A Review of the Blockly Programming on M5Stack Board and MQTT Based for Programming Education

Puput Dani Prasetyo Adi
Micro Electronics Research Laboratory (MeRL)
Kanazawa University
Ishikawa, Japan
puputdani@merl.jp , ORCID ID : 0000-0002-5402-8864

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

***Abstract*— The Blockly Programming is one way to represent Programming code or script languages on High-Level Programming like Python, C / C ++ Language to easily understandable image forms, these images have different forms that represent if, for, while, input, output values, etc. in this research, an analysis using M5STACK Board can be programmed using the Arduino IDE or flow.m5stack based on Blockly Programming. Basically, Blockly programming is used for Programming Education. In this research, the Blockly programming application is used flow.m5stack to run the Motion Sensor, basically, there are many sensors that can be used on the M5Stack board as well as the Arduino Board with various shapes and variants. In this research, the M5Stack Microcontroller application is easily explained using Blockly Programming, so programming feels easier and more enjoyable. Microcontroller based M5STACK board can be developed in the Internet of Things (IoT) application using MQTT Protocol. furthermore, Message from the Queuing Telemetry Transport (MQTT) The protocol that is used to represent commands and processes and produces the appropriate output in real time via Internet protocol (TCP / IP). M5Stack is using the MQTT API Cloud server, and its application to Smart home, furthermore, the M5Stack client board sends data to the Cloud MQTT or MQTT Broker using the establishment of IoT on Smart home using MQTT Server and M2M Technology using multi M5Stack boards.**

Keywords*— Blockly, MQTT, Programming, IoT, Sensor, Education

I. Introduction

This research talks about the application of Blockly Programming on the M5Stack. a Blockly Programming is a type of Block-based Programming as well as constructing a puzzle, upright it will appear easier and faster to be understood by beginner students to learn Python, C or C ++. Accordingly, to this theme, i.e. education, Computer Web-Based Software and New Technology on education, at this time will discuss the role of Blockly Programming to espouse an IoT Technology on the M5 Pack Board. Blockly Programming is supportive in GUI based programming.

GUI (Graphical User Interface), format and tools in programming languages continue to be developed to the stage of improvement, besides that the development of the technology of the Internet of Things (IoT) is increasing rapidly following the development of the Industrial Revolution 4.0. Automation technology continues to grow with the internet Protocol technology used. several types of messaging protocols for IoT Systems are used as a research [20] i.e, MQTT, CoAP, AMQP, and HTTP, for M5Stack uses the MQTT Protocol. moreover, M5Stack board is a microcontroller type, M5Stack board is an Open-Source modular toolkit for IoT devices based on ESP-32-update version of ESP8266 with stackable, user-friendly IDE, besides using Blockly Programming in M5Stack Flow, also based on the Arduino Integrated Development of Environment (IDE) in C ++ [15]. the two parts discussed in this research are Blockly Programming on M5 Pack Flow and MQTT Protocol on M5Stack communication server.

MQTT (Message Queuing Telemetry Transport) Protocol is used as a device developer based on IoT (Internet of Things). IoT is a technology that is used to monitor energy usage (energy monitoring), device status and current status of sensors when monitoring e.g, moisture soil sensors in ornamental plants. in research [6] uses the LM35 as a humidity sensor and the data is stored in a MySQL database. The Raspberry Pi 3 is used as an Internet Gateway, a Pulse and Blood Pressure Sensor [11,13,21], the position status of motorized vehicles, the status density of road users, all of which can be monitored through internet or internet protocols or TCP / IP technology.

The use of messages or characters is made to determine the output and output can be monitored in a graph report and real-time status on the device. furthermore, IoT technology continues to evolve using various advanced devices such as RFID, Smart Sensors, Communication technologies, and Internet Protocols [2, 11]. Sensor data transmission using IoT Technology is one example of technology implementation in Industry Revolution 4.0, in the previous research, Ultrasonic sensor data transmission still uses limited Global System for Mobile Communication (GSM) technology with Base Transceiver Station (BTS) and this is local. this sensor is sent using GSM SIM900A technology on Microcontrollers, limited to the presence of BTS and experiencing many weaknesses [7].

In the study [3] a comparative analysis was carried out between the Transmission Control Protocol (UTP) and the User Datagram Protocol (UTP) and transport protocols to obtain analysis or performance over a narrowband IoT (NB-IoT) Link. MQTT can also be analyzed for its performance using HTTP [10]. These environments where networks have low bandwidth and high latency [1]. One parameter of IoT quality assessment is Energy and Quality of Service (QoS) such as Delivery Latency or delay. It is proven that it can reduce delivery latency by 55% [5]. MQTT has three levels of QoS with various sizes of payloads [10]. In this research, M5Stack Core ESP32 is used, this 5x5 cm square board is small and dynamic, using dual MCU system technology, i.e, M32u4 and M328P link by SPI. M5Stack Core ESP32 is a small module that has 3 main functions, i.e, the Control, Sensors, and Actuator modules. furthermore, M5Stack Core ESP32 can be programmed using 2 ways, UI Flow and Arduino IDE. UIFlow uses block programming that is useful, while the Arduino IDE uses the C ++ programming language. Therefore, from the above review, it can be seen that there are many roles of M5Stack board in supporting the development of IoT Technology by MQTT Protocol. consequently, that it can be said to learn M5Stack by directly learning Blockly Programming.

II. Methods

A. Flowchart

The flowchart on Fig. 1 shows how the system on the M5Stack Board works from beginning to end.

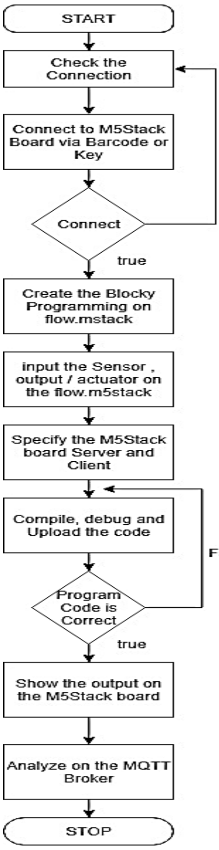

START
Check the Connection
Connect to M5Stack Board via Barcode or Key
False
Connect
true
Create the Blocky Programming on flow.mstack
input the Sensor , output / actuator on the flow.m5stack
Specify the M5Stack board Server and Client
Compile, debug and Upload the code
False
Program Code is Correct
true
Show the output on the M5Stack board
Analyze on the MQTT Broker
STOP

Fig 1. Flowchart System

B. Blockly Programming on M5Stack

Blockly Programming is the solution for beginner students to start learning C and Python programming languages because this programming is based on a form of blocks that is easy to apply following the shape of the block, as well as constructing puzzles. In blockly programming, there are commands in the programming script completely. The facilities include Logic, Loop, Math, Text, Lists, Color, Variables, and Functions [19].

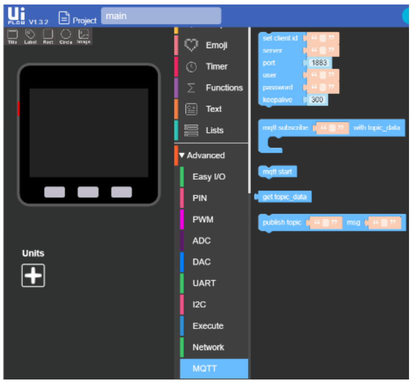
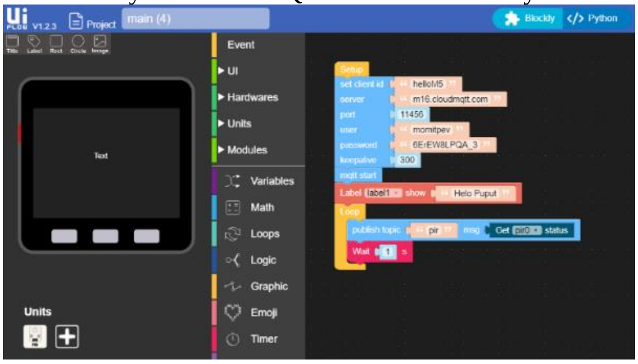


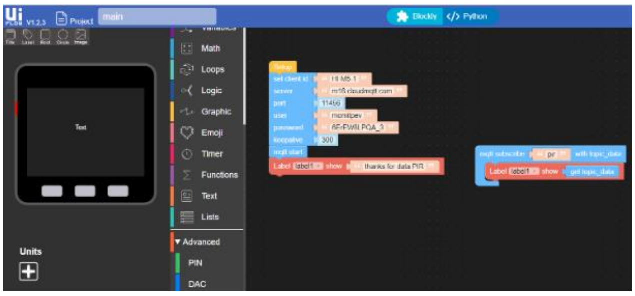
Fig.2 MQTT Blockly

Blockly programming can be exported or converted to other programming languages such as Javascript, Python, PHP, Lua, and Dart. Blockly Programming application on M5Stack, there are 2 parts, i.e, Client 1 and Client 2, furthermore, Client 1 and 2 are made to be able to communicate through cloudMQTT, the server used is m16.cloudmqtt.com, the parameter in MQTT Blockly is the Set Client id, server, port, user, password, keepalive, start to subscribe with a topic_data_mqtt start, get topic_data, publish topic and message. Fig. 2 show about M5Stack board blockly start from MQTT Parameter blockly.



Setup
set client id "helloM5"
server "m16.cloudmqtt.com"
port 11456
user "momitpev"
password "6ErEW8LPQA_3"
keepalive 300
mqtt start
Label label1 show "Helo Puput"
Loop
publish topic "pir" msg Get pir0 status
Wait 1 s

Fig 3. M5Stack Client 1 blockly code



Setup
set client id "Hi M5 1"
server "m16.cloudmqtt.com"
port 11456
user "momitpev"
password "6ErEW8LPQA_3"
keepalive 300
mqtt start
Label label1 show "thanks for data PIR"
mqtt subscribe "pir" with topic_data
Label label1 show get topic_data

Fig 4. M5Stack Client 2 block code

The two blocks on M5Stack Clients can be changed in Python form by moving the Blockly button to Python, so that the Blockly form can be seen in Python form, Fig. 3 and Fig. 4 show about the client 1 and 2 on M5Stack MQTT.

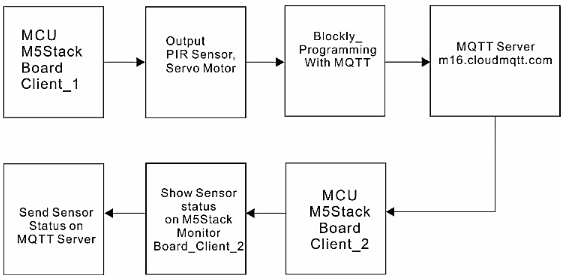

MCU M5Stack Board Client_1
Output PIR Sensor, Servo Motor
Blockly_ Programming With MQTT
MQTT Server m16.cloudmqtt.com
MCU M5Stack Board Client_2
Show Sensor status on M5Stack Monitor Board_Client_2
Send Sensor Status on MQTT Server

Fig. 5. Block Diagram System on this research

To facilitate understanding of this research approach can be seen in Fig. 5, Fig. 5 is the flow of thought in this research. The main topic is the use of MQTT on Blockly Programming. MQTT is Message Queuing Telemetry Transport, MQTT works through the TCP / IP protocol. Also called the protocol based on publishing or subscribing to the messaging protocol. There are 3 terms in MQTT Client communication and MQTT Server or Protocol Broker, namely Connect, Subscribe and Publish, i.e, Connect Acknowledge, Subscribe Acknowledgment and Publish Acknowledgment on Fig.7. Topic name, Client ID, User name and password are encoded as UTF- 8 Strings. furthermore, MQTT Can be seen in the Standard packet structure in Fig. 6. [14]

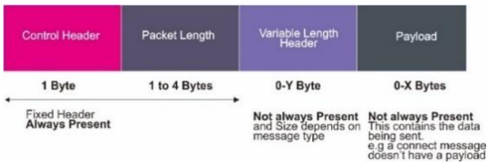

Control Header
Packet Length
Variable Length Header
Payload
1 Byte
1 to 4 Bytes
0-Y Byte
0-X Bytes
Fixed Header Always Present
Not always Present and Size depends on message type
Not always Present This contains the data being sent. e.g a connect message doesn't have a payload

Fig. 6. MQTT Standard Packet Structure

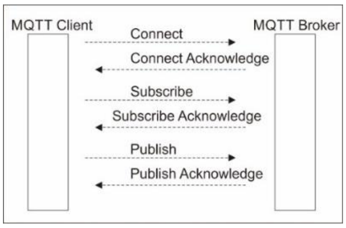


Fig. 7. MQTT Client to Broker Protocol

MQTT is one protocol that works by using binary bytes not string text. MQTT uses the Command and Command Acknowledge format.

III. The Hardware And Methods

A. *M5Stack Core ESP32 Board*

In Fig. 8 shows the M5Stack Core ESP32 board in the open state used in this research.

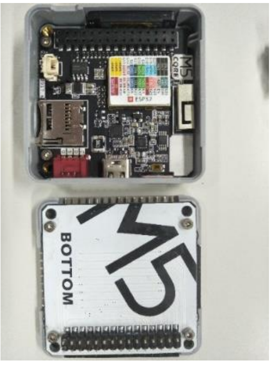


Fig. 8. M5Stack Core ESP32 Board

M5Stack Core ESP32 [13] is a complete board with the following specifications ie, ESP32 with MPU6050, MAG3110 16 M-FLASH and 4M-PSRAM, 2x15 Pin BUS, 3D Antenna, C-USB Type, Grove (I2C), TF Carder, IP5306 (I2C), CP2104, 9 DOF Sensors, 1 W Speaker, Microphone, LED Bar, Grove (UART), Grove (I / O), and 550 mAH Battery.

B. *Block Diagram on M5Stack IoT*

Fig. 9 shows a block diagram that illustrates the system in this research. There are 3 main parts, i.e. Sender, MQTT Broker and Receiver (devices). Data from sensors and UI Flow on M5Stack Client sent to MQTT Brokers are called publish.

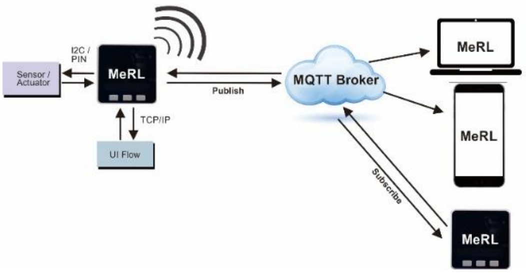
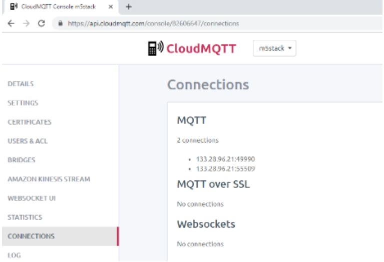


Fig. 9. Block Diagram System in this research

Subscribe is a term used to represent the connection between MQTT Brokers and Receiver devices e.g, Smartphone, Laptop Computer or M5Stack.. Fig. 13 is developed on Smart home using sensors that are scattered at home and sent to MQTT Brokers using MQTT Cloud. UI Flow initialized units are used for an e.g, Temperature sensor. After the initialization process is complete then the download process is done, based on a WiFi connection and MQTT Broker data can be received by other M5Stack Core ESP32 devices, and data can be displayed on 2 different devices through MQTT or MQTT Broker Cloud servers as the server. furthermore, In research [12] discussed MQTT on home automation systems using the Raspberry Pi as an Internet gateway or the main server.

C. *MQTT Brokers*

The MQTT Broker [9] is a server used on M2M i.e. M5Stack board devices, therefore, In this research MQTT Broker used is cloudmqtt, Cloud MQTT is used to hold data from M5Stack Core ESP32 board, then send data via TCP / IP to various devices connected to WiFi or Internet Access using SSID. furthermore, at MQTT Broker or MQTT Cloud contains data that will be used in UI Flow, i.e. server address, username, password and the port number on MQTT Cloud, it is determined how many devices can be connected, in this research, MQTT Cloud that can be connected is 5 devices. The appearance of the MQTT Cloud administration can be seen in Fig. 10.



Connections

MQTT

2 connections

- 133.28.96.21:49990
- 133.28.96.21:55509

MQTT over SSL

No connections

Websockets

No connections

Fig.10. MQTT Server

Menus included in MQTT Server include settings, certificates, user & ACL, Bridges, Amazon Kinesis Stream, Websocket UI, Statistics and Connection status. On this research will emphasize the statistic and connection status. The system to be built is shown in Fig.13. That is building a Smart home using the M5Stack board and MQTT Server (Cloud MQTT). To be able to connect to the MQTT Cloud, the UI Flow M5Stack board needs to be set on the Server, Port, User and password and client ID sections.

D. Passive Infrared Sensor Approach

The sensor used in this research is the Motion Sensor on Fig. 11, the motion sensor is a sensor that detects the movement that occurs around the sensor with a certain distance according to the capacity of the sensor, this sensor is used on lights that are connected with a 5 Volt relay. PIR Sensor Schematic can be seen on Fig. 12.

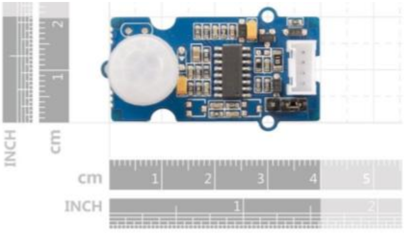


Fig. 11. PIR Sensor dimension

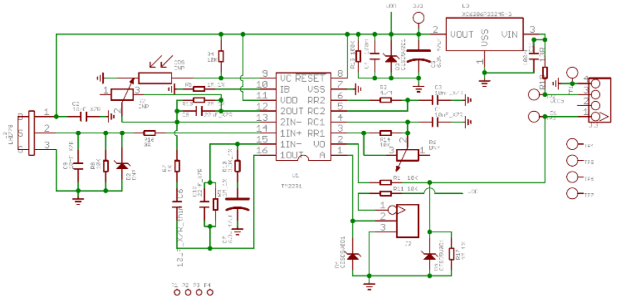


Fig. 12. PIR Sensor Schematic

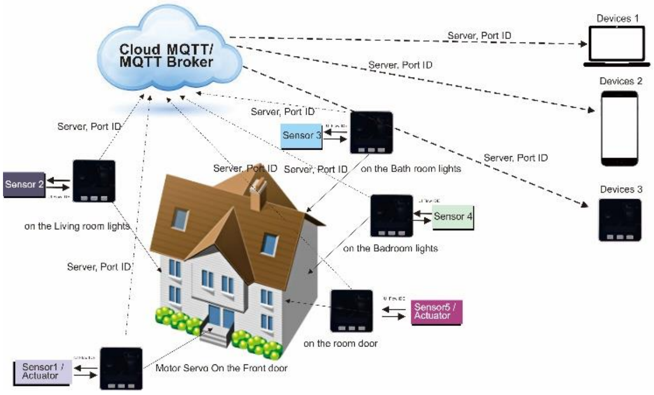


Fig. 13. Implementation of MQTT M5Stack to Smart Home

PIR Sensor is protected with a layer called Fresnel lens which allows more diffuse detection, in the PIR Sensor version used in this research is 120 degrees, PIR will work or provide a signal when detecting a motion, PIR Sensor detects a distance of 24.89 cm, when the Output signal selected, the M5Stack board converts analog signals to digital form. That is the value of 1 for detection and 0 when no motion is detected. More complete the PIR Sensor Specification can be seen in table 1.

TABLE I. PIR SENSOR SPESIFICATION

No	***Parameters***	***Spesification***
1	Interface	Grove Compatible Interface
2	Integrated Circuit (IC)	BISS0001 micro power PIR motion detector IC
3	Wivelength	7 - 14 um wavelength sensor
4	Voltage	3-5VDC
5	Dimension	20 x 40mm (0.78 x 1.57")
6	Detecting angle	120 degrees
7	Detection range	On this type is 9.8 inch or 24.89 cm detection range

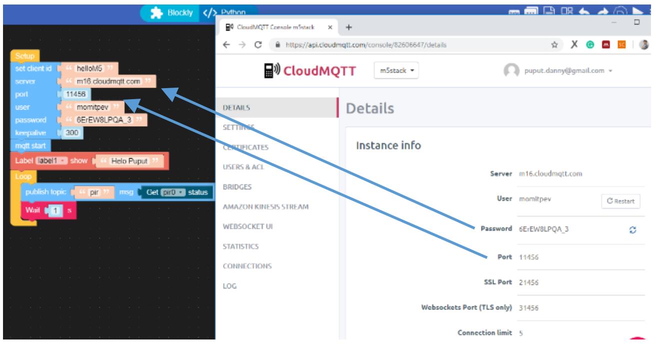
E. MQTT Security

MQTT Broker Security does not only use Username or ID and Password to be able to enter MQTT Brokers, but also media to do Authentication such as HiveMQ [15, 16]. Another method about IoT Security is SMQTT and SMQTT-SN is IoT security technology that uses augmented to existing MQTT based on Key / Cipher Protocol text Policy-Attributed Based Encryption (KP / CP-ABE) [4]. If you want to delete and replenish M5Stack with a new program, the M5Burner bootloader is needed.

IV. RESULT AND DISCUSSION

A. Blockly PIR Sensor Program Analyze on M5Stack

There are 2 types of programming offered by Flow.mstack, i.e. Blockly and Python, for Python experts is the choice, while Beginners choose to use Blockly because besides being easy and fast, there is no need to worry about minor errors in the syntax Python programming language. The Python programming language is a bit simpler than C ++ in the Arduino IDE, although that choice can be chosen. In this research, PIR sensors or motion sensors are used as sensors to detect motion. The status is pir0 = units.PIR (units.PORTB) on Fig.14.



Setup
set client id "helloM5"
server "m16.cloudmqtt.com"
port 11456
user "momitpev"
password "6ErEW8LPQA_3"
keepalive 300
mqtt start
Label label1 show "Helo Puput"
Loop
publish topic "pir" msg Get pir0 status
Wait 1 s

Details

Instance info

Server m16.cloudmqtt.com
User momitpev
Password 6ErEW8LPQA_3
Port 11456
SSL Port 21456
Websockets Port (TLS only) 31456
Connection limit 5

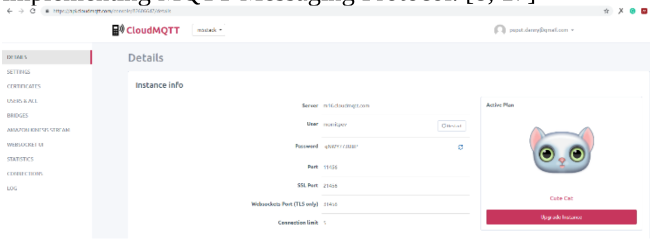
Fig.14. Comforting data Server, Username, Password and port from MQTT Broker and Blockly

The important part is equalizing data from MQTT Broker and Blockly programming, the equalized parts are Server, Username, Password and port from MQTT Broker and Blockly. M5Stack Pseudocode of Python client shows the Python form of Blockly Programming on this research.

```
I. Import the libraries
from m5stack import *
from m5ui import *
from m5mqtt import M5mqtt
import units

II. Initialize a Port
clear_bg(0x222222)
pir0 = units.PIR(units.PORTB)

III. Initialize a information MQTT Broker
m5mqtt = M5mqtt('helloM5', 'm16.cloudmqtt.com',
11456, 'momitpev', '6ErEW8LPQA_3', 300)

IV. Initialize a buttons
btnA = M5Button(name="ButtonA", text="ButtonA",
visibility=False)
btnB = M5Button(name="ButtonB", text="ButtonB",
visibility=False)
btnC = M5Button(name="ButtonC", text="ButtonC",
visibility=False)

V. Initialize a Text
label1 = M5TextBox(132, 97, "Text",
lcd.FONT_Default,0xFFFFFF, rotate=0)

VI.Starting MQTT Broker
m5mqtt.start()
label1.setText('Helo Puput')
while True:
VII. Initialize a Output

m5mqtt.publish(str('pir'),str((pir0.read())))

VIII. Initialize a timer
  wait(1)
  wait(0.001)
---Pseudocode of Python client M5Stack---
```

B. MQTT Analysis

Before carrying out the analysis, record any information on the MQTT Broker. For example in Fig. 15, the broker used is CloudMQTT with mosquito version 1.5.8. Mosquito provides standard compliant servers and clients implementing MQTT Messaging Protocol. [8, 17]



Details

Instance info

Server m16.cloudmqtt.com

User momitpev

Password [illegible]

Port 11456

SSL Port 21456

Websockets Port (TLS only) 31456

Connection limit 5

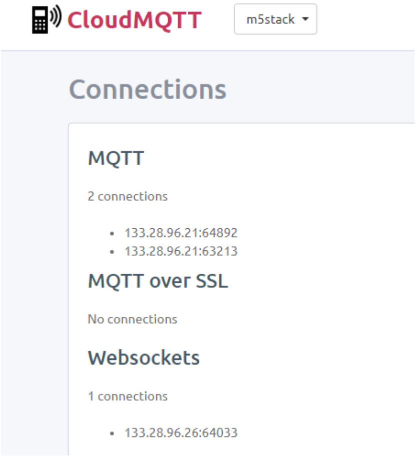
Fig.15. Information of CloudMQTT

TABLE II. INFO BROKER MQTT

No	MQTT Broker Parameter	Information
1	Server	Server Address of Cloud MQTT i.e, m16.cloudmqtt.com
2	User	User at CloudMQTT e.g.momixx
3	Password	Password at CloudMQTT Broker e.g, qNWYXxxxx
4	Port	Port use at CloudMQTT Broker, e.g, 11456
5	SSL Port	SSL Port use at CloudMQTT Broker, e.g, 21456
6	Websockets Port (TLS Only)	Port for Websockets e.g, 31456
7	Connection Limit	The Count of Limit Connection at CloudMQTT broker

C. Data IP Connection

IP Connection shows how many M5Stack devices are connected at CloudMQTT. In this research, there are 2 M5Stack Client devices. Fig.16 shows connections between MQTT Client and MQTT Broker (CloudMQTT).



Connections

MQTT

2 connections

- 133.28.96.21:64892
- 133.28.96.21:63213

MQTT over SSL

No connections

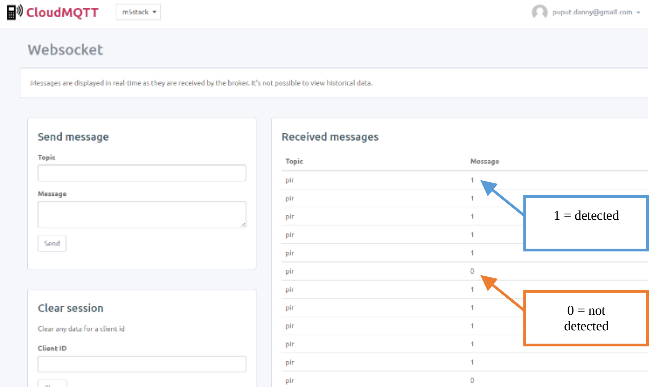
Websockets

1 connections

- 133.28.96.26:64033

Fig.16. Data IP Connection

Sensor Data from the Output or PIR Sensor can be seen in the real time on CloudMQTT Web socket broker [18] on Fig.17.



Websocket

Messages are displayed in real-time as they are received by the broker. It's not possible to view historical data.

Clear any data for a client id

Received messages

Topic	Message
pir	1
pir	1
pir	1
pir	1
pir	1
pir	0
pir	1
pir	1
pir	1
pir	1
pir	1
pir	0

Fig. 17. Data output PIR on Web Socket

The message in Fig. 17 is reading the initialize string 'PIR' from Blockly Programming i.e. `m5mqtt. publish (str ('pear'), str ((pir0.read ()))))` which means the value of PIR is 1 is detected by movement and PIR is 0 is not detected by movement, for blockly programming it can be seen in Fig. 18. in Fig.18 there are 3 types of Blockly Programming, i.e, Loop, (publish topic and message) and timer, and the loop functions to cover the publish topic and timer function.

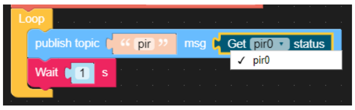


Fig. 18. Blockly Programming to recognize PIR

If stated in Python Programming :

```
m5mqtt.publish(str('pir'),str((pir0.state)))
wait(1)
```

Table III shows the Publish and Subscribe processes in MQTT Protocols, there are 3 trials time 1 min, 5 min and 15 minutes, and can be seen the results of bytes of data sent and receiving.

TABLE III. DATA QoS M5STACK BOARD ON CLOUD MQTT BROKER LOAD

Times	connections /+	bytes/received/+	bytes/sent /+	messages/received/+	messages/sent/+	dropped /+	publish/received/+	publish/sent /+	sockets/ +
1min	0.02	481.37	2806.62	65.3	119.94	0	58.37	112.8	0.01
5min	0.1	478.42	2772.06	64.52	119.55	0.00	57.64	112.57	0.1
15min	0.29	373.79	2077.18	49.38	93.08	0	44.4	87.60	0.13

V. CONCLUSION

M5Stack is designed for engineering education, it espouses programming education using Cloud-based Graphical User Interface (GUI), i.e, flow.m5stack.com with two types of Programming languages i.e, Blockly and Python, furthermore, in this research the M5Stack ability was tested using PIR sensor as a sensor that can detect motion, and has been running well and has been tested. PIR Sensor uses the Grove I2C hub to be able to connect with the M5Stack board. This is an example of using sensors in m5Stack to implement one of the sensors found in the M5Stack GUI directory unit. by learning M5Stack devices, students directly learn Blockly Programming and apply it directly to the Internet using the MQTT Protocol provided by the M5Stack GUI.

ACKNOWLEDGMENT

Thanks to Professor Akio Kitagawa who helped the author in correcting this research paper and Professor Akita who helped provide all the device equipment, so that this research could be completed and tested properly.

REFERENCES

[1] Abhishek Viswanathan, Analysis of Power Consumption of the MQTT Protocol,Bachelor of Engineering – Electronics & Telecommunication, Mumbai University; MS - Telecommunication, School of Information Sciences, University of Pittsburgh, 2017

[2] Al-Fuqaha, A Guizani, M Tutorials, M Mohammadi, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Communications Surveys & Tutorials (Volume : 17,Issue: 4 , Fourthquarter 2015), doi: 10.1109/COMST.2015.2444095

[3] Anna Larmo, Antti Ratilainen and Juha Saarinen, Impact of CoAP and MQTT on NB-IoT System Performance, Sensors 2019, 19, 7; doi:10.3390/s19010007

[4] Meena Singh, M.A. Rajan, V.L. Shivraj, P. Balamuralidhar; Secure MQTT for Internet of Things (IoT); 2015 Fifth International Conference on Communication Systems and Network Technologies, DOI: 10.1109/CSNT.2015.16

[5] Puneet Kumar and Behnam Dezfouli, Implementation and Analysis of QUIC for MQTT, Internet of Things Research Lab, Department of Computer Engineering, Santa Clara University, USA

[6] Puput Dani Prasetyo Adi and Akio Kitagawa, "Performance Evaluation WPAN of RN-42 Bluetooth based (802.15.1) for Sending the Multi-Sensor LM35 Data Temperature and RaspBerry Pi 3 Model B for the Database and Internet Gateway" International Journal of Advanced Computer Science and Applications (IJACSA), 9(12), 2018. DOI: dx.doi/10.14569/IJACSA.2018.091285

[7] Puput Dani Prasetyo Adi, Rahman Arifuddin, DESIGN OF TSUNAMI DETECTOR BASED SORT MESSAGE SERVICE USING ARDUINO AND SIM900A TO GSM/GPRS MODULE,JEEMECS (Journal of Electrical Engineering, Mechatronic and Computer Science) https://doi.org/10.26905/jeemecs.v1i1.1982

[8] Roger A Light, Mosquitto: server and client implementation of the MQTT protocol, The Journal of Open Source Software, DOI: 10.21105/joss.00265

[9] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, Hongtaek Ju, Correlation analysis of MQTT loss and delay according to QoS level, The International Conference on Information Networking 2013 (ICOIN), DOI: 10.1109/ICOIN.2013.6496715

[10] Tetsuya Yokotani, Yuya Sasaki, Comparison with HTTP and MQTT on required network resources for IoT, 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), DOI: 10.1109/ICCEREC.2016.7814989

[11] 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, Volume 10 Issue 5, 2019. DOI: 10.14569/IJACSA.2019.0100518

[12] Yiming Xu, V. Mahendran School of Computer Science, University of Oklahoma, Norman, 73019, USA; Sridhar Radhakrishnan, Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery, 2016 8th International Conference on Communication Systems and Networks (COMSNETS), DOI: 10.1109/COMSNETS.2016.7439974

[13] 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, Volume 10 Issue 5, 2019. DOI: 10.14569/IJACSA.2019.0100504

[14] Yuvraj Upadhyay, Amol Borole, D. Dileepan; MQTT based secured home automation system, 2016 Symposium on Colossal Data Analysis and Networking (CDAN), DOI: 10.1109/CDAN.2016.7570945

[15] m5stack.com, MeRL Internet Connection, Kanazawa University on April 8, 2019

[16] hivemq, MQTT Security Fundamentals MeRL Internet Connection, Kanazawa University on mei 5, 2019

[17] steves internet guide, MQTT Protocol Messages Overview MeRL Internet Connection, Kanazawa University on April 9, 2019

[18] ibm.com/developerworks/jp/cloud/library/cl-mqtt-bluemix-iot-node-red-app/index.html, Kanazawa University on mei 5, 2019

[19] developers.google.com/blockly/ MeRL Internet Connection, Kanazawa University on 12 Juni, 2019

[20] Naik, N, Choice of effective messaging protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP, 2017 IEEE International Systems Engineering Symposium (ISSE), DOI: 10.1109/SysEng.2017.808825

[21] Muhammad Niswar, Amil Ahmad Ilham, Elyas Palantei, Rhiza S.Sadjad, Andani Ahmad, Ansar Suyuti, Indrabayu, Zaenab Muslimin,Tadjuddin Waris, Puput Dani Prasetyo Adi, "Performance evaluation of ZigBee-based wireless sensor network for monitoring patients' pulse status", 2013 International Conference on Information Technology and Electrical Engineering (ICITEE) DOI: doi/10.1109/ICITEED.2013.6676255