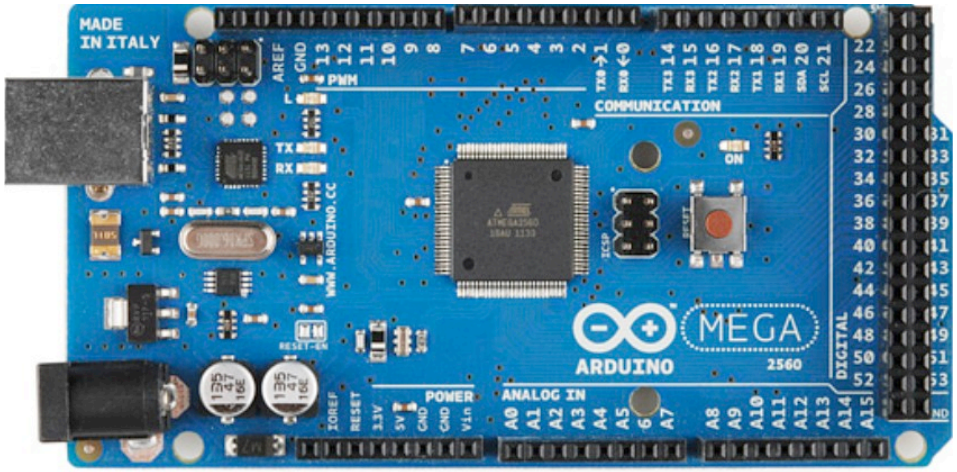
Low-cost LoRa IoT device: A step-by-step tutorial

RESSACS'2016
IRD, Bondy

May 10 th, 2016

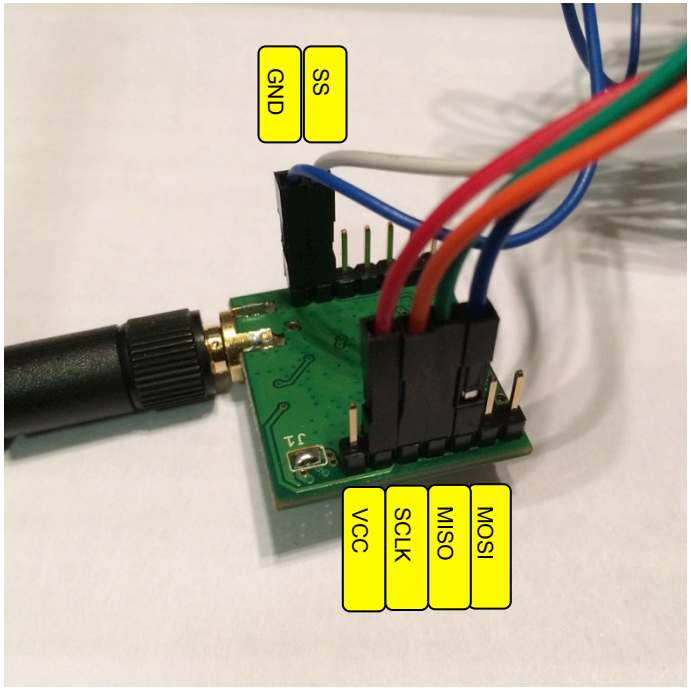
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

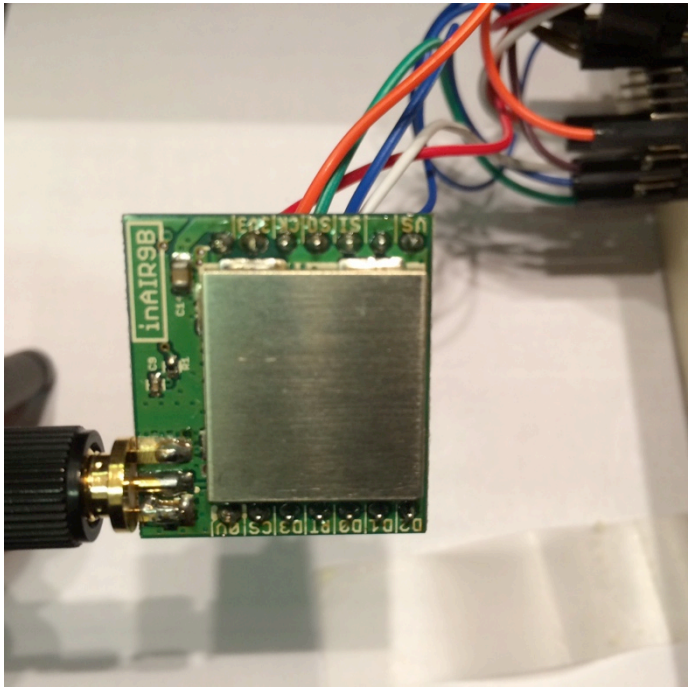
Prepare the board



You either have an Arduino UNO or an Arduino MEGA2650

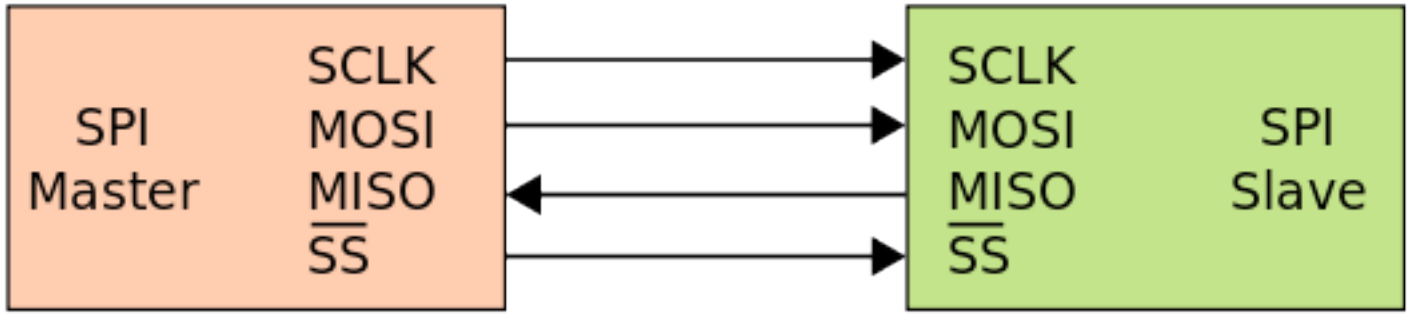
Now the radio module: modtronix inAir9

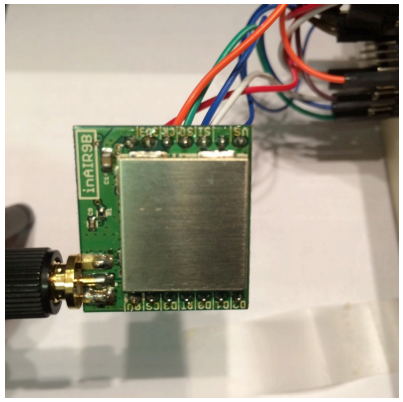




Connect the SPI pins (MOSI, MISO,SCLK) with F/F wires and the SS pin with a F/M wire. Try to use different colors. I use the following colors: MOSI (blue), MISO (green), SS (white), SCLK (orange). Then connect also the radio VCC (red) with a F/M wire and the radio GND (black or any other dark color) also with a F/M wire.

Serial Peripheral Interface (SPI)

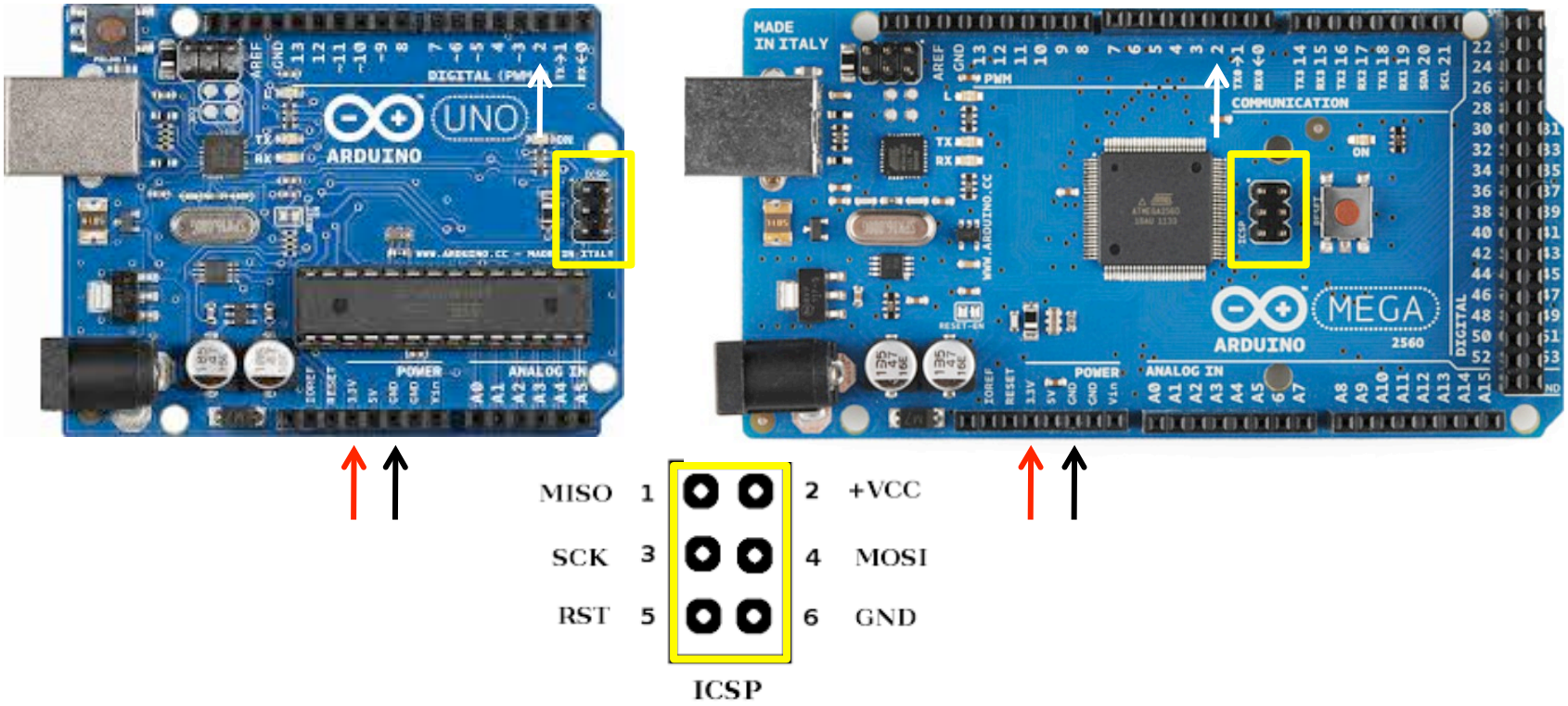




The SPI bus specifies four logic signals:

- SCLK : Serial Clock (output from master).
- MOSI : Master Output, Slave Input (output from master).
- MISO : Master Input, Slave Output (output from slave).
- SS : Slave Select (active low, output from master).

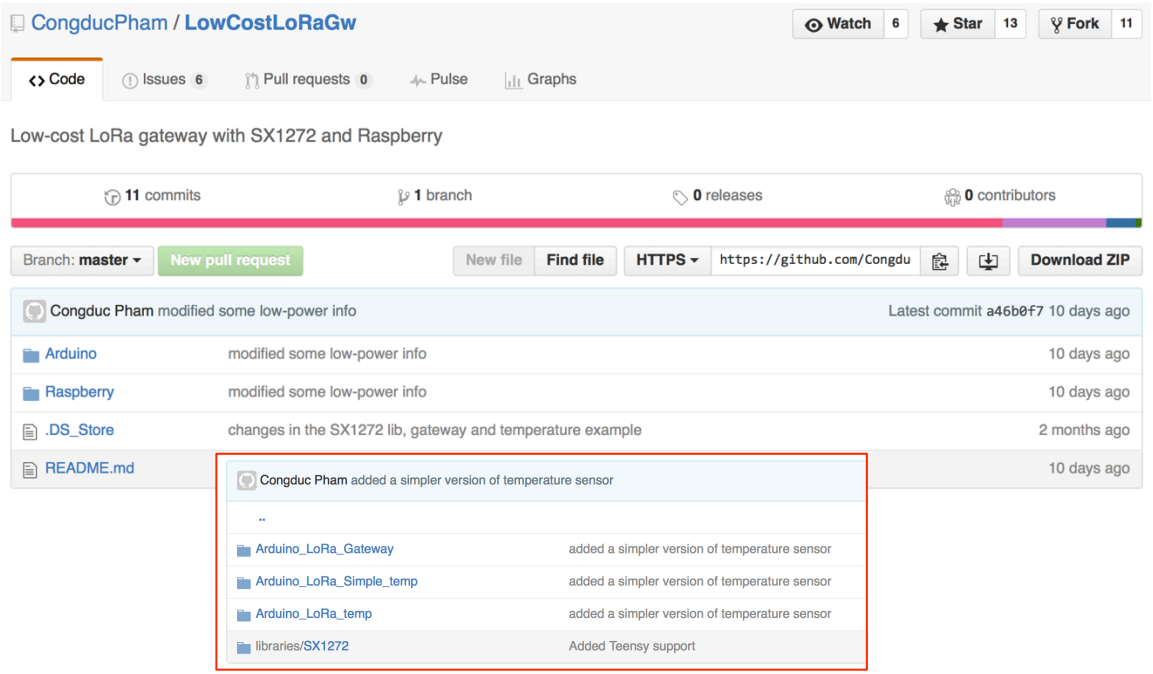
Connecting the radio module



Now, just connect the corresponding SPI pins of the radio module to the SPI pins on the board which are on the ICSP header. SS goes to digital 2. Then connect also the VCC (red) and the GND (black) of the radio board to the 3.3v and the GND of the board. The 3.3v pin of the board gets 3.3v from the on-board voltage regulator.

Getting, compiling & uploading the software

GETTING THE SOFTWARE: A SIMPLE SENDER



Fisrt, you will need the Arduino IDE 1.6.6 or later (left). Then get the LoRa library from our github: https://github.com/CongducPham/LowCostLoRaGw (right).

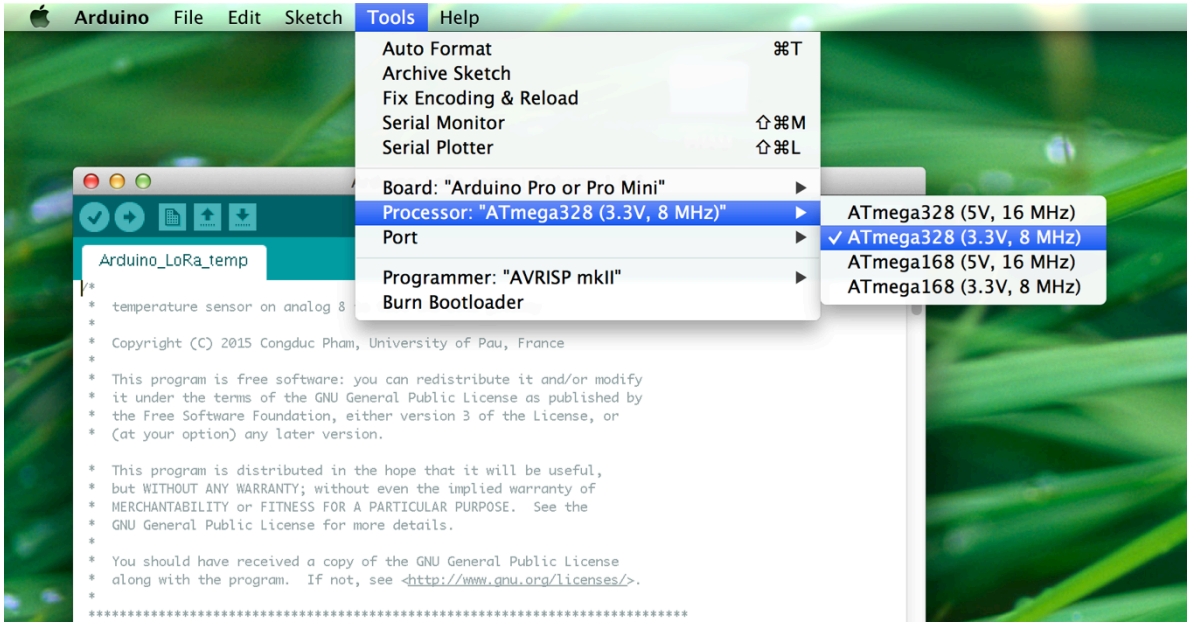
Get into the Arduino folder and get both Arduino_LoRa_Gateway and SX1272 folders. Copy Arduino_LoRa_Gateway into your “sketch” folder and SX1272 into “sketch/libraries”

Make it a sender

```
//#define RECEIVE_ALL
//#define IS_RCV_GATEWAY
#define IS_SEND_GATEWAY
//#define CAD_TEST
//#define LORA_LAS
//#define WINPUT
//#define WITH_SEND_LED
```

Uncomment the statement #define IS_SEND_GATEWAY

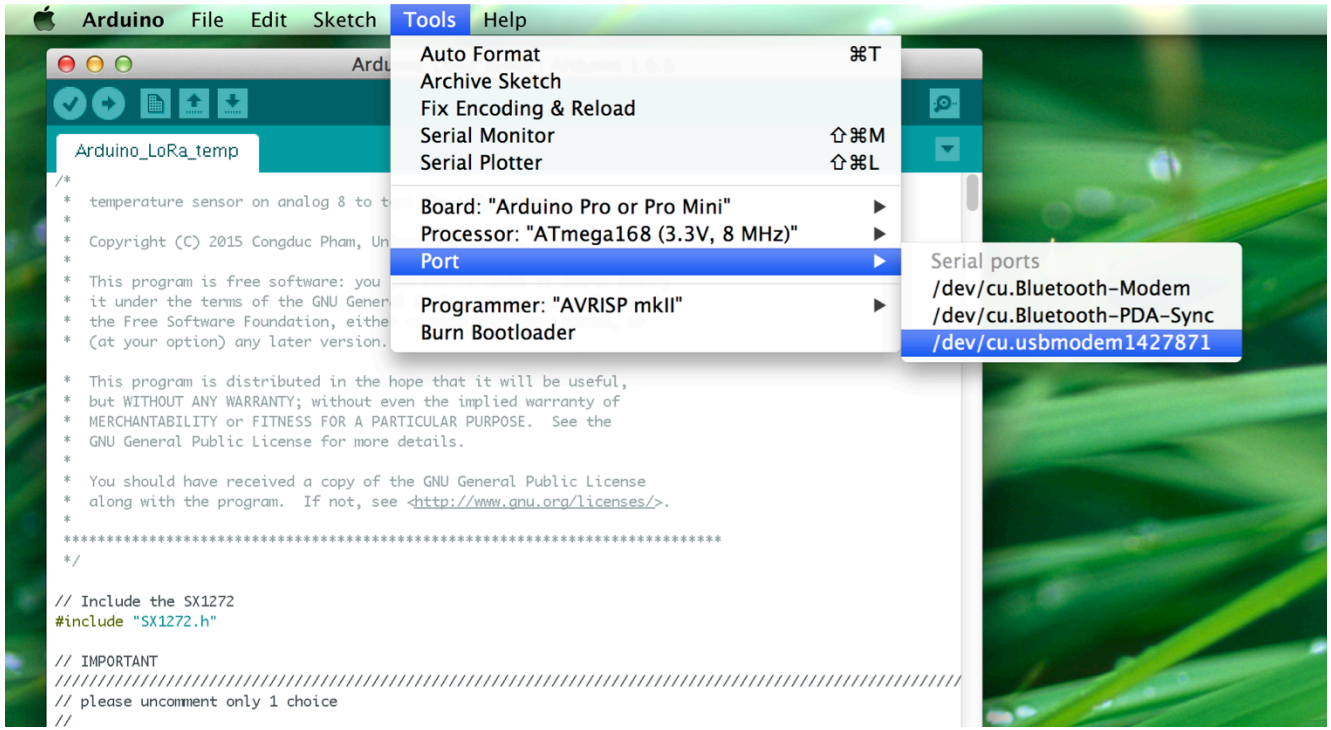
Compiling



Open the Arduino_LoRa_Gateway sketch and select the Arduino UNO or MEGA 2560 depending on the board you have

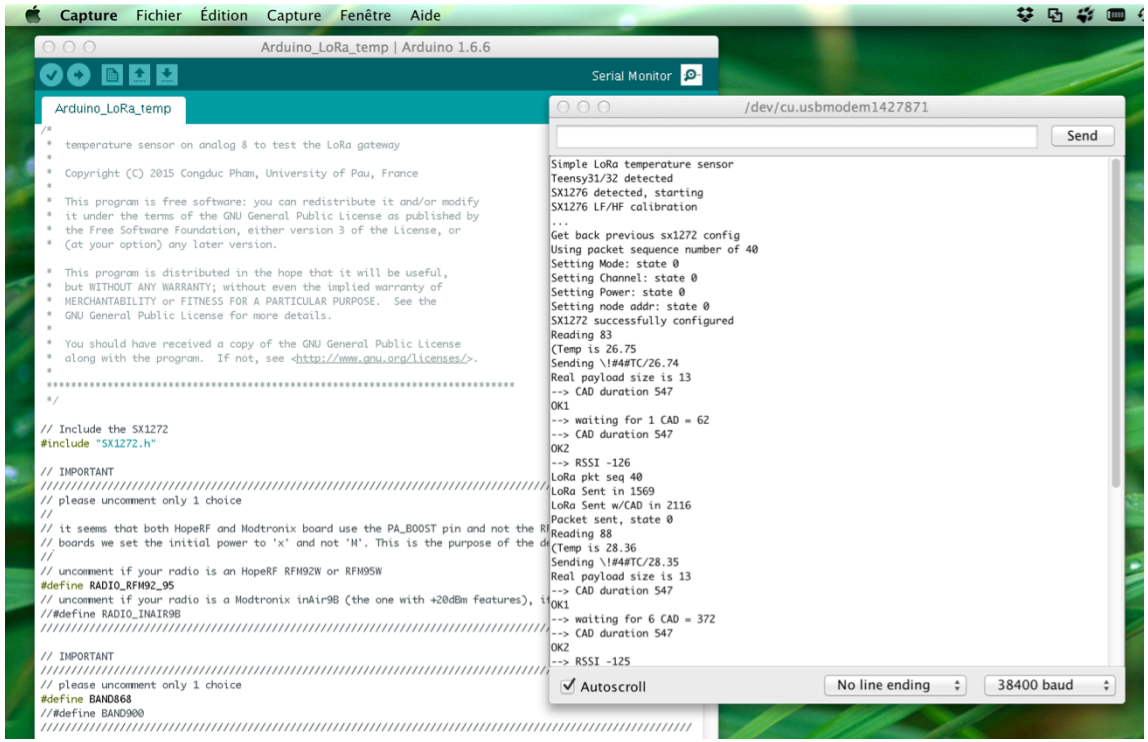
Then, click on the « verify » button

Uploading



Connect the USB end to your computer and the USB port should be detected in the Arduino IDE. Select the serial port for your device. It may have another name than what is shown in the example. Then click on the « upload » button

Serial monitor

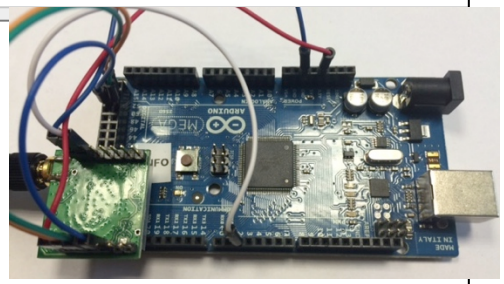


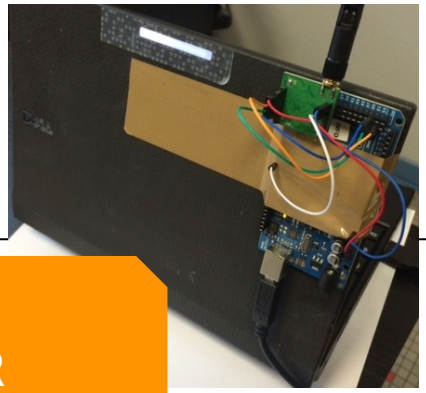
You can see the output from the sensor if it is connected to your computer. Use the Arduino IDE « serial monitor » to get such output, just to verify that the sensor is running fine, or to debug new code. Be sure to use 38400 baud.

Use the interactive sender

- Interactive end-device

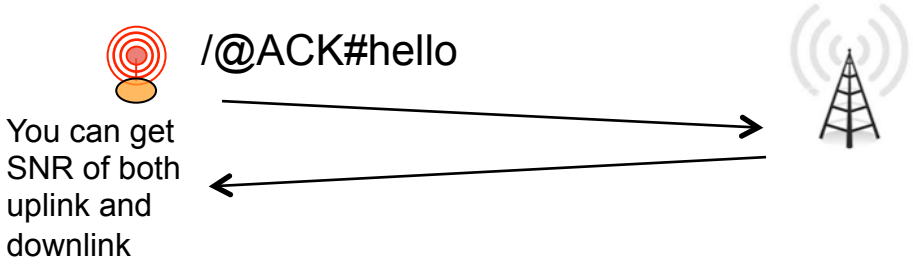
```
/dev/cu.usbmodemFA131 (Arduino/Genuino M
Hello world
6477 bytes of free memory.
SX1276 detected, starting
SX1276 LF/HF calibration
...
^$**********Power ON: state 0
^$Default sync word: 0x12
^$LoRa mode 4
^$Setting mode: state 0
^$Channel CH_10_868: state 0
^$Set LoRa Power to x
^$Power: state 0
^$Get Preamble Length: state 0
^$Preamble Length: 8
^$LoRa addr 6: state 0
^$SX1272/76 configured as device. Waiting serial input for serial-RF bridge
Rcv serial: hello world
Sending. Length is 11
hello world
Payload size is 11
ToA is w/5B Libelium header 322
Packet number 0
LoRa Sent in 545
LoRa Sent w/CAD in 545
Packet sent, state 0
```



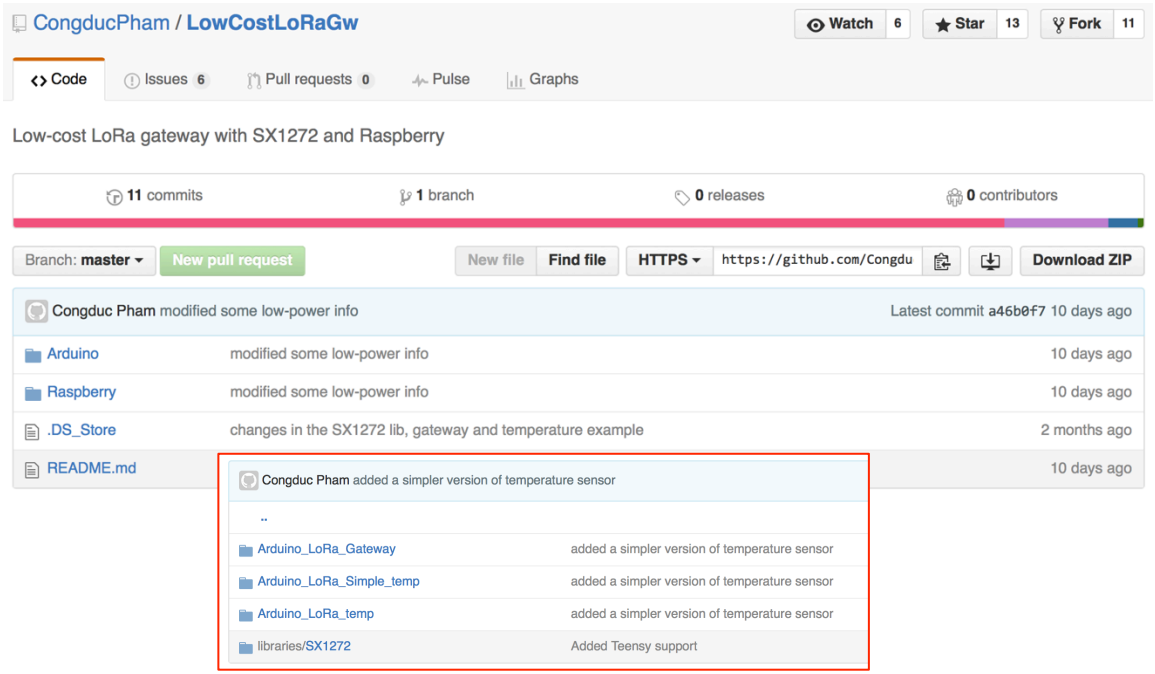


Enter hello world and press ENTER

Command	Action
/@M1#	set LoRa mode 1
/@C12#	use channel 12
/@PL/H/M/x/X#	set power to Low, High, Max, extreme (PA_BOOST), eXtreme (+20dBm)
/@A9#	set node addr to 9
/@ACK#hello w/ac	sends "hello w/ack" and request an ACK
/@ACKON#	enables ACK (for all messages)
/@ACKOFF#	disables ACK
/@CAD#	performs an SIFS CAD, i.e. 3 or 6 CAD depending on the LoRa mode
/@CADON3#	uses 3 CAD when sending data (normally SIFS is 3 or 6 CAD, DIFS=3SIFS)
/@CADOFF#	disables CAD (IFS) when sending data
/@RSSI#	toggles checking of RSSI before transmission and after CAD
/@EIFS#	toggles for extended IFS wait
/@T5000#	send a message at regular time interval of 5000ms. Use /@T0# to disable periodic sending
/@TR5000#	send a message at random time interval between [2000, 5000]ms.
/@Z200#	sets the packet payload size to 200 for periodic sending
/@S50#	sends a 50B user payload packet filled with '#'. The real size is 55B with the protocol header
/@D56#	set the destination node to be 56, this is permanent, until the next D command
/@D56#hello	send "hello" to node 56, destination addr is only for this message
/@D1#/@M1#	send the command string "/@M1#" to node 1 (i.e. gateway)



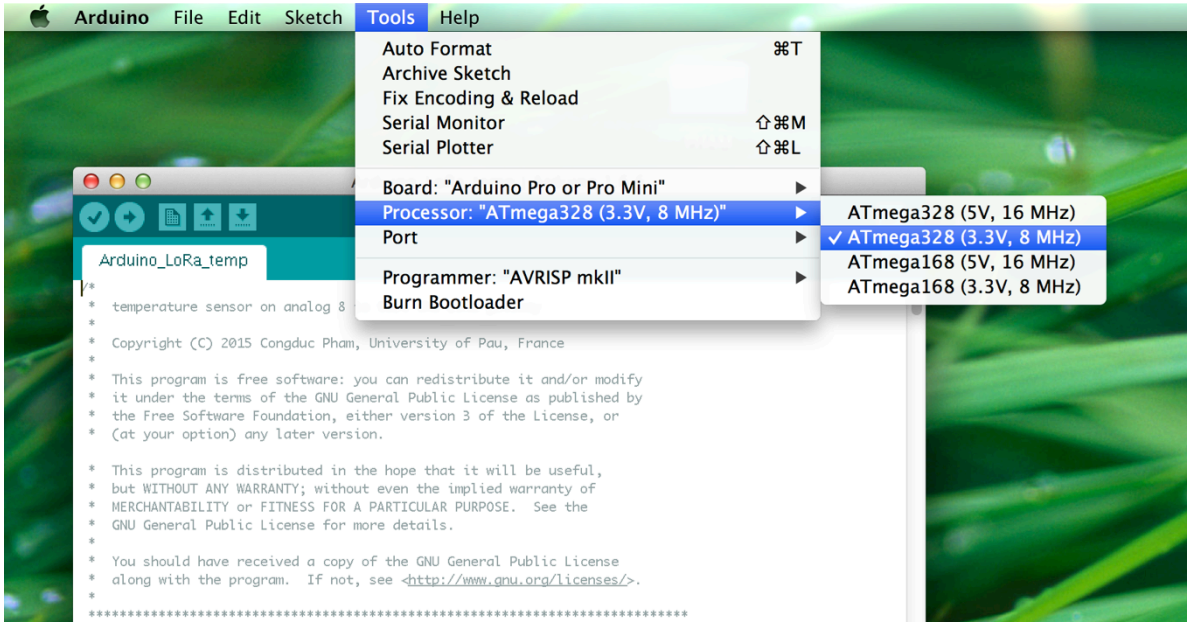
GETTING THE SOFTWARE: A TEMPERATURE SENSOR



Fisrt, you will need the Arduino IDE 1.6.6 or later (left). Then get the LoRa library from our github: https://github.com/CongducPham/LowCostLoRaGw (right).

Get into the Arduino folder and get Arduino_LoRa_Simple_temp. Copy Arduino_LoRa_Simple_temp into your “sketch” folder. Compile then Upload as previously.

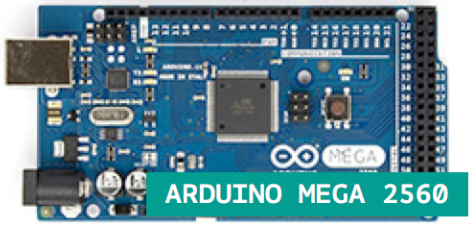
Compiling

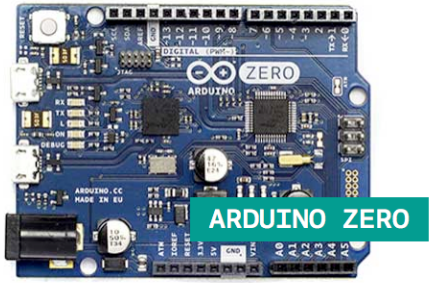


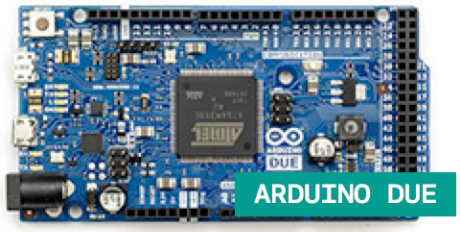
Open the Arduino_LoRa_Simple_temp sketch and select the Arduino UNO or MEGA 2560 depending on the board you have

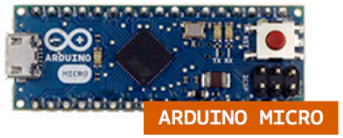
Then, click on the « verify » button

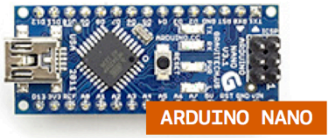
MORE PLATFORMS ARE SUPPORTED!

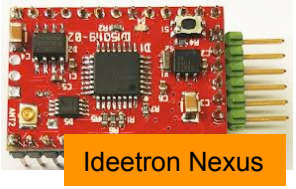




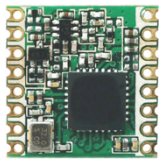








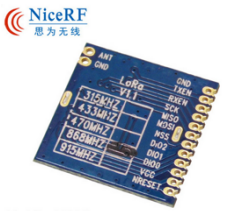
LoRa radios that our library already supports



HopeRF RFM92W/95W

Libelium LoRa

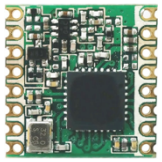
Modtronix inAir9/9B



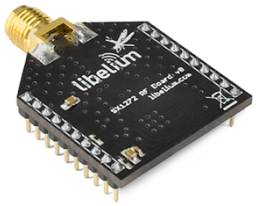
NiceRF LoRa1276

Long-Range communication library
(mostly sending functions)

LoRa radios that our library already supports



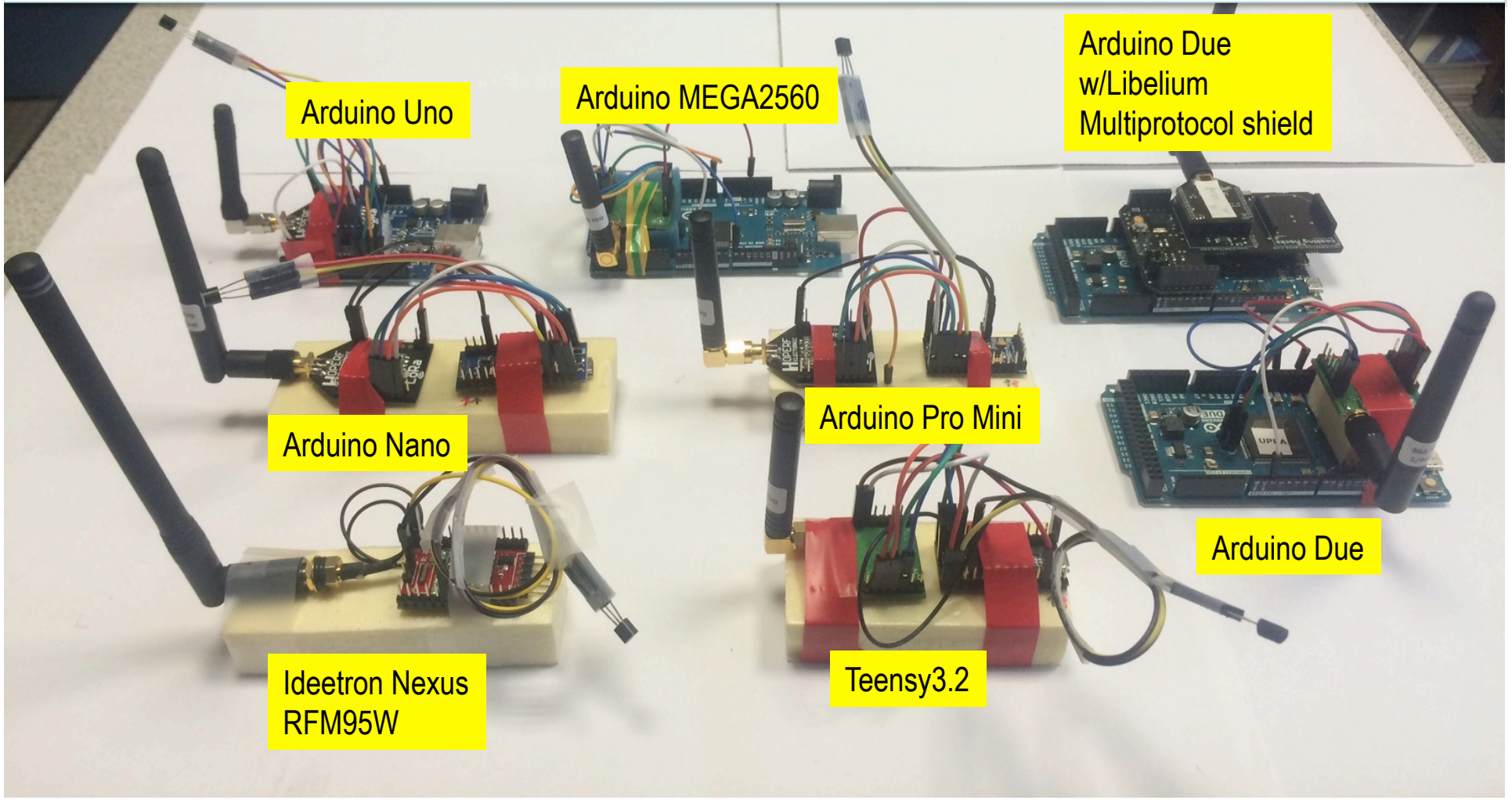
HopeRF RFM92W/95W



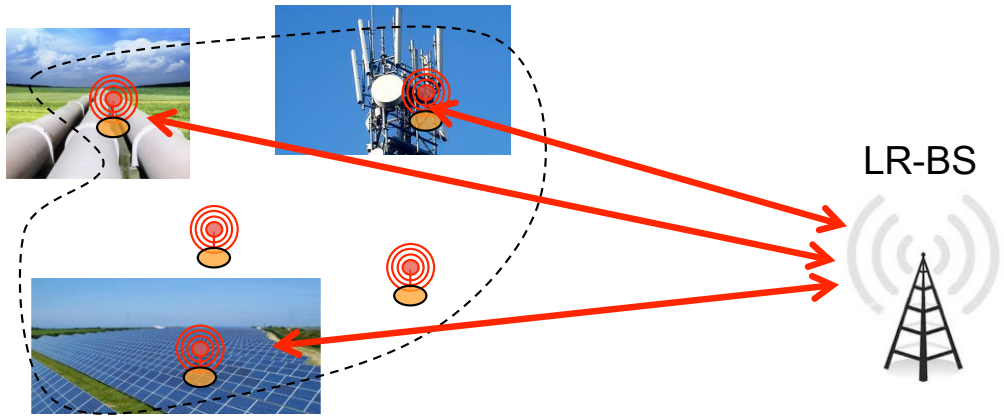
Libelium LoRa

Modtronix inAir9/9B

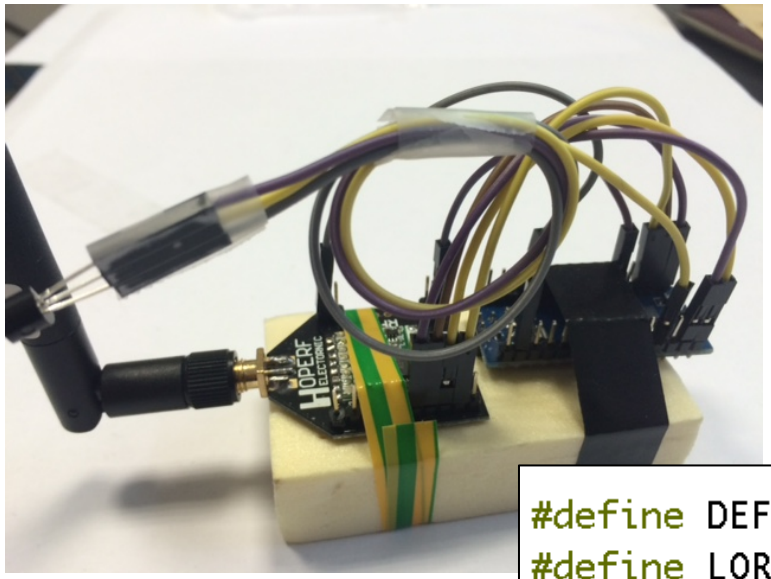
NiceRF LoRa1276



Sending to a gateway



DEFAULT CONFIGURATION



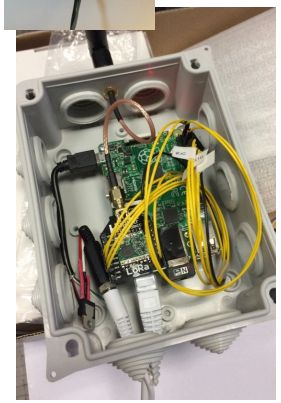
\!#4#18.5

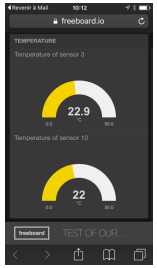
```
#define DEFAULT_DEST_ADDR 1
#define LORAMODE  1
#define node_addr 8
```

The default configuration in the Arduino_LoRa_Simple_temp example is:

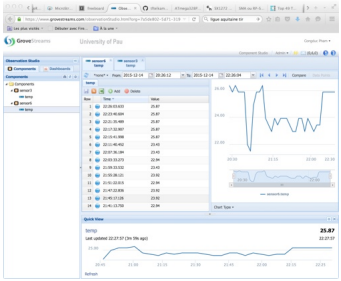
Send packets to the gateway (one or many if in range)
LoRa mode 1
Node short address is 8

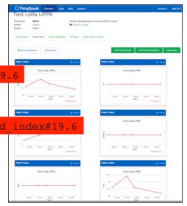
Gateway to cloud





freeboard





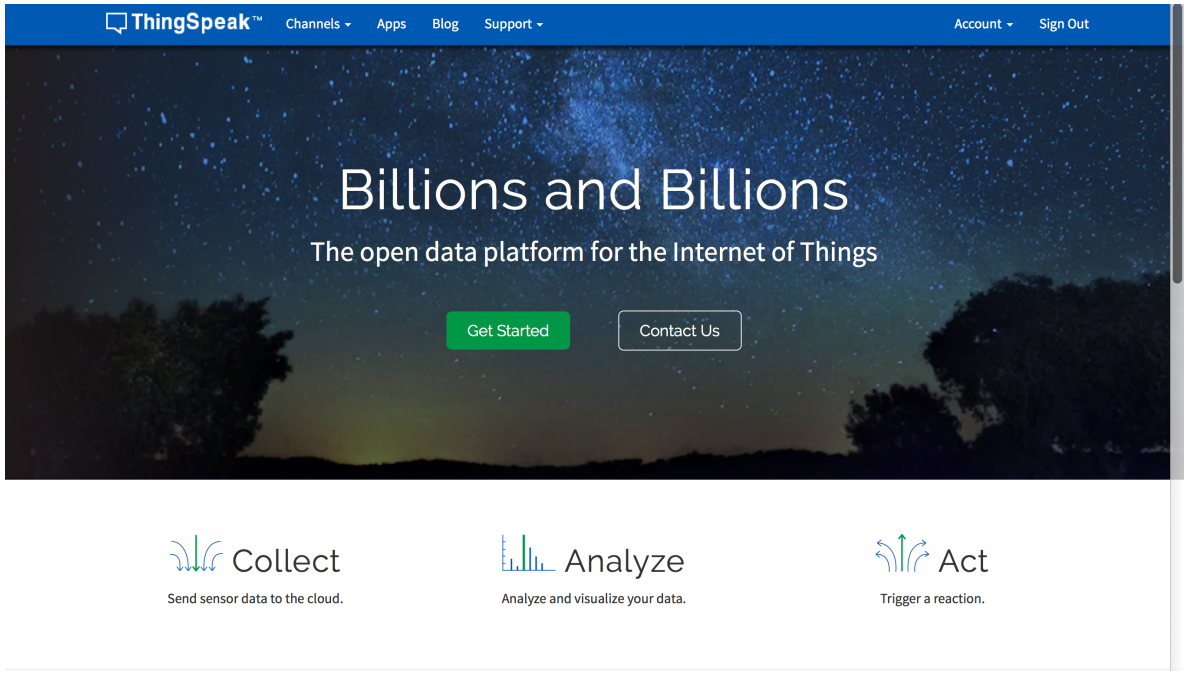
GroveStreams

SensorCloud™

Data received at the gateway will be pushed to IoT clouds. We provide python script examples for many IoT cloud platforms. Most of clouds with REST API can be easily integrated.

In this example, data will be pushed to the ThingSpeak cloud on field index 4.

Use your own cloud

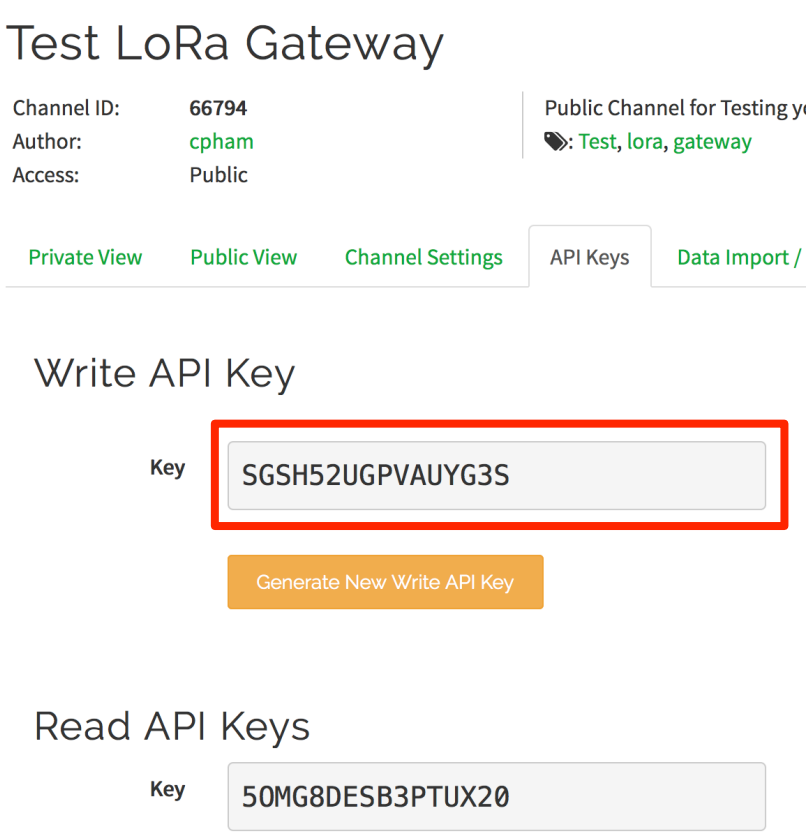


Go to https://thingspeak.com/

Create your own ThingSpeak account (Sign Up)

And create a new channel, called it ressac2016_lora_test for instance

Get the API key of your channel



The write key is the key that will allow you to write data to your ThingSpeak channel

CHANGE THE SENSOR TO SEND TO YOUR CHANNEL

```
#ifdef FLOAT_TEMP
      ftoa(float_str,temp,2);

#ifdef NEW_DATA_FIELD
      r_size=sprintf((char*)message+app_key_offset, "\\!#%d#TC/%s", field_index, float_str);
#else
      // this is for testing, uncomment if you just want to test, without a real temp sensor plugged
      //strcpy(float_str, "21.55567");
      r_size=sprintf((char*)message+app_key_offset, "\\!#%d#%s", field_index, float_str);
#endif
```

Get your ThnigSpeak channel's API write key: i.e. SGSH52UGPVAUYG3S

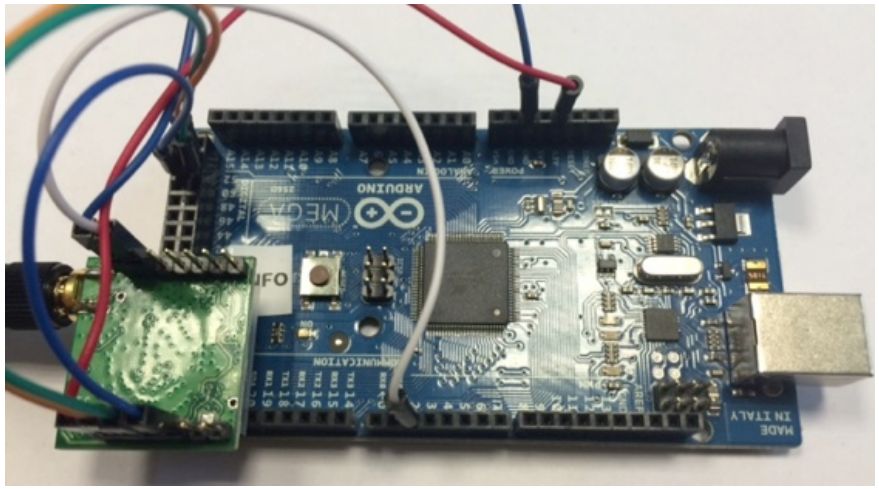
Change "\\!#%d#TC/%s » into "\\!SGSH52UGPVAUYG3S#%d#TC/%s"

Compile and upload.

Using the interactive sender

```
/dev/cu.usbmodemFA131 (Arduino/Genuino M

6477 bytes of free memory.
SX1276 detected, starting
SX1276 LF/HF calibration
...
^$**********Power ON: state 0
^$Default sync word: 0x12
^$LoRa mode 4
^$Setting mode: state 0
^$Channel CH_10_868: state 0
^$Set LoRa Power to x
^$Power: state 0
^$Get Preamble Length: state 0
^$Preamble Length: 8
^$LoRa addr 6: state 0
^$SX1272/76 configured as device. Waiting serial input for serial-RF bridge
Rcv serial: hello world
Sending. Length is 11
hello world
Payload size is 11
ToA is w/5B Libelium header 322
Packet number 0
LoRa Sent in 545
LoRa Sent w/CAD in 545
Packet sent, state 0
```



Assume your API write key is SGSH52UGPVAUYG3S

Enter \!SGSH52UGPVAUYG3S#1#TC/20.5

Try field value from 1 to 8. Try other value: i.e. 31.5