Serdica J. Computing **12** (2018), No 1–2, 23–45

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

HETEROGENEOUS IoT PLATFORM FOR DEVICE MANAGEMENT AND ENVIRONMENTAL SENSOR DATA GATHERING

Svetozar Ilchev, Rumen Andreev, Zlatoliliya Ilcheva

ABSTRACT. In this paper, we propose a new IoT platform which supports the use of different wired and radio connections to manage devices and gather environmental sensor data. We discuss the pilot implementation of our new platform which deals with the smart control of air-conditioning devices in shopping malls. The building blocks of the platform – two types of electronic controllers of our own design – are described in detail. The Internet connectivity of the platform enables the use of an Internet-based control center for storing historical data, performing statistical analyses, making automated corrections in the device control and providing possibilities for remote control by human operators. The end goal of the platform development is to achieve cost savings and increase the level of comfort of customers and employees.

1. Introduction. During the last decade, the Internet has developed to the point that almost any device, anywhere, can be connected so that it can

ACM Computing Classification System (1998): B.4.0, C.2.0.
Key words: IoT platform, heterogeneous network architecture, smart devices, remote control, wireless sensor networks, IoT network protocols.

be monitored and controlled remotely—either in an automated fashion or manually by a human user. Many electrical home appliances such as refrigerators, washing machines, boilers, etc. are sold in package with cloud-based services which aim at providing added value—e. g., analyzing the behavior of the end customer and controlling the appliance according to that behavior. For example, a smart refrigerator may order food and drinks automatically according to the usual preferences of its owner, a boiler may start heating the water several hours before its owner comes home from work, etc. In addition, when venues for data communication are present, several appliances may interact with each other and function as a group – e. g., lighting, heating and ventilation systems – which creates a potential for cost savings and the overall reliability of devices is increased due to the possibility for automated or remote fault identification via the network connection.

As there are many old-style appliances still in use that do not have any network connectivity, the question arises if they can be fitted with suitable control hardware and become part of the IoT. In addition, custom sensors at custom places may be needed for measurement of various environmental parameters such as temperature, humidity, atmospheric pressure, presence of people, etc. Ideally, all of them communicate with an intelligent service – provided either by a local device or a remote cloud server – which stores and analyzes the gathered data, makes decisions and controls the appliances.

In accordance to this concept, we propose an IoT platform consisting of electronic controllers, communication protocols and cloud-based services. The platform enables the automated management and monitoring of appliances with the purposes of achieving cost savings and raising the level of comfort for the end users.

For the initial development, we cooperated with a company from the air-conditioning sector and our pilot project was to implement smart control of large air-conditioning devices typically used in shopping malls. The main purpose was achieving cost savings while maintaining the same or better level of comfort for the clients and the workers in the shopping malls. An additional benefit was the better coordination of the support teams as the platform indicated in real-time possible problems with the air-conditioning devices.

In the next sections of this article, we will describe the buildings blocks of proposed platform and we will summarize the practical experience we gathered from using the platform to control air-conditioning devices in shopping malls.

2. Related work. The design and implementation of IoT platforms has been a popular research topic in recent years. In [1, 2], the authors discuss the transmission of sensor data and control messages and utilize for this purpose a secure and reliable communication protocol capable of transmitting small data packets. In [3], the design, structure and communication protocols of wireless cooperative networks are discussed. The authors present fixed relaying techniques like the so called decode-and-forward technique and dynamic relaying techniques. In [4], the authors discuss wireless sensors and ways for their efficient localization when classic media access control and network layer protocols are employed. The topic of [5] is the routing in wireless sensor networks which is implemented by means of fuzzy algorithms. These algorithms change the routing process depending on current traffic and network conditions. The building of smart clusters of IoT devices is proposed in [6] with the main purpose of making the network communication more efficient. The authors of [7] consider cases characterized by limitations in the power supply energy which imposes the need for optimized packet transmission in a wireless sensor network. In [8], a broader perspective of IoT platforms is presented with thoughts of offering citywide cloud-based services by means of smart city hubs.

Some thoughts on IoT platforms and services from the standpoint of their users and their role in corporate business processes are presented in [9] with an emphasis on the economic values created by IoT. In [10], an IoT architecture is proposed with the purpose of using it as part of the development and operation of smart cities. The architecture follows a top-down approach which, among other things, presents a high-level classification of IoT platforms. The authors of [11] present an extensible architecture for transmitting sensor and actuator data. The network is built of modules containing hardware from different manufacturers and aims at achieving plug and play functionality. In [12], several architectures for IoT communication are compared to one another and a new reference IoT architecture is proposed by the authors. A new IoT platform for providing micro services to smart cities is proposed in [13]. The authors discuss important design principles and propose suitable architecture layers, components and APIs for the presented platform. In [14], the authors propose a n architecture for IoT communication based on the principle of REST which is often used for provision of various Internet services. The authors of [15] discuss the importance of extensibility and adaptability in IoT environments. They give some specific guidelines for the design of IoT platforms and propose a reference IoT platform. Among the discussed topics are also network scaling, service discovery and network security.

3. Overview of the proposed IoT Platform based on the pilot implementation for control and monitoring of air-conditioning devices.

3.1. High-level goals and provided services. The pilot use case for our new IoT platform involves the control and monitoring of air-conditioning devices in shopping malls. The main goals from an end-user perspective are:

- lowering the electricity consumption,
- increasing the level of comfort in the rooms and
- facilitating the work of the support teams through automation.

Through the use of an Internet-based control center, the platform offers the following services depending on the application:

- Centralized management and observation of all connected devices.
- Storage of the device status and the sensor data into a database.
- Information system for the preparation of reports and statistics and the support of real-time decision making.
- Intelligent autonomous control of the connected devices with the purpose of lowering the electricity and maintenance costs.
- Sensor connectivity, e. g., for measurement of temperature, humidity, luminosity, etc.

- Transmission of telemetry data unrelated to its main use—e. g., communal services data (automated billing of electric and water meters, etc.), meteorological data, etc.

3.2. Platform design and architecture. The proposed IoT platform is composed of a wireless radio network functioning on 868 MHz or alternatively on 433 MHz, a GPRS/3G/4G connection to the Internet, which may be replaced by a LAN or Wi-Fi connection, and a control center for data gathering, intelligent analysis and management (Fig. 1). If need be, any radio link can be replaced with a wired communication line (e. g., RS-485, CAN, Ethernet, etc.) without changing the communication protocols. Such a heterogeneous communication environment has a better potential for adaptation to the specific needs of the application.

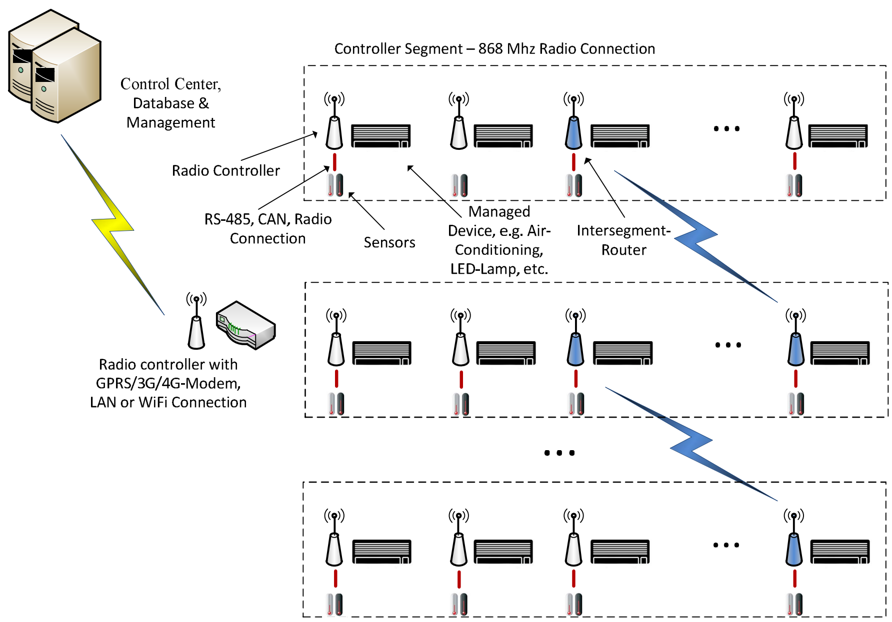

Fig. 1. Platform architecture

The radio network consists of controller segments, which are groups of radio controllers covering a maximum distance of 10 km when there is a direct line-of-sight or 1 km when there is no direct line-of-sight. The speed of data transmission depends on the distance—small distances allow higher transmission speeds and vice versa. The design allows for flexible grouping of the controllers into segments, so that the network can be adapted to the specific application scenario.

The radio communication is carried out via a custom communication protocol designed by us and implemented into each radio controller. The radio controller hardware and software is designed by our research group. Every radio controller can play the role of an end node or an intersegment router.

An end node is tasked with the control of the air-conditioning device connected to it and the observation and transmission of the sensor data coming from any connected sensors. The node performs preliminary sensor data analysis, makes decisions with regard to the control of the air conditioning device and communicates actively through the IoT platform with the control center. There are multiple sensors (e. g., SHT21, BMP180) measuring temperature, humidity, pressure, etc. that are connected by wire (RS-485 or CAN) or wirelessly (LORA) to each end node. They inform the end node of important parameters of the surrounding environment which are used in the decision making process and transmitted to the control center for further analysis and storage.

An intersegment router may or may not have an attached air-conditioning device or sensors. If it has an attached device and sensors, it carries out all the functions of an end node in addition to its network routing tasks. Network routing tasks encompass packet retransmissions in case of large networks where the radio controller that has Internet connectivity is not accessible from all network nodes.

In the network, there is one (or more than one if there is a need for redundancy) radio controller that has Internet connectivity via GPRS/3G/4G connection or LAN/Wi-Fi connection. In small networks, this controller may be accessible by every other node in the network, so that we have only one network segment and we do not need any intersegment routers. In larger networks, we perform a subdivision of the network into segments and designate intersegment routers.

The wireless radio communication within the IoT platform is implemented using the LORA technology on the free 868/433 MHz radio frequency bands. The same type of connectivity is used for the sensors connected to the end nodes. If a wireless connection is not suitable for a particular network branch, there is the option of replacing it with a wired connection—based on RS-485, CAN or Ethernet. The communication protocol remains unchanged and does not depend on the actual physical technology used for the connection. The protocol features AES-256 encryption, handshaking and automatic retransmission in case of lost packets. It also has routing capabilities employed by the intersegment routers and the end nodes in cases of large networks.

For the GPRS connection, we use a GPRS-modem (SIM900/908), which relies on the mobile network to gain Internet access. For LAN connectivity, we have designed an extension which uses WizNet W5500. Through the Internet connection, the IoT network connects to the control center and transmits and obtains information. The radio controller that implements the Internet connection may also play other roles such as controlling an air-conditioning device.

The control center is responsible for the storage and analysis of the gathered sensor data, making decisions and submitting remote control requests to the IoT network as well as for the implementation of a suitable interface for user access (GUI, API, etc.). It is composed of one or more servers depending on the number of users and the quantity of incoming data.

As an additional function, the IoT network may also be controlled by an end user directly through one of the radio controllers which provides access to the network through a USB port. In this way, a personal computer or a notebook may be connected locally to a suitable network node and control can be exercised by the local staff even if there is no Internet connectivity.

3.3. Technical overview. From a technical viewpoint, the IoT network consists of two types of controller—radio controllers with extensible functions and sensors with wired or radio connectivity. The radio controllers are implemented using 32-bit RISC-microcontrollers of the series STM32Fx. They have onboard 3.3V, 5V, 10V and galvanically isolated 5V power supplies. The radio connectivity is implemented through a Semtech SX1276 LORA radio module on 868 MHz or 433 MHz. Among the provided communication interfaces are: RS232, RS485, UART, I2C, SPI, USB, isolated CAN, 4 analog inputs, 2 analog outputs. In this way, the radio controller can communicate with various electronic devices such as GPRS-Modems, sensors or computers. There is support for a real-time clock and EEPROM for configuration options and up to 5 relays switching up to 10A loads at 250VAC/30VDC for actuator control.

The sensors are smaller and less powerful microcontrollers based on the series STM32F0 or STM32L0. The temperature and humidity is measured by sensor chips such as MCP9701 and SHT21. Other sensor chips can be added or removed depending on user needs.

The control center that the IoT network connects uses open-source server software. The operating system is a Linux flavor (Ubuntu Server). As DBMS we use MySQL/MariaDB at the moment. Firebird is also supported as an option. The web server for user access is Apache in combination with PHP.

4. Design and implementation of the radio controller. The radio controller is designed in an extendable way, so that periphery suitable for the specific application can be added on. It has integrated and extended peripheral blocks which house multiple interfaces for wired and radio communication through our developed communication protocol. It has suitable connector terminals for the wired connections, an antenna for 868 MHz or 433 MHz communication and the necessary communication software to control and read the sensors and actuators. The extended peripheral block implements any optional connection venues such as the GPRS/3G/4G connection, the USB connection to a PC or the Ethernet connection. The communication protocol developed by us offers a unified way of transmitting and receiving data both through local and remote connections independently of the physical implementation of the connection. It supports addressing, routing, encryption, integrity checking via checksums and retransmission of lost data packets. The protocol is designed in such a way as to allow the usage of different network topologies such as full-mesh, star and extended star topologies which may be needed depending on the application scenarios. Through this controller, the implementation of an intelligent device control and sensor data gathering is made possible for various industrial, commercial and home environments.

The radio controller consists of the function blocks shown in Fig. 2.

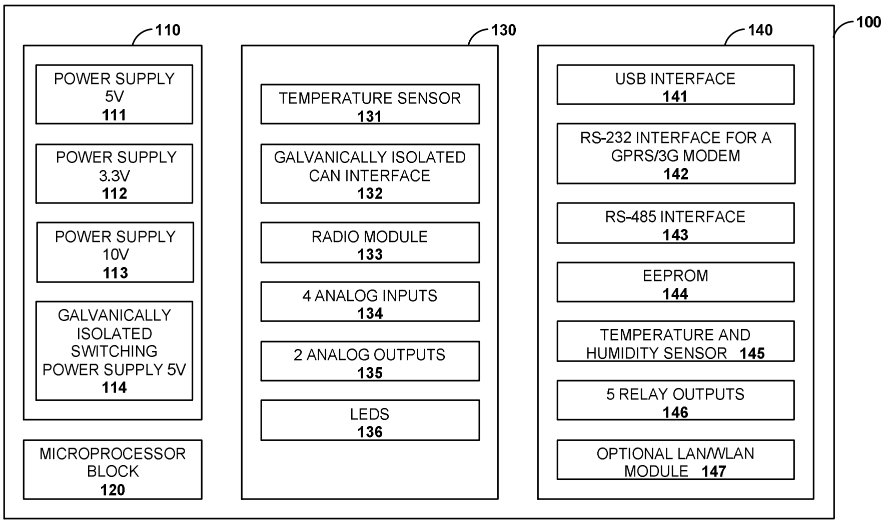

Fig. 2. Functional blocks of the radio controller

In Fig. 2, the radio controller (100) consists of a power supply block with an input voltage of 9V–24V (110), a microprocessor block (120), an integrated peripheral block (130) and an extended peripheral block (140).

The power supply block (110) is composed of several switching power supplies implemented on the controller printed-circuit board (PCB) that deliver power supply voltages needed by the other functional blocks. Blocks (111) and (112) convert the input voltage to voltage levels with values of respectively 5 volts and 3.3 volts. These voltage levels are standard levels for the operation of many integrated circuits. Each voltage level is generated by a special integrated circuit (IC) in combination with an inductor and several capacitors. The main purpose is achieving high efficiency and low levels of heat emissions from the power supply blocks. The 5V level of block (111) is used mainly by the two peripheral blocks (130) and (140) to power the analog inputs (134) and outputs (135), the LEDs (136) and the relay outputs (146).

The 3.3V level of block (112) is used by the microprocessor block (120) and the peripheral blocks (130) and (140). It powers the temperature sensor (131), the radio module (133), the USB interface (141), the RS-232 interface for the GPRS/3G modem (142), the RS-485 interface (143), the EEPROM (144) and the temperature and humidity sensor (145) and the optional LAN/WLAN module (147).

The power supply block (113) is a switching power supply with an output level of 10 volts. It is a standard voltage level in the industry used to measure or generate analog signals in blocks (134) and (135). It is implemented by an integrated circuit which uses as input voltage either the 5V or the 3.3V delivered by blocks (111) or (112).

Block (114) implements a galvanically isolated power supply voltage of 5 volts using as input voltage either 5V or 3.3V generated respectively by blocks (111) or (112). It is used to power the galvanically isolated CAN interface—block (132). It is often expensive but necessary to fully decouple the electric connection between two communicating devices, so that both the power lines and the communication signals between them do not have a direct electrical connection. This eliminates parasitic currents and reduces electromagnetic interferences.

The output voltages of all power supply functional blocks are accessible to the extended peripheral block (140) of the controller and they are also made accessible to external devices via terminals on the front panel of the controller. In addition, the power supply block (110) of the radio controller has a fuse for overcurrent protection and it is also protected from reversing the input voltage polarity and abnormal increases in the working temperatures of the power supply functional blocks.

The microprocessor block (120) consists of the microprocessor STM32Fx, a crystal oscillator with a frequency of 8–16 MHz for the generation of main working frequency, a crystal oscillator with a frequency of 32–40 KHz and a lithium CR2032 battery for the real-time clock, a battery, a JTAG interface for programming and debugging and an optional voltage monitoring integrated circuit for intelligent reset in case of a temporary power supply voltage loss. The microprocessor is a 32-bit RISC microprocessor with an ARM Cortex-Mx core and multiple integrated functions such as UART, I2C, SPI, USB and CAN interfaces, timers, analog-to-digital (ADC) and digital-to-analog (DAC) converters, real-time measurement, etc. The main frequency is can be up to 180 MHz, the integrated RAM memory can be up to 256 KB and the integrated flash memory can be up to 2 MB, which makes them a good choice for applications requiring communication and remote monitoring and control.

The microprocessors of the series STM32Fx can be used in conjunction with real-time operating systems (RTOS) such as FreeRTOS and they offer good possibilities for interactive debugging via the JTAG interface. There is a continuously developed and well-supported version of the popular open-source C/C++ compiler GCC and several free integrated development environments (IDE) such as TrueStudio recently acquired by ST or EmBitz. Thus, these microprocessors are a good choice for the central heart of our radio controllers.

The integrated peripheral block (130) consists of several peripheral sub-blocks. The temperature sensor (131) is an analog sensor with an output voltage between 0V and 3.3V measured by the ADC converter of the microprocessor.

The galvanically isolated CAN interface (132) is used for long-distance wired communication with other controllers. The low-level communication protocol of the CAN connection (bxCAN) is supported on a hardware level by the microprocessor. The CAN interface consists of an integrated circuit (IC) for galvanic isolation of the communication signals, a suitable physical-layer CAN driver IC, an optional terminating resistor, a noise filter and protection from overvoltages on the CAN line. The power supply is taken from block (114) as mentioned above. The communication distance is up to 1 km, the data transfer speed is up to 50 Kb/s.

The radio module (133) consists of an integrated circuit that controls the communication and generates the physical-layer signals, an active 20 MHz oscillator of high accuracy, a signal amplifier and the necessary filter components. IN the current design, we use the LORA technology, which allows communication at distances of up to 10–15 km in case of direct line-of-sight. The message size can be varied in a flexible manner. There is a trade-off between a reliable signal coding, the size of the message and the transmission speed. We use an omni-directional antenna with impedance of 50 Ohm attached to the controller front-panel. The connection of the radio module to the microprocessor is implemented through one of the SPI interfaces and in addition several digital interrupt lines are used—e. g., to notify the microprocessor of a received packet.

The controller also has 4 analog inputs—block (134). They can measure analog values in the interval 0V–5V or 0V–10V depending on the configuration and the application needs. The inputs are protected from overvoltage. Each one of them is connected to an analog input pin of the microprocessor through an operational amplifier which scales the intervals 0V–5V or 0V–10V to an interval 0V–3.3V that can be measured directly by the microprocessor. Block (135) consists of two analog outputs that can be scaled to the same voltage ranges of 0V–5V or 0V–10V depending on the controller configuration. Each output is connected to an analog output pin of the microprocessor through an operational amplifier which does the corresponding scaling. On each output, there is also a low-pass analog filter.

The LEDs—block (136) are used for status indication on the front panel of the controller.

The peripheral block (130) provides communication and measuring capabilities and contains sensitive analog components—temperature sensor (131), analog inputs (134) and outputs (135). They are placed in close proximity to the microprocessor block (120).

Besides the integrated peripheral block (130) for a specific application, there is often need for additional periphery, which becomes part of the extended peripheral block (140). For our target applications, we have designed and tested several functional sub-blocks.

The USB interface (141) implements two USB ports. One of them makes use of a special integrated circuit which emulates a serial communication port through the USB connection. The interaction with the microprocessor is realized via one of the UART interfaces. RTS/CTS signals for controlling the data flow are also supported. The advantage of this solution is the existence of verified working drivers for the USB connection and the provision of unique VID and PID identification numbers.

The second USB port makes use of the USB protocol support integrated into the microprocessor. It protected from overvoltage and polarity reversal. The advantage of the second USB port is its flexibility and versatility—e. g., if convenient, the controller may present itself as a HID device such as a mouse or a keyboard. In this case, drivers for the specific application must be created, which leads to a more complex microprocessor firmware and possibly specialized drivers for each supported operating system on the PC or notebook. In addition, the developer has to take care of the provision of VID and PID numbers, which for small device series may be inefficient.

The RS-232 interface for a GPRS/3G modem (142) is also implemented by means of a special integrated circuit which generates the voltage levels necessary for the RS-232 communication. The connection to the microprocessor is through a UART interface. RTS/CST signals for data flow control are supported. The RS-232 interface is included in the extended peripheral block to enable the connection to a GPRS/3G modem with or without a GPS function. This modem is used for Internet-access by the whole radio network. Some models provide also GPS functionality, which can be used by the radio controller for exact localization and real-time clock setting. The control of the modem is carried out through AT commands in ASCII format transmitted through the RS-232 interface. The modem needs an active SIM card allowing data transfer to the Internet. As a backup communication channel, sending and receiving SMS messages is supported. As an alternative way of connecting to the Internet, an optional LAN/WLAN module (147) can be used.

The RS-485 interface (143) is used for wired connection to sensors or other microcontrollers which measure temperature, humidity, pressure or control actuators such as heating elements, solenoids, etc. As is the case with the CAN interface, the signal is transmitted over two wires in a differential manner. The maximum transmission distance is 1.2 km. In contrast to CAN, RS-485 defines only the lowest physical layer of the data transmission. The unified communication protocol has to take care of all high-level details of the communication—e. g., framing, addressing, etc. The RS-485 interface on the controller includes the necessary terminating resistors, analog filters and protections from overvoltage or polarity reversal. The communication to the microprocessor is implemented by means of one of the UART interfaces. There is also an LED indication on the controller front panel.

The EEPROM (144) is a non-volatile memory used for storing configuration settings and other data related to the functioning of the radio controller. It endures more writing cycles than the flash memory integrated into the microprocessor and offers byte wise access. Supported EEPROM sizes are in the range from 1 Kbit to 1024 Kbits. The communication with the microprocessor is over an I2C interface.

The temperature and humidity sensor (144) is a specialized integrated circuit with a digital I2C interface for connection to the microprocessor. Its temperature measurement is more accurate in comparison with the analog temperature sensor (131).

The five relay outputs (146) may be used to control DC or AC loads and support currents up to 16A. A filtered 5V power supply is used for the energizing of the relay coils. For additional safety and reduction of the electromagnetic interference, each relay is connected to the microprocessor through an optocoupler. There is an LED indication on the controller front panel for each relay showing when it is activated.

The extended peripheral block (140) is essential for adapting the controller functionality to the specific needs of the application. Many of the inputs/outputs of the microprocessor block (120) are directly accessible, which gives complete freedom during the design of the hardware and software. This allow us to design the radio controller (100) in two stages—the first stage includes designing the power supply block (110), the microprocessor block (120) and the integrated peripheral block (130) and the second stage includes designing the extended peripheral block (140). In case of an application change, the first three blocks remain the same and only the extended peripheral block is modified as needed.

The physical implementation of the radio controller (100) consists of two printed-circuit boards (PCB) attached to each other via two multi-pin connectors. The first board is a base board which contains the power supply block (110), the microprocessor block (120) and the integrated peripheral block (130).

The second board is an extending board housing the extended peripheral block (140). The boards are situated directly next to each other and they are mounted on a plastic PCB enclosure designed for DIN-rail mounting.

Each board may be fully populated or some functional blocks may be omitted if they are not needed in a specific application. The front panel of the controller can be manufactured as a PCB or from another suitable material of thickness up to 1.6 mm. The front panel provides access to connection terminals for the power supplies and the communication interfaces, the antenna for the radio connection and some LED status indications. Fig. 3 and Fig. 4 illustrate the physical appearance of the radio controller.

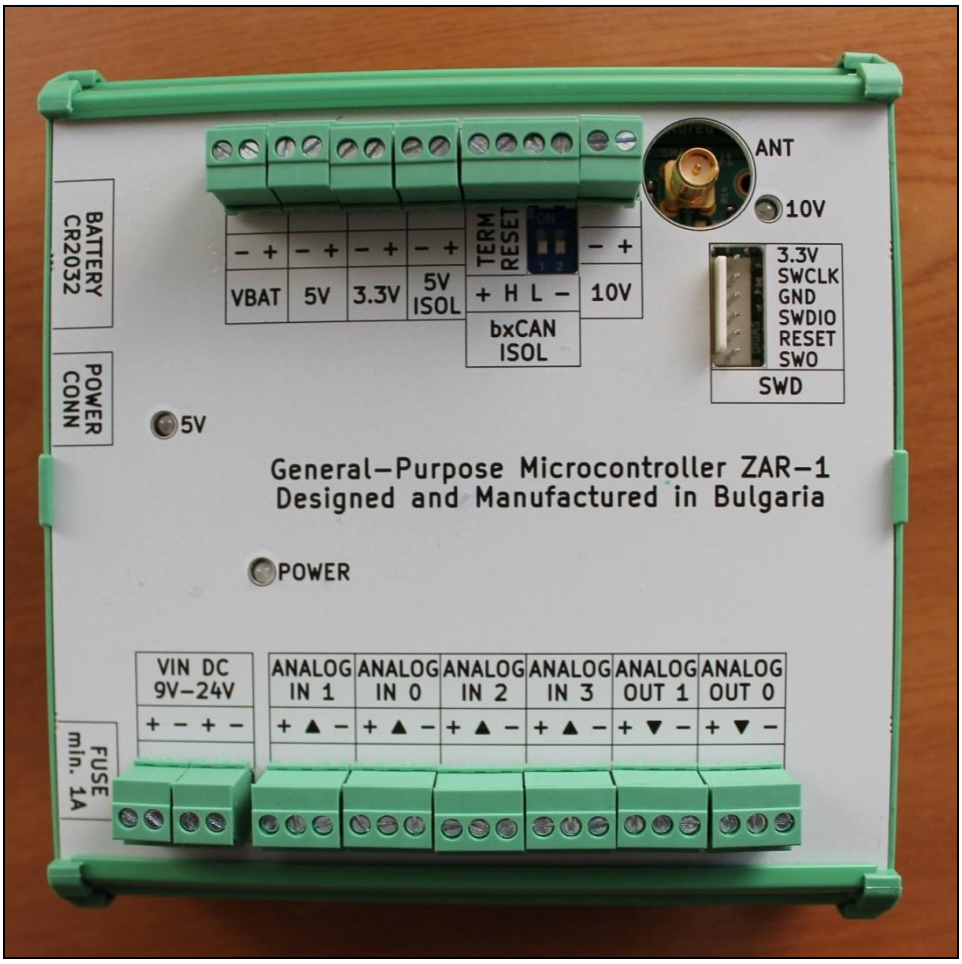

Fig. 3. Fully populated base board of the radio controller mounted in the DIN-rail enclosure together with the front panel and connection terminals

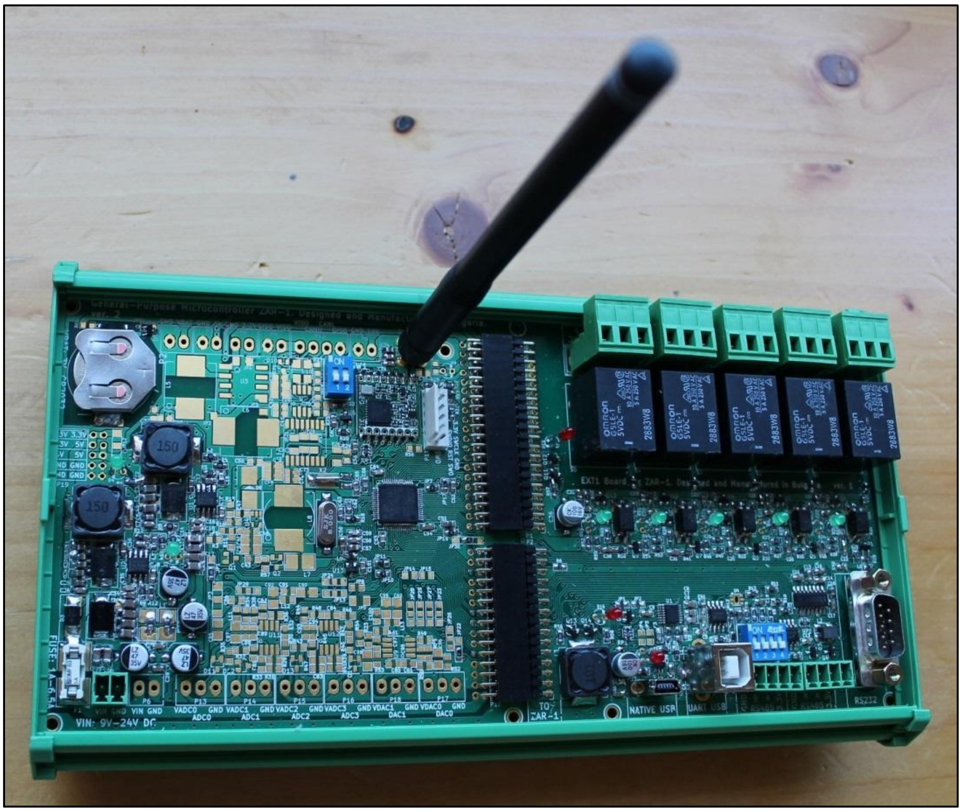

Fig. 4. Partially populated base board together with an extending board mounted in the DIN-rail enclosure

5. Design and implementation of the sensors. The sensors are designed to measure the temperature and humidity of the environment around them. They may communicate to the main radio controllers either by wire (RS-485) or by radio (LORA). Currently, we have two versions of the sensors—one larger version with radio connectivity and the possibility of adding additional sensors, and one smaller version, which is water resistant, has only wired connectivity and measures only temperature and humidity. The smaller water-resistant version is the one that is most suitable for our application scenario for air-conditioning control. This version consists of the functional blocks presented in Fig. 5.

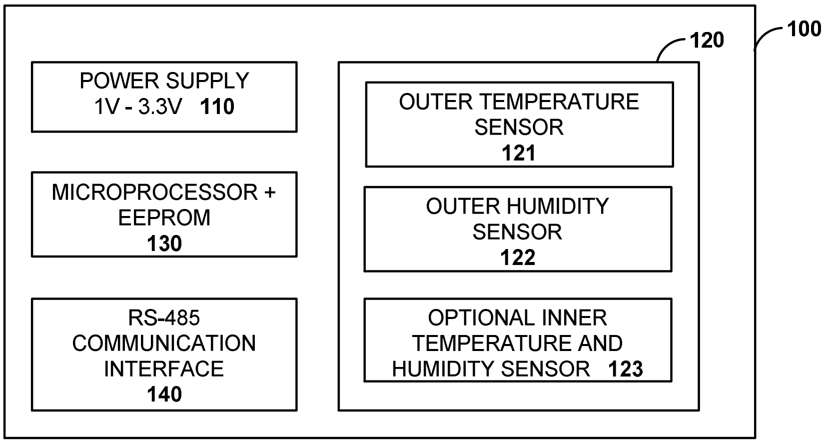

Fig. 5. Functional blocks of a sensor controller

In Fig. 5, the sensor controller (100) consists of a power supply block with an input voltage of 1V–3.3V (110), a sensor block (120), a microprocessor and EEPROM block (130), and an RS-485 communication interface block (140).

The power supply block (110) consists of a switching power supply that raises the input voltage through the so called “boost” topology. For the purpose, a small inductor and several ceramic capacitors are used. The input voltage can vary in the interval 1V–3.3V and the output voltage is 3.3V. Power sources can be a wall-mount adapter, batteries, solar panels, etc. The output voltage powers the other functional blocks. The power supply block has a fuse against overcurrent conditions and it is protected from overvoltage, polarity reversal and abnormal increases in the working temperatures. It also has an LED indication on the printed-circuit board.

The sensor block (120) consists of an outer temperature sensor (121), an outer humidity sensor (122) and an optional inner temperature and humidity sensor (123). The outer temperature sensor (121) is an NTC thermistor situated in a rust-proof metal case. It is mounted in a circular opening on the sensor controller enclosure. The connection to the printed-circuit board (PCB) is implemented through a standard terminal that is wired to one of the analog inputs of the microprocessor.

The outer humidity sensor (122) is also mounted in a circular opening on the sensor controller enclosure and fixed by a suitable adhesive, e. g., epoxy or other. The connection to the PCB is implemented through the same standard terminal. The communication to the microprocessor involves the use of several additional elements such as an LMC555 timer integrated circuit (IC). The humidity percentage is encoded through a frequency signal connected to one of the digital inputs of the microprocessor.

The optional inner temperature and humidity sensor (123) is a specialized IC soldered on the sensor controller PCB. It has high accuracy and communicates to the microprocessor via an I2C interface. It can be used for calibration of the outer sensors when the sensor enclosure is not sealed. When the enclosure is sealed, the inner sensor measures the working temperature and humidity inside the enclosure which is may be of use when the sensor controller is placed in an uncontrolled outdoor environment.

The microprocessor block (130) contains the microprocessor STM32F0 or STM32L0, an oscillator with a frequency of 8 MHz–16 MHz from which the main working frequency is generated and a JTAG programming and debugging interface. The STM32F0 and STM32L0 are 32-bit RISC microprocessors with an ARM Cortex-M0 core and multiple integrated functions such as UART, I2C and SPI interfaces, timers and analog-to-digital converters (ADC). The main working frequency is between 32 MHz and 48 MHz depending on the specific model. The size of the RAM and FLASH is enough for performing the data acquisition, preliminary data processing and data transmission. As mentioned in the previous section, these microprocessors have good software support in terms of RTOS, compilers, IDEs, debugging, etc. The EEPROM is used for storing configuration options and if need be some of the sensor data if it can be transmitted at the moment.

The RS-485 communication interface (140) was discussed in more details in the previous section. In this case, it is used to connect the sensor to the extended peripheral block of a corresponding radio controller by wire and transmit sensor data.

The sensor controller (100) is designed to fit in a small dust-proof and water-proof IP68 plastic enclosure with dimensions of about 12cm x 5cm. The outer sensors (121) and (122) are fixed on the enclosure and measure the temperature and humidity of the environment. Through the unified communication protocol, sensor data can be transmitted through the radio network and passed to the control center over the Internet connection. The sensor controller can be placed in dry or wet indoor rooms as well as in outdoor spaces to gather, process and transmit temperature and humidity data at regular intervals.

Fig. 6 shows the physical appearance of the printed-circuit boards of the sensor controller.

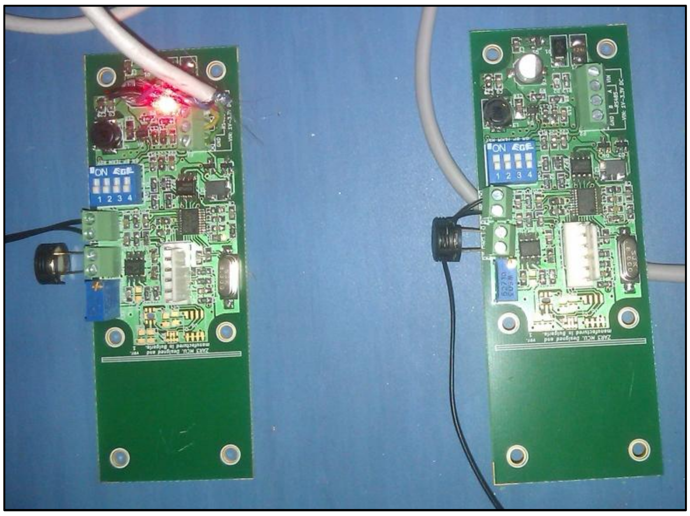

Fig. 6. Sensor controller printed-circuit boards

6. Experimental results. We are using the IoT radio network in a pilot implementation to control some air-conditioning devices in two shopping malls. The network coverage of the LORA radio technology within the commercial buildings has proved to be excellent. Data can be communicated point-to-point throughout the whole building which has allowed us to turn off the routing in the communication protocol and to use a simple star topology. We have configured the data packets to have a size of 26 bytes. This size allows most of the control commands and sensor data transfers to fit within a single packet. Here is some necessary overhead for some of the commands, which is mainly due to the requirements of a minimum size of 16 bytes for the encrypted payload within a data packet. This overhead is relatively small and does not lead to a significant delay in the communication or an increased number of lost and retransmitted data packets. The unified communication protocol works very well. The features of the protocol – the encryption, integrity checking via checksums and retransmission of lost data packets – are working as they should. The GPRS communication between the IoT radio network and the control center has been relatively reliable with a few cases of connectivity loss for short durations. The radio controller that implements the GPRS connectivity tracks such cases when they arise and as soon as the Internet connectivity is resumed, the connection to the control center is reestablished.

The security of the radio network is on a good level. The usual network sniffing and attack methods can only detect that there is some communication going on. No data can be retrieved and commands cannot be forged.

The radio network transmits to the Internet control center relatively small data packets which are processed in a very fast and efficient manner and do not impose a significant load on the server side. This solution has good scaling potential and should remain responsive even when multiple radio networks transmit data to the control center at the same time.

7. Conclusion. The IoT platform proposed by us proves to be working well in commercial environments. We employ both wired (RS-485) and radio communication (LORA) handled in a unified way above the physical layer of the network connection. This heterogeneous communication environment adapts well to the particular needs of the application and makes possible the reliable control of the devices and regular sensor data gathering. The Internet-based control center provides possibility for smart sensor data analysis and versatile remote control of the devices connected to the IoT platform. The controllers of our own design can be modified if the application requirements change which ensures that other devices can be integrated into the existing radio network providing for better environmental control and in result—cost savings and increased user comfort.

Acknowledgement. This research was supported by the National Science Fund at the Ministry of Education and Science of Bulgaria under the research project *Conceptual Modeling and Simulation of Internet of Things Ecosystems (KoMEIN).*

REFERENCES

[1] JAN M., P. NANDA, M. USMAN, X. HE. PAWN: A Payload-based mutual Authentication scheme for Wireless Sensor Networks. *Concurrency and Computation: Practice and Experience*, **29** (2017), No 17. DOI: 10.1002/cpe.3986

[2] FAROOQ M., M. WASEEM, A. KHAIRI, S. MAZHAR. A Critical Analysis on the Security Concerns of Internet of Things (IoT). *International Journal of Computer Applications*, **111** (2015), No 7, 1–6.

[3] MOHAMMED A., B. DAI, B. HUANG, M. AZHAR, G. XU, P. QIN, S. YU. A survey and tutorial of wireless relay network protocols based on network coding. *Journal of Network and Computer Applications*, **36** (2013), No 2, 593–610.

[4] PERILLO M., W. HEINZELMAN. Wireless Sensor Network Protocols. In: Fundamental Algorithms and Protocols for Wireless and Mobile Networks. CRC Hall, 2005.

[5] MINHAS M., S. GOPALAKRISHNAN, V. LEUNG. Fuzzy Algorithms for Maximum Lifetime Routing in Wireless Sensor Networks. In: IEEE Global Telecommunications Conference (GLOBECOM), New Orleans, USA, 2008, 1–6.

[6] HAO J., Q. CHEN, H. HUO, J. ZHAO. Energy efficient clustering algorithm for data gathering in wireless sensor networks. *Journal of Networks*, **6** (2011), No 3, 490–497.

[7] Kurt S., H. Yildiz, M. Yigit, B. Tavli, V. Gungor. Packet Size Optimization in Wireless Sensor Networks for Smart Grid Applications. *IEEE Transactions on Industrial Electronics*, **64** (2017), No 3, 2392–2401.

[8] Lea R., M. Blackstock. City Hub: A Cloud-Based IoT Platform for Smart Cities. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2014, 799–804. DOI: 10.1109/CloudCom.2014.65

[9] Wortmann F., K. Flüchter. Internet of Things. Technology and Value Added. *Business & Information Systems Engineering*, **57** (2015), No 3, 221–224. DOI: 10.1007/s12599-015-0383-3

[10] Ganchev I., J. Zhanlin, M. O'Droma. A Generic IoT Architecture for Smart Cities. In: 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014) Proceedings, Limerick, Ireland, 2014, 196–199. DOI: 10.1049/cp.2014.0684

[11] Mikhaylov K., J. Petäjäjärvi, M. Mäkeläinen, A. Paatelma, T. Hänninen. Extensible Modular Wireless Sensor and Actuator Network and IoT Platform with Plug&Play Module Connection. In: Proceedings of the 14th International Conference on Information Processing in Sensor Networks, Seattle, Washington, USA, 2015, 386–387. DOI: 10.1145/2737095.2742559

[12] Guth J., U. Breitenbücher, M. Falkenthal, F. Leymann, L. Reinfurt. Comparison of IoT platform architectures: A field study based on a reference architecture. In: Cloudification of the Internet of Things (CIoT), Paris, France, 2016, 1–6. DOI: 10.1109/CIOT.2016.7872918

[13] Krylovskiy A., M. Jahn, E. Patti. Designing a Smart City Internet of Things Platform with Microservice Architecture. In: 3rd International

Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 2015, 25–30. DOI: 10.1109/FiCloud.2015.55

[14] Zhang X., Z. Wen, Y. Wu, J. Zou. The implementation and application of the internet of things platform based on the REST architecture. In: Proceedings of the International Conference on Business Management and Electronic Information (BMEI), Guangzhou, China, 2011, Volume 2, 43–45. DOI: 10.1109/ICBMEI.2011.5917838

[15] Andersen M., G. Fierro, D. E. Culler. Enabling synergy in IoT: Platform to service and beyond, Journal of Network and Computer Applications, 81 (2017), 96-110. DOI: 10.1016/j.jnca.2016.10.017.

Svetozar Ilchev
e-mail: `svetozar@ilchev.net`
Rumen Andreev
e-mail: `rumen@isdip.bas.bg`
Zlatoliliya Ilcheva
e-mail: `zlat@iccs.isdip.bas.bg`
Institute of Computer and Communication Systems
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 2
1113 Sofia, Bulgaria

Received May 15, 2018
Final Accepted July 15, 2018