



# INTEL-IRRIS

Intelligent Irrigation System for Low-cost Autonomous Water Control  
in Small-scale Agriculture



# INTELLIGENT IRRIGATION SYSTEM FOR LOW-COST AUTONOMOUS WATER CONTROL IN SMALL-SCALE AGRICULTURE



## Building the Intel-IrriS LoRa IoT platform Part 1: soil sensor device



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



# Review: Technology components



# Review: Low-cost sensors



- Build on low-cost, low-power IoT expertise
- Increase accuracy of low-cost sensors by automatic and remotely controlled procedures for advanced calibration
- Enable deployment of several complementary low-cost sensors
- Include agricultural models / knowledge with corrective & predictive analytics

# Review: Smart embedded control

- Build on low-cost embedded & open IoT gateway expertise
- Implement the “Intelligent Irrigation in-the-box” with "plug-&-sense" approach
- Model complex water-soil-plant interaction
- Embed Decision Support System (DSS) and disruptive Artificial Intelligence (AI)
- Integration of various knowledge streams
- Fully autonomous

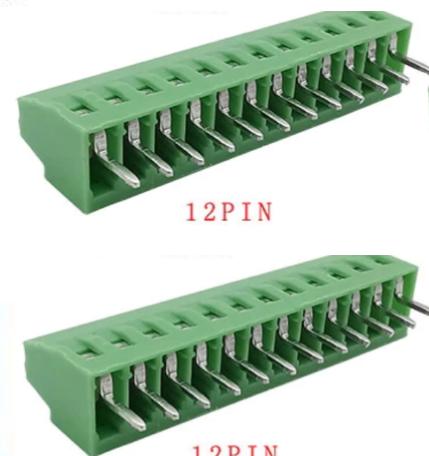
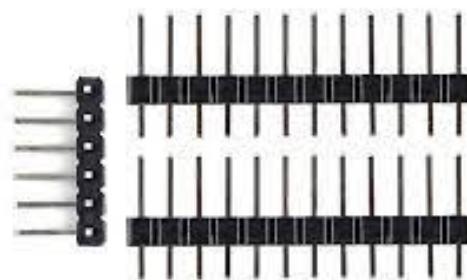
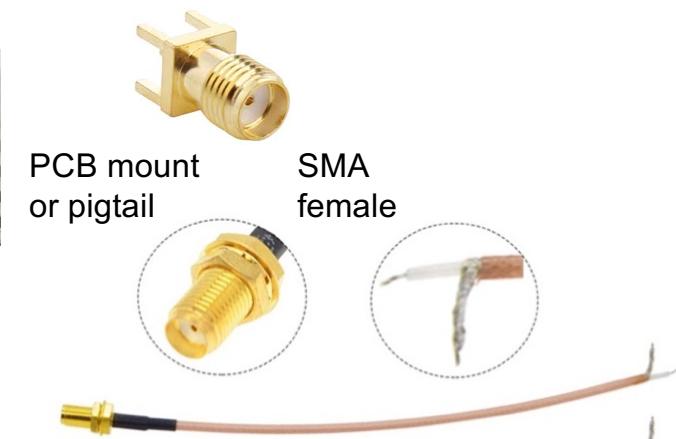
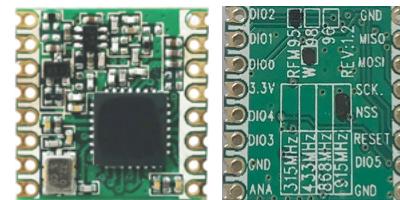
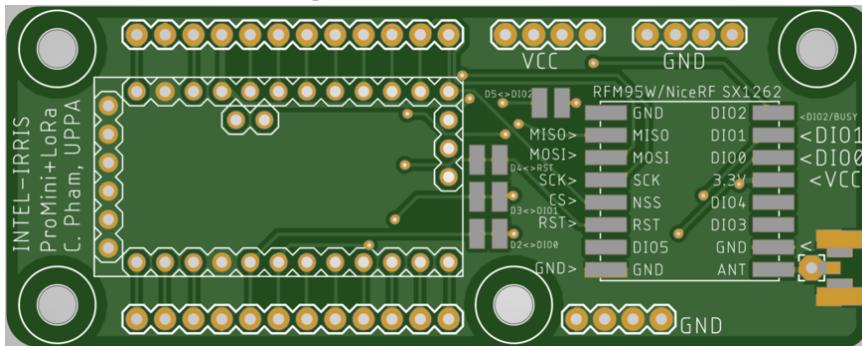


# Review: Starter-kits

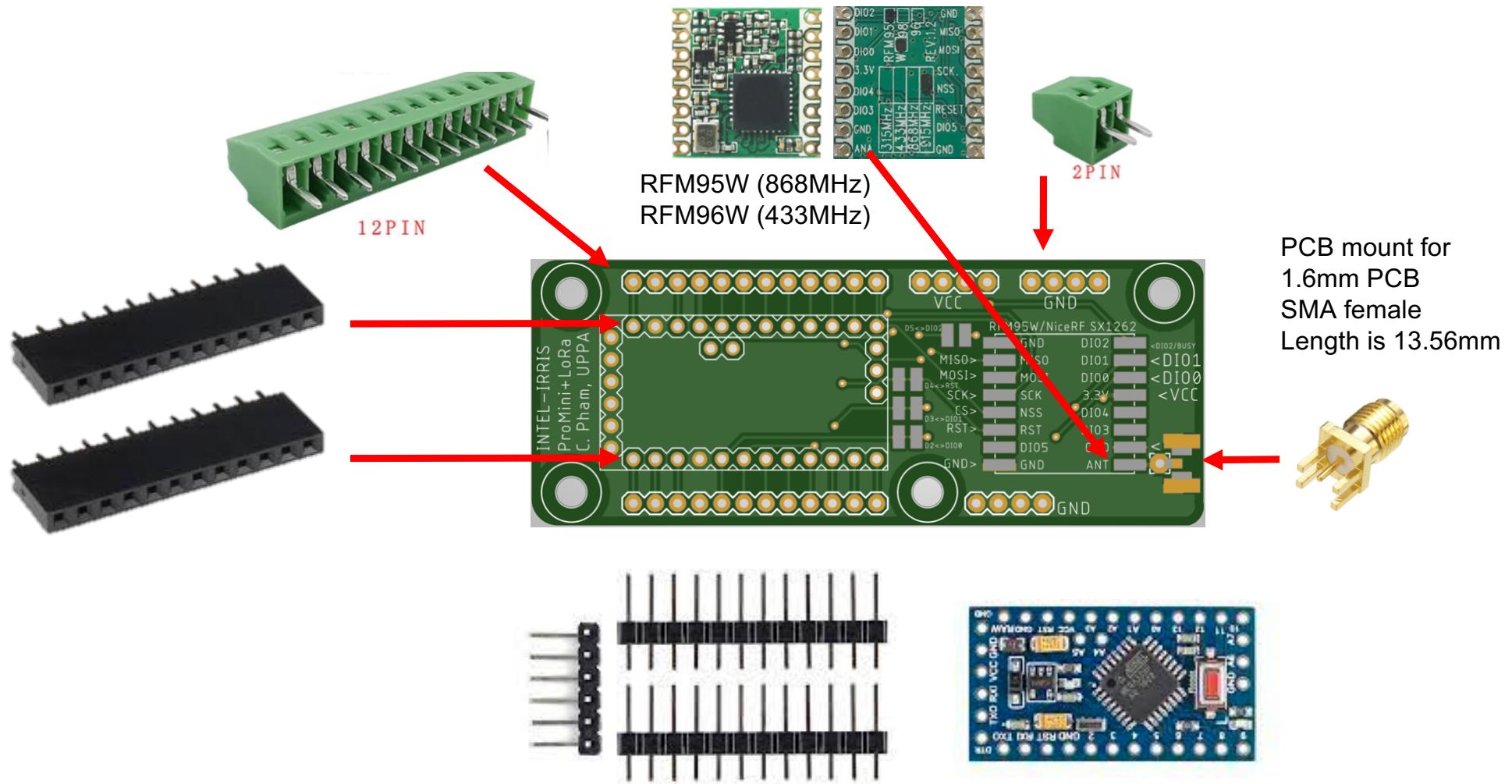
- "Intelligent Irrigation in-the-box", "plug-&-sense"
- At least 100 starter-kit will be distributed



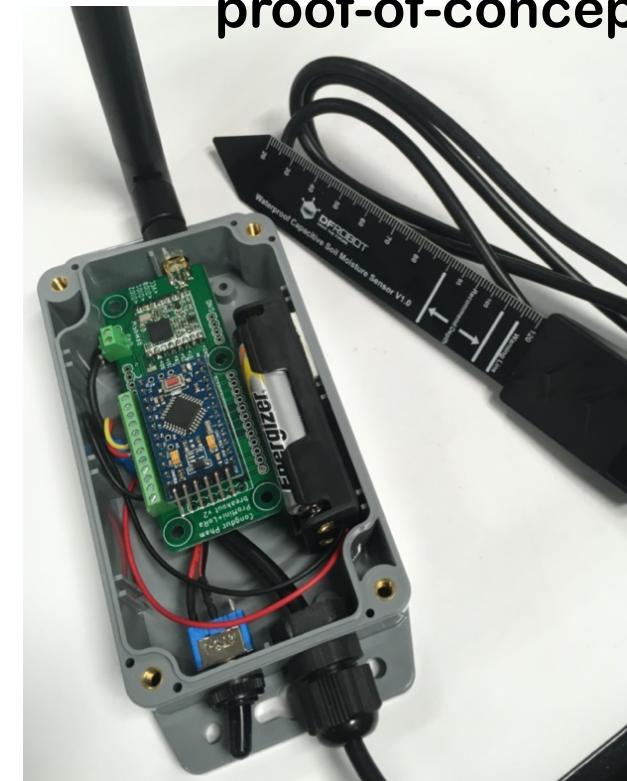
# Soil sensor: electronic parts complete version



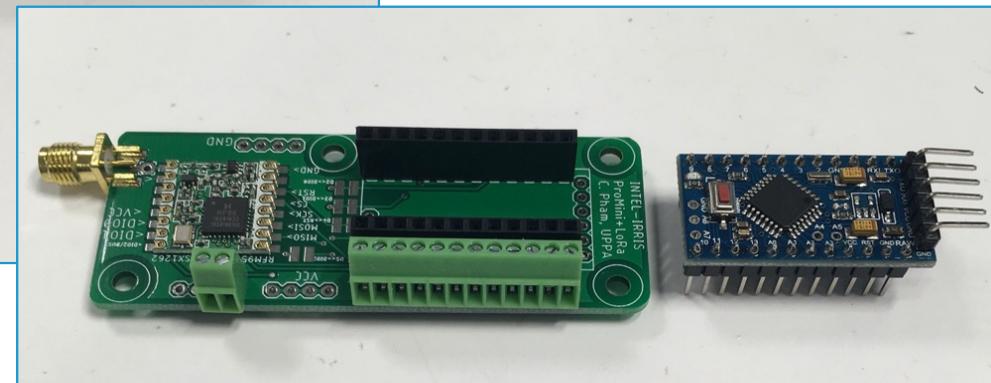
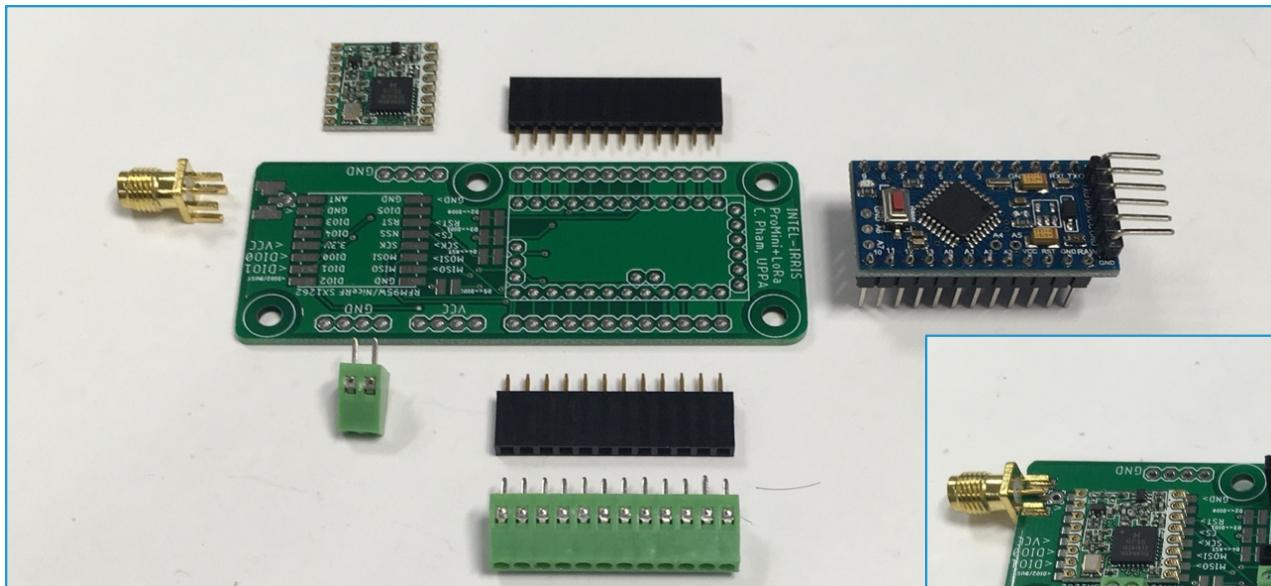
# Soil sensor: electronic parts starter-kit version



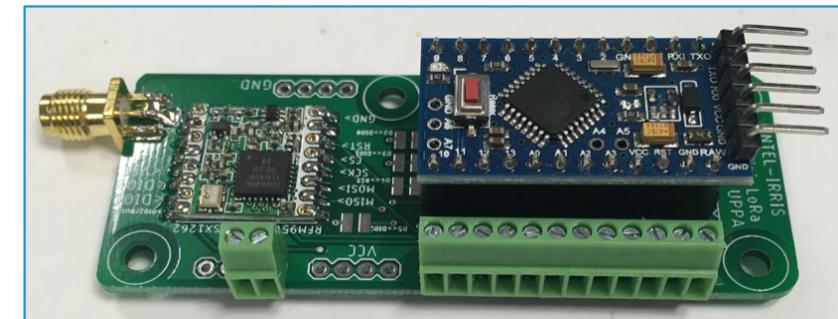
# Soil sensor: enclosure & integration



# Assembling the PCB board

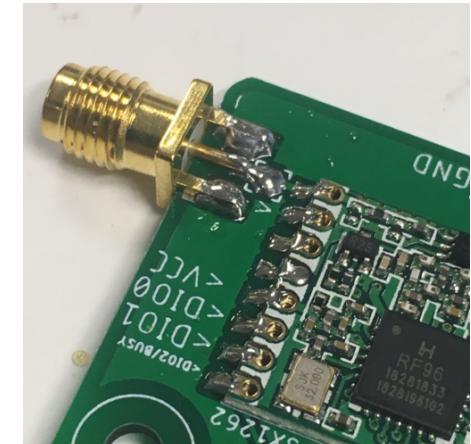
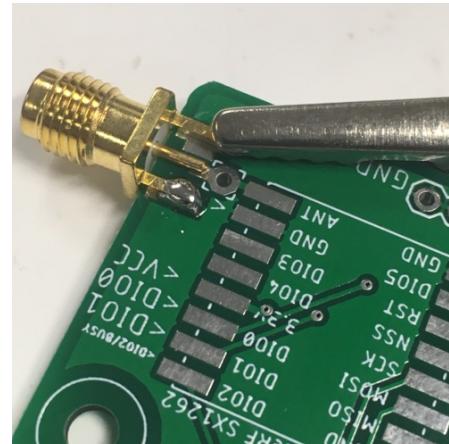
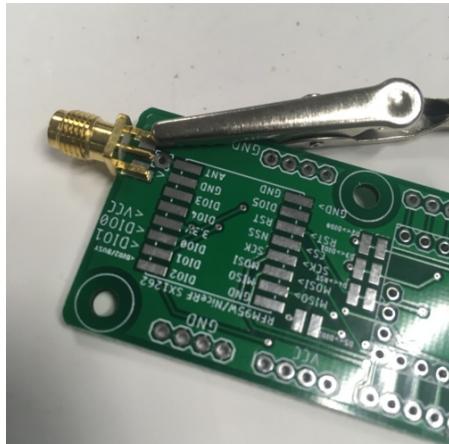


This dedicated video will show  
how to solder all the components  
Video n°1: <https://youtu.be/3jdQ0Uo0phQ>



# Updates on Video n°1

- Clip the SMA connector to the PCB to correctly align the connector before soldering one of the GND pad, then remove clip and finish soldering
- Or, gently press SMA connector pads so that insertion on PCB is harder and SMA connector will not move during soldering
- Make sure that the GND pads and the central pin are not touching each other, in doubt, do a continuity check





# Good soldering

- It is important to have good soldering to avoid any issue when deploying the devices
- Check this great soldering tutorial from Collin's lab
  - <https://youtu.be/RB4P1HTmwLc>

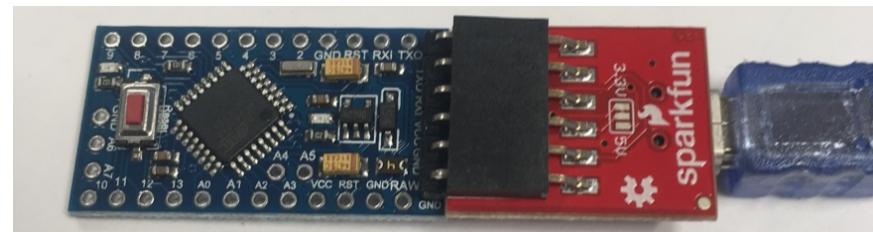
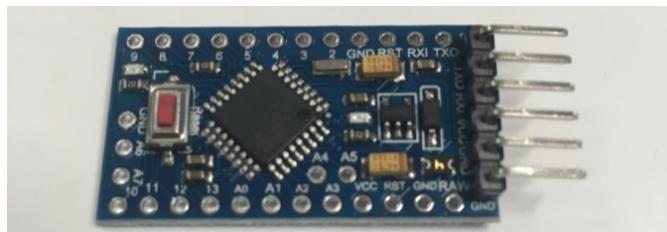


- Just put the required amount of lead, too much lead is not good!
- It is not a bad idea to train on soldering before making the device and not waste electronic components

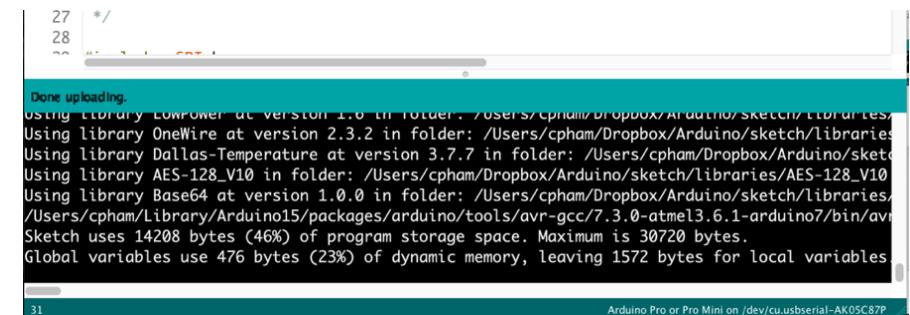


# Check the Arduino

- First, just solder the programming header



- To flash the Arduino board with the INTEL-IRRIS code
- See slides from "Getting the software: Arduino IDE" to "Uploading to your board"
- If uploading is successful, then you can continue with soldering the remaining header pins, otherwise, take another board as this one may have hardware issue



```
27 */  
28  
Done uploading.  
Using library LowPower at version 1.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/  
Using library OneWire at version 2.3.2 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/  
Using library Dallas-Temperature at version 3.7.7 in folder: /Users/cpham/Dropbox/Arduino/sketch/  
Using library AES-128_V10 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/AES-128_V10  
Using library Base64 at version 1.0.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/  
/Users/cpham/Library/Arduino15/packages/arduino/tools/avr-gcc/7.3.0-atmel3.6.1-arduino7/bin/avr  
Sketch uses 14208 bytes (46%) of program storage space. Maximum is 30720 bytes.  
Global variables use 476 bytes (23%) of dynamic memory, leaving 1572 bytes for local variables  
31  
Arduino Pro or Pro Mini on /dev/cu.usbserial-AK05C87P
```

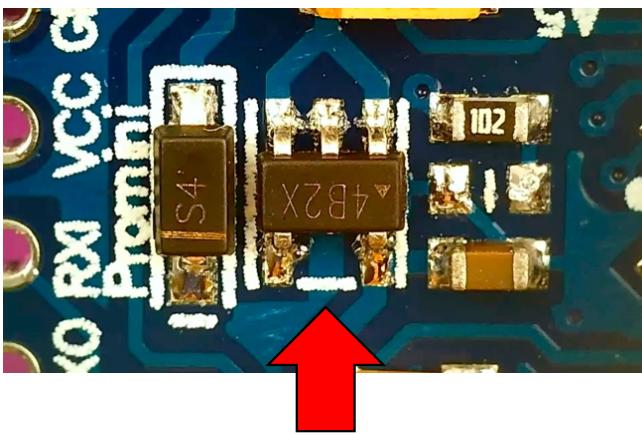
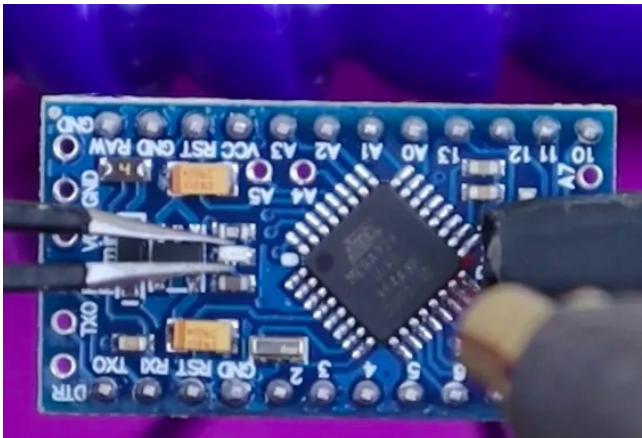


# Never transmit without antenna

- NEVER, NEVER transmit without an antenna
- Doing so can damage the radio module
- If your board is already connected to the radio module and you need to flash the board, connect the antenna
- If you need to update the existing code and your device already run a code that transmit data, connect the antenna
- It is safer when programming the device to remove the Arduino board from the female header and program it disconnected from the radio module
- If you deploy a device, make sure that the antenna is correctly connected before powering on the device and realizing any transmission test

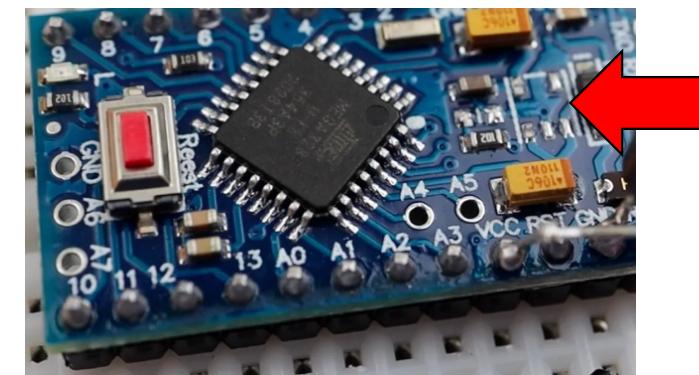


# Reduce power consumption



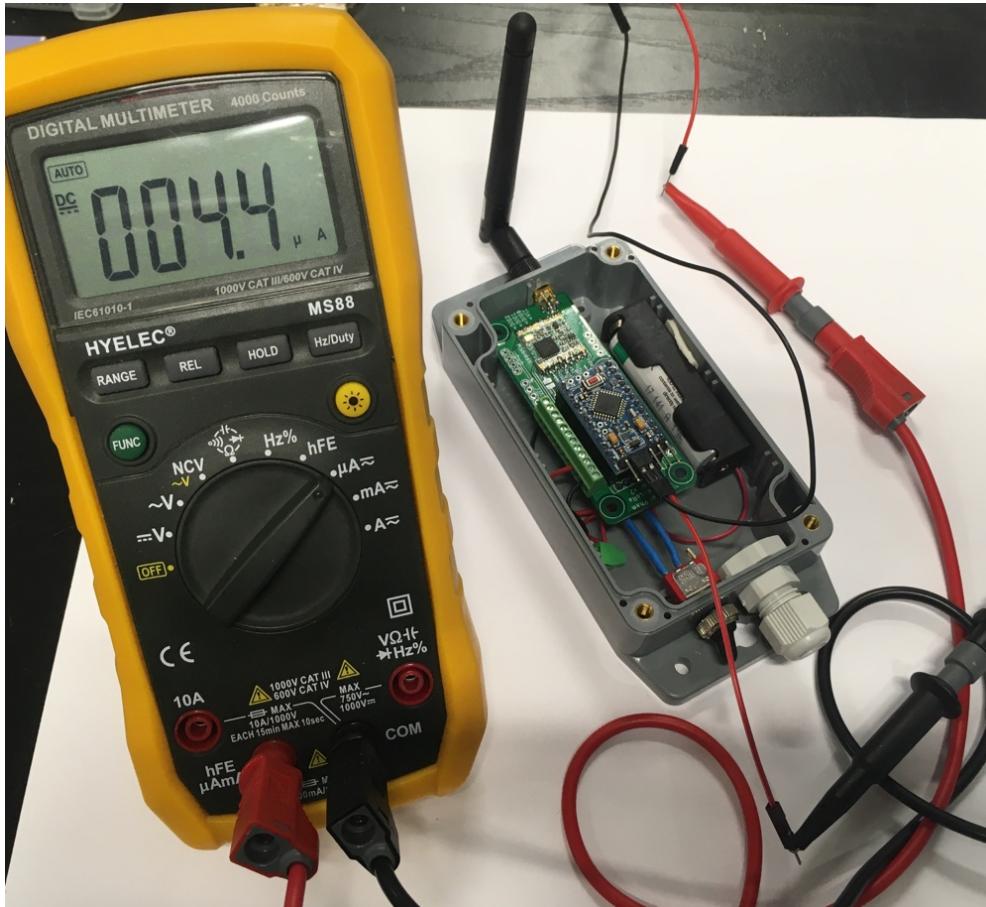
**Do not forget** to remove the power LED by just clipping it off with some wire cutters

**Do not forget** to remove the voltage regulator with a small plier



**Only inject up to 3.6V** through the VCC pin

# Power consumption in deep sleep



Measured below 5 $\mu$ A in deep sleep, between 2 active periods with transmissions

Expected autonomy with 1 transmission / hour is over 2 years with either 2 AA batteries or 1 AA 3.6V Lithium battery

# Get an enclosure for outdoor usage



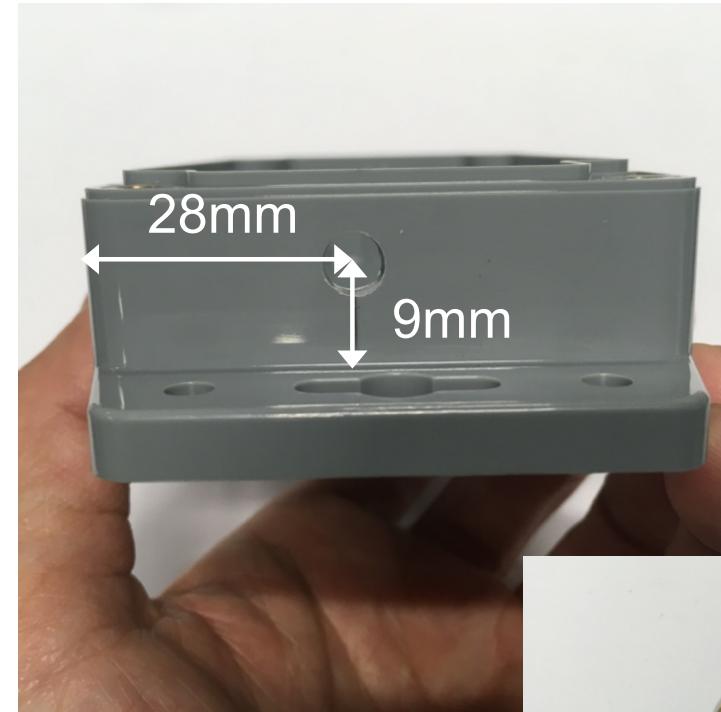
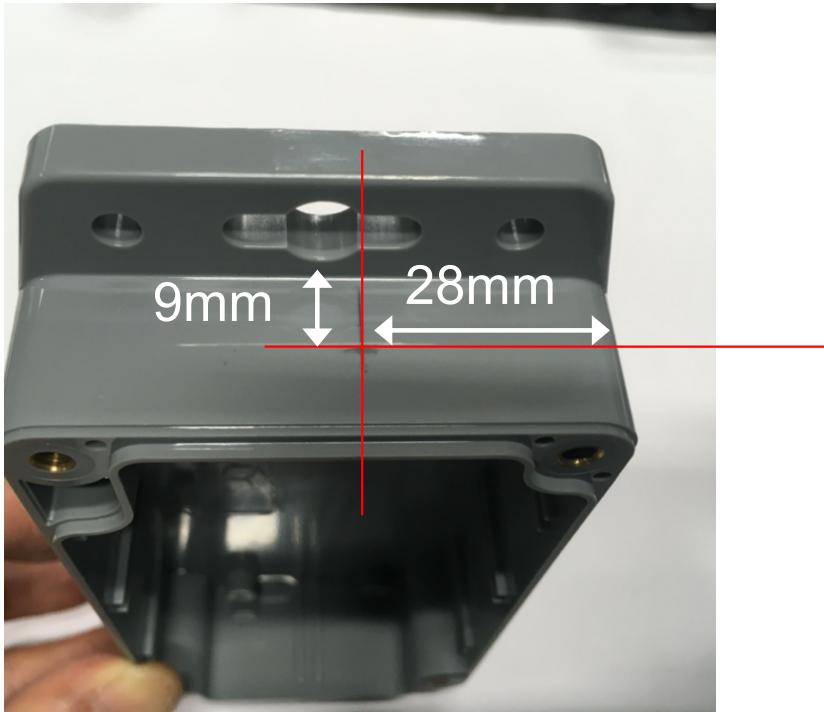
Here, it is an IP65 box which dimension is 115 x 65 x 40mm  
<https://www.gotronic.fr/art-boitier-abs-etanche-g304m-17977.htm>

# Drilling machine and drilling bits



At least a simple cordless drilling machine is necessary  
If you have a (small) bench drilling machine it is of course better  
Then you need an assortment of drilling bits for **metal**, not for  
wood nor concrete! Here, you will mainly need 7mm and 13mm bits  
It is also interesting to have step drill bits

# Drill a hole for the SMA connector



**28mm for the right edge:**

- measure from the flat side as the corner is round

**9mm from the outside bottom**

**use a 7mm drill bit for metal, not for wood nor concrete!**

# Remove unwanted plastic part



we need to remove the plastic reinforcement part for this particular enclosure  
use a flat cutter for instance to remove and smooth the inside part  
(a small plier can be used first to remove most of the part)

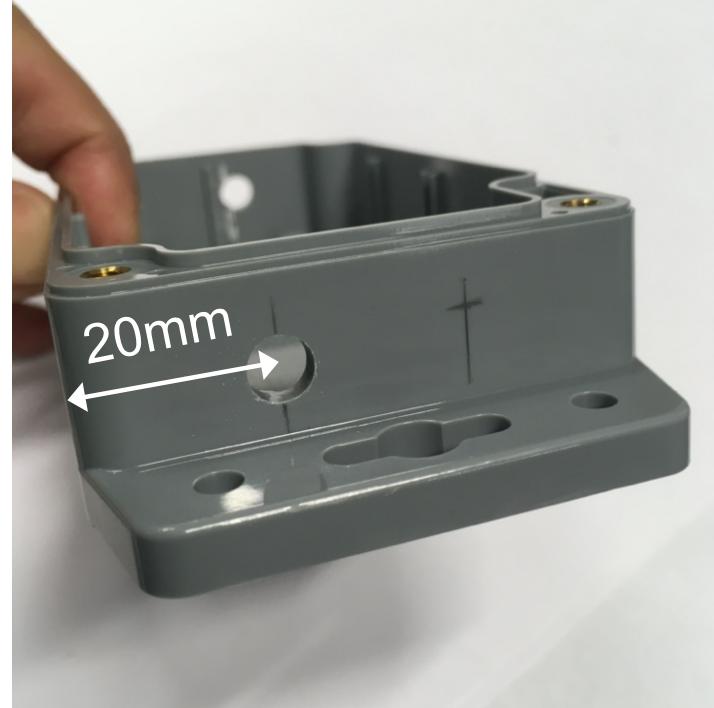
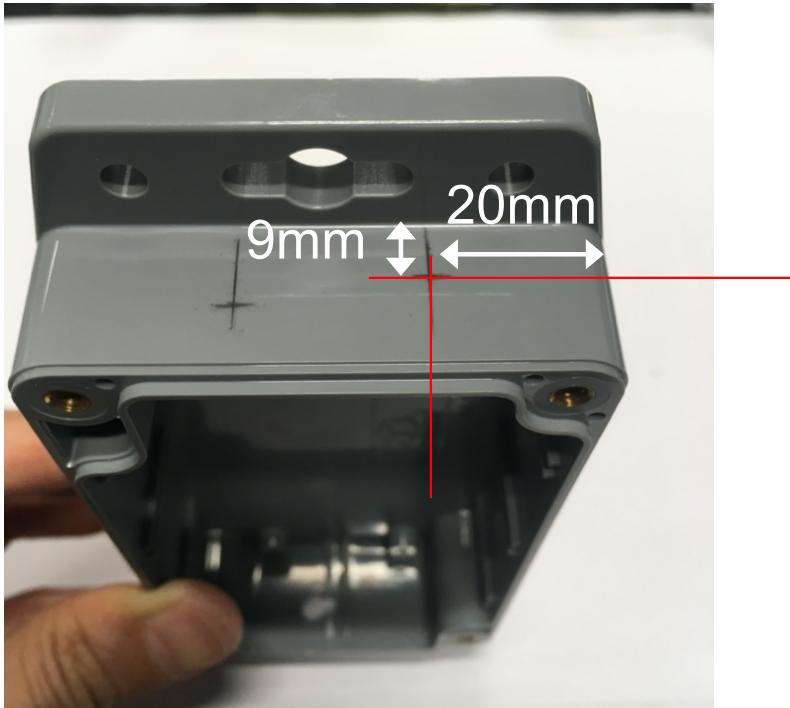
# Placing the PCB board



the PCB board can be placed, with the SMA connector going through the hole

both 2-AA (left) and 1-AA (right) battery holder can be used

# Drill a hole for the external switch



**20mm for the right edge:**

- measure from the flat side as the corner is round

**9mm from the outside bottom**

**use a 7mm drill bit for metal**

# Drill a hole for the cable gland



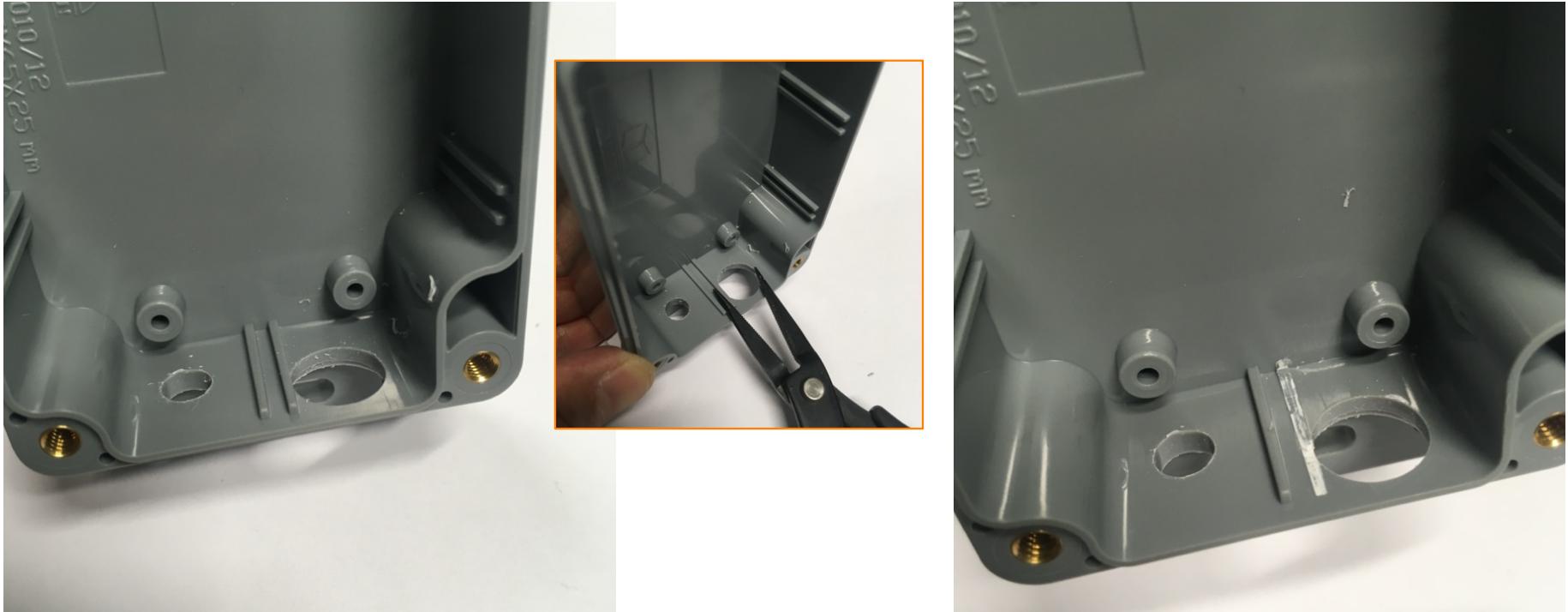
41mm for the right edge:

- measure from the flat side as the corner is round

9mm from the outside bottom

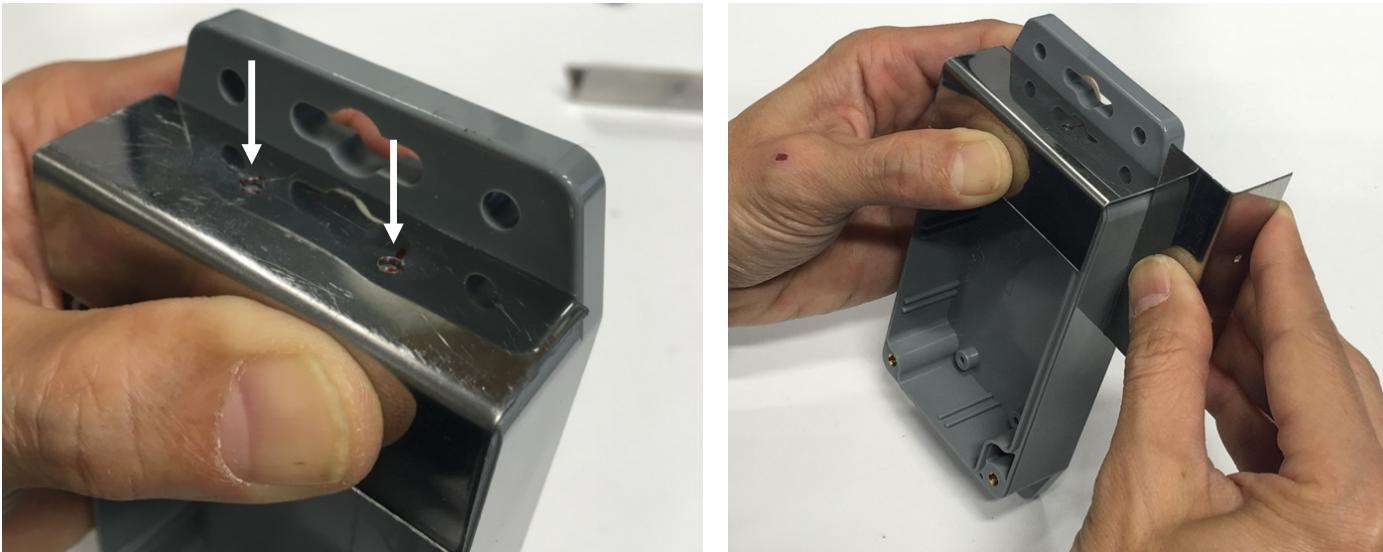
use a 13mm drill bit for metal

# Remove unwanted plastic part



again, we need to remove the plastic reinforcement part for this particular enclosure  
use a flat cutter for instance to remove and smooth the inside part  
(a small plier can be used first to remove most of the part)

# Going for larger production



If you need to prepare many enclosures, it may be faster to first make an assembly jig (here a piece of metal) for marking the holes

But be sure to precisely align the edge of the jig with the edge of the enclosure

# Placing switch and cable gland

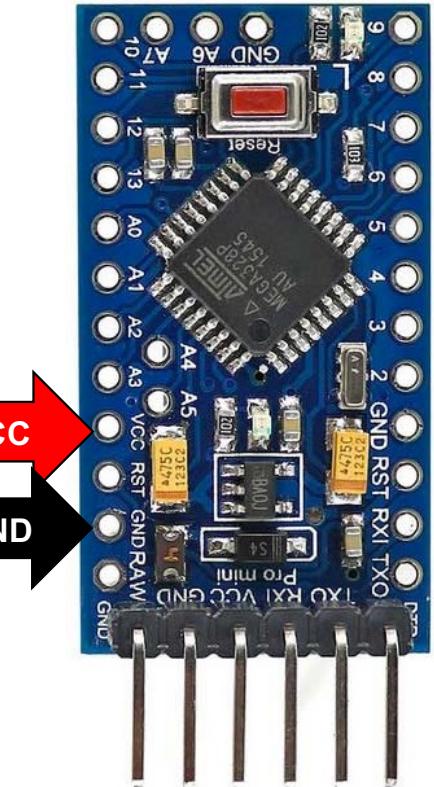
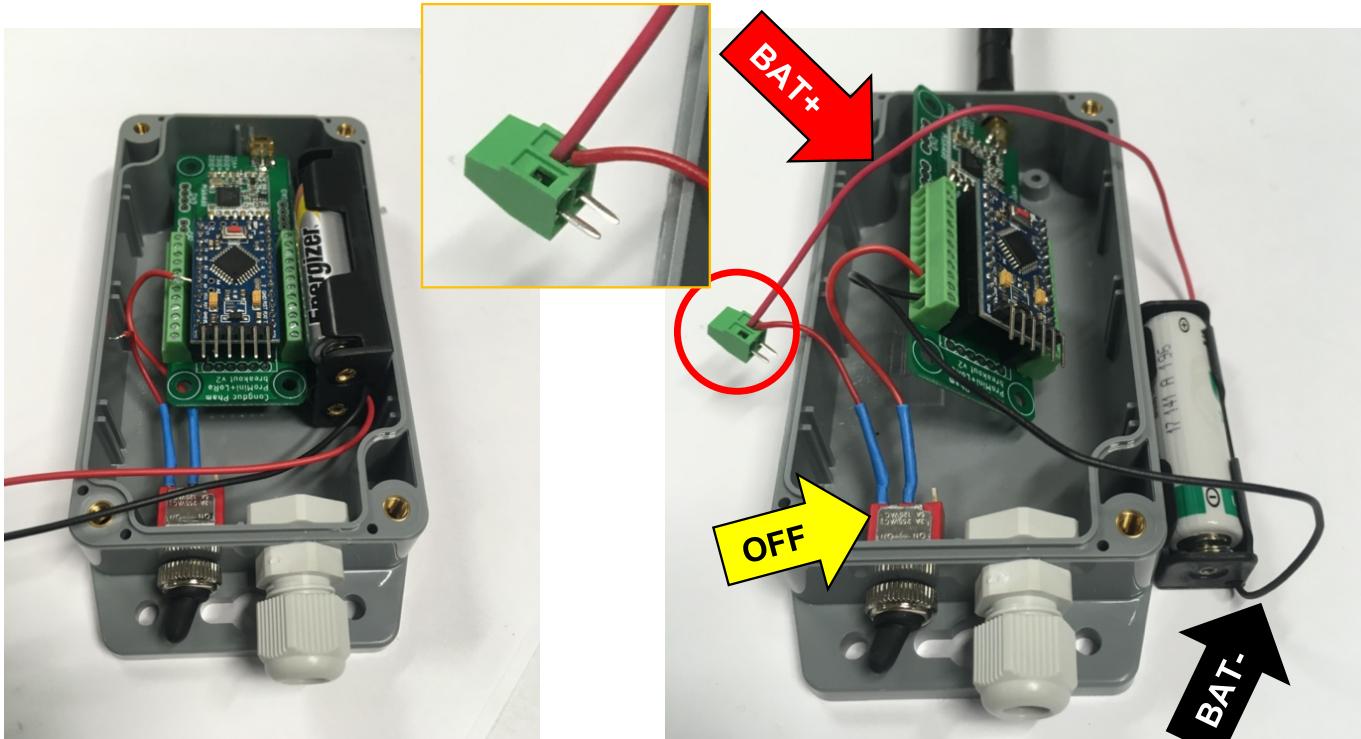


test that everything is OK

the switch has a water-proof cap

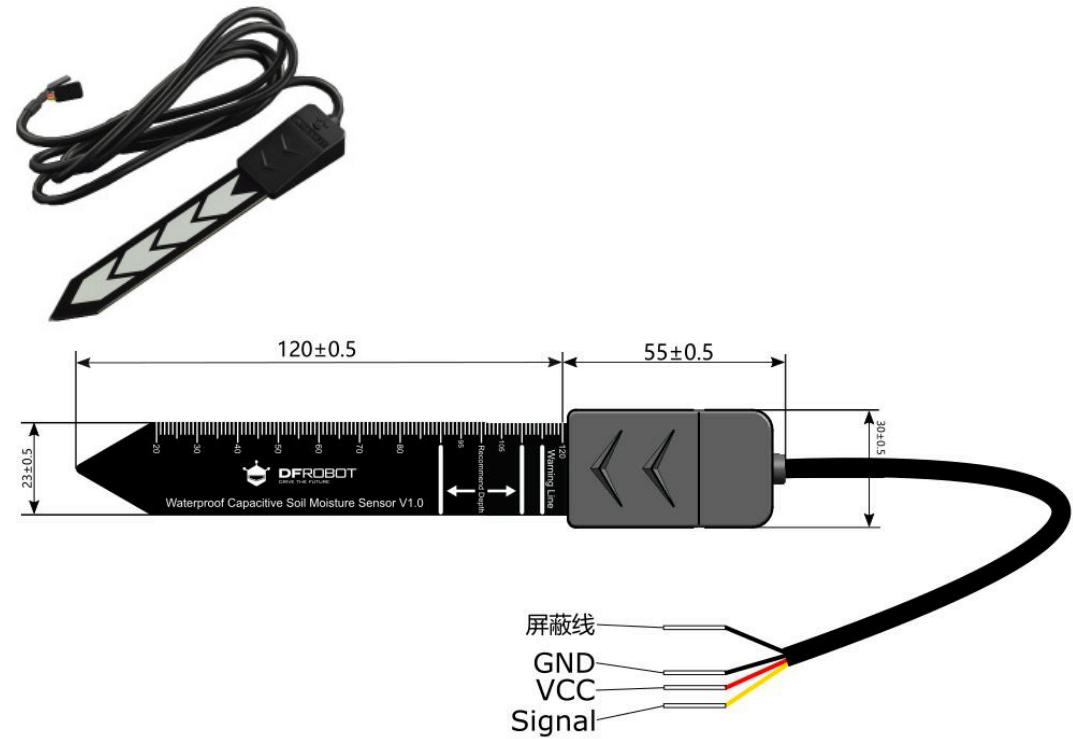


# Connecting switch, battery & board



Connect together switch "off" pin (left) with BAT +  
 Connect directly switch "on" pin (middle) to board's VCC  
 Connect directly BAT – to board's GND  
 Toggling the switch to the right will then power the board

# Wire the SEN0308 sensor

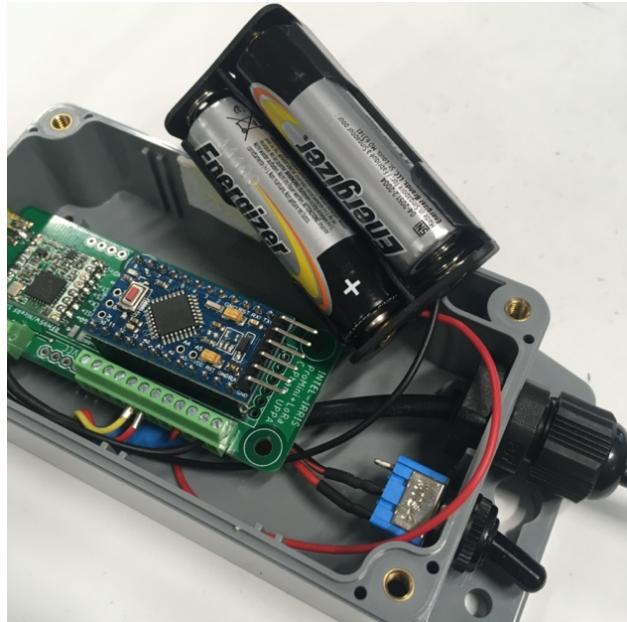


Insert sensor's wire through cable gland

Connect SEN0308's wires to board:

- VCC to board's A1
- GND to board's GND (there are 2 GND wires)
- Signal to board's A0

# Putting it altogether



You can fix the battery pack with double-side tape, e.g. those used to fix mirrors on wall

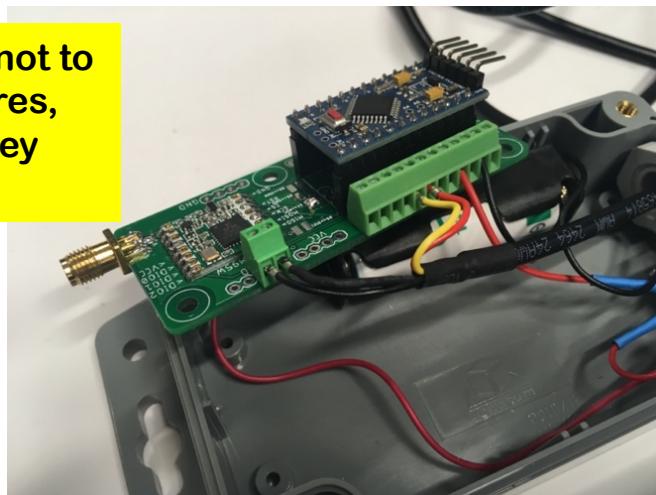
You can also use a 3.6V lithium battery with a 1-AA battery holder

This dedicated video will show how to build the outdoor soil device  
Video n°2: <https://youtu.be/zcazzDbXvHk>

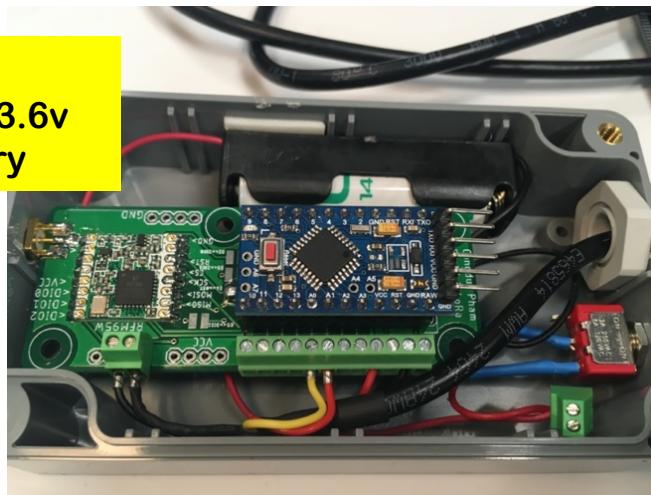


# The complete soil sensor device

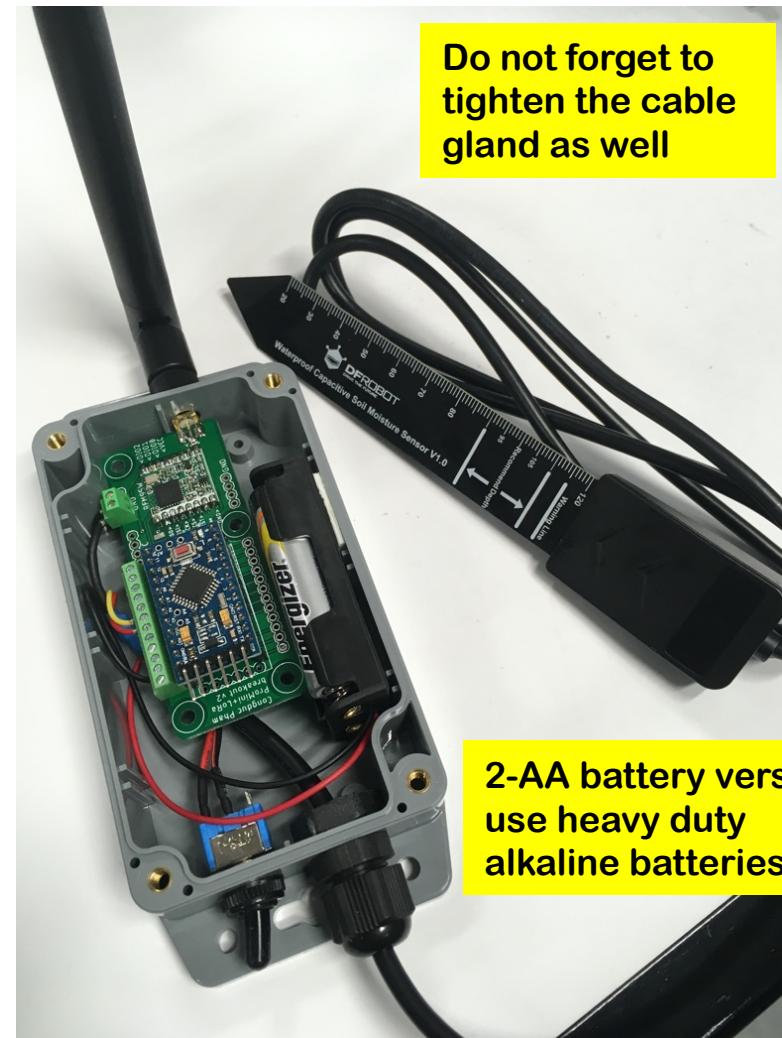
Tighten (but not too much!) all wires, check that they are not loose



1-AA battery version, use 3.6v lithium battery

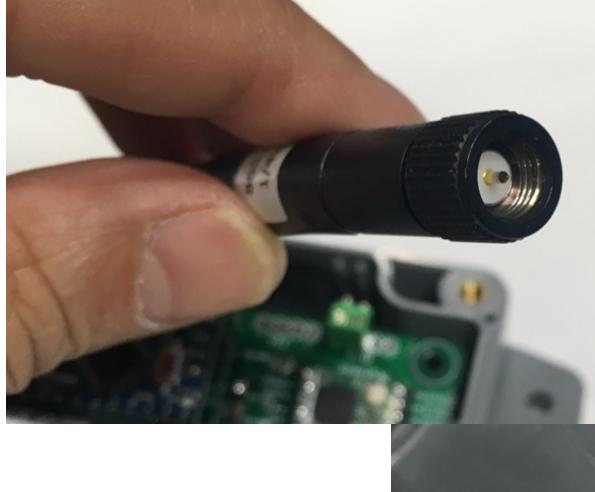


Do not forget to tighten the cable gland as well



2-AA battery version, use heavy duty alkaline batteries

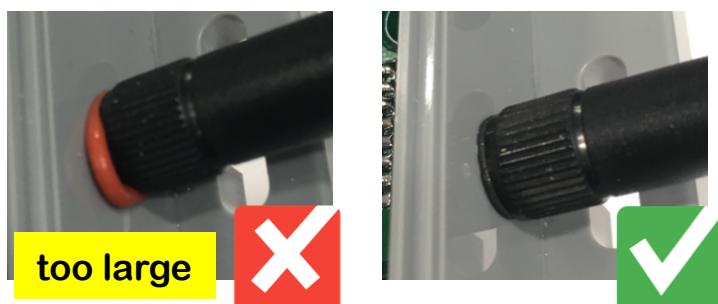
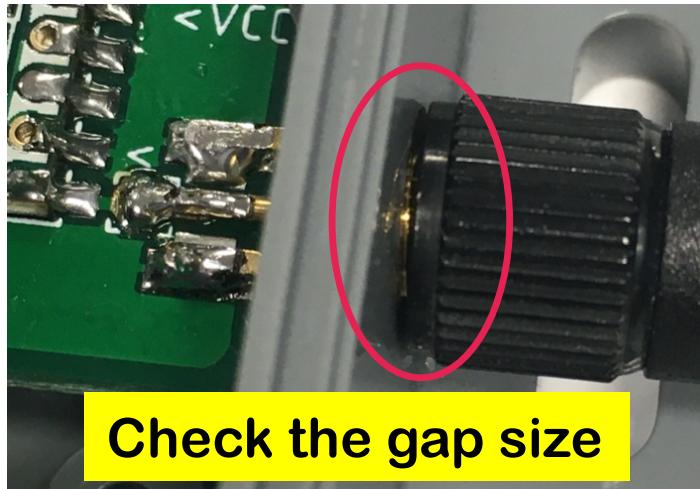
# Connecting antenna



Be sure to connect the matching antenna  
Here, SMA female with SMA male antenna  
Need to screw the antenna in all the way

The antenna junction is critical because this  
is where rain water can come in

# Waterproofing the antenna junction



Even when the antenna is screwed in all the way, there might still be a gap

Even with no apparent gap, it is necessary to waterproof the junction

Take an o-ring for that purpose, but do not take it too thick or too large!

Too thick: the antenna will not be screwed in all the way!

There are o-ring for SMA connector but the gap size is an indication

# Do not have or can not use o-ring?

Maybe the gap is too big? Use silicon joint sealant



Put small amount of silicon around the antenna junction (use a flat screw driver or other flat tool)



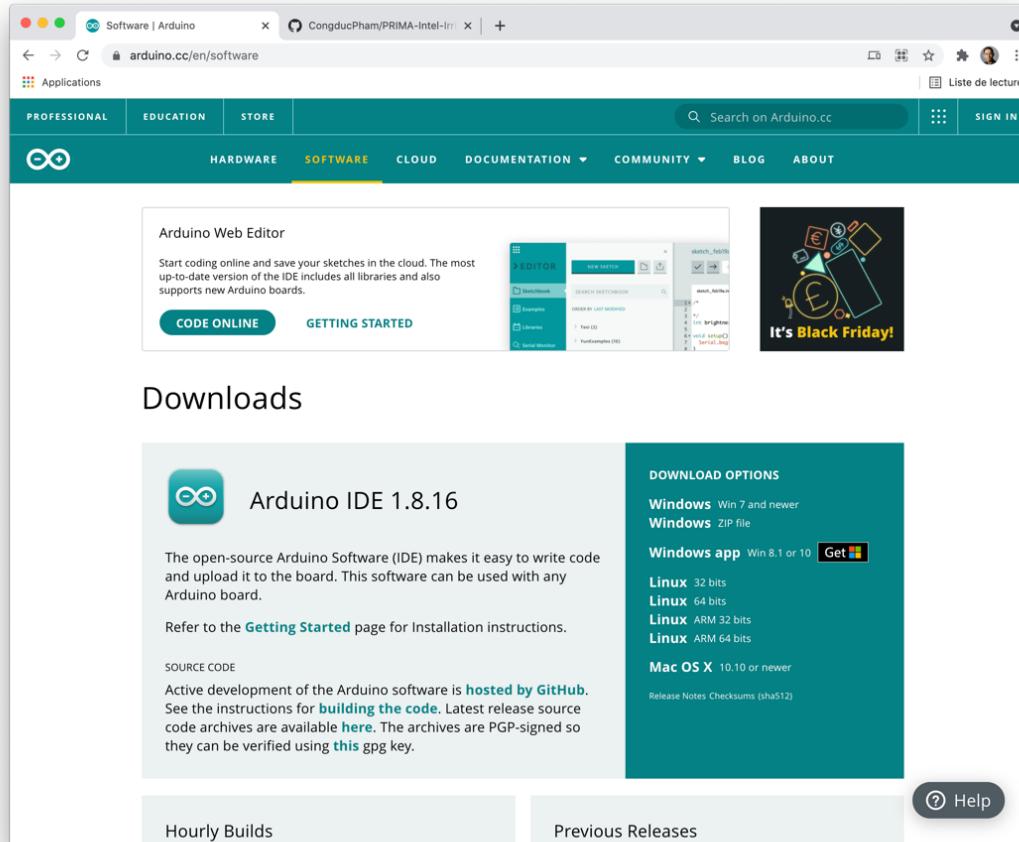
Use a wet toothpick to finish and clean the silicon all around the antenna junction



check especially  
the back side

**Even with o-ring, it is good to add silicon!**

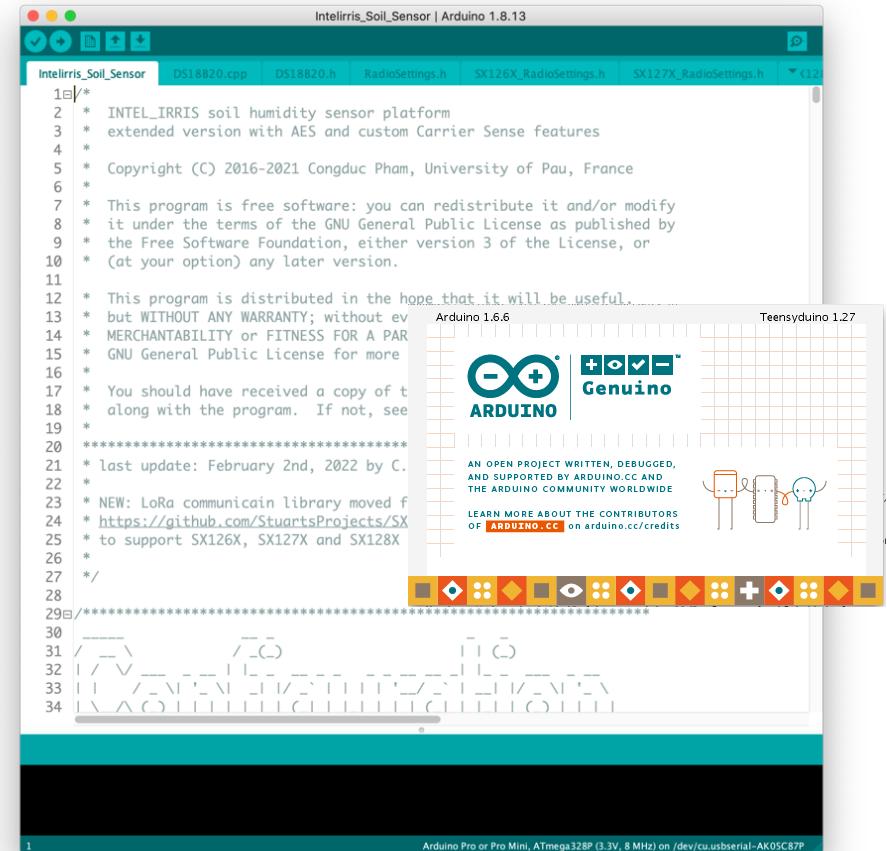
# Getting the software: Arduino IDE



The screenshot shows the Arduino website's software download section. It features the Arduino Web Editor on the left and download options for the Arduino IDE on the right. The download options include:

- Windows**: Win 7 and newer, ZIP file
- Windows app**: Win 8.1 or 10, Get
- Linux**: 32 bits, 64 bits, ARM 32 bits, ARM 64 bits
- Mac OS X**: 10.10 or newer

Below the download options, there are links for "SOURCE CODE" and "Release Notes Checksums (sha512)". At the bottom, there are links for "Hourly Builds" and "Previous Releases".



The screenshot shows the Arduino IDE interface with the sketch `Intelirris_Soil_Sensor` open. The code editor displays the following header comments:

```

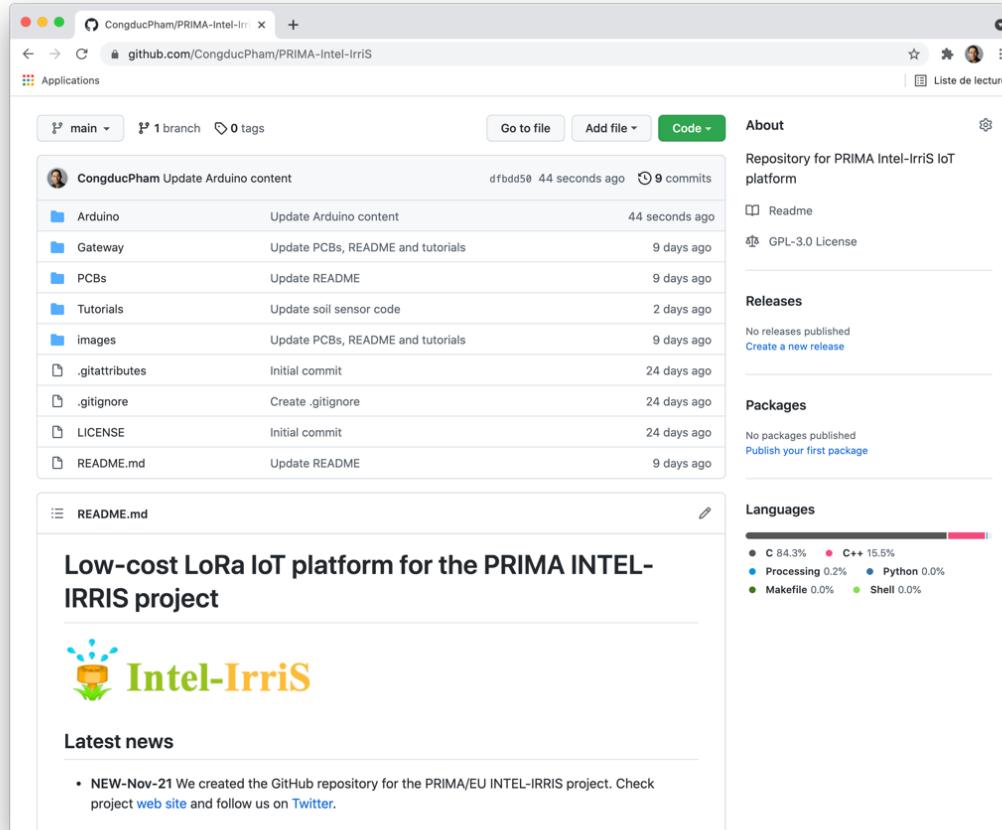
1 //*
2 * INTEL_IRRIS soil humidity sensor platform
3 * extended version with AES and custom Carrier Sense features
4 *
5 * Copyright (C) 2016-2021 Congduc Pham, University of Pau, France
6 *
7 * This program is free software: you can redistribute it and/or modify
8 * it under the terms of the GNU General Public License as published by
9 * the Free Software Foundation, either version 3 of the License, or
10 * (at your option) any later version.
11 *
12 * This program is distributed in the hope that it will be useful,
13 * but WITHOUT ANY WARRANTY; without even
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR
15 * GNU General Public License for more
16 *
17 * You should have received a copy of the
18 * along with the program. If not, see
19 *
20 ****
21 * last update: February 2nd, 2022 by C.
22 *
23 * NEW: LoRa communication library moved to
24 * https://github.com/StuartsProjects/SX
25 * to support SX126X, SX127X and SX128X
26 *
27 */
28 ****
29 */
30 */
31 */
32 */
33 */
34 */

```

The interface also shows the Arduino 1.6.6 and Teensyduino 1.2.7 boards selected at the top. The bottom status bar indicates "Arduino Pro or Pro Mini, ATmega328P (3.3V, 8 MHz) on /dev/cu.usbserial-AK05C87P".

Install latest version of Arduino IDE from  
<https://www.arduino.cc/en/software>

# Getting the software: Intel-IrriS code

The screenshot shows the GitHub repository page for CongducPham/PRIMA-Intel-IrriS. The repository has 1 branch and 0 tags. The main commit is by CongducPham, updating Arduino content. The repository is described as a "Repository for PRIMA Intel-IrriS IoT platform". It includes a Readme, a GPL-3.0 License, and no releases or packages published. The Languages section shows C at 84.3%, C++ at 15.5%, Processing at 0.2%, Python at 0.0%, Makefile at 0.0%, and Shell at 0.0%. The README.md file describes the Low-cost LoRa IoT platform for the PRIMA INTEL-IRRIS project.

On your computer, create a sketch folder

Then download the whole repository as ZIP file

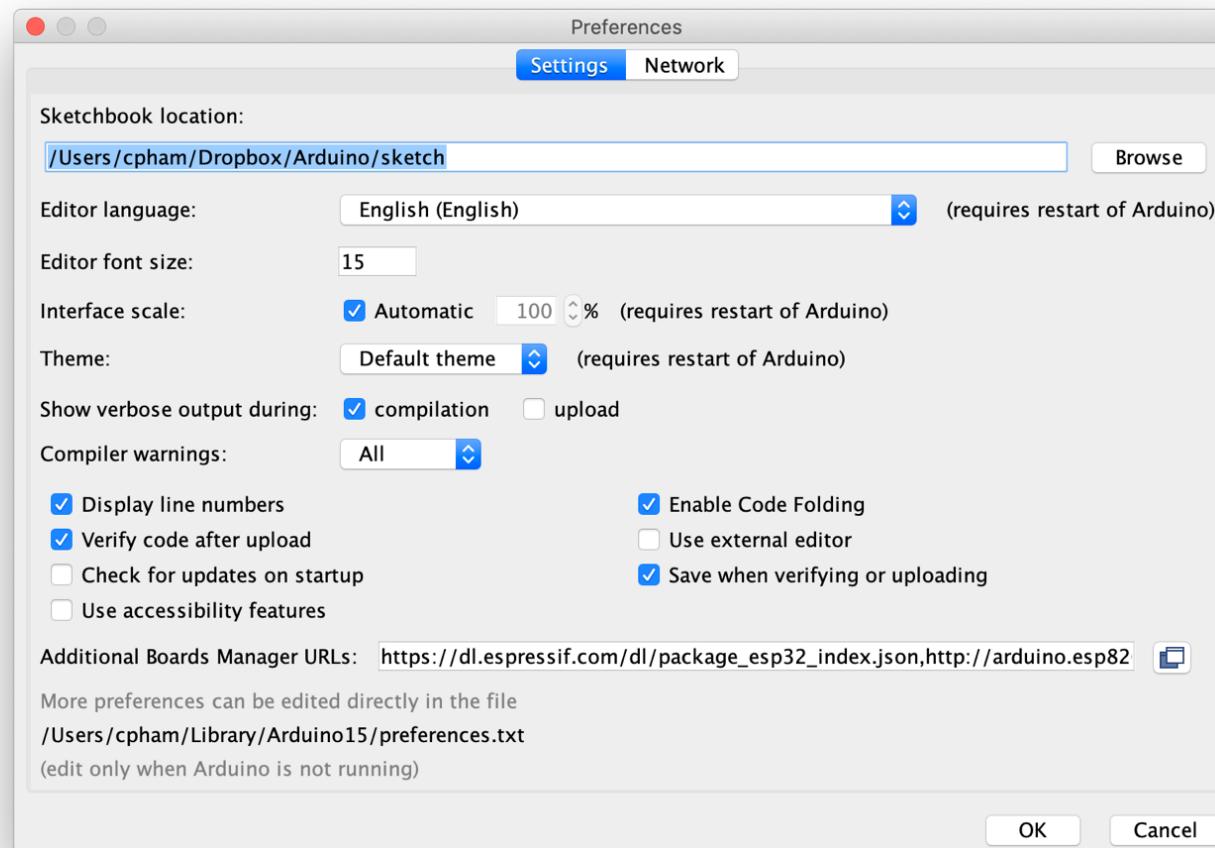
Unzip the file and copy the content of Arduino folder into your sketch folder

The entire Intel-IrriS GitHub repository is hosted here  
<https://github.com/CongducPham/PRIMA-Intel-IrriS>



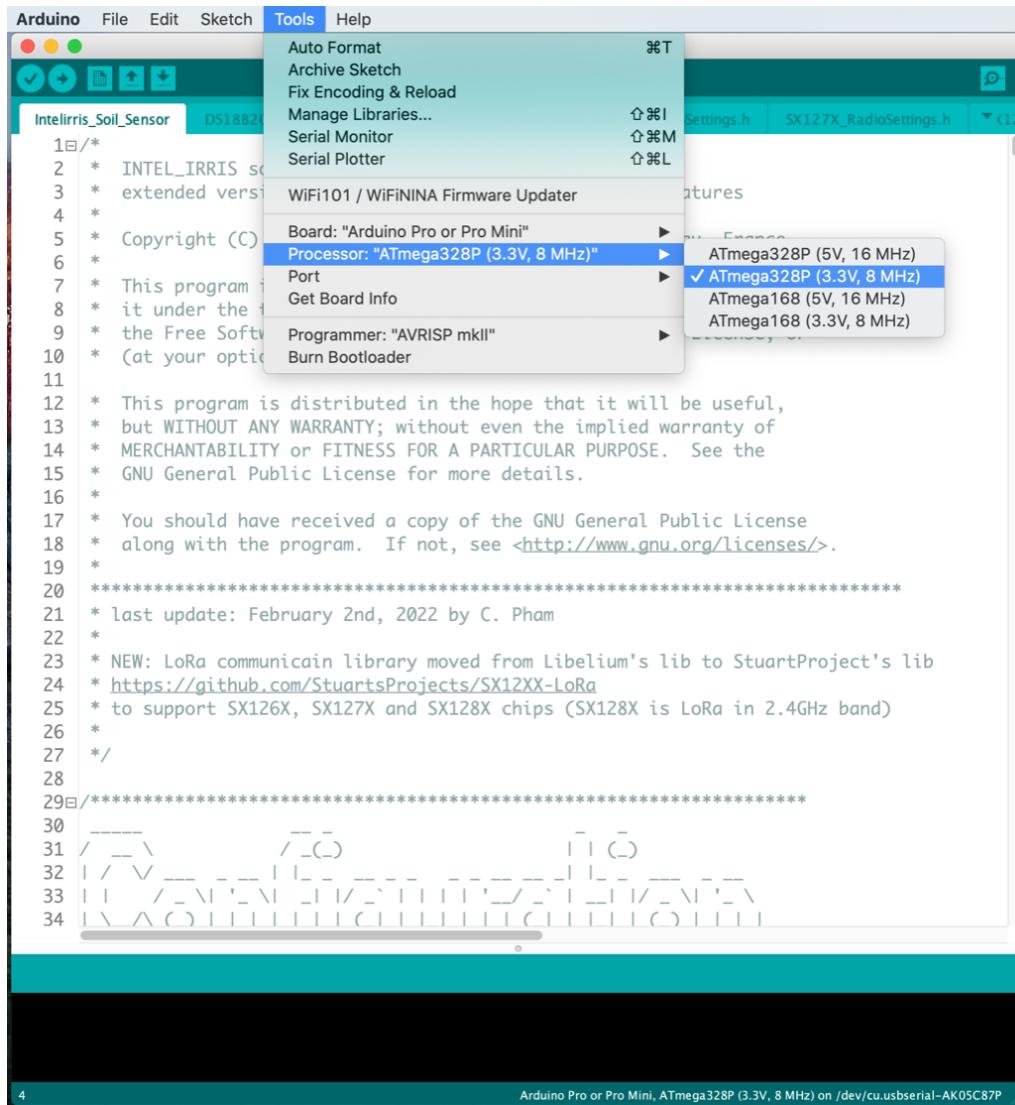
**GitHub**

# Setting your Arduino IDE



Run Arduino IDE, open Preferences  
 Indicate your sketch folder in Sketchbook location

# Compiling the soil sensor code



The screenshot shows the Arduino IDE interface. The sketch titled "Intelirris\_Soil\_Sensor" is open. The "Tools" menu is pulled down, showing various options like Auto Format, Archive Sketch, and Board selection. Under Board selection, it lists "Board: Arduino Pro or Pro Mini" and "Processor: ATmega328P (3.3V, 8 MHz)". A dropdown menu shows four options: ATmega328P (5V, 16 MHz), ATmega328P (3.3V, 8 MHz) (which is checked), ATmega168 (5V, 16 MHz), and ATmega168 (3.3V, 8 MHz). The main code area displays the source code for the soil sensor sketch.

```

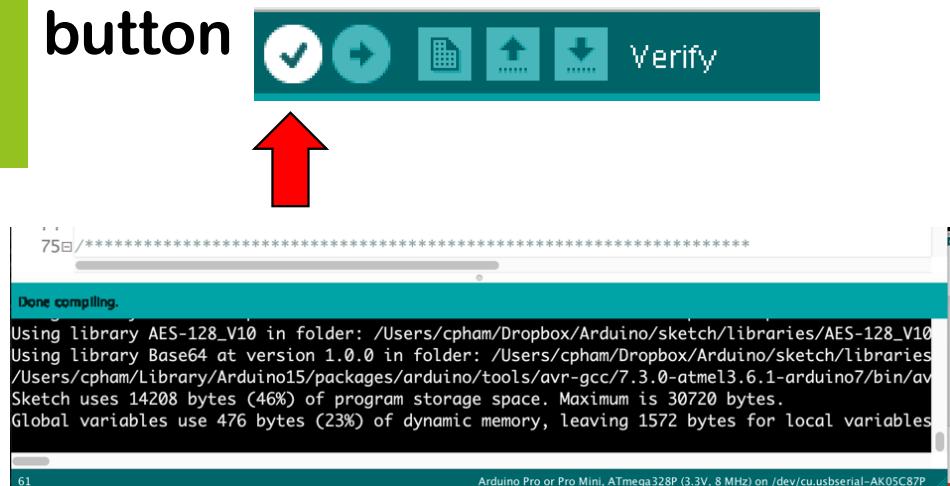
1 ///*
2 *  INTEL_IRRIS
3 *  extended version
4 *
5 *  Copyright (C)
6 *  This program is
7 *  it under the
8 *  the Free Software
9 *  (at your option)
10 *
11 *  This program is distributed in the hope that it will be useful,
12 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
13 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 *  GNU General Public License for more details.
15 *
16 *  You should have received a copy of the GNU General Public License
17 *  along with the program. If not, see <http://www.gnu.org/licenses/>.
18 *
19 ****
20 * last update: February 2nd, 2022 by C. Pham
21 *
22 * NEW: LoRa communicain library moved from Libelium's lib to StuartProject's lib
23 * https://github.com/StuartsProjects/SX12XX-LoRa
24 * to support SX126X, SX127X and SX128X chips (SX128X is LoRa in 2.4GHz band)
25 *
26 */
27 */
28 */
29 */
30 */
31 */
32 */
33 */
34 */

```

**Open Intelirris\_Soil\_Sensor sketch – no change required**

**Select the ProMini board,  
3.3V and 8MHz version**

**Then click on the "verify"  
button**

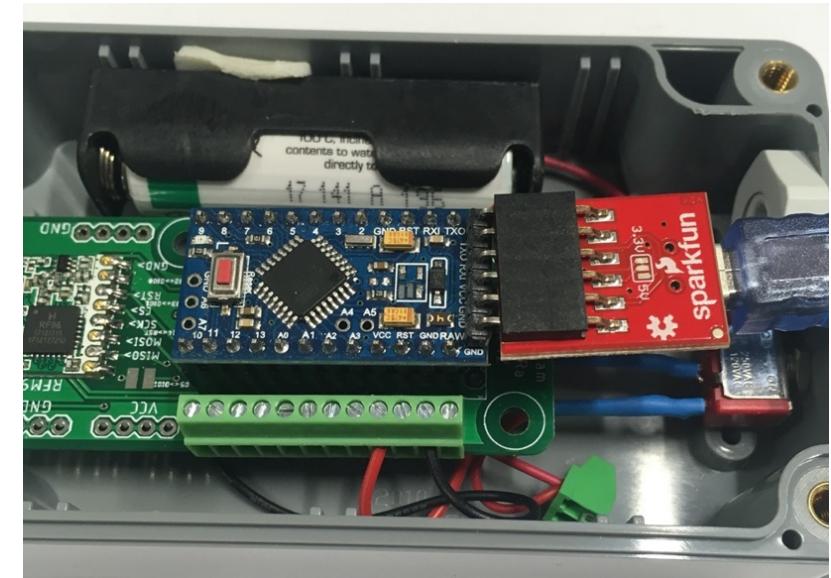
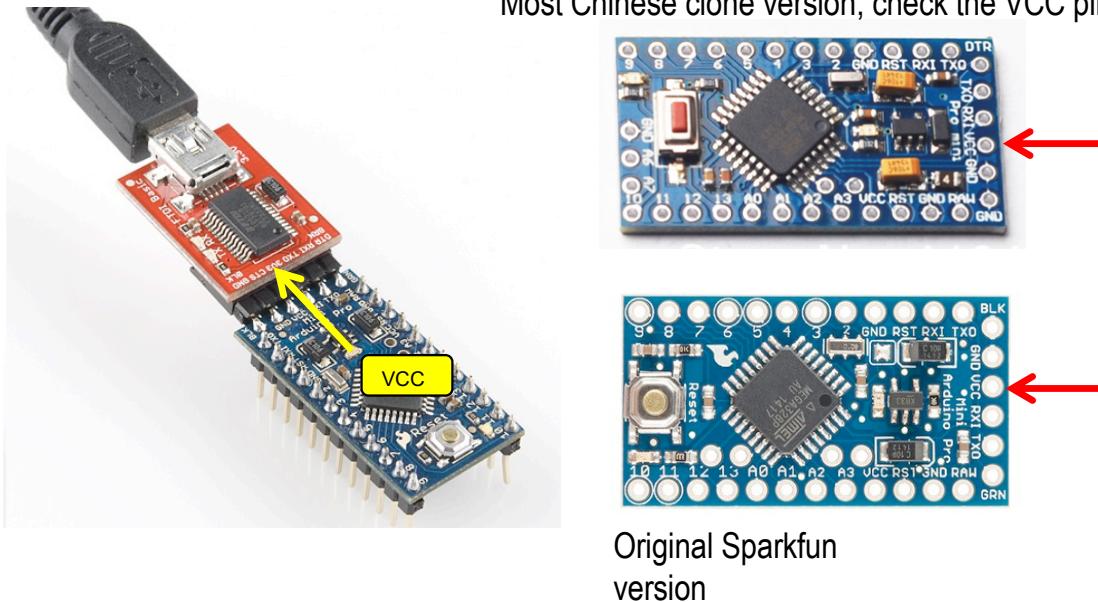




# Never transmit without antenna

- NEVER, NEVER transmit without an antenna
- Doing so can damage the radio module
- If your board is already connected to the radio module and you need to flash the board, connect the antenna
- If you need to update the existing code and your device already run a code that transmit data, connect the antenna
- It is safer when programming the device to remove the Arduino board from the female header and program it disconnected from the radio module
- If you deploy a device, make sure that the antenna is correctly connected before powering on the device and realizing any transmission test

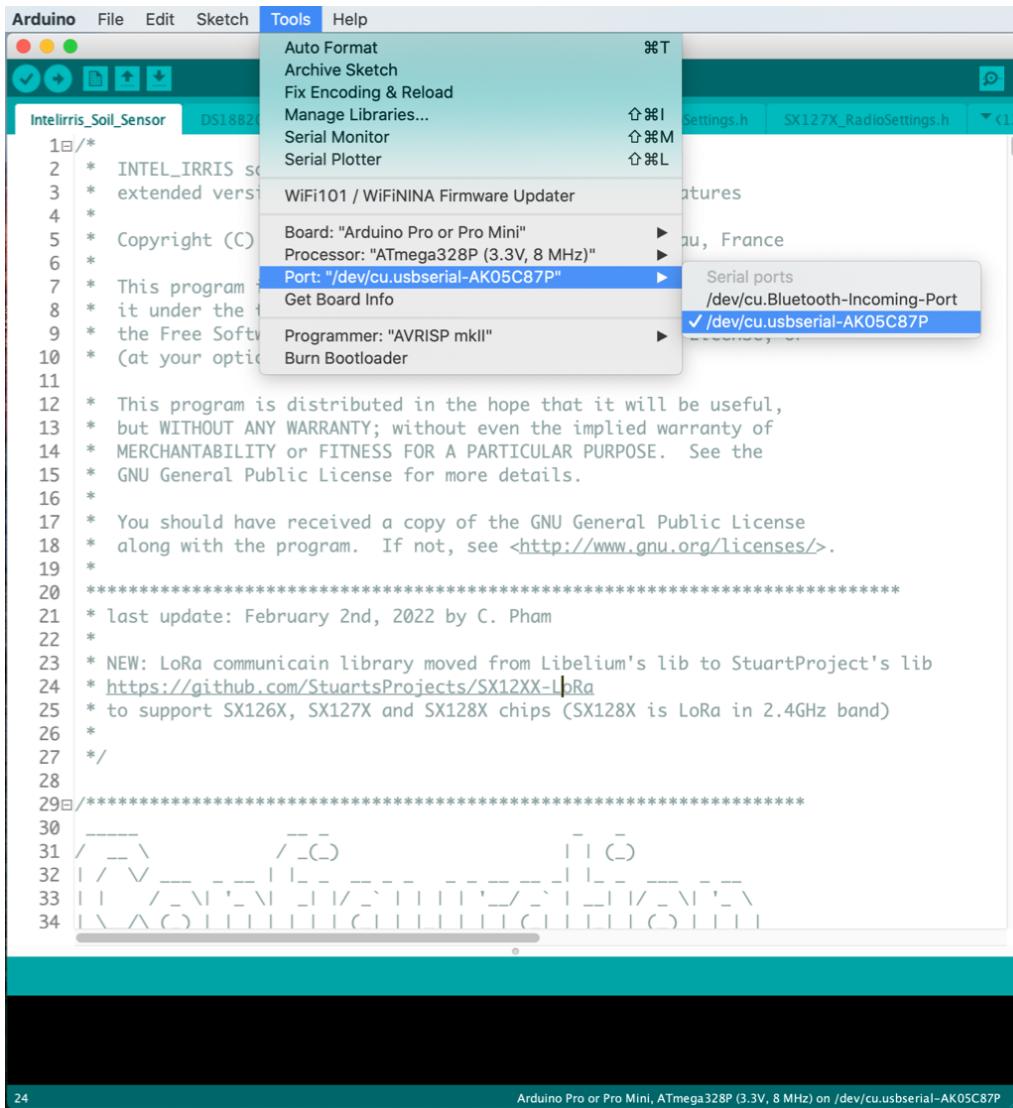
# Connecting with an FTDI cable



For the ProMini, you need to have an FTDI breakout cable working at 3.3v

Check the VCC pin position and make it to correspond to the VCC pin of the FTDI breakout.

# Select serial port for uploading

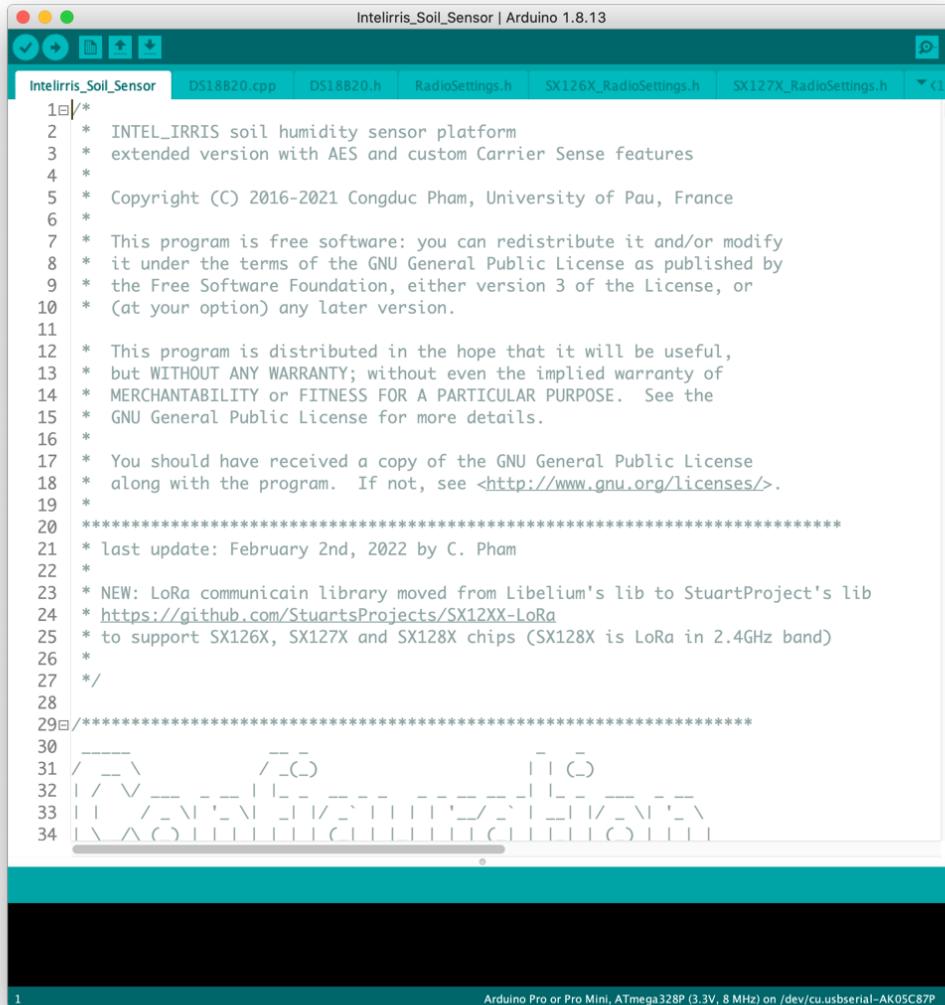


After connecting the cable to your computer/laptop USB port, try to find the serial port

If you don't find it, you may need to install specific drivers

<https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all>

# Uploading to your board



```

1/* 
2 * INTEL_IRRIS soil humidity sensor platform
3 * extended version with AES and custom Carrier Sense features
4 *
5 * Copyright (C) 2016-2021 Congduc Pham, University of Pau, France
6 *
7 * This program is free software: you can redistribute it and/or modify
8 * it under the terms of the GNU General Public License as published by
9 * the Free Software Foundation, either version 3 of the License, or
10 * (at your option) any later version.
11 *
12 * This program is distributed in the hope that it will be useful,
13 * but WITHOUT ANY WARRANTY; without even the implied warranty of
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15 * GNU General Public License for more details.
16 *
17 * You should have received a copy of the GNU General Public License
18 * along with the program. If not, see <http://www.gnu.org/licenses/>.
19 *
20 ****
21 * last update: February 2nd, 2022 by C. Pham
22 *
23 * NEW: LoRa communicain library moved from Libelium's lib to StuartProject's lib
24 * https://github.com/StuartsProjects/SX12XX-LoRa
25 * to support SX126X, SX127X and SX128X chips (SX128X is LoRa in 2.4GHz band)
26 *
27 */
28
29 ****
30 -----
31 / \ / \ -----
32 | \ \ | \ | \ | \ |
33 | \ | \ | \ | \ | \ |
34 | \ | \ | \ | \ | \ |

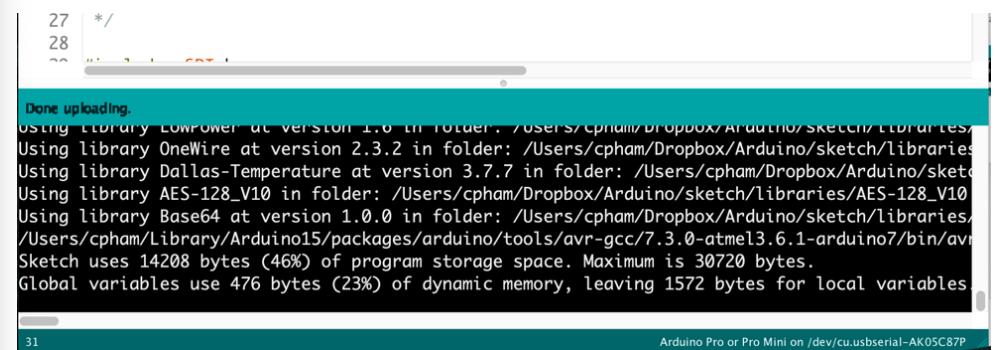
```

Arduino Pro or Pro Mini, ATmega328P (3.3V, 8 MHz) on /dev/cu.usbserial-AK05C87P

**Click on the "upload" button**



**And wait until upload is completed**



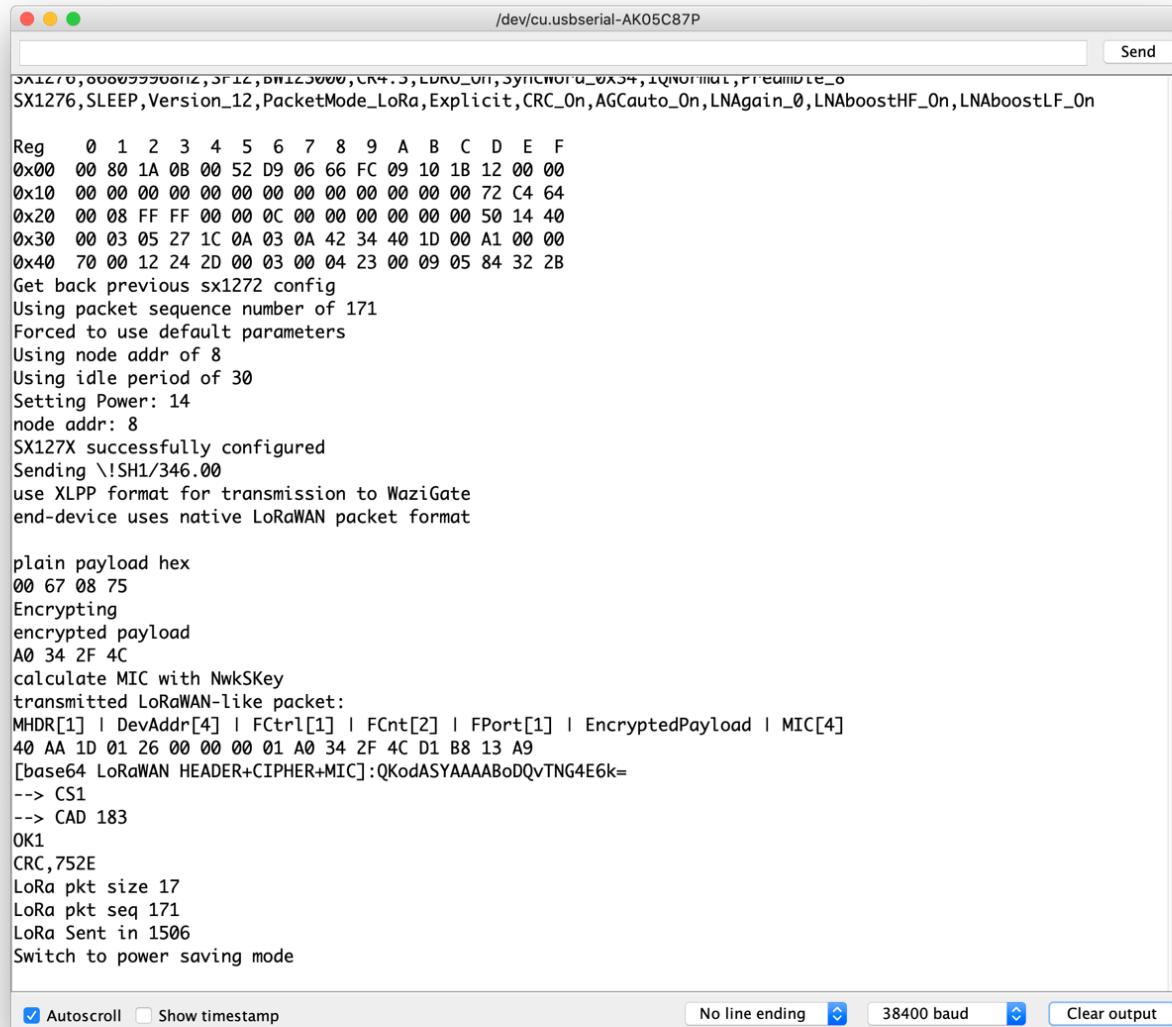
```

27 */
28
29 Done uploading.
Using library LowPower at version 1.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
Using library OneWire at version 2.3.2 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
Using library Dallas-Temperature at version 3.7.7 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
Using library AES-128_V10 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/AES-128_V10
Using library Base64 at version 1.0.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/Base64
Sketch uses 14208 bytes (46%) of program storage space. Maximum is 30720 bytes.
Global variables use 476 bytes (23%) of dynamic memory, leaving 1572 bytes for local variables

```

Arduino Pro or Pro Mini on /dev/cu.usbserial-AK05C87P

# Checking that device is operational



```

/dev/cu.usbserial-AK05C87P
Send

SX1276,0000000000000000,3F12,B0123000,CRC4.5,LDRU_001,SYNCHRO_0X34,TQ_NORM,PREDECODE_0
SX1276,SLEEP,Version_12,PacketMode_LoRa,Explicit,CRC_On,AGCAuto_On,LNAgain_0,LNAboostHF_On,LNAboostLF_On

Reg 0 1 2 3 4 5 6 7 8 9 A B C D E F
0x00 00 80 1A 0B 00 52 D9 06 66 FC 09 10 1B 12 00 00
0x10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 72 C4 64
0x20 00 08 FF FF 00 00 0C 00 00 00 00 00 00 00 50 14 40
0x30 00 03 05 27 1C 0A 03 0A 42 34 40 1D 00 A1 00 00
0x40 70 00 12 24 2D 00 03 00 04 23 00 09 05 84 32 2B
Get back previous sx1272 config
Using packet sequence number of 171
Forced to use default parameters
Using node addr of 8
Using idle period of 30
Setting Power: 14
node addr: 8
SX127X successfully configured
Sending \!SH1/346.00
use XLPP format for transmission to WaziGate
end-device uses native LoRaWAN packet format

plain payload hex
00 67 08 75
Encrypting
encrypted payload
A0 34 2F 4C
calculate MIC with NwkSKey
transmitted LoRaWAN-like packet:
MHDR[1] | DevAddr[4] | FCtrl[1] | FCnt[2] | FPort[1] | EncryptedPayload | MIC[4]
40 AA 1D 01 26 00 00 00 01 A0 34 2F 4C D1 B8 13 A9
[base64 LoRaWAN HEADER+CIPHER+MIC]:QKodASYAAABoDQvTNG4E6k=
--> CS1
--> CAD 183
OK1
CRC,752E
LoRa pkt size 17
LoRa pkt seq 171
LoRa Sent in 1506
Switch to power saving mode

 Autoscroll  Show timestamp
  No line ending  Clear output

```

**Open serial monitor**

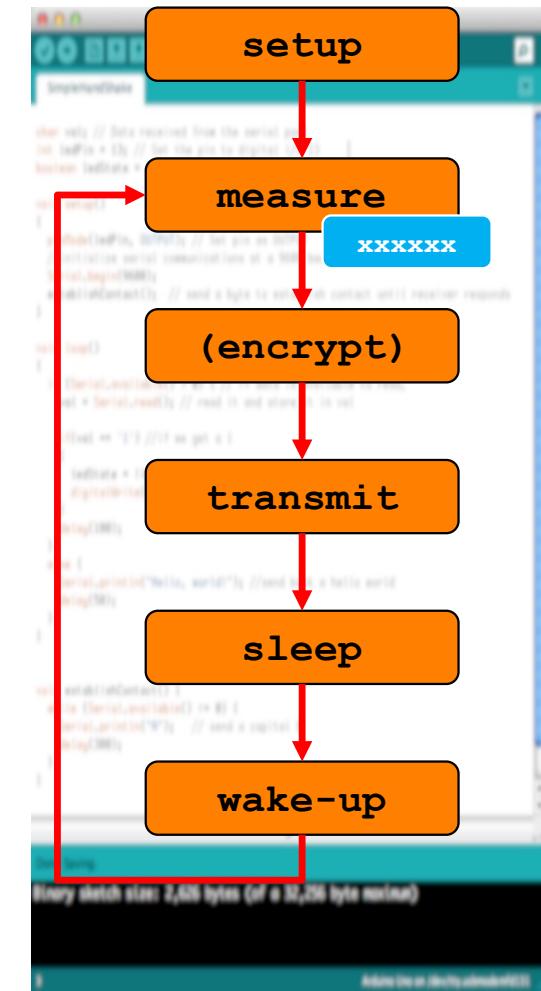
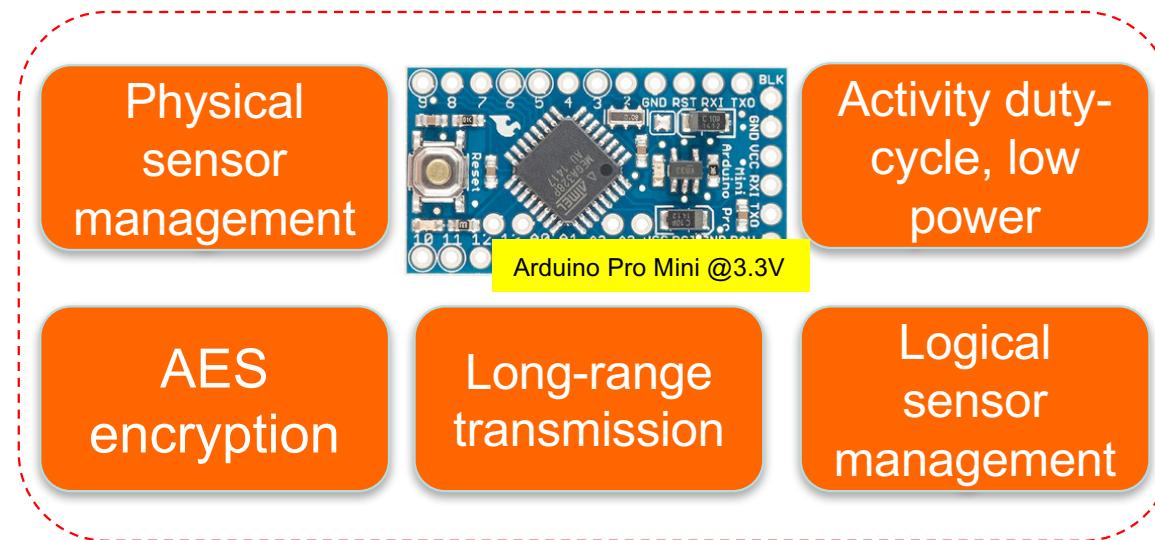
**Set baud rate to 38400**

**See output from board**

**Check that transmission is OK**

# Generic & cyclic behavior

SEN0308  
 capacitive soil humidity  
 A0 (signal), A1 (power)



# Transmission to WaziGate



Parameters for  
INTEL-IRRIS WaziGate



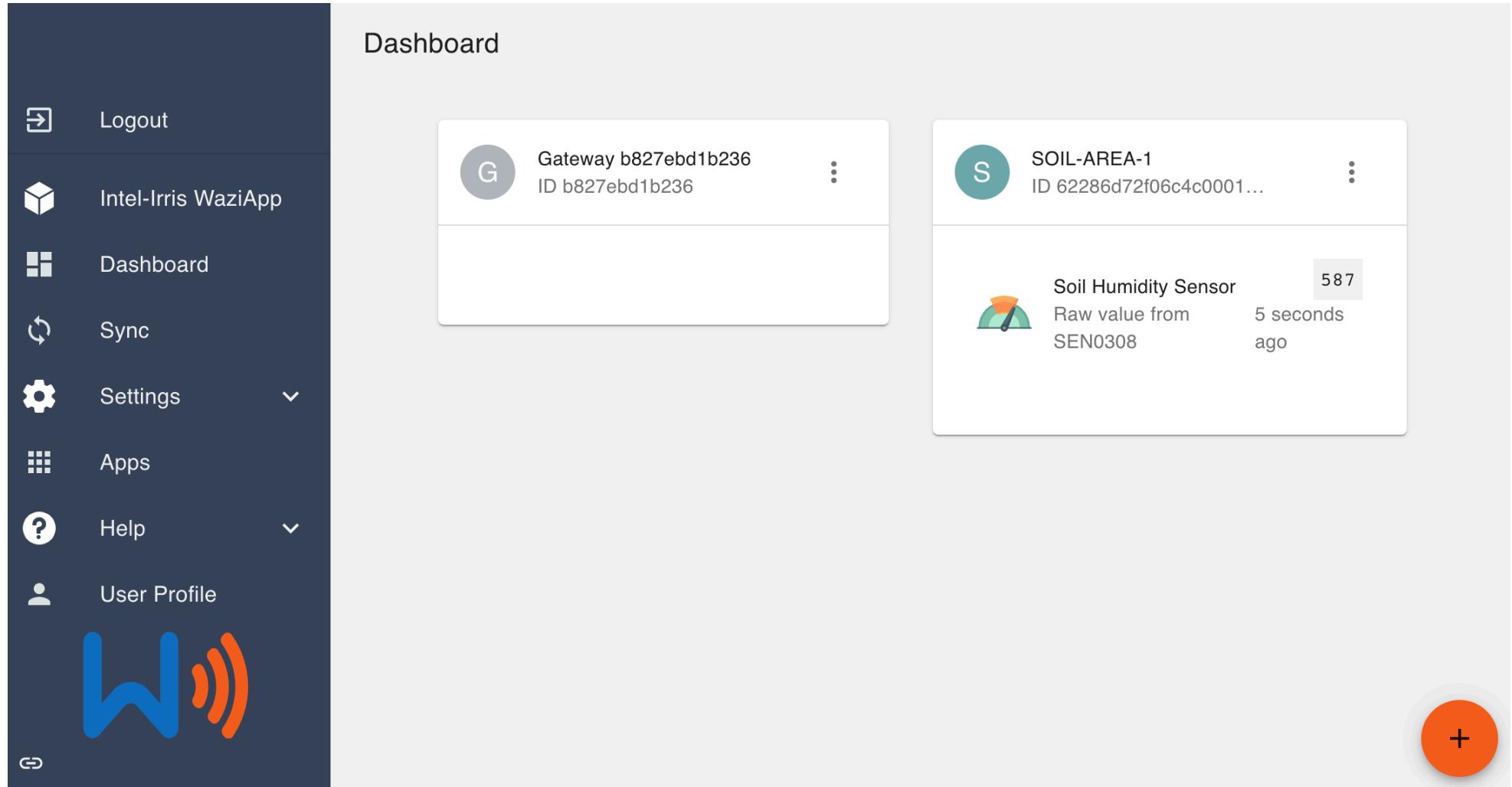
SF12BW125  
868.1MHz | 433.175MHz  
Node id is 26011DAA  
1 msg/60mins  
1 sensor  
XLPP data



This dedicated video will show all these steps, from connecting the SEN0308 to testing transmission to the WaziGate  
Video n°4: <https://youtu.be/j-1Nk0tv0xM>



# See it on the dashboard



The screenshot shows the Intel-Irris WaziApp dashboard. On the left is a dark sidebar with white icons and text:

- Logout
- Intel-Irris WaziApp
- Dashboard
- Sync
- Settings
- Apps
- Help
- User Profile

At the bottom of the sidebar is a blue icon with a blue 'W' and orange 'Wi-Fi' signals.

The main area is titled "Dashboard". It displays two cards:

- Gateway b827ebd1b236**  
ID b827ebd1b236
- SOIL-AREA-1**  
ID 62286d72f06c4c0001...

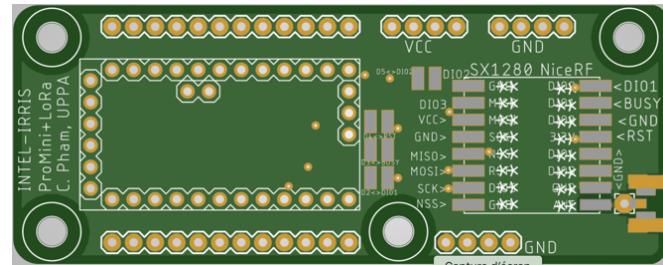
Below these cards is a detailed view of the Soil Humidity Sensor card:

- Soil Humidity Sensor**
- Raw value from SEN0308
- 587 (value)
- 5 seconds ago

A large orange button with a white plus sign is located at the bottom right of the dashboard area.

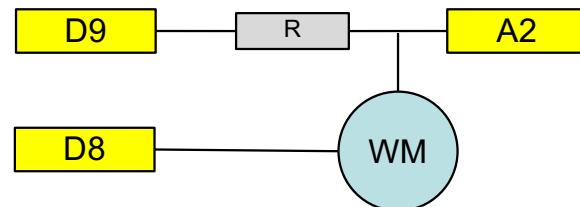
# Advanced options

- The Intelirris\_Soil\_Sensor sketch can
  - Drive SX127X, SX126X and SX128X LoRa chips
  - SX128X requires a dedicated PCB for NiceRF SX1280



- Send AES 128-bit encrypted packet
- Send uplink LoRaWAN packet (encrypted)
- Receive downlink LoRaWAN packet
- Can read from a Watermark water tension sensor
- Can read additionally from a DS18B20 soil temperature sensor
- Support WaziSense and WaziDev boards specific features

# With a Watermark sensor



with a Watermark sensor, the "pseudo-AC Short Pulse" method will be used – see <https://www.irrometer.com/200ss.html>

D9 and D8 will be used to alternating power the sensor

A2 will be used to read signal from sensor

R is a resistor from 7kOhms to 14kOhms

# Software configuration for WM

- Default resistor value is 10kOhms
- Change in watermark.h file the WM\_RESISTOR value to match the one you are using

```
//put here the resistor value, in Ohms
#define WM_RESISTOR 10000
//we defined WM_MAX_RESISTOR=32760 because the transmitted value would be 32760/10=3276
//and currently a bug in WaziGate XLPP decoding code will limit the maximum value to 3276
#define WM_MAX_RESISTOR 32760
```

- Uncomment in main program WITH\_WATERMARK

```
///////////
// uncomment to have a soil tensiometer watermark sensor
#define WITH_WATERMARK
#define WM_REF_TEMPERATURE 28.0
///////////
// uncomment to force the watermark to have default device address for WaziGate
//#define WM_AS_PRIMARY_SENSOR
```

# Transmission with the WM sensor

- Devices with a WM sensor will have a different default address
  - 26011DB1 instead of 26011DAA for capacitive
- 2 values are transmitted
  - Centibars converted from resistance value
  - Raw resistance value measured for WM sensor, but scaled down by a factor of 10 → 300 = 3000 Ohms



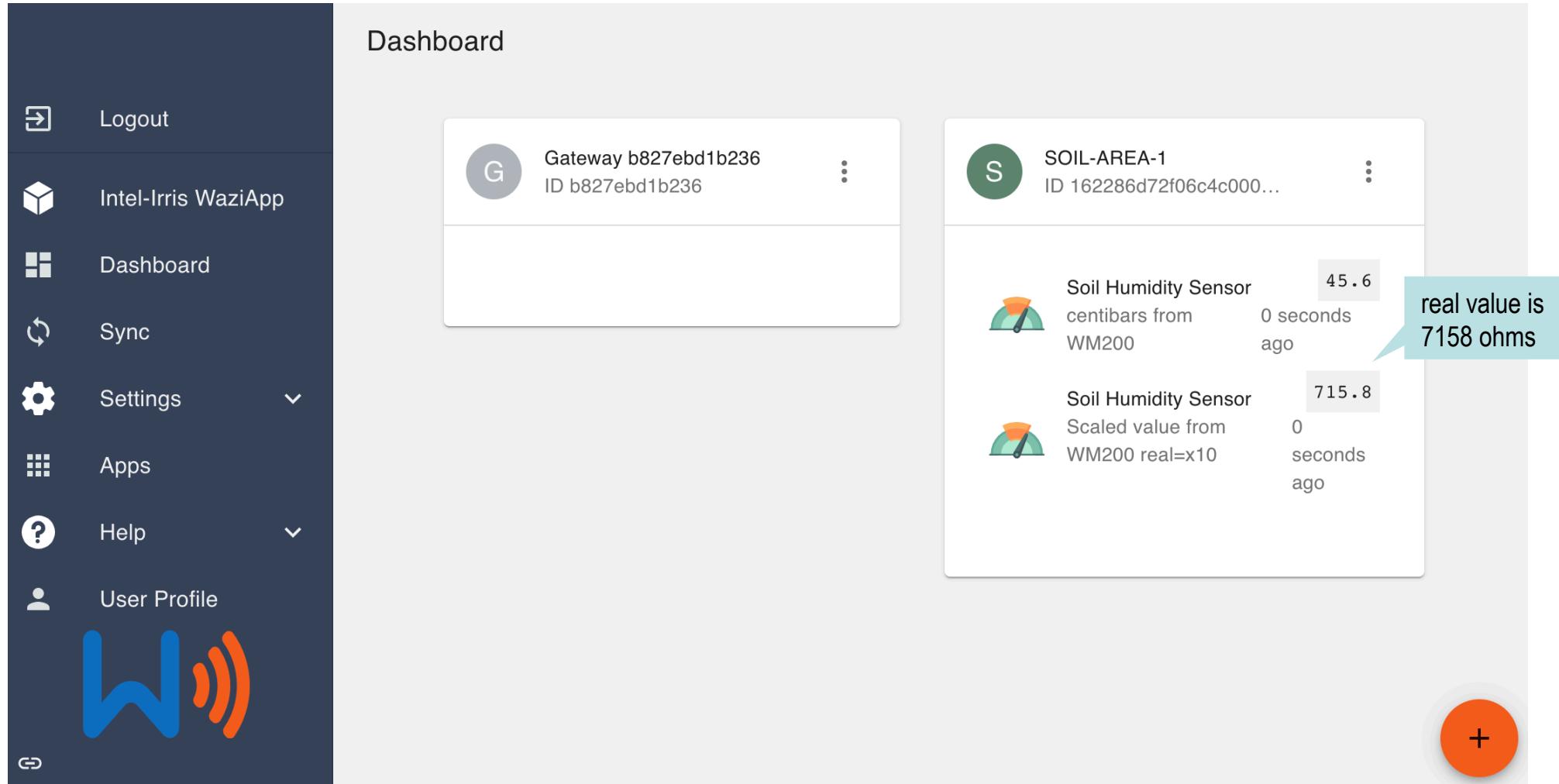
Parameters for  
INTEL-IRRIS WaziGate

**LoRaWAN™**

SF12BW125  
 868.1MHz | 433.175MHz  
 Node id is 26011**DB1**  
 1 msg/60mins  
 1 sensor  
 XLPP data



# See it on the dashboard



The screenshot shows the Intel-Irris WaziApp dashboard. On the left, a sidebar menu includes Logout, Intel-Irris WaziApp, Dashboard, Sync, Settings, Apps, Help, and User Profile. The main area is titled "Dashboard". It displays two main components: a "Gateway" card and a "SOIL-AREA-1" card.

**Gateway b827ebd1b236**  
ID b827ebd1b236

**SOIL-AREA-1**  
ID 162286d72f06c4c000...

**Soil Humidity Sensor**  
centibars from WM200      45.6  
0 seconds ago

**Soil Humidity Sensor**  
Scaled value from WM200 real=x10      715.8  
0 seconds ago

A blue callout bubble points to the second soil humidity reading with the text "real value is 7158 ohms".

# 2 versions of the soil device



# Soil temperature with WM sensor (1)

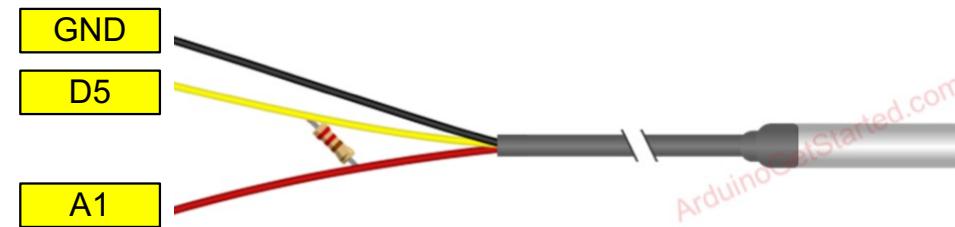
- Without a real temperature sensor, the Arduino code uses a default soil temperature set to 28°C to convert the resistance value into centibars (see <https://www.irrometer.com/200ss.html>)

```
//////////  
// uncomment to have a soil tensiometer watermark sensor  
#define WITH_WATERMARK  
#define WM_REF_TEMPERATURE 28.0  
//////////  
// uncomment to force the watermark to have default device address for WaziGate  
//#define WM_AS_PRIMARY_SENSOR
```

- You can change this setting in the code for testing purposes
- The final objective is to use a real temperature sensor to dynamically get the soil temperature

# Soil temperature with WM sensor (2)

- A DS18B20 temperature sensor can be connected to the device. A 4.7kOhms must be added as illustrated, connecting D5 and A1



```
//////////  

// uncomment to have 1 soil temperature sensor ST  

// using a one-wire DS18B20 sensor  

#define SOIL_TEMP_SENSOR  

// #define LINK_SOIL_TEMP_TO_CENTIBAR
```

- In this case, the soil temperature will also be sent to the gateway
- It is also possible to link the measured soil temperature to the centibar calculation performed by the device itself

# See it on the dashboard

- [!\[\]\(5a549831d91e212ae82682f6fafea2bf\_img.jpg\) Logout](#)
- [!\[\]\(1d88efe35d4cf0bb1e21b20c20f4f74a\_img.jpg\) Intel-Irris WaziApp](#)
- [!\[\]\(647ca7092c9d12c8a651457390adb59d\_img.jpg\) Dashboard](#)
- [!\[\]\(06db3d1411c33d5691f1491a53bdd106\_img.jpg\) Sync](#)
- [!\[\]\(995848000d32e60a1cd9531d9c1f0292\_img.jpg\) Settings](#)
- [!\[\]\(404445586bd5f995b01dfb1dd19d20e8\_img.jpg\) Apps](#)
- [!\[\]\(203bf135f25cc5fda98e06baa6453d87\_img.jpg\) Help](#)
- [!\[\]\(703773584c3927a1c265c35782e6612c\_img.jpg\) User Profile](#)



Dashboard

G
Gateway b827ebd1b236
ID b827ebd1b236
⋮

S
SOIL-AREA-1
ID 162286d72f06c4c000...
⋮


Soil Humidity Sensor
45 . 6

centibars from WM200 6 seconds ago


Soil Humidity Sensor
714 . 5

Scaled value from WM200 real=x10 6 seconds ago


Soil Temperature Sensor
26 . 3

degree Celsius 6 seconds ago

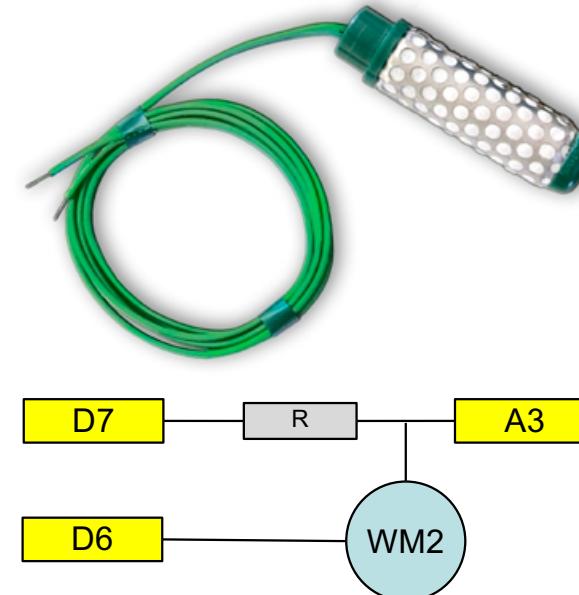
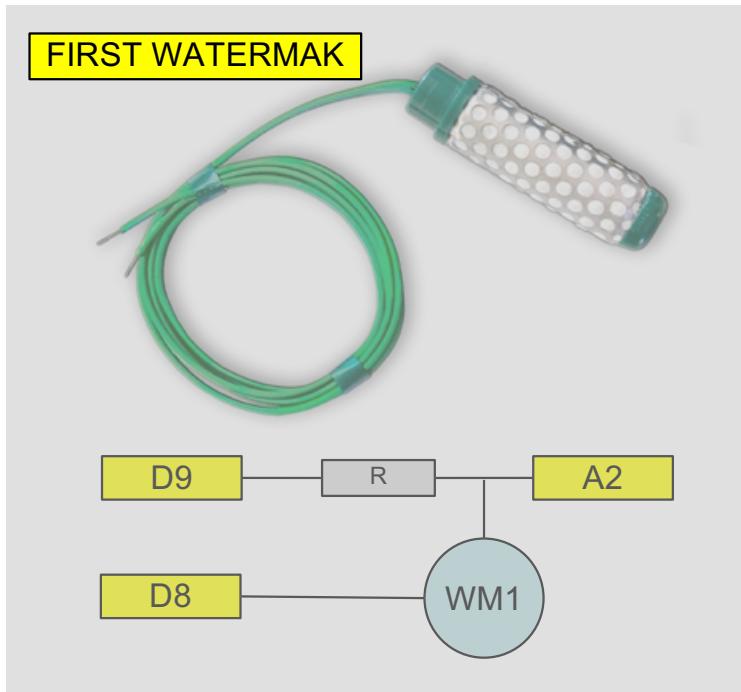
Here, real soil temperature is 26.3°C but the device is not linking the centibar conversion with the temperature sensor and uses the default 28°C value. The final centibar conversion could be realized by a dedicated irrigation application run on the WaziGate gateway.

+

Prof. Congduc Pham  
<http://www.univ-pau.fr/~cpham>

54

# Add a second Watermark sensors



a second Watermark sensor can be connected: the purpose is to determine the water movement in the soil  
 D7 and D6 will be used to alternating power the sensor  
 A3 will be used to read signal from sensor  
 R should be identical for both Watermark

# Software configuration for 2 WM

- Uncomment in main program TWO\_WATERMARK

```
//////////  
// uncomment to have a soil tensiometer watermark sensor  
#define WITH_WATERMARK  
#define WM_REF_TEMPERATURE 28.0  
  
//////////  
// uncomment to force the watermark to have default device address for WaziGate  
//#define WM_AS_PRIMARY_SENSOR  
  
//////////  
// uncomment to have 2 tensiometer watermark sensor on the same device  
#define TWO_WATERMARK
```

- You can still add a soil temperature sensor

# See it on the dashboard

Logout

Intel-Irris WaziApp

Dashboard

Sync

Settings

Apps

Help

User Profile

W

Dashboard

**G** Gateway b827ebd1b236  
ID b827ebd1b236

**S** SOIL-AREA-1  
ID 162286d72f06c4c000...

Here, the device has been compiled to link the centibar conversion with the real value measured by the temperature sensor

If the default 28°C value was used, centibar for WM1 with 1374 ohms would have been 13.1 and centibar for WM2 with 685 ohms would have been 2.0

Watermark 1  
centibars from WM200

Watermark 1  
Scaled value from WM200 real=x10

Watermark 2  
centibars from WM200

Watermark 2  
Scaled value from WM200 real=x10

Soil Temperature Sensor  
degree Celsius

12.3  
4 seconds ago

137.4  
4 seconds ago

2.1  
4 seconds ago

68.5  
4 seconds ago

24.9  
4 seconds ago



# Experimental INTEL-IRRIS soil device

- 2 Watermark sensors and 1 soil temperature sensor





Starter-kit

Autonomous

Intelligent Irrigation

Plug-&-Sense

In-the-box