Laboratory Report

Contactless Elevator

Group Members

6230123921	Thitaree	Setwipattanachai
6231301421	Kanokpich	Chaiyawan
6231304321	Kittipong	Deevee
6231353021	Raviporn	Akekunanon
6231372021	Atiwat	Deepo

Present to

Thanis Tangkitjaroenkun

This report is a part of subject

2110366 Embedded System Laboratory (2/2020) and

5500308 Technical Writing for Engineering (2/2021)

Chulalongkorn University

Abstract

Since the COVID-19 pandemic declaration in March 2020, technologies have become an important part of mitigating the situation. Elevator, narrow in dimensions, is one of the technologies that have been affected the most. The objective of this project is to construct a touchless embedded system for an elevator with QR codes and a smartphone to reduce the exposure to the surface resulting in preventing the spread of COVID-19. The system was divided into three sections, those are, elevator control, elevator simulation, and elevator capacity checking which were implemented by STM32F411 Nucleo Boards, NodeMCU ESP32, NETPIE, IR sensors, and speakers. As a result, a touchless elevator system with an ability to count the number of passengers was developed to serve the objective. Additionally, the UI was also implemented to accompany the new users to interact with the system easily which could result in the system being more popular. The solution is also intended to replace the existing toothpicks, as button pushers, in an elevator with several improvements that could be implemented in further research.

Introduction

Technology, nowadays, plays a crucial part in every living day. It could be as simple as a light switch to turn a light on and off, or as complex as a smartphone. Specifically, one of the important systems in computer technology is called the embedded system. It is often implemented with microcontrollers which are parts of some larger systems such as a washing machine, digital alarm clock, or smart watch. Despite its small size, the system acts like a brain controlling many major technologies like automobiles, medical equipment, or industrial machines. The concept mentioned earlier is called System on a Chip (SoC), where the system is implemented on a chip controlling all processes [1]. Trends in making SoC have been steadily growing [2] because of its simplicity and diversity in technologies able to be implemented.

Amid the COVID-19 situation, the old lifestyle has been changed. Social distancing and awareness in hygiene have become important as never before [3]. An elevator, for instance, is one of the places that has been changed the most. In it, people need to stay in their own corner separating from one another, hand sanitizer is equipped, and toothpicks are used to replace fingers to select floors to depart.

In the last few years, technologies in many fields have been developed to mitigate the spread of COVID-19 [4]. Most importantly, for the sake of the continuity of everyday life, telehealth, remote working, and distance learning have been developed. However, some people are obliged to go outside and interact with people, but few technologies have been made to accompany them. As mentioned before, toothpicks are used in the elevator as a finger replacement. Explicitly, the solution is still unhygienic as used toothpicks are often left in the elevator, and also not environmentally friendly as they produce a lot of garbage.

The objective of this project is to propose a solution to prevent the spread of COVID-19 by implementing a touchless system for an elevator with QR codes and a smartphone, to replace a legacy system covering two main functions. First, for an elevator controller, a smartphone is used to control the elevator as a replacement for elevator buttons. Second, for a capacity checker, IR sensors are used to track the amount of people in an elevator and a speaker is used to warn people when maximum capacity is reached. The system is implemented as an embedded system with STM32 as a microcontroller, NETPIE as a middleware to communicate with RESTful API between an elevator and a smartphone, and other equipment as mentioned.

Experimental methods

To start the project, the system workflow was divided into three sections including elevator control, elevator simulation, and elevator capacity checking. For equipment, two STM32F411 Nucleo Boards were used as the main processor and I/O controller, and one NodeMCU ESP32 (DOIT DEVKIT) was utilized as an intermediary in communication between STM32F411 and NETPIE cloud server. To count the number of people in the elevator, two CNY70 Reflective Photoelectric Switches (IR sensors) were applied along with a speaker module used to play a sound when the number of people exceeded the limit. The last one is NETPIE, an

IoT cloud-based platform-as-a-service which helps connect IoT devices. It was implemented to collect data from ESP32 and allowed external servers to retrieve data from the NETPIE cloud server via MQTT protocol or RESTful API.

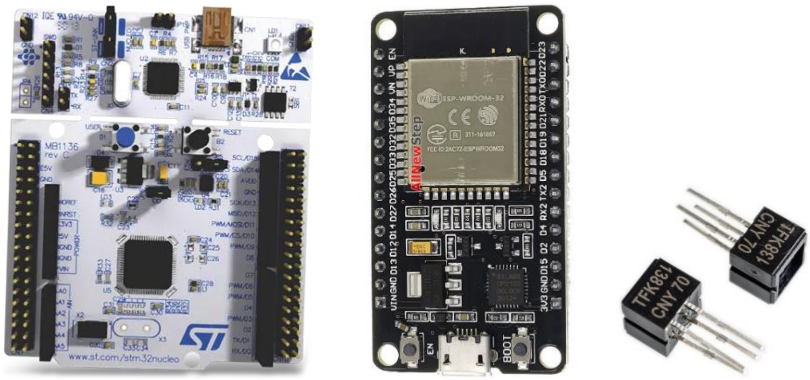

Figure 1: Equipment needed for this system

(Left: STM32F411 Nucleo Board, Middle: NodeMCU ESP32 (DOIT DEVKIT),

Right: CNY70 Reflective Photoelectric Switches)

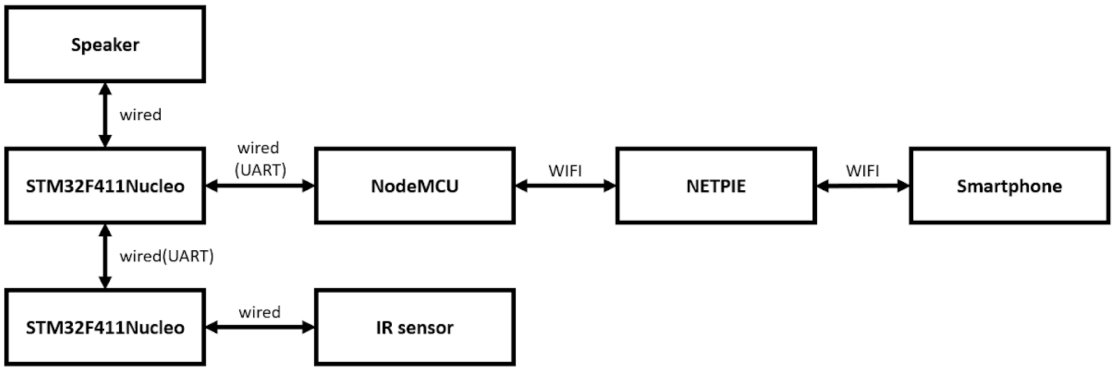

Figure 2: Diagram for the connection of all devices in this system

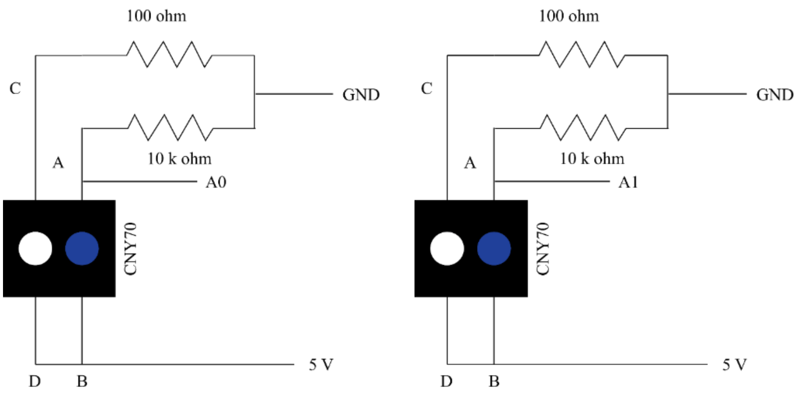

Figure 3: Diagram of IR sensors connection (Left: IR1, Right: IR2)

To explain more about the elevator control part, users could use their phones to scan QR codes that would be set inside the elevator to enter the website. Once they were on the website, they could select which floors they desired to go to by pressing buttons on the website. After the buttons had been pressed, a new message containing the floor number would be posted from the website to the NETPIE MQTT broker via its RESTful API. When the NETPIE MQTT broker got a new message, it would pass the data in the message to ESP32 which monitored changes in the data on the cloud server (NETPIE). If the data was updated, ESP32 would call its callback function that would send a message containing the floor number to STM32 saying that a user had selected a new floor. After receiving a message from ESP32, STM32 would simulate the elevator by moving to the destination floor and send back data of the current simulation state to ESP32 each time it reached a floor. Finally, ESP32 received the response data from STM32 and updated it to the cloud server so that the website could fetch the latest information and display it on the page for users to see.

Figure 4: User interface (UI) on the elevator controller website

Moving to the elevator simulation part, when the elevator reached the destination floor, one of the STM32 would control the speaker to play a sound, signaling that you have arrived. The STM32 would also send the updated state of the buttons back to ESP32. Moreover, the simulation would display all the elevator status including door status, current floor, direction, and button status on the web page by pulling data from the NETPIE cloud server.

Lastly, in the elevator capacity checking part, two IR sensors were used to count people getting in and out of the elevator. To operate the sensors, three program threads were provided in STM32, two threads for updating the status of each sensor separately, and one thread for combining the status of both sensors to identify whether people came in or out of the elevator. If the current number of people inside the elevator exceeded the maximum capacity (4 persons), STM32 would signal the speaker to play warning sounds until the number of people decreased to the maximum capacity.

Results and discussions

The attempt to construct a touchless embedded system for an elevator with QR codes and a smartphone was successful. Compared to using toothpicks instead of the bare hands as an alternative method, it was cleaner and less wasteful. Despite the successful outcome, many obstacles were presented in our attempt to implement the contactless system.

Overall, the final implementation of the system was simple and not at all complicated. Moreover, the user interface of the website which was designed for users to interact with the system, as shown in Figure 4, was easy to use and had not too many elements in one page, which contributed to making a good user experience with ease of use [5].

The procedure to count humans passing through the elevator door was an implementation of two IR sensors horizontally placed. So, when humans passed the elevator door, one of the sensors could detect the change of the value before the other one. Then, we could observe the difference between two sensors and compare them to determine whether humans were going in or out of the elevator. However, the established final system has some constraints. There are some limitations in the system; one is that the light threshold is dependent on the position of IR sensors in the elevator, so the threshold value must be configured after placing the whole system in the elevator and should be calibrated periodically. Additionally, the IR sensors are not able to differentiate between humans and objects. This problem occurs from the fact that the object detection algorithm used in this project requires values read from the sensors at two different points of time which cannot be deterministically determined, for how long the sensors should measure the object, as we have no information about the average time humans took to pass an elevator door area.

Despite the seemingly complete system, an issue occurred after the first implementation of the system. Due to the inexperience in dealing with the STM32F411 boards, there were struggles to have several threads working on the same board at the same time since the board has only one logical thread. This problem was solved by having another STM32F411 board connected to the primary board using UART to send signals between boards.

Another issue which was a concern in the implementation was about overusing the REST API calls and exceeding the free quota given. Therefore, to avoid that anticipated situation, an attempt was made to create a backend for the website to send MQTT messages to NETPIE instead of using the REST API calls since there was more quota offered than using the former method. However, this effort turned out to be in vain since it was unfeasible to integrate an MQTT client in sending messages as it kept disconnecting far too often, even after switching between different MQTT JavaScript libraries (Paho [6] and MQTT.js [7]). In the end, we decided to stick with using REST API calls after recalculating the free quota given.

Alternative to this solution is a touchless elevator using hand sensors over the elevator buttons [8], without the QR codes and website. Since it does not have to connect with the network, this hand-sensor alternative solution provides a better user experience from no delay in the system website and users seem to understand how to use hand sensors effortlessly. However, this project system can be further developed. Since the number of persons in the elevator on the cloud service can be retrieved from any system, it can be displayed outside the elevator or through the website so that the ones who waited for an elevator can decide whether he/she still wants to take the elevator. It can also make an analysis of the intensity of the capacity during the operation hour.

Other interesting alternatives to the system are toe-to-go [9] elevators, using feet to push the buttons near the elevator's floor, or bluetooth elevator [10] with an application. Even the touchless elevator seems to be the alternative solution for life during COVID-19, due to system migration constraints, Self-Cleaning Elevators Push Buttons by coating the button with thin-film copper [11] is still a great solution [12].

Conclusion

The purpose of this paper is to implement a touchless elevator system to prevent the spread of COVID-19. We implemented the system by connecting 4 parts together; a smartphone as a device to control the elevator and observe the number of people in it, STM32F411 Nucleo Boards as the main functionality of the system, NETPIE cloud server to store and manage data, and a NodeMCU ESP32 to connect all other three devices mentioned. For further implementation, one may develop a system for users to check the number of people in an elevator from anywhere.

Nevertheless, research on the elevator usage analysis amid the situation is still needed to further understand the needs of an elevator system so that an appropriate system could be developed.

References

[1] Dimitrios Serpanos & Tilman Wolf, 2011, ***Architecture of Network Systems (1st ed.)***, Morgan Kaufmann.

[2] Globe News Wire, 2021, ***Global System-On-Chip Market grow at CAGR of 8.2%: Qualiket Research***, https://www.globenewswire.com/news-release/2021/11/01/2324592/0/en/Global-System-On-Chip-Market-grow-at-CAGR-of-8-2-Qualiket-Research.html.

[3] WHO Africa, 2020, ***COVID-19 pandemic heightens the importance of handwashing with soap***, https://www.afro.who.int/news/covid-19-pandemic-heightens-importance-handwashing-soap.

[4] Mihalis Kritikos, 2020, ***Ten technologies to fight coronavirus***, https://www.europarl.europa.eu/RegData/etudes/IDAN/2020/641543/EPRS_IDA(2020)641543_EN.pdf.

[5] Interaction design foundation, ***Ease of Use is a central usability concept***, https://www.interaction-design.org/literature/topics/ease-of-use.

[6] Python Package Index, 2014, ***Paho MQTT project***, https://pypi.org/project/paho-mqtt.

[7] Adam Rudd, 2016, ***MQTT.js***, https://github.com/mqttjs/MQTT.js.

[8] MAD Elevator Inc., 2020, ***PHANTOM Touchless Sensor***, https://www.madelevator.com/files/support/phantom-touchless-sensor.pdf.

[9] MAD Elevator Inc., 2020, ***Toe-To-Go Foot-Activated Elevator Call Button***, https://www.madelevator.com/files/support/toe-to-go-brochure.pdf.

[10] MAD Elevator Inc., ***Germ-Free Elevator Solutions***, https://www.madelevator.com/germ-free-elevator-solutions/page.html.

[11] MAD Elevator Inc., 2020, ***BS Copper Pushbutton***, https://www.madelevator.com/files/support/bs-copper-pushbutton-brochure.pdf

[12] Technology Networks, 2021, ***Coating Surfaces With Copper has the Potential To Kill SARS-CoV-2 Faster***, https://www.technologynetworks.com/immunology/news/coating-surfaces-with-copper-has-the-potential-to-kill-sars-cov-2-faster-357106.