Proposal of Wireless Sensors networks via libelium technology

Alžbeta Kanáliková – Peter Holečko – Mária Franeková – Emília Bubeníková

Abstract

Wireless sensor networks (WSN) are spatially distributed autonomous sensors used to monitor physical or environmental quantities. The article deals with problems of WSN and description of different parts of the WSN, parameters of network and technologies to create a WSN on the base of the Libelium technology. A practical implementation of a small WSN is illustrated also set-up within the campus of University of Žilina. In the practical part the article describes the initial proposal of WSN and an application of sensors data processing.

Keywords:

wireless sensors networks, sensor, Libelium, Waspmote, data of sensors, REST architecture, web technology, web application.

ACM Computing Classification System:

Wireless access networks, Network management, Programmable networks, Web application, RESTful web services.

Introduction

Wireless sensor networks (WSN) are structures composed of several types of nodes (hundreds or thousands). The nodes of a WSN are dispersed and include dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind speed and direction, pressure, etc.

Sensor networks are to be utilized in various spheres of human activity. Examples of the use WSN are the following [1]:

- Environmental monitoring (transport, environment, safety).
- Protection of infrastructure (distribution networks, water and gas supply).
- Military purposes (classification and addressing of an objective, monitoring of the battlefield).
- Health (monitoring of human physiological data, patient follow-on remotely, care for the elderly).
- Parking systems.
- Monitoring and control of buildings.
- Control of production processes in industry (manufacturing automation, tracking chemical elements, protection against accidents).

– Logistics (cargo delivery tracking, inventory control).
  In the Figure 1 we can see the basic principle of WSN.

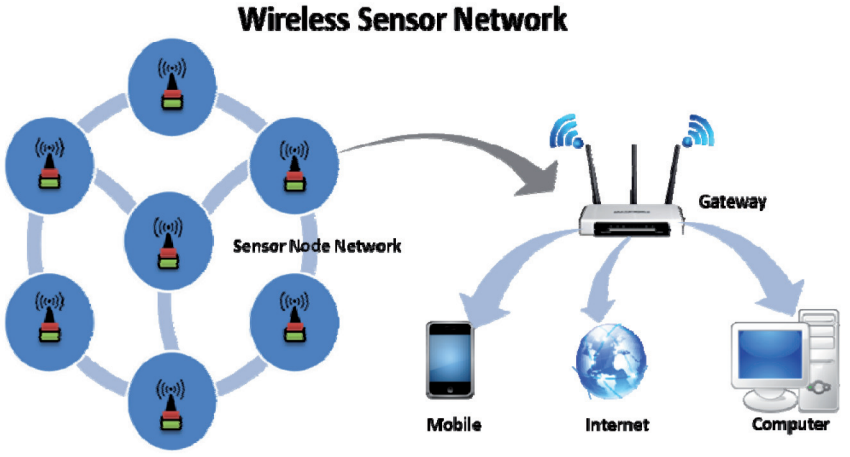

Figure 1: *Wireless Sensor Network – basic principle [1]*

The authors focused on a description of the benefits of technology Libelium which is available on Department of Control and Information System (DCIS), Faculty of Electrical Engineering, University of Žilina from the project ITMS: 26210120021 "Modernization of research infrastructure in the fields of electrical engineering, electrical materials, information and communication technology". In the article the authors describe a proposal of WSN network and data application, which were created at DCIS with cooperation of student works (bachelor and diploma works), e. g. [2], [3], [4], under leading of pedagogical staff from DCIS.

1 Description of Libelium technology for sensor networks implementation

The Spanish company Libelium provides a powerful, modular, easy to program open-source sensor platform for the Internet of Things enabling system integrators to implement reliable Smart Cities and M2M solutions [5]. The company offers products divided into four categories:

– **Waspmote** – is a modular board, which is the basis for wireless sensor node. The WASPMOTE sensors are complemented with communication modules according to the destination node. WASPMOTE provides maximum configurability and flexibility in the design and implementation.
– **Plug & Sense** – Waspmote Plug & Sense! Line – is a solution which allows to easily deploying wireless sensor networks. This solution has easy and scalable way ensuring and minimum maintenance costs. The platform consists of a robust waterproof enclosure with specific external sockets to connect sensors, a solar panel, antenna and even an USB cable in order to reprogram the node.

- **Meshlium** – Meshlium is a Linux router which works as a Gateway of the Waspmote Sensor Networks, cloud or internet for nodes of sensor network. It cancontain 6 different radio interfaces: WiFi 2.4GHz, WiFi 5GHz, 3G/GPRS, Bluetooth, ZigBee (868/915/2400 MHz), XBee.
- **Cooking Hacks** – selection boards, gadgets and components for the creation and modification of existing equipment based on the popular electronic platform (Raspberry Pi, Arduino, Waspmote).

1.1 Meshlium

Meshlium can be operated as an access point to the internet / intranet, as a scanner of devices and / or as a collector of data from nodes. Meshlium parses data and can process data with the following options [5]:

1. storing in the internal database (capacity of 8.16 or 32GB),
2. storing in an external database,
3. sending further to the Internet via 3G / GPRS,
4. sending further to the Internet via Ethernet or WiFi.

It is possible to upload own programs to the Meshlium router that are running on pre-installed runtime, for example in C++, Java, Ruby, PHP, Perl and C languages. Accessing the system of router is made through FTP and SSH protocols. The access to the basic configuration settings of the router is handled via *Meshlium System Manager* web interface.

1.2 Plug & Sense

The Plug & Sense products enable to establish wireless sensor networks with large horizontal scalability and low operating cost [5].

The principal characteristics of Plug & Sense products are the following:

- Hard packaging with IP 65.
- Quick connect / replacement of sensors into the prepared slots.
- The possibility of power from the built-in battery or solar panel.
- Available wireless communication protocols: ZigBee, 802.15.4, WiFi, 868MHz, 900MHz, 3G / GPRS and Bluetooth Low Energy
- Over the Air Programming (OTAP) – Wireless programming of multiple nodes simultaneously transmitting one broadcast.
- Graphical Programming Interface (Waspmote IDE).
- Contactless resetting magnet.
- SIM slot for 3G and GPRS.

Plug & Sense contains different kinds of boards such as:

- **Smart Cities** – scanning the urban environment to measure air parameters (temperature, humidity, dust etc.), the intensity of light and noise, strained state of building structures, stacked collection of waste containers.

- **Smart Parking** – module for detection of free parking spaces. The module is located under the ground in a protective cover. Tri-axial magnetic field sensor detects presence of a vehicle.
- **Smart Environment** – module monitoring air quality. For example, the presence of a gas content of CO_2, O_2, O_3, ethanol, toluene etc.

2 *Proposal of WSN*

The proposal of sensors network in the environment of University of Žilina was created on the base of Libelium products. The topology architecture of the proposed sensor network is a tree with a logical hierarchical structure of nodes. The nodes communicate using XBee protocols, especially XBee-PRO 868, XBee-PRO 802.15.4, or XBee-PRO ZB. The map of proposal wireless sensor network is illustrated at Figure 2 [2], [3].

Figure 2: *Proposal of WSN network on campus of University of Žilina [2], [3]*

2.1 XBee protocols

The ZigBee standard builds on the established IEEE 802.15.4 standard for packet-based wireless transport. A ZigBee wireless network includes three types of node [6]:

- **Co-ordinator** – selects the network number, is the first node to be started join other nodes with the network, co-ordinator has a routing role (is able to relay messages from one node to another) and is also able to send/receive data.
- **Router** – routes data and to send/receive data, also allows other nodes to join the network. Router cannot go to sleep mode in order not to cut a subordinate end device from network.
- **End node** – is a node which is only capable of sending and receiving data (it has no routing capability). A network may have many end devices. An example structure of nodes of WSN network with XBee protocol is at Figure 3 [7].

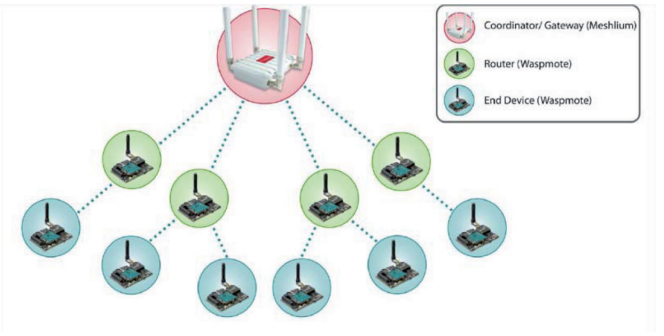

Figure 3: *Structure of nodes of WSN with XBee protocol [7]*

2.2 Programming sensor nodes

The communication parameters in the individual nodes need to be adapted for specific protocols. The programming of nodes (Waspmote) is possible in two ways by using the USB or OTAP (Over The Air Programming). The structure of the codes is divided into 2 basic functions*: setup()* and *loop()*. The function *setup()* is the first part of the code, which is only run once when the code is initialized (or Waspmote is reset). In this part it is recommended to include the initialization of the modules which are going to be used, as well as the part of the code which is only important when the Waspmote is started. The *loop()*function runs continuously, forming an infinite loop. The goal of this function is to measure, send the data and save energy by entering a low energy consumption state [8]. A basic segmentation of program is shown in Figure 4.

```
/ 1. Include Libraries
// 2. Definitions
// 3. Global variables declaration
      void setup()
      {
      // 4. Modules initialization
      }
      void loop()
      {
      // 5. Measure
      // 6. Send data
      // 7. Sleep Waspmote
      }
```

Figure 4: *Basic segmentation program of sensors nodes [8]*

The *loop()* function at the proposal WSN starts sensors boards, reads data, sends frame and shuts down sensors. At the end of the *loop()* function is a module listed into sleep and thus saving energy.

To set the individual parameters and communication protocols need to use the library:

#include <WaspXBee868.h> // XBee 868 MHz
#include <WaspXBeeDM.h> // XBee DigiMesh
#include <WaspXBee802.h> // XBee 802.15.4

The following example code implements the Smart Cities Sensor boards with XBee ZigBee PRO protocol.

In function *setup()*:

```
Packet=(packetXBee*) calloc(1,sizeof(packetXBee));// memory allocation for packet
  ..
xbeeZB.ON(); //initialize communications
  ..
xbeeZB.setDestinationParams(packet, MAC_ADDRESS, "....");// XBee packet setting parameters (address of the recipient, payload)
  ..
xbeeZB.sendXBee(packet);//send first message – registration by the recipient (network ZigBee)
```

In function *loop()*:

```
SensorCities.ON();// turning on the sensor board
SensorCities.setSensorMode(SENS_ON, SENS_CITIES_TEMPERATURE);// switch on transmitters on board
  ..
frame.createFrame(ASCII, nodeID); // create frame
frame.setFrameSize(ZIGBEE, UNICAST_64B, ENABLED, DISABLED);// set the frame size
  ..
// add sensors data to frame
frame.addSensor(SENSOR_BAT, PWR.getBatteryLevel());
frame.addSensor(SENSOR_TCA,SensorCities.readValue(SENS_CITIES_TEMPERATURE));
  ..
xbeeZB.sendXBee(packet);//send packet
```

2.3 Measurement of quantities

Measurement of quantities is performed after initialization of the node after *setup()* function. It begins to perform within the *loop()* function where there is a measurement of selected variables outside. It activates the Waspmote sensor board (in the proposed board WSN Environment and Smart Cities Smart Parking) and executes the test measurement. After performing a practice of measurement full measurement is performed. The measured data can then be calibrated and converted to other variables or inserted into the data frame and sent [8].

2.4 Data frames

To create a data frame the *WaspFrame.h* library is used, which should be attached to the source code. The framework is created using the *createframe()* method, for example: *frame.createFrame()*. The frames consist of two parts – a header – contains routing data and additional information of data part and data part – contains data. The proposed sensor network uses the ASCII frame format. The ASCII frame structure is shown in Figure 5 [9].

HEADER										PAYLOAD						
<=>	Frame Type	Num Fields	#	Serial ID	#	Waspmote ID	#	Sequence	#	Sensor_1	#	Sensor_2	#	...	Sensor_n	#

Figure 5: *Structure of the ASCII data frame [9]*

The framework starts with <=> and the individual data are separated by #. The data part of the frame contains data from various sensors which are also separated by the # character.

2.5 Configuration of data end

Data collected from nodes can be stored directly in the reception routers – local database, or externally in a dedicated database system within a broader network, or the internet. The Meshlium routers provide its own internal database and parser framework that decomposes their content and data stored in data tables. The proposed WSN data is routed to master nodes and synchronized to the central MySQL database.

3 Proposal of data application and its testing

The proposed WSN network disposes of a developed data interface, respectively a web application that is built on the REST architecture (REpresentational State Transfer). The REST architecture uses the HTTP protocol and the resources are accessible using HTTP methods (GET, POST, PUT, PATCH, DELETE) [10].

The data is stored in an external MySQL database. The application has developed a method of access to that database. The database tables contain except the sensors data also additional supporting tables. The database contains the following tables:

- **Sensorparser –** data from sensors.
- **Nodes** – information of nodes of the proposed WSN.
- **Users** – user data, user accounts etc.
- **Boards** – types of Libelium boards.
- **Protocols** – contain the types of various communication modules (protocols) for Waspmote.
- **Types** – various types of Libelium equipment.
- **Sensors** – various types of sensors.

Example of generated data from sensors – database table *Sensorparser* is illustrated in Table 1:

Table 1: Table database Sensorparser – data from sensors [2]

id wasp	id secret	frame type	frame number	sensor	value	time stamp	sync	raw	parser type
wf_codie	382552159	128	1	BAT	19	13.11.2014 15:13	0	NULL	1
wf_codie	382552159	128	1	TCA	24,19	13.11.2014 15:13	0	NULL	1
wf_codie	382552159	128	1	HUMA	79,4	13.11.2014 15:13	0	NULL	1
wf_codie	382552159	128	1	LUM	91,887	13.11.2014 15:13	0	NULL	1
wf_codie	382552159	128	2	BAT	19	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	2	TCA	24,19	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	2	HUMA	79,4	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	2	LUM	91,887	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	3	BAT	19	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	3	TCA	24,19	13.11.2014 15:14	0	NULL	1
wf_codie	382552159	128	3	HUMA	79,4	13.11.2014 15:14	0	NULL	1

3.1 Implementation of data application

The basic building block of the server platform is *NodeJS* with the module *ExpressJS* (framework is one of the frameworks used for creating web applications). The designed web application will be except browsing and data visualization also used to manage sensor networks. In addition to these technologies Asynchronous JavaScript and *JSON* so-called *AJAJ* is also used. AngularJS (Angular JS is open MVC framework written in JavaScript), library C3.js (allows files to websites reusable components for data visualization) and the Google Maps API (libraries and web service allowing to upload maps to Web applications) also were used in implementation of web application.

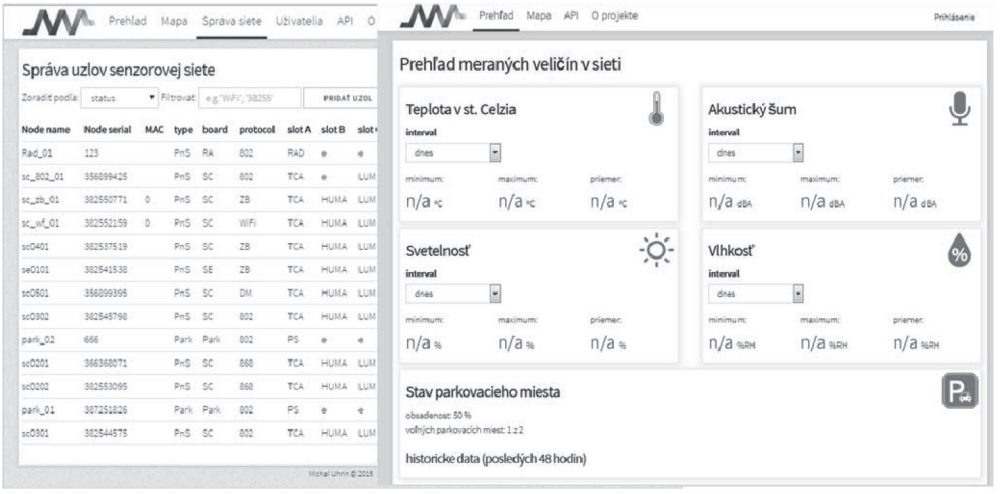

Figure 6: *Screen web application of proposal WSN [2]*

The Figure 6 shows the outputs – web application of WSN screen – sensor data (right) and the settings of WSN nodes (left). In realized applications via WSN were kept sensor

data from University of Žilina campus from several parameters of environment, e. g. temperature of atmosphere, luminosity, damp of climate, etc. Actual values of monitored parameters it is possible to observe via internet browser.

Conclusion

The Libelium products are currently the leading products on the market and implemented by many foreign universities. The results of article summarized the works which were realized at DCIS, Faculty of Electrical Engineering, University of Žilina within implementation of WSN in university campus via Libelium technology, which was bought within project ITMS: 26210120021 "Modernization of research infrastructure in the fields of electrical engineering, electrical materials, information and communication technology". Authors describe the opportunities of technology Libelium. The practical part the initial proposal of WSN within the university campus describes and creation of a proposal of a data application used for data processing from the selected sensors. In the future we are planning to extend the applications and to implement a higher number nodes and sensors and to expand the applications with extending their functionality (for example in data visualizations and data calibration). Other applications could use in the future the data collected in the database. The data will be analyzed further through data mining. The implementation of sensor networks within the university can get real data sources also for the IBM Intelligent Operations for Transportation platform. The operation of real WSN will enable students and employees to get real experience and practical knowledge that could be offered a re-utilized in future projects.

References

[1] https://www.techopedia.com/definition/25651/wireless-sensor-network-wsn [online 2. 9. 2015]
[2] Bc. Michal Uhrín: Proposal of application at data evaluation from sensors network, diploma work 2015, supervisor: Ing. A.Kanáliková, PhD.
[3] Bc. Jakub Kubalík: Smart campus University of Zilina, diploma work 2015, supervisor: Prof. Ing. Aleš Janota, PhD.
[4] Patrik Paukov: Conceptual design applications, sensor networks based on technologies LibeliumSmart campus University of Zilina, bachelor work 2015, supervisor: Prof. Ing. Aleš Janota, PhD.
[5] http://www.libelium.com/ [online 4. 9. 2015]
[6] http://www.nxp.com/documents/user_manual/JN-UG-3048.pdf [online 5. 9. 2015]
[7] ZigBee Networking Guide, Libelium documentation: http://www.libelium.com/development/waspmote/documentation/?cat=networki ng [online 7. 9. 2015]
[8] Programming Guide, Libelium documentation: http://www.libelium.com/development/waspmote/documentation/programming-guide/ [online 7. 9. 2015]

[9] Data Frame Guide, Libelium documentation: http://www.libelium.com/development/waspmote/documentation/data-frame-guide/ [online 7. 9. 2015]
[10] ELKSTEIN DR. M.: LEARN REST: A TUTORIAL: http://rest.elkstein.org/ [online 7. 9. 2015]

Ing. Alžbeta Kanálikova, PhD.
alzbeta.kanalikova@fel.uniza.sk

Ing. Peter Holečko, PhD.
peter.holečko@fel.uniza.sk

Prof. Ing. Mária Franeková, PhD.
maria.franekova@fel.uniza.sk

Ing. Emília Bubeníkova, PhD.
emilia.bubenikova@fel.uniza.sk

University of Žilina, Faculty of Electrical Engineering, Department of Control and Information Systems
Slovak Republic