

//LED Blinking CODE

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

// IR Sensor CODE

```
int SensorPin = 2;
int OutputPin = 13;

void setup()
{
  pinMode(OutputPin, OUTPUT);
  pinMode(SensorPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int SensorValue = digitalRead(SensorPin);

  Serial.print("SensorPin Value: ");
  Serial.println(SensorValue);
  delay(1000);

  if (SensorValue==LOW)
  {
    digitalWrite(OutputPin, HIGH);
  }
  else
  {
    digitalWrite(OutputPin, LOW);
  }
}
```

//Touch Sensor

```
#define tsPin7 // touch sensor pin
int ledPin = 3; // led pin

void setup()
{
  Serial.begin(9600); //sensor baud rate
  pinMode(ledPin, OUTPUT); // led
  pinMode(tsPin, INPUT); // touch sensor
}

void loop()
{
  int tsValue = digitalRead(tsPin);
  delay(100);

  if (tsValue == HIGH)
  {
    digitalWrite(ledPin, HIGH); // led on
    Serial.println("TOUCHED");
  }
  else
  {
    digitalWrite(ledPin, LOW); // led off
    Serial.println("not touched");
  }
}
```

// Ultrasonic Sensor

```
int pingPin = 6;
int echoPin = 5;
void setup()
{
  Serial.begin(9600); // Starting Serial Terminal
}

void loop()
{
  long duration, inches, cm;
  pinMode(6, OUTPUT);
  digitalWrite(6, 0);
  delayMicroseconds(2);
  digitalWrite(6, 1);
  delayMicroseconds(10);
  digitalWrite(6, 0);
  pinMode(5, INPUT);
  duration = pulseIn(5, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(100);
}

long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

//Vibration Sensor

```
intvib_pin=5;
intled_pin=13;

voidsetup()
{
  pinMode(vib_pin,INPUT);
  pinMode(led_pin,OUTPUT);
}
voidloop()
{
  intval;
  val=digitalRead(vib_pin);
  if(val==1)
  {
    digitalWrite(led_pin,HIGH);
    delay(1000);
    digitalWrite(led_pin,LOW);
    delay(1000);
  }
  else
    digitalWrite(led_pin,LOW);
}
```

//PIR Sensor

```
intpirPin = 2;
void setup()
{
  Serial.begin(9600); // Initialize serial communication
  pinMode(pirPin, INPUT); // Set the PIR pin as an input
}

void loop()
{
  intpirState = digitalRead(pirPin);
  if (pirState == HIGH)
  {
    Serial.println("Motion detected!");
  }
  else
  {
    Serial.println("No motion detected.");
  }
  delay(500);
}
```

//Soil Moisture

```
const int soilMoisturePin = A0;

void setup()
{
  // Initialize serial communication for debugging
  Serial.begin(9600);
}

void loop()
{
  // Read the analog value from the soil moisture sensor
  int soilMoistureValue = analogRead(soilMoisturePin);

  // Map the analog value to a percentage value (0-100%)
  // Adjust these values based on your sensor and soil conditions
  int soilMoisturePercent = map(soilMoistureValue, 0, 1023, 0, 100);

  // Print the soil moisture percentage to the serial monitor
  Serial.print("Soil Moisture: ");
  Serial.print(soilMoisturePercent);
  Serial.println("%");

  // Add a delay to prevent spamming the serial monitor
  delay(1000); // Adjust delay as needed
}
```

//Water Level Detection

```
#define POWER_PIN 7
#define SIGNAL_PIN A5

int value = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(POWER_PIN, OUTPUT);
  digitalWrite(POWER_PIN, LOW);
}

void loop()
{
  digitalWrite(POWER_PIN, HIGH);
  delay(10);
  value = analogRead(SIGNAL_PIN);
  digitalWrite(POWER_PIN, LOW);
}
```

// LM35 TEMPERATURE SENSOR

```
float temp;
int sensor = 0;

void setup()
{
  Serial.begin(9600); //start the serial monitor
}

void loop()
{
  temp = analogRead(sensor); //assigning the analog output to temp
  temp = temp * 0.48828125;
  //converting volts to degrees celsius ----- 0.48828125 =
  // [(5V*1000)/1024]10

  //print information on the serial monitor
  Serial.print("The temperature is :");
  Serial.print(temp);
  Serial.println("deg. Celsius");

  //wait 1 second
  delay(1000);
}
```

//Smoke Sensor

```
#define MQ2pin 0
float sensorValue; //variable to store sensor value
void setup()
{
  Serial.begin(9600); // sets the serial port to 9600
  Serial.println("Gas sensor warming up!");
  delay(20000); // allow the MQ-2 to warm up
}

void loop()
{
  sensorValue = analogRead(MQ2pin); // read analog input pin 0
  Serial.print("Sensor Value: ");
  Serial.print(sensorValue);

  if(sensorValue > 200)
  {
    Serial.print(" | Smoke detected!");
  }

  Serial.println("");
  delay(2000); // wait 2s for next reading
}
```

//Sound Detection Sensor

```
const int TOUCH_SENSOR_PIN = 7;
const int LED_PIN = 3;

int ledState = LOW;
int lastTouchState;
int currentTouchState;
unsigned long lastPrintTime = 0;

void setup() {
  Serial.begin(9600);
  pinMode(TOUCH_SENSOR_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);

  currentTouchState = digitalRead(TOUCH_SENSOR_PIN);
}

void loop() {
  lastTouchState = currentTouchState;
  currentTouchState = digitalRead(TOUCH_SENSOR_PIN);

  if (lastTouchState == LOW && currentTouchState == HIGH) {
    Serial.println("The sensor is touched");

    ledState = !ledState;

    digitalWrite(LED_PIN, ledState);

    unsigned long currentTime = millis();
    Serial.print("Time: ");
    Serial.print(currentTime);
    Serial.print(" ms - Touch State: ");
    Serial.println(currentTouchState);

    lastPrintTime = currentTime;
  }
}
```



```

//RFID
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(115200);
  while (!Serial);
  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  mfrc522.PCD_DumpVersionToSerial();
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
}

void loop() {
  if (!mfrc522.PICC_IsNewCardPresent())
    return;

  if (!mfrc522.PICC_ReadCardSerial())
    return;

  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}

```