Centro de Industria
y Construcción
Regional Tolima

MANUAL DE FUNCIONAMIENTO DEL PROTOTIPO KIT LOT

Sandra Milena Chilito Barón
schilitob@sena.edu.co
Autor principal

Hilson Javier Tibaduiza Álvarez
hjtibaduizaa@sena.edu.co
Co Autoripal

Contenido

Introducción 12

1. Manual de Funcionamiento de la Tarjeta Central 13

1.1 Entradas 14

1.2 Salidas 14

1.3 Tarjeta NodeMCU. 15

1.4 Recomendaciones 16

1.5 Iniciando a Programar la Tarjeta Central 16

1.5.1 Especificaciones Iniciales 16

1.5.2 Pasos para Descargar la Librería de NodemCu 8266 en Arduino 18

1.5.3 Código Placa Central 20

1.5.4 Procedimientos de Seguridad – Kit IoT 24

2. Sensores Diseñados para el Kit IoT 25

2.1 Sensor de Efecto Hall 25

2.2 Tipos de Salidas y Visualización de Datos 28

2.2.1 Salida Digital 28

2.2.2 Salida Análoga 28

2.3 Conexión a la Tarjeta de Desarrollo 28

2.3.1 Consideraciones Importantes 33

2.3.2 Resolución de Problemas 33

2.3.3 Recomendaciones de Seguridad 34

2.3.4 Mantenimiento 34

2.4 Sensor de Proximidad Infrarrojo 34

2.4.1 Funcionamiento 35

2.4.2 Preparación Inicial 35

2.4.3 Activación 36

2.5 Conexión a la Tarjeta de Desarrollo 38

2.5.1 Tener en Cuenta 43

2.6 Sensor de Temperatura 44
2.6.1 Funcionamiento 44
2.6.2 Preparación Inicial 45
2.6.3 Activación 45
2.6.4 Conexión a la Tarjeta de Desarrollo 46
2.6.5 Tener en Cuenta 50
2.7 Sensor de Sonido 51
2.7.1 Características Principales 52
2.7.2 Tener en Cuenta 52
2.7.3 Ajuste del Potenciómetro 53
2.7.4 Activación 53
2.7.5 Funcionamiento del Sensor 59
Referencias 62

Listado de Figuras

Figura 1. Tarjeta Central 11
Figura 2. Entradas tarjeta Central 12
Figura 3. Salidas tarjeta central 12
Figura 4. Tarjeta embebida Nodemcu 13
Figura 5. Librería Display LCD con I2C, ... 15
Figura 6. Librería Sonda sensor de temperatura 15
Figura 7. Descargar la librería de NodemCu 8266 en Arduino 16
Figura 8. Gestor de urls 17
Figura 9. Gestor de tarjetas 17
Figura 10. Gestor de tarjetas. insertar referencia de la placa. 18
Figura 11. Sensor de Efecto Hall 24
Figura 12. Vin 5V 25
Figura 13. Inicialización 25
Figura 14. Visualización de datos 26
Figura 15. Leds mostrarán el valor en formato binario 26
Figura 16. Conexión a la Tarjeta de Desarrollo 27
Figura 17. Conectar el cable de datos a uno de los puertos del pc 28
Figura 18. Conectar el cable de datos a la tarjeta de desarrollo 28
Figura 19. Abrir la aplicación “ARDUINO IDE” 28
Figura 20. Seleccionar la placa/tarjeta y puerto en el IDE 30
Figura 21. Seleccionar otra placa y puerto 31
Figura 22. Transferir el código a la tarjeta dando clic en “cargar” 31
Figura 23. Monitor serial y visualizar datos 32
Figura 24. Sensor de Proximidad Infrarrojo 33
Figura 25. Conectar la tarjeta a la fuente de 5V DC 34
Figura 26. Reset 35
Figura 27. Digital out 35
Figura 28. Salidas van conectadas a una tarjeta de desarrollo 36

Figura 29. Conexión a la Tarjeta de Desarrollo 37
Figura 30. Conectar el cable de datos a uno de los puertos del pc 37
Figura 31. Conectar el cable de datos a la tarjeta de desarrollo 37
Figura 32. Abrir la aplicación “ARDUINO IDE” 38
Figura 33. Selecciona la placa/tarjeta y puerto en el IDE 39
Figura 34. Selecciona la placa/tarjeta y puerto en el IDE 40
Figura 35. Transferencia del código a la tarjeta dando clic en “cargar” 40
Figura 36. Se abre el monitor serial y se visualizan los datos 41
Figura 37. Sensor de Temperatura 42
Figura 38. Conectar la tarjeta a la fuente de 5V DC (+/-) 43
Figura 39. Activar el conversor es necesario dar 1 un pulso en el reset luego de energizar el circuito 44
Figura 40. Se conectan los jumpers a cada una de las salidas digitales de la placa 44
Figura 41. Las salidas van conectadas a una tarjeta de desarrollo 45
Figura 42. Se realiza la conexión del cable de datos a uno de los puertos del pc 46
Figura 43. Se realiza la conexión al cable de datos a la tarjeta de desarrollo 46
Figura 44. Abrir la aplicación “ARDUINO IDE” 46
Figura 45. Se realiza la selección de la placa/tarjeta y puerto en el IDE
(en este caso Arduino uno, COM4) 48
Figura 46. Se realiza la selección de la placa/tarjeta y puerto en el IDE
(en este caso Arduino uno, COM4) 48
Figura 47. Se transfiere el código a la tarjeta dando clic en “cargar” 48
Figura 48. Se debe abrir el monitor serial y visualizar datos 49
Figura 49. Sensor de Sonido 50
Figura 50. Verificación del Led 51
Figura 51. Ajuste del Potenciómetro 52
Figura 52. Conectar el cable datos a un computado 53
Figura 53. Conectar los pines digitales desde el D0 hasta el D7 y de ultimo RX y TX 53
Figura 54. Abrir el entorno de programación IDE Arduino 54
Figura 55. Abrir el código del sensor ubicado en: carpeta Códigos Arduino- sensor de sonido-código sensor de sonido) y pegarlo en IDE Arduino 54

Figura 56. Selección puerto 57
Figura 57. Transferir el código a la tarjeta dando clic en “cargar” 58
Figura 58. Funcionamiento del Sensor 58
Figura 59. El led indicador solo debe encender al detectar sonido 59

Listado de Tablas

Tabla 15. Conexión a la Tarjeta de Desarrollo..... 26

Tabla 2. Conexión a la Tarjeta de Desarrollo..... 35

Tabla 3. Las salidas van conectadas a una tarjeta de desarrollo..... 43

Introducción

El presente Manual de Funcionamiento del Kit IoT tiene como propósito orientar de manera clara, técnica y sistemática el uso, configuración y aplicación del sistema de monitoreo y control basado en la tarjeta central de recepción de datos NodeMCU ESP8266. Este Kit ha sido diseñado como una herramienta didáctica y de experimentación tecnológica que integra componentes electrónicos, sensores y módulos de comunicación, permitiendo la adquisición, procesamiento y transmisión de datos hacia la nube dentro de un entorno de Internet de las Cosas (IoT).

El documento describe las características funcionales de cada componente, los procedimientos de conexión y configuración, así como las etapas de programación y validación del sistema mediante el entorno IDE de Arduino. Además, establece los protocolos de seguridad eléctrica y operativa, garantizando una manipulación responsable y segura de los dispositivos.

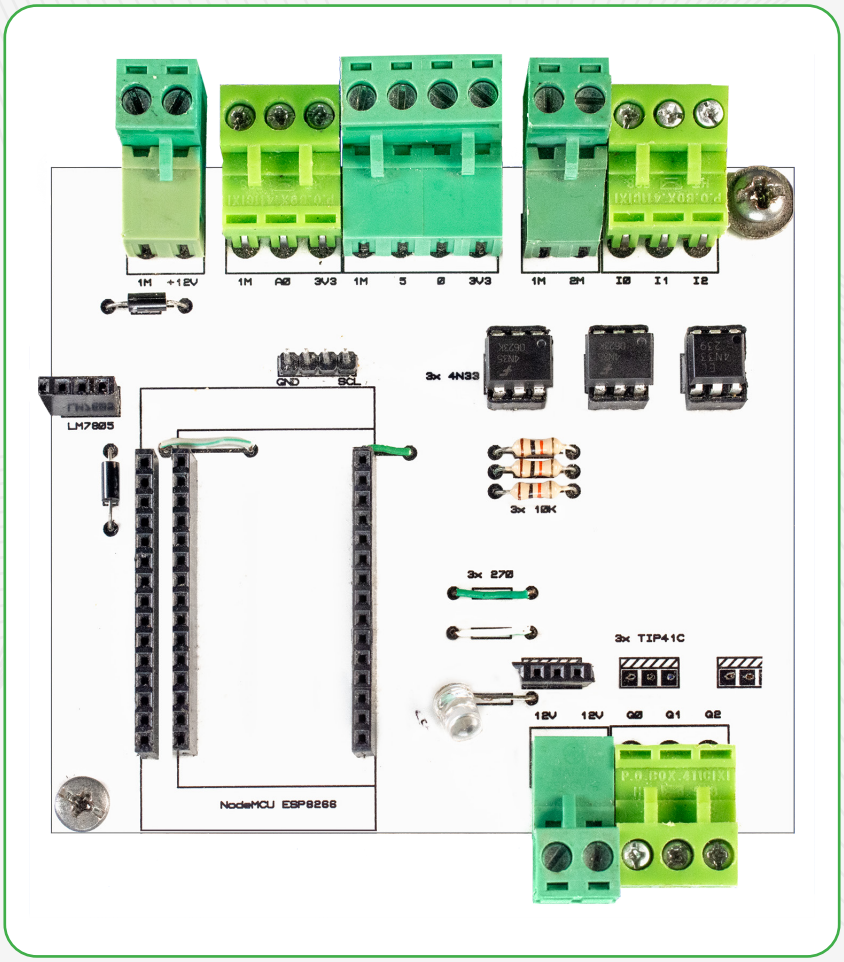
De igual forma, se detallan los principales sensores integrados en el Kit IoT, como los de temperatura, sonido, efecto Hall e infrarrojo, explicando su principio de funcionamiento, aplicación práctica y forma de integración con la tarjeta central. El sistema propuesto contribuye al fortalecimiento de competencias en automatización, electrónica e instrumentación, promoviendo el aprendizaje activo y el desarrollo de soluciones tecnológicas en entornos educativos y experimentales.

De esta manera, este manual constituye una guía integral de referencia técnica y pedagógica, orientada a facilitar el entendimiento, la implementación y el aprovechamiento del Kit IoT como plataforma base para el desarrollo de proyectos académicos, de investigación y de innovación tecnológica.

1. Manual de Funcionamiento de la Tarjeta Central

La tarjeta central de recepción de datos es un dispositivo electrónico diseñado para automatizar procesos eléctricos o electromecánicos y actuar como el “cerebro” del sistema. Esta permite recibir datos externos a través de sensores y enviarlos para su visualización en la nube, facilitando el monitoreo y control remoto de las variables del entorno.

Figura 1. Tarjeta Central



Fuente: Elaboración propia

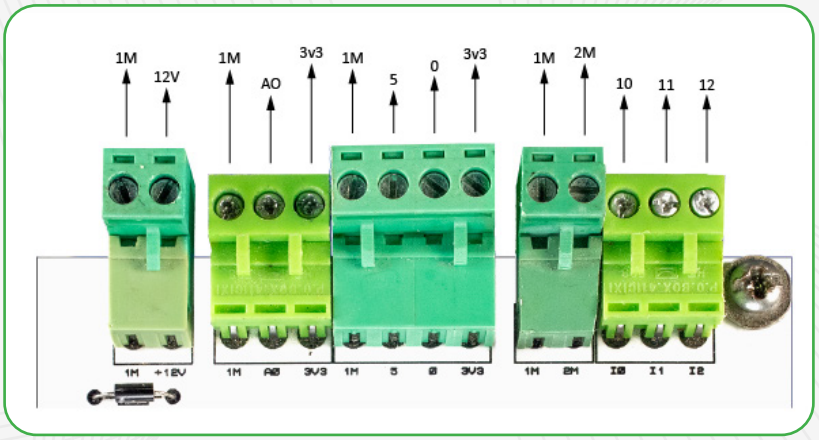
La tarjeta cuenta con múltiples entradas y salidas que permiten la conexión de diversos sensores y actuadores, e integra una tarjeta de desarrollo ESPNodecum, basada en el microcontrolador ESP8266, que gestiona la comunicación y el procesamiento de los datos, como se observa en la figura 1.

La tarjeta central utilizada en el prototipo Kit IoT, consta de las siguientes entradas (figura 2) y salidas (figura 3):

1.1 Entradas

- 1M: GND
- 2M: GND
- I0: Entrada Octocoplada
- I1: Entrada Octocoplada
- I2: Entrada Octocoplada
- A0: Entrada Analógica (A0)
- 12V: Fuente de 12V
- 5: Entrada Digital (D2)
- **0:** Entrada Digital (D4)

Figura 2. Entradas tarjeta Central

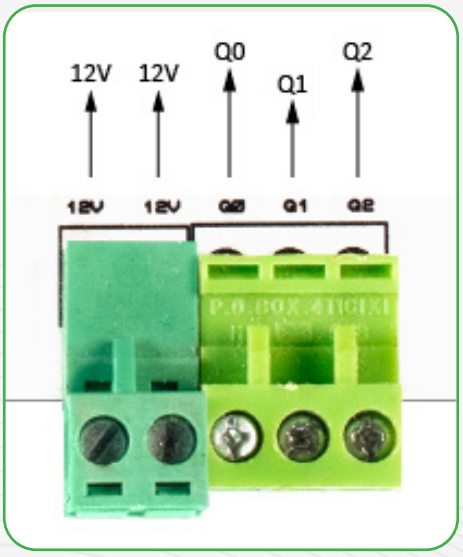


Fuente: Elaboración propia

1.2 Salidas

- **Q0:** Salida
- **Q1**: Salida
- **Q2**: Salida

Figura 3. Salidas tarjeta central

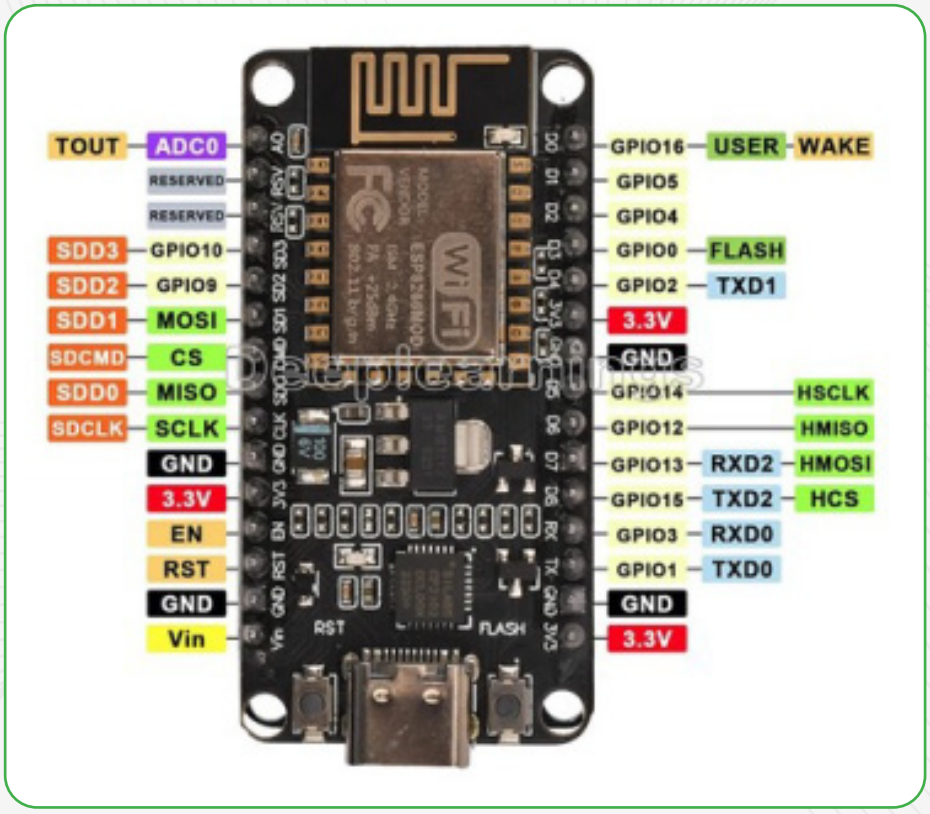


Fuente: Elaboración propia

1.3 Tarjeta NodeMCU.

La tarjeta central de recepción de datos utilizada en el Kit IoT incorpora una tarjeta embebida NodeMCU ESP8266, como se observa en la Figura 4. Esta placa de desarrollo de hardware y software de código abierto permite la programación directa del microcontrolador ESP8266 mediante el IDE de Arduino, facilitando la integración de sensores, la transmisión de datos a la nube y el control de dispositivos de manera eficiente y accesible.

Figura 4. Tarjeta embebida Nodemcu



Fuente: Ebay (2025)

1.4 Recomendaciones

Para dar inicio al uso del Kit IoT, es necesario realizar la configuración inicial de la tarjeta central de recepción de datos. Para ello, siga las siguientes recomendaciones.

De esta manera, antes de iniciar la configuración, es fundamental verificar el estado físico y funcional de la tarjeta central de recepción de datos. A continuación, se presentan las recomendaciones iniciales:

- Revise que la tarjeta cuente con todos sus componentes instalados, especialmente el regulador

de voltaje y los tres transistores TIP41C.

- Verifique el correcto funcionamiento del regulador de voltaje, ya que sin este componente la tarjeta ESP8266 podría dañarse.
- Mida el voltaje de salida del regulador, el cual debe ser de 5 V. Si el valor es diferente, reemplace el regulador antes de continuar.
- Conecte la tarjeta NodeMCU ESP8266 con la entrada USB orientada hacia abajo, enlazándola con la entrada USB del computador.
- La alimentación proveniente del puerto USB del computador debe ser suficiente para energizar la tarjeta y permitir la carga del programa.
- Una vez verificada la conexión, suba el código correspondiente desde el IDE de Arduino para completar la configuración inicial del Kit IoT.

1.5 Iniciando a Programar la Tarjeta Central

Para dar inicio a la programación en Arduino de la tarjeta central del Kit IoT, debe especificar los siguientes aspectos en el entorno del IDE de Arduino:

1.5.1 Especificaciones Iniciales

- Entradas:

  D0 es igual a I0

  D5 es igual a I1

  D6 es igual a I2
- Entrada solo para sensor: D3 que es igual a bornera 0

  Entrada análoga: 0
- Salida:

  D4 es igual a Q0

  D8 es igual a Q1

  D7 es igual a Q2
- Librerías de programación:
- Display LCD con I2C, como se observa en la figura 5

Figura 5. **Librería Display LCD con I2C,**



Fuente: Elaboración propia

- En el programa de arduino se debe incluir como:

```
#include<LiquidCrystal_PCF857.h>
```

- Sonda sensor de temperatura, como se observa en la figura 6

Figura 6. **Librería Sonda sensor de temperatura**



Fuente: Elaboración propia

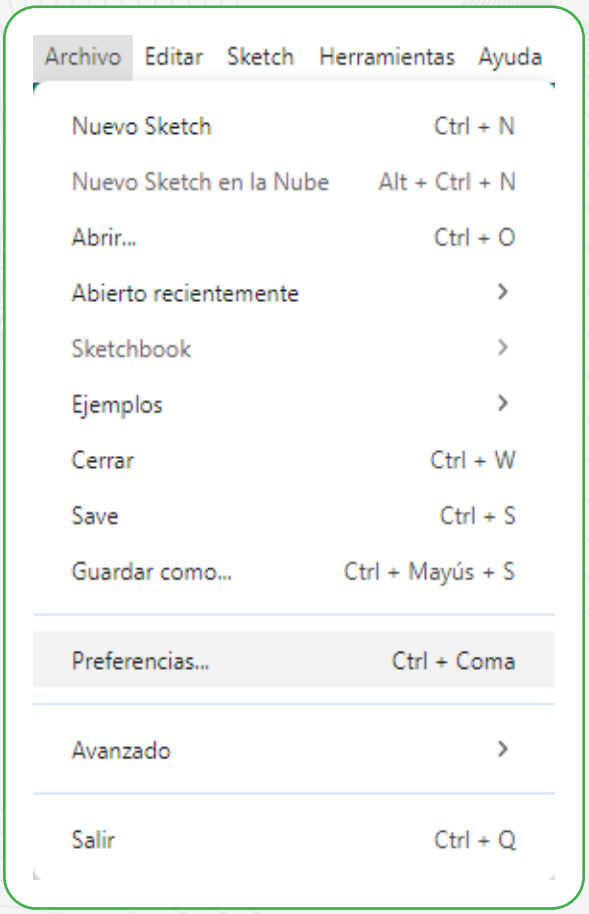
- En el programa de Arduino se incluye como:

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
```

1.5.2 Pasos para Descargar la Librería de NodemCu 8266 en Arduino

- En el IDE a trabajar, en el menú archivo, en preferencias como se muestra en la figura 7.

Figura 7. Descargar la librería de NodemCu 8266 en Arduino

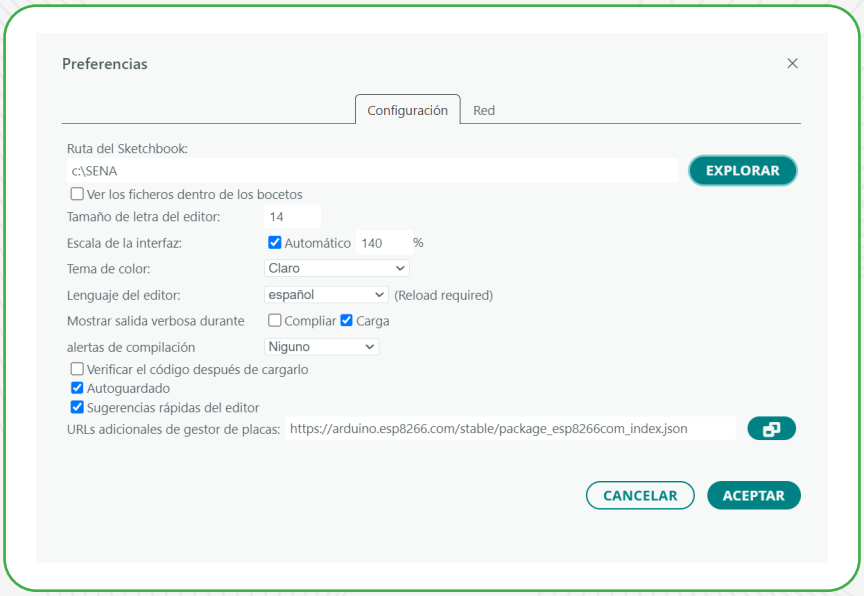


Fuente: Elaboración propia

- Agregar el siguiente url que indica el gestor de urls y dar aceptar.

URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json como se observa en la figura 8.

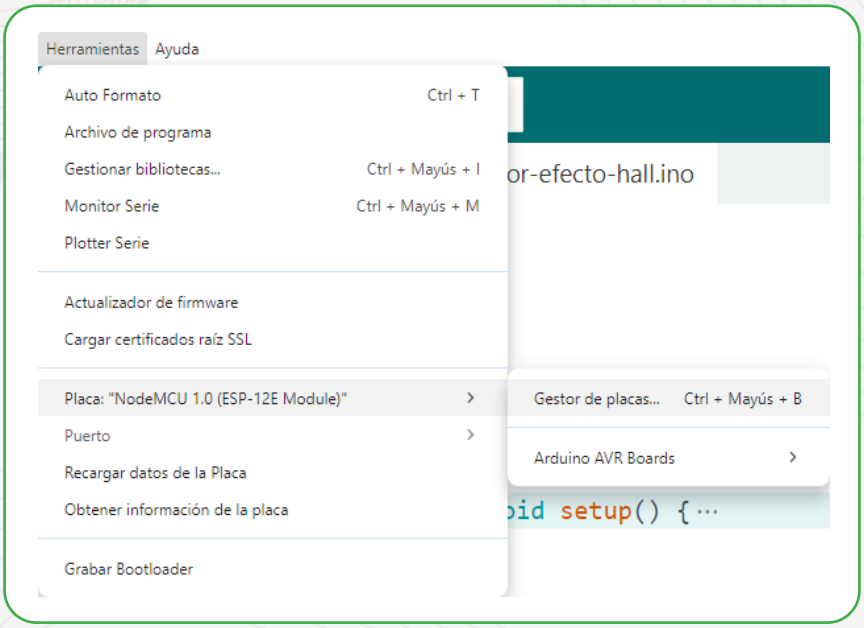
Figura 8. Gestor de urls



Fuente: Elaboración propia

- Ingresar a herramientas - placas - gestor de tarjetas, como se observa en la figura 9.

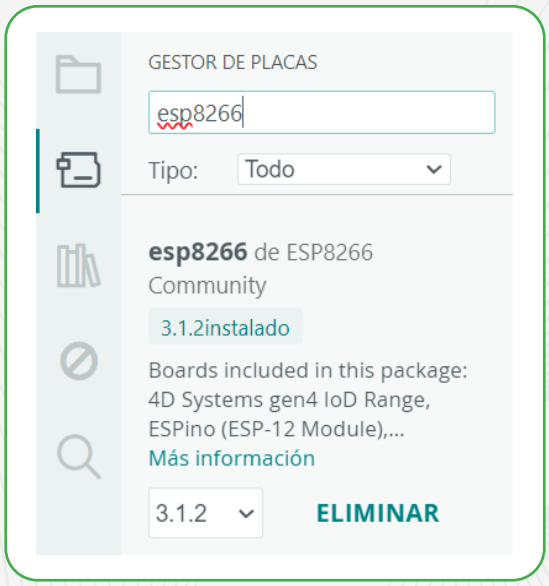
Figura 9. Gestor de tarjetas



Fuente: Elaboración propia

- Por último, en la pestaña se debe buscar la tarjeta como “esp8266” se instala y se pone en placa como se observa en la figura 10.

Figura 10. Gestor de tarjetas. insertar referencia de la placa.



Fuente: Elaboración propia

1.5.3 Código Placa Central

A continuación, se presenta el código necesario para el funcionamiento de la tarjeta central de recepción de datos, el cual debe cargarse mediante el IDE de Arduino, previa instalación de las librerías requeridas para su correcta ejecución.

- **Código:**

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_PCF8574.h>
LiquidCrystal_PCF8574 lcd(0x27); // Crea el objeto LCD con dirección I2C
int show = -1;
// 2 custom characters
byte dotOff[] = { 0b00000, 0b01110, 0b10001, 0b10001,
0b10001, 0b01110, 0b00000, 0b00000}; // Carácter personalizado 1 (apagado)
byte dotOn[] = { 0b00000, 0b01110, 0b11111, 0b11111,
0b11111, 0b01110, 0b00000, 0b00000}; // Carácter personalizado 2 (encendido)
void setup() {
```

```
int error;
Serial.begin(115200);
Serial.println("LCD...");
// wait on Serial to be available on Leonardo
while (!Serial)  ;
Serial.println("Probing for PCF8574 on address 0x27...");
// See http://playground.arduino.cc/Main/I2cScanner how to test for a I2C device.
Wire.begin();
Wire.beginTransmission(0x27);   // Verifica si el LCD está conectado
error = Wire.endTransmission();
Serial.print("Error: ");
Serial.print(error);
if (error == 0) {
Serial.println(": LCD found.");
show = 0;
lcd.begin(16, 2);  // // Inicializa LCD de 16 columnas y 2 filas
lcd.createChar(1, dotOff);  // Crea carácter personalizado 1
lcd.createChar(2, dotOn);   // Crea carácter personalizado 2
} else {
Serial.println(": LCD not found.");
}  // if
}  // setup()
void loop() {
 if (show == 0) {
lcd.setBacklight(255);  // Enciende la luz de fondo
lcd.home();
lcd.clear();
lcd.print("Hello LCD");  // Imprime texto
```

```
delay(1000);
lcd.setBacklight(0);
delay(400);
lcd.setBacklight(255);
} else if (show == 1) {
lcd.clear();  // Limpia la pantalla
lcd.print("Cursor On");
lcd.cursor();
} else if (show == 2) {
lcd.clear();  // Limpia la pantalla
lcd.print("Cursor Blink");
lcd.blink();
} else if (show == 3) {
lcd.clear();  // Limpia la pantalla
lcd.print("Cursor OFF");
lcd.noBlink();
lcd.noCursor();
} else if (show == 4) {
lcd.clear();  // Limpia la pantalla
lcd.print("Display Off");
lcd.noDisplay();
} else if (show == 5) {
lcd.clear(); // Limpia la pantalla
lcd.print("Display On");
lcd.display();
} else if (show == 7) {
lcd.clear();  // Limpia la pantalla
lcd.setCursor(0, 0);
```

```
lcd.print("* first line.");
lcd.setCursor(0, 1);
lcd.print("* second line.");
} else if (show == 8) {
lcd.scrollDisplayLeft();  // Desplaza texto a la izquierda
} else if (show == 9) {
lcd.scrollDisplayLeft();
} else if (show == 10) {
lcd.scrollDisplayLeft();
} else if (show == 11) {
lcd.scrollDisplayRight();
} else if (show == 12) {
lcd.clear();  // Limpia la pantalla
lcd.print("write-");
} else if (show == 13) {
lcd.clear();
lcd.print("custom 1:<\01>");
lcd.setCursor(0, 1);
lcd.print("custom 2:<\02>");
} else {
lcd.print(show - 13);
}  // if
delay(1400);
show = (show + 1) % 16; // Cambia el modo mostrado cada 1.4 segundos
}  // loop()
```

1.5.4 Procedimientos de Seguridad – Kit IoT

La operación segura del Kit IoT requiere seguir normas estrictas que garanticen la protección del personal y la integridad de los equipos electrónicos, especialmente de la tarjeta central de recepción de datos basada en NodeMCU ESP8266.

A continuación, se presentan los procedimientos básicos de seguridad que deben aplicarse durante la instalación, configuración y operación del sistema:

1. Desconexión de Energía

Antes de realizar cualquier conexión, mantenimiento o ajuste en el kit:

- Asegúrese de que la fuente de alimentación esté completamente desconectada.
- Espere algunos segundos después de la desconexión para permitir la descarga total de los condensadores y componentes con carga.
- Nunca conecte o desconecte sensores o actuadores con el sistema energizado.

2. Uso de Equipos de Protección Personal (EPP)

Durante las actividades de instalación y prueba del sistema:

- Utilice guantes aislantes al manipular la tarjeta y los componentes de potencia.
- Use gafas de seguridad para prevenir lesiones por chispas o fragmentos metálicos.
- Emplee calzado dieléctrico para reducir el riesgo de descargas eléctricas.

3. Cableado Seguro

Para garantizar una correcta conexión eléctrica:

- Verifique que los cables estén firmemente sujetos en sus terminales y sin peladuras.
- Utilice conductores del calibre adecuado, especialmente en las salidas conectadas a los transistores TIP41C o a relés.
- Respete el código de colores normalizado (por ejemplo, rojo para V+, negro para GND y azul o amarillo para señales).
- Evite cruzar cables de señal con líneas de potencia para prevenir interferencias.

4. Validación del Programa

Antes de activar el sistema con cargas reales:

- Compruebe el correcto funcionamiento del código cargado en la NodeMCU ESP8266 mediante el IDE de Arduino.
- Realice las pruebas iniciales con cargas simuladas (por ejemplo, LEDs o resistencias) antes de

conectar motores, bombas o actuadores.

- Verifique que las entradas y salidas digitales respondan correctamente según la lógica programada.

5. Respaldo del Programa

- Conserve siempre una copia del programa original antes de realizar modificaciones o actualizaciones.
- Guarde una versión de respaldo en un medio seguro (memoria USB o nube) identificando fecha, versión y descripción del cambio.

6. Documentación Técnica

- Etiquete todas las conexiones de sensores y actuadores indicando su función (por ejemplo: "Sensor de temperatura – D2", "Salida de bomba – D5").
- Mantenga actualizados los diagramas eléctricos y el mapa de pines de la tarjeta central.
- Incluya en la documentación el valor del voltaje de operación (5 V) y los componentes críticos como el regulador de voltaje y los TIP41C, ya que una conexión incorrecta puede dañar la ESP8266.

2. Sensores Diseñados para el Kit IoT

A continuación, se presentan los cuatro sensores que integran el Kit IoT, cada uno diseñado para captar variables del entorno y enviar la información hacia la tarjeta central de recepción de datos (NodeMCU ESP8266), donde los datos son procesados y transmitidos a la nube para su visualización y análisis.

Estos sensores permiten medir diferentes magnitudes físicas, facilitando la automatización, monitoreo y control de los sistemas conectados al Kit IoT.

2.1 Sensor de Efecto Hall

Un sensor de efecto Hall es un dispositivo electrónico utilizado para detectar la presencia, intensidad o variación de un campo magnético. Su principio de funcionamiento se basa en el efecto Hall, fenómeno descubierto por (Hall, 1879), quien observó que, cuando una corriente eléctrica atraviesa un material conductor o semiconductor y se aplica un campo magnético perpendicular a dicha corriente, se genera una diferencia de potencial transversal denominada voltaje Hall.

Estos sensores son ampliamente empleados en aplicaciones industriales y electrónicas, tales como la medición de velocidad de rotación, la detección de posición angular o lineal, el control de moto-

res eléctricos y los sistemas de encendido automotriz (Sedra y Smith, 2016).

Existen versiones analógicas, que entregan una señal proporcional a la intensidad del campo magnético, y digitales, que generan una señal binaria (encendido/apagado) cuando el campo supera un umbral definido.

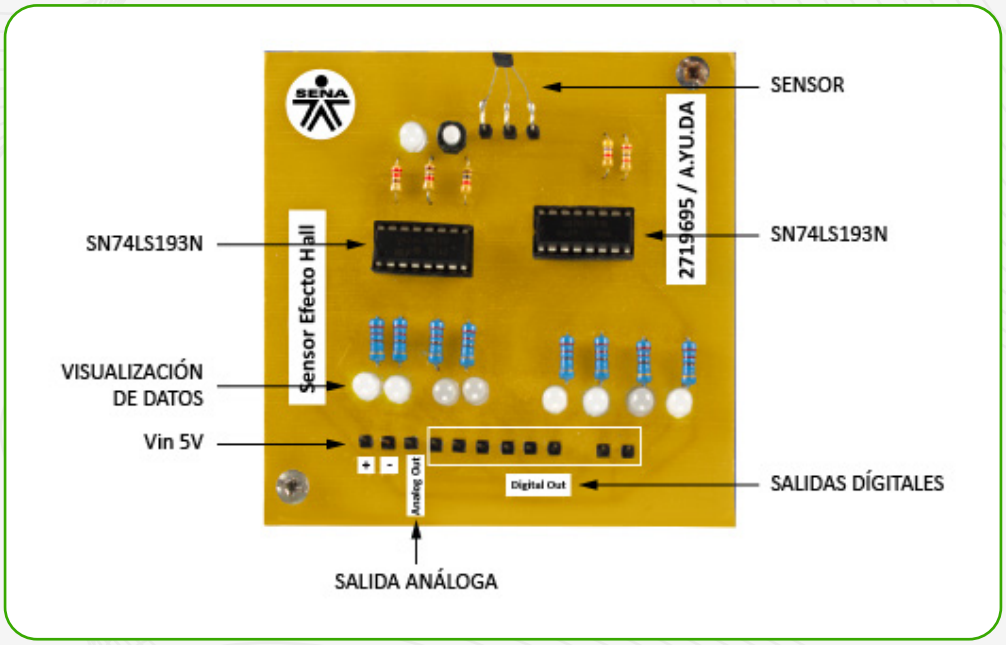
En el contexto del Kit IoT, el sensor de efecto Hall se utiliza para detectar el paso de objetos magnéticos o medir la velocidad de rotación de un eje, enviando la señal resultante a la tarjeta central de recepción de datos (NodeMCU ESP8266), donde se procesa y transmite a la nube para su visualización y análisis.

2.1.1 Funcionamiento

El sensor de efecto Hall, el cual se muestra en la figura 11, al detectar la presencia de un campo magnético, genera una señal o pulso eléctrico que es enviado al contador binario de 4 bits. Este contador interpreta cada pulso como un incremento en su secuencia binaria, encendiendo los LED conectados según la combinación correspondiente al número de detecciones realizadas.

De esta manera, el sistema permite visualizar en forma binaria la cantidad de veces que el sensor ha sido expuesto a un campo magnético, proporcionando una representación simple y efectiva del conteo de eventos detectados.

Figura 11. Sensor de Efecto Hall



Fuente: Elaboración propia

2.1.2 Preparación Inicial

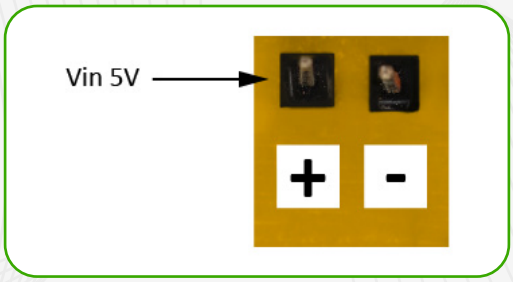
- Fuente de alimentación de 5 voltios DC
- Tarjeta sensor de efecto Hall
- Imán con polaridad correcta

2.1.3 Conexión y Puesta en Funcionamiento

1. Energización

- Conecte la placa al suministro de 5 voltios DC según las marcas indicadoras en la figura 12
- Asegúrese de que la conexión sea segura y estable

Figura 12. **Vin 5V**



Fuente: Elaboración propia

2. Inicialización

- Localice el botón de reset ubicado en la esquina superior izquierda de la placa, como se observa en la figura 13.
- Pulse el botón de reset para preparar un nuevo conteo
- Observará que un led se iluminará indicando que el sistema está listo

Figura 13. **Inicialización**



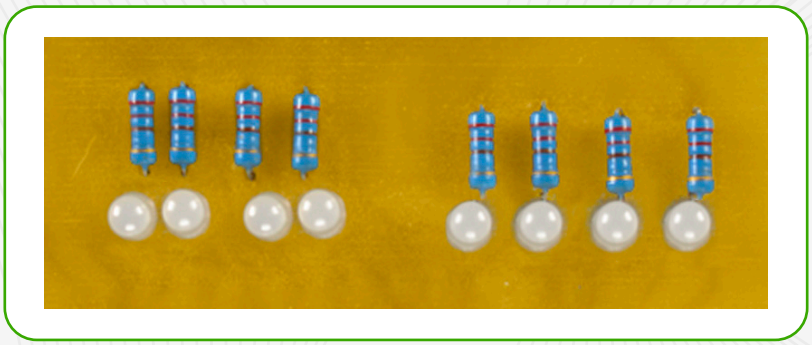
Fuente: Elaboración propia

2.2 Tipos de Salidas y Visualización de Datos

2.2.1 Salida Digital

El conteo se representa visualmente mediante los leds ubicados en la parte inferior de la placa como se observa en la figura 14.

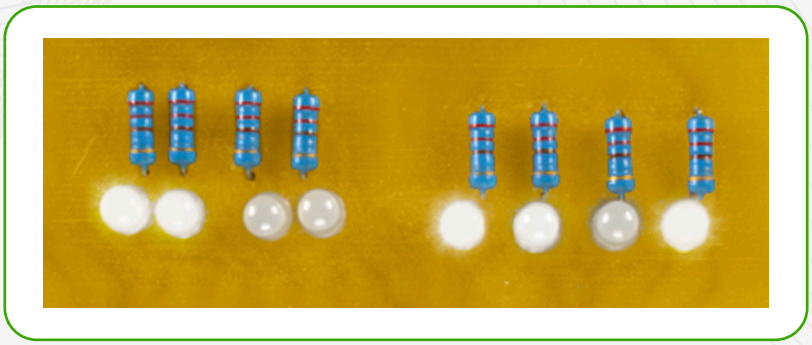
Figura 14. Visualización de datos



Fuente: Elaboración propia

- Cada pasada del imán incrementará el contador
- Los leds mostrarán el valor en formato binario como se observa en la figura 15

Figura 15. Leds mostrarán el valor en formato binario



Fuente: Elaboración propia

2.2.2 Salida Análoga

- El led ubicado junto al botón de reset mostrará el estado análogo del sensor
- La variación del led reflejará las características de la señal detectada
- El pin de salida se encuentra marcado como “Analog out” como se observa en la figura 11.

2.3 Conexión a la Tarjeta de Desarrollo

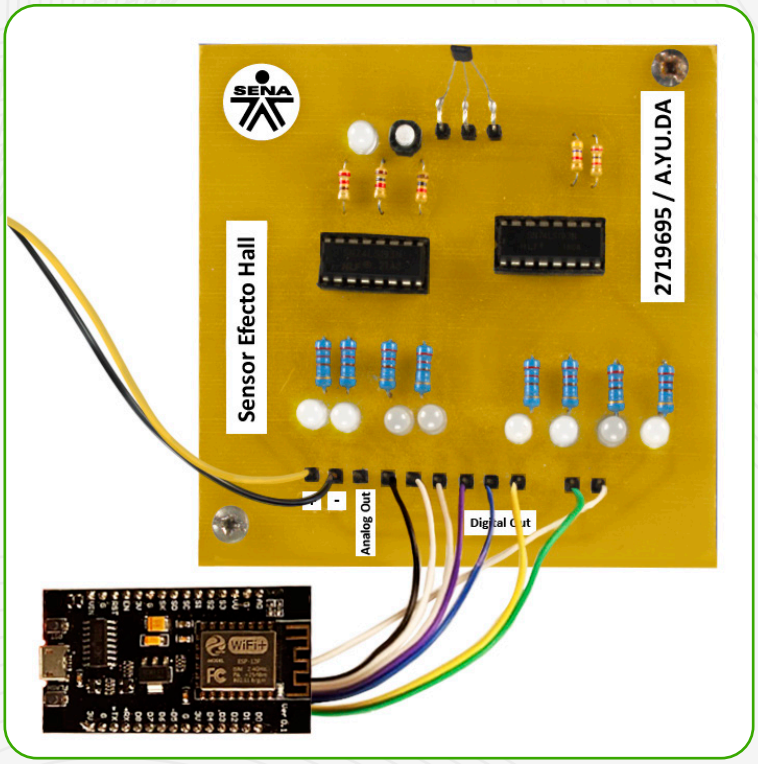
- Conectar las salidas digitales del sensor a la tarjeta de desarrollo (NODEMcu ESP8266), como se observa en la figura 16.

Tabla 15. **Conexión a la Tarjeta de Desarrollo**

Digital out 1	D4
Digital out 2	D5
Digital out 3	D6
Digital out 4	D7
Digital out 5	D0
Digital out 6	D1
Digital out 7	D2
Digital out 8	D3

Fuente: Elaboración propia

Figura 16. **Conexión a la Tarjeta de Desarrollo**



Fuente: Elaboración propia

- Conectar el cable de datos a uno de los puertos del pc como se muestra en la figura 17.

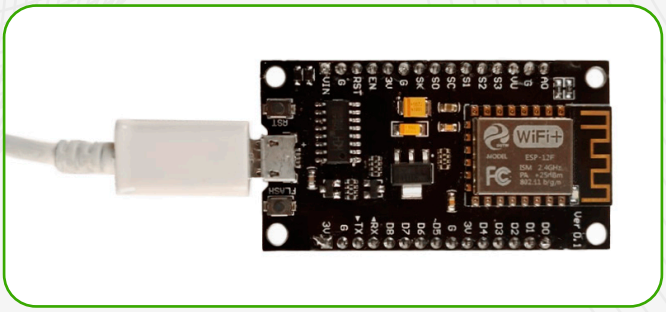
Figura 17. **Conectar el cable de datos a uno de los puertos del pc**



Fuente: Elaboración propia

- Conectar el cable de datos a la tarjeta de desarrollo como se observa en la figura 18.

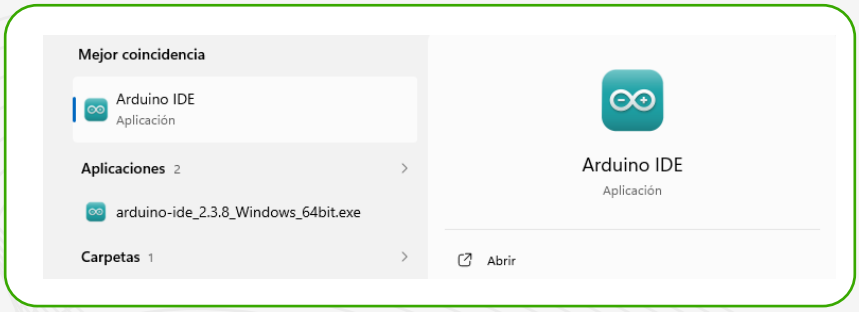
Figura 18. **Conectar el cable de datos a la tarjeta de desarrollo**



Fuente: Elaboración propia

- Abrir la aplicación “ARDUINO IDE” como se observa en la figura 19.

Figura 19. **Abrir la aplicación “ARDUINO IDE”**



Fuente: Elaboración propia

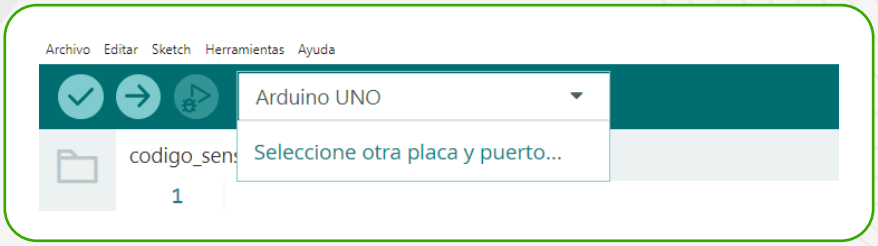
Abrir el código del sensor ubicado en: carpeta Códigos Arduino- sensor de efecto hall - código sensor de efecto hall)

```
/ Pines de entrada desde los contadores
const int bit0Pin = D0;
const int bit1Pin = D1;
const int bit2Pin = D2;
const int bit3Pin = D3;
const int bit4Pin = D4;
const int bit5Pin = D5;
const int bit6Pin = D6;
const int bit7Pin = D7;
void setup() {
Serial.begin(9600);
// Configurar pines como entrada
pinMode(bit0Pin, INPUT);
pinMode(bit1Pin, INPUT);
pinMode(bit2Pin, INPUT);
pinMode(bit3Pin, INPUT);
pinMode(bit4Pin, INPUT);
pinMode(bit5Pin, INPUT);
pinMode(bit6Pin, INPUT);
pinMode(bit7Pin, INPUT);
}
void loop() {
// Leer cada bit
int bit0 = digitalRead(bit0Pin);
int bit1 = digitalRead(bit1Pin);
int bit2 = digitalRead(bit2Pin);
```

```
int bit3 = digitalRead(bit3Pin);
int bit4 = digitalRead(bit4Pin);
int bit5 = digitalRead(bit5Pin);
int bit6 = digitalRead(bit6Pin);
int bit7 = digitalRead(bit7Pin);
// Combinar bits en un número
int count = (bit7 << 7) | (bit6 << 6) | (bit5 << 5) | (bit4 << 4) | (bit3 << 3) | (bit2 << 2) |
(bit1 << 1) | bit0;
// Mostrar resultado en el monitor serial
Serial.print("CONTADOR: ");
Serial.println(count);
delay(300);
}
```

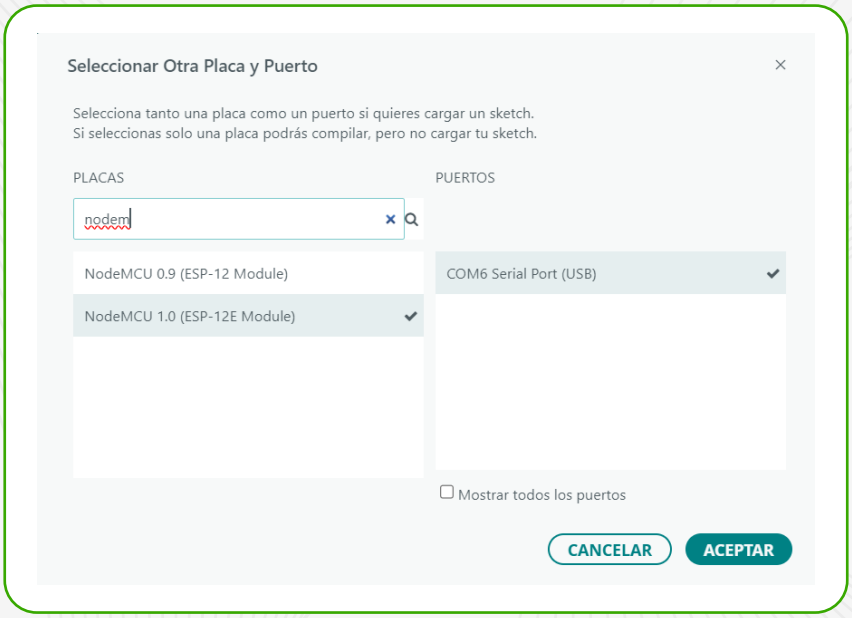
Seleccionar la placa/tarjeta y puerto en el IDE (en este caso nodeMCU 1.0, COM4), como se observa en la figura 20 y 21.

Figura 20. Seleccionar la placa/tarjeta y puerto en el IDE



Fuente: Elaboración propia

Figura 21. **Seleccionar otra placa y puerto**

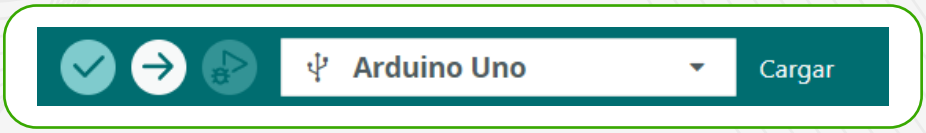


Fuente: Elaboración propia

Click en “aceptar”

- Transferir el código a la tarjeta dando clic en “cargar”, como se observa en la figura 22.

Figura 22. **Transferir el código a la tarjeta dando clic en “cargar”**

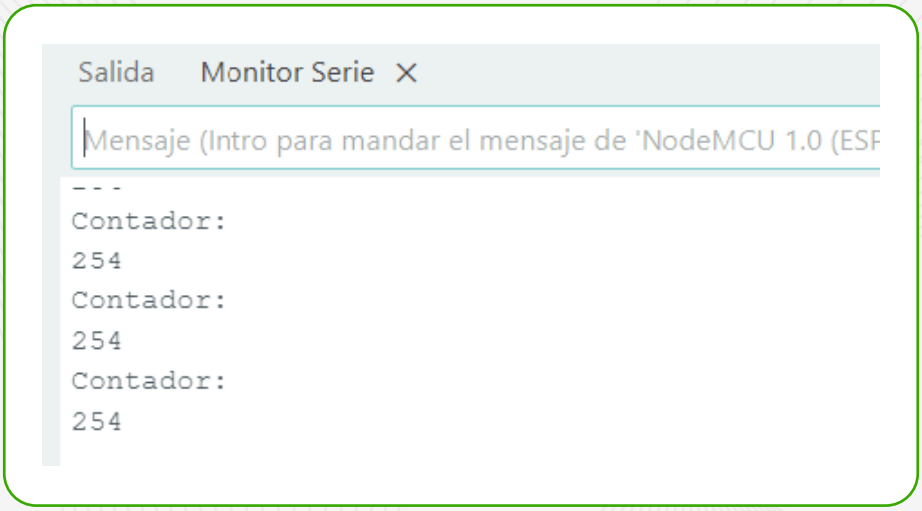


Fuente: Elaboración propia

- Abrir el monitor serial y visualizar datos

Los datos se visualizan en el monitor serial “contador: (número de veces en las que se sometió el sensor a un campo magnético)”, como se observa en la figura 23.

Figura 23. **Monitor serial y visualizar datos**



Fuente: Elaboración propia

2.3.1 Consideraciones Importantes

- Polarización de la fuente: Es de suma importancia energizar correctamente la placa o tarjeta debido a que si no se hace de esta manera quemará los componentes de este.
- Polaridad del Imán: Es fundamental que el imán se sitúe con la polaridad correcta
- Distancia de Detección: Mantenga una distancia apropiada entre el imán y el sensor
- Reinicio/reset: Pulse el botón de reset cada vez que desee iniciar un nuevo conteo

2.3.2 Resolución de Problemas

1. Posibles Inconvenientes

- Si los leds no responden: Verifique la alimentación y la conexión (la conexión debe de ser estable para que no haya error en el conteo)
- Conteo errático: Revise la polaridad del imán
- Sin respuesta del sensor: Compruebe que el botón de reset esté funcionando correctamente

2.3.3 Recomendaciones de Seguridad

- Evite exponer el sensor a descargas electrostáticas
- Evite alimentar la placa a más de 5v
- No someta la placa a golpes o vibraciones excesivas
- Desconecte la alimentación antes de realizar cualquier manipulación
- Energice correctamente la placa

2.3.4 Mantenimiento

- Mantenga la placa y los componentes libres de polvo.
- Verifique periódicamente las conexiones.
- Almacene el sensor en un lugar seco y libre de interferencias magnéticas.

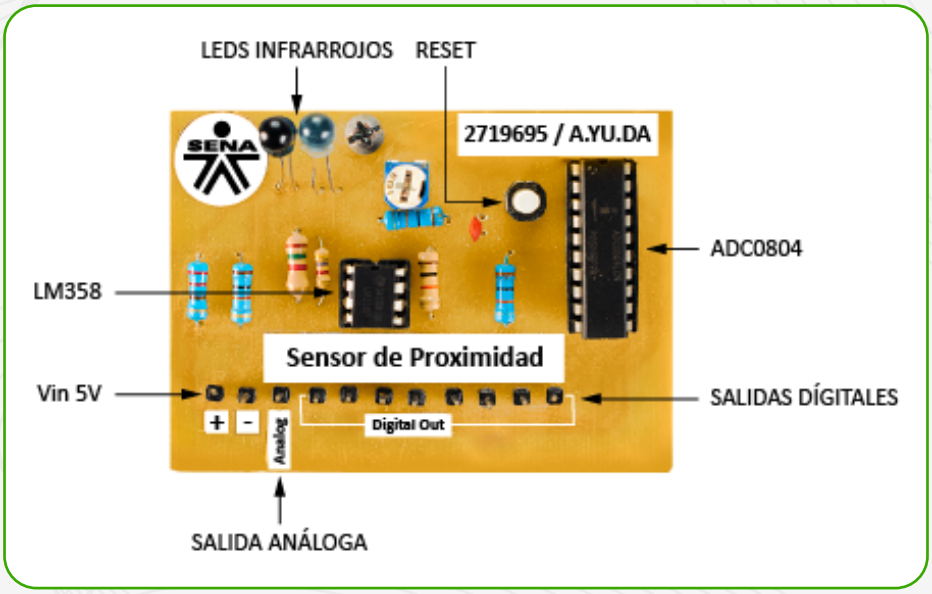
2.4 Sensor de Proximidad Infrarrojo

El sensor de proximidad infrarrojo (IR) es un dispositivo electrónico capaz de detectar la presencia de objetos cercanos mediante el principio de reflexión de la luz infrarroja. Este tipo de sensor emite un haz de luz IR a través de un diodo emisor (LED infrarrojo) y, cuando un objeto se encuentra dentro del rango de detección, parte de esa luz se refleja hacia un fotodiodo o fototransistor receptor. La variación en la cantidad de luz reflejada permite determinar la presencia y, en algunos casos, la distancia aproximada del objeto (Sedra y Smith, 2016).

Los sensores infrarrojos se utilizan ampliamente en sistemas de automatización, robótica, alarmas de seguridad y control de dispositivos, debido a su bajo costo, fácil integración y respuesta rápida ante objetos opacos o superficies reflectantes (Boylestad y Nashelsky, 2019).

En el Kit IoT, el sensor de proximidad infrarrojo (Figura 24) permite detectar el paso o presencia de elementos físicos, enviando una señal digital hacia la tarjeta central de recepción de datos (NodeMCU ESP8266), donde la información es procesada y visualizada en la nube para su análisis.

Figura 24. **Sensor de Proximidad Infrarrojo**



Fuente: Elaboración propia

2.4.1 Funcionamiento

- **Sin objeto:** Luz viaja y no se refleja significativamente ocasionando una variación de voltaje mínima.
- **Objeto cercano:** Luz refleja con más intensidad indicando una variación de voltaje graduales.
- **Objeto muy lejano:** Reflexión de luz mínima enviando variaciones de voltaje significativas que indicaran su lejanía.
- **Uso de salida digital:** al detectar diferentes niveles de voltaje el conversor mostrará su valor en binario
- **Salida análoga:** Al detectar un objeto su nivel de variación de voltaje será de 0V-5V DC

2.4.2 Preparación Inicial

- Fuente alimentación de 5 voltios de DC
- Tarjeta sensor de proximidad
- Objeto que refleje la luz infrarroja

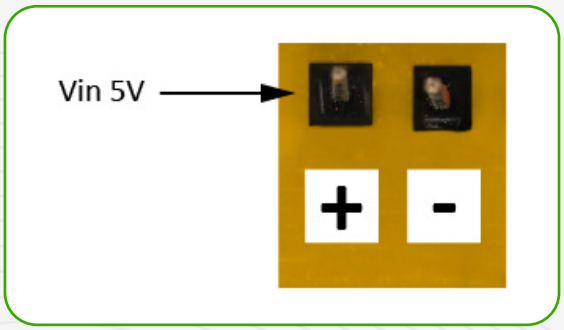
Importante:

- Comprobar el funcionamiento del PCB.
- Se debe usar objetos de colores claro para una mayor eficiencia de lectura en el infrarrojo.
- Siempre verificar que el código se ejecute correctamente.
- Verificar conexiones de puertos, salidas y entradas.

2.4.3 Activación

- Conectar la tarjeta a la fuente de 5V DC según la marca de referencia en la placa (+/-), como se observa en la figura 25

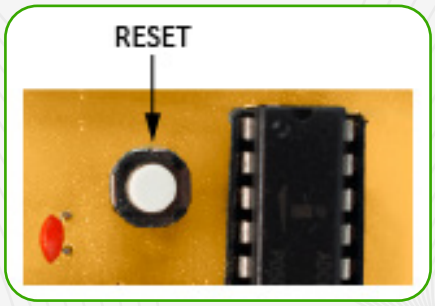
Figura 25. Conectar la tarjeta a la fuente de 5V DC



Fuente: Elaboración propia

- Para activar el conversor es necesario dar 1 un pulso en el reset luego de energizar el circuito, como se observa en la figura 26.

Figura 26. **Reset**



Fuente: Elaboración propia

- Luego de reiniciar el sistema conectar jumpers a cada una de las salidas digitales de la placa, como se observa en la figura 27.

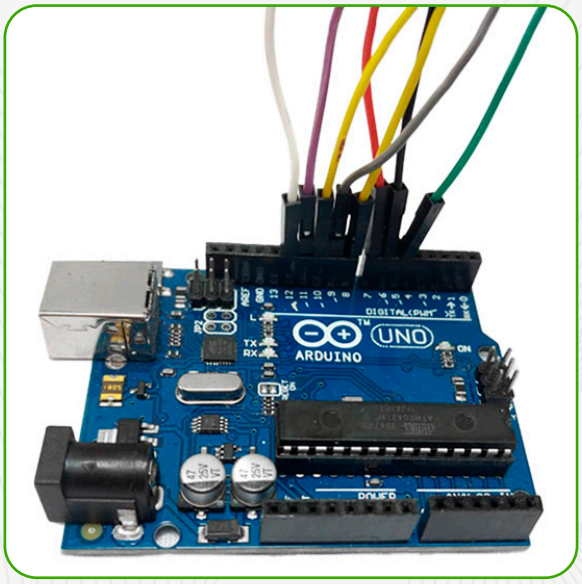
Figura 27. **Digital out**



Fuente: Elaboración propia

- Las salidas van conectadas a una tarjeta de desarrollo (en este caso se usará Arduino uno) de la siguiente manera, como se observa en la figura 28 y 29.

Figura 28. **Salidas van conectadas a una tarjeta de desarrollo**



Fuente: Elaboración propia

2.5 Conexión a la Tarjeta de Desarrollo

En la siguiente tabla, se relaciona la conexión entre los pines de la tarjeta electrónica y el Arduino uno

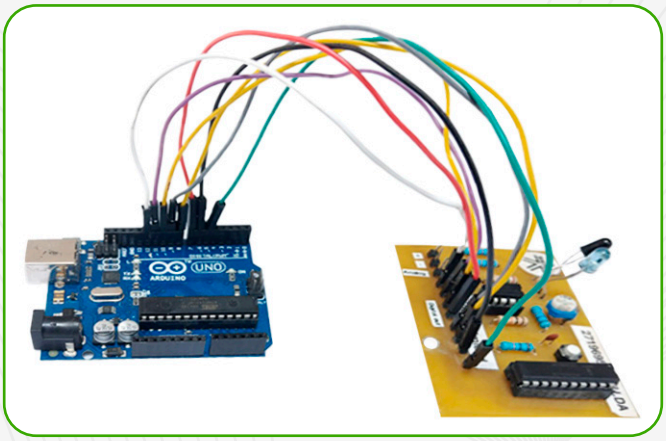
Entonces:

Tabla 2. **Conexión a la Tarjeta de Desarrollo**

PCB	Arduino uno
Digital out 1	Pin 2
Digital out 2	Pin 3
Digital out 3	pin 4
Digital out 4	pin 5
Digital out 5	Pin 6
Digital out 6	Pin 7
Digital out 7	Pin 8
Digital out 8	Pin 9

Fuente: Elaboración propia

Figura 29. **Conexión a la Tarjeta de Desarrollo**



Fuente: Elaboración propia

- Conectar el cable de datos a uno de los puertos del pc, como se observa en la figura 30

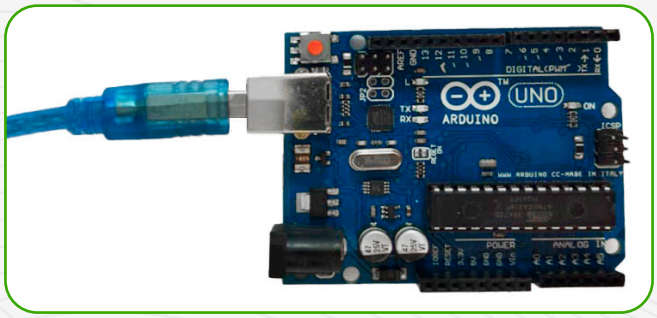
Figura 30. **Conectar el cable de datos a uno de los puertos del pc**



Fuente: Elaboración propia

- Conectar el cable de datos a la tarjeta de desarrollo, como se observa en la figura 31

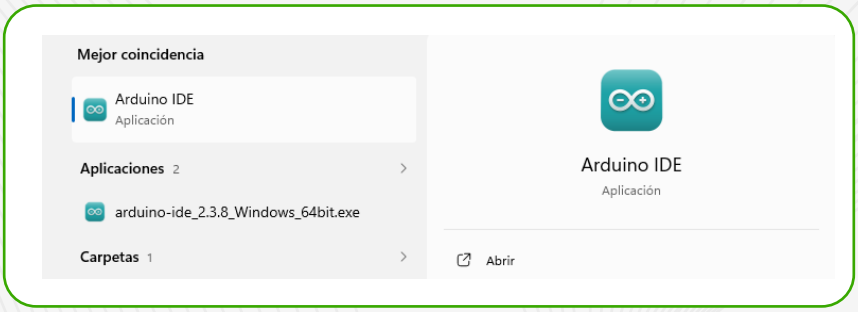
Figura 31. **Conectar el cable de datos a la tarjeta de desarrollo**



Fuente: Elaboración propia

- Abrir la aplicación “ARDUINO IDE”, como se observa en la figura 32

Figura 32. Abrir la aplicación “ARDUINO IDE”



Fuente: Elaboración propia

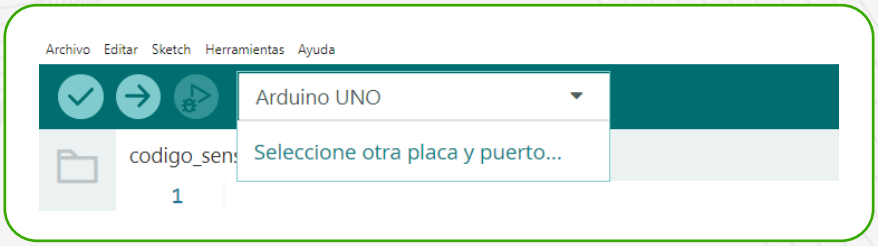
- Luego de ello, abrir el código del sensor ubicado en: carpeta Códigos Arduino- sensor de proximidad-código sensor de proximidad), el cual se muestra a continuación:
- **Código:**

```
const int adcPins[8] = {2, 3, 4, 5, 6, 7, 8, 9}; // D0 a D7 del ADC a pines digitales del Arduino
void setup() {
Serial.begin(9600);
for (int i = 0; i < 8; i++) {
pinMode(adcPins[i], INPUT);
}
}
void loop() {
int value = 0;
// Leer los 8 bits del ADC
for (int i = 0; i < 8; i++) {
int bit = digitalRead(adcPins[i]);
value |= (bit << i);
}
float voltage = value * (5.0 / 255.0);
Serial.print(“Valor ADC: “);
```

```
Serial.print(value);
Serial.print(“ | Voltaje: “);
Serial.print(voltage, 3); // 3 decimales
Serial.println(“ V”);
if (value <= 55) {
Serial.println(“Objeto detectado CERCANO”);
} else {
Serial.println(“Objeto detectado LEJANO”);
}
delay(500);
}
```

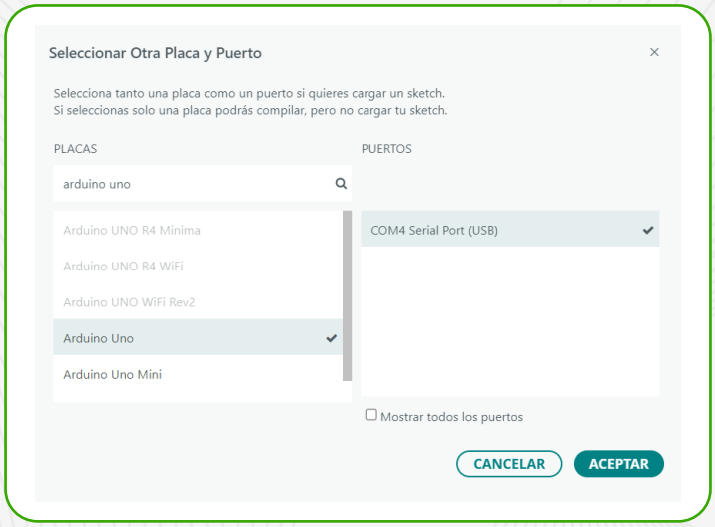
Luego de ello, se selecciona la placa/tarjeta y puerto en el IDE (en este caso Arduino uno, COM4), como se observa en las figuras 33 y 34.

Figura 33. Selecciona la placa/tarjeta y puerto en el IDE



Fuente: Elaboración propia

Figura 34. Selecciona la placa/tarjeta y puerto en el IDE

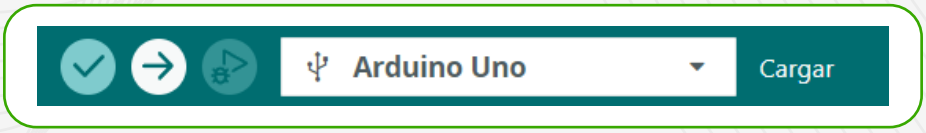


Fuente: Elaboración propia

Click en “aceptar”

- Luego de ello, se realiza la transferencia del código a la tarjeta dando clic en “cargar”, como se observa en la figura 35.

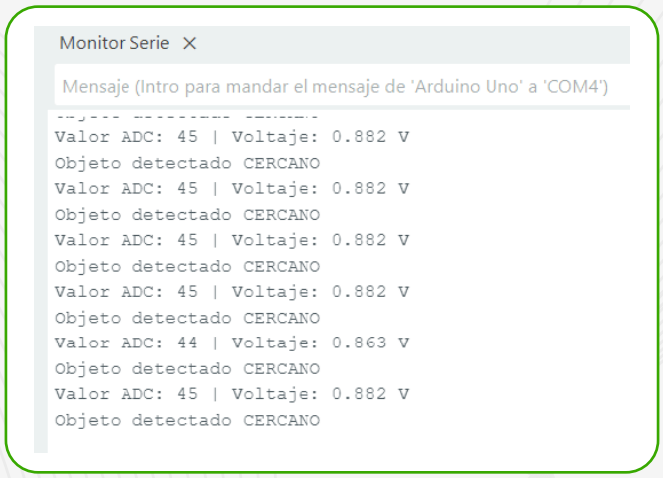
Figura 35. Transferencia del código a la tarjeta dando clic en “cargar”



Fuente: Elaboración propia

- Terminado esto, se abre el monitor serial y se visualizan los datos como se observa en la figura 36.

Figura 36. Se abre el monitor serial y se visualizan los datos



Fuente: Elaboración propia

- En el monitor serial se visualiza si el objeto está lejos (valores ADC mayores a 55) o cerca (valores ADC menores o iguales a 55)

2.5.1 Tener en Cuenta

Material y color de la superficie

Colores claros que causen interferencia y reflexión lumínica más no absorción de la luz

Factores cruciales

- Revisa el campo de detección
- Ajustar la sensibilidad con el potenciómetro
- Eliminé fuentes de luz infrarrojas cercanas o luz ambiente directa
- Pruebe visualmente con un montaje o con un multímetro para ser más preciso

Recomendaciones

- Mantenga el sensor limpio de polvo
- Evite trabajar a temperaturas extremas
- Proteger el sensor de golpes
- Revise la conexión de alimentación según las marcas en la parte inferior de la placa

Mantenimiento

- Limpie con paño seco
- Revise conexiones periódicamente
- Reemplace componentes si muestra signo de deterioro

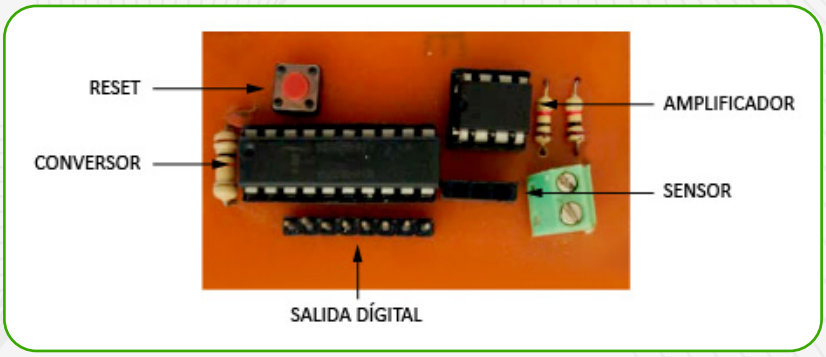
2.6 Sensor de Temperatura

El sensor de temperatura es un dispositivo electrónico diseñado para medir el nivel de calor o frío de un entorno, superficie o componente, convirtiendo esta magnitud física en una señal eléctrica proporcional. Su principio de funcionamiento se basa en la variación de una propiedad eléctrica (como la resistencia o el voltaje) en respuesta a los cambios de temperatura del medio (Sedra y Smith, 2016).

Estos sensores son fundamentales en aplicaciones industriales, domésticas y científicas, ya que permiten el monitoreo y control térmico en sistemas de climatización, automatización, equipos electrónicos y procesos productivos (Boylestad y Nashelsky, 2019).

En el contexto del Kit IoT, el sensor de temperatura (Figura 37) permite registrar las variaciones térmicas del ambiente y enviar los datos a la tarjeta central de recepción de datos (NodeMCU ESP8266), desde donde se procesan y visualizan en la nube, facilitando el seguimiento y análisis en tiempo real.

Figura 37. Sensor de Temperatura



Fuente: Elaboración propia

2.6.1 Funcionamiento

El sensor de temperatura genera una señal analógica proporcional a la variación térmica del entorno, la cual es enviada a un amplificador operacional (OP). Este amplificador incrementa la señal aproximadamente dos veces, mejorando su resolución y precisión antes de ser procesada. Posteriormente, la señal amplificada se dirige hacia el convertidor analógico–digital (ADC0804), encargado de transformar la señal analógica en datos digitales representados en 8 salidas binarias (8 bits).

Estos datos digitales de temperatura pueden visualizarse de dos maneras:

- Mediante LEDs, que muestran la lectura en combinación binaria, o
- A través de un monitor serial, que permite observar el valor numérico de la temperatura procesada por la tarjeta central.

Este proceso permite una adquisición y representación precisa de la temperatura, integrando de forma eficiente las etapas de censado, amplificación y digitalización dentro del sistema de control del Kit IoT.

2.6.2 Preparación Inicial

- Fuente alimentación de 5 voltios de DC
- Tarjeta sensor de temperatura

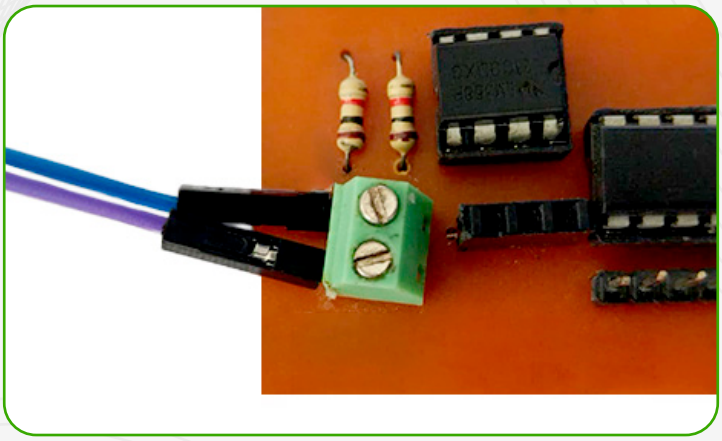
Importante:

- Comprobar el funcionamiento del PCB.
- Siempre verificar que el código se ejecute correctamente.
- Verificar conexiones de puertos, salidas y entradas.

2.6.3 Activación

- Conectar la tarjeta a la fuente de 5V DC (+/-), como se observa en la figura 38

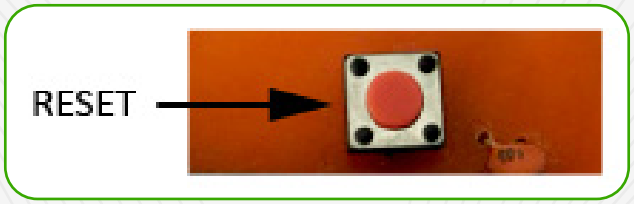
Figura 38. Conectar la tarjeta a la fuente de 5V DC (+/-)



Fuente: Elaboración propia

Para activar el conversor es necesario dar 1 un pulso en el reset luego de energizar el circuito, como se observa en la figura 39.

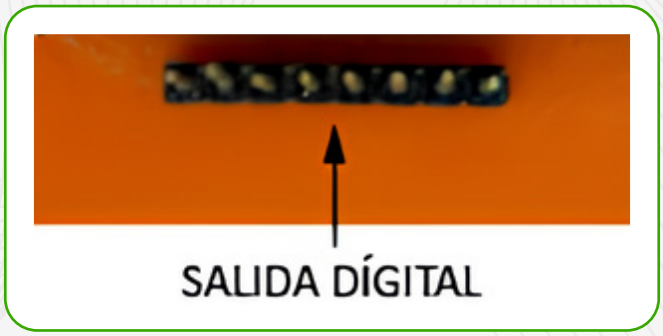
Figura 39. Activar el conversor es necesario dar 1 un pulso en el reset luego de energizar el circuito



Fuente: Elaboración propia

- Luego de reiniciar el sistema se conectan los jumpers a cada una de las salidas digitales de la placa, como se observa en la figura 40

Figura 40. Se conectan los jumpers a cada una de las salidas digitales de la placa



Fuente: Elaboración propia

2.6.4 Conexión a la Tarjeta de Desarrollo

En la siguiente tabla, se relaciona la conexión entre los pines de la tarjeta electrónica y la tarjeta Nodecum.

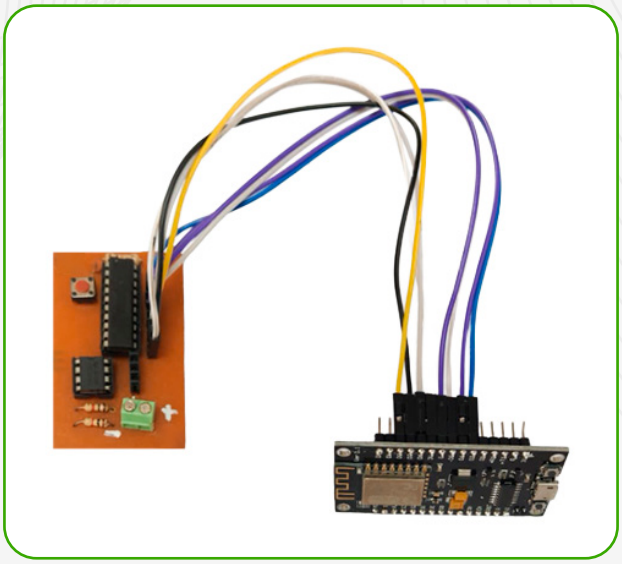
Las salidas van conectadas a una tarjeta de desarrollo (en este caso se usa nodMCU esp8266) de la siguiente manera, como se observa en la figura 41

Tabla 3. **Las salidas van conectadas a una tarjeta de desarrollo**

PCB	nodeMCU
Digital out 1	D0
Digital out 2	D1
Digital out 3	D2
Digital out 4	D3
Digital out 5	D4
Digital out 6	D5
Digital out 7	D6
Digital out 8	D7

Fuente: Elaboración propia

Figura 41. **Las salidas van conectadas a una tarjeta de desarrollo**



Fuente: Elaboración propia

- Luego de ello, se realiza la conexión del cable de datos a uno de los puertos del pc, como se observa en la figura 42.

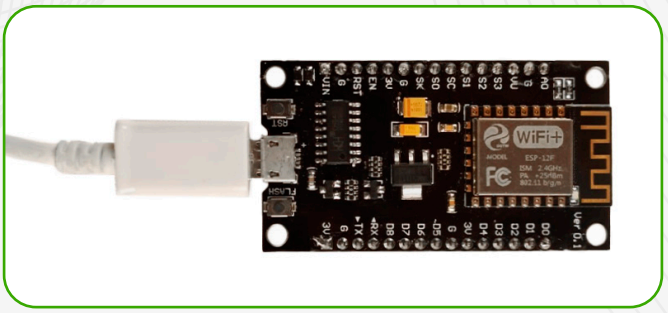
Figura 42. Se realiza la conexión del cable de datos a uno de los puertos del pc



Fuente: Elaboración propia

- Así mismo, se realiza la conexión al cable de datos a la tarjeta de desarrollo, como se observa en la figura 43.

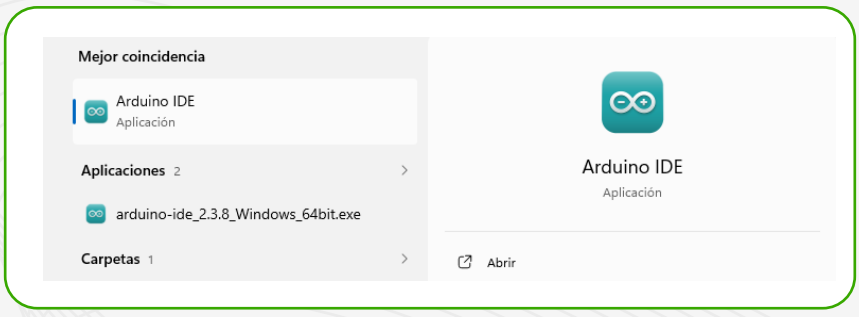
Figura 43. Se realiza la conexión al cable de datos a la tarjeta de desarrollo



Fuente: Elaboración propia

- Abrir la aplicación “ARDUINO IDE”, como se observa en la figura 44.

Figura 44. Abrir la aplicación “ARDUINO IDE”



Fuente: Elaboración propia

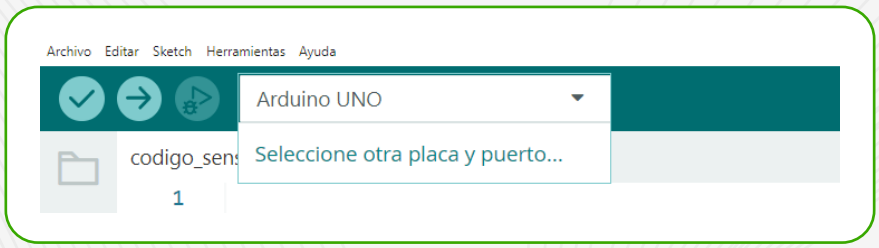
Para terminar, se debe cargar el código del sensor ubicado en: carpeta Códigos Arduino- sensor de temperatura -código sensor de temperatura), como se muestra a continuación:

- **Código:**

```
const int adcPins[8] = {D0, D1, D2, D3, D4, D5, D6, D7 };
void setup() {
Serial.begin(9600);
for (int i = 0; i < 8; i++) {
pinMode(adcPins[i], INPUT);
}
}
void loop() {
int value = 0;
// Leer los 8 bits del ADC
for (int i = 0; i < 8; i++) {
int bit = digitalRead(adcPins[i]);
value |= (bit << i);
}
Serial.println("TEMPERATURA °C: ");
Serial.println(value);
delay(500);
}
```

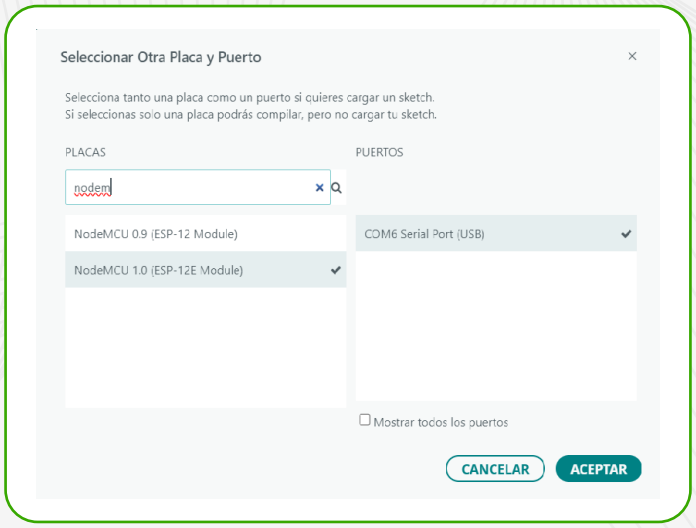
Luego de ello, se realiza la selección de la placa/tarjeta y puerto en el IDE (en este caso Arduino uno, COM4) como se observa la figura 45 y 46.

Figura 45. Se realiza la selección de la placa/tarjeta y puerto en el IDE (en este caso Arduino uno, COM4)



Fuente: Elaboración propia

Figura 46. Se realiza la selección de la placa/tarjeta y puerto en el IDE (en este caso Arduino uno, COM4)

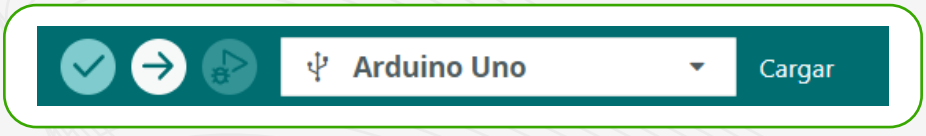


Fuente: Elaboración propia

Click en “aceptar”

Para terminar, se debe cargar el código del sensor ubicado en: (carpeta Códigos Arduino- sensor de temperatura -código sensor de temperatura), como se muestra a continuación:

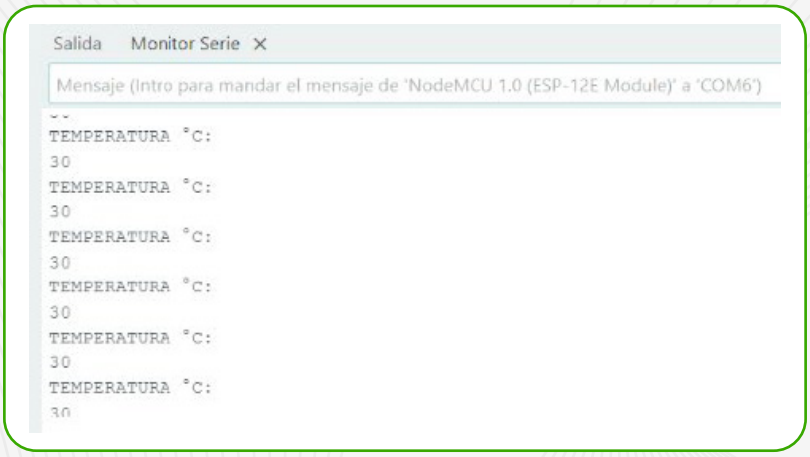
Figura 47. Se transfiere el código a la tarjeta dando clic en “cargar”



Fuente: Elaboración propia

- Se debe abrir el monitor serial y visualizar datos, como se observa en la figura 48

Figura 48. Se debe abrir el monitor serial y visualizar datos



Fuente: Elaboración propia

En el monitor serial se visualiza la temperatura a la que el sensor está sometido.

2.6.5 Tener en Cuenta

Factores cruciales

- Pruebe visualmente con un montaje o con un multímetro para ser más preciso
- En caso de usar un encendedor para notar los cambios de temperatura no acercarlo mucho al sensor ya que puede causar daños a los componentes de la tarjeta.

Importante:

- Mantenga el sensor limpio de polvo
- Proteger el sensor de golpes
- Revise la conexión de alimentación según las marcas en la parte inferior de la placa

Mantenimiento

- Limpie con paño seco
- Revise conexiones periódicamente
- Reemplace componentes si muestra signo de deterioro

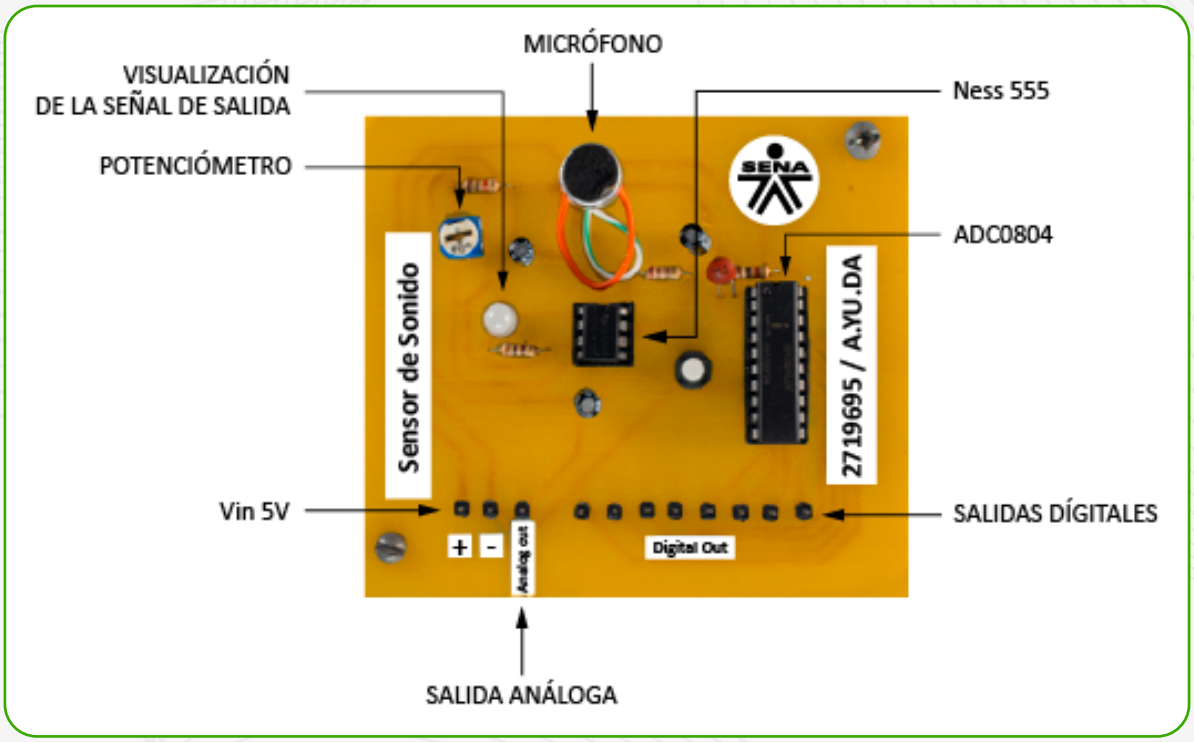
2.7 Sensor de Sonido

El sensor de sonido es un dispositivo electrónico que permite detectar y medir las variaciones de presión acústica en el ambiente, convirtiendo las ondas sonoras en señales eléctricas que pueden ser interpretadas por un sistema de control o procesamiento. Generalmente, utiliza un micrófono electret o MEMS (Micro-Electro-Mechanical System) como elemento principal, el cual genera una señal de voltaje proporcional a la intensidad del sonido (Horowitz y Hill, 2015).

Estos sensores son ampliamente empleados en sistemas de automatización, monitoreo ambiental, robótica, domótica y control de dispositivos por voz, ya que permiten reconocer niveles de ruido, patrones acústicos o eventos específicos (Khandpur, 2017).

En el Kit IoT, el sensor de sonido como se observa en la figura 49 se encarga de detectar cambios en el nivel sonoro del entorno y enviar la señal correspondiente a la tarjeta central de recepción de datos (NodeMCU ESP8266), donde es procesada y transmitida a la nube, permitiendo su visualización y análisis en tiempo real.

Figura 49. Sensor de Sonido



Fuente: Elaboración propia

2.7.1 Características Principales

- Voltaje de Alimentación: 5V DC
- Corriente: 300mah
- Tipo de sensor: Detección de Niveles de Sonido
- Función: Conversión de Señal Analógica a Digital
- Activación: Mediante Pulsador
- Indicación: Led Indicador

2.7.2 Tener en Cuenta

Preparación inicial

- Fuente alimentación de 5 voltios de DC
- Tarjeta sensor de sonido

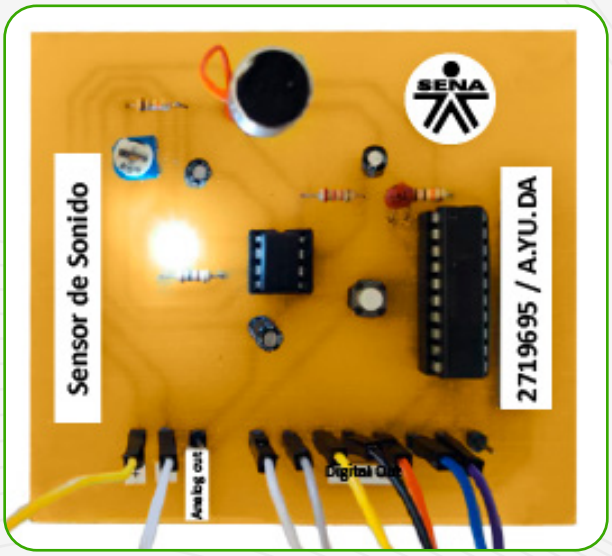
Importante

- Comprobar el funcionamiento del PCB.
- Siempre verificar que el código se ejecute correctamente.
- Verificar conexiones de puertos, salidas y entradas.

Energización

- Conecte la placa al suministro de 5V DC utilizando una fuente de alimentación estable.
- Verificación del Led, como se observa en la figura 50

Figura 50. Verificación del Led



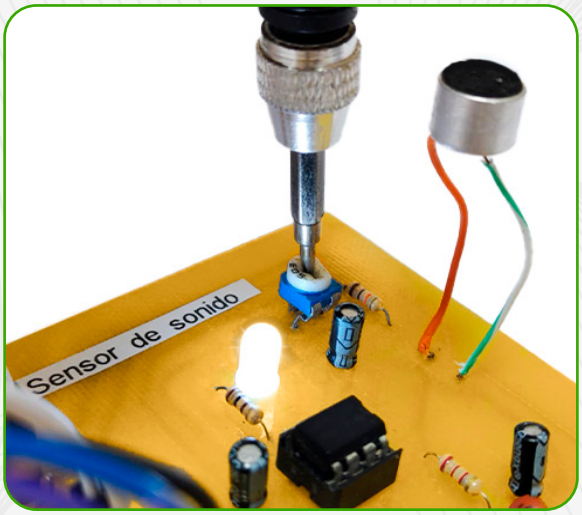
Fuente: Elaboración propia

- Al energizar, el led/indicador análogo debe estar apagado.
- Si el led/ indicador análogo permanece encendido, proceda con el ajuste del regulador/potenciómetro.

2.7.3 Ajuste del Potenciómetro

Localice el potenciómetro de color azul en la placa, como se observa en la figura 51

Figura 51. Ajuste del Potenciómetro



Fuente: Elaboración propia

Si el led permanece encendido:

- Gire suavemente el potenciómetro.
- Objetivo: Apagar el led levemente.

2.7.4 Activación

Pulse el botón de inicio para activar la conversión analógica a digital.

Importante:

- No lo apague completamente, ya que esto impedirá la lectura correcta de los valores.
- Hacer las conexiones a la nodemcu
- Conectar el cable datos a un computador, como se observa en la figura 52

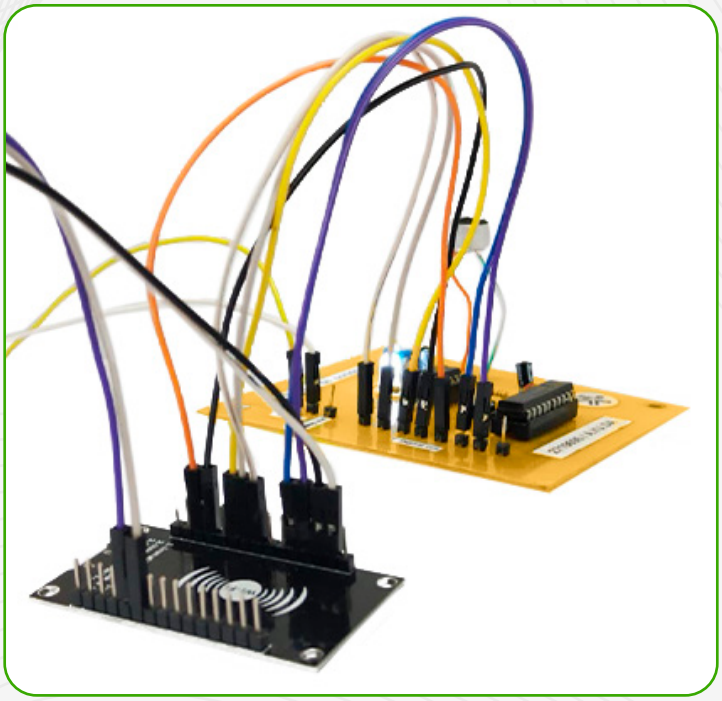
Figura 52. Conectar el cable datos a un computado



Fuente: Elaboración propia

- Conectar el positivo al pin 3v de la nodemcu y el negativo al pin GND
- Conectar los pines digitales desde el D0 hasta el D7 y de ultimo RX y TX, como se observa en la figura 53.

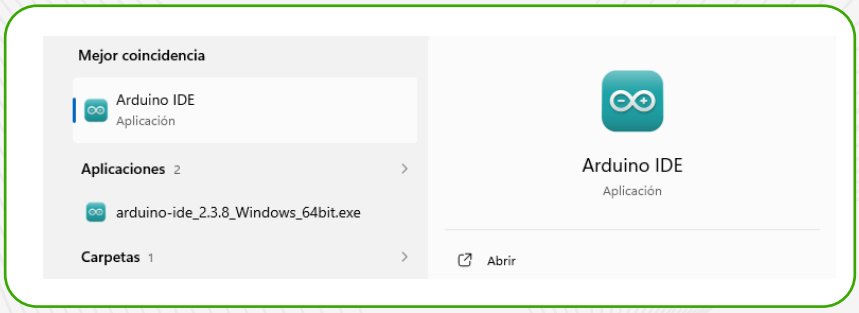
Figura 53. Conectar los pines digitales desde el D0 hasta el D7 y de ultimo RX y TX



Fuente: Elaboración propia

- Abrir el entorno de programación IDE Arduino, como se observa en la figura 54

Figura 54. Abrir el entorno de programación IDE Arduino



Fuente: Elaboración propia

Abrir el código del sensor ubicado en: carpeta Códigos Arduino- sensor de sonido-código sensor de sonido) y pegarlo en IDE Arduino, como se observa en la figura 55.

Figura 55. Abrir el código del sensor ubicado en: carpeta Códigos Arduino- sensor de sonido-código sensor de sonido) y pegarlo en IDE Arduino



Fuente: Elaboración propia

- **Código:**

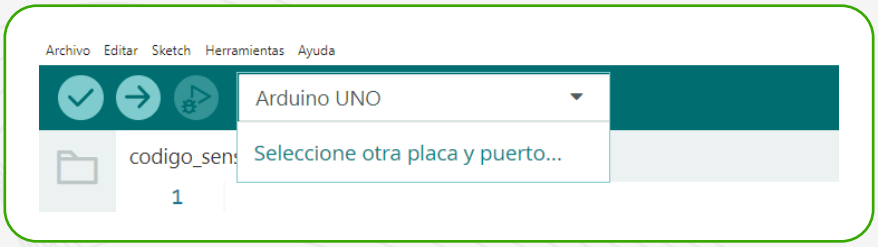
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
// OLED config
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
// Pines de datos del ADC0804
const uint8_t dataPins[8] = {
  16,  // D0
   5,  // D1
   4,  // D2
   0,  // D3
   2,  // D4
  14,  // D5
  12,  // D6
  13   // D7
};
void setup() {
  Serial.begin(9600);
  // Configurar pines del ADC como entradas
  for (int i = 0; i < 8; i++) {
    pinMode(dataPins[i], INPUT);
  }
 // Inicializar pantalla OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
```

```
    Serial.println(F("No se encontró pantalla OLED"));
    while (true);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Sensor de Sonido");
  display.display();
  delay(1000);
}
void loop() {
  uint8_t digitalValue = 0;
  // Leer los 8 bits del ADC
  for (int i = 0; i < 8; i++) {
    digitalValue |= (digitalRead(dataPins[i]) << i);
  }
  // Convertir valor a voltaje (0-5V)
  float voltage = (digitalValue * 5.0) / 255.0;
  // Mostrar en consola
  Serial.print("Valor ADC: ");
  Serial.print(digitalValue);
  Serial.print(" -> Voltaje: ");
  Serial.print(voltage, 2);
  Serial.println(" V");
  // Dibujar en pantalla
  display.clearDisplay();
  // Título
```

```
 display.setCursor(0, 0);
 display.setTextSize(1);
 display.println("Sensor de Sonido");
 // Valor ADC
 display.print("Valor ADC: ");
 display.println(digitalValue);
 // Voltaje
 display.setTextSize(2);
 display.setCursor(0, 20);
 display.print(voltage, 2);
display.println(" V");
 // Barra de nivel (tipo medidor)
 int barWidth = map(digitalValue, 0, 255, 0, 100); // Escala 0-100 píxeles
 display.drawRect(0, 50, 100, 10, SSD1306_WHITE);   // Marco de la barra
 display.fillRect(0, 50, barWidth, 10, SSD1306_WHITE); // Relleno según valor
 display.display();
delay(200); // Refresco
}
```

- Seleccionar la placa/tarjeta y puerto en el IDE (en este caso Arduino uno, COM4), como se observa en la figura 56.

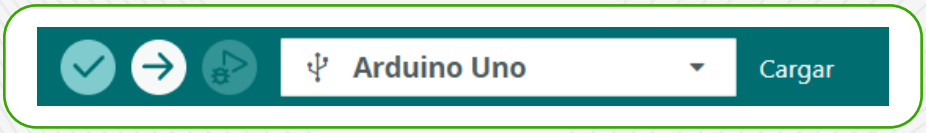
Figura 56. Selección puerto



Fuente: Elaboración propia

- Transferir el código a la tarjeta dando clic en “cargar”, figura 56

Figura 57. Transferir el código a la tarjeta dando clic en “cargar”



Fuente: Elaboración propia

2.7.5 Funcionamiento del Sensor

Detección de Sonido

- El sensor detecta niveles de sonido: este es representado en valor ADC y con conversión a voltios junto a una barra que se mueve en base a la intensidad del sonido, como se observa en la figura 58.

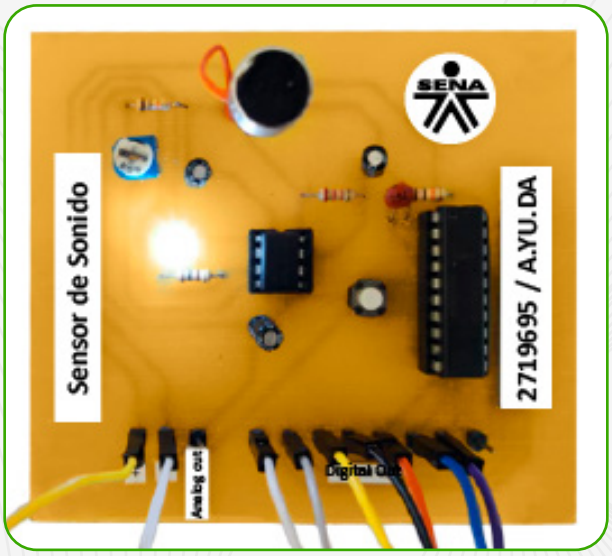
Figura 58. Funcionamiento del Sensor



Fuente: Elaboración propia

- Sonidos altos
- Sonidos muy bajos
- Convierte las señales detectadas en salidas digitales (binarias)
- El led indicador solo debe encender al detectar sonido, como se muestra en la figura 58

Figura 59. El led indicador solo debe encender al detectar sonido



Fuente: Elaboración propia

Recomendaciones de Uso

- Mantenga el sensor alejado de fuentes de interferencia electromagnética como imanes.
- Asegure una conexión estable de 5V DC.
- Evite movimientos bruscos cerca del sensor durante la medición para evitar malas lecturas
- Utilizar el código para tener el correcto funcionamiento

Solución de Problemas

Led Siempre Encendido:

- Ajuste el potenciómetro azul hasta que se apague
- Verifique la alimentación

Sin Respuesta:

- Compruebe las conexiones eléctricas
- Verifique la fuente de alimentación

Precauciones

- No exponga el sensor a condiciones extremas de humedad o temperatura.
- Evite tocar los componentes electrónicos directamente.
- Desconecte la alimentación antes de realizar cualquier manipulación.

Especificaciones Técnicas

- Voltaje de Operación: 5V DC
- Tipo de Salida: Digital (Binaria) con conversión a ADC y a voltios
- Método de Activación: Pulsador
- Indicador: Led

Nota Importante:

El ajuste preciso del potenciómetro es crucial para el funcionamiento correcto del sensor, ajuste demasiado sensible o poco sensible puede afectar la precisión de las lecturas y directamente el funcionamiento.

- El código se ejecutó uy probó en NODEMCU y esp8266

Referencias

Boylestad, R. L., & Nashelsky, L. (2019). Electronic Devices and Circuit Theory (12th ed.). Pearson Education.

Hall, E. H. (1879). On a New Action of the Magnet on Electric Currents. American Journal of Mathematics, 2(3), 287–292. https://doi.org/10.2307/2369245

Horowitz, P., & Hill, W. (2015). The Art of Electronics (3rd ed.). Cambridge University Press.

Khandpur, R. S. (2017). Handbook of Biomedical Instrumentation (3rd ed.). McGraw-Hill Education.

Ebay. (2025). Nuevo módulo Serie NodeMCU V3 ESP8266 CP2102 ESP-12E Placa de Desarrollo WiFi USB-C-.

Sedra, A. S., & Smith, K. C. (2016). Microelectronic Circuits (7th ed.). Oxford University Press.