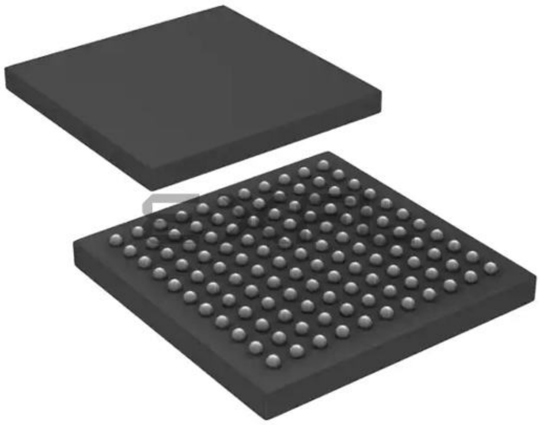
Microchip Technology - PIC32MX360F256LT-80I/BG Datasheet



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	-
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	121-TFBGA
Supplier Device Package	121-TFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx360f256lt-80i-bg

NOTES:

DOCUMENT LAYOUT

This document is divided into 6 sections. Sections 1 through 4 provide an overview of PIC32 products. Section 5 provides an overview of PIC32 development tools. Section 6 describes the procedure to setup a development environment and debug a sample application. The manual layout is as follows:

- **Chapter 1. PIC32 Features**
- **Chapter 2. PIC32 Product Family**
- **Chapter 3. PIC32 Architecture**
- **Chapter 4. PIC32 Tools**
- **Chapter 5. Step-by-Step Procedures to Setup, Build, and Run a Demo Project**
- **Chapter 6. Technical Support Resources**

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)`
		`{ ...`
		`}`

RECOMMENDED READING

This user's guide describes how to use PIC32. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Readme for the PIC32MX
For the latest information on using PIC32 microcontrollers, read the file (an ASCII text file) at the root level of the CD included in the PIC32 Starter Kit. The file generally contains the most current update information, as well as any issues that may not have been available when this document was published.

Readme Files

For the latest information on using other tools, read the tool-specific readme files in the `Readmes` subdirectory of the PIC32MX Starter Kit installation directory. The files contain update information, as well as any issues that may not have been available when this document was published.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

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

Ease of Use:

- PIC® microcontroller “look and feel” peripherals
- Standard MPLAB® tool-suite – MPLAB IDE, MPLAB C32 C Compiler, MPLAB REAL ICE™, and MPLAB ICD 2.
- Software Peripheral Libraries compatible with those for Microchip 16-bit microcontrollers
- Microchip developed middleware modules such as TCP/IP and 16-bit file system

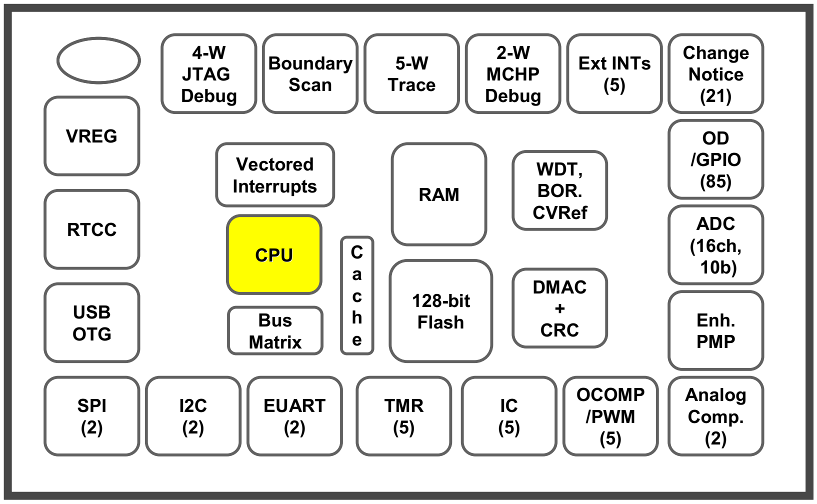
Chapter 2. PIC32 Product Family

2.1 INTRODUCTION

The PIC32 family includes scalable devices ranging from 32KB to 512KB of Flash memory. Also a rich set of peripherals – Five timers, 16 channels of 10-bit A/D Converters and communication interfaces: SPI, I^{2}C™ and UART.

Please consult the “*PIC32MX Family Data Sheet*” (DS61143) for a complete list of family variants, core and peripheral characteristics.

FIGURE 2-1: PIC32 MCU MODULES



NOTES:

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^{2}C 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

Chapter 4. PIC32 Tools

4.1 INTRODUCTION

The PIC32 microcontrollers are supported by the MPLAB Integrated Development Environment and its full range of hardware and software tools.

4.2 HIGHLIGHTS

Items discussed in this chapter are:

- MPLAB IDE
- MPLAB C32 C Compiler
- Peripheral Libraries
- Software Solutions
- Demonstration, Development and Evaluation Boards
- Technical Documentation

4.3 MPLAB IDE

The Microchip MPLAB IDE is a FREE development toolsuite for Microsoft Windows® that contains:

- A single graphical interface to all debugging tools:
  - Simulator
  - Programmers
  - Emulators
  - In-Circuit Debuggers
- A full-featured editor with color-coded context
- A project manager
- Customizable data windows with in-place editable contents
- High-level source code debugging
- Extensive on-line help

The MPLAB IDE provides integrated debugging and programming facilities using any of the following probes connected to the target processor/Development Board:

- MPLAB ICD 2 In-Circuit Debugger: a low-cost, run-time development tool.
- MPLAB REAL ICE In-Circuit Emulator: a high speed in-circuit emulator with hardware and software trace capabilities.
- FS2 JTAG probe: Developed by First Silicon Solutions, this probe uses 4-wire EJTAG interface to debug and program the PIC32 microcontrollers.
- MPLAB PM3 Device Programmer: a Microchip universal device programmer suitable for development and manufacturing purposes.
- MPLAB SIM32 Device Simulator: Accurately simulates many PIC32 peripherals and the CPU in selectable cycle and clock-accurate modes

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

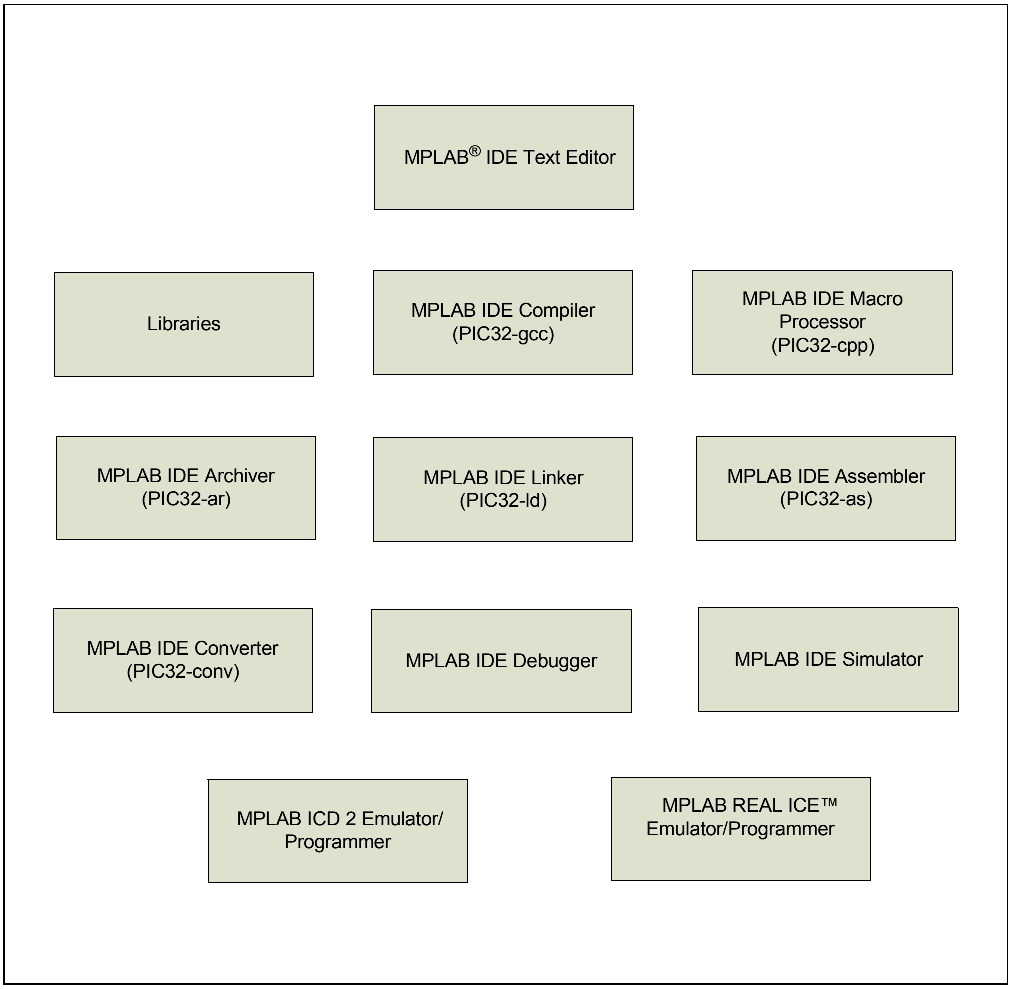

MPLAB® IDE Text Editor
Libraries
MPLAB IDE Compiler (PIC32-gcc)
MPLAB IDE Macro Processor (PIC32-cpp)
MPLAB IDE Archiver (PIC32-ar)
MPLAB IDE Linker (PIC32-ld)
MPLAB IDE Assembler (PIC32-as)
MPLAB IDE Converter (PIC32-conv)
MPLAB IDE Debugger
MPLAB IDE Simulator
MPLAB ICD 2 Emulator/ Programmer
MPLAB REAL ICE™ Emulator/Programmer

TABLE 4-1: PERIPHERAL LIBRARY

Peripheral	Header File	Description
ADC	`adc10.h`	Library for the Analog-to-Digital converters support
Memory Org	`bmx.h`	Library for configuring the Bus Matrix internal buses, priorities and memory layout
Analog Comparators	`cmp.h`	Library for the Comparator modules support
Comparator Voltage Reference	`cvref.h`	Library for the Comparators Voltage reference support
DMA Controller	`dma.h`	Library for the DMA Controller and CRC module support
Nonvolatile (Flash) Memory	`nvm.h`	Library providing access to Flash erase/program functions
I^{2}C™	`i2c.h`	Library for I^{2}C™ module support
Input Capture	`incap.h`	Library for the Input Capture module suport
Interrupts	`int.h`	Library for the vectored Interrupt Controller support
Lock	`lock.h`	Library for the system lock and unlock support
Output Compare	`outcompare.h`	Library for Output Compare module support
Oscillator	`osc.h`	Library for Oscillators support
Prefetch Cache	`pcache.h`	Library for prefetch-cache module support
Power Modes	`power.h`	Library for Sleep and Idle operationsl power modes support
Parallel Master Port	`pmp.h`	Library for Parallel Master Port support
GPIO	`ports.h`	Library for configuring the I/O ports, reading or writing I/O data
Reset	`reset.h`	Library for Reset control and status
RTCC	`rtcc.h`	Library for Real-Time Clock and Calendar support
SYSTEM	`system.h`	Library for system level operations
SPI	`spi.h`	Library for Serial Peripheral Interface support
Timer	`timer.h`	Library to support 16 and 32-bit timers
UART	`uart.h`	Library supporting the enhanced UART module
Watchdog	`wdt.h`	Library to support the Watchdog Timer module

4.6 SOFTWARE SOLUTIONS

At the time of writing this document, Microchip had not released any drivers. Microchip does intend to develop drivers for key communication peripherals.

4.6.1 Middleware

The following list provides the Microchip middleware components available at the time of writing this document:

- 16-bit file system on SD Memory to support Microsoft MS-DOS file system.
- Two types of TCP/IP stack – 1) Microchip legacy TCP/IP Stack – fully compatible with Microchip 8- and 16-bit Stack, 2) Microchip BSD TCP/IP Stack – A Berkeley socket API stack with many advanced features.
- TCP/IP middleware:
  - FTP server to allow the application running on the target PIC32 machine to be a File Transfer Protocol server.
  - SNMP Agent to monitor PIC32 products using SNMP protocol.
  - Web server to accept HTTP requests from Web browsers clients.
- USB middleware:
  - USB Embedded Host Stack (HID and Mass Storage)
  - USB Device Stack (HID, Mass Storage and CDC class)

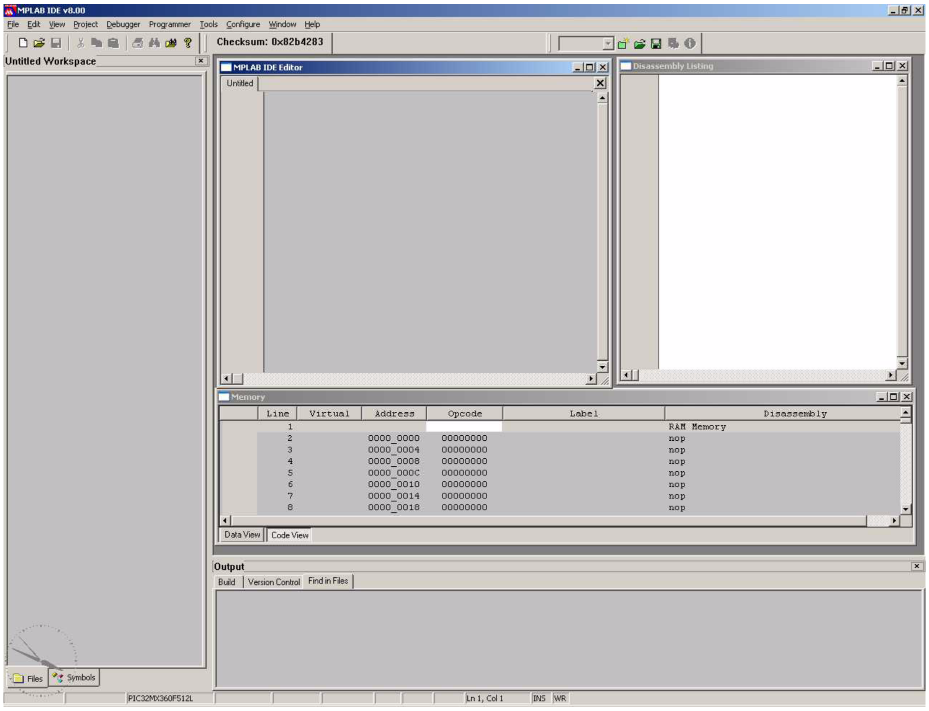
4.6.2 3rd Party Tools

Check the PIC32 home page (www.microchip.com/PIC32) to learn about the list of companies with support for PIC32.

5.3.3 Running MPLAB IDE

To start MPLAB IDE, double click on the icon installed on the desktop after installation or select *Start>Programs>Microchip>MPLAB IDE vx.xx>MPLAB IDE*. A screen will display the MPLAB IDE logo followed by the MPLAB IDE desktop.

FIGURE 5-1: MPLAB IDE START SCREEN



Line	Virtual	Address	Opcode	Label	Disassembly
1					RAM Memory
2		0000_0000	00000000		nop
3		0000_0004	00000000		nop
4		0000_0008	00000000		nop
5		0000_000C	00000000		nop
6		0000_0010	00000000		nop
7		0000_0014	00000000		nop
8		0000_0018	00000000		nop

5.4 STEP-BY-STEP GUIDE OVERVIEW

To create code that is executable by the PIC32 MCU, source files need to be part of a project. The code can then be built into executable code using selected language tools (assemblers, compilers, linkers, etc.). In MPLAB IDE, the project manager controls this process and will guide us through most of these steps.

All projects will have these basic steps:

- Select Device

The capabilities of MPLAB IDE vary according to which device is selected. Device selection should be completed before starting a project.

- Create Project

MPLAB IDE Project Wizard will be used to create a project.

- Select Language Tools

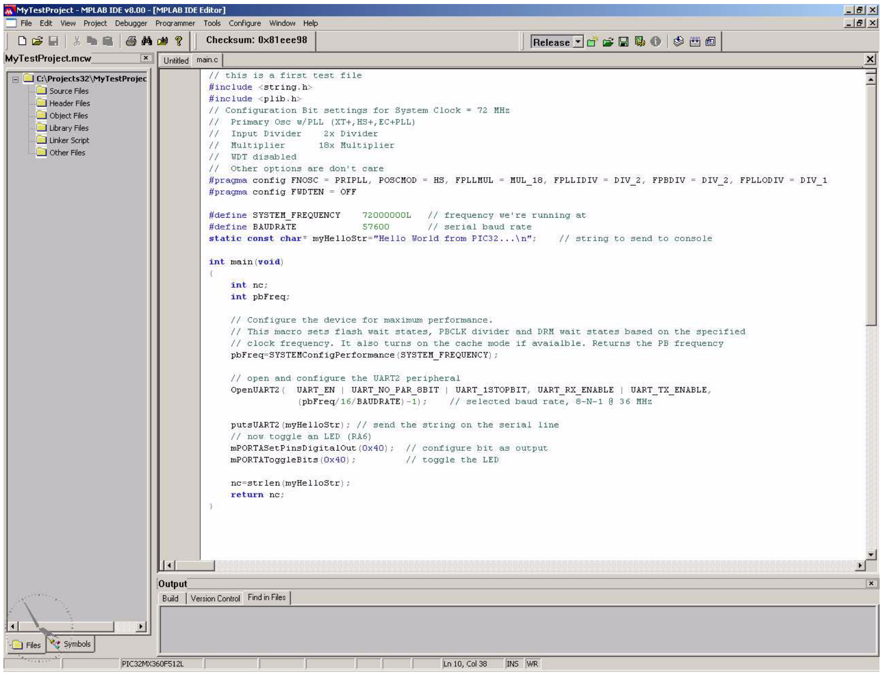
In the Project Wizard the language tools will be selected. For this tutorial, the PIC32 tools will be used. For other projects, either other Microchip or third party tools might be selected.

- Add Files in Project

We'll add a template file and a linker script to the project.

- Create Code

FIGURE 5-9: MPLAB IDE MAIN.C FILE



```c
// this is a first test file
#include <string.h>
#include <plib.h>
// Configuration Bit settings for System Clock = 72 MHz
//  Primary Osc w/PLL (XT+,HS+,EC+PLL)
//  Input Divider    2x Divider
//  Multiplier      18x Multiplier
//  WDT disabled
//  Other options are don't care
#pragma config FNOSC = PRIPLL, POSCMOD = HS, FPLLMUL = MUL_18, FPLLIDIV = DIV_2, FPBDIV = DIV_2, FPLLODIV = DIV_1
#pragma config FWDTEN = OFF

#define SYSTEM_FREQUENCY    72000000L   // frequency we're running at
#define BAUDRATE            57600       // serial baud rate
static const char* myHelloStr="Hello World from PIC32...\n";    // string to send to console

int main(void)
{
    int nc;
    int pbFreq;

    // Configure the device for maximum performance.
    // This macro sets flash wait states, PBCLK divider and DRM wait states based on the specified
    // clock frequency. It also turns on the cache mode if avaialble. Returns the PB frequency
    pbFreq=SYSTEMConfigPerformance(SYSTEM_FREQUENCY);

    // open and configure the UART2 peripheral
    OpenUART2(  UART_EN | UART_NO_PAR_8BIT | UART_1STOPBIT, UART_RX_ENABLE | UART_TX_ENABLE,
                (pbFreq/16/BAUDRATE)-1);    // selected baud rate, 8-N-1 @ 36 MHz

    putsUART2(myHelloStr); // send the string on the serial line
    // now toggle an LED (RA6)
    mPORTASetPinsDigitalOut(0x40);  // configure bit as output
    mPORTAToggleBits(0x40);         // toggle the LED

    nc=strlen(myHelloStr);
    return nc;
}
```

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.

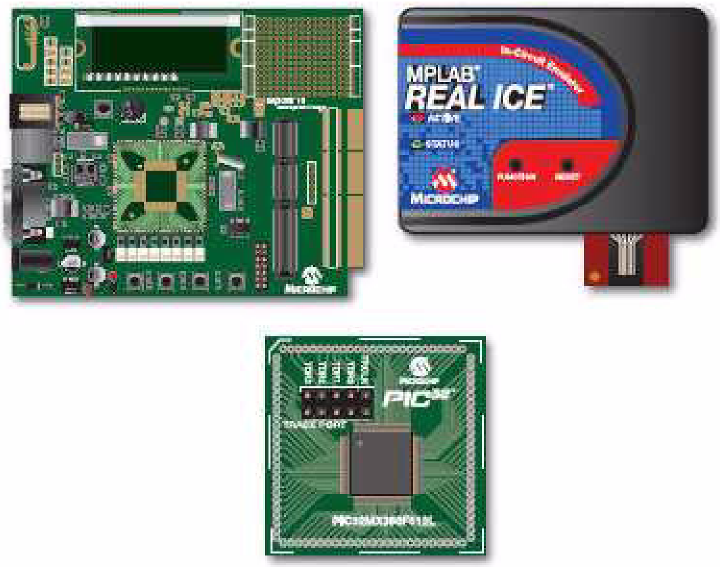
The linker script that's needed for this project is selected automatically by the MPLAB IDE. We don't need to add any other file to our project.

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.

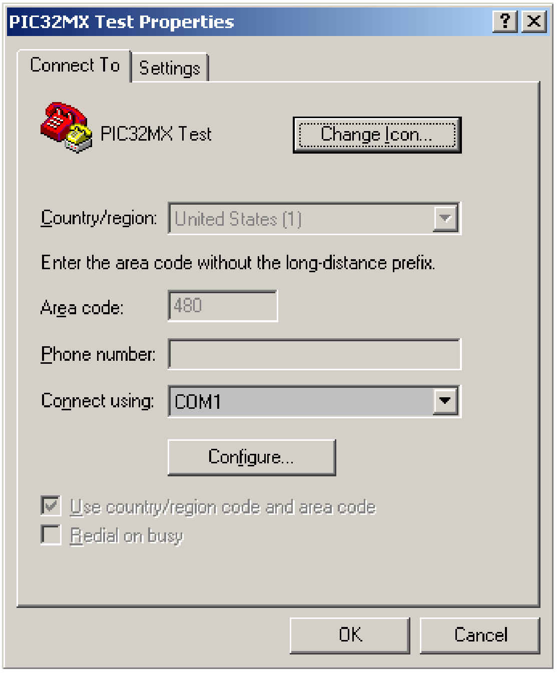
> **Note:** MPLAB IDE may need to download new firmware if this is the first time the MPLAB REAL ICE is being used with a PIC32 device. Allow it to do so.

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.

FIGURE 5-13: HYPER TERMINAL CONNECTION SCREEN



PIC32MX Test

Country/region: United States (1)

Enter the area code without the long-distance prefix.

Area code: 480

Phone number:

Connect using: COM1

☑ Use country/region code and area code

☐ Redial on busy

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.