

Unit 1: Introduction to Microcontroller

Introduction to Arduino: Microcontrollers used in Arduino, Pin configuration and architecture, Concept of digital and analog ports.

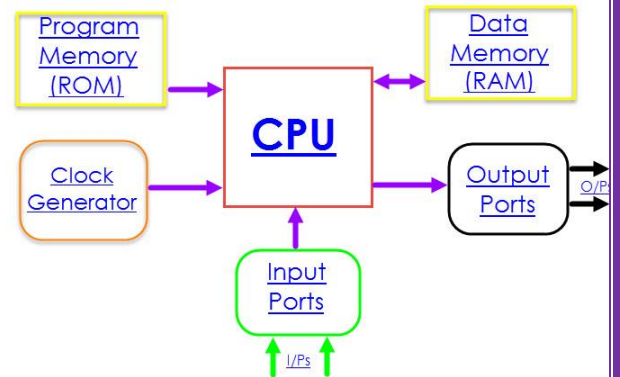
What is a Microcontroller?

A microcomputer made on a single semiconductor chip is called single-chip microcomputer. Since, single chip microcomputers are generally used in control applications, they are also called microcontrollers.

Microcontroller contains all essential components of a microcomputer such as CPU, RAM, ROM/EPROM, I/O ports, Timers/Counters, Clock generator circuitry, Interrupt circuit, Accumulator, Program Counter, Stack Pointer, and General-Purpose Registers etc.

A microcontroller is denoted with μC or uC . The fabrication technology used for its controller is VLSI. An alternate name of the microcontroller is the embedded controller. At present, there are different microcontrollers types that are existing in the market like 4-bit, 8-bit, 64-bit & 128-bit. It is a compressed microcomputer used to control the embedded system functions in robots, office machines, motor vehicles, home appliances & other electronic gadgets.

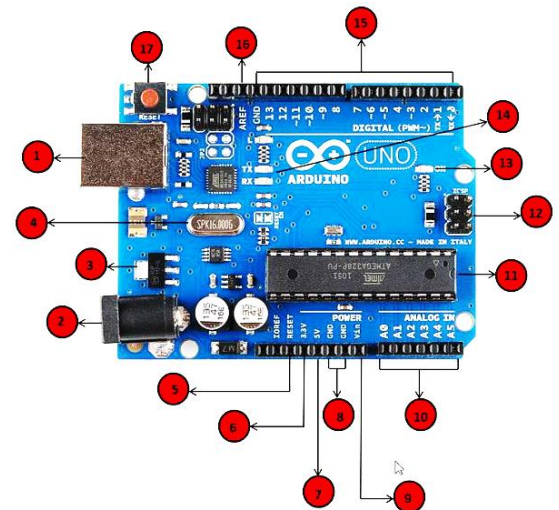
The following image shows the basic components of a Microcontroller. As all the components (and a few other components) are integrated on a single chip (Integrated Circuit – IC), a Microcontroller can be considered as a Microcomputer (or a Computer on – chip).
















Defining Arduino

Arduino is an open-source prototyping platform in electronics based on easy-to-use hardware and software. Subtly speaking, Arduino is a microcontroller based prototyping board which can be used in developing digital devices that can read inputs like finger on a button, touch on a screen, light on a sensor etc. and turning it in to output like switching on an LED, rotating a motor, playing songs through a speaker etc.

The Arduino board can be programmed to do anything by simply programming the microcontroller on board using a set of instructions for which, the Arduino board consists of a USB plug to communicate with your computer and a bunch of connection sockets that can be wired to external devices like motors, LEDs etc. The aim of Arduino is to introduce the world of electronics to people who have small to no experience in electronics like hobbyists, designers, artists etc.



| | |
|---|--|
|  | <p>Power USB Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).</p> |
|  | <p>Power (Barrel Jack) Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).</p> |
|  | <p>Voltage Regulator The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.</p> |
|  | <p>Crystal Oscillator The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. It's frequency is 16 MHz.</p> |
|  | <p>Arduino Reset You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).</p> |
|  | <p>Pins (3.3, 5, GND, Vin)</p> <ul style="list-style-type: none"> • 3.3V (6) – Supply 3.3 output volt • 5V (7) – Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) – There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) – This pin also can be used to power the Arduino board from an external power source, like AC mains power supply. |
|  | <p>Analog pins The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.</p> |
|  | <p>Main microcontroller Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.</p> |
|  | <p>ICSP pin Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial</p> |

| | |
|--|--|
| | Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus. |
|  | Power LED indicator This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection. |
|  | TX and RX LEDs On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process. |
|  | Digital I/O The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM. |
|  | AREF AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. |

Board Types

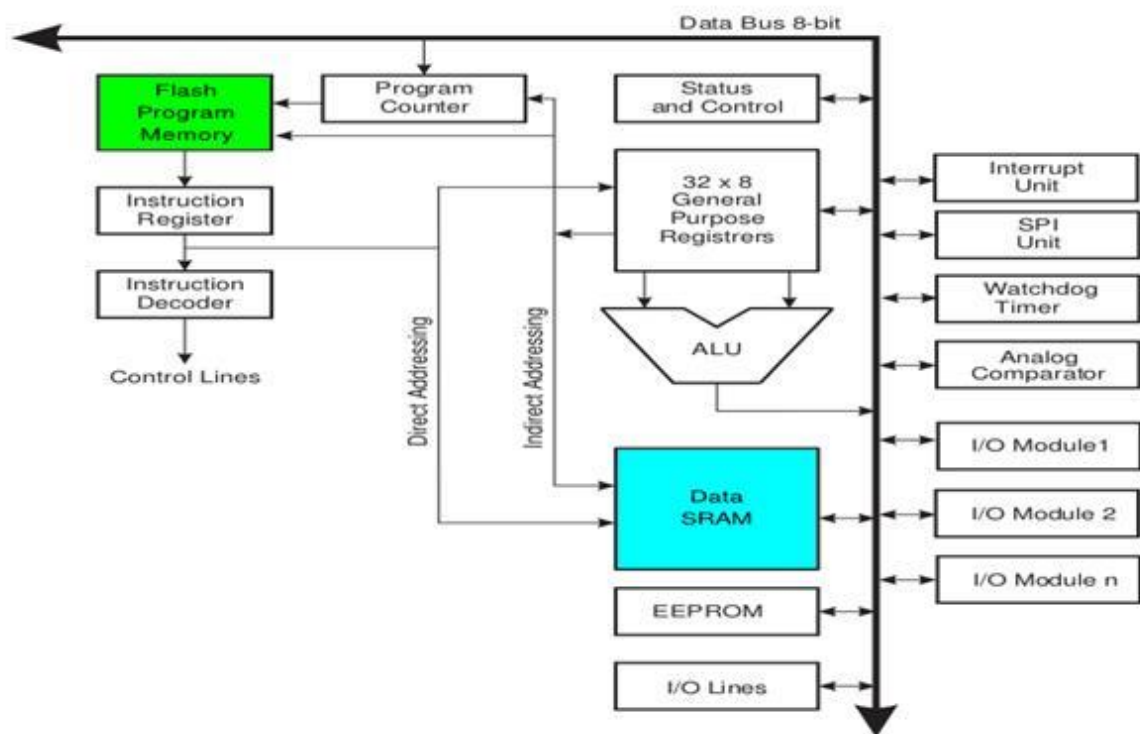
Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

| Board Name | Operating Volt | Clock Speed | Digital i/o | Analog Inputs | PWM | UART |
|--|----------------|-------------|-------------|---------------|-----|------|
| Arduino boards based on ATMEGA328 microcontroller | | | | | | |
| Arduino Uno R3 | 5V | 16MHz | 14 | 6 | 6 | 1 |
| Arduino mini 05 | 5V | 16MHz | 14 | 8 | 6 | 1 |
| Arduino Nano | 5V | 16MHz | 14 | 8 | 6 | 1 |
| LilyPad Arduino simple board | 3.3V | 8MHz | 9 | 4 | 5 | 0 |

| Arduino boards based on ATMEGA32u4 microcontroller | | | | | | |
|---|------|-------|----|----|----|---|
| Arduino Leonardo | 5V | 16MHz | 20 | 12 | 7 | 1 |
| Pro micro 5V/16MHz | 5V | 16MHz | 14 | 6 | 6 | 1 |
| Arduino boards based on ATMEGA2560 microcontroller | | | | | | |
| Arduino Mega 2560 R3 | 5V | 16MHz | 54 | 16 | 14 | 4 |
| Arduino boards based on AT91SAM3X8E microcontroller | | | | | | |
| Arduino Duo | 3.3V | 84MHz | 54 | 12 | 12 | 4 |

Arduino Architecture:



Arduino's processor basically uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories- Program memory and the data memory. The code is stored in the flash program memory, whereas the data is stored in the data memory. The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader), 2 KB of SRAM and 1 KB of EEPROM and operates with a clock speed of 16MHz.

- **Flash Program Memory:** The ATmega328P contains 32Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 16K x 16.
- **General Purpose I/O Registers:** The ATmega328P contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags.

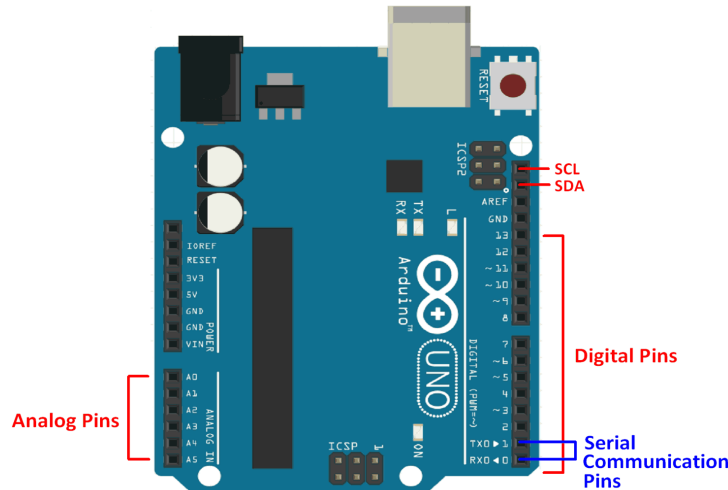
- **I/O Memory (64 x 8-bit) (002016–005F16):** Contains addressable space for peripheral control registers and other I/O registers
- **Extended I/O Memory (device dependent) (160 x 8-bit) (006016–00FF16):** Some AVR microcontrollers with more peripherals need more space than the I/O memory can address so some of the SRAM is used as Extended I/O memory to handle the extra peripherals control registers and other I/O functions
- **Internal SRAM (2048 x 8-bit) (010016–08FF16):** SRAM is used for temporarily storing intermediate results and variables, stack within a software application.
- **EEPROM Data Memory:** The ATmega328P contains 1Kbytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written.

Features of Arduino Uno Board

The features of Arduino Uno ATmega328 includes the following.

- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- The input voltage ranges from 6v to 20V
- Digital input/output pins are 14
- Analog input pins are 6
- DC Current for each I/O pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16 MHz

Concept of digital and analog ports.:



Digital I/O Pins: *UNO has 14 digital input – output (I/O) pins* which can be used as either input or output by connecting them with different external devices and components. Out of these 14 pins, 6 pins are capable of producing PWM signal. All the digital pins operate at 5V and can output a current of 20mA.

The digital inputs and outputs (digital I/O) on the Arduino are what allow you to connect the Arduino sensors, actuators, and other ICs. Learning how to use them will allow you to use the Arduino to do some really useful things, such as reading switch inputs, lighting indicators, and controlling relay outputs.

Digital signals: Unlike analog signals, which may take on any value within a range of values, digital signals have two distinct values: HIGH (1) or LOW (0). digital signals can be used in situations where the input or output will have one of those two values. For example, one way that you might use a digital signal is to turn an LED on or off.

Functions: The Arduino functions associated with digital are:

pinMode()

digitalRead()

digitalWrite()

pinMode (pin_number, mode)

analogWrite (value)

Because the Arduino digital I/O pins can be used for either input or output, you should first configure the pins you intend to use for digital I/O with this function. pin is the number of the pin you wish to configure. mode must be one of three values: INPUT, OUTPUT, or INPUT_PULLUP. When mode is set to INPUT_PULLUP, a 20 kΩ pullup resistor is internally connected to the pin to force the input HIGH if there is nothing connected to the pin.

digitalWrite(pin_number,value)

This function writes a digital value to a pin. pin specifies which Arduino pin the digital value will be written to, and value is the digital value to which the pin is set. value

must be either HIGH or LOW.

digitalRead(pin_number)

This function reads a digital value from a pin. pin is the number of the digital I/O pin you want to read. This function returns one of two values: HIGH or LOW.

What is PWM?

PWM stands for Pulse Width Modulation and it is a technique used in controlling the brightness of LED, speed control of DC motor, controlling a servo motor or where you have to get analog output with digital means.

The Arduino digital pins either gives us 5V (when turned HIGH) or 0V (when turned LOW) and the output is a square wave signal. So if we want to dim a LED, we cannot get the voltage between 0 and 5V from the digital pin but we can change the ON and OFF time of the signal. If we will change the ON and OFF time fast enough then the brightness of the led will be changed.

Before going further, let's discuss some terms associated with PWM.

TON (On Time): It is the time when the signal is high.

TOFF (Off Time): It is the time when the signal is low.

Period: It is the sum of on time and off time.

Duty Cycle: It is the percentage of time when the signal was high during the time of period.

So, at 50% duty cycle and 1Hz frequency, the led will be high for half a second and will be low for the other half second. If we increase the frequency to 50Hz (50 times ON and OFF per second), then the led will be seen glowing at half brightness by the human eye.

The Arduino IDE has a built in function “*analogWrite()*” which can be used to generate a PWM signal. The frequency of this generated signal for most pins will be about 490Hz and we can give the value from 0-255 using this function.

analogWrite(0) means a signal of 0% duty cycle.

analogWrite(127) means a signal of 50% duty cycle.

analogWrite(255) means a signal of 100% duty cycle.

On Arduino Uno, the PWM pins are 3, 5, 6, 9, 10 and 11. The frequency of PWM signal on pins 5 and 6 will be about 980Hz and on other pins will be 490Hz. The PWM pins are labeled with ~ sign.

Analog Input Pins

The ATmega controllers used for the Arduino contain an onboard 6 pins (A0 to A5). The ADC has 10 bit resolution, returning integers from 0 to 1023. While the main function of the analog pins for most Arduino users is to read analog sensors, the analog pins also have all the functionality of general purpose input/output (GPIO) pins (the same as digital pins 0 - 13).

analogRead(pin): Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023. On an Arduino UNO, for example, this yields a resolution between readings of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit.