🏆 A.3.2 Learning activity

Tact sensor circuit by a NodeMCU ESP32
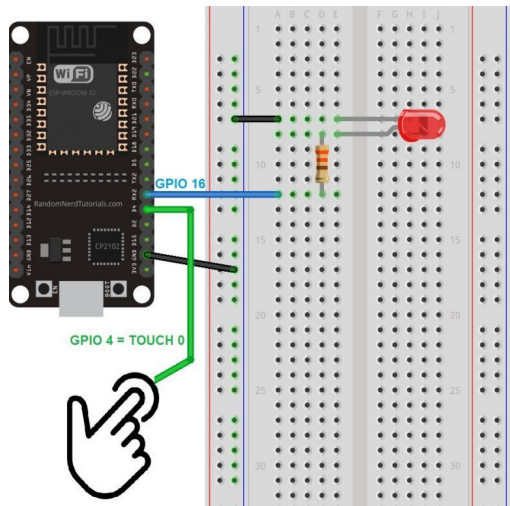
Complimentary sources for developing the activity

- ☑ Capacitive Touch Sensor
- ☑ Analog Output PWM

Development

1. Use the following list of materials for the activity's development

Quantity	Description
1	Red LED diode
1	330 ohms Resistor
1	5V Power supply
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M
1	Aluminum foil

2. Once the previously shown circuit is assembled, create the code needed for the hardware complies with the following conditions:

   - The sistem should be able to turn ON and OFF **an LED** when interacting with the touch sensor.
   - The system should start a 3 second long sequence (in which it turns OFF 1 second, then ON 1 second, then OFF), if the sensor is touched while the LED is turned ON.

o The system should gradually increase the LED's brightness until it reaches its maximum state, if the sensor is touched while de LED is turned OFF.

3. Insert important evidence during the activity's development, with pictures, such as reunions.

```cpp
// Configuracion pin GPI04 y GPI016
const int touchPin = 4;
const int ledPin = 16;
// Configuracion del limite de deteccion
const int limite = 20;
// Guarda el valor del touchPin
int touchValue;
// Controla el estado del LED
int estado = 0; // Inicia apagado
// Configuracion de propiedades del PWM
const int ledChannel = 0; // Canal del LED
const int freq = 5000; // Frecuencia del canal del LED
const int resolution = 8; // Resolucion CDT
```

```cpp
void loop(){
  // Guarda la lectura del touchPin
  touchValue = touchRead(touchPin);
  Serial.println(touchValue);
  // Encender led primera vez
  if((touchValue < limite) && estado == 0){ // 2.1 y 2.3
    // Enciende el LED de menor a mayor iluminacion
    for(int i = 0; i < 255; i++){ // Rango = (2^Resolucion)-1
      ledcWrite(ledChannel, i);
      delay(15);
    }
    Serial.println(" - LED on");
    estado = 1; // Indica que el LED esta encendido
    touchValue = 30;
  }
```

```cpp
void setup(){
  Serial.begin(115200); // Comienza conexion serial
  delay(1000); //
  // Configura las especificaciones del LED PWM
  ledcSetup(ledChannel, freq, resolution);
  // Conecta el canal PWM del LED al pin GPIO
  ledcAttachPin(ledPin, ledChannel);
}
```

```cpp
  // Detecta si el led esta encendido para apagarlo interminentemente
  if((touchValue < limite) && estado == 1){ // 2.2
    ledcWrite(ledChannel, 0);
    delay(1000);
    ledcWrite(ledChannel, 255);
    delay(1000);
    ledcWrite(ledChannel, 0);
    estado = 0; // Indica que el LED esta apagado
  }
  delay(500);
}
```
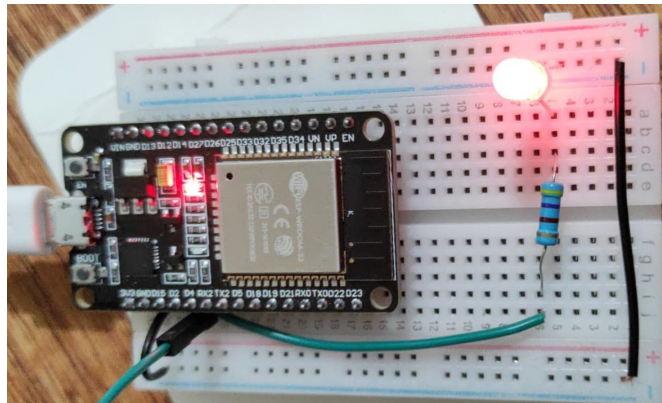

4. Insert picture **evidence** from the team meetings while developing the activity.
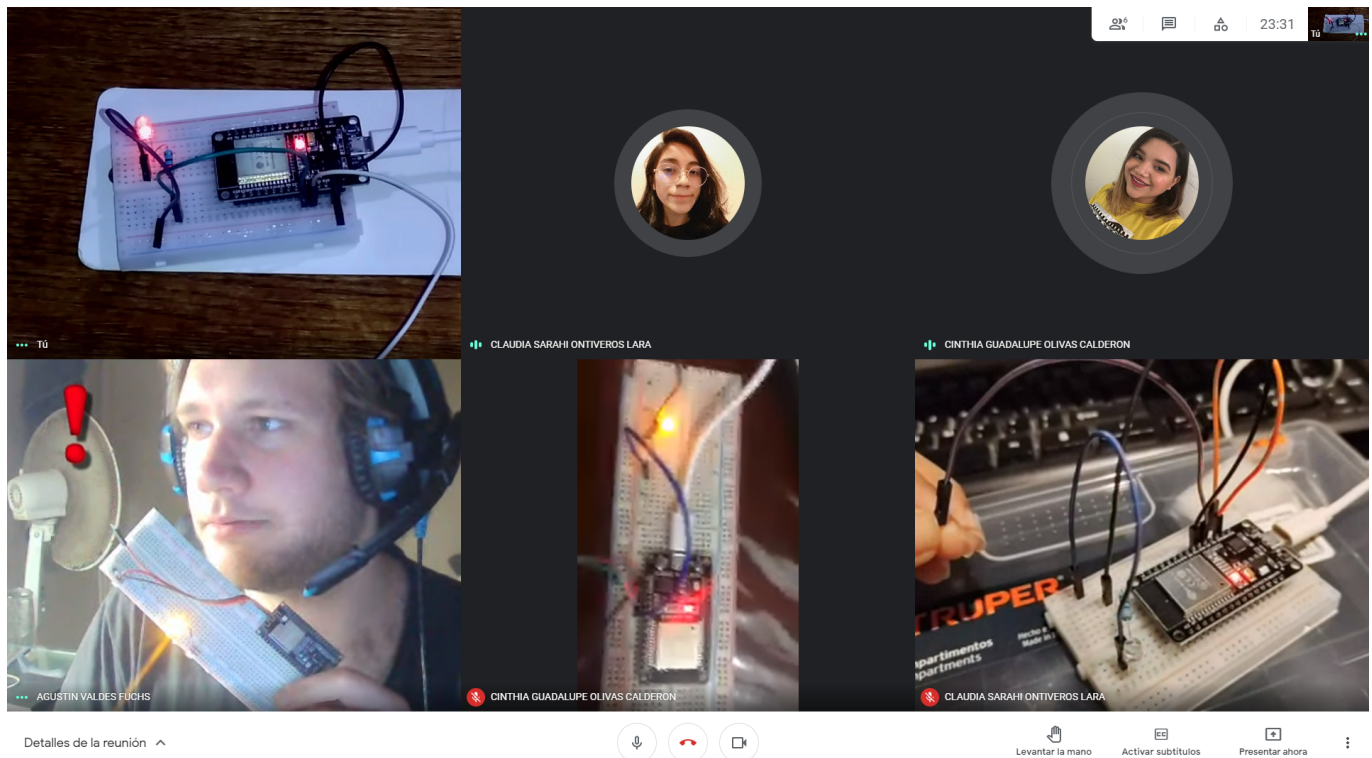
Los grumosos 🐻 Conclusions.

Nava Reyes Carlos

> In this practice an LED was used and as controller an ESP32, were put into
> practice elements that we had previously used as is the concept of the PWM, with
> the help of certain special features that this controller has as are touch-
> sensitive pins, due to these features an LED can be turned on and off by touching
> a cable, Thanks to these features of the controller we can perform the practice
> correctly.

Olivas Calderon Cinthia Guadalupe

> To carry out this practice, we had to make a circuit in which an LED was lit by
> touch but with different conditions, for example that it could be turned on and
> off, had a 3-second flashing sequence, and the LED would light up for a short
> time. little by little until reaching its maximum level. To achieve this we did it
> through code in the Arduino IDE and we were placing different instructions, from
> what I could tell is that sometimes the cable was not touched and the Led turned
> on by itself. Investigating we were able to do the activity correctly.

Ontiveros Lara Claudia Sarahi

> In carrying out this practice we used the ESP32 and a program called Arduino IDE
> to turn on an LED when touching a jumper, in addition to modifying the code used

> to achieve some requirements that we requested, such as modifying the LED lighting or varying its status. It was an interesting practice, because we used physical components.

Valdés Fuchs Agustín

> In this practice we used the ESP32, along with an LED, which when using the TOUCH sensitive pins as a sensor, came up with a circuit capable of simulating a touch lamp, along with code designed to be used for PWM functions which allow the LED to power up gradually.

🐙 Github links

🏫 **Carlos Nava**

👸🏼 **Cinthia Olivas**

🙀 **Claudia Ontiveros**

👽 **Agustin Valdés**