



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](#)

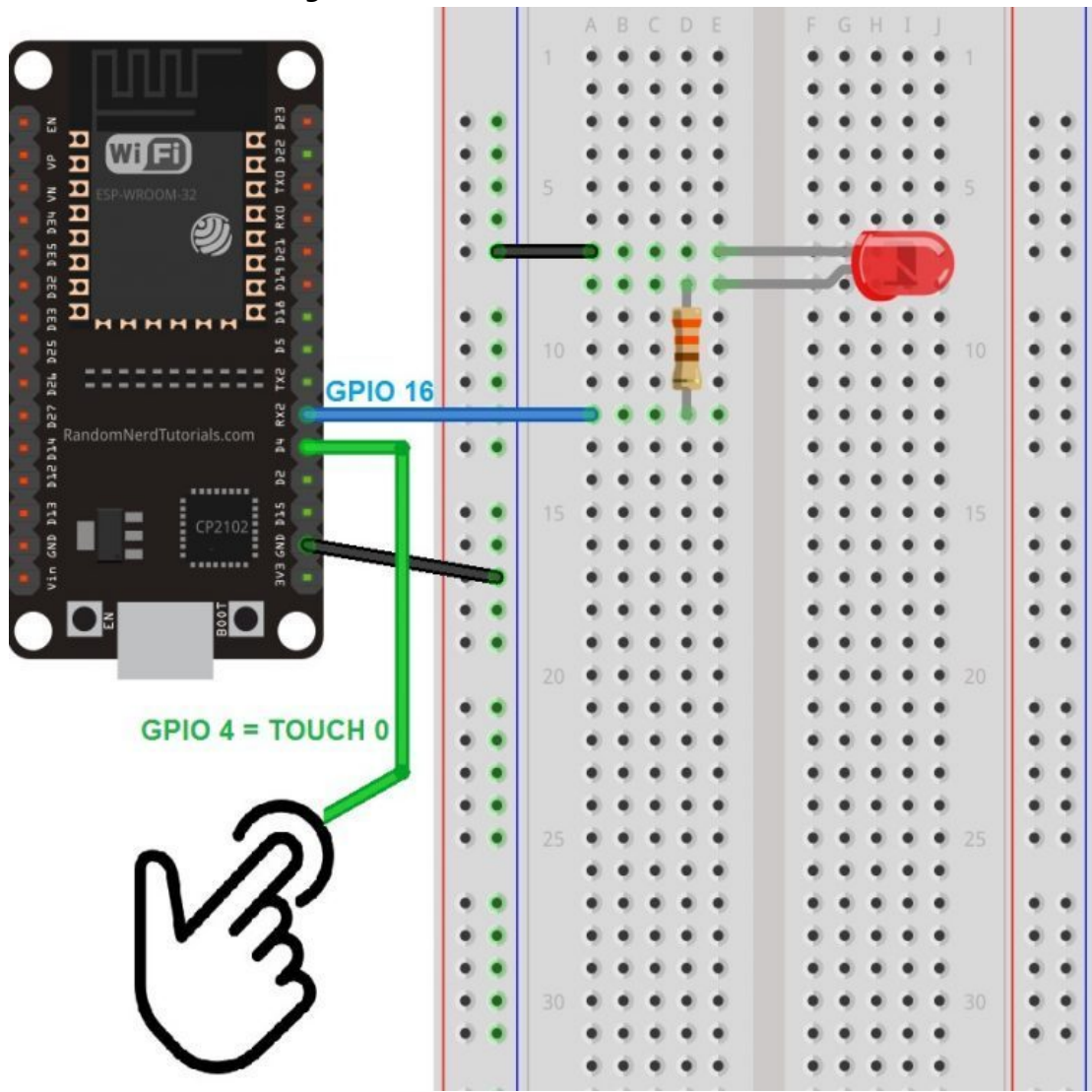
- ☒ Salida analogica PWM

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.



Codigo

```
//declaracion de variables
const int touchPin = 4;
const int ledPin = 16;
//Se estable el numero maximo que retorna
//el ESP32 cuando se toca el sensor, el cual
//es el punto entre los valores de cuando
//se toca el sensor y cuando no.
const int limite = 20;
int touchValor;
const int freq = 5000;
const int ledCanal = 0;
const int resolucion = 8;

void setup() {
  Serial.begin(9600);
  delay(1000);
  //Se establece el pin donde esta conectado el led como
  //pin de salida.
  pinMode(ledPin, OUTPUT);

  //se utiliza la siguiente funcion para inicializar la configuracion para
  //el PWM, tomando como parametros el canal, la frecuencia y la resolucion
  //respectivamente.
  ledcSetup(ledCanal,freq, resolucion);

  //Se relaciona el pin GPIO con el canal del PWM.
  ledcAttachPin(ledPin, ledCanal);
}

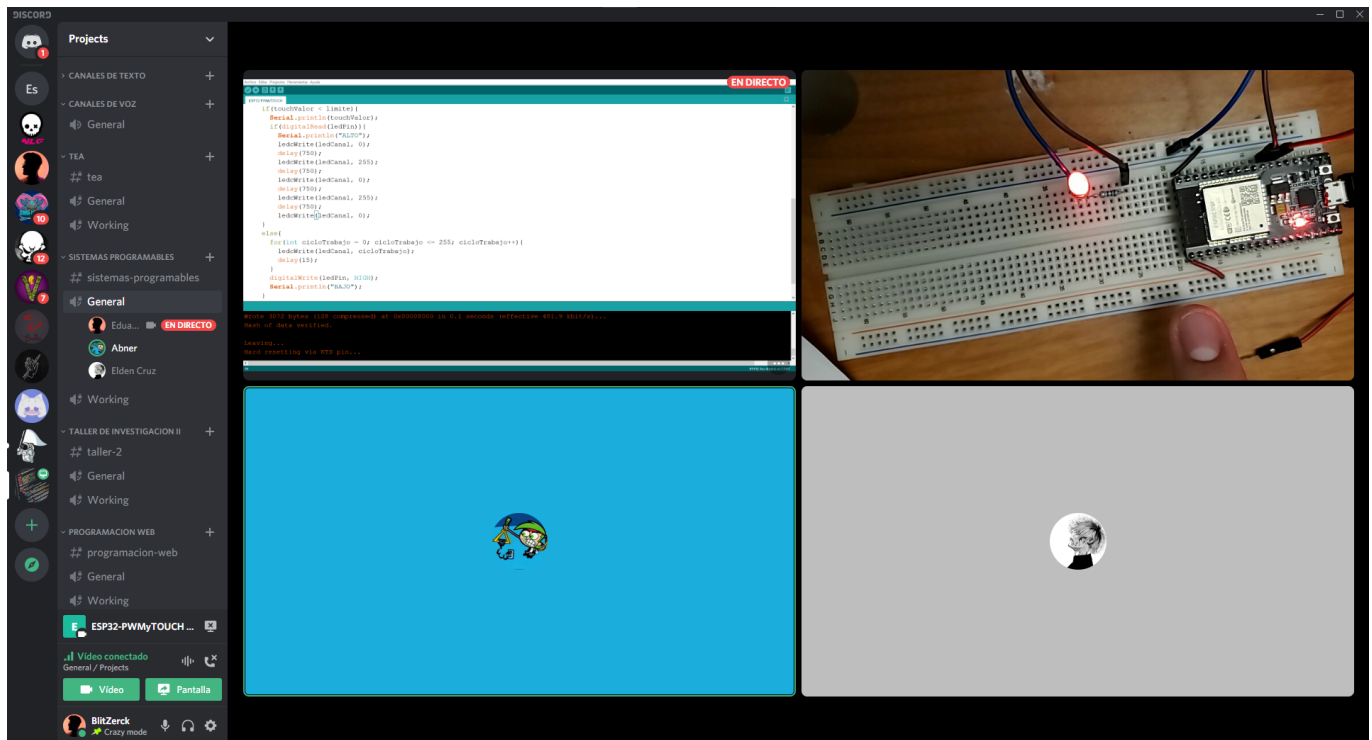
void loop() {
  //Se guarda en una variable el valor que retorna
  //la funcion de touchRead que necesita como parametro
  //el pin que detecta el touch.
  touchValor = touchRead(touchPin);

  //Si este valor medido es menor al limite establecido
  //enciende el LED.
  if(touchValor < limite){
```

```
//Si el estado del led es HIGH, entra a la primera
//condicion.
if(digitalRead(ledPin)){
    //Se enciende y se apague el LED en un tiempo de
    //3 segundos.
    ledcWrite(ledCanal, 0);
    delay(750);
    ledcWrite(ledCanal, 255);
    delay(750);
    ledcWrite(ledCanal, 0);
    delay(750);
    ledcWrite(ledCanal, 255);
    delay(750);
    ledcWrite(ledCanal, 0);
}
//Si el estado del led es LOW...
else{
    //ciclo para subir la intensidad del LED de 0 a 255 porque se eligio
    //una resolucion de 8 bits = 256
    for(int cicloTrabajo = 0; cicloTrabajo <= 255; cicloTrabajo++){
        ledcWrite(ledCanal, cicloTrabajo);
        delay(15);
    }
}
}
```

3. Coloque aquí evidencias que considere importantes durante el desarrollo de la actividad.

The image is a screenshot of a Discord application window. On the left is the server sidebar for 'sistemas-programables', listing channels like 'General', 'Working', and 'programacion-web'. The main content area is divided into three video feeds. The top-left feed shows a breadboard with an Arduino Uno and a red LED. The top-right feed is a solid blue screen with a small cartoon character. The bottom feed is a grey screen with a circular profile picture. At the bottom, the status bar indicates 'Video conectado' and provides buttons for 'Video' and 'Pantalla'.



4. Incluya las conclusiones individuales y resultados observados durante el desarrollo de la actividad.

- *Morgado Jacome Eduardo*

La realizacion de la actividad no fue complicada ya que era una combinacion de las ultimas practicas que realizamos con el ESP32, lo cual llevo a posicionar todo nuestro esfuerzo en la logica del codigo para que hiciera lo que pide la actividad.

En este punto primero pensamos en como detectar cuando el LED esta encendido o apagado, donde mi compañero Elden fue de gran ayuda al investigar la documentacion de arduino y decirnos que la funcion "digitalRead" retornaba un booleano para saber si estaba encendido o no. Tambien fue de util ayuda investigar el funcionamiento de las funciones de la libreria de LED Control de Espressif ya que nos ayudo a saber como debiamos encender y apagar el Led con PWM. Una vez mas confirmo que el hacer una investigacion previa de los componentes electronicos y las funciones del entorno de desarrollo es de gran ayuda para evitar problemas durante la realizacion del ejercicio, y mas aun cuando estamos utilizando componentes fisicos con el peligro de estropearlos.

- *Perales Niebla Abner Jesus*

Esta práctica me resulto especialmente emocionante; a pesar de que, al igual que los demás equipos, solo un integrante tiene los componentes físicos, trabajamos juntos para realizar el proyecto. Esto se debe en gran manera a que mi compañero Morgado, quien tiene los componentes, nos comparte en vivo de la cámara con la que presenta el circuito y a la vez su pantalla, donde tiene el código. De esta forma podemos interactuar con el circuito y el código los otros dos. Sobre la práctica en sí, me tocó comenzar con el código, y me ayudó mucho el hecho de que Morgado nos haya explicado los circuitos anteriores, así como que tenga muy bien comentado su código. Eso me permitió combinar el código de los programas anteriores y, aunque tuvimos que hacer modificaciones al código, no provoqué ningún error de sintaxis, a pesar que yo trabajo las simulaciones con arduino. Luego los 3 como equipo, nos pusimos a razonar las instrucciones y cada quien aportó algo importante para la realización del programa.

- *Cruz Vera Elden Humberto*

Durante la realización de la práctica se tuvo un pequeño inconveniente al realizar la intermitencia y apagado del led. Debido a que el programa si podía determinar cuando el led estuviera encendido con la función digitalRead, pero a la hora de utilizar digitalWrite para apagar el led no sucedía nada, esto se corrigió con la función ledcWrite. Y me pareció interesante que al cambiar el valor del canal del led de 0 a un valor distinto no había un cambio, lo único que podía producir una diferencia notable era cambiar el valor límite para detectar un objeto con el sensor del ESP32.



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 inicio](#)



[Repositorio de Github de Morgado Jacome Eduardo](#)



[Repositorio de Github de Cruz Vera Elden Humberto](#)



[Repositorio de Github de Perales Niebla Abner Jesús](#)

