Table of Contents

This lesson provides an introduction to electronics. We will cover what forms components form basic circuits. We will discuss sensors and actuators and finally we will take a look at microcontrollers.

1 Basic Electronics

Electricity is the movement of electrons. Electrons create charge, which we can harness to do work.

Voltage-We define voltage as the amount of potential energy between two points on a circuit. One point has more charge than another. This difference in charge between the two points is called voltage. It is measured in volts, which, technically, is the potential energy difference between two points that will impart one joule of energy per coulomb of charge that passes through it.

Current-Current is a flow of electrical charge carriers, usually electrons or electron-deficient atoms. The common symbol for current is the uppercase letter I. The standard unit is the ampere, symbolized by A. One ampere of current represents one coulomb of electrical charge (6.24 x 1018 charge carriers) moving past a specific point in one second.

AC-Alternating current describes the flow of charge that changes direction periodically. As a result, the voltage level also reverses along with the current. AC is used to deliver power to houses, office buildings, etc.

DC-DC is defined as the “unidirectional” flow of current; current only flows in one direction. Voltage and current can vary over time so long as the direction of flow does not change.

Power-In addition to voltage and current, there is another measure of free electron activity in a circuit: power. Electric power is the rate, per unit time, at which electrical energy is transferred by an electric circuit. The SI unit of power is the watt, one joule per second.

1.1 Common Components in Circuits

An electric circuit is like a pathway made of wires that electrons can flow through. A battery or other power source gives the force (voltage) that makes the electrons move. When the electrons get to a device like a light bulb, your computer, or a refrigerator, they give it the power to make it work.

Many circuits have a switch so that they can be turned on and off. When the switch is off, it makes a gap in the circuit and the electrons are not able to flow around. When the switch is turned on, it closes the gap and the electricity is able to move and make the device work.

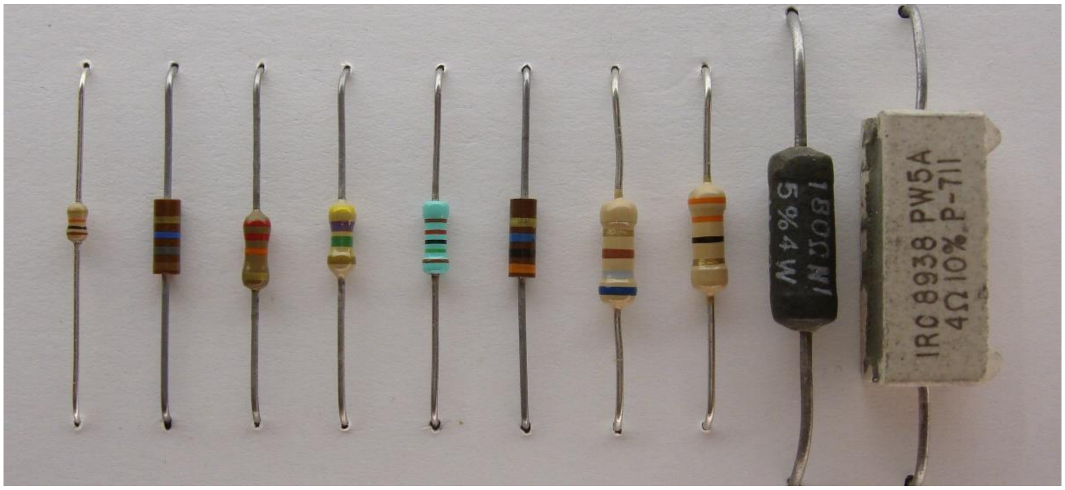
Some of the most common components in circuits are:

i) **Wires, Cables, and Connectors-**Wires and cables provide low-resistance pathways for electric currents. Most electrical wires are made from copper or silver and typically are protected by an insulating coating of plastic, rubber, or lacquer. Cables consist of a number of individually insulated wires bound together to form a multi-conductor transmission line. Connectors, such as plugs, jacks, and adapters, are used as mating fasteners to join wires and cable with other electrical devices

ii) **Batteries-** A battery is made up of a number of cells. Each cell contains a positive terminal and a negative terminal. A typical cell maintains about 1.5 V across its terminals and is capable of delivering specific amount of current that depends on the size and chemical makeup of the cell. If more voltage or power is needed, a number of cells can be added together in either series or parallel configurations. By adding cells in series, a larger-voltage battery can be made, whereas adding cells in parallel results in a battery with a higher current-output capacity. A cell converts chemical energy into electrical energy by going through what are called oxidation-reduction reactions (reactions that involve the exchange of electrons). Batteries are given a capacity rating that indicates how much electrical energy they are capable of delivering over a period of time. The capacity rating is equal to the initial current drawn multiplied by the time until the battery dies. The units for battery capacities are ampere-hours (Ah) and millampere-hours (mAh).

iii) **Switches-** A switch is a mechanical device that interrupts or diverts electric current flow within a circuit.

iv) **Resistors-** Resistors are electrical devices that act to reduce current flow and at the same time act to lower voltage levels within circuits. The relationship between the voltage applied across a resistor and the current through it is given by $V = IR$. There are numerous applications for resistors. Resistors are used to set operating current and signal levels, provide voltage reduction, set precise gain values in precision circuits, act as shunts in ammeters and voltage meters, behave as damping agents in oscillators, act as bus and line terminators in digital circuits, and provide feedback networks for amplifiers. Resistors may have fixed resistances, or they may be designed to have variable resistances. They also may have resistances that change with light or heat exposure (e.g., photoresistors, thermistors).



v) **Capacitors-**A capacitor is a passive two-terminal electrical component used to store energy electrostatically in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors (plates) separated by a dielectric (i.e., insulator). The conductors can be thin films of metal, aluminum foil or disks, etc. The 'nonconducting' dielectric acts to increase the capacitor's charge capacity. A dielectric can be glass, ceramic, plastic film, air, paper, mica, etc. Capacitors are widely used as parts of electrical circuits in many common electrical devices. Unlike a resistor, a capacitor does not dissipate energy. Instead, a capacitor stores energy in the form of an electrostatic field between its plates. When there is a potential difference across the conductors (e.g., when a capacitor is attached across a battery), an electric field develops across the dielectric, causing positive charge (+Q) to collect on one plate and negative charge (-Q) to collect on the other plate. If a battery has been attached to a capacitor for a sufficient amount of time, no current can flow through the capacitor. However, if an accelerating or alternating voltage is applied across the leads of the capacitor, a displacement current can flow. Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass. In analog filter networks, they smooth the output of power supplies. In resonant circuits they tune radios to particular frequencies. In electric power transmission systems they stabilize voltage and power flow.

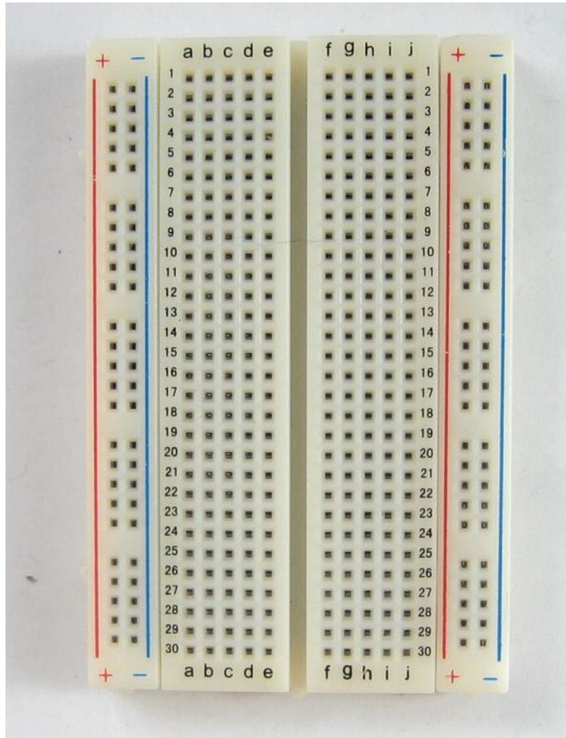
vi) **Inductors-** An inductor, also called a coil or reactor, is a passive two-terminal electrical component that stores electrical energy in a magnetic field when electric current flows through it. An inductor typically consists of an electric conductor, such as a wire, that is wound into a coil around a core. When the current flowing through an inductor changes, the time-varying magnetic field induces a voltage in the conductor, described by Faraday's law of induction. According to Lenz's law, the direction of induced electromotive force (e.m.f.) opposes the change in current that created it. As a result, inductors oppose any changes in current through them. Inductors are used extensively in analog circuits and signal processing. Applications range from the use of large inductors in power supplies, which in conjunction with filter capacitors remove residual hums known as the mains hum or other fluctuations from the direct current output, to the small inductance of the ferrite bead or torus installed around a cable to prevent radio frequency interference from being transmitted down the wire. Inductors are used as the energy storage device in many switched-mode power supplies to produce DC current. The inductor supplies energy to the circuit to keep current flowing during the "off" switching periods.

2 Building Your Own Circuits

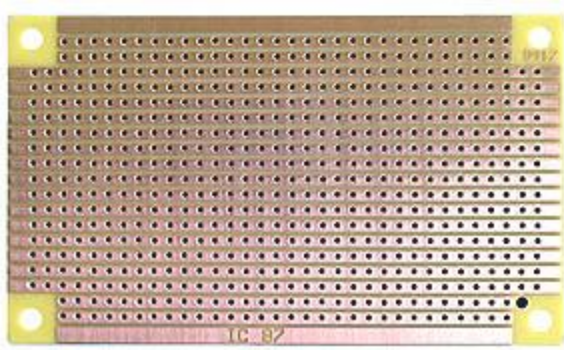
In almost all IoT 'thing' design, you will need to construct your own circuit.

The Breadboard- Breadboards are one of the fundamental components when learning how to build circuits. An electronics breadboard (as opposed to the type on which sandwiches are made) is actually referring to a solderless breadboard. These are great units for making temporary circuits and prototyping, and they require absolutely no soldering.

Prototyping is the process of testing out an idea by creating a preliminary model from which other forms are developed or copied, and it is one of the most common uses for breadboards. If you aren't sure how a circuit will react under a given set of parameters, it's best to build a prototype and test it out.



Protoboards -If you’re ready to finalize a breadboarded circuit, but not quite ready to create a printed circuit board, then your next step is protoboard. There are multiple varieties of prototyping board, but the two most common are referred to as perf board and stripboard. Both are made from a flat sheet of resin with a grid of holes drilled in them, however the conductive copper on the underside is different.



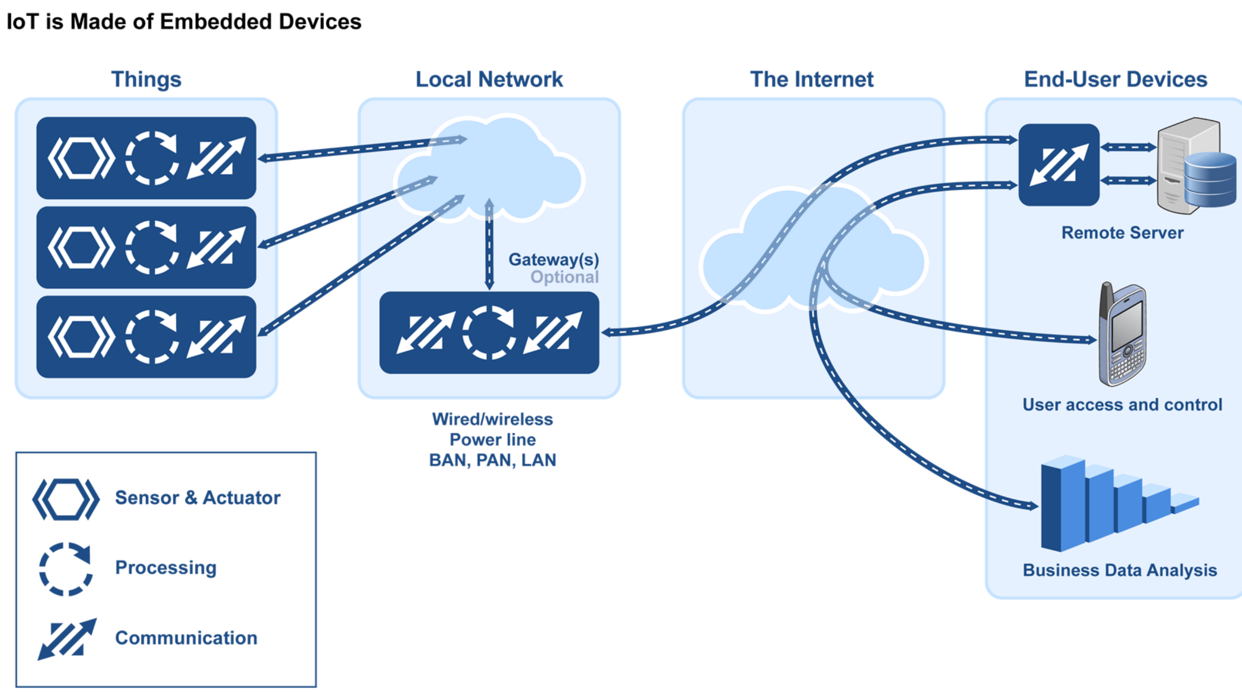
PCBs(Printed Circuit Boards)- PCB is an acronym for printed circuit board. It is a board that has lines and pads that connect various points together. There are traces that electrically connect the various connectors and components to each other. A PCB allows signals and power to be routed between physical devices. Solder is the metal that makes the electrical connections between the surface of the PCB and the electronic components. Being metal, solder also serves as a strong mechanical adhesive.

- **Pad** - a portion of exposed metal on the surface of a board to which a component is soldered.

Good, now we have a good understanding of basic electronics. We will apply this knowledge in the preceding discussions as we deal more with the IoT end devices, what is commonly known as the things.

3 Major Components of an IoT Ecosystem

The following diagram depicts the major components that make up an IoT system



Our focus for this course is on two of the parts that make the ‘things’, namely sensors and actuators, and processing. Another important unit not shown in the ‘things’ part of the figure above is the power source.

Before we begin, let us get out of our way some important definitions you will hear a lot when discussing sensors and microcontrollers.

3.1 Analog Vs Digital Signals

We live in an analog world. There is an infinite amount of colors to paint an object (even if the difference is indiscernible to our eye), there are an infinite number of tones we can hear, and there are an infinite number of smells we can smell. The common theme among all of these analog signals is their infinite possibilities.

Digital signals and objects deal in the realm of the discrete or finite, meaning there is a limited set of values they can be. Working with electronics means dealing with both analog and digital signals, inputs and outputs. Our electronics projects have to interact with the real, analog world in some way, but most of our microprocessors, computers, and logic units are purely digital components. These two types of signals are like different electronic languages; some electronics components are bi-lingual, others can only understand and speak one of the two.

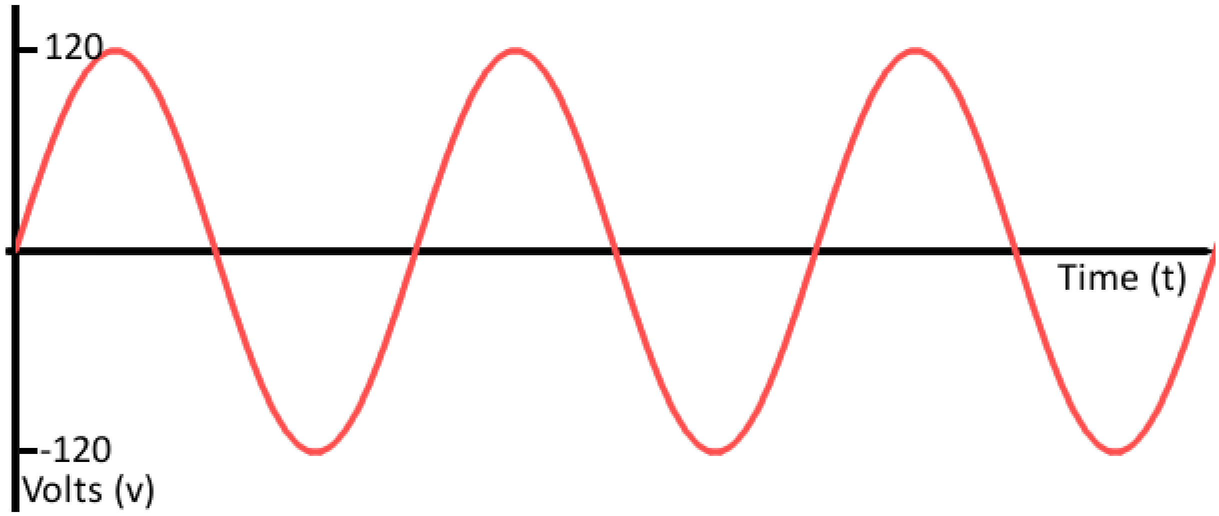
The signals we're talking about are time-varying "quantities" which convey some sort of information. In electrical engineering the quantity that's time-varying is usually voltage (if not that, then usually current). So when we talk about signals, just think of them as a voltage that's changing over time.

Signals are passed between devices in order to send and receive information, which might be video, audio, or some sort of encoded data. Usually the signals are transmitted through wires, but they could also pass through the air via radio frequency (RF) waves. Audio signals, for example might be transferred between your computer's audio card and speakers, while data signals might be passed through the air between a tablet and a WiFi router.

3.1.1 Analog Signals

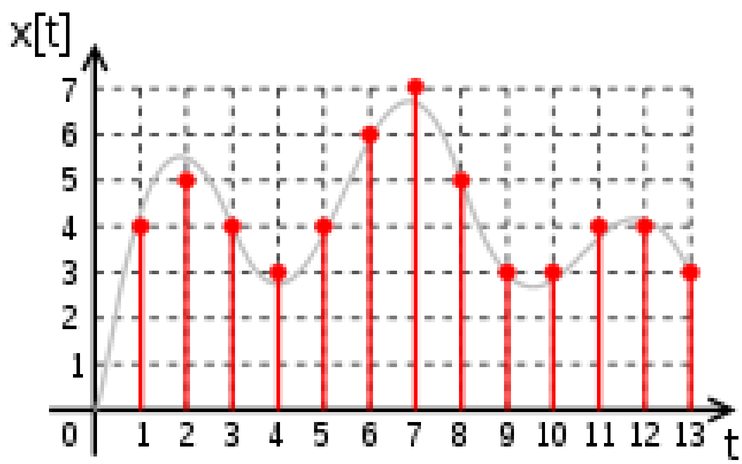
An analog signal is a kind of signal that is continuously variable. While these signals may be limited to a range of maximum and minimum values, there are still an infinite number of possible values within that range. Video and audio transmissions are often transferred or recorded using analog signals. Pure audio signals are also analog. The signal that comes out of a microphone is full of analog frequencies and harmonics, which combine to make beautiful music.

The figure below shows an analog signal



3.1.2 Digital Signals

A digital signal is a kind of signal that has a limited number of steps along its range. The number of values in the set can be anywhere between two and a-very-large-number-that's-not-infinity. Not all audio and video signals are analog. Standardized signals like HDMI for video (and audio) and MIDI, I2S, or AC'97 for audio are all digitally transmitted.



Most communication between integrated circuits is digital. Interfaces like serial, I2C, and SPI all transmit data via a coded sequence of square waves.

Now let us begin our discussion on the power source, sensors, actuators and microcontrollers.

3.2 Power Source

The smart object(things) need to be powered. Power requirements vary from application to application. Typically, smart objects are limited in power and are deployed for a very long time in places that are not easily accessible. If the things rely on battery power, power efficiency, judicious power management, sleep modes, ultra-low power consumption hardware and so on are critical design elements.

3.3 Sensors

A sensor does exactly what the name suggests. A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. In the broadest definition, a sensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics, whether as simple as a light or as complex as a computer. A sensor can either be analog or digital.

There are many different types of sensors to measure all kinds of quantities.

Position -A position sensor measures the position of an object; the position of an object can either be in absolute terms or in relative terms e.g. inclinometer, proximity.

Occupancy and Motion Sensors- Occupancy sensors detect the presence of objects in a room, motion sensors detect movement. E.g. radar.

Velocity and acceleration Sensors- Detect velocity and acceleration e.g. accelerometer, gyroscope.

Force -e.g. force gauge, tactile sensor (sound sensor)

Pressure Sensor- Pressure sensors are related to force sensors; they measure force applied by liquids or gases e.g. barometer.

Flow Sensors- Detect the rate of fluid flow e.g. water meter.

Acoustic – Measure sound levels and convert that information into digital or analog signals e.g. microphone

Humidity Sensors- Detect the amount of water vapor in the air or a mass e.g. hygrometer, soil moisture sensor.

Light- detect the presence of light e.g. infrared sensor, photodetector.

Radiation Sensors- detect radiation in the environment. e.g. neutron detector.

Temperature Sensor- Measure the amount of heat or cold present in the environment. e.g. thermometer.

Chemical Sensor-measure the concentration of chemicals in a system. e.g. smoke detector.

Biosensors- detect various biological elements such as organisms, tissues, cells, enzymes, antibodies and nucleic acid.

3.4 Actuators

Actuators are natural complements to sensors. These devices receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect, usually some type of motion, force and so on.

3.5 Processing Unit

This is the part of the thing responsible for acquiring data, processing and analyzing sensing information received by the sensors, coordinating control signals to any actuators and controlling a variety of functions on the smart objects, including the communication and power systems. The specific type of processing unit that is used can vary greatly depending on the specific processing needs of different applications. The most common is a microcontroller because of its small form factor, flexibility, programming simplicity, ubiquity, low power consumption and low cost.

A microcontroller is like a small computer on a single IC. It contains a processor core, ROM, RAM and I/O pins dedicated to perform various tasks. Microcontrollers are generally used in projects and applications that require direct control of user. As it has all the components needed in its single chip, it does not need any external circuits to do its task so microcontrollers are heavily used in embedded systems. A microcontroller can be called the heart of embedded system. Microcontroller is a compressed microcomputer manufactured to control the functions of embedded systems in office machines, robots, home appliances, motor vehicles, and a number of other gadgets. A microcontroller comprises of components like – memory, peripherals and most importantly a processor. Microcontrollers are basically employed in devices that need a degree of control to be applied by the user of the device. Some examples of popular microcontrollers are 8051, AVR, PIC series of microcontrollers.

3.5.1 Types of Microcontroller:

Microcontrollers are divided into categories according to their memory, architecture, bits and instruction sets. So let's discuss types of microcontrollers: -

3.5.1.1 Bits:

- 8 bits microcontroller executes logic & arithmetic operations. Examples of 8 bits micro controller is Intel 8031/8051.
- 16 bits microcontroller executes with greater accuracy and performance in contrast to 8-bit. Example of 16 bit microcontroller is Intel 8096.
- 32 bits microcontroller is employed mainly in automatically controlled appliances such as office machines, implantable medical appliances, etc. It requires 32-bit instructions to carry out any logical or arithmetic function.

3.5.1.2 Memory:

- External Memory Microcontroller – When an embedded structure is built with a microcontroller which does not comprise of all the functioning blocks existing on a chip it is named as external memory microcontroller. For illustration- 8031 microcontroller does not have program memory on the chip.
- Embedded Memory Microcontroller – When an embedded structure is built with a microcontroller which comprise of all the functioning blocks existing on a chip it is named as embedded memory microcontroller. For illustration- 8051 microcontroller has all program & data memory, counters & timers, interrupts, I/O ports and therefore its embedded memory microcontroller.

3.5.1.3 Instruction Set:

- CISC- CISC means complex instruction set computer, it allows the user to apply 1 instruction as an alternative to many simple instructions.
- RISC- RISC means Reduced Instruction Set Computers. RISC reduces the operation time by shortening the clock cycle per instruction.

3.5.1.3.1 RISC

Reduced instruction set Computer. It is a type of microprocessor that has been designed to carry out few instructions at the same time. As instruction are few it can be executed in a less amount of time. Another advantage is the use of fewer transistor reducing its cost.

Features include:

- Demand less decoding
- Uniform instruction set
- Identical general purpose register
- Simple addressing modes
- Fewer data types in hardware

3.5.1.3.2 CISC

Complex instruction set computer. It's actually a CPU designed to carry out many operation in a single in a single instruction. These can be loading from and to memory and performing mathematical operation etc.

Features Include:

- Complex instruction
- More number of addressing modes
- Highly Pipelined
- More data types in hardware

3.5.1.4 Memory Architecture:

- Harvard Memory Architecture Microcontroller
- Princeton Memory Architecture Microcontroller

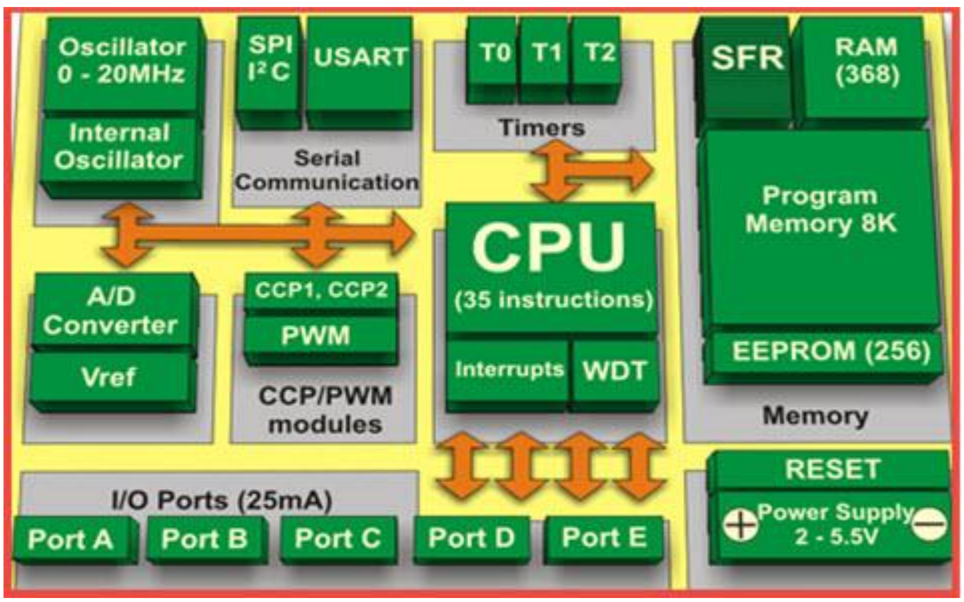
3.5.2 Major Components of Microcontrollers

Any electric appliance that stores, measures, displays information or calculates comprise of a microcontroller chip inside it. The basic structure of a microcontroller comprise of:-

1. CPU – Microcontrollers brain is named as CPU. CPU is the device which is employed to fetch data, decode it and at the end complete the assigned task successfully. With the help of CPU all the components of microcontroller is connected into a single system. Instruction fetched by the programmable memory is decoded by the CPU.
2. Memory – In a microcontroller memory chip works same as **microprocessor**. Memory chip stores all programs & data. Microcontrollers are built with certain amount of ROM or RAM (EPROM, EEPROM, etc) or flash memory for the storage of program source codes.
3. Input/output ports – I/O ports are basically employed to interface or drive different appliances such as- printers, LCD's, LED's, etc.
4. Serial Ports – These ports give serial interfaces amid microcontroller & various other peripherals such as parallel port.
5. Timers – A microcontroller may be in-built with one or more timer or counters. The timers & counters control all counting & timing operations within a microcontroller. Timers are employed to count external pulses. The main operations performed by timers are- pulse generations, clock functions, frequency measuring, modulations, making oscillations, etc.
6. ADC (Analog to digital converter) – ADC is employed to convert analog signals to digital ones. The input signals need to be analog for ADC. The digital signal production can be employed for different digital applications (such as- measurement gadgets).

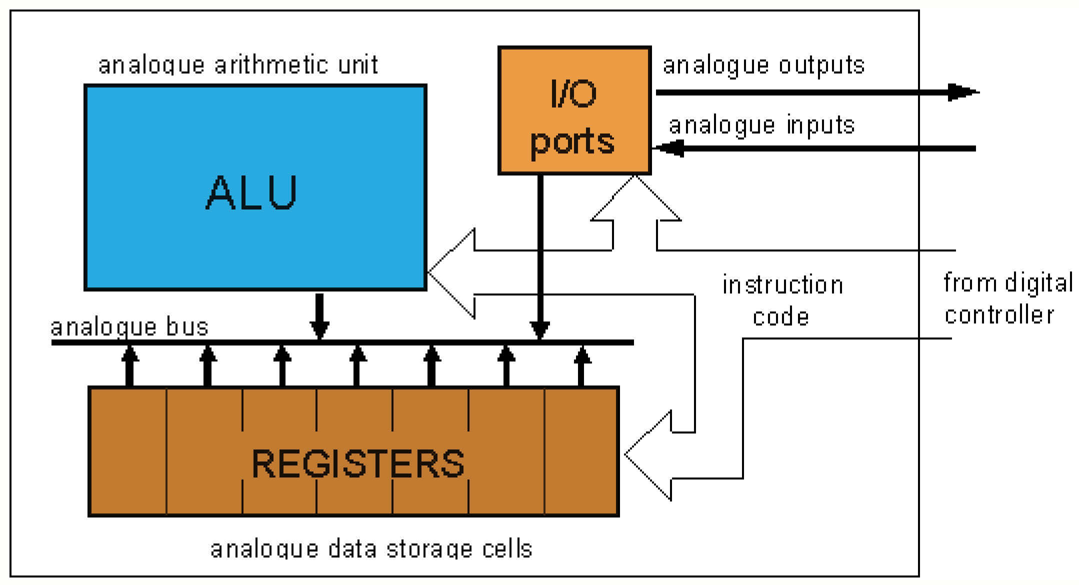
7. DAC (digital to analog converter) – this converter executes opposite functions that ADC perform. This device is generally employed to supervise analog appliances like- DC motors, etc.
8. Interpret Control- This controller is employed for giving delayed control for a working program. The interpret can be internal or external.
9. Special Functioning Block – Some special microcontrollers manufactured for special appliances like- space systems, robots, etc, comprise of this special function block. This special block has additional ports so as to carry out some special operations.

A PIC microcontroller architecture is shown below:



3.5.3 How Is a Microcontroller Different from A Microprocessor?

Microprocessor has only a CPU inside them in one or few Integrated Circuits. Unlike microcontrollers it does not have RAM, ROM and other peripherals. They are dependent on external circuits of peripherals to work. Microprocessors are not made for specific task but they are required where tasks are complex and tricky like development of software's, games and other applications that require high memory and where input and output are not defined. It may be called heart of a computer system. Some examples of microprocessor are Pentium, I3, and I5 etc.



From this image of architecture of microprocessor, it can be easily seen that it have registers and ALU as processing unit and it does not have RAM, ROM in it.

The notable differences are listed below:

1. Key difference in both of them is presence of external peripheral, where microcontrollers have RAM, ROM, EEPROM embedded in it while we have to use external circuits in case of microprocessors.

2. As all the peripheral of microcontroller are on single chip it is compact while microprocessor is bulky.

3. Microcontrollers are made by using complementary metal oxide semiconductor technology so they are far cheaper than microprocessors. In addition, the applications made with microcontrollers are cheaper because they need lesser external components, while the overall cost of systems made with microprocessors are high because of the high number of external components required for such systems.

4. Processing speed of microcontrollers is about 8 MHz to 50 MHz, but in contrary processing speed of general microprocessors is above 1 GHz so it works much faster than microcontrollers.

5. Generally microcontrollers have power saving system, like idle mode or power saving mode so overall it uses less power and also since external components are low overall consumption of power is less. While in microprocessors generally there is no power saving system and also many external components are used with it, so its power consumption is high in comparison with microcontrollers.

6. Microcontrollers are compact so it makes them favorable and efficient system for small products and applications while microprocessors are bulky so they are preferred for larger applications.

7. Tasks performed by microcontrollers are limited and generally less complex. While task performed by microprocessors are software development, Game development, website, documents making etc. which are generally more complex so require more memory and speed so that's why external ROM, RAM are used with it.

As Microcontrollers, you can classify microprocessors according to their instruction set as follows

There are basically 5 kinds of microprocessors namely:

- **Complex Instruction Set Microprocessors:** They are also called as CISM in short and they categorize a microprocessor in which orders can be executed together along with other low level activities. It mainly performs the task of uploading, downloading and recalling data into and from the memory card. Apart from that it also does complex mathematical calculations within a single command.
- **Reduced Instruction Set Microprocessor:** This processor is also called as RISC. These kinds of chips are made according to the function in which the microprocessor can carry out small things within a particular command. In this way it completes more commands at a faster rate.
- **Superscalar Processors:** This is a processor that copies the hardware on the microprocessor for performing numerous tasks at a time. They can be used for arithmetic and as multipliers. They have several operational units and thus carry out more than a one command by constantly transmitting various instructions to the superfluous operational units inside the processor.
- **The Application Specific Integrated Circuit:** This processor is also known as ASIC. They are used for specific purposes that comprises of automotive emissions control or personal digital assistant computer. This kind of processor is made with proper specification but apart from that it can also be made using the off the shelf gears.
- **Digital Signal Multiprocessors:** Also called as DSP's, these are used for encoding and decoding videos or to convert the digital and video to analog and analog to digital. They need a microprocessor that is excellent in mathematical calculations. The chips of this processor are employed in SONAR, RADAR, home theaters audio gears, Mobile phones and TV set top boxes.

Now we go back to our discussion on microcontrollers and we will focus on AVR microcontrollers.

3.5.4 AVR Microcontrollers

AVR also known as Advanced Virtual RISC, is a customized Harvard architecture 8-bit RISC solitary chip micro-controller. It was invented in the year 1966 by Atmel. Harvard architecture signifies that program & data are amassed in different spaces and are used simultaneously. It was one of the foremost micro-controller families to employ on-chip flash memory basically for storing program, as contrasting to one-time programmable EPROM, EEPROM or ROM, utilized by other micro-controllers at the same time. Flash memory is a non-volatile (constant on power down) programmable memory.

The SRAM, Flash and EEPROM all are incorporated on a single chip, thereby eliminating the requirement of any other external memory in maximum devices. Several appliances comprise of parallel external bus alternative, so as to add extra data memory gadgets. Approximately all appliances, except TinyAVR chips comprise serial interface, which is used to link large serial Flash & EEPROMs chips.

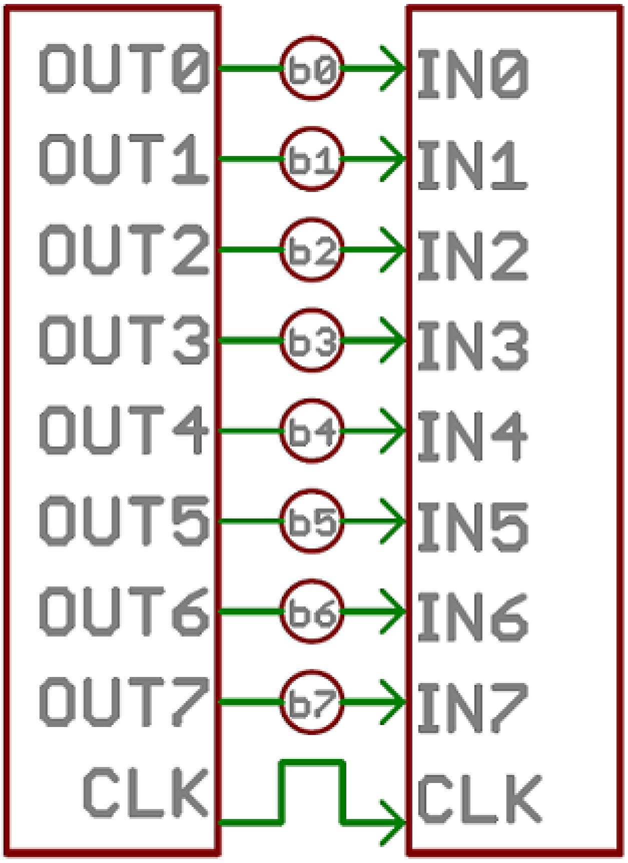
AVR microcontrollers find many applications as embedded systems; they are also used in the Arduino line of open source board designs. And now we can begin our discussion of the Arduino Platform, but before then, we need to discuss interfacing.

3.5.5 Interfacing

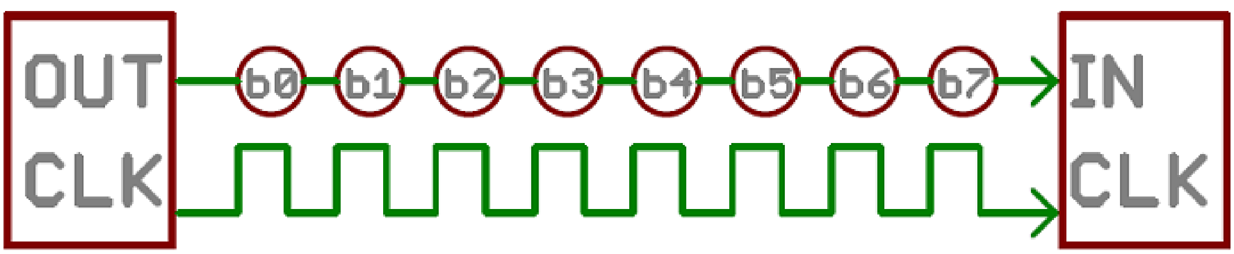
Embedded electronics is all about interlinking circuits (processors or other integrated circuits) to create a symbiotic system. In order for those individual circuits to swap their information, they must share a common communication protocol. Hundreds of communication protocols have been defined to achieve this data exchange, and, in general, each can be separated into one of two categories: parallel or serial.

3.5.5.1 Parallel vs. Serial

Parallel interfaces transfer multiple bits at the same time. They usually require buses of data - transmitting across eight, sixteen, or more wires. Data is transferred in huge, crashing waves of 1’s and 0’s.



Serial interfaces stream their data, one single bit at a time. These interfaces can operate on as little as one wire, usually never more than four.



Parallel communication certainly has its benefits. It's fast, straightforward, and relatively easy to implement. But it requires many more input/output (I/O) lines

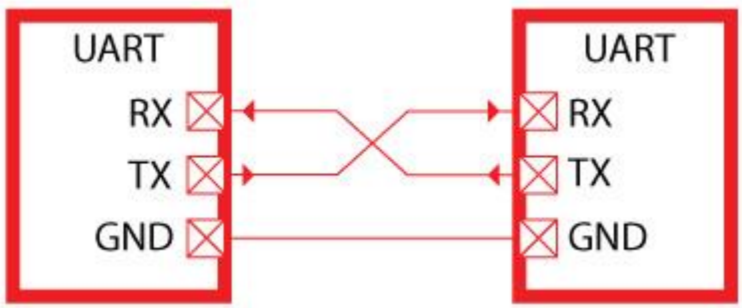
3.5.5.2 Synchronous Vs Asynchronous Serial Communication

Over the years, dozens of serial protocols have been crafted to meet particular needs of embedded systems. USB (universal serial bus), and Ethernet, are a couple of the more well-known computing serial interfaces. Other very common serial interfaces include SPI, I2C and UART. Each of these serial interfaces can be sorted into one of two groups: synchronous or asynchronous.

Asynchronous means that data is transferred without support from an external clock signal. This transmission method is perfect for minimizing the required wires and I/O pins, but it does mean we need to put some extra effort into reliably transferring and receiving data. UART is the most common form of asynchronous transfers.

3.5.5.2.1 UART

UART stands for Universal Asynchronous Receiver/Transmitter and is really just a fancy way of referring to a serial port. It is really easy to understand as it only requires two lines: a transmission line (TX) and a receiving line (RX).



UART transmissions begin with a start bit where the appropriate line (TX or RX) is pulled low by the sending party. Then 5 to 8 data bits are sent.

Following the data, an optional parity bit is sent, followed by 1 or 2 stop bits, where the sending module pulls the pin high.

For this protocol to work, the sender and receiver have to agree on a few things.

1. How many data bits are sent with each packet (5 to 8)?
2. How fast should the data be sent? This is known as the baud rate.
3. Is there a parity bit after the data, and is it high or low?
4. How many stop bits will be sent at the end of each transmission?

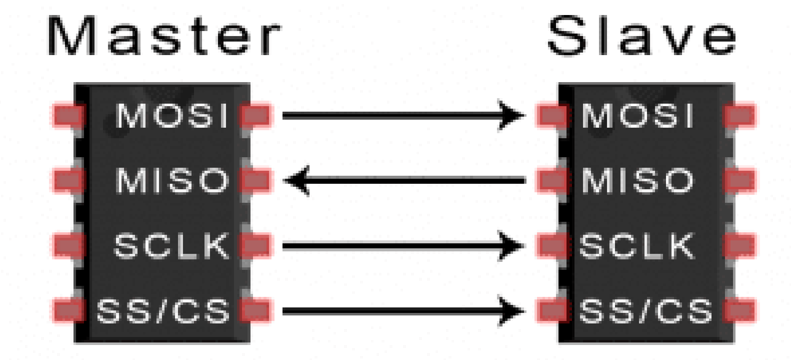
A synchronous serial interface always pairs its data line(s) with a clock signal, so all devices on a synchronous serial bus share a common clock. This makes for a more straightforward, often faster serial transfer, but it also requires at least one extra wire between communicating devices. Examples of synchronous interfaces include SPI, and I2C.

3.5.5.2.2 SPI

SPI stands for Serial Peripheral Interface. SPI is a common communication protocol used by many different devices. For example, SD card modules, RFID card reader modules, and 2.4 GHz wireless transmitter/receivers all use SPI to communicate with microcontrollers.

One unique benefit of SPI is the fact that data can be transferred without interruption. Any number of bits can be sent or received in a continuous stream. With I2C and UART, data is sent in packets, limited to a specific number of bits. Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

Devices communicating via SPI are in a master-slave relationship. The master is the controlling device (usually a microcontroller), while the slave (usually a sensor, display, or memory chip) takes instruction from the master. The simplest configuration of SPI is a single master, single slave system, but one master can control more than one slave (more on this below).



MOSI (Master Output/Slave Input) – Line for the master to send data to the slave.

MISO (Master Input/Slave Output) – Line for the slave to send data to the master.

SCLK (Clock) – Line for the clock signal.

SS/CS (Slave Select/Chip Select) – Line for the master to select which slave to send data to.

Advantages of SPI

- No start and stop bits, so the data can be streamed continuously without interruption
- No complicated slave addressing system like I2C
- Higher data transfer rate than I2C (almost twice as fast)
- Separate MISO and MOSI lines, so data can be sent and received at the same time

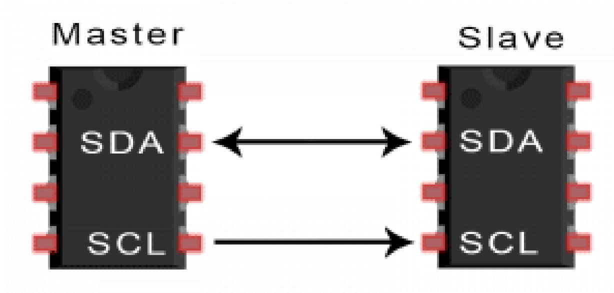
Disadvantages of SPI

- Uses four wires (I2C and UARTs use two)
- No acknowledgement that the data has been successfully received (I2C has this)

- No form of error checking like the parity bit in UART
- Only allows for a single master

3.5.5.2.3 I2C

I2C stands for Inter-Integrated Circuit and is pronounced “I squared C”, “I two C” or “I-I-C”. I2C is a protocol that allows one device to exchange data with one or more connected devices through the use of a single data line and clock signal.I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.



Like UART communication, I2C only uses two wires to transmit data between devices:

SDA (Serial Data) – The line for the master and slave to send and receive data.

SCL (Serial Clock) – The line that carries the clock signal.

I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line).

Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master. Since multiple slave devices can use the same SDA line, the master needs a way to distinguish between them and talk to a single device at a time. The I2C protocol uses the concept of device addressing to coordinate traffic on the data line.

advantages of I2C

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol

disadvantages of I2C

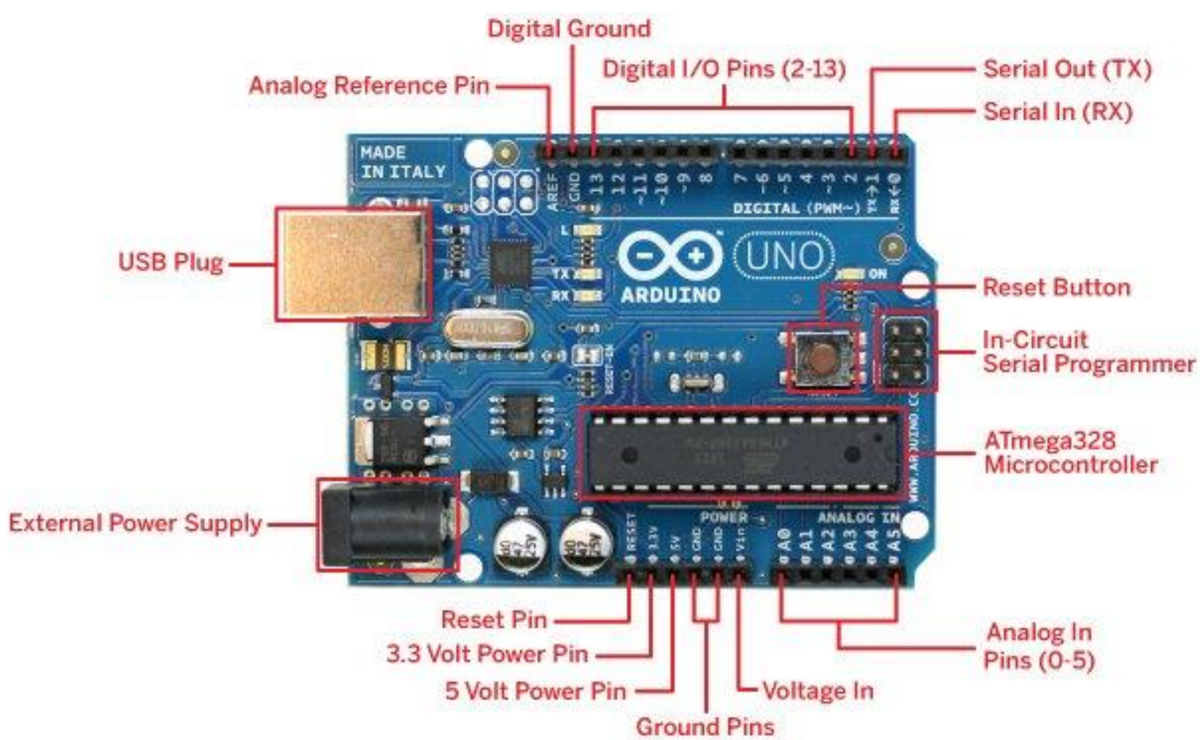
- Slower data transfer rate than SPI
- The size of the data frame is limited to 8 bits
- More complicated hardware needed to implement than SPI

4 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Most Arduino boards are based on AVR microcontrollers. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language , and the Arduino Software (IDE).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.



4.1 Advantages of Arduino

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

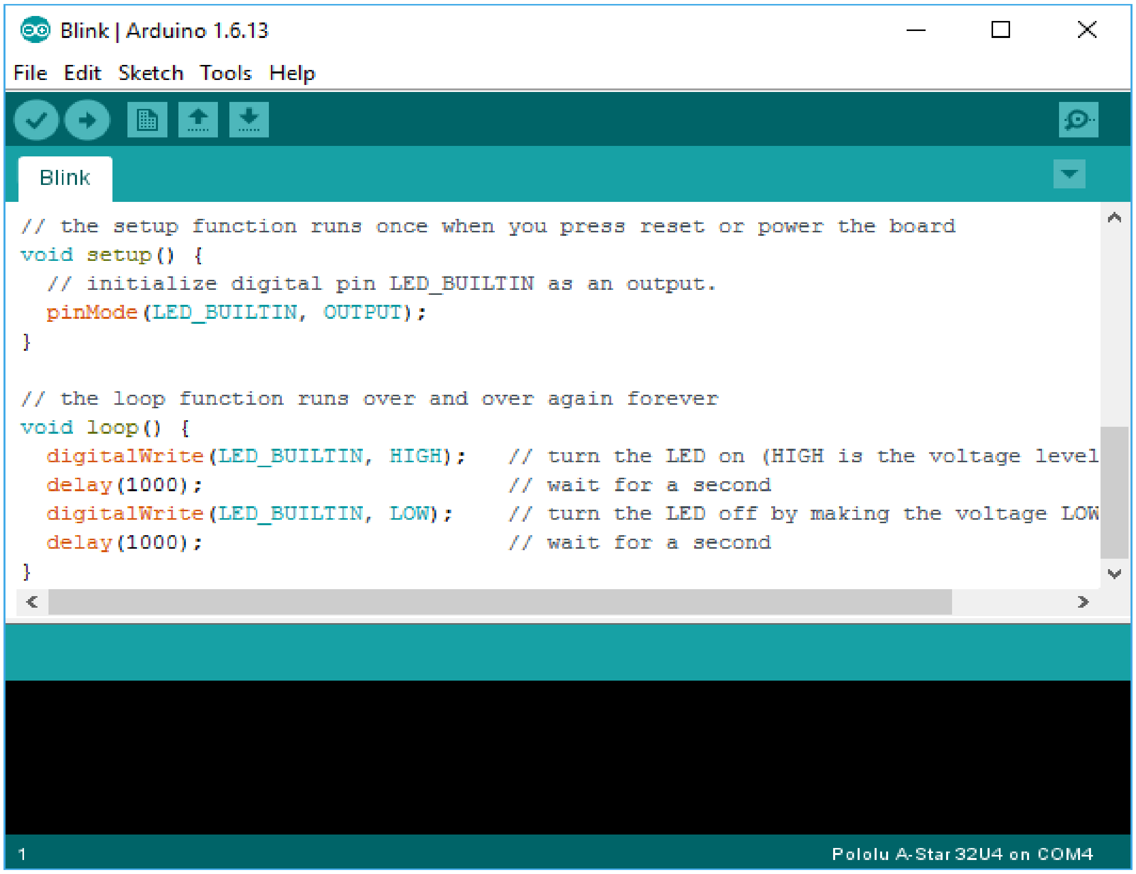
Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

4.2 Arduino IDE

An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications.

Arduino provides an easy to use IDE to program Arduino and Arduino compatible boards.



4.2.1 Arduino Libraries

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. In programming, a library is a collection of precompiled routines that a program can use. The routines, sometimes called modules, are stored in object format. Libraries are particularly useful for storing frequently used routines because you do not need to explicitly link them to every program that uses them.

Standard Libraries

- EEPROM- reading and writing to "permanent" storage
- Ethernet / Ethernet 2 - for connecting to the internet using the Arduino Ethernet Shield, Arduino Ethernet Shield 2 and Arduino Leonardo ETH
- Firmata- for communicating with applications on the computer using a standard serial protocol.
- GSM - for connecting to a GSM/GRPS network with the GSM shield.
- LiquidCrystal - for controlling liquid crystal displays (LCDs)

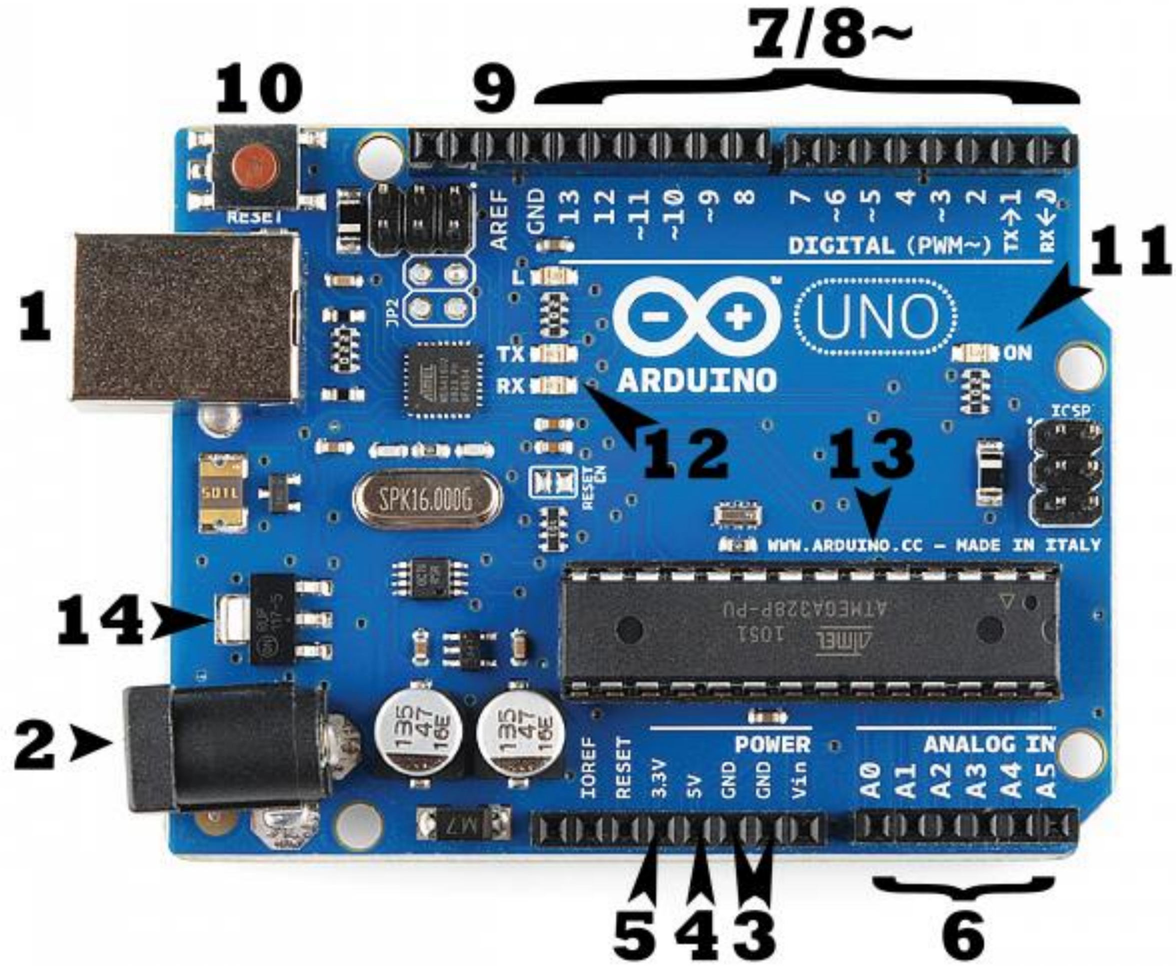
- SD - for reading and writing SD cards
- Servo - for controlling servo motors
- SPI - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- SoftwareSerial - for serial communication on any digital pins.
- Stepper - for controlling stepper motors
- TFT - for drawing text , images, and shapes on the Arduino TFT screen
- WiFi - for connecting to the internet using the Arduino WiFi shield
- Wire - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

4.3 Arduino Boards

This is the hardware component of Arduino. Some of the most popular boards are listed below.

-Arduino / Genuino MKR1000	-Arduino 101/Genuino 101
-Arduino Zero	-Arduino Due
-Arduino Yún	-Arduino Leonardo
-Arduino Uno	-Arduino Mega2560
-Arduino Ethernet	-Arduino Fio
-Arduino Nano	-Arduino Mega ADK

Let us look at the parts and functions of one of the most popular Arduino boards, the Uno.



Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2).

The USB connection is also how you will load code onto your Arduino board.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM), for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Arduino has a reset button (10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’ (11). This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit (13). Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

Voltage Regulator

The voltage regulator (14) is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

Now you have all the necessary knowledge you need to start programming the Arduino. The next section deals with Arduino programming language.