

Design and Build an Android Based Gateway Controller using the MQTT Protocol

R Moch Maghrifi Dwi Candra¹, Mochammad Taufik², Nugroho Suharto³

^{1,2,3} Digital Telecommunication Network Study Program,
Department of Electrical Engineering, State Polytechnic of Malang, Indonesia

[1maghrificandra14@gmail.com](mailto:maghrificandra14@gmail.com), [2moch.taufik@polinema.ac.id](mailto:moch.taufik@polinema.ac.id), [3nugroho.suharto@polinema.ac.id](mailto:nugroho.suharto@polinema.ac.id)

Abstract—The gate is the most important component of a building's security system. Although almost all homes have gates, they are still opened and closed in the traditional manner. As technology advances, a remote control may be used to automatically open and close the gate. In this study, the author attempts to develop a gate utilizing an internet-connected microcontroller and an android smartphone application to make it simpler for users to manage and monitor the motor's status and eliminate the need for a remote control. The automatic gate prototype includes a track length of 3 meters, a 12 V dc gearbox motor, and a control system. The gate load standard is 15.40 kg. The study's findings employ the Internet of Things (IoT) to regulate and observe gate opening and closing as well as gather data on gate status and maintenance. According to test results, an average torque value of 78.44 Nm at an average current value of 10.25 A was obtained utilizing a mass load variation between 1 and 15 kg. Qos performance between Node MCU and Android application shows the quality of data transfer during deceleration with an average value of 34.3 seconds. The average throughput value is 149.6719 bps.

Keywords— Gate Controller, MQTT Protocol, Microcontroller, Torque, Android, Node MCU, DC Motor.

I. INTRODUCTION

The gate is the foremost part of security in guarding a building. Almost every house has a gate but still uses the conventional way of opening and closing. As technology develops, the gate can be opened and closed automatically using a remote control. Every technological system certainly requires a device or equipment consisting of various electronic components, while the drive to open and close the gate usually uses a large capacity electric motor, requiring expensive installation and repair costs. [1]

The Internet of Things (IoT) is a technology that enables internet-connected gadgets to connect with one another and gather and exchange data [2]. One of his IoT applications on home devices, also called Smart home/Home Automation [3]. The device can provide information from sensor readings to its owner via a smartphone, and the owner can issue commands via the same application [4]. The most important challenges in the IoT network, particularly in the smart home, are now privacy and security [2]. It encourages researchers to develop technologies such as gate operator control systems that benefit people living in homes with vehicles. Neighbors will not be disturbed by sirens. When it rains, the door can be opened from the inside, so people at home don't have to go out, and drivers and people sitting in the car don't have to get out of the car and open the door from inside with remote control [5].

However, from an efficiency point of view, using the remote control as a means of automatically opening and closing the gate is not possible because sometimes you forget to bring the remote control and something blocks it, and people are outside. Not beneficial. The house finds it difficult to open. This

problem can be solved using a remote control, server, system and communication protocol for Android smartphones.

Smartphones have evolved into multipurpose portable devices and taken over many people's daily lives in the last ten years [6], they are important as communication and information that can be easily accessed anytime and anywhere, smartphones are always with us and close to us. For these reasons, the functions of mobile phones, which are commonly used as communication tools, can now be used as door remote controls. Hopefully, this system will help owners control and monitor, as well as maintain at an affordable price [7].

The Internet of things (IoT) consists of a wide range of unique devices with various processing, sensing, acting, and communicating abilities. In order to make communication in IoT contexts easier, a number of new protocols are being developed in the protocol stack [8]. The widely used Internet of Things (IoT) protocol known as Message Queuing Telemetry Transport (MQTT) [9]. The Message Queue Telemetry Transport protocol (MQTT), one of the protocols in the IoT application layer that has been specifically created for restricted devices, is of interest in this study.

MQTT is a transmission protocol developed by IBM primarily used for the Internet of Things, MQTT is designed to be able to publish / subscribe to a lightweight message transmission. MQTT can be used for network optimization with low bandwidth and minimum energy computing equipment, MQTT is used for various application scenarios from the Internet of Things (IOT). Local charger management and establishing a connection to the cloud server via the Message Queue Telemetry Transport (MQTT) protocol are the responsibilities of the IoT gateway [10]. Several previous studies stated that MQTT can be used as a solution for remote

wireless systems. The application of MQTT to the IoT environment is also a low power solution used to improve failure detection. MQTT in sending messages experiences lower delay than COAP for lower packet loss, and higher delay than COAP for higher packet loss. [11]

The need for electric motors is a basic need to advance an industry. Electric motors, one of which is a DC motor, play an important role in production speed and quality. A separate amplifier DC motor is a type of DC motor that can increase power and speed capabilities because it has a field flux generated by the field coil, which is located separately and has its own generator source in the form of a DC voltage. So that this type of separately amplifying DC motor is very possible to be able to generate a field flux when compared to a permanent magnet DC motor [12]. Power Window Motor Gear with a voltage of 12 Volt DC is one of the components that is easy to get on the market, besides the low electric power used, maintenance is easy to do [13].

A microcontroller board called Arduino UNO is based on the ATmega328P. It contains 6 analog inputs, a 16 MHz ceramic resonator, 14 digital input/output pins (six of which can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. It comes with everything needed to support the microcontroller; to get started, just plug in a USB cable, an AC-to-DC adapter, or a battery. Fig. 1 shows an image of the Atmega328P-based Arduino UNO board, which is the current iteration of the board.



Figure 1. Arduino Uno [14]

Even though the UNO board has been around for about ten years, it is still often utilized for various scientific and educational endeavors. The board's rigorous standards and top-notch performance make it an excellent tool for real-time sensor data collection and for triggering sophisticated laboratory equipment, to name a few applications.

The ease of programming makes the UNO more appealing than a basic Atmega328P chip [15]. Sketches can be swiftly uploaded from the computer to the Arduino UNO board in a matter of seconds thanks to the Atmega16U2 IC/USB microcontroller. Contrarily, when utilizing a bare Atmega328P microcontroller, sketches must be uploaded using either a separate Arduino acting as an in-circuit serial (ISP)

programmer or using an FTDI programmer [16]; however, if the wiring is incorrect, this method may result in time-out failures and possible chip damage.

As a result, the UNO is unquestionably the superior option for beginners to projects, but it hasn't been determined whether the UNO is suitable for battery-powered projects, especially when compared to using the Atmega328P as a standalone chip.

II. METHOD

A. System Plan

The behavior of the tools that run the system is represented systematically in the form of a block diagram shown in Fig. 2 below.

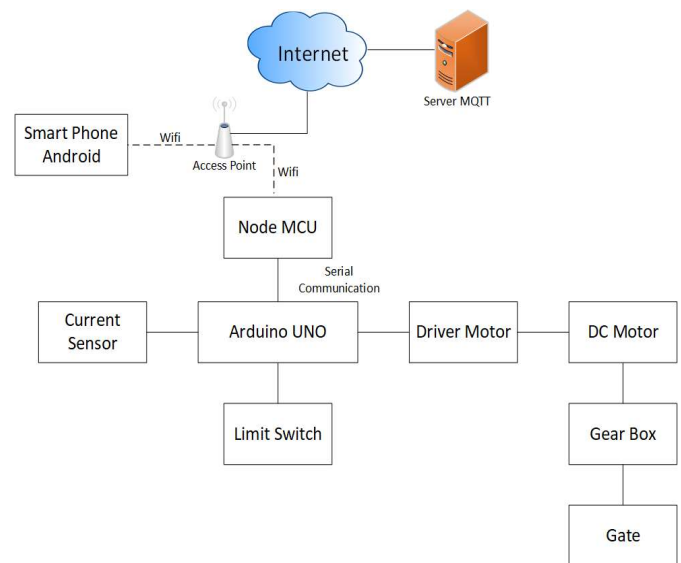


Figure 2. Block diagram of the system

Fig. 2 shows a block diagram of the system. The first step is to design an iron gate and prepare the necessary equipment and materials. All tools and materials are connected by cables, NodeMCU and smartphones use an internet-connected WiFi connection. Arduino UNO connection to NodeMCU uses serial communication i.e. Rx and Tx. Data obtained from sensors and other devices are collected with an Arduino that already contains programs. Arduino sends data to NodeMCU via serial communication. NodeMCU uses the MQTT communication protocol to send data already contained in the program. Data sent from NodeMCU via MQTT protocol communication is sent to the broker or his MQTT server. Once the software is configured and installed on your smartphone, the data display will appear on your smartphone's software, allowing you to monitor and control it.

B. Electric Plan

Electrical planning is the phase of functioning of electronic components. During this phase, each component is displayed.

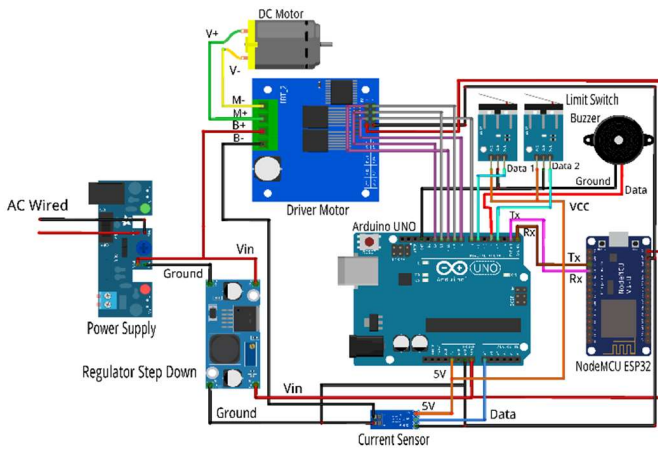


Figure 3. Entire network of remote control devices

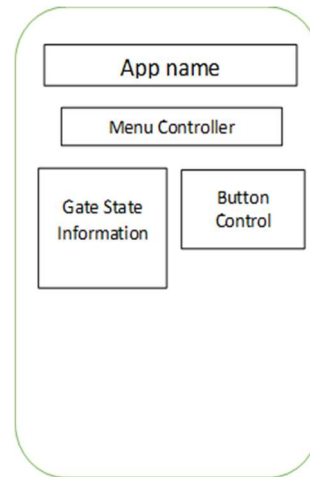


Figure 5. Android application

C. Machine Plan

This plan is the first part of the tool discussion to help facilitate tool fabrication in the next stage, and in the schematic plan it appears in the form of a fence as shown in Fig. 4.

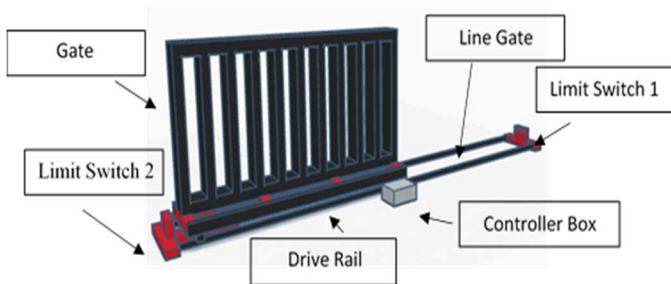


Figure 4. View from gate

In Fig. 4, a gate is used as an object to serve as a test weight of 1.5 m in length and 1 m in height. There are two gate lanes that help drive through the gate speed, evening out and reducing slippage. The presence of drive rails as drive lines for DC motors. Inside the controller box is a DC motor, an Arduino Uno, which acts as the DC motor controller while the MCU node communicates with the Android application. Use of limit switches to control the state of the gate when it is opened or closed. When the gate touches the limit switch, the DC motor stops.

D. Android Application Design Plan

Android Application Scheduling is required for the Scheduling Tool to work as designed. The purpose of this plan is to design the application software used to operate the system. Below is the plan used by the tool shown in Fig. 5.

To control and monitor this tool using an IoT MQTT panel application, the IoT MQTT panel application provides control, monitoring, and graphics capabilities that can be connected to an Arduino, allowing home selectors to easily move fences using the application. make it possible.

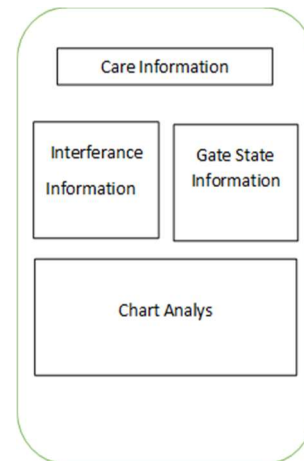


Figure 6. Monitor displays

III. RESULTS AND DISCUSSION

A. System Planning Results

System planning results are divided into hardware planning results and software planning results. The hardware design results used a gouge fence with a fence frame length of 1.5m, a height of 1m, a track or rail length of 3m for the wheels, and four wheels. This gate is connected to a DC motor gear with gears and chains as the driving means for the gate.



Figure 7. Implementation of the whole tool

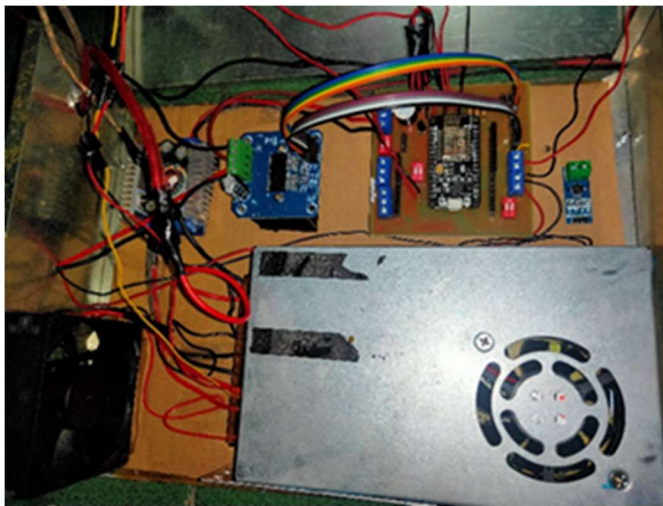


Figure 8. Components in the electronics box

Based on the Fig. 8 above, the components are arranged in a box containing an arduino uno, node mcu esp8226, horn, resistor, cooling fan, buck regulator, motor driver, sensor current, LCD display, LED lights, safety circuit and power supply. While the components not included in the box are the limit switch and the geared DC motor. Some of the above components will be assembled to become access control devices.

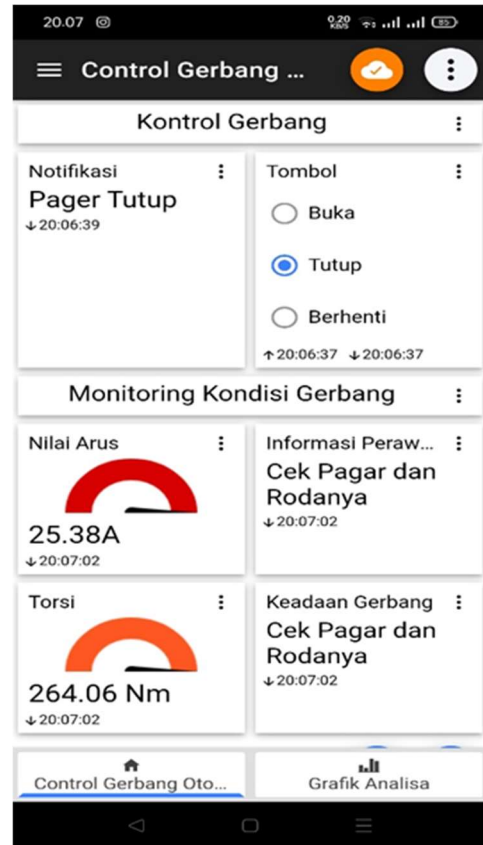


Figure 9. Application implementation of the control menu section

In Fig. 9 above, the home page shows information about the status of the tool and the buttons. This page contains information about the state of the tool, including when the gate opens, closes, stops, while regarding maintenance information data, e.g. blocked instruction, current value (A) and torque (Nm).

In Fig. 10 above, the graph page shows the current and torque data values monitored over time. The graph display is presented on a separate page after successfully pressing the graph analysis button below. Users can monitor tool development data through this graphical display.

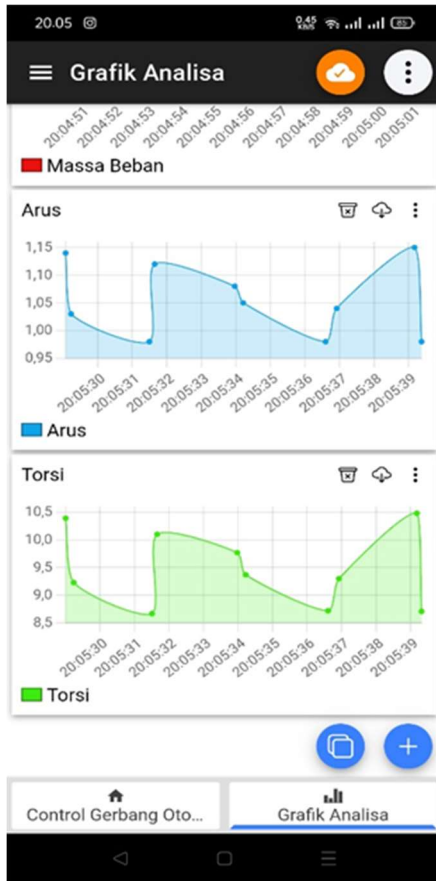


Figure 10. Implementation of the graph monitor part in the application

B. Research Data Results

Perform a load test with a torque wrench to obtain the maximum torque value. When the mass of the object is added, the current value increases, and when controlled by the ADC voltage, the torque increases further. After testing the torque wrench at ground, the ADC voltage at Tosi, and the current at the load, I obtained the following data: Regression line formula:

$$y = mx + C \tag{1}$$

The value of m can be obtained from the formula :

$$m = - \frac{n\sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i) (\sum_{i=1}^n y_i)}{n(\sum_{i=1}^n x_i^2) - (\sum_{i=1}^n x_i)^2} \tag{2}$$

In the meantime, to get the value of c, use :

$$c = \frac{(\sum_{i=1}^n y_i)}{\text{amount of data}} - \frac{(\sum_{i=1}^n x_i)}{\text{amount of data}} \tag{3}$$

TABLE I
TEST RESULTS LOAD MASS OF TORQUE WRENCH LOCK

Load (Kg)	Torque Wrench Lock (Nm)
1	9,81
2	19,61
3	29,42
4	39,22
5	49,03
6	58,83
7	68,64
8	78,44
9	88,23
10	98,05
11	107,86
12	117,66
13	127,47
14	137,37
15	147,08

One of the ways to collect the data in Table 1 for this study is to measure the load with a torque wrench to produce the maximum torque value. Measurements are made by attaching it to a socket on a post that already has a socket for mounting the shaft end. Set the load value by rotating the torque wrench handle to be used on the force wrench according to the load used. If the load used is in the force wrench setting, the wrench will click.

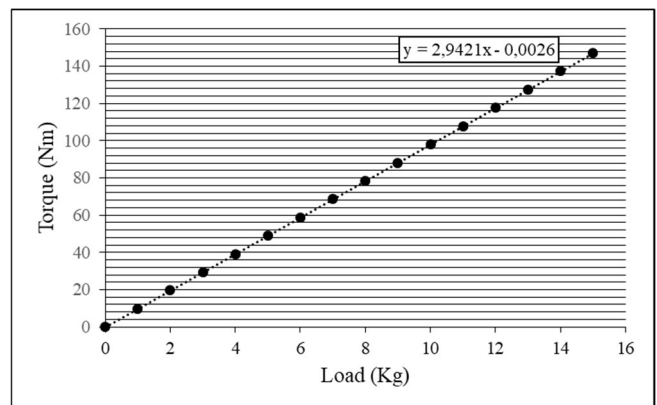


Figure 11. Graph relationship load with torque wrench

From the line graph above, we can see that the load on the torque wrench is directly proportional. The greater the given load value, the greater the value of the torque wrench. The plot above has a linear line with values $y = 9.8068x - 0.0070$ and a regression value of 1.

TABLE II
TEST RESULTS CURRENT SENSOR FOR LOAD MASS AND TORQUE

Load (Kg)	Torque (Nm)	Voltage (V)	Current (A)
1	9,81	12	1,28
2	19,61	12	2,56
3	29,42	12	3,84
4	39,22	12	5,12
5	49,03	12	6,41
6	58,83	12	7,69
7	68,64	12	8,97
8	78,44	11,9	10,25

9	88,23	11,9	11,53
10	98,05	11,9	12,82
11	107,86	11,9	14,1
12	117,66	11,9	15,38
13	127,47	11,8	16,66
14	137,37	11,8	17,96
15	147,08	11,8	19,23

Get data from table 2 which includes performing current measurements with different load masses using current sensors to generate maximum current value. Measurements are made by attaching a rope to the motor and the load forming a pulley. Already advanced mounted on the gear while the lower end is attached to the load used.

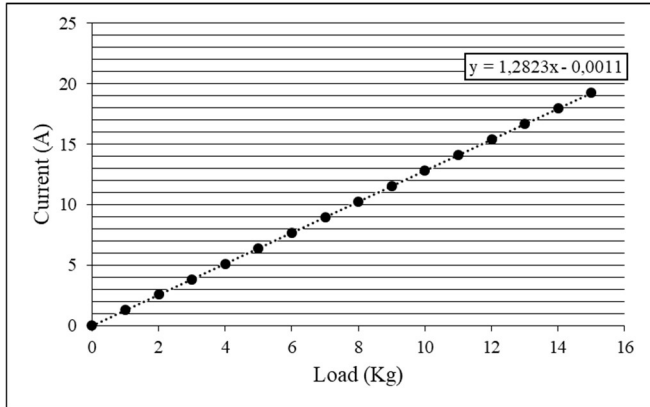


Figure 11. Graph relationship load with current

From the line graph above, we can see that the load on the current is directly proportional. The greater the given load value, the greater the value of the current. The plot above has a linear line with values $y = 1.2823x - 0.0061$ and a regression value of 1.

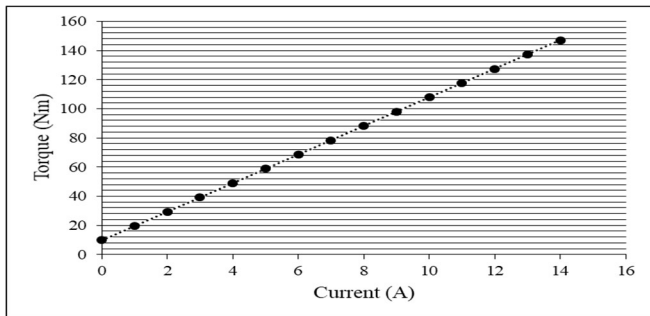


Figure 12. Graph relationship current with torque

From the line graph above, we can see that the current on the torque is directly proportional. The greater the given current value, the greater the value of the torque. The plot above has a linear line with values $y = 7.648x + 4.9323$ and a regression value of 1.

TABEL III
TEST RESULTS VOLTAGE (ADC) OF TORQUE

Voltage (ADC)	Torque (Nm)	Value Adc
0	0	0
1,25	38,52	56
2,5	74,94	112
3,75	115,76	168
5	140,82	255

Data retrieval from Table 3 includes performing current measurements with different load masses using current sensors to generate the maximum current value. The measurement is made by coupling a potentiometer for 0V-5V adc voltage input with an output value of 0-1024 or 0-255 for the purpose of controlling the variable voltage drive of the motor. Measurements are made by attaching a rope to the motor and the load forming a pulley. Already advanced mounted on the gear while the lower end is attached to the load used.

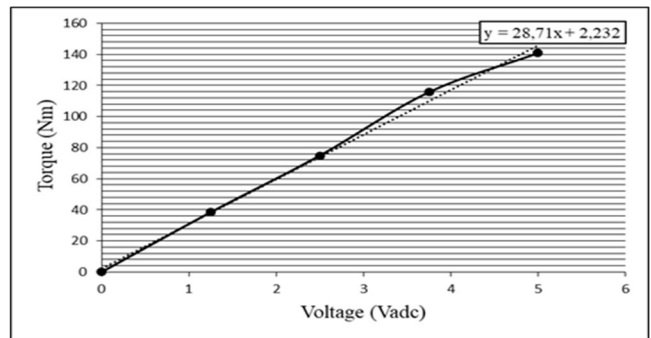


Figure 13. Graph relationship Voltage (ADC) with Torque

From the line graph above, we can see that the voltage (adc) on the torque is directly proportional. The greater the given voltage (adc) value, the greater the value of the torque. The plot above has a linear line with values $y = 28.71x + 2.232$ and a regression value of 0,99.

C. Quality of Service (QoS) Analysis Results

Quality of Service (QoS) testing aims to find the quality of the network and define the system by measuring several QoS parameters such as throughput and delay.

TABEL IV
WIRESHARK STATISTICS

Measurement	Captured	Displayed	Marked
Packets	1537	80 (5,2%)	--
Time Span, s	82.153	52.722	--
Average pps	18.7	1.5	--
Average packet size, B	203	114	--
Bytes	31:714	9096 (2.9%)	0
Average bytes/s	3794	172	--
Average bits/s	30k	1380	--

Based on table iv, 3 experiments were performed, the total number of packets was 1537 packets and the time span was 82.153 seconds. Tested with Wireshark. The data obtained from

Wireshark is the whole data communication. Throughput usage in data communication refers to the following formula:

$$\text{Throughput} = \frac{\sum \text{packets sent}}{\text{time span}} \quad (4)$$

$$\text{Throughput} = \frac{1537}{82,153}$$

$$\text{Throughput} = 18,7089 \text{ byte/s}$$

$$\text{Throughput} = 18,7089 \times 8 \text{ bit}$$

$$\text{Throughput} = 149,6719 \text{ bit/s}$$

The calculated throughput utilization when using full data communication when opening and closing the motor is 149,672 bps.

Based on the data above, a 1x test was performed to retrieve 79 data points, with a total delay of 52.721927 seconds. Testing is done in Wireshark. All data communication is based on the acquired data. Average latency for data communication is related to the following formula:

$$\text{Average Delay} = \frac{\sum \text{Delay}}{\sum \text{Packet Received}} \quad (5)$$

$$\text{Average Delay} = \frac{52,721927}{1537}$$

$$\text{Average Delay} = 0,03430s \times 1000$$

$$\text{Average Delay} = 34,3ms$$

Next, the average delay for the entire data communication with the motors open and closed is 0.0343s.

IV. CONCLUSION

Based on the design, observations and tests performed, the following conclusions can be drawn.

An Android-based smartphone can remotely control the gate through an interface on the smartphone screen over the internet from inside or outside the home using a WiFi network. A current sensor that detects load fluctuations and gate status is tested, and the gate status is determined from the current value for maintenance. When tested with a mass load variation from 1 to 15 kg, an average torque value of 78.44 Nm was obtained at an average current value of 10.25 A, resulting in data communication performance using the mqtt protocol, the calculation of the throughput value shows a very good performance. is obtained. At 149.6719, the average delay value is 34.3ms.

For further investigation, consider the design of the gate, the gear ratio of the drive motor, and the appropriate detection sensor for the gate. Also note that this control gate design uses an internet-based Android application over a Wi-Fi network when the internet is shut down. For further research, it can be developed to add a local server without using the Internet or based on a wide area network.

REFERENCES

- [1] Usman, A. A. Rahmansyah, and N. F. Apriadi, "Rancang Bangun pagar Otomatis dengan Finger Print Berbasis Mikrokontroler," *Jurnal Teknologi Terapan*, vol. 3, 2017.
- [2] F. B. Setiawan and Magfirawaty, "Securing Data Communication Through MQTT Protocol with AES-256 Encryption Algorithm CBC Mode on ESP32-Based Smart Homes," *2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE)*, 2021, pp. 166-170.
- [3] A. Budiyanto, G. B. Pramudita, and S. Adinandra, "Kontrol Relay dan Kecepatan Kipas Angin Direct Current (DC) dengan Sensor Suhu LM35 Berbasis Internet of Things (IoT)," *Jurnal Ilmiah Elektronika*, vol. 19, pp. 43-54, 2019.
- [4] Y. E. Anwar, N. Soedjarwanto, and A. S. Repelianto, "Prototype Penggerak Pintu Pagar Otomatis Berbasis Arduino Uno ATMEGA 328P dengan Sensor Sidik Jari," *Jurnal Rekayasa dan Teknologi Elektro*, vol. 9, 2015.
- [5] A. Syofian, "Pengendalian Pintu Pagar Geser Menggunakan Aplikasi Smartphone Android dan Mikrokontroler Arduino Melalui Bluetooth," *Jurnal Teknik Elektro ITP*, vol. 5, 2016.
- [6] A. Eleyan and J. Fallon, "IoT-based Home Automation Using Android Application," *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1-4.
- [7] S. P. Santosa, and R. M. W. Nugroho, "Rancang Bangun Alat Pintu Geser Otomatis Menggunakan Motor DC 24V," *Jurnal Ilmiah Elektro*, vol. 9, no. 2, 2021.
- [8] E. I. Nwankwo, E. N. Onwuka, M. David, and Z. Suleiman, "Hybrid MQTT-COAP Protocol for Data Communication in Internet of Things," *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, 2020, pp. 1-5.
- [9] A. Velinov, A. Mileva, S. Wendzel and W. Mazurczyk, "Covert Channels in the MQTT-Based Internet of Things," in *IEEE Access*, vol. 7, pp. 161899-161915, 2019.
- [10] A. Banks and R. Gupta, "MQTT Version 3.1. 1," *OASIS standard*, vol. 29, 2014.
- [11] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, and F. J. Pamungkas, "Penerapan protokol MQTT pada teknologi WAN (studi kasus sistem parkir Universitas Brawijaya)," *Jurnal Informatika Mulawarman*, vol. 12, no. 2, pp. 69-75, Sept. 2017.
- [12] A. W. Ibrahim, T. W. Widodo, and T. W. Supradi, "Sistem Kontrol Torsi pada Motor DC," *IJEIS*, vol. 6, no. 1, pp. 93-104, 2016.
- [13] H. Firdaus. "Rancang Bangun Penggerak Pintu Pagar Geser Menggunakan 12 volt Direct Current (DC) Power Window Motor Gear," *Jurnal Media Teknologi*, vol. 4, no. 2, pp. 155-164, March 2018.

- [14] "Arduino Uno data sheet," Arduino S.r.l, Italy.
- [15] L. J. Bradley and N. G. Wright, "Optimising SD Saving Events to Maximise Battery Lifetime for Arduino™/Atmega328P Data Loggers," in *IEEE Access*, vol. 8, pp. 214832-214841, 2020.
- [16] Arduino ISP Programming Tutorial. Accessed: Jul. 21, 2020. [Online]. Available: www.arduino.cc/en/Tutorial/ArduinoISP