PROGRAMMABLE MACHINES (COMPUTERS)

- Programmable systems:

When complex automations are performed, it may be convenient to use a multi-purpose control system; it means it can be programmed according to the wishes of the user. For this they use machines that are able to execute a program elaborated by the designer or by the own user.

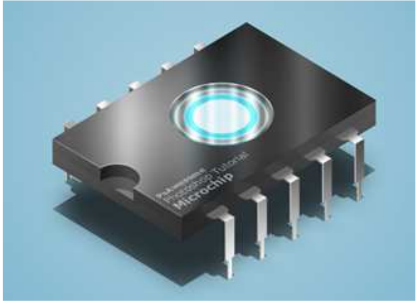
All the programmable machines have the same internal structure and are composed among others of the following fundamental elements:

- **Microprocessor**: It is the brain of the machine. It is responsible for searching and processing the information.
- **Memory**: In it, the program that must run the microprocessor is written. It is also used by the microprocessor to store information needed to perform operations. Types:
    - **RAM**: The memory used by the machine to save information temporarily. It is reusable, temporary and there is no information on it indefinitely. In personal computers, it is sold in cards that can be removed and put on the computer. The main memory or RAM (Random Access Memory) is where the computer stores the data it is currently using. The storage is considered temporary because the data and programs remain in it while the computer is on or is not restarted. Physically, they consist of a set of chips or chip modules normally connected to the motherboard. Memory chips are black rectangles that are usually soldered into clusters on "pin" plugs or contacts.
    - **ROM**: (read only memory) Permanent memory that can't be erased or written once configured. In it, in computer science, the data of access to the operating system is saved. If it is damaged, the computer does not work.
- **Clock**: Marks the working speed of the programmable machine. The faster the machine is, the more powerful it is but more complex and expensive. A Hertz is one operation per second, so a computer that works at 1KHz works at 1,000 operations per second, at 1MHz to 1,000,000 at 1Ghz 1,000,000,000 ...
- **Peripherals**: These are the elements that the user manages to communicate with the machine. On a computer, they are the keyboard, mouse, screen, printer, etc. They can be data input or output.
    - Input / Output Units: These are circuits that translate signals from peripherals to electrical signals that can be interpreted by the microprocessor and vice versa. They are sold in the form of cards that are inserted into slots in the computer's motherboard, where the microprocessor is. In some computers are integrated into the motherboard. In a programmable automaton can be thermometers, sensors ...
- **Programming languages**: These are codes with very simple commands that are easily translatable to the language that a microprocessor understands (0, 1, absence or presence of current). They are handled by experts. Special devices are required to transmit this data to the machine.
- **Buses**: These are the internal cables through which the electrical signals circulate between the micro and the other circuits (memories, I / O units). They can be SATA (modern) or IDE.
- **Data storage disks**. Not being essential in a small programmable automaton, they are essential in a personal computer. They are devices that save data for later use

or access. They are removable and can be changed, added, replaced. The best known are the hard drive C, floppy disk, CD, DVD, USB stick ...

- **Ports of entry and / or exit**. They are the name of the pins where we connect our peripherals. The most common are keyboard, mouse, screen, serial ports, printer or scanner (parallel ports), USB, for devices with this pin etc.

- When all these circuits (micro, memory and I / O units) are integrated in a single chip, this is called microcontroller. Currently they are not very expensive and can perform many functions in appliances, mobile phones, automation of garages, elevators, etc.



- When a complex set of chips and connections is integrated, it is called a computer or PC (Personal Computer).

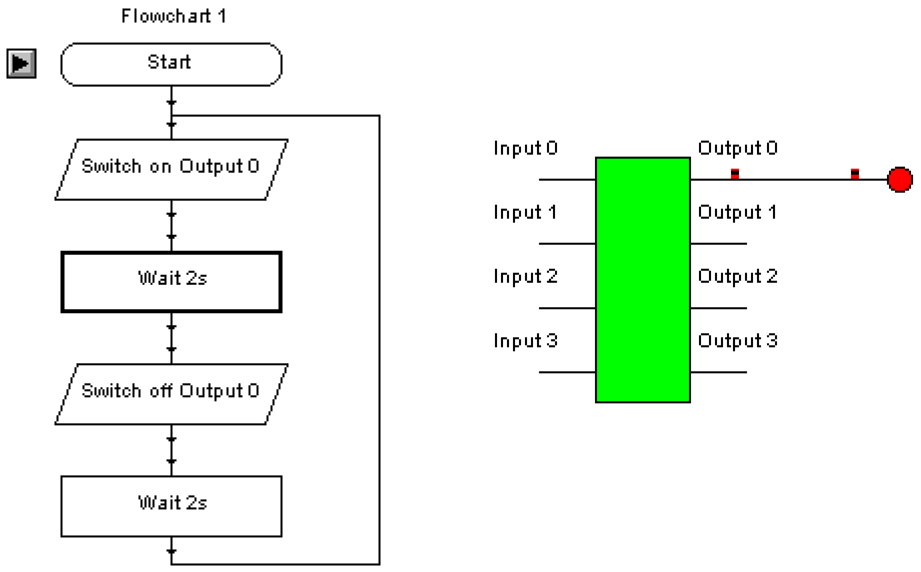
  A simple microcontroller would have input ports, output ports, and external power. Internally they have memory where the user can store the desired program by means of certain orders that compose its programming language.

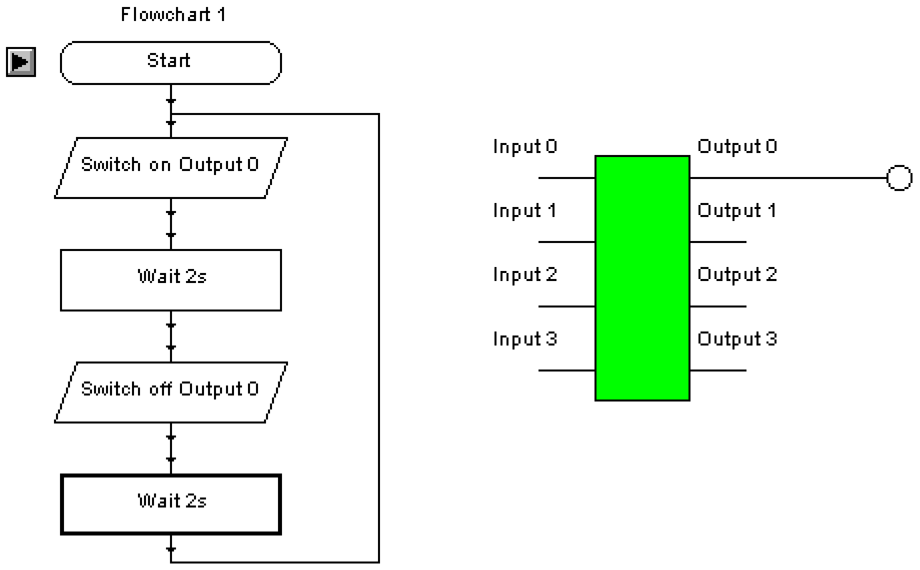
- The automation of a machine or a simple production process results in the physical and mental liberation of the man from that job. An automatism is the physical device that performs a certain function controlling its operation.

STEPS TO DESIGN A REPETITIVE TASK. (TO GET THE MACHINE MAKES WHAT WE WANT).

- Find a need. We want a machine to do a task, for example, because it is repetitive and lacking in ingenuity.
- Search for a COMPUTER ANALYST. It is the professional who breaks down this complex task into all the simple steps needed to be carried out. None can be lacking, nor can any be complex. It has to be broken down into the simplest possible.

  Search for a PROGRAMMER. It is the professional that transforms simple commands into instructions that are executable by the microprocessor and can be recorded in the ROM or on a hard disk.
- EXAMPLE: We want a flashing light to alert you that there is a level crossing without barriers. Instead of connecting the bulb to the current, at one of its ends I connect it to a microcontroller.



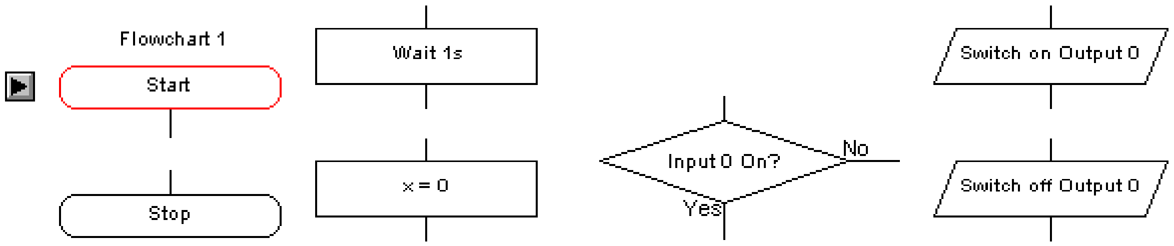


We have done the work of the analyst, decomposing the task into simple orders. The crocodile program makes us the task of the programmer (only if the analyst has done well).

The commands that the microchip understands are very basic, so to talk to him you have to break the task down into simple commands. The fundamental orders are:

- Start (empieza)
- Stop (termina)
- Wait x (espera x segundos)
- Switch on output x (enciende la salida x)
- Switch off output x (apaga la salida x)
- Input x on? yes/no (¿está la entrada x conectada?, sí o no.
- X=0 (toma una variable X un valor que le damos, en este caso 0).
- Add one to X (suma 1 a X)
- X=0? Yes/no (X = 0 sí o no)

The symbols used to represent these orders are the follows:



OVAL	RECTANGLE	RHOMB	RHOMBOID
Orders of beginning and end of the programming.	Processor commands or input variables.	Input reading and, depending on the result, action.	Orders to the outputs of the microprocessor.

COMPUTER JOBS. Initially, the profession was formalized from the Taylorian approach of the specialization of functions in the company. Thus, the software production process is conceived as a set of highly specialized tasks where the role of each professional category is clearly defined:

- **The analyst** has the task of analyzing a problem and describing it in order to be solved through an information system.

- **The programmer** whose only function was to translate the analyst's specifications into executable code by the computer. These specifications are included in a document called the cargo book, a means of communication between the two. Note that this was considered a mechanical work with low qualification.

Today it is recognized that this approach is not valid to organize tasks of an intellectual type, such as the production of software. So the programmer profession has evolved. Communication difficulties between analysts and programmers (a mere document is not enough to describe what one wants to do) gave rise to an intermediate professional category, called analyst-programmer. The original conception of the programmer has disappeared being replaced by this one: the one of a professional much more formed and with functions less "mechanical".

PROGRAMMING:
BASIC is a programming language that was originally developed as a teaching tool; has spread among home computers since the 1980s, and is still very popular today, in many dialects quite different from the original.
The minimal BASIC syntax only needs the LET, INPUT, PRINT, IF, and GOTO commands. An interpreter running programs with this minimal syntax does not need a subroutine. Some of the early implementations were that simple. If a subroutine is added, nested FOR cycles and the GOSUB command can also be added. A BASIC interpreter with these characteristics needs the code to have line numbers.
1.- Perform a program that causes the machine to wait (it entertains itself counting)

```
10 x = 1
20 x = x + 1
30 print x
40 if x = 1000 goto 50 else  goto 20
50 end
```