

# **PHONE VIRTUAL ENVIRONMENT FOR RC ASSISTIVE ROBOT**

**AHMED MUFTAH ALI ZARGOUN**

**A project report submitted in partial  
Fulfillment of the requirement for the award of the  
Degree of Master of Electrical Engineering**

**Faculty of Electrical and Electronic Engineering  
Universiti Tun Hussein Onn Malaysia**

**JUNE 2014**

## **ABSTRACT**

Robotic systems nowadays are all around us and interact in many aspects such as homeowners use robotic vacuums in their homes. However, a push needed to make these systems more and more interactive with humans. The complexity of the robot controller might be one of the boundaries between the robot interactions with human. An assistive robot with an Android phone controller needed since smart phones nowadays are used almost by everyone. This assistive robot controller does not require much effort from the user, just needs moving a finger to control them. This project uses SolidWorks and Processing in the software development phase. The IOIO board has been applied as a main controller to run the system as an embedded system, and integrate the android application with the assistive robot. The communication has been established using the standard 0 dBm radio Bluetooth with range of 10 meters radius. The joint have a good desired tracking with less than 1% error. Finally the aim of this project, which is a user friendly smart phone application to monitor and control the assistive robot has been developed.

## ABSTRAK

Sistem robotik pada hari ini berada di sekeliling kita dan berinteraksi dalam pelbagai aspek contohnya, seperti pemilik rumah menggunakan vakum robotik di rumah mereka. Walau bagaimanapun, satu dorongan diperlukan untuk membuat sistem ini lebih berinteraktif dengan manusia. Sistem kawalan robot yang rumit mungkin menjadi salah satu batasan antara interaksi robot dengan manusia. Robot bantuan menggunakan telefon Android diperlukan kerana telefon pintar kini digunakan hampir oleh semua orang. Pengawal robot bantuan ini tidak memerlukan usaha yang banyak dari pengguna, pengguna hanya perlu menggerakkan jari untuk mengawal robot tersebut. Projek ini menggunakan SolidWorks dan Pemprosesan dalam fasa pembangunan perisian. Papan elektronik IOIO telah digunakan sebagai pengawal utama untuk menjalankan sistem sebagai sistem terbenam, dan mengintegrasikan aplikasi Android dengan robot bantuan. Sistem komunikasi ini telah berjaya dilaksanakan dengan menggunakan standard radio Bluetooth 0dBm dengan lingkungan 10 meter. Sesendi robot mengikuti jejak yang dikehendaki dengan baik dengan ralat kurang daripada 1%. Akhirnya tujuan projek ini, iaitu aplikasi telefon pintar yang mesra pengguna untuk memantau dan mengawal robot bantuan telah dibangunkan.

## TABLE OF CONTENTS

	<b>TITLE</b>	<b>i</b>
	<b>DECLARATION</b>	<b>ii</b>
	<b>DEDICATION</b>	<b>iii</b>
	<b>ACKNOWLEDGEMENT</b>	<b>iv</b>
	<b>ABSTRACT</b>	<b>v</b>
	<b>ABSTRAK</b>	<b>vi</b>
	<b>TABLE OF CONTENTS</b>	<b>v</b>
	<b>LIST OF FIGURE</b>	<b>ix</b>
	<b>LIST OF TABLE</b>	<b>xi</b>
<b>CHAPTER1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Problem statement	2
	1.2 Aim and objective	2
	1.3 Scope	3
	1.3.1 Hardware	3
	1.3.2 Software	3
<b>CHAPTER2</b>	<b>LITERATURE REVIEW</b>	<b>4</b>
	2.1 Servo arm robot design	4
	2.2 The IOIO board	6
	2.3 Virtual reality based robots	8
<b>CHAPTER3</b>	<b>RESEARCH METHODOLOGY</b>	<b>13</b>
	3.1 Project workflow	13
	3.2 Hardware Design and Construction	15
	3.2.1 G15 servo motor	17
	3.2.2 G15 driver	17
	3.2.3 Robot arm inverse kinematics	18
	3.2.4 Payload calculations	19

3.3	Controller design	21
3.4	Monitor system design	21
3.5	Speed control	22
<b>CHAPTER4</b>	<b>RESULTS, ANALYSIS AND DISCUSSION</b>	<b>24</b>
4.1	Payload of the robot	24
4.2	Communication range and interferences	25
4.3	Work envelope	26
4.4	Joint analysis	27
<b>CHAPTER5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>32</b>
5.1	Conclusion	32
5.2	Recommendation	33
	<b>REFERENCES</b>	
	<b>APPENDIX A</b>	

## LIST OF FIGURES

2.1	Six-servo robot arm	4
2.2	Robot arm complete assembly	5
2.3	The system startup diagram	6
2.4	Block Diagram of Complete System	7
2.5	Virtual reality simulation	8
2.6	The game simulation	10
2.7	Virtual robot workspace	11
2.8	Virtual reality robot teleoperation system	9
3.1	The project workflow	14
3.2	The system architecture	15
3.3	The assembled robot arm design	16
3.4	The G15 block diagram	17
3.5	G15 serial driver	17
3.6	The serial servo driver schematic	18
3.7	The $x$ - $z$ Plane of the robot	19
3.8	Force diagram of the robot arm	20
3.9	Controller design of the control mode	21
3.10	Monitor system design	22
3.11	The flowchart of making the phone model	23
4.1	Force diagram of the robot arm	23
4.2	The connection interferences	25
4.3	Work envelope of the robot arm	26
4.4	The part of moving the robot arm to specific point	27
4.5	The shoulder joint actual and desired	28
4.6	The real model assembled	28

4.7	The robot in feeding operation	29
4.8	The android controller design	31

**LIST OF TABLE**

4.1	Six-servo robot arm	24
4.2	Robot arm complete assembly	26



## **CHAPTER 1**

### **INTRODUCTION**

Robotic systems nowadays are all around us and interact in many aspects of our everyday lives. For example, many homeowners use robotic vacuums in their homes. Robotic components are used to assemble the cars that many of us drive. However, a push needed to make these systems more and more interactive with humans. Interaction with real people is a difficult task, and much more complex than the simple, monotonous behaviours most robots perform today. Assistive technologies such as those to help the elderly require a delicate, refined process, simplicity and easy visualization in order to gain more popularity.

In this project a smart phone controller will have a CAD model which has the same parameters as the designed RC servo robot arm, the robot arm will move as the CAD model move and vice versa.

The IOIO is a board that provides a host machine the capability of interfacing with external hardware over a variety of commonly used protocols. The original IOIO board has been specifically designed to work with Android devices. The IOIO board can be connected to its host over USB or Bluetooth, and provides a high-level Java API on the host side for using its I/O functions as if they were an integral part of the client[1].

Processing is software to make images, animations, and interactions used the sketching with code concept. The idea of sketching with code is to write a single line of code, and have a circle show up on the screen. Add a few more lines of code, and the circle follows the mouse. Another line of code and the circle changes color when the mouse is pressed[2].

Virtual reality is the digital-generated simulation of a two or three dimensional image or environment that can be interacted with in a seemingly real or physical way by a user using computer or any electronic devices.

Combination of IOIO board and Processing will be done to achieve the monitoring and controlling process. IOIO board for the I/O interface, such as digital input, digital output, analog read, analog write and I<sup>2</sup>C. Processing is an android game development kit will be used to design the CAD model.

## **1.1 Problem statement**

Number of patients keeps increasing every year due to more disease nowadays. The duty of the nurse also keeps increasing. Patients who cannot move their arm need a nurse to feed them. Unfortunately nurse sometimes is very busy to serve all the patients in the hospital. This issue cannot be solved without help from other parties such as involvement of the assistive robot. Therefore, a robot is needed in order to assist the nurse serve the patients. However, the normal robot is complex and very difficult to control the movement, since more training is needed before operate this type of robot. On the other hand Smart phones nowadays are used almost by everyone. Therefore, in this study a combination of smart phone and assistive robot will be developed to overcome this problem. This assistive robot does not require the patients more than moving a finger to assist them.

## **1.2 Aim and objectives**

The aim of this project is to make a user friendly smart phone application to monitor and control the robot arm. In order to achieve the aim there are three objectives need to be done.

- Design an RC assistive servo robot arm in a practical environment.
- Design a user friendly smart phone controller for the RC servo assistive robot arm.

- Monitor and control the assistive robot arm through the smart phone application

### **1.3 Scope**

The scopes of the project as outline below:

#### **1.3.1 Hardware:**

- RC servo robot arm.
- Android IOIO board.
- 6 channels servo motor driver.

#### **1.3.2 Software:**

- Android SDK development tools.
- Processing Android for game development.
- SolidWorks for the robot arm design.

The robot arm will be connected with the smart phone application using a USB cable or Bluetooth.

## CHAPTER 2

### LITERATURE REVIEW

This chapter presents the current navigation technologies with an overview of IR sensors detecting processing. The details in robot design is discussed in this chapter and followed by the details of the creating a virtual reality of the real robot.

#### 2.1 Servo arm robot design

In [1] six servo arm structure designed, it has 6-DOF robot arm movements. It uses three pieces of 13 kg torque metal gear, a 3.2 kg, for the first three joints because it will have a higher load compared with the rest of the joints, two pieces of 2.3 kg of servo for the joints closer to the end effector, and some sophisticated combination of aluminum alloy processing components. The robot arm is mainly 390 mm length, use 32-way controller to control the operation of arm action.



Figure 2.1: six-servo robot arm

In [3] the robot arm joints are typically actuated by electrical motors. The servo motors were chosen, since the encoders are included which automatically provide feedback to the motors and adjust the position accordingly. However, the disadvantage stated of these motors is that the rotation range of these motors is less than  $180^\circ$  span, which greatly decreases the work space of the arm and the possible positions.

The qualifications of servo motors were selected based on the maximum torque required by the structure and possible loads.

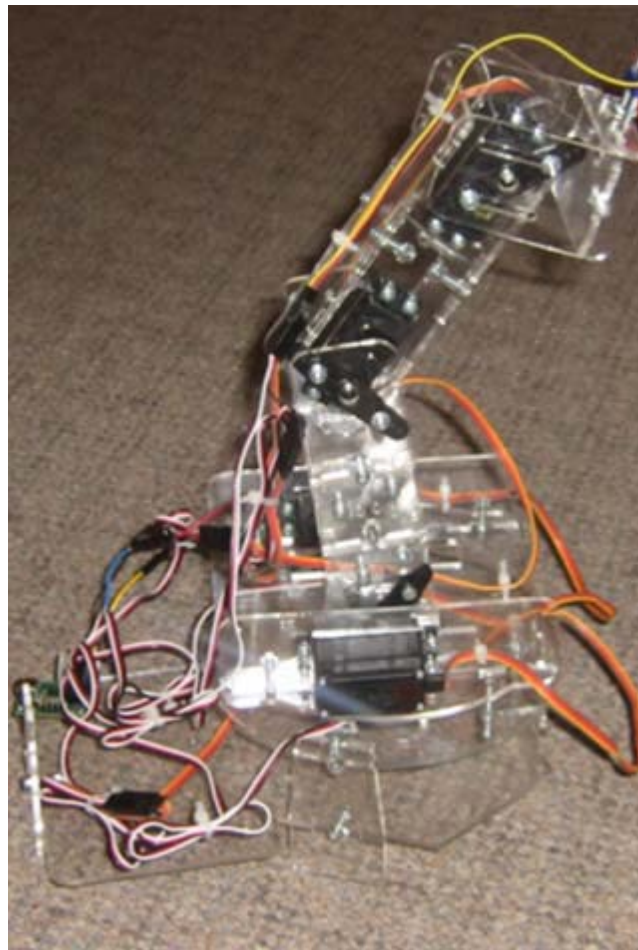


Figure 2.2: Robot arm complete assembly.

## 2.2 The IOIO board

In [4] the IOIO Board is used as direct extension of the Android device. Therefore, the board used as the MicroBridge approach, it was connected to the device as USB host, but it did not featured a user programmable microcontroller CPU. The onboard PIC CPU has a fixed firmware. The idea was to control the boards IO ports directly via the Android host. In order to use the powerfulness Java API provided. With that API, it was easy to directly access the board's general I/O pins or any dedicated communication pins for e.g. SPI, I2C and Serial communication. Internally, it also stacks upon the ADB protocol and a virtual TCP connection. To connect the Android host to NXT components, the IOIO approach is the most general one to connect any electrically interfaced hardware, actuator or sensor, to Android. The system implemented as shown in Figure 2.3.

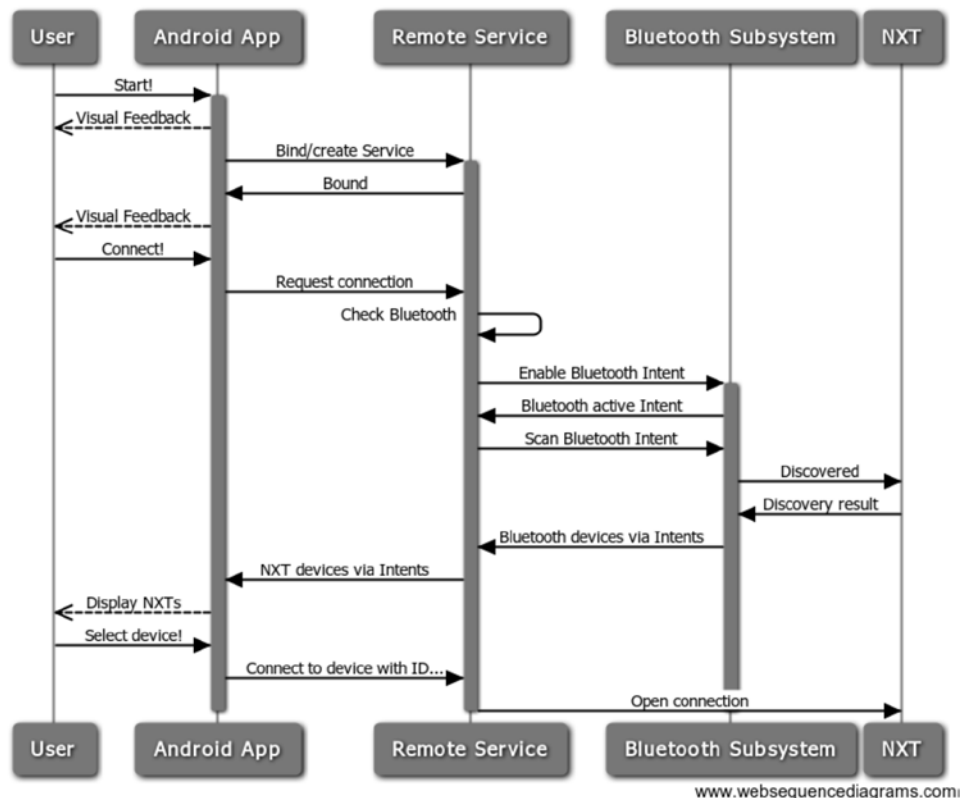


Figure 2.3: The system startup diagram

In [5] a person operates the android device with an established connection to the IOIO board. A signal is generated against each button which represents specific movements. These signals are attend by a specific PIC on the IOIO board which is directly connected to the input manipulations in this data using different scaling factors through programming and controls the direction and speed of port of 89C51 microcontroller. This digital data is fed in a microcontroller, which performs several manipulations in this data using different scaling factors through programming and controls the direction and speed of motors which results in the motion of robotic arm.

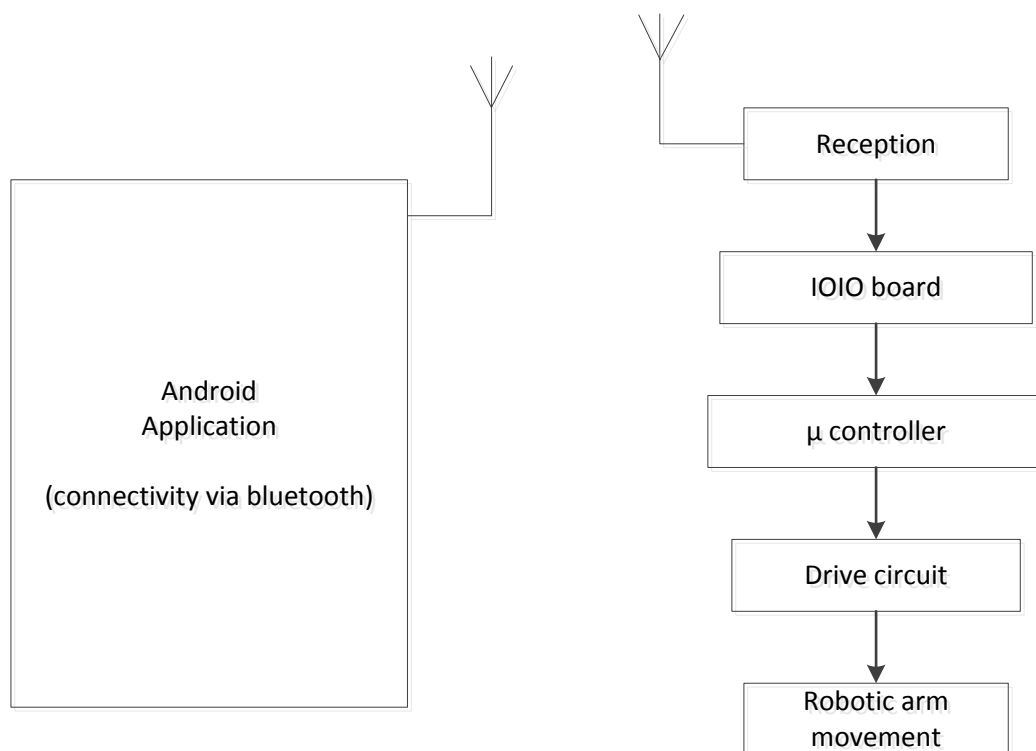


Figure 2.4: Block Diagram of Complete System

### 2.3 Virtual reality based robots

In [6] the advantages of mechanics simulation based on game engine discusses, introduces the physics engine technology in detail and emphasizes on showing how to use the game engine to simulate mechanical devices.

The virtual device can run very well. Still, it can't interact with it in real time. So a necessary variety of interactive functions add to the model. Here's a switch design for the device. Users can hold it to turn on or turn off the machine at run time. Figure 2.5 shows the final simulation drawing. When the switch is turned on, the wheel will rotate. When it is turned off, the wheel will stop rotating.

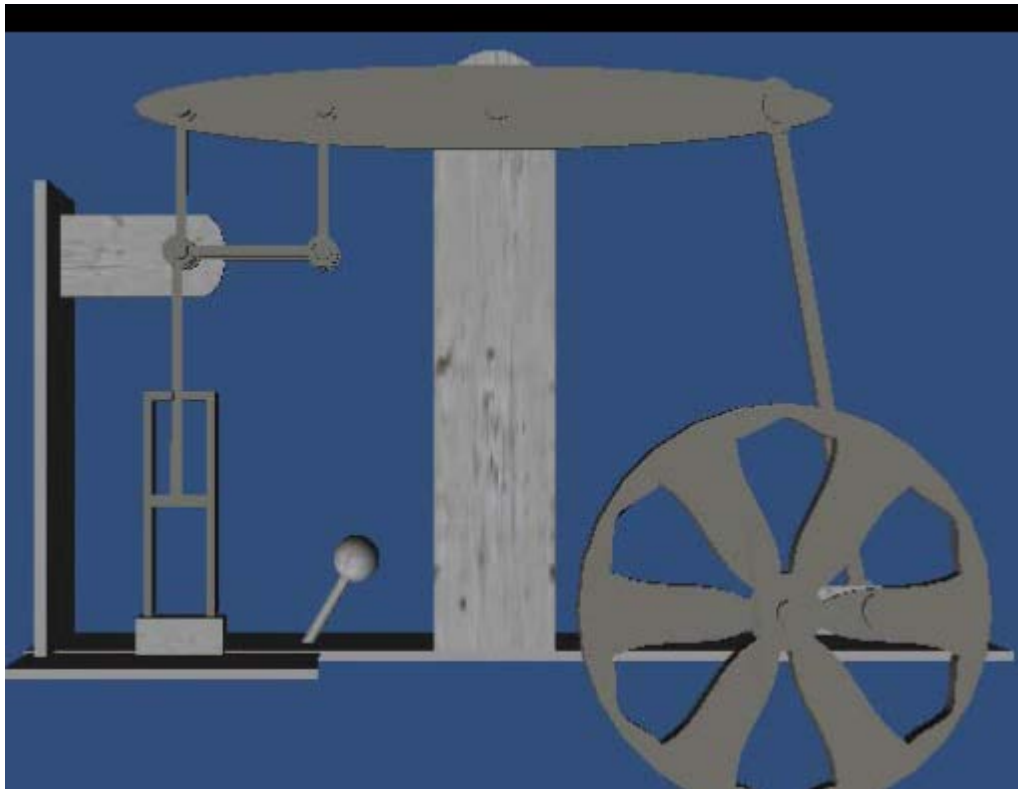


Figure 2.5: virtual reality simulation



In [7] the Karma physics engine was used to simulate interactions among solid objects such as crates, tires or bones, as well as different joints, motors or springs that make up mechanical objects. It provides physics modelling, rigidbody dynamics with constraints and collision detection.

The robot dynamics simulation is greatly simplified by the Unreal engine's vehicle class which uses the Karma engine's Car wheel Constraint joint. A vehicle is made of a chassis, one or more wheels and the joints that connect the wheels to the chassis. The wheel rotates about its rolling or hinge axis; the chassis travels along the suspension direction and the wheels steer about the steering axis. The vehicle is driven by a motor whose output torque is provided by interpolating along its Torque-Speed curve.

The proximity sensor modelling was modelled by the entity in Unreal has a vector representing its location as a triple: X-coordinate, Y-coordinate, and Z-coordinate. The Trace function will trace a line through the environment and return a reference to the first object the line collides with being it a wall, another entity, or some other obstruction. The location data returned can be manipulated in much the same way sonar data is.

Several specialized graphics packages, 3D Studio Max and NuGraf were used to speed development, the lion's share of the work was done using tools provided with a low cost video game on the conventional personal computer.

In[8] The Open Dynamics Engine (ODE) used which is an open source, high performance library for simulating rigid body dynamics. It is a fully featured, stable, mature and independent platform with an easy to use C/C++ API. It has advanced joint types and integrated collision detection with friction.

ODE is useful for simulating vehicles, objects in virtual reality environments and virtual creatures. It is currently used in many computer games, 3D authoring tools and simulation tools.

The graphical model consists of a set of 3D objects modelled using 3D Studio and imported into the XVR development environment where they are managed using the XVR scene graph.

The modelled objects are: the billiard table, the cue, the billiard balls and the skittles. An example of billiards with five skittles has been implemented.

XVR provides classes for the lighting, shading and observation point (virtual camera) management; in addition, it allows the user to superimpose 2D text onto the scene and this functionality is used in order to provide the user with messages about the system conditions, working modality and error states. Figure 2.6 shows the game simulation of the billiard.

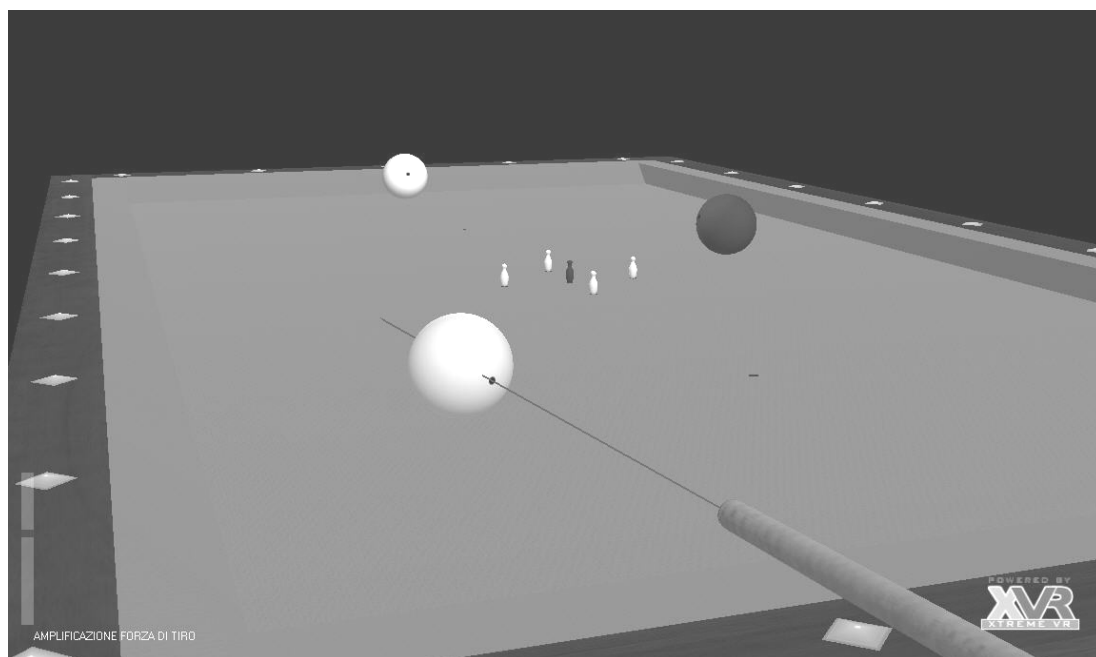


Figure 2.6: The game simulation

In [9] a research into an interface between industry robots and Virtual Reality systems and creates a virtual robot for robot arm simulation, control language conversion, physical communication, robot networking and teleportation control. The main objective of this project was to combine VR and robot technology for system testing, to establish VR based simulation workshop for defining and checking robot arm movement trajectory, in order to eliminate the chance of potential collisions.

A robot executes the defined task in the workspace by replicating operator's movements in the virtual workspace.

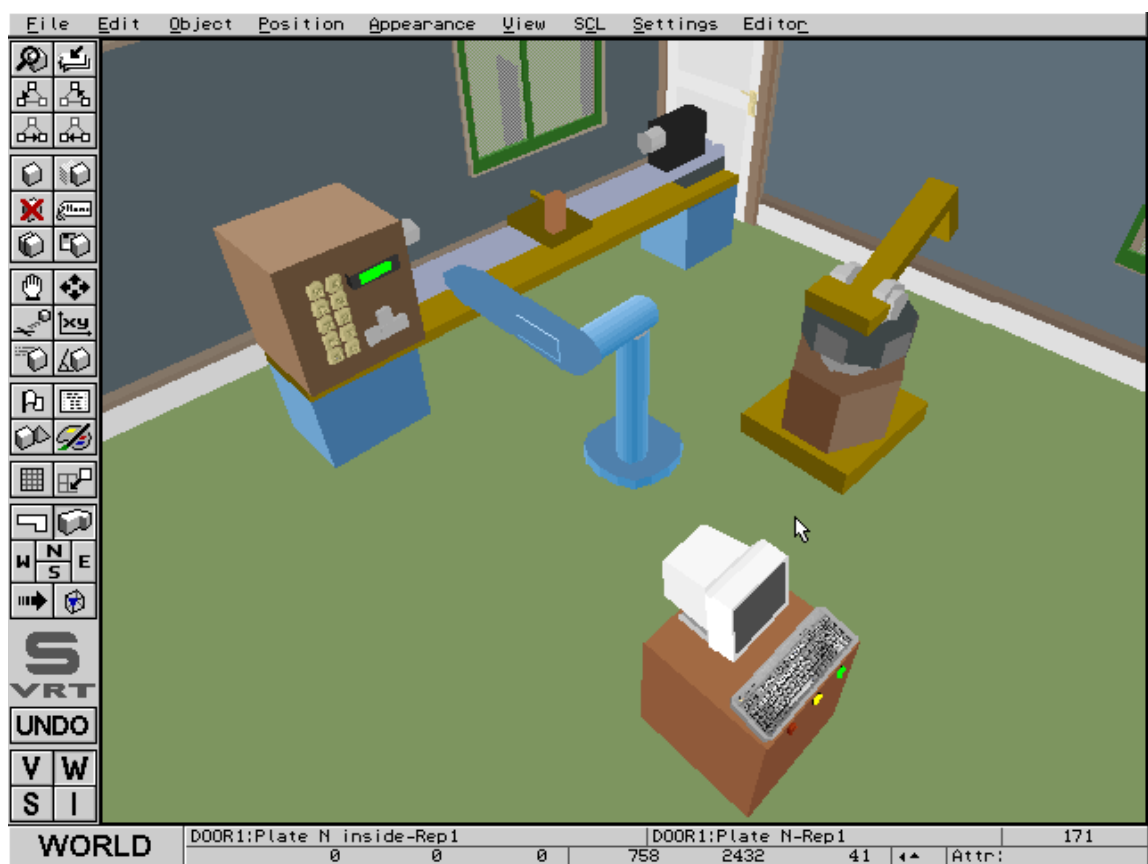


Figure 2.7: Virtual robot workspace

In [10] the virtual reality teleoperation system was made up of data glove and stereoscopic glasses combine together with virtual reality software. The virtual robot was controlled by an operator using a data glove to complete task, the virtual robot is the agent of real robot, and thus the control to the real robot is carried out by controlling the virtual robot. Every movement of the virtual robot is transmitted to the real robot, so that the virtual robot and the real robot always move synchronously.

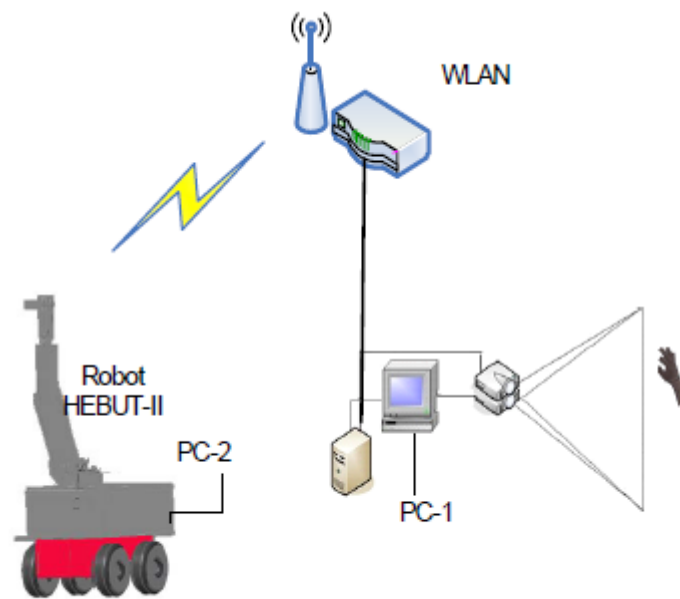


Figure 2.8: Virtual reality robot teleoperation system

The virtual robot is set up using UG, for validating whether the model is similar to the real robot or not. After the experimental results were proved that the model is very similar to the real robot. The environment scene is set up using 3Dmax. EON was selected as virtual reality software for its simple and convenient operation.

## **CHAPTER 3**

### **RESEARCH METHODOLOGY**

This chapter describes the methodology employed into account for this project. It begins with analysing the project workflow, followed by hardware and software system design.

#### **3.1 Project workflow**

This project involves both software development and hardware design and construction, in this project the designed arm robot will react according to the input from the android phone in order to achieve the objectives, the project workflow is shown in Figure 3.1.

There are several tools, which have to be mastered in order to produce the complete system. This comprises SolidWorks and Processing use the Java programming language in the software development phase. The IOIO board has been applied as a main controller to run the system as an embedded system, and integrate the android application that has been designed with the hardware design.

The project workflows continue with analysing the data, developing the complete prototype and test it on real life applications. The process is repeated until the most successful and best prototype is obtained. For the programming work, Java Programming was used to create the android application in order to control and monitor the robot arm. In order to verify the results obtained from the design.

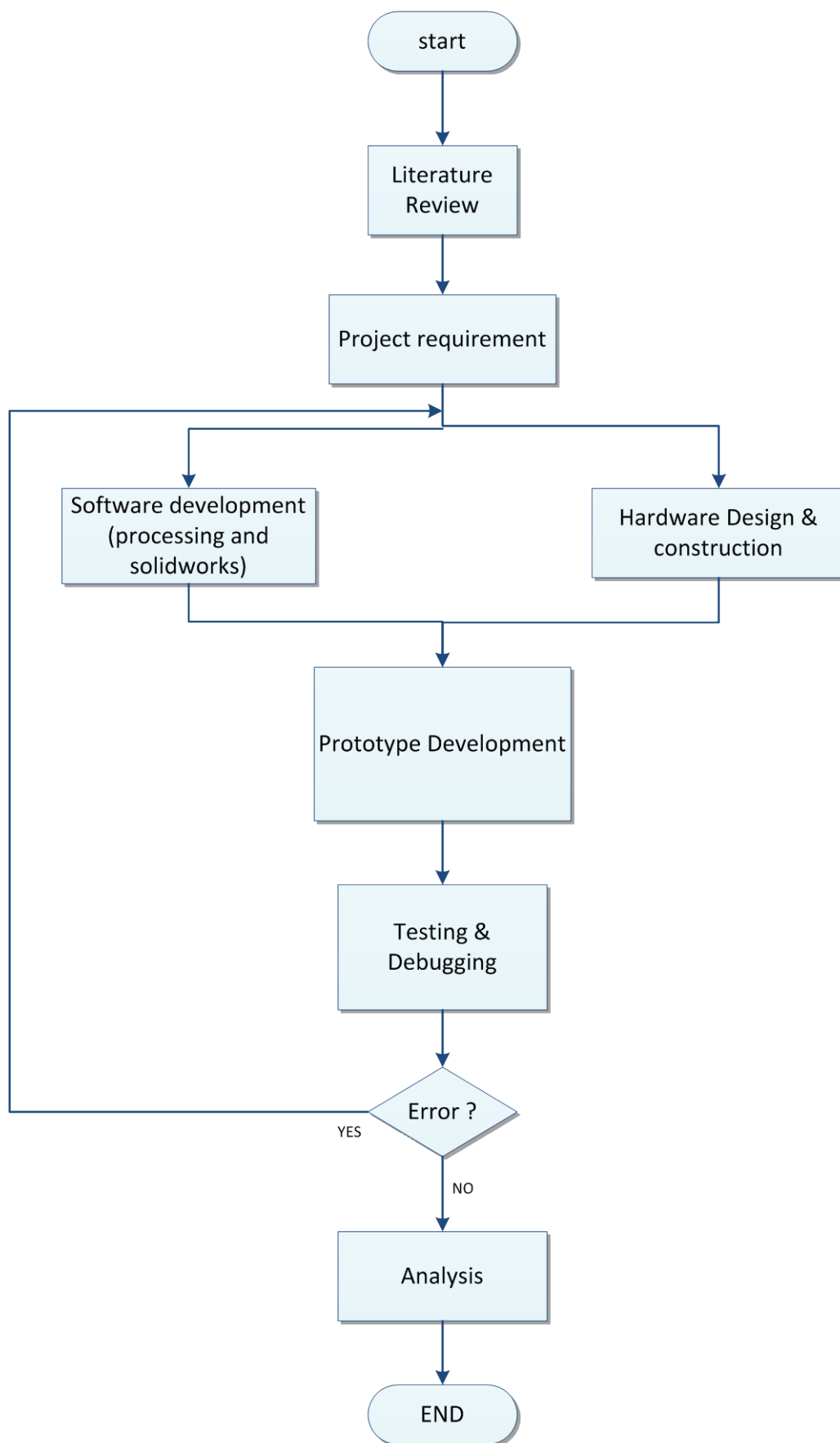


Figure 3.1: The project workflow

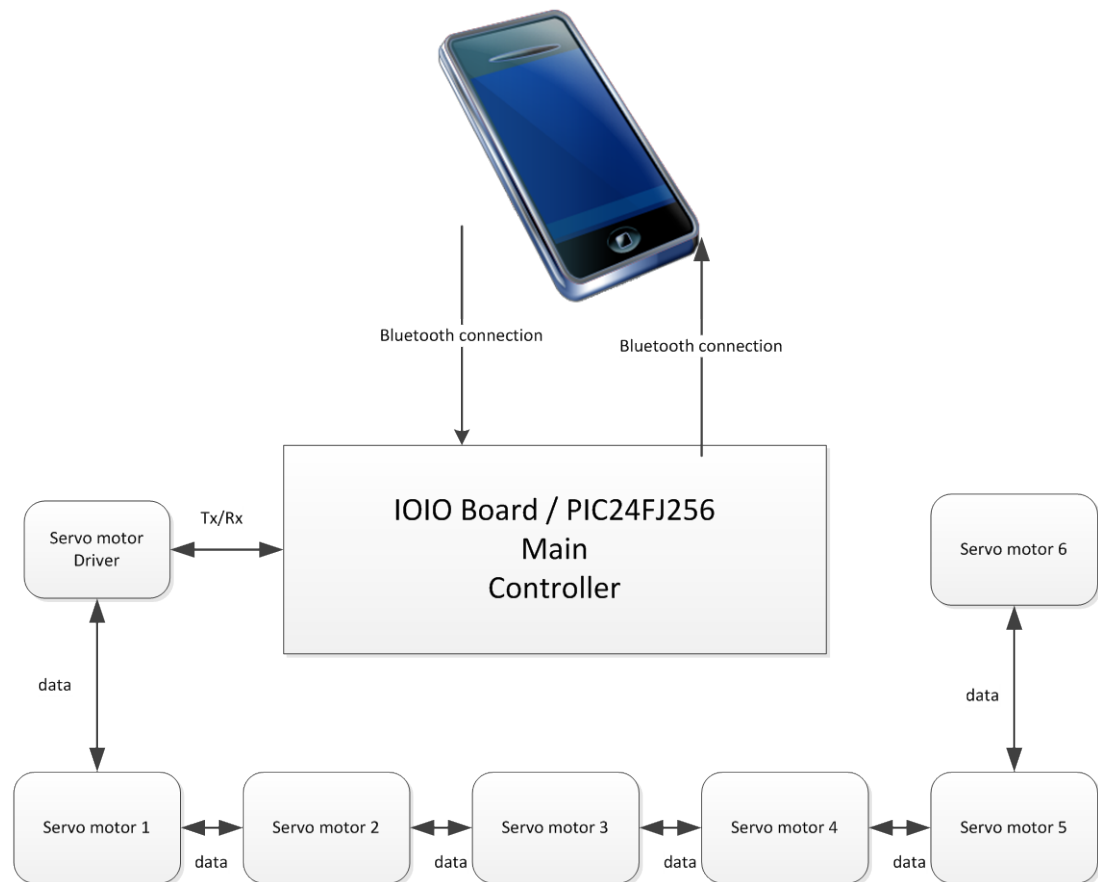


Figure 3.2: The system architecture

The system architecture of the project shown in Figure 3.2, the robot arm will move according to the user's phone model, the process of controlling the motor will be all through the Android API. Therefore, the faster the phone processor the faster process and response, the higher the phone memory the more complex system can be done.

### 3.2 Hardware Design and Construction

The robot arm shown in Figure 3.3 has 5 DOF (degree of freedom), two on the shoulder, one on the elbow and two on the wrist. This is a very common configuration, but it can replicate an important portion of the manipulative abilities of the human arm when performing table top manipulation. All joints are driven by series servos connected over PWM pins. All the links and the base of this robot are

designed using SolidWorks and created using 3D printer machine. The material has been used is ABS.

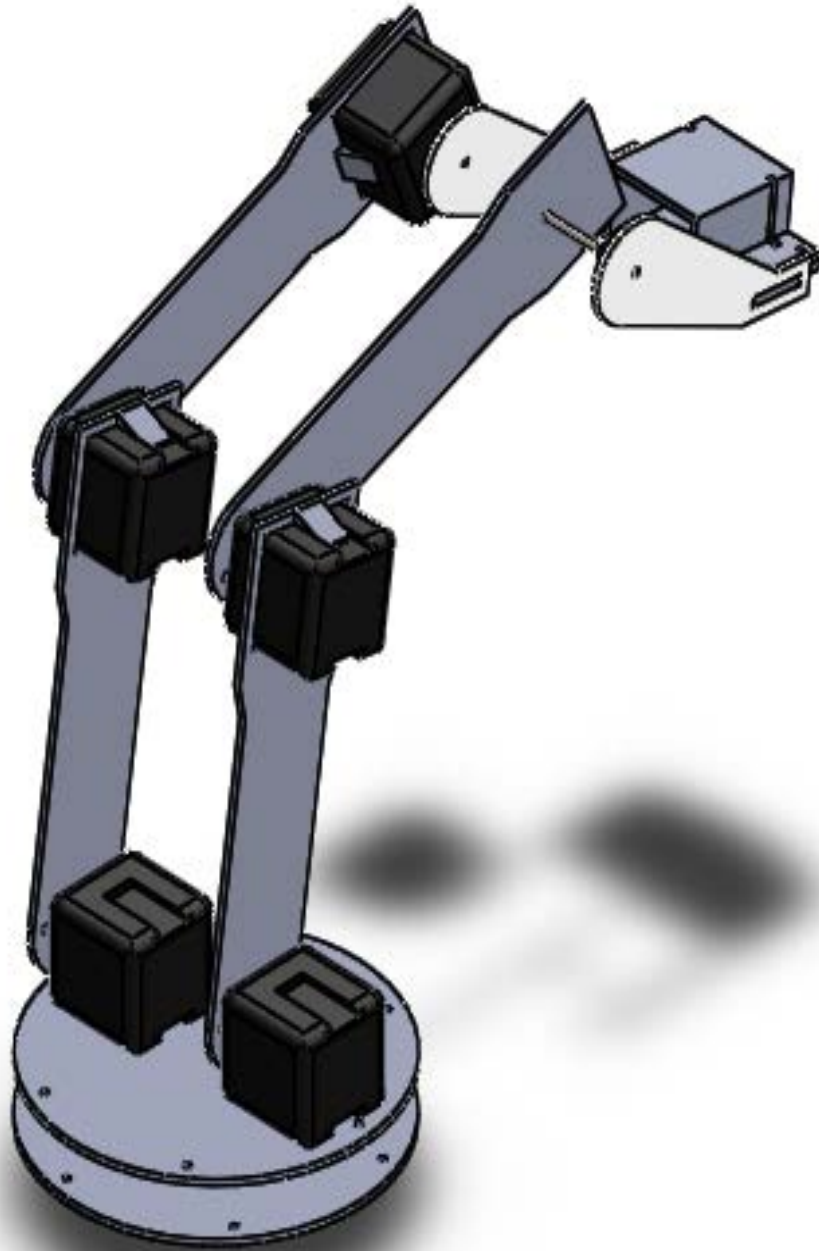


Figure 3.3: The assembled robot arm design



### 3.2.1 G15 servo motor

G15 servo motor is a smart half-duplex serial servo which incorporates gear reducer. G15 provides 360° endless turn control with resolution up to 0.33°. It allows daisy chain connection to multiple servos to create a servo network on a single line and comes with over temperature and over torque protection to prevent damage to the servo. The block diagram of the G15 servo motor shown in Figure 3.4.

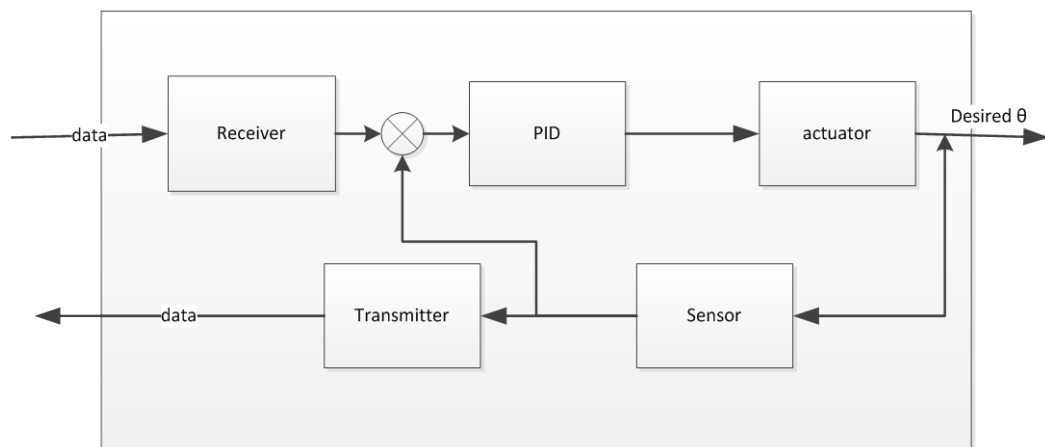


Figure 3.4: The G15 block diagram

### 3.2.2 G15 driver

It's a Half-Duplex Communication converter board. It converts UART duplex communication to half-duplex single line communication required to move the servo motor the desired position or read the actual position of the servomotor. Figure 3.5 shows the serial servo motor driver used.



Figure 3.5: G15 serial servo driver

The circuit shown in Figure 3.6 converts the UART communication lines to G15 DATA line. The UART-TX and UART-RX shown in the circuit is connected to controller's UART pin of TX and RX respectively.

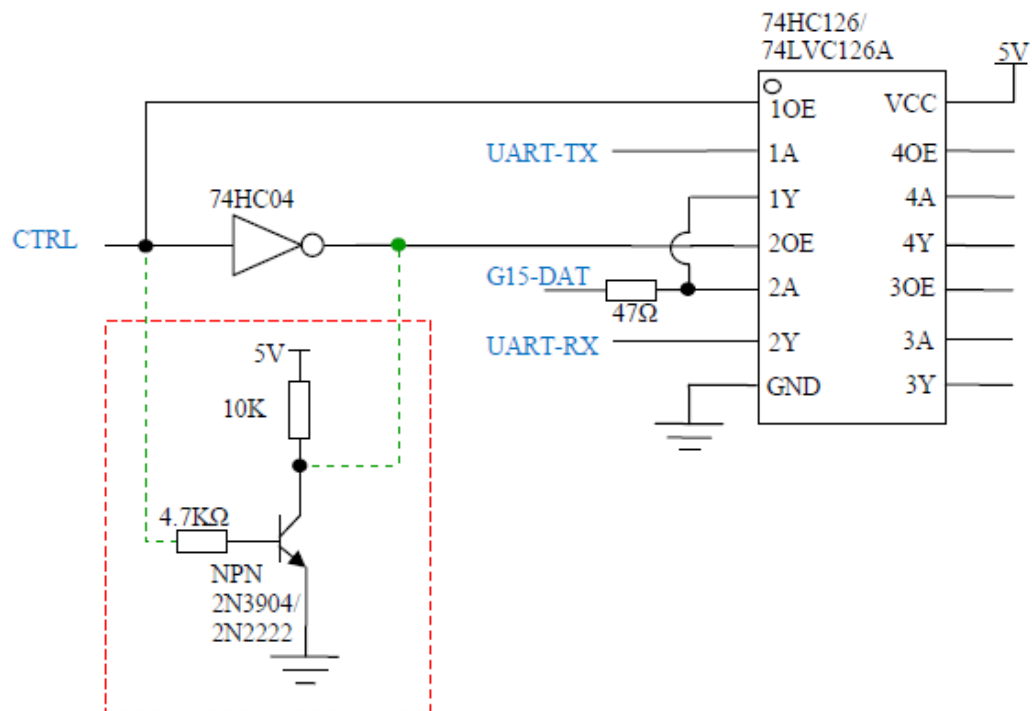


Figure 3.6: The serial servo driver circuit

### 3.2.3 Robot arm inverse kinematics

The position given by using the Cartesian coordinate system. Each motor will have a specific function: the motor for the base rotation position the final element in the  $y$  axis, the motors for the forearm and arm positions the final element in the  $x$  and  $z$  axis. Therefore,  $x$ - $z$  plane was used to simplify calculation of the joint angles ( $\theta_1$  and  $\theta_2$ ) as shown in Figure 3.7.

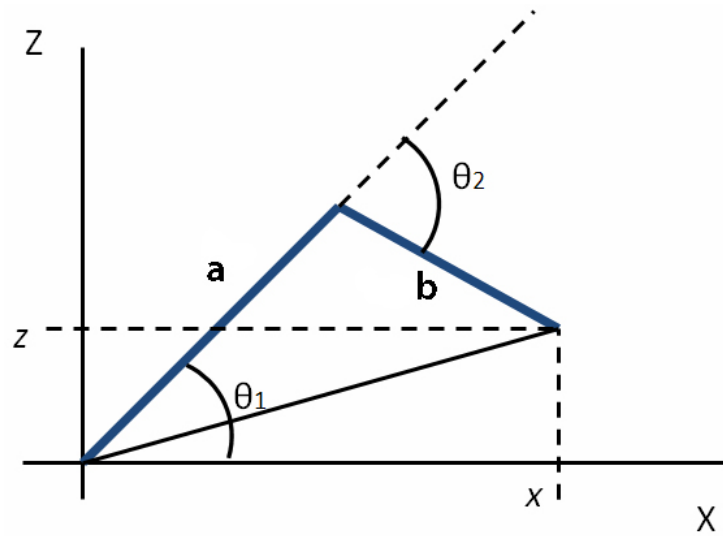


Figure 3.7: xz Plane.

Using trigonometric relations, the joint angles are determined, as follows.

$$\theta_0 = \text{arcTan}\left(\frac{z}{x}\right) \quad (3.1)$$

$$\theta_1 = \text{arcTan}\left(\frac{z}{x}\right) + \text{arcCos}\left(\frac{a^2 - b^2 + x^2 + z^2}{2 \times a \times \sqrt{x^2 + z^2}}\right) \quad (3.2)$$

$$\theta_2 = 180^\circ - \text{arcCos}\left(\frac{a^2 + b^2 - x^2 - z^2}{2 \times a \times b}\right) \quad (3.3)$$

The base rotation is going to use  $\theta_0$  and the two motors going to use  $\theta_1$  and  $\theta_2$  as shown in Figure 3.7.

### 3.2.4 Payload

Figure 4.1 shows the force diagram used to calculate the maximum load of the robotic arm. The calculations were carried out the first joint which has the largest load. The weight of the motors has been included in this calculation.

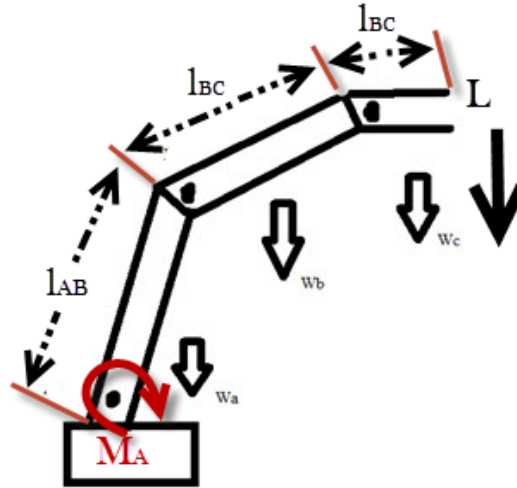


Figure 3.8: Force diagram of the robot arm

$$\sum M_A = -L(l_{AB} + l_{BC} + l_{CD}) - W_c \left( l_{AB} + l_{BC} + \frac{l_{CD}}{2} \right) - W_{motorB} (l_{AB} + l_{BC}) - W_b \left( l_{AB} + \frac{l_{BC}}{2} \right) - W_{motorC} (l_{AB}) - W_a \left( \frac{l_{AB}}{2} \right) + M_A \quad (3.4)$$

$$L = \frac{\left( -W_c \left( l_{AB} + l_{BC} + \frac{l_{CD}}{2} \right) - W_{motorB} (l_{AB} + l_{BC}) - W_b \left( l_{AB} + \frac{l_{BC}}{2} \right) - W_{motorC} (l_{AB}) - W_a \left( \frac{l_{AB}}{2} \right) + M_A \right)}{(l_{AB} + l_{BC} + l_{CD})} \quad (3.5)$$

Where  $W_a$  (wight of the first link AB) = 0.05 Kg

$W_b$  (wight of the second link BC) = 0.05 Kg

$W_c$  (wight of the third link CD) = 0.095 Kg

$W_{motor}$  (wight of the motor) = 0.063Kg

$l_{AB}$  (lenght of AB) = 0.24 m

$l_{BC}$  (lenght of BC) = 0.24 m

$l_{CD}$  (lenght of CD) = 0.01 m

$M_A$  (tourqe of the motor) = 24 Kg.cm = 2.35 N.m

L ( load) = 5.25 Kg

### 3.3 Controller design

The controller has been implemented on the IOIO board which contains PIC24FJ256. This microcontroller is the richest peripherals PIC controller, it has 24 PWM pin, 3 I<sup>2</sup>C communications and 16 analog inputs and a lot more.

The controller of this project is as shown in Figure 3.8. When the user move the phone model processing will calculate the angle of each joint, the angel value will be sent through Bluetooth communication to the microcontroller, the microcontroller transmitter (TX) the desired position of the serial driver which convert that Tx-UART data to signal line data, then sent it to the G15 servo motor. The G15 servo motor will respond to the received DATA and return another line of data which tells the system that position is determined.

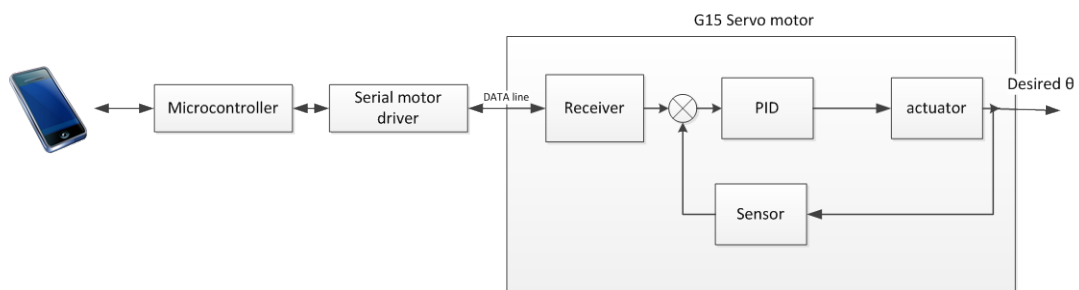


Figure 3.9: Controller design of the control mode.

### 3.4 Monitor system design

One of the objectives of this project is to monitor and control the movement of the robot, the monitor system works as follows. When there is a force or trajectory movement in progress on the real model, the user won't be able interfere the movement of the robot from the phone. Therefore, the phone model will be in monitor mode. There is a button provided to the user in order to stop any trajectory movement and make the phone model go to control mode.

The monitor system is shown in Figure 3.9, the user move the real model by force the potentiometer sense the angle of each joint and send it to the microcontroller, then the microcontroller sends the angle value to the phone model.



Figure 3.10: Monitor system design

### 3.5 The phone model design

Processing was used to develop the phone model, Processing is an open source software for various applications, in this project Processing in android mode is used to create a game represents the real model. However, processing alone won't be able to perform the task without the box2D library (physics engine). Box2D is a 2D rigid body simulation library for games used in games to make objects move in believable ways and make the game world more interactive. The procedure of making a game in Processing is as shown in Figure 3.10.

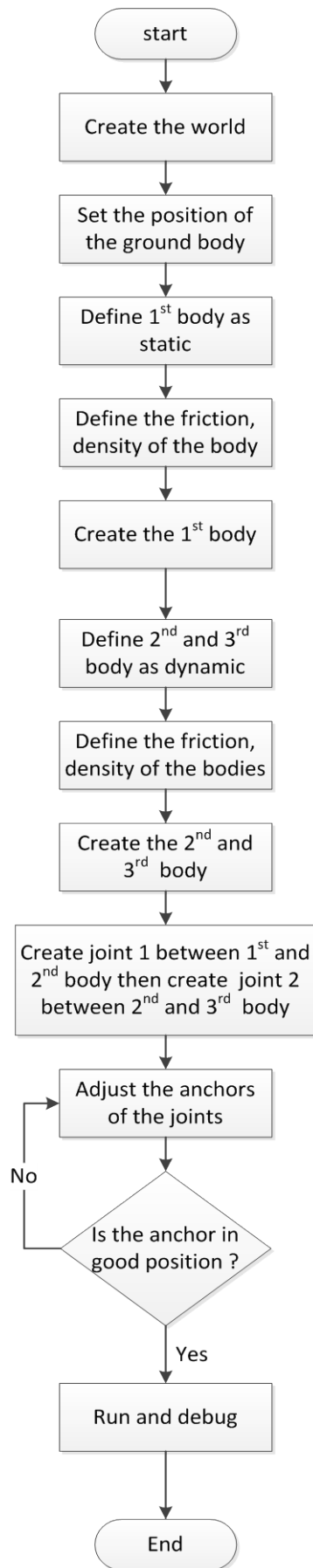


Figure 3.11: The flowchart of making the phone model

## CHAPTER 4

### RESULTS, ANALYSIS AND DISCUSSION

In this chapter there are two tests to be carried out to analyse the developed system performance; analysis on payload, communication range and work envelop have been done.

#### 4.1 Payload of the robot

By performing the sum of the moments around point A, Equation (3.4), since the  $M_A$  is known the equation to calculate the maximum load became as in Equation (3.5).

Based on the analysis and calculations have been done the maximum load of the robot arm is 5.25 Kg. It's more than enough for feeding and some basic light duties.

Table 4.1: The load test

The load value	The state
0.6 Kg	Moving smoothly
1.2 Kg	Moving smoothly
1.8 Kg	Moving smoothly
2.4 Kg	Moving slowly



## REFERENCES

1. Monk, S., *Making Android Accessories with IOIO*. 2012: O'Reilly.
2. Reas, C. and B. Fry, *Getting Started with Processing*. 2010: O'Reilly.
3. Elfasakhany, A., *Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom*. *Modern Mechanical Engineering*, 2011. **01**(02): p. 47-55.
4. Gobel, S., et al., *Using the Android Platform to control Robots*. Kassel University Germany.[Online]. Available: [www.innoc.at/fileadmin/user\\_upload/\\_temp\\_/RiE/.../65.pdf](http://www.innoc.at/fileadmin/user_upload/_temp_/RiE/.../65.pdf).
5. Ali, Z.A., et al., *Android Operated Robotic Arm*. 2014.
6. Wenfeng, H., Q. Zhouqing, and Z. Xiaoyuan. *A New Approach of Mechanics Simulation Based on Game Engine*. in *Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on*. 2012.
7. Lewis, J., et al. *Simulating visual impairments using the Unreal Engine 3 game engine*. in *Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on*. 2011.
8. De Paolis, L.T., M. Pulimeno, and G. Aloisio. *An Interactive and Immersive 3D Game Simulation Provided with Force Feedback*. in *Advances in Computer-Human Interaction, 2008 First International Conference on*. 2008.
9. Xu, Z., et al. *Virtual Reality Based Robot Graphic Simulation and Virtual Manufacturing System*. 1997. The 1997 Chinese Automation Conference in the UK (CACUK'97).
10. Guangming, Y., L. Xiaoling, and Z. Minglu. *Design of Mending Robot Based on Hearing and Virtual Reality*. in *Computer Science and Software Engineering, 2008 International Conference on*. 2008.
11. Buttussi, F., et al., *Evaluation of a 3D serious game for advanced life support retraining*. *Int J Med Inform*, 2013. **82**(9): p. 798-809.
12. Cashion, J., C. Wingrave, and J.J. LaViola, Jr., *Dense and dynamic 3D selection for game-based virtual environments*. *IEEE Trans Vis Comput Graph*, 2012. **18**(4): p. 634-42.
13. Cheah, T.C.S. and K.W. Ng. *A practical implementation of a 3D game engine*. in *Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on*. 2005.
14. de la Croix, A. and J. Skelton, *The simulation game: an analysis of interactions between students and simulated patients*. *Med Educ*, 2013. **47**(1): p. 49-58.
15. De Paolis, L.T., G. Aloisio, and M. Pulimeno. *A Simulation of a Billiards Game Based on Marker Detection*. in *Advances in Computer-Human Interactions, 2009. ACHI '09. Second International Conferences on*. 2009.

16. Lamiable, A., et al., *An algorithmic game-theory approach for coarse-grain prediction of RNA 3D structure*. IEEE/ACM Trans Comput Biol Bioinform, 2013. **10**(1): p. 193-9.
17. Muehl, W. and J. Novak, *Game development essentials : game simulation development*. 2008, Clifton Park, NY: Thomson / Delmar Learning. xxvii, 242 p.
18. Setiawan, A., E. Kartikadarma, and H. Haryanto. *Preservation of Gobak Sodor traditional games using augmented reality computer game simulation*. in *Information and Communication Technology (ICoICT), 2013 International Conference of*. 2013.