# Assembler Code

The Programming Editor software is a very powerful software tool. It is designed to program PICAXE chips by a direct serial cable link, or to program blank microcontrollers with assembler code using a PIC Programmer. This datasheet explains the 'process' of how to program microcontrollers with assembler code. It does not tell you how assembler code actually works - there are many books available on this subject!

## Hardware / Software

To program microcontrollers in assembler code you require a Serial PIC Programmer (part BAS800). You cannot download assembler programs to a PICAXE chip using the PICAXE direct cable download method.

After starting the Programming Editor software click
View>Options>Mode>PICMicro - Assembler (then click OK)
to ensure the software is in the correct mode.

## Assembler Code Programs

There are two ways to generate an assembler code program:

1. Write a program in BASIC and then automatically convert it to assembler
2. Write the program directly in assembler.

### 1) Write a program in BASIC and automatically convert it to assembler

The traditional (and industrial) way of programming microcontrollers is to write your program in assembler code. This is a much more complex language than BASIC, but also allows far more complicated and powerful programs to be developed.

For those new to assembler code, the Programming Editor software allows BASIC programs to be automatically converted into assembler code. This allows students to learn assembler code by 'disassembly' of the generated program. The conversion process has specifically been designed to be 'sequential' so that each BASIC command becomes a recognisable 'block' of assembler code to make this process much easier.

Note that the BASIC to Assembler conversion function will also not operate unless the Revolution Serial PIC programmer is attached to the computer.

**Detailed Instructions:**

1) Click File>New to start a new program.
2) Type in your BASIC program.
3) Save the BASIC program. IMPORTANT - you must save the program as a 'BASIC' type file (ie with the suffix .bas)
4) Click PICMicro>Convert BASIC to Assembler
5) Select the target microcontroller you wish to use from the options given and the click OK. This will generate a new assembler code file ending .asm. At this point you can study the assembler code file and compare it with the BASIC - each line of BASIC is included as a 'comment' in the translated assembler code file.

6) The next stage of the process is to convert the assembler code into 'machine code' (numbers) for burning into the microcontroller. To do this click PICMicro>Assemble.

Presuming no assembler code errors, the software will then ask you whether you wish to program the PICmicro. Ensure the blank microcontroller is correctly inserted in the programmer and then click 'OK'. A progressbar will appear and the microcontroller will be programmed.

If there are errors in the program, an error window will appear. Double click on the error description to jump to the correct program line in the original assembler code program. Errors will only normally occur in converted BASIC programs if you try to use a BASIC command that is not supported by the conversion process.

**2. Write the program directly in assembler.**

The traditional (and industrial) way of programming microcontrollers is to write your program in assembler code. This is a much more complex language than BASIC, but also allows far more complicated and powerful programs to be developed.

Assembler code files are complex and many complete books are available to teach you assembler code programming. However the Programming Editor software includes common 'template' files that make setting up the configuration bits etc. more straight forward.

**Detailed Instructions:**

1) Click File>New Template to open a new template program for the PICmicro.
2) Type in your assembler code program in the main section of the template.
3) Save the assembler code program. IMPORTANT - you must save the program as a assembler type file (ie with the suffix .asm)
4) The next stage of the process is to convert the assembler code into 'machine code' .hex file (numbers) for burning into the microcontroller. To do this click PICMicro>Assemble.

Presuming no assembler code errors, the software will then ask you whether you wish to program the PICmicro. Ensure the blank microcontroller is correctly inserted in the programmer and then click 'OK'. A progressbar will appear and the microcontroller will be programmed.

If there are errors in the program, an error window will appear. Double click on the error description to jump to the correct program line in the original assembler code program and then correct the mistake.

Note that you can also open the main programmer interface at any time by clicking PICmicro>Program. This is for advanced users familiar with working with machine code .hex files.