# A low-cost wireless sensor network for long term monitoring of energy performance and sustainability of buildings

*Jevgenijs* Telicko[1,*], *Andris* Jakovics[1], and *Ivars* Drirkis[1]

[1]Latvian University, Laboratory for Mathematical Modelling of Environmental and Technological Processes, Riga Jelgavas 3, Latvia

**Abstract.** This manuscript describes the development of a wireless sensor system for long term monitoring of temperature, humidity and heat flux reading within building structural elements, including places that are hard to reach using wired sensors. The system was tested in cold Latvian climate in 3 different buildings. The main objectives during the development phase were the maximization of network operational lifetime, ensurance of work stability and maintenance cost reduction to make the system feasible for wide use in practical applications. An optimal radio module and microcontroller combination yielded sufficient signal range and data transfer stability, as well as efficient control of energy consumption.

## 1 Introduction

Nowadays, monitoring and automated control of building environment is one of the most promising paths towards energy efficient and comfortable use of buildings. Wireless sensor networks can be used for these applications, since there have several benefits. They are easier to install, gathered data is more accurate under certain conditions; also, wireless sensors can be installed in places that would be otherwise difficult to access using wired sensors. Aside from building monitoring, if sensors are put into place during construction, a network thereof can provide information on how materials properties change over time, as well as inform one of mold formation within walls and monitor other characteristics. The goal of the present project was to create a wireless sensor system for monitoring of temperature, humidity and heat flux readings inside the building structural elements over a period of at least 5 years. The challenge was to ensure stable data transfer using power efficient components and to minimize maintenance costs. This would allow for wider use of such systems in practical applications.

[*] Corresponding author: jevgenijs.telicko@gmail.com

## 2 Data transfer

The first objective is to create a radio system with satisfactory energy performance and a stable signal range of at least 10 m. Since time scales for heat flux, humidity and temperature changes within walls are in the order of several minutes, the sampling rate for sensors was set to one measurement per minute, to conserve energy.

### 2.1 Data transfer with esp8266

The initial idea was to use the ESP8266. It is a cheap microchip with built-in Wi-Fi and a complete internet protocol suite. In deep sleep mode, it has low power consumption and an I2C communication capability. Our tests have shown that in deep sleep mode with USB communication, current is about 0.14 mA (5V), and with no communication it can be even less than 15 µA. However, during data transfer, current can be up to 80 mA. The biggest problem with the ESP8266 was the Wi-fi connection time, which took about 3 seconds during which current is more than 80 mA, so it is not power efficient at all.

### 2.2 Nrf24 radio module

NRF 24 has very low power consumption in operating mode, 11.3 mA (3.3V) in TX mode, and it is cheap. It can be found on eBay for less than 0.8 $. After some tests with a standard open source library RF24, it became clear that it only supports up to 6 radio modules within the sensor network. This limitation comes from the fundamental capability of the NRF24, which supports only 6-pipe reception. To circumvent that, we used the open source library NRF24 Network created by James Coliz. The idea is to create a tree topology to handle more nodes with sensors. On the first tree level, one has a master node with the number 00, as shown in Figure 1.

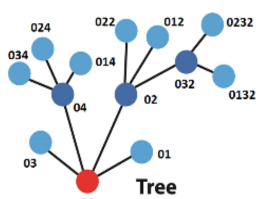


**Fig. 1.** NRF24 communication in tree topology with node addressing

The 2nd level can handle sensors or slave nodes for the upper level. The largest node address is 05555 and with that the network can handle up to 3125 nodes.

#### *2.2.1* Minimization of energy consumption

Arduino Nano was chosen for first test of the radio. It is a widely used board based on the ATmega328P. It can be programmed via Arduino IDE, so after the code is written it is easy to reproduce the result. Having connected the Arduino and the radio we obtained a much better result – during data transfer, the average current was 23 mA and data trasfer was mach faster than with ESP8266: 0.02 sec. However, in sleep mode the current was 4.65 mA. Since there are many power consuming elements on boards, such as a stabilizer, ch340 for USB communication and LEDs, we decided to make our own board without unnecessary components. As the microcontroller (Atmega328P) from Arduino Nano was working fine with NRF24, it was decided to use it in all other boards. For a more power efficient sleep mode, it was decided to use the Real Time Clock (RTC) module to wake the board up from deep sleep. We chose the MCP7940M RTC because of its low power consumption rate, with 1.2 μA operating current at 3.3 V . With this configuration, in deep sleep power down mode (DSPD) current was ~ 0.14 mA at 3.3V and tests have shown that RTC does not reduce energy consumption, as seen in Figure 2.

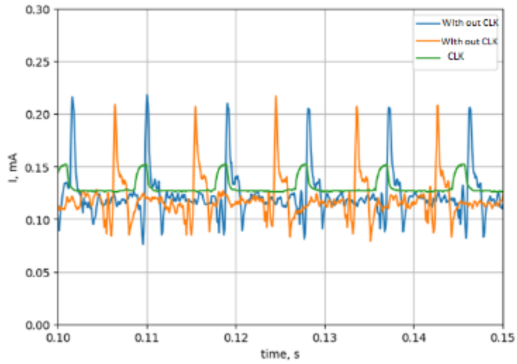


**Fig. 2.** Atmega328P with NRF 24 in deep sleep power down mode (3.3V)

Therefore, RTC was not used in future versions. Instead, the system uses a timer watchdog, which also prevents the node from getting stuck in the middle of code execution – should that happen, the node is reset. Looking at the ATmega328P datasheet, current values are considerably different from expected power consumption. We found that there are some modules within ATmega328P that consume power in DSPD mode. Switching off ADC and brown-out detector, and reprogramming all unnecessary pins as inputs, the average current in sleep mode was ~ 36.5 μA (3.3V), as seen in Figure 3.

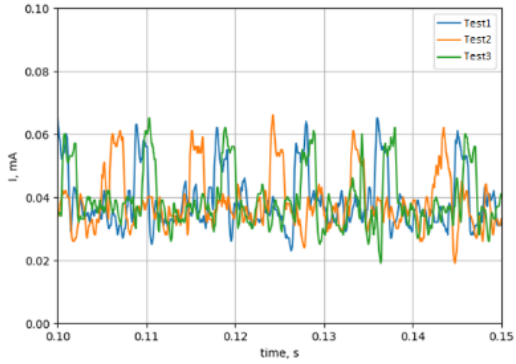


**Figure 3.** Current diagram at 3.3V of Atmega328P with NRF 24 in sleep mode after switching of unnecessary modules

Therefore, the average power consumption was ~ 0.1386 mW, which implies the theoretical operational time of ~ 8 years for 18650 battery, without considering battery time discharge.

#### *2.2.2* Signal improvement

Tests indicated that some packets are lost between modules. Experiments lead us to conclude that the following factors can influence signal quality: sensor geometry, the fact that NRF24 modules are not all perfectly identical, and the choice of the correct frequency. There is even a difference in performance between a sensor resting on a table and one that is being held in hand. We found that shielding the radio with aluminum foil and connecting a ceramic capacitor to the antenna and ground greatly improves the signal, especially when combined with the right radio frequency. For better data transfer and greater signal range, nodes which send data from sensors and are connected to power supply at all times can be equipped with the NRF24 module with a power amplifier (PA) and an SMA antenna. As power efficiency is not important for these nodes, Arduino Nano can be used. Standard 50 mA output current provided by Arduino Nano is insufficient, since it is expected that the above-mentioned module could require more than 90 mA. In order to use the module with an amplifier, one must also have a 3.3 V stabilizer installed. Two radio modules with PA and SMA were tested: NRF24l01+ and E01-ML01DP5. Results indicate that E01-ML01DP5 is by far superior, both in terms of stability and data transfer range. Also, unlike NRF24l01+, it does not require additional shielding.

### 2.3 Data transfer to Internet

Raspberry Pi (RPI) was used for data transfer and centralized storage. It is essentially a small PC with a Linux interface, which is equipped with wired and Wi-Fi Ethernet hardware, and an SPI interface for connection with NRF24. RPI is also fully capable of data processing (with a wide range of supported programming languages) and controlling different smart house systems. The main concern with long term RPI use is having good power supply, using supported SD carts (with sufficiently high speed), as well as keeping RPI within the working temperature range, using a radiator with a fan.

## 3 Sensors

The Atmega328P microcontroller has many communication protocols for external hardware. It also comes with a built-in analog to digital converter (ADC) for analog signal measurements. This makes the system compatible with a wide variety of sensors. It is essential sensors have low power consumption rates and short measurement intervals, because during the sampling process the plate is in the operational mode and consumes ~18 mW.

### 3.1 Temperature and humidity sensors.

For temperature and humidity measurements, we chose the HDC1080 sensor. Firstly, this module has a high data acquisition speed, taking only 6.5 ms for humidity sampling and 6.35 ms for temperature. Secondly, it has a low power consumption rate: 1.3 μA in operational mode and 100 nA in sleep mode. Signal distortion probability for data transfer to the plate is minimized, since data is transferred digitally via the I2S protocol. One of the aspects of using these sensors is that they can request calibration.

### 3.1 Heat flux sensors.

Heat flux sensors output an analog signal in the microvolt range. Since the built-in Atmega328P ADC does not have sufficient precision, we added an external 16Bit ADC ADS1115 to the circuit. This module performs measurements in 9 ms or less (using a non-standard library) with a range of $-0.256$ to $+0.256$ V and a 0.0078125 mV step in (x16 gain) differential mode. It also uses the I2S protocol to communicate with the microcontroller. The signal is relatively weak and, although it only travels a small distance through the cable to the ADC, the first round of tests showed that there was considerable noise and distortion in the incoming data. To improve measurement quality, the wires connecting sensors and the ADC were replaced with their shielded counterparts (shielding was grounded at the ADC). Afterwards, a ceramic 4.4 nF capacitor was added to the circuit to physically average the incoming data. In addition, a 10-point moving average window was applied. We used sensors from three different manufacturers (Ahlborn, Hukseflux and Almemo) for a series of tests, wherein the three sensors were installed into a wall in close proximity to one another. Two sensors were attached to the wall using acrylic glue, and the third one was duct taped along its perimeter. After one day of testing, when the temperature had stabilized, there was a notable difference between the readings of the taped sensor and the two glued sensors. It was found that good surface contact is crucial for this type of measurements and, after the duct tape was removed and the sensor was glued to the surface like the other two, the discrepancies in sensor readings disappeared.

## 4 Testing under real conditions

Three benchmark structures (constructed within the framework of the "Viedo risinājumu gandrīz nulles enerģijas ēkām izstrāde, optimizācija un ilgtspējas izpēte reāla klimata apstākļos" project) situated in the botanical garden (University of Latvia) were chosen for the first round of long terms tests. Each structure was built using different materials and the maximum distance between these structures is 10 m. 12 sensors were installed into each building and a master node for each sensor set was placed in between the other two buildings. Sensors were placed into floors (between insulation layers), walls, attic and the ventilated façade, see Figure 4.


**Fig. 4.** 3 different heat flux sensors and temperature/humidity sensor inside of the wall.

It immediately became clear by that placing the sensos at an optimal angle with respect to the receiver plate (which sends data to its master node) one can greatly improve signal quality.

After 6 months of testing at various temperatures, ranging from 25 degrees Celsius to sub-zero values, it was evident that only 0.5 % of data packets were lost. Heat flux measurements were much better compared to the older sensors, some of which still remained within the test buildings. The difference between two measurement points on average did not exceed ~0.12W/ $m^2$, as opposed to 5W/ $m^2$, which occurred when sensors were simply duct taped to the interior wall surface. Temperature sensor readings were in very good agreement with those of the system that was previously installed within the test structures. Relative humidity (RH) measurements differed from those of a data logger placed in close proximity by 5% RH. Tests indicated that for temperatures close to 0 or lower, sensor readings could be unrealistic, which is probably due to the formation of ice layers on sensors' active surfaces. It should be noted that RH measurements are indirect, and many factors can affect measurement quality [1].

The second test of the developed system and, at the same time, its practical application, took place in the "SALA" business center in Aloja. It was planned to have a smart building management system (BMS) installed within the building to maximize building exploitation efficiency and interior comfort for its occupants. Implementation of such systems can take a significant amount of time, which is why the newly developed wireless system was installed within the building in advance – data gathered by this system will later be used to establish an algorithm for optimal building control via BMS.

## 5 Summary

The developed wireless system can be used as a fast and cheap solution for monitoring of various characteristic parameters of buildings, wherein such a system was not initially implemented. Since data transfer occurs via a wireless network, this solution could be convenient, as it allows one to access places that are hard to reach using wired sensors, for which cabling installation might be an issue. The central unit of the system being the RPI, it is relatively easy to integrate this solution into BMS. In addition, one can create various algorithms for data processing, since RPI is Linux based and thus supports many programming languages. Processed data could then be used to compute, for instance, mould growth risk indices from temperature and relative humidity datasets, or the U value, given heat flux and temperature readings. Linux also enables one to use a wide range of software: for example, VNC (a remote desktop application) can be used for convenient remote control of the system from different devices.

This research was done with financial support of European Regional Development Fund, project "Development, optimization and sustainability evaluation of smart solutions for nearly zero energy buildings in real climate conditions" (1.1.1.1/16/A/192).

## References

1. M. Bogdan, Humidity Measurement System, ICVL, 2 (2016)