Finger Robotic control use M5Stack board and MQTT Protocol based

Puput Dani Prasetyo Adi [1,2]
University of Merdeka Malang,
Malang, East Java, Indonesia[1]
Micro Electronics Research Laboratory,
Kanazawa University, Japan[2]
puputdani@merl.jp

Akio Kitagawa
Micro Electronics Research Laboratory
Kanazawa University
Kanazawa, Ishikawa - Japan
kitagawa@merl.jp

Junichi Akita
ifDL Laboratory
Kanazawa University
Kanazawa, Ishikawa - Japan
akita@is.t.kanazawa-u.ac.jp

***Abstract*—In this research, remote robot hand control using (Message Queuing Telemetry Transport) MQTT Protocol and M5Stack Board. The purpose of this research is to develop remote control technology on the robotic fingers, the function of the robotic fingers is used as an actuator that works to control or perform certain jobs, for example, a robot that can press buttons automatically with remote control, or can adjust the volume, pressing the lever, catching the ball, giving symbols like counting abacus in mathematics, and other robot arm activities that can be developed remotely using internet technology. In general, the MQTT Protocol works by providing string input. Furthermore, the position of the servo angle is represented by the values x and y or *θ1* and *θ2*. The results of the analysis in this research consisted of blockly programming and python used to control the robot finger and activities on MQTT Brokers which consisted of publish and subscribe and how the packet data was in MQTT Broker.**

Keywords—finger, robot, m5stack, MQTT, automation, mini servo

I. INTRODUCTION

Recently, the development of the Internet of Things (IoT) and its application in various fields continue to experience significant developments, in the world of industry known as the IoT-based Programmable Logic Controller (PLC), in the military world known as bomb squad robots, which are being worked on this research, furthermore, this research provides an approach to how bomb disposal robots can work. In the bomb disposal robot, the robot's hand works to pick and break the desired cable while to control the robot it needs to be seen that the robot's vision factor is using a camera, furthermore, the camera and hands are controlled through the IoT using an application and internet server. This research approaches the finger robot, the first step in building the more specific next steps. An important factor to consider when connecting to an internet network is the security factor [1] specifically for handling MQTT.

In research [2], MQTT is used on Passive Infrared (PIR) sensors connected to the M5 stack board and MQTT Broker, the approach is Smart home. MQTT is a very popular IoT protocol, but due to the large number of devices connected to MQTT Brokers, latency is undeniable, in research [3], developing MQTT with the term MQTT-ST, a protocol able to create such a distributed architecture of brokers, organized through a spanning tree. The protocol uses in-band signaling. In references [4], discussed the basic of MQTT, IoT [13],[14],[15], and Implementation and Analysis of QUIC for MQTT., MQTT-based IoT was also developed by references [5] which were applied to hospitals, a technology-based IoT will reduce the physical needs especially in health applications in hospitals, for example checking the condition of the patient's body or checking up, to become an IoT-based technology [5]. In addition to the hospital, the MQTT-based IoT application is used to detect motorcycle accidents [6], the sensor used is an accelerator sensor to detect the tilt, and GPS to determine the accident's position in realtime.

II. BASIC THEORY

II.1 Finger motion Mathematic approach

Finger motion illustrates The finger motion of the robot is depicted in the planar two-link manipulator shown in Figure 1. furthermore, the forward kinematics can be determined using plane geometry in equations 1 and 2 [7].

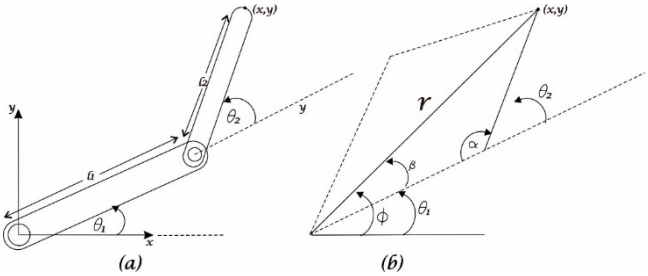

Fig 1. Inverse kinematics of a planar two-link manipulator.

Where the x and y values are represented in equations 1 and 2.

$$x = l1\cos\theta1 + l2\cos(\theta1 + \theta2) \quad (1)$$

$$y = l1\sin\theta1 + l2\sin(\theta1 + \theta2) \quad (2)$$

Values x and y are used to represent the values θ1and θ2. The polar coordinates in figure 1 are (r, ∅), and *θ2* is determined from the value of r.

$$r = \sqrt{x^2 + y^2} \quad (3)$$

In detail to determine the values of θ2 and α, seen in equation 4.

$$\theta2 = \pi \pm \alpha. \qquad \boldsymbol{\alpha} = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - r^2}{2l1l2}\right) \quad (4)$$

If $\alpha \neq 0$, there is a value of *θ2* corresponding to the radius, the second is called "Flip solution", which is shown by the dotted line in Figure 1b. furthermore, ∅ is used to determine θ1. Equation 5 is a calculation at θ1 and β.

$$\theta1 = \text{atan2}(y, x) \pm \beta \quad \beta = \cos^{-1}\left(\frac{r^2 + l_1^2 - l_2^2}{2l1r}\right) \quad (5)$$

978-1-7281-7226-2/20/$31.00 ©2020 IEEE

III. METHODS AND HARDWARE

One important factor in end device sensor devices is power consumption, on IoT and MQTT based sensor devices, power consumption management is needed considering the number of devices connected to the application server and sending data continuously. In research [8], research was conducted with the energy efficiency approach to the sensor node named EES-MQTT (Energy Efficient and Secured MQTT) This EES-MQTT technology uses MQTT.fx simulation tool and the Eclipse Paho. in research [9], research on memory analysis of several IoT-based IoT applications, i.e., Mosquitto, Paho MQTT client, and MQTT fx. research [10] also analyzes memory on MQTT using Transport Layer Security (TLS). Moreover, research [10] uses fuzzy logic methods to identify the possibility of DoS Attack on MQTT, this plays a big role in the area of MQTT based Internet security. Moreover, in other studies [11], based on the AugPAKE security algorithm, in the research the topic is IoT security with MQTT Based. M5Stack is a board that supports the IoT technology, MQTT-based internet servers, and uses MQTT Brokers. Parts of M5Stack are very complete ie, 1W Speaker, CP2104, 3D Antenna, ESP32 SH200Q + BMM150, Type C USB, Grove (I2C), TF Carder, Ip5306 (I2C), CP2104, LED Bar, Grove (I / O), GROVE (UART), and Microphone, this is a Powerful board [Figure 2].

Fig 2. M5Stack board (m5stack)

To connect to the Finger robot or hand robot, a minimum of 10 servos is required to be able to connect to the mini servo at the finger position. An additional 9 volts of power is needed because to handle the servo that will be running, increasing the ability of the M5Stack battery cannot handle 10 mini servos [Figure 3].

Fig 3. 12 Channel Servo for M5Stack board (m5stack)

Furthermore, an important part of the robot finger is the forming part of the finger, and this is a small and complicated part, for that it needs a careful installation process, the possibility of reversing can occur. This part of the finger starts from the small part of the tip to the large part or base of the finger [Figure 4], [Figure 5].

Fig 4. robot finger forming components

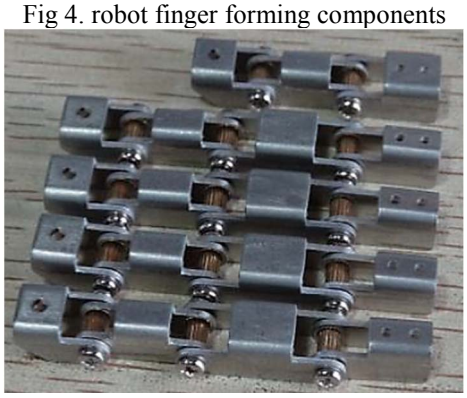

Fig 5. A Five finger parts robot components

This part is arranged like a chain on a bicycle, so it looks neat, with the shape and size of the different finger-forming components. After this component is formed, then the finger will look weak because no support or puller is elastic cable to connect to the servo motor [Figure 6], [Figure 7]. Therefore, after properly installed, the robot's fingers will look stiff and ready to be pulled by the angle of the servo motor in the M5S5tack programming or blockly.

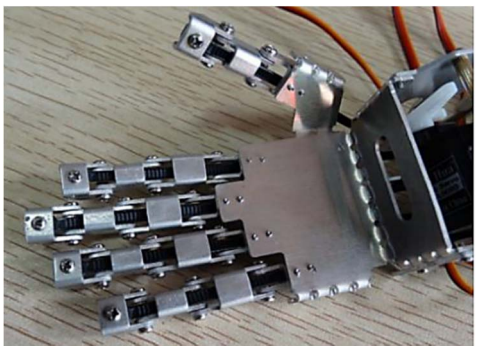

Fig.6 Hand Robot is well connected to the palm

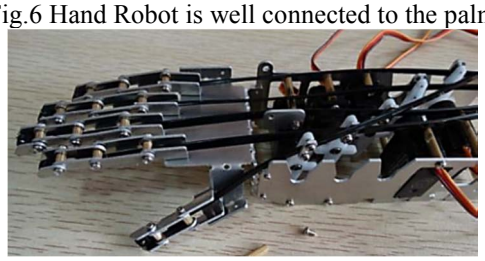

Fig 7. A series of robot hands

The next device is 5 mini servo [Figure 8], which will be part of 5 fingers.

Fig 8. Mini Servo

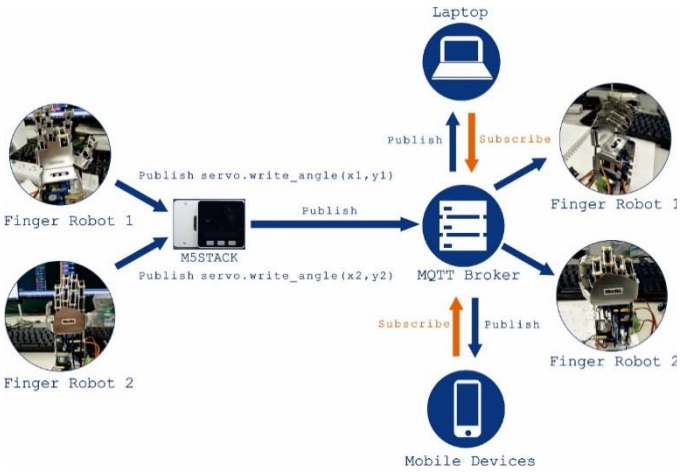

Fig.9 Architecture Network Design on this research

Figure 9 is the method used in this research, where Finger Robot has 5 mini servo motors that are controlled by M5 stack board and servo board for M5 stack board compatible. Furthermore, the M5 stack board is equipped with a WiFi module as ESP8266. So that the process of publishing or uplink to the server occurs, in this case what is published is the position or angle of the servo represented by the values x_1 and y_1. For angles on the servo, it varies depending on the shape of the finger, e.g., when the finger is grasping or showing the number 2. Figure 8 represents the entire research to be done.

Figure 10 is a flowchart showing step by step arm robot or finger robot that can be controlled via the internet using the M5 stack board. Furthermore, Pseudocode_1 is a programming language to control the finger robot via M5stack board, M5stack has two kinds of programming views, i.e., Blockly programming, and Python. This is a display of python on Pseudocode 1.

```
 1.  Import Library for M5Stack
from m5stack import *
from m5ui import *
from uiflow import *
import module
   2.   Determine the Screen Color
setScreenColor(0x222222)
   3.  Determine the Servo module
servo = module.get(module.SERVO)
   4.  Determine the Label
label1 = M5TextBox(115, 108, "Arm Robot",
lcd.FONT_Default,0xFFFFFF, rotate=0)
while True:
   5.  Determine the servo angle for 5 finger
       (2x), and time (s)
  5.1 time number 1 for finger move
  servo.write_angle(1, 270)
  wait(1)
  servo.write_angle(1, 0)
  wait(1)
  servo.write_angle(0, 90)
  wait(1)
  servo.write_angle(0, 0)
  wait(1)
  servo.write_angle(7, 100)
  wait(1)
  5.2 time number 2 for finger move
  servo.write_angle(7, 0)
  wait(1)
  servo.write_angle(6, 90)
  wait(1)
  servo.write_angle(6, 0)
  wait(1)
  servo.write_angle(4, 90)
  wait(1)
  servo.write_angle(4, 0)
  wait(1)
  wait_ms(2)
---------- Pseudocode_1---------
```

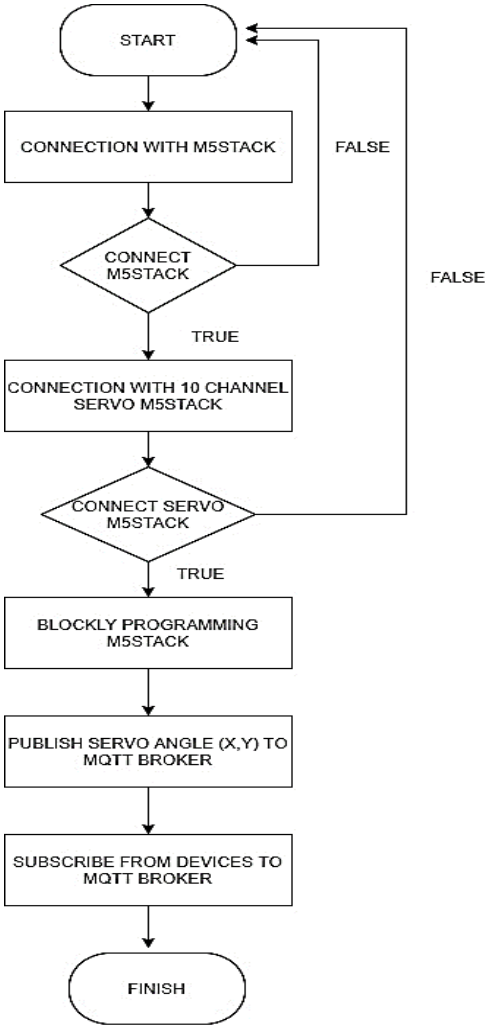

Fig.10. Flowchart the research

The display programming in Figure 11, can be seen in the default Software M5Stack, i.e., Ui Flow v.1.3.2, on this software, then setting the blockly i.e., Set Servo to rotate 0 to 270 degrees in the case loop. Furthermore, Figure 12 shows the python display of Figure 11.

Fig 11. Blockly Programming on M5Stack

Fig 12. Python Programming on M5Stack

IV. Result And Discussion

IV.1. Analysis on Finger robot

For examples of the results of robot movements can be seen in Figure 13, examples of movements are at the fingertips. Moreover, to adjust the speed of the robot finger, initialization of the timer is required, as shown in Figure 14.

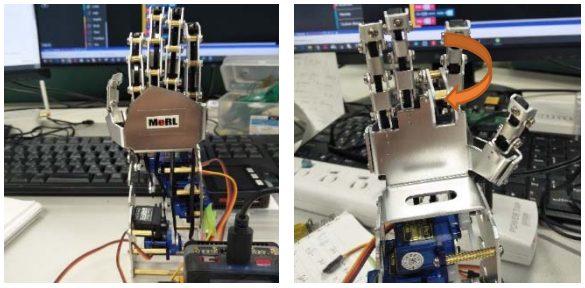

Fig.13 Initial position of the robot finger and examples of movements at the fingertips

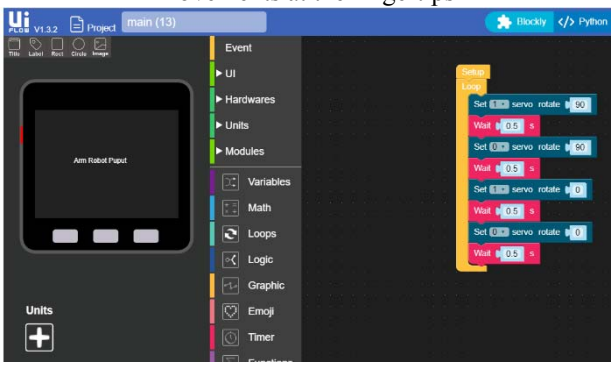

Fig.14 initialize the timer to adjust the speed of the robot finger movement

12 Channel M5Stack servo pins can be initialized, 12 M5Stack Servo Pins are Pin 0-11 that can be connected to the Arm Robot. In Arm Robot, there are 5 servo motors, in this case, 7 servo pins are still free or unused. In the example program above, sample initialization has done on servo pins 0 and 1, in the example program time used to move from the 90 degrees rotate is 0.5 second on each servo motor, therefore, 0.5 seconds alternate each robot finger will bend and return parallel or straight his position. furthermore, if it is made to hold a grasp it means that on fingers 1 to 5 a relatively fast time eg 0.3 seconds on the mother's finger, middle finger, ring finger, and little finger, therefore, this process occurs in the looping. Examples of programs are as follows on Figure 15, and Figure 16 is a Finger robots with a grasping position, and Figure 17 is a grasping position and returning to its original position.

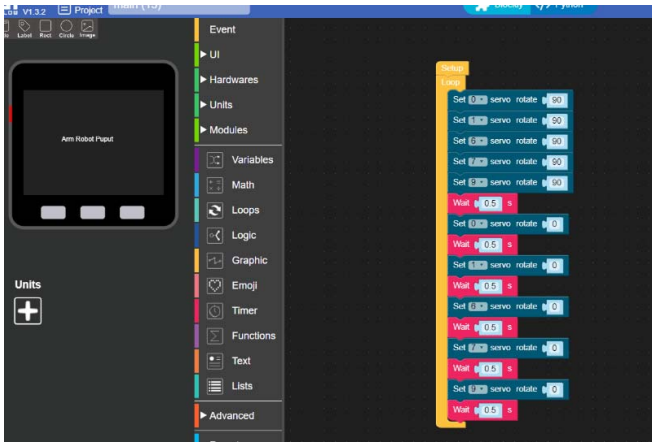

Fig.15. Program for making robots with a grasping position

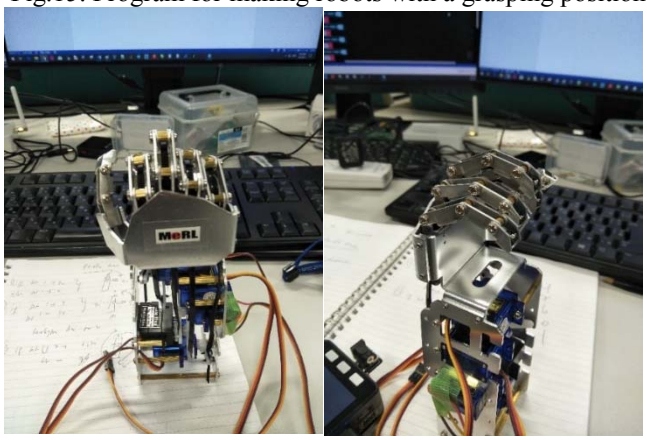

Fig.16. Finger robots with a grasping position

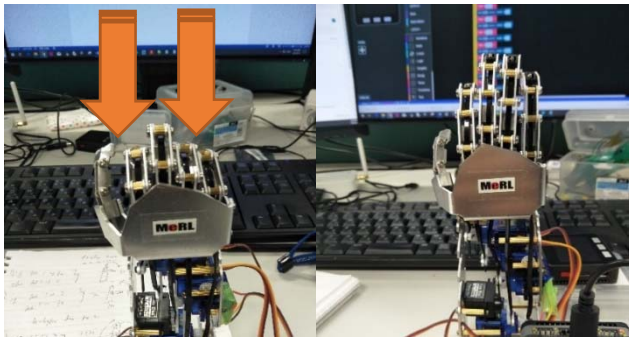

Fig.17. a grasping position and returning to its original position

```
  servo.write_angle(0,   90)   ->   Finger   1
Position
  servo.write_angle(1,   90)   ->   Finger   2
Position
  servo.write_angle(6,   90)   ->   Finger   3
Position
  servo.write_angle(7,   90)   ->   Finger   4
Position
  servo.write_angle(9,   90)   ->   Finger   5
Position
  wait(0.5)
  servo.write_angle(0, 0) -> Finger 1 Position
  wait(0.5)
  servo.write_angle(1, 0) -> Finger 2 Position
  wait(0.5)
  servo.write_angle(6, 0) -> Finger 3 Position
  wait(0.5)
  servo.write_angle(7, 0) -> Finger 4 Position
  wait(0.5)
  servo.write_angle(9, 0) -> Finger 5 Position
  wait(0.5)
  wait_ms(2)
```

Program for making robots with a grasping position and returning to its original position

The process of sending data from End devices or Arm robots on M5Stack board to MQTT Brokers depending on an internet connection or the signal Power (dB) on ESP32 M5Stack onboard, the possibility of interference or noise from other radio frequency (RF) devices may occur, such as another WiFi, LoRa RF, or Bluetooth that is active. Therefore, it makes packet loss so that the data bytes on receive is lower than the data sent. Comparison of publish data send from M5 Pack board and publish receive or subscribe from devices to M5 This stack board can be seen in Figure 18 and Figure 19. Moreover, packet loss (bytes) on MQTT Brokers is shown in Figure 20.

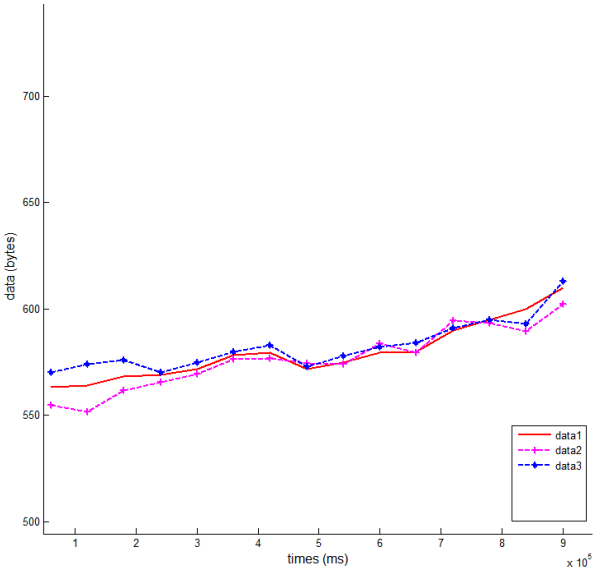

Fig 18. Publish send data (bytes) on MQTT Broker

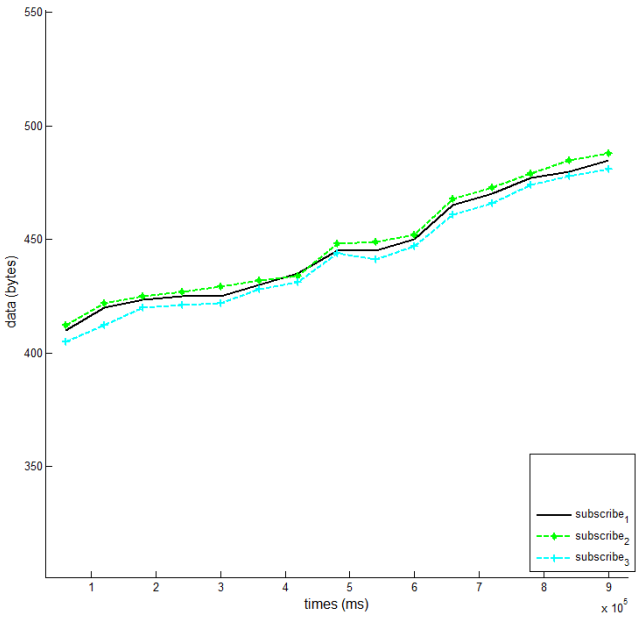

Fig 19. Publish receive data (bytes) on MQTT Broker

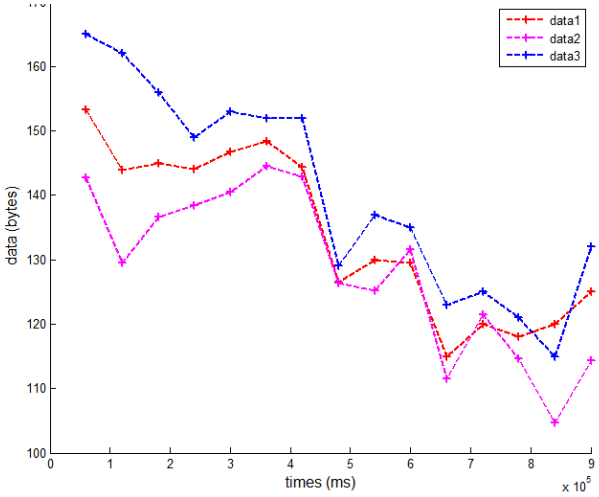

Fig 20. Packet Loss (bytes) on MQTT Broker

Publish send data (bytes) comes from a finger robot connected to the M5 stack board, the data range is ~ 550 bytes and continues to increase up to ~ 600 bytes of published data send, while the publish receive or subscribe data from deceives or laptops or mobile devices to MQTT brokers ranging from ~ 400 bytes to ~ 480 bytes of data, the difference in publish send and receive on MQTT Brokers is caused by interference, noise or interference Radio Frequency or other Wireless devices that cause packet loss on the onboard M5Stack ESP32 device.

The complete data can be seen in Table I and II, this data comes from mosquito version 1.5.8. MQTT Broker is used in this research, MQTT Broker is compatible with the M5stack board and represented in figures 18, 19, and 20.

Table I. Data I M5Stack board on MQTT Broker

Times (minutes)	connections/+	bytes/received/+	bytes/sent/+	messages/received/+	messages/sent/+
1	0.02	481.37	2806.62	65.3	119.94
5	0.1	478.42	2772.06	64.52	119.55
15	0.29	373.79	2077.18	49.38	93.08

Table II. Data II M5Stack board on MQTT Broker

Times (minutes)	connections/+	dropped/+	publish/received/+	publish/sent/+	sockets/+
1	0.02	0	58.37	112.8	0.01
5	0.1	0.00	57.64	112.57	0.1
15	0.29	0	44.4	87.60	0.13

V. CONCLUSION

The robot finger can be controlled properly using the M5Stack Board and the M5Stack 12 Channel servo, with the help of an adapter or 9 volt Power supply, AC / DC to help provide power supply to 5 active servo motors. The application server or internet server is used is MQTT Broker, MQTT Broker processes the transmitting data or angle of a servos motor or publish to MQTT Broker process, and MQTT Broker transmits the data to the devices (laptops or mobile devices), and the devices subscribe the data go to the MQTT Broker, this process can be called uplink and downlink, the difference in publish send and receive ranges from ~ 100 bytes, due to packet loss due to signal interference on the onboard M5Stack ESP32 with router or internet connection.

ACKNOWLEDGMENT

Thanks for MeRL and ifDL Laboratories by Prof.A.Kitagawa and Prof. J.Akita for guidance, tools, ideas, research sites, and all the resources that have been given to the author to complete this research, thank you. provide all the device equipment, so that this research could be completed and tested properly.

REFERENCES

1. Francesco Buccafurri, Vincenzo De Angelis, and Roberto Nardone,"Securing MQTT by Blockchain-BasedOTP Authentication", Sensors MDPI, DOI :10.3390/s20072002, 2020
2. Puput Dani Prasetyo Adi, Akio Kitagawa, "A Review of the Blockly Programming on M5Stack Board and MQTT Based for Programming Education", 2019 IEEE 11th International Conference on Engineering Education (ICEED), DOI: 10.1109/ICEED47294.2019.8994922, 2019
3. Edoardo Longo, Alessandro Enrico Cesare Redondi, Matteo Cesana, Andrès Arcia-Moret, Pietro Manzoni, "MQTT-ST: a Spanning Tree Protocol for Distributed MQTT Brokers", arXiv:1911.07622, 2019
4. Puneet Kumar, Behnam Dezfouli, "Implementation and Analysis of QUIC for MQTT", Internet of Things Research Lab Department of Computer Engineering, Santa Clara University, February 2019
5. A.Irfana Kader Nisha, K.Janani, G.Rathi Devi, S.Meivel, "Blynk and MQTT Based Smart Hospital System", International Journal of Disaster Recovery and Business Continuity, Vol.11, No. 1 (2020), pp. 69-79
6. Sudha Senthilkumar, K. Brindha, Shashank Bhandari, "Vehicle accident management and control system using MQTT",International Journal of Advances in Applied Sciences (IJAAS), Vol. 9, No. 1, March 2020, pp. 1~11, ISSN: 2252-8814, DOI: 10.11591/ijaas.v9.i1.pp1-11
7. Richard M. Murray, Zexiang Li, S. Shankar Sastry, "A Mathematical Introduction to Robotic Manipulation", CRC Press, 1994.
8. Selvi M., Gayathri A., Santhosh Kumar SVN, Kannan A,"Energy Efficient and Secured MQTT Protocol using IoT", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-9 Issue-4, February 2020, DOI: 10.35940/ijitee.B6264.029420
9. Anal Shah, Palak Rajdev, Jaidip Kotak, "Memory Forensic Analysis of MQTT Devices", arXiv:1908.07835v1, Cornell University, 2019
10. Haripriya A. P., and Kulothungan K. "Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things", Journal on Wireless Communications and Networking (2019) 2019:90, DOI : 10.1186/s13638-019-1402-8
11. Marco Calabretta, Riccardo Pecori, Massimo Vecchio, and Luca Veltri, "MQTT-Auth: a Token-based Solution to Endow MQTT with Authentication and Authorization Capabilities", JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 14, NO. 4, DECEMBER 2018, (DOI): 10.24138/jcomss.v14i4.604
12. Rizka Reza Pahlevi, Parman Sukarno, and Bayu Erfianto, "Implementation of Event-Based Dynamic Authentication on MQTT Protocol", Jurnal Rekayasa Elektrika, VOLUME 15 NOMOR 2, Agustus 2019.
13. Puput Dani Prasetyo Adi and Akio Kitagawa, "Quality of Service and Power Consumption Optimization on the IEEE 802.15.4 Pulse Sensor Node based on Internet of Things" International Journal of Advanced Computer Science and Applications(IJACSA), 10(5), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0100518
14. Puput Dani Prasetyo Adi and Akio Kitagawa, "ZigBee Radio Frequency (RF) Performance on Raspberry Pi 3 for Internet of Things (IoT) based Blood Pressure Sensors Monitoring" International Journal of Advanced Computer Science and Applications(IJACSA), 10(5), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0100504
15. Puput Dani Prasetyo Adi and Akio Kitagawa, "Performance Evaluation of E32 Long Range Radio Frequency 915 MHz based on Internet of Things and Micro Sensors Data" International Journal of Advanced Computer Science and Applications(IJACSA), 10(11), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101106