

PRÁCTICA DE ARDUINO - SISTEMA DE ALARMA CON SENSORES PIR Y NOTIFICACIONES MÓVILES

NIVEL: 2

DURACIÓN: 2 horas

OBJETIVO: Los estudiantes aprenderán a crear un sistema de alarma con sensores PIR y notificaciones móviles usando ESP8266 y Blynk

MATERIALES

- ESP8266 o ESP32
- Sensor PIR
- Cables de conexión
- Protoboard (opcional)
- Alimentación
- Aplicación móvil o servicio de notificaciones (Blynk)
- Cables y protoboard
- Computadora con el software Arduino IDE instalado
- Proteus

1. DESARROLLO DE LA PRÁCTICA

1.1 SIMULACIÓN EN PROTEUS

1.1.1 MONTAJE

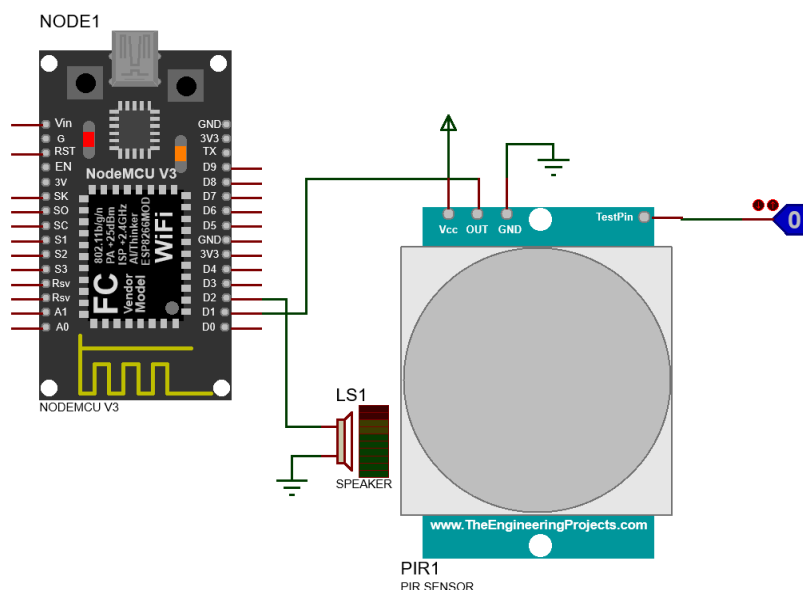


Figura 1. Montaje en pines Digitales

2.2 PROGRAMACIÓN EN ARDUINO IDE

```
#define BLYNK_TEMPLATE_ID "TuTemplateID"// Coloca aquí tu Template ID
#define BLYNK_TEMPLATE_NAME "TuTemplateName"// Coloca aquí tu Template Name

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Credenciales de Blynk
char auth[] = "TuAuthToken";// Token de autenticación de Blynk
char ssid[] = "TuSSID"; // Nombre de la red WiFi
char pass[] = "TuPassword";// Contraseña de la red WiFi

#define pirPin 5 // GPIO 5 (D1 en NodeMCU) para el sensor PIR
#define sirenaPin 4 // GPIO 4 (D2 en NodeMCU) para la sirena

void setup() {
  // Configuración de la conexión serial y Blynk
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);// Conectar a Blynk

  // Configuración de pines
  pinMode(pirPin, INPUT); // Configura el pin del PIR como entrada
  pinMode(sirenaPin, OUTPUT);// Configura el pin de la sirena como salida
  digitalWrite(sirenaPin, LOW); // Asegúrate de que la sirena esté apagada al inicio

  Serial.println("Sistema de alarma listo.");
}
```

```
void loop() {  
  Blynk.run(); // Mantiene la conexión con Blynk  
  
  // Leer el estado del sensor PIR  
  int pirValue = digitalRead(pirPin);  
  
  if (pirValue == HIGH) { // Si detecta movimiento  
    Serial.println("¡Movimiento detectado!");  
  
    // Activar la sirena  
    digitalWrite(sirenaPin, HIGH);  
  
    // Mantener la sirena encendida por 5 segundos  
    delay(5000);  
  
    // Apagar la sirena  
    digitalWrite(sirenaPin, LOW);  
  }  
}
```

Explicación

```
#define BLYNK_TEMPLATE_ID "TuTemplateID"
```

```
#define BLYNK_TEMPLATE_NAME "TuTemplateName"
```

- Estos identificadores son proporcionados por la plataforma Blynk. El Template ID y el Template Name permiten que el dispositivo se conecte a un proyecto específico en la aplicación de Blynk.
- La función Blynk.run() se encarga de mantener la conexión entre el ESP8266 y el servidor de Blynk.
- La lectura del sensor PIR se realiza con digitalRead(pirPin), y si detecta movimiento (HIGH), se enciende la sirena (digitalWrite(sirenaPin, HIGH)).
- La sirena se apaga automáticamente después de 5 segundos (delay(5000)), apagándose con digitalWrite(sirenaPin, LOW).

2. ANÁLISIS DE RESULTADOS

- ¿Cuál es la función principal del sistema y cómo responde cuando se detecta movimiento?

La función principal del sistema es detectar movimiento utilizando un sensor PIR y activar una sirena cuando se detecte. Cuando el sensor PIR detecta movimiento, el sistema activa la sirena durante 5 segundos y luego la apaga automáticamente. Además, se imprime un mensaje en el monitor serial indicando que se ha detectado movimiento.

- ¿Qué papel juega la conexión a Blynk en este sistema?

La conexión a Blynk permite que el sistema se integre con una aplicación móvil para monitoreo o control remoto. Aunque en este código no se implementan funciones avanzadas de Blynk, se establece la conexión básica que podría usarse para enviar notificaciones o controlar la sirena de manera remota desde la aplicación de Blynk en un dispositivo móvil.

- ¿Qué sucede si el sistema no detecta movimiento durante un largo período de tiempo?

Si no se detecta movimiento, el sistema simplemente seguirá monitoreando de forma pasiva el estado del sensor PIR. No se ejecutará ninguna acción ni se activará la sirena hasta que se detecte movimiento. Esto asegura que el sistema no consuma energía innecesaria o active la sirena de forma errónea.

- ¿Cómo se garantiza que la sirena se apaga después de sonar durante 5 segundos?

La función `digitalWrite(sirenaPin, LOW)` se ejecuta después de un `delay(5000)`, lo que garantiza que la sirena esté activa durante exactamente 5 segundos. Después de este tiempo, la sirena se apaga automáticamente.

- ¿Cómo podrías extender este sistema para que envíe notificaciones cuando se detecte movimiento?

Para extender el sistema y enviar notificaciones, podrías agregar una función de `Blynk.notify()` dentro del bloque condicional que detecta el movimiento. Esto permitiría que el sistema envíe una alerta a la aplicación móvil de Blynk cuando se active la sirena o cuando el sensor PIR detecte movimiento.

- ¿Qué cambios podrían hacerse para mejorar la funcionalidad del sistema?
 - Agregar notificaciones automáticas a través de Blynk cuando se detecte movimiento.
 - Implementar un botón virtual en la aplicación Blynk para permitir a los usuarios activar o desactivar manualmente la sirena desde su dispositivo móvil.
 - Ajustar la sensibilidad del sensor PIR para minimizar falsos positivos en ambientes con muchas interferencias.

3. CONCLUSIONES

- a. Este código proporciona una solución sencilla y eficaz para implementar un sistema de alarma de bajo costo basado en un ESP8266, un sensor PIR y una sirena. La detección de movimiento está bien implementada, y el sistema funciona de manera reactiva al detectar intrusiones.

- b. Aunque el código establece la conexión con la plataforma Blynk, no se implementan todas las características que Blynk ofrece, como notificaciones automáticas, botones de control o monitoreo remoto desde una aplicación móvil. Aun así, el sistema puede expandirse fácilmente para aprovechar todas estas funcionalidades.