Industrial Robotic Arm Vehicle Controlled Wirelessly Using Joystick

by

Benson Zindere (R146628T)

Submitted in partial fulfilment of the requirement for the degree of

BSC (HONS) in TELECOMMUNICATIONS

Department of Applied Physics &Telecommunications in the Faculty of Science & Technology

Midlands State University

Gweru, Zimbabwe

June 2018

Supervisors: [Dr. C. Nyamhere and Mr Mudzingwa]

[HTEL 438 Dissertation]

Abstract

These days, robots, for example, line robot car, mechanical arm, robotic finger and other products were being utilized for several purpose. Robots convey a considerable measure of advantages to the general public, so they are prominently utilized as a part of industry, house and work put. In this undertaking, the outline of wireless industrial robotic vehicle ought to have the capacity to deal with and lift up an object and furthermore can be controlled utilizing remote joystick. The Play Station 2 controller interface between the mechanical arm vehicle and the microcontroller (Arduino UNO), if the signal is transferred from the joystick, at that point Arduino UNO will get signall and begin to control automated arm. The remote part utilizes the Radio Frequency link that is 2.4GHz between joystick and Arduino UNO. Any signal transferred to the Arduino UNO should go through the PS2 controller in order to run the industrial robotic arm vehicle.

Declaration

I, **Benson Zindere** hereby declare that I am the sole author of this dissertation entitled "Industrial Robotic Arm Vehicle Controlled Wirelessly Using Joystick". I authorize Midlands State University to this dissertation only for purposes of scholarly research.

Signature……………………………………….. Date………………………………………

Approval

This dissertation entitled "Industrial Robotic Arm Vehicle Controlled Wirelessly Using Joystick" by **Benson Zindere** meets the regulations governing the award of the degree of *Bsc Telecommunications Honours* of the Midlands State University, and is approved for its contribution to knowledge and literal presentation.

Supervisor…………………………….............Date…………………………………………….....

Acknowledgements

For this awesome opportunity and to every one of the general people who made my work success I want to offer my genuine thanks. For the times of constant studying thanks are to God and my parents first. For with their extraordinary effect and inspiration I have figured out how to get this far in my studies. Likewise an abundance of thanks goes to PowerTel Communications for giving a business related place in the Organization, Coming from University with non to very little knowledge of the industry; the team at PowerTel received me well and built the me to what I am presently. To wrap things up my special thanks and appreciation goes to my supervisors Dr. C. Nyamhere and Mr Mudzingwa for the direction and supervision all through the entire study.

Abbreviations

RF Radio Frequency

PS2 Play Station 2

IDE Integrated Development Environment

USB Universal Serial Bus

AC Alternating Current

DC Direct Current

PWM Pulse Width Modulation

ENA Enable A

ACK Acknowledge

ATT Attention

CMD Command

SCK Synchronous

CLK Clock

GND Ground

SRAM Static random access memory

DOF Degree of Freedom

CONTENTS

CHAPTER ONE 12

INTRODUCTION 13

1.0 Introduction 13

1.1 Background 13

1.2 Scope and objective of the research 14

1.2.1 Problem statement 14

1.2.2 Aim of study 14

1.2.3 Objectives and scope 14

1.3 Literature Review 15

1.3.1 Robotic arm 15

1.3.2 Wired Joystick controller 16

1.3.3 Hand gesture 17

1.3.4 Robotic car 18

1.3.5 Conclusion 20

1.4 References 21

CHAPTER TWO 23

THEORATICAL ASPECTS 23

2.0 Introduction 23

2.1 Overview of a wirelessly controlled robotic arm vehicle 23

2.1.1 Mechanical design of robotic arm 23

2.1.2 Robotic arm forces 24

2.1.3 Robotic arm inverse kinematics analysis 26

2.1.4 Mechanical design of robotic vehicle 28

2.2 Driving engines using complementary angle 29

2.3 Robotic Arm Vehicle Control 29

2.4 Prototype components description 30

2.4.1 Arduino Uno board 30

2.4.2 Technical Specifications of the Arduino Uno Board 31

2.4.3 Servo motor 32

2.4.4 Operation 32

2.4.5 DC motor 33

2.4.6 DC motor operation 33

2.4.7 L298N motor drive 34

2.5 References 36
CHAPTER THREE 37
METHODOLOGY 37
3.0 Introduction 37
3.1 Project Flowchart 37
3.2 Structure of the overall system 38
3.3 The Design 39
3.3.1 Interfacing Arduino Uno with Servos 39
3.3.2 Interfacing Arduino Uno with Play Station 2 Controller 40
3.3.3 Pinning the PS2 controller 41
3.3.4 PS2 Signals 42
3.3.5 Modifying servo for continuous rotation 42
3.3.6 Interfacing Arduino Uno with L298N Motor Drive and DC Motor 43
3.4 Programming the Arduino Uno 44
3.5 Conclusion 44
3.6 Refrences 45
CHAPTER FOUR 46
RESULTS AND ANALYSIS 46
4.0 Introduction 46
4.1 Prototype results and analysis 46
4.1.1 Results 46
4.1.2 Discussion 47
4.2 Software Development 48
4.3 Prototype Overview 48
CHAPTER 5 49
CONCLUSION 49
5.0 Introduction 49
5.1 Discussion 49
5.2 Recommendations 49
APENDIX 1 51

List of Figures

1.1 Robotic arm…………………………………………………………………….….……..3
1.2 Actuation scheme of the three finger robot hand……………………………….....……4
1.3 Actual view of arduino joystick shield module…………………………….……..…..3
1.4 System flow of hand gesture control robot……………………………………..….6
1.5 Placing of flexible sensors to hand glove…………………………………...………..6
1.6 Six wheel mechanical view……………………………………………………..……..…7
1.7 Six wheel slope climbing…………………………………………….……..…………7
1.8 Moving direction of a four wheel car……………………….………………...…….8
2.1 Mechanical design of robotic arm…………………………………………..…….11
2.2 Gripper closed……………………………………………………………….………12
2.3 Gripper open…….………...………………………………………………...…..…12
2.4 Work region of robotic arm…….….……………………………………………………12
2.5 Force diagram of robotic arm……………..………………………………......…13
2.6 Angle and parameter of robotic arm…………………………….………………......15
2.7 Omni-directional robotic vehicle conceptual views……………………………………..16
2.8 Graphical view of robotic vehicle motion……………………………………………….16
2.9 PS2 controller/ RF receiver.………………………………….................................18
2.10 Wireless joystick/ RF transmitter ……………………………………………….…….18
2.11 Arduino Uno board with parts labelled…………………………...………………….…...19
2.12 Two type of servo motor……………………...……………….………………….…20
2.13 Servo motor operations……………………………..……………….………………….……21
2.14 Dc motor……….…………………………….…………………………….………………21
2.15 DC motor operation………………………………………………………...…...………22
2.16 H-bridge……………………………………………………………………….………22
2.17 L298N motor drive………………….…………………….......................................23
2.18 Pulse width modulation………………………………………………..…………23
3.1 Project flow chart……………………………………………………….………………25
3.2 Block diagram of robotic arm vehicle ……………………………………….………27
3.3 Interfacing Arduino Uno to servos………………………..27
3.4 Interfacing Arduino Uno to PS2 controller………………………………………….….28
3.5 Pinning the PS2 controller………………………………………………………………29

3.6 Interfacing Arduino Uno to L298N motor drive and DC motor…………………………31
3.7 Modifying servo for continuous rotation……..…………………………………..……32
4.1 Robotic arm vehicle………………………………………………………..……..36

List of Tables

2.1 Values used for toque calculations…………………………………………………..13

2.2 Motors and their respective direction…………………………………………….….15

2.5 Technical specifications of the Arduino ……………………………...…...……13

4.1 Robotic arm vehicle test results……….……………………………………….29

4.2 RF link response over distance…………………………………………………...35

Dissertation layout

Chapter 1 is the introduction

Chapter 2 is the Theoretical aspects. In this chapter a review of the current wireless antenna optimization systems and all the aspects that make up the dissertation topic will be outlined.

Chapter 3 outlines the research methods used and all the steps that were taken to complete this dissertation.

Chapter 4 is the Results and Analysis section. The results obtained from the designed prototype are analyzed and compared to the expected results and a conclusion will be made based on the relationship of the two.

Chapter 5 is the conclusion and recommendations chapter.

CHAPTER ONE

INTRODUCTION

1.0 Introduction

This venture shows the execution of a robotic arm vehicle which is programmable in nature and it can be controlled [1]. For all intents and purposes, a robot is fundamentally an electro-mechanical machine that is guided by methods for a computer and electronic programming. The robotic arm is sometimes referred to as anthropomorphic as it is fundamentally the same as that of a human hand. An automated arm mounted on a moving vehicle can be utilized for different undertakings, for example, moving hardware, welding, drilling, spraying and many more. This independent automated arm vehicle is created by utilizing components like arduino Uno processor and motors. This increases its speed of operation and reduces the complexity nature. It also brings about an increase in efficiency which makes it simple to move to hazardrous materials. The fundamental part of the design is arduino Uno which facilitates and controls the product's activity. Apply autonomy includes components of mechanical and electrical engineering, and in addition control theory, processing and now artificial intelligence. As per the Robot Institute of America, "A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks".

1.1 Background

The advances in industry are glowing sharply in these ongoing decades. As glowing day by day, an ever increasing number of electronic items created are more advanced than the olden day. In the meantime, the robotic industry demand and want to deliver a robot that is ready to catch up with technologies and in the meanwhile produce more and more robots that are helpful to people in general, particularly for the high hazard work place. Because of innovations are getting higher, the advancement charges likewise increment with it. In this venture, a robotic arm vehicle intended to play out the hazardous undertakings or some other modern assignments like pick and place will be talked about. However, to deliver a robotic arm vehicle for dangerous task, it ought to be low in development cost and with high innovations incorporated. If it is considered to be manufactured for one time purpose or one time use, at that point it ought to be low in production fee as well.

1.2 Scope and objective of the research

1.2.1 Problem statement

People today do every one of the tasks involved in the manufacturing industry by them. Anyway there are a few tasks that are considered excessively risky, making it impossible to be performed by people, or a few spots like where there are higher radioactive components and dangerous materials. Likewise there is some work that should be done in hazardous ways like in electrical industry where there live cables that can cause harm, in mining or construction where falling can occur or where more strength is required to lift things, military to diffuse bombs, also in bee farming, etc. The researcher proposes that a robotic arm vehicle can be utilized. A robotic arm is a robotic manipulator, typically programmable, with comparative capacities to a human arm. The robotic arm has a couple of joints, like a human arm, in addition shoulder, elbow, and wrist. This mechanical arm can be controlled remotely and in this way helps increment adaptability at work.

1.2.2 Aim of study

To design a robotic arm vehicle that is used for dangerous tasks, or in radioactive conditions and for pick and place of products or equipments in industries. In this venture a demo model will be used to show the project idea.

1.2.3 Objectives and scope

In this project, the goal is to design, implement and built a minimal cost robotic arm vehicle for different tasks. There are numerous sub-objectives that needed to be done so as to effectively accomplish the target which are as underneath:

1. Define, design and build the structure of the robotic arm vehicle

2. Define the grasping mechanism of the robotic arm

3. Define the moving mechanism of the robotic vehicle

4. Define the control mechanism of the remote controlling for the robotic arm vehicle

5. Build the RF remote controlling for the robotic arm vehicle

By the end of this project, a mobile robotic arm vehicle is developed and it can be controlled by joystick wirelessly.

1.3 Literature Review

From the past projects, mechanical robotic arms have been created utilizing potentiometers, each motor being controlled manually by varying the potentiometer resistance [2]. Likewise some mechanical arms have been created using joystick modules which are not wireless [3]. Additionally some mechanical arms were created and controlled using Visual Studio, which is Integrated Development Environment (IDE) from Microsoft. This software can be used as a development of computer programs for the Microsoft Windows. Visual Studio is using Microsoft software development platforms such as Windows API, Windows Forms and others platforms with a specific end goal to control devices. It can create both native code and managed code. These days, most of the industrial robotic are controlled manually in the machine, anything done by the engineer cannot be considered as 100% flawless, everything is made in view of safety and health of user, it is exceptionally hazardous when user control robotic arm in the machine, it will course individuals be harmed if falling happened.

1.3.1 Robotic arm

A robotic arm using 4 Degree of Freedoms (DOF) has been developed by the researchers [4] which consist of four motors. The following diagram shows the general structure of the robotic arm.

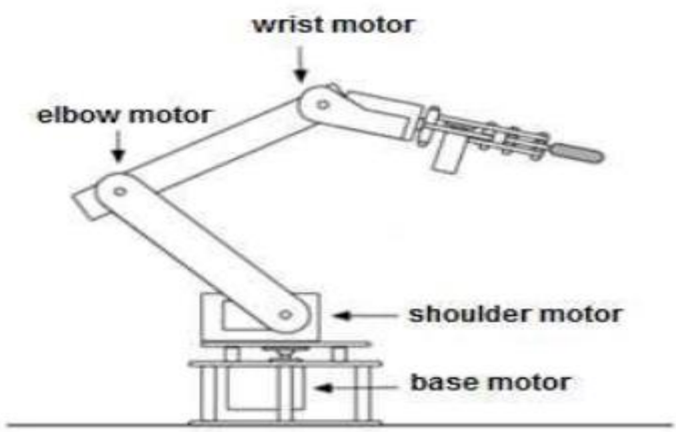


Figure 1.1 robotic arm

For a robotic hand with multiple fingers, basically the actuation structure of the fingers can be classified into two types, the under actuated structure and fully actuated [5]. For under actuated structure, the design used to coupled more than one DOF of the finger associate to a single actuator while fully actuated structure define to associate one degree of freedom to one actuator [6]. There are pros and cons for each structure. The under actuated structure is small in size since mount one actuator to associate several DOFs, lower weight, easy to develop and lower cost of design since using lesser actuator. But in terms of accuracy, since using lesser actuator to control more DOFs, the DOFs of the finger would be limited. For fully actuated structure, it is of benefit that it can accurately achieve all the DOFs of the finger since using one actuator to control one DOF. However, in another side of view, increasing in size, weight, complexity of control, development time and cost of development would restrict researchers to do any further development.

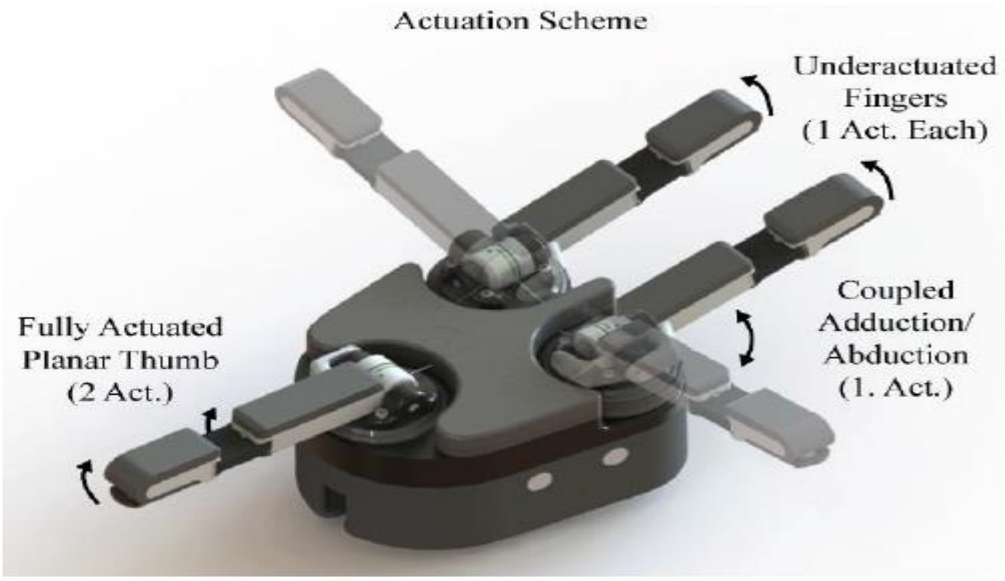


Figure 1.2 Actuation scheme of the three fingers robotic hand

1.3.2 Wired Joystick controller

To control the motion of the robotic car and robotic arm, the joystick controller is used. The joystick shield used originally is designed for Arduino UNO. It is mainly plug-and-play for Arduino UNO and no extra circuit needed. For in this case, since the development board and microcontroller are SKds40A and dsPIC30F4013 respectively instead of Arduino UNO with ATmega328P microcontroller inside, no plug-and-play for the development board of this project and extra circuitry interfacing is needed. First thing to do is to identify the original interfacing circuit between Arduino UNO and joystick shield, that means to find out the pin allocation of the joystick shield and identify the function of each pin. Diagram below shows the joystick shield module and the pin diagram of Arduino UNO interface with joystick shield.

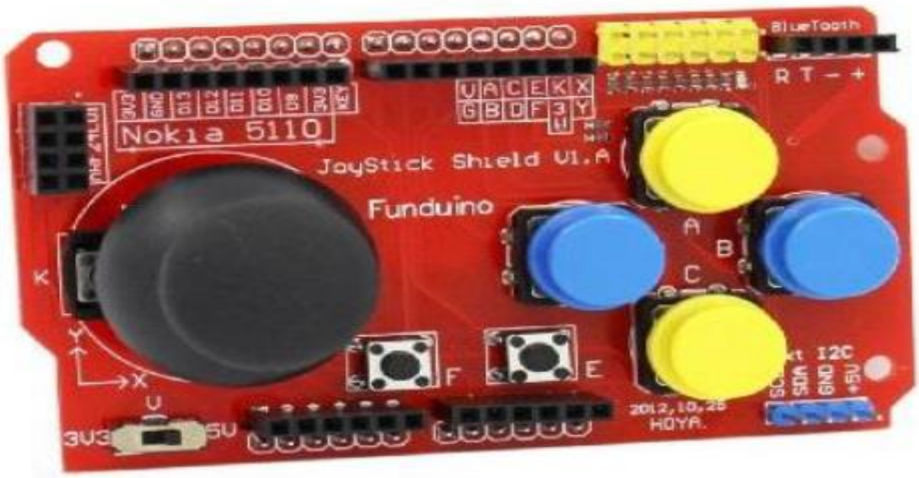


Figure 1.3 Actual view of Arduino joystick shield module

1.3.3 Hand gesture

There is a research propose using hand gesture to control the robotic vehicle [7]. The robotic car is able to moving forward, reverse, left or right by performing some hand gesture. A 2-axis accelerometer is used to get the x-axis measurement and the y-axis measurement of the human hand and transmit the data to the robotic car via RF. Thus, based on the x-axis measurement and the y-axis measurement get from the accelerometer, researcher decided to use x-axis measurement to perform left and right turn of the robotic car while y-axis measurement to perform forward and reverse of the robotic car. Diagram below shows the full system flow of the program that transforms hand gesture into action of the robotic car.

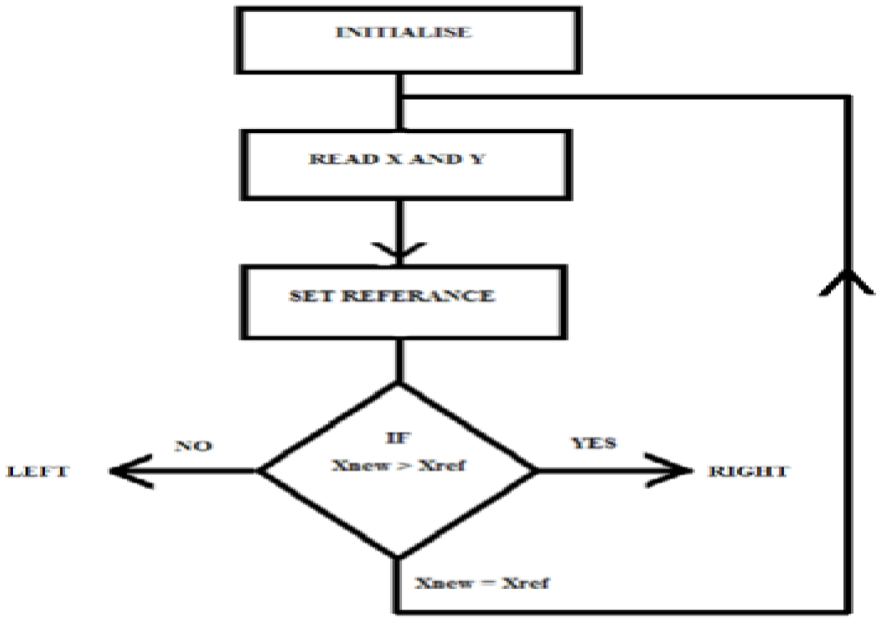


Figure 1.4 System flow of the hand gesture control program

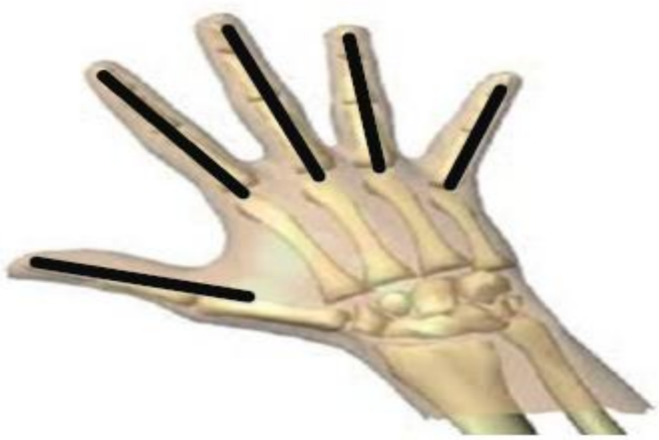


Figure 1.5 Placing of flexible sensor to hand glove

1.3.4 Robotic car

A research on robot with six-wheels has been come out by Viboon Sangveraphunsiri and Mongkol Thianwiboon [8]. This robot consists of six wheels as locomotion based with each side of three wheels. The steering mechanism, which is the way the robot turn directions is by putting 4 steering joint in the rear and front wheels. Following diagram shows the mechanical view of the six-wheel mobile robot developed by the researchers.

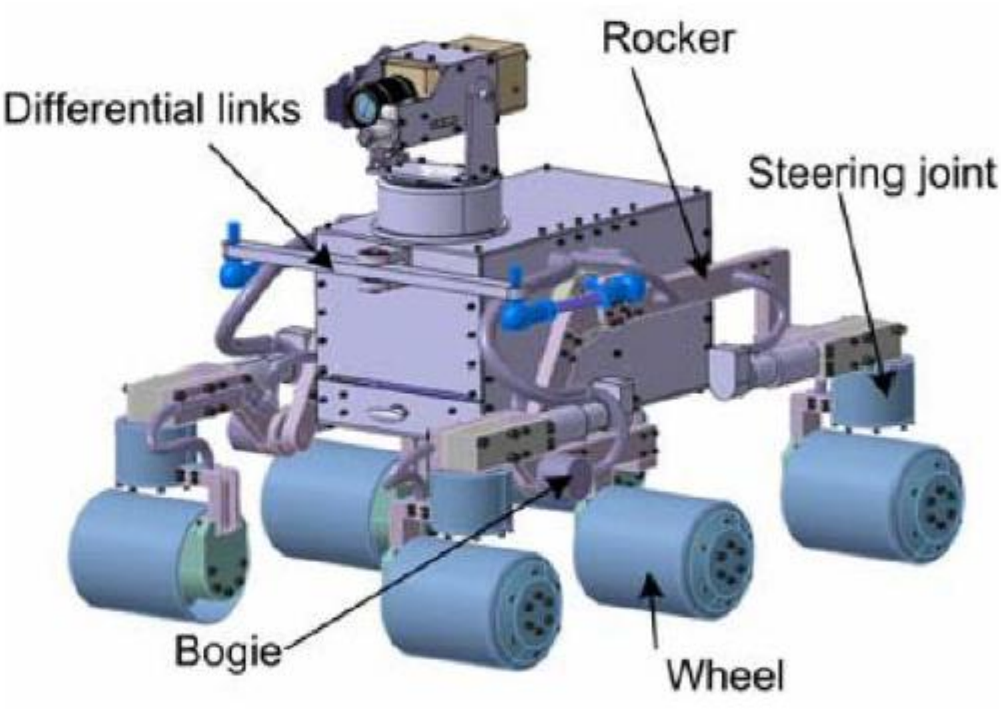


Figure 1.6 six wheel mechanical view

An exclusive view of this design is that it is equipped with suspension in which to allow one wheel of the robot can be lifted vertically while other wheels remain in contact with the ground [9]. This is very useful when mobile robot run through an extreme surface area, for example climbing up a slope, traversing over a ditch, traversing over obstacles, climbing upstairs and etc. Following diagrams show the view on how the six-wheel robot performs such action as stated.

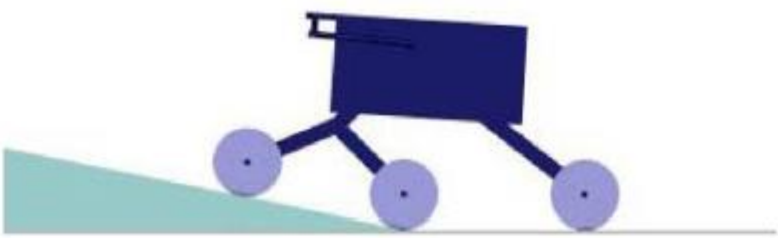


Figure 1.7 six wheel slope climbing

The diagram below shows the movements of a four wheel car, to turn right, wheel 1 and wheel 2 dc motor are selected to rotate forward meanwhile wheel 3 and wheel 4 dc motor rotate backward. Whereas to turn left, wheel 3 and wheel 4 dc motor are selected to rotate forward and wheel 1 and wheel 2 dc motor rotate backward. In mechanical view, the robotic car is now able to move in 4 directions, i.e. forward, reverse, left and right. In order to make the robotic car more flexible to control, it should be design with 8 directions moving, i.e. forward, reverse, left, right, forward-left, forward-right, reverse-left and reverse-right. The existing mechanical part of the robotic car is completely satisfied for all 8 directions and diagram below shows how the basic concept for the robotic car to move to all 8 directions.

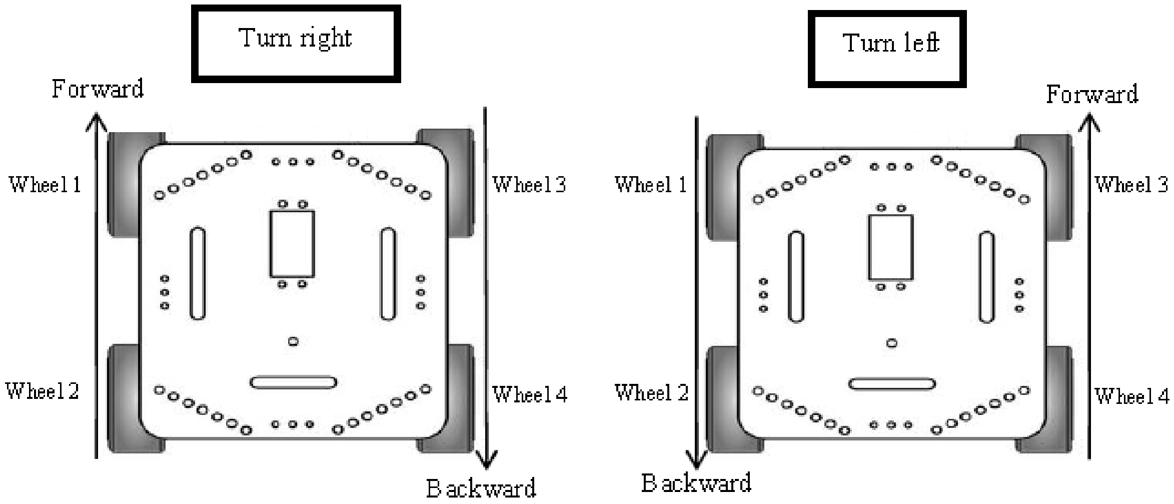


Figure 1.8 moving directions of a four wheel car

1.3.5 Conclusion

In light of the robotic arm review the developer chose to build a robotic arm mounted on a moving vehicle and both the arm and vehicle are controlled remotely using the wireless Play Station 2 (joystick) with a 2.4GHz link between the transmitter and the receiver.

1.4 References

[1] Anthony L. Crawford, Jeffrey Molitor, Alba Perez-Gracia, Steve C. Chiu Design of a Robotic Hand and Simple EMG Input Controller with a Biologically-Inspired Parallel Actuation System for Prosthetic Applications Available: http://digital.csic.es/bitstream/10261/97426/1/Design%20of%20a%20robotic%20hand.pdf Last accessed: 27th April 2018

[2] J. C. Trinkle and R. James Milgram, Complete Path Planning for Closed Kinematics Chains with Potentiometers; SAGE International Journal of Robotic Research 21, 9, 773-789, 2013.

[3] Shamsheer Verma (2013) Joystick Controlled Robotic Arm Available: http://www.ripublication.com/aeee/80_pp%20%20%20%20601-606.pdf Last accessed: 27th April 2018

[4] Ankit Gupta, Mridul Gupta, Neelakshi Bajpai, Pooja Gupta, Prashant Singh (2013) Efficient Design and Implementation of 4-Degree of Freedom Robotic Arm Available: http://www.ijeat.org/attachments/File/v2i5/E1720062513.pdf Last accessed: 27th April 2018

[5] Raphael Deimel, Oliver Brock (2013) A Compliant Hand Based on a Novel Pneumatic Actuator Available: https://www.robotics.tu-berlin.de/fileadmin/fg170/Publikationen_pdf/2013-icra13_Deimel_Brock.pdf Last accessed: 27th April 2018

[6] M. Gemeinder and M. Gerke, GA-based Path Planning for Mobile Robot Systems employing an active Search Algorithm; Journal of Applied Soft Computing 3, 2, 149-158, 2015.

[7] Riyaz Mansuri, Sandesh Vakale, Ashish Shinde, Tanveer Patel (2013) Hand Gesture ControlRobotVehicleAvailable:http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5EB9FCE0F8A1B119E8DF5CDB63919310?doi=10.1.1.416.8918&rep=rep1&type=pdf Last accessed: 27th April 2018

[8] Viboon Sangveraphunsiri, Mongkol Thianwiboon Traction Control of a Six-wheel Robot using Wheel Slip Dynamics Available: http://www.regional-robotics.org/lab_info/Paper/Journal/Traction%20Control%20of%20a%20Six-

wheel%20Robot%20using%20Wheel%20Slip%20Dynamics.pdf Last accessed: 27th April 2018

[9] F. Ribeiro, I. Moutinho, P. Silva, C. Fraga, N. Pereira Three Omni-directional Wheels Control On A Mobile Robot Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.1083&rep=rep1&type=pdf Last accessed: 27th April 2018

CHAPTER TWO

THEORATICAL ASPECTS

2.0 Introduction

In this chapter, the researcher gives the necessary theoretical background in line with the robotic arm vehicle overview. Diagrams and tables have been included to clarify various important aspects. Also the components used in the project were discussed.

2.1 Overview of a wirelessly controlled robotic arm vehicle

2.1.1 Mechanical design of robotic arm

The mechanical design of the robot arm is based on a robot manipulator with similar functions to a human arm. The links of such a manipulator are connected by joints allowing rotational motion and the links of the manipulator is considered to form a kinematic chain [1]. The business end of the kinematic chain of the manipulator is called the end effector or end of arm tooling and it is analogous to the human hand. Figure 2.1 shows the Free Body Diagram for mechanical design of the robotic arm.

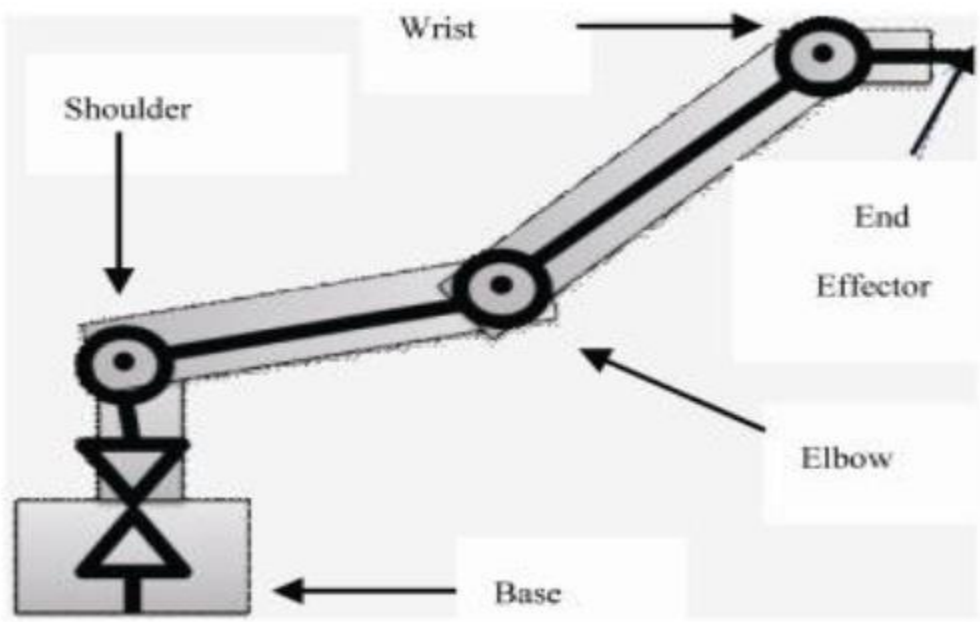


Figure 2.1 Mechanical design of robotic arm

As shown, the end effector is not included in the design because a commercially available gripper is used. This is because that the end effector is one of the most complex parts of the system and, in turn, it is much easier and economical to use a commercial one than build it.

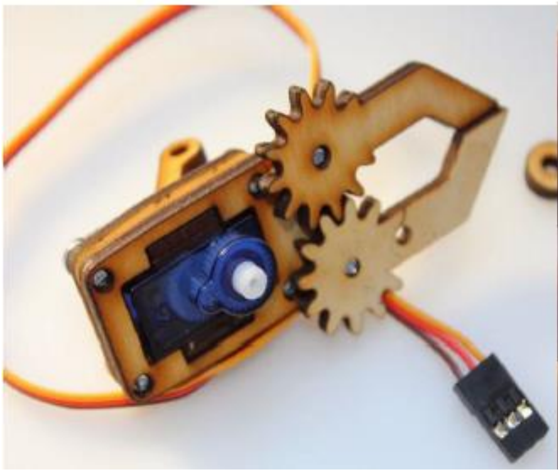


Figure 2.2 gripper closed

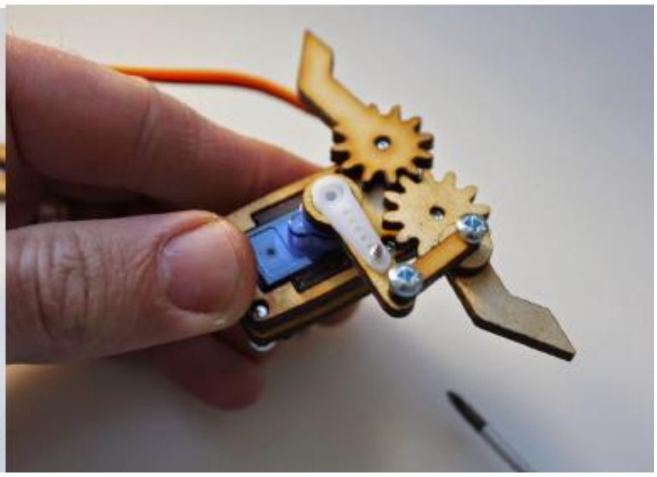


Figure 2.3 gripper Open

The end effector is probably one of the most important and most complex parts of the system. The end effector varies mainly according to the application and the task that the robot arm accomplishes for; it can be pneumatic, electric or hydraulic. Since the robot arm is based on an electric system, an electric basis end effector may be chosen. Besides, the main application of the system is handling, accordingly, the recommended type of our end effector is a gripper, as shown in Figurer 2.2 and Figurer 2.3. The end effector is controlled by a servo motor.

2.1.2 Robotic arm forces

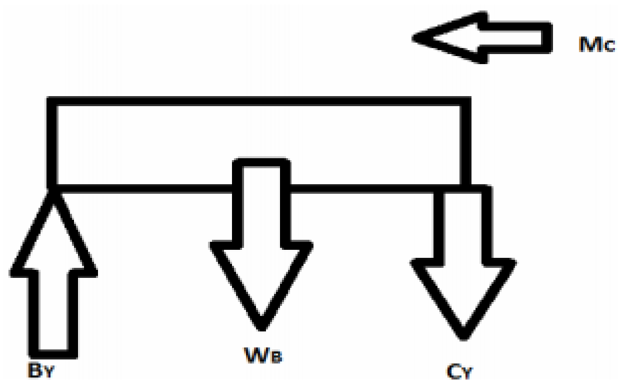


Figure 2.4 work region of robotic arm

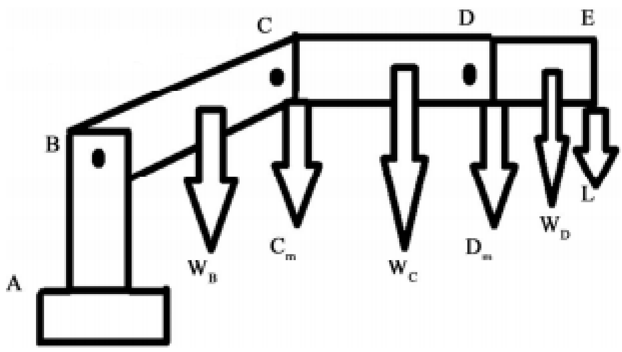


Figure 2.5 force diagram of robotic arm

Figure 2.4 shows the force diagram used for load calculations. The calculations are carried out only for the joints that have the largest loads, since the other joints would have the same motor that is the motor can move the links without problems [2]. The calculations considered the weight of the motors, about 50 grams, except for the weight of motor at joint B, since it is carried out by link BA. Figure 2.5 shows the force diagram on link CB, which contains the joints (B and C) with the highest load (carry the links DC and ED) and the calculations are carried out as follows.

Table below shows the values used for the torque calculations for load L = 1kg (load) and weight of servo Cm = dm = 0.050 kg

Weight	Length
WD = 0.020 kg (weight of link DE)	LBC = 0.14 m (length of link BC)
WC = 0.030 kg (weight of link CD)	LCD = 0.14 m (length of link CD
WB = 0.030 kg (weight of link CB)	LDE = 0.05 m (length of link DE)

Table 2.1 values used for toque calculations

Performing the sum of forces in the Y axis, using the loads as shown in Figure 2.4, and solving for CY and CB, see Equations (1).(4). Similarly, performing the sum of moments around point C, Equation (5), and point B, Equation (6), to obtain the torque e in C and B, Equations (7) and (8), respectively.

$$\sum Fy = \left(L + W_D + D_M + W_C + C_M\right) g - C_Y = 0 (1)$$
$$C_Y = (1.141kg) 9.8m/s^2 = 11.18N (2)$$
$$\sum F_{Y=} \left(L + W_D + D_M + W_C + C_M + W_B\right) g - C_B = 0 (3)$$
$$C_B = (1.171kg) 9.8m/s^2 = 11.4758N (4)$$
$$\sum M_C = -\left(\frac{W_C L_{CD}}{2}\right) - W_D\left(L_{CD} + \frac{L_{DE}}{2}\right)$$
$$-L\left(L_{CD} + L_{DE}\right) - D_M\left(L_{CD}\right) + M_C = 0 (5)$$
$$\sum M_B = -L\left(L_{BC} + L_{CD} + L_{DE}\right) - W_D\left(L_{BC} + L_{CD} + \frac{L_{DE}}{2}\right)$$
$$-D_M\left(L_{BC} + L_{CD}\right) - W_C\left(L_{BC} + \frac{L_{CD}}{2}\right)$$
$$-C_M\left(L_{BC}\right) - W_B\left(\frac{L_{BC}}{2}\right) + M_B = 0 (6)$$
$$M_C = 1.968Nm = 278.6\sigma z / in (7)$$
$$M_B = 3.554Nm = 503.38\sigma z / in (8)$$

The servo motors that was used, based on the calculations, is small servo with torque 1.8kg/cm and big servo with torque 10kg/cm

These motors were recommended because it is much cheaper than any other motor with same specifications. Since we need more torque at joint B, see Equation (8), we used two motors at point B to comply with the torque requirements; however, one motor is enough for the other joints. Using two motors at joint B is much cheaper than using one big motor with 10kg/cm. Other relevant characteristics of the motors, which can be shown in Figure 2.24, are that they can turn 60 degrees in 130 milliseconds and they have a weight of 47.9 grams each. Once the initial dimensions for the robot arm and the motor are defined, the design is carried out [3].

2.1.3 Robotic arm inverse kinematics analysis

The researcher mainly concern on the inverse kinematic analysis for the robotic arm design [3]. The researcher is able to get the angle of the wrist motor, elbow motor, shoulder motor and the base motor of the robotic arm based on a given coordinate. The following diagram shows the parameter used to define the robotic arm and the mathematical steps define by the researcher using the parameter shown in the diagram to get the respective angles of the motor.

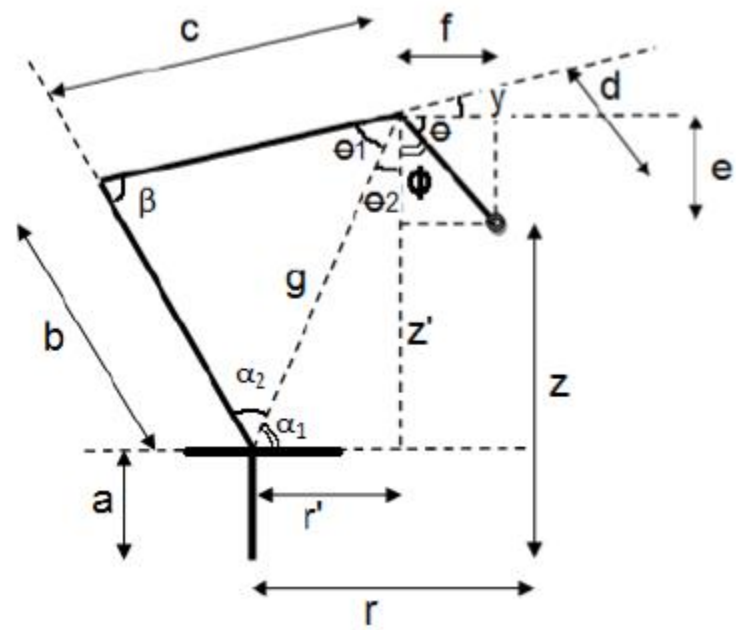


Figure 2.6 Angle and parameter of the robotic arm

1) Given (*x*, *y*, *z*) coordinate of end effector.

2) Let base length = *a*, shoulder length = *b*, arm length = *c*, gripper length = *d*, gripper angle = θ and radial length = *r*.

3) Base angle = $\tan^{-1}\left(\frac{y}{x}\right)$

4) $r=\sqrt{x^2+y^2}, e=dx\sin(\theta), f=dx\cos(\theta), r'=r-dx\cos(\theta), z'=z+dx\sin(\theta)-a$

5) $g=\sqrt{\left[(r')^2+(z')^2\right]}$

6) $\alpha_1=\cos^{-1}\left\{\left[g^2+(r')^2-(z')^2\right]/\left(2*g*r'\right)\right\}=\cos^{-1}\left\{\left[g^2+b^2-c^2\right]/\left(2*g*b\right)\right\}$

7) Shoulder angle = $\alpha_1+\alpha_2$

8) $\beta=\cos^{-1}\left\{\left[c^2+b^2-g^2\right]/(2*c*b)\right\}$

9) Arm angle = $180-\beta$

10) $\theta_1=180-\left(\beta+\alpha_2\right), \theta_2=180-\left(\alpha_1+90\right), \phi=90-\theta$

11) Wrist angle = $180-\left(\theta_1+\theta_2+\phi\right)$

2.1.4 Mechanical design of robotic vehicle

A research on three Omni-directional wheels mobile robot has been done. The physical structure consists of 3 wheels and the wheels are coupled with one motor each and are tied to a single centre [4]. Since all the wheels coupled with motor are tied to a single centre, each wheel will have 120 degree distance with the other 2 wheels, assume that the length of end to end from wheel to motor are same among 3 wheels, it is forming an equilateral triangle as shown in below diagram.

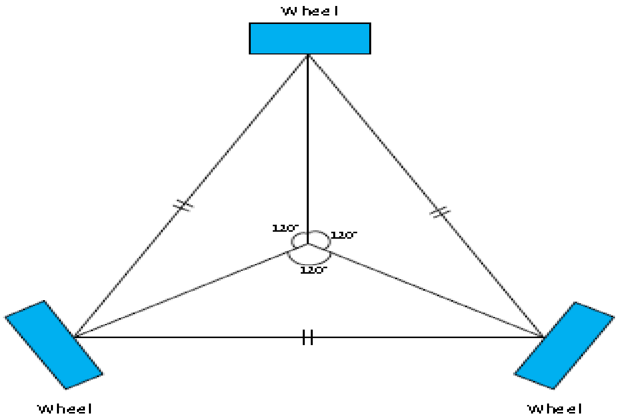


Figure 2.7 Omni-directional robotic vehicle conceptual views

To control a movement to a desire direction, three wheels are rotated with specific velocity in order to allow the mobile robot towards the direction wanted. From the research paper, with an example the mobile robot desire to go to α degree direction and the following diagram shows the graphical description of control the velocity of each wheel in order for the mobile robot moving to degree direction [5].

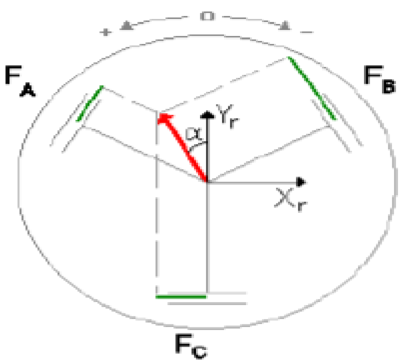


Figure 2.8 Graphical view of robotic vehicle motion

The green line shown in the diagram above indicate the length of velocity, the longer line the, higher velocity while the shorter line, the lower velocity. With 3 motor denoted as FA, FB and FC and velocity as linear velocity of the robot, following formula shows how researcher identify the length of velocity of each wheel [6].

Table below shows the motors and their respective velocity.

Motor	Velocity
F A	velocity · cos (150 − DesiredDirection)
F B	velocity · cos (30 − DesiredDirection)
F C	velocity · cos (270 − DesiredDirection)

Table 2.2 motors and their respective direction

From this research, a series of calculation need to be taken out before any of the development and thus, it is not very suitable for a short period of development time as I plan to do so in this project.

2.2 Driving engines using complementary angle

This operation is used to control tow servo motors in opposite direction for each other's. The small servo G9 used have a rotation limit 180 degree so the command for first servo motor is to move to angle X and command the opposite servo motor to move to angle 180.X then the tow motors well move in Synchronous movement that meets the needs [7].

2.3 Robotic Arm Vehicle Control

The robot arms can be autonomous or controlled manually. In manual mode, a trained operator (programmer) typically uses a portable control device (a teach pendant) to teach a robot to do its tasks manually [8]. Anyway in this project a wireless handle is used, that is a Play Station 2 controller (joystick). The PS2 wireless controller is a standard controller for the PlayStation 2 and is identical to the first Dual Shock controller for the PlayStation console. It features twelve analog (pressure sensitive) buttons (X, O, Π, Δ, L1, R1, L2, R2, Up, Down, Left and Right), three digital buttons (Start, Select and the analog mode button) and two analog sticks. The controller likewise includes two vibration motors, the left one being bigger and more effective than the one on the right. It is powered by two AAA batteries.

Figure 2.9 PS2 controller/ RF receiver

Figure 2.10 wireless joystick/ RF transmitter

The PS2 controller goes about as the RF (Radio Frequency collector), it gets orders from the remote joystick that is RF transmitter and send them to arduino board where it is associated with control engines. The transmitter and beneficiary impart through a 2.4GHz RF convention.

2.4 Prototype components description

2.4.1 Arduino Uno board

It is a microcontroller board based on the ATmega 328P. It has 14 digital pins, 6 analogues Inputs, a 16MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset Button. Fig 2-5 shows a typical Arduino Uno board. One has to simply connect it to a Computer with a USB cable or power it with an AC-to-DC adapter or battery to get started [9].

2.4.2 Technical Specifications of the Arduino Uno Board

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7V-12V
Input Voltage (limits)	6V-20V
Digital I/O Pins	14 (of which six provide PWM output)
Analogue Input Pins	6
DC Current per I/O pin	40Ma
DC current for 3.3V pin	50mA
Flash Memory	32KB of which 0.5KB is used by boot loader
SRAM	2KB
EEPROM	1KB
Clock Speed	16

Table 2.5 Technical Specs of the Arduino board

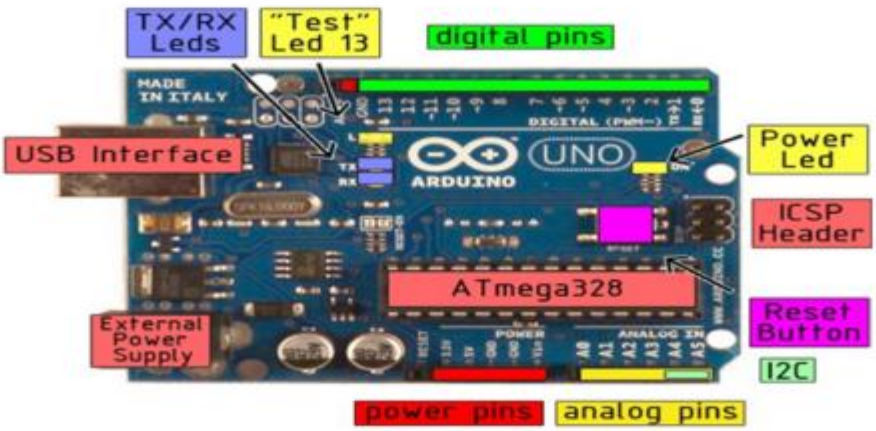


Figure 2.11 Arduino Uno board with parts labelled

This is the type of the Arduino microcontroller board which is going to be used for this robotic arm vehicle prototype. It can output two levels of power 3V and 5V.

2.4.3 Servo motor

Figure 2.12 Two Type of Servo Motor

The servo motors are motors that allow for precise control. They consist of a motor coupled to a sensor for position feedback, through a reduction gearbox [10]. As a result, servo motors are used to control position of objects, rotate objects, move legs, arms or hands of robots and move sensors with high precision. The servo motor serves as an actuator in a robotics system. They are small in size. They have built-in circuitry to control their movement therefore they can be connected directly to an Arduino Uno processor.

2.4.4 Operation

The principle of operation is quite simple. The position of the servo motor is set by the length of a pulse. The servo expects to receive a signal pulse roughly every 2.0 milliseconds. If a pulse is high for 1 millisecond, then the servo angle will be zero, there is no rotation. If the pulse is 1.5 milliseconds, then it will be at its centre position, the servo rotates through 90 degrees. If it is 2 milliseconds it will be at 180 degrees. Thus a servo motor can be programmed to rotate to a specific desired position.

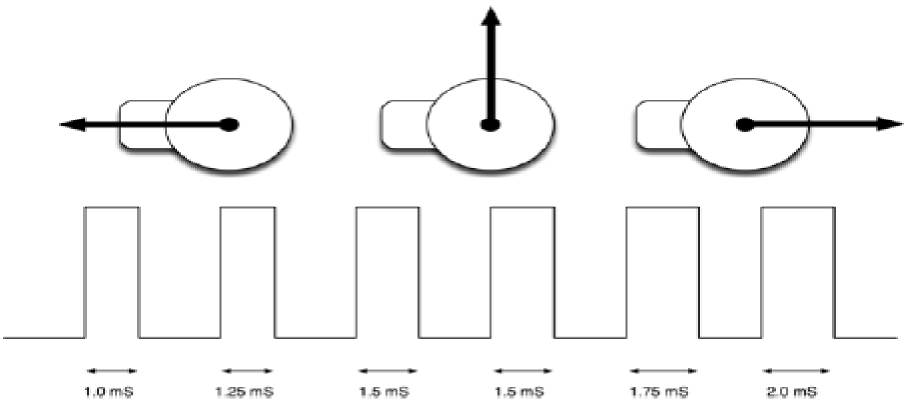


Figure 2.13 Servo motor operations

2.4.5 DC motor

DC motors are motors that operate on direct current. The first motor was invented in 1827 by Anyos Jedlik and practical motor in 1832 by William Sturgeon. DC motors are available in several different configurations from tiny little motors to large motors. DC motors can be used in robot bases, Quad copters, and also in model planes and boats.

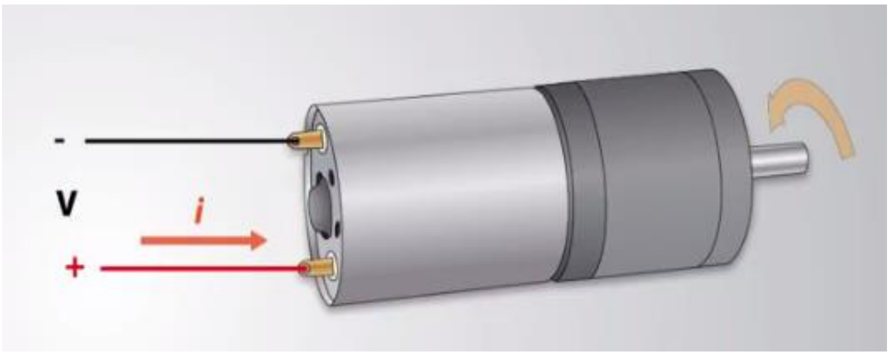


Figure 2.14 dc motor

2.4.6 DC motor operation

DC motor consists of motor shaft which is the Armature, the Armature has coils of wire and the coils are connected to the commutator, these connections are called brushes. Also outside of motor has permanent magnets and when DC current is applied to commutator magnetic field is formed in coils, coil magnets interact with permanent magnets and as armature rotates polarity is continually reversed causing the magnetic field to be reversed and the rotation continues. This is a Brushed motor.

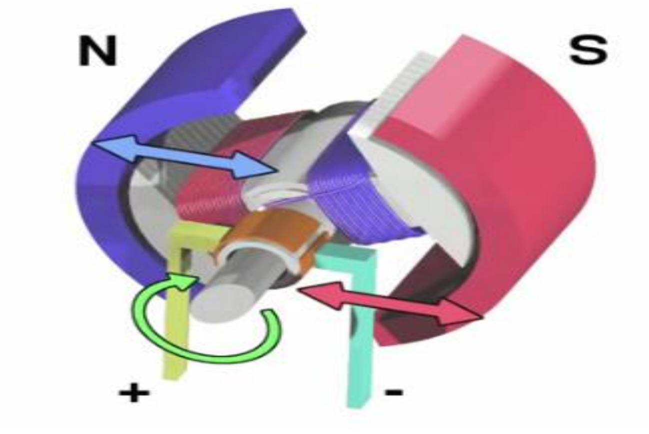


Figure 2.15 dc motor operations

2.4.7 L298N motor drive

It is based on the H Bridge circuit, this module can be used to control dc motors. An H Bridge consists of two voltages that are positive and negative voltages. When the positive is connected to the positive side of the motor and the negative to negative side of the motor, the motor rotates clockwise. When a negative voltage is applied to the other side of the motor and positive applied to the opposite side of the motor the motor rotates counter-clockwise. So to change motor direction the direction of current is just inversed using the H Bridge circuit.

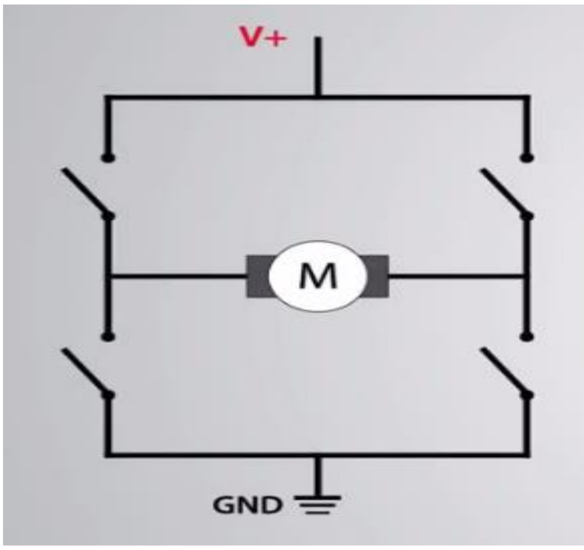


Figure 2.16 H Bridge

The L298N allows speed and direction control of motor at the same time. Also can drive motors with voltages of 5V-35V with peak current of 2A.

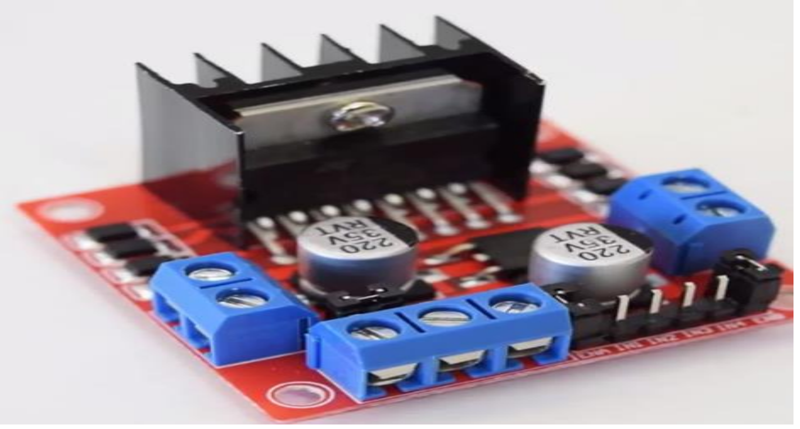


Figure 2.17 L298N motor drive

The speed of the motor can be controlled by changing the voltage applied to the motor and this can be achieved by using the PWM (Pulse Width Modulation) signal. PWM is a technique that allows to adjust the average value of voltage applied to an electrical device by turning the power on or off at a faster rate. The average voltage depends on the duty cycle or the amount of time the signal is on versus the amount of time it is off in a single period of time.

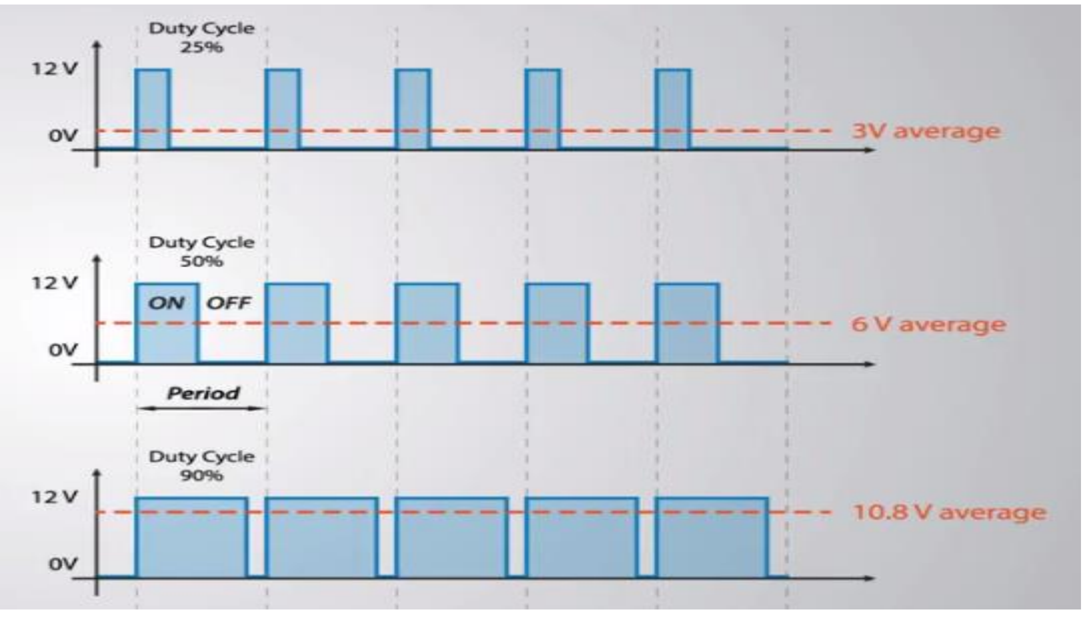


Figure 2.18 Pulse Width Modulations

2.5 References

[1] Jegede Olawale, Awodele Oludele, Ajayi Ayodele, Development of a Microcontroller Based Robotic Arm, in Proceedings of the 2016 Computer Science and IT Education Conference pg: 549-557.

[2] Lael U. Odhner, Leif P. Jentoft, Mark R. Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R.Ma1, Martin Buehler, Robert Kohout,Robert D. Howe, Aaron M. Dollar (2013) A Compliant, Underactuated Hand for Robust Manipulation Available: http://arxiv.org/ftp/arxiv/papers/1301/1301.4394.pdf Last accessed: 30th April 2018

[3] J. C. Trinkle and R. James Milgram, Complete Path Planning for Closed Kinematics Chains with Spherical Joints; SAGE International Journal of Robotic Research 2015.

[4] M. Gemeinder and M. Gerke, GA-based Path Planning for Mobile Robot Systems employing an active Search Algorithm; Journal of Applied Soft Computing 2014.

[5] Brian Bonafilia, Nicklas Gustafsson, Per Nyman, Sebastian Nilsson Self-balancing Two-wheeled Robot Available: http://sebastiannilsson.com/wp-content/uploads/2013/05/Self-balancing-two-wheeled-robot-report.pdf Last accessed: 30th April 2018

[6] M. Gemeinder and M. Gerke, GA-based Path Planning for Mobile Robot Systems employing an active Search Algorithm; Journal of Applied Soft Computing 3, 2, 149-158, 2015

[7] Devendra P. Garg and Manish Kumar, Optimization Techniques applied to multiple manipulators for path planning and torque minimization; Engineering Applications of Artificial Intelligence 2016.

[8] R.B. Gillespie, J. E. Colgate, M. A. Peshkin, A general framework for robot control; IEEE Transactions on Robotics and Automation 2015

[9] Servomotors & system design components, www.bhashaelectronics.Com

[10] Six-servo Robot Arm from www.arexx.com.cn on 13- 11-2016.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This section will give a detailed investigation of each of the subsystem that leads to the best possible working of the model. It gives an account on the design of all the subsystems and how these parts are interfaced with each other that are the central processing unit, the software that was used as a part of the project and how the system was appropriately implemented, a flow chart showing the execution of the and furthermore the block diagram of the overall system.

3.1 Project Flowchart

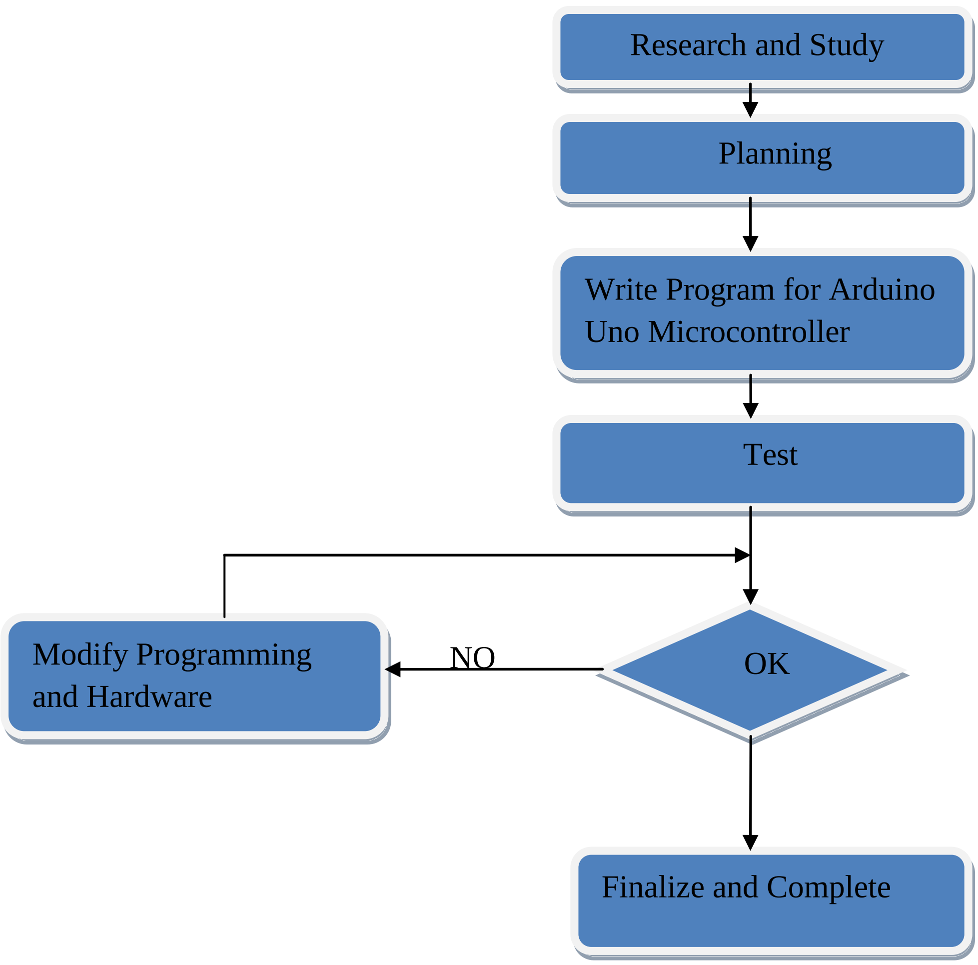


Figure 3.1 Project Flow Chart

3.2 Structure of the overall system

The overall system consists of the following hardware components

i. Servo motors

ii. DC motor

iii. Play Station 2 controller

iv. Wireless Joystick

v. Arduino Uno Board

vi. L298N motor driver

The Arduino Uno Microcontroller is the central processing unit which is responsible for controlling all the processes and is interfaced with three main subsystems which include:

The Motion Unit- consist of the servo motors which control the motion of the robotic arm that is the base, shoulder, elbow and gripper. Also one servo motor is used to control direction of the vehicle. It also consists of a DC motor that is interfaced with L298N motor driver that control the motion of the robotic vehicle.

Transmitter and Receiver Unit- consists of the play station 2 controller which accepts command from the wireless joystick and send commands to the microcontroller to control servos and dc motor.

Control Unit- consists of a microcontroller which controls the DC motor and servo motors.

All units are all interfaced with the microcontroller which acts as the central processing unit. The microcontroller is then interfaced with the controlling devices which perform the intended actuation. The structure of the system is as shown below.

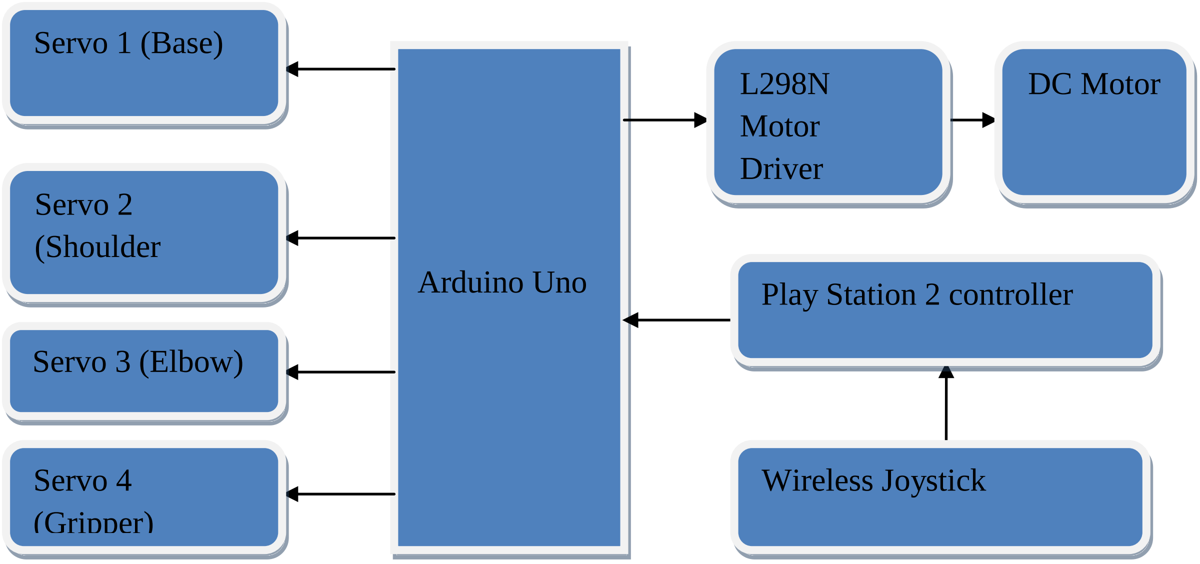


Figure 3.2 Block Diagram of Robotic Arm Vehicle

Figure 3.2 shows the robotic arm vehicle control block diagram responsible for controlling the robotic arm vehicle. All servo motors, play station 2 controller and L298N DC motor driver are connected to the central processing unit (Arduino Uno).

3.3 The Design

The design shows the schematic diagrams and the connections between the components and their microcontroller that is the Arduino Uno. It also shows the pin connections and their explanations.

3.3.1 Interfacing Arduino Uno with Servos

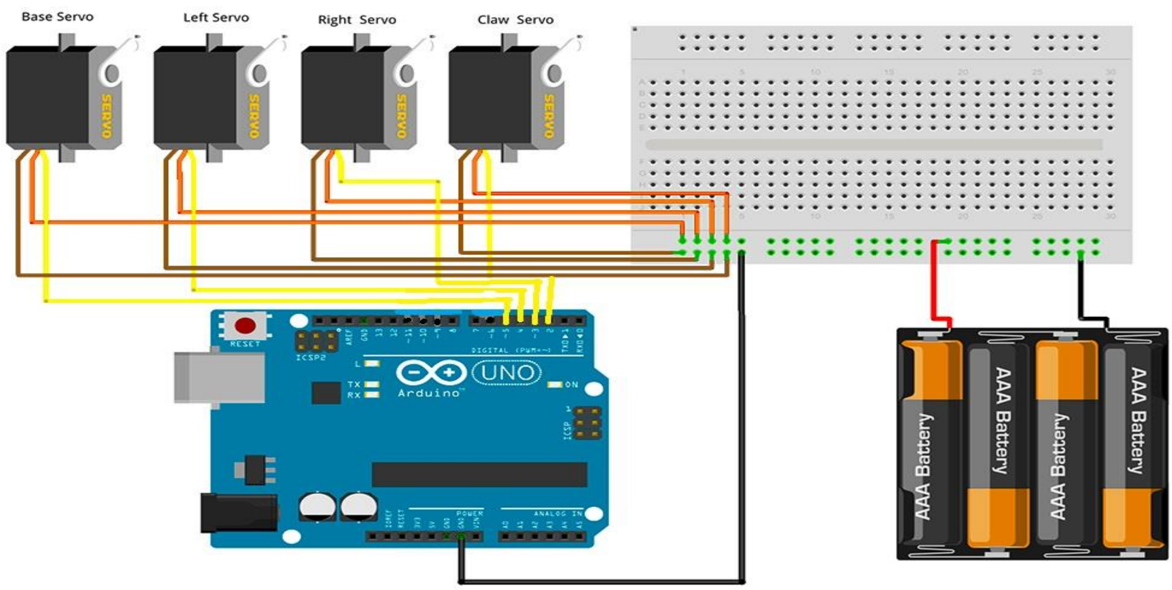


Figure 3.3 circuit diagram for interfacing arduino with servos

1. The base servo was connected to arduino pin number 4 and uses the LX (Left joystick stick in the x-axis).
2. The shoulder/ left servo were connected to arduino pin number 5 and use the RY (Right joystick stick in the y-axis).
3. The elbow servo/ right were connected to arduino pin number 3 and use the RX (Right stick joystick in the x-axis).
4. The gripper servo was connected to arduino pin number 2 and uses the L1 and L2 controls of the joystick.

3.3.2 Interfacing Arduino Uno with Play Station 2 Controller

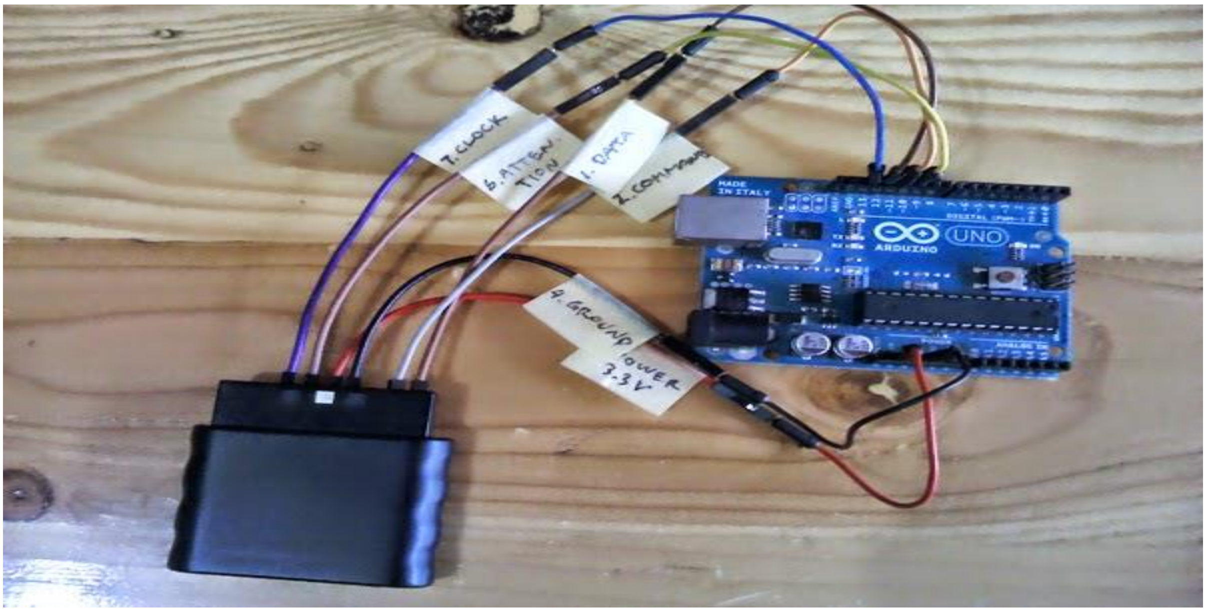


Figure 3.4 interfacing PS2 with arduino

Figure 3.4 interfacing PS2 with arduino

1. The PS2 data signal pin was connected to arduino digital pin 11
2. The PS2 clock signal pin was connected to arduino digital pin 12
3. The PS2 attention signal pin was connected to arduino digital pin 9
4. The PS2 command signal pin was connected to arduino digital pin 10

3.3.3 Pinning the PS2 controller

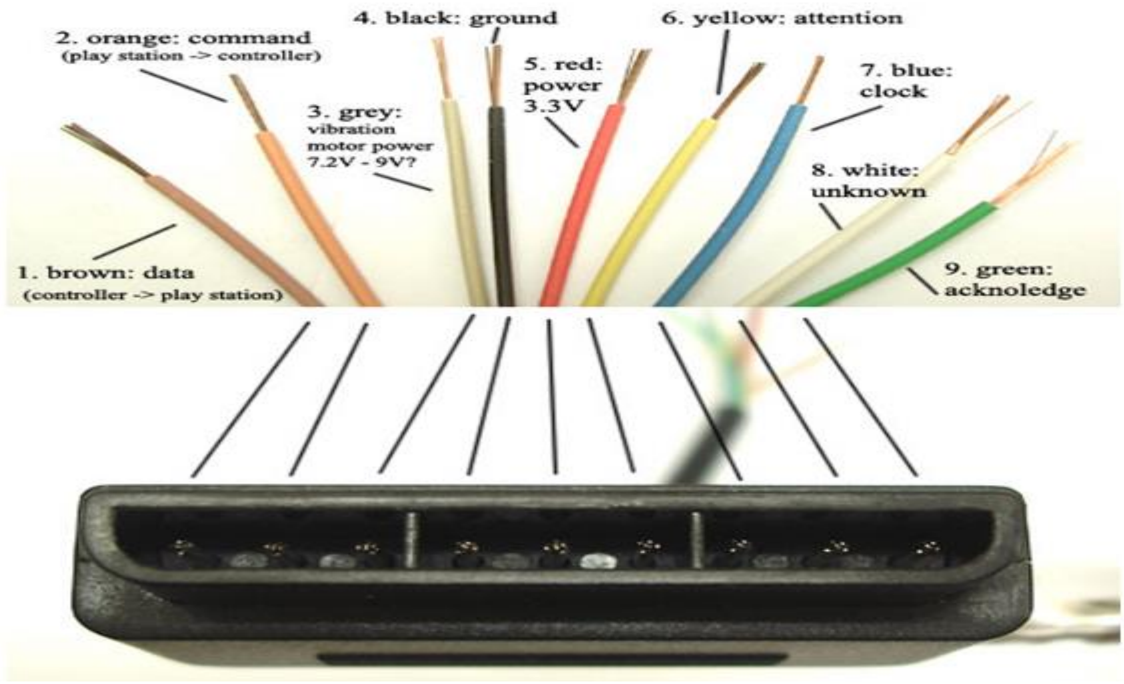


Figure 3.5 pinning PS2

1. DATA: This is the data line from Controller to PS2.

2. COMMAND: This is the data line from PS2 to Controller.

3. VIBRATION MOTOR POWER

4. GND: Ground

5. VCC: VCC can shift from 5V down to 3V.

6. ATTENTION: ATT is used to get the attention of the controller. This line must be pulled low before each group of bytes is sent/ received, and after that set high again subsequently. This pin consider as "Chip Select" or "Slave Select" line that is used to address different controllers on the same bus.

7. CLK: 500kH/z, normally high on. The communications appears to be SPI bus.

8. Not Connected

9. ACK: Acknowledge signal from Controller to PS2. This regularly high line drops low around 12us after each byte for half a clock cycle, however not after the last bit in a set.

3.3.4 PS2 Signals

PS2 remote controller speaks with Arduino utilizing a protocol that is fundamentally SPI. The play station sends a byte at the same time as it receivers one (full duplex) through serial communication. There's a clock (SCK) to synchronize bits of data across two channels: DATA and CMD. Furthermore, there's an "Attention" (ATT) channel which tells the slave whether or not it is "active" and should tune in to data bits running over the CMD channel, or send data bits over the DATA channel (Reasonably, one of the slave device should be active at a time). The PlayStation 2 actually uses this plus an additional line that is not specifically part of the SPI protocol – an "Acknowledge" (ACK) line.

The clock is held high until the point that a byte is to be sent. It at that point drops low (active low) to begin 8 cycles amid which data is at the same time sent and received. The logic level on the data lines is changed by the transmitting device on the falling edge of clock. This is then read by the receiving device on the rising edge enabling time for the signal to settle. After each Command is received from the controller, that controller needs to pull ACK low for at least one clock cycle. If a selected controller does not ACK the PS2 will assume that there is no controller present. LSBs (least significant bits) are transmitting first.

3.3.5 Modifying servo for continuous rotation

Because the dc motors I wanted to use have no gears for them to drive a large robotic arm vehicle load. I modify the servo for continuous rotation since it has gears, this was done by disabling all the electronics in the servo circuit and connected the two wires to the dc motor in the servo and the control pin was also not disconnected. The potentiometer in the servo was disconnected since it can only turn 180 degrees.

3.3.6 Interfacing Arduino Uno with L298N Motor Drive and DC Motor

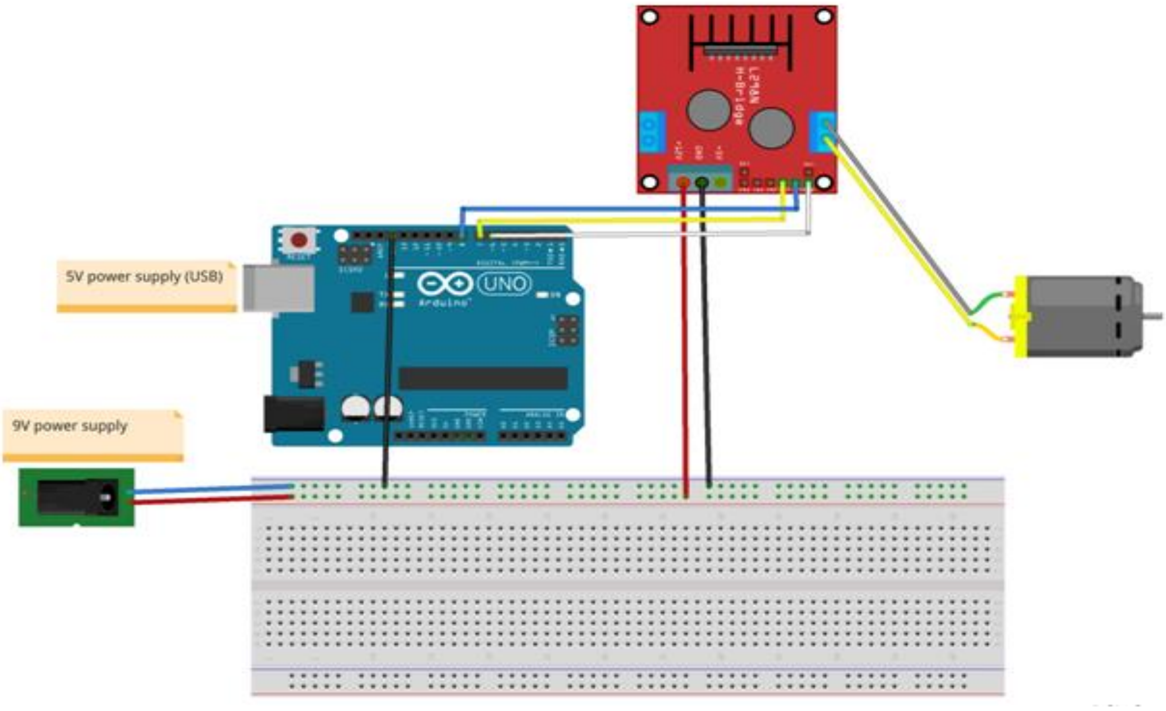


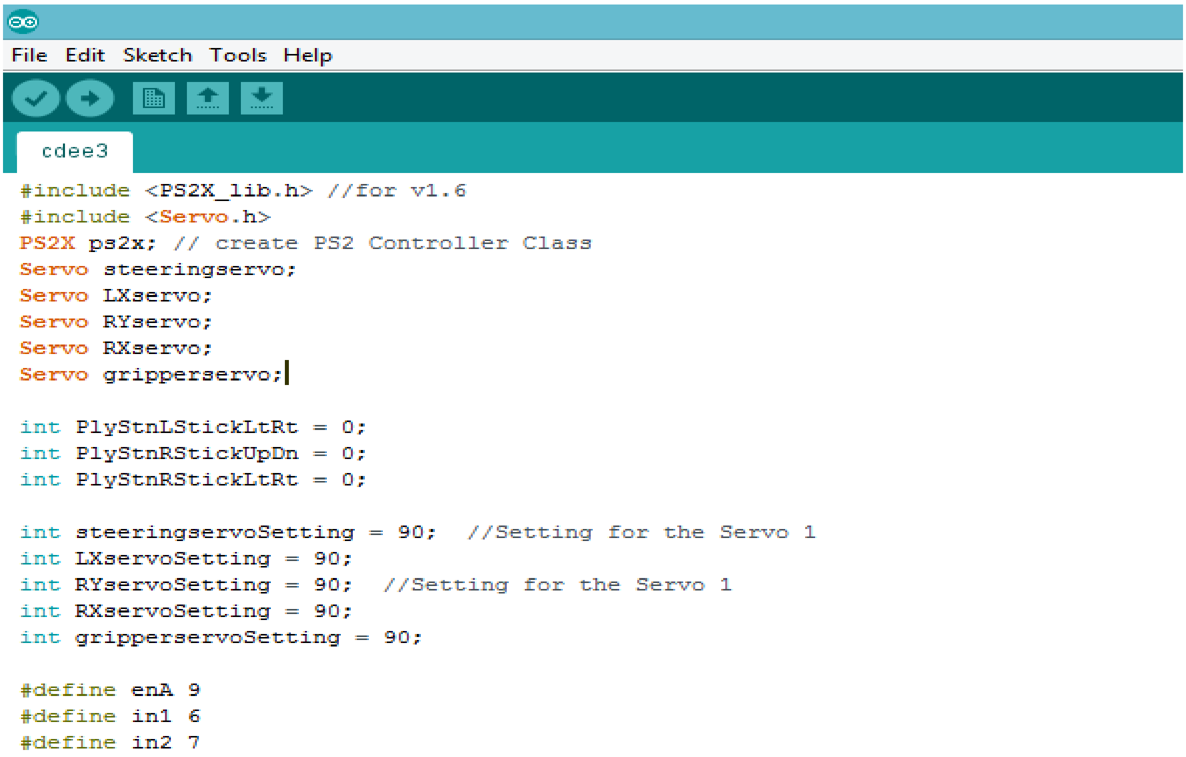
Figure 3.6 interfacing Arduino Uno to L298N motor drive and DC motor

1. ENA pin of the L298N motor drive was connected to pin 6 of arduino
2. IN1 of motor drive was connected to pin 7 of arduino
3. IN2 of motor drive was connected to pin 8 of arduino
4. 9 volt power supply was connected to L298N 12 volts pin

ENA controls the speed of the motor, when it is high the motor will move at full speed by connecting the pin to the arduino we can control the speed of the motor. IN1 and IN2 are used to control the direction of the motor.

3.4 Programming the Arduino Uno

The IDE was used to write sketches to the Arduino Uno so that it controls these sub components. The A to B USB cable was used to upload sketches to the microcontroller.



File Edit Sketch Tools Help

cdee3

```
#include <PS2X_lib.h> //for v1.6
#include <Servo.h>
PS2X ps2x; // create PS2 Controller Class
Servo steeringservo;
Servo LXservo;
Servo RYservo;
Servo RXservo;
Servo gripperservo;

int PlyStnLStickLtRt = 0;
int PlyStnRStickUpDn = 0;
int PlyStnRStickLtRt = 0;

int steeringservoSetting = 90;  //Setting for the Servo 1
int LXservoSetting = 90;
int RYservoSetting = 90;  //Setting for the Servo 1
int RXservoSetting = 90;
int gripperservoSetting = 90;

#define enA 9
#define in1 6
#define in2 7
```

Figure 3.7 arduino IDE sketch

3.5 Conclusion

In short the results obtained in this chapter are going to be discussed in the next chapter and the next chapter marks if the results are in line with the theory. The accuracy of this chapter is determined in the following chapter.

3.6 References

[1] Servomotors & system design components, www.bhashaelectronics.

[2] R.B. Gillespie, J. E. Colgate, M. A. Peshkin, A general framework for robot control; IEEE Transactions on Robotics and Automation, 17,4, 391-401, 2016.

[3] Devendra P. Garg and Manish Kumar, ―Optimization Techniques applied to multiple manipulators for path planning and torque minimization‖; Engineering Applications of Artificial Intelligence 15, 3-4, 241-252, 2016.

CHAPTER FOUR

RESULTS AND ANALYSIS

4.0 Introduction

This is going to present the analysis of the data obtained through the use of the prototype System designed. The results obtained during the circuit test operation will be presented in the form of a schematic, tabular form and pictorials showing the results obtained.

4.1 Prototype results and analysis

4.1.1 Results

In order to ensure the robot perform well and ready for dangerous task, several test case has been defined and following table shows the result of each test case. Also

Number	Test Case	Results and Discussion
1	Power on both robotic arm vehicle and wireless joystick control unit	-robotic arm vehicle start normally, robotic car stay still and not moving, robotic arm moving to preset position - remote control unit, that is joystick start normally
2	Press UP button of joystick or DOWN button	- robotic car moving forward when UP is pressed and backward when DOWN is pressed, robotic arm stay still.
3	Push left thumb of joystick to right or left	- Robotic arm base move right when stick is pushed right, robotic car stay still. - Robotic arm base move to the left when st4ick is pushed left, robotic car stay still.

4	Push right thumb of joystick to up or down	- Robotic arm shoulder move forward when stick is pushed up, robotic car stay still. - Robotic arm shoulder base move backward when stick is pushed down, robotic car stay still.
5	Push right thumb of joystick to left or right	- Robotic arm elbow move forward when stick is pushed right, robotic car stay still. - Robotic arm elbow move to the backward when stick is pushed left, robotic car stay still.
6	Press L1 or L2	- Robotic arm gripper opens when L1 is pressed, robotic car stay still. - Robotic arm gripper closes when L2 is pressed, robotic car stay still.

Table 4.1 robotic arm vehicle test results

The table below shows the RF link response over distance

Servo response	Distance from receiver (in meters)
normal	1
normal	5
normal	7
normal	9
normal	12

Table 4.2 RF link response over distance

4.1.2 Discussion

The results above shows the various joystick inputs and the different reactions based on the joystick commands, from the above it can be seen that the servos were connected to the arduino digital pins and play station 2 controller also to the digital arduino pins. The servos response linearly with joystick commands, the joystick consists of two potentiometers in the Y-Axis and Y-Axis with values from 0 to 255, therefore by varying the resistance we change the voltage and the servo moves with different speeds depending on how fast the stick is pushed. The L298N motor drive was connected to the PWM pins of the arduino and a 9V motor was used, the DC motor changes speed depending on the voltage applied to it. If voltage increases the speed increases and vice versa. The ENA pin is used to control the motor speed.

4.2 Software Development

The software of the prototype Arduino C was successfully implemented and was uploaded on Arduino Uno microcontroller. The hardware components responded well with code snippets and statements.

4.3 Prototype Overview

The prototype was successfully designed and built within the planned time frame. The hardware components were successfully integrated into one complete working system and responded well to programming codes from the Arduino Uno software. The development system has inputs and output pins. Some joystick inputs were analog and others digital but the outputs were digital. All the outputs are connected to digital pins of the Arduino Uno board and assigned to fixed memory address.

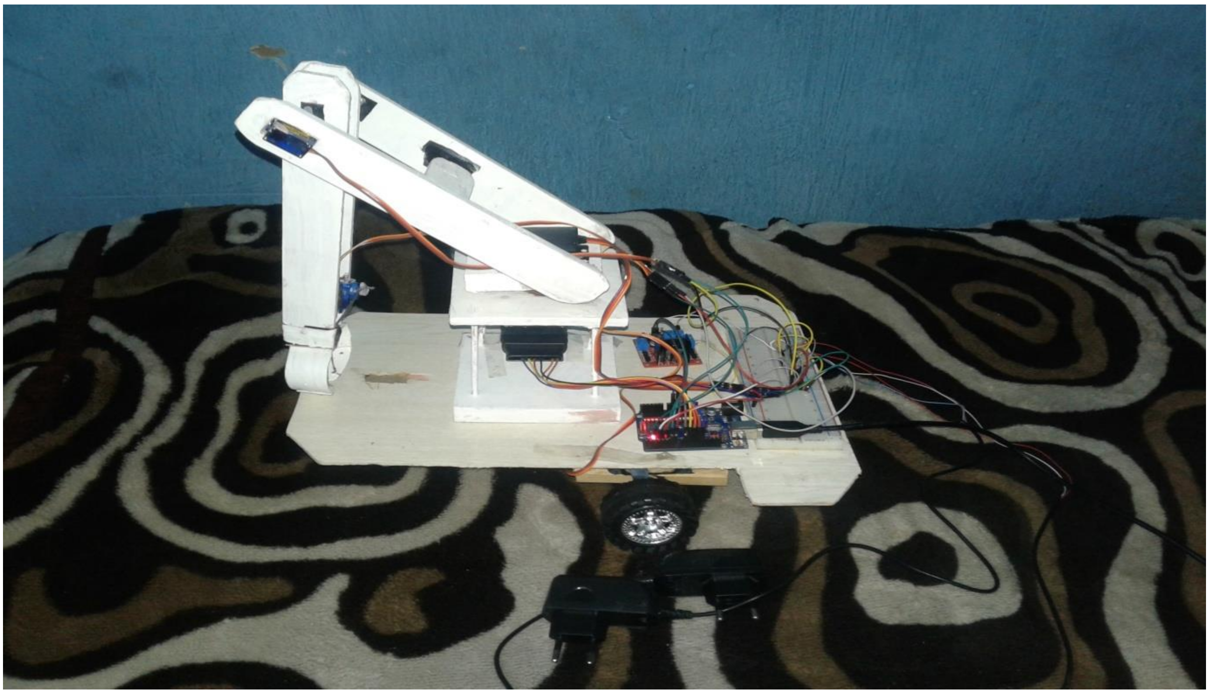


Figure 4.1 Robotic arm vehicle

CHAPTER 5

CONCLUSION

5.0 Introduction

The wireless robotic arm vehicle was successfully built. The hardware components were successfully integrated into a complete working system and operated well with the software.

5.1 Discussion

Wireless Robotic arm vehicle design in this project is able to perform dangerous task and any other industrial tasks. With the robotic hand attached to the robotic car, the direct action which is to grasps an object will be more precise and accurately. The robotic hand can act like a normal human's hand to hold, put and grasp an object without damaging the object. Furthermore, currently in market, the most popular mobile robot are all equipped with its own artificial intelligent and at the same time the development fee will getting high. With the limit of resources, the mobile robot that was designed in this project would have balance between intelligence and resources.

As part of performing dangerous task, mobile robot design in this project can used to help and assist humans to avoid any dangerous issue to the human body especially for those that may bring death. Furthermore, technologies to solve the dangerous issue are always the most wanted. Therefore, there is a demand in producing a mobile robot with high technology, efficient to perform task within limited time and effective to doing any task assign. For this project, the technology used would be moderate but will achieve both efficiency and effectiveness.

5.2 Recommendations

The wireless robotic arm vehicle in future must be made autonomous and intelligent with some sensors like ultrasonic sensor for obstacle avoidance. Also the arm must be intelligent and detect the type of material it must lift and where to put them. The robotic arm vehicle must also have a camera that detects objects and some fire sensors. In future the robotic arm must be controlled remotely via web or internet. It must also have multi degree of freedom manipulators.

5.3 Conclusion

The results obtained during testing and implementing of the project were as expected and fulfilled the objectives of the project.

APENDIX 1

```
#include <PS2X_lib.h> //for v1.6

#include <Servo.h>

PS2X ps2x; // create PS2 Controller Class

Servo LYservo;

Servo RYservo;

Servo RXservo;

Servo gripperservo;


int PlyStnLStickUpDn = 0;

int PlyStnRStickUpDn = 0;

int PlyStnRStickLtRt = 0;


int LYservoSetting = 90;

int RYservoSetting = 90;  //Setting for the Servo 1

int RXservoSetting = 90;

int gripperservoSetting = 90;


#define enA 9

#define in1 6

#define in2 7



void setup() {

  // put your setup code here, to run once:
```

```
ps2x.config_gamepad(13,11,10,12, true, false); //setup pins and settings:
GamePad(clock,command, attention, data, Pressures?,Rumble?) check for error

pinMode(enA, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(enA, OUTPUT);


LYservo.attach(4);

RYservo.attach(5);

RXservo.attach(3);

gripperservo.attach(2);


LYservo.write(90);

RYservo.write(90);

RXservo.write(90);

gripperservo.write(0);

delay(5000);

}



void loop() {

  // put your main code here, to run repeatedly:

ps2x.read_gamepad(); //This needs to be called at least once a second

                    // to get data from the controller.

// digital button
```

```
if(ps2x.Button(PSB_PAD_UP))
  {
 //set motor A backward
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
       }

 else if(ps2x.Button(PSB_PAD_DOWN))
 {
 //set motor A forward
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
 }
 else if(ps2x.Button(PSB_L1))
 {
  gripperservo.write(180);

    }
  else if(ps2x.Button(PSB_L2))
   {

  gripperservo.write(0);

    }

  else
```

```
{
   digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);



}
  //Readings from PS2 Controller Sticks are from 0 to 255
//with the neutral being 128.  The zero positions are to
//the left for X-axis movements and up for Y-axis movements.



//Analogue Stick readings
PlyStnLStickUpDn = ps2x.Analog(PSS_LY);
PlyStnRStickUpDn = ps2x.Analog(PSS_RY);
PlyStnRStickLtRt = ps2x.Analog(PSS_RX);

LYservoSetting = map(PlyStnLStickUpDn,-256,256,0,179);
RYservoSetting = map(PlyStnRStickUpDn,-256,256,0,179);
RXservoSetting = map(PlyStnRStickLtRt,-256,256,0,179);

//Write it to the Servos
LYservo.write(LYservoSetting);
RYservo.write(RYservoSetting);
RXservo.write(RXservoSetting);
delay(15);
}
```