A Low Cost Platform for Sensor Network Applications and Educational Purposes

Arian DIKOVIC, Gordan SISUL, Borivoj MODLIC

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb

arian@kset.org, gordan.sisul@fer.hr, borivoj.modlic@fer.hr

Abstract. *In this paper we describe the design, key features and results obtained from the development of a generic platform usable for sensor network applications operational in the ISM band. The goal was to create an open source low cost platform suitable for use in educational environment. The platform should allow students to easily grasp the fundamentals of wireless sensor networks so special attention was paid to basic concepts related to their functioning.*

Two versions of this platform were designed, the first one being a proof of concept and the second one more adequate to field test and measurements. Practical aspects of implementation such as network protocol, power consumption, processing speed, media access are discussed.

Keywords

Sensor network, media access control, microcontroller, GFSK, Si4432, low power, star topology, battery power.

1. Introduction

Wireless sensor network usage and deployment is inevitable in today's industrial environment [1]. Home automation, consumer electronics, security systems, structure monitoring and intelligent agriculture are just some areas that are profiting from the development of remote measurements and control.

Bearing that in mind, we focused on developing a configurable and simple platform that can be used as a development and learning tool. We did not want (or try) to create a state of the art platform, but rather one where basic concepts can be easily shown and explained so they can be adopted by students. On the other hand, fine tuning the platform for a specific application could give us a reliable workhorse for a more complex sensor network on which measurements can be taken. Our main goals were modularity, hardware availability, price, and most of all clear and simple firmware coding. The language of choice was C, allowing for code portability and reusability. By making things simple (no heavy software stack), we tried to stress that PHY and MAC layers play a big role particularly for low duty cycled radios with small traffic payloads [2].We tried to use open source solutions as much as possible to ease code development and its maintenance. The platform is composed from two main building blocks: an 8-bit microcontroller (AVR ATMega16L, Atmega8A or ATTiny2313) and RFM22 radio module based on Si4432 transceiver chip.

2. Platform Design

As every wireless sensor node can generally be divided in three main building blocks (power supply + RF transceiver + microcontroller) we concentrated on their evaluation before choosing the most adequate for our needs. To simplify the process of platform design and development we decided to highlight the following things:

- Node partitioning design
- Power source
- Network topology
- Media access control

Related to network topology the choice was between Star, Peer to Peer and Cluster-Tree topology. This choice will impact directly the complexity of the software running on the nodes so star topology was implemented. One advantage here is that the main node in the star does not have to be battery powered so we can sacrifice power consumption for receiver sensitivity, transmit power and processing time.

How the nodes access the media and relay the information heavily impacts energy management. As we are dealing only with the basics, a very simple way of addressing this problem is described later.

2.1 Node Partitioning Design

Two options were taken in consideration: one chip solution (SoC) with integrated RF transceiver and microcontroller or separated integrated circuits for each task [3] (a and b In Fig. 1). For future testing and education purposes we think that we can benefit more from having the two separated.

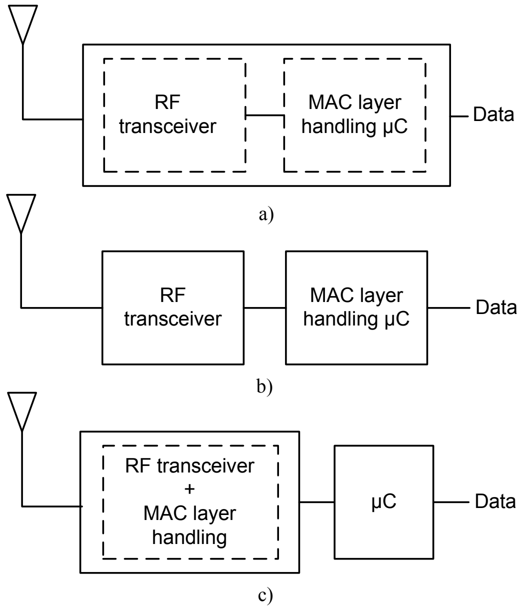

Fig. 1. Node partitioning possibilities.

In reality the line between them is not so sharp. For example, the transceiver module is able to packetize and calculate checksum for the received data from the microcontroller. Retransmission is also easy since the module keeps the last transmitted data buffered. In that way we are left with more computational power available for other purposes.

2.2 Power Source

Usually not of primary concern when we already have an infrastructure with available power. Since this is a battery (and in the future possibly a solar) operated node, energy must be considered a scarce resource. Batteries have a finite amount of energy, and some chemistries can suffer from internal resistance increase in long term operation. That will in fact reduce the value of peak currents available for node operation. The most used primary sources of energy in wireless nodes are alkaline and lithium batteries. Alkaline batteries can provide high peak currents at the expense of battery capacity and higher self discharge rates. On the other hand lithium batteries suited for long term operation have high capacity ratings and low self discharge rate, but care must be taken of limited peak current draw. It is important to keep in mind that we cannot use the full battery capacity rating because during discharge at some voltage level the node will stop functioning. From this we can conclude that it would be wise to use microcontroller devices that can operate down to very low voltage levels. In Fig. 2, we see a typical sensor node current consumption profile.

If we assume that we have a platform built from a microcontroller (MCU) and a RF transceiver module we can analyze how the current drain of each part affects the total current drain, and in the final total energy consumption. Most nodes work in a way that the node spends most of its time in sleep mode. RF part is disabled with only the MCU running in some power save mode. Then the MCU wakes up, processes the data and activates the RF part to transmit and returns to sleep.

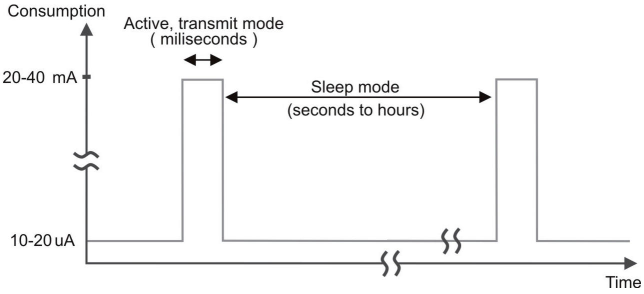

Fig.2. Typical node current consumption profile.

It is shown in [4] that the consumption of the MCU contributes more to the overall energy consumption the longer the node stays in sleep mode. If we calculate the energy spent in sleep and transmit mode [5], we get the following results (Fig. 3).

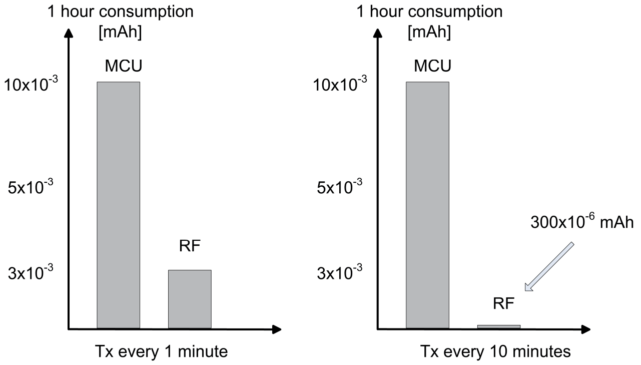

Fig. 3. MCU versus RF consumption with different transmit (wakeup) interval.

With a packet length of 144 bits, transfer speed R_b= 20 kbit/s, transmit consumption I_{tx}=25 mA and sleep consumption I_s=10 μA we first calculate packet duration

$$T_{pac} = (1/R_b) \times 144. \quad (1)$$

After that we calculate how long we are in transmit mode during one hour

$$T_{tx} = (60 \times T_{pac}) / 3600. \quad (2)$$

Multiplying that with transmit consumption

$$E_{tx} = T_{tx} \times I_{tx} \quad (3)$$

gives us 3×10^{-3} mAh. The MCU itself sleeping for one hour (T_{sleep}=1 h) spends

$$E_{sleep} = T_{sleep} \times I_s = 10\times10^{-3} \text{ mAh}. \quad (4)$$

It is now seen that reducing power consumption in sleep mode is one of the crucial tasks. This calculation is made using ATtiny2313 MCU specifications [6] and RFM22 [7] lowest output power.

3. MCU Comparison and Selection

There are many commercial solutions and platforms that fit perfectly for our application. Examples for that are RfPIC and AtmelRF series of microcontrollers with embedded RF functionality. These controllers have on disposal prebuilt software stacks that includes a lot of functionality inside them. If we want to make a simple platform we do not need all the functionality, but a simple and clean software structure. We are now left with some of the most popular general purpose microcontrollers.

Some of them are the PIC 16F and 18F series, AVR ATmega series and MSP430 series well known for its low consumption. As we did not have an MSP MCU by hand, but we had the hardware and development boards for ATmega controllers we decided to make a prototype based on them. A good measure of MCU power performance is their MIPS/W rating, which is obtained by dividing processor performance (MIPS) with power consumption (W). In this calculations supply voltage is fixed to 3.0 V to make the comparison possible. In Fig. 4 we show some popular MCU compared by their energy efficiency.

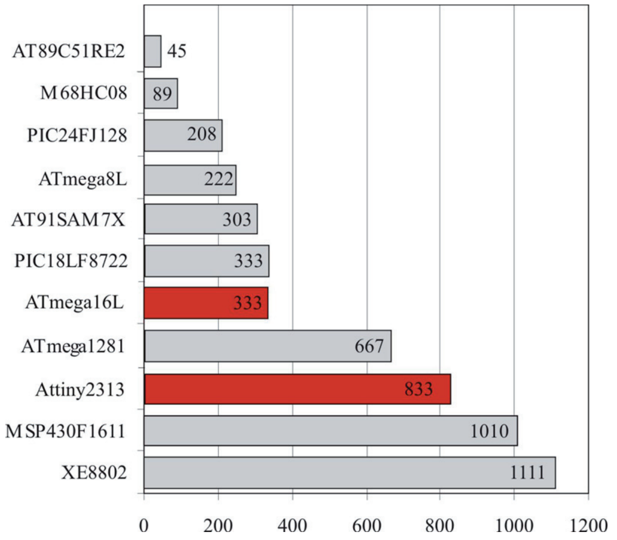

Fig. 4. MCU energy efficiency (MIPS/W, more is better).

Here only orders of magnitude are important because the instruction set affects the performance to some extent [8]. It can be seen that the ATtiny compares quite good with stronger MCUs, but its disadvantage is lack of memory space for user programmable code. Some other platforms like Mica [9] are also known to use AVR series MCU for their computing power.

4. Frequency Band and Transceiver

One of the common frequency bands for sensor networks is the 2.4 GHz band, popular for its worldwide availability. Practical realization on the 2.4 GHz is harder to achieve than on lower frequencies like 868 and 433 MHz. Also, this band is getting more crowded due to 802.11 wireless networks and bluetooth popularity. 433 MHz is one of the ISM bands and its characterized by good propagation properties in urban environment so it was our frequency of choice. The disadvantages here are larger antenna dimensions and channel bandwidth.

When we started looking for an adequate transceiver we outlined the following:

- at least 10 dBm output power,
- supported amplitude and frequency modulation,
- quick frequency change and frequency settling time (maybe future FHSS),
- data transfer rate up to around 50 kbit/s,
- supply voltage targeted between 1.6 – 3.6 volts,
- at least one low energy (sleep) mode,
- fast transition sleep-transmit-receive mode,
- optional packet handling.

In many commercial solutions from manufacturers like Texas Instruments, Semtech, Nordic Semi a less known chip from Silicon Labs was used. Silicon Labs is a very well known player in the world of analog-intensive and mixed signal IC. Si4432 is one of their EZradioPRO series available on the market for a long time, targeted at more sophisticated applications. Telit Wireless Solution is known for using this series in their modules for wireless M-Bus data exchange. The problem is that their first batches (early revisions, V2) had a silicon errata and could not fully show their capabilities. As those revisions are embedded in HopeRF RFM22 modules, and a whole lot of them was available at the time for an attractive price (due to silicon errata) we decided to go with them. We wanted to provide spare ones so students could work on them, test them and break them. The module can be seen in Fig. 5. You can note the transceiver IC, the 30.000 MHz quartz crystal and the transmit/receive switch. The module measures 16x16 mm.

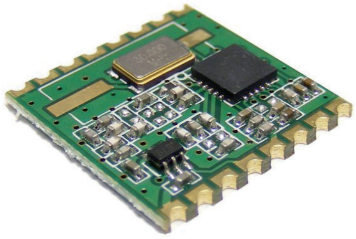

Fig. 5. The RFM22 transceiver module based on Si4432 IC.

Some of the features (as specified by the manufacturer) of this module are: supply voltage range from 1.8-3.6 V, data rates up to 128 kbit/s, +17 dBm output power, -118 dBm input sensitivity at 1.2 kbit/s, supported FSK, GFSK, OOK modulation, FHSS capability. With a cost of around 5 Euro per module this truly is a low cost solution.

Calculated energy efficiency is not very good (Fig. 6) but considering the burst mode of module operation (low duty cycle ON time) it has a smaller impact on total power consumption.

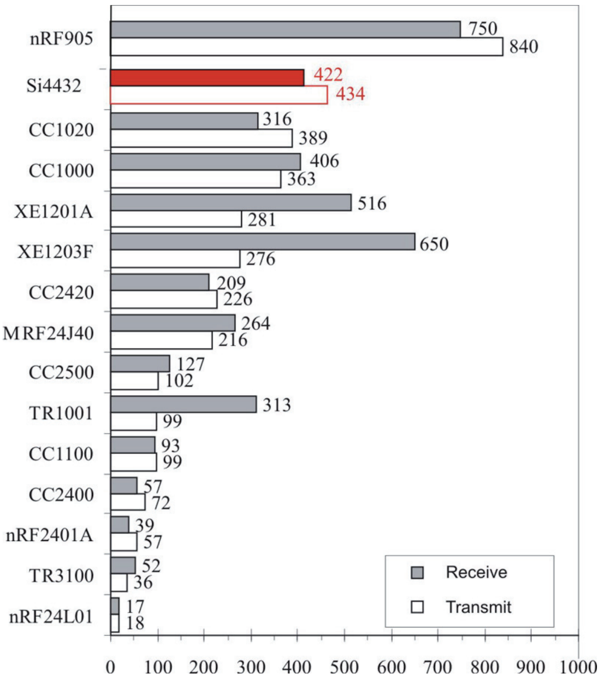

Fig. 6. Transceiver module energy consumption (nJ/bit).

Comparison is made on energy per bit ratings (nJ/bit), obtained by dividing power consumption with maximum specified transceiver data rate, with the voltage supply fixed to 3.0 V.

5. P1 Series Prototype Nodes

P1 series was designed to make measurements and hardware testing easy. Measurements were made on MCU and transceiver part. MCU programming connector is available and every signal of interest has its own test point or jumper. Breakout connectors are mounted for A/D converters and I/O ports plus DIP switches attached to I/O ports make it almost a small development board. Two programmable LED are also provided on board.

Platform P1 did not implement any power save modes, but that was corrected with P2. The node was in active state the entire time allowing us to make measurements on transceiver part. P1 series has two types of nodes: a stronger one based on ATmega16L (Fig. 7) with 16 KB flash capable of encrypted communication (AES, DES, Serpent) for future testing and a lighter version (Fig. 8) for unencrypted traffic based on ATtiny2313.

The main weakness of this node (if used in a real sensor node application) is its high cutoff voltage. At 2.7 volts microcontroller operation is halted, so we have a very narrow region of operating voltage (2.7-3.6 V). The microcontroller runs with a 7.3728 MHz quartz. In Fig. 7 the following parts are marked:

- A-reset button,
- B-ISP microcontroller programming interface,
- C-general purpose button,
- D-MCU supply current test point,
- E-serial port diagnostics (needs RS232),
- F-power supply,
- G-SPI bus test points,
- H-programmable DIP switches,
- I-breakout connector for A/D + I/O ports.

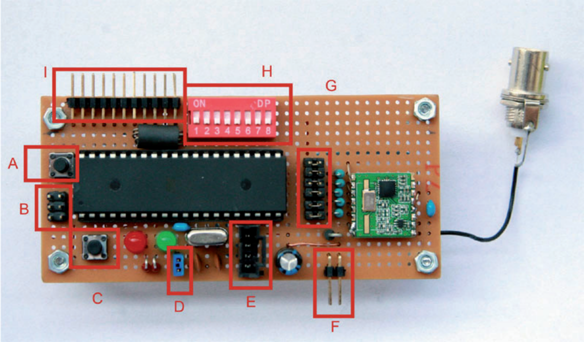

Fig. 7. P1 series node (Atmega16L) with main parts highlighted.

In Fig. 8 we see a lighter node version with 2 KB of program space. Active clock frequency is 4.000 MHz. Both types of nodes were powered from two standard alkaline AA batteries with 2000 mAh capacity rating.

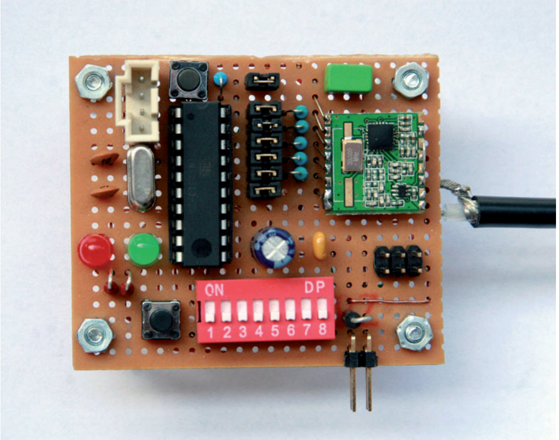

Fig. 8. P1 series node (ATtiny2313).

5.1 Packet and Payload Organization

The Si4432 operates as a time division duplexing transceiver and it alternately transmits and receives data packets. It uses a single-conversion, image reject mixer to downconvert the 2-level FSK/GFSK/OOK modulated signal to a low IF frequency. From there, the signal is converted to the digital domain using a $\Delta\sum$ ADC and passed to the built-in DSP. The DSP core provides other functionalities like auto frequency control (AFC) and data packet handler.

The structure of the packet (ready for air) with number of bytes that each field occupies is visible in Fig. 9 and it is generally the same for P1 and P2 platform, with only the user data part different. We can see that the packet structure consists of the following main parts: preamble, sync word, packet length, user data, and CRC (cyclic redundancy check) polynomials. It can also be seen why there are 144 bits in our first calculations made in sect. 2.

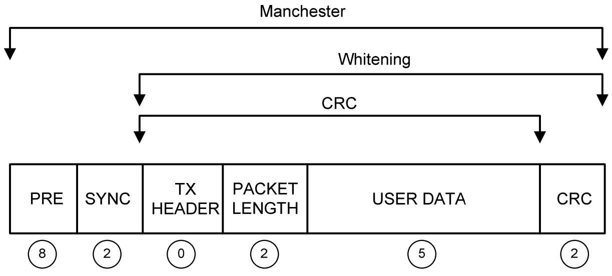

Fig. 9. Packet structure for P1 and P2 platform.

The preamble detector continuously searches for a preamble pattern, with its length configurable from 1-256 bytes. A shorter preamble might be chosen when occasional false detects are tolerable. Next the sync word is searched inside the packet. If it matches the predefined words the packet parser continues to the packet length field. The packet length field specifies how many bytes are there in the user data field. The last 2 bytes are checksums calculated using one of the predefined polynomials.

It is possible to turn on Manchester data encoding (to ensure a DC free transmission, but effectively halving the data rate) and data whitening that operate on parts of the packet as specified in Fig 9. Manchester encoding is not used, but data whitening is enabled. In this way we obtain a more uniform spectrum (no long 0s or 1s in the stream). Whitening is done with the help of the internal PN9 generator and then XORing its output with the payload.

From our tests on platform we concluded that a preamble of 64 bits is more than enough for the receiver to properly synchronize with the incoming data stream. With this length we had no false packet detections. The sync word has a value of 0x2D and 0xD4 in case the receiver picks up some possible random communication with a pattern equal to our preamble. The packet length field is set to 5 bytes, and the CRC calculation is done using the CRC-16 (IBM) polynomial over the packet length and user data fields.

NODE ID	BATT. STATE	DATA BYTE 1	DATA BYTE 2	DATA BYTE 3

Fig. 10. User data structure for P1.

The internal structure of the user data field can be observed in Fig. 10. It consists of a node ID, which is a number from 1-255 uniquely identifying each node. In this scenario ID number 0 is reserved for the master, leaving space for max. 255 nodes. Then the battery status (measured voltage) follows, and then 3 data bytes from I/O pins or A/D converter follow. Version P1 had no acknowledge return to the node from the master to confirm that the packet arrived at its destination. The master node is always active, his only assignment is to unpack the data and pass them to the PC serial port for visualization.

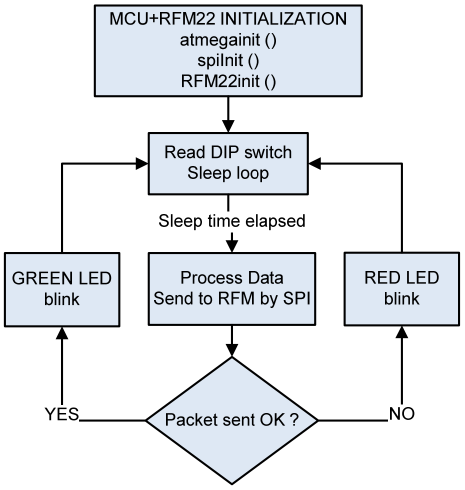

Fig 11. P1 flow diagram.

Si4432 retains register configuration even in sleep mode (600 nA current consumption according to datasheet) so it quickly gets ready to transmit when waken up. As said before a very simple packet transmit scheme is used, with no prior channel check or collision detection (Fig. 11).

5.2 Transceiver Measurements on P1

RFM22 is a module based on Si4432 IC and its output power depends greatly on board layout and output matching network design. Since we had no influence on those parameters (the module comes prebuilt with SMD parts smaller than 0805 size), some measurements were taken. Those were obtained using spectrum analyzer Anritsu MS2661C and HP 5348A microwave counter/power meter. Tab. 1 shows deviations from manufacturer specifications regarding output power. The output power is adjustable in 3 dB increments.

Measured output power [mW]	Specified output power [mW]	Current consumption [mA]
2.12	6.3	27.67
4.48	12.6	33.4
10.25	25.1	43
30.1	50	63.6
Carrier frequency : f_c=433 919 885 Hz		

Tab.1. Si4432 specified vs. obtained output power.

The mismatch between power is partially due to silicon errata, but we believe the output network has a greater impact on reduced output power.

Fig. 12 shows the unmodulated carrier spectrum with output power set to 8 dBm and Fig. 13 shows the second harmonic amplitude. These measurements were carried out to see how the harmonic termination and lowpass filter of the output network are performing.

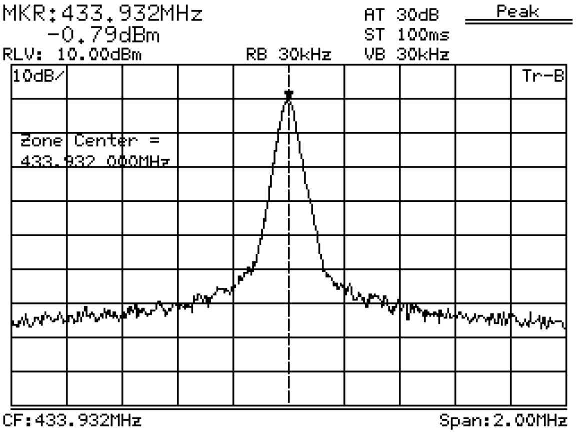

Fig. 12. Unmodulated carrier spectrum.

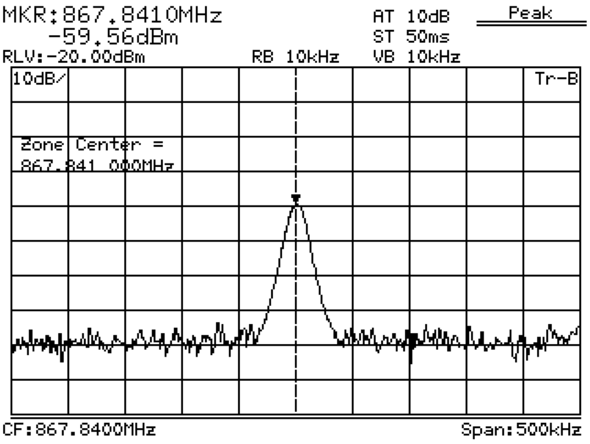

Fig 13. Second harmonic amplitude.

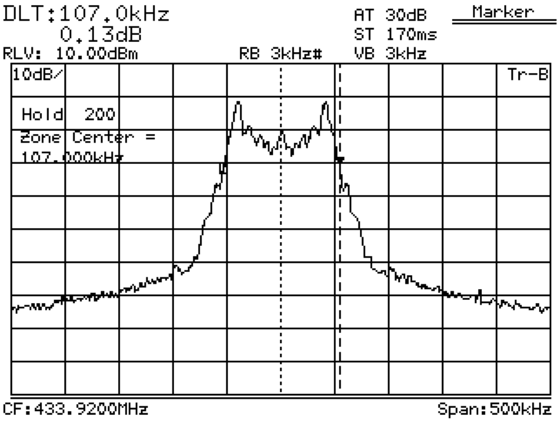

Fig. 14. GFSK modulated carrier.

Fig. 14 shows the carrier modulated with PN9 random pseudo-sequence as the modulation source. Modulation type is GFSK ($BT = 0.5$), data rate R_b= 20 kbit/s, deviation $\Delta f = 40$ kHz. These are the platform P1 default working parameters. Channel bandwidth is observed as 20 dB less than unmodulated carrier.

One version of microcontroller firmware is developed especially for RF module tests. It allows us to directly manipulate the Si4432 registers. The firmware features a simple command line interpreter connecting to the PC through the serial port. In that way we can write, read or take a snapshot of the complete 128 registers. Commands for immediate modulation type change, start and stop of transmission, operating mode (idle, sleep, ready) are implemented. In this operating mode DIP switches can directly control on air data transfer rates and frequency deviation for selected speed. Fine tuning the center frequency is also possible by setting the desired frequency in register, then writing calibration values to oscillator tune register until the set and measured frequencies match.

6. Platform P2

The P2 platform (Fig. 15) is also based on the Atmega8 microcontroller featuring reduction in size and a simple access protocol somewhat similar to Aloha. A similar approach can be seen in [10]. The main node (still from P1 series) has a modified firmware to accept a different user data structure. The power source mostly used for this node are Lithium manganese dioxide and Lithium thionyl chloride batteries, both good for long term run so we can accumulate data from nodes.

The P2 runs from its internal calibrated 8 MHz clock when active, switching to external 32.768 kHz low power oscillator in sleep mode. Integrated on the board are also 2 LEDs along with one reset and one user programmable button. The default firmware supports sleep time from 1 to 32 minutes, selectable with the DIP switch.

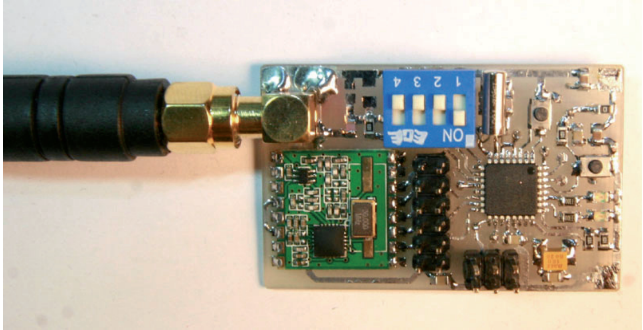

Fig.15. The P2 platform with 433 MHz antenna.

We can see the modified algorithm in Fig. 17. In the initialization routine, calibration values are written to the module together with all necessary data for a successful transmission. Once set, the only thing to take care of is properly sending the data to the RF module. After the sleep time elapses, the MCU wakes up and transfers the data to be sent to the module. After the packet is sent, the RF module is switched to reception mode and waits a preprogrammed time for ACK (acknowledge) from the master node to arrive. If no ACK is received, the MCU will generate a random delay before trying to transmit again. If after *N* times the transmission fails, the MCU will return back to sleep.

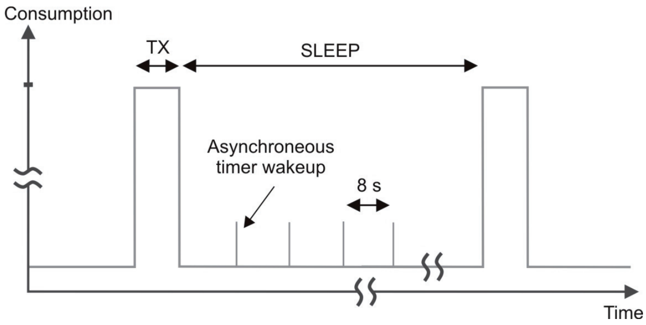

Fig. 16. Real current consumption profile of P2 (not to scale).

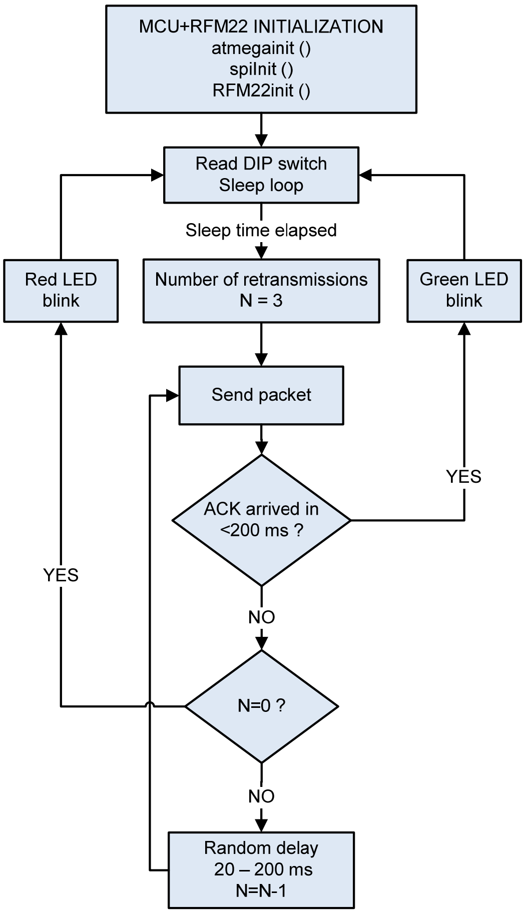

Fig 17. P2 flow diagram with ACK packet.

We stress here that our firmware development was with the help of open source tools. First schematic design was done using Eagle CADSoft Light edition, and later designs involved Altium Designer as a CAD tool. Programming was carried out in C using AVR-GCC. It is a cross compiler for the AVR platform and a port of the very popular GNU Compiler Collection. Actually, it is a whole toolchain for linux and windows platforms. On windows machines the precompiled package WinAVR was used in the form of a plug-in for the Atmel AVRStudio4 development tool. The components most used from user perspective are the C compiler itself, runtime libraries (avr-libc), simulation package (simulavr) and programming package (avrdude). All documentation and source code was first available to the public from our servers (http://marvin.kset.org/~arian/), but was later moved to GitHub for ease of maintenance. It can be found and viewed at https://github.com/ariandikovic.

Fig. 16 shows the real current consumption profile of P2. When in sleep mode, all MCU systems are halted except for the asynchronous timer that runs from the onboard 32.768 kHz quartz. This timer will overflow and generate an interrupt (with its maximum prescaler settings) every 8 seconds. So if we want to keep track of intervals longer than this we must wakeup and modify some variable that keeps the number of wakeups. We note that less current is consumed during wakeup spikes because the RF module stays in sleep.

USER DATA field is modified compared to P1 and contains more data (Fig 18, broken in two for easier understanding). Modifications were made to accommodate two-way communication and some to provide more details about node status.

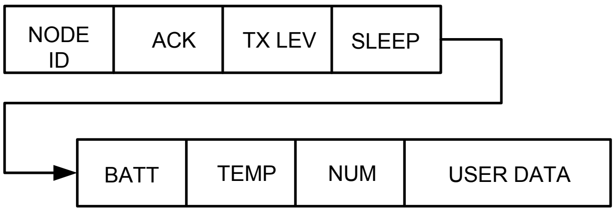

Fig. 18. USER DATA field on P2.

When the node sends data to master node, it fills up the NODE ID field with his unique ID. The ACK field is left empty, TX LEV represents the current node TX power level, SLEEP is sleep time in minutes, BATT is battery voltage level, TEMP is the integrated Si4432 temperature sensor readout, NUM is a number increasing with each sent packet (so if the master misses a number in two consecutive packets we know there is one packet missing). Then the data bytes (from A/D converters for example) follow.

The master replies by filling the NODE ID with the destination node ID and the ACK field with a predefined value. It is possible to override the node default TX power and sleep time if the master fills TX LEV and SLEEP fields with values different from zero. At the time of the next transmission new values will be used. The reply consists only of these 4 bytes.

7. Conclusion

Our conclusion regarding the RF transceiver part is that higher output power may be obtained if we had access to RFM22 module schematics to check the output network. A limiting factor here is also the silicon errata present on V2 revision of Si4432 chip. This errata also impacts the low duty cycle mode, wake up timer and the integrated supply voltage A/D converter. Thus we may say that this lowers overall system performance. Exploring further possibilities in MAC layer handling provided by the RF module could also prove to ease the requirements on MCU processing. The propagation on the selected frequency of 433 MHz was better than expected. Pending measurements for now are input sensitivity measurements and exact current consumption profile recording. Regarding the microcontroller part a MCU with better efficiency in sleep modes and lower operating voltage would be beneficial. Future work consists in implementation of crypto algorithms on the AVR platform and (maybe) porting the code to the microchip PIC18 series of microcontrollers.

We think that by developing and testing with this kind of platform we have created a quick and easy way to introduce students to wireless sensor networks, keeping the costs and complexity at its minimums.

References

[1] AKYILDIZ, I.F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E. A survey on sensor networks. *IEEE Comm. Magazine*, Aug. 2002, vol. 40, no. 8, p. 102-114.

[2] EL-HOIYDI, Y A., ARM, C., CASEIRO, R., CSERVENY, S., DECOTIGNIE, J.-D., ENZ, C., GIROUD, F., GYGER, S., LEROUX, E., MELLY, T., PEIRIS, V., PENGG, F., PFISTER, P.-D., RAEMY, N., RIBORDY, A., RUFFIEUX, D., VOLET, P. The ultra low-power WiseNET system. *IEEE Proceedings of the Design Automation & Test in Europe.* Conference DATE 2006, vol. 1, p.206.

[3] CALLAWAY, E. H. *Wireless Sensor Networks: Architectures and Protocols*. Auerbach publications, Boca Raton, 2004.

[4] JURDAK, R., RUZZELLI, A. G., O'HARE, G.M.P. Radio sleep mode optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, July 2010, vol. 9, no. 7.

[5] FARAHANI, S. *ZigBee Wireless Networks and Transceivers*. Elsevier Ltd., Burlington, 2008.

[6] ATMEL CORPORATION, ATtiny2313 datasheet, available at htt p://www.atmel.com/dyn/resources/prod_documents/doc2543.pdf

[7] HOPERF ELECTRONIC, RFM22 Datasheet, available at http://www.hoperf.com/upload/rf/RFM22.PDF

[8] KUORILEHTO, M., KOHVAKKA, M., SUHONEN, J., HÄMÄLÄINEN, P., HÄNNIKÄINEN, M., HÄMÄLÄINEN, T.D. *Ultra-Low Energy Wireless Sensor Networks in Practice - Theory, Realization and Deployment*. Chichester: Wiley, 2007.

[9] HILL, J.L., CULLER, D.E. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, Nov/Dec 2002, vol. 22, no. 6, p. 12-24.

[10] COHN, G., STUNTEBECK, E.P., PANDEY, J., OTIS, B., ABOWD, G.D., PATEl, S.N. SNUPI: Sensor nodes utilizing powerline infrastructure. In *Proceedings of UbiComp2010. The ACM International Conference on Ubiquitous Computing*, Copenhagen (Denmark), 2010, p.159-168.

About Authors

Arian DIKOVIC received B.Sc. in Electrical Engineering from the Faculty of Electrical Engineering, University of Zagreb, Croatia in 2010. He has background in industrial electronics repair, his current area of interest include embedded electronics and low power radio COM links.

Gordan SISUL received B.Sc., M. Sc. and Ph.D. in Electrical Engineering from the Faculty of Electrical Engineering, University of Zagreb, Croatia in 1996, 2000 and 2004, respectively. He is currently employed as a researcher at the same Faculty. His academic interests include wireless communications, signal processing applications in communications, modulation techniques and coding.

Borivoj MODLIC received B.Sc., M. Sc. and Ph.D. in Electrical Engineering from the Faculty of Electrical Engineering, University of Zagreb, Croatia, in 1972, 1974 and 1976, respectively. He started his professional career as an assistant professor, Department of Radio-frequency Engineering (presently Department of Wireless Communications), Faculty of Electrical Engineering and Computing, University of Zagreb where he has worked ever since. He is the coauthor of six university textbooks and editor of the Engineering Handbook. His research interests are: signal processing in communications, especially modulation methods, wireless access systems, electromagnetic compatibility and electromagnetic field impacts on human health and the related health hazards estimation.