

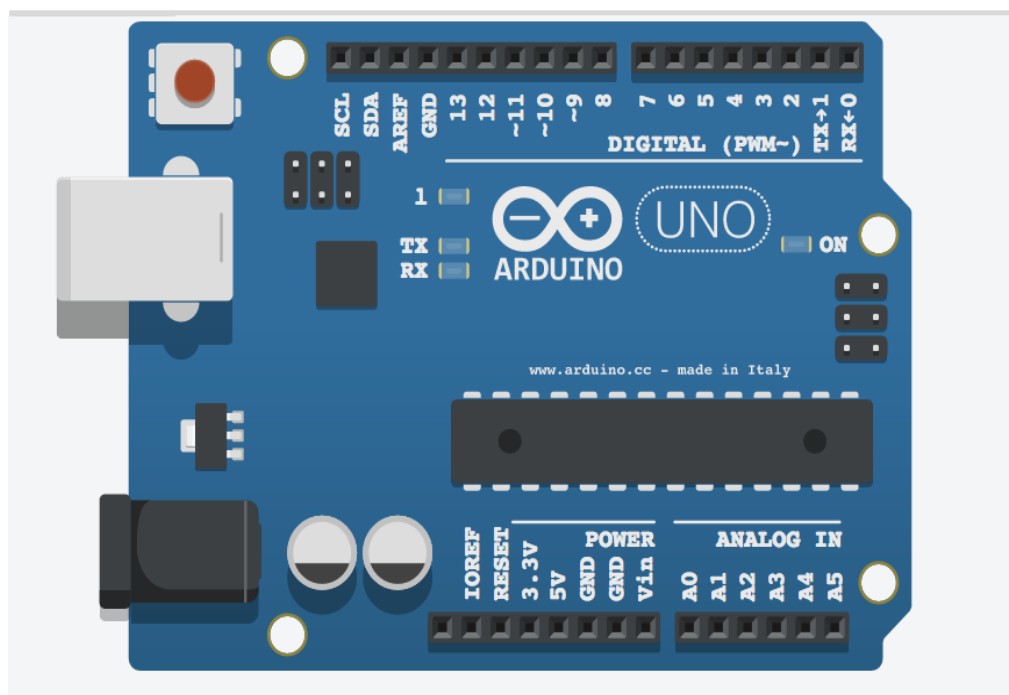
## Experiment No: 01

**Experiment Name: Introduction to Arduino.**

## Introduction:

**Arduino** is a very small or pocket size microcontroller like computer. Arduino is an easy-to use electronics platform that combines hardware and software. It includes microcontroller boards that you can program to control lights, motors, and sensors. Arduino boards come with different types of microprocessors and controllers. They have digital and analog input/output pins that can connect to expansion boards called "shields" or to breadboards for prototyping. This makes it easy to add new features or connect other circuits to your projects.

The microcontrollers can be programmed using the C and C++ programming language, using a standard API which is also known as the Arduino Programming Language. With Arduino, we can create interactive projects like robots or home automation systems. It's popular among beginners and experienced users because it's user-friendly and has a lot of community support. With its help many science based projects can be created very easily. Whether we're just starting out or building advanced projects, Arduino helps us to implement our ideas to life.



**Figure No: 01: Picture of Arduino.**

## Arduino Pin Description:

**Reset:** This pin in the board is used to reset the microcontroller.

**3V3:** This pin on the board supplies 3.3V, generated by a voltage regulator.

**VIN:** This input voltage pin connects Arduino with an external power supply which range is 7 to 12 v.

**GND:** This pin is used to connect the Arduino to ground.

**Analog Pins:** the pins A0 to A5 is used as analog input.

**Digital Pins:** the pins 0 to 13 is used as digital input.

**Serial Pins:** These pins are also known as a UART pin. These pins are helps to connect Arduino with computers or other devices. Here **D1** is the transmitter pin and **D0** is the receiver pin which is used to transmit and receive data.

**External Interrupt Pins:** This pin is used to produce the External interrupt and it is done by pin numbers 2 and 3.

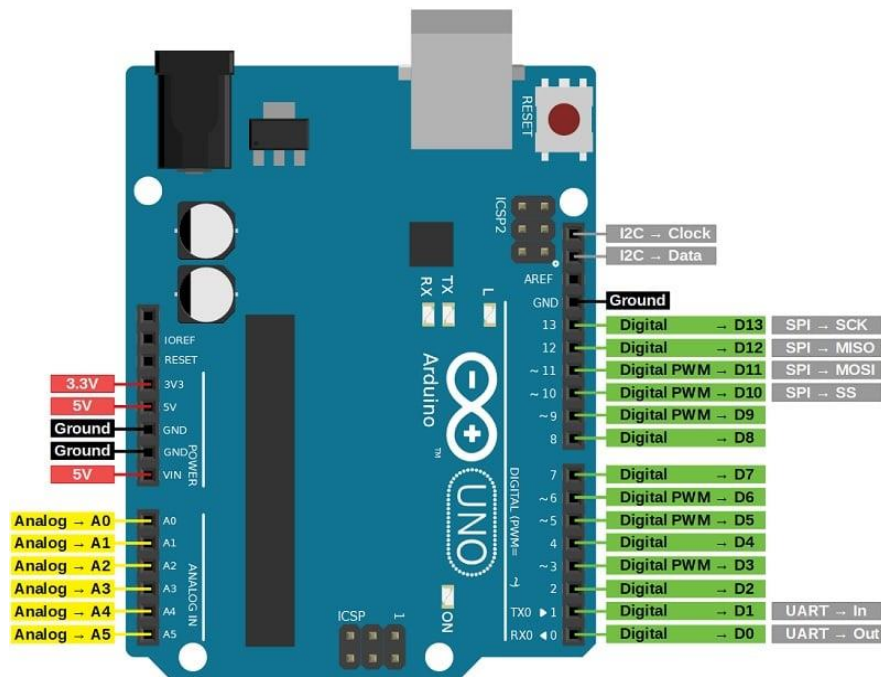
**PWM (Pulse Width Modulation) pins:** PWM pins on the Arduino convert digital signals into analog by changing the pulse width. Pins 3, 5, 6, 9, 10, and 11 are used for PWM.

**SPI (Serial Peripheral Interface) pins:** it is used to maintain SPI communication with the help of the SPI library. SPI pins include:

- **SS (Slave Select)** - Pin number 10
- **MOSI (Master Out Slave In)** - Pin number 11
- **MISO (Master In Slave Out)** - Pin number 12
- **SCK (Serial Clock)** - Pin number 13

**LED :** The board has a built-in LED connected to digital pin 13, which lights up when the pin is set to high.

**AREF:** The AREF pin on the Arduino board is an analog reference pin used to provide a reference voltage from an external power supply.



**Figure No: 02: Pin diagram of Arduino.**

## Why do we need Arduino:

Arduino is a versatile component in Computer Engineering and Electrical Engineering sectors which is widely used for a variety of projects in electronics and programming. Arduino is very helpful for beginners and professionals to implement their project idea easily like Home automation, Robotics, Wearable Technology, Educational Teaching, IoT, Data Logging, 3D Mapping & Printing, Automated Gardening .We need arduino for several reasons-

- Arduino is very easy to use. It makes it simple to create and program electronic projects, even for beginners.

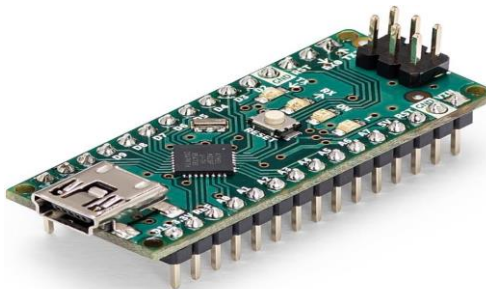
- Arduino can be utilized for a variety of projects, like basic LED blinkers to intricate robotics system.
- Arduino can interact with various sensors and actuators, making it ideal for interactive projects.
- It is great for quickly prototyping ideas and testing concepts before building final products.
- Arduino is open-source, meaning users can share and modify designs and code freely.

Overall, it is a powerful tool for learning and creating in electronics and programming.

### **Different Types of Arduino:**

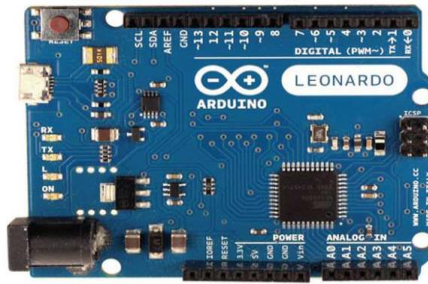
**Arduino UNO:** It is open source based on the **ATmega328P** microcontroller. It is a commonly used Arduino board and the best option for making projects and programming different components. It has 14 pins.

**Arduino NANO:** Arduino NANO is small in size making it perfect for breadboards and tight spaces, and it has much common features as Arduino UNO. It is most powerful for that developers and users like it most.



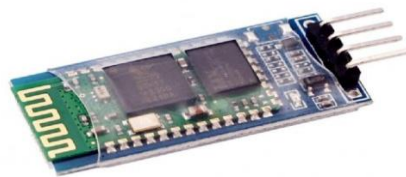
**Figure No: 03: Arduino NANO.**

**Arduino Leonardo:** Arduino Leonardo is different from other boards. It uses an **ATmega32U4** chip in place of **ATmega328P**. It offers more pins for IO (20), PWM (7), and analog input signals (12).



**Figure No: 04: Arduino Leonardo.**

**Arduino Bluetooth:** Arduino Bluetooth is one of the most popular types of Arduino boards. This Arduino board is powered by the **ATmega168** microcontroller. It has 16 digital pins, 06 analog pins and a 16 MHz crystal oscillator reset button.



**Figure No 05: Arduino Bluetooth.**

### **Other types of Arduino Boards:**

- I. Arduino Micro
- II. Arduino Due
- III. LilyPad Arduino Board
- IV. Arduino NANO 33 BLE
- V. Arduino Diecimila
- VI. RedBoard Arduino Board
- VII. Arduino MKR GSM 1400.etc

Arduino program can be divided in three main parts:

- Functions
- Values
- Structures

## Arduino Functional Keywords:

- **setup() and loop():**

A sketch is the complete program that runs on an Arduino board. It must contain at least two main functions:

**setup() function:** The **setup()** function contains initialization of the libraries, variables used for the code.

**loop() function:** The **loop()** function keeps on repeating the instructions and actively controls and monitors the Arduino.

### **Structure:**

```
void setup(){
```

//The setup block contains the “settings” and runs just once when: The Arduino is reset/restarted or turned on and new sketch is uploaded

```
// put setup code here, to run once.  
  
}
```

```
void loop(){
```

```
// put your main code here, to run repeatedly.
```

//the loop block contains the main code and run it continuously after the setup function has been executed.

}

- **Serial.begin ()**

The serial.begin( ) sets the baud rate for serial data communication. Here baud means the data rate in bits per second. The default baud rate in Arduino is 9600 bps. we can use this function in program using two formats.

begin( speed )

begin( speed, config)

---

**Example:**

```
void setup ( )
{
  Serial.begin(4800); //4800 is the baud rate
}                      //sets the stop, parity and data bits.
void loop ( )
{
}
```

- **Serial.print ()**

The serial.print() function prints the data to serial port. The ASCII (American Standard Code for Information Interchange) format is a human-readable text in which the printed data are stored. Print ASCII character for each digit of a number. The Serial.print ( ) is defined as two types:

print( value )

print( value, format)

**Example:**



```
Serial.print("Hello World") //Hello World is value
```

Output: Hello World

```
Serial.print(25, BIN) // Here 25 is value and BIN is format
```

Output: 11001

- **Serial.println()**

The Serial.println() function sends data from the Arduino to a connected device, converting it into a string format and automatically adding a newline character. It's commonly used for debugging and monitoring output in the Serial Monitor.

**Example:**

```
void setup() {  
  
  Serial.begin(9600);    // Initialize serial communication at 9600 bits per second  
  
}  
  
void loop() {  
  
  Serial.println("Hello Arduino!");    // Print a string  
  
  Serial.println(123);                // Print an integer  
  
  delay(1000);                        // Wait for 1 second  
  
}
```

- **digitalRead() :**

The digitalRead() function in Arduino is used to read the state of a digital pin, which can be either HIGH or LOW.

HIGH: ON-Power Supply

LOW: OFF-Ground Connect

**Example:**

```
const int buttonPin = 2; // Pin for the button

int buttonState = 0; // Variable to hold the button state

void setup() {

  pinMode(buttonPin, INPUT); // Set the button pin as an input

}

void loop() {

  buttonState = digitalRead(buttonPin); // Read the button state

}
```

- **Serial.read()**

The Serial.read() function in Arduino reads incoming serial data. It returns the first byte of data as an int. If no data is available, it returns -1.

```
void setup() {

  Serial.begin(9600); // Start serial communication

}

void loop() {

  if (Serial.available() > 0) {

    char incomingChar = Serial.read();    // Read the character

    Serial.print("You typed: ");          // Print a message

    Serial.println(incomingChar);          // Show the character

  }

}
```

```
}
```

- `delay()`

The `delay()` function in Arduino is used to pause the program for a specified amount of time. It takes one argument, which is the duration of the delay in milliseconds.

```
delay(milliseconds);
```

## Conclusion:

Arduino makes working with electronics easy and fun for everyone. We can create all sorts of projects, from simple gadgets to complex robots. The hardware includes a microcontroller, and the software helps us write and upload code. Arduino is perfect for learning and building innovative projects.