**REPORT**

## 1. INTRODUCTION

1) We are using an ESP-WROOM-32. A successor to the ESP 8266 low-cost, low-power system on a chip microcontroller manufactured by Espressif Systems with integrated Wi-Fi and dual-mode Bluetooth. It employs a single-core 2.4GHz RISC-V microprocessor manufactured by TSMC using their 40nm Process and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. It has proved reliable in a wide variety of applications and power scenarios.
2) The ESP32 has a total of 30 Physical pins similar to Arduino digital pins which allows you to add LED, Resistors, sensors, buttons, etc. to our projects.
3) These pins are interfaced using the help of internal pull-up, pull-down, and high impedance status as well for the purposes of connecting buttons and matrix keyboards.
4) Maximum current drawn per a single GPIO is 40mA according to the “Recommended Operating Conditions” section in the ESP32 datasheet.
5) The Nature of it’s size and ease of use makes it ideal for Portable as well as wearable IoT Usage. It is capable of simulating a Nano-second level Clock gating, power gating and scaling.

**Fig 1.1: ESP32**

**2. Peripherals and Sensors (**Interface Pins)

ESP32 has 34 pins which can be designated for duties by differing the allocation of the appropriate registers.

The pins have differing general operating methods. Digital, Analog, Capacitive-touch are among a few.

The modes can be interchangeably configured based on a “Threshold” Value using the Analog to Digital Convertor. This is available on 18 of the 30 pins.

This helps the board claim it’s utility as a Low-Power Consumption board, using these 18 pins in sleep mode, waking up the CPU. Interchangeably, 8-but channels are available, to convert digital signals to analog.

The design comprises of using Resistors structure and an intermediate buffer.This supports the input Voltage supply as the reference.

3. **Calculations**

1] The calculation for the parameters of the Pulse Stream have been made using the first Five letters of the Students’ Surname.

The legend used is as follows

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Z | Y | X | W | V | U | T | S | R | Q | P | O | N |

Parameters for the above normal waveform are:

Width of The pulse: (Letter K)

A = 8 * 100 µS = 1100 µS

Pulses get incremented by 50 µS.
(1100, 1150, 1200, 1250, 1300, 1350, 1400, 1450, 1500, 1550, 1600, 1650, 1700, 1750, 1800 )

Gap Duration: (Letter A)

B =1 * 100 µS = 100 µS

Number of Pulses: (Letter S)

C = 8+4 = 12

Delay: (Letter H)

D = 8 * 500 µS = 4000 µS = 4mS

2] The possible system mode (Based on the letter Y) is - 2

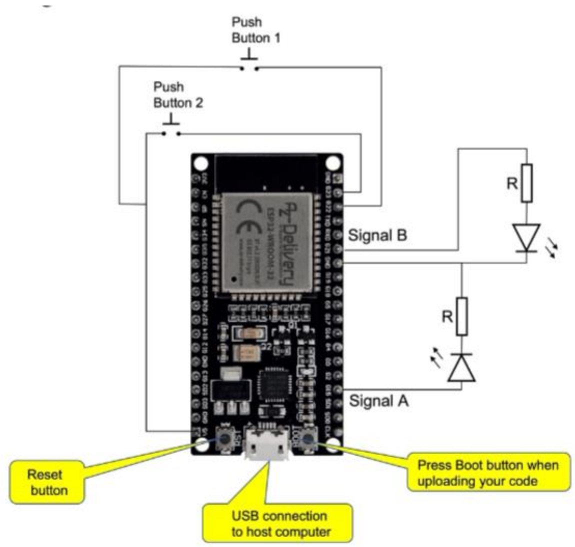


*Figure 1: Circuit Schematic*

Mode 2: Number of changed Pulses increases by 3.
Based on this, the Number of Pulses will change from 12 to 15.

**4. Generating circuit diagram.**

To build the circuit we are using the provided equipment.

- ESP-WROOM-32 Board.
- Jumper Cables.
- LEDs.
- Push Buttons.
- Pull Down Resistors.
- Oscilloscope

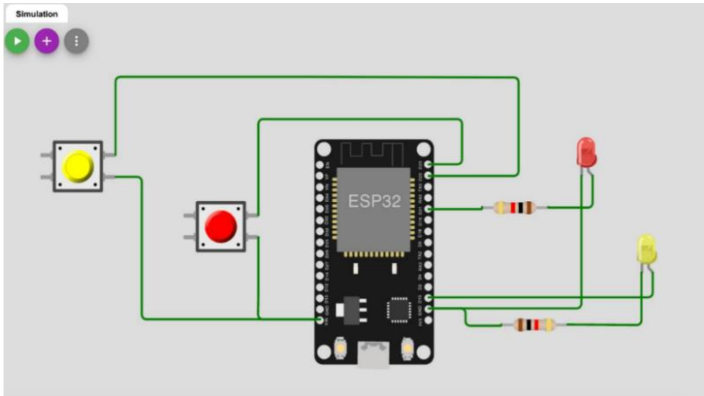


*Figure 2: Schematic on WOKWI*

**4. ESP32 Circuit Diagram**

ESP32 circuit diagram was initially simulated and prototyped using an Open Source **online simulator**. The one in question is Wokwi. Shown above is the schematic.

**Task 1**

**Code with Arduino:**

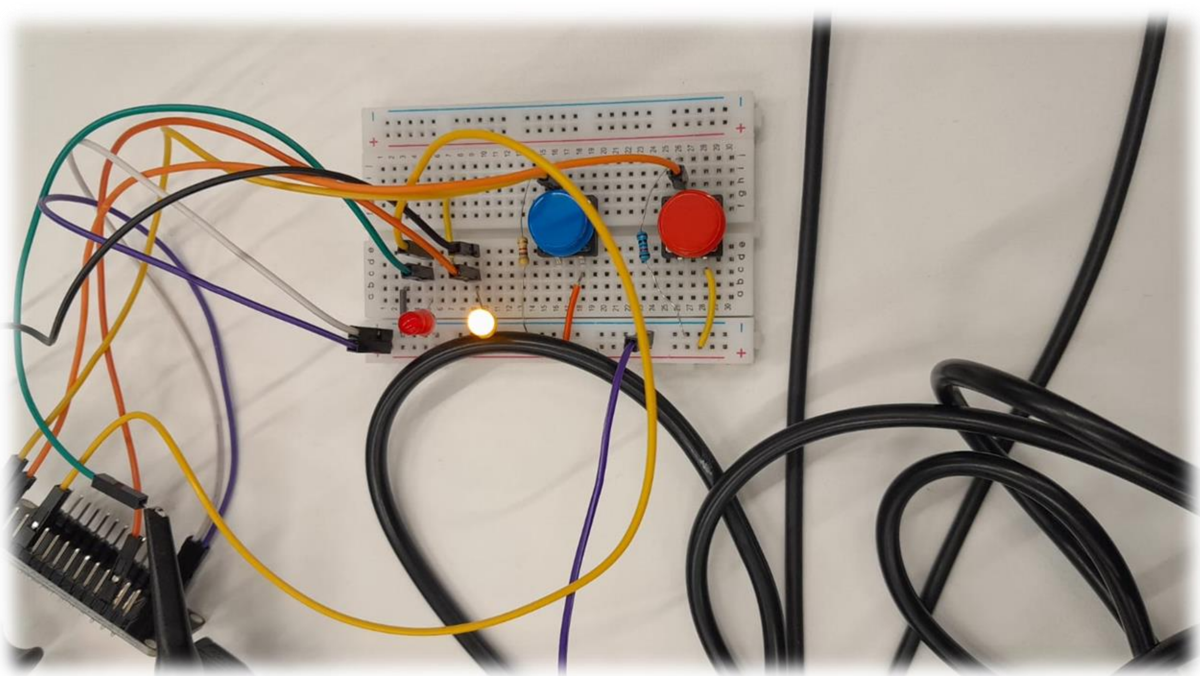
//Assignment1 ESP32

```
const int led_1 = 15;
const int led_2 = 21;
const int switch_1 = 22;
const int switch_2 = 23;
 void setup() {
   Serial.begin(115200);
   pinMode(15, OUTPUT);
   pinMode(21, OUTPUT);
   pinMode(22, INPUT);
   pinMode(23, INPUT);
 }
 void sig_b(int ppin, int len, int b){
   digitalWrite(ppin, HIGH);
   delay(b);
   digitalWrite (ppin, LOW);
   delay(len);
 }

 void sig_a(int pin, int pul_dur_a, int time_step, int pause_b,int num_pul, int
d) {
   for(int i = 0; i < num_pul; i ++) {
      digitalWrite(pin, HIGH);
     delay(pul_dur_a);
     pul_dur_a = pul_dur_a+time_step;
     digitalWrite(pin, LOW);
     delay(pause_b);
   }
  digitalWrite(pin, LOW);
  delay (d);
 }


 void loop()
 {
   if (digitalRead(switch_1)==HIGH)
   {
     digitalWrite(led_1, LOW);
     digitalWrite(led_2, LOW);
```

```
    }
    else if (digitalRead(switch_1) == LOW)
     {
     if (digitalRead(switch_2) == HIGH)
     {
     sig_a(led_1, 800, 50, 100, 15, 4000);
     sig_b(led_2, 18900, 50);
     }
     else
     {
     sig_a(led_1, 800, 50, 100, 12, 4000);
     sig_b(led_2, 19550, 50);
     }
     }
 }
```

**Task 2**

Making the connections on the Test bench



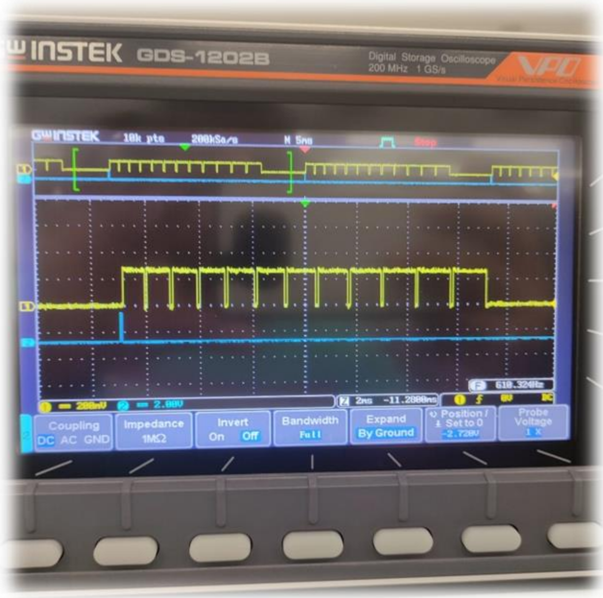



Figure 5: Normal Mode

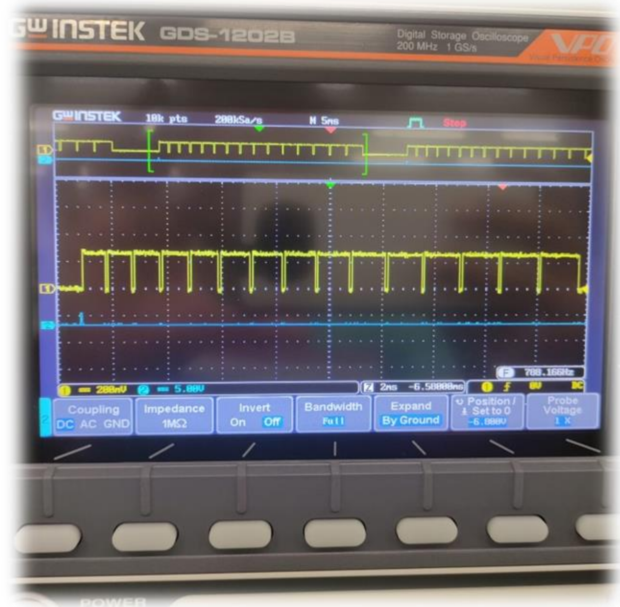



Figure 4: Changed Mode


Figure 6: Zoomed Pulse

## Task 3

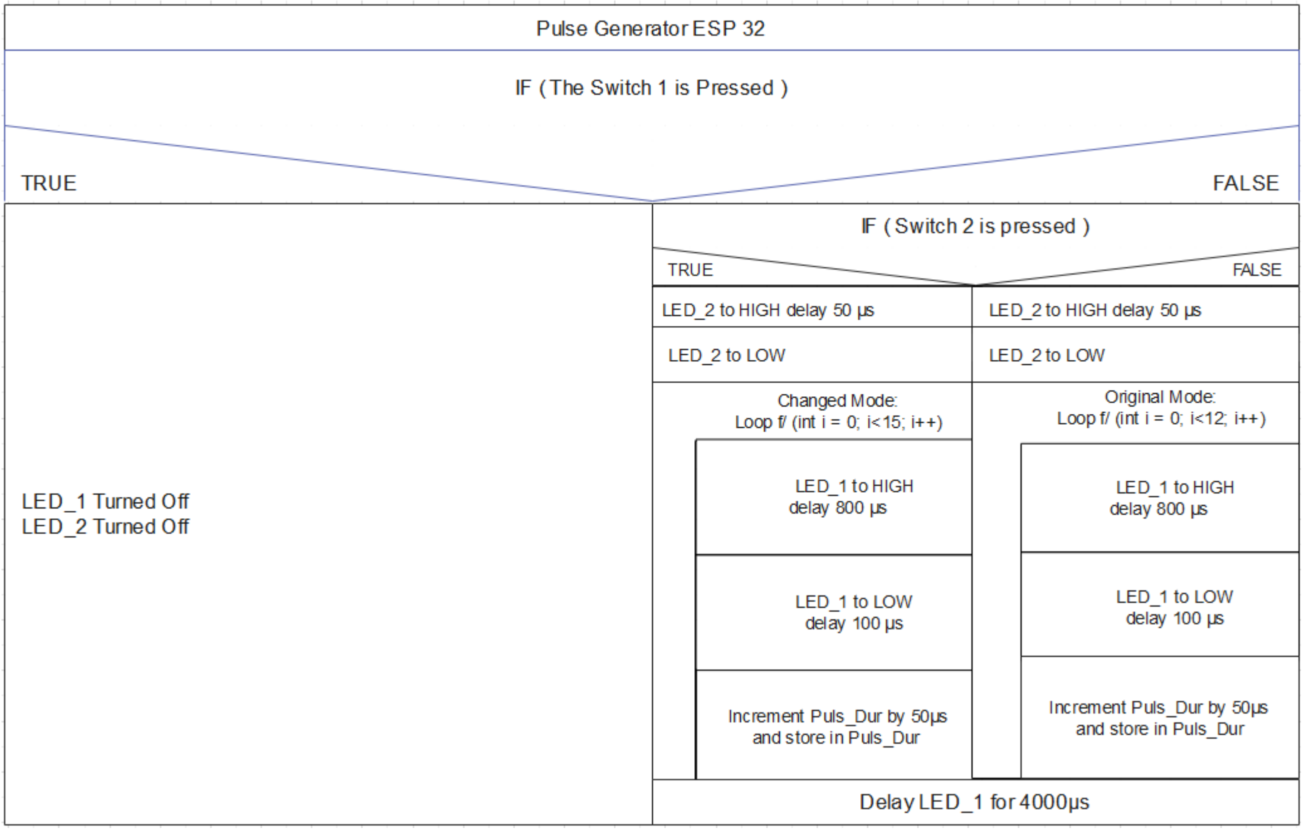



*Figure 7: Nassi Schneiderman Chart*

## Github Link :

[EmbeddedSoftware/Assignment1ESP32 at main · SanjeevKay19/EmbeddedSoftware (github.com)]