Development of Robotic Arm Controller Using Arduino Microcontroller and Mobile Device Application

Nanang Ali Sutisna*, Muhammad Irfan Satria[b]
Mechanical Engineering Department, Faculty of Engineering, President University
Jl. Ki Hajar Dewantara, Jababeka, Cikarang Bekasi
*E-mail: nanang.ali@president.ac.id

Abstract

In today's world, automation is widely used in many aspects to help humans do their job and increase the efficiency of the job. The robot is one of the kinds of an automation system that mostly used to help human to completed their job. Robots have many roles in human life, and robots are used to help humans do various kinds of jobs more easily. Robots are widely used because they can make it easier and done the work effectively and efficiently. There are many implementations of the robot to help humans do their jobs, such as welding robot, agriculture robot, drone robot, and many more. A robot arm is one type of robot that we often encounter, and a robot arm has a function similar to a human arm. In its application, the robot can be controlled through several methods, some are directly using a cable connection, and some are using a wireless method. Also, to control a robot, a controller is needed, which will be the robot's brain. This project aims to develop a controller based on Arduino, one of the most widely used microcontroller boards for education, hobby, or professional, and develop a mobile device application to control the robot arm via Bluetooth connection. The methodology used in this research is starting from the identification and data collection of the program from several sources such as a journal or discussion forum, then continue with the systems schematic design of the hardware that used in this project and will continue by the program of the Arduino and smartphone app. The testing result shows the accuracy of actual motion is quite close to the desired position as programmed, and the servo test shows that the error value average is about 3,54% for the MG996R servo and 4,44% for the SG90 servo. Thus, the robot error value is about 1,38% on the x-axis and 12,14% on the y-axis in the first test. And for the second test is about 1,96% on the x-axis and 3,15% on the y-axis.

Keywords: Robotic arm, Arduino, Wireless, Automation

Abstrak

Di dunia sekarang ini, otomatisasi banyak digunakan dalam banyak aspek untuk membantu manusia melakukan pekerjaan mereka dan meningkatkan efisiensi pekerjaan. Robot merupakan salah satu jenis sistem otomasi yang banyak digunakan untuk membantu manusia dalam menyelesaikan pekerjaannya. Robot memiliki banyak peran dalam kehidupan manusia, dan robot digunakan untuk membantu manusia melakukan berbagai macam pekerjaan dengan lebih mudah. Robot banyak digunakan karena dapat mempermudah dan melakukan pekerjaan secara efektif dan efisien. Ada banyak implementasi robot untuk membantu manusia melakukan pekerjaannya, seperti robot pengelasan, robot pertanian, robot drone, dan banyak lagi. Lengan robot merupakan salah satu jenis robot yang sering kita jumpai, dan lengan robot memiliki fungsi yang hampir sama dengan lengan manusia. Dalam penerapannya robot dapat dikendalikan melalui beberapa metode, ada yang secara langsung menggunakan koneksi kabel, dan ada pula yang menggunakan metode nirkabel. Selain itu, untuk mengendalikan sebuah robot diperlukan sebuah pengontrol yang nantinya akan menjadi otak robot tersebut. Proyek ini bertujuan untuk mengembangkan pengontrol berbasis Arduino, salah satu papan mikrokontroler yang paling banyak digunakan untuk pendidikan, hobi, atau profesional, dan mengembangkan aplikasi perangkat seluler untuk mengontrol lengan robot melalui koneksi Bluetooth. Metodologi yang digunakan dalam penelitian ini dimulai dari identifikasi dan pendataan program dari beberapa sumber seperti jurnal atau forum diskusi, kemudian dilanjutkan dengan perancangan skema sistem perangkat keras yang digunakan dalam proyek ini dan akan dilanjutkan dengan pembuatan program aplikasi Arduino dan smartphone. Hasil pengujian menunjukkan akurasi gerak aktual cukup mendekati posisi yang diinginkan seperti yang diprogramkan, dan pengujian servo menunjukkan nilai error rata-rata sebesar 3,54% untuk servo MG996R dan 4,44% untuk servo SG90. Dengan demikian, nilai error robot adalah sekitar 1,38% pada sumbu x dan 12,14% pada sumbu y pada pengujian pertama. Dan untuk pengujian kedua sebesar 1,96% pada sumbu x dan 3,15% pada sumbu y.

Kata kunci: *Robotic arm, Arduino, Nirkabel, Otomasi*

1. Introduction

Nowadays, humans live in an age where technology is very developed, and humans get many advantages that a lot of work can be done easily with the help of an automation system. Automation systems cover many human shortcomings in doing work so that work can be done efficiently in terms of time, consistency, and quality. One of the branches of automation that are often found in daily life is robots, which are widely applied in various aspects of life, from industry, agriculture, household, health, military, and even entertainment.

In the industrial world, one of the types of robot that mostly used is the robot arm. This type of robot is used in various aspect. One of which is assembly component, welding, painting process, move things and many more. The main reason for the usage of the robot is because it has high accuracy, and it can work continuously, efficiently, and stably [1]

The advancement in communication technologies has made people very dependence on a gadget such as personal computer, laptop, and especially the smartphone. One of the reasons for this dependence is that with the existence of gadgets, especially smartphones, almost all needs can be resolved in just the grip and touch of a finger. Starting from contacting friends, setting reminders, opening and composing electronic mail, even doing various office and school work can be done only via a smartphone.

With the development of automation systems and also the increase in the use of smartphones for learning purposes, the aim of this project is to create a 5 DOF pick and place robotic arm system based on an Arduino controller that can be controlled using a smartphone via a Bluetooth module for educational purposes.

Robot Arm Components

A robotic arm is a system that consists of three main parts that connected and supported each other such as, the body or mechanical structure, the actuator, and the control system [2]. A robotic arm is similar to a human arm that has several joints that can allow the robotic arms to move and do several tasks like a human arm. The robotic arm can complete some tasks such as gripping, welding, and spinning depends on the design and application [3]. The robot arm design is very important in order to make a move of the robot arm. According to Singh et al. [4], one of the most important aspects is the degree of freedom in designing or create the robotic arm. Whereas the degree of freedom is the joint in the arm construction, which will allow the robot to translate, rotate or bend. The amount of the degree of freedom can be determined by the number of the actuator. The robot arm is consisting of several components that connected and supported each other. However, we can divide the components into three main parts starting from the mechanical structure, the control system, and the actuator.

1.1 Mechanical Structure

The mechanical structure is the first important parts due to the basic foundation of the robot arm build. This is because mechanical construction is a place where other components will be placed and also has a very important role in the work process of the robot.

The robot arm design is very important in order to make a move of the robot arm. According to Singh et al. [4], one of the most important aspects is the degree of freedom in designing or create the robotic arm. Whereas the degree of freedom is the joint in the arm construction, which will allow the robot to move such as translate, rotate or bend.

1.2 Control System

The second component is the control system which will be the brain in order to control the robot arm itself. Arduino Uno is an open-source microcontroller board powered by an ATmega328P microchip controller and developed by Arduino. Cc. Arduino is one of the most popular in the world for the reason that the Arduino has equipped with digital and analogue input/output pins that give Arduino flexibility in order to expand and connect it to another expansion board nor other circuits [3,5]. The Arduino is using a C programming language that very popular and widely used in programming. However, the Arduino website provides the software for programming the Arduino board [6].

The Bluetooth module is used to receive data and act as a slave. The master and slave in Bluetooth communication are determined by the state of the connection when it is formed, and the connection can only be started by the master module [7]. According to Reddy et al. [3], HC-05 is one of the Bluetooth modules that communicates with USART helps at a 9600 baud rate.

1.3 The Actuator

The actuator is the output component, which responsible for moving the mechanism. The actuator used in this project is a servo motor that 180 degrees of movement.

According to Latifa et al. [8], a servo is made up of four components, which the DC motor, a gear train, control circuit, and potentiometer. They also mentioned that the mechanism uses in the servo motor is a closed-loop feedback control system that will determine and ensure the angle of the motor output shaft.

2. Method

This research focuses on developing the robot arm controller using Arduino microcontroller board. The flow of the research will start from the observation and data collection, the system schematic design including the circuit and the mechanical structure, the Arduino program, the MIT App inventor program, and the last is the system testing.

2.1 Observation And Data Collection

In this observation, it is focused on learning and gather some information about the robot arm controller using Arduino. The aim of this project is to develop the 5 DOF robot arm controller using Arduino and smartphone as the remote control. There are several Arduino based robot arm project in several Arduino forum related to the main objective of the project. Then, modify and develop the system in order to achieve the aim of the project. There is also several information about the controller and servos specification.

Table 1. Arduino program and application data

No.	Servo Name	Servo Placement	Initial Position (Degree)	Min Value (Degree)	Max Value (Degree)
1	Servo01	Base	90°	0°	180°
2	Servo02	Shoulder	90°	40°	100°
3	Servo03	Elbow	180°	90°	180°
4	Servo04	Wrist	0°	0°	180°
5	Servo05	Grip	180°	110°	180°

2.2 System Schematic Design

The system consists of three main parts which is the input, the controller, and the output. In this system, the input command will be sent by the android device to the Arduino as the controller using Bluetooth connection where the HC-06 Bluetooth module is used. Then, the Arduino will process the input and send the output results to the servo motor as the actuator, and the final result is the movement on the robot arm. The schematic diagram is shown in Figure 1.

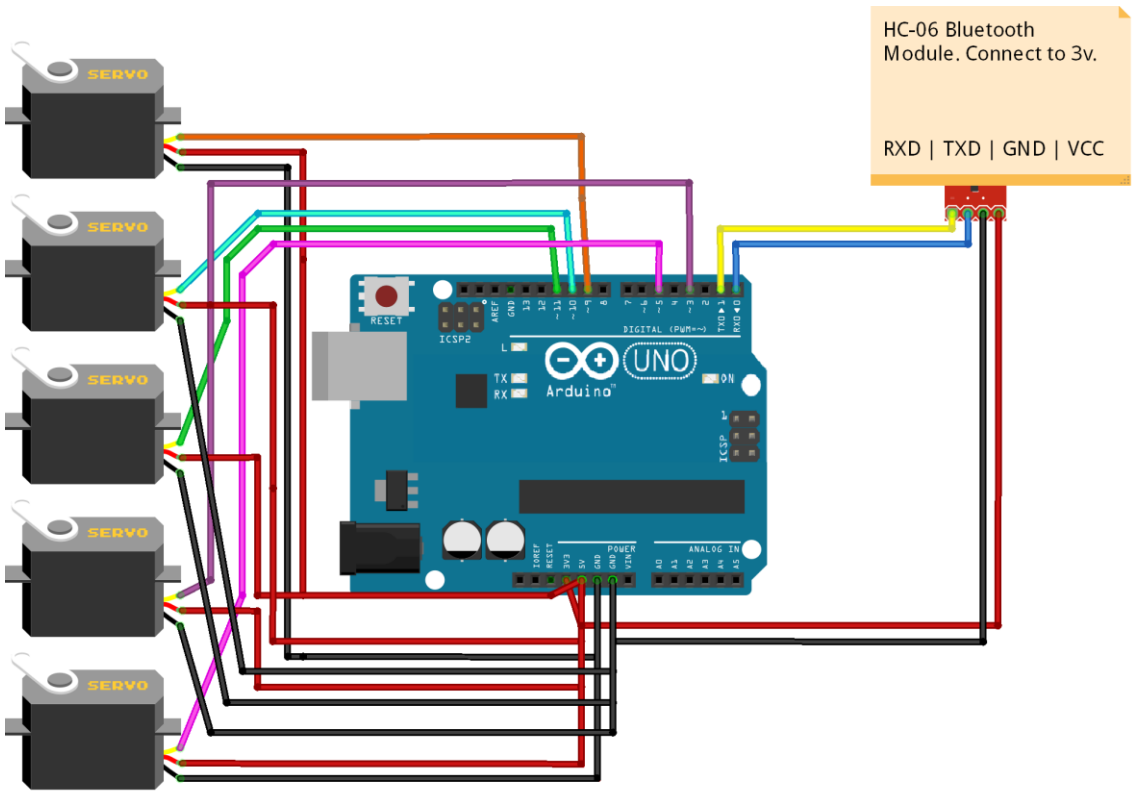

Figure 1. Arduino Schematic

2.3 Mechanical Schematic Design

The structure of the robot exhibited in Figure 2 consists of three main parts which is the body or link, the actuator or joint, and the end of effector. The robot has 4 actuators that placed in the link and one actuator in the end of effector. The four actuators have responsible to determine the position of the end of effector or in this case is the gripper, and the fifth actuator is used to move the mechanism of the gripper.

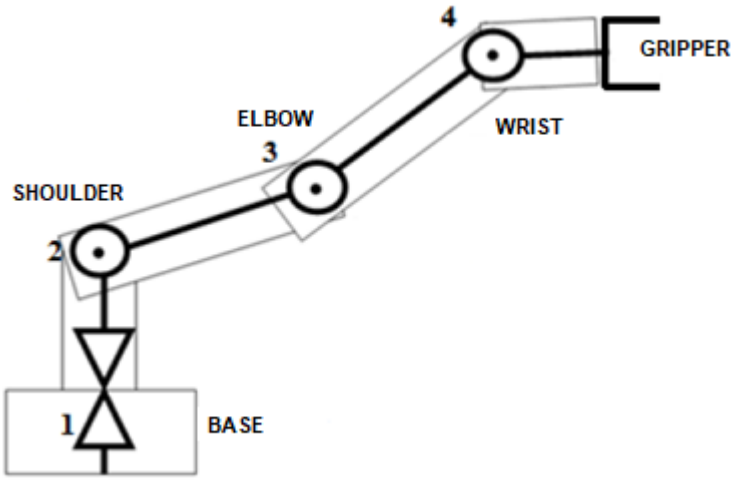

Figure 2. Mechanical system schematic

2.4 Arduino Programming

The Arduino program was made in Arduino IDE software with a 1.8.13 version that installed on PC. The first step is we need to define all the servos and also the Bluetooth module, also include the Software Serial library for the serial communication of the Bluetooth module and the servo library that included in Arduino IDE software. However, we also need to define several variables in order to store the position of the servos, as well as arrays for the automatic mode. The sample coding for preparation setup is shown in Figure 3.

robot_arm_5_dof

```
/*
      3D PRINTED ROBOTIC ARM CONTROLLED WITH ARDUINO
      BY; MUHAMMAD IRFAN SATRIA
      MECHANICAL ENGINEERING 2017
*/

#include <SoftwareSerial.h>
#include <Servo.h>

Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;

Servo servo05;

SoftwareSerial Bluetooth(0, 1); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos; // current position
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos; // previous position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50]; // storing positions or steps
int speedDelay = 20;
int index = 0;
String dataIn = "";
```

Figure 3. Arduino coding for preparation setup

In the setup section, the servos and Bluetooth module need to be initialized in order to make a move to the initial position. The write() function is used to moves the servo to a position in the 0 to 180 degrees range. Figure 4 shows the sample code of setup initialization.

```
void setup() {
  servo01.attach(5);
  servo02.attach(6);
  servo03.attach(7);
  servo04.attach(8);

  servo05.attach(10);
  Bluetooth.begin(9600); // Default baud rate of the Bluetooth module
  Bluetooth.setTimeout(5);
  delay(20);
  Serial.begin (9600);


  // Robot arm initial position
  servo1PPos = 90;
  servo01.write(servo1PPos);
  servo2PPos = 90;
  servo02.write(servo2PPos);
  servo3PPos = 120;
  servo03.write(servo3PPos);
  servo4PPos = 140;
  servo04.write(servo4PPos);
 servo5PPos = 80;
  servo05.write(servo5PPos);
}
```

Figure 4. Setup initialization

In the loop section, the Bluetooth.available() function is used. Then the data that come from the smartphone is constantly checked. If true, the data will read a string and store it into dataIn variable using the readString() function. Then the robot will do the order received by the Bluetooth module.

```
void loop() {
  // Check for incoming data
  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString();  // Read the data as string
```

Figure 5. Incoming data

In order to move the servo, the startsWith() function are used to check the prefix of each incoming data. However, we will know what to do. We get the location value using the Substring() function, convert it to an integer, and use the value to transfer the servo to the position.

```
// If "Waist" slider has changed value - Move Servo 1 to position
if (dataIn.startsWith("s1")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo1Pos = dataInS.toInt();  // Convert the string into integer
```

Figure 6. Slider changed value

However, the servo motor would run at maximum speed, which is too fast for the robot arm. Then, the speed of the servos needs to be controlled. In this case, we used for loop to shift the servo from the previous to the current position gradually by applying a delay period between iterations. Thus, the speed of the servo can be changed by adjusting the delay time. The sample code is shown in Figure 7.

```
  // If previous position is bigger then current position
  if (servo1PPos > servo1Pos) {
    for ( int j = servo1PPos; j >= servo1Pos; j--) {   // Run servo down
      servo01.write(j);
      delay(20);     // defines the speed at which the servo rotates
         Serial.print("Servo 1 = ");
  Serial.println(j);
  delay(60);
    }
  }
  // If previous position is smaller then current position
  if (servo1PPos < servo1Pos) {
    for ( int j = servo1PPos; j <= servo1Pos; j++) {   // Run servo up
      servo01.write(j);
      delay(20);
       Serial.print("Servo 1 = ");
  Serial.println(j);
  delay(60);
    }
  }
  servo1PPos = servo1Pos;   // set current position as previous position
}
```

Figure 7. Servo movement & delay time

The next code as it is exhibited in Figure 8, is the "SAVE" button, where the position of each servo is saved in an array when the "SAVE" button pressed. The index increases with each button gradually filling the list. On the other side, when the "RESET" button pressed, all the data from the arrays and the index will be cleared.

```
// If button "SAVE" is pressed
if (dataIn.startsWith("SAVE")) {
  servo01SP[index] = servo1PPos;  // save position into the array
  servo02SP[index] = servo2PPos;
  servo03SP[index] = servo3PPos;
  servo04SP[index] = servo4PPos;
  servo05SP[index] = servo5PPos;
  index++;                        // Increase the array index
}
// If button "RUN" is pressed
if (dataIn.startsWith("RUN")) {
  runservo();  // Automatic mode - run the saved steps
}
// If button "RESET" is pressed
if ( dataIn == "RESET") {
  memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
  memset(servo02SP, 0, sizeof(servo02SP));
  memset(servo03SP, 0, sizeof(servo03SP));
  memset(servo04SP, 0, sizeof(servo04SP));
  memset(servo05SP, 0, sizeof(servo05SP));
  index = 0;  // Index to 0
}
```

Figure 8. "SAVE" & "RUN"

2.5Custom Run Function

However, the run function is a custom function. When it activated, the system runs the stored steps repeatedly until the "RESET" button pressed. By using *for loop,* the system runs all the positions stored in the arrays. If the "PAUSE" button pressed, the robot would be paused, and if the button pressed again, the robot would continue with the automatic movement. Also, if the speed slider changed, the value will be used to change the delay time between each repetition.

2.6MIT App Inventor Program

The smartphone app program is made using MIT App Inventor with version nb186a. In this stage, the interface of the app is shown in figure 9.

Figure 9. Mobile phone App interface

At the top of the interface, there are two buttons for connecting the smartphone to the Bluetooth module. On the left side, there is three-button which is "SAVE", "RUN", and "RESET" with the "Position saved" display below it to see how many positions saved. And on the right side, there are six sliders, which five of them are used to move the servo, and the other one is used to adjust the delay time.

The input is using sliders instead of button and for the input data will based on the thumb position of the sliders. There is also some labels to each slider to show the value of the current position.

2.7 System Testing

After all of the program and application design also the schematic design of the system, an accuracy testing process will be conducted. The parameter that used in this accuracy test is the data from table 1, which comes from the Arduino program and a smartphone app that has a built-in previous chapter. There will be two types of accuracy test. The first one is the servo accuracy test, where the accuracy of the servo will be tested one by one using a protractor as the measuring tool in order to measure the deviation of the output. Then, the second test will be the accuracy test for the end of the effector of the robot. The second test will conduct after the servo assembled with the robot arm body. The method that will be used in this second test is by moving the robot arm from one to another desired position and measure the displacement on the target. Then, the results will be compared with the target position data in order to get the error value. The result of the test will be shown in the next chapter.

3. Result And Discussion

3.1 Robot Assembly and Test Template

Here is the result of the assembled robotic arm. All of the parts that have been assembled are made using the 3D printer with PLA material. And the base of the robot arm is screwed on the top of the table and also the test template. This robot arm has 5 degrees of freedom. However, there are only four servos that will affect the positioning of the robot, which is the waist, shoulder, elbow, and wrist.

The last servo is attached to the gripper mechanism. Figure 10 shows the result of the assembled robot arm, the controller, and the test template.

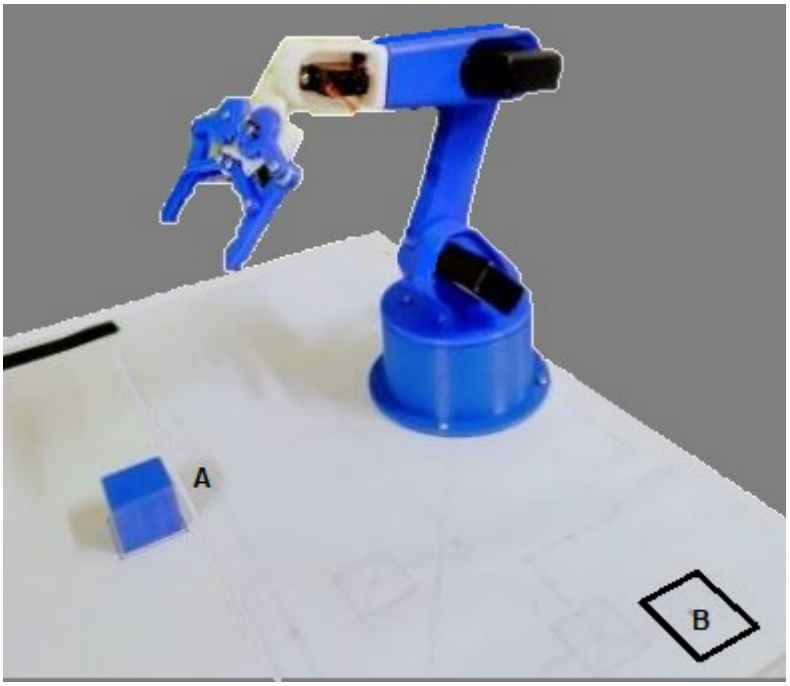

Figure 10. Robot assembly on test template

3.2 Accuracy Of The Servos

The first test that will conduct is the servo accuracy test. This simple test is used to know the error percentage of each servo. The servo rotation will be measured using a protractor. After we get the value, then the value of the test result will be compared with the value from the program. The picture of the test is shown in the Figure 11 and the result of the servo test shown in Table 2.

Figure 11. Servo accuracy test

Table 2. Servo test result

Servo name	Test	1	2	3	4	5	6	7	8	9	10
Servo01 (Waist)	Input (degree)	75	90	100	180	133	140	63	162	117	154
Initial position 90°	Output (degree)	78	93	103	175	130	135	68	158	115	150
	Error	3,85%	3,23%	2,91%	2,86%	2,31%	3,70%	7,35%	2,53%	1,74%	2,67%
									Average error		3,31%
Servo01 (Shoulder)	Input (degree)	40	63	77	100	55	74	49	90	62	53
Initial position 90°	Output (degree)	41	66	79	98	59	75	52	92	65	57
	Error	2,44%	4,55%	2,53%	2,04%	6,78%	1,33%	5,77%	2,17%	4,62%	7,02%
									Average error		3,92%
Servo01 (Elbow)	Input (degree)	90	180	140	160	112	132	105	161	147	165
Initial position 180°	Output (degree)	91	175	135	166	108	126	102	155	141	160
	Error	1,10%	2,86%	3,70%	3,61%	3,70%	4,76%	2,94%	3,87%	4,26%	3,13%
									Average error		3,39%
Servo01 (Wrist)	Input (degree)	47	79	40	72	20	108	81	58	90	63
Initial position 0°	Output (degree)	50	75	44	75	19	104	83	60	92	65
	Error	6,00%	5,33%	9,09%	4,00%	5,26%	3,85%	2,41%	3,33%	2,17%	3,08%
									Average error		4,45%
Servo01 (Gripper)	Input (degree)	50	80	34	70	25	105	80	55	90	60
Initial position 180°	Output (degree)	52	75	35	72	27	100	83	58	92	63
	Error	3,85%	6,67%	2,86%	2,78%	7,41%	5,00%	3,61%	5,17%	2,17%	4,76%
									Average error		4,43%

From the result above, we can see that the average error of the MG996R servo that placed in the waist, shoulder, and elbow is around 3,31% to 3,92%. Then, the SG90 servo that placed in the wrist and gripper has 4,45% and 4,43% of error.

3.3 Pick – Place And Repeatability Test

In this test, the robot arm will be mounted on top of the test template. Then, the robot will perform the pick and place also the repeated movement test. The robot will be assigned to move the object from point A to point B, then from point A to point C. The task will be repeated ten times for both points A to B and point A to C. The first movement will be the pick and place, then from number two until nine will be the repetition. The test results are shown in Table 3 and Table 4.

Table 3. Point A to B result

Test No.	Target Position		Displacement		Error Percentage	
	X	Y	X	Y	X	Y
1	215	15	216	14,5	0,46%	3,45%
2			214	14	0,47%	7,14%
3			218	12,5	1,38%	20,00%
4			212	13,5	1,42%	11,11%
5			221	13,2	2,71%	13,64%
6			211	14	1,90%	7,14%
7			214	13	0,47%	15,38%
8			216	13,3	0,46%	12,78%
9			207	13	3,86%	15,38%
10			216,5	13	0,69%	15,38%
			Average error		1,38%	12,14%

Table 4. Point A to C result

Test No.	Target Position		Displacement		Error Percentage	
	X	Y	X	Y	X	Y
1	170	114	171	117	0,58%	2,56%
2			169	118	0,59%	3,39%
3			169	116	0,59%	1,72%
4			174	121	2,30%	5,79%
5			163	113	4,29%	0,88%
6			171	118,5	0,58%	3,80%
7			172	120	1,16%	5,00%
8			168	117	1,19%	2,56%
9			173,5	121	2,02%	5,79%
10			160	114	6,25%	0,00%
			Average error		1,96%	3,15%

From the result in Table 3, we get an average error of 1,38% in the x-axis and 12,14% in the y axis respectively. In the second test, which movement from point A to point C, we get 1,96% error for the x-axis and 3,15% for the y axis. The error at the y-axis in A to B movement is higher than the error in A to C movement.

3.4 Discussion

There is a high error percentage in the y-axis on the first test, which the movement from point A to point B. The problem is because the servo cannot reach the minimum and maximum value based on the service specification. It is believed that the problem is due to an issue with the servo specification. The robot also has problems in the shoulder, elbow, and gripper mechanism. There are many vibrations and the movement of the robot are hobbled. A small spring that placed in the shoulder is one of the best options to handle that. The gripper mechanism also needs some improvement in order to minimize and prevent the slip of the object.

4. Conclusion

After creating, analyzing, and testing the program and the prototype, several conclusions can be drawn:

- The Arduino program and the simple android apps that have made using MIT App Inventor can perform the pick and place task wirelessly using a Bluetooth connection.
- The program also has the ability to perform save and repeat the position.
- The average error value of the servo motor is about 3,54% for the MG996R servo and 4,44% for the SG90 servo.
- The error of the robot for the x-axis is about 1,38% in the first test and 1,96% for the second test. However, the result for the y-axis is a little bit higher, which 12,14% for the first test and 3,15% for the second test.
- The high error value in the y-axis at the first test is caused by the servo efficiency and the movement range of the servo. We only get a 5° - 175° rotation range instead of 0° - 180° of rotation.

- The other factor that affects the accuracy of the robot is the vibration that happens when the servo moves. It because the servo is running in a critical situation. Thus, the servo needs several supports, such as adding a small spring in several positions, such as at the shoulder, in order to help the servo in carrying the load.

This project still lacks in many aspects. Hopefully, in the future, there will be more research to improve the robot arm performance. In the coming future, for this project, it is suggested to develop the Arduino and android application program in order to make it easier for the user to get the specific value of the input. Implementing inverse and forward kinematics into the program will improve the input and also make easier to control and calibrate the robot. The second is using a better servo motor to carry more load and create a smoother movement of the robot. The stepper motor is recommended to fulfil that task.

References

[1] Siafudin, A., Sumardi, S. and Darjat, D., 2017, "*Perancangan Sistem Kendali Pergerakan Arm Manipulator Berbasis Sensor Inertial Measurement Unit (Imu) Dan Sensor Flex*," Transient, 6(3): 424−431.
[2] Didi, M., Marindani E. D. and Elbani, A., 2015,"*Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno*," Jurnal Teknik Elektro Universitas Tanjungpura: 1−11
[3] Reddy, R.K, et al., 2020, "*Robotic Arm Control Using Arduino*," Journal of Emerging Technologies and Innovative Research, 7(6): 453−455.
[4] Singh, P., Kumar, A and Vashisth, M., 2013, "*Design of a Robotic Arm with Gripper & End Effector for Spot Welding*," Universal Journal of Mechanical Engineering, 1(3): 92−97.
[5] Sutisna, N.A & Farand, M.Q., 2020, "*Design and Prototyping of Mini AGV with Arduino Microcontroller*," Rotasi, 24(2): 119−126
[6] Bonini, N. et al., 2014, "*Robotic Hand in Motion Using Arduino- Controlled Servos*."
[7] Ghiet, M. A. and. Baba, A., 2017, "*Robot Arm Control With Arduino*," Ankara.
[8] Latifa, U. and Saputro, J.S., 2018, "*Perancangan Robot Arm Gripper Berbasis Arduino Uno Menggunakan Antarmuka Labview*," Barometer, 3(20): 138−141.