**1. Arduino: LED Interfacing**

**Code:**

void setup() {

pinMode(13, OUTPUT); // built-in LED

}

void loop() {

digitalWrite(13, HIGH); // ON

delay(500);

digitalWrite(13, LOW); // OFF

delay(500);

}

**Connection Sketch:**

* Use the built-in LED on pin 13, or connect an external LED:
  + Anode (long leg) → Pin 13
  + Cathode → 220Ω resistor → GND

**2. Arduino: Buzzer, Switches, LCD, Keypad, LDR, Ultrasonic, PWM**

I’ll break this into parts.

**(a) Buzzer**

void setup() {

pinMode(9, OUTPUT);

}

void loop() {

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(1000);

}

* **Connection:**
  + Buzzer + → Pin 9
  + Buzzer - → GND

**(b) Switch**

void setup() {

pinMode(2, INPUT\_PULLUP); // button input

pinMode(13, OUTPUT); // LED

}

void loop() {

if (digitalRead(2) == LOW) {

digitalWrite(13, HIGH);

} else {

digitalWrite(13, LOW);

}

}

**(c) 16x2 LCD (with I2C)**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup() {

lcd.init();

lcd.backlight();

lcd.print("Hello, World!");

}

void loop() {}

* Connect I2C LCD to SDA (A4) and SCL (A5).

**(d) Keypad**

Install library: **Keypad.h**

#include <Keypad.h>

const byte ROWS = 4, COLS = 4;

char keys[ROWS][COLS] = {

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

};

byte rowPins[ROWS] = {9,8,7,6};

byte colPins[COLS] = {5,4,3,2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {

Serial.begin(9600);

}

void loop() {

char key = keypad.getKey();

if (key) {

Serial.println(key);

}

}

**(e) LDR**

void setup() {

Serial.begin(9600);

}

void loop() {

int ldrVal = analogRead(A0);

Serial.println(ldrVal);

delay(500);

}

**(f) Ultrasonic Sensor (HC-SR04)**

#define trigPin 9

#define 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);

int distance = duration \* 0.034 / 2;

Serial.println(distance);

delay(1000);

}

**(g) PWM: Fade LED**

void setup() {

pinMode(9, OUTPUT);

}

void loop() {

for (int i = 0; i < 255; i++) {

analogWrite(9, i);

delay(10);

}

for (int i = 255; i > 0; i--) {

analogWrite(9, i);

delay(10);

}

}

**3. Arduino: Serial Communication (Device Control)**

void setup() {

Serial.begin(9600);

pinMode(13, OUTPUT);

}

void loop() {

if (Serial.available()) {

char command = Serial.read();

if (command == '1') digitalWrite(13, HIGH);

if (command == '0') digitalWrite(13, LOW);

}

}

* Send '1' or '0' from Serial Monitor to control the LED.

**4. NodeMCU: LED Interfacing**

void setup() {

pinMode(D1, OUTPUT);

}

void loop() {

digitalWrite(D1, HIGH);

delay(500);

digitalWrite(D1, LOW);

delay(500);

}

**5. NodeMCU: Thingspeak Data Publish**

#include <ESP8266WiFi.h>

#include "ThingSpeak.h"

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

WiFiClient client;

unsigned long myChannelNumber = YOUR\_CHANNEL\_NUMBER;

const char \* myWriteAPIKey = "YOUR\_API\_KEY";

void setup() {

Serial.begin(115200);

WiFi.begin(ssid, password);

while(WiFi.status() != WL\_CONNECTED) delay(1000);

ThingSpeak.begin(client);

}

void loop() {

int temp = random(20, 30);

ThingSpeak.setField(1, temp);

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

delay(15000);

}

**6. NodeMCU MQTT + Adafruit (Control Appliances)**

Use library: Adafruit\_MQTT.h

Let me know if you want full code for this with your Adafruit credentials.

**7. Raspberry Pi (Wokwi-style emulation)**

For now, no direct Raspberry Pi support in Wokwi. But basic concepts:

python

CopyEdit

# Raspberry Pi LED using GPIO

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.OUT)

while True:

GPIO.output(18, GPIO.HIGH)

time.sleep(1)

GPIO.output(18, GPIO.LOW)

time.sleep(1)

**8. Raspberry Pi: LED**

Same as above. GPIO 18 → LED → Resistor → GND.

**9. Node-RED: Study**

* Use browser-based Node-RED.
* Drag:
  + Inject Node → Set value
  + Function Node → process
  + Dashboard → Gauge/Chart
* Use MQTT/HTTP for sensor inputs.

**10. Sensor Data → Node-RED Dashboard**

* Send data from NodeMCU via MQTT.
* Node-RED subscribes and shows on Chart/Gauge.

**11. Upload Temp/Light to Thingspeak**

Use DHT or LDR:

#include <DHT.h>

#include <ThingSpeak.h>

#include <ESP8266WiFi.h>

const int DHTPin = D2;

DHT dht(DHTPin, DHT22);

WiFiClient client;

unsigned long channel = YOUR\_CHANNEL;

const char\* key = "API\_KEY";

void setup() {

Serial.begin(115200);

WiFi.begin("Wokwi-GUEST", "");

ThingSpeak.begin(client);

dht.begin();

}

void loop() {

float temp = dht.readTemperature();

float hum = dht.readHumidity();

ThingSpeak.setField(1, temp);

ThingSpeak.setField(2, hum);

ThingSpeak.writeFields(channel, key);

delay(15000);

}