Introduction to PIC Programming

by David Meiklejohn, Gooligum Electronics

Lesson 0: Recommended Development Environment

About PICs

"PIC" refers to an extensive family of microcontrollers manufactured by Microchip Technology Inc. – see www.microchip.com.

A microcontroller is a microprocessor which has I/O circuitry and peripherals built-in, allowing it to interface more or less directly with real-world devices such as lights, switches, sensors and motors. They simplify the design of logic and control systems, allowing complex (or simple!) behaviours to be designed into a piece of electronic or electromechanical equipment. They represent an approach which draws on both electronic design and programming skills; an intersection of what was once two disciplines, and is now called "embedded design".

Modern microcontrollers make it very easy to get started. They are very forgiving and often need little external circuitry. Among the most accessible are the PIC microcontrollers.

The range of PICs available is very broad – from tiny 6-pin 8-bit devices with just 16 bytes (!) of data memory which can perform only basic digital I/O, to 100-pin 32-bit devices with 512 kilobytes of memory and many integrated peripherals for communications, data acquisition and control.

One of the more confusing aspects of PIC programming for newcomers is that the low-end devices have entirely separate address and data buses for data and program instructions. When a PIC is described as being 8- or 16-bit, this refers to the amount of data that can processed at once – the width of the data memory (registers in Microchip terminology) and ALU (arithmetic and logic unit).

The low-end PICs, which operate on data 8-bits at a time, are divided into three architectural families:

- **Baseline (12-bit instructions)**
  These PICs are based on the original PIC architecture, going back to the 1970's and General Instrument's "Peripheral Interface Controller".
  They are quite limited, but, within their limitations (such as no interrupts), they are simple to work with – particularly in assembler.
  Modern examples include the 6-pin 10F series, the 8-pin 12F509 and the 14-pin 16F506
- **Midrange (14-bit instructions)**
  This is an extension of the baseline architecture, adding support for interrupts, more memory and on-chip timers and peripherals, including PWM (pulse width modulation) for motor control, support for serial, I^2C and SPI interfaces and LCD (liquid crystal display) controllers.
  Modern examples include the 8-pin 12F629, the 20-pin 16F690 and 40-pin 16F887
- **High-end (16-bit instructions)**
  Otherwise known as the 18F series, this architecture overcomes some of the limitations of the midrange devices, providing for more memory (up to 128k program memory and almost 4k data memory) and advanced peripherals, including USB, Ethernet and CAN (controller area network) connectivity.
  The 18F architecture is designed to support C programming, and is the only one of the 8-bit PIC

families for which Microchip offer a C compiler.
Examples include the 18-pin 18F1220, the 28-pin 18F2455 and the 80-pin 18F8520.

This can be a little confusing; the PIC18F series has 16-bit program instructions which operate on data 8 bits at a time, and is considered to be an 8-bit chip.

We won't consider the 16-bit PICs, such as the PIC24F microcontrollers or the dsPIC30 or dsPIC33 digital signal controllers, nor the new 32-bit PIC32MX devices, in this tutorial series.

Since the midrange and high-end (18F) 8-bit PICs build on the baseline architecture, the first tutorials consider only the baseline, 12-bit instruction-set PICs, such as the 12F509. This allows us to start by exploring basic concepts without the complicating influence of more advanced features, which will be introduced as appropriate.

Development Environment

For PIC development, you'll need:

- A PC, preferably running Windows XP with a spare USB port
- A PIC programmer
- Development software, including assembler and editor, and preferably a software debugger
- A prototyping environment, such as breadboard

And optionally:

- A C compiler
- A hardware debugger or emulator

PIC Programmers

There are many PIC programmers available, including some that you can build yourself.

In times gone by, PICs could only be erased by shining UV light through a window on the chip (except for parts without a window, which could only be programmed once), and programmed by placing them into a special programmer.

These days, PICs use flash memory, which can be electrically erased – thousands of times, without specialised equipment. They can be programmed without having to be taken out of the prototyping (or even production) environment, through a protocol called In-Circuit Serial Programming (ICSP). But instead of worrying about designing your circuit to accommodate the ICSP protocol, it's easier to remove the PIC from your circuit and place it into a prototyping board or programming adapter connected to an ICSP programmer. A programming adapter is simply a minimal circuit which allows a PIC to be programmed by an ICSP programmer. It's a really good idea to buy an ICSP programmer; you can use it as a simple, stand-alone programmer (with an adapter) to start with, while keeping the option of later using it with your own circuitry when you're ready for that.

An excellent PIC programmer to start with is Microchip's PICkit 2, shown on the right:

Although there are cheaper equivalents available, the PICkit 2 is not expensive, available for around US$35. And being a Microchip product, you can be sure that it will work with Microchip's tools, development environment, and (importantly!) PICs.

Development Software

Every PIC developer should have a copy of Microchip's MPLAB integrated development environment (IDE) – even if you primarily use a third-party *tool chain* (a set of development tools that work together).

It includes Microchip's assembler (MPASM), an editor, and a fully-featured software simulator, which allows you to debug your application before committing it to the chip. Not long ago, a development environment as sophisticated as this would have cost thousands. But MPLAB is free, including support from Microchip, so there is no reason not to have it. Download it from www.microchip.com.

MPLAB directly supports the PICkit 2 as a programmer for most of the modern baseline and midrange PICs, although some of that support is considered "beta" (i.e. under development, not fully tested and supported), as of version 8.10. You will also need the software that comes with the PICkit 2. It comes on CD with the PICkit 2, but you should always download the latest version from www.microchip.com, as each new release adds support for more PICs.

MPLAB, from version 7.41, includes a free copy of CCS's C compiler for the baseline PICs. It's nice to have, and it is used in the Baseline PIC C Programming tutorial series, but in practice most people wouldn't use a C compiler on the baseline PICs; resources are so tight that to make the most of them, you need to use assembly. That explains why CCS are able to give away their baseline compiler; few people would have bought it, but it provides a lead into the rest of their range of PIC C compilers.

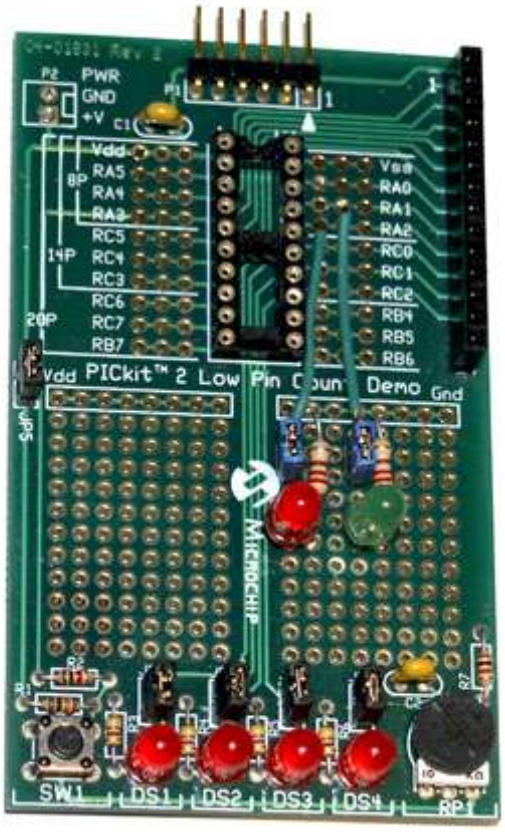
MPLAB (as of version 8.10) also comes with a limited version of HI-TECH's PIC C compiler – "PICC-Lite". It supports all of the baseline PICs, as well as a small number of midrange PICs – but for the larger devices, such as the 16F690, it restricts the amount of memory that can be used. Nevertheless it is very useful for exploring C programming on the baseline and midrange PICs. It comes with a very nice IDE – "HI-TIDE", but also, like the CCS compilers, integrates into MPLAB, meaning that it is possible to use MPLAB as a complete development environment for assembler as well as C programming.

Microchip don't sell a for the baseline or midrange PICs (which aren't really well suited to C programming), but they do sell and support a C compiler for the 18F series, called C18. At around US$500 it's not cheap, but a student edition, which has some optimisations disabled but is otherwise fully functional, is available as a free download. Although called a "student" edition, it can be used without restrictions by anyone. And like the other C compilers, C18 fully integrates into MPLAB.

Prototyping

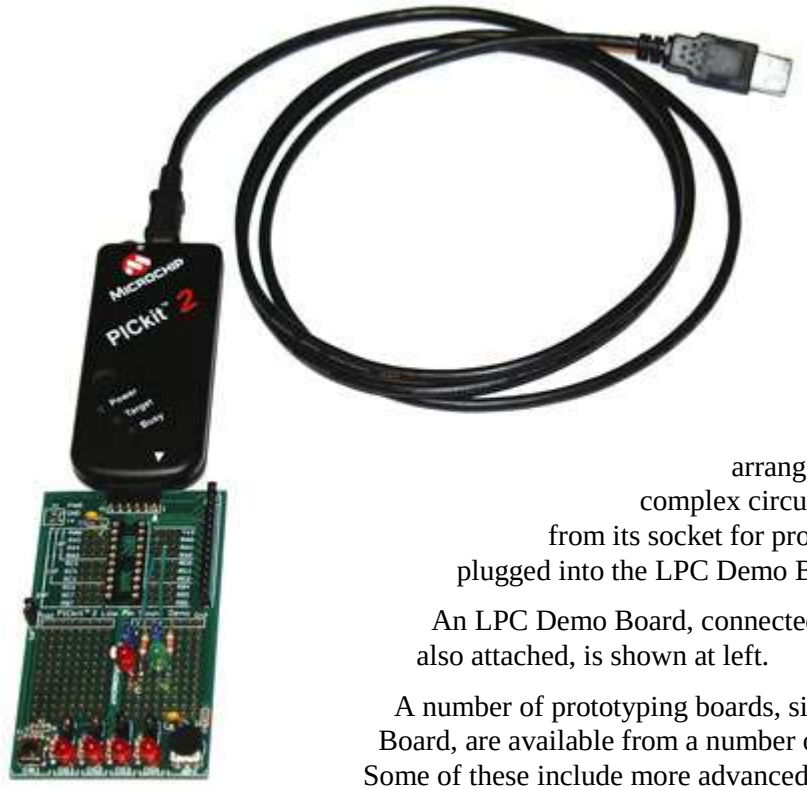
If you use an ICSP programmer, then you'll need a way to connect your PIC to it, for programming. You also need to be able to test your PIC in a real circuit, before building the final design.

An excellent solution, satisfying both these purposes, is Microchip's "Low Pin Count (LPC) Demo Board", designed to be used with the PICkit 2. It is available for around US$24, including a PIC16F690. Or you can buy a "PICkit 2 Starter Kit" bundle, including a PICkit 2 programmer, LPC Demo Board and PIC16F690, for around US$50. That's excellent value; given that the MPLAB software is free, that's everything you need to get started, including a PIC chip, for only US$50!



A modified LPC Demo Board is pictured on the right. You'll see that it includes, at the bottom of the board, four LEDs, a pushbutton switch and a trimpot (variable

resistor). Above this, a prototyping area is provided, and on the pictured board, another two LEDs have been added. This is the configuration used for the first tutorials. The board doesn't come with jumpers installed, but it is easy to add them, as shown, so that the LEDs and trimpot can be selectively disconnected from the PIC (which you insert into the IC socket).



The LPC Demo Board supports all of the modern (flash memory based) 8-, 14- and 20-pin baseline and midrange PICs. Most of the I/O lines are brought out to the 14-pin socket on the side of the board, allowing the board to be connected to another circuit. For example, the prototype circuit can be constructed on a breadboard and the power and signal lines connected back to the LPC Demo Board using hook-up wire. This arrangement allows you to develop a complex circuit without needed to remove the PIC from its socket for programming; the PICkit 2 can remain plugged into the LPC Demo Board during development.

An LPC Demo Board, connected to a PICkit 2, with its USB cable also attached, is shown at left.

A number of prototyping boards, similar in concept to the LPC Demo Board, are available from a number of sources, including Microchip. Some of these include more advanced peripherals, such as LCD displays, while others are intended to be an introduction to "mechatronics" (microcontroller-controlled robotics), and include motors, gears, etc. And of course, as you progress, you can build your own custom prototyping boards, perhaps supporting larger PICs.

Recommendation

To make a start in PIC development, it's difficult to do better than the low-cost combination of:

- PICkit 2 programmer
- Low Pin Count Demo Board
- MPLAB integrated development environment

These are available as a "PICkit 2 Starter Kit" bundle for only US$50 (as of July 2008).

These tutorials assume that you are using the "PICkit 2 Starter Kit" bundle, although most of the lesson content is of course applicable to other tool chains.

Other than a PC, the only other thing you need is a PIC! Although the PICkit 2 Starter Kit includes a PIC16F690, we will initially put that aside and start with the (simpler, less confusing) baseline architecture.

And for a thorough understanding of the PIC hardware, it is best to start by learning assembly language first, by working through the Baseline PIC Assembler tutorial series, and then perhaps looking at C programming on the baseline PICs, before moving on to the midrange architecture.