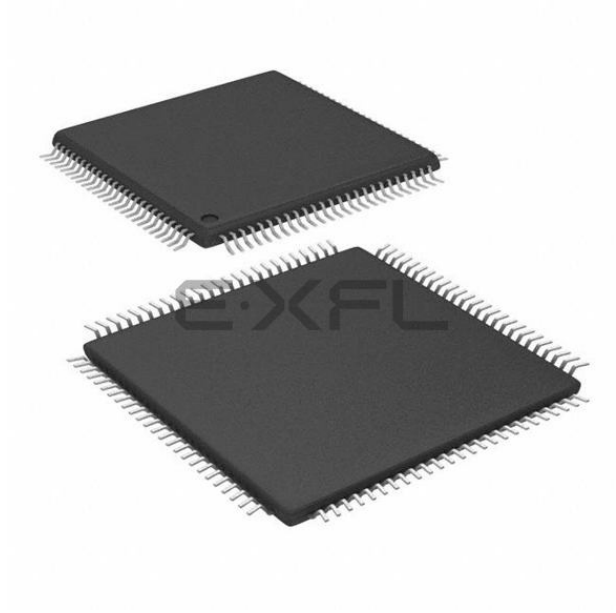
Microchip Technology - PIC32MX360F256LT-80I/PT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	MIPS32® M4K™
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	I²C, IrDA, LINbus, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	85
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx360f256lt-80i-pt

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	*MPLAB® IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF, 'A'`
Italic Courier New	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mcc18 [options]` *`file`* `[options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`
	Represents code supplied by user	`void main (void) { ... }`

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the MPLAB C18 and MPLAB C32 C compilers; and MPLAB LIB30 object librarians.
- **Emulators** – The latest information on Microchip in-circuit emulators.This includes the MPLAB ICE 2000 and MPLAB ICE 4000.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 2.
- **MPLAB® IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB SIM simulator, MPLAB IDE Project Manager and general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include the MPLAB PM3 and PRO MATE® II device programmers and the PICSTART® Plus, PICkit® 1 and PICkit® 2 development programmers.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

DOCUMENT REVISION HISTORY

Revision A (October 2007)

- Initial Release of this Document.

Revision B (March 2008)

- Updates

Chapter 1. PIC32 Features

1.1 INTRODUCTION

The PIC32 is a 32-bit family of general purpose microcontrollers from Microchip Technology. It offers 80+ DMIPS performance with a wide variety of on-chip peripherals. It employs industry leading M4K MIPS32 core from MIPS Technologies, Inc. All members in the PIC32 family use programming interface similar to other Microchip PIC® microcontrollers. In addition, PIC32 microcontrollers are pin-to-pin compatible with the PIC24FJ128GA family of 16-bit microcontrollers.

1.2 HIGHLIGHTS

The PIC32 family offers a number of features to enable a wide variety of applications. The following subsections list all the key features grouped in major categories.

Performance:

- Up to 80 MHz, MIPS M4K 32-bit core with 5 stage pipeline
- High-performance hardware multiply/divide unit – 1 multiply per clock
- Programmable user and kernel memory partition for enhanced application stability
- Multiple register sets for reduced interrupt latency
- Hardware assisted single-cycle register bits manipulations
- 128-bit wide Flash memory to shorten individual instruction fetch time
- 256 bytes of high-speed cache memory with instruction and ROM data prefetch buffer
- Available DMA controller with integrated CRC calculation and pattern-based transfer termination
- Includes USB On-The-Go controller for USB device, host, or dual-role applications
- USB controller has own dedicated DMA interface

Power Management:

- 2.3 to 3.6V operation
- Full-speed operation over entire voltage range
- Various low-power modes including RUN, IDLE and SLEEP
- I/O transfers via DMA in IDLE mode
- Programmable peripheral clock
- Individual peripheral ON/OFF control and operation during IDLE mode
- Multiple clock sources

Scalability:

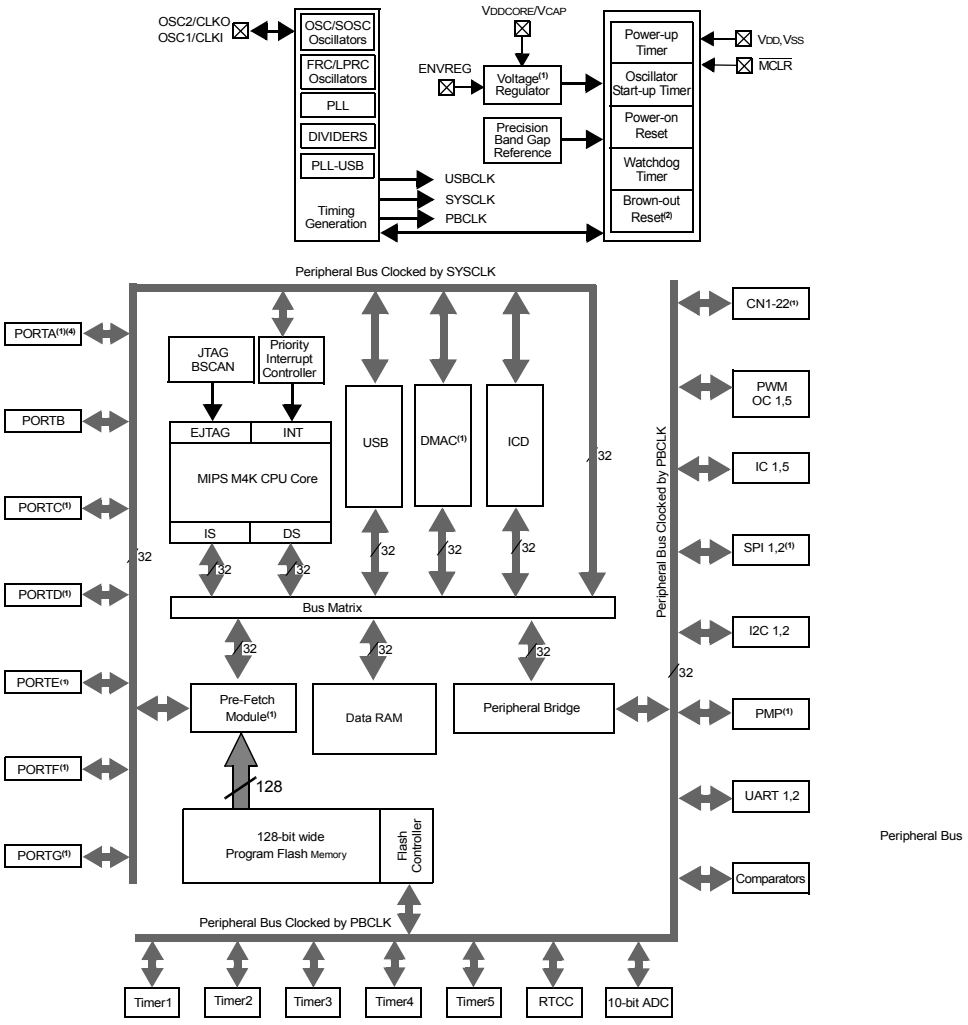
- Industry known MIPS32-compatible M4K CPU core with 5 stage pipeline
- Large family of devices with Flash memory options from 32 KB to 512 KB
- Pin compatible with 64/100 pin PIC24FJXXXGA family of 16-bit microcontrollers

Chapter 3. PIC32 Architecture

3.1 INTRODUCTION

The PIC32 family of MCUs combines the MIPS M4K core together with powerful peripherals and embedded Flash and RAM memory to address a wide range of applications.

FIGURE 3-1: PIC32 BLOCK DIAGRAM



Note 1: Not all pins or features are implemented on all device pinout configurations. See Table 1-4 for I/O port pin descriptions.

2: Some features are not available on certain device variants.

3: BOR functionality is provided when the on-board voltage regulator is enabled.

4: PORTA is not present on 64 pin devices

Processor core:

- MIPS M4K with 5-stage pipeline
- MIPS32-compatible Release 2 Instruction Set
- MIPS16e™ Code Compression to improve code density by up to 40%
- GPR shadow registers to minimize latency for interrupt handlers
- Bit field manipulation instructions
- High-performance Multiply/Divide Unit:
  - Maximum issue rate of one 32x16 multiply per clock
  - Maximum issue rate of one 32x32 multiply every other clock
- Static implementation: minimum operating frequency 0 MHz
- 2.3 to 3.6V operation with full speed over entire range
- Low-power modes including RUN, IDLE, and SLEEP

Memory:

- Unified 4GB virtual memory space
- Fixed Memory Mapping Translation (FMT) mechanism
- Flexible partitioning into kernel and user accessible memory segments for increased application stability

Pre Fetch Cache:

- 16 lines, each 128-bit wide, instruction Prefetch buffer
- Ability to load and lock lines – useful to create SW breakpoints in Flash and minimize interrupt latency

Interrupt Controller:

- Fully programmable interrupt controller with Single or Multi vector mode, supporting up to 95 IRQs.
- Multiple priorities and subpriorities for each vector
- Highest priority interrupt has dedicated register set for reduced interrupt latency

DMA Controller:

- Up to 4 independent channels
- Memory-to-Memory, Memory-to-Peripheral, and Peripheral-to-Memory transfers
- Programmable trigger from any IRQ
- Chainable channels, stop on match detection, Auto-Enable mode
- Data transfers can occur while the core is in IDLE mode
- Integrated programmable CRC engine: calculates on the fly while the data is transferred.

Enhanced Parallel Master Port:

- 8- and 16-bit data interface
- Up to 16-bit address lines, expandable using GPIO lines
- 2 Chip Select lines

Communication channels:

- USB 2.0 compliant (FS. 12 Mbps), OTG, Host and Device-only capable
- 2 enhanced UART channels with hardware IrDA®
- 2 Master/Slave/Frame mode SPI channels
- 2 Master/Slave I^2C channels, 10/7 bits mode addressing, broadcast capable

Analog-to-Digital Converters:

- Up to 16 Channels, each 10-bit resolution ADCs
- Up to 500+ kilo-samples per second (ksps) conversion speed
- Software selectable Internal or External voltage reference
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 16 word conversion result buffer
- Selectable Buffer Fill modes
- Eight result alignment options
- Operation during CPU Sleep mode

Timers:

- 5 16-bit timer/counter with the ability to form up to (2) 32-bit timer/counters
- Software-selectable internal or external clock source
- Asynchronous timer/counter with built-in oscillator
- Programmable interrupt generation and priority
- Gated external pulse counter
- Software-selectable prescalers.
- Operational during CPU Sleep mode

Core Timer:

- 32 bit timer in CPU for implementing a timer interrupt function.

RTCC (Real-Time Clock and Calendar):

- Time with hours, minutes and seconds
- Calendar with weekday, date, month and year
- Leap year detection
- Highly configurable alarm
- Calibration of up to 260 ppm of crystal error

Debug and Programming:

- 6 instructions and 2 data breakpoints
- 2 complex breakpoint logic blocks with qualified/primed breakpoint triggers, Pass counters, and stopwatch timers.
- 4-wire EJTAG and 2-wire Microchip interface
- 2-wire Microchip Interface:
  - 6 real-time read/write capture logic blocks
  - Read/write access to all data RAM and SFRs without stopping CPU
- Instruction Trace Port:
  - 5-Wire, nonintrusive trace port
  - Triggered by complex breakpoint logic block

GPIO:

- 5V tolerant inputs
- Individual output pin open-drain enable/disable
- Individual input pin weak pull-up enable/disable
- Monitor selective inputs and generate interrupt on mismatch condition

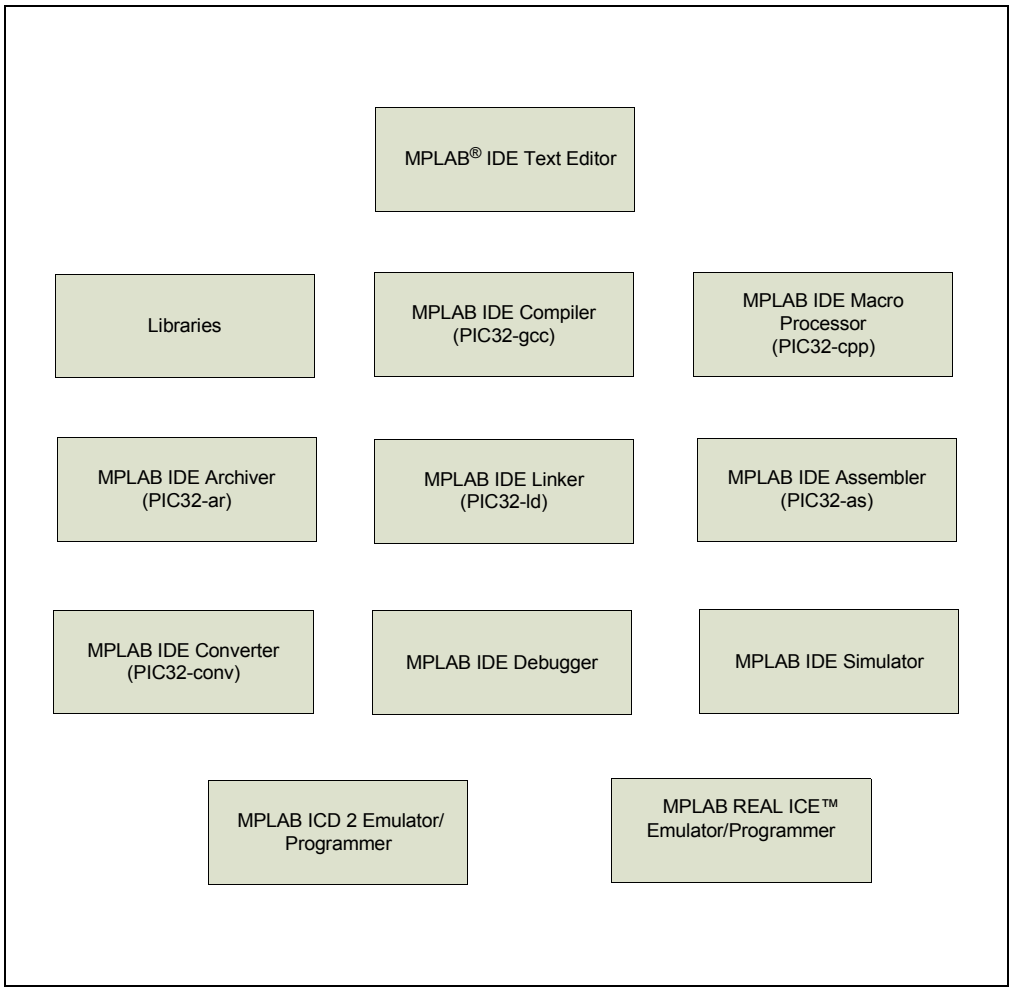
Note:	Input pin weak pull-up and interrupt on mismatch features are available only on select I/O pins.

4.4 MPLAB C32 C COMPILER

The MPLAB C32 C Compiler package allows you to build your applications and contains the following key software components (see Figure 4-1).

- **PIC32-gcc** Compiler: complete ANSI C compiler with powerful integration capabilities and efficient code optimization. The compiler provides symbol information that is used by the MPLAB IDE debugger.
- **PIC32-gpp** Macro Processor: used automatically by the compiler to transform the program before compilation.
- **PIC32-as** Assembler: full-featured macro assembler.
- **PIC32-ld** Object Linker: links relocatable objects created by the assembler/compiler with those from precompiled libraries.
- **PIC32-ar** Archiver and Librarian: manages the creation and modification of library files of precompiled code.
- **PIC32-conv**: converts ELF executable files into ASCII or binary format, suitable for downloading onto a PROM programmer or evaluation board.

FIGURE 4-1: MPLAB® IDE DEVELOPMENT TOOLS COMPONENTS



4.5 PERIPHERAL LIBRARIES

PIC32 MCUs integrate a large number of on-chip high-performance peripherals.

To accelerate the usage of these peripherals, the MPLAB C32 compiler for PIC32 includes software peripheral libraries compatible with the 16-bit Microchip MCUs. The peripheral libraries are distributed in source and object format along with a detailed API description document. Software applications using MPLAB C32 compiler may call peripheral library functions by simply including the appropriate header file in their source files – the MPLAB C32 compiler has built-in knowledge of library header and archive files.

The peripheral include files are located in `C:\Program Files\Microchip\MPLAB C32\pic32mx\include\peripheral` and the full source code is located in `C:\Program Files\Microchip\MPLAB C32\pic32-libs\peripheral` location in corresponding peripheral sub-directory.

The Peripheral Library contains following include files:

4.7 DEMONSTRATION, DEVELOPMENT AND EVALUATION BOARDS

A wide variety of demonstration, development and evaluation boards for PIC32 MCUs allow quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification. These boards support a variety of features, including LEDs, switches, RS-232 interfaces, LCD displays, etc. The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

Currently, the following boards are available:

1. PIC32MX Starter kit (DM320001).
2. Explorer 16 board (DM240001) with PIC32 Plug-in Modules.

4.8 TECHNICAL DOCUMENTATION

The current set of documents that are available for PIC32 MCUs:

- Application Notes:
  - AN833, "*Microchip TCP/IP Stack"* (DS00833)
  - AN1107, "*HTTP Server for the Microchip BSD TCP/IP Stack"* (DS01107)
  - AN1108, "*Microchip TCP/IP Stack with BSD Socket API"* (DS01108)
  - AN1109, "*An SNMP Agent for the Microchip TCP/IP Stack"* (DS01109)
  - AN1111, "*The Microchip FTP Server Using BSD Socket API"* (DS01111)
  - AN1140, "*USB Embedded Host Stack*" (DS01140)
  - AN1141, "*USB Embedded Host Stack Programmer's Guide*" (DS01141)
  - AN1142, "*USB Mass Storage Class on an Embedded Host*" (DS01142)
  - AN1143, "*USB Generic Client on an Embedded Host*" (DS01143)
  - AN1144, "*USB HID Class on an Embedded Host*" (DS01144)
  - AN1145, "*Using a USB Flash Drive on an Embedded Host*" (DS01145)
  - AN1176, "*USB Devcie Stack for PIC32 Programmer's Guide*" (DS01176)
  - AN1166, "*USB Generic Function on an Embedded Device*" (DS01166)
  - AN1163, "*USB HID Class on an Embedded Device*" (DS01163)
  - AN1169, "*USB Mass Storage Class on an Embedded Device*" (DS01169)
  - AN1164, "*USB CDC Class on an Embedded Device*" (DS01164)
- Data Sheets:
  - DS61143 – *PIC32MX Family Data Sheet*
- Family Reference Manuals:
  - DS61132 – *PIC32MX Family Reference Manual*
- Code Examples:
  - PIC32 examples available in `C:\ProgramFiles\Microchip\MPLAB C32\examples` and on www.microchip.com/pic32.
- Errata (DS80350, DS80367)
- Migration Documents – available in future
- Design Notes, Tips and Tricks – available in future
- Development Tool Ordering Guide – available in future

Chapter 5. Step-by-Step Procedures to Setup, Build, and Run a Demo Project

5.1 INTRODUCTION

In this step-by-step procedure, the basic concepts of the MPLAB Project Manager, Editor and Debugger will be presented. You will create a simple project and understand the debug capabilities of MPLAB IDE.

No previous MPLAB IDE knowledge is assumed. For complete features set and comprehensive technical details of MPLAB IDE and its components, please visit our web site (www.microchip.com/ide).

5.2 HIGHLIGHTS

Items discussed in this chapter are:

- MPLAB IDE Setup
- Step-by-Step Guide Overview
- Selecting the Device
- Creating the Project
- Setting Up Language Tools
- Naming the Project
- Adding Files to the Project
- Attaching the Debugger
- Building the Project
- Testing the Code

5.3 MPLAB IDE SETUP

5.3.1 Install MPLAB IDE

To install the MPLAB on your system, you could either use the supplied installation CD or download the latest MPLAB IDE from the Microchip web site.

- To install from a CD-ROM, just place the disk into a CD drive and follow the on-screen prompts (you could use Windows Explorer to find and execute the CD-ROM menu, `menu.exe`).
- If the MPLAB IDE was downloaded from the Microchip web site, unzip the file and execute the resulting file to install.

Note: Administrative access will be required to install the MPLAB on a PC.

5.3.2 To uninstall MPLAB IDE

- Select *Start>Settings>Control Panel* to open the control panel.
- Double click on Add/Remove Programs. Find MPLAB IDE and select it.
- Click Change/Remove to remove the program from your system.

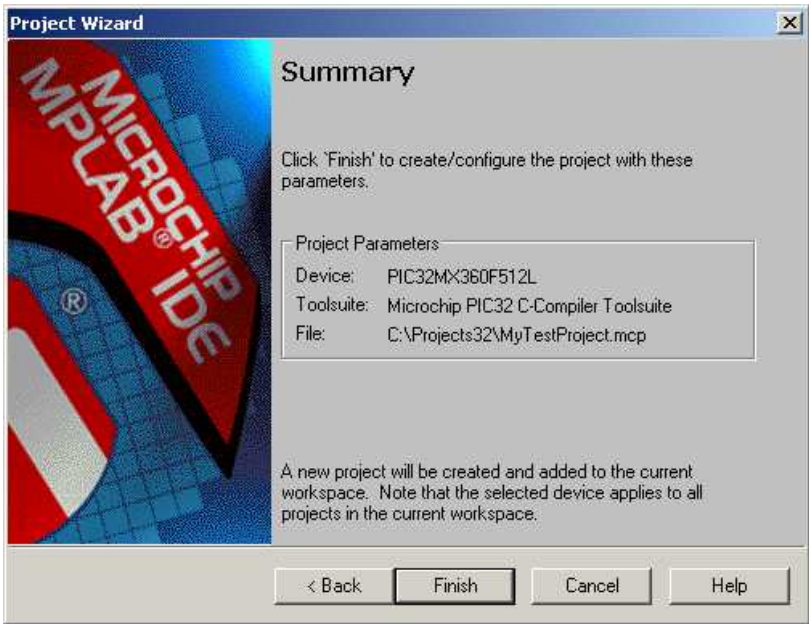
Note: Administrative access might be required in order to uninstall the MPLAB.

5.9 ADDING FILES TO THE PROJECT

Step Four of the Project Wizard allows file selection for the project. This is where we can add existing files to our project. In the current example we don't have a previously created file but we'll create one once we're done with the new project setting.

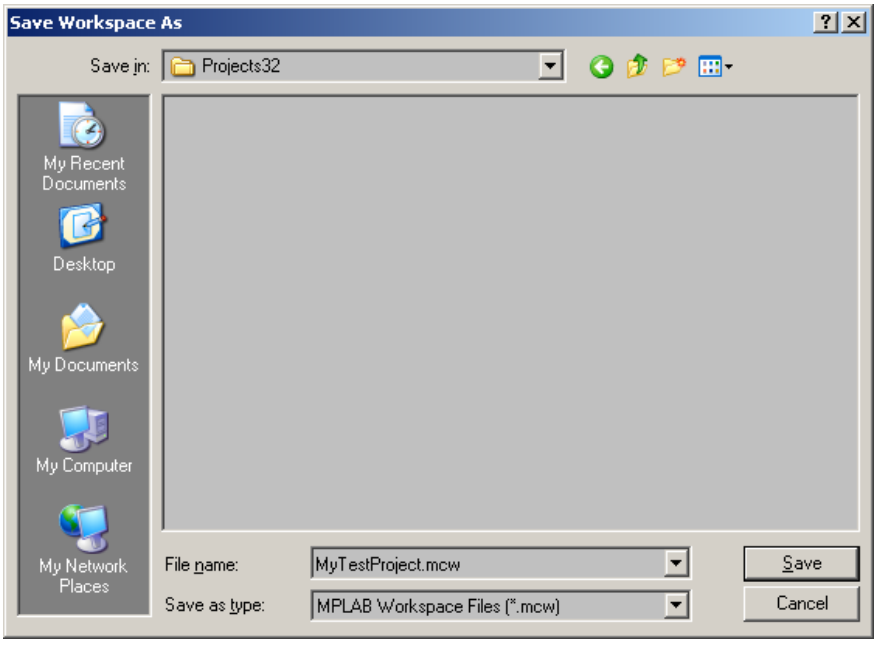
Just press **Next>** and the following Summary Screen will show up:

FIGURE 5-6: MPLAB IDE SUMMARY SCREEN



Press **Finish** and the workspace dialog shows up:

FIGURE 5-7: MPLAB IDE SAVE WORKSPACE

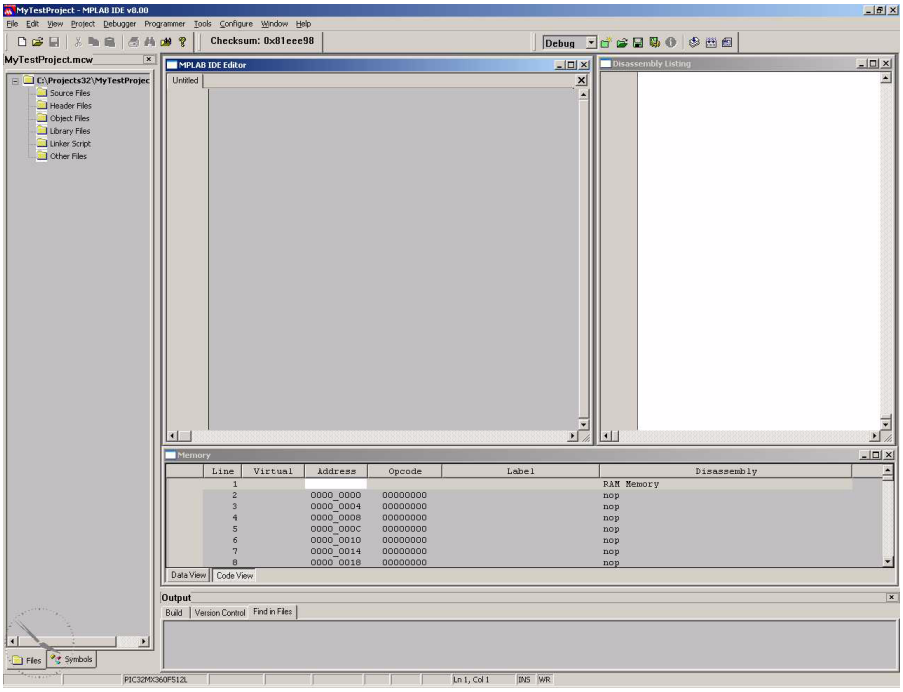


Be sure to save the workspace in the Projects32 directory and name this workspace using the same name as for the project: MyTestProject.

Click **Save**.

The project space is now completely created and it should look like that:

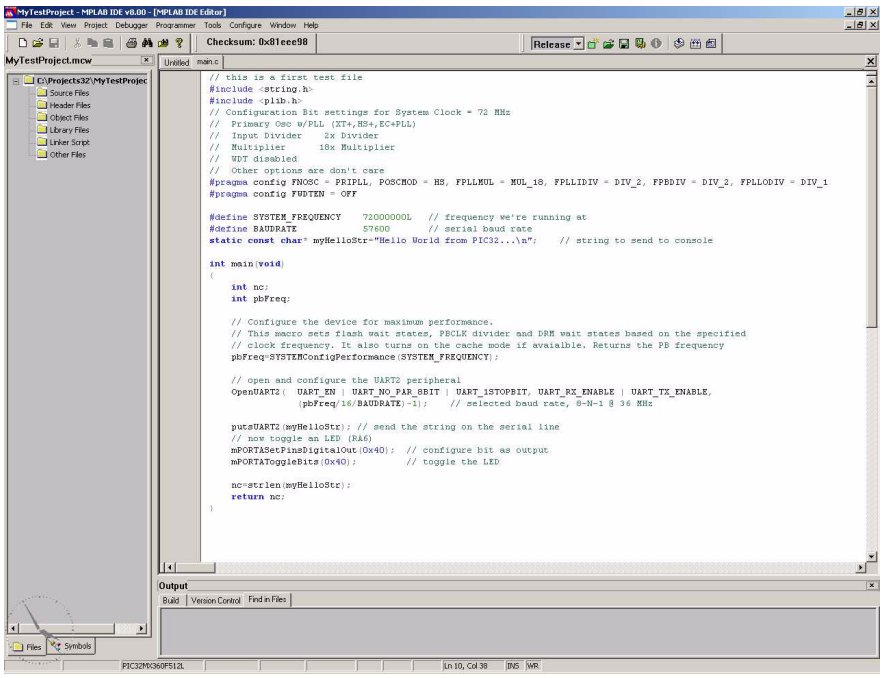
FIGURE 5-8: MPLAB IDE PROJECT SAVE



Now we'll create our C source file. If the MPLAB IDE Editor window is not open, click *File->New* from the top menu or the New File menu shortcut on the standard toolbar. The Editor window will show up.

Let's type in a very simple Hello World program in the Editor window and save it in Projects32 directory as `main.c`. We'll use the UART Peripheral Library provided by Microchip:

FIGURE 5-9: MPLAB IDE MAIN.C FILE



To add the newly created `main.c` file to our project just right click on the Source Files folder in the MPLAB IDE MyTestProject.mcp window and select Add File. Browse and select `main.c` file from the Projects32 directory.

Note: If the Project window is not open, select View->Project from the top IDE menu.

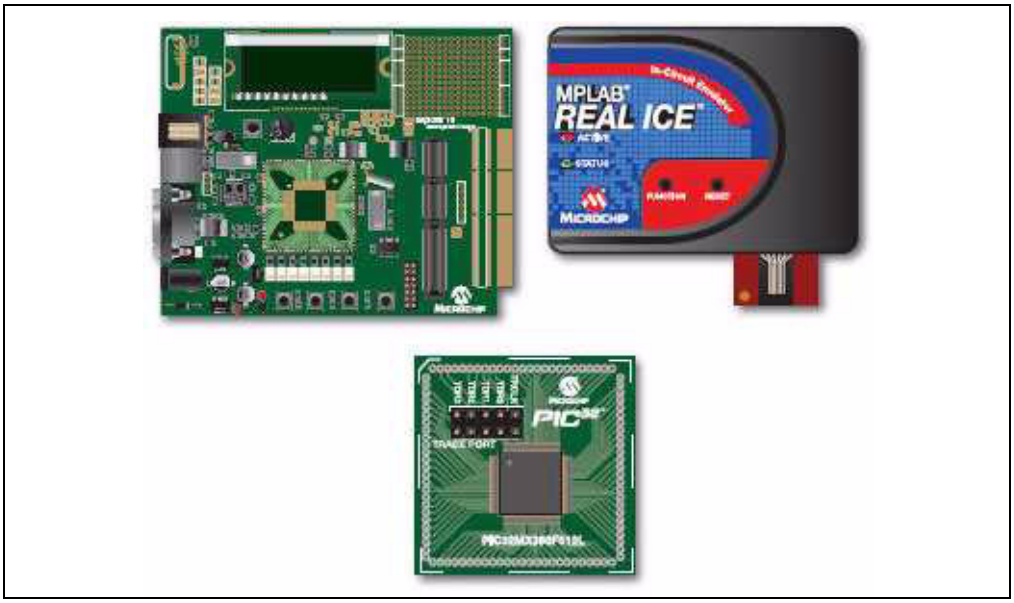
The linker script that's needed for this project is selected automatically by the MPLAB IDE. We don't need to add any other file to our project.

5.10 ATTACHING THE DEBUGGER

In order to test the code using the PIC32 Starter Kit, please refer to the PIC32MX Starter Kit User's Guide (DS61144) for a sample project and a step-by-step getting started info.

For the purpose of testing our code in this document we will use an Explorer 16 Development board (DM240001), a MPLAB REAL ICE In-Circuit Debugger (DV244005) and a PIC32MX360F512L PIM (MA320001) together with a 9V universal power supply, a serial cable and a USB cable for connecting the REAL ICE to the development board.

FIGURE 5-11: EXPLORER 16, MPLAB® REAL ICE™ AND PIC32MX360F512L PIM



Take the following steps to ensure proper connection of the REAL ICE to the Explorer 16 development board:

1. Connect the MPLAB REAL ICE module to the PC with the USB cable.
2. Connect the MPLAB REAL ICE to the Explorer 16 Development Board with the short RJ-11 cable.
3. Apply power to the Explorer 16 board.
4. From the Debugger menu, click *Select Tool > MPLAB REAL ICE* to set the MPLAB REAL ICE as the debug tool in MPLAB IDE.
5. From the Debugger menu, select Connect to connect the debugger to the device. MPLAB IDE should report in the Output window that it found the PIC32MX360F512L device.

Note: MPLAB IDE may need to download new firmware if this is the first time the MPLAB REAL ICE is being used with a PIC32 device. Allow it to do so.

Once you have performed these steps, go to the MPLAB IDE window and from the *Debugger->Select Tool* menu select the debugging tool you're using to connect to the board (i.e., MPLAB REAL ICE should be within the available choices).

Once you have selected the tool, the "Debug Toolbar" should be present just below the main menu bar, together with other toolbars that may be selected.

Note: Be sure to select Debug from the MPLAB IDE Build Configuration drop down list.

5.11 BUILDING THE PROJECT

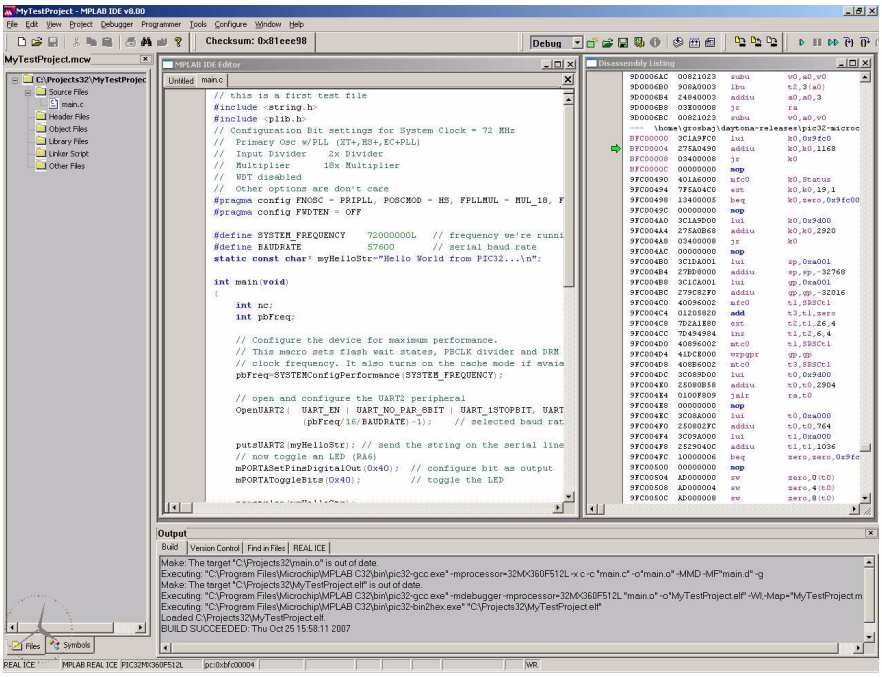
From the Project menu, we can compile and link the current files.

To build the project, select either:

- *Project>Build All*
- Right click on the project name in the project window and select Build All
- Click the Build All icon on the Project toolbar. Hover the mouse over icons to see pop-up text of what they represent.

The Output window shows the result of the build process. There should be no errors on any step.

FIGURE 5-12: MPLAB IDE PROJECT BUILD



5.12 TESTING THE CODE

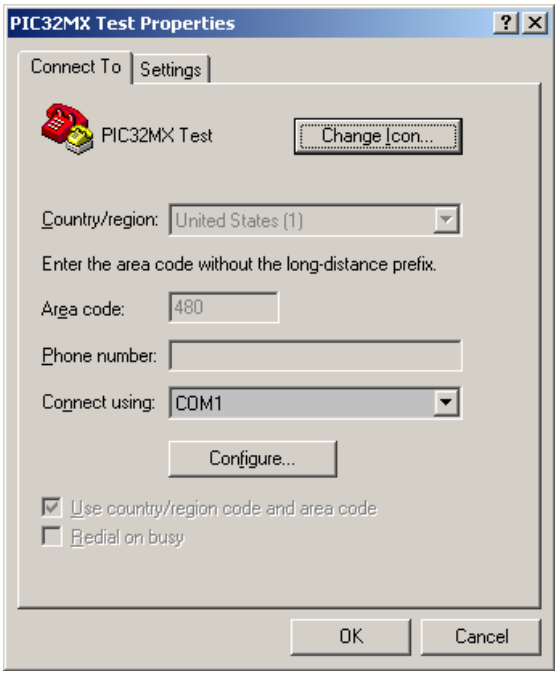
First download the hex image of the program we just built by selecting Debugger->Program from the Debugger menu or directly program from the “Debug Toolbar”. The programming operation should occur with no errors.

In order to test our example application, we need to set-up the PC part too.

Connect a serial cable between the board RS-232 connector and one of the PC’s COM ports and open a Hyper Terminal application on the PC (Programs->Accessories->Communications->Hyper Terminal).

Select a proper name for this connection, click **OK** and then, in the “Connect To” screen select the COM port that’s connected to the development board.

FIGURE 5-13: HYPER TERMINAL CONNECTION SCREEN



Click **OK** and, on the next screen, select the following communication settings:

- Bits Per Second: 57600
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

Click **OK** and the Hyper Terminal session should be connected to the serial port on the development board.

Now return to the MPLAB IDE and double click on the line:

return nc;

in the MPLAB IDE Editor window, the `main.c` program, to set a breakpoint on that line, just after the call to `strlen()`.

Click Debugger->Run from the main menu, or Run from the Debug toolbar. The program should start running and it will reach the set breakpoint. The MPLAB IDE window will look very similar to this one:

Chapter 6. Technical Support Resources

6.1 INTRODUCTION

For information about the Technical Support provided please visit Support.microchip.com.

Index

A
Attaching the Debugger
Connecting MPLAB REAL ICE 31
C
Customer Notification Service 5
Customer Support 5
D
Documentation
Conventions 3
Layout 2
I
Internet Address 4
M
Microchip Internet Web Site 4
MPLAB 15
MPLAB C32 C Compiler
PIC32-ar Archiver and Librarian 16
PIC32-as Assembler 16
PIC32-conv Converts ELF 16
PIC32-gcc Compiler 16
PIC32-gpp Macro Processor 16
PIC32-ld Object Linker 16
MPLAB IDE
First Silicon Solutions 15
MPLAB IDE MyTestProject.mcp 29
MPLAB IDE Setup 21
Install MPLAB IDE 21
Running MPLAB IDE 22
To Uninstall MPLAB IDE 21
P
PIC32 Features
MPLAB Tool-Suite 8
PIC32 Tools
MPLAB C32 C Compiler 16
MPLAB ICD 2 15
MPLAB Integrated Development Environment 15
MPLAB PM3 15
MPLAB REAL ICE In-Circuit Emulator 15
MPLAB SIM32 Device Simulator 15
R
Reading, Recommended 4
Readme 4
S
Step-by Step Procedures to Setup, Build, and Run a Demo Project
MPLAB Project Manager 21
Step-by-Step Guide Overview 22
W
WWW Address 4