IoT Monitoring of a Master-Slave Robot System using MIT App Inventor

Siti Noramira Zulkarnain, Ruhizan Liza Ahmad Shauri*, Mohamad Haidil Saidin and Ahmad Zaidiel Afiqie Zamanhuri

***Abstract*—A previously developed 4-DOF master-slave robot system showed that the master device was able to control the motion of the slave robot but due to the wired connection between the systems, the user must be in the same room to monitor the robot's movement. In order to monitor the robot operation wirelessly from a distance or outside of the laboratory room, an IoT-based platform is desirable. With the IoT method selected, the interface between the application, hardware and software is essentially important. Each part requires specific programming codes to ensure transmission of data runs smoothly between the chosen interface platforms. MIT App Inventor was selected as the IoT application that uses Firebase Cloud for storing data acquired from the master and slave controllers. Bluetooth modules are used as the interface between the master and slave controllers, while NodeMCU ESP32 enables the Wi-Fi connectivity between all three i.e. the controllers, the cloud storage and the MIT App Inventor. As a result, selected robot data were observed to be viewable from the user's mobile devices using the MIT AI2 companion application. The verification test with the execution of the robot showed that the IoT platform has successfully displayed numerical and graphical data of the desired robot's joint angle and motor increments from the master and the slave controllers, respectively, based on several user's arm gestures measured by force sensors.**

***Index Terms*—Firebase cloud, human arm gesture, IoT application, master-slave robot, MIT App.**

I. INTRODUCTION

The development of master-slave arm robots has started from late 1970s and early 1980s, when researchers first began exploring the use of multiple robots for complex tasks [1]. Master-slave robot refers to a configuration of a master that controls a robot called "slave" through it command. The development of master-slave systems has advanced in many applications such as in industrial assembly, auxiliary medical treatment, extreme environments, and inspection [2] - [5].

The advantage of operating a slave robot using a master as a separate device or hardware can be extended to a monitoring system that can be flexible for user to use. Internet of Things (IoT) has recently gained popularity and extensively applied for many applications, including automated systems and robotics.

One example of these is an IoT system developed by [6] that monitors dental x-rays equipment performance for maintenance. When the performance of the machine starts deteriorating, user gets the data log of the machine parameters uploaded to an IoT system on the internet to determine whether a maintenance is needed or not. A PCB-based communication port module which consists of SPI interface, CAN, UART, USB and LAN was developed as the interface hardware for the x-rays system comprising of a remote controller as the master and the x-rays controller as the slave. A testbed that is prepared for deploying multiple types of robots in an IoT enabled environment was proposed by [7]. In order to make sure the ease and safe implementation of their robots in human society, RobotNEST allows any kind of robot operating system to be used by user, provides 3D LiDAR data for robot localization and applies 5G private networks for fast cloud access and wireless communication of sensor data and hardware actuation for the robot navigation.

Meanwhile, a master-slave IoT device has been developed by [8] for improving human walking performance based on actual measurement of cadence during brisk walking. The slave part is responsible to measure cadence parameters using IMUs and linear accelerometer equipped in the smart shoes that the user wears. The master part provides the pattern of walking performance as advised by therapist based on the biofeedback obtained from the slave IoT using Deep Neural Network model. Samsung smart watch was used as the IoT device to monitor and display the performance of user via cloud computing and internet communication.

Due to the high cost for setting up expensive fabrication laboratories (Fab Labs) for schools and universities, an IoT based Fabrication-as-a-Service (FaaS) platform was developed [9]. It enables the students to get access to computer controlled tools and equipment of the fabrication laboratories via internet. Application programming interfaces (APIs) that they created allows third-party applications to access the virtual Fab Labs as a Web service where the configuration of the equipment as well as the communication among Fab Labs from widespread users

This manuscript is submitted on 14 December 2023, revised on 22 January 2024, accepted on 24 January 2024 and published on 30 April 2024.

"This work was supported by Malaysia Ministry of Higher Education (MOHE) under Grant FRGS (FRGS/1/2019/TK04/UITM/02/9)".

Zulkarnain, S. N. was a student in the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Malaysia.

Shauri, R. L. A. is currently a senior lecturer with the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Malaysia.

Saidin, M. H. was a student in the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Malaysia.

Zamanhuri, A. Z. A. was a student in the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Malaysia.

*Corresponding author Email address: ruhizan@uitm.edu.my

and locations can be implemented. Each Fab Lab employs a two-tier architecture, interacts with the hub, deployed in the cloud, and communicates with each other via a network.

Self-adaptive software framework is another concern to ease IoT implementation for stable execution of the software under dynamic environment and software changes. An example of this was presented by [10] to support stable execution of the IoT devices with minimized human involvement under dynamic changes in user requirements, inserting or removing IoT devices, selection criteria of measurement and selection of environment parameters.

A massage robot called EMMA which includes a robot arm and a massage end-effector was developed as an IoT device that can record the robot task for the analysis of the patient treatment [11]. Control method adopted strict safety framework that limits the movement speed and excess of applied forces to avoid danger and pain to the patient when the robot carries out the task. Recorded data via IoT helps physiotherapists to conduct and optimize the next massage treatment routine. Similar work has been done by [12] where a Home-TeleBot system is developed occupying an IoT architecture to support healthcare treatment at home by a dual-arm robot called YuMi. The IoT framework layer is responsible to collect human motion made by a teleoperator who is wearing a wearable inertial motion unit device and transmit it to the robot via wireless 5G network. The IoT teleoperated robot imitates the human motions made by the teleoperator who can operate the device from a hospital or from any distance.

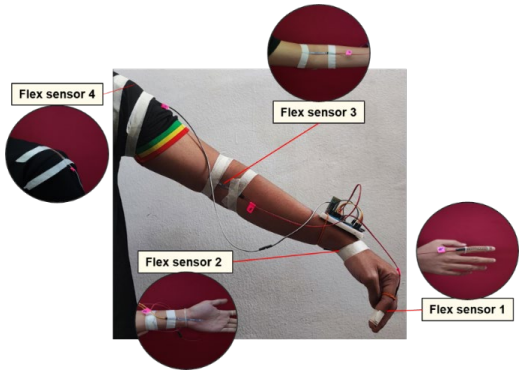
At the beginning of the work, a teleoperation robot has been developed consisting of a master device and a 4-DOF slave robot. The master includes a microcontroller and several flex force sensors placed on a user arm's joints, whereas the slave robot receives commands from the master's movement to move its links accordingly. The master is required to provide the user gestures information i.e. the desired angle of the joints and the direction of joints' rotation to the slave controller for driving the robot motion. However, the execution of the master-slave operation can only be done in the same laboratory due to the wired configuration between the hardware and controllers and therefore could not be monitored or controlled in other location. From the literature review, most of the studies applied IoT for non-robotic applications such as monitoring in the fields of agriculture, rehabilitation, education and machine performance. Due to safety and complexity of the interface system that needs to be established between the IoT device and hardware platforms, detailed publications that applied IoT as the modern technology approach for robot systems is quite difficult to find. Therefore, a development of IoT application method using MIT App Inventor is presented with the purpose of monitoring robot information from both the master and the slave robot remotely or from a distance. The second section of this paper presents the hardware specifications of the developed master-slave robot, the IoT system architecture and software specifications, as well as the interface platforms that have been selected. Next section explains the modifications made to the controller and software to enable communication between the interfaces. Consequently, the validation results of the proposed method are presented and discussed for the offline and online tests, followed by the conclusion in last section.

II. METHODOLOGY

Initially, the master as the arm's gesture measurement system and the slave robot were designed and developed as the hardware part of the robot system. Then, the information from the master and slave controllers are sent to the Firebase Cloud to be accessible by MIT App Inventor installed on user's Android devices to display the robot's information wirelessly.

A. Phase I: Development of the Master Slave Robot

The developed master-slave robot shown in Fig. 1 consists of a master which provides user arm's gesture information and a 4-DOF slave robot that moves according to the angle information from the master. The master is called wearable arm gesture measurement system (WAGMS) which consists of a controller and flex sensors positioned at several locations of user's arm to measure the bending angles of user's joints [13]. The sensor measurements mapped to the angles of user's arms are transmitted from the master's controller to the slave's controller via HC-05 Bluetooth Module for actuating the servo motors of the robot joints. The slave robot is a modified design from a commercial arm robot with fully actuated links consisting of a shoulder, elbow, wrist and a gripper type end effector [14].



a) The assembled WAGMS.

b) The modified design of 4-DOF slave robot.

Fig. 1. The developed WAGMS and 4-DOF robot [13], [14].

They are arranged in sequential order where the first three joints rotate around the same axis x, followed by the fourth joint that provides the open-and-close motion of the gripper. The specifications of the slave and the master are tabulated in Table I and Table II, respectively. Fig. 2 shows the flowchart for the development of the 4-DOF robot.

TABLE I. COMPONENT LIST OF ROBOT ARM [14]

No	Component	Specification
1	Actuator	• MG996R Servo Motor: Max Stall Torque at 11 kg/cm (6V) • LDX-218 Full Metal Gear Digital Servo: Max Stall Torque17 kg/cm (7.4V)
2	Controller	Arduino Uno R3
3	Feedback	Adafruit PCA9685 16-Channel Servo Driver
4	Input	Gesture movement based from the flex sensor
5	DOF	4-DOF
6	Programming Software	Arduino IDE
7	Interface	HC-05 Bluetooth Module: Communicate with the sensor system

TABLE II. SPECIFICATIONS OF WAGMS [13]

Component	Description
Controller	Arduino Nano
Sensor	Flex sensor to provide input signal
Interface	Bluetooth Module (HC-05) to transmit signal

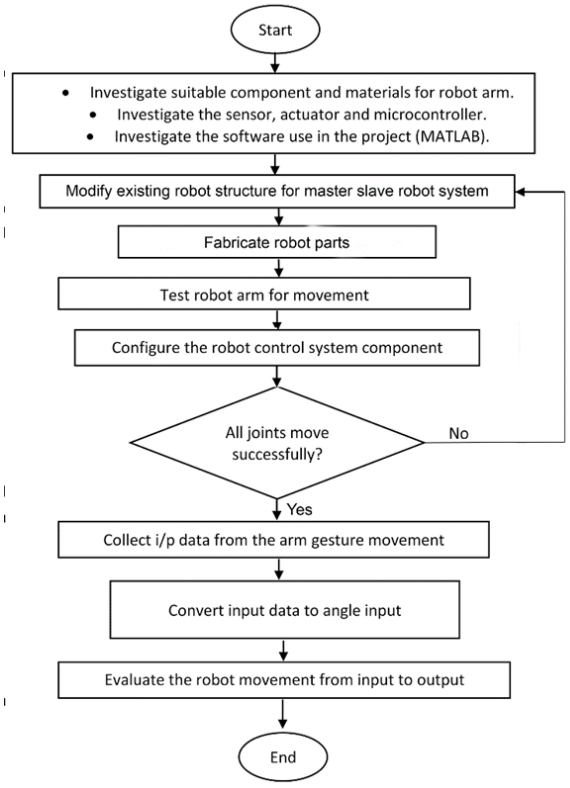


Fig. 2. Phase I involving the development of the master-slave robot [14].

B. Phase II: Setting up the IoT interface platform and programming the codes

As shown in Fig. 3, the steps involved in phase II comprises of selection of IoT application, replacement of previous microcontrollers, writing the codes for the new ESP32 microcontroller, establishing the interface between the hardware and software platforms, followed by creating the programming codes for each of the IoT platforms, and finally the validation test in offline and online modes with the robot.

1) Replacement of Microcontroller and Establishing the IoT Interface Platform

Firstly, two platforms for developing the applications have been explored i.e. Flutter and Android Studio. However, these platforms can be quite tough for designing mobile applications without prior programming skills and they take longer time to develop. Following that, the MIT App Inventor has been explored where it is found to be much easier to be implemented. MIT App Inventor is a free, browser-based and blocks-based programming platform that allows creation of personalized mobile apps and access from any device [15], [16].

Next, the replacement of previous WAGMS master and slave robot's Arduino Uno and Nano controllers with NodeMCU ESP32 microcontroller was required in order to have Wi-Fi connectivity for the data transfer from both parts to the cloud database. Programming codes in NodeMCU ESP32 of the master carries the functions for receiving and processing data from the flex sensor before sending it via Bluetooth interface to the slave robot's controller. The bending actions measured by flex sensors are converted to digitized values and represented in terms of bending angle which is used by the slave controller to calculate the motor increments before sending pwm signals to actuate the motor. The specifications and detailed information of the NodeMCU ESP32 are tabulated in Table III.

For data transfer and storage, the system uses Firebase Cloud technology. The NodeMCU ESP32 board's built-in Wi-Fi capabilities enables real-time data transfer from the master and slave controllers to the Firebase Cloud as a secure centralized data storage site. The system architecture of the IoT based master-slave robot system in Fig. 4 shows Bluetooth module is used for communication between the master and slave controller boards whereas Wi-Fi communication is used by the NodeMCU ESP32 board to send data to Firebase Cloud, Firebase Cloud for data storage and retrieval, and MIT App Inventor for visualizing the selected data as IoT application. After installing MIT AI2 Companion application on user's mobile devices, these data can be accessed by user in real time during the operation of the master-slave robot.

The Bluetooth module is linked to the NODEMCU ESP32 microcontroller board via UART, and the NodeMCU ESP32 board uses pins labeled TX (transmit) and RX (receive) for computer communication. Correct connection between the boards to the communication pins is important to ensure successful data transfer and uploads between the hardware and software platforms.

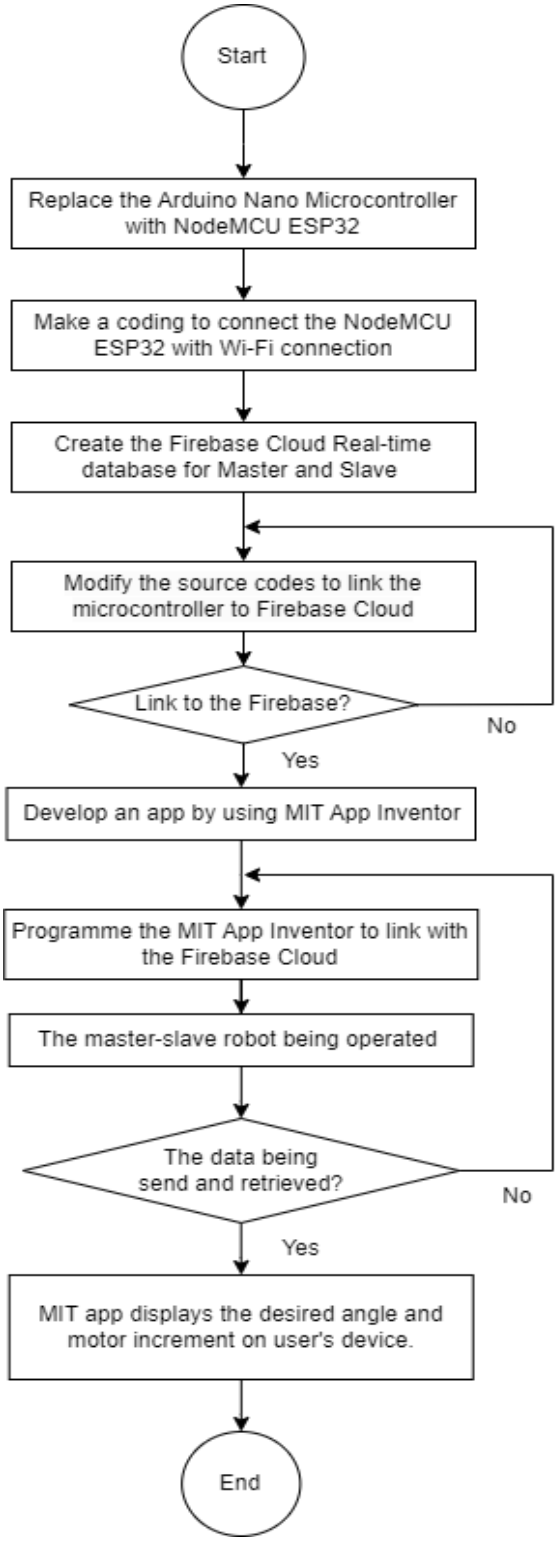


Fig. 3. Phase II steps consisting of the programming of different boards, database and MIT App Inventor.

TABLE III. ESP32 SPECIFICATIONS

Features	ESP32
MCU	Xtensa Dual-Core 32-bot LX6 600 DMIPS
Frequency	80-240 MHz
Wi-Fi	802.11 b/g/n
Bluetooth	BL v4.2, BLE
SRAM	512kB
Flash	SPI Flash, up to 16 MB
GPIO	36
HW/SW PWM	1/16 channels
SPI/12C/I2S/UART	4/2/2/2
ADC	12 bits
CAN	1
Ethernet Mac Interface	1

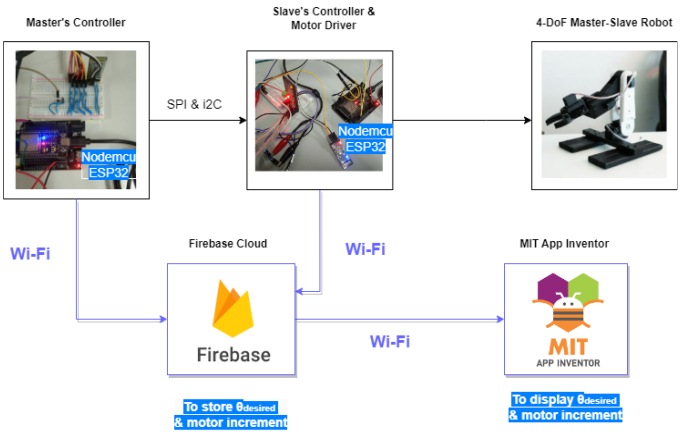


Fig. 4. Master-Slave robot monitoring system architecture.

2) Designing MIT App Inventor and Firebase Database Console

The Realtime Database page of the Firebase Console shown in Fig. 5 is important for managing the uploaded data in Firebase Cloud system effectively. Here, four position angles of the arm as instructed by WAGMS as reference to the slave robot have been selected to be stored. Besides, the motor increments calculated by the slave controller was also selected to be stored in the database. Meanwhile, MIT App Inventor programming platform consists of three main components: the "Designer" (Fig. 6) for creating the user interface, the "Blocks Editor" (Fig. 7) for designing the application, and the "AI Companion" for displaying the application on user's mobile devices.

3) Validation of MIT Inventor App with Robot Values

The proposed method is validated by comparing the robot information displayed on the MIT App Inventor with the movement of the master-slave robot at several angles. Firstly, a

set of desired position angles $\theta_{desired}$ are given to the slave microcontroller to calculate the motor increments and the comparison is made from the values obtained from the serial monitor of Arduino IDE window with the Firebase window to prove the success of data transmission between the microcontroller and the cloud storage platform. Secondly, $\theta_{desired}$ (set in integer data type) for several angles at 0°, 45°, and 90° measured by WAGMS in an experiment were used by the slave microcontroller to calculate the motor increments for each joint of the shoulder, elbow, wrist and gripper of the slave robot, where these $\theta_{desired}$ and motor increments are to be displayed on the MIT App Inventor's emulator.

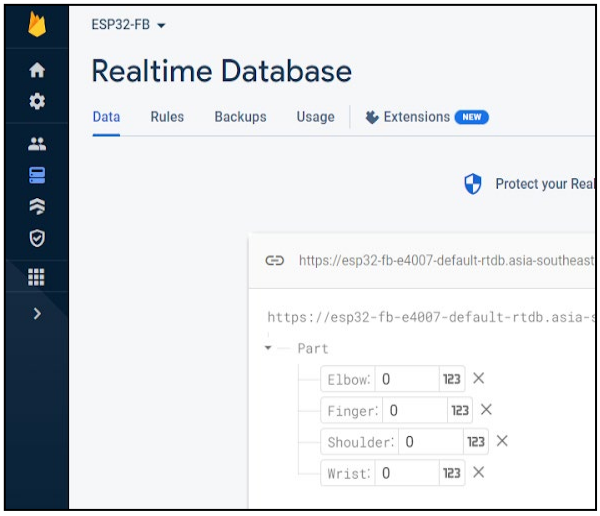


Fig. 5. The real-time database page.

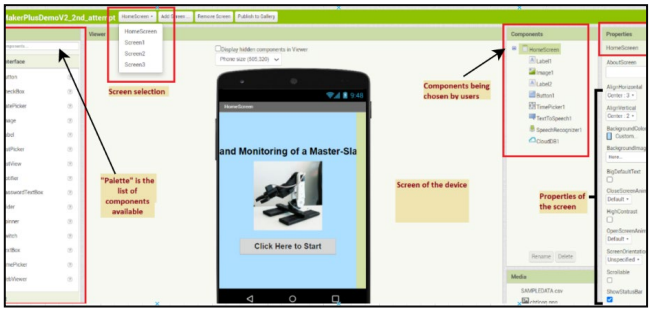


Fig. 6. Design editor page for specifying the display layout.

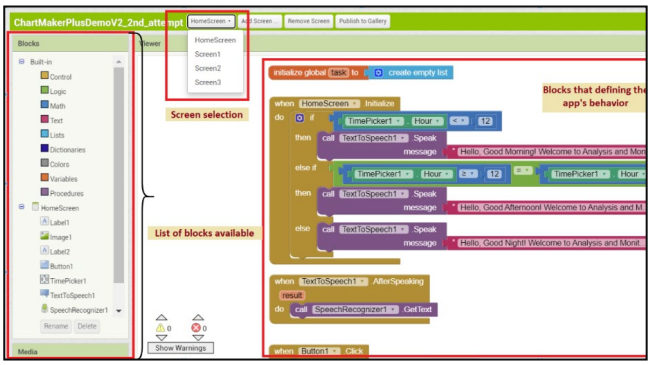


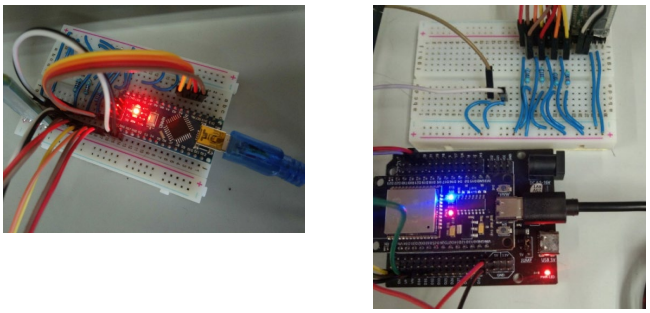
Fig. 7. Block editor for programming MIT App Inventor.

III. Results and Discussion

A. Modified Hardware and IoT Software

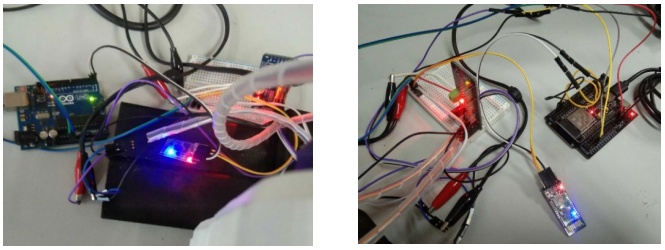
The Arduino Nano (Master) and Uno (Slave) were replaced with NodeMCU ESP32 microcontrollers that can provide the wireless communication feature between the boards with the Firebase Cloud storage. Fig. 8 and Fig. 9 show the replacement of the Arduino boards with NodeMCU ESP32.

Then, after connecting MIT App Inventor (on PC) and MIT AI2 Companion (on mobile device) to the same Wi-Fi network, MIT AI2 Companion application will display two options of linking both devices i.e. via coding or QR code scanning for the user to choose (Fig. 10a). This will then bring user to the home screen of the application that has been designed with MIT App Inventor as shown in Fig. 10b.



a) Before (Arduino Nano). b) After (NodeMCU ESP32).

Fig. 8. WAGMS master part.



a) Before (Arduino Uno R3). b) After (NodeMCU ESP32).

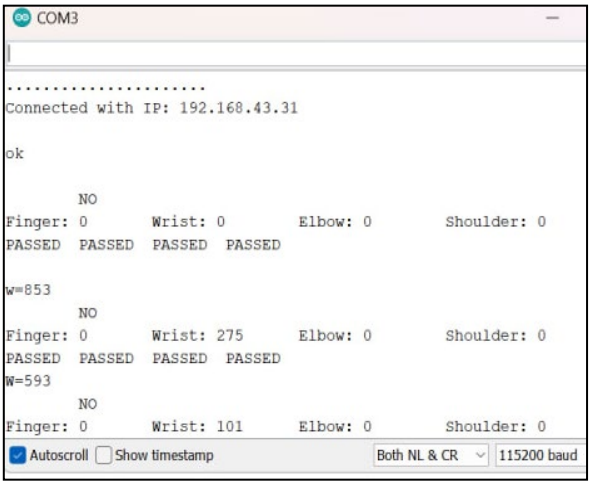
Fig. 9. The slave part.

a) Linking MIT AI2 Companion with the designed application. b) Application's home screen on user's mobile device.

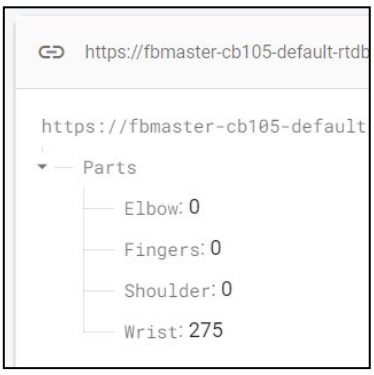
Fig. 10. Home screen view on user's mobile device.

B. Validation of MIT Inventor App with Robot Values

The joint reference values of the WAGMS appeared on the serial monitor of Arduino IDE were compared to the Firebase's real-time database to confirm the data transferred from the hardware controllers to the cloud. As shown in Fig. 11, the data transfer run smoothly to confirm the feasibility of the interface and communication between the wireless platform. The changes of $\theta_{desired}$ value versus time on a line graph shown in Fig. 12 allows user to monitor the reference angle position from WAGMS that are sent to the slave controller.



a) Serial monitor Arduino IDE.



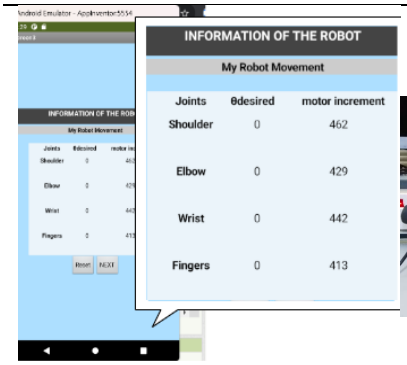
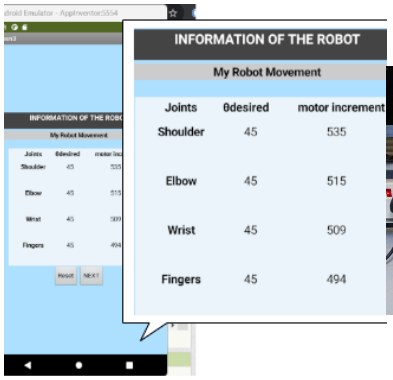
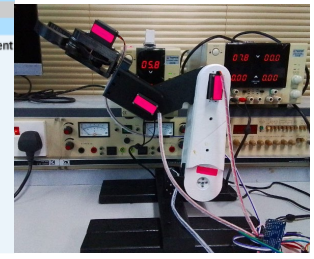
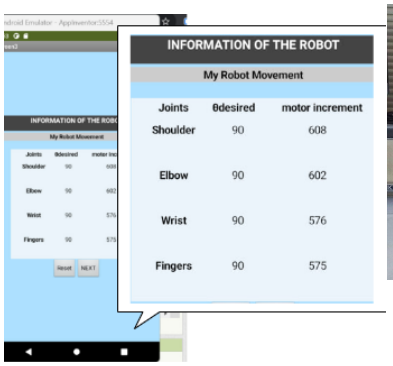
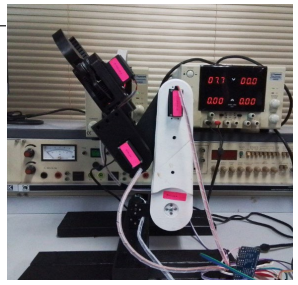
b) Firebase Realtime Database

Fig. 11. Comparing the transmitted values.

Fig. 12. Line graph of the desired angle from the master controller.

Table IV shows the comparison between the $\theta_{desired}$ values and motor increments displayed on MIT Inventor App's emulator (on the left column) and the robot's movement for the respective angles of 0°, 45°, and 90° measured by WAGMS for the arm gestures made by the user (on the right column). The results showed that the values from WAGMS can be displayed on MIT app in real time and the calculated motor increments moved the robot joints' accordingly. Hence, this proves the applicability of the designed application to display the robot information via IoT method during the master-slave robot operation.

TABLE IV. COMPARISON RESULTS BETWEEN MIT APP INVENTOR AND MASTER-SLAVE ROBOT

MIT display	Robot Movement
	
	
	

Position tracking accuracy for the robot is not discussed in this paper as the evaluation of robot control performance is

beyond the scope of this work which is focusing on the IoT monitoring for the operation of the robot via network communication. Therefore, evaluation for different angle movements of robot is not necessary because instructions by the user's gesture will provide robot information to be displayed on MIT application as they are calculated not on the IoT platform but by both robot controllers.

IV. CONCLUSION

The work has shown the implementation of IoT using MIT Inventor App and cloud based storage for a master-slave robot system. Robot information such as the desired joint position and the instructed motor increments have been successfully displayed on the user's mobile devices which proves the feasibility of the IoT monitoring of the robot via wireless communication interface between the master and slave microcontrollers, cloud storage platform and the user's mobile device. This approach can be further improved to embed bilateral communication on IoT system in a less interrupted network environment for robot monitoring information and instructing robot from user devices wirelessly and remotely. Moreover, informative graphs and analysis of the robot performance can also be added to further elevate the advantage of using IoT approach for robot control in the future study.

ACKNOWLEDGMENT

This research is fully supported by Ministry of Higher Education (MOHE) (FRGS/1/2019/TK04/UITM/02/9) grant. The authors acknowledged Ministry of Higher Education (MOHE) for the approved fund and to Universiti Teknologi MARA for providing the laboratory space and equipment.

REFERENCES

[1] M. E. Moran, "Evolution of robotic arms," *Journal of Robotic Surgery*, vol. 1, no. 2, pp. 103-111, May 2007, doi: https://doi.org/10.1007/s11701-006-0002-x.

[2] F. Li, Q. Jiang, W. Quan, S. Cai, R. Song, and Y. Li, "Manipulation skill acquisition for robotic assembly based on multi-modal information description," *IEEE Access*, vol. 8, pp. 6282–6294, 2020, doi: https://doi.org/10.1109/ACCESS.2019.2934174.

[3] Y. Zhang et al., "Design and experimental study of a novel 7-DOF manipulator for transrectal ultrasound probe," *Science Progress*, vol. 103, no. 4, pp. 1–24, 2020, doi: https://doi.org/10.1177/0036850420970366.

[4] W. Gao, W. Wang, H. Zhu, S. Zhao, G. Huang, and Z. Du, "Irradiation test and hardness design for mobile rescue robot in nuclear environment*," Industrial Robot: The International Journal of Robotics Research and Application*, vol. 46, no. 6, pp. 851–862, Oct. 2019, doi: https://doi.org/10.1108/ir-01-2019-0010.

[5] D. Seward and M. Bakari, "The use of robotics and automation in nuclear decommissioning," 2005. Accessed: Jan. 21, 2023. [Online]. Available: https://www.researchgate.net/publication/228663007_The_Use_of_Robotics_and_Automation_in_Nuclear_Decommissioning/citations.

[6] W. -Y. Lee, C. -L. Shih, T. -H. Chen and Y. -H. Chen, "Scalable Master-slave isomorphic module for IoT service system," in *Proc. 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media)*, Bali, Indonesia, 2019, pp. 224-229, doi: 10.1109/Ubi-Media.2019.00051.

[7] S. Aoki, T. Yonezawa and N. Kawaguchi, "RobotNEST: Toward a viable testbed for IoT-enabled environments and connected and autonomous robots," *IEEE Sensors Letters*, vol. 6, no. 2, pp. 1-4, Feb. 2022, Art no. 6000304, doi: 10.1109/LSENS.2021.3139624.

[8] S. A. Senanayake, N. H. Kadir, M. S. A. Suhaimi and M. Sasaki, "Master-slave IoT for active healthy life style," in *Proc. 2019 12th International Conference on Human System Interaction (HSI)*, Richmond, VA, USA, 2019, pp. 151-157, doi: 10.1109/HSI47298.2019.8942640.

[9] G. Cornetta, A. Touhafi, M. A. Togou and G. -M. Muntean, "Fabrication-as-a-Service: A web-based solution for stem education using Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1519-1530, Feb. 2020, doi: 10.1109/JIoT.2019.2956401.

[10] E. Lee, Y. -D. Seo and Y. -G. Kim, "Self-adaptive framework with master–slave architecture for Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16472-16493, 1 Sept.1, 2022, doi: 10.1109/JIoT.2022.3150598.

[11] W. Si, G. Srivastava, Y. Zhang and L. Jiang, "Green Internet of Things application of a medical massage robot with system interruption," IEEE Access, vol. 7, pp. 127066-127077, 2019, doi: 10.1109/ACCESS.2019.2939502.

[12] H. Zhou, G. Yang, H. Lv, X. Huang, H. Yang and Z. Pang, "IoT-enabled dual-arm motion capture and mapping for telerobotics in home care," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 6, pp. 1541-1549, June 2020, doi: 10.1109/JBHI.2019.2953885.

[13] M. H. Saidin, "Development of wearable arm gesture measurement system," B.S. thesis, Universiti Teknologi MARA, Malaysia, 2023.

[14] A. Z. A. Zamanhuri, "Master-slave controlled robotic arm: hardware development and control," B.S. thesis, Universiti Teknologi MARA, Malaysia, 2023.

[15] S. B. Mir and G. F. Llueca, "Introduction to programming using mobile phones and MIT App Inventor," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 15, no. 3, pp.192–201, Aug. 2020, doi: https://doi.org/10.1109/rita.2020.3008110.

[16] System Requirements. appinventor.mit.edu. [Online]. Available: https://appinventor.mit.edu/explore/content/system-requirements.html.