

Z C3.8 Programación Microcontrolador NodeMCU ESP32

Arduino y sensor de tacto integrado al NodeMCU ESP32



Instrucciones

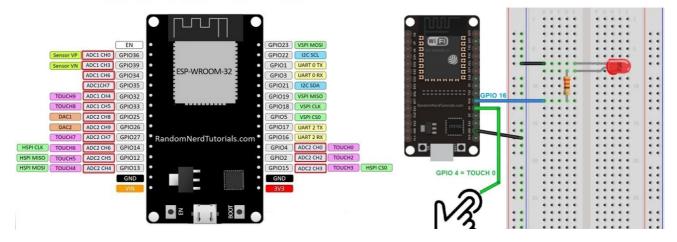
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo MarkDown con extension .md y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento single page, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo Enlace a mi GitHub
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura C3.8_NombreAlumno_Equipo.pdf, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma oficial aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- 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.

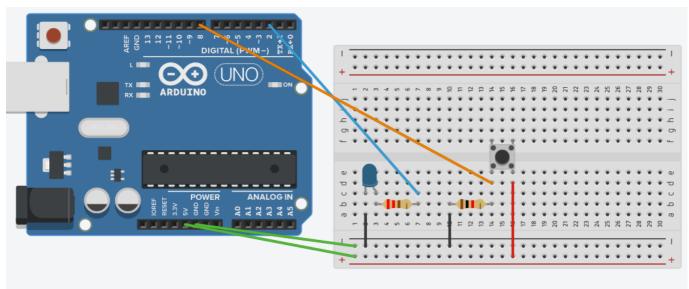
```
| readme.md
 blog
   C3.1_TituloActividad.md
 C3.2_TituloActividad.md
  C3.3 TituloActividad.md
 | C3.4 TituloActividad.md
   C3.5_TituloActividad.md
 | C3.6_TituloActividad.md
  C3.7 TituloActividad.md
 | C3.8_TituloActividad.md
 | img
 docs
 | A3.1 TituloActividad.md
 | A3.2_TituloActividad.md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



1. Basado en el siguiente circuito y ensamblarlo, utilizando alguno de los simulados propuesto, explicando el resultado que se desea obtener del mismo.

ESP32 DEVKIT V1 - DOIT





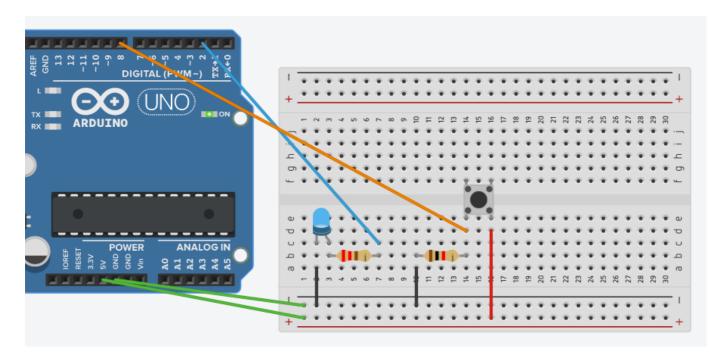
2. Analice y escriba el programa que se muestra a continuación.

```
// set pin numbers
const int touchPin = 4;
const int ledPin = 16;
// change with your threshold value
const int threshold = 20;
// variable for storing the touch pin value
int touchValue:
void setup(){
 Serial.begin(115200);
 delay(1000); // give me time to bring up serial monitor
 // initialize the LED pin as an output:
 pinMode (ledPin, OUTPUT);
void loop(){
  // read the state of the pushbutton value:
  touchValue = touchRead(touchPin);
  Serial.print(touchValue);
  // check if the touchValue is below the threshold
  // if it is, set ledPin to HIGH
  if(touchValue < threshold){</pre>
    // turn LED on
    digitalWrite(ledPin, HIGH);
    Serial.println(" - LED on");
  else{
   // turn LED off
    digitalWrite(ledPin, LOW);
    Serial.println(" - LED off");
  delay(500);
```

Fuente de consulta: Random Nerd Tutorials

```
int pinButton = 8; //pin del boton debido a la simulacion
   int LED = 2; //pin conectado al led
4
5
   void setup() {
6
     Serial.begin(115200); //inicia comunicacion serial
7
     delay(1000); //delay de 1 segundo
8
     pinMode(pinButton, INPUT); //se utiliza el boton como entrada
9
     pinMode(LED, OUTPUT); //se utiliza el led como salida
.0
.1
   }
.2
.3
   void loop() {
.4
     int stateButton = digitalRead(pinButton); //lee el estado del boton
.5
     if(stateButton == 1) { //si el boton es presionado
.6
        digitalWrite(LED, HIGH); //pone el led en high
.7
       Serial.println("Led encendido"); //escribe el estado del led
.8
     } else { //si el boton no es presionado
.9
        digitalWrite(LED, LOW); //pone el led en low
:0
       Serial.println("Led apagado"); //escribe el estado del led
1
2
     delay(500); //delay
13
   }
```

3. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.



- Led apagado
- Led apagado Led apagado
- Led encendido
- Led encendido
- Led encendido
- Led encendido

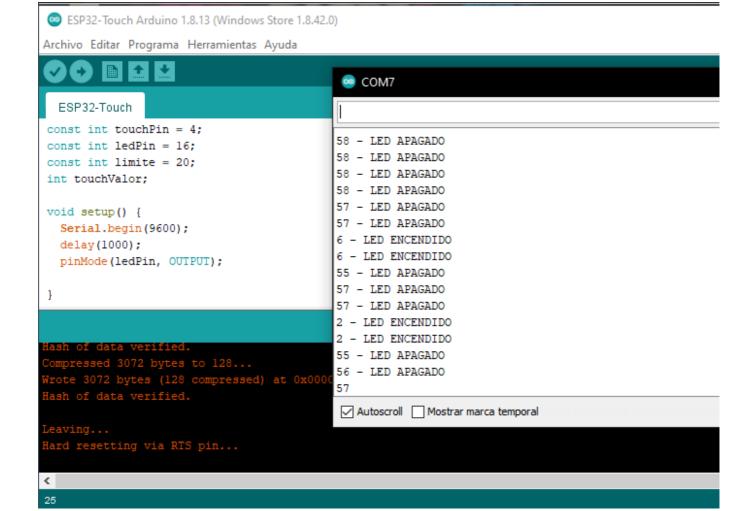
o ESP32-Touch Arduino 1.8.13 (Windows Store 1.8.42.0)

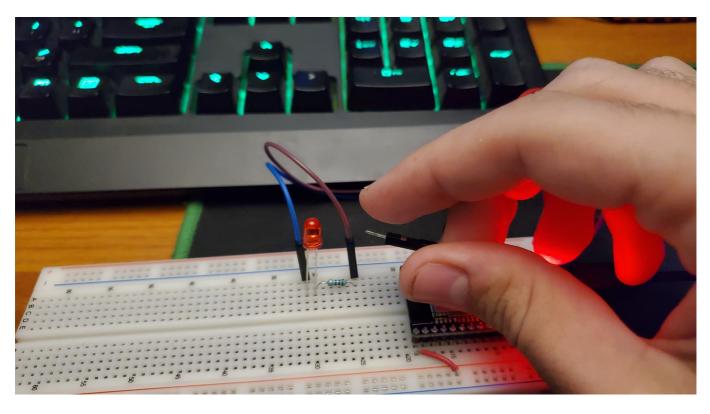
Archivo Editar Programa Herramientas Ayuda

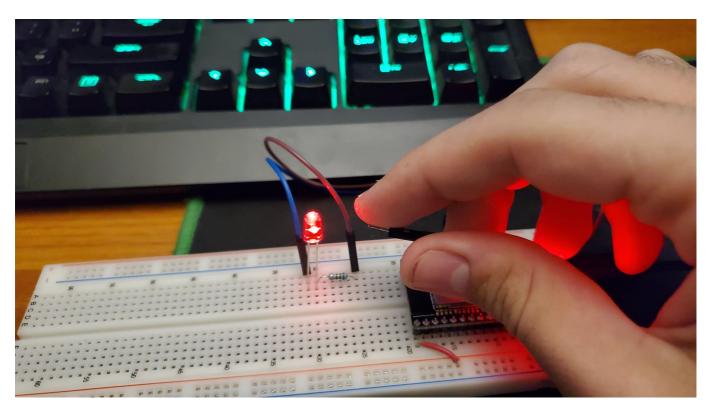


ESP32-Touch §

```
//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;
void setup() {
  Serial.begin(9600);
  delay(1000);
  //Se establece el pin donde esta conectado el led como
  //pin de salida.
 pinMode (ledPin, OUTPUT);
}
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);
  Serial.print(touchValor);
  //Si este valor medido es menor al limite establecido
  //enciende el LED e imprime en pantalla.
  if (touchValor < limite) {
   digitalWrite(ledPin, HIGH);
   Serial.println(" - LED ENCENDIDO ");
  }
  //Si este valor medido es mayor al limite establecido
  //apaga el LED e imprime en pantalla.
  else{
   digitalWrite(ledPin, LOW);
   Serial.println(" - LED APAGADO");
  delay(500);
1
```









Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

Mi repositorio de Github