MkrfoxWindShield manual

pascal.caunegre@gmail.com, Mai 4, 2022

Building your own wind sensor station

This manual shows how to build an OpenWindMap-compatible wind station.
Refer to https://www.openwindmap.org for more information on this network (previously known as *Pioupiou*).

You will need to gather the following parts:

- One of the three possible sensors:

Davis

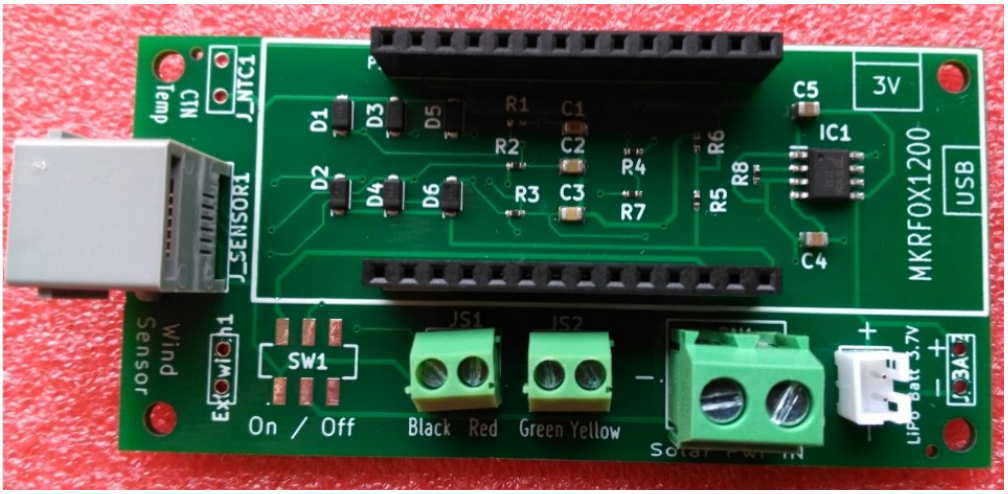
OR

Peet Bros

OR

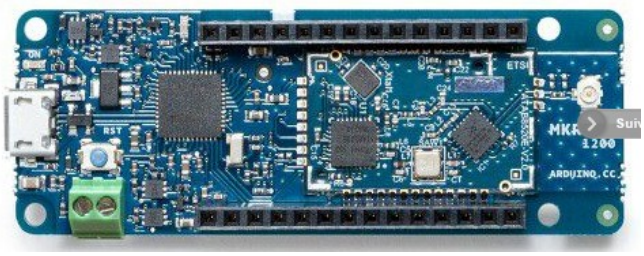
Mi.Sol
(Shenzen)

- The MkrfoxWindShield (version 1.1 on the picture)



(the author can resell some boards if available, otherwise you can fabricate them with schematics and part list given in hardware/ directory of this project)

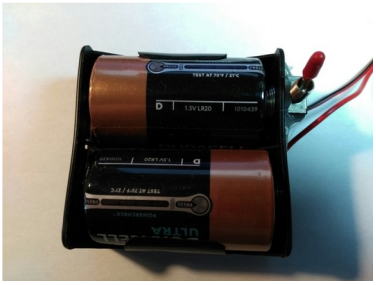
- A MkrFox1200 Arduino board



- A 868 Mhz Antenna

- Either batteries or solar panel

OR



A Solar Panel (5V, 0.5W) + a LiPo 3.7V battery

A set of two 1.5V Batteries

(note: 1.5V batteries will last about a year while solar panel solution will not need maintenance)

Uploading the station software

Your MkrFox1200 needs a software to operate the wind sensor. But first you need to read internal board IDs with a specific program. Process as follows:

- If necessary, install the Arduino software (IDE) on your computer.
- Download all code files from the github: https://github.com/pcaunegre/MkrfoxWindShield/tree/main/src/firmware
- Connect the board to your computer by a USB cable

Board ID reading:

- Open the src/board_infos/board_infos.ino file from the IDE
- Double click on the board reset button, the yellow led is dimmed and blinks, meaning the board is ready to receive the program
- Click the upload button from the IDE
- Open the Serial monitor
- The ID and PAC values will be printed. Note them down.

Registration of the board through OpenWindMap network

Send the board IDs you have noted down to OpenWindMap through the forum.
You will get a station ID in return.

Your measurements will be available through http://www.openwindmap.org/<ID>

Note: it is important to register the board via OWM (and not directly via Sigfox server) because it allows benefiting the group contract and shares your measurements with the community.

Final software installation:

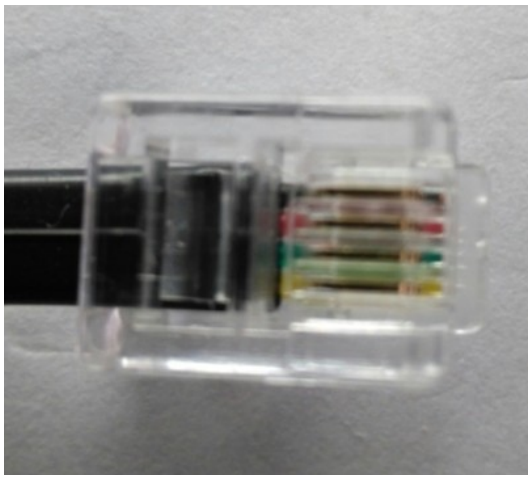
Important: Don't do this before your station is registered in OpenWindMap because this step will trigger a Sigfox message dispatch. **(Alternately, you can put a jumper between the pin GND and pin 14 to disable the Sigfox messages.)**

- Open the src/firmware/firmware.ino file from the IDE
- Double click on the board reset button, the yellow led is dimmed and blinks, meaning the board is ready to receive the program
- Click the upload button from the IDE
- The led should then blink frankly 10 times and then switch off.
- Your board is ready, disconnect it from USB.

Assembling parts

Assemble the various parts and install them into a rainproof box. To assemble you will have to:

- Plug the MkrFox1200 on top of the wind shield.
- Connect the antenna through the coaxial RF socket.
- If you use 1.5V batteries, connect them through the MkrFox1200 screw connector (take care of the polarity). Preferably use solar panel+Lipo battery, by connecting them on the wind shield (solar panel on screw connector, battery on the white socket).
- Connect your wind sensor either by plugging the RJ11 jack or by connecting the wires by the screw connector (from left to right):
  - for Davis and Peet sensors: connect 1: **black** wire**, 2: red, 3: green, 4: yellow**
  - for MiSol: connect 1: **black** wire**, 2: red, 3: yellow, 4: green**



Notes about v1.1 of the board

- There are 2 holes to solder an external switch (in this case, do a solder point between the pads above the “on” mark)
- There are 2 holes to solder the battery (instead of using the connector)
- J_NTC1 is to solder a 10k NTC for optional temperature measurement (but software get temperature only once a day)

Starting the wind sensor

Now that all components are connected together, let’s switch on the system.

- If using the Lipo battery, turn the shield switch on
- If using 1.5V batteries the power is on as soon as you place the batteries in their harness

The board will boot and you will see 10 flashes of the yellow led.

Then you will see 1, 2 or 3 long flashes. They mean that the system identifies your sensor.
1 flash means Davis sensor, **2** flashes means Peet sensor, and **3** flashes means MiSol sensor.
If the sensor recognition is wrong, the led desperately blinks. Verify the sensor connection, move a bit the wind vane position and push the reset button.

If the led keeps blinking, this means that the sensor is wrongly connected or the sensor type is not recognized or has any default.

The sensor starts immediately the measurements (1 measure every 3s). Every 5 minutes, minimum, maximum and average wind speed and average wind direction values are computed. After 10 minutes, the 2 sets of previous measurements are send to Internet via Sigfox network.

It means that 10 min after power up you should see your measurements on OpenWindMap site.
SpotAir (https://www.spotair.mobi) will also reflect the same measurements after a little time (about a day) without any action from you.

Debug mode

You can put a jumper between GND and Pin 11 to enter the debug mode. In this mode, if you connect the board via the USB cable and read the USB input from your computer, the measurements will be printed.
To read the USB input, use the serial monitor from the Arduino IDE.
If you want to capture values in a file, you can use this Python script:

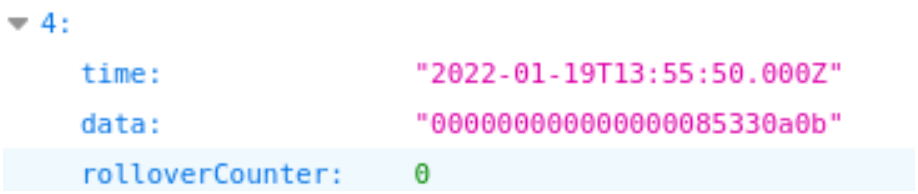
```
#!/usr/bin/python3
###############
## Script to listen serial port and write contents into a file for debug purpose
###############
## requires pySerial to be installed ( pip3 install pyserial )
# call it
#  serialListen.py [ <filename.csv> ]

import serial
from datetime import datetime
import sys
import time
```

```
now = datetime.now()
dt_string = now.strftime("%H:%M")

n=len(sys.argv)
if n>1:
   write_to_file_path=sys.argv[1]
else:
   write_to_file_path = "Imeas" + dt_string +".csv"

serial_port = '/dev/ttyACM0';
baud_rate = 9600; #In arduino, Serial.begin(baud_rate)

output_file = open(write_to_file_path, "w+")

while True:
   try:
      ser = serial.Serial(serial_port, baud_rate)
      break
   except:
      time.sleep(1)


while True:
   line = ser.readline();
   line = line.decode("utf-8") #ser.readline returns a binary, convert to string
   print(line);
   output_file.write(line)
   output_file.flush()
```

Admin reports

When you switch on the wind sensor and once a day, an admin report is sent.
The admin report can be seen here:
https://api.pioupiou.fr/v1/sigfox-messages/<ID>

An admin report is a data of 12 bytes, e.g.:



Only the 4 last bytes (8 characters) encode the admin report with the following infos:

byte 9	:	battery voltage (by adding 199.5 and dividing by 100)
byte 10	:	temperature (by substracting 50.5)
byte 11	:	sensor type (10=Davis, 20=Peet bros, 30=MiSol Shenzen)
byte 12	:	software version

Example of python script to decode the admin report:

```
#!/usr/bin/python3
import sys
entry=sys.argv[1]
print("Soft = %d" % int(entry[22:24],16))
print("Sensor = %d" % int(entry[20:22],16))
print("Temp = %d" % (int(entry[18:20],16)-50.5))
print("Vbatt = %f" % ((int(entry[16:18],16)+199.5)/100.0))
```