Bee-IOT, System for measurement of temperature and humidity in Bee hives

A short background
Simple project for measurements in bee hives with the aim of learning how to improve winter survival of bees.

During winter a lot of bees die and if the bee population gets to small, they cannot keep temperature up and the whole hive dies. Improving winter conditions in the hive can hopefully make more hives survive the winter and produce honey the next year. When you open a hive in spring it can look terrible inside. Dead bees, moisture, mold and bee poo in a mess makes you wonder how anything can live in there.

Winter losses of hives is a problem that varies a lot between different parts of the world. In Sweden, where I'm active, the winter losses are about 10% in average and in the US it is over 50%, at least in some areas. That’s the highest winter losses I know of.

When measuring in a bee hive you almost always have to rely on battery power since the hives seldom are in reach of any electric power installation. So the equipment needs to be energy efficient to enable it to last through the winter without opening the hive to change batteries.

My set up
I have chosen to use the Arduino mini pro as the active part in the hive. It is used in many similar projects and can be used over a long period of time using very little energy, especially if you remove the power LED and the internal voltage regulator. And it is very small.

For temperature and humidity measurements I’ve chosen the BME280 from Bosch which is accurate, small, power efficient and it goes all the way up to the dew point, 100% relative humidity. You also get barometric pressure but I don’t use it in this case.

To transfer the data from the hive I use a nRF24L+ which is a very small radio communication device that also is energy efficient. The drawback is short range and you need a free line of sight between transmitter and receiver, at least if you try to stretch the distance in between. It is also very sensitive to noise on the power supply. There are two major versions of this radio. One with the antenna integrated on the circuit board and one with external antenna and some extra amplification of the signal. I use the simple on in the hive and the more advanced as receiver. I also plan to use one as a “relay station”, disguised as a birdhouse, to extend the reach and go around a corner of a building.

I connect the units together on a PCB which I recently got information about from a colleague. A battery box with 3 AA batteries provide the power through a voltage regulator and I use three capacitors for filtering.

Power supply and electric noise
Since noise on the power supply might impair the function of the radio I first wanted to make sure my setup worked in this respect. There are different ways to connect power to the PCB but the ones described did not quite suite my needs. I have 3,3 V Arduino boards and want to

use them without the internal voltage regulator. I also don't want to use any voltage booster since it's an inefficient way of using power as well as a well-known source of electric noise. So I use 3 AA batteries giving a nominal 4.5 V and a regulator getting it down to 3.3 V to the board. For filtering I use one capacitor on the battery side of the regulator, one at the connection of the board and one at the radio. All of them are 10 uF tantal capacitors, 6 V on the board and 10 V at the regulator. Recommendation is to use capacitors in a voltage range 2 - 3 times higher than your actual voltage. Higher voltage rating also gives higher leak current so you want to keep it down when energy saving is an issue.

Testing

To see if the power supply to the radio was reasonable noise free I made two units with just the Arduino and the radio connected. A simple program in one unit was sending the programmers happy message "Hello world" to the other every 5 seconds. Measuring with an oscilloscope at the power connection of the radio I had a voltage swing of less than 5 mV when the radio was idle or receiving and 20 mV when the radio was sending.

I also tested the range putting the sending device under the roof of a hive and walking away with the receiver, connected to my laptop. With the simple version of the radio on the receiver I got about 10 meters range. With the one with proper antenna I got well over 100 meters and lost signal when I couldn't keep within line of sight.

So I concluded that my setup with respect to power and radio was ok.

I also tried to make a modification and replace the on board antenna with a quarter wave dipole made from copper wire, as described on various web pages. When used as a transmitter it gave a slightly reduced range. When used as a receiver the range was extended to about 50 meters. Since I need almost 100 m, that was not an alternative for my needs.

Problems arose

After I connected the temp- and humidity sensors the Arduinos began to behave in a strange manner. They could start and run for a while, stop and start over or do just anything but run the program properly. Behavior got quite unpredictable. Some worse, some better.

I did have some soldering problem initially. At least I thought it was a soldering problem because my first boards where a bit difficult to solder and they didn't work ok. I use solder with flux but especially with the Arduino boards it was difficult to get the solder adhere to the connection surfaces. So I got some extra flux which made soldering easier and hopefully better.

With that extra flux I know the soldering looked ok so I suspected there might be a voltage problem. The batteries deliver nominal 4.5 V and the regulator is supposed to deliver 3,3 and it needs >1 V difference to keep this. With batteries that has begun the decline in voltage there might be an issue. So I put fresh batteries in a unit that had performed badly and the worst batteries I had (1.3 – 1.4 V) in a unit that had performed ok. There was no difference in behavior.

So I started to suspect the cables on the I2C bus which I use to connect the sensors. As a rule of thumb the cable should be shorter than 1 meter, mine where roughly 0.5 meter. But that also depend on application and quality of the cable. So I tried another cable with better performance and shorter cables but no improvement. I even disconnected the sensor and still had the unpredictable behavior so it couldn't be the cables.

Eventually I measured on my soldering iron which was connected to a power outlet without protective ground. Between the soldering iron and a heat radiator (water system, not electric) I measured almost 200 v AC. The current was not high. I got about 0.5 mA unstable reading on the last digit of my instrument. With the soldering iron in a socket with protective ground I got 0 v and consequently 0 mA.

So bottom line is that I probably destroyed most of my Arduino boards by not having the soldering iron in a socket with protective ground. For this reason and that the problems have taken much time to investigate I decided to make a simplified setup for this record.

Simplified setup
One unit measuring and sending result over the radio. One unit receiving data over the radio and sending it over a serial line to a LoPy4, serial port(1). The LoPy sends it over WiFi to the server at PyCom where it is displayed in a couple of diagrams. The whole setup is powered from the USB-outlet of a laptop which also monitors the serial port(0) of the LoPy. There are also rather rudimentary programs in each unit for the task.

I will make the final setup for the bee hives when I get some new Arduino boards.

Bill of material:

Unit one, transmitter:
1 Arduino mini pro, 3.3 V version
1 nRF24L+ with chip antenna
1 BME280 (Aliexpress China, no European supplier have a reasonable price tag, sorry)
Wires to connect them or a PCB https://www.openhardware.io/view/4/EasyNewbie-PCB-for-MySensors

Unit two, receiver:
1 Arduino mini pro, 3.3 V version
1 nRF24L+ with external antenna (Aliexpress China)
1 PCB https://www.openhardware.io/view/4/EasyNewbie-PCB-for-MySensors

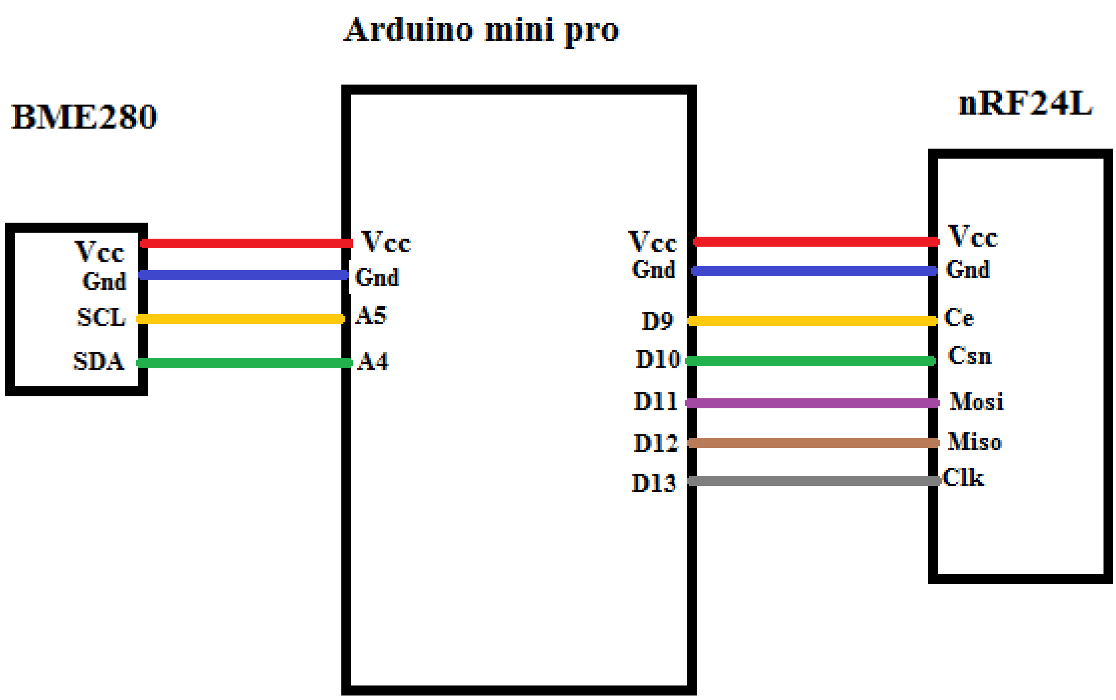
LoPy:
1 PyCom LoPy4
1 PyCom Expansion board 3
1 breadboard (optional)
1 backup battery (optional) (Be sure to check polarity on the connector!)

Battery power supply:
For each unit to run on battery

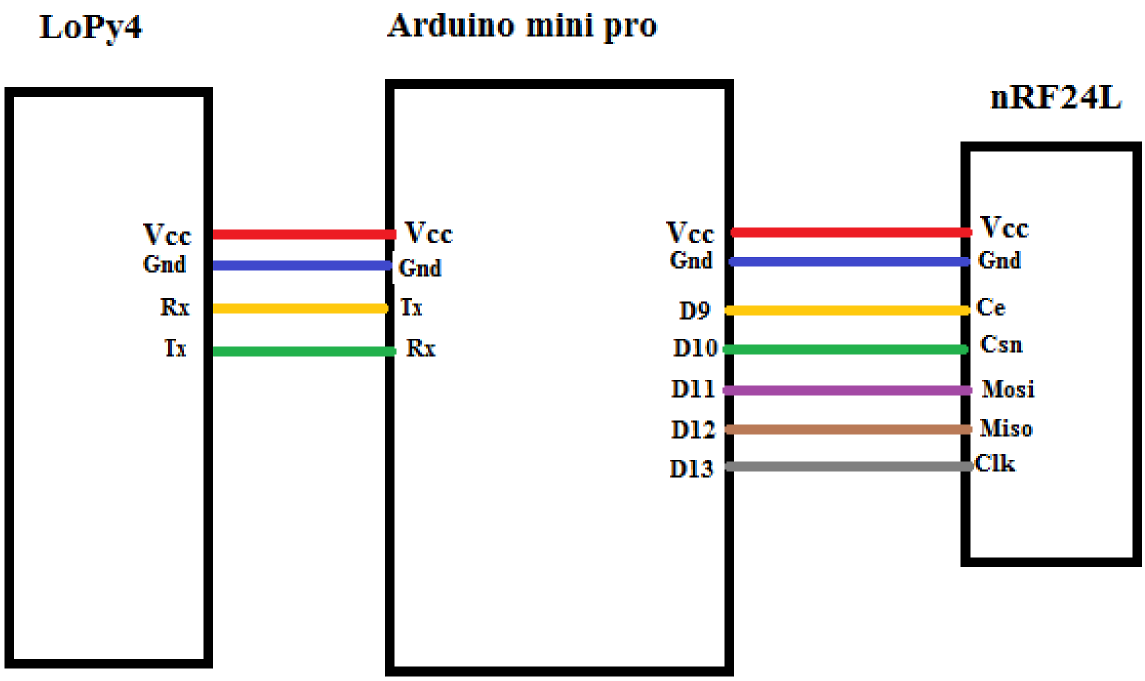
1 battery box 3xAA
1 Voltage regulator MCP 1700 3.3 V
1 tantal or ceramic capacitor 10 uF 10V
1 3-pole connector

Additionally you need some cables for interconnection and a FTDI (make sure you get the right voltage for your Arduino) for programming the Arduino boards since the mini pro does not have an integrated USB port. If you by one, make sure it has the right voltage for your Arduinos. And of course USB cables to connect to both Arduino and LoPy which in my case have different connections. Mini-USB for Arduino and micro-USB for the LoPy.

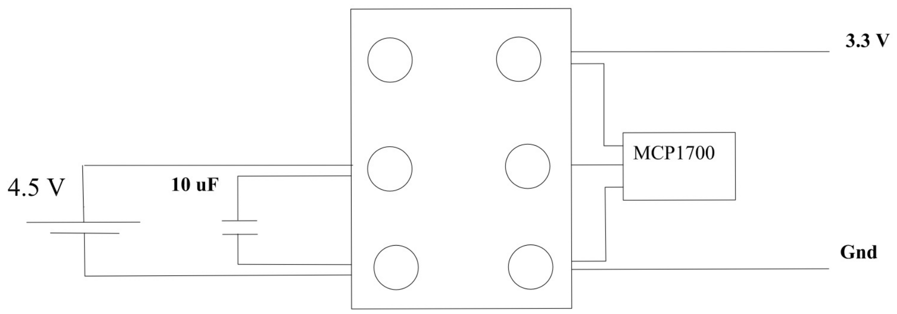
Wiring:
Unit one:



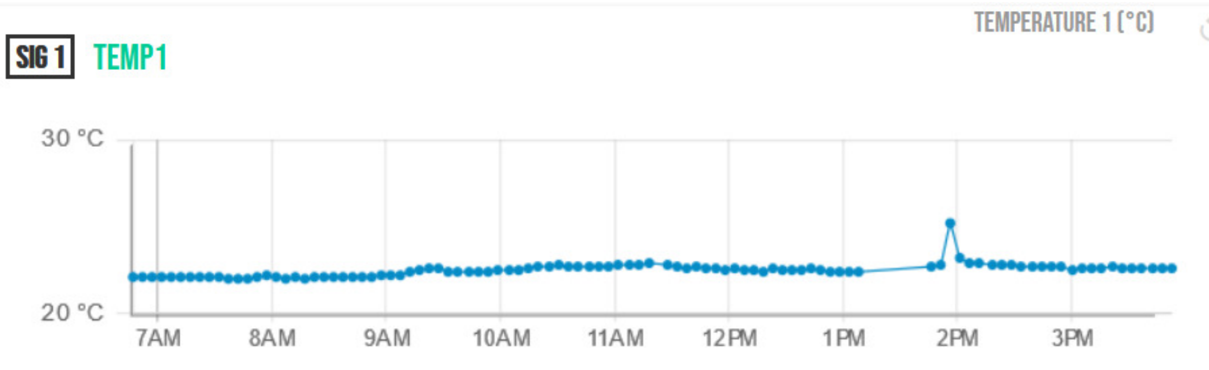
Unit two:



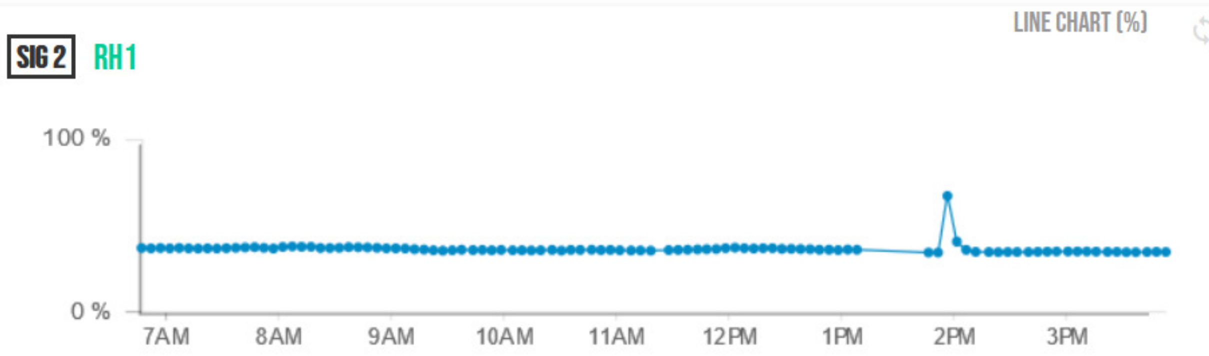
Connection of battery pack with regulator



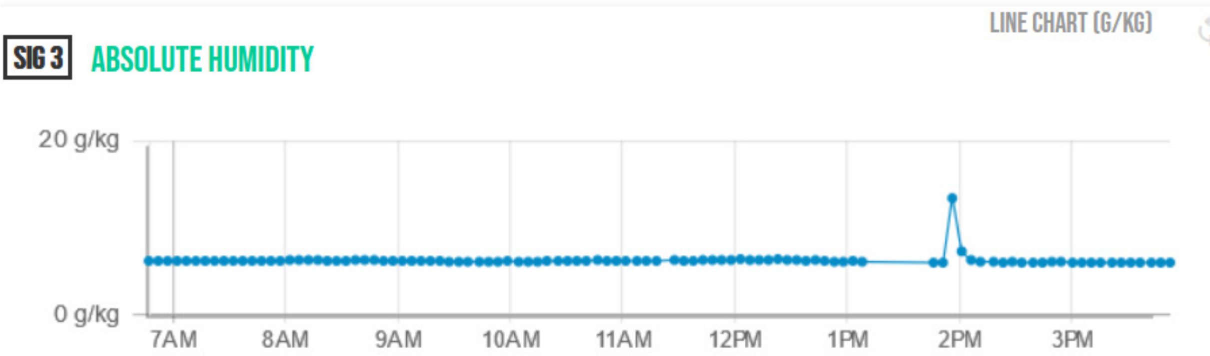
Graphs with measured values:



Temperature



Relative humidity



Absolute humidity calculated from temperature and relative humidity:

$$AbsHum = round((((38.214 + Temp * 1.698 + (Temp ** 2 * 0.1871)))/ 10) * (RelHum/100), 1)$$

The peak in the diagrams at about 3/4 on the time line is caused by me breathing out on the sensor. Just to make something happen on the otherwise quite boring, almost straight lines.

Future activities:

I will make some new board assemblies with the soldering iron connected to protective ground so they will hopefully be more useful. The sensors will be placed in queen cages to protect them from the bees that put wax on everything that comes into the hive. These assemblies will be put to work in december when this year's last treatment against varroa mites is done. This is done when the temperature gets lower than 5 °C and the bees form their winter cluster. Sometimes around Lucia and Christmas we usually get that temperature.

There will be three hives with active measurements during this winter that will send data via a relay station to my LoPy and further to the Internet. Of course that will also need some reprogramming from the now very simple and not very robust programs.