

A digital temperature and humidity sensor with a black cable, a multi-colored wire bundle, and a cylindrical metal probe mounted on a beige plastic base. The probe has a label with the number "DB130" and "04093922". The base has the word "DABEC" embossed on it.

MANUAL DE USUARIO

Sensor de temperatura y humedad digital

MCI-SEN-01561 | REV. 1.0

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 of. 1105, Providencia, Santiago, Chile.
+56 2 23339579 | www.olimex.cl | info@olimex.cl

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 Oficina 1105
Providencia, Santiago, Chile

www.olimex.cl
info@olimex.cl

Tel: +56 2 23339579
Fax: +56 2 23350589

® MCI Ltda. 2016

Atención: *cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía.*

Código Manual: MCI-MA-1027

CONTENIDO

INTRODUCCIÓN.....	4
MATERIALES NECESARIOS.....	4
PINES DE CONEXIÓN.....	5
ESQUEMA DE CONEXIÓN	6
PROGRAMACIÓN DE ARDUINO	8
EXPLICACIÓN DE CÓDIGO ARDUINO	9
HISTORIA DEL DOCUMENTO	10

INTRODUCCIÓN

En este manual explicaremos como utilizar el sensor de temperatura y humedad digital. Veremos sus pines de conexión, el significado de cada color y como realizar un circuito básico junto a Arduino Uno.

Nota: El Pin-Out de nuestro producto con código [SKU: MCI-SEN-01852](#) es distinto al mencionado en este manual. Para este modelo de sensor utilizar el siguiente Pin-Out:

- Rojo - 5v
- Verde - GND
- Azul - Data wire
- Amarillo - Clock wire

Antes de realizar las conexiones del manual de usuario, revisar el Pin-Out anteriormente mencionado para poder compatibilizar este sensor con el manual de usuario.

MATERIALES NECESARIOS

Los materiales necesarios para realizar el siguiente ejemplo son:

- ✓ Sensor de temperatura y humedad digital [MCI-SEN-01561](#).
- ✓ Arduino Uno [MCI-TDD-00756](#) (puede ser cualquier Arduino).
- ✓ Resistencia de 10Kohm.
- ✓ Protoboard [MCI-PRT-00170](#).
- ✓ Cables Jumper [MCI-ACC-00872](#).

Para este ejemplo en específico se ha utilizado Arduino Uno, pero no es necesario que utilices el mismo dispositivo. Puedes utilizar un Arduino Mega, Leonardo, etc.

Los programas necesarios para realizar este ejemplo son:

1. IDE Arduino, el cual se puede descargar desde este link de [aquí](#).
2. La librería SHT1X que puede ser descargada desde el siguiente [link](#).
3. El código fuente de programación de Arduino estará contenido en un archivo RAR, que contendrá el Sketch de Arduino y también la librería SHT1X. Este archivo RAR puede ser descargado desde este link de [aquí](#).

PINES DE CONEXIÓN

El sensor de temperatura posee 4 cables en total. Cada cable posee un color y un significado que se puede ver en la siguiente tabla:

Pin	Nombre
1 Amarillo	SCK (ClockPin)
2 Rojo	VCC (+5V)
3 Azul	GND
4 Negro	Data (DataPin)

Los cables 2 y 3 correspondientes a los colores rojo y azul son los cables de alimentación del sensor. La alimentación debe ser de 5VDC.

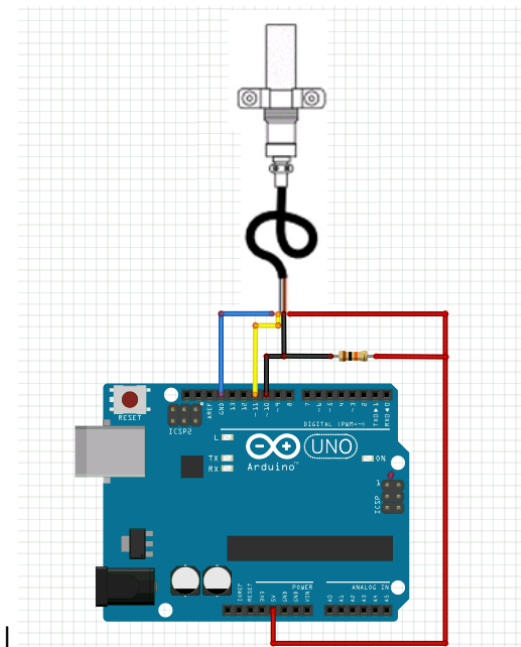
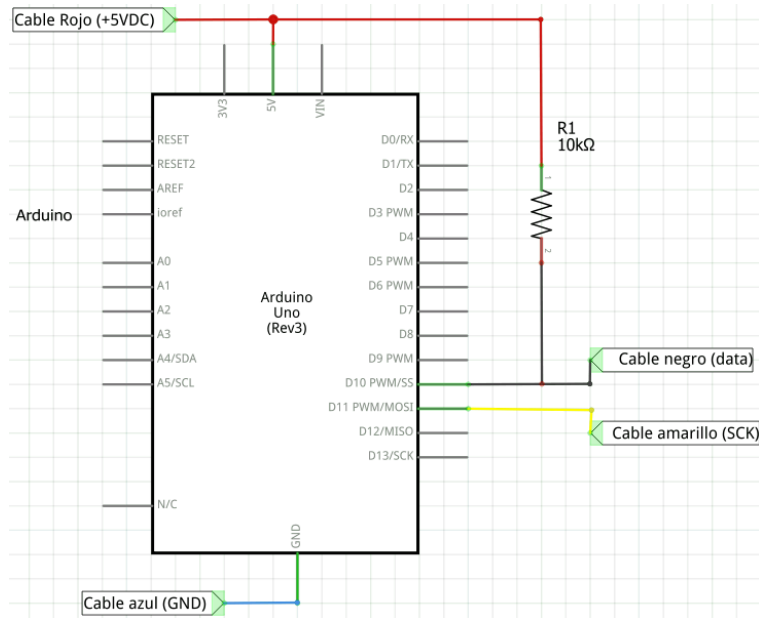
Los cables 1 y 4 correspondientes a los colores amarillo y negro son los cables de comunicación entre sensor y Arduino.

El cable 1, de color amarillo, corresponde a la señal SCK y su función es marcar el pulso para la sincronización de dispositivos. Con cada pulso de este reloj, se lee o se envía un bit.

El cable 4, de color negro, corresponde a la señal de datos (data) y su función es transmitir los datos de temperatura y humedad hasta otro dispositivo, en nuestro caso será Arduino.

ESQUEMA DE CONEXIÓN

Las siguientes imágenes muestran la conexión del sensor de temperatura y humedad con Arduino:

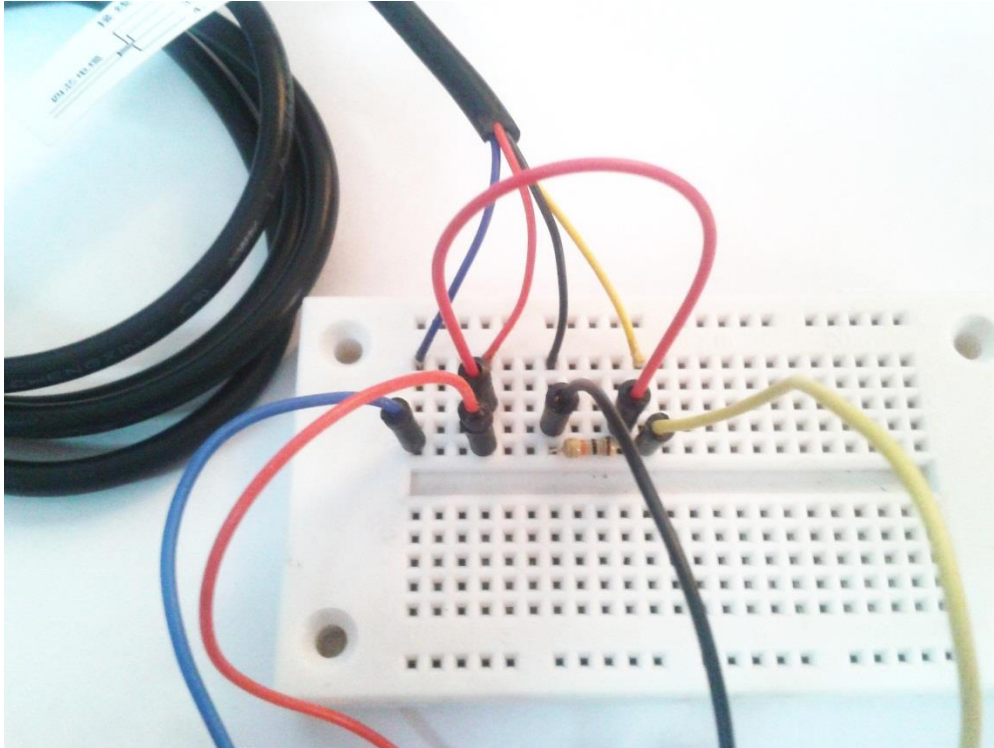


Para este ejemplo, los pines usados de Arduino son:

- +5V: usado con cable rojo.
- GND: usado con cable azul.
- Digital 10: usado con cable negro.
- Digital11: usado con cable amarillo.

Como se muestra en las figuras anteriores se debe utilizar una resistencia pull-up de 10Kohm conectada en paralelo entre la alimentación +5VDC y el pin de datos (pin 10 de Arduino).

Este circuito se puede montar en un protoboard como se muestra en la siguiente imagen:



PROGRAMACIÓN DE ARDUINO

Para ejecutar el programa debemos de copiar y pegar la librería del sensor SHT1X en la carpeta "Libraries" de Arduino.

Luego, basta con abrir el archivo con extensión *.ino* contenido en la carpeta descargada, y seleccionar la tarjeta Arduino que estemos utilizando junto a su puerto COM. Cargamos el sketch en la tarjeta y abrimos el monitor serial de Arduino; en esta consola que se ha abierto podemos observar que Arduino está enviando por su puerta serial los valores que obtiene del sensor de temperatura y humedad digital.

Si observamos la programación contenida en el sketch de Arduino podemos observar que los pines 10 y 11 están configurados por software, por lo tanto podemos cambiarlos a los pines que más nos acomoden o utilizar los pines que tengamos disponibles.

El programa al cual nos referimos es el siguiente:

```
#include <SHT1x.h>//Se incluye la libreria SHT1X.

#define dataPin 10 //Se define el pin 10 de Arduino para entrada de dato.
#define clockPin 11 //Se define el pin 11 de Arduino para sincronización reloj.
SHT1x sht1x(dataPin, clockPin); //Se inician los pines para utilizar el sensor.

void setup() {
    Serial.begin(9600); //Se inicia la comunicación Serial
    Serial.println("Programa en ejecución: ");
}

void loop() { //Se crean variables decimales para guardar valores de sensores.
    floattemp_c;
    floattemp_f;
    float humedad;

    //Lee los valores de los sensores y los guarda en variables.
    temp_c = sht1x.readTemperatureC();
    temp_f = sht1x.readTemperatureF();
    humedad = sht1x.readHumidity();
```



```
// Envía por la puerta serial los valores.  
Serial.print("Temperatura: ");  
Serial.print(temp_c);  
Serial.print(" grados Celsius / ");  
Serial.print(temp_f);  
Serial.println(" grados Fahrenheit");  
Serial.print("Humedad relativa: ");  
Serial.print(humedad);  
Serial.println("%");  
delay(2000);  
}
```

EXPLICACIÓN DE CÓDIGO ARDUINO

El código de programación anterior es el encargado de controlar las funciones de Arduino y leer los datos provenientes del sensor de temperatura y humedad digital.

En un orden secuencial lo que este código está haciendo es:

1. Incluyendo la librería del sensor SHT1X, la cual contiene la información básica de nuestro sensor.
2. Configura por cual pin de Arduino se sincroniza el sensor y cual pin transmitirá datos.
3. Se da inicio a la configuración del void setup, en el cual se inicializa la comunicación Serial y se envía un mensaje por esta puerta con un mensaje de inicio del programa.
4. Se crea un void loop para indicar a Arduino lo que debe hacer mientras el equipo este encendido.
5. Dentro de void loop se crean nuevas variables que contendrán los valores obtenidos desde el sensor de temperatura y humedad digital.
6. Se crean instrucciones para enviar por la puerta serial los valores contenidos en las variables de almacenamiento de datos que se obtienen desde el sensor.
7. Finalmente se crea la instrucción de retardo (delay) de 2 segundos. Esta instrucción se crea con el fin de poder visualizar de mejor manera la información del sensor en la pantalla. En otras palabras se le dice a Arduino que espere 2 segundos para iniciar nuevamente el ciclo contenido en void loop.

HISTORIA DEL DOCUMENTO

Revisión	Fecha	Editado por	Descripción/Cambios
1.1	23 de Junio de 2015	Patricio Peña	Segunda versión de manual