**1. LED**

**a) blink**

int LED\_BUILTIN = 3;

void setup() {

pinMode(LED\_BUILTIN, OUTPUT);

}

void loop() {

digitalWrite(LED\_BUILTIN, HIGH);

delay(1000);

digitalWrite(LED\_BUILTIN, LOW);

delay(1000);

}

**b) fade**

int led = 3;

int brightness = 0;

int fadeAmount = 5;

void setup() {

pinMode(led, OUTPUT);

}

void loop()

{

for(brightness = 0; brightness <= 255 ; brightness += 5)

{

analogWrite(led, brightness);

delay (100);

}

for(brightness = 255; brightness >= 0 ; brightness -= 5)

{

analogWrite(led, brightness);

delay (100);

}

}

**2.TOUCH SENSOR**

1. **led**

int Led = 13 ;

int touchpin = 3;

int val ;

void setup ()

{

Serial.begin(9600);

pinMode (Led, OUTPUT) ;

pinMode (touchpin, INPUT) ;

}

void loop ()

{

val = digitalRead (touchpin) ;

if (val == HIGH)

{

digitalWrite (Led, HIGH);

Serial.print("Touch Sensor Status:");

Serial.println("1");

delay(1000);

}

else

{

digitalWrite (Led, LOW);

Serial.print("Touch Sensor Status:");

Serial.println("0");

delay(1000);

}

}

**b) buzzer**

int buzzpin=5;

int touchpin=3;

int val;

void setup()

{

pinMode(touchpin,INPUT);

pinMode(buzzpin,OUTPUT);

Serial.begin(9600);

}

void loop(){

val=digitalRead(touchpin);

if(val==HIGH)

{

Serial.println("detected");

tone(5,100,2000);

}

else

{

Serial.println(" Not detected");

}

}

**3.IR SENSOR**

#define IR 11

int detection = HIGH;

void setup()

{

Serial.begin(9600);

pinMode(IR, INPUT);

}

void loop()

{

detection = digitalRead(IR);

if(detection == LOW){

Serial.print("There is an obstacle!\n");

}

else

{

Serial.print("No obstacle!\n");

}

delay(500);

}

**4. PIR SENSOR**

int sensor=7;  
int sensor\_value;  
void setup()

{  
 pinMode(sensor,INPUT);  
Serial.begin(9600);  
}  
void loop()  
{  
sensor\_value=digitalRead(sensor);  
Serial.print("PIR:00008:Volts:");  
Serial.println(sensor\_value);  
}

**5.TEMPERATURE HUMIDITY SENSOR**

1. **digital**

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println("DHTxx test!");

dht.begin();

}

void loop()

{

delay(2000);

float h = dht.readHumidity();

float t = dht.readTemperature();

float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

float hif = dht.computeHeatIndex(f, h);

float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" \*C ");

Serial.print(f);

Serial.print(" \*F\t");

Serial.print("Heat index: ");

Serial.print(hic);

Serial.print(" \*C ");

Serial.print(hif);

Serial.println(" \*F");

}

**b) analog**

#include "dht.h"

#define dht\_apin A0

dht DHT;

void setup()

{

Serial.begin(9600);

delay(500);

Serial.println("DHT11 Humidity & temperature Sensor\n\n");

delay(1000);

}

void loop()

{

DHT.read11(dht\_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

Serial.println("C ");

delay(5000);

}

**6.VIBRATION SENSOR**

int ledPin = 13;

int vib=A4;

void setup()

{

pinMode(ledPin, OUTPUT);

pinMode(vib, INPUT); //set EP input for measurment

Serial.begin(9600); //init serial 9600

}

void loop()

{

long measurement=pulseIn (vib, HIGH);

delayMicroseconds(50);

Serial.print(" VIB : 00001 : hertz :" );

Serial.println(measurement);

if (measurement > 10)

{

digitalWrite(ledPin, HIGH);

}

else

{

digitalWrite(ledPin, LOW);

}

}

**7.PUSH BUTTON**

const int buttonPin = 2;

const int ledPin = 13;

int buttonState = 0;

void setup() {

pinMode(ledPin, OUTPUT);

pinMode(buttonPin, INPUT);

}

void loop() {

buttonState = digitalRead(buttonPin);

if (buttonState == HIGH)

{

digitalWrite(ledPin, HIGH);

}

else

{

digitalWrite(ledPin, LOW);

}

**8. DC MOTOR**

const int enB = 2;

const int in3 = 8;

const int in4 = 9;

void setup()

{

pinMode(enB, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

}

void loop()

{

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enB, 255);

delay(3000);

digitalWrite(in3, HIGH);

digitalWrite(in4, HIGH);

delay(1000);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(3000);

digitalWrite(in3, HIGH);

digitalWrite(in4, HIGH);

delay(1000);

}

**9.STEPPER MOTTOR**

**a) revolution:**

#include <Stepper.h>

const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {

myStepper.setSpeed(60);

Serial.begin(9600);

}

void loop() {

Serial.println("clockwise");

myStepper.step(stepsPerRevolution);

Serial.println("counterclockwise");

myStepper.step(-stepsPerRevolution);

}

1. **speed control:**

#include <Stepper.h>

const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

int stepCount = 0;

void setup() {

}

void loop() {

int sensorReading = analogRead(A0);

int motorSpeed = map(sensorReading, 0, 1023, 0, 100);

if (motorSpeed > 0) {

myStepper.setSpeed(motorSpeed);

myStepper.step(stepsPerRevolution / 100);

}

}

**10. ASSIGNING IP ADDRESS TO NODE MCU**

#include <ESP8266WiFi.h>

const char \*ssid="IoT";

const char \*pass="password";

int val;

String strs;

WiFiClient client;

void setup() {

Serial.begin(115200);

delay(10);

Serial.println("connecting to");

Serial.println(ssid);

WiFi.begin(ssid,pass);

while(WiFi.status()!=WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println(" ");

Serial.println("WiFi connected");

Serial.println("IP address:");

Serial.println(WiFi.localIP());

}

void loop()

{

}

**11. COMMUNICATING SENSOR VALUES FROM ARDUINO TO NODE MCU**

1. **ultrasonic data transmission from arduino:**

const int trigPin=9;

const int echoPin=10;

long duration;

int distance;

void setup() {

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

Serial.begin(115200);

}

void loop() {

digitalWrite(trigPin,LOW);

delayMicroseconds(2);

digitalWrite(trigPin,HIGH);

delayMicroseconds(10);

digitalWrite(trigPin,LOW);

duration=pulseIn(echoPin,HIGH);

distance=duration\*0.034/2;

//Serial.print("ULT:00002:cm:");

Serial.println(distance);

delay(500);

}

**b) ultrasonic data received by node mcu :**

#include <ESP8266WiFi.h>

const char \*ssid="IoT";

const char \*pass="password";

int val;

String strs;

WiFiClient client;

void setup() {

Serial.begin(115200);

delay(10);

Serial.println("connecting to");

Serial.println(ssid);

WiFi.begin(ssid,pass);

while(WiFi.status()!=WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println(" ");

Serial.println("WiFi connected");

Serial.println("IP address:");

Serial.println(WiFi.localIP());

}

void loop() {

strs=Serial.readStringUntil('\n');

val=strs.toInt();

Serial.print("ultrasonic through node mcu");

Serial.println(val);

delay(1000);

}

**12.UPLOADING SENSOR VALUES FROM NODE MCU TO THINGSPEAK CLOUD**

**a) ultrasonic thingspeak code:**

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ThingSpeak.h>

const char \*ssid="DrTAmudha";

const char \*password=" ";

int val;

String strs;

WiFiClient client;

unsigned long myChannelNumber=1898294;

const char \*myWriteAPIkey="QCDA91DJFGPL6WO8";

void setup() {

Serial.begin(9600);

delay(10);

WiFi.begin(ssid,password);

ThingSpeak.begin(client);

}

void loop() {

strs=Serial.readStringUntil("\n);

val=strs.yoInt();

Serial.println(val);

ThingSpeak.writeField(myChannelNumber,1,val,myWriteAPIKey);

delay(1000);

}

1. **ultrasonic code:**

const int trigPin=9;

const int echoPin=10;

long duration;

int distance;

void setup() {

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

Serial.begin(115200);

}

void loop() {

digitalWrite(trigPin,LOW);

delayMicroseconds(2);

digitalWrite(trigPin,HIGH);

delayMicroseconds(10);

digitalWrite(trigPin,LOW);

duration=pulseIn(echoPin,HIGH);

distance=duration\*0.034/2;

//Serial.print("ULT:00002:cm:");

Serial.println(distance);

delay(500);

}