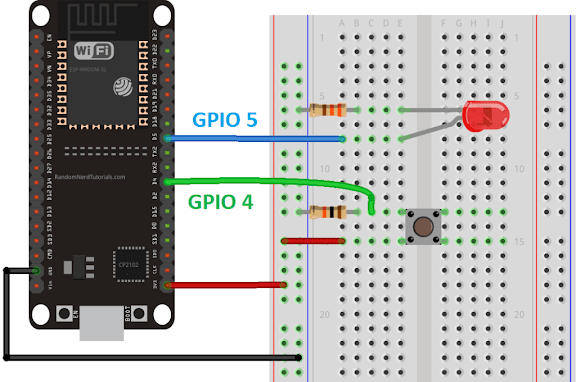
2. Latency measurement of ESP 32 Microcontroller(Practical)



**1.Plug a male to male wire from 3V3 to positive side of breadboard  
2. Plug a male to male wire from postive side of breadboard to a push button  
3. Plug a 10k Ohm resistor beside the other pin of the push button  
4. Plug a male to male wire from the push button to ESP32 on any GPIO 4 pin you want  
5. Plug a LED to ESP32 and put a 220 Ohm resistor next to it  
6. Plug a male to male wire from GND to negative side of breadboard**

Code:

#include <Arduino.h>

#define GPIO\_INPUT\_PIN 5  // Input pin to trigger the interrupt(LED)

#define GPIO\_OUTPUT\_PIN 4 // Output pin to measure latency(push button)

volatile uint32\_t start\_time = 0; // Stores the time when the interrupt is triggered

volatile uint32\_t end\_time = 0;   // Stores the time when the ISR starts executing

volatile uint32\_t latency = 0;    // Stores the calculated interrupt latency

void IRAM\_ATTR handleInterrupt() {

  // Read the current time in microseconds

  end\_time = micros();

  // Calculate the latency

  latency = end\_time - start\_time;

  // Toggle the output pin for visual indication (optional)

  digitalWrite(GPIO\_OUTPUT\_PIN, !digitalRead(GPIO\_OUTPUT\_PIN));

}

void setup() {

  // Configure the input pin

  pinMode(GPIO\_INPUT\_PIN, INPUT\_PULLUP);

  // Configure the output pin

  pinMode(GPIO\_OUTPUT\_PIN, OUTPUT);

  // Attach interrupt to the input pin

  attachInterrupt(GPIO\_INPUT\_PIN, []() {

    start\_time = micros(); // Record the time when interrupt is triggered

    handleInterrupt();     // Call the ISR

  }, FALLING);

  // Start serial communication

  Serial.begin(115200);

}

void loop() {

  // Print the interrupt latency periodically

  if (latency > 0) {

    Serial.print("Interrupt Latency: ");

    Serial.print(latency);

    Serial.println(" microseconds");

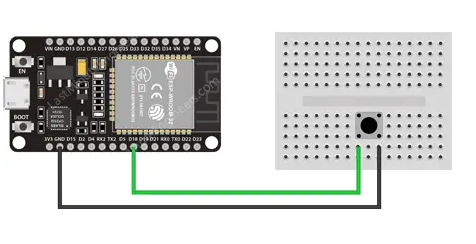
    latency = 0; // Reset for the next measurement

  }

  delay(1000); // Short delay to prevent spamming the Serial Monitor

}

2. Interrupt handling of ESP 32 Microcontroller.(skill)



Code:

struct Button {

  const uint8\_t PIN;

  uint32\_t numberKeyPresses;

  bool pressed;

};

Button button1 = {18, 0, false};

void IRAM\_ATTR isr() {

  button1.numberKeyPresses++;

  button1.pressed = true;

}

void setup() {

  Serial.begin(115200);

  pinMode(button1.PIN, INPUT\_PULLUP);

  attachInterrupt(button1.PIN, isr, FALLING);

}

void loop() {

  if (button1.pressed) {

    Serial.printf("Button has been pressed %u times\n", button1.numberKeyPresses);

    button1.pressed = false;

  }

}