BEGV627M

USER MANUAL

- LCD Embedded System
- Microchip PIC24FJ64GA002 MCU
- 64KB program flash/ 8KB SRAM
- I^2C-EEPROM 64KB(Additional 64KB option)
- Dot Matrix 128*64 I^2C-STN LCD w/ white LED backlight
- 1* RS232 + (shared RS232/RS422/RS485)
- 3 GPIOs
- 4-wire resistive touch panel
- Operating at 150 mA
- Free SDK Support

Table of Content

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 BEGV627M parcel, make sure that all materials and options are packed inside parcel according to your order.

Packing Contents Check-List

- □ BEGV627M 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 BEGV627M, to help you incorporate BEGV627M 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

BEGV627M is designed based on PIC24FJ64GA002 microprocessor and ST7588T LCD controller. Bolymin offers free Software Design Kit(SDK) for single-chip program development. Together with a 128x64 I2C STN LCD and LED backlight built-in, this all-in-one LCD embedded system BEGV627M offers designer a ideal solution to save hardware/software integration cost, space, and design time.

Armed with RS232, RS422/485, I^2C-EEPROM, 3 GPIOs, BEGV627M may communicate a variety of devices and peripherals. The BEGV627M is targeted for 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.

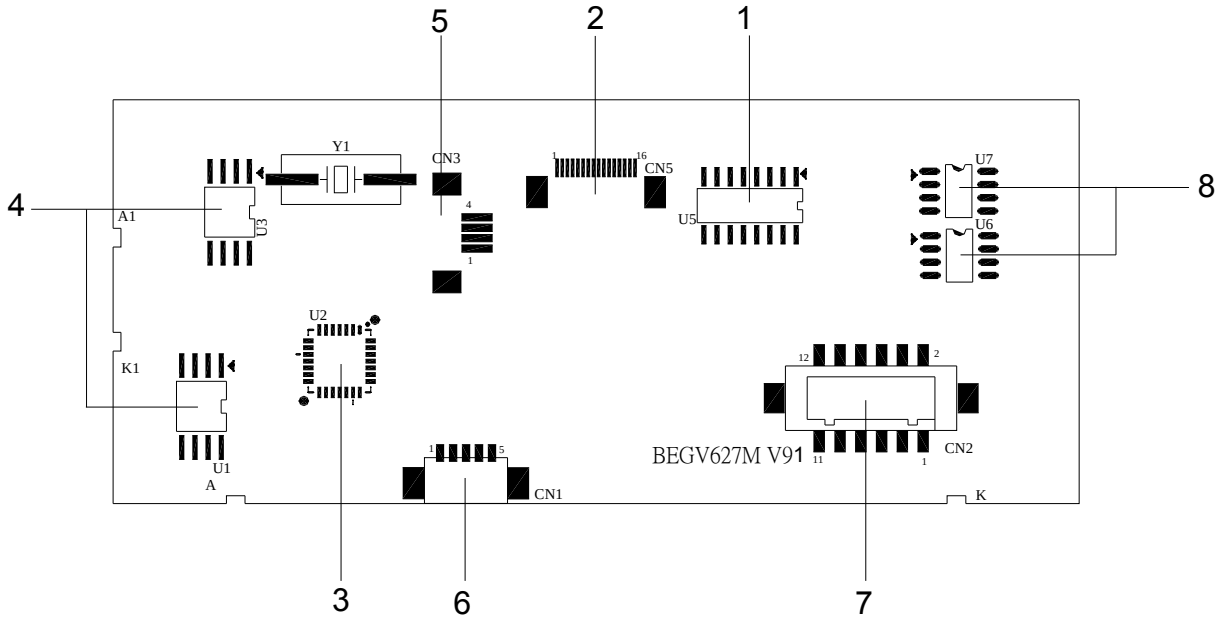
64KB in-system programmable (ISP) flash, additional 64KB extendable, offers sufficient ROM size for designer to develop software and operate i2c-controlled dot matrix LCD and 3 GPIOs (IOA, IOB, IOC).

BEGV627M 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.

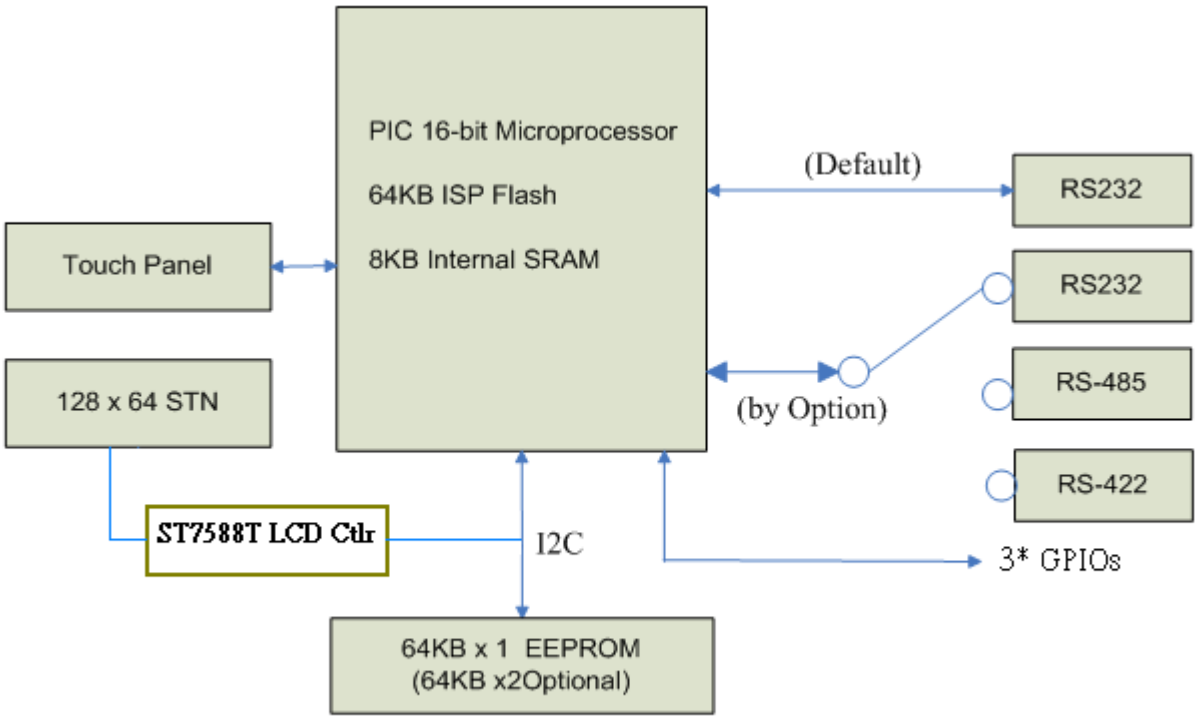
(Drawing 1.2)



1 :RS232
2 :CN5
3: Microprocessor
4 :EEPROM
5 :CN3
6 :CN1 – ISP
7 : CN2 - IO
8 : RS422/485

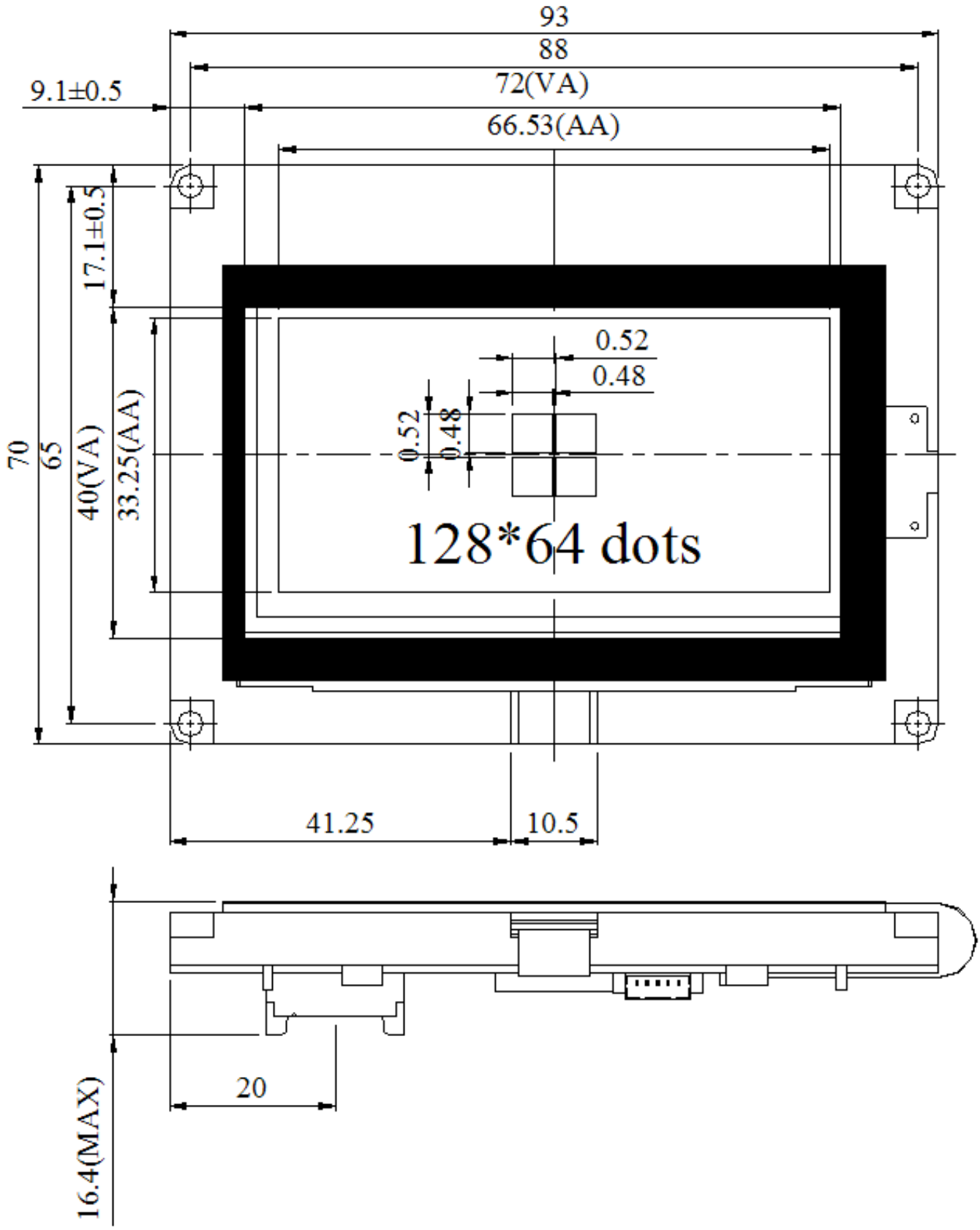
1.3 Block Diagram

(Drawing 1.3)

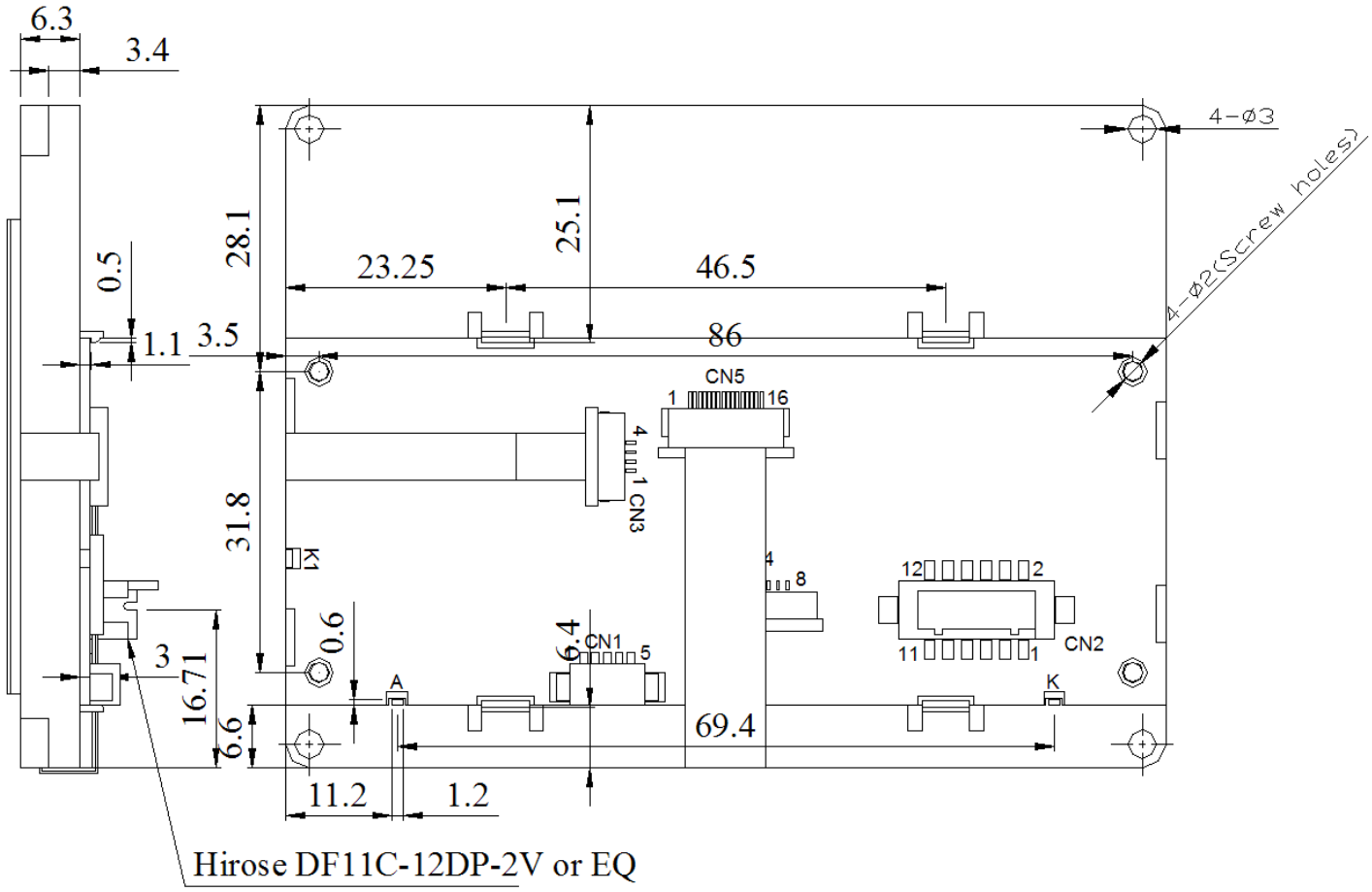


1.4 Mechanical Dimensions

(Drawing 1.4) (Front & top view)



(Side & Back view)



1.5 Board Specifications

(Table 1.5)

MCU	High-performance, Low-power PIC 16-bit microprocessor Microchip PIC24FJ64GA002
Memory	64K Bytes In-System Programmable Flash 8K Bytes Internal SRAM 1 x 64K Bytes External I2C-EEPROM (Additional 64KB optional)
Display	Support I2C interface, resolution 128 x 64 monochrome STN LCD, with edge LED white backlight only; use ST7588t for LCD controller.
Touch Panel	Support four-wired resistive touch panel
Serial Ports	Support 1 x RS232 port, and 1 x RS232/RS422/RS485 co-shared port

1.6 Ordering Information

(Table 1.6)

Part No.	Description	RS232-A	RS232-B	RS422	RS485
BEGV627M	Dual RS232	☆	☆		
BEGV627M1	One RS232	☆			
BEGV627M2	One RS232/One RS422	☆		☆	
BEGV627M3	One RS232/One RS485	☆			☆

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

Chapter 2 Installation

Abstract

This chapter is to offer designer fundamental information of BEGV627M connectors, in order to help designer configure correct setting and connection between BEGV627M and system application.

Contents include:

2-1 Connectors

2.1 Connectors

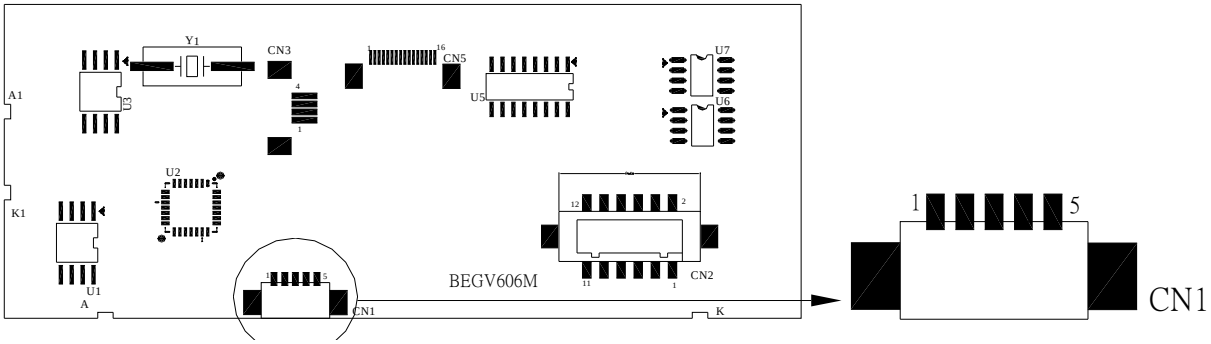
Connectors are the key link between BEGV627M and external devices. Detail locations and functions of available connectors are tabled and illustrated below.

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 627M to J2 of MGI$02 for powering up target board(627M) and for Hyper terminal operation.	DF11C-12DP-2V(57)

2.1.1 Connectors & Pin Definition of CN1

(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)

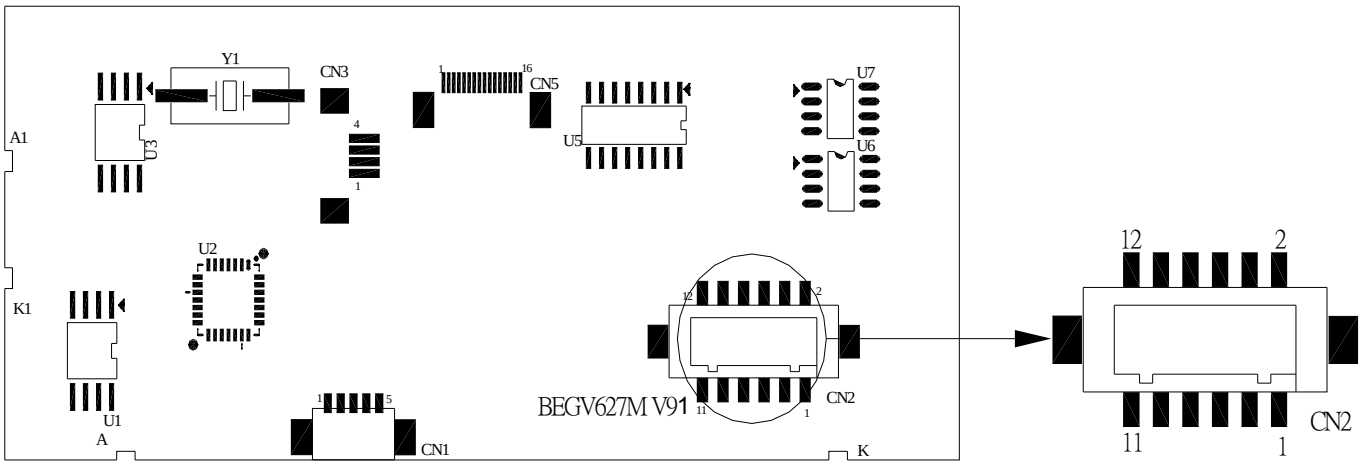
- **ISP Power & Ground** (Table 2.1.1c)

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 Connector CN2

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

(Drawing 2.1CN2)



- **CN2 Pin Definition of BEGV627M - 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 BEGV627M1 - 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 BEGV627M2 - 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 BEGV627M3 - 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 627M 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)

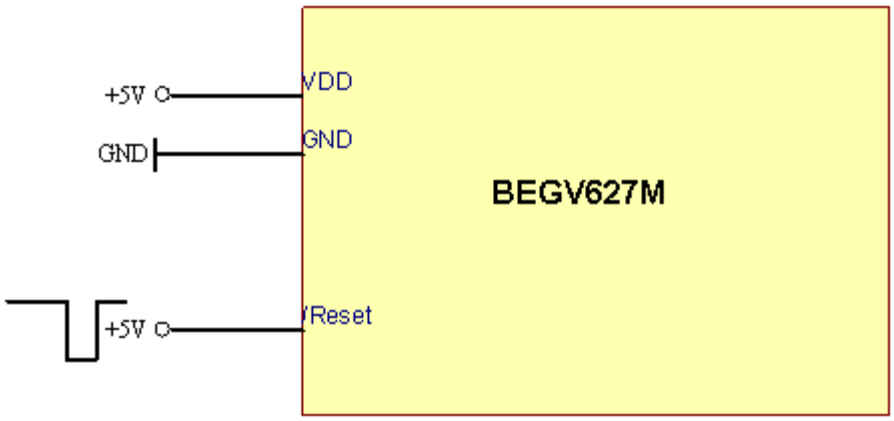
Signal	Type	Pin No.	Description
/Reset	I	11	Auxiliary moment reset for external input
IOA	U	8	I/O port (PIC24FJ64GA002 port RP4, #8)
IOB	U	10	I/O port (PIC24FJ64GA002 port RP7,#13)
IOC	U	12	I/O port (PIC24FJ64GA002 port RP11,#19)

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

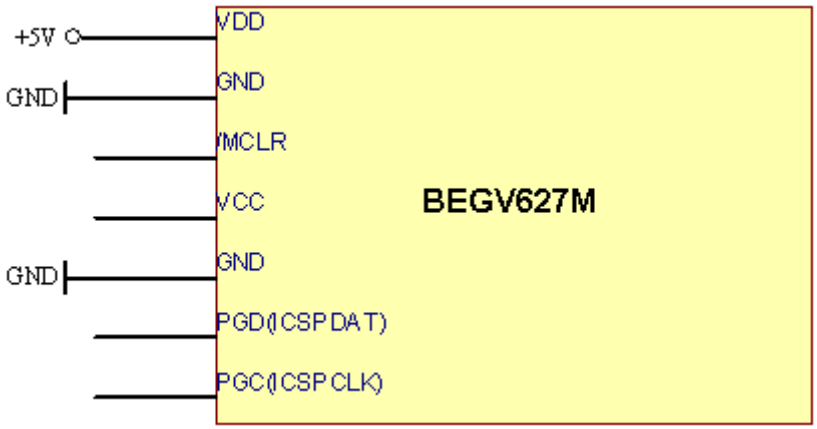
2.2.1 Power/LCD/Backlight

Diagram of system power supply.



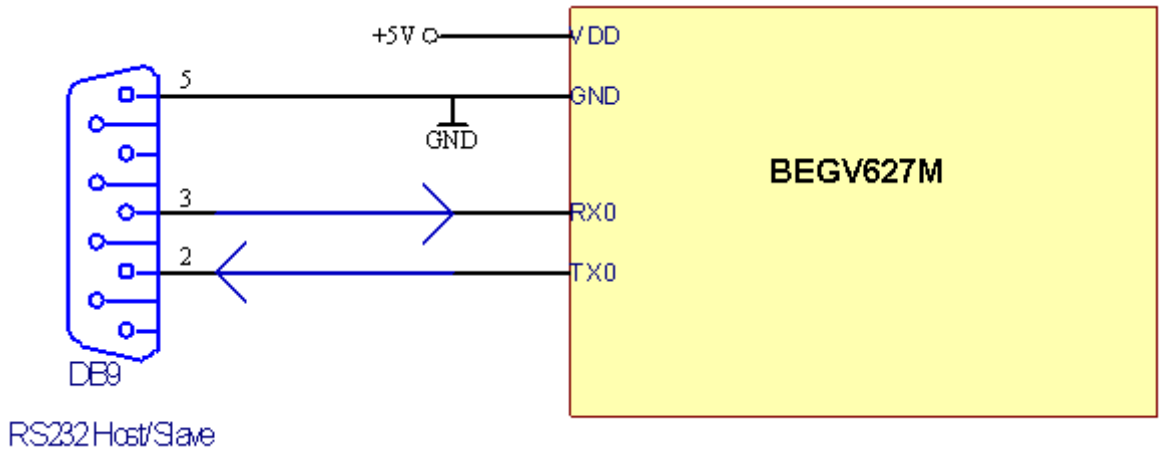
2.2.2 In-System Programming

Customer need to buy Microchip ICD3 for ISP. Also need to buy converter ISP cable to connect ICD3 to ISP port on 627M.



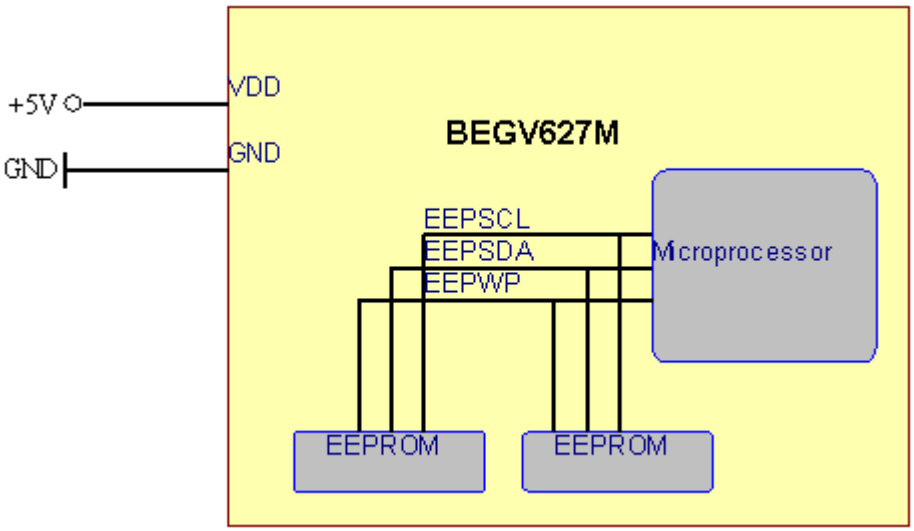
2.2.3 RS-232

BEGV627M offers RS-232 port to contact PC or other RS232 device directly without ICL232.



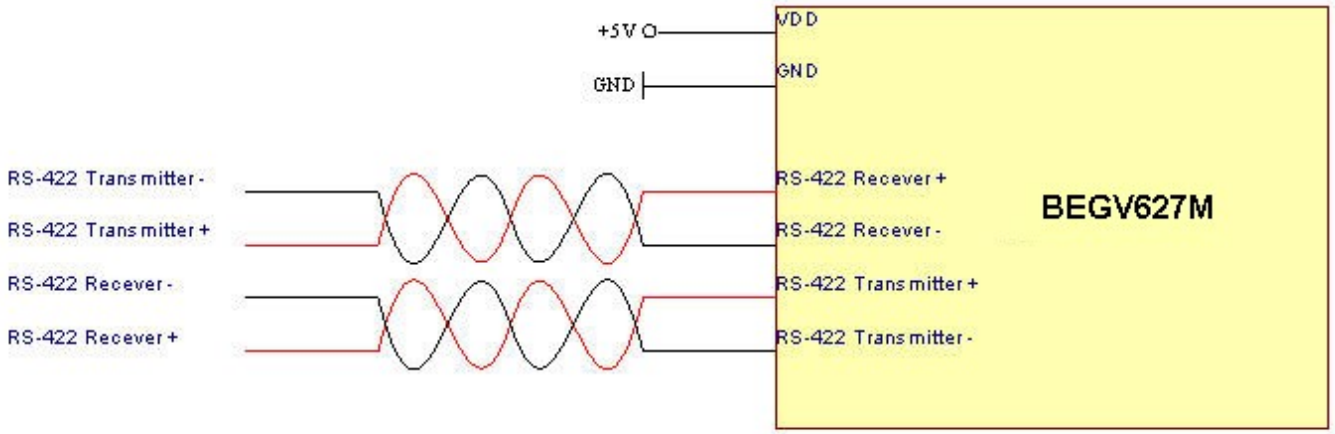
2.2.4 I²C-EEPROM

BEGV627M offers I²C port. Via this I²C port, designer may control 64Kbytes x 2 in-system EEPROM.



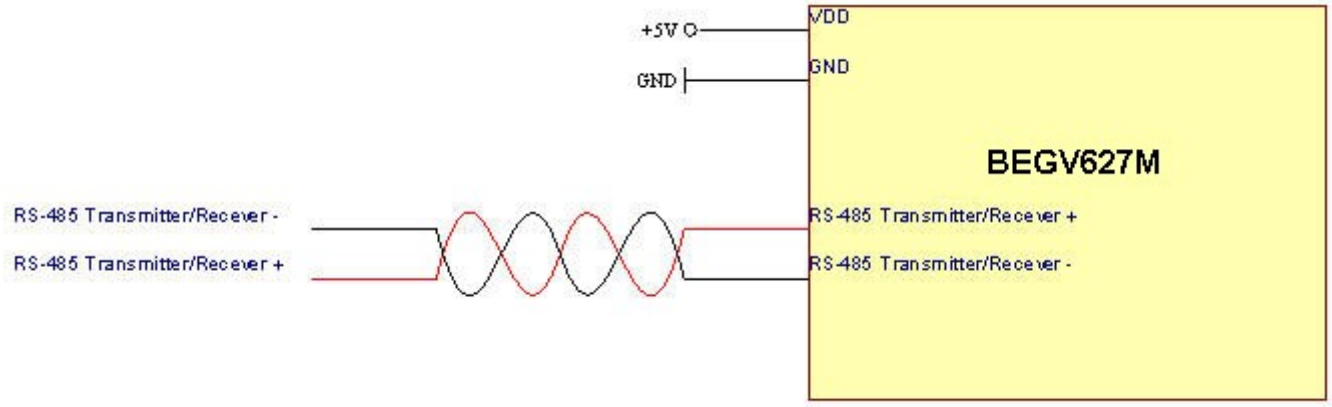
2.2.5 RS-422

BEGV627A offers 1 x RS-422(isolated) port.



2.2.6 RS-485

BEGV627A offers 1 x RS-485(isolated) port.



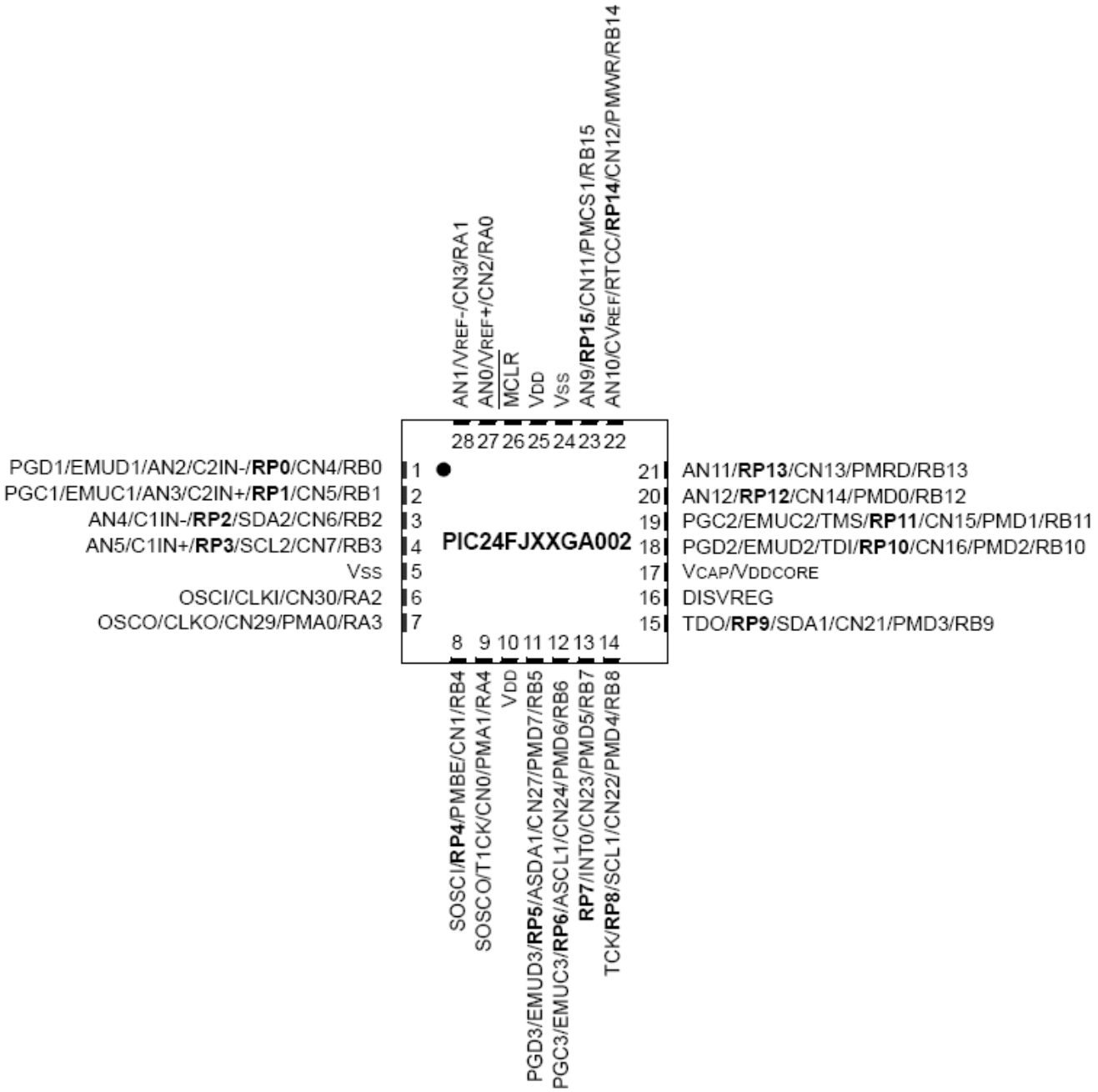
Chapter 3 MCU port mapping

Abstract

This chapter explains PIC24FJ64GA002 MCU pin configuration and port mapping toward key elements such as LCD, Touch Panel, RS-232, RS-422, RS-485, LED Backlight, I^2C- EEPROM and 3 General purpose I/O.

3.1 MCU Pin Configuration

(Drawing 3.1, PIC24FJ64GA002)



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 Touch Panel

(Table 3.2b)

MCU PIC24FJ64GA002	Touch Panel
AN0, #27	X1 (CN3#1)
AN1, #28	Y1 (CN3#2)
AN11, #21	X2 (CN3#3)
AN12, #20	Y2 (CN3#4)

3.2.3 RS-232/RS-422/RS-485

(Table 3.2c)

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 3.2d)

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

3.2.4 Backlight PWM

(Table 3.2e)

MCU PIC24FJ64GA002	LED backlight
RP10, #18	Backlight PWM

3.2.5 I²C-EEPROM

(Table 3.2f)

MCU PIC24FJ64GA002	EEPROM/ I²C
RP2/SDA2, #3	EEPROM SDA
RP3/SCL2, #4	EEPROM SCL
RP8/TCK, #14	EEPROM WP

3.2.6 General Purpose I/O

(Table 3.2g)

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

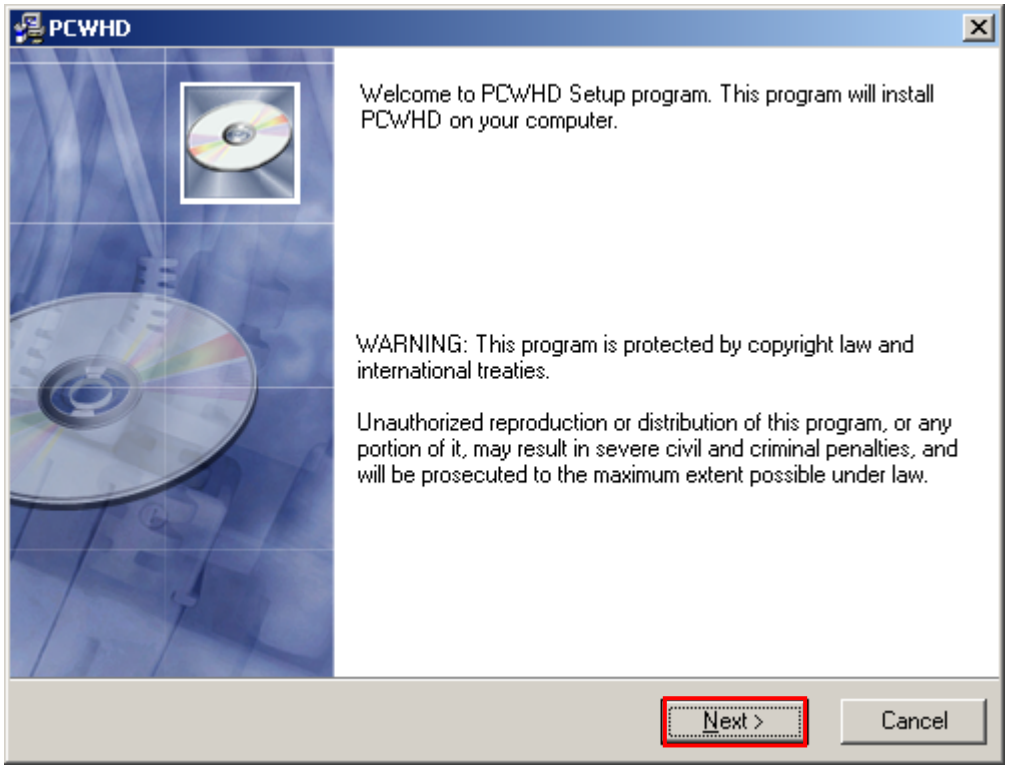
Chapter 4 Software Development Tool & Utility

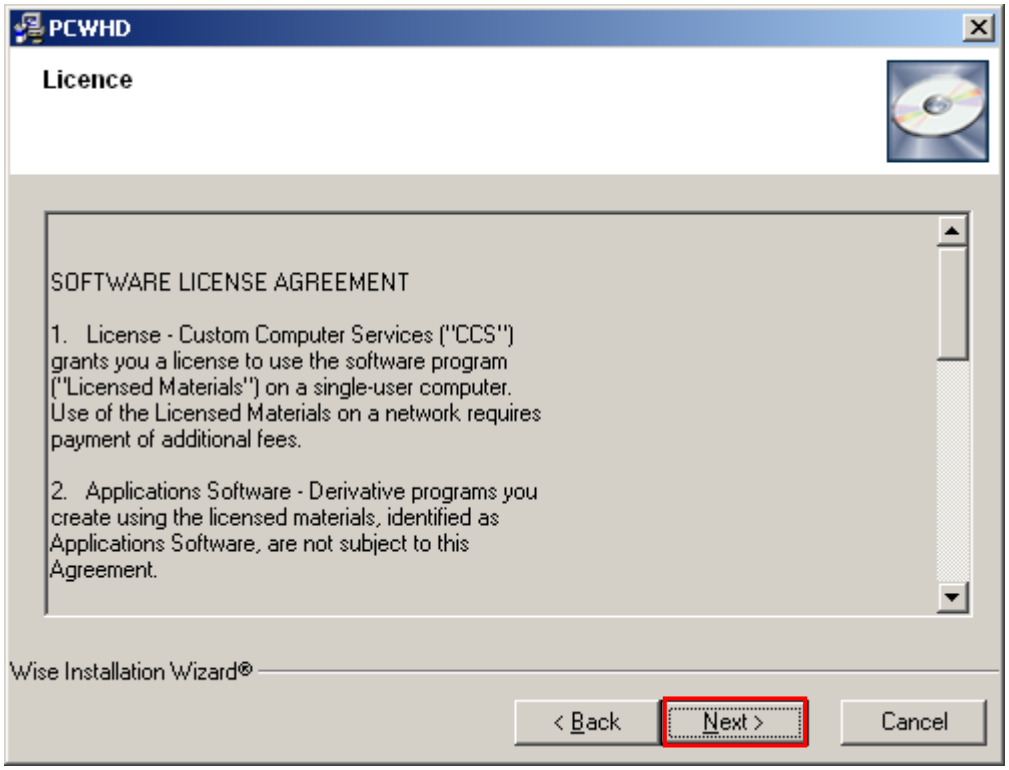
Abstract

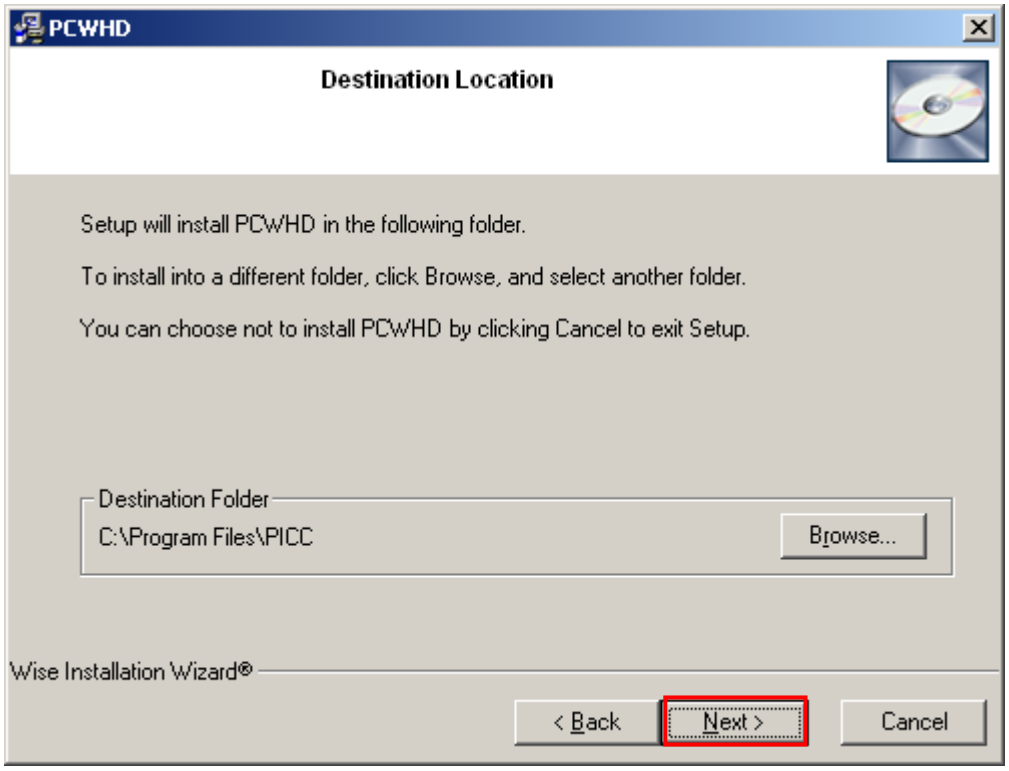
This chapter explains PIC24FJ64GA002 MCU software development tool Microchip ICD3 and Bolymin free software utilities.

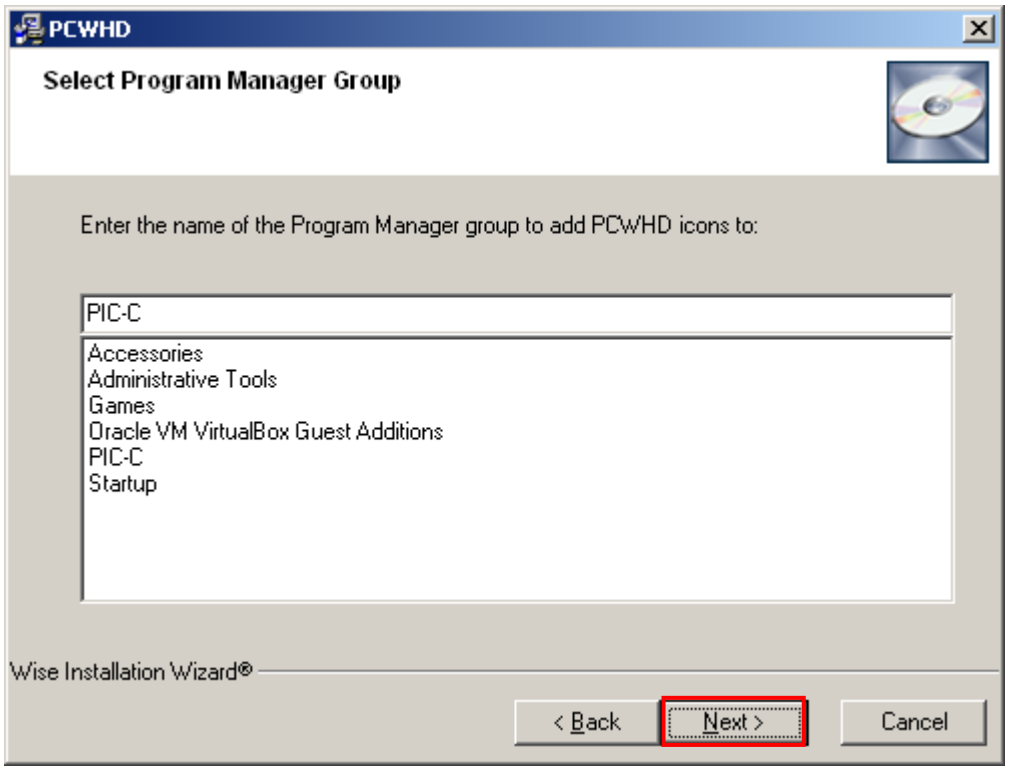
Please follow the step guide to set up Microchip PIC24FJ64GA002/004 Software Development Tools.

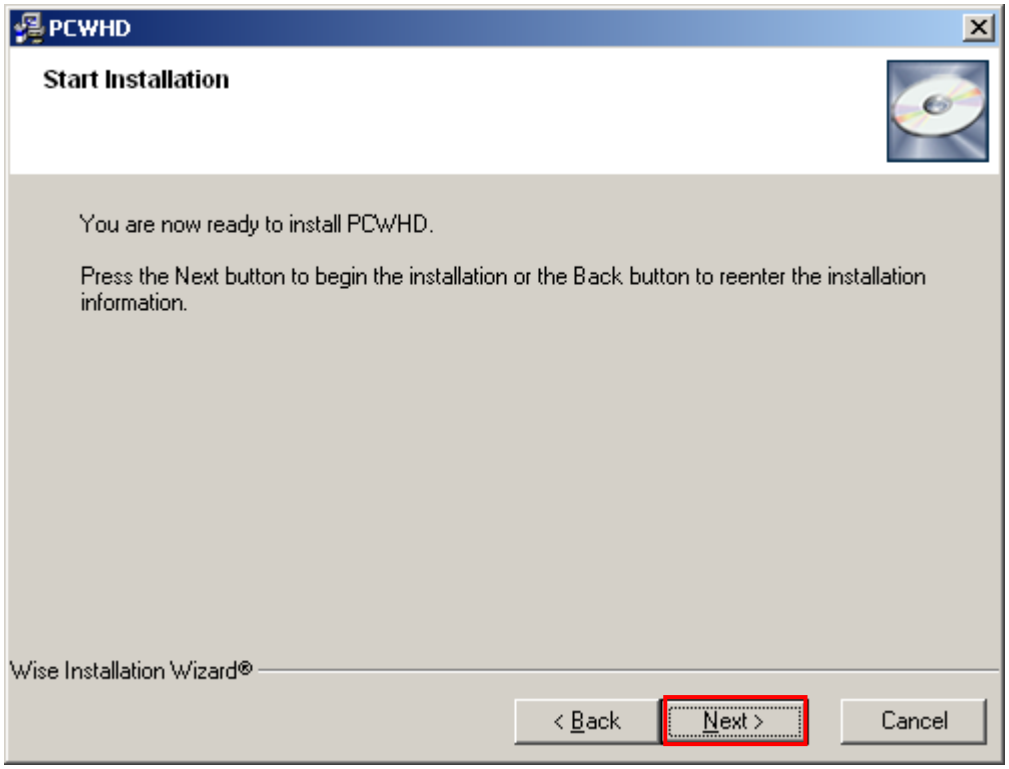
4.1 Install CCSC 4.093

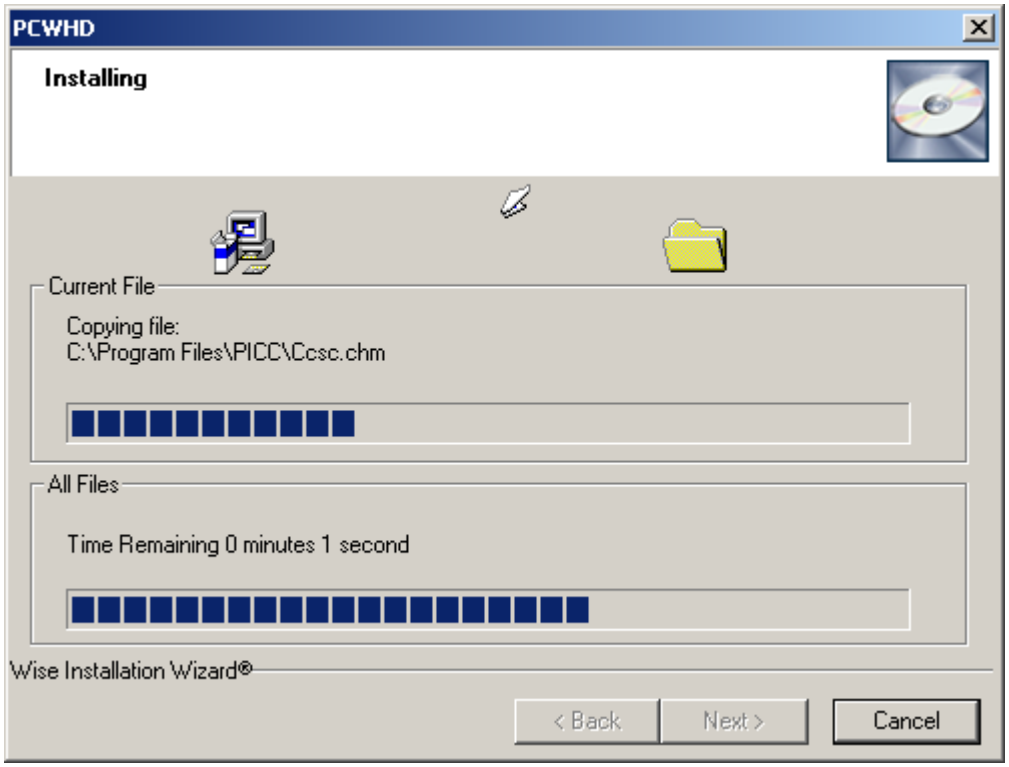


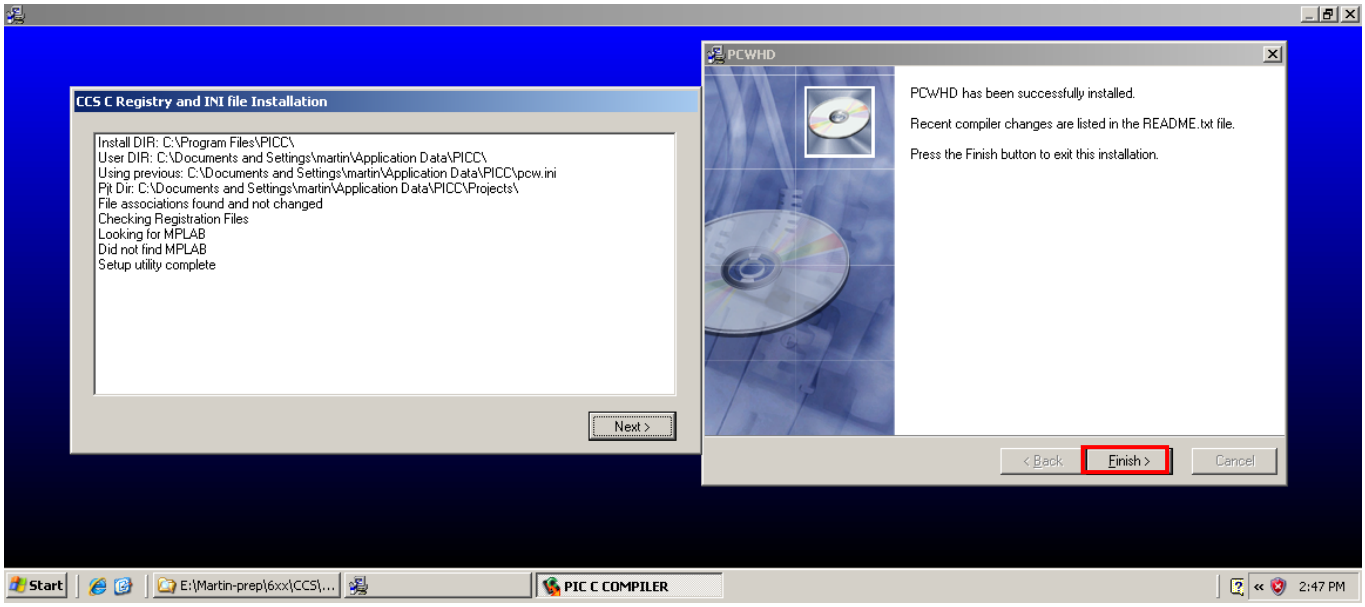


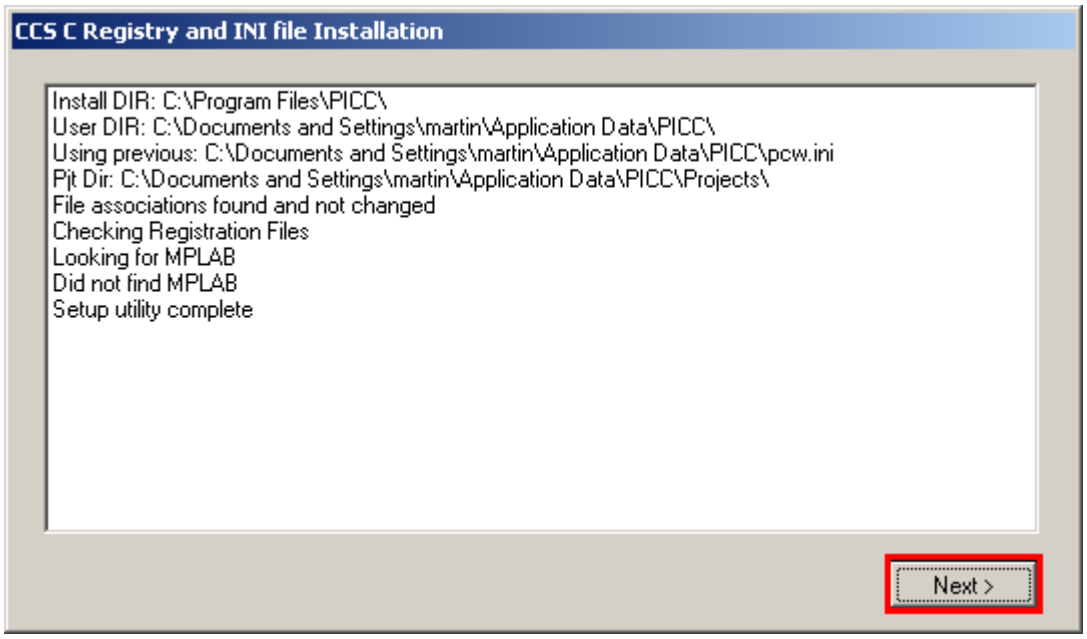

PCWHD
Destination Location
Setup will install PCWHD in the following folder.
To install into a different folder, click Browse, and select another folder.
You can choose not to install PCWHD by clicking Cancel to exit Setup.
Destination Folder
C:\Program Files\PICC
Browse...
Wise Installation Wizard®
< Back
Next >
Cancel


PCWHD
Select Program Manager Group
Enter the name of the Program Manager group to add PCWHD icons to:
PIC-C
Accessories
Administrative Tools
Games
Oracle VM VirtualBox Guest Additions
PIC-C
Startup
Wise Installation Wizard®
< Back
Next >
Cancel


PCWHD
Start Installation
You are now ready to install PCWHD.
Press the Next button to begin the installation or the Back button to reenter the installation information.
Wise Installation Wizard®
< Back
Next >
Cancel

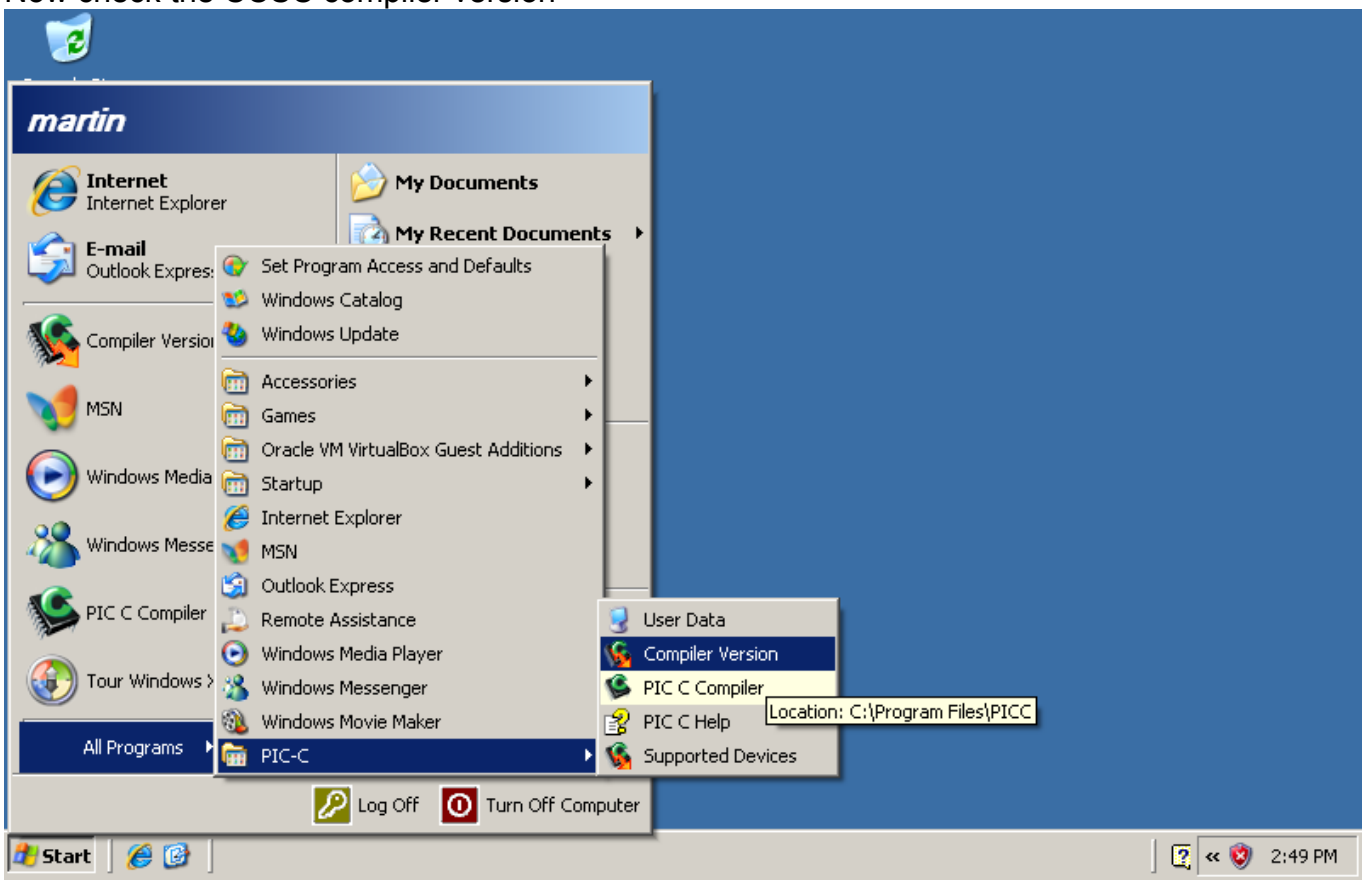

PCWHD
Installing
Current File
Copying file:
C:\Program Files\PICC\Ccsc.chm
All Files
Time Remaining 0 minutes 1 second
Wise Installation Wizard®
< Back
Next >
Cancel



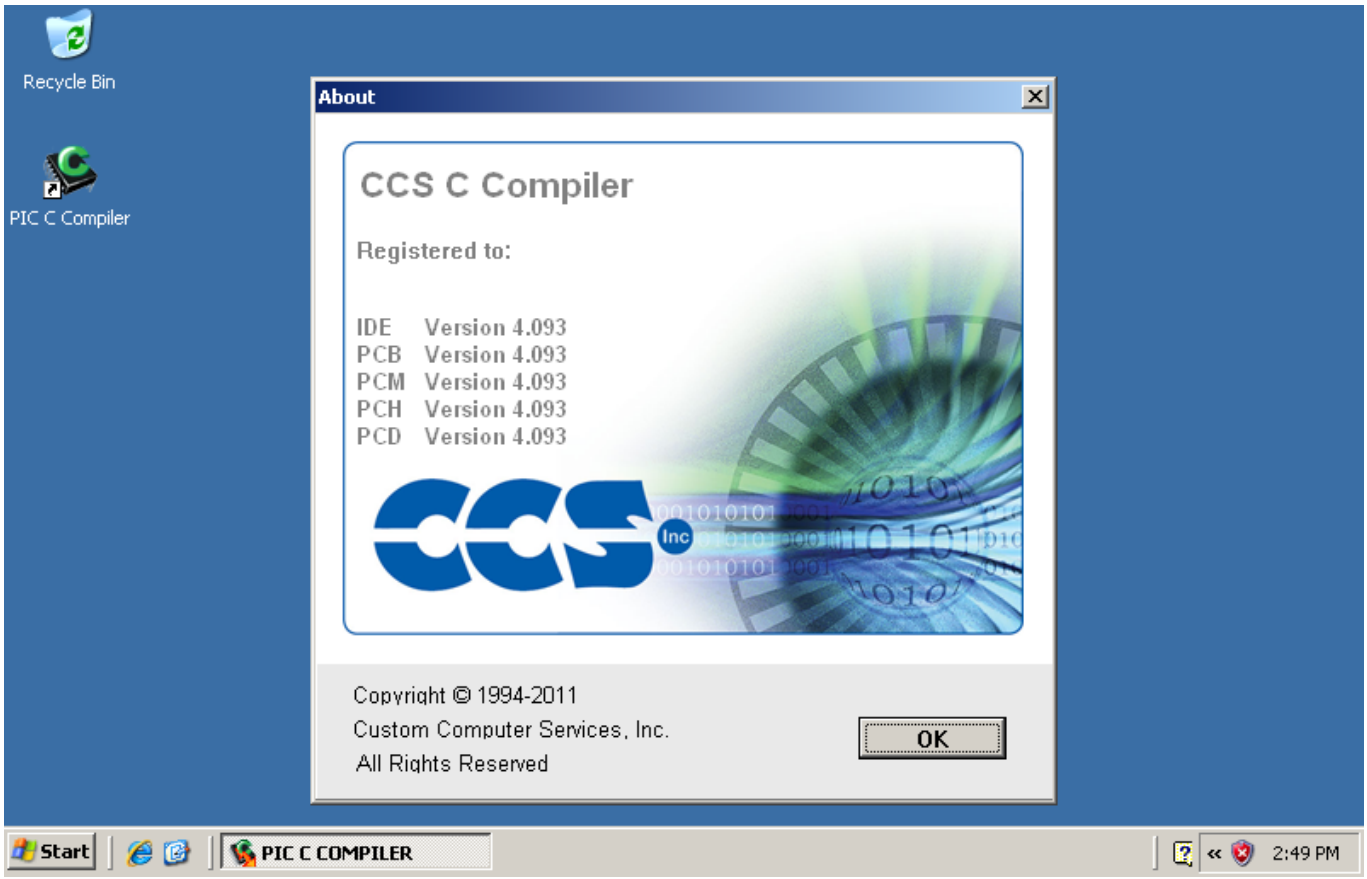


Click **Next** to complete installation of CCSC 4.093 .

Now check the CCSC complier version



CCSC version is 4.093.



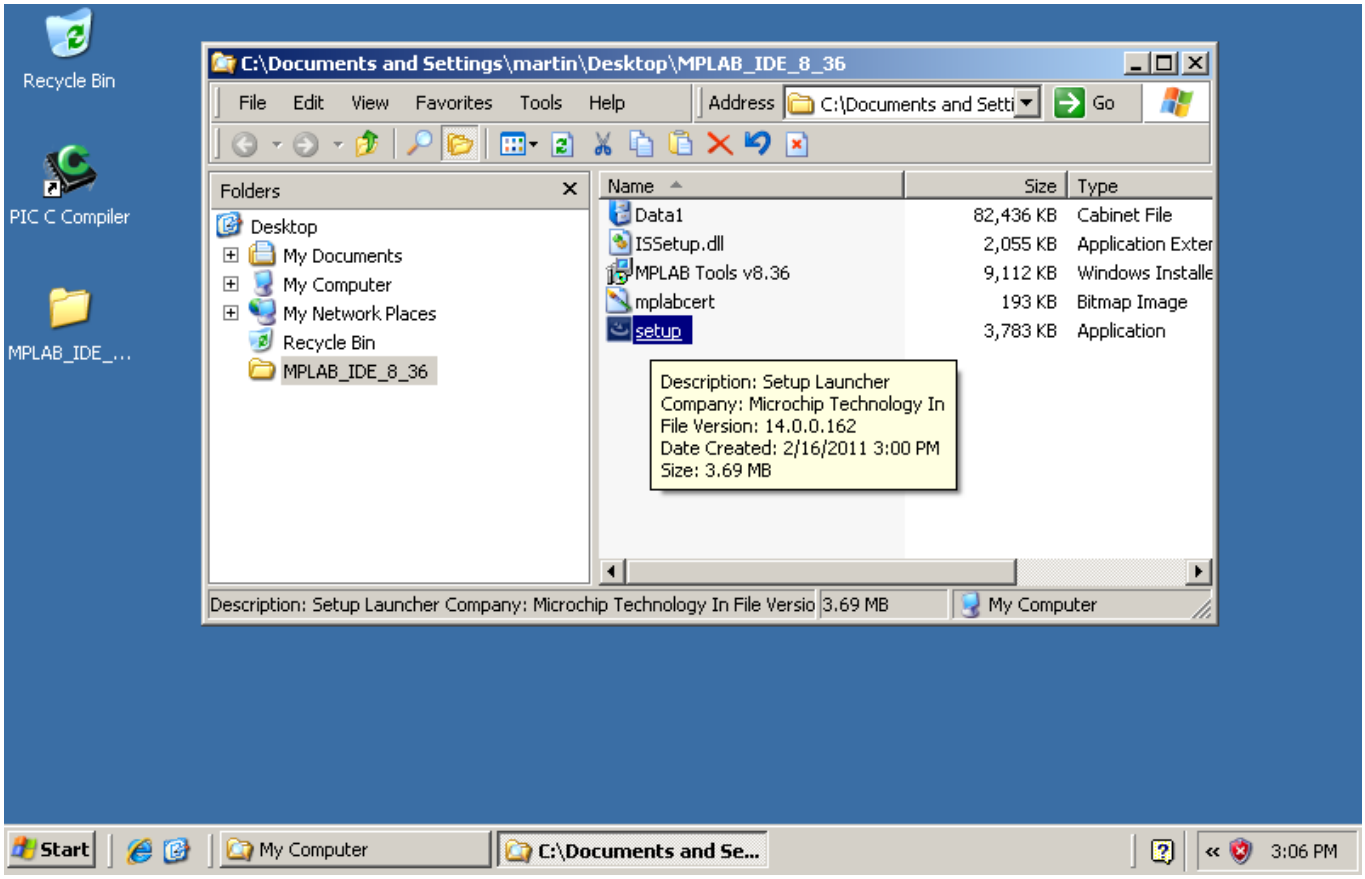
4.2 Install MPLAB ver 8.36

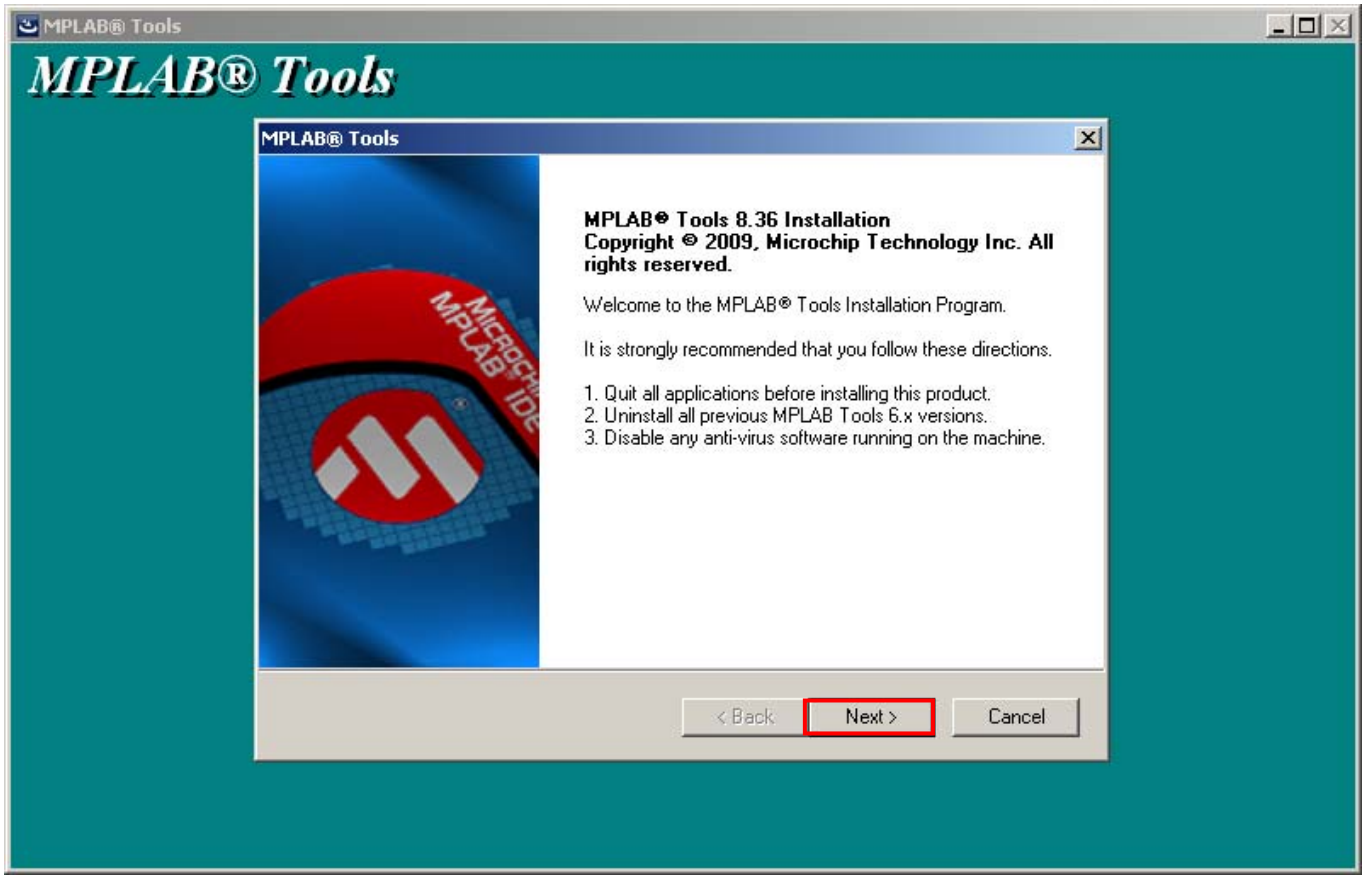
Designers may download software development too MPLAB ver 8.36 from the following URL:

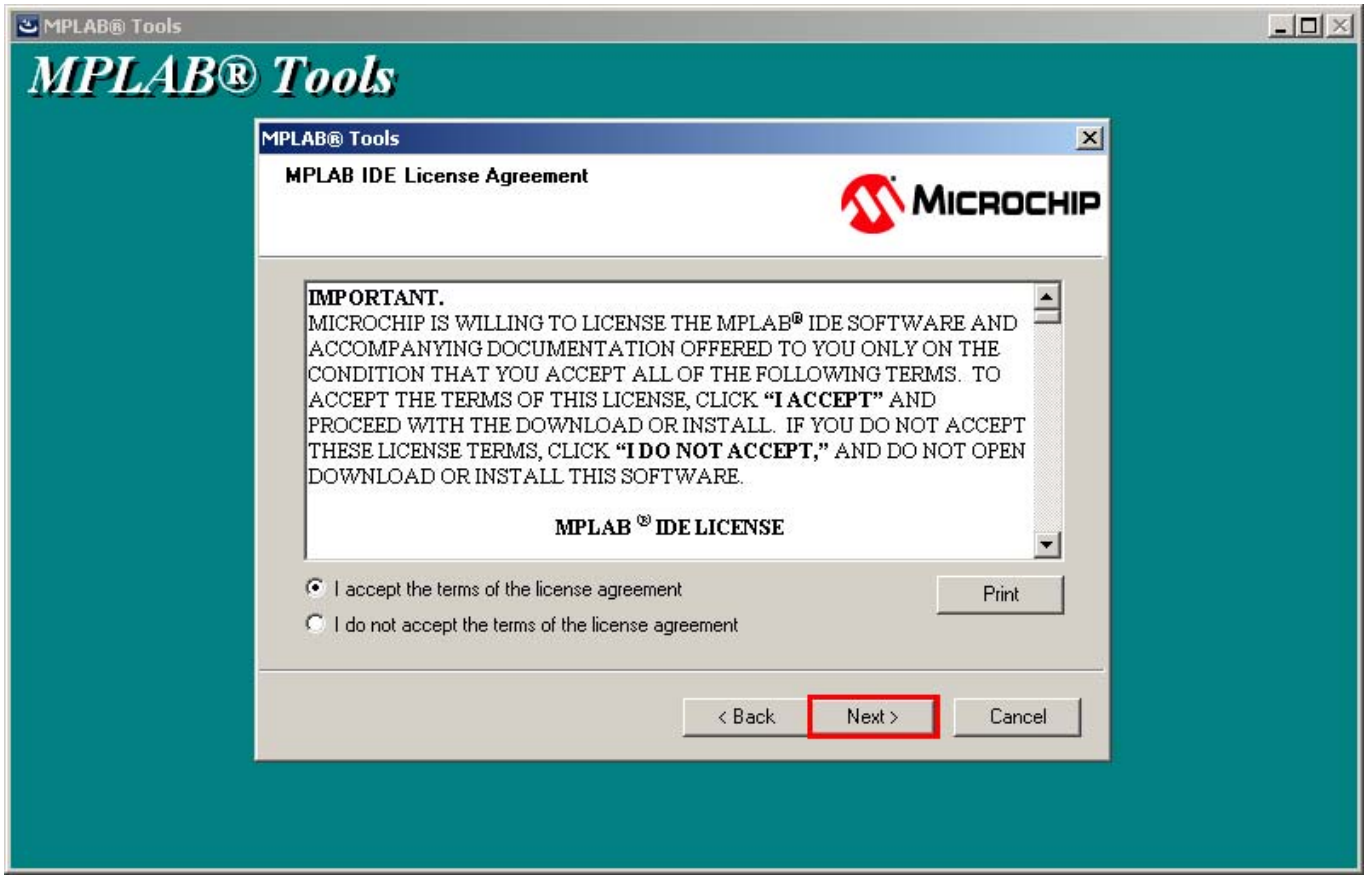
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023073

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

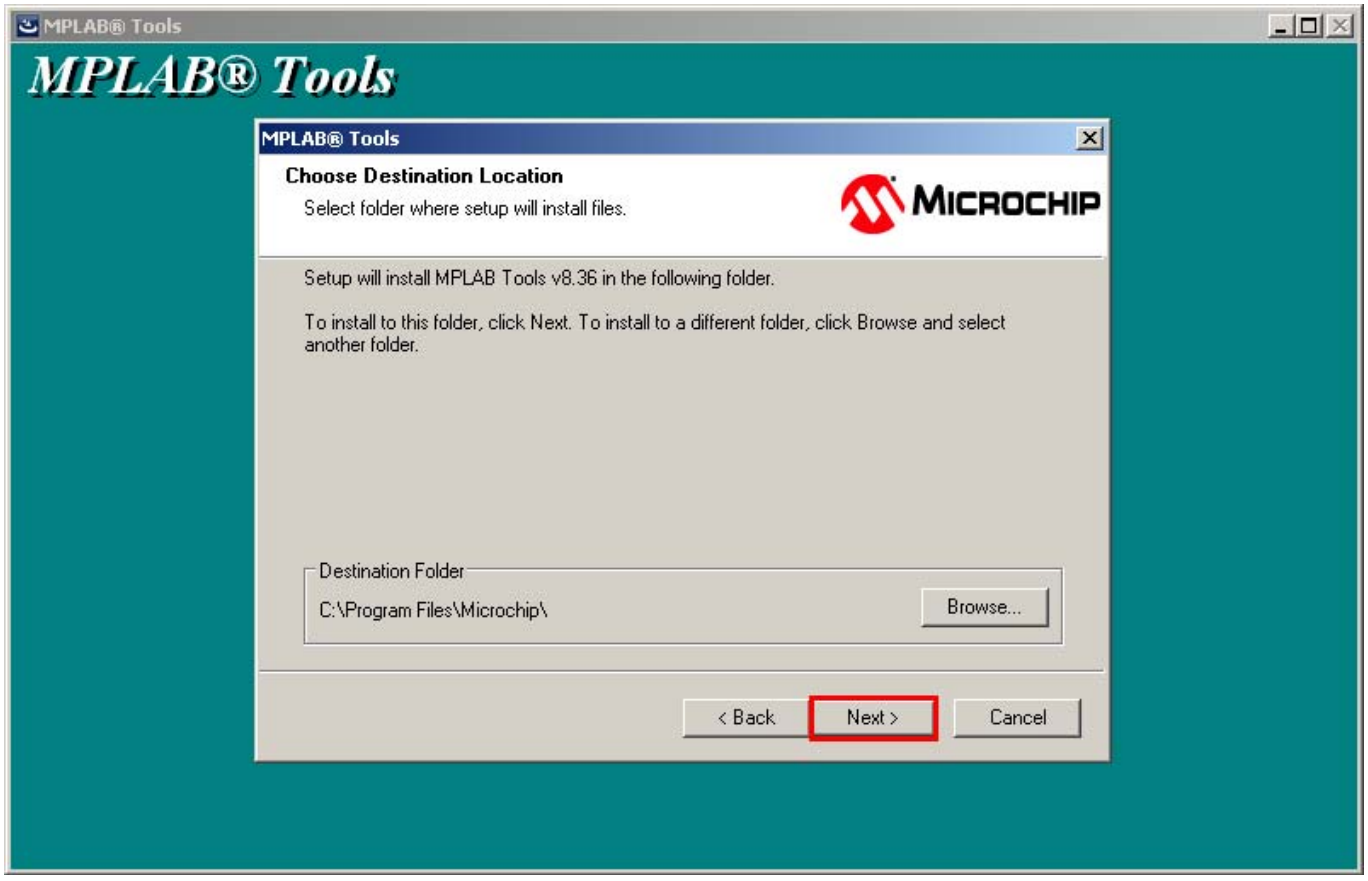
or from BOLYMIN utility disk.

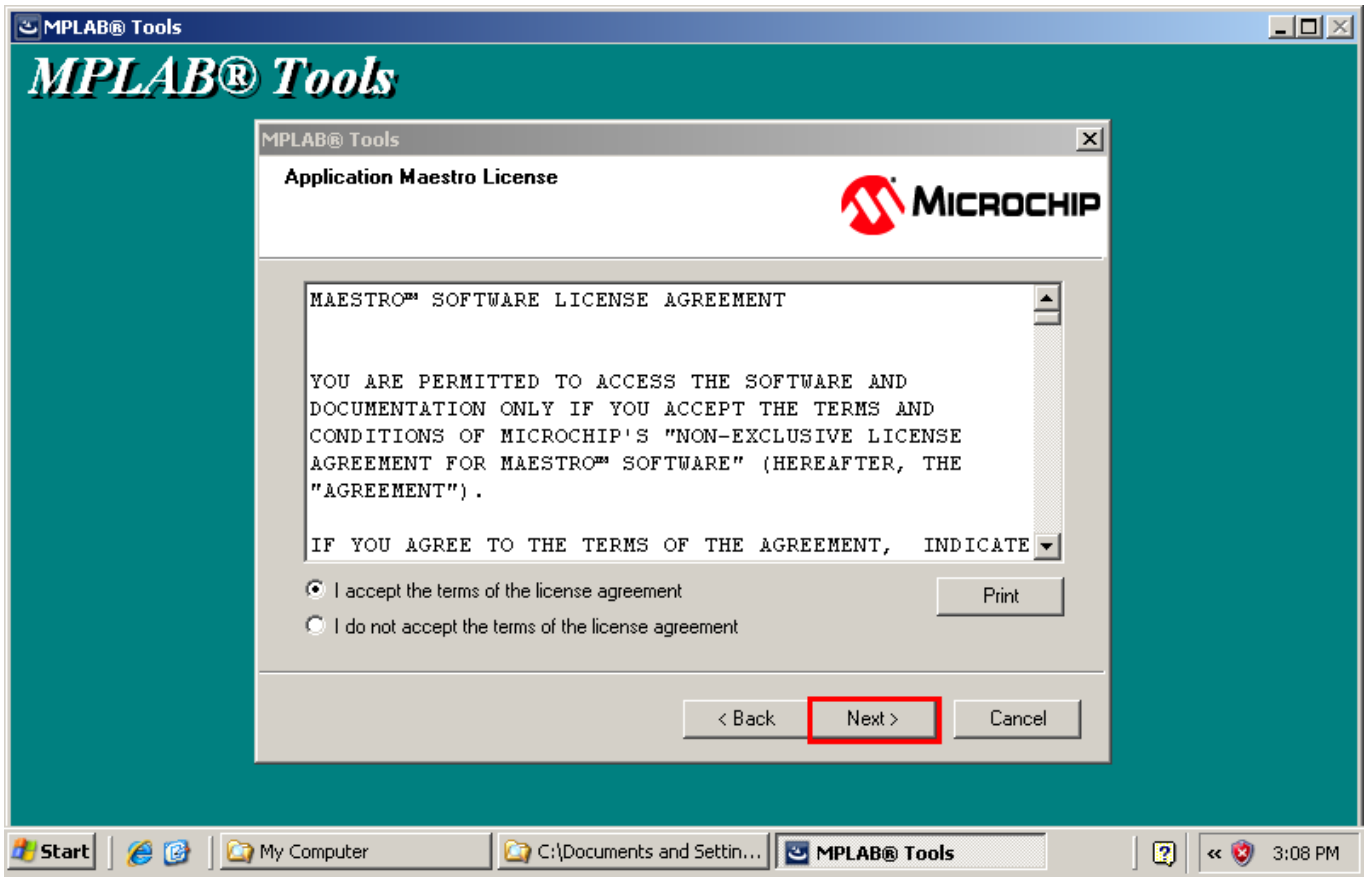


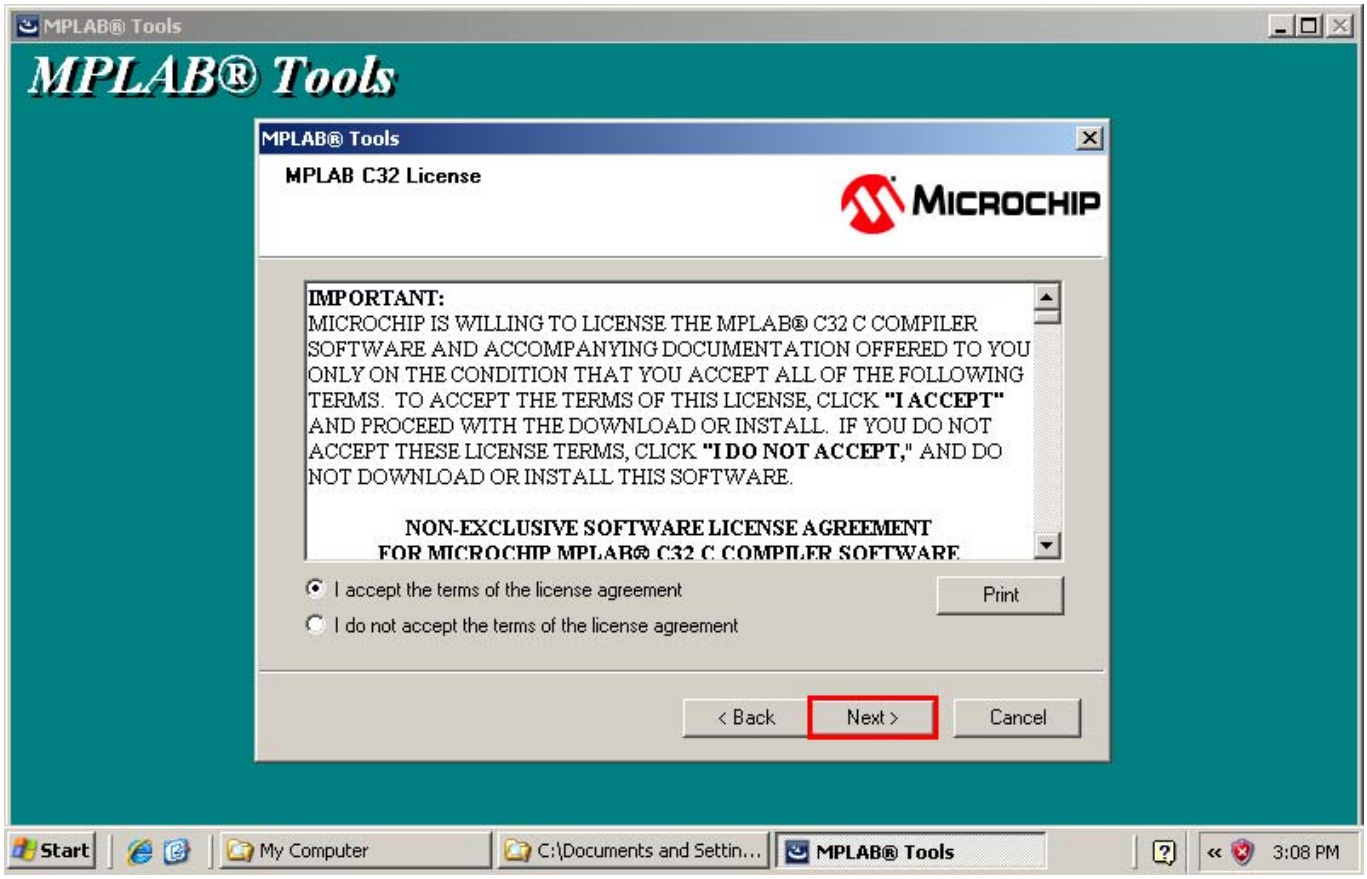

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
MPLAB® Tools 8.36 Installation
Copyright © 2009, Microchip Technology Inc. All rights reserved.
Welcome to the MPLAB® Tools Installation Program.
It is strongly recommended that you follow these directions.
1. Quit all applications before installing this product.
2. Uninstall all previous MPLAB Tools 6.x versions.
3. Disable any anti-virus software running on the machine.
< Back
Next >
Cancel

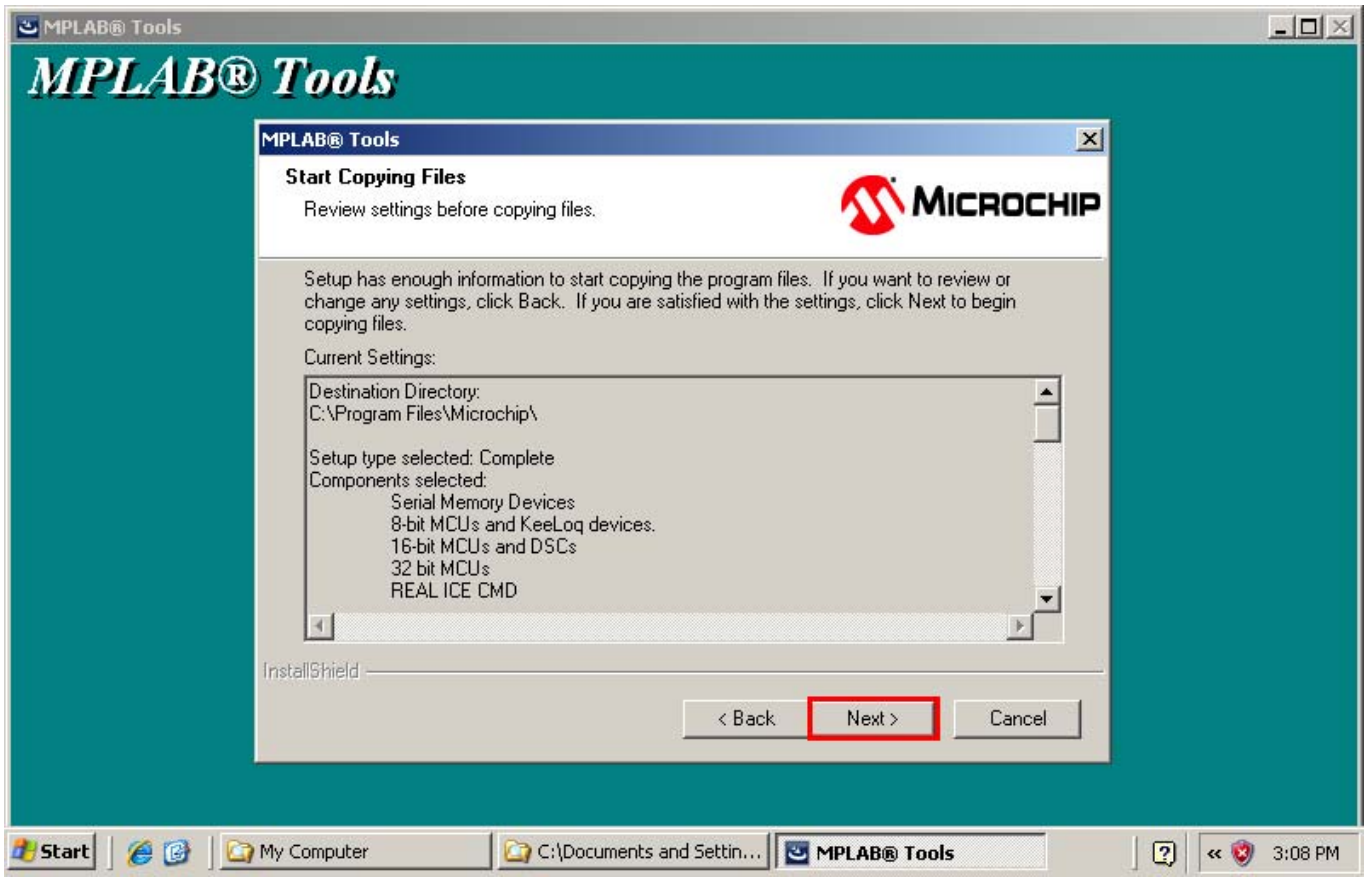

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
MPLAB IDE License Agreement
MICROCHIP
IMPORTANT.
MICROCHIP IS WILLING TO LICENSE THE MPLAB® IDE SOFTWARE AND ACCOMPANYING DOCUMENTATION OFFERED TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE FOLLOWING TERMS. TO ACCEPT THE TERMS OF THIS LICENSE, CLICK "I 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
Print
< Back
Next >
Cancel

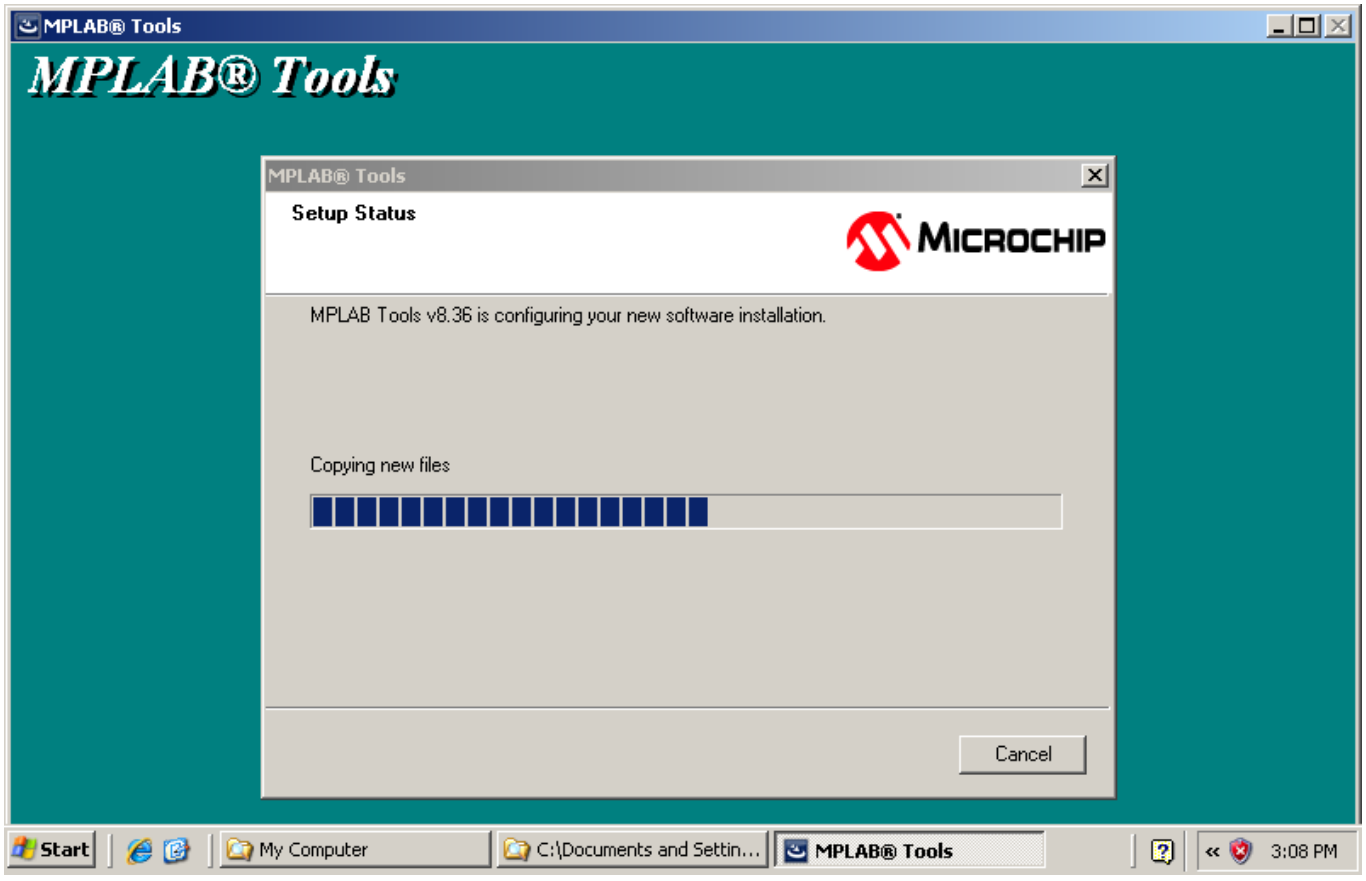

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
Setup Type
Select the setup type to install.
MICROCHIP
Please select a setup type.
Complete
All program features will be installed. (Requires the most disk space.)
Custom
Select which program features you want installed. Recommended for advanced users.
< Back
Next >
Cancel

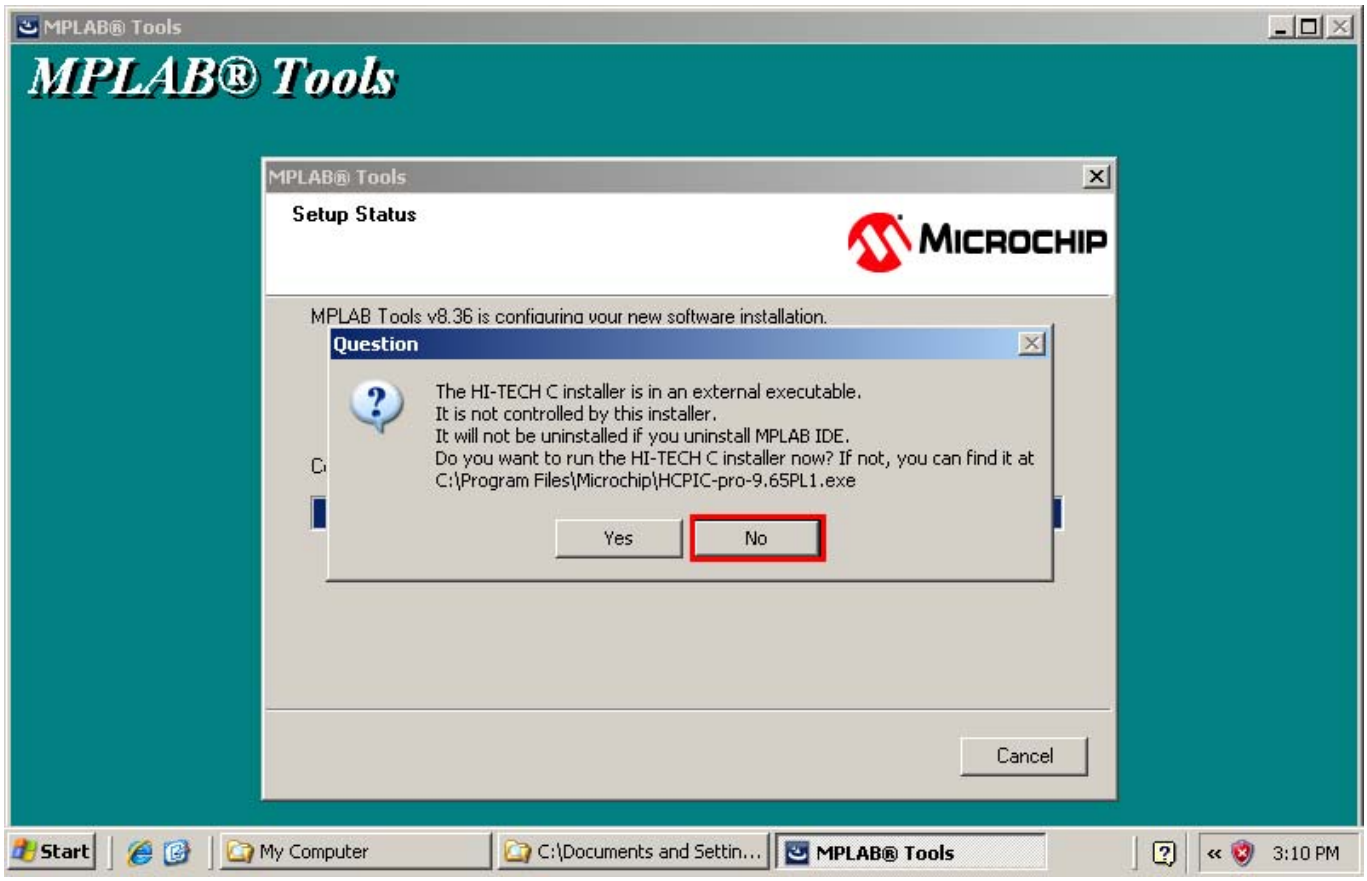

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
Choose Destination Location
Select folder where setup will install files.
MICROCHIP
Setup will install MPLAB Tools v8.36 in the following folder.
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.
Destination Folder
C:\Program Files\Microchip\
Browse...
< Back
Next >
Cancel


MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
Application Maestro License
MICROCHIP
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
Print
< Back
Next >
Cancel
Start
My Computer
C:\Documents and Settin...
MPLAB® Tools
3:08 PM

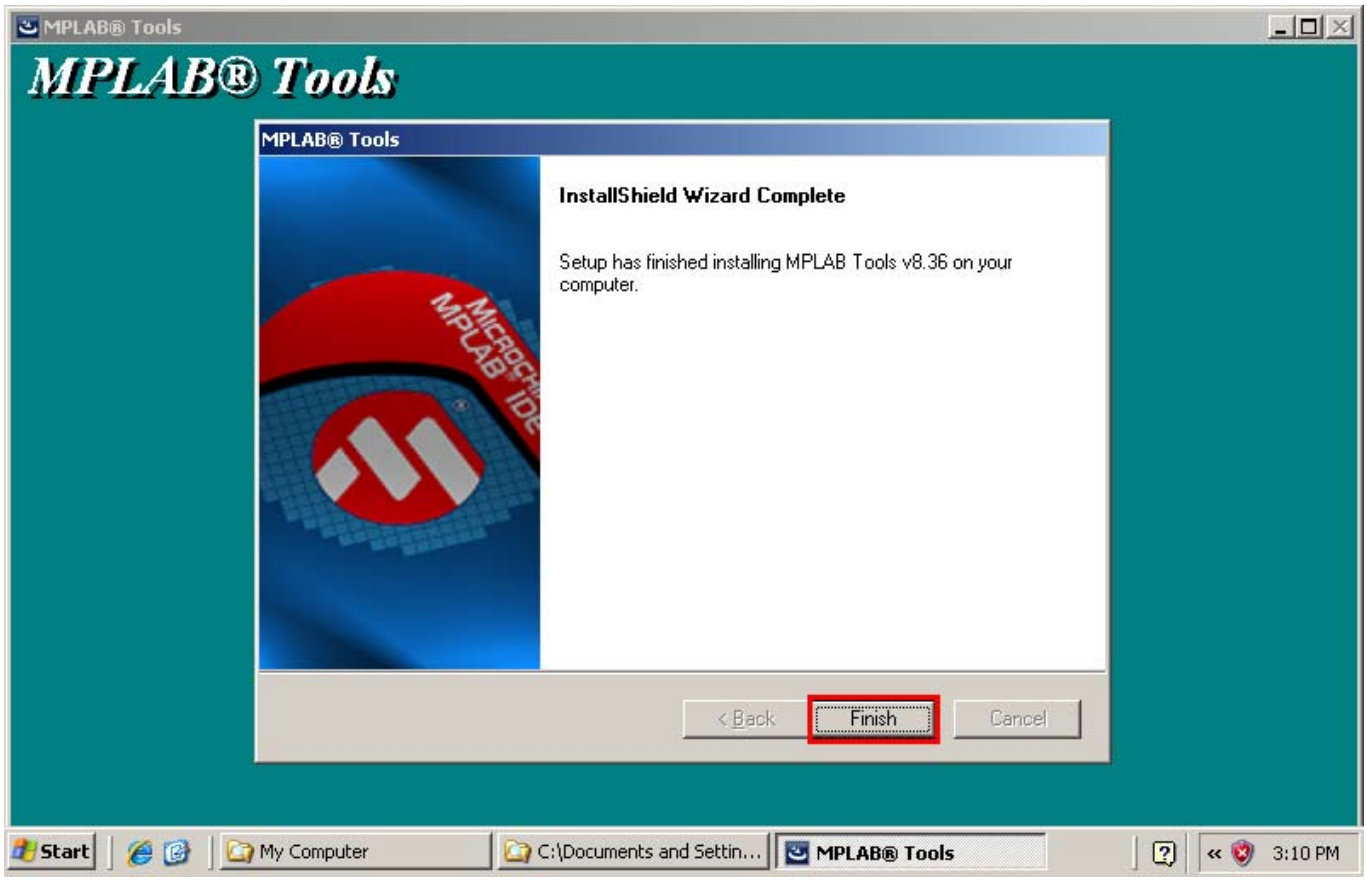

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
MPLAB C32 License
MICROCHIP
IMPORTANT:
MICROCHIP IS WILLING TO LICENSE THE MPLAB® C32 C COMPILER SOFTWARE AND ACCOMPANYING DOCUMENTATION OFFERED TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE FOLLOWING TERMS. TO ACCEPT THE TERMS OF THIS LICENSE, CLICK "I ACCEPT" AND PROCEED WITH THE DOWNLOAD OR INSTALL. IF YOU DO NOT ACCEPT THESE LICENSE TERMS, CLICK "I DO NOT ACCEPT," AND DO NOT DOWNLOAD OR INSTALL THIS SOFTWARE.
NON-EXCLUSIVE SOFTWARE LICENSE AGREEMENT
FOR MICROCHIP MPLAB® C32 C COMPILER SOFTWARE
I accept the terms of the license agreement
I do not accept the terms of the license agreement
Print
< Back
Next >
Cancel
Start
My Computer
C:\Documents and Settin...
MPLAB® Tools
3:08 PM

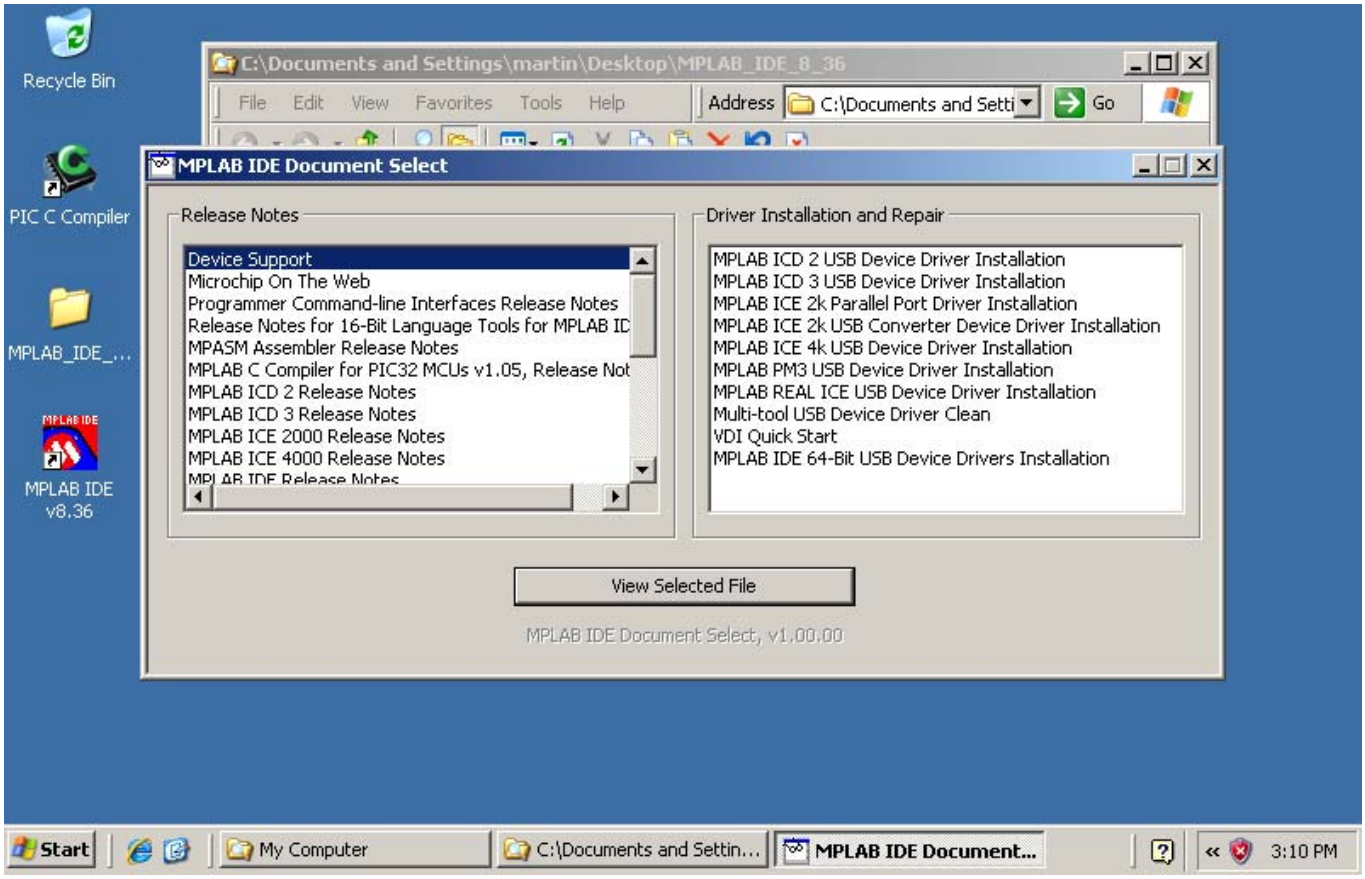

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
Start Copying Files
Review settings before copying files.
MICROCHIP
Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.
Current Settings:
Destination Directory:
C:\Program Files\Microchip\
Setup type selected: Complete
Components selected:
Serial Memory Devices
8-bit MCUs and KeeLoq devices.
16-bit MCUs and DSCs
32 bit MCUs
REAL ICE CMD
InstallShield
< Back
Next >
Cancel
Start
My Computer
C:\Documents and Settin...
MPLAB® Tools
3:08 PM

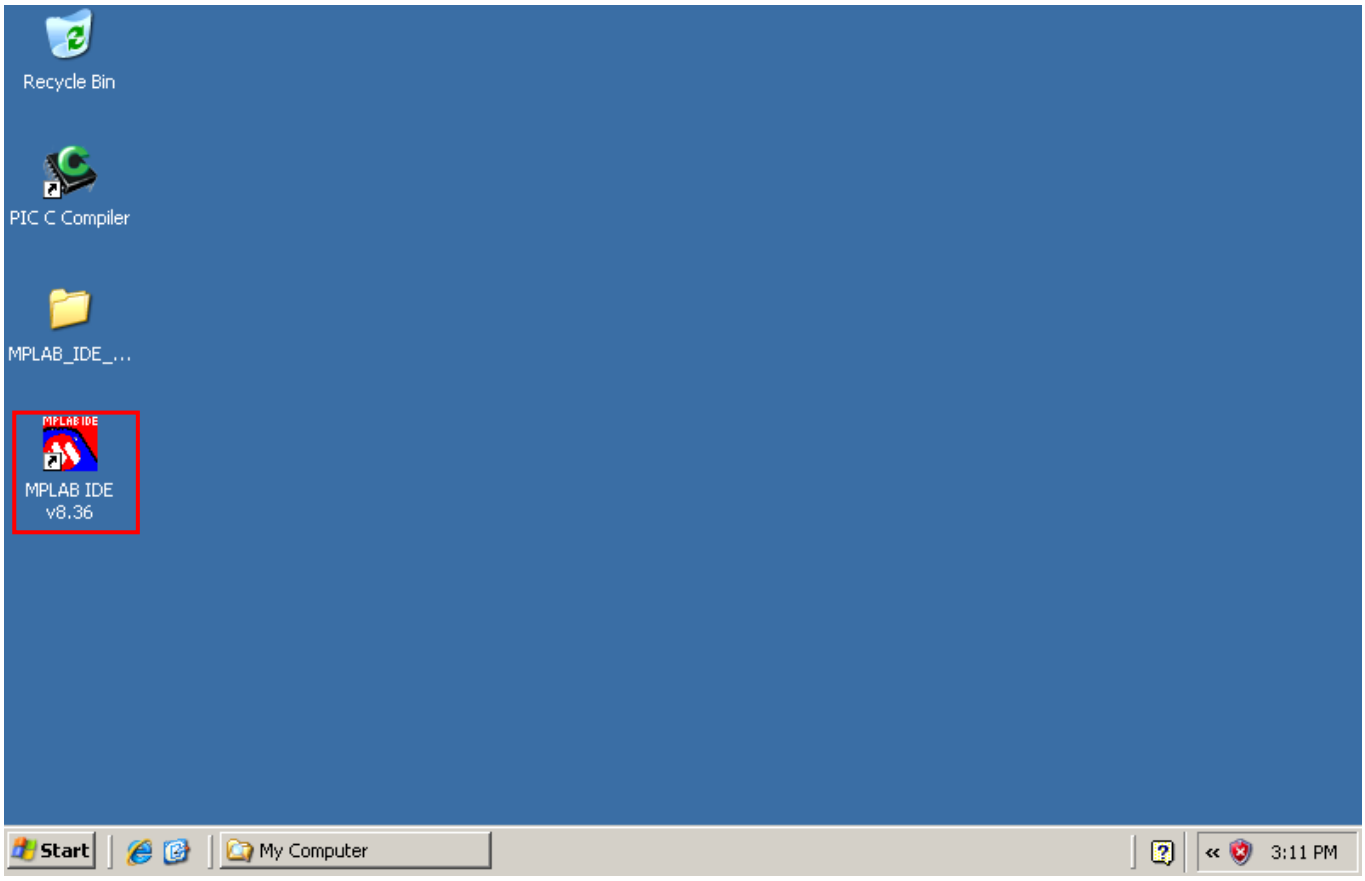

MPLAB® Tools
MPLAB® Tools
MPLAB® Tools
Setup Status
MICROCHIP
MPLAB Tools v8.36 is configuring your new software installation.
Copying new files
Cancel
Start
My Computer
C:\Documents and Settin...
MPLAB® Tools
3:08 PM



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.







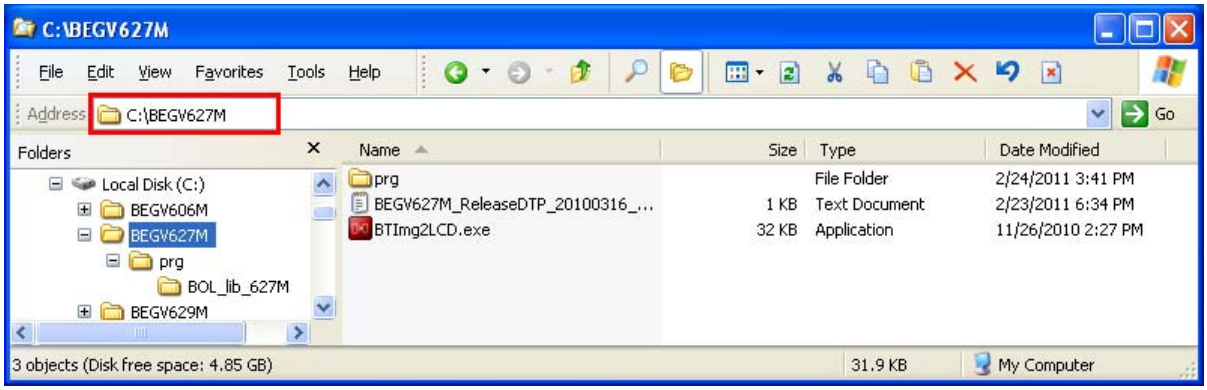
Note the **MPLAB IDE v8.36** is added on the desktop.

4.3 Install Bolymin Demo Code and library (SDK)

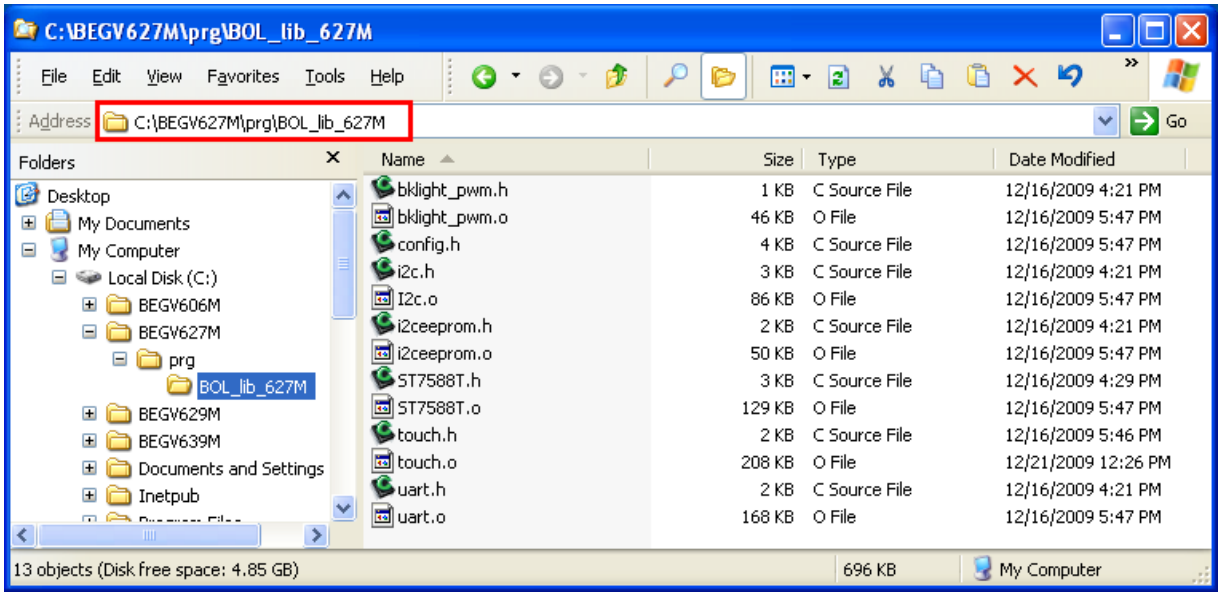
Here is the URL to download latest version of source code and libraries with pre-configured compile environment. http://www.bolymin.com.tw/embedded/BEGV627M.rar

Decompress the rar file into c:\BEGV627M and here is the file listing –

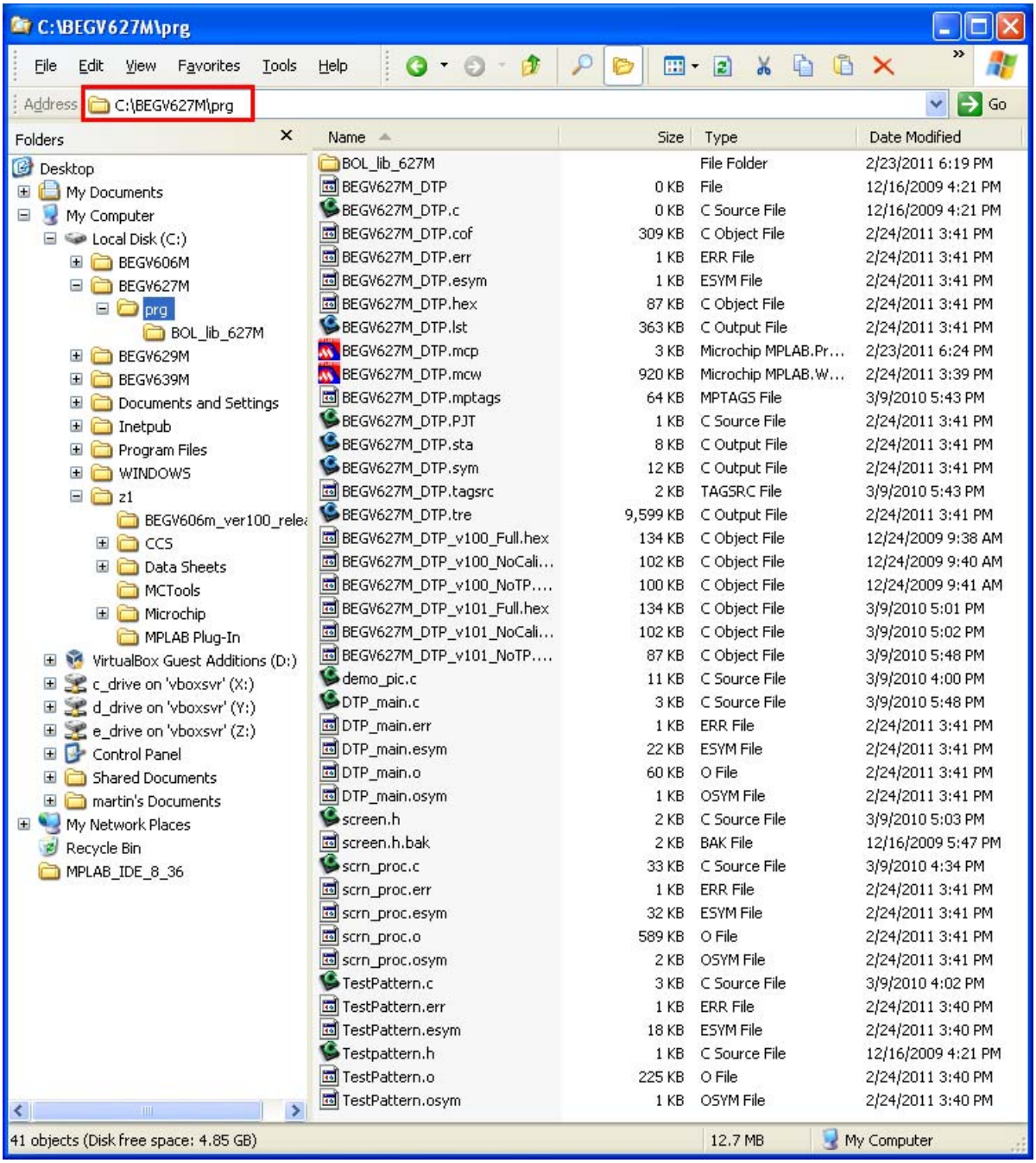
- **C:\BEGV627M** folder contains version info README file and BTImg2LCD, a bitmap font conversion utility program. Please refer to last appendix for step guide.



- 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

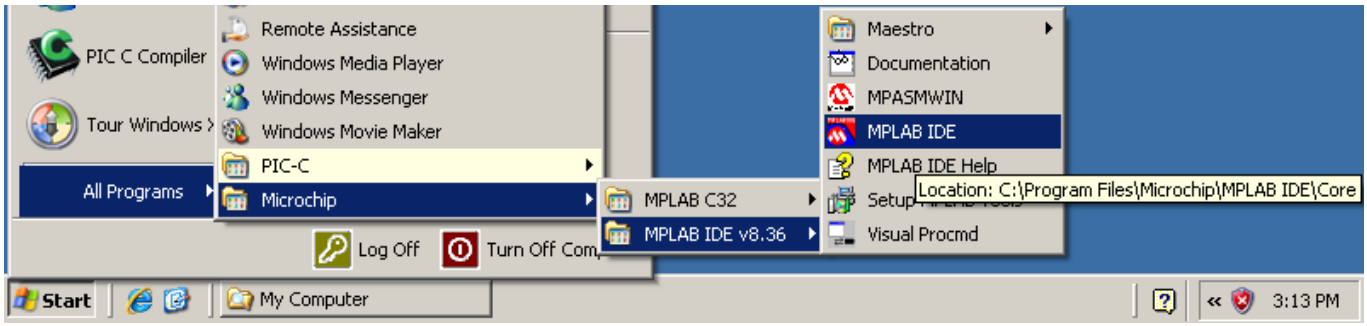


4.4 Compile source code using MPLAB IDE ver 8.36

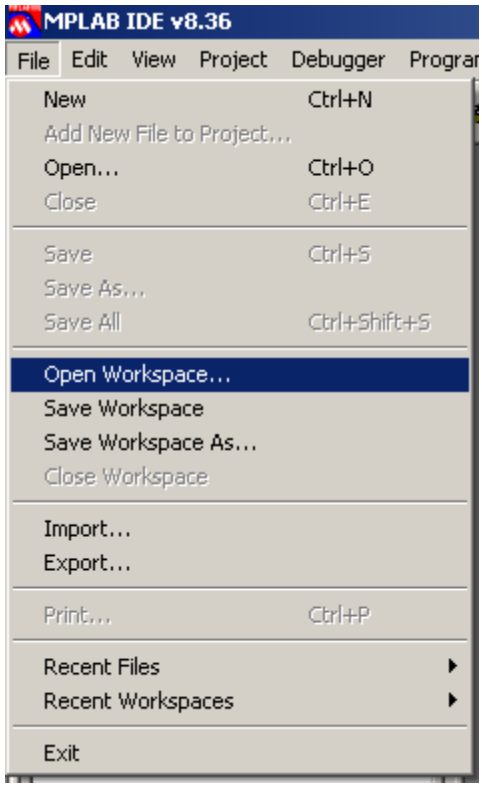
Please follow the step guide to make your 1st successful compilation.

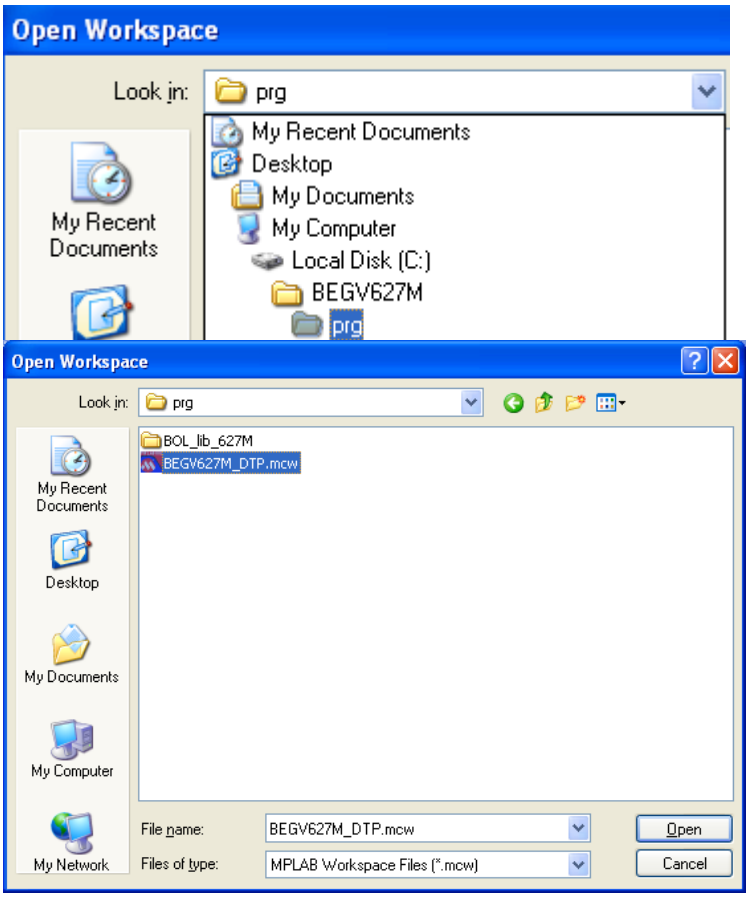
Step 1 - click left icon on desktop 

or **Start→All programs→Microchip→MPLAB IDE**

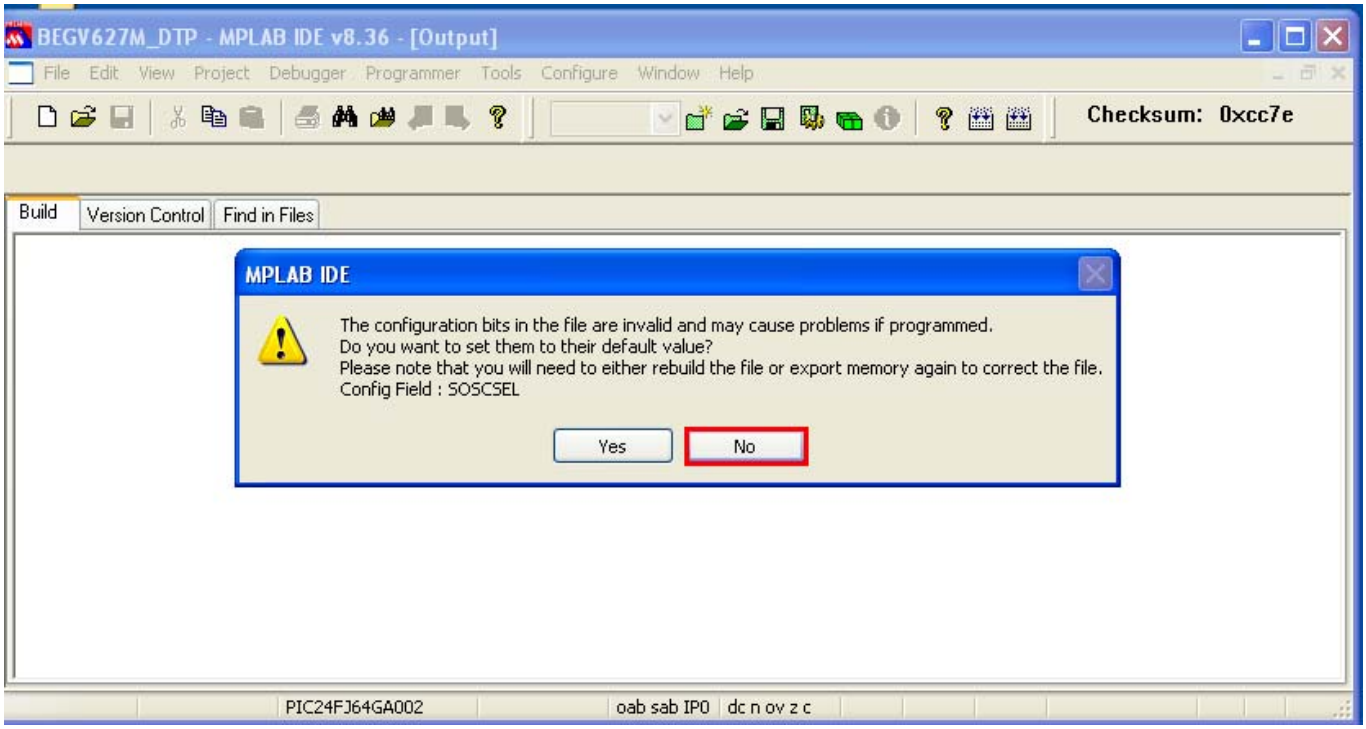


Step2 – Click **File-Open Workspace** – browse into **c:\BEGV627M\prg** folder , and open **BEGV627_DTP.mcw**

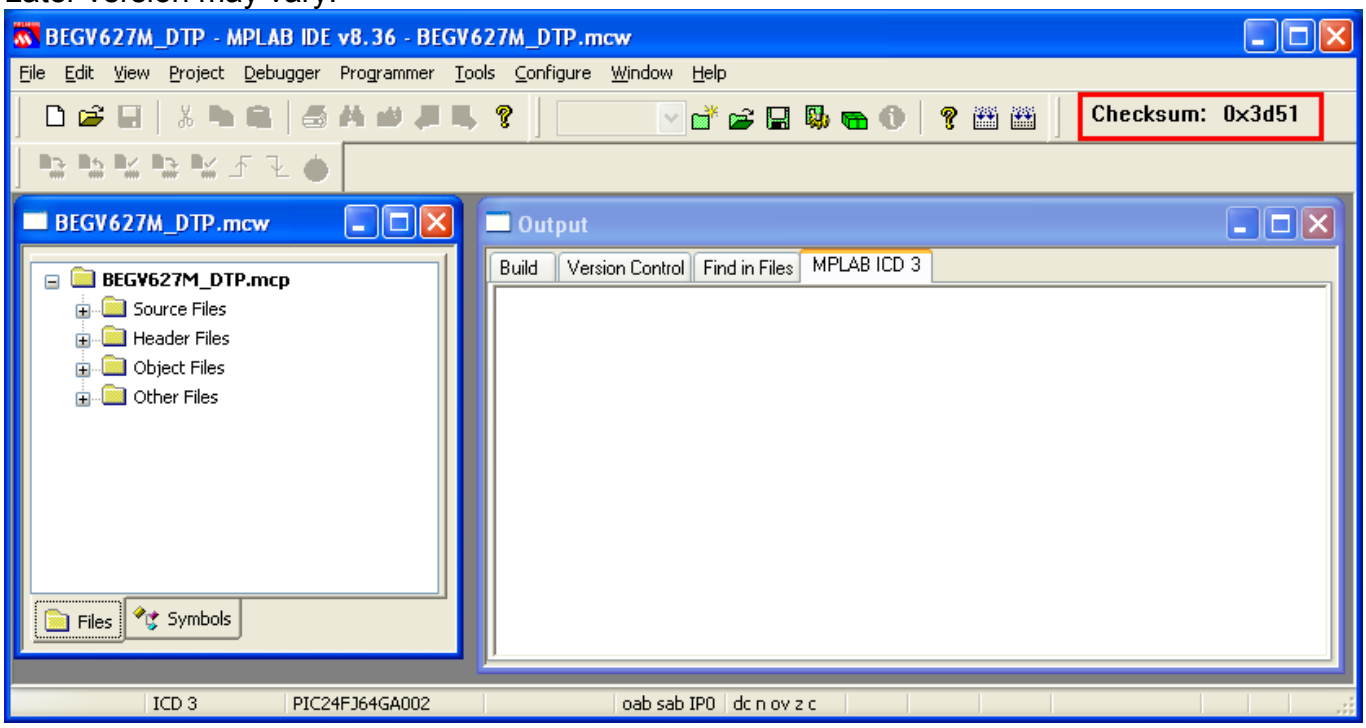




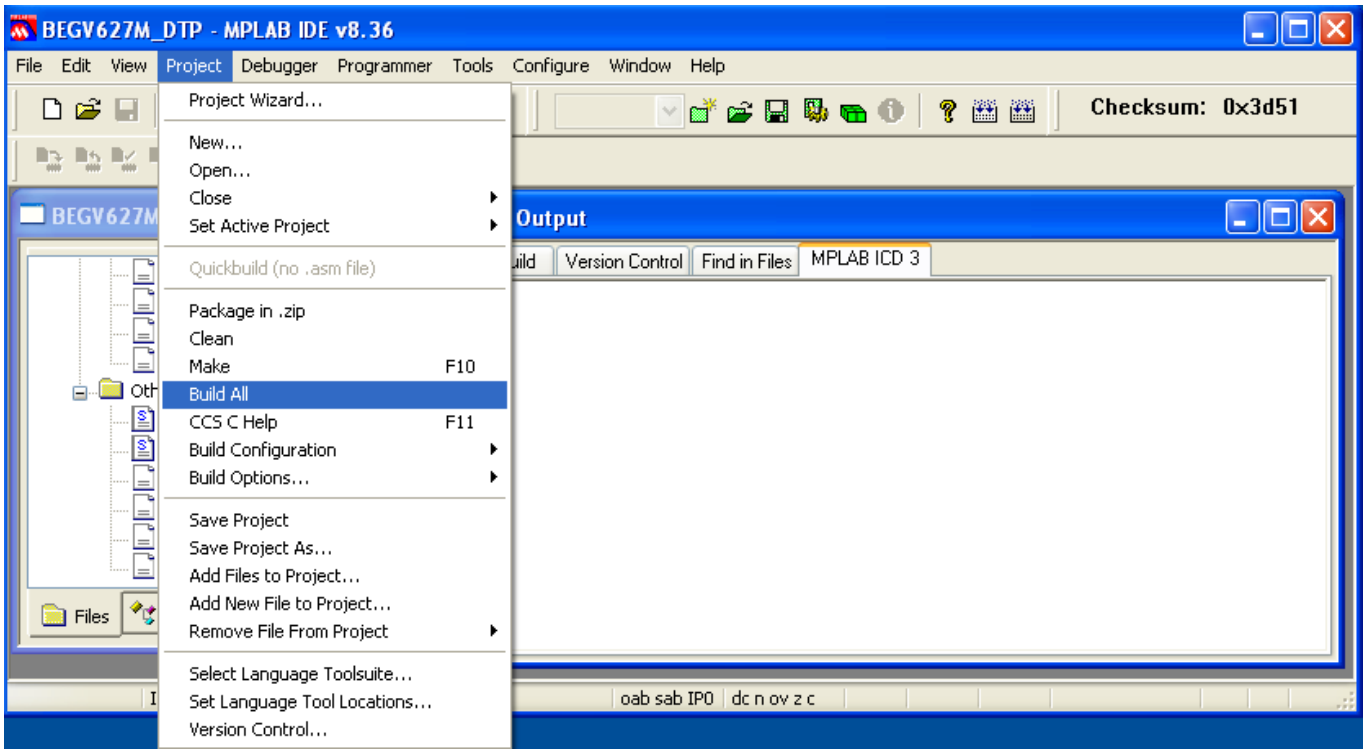
Step3 – Click **No**. The configuration bits setting is written in C source code already and belay the warning message..



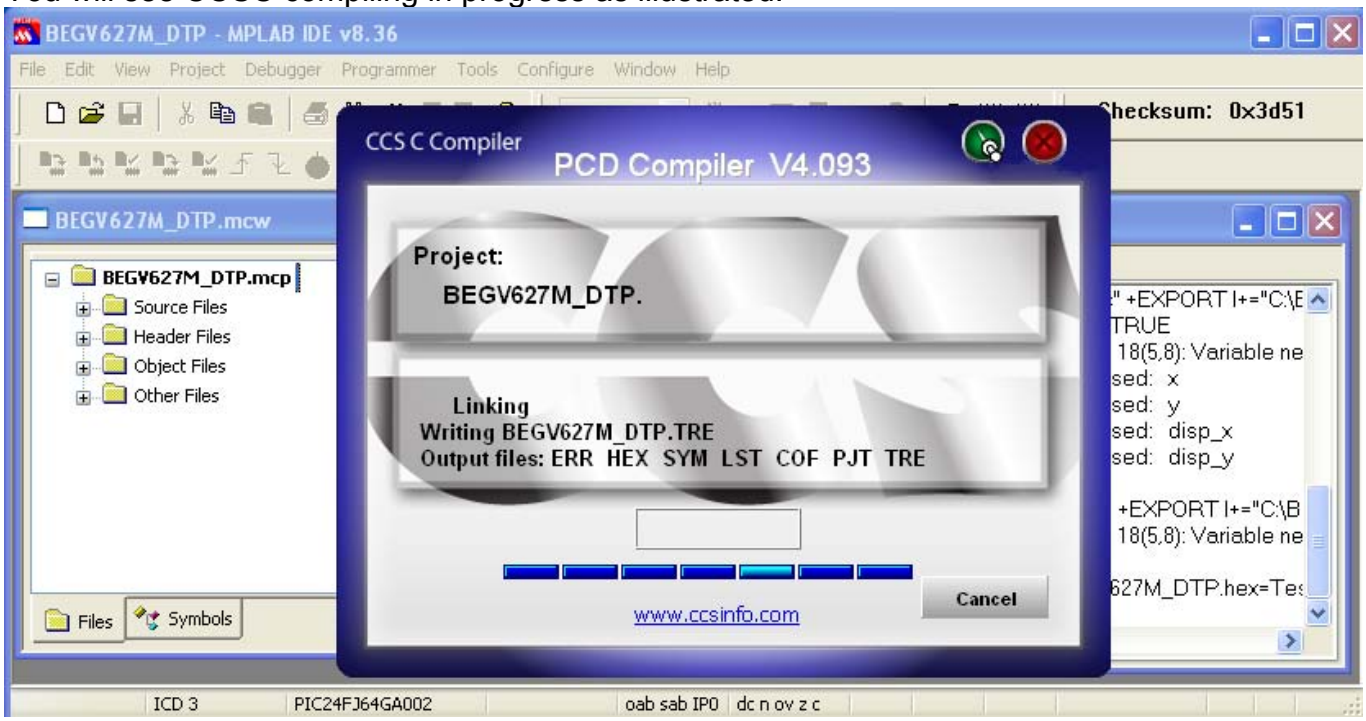
Step 4 – Your screen should be like this. Mind the checksum for version ver 1.00 source code. Later version may vary.



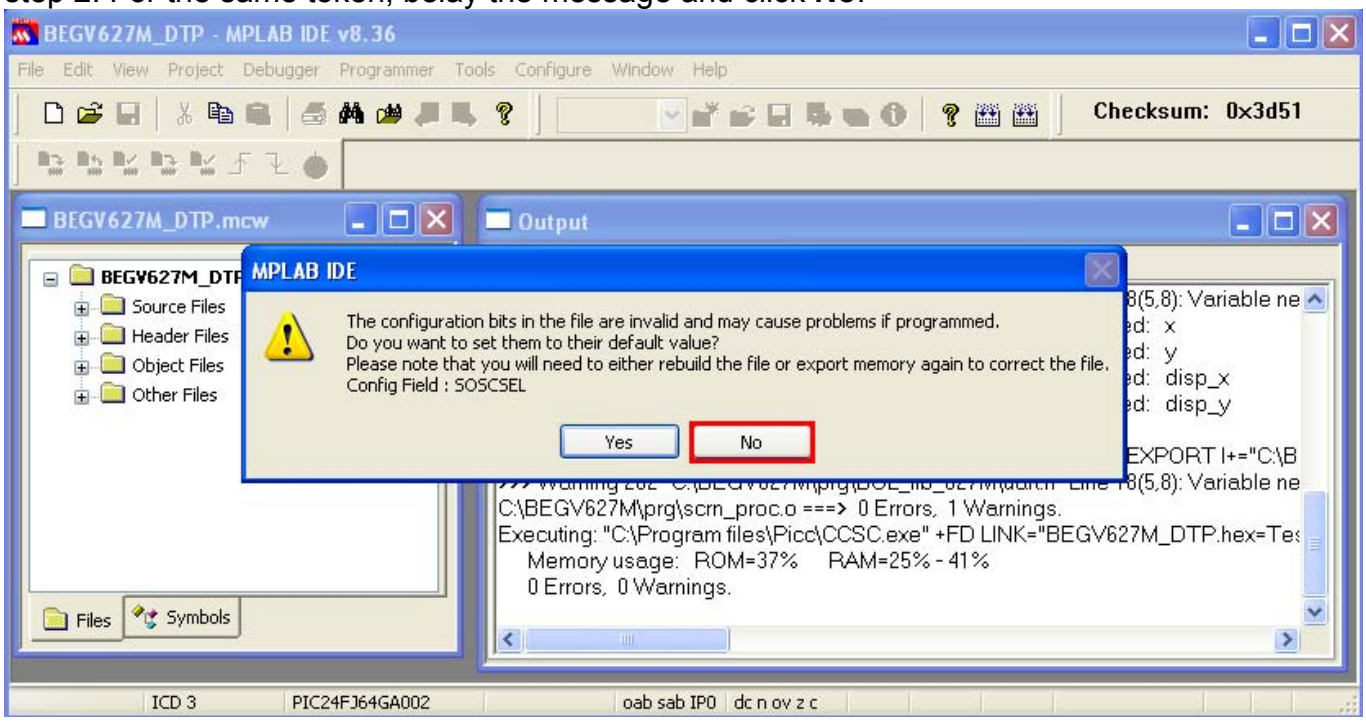
Step 5 – Now you are all set for 1st compilation. Click **Project-Build all**.



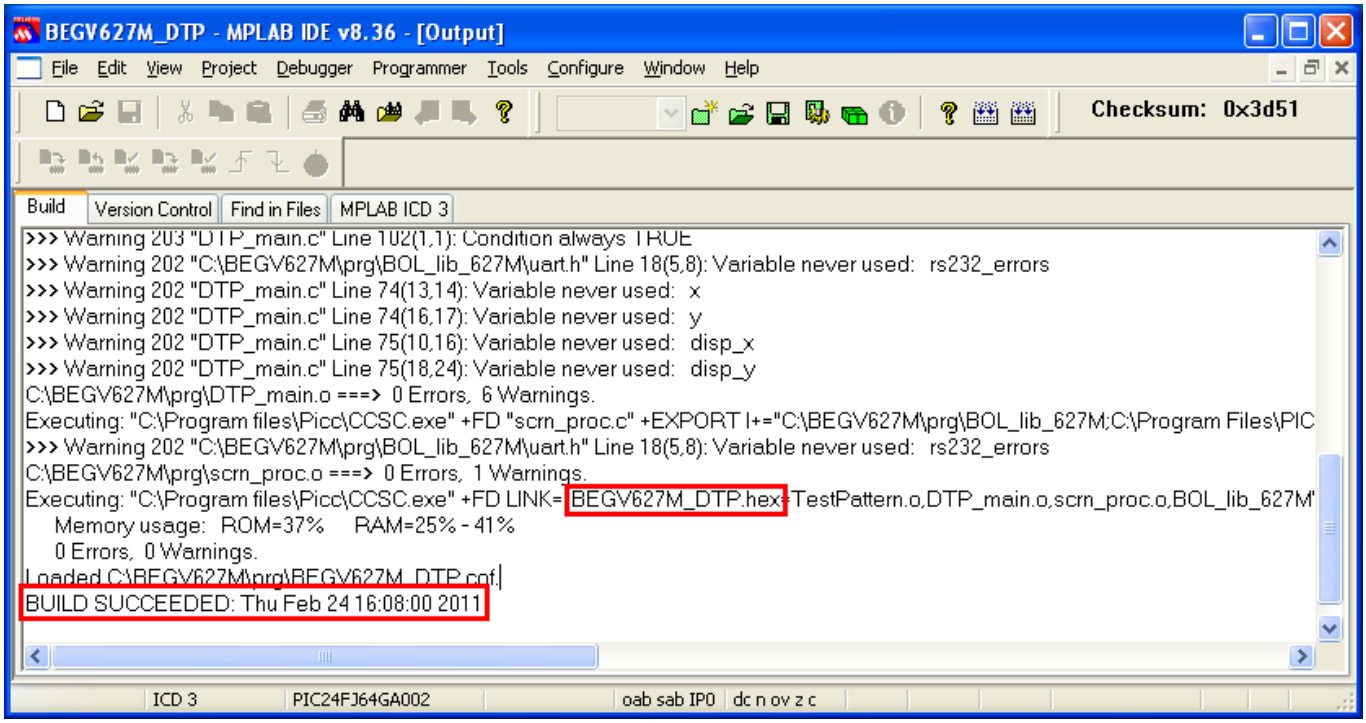
You will see CCSC compiling in progress as illustrated.



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**.

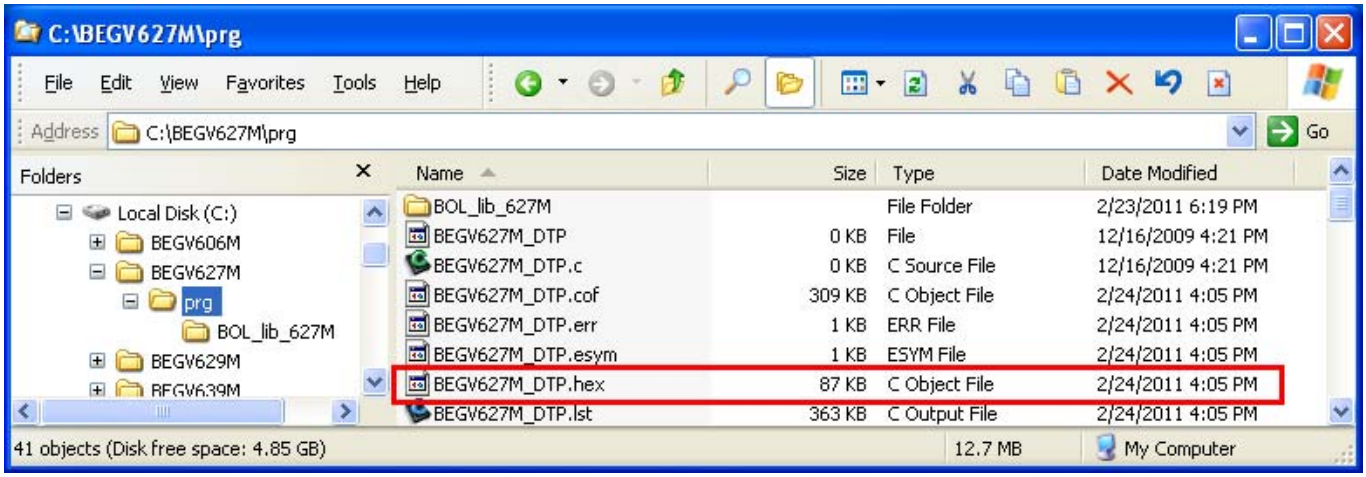


Step 7 – check the compilation result – go to output window by clicking **Windows-Output**, maximize it, and read the last line – **BUILD SUCCEEDED**. Congratulations for what you have achieved this far.

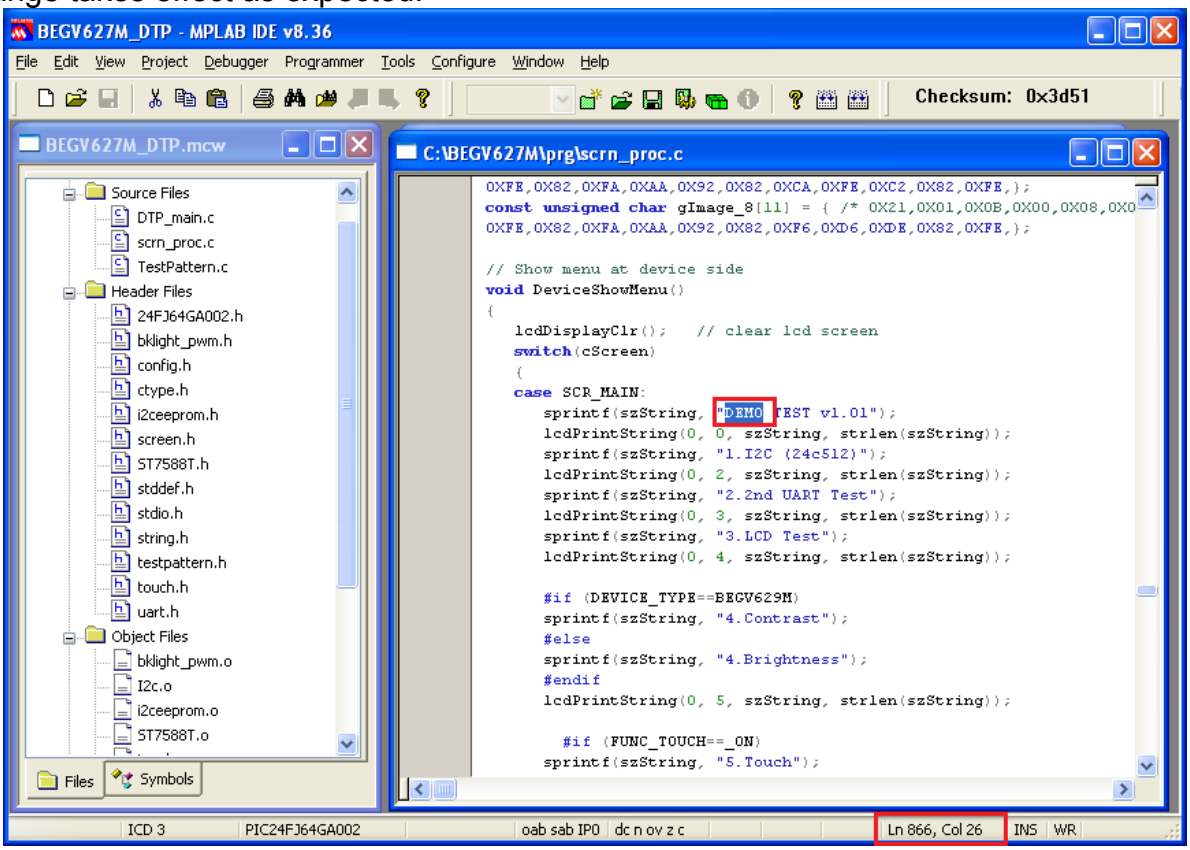


Also note that a new BEGV627M_DTP.hex file, the program HEX code for EEPROM programming, is generated. In the next section, we will introduce you the steps for writing EEPROM HEX code to target device using ICD3.

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 “**Moon**“ for instance. Then recompile to generate new program HEX and write it into EEPROM. You will see the change takes effect as expected.

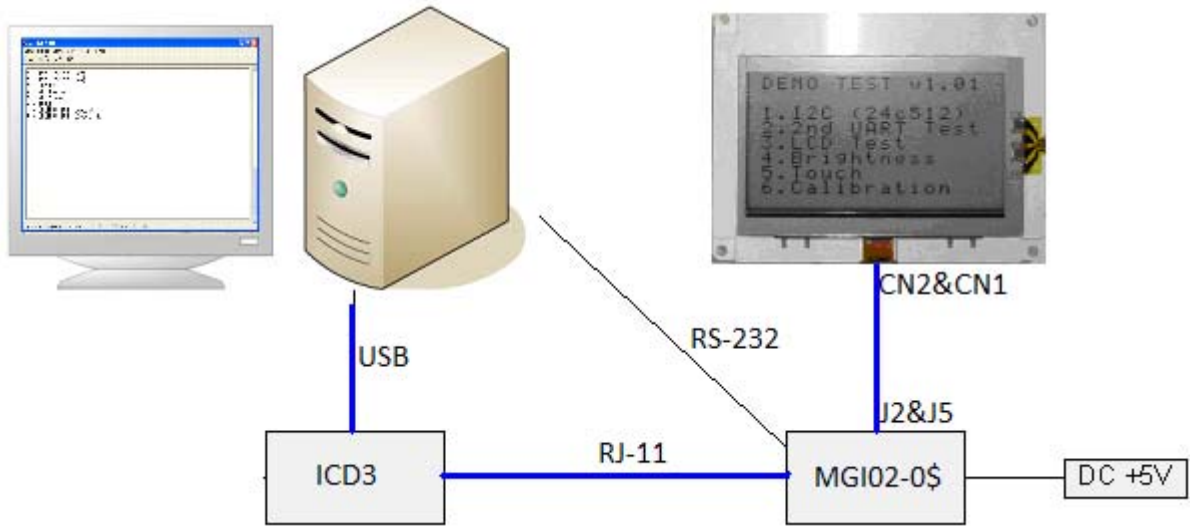


4.5 Programming HEX code using ICD3

** 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, BEGV627M (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

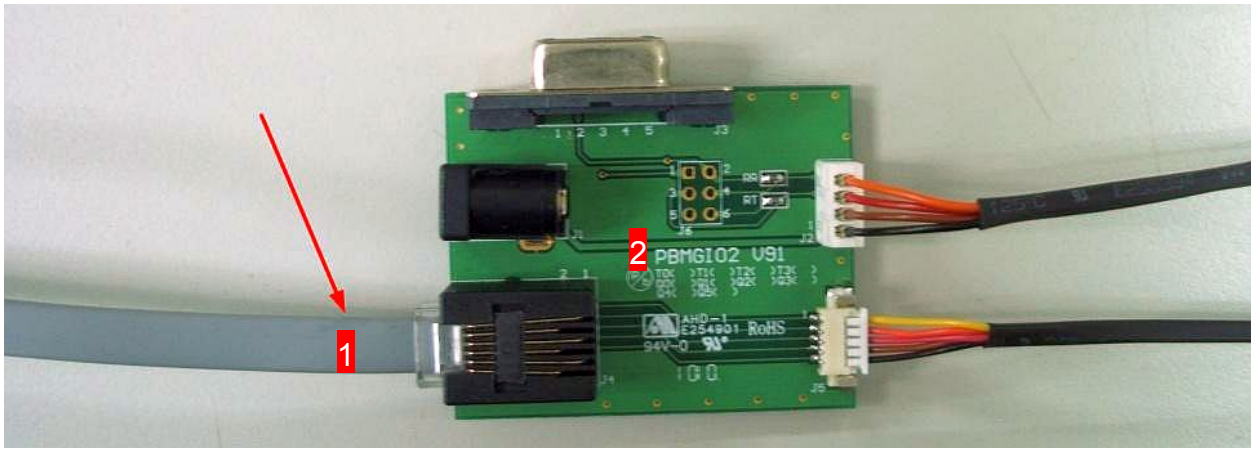
#	HW Item	Qty	Description
1	PC with COM port	1	Requires RS-232 port
2	Programmer-ICD3	1	
3	Converter board	1	MGI02-0$
4	USB cable	1	Type A male←---→ Type B male
5	RS-232 ext. cable	1	Straight through RS-232 extension cable DB-9 maile-DB-9 male; or use a USB-serial cable

Software Requirements

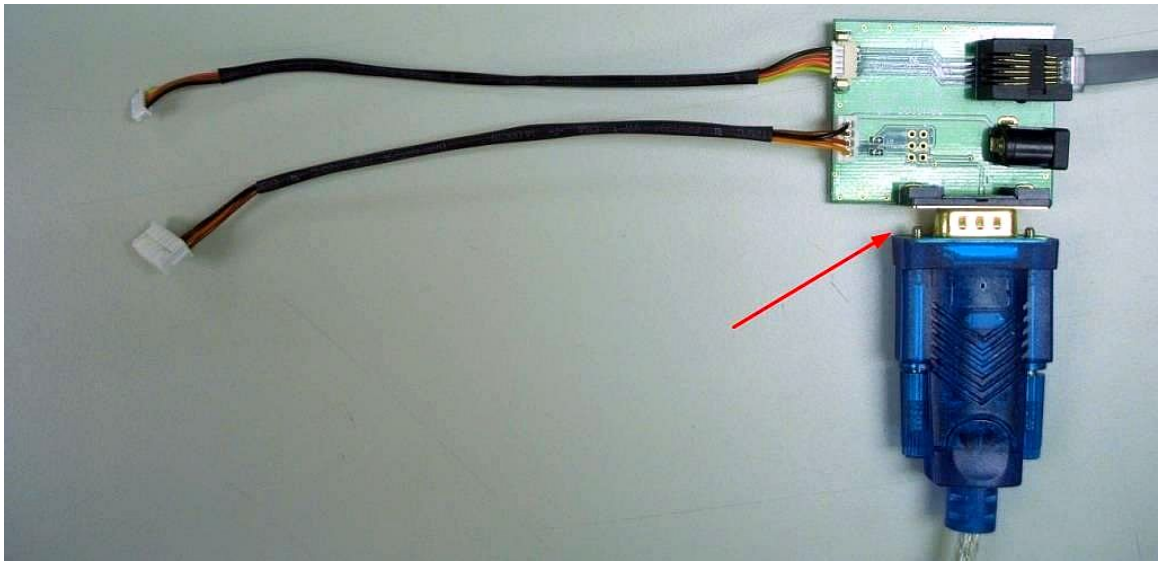
#	SW item	Qty	Description
1	Programming IDE	1	MPLAB IDE 4.83 by Microchip
2	Terminal Emulation	1	Hyper Terminal (XP build in)
3	Bolymin DEMO program	1	Demo source code, DLL, and program HEX

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

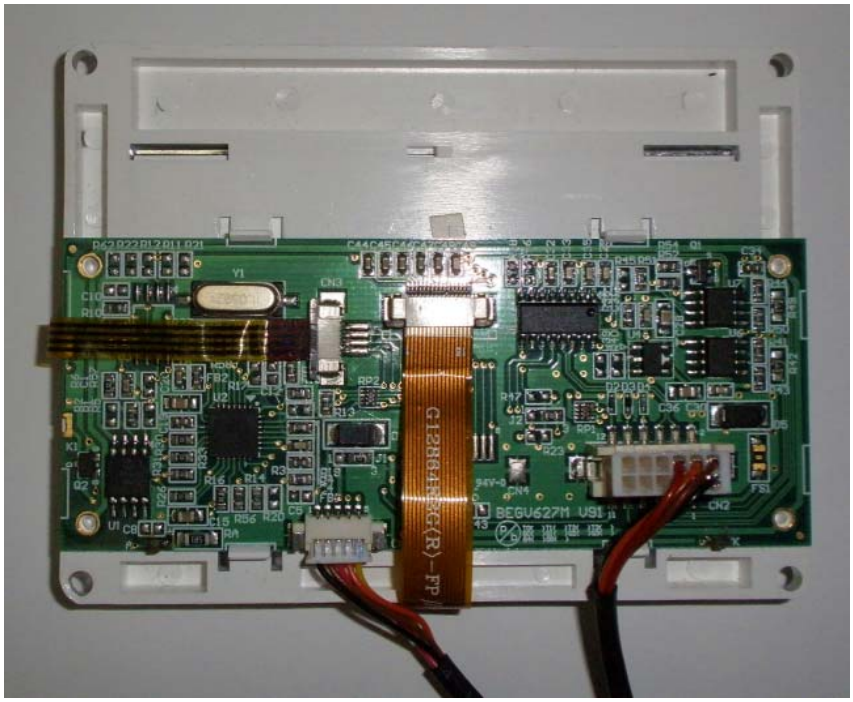
Step 2 - Connect 1 RJ-11 and cables to 2 MGI02-0$



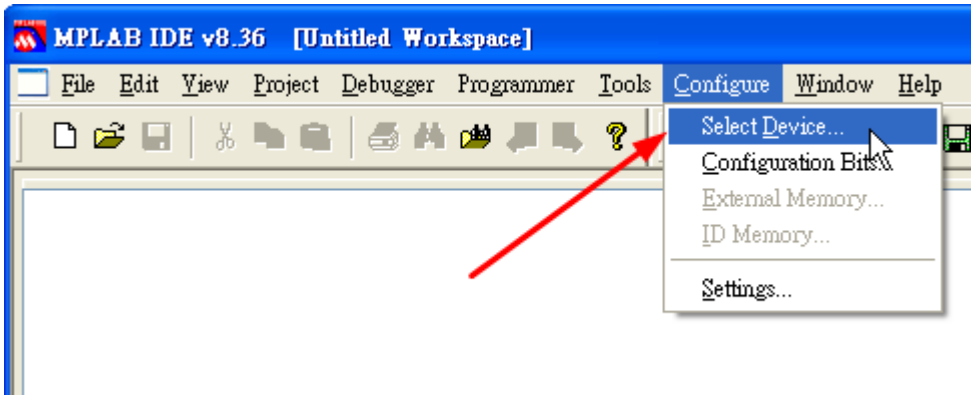
Step 3 - Connect MGI02-0$ to PC using a RS-232 cable



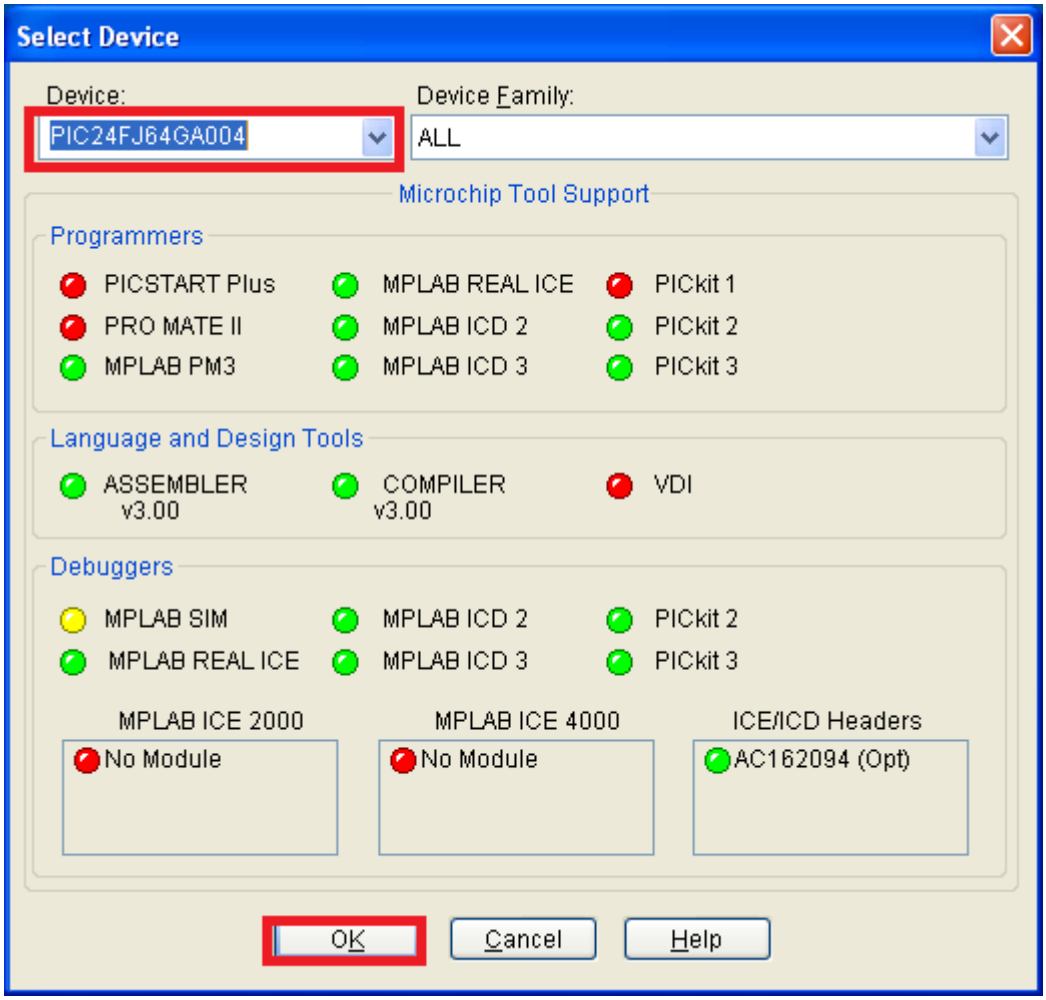
Step 4 - Connect MGI02-0$ to BEGV627M.



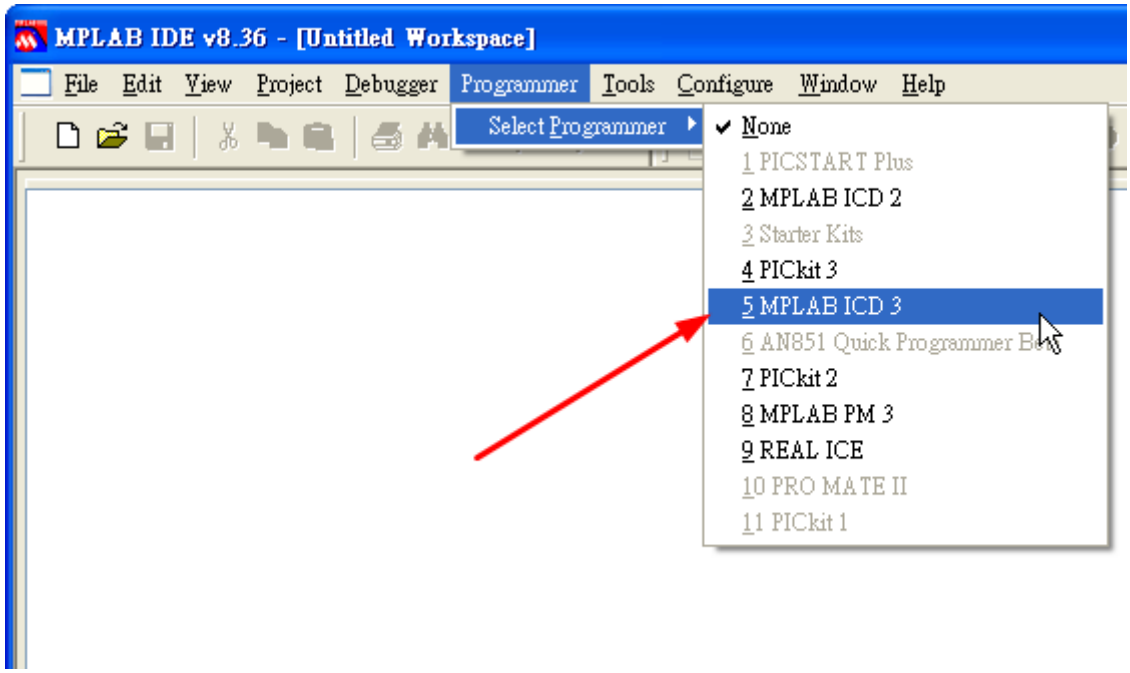
Step 5 - Open **MPLAB IDE v8.36**，Click on **Configure-Select Device**



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

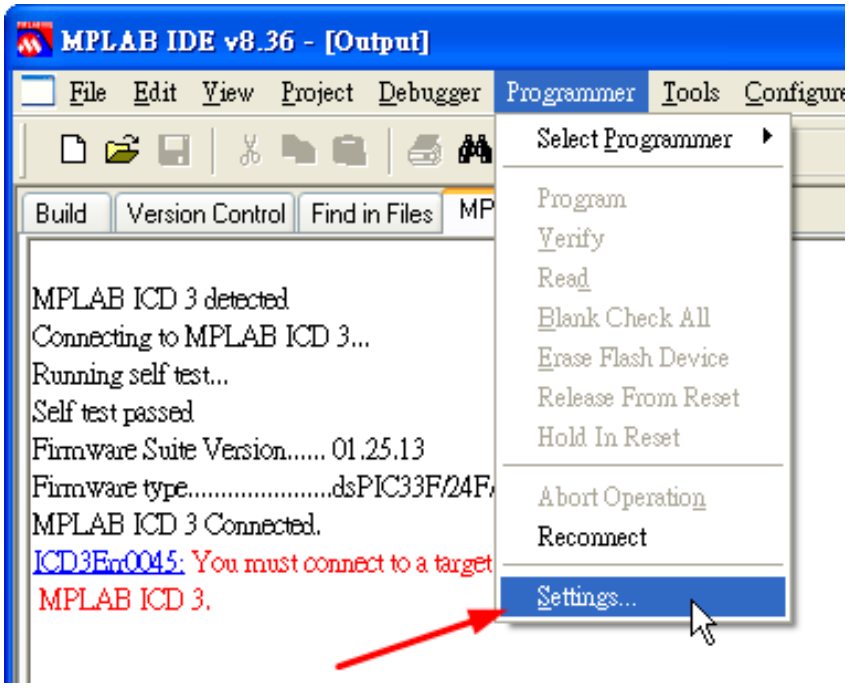


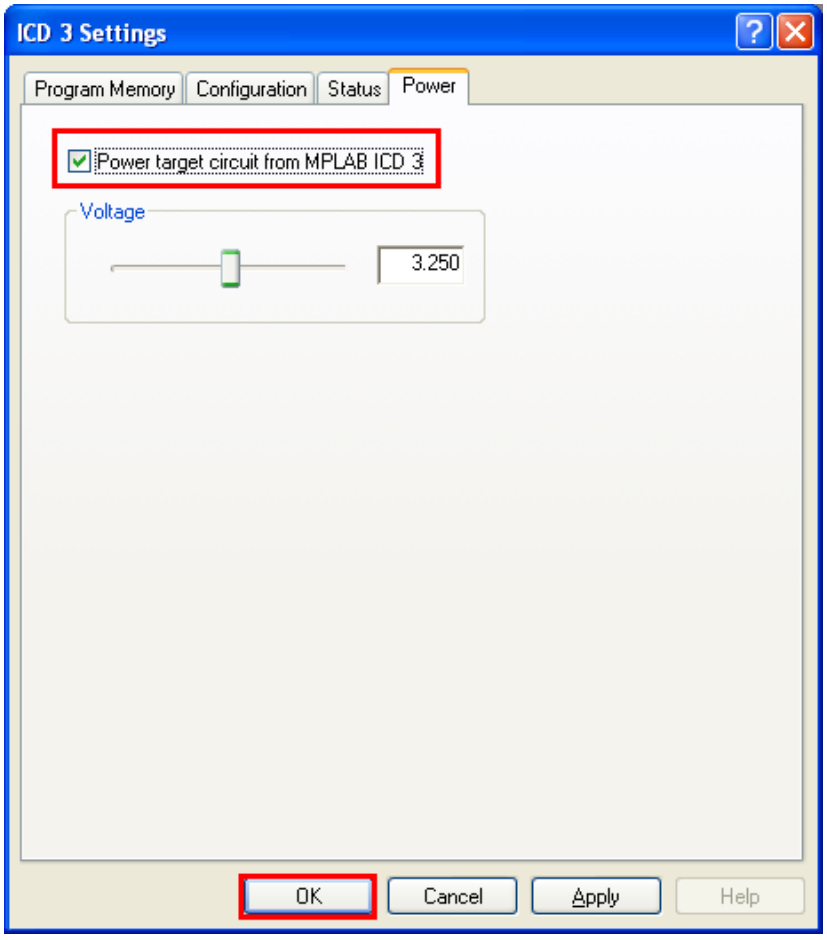
Step 7 – click **Programmer-Select Programmer - 5 MPLAB ICD3.**



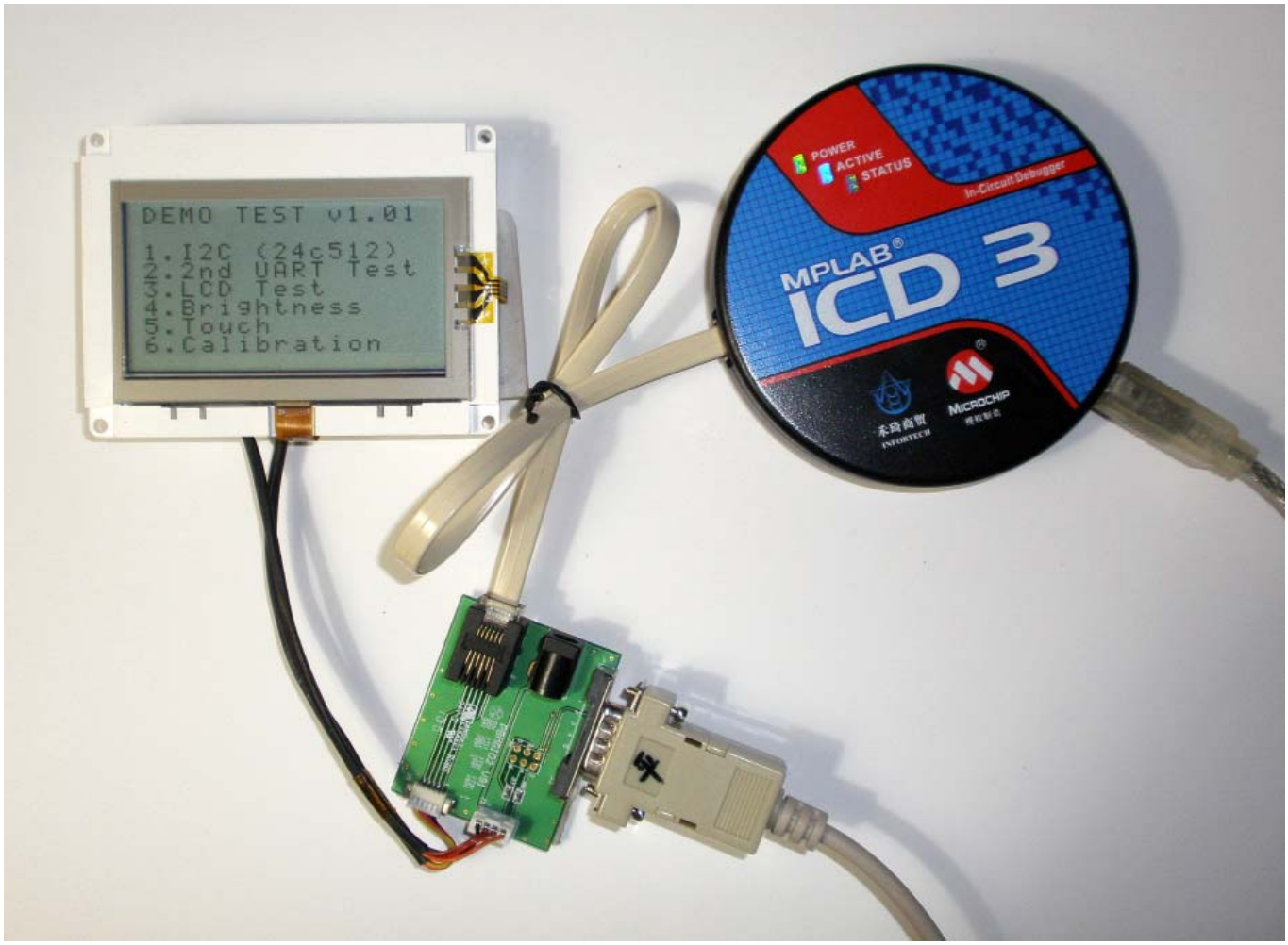
Note target device is not powered as yet. Therefore there is no display on LCD.

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.

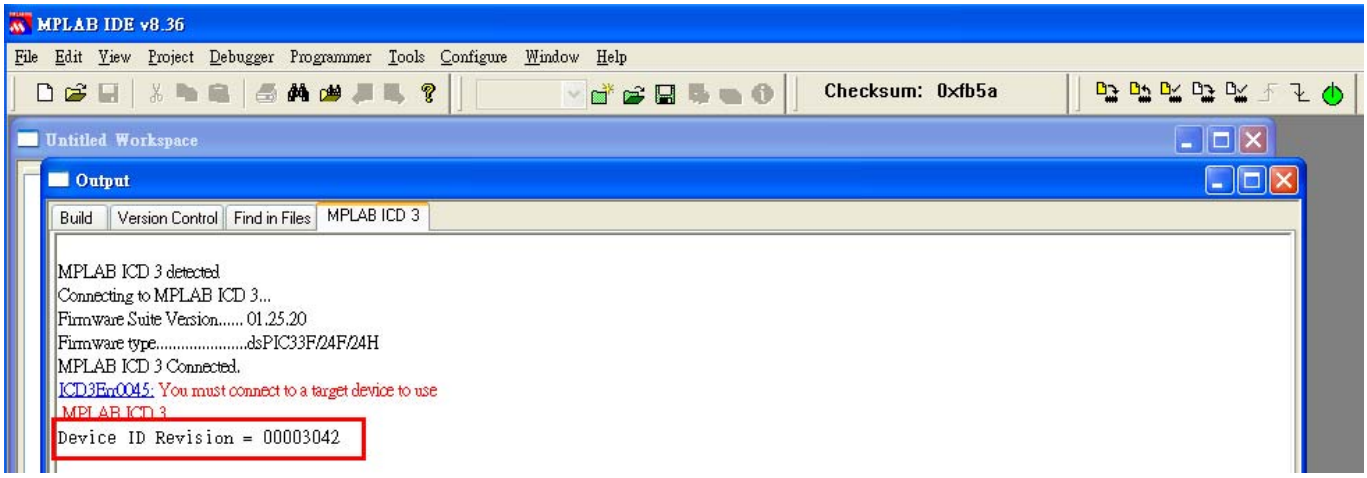




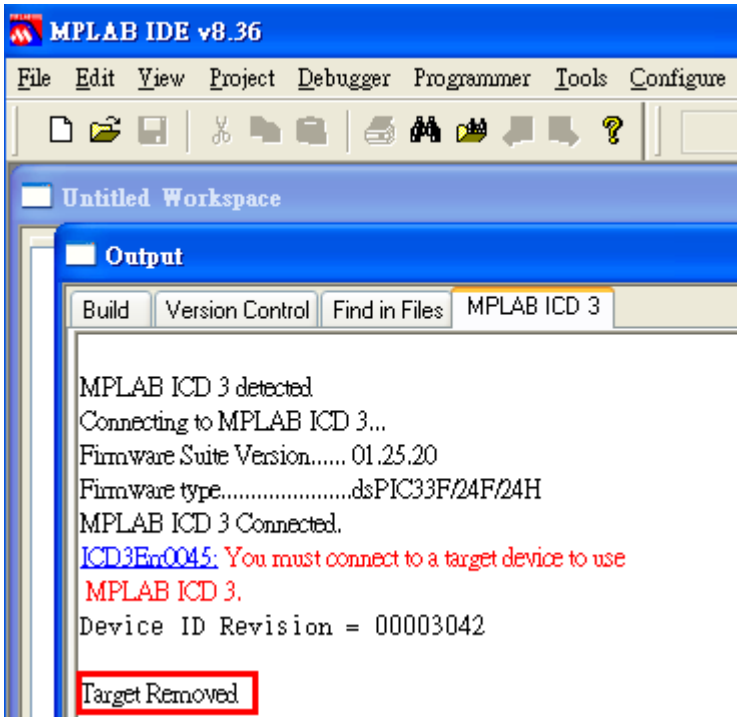
Step 9 - As soon as the power setting is done, note target device is now powered by ICD3 and LCD will display the default demo program as illustrated. Also on ICD3, observe the blue Active LED is on.



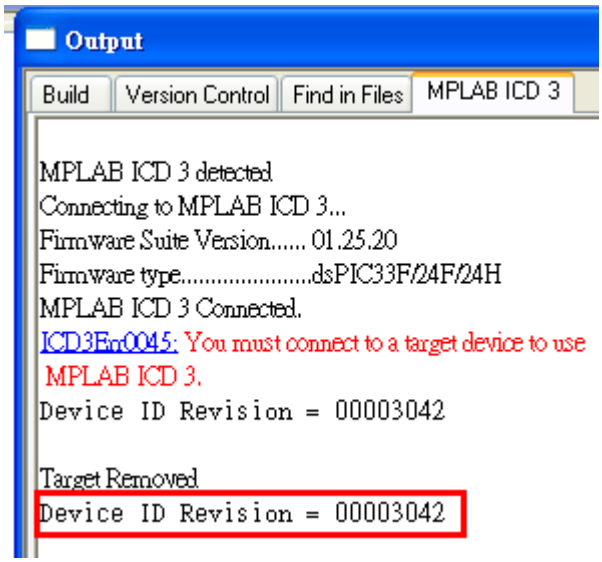
Notice a **Device ID Revision** will display on the last line of output window.



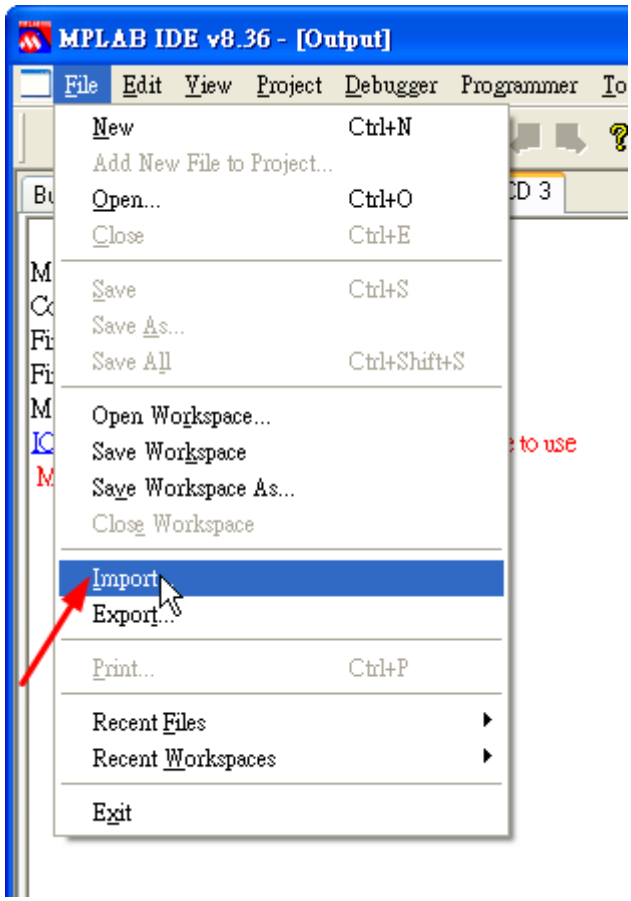
Step 10 - User may try toggle the power target device from ICD3 and observe the output windows message as follows, blue LED on ICD3 is off – 627M LCD display is off.



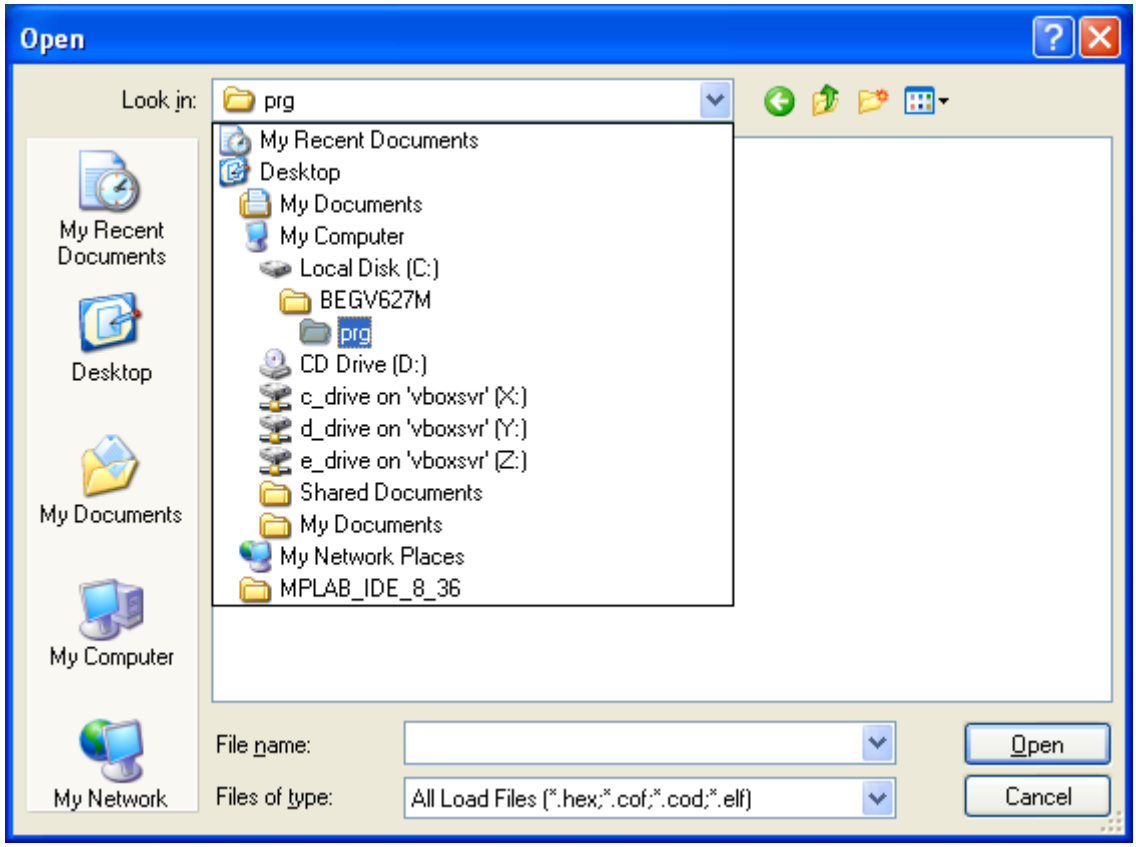
Now switch back to powered by ICD3.



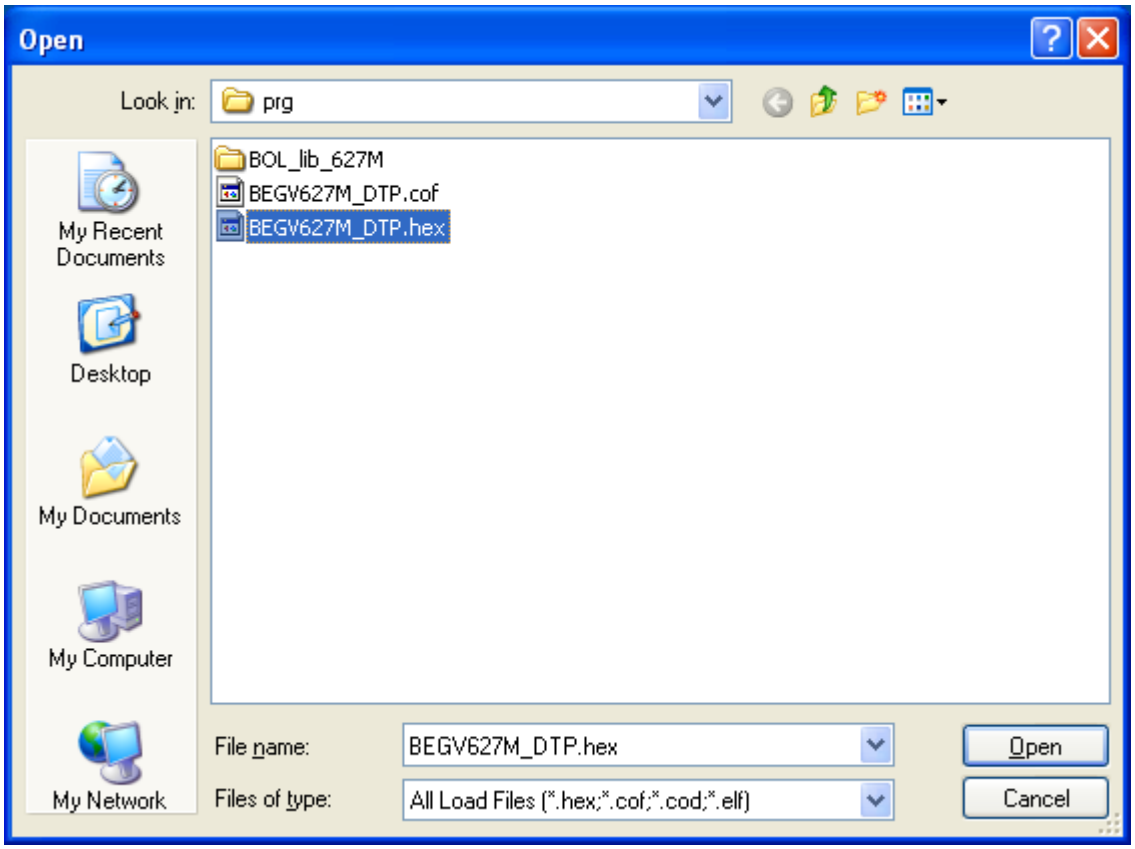
Step 11 - Select the Hex code for programming.



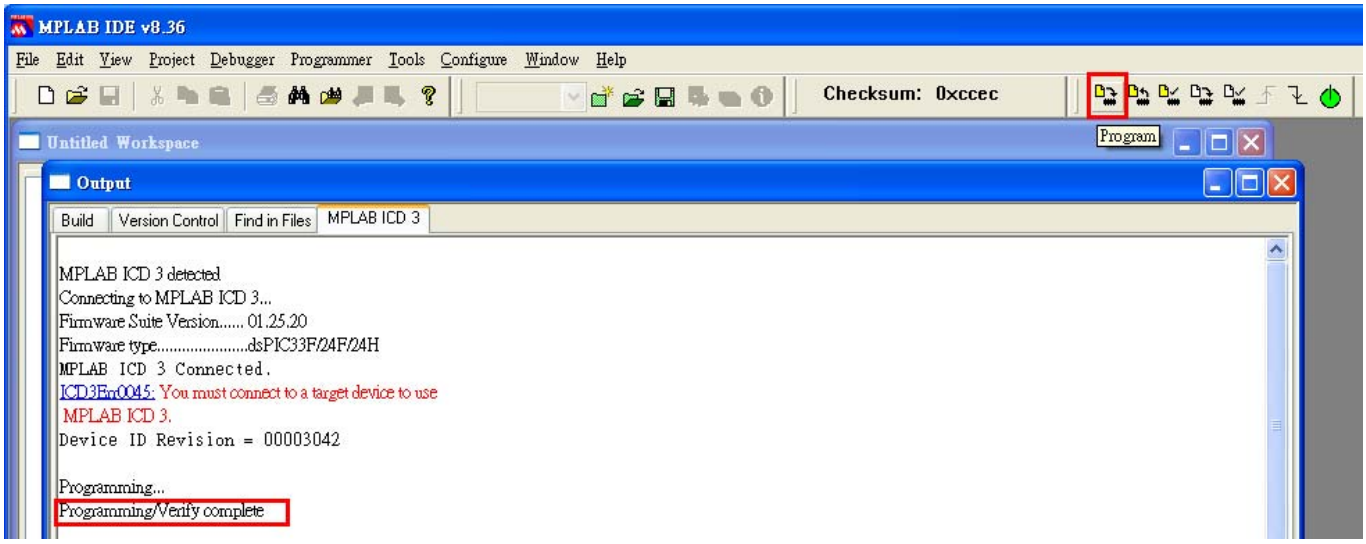
Browse into **c:\BEGV627M\prg** folder



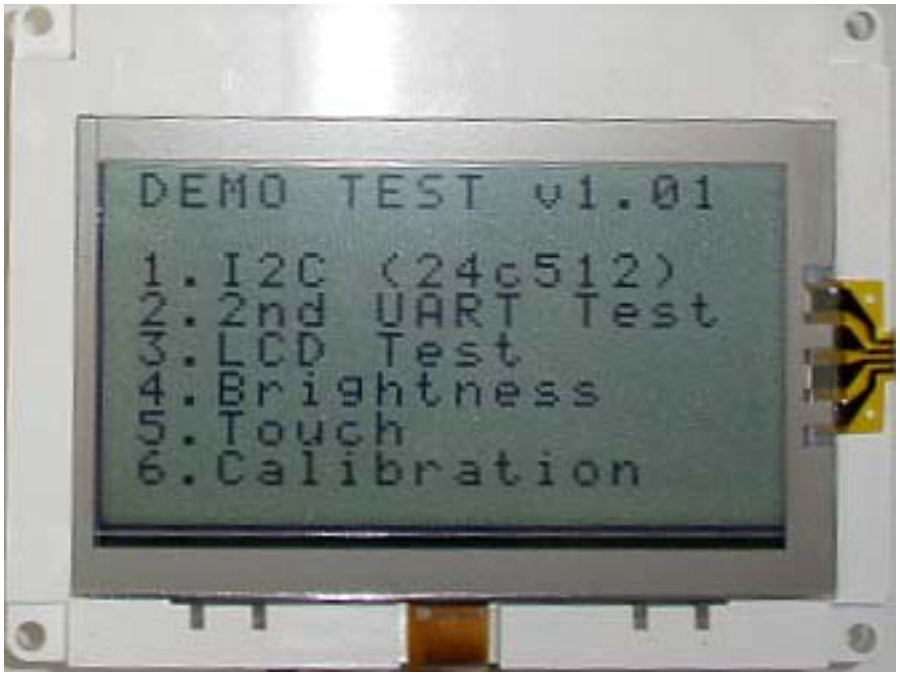
And select the HEX file



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.

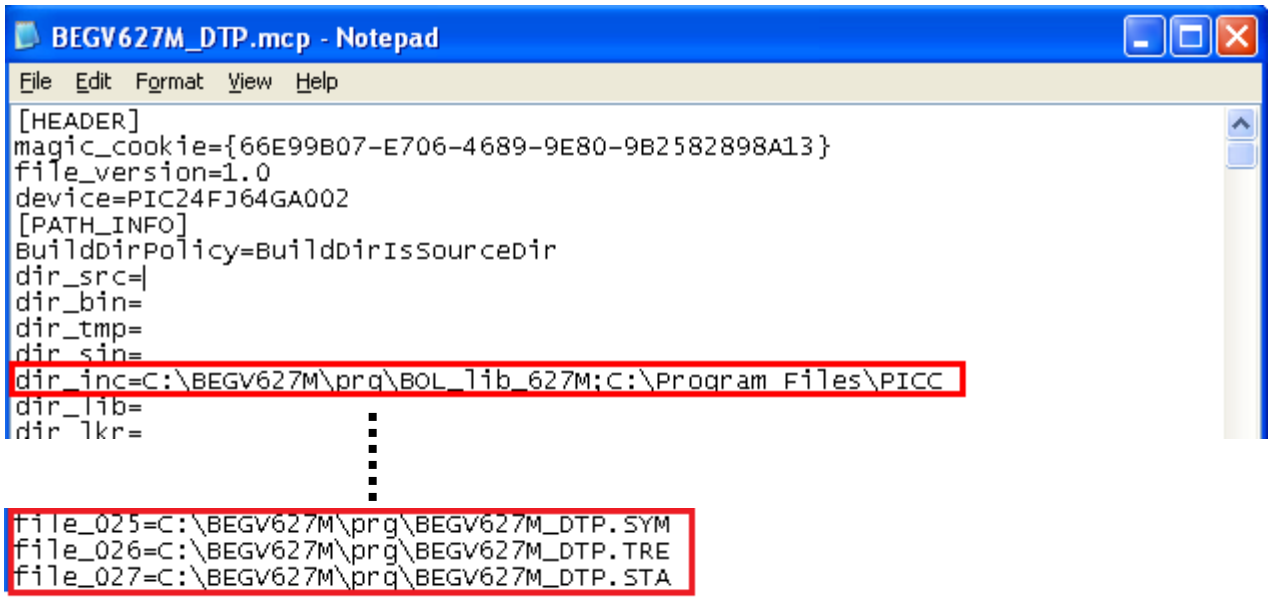


Step 13 - After HEX code programming, target device is reset and BEGV627M LCD should display as follows:



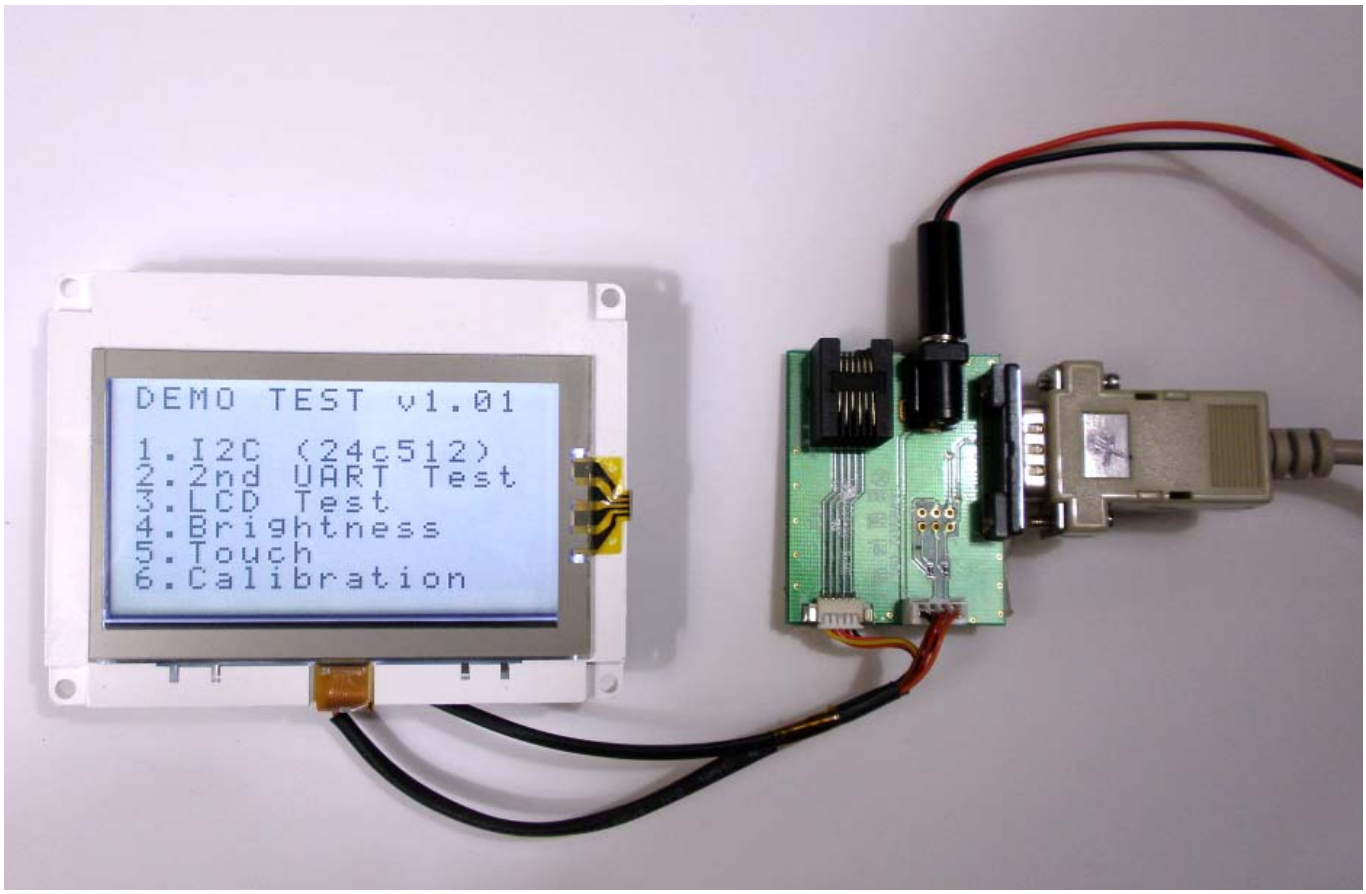
For more detail operation about ICD3, please also refer to Micrcochip URL:
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en537580&redirects=icd3

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.



4.6 Running Hyper Terminal

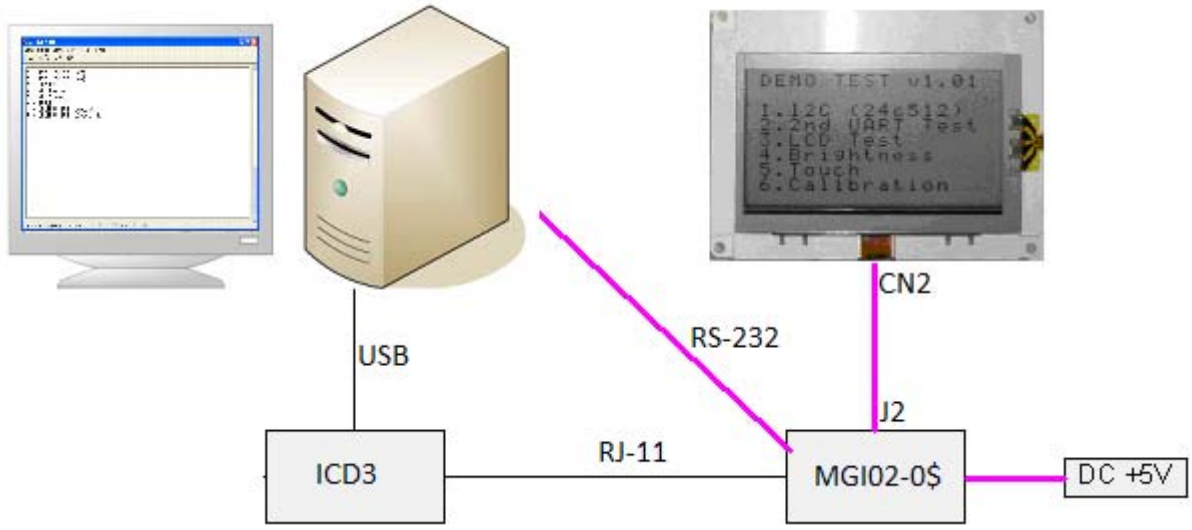
After EEPROM programming is done, disconnect ICD3, and power target device using 5Vdc. Now connect RS-232 from convert board to XP/Win7 PC. (Photo applies configuration #1, see next page)



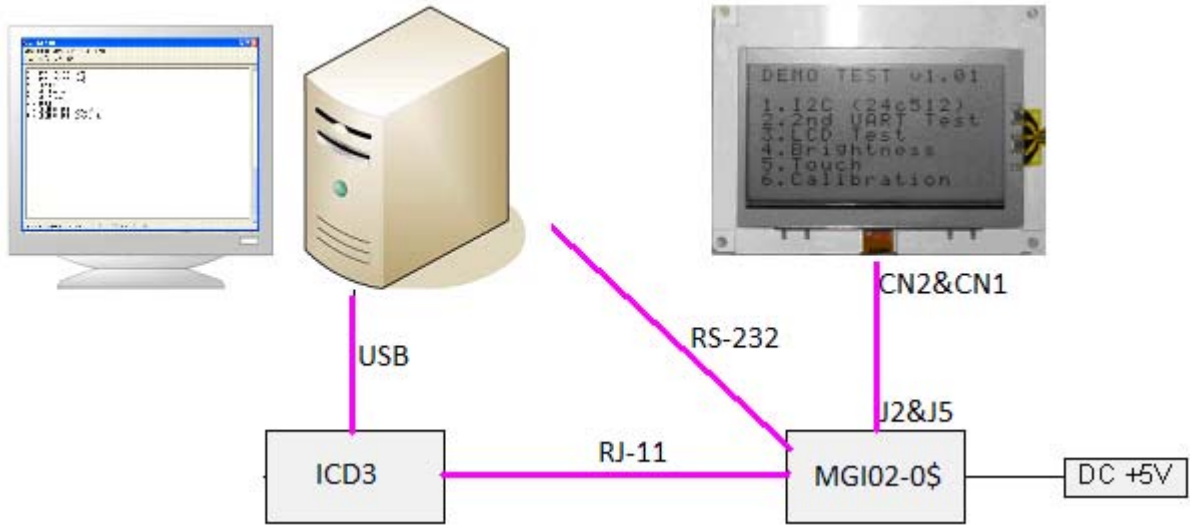
. Connect **CN2** of 627M to **J2** of MGI$02 converter board. At least 4 pins are required as marked.

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.



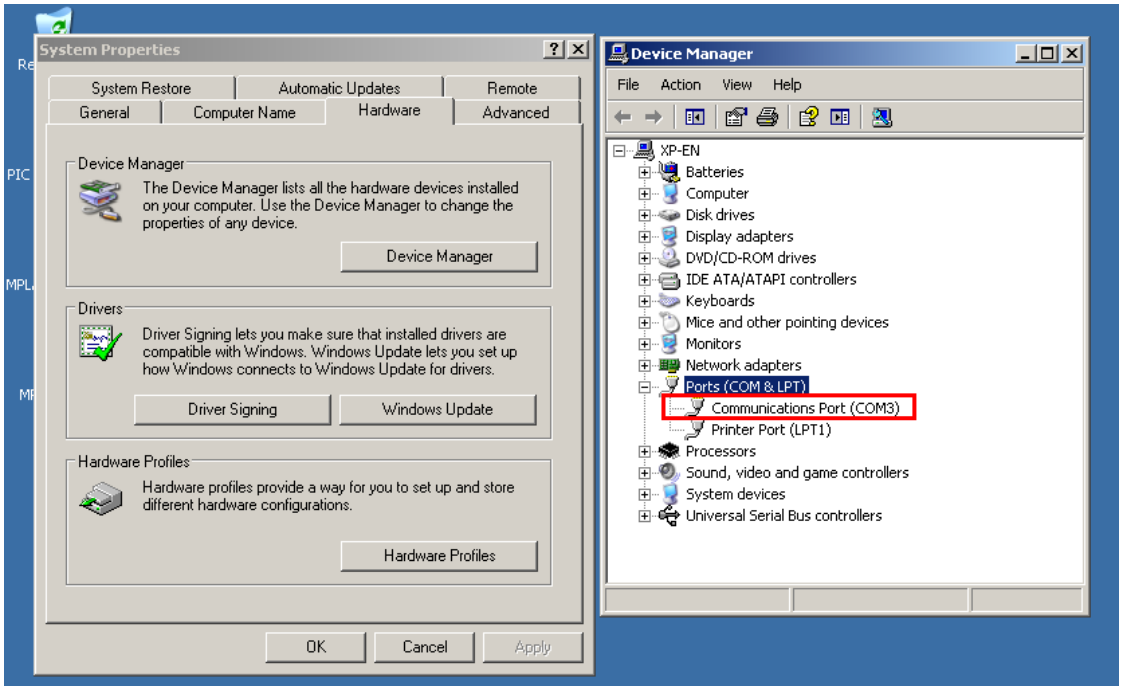
Either configuration works, configuration #1 might be easier if there is no source code change involved. Also configuration #1 provides a 5 volt rather than 3.3volt on target device.

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

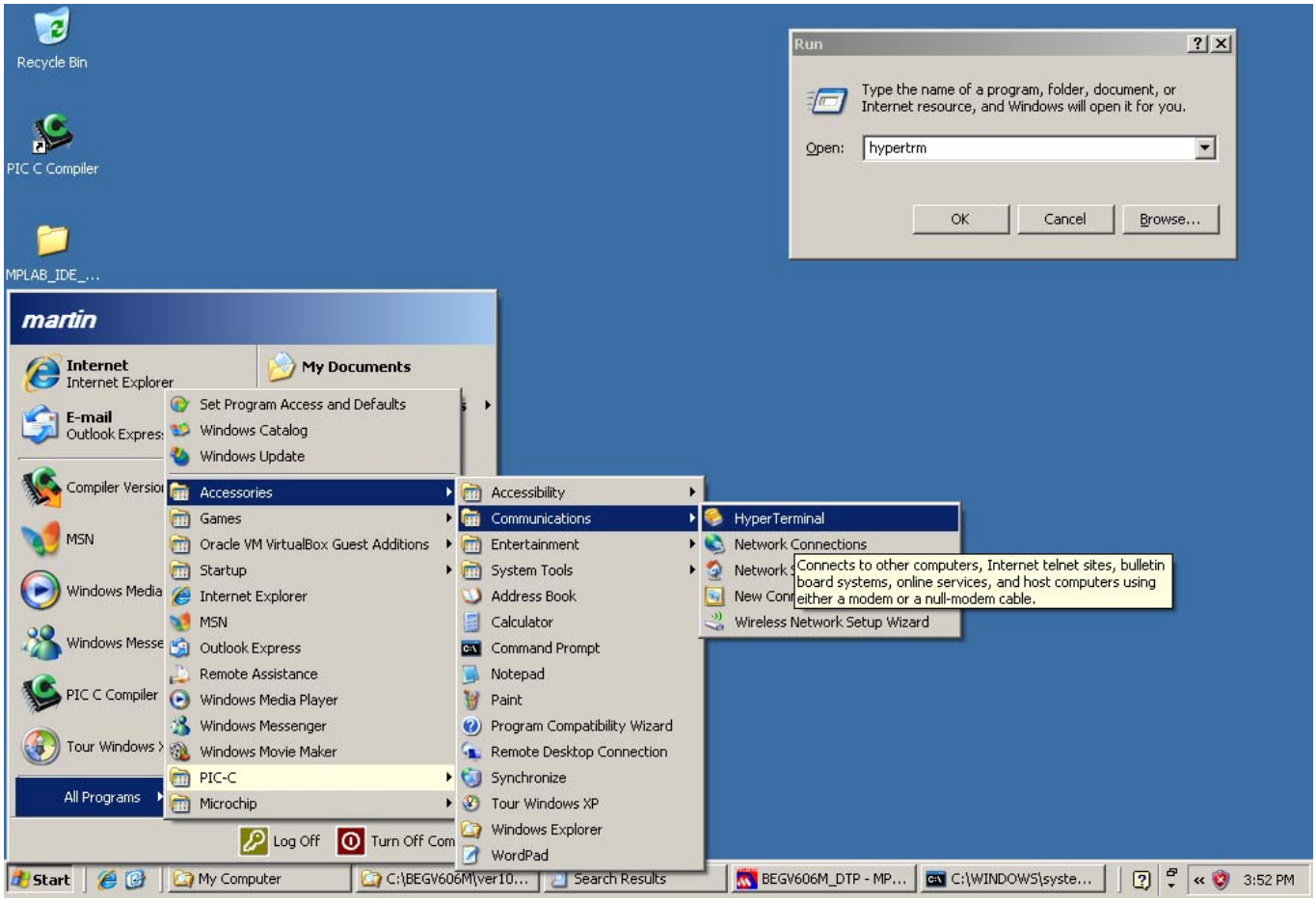
BEGV627M supports touch panel. Bolymin calibrated the touch panel before shipping out. User may try demo program by using touch panel as input device.

Please use hyper terminal for terminal emulation and kernel debugging. The PC keyboard can then emulate as a input device to 627M. Here is the step guide -

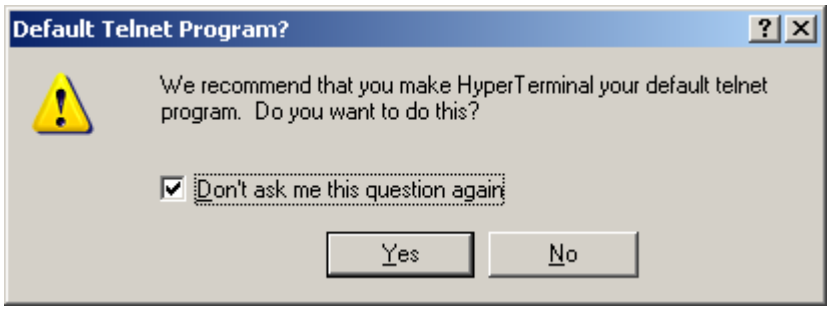
Step 1 - Make sure you have at least one **RS-232 serial port** available. The following example shows a **COM3** is available.



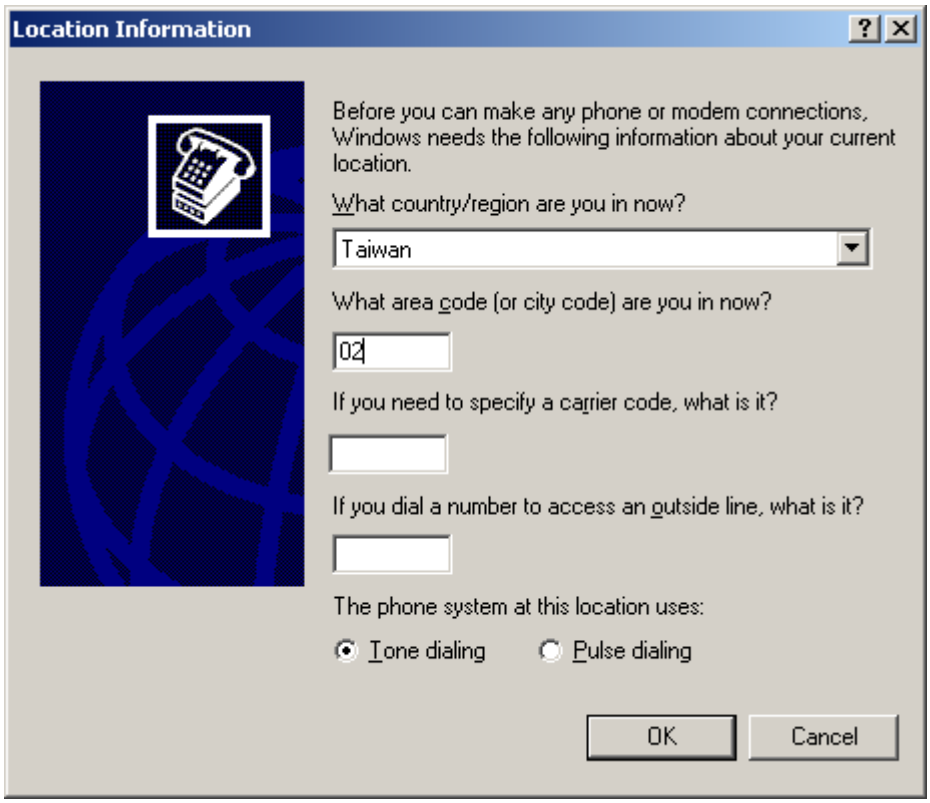
Step 2 - On XP PC: **Start→All programs→Telecommunication→HyperTerminal (or Windows+R, hypertrm)**



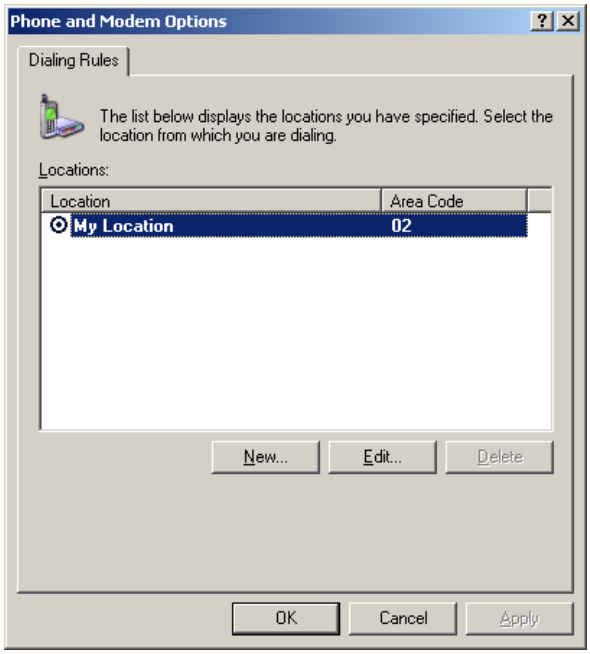
Step 3 - Click **No** if you do not use hyper terminal to telnet to UNIX host.



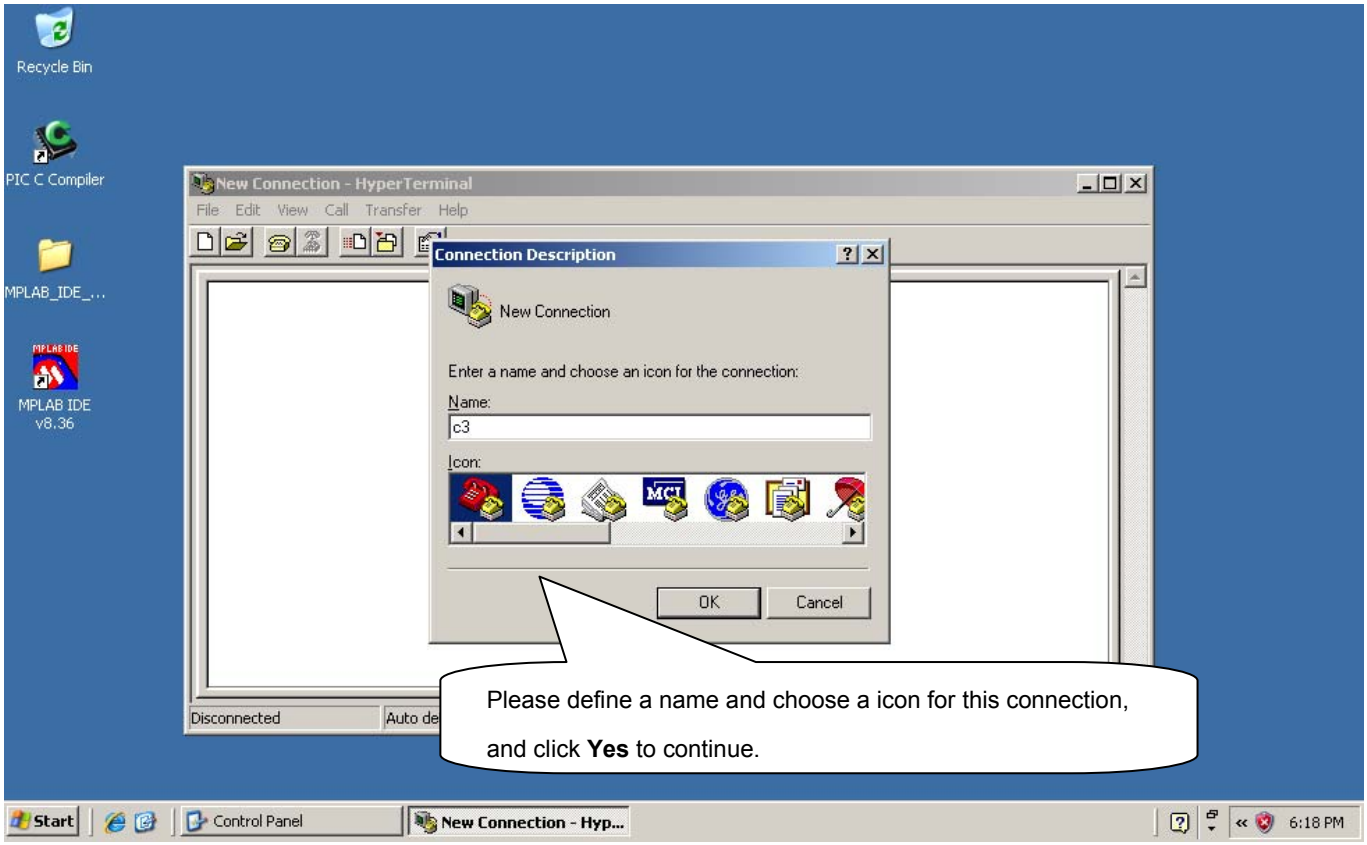
Step 4 – enter **country** and **area code** as appropriate.



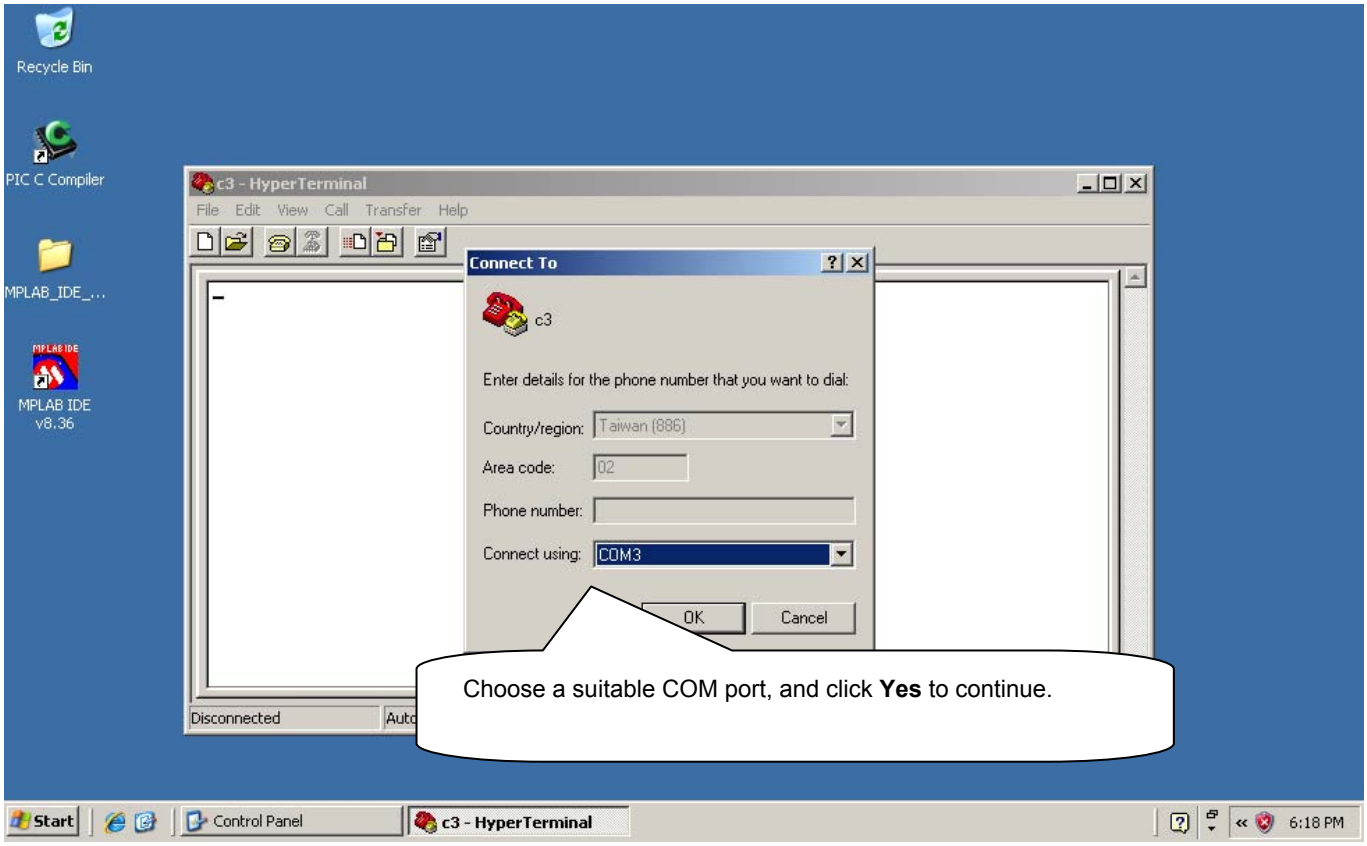
Step 5 – Click **OK**.



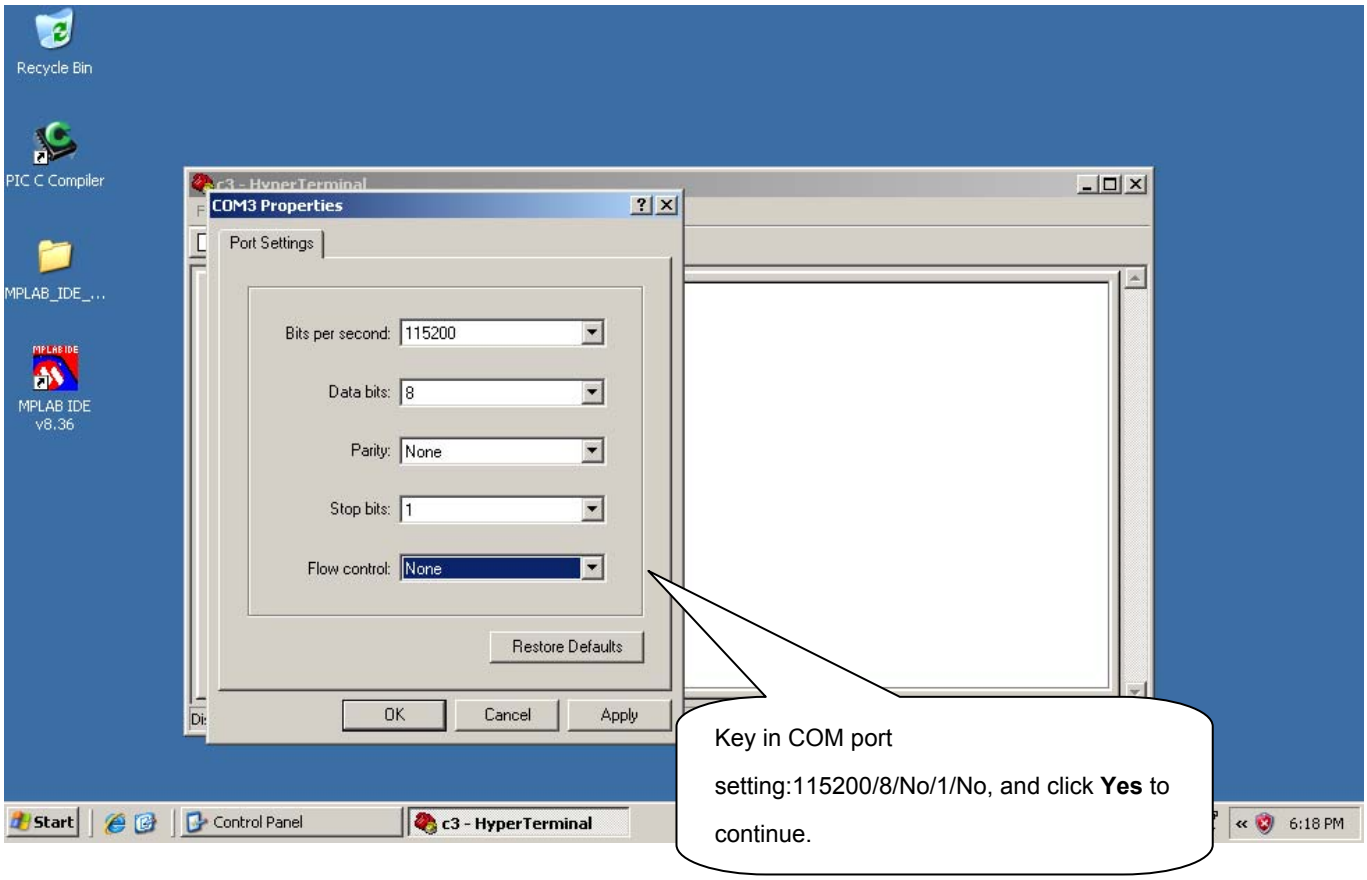
Step 6 - Enter the file name to store the hyper terminal settings. System will auto add a **.ht** extension name.



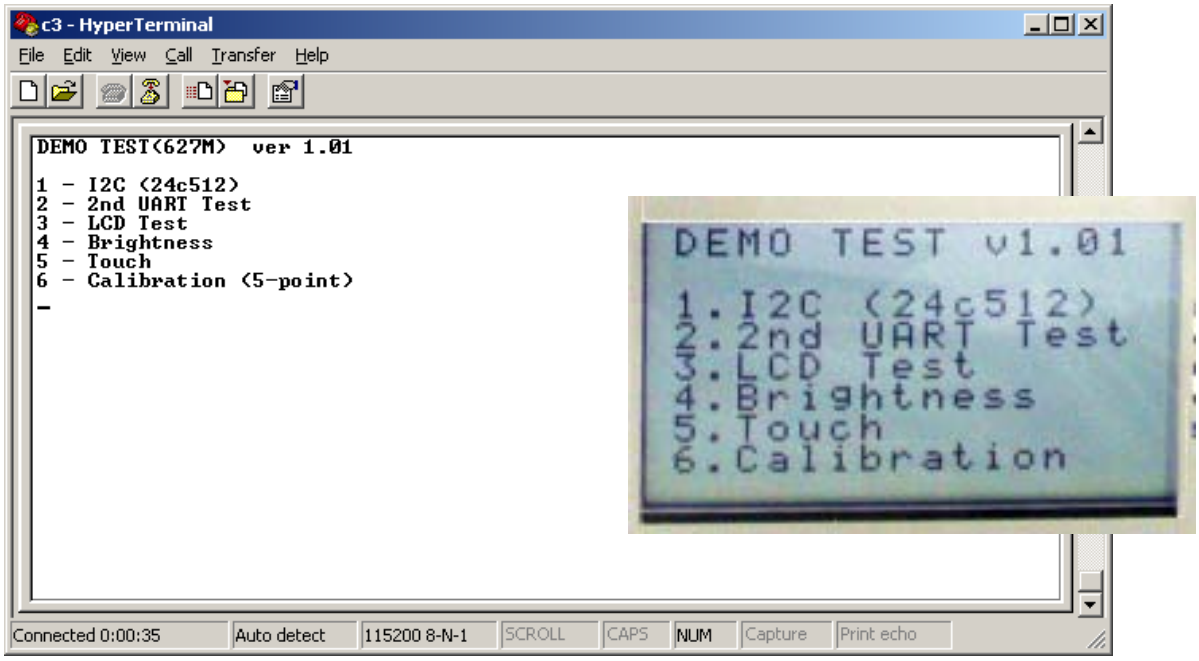
Step 7 -Select **COM** port as appropriate. Hyper terminal will pull down only valid COM ports.



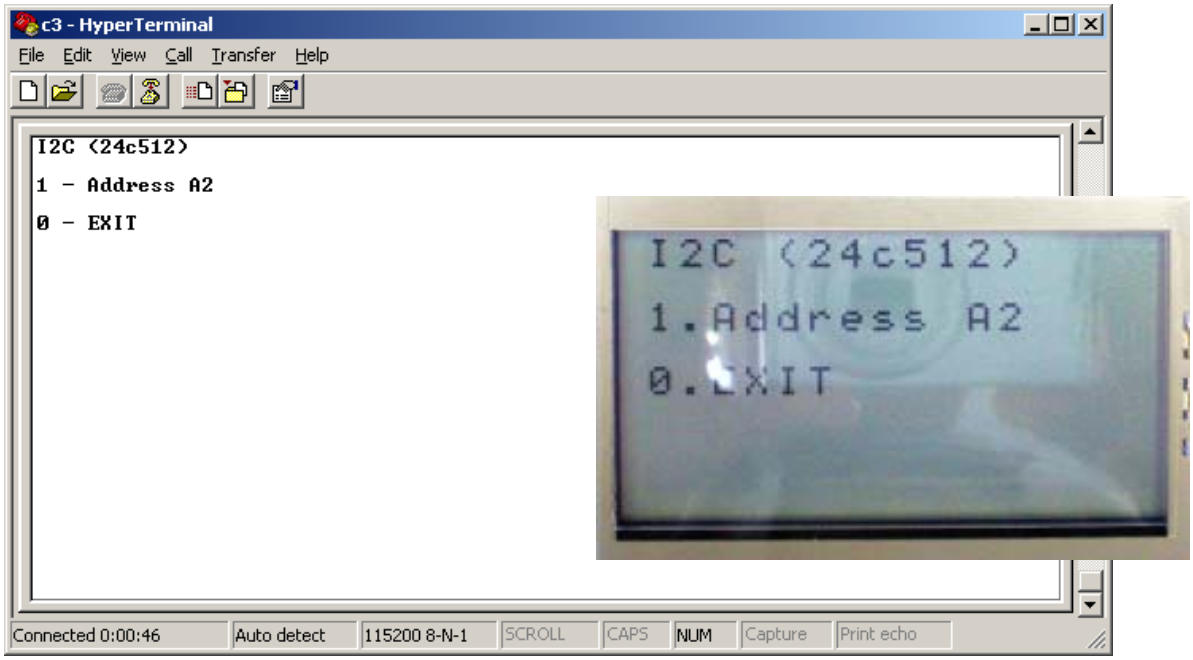
Step 8 - It is required to set the serial communication as follows – 1**15200 bps, N/8/1/None.**



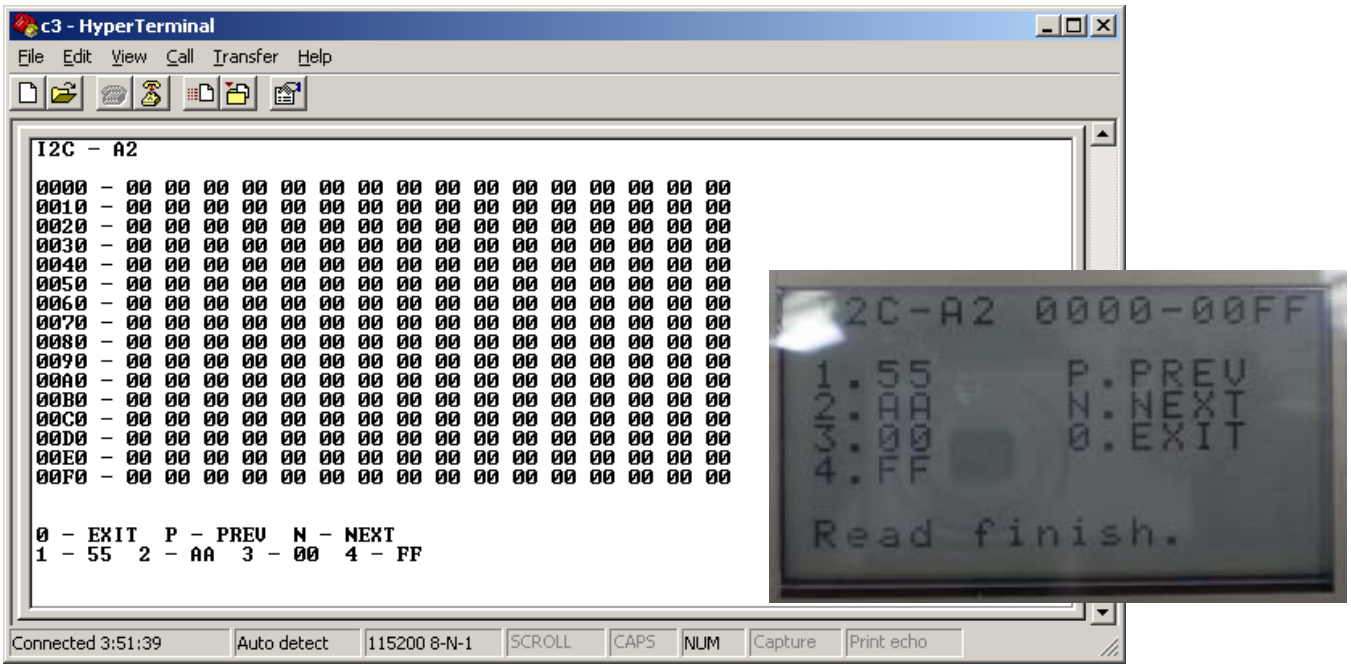
Step 9 – Press **Enter** to bring up top menu. Press **1** to get to i2c (24c512)



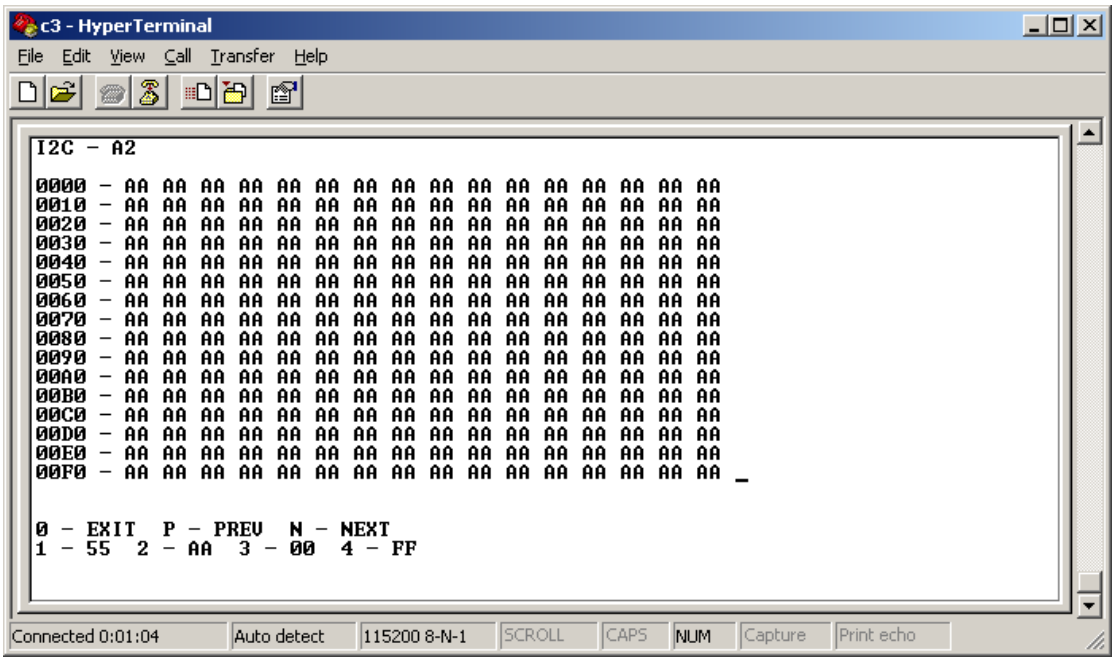
Step 10 - Press **1** to browse through the page content of i2c-eeprom with i2c hardware address of $A2.



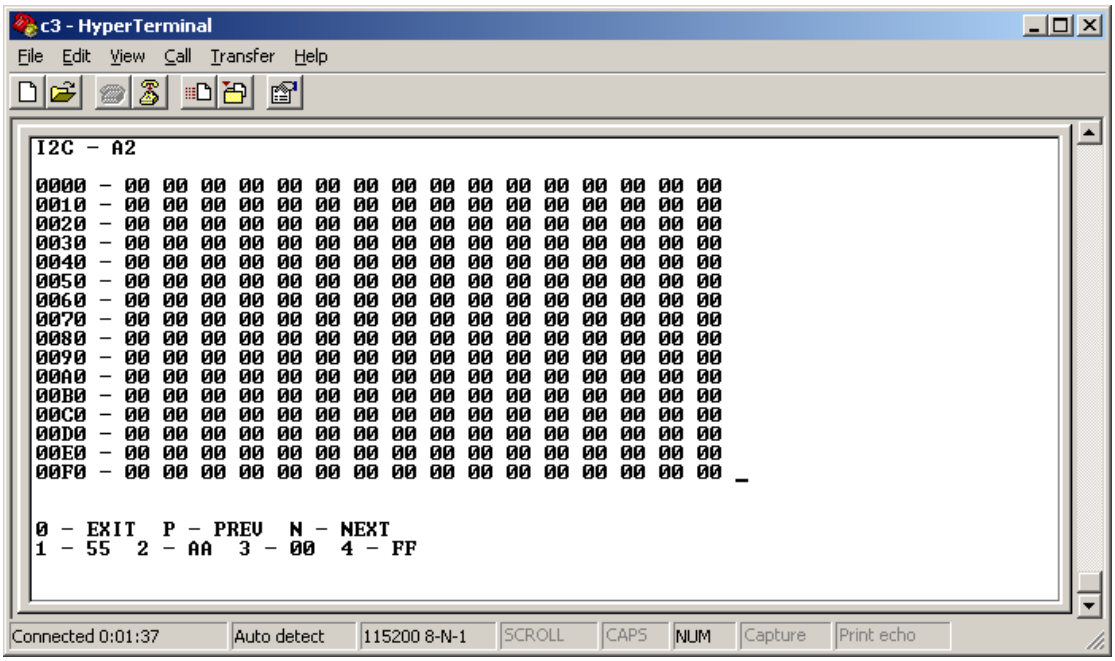
Step 11 - Here is I2c-eeprom content listing starting from 0x0 to 0xff.



Step 12 - Press **2** to write 0xAA into whole page.

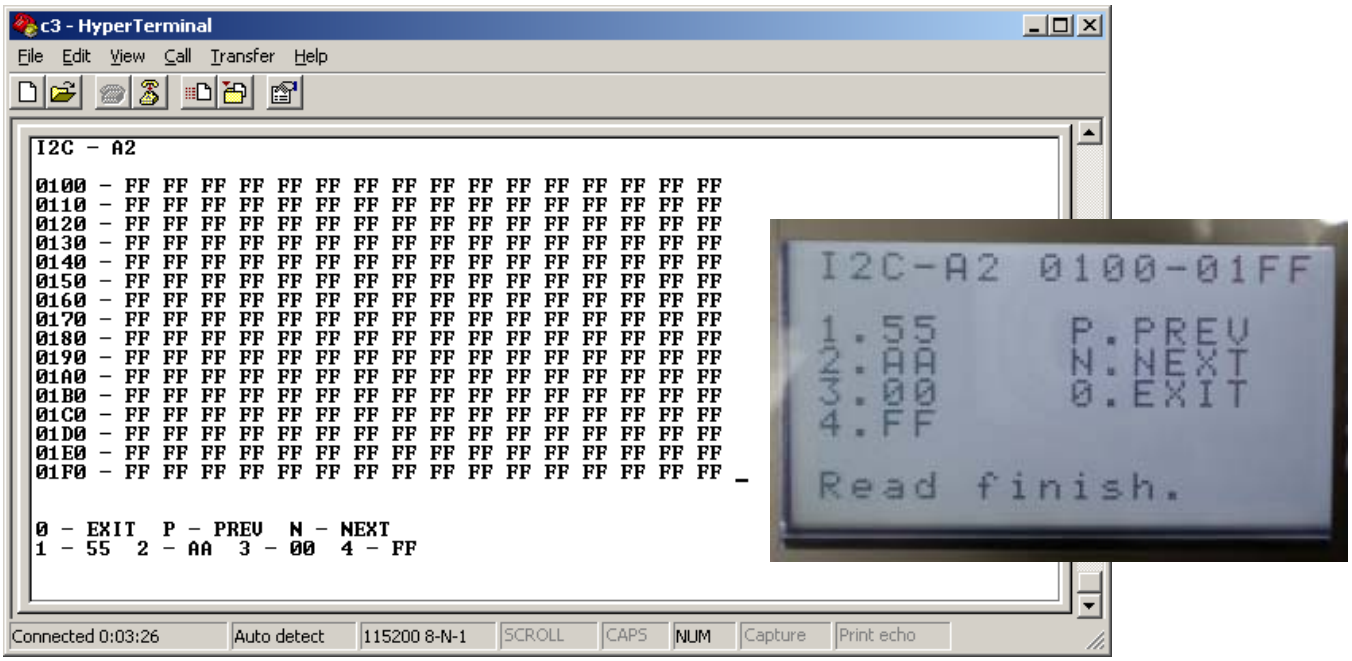


Step 13 -Press **3** – write 0 to the whole page via i2c to eeprom (starting from $0 to $00ff)

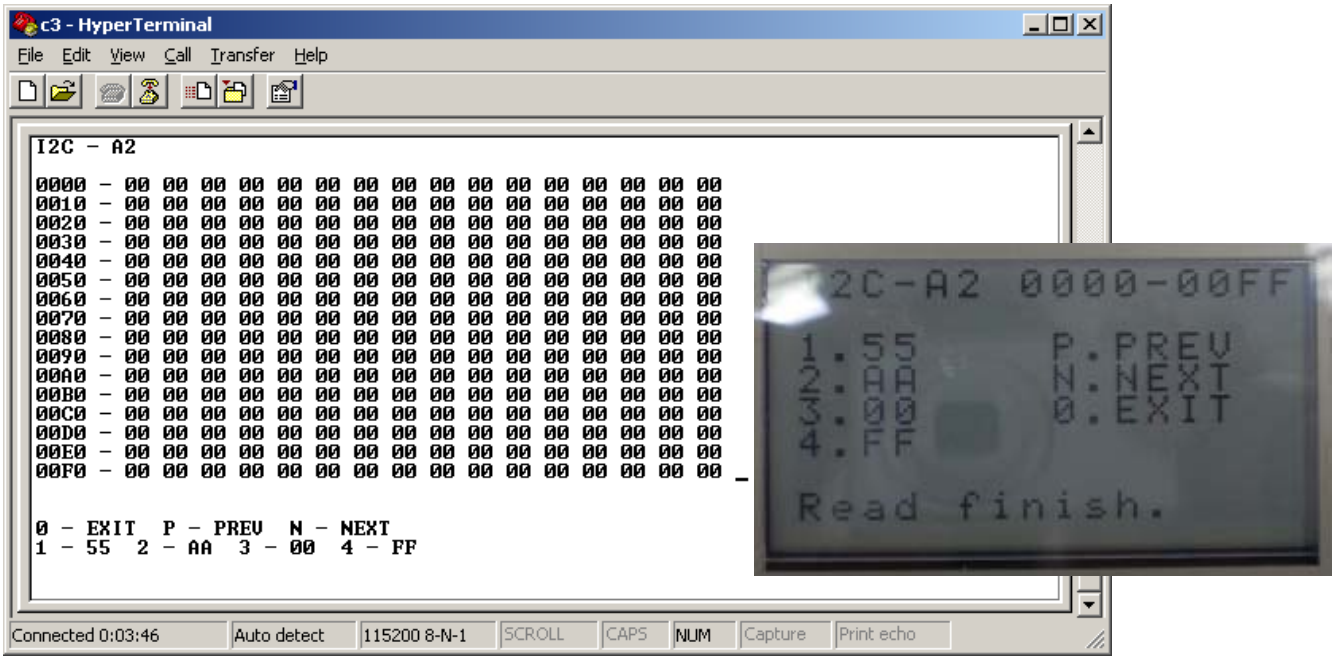


Press **N** for next page, **P** for previous page to confirm the change.

Step 14 - Press **N** to browse content of next page, which is with a starting address of 0x100 .

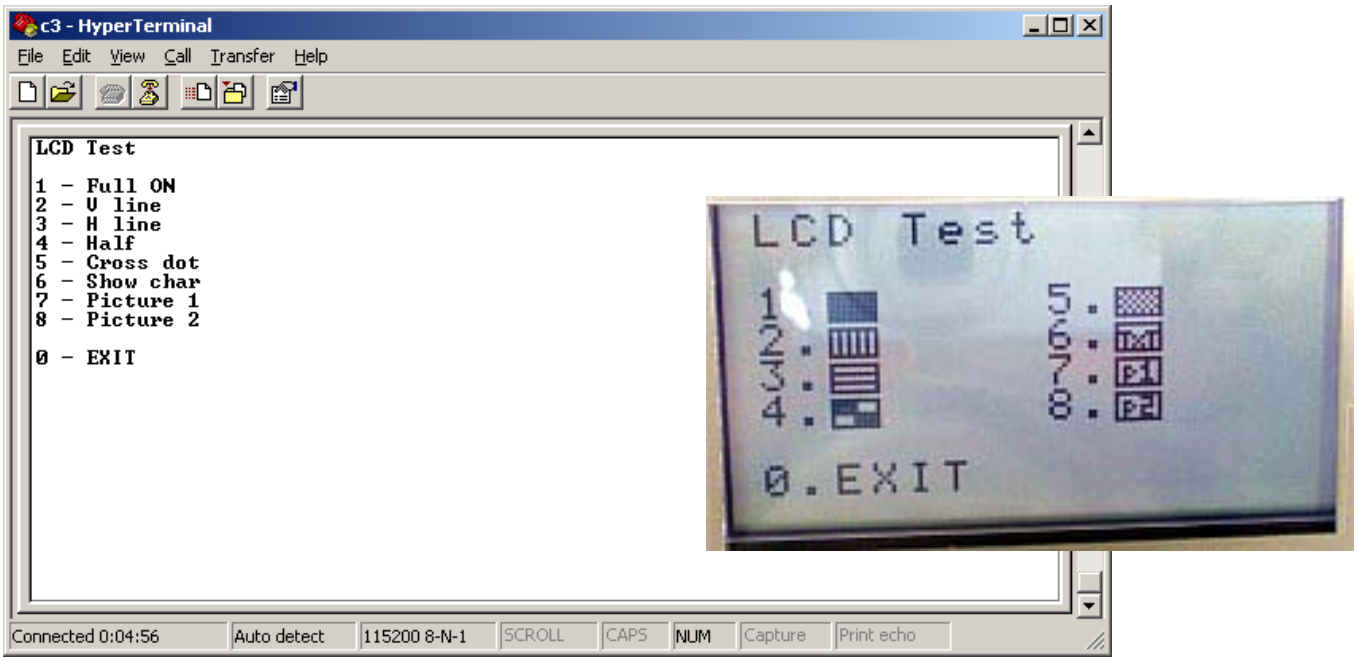


Step 15 -Press **P** to browse content of previous page, which is with a starting address of 0x00. Confirm that previous write operation of 0x00(data) to 0x00 (address) page is indeed succssful.

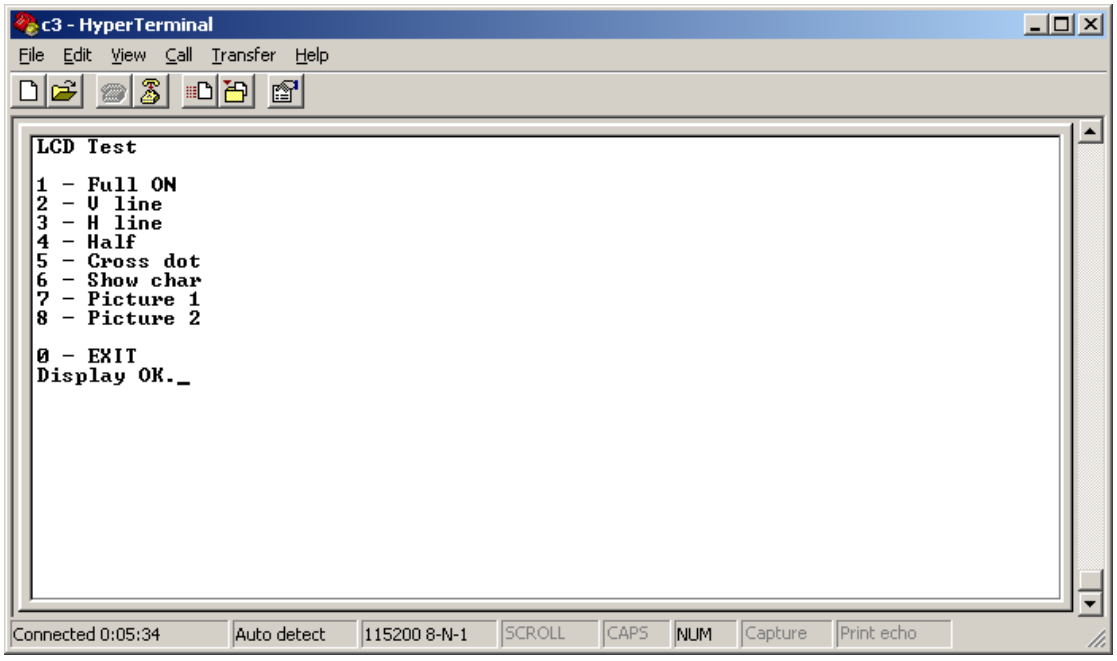


Press **0 0** to return to top menu.

Step 16 -Press **3** to enter LCD test. There are 8 patterns available for the demo.

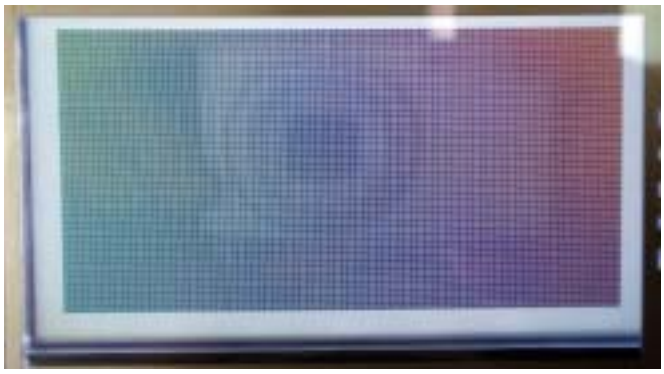


Step 17 - Cycle through **1** to **8** to observe the various demo patterns as shown next page.



Now press 0 to return to top menu.

1. Full on



2. Vertical line

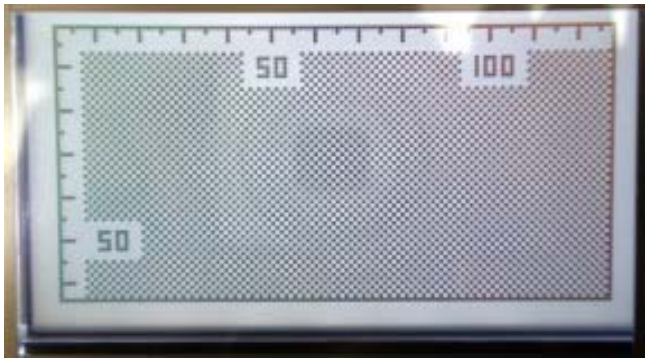
3. Horizontal line

4.Half blocks

5. Cross dots (2x2 checker board pattern)

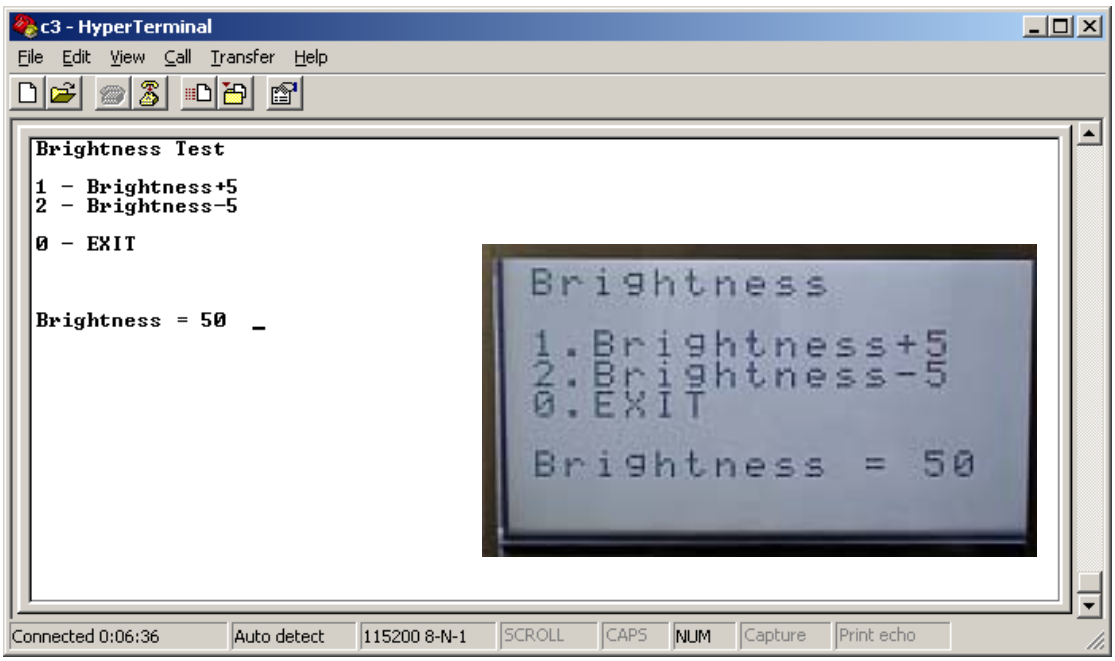
6. Show characters

7. Picture pattern #1

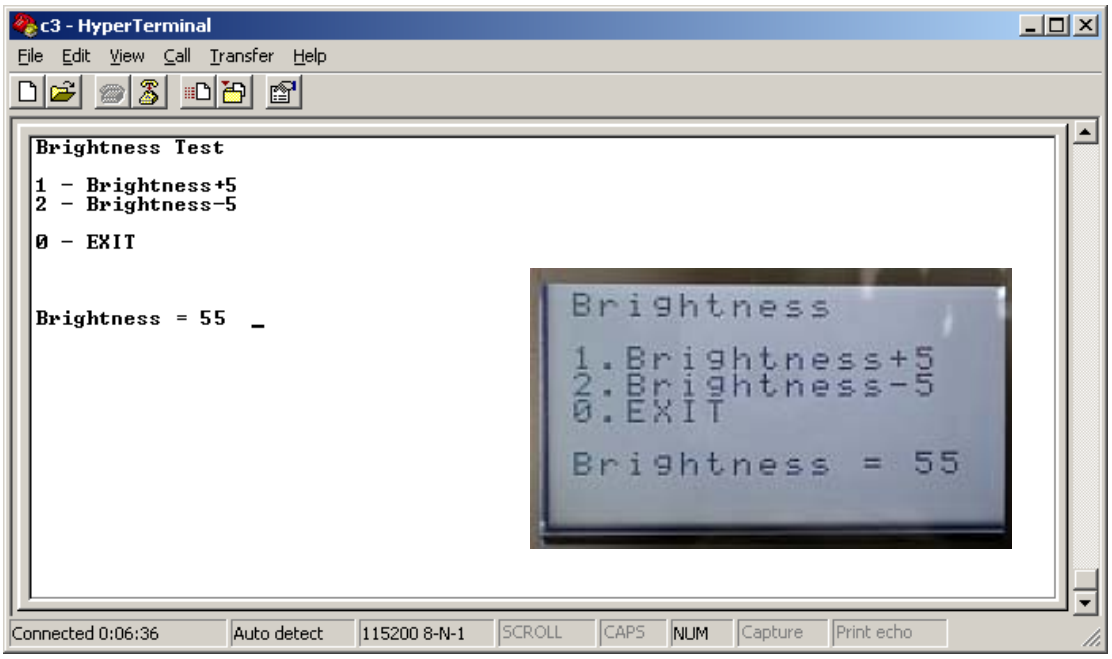


8. Picture Pattern #2

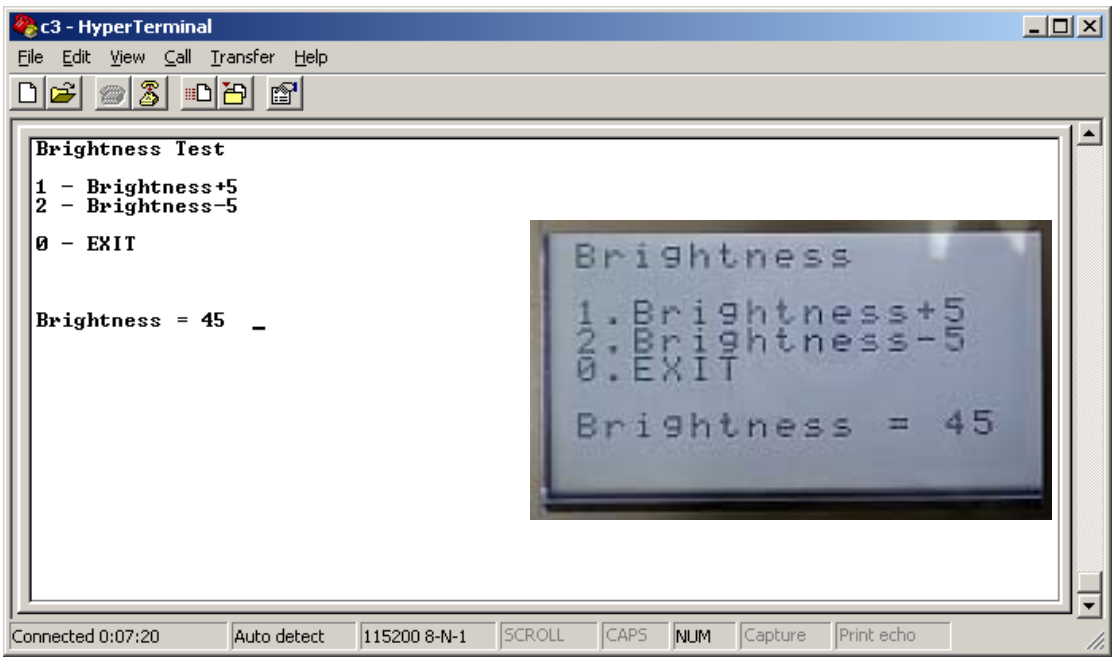
Step 18 - Press **4** to enter brightness test.



Step 19 -Press **1** to increment brightness level by 5. The brightness ranges from 0 to 100 with a step of 5 for each adjustment.

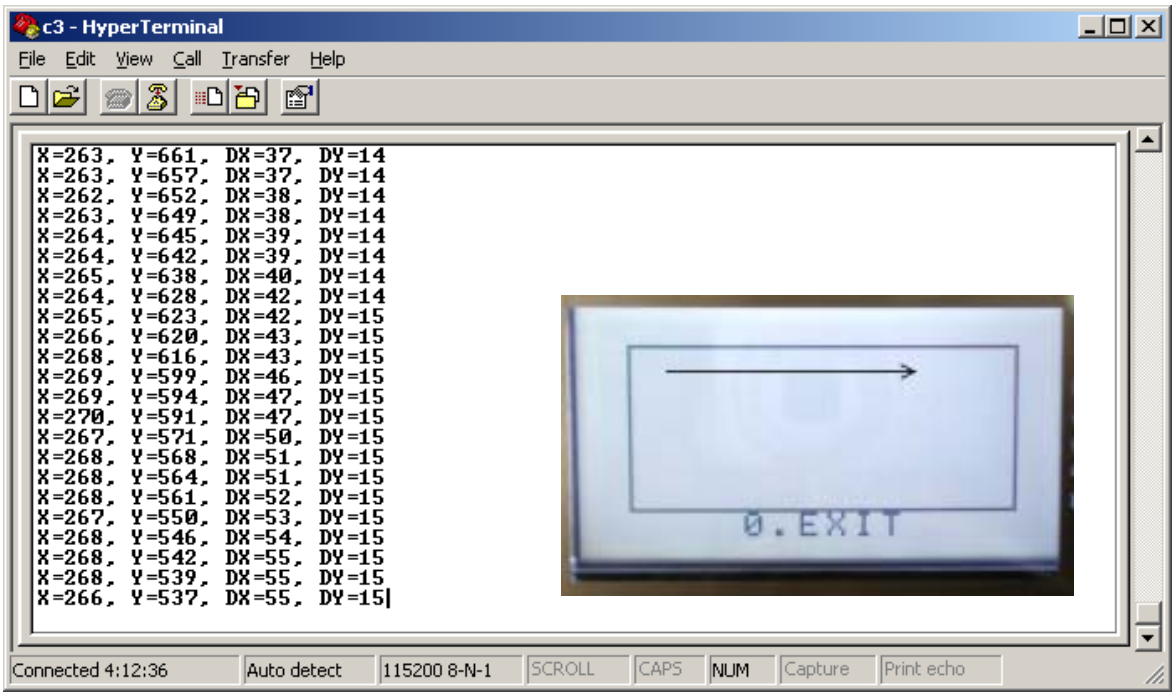


Step 20 -Now try press **2 2** to reduce brightness level by 10.

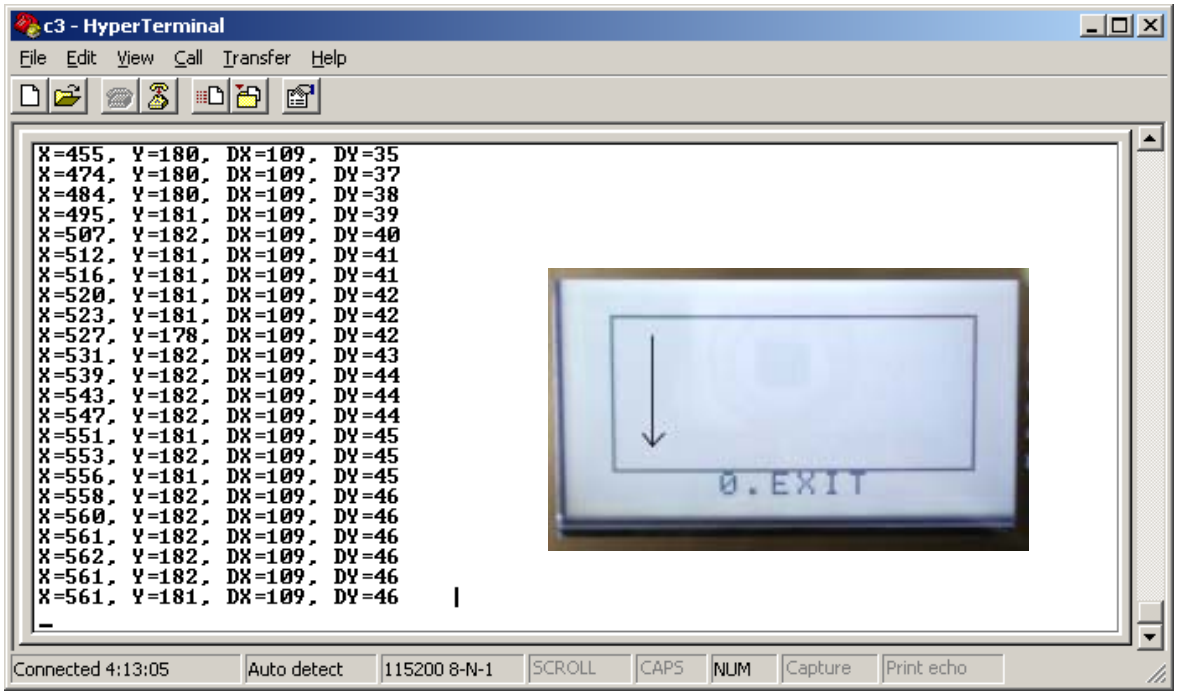


0 to return to top menu, **5** to enter Touch

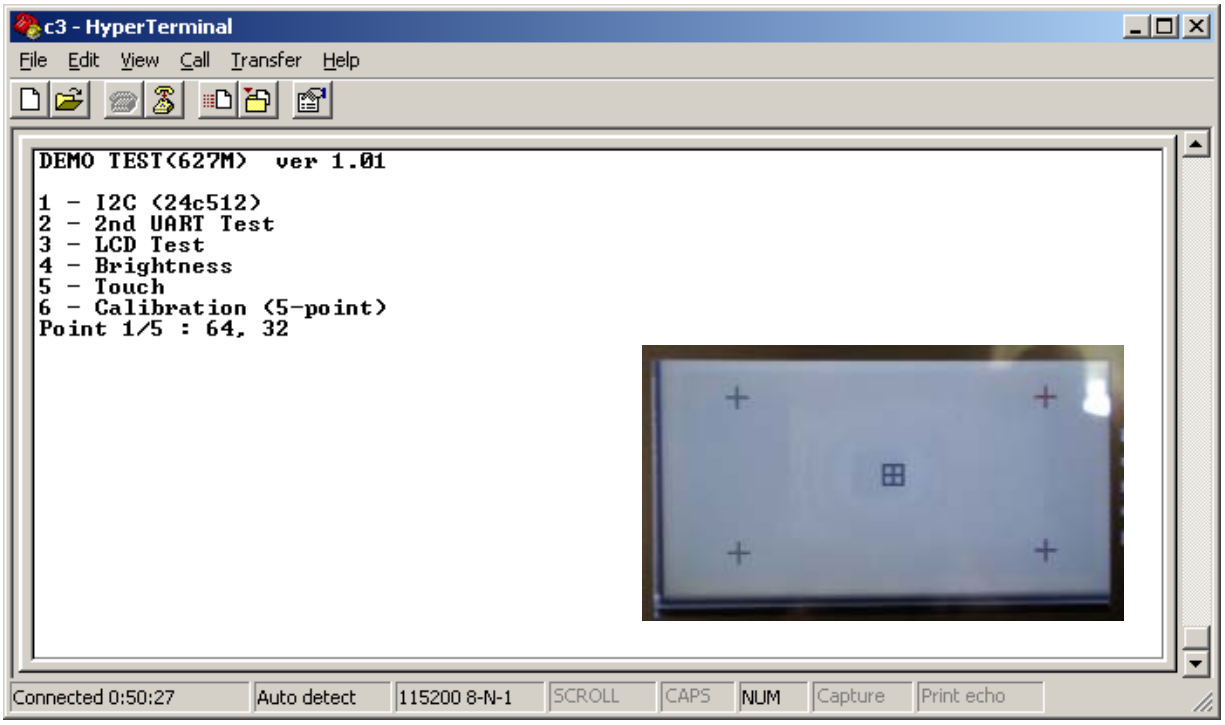
Step 21 -Now use light pen to touch from left to right screen, you will observe the **DX** value increases as the click goes east-bound.



Step 22 -Use light pen to touch from top to bottom, note the **DY** value increase as the click goes south-bound.

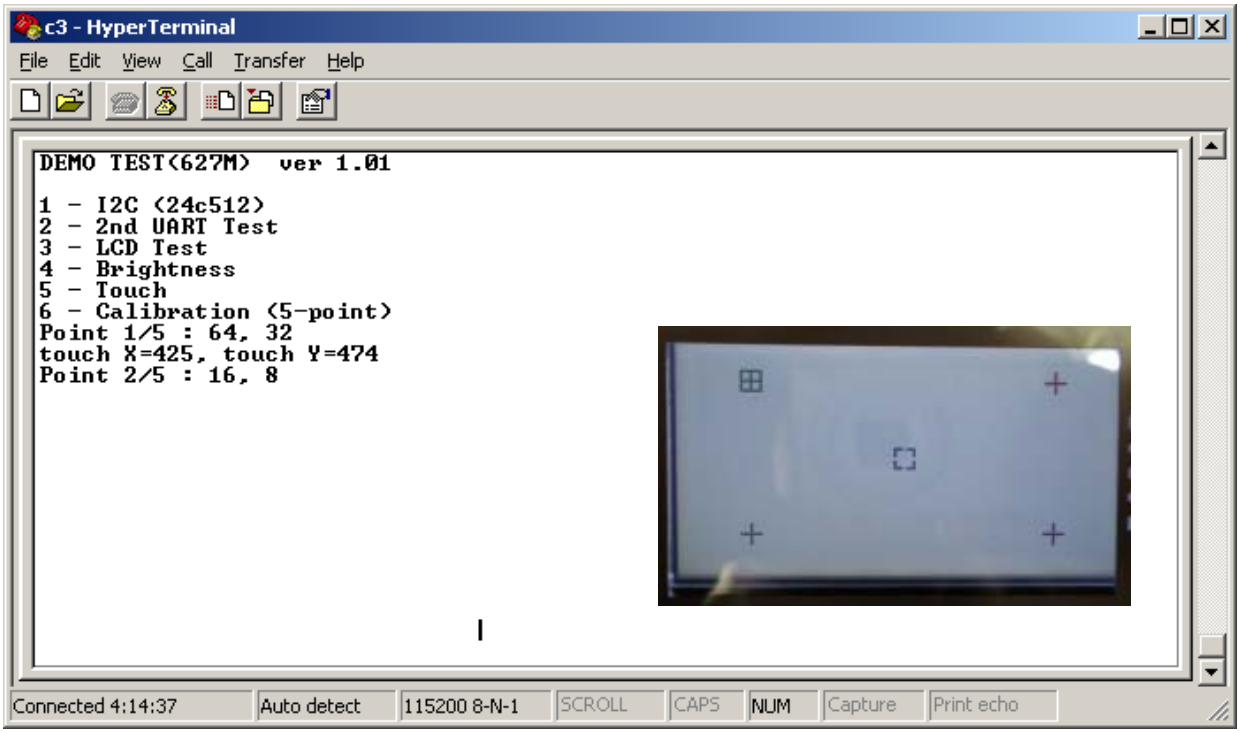


Step 23 - **0** to return to top menu, **6** to calibrate (5-point)

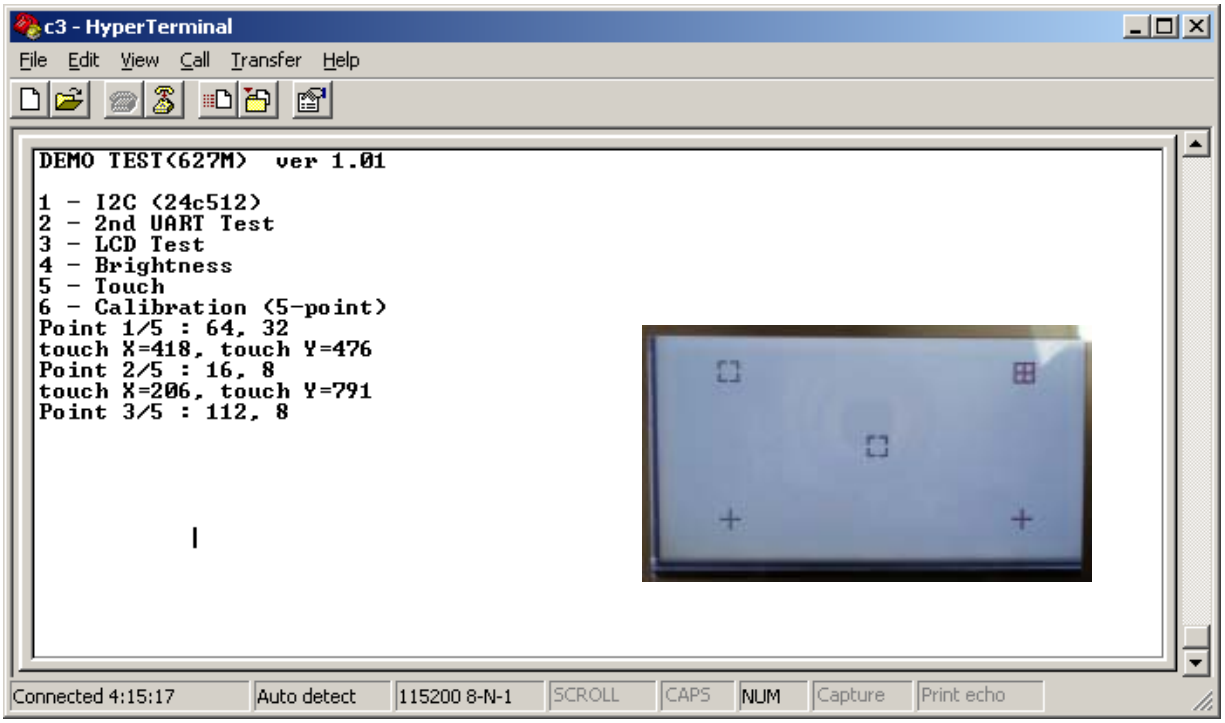


Press and hold **box with cross** for around 2 seconds by the order to center, upper-left, upper-right, lower-right, lower-left. After this calibration, demo program auto return to top menu.

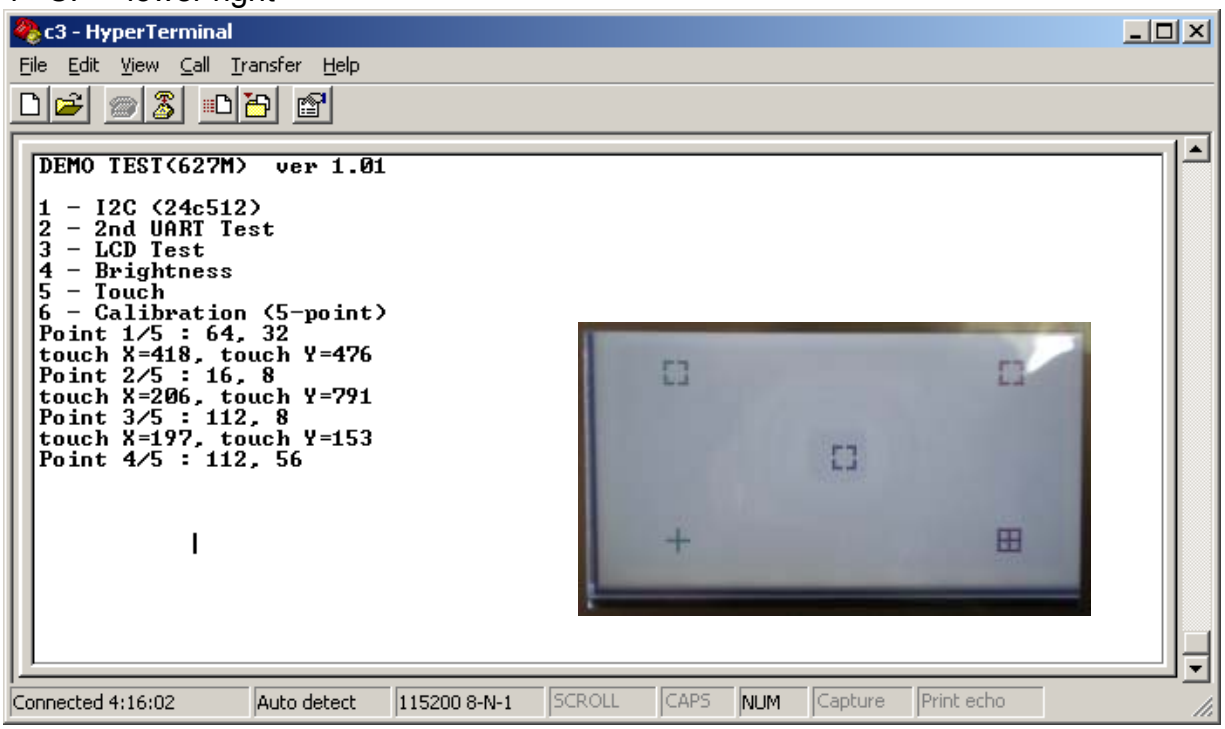
2nd CP – upper left.



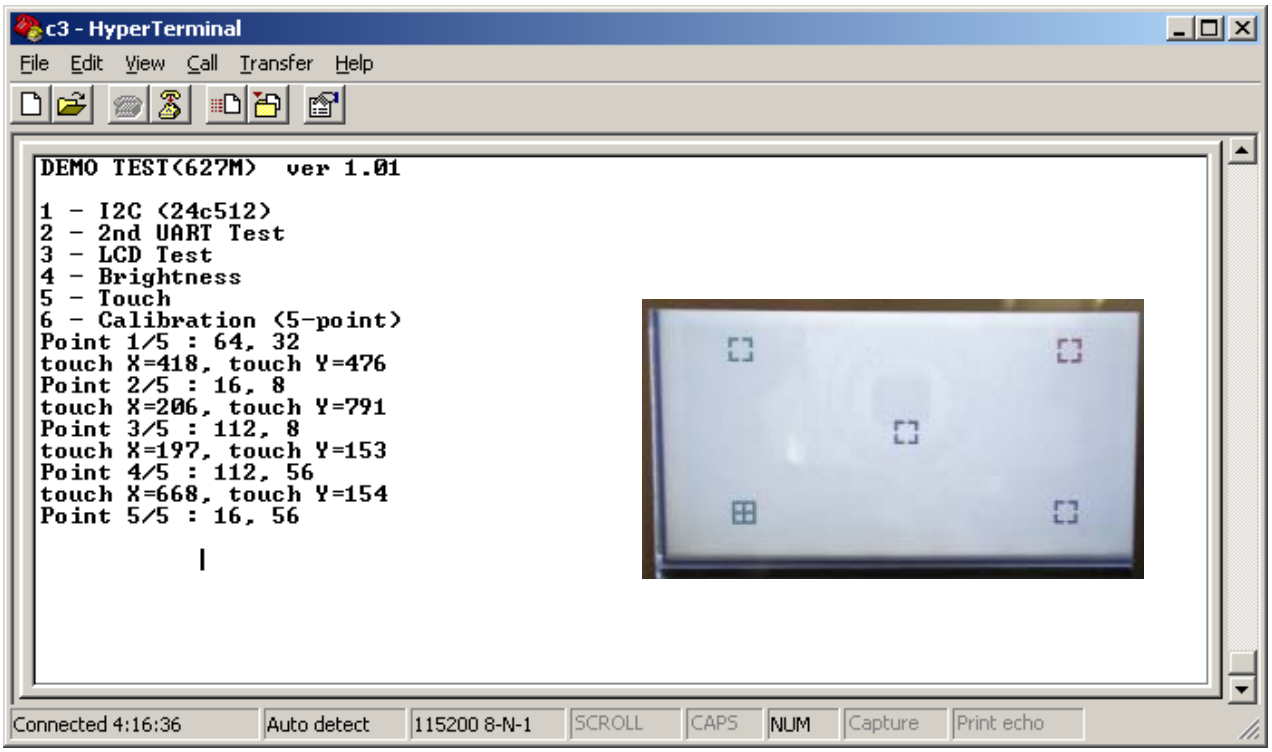
3rd CP – upper right.



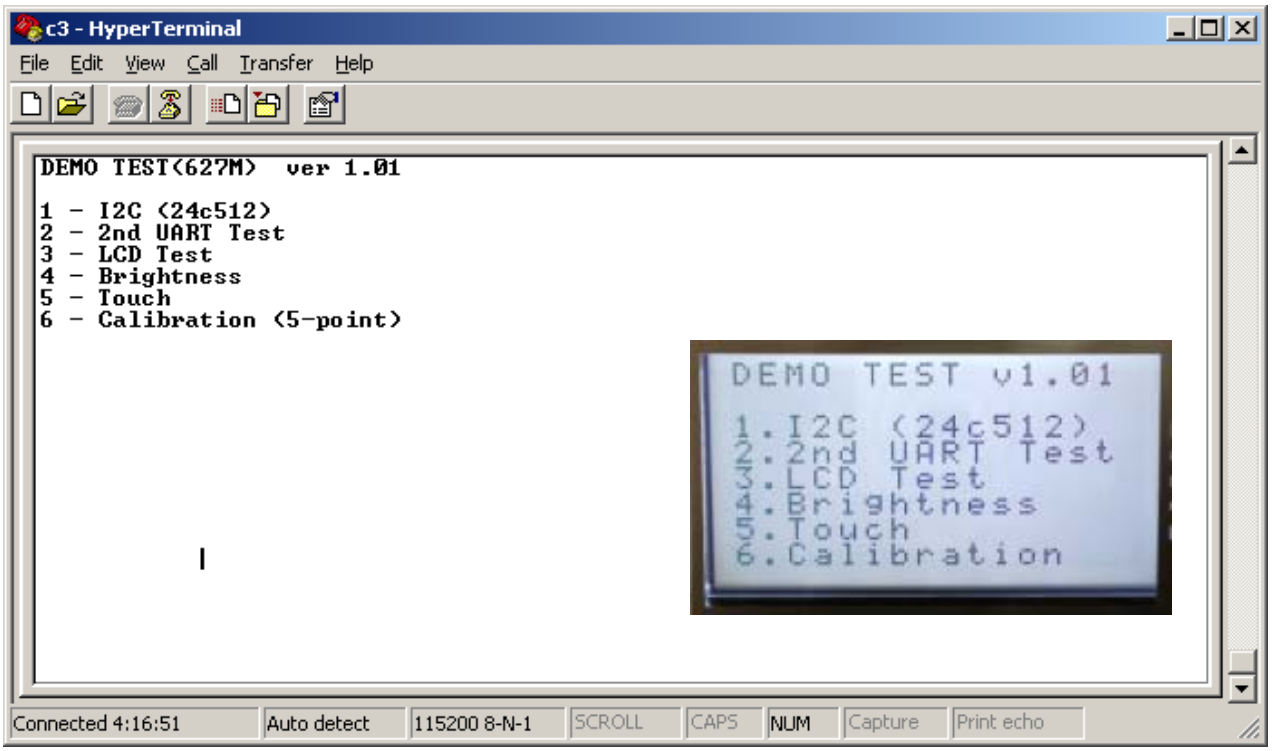
4th CP – lower right



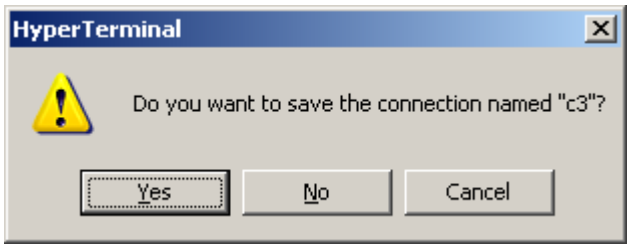
5th CP –lower left



Automatically return to top menu upon completion of 5 CP calibrations.



Step 24 – Close hyper terminal .



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

Chapter 5 Software Reference Manual

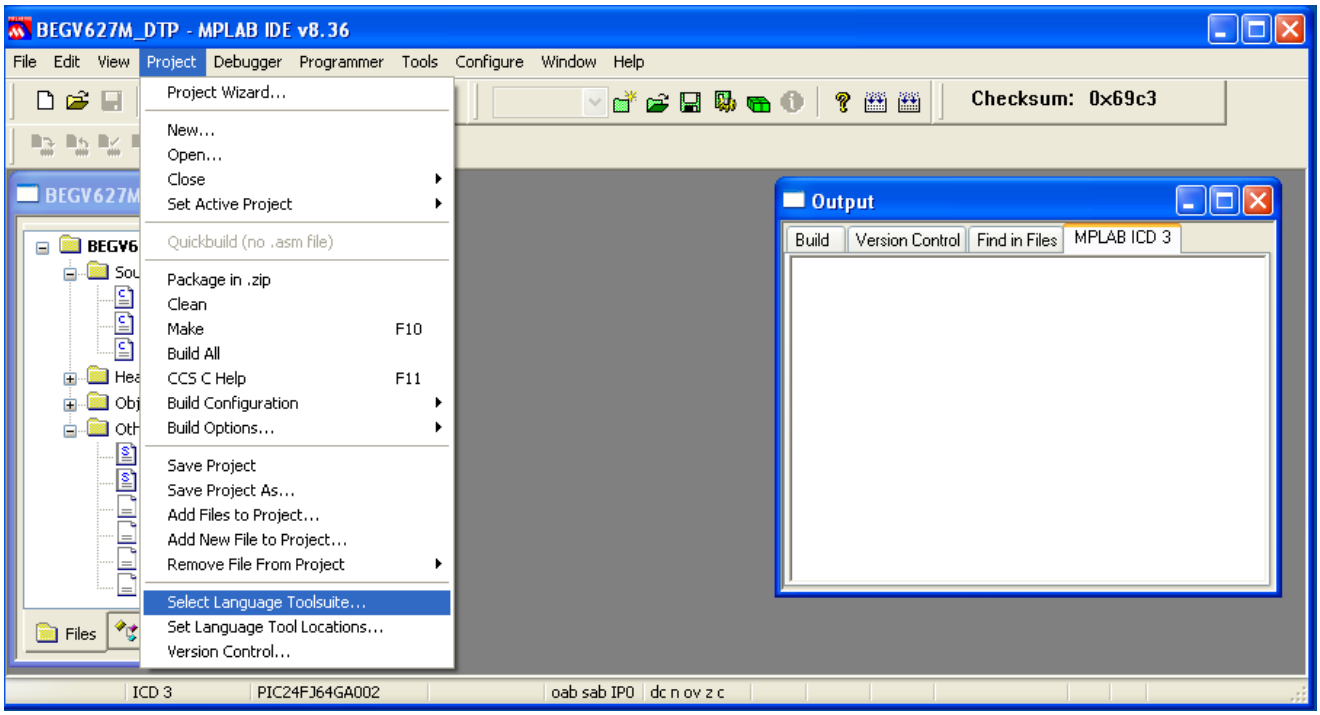
Abstract

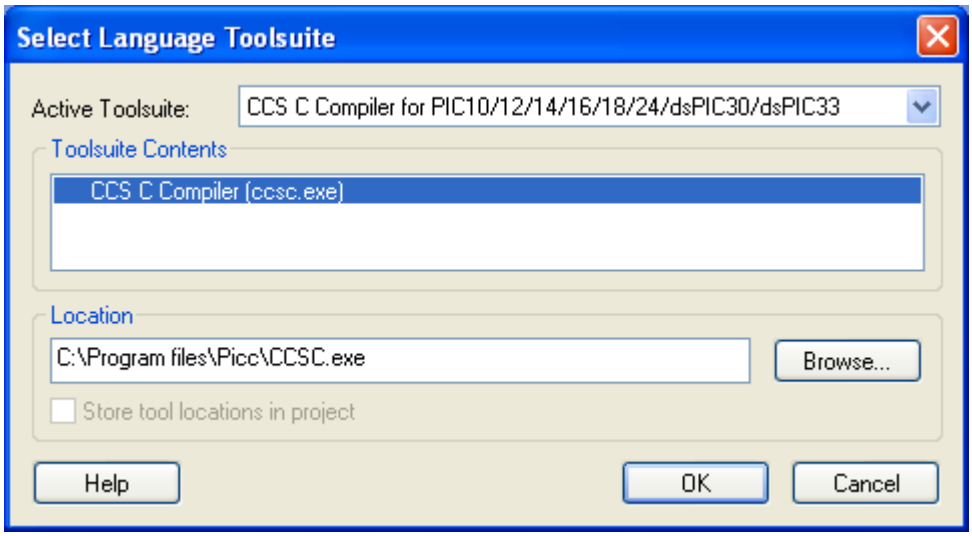
This chapter introduces all functions of device driver library for BEGV627M device and describes major issues about application program development.

5.1 Setup development environment

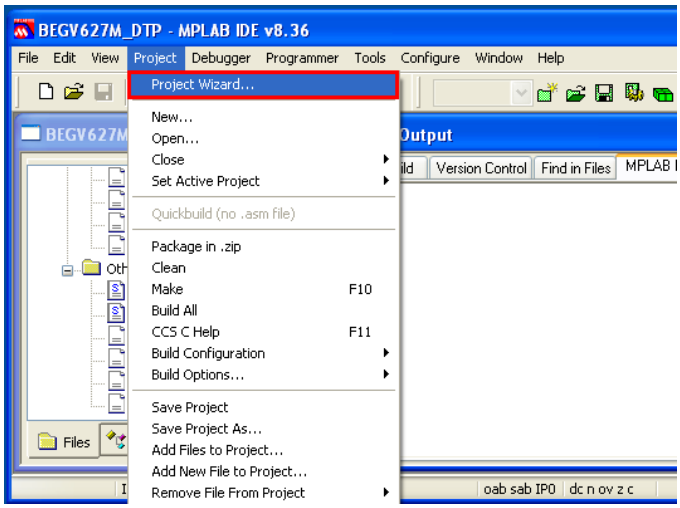
Bolymin recommends to use the Integrated Development Environment (IDE) provided by Microchip, MPLAB IDE v8.36, and CCS C complier v4.093 to develop new application software. Here is the step guide to set up the development environment:

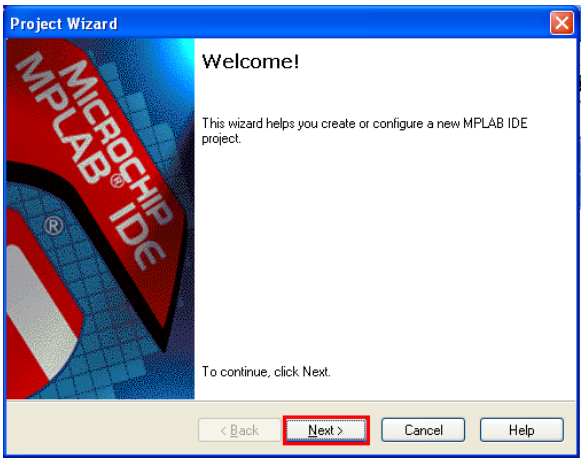
Step 1 - Install **CCS C complier v4.093** – refer to 4-1
Step 2 - Install **MPLAB IDE v8.36** – refer to 4-2
Step 3 - Open **MPLAB 8.36** ,then select **Project-Set Language Tool Locations…** to confirm default complier is **CCS C v4.093** complier. User may not need to change the setting if she follows chapter 4.1and 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.

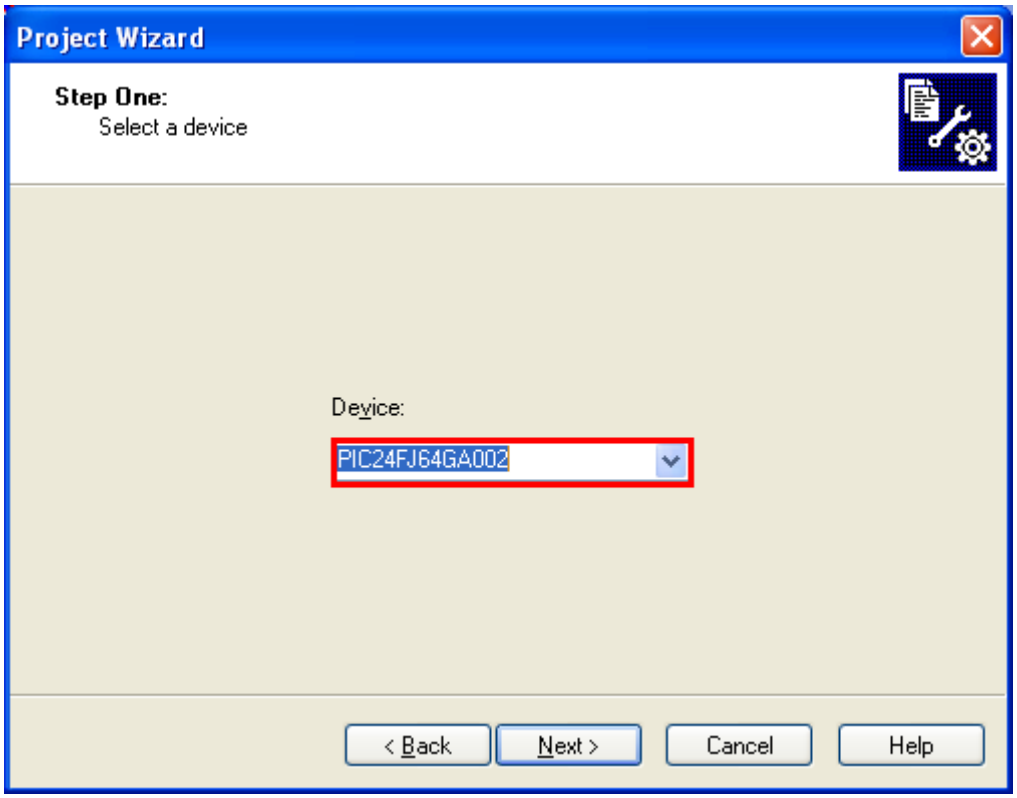




NOTE:

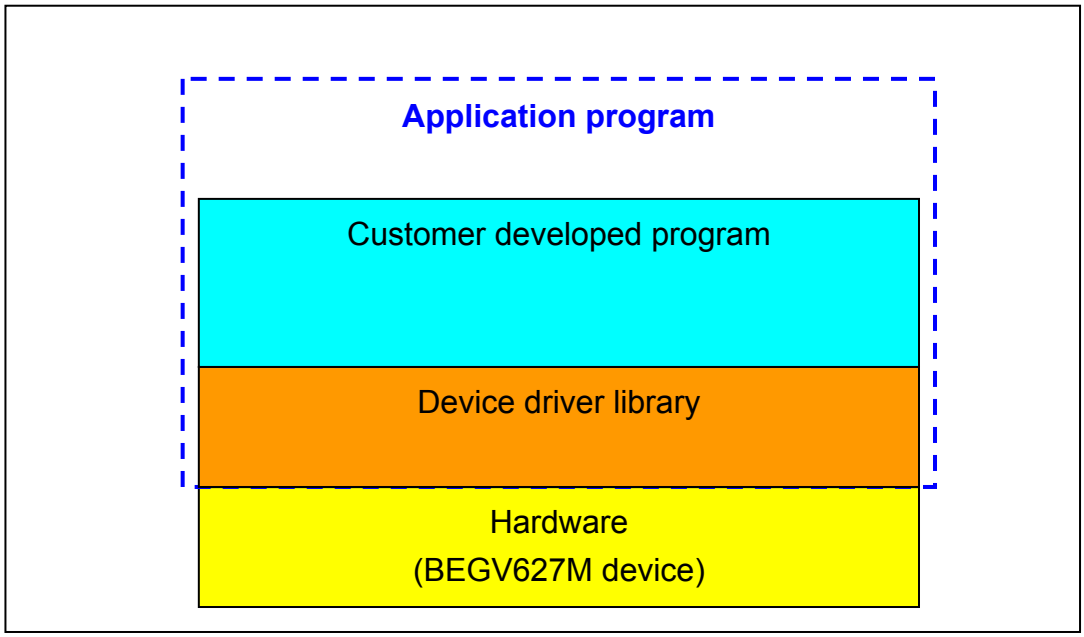
1. After creating a project, user need to create an empty file by the name of project in the directory of project so that the HEX file could be created successfully. For example, Project DTP_627M needs a file named **DTP_627M** in the same directory.
2. Please include **config.h** for all *.C file.
3. We need to select device type of MCU while creating new project at the screen shown as follows:

Step 5 - For **BEGV627M** , select **PIC24FJ64GA002** as device.



5.2 Introduction of library

Bolymin supports device driver library for application development of BEGV627M. Refer to the following figure for program hierarchy.



5.2.1 Summary of modules of library

BEGV627M driver library consists of 5 modules. There are several functions for each module. Following table is a summary of modules in library:

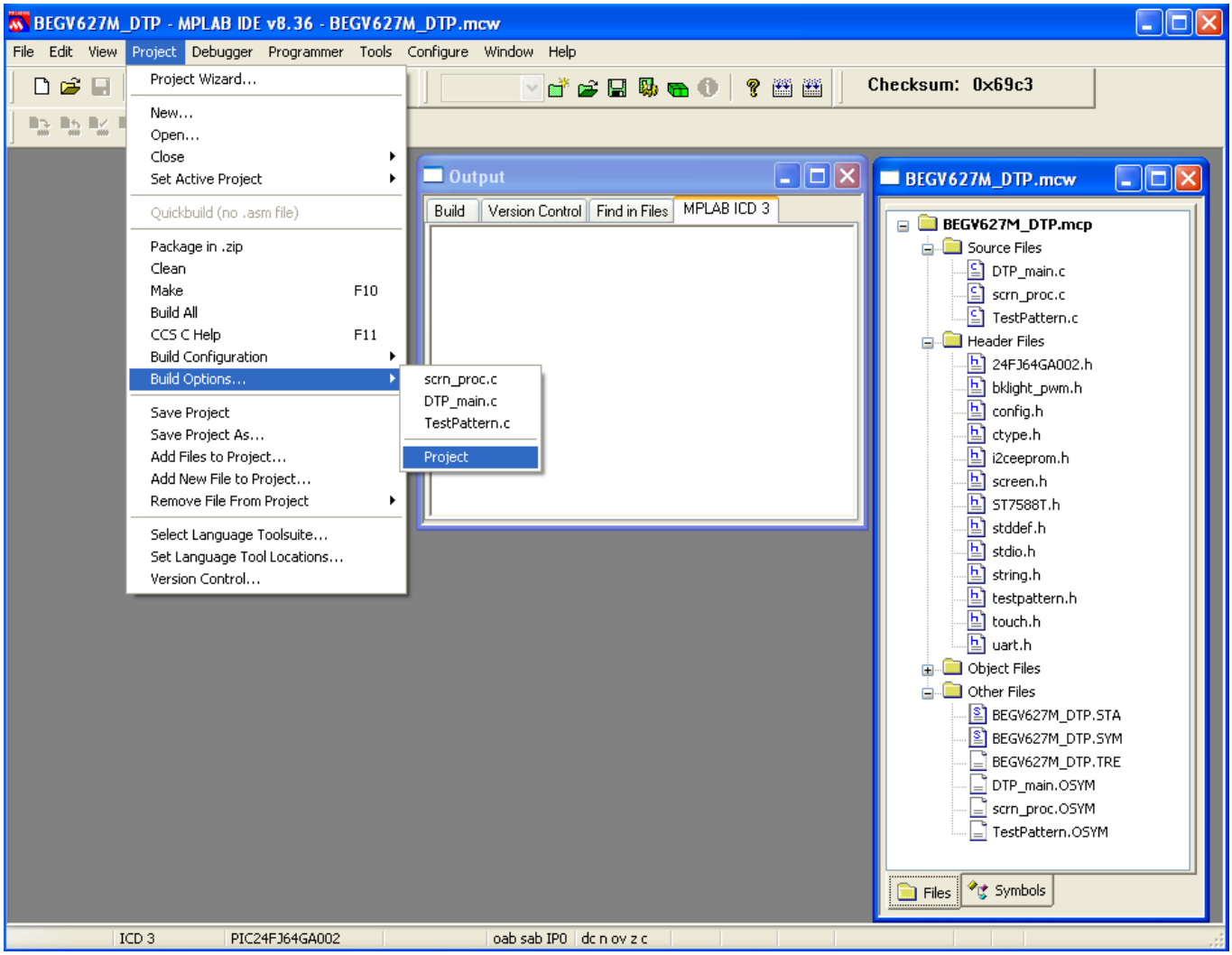
Module name	Description	Header file and compiled object
UART	Include all functions about operation of UART port.	uart.h uart.o
I^2C eeprom	Read and write functions for I^2C eeprom.	I2c.h I2c.o I2ceeprom.h I2ceeprom.o
LCD display	LCD display fuctions	ST7588T.h ST7588T.o
PWM for backlight	Set or get current PWM value to adjust backlight of LCD.	bklight_pwm.o bklight_pwm.h
Touch	Touch panel	touch.h touch.o

5.2.2 How to use library in application program

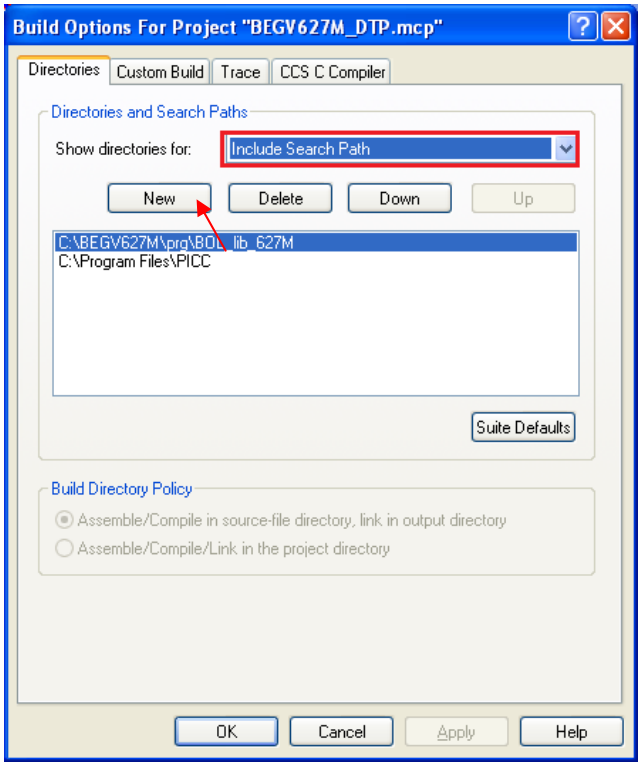
Please follow below steps to use device library provided by Bolymin:

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

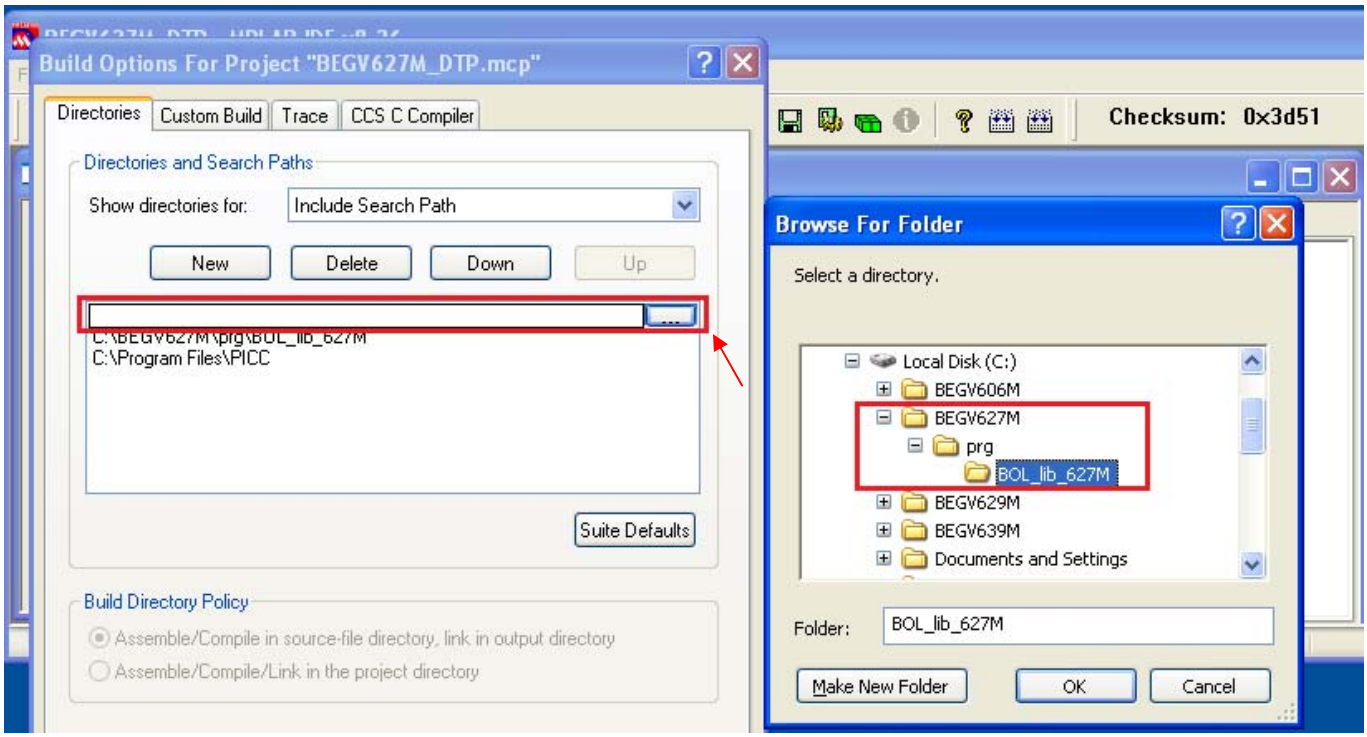
Step 2.- Select **Project-Build Options…-Project**.



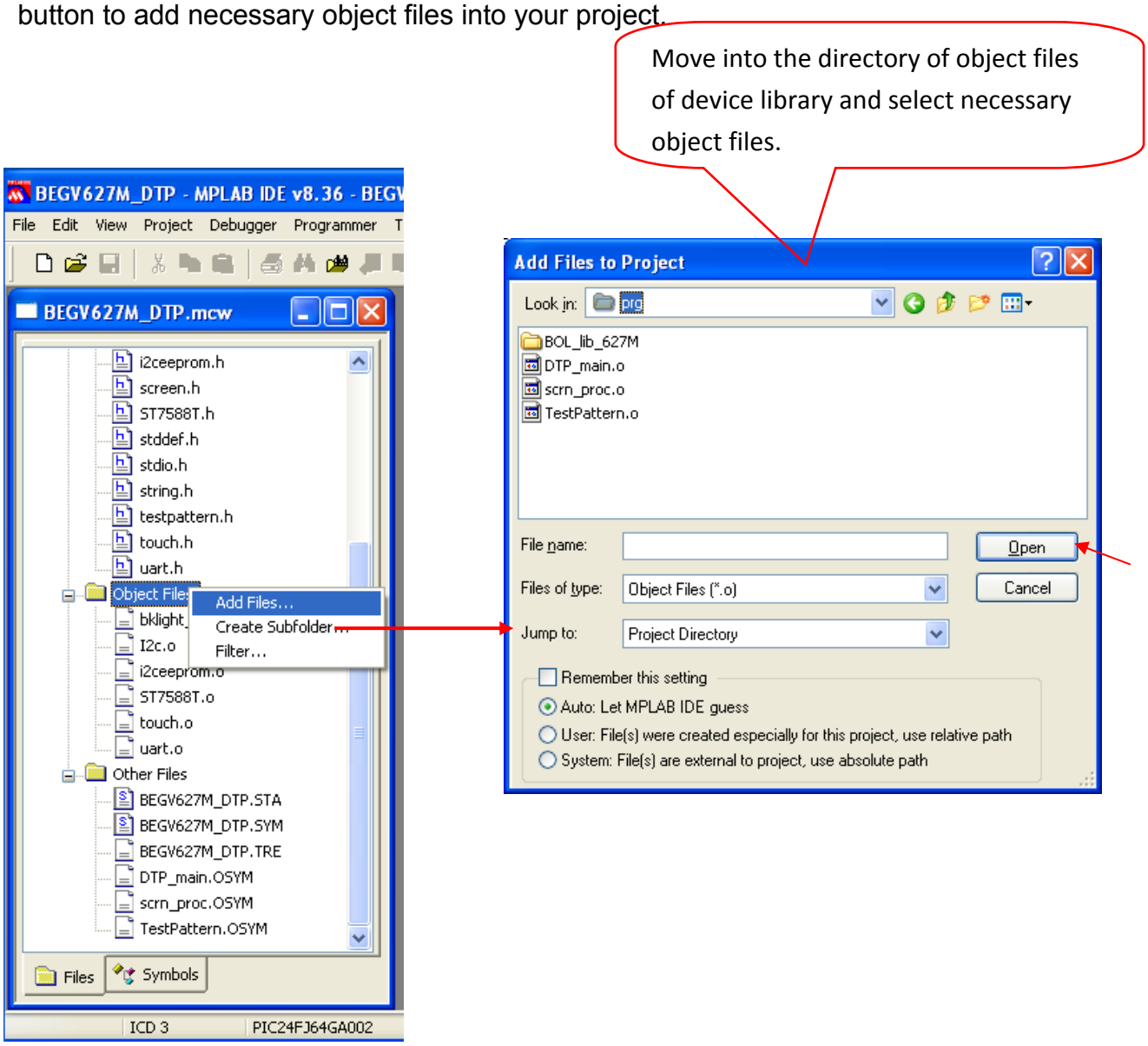
Step 3 – Set **Include Search Path** to the path of header files of library provided by Bolymin.



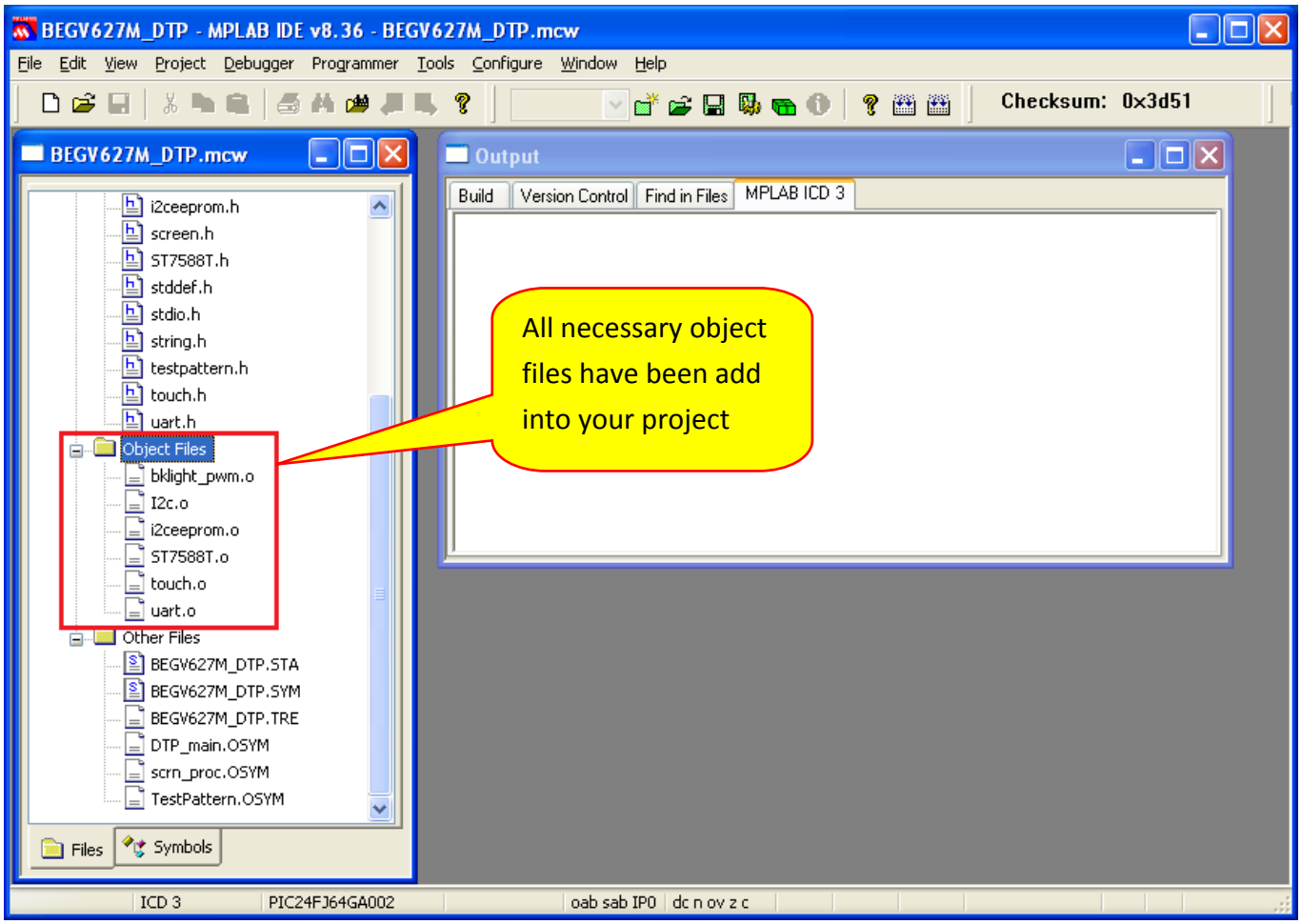
Click on **New** to bring up a new row and a folder browse button. Click on browse button to bring up a **Browse For Folder dialog** windows to add it to **include search path**.



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.



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	void uartInit(uint8_t byPort, uint32_t uBaudrate, uint8_t byParity, uint8_t uDatabit, uint8_t uStopbit, uint8_t nTxMode);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port nBaudrat Baud rate, ex: 9600. byParity Parity Check, 'N' – None, 'E' – EVEN, 'O' - ODD. uDatabit data bit , 5 ~ 8. uStopbit data bit , 1 ~ 2. nTxMode 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	void uartSetBaudRate(uint8_t byPort, uint32_t uBaudrate);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port nBaudrate Baud Rate, ex: 9600.
Return value	None.

uartSendByte Function: Send 1 byte from assigned UART port

Syntax	void uartSendByte(uint8_t byPort, uint8_t txData);
Parameters	byPort UART_PORT0 – 1st Uart port UART_PORT1 – 2nd Uart port txData 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 I²C device address. (address of eeprom on board are: $A2_{hex}$, $A4_{hex}$. nAddr Address will be read.
Return value	Data reading from I²C eeprom.

i2cWriteByte Function: Write 1 byte data to I²C eeprom.

Syntax	void i2cWriteByte(uint8_t uDevAddr, int16 nAddr, uint8_t byData);
Parameters	uDevAddr I²C 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 I²C 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	BEGV627M
Header file	ST7588T.h
Object file	ST7588T.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.

lcdDraw Function: Draw input binary picture on specified area of graphic area.

Syntax	void lcdDraw (uint16_t x_start, uint16_t y_start, uint16_t x_end, uint16_t y_end, uint8_t * pic_data, uint8_t mode);
Parameters	x_start X coordinate of the top-left point of input picture. (UNIT=pixel) y_start Y coordinate of the top-left point of input picture. (UNIT=pixel) x_end X coordinate of the bottom-right point of input picture. (UNIT=pixel) y_end Y coordinate of the bottom-right point of input picture. (UNIT=pixel) pic_data Bit map data will be drawn. Input 0 will reverse pixels of specified area. mode DRAW_NORMAL:Draw the picture normally. DRAW_REVERSE : Reverse the picture and then draw the picture.
Return value	None.

lcdFillByte Function: Fill input byte value on specified area of graphic layer.

Syntax	void lcdFillByte (uint16_t x_start, uint16_t y_start, uint16_t x_end, uint16_t y_end, uint8_t data, uint8_t mode);
Parameters	x_start X coordinate of the top-left point of specified area. (UNIT=pixel) y_start Y coordinate of the top-left point of specified area. (UNIT=pixel) x_end X coordinate of the bottom-right point of specified area. (UNIT=pixel) y_end Y coordinate of the bottom-right point of specified area. (UNIT=pixel) pic_data Byte value will be filled. Mode DRAW_NORMAL: Fill input value normally. DRAW_REVERSE : Reverse the input value and then fill it on the specified area
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.

lcdDrawBit Function: ON/OFF the pixel on specified location of graphic layer.

Syntax	void lcdDrawBit (uint16_t x, uint16_t y, char bit_value);
Parameters	x X coordinate of the location will be drawn. (UNIT=pixel) y Y coordinate of the location will be drawn. (UNIT=pixel) bit_value 1 : ON the pixel 0 : OFF the pixel
Return value	None.

lcdDrawRect Function: Draw rectangle by single line on graphic layer.

Syntax	void lcdDrawRect (uint16_t x_start, uint16_t y_start, uint16_t x_end, uint16_t y_end,);
Parameters	x_start X coordinate of the top-left point of rectangle. (UNIT=pixel) y_start Y coordinate of the top-left point of rectangle. (UNIT=pixel) x_end X coordinate of the bottom-right point of rectangle. (UNIT=pixel) y_end Y coordinate of the bottom-right point of rectangle. (UNIT=pixel)
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)

5.3.5 Touch function

Header file : touch.h
object file : touch.o

touchInit Function: Initial Touch panel.

Syntax	void touchInit(int res_x, int res_y,);
Parameters	res_x X resolution of touch panel. res_y Y resolution of touch panel.
Return value	None.

touchGet Function: Read touch data from touch panel

Syntax	uint8_t touchGet(uint16_t * pX, uint16_t * pY uint8_t is_calibration);
Parameters	pX X Coordinate value from touch data pY Y Coordinate value from touch data is_calibration Flag to indicate current operation is for calibration or not.
Return value	TRUE There is data from touch panel and the data is saved at pX and pY. FALSE There is no data from touch panel.

setCalibrationMatrix Function: Set Calibration calculation matrix

Syntax	void setCalibrationMatrix(POINT * ptDisplay, POINT * ptTouch, int n);
Parameters	ptDisplay LCD reference Coordinate for calibration. ptTouch Touch Coordinate for calibration n Coordinate No. for calibration
Return value	None.

getDisplayPoint Function: Translate touch coordinate into LCD coordinate.

Syntax	void getDisplayPoint(uint16_t x, uint16_t y, int16 * pX, int16 * pY);
Parameters	x Touch X Coordinate. y Touch Y Coordinate. pX LCD X Coordinate translated from Touch X Coordinate pY LCD Y Coordinate translated from Touch Y Coordinate
Return value	None.

touchSaveCaliParam Function: Save calibration parameters into i2c eeprom

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

touchReadCaliParam Function: Read calibration parameters from i2c eeprom

Syntax	void touchReadCaliParam ( );</br>
Parameters	None.
Return value	TRUE: Read calibration parameters successfully. FALSE: There is no calibration parameters in eeprom.

touchDrawCalPoint Function: Draw all calibration cross marks on LCD

Syntax	uint8_t touchDrawCalPoint (POINT* ptCal, int n, char bDraw);
Parameters	ptCal Calibration Coordinates. n Calibration Coordinate No. bDraw TRUE: Draw all marks. FALSE: Clear all marks.
Return value	None.

touchDrawCross Function: Draw the specified calibration cross mark on LCD

Syntax	uint8_t touchDrawCross (POINT* ptCal, int inx, char bDraw);
Parameters	ptCal Calibration Coordinates. n The index of the cross will be draw. bDraw TRUE: Draw the specified mark. FALSE: Clear the specified mark.
Return value	None.

touchDrawFixPoint Function: Draw a fixed mark at specified position which finished calibration on LCD

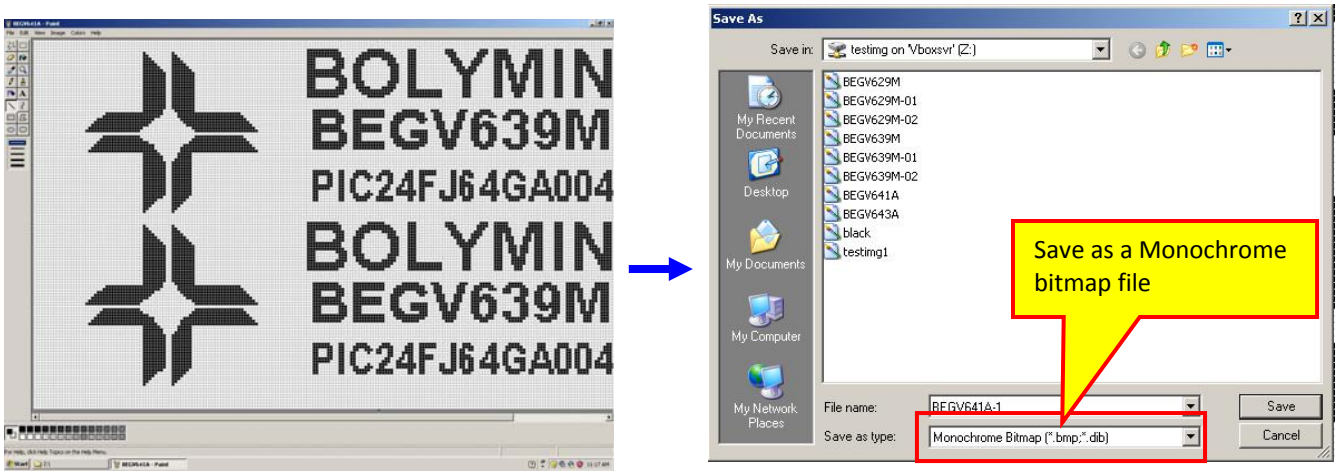
Syntax	uint8_t touchDrawCross (POINT* ptCal, int inx);
Parameters	ptCal Calibration Coordinates. n The index of the cross will be draw.
Return value	None.

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.

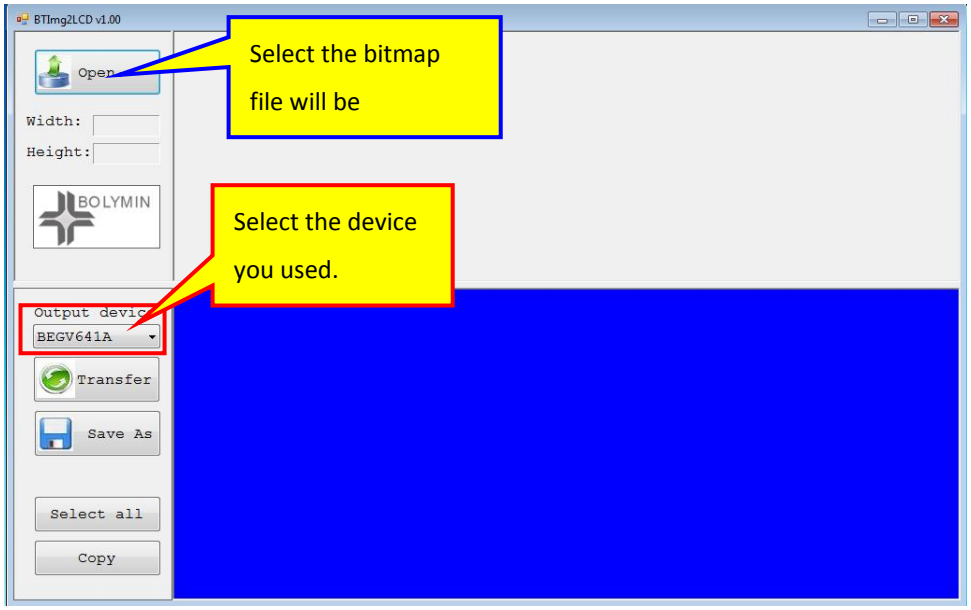
Step 1 - User needs to create a monochrome bitmap file (no grayscale) by drawing software, such as **msPaint** before running BTImg2LCD program. **It is recommended that the width and height of the picture to be a multiple of 8.**



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 

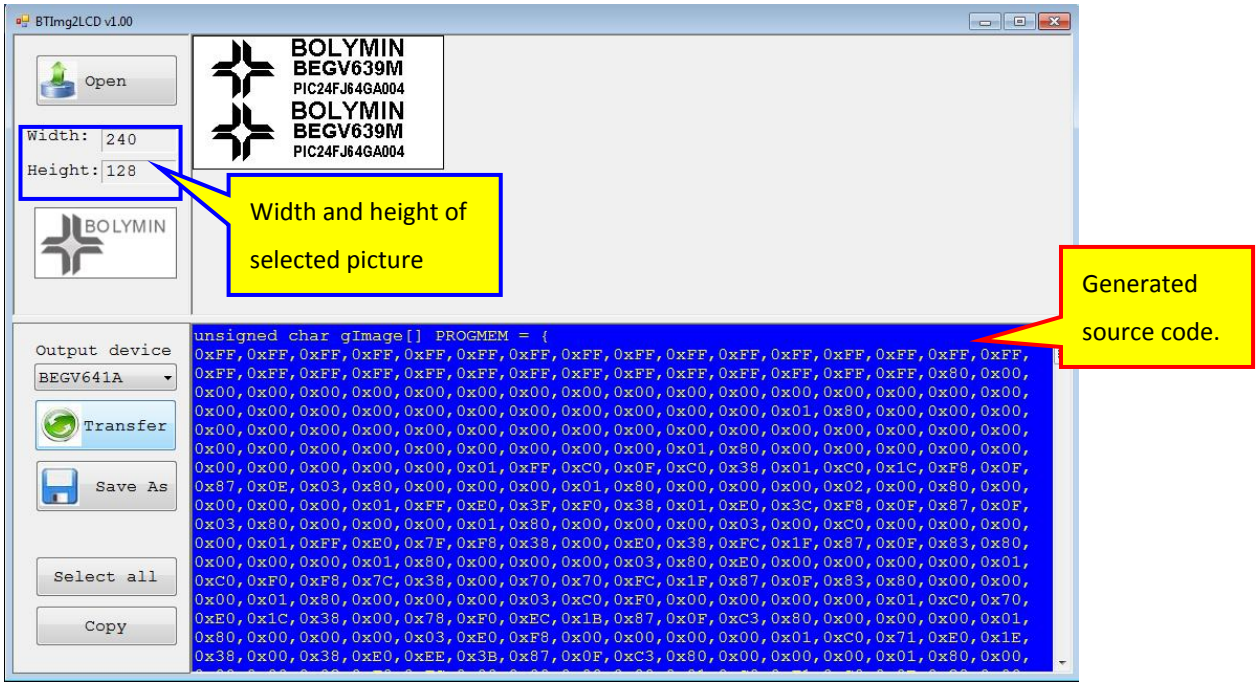
to run **BTImg2LCD.exe**. Here is the main screen :.



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.

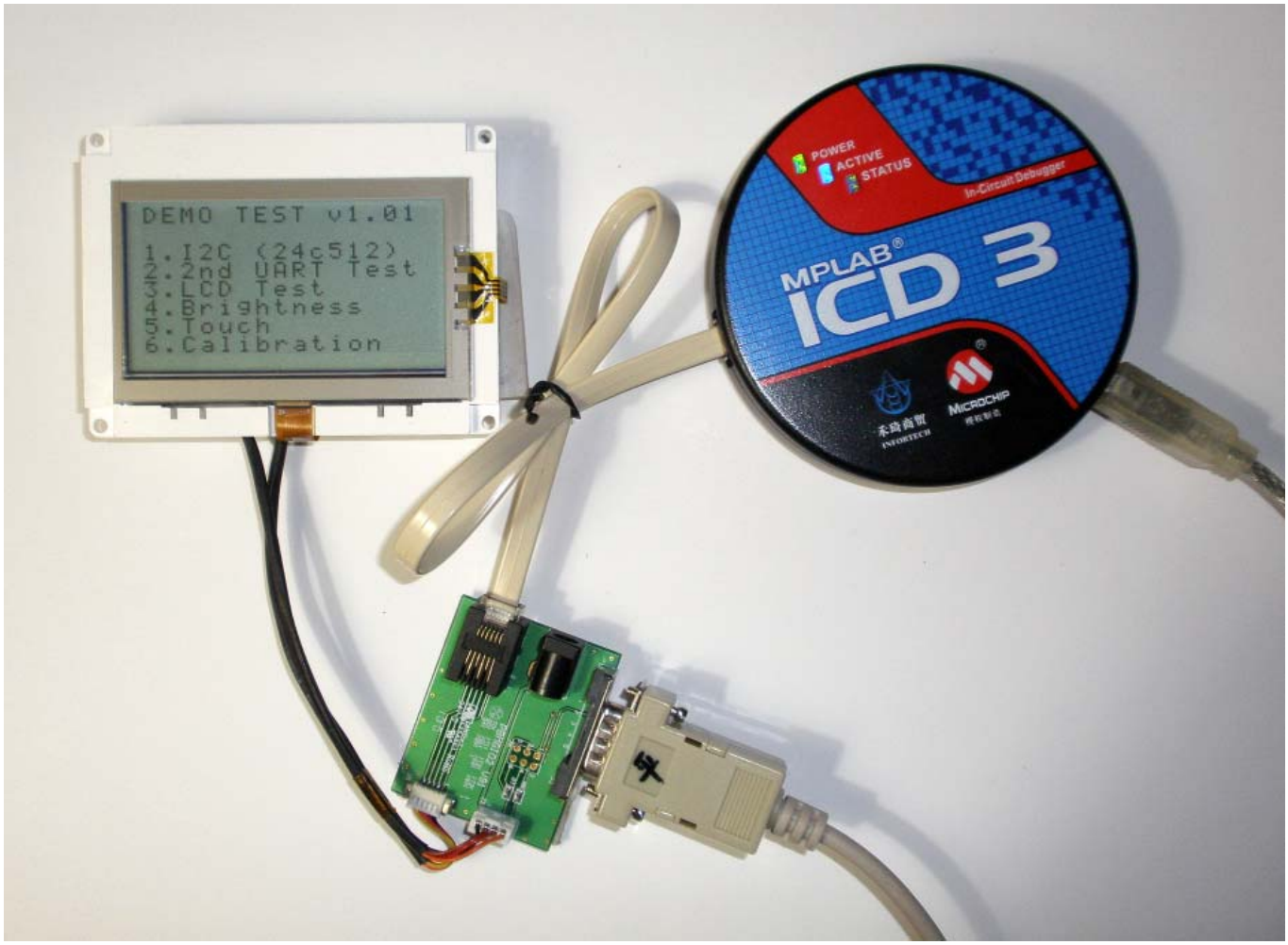
There are 2 methods to use the generated source code in your program:

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 BEGV627M is MGI02-0$ - Bolymin proprietary.
To operate our BEGV627M, the customer will need MPLAB ICD3 and MGI02-0$.



Appendix C - Notes when running under Windows 7

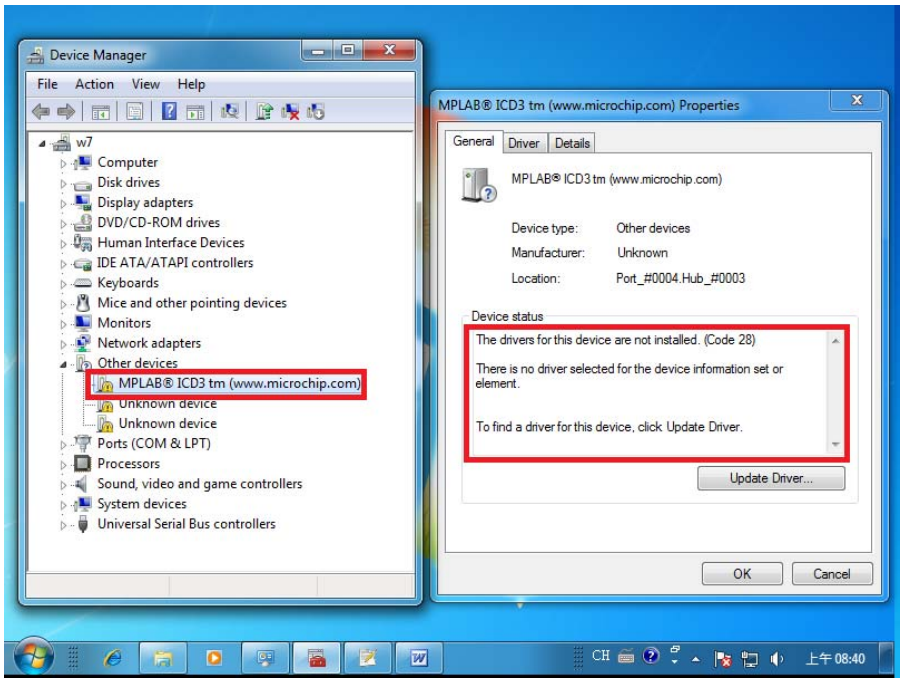
1. User may notice the ICD3 status LED blinks when running under windows 7 and the OS does not recognize ICD3 as a valid USB device. To fix the problem, install ICD3 from the following folder –
   for 32bit OS, find driver under **c:\Program Files\Microchip\MPLAB IDE\ICD3\drivers**
   for 64bit OS, find driver under **c:\Program Files\Microchip\MPLAB IDE\ICD3\VistaXP64**

2. There is no native hyper terminal program under windows 7. Users may copy **hypertrm.exe** and **hypertrm.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.

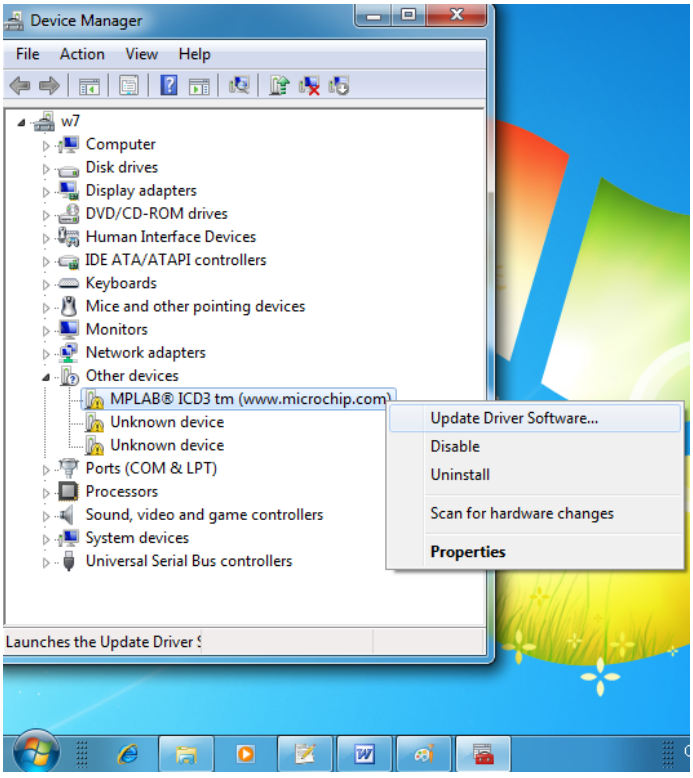
Here is the step guide –

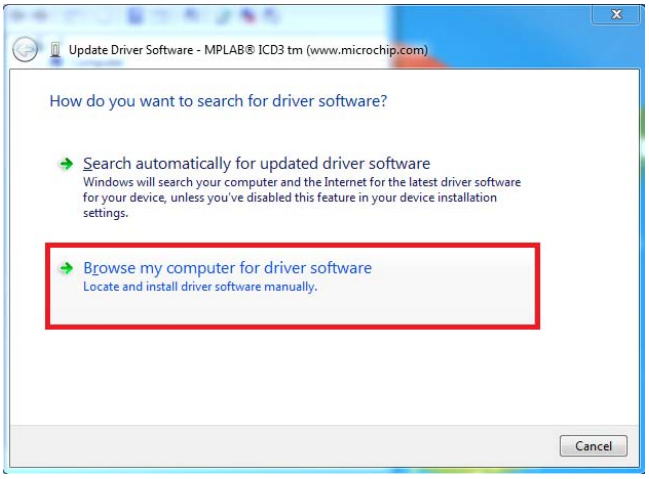
1. ICD3 driver installation – After plug in ICD3 with PC-USB, user will notice the blink of status LED on ICD which indicates an error condition.

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

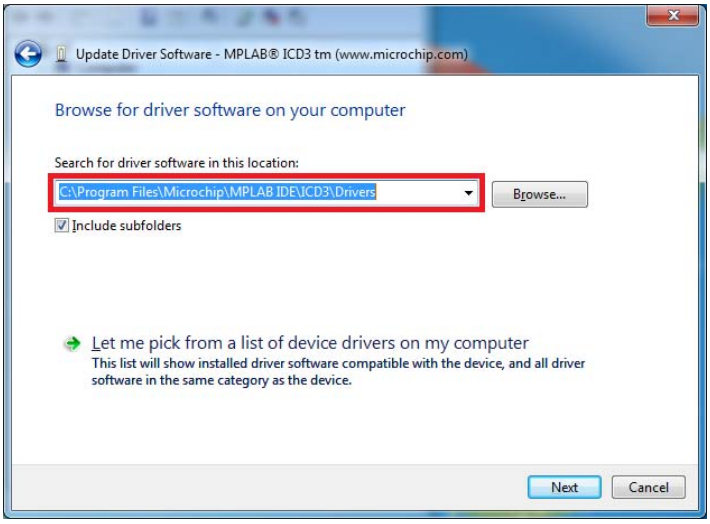


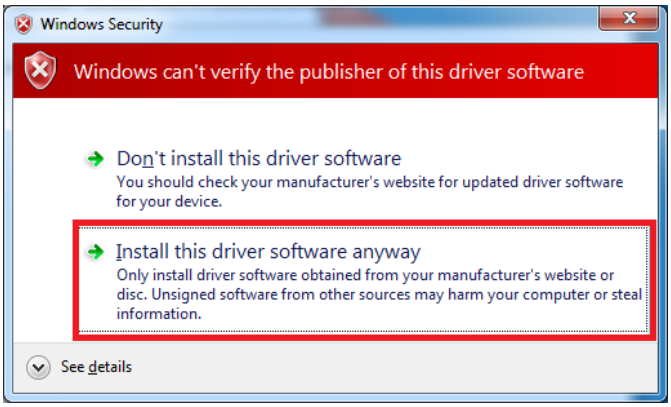
Right-click to update driver as illustrated below -



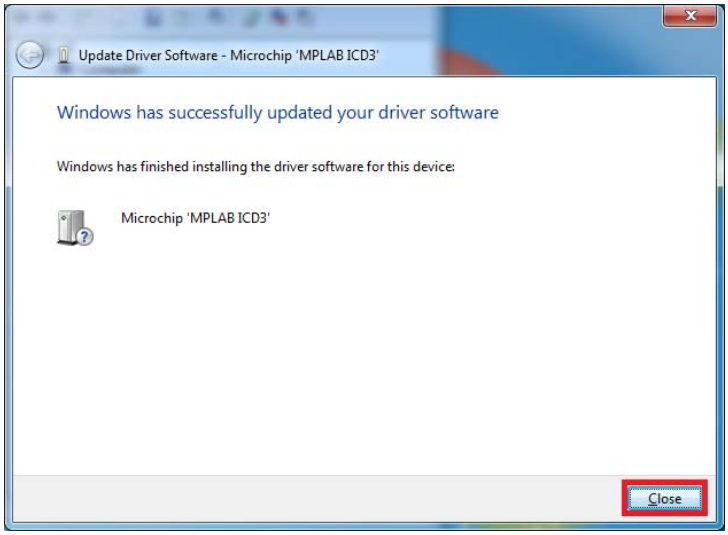


Select driver from as illustrated, and bypass the warning.

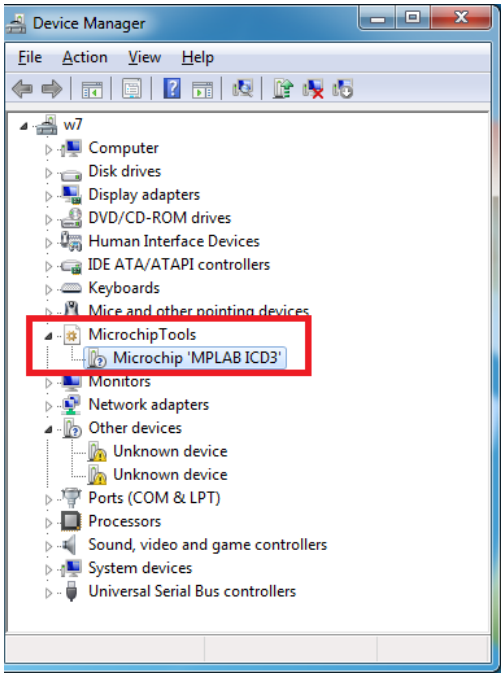




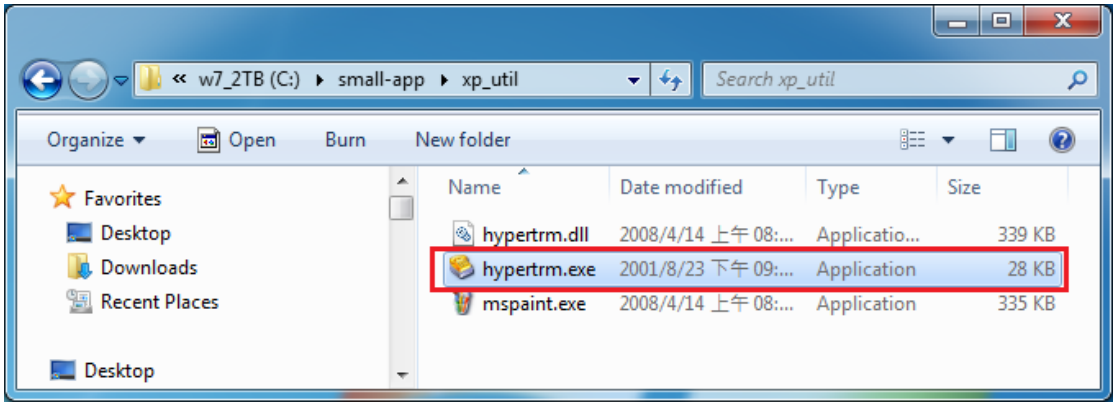
Wait for a while for driver installation, user will see the following screen upon completion.



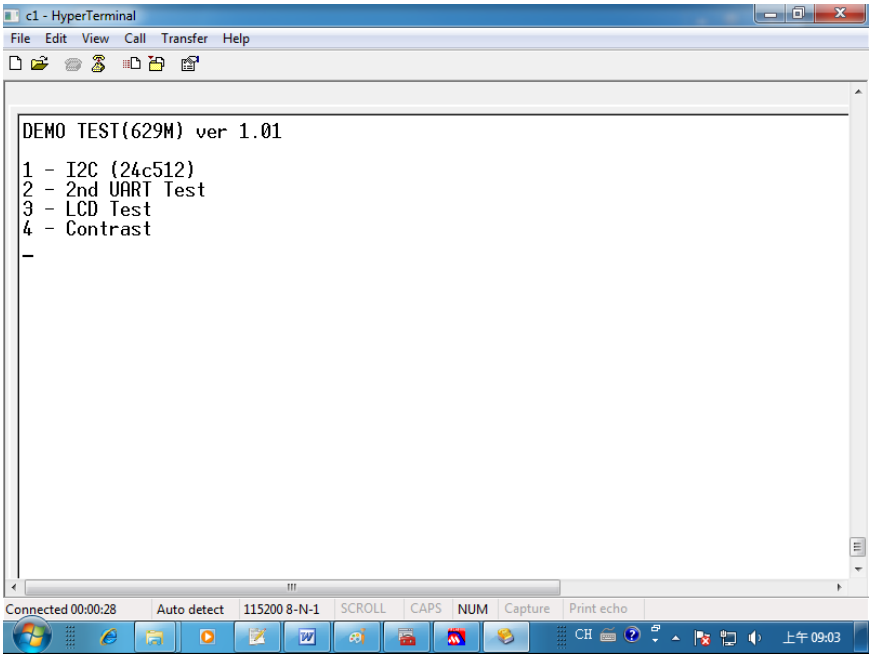
Note the status LED on ICD is now off and the device manager will show ICD3 driver installed.



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 BEGV627M User Manual>

Copyright

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