Low-cost LoRa Collar for cattle rustling applications

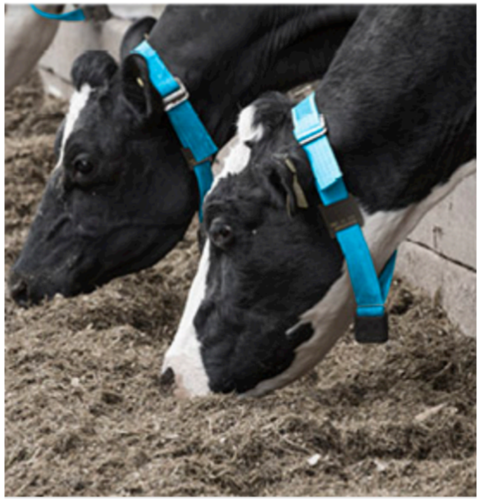
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

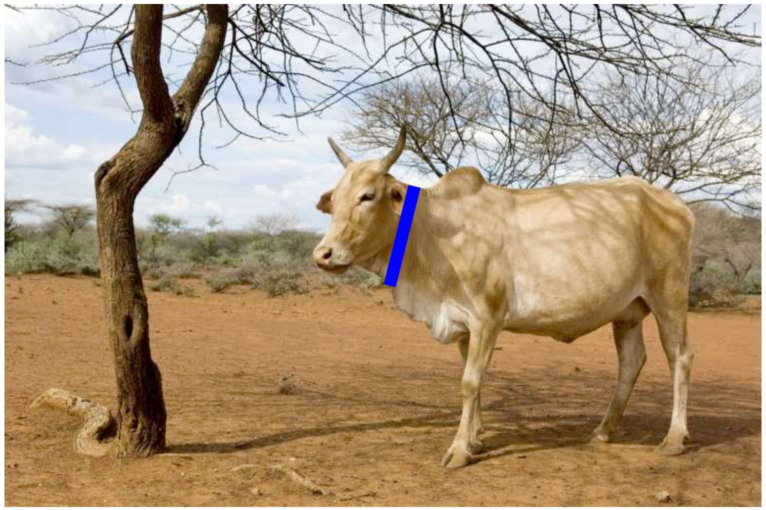
Overview

- We will show how to build a simple low-cost collar for preventing cattle rustling in developing countries. The system is proposed in 2 versions
- **Version 1: simple beacon**
  - When powered on, sends every 10 minutes a beacon
  - The distance can be estimated with the beacon's RSSI
  - The gateway will receive the beacon messages and tries to detect whether an alarm should be raised or not (no beacons for instance or very low RSSI)
- **Version 2: GPS beacon with GPS module**
  - When powered on, sends every 10 minutes a GPS beacon
  - The position can be determined with the GPS coordinates
  - The gateway will receive the GPS beacon messages and tries to detect whether an alarm should be raised or not (no beacons for instance or out-of-range position)

How it works?

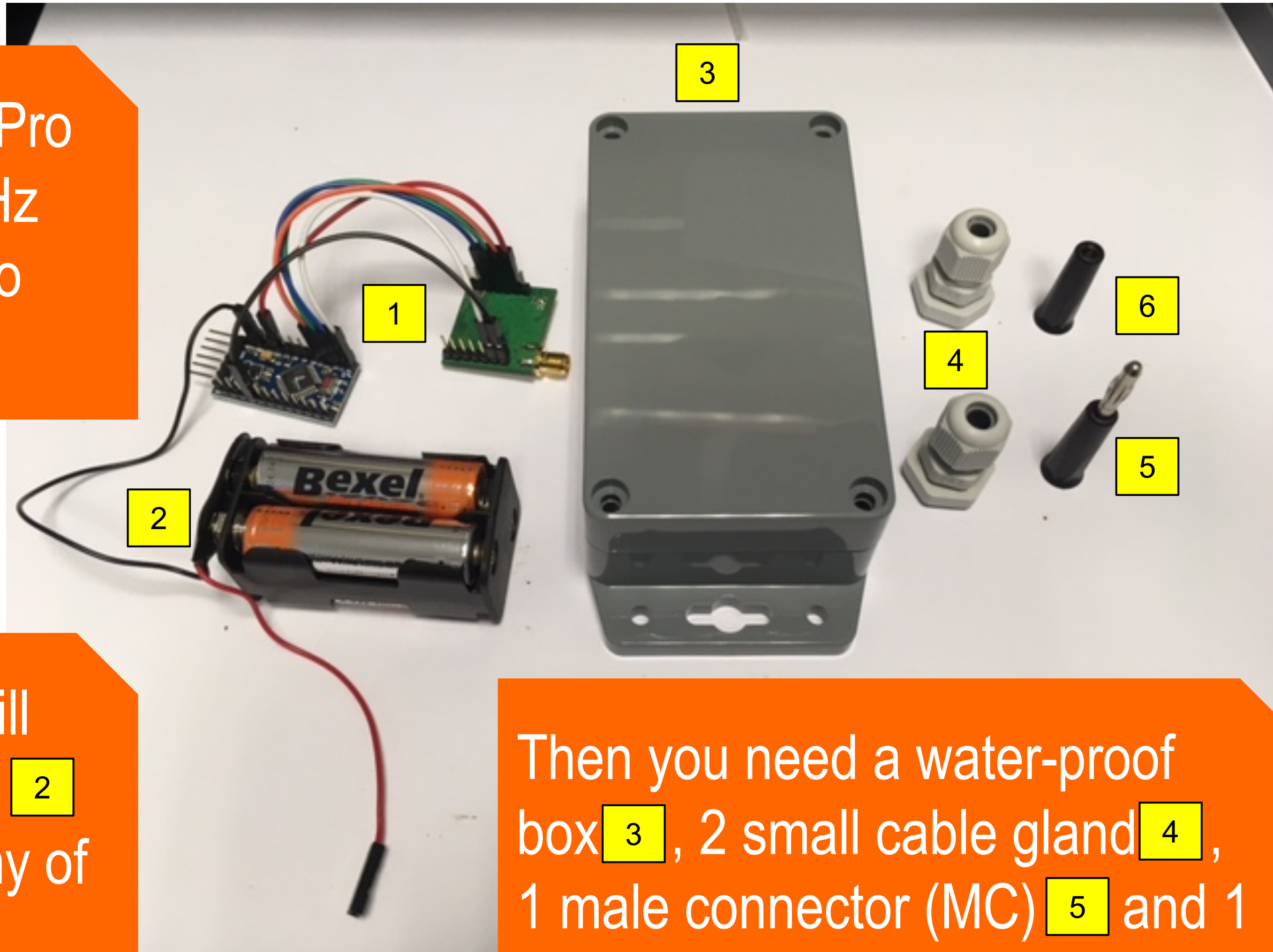
- The collar will be fixed to the cow, around neck. Example picture from Afimilk Silent Herdsman for health monitoring
- In our case, reception of beacon means that the cattle is in range (with GPS, the exact position can be determined)
- If out-of-range, disconnected or damaged device, an alarm can be raised
- To detect collar cutting, the power wire will also goes around the catlle's neck





Building the active beaconing system

Use an Arduino Pro Mini 3.3v at 8MHz and a LoRa radio module 1



4 AA batteries will power the board 2 with an autonomy of several months

Then you need a water-proof box 3, 2 small cable gland 4, 1 male connector (MC) 5 and 1 female connector (FC) 6

Drill holes

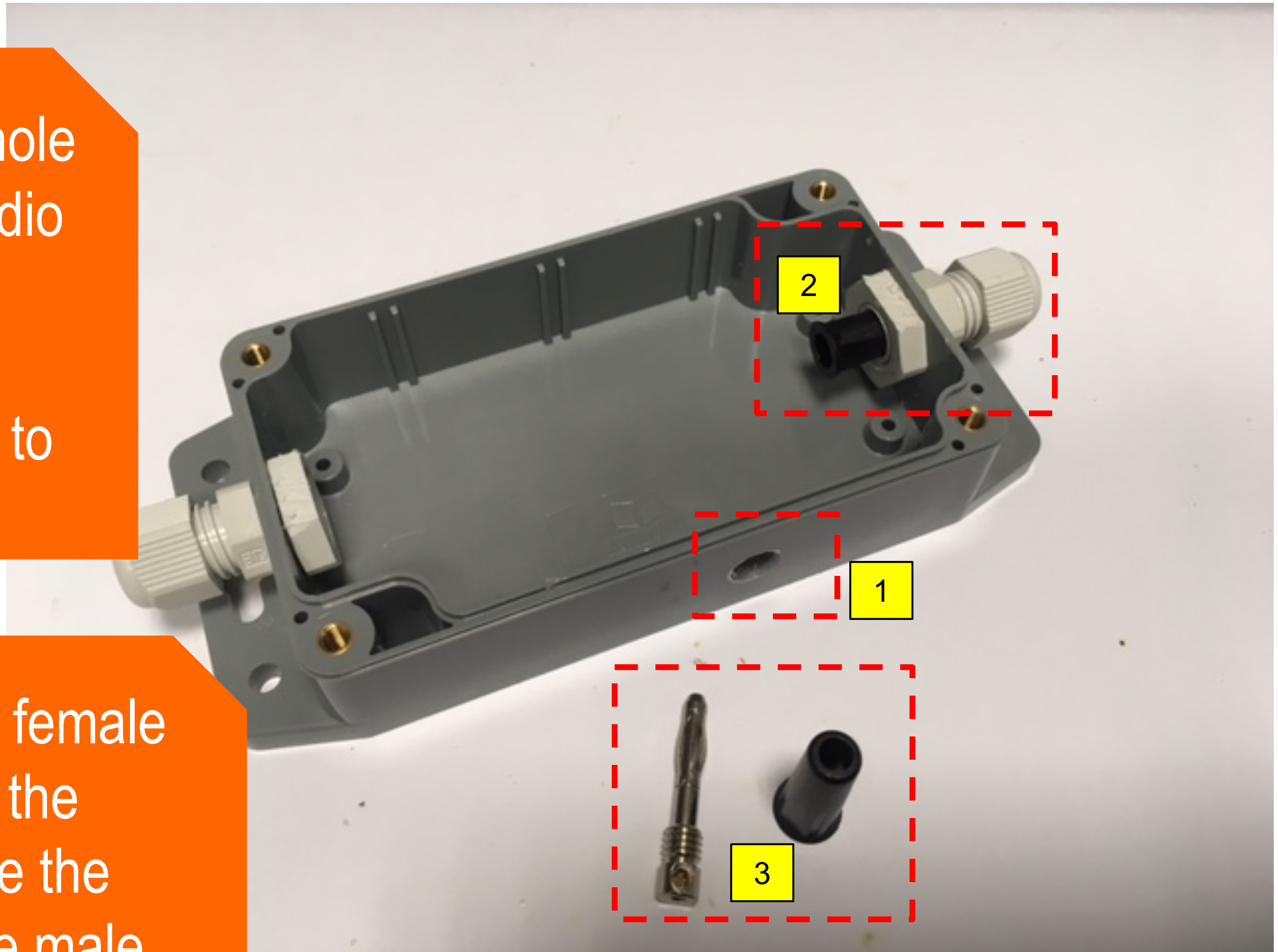
Drill 2 holes depending on the gland diameter (here 12mm, so hole of 13mm)

Place the glands

There is also 1 hole (8mm) for the radio module. 1

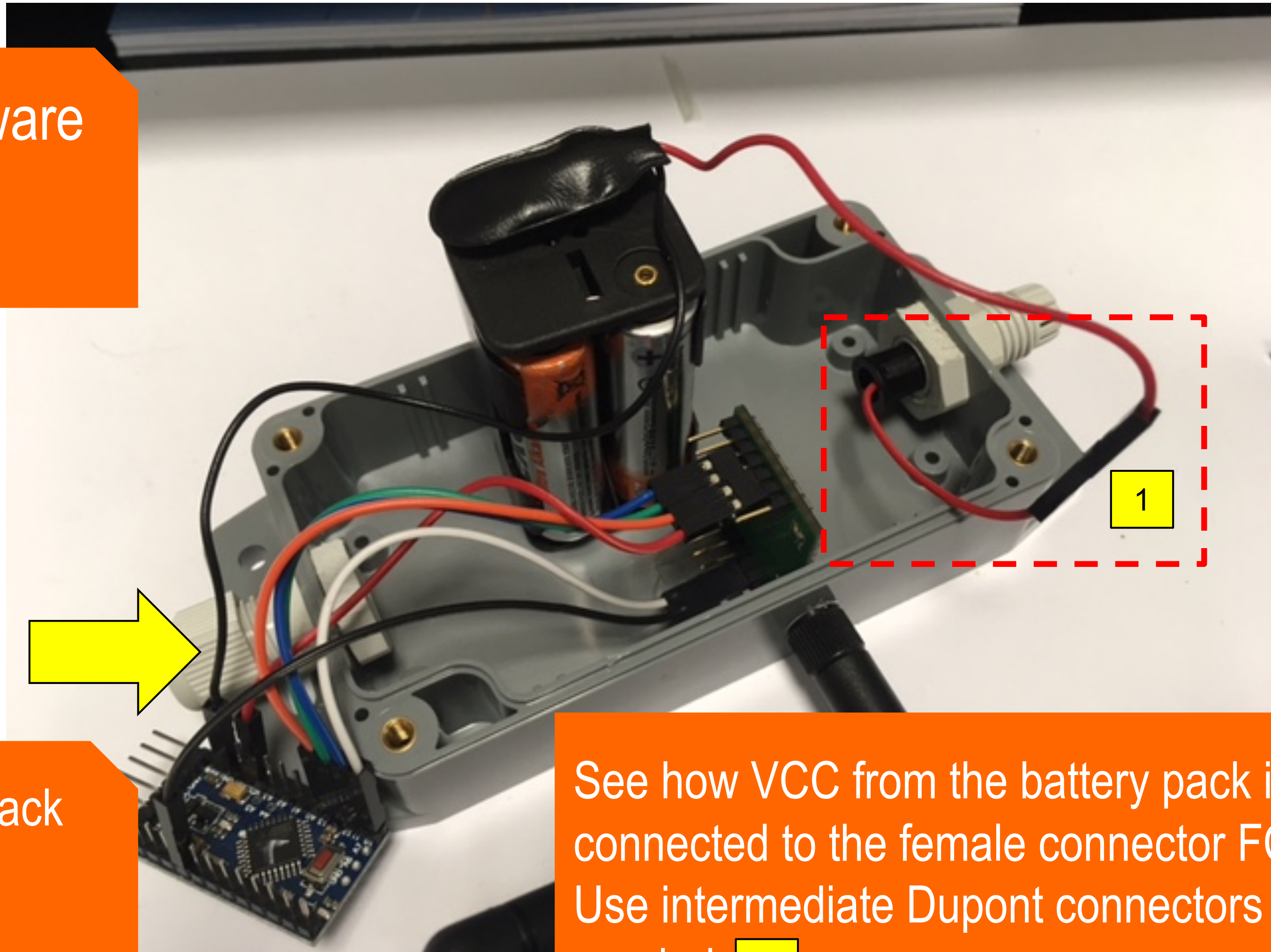
Screw the gland to the box

Fix (in force) the female connector FC to the gland 2, remove the plastic part of the male connector MC 3



Put hardware in place

Put all the hardware parts in the box



GND from battery pack is connected to the board (see arrow)

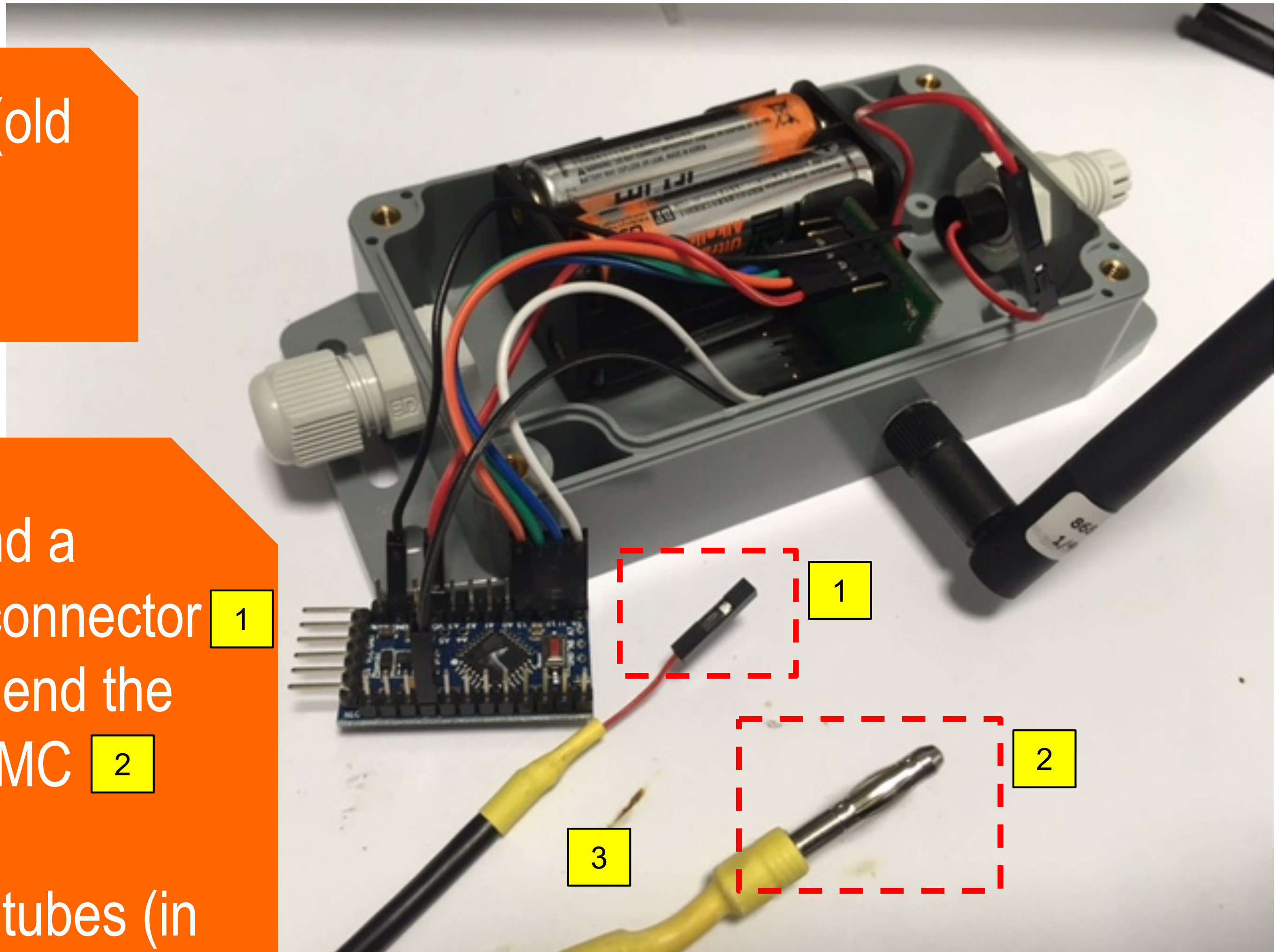
See how VCC from the battery pack is connected to the female connector FC. Use intermediate Dupont connectors if needed. 1

Get a long wire

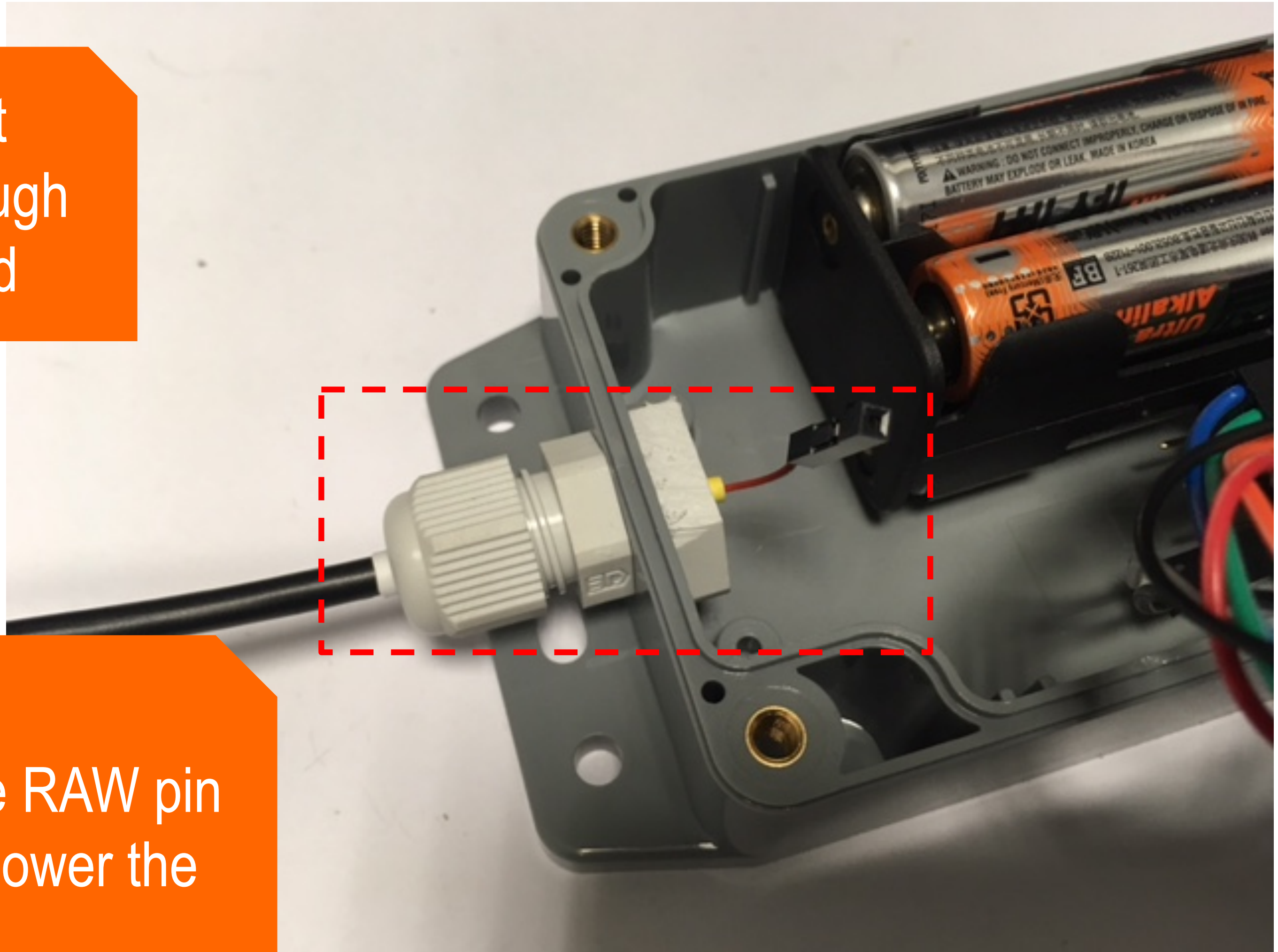
Take a 1m wire (old USB wire for instance)

Solder at one end a female Dupont connector 1 and at the other end the male connector MC 2

Use heat-shrink tubes (in yellow) 3



Connecting the power wire (1)

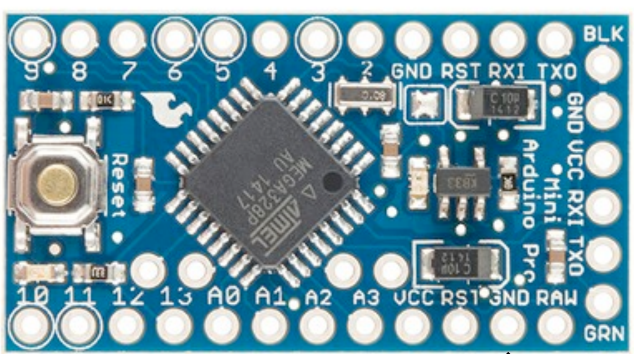


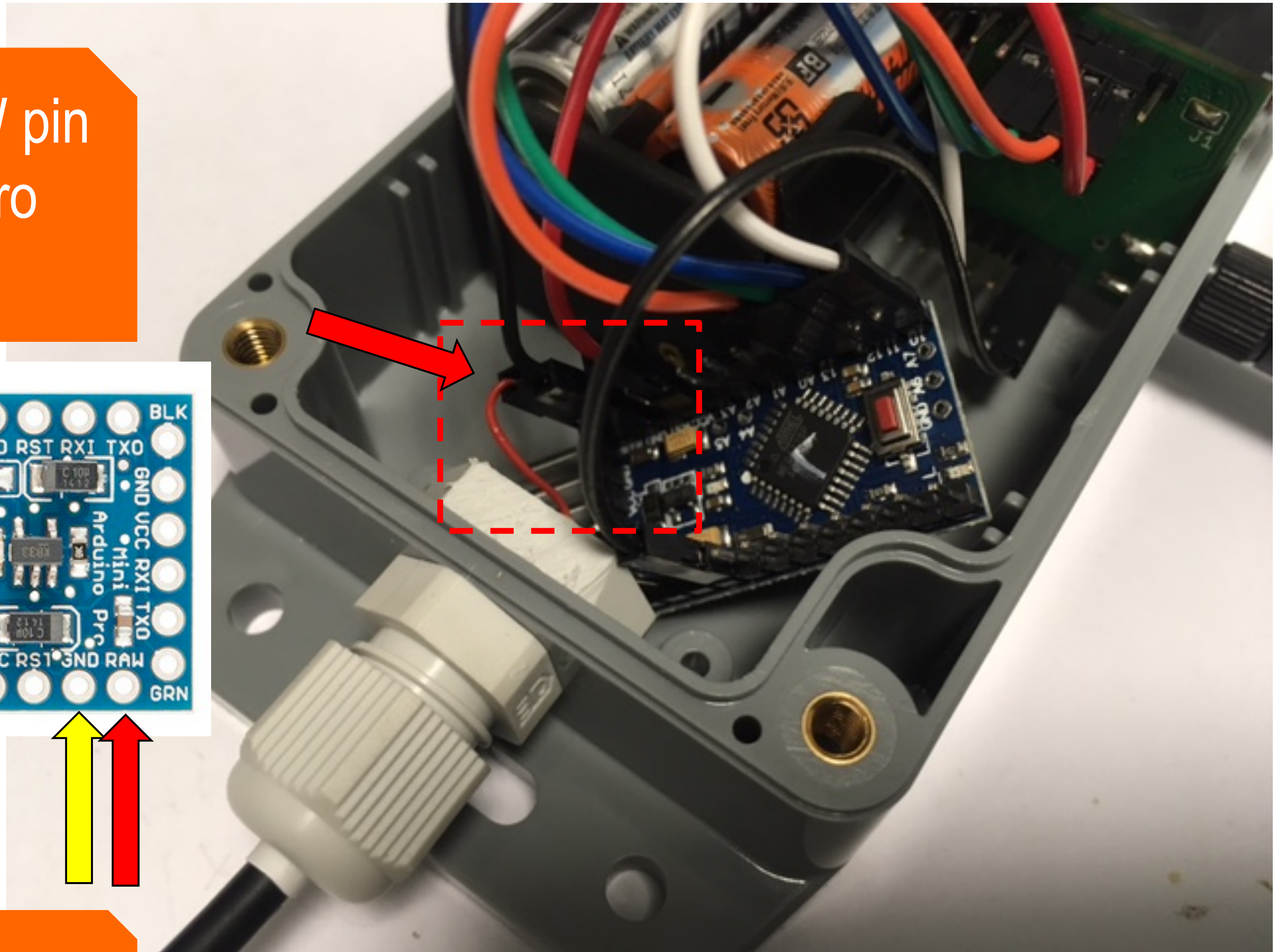
Pass the Dupont female end through the second gland

This end will be connected to the RAW pin of the board to power the board

Connecting the power wire (2)

Connect to RAW pin of the Arduino Pro Mini

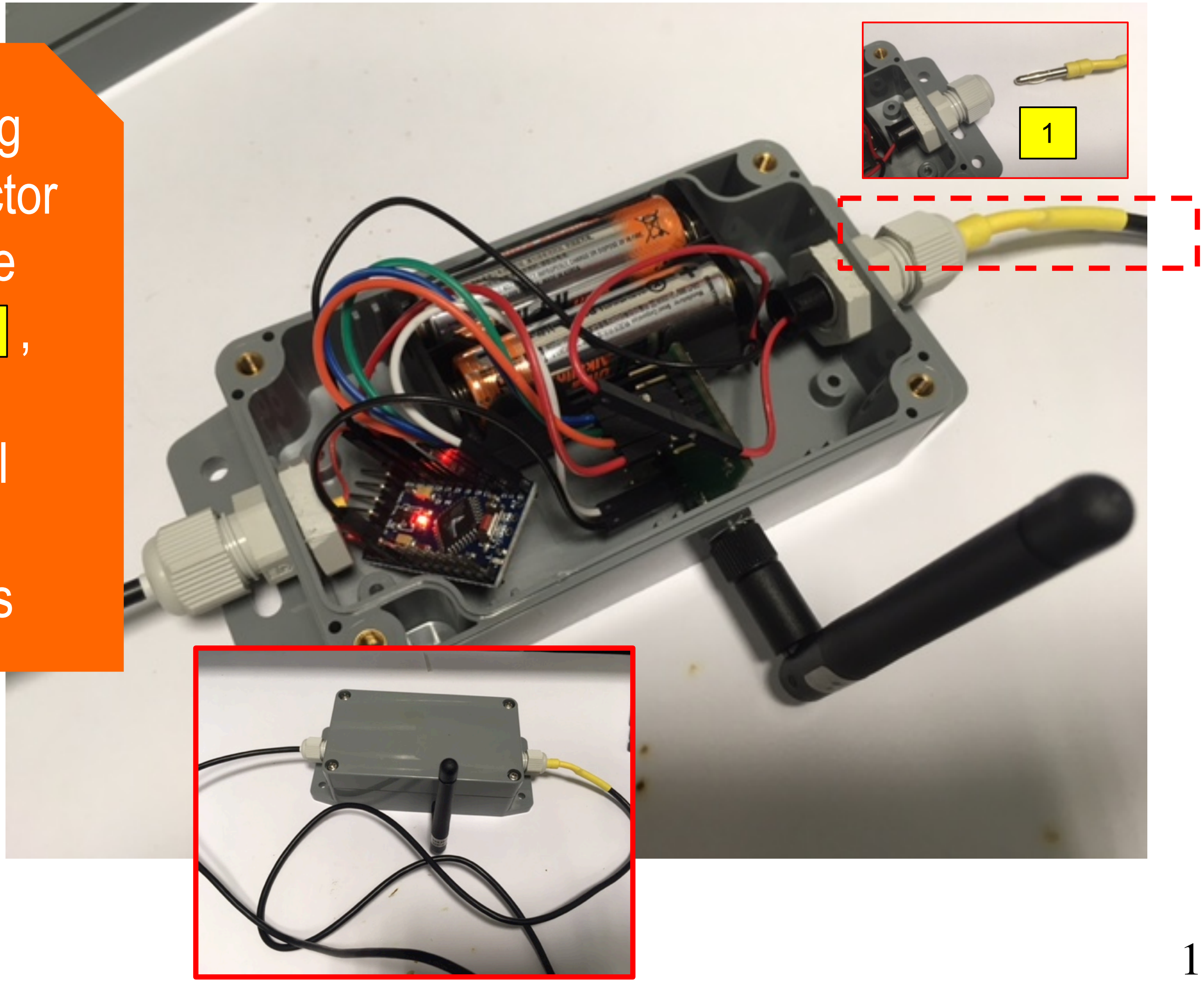




GND was already connected at previous step

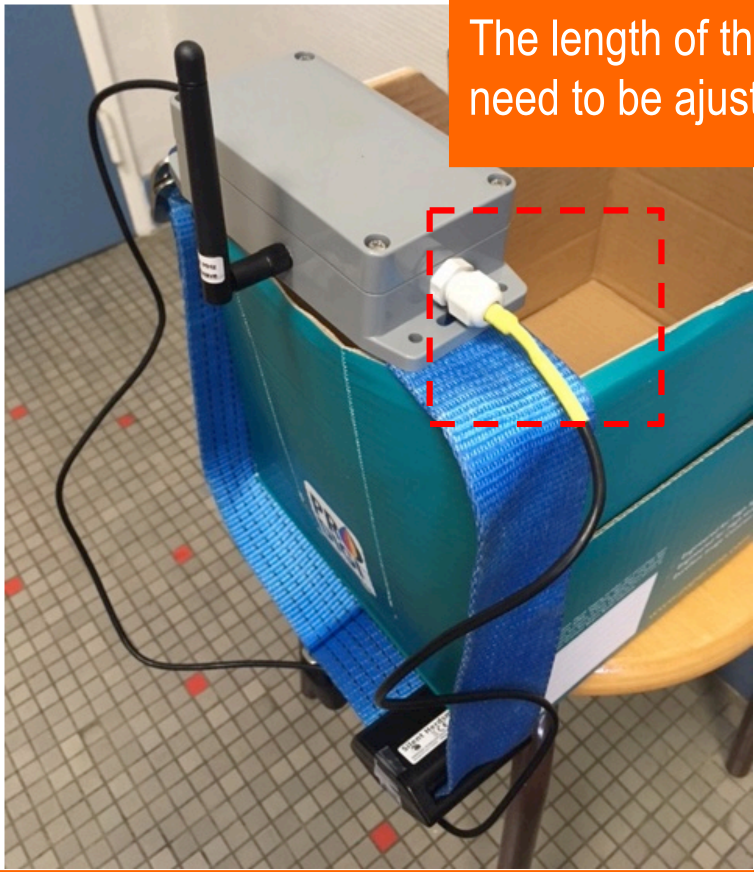
Activating the beacon system

When connecting the male connector MC to the female connector FC 1 , the board will be powered and will start sending periodic beacons

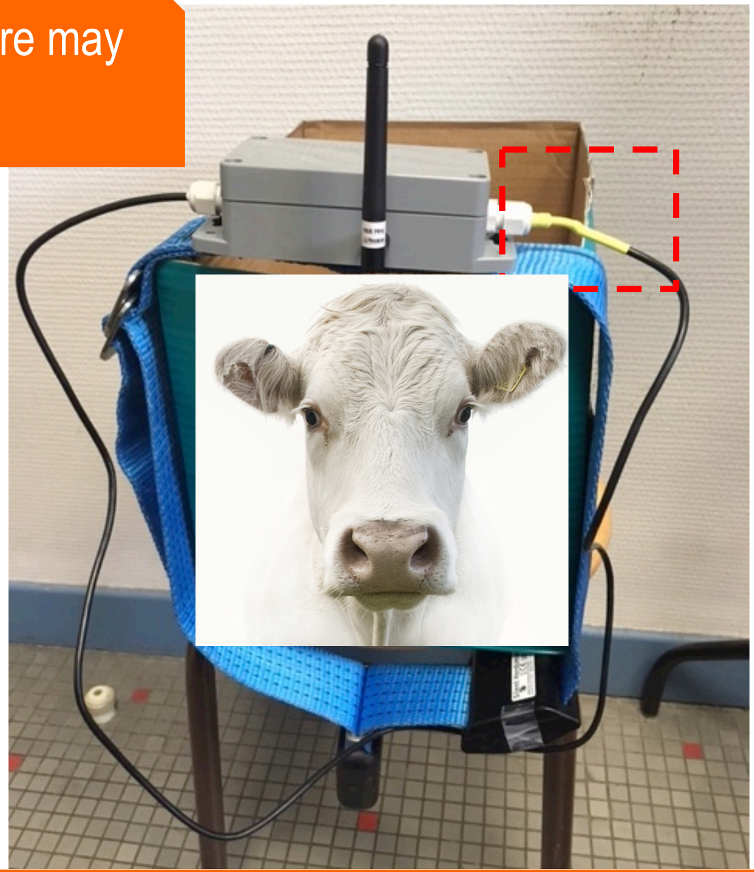


From beacon system to beacon collar

Afimilk collar courtesy of I. Andonovic from University of Strathclyde

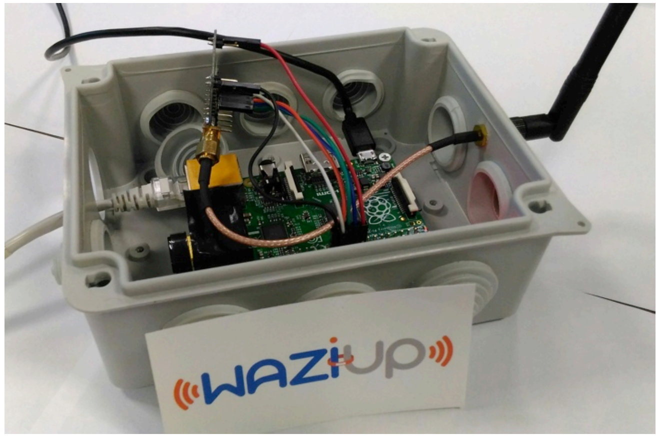


The length of the wire may need to be ajusted

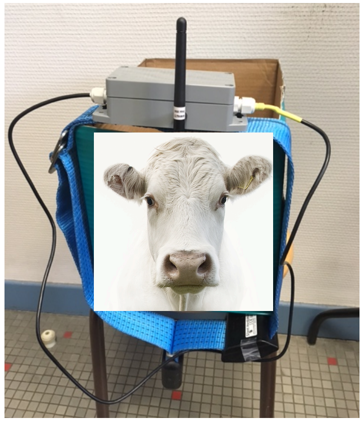


Use a robust belt for the collar, fix the box on the belt. Activate the beacon collar by connecting MC to FC by passing the long wire around the cattle's neck. Any attempt to remove or cut the collar will disconnect the beacon system.

Default configuration



\!BC/0



The default configuration in the Arduino_LoRa_Simple_Beacon_collar example is:

Use LoRa mode 1
Send beacons to the gateway every 20 minutes
Node short address is 14
BC (beacon counter) starts at 0, increases by 1 at each beacon, returns to 0 after 65536 beacons

Beacon messages can be encrypted to avoid malicious spoofing: uncomment WITH_AES

Receiving beacons

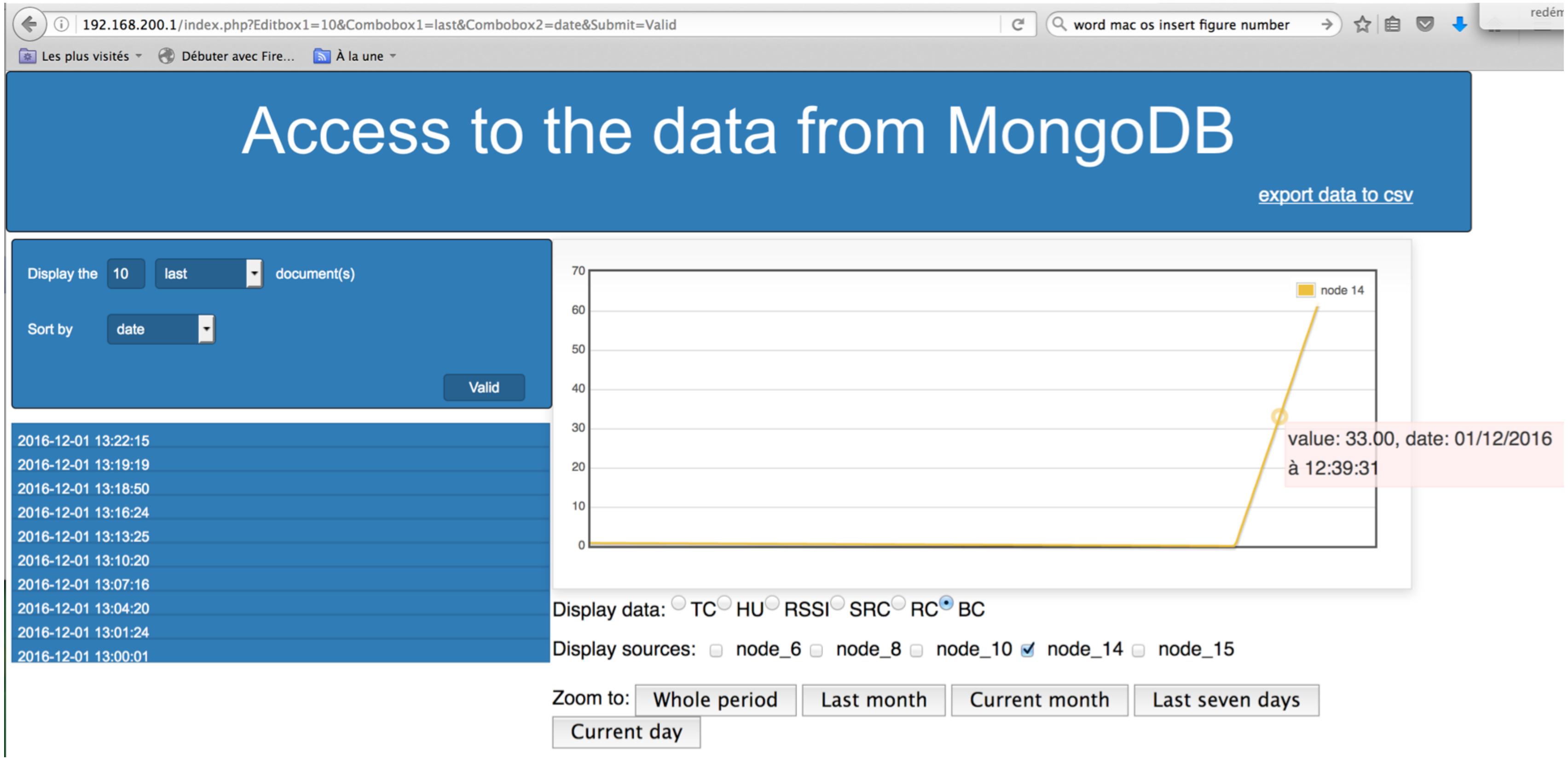
```
--- rxlora. dst=1 type=0x12 src=14 seq=14 len=18 SNR=7 RSSIpkt=-53 BW=125 CR=4/5 SF=12
2016-11-18T12:39:42.566460
rcv ctrl pkt info (^p): 1,18,14,14,18,7,-53
splitted in:  [1, 18, 14, 14, 18, 7, -53]
(dst=1 type=0x12(DATA WAPPKEY) src=14 seq=14 len=18 SNR=7 RSSI=-53)
rcv ctrl radio info (^r): 125,5,12
splitted in:  [125, 5, 12]
(BW=125 CR=5 SF=12)
rcv timestamp (^t): 2016-11-18T12:39:42.565

got first framing byte
--> got data prefix
--> DATA with_appkey: read app key sequence
app key is  0x05 0x06 0x07 0x08
in app key list
valid app key: accept data
BC/14
number of enabled clouds is 1
--> cloud[0]
uploading with python CloudMongoDB.py
MongoDB with max months to store is 2
MongoDB: removing obsolete entries
MongoDB: deleting data older than 2 month(s)...
MongoDB: 0 documents deleted
MongoDB: saving the document in the collection...
MongoDB: saving done
--> cloud end
```

Here, we store in the local MongoDB database

CONNECT TO THE GATEWAY

- Use the WiFi interface
  - SSID is for instance WAZIUP_PI_GW_24EBD4F300
  - Password is loragateway
  - Gateway address is 192.168.200.1

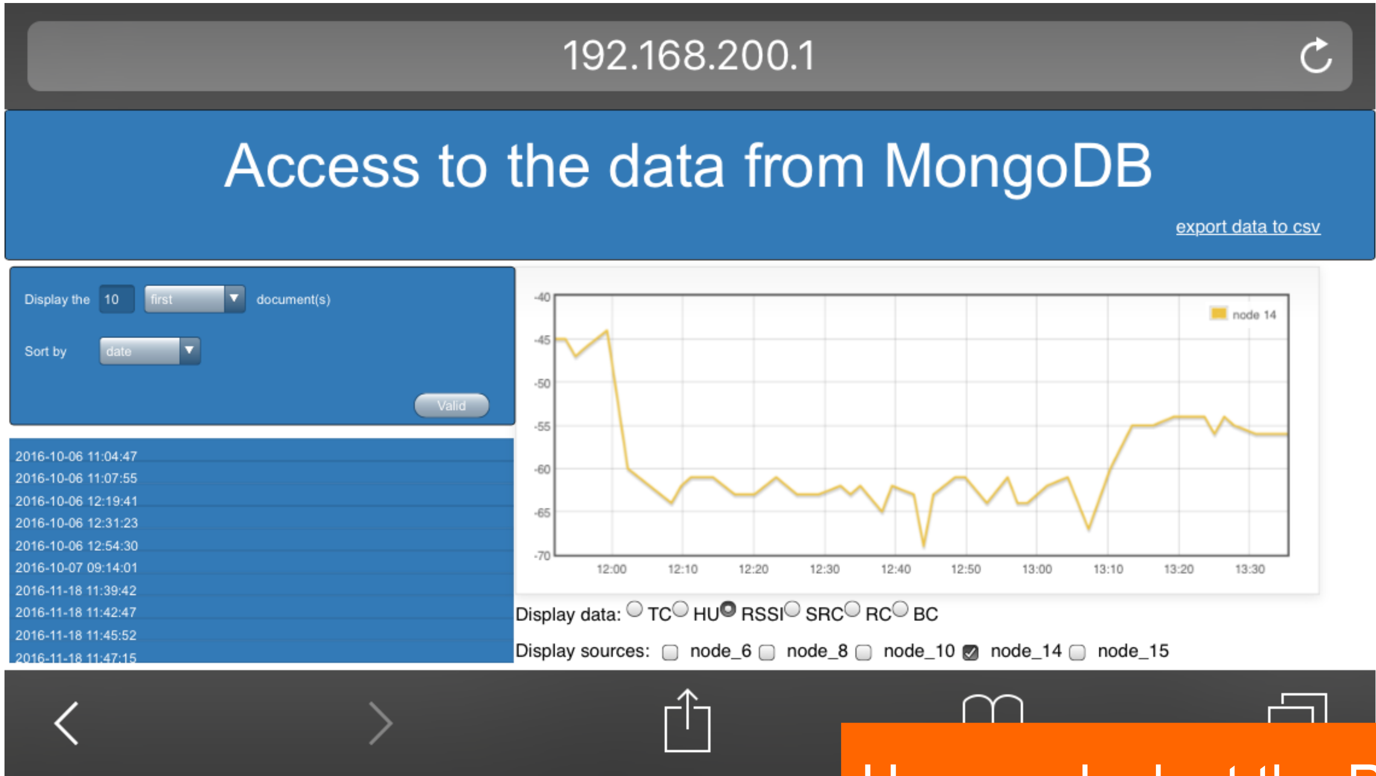
VIEW DATA FROM THE LOCAL WEB SERVER



You can check Beacon Counter to see for gaps (packet losses) and also check the time between 2 beacons. If BC suddenly comes back to 0 it may mean that the system has been reset (disconnected and reconnected?) which is likely an alarm!

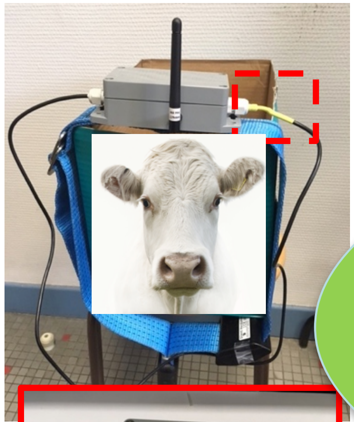
Visualize it on your smartphone!

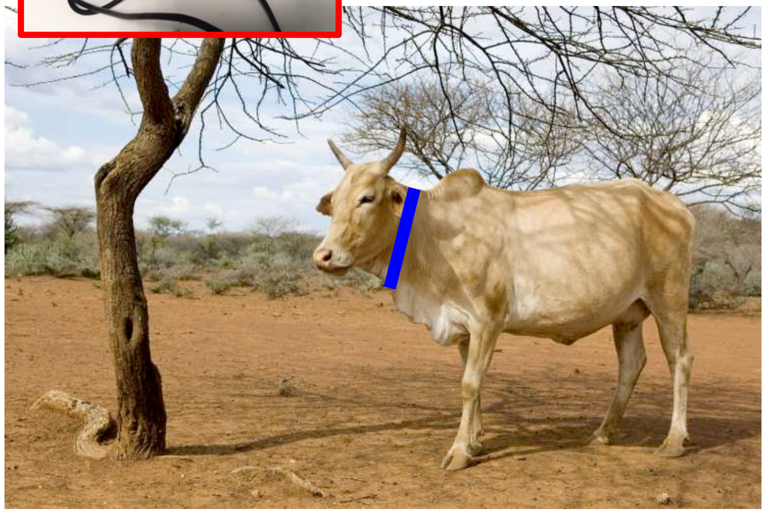
- Don't forget to join the WAZIUP_PI_GW_xxxxxxxxxx WiFi

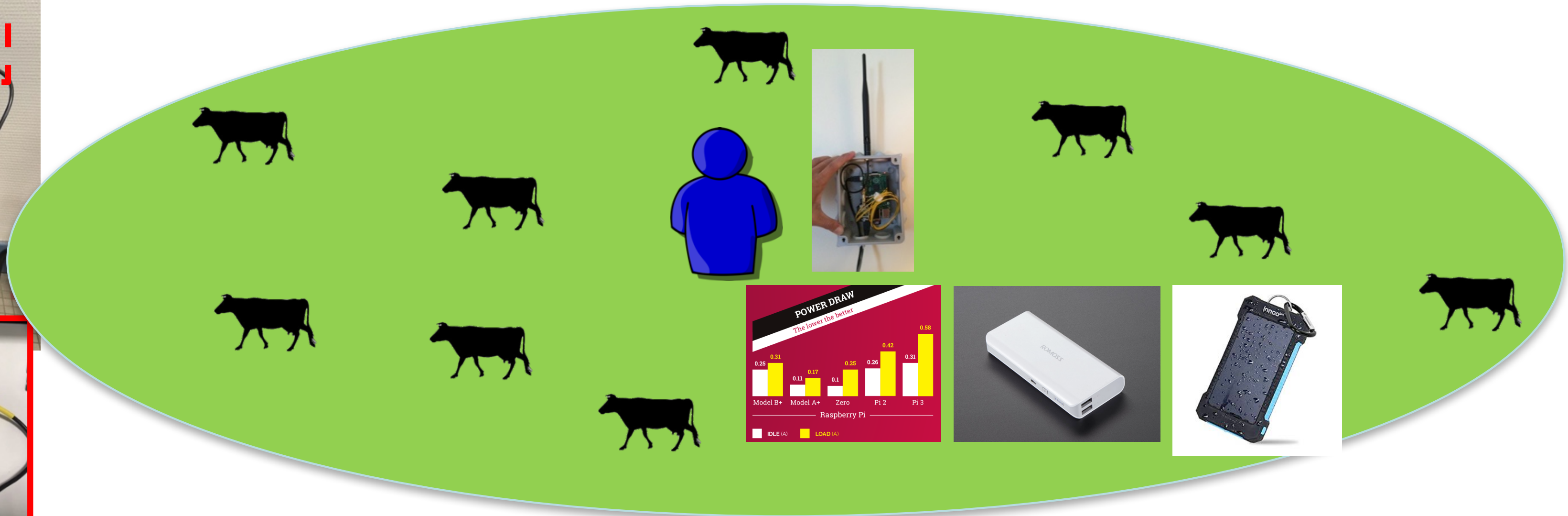


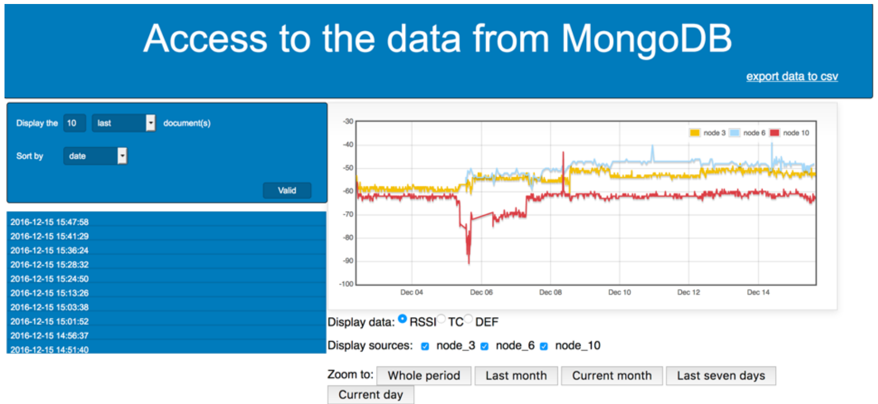
Here we look at the RSSI field

Use an autonomous gateway









Adding a GPS module

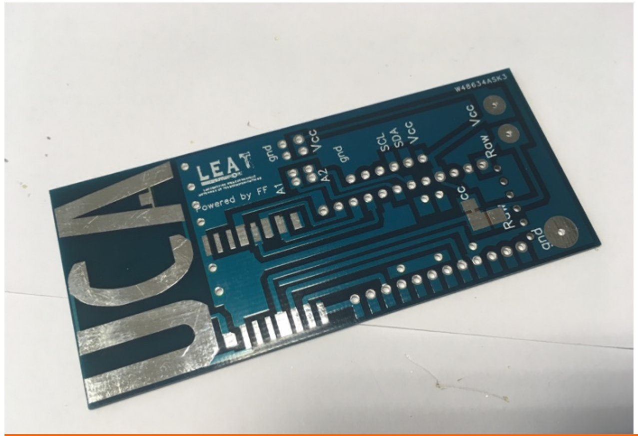
GPS ADD-ON AND NEW FEATURES

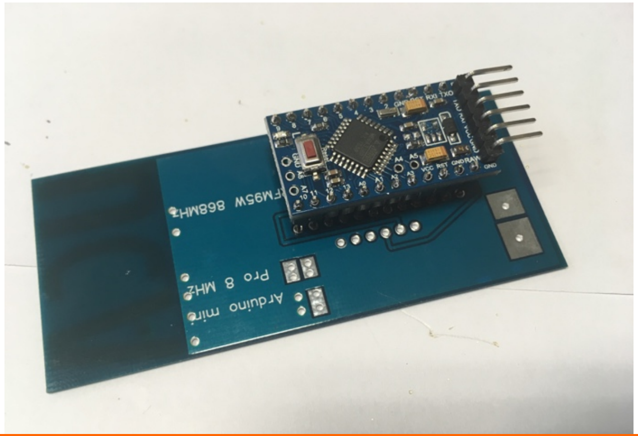
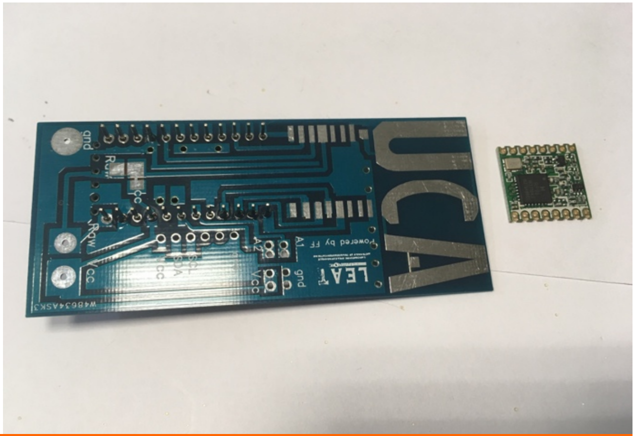
- Here, a GPS module is added to get an accurate positioning system
  - We use ublox-based GPS modules
  - These modules can be easily obtained from Chinese integrators
  - Ublox NEO-6M are sufficient and their cost is lower: about 6€.
  - Ublox NEO 7/8 are supported and have lower power consumption

- We will also describe some additional design
  - using a PCB with integrated antenna and a smaller radio module (HopeRF RFM95W)
  - injecting directly 3.2v-3.6v to further reduce power consumption

PCB WITH INTEGRATED ANTENNA

- A PCB with an integrated antenna can be used
  - To facilitate the integration of the Arduino Pro Mini and a smaller LoRa radio module (HopeRF RFM95W)
  - To avoid having an external antenna that can be very fragile

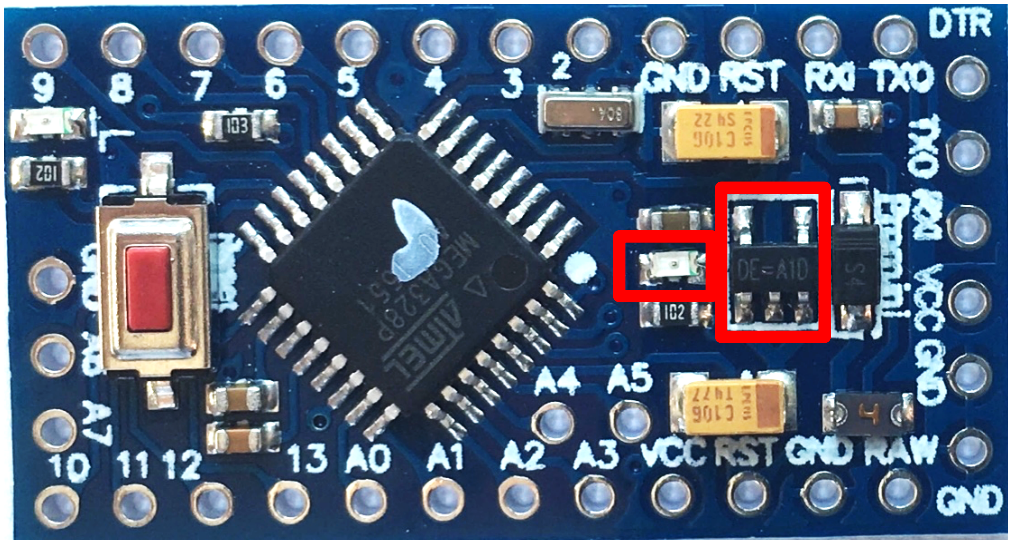


PCB and antenna designed by Fabien Ferrero from LEAT, University Côte d'Azur, France
The PCB file is available as open source and the making can be ordered from Chinese manufacturers

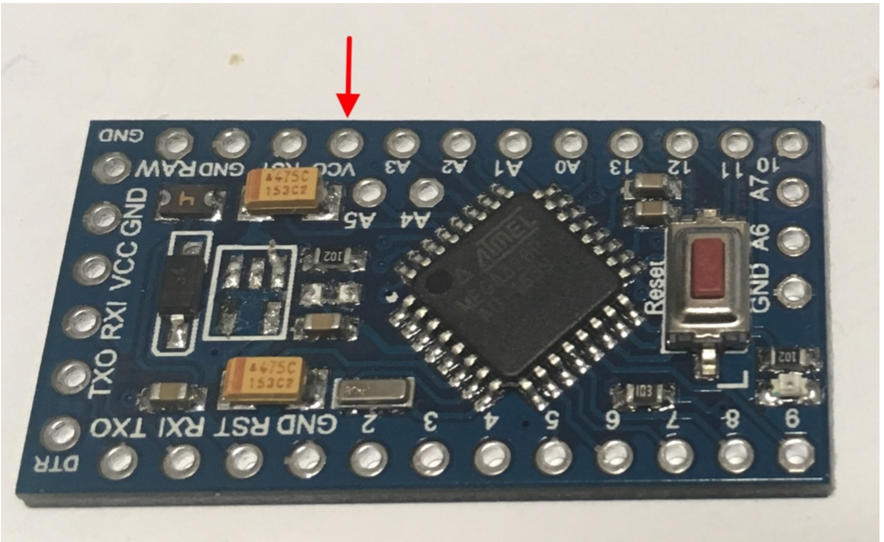
Extreme low power (1)

- You can greatly reduce the power consumption by removing the power led (left box) and the voltage regulator (right box)
- See our "Extreme low-cost & low-power LoRa IoT DIY" video tutorial at https://www.youtube.com/watch?v=2_VQpcCwdd8

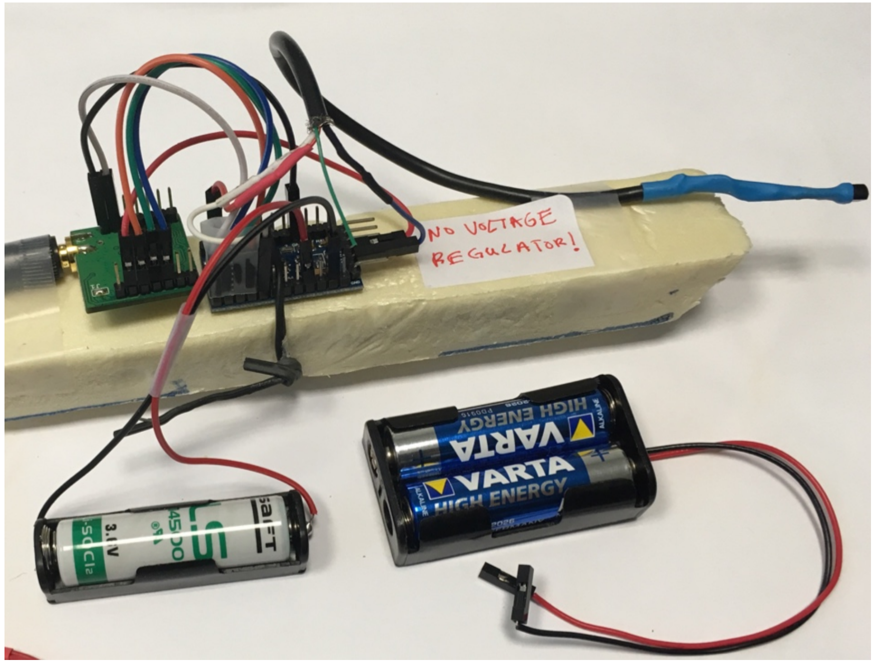


Extreme low power (2)

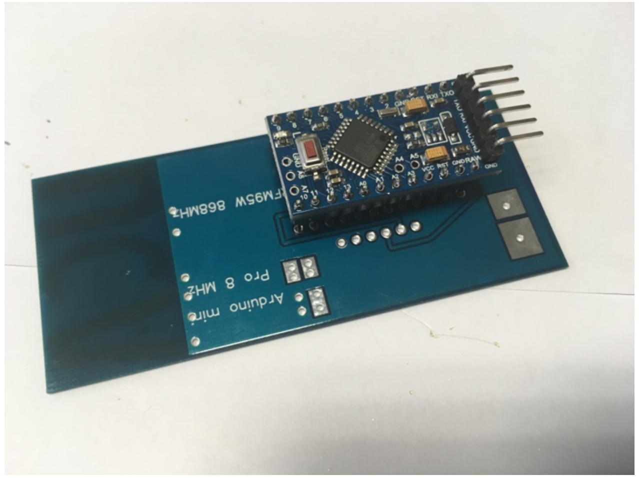
- Then, inject directly between 3.2v and 3.6v into the VCC pin instead of the RAW pin
- You can use 2 AA batteries (3.2v) or a 3.6v Lithium battery

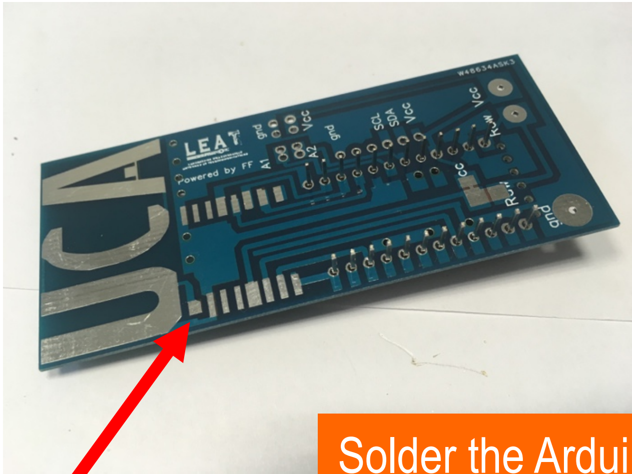


WARNING, do not inject more than 3.6v when the regulator is removed! You can destroy your board!



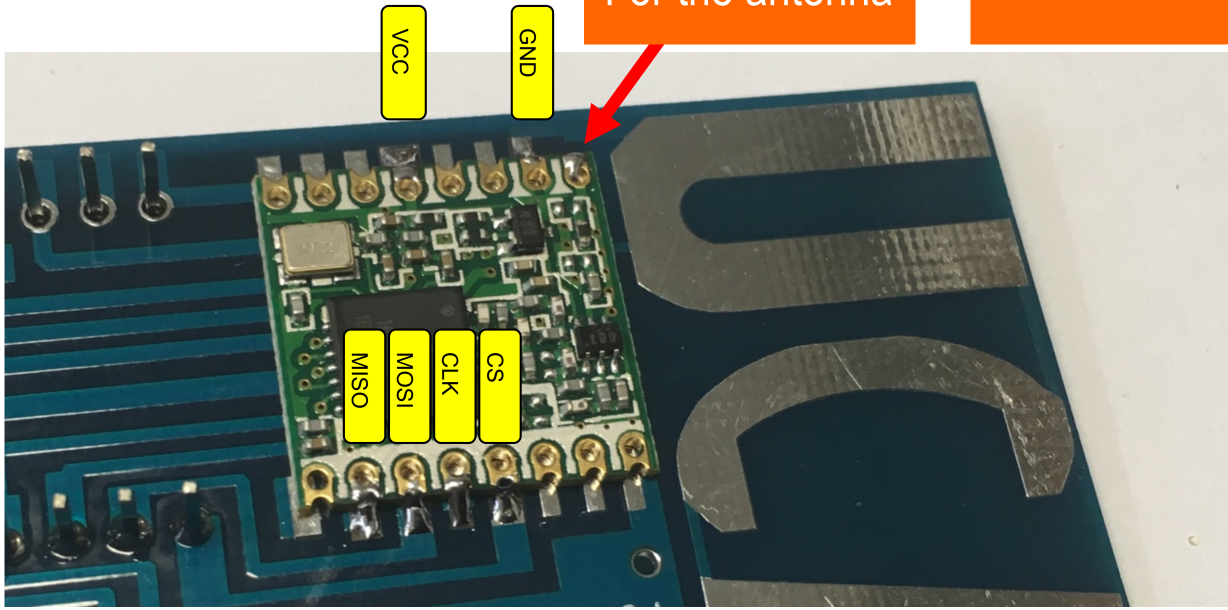
Soldering the Pro Mini and the RFM95W



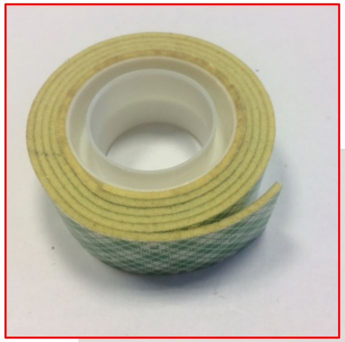


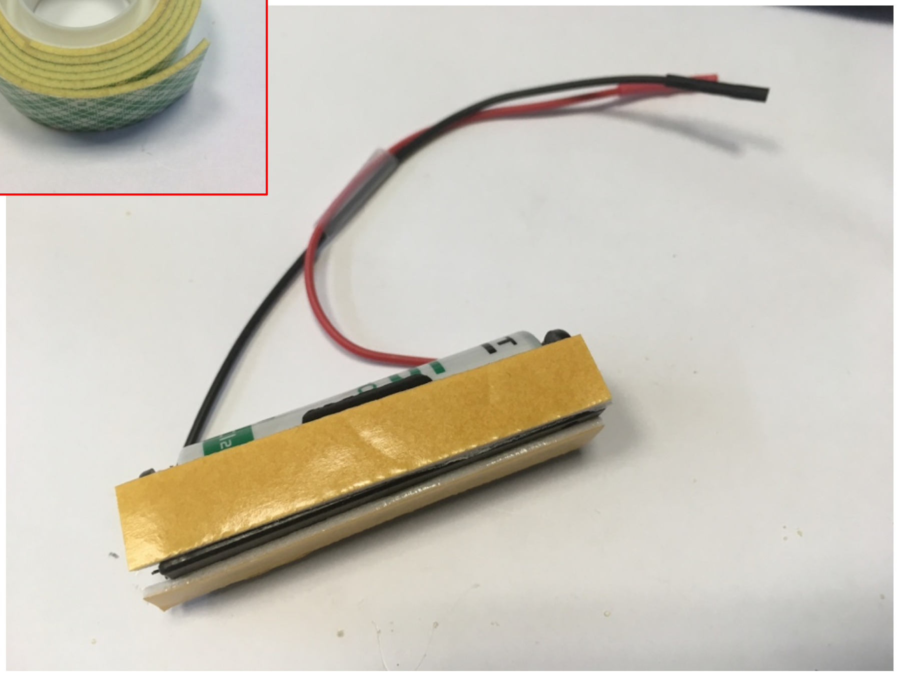
Solder the Arduino Pro Mini and the radio module, all wire connections are realized by the PCB

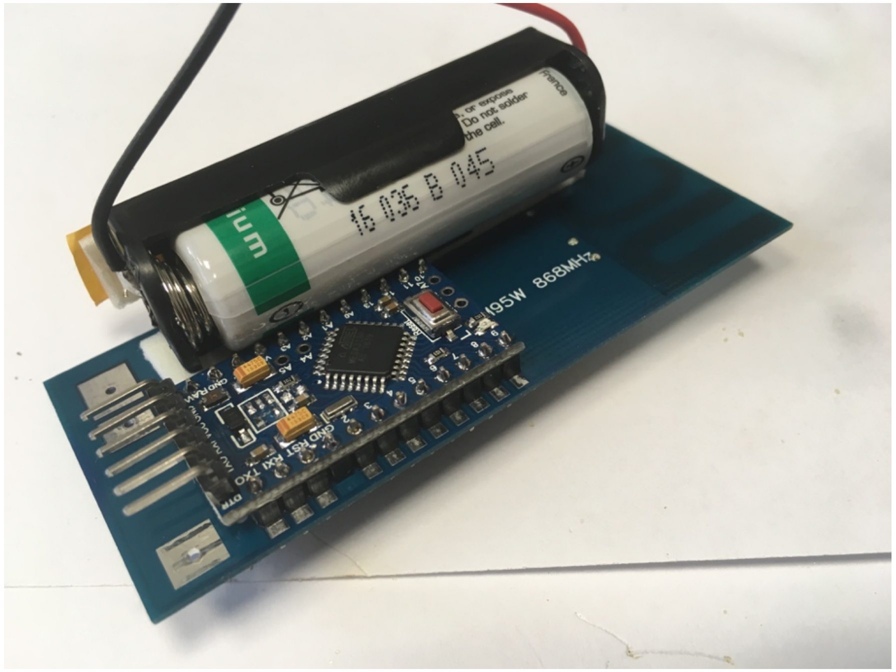
For the antenna



Adding the battery pack



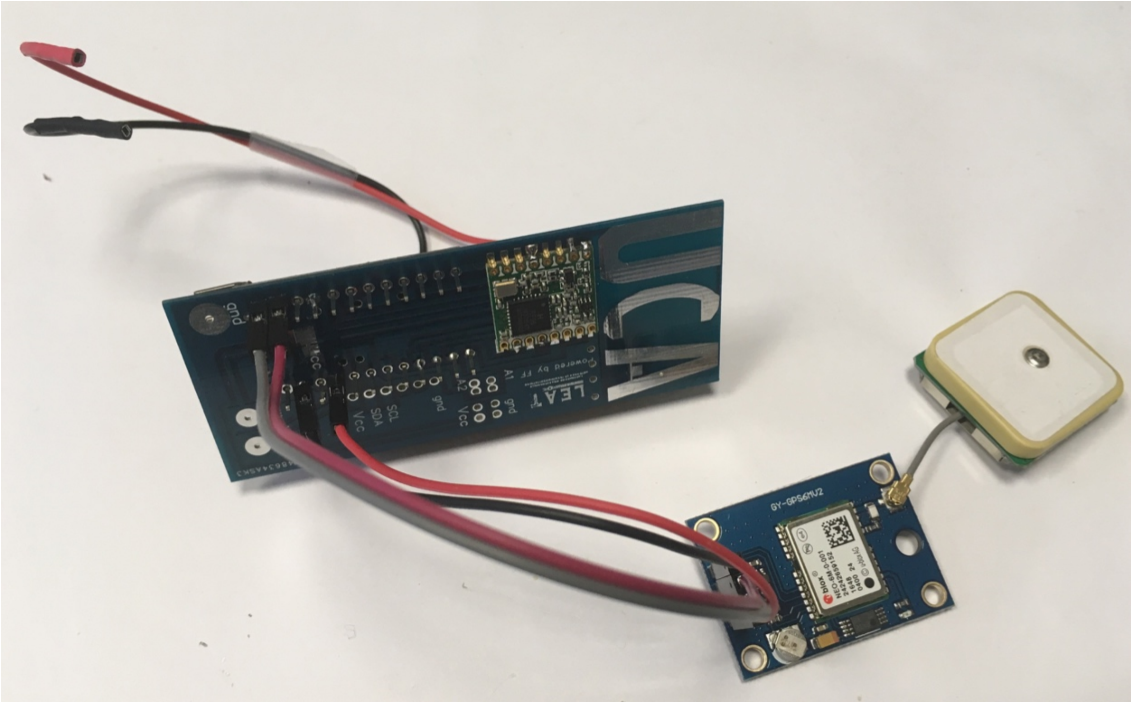




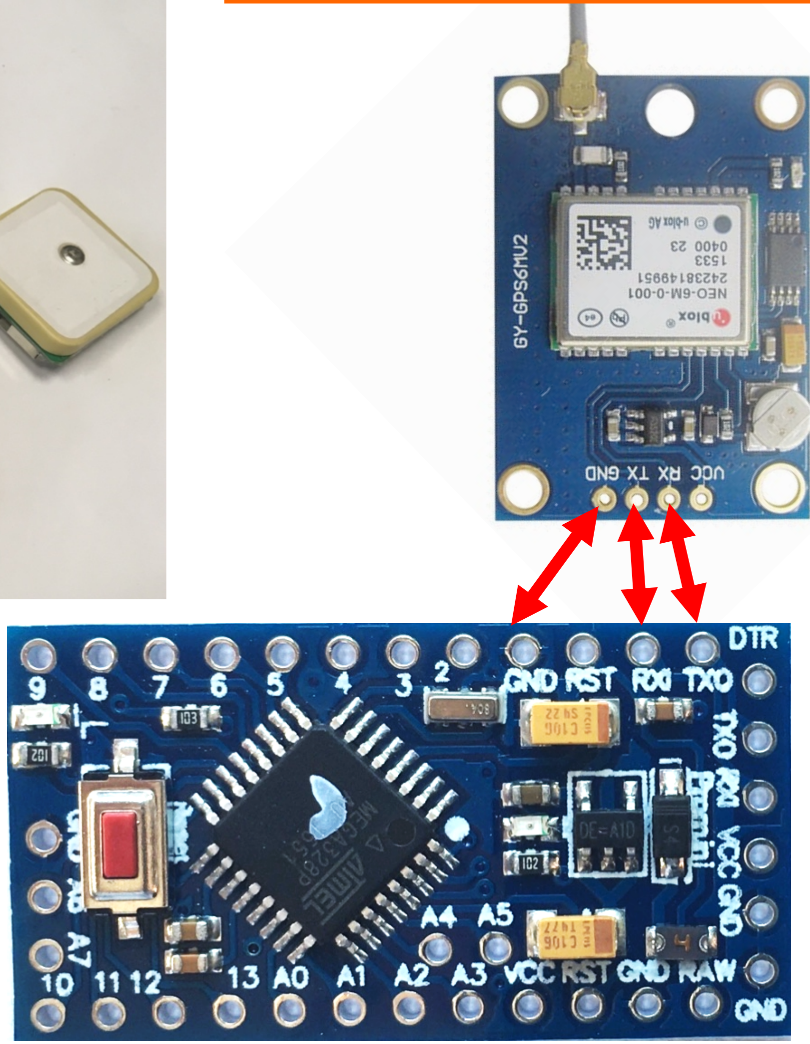
Here we use a single 3.6v battery. Use double-side tape, those used to fix mirror on walls…

… to fix the battery pack to the PCB. Do not put the battery to close to the antenna part.

Connecting the GPS

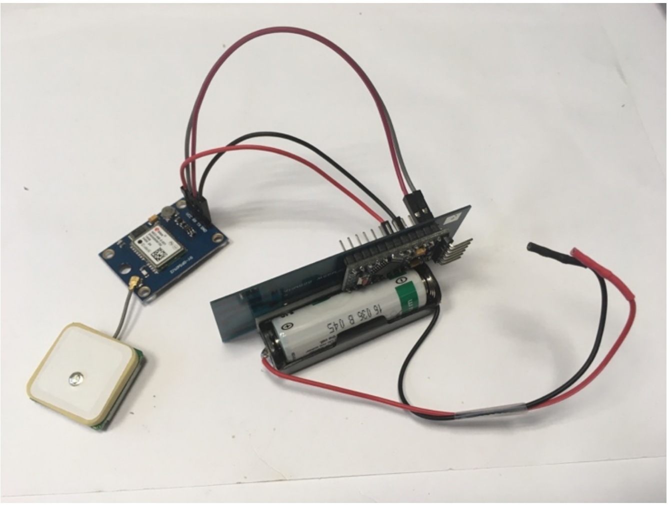


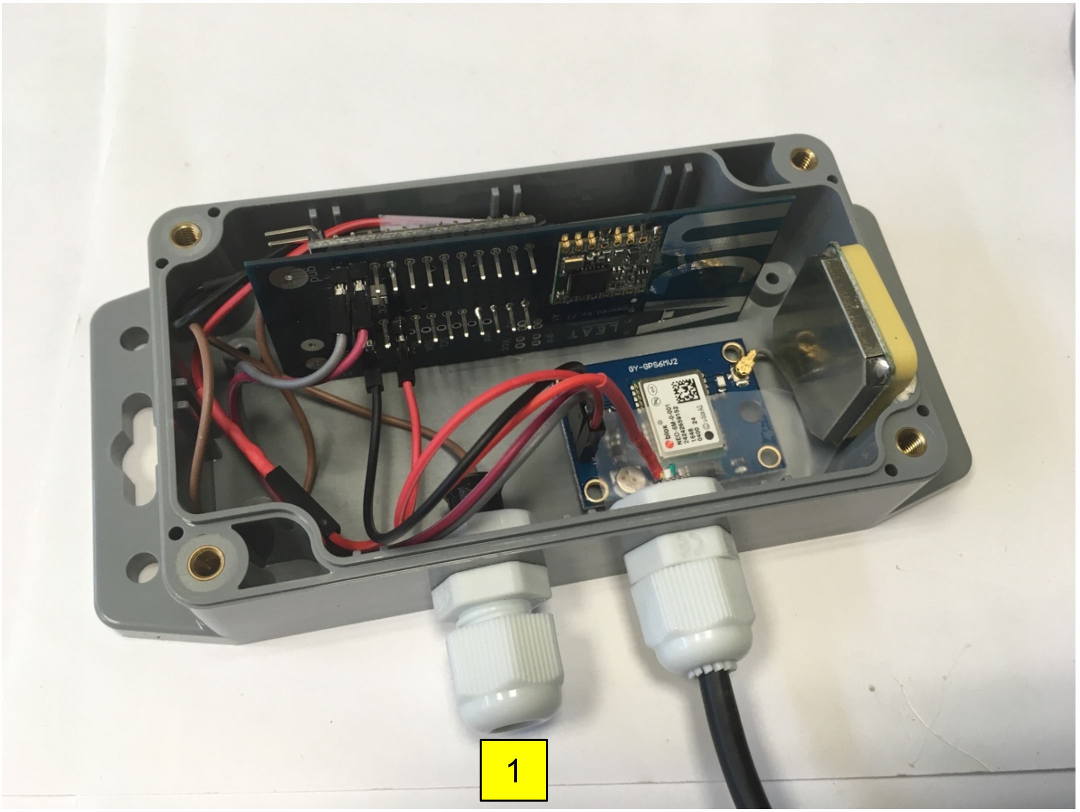
GPS uses serial TX/RX connection



The PCB shows extra GND and VCC pins to easily connect the GPS module

Putting everything in a box



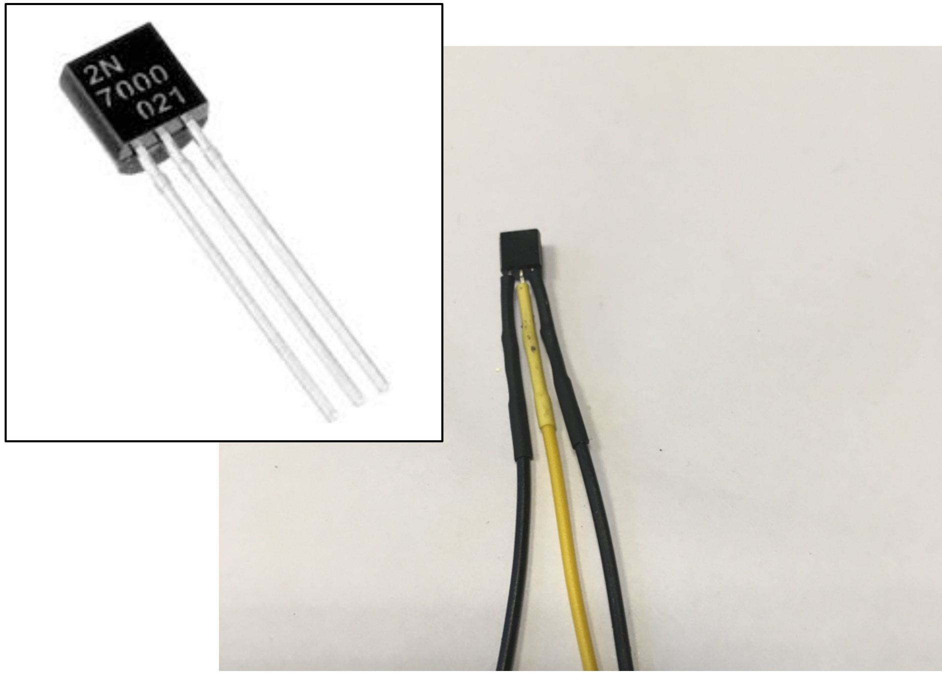


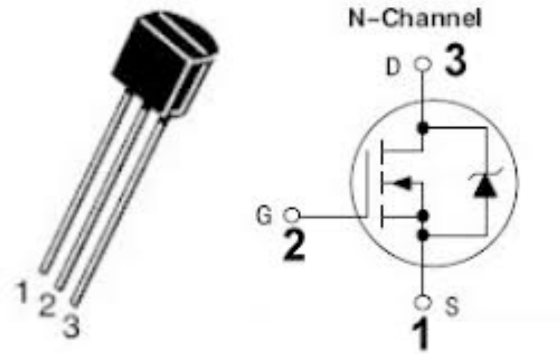
Like in the first version, when connecting the male connector MC to the female connector FC 1 , the board will be powered and will start sending periodic GPS beacons

Low power for GPS

- The best option to reduce the GPS power consumption is to completly switch off the GPS between 2 wake up
- Each time the GPS is powered on again it can be considered as a cold start but a GPS fix can usually be obtained in about 35s
- We will use digital pin 8 coupled to a MOSFET transistor to realize a very low cost and simple ON/OFF switch

Using the 2N7000 MOSFET

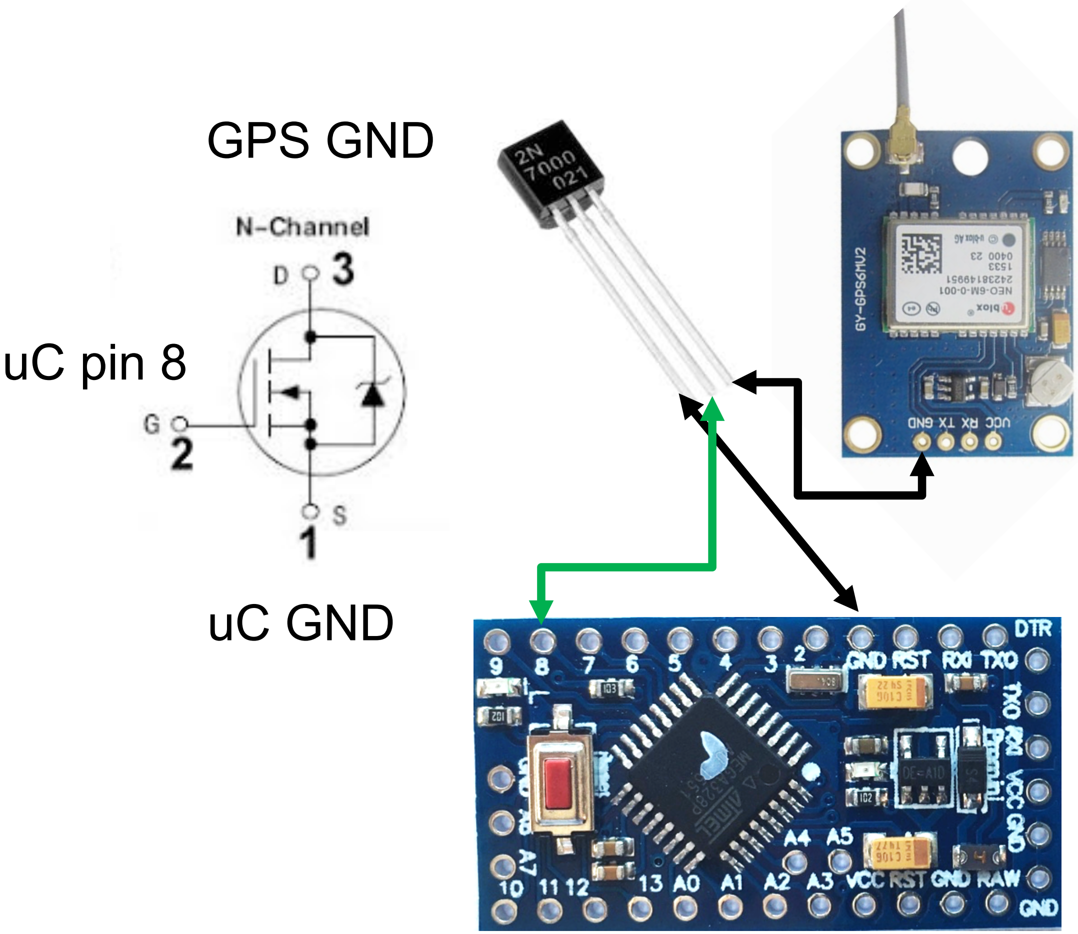




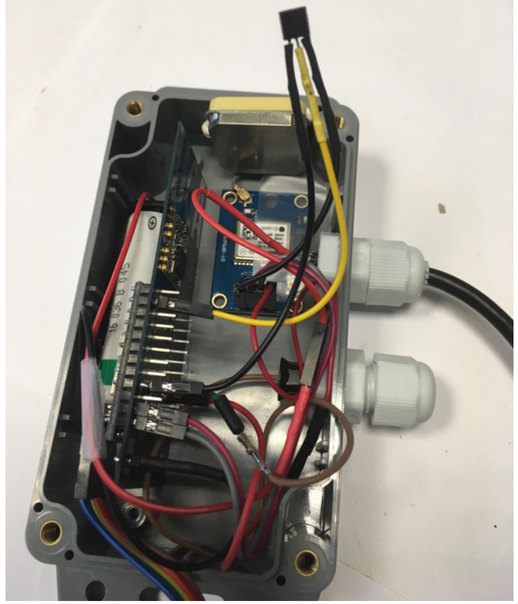
We use a 2N7000 MOSFET transistor where the maximum current of 200mA is sufficient to power the GPS (about 60mA in acquisition mode)

The basic principle is as follows: when the voltage V_{GS} applied on the Gate pin (G) is sufficient, current can flow between the Drain (D) and the Source (S). With the 2N7000, V_{GS} is typically 2v with a max of about 3v but it can work with the output of the Arduino Pro Mini 3.3v digital pin

Connecting the 2N7000

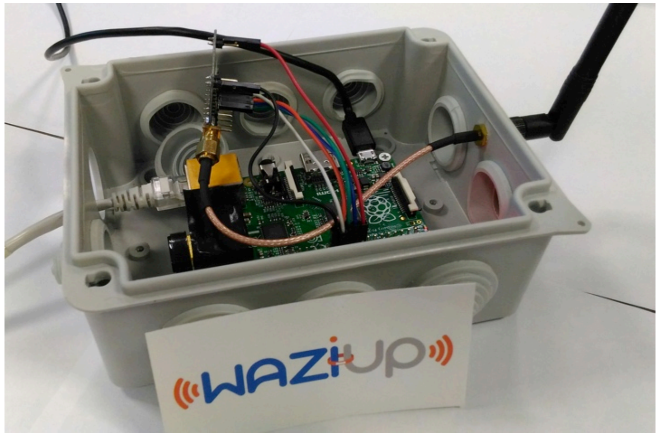


In acquisition mode, the device consumes about 50mA (NEO 7/8) and about 75mA (NEO 6M). In sleep mode with GPS off, it consumes only 5uA!

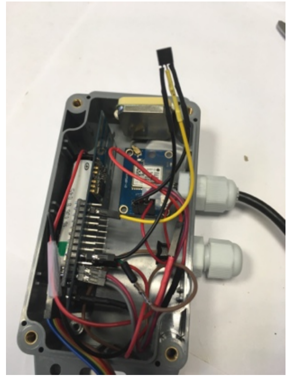


The GPS VCC is directly connected to one of the PCB's VCC that will deliver 3.3v

DEFAULT CONFIGURATION



\!BC/0/LAT/43.31408/LGT/-0.36383/FXT/32503



The default configuration in the Arduino_LoRa_GPS example is:

Use LoRa mode 1, node short address is 15
Digital pin 8 drives the MOSFET transistor
Send GPS beacons to the gateway every 20 minutes
Maximum GPS acquisition time is 180000ms (3min), will go to sleep after that if no GPS fix
BC (beacon counter) starts at 0, increases by 1 at each beacon, returns to 0 after 65536 beacons
FXT is the time to get the GPS fix in ms
If no GPS fix, then the beacon is \!BC/0/LAT/0/LGT/0/FXT/-1
If FXT<0 then LAT and LGT fields are not valid

What's next?

- Use beacon information to detect abnormal situation
- When a beacon is received, the following information can be collected
  - Beacon number
  - RSSI of the packet
- Perform field tests for correlation between LoRa BW & SF parameters and range
- Set BW & SF for maximum usage range and use statistical methods to reduce false alarms
- Use machine learning, fingerprinting?
  - Can look at RSSI analysis studies for geolocalization purposes
- Investigate whether accelerometer data can improve the system