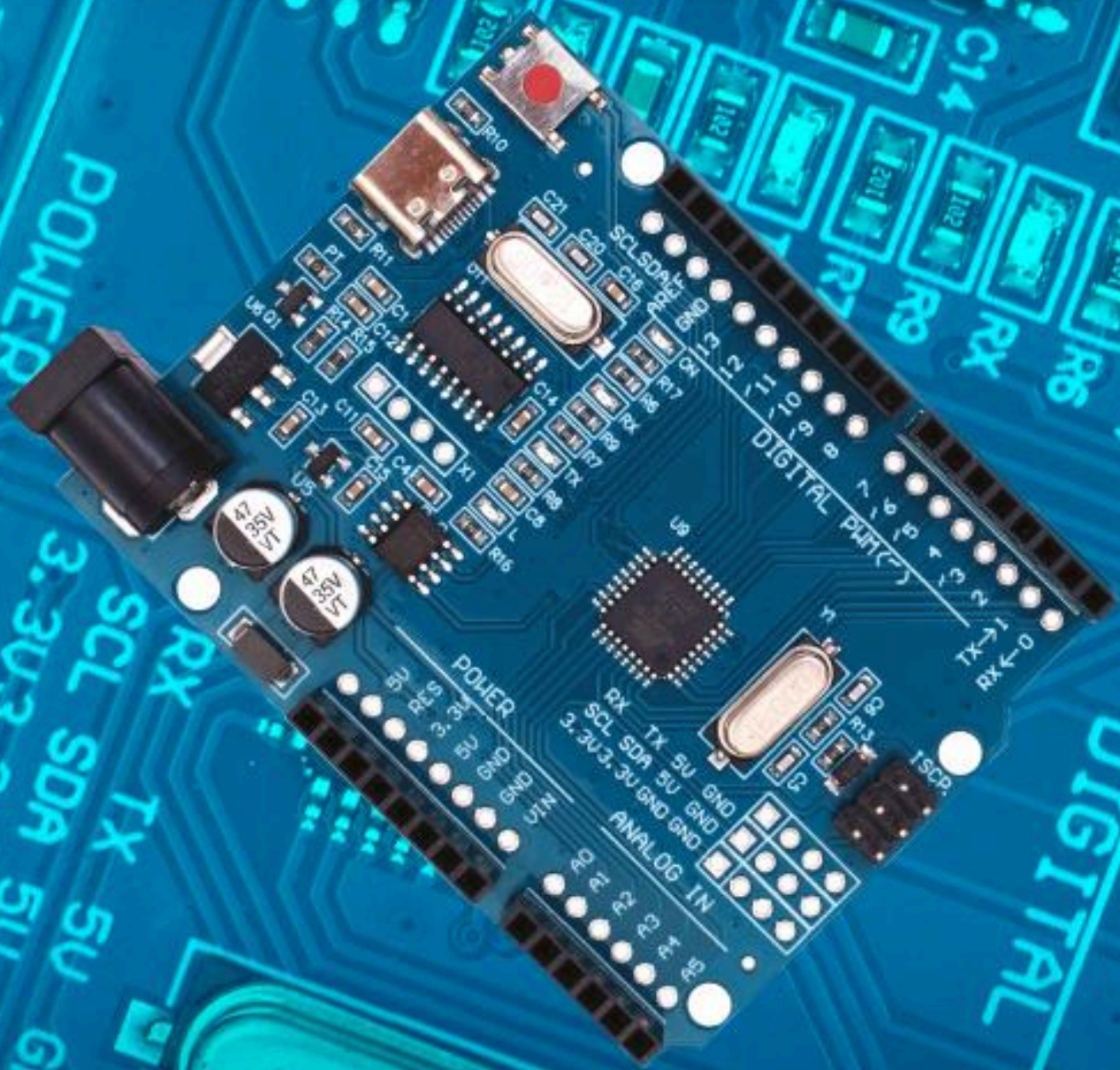


Starter KIT UNO R3

Tus primeros pasos en electrónica.

Manual de prácticas



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 1 LED Intermitente	3
Lección 2 RGB LED Intermitente	9
Lección 3 Control de un LED por un botón	14
Lección 4 Buzzer Activo	17
Lección 5 Buzzer Pasivo	20
Lección 6 Ocho Led Controlados con el 74HC595	24
Lección 7 Fotorresistor	28
Lección 8 74HC595 y Display 4 Dígitos 7 segmentos	33
Lección 9 Termómetro	37
Lección 10 Sensor de Inclinación	43
Lección 11 DHT11 Sensor de temperatura y humedad	46
Lección 12 Sensor de sonido	51
Lección 13 Sensor de humedad del suelo	55
Lección Bonus: Nivel de Agua	60

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 1 LED Intermitente

En esta lección, aprenderás a utilizar la tarjeta UNO R3 CH340G para que el LED parpadee en intervalos de tiempo random.

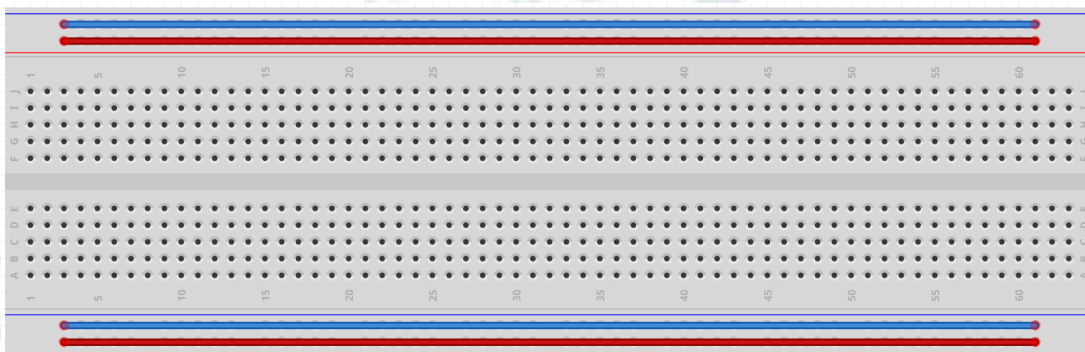
Materiales necesarios

- Tarjeta UNO R3 CH340G
- Led rojo 5mm
- Resistencia de 220 ohm
- Protoboard x1
- M-M Cables jumper

Conocimientos previos

Protoboard

Una placa de pruebas o también llamada protoboard es un pequeño tablero lleno de agujeros unidos electrónicamente y sirve para interconectar diferentes componentes, cuenta buses de alimentación y una serie de agujeros acomodados en filas (1 al 63) y columnas(A - J) con un espacio estándar de 2.54mm.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Starter KIT UNO R3

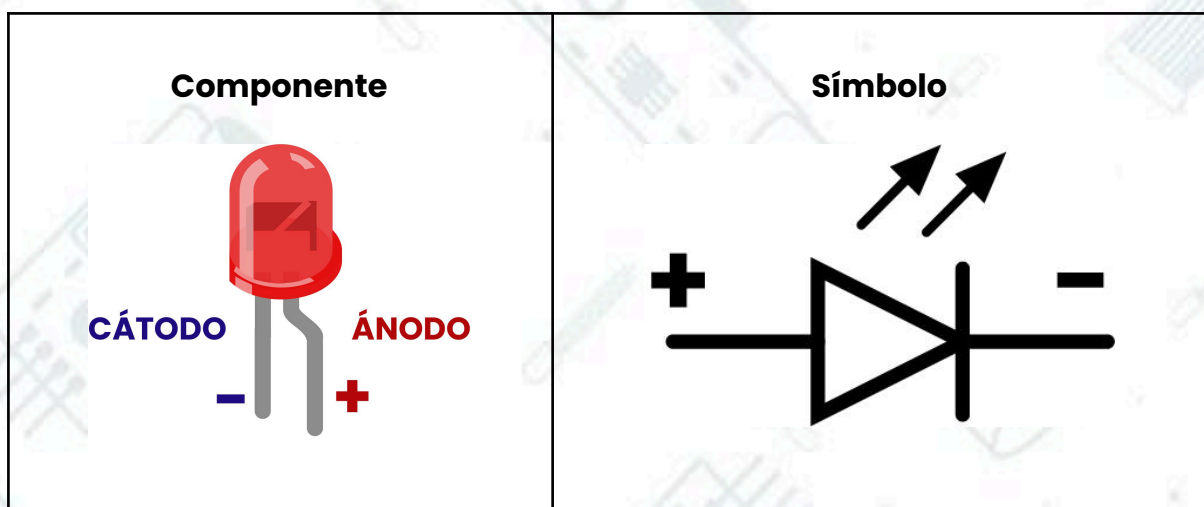
Tus primeros pasos en electrónica.

1. **Buses de alimentación:** Los buses se localizan en ambos extremos del protoboard, se representan por las líneas rojas (buses positivos o de voltaje) y azules (buses negativos o de tierra) no existe conexión física entre ellas. La fuente de alimentación generalmente se conecta aquí.
2. **Filas:** Los puntos que se encuentran consecutivos en la misma fila están interconectados entre sí, teniendo continuidad.
3. **Columnas:** Los puntos que se encuentran juntos de manera horizontal están aislados entre sí.

LED

Led (Light Emisor Diode) es un diodo semiconductor capaz de emitir luz cuando se le aplica tensión, es decir, convierte energía eléctrica en energía luminosa. Se debe conectar de forma correcta donde, la terminal más larga es positiva (+) y la terminal más corta es negativa (-); otra forma de diferenciar la terminal negativa, el led tiene un lado plano para indicar que esa es la terminal negativa.

Normalmente la tensión utilizada en los leds es de 1V a 3V, si la tensión es superada podrías dañar el led. Para evitar eso es conveniente usar una resistencia.

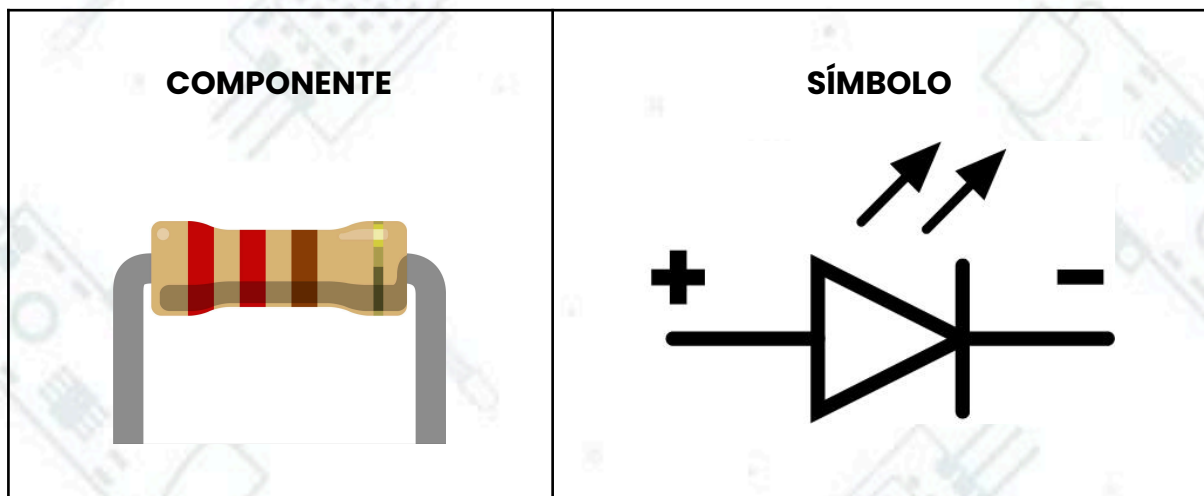


Starter KIT UNO R3

Tus primeros pasos en electrónica.

Resistencia

Una resistencia es un componente electrónico pasivo que sirve para limitar la corriente en un circuito eléctrico o ayudar al correcto funcionamiento de otros componentes electrónicos. La unidad de medida es Ohmio que se representa con la letra griega omega (Ω).

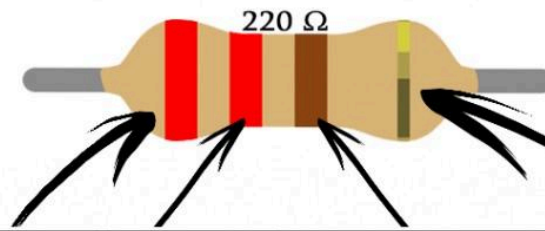


Las resistencias no tienen polaridad. Pueden ser conectadas de cualquier manera.

Las resistencias incorporan bandas de colores que indican cuál es su valor. Para saber el valor de una resistencia se utiliza el código de colores, que facilita la identificación y el valor de las resistencias de acuerdo al color que incorporan en sus bandas. En la siguiente imagen se muestra el código de colores:

Starter KIT UNO R3

Tus primeros pasos en electrónica.



Color	Banda 1	Banda 2	Banda 3 (multiplicadora)	Tolerancia
Negro	0	0	0 x1	
Café	1	1	1 x10	1%
Rojo	2	2	2 x100	2%
Naranja	3	3	3 x1000	
Amarillo	4	4	4 x10000	
Verde	5	5	5 x100000	0.5%
Azul	6	6	6 x1000000	0.25%
Morado	7	7	7 x10000000	0.10%
Gris	8	8	8 x100000000	0.05%
Blanco	9	9	9 x1000000000	

Dorado 5%

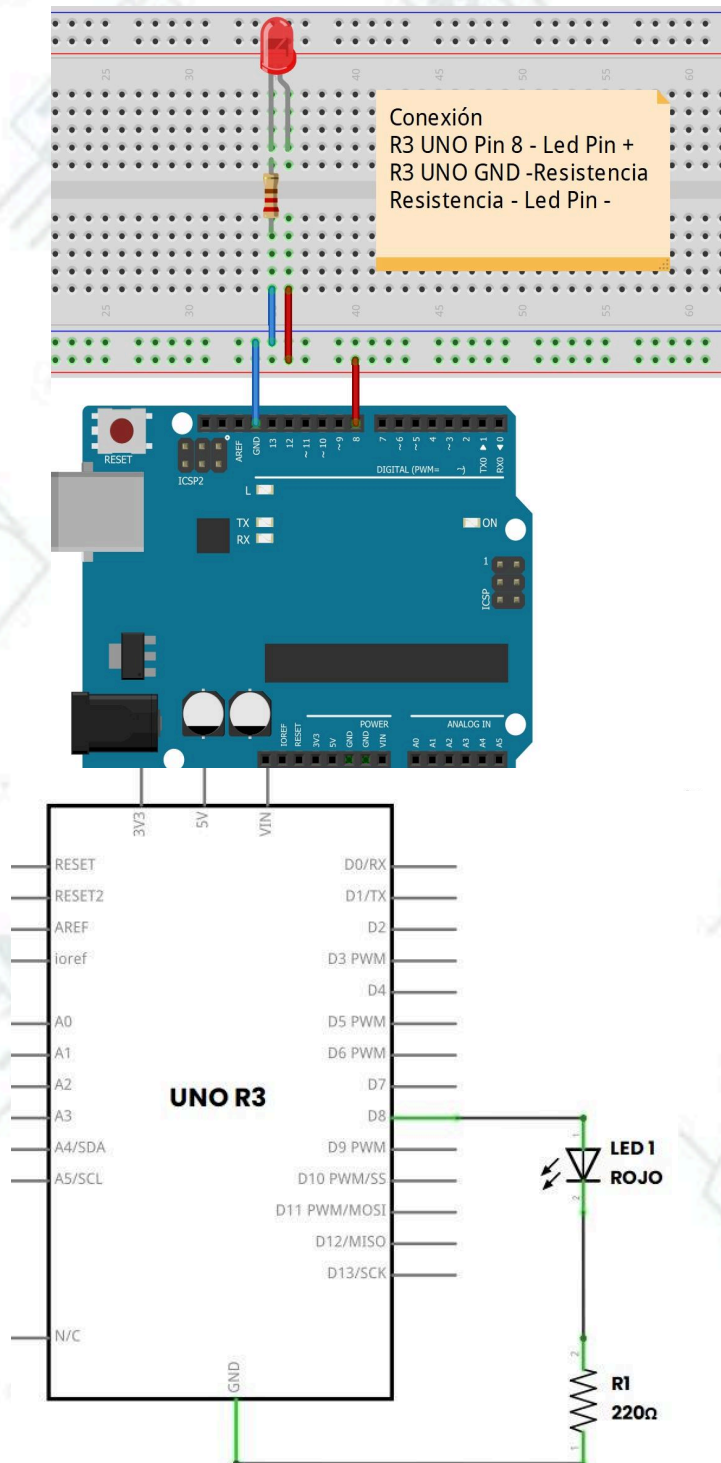
Plata 10%

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión

A continuación se muestran las conexiones entre la resistencia y led en la protoboard.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int led= 8; //digital pin2 del arduino
int ranNum; // se define la variable entero con nombre ranNum para
almacenar el valor random
void setup() {
  Serial.begin(9600);
  pinMode(led,OUTPUT); //definición de pin 2 como salida
}
void loop() {
  ranNum = random(0, 1000); //uso de la función random colocando un
rango de valores donde podrá variar de 0 a 1000
  //impresion en el monitor serial del valor que sera tomado para encender
el led en un tiempo de ms
  Serial.print("ON:");
  Serial.println(ranNum);
  //El led encenderá en el intervalo de ms que el valor ranNum defina
  digitalWrite (led,HIGH);
  delay(ranNum);
  //impresion en el monitor serial del valor que sera tomado para
encender el led en un tiempo de ms
  Serial.print("OFF:");
  Serial.println(ranNum);
  digitalWrite(led,LOW);
  //El led apagara en el intervalo de ms que el valor ranNum defina
  delay(ranNum);
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 2 RGB LED Intermitente

En esta lección, aprenderás a utilizar LED RGB con la tarjeta UNO R3 CH340G.

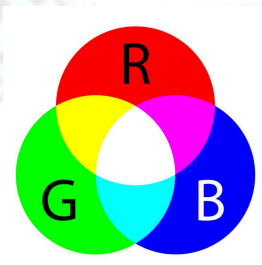
Materiales necesarios

- Tarjeta UNO R3 CH340G
- Led RGB
- Resistencias de 220 ohm
- Protoboard
- Cables jumper M-M

Conocimientos previos

LED RGB

RGB es un sistema de color basado en tres colores primarios Rojo, Verde y Azul (por sus siglas en inglés (R) Red, (G) Green, (B) Blue), el cual en base a la combinación de estos 3 colores podemos representar otros colores como se muestra a continuación.

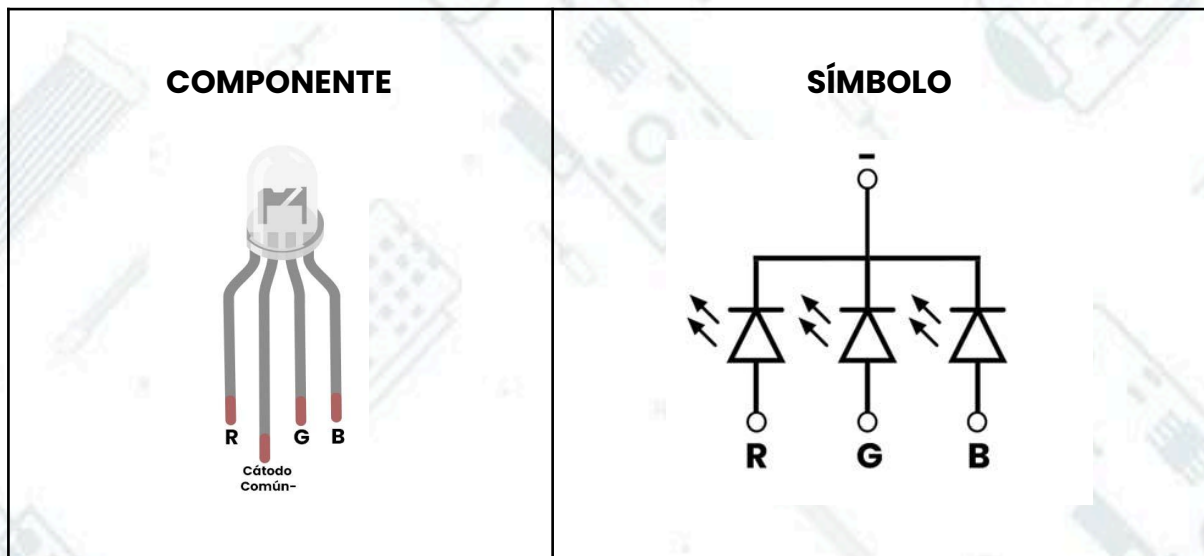


Por lo tanto, un Led RGB está formado internamente por tres diodos emisores de luz, uno rojo, uno verde y por último uno azul, con el propósito de poder crear, gran variedad de colores mediante la combinación de cada color con intensidades distintas.

Los Led's RGB tiene 4 terminales y es común utilizar el encapsulado de 5mm y hay 2 tipos de cátodo o ánodo común, que define cómo está conformado internamente el led RGB, (como se muestra a continuación), aunque en esta lección vamos a trabajar con un LED RGB de cátodo común.

Starter KIT UNO R3

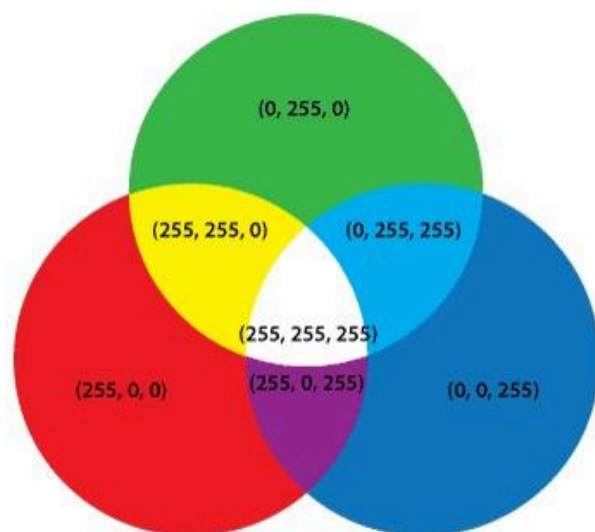
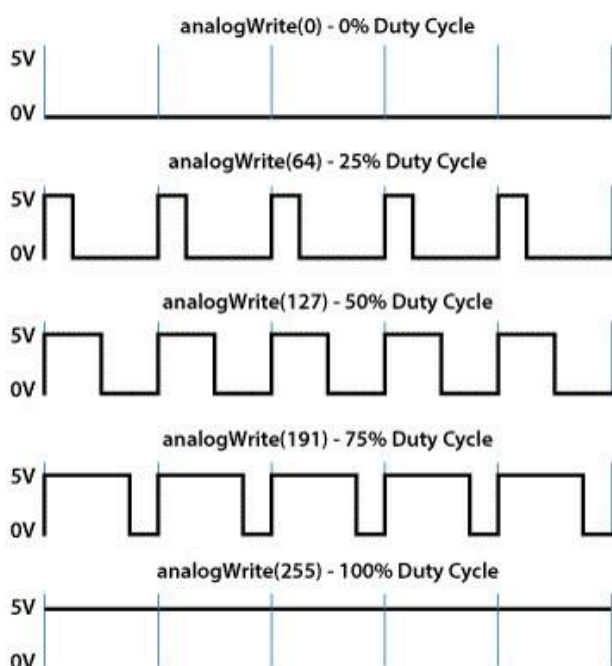
Tus primeros pasos en electrónica.



El control se recomienda que se haga por medio de PWM, de tal forma que, la resolución del PWM en cada color es de 8 bits, tendremos la posibilidad de generar más de 16 millones de colores distintos.

De esta manera el led RGB brillará con una intensidad de color determinada en base al voltaje que se le suministre a cada diodo led interno, pudiéndose representar cualquier color derivado de los primarios.

PWM - Pulse Width Modulation

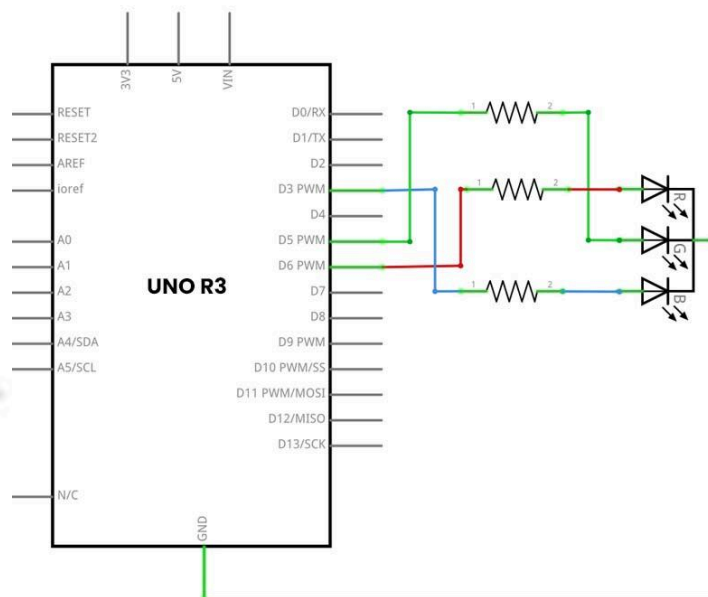
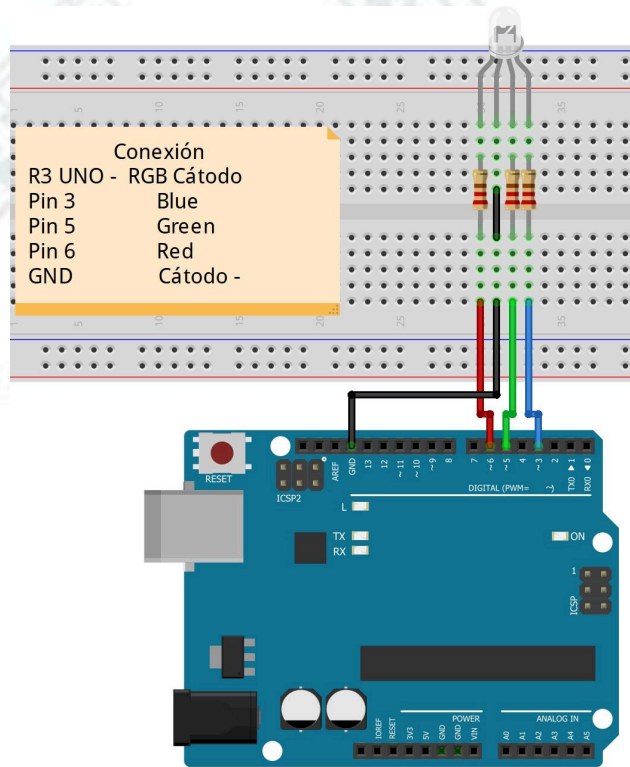


Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión

A continuación se muestran las conexiones entre la resistencia y led en la protoboard.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int rojoPin= 3; // El control del color Rojo por el pin 3
int verdePin = 5; //El control del color Verde por el pin 5
int azulPin = 6; //El control del color Azul por el pin 6
int ranNum; // se define la variable entero con nombre ranNum para
almacenar el valor random
// Se definen como salidas a los pines 3, 5, 6; correspondientes al color del
RGB
void setup() {
  pinMode(rojoPin, OUTPUT);
  pinMode(verdePin, OUTPUT);
  pinMode(azulPin, OUTPUT);
}
void loop() {
  // Valor del ciclo de trabajo PWM (0 a 255), que nos ayudará a tener
control de la intensidad de brillo por cada uno de los colores (rojo, verde,
azul), con ayuda de la función "setColor"
  ranNum = random(0, 1000); //uso de la función random colocando un
rango de valores donde podrá variar de 0 a 1000

  setColor(255, 0, 0); //Brillará en color Rojo
  delay(ranNum); //Instrucción que durará entre 0 a 1000ms, dependiendo
el valor random
  setColor(0, 255, 0); //Brillará en color Verde
  delay(ranNum); //Instrucción que durará entre 0 a 1000ms, dependiendo
el valor random
  setColor(0, 0, 255); //Brillará en color Azul
  delay(ranNum); //Instrucción que durará entre 0 a 1000ms, dependiendo
el valor random
  //Si colocamos el PWM de todos los colores en máximo
  setColor(255, 255, 255); //Brillara en Blanco
  delay(ranNum); //Instrucción que durará entre 0 a 1000ms, dependiendo
el valor random
```

//Si realizamos combinación de los valores podremos obtener más gama de colores

```
setColor(255, 0, 255); // Brillará en color Morado
```

```
delay(ranNum); //Instrucción que durará entre 0 a 1000ms, dependiendo el valor random
```

```
}
```

```
//Función "setColor()" tendremos acceso al control de los pines PWM
```

```
//mostrando a la salida los colores que previamente indicamos
```

```
//por medio del argumento (rojo, verde, azul); que serán valores de entra:
```

```
//rojoValue, verdeValue and azulValue.
```

```
void setColor(int rojoValue, int verdeValue, int azulValue) {
```

```
  analogWrite(rojoPin, rojoValue);
```

```
  analogWrite(verdePin, verdeValue);
```

```
  analogWrite(azulPin, azulValue);
```

```
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 3 Control de un LED por un botón

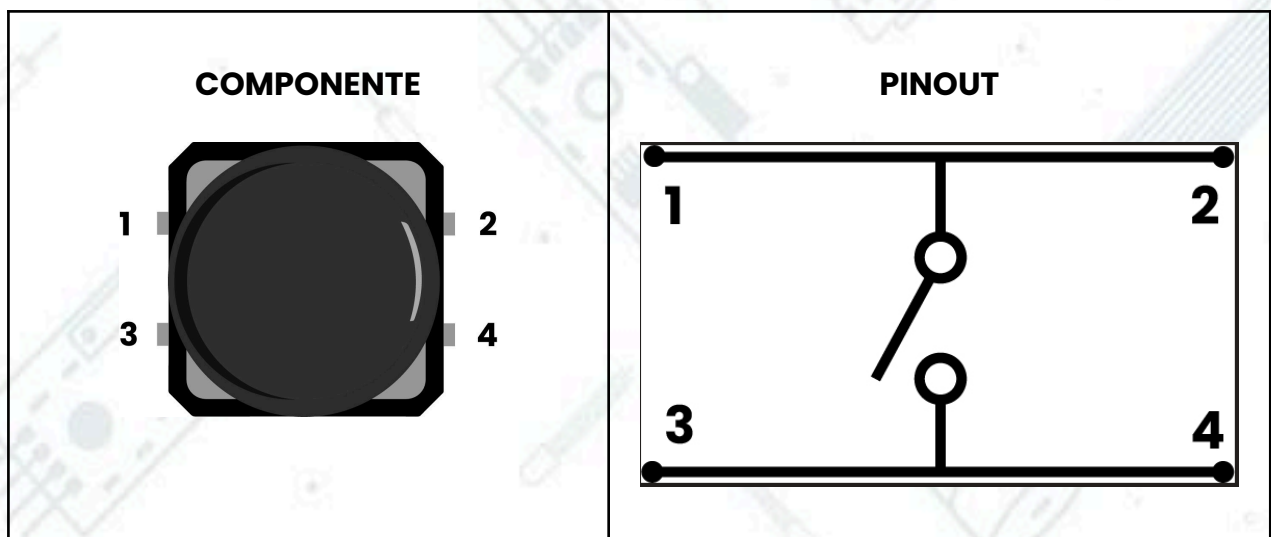
En esta lección, aprenderás cómo utilizar pulsadores con entradas digitales en la tarjeta UNO R3 CH340G para tener control de encendido y apagado de un led

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard (Placa de pruebas)
- Led de 5mm rojo
- Resistencia de 220 Ohms
- Push Button 4 pines
- Cables M- M

Conocimientos previos

Push Botton 4 pines es un dispositivo táctil que sirve como interruptor ya que puede ser activado, al ser pulsado con el dedo y permiten el flujo de corriente mientras es accionado.

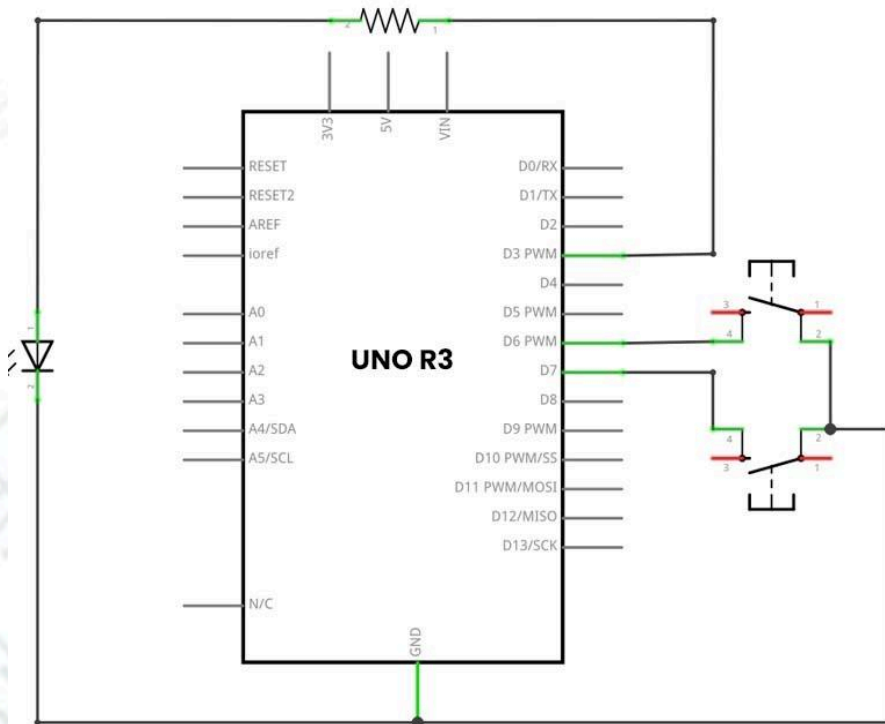
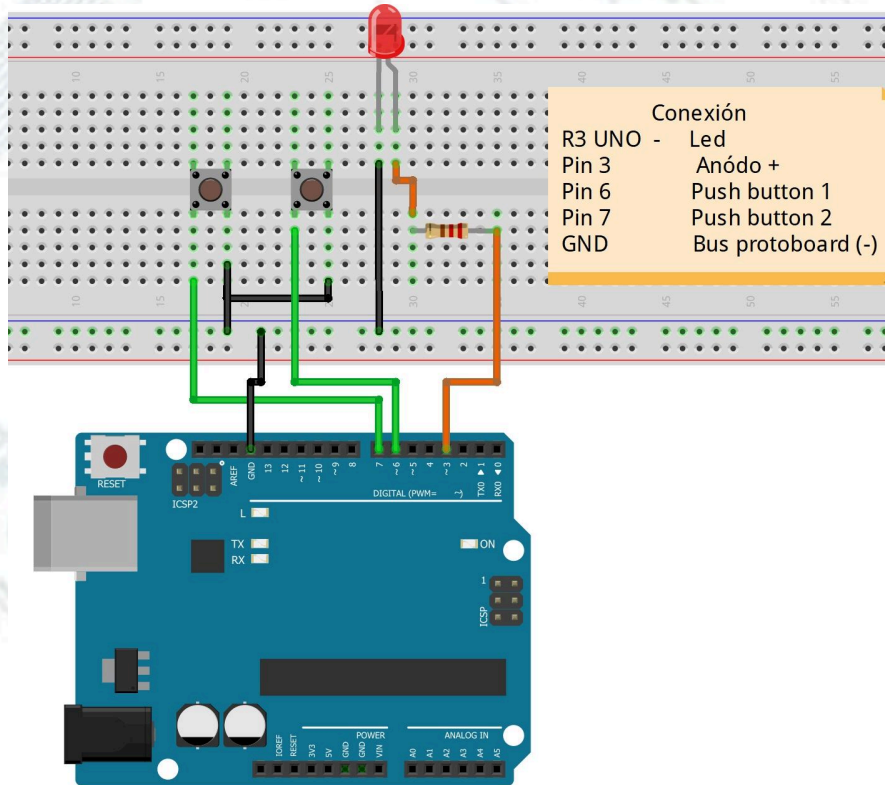


Para realizar el circuito en el protoboard se utilizará 2 botones, 1 para encender el led y otro para apagarlo, también se utilizará una resistencia de 220 Ohms para proteger al led.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de Conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int ledPin = 3; // Asignación del Pin 10 para el Led

int onApin = 6; //Asignación del Pin 9 para el push button que nos ayudará
a encender el Led

int offBpin = 7; //Asignación del Pin 8 para el push button que nos ayudará
a apagar el Led

//Se declara una variable leds para controlar en estado, actual del led
(on/off)Tipo byte (números que representan 8 bits)

byte leds = 0;

void setup(){

    pinMode(ledPin, OUTPUT); //Declaración del pin de salida a ledPin / pin 10

    //Declaración de los push button en modo PULL UP () y así conectar
    directamente al Arduino sin apoyo de una resistencia física

    pinMode(onApin, INPUT_PULLUP);

    pinMode(offBpin, INPUT_PULLUP); }

void loop(){

    //Realizaremos lectura de la entrada digital del pin 9 / pin 8 y realizando
    acciones dependiendo el estado en que se encuentre

    if (digitalRead(onApin) == LOW) {

        //Si se presiona el botón del pin 9 que se encuentra en nivel
        //Alto pasará a nivel Bajo y por consiguiente...

        digitalWrite(ledPin, HIGH); } //El led se encenderá

    if (digitalRead(offBpin) == LOW) //Si se presiona el botón del pin 8
    { digitalWrite(ledPin, LOW); } //El led se apagará
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 4 Buzzer Activo

En esta lección aprenderemos a utilizar un buzzer activo para generar un tono mediante un pin digital de la tarjeta UNO R3 CH340G.

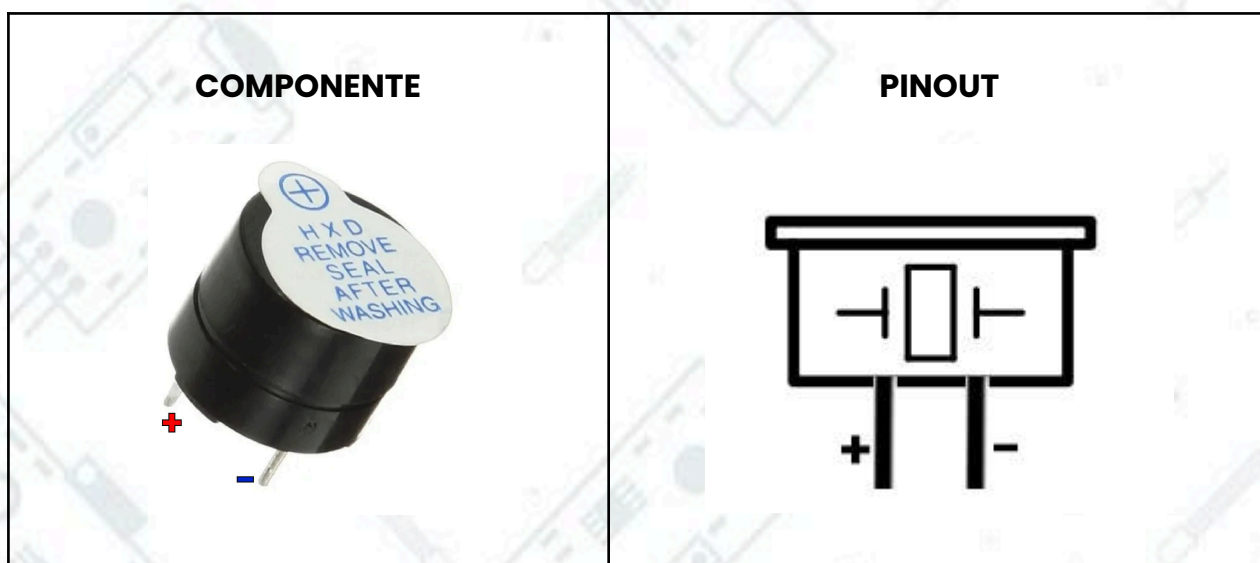
Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard
- Zumbador activo
- Cables jumper M-M

Conocimientos previos

Buzzer activo es un pequeño dispositivo que puede producir un sonido cuando se aplica una corriente directa DC. Por ejemplo, si se conecta directamente a los 5V de la tarjeta UNO R3, este genera un tono fijo con una frecuencia ya predeterminada debido a que incorpora un oscilador interno.

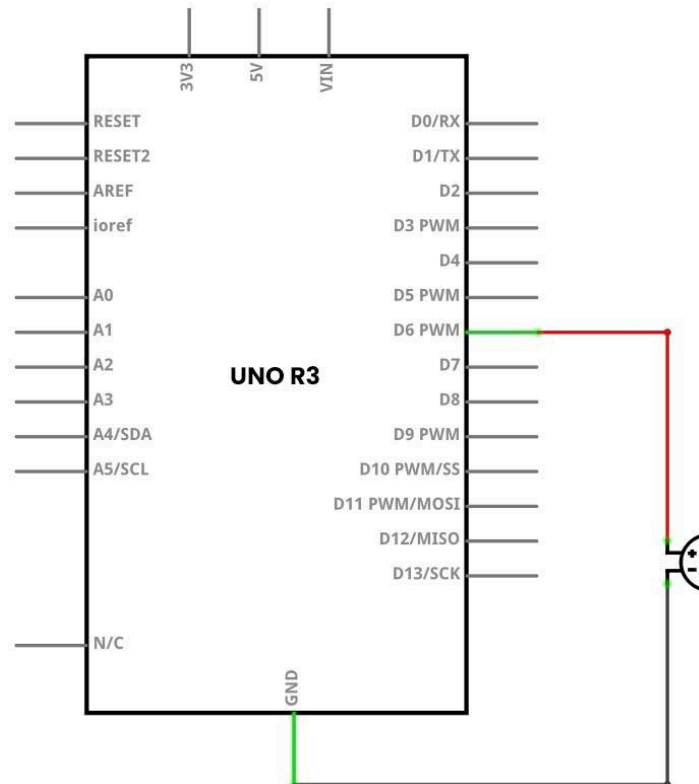
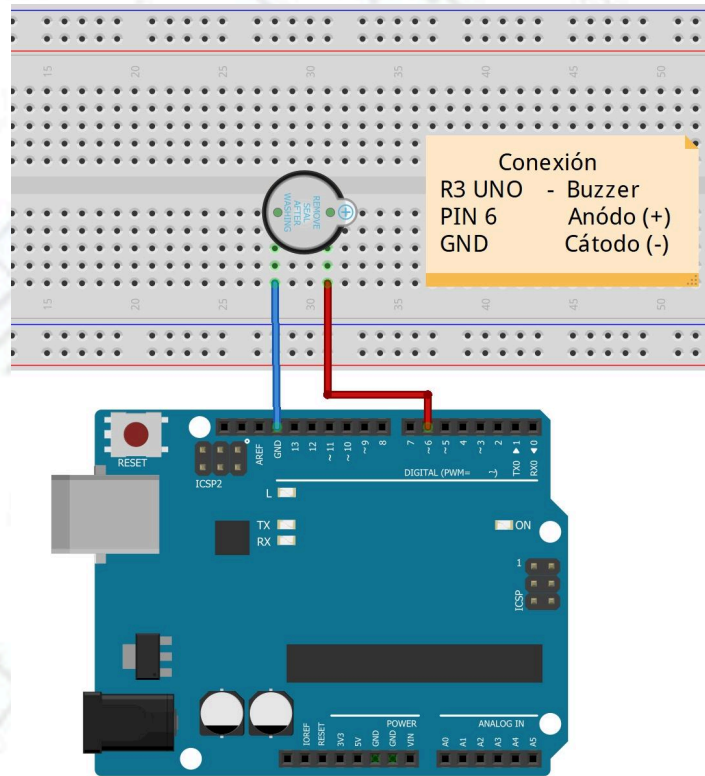
El buzzer activo tiene 2 terminales, negativo y positivo, comúnmente incorporan una etiqueta que nos indica cual es la terminal positiva mediante el símbolo (+), la terminal que se encuentre más cerca a esta referencia será la terminal positiva.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
//Encendido y control de tono por medio de la tarjeta Uno R3

int buzzer = 6; // Pin 12 para el buzzer activo

void setup()
{
    pinMode(buzzer,OUTPUT); //Declaración de pin de salida asignado al buzzer

    Serial.begin(9600);
}

void loop(){
    unsigned char i;//Variable tipo carácter sin signo llamada "i" para simular una frecuencia

    for(i=0;i<80;i++){ //Generación de frecuencia de pitidos

        digitalWrite(buzzer,HIGH); //El buzzer emitirá un pitido...

        delay(10);//...por 50ms, variando este tiempo puedes cambiar la frecuencia del pitido

        digitalWrite(buzzer,LOW); //El buzzer se apagará...

        delay(10); //...por 50ms

    }
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 5 Buzzer Pasivo

En esta lección aprenderemos a utilizar el buzzer pasivo y la tarjeta UNO R3 CH340G para generar ocho sonidos diferentes, cada sonido durará 1 segundo desde Do (262Hz), Re (294Hz), Mi (330Hz), Fa (349Hz), Sol (392Hz), La (440Hz), Si (494Hz) hasta Do Agudo (524Hz).

Material necesarios

- Tarjeta UNO R3 CH340G
- Protoboard
- Zumbador pasivo
- Cables jumper M-M

Conocimientos previos

Un buzzer pasivo es un dispositivo electromecánico que produce sonido cuando se le aplica una señal eléctrica alterna. A diferencia de un buzzer activo (como el que se utilizó en la práctica anterior), que tiene un oscilador interno y solo necesita una tensión continua para funcionar, el buzzer pasivo requiere una señal de frecuencia, la cual, generamos por medio de la tarjeta UNO R3.

Es más versátil en términos de control de tono y frecuencia, pero necesita un circuito adicional para generar la señal adecuada. Se usa comúnmente en alarmas, timbres y sistemas de notificación.

Para cambiar la frecuencia del tono y generar diferentes melodías se recomienda aplicar una señal de onda cuadrada con una frecuencia específica, para este caso se recomienda utilizar en programación la función `Tone()` la cual permitirá definir 3 parámetros, el pin del buzzer la frecuencia y la duración de la nota.

El buzzer pasivo tiene 2 terminales, negativo y positivo, para identificar la terminal positiva el buzzer comúnmente incorpora en la parte superior una referencia con el símbolo (+), la terminal que se encuentre más cerca a esta referencia será la terminal positiva.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

COMPONENTE



PINOUT

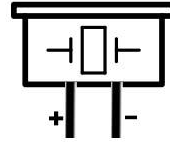
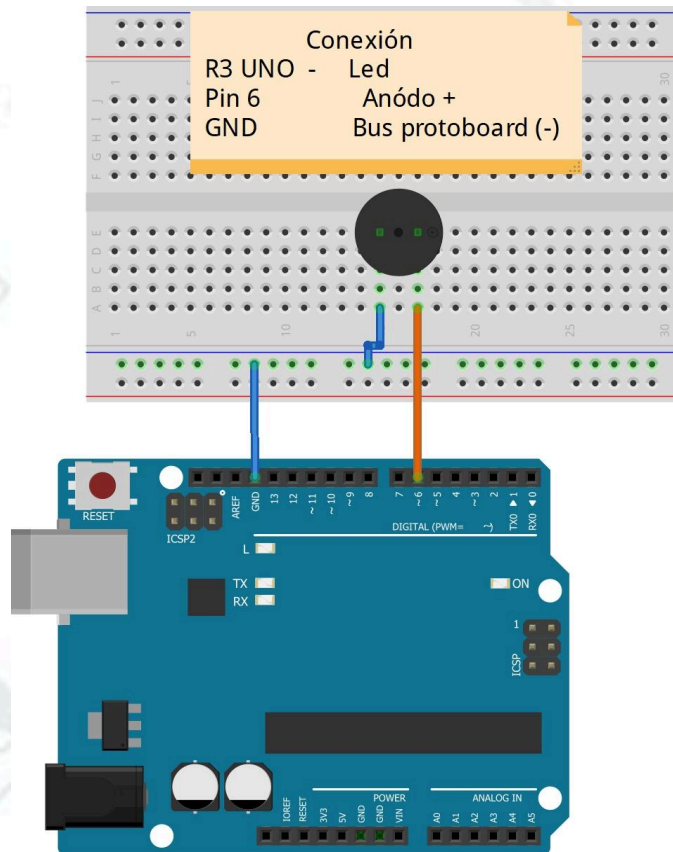
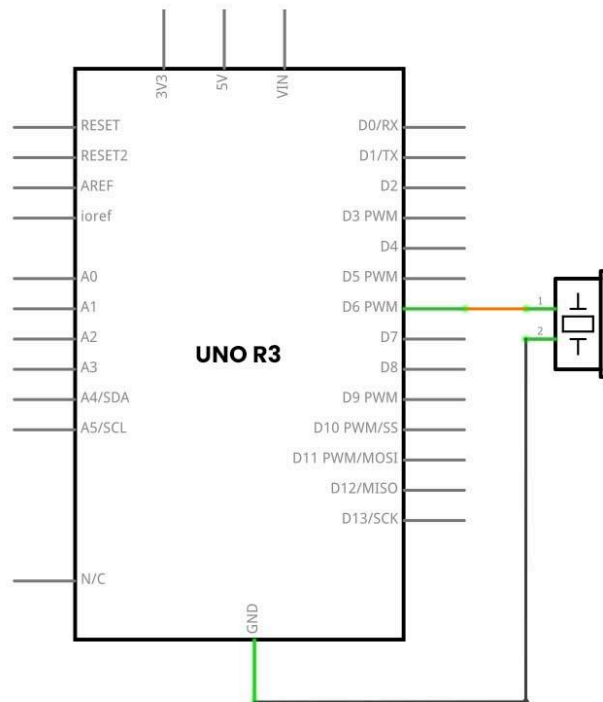


Diagrama de conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Código de funcionamiento

```
int Do = 262, Re = 294, Mi = 330, Fa = 349, Sol = 392, La = 440, Si = 494, Do2 = 524; // valor de las frecuencia de cada una de las notas de la escala principal de Do
int buzz = 6; // Pin 6 para control del buzzer pasivo
int inicio = 0; // Variable para control de tiempo inicializada con valor 0
void setup()
{
  pinMode(buzz, OUTPUT); // Inicialización del pin 12 como salida
}
void loop()
{
  inicio = 500; // Duración de cada una de las notas será de 500ms / .5 segundos
  tone(buzz, Do, inicio); // Generación de onda cuadrada pin 6, con valor de DO
  delay(1000); // Espera 1s
```

```
tone(buzz, Re, inicio);//Generación de onda cuadrada pin 6, con valor de
RE
delay(1000);           //Espera 1s
tone(buzz, Mi, inicio);//Generación de onda cuadrada pin 6, con valor de
MI
delay(1000);           //Espera 1s
tone(buzz, Fa, inicio);//Generación de onda cuadrada pin 6, con valor de
FA
delay(1000);           //Espera 1s
tone(buzz, Sol, inicio);//Generación de onda cuadrada pin 6, con valor de
SOL
delay(1000);           //Espera 1s
tone(buzz, La, inicio);//Generación de onda cuadrada pin 6, con valor de
LA
delay(1000);           //Espera 1s
tone(buzz, Si, inicio);//Generación de onda cuadrada pin 6, con valor de SI
y una
delay(1000);           // Espera 1s
tone(buzz, Do2, inicio);//Generación de onda cuadrada pin 6, con valor de
DO sostenido
delay(1000);           //Espera 1s
noTone(buzz);          //Terminara la reproducción de sonido
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 6 Ocho Led Controlados con el 74HC595

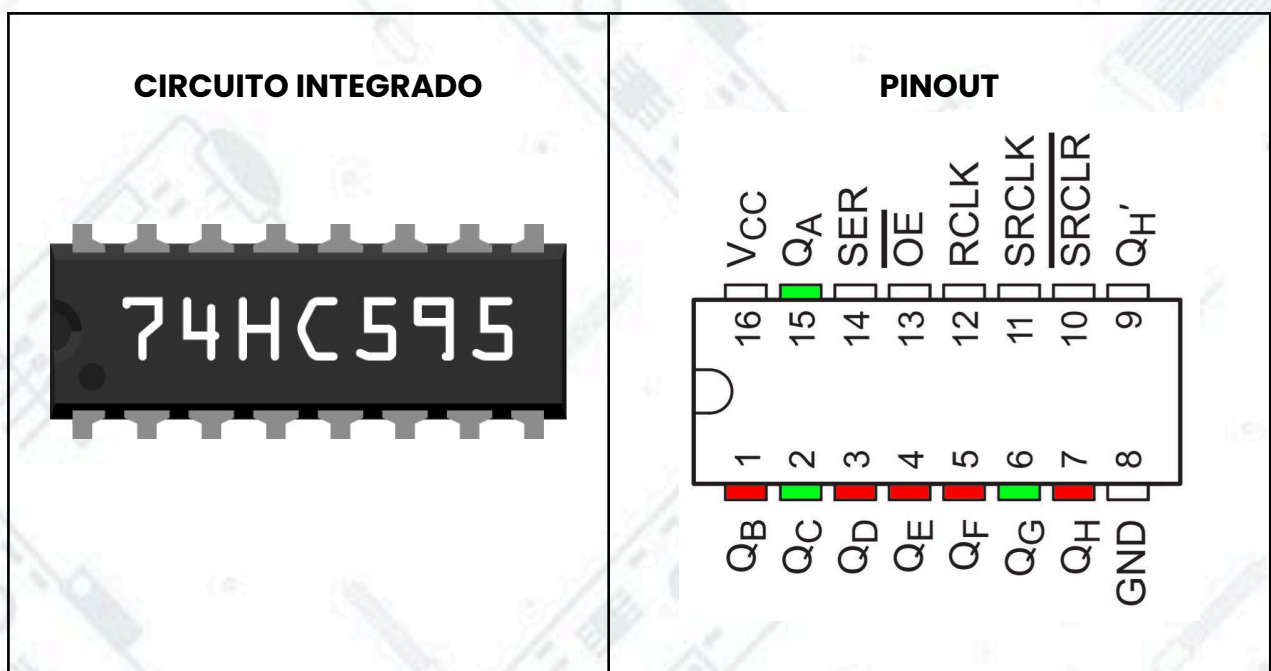
En esta lección aprenderemos a utilizar el circuito 74HC595 para encender 8 led mediante la utilización de 3 pines de la tarjeta UNO R3 CH340G. Ya que el CI 74HC595 es un registro de desplazamiento que permite ampliar el número de pines de E/S de la tarjeta tarjeta UNO R3 CH340G.

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard (Placa de pruebas)
- Led de 5mm
- Resistencia de 220 Ohms
- IC 74HC595
- Cables M- M

Conocimientos previos

El IC 74HC595 o SN74HC595N es un registro de desplazamiento de 8 bits con salida en serie y paralelo . Permite controlar múltiples salidas (como LEDs o displays) utilizando solo unos pocos pines de un microcontrolador.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Funciona recibiendo datos en serie a través de un pin de entrada (DS) y desplazándose internamente mediante pulsos de reloj (SHCP). Una vez cargados los 8 bits, se transfieren a los pines de salida paralela mediante un pulso en el pin de almacenamiento (STCP).

También tiene una salida en serie (Q7') para conectar varios ICs en cascada y ampliar el número de salidas. Es ampliamente usado en proyectos de electrónica para ahorrar pines de control.

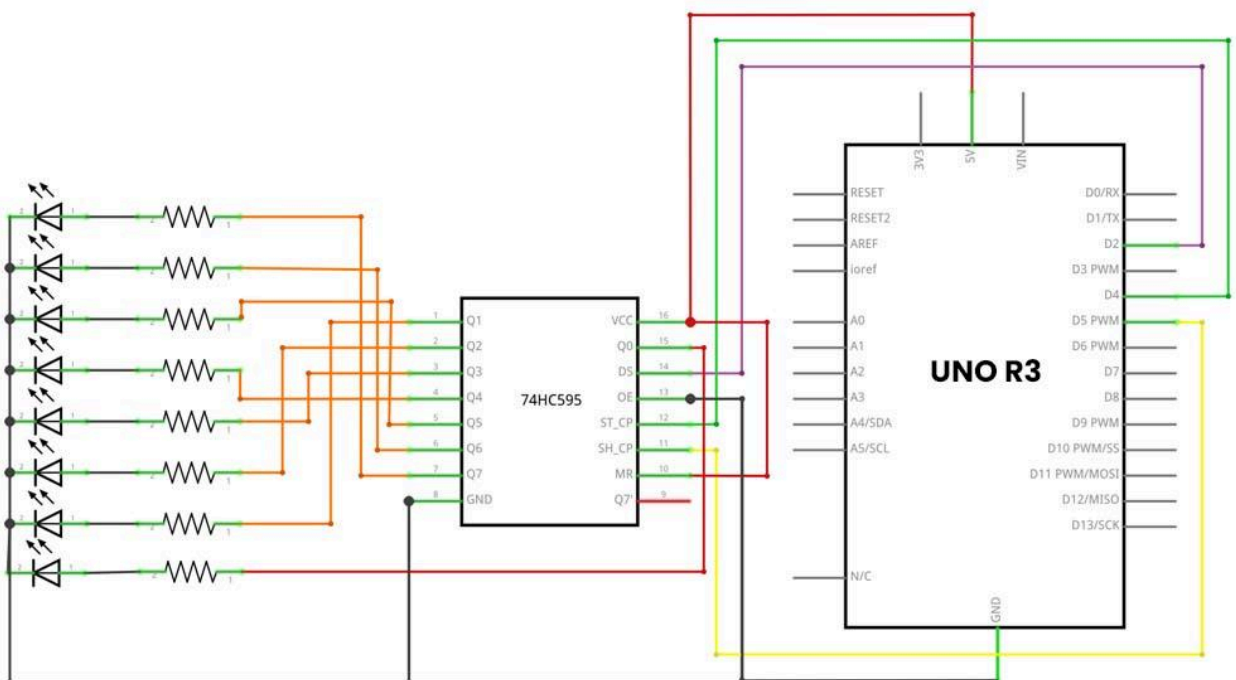
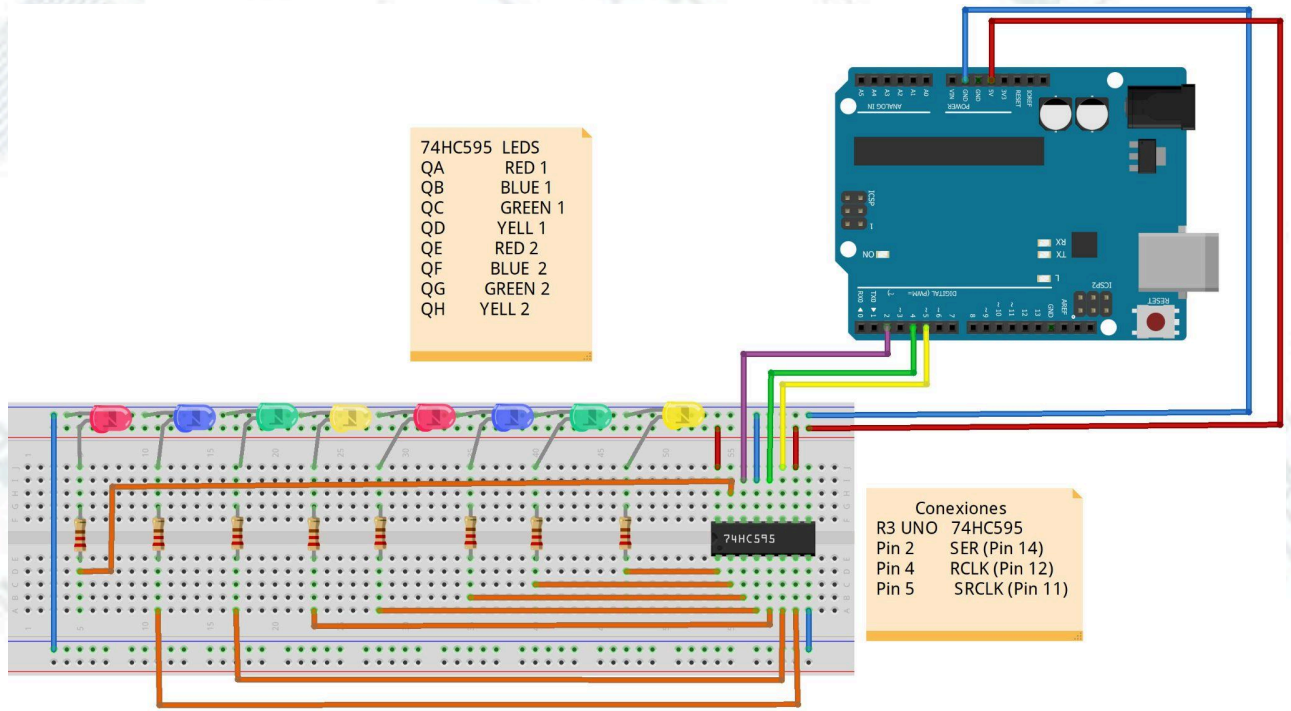
El chip contiene ocho pines que podemos utilizar como salida, cada uno de ellos está asociado con un bit en el registro. En el caso del CI 74HC595, nos referimos a ellos desde QA hasta QH.

Para escribir estas salidas con la tarjeta UNO R3, tenemos que enviar un valor binario al registro de desplazamiento y con ese número el registro de desplazamiento puede comprender qué salidas utilizar.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int RCLKPin = 4; //Usaremos el Pin 4 de la tarjeta UNO R3 para el manejo del
Reloj de Registro (Pin 12 RCLK del 74HCC595).
int SRCLKPin = 5; //Usaremos el Pin 5 de la tarjeta UNO R3 para el manejo
del Reloj de Registro de desplazamiento (Pin 11 SRCLK del 74HCC595).
int datoPin = 2; //Usaremos el Pin 2 de la tarjeta UNO R3 para entrada
Serial del Dato (Pin 14 SER del 74HCC595).
byte leds = 0; //Variable llamada leds con espacio de 8 bits, donde se
guardará el patrón de encendido/apagado de leds
int nextLED = 0; //Variable para detectar el byte en la variable leds
void setup() {
    //Declaración de variables, como salidas digitales
    pinMode(RCLKPin, OUTPUT);
    pinMode(datoPin, OUTPUT);
    pinMode(SRCLKPin, OUTPUT);
    leds = 0; } //Inicialización del programa y de la variable tipo byte en 0
void loop() {
    leds = 0; //Limpieza del bucle de la variable byte en 0, para que el led
encienda una vez
    if (nextLED == 7) //Se comienza el corrimiento de los bits 0, si se llega a 7
    {
        nextLED = 0; //Reiniciamos el avance en 0
    } else {
        nextLED++; //De lo contrario seguirá avanzando hasta llegar a 8 bits
    }
    bitSet(leds, nextLED); //Usará la función bitSet para el control de c/led
dependiendo el avance del bit
    digitalWrite(RCLKPin, LOW); //Pone en nivel bajo a latchPin durante la
transmisión
    shiftOut(datoPin, SRCLKPin, MSBFIRST, leds);
    digitalWrite(RCLKPin, HIGH); //Devuelve el latchpin a nivel alto
    delay(1000); } //Un segundo de tiempo para esta rutina
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 7 Fotorresistor

En esta lección aprenderás a usar una fotorresistencia, se utilizará el circuito de la lección anterior, con la diferencia de que se modulara la intensidad de luz de los LED por medio de la fotorresistencia utilizando de una entrada analógica de la tarjeta UNO R3 CH340G

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard
- Led de 5mm
- Resistencia de 220 Ohms
- IC 74HC595
- Fotorresistencia LDR
- Resistencia de 1k
- Cables M – M

Conocimientos previos

Una fotorresistencia o LDR (por sus siglas en inglés "light-dependent resistor") es un componente electrónico cuya resistencia varía en función de la luz. Se trata de un sensor que actúa como una resistencia variable en función de la luz que capta. A mayor intensidad de luz, menor resistencia.

Para conocer la cantidad de luz que el sensor capta en cierto ambiente, necesitamos medir la tensión de salida del mismo. Para lograr esto se debe realizar un divisor de tensión colocando una resistencia en serie. Hay dos formas básicas para conectar una LDR:

1. Mayor luz, mayor voltaje: Al conectar la fotorresistencia al nodo positivo de nuestra fuente de voltaje tendremos que, al incidir una mayor cantidad de luz provocará una menor caída de voltaje o diferencia de potencial entre la fuente y el pin de referencia (V_{out}), por lo tanto, se tendrá una lectura mayor.

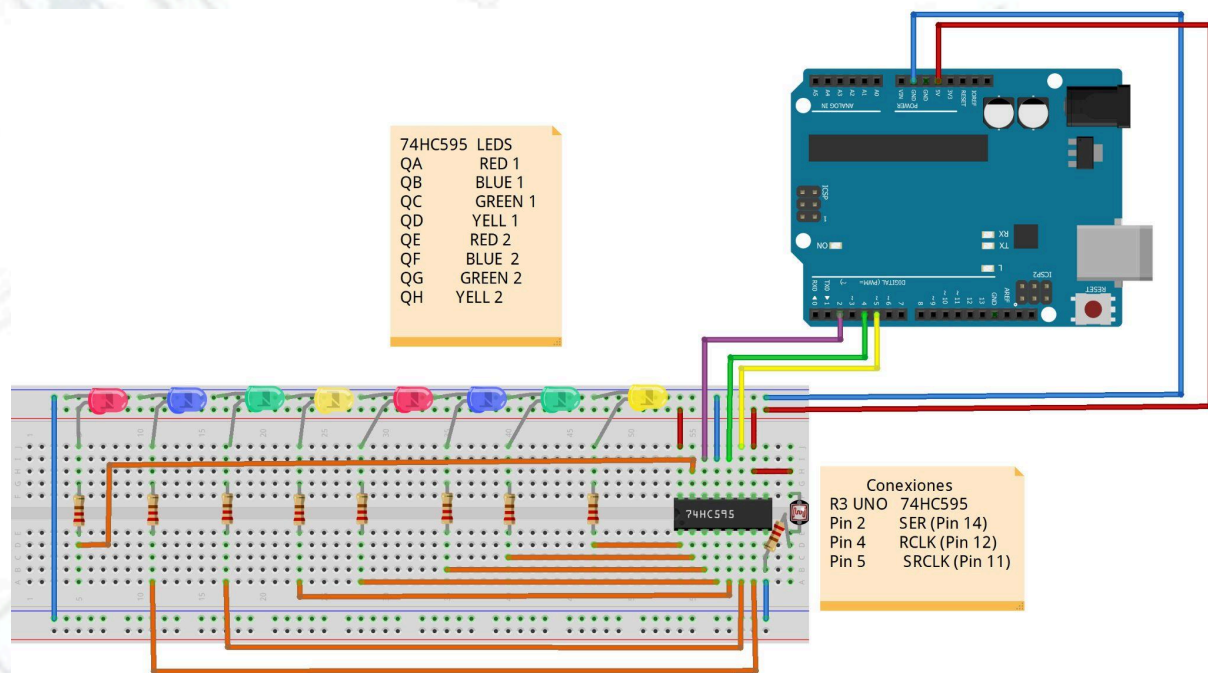
Starter KIT UNO R3

Tus primeros pasos en electrónica.

2. Mayor luz, menor voltaje: En pocas palabras la fotorresistencia se conecta al nodo de GND y provocará un comportamiento opuesto al punto 1.

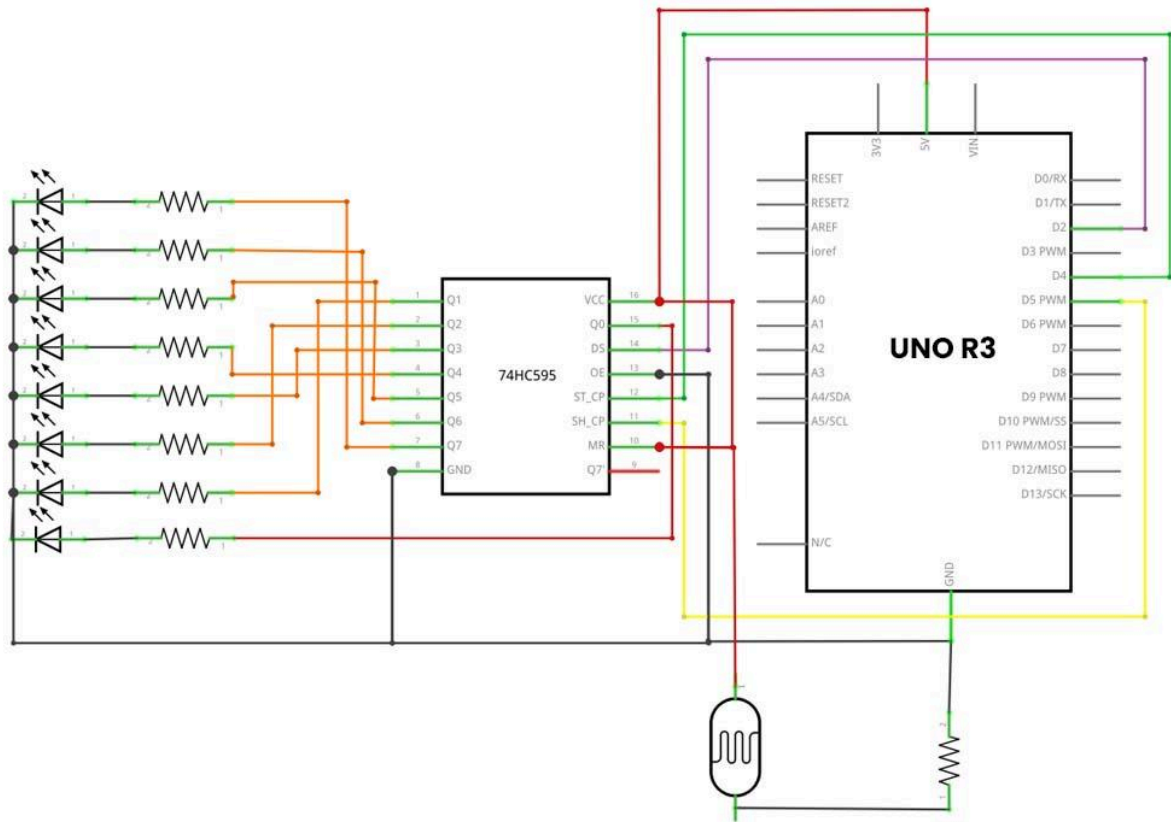


Diagrama de conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int lightPin = 0; // Pin analógico A0 de la tarjeta UNO R3 para conectar la LDR
int latchPin = 4; // Usaremos el Pin 11 para el manejo del Reloj de Registro (Pin 12 RCLK del 74HCC595).
int clockPin = 5; // Usaremos el Pin 12 para el manejo del Reloj de Registro de Desplazamiento (Pin 11 SRCLK del 74HCC595)
int dataPin = 2; // Usaremos el Pin 9 para entrada Serial del Dato (Pin 14 SER del //74HCC595)
int leds = 0; // Variable entera para el registro de leds
void setup()
{
    // Declaración de variables, como de salida digital
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    Serial.begin(9600);
}
// El valor que se obtiene la fotorresistencia es analógico entre 0 a 1023 0 (oscuro) y 1023 (luz)
void loop()
{
    int reading = analogRead(lightPin); // Lectura del valor analógico del fotorresistor
    Serial.println(reading);
    int numLEDSLit = reading / 10; // 1023 entre 8 (8 bites) / 2 = ~63
    numLEDSLit = 8 - numLEDSLit; // Para cambiar el sentido de cambio de encendido en los leds
    Serial.println(numLEDSLit);
    if (numLEDSLit > 8)
        numLEDSLit = 8; // Si el valor del led es mayor que 8, el valor será igual a 8
    leds = 0; // ...no habrá iluminación en los leds
    for (int i = 0; i < numLEDSLit; i++) // tendremos un incremento de la variable i, hasta que sea menor a numLEDSLit
```

```
{
  leds = leds + (1 << i);      // Establece el i-décimo bit
}
actualizaRegistro();      //Se ejecuta la función
}
void actualizaRegistro()    //Función para realizar el encendido de los
leds a partir del Reloj de Registro
{
  digitalWrite(latchPin, LOW); //Pone en nivel bajo a latchPin , led
apagado
  shiftOut(dataPin, clockPin, MSBFIRST, leds); //función shiftOut para
desplazar el bit,
  digitalWrite(latchPin, HIGH); //pone en nivel bajo a latchPin ,
led encendido
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 8 74HC595 y Display 4 Dígitos 7 segmentos

En esta lección aprenderás a utilizar el display 4 dígitos 7 segmentos de ánodo común con el registro de desplazamiento 74HC595.

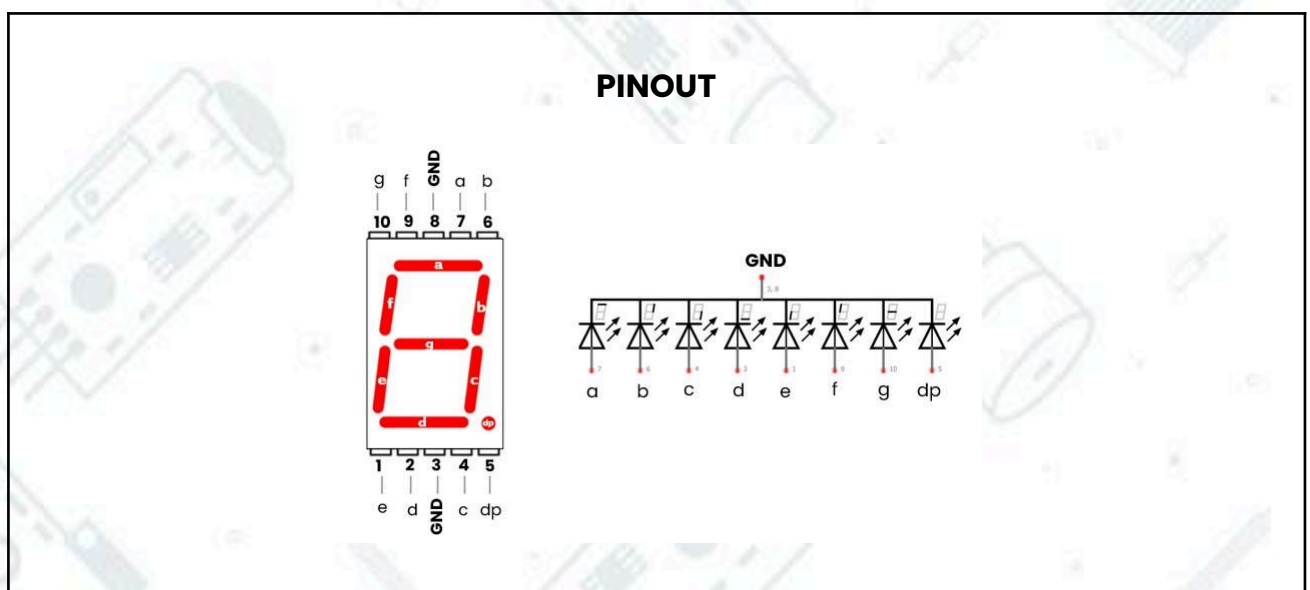
Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard
- Resistencia de 220 Ohms
- IC 74HC595
- Display 4 Dígitos 7 Segmentos cátodo común
- Cables M- M

Conocimientos previos

El display 4 dígitos 7 Segmentos es un dispositivo opto-electrónico que permite visualizar números del 0 al 9. Se utiliza para representar visualmente números y algunos caracteres. El display está compuesto por 4 dígitos, cada dígito tiene 7 segmentos y un punto decimal que pueden encender o apagar individualmente.

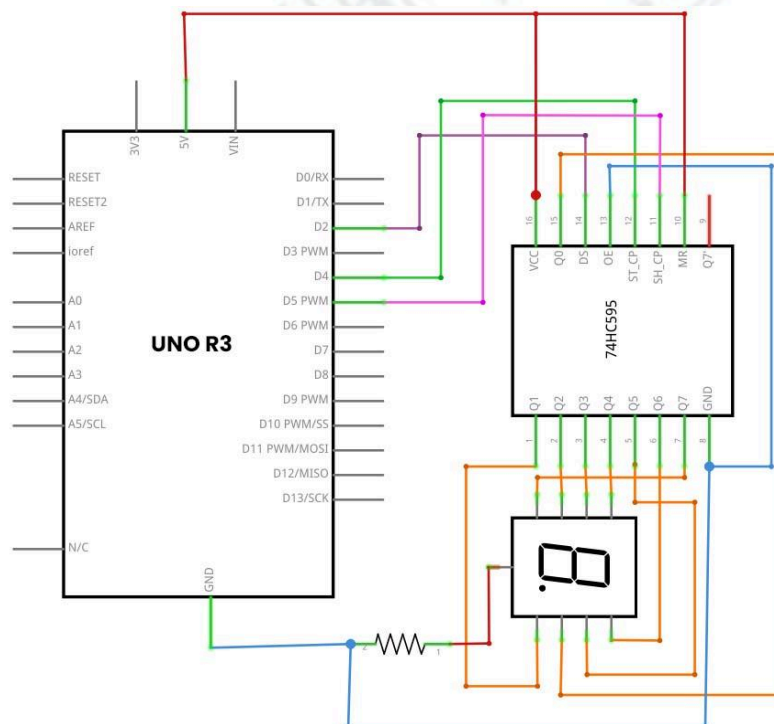
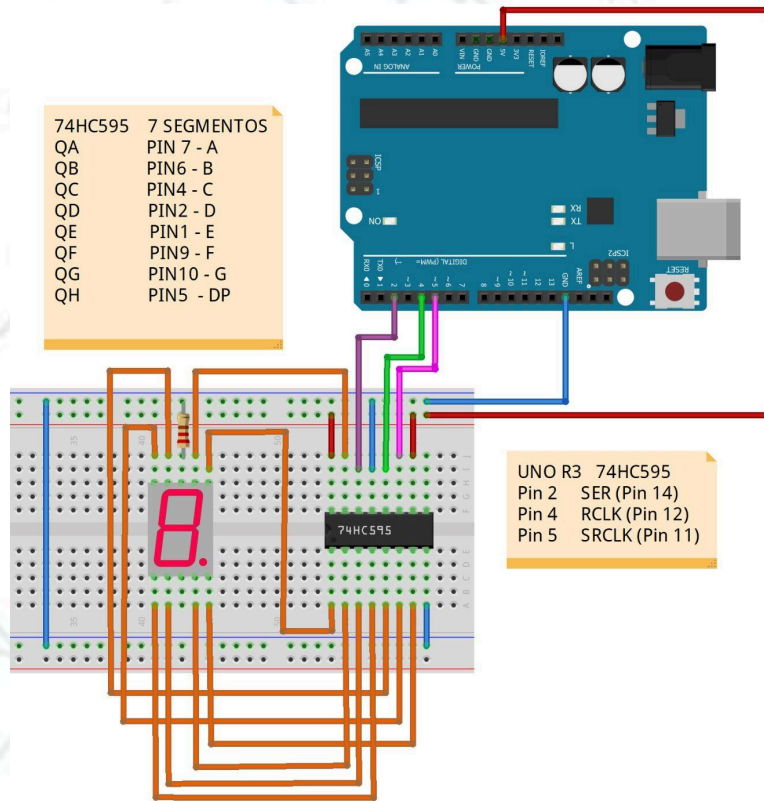
En la siguiente imagen se muestra los pines del display 4 dígitos 7 Segmentos:



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

```
int latchPin = 4; // Pin conectado al Pin 12 del 74HC595 (Latch)
int clockPin = 5; // Pin conectado al Pin 11 del 74HC595 (Clock)
int dataPin = 2; // Pin conectado al Pin 14 del 74HC595 (Data)
int i = 0; //Variable para realizar conteo
//Combinación de bit por led del display de 7 segmentos realizando una
//secuencia desde 0 a la letra C: A,B,C,D,E,F,G,PUNTO DECIMAL
const byte numeros[14] = {
  B1111101, // = 0.
  B01100000, // = 1
  B11011010, // = 2
  B11110010, // = 3
  B01100110, // = 4
  B10110110, // = 5
  B10111110, // = 6
  B11100000, // = 7
  B11111110, // = 8
  B11100110, // = 9
  B11101110, // = A
  B00111110, // = B
  B10011101, // = C
};
void setup() {
  Serial.begin(9600);
  //Declaración de variables, como de salida digital
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}
void loop() {
  for (i = 0; i < 14; i++) { //repetición de la secuencia de encendido y
  apagado de byte del 1 al 14 anteriormente declarada en la función
  números
```

```
digitalWrite(latchPin, LOW); //desactiva a Reloj de Registro de  
Desplazamiento función shiftOut que nos ayudará a controlar al 74HC para  
desplazar el bit, indicando por que pin sale el bit (dataPin), cambio del pin  
una vez que dataPin tenga el valor adecuado (clockPin), dirección de  
desplazamiento (LSBFIRST o MSBFIRST) los datos que se  
desplazaran (leds) byte
```

```
shiftOut(dataPin, clockPin, LSBFIRST, numeros[i]);  
    digitalWrite(latchPin, HIGH); //activado el Reloj de Registro de  
Desplazamiento  
    delay(1000); //espera de 1 segundo entre cada secuencia  
}  
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 9 Termómetro

En esta lección, aprenderás a utilizar el Termistor NTC MF52 10K para medir la temperatura y visualizar los valores en un display LCD 16x2 con interfaz I2C, utilizando una tarjeta UNO R3 CH340G

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Termistor NTC MF52 10K
- Protoboard
- Display LCD 16x2 con I2C(no incluida)
- Cables para protoboard HM y MM
- Resistencia de 10K

Conocimientos previos

El Termistor NTC MF52 10K es un sensor cuya resistencia disminuye al aumentar la temperatura. Se usa en monitoreo térmico, control de temperatura y protección contra sobrecalentamiento. Antes de utilizar este sensor, es útil entender los siguientes conceptos básicos.

Propiedades clave

Para usar este sensor correctamente, es importante conocer sus características y valores esenciales.

Característica	Valor
Tipo	NTC (Coeficiente de temperatura negativo)
Resistencia nominal (R25)	10K Ω a 25°C
Coeficiente Beta (β)	3950K
Tolerancia	$\pm 1\%$, $\pm 3\%$, $\pm 5\%$ (según fabricante)
Rango de temperatura	-40°C a 125°C

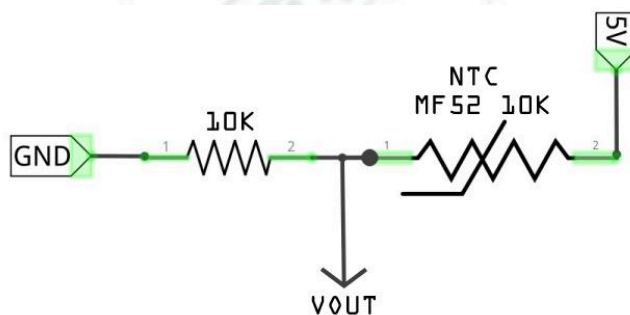
Starter KIT UNO R3

Tus primeros pasos en electrónica.

¿Cómo funciona el termistor NTC?

Su resistencia disminuye cuando la temperatura aumenta, lo que permite medir la temperatura a través de un divisor de voltaje. Para mayor precisión, se usa la ecuación de Steinhart-Hart o la ecuación Beta.

A continuación, se muestra el esquemático del divisor de voltaje usando para el **Termistor**:



Implementación del Termistor NTC

Para medir la temperatura con la tarjeta UNO R3 u otra tarjeta de desarrollo, el termistor se conecta en un divisor de voltaje junto con una resistencia de 10KΩ. La tarjeta Uno R3 mide el voltaje en el nodo central y calcula la resistencia del termistor con la fórmula:

$$R_{NTC} = \left(\frac{V_{cc} \times R_{fija}}{V_{medido}} \right) - R_{fija}$$

Donde:

- **Vcc** = 5V (Alimentación)
- **Rfija** = 10KΩ (resistencia en serie)
- **Vmedido** = Voltaje leído por la tarjeta Uno R3 en el pin analógico

Luego, la temperatura se obtiene con la ecuación Beta:

$$T = \frac{B \times T_0}{B + (T_0 \times \ln(R_{NTC}/R_0))}$$

Starter KIT UNO R3

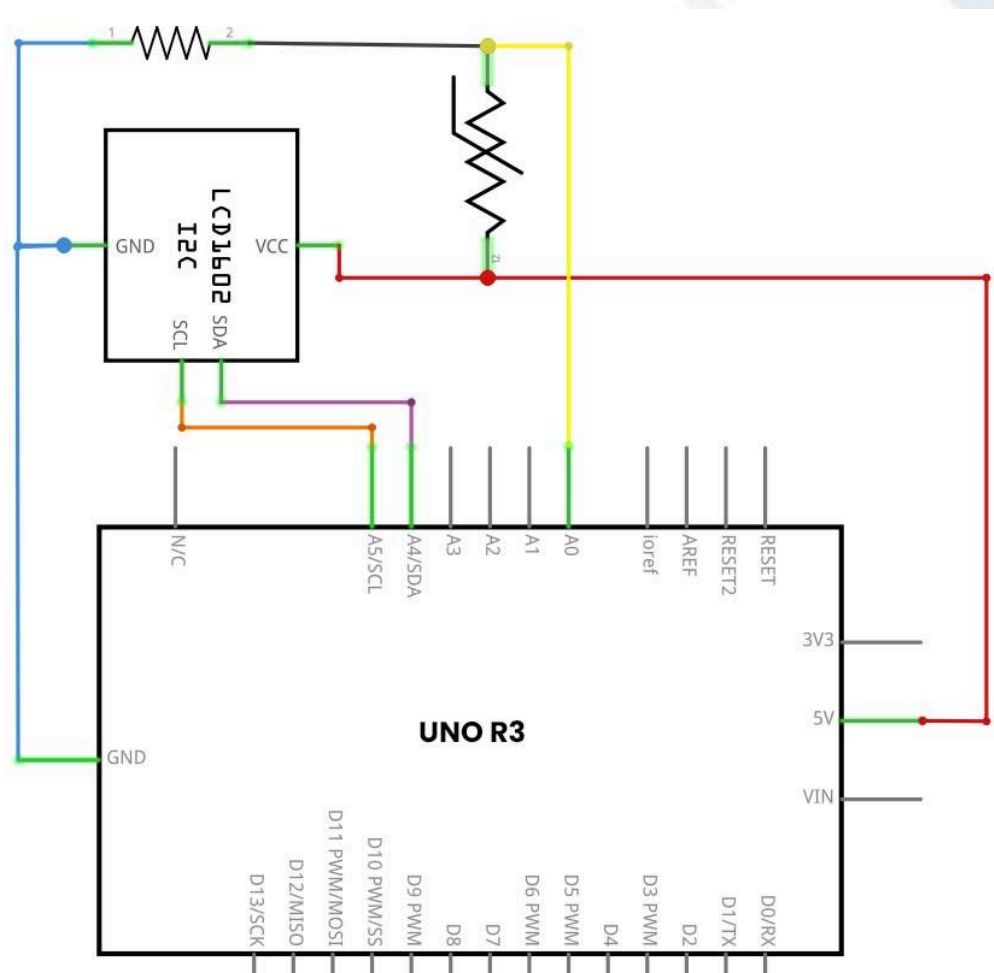
Tus primeros pasos en electrónica.

Donde:

- **B** = 3950K (coeficiente Beta)
- **T₀** = 298.15K (temperatura nominal en Kelvin, equivalente a 25°C)
- **R₀** = 10KΩ (resistencia a 25°C)
- **ln** = Logaritmo natural

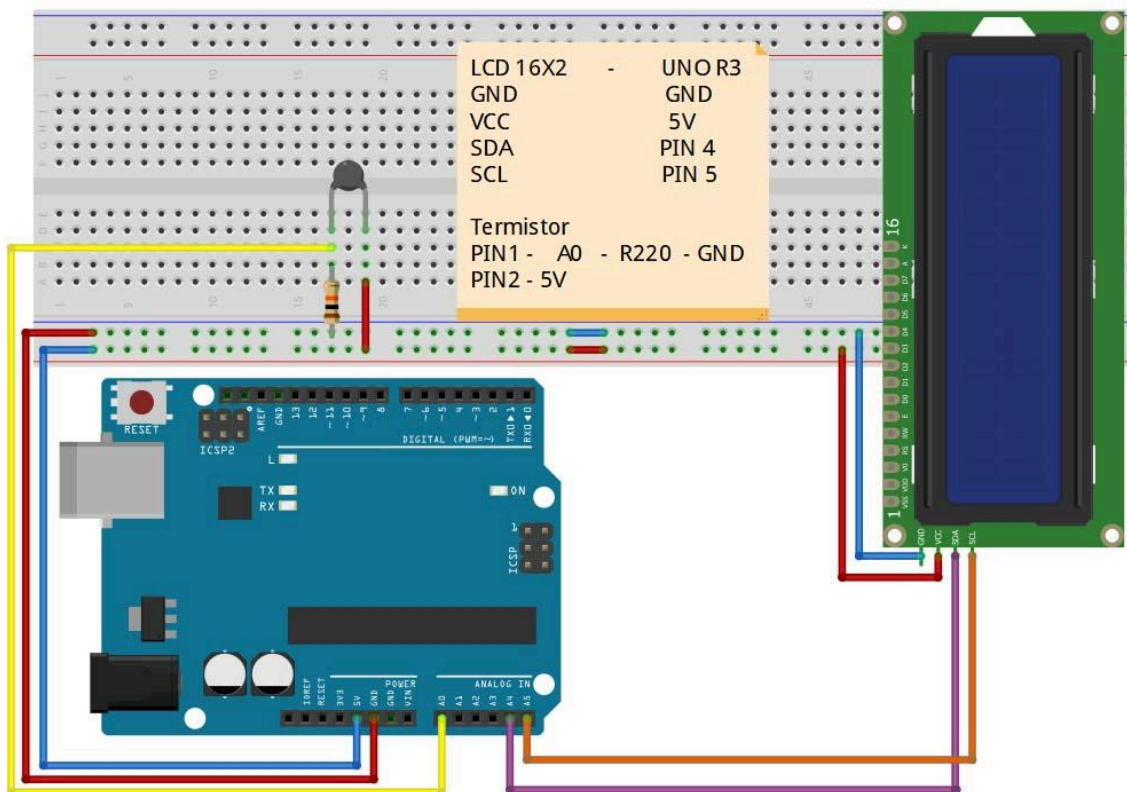
Diagrama de conexión

A continuación se muestra cómo conectar el sensor NTC MF52 10K y la pantalla LCD 16x2 a la tarjeta UNO R3 CH340G



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Código de funcionamiento

El siguiente código permite leer el estado del sensor NTC MF52 10K y visualizar la temperatura con el display LCD 16x2.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Definiciones de pines y constantes
#define TERMISTOR_PIN A0 // Pin analógico donde está conectado el
termistor
#define RESISTENCIA 10000 // Valor de la resistencia en ohmios

// Configuración del LCD con dirección I2C 0x27 y dimensiones 16x2
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Función para inicializar el LCD y la comunicación serie
void iniciarSistema() {
    lcd.init(); // Inicializa la pantalla LCD
    lcd.backlight(); // Enciende la retroiluminación del LCD
    Serial.begin(9600); // Inicializa la comunicación serie
    lcd.setCursor(0, 0);
    lcd.print("Temperatura:"); // Muestra el título en la pantalla
}

// Función para obtener la temperatura en grados Celsius
float obtenerTemperatura() {
    int lectura = analogRead(TERMISTOR_PIN); // Lee el valor analógico del
termistor
    float voltaje = lectura * (5.0 / 1023.0); // Convierte la lectura a voltaje
    float resistencia = (5.0 * RESISTENCIA / voltaje) - RESISTENCIA; // Calcula
la resistencia del termistor

    // Cálculo de temperatura en grados Celsius usando la ecuación de
Steinhart-Hart
    float temperatura = 1.0 / (0.001129148 + (0.000234125 * log(resistencia)) +
(0.0000000876741 * pow(log(resistencia), 3))) - 273.15;
```

```
return temperatura;
}

// Función para mostrar la temperatura en la pantalla LCD y en el monitor
serie
void mostrarTemperatura(float temperatura) {
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.println(" C");

    lcd.setCursor(0, 1);
    lcd.print(temperatura);
    lcd.print(" C "); // Espacios extra para sobrescribir datos previos
}

void setup() {
    iniciarSistema(); // Llama a la función para inicializar el sistema
}

void loop() {
    float temperatura = obtenerTemperatura(); // Obtiene la temperatura
actual
    mostrarTemperatura(temperatura); // Muestra la temperatura en el LCD
y en el monitor serie
    delay(1000); // Espera 1 segundo antes de la siguiente lectura
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 10 Sensor de Inclinación

En esta lección, aprenderás a utilizar el sensor de inclinación SW-520D, un dispositivo sencillo pero muy útil que detecta cambios en la inclinación cuando se supera su umbral.

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Protoboard
- Resistencias de 220 ohms
- Sensor de inclinación SW-520D
- Cables para protoboard MM
- Led rojo

Conocimientos previos

El sensor de inclinación SW-520D es un dispositivo que se utiliza para detectar cambios en la inclinación o la posición de un objeto. Antes de utilizar este sensor, es útil entender los siguientes conceptos básicos.

Funcionamiento interno: El sensor SW-520D contiene una pequeña esfera metálica en su interior que actúa como un interruptor mecánico. Dependiendo de su inclinación, la esfera se mueve y permite cerrar o abrir su circuito eléctrico.

Señal de salida: La señal de salida del sensor es digital, lo que significa que solo tiene dos estados: Alto (ON) o Bajo (OFF), dependiendo de su inclinación. En la siguiente imagen se muestran los estados dependiendo de su inclinación:



Este comportamiento lo hace ideal para detectar cambios simples, si un objeto está inclinado o no.

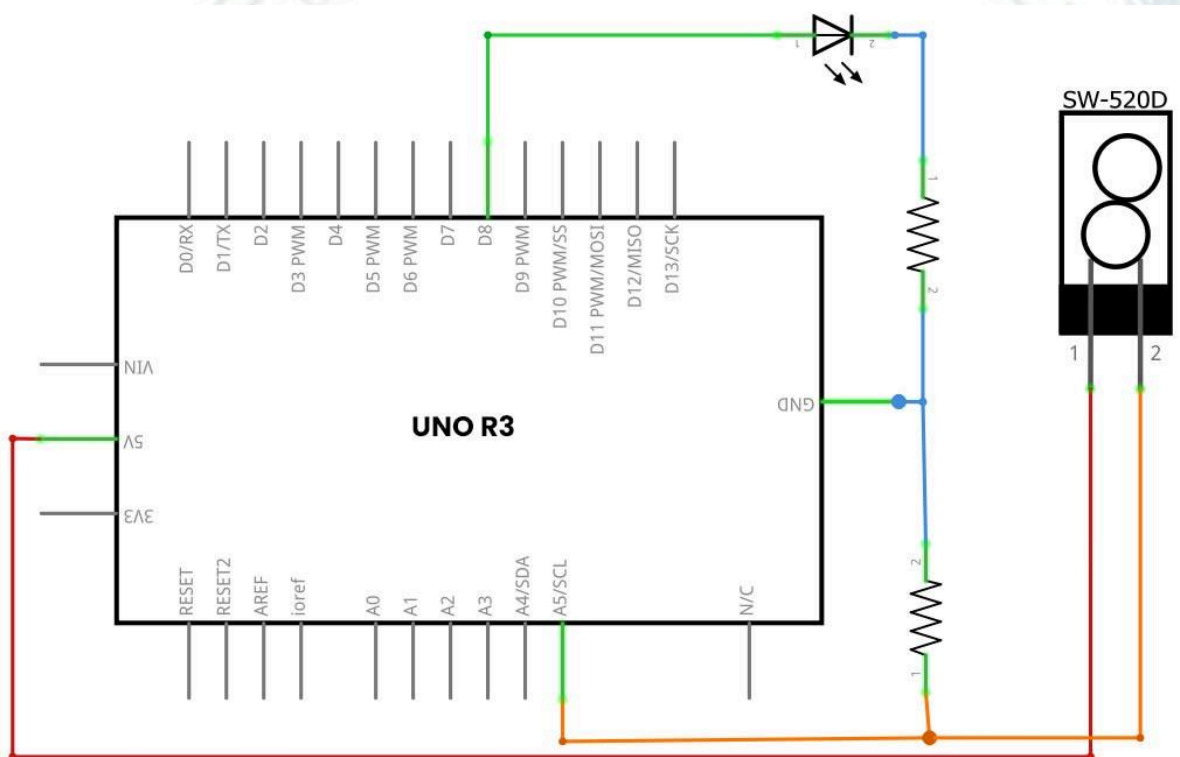
Starter KIT UNO R3

Tus primeros pasos en electrónica.

Precisión y sensibilidad: El SW-520D no detecta ángulos específicos, sino cambios de posición. Para mediciones más precisas de inclinación, se requieren otros sensores como acelerómetros.

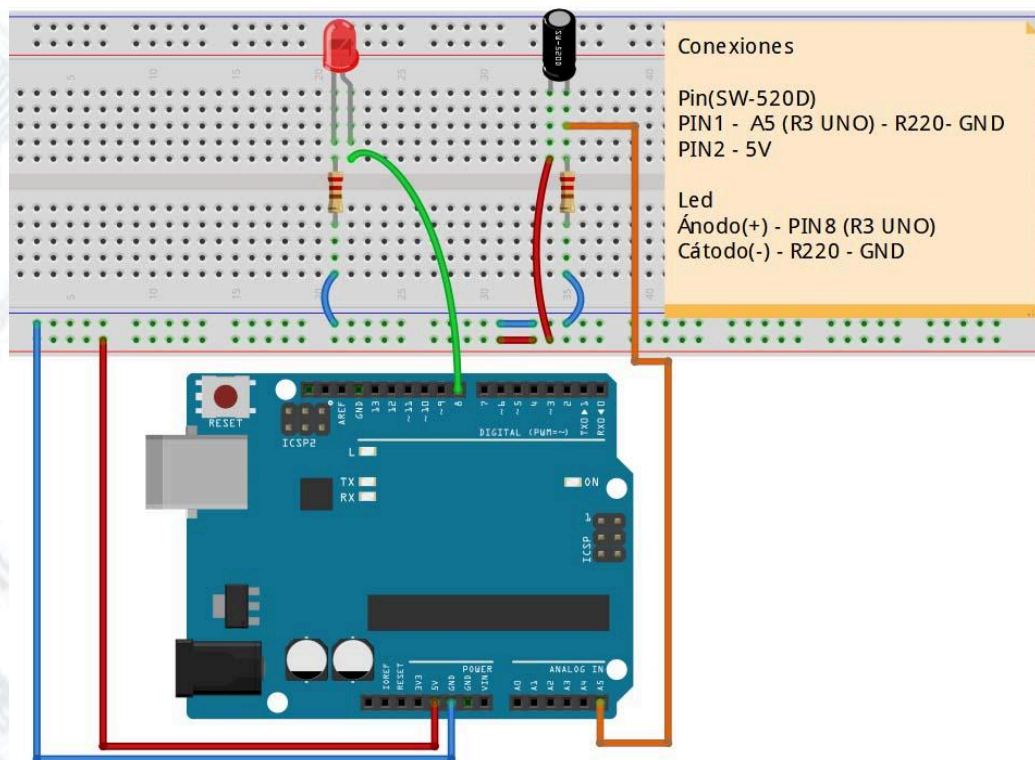
Diagrama de conexión

A continuación, se detallan las conexiones entre el sensor de inclinación, la tarjeta UNO R3 CH340G y el LED, el cual se encenderá o apagará en función de la respuesta del sensor de inclinación.



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Código de funcionamiento

El siguiente código permite leer el estado del sensor de inclinación para encender o apagar un led dependiendo de la inclinación del sensor.

```
const int LedPin = 8; //Variable para el Pin 8 que se conectará con el Led
void setup() {
  pinMode(LedPin, OUTPUT);}
void loop(){ //Función Loop
  int i;
  while (1) { //While para tener una instrucción bucle sin fin
    i = analogRead(5); //La variable i tendrá lectura de los valores
    provenientes del pin 5 analógico
    if (i > 512){ //rango ajustable de 0 a 1023, 512 tendremos una sensibilidad
    media en el sensor
      digitalWrite(LedPin, LOW); } //El led apagará, en posición vertical del
    sensor else {
      digitalWrite(LedPin, HIGH); }}} //El led se encenderá, en posición horizontal
    del sensor
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 11 DHT11 Sensor de temperatura y humedad

En esta lección, aprenderás a utilizar el sensor DHT11 para medir temperatura y humedad, mostrando los valores en un display LCD 16x2 con interfaz I2C. Esta combinación es útil en proyectos de monitoreo ambiental, estaciones meteorológicas y control de temperatura en interiores.

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Sensor DHT11
- Display LCD 16x2 I2C (no incluida)
- Protoboard
- Cables para protoboard MH y MM

Conocimientos previos

El DHT11 es un sensor digital que mide temperatura y humedad. Es utilizado en proyectos de domótica, estaciones meteorológicas y sistemas de control ambiental.

Antes de utilizar este sensor, es útil entender los siguientes conceptos básicos.

- **Temperatura:**
 - Rango de medición: **0°C a 50°C**
 - Precisión: **±2°C**
- **Humedad:**
 - Rango de medición: **20% a 90%**
 - Precisión: **±5%**

Funcionamiento

El sensor combina dos tecnologías de medición: un termistor para detectar la temperatura ambiente y un sensor capacitivo para medir la humedad relativa del aire.

Starter KIT UNO R3

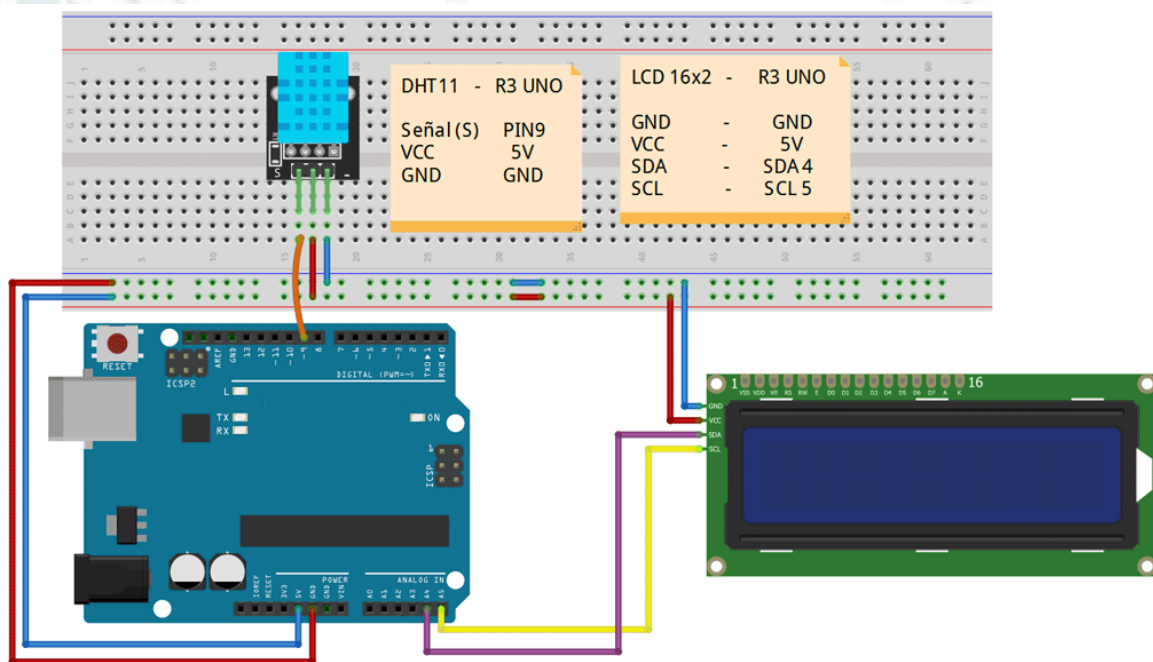
Tus primeros pasos en electrónica.

Principio de Operación

- La comunicación es digital, eliminando la necesidad de conversión analógica.
- Para obtener una medición, el microcontrolador envía una señal de inicio al sensor.
- El DHT11 responde enviando un conjunto de bits digitales, donde se codifican los valores de temperatura y humedad.
- Solo requiere un único pin de datos, lo que simplifica su conexión con placas Arduino, ESP32, Raspberry Pi y otras tarjetas.

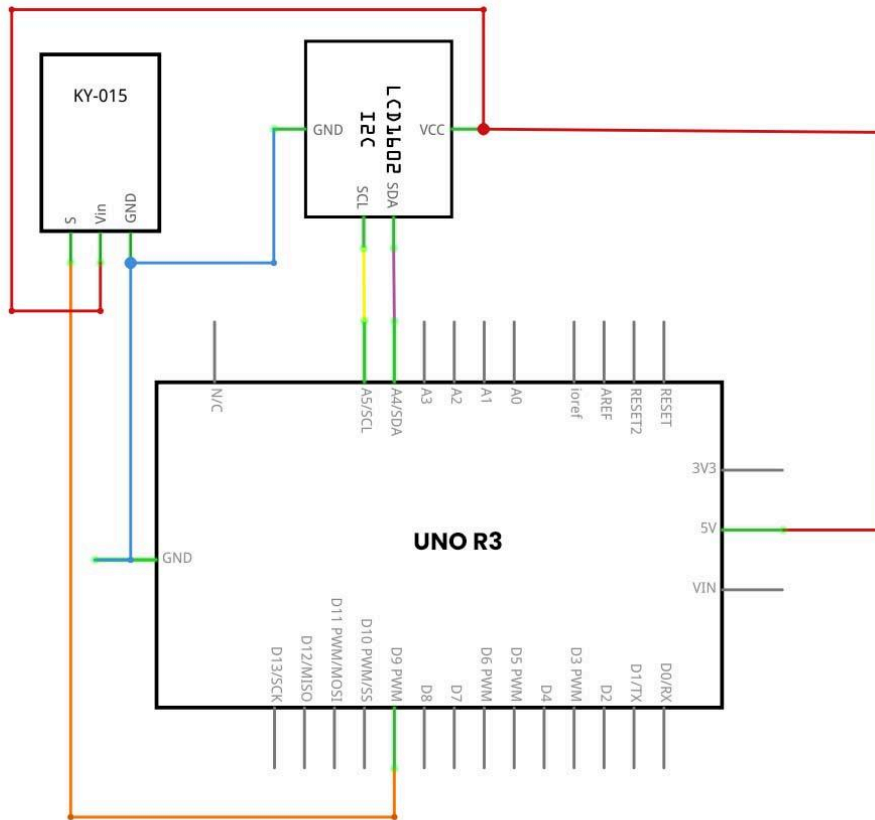
Diagrama de conexión

A continuación se muestra cómo conectar el sensor DHT11 y la pantalla LCD 16x2 a la tarjeta UNO R3 CH340G



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Código de funcionamiento

El siguiente código permite leer el estado del sensor DHT11 y visualizar la temperatura y humedad con el display LCD 16x2. Antes de cargar el código, asegúrate de instalar las siguientes librerías en el Arduino IDE para su correcto funcionamiento.

LiquidCrystal_I2C.h



DHT.h



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Para utilizar la librería DHT.h correctamente, es fundamental instalar todos sus complementos y dependencias. Se recomienda instalar la librería desde el **Gestor de Librerías** de Arduino IDE e instalar todas sus dependencias para contar con la versión más reciente, compatible y evitar errores en la lectura de datos.

Instalar las dependencias de la biblioteca



La biblioteca DHT sensor library:1.4.6 necesita otra dependencia que actualmente no está instalada:

- Adafruit Unified Sensor

¿Quieres instalar la dependencia faltante?

INSTALAR SIN DEPENDENCIAS

INSTALAR TODO

```
#include <Wire.h> // Librería para comunicación I2C
#include <LiquidCrystal_I2C.h> // Librería para el LCD con I2C
#include <DHT.h> // Librería para el sensor DHT
// Definimos el pin donde está conectado el sensor DHT11
#define DHTPIN 9
// Definimos el tipo de sensor DHT (puede ser DHT11, DHT22, etc.)
#define DHTTYPE DHT11
// Creamos un objeto DHT con el pin y tipo de sensor especificados
DHT dht(DHTPIN, DHTTYPE);
// Creamos un objeto para el LCD I2C con la dirección 0x27 y dimensiones
// 16x2 (16 columnas, 2 filas)
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  Serial.begin(9600); // Iniciamos la comunicación serie a 9600 baudios
  dht.begin(); // Inicializamos el sensor DHT11
  lcd.init(); // Inicializa el LCD con la configuración establecida
  lcd.backlight(); // Enciende la luz de fondo del LCD
  // Mostramos un mensaje de inicio en el LCD
  lcd.setCursor(0, 0); // Posicionamos el cursor en la primera columna y
  // primera fila
  lcd.print("Iniciando..");
  delay(2000); // Esperamos 2 segundos antes de comenzar las lecturas
}
```

```

void loop() {
    // Leemos la temperatura y la humedad desde el sensor
    float temperatura = dht.readTemperature(); // Obtener temperatura en
    grados Celsius
    float humedad = dht.readHumidity(); // Obtener humedad relativa en %
    // Verificamos si la lectura es válida (el sensor puede fallar
    ocasionalmente)
    if (isnan(temperatura) || isnan(humedad)) {
        Serial.println("Error al leer DHT11"); // Mensaje de error en el Monitor Serie
        return; // Salimos de la función loop() y esperamos la siguiente iteración
    }
    // Mostrar los valores en el Monitor Serie
    Serial.print("Temp: ");
    Serial.print(temperatura);
    Serial.print(" °C | Hum: ");
    Serial.print(humedad);
    Serial.println(" %");
    // Mostrar los valores en la pantalla LCD
    lcd.clear(); // Limpiamos la pantalla antes de escribir nuevos valores
    lcd.setCursor(0, 0); // Posicionamos el cursor en la primera columna y
    primera fila
    lcd.print("Temp: ");
    lcd.print(temperatura); // Mostramos la temperatura en la pantalla
    lcd.print(" C"); // Agregamos la unidad de medida (°C)
    lcd.setCursor(0, 1); // Posicionamos el cursor en la primera columna y
    segunda fila
    lcd.print("Hum: ");
    lcd.print(humedad); // Mostramos la humedad en la pantalla
    lcd.print(" %"); // Agregamos la unidad de medida (%)
    delay(2000); // Esperamos 2 segundos antes de realizar una nueva
    lectura
}

```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 12 Sensor de sonido

En esta lección aprenderás a construir un sencillo sonómetro cuyas lecturas podrás visualizar en el serial plotter del IDE además de un interruptor por aplausos con el que podrás controlar un relevador mediante el sensor de sonido KY-037 y la tarjeta UNO R3 CH340G

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Transistor BC547
- Diodo 1N4007
- LED
- Relevador SPDT 5V
- Fuente de alimentación para protoboard
- Eliminador 9V 1A
- Resistencias 10k, 22k, 100 todas a 1/4w
- Protoboard
- Sensor de sonido KY-037

Conocimientos previos

El KY-037 es un sensor de sonido que puede detectar sonidos o niveles de ruido en su entorno. Es un dispositivo comúnmente utilizado con microcontroladores como la tarjeta UNO R3 CH340G, el cual tiene dos partes principales.



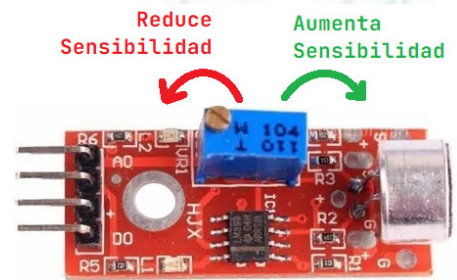
Micrófono incorporado: El sensor tiene un micrófono que capta las ondas sonoras que se generan en su entorno. Estas ondas sonoras son transformadas en señales eléctricas por el micrófono.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Amplificador de señal: La señal del micrófono es muy débil, por lo que pasa por un amplificador operacional LM393 para aumentar su intensidad, lo que facilita que el sensor detecte el sonido de manera más eficiente.

Dicha amplificación estará relacionada con el potenciómetro de sensibilidad que viene integrado en el sensor, para lo cual, según la aplicación, primero deberás girar la perilla hasta que obtengas la activación del sensor, que notarás cuando se encienda el led a la intensidad sonora que desees.



Salida digital y analógica

Salida digital: El sensor tiene una salida digital (pin DO) que activa un "alto" (1) o "bajo" (0) dependiendo de si el sonido detectado supera un umbral determinado. Este umbral se ajusta a través de un potenciómetro en el módulo, lo que te permite determinar la sensibilidad del sensor. Si el sonido detectado es lo suficientemente fuerte como para superar ese umbral, la salida digital cambiará.

Salida analógica: También tiene una salida analógica (pin AO) que da una señal proporcional a la intensidad del sonido. Esto permite obtener una lectura más precisa de la amplitud del sonido, en lugar de solo detectar su presencia.

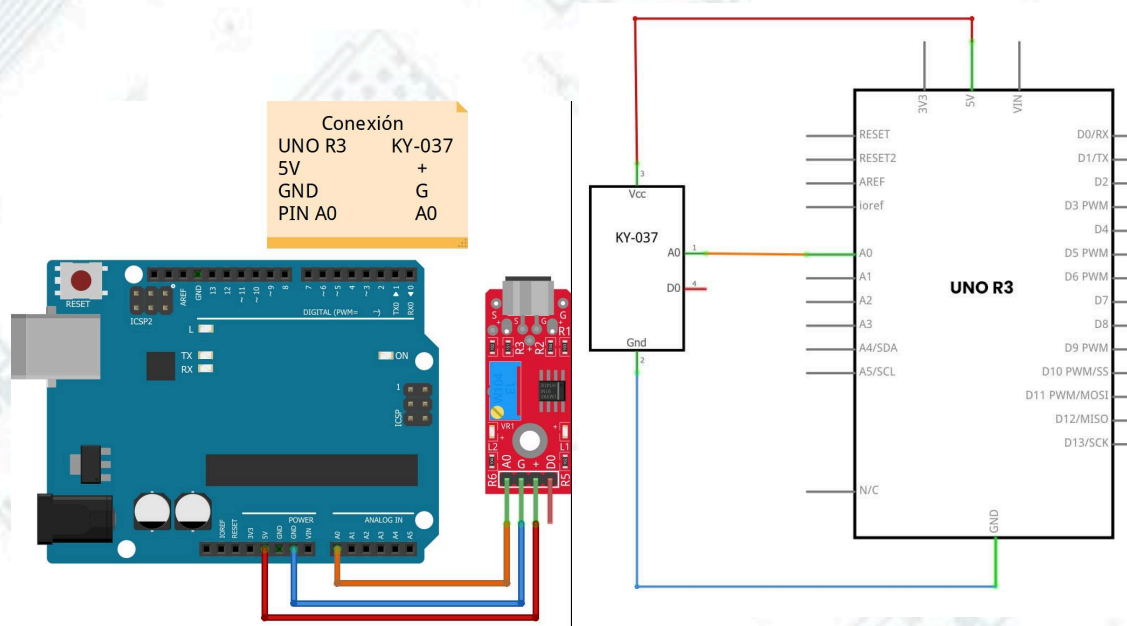
Puedes utilizar este sensor en proyectos de detección de sonido, control de volumen, alarmas de sonido, y otros sistemas que necesiten medir o reaccionar al sonido.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Diagrama de conexión

Para este pequeño prototipo haremos uso de la salida analógica del sensor de sonido (A0)



Código de funcionamiento

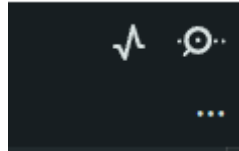
Como podrás observar, en este programa se realizará el despliegue de las lecturas en el serial plotter a través de la sentencia `Serial.println(sensor)`; por ello para poder tener un instrumento de medición sonora tendrás que interpretar los valores que te entrega el ADC de la tarjeta UNO R3 respecto a una escala.

```
#define sonido A0
float sensor = 0.0;
void setup(){
  pinMode(sonido, INPUT);
  Serial.begin(9600);}
void loop(){
  sensor = analogRead(sonido);
  Serial.println(sensor);
  delay(300);
}
```

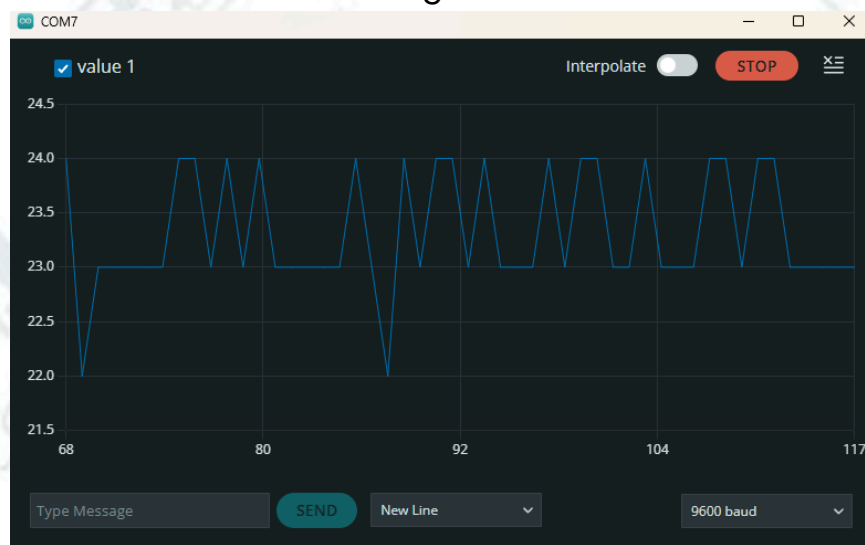
Starter KIT UNO R3

Tus primeros pasos en electrónica.

Para abrir el serial plotter deberás dar click en el siguiente icono ubicado en la parte superior derecha del IDE mientras la tarjeta R3 UNO permanece conectado a tu PC



Y obtendrás una ventana como la siguiente:



Si tienes problemas para obtener el serial plotter, revisa que puerto COM pertenece al R3 UNO.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección 13 Sensor de humedad del suelo

En esta lección, aprenderá cómo usar el sensor de humedad del suelo en sus proyectos con la tarjeta UNO R3 CH340G. En este ejemplo, se leerán los valores de salida del sensor encendiendo un color diferente en un led RGB además de una bocina como alarma para los diferentes niveles de humedad detectados.

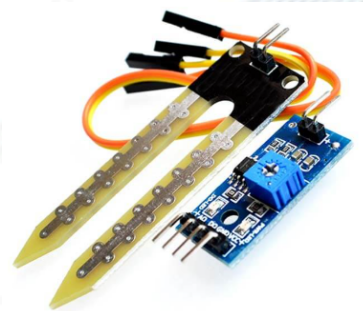
Materiales necesarios

- Tarjeta UNO R3 CH340G
- Led RGB catodo comun
- Resistencias de 220Ω a 1/4w
- Buzzer activo
- Módulo sensor de humedad del suelo FC-28
- Protoboard

Conocimientos previos

Este sensor FC-28 está compuesto por dos partes principales:

Sonda de Humedad: Es la parte que se inserta en el suelo y se encuentra formada por dos electrodos que actúan como sensores para medir la resistencia eléctrica del suelo. La resistencia de los electrodos varía dependiendo de la cantidad de agua en el suelo. Cuanta más agua haya en el suelo, menor será la resistencia, ya que el agua conduce la electricidad mejor que el suelo seco.



Circuito de Medición: El circuito toma la señal de los electrodos y la convierte en un valor analógico que puede ser leído por un microcontrolador. Este valor puede ser interpretado para determinar el nivel de humedad del suelo.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

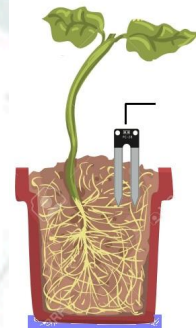
Starter KIT UNO R3

Tus primeros pasos en electrónica.

¿Cómo se mide la humedad?

Suelo seco: La resistencia entre los electrodos será alta, ya que el agua es un buen conductor de electricidad y en su ausencia el suelo no conduce bien la corriente. Esto indica un bajo nivel de humedad.

Suelo mojado: La resistencia será baja, ya que el agua presente en el suelo facilita la conducción de electricidad, lo que indica un alto nivel de humedad.

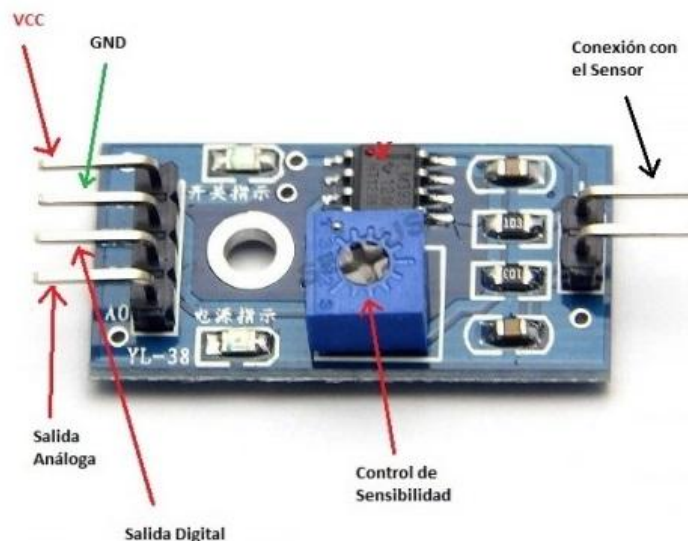


Salida del sensor

El FC-28 tiene dos salidas:

Salida analógica(A0): Esta salida proporciona un valor proporcional al nivel de humedad. A medida que la humedad aumenta, el valor de la señal analógica disminuirá (menos resistencia = más humedad).

Salida digital(D0): Este pin genera una señal de encendido (HIGH) o apagado (LOW) basada en un umbral preestablecido. Por ejemplo, cuando la humedad baja por debajo de un nivel determinado, el pin digital se activa para encender una bomba de riego o enviar una alarma.



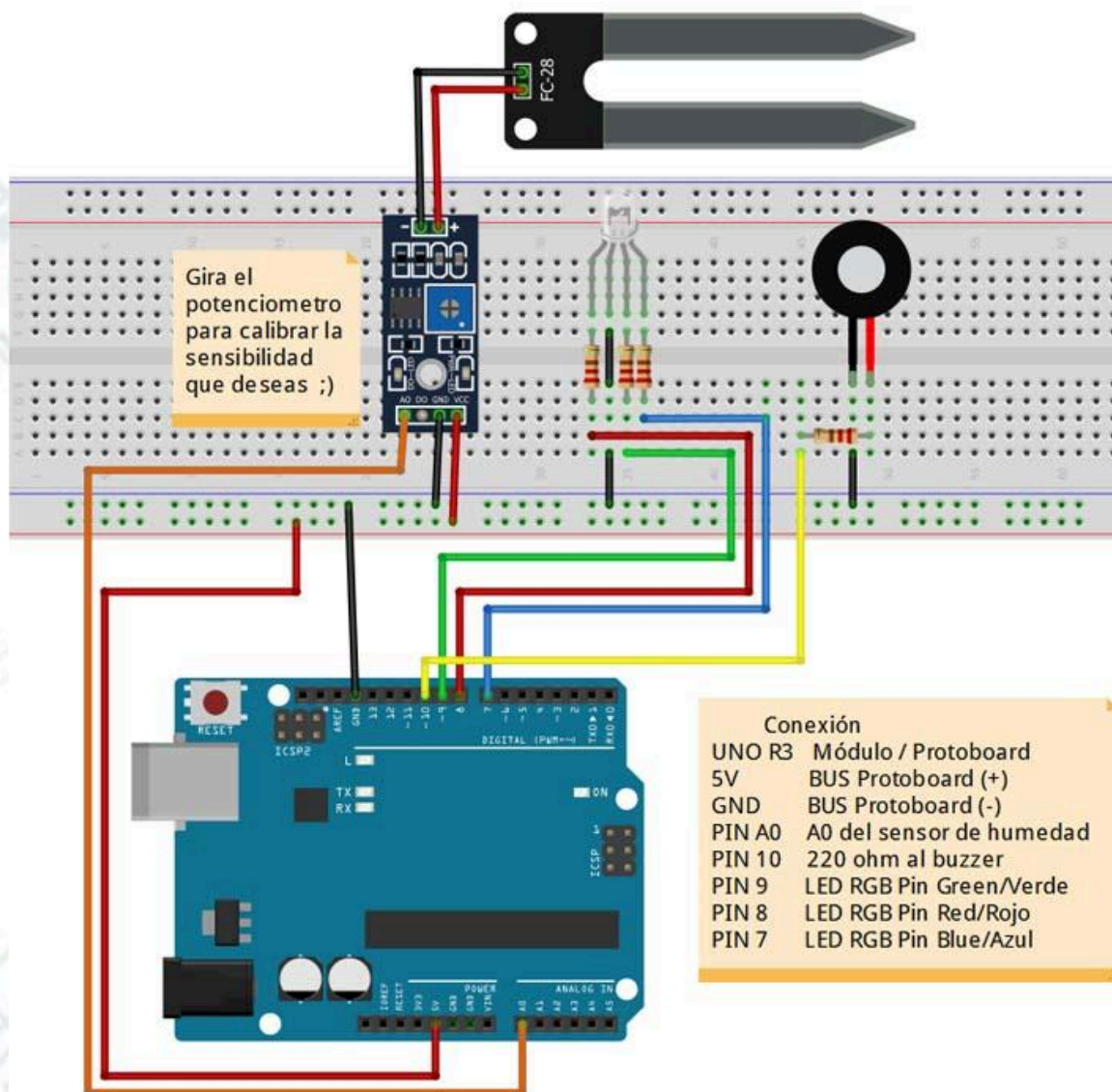
La precisión del sensor puede verse afectada por el tipo de suelo, ya que su conductividad varía según la cantidad de sal y minerales que tenga.

Starter KIT UNO R3

Tus primeros pasos en electrónica.

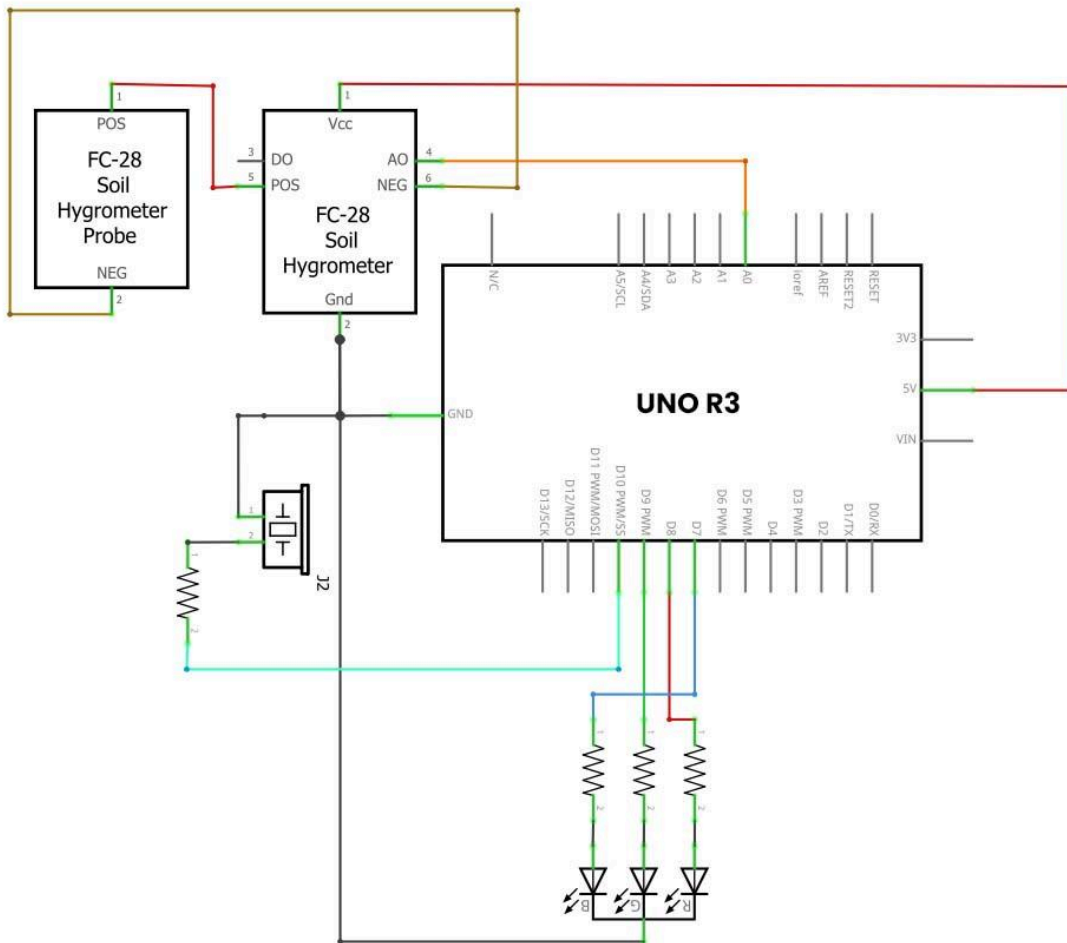
Diagrama de conexión

A diferencia de las prácticas donde se utilizaba el módulo de fuente de alimentación, en este caso alimentamos a todos los componentes desde la tarjeta R3 UNO y este a su vez está conectado por medio del cable USB a la PC.



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Código de funcionamiento

Este código realiza la lectura analógica del sensor de humedad que adquiere valores de 0 a 1023 interpretando el 0 como máxima humedad detectada y el 1023 como el ambiente más seco al que la sonda puede estar expuesta y evaluando tres rangos de dichas lecturas para encender cada uno de los leds internos del led RGB y una alarma sonora mediante un buzzer.

```
// Declaramos variables con el número de Pin en el Arduino
int sensor = A0;
int verde = 24;
int rojo = 26;
int azul = 28;
int alarma = 22;
```

```
void setup(){
  // Configuramos el pin del sensor como entrada
  pinMode(sensor, INPUT);
  // Configuramos salidas
  pinMode(verde, OUTPUT);
  pinMode(rojo, OUTPUT);
  pinMode(azul, OUTPUT);
  pinMode(alarma, OUTPUT);
  digitalWrite(alarma, LOW);
}

void loop() {
  // Leemos en el pin analógico 0 conectado a la salida del sensor de
  humedad
  int humedad = analogRead(sensor);
  // Si el sensor entrega valores entre 0 y 341 se enciende el led azul
  if(341 >= humedad && humedad>0){
    digitalWrite(azul, HIGH);
    digitalWrite(rojo, LOW);
    digitalWrite(verde, LOW);
    digitalWrite(alarma, LOW);
  }
  // Si el sensor entrega valores de 341 a 682 se enciende el led verde
  else if(682 >= humedad && humedad>341){
    digitalWrite(azul, LOW);
    digitalWrite(rojo, LOW);
    digitalWrite(verde, HIGH);
    digitalWrite(alarma, LOW);
  }
  // Si el sensor entrega valores de 682 a 1023 se enciende el led rojo a la par
  que una alarma
  else if(1023 >= humedad && humedad>682){
    digitalWrite(azul, LOW);
    digitalWrite(rojo, HIGH);
    digitalWrite(verde, LOW);
    digitalWrite(alarma, LOW);
    digitalWrite(alarma, HIGH);
    delay(50);
    digitalWrite(alarma, LOW);
    delay(50);
  }
  delay(500);
}
```

Starter KIT UNO R3

Tus primeros pasos en electrónica.

Lección Bonus: Nivel de Agua

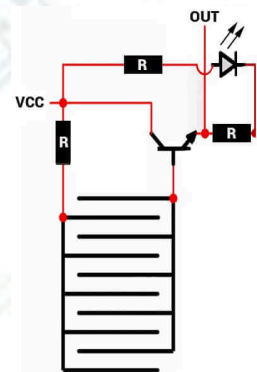
En esta lección aprenderás a construir y programar un sencillo control a través de un motor a pasos de una ventana que se cierra cuando se detecta lluvia sobre el sensor de agua. Estas acciones serán desplegadas en forma de mensajes en una LCD 16X2

Materiales necesarios

- Tarjeta UNO R3 CH340G
- Módulo fuente de alimentación para protoboard (no incluido)
- Sensor de Humedad del Suelo FC-28 Higrómetro
- Motor a pasos 28BYJ-48 (no incluido)
- Driver para motor a pasos ULN2003 (no incluido)
- LCD 16X2 (no incluido)
- Cables jumper M-M y M-H
- Protoboard

Conocimientos previos

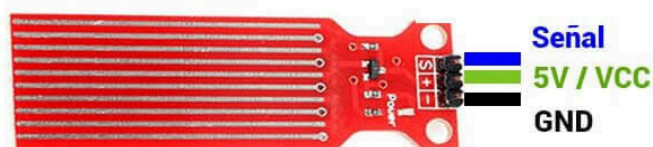
El sensor de nivel de líquidos es un dispositivo que entrega una señal analógica de acuerdo con el cambio en la conductividad eléctrica de las pistas impresas en su placa cuando entran en contacto con el agua común, que, según las sales disueltas, tendrá más o menos conductividad, dicho cambio de la resistencia eléctrica provocará una corriente de base que para un transistor BJT tipo NPN significa una corriente proporcional en el colector, lo que a su vez implica una tensión capaz de ser leída por el ADC de un microcontrolador.



Starter KIT UNO R3

Tus primeros pasos en electrónica.

Podemos caracterizar los diferentes niveles de agua que pueda detectar el sensor para un ADC con una resolución de 10 bits, es decir, de 1023 pasos, tal como a continuación se observa en la imagen



Valor Analógico	Nivel de Líquido
0	0%
256	25%
512	50%
1023	100%

**Resolución
10 bit
(0 a 1023)**

Starter KIT UNO R3

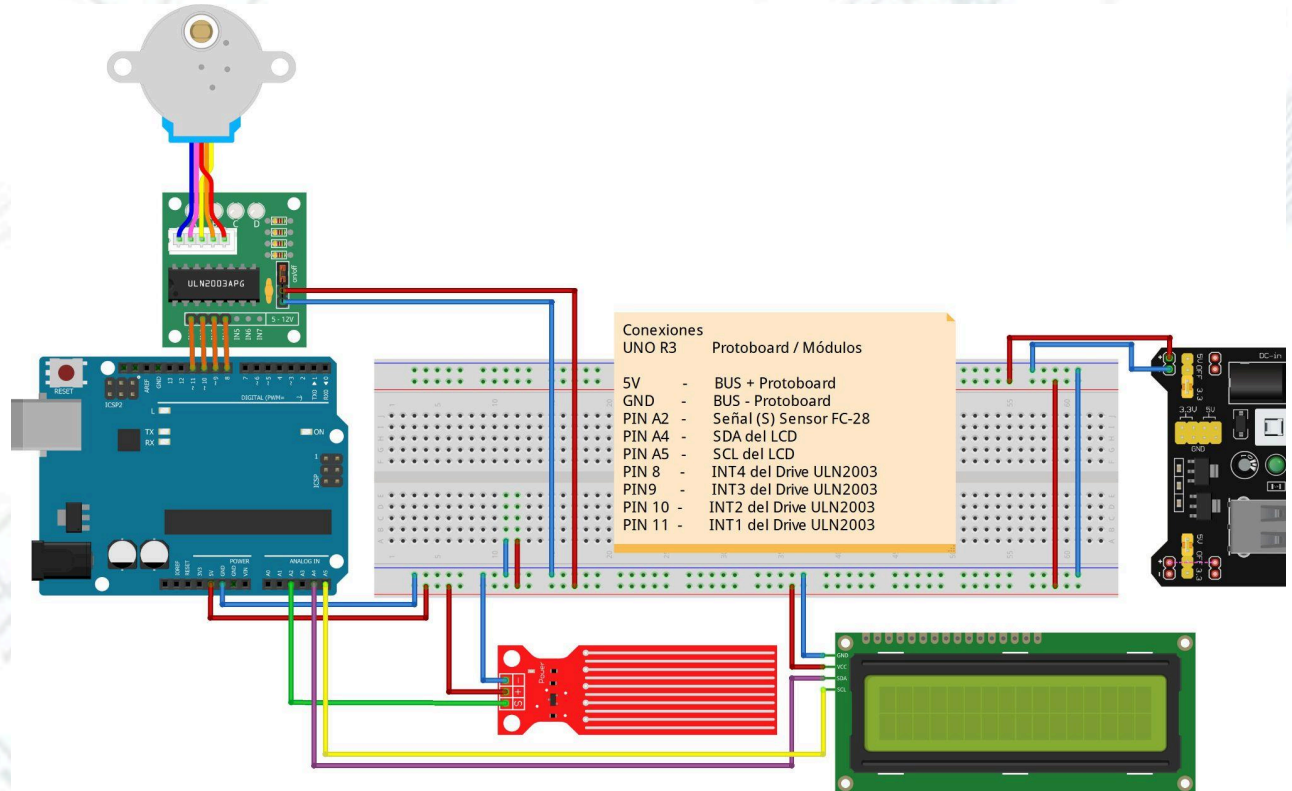
Tus primeros pasos en electrónica.

Diagrama de conexión

Sigue exactamente el diagrama como se muestra en la imagen, sobre todo teniendo en cuenta que jamás deberías conectar la alimentación del driver del motor a pasos directamente a la tarjeta UNO R3 CH340G ya que consume demasiada corriente y corres el riesgo de quemar tu placa.

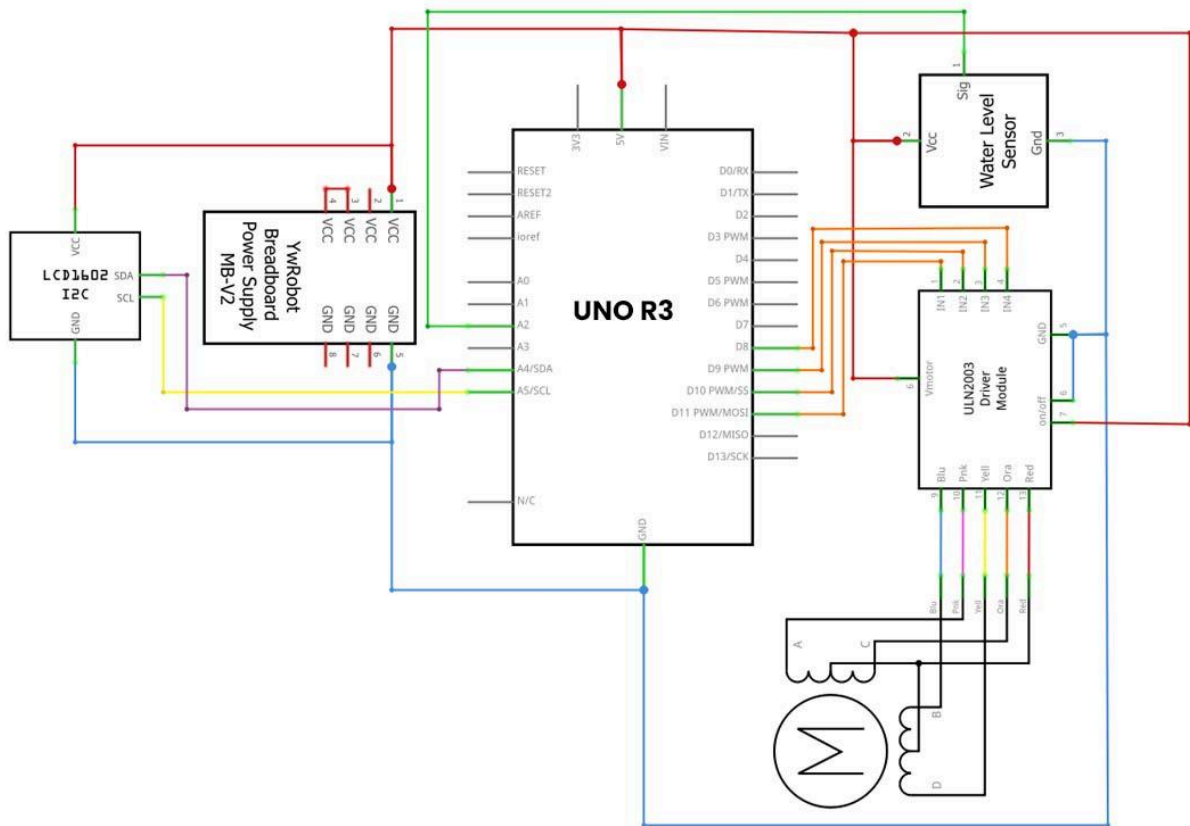
Presta especial atención a que el módulo de interfaz I2C con la LCD nos evita tener que conectar los 16 pines de la pantalla al UNO R3, ocupando sólo dos pines de comunicación (SCL y SDA) además de los pines de alimentación.

Si no tienes imagen en la pantalla prueba presionando el botón del reset del UNO R3 o moviendo el potenciómetro del módulo interfaz I2C para variar el contraste hasta que aparezcan los caracteres.



Starter KIT UNO R3

Tus primeros pasos en electrónica.



Código de funcionamiento

Una vez que cargues el código te retamos a modificarlo añadiendo un buzzer que se active cuando la venta se cierra o se abre ;)

```
//Incluimos las librerías para la comunicación I2C y
// para el control de la LCD
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Definimos los pines que van a usar el driver ULN2003 y
// la salida del sensor de agua
#define sensoragua A2
#define motorPin1 11 // IN1
#define motorPin2 10 // IN2
#define motorPin3 9 // IN3
#define motorPin4 8 // IN4
```

```

// Definir el arreglo de secuencia de pasos para el motor 28BYJ-48
int pasos[8][4] = {
  {1, 0, 0, 1}, // Paso 1
  {1, 0, 0, 0}, // Paso 2
  {1, 1, 0, 0}, // Paso 3
  {0, 1, 0, 0}, // Paso 4
  {0, 1, 1, 0}, // Paso 5
  {0, 0, 1, 0}, // Paso 6
  {0, 0, 1, 1}, // Paso 7
  {1, 0, 1, 0} // Paso 8
};

// Declaramos las variables que nos ayudarán en la lógica de
programación
int nivelagua = 0, op = 0, og = 0;
// Esta variable determinará la sensibilidad del sensor de agua
// si incrementas su valor, entonces se necesitará de más agua para
activar el sistema
int umbral = 100;
// Colocamos la dirección I2C de la interfaz que es 0x27 además de
declarar que es un LCD DE 16X2
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  //Inicializamos la LCD y encendemos la luz de fondo de la pantalla
  lcd.init();
  lcd.init();
  lcd.backlight();
  // Declaramos salidas y entradas
  pinMode(sensoragua, INPUT);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
}

void loop()
{
  //Realizamos un promedio de 150 lecturas del sensor de agua

```

```

for(int i = 0; i < 150; i++){
    nivelagua += analogRead(sensoragua);
    delay(10);
}
int lectura = (nivelagua / 150);
// Comparamos la lectura del sensor con el valor umbral
// Si hay agua, es decir, si la lectura es mayor al umbral entonces
// se muestra en la LCD el mensaje "LLUVIA DETECTADA" y el motor a pasos
gira
if(lectura > umbral)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("LLUVIA DETECTADA ");
    lcd.setCursor(0,1);
    lcd.print("Ventana cerrada");

    // Con este bucle while logramos que el motor gire una vuelta y se
detenga
while(op == 1){
    for (int i = 0; i < 512; i++) { // 512 pasos para una vuelta completa
        for (int j = 7; j >= 0; j--) {
            digitalWrite(motorPin1, pasos[j][0]);
            digitalWrite(motorPin2, pasos[j][1]);
            digitalWrite(motorPin3, pasos[j][2]);
            digitalWrite(motorPin4, pasos[j][3]);
            delayMicroseconds(1000); //Determina la velocidad de giro
        }
        // Aquí apagamos el motor porque sin poner a LOW los pines el motor
podría quemar a la fuente
        if(i == 511) {
            op = 0;
            digitalWrite(motorPin1, LOW);
            digitalWrite(motorPin2, LOW);
            digitalWrite(motorPin3, LOW);
            digitalWrite(motorPin4, LOW);
        }
    }
}

//Leemos el sensor para que en el momento que se detecte que ya no
hay agua salimos del if

```

```

for(int i = 0; i < 150; i++){
    nivelagua += analogRead(sensoragua);
    delay(10);
}
int lectura = (nivelagua / 150);
op = 0;
og = 1;
}

// Si el sensor está seco entonces se abre la ventana y se manda el
mensaje en la LCD
else{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Ventana abierta");

    nivelagua = analogRead(sensoragua);
    lectura = constrain( nivelagua, 0,1023);

//Giramos el motor en sentido inverso y cuando da la vuelta se apaga
while(og == 1){
    for (int i = 0; i < 512; i++) { // 512 pasos para una vuelta completa
        for (int j = 0; j < 8; j++) {
            digitalWrite(motorPin1, pasos[j][0]);
            digitalWrite(motorPin2, pasos[j][1]);
            digitalWrite(motorPin3, pasos[j][2]);
            digitalWrite(motorPin4, pasos[j][3]);
            delayMicroseconds(1000); // Retardo de 2000 microsegundos (2ms)
        }
        if(i == 511) {
            og = 0;
            digitalWrite(motorPin1, LOW);
            digitalWrite(motorPin2, LOW);
            digitalWrite(motorPin3, LOW);
            digitalWrite(motorPin4, LOW);
        }
    }
}
op = 1; // Esta variable junto con og nos ayudan a detener al motor
}
}

```

Gracias por su preferencia



Manual elaborado por el Equipo UNIT Electronics
Marzo 2025