Introduction to Arduino

Johar M. Ashfaque

April 9, 2025

Abstract

Arduino is an open-source electronics platform that combines user-friendly hardware and software, making it ideal for artists, designers, hobbyists, and anyone interested in creating interactive electronic projects. This document provides an overview of the Arduino platform, including its history, core components, software interface, programming concepts, common and advanced applications, and guidance for aspiring developers.

1 What is Arduino?

Arduino is an open-source electronics prototyping platform that combines both hardware (a programmable circuit board) and software (an IDE for writing and uploading code). Arduino boards are designed to read inputs such as light on a sensor or a finger on a button and turn them into outputs like activating a motor or turning on an LED.

2 History of Arduino

The Arduino project was initiated in 2005 by Massimo Banzi and David Cuartielles at the Interaction Design Institute Ivrea in Ivrea, Italy. The primary aim was to create low-cost hardware for students and hobbyists. The platform has since evolved into a cornerstone of the maker community.

3 Arduino Hardware

There are many versions of Arduino boards, including:

- **Arduino Uno** – the most popular board.
- **Arduino Mega** – for larger and more complex projects.
- **Arduino Nano** – compact and breadboard-friendly.
- **Arduino Leonardo** – has built-in USB communication.

Most Arduino boards share common components:

- **Microcontroller**: The brain of the board (e.g., ATmega328P).

- **Digital and Analog I/O Pins**: For input/output operations.
- **Power Supply**: Through USB or external adapter.
- **USB Port**: For uploading code from the computer.

4 Arduino Software (IDE)

The Arduino IDE (Integrated Development Environment) is used to write, compile, and upload code to the Arduino board. It supports C/C++ and uses a simplified syntax.

4.1 Basic Structure of Arduino Code

Arduino code consists of two main functions:

```
void setup() {
    // runs once
    pinMode(13, OUTPUT);
}

void loop() {
    // runs continuously
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Listing 1: Basic Arduino Sketch

5 Programming Arduino

Arduino programming is based on C/C++ with some abstractions. Important functions include:

- `pinMode(pin, mode)` – sets a pin as INPUT or OUTPUT.
- `digitalWrite(pin, value)` – sets a digital pin HIGH or LOW.
- `digitalRead(pin)` – reads a digital pin's value.
- `analogRead(pin)` – reads the value from an analog pin.
- `analogWrite(pin, value)` – outputs PWM signal.
- `delay(ms)` – pauses execution for specified milliseconds.

6 Applications of Arduino

Arduino is used in various fields due to its versatility:

- **Education**: Teaching electronics and programming.

- **Robotics**: Controlling motors, sensors, and actuators.
- **Home Automation**: Lights, fans, and security systems.
- **IoT Devices**: Connecting physical devices to the internet.
- **Wearables**: Fitness trackers, health monitors.

7 Sample Project: Blinking LED

This is the classic first Arduino project.

7.1 Components Required

- Arduino Uno
- LED
- 220-ohm resistor
- Breadboard
- Jumper wires

7.2 Code

```
void setup() {
    pinMode(9, OUTPUT);
}

void loop() {
    digitalWrite(9, HIGH); // LED on
    delay(500);            // wait
    digitalWrite(9, LOW);  // LED off
    delay(500);            // wait
}
```

Listing 2: Blinking LED

8 Advanced Projects with Arduino

8.1 Project 1: Temperature Logger with SD Card

Components Required:

- Arduino Uno
- DHT11 Sensor
- SD Card Module + SD Card
- Breadboard, jumper wires

Code Snippet:

```
#include <DHT.h>
#include <SD.h>

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

File myFile;

void setup() {
  dht.begin();
  SD.begin(4);
  myFile = SD.open("datalog.csv", FILE_WRITE);
  myFile.println("Temp,Humidity");
  myFile.close();
}

void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  myFile = SD.open("datalog.csv", FILE_WRITE);
  if (myFile) {
    myFile.print(temp);
    myFile.print(",");
    myFile.println(hum);
    myFile.close();
  }
  delay(2000);
}
```

Listing 3: Temperature Logger (Snippet)

8.2 Project 2: Bluetooth-Controlled Robot

Components Required:

- Arduino Uno
- L298N Motor Driver
- HC-05 Bluetooth Module
- 2 DC Motors + chassis
- Battery, jumper wires

Concept: Control the robot via mobile app using Bluetooth commands (F, B, L, R, S).

9 Using Shields and Modules

- **Shields**: Motor Shield, Ethernet Shield, Relay Shield
- **Modules**: Ultrasonic Sensor, RTC Module, IR Receiver

10 Common Issues and Troubleshooting

Upload Errors

- Verify board and COM port
- Close other serial monitor tools

Unresponsive Hardware

- Check wiring
- Test with built-in LED

Sensor Issues

- Test sensor independently
- Verify power and data pins

11 Next Steps for Arduino Enthusiasts

- Explore advanced libraries
- Combine Arduino with IoT (ESP32/ESP8266)
- Learn PCB design
- Contribute to open-source projects

12 Conclusion

Arduino is more than just a development board—it's a gateway to innovation. Whether you're building blinking LEDs or an autonomous robot, the platform supports a wide range of creative ambitions. As you progress, you'll find Arduino to be a launchpad into more advanced fields such as robotics, IoT, and embedded systems.

References

- `https://www.arduino.cc/`
- Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino* (3rd ed.)
- Margolis, M. (2011). *Arduino Cookbook*. O'Reilly Media
- `https://forum.arduino.cc/`