## **Basic Code Structure**

C++

Every Arduino program has two main parts:

cpp

CopyEdit

void setup() {

// Runs once when the board starts

}

void loop() {

// Runs forever after setup()

}

🔹 **setup()** → Runs **once** at the start.  
🔹 **loop()** → Runs **continuously** (repeats forever).

## **Basic Commands**

### **1. Controlling Pins (Input/Output)**

#### **Turn an LED on and off**

cpp

CopyEdit

void setup() {

pinMode(13, OUTPUT); // Set pin 13 as an output

}

void loop() {

digitalWrite(13, HIGH); // Turn LED ON

delay(1000); // Wait 1 second

digitalWrite(13, LOW); // Turn LED OFF

delay(1000); // Wait 1 second

}

🔹 pinMode(pin, mode) → Set **INPUT** or **OUTPUT**🔹 digitalWrite(pin, HIGH/LOW) → Turn on/off  
🔹 delay(ms) → Pause for milliseconds

### **2. Reading a Button Press**

cpp

CopyEdit

void setup() {

pinMode(7, INPUT); // Set pin 7 as input

pinMode(13, OUTPUT);

}

void loop() {

int buttonState = digitalRead(7); // Read button state

if (buttonState == HIGH) {

digitalWrite(13, HIGH); // Turn LED ON if button pressed

} else {

digitalWrite(13, LOW); // Turn LED OFF

}

}

🔹 digitalRead(pin) → Reads **HIGH** (pressed) or **LOW** (not pressed).

### **3. Analog Input (Reading a Sensor)**

#### **Read a potentiometer (knob) value**

cpp

CopyEdit

void setup() {

Serial.begin(9600); // Start Serial Monitor

}

void loop() {

int sensorValue = analogRead(A0); // Read sensor on pin A0

Serial.println(sensorValue); // Print the value

delay(500); // Wait half a second

}

🔹 analogRead(pin) → Reads a value **0 to 1023** from a sensor.  
🔹 Serial.println(value) → Print the value in the Serial Monitor.

### **4. Analog Output (Controlling LED Brightness)**

cpp

CopyEdit

void setup() {

pinMode(9, OUTPUT); // Set pin 9 as an output

}

void loop() {

analogWrite(9, 128); // Dim LED (0-255)

delay(1000);

analogWrite(9, 255); // Full brightness

delay(1000);

}

🔹 analogWrite(pin, value) → Set brightness **(0 = OFF, 255 = FULL)**.

### **5. Looping with For Loop**

#### **Blink LED 5 times quickly**

cpp

CopyEdit

void setup() {

pinMode(13, OUTPUT);

}

void loop() {

for (int i = 0; i < 5; i++) {

digitalWrite(13, HIGH);

delay(200);

digitalWrite(13, LOW);

delay(200);

}

delay(2000); // Pause after 5 blinks

}

🔹 for (int i = 0; i < 5; i++) → Repeat **5 times**.

### **6. Using If Statements**

#### **Turn on LED when sensor value is high**

cpp

CopyEdit

void setup() {

pinMode(9, OUTPUT);

Serial.begin(9600);

}

void loop() {

int sensorValue = analogRead(A0);

if (sensorValue > 500) { // If value is greater than 500

digitalWrite(9, HIGH); // Turn LED ON

} else {

digitalWrite(9, LOW); // Turn LED OFF

}

Serial.println(sensorValue);

}

🔹 if (condition) → Checks a condition.

### **7. Timing Without Delay (Blink Without Pausing)**

#### **Blink LED without stopping the program**

cpp

CopyEdit

int previousTime = 0; // Store last time LED changed state

int interval = 1000; // Blink every 1 second

void setup() {

pinMode(13, OUTPUT);

}

void loop() {

int currentTime = millis(); // Get current time

if (currentTime - previousTime >= interval) {

previousTime = currentTime; // Update last time

digitalWrite(13, !digitalRead(13)); // Toggle LED

}

}

🔹 millis() → Returns time since the Arduino started.  
🔹 **No delay()!** Keeps running other tasks.

### **8. Serial Communication (Talking to Your Computer)**

#### **Send messages to Serial Monitor**

cpp

CopyEdit

void setup() {

Serial.begin(9600); // Start Serial Monitor

}

void loop() {

Serial.println("Hello, Arduino!"); // Print message

delay(1000); // Wait 1 second

}

🔹 Open the **Serial Monitor** (Tools > Serial Monitor).

### **9. Storing Data (EEPROM)**

#### **Save a number so it stays even after power off**

cpp

CopyEdit

#include <EEPROM.h>

void setup() {

EEPROM.write(0, 42); // Store number 42 at memory address 0

}

void loop() {

int storedValue = EEPROM.read(0); // Read from memory

Serial.println(storedValue);

delay(1000);

}

🔹 **EEPROM** keeps data even after you turn off Arduino!

### **10. Using Interrupts (Reacting to Events)**

#### **Turn LED on when button is pressed, without waiting**

cpp

CopyEdit

void buttonPress(); // Function prototype

void setup() {

pinMode(2, INPUT\_PULLUP);

pinMode(13, OUTPUT);

attachInterrupt(digitalPinToInterrupt(2), buttonPress, FALLING);

}

void loop() {

// Nothing here! LED turns on instantly when button is pressed.

}

void buttonPress() {

digitalWrite(13, HIGH); // Turn LED ON

}

🔹 **Interrupts react immediately** to button presses.

## **Quick Reference**

| **Function** | **Description** |
| --- | --- |
| pinMode(pin, mode) | Set **INPUT** or **OUTPUT** |
| digitalWrite(pin, HIGH/LOW) | Turn pin ON/OFF |
| digitalRead(pin) | Read **HIGH** or **LOW** |
| analogRead(pin) | Read sensor value (0-1023) |
| analogWrite(pin, value) | Control brightness (0-255) |
| delay(ms) | Pause (milliseconds) |
| millis() | Get current time (milliseconds) |
| Serial.begin(9600) | Start serial communication |
| Serial.print(value) | Print text (no new line) |
| Serial.println(value) | Print text (with new line) |