🏆 A.3.2 Learning activity

Touch sensor circuit through a NodeMCU ESP32

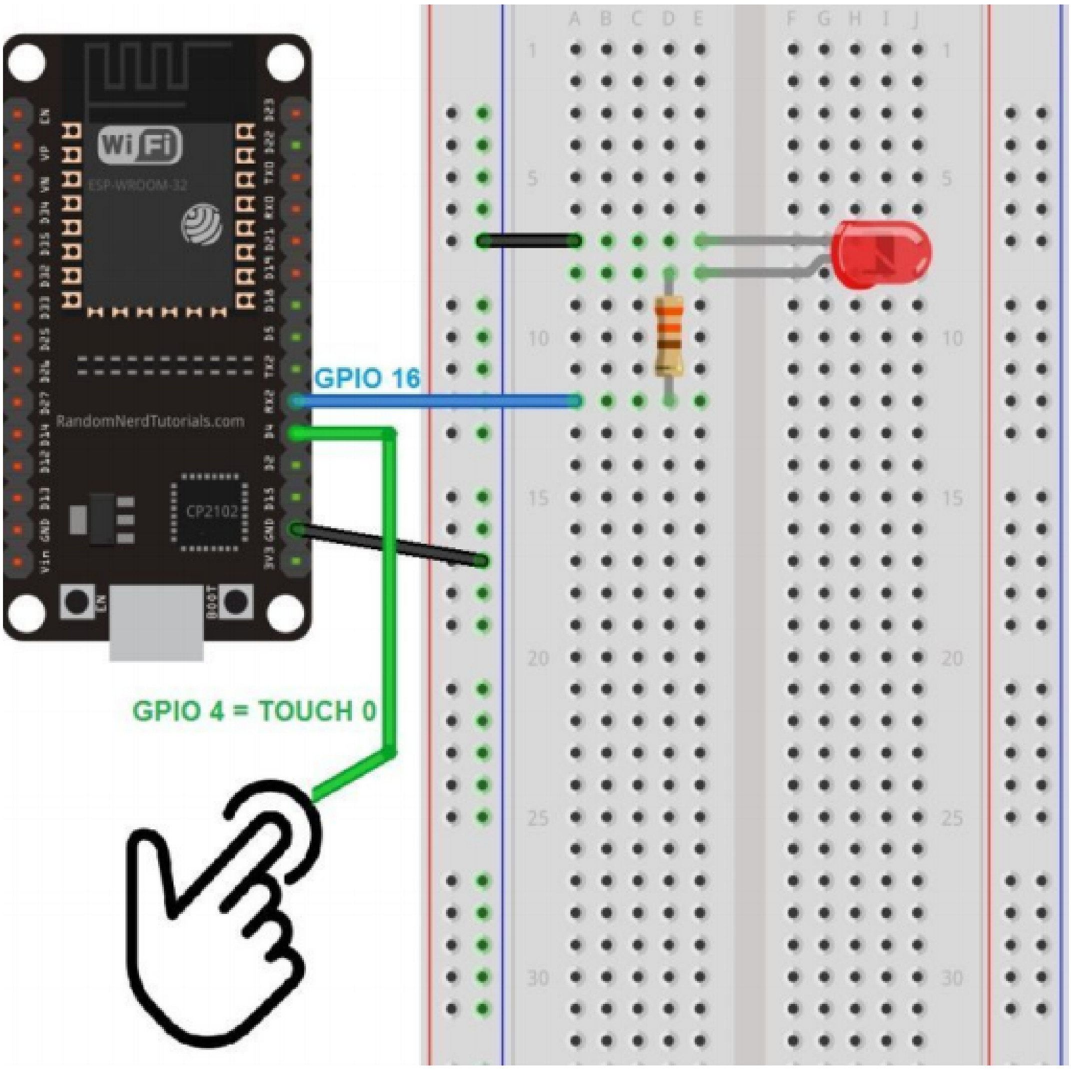
Sources of support to develop the activity

- ☑ Capacitive touch sensor
- ☑ PWM analogue output

Development

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

Quantity	Description
1	Red led diode
1	330 ohms resistance
1	Power supply of 5v
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M
1	Aluminium sheet

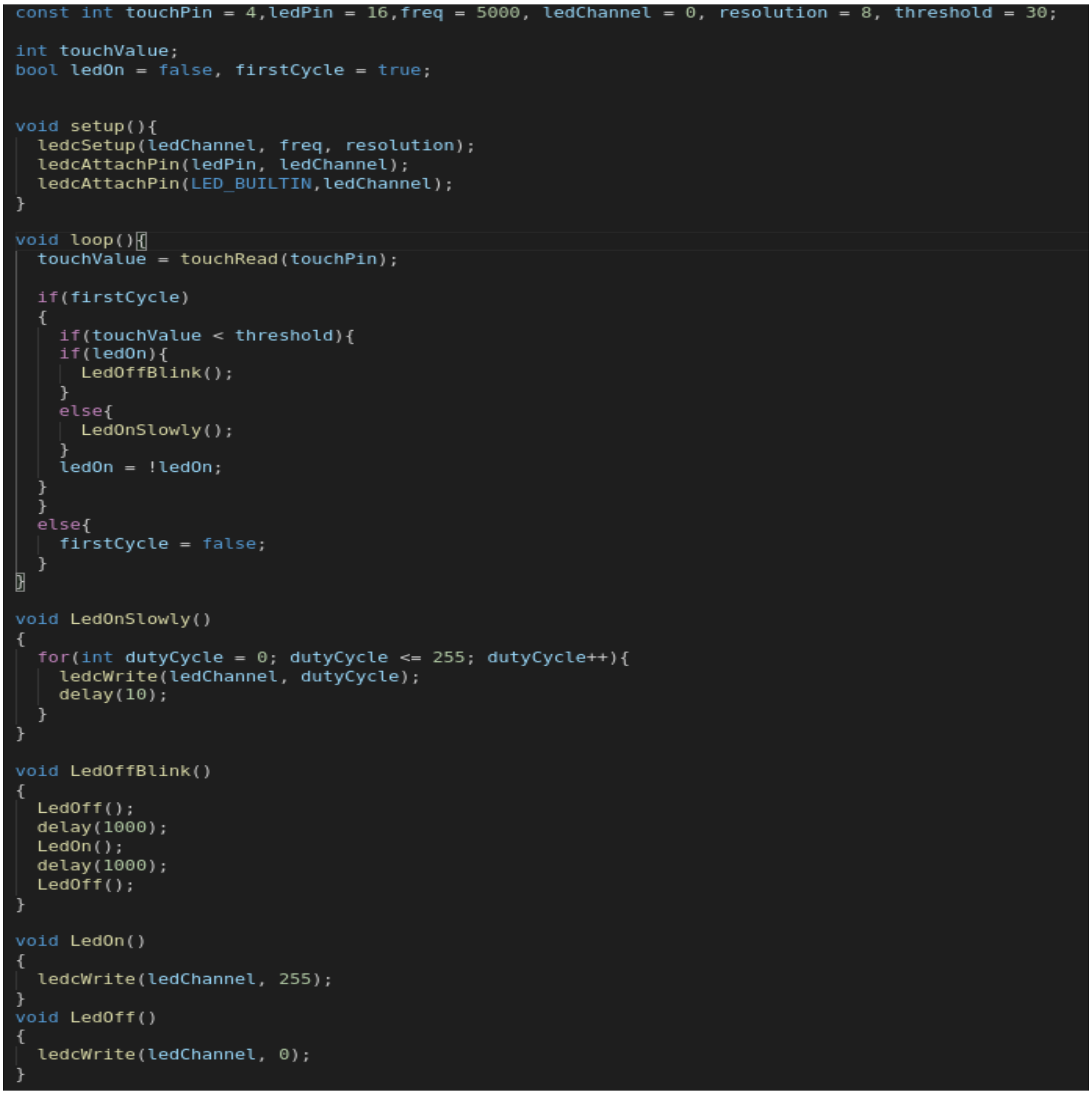


2. Once the previous circuit is assembled, perform a program that allows the assembly to meet the following conditions:
    - The system shall be able to switch on and off **one led** when pressing the touch sensor.
    - The system must have the characteristic that if the Led is on, when touching the touch sensor, it will start a sequence of flashing of 3 seconds (that is to say it turns off 1 second, it turns on a second and it turns off finally).
    - The system must have the characteristic that if the Led is turned off, when touching the touch sensor, it will turn on gradually until reaching its maximum lighting level.

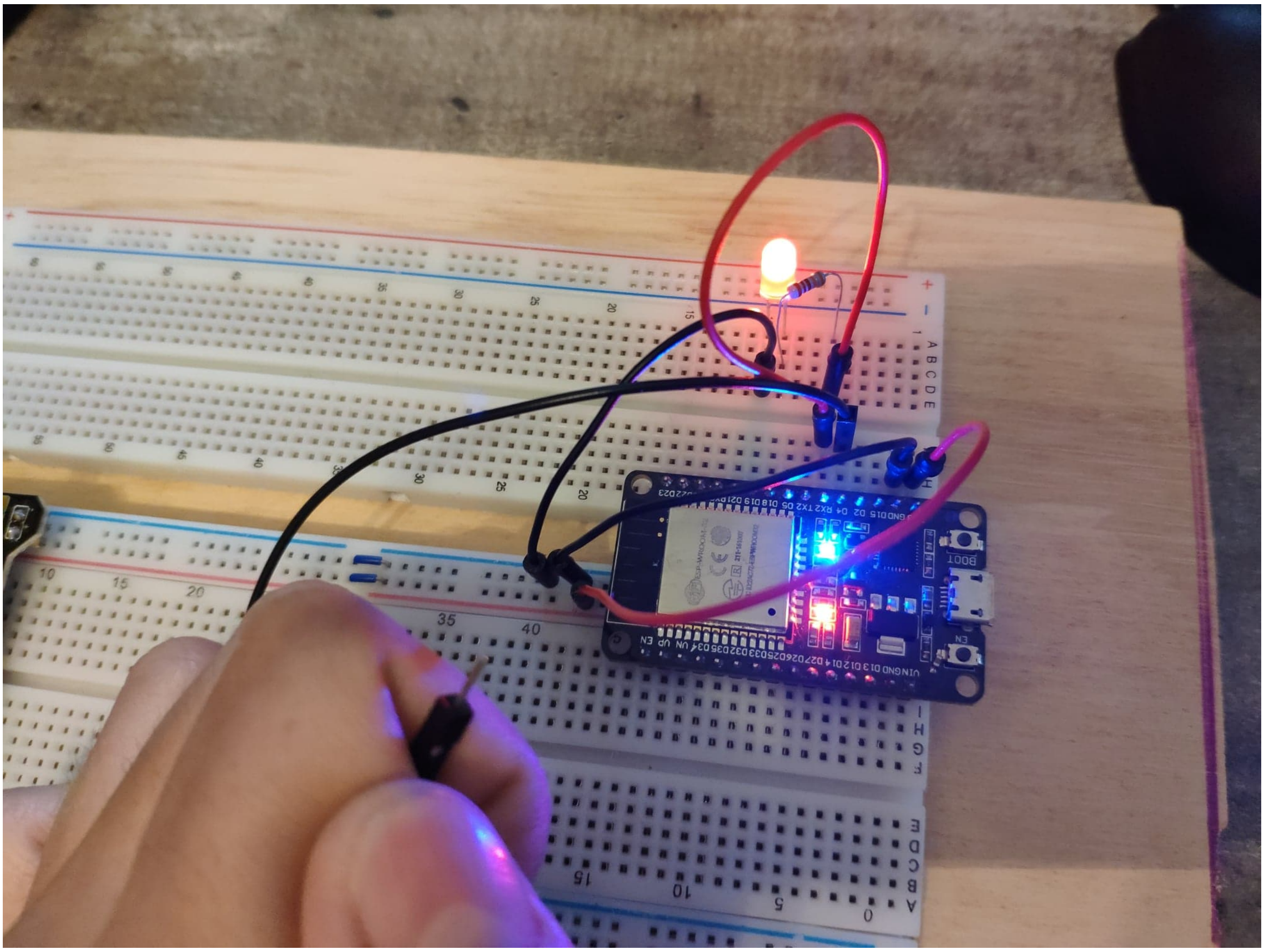
```
const int touchPin = 4, ledPin = 16, freq = 5000, ledChannel = 0, resolution = 8,
threshold = 30;

int touchValue;
bool ledOn = false, firstCycle = true;

void setup()
{
  ledcSetup(ledChannel, freq, resolution);
  ledcAttachPin(ledPin, ledChannel);
  ledcAttachPin(LED_BUILTIN, ledChannel);
}
```

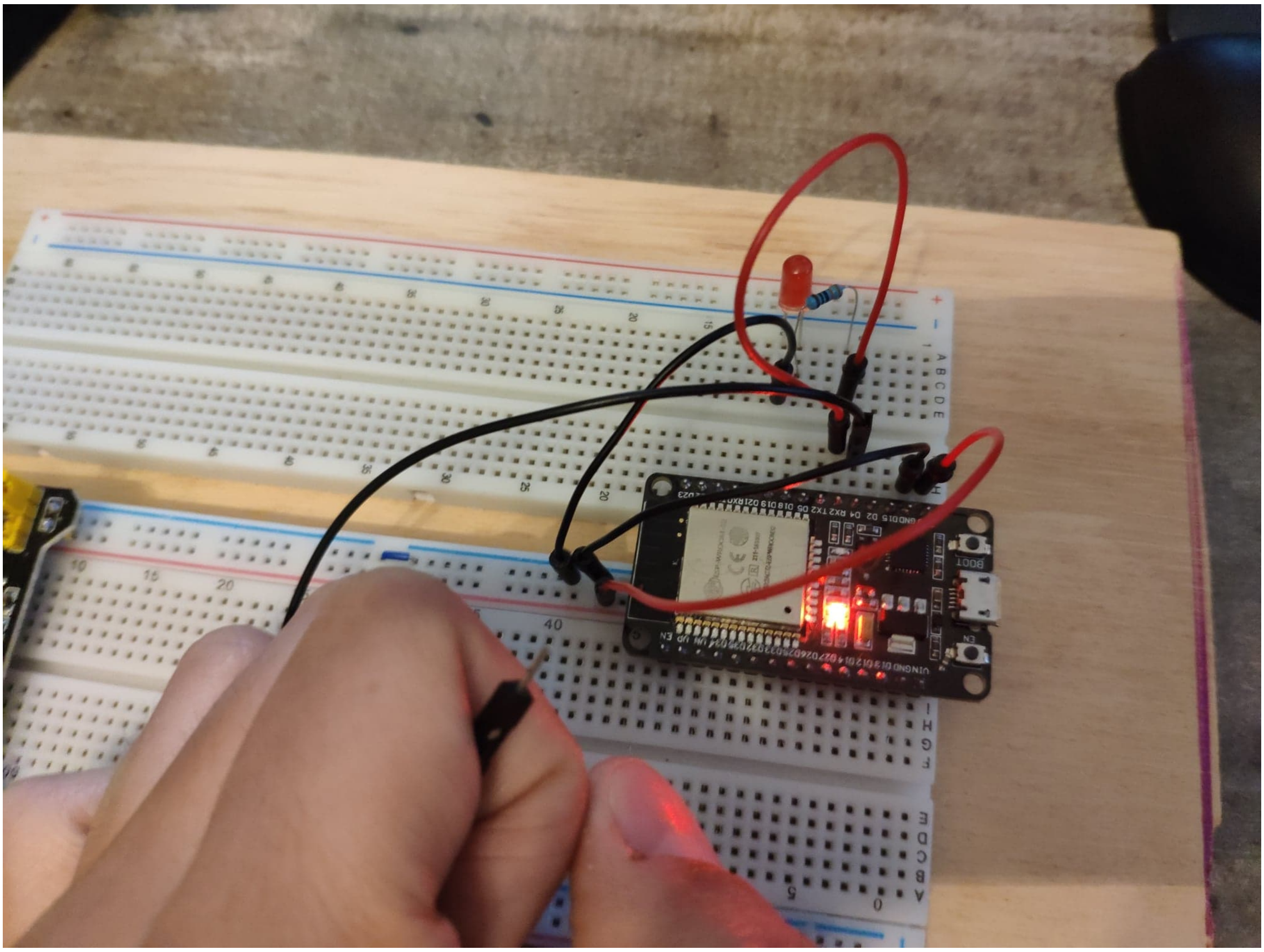
```
void loop()
{
  touchValue = touchRead(touchPin);
  if (!firstCycle)
  {
    if (touchValue < threshold)
    {
      if (ledOn)
      {
        LedOffBlink();
      }
      else
      {
        LedOnSlowly();
      }
      ledOn = !ledOn;
    }
  }
  else
  {
    firstCycle = false;
  }
}

void LedOnSlowly()
{
  int dutyCycle = 0;
  while(dutyCycle <= 255){
    ledcWrite(ledChannel, dutyCycle);
    dutyCycle++;
    delay(15);
  }
  // for (int dutyCycle = 0; dutyCycle <= 255; dutyCycle++)
  // {
  //   ledcWrite(ledChannel, dutyCycle);
  //   delay(15);
  // }
}

void LedOffBlink()
{
  LedOff();
  delay(1000);
  LedOn();
  delay(1000);
  LedOff();
}

void LedOn()
{
  ledcWrite(ledChannel, 255);
}
void LedOff()
{
  ledcWrite(ledChannel, 0);
}
```

3. Place here evidence that you consider important during the development of the activity

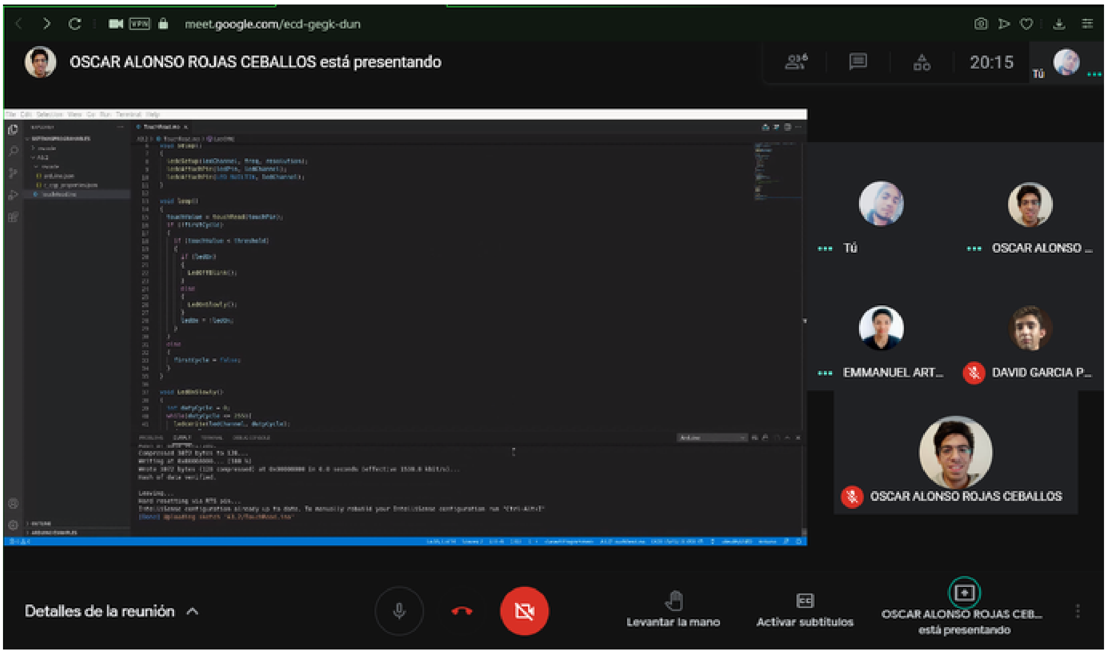


```
const int touchPin = 4,ledPin = 16,freq = 5000, ledChannel = 0, resolution = 8, threshold = 30;

int touchValue;
bool ledOn = false, firstCycle = true;


void setup(){
  ledcSetup(ledChannel, freq, resolution);
  ledcAttachPin(ledPin, ledChannel);
  ledcAttachPin(LED_BUILTIN,ledChannel);
}

void loop(){
  touchValue = touchRead(touchPin);

  if(firstCycle)
  {
    if(touchValue < threshold){
    if(ledOn){
      LedOffBlink();
    }
    else{
      LedOnSlowly();
    }
    ledOn = !ledOn;
  }
  }
  else{
    firstCycle = false;
  }
}

void LedOnSlowly()
{
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    ledcWrite(ledChannel, dutyCycle);
    delay(10);
  }
}

void LedOffBlink()
{
  LedOff();
  delay(1000);
  LedOn();
  delay(1000);
  LedOff();
}

void LedOn()
{
  ledcWrite(ledChannel, 255);
}
void LedOff()
{
  ledcWrite(ledChannel, 0);
}
```





4. Insert images of **evidences** such as meetings of team members held for the development of the activity.



Conclusions

Axel: To conclude I can say that in practice we learned to use the ESP32 as a touch sensor, in which for demonstration way we turned on an LED, which had some specific characteristics in its on and off to practice the digital and analog outputs of the same microcontroller.

David: In the development of the activity we were able to observe both the operation of the ESP32 and the touch sensor, in both we could observe the result and the facility of the programation, our only difficulty was the programming of the ESP32 but after a small learning curve we were able to to program it.

Emmanuel: In conclusion, the activity was based on the implementation of an ESP32 module and a touch sensor, which were programmed so that an LED would light up slowly each time a piece of aluminum was touched and turn off when it stopped playing. It was complicated when it did not detect the ESP32 but it was possible to solve and achieve the objective.

Oscar Rojas: With the esp32 we can read an electricar charge via the touch pin from the esp32 with this we can trigger some actions, in this case we turned on a led and give it some custom actions when it turned on or off.

Repositories

Axel Go to my repository

David Go to my repository

Emmanuel Go to my repository

Oscar Rojas Got to my repository