IoT-Enabled Door Lock System

Trio Adiono¹, Syifaul Fuada²
School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jln. Ganesha No.10, Gd. Ahmad Bakrie (LABTEK VIII) Lt. IV, ZIP 40116, Indonesia³
Program Studi Sistem Telekomunikasi, Universitas Pendidikan Indonesia²

Sinantya Feranti Anindy³, Irfan Gani Purwanda⁴
Maulana Yusuf Fathany⁵
University Center of Excellence on Microelectronics, Institut Teknologi Bandung, Jln. Tamansari No.126, IC Design Laboratory, Gd. PAU Lt. IV ITB Campus, 40132

Abstract—This paper covers the design of a prototype for IoT and GPS enabled door lock system. The aim of this research is to design a door lock system that does not need manual input from user for convenience purpose while also remaining secure. The system primarily consists STM32L100 microcontroller as its core, TIP102 transistor that controls 12 V DC solenoid, and XBee module to communicate with the smart home’s host and receive status regarding user’s GPS position. The system is tested by measuring the user's distance from the predetermined location using GPS coordinate captured by an Android application, which serves to test whether the system is able to operate as intended and measure the device’s power usage. The test result shows that the device is able to work based on GPS coordinate data received, using 42.3 mA and 587 mA current in idle and active modes, respectively.

Keywords—Internet of Things; smart home; smart lock

I. INTRODUCTION

The Internet of Things (IoT) is a concept revolving around global information network that consists ‘things’ such as smart devices, sensors and actuators, and even smaller networks with their own identities and ability to self-configure and make their own decisions within certain extent, whether individually or collectively [1-2]. The advent of IoT foreshadows the future where everything and everyone is connected to the Internet through all devices they possess, whether from computer, smart phone, and other consumer devices. Objects within IoT network can also communicate using various communication technologies such as WiFi, Bluetooth, near-field communication, and many more.

One of the most common applications of the IoT technology is for supporting a smart home system. Recent smart home systems make good model on how an IoT architecture behaves, as objects in a house are connected to a gateway in wireless manner to communicate whether with each other and the house’s inhabitants. As a smart home system is intended to enhance the inhabitants’ living quality in terms of comfort of convenience, the IoT is utilized to at least enable easier control and monitoring over the home devices.

A smart home system can be categorized into at least one type based on its functionality: health care, entertainment, energy, and/or security [3]. Among these functionalities, security becomes one of the most crucial factors behind installation of the system. Enhancing home security using smart home can be done in a number of ways, including but not limited to installation of smart, customized door lock. Several examples of smart door lock implementation have already existed, such as camera-based door security systems [4-6], passwords [6-7], smart card [8], and proximity or location detection [9]. Each of the aforementioned methods has its own strengths and weaknesses, such as interoperability value of devices.

In this research, a GPS-based smart door lock is designed. The ultimate aim of this research is to design a door lock system that does not need manual input from user for convenience purpose while also remaining secure. However, for this publication, the scope is limited for examining the feasibility of utilizing GPS for location-based lock to achieve the aforementioned goal. Furthermore, the scope of this paper will be mostly limited to the lock hardware side, while the server and the Android application code handling the GPS tracking will only be briefly explained. This device is part of MINDS smart home system we designed, which also include RGB lamp [10], curtain controller [11], humidity and temperature sensor [12], fan controller [13], and infrared remote [14]. The smart home is controlled using Android-based application detailed in [15]. This paper details the design of the smart lock design and implementation, as well as the testing result.

This paper is divided into following sections: Section I presents the research background. Section II discusses the methodology that cover: highlighted research limitation, specification of developed system, hardware, system workflow, design of protocol, android and server. Section III reports the research result while the research conclusion and future work are given in Section IV and Section V, respectively.

II. SYSTEM DESIGN

A. Research Limitation

As mentioned above, this work is more focuses on the hardware part including defined diagram block, electronic circuit and its assembly, whereas the software part including system flowchart, android apps, and server are presented briefly, and it will be elaborated in detail for other publications. The test limit on the functional test (locked and unlocked condition) and power consumption measurement using digital multimeter. Further performance metrics will be explored in future.
B. System Specification

As the designed system is part of MINDS smart home system, the overall architecture is similar to other devices designed for the aforementioned smart home. The smart door lock is one of the nodes supported by MINDS system, which is connected to a Raspberry Pi-based host that serves as a ‘bridge’ for the nodes to receive commands from user device in the form of Android-based smart phone. As for the smart phone, it is connected to the host through either Bluetooth or the Internet (via cloud server). The architecture of the MINDS system is depicted in Fig. 1.

C. Hardware Design

The door lock works like an on/off switch, which is controlled based on user’s proximity from the door. The lock system utilizes a battery-powered STM32L100 microcontroller as its core, which is used for controlling 12 VDC/630 mA solenoid and an Xbee module to receive signals from the central host. The solenoid is controlled using TIP 102 transistor by utilizing its cut-off and saturation modes to switch the solenoid’s state. Furthermore, the system is equipped with components such as DC power jack, microUSB port for development purpose, switch, and reset and mode buttons. The block diagram of the door lock’s hardware is depicted in Fig. 2, while the circuit diagram for the solenoid control is depicted in Fig. 3.

As the transistor is required to work in saturation and cut-off region, the value of components such as resistor must be determined in such way so they can support these operation regions. According to TIP 102’s datasheet, the base current (IB) for saturation region is 1.9 mA. As such, to achieve the current, the equation (1) is used for calculating the base resistor value.

\[ I_B = \frac{V_{out} - V_{BE(on)}}{R_{bias}} \]  

(1)

The value of the \( V_{out} \) is based on STM32L100’s output port voltage, which is 3.3 VDC, while the voltage between base and emitter ports of the transistor is based on the transistor’s Darlington configuration, hence 1.4 VDC (0.7 VDC+0.7 VDC). Based on the calculation, it can be inferred that 1kΩ resistor is required to make the base current into 1.9 mA. Aside from the base current, the maximum power the transistor can handle also needs to be considered, especially when the solenoid is active. According to the datasheet, the transistor is able to handle 80 watt of power dissipation. Assuming the \( V_{CE} \) during saturation is 2 VDC, then the solenoid power is:

\[ P(\text{transistor}) = V_{CE}(sat) \times I_C = 2V \times 630mA = 1.26\text{Watt} \]

D. System Workflow

This section describes two separate programs required to operate the lock: the firmware inside the microcontroller and the mobile phone (Android) software to transmit GPS coordinate of the user to the host. The flowchart for the firmware is depicted in Fig. 4, while the flowchart for the Android application is depicted in Fig. 5.
E. Protocol Design

The message protocol used to transmit GPS data to the host is based on the protocol elaborated in [16]. The specific structure of the message for door lock control is depicted in Fig. 6 (consisting of the packet header, address, packet init, data payload, and checksum).

F. Android Application and Server Design

The door lock works by having the Android application that serves as user interface of the entirety of MINDS system to periodically detect GPS coordinate of the user, then send it to the server (and host) to be compared with the house’s coordinate. If the distance is 10 meters or less, the application will send command to both the server to unlock the house, which will be relayed to the host then to the lock hardware. Likewise, if the user’s distance with the house is more than 10 meters, then the system will be locked. The system will send a warning if an unauthorized attempt to access the device within the house is made while the house is locked.

III. RESULTS AND DISCUSSION

A. System Implementations

Based on the block diagram depicted in Fig. 2 earlier, the printed-circuit board for the controller is designed with the result depicted in Fig. 7. This controller design is similar to the controller from previous project [13]. The reasoning behind the similar design is to add interoperability value to the system while also enhancing its performance.

To control the solenoid lock, the 0th pin of STM32L100’s GPIO-B port is used as the output port. The software implementation for the port configuration is elaborated in Table I, while the control process is elaborated in Table II.

As previously explained, this system works together with the host and MINDS Android application. Fig. 8 displays the comparison between the MINDS application during normal (unlocked) condition and locked condition. The application captures the user’s location every 5 minutes using GPS, then send the result to the server (and host), after which the host will send command to the lock device. If the house is supposed to be locked, then a warning will be sent if there is any unauthorized attempt to control the device. The code implementation for the GPS tracking is depicted in Table III, while the code implementation for the control and warning by host is depicted in Tables IV, V and VI.

### Table I. GPIO-B Port Configuration in STM32L100 Microcontroller

| Header (3 Bytes) | Address (2 Bytes) | Packet Init (1 Byte) | Data Payload (1 Byte) | Checksum (1 Byte) |
|------------------|-------------------|----------------------|-----------------------|-------------------|
| 50 4D 45         | 07                | C1                   | 80 01                 | 00                |
| 50 4D 45         | 07                | C1                   | 80 01                 | 00                |

### Table II. Software Implementation for the Door Lock On/off Control

```c
if(rawlock==0x00) GPIO_LOW(GPIOB,GPIO_Pin_0);
else if(rawlock==0x01) GPIO_HIGH(GPIOB,GPIO_Pin_0);
```

![Fig. 7. The Appearance of PCB DC12V Controller for MINDS Door Lock Device at Actual Size (40.21 mm x 39.84 mm).](image)
TABLE III. CODE IMPLEMENTATION FOR GPS TRACKING IN ANDROID APPLICATION

if (head.equals("LC")) {
    if (SensorService.lactur != 0.0 && SensorService.lngcur != 0.0) {
        Toast.makeText(getApplicationContext(), "Locked", Toast.LENGTH_SHORT).show();
    }
} else {
    try {
        if (SensorService.lactur != 0.0 && SensorService.lngcur != 0.0) {
            Toast.makeText(getApplicationContext(), "Failed to Update Location", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Failed to Update Location", Toast.LENGTH_SHORT).show();
        }
    } catch (Exception e) {
        Toast.makeText(getApplicationContext(), "Failed to Update Location", Toast.LENGTH_SHORT).show();
    }
}

TABLE IV. SOFTWARE IMPLEMENTATION FOR THE RASPBERRY PI HOST LOCATION COMPARISON

TABLE V. SOFTWARE IMPLEMENTATION FOR RASPBERRY PI HOST DEVICE LOCKING

eif (sent['homeid'] == homeid and sent['head'] == 'SC'):
    try:
        if (sent['name'] == 'lock'):
            cnx = mysql.connector.connect(user='root',
                                           password='28031995',
                                           host='localhost',
                                           database='home'+homeid)
            cursor = cnx.cursor()
            cursor.execute("UPDATE info SET lockstatus = "lock")
            cursor.execute("SELECT homekey FROM info")
            ch.basic_publish(exchange='amq.topic',
                             routing_key='homeid',
                             body=AESEncrypt(json.dumps(sent),
                                             cursor.fetchone()[0]))
            print(setlock)
        elif (sent['name'] == 'unlock'):
            cursor.execute("UPDATE info SET lockstatus = "unlocked")
            cursor.execute("SELECT homekey FROM info")
            ch.basic_publish(exchange='amq.topic',
                             routing_key='homeid',
                             body=AESEncrypt(json.dumps(sent),
                                             cursor.fetchone()[0]))
        else:
            run_command('unlock', sent['param'],
                        cursor.fetchone()[0])
            BODY=AESENCYPRTJSONDUMPS(SENT),
            CURSOR=FETCHONE(0)
    except:
        print(sys.exc_info()[0])
channel.basic_consume(callback, queue='queue_name',
                       no_ack=True)
channel.start_consumming()
Fig. 9. A Setup for Functional Test of Door Lock.

### TABLE VII. POWER MEASUREMENT DOOR LOCK DEVICE

| Commands on GUI | Device Condition | Results |
|-----------------|------------------|---------|
| Lock            | The door is locked | √       |
| Unlock          | The door is unlocked | √      |

### C. Power Measurement

To measure the power consumption, simply we used digital multi-meter to know the current flow during two conditions (idle and process). The result of the test is elaborated in Table VIII. The power consumption in idle condition is 507.6 mW that obtained from 12 V<sub>DC</sub> * 42.3 mA, while in process condition is 7044 mW.

### TABLE VIII. POWER MEASUREMENT OF DOOR LOCK DEVICE

| Input voltage | Current |
|---------------|---------|
|               |         |
| Idle condition| 42.3 mA |
| Process condition| 587.5 mA |

### IV. CONCLUSION

In this paper, a prototype of location-based smart door lock system is designed. The system utilizes the user’s GPS coordinate that is captured from a mobile application, which is then sent to a smart home system’s central host to enable or disable the door lock based on the user’s proximity to the door’s designated GPS coordinates. Based on the testing conducted, it can be concluded GPS coordinates can be used for controlling door lock. However, further study is required to improve the quality of the system, whether in terms of power efficiency, area tracking and indoor accuracy, and further increase the security.

### V. FUTURE DIRECTION

For the next phase of the work, Android’s geo-fencing implementation will be studied to examine its effectiveness on improving the lock’s accuracy.