Design and Built IoT Home Panic Button for Smart City

Sandy Suryo Prayogo¹, Farid Al Rafi² and Yulisdin Mukhlis³

¹Electrical Engineering of Technology Industry, Gunadarma University Jalan Margonda Raya 100 Depok, Indonesia
²Electrical Computer System of Technology Information,Jalan Margonda Raya 100 Depok, Indonesia
³Electrical Engineering of Technology Industry, Gunadarma University Jalan Margonda Raya 100 Depok, Indonesia

*sunariver@student.gunadarma.ac.id

Abstract. Smart City is a concept where a city or regional region adopts Internet of Things (IoT) and Internet communication technologies (ICTs) to help manage infrastructure. A home security system to the public security is essential in Smart City to help provide a warning in case of unfortunate disaster in the home, so security officers can quickly take action. This paper aims to create a design one form of Smart City for the security system is IoT home Panic Button which gives warning to the security in case of emergency that requires help. Panic Button is made using a NodeMCU ESP8266 microcontroller that connects to a home Wi-Fi network. Message Queue Telemetry Transport (MQTT) is a communication system used, with the condition if the panic button is being pressed it will send a message to the MQTT broker and continue the message to subscriber. The subscriber is a web that notifies warning alarm and a map of the area covered that will be placed on the public security. In this way, a secure, flexible and reliable system is developed to solve the above mentioned.

1. Introduction

Smart city is one of the biggest and latest concepts that are popular. Smart City is a term to describe the vision of urban development to integrate information and communication technology (ICT) and Internet of Things (IoT) to manage city assets. These assets include local information systems, schools, hospitals, transport systems, power stations, law enforcement, security and other public services.

One of the important things is about public safety, from a simple thing that is an emergency occurrence that often occurs inside the house, but the surrounding security is not aware of the incident. Due to the absence of rapid handling so that the victim emerged. As an example of a robbery in a home, pregnant women who will give birth in need of help, or a minor accident that can cause a fire.

Based on the above cases the idea came up to create a simple home security system that could help the house in need of help to inform the security authorities. The security system is a panic button. Panic button is a tool in the form of a button that when in the press will activate the alarm in the security room and show which house is in an emergency need help.

The purpose of this project is to build and create a panic button using NodeMcu ESP8266 12F board microcontroller with MQTT (Message Queue Telemetry Transport) communication over TCP/IP. NodeMcu will connected to a home-owned Wi-Fi network acting as a publisher that sends
messages to the broker (smart city server) and forwarded to the subscriber. Subscriber here is a device and web monitors are on the security side. The web monitor displays the map of the area covered, and displays every home position, if the alarm is pressed from a house, the alarm will sound and the map will show which house is in need of help.

2. Literature Review

A. Internet of Things

The Internet of things (IoT) is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

![Figure 1. Internet of Things](image1.png)

The Internet of Things vision is grounded in the belief that the steady advances in microelectronics, communications and information technology we have witnessed in recent years will continue into the foreseeable future. In fact – due to their diminishing size, constantly falling price and declining energy consumption – processors, communications modules and other electronic components are being increasingly integrated into everyday objects today.

With myriad technological and other elements such big data analytics, actuators, data hubs, artificial intelligence, connectivity, all sorts of sensors, IoT nodes, IoT gateways, networks (whereby the use case drives the choice of technology), the cloud, edge computing, information processes, business process optimization, people, smart goals, increasingly robotics, IoT platforms/middleware, new ecosystems of value, IoT data brokerage platforms and much more the Internet of Things enables an unseen new wave of innovation and optimization.

![Figure 2. Application of Internet of Things](image2.png)

When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.
B. **MQTT (Message Queue Telemetry Transport)**

MQTT is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message. Andy Stanford-Clark and Arlen Nipper of Cirrus Link authored the first version of the protocol in 1999.

MQTT defines methods (sometimes referred to as verbs) to indicate the desired action to be performed on the identified resource. What this resource represents, whether pre-existing data or data that is generated dynamically, depends on the implementation of the server. Often, the resource corresponds to a file or the output of an executable residing on the server.

a. **Connect**
   - Waits for a connection to be established with the server.

b. **Disconnect**
   - Waits for the MQTT client to finish any work it must do, and for the TCP/IP session to disconnect.

c. **Subscribe**
   - Waits for completion of the Subscribe or unsubscribe method.

d. **Unsubscribe**
   - Requests the server unsubscribe the client from one or more topics.

e. **Publish**
   - Returns immediately to the application thread after passing the request to the MQTT client.

![Figure 3. MQTT OSI Layer](image)

The MQTT connection itself is always between Information and communication technology (ICT) is used to enhance quality, performance and interactivity of urban services, to reduce costs and resource consumption and to increase contact between citizens and government. The MQTT protocol is based on top of TCP/IP and both client and broker need to have a TCP/IP stack.

Smart city applications are developed to manage urban flows and allow for real-time responses one client and the broker, no client is connected to another client directly. The connection is initiated through a client sending a CONNECT message to the broker. The broker response with a CONNACK and a status code. Once the connection is established, the broker will keep it open as long as the client doesn’t send a disconnect command or it loses the connection.

C. **Smart City**

A smart city is an urban area that uses different types of electronic data collection/sensors to supply information used to manage assets and resources efficiently. This includes data collected from citizens, devices, and assets that is processed and analyzed to monitor and manage traffic and transportation systems, power plants, water supply networks, waste management, law enforcement, information systems, schools, libraries, hospitals, and other community services.
Figure 4. Smart City

The smart city concept integrates information and communication technology (ICT), and various physical devices are connected to the network (the Internet of things or IoT). In order to optimize the efficiency on Smart city technology, the city officials are allowed to interact directly with both community, infrastructure and monitor what happens in the city\textsuperscript{6,7}.

D. NODEMCU ESP8266 12-F

NodeMCU a microcontroller with an integrated Wi-Fi, which means that there is no need for an additional Wi-Fi chipset. The design of the system on chip (SoC) allows communication through the GPIOs by connecting to the Internet and transmitting data over the Internet\textsuperscript{4}.

Figure 5. NodeMCU Pinout

NodeMCU uses the standard Lua Programming firmware script, in order to simplify programming, it is necessary to add the existing library in the Arduino IDE software.

3. Internet of Things Panic Button Design for Housing District in Smart City

One of the goal of smart city is to make the people feeling safer by support of the technology. The IoT Panic button devices can help the law enforcement to get alerted from citizens in real-time. The use MQTT for the communication of this devices because of the lightweight bandwidth usage and real-time delivery messaging.

Figure 6. Flowchart of Panic Button Process
Figure 7 show the process of panic button. When Panic Button pushed, it will send message as a publisher to broker, then server will subscribe the broker to get the message to turning on alert on the web interface of security.

![Panic Button Communication System](image)

**Figure 7. Panic Button Communication System**

The microcontroller used is NodeMcu ESP8266 12F board in Panic button to send the message through MQTT protocol. NodeMcu is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Expressive Systems and Hardware which is based on the ESP-12 module. NodeMcu was able to support the MQTT IoT protocol, using Lua to access the MQTT broker.

![NodeMcu Panic Button Schematic](image)

**Figure 8. NodeMcu Panic Button Schematic**

In the schematic, we use pull-down resistor for the push button. Pull-down resistors are resistors which are used to ensure that a wire is pulled to a low logical level in the absence of an input signal. If there weren’t for the pull-up/pull-down resistor, the MCU’s input would be floating when the switch is open and brought down only when the switch is closed.

![Panic Button 3D Model](image)

**Figure 9. Panic Button 3D Model**
4. Experimental Result
IoT Panic Button must be connected to Wi-Fi router that have internet connection. The SSID and the password of the router need to be inserted by flashing the NodeMcu.

In the server, we use an application to receive MQTT message from all of panic button device. We write this app in python code. After received a new message, the app will insert the message to database.
5. Conclusion

In this project, we have tried to make an IoT panic button, using Esp8266 NodeMCU-12F which is useful for security system. Esp8266 NodeMCU-12F is small size microcontroller which makes proposed system appropriate for given application. The panic button which connected to the Wi-Fi network can send messages in the form of the address to the broker and then proceed to the subscriber properly. Subscriber received messages are then displayed on the web and enable alarm warning.

Acknowledgments

This research was fully supported by Gunadarma University, Jakarta, Indonesia. The authors gratefully acknowledge Gunadarma University for providing research funding and for permission in using the research facilities.

References

[1] Andy Stanford-Clark and Hong Linh Truong. “MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2”. (andysc@uk.ibm.com, hlt@zurich.ibm.com) (2013).
[2] ISO/IEC 20922. "MQTT 3.1.1 specification, Information technology Message Queuing Telemetry Transport (MQTT) v3.1.1", OASIS (2016).
[3] TU-T (2012). "Internet of Things Global Standards Initiative". ITU. Retrieved 23 October 2017.
[4] Lamir Shkurti. “Development of Ambient Environmental Monitoring System Through Wireless Sensor Network (WSN) Using NodeMCU and “WSN Monitoring”. 6th Mediterranean Conference On Embedded Computing (2017).
[5] Mattern, Friedemann; Floerkemeier, Christian. "From the Internet of Computers to the Internet of Things" Informatik-Spektrum. 33 (2): 107ñ121(2010)
[6] Matt Hamblen. 2015. “Just what IS a smart city?”. computerworld.com [Online]. Available: https://www.computerworld.com/article/2986403/internet-of-things/just-what-is-a-smart-city.html. [Accessed: 20 October 2017].

[7] McLaren, Duncan; Agyeman, Julian. “Sharing Cities: A Case for Truly Smart and Sustainable Cities”. MIT Press. ISBN 9780262029728 (2015).

[8] Santucci, Gerald. "The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects" ETH Zurich. Retrieved 23 October 2017.

[9] Vermesan, Ovidiu; Friess, Peter. “Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems”. Aalborg, Denmark: River Publishers. ISBN 978-87-92982-96-4 (2013)