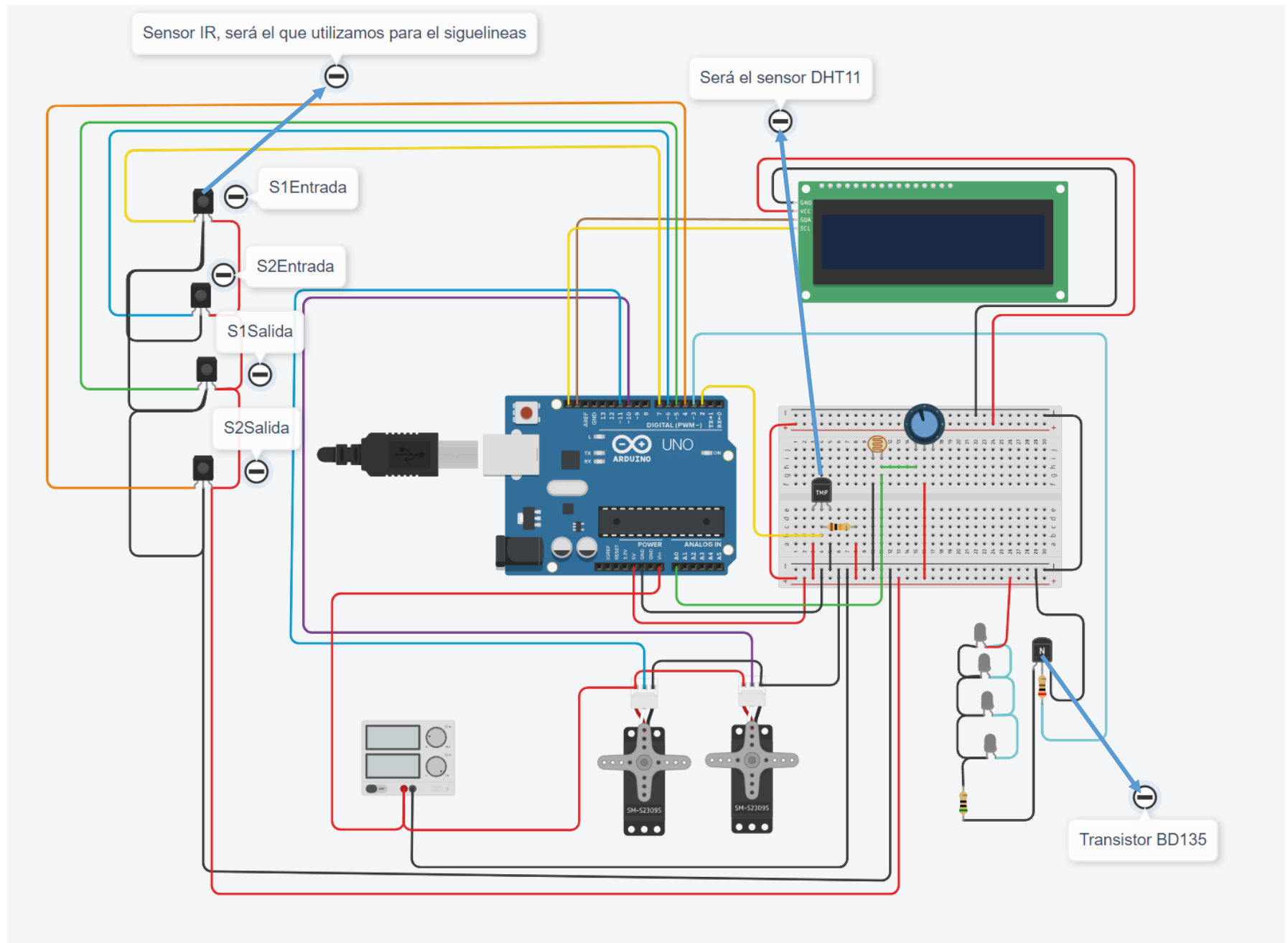
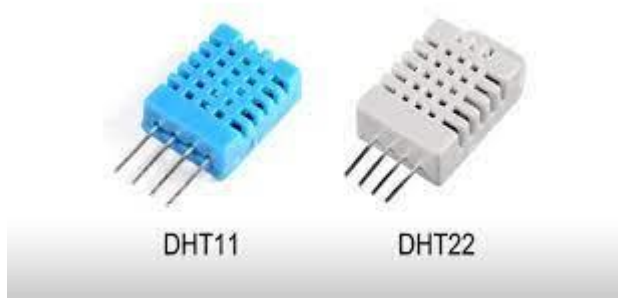


# PARKING



# PARKING

## SENSOR DE TEMPERATURA Y HUMEDAD



El DHT11 y el DHT22 son dos modelos de una misma familia de sensores, que permiten realizar la medición simultánea de temperatura y humedad.

Ambos sensores presentan un encapsulado de plástico similar. Podemos distinguir ambos modelos por el color del mismo. El DHT11 presenta una carcasa azul, mientras que en el caso del sensor DHT22 el exterior es blanco.

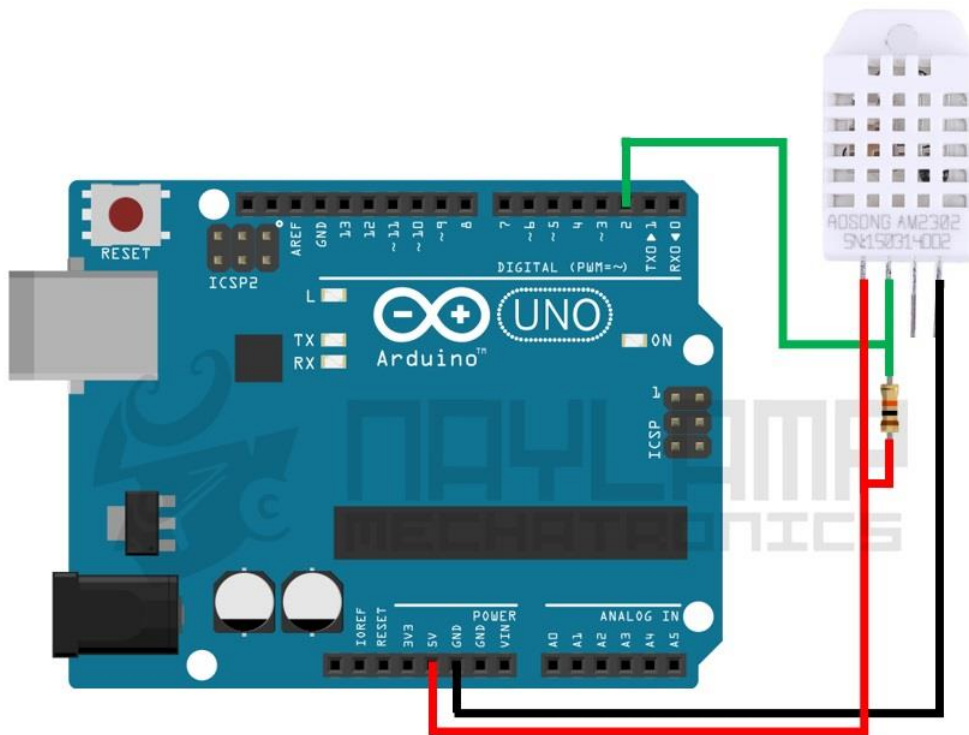
Las características del DHT11 son realmente escasas, especialmente en rango de medición y precisión.

- Medición de temperatura entre 0 a 50, con una precisión de 2°C
- Medición de humedad entre 20 a 80%, con precisión del 5%.
- Frecuencia de muestreo de 1 muestras por segundo (1 Hz)

Por el contrario, el modelo DHT22 tiene unas características mucho más aceptables.

- Medición de temperatura entre -40 a 125, con una precisión de 0.5°C
- Medición de humedad entre 0 a 100%, con precisión del 2-5%.
- Frecuencia de muestreo de 2 muestras por segundo (2 Hz)

## CONEXIONADO



# PARKING

## PROGRAMACIÓN

```
#include <DHT11.h> //DHT11 by Dhruba Saha
DHT11 dht11(3); // digo en que pin esta el sensor DHT11

void loop() {

    int result = dht11.readTemperatureHumidity(temperatura, humedad); // en
    la variable temperatura guarda la temperatura y en la variable humedad la
    humedad. Las dos variables las tengo que crear antes como int

    delay(2000); //espera entre toma de medidas, es el tiempo mínimo que el
    sensor necesita para tomar una medida (2 segundos) para esta espera os
    enseñare una forma para que el programa siga corriendo y no este parado 2
    segundos

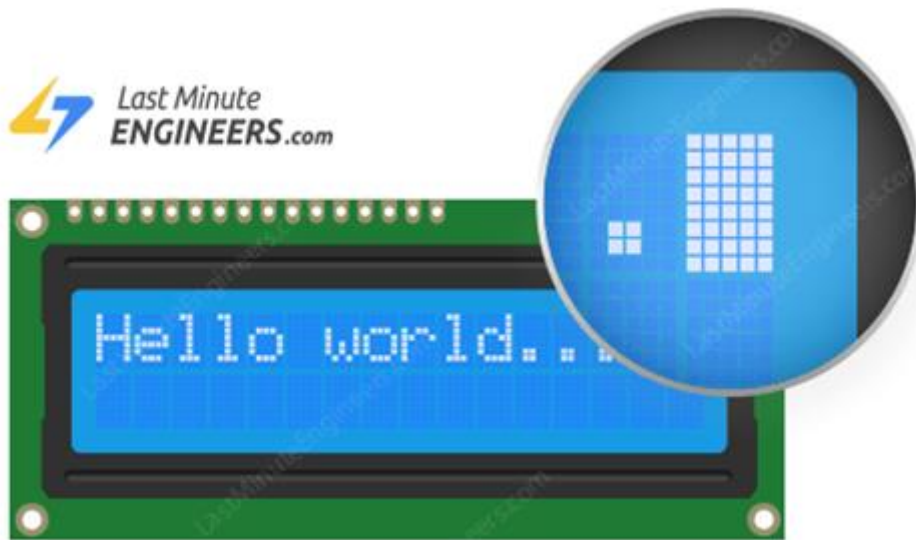
}
```

# PARKING

## PANTALLA LDC I2C

Una pantalla LCD I2C típica consiste en una pantalla LCD de caracteres basada en HD44780 y un adaptador LCD I2C. Vamos a conocerlos uno por uno.

Fieles a su nombre, estas pantallas LCD son ideales para mostrar solo texto/caracteres. Una pantalla LCD de 16 x 2 caracteres, por ejemplo, tiene retroiluminación LED y puede mostrar 32 en dos filas de 16 caracteres cada una.

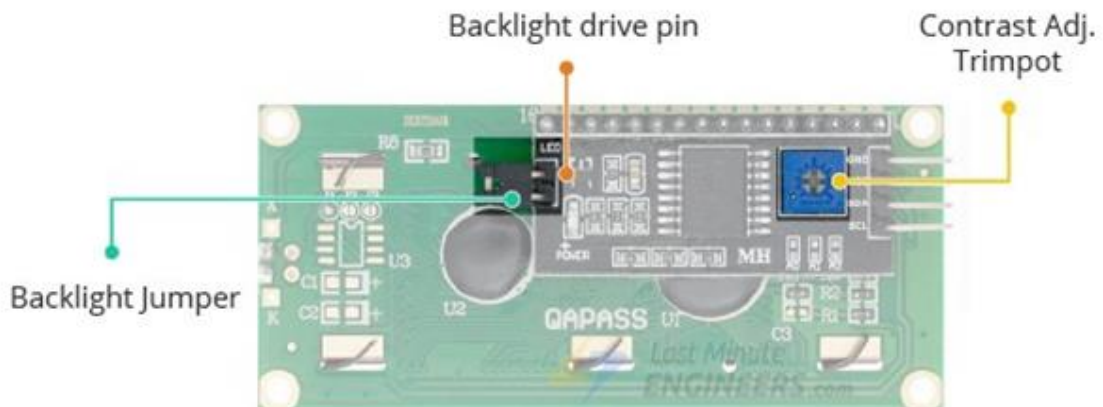


## ADAPTADOR I2C

En el corazón del adaptador hay un chip expensor de E/S de 8 bits: PCF8574. Este chip convierte los datos I2C de un Arduino en los datos paralelos necesarios para una pantalla LCD.

La placa también viene con un pequeño potenciómetro para realizar ajustes precisos en el contraste de la pantalla.

Además, hay un puente en la placa que suministra energía a la luz de fondo. Para controlar la intensidad de la luz de fondo, puedes quitar el puente y aplicar voltaje externo al pin del encabezado que está marcado como 'LED'



# PARKING

Para seleccionar un dispositivo I2C es necesario poner la dirección de ese dispositivo, en el caso de **nuestra LCD la dirección es 0x27**.

## PINES



I2C LCD Pinout

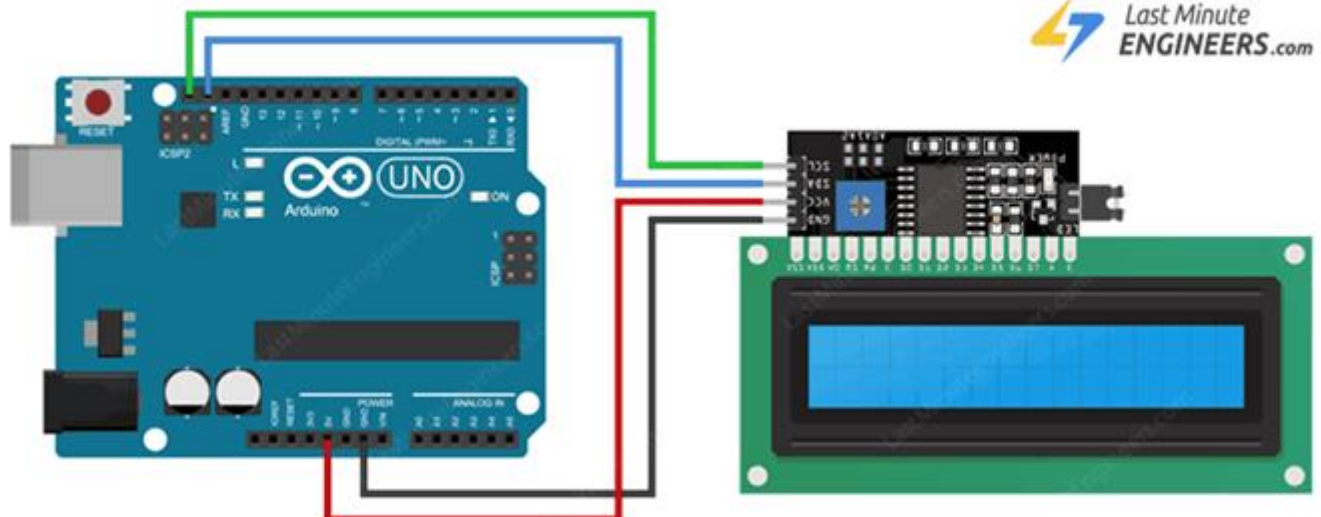


**GND** es un pin de tierra. Conéctalo a tierra del Arduino.

**VCC** suministra energía al módulo y al LCD. Conéctalo a la salida de 5V de Arduino o a una fuente de alimentación externa de 5V.

**SDA** es el pin de datos I2C. Conéctalo al pin de datos I2C de Arduino.

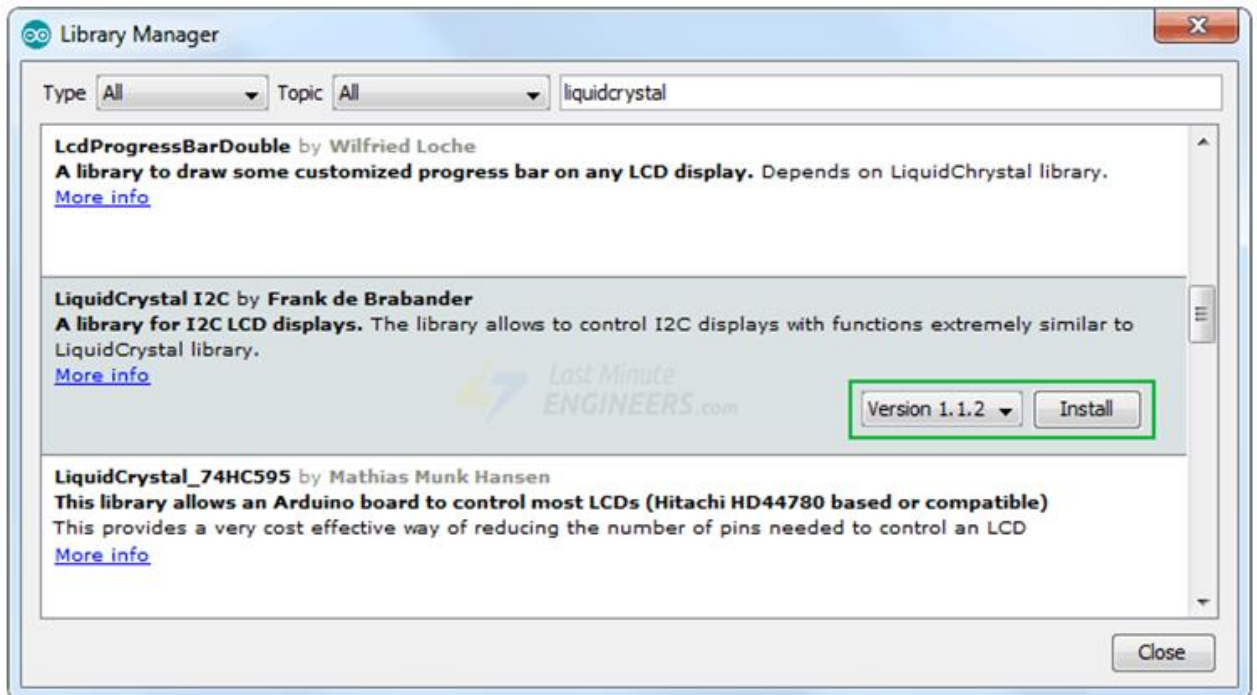
**SCL** es el pin del reloj I2C. Conéctalo al pin de reloj I2C de Arduino.



# PARKING

## PROGRAMACIÓN

### Librería



## PROGRAMA

```
#include <LiquidCrystal_I2C.h> // incluyo la librería

LiquidCrystal_I2C lcd(0x27,20,4); // creo el objeto lcd, Primer número dirección de la LCD en
nuestro caso 0x27, segundo número es el número de caracteres en nuestro caso 20 y el segundo
el número de líneas en nuestro caso 4 líneas

void setup() {
  lcd.init(); // inicializo el objeto lcd
  lcd.clear(); // limpio la LCD, quito todo lo que estuviese escrito
  lcd.backlight(); // pongo la luz de fondo de la lcd
}

void loop() {
}
```

Lo anterior es imprescindible para la inicialización de la LCD, ahora vamos a ver las ordenes que tiene para poner el texto

**lcd.clear()** Borra la pantalla y posiciona el cursor en la esquina superior izquierda (0,0).  
**lcd.setCursor(columna, fila)** Posiciona el cursor del LCD en la posición indicada por columna y fila. Recordar la primera fila es la 0 y la primera columna es la cero  
**lcd.print("texto")** Escribe el texto a partir de la posición indicada en la orden anterior  
**lcd.backlight()** Enciende la Luz del Fondo del LCD  
**lcd.noBacklight();** Apaga la Luz del Fondo del LCD

# PARKING

## MATERIALES

U1	1	Arduino Uno R3
U2	1	Sensor de temperatura [DHT11]
R1	1	10 k $\Omega$ Resistencia
R2	1	Fotorresistencia
D1 D2 D3 D4	4	Blanco LED
R4	1	50 $\Omega$ Resistencia
SERV01 SERVO2	2	Microservomotor
P1	1	6v Fuente de alimentación
U3	1	LCD 20 x 4 (I2C) (0x27)
T1	1	Transistor NPN (BJT) BD135
R5	1	200 $\Omega$ Resistencia
U5 U6 U7 U8	4	Sensor IR
Rpot1	1	10 k $\Omega$ Potenciómetro