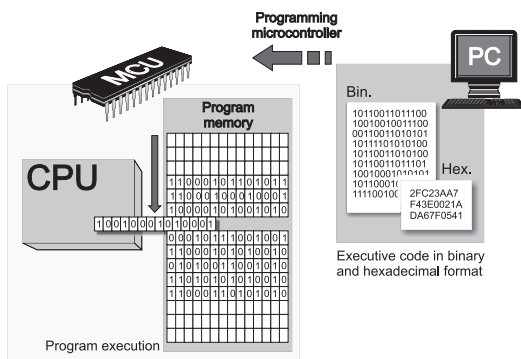


You certainly know that it is not enough just to connect the microcontroller to other components and turn the power supply on to make it work, don't you? There is something else that must be done. The microcontroller needs to be programmed to be capable of performing anything useful. If you think that it is complicated, then you are mistaken. The whole procedure is very simple. Just read the following text and you will change your mind.

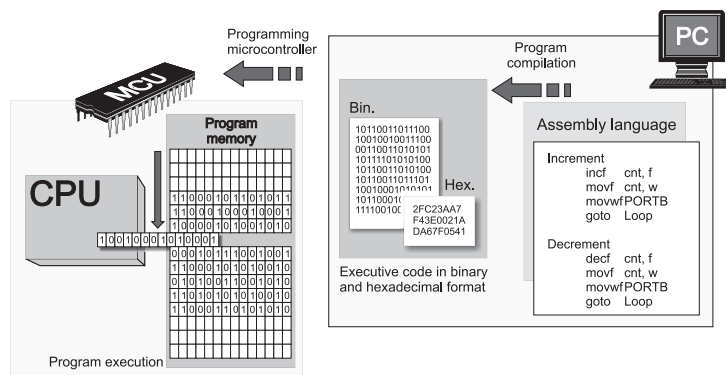
2.1 PROGRAMMING LANGUAGES

The microcontroller executes the program loaded in its *Flash* memory. This is the so called executable code comprised of seemingly meaningless sequence of zeros and ones. It is organized in 12-, 14- or 16-bit wide words, depending on the microcontroller's architecture. Every word is considered by the CPU as a command being executed during the operation of the microcontroller. For practical reasons,



as it is much easier for us to deal with hexadecimal number system, the executable code is often represented as a sequence of hexadecimal numbers called a *Hex code*. It used to be written by the programmer. All instructions that the microcontroller can recognize are together called the Instruction set. As for PIC microcontrollers the programming words of which are comprised of 14 bits, the instruction set has 35 different instructions in total.

As the process of writing executable code was endlessly tiring, the first “higher” programming language called assembly language was created. The truth is that it made the process of programming more complicated, but on the other hand the process of writing program stopped being a nightmare. Instructions in assembly language are represented in the form of meaningful abbreviations, and the process of their compiling into executable code is left over to a special program on a PC called compiler.



The main advantage of this programming language is its simplicity, i.e. each program instruction corresponds to one memory location in the microcontroller. It enables a complete control of what is going on within the chip, thus making this language commonly used today.