



A.3.2 Learning activity

(Touch sensor circuit through a NodeMCU ESP32)



Desarrollo

1. Use the following list of materials to prepare the activity

Quantity	Description
1	Red led diode
1	330 ohms resistance
1	5V power supply
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M / M
1	Aluminum sheet

2. Once the previous circuit has been assembled, write a program that allows the circuit to meet the following conditions:

[Link to the code](#)

```
// Primero establecemos los pines que vamos a utilizar
//gpio
int gpio_touch = 4;
int gpio_led = 17;

//lectura y estado
int reading;
int state;
int threshold = 20;

bool flag_touched = false;

//Establecemos las propiedades de señal pwm
int frequency = 5000;
int channel = 0;
int resolution = 8;
```

```

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    state = 0;
    //pwm: setup
    ledcSetup(channel, frequency, resolution); // configurar funcionalidades LED PWM
    pinMode(gpio_led, OUTPUT); // sirve para configurar el modo de trabajo de un
    pin pudiendo ser INPUT (entrada) u OUTPUT (salida).
}

```

- The system should be able to turn on and off a LED by pressing the touch sensor.

```

reading = touchRead(gpio_touch); // leemos el valor del sensor touch y lo
asignamos a la variable reading
Serial.println(reading); // imprime el dato de reading
flag_touched = reading < threshold;

if(state == 0) {
    if(flag_touched) {
        state = 1;
        ledcAttachPin(gpio_led, channel);
        for(int duty = 0; duty <= 255; duty++) {
            Serial.println(duty);
            ledcWrite(channel, duty);
            delay(15);
        }
    }
} else {
    if(flag_touched) {
        state = 0;
        ledcDetachPin(gpio_led);
        for(int i = 0; i < 3; i++) { //blinking
            digitalWrite(gpio_led, LOW);
            delay(1000);
            digitalWrite(gpio_led, HIGH);
            delay(1000);
        }
        digitalWrite(gpio_led, LOW); //turn off
    }
}

```

- The system must have the feature that if the LED is ON, when the touch sensor is touched, it will initiate an intermittent sequence of 3 seconds (that is, it turns off for 1 second, turns on for one second and finally turns off).

```

if(flag_touched) {
    state = 0;
    ledcDetachPin(gpio_led);
    for(int i = 0; i < 3; i++) { //blinking

```

```
    digitalWrite(gpio_led, LOW);
    delay(1000);
    digitalWrite(gpio_led, HIGH);
    delay(1000);
}
digitalWrite(gpio_led, LOW); //turn off
}
```

- The system must have the feature that if the LED is OFF, when touching the touch sensor, it will light up little by little until it reaches its maximum illumination level.

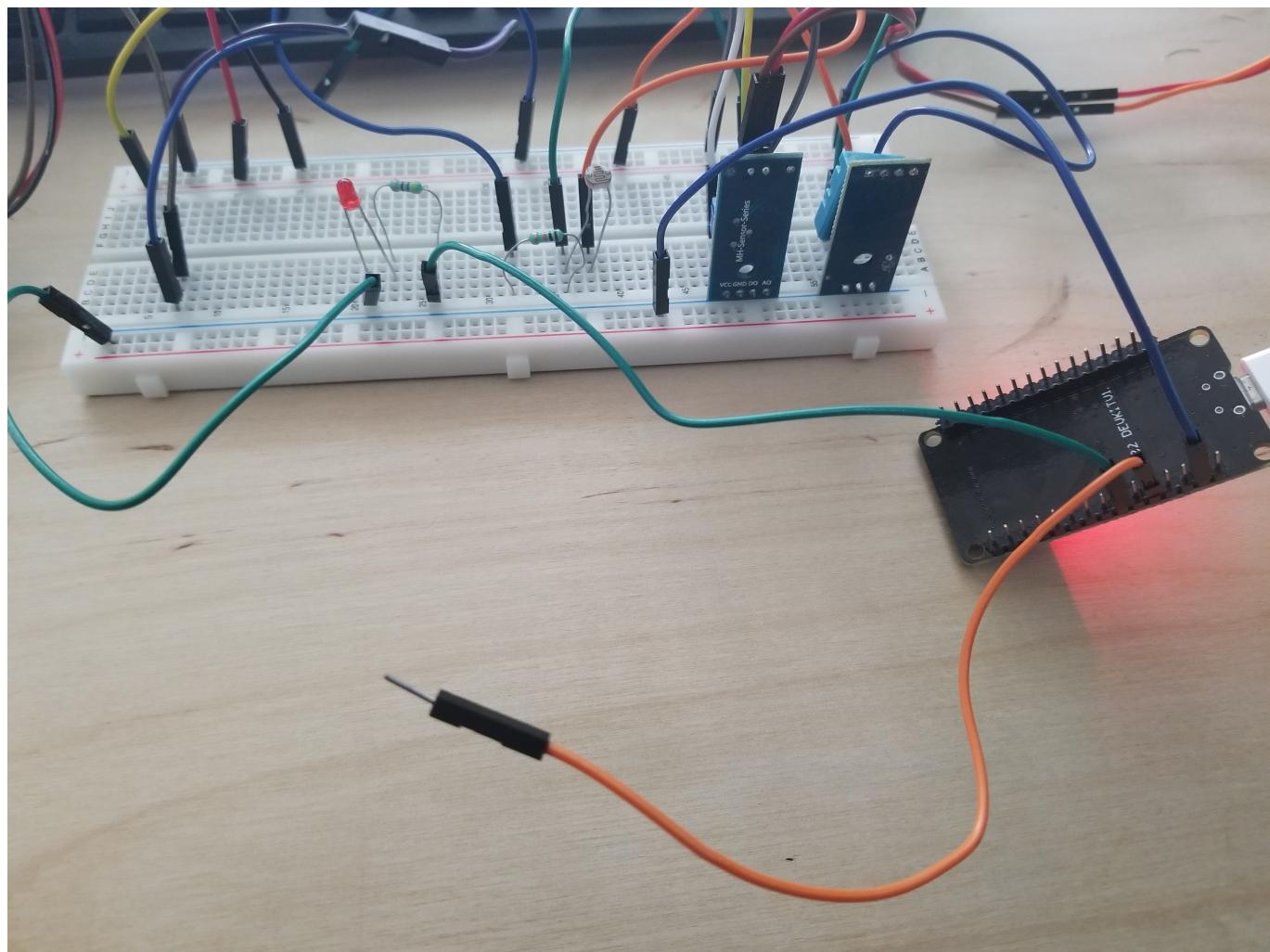
```
if(state == 0) {
    if(flag_touched) {
        state = 1;
        ledcAttachPin(gpio_led, channel);
        for(int duty = 0; duty <= 255; duty++) {
            Serial.println(duty);
            ledcWrite(channel, duty);
            delay(15);
        }
    }
}
```

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

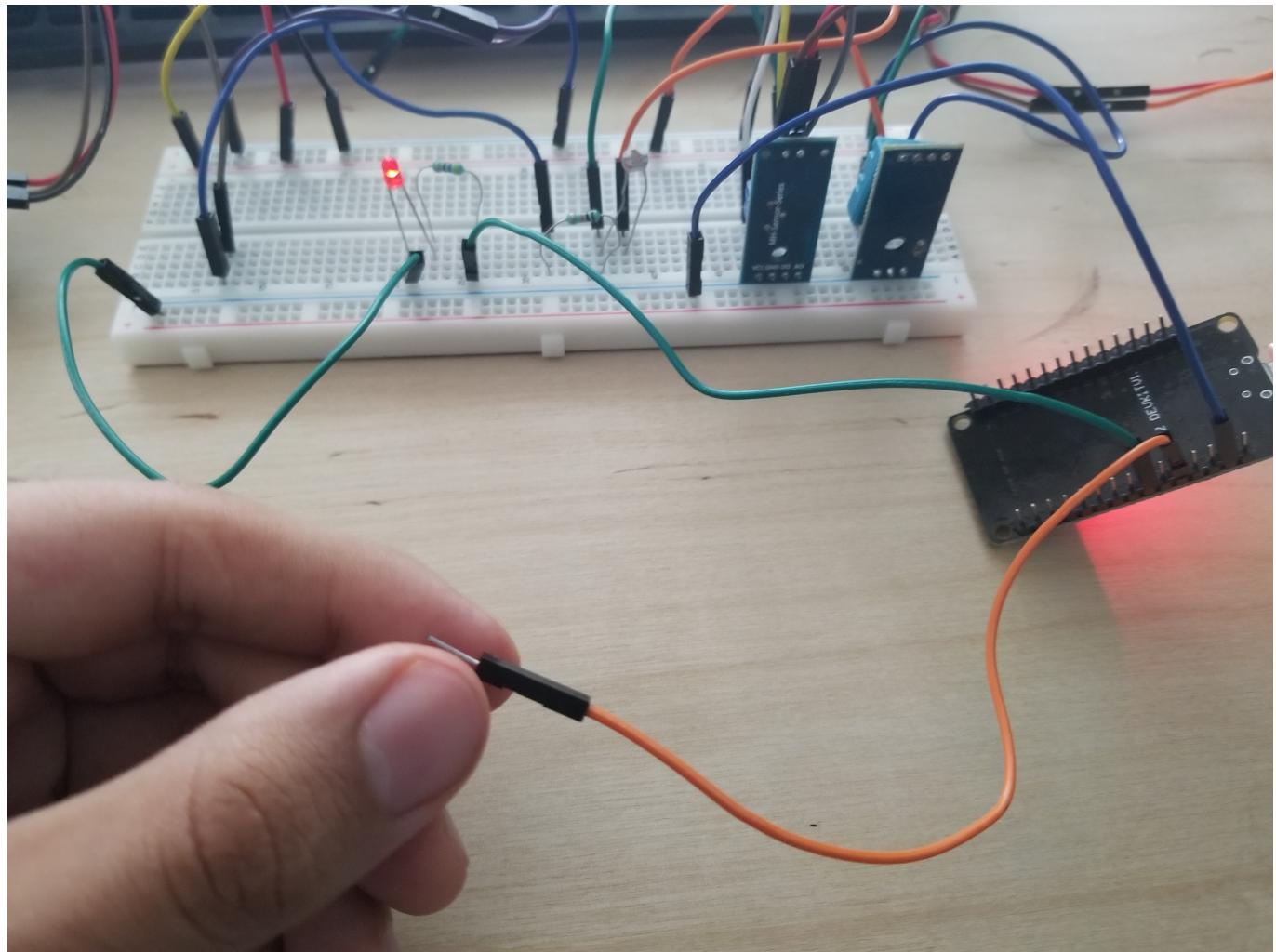
This is the circuit before touching the sensor pin.

NOTE: ONLY THE LED AND RESISTANCE CONNECTED ARE RELEVANT FOR THIS, THE REST IS FROM ANOTHER PROJECT AND WOULD BE TIME WASTING TO DISCONNECT EVERYTHING EVERYTIME I HAVE

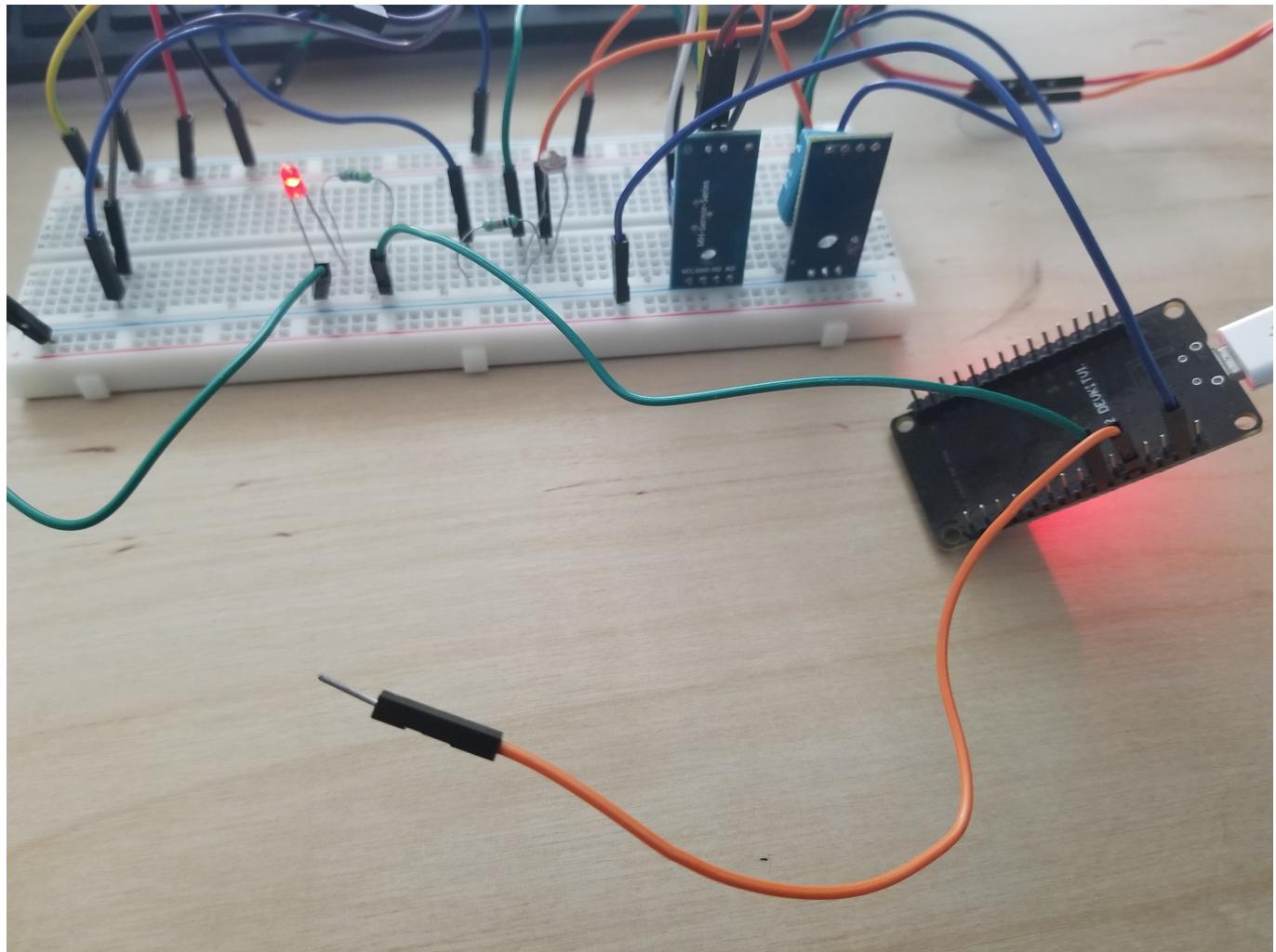
TO TEST THE CIRCUIT, MY APOLOGIES.



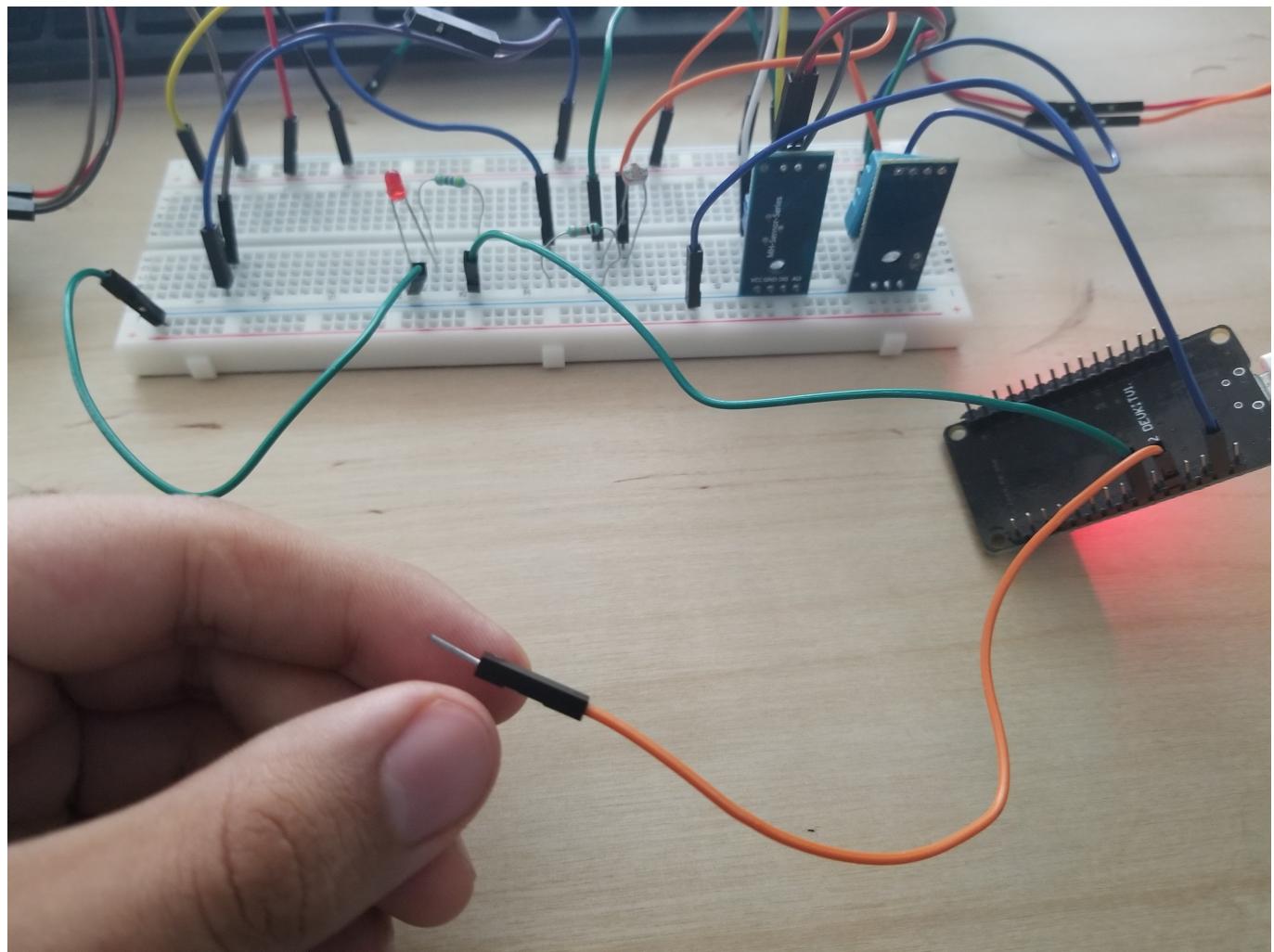
This is the LED after the first touching, the light is getting more intense, but in this picture doesnt show highest level of light yet.



This IS the highest level of light, once the LED reaches this points it will stay ON until another touch interaction in registered.



The moment the touch sensor is touched the LED starts blinking (it blinks 3 times, for 3 seconds) and then turns off, going back to the first state.



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

5. Include the individual conclusions and results observed during the development of the activity.

Edgar Regalado

This one was tricky, at first i couldn't figure out how the PWM worked, and once i read the tutorials given, i had a bug where once the pwm was set to the maximum at 255 i just couldn't turn the LED OFF to make the blinking effect. Turns out you need to detach the LED pin from the PWM channels so it will stop receiving its values, other way it would keep reading a HIGH (255) value. Using PWM to 'mimic' analog values using digital signals seems very useful in a lot of things. Also the touch sensor is kinda faulty, sometimes it would turn on with no touching at all.

Erick Garcia

As conclusion for this practice, I can say that the part that is interesting for me was the pin touch because I had the impression that it was more like a sensor, to know it can be used with the elbow makes think how does it work the touch sensor of my phone, about the code, it looks very similar to what we have been doing but also about how make it turn on slowly and do the intermittence it was kinda tricky.

Jose Gerardo

Honestly, it was a practice in which I had to work very little on it, it was difficult for me since the circuit had to be physical, therefore I did not understand much about it, I had to investigate about it separately.

Rubrica

Criteria	Description	Score
Instructions	Is each of the points indicated in the Instructions section fulfilled?	10
Developing	Was each one of the points requested within the development of the activity answered?	60
Demonstration	Does the student introduce himself during the explanation of the functionality of the activity?	20
Conclusions	Is a personal opinion of the activity included by each of the team members?	10

[My Github](#)