BOREDOM PROJECTS

(/)

Additional Environmental Sensors for Home Energy Monitoring System

Details

⚙▾

📅 Created: 31 August 2014
📅 Last Updated: 15 May 2017

Contents

---

1. The Purpose (Mission)

- To integrate various environmental and security sensors into my Home Energy Monitoing System (/index.php/projects/home-energy-monitor)
  - Outdoor temperature
  - Indoor temperature
  - Humidity
  - Atmospheric pressure
  - Door/window security sensors

2. Components

- Concept based on OpenEnergyMonitor (http://www.openenergymonitor.org) project
- Hardware: JeeNodeUSB (http://www.digitalsmarties.net/products/jeenode-usb), Arduino-Pro-Mini (http://arduino.cc/en/Main/ArduinoBoardProMini), RFM12B (http://openenergymonitor.org/emon/buildingblocks/rfm12b-wireless), BMP085 (http://www.adafruit.com/products/391), DS18B20 (https://www.sparkfun.com/products/245)
- Software: Arduino IDE (http://www.arduino.cc/en/Main/Software), EmonLib (https://github.com/openenergymonitor/EmonLib) and RFM12B library by LowPowerLab (http://lowpowerlab.com/blog/2012/12/28/rfm12b-arduino-library/)

3. Progress Report

3.1. Outdoor Temperature Sensor

Naturally, the main feature of such device had to be autonomous, wireless operation. This means battery powered operation, low current consumption and wireless data transmission.

RFM12B transceiver was an obvious choice, since I already used them in other parts of my Energy Monitor.

For the temperature sensor itself, I chose DS18B20 (http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html) made by Maxim. They were easy to use and Arduino library for it was readily available.

Finally, for the processor, I decided to use Arduino Pro Mini board, made by Sparkfun (https://www.sparkfun.com/products/11114), for its simplicity, which is what's needed for battery powered operation. The board detailed description is here: https://www.sparkfun.com/products/11114 (https://www.sparkfun.com/products/11114)
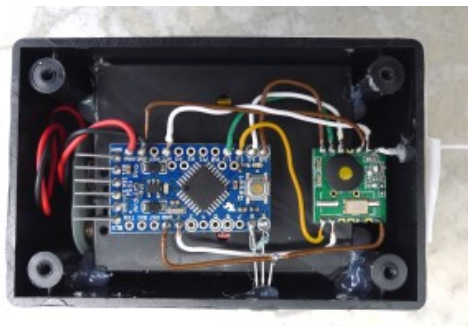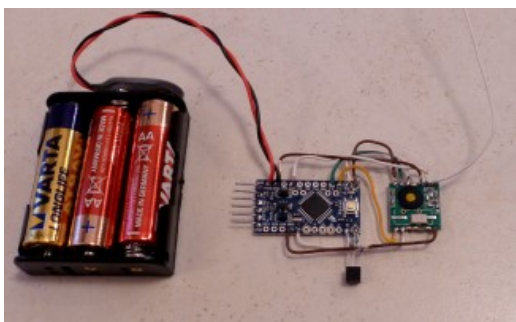
(/images/Articles/HomeEnergy/TempSensor_out.JPG) (/images/Articles/HomeEnergy/TempSensor_box.JPG)

All the above components are in sleep more for most of the time, only waking up for short periods to take a temperature reading and transmit data out.

Here's the schematic diagram of the sensor:
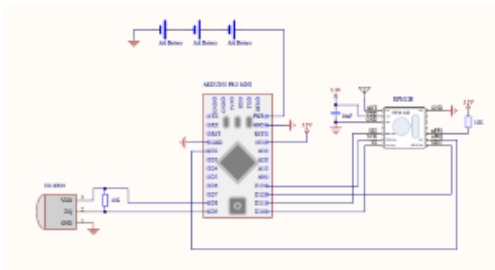
(/images/Articles/HomeEnergy/Outdoor_Temperature_Sensor_sch.pdf)

I chose to use 3 AA batteries to power the unit through RAW power input. The on board LDO has a low quiescent current, so during sleep times, the system would draw only about 100µA. And during wake times, the current would be limited by the temperature sensor + radio, anyways. I calculated that the battery should last about 5-6 months. We'll see...

UPDATE (Dec. 2014): The batteries are finally dead. After almost 8 months! I'd say it's pretty damn good! :)
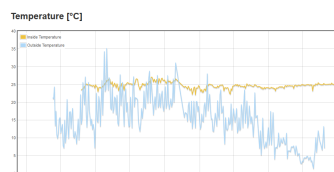
(/images/Articles/HomeEnergy/Temperature_2014.png)

And, of course, here's the Arduino sketch: **Sensor_Outside.ino (/images/Articles/HomeEnergy/Sensor_Outside.ino)**

Part of the sketch is a code enabling low battery threshold on RFM12B and reporting it back to the base together with temperature reading.

3.2. Indoor Temperature and Pressure Sensor

Since I already used JeeNodeUSB with RFM12B as a main data collection interface, it seemed easiest just to add the additional indoor sensor directly to it. The BMP085 (http://www.adafruit.com/products/391) from Adafruit was perfect for that, providing both Pressure and Temperature readings in one package.
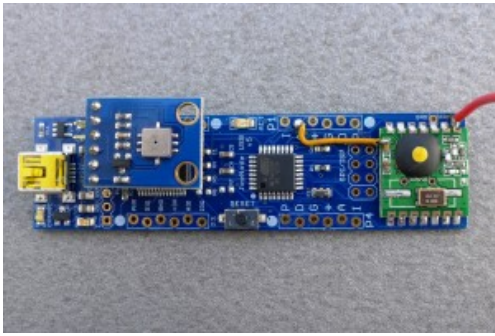
(/images/Articles/HomeEnergy/EmonTx_top.jpg)
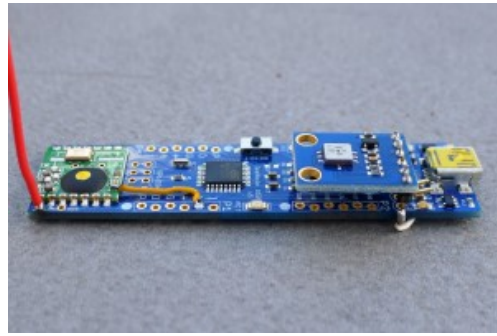
(/images/Articles/HomeEnergy/EmonTx_side.jpg)

(/images/Articles/HomeEnergy/EmonTx_det1.jpg)

(/images/Articles/HomeEnergy/EmonTx_det2.jpg)

(/images/Articles/HomeEnergy/EmonTx_bot.jpg)

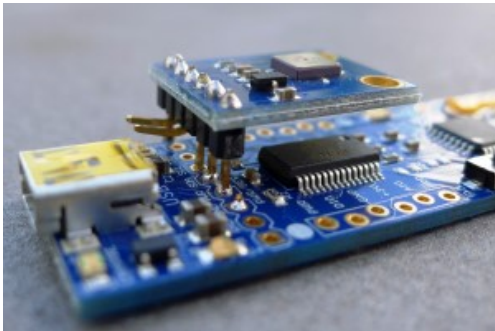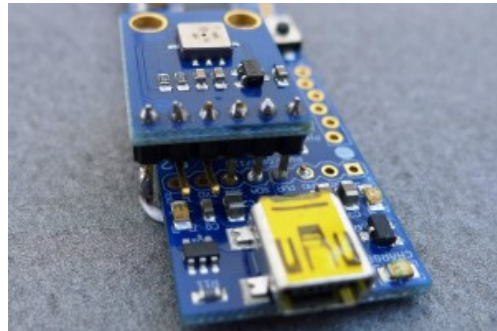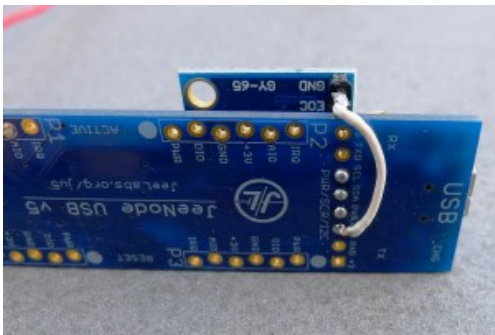Interesting thing that I discovered, which was new to me, is that absolute atmospheric pressure reading produced by the sensor needs to be converted to pressure at sea level. This is what is used by all weather reports. That means the following calculation is performed on the receiving end (by EmonCMS input processing engine):

$$p = p_0 * \left(1 - \frac{altitude}{44330}\right)^{5.255}$$

For excellent explanation of this topic see this Sparkfun tutorial (https://www.sparkfun.com/tutorials/253).

In addition, I set up a dummy pressure input that always shows sea level pressure, to be used as reference. This makes it easy to judge high or low pressure tendencies.

(/images/Articles/HomeEnergy/Pressure_dashboard.png)

This dummy pressure input is generated inside my RaspberryPi server running EmonCMS. I simply set up a cron job with this command to run a script every hour:

0 * * * * /home/pi/log_reference_pressure.sh

and the script itself is very simple:

curl -k -u "*username*:*password*" "https://127.0.0.1/emoncms/input/post.json?node=1&csv=1013.25&apikey=*emonApiKey*"

where *username*, *password* and *emonApiKey* must be replaced with actual parameters from your box, and 1013.25 is average sea level pressure in hPa.

3.3. Indoor Humidity and Temperature Sensor

This one is very similar to Outdoor Temperature Sensor, except here I used DHT22 - all-in-one humidity and temperature measurement device. The controller is again Arduino Pro Mini, powered by three AA batteries. And the radio is RFM12B.
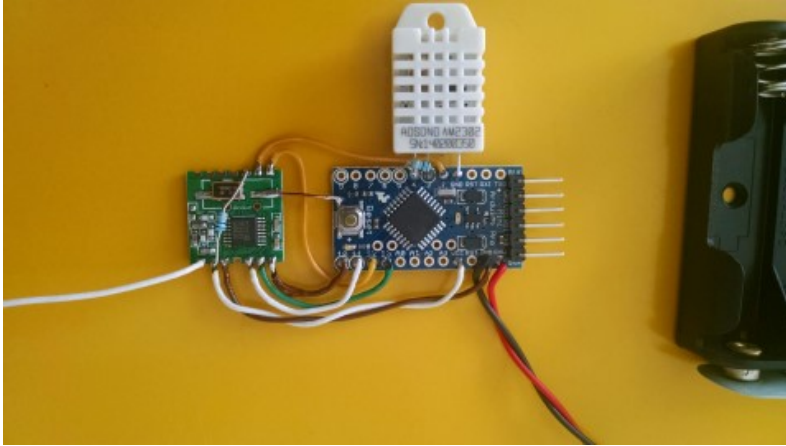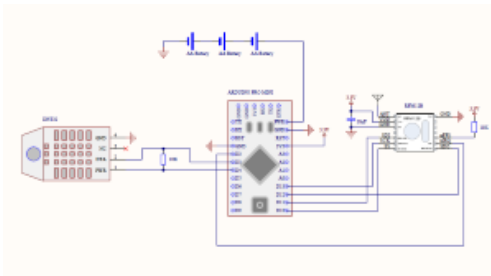
 (/images/Articles/HomeEnergy/Humidity_sensor_large.jpg)

I found that when idle, the DHT sensor draws about 1mA, which is too much for battery powered system. Therefore, I connected its VCC line to a digital output pin on Arduino, which is turned ON only during measurement time. However, to function properly, the power on DHT must be present and stable for more than 1 sec, before attempting to take readings.

I used DHT library made by Rob Tillaart (http://playground.arduino.cc/Main/DHTLib (http://playground.arduino.cc/Main/DHTLib)), however with one small modification - I commented out the very last two lines, which normally configure communication pin as OUTPUT and set it HIGH. The reason is, if this pin is driven HIGH and the power pin is driven LOW, there will be a useless current flow through the pull up resistor, connected between these two pins. Besides, if power supply to the sensor is cut off, there is no need to worry what state the comm pin is left in.

Here is the complete schematic diagram:

 (/images/Articles/HomeEnergy/Indoor_Humidity_Sensor_sch.pdf)

And here's the Arduino sketch: **Sensor_Inside.ino (/images/Articles/HomeEnergy/Sensor_Inside.ino)**

3.4. Door/Window Security Sensors

Although these do not strictly fall under category of environmental, but rather security related sensors, still I decided to include them in this post.

Basically, the sensors contain a Reed switch (https://en.wikipedia.org/wiki/Reed_switch), ATmega328P controller and RFM12B radio. All of this is powered by 3 x LR44 coin cell batteries. Since I planned to have several of these devices throughout the house, I decided to make a proper PCB, designed and ordered through Fritzing.org (http://fritzing.org). Here's the link to the design:

Fritzing Wireless Door/Window sensor (http://fritzing.org/projects/wireless-doorwindow-openclosed-sensor)

and schematic diagram:
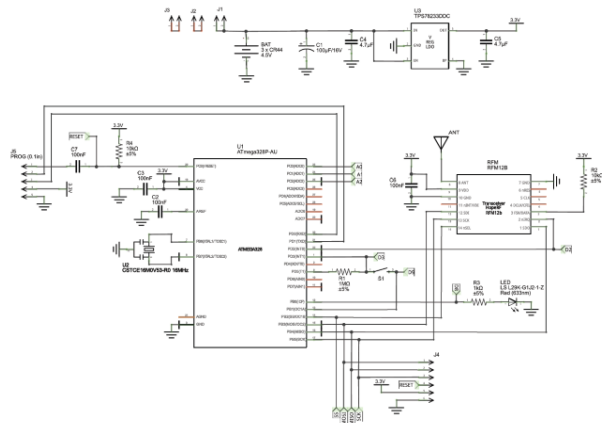
(/images/Articles/HomeEnergy/Door_Sensor_schem.pdf)

and some pictures:

(/images/Articles/HomeEnergy/window2_small.jpg)

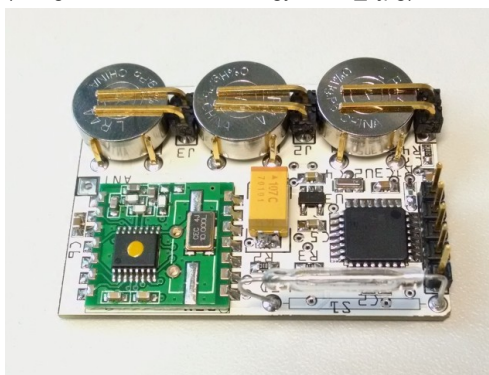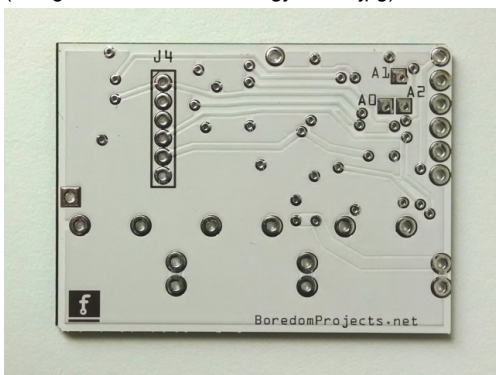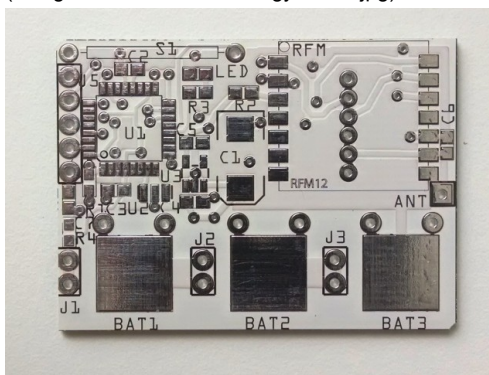(/images/Articles/HomeEnergy/door5_r.jpg)

(/images/Articles/HomeEnergy/door1.jpg)

(/images/Articles/HomeEnergy/door2.jpg)

(/images/Articles/HomeEnergy/door_pcb_bot.jpg)

(/images/Articles/HomeEnergy/door_pcb_top.jpg)

I must say, quality of Fritzing PCB manufacturing is pretty good. Although I had to pay about 70€ for 10 pieces. Not the cheapest service, I guess.

In order to make the sensor as small as possible, I had to use improvised battery holder connectors. I couldn't find any ready made connectors, surface mount or through-hole, which could fit three batteries into the small plastic case (https://www.buerklin.com/en/klarsicht-rechteckdose-mit-schiebedeckel/p/19l3272). These header pins are not so easy to solder, but they work pretty damn good!

Magnets and Reed switches I bought from Pollin.de: Magnet 7x3mm (http://www.pollin.de/shop/dt/MjA0ODU1OTk-/Bauelemente_Bauteile/Mechanische_Bauelemente/Montagematerial/Magnet_7x3_mm.html) and Reedkontakt PRK-2,5x16 (http://www.pollin.de/shop/dt/MTIzOTc1OTk-/Bauelemente_Bauteile/Mechanische_Bauelemente/Schalter_Taster/Reedkontakt_PRK_2_5x16_10_Stueck.html)

Connector J4 (2mm pitch) is used as ISP port, to program Arduino bootloader into ATmega328. This tutorial (https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard) has detailed procedure on how to do that. Once the bootloader is in, all the programming and reprogramming is done through J5 connector (0.1" pitch) in a normal way through Arduino software and a compatible Serial-to-USB converter.

To conserve battery power, the device is in sleep mode for most of the time. The reed switch is attached to hardware interrupt pin, which wakes up the micro, as soon as a change of state is detected. Tree packets of new state data are transmitted out (for redundancy) and device enters sleep mode again.

In addition to state change driven transmissions, the program also enables watch dog timer, set for approximately 1 hour. So, if nothing is happening, the sensor will wake itself up every hour to send out its current state. This is done to monitor sensor's health status from the central hub, in order to detect dead battery, for example.

Here's Arduino sketch for this sensor: **Sensor_Door.ino (/images/Articles/HomeEnergy/Sensor_Door.ino)**

Few notes on power consumption:

Brown Out Detector has a big influence on power consumption during sleep mode. The above code includes a way to turn it off.

Since external interrupt pin only works as a low level triggering, the direction of the pull resistor (R1) is flipped from GND to VCC. At the same time, the opposite pin of reed switch is flipped from VCC to GND. This is done to ensure that no matter whether the switch is open or closed, the next state change will always produce low level hardware interrupt. One side effect is when reed is magnetized (door is closed) there is a small current flow through R1 (acting as pull-down in this case), resulting in a few additional µA of current draw.

Measured power consumption from battery port (including LDO bias current):

- Sleeping - door opened 13uA
- Sleeping - door closed 16uA
- Active   - preparing 3mA (4msec)
- Active   - transmitting 25mA (3msec)

The active current consumption was measured by attaching a 6.8 Ohm resistor in series with supply and monitoring voltage across its terminals on a oscilloscope:
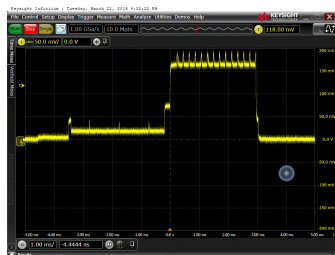
 (/images/Articles/HomeEnergy/Transmit_current_Rsense_6.7ohm.gif)
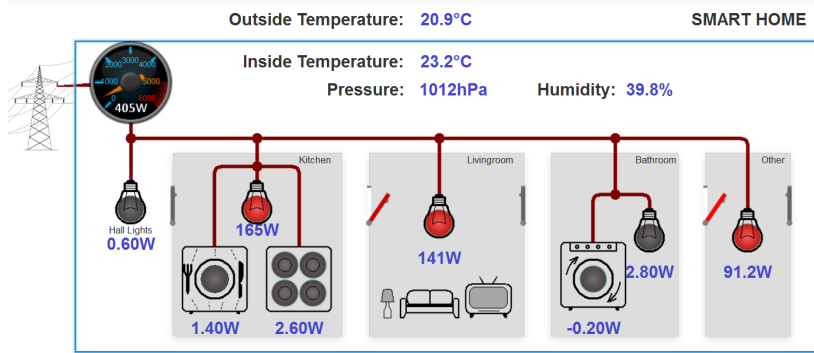
I estimate the batteries should last for about 8 months. Let's see...

UPDATE (May 2017): The batteries finally started to die. That's about a year of battery life! :-)

3.5. Dashboard for EmonCMS

Finally, with all these sensors, I was able to create this nice (i.m.o.) looking dashboard for my Home Energy Monitoring System (/index.php/projects/home-energy-monitor):

Inside Temperature: 23.2°C
Pressure: 1012hPa   Humidity: 39.8%

405W

Kitchen   Livingroom   Bathroom   Other

Hall Lights
0.60W

165W

141W

2.80W   91.2W

1.40W   2.60W

-0.20W

(/images/Articles/HomeEnergy/live_dashboard.png)

By the way, to represent door/window state, I created a simple Door widget, by modifying led widget example. Here is the JavaScript code:

door_render.zip (/images/Articles/HomeEnergy/door_render.zip)

Comments   (/index.php/component/jcomments/feed/com_content/8)

(/index.php/projects/sensors-for-hems#comment-254) **RE: Atmospheric Pressure ? — boredman**   2016-02-18 07:10

Quoting Georgi Tanev:

> Hello again
> I do not know if I dabble or anything else there, but graphics with atmospheric pressure is negative. If it can "fix" somehow?

I would guess you have a wrong formula in Node processing in your EmonCMS. Mine looks like that:

Order	Process	Arg
1	+	100000
2	x	1.06563
3	x	0.01
4	Log to feed	Pressure (feedvalue:1016.09)

(/index.php/projects/sensors-for-hems#comment-256) **RE: Atmospheric Pressure ? — gtanev**   2016-02-18 07:43
Thank you very much!
Comparing the measurement of atmospheric pressure sensor with the official airport which is close to my city and there is little difference. The second factor 1.06563 how is it determined? Does it matter the altitude or other factors?

(/index.php/projects/sensors-for-hems#comment-257) **RE: Atmospheric Pressure ? — boredman**   2016-02-18 14:27
The factor is determined according to this formula:

$$p = p_0 * \left(1 - \frac{altitude}{44330}\right)^{5.255}$$

SparkFun website has an excellent explanation of this:
(http://https://www.sparkfun.com/tutorials/253)   www.sparkfun.com/tutorials/253 (https://www.sparkfun.com/tutorials/253)
Good luck!

(/index.php/projects/sensors-for-hems#comment-260) **RE: Atmospheric Pressure ? — gtanev**   2016-02-22 09:06
Thanks again.
In my second factor get 1.019386 at 170 meters altitude.
Good luck to you!

Refresh comments list

RSS feed for comments to this post (/index.php/component/jcomments/feed/com_content/8)

Please login or register to post comments

You are here: Home (/index.php)   ▶   PROJECTS (/index.php/projects)   ▶   Additional Sensors for Home Energy Monitor

Like 0   Share

Login Form

👤 | Username

🔒 | Password

☐ Remember Me

Log in

Create an account ❯ (/index.php/component/users/?view=registration&Itemid=435)
Forgot your username? (/index.php/component/users/?view=remind&Itemid=435)
Forgot your password? (/index.php/component/users/?view=reset&Itemid=435)

Latest Article

- Robot Navigation using Stereo Vision (/index.php/projects/robot-navigation-using-stereo-vision)
- Additional Sensors for Home Energy Monitor (/index.php/projects/sensors-for-hems)
- Mobile Robotic Platform (/index.php/projects/robotic-platform)

Back to Top