Arduino and IoT

Johar M. Ashfaque

April 9, 2025

Abstract

This document outlines beginner-level Arduino and Internet of Things (IoT) projects including Hello World, Gas Sensor, Water Sensor, and DHT Sensor applications. These projects introduce fundamental concepts of electronics, coding, and data sensing, offering a stepping stone into the world of embedded systems and IoT. All projects include network connectivity setup to demonstrate IoT readiness.

Contents

1 Introduction 2

2 Components and Tools 2

3 WiFi Setup (For All IoT Projects) 2

4 Project 1: Hello World (Blinking LED) 3
4.1 Objective 3
4.2 Arduino Code 3

5 Project 2: Gas Sensor (MQ-2) 3
5.1 Objective 3
5.2 Arduino Code 3

6 Project 3: Water Level Sensor 4
6.1 Objective 4
6.2 Arduino Code 4

7 Project 4: Temperature and Humidity Sensor (DHT11) 5
7.1 Objective 5
7.2 Arduino Code 5

8 Conclusion 6

9 Future Work 6

10 References 6

1 Introduction

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's widely used in prototyping and educational contexts. This document covers several foundational projects using Arduino and commonly available sensors to introduce the concept of sensing, processing, and potentially communicating data as part of an Internet of Things (IoT) ecosystem.

2 Components and Tools

- Arduino Uno / NodeMCU ESP8266
- Breadboard and Jumper Wires
- DHT11 / DHT22 Temperature and Humidity Sensor
- MQ-2 Gas Sensor
- Water Level Sensor
- LEDs and Resistors
- Arduino IDE
- WiFi Network (SSID and Password)

3 WiFi Setup (For All IoT Projects)

Before any sensor data can be uploaded or logged remotely, the device must be connected to WiFi.

```
#include <ESP8266WiFi.h>

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

void connectWiFi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");
}
```

Listing 1: Common WiFi Setup

Add this function inside your `setup()` block in each IoT project.

4 Project 1: Hello World (Blinking LED)

4.1 Objective

Blink an LED and ensure the device is connected to the network.

4.2 Arduino Code

```
#include <ESP8266WiFi.h>

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, LOW); // LED ON
  delay(1000);
  digitalWrite(LED_BUILTIN, HIGH); // LED OFF
  delay(1000);
}
```

Listing 2: Blinking LED with WiFi

5 Project 2: Gas Sensor (MQ-2)

5.1 Objective

Detect gas and send values to serial, WiFi-enabled for IoT expansion.

5.2 Arduino Code

```
#include <ESP8266WiFi.h>

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
int gasSensor = A0;
```

```
void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");
}

void loop() {
  int gasLevel = analogRead(gasSensor);
  Serial.print("Gas␣Level:␣");
  Serial.println(gasLevel);
  delay(1000);
}
```

Listing 3: MQ-2 Gas Sensor with WiFi

6 Project 3: Water Level Sensor

6.1 Objective

Monitor water levels and prepare system for future IoT logging.

6.2 Arduino Code

```
#include <ESP8266WiFi.h>

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
int waterSensor = A1;

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");
}

void loop() {
  int waterLevel = analogRead(waterSensor);
  Serial.print("Water␣Level:␣");
  Serial.println(waterLevel);
  delay(1000);
}
```

Listing 4: Water Sensor with WiFi

7 Project 4: Temperature and Humidity Sensor (DHT11)

7.1 Objective

Measure temperature and humidity and prepare to send it to the cloud.

7.2 Arduino Code

```
#include <ESP8266WiFi.h>
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting␣to␣WiFi...");
  }
  Serial.println("Connected␣to␣WiFi");
  dht.begin();
}

void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed␣to␣read␣from␣DHT␣sensor!");
    return;
  }

  Serial.print("Humidity:␣");
  Serial.print(humidity);
  Serial.print("%␣␣Temperature:␣");
  Serial.print(temperature);
  Serial.println(" C ");

  delay(2000);
}
```

Listing 5: DHT11 with WiFi

8 Conclusion

Each of these simple projects now includes WiFi setup to emphasize their IoT potential. Even if they don't yet connect to cloud services, the groundwork is laid for future upgrades involving platforms like ThingSpeak, Blynk, or MQTT.

9 Future Work

- Upload sensor data to cloud platforms
- Create dashboards and notifications
- Combine multiple sensors into an integrated system
- Trigger actuators based on sensor thresholds

10 References

1. Arduino Documentation: `https://www.arduino.cc/`
2. DHT Sensor Library: `https://github.com/adafruit/DHT-sensor-library`
3. ESP8266 WiFi Reference: `https://arduino-esp8266.readthedocs.io/`