ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20190730075217
Subject review

Smart Home Solutions Using Wi-Fi-based Hardware

Darko HERCOG*, Dominik SEDONJA, Blaž RECEK, Mitja TRUNTIČ, Bojan GERGIČ

Abstract: Home automation technology has been increasingly important in our lives, since it offers numerous advantages, e.g., greater comfort, safety, security and energy efficiency. A smart home automation system usually includes a central computer with deployed home automation software and several distributed sensors and actuators. Wired connections between a central computer and sensor/actuator nodes are already well established, however, wireless solutions are an emerging trend. This work addresses smart home automation solutions that are based on wireless Wi-Fi network. Such solutions enable an upgrade of an existing house into a smart house without modifications of hardware installations. The article includes an overview of related works in this research field, and a case study of cost effective home automation solution that is based on open source home automation software and wireless, custom developed, Wi-Fi based hardware.

Keywords: domoticz; ESP8266; home automation; MQTT; Node-RED; Raspberry Pi; smart home; wi-fi node; wireless wi-fi module

1 INTRODUCTION

Nowadays, home automation technology is becoming a popular paradigm because of its numerous advantages. In a home automation system, home appliances are connected to an existing or dedicated network, and can be controlled, monitored and automated through a central application [1]. Home automation systems could improve the quality of daily living because they make human life more: (1) Comfortable, (2) Healthy, (3) Safe, and (4) Energy efficient. Comfortability is related mainly to the measurement and analysis of home ambient conditions and adjustment of these conditions to custom needs. Comfortability includes control of HVAC (Heating, Ventilation and Air conditioning) systems, control of windows' shades and blinds, control of light intensity and light color, voice control, etc. In addition to comfortability, also health improvement can be achieved using a home automation system, by continuous monitoring of the air quality and alert generation in the case of bad air or gas leakage detection. More advanced home automation solutions could also include measurement of human vital functions such as blood pressure and heart rate monitoring [2]. From the safety and security point of view, home automations systems take care of fire, theft and burglary detection. This includes detection of doors' and windows' openness, control of gates' locks, and other systems, like face recognition, home surveillance cameras, etc. The last, but not least, is energy saving**.** Using home automation systems and appropriate scheduling algorithms, energy efficient control could be achieved of electric consumers, like lights, water heaters, HVAC systems and home appliances. Consequently, the cost of living in such a house could be a lot reduced, since some consumers turn off or operate with much less power automatically in certain situations. In addition to the above benefits for regular users, home automation solutions are also very suitable for users with some types of physical disabilities, such as visual or hearing impairment or, for example, dementia [3]. Home automation technology can give them independence, and make their lives much safer and more comfortable. Home automation systems also enable a more independent life for elderly people who are a rapidly growing population in the western world. According to Eurostat's projections [4], the share of very old people (80 years and over) in the total EU-28 population is projected to increase from 5.4% (27.3 million) in 2016 to 12.7% (66.1 million) by 2080. According to evaluation performed in [5], homes with voice control have a great potential to ease daily living for elderly and frail persons.

Although home automation systems offer numerous advantages to users (regular, elderly and users with disabilities), they are usually very expensive and relatively closed (the users cannot adjust the system to their own needs). In addition, in a majority of cases, modifications of electrical installations are also required, which discourages many users from investing in the new smart home technology. On the other hand, by using inexpensive hardware and one of the open source type home automation software [6], a powerful, flexible and reasonably priced home automation solution can be achieved. This article provides a review of smart home automation solutions that are based on wireless technologies, especially on Wi-Fi (Wireless Fidelity) network. In addition, an article also includes a case study of cost effective home automation solution that is based on open source home automation software and wireless, custom developed, Wi-Fi based hardware. Presented solution includes embedded computer, with deployed open source home automation software, and several Wi-Fi enabled sensors and actuators (Wi-Fi nodes). The central computer retrieves data from sensors on Wi-Fi nodes, performs data processing and decision-making algorithms, and uploads executive data to actuators on the connected Wi-Fi nodes. Using presented solution, a user is able to control electrical appliances remotely by several ubiquitous devices (smartphone, laptop or any Wi-Fi enabled device).

The remainder of this paper is organized as follows: Section 2 contains state-of-the-art in the field of Home Automation Solutions using wireless Wi-Fi network. A case study example is presented in Section 3. This section includes a short description of system architecture, communication protocol and hardware/firmware description of the developed Wi-Fi nodes. Section 4 contains an example of the use of developed Wi-Fi nodes in combination with open source home automation system. Finally, the conclusions are stated in Section 5.

2 LITERATURE REVIEW

In recent years, there have been numerus studies focusing on home automation and smart homes. In [7] the authors present the design and implementation of a simple smart home system that is based on Raspberry Pi and Android application. The system is composed of a group of sensors, a Raspberry Pi device as the main server system, a Wi-Fi router and home appliances. All sensors are connected directly to the Raspberry Pi, while home appliances communicate with the main computer via the Wi-Fi router. Raspberry Pi acquires the data from sensors and forwards it to the cloud, using an existing Wi-Fi network. The users are able to observe and control home appliances using the Android app, which communicates with the main computer using Bluetooth communication. In [8] Monika Rana et al. present a smart home system based on ATMEL ATmega128 microcontroller, light and temperature sensor, and Bluetooth to UART module (HC-05). Connected appliances could be controlled remotely using Bluetooth and an Android app. In [9] the Raspberry Pi computer has been used as a sensor node for home automation. The Raspberry Pi gathers information from 2 analog temperature sensors (B57045K10 and NTC 10 k) via an external I2C 8-bit A/D (analogue to digital) and D/A (digital to analogue) converter (NXP PCF8591). Acquired data is available to the remote user using an Apache Tomcat server and RESTful based Web services. In [10] the authors present a home automation solution that is based on Microchip PIC and Arduino microcontrollers and an LM400 Bluetooth communication module. The developed security subsystem is based on PIC12F675 and a PIR sensor, lighting control subsystems on Arduino UNO and light sensor, while the temperature sensing subsystem uses an LM35 sensor. Devices could be controlled remotely using an Android app. Another solution of home automation solution is proposed in [11]. A wireless node, which is composed of a NodeMCU development board, Light Dependent Resistor (LDR), Light-Emitting Diode (LED) and buzzer, sends/retrieves data to/from an Eclipse Mosquitto message broker [12] that runs on a Windows PC. The wireless node acts as publisher and subscriber. The intensity of light is sensed by NodeMCU and published to the MQTT broker, while commands to control the LED and buzzer are retrieved from the broker. In the presented solution, an MQTTLens (Google Chrome application) and MyMQTT (Android app) client applications have been used for retrieving sensor data and toggling the LED and buzzer. Both applications are able to subscribe and publish MQTT topics. In [13], authors present slightly modified version of [11]. In this solution, an ESP8266 NodeMCU board interfaces with DHT11 and an LDR sensor to monitor the ambient condition. The brightness level of 8 × 8 Neopixel matrix is controlled with regard to the measured light intensity level. In this solution, an Adafruit.io [14] cloud-service has been used as an MQTT broker, while the Graphical User Interface (GUI) has been developed using Adafruit dashboard. In [15] the authors present a home automation solution using NodeMCU connected to four channel Single Pole Double Throw (SPDT) relays. In this solution, the NodeMCU board acts as a web server. The user is able to control domestic appliances (lights, fan, bulb and charger) using an Android app which has been designed using open source MIT App Inventor 2. The developed app enables remote control of connected appliances using GUI objects or by voice commands. In [16], the authors present a wireless emergency response system for fire hazards using IoT and an ESP-32 module. The wireless node includes an ESP-32 module, flame detection sensor, gas sensors (MQ-2 and MQ-5) and GPS module. The sensors detect the hazard and alert the local emergency rescue organizations like Fire Departments and Police by sending the hazard location to the Adafrui.io cloud-service. The sensed data and the location coordinates are published on to the Arduino.io cloud service using the MQTT protocol. In [17], an indoor monitoring system has been presented for ambient assisted living. The system is composed of an iAQ (indoor Air Quality) sensor, iAQ Gateway, iAQ Web portal and iAQ mobile application. An iAQ sensor system, which is based on an Arduino Mega board contains five sensors: Air temperature (SHT10), relative humidity (SHT10), carbon monoxide (MQ7), carbon dioxide (T6615), and a light sensitive resistor (GL5528). Communication between the iAQ sensor and iAQ Gateway is performed using ZigBee technology with an XBee module. The iAQ Gateway, which is based on a Wemos Mini D1 module, forwards the retrieved data (from the iAQ sensor) to a MySQL database using RESTfull web services. The user can access the stored data via the iAQ Web portal, or using the Android iAQ mobile application. In [18], Quintana-Suarez et al. present a low cost wireless acoustic sensor for remote monitoring of elderly people. The system is composed of an electret omnidirectional microphone (CMA-4544PF-W), an operational amplifier (op-amp) optimized specifically for use as a microphone preamplifier (Maxim MAX4466), and an ESP32 microcontroller which is capable of recording ambient sounds at least at 10 kHz and 12-bit resolution. The acquired data is firstly stored in a circular buffer on an ESP32 microcontroller, and then the raw data are transmitted to the server. The presented solution infers hazardous situations via environmental sounds' identification and, consequently enables enhancement of the quality and safety of elderly persons. In [19], the authors present "Smart Lamp", that enables indoor thermal comfort optimization and energy savings. The Smart Lamp includes an Arduino Mega board, temperature and relative humidity sensor (DHT22), Real-Time Clock (RTC) module, an infrared (IR) LED for HVAC system control, and a Bluetooth module for data transmission to a smartphone or a tablet. Electronic components have been integrated into the standard lamp base. Using an Android app, which was created using an MIT App Inventor, the user is able to monitor the temperature and the relative humidity and verify the operating status of the HVAC system. A control algorithm, deployed on a microcontroller, switches off the HVAC system automatically using IR communication, outside the office hours. In [20], the same authors present a Do-It-Yourself (DIY) indoor air and lighting quality control solution. The system is composed of monitoring and actuation stations. The first one includes an Arduino UNO, XBee S2 communication module, Real-Time Clock (DS1307), CO2 sensor (K30) and photo resistor, while the second one an XBee S2 module and a 2-channel relay module. Based on actual and desired values, the system enables activation/deactivation of the lighting and air exchange systems. In [1], Bhatt et al. present a home automation solution, that is based on wireless Wi-Fi nodes, a central computer (Raspberry Pi) with deployed MQTT Broker and openHAB home automation software. Wireless nodes, which are based on ESP8266 communicate over a dedicated or existing Wi-Fi network with a central computer using the MQTT protocol. OpenHAB has also been used in [21], where the authors present an accelerometer-based fall detection system. The smart fall system includes a Nordic Semiconductor NRF51822 microprocessor with integrated Bluetooth Low Energy (BLE) transceiver and external I2C triaxial accelerometer (Bosch BMI055). The microprocessor acquires acceleration values from the BMI055 sensor, analyzes the data, and, on fall detection events, sends notification information into OpenHAB GUI. A home automation solution using the Internet of Thing (IoT) is presented in [22]. In this solution, Amazons cloud server has been used for data storage, sensor feedback and control. The experimental setup is composed of a Raspberry Pi computer, relay board, temperature and humidity sensors, mechanical limit switches, light sensor and a PIR sensor for motion detection. Raspberry Pi acquires the data from connected sensors periodically and uploads it into the Amazon cloud server. The user is able to observe the status of connected sensors and control actuators using the Android application. Connection between the GUI (Android app) and Amazon server has been implemented using the Google Cloud Messaging (GCM) service.

In addition to the stated work, there have also been many studies focusing on detection of anomalous behaviors in smart homes using advanced data processing algorithms. In [23], for example, the authors present a visual system that detects and comprehends unusual behaviors of residents, and predicts their health information. In [24], the authors present a detection of resident anomalous behaviors using machine learning algorithms, while in [25], an estimation of elderly well-being conditions based on usage of house-hold appliances, has been presented. In [26], the authors introduce a fuzzy logic controller in order to improve home energy management. Fuzzy logic has also been used in [27], where the authors present a predictive model for detecting gases.

Table 1 Summary of research works regarding the used microcontroller

Microcontroller	Reference
ATmega128	[8]
ESP8266	[1, 11, 13, 16]
ESP32	[16, 18]
Raspberry Pi	[7, 9, 22, 28]
Arduino UNO	[10, 20]
Arduino Mega	[17, 19]
PIC (16F877, PIC12F675)	[10]

In the last few years, the use of smart speakers (Amazon's Echo, Google's Home or Apple's HomePod) in combination with voice assistants (Amazon's Alexa, Google's Assistant and Apple's Suri) have been on the rise. These solutions enable voice control of domestic appliances, however, in the current version, they do not represent a replacement for a dedicated home automation system. The key weakness lies in the very limited support of automation scripts, which are indispensable in this area.

In the following Tables, the usage of microcontrollers and GUI in stated home automation solutions is summarized.

Table 2 Summary of research works regarding the used GUI

GUI	Reference
Android app	[7, 8, 10, 15, 17, 19, 22]
Open HAB	[1, 21]
Web services (RESTful)	[9, 17]
Adafruit.io dashboard	[13, 16]

2.1 Comparison of Stated Solutions

Stated solutions, which are based on microcontrollers without an integrated communication unit, include an additional communication module (like ZigBee, Bluetooth [8, 10, 19], or other), that enables data exchange between the central computer and the wireless node. Since the communication modules increase the overall size and the price of such node, an optimized and cost effective solution is quite difficult to achieve using this approach. In some other solutions [7, 9, 22, 28], the sensor/actuator nodes have been realized using Raspberry Pi computers. Although this embedded computer is very powerful, it is not very suitable for wireless node due to the following reasons: (1) Price, (2) Power consumption, (3) Size, and (4) The lack of A/D converter. Raspberry Pi costs more than 30 EUR, and, with additional sensors and actuators, the price of such wireless node could be quite high. In addition, the Raspberry Pi is not very small and has much higher power consumption than the majority of microcontrollers. Raspberry Pi does not include an A/D converter; therefore, an external converter must be utilized in the case of use of analog sensors. Consequently, the complexity and the price of such wireless node could be relatively high. In contrast, using solutions that are based on a low power and small size microcontroller with integrated A/D converter and Wi-Fi module, cost effective wireless nodes could be produced. Some mentioned solutions operate only using the cloud services (for example [16, 22]), which means that access to the Internet is urgently needed for proper functionality. Such smart home solutions are somewhat easier to implement; however, they always raise safety, reliability, security and privacy issues. In the majority of stated solutions [7, 8, 10, 17, 19, 22], the users are able to observe the status of connected sensors, and control actuators using a developed Android app. In most cases, an Android app has been designed for specific configuration. Consequently, in the case of additional sensor(s) and/or actuator(s), an app has to be modified and rebuilt.

According to pros and cons of stated solutions it is evident that an optimal, low costhome automation solution could be realized using low cost microcontroller with integrated Wi-Fi chip and an open source home automation software. Such solution would overcome some limitations of the stated solutions and enable more advanced users to implement their own control or scheduling algorithms. An example of such a solution, using custom developed Wi-Fi nodes and Domoticz home automation software is presented in the next section.

3 A CASE STUDY EXAMPLE

3.1 System Architecture

The system is composed of three main parts: (1) Central computer with deployed home automation software, (2) Several custom developed wireless sensor/actuator nodes (Wi-Fi nodes), and (3) A wireless router. All developed Wi-Fi nodes are based on microcontroller with integrated Wi-Fi chip, and communicate with the central computer via wireless router. A home automation system deployed on central computer, retrieves information from different sensors (temperature, humidity, motion detection, etc.), and, based on the user settings and automation scripts, controls actuators (lights, heaters, etc.). In addition to home automation software, the central computer also hosts an Mosquitto MQTT Broker [12], which implements the MQTT protocol [29]. Using this protocol, the data exchange between the central computer and the Wi-Fi nodes is realized. In the presented solution, the users can access the home automation GUI by using either the web browser or a smartphone application. Using the GUI, the user can observe the state of installed sensors and turn on/off individual actuators.

3.2 Central Computer and Communication Protocol

The Raspberry Pi 3 model B has been selected as the central computer. Raspberry Pi is very suitable as a main home automation computer, since it has many advantages: (1) A powerful processor, (2) integrated LAN, wireless LAN and Bluetooth, (3) Relatively low power consumption, (4) Small dimensions, and (5) Low price.

Communication between the central computer and developed Wi-Fi nodes has been implemented using the MQTT protocol [29], although there are also other light messaging protocols for M2M communications ([30, 33, 34]). MQTT is an open, simple and lightweight messaging protocol designed for Machine-to-Machine (M2M) communications. MQTT is a publish/subscribe messaging protocol, where every message is published to an address, known as a topic. An MQTT client (Wi-Fi node) can be either a publisher or subscriber. A publisher publishes topics to the MQTT server (broker), while subscribers request topics from the MQTT broker. Every client subscribed to a specific topic receives every message that is published to that topic.

3.3 Custom Wi-Finodes

3.3.1 Processing Unit

All developed Wi-Fi nodes are based on Espressif Systems ESP8266EX chip [35]. ESP8266 is a series of low cost and low power microcontrollers with integrated embedded Wi-Fi module. ESP8266 integrates 32-bit processor, on-chip SRAM, 10-bit A/D converter, antenna switches, RF balun, and other peripheral [35]. ESP8266 could achieve ultra-low power consumption [35] and it is therefore suitable for battery powered devices. Developed Wi-Fi nodes are actually based on the Olimex MOD-WIFI-ESP8266-DEV module. In addition to an ESP8266EX chip, this module also contains 2 MB of SPI flash memory, an integrated PCB antenna, castellated holes for board-to-board soldering, and a small U.FL connector for external antenna connection.

3.3.2 The Hardware

Several Wi-Fi nodes have been developed within the project. The first Wi-Fi node is a temperature/relay node (Fig. 1), that was designed for control of different heating elements (like water heaters, electric radiators, infrared heating panels, etc.). The node contains an Olimex ESP8266-DEV module, Maxim Integrated DS18B20 sensor in the waterproof housing, a 1-channel 16 A relay, AC/DC converter, and other electronic components. The DS18B20 is a digital temperature sensor that provides 9 to 12-bit temperature measurement in °C. The sensor enables temperature measurement in the range of −55 °C to +125 °C, with an accuracy of ±0.5 °C (only in the temperature range −10 °C to +85 °C).

The second developed module is a temperature/humidity Wi-Fi node (Fig. 2). The node contains an Olimex ESP8266-DEV module, temperature and humidity sensor DHT11, an AC/DC converter, and other electronic components. The DHT11 sensor includes a resistive-type humidity measurement component, an NTC temperature measurement component, and a high performance 8-bit microcontroller. The temperature range of DHT11 is 0 – 50 °C and relative humidity range between 20% and 90%.

The last module is a motion detection node (Fig. 3) that contains an Olimex ESP8266-DEV module, Passive Infrared (PIR) sensor, an AC/DC converter, and other electronic components. The PIR sensor is used to measure Infrared (IR) light that radiates from objects in its field of view.

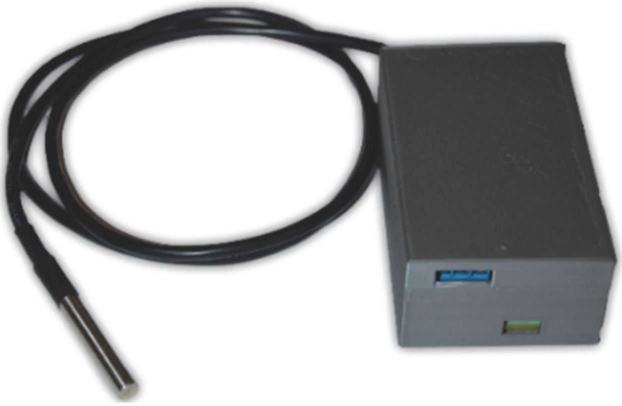


Figure 1 Temperature/relay Wi-Fi node

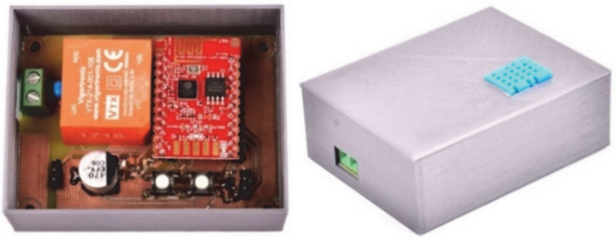


Figure 2 Temperature/humidity Wi-Fi node

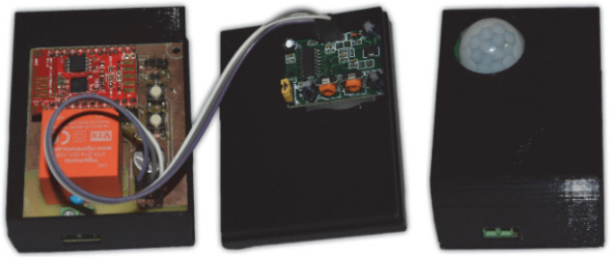


Figure 3 Motion detection Wi-Fi node

3.3.2 The Firmware

The firmware for all Wi-Fi nodes has been developed using C++. At startup, the microcontroller enters into an initialization phase, where the microcontroller initializes local and global variables, the Wi-Fi chip, MQTT objects and other necessary peripherals. In the next phase, the microcontroller connects to the local wireless network, and, when the connection is established, it enters into an endless loop. In this loop, the microcontroller performs the following tasks periodically: (1) Acquires data from the connected sensor(s), processes these data, creates MQTT topic(s), and publishes topic(s) to the MQTT broker and (2) Checks if new data have been received from the MQTT broker, processes the new data, and updates the connected actuator(s). An actual firmware is slightly different, since each module contains a special firmware, which differs according to the used peripheral and the purpose of the node.

Developed Wi-Fi nodes retrieve and generate different MQTT topics (Tab. 3). The motion detection Wi-Fi node publishes "md_node/out/m" topic on a motion detection event. The temperature/humidity node publishes measured temperature ("th_node/out/t") and humidity ("th_node/out/h") periodically (every minute). The temperature/relay node publishes an actual temperature topic ("tr_node/out/t") and receives relay set ("tr_node/in/r") and temperature request ("tr_node/in/t") topics. On a temperature request topic, the Wi-Fi node acquires and publishes the actual temperature, while regarding the retrieved value in "tr_node/in/r" topic, closes or opens relay.

Table 3 Wi-Fi nodes MQTT topics

Wi-Fi node	MQTT topic	Description
Temperature/relay	tr_node/in/r	Close/open relay
	tr_node/in/t	Temperature request
	tr_node/out/t	Measured temperature
	tr_node/out/s	Relay status
Temperature/humidity	th_node/out/t	Measured temperature
	th_node/out/h	Measured relative humidity
Motion detection	md_node/out/m	Motion detected

4 HOME AUTOMATION USING DEVELOPED NODES

Developed Wi-Fi modules have been "integrated" successfully into openHAB and Domoticz home automation system. This section contains an example of the use of developed modules. An example refers to the preparation of hot sanitary water regarding the schedule presented in Fig. 4.

The majority of domestic users use the water heater for sanitary water preparation. Water heaters contain a simple mechanism that maintains the water temperature around the desired temperature throughout the day, since they do not allow setting of different temperatures at different time periods. This solution is energy wasteful, since the warm sanitary water is needed mostly in the morning and evening peaks. To reduce energy consumption, some users utilize programmable plugs, which turn off the water heater in certain time intervals, for example, when they are at work. This solution is relatively cheap and easy; however, it is not useful in cases where some family member stays at home. In this situation, a solution that maintains water temperature at a lower value during the day, and at higher values inside the expected peaks is much more energy efficient. An example of such time schedule of desired water temperature is presented in Fig. 4. In the inside period defined, by t_1 and t_2 (and t_3, t_4), the water temperature would be $T_{\mathrm{H\ (HIGH)}}$, while outside this period $T_{\mathrm{L(LOW)}}$.

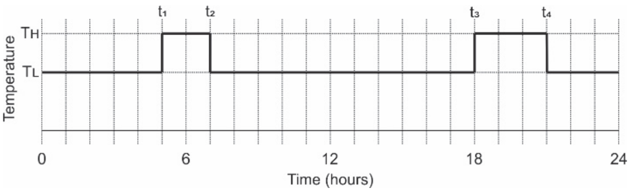


Figure 4 Time schedule of desired temperature

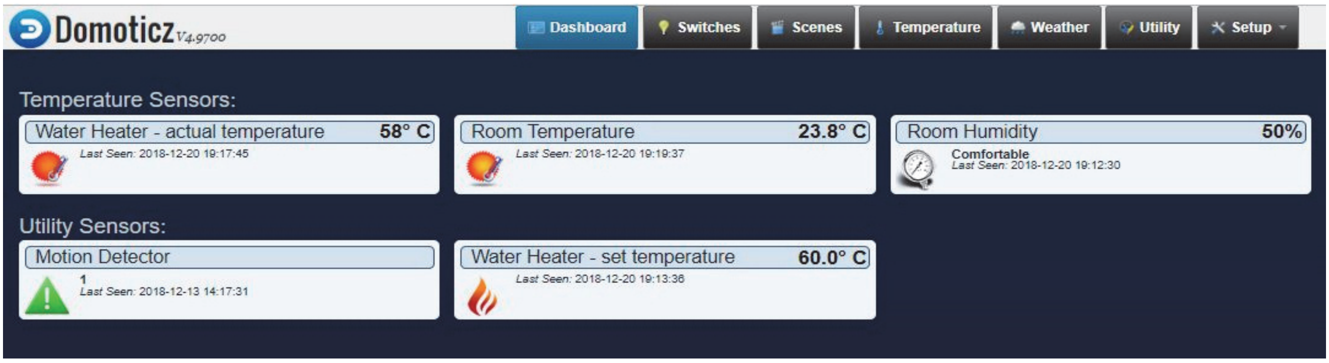
4.1 Home Automation Using Domoticz

Domoticz is an open source and lightweight home automation system that could be installed on different operating systems, and deployed on several embedded systems, such as Raspberry Pi or Synology NAS. Domoticz enables configuration, monitoring and control of miscellaneous devices such as lights, switches and various sensors or meters. It has HTML5 user interface (Fig. 5), which adapts automatically to desktop and mobile devices, and is compatible with the majority of web browsers.

Developed Wi-Fi nodes have been included into the Domoticz environment using a "dummy" hardware option. When such (virtual) hardware exists in Domoticz environment, several different objects can be attached to it and could be configured as switch, slider, display, light, or other type. Created (virtual) hardware is distinguished from others by index ("Idx") number, which is assigned to them automatically during a creation phase. Domoticz GUI (Fig. 5) is accessible using the web browser on the Raspberry Pi internet web address or by using the Domoticz app on Android device.

Communication between the Domoticz and Wi-Fi nodes has been established via the Mosquitto MQTT broker. Domoticz supports only two MQTT topics, i.e. "domoticz/in" for incoming and "domoticz/out" for outgoing MQTT messages. Consequently, only one Wi-Fi node with one input and/or one output topic could be connected to Domoticz. To overcome this limitation, an additional program named Node-RED has been used on Raspberry Pi. Using Node-RED, a transformation of Domoticz MQTT topics into Wi-Fi node topics and vice versa has been achieved.

Node-RED is a flow-based programming tool, developed by IBM's Emerging Technology Services team and is now a part of the JS Foundation. It is composed of two components: (1) Input, output and processing nodes and (2) Flows. Node-RED provides a browser-based editor that makes it easy to create flows using the wide range of built-in and custom nodes. Created flow, that processes data, controls devices, or sends alerts, could easily be deployed to the runtime.



Temperature Sensors:

Water Heater - actual temperature 58° C
Last Seen: 2018-12-20 19:17:45

Room Temperature 23.8° C
Last Seen: 2018-12-20 19:19:37

Room Humidity 50%
Comfortable
Last Seen: 2018-12-20 19:12:30

Utility Sensors:

Motion Detector
1
Last Seen: 2018-12-13 14:17:31

Water Heater - set temperature 60.0° C
Last Seen: 2018-12-20 19:13:36

Figure 5 Domoticz Graphical User Interface

4.1.2 MQTT Communication

The general principle of MQTT communication between Domoticz and Wi-Fi nodes is presented in Fig. 6. Whenever the state of an object in the Domoticz environment changes, "domoticz/out" topic is generated and published by Domoticz software. Node-RED is subscribed to "domoticz/out" topic, therefore, it receives this topic from the MQTT broker and, according to the object index value ("idx"), generates an appropriate topic for the Wi-Fi node ("node/in"), and publishes it to the MQTT broker. Hereafter, the Wi-Fi node receives a "node/in" topic and performs certain action(s). In the opposite direction, Node-RED creates a "domoticz/in" topic, according to the received incoming Wi-Fi nodes topic ("node/out"), and publishes generated topic to the MQTT broker. Since Domoticz is subscribed to "domoticz/in" topic, it updates the state of an object in the Domoticz environment according to the received topic.

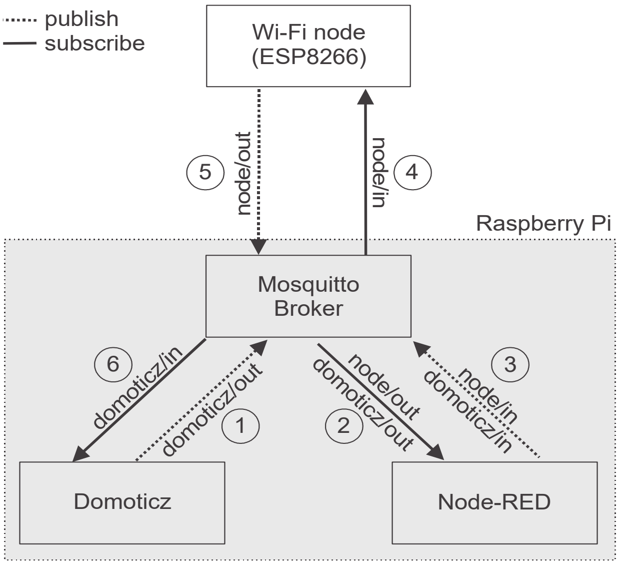
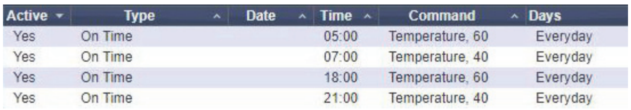


Figure 6 General connection between Domoticz, Node-RED, Mosquitto and Wi-Fi node

4.1.3 Control Algorithm

An algorithm that enables automatic control of water heater is implemented in Node-RED. An algorithm retrieves the desired values from the Domoticz user interface (Fig. 5), and, based on the desired and actual values, controls the water temperature by switching the applied power on and off. The time schedule of hot water presented in Fig. 4 has been implemented using the timer settings of "Water Heater - set temperature" Domoticz object. According to these settings (Fig. 7) the water temperature is set to be 60 °C between 5 AM and 7 AM, and 6 PM and 9 PM, while outside these two periods, the water temperature should be 40 °C.



Active	Type	Date	Time	Command	Days
Yes	On Time		05:00	Temperature, 60	Everyday
Yes	On Time		07:00	Temperature, 40	Everyday
Yes	On Time		18:00	Temperature, 60	Everyday
Yes	On Time		21:00	Temperature, 40	Everyday

Figure 7 Schedule settings in Domoticz

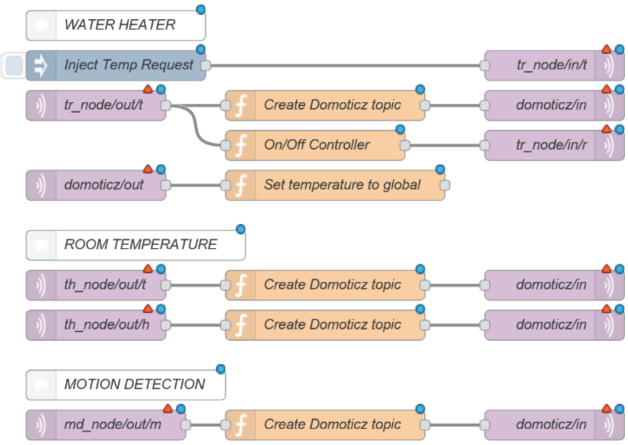


Figure 8 Node-RED flow

The Node-Red algorithm that performs water temperature control is presented in Fig. 8. At each desired water temperature change event, Node-RED retrieves this temperature (T_H or T_L) from the Domoticz environment via "domoticz/out" topic, and stores it into the Node-RED global variable. The "Inject Temp Request" node, periodically (every minute), triggers the flow that generates and publishes the "tr_node/in/t" topic. When the Wi-Fi node retrieves this topic, it measures the actual water temperature and publishes it using a "tr_node/out/t" message. This topic triggers the Node-RED flow that performs simple bang-bang control (On/Off Controller). Based on the desired and actual temperatures and temperature hysteresis, an algorithm calculates, generates and publishes at "tr_node/in/r" topic, which turns the water heater on/off.

5 CONCLUSION

In this paper, smart home solutions that are based on wireless technologies, especially on Wi-Fi network, have been presented. The paper includes an overview of related works in this field, and a case study of cost effective home automation solution that is based on open source home automation software and wireless, custom developed, Wi-Fi based hardware. Presented hardware and software solutions represent a low cost framework for more advanced student projects in the field of home automation systems. Since it is based on a Wi-Fi network, there is no need for additional communication hardware (like ZigBee or Z-wave to Ethernet/Wi-Fi bridge). In most cases, there is also no need for additional Wi-Fi router, since many domestic users use cable modems with integrated Wi-Fi access point.

Developed Wi-Fi nodes have been tested with Domoticz and openHAB software; however these nodes can also communicate with other home automation software that supports MQTT communication. The presented solution enables control of home appliances using only a local network. Access to Internet and cloud services is not necessary, however it is definitely possible.

Using the presented hardware and software solution together with appropriate scheduling algorithms an efficient energy control of home appliances could be achieved. In most cited solutions, the users cannot implement their own control or scheduling algorithms. From this point of view, the presented solution is fairly open. Using Node-RED algorithms (flows), the more advanced users could implement their own control or scheduling algorithms, or even implement more advanced functionalities, like PID or fuzzy logic controllers and machine-learning algorithms.

Acknowledgements

The authors are grateful to the Slovenian Research Agency for partial support of this work through Program Grant P2-0028.

6 REFERENCES

[1] Bhatt, A. & Patoliya, J. (2016). Cost effective digitization of Home Appliances for Home Automation with low-power WiFi devices. Paper presented at *the 2016 IEEE 2nd International Conference on Advances in Electrical & Electronics, Information, Communication & Bio Informatics (AEEICB 2016)*, Chennai, India. https://doi.org/10.1109/AEEICB.2016.7538368

[2] Adib, F., Mao, H. Z., Kabelac, Z., Katabi, D., & Miller, R. C. (2015). Smart Homes that Monitor Breathing and Heart Rate. Paper presented at *the 33rd Annual Conference on Human Factors in Computing Systems (CHI 2015)*, Seoul, Republic of Korea. https://doi.org/10.1145/2702123.2702200

[3] Del Campo, A., Gambi, E., Montanini, L., Perla, D., Raffaeli, L., & Spinsante, S. (2016). MQTT in AAL Systems for Home Monitoring of People with Dementia. Paper presented at *the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2018)*, Valencia, Spain. https://doi.org/10.1109/PIMRC.2016.7794566

[4] Explained Eurostat Statistics. (2017). People in the EU - population projections. Luxembourg: Eurostat. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php/People_in_the_EU_-_population_projections

[5] Portet, F., Vacher, M., Golanski, C., Roux, C., & Meillon, B. (2013). Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing*, *17*(1), 127-144. https://doi.org/10.1007/s00779-011-0470-5

[6] CAO, J., XU, L., Raef, A., & SHI, W. (2017). An OS for Internet of Everything: Early Experience from A Smart Home Prototype. *ZTE COMMUNICATIONS*, *15*(4), 12-22.

[7] Davidovic, B. & Labus, A. (2016). A Smart Home System Based on Sensor Technology. *Facta Universitatis-Series Electronics and Energetics*, *29*(3), 451-460. https://doi.org/10.2298/FUEE1603451D

[8] Rana, M. & Singh, R. (2014). Smart Homes for a better living using Bluetooth communication based on ATmega Microcontroller. *International Journal of Research in Engineering and Technology*, *3*(6), 210-213. https://doi.org/10.15623/ijret.2014.0306038

[9] Vujovic, V. & Maksimovic, M. (2015). Raspberry Pi as a Sensor Web node for home automation. *Computers & Electrical Engineering*, *44*, 153-171. https://doi.org/10.1016/j.compeleceng.2015.01.019

[10] Mowad, M. A. E.-L., Fathy, A., & Hafez, A. (2014). Smart home automated control system using android application and microcontroller. *International Journal of Scientific & Engineering Research*, *5*(5), 935-939.

[11] Kodali, R. K. & Soratkal, S. (2016). MQTT based Home Automation System Using ESP8266. Paper presented at *the 2016 IEEE Region 10 Humanitarian Technology Conference (R10 HTC)*, Agra, India. https://doi.org/10.1109/R10-HTC.2016.7906845

[12] Eclipse Foundation. Eclipse Mosquitto. An open source MQTT broker. Retrieved from https://mosquitto.org

[13] Kodali, R. K. & Mahesh, K. S. (2016). A low cost implementation of MQTT using ESP8266. Paper presented at *the 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Noida, India. https://doi.org/10.1109/IC3I.2016.7917998

[14] Adafruit Industries. Adafruit IO. Retrieved from https://io.adafruit.com/

[15] Kodali, R. K. & Mahesh, K. S. (2017). Low Cost Implementation of Smart Home Automation. Paper presented at t*he 2017 International Conference on Advances in Computing, Communications and Informatics (Icacci)*, Udupi, Karnataka, India. https://doi.org/10.1109/ICACCI.2017.8125883

[16] Maguluri, L. P., Srinivasarao, T., Ragupathy, R., Syamala, M., & Nalini, N. J. (2018). Efficient Smart Emergency Response System for Fire Hazards using IoT. *International Journal of Advanced Computer Science and Applications*, *9*(1), 314-320. https://doi.org/10.14569/IJACSA.2018.090143

[17] Marques, G. & Pitarma, R. (2016). An Indoor Monitoring System for Ambient Assisted Living Based on Internet of Things Architecture. *International Journal of Environmental Research and Public Health*, *13*(11). https://doi.org/10.3390/ijerph13111152

[18] Quintana-Suarez, M. A., Sanchez-Rodriguez, D., Alonso-Gonzalez, I., & Alonso-Hernandez, J. B. (2017). A Low Cost Wireless Acoustic Sensor for Ambient Assisted Living Systems. *Applied Sciences-Basel*, *7*(9). https://doi.org/10.3390/app7090877

[19] Salamone, F., Belussi, L., Danza, L., Ghellere, M., & Meroni, I. (2016). An Open Source "Smart Lamp" for the Optimization of Plant Systems and Thermal Comfort of Offices. *Sensors*, *16*(3). https://doi.org/10.3390/s16030338

[20] Salamone, F., Belussi, L., Danza, L., Galanos, T., Ghellere, M., & Meroni, I. (2017). Design and Development of a Nearable Wireless System to Control Indoor Air Quality and Indoor Lighting Quality. *Sensors*, *17*(5). https://doi.org/10.3390/s17051021

[21] Sprute, D., Portner, A., Weinitschke, A., & Konig, M. (2015). Smart Fall: Accelerometer-Based Fall Detection in a Smart Home Environment. Paper presented at *the Inclusive Smart Cities and e-Health: 13th International Conference on Smart Homes and Health Telematics (ICOST 2015)*, Geneva, Switzerland. https://doi.org/10.1007/978-3-319-19312-0_16

[22] Dey, S., Roy, A., & Das, S. (2016). Home Automation Using Internet of Thing. Paper presented at *the 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON 2016)*, New York City, New York, USA. https://doi.org/10.1109/UEMCON.2016.7777826

[23] Liao, Z. F., Kong, L. Y., Wang, X., Zhao, Y., Zhou, F. F., Liao, Z. N., & Fan, X. P. (2017). A Visual Analytics Approach for Detecting and Understanding Anomalous Resident Behaviors in Smart Healthcare. *Applied Sciences-Basel, 7*(3). https://doi.org/10.3390/app7030254

[24] Dawadi, P. N., Cook, D. J., & Schmitter-Edgecombe, M. (2016). Automated Cognitive Health Assessment From Smart Home-Based Behavior Data. *IEEE Journal of Biomedical and Health Informatics*, 20(4), 1188-1194. https://doi.org/10.1109/JBHI.2015.2445754

[25] Suryadevara, N. K. & Mukhopadhyay, S. C. (2012). Wireless Sensor Network Based Home Monitoring System for Wellness Determination of Elderly. *IEEE Sensors Journal*, *12*(6), 1965-1972. https://doi.org/10.1109/JSEN.2011.2182341

[26] Collotta, M. & Pau, G. (2015). A solution based on bluetooth low energy for smart home energy management. *Energies*, *8*(10), 11916-11938. https://doi.org/10.3390/en81011916

[27] Hernandez, C., Villagran, S., & Gaona, P. (2016). Predictive Model for Detecting MQ2 Gases Using Fuzzy Logic on IoT Devices. Paper presented at *the Engineering Applications of Neural Networks (EANN 2016)*, Aberdeen, UK. https://doi.org/10.1007/978-3-319-44188-7_13

[28] Werapun, W., Kamhang, A., & Wachiraphan, A. (2015). Design of Home Automation Framework with Social Network Integration. *Advances in Digital Technologies*, *275*, 219-226.

[29] OASIS Open. MQTT Version 3.1.1. Retrieved from http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[30] Alvarez-Campana, M., Lopez, G., Vazquez, E., Villagra, V. A., & Berrocal, J. (2017). Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus. *Sensors*, *17*(12). https://doi.org/10.3390/s17122856

[31] Belmonte-Fernandez, O., Puertas-Cabedo, A., Torres-Sospedra, J., Montoliu-Colas, R., & Trilles-Oliver, S. (2017). An Indoor Positioning System Based on Wearables for Ambient-Assisted Living. *Sensors*, *17*(1), 1-22. https://doi.org/10.3390/s17010036

[32] Froiz-Míguez, I., Fernández-Caramés, T. M., Fraga-Lamas, P., & Castedo, L. (2018). Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes. *Sensors*, *18*(8), 1-42. https://doi.org/10.3390/s18082660

[33] Foster, A. (2014). Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper: A Comparison Between DDS, AMQP, MQTT, JMS and REST. In *Messaging Technologies Whitepaper*.

[34] Pereira, C. & Aguiar, A. (2014). Towards Efficient Mobile M2M Communications: Survey and Open Challenges. *Sensors*, *14*(10), 19582-19608. https://doi.org/10.3390/s141019582

[35] Espressif Systems. ESP8266 Datasheet. Retrieved from https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Contact information:

Darko HERCOG, PhD, Assistant Professor
(Corresponding author)
University of Maribor,
Faculty of Electrical Engineering and Computer Science,
Koroška cesta 45, 2 000 Maribor, Slovenia
E-mail: darko.hercog@um.si

Domink SEDONJA, MSc student
University of Maribor,
Faculty of Electrical Engineering and Computer Science,
Koroška cesta 45, 2 000 Maribor, Slovenia

Blaž RECEK, MSc student
University of Maribor,
Faculty of Electrical Engineering and Computer Science,
Koroška cesta 45, 2 000 Maribor, Slovenia

Mitja TRUNTIČ, PhD, Assistant Professor
University of Maribor,
Faculty of Electrical Engineering and Computer Science,
Koroška cesta 45, 2 000 Maribor, Slovenia
E-mail: mitja.truntic@um.si

Bojan GERGIČ, PhD, Assistant Professor
University of Maribor,
Faculty of Electrical Engineering and Computer Science,
Koroška cesta 45, 2 000 Maribor, Slovenia
E-mail: bojan.gergic@um.si