Modular Smart House System Based on a Wireless Sensor Network

Preliminary Communication

Davor Bogdanović
Josip Juraj Strossmayer University of Osijek,
Faculty of Electrical Engineering, Department of Communications
KnezaTrpimira 2b, Osijek, Croatia
davor.bogdanovic2@gmail.com

Tea Kvolik
Josip Juraj Strossmayer University of Osijek,
Faculty of Electrical Engineering, Department of Communications
KnezaTrpimira 2b, Osijek, Croatia
tea.alagic@etfos.hr

Goran Horvat
Josip Juraj Strossmayer University of Osijek,
Faculty of Electrical Engineering, Department of Communications
KnezaTrpimira 2b, Osijek, Croatia
goran.horvat@etfos.hr

Abstract *– Smart House is an automated and controlled system, which enables adjustment of living environment according to user demands. Advancements of wireless sensor network (WSN) technology give us an opportunity to improve, simplify and ensure a cost-effective smart home system. The paper describes the system composed of the BeagleBoard-xM, an XBeePro S2B coordinator, an XBee smart plug and a sensor, a mobile and a web application. The test solution was made in the laboratory environment. Compared to other existing solutions, the proposed system has benefits such as low-power consumption, cost effectiveness, modularity, module placement independence of power source, etc. In future, the system can be supplemented by power consumption regulation, speech and face recognition software, etc.*

Keywords *– embedded system, Smart House, wireless sensor network, ZigBee, XBee.*

1. INTRODUCTION

When electrical energy and information technology were introduced into households at the end of the 20th century, the use of electrical household devices has increased rapidly. It created the need to automatize and control the living environment, and this is where the term "Smart House" originates from. A "Smart House" is commonly defined as an "electronic networking technology to integrate devices and appliances so that the entire home can be monitored and controlled centrally as a single machine" [1]. Even though the idea of a smart house existed before, system implementation was possible only after appropriate technical solutions were offered. With the appearance of a microcontroller, cheap processor production and personal computers, the smart home system has become cheaper and more accessible. Today, it is possible to have a low-price, fast and simple home control system (HCS) [2].

The primary goal of the HCS was to ensure a more comfortable and easier way of life. In a short period of time, the primary goal of the HCS was expanded to optimize energy consumption and increase environmental safety. Today, the smart home system unifies devices managing lighting, windows, doors, heating, air conditioning, security and surveillance, various multimedia devices, etc.

In order to achieve this level of complex automation, the HCS should be modular and robust. When upgrading or downgrading the system (depending on the end-user needs) with various types of sensors, switches, valves, motors, etc., the HCS has to be stable, adaptable and responsive.

To achieve a stable, adaptable and responsive system, future work should be focused on expanding hardware support and running stability tests for various elements and operational conditions. Considering HCS software,

extra functionalities should be implemented to expand and simplify user's interaction with the system. It can be accomplished with the implementation of speech and face recognition software, probabilistic behavior recognition methods [3] and creating a simple, efficient and enjoyable user interface.

This paper describes the HCS with the WSN based on the ZigBee network. The HCS uses low-cost modules for easy installation and communication between terminal elements and their base station (called the Coordinator), a Linux-based server for communication between user, web application and Google cloud messaging (GCM) for real-time bidirectional communication between the user's mobile phone and the system.

The system is modular so it can be modified to satisfy user's needs with other WSN elements (sensors, switches, valves, motors, etc.) based on the ZigBee protocol that can be associated with the system for different types of regulation. It allows users to create various subsystems within the HCS.

In the next section, we describe the implementation of the WSN in the HCS system. Section 3 describes HCS system architecture implementation in a laboratory environment with its possible usage and advantages, while Section 4 gives a conclusion indicating future improvements of the system.

2. HOME CONTROL SYSTEM OVERVIEW

The home control system is based on low-cost, low-power devices that represent system end devices for building a more complex overall system. The HCS system is logically divided into the following main components: a WSN, a Coordinator, a virtual private server (VPS), a GCM server, a web application and a mobile application. System architecture is shown in Figure 1.

The WSN is a system's backbone for gathering telemetry data and controlling the environment. It represents the cluster of small-size sensor and actuator devices. Devices within the WSN are unaware of other devices within the cluster. Each device within the cluster is autonomous and capable of bidirectional communication with the coordinator. Sensors within the cluster gather environment telemetry data and send them periodically to the coordinator. Sensors are low-power embedded devices working only for a small amount of time until the data is not sent to the coordinator. After successful data transfer to the coordinator, the device enters hibernation mode which consumes a small amount of power. In a reverse communication scenario, the sensor device that is in hibernation mode is woken up by the RF part of the module as soon as the packet is received from the coordinator. Such device operation minimizes power consumption. Sensors are typically battery-powered devices wirelessly connected to the coordinator which can (depending on the system precision requirement) operate for several years on a single battery power source.

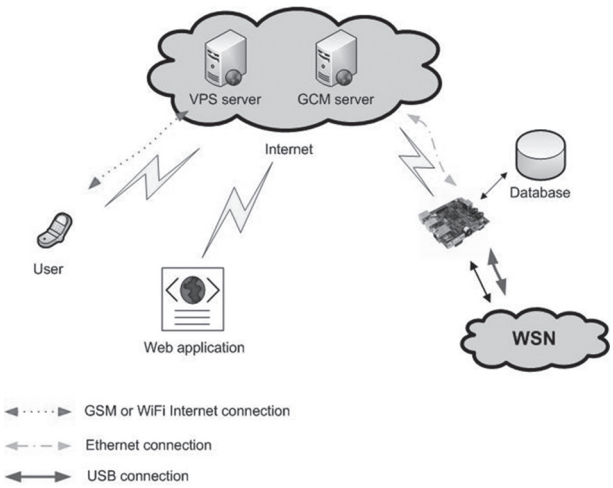

Fig. 1. Home control system architecture

The coordinator provides and maintains a wireless connection to WSN devices. It is used to periodically collect sensors data, monitor the status of devices within the WSN, issue a timely warning notice about critical situations, collect sensor data and regulate actuators by a request from higher level applications. However, one of the most important roles of the coordinator is to serve as a bidirectional bridge between WSN devices and the VPS server (which provides the data to end devices) and also to provide temporary data storage for gathered sensor information and changes of actuator devices. The coordinator is typically based on a microprocessor capable of running a general-purpose operating system and should be equipped with some standard communication interfaces like Ethernet, UART, I2C, etc. The coordinator is connected to a continuous power supply so it does not have low-power consumption constraints like WSN devices.

The purpose of the VPS is to enable constant access to the system over the Internet. The VPS should be able to run a general-purpose operating system (like Linux) and be capable of storing information into the database. It provides permanent storage for sensor and actuator data gathered from the coordinator. A two-way communication between the end-user mobile phone and the web application with WSN devices is achieved over the VPS. One of the most important roles of the VPS is to authenticate users, provide a secure bidirectional communication protocol, respond to alarm events and forward them to the GCM server.

To overcome the downsides of the HTTP protocol, the GCM server is added to the system. Since the communication between the end user and the VPS is done over the Internet and the HTTP is commonly used as a web application, it is also convenient to use it for a mobile application. The application HTTP protocol is a request/response protocol which has a downside when the VPS tries to send data to the end device without any former data request from the end user. The main purpose of the GCM server is to pro-

vide real-time information to the end device which is a crucial part of the alarm reporting mechanism. The transport layer protocol is an underlying protocol for the HTTP, so some layer of security for a successful data exchange between the end user and the system is provided automatically. To ensure that data is transferred successfully between the VPS and the end device, some extra mechanisms are implemented in the GCM server (like Quality-of-Service (QoS)). Also, an authentication mechanism is implemented in the GCM server so it can be associated only to authorized devices that are reported by the VPS server.

In the WSN, a web application can be used to present measured parameters but also to allow users to control devices in the network. Data shown in web applications can be obtained from the database and/or directly from devices. If data is stored in the database, a server must be installed and configured. For a typical web application, a single server which includes web, application and database servers can be used, for example a LAMP server. If data is obtained directly from a device in the network, there must be a connection between a web application and a device. The connection can be accomplished by using a TCP socket. It is a mechanism that allows messages to be exchanged between two or more applications. A message can be ordinary information (e.g., temperature values) or a command to execute (e.g., turn on/off a device in the network). Since the WSN is adaptable, modular and responsive, the corresponding web application must satisfy the same requirements.

Because of the widespread use of mobile phones, their necessity and considerable coverage of the GSM network, a mobile phone is a convenient choice as a real-time alarm notification device. To accommodate a vast variety of different possible configurations and devices in WSN clusters, the application is modular and easily adaptable to the user's HCS system. Since Wi-Fi connections are not always available and the application relies on the Internet connection to exchange data with the VPS server, it is reasonable to assume that a mobile phone will mostly use a mobile network for the Internet connection. Most mobile network operators charge for Internet traffic based on the amount of data exchanged between a phone and the Internet so it is important to keep data packets as lightweight as possible. Critical information like alarms and certain events are reported in real-time while the rest of data is exchanged only if there is a Wi-Fi connection available. In this way, Internet data load is reduced from the mobile network operator. Choosing the preferable way of data exchange (like GSM or Wi-Fi) is an optional setting that reduces data load on the mobile network operator to a minimum. Enabled Wi-Fi mode of operation leaves only alarm data packets to exchange between a mobile phone and the VPS server over the mobile network and in this way mobile network data load is reduced to a minimum.

3. PRACTICAL IMPLEMENTATION OF TESTBED ENVIRONMENT

The implementation of the HCS system architecture proposed and described in the previous section is done in the laboratory environment and described in this section.

3.1. AN OVERVIEW OF THE WSN

Wireless communication between the coordinator and end devices is achieved by an IEEE 802.15.4 standard based ZigBee protocol in proprietary of ZigBee Alliance. The 802.15.4 standard operates on a subset of 27 available radio channels in specific unlicensed 868 - 868.8MHz, 902 - 928 MHz or 2400 - 24835 MHz Industrial, Scientific and Medical (ISM)bands [4]. Since a 2.4GHz band is open in most countries and provides the highest over-the-air data throughput of 250 kbps with the channel width of 5 MHz, it is a reasonable choice for the WSN RF carrier [5]. The main purpose of the ZigBee protocol is to achieve a low-power, cost-effective wireless network for the purpose of monitoring and control. As the ZigBee protocol is specifically designed for automation applications, it is the best choice to achieve the WSN within the HCS system. A Direct-Sequence Spread Spectrum mechanism built in the physical layer in IEEE 802.15.4. for data transmission and the Offset Quadrature Phase-Shift Keying (O-QPSK) method of modulation ensures a high level of robustness for the ZigBee protocol [6].

The ZigBee protocol supports three different devices within the network: the coordinator, the router and the end device:

- **Coordinator**: The coordinator is responsible for establishing the network, defining mode of operation, allowing and associating other nodes to the network. It is a basic component of the network.
- **Router**: The router is present only in tree and mesh topology networks, it is capable of associating end devices to the network and is used as a packet hopping node.
- **End device**: The end device is a simple node used as a telemetry or actuator device. Other devices cannot join the network through the end device.

The WSN based on the ZigBee protocol can be formed in three different network topologies which directly affects the way messages are exchanged between the devices, messages delay, network stability, etc. The ZigBee protocol provides star, tree and mesh network topologies. Star topology is a centralized network consisting of a single coordinator and multiple end devices. The coordinator represents a central hub which handles every packet within the WSN network. Even though nearby end device fulfill every condition for mutual packet exchange, communication is always handled through the coordinator which can become a network bottleneck. While star topology is simple to

implement, ZigBee clustering is cumbersome when addressing large-scale networks, which makes this topology unstable for a conventional wireless sensor network [7]. Tree topology consists of a coordinator (which represents the central node), routers (which represent intermediate nodes) and end devices. The coordinator and routers represent parents and connected end devices are called children. Parent nodes within the network are only capable of communication with end nodes. A disadvantages of the tree network is that nearby end nodes communicate only through their parent and if one parent becomes disabled, all children connected to it are no longer reachable within the network. Mesh topology supports the same nodes as tree topology but it is implemented as a peer-to-peer network which supports packet multihopping. In mesh topology, devices can easily be added to or removed from the network without affecting its stability. Multihop capability enables better power efficiency management and battery usage than star topology [7]. Mesh topology enables a simple network modularization which fits into described of the WSN within the HCS system.

A test solution made in the laboratory environment consists of a low-cost single-board computer, an XBee-Pro S2B coordinator, an XBee smart plug and a sensor, a mobile and a web application.

XBeePro S2B is a ZigBee module that supports unique needs of an affordable low-power wireless network. The main task of this module is to establish and maintain the ZigBee network for a real-time and stable communication. In combination with an XBee smart plug and an XBee sensor, it creates an HCS WSN. Node objects are *smart plugs* which have both sensing and actuating capabilities, as well as different types of smart devices [8].

Key features of XBeePro S2B modules are high performance, low-cost, low-power, advanced networking and security. Modules can communicate within 90m indoor and 60m in an urban area. TX peak current is 205 mA, RX current is 47 mA, and the power-down current 3.5 μA at 25°C. The Xbee smart plug is an intelligent outlet that can control electrical devices plugged into its electrical outlet. It also includes light and electricity sensors which send real-time data about light intensity and electrical current values. Any electrical device can be connected to it and controlled. The smart plug can extend the range of a wireless network since it has an indoor range of 120m, while maximizing energy efficiency and reducing cost. The XBee sensor sends real-time data about light intensity, humidity and temperature across a Zigbee network. It has the transmit power of 1.25 mW, an indoor range of 40 m, and an outdoor range of 120 m. It can work in different networking topologies and be used with other XBee modules because it is easy to install, configure and integrate in a ZigBee network.

Communication and data gathering starts when the terminal node (sensor, valve, motor, etc.) successfully connects to the coordinator. After successful connection the sensor starts sending telemetry data periodically while state change events are sent after they occur. Each sensor is in hibernation mode between two consistent telemetry readings, so power consumption of sensor devices is directly related to the frequency of sensor telemetry readings. The network passes data from the Xbee smart plug and the sensor to the BeagleBoard computer which stores the received data into a database. Later, the received data can be used for power consumption analyses, prediction of user daily behavioral habits, etc. Wireless sensor network architecture is shown in Figure 2.

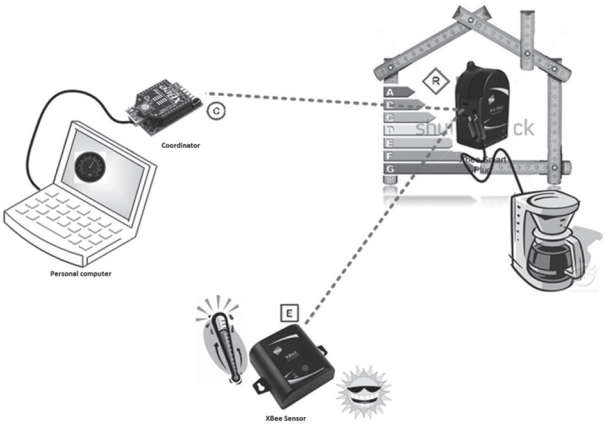

Fig. 2. Wireless sensor network architecture

The XBeePro S2B coordinator is connected through a USB cable with the BeagleBoard-xM computer, the sensor and the smart plug are connected with the coordinator over the WSN as shown in Figure 2. Since any electrical device can be plugged into a smart plug, a desk lamp was chosen for a test solution. There are two reasons for this. First, the lamp can be turned on or off by using a mobile or a web application, so it is visually effective. Second, since the sensor measures light intensity, humidity and temperature, the lamp can change these parameters, which is also important for testing.

3.2. GCM AND VPS IMPLEMENTATION

In the laboratory environment, the VPS was emulated on a single board computer together with the GCM service.

The BeagleBoard-xM is a single-board computer composed of an AM37x 1GHz ARM processor, a PowerVR SGX TM320C64X+DSP graphical unit, 512 MB LPDDR RAM, four USB 2.0 HS ports, DVI-D and S-Video ports, an MMC/SC connector USB mini AB connector and a 10/100 Ethernet connector. It is compatible with Angstrom Linux, Android, Ubuntu and XBMC. BeagleBoard is an improved version of BeagleBoard with a faster processor unit (CPU), more RAM memory and with Ethernet capability. Because of these advantages, the BeagleBoard-xM was chosen in a test solution as a single-board computer.

The Apache server was configured on the BeagleBoard-xM computer and a web application was designed for the HCS. The computer has predefined critical conditions (fire, flood, burglary, etc.) and warning values (minimum and maximum values) stored. If any of the predefined conditions are met, the computer sends (by using the GCM service) a notification in real-time to the user (assuming that the user has a working Internet connection). The GCM has built-In Quality-of-Service (QoS), so in case the user is unavailable, the GCM server will handle all incomplete data transfer sequences. Predefined warning values can be modified by the user. In case the user forgot to close potentially dangerous appliance, it can be turned off by using a mobile phone. A path of the notification message is the same as described previously, only in reverse direction.

3.3. MIDDLEWARE COMMUNICATION

The message transfer system is different for messages sent to and from the user.

If a critical condition is detected by an end node, it sends a message over a ZigBee network to the coordinator which passes it to the BeagleBoard-xM via a USB cable. The BeagleBoard-xM reads a registration key stored in the database and together with the message, it sends it to the VPS server which passes all the data to the GCM server. Upon the reception of a valid registration key, the GCM server passes a received message to registered phones. If a message could not be sent (due to lack of Internet connectivity of the phone), it will be sent as soon as registered phones connect to the Internet.

If a user wants to modify something or turn some appliance on or off via his/her mobile phone, the message is sent over the HTTP protocol through the TCP socket to the VPS server. The VPS server uses the same protocol to communicate with the BeagleBoard-xM. The BeagleBoard-xM will pass the received message to the WSN coordinator which will send it to an appropriate device based on device location and its ID. Middleware communication is shown in Figure 3.

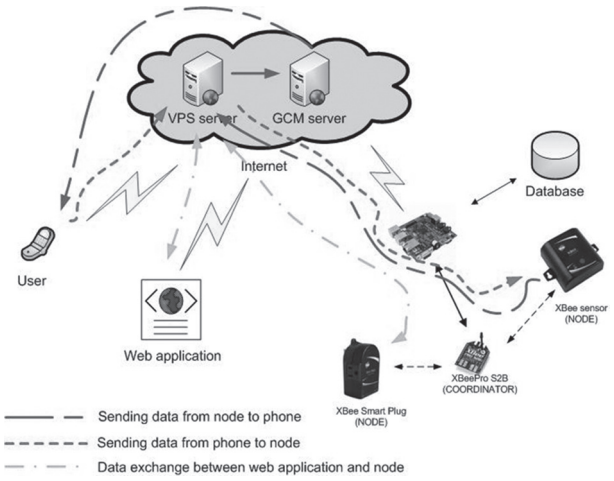

Fig. 3. Middleware communication

The HTTP protocol is a request/response protocol [9], so it is inconvenient to use it for communication in both directions. If it were used for both directions, the user phone would have to constantly keep pulling the VPS server for data to notify the user of any changes. In this operation mode, the user phone would rapidly consume battery and Internet traffic (if connected over GSM to the Internet). To avoid this problem, the HCS uses the GCM server for sending messages from the system to the user phone.

Both scenarios include sending unsecured traffic over the HTTP protocol through the TCP socket which is an alarming security issue. It is left for future work to correct it by using strong cypher algorithms like AES-128 or similar.

3.4. WEB AND MOBILE APPLICATION

A web application consists of two parts, i.e., data display and an interface for adding advanced features. Data is retrieved from the database and displayed on the web by using PHP: Hypertext Preprocessor (PHP) code. If one of the devices in the system is a smart plug, the web application will enable the user to control it by using the TCP socket. The socket server on the computer communicates with the socket client (coordinator) directly. When the coordinator receives a message with the device address and a new state from the socket server, it will check if the device address corresponds with addresses in the database. If the addresses match, the coordinator will send a command to the smart plug and change the state in the database. The interface enables user friendly system upgrades by adding new devices, rooms and other subsystems in the database. Furthermore, it makes the system modular and adaptable. Removing data in the database is not enabled in the web application. It can be done only by connecting to the database directly.

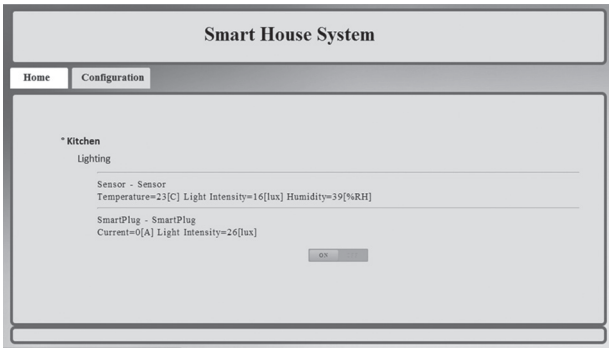

Fig. 4. Web application showing the measured values and the control button for the smart plug

Based on a large number of different living environments, the application has to be adaptable to various WSN systems. Bearing this in mind, the application user interface cannot be statically defined. To solve this issue, the application gathers all information from the BeagleBoard-xM database when it is started for the first

time and when the user is authenticated. When the database is downloaded and stored on the user phone, the application adapts the user interface to the number of elements located in the database. The application consists of three screens shown in Figure 5.

Fig. 5. Application user interface

The first image in Figure 5 shows the application starting screen. This screen is shown only when the user runs the application for the first time. The purpose of this screen is user authorization based on a username and password stored on the BeagleBoard-xM. In this way, a user has some level of access secure from outside. The second image shows the application default screen after the database from the BeagleBoard-xM has been collected and a number of elements have been retrieved. Each element within the WSN has a specific purpose represented by an appropriate screen in the application. The third image in Figure 5 shows the default screen for controlling the environment lightning system controlled by the WSN actuator.

After successful user authorization and application database update, the default start screen, which is shown in the second image in Figure 5, is divided into the following five categories: floors, doors, lights, sensors and windows. Each category within the application corresponds to a group of general devices divided by their functionality and possible location. Floors within the application are presented by a drop-down menu placed at the top right corner representing the location of devices. The rest of the categories are placed in the center of the screen since they represent a group of devices by their function within the HCS. A group of devices that are present on a specific floor are marked by a white transparent background of the icon which notifies the user what devices are located on the selected floor.

One separate screen is dedicated only to providing statistic information of history usage for a specific element. The history screen is shown when the user performs a long-click on a specific device within the room category.

The third image shown in Figure 5 presents a specific screen for light actuators which is categorized into the following three sections: floor, room and light actuator. Floor category is the same as on the default start screen while the light actuators are divided by rooms they are present in. Each light actuator contains current status description, the name of the element that can be customized and the button for turning the light on or off. The device history screen is shown in Figure 6.

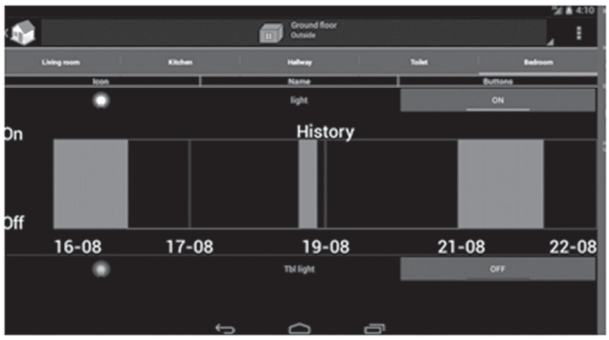

Fig. 6. Element history screen

Before using the GCM service, the user has to complete a web activation of the GCM service to receive a unique ID for mobile phone registration.

Phone registration is a four-step procedure:

1. The phone sends a unique ID (every user receives this ID upon GCM service activation);
2. The GCM returns a registration key;
3. The phone sends the registration key to the VPS and it is stored to the database on the BeagleBoard-xM;
4. The VPS uses this registration key for sending push notifications to the user phone.

Phone registration is shown in Figure 7.

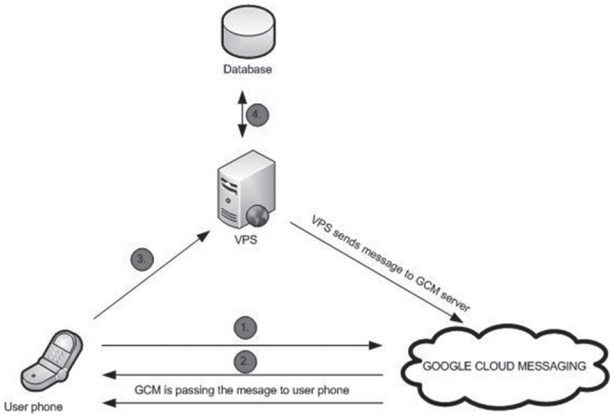

Fig. 7. Phone registration procedure with GCM

The GCM phone registration procedure automatically starts after successful registration with the HCS, and UI adaptation based on the downloaded database is completed.

4. CONCLUSION

In this paper, we have presented a web and a mobile application for regulation of the ZigBee network based home control system. It is a low-cost, easy installation home automation system. A web application allows

the user to control devices over a TCP socket associated with the coordinator through the ZigBee network. Since the web application cannot offer real-time awareness of the system, the mobile application was built to overcome this issue. A mobile application is a bidirectional real-time warning system that notifies the user as soon as warning or alarming conditions are met.

Future work will be dedicated to testing, building and adding (to the presented HCS) extra functionally to simplify user interaction with the system, as well as improving automation for energy efficiency. Additional attention should be paid to system security access from the Internet to the VPS and data encryption for packets sent over the Internet (like VPN implementation).

The entire system was built and tested in the laboratory environment.

5. REFERENCES

[1] Pragnell M., Spence L., Moore R., "The market potential for Smart Homes", Joseph Rowntree Foundation, York, UK, 2000.

[2] L. Bounegru, "Smart Houses: From managing the house at a distance to the management of life itself", University of Amsterdam, Media Studies, New Media, 2009.

[3] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong, M. Poland, "A logical Framework for Behaviour Reasoning and Assistance in a Smart Home", School of Computing and Mathematics and Computer Research Institute, University of Ulster, North Ireland, 2008.

[4] J. T. Adams, "An Introduction to IEEE STD 802.15.4", Frescale Semiconductor Inc, 2006.

[5] White paper, "Demystifying 802.15.4 and ZigBee", Digi International Inc. 2008.

[6] M. Varchola, M. Drutarovsky, "ZigBee based home automation wireless sensor network", Department of Electronics and Multimedia Communications, Technical University of Košice, Acta Electrotechnica et Informatica No. 4, Vol. 7, 2007.

[7] S. Vancin, E. Erdem, "Design and Simulation of Wireless Sensor Network Topologies Using the ZigBee standard", Firat University, Elazig, Turkey, International Journal of Computer Networks and Applications, Volume 2, Issue 3, 2015

[8] F. L. Bellifemine, C. Borean, G. Dini, P. Perazzo, M. Tiloca, "A Home Manager Application for ZigBee Smart Home Networks", Telecom Italia SpA, Dipartimento di Ingegneria dell'Informazione, University of Pisa – Pisa, Italy, 2010.

[9] R. Fielding, U.C, Irvine, J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee: "Hypertext Transfer Protocol – HTTP/1.1", Network Working Group, The internet Society, RFC Editor. United States, 1999