Peer-Reviewed Article

IOTA (Internet of Things and Artificial Intelegence) Journal
Volume 1, Issue 1, 2021

RCWS Control System Design Based on Raspberry Pi

Aswin Danu[1,2*], Lukman Maajid[1], Kasiyanto[2]
[1,2]*Department of Electrical Engineering, University of Merdeka Malang, Indonesia,*
[1,2]*Institute for Technology Studies (Lemjiantek), Malang, Indonesia*

*Corresponding Author: Aswin Danu, aswindanu@unmer.ac.id

Abstract
The Raspberry Pi was used in Control system projects, this project is intended to realize a system, i.e., the Raspberry Pi-based RCWS control system, where the RCWS movement uses a DC motor therefore, that it will make it easier for shooters to reach their targets. Thus, the shooter can determine the position of the weapon that will be used at a distance from the shooter's position by using the smartphone screen to monitor the target or target to be shot. The test results show that [1] Raspberry Pi can be used to control RCWS to move towards elevation and azimuth angles when determining the target determined via a web browser, where the voltage generated by the GPIO pin is in a low condition of 0.5mV while in high conditions it is equal to 3,282V.

Keywords: Raspberry Pi, elevation, azimut, DC Motor, RCWS

1. Introduction

The development of science and technology has been sophisticated in some developed countries, therefore, that weapons systems that can be controlled remotely with electronic control are implemented. Most of the defense equipment owned by the Indonesian Army are still operated manually therefore, that the need for the defense equipment of the Indonesian Army requires a new alternative in weapon development according to technological advances in the field of electronics. By utilizing Android technology, it can be used to assist weapon technology in the military field. Therefore, the shooter can determine the position of the weapon to be used at a distance from the shooter's position, by using the smartphone screen to monitor the target or target to be shot. The use of this technology can aim to help reduce losses in the field of personnel so that the shooter will have a high level of security. Based on this, a Raspberry Pi-based RCWS Control System Design was made (Simon, 2015).

2. Literature Review

Raspberry Pi (Figure 1) is a mini kit that can be turned into a mini-computer the size of a credit card and weighs only 45 grams. This computer, named *Raspberry Pi*, runs the Linux operating system (Craven, Paul Vincent, 2011). Moreover, DC motor (Figure 2) is a basic electromechanical device that functions to convert electric power into mechanical power. Furthermore, Hardware used is SS1 V3 (assault rifle 1 variant 3) in accordance with the codification of PT Pindad as the domestic producer of defense equipment. This weapon was created to continue the success and improvement of its predecessor, the SS1 V2 rifle.

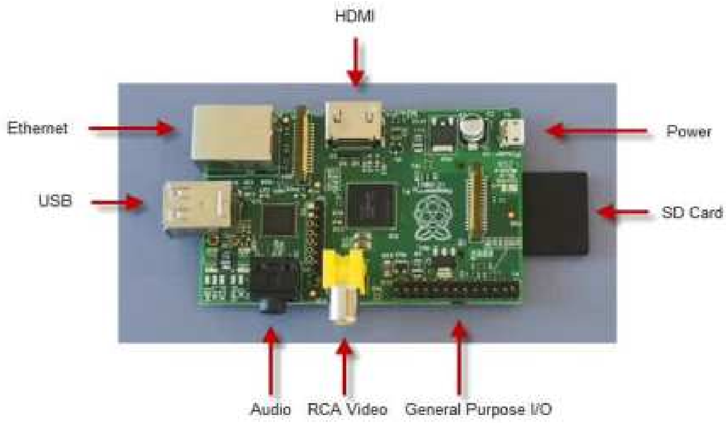


Figure 1. ***Raspberry pi***

Python is a multipurpose programming language with a design philosophy that focuses on the readability of the code. Python is claimed to be a language that combines capabilities, capabilities, with clear code syntax and is equipped with a large and comprehensive standard library functionality. One of the features available in python is a dynamic programming language equipped with automatic memory management. As with other dynamic programming languages, Python is generally used as a scripting language even though in practice the use of this language is broader covering contexts of use that are generally not done using a scripting language. Python can be used for various software development purposes and can run on various operating system platforms.

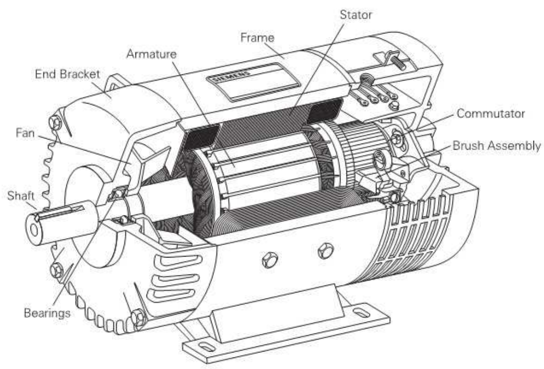


Figure 2. ***DC Motor***

Figure 3. ***SS1 V3 Weapons***

3. Method, Data, and Analysis

The block diagram of the RCWS Control system is shown in Figure 4.

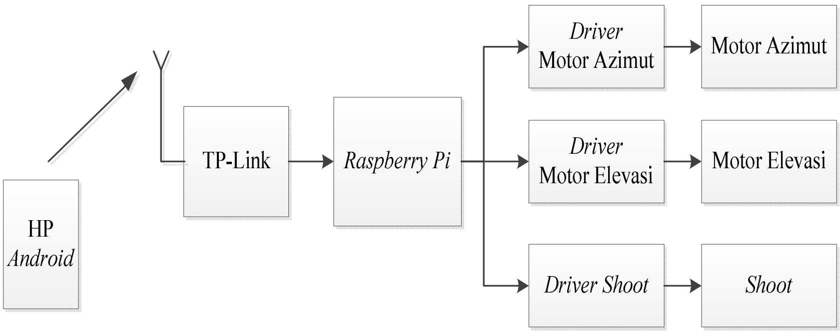


Figure 4. ***Block Diagram of the RCWS Control System***

The following is the function of each block in Figure 4:

1. Raspberry Pi functions to control motor azimuth and motor elevation by receiving commands from the cellphone.
2. Android functions to send commands to Raspberry Pi with an access point on a certain IP via TP-link.
3. TP-Link is a function of bridging communication between Android and Raspberry Pi by going through an access point according to a predetermined IP.
4. The azimuth motor driver and elevation motor driver function to control (on/off) the motor and determine the direction of motor rotation
5. Driver shoot functions to control the weapon (on / off) SS1 V3.

The Raspberry Pi is used in the RCWS control system design, as shown in Figure 5.

Figure 5. ***SS1 V3 Weapons***

The ports used to move the motor to azimuth and elevation angles in the RCWS control system are as follows:

a) GPIO 2 as azimuth motor on/off
b) GPIO 3 as rotating left / right azimuth motor
c) GPIO 18 as the on/off elevation motor
d) GPIO 17 as rising / falling motor elevation
e) GPIO 27 as a shoot

One of the most popular methods for controlling the rotation speed of a DC motor is using PWM (Pulse Width Modulation). With this PWM technique, the DC motor is given a stable voltage source with the same working frequency buttons of the duty cycle, the DC motor speed control pulse varies. In designing the Raspberry Pi-based RCWS control system, it has a PWM concept found on the GPIO 18 pin which functions to regulate the RCWS movement towards the elevation angle. The PWM concept on the Raspberry Pi is to adjust the width of the positive and negative sides of the control pulse at a fixed working frequency. The wider the positive pulse side, the higher the DC motor rotation speed, and the wider the negative pulse side, the lower the DC motor rotation speed. The motor driver circuit used in the RCWS control system design is as shown in Figure 6 and Figure 7.

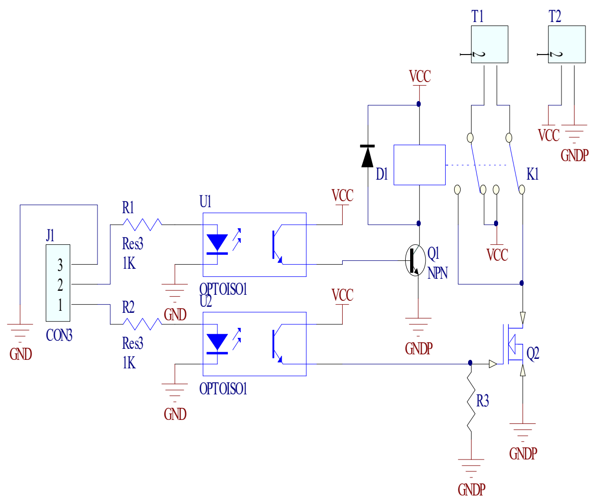


Figure 6. ***Motor Driver Schematic***

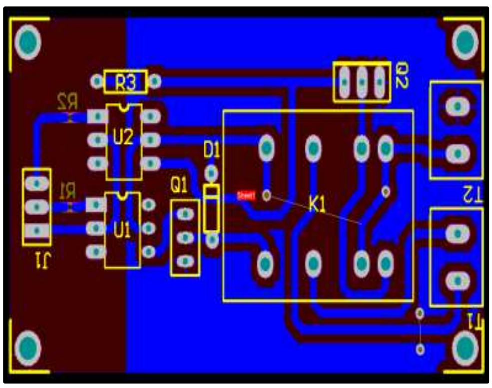


Figure 7. ***Motor Driver Layout***

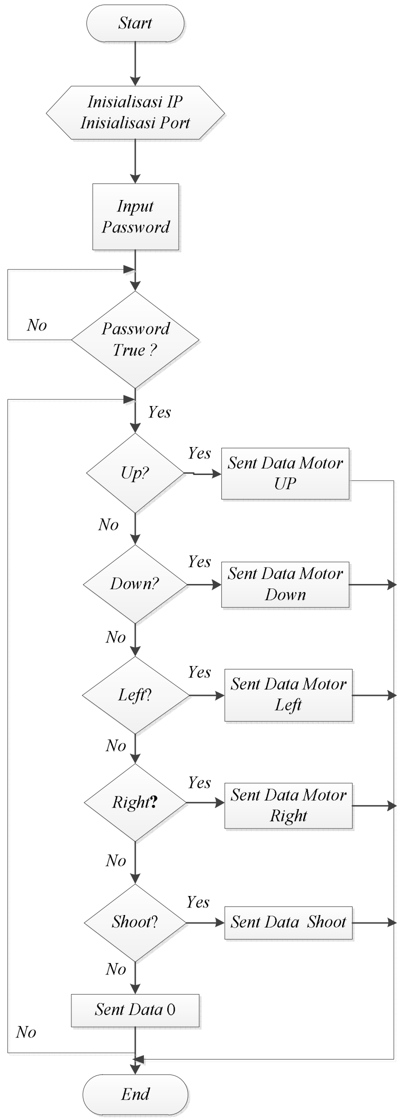


Figure 8. Flowchart of the Android Program Work System

The web interface is a communication system between the user (user) and the device (Raspberry Pi), where the web interface can control the GPIO on the device and provide information about the device itself. The Raspberry Pi based RCWS control design uses an open-source application designed to control the GPIO via a web browser. The application used is a framework for the Python programming language which includes the Apache Web Server.

Following are the steps for installing WebIOPi are:

1. This application can be downloaded directly via the Raspberry Pi with the wget command, written with the following command:

@ raspberry $ wget http: //webiopi.googlecode.com/files/WebIOPi- 0.7.0.tar.gz

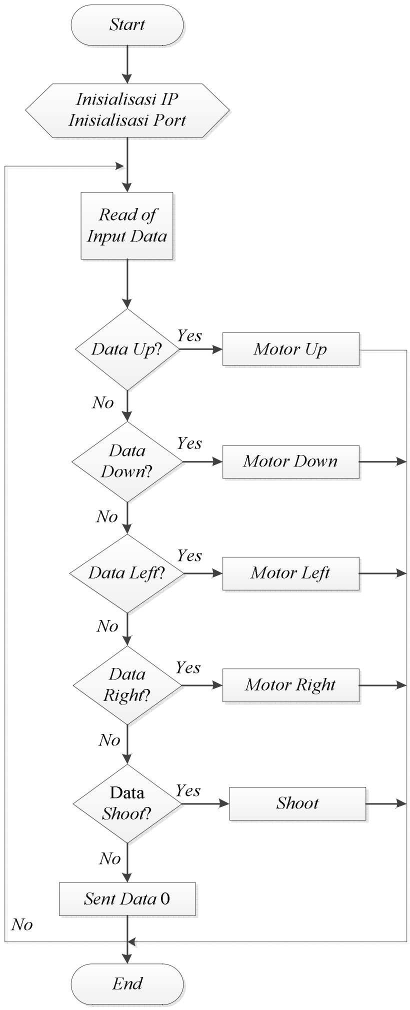


Figure 9. Flowchart of the Raspberry Pi Program Work System

2. The downloaded data is in the form of tar.gz data which is a collection of data that must be extracted, to extract the data the zxvf tar command is used, written with the following command:

pi @ raspberrypi $ tar zxvt WebIOPi-0.7.0.tar.gz

3. The data that has been extracted is in the form of raw data for the WebIOPi installation that is already in one folder and then entered into that folder, written with the following command:

pi @ raspberrypi $ cd WebIOPi-0.7.0

4. After entering the folder, the WebIOPi application is ready to be installed with the command "sudo./setup.sh", written with the following command:

pi @ raspberrypi $ sudo./setup.sh

5. If it's finished correctly, then check whether this application has been installed correctly, written with the following command:

pi @ raspberrypi $ ls -l

6. Next, run WebIOPi which has been installed, written with the following command:

pi @ raspberrypi $ sudo /etc/init.d/webiopi start

7. This command is used to check whether WebIOPi is running properly:

pi @ raspberrypi $ sudo /etc/init.d/webiopi status

8. Open a browser and enter http: // IpRaspberry: 8000.

The working system of WebIOPi cannot be separated from the Python programming language, where the Python program functions to bridge the Raspberry Pi device with a web browser. The web browser uses HTML and Javascript programming languages. WebIOPi includes an HTTP server that provides an HTML source and a REST API for controlling the Raspberry Pi. Where the browser will load the HTML file including Javascript and will make an Asynchronous call to the REST API which functions to control and update the UI (User Interface). This method is very efficient, as there is no need to reload and download entire pages.

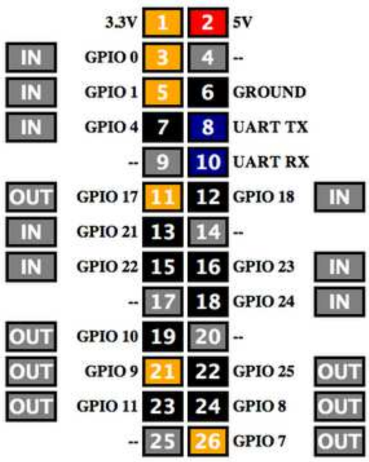


Figure 10. Pin GPIO pada WebIOPi

The design and programming of the RCWS control system use the Python language which consists of several program scripts. The following Python programming language for controlling weapons (gunfire) :

```
import webiopi #inisialisasi WebIOPi
import datetime #inisialisasi delay time

GPIO = webiopi.GPIO #inisialisasi  GPIO pada webiopi

MAZIMUT = 2   # GPIO pin motor azimut
DAZIMUT = 3   # GPIO pin direction motor azimut
MELEVASI = 4  # GPIO pin motor elevasi
DELEVASI = 17 # GPIO pin direction motor elevasi
SHOOT = 27    # GPIO pin shoot
```

```
DATA_A = 0
DATA_B = 0

# setup function is automatically called at WebIOPi startup
def setup():
    # set the GPIO used by the signal to output
    GPIO.setFunction(MAZIMUT, GPIO.OUT)
    GPIO.setFunction(DAZIMUT, GPIO.OUT)
    GPIO.setFunction(MELEVASI, GPIO.OUT)
    GPIO.setFunction(DELEVASI, GPIO.OUT)
    GPIO.setFunction(SHOOT, GPIO.OUT)

    # Set semua ouput pada logika 0
    GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
    GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
    GPIO.digitalWrite(MELEVASI, GPIO.LOW)
    GPIO.digitalWrite(DELEVASI, GPIO.LOW)
    GPIO.digitalWrite(SHOOT, GPIO.LOW)

# loop function is repeatedly called by WebIOPi
def loop():

    if (DATA_A == 1):
        GPIO.digitalWrite(MAZIMUT, GPIO.HIGH)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.LOW)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)
    elif (DATA_A == 2):
        GPIO.digitalWrite(MAZIMUT, GPIO.HIGH)
        GPIO.digitalWrite(DAZIMUT, GPIO.HIGH)
        GPIO.digitalWrite(MELEVASI, GPIO.LOW)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)
    elif (DATA_A == 3):
        GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.HIGH)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)
    elif (DATA_A == 4):
        GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.HIGH)
        GPIO.digitalWrite(DELEVASI, GPIO.HIGH)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)
    elif (DATA_A == 5):
        GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.LOW)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.HIGH)
    else:
        GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.LOW)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)
```

```
    # gives CPU some time before looping again
    webiopi.sleep(1)

# destroy function is called at WebIOPi shutdown
def destroy():
        GPIO.digitalWrite(MAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(DAZIMUT, GPIO.LOW)
        GPIO.digitalWrite(MELEVASI, GPIO.LOW)
        GPIO.digitalWrite(DELEVASI, GPIO.LOW)
        GPIO.digitalWrite(SHOOT, GPIO.LOW)

@webiopi.macro
def getData():
    return "%d;%d" % (DATA_A, DATA_B)

@webiopi.macro
def setDATA(A, B):
    global DATA_A, DATA_B
    DATA_A = int(A)
    DATA_B = int(B)
    return getData()
```

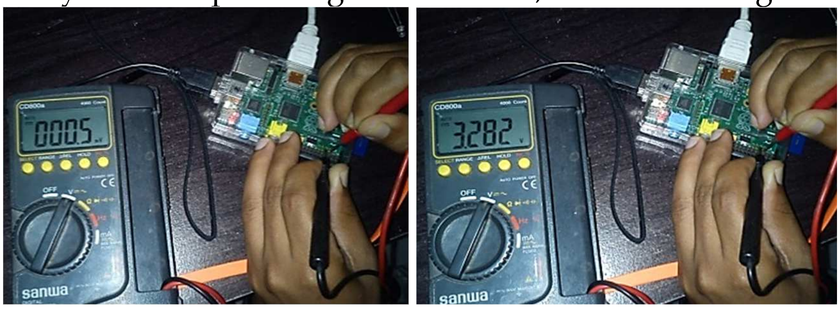
Meanwhile, the following is a WebIOPi program using HTML and Javascript, which are:

```
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>WebIOPi | RCWS Control</title> //Title judul web
        <script type="text/javascript" src="/webiopi.js"></script>
//inisialiasasi java script
        <script type="text/javascript">
        //fungsi pembacaan data
        webiopi().ready(function() {
                    var updateData = function(macro, args, response) {
                    var data = response.split(";");
                    // Following lines use jQuery functions
                    $("#input1").val(data[0]);
                    $("#input2").val(data[1]);
                }

                // pengambilan informasi dan memperbarui informasi pada
device
                webiopi().callMacro("getData", [], updateData);
                var sendButton = webiopi().createButton("sendButton",
"Send", function() {
                    var hours = [$("#input1").val(),
$("#input1").val()];
                    // pemanggilan fungsi macro pada phyton
                    webiopi().callMacro("setData", data, updateData);
                });
                webiopi().refreshGPIO(true);
        });

        </script>
</head>
```

```
<body>
      <div align="center">
      Send Android Data :<input type="text" id="input1" /><br/>
      <div id="controls"></div>
      </div>
</body>
</html>
```

4. Results

Testing the Raspberry Pi GPIO pin is done by measuring the output voltage on the Raspberry Pi GPIO pin using a multimeter, as shown in Figure 11 a and b.



(a) GPIO Low Logic (b) GPIO High Logic

Figure 11. GPIO Logic

The results of testing the GPIO output on the Raspbby Pi can be seen as shown in Table 1.

Table 1. GPIO pin test results on the Raspberry Pi

No	Condition	V out GPIO
1	Low (0)	0.5 mV
2	High (1)	3.282 V

The Raspberry Pi GPIO pins are used as outputs that are connected to a motor driver circuit that will control the direction of the motor rotation. The pins used are GPIO02, GPIO03, GPIO04, GPIO17, and GPIO27, each pin outputs the same output voltage, which is 3.282V for high logic and 0.5mV for low logic. When the logic conditions are high, the output generated by the GPIO pin will be sent to the motor driver circuit so that the motor driver turns on. Conversely, when the logic condition is low, the motor driver will turn off. The test results of the motor driver circuit consist of the test results of the azimuth direction motor driver circuit and the test results of the elevation direction motor driver circuit. The resulting motor rotation direction can be seen as shown in Table 2 and Table 3.

Table 2. Azimut Direction Motor Driver Data

No	Driver Logic		DC Motor
	Pin Direction	Pin On/Off	
1	0	0	Off
2	1	0	Off
3	0	1	Turn Right
4	1	1	Turn Left

Table 3. Elevation Direction Motor Driver Data

No	Driver Logic		DC Motor
	Pin Direction	Pin On/Off	
1	0	0	Off
2	1	0	Off
3	0	1	Up
4	1	1	Down

The WebIOPi test results of the RCWS control system can be seen as shown in Figure 12 and Figure 13.

(a) Test Results when the GPIO Pin is Off (b) Test Results when the GPIO Pin is On

Figure 12.a and b Test Results when the GPIO

The LED as a temporary output is connected to the GPIO 3 pin of the Raspberry Pi and then controlled using Android via a web browser according to the system configuration procedure. Figure 12a shows that the GPIO 3 pin is off, this condition is indicated by the LED which is off. Whereas Figure 12b shows that the GPIO 3 pin is on, this condition is indicated by the LED which is on. Apart from using Android, the WebIOPi test for the RCWS control system is also carried out by using another device that has a web browser.

The purpose of testing the whole tool is to determine the function of the RCWS control system as a whole. The desired work order is to control the RCWS towards the X-axis and the Y-axis (azimuth direction and elevation direction). Besides, the objectives of the overall tool testing include 1. Know the accuracy of the weapon control system. 2. Calculate the average accuracy of the number of repeated shots in the same distance. 3. Knowing the accuracy of the target at a shooting distance of 5 meters. The results of the lesion test with a distance of 5 meters from the RWCS position can be seen as shown in Figure 13.

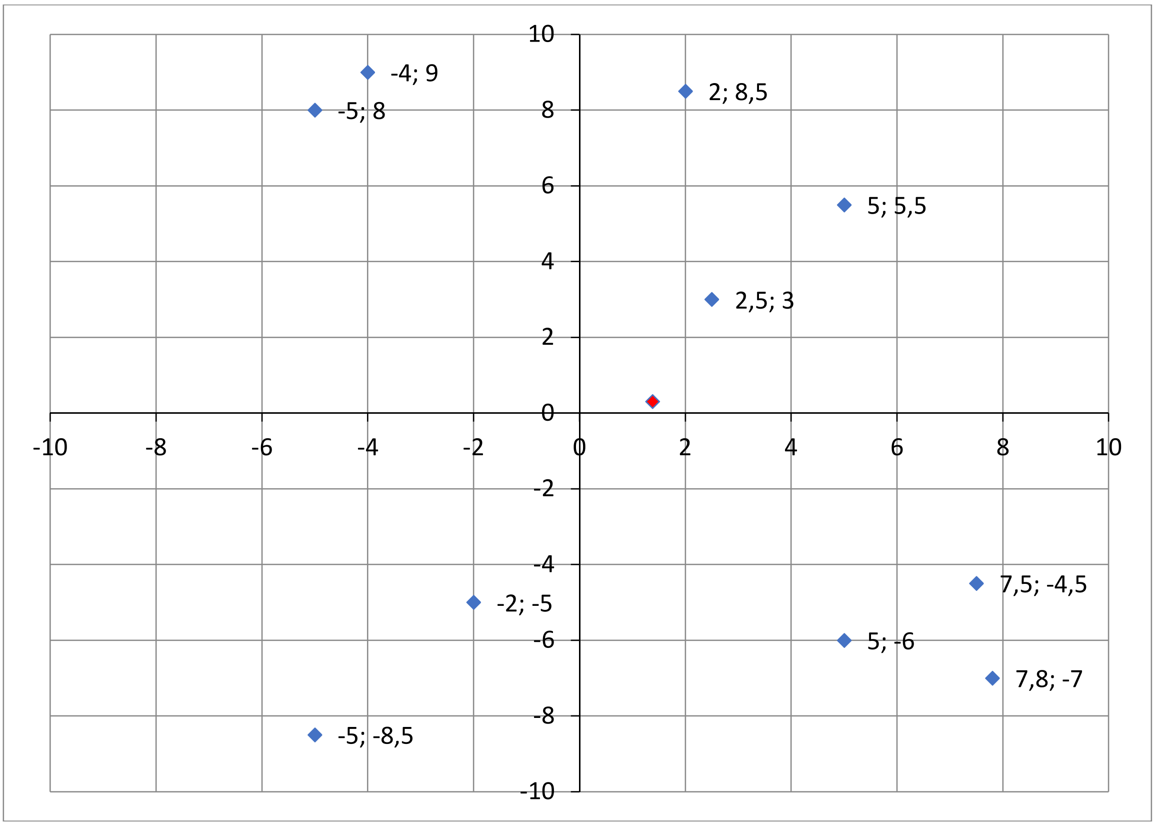


Figure 13. Results of the TKRR Coordinates with a Distance of 5 Meters

Based on the test results in Figure 4.10, it can be seen that the results of the shooting with the closest distance from the target center of the lesion are located on the axis (2,5; 3), meaning that the position of the RCWS when firing was not right according to the position of the center of the lesion. The results of the firing show that the position of the RCWS is on the X axis (2,5) and the Y axis (3), this axis indicates that the position of the weapon is to the right of the upper part of the target center of the lesan.

Table 4. Test Results Shooting Lesan Distance of 5 Meters

No	Sumbu X	Sumbu Y
1	-2	-5
2	5	5.5
3	2.5	3
4	-5	-8.5
5	7.8	-7
6	2	8.5
7	5	-6
8	-4	9
9	-5	8
10	7.5	-4.5

Shots were fired 10 times to obtain an average yield of 5 meters distance, as shown in Table 4.

5. Conclusion

Based on the results of designing, manufacturing, and testing the Raspberry Pi-based RCWS Control System Design, it can be concluded that: 1. Raspberry Pi can be used to control RCWS, where the test results show that the voltage generated by the GPIO pin in a low condition is 0.5mV while at the high condition it is 3.282V. 2. The motor driver can work as desired, which is to receive commands from the Raspberry Pi and then control the DC motor. The azimuth motor driver functions to move the motor towards the azimuth angle (right / left), while the elevation motor driver functions to move the motor towards elevation (up / down). 3. The Raspberry Pi-based RWCS control system can be controlled via a web browser / WebIOPi using all devices that have a web browser. 4.Based on the test shooting lesions carried out several times, it can be concluded that the Raspberry Pi-based RCWS control system can function as desired, where the RCWS can move towards the azimuth angle and to the elevation angle when determining targets according to the command when controlled.

6. Suggestions

As a development of the research that has been done, the authors suggest that the motion of the elevation angle and azimuth in the RCWS control system be replaced by using a stepper motor instead of a DC motor with the aim of eliminating centrifugal force.

References

Monk, Simon., Adafruit's Raspberry Pi Lesson 4. GPIO, Adafruit Learning System, (Online), mei 2015. (http://learn.adafruit.com/downloads/pdf/adafruits-raspberry-pi-lesson-4-gpio-setup.pdf, access mei, 5 2015).

Craven, Paul Vincent. Introduction to Computer Science Using Python and Pygame. 2011.

Python Software Foundation Team. Python v2.7.2 Documentation 1990 – 2011. The Python Software Foundation.

Upton, E. & Halfacree, G., Raspberry Pi User Guide, (Online) april 2013, (http://www.myraspberry-pi.org/wp-content/uploads/2013/02/Raspberry.Pi_.User_.Guide_.pdf, access april, 4, 2015).

Daryanto, Drs. 2000. Pengetahuan Teknik Elektronika. Jakarta: Penerbit PT Bumi Aksara.

Harahap, Rusli. 1996. Mesin Listrik Arus Searah. Jakarta: Penerbit PT Gramedia Pustaka Utama.

Malvino, Albert Paul. 1986. Prinsip-prinsip Elektronika. Jakarta: Penerbit Erlangga.

Malvino, Albert Paul. 1979. http://elektronikadasar.net/Prinsip-Prinsip Elektronika. Jakarta: Erlangga, access April, 15 2015

Teori dasar-dasar elektronika, Erlangga, 2000. Yogyakarta.

http://depokinstruments.com/2012/08/09/teori-motor-dc-bersikat-brushed-dc-motor-theory/) , access 2015.