Cena 35,00 zł (w tym 5% VAT)

nr 8/2017

ISSN 0033-2089
e-ISSN 2449-9528

Elektronika

konstrukcje • technologie • zastosowania

Miesięcznik naukowo-techniczny Stowarzyszenia Elektryków Polskich

WYDAWNICTWO *SIGMA NOT*

Wojskowa Akademia Techniczna

Wydział Elektroniki

studia wojskowe · cywilne

studia inżynierskie · magisterskie · doktoranckie

Łączność · Radiolokacja · Metrologia

Elektronika
konstrukcje • technologie • zastosowania
rok LVIII nr 8/2017
MIESIECZNIK NAUKOWO-TECHNICZNY
• MATERIAŁY • KONSTRUKCJE • UKŁADY
• SYSTEMY • MIKROELEKTRONIKA
• OPTOELEKTRONIKA • FOTONIKA
• ELEKTRONIKA MIKROFALOWA
• MECHATRONIKA
• ENERGOELEKTRONIKA • INFORMATYKA

Adres redakcji: ul. Czackiego 3/5, lok. 108 00-043 Warszawa, tel./fax (22) 827 38 79; tel.: (22) 826 65 64, e-mail: elektronika@red.pl.pl, www.elektronika.orf.pl

Zamówienia na reklamę przyjmuje redakcja lub Dział Reklamy i Marketingu, ul. Ratuszowa 11, 00-950 Warszawa, skr. 1004, tel./fax (22) 827 43 65, e-mail: reklama@sigma-not.pl

Kolportaż: ul. Ku Wiśle 7, 00-716 Warszawa, tel. (22) 840 35 89; tel./fax: (22) 840 59 49, (22) 891 13 74, e-mail: prenumerata@sigma-not.pl

Ministerstwo Nauki i Szkolnictwa Wyższego, za opublikowane artykuły przyznaje 8 punktów.

ELEKTRONIKA jest wydawana przy współpracy Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk

Redakcja współpracuje z Polską Sekcją IEEE

Wydawnictwo SIGMA-NOT Sp. z o.o.

PRASA FACHOWA SIGMA-NOT www.sigma-not.pl

00-950 Warszawa, skr. poczt. 1004
ul. Ratuszowa 11, tel. 22 818 09 18
e-mail: sekretariat@sigma-not.pl
Internet: http://www.sigma-not.pl
Indeks 35722

Nakład do 2000 egz.
Skład i druk:
Drukarnia Wydawnictwa SIGMA-NOT

Publikowane artykuły naukowe są recenzowane przez samodzielnych pracowników nauki

Redakcja nie ponosi odpowiedzialności za treść ogłoszeń. Zastrzega sobie prawo do skracania i adiustacji nadesłanych materiałów.

Artykuły publikowane w Elektronice są rejestrowane w bazie danych CrossRef. Każdemu artykułowi przyznawany jest numer identyfikacyjny DOI (*Digital Object Identifier*). Autorzy artykułów publikowanych w Elektronice są proszeni do stosowania numerów DOI w wykazach literatury, wszędzie tam gdzie jest to możliwe.

Wersja papierowa ELEKTRONIKI jest wersją pierwotną.

W ocenie Index Copernicus ELEKTRONIKA uzyskała – 71,36 pkt. Ponadto ELEKTRONIKA jest indeksowana w następujących bazach danych: BazTech, INSPEC oraz OCLC – WorldCat: http://www.worldcat. org/title/elektronika/oclc/10475423

SPIS TREŚCI ● CONTENTS

fotografia na okładce *Robert Krawczak*
projekt graficzny *Mateusz Pasternak*

DOI: 10.15199/13.2017.8.4

Measurement system based on USB Z-Wave controller

(System pomiarowy oparty na kontrolerze USB Z-Wave)

dr inż. MICHAŁ KOWALEWSKI

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Metrology and Optoelectronics

Abstract

A wireless measurement system based on the Z-Wave standard is presented in this paper. The system is composed of a USB Z-Stick Gen5 controller connected to a Personal Computer and a Fibaro controller FGRGBWM-441. Operation of the system is controlled by software written in C++ using OpenZWave library. Some metrological aspects of the system are evaluated: accuracy, linearity, resolution and frequency of voltage measurements.

Keywords: home automation systems, Z-Wave standard, measurement systems

Streszczenie

W artykule zaprezentowano bezprzewodowy system pomiarowy bazujący na standardzie Z-Wave. System jest złożony z kontrolera USB Z-stick Gen5 dołączonego do komputera osobistego oraz modułu Fibaro FGRGBWM-441. Praca systemu jest kontrolowana przez program napisany w języku C++ w oparciu o funkcje biblioteki OpenZWave. Przeanalizowano wybrane aspekty metrologiczne systemu związane z pomiarem napięć: dokładność, liniowość, rozdzielczość i częstotliwość pomiarów.

Słowa kluczowe: systemy automatyki domowej, standard Z-Wave, systemy pomiarowe

Wireless Sensor Networks (WSN) have been widely used in metrology for many years. There are many areas of application: infrastructure protection, diagnosis, industrial measurements, monitoring environmental parameters [7]. One area of application is home automation systems. Different standards can be used for wireless communication in these systems: X10, ZigBee, Z-Wave, EnOcean, INSTEON [3], Bluetooth low energy. Most widely spread is the Z-Wave standard [10], which is distinguished by the following features: high inter-brand operability, ability to work as repeaters which increases the transmission range up to 150 m, ease of installation and a high level of transmission security resulting from using the 128-bit Advanced Encryption Standard (AES).

Typically, the Z-Wave standard is applied in remote monitoring and controlling devices in smart homes. It can also be used for remote measurement of signals acquired from different sensors: in energy meters [1], or in impedance spectroscopy of objects either of considerable dimensions or installed in hard to access locations (bridges, poles) [4-6].

A solution proposed in this paper consists in creation of a simple ad-hoc Z-Wave network composed of a USB Z-Stick Gen5 controller connected to a Personal Computer and a Fibaro controller FGRGBWM-441. The Fibaro RGBW controller enables to measure voltage signals, e.g. from analogue sensors. A brief description of the software and metrological aspects of the system are discussed.

Z-Wave network structure

The Z-Wave network is a mesh-type one. We can distinguish three types of network nodes: Controllers (C), Routing Slaves (RS) and Normal Slaves (NS) [9]. An example of the Z-Wave network structure with different types of nodes is shown in *Fig. 1.*

Controller is the main node in the network. It can build and manage the network and communicate with each node, if a route to that node exists. There can be more than one controller in the network, but only one of them is Primary Controller. All other

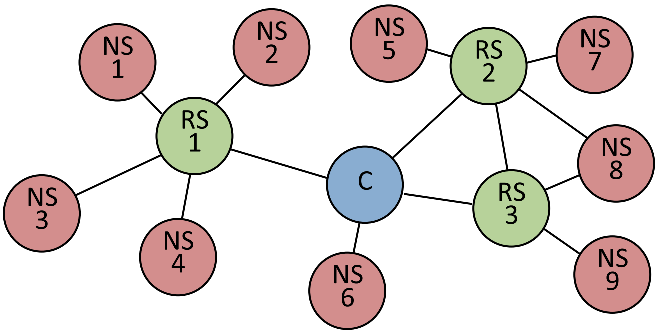

Fig. 1. An example of the Z-Wave network structure
Rys. 1. Przykładowa struktura sieci Z-Wave

controllers are called Secondary Controllers. Primary Controller can add new nodes to the network by assigning its Home ID and providing Node IDs to other nodes (also to Secondary Controllers). This process is called Inclusion. Primary Controller also can exclude existing nodes. Exclusion of a node is accomplished by reverting the actions performed in Inclusion. The Home ID field for an excluded node is deleted and all information about this node is deleted from the routing tables of the network.

All other nodes in the network are slaves. There are two types of slaves: the normal and the routing ones. A Normal Slave has no information about the routing table. It means that it does not know its neighbours and can only reply to requests from other nodes. A Routing Slave, on the other hand, has a partial knowledge about the routing table associated with the possibility of communication with neighbouring nodes. It can also initiate communication with other nodes.

Basic specification of Z-Wave network

The Z-Wave network uses license-free and regulated frequency bands. In Europe it operates in a frequency range between 863 and 870 MHz.

The Frequency Shift Keying modulation and one of two encoding algorithms are used: Manchester for a data rate of 9.6 kbit/s and Non Return to Zero (NRZ) for 40 kbit/s and 100 kbit/s data rates.

The Z-Wave network works with a peak transmission power of 10 mW. This power level is applied for 1% of the time, which corresponds to an average radiation power of only 0.1 mW. Hence, the maximum distance between nodes in an open space is limited to about 30 m [10]. Hopefully, the Z-Wave standard makes possible the multi-hop transmission from a transmitter to a receiver involving up to 4 intermediate nodes. This increases the available distance between the most distant nodes to about 150 m.

Each node in the Z-Wave network has two identifiers: Home-ID and Node-ID. Home ID is a 4-byte constant for all nodes in one Z-Wave network, while Node-ID is a unique 1-byte identifier used to distinguish nodes in the network.

A few Node-IDs are reserved for network organization procedures, hence the maximum number of devices in one Z-Wave network is 232.

Nodes in the Z-Wave network can communicate with each other in three patterns: Single-cast, Broadcast and Multicast. In the Single-cast communication pattern a data frame is transmitted from the source node to one destination node. In the Broadcast communication it is transmitted from the source node to all nodes in the network. Finally, in the Multicast communication pattern it is transmitted from the source node to a set of nodes. In this case addresses of nodes, to which a data frame is to be transmitted, are encoded in a Multicast Data Frame.

In the Z-Wave network messages can be transmitted from the source node to the destination node via different routes. This is characteristic feature of a mesh network. Information about possible routes between nodes is included in the routing table. In the example of network presented in *Fig. 1* there are 13 nodes: one Controller, 3 Routing Slaves and 9 Normal Slaves. In this network not all nodes can communicate directly with each other, e.g. Controller can communicate directly with nodes RS2 and RS3, but not with node NS8. According to the routing table there are different possible routes from Controller to node NS8: C-RS2-NS8, C-RS3-NS8, C-RS2-RS3-NS8 and C-RS3-RS-2-NS8. If communication via a selected route fails, it is marked as a failed one and another possible route is selected.

The Z-Wave network assures reliable communication. This is accomplished by two way communication using a message acknowledgement. Except Broadcast communication every message sent from a source node to a destination node must be acknowledged. If a message is not acknowledged the source node retransmits it. After three unsuccessful attempts the source node reports a failure and an alternative way, if it exists in the routing table, is selected.

Measurement system architecture

The architecture of Z-Wave measurement system, prepared for the needs of this paper, is shown in Fig. 2. It is composed of a USB Z-Stick Gen5 controller connected to a Personal Computer and a test module with the Fibaro RGBW controller. The test module is equipped with four potentiometers used to simulate signals from analogue sensors of different parameters e.g. temperature, humidity, pressure, light intensity. The USB Z-Stick controller enables to communicate with the test module and to acquire measured values using wireless transmission based on the Z-Wave standard. The acquired data are processed with a PC test program written on the basis of an OpenZWave library. It is connected with a selected Z-Wave controller and enables to control Z-Wave devices.

Fig. 2. The measurement system architecture
Rys. 2. Architektura systemu pomiarowego

Fibaro RGBW controller

The Fibaro RGBW Controller is a universal Z-Wave-compatible device with four controllable outputs (R,G,B and W) (*Fig. 3*) [8]. The signals generated on these outputs are Pulse Width Modulated (PWM) ones and they allow to control LED, RGB or RGBW strips, halogen lights and fans. The controlled devices can be powered by a DC voltage in a range from 12 V to 24 V. The device supports up to four analogue sensors operating in a range from 0 V to 10 V. Signals from these sensors are applied to IN1, IN2, IN3 or IN4 terminals.

The Fibaro RGBW controller can operate in one of two modes: RGB/RGBW mode or IN/OUT mode. In the RGB/RGBW mode IN1, IN2, IN3 and IN4 inputs can work with momentary or toggle switches to control designated channels (IN1 controls R channel, IN2 controls G channel, IN3 controls B channel and IN4 controls W channel). If this mode is chosen, settings for all channels are identical. In the IN/OUT mode all inputs and outputs may be configured independently according to the manual.

The Fibaro RGBW controller contains a set of configurable parameters, which enable to switch between modes and to specify details of its operation. The most important are: outputs state change mode (8), step value (9), time between steps (10), time to change value (11), inputs/outputs configuration (14) and reporting analogue inputs change threshold (43). These parameters are modified in the later-described test program.

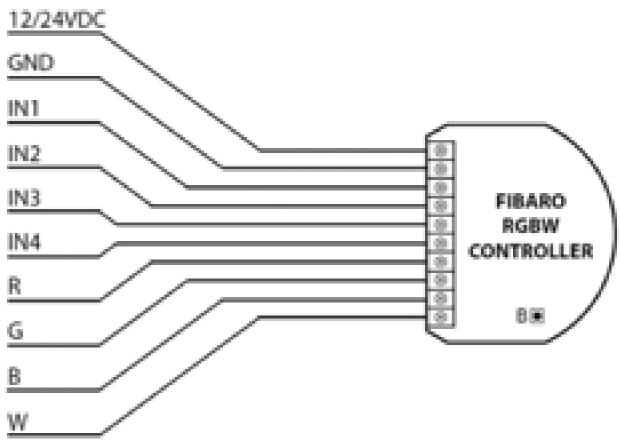

Fig. 3. Description of the Fibaro RGBW controller terminals
Rys. 3. Opis wyprowadzeń kontrolera Fibaro RGBW

Test module

For the purpose of this paper a test module was constructed. It is equipped with the Fibaro RGBW controller, a 10 V voltage stabilizer, a set of four 10 kW potentiometers used to apply

Fig. 4. A test module with the Fibaro RGBW controller
Rys. 4. Moduł testowy z kontrolerem Fibaro RGBW

voltages from 0 V to 10 V to IN1, …, IN4 inputs, four toggle switches to select a mode of operation for each channel (analogue voltage measurement or configuration with momentary switches) and four buttons to switch PWM signals on or off or change their duty cycles.

OpenZWave library

OpenZWave [2] is an open-source, cross-platform library designed to support the Z-Wave communication protocol in WSN. It offers a set of classes, structures, unions and interfaces to manage nodes in the Z-Wave network. All Z-Wave functionality is accessed through the ***Manager*** class.

Communication between the PC and nodes in the Z-Wave network is asynchronous. Some slave nodes sleep most of the time to save battery power, and can only receive commands when awake. Hence, e.g. a response to a query send from Controller to read a value from a node may not occur immediately. For this reason, many functions in the OpenZWave library use a system of notification callbacks. The notification handler is in the core of any application using the OpenZWave library. It is the place where all information regarding device configuration, modification of values and determination of a state will be reported. The notification callback system can also be used to inform the application about any changes to the structure of the network (adding or removing nodes).

Controlling operation of a node in an application based on the OpenZWave library depends on modification of so called “values”. This is achieved by means of the ***ValueID*** class. This class is used to uniquely identify a value reported by a Z-Wave node. Various identifying characteristics are packed into a single 32-bit number. These are: Node ID of a device, genre of value, ID of a command class that created and manages this value, Index of value within all the values created by the command class, an instance (when changing a configuration parameter value an instance is a number of the parameter) and a type of value (bool, byte, string, etc.).

Usually, a Z-Wave node contains a set of configurable parameters. In the OpenZWave library the configuration parameters are values that are managed by the ***Configuration Command Class***. These values are device-specific and are not reported by the nodes. To obtain information about configuration parameters one needs to use the device manual.

In a template of a project based on the OpenZWave library one needs to perform the following tasks:

1. On the start of application – call ***Options::Create*** providing paths to the OpenZWave configuration folder with manufacturer-specific settings of devices, the User data folder and any command line string containing program options.
2. Call the ***AddOptionBool***, ***AddOptionInt*** or ***AddOptionString*** methods of the class ***Options*** to add any application-specific configurable options.
3. Call the ***Options::Lock*** method.
4. Call the ***Manager::Create*** method to create an instance of the ***Manager*** class (There can be only one instance of that class).
5. Call ***Manager::AddWatcher*** to install a notification callback handler.
6. Call ***Manager::AddDriver*** for the Z-Wave controller attached to the PC. The controller will handle all messages transmitted between the controller and other nodes. The Driver will read any previously saved configuration and then query the Z-Wave controller for any missing information. Once that process is complete, a ***DriverReady*** notification callback will be sent containing Home ID of the controller. The Driver will then poll each node in the network to update information about each node. After all “awake” nodes have been polled, an ***AllAwakeNodesQueried*** notification is sent. Finally, after all nodes (whether listening or sleeping) have been polled, an ***AllNodesQueried*** notification is sent.
7. Perform any action in the Z-Wave network with use of appropriate methods of the ***Manager*** class.
8. Call ***Manager::Destroy*** to allow OpenZWave to clean up and delete other objects it has created.
9. On the application exit - call the ***Options::Destroy*** method.

The network configuration is saved automatically on the application exit and restored on the next run of the application. All changes that occur to the network while the application is not running are detected by the ***Manager::AddDriver*** method called during the next run of the application.

Integration of software components

The test program was written in C++ in the Visual Studio 2015 programming environment on the basis of discussed earlier template of the project based on the OpenZWave library. The program enables to control basic functions of the Fibaro RGBW controller. In order to eliminate the need to dynamically create the network structure with functions of the OpenZWave library, inclusion and exclusion of the RGBW controller is performed with use of “IMA tool” software of Aeon labs. The test program - through the USB Z-Stick controller - configures the mode of operation of the RGBW controller, modifies configurable parameters, changes the duty cycles of PWM signals generated on R, G, B and W outputs, shows the measured analogue voltage values on IN1, …, IN4 inputs and displays the measured power load of the RGBW controller.

The panel of the test program is shown in Fig. 5.

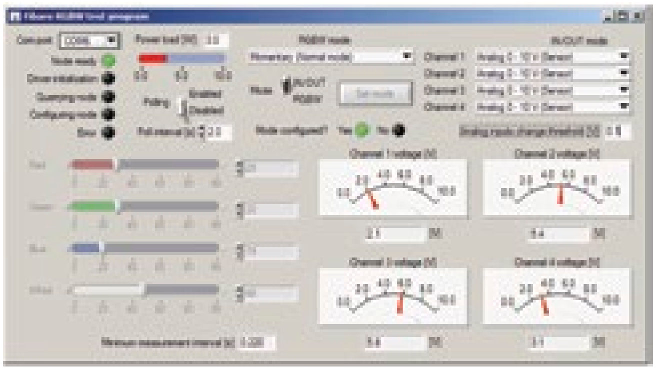

Fig. 5. The panel of the test program
Rys. 5. Panel programu testowego

A graphical interface of the program was created in the National Instruments LabWindows/CVI environment.

An object file with prototypes of panel and control callback functions was created. This file with additional library files of LabWindows/CVI (***cvirt***, ***cvisupp***, ***cviwmain***) was attached to the project in the Visual Studio.

Test program algorithm

The test program operates as follows:

- Described earlier tasks from 1 to 5 are performed,
- A list of available USB ports in the PC is created,
- The panel of program is displayed,
- The program enters the main loop, where the following tasks are performed:
  - o when the user selects an appropriate port number, task 6 is executed,
  - o after successful initialization the access to the panel controls is activated,
  - o the states of controls are updated in response to notifications from the notification callback handler,
  - o when the user tries to terminate the program, it exits the main loop,
- The panel of program is destroyed,
- Described earlier tasks 8 and 9 are performed.

Evaluation of metrological aspects

A series of measurements with use of a digital oscilloscope was performed (Fig. 6). The aim was to verify the Fibaro RGBW controller specification and to evaluate the accuracy of measurements.

The measurements are burdened with a significant linearity error, hence obtaining the correct value of voltage requires multiplying the value indicated by the software by a constant factor equal to 1.08. The real measurement voltage range is increased up to about 10.5 V.

Resolution of voltage measurements is limited by the firmware of Fibaro RGBW controller to 100 mV. Many analogue temperature sensors (e.g. LM35) has a scale factor equal to 10 mV/°C, hence in order to increase resolution of temperature measurements to 1°C one needs to use an additional 10x amplifier based on an operational amplifier.

A frequency of PWM signals generated on R, G, B and W is set to a constant value equal to 244.8 Hz. The duty cycle of these signals can by controlled by a value ranging from 0 to

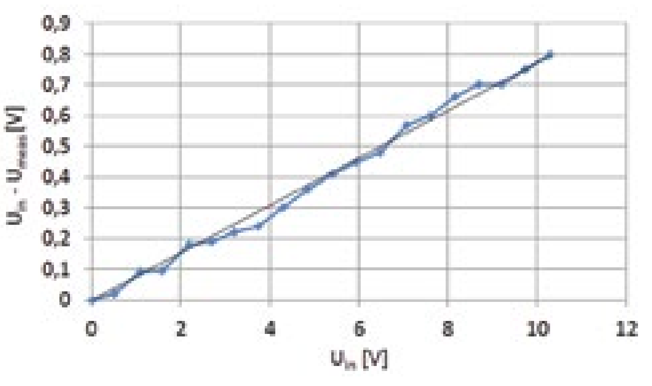

Fig. 6. An accuracy of voltage measurements (U_{in} – the value indicated by the software, U_{meas} – the value measured with use of the oscilloscope)
Rys. 6. Dokładność pomiarów napięć (U_{in} – wartość wskazana przez oprogramowanie, U_{meas} – wartość zmierzona przy użyciu oscyloskopu)

99, hence the minimum change in the duty cycle is 1%. The measurements made with the oscilloscope indicate that the maximum difference between the set and measured values does not exceed 1,5%.

One of the metrological aspects of a WSN is a minimum delay between voltage measurements. It depends on many factors, e.g.: a method of voltage measurement, a structure of the network (number of intermediate nodes used in transmission), a protocol of communication, a transmission data rate, a security mechanism. In the test system composed of only two nodes (Controller and the Fibaro RGBW controller) this time should not vary significantly. Besides, in the Fibaro RGBW controller there is no need to initiate each measurement, because the device has a mechanism for reporting a voltage change if it exceeds the programmed threshold value (parameter 43). This eliminates the need of sending a query that would initiate each measurement. The minimum time delay obtained in a series of several hundred measurements made in the same conditions is 50 ms. This value may increase, because it depends on the load of the operating system.

Conclusions

A growing interest in WSNs stimulates the development of wireless communication standards. The widespread Z-Wave standard is designed to control operation of home automation systems. As was presented in this paper, it can also be used to design a measurement system for wireless acquisition of signals from different analogue sensors, e.g. of environmental parameters. In order to create a simple ad-hoc Z-Wave network one needs only a USB Z-Wave controller connected to a Personal Computer and a measurement unit with the Fibaro RGBW controller. The software solution based on the open-source OpenZWave library does not require a significant time for preparation of a program that controls the operation of the system. While the measurement speed and accuracy obtained by using the Fibaro RGBW controller are limited, this solution is adequate for measuring slow change signals, where a high precision of measurements is not the key aspect.

References

[1] Amaro P., Cortesao R., Landeck J., Santos P. 2011. „Implementing an Advanced Meter Reading infrastructure using a Z-Wave compliant Wireless Sensor Network". *Proc. of the 3rd Int. Youth Conference on Energetics (IYCE)*, 2011, p. 1–6.

[2] Documentation of the OpenZWave Library avalilable from: www.openzwave.com.

[3] Gomez C., Paradells J. 2010. „Wireless home automation networks: A survey of architectures and technologies". *IEEE Communications Magazine*, vol. 48, no. 6, 2010, p. 92–101.

[4] Hoja J., Lentka G. 2010. „System do bezprzewodowego monitorowania jakości powłok antykorozyjnych obiektów trudnodostępnych". *Pomiary Automatyka Kontrola*, vol. 56, no. 9, 2010, p. 1100–1103.

[5] Hoja J., Lentka G., Rolbiecki P. 2011. „Rozproszony system spektroskopii impedancyjnej". *Przegląd Elektrotechniczny*, vol. 87, no. 9, 2011, p. 20–23.

[6] Kowalewski M., Lentka G. 2012. „Remote Monitoring System for Impedance Spectroscopy using Wireless Sensor Network", *XX IMEKO World Congress*, Busan, Republic of Korea, 2012.

[7] Kulesza W. 2010. „Wireless Sensor Networks – Measurement, Telecommunication and Energy Approach". *V Kongres Metrologii*, Łódź 2010.

[8] Manual of the Fibaro FGRGBWM-441 controller.

[9] Paetz C. 2005. „Z-Wave Basics: Remote Control in Smart Homes", eBook, 2005.

[10] Withanage C, Ashok R., Yuen C., Otto K. 2014. „A Comparison of the Popular Home Automation Technologies". *IEEE Innovative Smart Grid Technologies* ISGT ASIA, 2014.