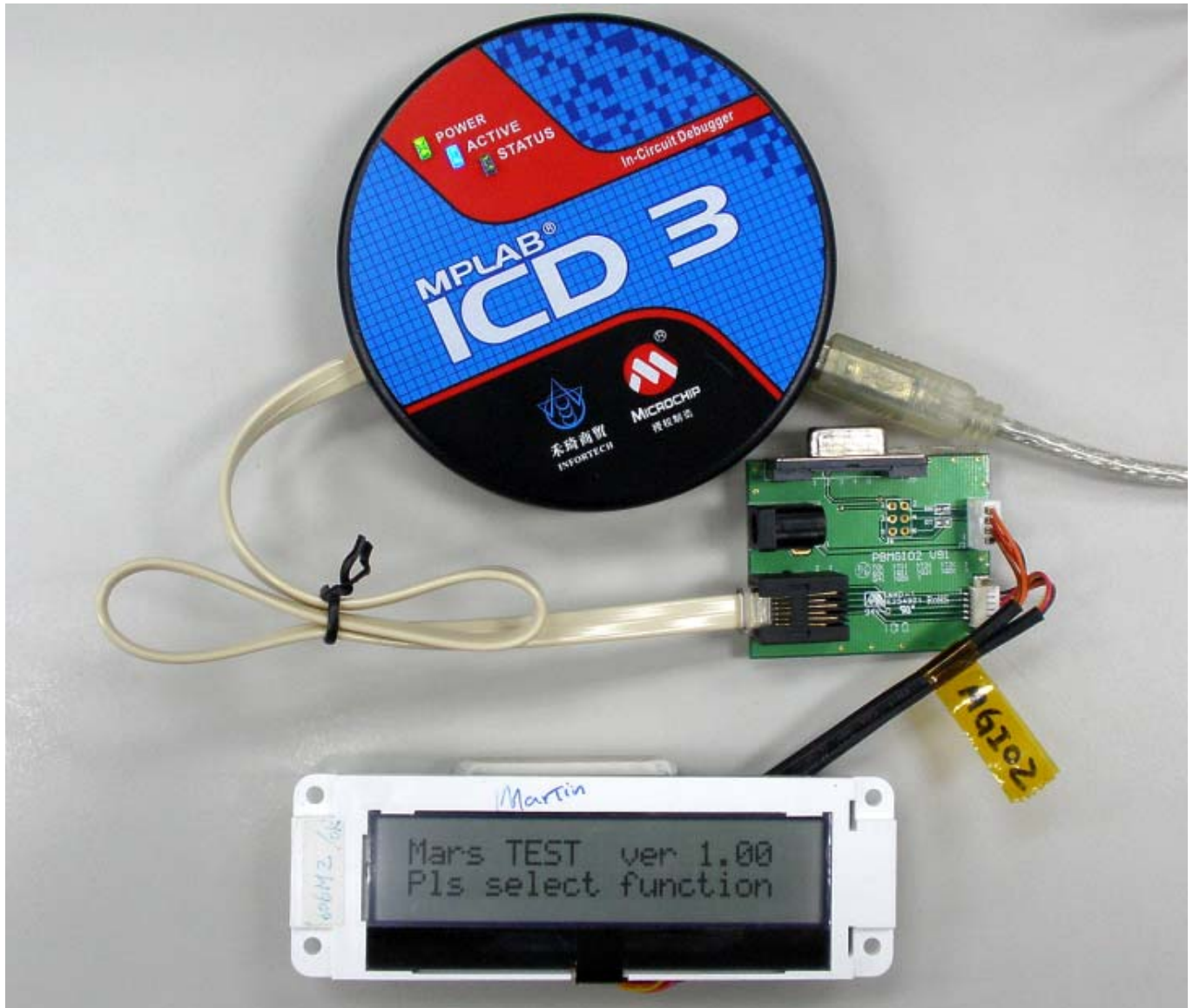
Method 2: Copy all source code into system clipboard by **Select all** and **Copy** buttons then paste it into your code.

Appendix B - Important Notice

We recommend using CCS compiler **ver4.093** and MPLAB IDE **ver8.36**.

The transfer board+ cable for BEGV606M is **MGI02-0\$** - Bolymin proprietary.

To operate our BEGV606M, the customer will need MPLAB **ICD3** and MGI02-0\$.



Appendix C - Notes when running under Windows 7

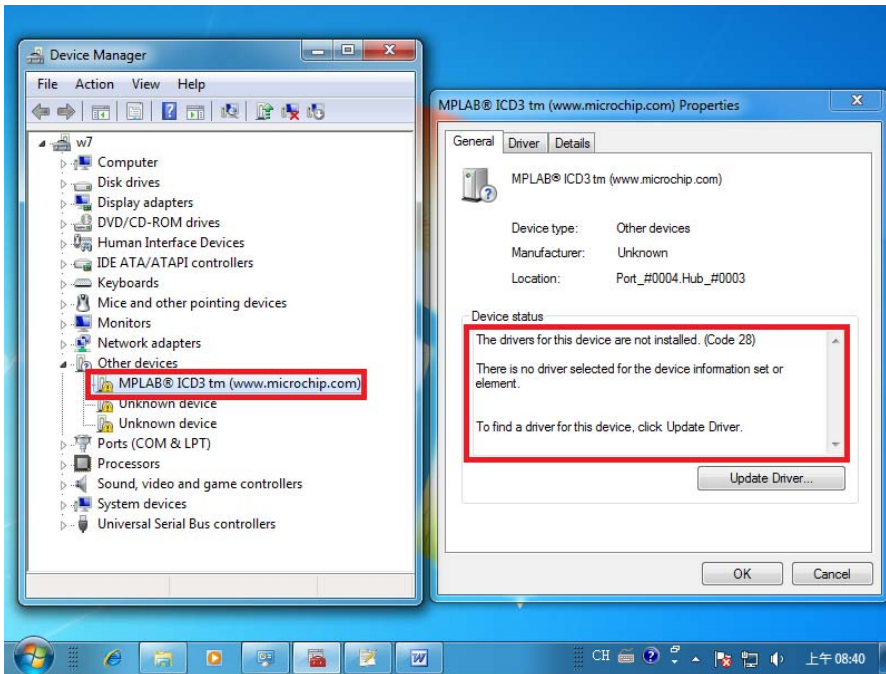
1. User may notice the ICD3 status LED blinks when running under windows 7 and the OS does not recognize ICD3 as a valid USB device. To fix the problem, install ICD3 from the following folder –
for 32bit OS, find driver under **c:\Program Files\Microchip\MPLAB IDE\ICD3\drivers**
for 64bit OS, find driver under **c:\Program Files\Microchip\MPLAB IDE\ICD3\VistaXP64**
2. There is no native hyper terminal program under windows 7. Users may copy **hyperterm.exe** and **hyperterm.dll** from XP PC to Windows 7 PC and still works fine. Or users may use **putty** , a 3rd party utility program, as terminal emulation program.

Here is the step guide –

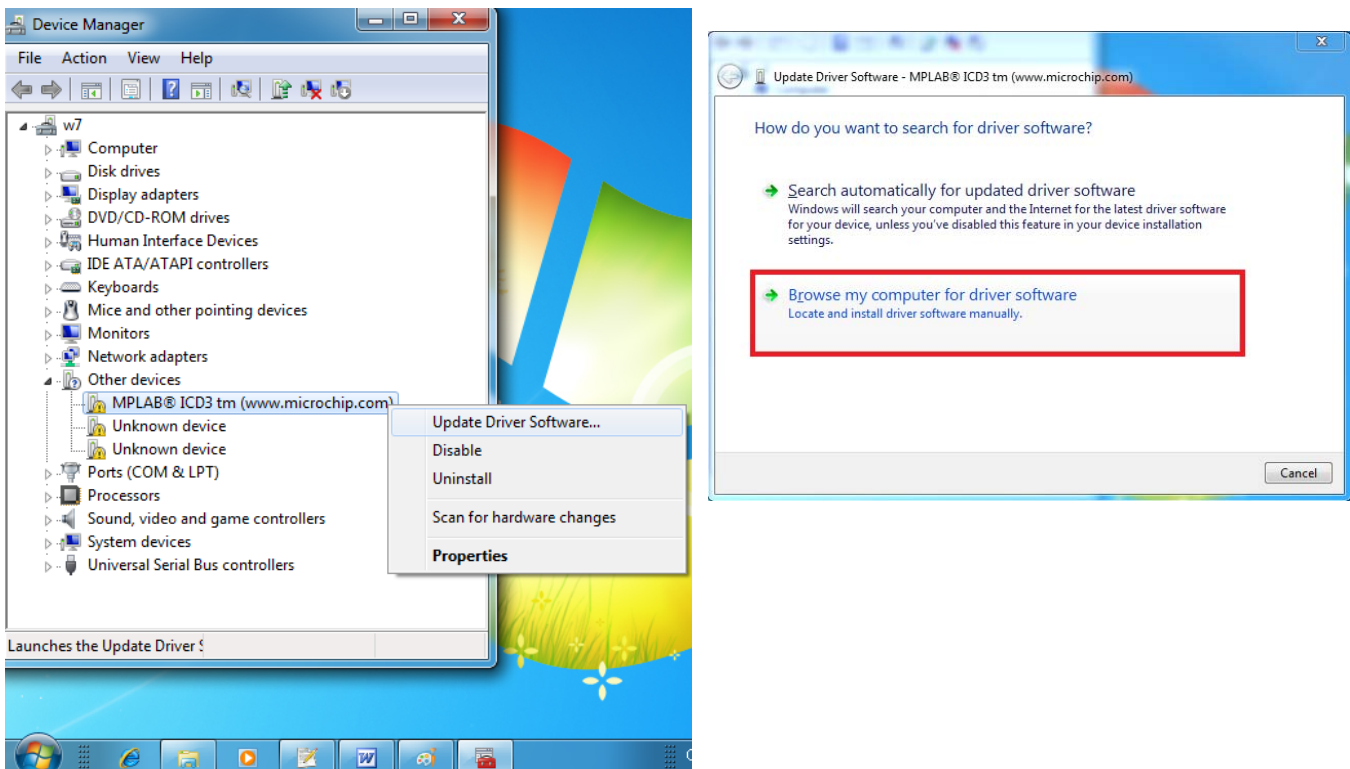
1. ICD3 driver installation – After plug in ICD3 with PC-USB, user will notice the blink of status LED on ICD which indicaties an error condition.



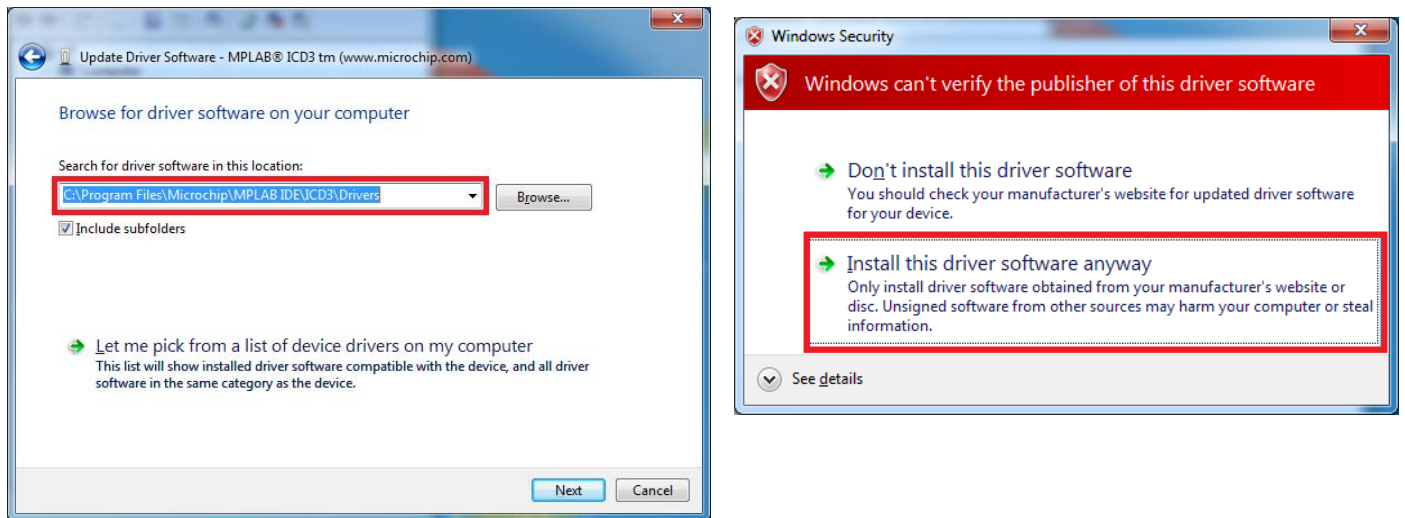
Now go to windows 7 PC, Press **Win+Break** key and click on **Device Manager**, or click on **control panel-hardware and sound-Devices and printers-Device Manager**. Note system will identify ICD3 as unknown devices in the other devices



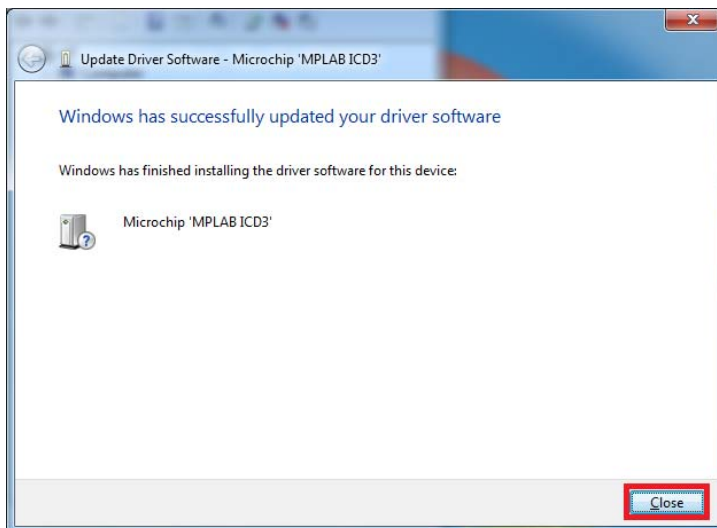
Right-click to update driver as illustrated below -



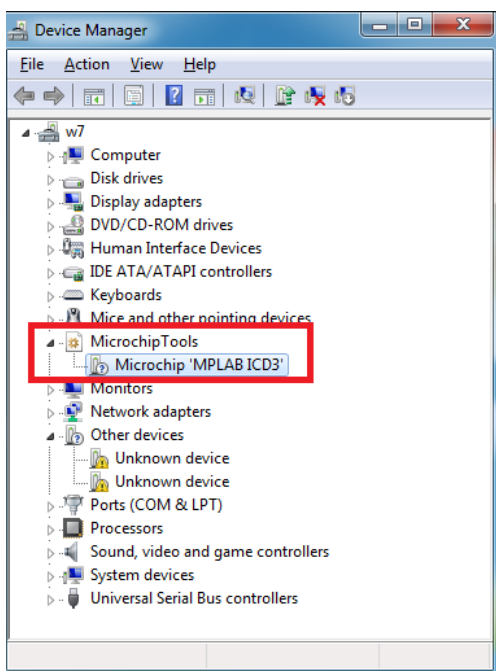
Select driver from as illustrated, and bypass the warning.



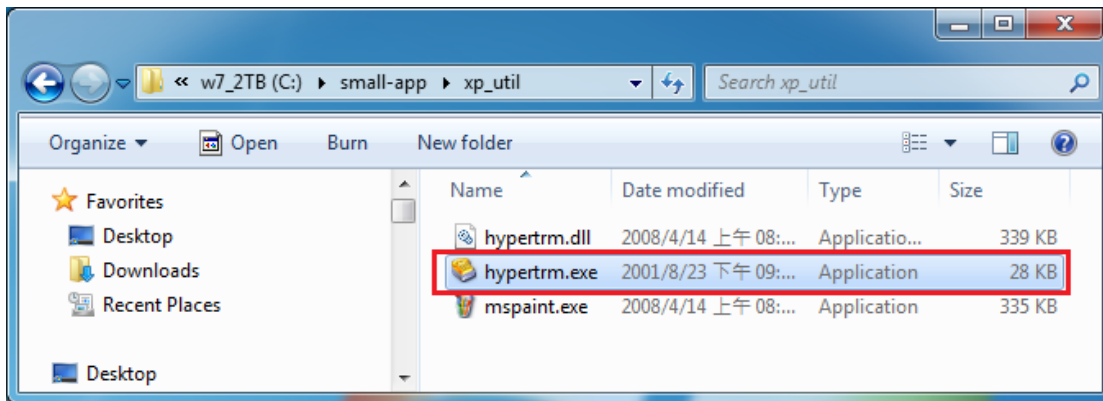
Wait for a while for driver installation, user will see the following screen upon completion.



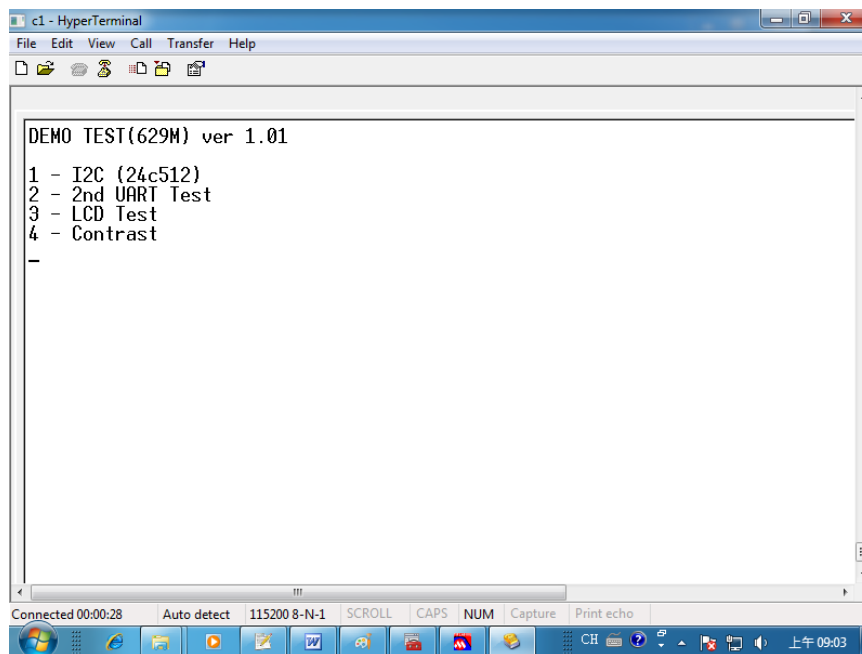
Note the status LED on ICD is now off and the device manager will show ICD3 driver installed.



2. Hyper Terminal – copy **hypertrm.exe** and **hyptertrm.dll** from XP into a folder, and open the executable.



Make sure the serial protocol is set right (115200/N/8/1/No flow control) on correct COM port. The resulting hyper terminal should look like the following -



Now you are all set to run the development environment under Windows 7. Enjoy your ride with **Bolymin** BEGV6xx series display embedded systems.

<END of BEGV606M User Manual>

Copyright

Copyright © 2010 BOLYMIN, INC. All rights reserved. No part of the materials may be reproduced in any form or by any means without prior written consent of BOLYMIN, INC.

Disclaimer

THE CONTENTS OF THIS DOCUMENT ARE SUBJECT TO CHANGE WITHOUT NOTICE. BOLYMIN, INC. RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HERIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. BOLYMIN, INC. DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HERIN; NEITHER DOSE IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

CUSTOMERS ARE ADVISED TO CONSULT WITH BOLYMIN, INC. OR ITS COMMERCIAL DISTRIBUTORS BEFORE ORDERING.

BOLYMIN, INC.

5F, 38 Keya Road, Daya Dist., 42881 Taichung City, Taiwan, R.O.C.

WEB SITE: <http://www.bolymin.com.tw>

TEL: +886-4-2565-8689

FAX: +886-4-2565-8698