High-Confidence Computing 2 (2022) 100036

Contents lists available at ScienceDirect

High-Confidence Computing

homepage: www.elsevier.com/locate/hcc

Access control and surveillance in a smart home

Cristina Stolojescu-Crisan [a,*], Calin Crisan [b], Bogdan-Petru Butunoi [c]

[a] Politehnica University of Timisoara, Romania
[b] SafeFleet Telematics, Timisoara, Romania
[c] West University of Timisoara, Romania

ARTICLE INFO

Keywords:
Access control
Internet of things
Security
Smart homes
Video surveillance

ABSTRACT

In the past years smart homes solutions have become more and more popular with the introduction of a high number of both Internet of Things (IoT) applications and smart devices. Home automation and security systems market is in a continuous growth, traditional security systems are evolving fast, and more and more people choose Smart home solutions. In this paper, we propose two IoT based systems in the context of Smart homes: qToggle for multiple home automation, and MotionEyeOS, a video surveillance OS for single-board computers. Most qToggle devices are based on ESP8266/ESP8285 chips or on Raspberry Pi boards and smart sensors, while MotionEye uses Raspberry Pi boards.

1. Introduction

The Internet of Things (IoT) [1,2] represents a new paradigm in computing, in which billions of smart devices that are connected to the Internet, directly communicate with each other and with the users. IoT uses the Internet Protocol (IP) to connect devices, which include smartphones, tablets and digital assistants to various types of sensors, appliances, and systems such as lighting, temperature, or security. The characteristics of the IoT are very wide and include a variety of physical elements, as shown in Fig. 1 [3].

In the last years, the IoT concept has had a strong evolution, being currently used in various domains such as smart homes, telemedicine, industrial environments, etc. The associated network can bring several challenges in the applications development process. The IoT is not a single technology, but it is a concept in which new things are connected and enabled, and this brings many business opportunities [3]. According to Intel [4] the IoT world is continuously growing, from two billion objects in 2006, to a projected 200 billion by 2020. This means around twenty-six smart devices for every human being on Earth.

Intelligent Building Management Systems (BMS) represent an important part of the IoT paradigm that aims to control and monitor the mechanical and electrical equipment of a building. Allowing objects and devices in a building to be connected to the Internet enables users to remotely monitor and control them. From light switches that can be controlled to turn on and off by using a smartphone, tablet, laptop or by voice command, to thermostats that will adjust the indoor temperatures, access control, or air conditioning on/off switching, intelligent BMS solutions have become popular in the last years and more nowadays. Smart home aims to integrate home automations. Home automation represents a smart solution that offers several advantages, including comfort, security, energy saving, money saving in the long run, convenience, and peace of mind. Fig. 2 shows an example of an intelligent building employed with different IoT connected utilities. Although, it presents only a few examples, there are many more applications that can be implemented, upon request.

Safety and home/building security are the most important concerns. Integrating security and access control systems into a building automation architecture provides the ultimate security, while adding a new level of convenience to a user's lifestyle. Smart solutions allow the alarm, access control, lighting, and cameras to be managed and controlled remotely, using, for example, a smartphone and an app. One reason for the growth of smart technologies is to avoid the increasing risk of burglary and accidents. Because most people have usually a busy lifestyle, the need to monitor and control the status of their home remotely has become a big issue.

A suite of IoT based smart home automation systems are described in the literature or already available on the market [5–16] The combination of technology and setup difficulty associated with each on them varies, as do their pros and cons regarding complexity, costs, performances, and so on. The IoT plays a significant role in implementing security features in smart buildings [5]. Classic security systems involve having an alarm that would turn on and sound when someone breaks in. A smart security system is meant to do much more than that: it can alert the owners by sending a SMS alert on their smartphone, and the owners are able to arm and disarm the alarm remotely, using an app on their smartphones. Smart security systems have been developed over the years, using various technologies. For example, Bluetooth [9] based

* Corresponding author.
E-mail address: cristina.stolojescu-crisan@upt.ro (C. Stolojescu-Crisan).

https://doi.org/10.1016/j.hcc.2021.100036
Received 4 March 2021; Received in revised form 7 May 2021; Accepted 27 August 2021

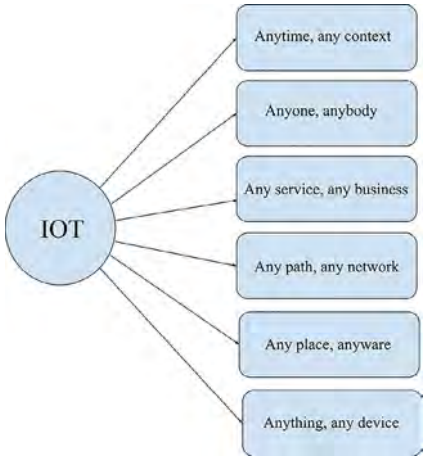

Fig. 1. Characteristics of the IoT.

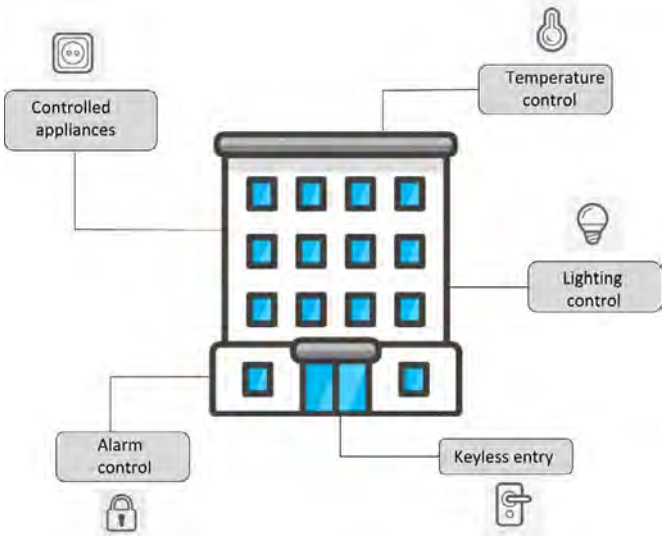

Fig. 2. An IoT based intelligent building depicting the use of smart sensing devices for different purposes.

automation is low cost, fast, and easy to be installed, but it is limited to short distances. Zigbee [13,14] is a wireless mesh network standard, designed to be low-cost and with a low power consumption, targeted at battery-powered devices in wireless control and monitoring applications. However, it has a low data speed, low transmission, as well as low network stability, and has a high maintenance cost. Wi-Fi technology is used in Syafaah et al. [7], Jabbar et al. [10]. Low-cost, open source hardware components, like Arduino and Raspberry Pi microcontroller unit (MCU) boards, and a combination of sensors have been very used in the home automation domain. Home automations using Arduino boards are proposed in Shafana and Aridharshan [6], Nayyar et al. [15], Wadhwani et al. [16]. Arduino is highly flexible, open source, not expensive, and easy to program. In addition, the existence of a large and active community of users is a great plus. However, Arduino is not designed to handle the large complexity that comes with advanced projects. For more advanced and real time projects, Raspberry Pi is a better option. Raspberry Pi is an exciting technological development, much cheaper than any desktop computer or mobile device. Most of the software and projects done on Raspberry Pi are open source and are maintained by online user communities, always excited about new projects. When developing software on Raspberry Pi, Python is the language of choice, since it is relatively simple (fewer lines and less complexity) compared to other programming languages. Besides its low price, Raspberry Pi is energy efficient and does not require any cooling systems. Smart home automations with Raspberry Pi are proposed in Patchava et al. [8]. ESP8266 chips are low price Wi-Fi modules, perfectly suited for projects in the IoT field. ESP8266 is a single core processor that runs at 80 MHz. ESP8266 chips were used for home automations related projects in Syafaah et al. [7]. GPRS/GSM were used in Zhao and Ye [11], Felix and Raglend [12].

Using a smart system permits the following scenario: on approaching home a user opens the mobile app on the smartphone or smartwatch, presses a button to open the gate, the garage door, and the main entrance door, disarms the alarm, and switches on the lights. The automation system proposed in this paper can perform all these tasks.

The first proposal of this paper is qToggle, a system for interconnecting sensors, actuators, and other data sources, with the purpose of multiple automations. qToggle aims to propose a standard that allows managing, provisioning, and talking to different devices. qToggle devices are based on ESP8266/ESP8285 chips and on Raspberry Pi boards. Also, we have developed a user-friendly smartphone application that allows users to control a series of home appliances and sensors. The second proposal is MotionEye, a video surveillance OS for single board computers. It is a great solution to build surveillance systems, because it is simple to install and has a web-based, user-friendly interface that can be used in any browser.

Security systems can be developed using various technologies and electronic equipment, each of them with their advantages and disadvantages regarding complexity, performances, cost, and so on. We have studied the market and the available literature, and we tried to develop a system that is simple, cheap, but performant. The contributions of this paper are hardware and software. Both hardware and software parts of project have been designed, installed, and tested by the authors. The proposed security system (including access control and alarm) is just a small part of the qToggle project, which was thought to be a complex automation project, with many functionalities that could transform a normal house into a smart home. We intend to make qToggle a serious candidate in the automations market, in the future. In this paper, we present a real case study (a real home) and all the features the proposed systems, qToggle and MotionEye (including the apps), offer to make life easier and cheaper. The proposed solutions can be implemented by any user using the codes available on Github and the paper offers a description on how the system is implemented, how the app can be installed, and what functionalities are covered in order to have a smart home.

2. Access control and security system. QToggle

For the purpose of access control and security, this paper proposes a simple and flexible solution, called qToggle, that can be used for controlling devices in a house or building. qToggle aims to propose a standard that allows managing, provisioning and talking to different devices. qToggle does not attempt to reinvent the wheel, but it makes use of the existing and widely used technologies. The classic Ethernet and/or a Wi-Fi local network are usually enough for a working qToggle setup. Most qToggle devices are based on ESP8266/ESP8285 chips or on Raspberry Pi boards, both choices being cost effective, small and easy to work with. Keeping the device firmware up to date represents an essential task, that is often neglected when dealing with large fleets of devices. qToggle facilitates this task by allowing updates of the firmware in a common way for different types/models of devices. The idea behind qToggle is to control programmable systems having a TCP/IP stack via simple HTTP requests. These systems can be, for example, single-board computers or TCP/IP-enabled microcontrollers. API specifications seem complex, offering many functionalities and use cases. Yet, most of them are optional, only a small set of functions being mandatory for implementing qToggle.

The Raspberry Pi is an open source hardware technology which combines a programming language and an Integrated Development Environment (IDE) [17]. Raspberry Pi is an exciting technological development, much cheaper than any desktop computer or mobile device [19,18].

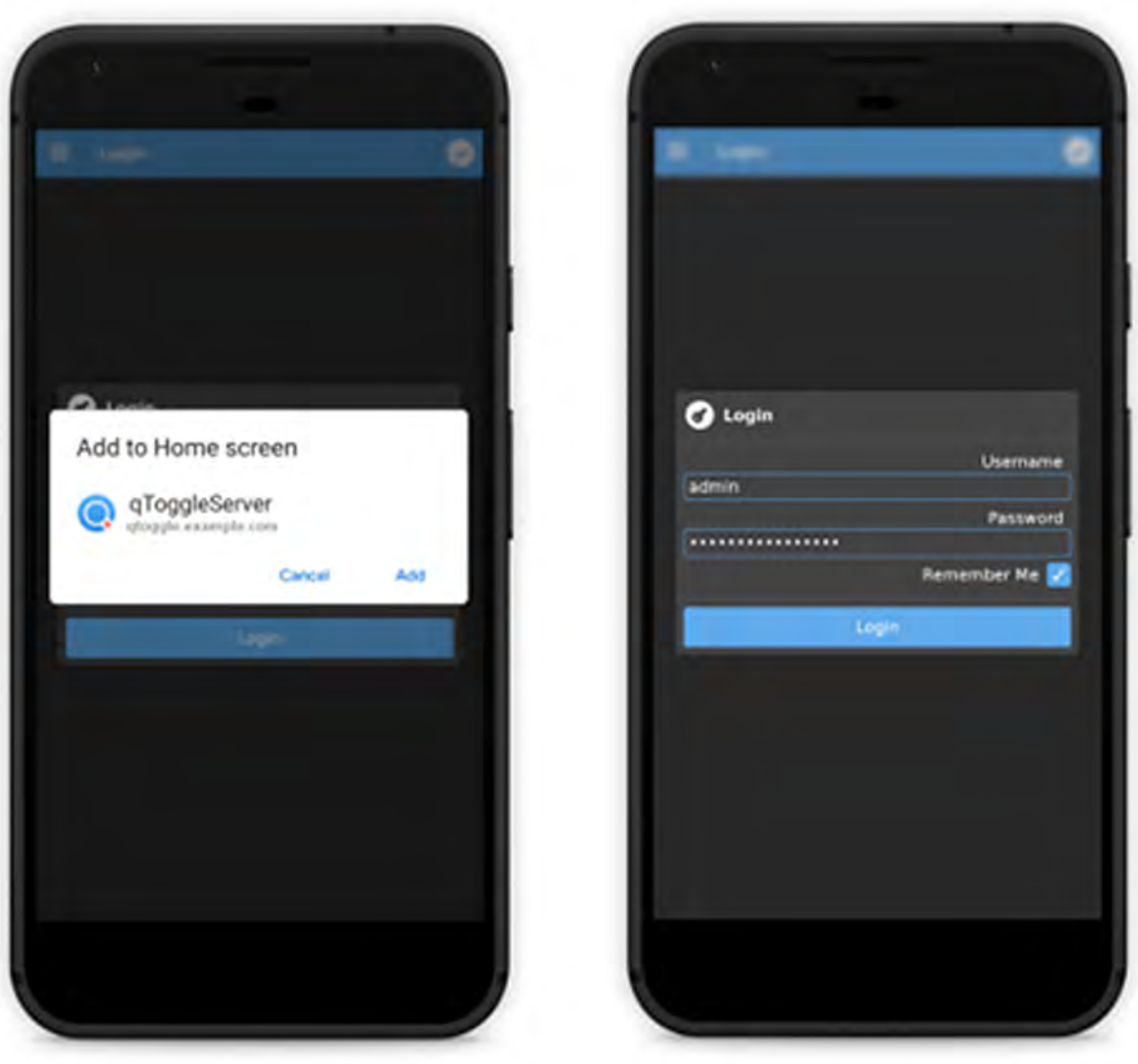

Fig. 3. The login process on qToggleServer.

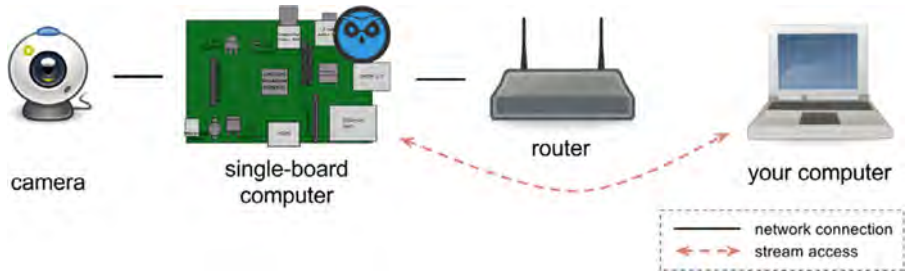

Fig. 4. A single device with a single camera scenario.

Most of the software and projects done on Raspberry Pi are open source and are maintained by online user communities, always excited about new projects. When developing software on Raspberry Pi, Python is the choice, since it is a relatively simple language, with fewer lines and less complexity, as compared to other programming languages. Besides its low price, Raspberry Pi is energy efficient, and does not require any cooling systems. The Raspberry Pi version used for this project is Raspberry Pi 4, due to the improvements brought as compared to previous versions. The Raspberry Pi board, in a qToggle setup, may have three roles: first, the board could act as a qToggle device, when it is equipped with peripherals such as sensors or relay boards, it could also act as a master hub for other qToggle devices and, finally, it could help installing the ESP firmware on some devices, when running Tuya Convert OS. Tuya is a Chinese smart devices platform that offers cloud services for ESP8266/ESP8285-based devices. Tuya Convert OS helps replacing this proprietary Tuya firmware with a custom firmware, without disassembling the device.

The main part of the home automation system based on the IoT is the microcontroller. The ESP8266 Wi-Fi module [20] represents a set of low-cost, performing, highly integrated wireless System on Chip (SoC), that provides a complete and standalone Wi-Fi network client. ESP8266EX version is one the most integrated Wi-Fi chip in the indus-

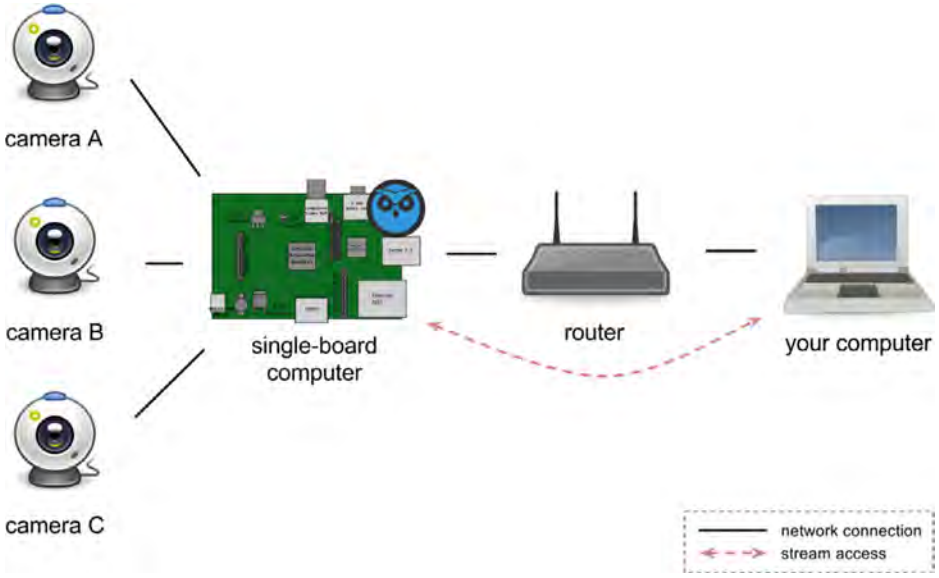

Fig. 5. Single device with multiple cameras scenario.

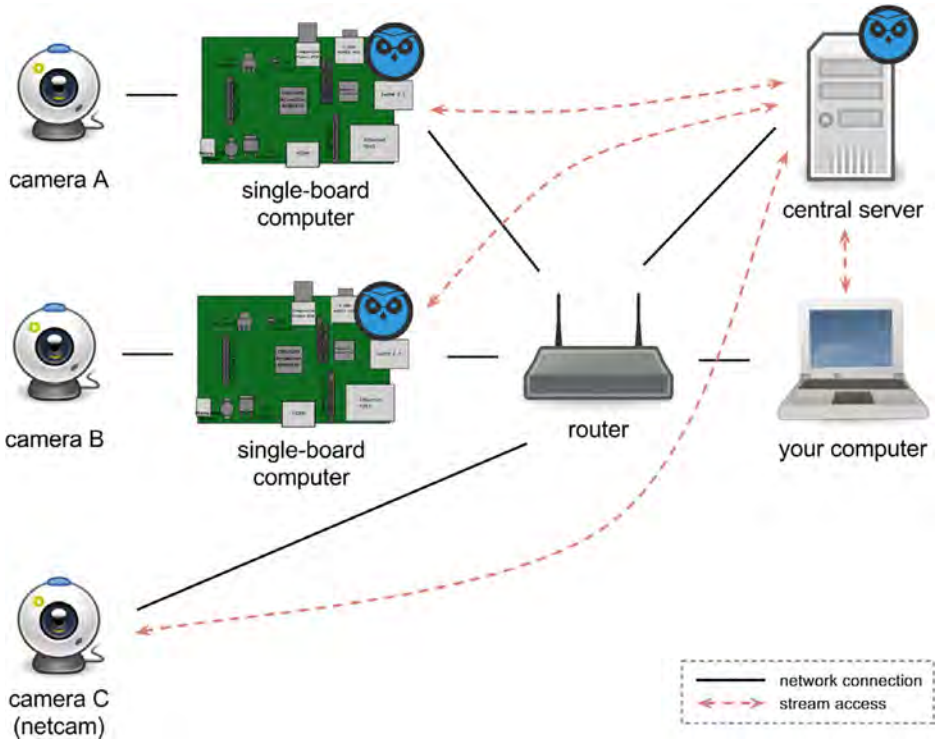

Fig. 6. Multiple devices with a hub scenario.

try. Besides its Wi-Fi functionalities, ESP8266EX integrates an enhanced version of L106 Diamond series 32-bit processor from Tensilica, with on-chip SRAM. ESP8266EX has seventeen GPIO pins, which can be assigned to various functions by programming the appropriate registers, two power pin, one ground pin, reset pin, and two clock pins.

The proposed automation system can be monitored and controlled very easy, using a mobile app (and its associated ecosystem). It provides a friendly user interface, called frontend, which comes in the form of a progressive web application (PWA). qToggle is designed to be used on smartphones, tablets and just as well on laptops/desktop machines. qToggle app is linked with the qToggleServer package. This means that users will get an app update whenever they update their qToggleServer installation. Since qToggle is a web app, the update process is done automatically by the browser, when the user reopens or refreshes the app. The user can either close it and reopen it, or he/she can use the pull-to-refresh function to make sure the app is up to date. The code and documentation for qToggle can be found on Github. qToggle has been designed as a way of interconnecting sensors, actuators and other data sources with the purpose of multiple automations. It is in fact an extensive automation project. Firstly, the app should be installed and, being a PWA, it should be simply added to the home screen of the mobile phone. After installation, qToggle app will be found through the apps list of the device, and it can be uninstalled any time the used wants to. When prompted for login for the first time (see Fig. 3), admin with

empty password should be used, but it is highly recommended to set proper passwords in the Settings page of the app, for obvious security reasons. In panel edit mode, the user can add, move around, resize, remove and configure widgets. Widgets usually require selecting one or more ports, whose values will be displayed and/or changed by the widget upon interaction.

The ports section is only accessible to administrators and allows adding, removing and configuring ports. Also, only the administrators are allowed to add, remove and configure slave devices.

3. Video surveillance. MotionEye

Not long ago, having a home video surveillance system for security reasons was seen as an extravagance. Surveillance cameras were used mostly by government agencies, parking lots, or enterprises. Nowadays, with the advancements of technology, surveillance systems have become more affordable and accessible. Thus, having video cameras installed outdoors seems like a normal decision to make in a residential home. Surveillance cameras bring a series of benefits, one of the most important being that visible outdoor cameras may reduce, or even avoid, thefts and vandalism. In most cases, when burglars spot cameras, they will most likely give up on the burglary attempt. Moreover, in case of a burglary, the cameras will record the incident, and will help the authorities to capture the criminal. Having a video surveillance system and a mobile app also offers the possibility to monitor children when playing in the yard or in front of the house, or to see when they have arrived home from school. Also, monitoring the activity in the yard, including pets, when nobody is at home brings peace of mind.

Digital video recorder (DVR) and Network video recorder (NVR) security systems represent the most important choice before setting up a video surveillance system, since they influence the cost, the installation, and the expected video quality. NVRs record videos from IP cameras, from the network, while DVRs have a small chip inside to encode and convert the analog videos into digital format. This way the user is able view and playback the recordings. The main difference between NVR and DVR is the type of cameras and cabling used by them: a NVR is used with IP cameras via wireless or using Ethernet cables, while a DVR records analog cameras using coaxial cables or Ethernet cables using baluns. NVR security systems record higher quality videos, are easy to wire, and are easy to use and configure. Moreover, a NVR uses IP network cameras, and thus can be placed virtually anywhere, as long as the network is available. Hence, our choice was, obviously the NVR system. While USB cameras are not technically network cameras, they are usually associated with NVRs since they transmit video data in a digital format.

MotionEye is the web user interface, also called frontend, that a user accesses in the browser to manage the video cameras. MotionEyeOS is the operating system that uses motionEye as a frontend and turns single board computers into a video surveillance system. The OS is based on BuildRoot and uses motion as a backend and motionEye for the frontend. In this context, a device is a computer that runs motionEye or motionEyeOS. A device can have zero, one, or more cameras attached. A hub is a device configured to remotely manage other motionEye devices and usually has no cameras attached. A network camera (IP camera) is a camera that can stream over the IP protocol. Network cameras can be either standalone camera units designed to stream over IP, or they can be motionEye-based devices configured for streaming. MotionEye offers the following features: a web-based mobile and tablet friendly user interface, compatibility with most USB cameras, as well as with the Raspberry PI camera module, support for IP network cameras, motion detection with email notifications and working schedule, JPEG files for still images, AVI files for videos, time lapse movies, and uploading media files to Google Drive and Dropbox. Raspberry Pi boards are the immediate, obvious choice when it comes to the hardware supporting motionEye and motionEyeOS. Nevertheless, their limited performance becomes visible when dealing with more than 4–6 cameras running at higher resolutions and/or framerates. For users with a larger number of cameras, we recommend devices from the ODROID family. They represent a series of single-board computers, running Android and Linux distributions.

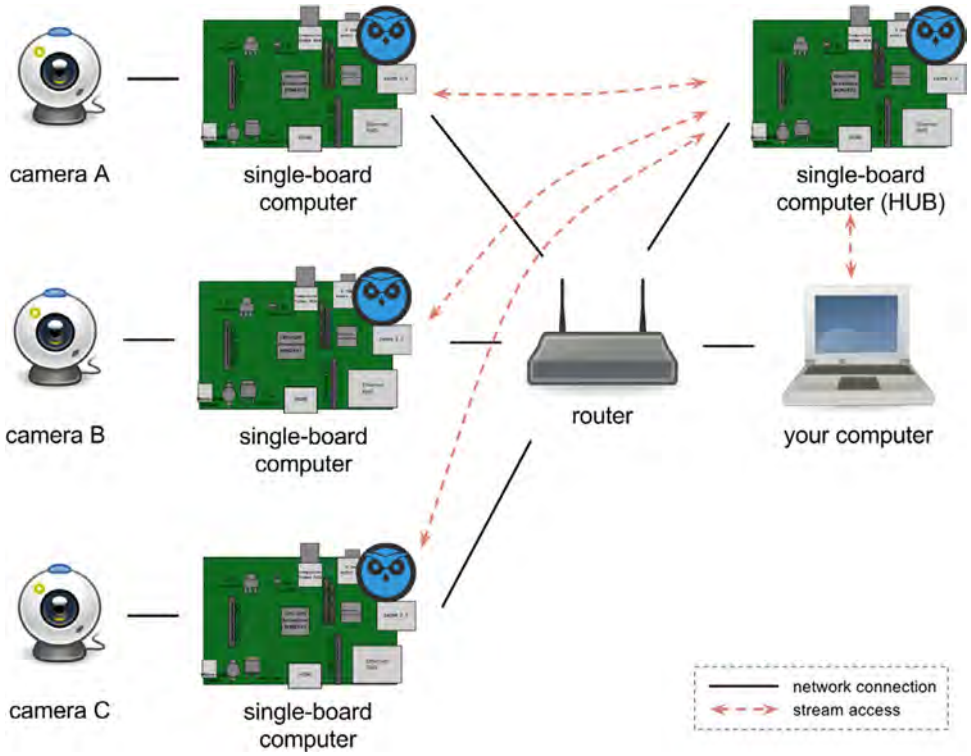

Fig. 7. Multiple devices with a central server scenario.

MotionEye can be used in four scenarios. A single device with a single camera (Fig. 4) is the simplest scenario that will help users get started and will be, in many cases, sufficient. The advantage of this first scenario is the simple setup. However, the users need to separately visit the address of each device, meaning to open one browser tab per device. Also, accessing the cameras from the Internet requires a separate port forwarding for each of the devices.

The second scenario involves a single device with multiple cameras (see Fig. 5) and is suitable for places where two or more cameras can be installed in a common spot, preferably facing different angles. The advantage brought by this second scenario is that the cameras can be controlled by a single motionEye device. Moreover, only one IP address and only one port forwarding are required for all the connected cameras. However, the board controlling all cameras should be powerful enough to handle all streams and all cameras and can be disconnected at once by disconnecting the controlling device.

The third scenario, presented in Fig. 6, involves multiple devices with a hub. A local motionEye setup can remotely control another motionEye camera. The user can simply designate one of the motionEye devices as a hub and add all the other motionEye-based cameras to this hub as remote motionEye cameras.This third scenario has various advantages. One of them is that the user can manage all the cameras from one single place, for example one browser tab. Then, only one IP address and only one port forwarding are required for all cameras and the hub device does not need to be more powerful than the others, maybe just to have a better network connection, since it only relays the traffic between the user's browser and the rest of devices. The disadvantages for this scenario include the following: all cameras must be motionEye-based, all devices must be reachable by the hub (in the HTTP sense), media files, pictures, and movies, if used, must travel through the hub when downloaded, and finally, the media files are not centralized, they are created and stored individually by each device.

In the case of the last scenario, presented in Fig. 7, multiple devices with a central server, the main difference from the hub case is that the rest of devices are used as simple network cameras rather than as remote motionEye cameras.When using Raspberry PIs, the Fast Network Camera function should be enabled, for a considerably better performance. The server can be motionEyeOS-based, preferably running on a multicore board, or it can be a regular Linux system that runs motionEye. The advantages brought by this scenario include: the user can view and manage all cameras from one browser tab, other MJPEG network cameras which are not motionEye-based can be added to the central server, only one IP address and only one port forwarding are required for all cameras, and media files can be saved on a hard drive, directly attached to the server, thus increasing the available space and download throughput. However, all media files are stored in one place, making it easier for attackers to destroy the recordings, the server must be powerful enough to process the streams from all cameras, and all devices must be reachable by the server.

Turning a Raspberry PI into a simple but fast network camera can be done using the option called Fast Network Camera, in the Expert Settings section. But the Fast Network Camera option works only on Raspberry PI boards, equipped with the standard Camera Serial Interface (CSI) module. MotionEyeOS normally uses the motion daemon internally to capture, stream, and process the video frames. While motion does a great job in most cases, it will not take advantage of the Raspberry PI's GPU processing power. All processing, including JPEG encoding, is done by the central processing unit, the CPU.

When enabling the Fast Network Camera option, motionEyeOS switches from using motion to using a completely different backend, called streamEye. streamEye, together with raspimjpeg.py will capture JPEG frames from the Raspberry PI's GPU and will stream them as MJPEG over HTTP. With Fast Network Camera enabled, the motionEyeOS-based camera can reach a significantly higher frame rate, at a higher resolution, the user can tweak many CSI camera-specific parameters directly from the UI, and the browser will consume less CPU at the same frame rate/resolution, because it uses a pure MJPEG stream, rather than triggering every refresh from JavaScript. The aim of streamEye is to be a simple but efficient MJPEG streamer. It waits for successive JPEG frames at input and serves them as MJPEG to all connected HTTP clients. It is written entirely in C and has minimal overhead. Raspimjpeg.py is a Python script that uses picamera (a package that provides a Python interface to the Raspberry Pi camera module) to program the Raspberry PI GPU, to continuously capture frames in video mode and output ready-made JPEGs. It supports most of the camera options accepted by raspistill and raspivid (Raspberry Pi camera applications intended for capturing images and videos). The hardware requirements for starting a MotionEye based security system are the following: the Raspberry Pi board, a USB camera or a board-specific camera module, a power supply capable of delivering enough current for the board (as well as for all attached peripherals), a storage device compatible with the board (usually an SD card or USB drive), and a Wi-Fi adapter or Ethernet connection.

4. Results

In the following, an example of security system in a real home will be presented. The system is composed by the access control and security part, accomplished by qToggle and controlled using qToggle app, and the MotionEye based video surveillance system.

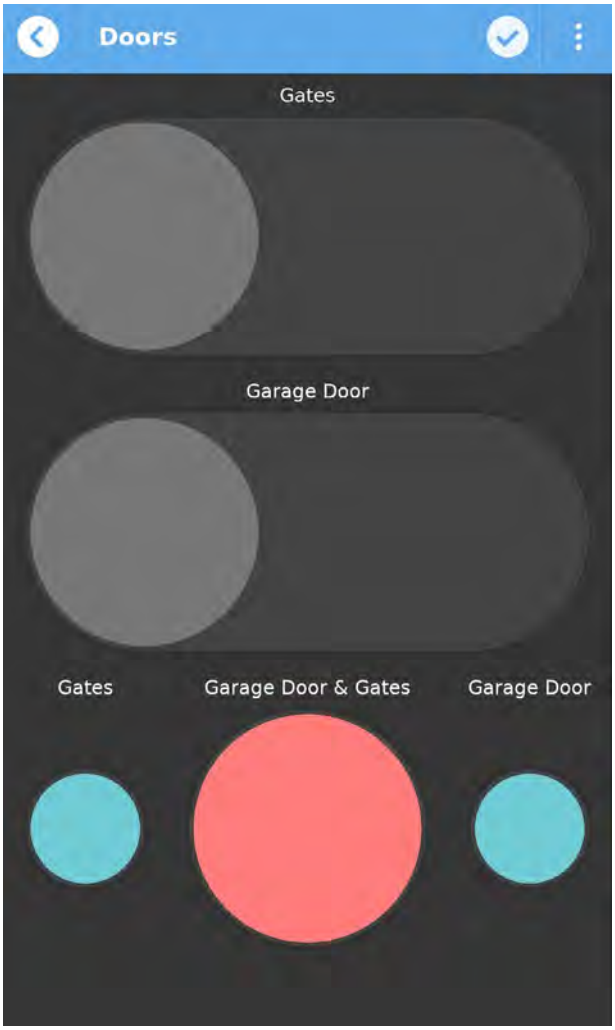

Fig. 8. Access control with qToggle.

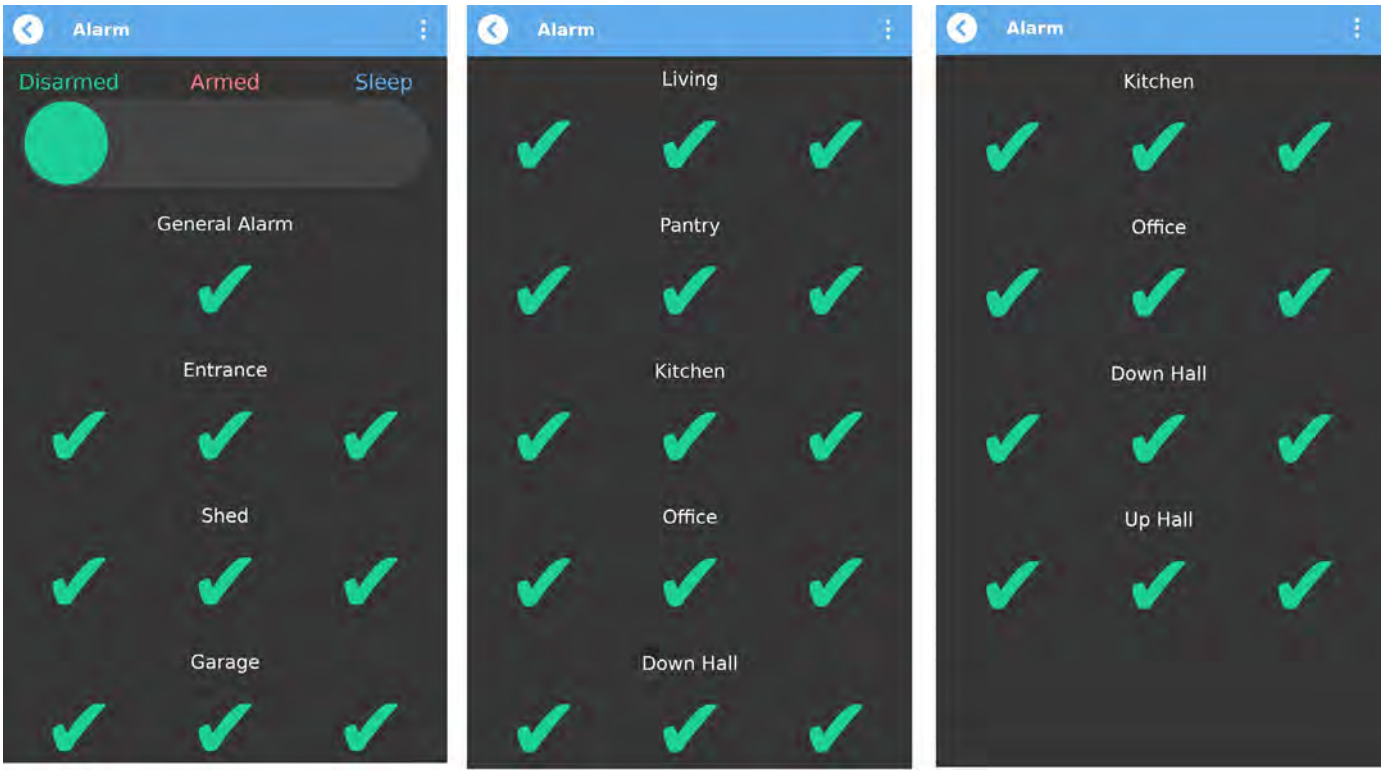

Fig. 9. Controlling the security system with qToggle.

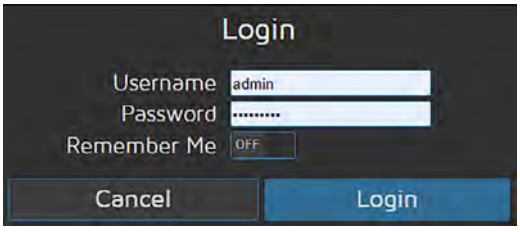

Fig. 10. The login process on MotionEye.

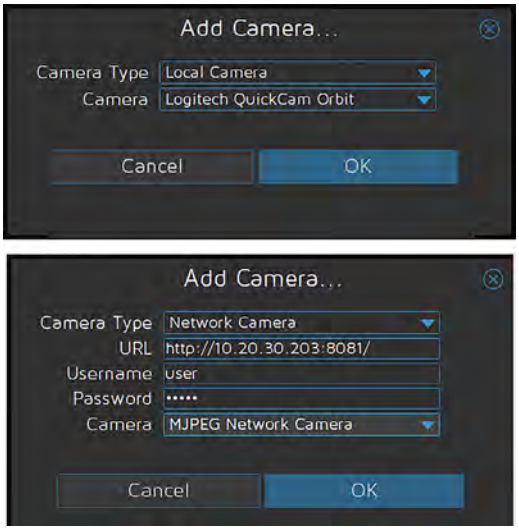

Fig. 11. Adding local cameras or network cameras.

4.1. Access control and security using QToggle

Access control implies controlling various doors and gates in a house or building. These tasks can be monitored and controlled very easily, using qToggle app, and its associated ecosystem. The example given in Fig. 8 presents access control on qToggle and involves the control of gates and garage door in a residential home. The system provides several options: to fully open or close the gates, to fully open or close the garage doors, open/close both the gates and the garage at the same time, or to keep half open one of them or both. Access control can be done manually, by using the qToggle app and by voice, using a smart assistant (Google or Amazon Alexa) on a smart watch, for example and certain commands, set by the user. To control gate motors and the garage door, we used two Blitzwolf SS1smart relay boards that enable remote opening/closing. This allowed us to mimic the conventional gate remote control using our Wi-Fi-based system.

A usual security system consists of a master control panel, the keypad, motion sensors (Paradox PIR sesors), and the siren. In qToggle app, we have the options to arm and disarm the security system, without using the keypad, as shown in Fig. 9. Arming and disarming are performed manually using the qToggle app, and by voice commands, using a smart assistant. We have also implemented the Sleep option, which is used during the night and arms only the ground floor of a two floors house, for security reasons. If motion is detected downstairs during the night, the alarm will trigger. Arming and Disarming, using Sleep mode, can be done manually, using qToggle app, but also using an extra light switch (Sonoff touch), placed upstairs, for added comfort. A custom integration module has been developed using a Raspberry Pi board to be able to control the alarm unit remotely.

The response time of qToggle for different features are shown in Table 1.

For computing the time, we used a Linux command line to send HTTP API requests to the devices and monitored the response time. The qToggle API ensures that a successful response is only received when the command has been completed.

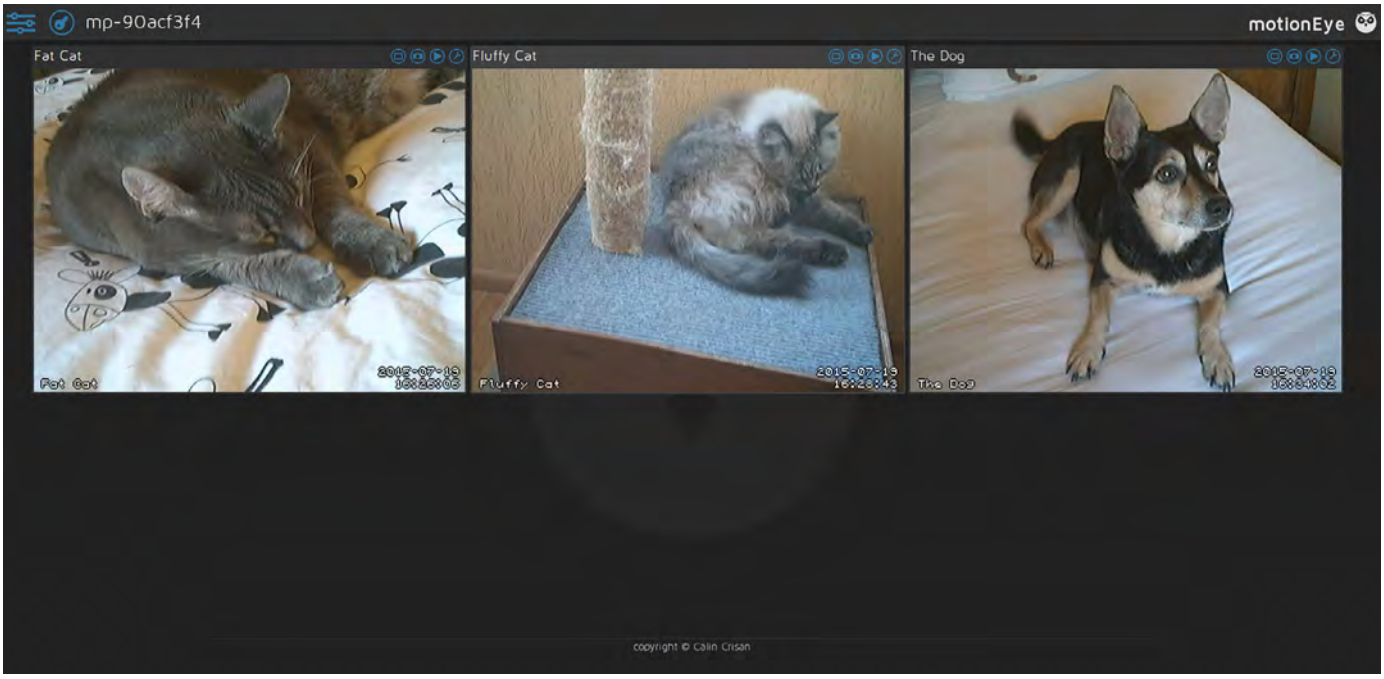

Fig. 12. Normal desktop view.

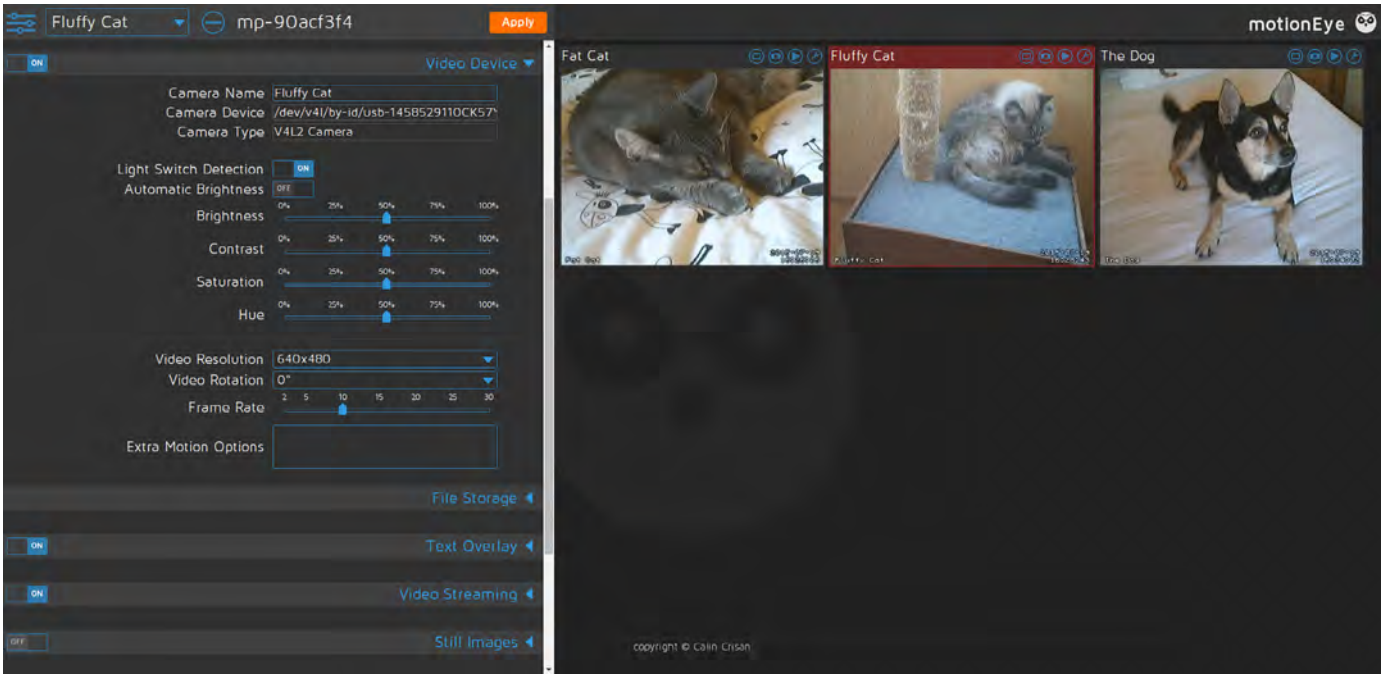

Fig. 13. Settings panel.

Table 1
The response time of qToggle.

Action	Time
Open/close garage door	19.81 s / 25 s
Open/close gates	18.63 s / 21.96 s
Turn on/off the alarm (arming/disarming time)	21 s/0.8 s
Notify turn on/off the alarm through sms	<30 s

4.2. MotionEye screenshots

The login process is very simple and implies a username and a password, as shown in Fig. 10.

Adding local cameras or network cameras can be done as shown in Fig. 11. Local cameras are camera devices, usually USB cameras or board-specific cameras, that are connected to the motionEye system. Network cameras (IP cameras) are devices that natively stream MPEG video or pain JPEG images.

The normal desktop view will be as shown in Fig. 12.

All the configurations regarding a certain cam can be performed in the Settings Panel shown in Fig. 13.

All pictures and movies recorded are stored and can be seen in the media browser shown in Fig. 14.

MotionEyeOS was founded in June 2014. Since then, the project has had 50 releases on Github and more than 650k downloads. The

Fig. 14. Media browser.

last release was in June 2020, with more than 22k downloads in one month.

5. Conclusions

Technology plays an important role in peoples life. Nowadays, our homes are full of electronics and devices and modern homes need smart solutions. Due to the growing popularity of smart home systems, smart home security has become a hot topic in the past years. Home security is a useful application of the IoT, used to build a low-cost security system for homes and, why not, for the industrial and commercial sectors. In this project, we proposed two simple solutions for home security based on Raspberry Pi boards and/or ESP8266 chips. Both choices are cost effective, small, and easy to work with. The installation of the proposed systems can be done easily, and it does not require a big infrastructure which is an advantage as compared to other existing systems. Our main goal is to make qToggle a smart home prototype, with various functionalities, a system that could be continuously developed, improved and extended to other functionalities. Our main future work is to integrate the video surveillance system intro qToggle, since we consider qToggle a promising project, with a lot of capabilities that could be monetized in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805.
[2] I. Yaqoob, E. Ahmed, I. Abaker, T. Hashem, A. Ibrahim, A. Ahmed, A. Gani, M. Imran, Internet of things architecture: recent advances, taxonomy, requirements, and open challenges, IEEE Wirel. Commun. 24 (3) (2017) 10–16.
[3] S. Vashi, J. Ram, J. Modi, S. Verma, C. Prakash, Internet of things (IoT) a vision, architectural elements, and security issues, in: International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2017, pp. 492–496.
[4] Internet of Things. Retrieved from: https://www.intel.co.uk/content/www/uk/en/internet-of-things/infographics/guide-to-iot-new.html.
[5] P.K. Madupu, B. Karthikeyan, Automatic service request system for security in smart home using IoT, in: Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology, 2018, pp. 1413–1418.
[6] A.R.F. Shafana, A. Aridharshan, Android based automation and security system for smart homes, Int. J. Comput. Sci. Inf. Technol. 5 (3) (2017) 26–30.
[7] L. Syafaah, A.E. Minarno, F.D.S. Sumadi, D.A.P. Rahayu, ESP 8266 for control and monitoring in smart home application, in: Proceedings of the International Conference on Computer Science, Information Technology, and Electrical Engineering, Jember, Indonesia, 2019, pp. 123–128.
[8] V. Patchava, H.B. Kandala, P.R. Babu, A smart home automation technique with raspberry Pi using IoT, in: Proceedings of the International Conference on Smart Sensors and Systems (IC-SSS), Bangalore, India, 2015, pp. 1–4.
[9] S.R. Khan, F.S. Dristy, Android based security and home automation system, Int. J. Ambient Syst. Appl. 3 (1) (2015) 15–24.
[10] W.A. Jabbar, T.K. Kian, R.M. Ramli, S.N. Zubir, N.S.M. Zamrizaman, M. Balfaqih, V. Shepelev, S. Alharbi, Design and fabrication of smart home with internet of things enabled automation system, IEEE Access 7 (2019) 144059–144074.
[11] Y. Zhao, Z. Ye, A low cost GSM/GPRS based wireless home security system, IEEE Trans. Consum. Electron 54 (2008) 567–572.
[12] C. Felix, I.J. Raglend, Home automation using GSM, in: The International Conference on Signal Processing, Communication, Computing and Networking Technologies, 2011, pp. 15–19.
[13] A.C. Jose, R. Malekian, Improving smart home security: integrating logical sensing into smart home, IEEE Sens. 17 (13) (2017).
[14] S. Zhang, P. Xiao, J. Zhu, C. Wang, X. Li, Design of smart home control system based on Cortex-A8 and ZigBee, in: Proc. IEEE 5th Int. Conf. Softw. Eng. Service Sci., 2014, pp. 675–678.
[15] C. Nayyar, B. Valarmathi, K. Santhi, Home security and energy efficient home automation system using arduino, in: Proceeding of the International Conference on Communication and Signal Processing, 2017, pp. 1217–1221.
[16] S. Wadhwani, U. Singh, P. Singh, S. Dwivedi, Smart home automation and security system using arduino and IOT, Int. Res. J. Eng. Technol. (IRJET) 5 (2) (2018) 1357–1359.
[17] A. Pawar, V.M. Umale, Internet of things based home security using raspberry Pi, in: 2018 4th International Conference on Computing Communication Control and Automation, 2018, pp. 1–6.
[18] A. Goel, Raspberry Pi project - home automation, Retrieved from: https://engineering.eckovation.com/raspberry-pi-project-home-automation/.
[19] E. Upton, G. Halfacree, Raspberry Pi User Guide, A John Wiley and Sons Ltd., 2012.
[20] M. Shahajan, G.M. Jahedul, S.K. Das, S. Islam, M. Islam, K.K. Islam, Internet of things (IoT) based automatic electrical energy meter billing system, J. Electron. Commun. Eng. 14 (4) (2019) 39–50.