

BLINK WITHOUT DELAY EXP 1

```
const int ledPin1 = 2;//arduino pin 2
```

```
const int ledPin2= 3;//arduino pin 3
```

```
const int ledPin3 = 4;//arduino pin 4
```

```
int ledState = LOW;
```

```
void setup() {
```

```
    pinMode(ledPin1, OUTPUT);
```

```
    pinMode(ledPin2, OUTPUT);
```

```
    pinMode(ledPin3, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    digitalWrite(ledPin1, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(ledPin2, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(ledPin3, HIGH);
```

```
    delay(1000);
```

```
}
```

BUTTON EXP 2

```
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);
  }
}
```

DTH TEMPERATURE EXP 3

```
#include <DHT11.h>
```

```
DHT11 dht11(2);
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
    int humidity = dht11.readHumidity();
```

```
    if (humidity != DHT11::ERROR_CHECKSUM && humidity != DHT11::ERROR_TIMEOUT)
```

```
    {
```

```
        Serial.print("Humidity: ");
```

```
        Serial.print(humidity);
```

```
        Serial.println(" %");
```

```
    }
```

```
    else
```

```
    {
```

```
        Serial.println(DHT11::getErrorString(humidity));
```

```
    }
```

```
    delay(1000);
```

```
}
```

RELAY EXP 4

```
#define RELAY1 7

void setup() {
  pinMode(RELAY1, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(RELAY1, HIGH);
  Serial.println("RELAY1");
  delay(2000);
  digitalWrite(RELAY1, LOW);
  Serial.println("RELAY1");
  delay(2000);
}
```

POTENTIOMETER EXP 5

```
void setup() {
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    int sensorValue = analogRead(A0);
```

```
    Serial.println(sensorValue);
```

```
    delay(1);
```

```
}
```

EXP 6.A

```
import RPi.GPIO as GPIO
import time
pin=18
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin, GPIO.OUT)
GPIO.output(pin, GPIO.HIGH)
time.sleep(1)
GPIO.output(pin, GPIO.LOW)
time.sleep(1)
GPIO.cleanup()
```

EXP 6.B

```
import picamera
camera = picamera.PiCamera()
camera.capture('image.jpg')
```

EXP 7 BMP280

```
#include <Wire.h>
#include "SPI.h"
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP280.h"
Adafruit_BMP280 bmp;
/*//For SPI connection!
#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_MOSI 11
#define BMP_CS 10 */
float pressure;
float temperature;
int altimeter;
void setup() {
  bmp.begin();
  Serial.begin(9600);
  Serial.println("Adafruit BMP280 test:");
}

void loop() {
  pressure = bmp.readPressure();
  Serial.print(F("Pressure: "));
  Serial.print(pressure);
  Serial.print(" Pa");
  Serial.print("\t");
  delay(5000);
}
```

EXP 8 BLUETOOTH SERIAL

```
#include "BluetoothSerial.h"
```

```
BluetoothSerial SerialBT;
```

```
void setup(){
```

```
    SerialBT.begin("ESP32");
```

```
}
```

```
void loop(){
```

```
    SerialBT.println("HELLO WORLD");
```

```
    delay(1000);
```

```
}
```


INTERFACING MPU6050 WITH ARDUINO EXP 9

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
Adafruit_MPU6050 mpu;
void setup(void) {
    Serial.begin(115200);
    if (!mpu.begin()) {
        Serial.println("Failed to find MPU6050 chip");
        while (1) {
            delay(10);
        }
    }
    Serial.println("MPU6050 Found!");
    mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
    mpu.setGyroRange(MPU6050_RANGE_500_DEG);
    mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
    delay(100);
}
void loop() {

    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);
    Serial.print("Acceleration X: ");
    Serial.print(a.acceleration.x);
    Serial.print(", Y: ");
    Serial.print(a.acceleration.y);
    Serial.print(", Z: ");
    Serial.print(a.acceleration.z);
    Serial.println(" m/s^2");
    Serial.print("Rotation X: ");
    Serial.print(g.gyro.x);
    Serial.print(", Y: ");
    Serial.print(g.gyro.y);
    Serial.print(", Z: ");
    Serial.print(g.gyro.z);
    Serial.println(" rad/s");

    delay(500);
}
```