IoT smart home concept

David Vasicek, Jakub Jalowiczor, Lukas Sevcik and Miroslav Voznak, *Senior Member, IEEE*

***Abstract* — In recent years, there has been a tremendous increase in the world of smart devices, including intelligent homes. These allow interaction between people and everyday activities in the home that can be automated. Through sensory activity and data analysis, the household can autonomously respond to situations at home and warn users against possible anomalies and shortcomings. The IoT concept of a smart home described in this paper uses a low energy wireless IoT elements with simple installation and implementation of sensors to create a smart home without a necessity to reconstruct a household. The concept of this project is based on own solution using Google cloud services and it provides an overview of current data and information about the smart home anywhere and anytime in a unique application for Android. Using this application, a user can control the lighting, regulate the brightness of light, control object intrusion by a stranger or protect property from natural elements.**

***Keywords* — Home automatization, Internet of Things (IoT), Smart home, Smart house.**

I. Introduction

NOWADAYS, the term Internet of Thing (IoT) is becoming widely used in the area of Information and communication technology. The primary objective of IoT is an interconnection of electronic devices, systems and services for the purpose to provide more data that can be converted into information and information converted to the knowledge that can be subsequently applied [1]-[7]. From the knowledge gained in this way, the systems can make decisions and autonomously perform activities. In practice, it means that the more devices can provide data about the real world, the more available data that can be analyzed we get and the more knowledge that can be used.

The IoT can be divided into two categories. The first category is formed of Industrial Internet of Things (IIoT) into which belongs, for example, industrial automation, transport industry, power industry and healthcare. The second category is formed of Consumer Internet of Things (CIoT) that is focused primarily on smart home, smart shopping or payment for goods and services using the NFC included in the mobile phone.

This paper is mainly focused on the second category of the IoT, the Consumer Internet of Things, more specifically on smart home applications. These applications are very popular among users especially because they allow to automate activities in the household that the users would otherwise have to take care of ourselves. They can, for example, control air condition, regulate heating in individual rooms, manage swimming pool heating, care of aquariums, control lighting in rooms with the ability to control its intensity, operate alarm or mechanical lock setting. If you are not sure whether you turned off an iron or other electric appliance in the morning rush, you can remotely disconnect it from your socket by using your mobile phone, tablet or computer.

These applications can excite their users not only with the convenience they offer but also with their low price. Nowadays, it is possible to buy devices that are capable of integrations with IoT, whose price does not exceed several ones or tens of United States dollars.

II. State of the Art

Home automation is one of the most common applications in the world of IoT, and not only a lot of scientific works and institutions are concerned, but also many companies that offer tailored solutions. Unlike expensive commercial solutions where the price of the sensor can run into several tens of United States dollars, open-source projects offer much cheaper solutions.

One of the open source applications is the Frugal Labs IoT Platform (FLIP) [8], whose architecture consists of the FLIP control board, the Raspberry Pi-based gateway, custom Cloud solutions and web applications. The control board is based on Atmel's ATmega328p [9] microchip with expandable WiFi and Bluetooth communication modules. The control board supports a total of 6 sensor types that can be managed by web applications that are used to control FLIP devices and to acquire sensor data.

There are also other home automation companies (Xiaomi, the Smart Home Kit [10], Samsung SmartThings [11]) that are focused on developing their default gateways and sensors that can be managed by mobile applications. The advantage of these projects is the implementation of a smart household without more in-depth technical knowledge.

This is not the ultimate list of all products and projects

This experimental development received a financial support from student grant SGS SP2018/59 - Networks and Telecommunications Technologies for Smart Cities, conducted by the VSB-Technical University of Ostrava, Czech Republic and partially this work was supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project „IT4Innovations National Supercomputing Center – LM2015070“.

David Vasicek and Miroslav Voznak are with Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic (e-mail: david.vasicek1@vsb.cz, miroslav.voznak@vsb.cz).

Jakub Jalowiczor is with CESNET z.s.p.o, Zikova 4, 16000 Prague, Czech Republic (e-mail: jalowiczor@cesnet.cz).

with Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic (e-mail: jakub.jalowiczor@vsb.cz).

Lukas Sevcik is with IT4Innovations, VSB–Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic (e-mail: lukas.sevcik@vsb.cz).

978-1-5386-7171-9/18/$31.00 ©2018 IEEE

on the market. The popularity of smart homes is growing in a huge way. However, most open source applications offer the user only a small portfolio of integrable sensors, low depth of logic, and require users to have the technical skills to integrate the smart home. The project we are explaining in this paper describes the implementation steps and benefits of an open source concept that also combines home automation. Compared to other projects, it extends the portfolio of new sensors that can be easily integrated into the existing solution as well as a broader implementation of the automation logic that the users can define themselves. In a more profound logic, individual sensors can communicate with each other, making automation much smarter.

III. Architecture of Network

Home sensor network described in this paper is network based on own solution. Infrastructure contains end-node devices for information collection, default gateway, commercial cloud solution Firebase real-time Database and end applications to visualize results, which are interconnected with the TCP/IP protocol. This architecture is described in figure 1.

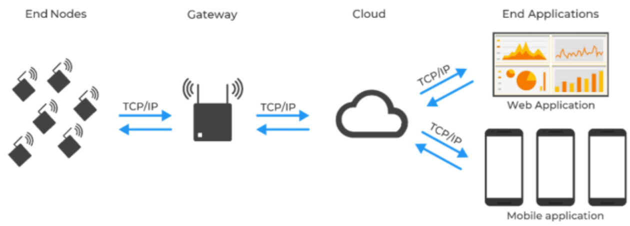

Fig. 1. The architecture of the smart home concept from end-device to end-application.

A. *End-Nodes*

End-node devices are based on low-cost and low-energy microcontrollers ESP8266 [12] and ESP32 [13] with integrated Wi-Fi standard IEEE 802.11 b/g/n [14]. These microcontrollers communicate via TCP/IP protocol with default gateway in ISM zone 2,4GHz. The main task of these end-devices is not to understand and process the data but collect data from sensors and resend these data to a default gateway.

Each end-device is required to be configured for integration into an IoT home network. This is done wirelessly via an Android mobile phone with the special application. After connection to the electrical network, end-devices work in the AP mode and transmit own specific SSID code in the form of IoT_[sensor type] (for example IoT_TMPH where TMPH means temperature and humidity sensor or IoT_PLNT, where PLNT means a Plant Soil Moisture sensor). The mobile application gradually scans neighbour Wi-Fi networks and separates them from networks with the specific SSID codes. Depending on the sensor type in SSID code, the configuration form for the sensor is offered to the user. The configuration form includes, for example, SSID and network password, to which the end device is supposed to connect.

For example, if we look at the configuration form for a plant soil moisture sensor, the configuration form would require information from the user about the plant name, room location, watering interval or information about light intensity which a plant requires. Then the mobile application automatically assigns an ID to the sensor. When the configurational form is completed by a user, the configuration is JSON packaged and sent to a sensor, which receives and processes the JSON object.

After unpacking the packet, the end-device tries to connect to the home network to which the user assigned access data. Once the end-device connects to a home network, it switches from the AP mode to client mode, and the next device registration is automatic.

After connection to a wireless network and assignation of an IP address, a broadcast message “Hello IoT server” is sent, throughout which the end-device searches a default gateway in a network. If the default gateway accepts such a message, it sends a message with its IP address on the network to the sensor. Thereafter, registration JSON object is sent to this address, which contains information that the user filled in the configurational form. Thanks to this the gateway can respond autonomously to the data that it receives from sensors and inform a user about a need of watering a flower or a need of relocation to the sunniest place.

Thanks to the usage of low-energy microchips ESP8266 and ESP32, the devices may operate on a battery with 3.3 voltage for a couple of months. The micro-chips report consumption up to 10 μA during a deep sleep mode and in a transmission mode 120160 mA with the standard IEEE 802.11n and transmitting power POUT = +13dBm.

Table 1: Sensors integrated into IoT smart home concept.

Sensor	***Obtained values***
Soil moisture sensor	Temperature and humidity around the plant, ambient lighting and soil moisture
Temperature and humidity sensor	Temperature, humidity and barometric pressure
Light Intensity sensor	Ambient lighting level
Rainwater level sensor	Rainwater level
Windows/Door sensor	Window/door open status
Passive infrared sensor (PIR)	Motion detection status
Fire detector sensor	Fire detection status
Co2 sensor	Carbon dioxide level
Flammable gases sensor	Flammable gases level, such as LPG, butane, methane, alcohol, propane, hydrogen, etc.
Noise sensor	Noise intensity level

Many types of existing sensors have been tested and integrated into this IoT concept, the Table 1 gives a summary of all tested sensors.

B. Gateway

The gateway forms the bridge between end-nodes and real-time database Firebase, and it is built on Raspberry Pi [15] device with a system Raspbian [16]. The main task of this gateway is receiving, processing and transmitting of UDP datagrams from end node devices to real-time Firebase database and also to internal MySQL database working on the default gateway. The default gateway is not just the gateway itself, but it also serves as a server for providing the necessary services and programs, which are subsequently provided to clients.

The task of the gateway is, therefore, data collection, data storage, data analysis and results sharing. Whereas the all data is processed by the gateway before a handover, it has an overview of the individual data. Therefore, the gateway is able to react and respond to any stimulus in the home network. The gateway may, for example, create security disruption notifications, contact the emergency system in case of the outbreak of the fire, inform users about open windows or lights or automatically set the thermostat in the room.

The gateway is adapted to integrate new sensors into the home network and can deal with even unknown sensor types. It is capable of serving hundreds of sensors operating in the home network and communicating with them in downlink and uplink direction.

The program that processes all data from end-devices is written in JavaScript and runs in a Node.js environment. This allowed easily implement services from Firebase corporation to the server that fully support the Node.js [17]. This is primarily about a real-time database and services for sending Firebase cloud messaging notifications. The program takes care of checking the lifetime of the individual sensors. In the parallel thread, it checks if all registered end-devices in-home network works correctly. If any of the sensors is inactive for more than 30 minutes, the program changes the status of the sensor from online to offline and notifies the system administrator about this change throughout a notification message.

C. Cloud solution

Firebase cloud services by Google Inc. were selected for the concept of a smart home, specifically Firebase real-time database and Firebase cloud messaging. These services were chosen for their full support of JavaScript or C++. Therefore, Firebase can run alone on microchips ESP8266 and ESP32.

Firebase real-time database store and sync data with NoSQL cloud database. Data is synced across all clients in real-time and remains available when the app goes offline. The Firebase real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with iOS, Android, and JavaScript SDKs, all of the clients share one real-time Database instance and automatically receive updates with the newest data. [18]

Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that lets reliably deliver messages at no cost. Using FCM can notify a client app that new data or important information are available. [19]

D. End mobile application

Data from the end-devices are processed by the default gateway and visualized in a single and unique mobile application. This Smart Home application enables a home automation system and allows users to control their home devices remotely via smartphone. In just a few clicks, access to the Smart home services or data: thermostat, smart sockets, lights, temperatures or security from any-where and any-time.

The mobile application supports Android OS 4.0.3 and above and is best optimized for the Samsung Galaxy S series, Huawei, Honor, LG, Xiaomi and Nokia. Some features may not be supported in different smart devices.

Mobile application synchronizes data from all devices in real time. It is able to respond to information change in databases and to an addition or removal of the sensors in the database. In practice, if the user changes the state of a smart socket from OFF to ON, this change will occur in a real-time in all devices that are logged in to the same real-time database.

For users' comfort, it is possible to distinguish different categories of sensors according to the room where the sensors are located, or we can choose an individual room and all sensors registered to the room are displayed. These sensors are automatically registered to the application. If a new sensor is added and properly registered to the smart home, it is automatically synchronized with all mobile devices.

A mobile application allows results visualization via line charts. The visualization serves for example, for better information about a room temperature progression at a certain time. The time interval may be set by the user for one day, one week, one month or one year.

E. End web application

Similarly, to the mobile application, the web application is used to visualize user data acquired from sensors and to manage the device. When designing the application, it was put emphasis on user-friendliness and simple design and to create an application that looks similar to the mobile application.

A web application is created in PHP language, thanks to which the application is able to task database server MySQL via scripts. In this database, all data from sensors that the gateway has received are stored permanently. The web application is able to offer the user a view of the user data with graphical outputs in a larger spectrum than mobile application does.

An administration template created by popular framework Bootstrap that uses HTML, CSS and JS were selected for the web application. The framework is designed to develop responsive pages that may be viewed on mobile and desktop devices with web browser support. The used template is offered for free, and it allows you to speed up your own application development.

IV. Conclusion

The paper introduces the concept of a smart home that can be easily adapted to the real house. The solution described in this paper is used to deploy security mechanisms, smart lighting, energy savings, and other features that can be remotely controlled and managed. All features of this house are implemented through one interface.

The mobile application is available for all mobile phones with Android operating system since version 4.0.3 with Internet access. Because the application interface is very easy to understand, this project can be used by both system engineers and people with low technical knowledge.

The market offers a plenty of solutions providing similar functionality as the concept, advantages here are an open-source development, extensibility of individual sensors in the form of gadgets that can be implemented in a home network without the need for complex reconstruction and modernization of wiring, and the solution is based on open-source systems. The sensor only needs to be turned on in the local home network and the data is immediately available. After that, it is only necessary to define logic.

This project is an early stage of development, and as a further development step, it expects security extension and improvement, integration of new sensor types, an extension of battery life, extension of applications for other platforms and it is also expected implementation of neural networks for an even smarter house.

References

[1] A.F. Santamaria, P. Raimondo, N. Palmieri, M. Tropea, F. De Rango, "Cooperative video-surveillance framework in internet of things (IoT) domain," Internet of Things, vol. Part F2, 2019, pp. 305-331, 2019.

[2] L. Kvarda, P. Hnyk, L. Vojtech, M. Neruda, "Software implementation of secure firmware update in IoT concept," Advances in Electrical and Electronic Engineering, vol. 15 (4 Special Issue), pp. 626-632, 2017.

[3] A.F. Santamaria, F. De Rango, A. Serianni, P. Raimondo, "A real IoT device deployment for e-Health applications under lightweight communication protocols, activity classifier and edge data filtering," Computer Communications, vol. 128, pp. 60-73, 2018.

[4] L. Kvarda, P. Hnyk, L. Vojtech, Z. Lokaj, M. Neruda, T. Zitta, "Software implementation of a secure firmware update solution in an IOT context," Advances in Electrical and Electronic Engineering, vol. 14 (4Special Issue), pp. 389-396, 2016.

[5] F. De Rango, D. Barletta, A. Imbrogno, "Energy aware communication between smart IoT monitoring devices," in Proc. the 2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2016 - Part of SummerSim 2016 Multiconference, art. no. 7570517, 2016.

[6] A.F. Santamaria, P. Raimondo, F. De Rango, A. Serianni, "A two stages fuzzy logic approach for Internet of Things (IoT) wearable devices," in Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, art. no. 7794563, 2016.

[7] H. Nguyen, H. Dang, S. Le, T. Le, D. Do, M. Voznak, M and J. Zdralek, "Enabling D2D transmission mode with energy harvesting and information transfer in heterogeneous networks," Advances in Electrical and Electronic Engineering, 16 (2), pp. 178-184, 2018.

[8] "Frugal Labs IoT Platform". [Online]. Available: http://myflip.io/. [cit. 2018-10-05].

[9] "ATmega328/P, DATASHEET COMPLETE". [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. [cit. 2018-10-05].

[10] "Xiaomi Mi Smart Home". [Online]. Available: https://xiaomi-mi.com/mi-smart-home/. [cit. 2018-10-05].

[11] "Samsung Smartthings: Smart Home App for Home Automation". [Online]. Available: https://www.samsung.com/us/explore/smartthings/ [cit. 2018-10-05].

[12] "ESP8266EX Datasheet". [Online]. Available: https://www.espressif.com. [cit. 2018-10-05].

[13] "ESP32 Series Datasheet". [Online]. Available: https://www.espressif.com. [cit. 2018-10-05].

[14] "WG802.11 - Wireless LAN Working Group". [Online]. Available: https://standards.ieee.org. [cit. 2018-10-05].

[15] "Raspberry Pi Documentation". [Online]. Available: https://www.raspberrypi.org/. [cit. 2018-10-05].

[16] "Raspbian". [Online]. Available: https://www.raspberrypi.org. [cit. 2018-10-05].

[17] "API Reference Documentation". [Online]. Available: https://nodejs.org/en/docs/. [cit. 2018-10-05].

[18] "Firebase Realtime Database". [Online]. Available: https://firebase.google.com/docs/database/. [cit. 2018-10-05].

[19] "Firebase Cloud Messaging". [Online]. Available: https://firebase.google.com/docs/cloudmessaging/. [cit. 2018-10-05].