April, 2006

Wakefield Robotics Ltd.

Robotics 101 Online Education

Robotics 101 – An Introduction to PIC Microcontrollers

Chapter 1 - An Introduction to PIC Microcontrollers

1.1 Objectives

To provide the student with an understanding about what microcontrollers are and how they are used.

To provide the student with sufficient background information that he can select a microcontroller suitable for his desired application.

1.2 Introduction

What is a microcontroller?

A microcontroller is a tiny digital computer. All computers, from microcontrollers up to large mainframe computers, have many things in common:

All computers process and store information in a binary format. This means that all the computer's processes used just two electronic states: a high or a low voltage. For instance, a high voltage could represent the number one and a low-voltage could represent the number zero.

All computers have a CPU (central processing unit) that executes the programs. Whenever you are running a program on your desktop computer, the CPU is processing the information according to the program, such as a word processor or web client. A microcontroller can control the temperature in a room according to the program entered into its program memory.

The CPU loads the program from storage. On your desktop computer, the program is loaded from the hard disk or CD drive. On a microcontroller, the program is loaded from the Flash or ROM memory.

A computer has RAM (random-access memory) which it can use as a scratchpad. This is where variable contents and intermediate calculations are stored while the CPU is running its program. Dedicated portions of RAM are used as registers, to control the operation of the microcontroller.

Computers have input and output devices so they can communicate with the real world. On your desktop computer, the keyboard and mouse are input devices and the monitor, printer and speaker are output devices. Microcontrollers use I/O (Input/Output) pins to communicate with the real world.

The desktop computer you are using is a general purpose computer, that can run any of thousands of programs. Desktop computers also have expansion cards which allow a computer to be modified for particular purposes.

Microcontrollers are special purpose computers. They are not as versatile as desktop computers. However, microcontrollers have built-in features, which reduces the number of external electronic components required to make a usable application.

Microcontrollers are usually embedded inside some other device, so that they can control its actions. This is why another name for a microcontroller is an embedded controller.

Microcontrollers are typically dedicated to one task and run one specific program. The program is stored in FLASH memory or ROM (read-only memory) and generally does not change.

Microcontrollers are often low-power devices. A desktop computer is almost always plugged into a wall outlet and it might consume as much as 300 watts of electricity. A battery-operated microcontroller might consume as little as 50 milliwatts.

What is a PIC microcontroller?

PIC is a family of RISC (Reduced Instruction Set Computer) microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division.

Microchip Technology does not use PIC as an acronym; in fact the brand name is PICmicro. It is generally regarded that PIC stands for Peripheral Interface Controller, although General Instruments' original acronym for the PIC1650 was Programmable Intelligent Computer. The original PIC, was built to be used with General Instrument's new 16-bit CPU, the CP1600. While generally a good CPU, the CP1600 had poor I/O performance, and the 8-bit PIC was developed in 1975 to improve performance of the overall system by offloading I/O tasks from the CPU. The PIC used simple program stored in ROM to perform its tasks, and although the term wasn't used at the time, it is a RISC design that runs one instruction per instruction cycle (4 oscillator cycles).

In 1985, General Instrument sold off their microelectronics division, and the new ownership cancelled most of the product line, which by this time was mostly out-of-date. The PIC, however, was upgraded with EPROM (see *Section 1.5*) to produce a programmable channel controller. Today, a huge variety of PICs are available with a variety of on-board peripherals, program memory, RAM, and I/O pins.

The old PROM and EPROM PICs are now gradually being replaced by chips with Flash memory. Likewise the original 12-bit instruction set of the PIC1650 and its direct descendants has been superseded by 14-bit and 16-bit instruction sets.

Microchip still sells OTP (one-time-programmable, or PROM) and UV-erasable (EPROM) versions of most of its PICs for legacy support, or volume orders. It should be noted that the Microchip website lists PIC microcontrollers that are not electrically erasable as OTP despite the fact that UV erasable windowed versions of these chips can be ordered.

How do we program a microcontroller?

PIC microcontrollers use a RISC assembly-language instruction set. This means that there is only about 35 instructions for the low-end PICs, and up to 72 instructions for the high-end PICs. The assembler language instruction set includes instructions to perform a variety of operations on:

- the accumulator and a constant, or
- the accumulator and a memory location, or
- for conditionally executing code, or
- jumping/calling other parts of the program and returning from them, or
- specific hardware features like interrupts and low-power sleep mode.

Third parties (companies other than Microchip) make C and BASIC language compilers for the PIC microcontrollers. Microchip also sells compilers for the PIC microcontrollers Open-source compilers have been developed for C, Pascal, JAL, and for the Forth programming language. We will be using the MikroBasic compiler and IDE (Integrated Development Environment). This free download is adequate for the needs of this course, and covers all the PIC microcontroller range. It is also easily upgraded to a full compiler package.

Devices called programmers are traditionally used to get the compiled program code (program converted to a form that the microcontroller can understand) into the target PIC microcontroller. Most PIC microcontrollers that Microchip sells nowadays have ICSP (In Circuit Serial Programming) and/or LVP (Low Voltage Programming) capabilities, allowing the PIC to be programmed while it is sitting in the final target circuit. We will be downloading the compiled program to our microcontroller using a programmer, either a David Tait style serial programmer (my design for this programmer will be available soon) or a USB programmer.

Many of the higher-end Flash-based PICs can also write to their own program memory. Demo boards are available with a small bootloader program that can be used to load user programs over an interface such as RS232 or USB.

1.3 Applications

Microcontrollers are hidden inside a surprising number of products these days. If your microwave oven has an LED or LCD screen and a keypad, it contains a microcontroller.

All modern automobiles contain at least one microcontroller, and can have as many as six or seven: The engine is controlled by a microcontroller, as are the anti-lock brakes, the cruise control and so on.

Any device that has a remote control almost certainly contains a microcontroller. Televisions, VCRs and high-end stereo systems all fall into this category. Good quality SLR and digital cameras, cell phones, camcorders, answering machines, laser printers, telephones (the ones with caller ID, 20-number memory, etc.), pagers, and feature-laden refrigerators, dishwashers, washers and dryers (the ones with displays and keypads), etc.

Basically, any product or device that interacts with its user has a microcontroller inside.

1.4 PIC Capabilities

Current PIC microcontrollers offer a wide range of built-in features, such as:

- 8/16 bit Modified Harvard Architecture CPU cores
- Flash, EEPROM and RAM Memory options in 256 byte to 256 kilobyte array sizes
- I/O Ports (0 to 5.5 volts Typical)
- 8/16 Bit Timers
- Nanowatt Technology for Power Moding
- Synchronous/Asynchronous Serial Peripherals USART, AUSART, EUSARTs
- Analog-to-digital converters, 10/12 bit
- Voltage Comparators
- Capture/Compare/PWM modules
- LCD Drivers
- MSSP Peripheral for I2C, SPI, and I2S Communications
- Internal EEPROM Memory - up to 1M erase/write cycles durability
- Motor Control Peripherals
- USB interfacing support
- Ethernet controller support
- CAN controller support
- LIN controller support
- IRDA controller support
- Integrated analog RF front ends (PIC16F639, and rfPIC)
- KEELOQ Rolling code encryption peripheral (encode/decode)

I am sorry that I cannot stop and discuss all these features at this time. Many of the features will be covered later as we proceed through the courses.

1.5 Memory

At this time, I would like to discuss the different type of memory available in microcontrollers.

Random Access Memory

RAM is short for Random Access Memory. This type of memory is used as a scratchpad by the microcontroller's CPU. It is also used to store program variables and registers. Program variables can have different values as the program is running. Registers are used to control the microcontrollers hardware. Both program variables and registers will be discussed at a later time.

RAM is volatile. This means that it loses the data when the power is removed.

All PIC microcontrollers have built-in RAM.

Read Only Memory

Read-Only Memory (ROM) is used to permanently store data or programs, so that they can be retrieved at a later time. Data is usually put into the ROM at the factory. At this time only OTP (One-Time Programmable) microcontrollers used this type of memory.

Programmable Read-Only Memory

Programmable Read-Only Memory (PROM) is very similar to ROM. However, the data can be downloaded to the PROM electronically by the end user. Programmable read-only memory cannot be erased and reused. Therefore, the data or computer program must be thoroughly debugged and tested.

Erasable Programmable Read-Only Memory

Erasable Programmable Read-Only Memory (EPROM) is very similar to PROM. The data or program can be stored permanently, and it can be downloaded by the end user. However, this memory can be erased. When an ultraviolet (UV) light shines through a small window on the microcontroller package, the contents are erased. *Warning, if the window is left uncovered, the contents may be accidentally erased by sunlight or eventually by normal room light.*

Electronically-Erasable Programmable Read-Only Memory

Electronically-Erasable Programmable Read-Only Memory (EEPROM) is also very similar to PROM. It can also be erased. In this case, the data can be erased electronically. This eliminates the need for a UV light source and reduces the chance that the memory is accidentally erased. The memory is erased one section at a time.

In the PIC microcontrollers, EEPROM is used by the program, to store and retrieve data. Therefore, a microcontroller can act as a data acquisition system. It can read the weather information and store it in the EEPROM. Later, this information can be retrieved and processed or uploaded.

Flash

Flash memory is very similar to EEPROM. It is also erased electronically. However all the flash memory is erased simultaneously. Flash memory is also quicker to read and write than EEPROM.

In the newer PIC microcontrollers, the main program is stored in the Flash memory. Therefore, the larger the flash memory, the larger the program can be.

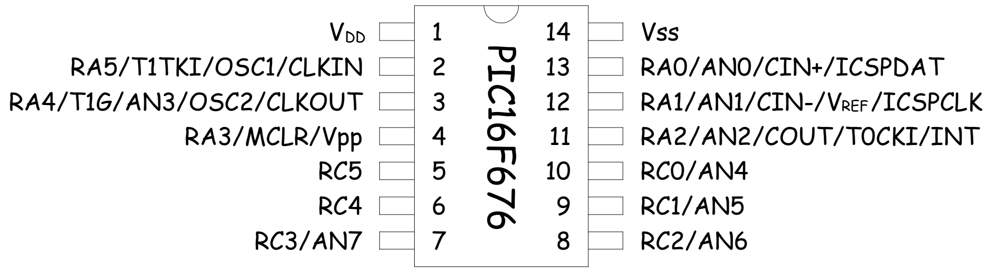
1.6 Power Pins, I/O Pins and Registers

PIC microcontrollers can have 6 to 80 interface pins, with 100 interface pins available on models presently under development.

Care must be taken when connecting to the pins on the microcontroller. If the pins are not connected correctly, the microcontroller or the pins may be damaged.

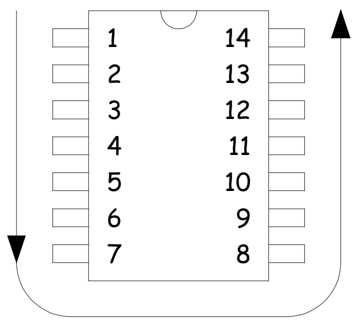
Consider *Figure 1.1,* which shows a plan (top) view of the PIC16F676 microcontroller that we will use in this course. Notice that there is a series of numbers on the inside of the microcontroller. This represents the pin number on this, a dual-inline pin (DIP), package. In fact, this numbering system is standard on all DIP packages for electronic components.

Figure 1.1 PIC16F676 Pin Layout



If you orient the DIP package with the end with the semi-circle in the upper most position, then Pin 1 is in the upper left corner. Count down the left side incrementing the pin number and then up the right side in a "U" fashion. See *Figure 1.2*. The pin with the highest number should be opposite Pin 1.

Figure 1.2 Pin Numbering on a DIP Package



Since the PIC16F676 microcontroller has 14 pins, the last pin will be Pin 14. *Note that Pin 14 is opposite Pin 1.*

Let's look again at *Figure 1.1*. Notice all the abbreviations on the outside of the diagram. Some pins have several abbreviations beside them. This is because these pins can have multiple uses. For instance, Pin 4 can be an I/O pin, a MCLR (master clear latch reset), or V_{PP} - a pin used during programming.

At this time, we will discuss power and ground pins, I/O pins, and the MCLR pin.

Power and Ground Pins

The PIC microcontrollers operate well from a 5VDC power source. However, the PIC16F676 microcontroller, like most PIC microcontrollers, can operate on a DC (direct current) voltage ranging from 2.0 to 5.5 volts. This means that PIC microcontrollers can be used in battery-operated applications.

To power the microcontroller, the positive terminal of your 5VDC power supply must be connected to the V_{DD} pin on the microcontroller. On the PIC16F676 microcontroller, this means the +5VDC is connected to Pin 1. To complete the power circuit, the negative terminal of the power supply is connected to V_{SS}, or Pin 14 on the PIC16F676.

If you are using long leads from the power supply to your circuit, or if your circuit experiences large loads like motors, solenoids, etc., then it is a good idea to place a 0.1

µF or larger capacitor between the V_{DD} pin and V_{SS} pin (or ground). This will help prevent voltage fluctuations that may adversely affect the microcontroller's operation.

Input / Output Pins

The majority of the pins on a PIC microcontroller are input/output (I/O) pins. The pins are called I/O pins, because they can be configured as either inputs or outputs. I will tell you how to configure these pins a little later on.

The I/O pins are grouped together in ports. A port can contain up to eight I/O pins. The PIC16F676 microcontroller has 12 I/O pins. These I/O pins are separated into two ports, the A port and the C port. Each port has 6 pins. Port A has pins RA0, RA1, RA2, RA3, RA4, and RA5. Similarly, Port C has pins RC0, RC1, RC2, RC3, RC4, and RC4. We will discuss why the I/O pins are designated starting with zero (0) later, when we discuss the binary numbering system.

When a pin is set as an output, it can provide a voltage equal to V_{DD} (supply voltage) with a current of 25 milli-Amperes (mA) or a voltage equal to the V_{SS} with a current of 25 milli-Amperes (mA). The high voltage output is referred to as sourcing, and the low voltage output is referred to as sinking. *Note: the current provided by the I/O pins is very low. Additional electronic devices are required to boost the power from the I/O pins.*

Registers

Since the I/O pins are controlled by the registers, I will introduce the registers now. Registers are dedicated spaces in the RAM. They can be accessed by using specific names. For instance, setting the I/O pins as inputs or outputs, is controlled by the TRISA and TRISC registers.

When using binary numbers to indicate the state of each pin, a 1 represents an input and a 0 represents an output.

For example, let's consider the following line from our MikroBasic program:

TRISA = %00110011

TRISA means that we are setting Port A pins as inputs or outputs. The percent sign (%), shows that the following number is in the binary format. This makes it easier to tell whether a pin is an input or output. The pins are read from right to left in the following fashion:

Port A Pin 0 is set to an input, 1 => Input (1 looks like an I for input)
Port A Pin 1 is set to an input
Port A Pin 2 is set to an output, 0 => Output (0 looks like an O for output)
Port A Pin 3 is set to an output
Port A Pin 4 is set to an input
Port A Pin 5 is set to an input
Port A Pins 6 and 7 do not exist. However, a 1 or 0 must be placed in the binary number so that there are eight (8) digits.

If an I/O pin is not connected to anything, then it should be set as an output. A disconnected input pin is said to be floating. Floating pins can build up high voltages and they can be permanently damaged.

If the pin is set as an output, then the pin can provide a high or low voltage. A output pin provides a high voltage when its register is set to a one. It provides a low voltage when its register is set to a zero.

For instance, Port A Pin 0 can be set to a high voltage by using the following command:

PORTA.0 = 1

Similarly it can be set to a low voltage:

PORTA.0 = 0

Or all the pins can be set high or low using the following style of command:

PORTA = %00001010

Where: Port A Pin 0 was set as an input. Setting it low does not affect the input pins readings.
Port A Pin 1 was also set as an input.
Port A Pin 2 outputs a low voltage.
Port A Pin 3 outputs a high voltage.
Port A Pins 4 and 5 were set as inputs.
Pins 6 and 7 do not exist, so these digits are necessary to provide 8 digits, but ignored as pin settings.

If you are using the smaller PIC12F629 and PIC12F675 microcontrollers, which have only six (6) I/O pins, the port is called GPIO and the input or output is set by TRISIO.

For example:

```
TRISIO = %00010000        ‘ All I/O pins set as outputs, except Pin 4
GPIO = %00111111          ‘ The outputs of all pins set high, except Pin 4 an input
```

Note: The apostrophe indicates that a comment follows. Therefore, the compiler does not process the comments.

MCLR

MCLR is an abbreviation for Master Clear Latch Reset.

When the MCLR is enabled, Pin 4 acts as a microcontroller control switch – like the ignition switch in an automobile. If the input to Pin 4 is high, the microcontroller runs. If the input is low, the microcontroller stops and resets its program counter to the beginning of the program.

If you don’t need the MCLR, it can be disabled and Pin 4 can now be used as an input pin (Port A Pin 3). Unfortunately, this pin cannot be used as an output pin.

1.7 Clocks and Resonators

A clock is required for the proper operation of a microcontroller. Think of a microcontroller clock as the drum in an orchestra. If the beat is too fast, the instruments cannot keep up and the music fails. The same is true with a microcontroller, if the clock speed is too fast, the microcontroller cannot keep up. If the clock speed is too slow, the microcontroller may not be able to keep up with the changes in the real world around it.

Many PIC microcontrollers have a built-in 4 MHz oscillator (clock). This oscillator is great for most applications. However, if you need to run the microcontroller at a faster speed, you can add an external crystal or ceramic resonator up to 20 MHz. This enables the microcontroller up to run 5 times faster.

Note: 1 MHz is 1 million cycles per second.

There is one (1) instruction cycle for every four (4) oscillator cycles. Therefore, if the microcontroller is running on the internal 4 MHz oscillator, it can perform 1 million instruction cycles per second. For most real-world applications, the microcontroller will have no problems keeping up!

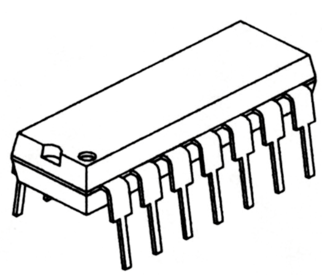
There are other ways of providing clock signals to microcontrollers. Two or more microcontrollers can share a clock signal and therefore their operations will be synchronized. For slower speeds, a simple resistor-capacitor can be used. We will discuss alternate oscillator configurations in the intermediate course on microcontrollers.

1.8 Packages and Ratings

Packages

We will be using the PIC16F676 microcontroller in a 14-pin DIP package. This package conveniently fits in the ZIF (Zero Insert Force) socket on the programmer and in the electronic breadboard on which we will assemble our circuits. All the adjacent pins are at 0.1 inch (2.54 millimeters) apart. Pins on opposite rows are 0.3 inches (7.63 mm) apart. These microcontrollers can be placed on the top of a PCB (Printed Circuit Board) with the pins extending through holes in the board. The pins are soldered to conductive traces on the bottom of the board. These packages can also be placed in tight fitting sockets, so they can be removed at a later time.

Figure 1.3 14-Pin DIP Package



There are two types of surface mount packages available. The SOIC (Small Outline Integrated Circuit) and the TSSOP (Thin Shrink Small Outline Package). This type of package is soldered to traces on the top side of the PCB. This allows for automated machinery to populate the circuit board.

Figure 1.4 SOIC Package

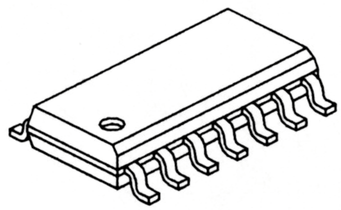
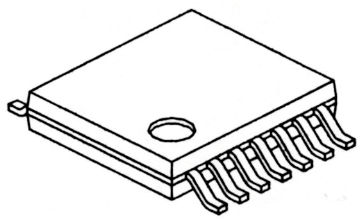


Figure 1.5 TSSOP Package



Note: The small dot on the packages indicate Pin 0.

Temperature Ratings

Most PIC microcontrollers come in two temperature ratings, Industrial and Extended.

The lesser expensive rating is the Industrial. This rating states that the microcontroller will be able to operate if the ambient temperature is between -40°C and 85°C (-40°F and 185°F).

The extended temperature rating allows the microcontroller to operate between -40°C and 125°C (-40°F and 257°F). This means that the microcontroller will still be able to function if the temperature is hot enough to boil water!

For this course, we will be using the PIC16F676-I/P microcontroller. This means that it is a PIC16F676 microcontroller in DIP package with an Industrial temperature rating.

1.9 Advanced PIC Microcontrollers

Wireless PICs

The rfPIC microcontroller integrates the power of the PIC microcontroller with UHF (Ultra-High Frequency) wireless communication capabilities for low power RF (Radio Frequency) applications. The devices offer small package outline and low external component count to fit the most space-constrained applications.

dsPICs (Digital Signal PICs)

dsPICs are Microchip's newest microcontroller family, which entered mass production in late 2004. They are designed as a PIC microcontroller with DSP (digital signal processing) capabilities. These are Microchip's first inherent 16-bit (data) microcontrollers. They build on the PIC microcontroller's existing strengths by offering hardware MAC (multiply-accumulate), barrel shifting, bit reversal, (16x16)-bit multiplication and other digital signal processing operations. This family or microcontrollers will be discussed by themselves in a later course.