Design of Robotic Arm Controller based on Internet of Things (IoT)

Mohamad Khairi Ishak[1], Muhammad Izzat Roslan[1] and Khairol Anuar Ishak[2]
[1]*School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300, Penang, Malaysia.*
[2]*School of Business Management, UUM College of Business, Universiti Utara Malaysia, Sintok,06010, Kedah, Malaysia.*
khairiishak@usm.my

***Abstract*—This paper presents the process of developing a controller for a robotic arm that is built through the Internet of Things (IoT). The direction of the robotic arm can be monitored and controlled using internet facilities. The Raspberry Pi board is utilized in this project for the robotic arm controller as well as the web server system. The robotic arm comprises four servo motors and each of the servo motors is assigned with a single pulse width modulation (PWM) output that can be individually controlled. The controller system is implemented on Raspberry Pi board using Python 2.7 programming language. Node-Red is used as a web server in this project to communicate with the web browser through TCP/HTTP. Hence, this allows the user to access the web browser using computer or smartphones. In addition, it enables the monitoring and controlling of the robotic arm direction as well as performing pick and place task similar to the manufacturing industry. The results of this study are verified through practical test implementation.**

***Index Terms*—Raspberry Pi; Robotic Arm; Web Server; Internet of Things.**

I. Introduction

The emergence of technology has managed to witness the evolvement of a new era known as the Internet of Things (IoT). The Internet of things (IoT) further describes the internetworking of physical devices, vehicles, buildings, and other items which include electronics, software, sensors, actuators, and network connectivity that allow these objects to collect and exchange data [1]. In addition, it also allows objects to be sensed or remotely controlled across an existing network infrastructure, which will result in increased performance, accuracy, and economic benefit [2]. Presently, robot is widely utilized in most of the industries due to its added advantages that can compensate the human inability to operate in hazardous area, reliability, high precision, and accuracy. Hence, it is safe to say that robot is used as a human replacement.

A robot is described as a machine designed to perform a particular task based on the programming created by the user. More importantly, it allows multitasking to be performed at a time. The most common robot used for industrial purposes is the robotic arm. In general, the robotic arm is a mechanical arm that can be programmable similar to the functions of a human arm. The main task of the robotic arm is to move an end effector from place to place such as pick up as well as carry different objects [3].

The current revolution of IoT coupled with the growing usage of robots in daily activities have turned the Internet of Robotics Things (IoRT) applications into a tangible reality in the nearest future. The advantage of IoRT is that it allows robotic systems to connect, share, and pass the distributed computation resources, business activities, context information, and environmental data with each other [4]. Interestingly, users can delegate the task to it remotely through networks provided that the robot is connected to the internet; hence, this implies that user does not have to be present on the site because the job can be entirely performed by the robot. As a result, this opens a new horizon in the domain of connected robotics that is expected to lead to fascinating futuristic developments. Apart from that, it will also examine the ways in which IoT technologies and robotic “devices” intersect to produce advanced robotic capabilities.

The IoT technologies have been observed to expand over the recent years; however, most of the established work only aimed at utilizing the IoT technologies for extremely resource-constrained nodes such as sensor network nodes that can only send collected data to base stations [5]. Hence, it is safe to say that the application of IoT technologies into the robotics field has received little attention. In relation to this, there are high demands from various industries to use IoT technology in a robotic application based on the Industry 4.0 revolution [6].

Hence, an effectively embedded device is required to control and monitor robotic arm. In the context of this study, a Raspberry Pi is chosen to assist in the implementation of these systems considering that it has more advantages than others [7]. Raspberry Pi is used as a controller and web server for the robotic arm. Specifically, a web server will deliver a web page to the other computer in the same network, thus allowing the user to access the web pages in order to control the robot arm.

In this project, the implementation of web server is for the purpose of providing remote control and monitoring the robotic arm across existing network infrastructure. Raspberry Pi board is used as the embedded device in this project, while Analog Radio Control (RC) servomotor is employed as the actuator of the designed robot arm. More specifically, Raspberry Pi is designed to control the analog RC servomotor of angle as well as the direction of rotation. Not to mention, it is also used as a web server for remote controlling as well as robotic arm monitoring. It is worth to note that both the hardware and software are involved in this project.

II. System Design Development

In the design process, the system design development of robotic arm requires users to use devices that are connected to the Internet in order to access the web server through a Node-Red platform. The web server communicates with a Raspberry Pi that is accompanied by servomotor PWM drive for the purpose of controlling the robotic arm. The hardware and software function are combined to ensure that the system is reliable. The overall system design is shown in Figure 1.

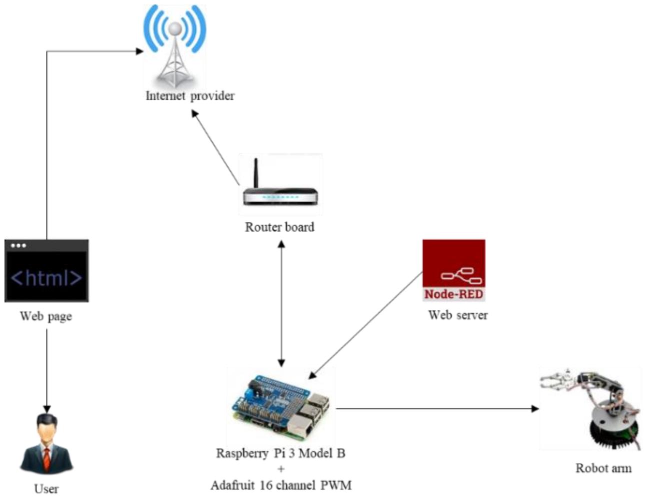

Figure 1: Block diagram of the overall system design

A. System structure

The project specification for the robotic arm controller based on through internet of things is shown in Table 1.

Table 1
Robotic arm controller specification

Module	Specification
Microcomputer	Raspberry Pi
Programming language	Python 2.7, Node-RED and JavaScript
Actuator	Servo motor
Robot Arm	3-DOF with gripper
Motor Drive	Adafruit 16-channel 12-bit PWM/Servo HAT

B. Robotic arm description

The robotic arm has three directions of motion (DOF) as well as a grip movement (3+1) with four servo motors as in Figure 2. Each RC servomotor represents the joints that enable the movement of the robotic arm. In this case, a frame that is made from a lightweight and durable aluminium is chosen for the purpose of reducing the overall weight of robot arm and further lessen the load of the servo motor.

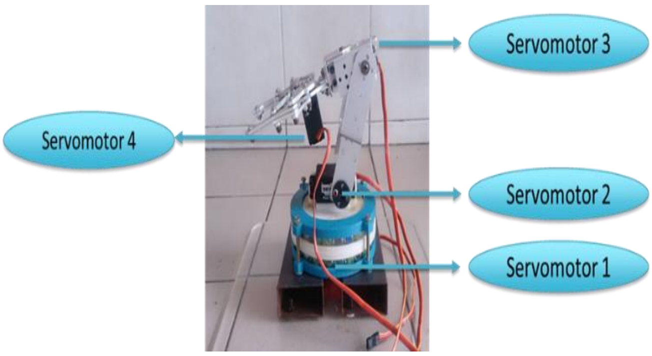

Figure 2: Location of each RC servo motor on the robotic arm

All the RC servo motors are equipped with a low cost, RC servos from Cytron Technologies model HD-1501MG that comes with full metal gear. It produces a 180-degree rotation with torque rating 17.0 Kg.cm and turning rate at 0.14s/60° at 6.0V [8]. The process of controlling the RC servo motor is by sending a pulse of variable width. The angle is determined based on the duration of a pulse that is applied to the signal wire. The generated PWM pulses must be in the range of 0.5ms to 2.5ms on every 20ms for every period. The length of the pulse will determine how far the motor turns. Meanwhile, the position of the pulse must be repeated to instruct the servo to stay in position

C. Development of Raspberry Pi controller

In the case of this study, a Raspberry Pi 3 Model B+ board is utilized to design a controller for the robotic arm. The pulse width modulation is designed based on the RC servomotor specification. As has been mentioned, there are four RC servo motors used in this project. Each servo motor is designed with specifying pulse width to perform a specific task. In addition, Python 2.7 programming language is used to program each of the servo motors for the purpose of controlling the angle of rotation as well as the pick and place operation. Figure 3 depicts the overall flowchart of Raspberry Pi controlled RC servo motor.

Servo motors are driven using the PWM outputs available on Raspberry Pi considering that it contains native hardware support for PWM. However, the generated PWM is not stable. Thus a bit of jitter with the servo is expected. The robotic arm needs to drive up to four servo motors in this project, but only one PWM channel is available for users at GPIO18. Therefore, Adafruit's 16-channel 12-bit PWM/Servo HAT is chosen as the servo motors drive.

Adafruit's 16-channel 12-bit PWM/Servo HAT is able to control up to 16 servos or PWM channels using the I2C interface of the Raspberry Pi. The servos are plug straight into it and the power is supplied to the logic circuits of the module from the 3.3V connection of the Raspberry Pi. The source of the power supply for the servo motors comes from an external 5V power adapter which is different from the Raspberry Pi, with the purpose of avoiding excess current draw that could cause the Raspberry Pi to act erratically, reset, or overheat. The HAT can also generate pulse width up to 1.6 KHz with 12-bit precision and capable of very precise motion control.

D. Development of web server

The primary role of web servers is to deliver web contents that are accessible to the clients through the internet. In this context, Node-Red is used to develop the web server where the robotic arm can be completely controlled on the web server. Node-Red is a flow-based programming tool, which refers to the way of describing an application`s behavior as a network "nodes" [10]. It is important to note that each node has a well-defined purpose. Node-RED provides a browser-based flow editor that makes it easy to wire together the flows using a wide range of nodes in the palette.

Web pages are designed in such a way to allow the user to control the angle of rotation of each servomotors using a slider. In this case, the user can use the slider to vary the angle parameters between 0 and 180 degrees as shown in Figure 4. Moreover, a graphical user interface will show the angle of individual joints in the web pages. Meanwhile, a button is also created to perform a pick and place operation. The corresponding Python program for the movement of each joint is executed on the web server when the button is clicked. Figure 5 shows the flowchart of the overall web server system. The users can conveniently access the web server using their devices such as computer, tablet, and smartphone provided that they are connected to the internet via Ethernet or Wi-Fi.

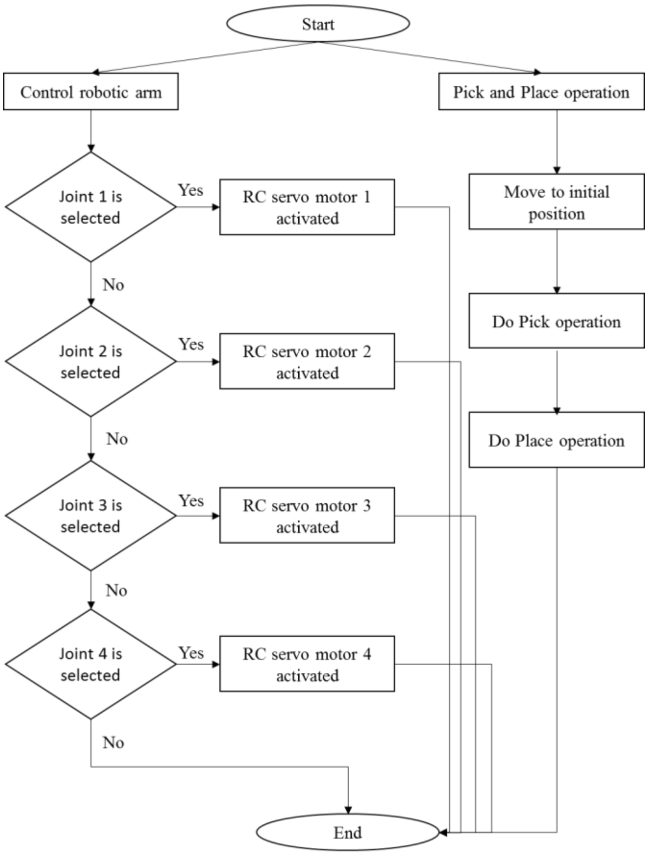

Figure 3: Flowchart of Raspberry Pi controlled RC servo motors

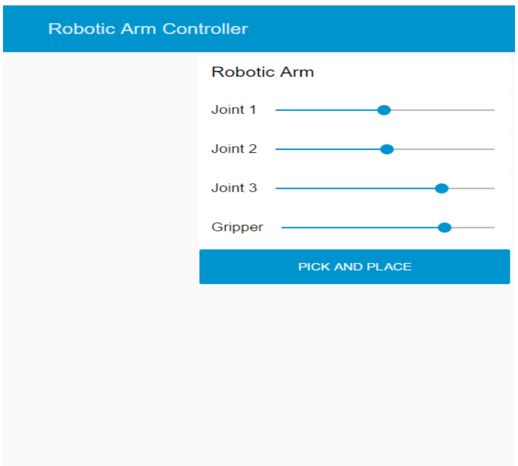

Figure 4: Slider for controlling the angle of rotation.

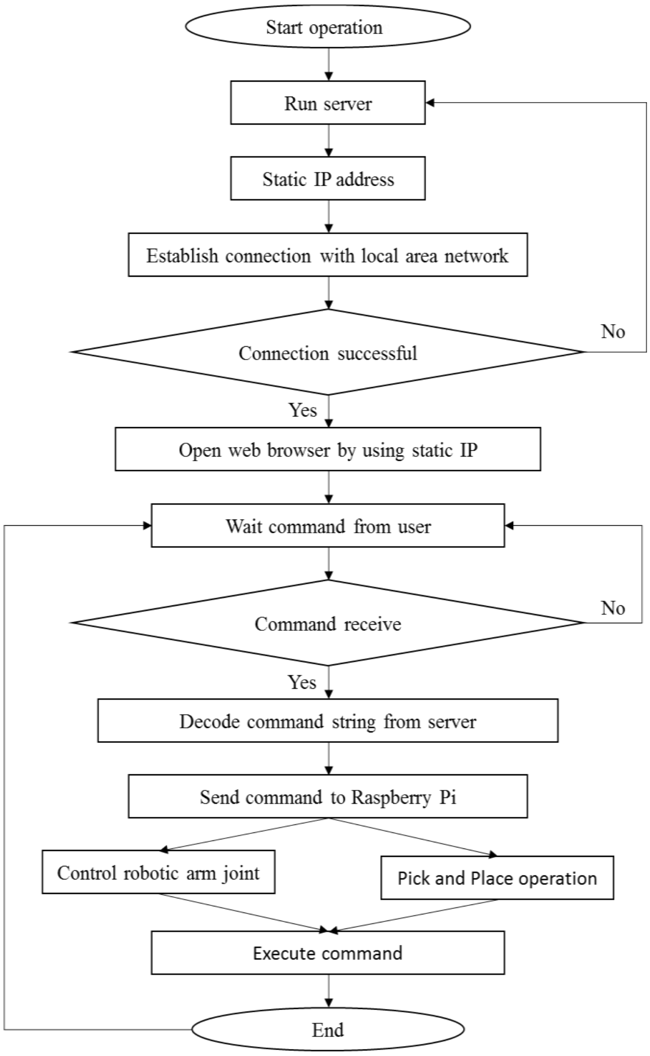

Figure 5: Flowchart of the overall web server system

III. Result and Discussion

The process of accessing the web server can be easily done by pointing to a local browser at http://localhost:1880/ui. Alternatively, the user can also use a web browser from another machine provided that the IP address or name of the web server instance is known for example HTTP://{Web-server-machine-IP-address}:1880/ui. The complete web page interface is shown in Figure 6.

Figure 6: Web page displayed in browser

A. Pick and place operation

The pick and place operation can be performed by the user by only clicking the “Pick and Place” button available on the web pages. The robotic arm will then perform the pick and place of an object as illustrated in Figure 7. As can be observed in Figure 7, the robotic arm has successfully conducted the instruction precisely to move within their desired angular displacement degree.

B. Time delay through a network

The configured network has been examined by sending the data packets from one host to another destination host that is placed at another location, while the receiving destination host will reply to the sent data packets. The purpose of this test is to verify the reachability of a host on an Internet Protocol (IP) network and calculate the time delay between the two packets. This test can be carried out using the Wireshark software to analyze the network performance [11]. The results of the analysis are displayed in Table 2.

Table 2
Time delay through wire and wireless network

Connection	Max (*ms*)	Min (*ms*)	Average (*ms*)
Wired	42.046	0.001	6.082
Wireless	219.902	0.001	12.679

.

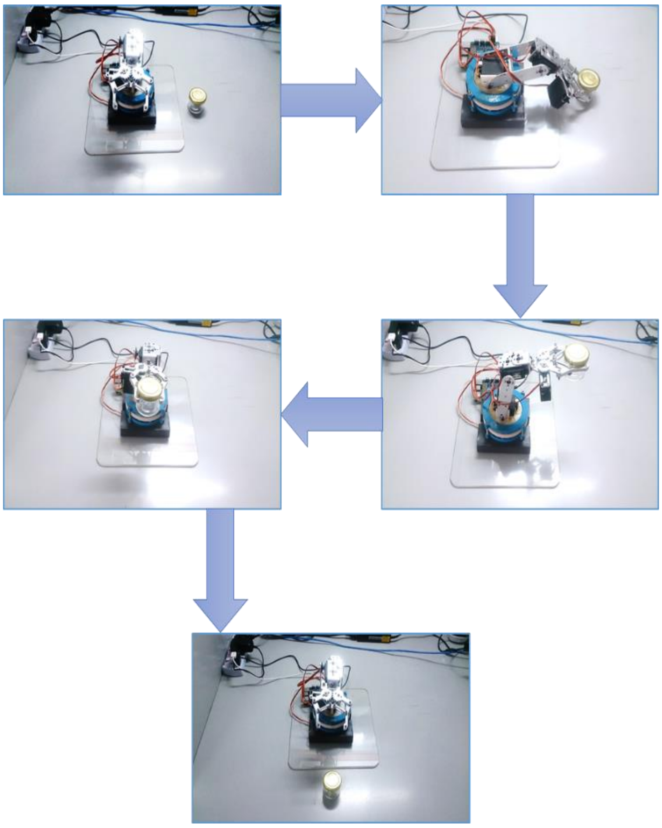

Figure 7: Pick and place operation

As shown in Table 2, wired connection can better reduce time delay compared to a wireless connection with the average time delay of 6.082*ms* and 12.679*ms*, respectively. Wired connection is shown to be more stable compared to a wireless connection considering that its maximum time delay is only 42.046*ms*. Meanwhile, the maximum time delay for wireless connection is 219.902*ms*. Hence, it can be concluded that a wired connection is faster and more stable compared to wireless connection. Overall, it is safe to say that the duration of time delay is highly dependent on the network type, speed, and distance from the network.

IV. CONCLUSION

This paper has successfully developed a monitoring and controlling system of the robotic arm via the Internet of Things (IoT) using Raspberry Pi board which is performed on a web browser. The robotic arm can be smartly controlled using personal computer or smartphone connected to the same network. This implementation has allowed the robotic arm to be controlled by the user from any part of the world through the Internet of Things. Additionally, it is worth to note that any system that adopts internet facilities will experience small time delay depending on the type, network speed, and distance from the network. The practical test results have revealed that wired connection is able to minimize the time delay and make it more stable in comparison to a wireless connection.

REFERENCES

[1] Eric Brown, “Who Needs the Internet of Things? | Linux.com | The source for Linux information,” *Linux*, 2016. [Online]. Available: https://www.linux.com/news/who-needs-internet-things. [Accessed: 08-Mar-2017].

[2] Harvard Business Review, “Internet of Things: Science Fiction or Business Fact? ” *Harv. Bus. Rev.*, vol. Analytics, p. 8, 2014.

[3] M. E. Moran, “Evolution of robotic arms,” *J. Robot. Surg.*, vol. 1, no. 2, pp. 103–111, 2007.

[4] P. P. Ray, “Internet of Robotic Things: Concept, Technologies, and Challenges,” *IEEE Access*, vol. 4, pp. 1–1, 2017.

[5] T. Yashiro, S. Kobayashi, N. Koshizuka, and K. Sakamura, “An Internet of Things (IoT) architecture for embedded appliances,” *2013 IEEE Reg. 10 Humanit. Technol. Conf. R10-HTC 2013*, pp. 314–319, 2013.

[6] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, “Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data-based feedback and coordination,” *Comput. Networks*, vol. 101, pp. 158–168, 2015.

[7] M. Maksimović, V. Vujović, N. Davidović, V. Milošević, and B. Perišić, “Raspberry Pi as Internet of Things hardware : Performances and Constraints,” *Des. Issues*, vol. 3, no. JUNE, p. 8, 2014.

[8] M. G. Rc and S. Motor, “RC Servo Motor User's Manual,” May 2014, pp. 1–14.

[9] P. M. Edt, “Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi,” 2016.

[10] M. Blackstock, E. Ave, and R. Lea, “FRED : A Hosted Data Flow Platform for the IoT,” pp. 0–4, 2016.

[11] A. Dabir and A. Matrawy, “Bottleneck Analysis of Traffic Monitoring using Wireshark,” *2007 Innov. Inf. Technol.*, pp. 158–162, 2007.