



TEMA 10. PROYECTOS INTERMEDIOS

.SENSOR PIR

El sensor PIR (Passive Infrared Sensor) es un dispositivo utilizado para detectar movimiento a través de cambios en los niveles de radiación infrarroja emitida por los objetos dentro de su rango de detección. Es comúnmente utilizado en sistemas de seguridad, domótica y automatización.

1. Principio de Funcionamiento

El sensor PIR detecta la radiación infrarroja emitida por objetos cálidos (como humanos o animales).

1. Sensor PIR:

- Contiene un material piroeléctrico que responde a los cambios en la radiación IR.
- No emite radiación; solo detecta.

2. Lente Fresnel:

- Aumenta el rango y el ángulo de detección.
- Divide el campo de visión en zonas para detectar cambios en la radiación IR.

3. Cambio de estado:

- Detecta movimiento al identificar diferencias en las zonas de radiación IR.
- Cuando una fuente de calor entra o sale de su rango, genera una señal.

Características del Módulo PIR

Característica	Especificación
Voltaje de operación	5V - 12V
Corriente de operación	~50 μ A (muy bajo consumo)
Rango de detección	3 - 7 metros
Ángulo de detección	100° - 120°
Tiempo de retardo ajustable	5 segundos a 5 minutos
Modo de repetición	Único disparo o repetitivo (configurable)
Salida	Digital (HIGH cuando detecta movimiento)





DESCRIPCIÓN

Los sensores infrarrojos pasivos (PIR) son ideales para proyectos de detección de movimientos. El sensor HC-SR501, sólo funciona cuando alguien se mueve en la franja que puede barrer su detector. Puede detectar movimiento de 3 hasta 7 metros de distancia. Este sensor de movimiento PIR tiene 3 pines, VCC, OUTPUT y GND, 2 potenciómetros para ajustar la sensibilidad y la demora. El retardo se puede configurar entre 5 y 300 segundos mientras que el potenciómetro de sensibilidad ajusta el rango de detección de aproximadamente 3 metros a 7 metros.



CARACTERÍSTICAS TÉCNICAS

- Tipo de producto: HC-SR501
- Rango de voltaje: DC 4,5-20V
- Consumo de corriente: <60uA
- Voltaje de salida: señal de nivel alto / bajo: 3.3V salida TTL
- Distancia de detección: 3 – 7 mts (se puede ajustar)
- Rango de detección: <140 °
- Tiempo de retardo: 5-200S (se puede ajustar, 5s por defecto +/-3%)
- Tiempo de bloqueo: 2.5 S (predeterminado)
- Disparador: L: disparador no repetible
- Temperatura de trabajo: -20°C a +80°C
- Método de disparo: disparador L irrepitable / disparador H repetible
- Dimensiones: 3.2cm x 2.4cm x 1.8cm (aproximado)





Configuración de Ajustes del PIR

1. Tiempo de Retardo:

- Se ajusta mediante un trimmer.
- Define el tiempo que la salida permanece en HIGH tras detectar movimiento.
- Rango típico: 5 segundos a 5 minutos.

2. Sensibilidad:

- Ajusta el rango de detección (distancia).
- Se configura con otro trimmer.

3. Modo de Repetición:

- H (Repetitivo): Mientras detecte movimiento, mantiene la salida en HIGH.
- L (Único disparo): Genera una señal HIGH una sola vez por evento.

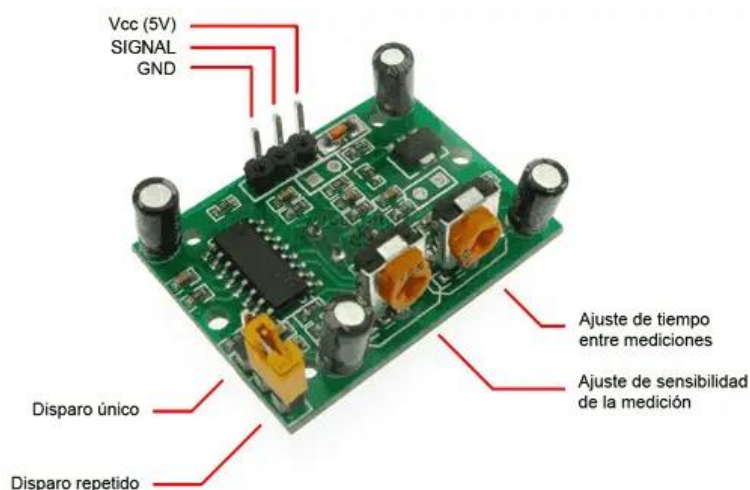
6. Conexión con Arduino

Circuito Básico

1. Conecta el pin VCC del módulo PIR al pin 5V de Arduino.
2. Conecta el pin GND del módulo PIR al GND de Arduino.
3. Conecta el pin OUT del PIR a un pin digital de Arduino (ej. D2).

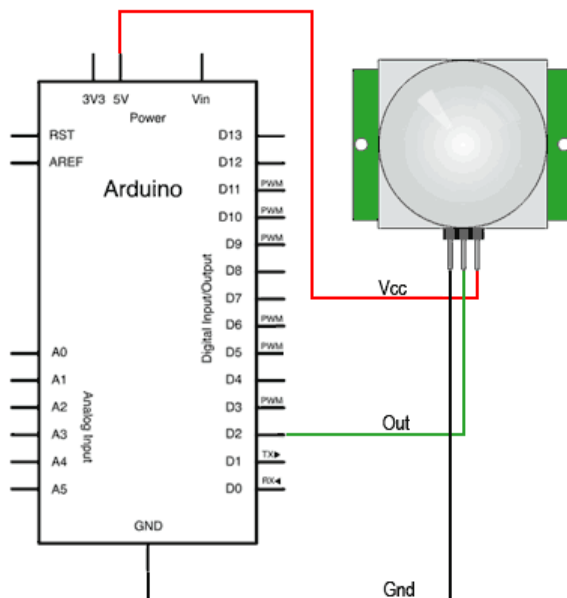
Esquema eléctrico

Este es el esquema de patillaje de un sensor PIR.



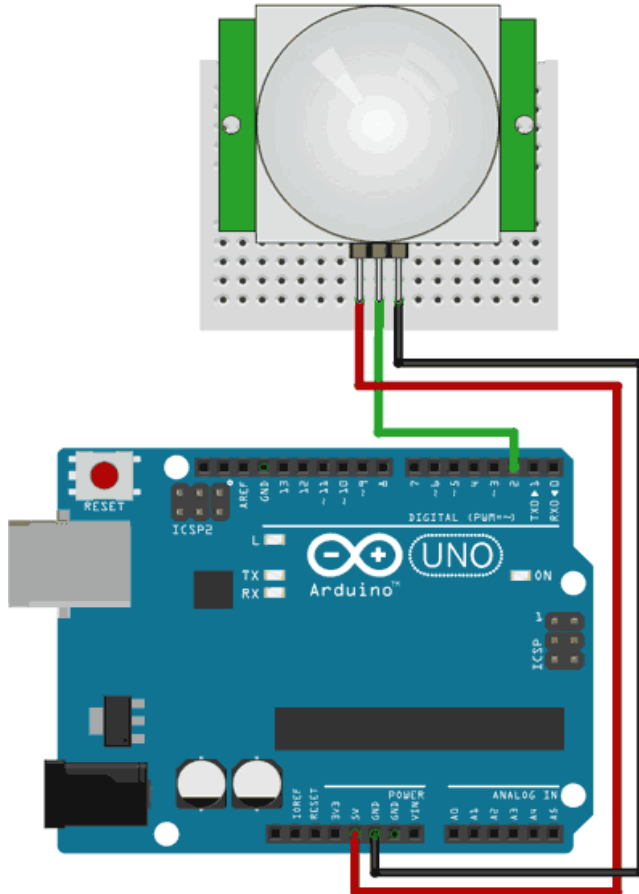


El esquema eléctrico que necesitamos es el siguiente.



arduino-pir-esquema-electrico

Mientras que el montaje en una protoboard sería el siguiente.





Ejemplo de Código

Este programa enciende un LED cuando el PIR detecta movimiento y envía un mensaje al monitor serie.

```
const int pirPin = 2;  // Pin conectado al OUT del PIR

const int ledPin = 13; // Pin del LED (integrado en Arduino)

int pirState = LOW;    // Estado inicial del sensor

void setup() {
  pinMode(pirPin, INPUT); // Configura el pin del PIR como entrada
  pinMode(ledPin, OUTPUT); // Configura el LED como salida
  Serial.begin(9600);      // Inicia la comunicación serial
  Serial.println("Sensor PIR listo...");
}

void loop() {
  int motion = digitalRead(pirPin); // Leer el estado del PIR

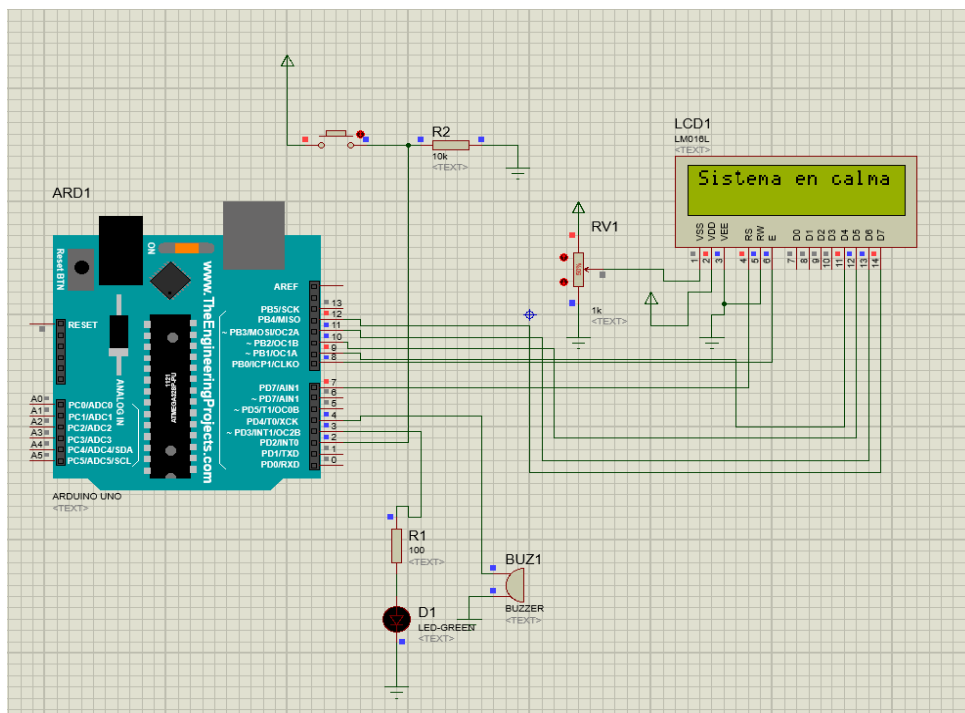
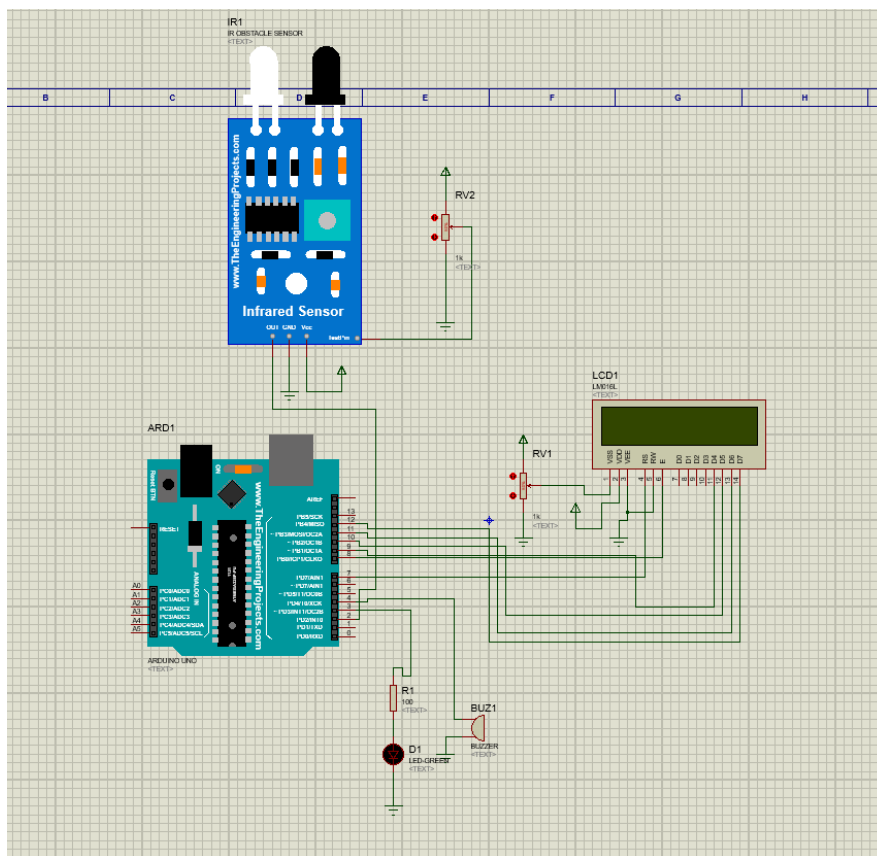
  if (motion == HIGH && pirState == LOW) {
    Serial.println("Movimiento detectado!");
    digitalWrite(ledPin, HIGH); // Enciende el LED
    pirState = HIGH;           // Actualiza el estado del PIR
  } else if (motion == LOW && pirState == HIGH) {
    Serial.println("No hay movimiento.");
    digitalWrite(ledPin, LOW); // Apaga el LED
    pirState = LOW;           // Actualiza el estado del PIR
  }
}
```





PROGRAMA

Construir una alarma con un sensor PIR pantalla LCD 2X16





Componentes necesarios:

1. Arduino Uno o similar.
2. Sensor PIR (como el HC-SR501).
3. Pantalla LCD 16x2.
4. Potenciómetro (opcional, para ajustar el contraste del LCD).
5. LED (para la indicación de alarma).
6. Resistor de 220 Ω (para el LED).
7. Zumbador (opcional, para alerta sonora).
8. Cables y protoboard.

Conexiones:

Sensor PIR:

- VCC \rightarrow 5V del Arduino.
- GND \rightarrow GND del Arduino.
- OUT \rightarrow Pin digital 2.

Pantalla LCD 16x2:

- RS \rightarrow Pin digital 7.
- EN \rightarrow Pin digital 8.
- D4 \rightarrow Pin digital 9.
- D5 \rightarrow Pin digital 10.
- D6 \rightarrow Pin digital 11.
- D7 \rightarrow Pin digital 12.
- VCC \rightarrow 5V.
- GND \rightarrow GND.
- Potenciómetro \rightarrow Ajustar contraste.

LED:

- Ánodo (lado largo) \rightarrow Pin digital 3 (a través de un resistor de 220 Ω).
- Cátodo \rightarrow GND.





Zumbador (opcional):

- Positivo → Pin digital 4.
- Negativo → GND.

PROGRAMA

```
#include <LiquidCrystal.h>
```

```
// Configuración de pines del LCD: RS, EN, D4, D5, D6, D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

```
// Pines del sistema
```

```
const int pirPin = 2;    // Sensor PIR
```

```
const int ledPin = 3;    // LED
```

```
const int buzzerPin = 4; // Zumbador (opcional)
```

```
// Variables
```

```
int pirState = LOW;      // Estado inicial del sensor PIR
```

```
int motionDetected = 0; // Variable para almacenar detección de movimiento
```

```
void setup() {
```

```
    pinMode(pirPin, INPUT);
```

```
    pinMode(ledPin, OUTPUT);
```

```
    pinMode(buzzerPin, OUTPUT);
```

```
// Inicializar la pantalla LCD
```

```
lcd.begin(16, 2);
```

```
lcd.print("Alarma activada");
```

```
delay(2000);
```

```
lcd.clear();
```





```
}  
  
void loop() {  
    // Leer el estado del sensor PIR  
    motionDetected = digitalRead(pirPin);  
    if (motionDetected == HIGH) {  
        // Movimiento detectado  
        digitalWrite(ledPin, HIGH); // Encender LED  
        digitalWrite(buzzerPin, HIGH); // Activar zumbador (opcional)  
  
        lcd.setCursor(0, 0);  
        lcd.print("MOVIMIENTO");  
        lcd.setCursor(0, 1);  
        lcd.print("DETECTADO!");  
        pirState = HIGH; // Actualizar estado  
    } else {  
        // No hay movimiento  
        digitalWrite(ledPin, LOW); // Apagar LED  
        digitalWrite(buzzerPin, LOW); // Apagar zumbador  
        lcd.setCursor(0, 0);  
        lcd.print("Sistema en calma");  
        lcd.setCursor(0, 1);  
        lcd.print(" "); // Limpiar segunda línea  
  
        pirState = LOW; // Actualizar estado  
    }  
  
    delay(200); // Pequeña pausa  
}
```

2.SISTEMA DE RIEGO CON ARDUINO





SENSOR DE HUMEDAD



El Sensor de humedad de Suelo FC-28 permite medir de forma sencilla la humedad del suelo por medio de 2 electrodos resistivos. Compatible con Arduino, PIC, ESP8266/NodeMCU/NodeMCU-32. El sensor es ideal para monitorear el nivel de humedad de tus plantas y así recordar cuando necesitan ser regadas o incluso para realizar un sistema totalmente automatizado de riego

añadiendo una válvula o una bomba de agua. Si el sistema se conecta a internet podríamos controlar/monitorear nuestro jardín desde cualquier lugar del mundo!

El funcionamiento del sensor se basa en medir la resistencia entre 2 electrodos insertados dentro del suelo, la resistencia entre los electrodos dependerá de la humedad del suelo, por lo que para un suelo muy húmedo tendremos una resistencia muy baja (corto circuito) y para un suelo muy seco la resistencia será muy alta (circuito abierto). El electrodo va conectado a una tarjeta de acondicionamiento (YL-38) que entrega una salida digital y otra analógica. La salida digital (DO) es la salida de un opamp en modo comparador, la salida digital se activa cuando el nivel de humedad es menor al nivel deseado, este nivel (umbral o threshold) se puede regular con el potenciómetro de la tarjeta. La salida analógica (AO) es la salida de un divisor de tensión entre una resistencia fija y la resistencia entre los electrodos, entrega un voltaje analógico desde 0V para un suelo muy húmedo hasta 5V para un suelo muy seco. Para la conexión a Arduino podemos optar por utilizar la salida analógica del módulo conectada a una entrada analógica del Arduino o por utilizar la salida digital (DO) conectada a una entrada digital del Arduino.

Recomendaciones: Introducir en la tierra solo la parte de los electrodos, los circuitos debe ser protegidos de la humedad. Para extender la vida útil del sensor se puede alimentar el sensor solo cuando va a ser leído y no permanentemente. Una alternativa de mayor vida útil es el Sensor de Humedad de Suelo Capacitivo.

Nota: La salida del sensor es de tipo cualitativo más que de tipo cuantitativo.

ESPECIFICACIONES TÉCNICAS

- Voltaje de alimentación: 3.3V - 5V DC (VCC)
- Corriente de operación: 35mA
- Voltaje de señal de salida analógico (AO) : 0 a VCC
- Voltaje de señal de salida digital (DO) : 3.3V/5V TTL

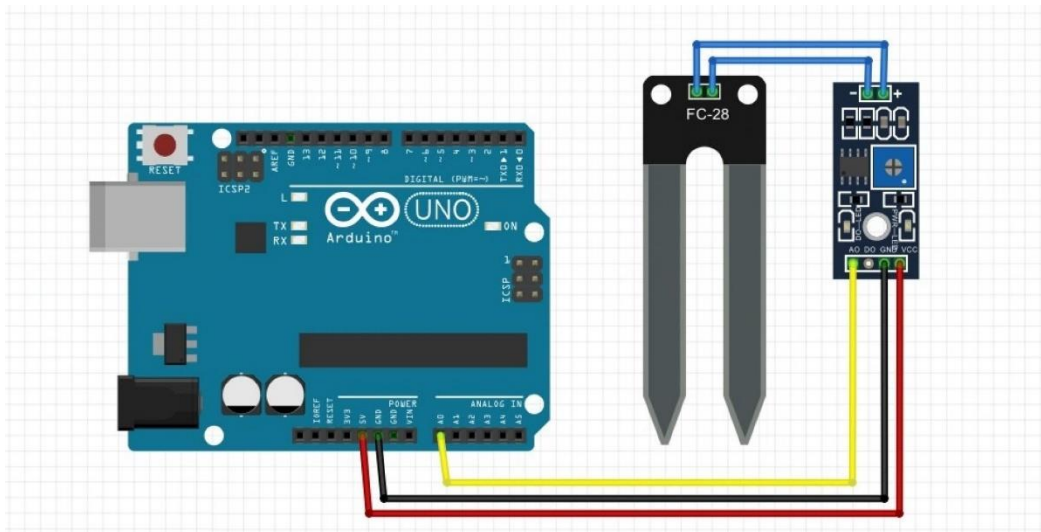




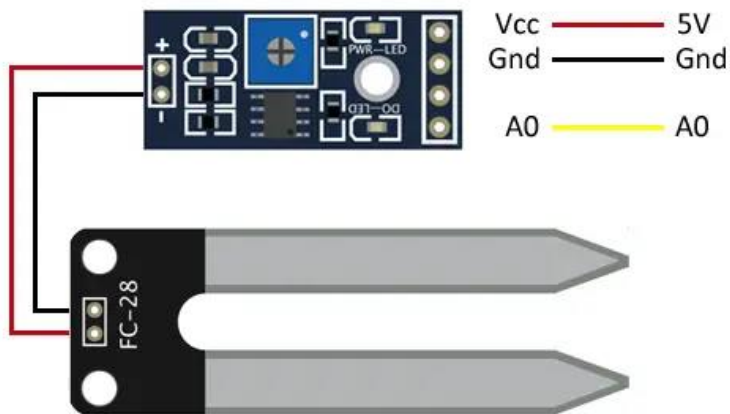
- Opamp LM393 en modo comparador, umbral (threshold) regulable por potenciómetro
- Superficie de electrodo: Estaño
- Incluye: Electrodo, Placa y cable de conexión
- Vida útil electrodo sumergido: 3 a 6 meses
- Dimensiones YL-38: 30*16 mm
- Dimensiones YL-69: 60*20*5 mm
- Peso:

CONEXIONES

- VCC: Voltaje de alimentación (3.3V - 5V DC)
- GND: Tierra (GND 0V)
- DO: Salida digital
- AO: Salida analógica



Sensor de Humedad del Suelo FC-28 para Arduino



El **FC-28** es un sensor diseñado para medir la humedad del suelo mediante la conductividad eléctrica. Es una herramienta ideal para proyectos de riego automático, monitoreo de cultivos y sistemas de domótica.

Componentes del Módulo FC-28

1. Sondas de humedad:
 - Dos sondas metálicas que se insertan en el suelo.
 - Detectan la conductividad del suelo (mayor en suelos húmedos, menor en suelos secos).
2. Módulo amplificador:
 - Procesa la señal de las sondas y proporciona salidas digital y analógica.
 - Incluye un potenciómetro para ajustar el umbral de sensibilidad de la salida digital.
3. Pines de conexión:
 - VCC: Alimentación (3.3V o 5V).
 - GND: Tierra.
 - AO: Salida analógica (valor proporcional a la humedad).
 - DO: Salida digital (HIGH o LOW según el nivel de humedad y el ajuste del potenciómetro).

Funcionamiento del Sensor

El sensor mide la conductividad eléctrica entre las dos sondas:

- Mayor humedad: Mayor conductividad → Mayor salida analógica.





- Menor humedad: Menor conductividad → Menor salida analógica.
- Conexión del FC-28 con Arduino
- Pines de Conexión

Pin del Sensor	Conexión en Arduino
VCC	5V
GND	GND
AO	Pin analógico (ej. A0)
DO	Pin digital (opcional)

Circuito Básico

1. Conecta el pin VCC del módulo al pin 5V de Arduino.
2. Conecta el pin GND del módulo al pin GND de Arduino.
3. Conecta el pin AO a un pin analógico (ejemplo: A0).
4. Conecta el pin DO a un pin digital si deseas usar la salida digital.

BOMBA DE AGUA



- Descripción

La mini bomba es una pequeña bomba de agua que puede ser usada en acuarios, piletas, caño, sistemas hidropónicos o en cualquier otro lugar donde pueda ser útil. Esta bomba funciona sumergida en el agua y debe conectarse una manguera de 6mm de diámetro interno. Muy útil para cualquier proyecto pequeño.

TIP: Si necesita colocarlo en el agua durante un período prolongado, selle el orificio del cable con pegamento para evitar que el agua invada.





Seguro para peces sin cobre expuesto. Fácil de limpiar sin necesidad de herramientas para desmontar y limpiar. Bajo nivel de ruido, bajo consumo de energía, tamaño pequeño y alta eficiencia.

#	Descripción	Valor
1	Corriente eléctrica	0.18A (promedio)
2	Ruido	40dB máximo
3	Fluido	Agua del grifo, agua subterránea, agua de mar
4	Temperatura de funcionamiento	-20°C – 50°C
5	Caudal	100L/H
6	Vida útil	Más de 200 horas
7	Tipo de motor	De cepillo importado
8	Voltaje nominal	3VDC – 5VDC
9	Potencia	0.36W
10	Altura de bombeo	0.3 – 0.8 metros
11	Dimensiones	Longitud: 43,5mm / Diámetro: 23,3mm
12	Dimensiones del conector	Longitud: 8mm / Diámetro: 7.4mm
13	Función del producto	Bombee agua desde un recipiente a través de una tubería.

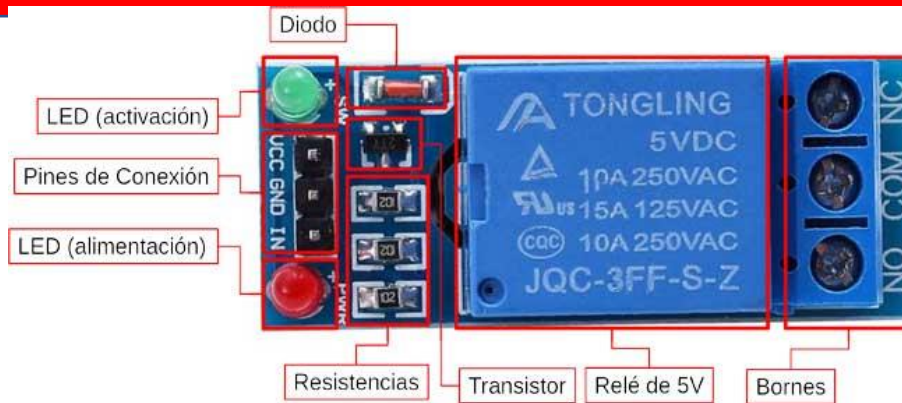
MODULO RELE

Módulo relé simple



Como su nombre lo indica es el módulo relé más simple que puedes encontrar para Arduino y ESP8266.

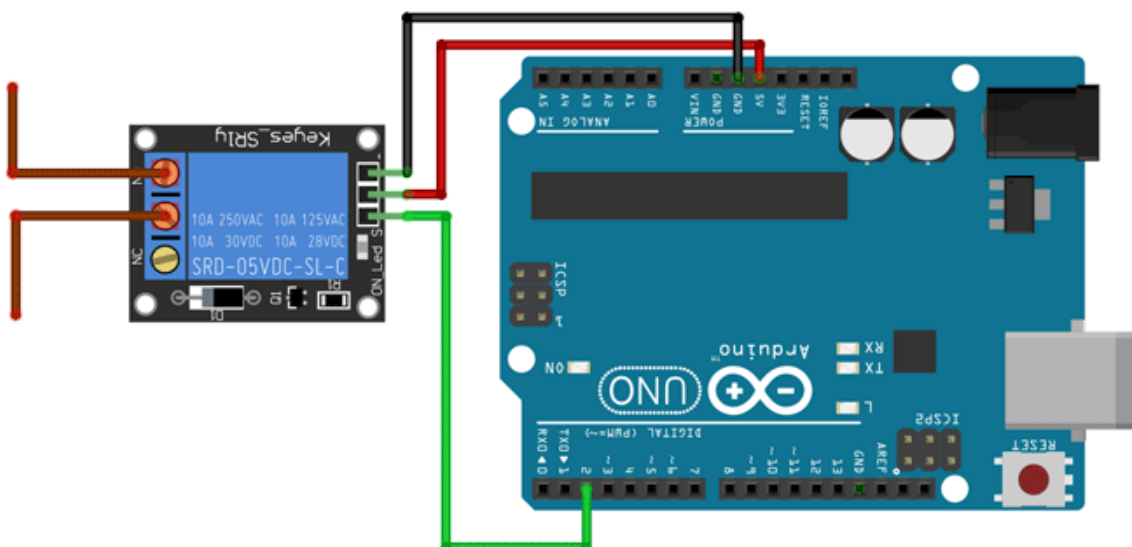




Los partes principales son:

- Relé: es donde está el interruptor y la parte de control.
- Pines de conexión: sirven para conectar el relé con Arduino o con otro microcontrolador como el ESP8266. Al final no son más que pines y cables que facilitan la conexión con la placa. Es la etapa de control.
- Bornes: se utilizan para conectar el dispositivo que se quiere controlar con el relé. Es la parte que va a la red eléctrica de casa por ejemplo y pertenece a la etapa de interruptor.
- Diodo, resistencias y transistor: facilitan el uso del relé con Arduino y ESP8266. Pertenece a la etapa de control.
- LED verde (activación): se enciende cuando el relé está activo es decir, la bobina está alimentada.
- LED rojo (alimentación): se enciende cuando el relé está alimentado.

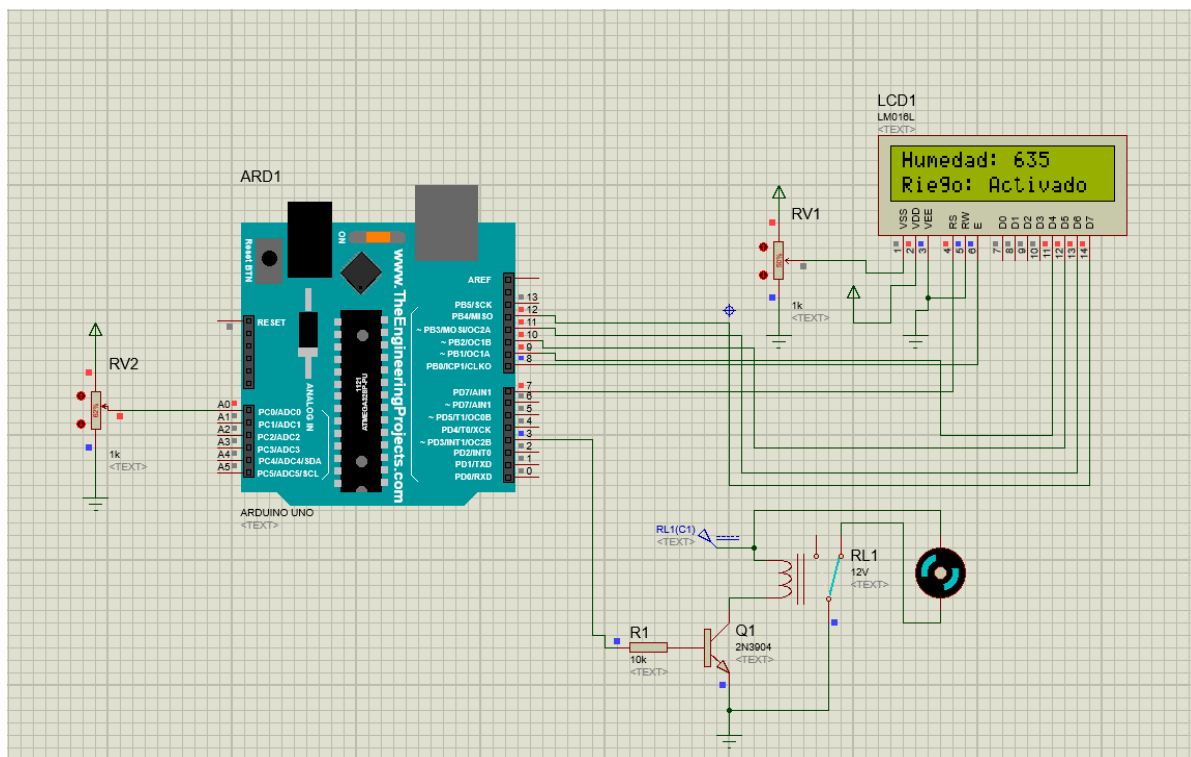
CONEXIÓN





10.2. Crear un sistema de riego automático

Uso de un sensor de humedad del suelo para activar una bomba de agua



Componentes necesarios:

1. Arduino Uno o similar.
2. Sensor de humedad del suelo (como el FC-28).
3. Pantalla LCD 16x2.
4. Relé para controlar la bomba.
5. Bomba de agua.
6. Potenciómetro (opcional, para el contraste del LCD).
7. Cables y protoboard.

Conexiones:

Sensor de humedad del suelo:

- VCC → 5V del Arduino.
- GND → GND del Arduino.
- OUT → Pin analógico A0.





Pantalla LCD 16x2:

- RS → Pin digital 7.
- EN → Pin digital 8.
- D4 → Pin digital 9.
- D5 → Pin digital 10.
- D6 → Pin digital 11.
- D7 → Pin digital 12.
- VCC → 5V.
- GND → GND.
- Potenciómetro → Ajustar contraste.

Relé y bomba:

- IN → Pin digital 3.
- VCC y GND → Fuente de alimentación del relé.
- Conectar la bomba de agua al relé según las especificaciones del fabricante.

PROGRAMA

```
#include <LiquidCrystal.h>

// Configuración de pines del LCD: RS, EN, D4, D5, D6, D7
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

// Pines del sistema

const int sensorPin = A0; // Sensor de humedad
const int relayPin = 3; // Relé para la bomba

// Umbrales de humedad (ajusta según tu sensor)
const int humedadBaja = 600; // Valor del sensor que indica suelo seco

void setup() {
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH); // Relé apagado inicialmente
```





```
// Inicializar LCD

lcd.begin(16, 2);

lcd.print("Sistema de Riego");

delay(2000);

lcd.clear();

}

void loop() {

    // Leer valor del sensor de humedad

    int humedad = analogRead(sensorPin);

    // Mostrar humedad en el LCD

    lcd.setCursor(0, 0);

    lcd.print("Humedad: ");

    lcd.print(humedad);

    // Controlar bomba de agua

    if (humedad > humedadBaja) {

        // Suelo seco, activar riego

        digitalWrite(relayPin, LOW); // Activar relé

        lcd.setCursor(0, 1);

        lcd.print("Riego: Activado");

    } else {

        // Suelo húmedo, desactivar riego

        digitalWrite(relayPin, HIGH); // Apagar relé

        lcd.setCursor(0, 1);

        lcd.print("Riego: Apagado ");

    }

    delay(1000); // Actualizar cada segundo

}
```

