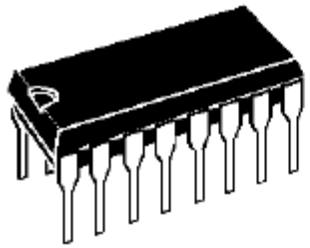
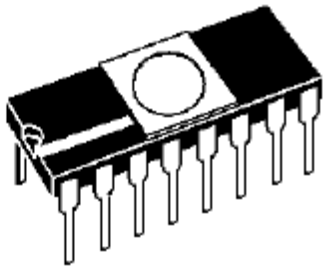
MICROCONTROLLER PRACTICE

BY Mr Serge Paulin TCHASSO

Definition of a microcontroller

- A microcontroller (MCU) could be very summarily defined like a system of complete control, dedicated to a particular application which is equipped with logic functions and most of the other elements necessary to his operation. The **microcontroller** is therefore as a microcomputer built in a chip.





The microcontroller is is programmable integrated circuit. It behaviour depends on a program that must be written and uploaded into it memory

Elements of the internal structure of a microcontroller

- The memory unit
- The central processing unit
- The Input-Output unit
- Buses
- The Timer unit
- The Watchdog
- The Analogue To Digital Converter unit
- The Serial Communication unit
- The USB unit

The Memory unit

Memory is part of the microcontroller whose function is to store data. It is divided into 03 three parts:

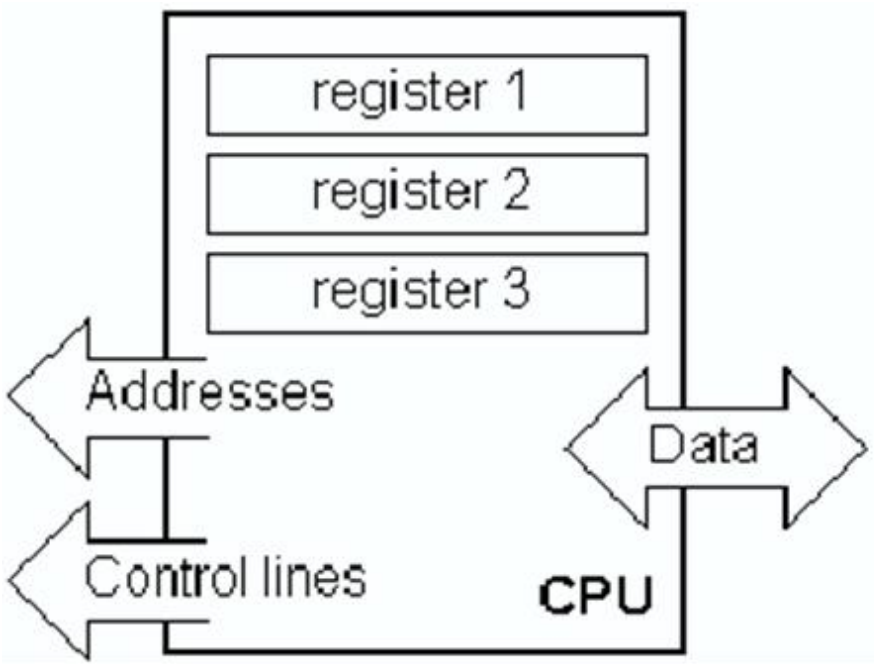
- **The program memory** contains the program, therefore the instructions to achieve, and possibly data needed by the program (for example variables). The program memory can be a **ROM**. In this case the program is sent to the microcontroller **once**. The ROM is sometimes replaced by an **EEPROM or flash memory**, which can be electrically written and erased **several times**.

- **The RAM** (*Random Access Memory*): makes it possible to store data during the execution of the program.

- The **EEPROM** (*Programmable Electrically Erasable Read Only Memory*): this memory can be erased and reprogrammed, like the flash. It is used to store data, in case for example of power supply failure.

The Central Processing Unit

- **CPU** (*Central Processing Unit*) it is the "brain" of the system. It is The internal microprocessor that reads and executes the instructions of the program stored in the program memory. Its memory locations are called registers.



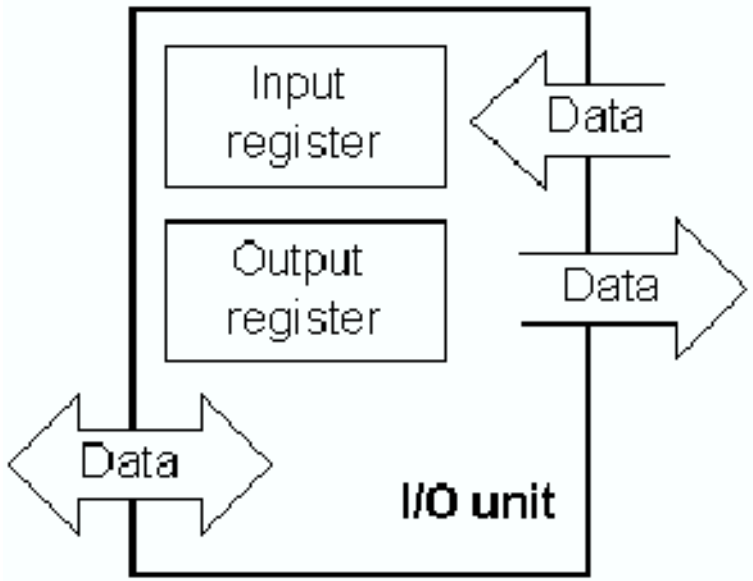
Example of simplified central processing unit with three registers

The Input-output unit

This unit enables the microcontroller to communicate with the outside world through locations called PORTS. There are three types of ports:

- **Input ports**
- **Output ports**
- **Bidirectional ports**

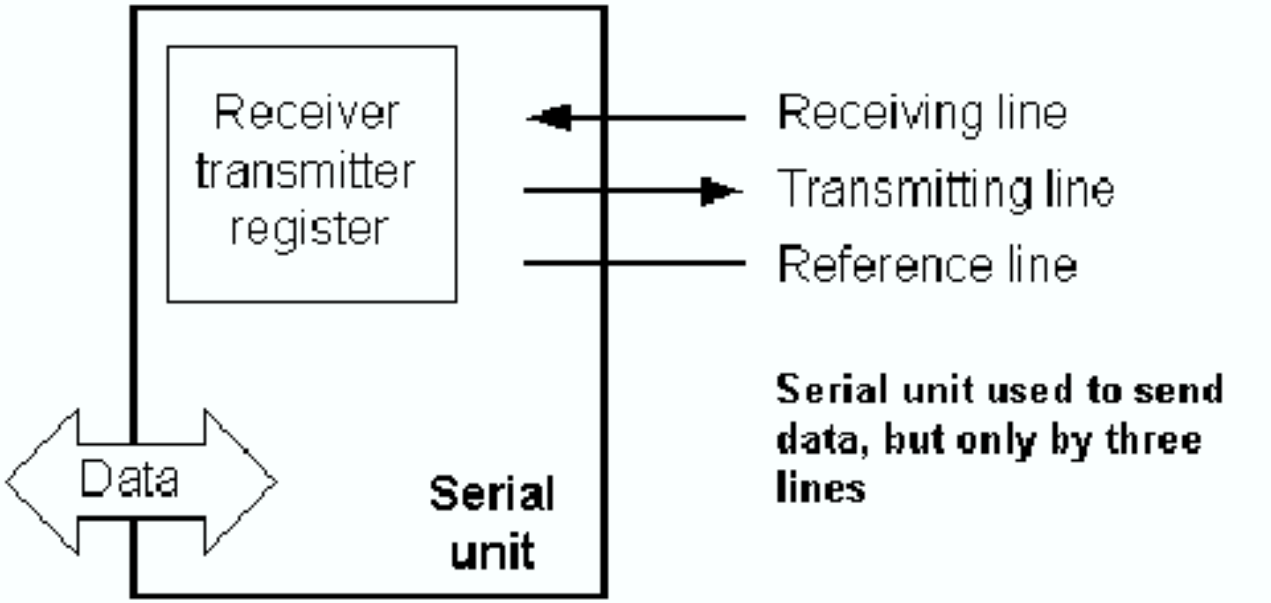
When working with ports, we first of all choose the port, then send data to it or receive data from it.



Example of a simplified input-output unit that provides communication with external world

The Serial communications unit

- It is used to exchange data with the external world while using only three cables: **transmit**, **receive** and **ground**. There are two types of serial communications: asynchronous (*serial communication interfaces*, SCI or UART) and synchronous (*serial peripheral interface*, SPI). These interfaces are used, for example, to connect the MCU to a PC (SCI) or an EEPROM or another microcontroller, a GSM MODEM, a Bluetooth module...

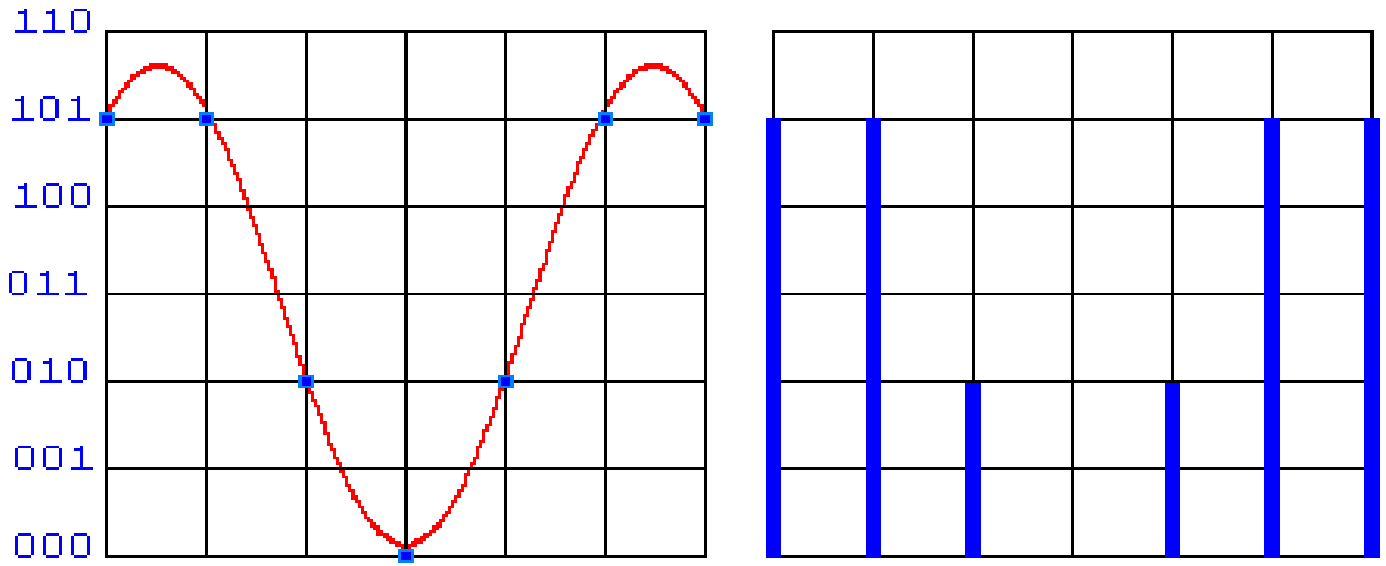


The Timer unit

- It is used as an internal base time for the system; it also generates signals and counts events or interruptions. Some microcontrollers have *a watchdog timer* (watchdog): if this device is not re-initialized by the program after preset intervals, it will consider that there is a problem in the software and causes a *hardware reset*.

The A/D converter unit

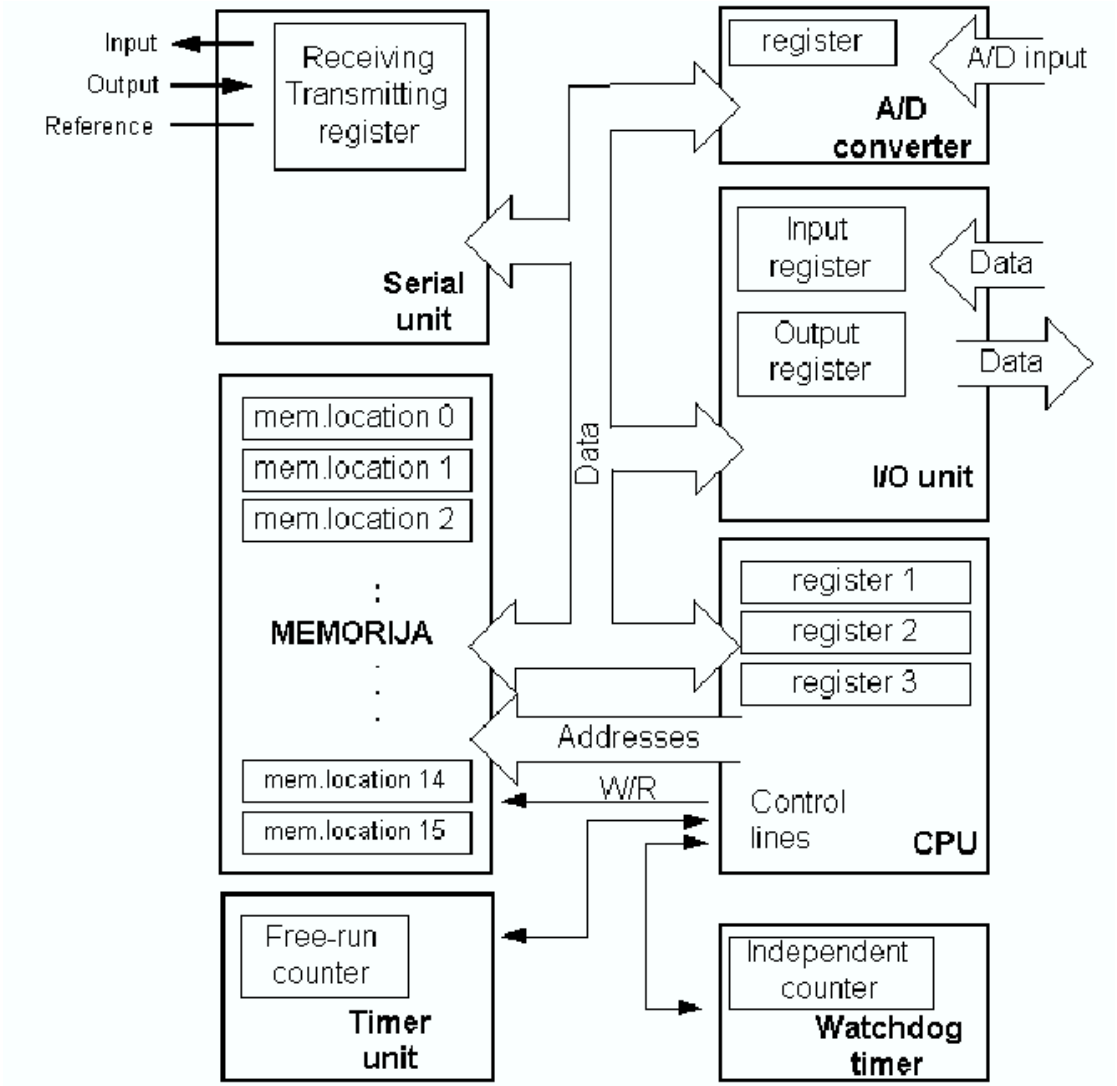
This module first of all converts the analogue signal into a digital signal before it is processed by the CPU Unit



Generally the Analogue to Digital Converter unit has a resolution of 10 bits and the range of analogue value to be converted is 0V to 5V

The block diagram of a microcontroller

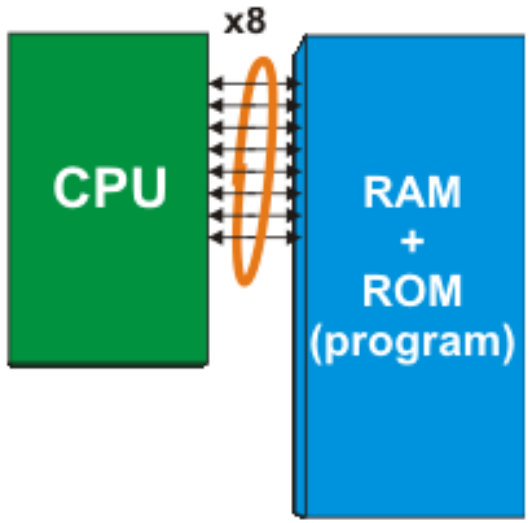
- The diagram below summarizes the working procedure of a microcontroller.



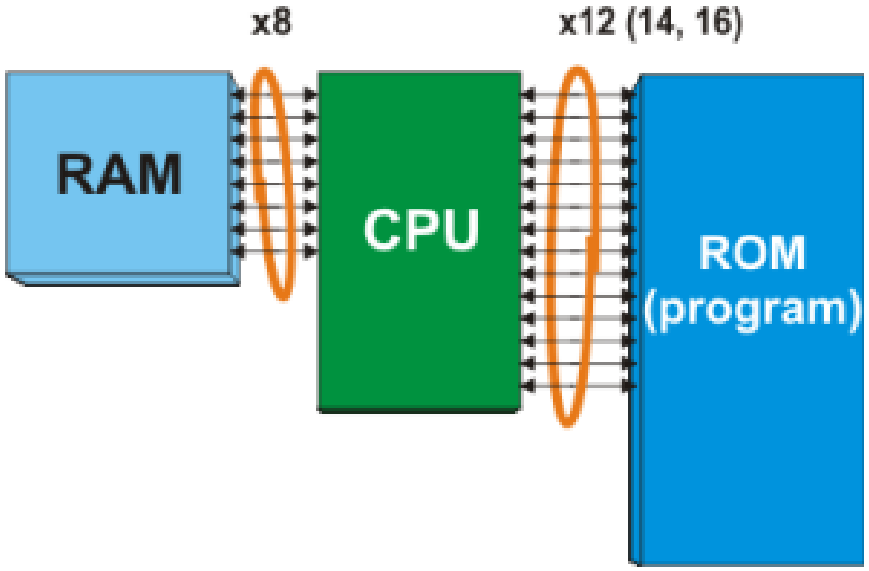
The internal architecture and Instruction Set

- All upgraded microcontrollers use one of two basic design models. It is about two different ways of data exchange between CPU and memory unit.

Von-Neumann architecture



Harvard Architecture



The Instruction Set

All instructions that can be understood by the microcontroller are known as **instruction set**.

RISC (Reduced Instruction Set Computer)

In this case, the idea is that the microcontroller recognizes and executes only basic operations addition, subtraction, etc.).

CISC (Complex Instruction Set Computer)

Instruction Set

16F87x Instruction Set [35]

TABLE 13-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into standby mode	1	00 0000 0110 0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

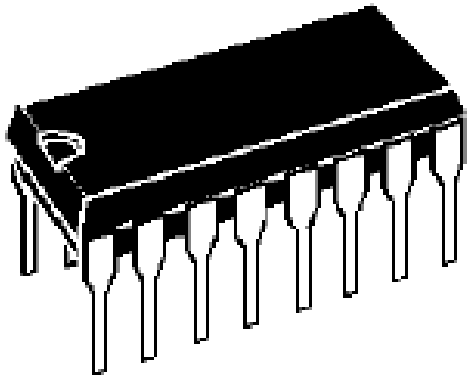
Families of microcontrollers

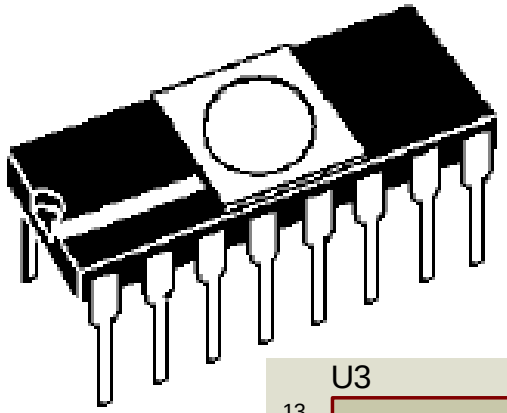
- Motorola Microcontrollers
- Atmel Microcontrollers
- Atemega Microcontollers
- PIC Microcontrollers
- …

Necessary material for the implementation of a microcontroller system

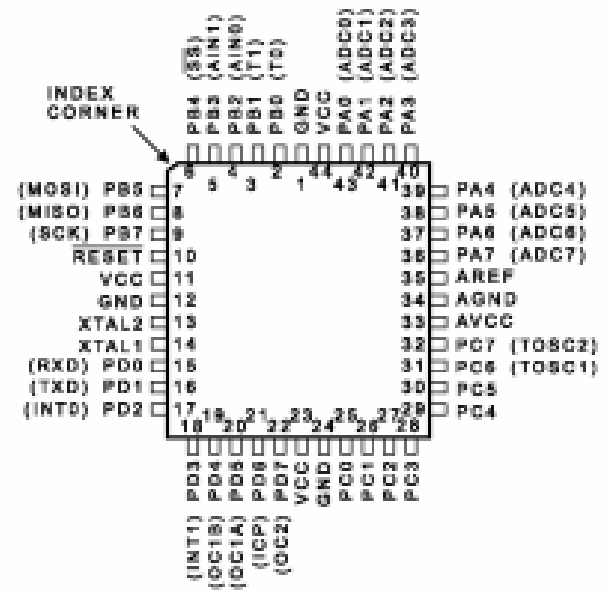
- The microcontroller itself
- A compiler (MPLAB, MikroC, Proton…)
- A simulator (ISIS-PROTEUS)
- The power supply (generally DC 5V)
- The Oscillator
- The reset button (not compulsory)
- A microcontroller programmer
- Other electronics components

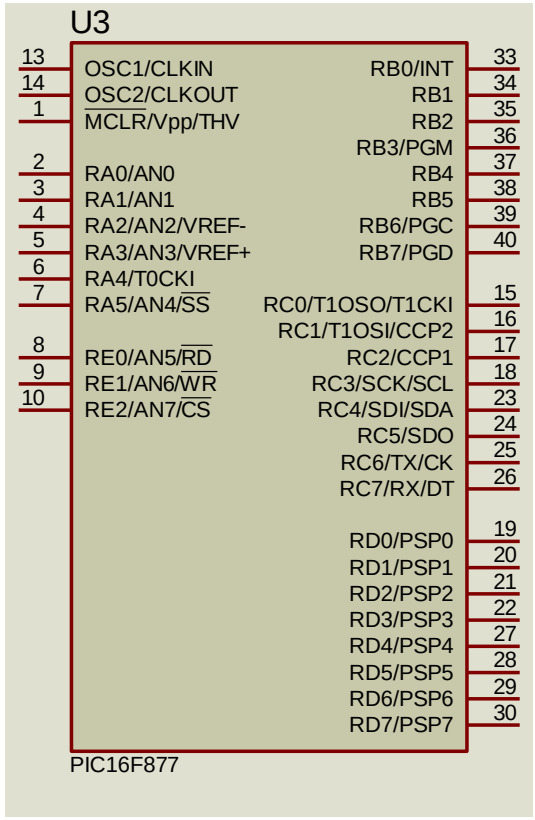
The microcontroller itself

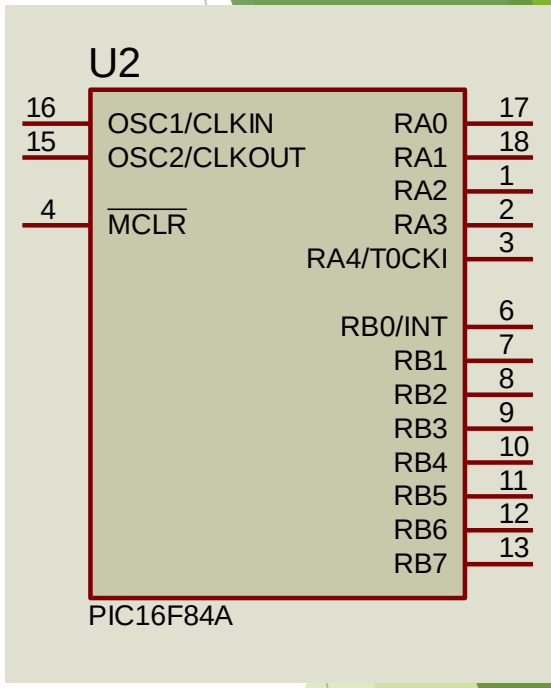




Physical aspect of the microcontroller

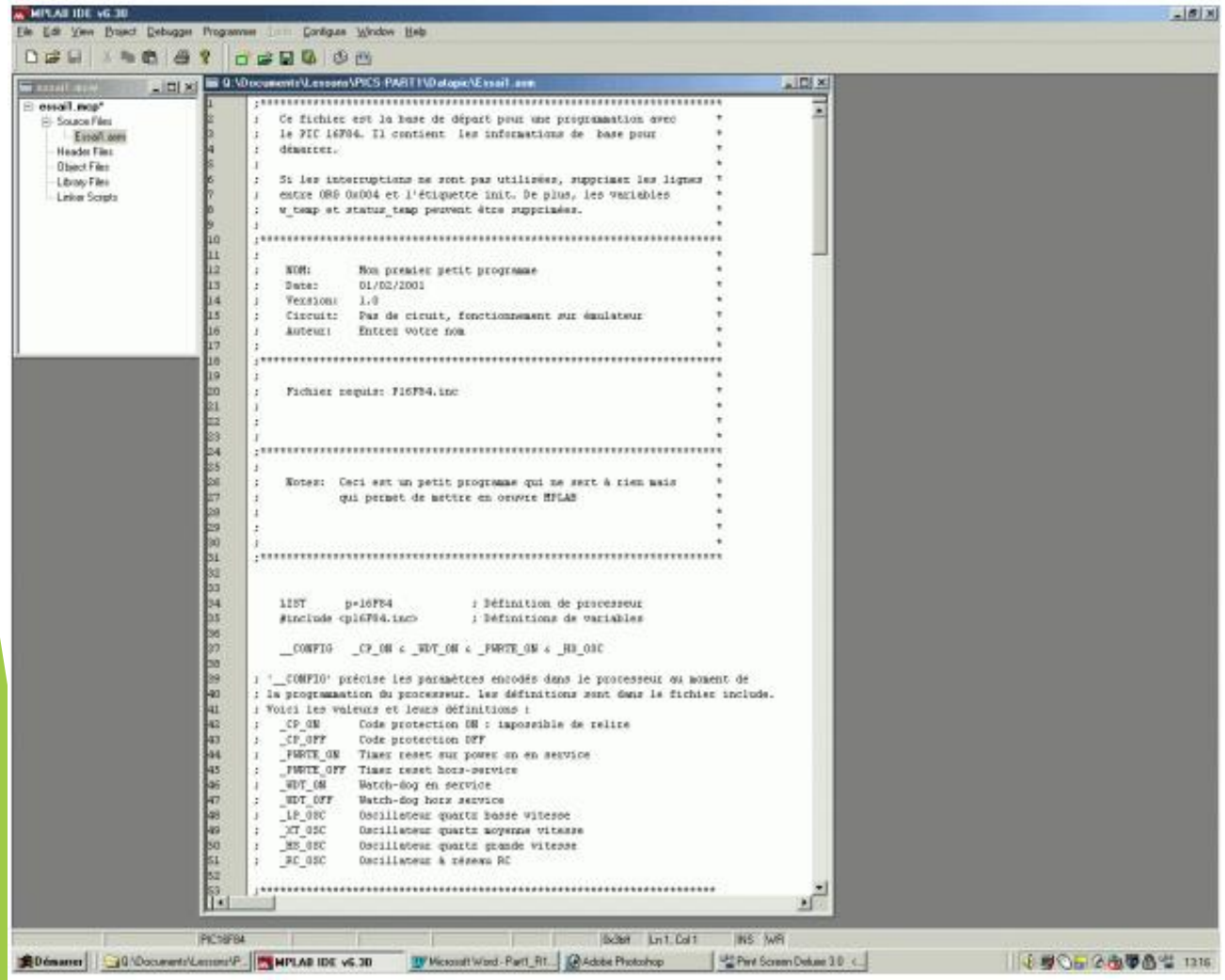




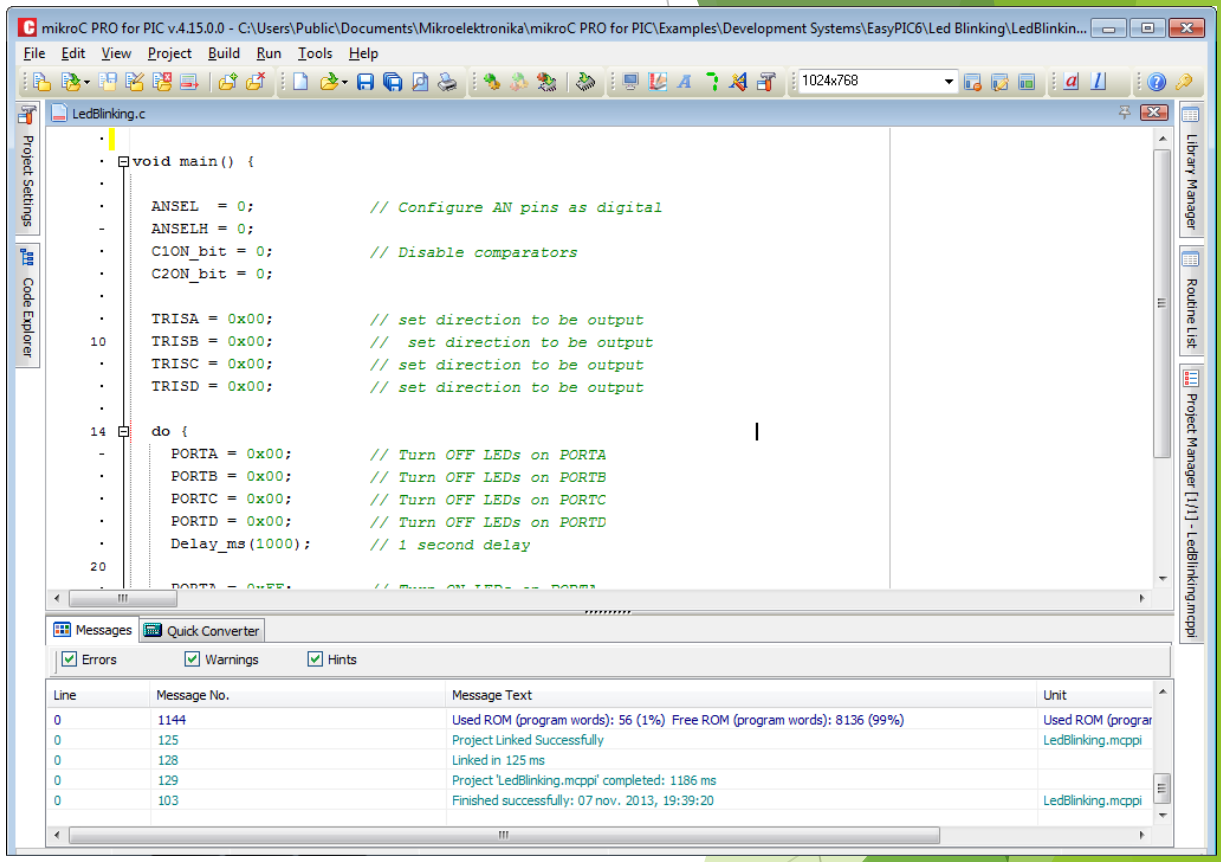


The microcontroller in the Proteus simulator

A compiler (MPLAB, MikroC, Proton…)



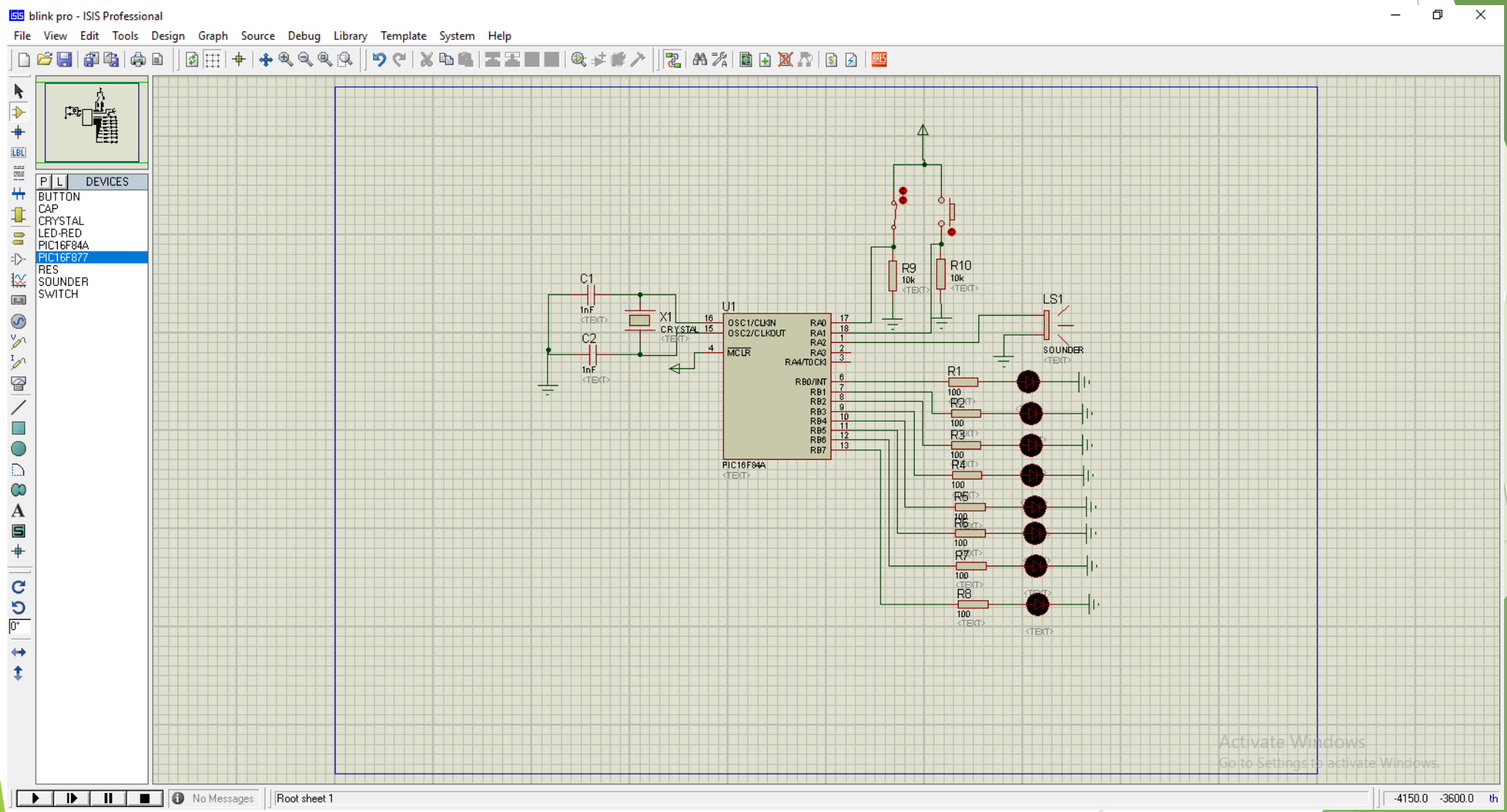
MPLAB environment



MikroC environment

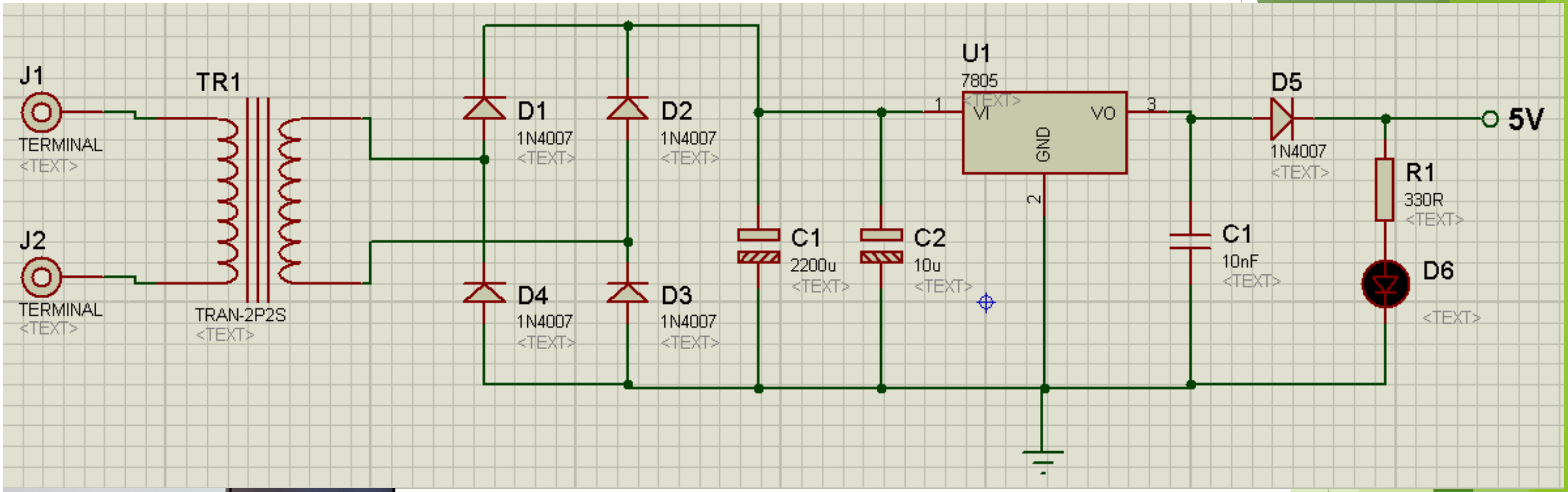
A simulator (ISIS-PROTEUS)

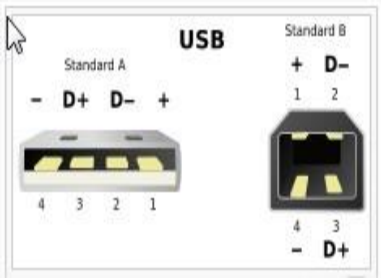
Many softwares can be used as simulator: ISIS-Proteus, Orcad, PSpice, Fritzing … But we will use ISIS-Proteus as working tool



ISIS-Proteus environment

The power supply (generally DC 5V)





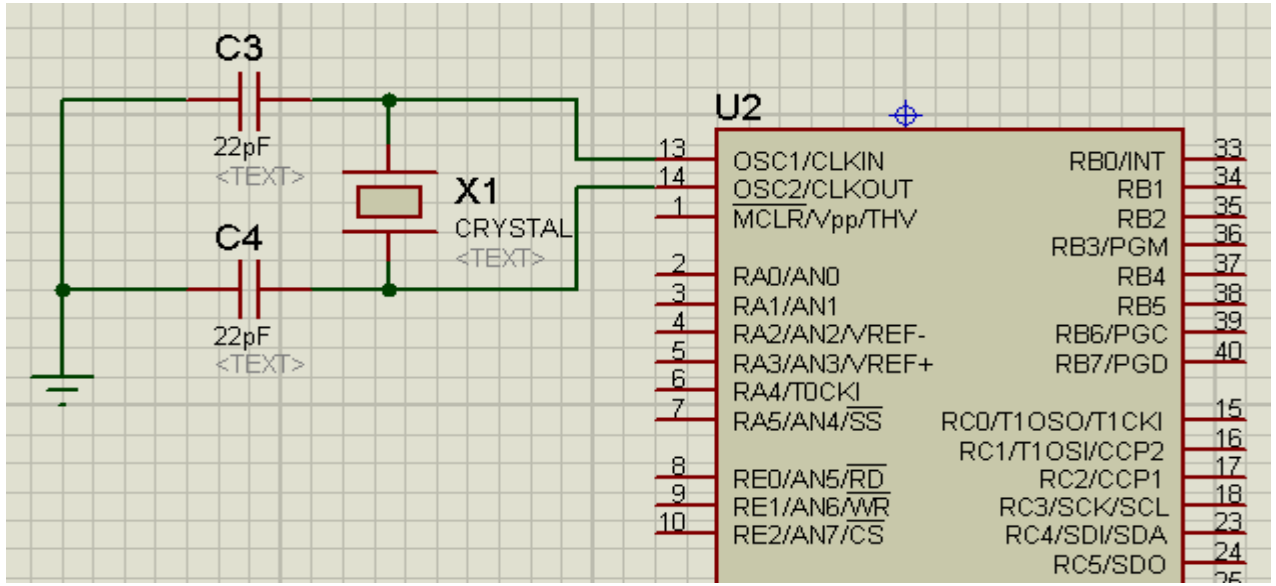
Pin 1: Vcc, representing +5V, and connected to the red wire.

Pin 2: Data-, connected to the white wire.

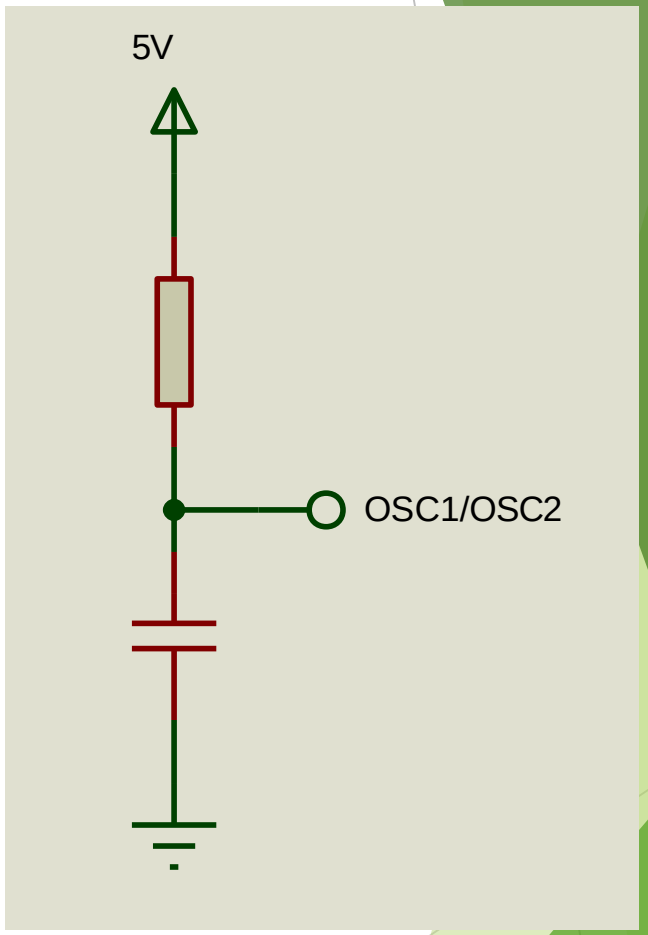
Pin 3: Data+, connected to the green wire.

Pin 4: representing the Ground, connected to the black wire.

The Oscillator circuit

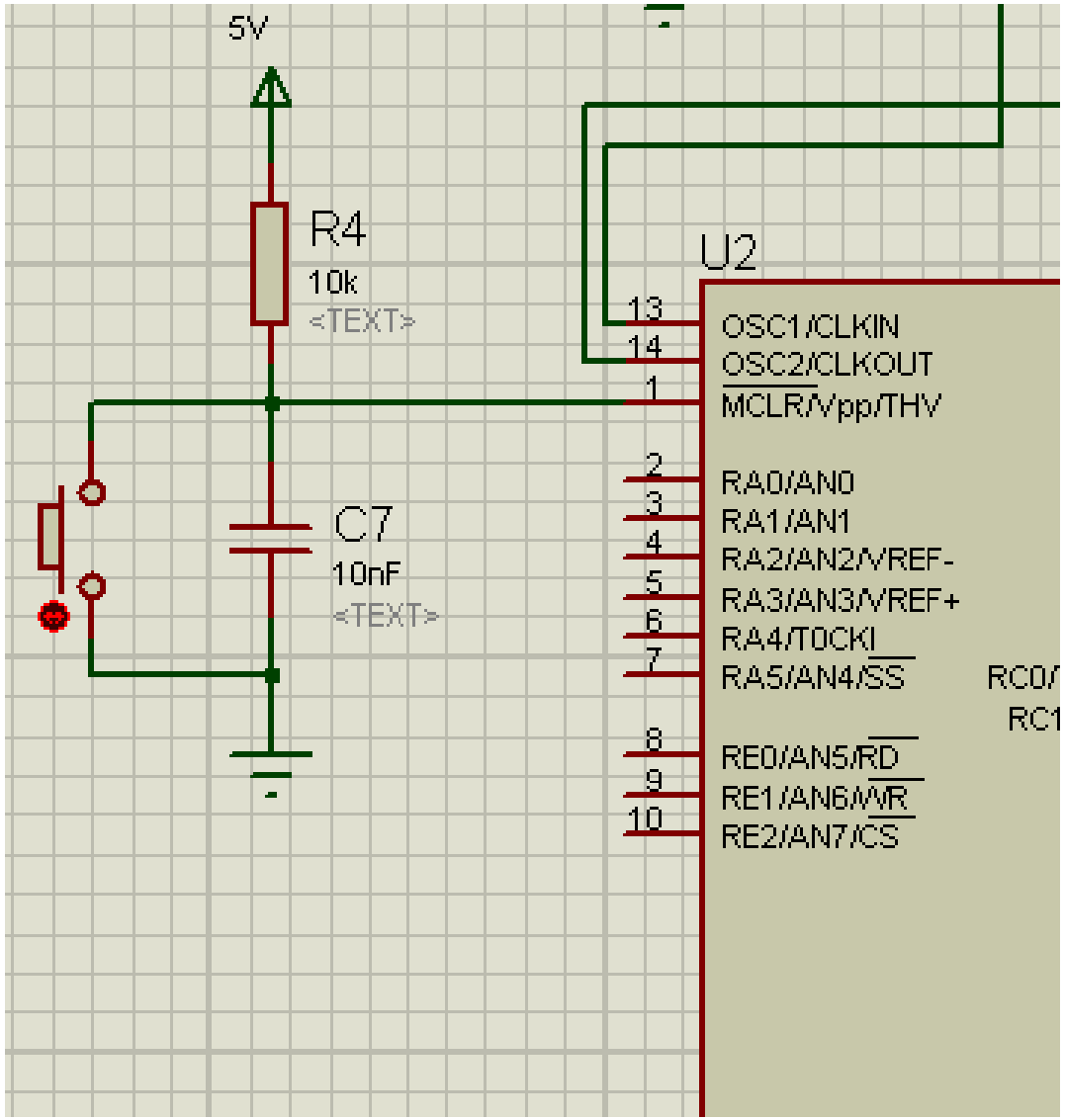


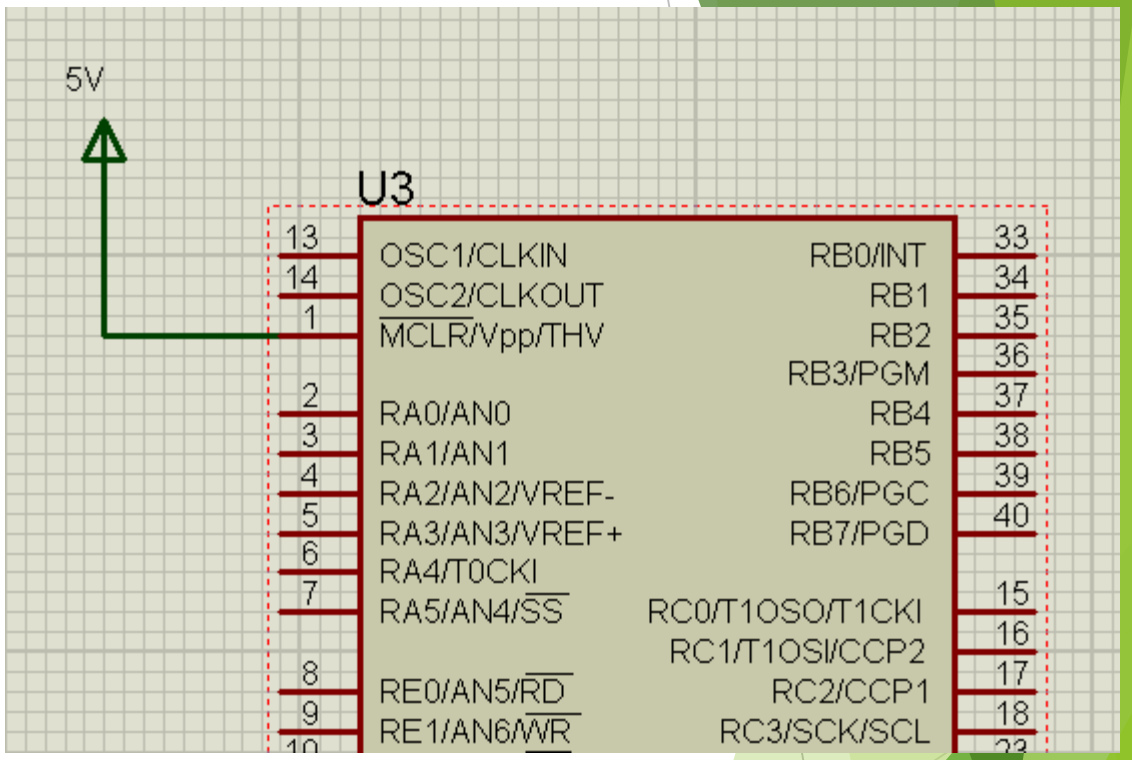
The frequency of the oscillator is the one written on the crystal oscillator



The frequency of the oscillator is $f = \frac{1}{RC}$

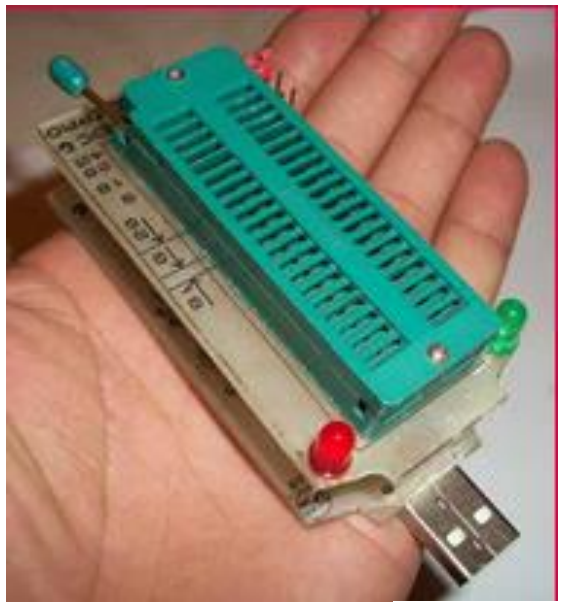
The reset button (not compulsory)

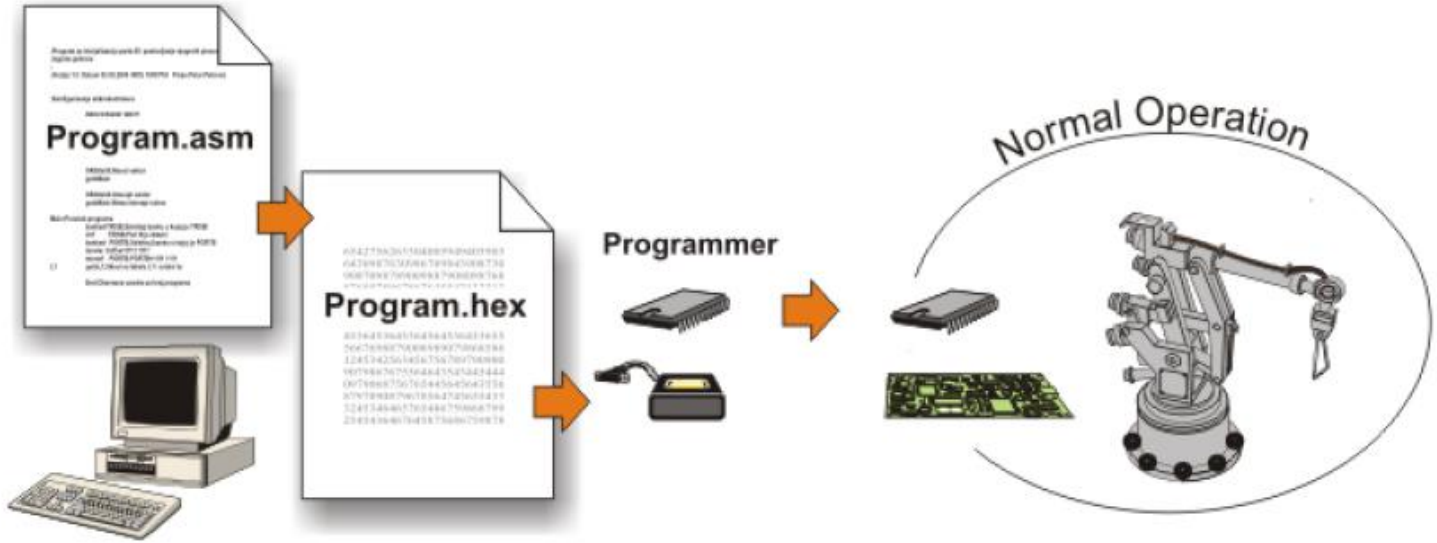




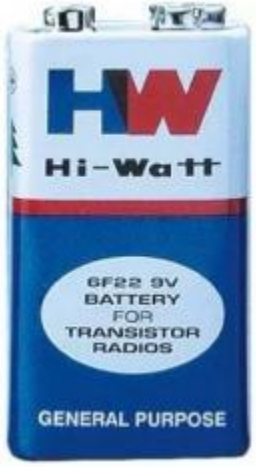
The RESET pin is labeled $\overline{MCLR}$

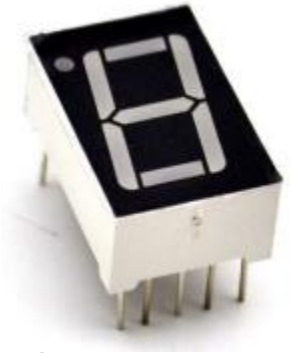
A microcontroller programmer





Other electronics components





Microcontroller Manufacturers

- Analog Devices
- Atmel
- Dallas Semiconductor
- Freescale Semiconductor
- Hitachi Semiconductor
- Intel
- Microchip
- National Semiconductor
- Renesas
- STMicro
- Texas Instruments
- Zilog

Microcontroller applications

- Microcontrollers are integrated in all domains of our life such as:
  - Super market, in cash registers;
  - Alarm clock;
  - Digital camera;
  - Mouse;
  - Mobile phone;
  - Automobile industry (safety systems, fuel injection, …);
  - Traffic signals;
  - Refrigerator;

  Etc.