🏆 A.3.2 Actividad de aprendizaje

Circuito sensor de tacto a través de un NodeMCU ESP32

Instrucciones

- Basado en la figura 1, ensamblar un sistema, capaz de responder al tacto, a través de un circuito electrónico, utilizando un NodeMCU **ESP32**, un **Sensor de tacto capacitivo**.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A3.2_NombreApellido_Equipo.pdf.**
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme**.md dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
- readme.md
  - blog
    - C3.1_TituloActividad.md
    - C3.2_TituloActividad.md
    - C3.3_TituloActividad.md
    - C3.4_TituloActividad.md
    - C3.5_TituloActividad.md
    - C3.6_TituloActividad.md
    - C3.7_TituloActividad.md
    - C3.8_TituloActividad.md
  - img
  - docs
    - A3.1_TituloActividad.md
    - A3.2_TituloActividad.md
    - A3.3_TituloActividad.md
```

Fuentes de apoyo para desarrollar la actividad

- ☑ Sensor de tacto capacitivo

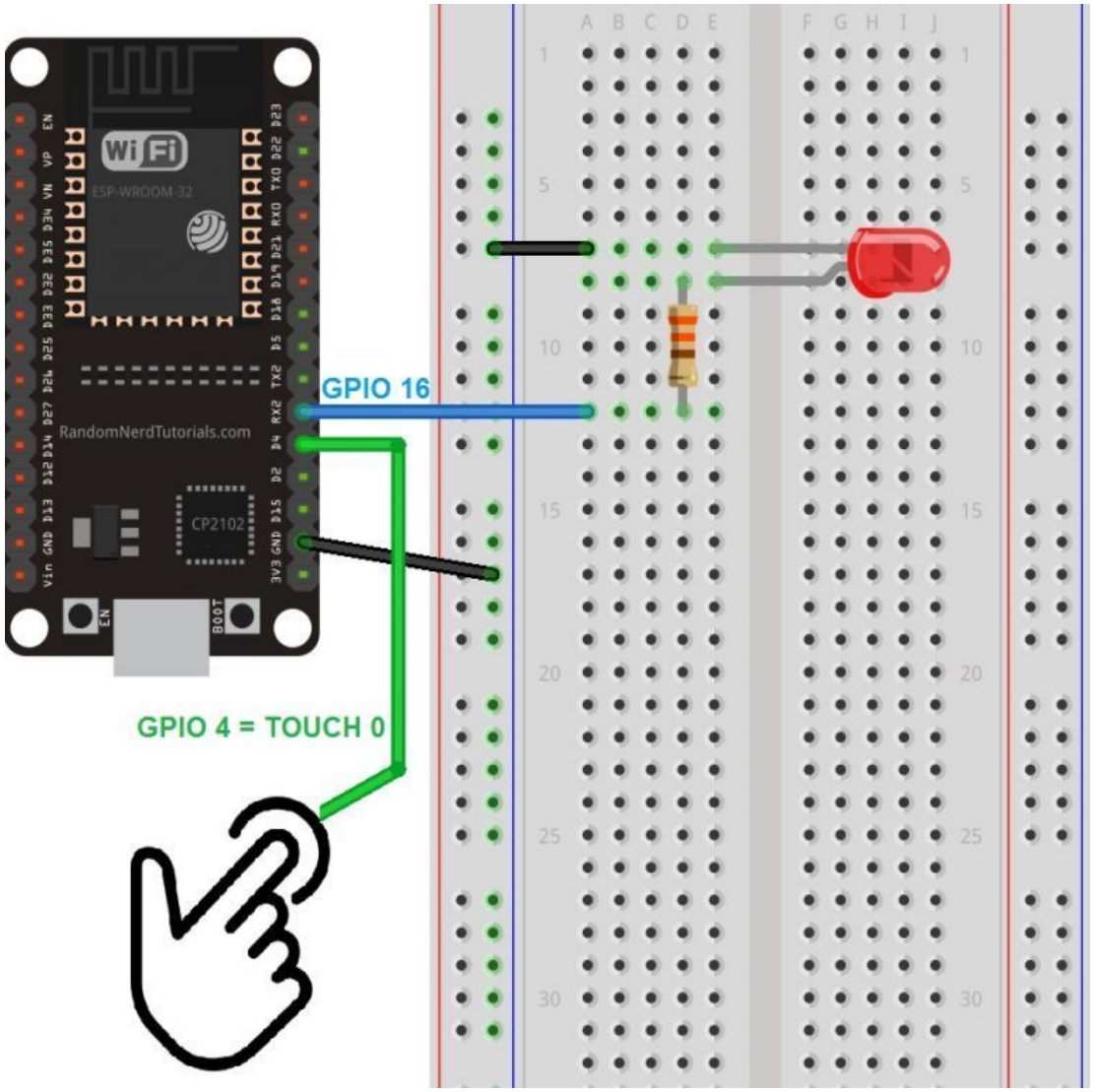
- [x] Salida analogica PWM

Desarrollo

1.Utilice el siguiente listado de materiales para la elaboración de la actividad

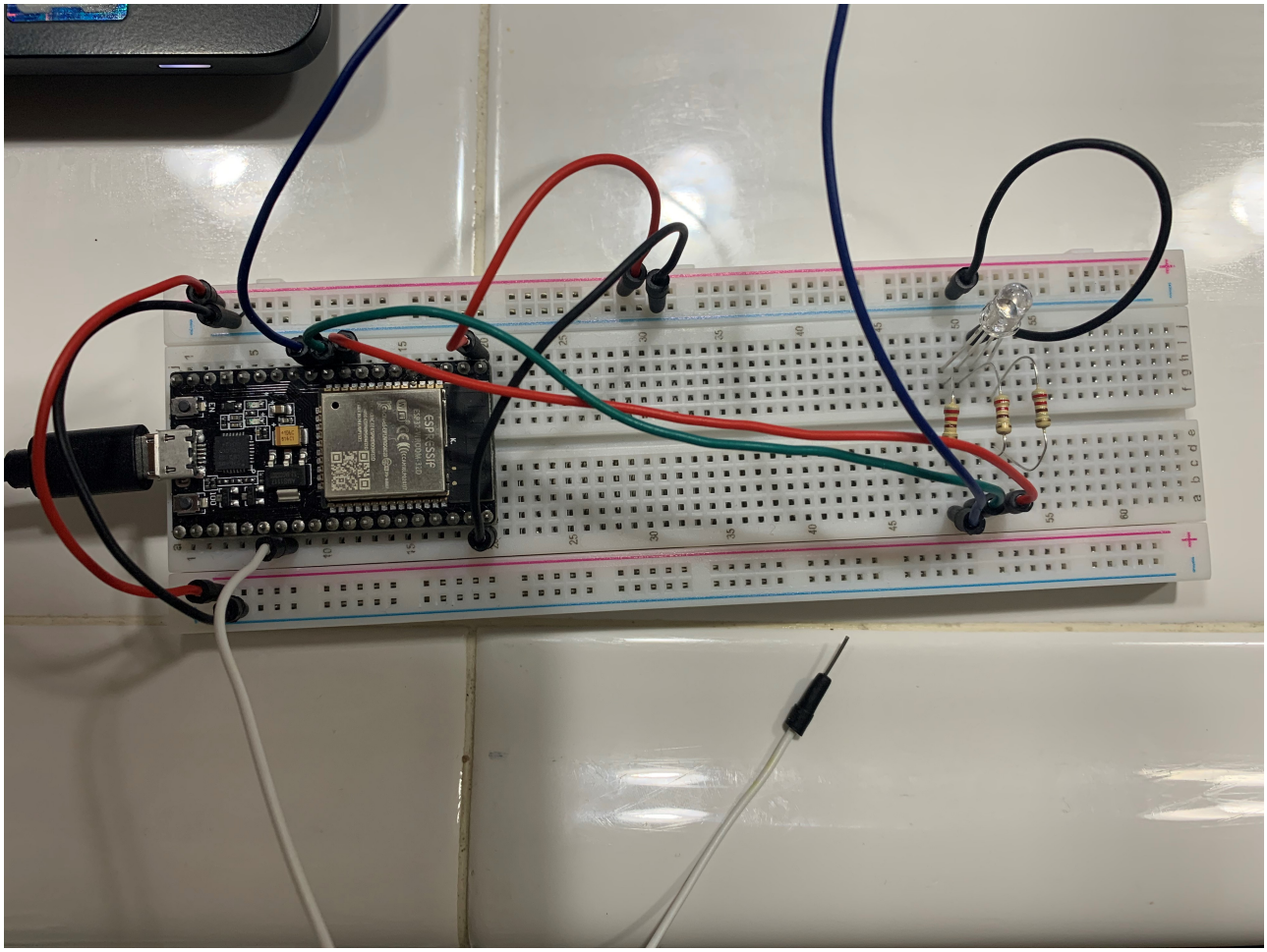
Cantidad	Descripción
1	Diodo led Rojo
1	Resistencia 330 ohms
1	Fuente de voltaje de 5V
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M
1	Hoja de aluminio

Figura 1 Circuito ESP32 IC L293 Motor DC



2. Una vez ensamblado el circuito anterior, realice un programa que permita al ensamble cumplir con las siguientes condiciones:
    - El sistema deberá ser capaz de encender y apagar **un led** al presionar el sensor de tacto.
    - El sistema deberá tener la característica que si el Led esta encendido, al tocar el sensor de tacto, este iniciara un secuencia de intermitencia de 3 segundos (es decir se apaga 1 segundo, se enciende un segundo y se apaga finalmente).
    - El sistema deberá contar con la característica que si el Led esta apagado, al tocar el sensor de tacto, este se encenderá poco a poco hasta llegar a su nivel máximo de iluminación.
3. Coloque aquí evidencias que considere importantes durante el desarrollo de la actividad.

- Video de youtube

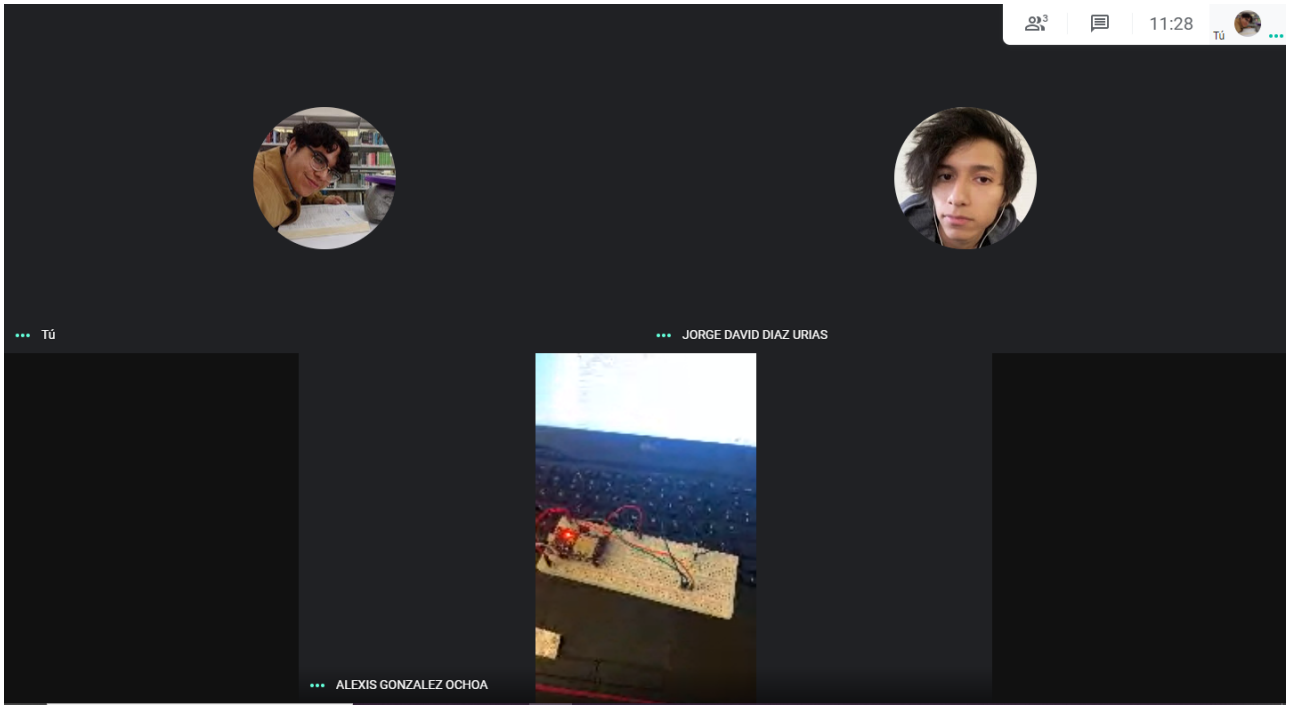


```
#define LEDR 25 // Se declaran los pines para cada parte del led RGB
#define R_channel 0 // Canales dedicados a cada uno de los colores
#define pwm_Frequency 5000 // Frecuencia
#define pwm_resolution 8 // 8 bit de resolucion
bool estado = true;
#define touchPin T0  // A donde va conectado el sensor de tacto
const int threshold = 30;
int touchValue;      // El valor que se recogera del sensor de tacto

void setup()
{
  Serial.begin(115200);
  delay(1000);
    // Configuracion de las funciones del led RGB
  ledcAttachPin(LEDR, R_channel);
  // Conectar el canal al GPIO para controlarlo
  ledcSetup(R_channel, pwm_Frequency, pwm_resolution);
}

void loop()
{
   touchValue = touchRead(touchPin);  // Tomar el valor analogico del sensor y se le asigna a la variable
   Serial.println(touchValue);        // Imprime el valor en el monitor

   if(touchValue < threshold) // lee el valor del estado: si este es menor al dado entrara al siguiente paso.
   {
      if(estado) // Lee el estado como true y entra al ciclo para prender el led.
    {
      Serial.println(" - LED on");
        for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++)  // Enciende el led hasta su punto maximo
        {
          ledcWrite(R_channel, dutyCycle);
          delay(15);
          estado = false; // Cambia al estado a false y termina sale.
        }
    }
      else
      {
        if(!estado)  // Toma el estado negado false y para hacerlo true
        {            // Entonces entrara y apagara el led.
         ledcWrite(R_channel, 0);
         delay(1000);
         ledcWrite(R_channel, 255);
         delay(1000);
         ledcWrite(R_channel, 0);
         Serial.println(" - LED off"); // imprime que el led se encuntra apagado
         estado = true;
        }
      }
   }
   delay(500);
}
```



1. Conclusiones

Alexis Gonzalez Con esta práctica realizada se puso a prueba las dos prácticas anteriores, la C3.7 y la C3.8. Para mi fue una pequeña combinación de ambas y un valor extra que viene siendo el Booleano para poder complementar lo que faltaba para que el led se mantuviera encendido una vez tocado el sensor y se apagar hasta que se volviera a tocar. Esto es como las lámparas para escritorio o las que van a lado de la cama. Y esto da una idea de todo lo que podemos hacer con el microcontrolador ESP32 ya que integra muchas funciones con las que se pueden experimentar.

Jorge Diaz Con la realización de esta práctica nos podemos dar cuenta cuenta de el funcionamiento que puede tener un esp32 y los GPIO que lo conforman siendo capaz de detectar cualquier cosa que tenga una carga eléctrica y en el caso de esta práctica se demuestra que incluso puede detectar con la piel humana usando los dedos, en el esp32 podemos usar cualquier pin como pin de entrada teniendo así un valor máximo después de hacer contacto con el pin y un valor mínimo sin hacer contacto siendo así un componente en el que se puedan realizar diferentes variaciones para prácticas interesantes.

Julio Jimenez En conclusión podemos decir que en esta práctica se puede observar una de las funciones principales del ESP32 y ver porqué es tan flexible su uso. En la práctica se utilizó el sensor para poder detectar el tacto y así accionar el resto del circuito. Aunque por otra parte la programación implementada fue esencial para el funcionamiento y aprendizaje de la práctica.

💣 Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10

🏠 Ir a GitHub de Julio Jimenez

🏠 Ir a GitHub de Jorge Diaz

🏠 Ir a GitHub de Alexis Gonzalez