

# Extended Arduino/ESP32 Examples for Practice

## Blink with Different Delays

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); delay(200);  
  digitalWrite(13, LOW); delay(1000);  
}
```

### Explanation

LED blinks fast ON (200ms), slow OFF (1s).

## Fade LED using PWM

```
int ledPin = 9;  
  
void setup() { pinMode(ledPin, OUTPUT); }  
  
void loop() {  
  for (int b = 0; b <= 255; b++) {  
    analogWrite(ledPin, b); delay(10);  
  }  
  for (int b = 255; b >= 0; b--) {  
    analogWrite(ledPin, b); delay(10);  
  }  
}
```

### Explanation

LED brightness smoothly increases and decreases.

## Multiple LEDs Sequential (Chaser)

```
int leds[] = {2, 3, 4, 5, 6};

void setup() {
  for (int i = 0; i < 5; i++) pinMode(leds[i], OUTPUT);
}

void loop() {
  for (int i = 0; i < 5; i++) {
    digitalWrite(leds[i], HIGH);
    delay(200);
    digitalWrite(leds[i], LOW);
  }
}
```

### Explanation

LEDs light one by one, creating a chasing effect.

## Button-Controlled Buzzer

```
int buttonPin = 2, buzzerPin = 8;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  if (digitalRead(buttonPin) == HIGH)
    digitalWrite(buzzerPin, HIGH);
  else
    digitalWrite(buzzerPin, LOW);
}
```

### Explanation

Pressing the button turns the buzzer ON.

## Analog Sensor → LED Brightness

```
int sensorPin = A0, ledPin = 9;

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

void loop() {
  int sensorValue = analogRead(sensorPin);
  int brightness = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(ledPin, brightness);
  Serial.println(brightness);
  delay(100);
}
```

### Explanation

Potentiometer controls LED brightness.

## Temperature Sensor (LM35) Reading

```
int sensorPin = A0;

void setup() { Serial.begin(9600); }

void loop() {
  int val = analogRead(sensorPin);
  float voltage = val * (5.0 / 1023.0);
  float tempC = voltage * 100; // LM35: 10mV per C
  Serial.print("Temperature: ");
  Serial.print(tempC); Serial.println(" C");
  delay(1000);
}
```

#### Explanation

Reads LM35 analog output and converts it to °C.

## Ultrasonic Distance Sensor (HC-SR04)

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

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

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

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

#### Explanation

Uses sound pulses to measure distance.

## Servo Motor Control

```
#include <Servo.h>
Servo myservo;

void setup() { myservo.attach(9); }

void loop() {
  for (int pos = 0; pos <= 180; pos++) {
    myservo.write(pos); delay(15);
  }
}
```

```

for (int pos = 180; pos >= 0; pos--) {
    myservo.write(pos); delay(15);
}
}

```

#### Explanation

Moves a servo smoothly from 0° to 180° and back.

## ESP32: Connect to Wi-Fi

```

#include <WiFi.h>

const char* ssid = "YourWiFi";
const char* password = "YourPassword";

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    Serial.print("Connecting");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }
    Serial.println("\nConnected!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void loop() {}

```

#### Explanation

ESP32 connects to a Wi-Fi network and prints its IP.

## ESP32 Web Server (LED Control)

```

#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "YourWiFi";

```

```

const char* password = "YourPassword";

WebServer server(80);
int ledPin = 2;

void handleRoot() {
  server.send(200, "text/html",
    "<h1>ESP32 LED Control</h1>"
    "<a href=\"/on\">Turn ON</a><br>"
    "<a href=\"/off\">Turn OFF</a>");
}

void handleOn() { digitalWrite(ledPin, HIGH); server.send(200,"text/html","LED
  ON"); }
void handleOff(){ digitalWrite(ledPin, LOW); server.send(200,"text/html","LED
  OFF"); }

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) { delay(500); }
  Serial.println("Connected!");
  server.on("/", handleRoot);
  server.on("/on", handleOn);
  server.on("/off", handleOff);
  server.begin();
}

void loop() { server.handleClient(); }

```

### Explanation

Creates a simple web server: open ESP32 IP in browser → turn LED ON/OFF.

## ESP32: Send Sensor Data to Web Server

```

#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "YourWiFi";

```

```

const char* password = "YourPassword";

// Example: ThingSpeak server (replace with your own if needed)
String serverName = "http://api.thingspeak.com/update?api_key=YOUR_API_KEY";

int sensorPin = 34; // ESP32 analog pin

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("\nConnected to WiFi!");
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    int sensorValue = analogRead(sensorPin);
    float voltage = sensorValue * (3.3 / 4095.0); // ESP32 12-bit ADC
    Serial.print("Sensor Voltage: ");
    Serial.println(voltage);

    HTTPClient http;
    String url = serverName + "&field1=" + String(voltage);
    http.begin(url.c_str());
    int httpResponseCode = http.GET();
    if (httpResponseCode > 0) {
      Serial.print("Server Response: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
    http.end();
  }
  delay(20000); // Upload every 20 seconds
}

```

### Explanation

ESP32 reads an analog sensor (e.g., potentiometer, LM35) and sends the value to a server (ThingSpeak or any HTTP server).

- Connects to Wi-Fi.
- Reads sensor from pin 34 (0–3.3V, 12-bit ADC).
- Sends data via HTTP GET request.
- Can be adapted for REST APIs, cloud dashboards, or custom servers.