Decoding the Oregon Scientific V2 protocol

by Dominique Pierre (zzdomi)

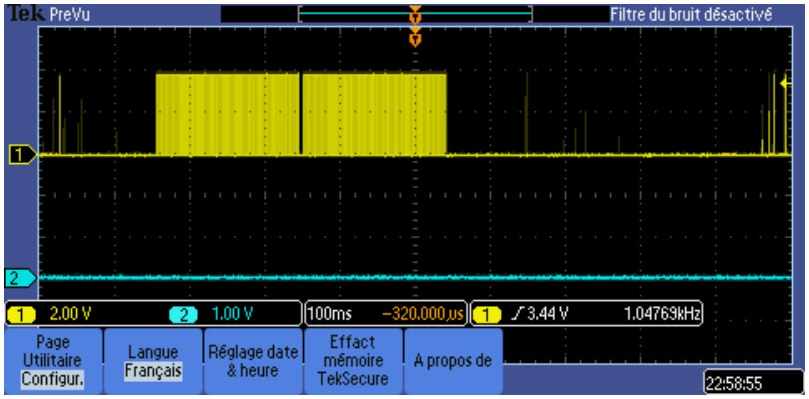
History

Few weeks ago, while I was googling about informations on the Oregon Scientific protocol, I discovered the Jeelabs site and found the post relative to the OOK decoder.
I tried it and was very happy to see the results from my THGR810 sensor. Unfortunately the informations emitted by my Geonaute sensor were not decoded.
This sensor is an Oregon Scientific sensor using the V2 protocol.

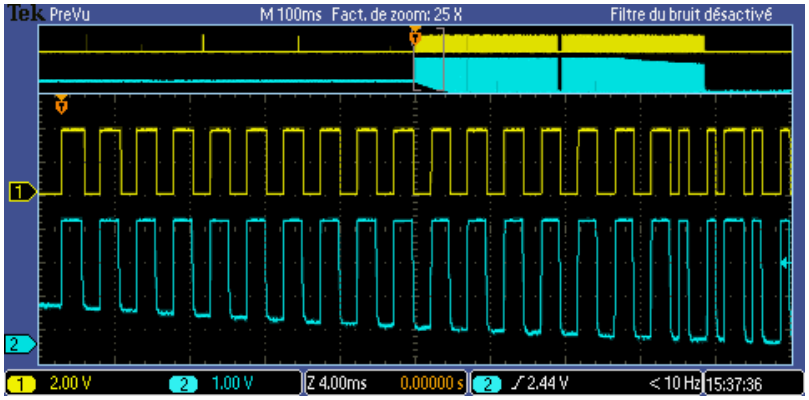
V2 Protocol

The V2 protocol uses a Manchester encoding. A long pulse toggles the current value and 2 short pulses leave the current value inchanged like the V3 protocol.
The major differences between the V2 and V3 protocol are:

- the packet is sent twice
- the preamble is based on long pulses and the start bit is a short bit.
- each bit of data is serialized twice but the second is logically inverted.



A global view of an OSV2 transmission.



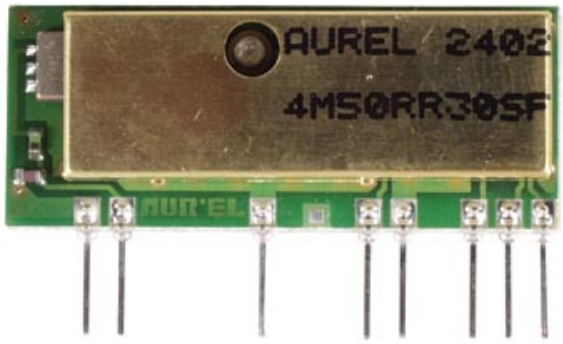
Some details on the preamble and the start bit.

Packet structure

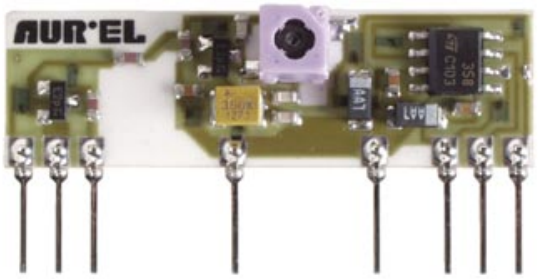
The informations are presented in the same way they are used in major domestic applications. You will find more informations in browsing the DomotiGa source code and in particular the CRFXComRX.class.

RF receiver

For this dev I tested two 433 mhz modules from Aurel: * RX-4M50RR30SF (~15 euros)



Very good selectivity * AC-RX (~7 euros)



Less selective, the signal is more noised.

Oregon Scientific sensors

The two first bytes of an OS frame represent the type of sensor. Here is all the sensors code I found on the net.

|.*Sensor name*|.Code|_.Type|
|Oregon-THR128
Oregon-THR138
Oregon-THC138|0x0A4D|Inside Temperature|
|Oregon-THC238
Oregon-THC268
Oregon-THN132N
Oregon-THWR288A
Oregon-THRN122N
Oregon-THN122N
Oregon-AW129
Oregon-AW131|0xEA4C|Outside/Water Temp|
|Oregon-THWR800|0xCA48|Water Temp|
|Oregon-THGN122N
Oregon-THGN123N
Oregon-THGR122NX
Oregon-THGR228N
Oregon-THGR238
Oregon-THGR268|0x1A2D|Inside Temp-Hygro|
Oregon-THGR810	0xFA28	Inside Temp-Hygro
Oregon-RTGR328N	0x*ACC	Outside Temp-Hygro
Oregon-THGR328N	0xCA2C	Outside Temp-Hygro
Oregon-WTGR800	0xFAB8	Outside Temp-Hygro
Oregon-THGR918
Oregon-THGRN228NX
Oregon-THGN500	0x1A3D	Outside Temp-Hygro
Huger - BTHR918	0x5A5D	Inside Temp-Hygro-Baro
Oregon-BTHR918N
Oregon-BTHR968	0x5A6D	Inside Temp-Hygro-Baro
Oregon-RGR126

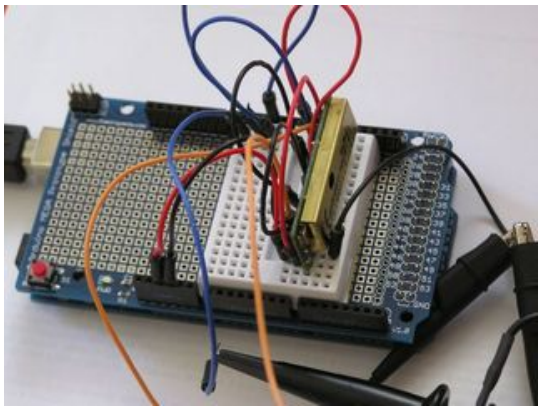
Oregon-RGR682
Oregon-RGR918|0x2A1D|Rain Gauge|
Oregon-PCR800	0x2A19	Rain Gauge
Oregon-WTGR800	0x1A99	Anemometer
Oregon-WGR800	0x1A89	Anemometer
Huger-STR918
Oregon-WGR918	0x3A0D	Anemometer
Oregon-UVN128
Oregon-UV138	0xEA7C	UV sensor
Oregon-UVN800	0xDA78	UV sensor
Oregon-RTGR328N	0x*AEC	Date & Time
cent-a-meter
OWL CM113
Electrisave	0xEAC0	Ampere meter
OWL CM119	0x1A*
0x2A*
0x3A**	Power meter

Preliminary results

Here is the log of the ookDecoder sketch (Ook_OSV2.pde) I modified:

```
[ookDecoder]
OSV2 EA4C106F7011D0D30300
OSV3 FA28A428202290834B46
OSV2 EA4C106F7011D0D30300
OSV2 EA4C106F7011D0D30300
OSV3 FA28A428202290834B46
OSV2 EA4C106F7011D0D30300
OSV2 EA4C106F7011D0D30300
OSV3 FA28A428202290834B46
OSV3 FA28A428202290834B46
OSV2 EA4C106F6011C0A30600
OSV2 EA4C106F6011C0A30600
OSV2 EA4C106F6011C0A30600
OSV3 FA28A428202290834B46
OSV2 EA4C106F6011C0A30600
OSV3 FA28A428202290834B46
OSV2 EA4C106F6011C0A30600
OSV2 EA4C106F6011C0A30600
```

The code has been tested on my Arduino Mega:



…but should run as is on JeeNodes.

Instead of using the analog pin, I used the external interrupt5 for the pulse length measure.
The specific code for the Mega is conditionally compiled with the define AVR_ATmega1280.

Validation

Validation as been done with a program which decodes the Oregon frames:

```
Simulate: 50FA28A428202290834B46
50FA28A428202290834B46   TH2[10250] THGR810 CH 10 addr:28 temp:22,2°C | 71,96°F hum:39 Dry      Ba
ttery OK bits=80
Simulate: 50EA4C106F301190630600
50EA4C106F301190630600   TEMP2[28417] THN132N,THWR288,AW131 CH 1 addr:6F temp:11,3°C | 52,34°F  Ba
ttery OK bits=80
```

Improvements

In order to have a robust decoding, it would be interesting to have a checksum control based on the sensor type for each packet.

Conclusion

With this small add-on you are now able to decode the messages from all Oregon Scientific sensors. I omitted the decoding of the V1 protocol because it is less and less used. People interested in that development can have look at the page of Alexander Yerezeyev.

Happy sensoring!

Files

Ook_OSV2.pde	15.9 KB	2012-06-20	jcw
Aurel_ACRX.jpg	11.8 KB	2012-06-20	jcw
Aurel_RX4M50RR30SF.jpg	14.3 KB	2012-06-20	jcw
OSV2Frame1.PNG	6.27 KB	2012-06-20	jcw
OSV2Frame2.PNG	8.11 KB	2012-06-20	jcw
MegaProtoSmall.jpg	30.2 KB	2012-06-20	jcw