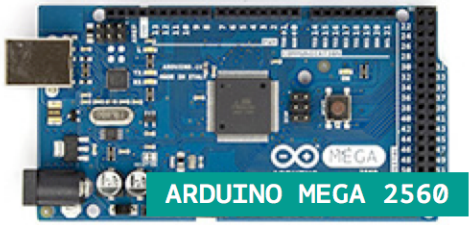
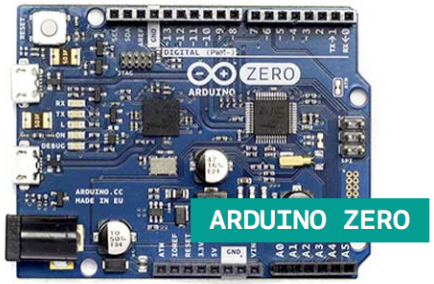
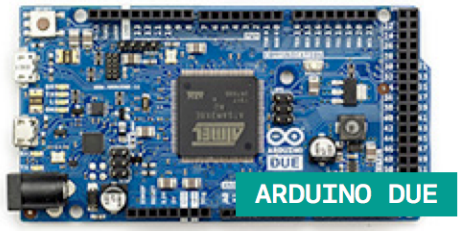
Low-cost LoRa IoT device: supported physical sensors

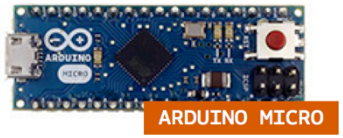
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

Review of WAZIUP IoT platform

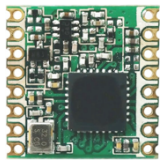








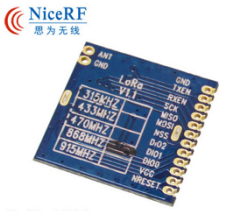
LoRa radios that our library already supports



HopeRF RFM92W/95W

Libelium LoRa

Modtronix inAir9/9B

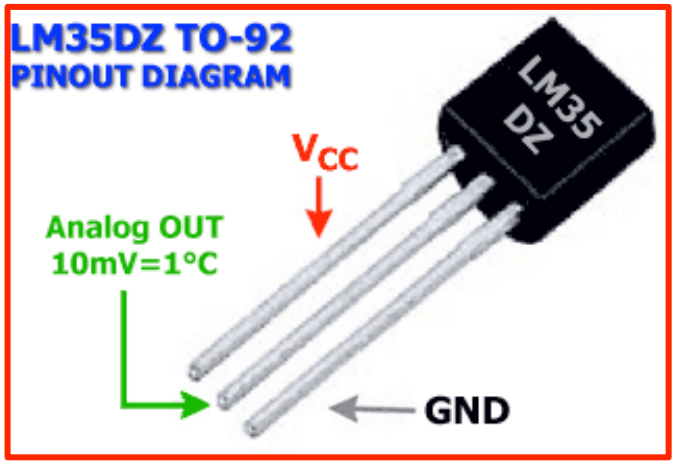


NiceRF LoRa1276

Long-Range communication library
(mostly sending functions)

Connecting a physical sensor

BASIC: ANALOG OUTPUT

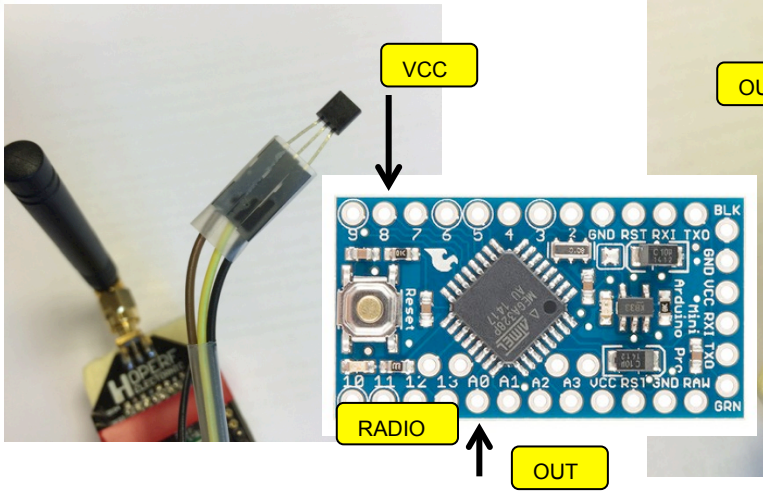


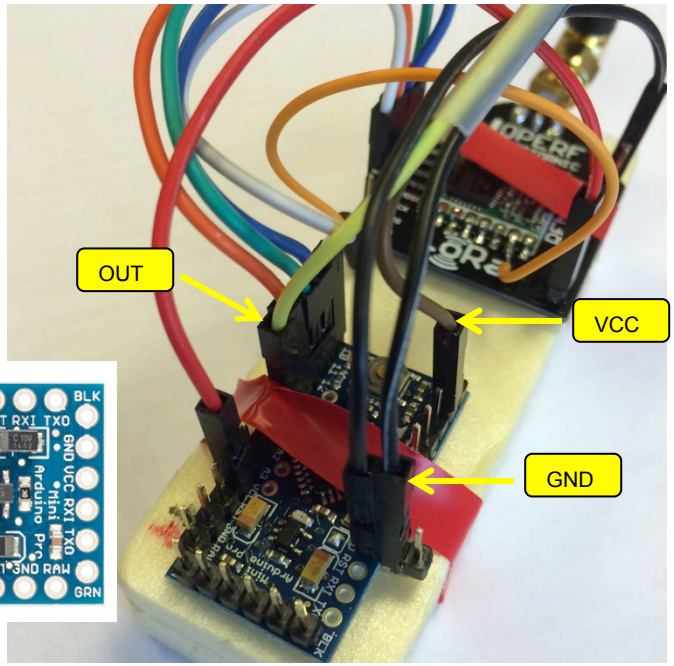
Vcc is 3.3V (the output of analog 8 to power the sensor)

If 0 means 0V and 1024 means 3300mV (10-bit resolution) then 3300mV/1024=3.22mV is the granularity of the measure

A digital value of 100 means 100*3.22mV=322mV
If the sensor output is 10mV/1°C then the physical temperature is 322mV/10mV=32.2°C

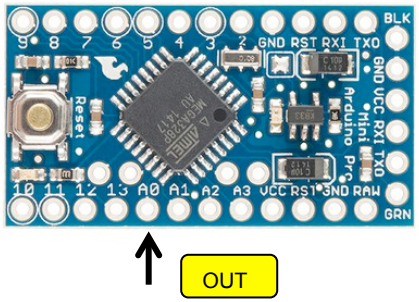
Connecting to board





For the moment, there is no physical sensor connected to the board, so you will probably get random value when running the sensor. The Arduino_LoRa_temp example uses the LM35DZ physical sensor to get the ambient temperature. The GND should be connected to one of the board's GND, the VCC should be connected to the digital pin 8 and the OUT pin should be connected to the analog A0 pin.

READING ANALOG PIN VALUE




```
// sensor output connected to A0 analog pin

value = analogRead(A0);

// now need to convert to Celcius degree
```

And converting into Celcius

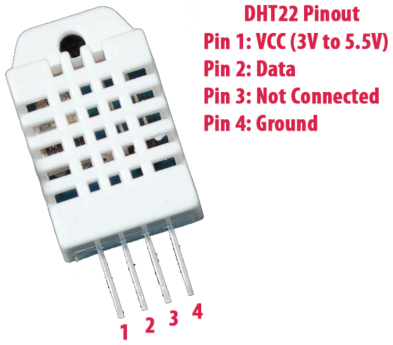
```
Temp = value * 3300.0/1024.0; // 3300/1024=3.22

Temp = Temp / 10;             // 10mV means 1°C

// now process and transmit the data
```

ADVANCED: DIGITAL OUTPUT

- Sensors with a digital output usually provide the sensed data in some advanced digital format, not only an on/off value.
- In that case, additional libraries are often needed to read the sensor's data depending on the communication protocols, e.g OneWire, I2C, SPI,...
- In many cases, there are both library and example provided with the sensor to read the data
- Example with the DHT22 sensor follows. You can search for DHT22 on the web

DHT22 AND DIGITAL OUTPUT

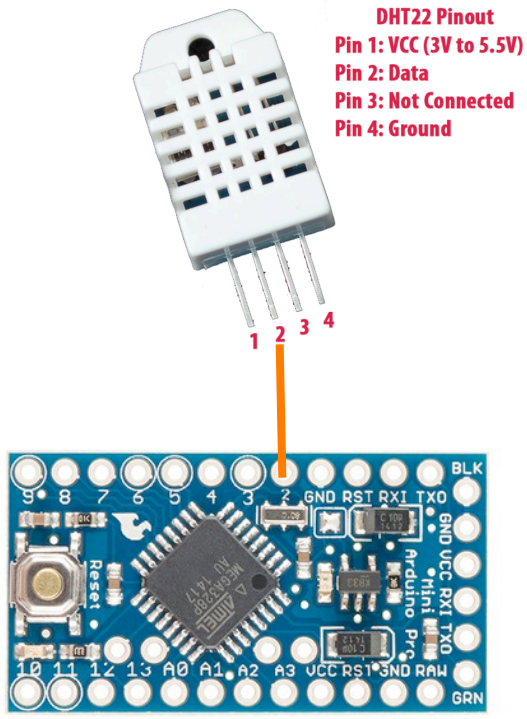
- The DHT22 is a temperature and humidity sensor with a digital output on pin 2



- There are various libraries available (search Arduino+DHT22 on the web)
- We use the library developed by Ben Adams: https://github.com/nethoncho/Arduino-DHT22

EX: READING THE DHT22

```
// include Ben Adams's DHT22 library
#include "DHT22.h"

DHT22* dht = NULL;

// provide the digital pin to which the
// DHT22 data pin is connected to. Here
// digital pin 2 on the Arduino
dht = new DHT22(2);

// start reading
dht->readData();

// then get both temperature and humidity
double temp=(double)dht->getTemperatureC();
double hum=(double)dht->getHumidity();
```

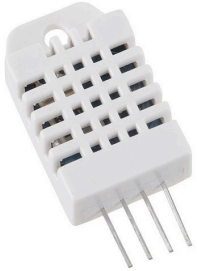


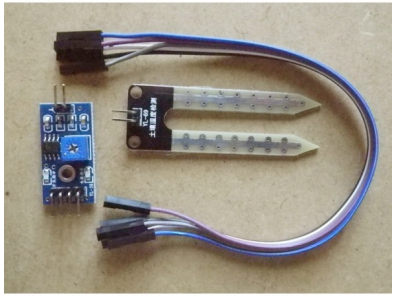
WARNING: we did not represent yet the other needed connections

Generalization

- Depending on the sensor type, getting the physical measure from the analog/digital value follows a specific function/library provided by the sensor's manufacturer or other contributors
- Depending on the microcontroller board, the number of I/O pins and the operating voltage may differ
- However the process is always the same:
  - Connect the sensor to the microcontroller board
  - Read analog or digital pin possibly using dedicated libraries
  - Convert read value into meaningful physical measure
  - Then process and/or transmit

Supported sensors





Supported sensors

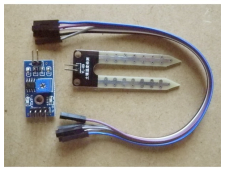
- We currently provide all the necessary code for a number of sensors to serve as examples:
  - Analog LM35DZ temperature sensor

  - Digital temperature/humidity DHT22

  - Digital DS18B20 temperature sensor

  - Analog LeafWetness sensor

  - Analog soil humidity sensor



  - More to come

Sensor preparation

- Many sensors need a little preparation before you can safely connect them to a microcontroller board
- Usually, you may need to add some resistor and/ or condensator to some pins or wires
- Consult the sensor datasheet or search the web on the required schematic for a safe and proper operation of the sensor

Ex: DHT22

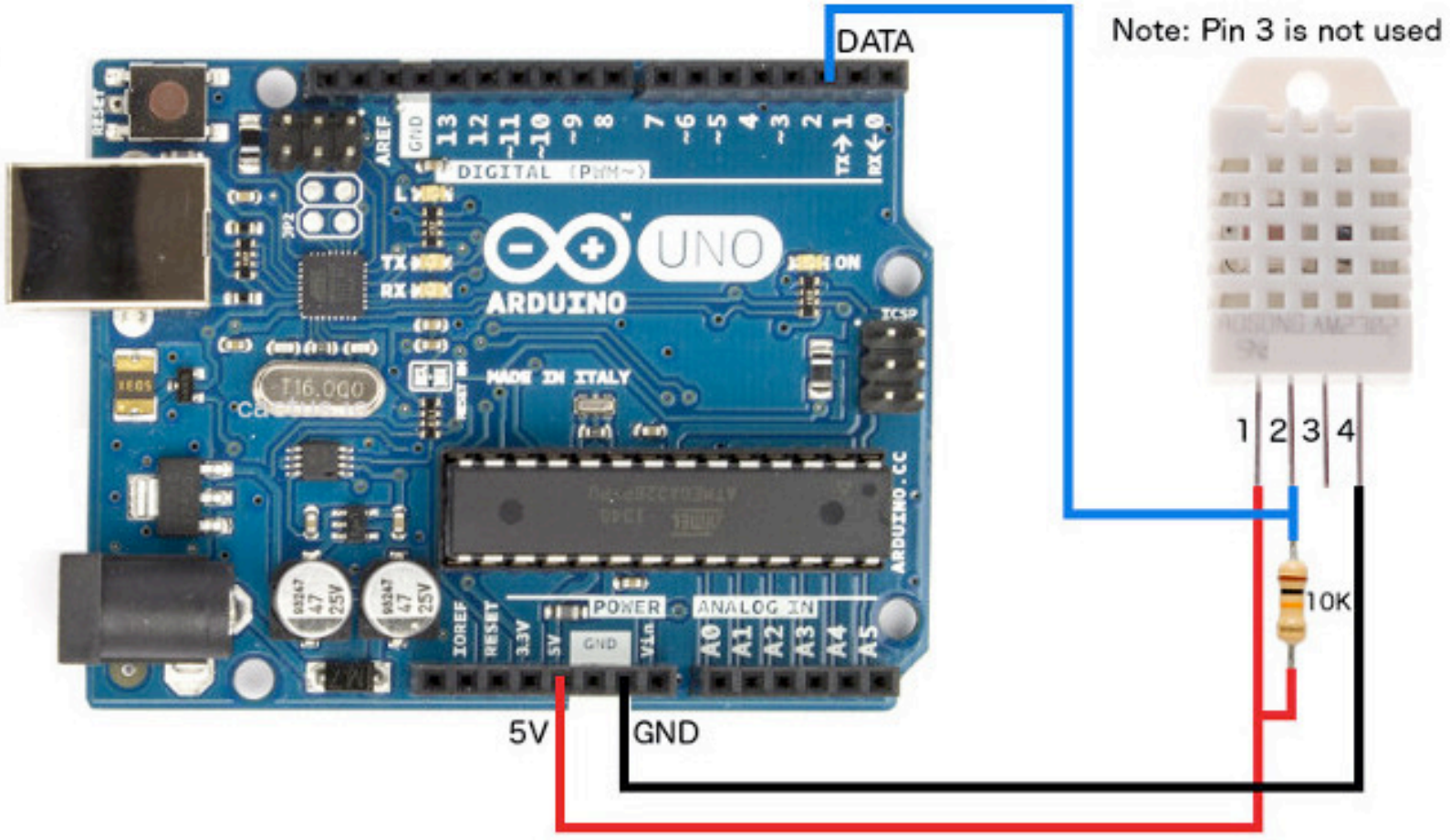


Image from http://cactus.io/hookups/sensors/temperature-humidity/dht22/hookup-arduino-to-dht22-temp-humidity-sensor

Ex: DS18B20

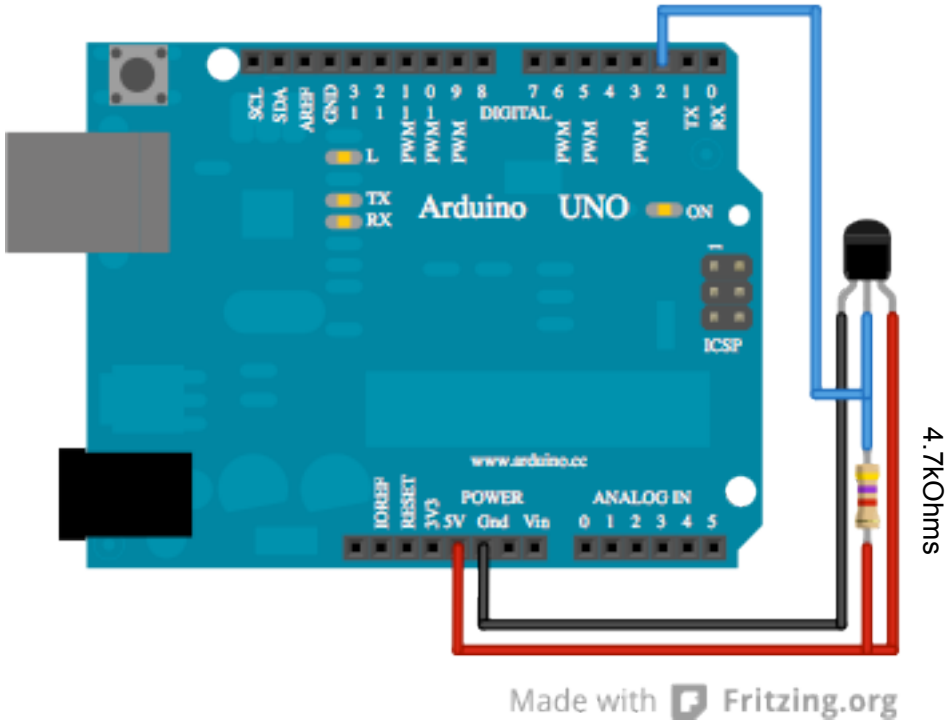
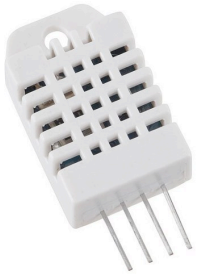
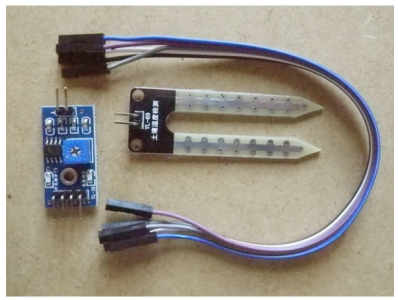


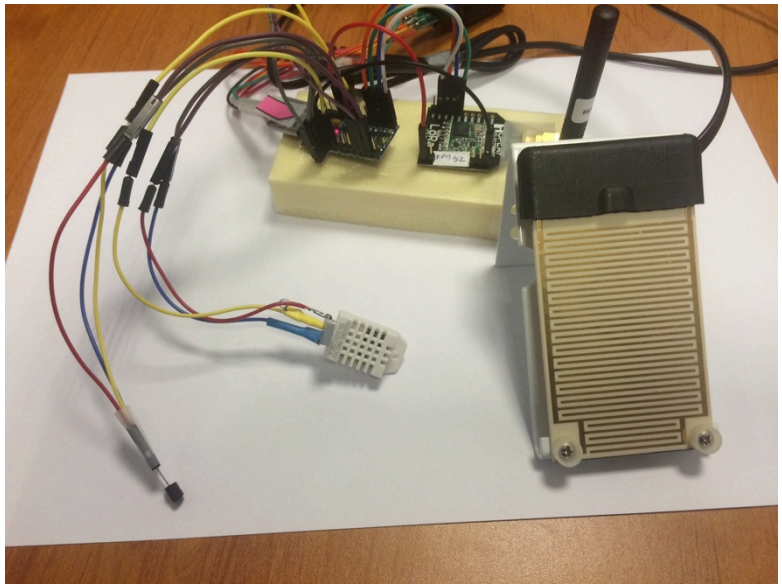
Image from http://fun-tech.se/FunTechHouse/RoomTemperature/

Generic Sensor class & adding a new sensor





Several physical sensors per device



Physical sensor will be derived from `Sensor` base class and offers a unique `get_value()` function that will be called periodically

A sensor with several physical sensor will simply loop over all the connected sensors to call the `get_value()` function

Several physical sensors can be connected to a board. Currently supported sensors:

- All simple analog sensors (value on 1 analog wire: e.g. LM35Z temp. sensor or Davis leaf wetness Vantage)
- All simple digital sensors (binary value low/high)
- Most of one-wire digital sensor such as DHT22 sensor

For specific sensors, `get_value()` will simply use specific provided/developped library