

Table of Contents



- [1. What is Arduino?](#)
- [2. Setting Up Arduino IDE](#)
- [3. Basic Structure of Arduino Code](#)
- [4. Uploading Code](#)
- [5. Pin Modes](#)
- [6. Digital I/O](#)
- [7. Analog I/O](#)
 - [Analog Read \(0-1023\):](#)
 - [Analog Write \(PWM, 0-255\):](#)
- [8. Serial Communication](#)
- [9. Control Structures](#)
 - [If-Else Statement:](#)
 - [For Loop:](#)
 - [While Loop:](#)
- [10. Common Functions](#)
- [11. Sensors and Modules](#)
 - [LED Blinking \(Basic Example\):](#)
 - [Button Input:](#)
 - [Temperature Sensor \(LM35\):](#)
- [12. Controlling Motors](#)
 - [Servo Motor \(Using Servo Library\):](#)
- [13. Ultrasonic Sensor \(Distance Measurement\)](#)
- [14. Troubleshooting Tips](#)
- [15. Useful Arduino Libraries](#)
 - [Example: Full Project \(Temperature and LED Control\)](#)

1. What is Arduino?

- Arduino is an open-source electronics platform combining hardware and software.
- Microcontroller Boards: Arduino UNO, Mega, Nano, etc.
- Languages: C/C++ (with Arduino libraries).

2. Setting Up Arduino IDE

- Download: <https://www.arduino.cc/en/software>
 - Install on Windows/Mac/Linux.
 - Connect Board: Use USB cable.
 - Select Board and Port:
 - Tools > Board > Arduino Uno
 - Tools > Port > COM (USB)
-

3. Basic Structure of Arduino Code

```
// Runs once when the board starts
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // Initialize pin
}

// Loops continuously after setup
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // Turn LED on
  delay(1000);                     // Wait 1 second
  digitalWrite(LED_BUILTIN, LOW);  // Turn LED off
  delay(1000);                     // Wait 1 second
}
```

Key Functions:

- setup() - Runs once when the board powers on.
 - loop() - Repeats continuously.
-

4. Uploading Code

1. Write code in the Arduino IDE.
 2. Click Upload (→) or press Ctrl + U.
 3. The board will reset and run the code.
-

5. Pin Modes

```
pinMode(pin, mode);
```

- INPUT – Receives data (sensors).
 - OUTPUT – Sends data (LEDs, motors).
 - INPUT_PULLUP – Internal pull-up resistor (for switches).
-

6. Digital I/O

```
digitalWrite(pin, HIGH); // Turn on  
digitalWrite(pin, LOW);  // Turn off
```

```
int value = digitalRead(pin); // Read pin (HIGH/LOW)
```

- HIGH – 5V (On).
 - LOW – 0V (Off).
-

7. Analog I/O

Analog Read (0-1023):

```
int sensorValue = analogRead(A0); // Read value from pin A0
```

Analog Write (PWM, 0-255):

```
analogWrite(9, 128); // Write PWM signal to pin 9
```

8. Serial Communication

```
Serial.begin(9600);           // Start Serial at 9600 baud
Serial.println("Hello!");     // Print with newline
Serial.print("Value: ");      // Print without newline
Serial.print(sensorValue);
```

9. Control Structures

If-Else Statement:

```
if (sensorValue > 500) {
    digitalWrite(LED_BUILTIN, HIGH);
} else {
    digitalWrite(LED_BUILTIN, LOW);
}
```

For Loop:

```
for (int i = 0; i < 10; i++) {
    Serial.println(i);
}
```

While Loop:

```
while (digitalRead(2) == LOW) {
    Serial.println("Waiting...");
}
```

□ 10. Common Functions

Function	Description
pinMode()	Sets pin mode (INPUT/OUTPUT)
digitalWrite()	Writes HIGH/LOW to a digital pin
digitalRead()	Reads value from a digital pin
analogRead()	Reads value (0-1023) from analog pin
analogWrite()	Outputs PWM signal (0-255)
delay(ms)	Pauses for milliseconds
millis()	Returns time since start (ms)

11. Sensors and Modules

LED Blinking (Basic Example):

```
pinMode(LED_BUILTIN, OUTPUT);

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}
```

Button Input:

```
pinMode(2, INPUT);

void loop() {
  if (digitalRead(2) == HIGH) {
    Serial.println("Button Pressed");
  }
}
```

Temperature Sensor (LM35):

```
int temp = analogRead(A0);
float temperature = (temp / 1023.0) * 500;
```

```
Serial.println(temperature);
```

12. Controlling Motors

Servo Motor (Using Servo Library):

```
#include <Servo.h>

Servo servoMotor;
servoMotor.attach(9); // Attach to pin 9

void loop() {
    servoMotor.write(90); // Move to 90 degrees
    delay(1000);
    servoMotor.write(0); // Move to 0 degrees
    delay(1000);
}
```

13. Ultrasonic Sensor (Distance Measurement)

```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);

long duration = pulseIn(echoPin, HIGH);
int distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
delay(500);
}
```

14. Troubleshooting Tips

- Check Wiring – Ensure correct connections to pins.
 - Verify Board/Port – Recheck board and port selection.
 - Test with Basic Sketch – Upload a simple LED blink to verify hardware.
 - Restart Arduino IDE – Sometimes restarting the IDE resolves upload issues.
-

15. Useful Arduino Libraries

```
#include <Servo.h>           // Servo motor control
#include <Wire.h>             // I2C communication
#include <Adafruit_Sensor.h> // Adafruit sensor library
```

- Install libraries via Sketch > Include Library > Manage Libraries.
-

Example: Full Project (Temperature and LED Control)

```
const int tempPin = A0;
const int ledPin = 9;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
```

```
}
```

```
void loop() {  
  int temp = analogRead(tempPin);  
  float temperature = (temp / 1023.0) * 500;  
  Serial.print("Temp: ");  
  Serial.println(temperature);  
  
  if (temperature > 30) {  
    digitalWrite(ledPin, HIGH);  
  } else {  
    digitalWrite(ledPin, LOW);  
  }  
  delay(1000);  
}
```