Arduino Inicial

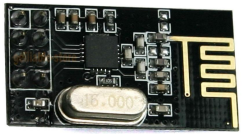
Ing. Juan C. Abdala

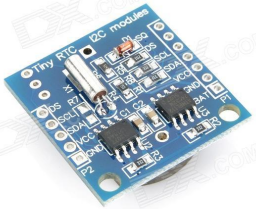
Clase 3

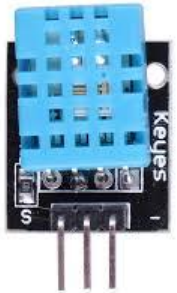
Objetivo de esta clase

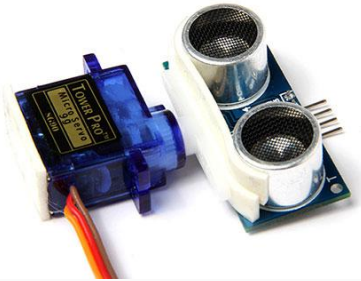
- Aprender sobre los periféricos de Arduino.
- Aprender qué son y cómo se usan los buses de comunicación.
- Aprender qué es y como se descarga una librería.

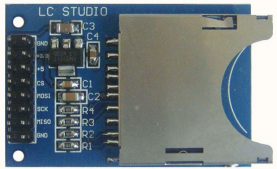
Perifericos



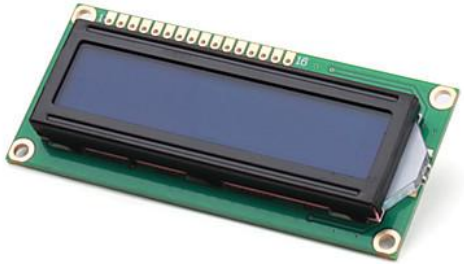








SD Card Reader Module



Perifericos

¿Como se usan?

Librerias.

- Son junto a la cantidad de periféricos la parte más importante de arduino.

- Existen las oficiales (página de arduino) y miles como Software Libre.

Nuestro caso.

1. Descargar
2. Descomprimir
3. Colocar en arduino/libraries/
4. Agregarla con #include <”Nombre de Librería”>

Tipo de Periféricos.

Analogicos: Estos reciben alimentacion y devuelven valores en mV(mili Volt) que deben ser traducidos.

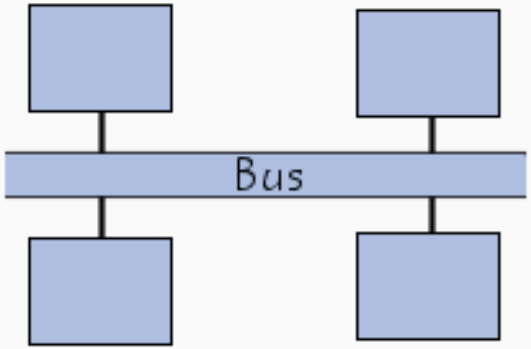
Digitales: Nos devuelven el valor procesado solo debemos tomarlo.

Perifericos

¿Pero cómo se conectan?

Bus de comunicacion

El bus (o canal) es un sistema digital que transfiere datos entre los componentes de una computadora o entre varias computadoras.



Bus Arduino UNO

1. **SPI**
2. **I2C**
3. **Serial.**
4. **Onewire**

SPI

- SPI (Serial Peripheral Interface)
- Soportado por hardware.
- Modo Maestros esclavos.

SPI

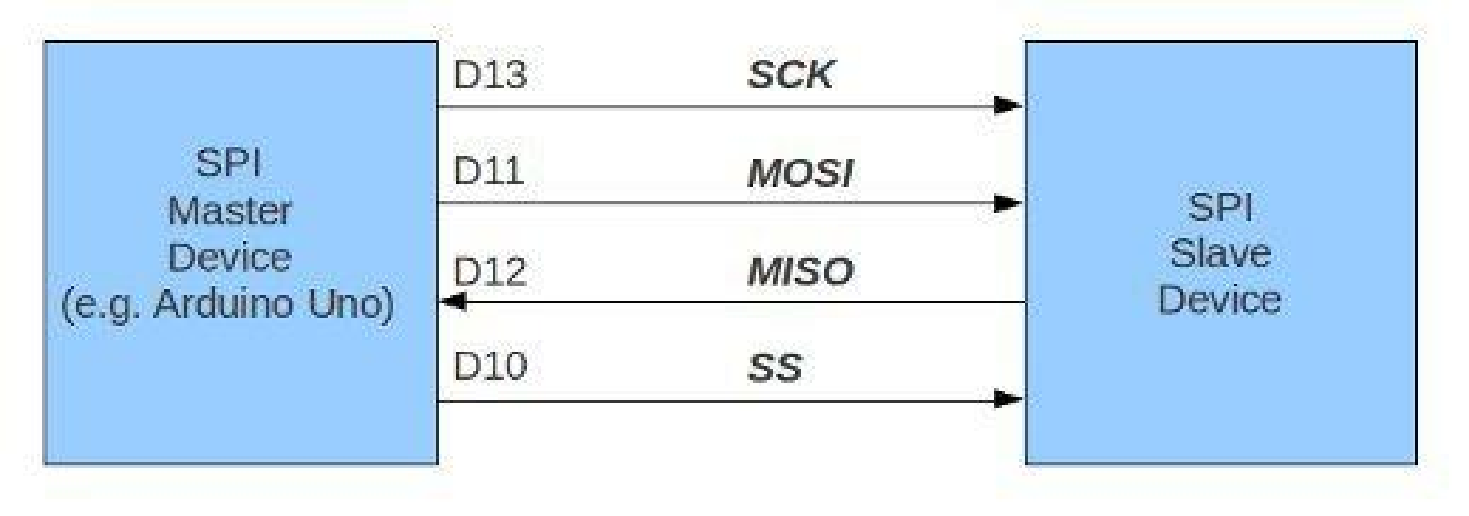
SCLK: Es la señal de reloj, impuesta por el dispositivo maestro.

MOSI: Corresponde a las siglas “Master Output – Slave Input”, es decir, el maestro enviará los datos a través de esta línea y el esclavo los recibirá.

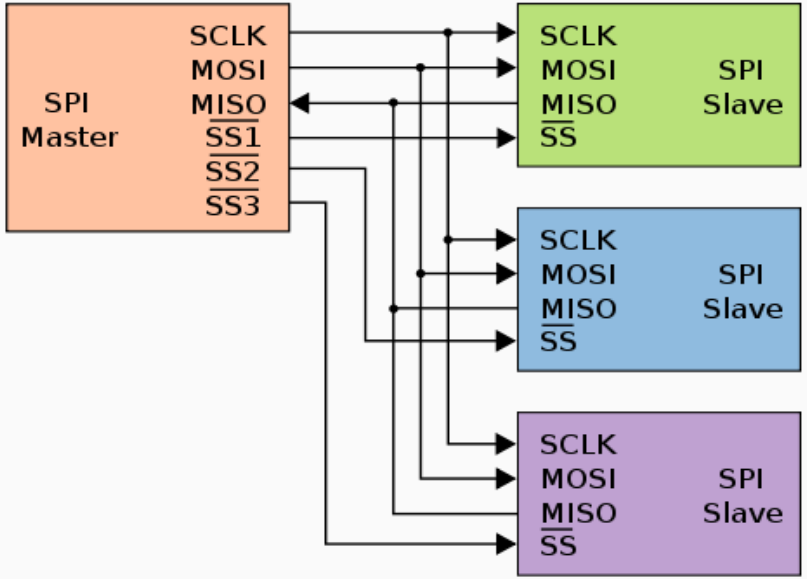
MISO: Corresponde a las siglas “Master Input – Slave Output”, y es la línea por la que los esclavos enviarán datos al dispositivo maestro.

SS: Es la señal de “Slave Select”, es decir, la línea que el maestro activará para indicar al esclavo que se va a establecer la comunicación con él.

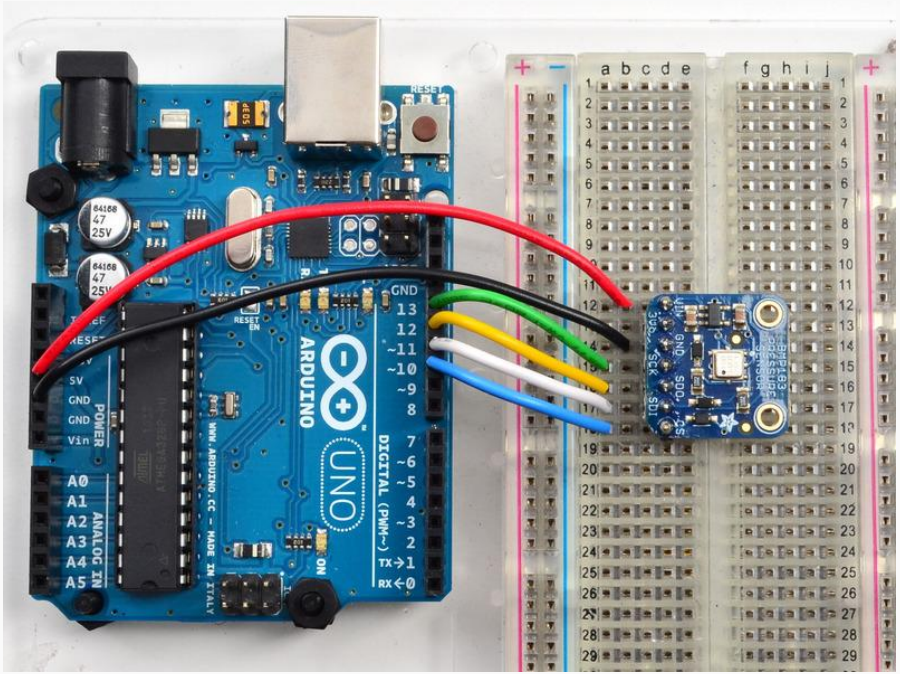
SPI


SPI
Master
Device
(e.g. Arduino Uno)
D13
SCK
D11
MOSI
D12
MISO
D10
SS
SPI
Slave
Device

SPI


SPI
Master
SCLK
MOSI
MISO
SS1
SS2
SS3
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave

Ejemplo Bmp183 Presión



Ejemplo Bmp183 Presión

```
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP183.h>

#define BMP183_CS 10
Adafruit_BMP183 bmp =Adafruit_BMP183(BMP183_CS);

void setup(void)
{
  Serial.begin(9600);
  if(!bmp.begin()) { Serial.print("No se encuentra el sensor");}
}
```

```
void loop()
{
    Serial.print("Presion = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");
    delay(500);
}
```

I2C

- Inter-Integrated Circuit (Inter-Circuitos Integrados).
- Soportado por hardwares.
- Tipo serial, solo 3 conexiones, pero mas complejo a nivel software.

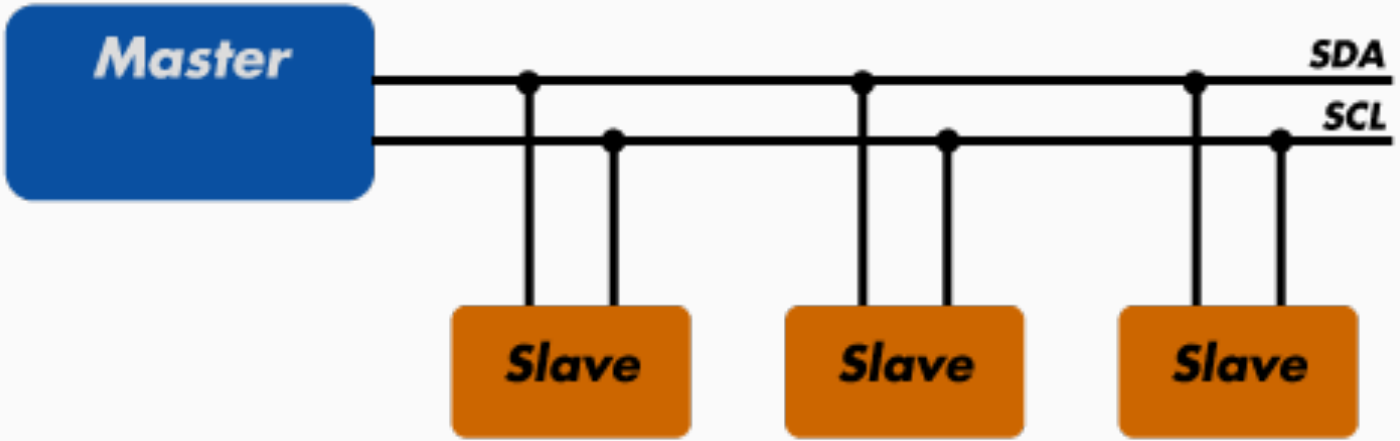
I2C

SDA: Datos

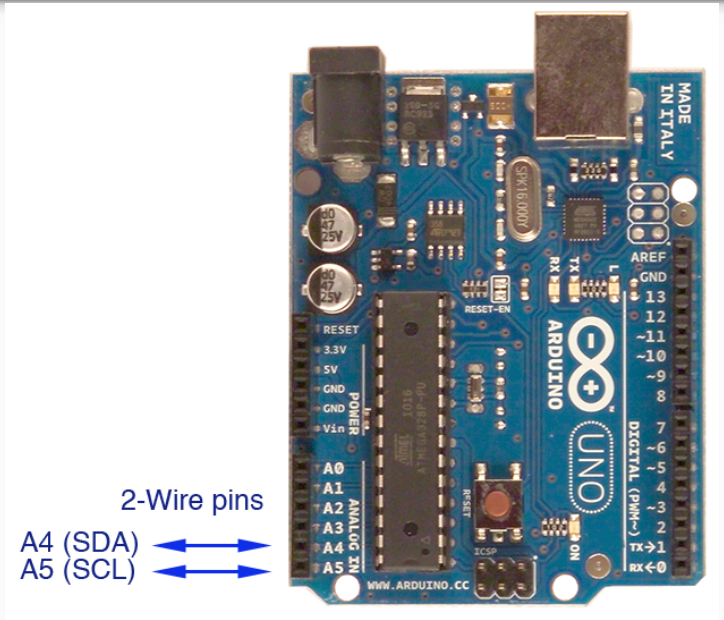
SCL: Reloj

GND: Tierra

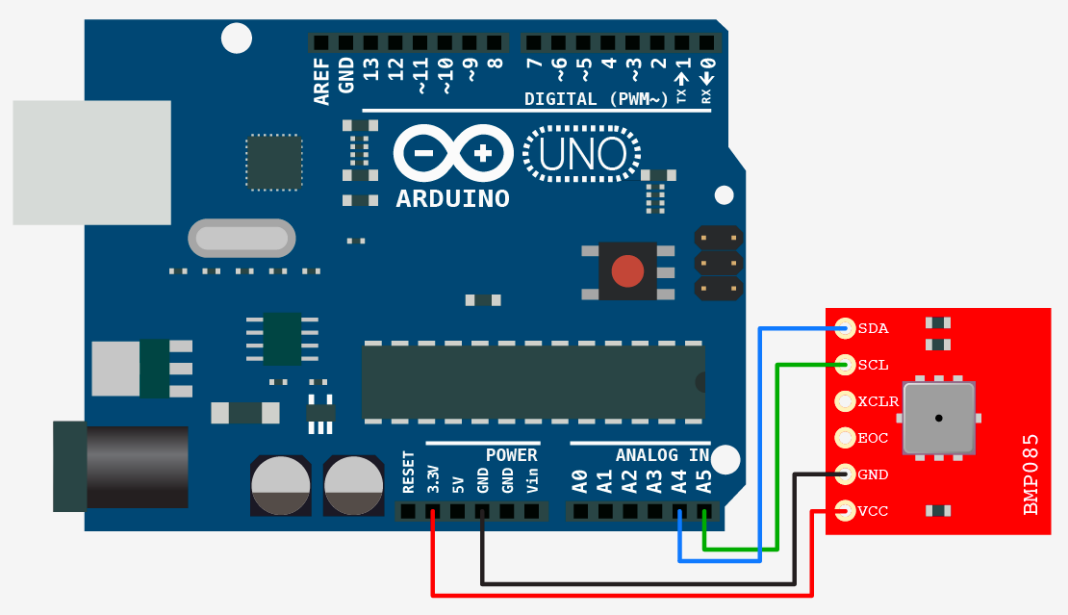
I2C



I2C


2-Wire pins
A4 (SDA)
A5 (SCL)
MADE IN ITALY
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC

Ejemplo Bmp85 - Presión



Ejemplo Bmp85 - Presión

```
#include <Wire.h>
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);
  if (!bmp.begin()) {
        Serial.println("No se encuentra sensor");
        while (1) {}
  }
}
```

```
void loop()
{
    Serial.print("Presion = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");
    delay(500);
}
```

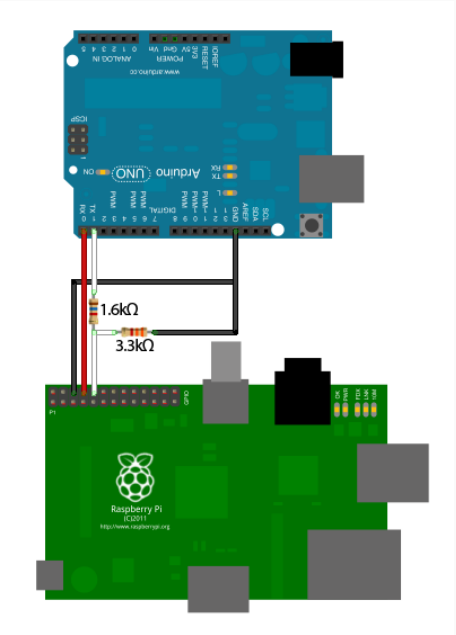
Serial-RS232

- Soportado por hardware y también por software, algunos arduinos poseen varios seriales.

- A nivel hardware se transforma el serial rs232 a USB.

Serial

No solo como debugger,
puede ser el nexo a otra placa.

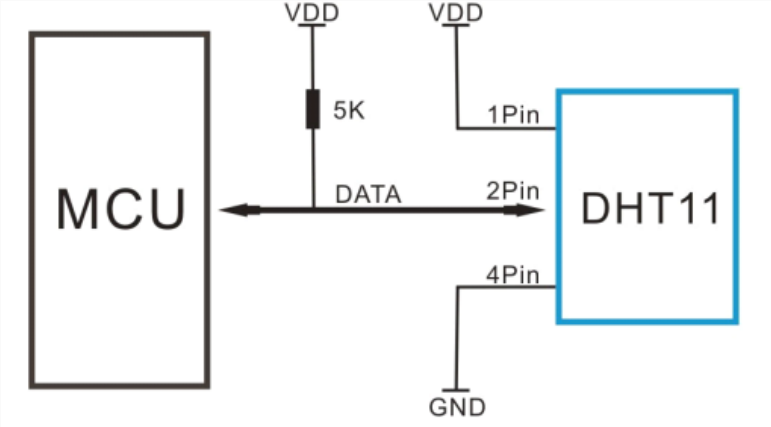


Onewire

- Una sólo conexión, bidireccional.
- Soportados por software, se descarga una librería.
- Muchos periféricos arduinos lo soportan. Ej DHT

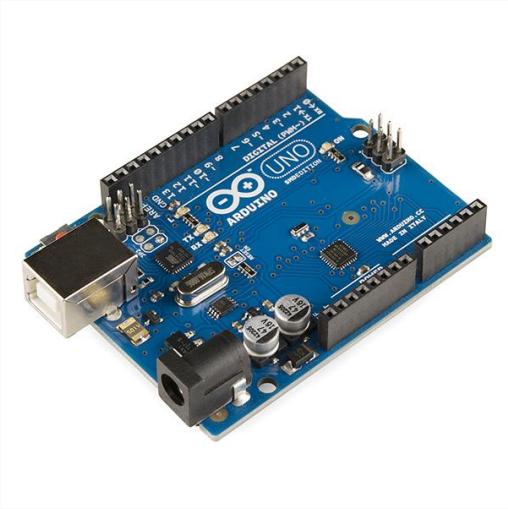
DHT ¿que es?

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

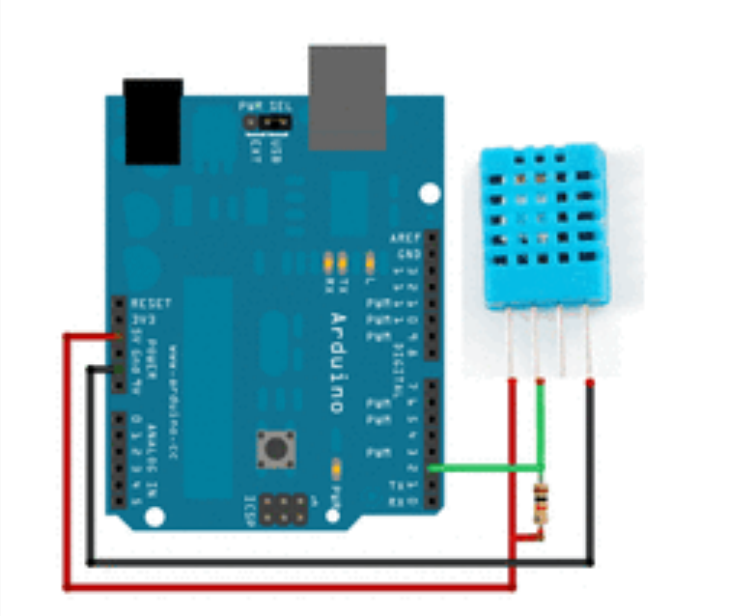


DHT

Arduino + Sensor de Temp. y Humedad



Conexion



DHT LIB

Libreria:

https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib

Se incluye:

#include <dht.h>

Funciones

Inicio:

dht DHT;

Se define pin:

#define DHT11_PIN 5

Funciones

Leer:

DHT.read11(DHT11_PIN);

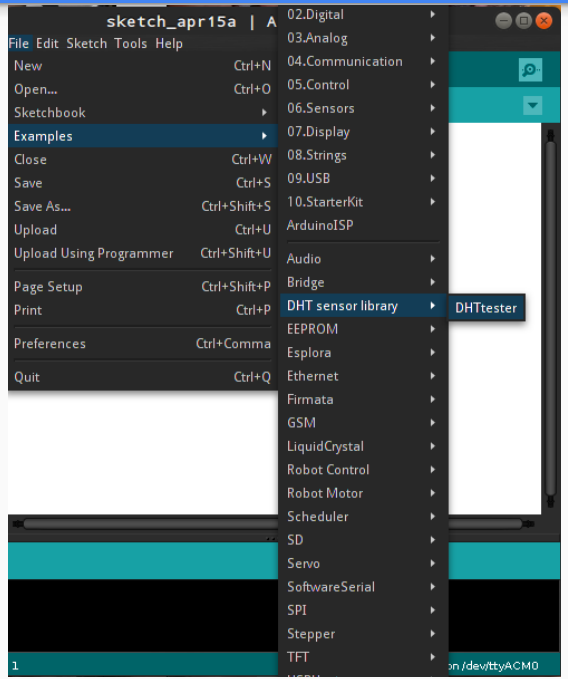
Los datos:

DHT.humidity
DHT.temperature

Otros DHT

Modelo	DHT11	DHT22
Rango de medición de humedad	20-90 % HR	0-100 % HR
Rango de medición de temperatura	0 hasta 50 °C	-40 hasta 80 °C
Precisión de temperatura	±2 °C	±0.5 °C
Precisión de humedad	±5 % HR	±2 % HR

Aprovechemos Examples



Ejemplo

```
#include <dht.h>

dht DHT;

#define DHT11_PIN 5

void setup()
{
  Serial.begin(115200);
  Serial.println("DHT TEST PROGRAM ");

}
```

Libreria

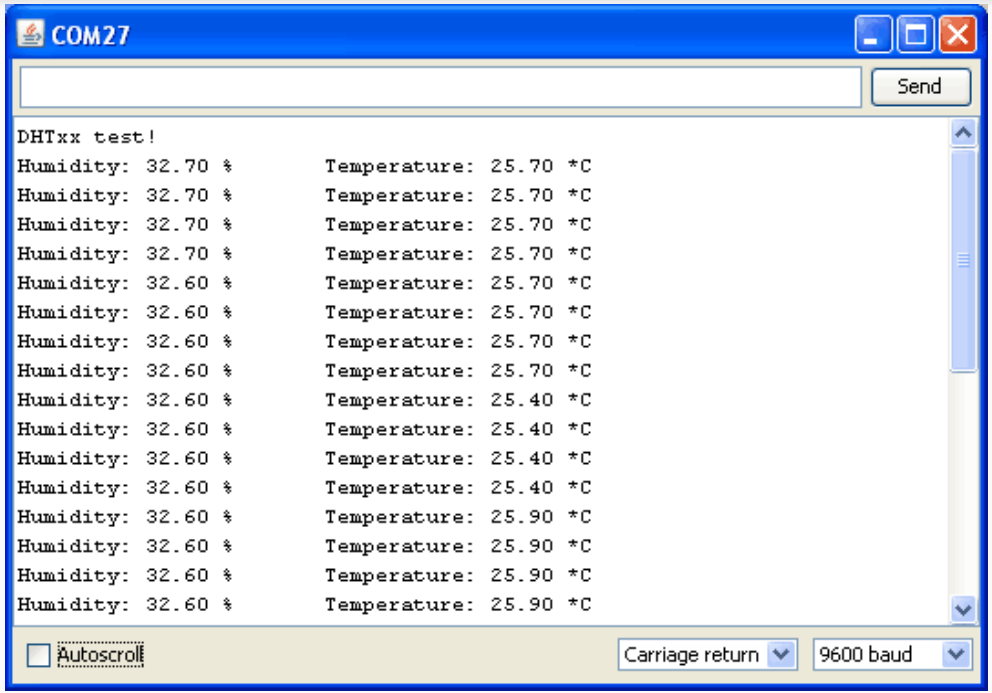
Pin de configuracion

```
void loop()
{
  // READ DATA
  Serial.print("DHT11, \t");
  DHT.read11(DHT11_PIN);

  Serial.print(DHT.humidity, 1);
  Serial.print(",\t");
  Serial.println(DHT.temperature, 1);

  delay(2000);
}
```

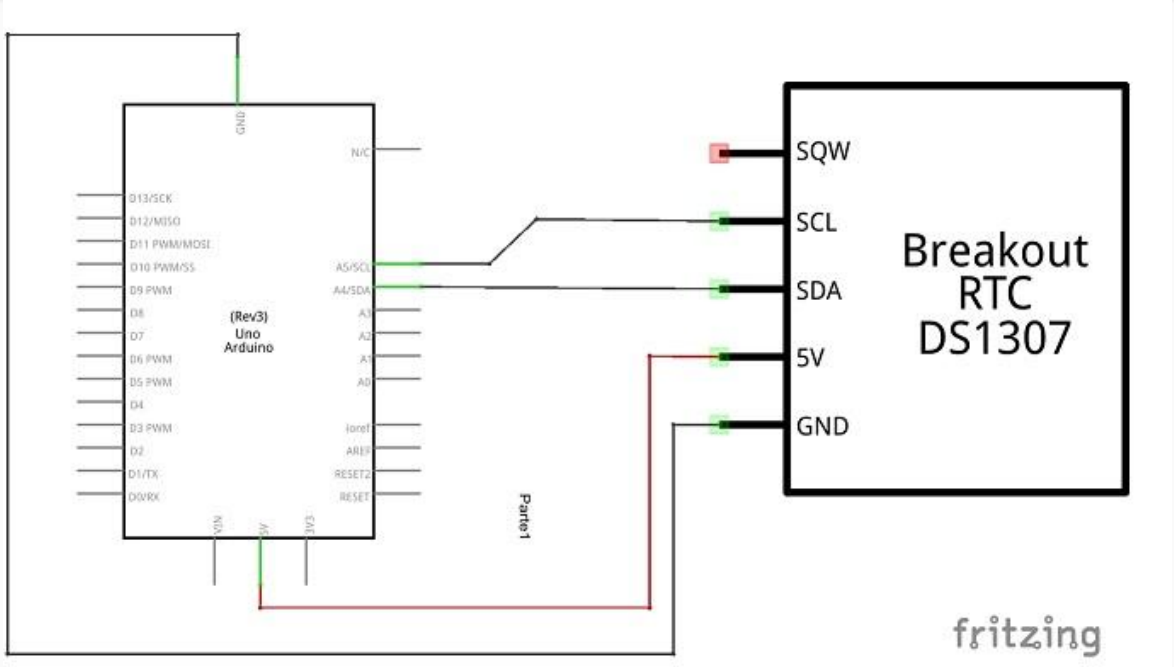
Salida


COM27
Send
DHTxx test!
Humidity: 32.70 % Temperature: 25.70 *C
Humidity: 32.70 % Temperature: 25.70 *C
Humidity: 32.70 % Temperature: 25.70 *C
Humidity: 32.70 % Temperature: 25.70 *C
Humidity: 32.60 % Temperature: 25.70 *C
Humidity: 32.60 % Temperature: 25.70 *C
Humidity: 32.60 % Temperature: 25.70 *C
Humidity: 32.60 % Temperature: 25.70 *C
Humidity: 32.60 % Temperature: 25.40 *C
Humidity: 32.60 % Temperature: 25.40 *C
Humidity: 32.60 % Temperature: 25.40 *C
Humidity: 32.60 % Temperature: 25.40 *C
Humidity: 32.60 % Temperature: 25.90 *C
Humidity: 32.60 % Temperature: 25.90 *C
Humidity: 32.60 % Temperature: 25.90 *C
Humidity: 32.60 % Temperature: 25.90 *C
Autoscroll
Carriage return
9600 baud

RTC

- RTC: Real Time Clock, o reloj de tiempo real. Que lleva la cuenta de segundos minutos y horas además de día mes y año automáticamente, válido hasta el año 2100.
- Una batería exterior, que mantiene la fecha y hora cuando no hay corriente.
- Detección automática de corte de corriente y cambio a modo batería.
- Muy bajo consumo, lo que ayuda a que la batería dure entre 5 y 10 años.
- I2C integrado en el mismo chip.

Esquematico RTC DS1307



RTC Lib

Hay muchas Librerías pero vamos a usar:

http://www.prometec.net/wp-content/uploads/2014/11/DS1307RTC.zip

Después la incluimos:

#include <DS1307RTC.h>

Funciones

setSyncProvider(RTC.get); //podemos definir el modelo de reloj que vamos a usar

setTime(21,46,00,8,11,2014); //para ajustar el reloj ej 21:45:00 del dia 8 de Nov de 2014

Funciones

Se accede por:

hour();

minute();

second();

day()

month()

year();

Variables de Ayuda.

El compilador de Arduino, dispone de un par de argumentos que nos informan de la fecha y hora de la última compilación:

__DATE__ y __TIME__

Ojo SetTime no lo soporta directamente en nuestra libreria.

Descansemos 15 min

Practico

Vamos al Practico 3:

Material: https://github.com/jcabdala/ArduinoInicialUNC

Consultas:

abdalajc@gmail.com

@toniabdala

Google Group: arduino-unc-septiembre@googlegroups.com