Sudan University of Science and Technology
College of Engineering
Electrical Engineering Department

Hand Gesture Controlled Car Using Arduino

التحكم في سياره بايمائات اليد عن طريق اردوينو

A Project Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering

Prepared By:

1. Mohamed Hassan Mohamed Mohamed Ahmed
2. Khalid Alwaleed Ali Altayeb
3. Yousif Fatharhman Yousif Eltayeb
4. Motwakel Kamal Osman Mohammed

Supervised By:

UST. Galal Abdalrahman Mohammed

October 2018

الآية

بسم الله الرحمن الرحيم

قال تعالي:

۞ وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوا إِلَّا إِيَّاهُ وَبِالْوَالِدَيْنِ إِحْسَانًا ۚ إِمَّا يَبْلُغَنَّ عِندَكَ الْكِبَرَ أَحَدُهُمَا أَوْ كِلَاهُمَا فَلَا تَقُل لَّهُمَا أُفٍّ وَلَا تَنْهَرْهُمَا وَقُل لَّهُمَا قَوْلًا كَرِيمًا (23)

صدق الله العظيم

سورة الإسراء الاية(23)

DEDICATION

As well as everything that we do, we would be honored to dedicate this work to our parents for their emotional and financial support, our brothers, our sisters and our friends. They have given us the drive and discipline to tackle any task with enthusiasm and determination. Without their love and support this project would not have been made possible.

ACKNOWLEDGEMENT

Great thankful to “Allah” who lightens our ways, our lives and our hearts. We would like to convey our appreciation to the Sudan University of Science and Technology. Apart from the efforts of us, the success of any research depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this research. We would like to show our greatest appreciation to our supervisor Galal Abdalrahman. We cannot say thank you enough for his tremendous support and help.

ABSTRACT

Robots are an important technology in science to be used in uncertain conditions such as security operations as they can be followed where the operator's instructions are followed and the task performed. We have designed a car so that it is controlled by hand gesture, by using accelerometer. It consists mainly of two parts, the first is the transmitter and the other is the receiving part. The transmitter will transmit the signal according to the position of the accelerometer attached to the hand and the receiver will receive the signal and make the vehicle moving in the certain direction. Here the program was designed using Arduino. Any robot can be controlled using Arduino, not only can we control it, but we can use it to do at least 256 different functions.

مستخلص

تعد الروبوتات تكنولوجيا مهمه في مجال العلوم ليتم استخدامها في الظروف غير المؤكده مثل العمليات الامنيه كما يمكن عنلها حيث تتبع تعليمات المشغل البشري وتنفيذ مهمته. وقد صممنا سيارة بحيث يتم التحكم بها عن طريق ايمائات اليد بواسطة مقياس التسارع. وهو يتألف أساسا من جزأين ، الاول هو الجزء المرسل والآخر هو الجزء المتلقي. سوف يقوم جهاز الإرسال بنقل الإشارة وفقًا لموضع مقياس السرعة المرفق على اليد وسوف يتلقى جهاز الاستقبال الإشارة ويجعل السياره تتحرك في الاتجاه المعني. هنا تم تصميم البرنامج باستخدام اردوينو. يمكن التحكم في أي روبوت باستخدام Arduino ، وليس فقط يمكننا التحكم فيه ، ولكن يمكننا استخدامه للقيام بما لا يقل عن 256 وظيفة مختلفة.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SISO	Single Input Single Output
PID	Proportional Integral Derivative
MIMO	Multiple Input Multiple Output
RAM	Random Access Memory
ROM	Read Only Memory
LCD	Liquid Crystal Display
LED	Light Emitting Diode
ADC	Analog Digital Converter
ROV	Remotely Operated Vehicle
GUSI	Graphical User Interfaces
RF	Radio Frequency
MPU	Microprocessor Unit
IMU	Internal Measurement Unit
IDE	Integrated Development Environment
MEMS	Micro Electro Mechanical System
DC	Direct Current
IC	Integrated Circuit
PCB	Printed Circuit Bard
DOS	Disk Operating System
MCAD	Microsoft Certified Application Developer

CHAPTER ONE
INTRODUCTION

1.1Overview

Gesture recognition is an important achievement in control science and language technology which aims to interpret human gesture via mathematical algorithm. Gesture can emanate from any bodily motion but commonly from the face or hand is taken. It is therefore a way for computer to start understanding human body language, thus building a strong bridge between machine and human than the primitive text user interfaces or graphical user interface (GUIs). It still limits the mouse and keyboards for inputs. Gesture recognition makes us able to communicate with the machines naturally without any mechanical devices. Here we are using an accelerometer in conjunction with a microcontroller and RF links to make a gesture controlled.

1.2 Problem Statement

The minimization of the distance between the physical world and the digital world .The way humans interact among themselves could be implemented in communication with the digital world by interpreting gestures via mathematical algorithm .Difficulty controlling for people with special needs (disabled).The risk that humans have in hazardous workplaces.

1.3 Objectives

The main objectives of this study are:

- ❖ Design a system works on the interaction of humans and machines through the gesture to facilitating the movement of disabled peoples, minimize the space between physical world and digital world and facilitate reaching dangerous places.

1.4 Methodology

The circuit diagram was drown and checked by simulation program called proteus. The model has been installed and consists of two parts: The first part is getting data from the Accelerometer by the Arduino. The Arduino continuously acquires data from the IMU and based on the predefined parameters, it sends a data to the RF Transmitter. The second part of the project is the Wireless Communication between the RF Transmitter and RF Receiver. The RF Transmitter, upon receiving data from Arduino, transmits it through the RF Communication to the RF Receiver.

1.5 Project Layout

This project consists of five chapters: Chapter one gives an introduction about the principles of the project, in addition its reasons, motivation and objectives. Chapter two discusses the literature review of control systems, microcontroller System and vehicle (car). Chapter three describes the model circuit component, details of each component and block diagram of circuit. Chapter four shows the software, simulation and operation. Finally chapter five provides the conclusion and recommendations.

CHAPTER TWO
LITERATURE REVIEW

2.1 Control System

There are two major divisions in control theory, namely, classical and modern, which have direct implications over the control engineering applications. The scope of classical control theory is limited to Single-Input and Single-Output (SISO) system design, except when analyzing for disturbance rejection using a second input. The system analysis is carried out in the time domain using differential equations, in the complex-s domain with the Laplace transform, or in the frequency domain by transforming from the complex-s domain. Many systems may be assumed to have a second order and single variable system response in the time domain. A controller designed using classical theory often requires on-site tuning due to incorrect design approximations. Yet, due to the easier physical implementation of classical controller designs as compared to systems designed using modern control theory, these controllers are preferred in most industrial applications. The most common controllers designed using classical control theory is Proportional-Integral-Derivative (PID) controllers. A less common implementation may include either or both a lead and lag filter [1].

Ultimate the end goal is to meet requirements set typically provided in the time-domain called the Step response, or at times in the frequency domain called the Open-Loop response. The Step response characteristics applied in a specification are typically percent overshoot, settling time, etc. The open-loop response characteristics applied in a specification are typically Gain and Phase margin and bandwidth. These characteristics may be evaluated through simulation including a dynamic model of the system under control coupled with the compensation model. In contrast, modern control theory is carried out

in the state space, and can deal with Multiple-Input and Multiple-Output (MIMO) systems. This overcomes the limitations of classical control theory in more sophisticated design problems, such as fighter aircraft control, with the limitation that no frequency domain analysis is possible. In modern design, a system is represented to the greatest advantage as a set of decoupled first order differential equations defined using state variables. Nonlinear, multivariable, adaptive and robust control theories come under this division. Matrix methods are significantly limited for MIMO systems where linear independence cannot be assured in the relationship between inputs and outputs[2].

A control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems. Industrial control systems are used in industrial production for controlling equipment or machines [2]. There are two common classes of control systems, open loop control systems and closed loop control systems. In open loop control systems output is generated based on inputs. In closed loop control systems current output is taken into consideration and corrections are made based on feedback. A closed loop system is also called a feedback control system.

2.1.1 Open-loop control systems

In an open-loop control system, the controller independently calculates exact voltage or current needed by the actuator to do the job and sends it as shown in Figure 2.1. With this approach, however, the controller never actually knows if the actuator did what it was supposed to because there is no feedback. This system absolutely depends on the controller knowing the operating characteristics of actuator [3]. The actuator on the process is very repeatable and reliable. Relays and stepper motors are devices with reliable characteristics and are usually open-loop operations. Actuators such as motors or flow valves are sometimes used in open-loop operation, but they must be calibrated and adjusted at regular intervals to ensure proper system operation [3].

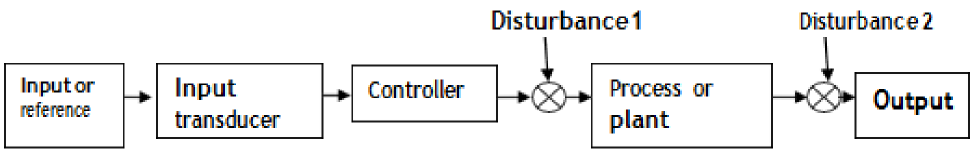

Figure 2.1: Open Loop Control System

2.1.2 Closed-loop control systems

In a closed-loop control system, the output of the process (controlled variable) is constantly monitored by a sensor; the sensor samples the system output and converts this measurement into an electric signal that it passes back to the controller as shown in Figure 2.2. Because the controller knows what the system is actually doing, it can make any adjustment necessary to keep the output where it belongs. The signals from the controller to the actuator are the forward path, and the signal from the sensor to the controller is the feedback. The feedback signal is subtracted from the set point at the comparator [3].The self-correcting feature of closed-loop control makes it preferable over open-loop control in many applications, despite the additional hardware required. This is because closed-loop system provides reliable, repeatable performance even when the system components themselves are not absolutely repeatable or precisely known [3].

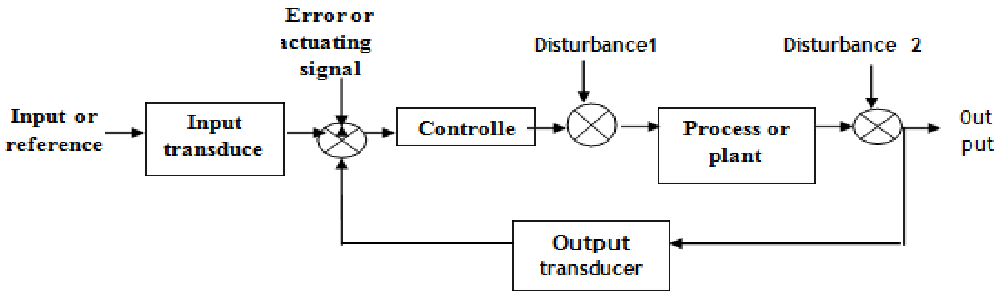

Figure 2.2: Closed Loop Control System

2.2 Microcontroller

A microcontroller is a single-chip computer .Micro suggests that the device is small, and controller suggests that it is used in control applications. Another term for microcontroller is embedded controller, since most of the microcontrollers are built into or embedded in the devices that controlling. Microcontrollers have traditionally been programmed using the assembly language of the target device. Although the assembly language is fast, it has several disadvantages. An assembly program consists of mnemonics, which makes learning and maintaining a program written using the assembly language difficult. Also, microcontrollers manufactured by different firms have different assembly languages, so the user must learn a new language with every new microcontroller he or she uses. Microcontrollers can also be programmed using a high-level language, such as BASIC, PASCAL, or C. High-level languages are much easier to learn than assembly languages and also facilitate the development of large and Complex programs.

Microcontroller is a highly integrated chip that contains Central Processing Unit (CPU), Random Access Memory (RAM), Read Only Memory (ROM) and Input/output (I/O) ports. Unlike general-purpose computer, which also includes all of these components, microcontroller is designed for a very specific task to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production cost.

Figure 2.3: Microcontroller.

A microcontroller basically contains one or more following components:

❖Central processing unit (CPU).

❖Random Access Memory (RAM).

❖Read Only Memory (ROM).

❖Input/output ports.

❖Timers and Counters.

❖Interrupt Controls.

❖Analog to digital converters.

❖Digital Analog converters.
❖Serial interfacing ports.

❖Oscillatory circuits.
The microcontrollers have the following advantages
❖Microcontrollers act as a microcomputer without any digital parts.

❖As the higher integration inside microcontroller reduce cost and size of the system.

❖Usage of microcontroller is simple

❖Most of the pins are programmable by the user for performing different functions.

❖Easily interface additional RAM, ROM, I /O ports.

❖Low time required for performing operations.
The Disadvantages of Microcontrollers are:
❖Microcontrollers have got more complex architecture than microprocessors.

❖Only perform limited number of executions simultaneously.

❖Mostly used in micro-equipment.

❖Cannot interface high power devices directly.

2.2.1 Applications of microcontrollers

❖Light sensing and controlling devices.

❖Fire detection and safety device.

❖Industrial instrumentation devices.

❖Process control devices.

❖Volt meter.

❖Current meter.

❖Hand-held metering systems.

The architecture of micro controller:

❖ Harvard architecture

The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data.

❖ von Neumann architecture

The von Neumann architecture also known as the von Neumann model or Princeton architecture is a computer architecture based on a 1945 description by the mathematician and physicist John von Neumann and others in the First Draft of a Report on the EDVAC.

2.3 Vehicle

The programmed vehicles are small vehicles programmed to control their movement in order to accomplish certain operations, and can be divided into two main types, remotely and autonomous vehicles according to the method of control and each type of them can be apportioned according to the work environment to the aircraft vehicles, car vehicles and underwater vehicles.

Car vehicle is a vehicle which works on land for certain objectives and it varies by the purpose of the designing and the method of control and size.

2.3.1 Manned vehicles (car)

Manned Vehicles is any vehicles that are not able to operate without a human occupant .

❖CAR:

The word car is believed to originate from the Latin word carrus or carrum ("wheeled vehicle"), or the Middle English word carre (meaning "two-wheel cart", from Old North French). In turn, these originated from the Gaulish word karros (a Gallic chariot). It originally referred to any wheeled horse-drawn vehicle, such as a cart, carriage, or wagon. "Motor car" is attested from 1895.

The first working steam-powered vehicle was designed and quite possibly built by Ferdinand Verbiest, a Flemish member of a Jesuit mission in China around 1672. It was a 65-cm-long scale-model toy for the Chinese Emperor that was unable to carry a driver or a passenger. It is not known with certainty if Verbiest's model was successfully built or ran.

❖motorized wheelchair:

An electrically propelled tricycle was developed by the R.A. Harding Company in England in the 1930s. The electric-powered wheelchair was invented by George Klein who worked for the National Research Council of Canada, to assist injured veterans after World War.

Power chairs are generally four-wheeled or six-wheeled and non-folding, however some folding designs exist and other designs may have some ability to partially dismantle for transit.

Four general styles of power chair drive systems exist: front, center or rear wheel drive and all-wheel drive. Powered wheels are typically somewhat larger than the trailing/castor wheels, while castor wheels are typically larger than the castors on a manual chair. Centre wheel drive power chairs have castors at both front and rear for a six-wheel layout.

Figure 2.4: Motorized Wheelchair.

2.3.2 Autonomous car

Fully autonomous vehicles, also known as driverless cars, already exist in prototype (such as the Google driverless car), and are expected to be commercially available around 2020. According to urban designer and futurist Michael E. Arth, driverless electric vehicles in conjunction with the increased use of virtual reality for work, travel, and pleasure could reduce the world's 800 million vehicles to a fraction of that number within a few decades. This would be possible if almost all private cars requiring drivers, which are not in use and parked 90% of the time, would be traded for public self-driving taxis that would be in near constant use. This would also allow for getting the appropriate vehicle for the particular need a bus could come for a group of people, a limousine could come for a special night out, and a Segway could come for a short trip down the street for one person. Children could be chauffeured in supervised safety, DUIs would no longer exist, and 41,000 lives could be saved each year in the US alone.

Figure 2.5: Autonomous Car.

2.3.3 Remotely operated vehicle

Remotely Operated Vehicle is an unmanned vehicle also is a tethered vehicle which can be controlled by an operator on the surface, which controls its movement and actions without the need for a diver. The first ROV known, called (Poodle), was created by a French scientist, engineer and explorer Dimitri Rubik off in 1953. The first funding attempt of the early ROV technology was made by the US Navy in the 1960’s.

Another technology development of the ROVs was made by commercial firms that saw the future in ROV support of the offshore oil fields. These ROVs were first used as inspection vehicles and a decade later as vehicles for maintenance of the offshore oil fields [5]. Nowadays ROVs are responsible for numerous tasks in many fields. All in all, ROVs can work on all stages, from construction to maintenance. ROVs are better than divers by the following advantages:

- ❖ ROVs perform tasks faster than divers and much more consistently and accurately.
- ❖ ROVs do not get cold, tired, or hungry.
- ❖ ROVs do not have a waiting period after a dive before they can fly.
- ❖ ROVs can spend unlimited time at depth and do not have to deal with decompression.
- ❖ ROVs can safely perform penetrations and can safely work around (oil, sewage, etc.).
- ❖ Micros ROV can be set up quickly, make a brief dive, and move to numerous sites in a day.
- ❖ A ROV can be operated from a variety of platforms, so a ladder or swim platform is not necessary.
- ❖ While an ROV can be lost or destroyed in extreme situations, loss of life is not a factor; Global Positioning System (GPS) can be added to find ROV.

2.4 Motor

Motor a machine that converts electrical energy into mechanical energy by means of the forces exerted on a current-carrying coil placed in magnetic field.

2.4.1 Types of motor

Motors are classified into three types which are:

❖DC motor: dc permanent magnet motor, series motor, compounded motor, shunt motor, separately excited motor.

❖AC motor: induction motor single phase and three phase, synchronous motor.

❖Special motor: stepper motor, brushless DC motor, hysteresis motor, reluctance motor, universal motor.

2.4.2 Applications of motors

Electric motors are found in applications as diverse as:

❖Industrial fans.

❖Blowers.

❖Pumps.

❖Machine tools.

❖Household appliances.

❖Power tools

CHAPTER THREE

SYSTEM COMPONENTS AND IMPLIMENTATION

3.1 Introduction

We have designed a simple Hand Gesture Controlled car using Arduino. This Hand Gesture Controlled car is based on Arduino Nano, MPU6050, RF Transmitter-Receiver Pair and L293D Motor Driver.

Even though the title says it as a Hand Gestured Controlled Robot, technically this robot is controlled by the tilt of the hand.

Majority of the industrial robots are autonomous as they are required to operate at high speed and with great accuracy. But some applications require semi-autonomous or human controlled robots.

Some of the most commonly used control systems are voice recognition, tactile or touch controlled and motion controlled.

One of the frequently implemented motion controlled robot is a Hand Gesture Controlled Robot. In this project, a hand gesture controlled robot is developed using MPU6050, which is a 3-axis Accelerometer and 3-axis Gyroscope sensor and the controller part is Arduino Nano.

Instead of using a remote control with buttons or a joystick, the gestures of the hand are used to control the motion of the robot.

The project is based on wireless communication, where the data from the hand gestures is transmitted to the robot over RF link (RF Transmitter – Receiver pair).

The project is divided into transmitter and receiver section. The circuit diagram and components are explained separately for both transmitter and receiver sections.

The Applications is Wireless controlled robots are very useful in many applications like remote surveillance, military etc.

Hand gesture controlled robot can be used by physically challenged in wheelchairs.

Hand gesture controlled industrial grade robotic arms can be developed.

3.2 MODEL CIRCUIT COMPONENT

the components that we need are:

- arduino nano.
- imu (accelerometer).
- dc motor and wheels.
- motor driver (l293d).
- voltage regulator.
- rf module.

- ❖ battary.
- ❖ wires.

3.3 Details Of Each Component

Here we describe the component:

3.3.1 Arduino nano

Arduino is an open source hardware technology coupled with a programming language and an Integrated Development Environment (IDE).

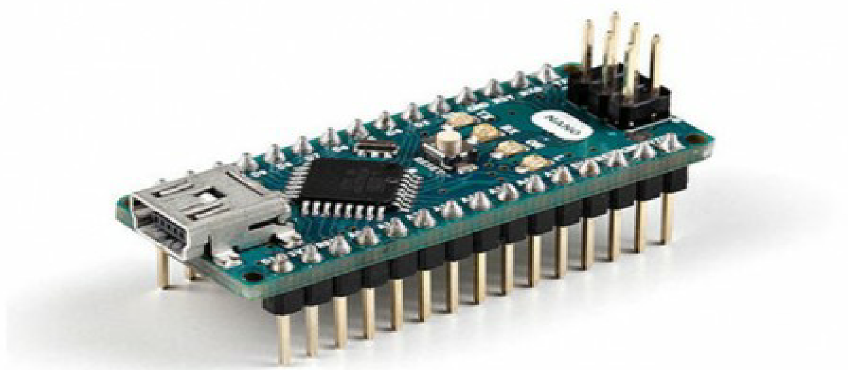

Figure 3.1: Arduino Nano.

3.3.1.1Arduino Nano specification

Table 3.1: Arduino Nano specification.

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

3.3.1.2Arduino Nano Pin Layout

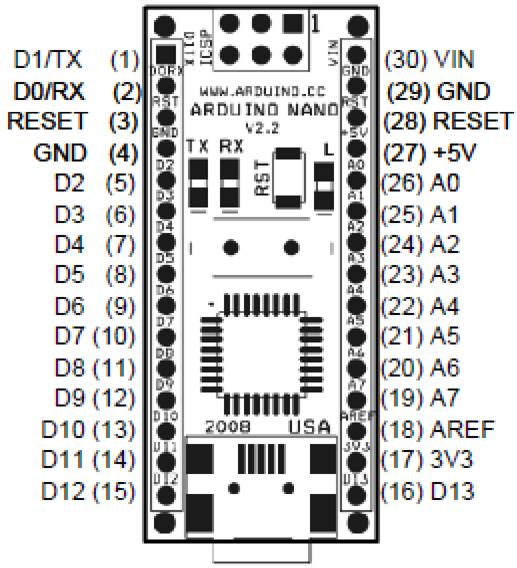

Figure 3.2: Arduino Nano Pin Layout.

Table 3.2: arduino data sheet.

Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino was connected to transmitter circuit to take information from IMU and send it by RF transmitter.

Arduino was connected to receiver circuit to receive information from RF receiver and give command to the driver [6].

3.3.2 IMU (accelerometer)

An accelerometer is a device that measures proper acceleration. Proper acceleration, being the acceleration (or rate of change of velocity) of a body in its own instantaneous rest frame is not the same as coordinate acceleration, being the acceleration in a fixed coordinate system. For example, an accelerometer at rest on the surface of the Earth will measure an acceleration due to Earth's gravity, straight upwards (by definition) of $g \approx 9.81\ m/s^2$. By contrast, accelerometers in free fall (falling toward the center of the Earth at a rate of about $9.81\ m/s^2$) will measure zero. The Structure Concept, an accelerometer behaves as a damped mass on a spring. When the accelerometer experiences an acceleration, the mass is displaced to the point that the spring is able to accelerate the mass at the same rate as the casing. The displacement is then measured to give the acceleration. In commercial devices, piezoelectric, piezoresistive and capacitive components are commonly used to convert the mechanical motion into an electrical signal. Piezoelectric accelerometers rely on piezoceramics (e.g. lead zirconate titanate) or single crystals (e.g. quartz, tourmaline). They are unmatched in terms of their upper frequency range, low packaged weight and high temperature range. Piezoresistive accelerometers are preferred in high shock applications. Capacitive accelerometers typically use a silicon micro-machined sensing element. Their performance is superior in the low frequency range and they can be operated in servo mode to achieve high stability and linearity.

Modern accelerometers are often small micro electro-mechanical systems (MEMS), and are indeed the simplest MEMS devices possible, consisting of little more than a cantilever beam with a proof mass (also known as seismic mass). Damping results from the residual gas sealed in the device. As long as the Q-factor is not too low, damping does not result in a lower sensitivity.

Under the influence of external accelerations the proof mass deflects from its neutral position. This deflection is measured in an analog or digital manner. Most commonly, the capacitance between a set of fixed beams and a set of beams attached to the proof mass is measured. This method is simple, reliable, and inexpensive. Integrating piezoresistors in the springs to detect spring deformation, and thus deflection, is a good alternative, although a few more process steps are needed during the fabrication sequence. For very high sensitivities quantum tunneling is also used; this requires a dedicated process making it very expensive. Optical measurement has been demonstrated on laboratory scale.

Another, relatively new type of MEMS-based accelerometer is a thermal (or convective) accelerometer that contains a small heater at the bottom of a very small dome, which heats the air/fluid inside the dome, producing a thermal bubble that acts as the proof mass. An accompanying temperature sensor (like thermistor; or thermopile) in the dome is used to determine the temperature profile inside the dome, hence, letting us know the location of the heated bubble within the dome. Now, due to any applied acceleration, there occurs a physical displacement of the thermal bubble and it gets deflected off its center position within the dome. Measuring this displacement, the acceleration applied to the sensor can be measured. Due to the absence of solid proof mass, thermal accelerometers yields high shock survival rating.

Most micromechanical accelerometers operate in-plane, that is, they are designed to be sensitive only to a direction in the plane of the die. By integrating two devices perpendicularly on a single die a two-axis accelerometer can be made. By adding another out-of-plane device, three axes can be measured. Such a combination may have much lower misalignment error than three discrete models combined after packaging.

Micromechanical accelerometers are available in a wide variety of measuring ranges, reaching up to thousands of g's. The designer must make a compromise between sensitivity and the maximum acceleration that can be measured. It placed in the transmitter circuit.

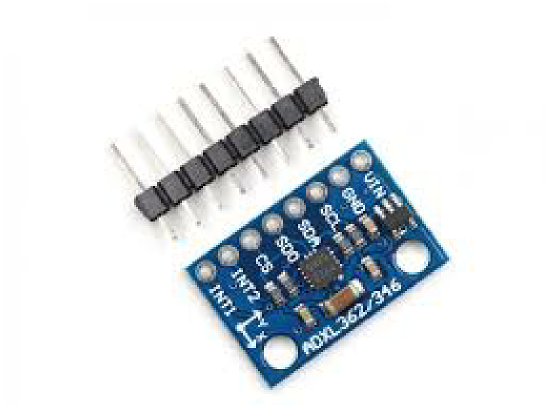

Figure 3.3: Accelerometer.

3.3.3 DC Motor and wheels

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy.

A coil of wire with a current running through it generates an electromagnetic field aligned with the center of the coil. The direction and magnitude of the magnetic field produced by the coil can be changed with the direction and magnitude of the current flowing through it.

A simple DC motor has a stationary set of magnets in the stator and an armature with one or more windings of insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commutator [7]. The DC motor constructed by:

- ❖ Yoke: The outer frame of a dc machine is called as yoke. It is made up of cast iron or steel. It not only provides mechanical strength to the whole assembly but also carries the magnetic flux produced by the field winding.
- ❖ Poles and pole shoes: Poles are joined to the yoke with the help of bolts or welding. They carry field winding and pole shoes are fastened to them. Pole shoes serve two purposes; (i) they support field coils and (ii) spread out the flux in air gap uniformly.
- ❖ Field winding: They are usually made of copper. Field coils are former wound and placed on each pole and are connected in series. They are wound in such a way that, when energized, they form alternate North and South poles.
- ❖ Armature core: Armature core is the rotor of a dc machine. It is cylindrical in shape with slots to carry armature winding. The armature is built up of thin laminated circular steel disks for reducing eddy current losses. It may be provided with air ducts

for the axial air flow for cooling purposes. Armature is keyed to the shaft.

- Armature winding: It is usually a former wound copper coil which rests in armature slots. The armature conductors are insulated from each other and also from the armature core. Armature winding can be wound by one of the two methods; lap winding or wave winding. Double layer lap or wave windings are generally used. A double layer winding means that each armature slot will carry two different coils.
- Commutator and brushes: Physical connection to the armature winding is made through a commutator-brush arrangement. The function of a commutator, in a dc generator, is to collect the current generated in armature conductors. Whereas, in case of a dc motor, commutator helps in providing current to the armature conductors. A commutator consists of a set of copper segments which are insulated from each other. The number of segments is equal to the number of armature coils. Each segment is connected to an armature coil and the commutator is keyed to the shaft. Brushes are usually made from carbon or graphite. They rest on commutator segments and slide on the segments when the commutator rotates keeping the physical contact to collect or supply the current.

It placed in the receiver circuit.

Figure 3.4: Dc Motor and Wheel.

3.3.4 Motor driver

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. Dual H-bridge Motor Driver integrated circuit(IC).

The l293d can drive small and quiet big motors as well. There are 4 input pins for l293d, pin 2,7 on the left and pin 15 ,10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.It placed in the receiver circuit.

Figure 3.5: Motor Driver.

3.3.5 Voltage regulator

A voltage regulator is a system designed to automatically maintain a constant voltage level. A voltage regulator may use a simple feed-forward design or may include negative feedback. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. It drops voltage to fixed value of voltage 5v. It placed in the receiver circuit and transmitter circuit.

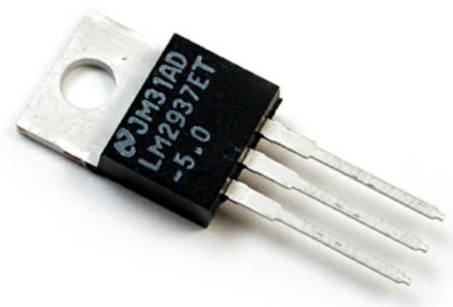

Figure 3.6: Voltage Regulator.

3.3.6 RF Module

An RF module (radio frequency module) is a (usually) small electronic device used to transmit and/or receive radio signals between two devices. In an embedded system it is often desirable to communicate with another device wirelessly. This wireless communication may be accomplished through optical communication or through radio frequency (RF) communication.

For many applications the medium of choice is RF since it does not require line of sight. RF communications incorporate a transmitter and a receiver. They are of various types and ranges.

❖ Transmitter modules:

An RF transmitter module is a small PCB sub-assembly capable of transmitting a radio wave and modulating that wave to carry data. Transmitter modules are usually implemented alongside a micro controller which will provide data to the module which can be transmitted. RF transmitters are usually subject to regulatory requirements which dictate the maximum allowable transmitter power output, harmonics, and band edge requirements.

❖ Receiver modules:

An RF receiver module receives the modulated RF signal, and demodulates it. There are two types of RF receiver modules: super heterodyne receivers and super-regenerative receivers. Super-regenerative modules are usually low cost and low power designs using a series of amplifiers to extract modulated data from a carrier wave. Super-regenerative modules are generally imprecise as their frequency of operation varies considerably with temperature and power supply voltage super heterodyne receivers have a performance advantage over super-regenerative; they offer increased accuracy and stability over a large voltage and temperature range. This stability comes from a fixed crystal design which in the past tended to mean a comparatively more expensive product. However, advances in receiver chip design now mean that currently there is little price difference between super heterodyne and super-regenerative receiver modules. It placed in the receiver circuit and transmitter circuit.

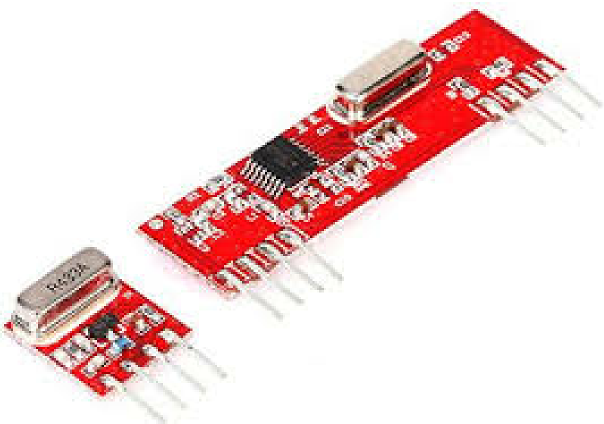

Figure 3.7: RF Module.

3.3.7 Battery

The nine-volt battery, or 9-volt battery, is a common size of battery that was introduced for the early transistor radios. It has a rectangular prism shape with rounded edges and a polarized snap connector at the top. This type is commonly used in walkie-talkies, clocks and smoke detectors.

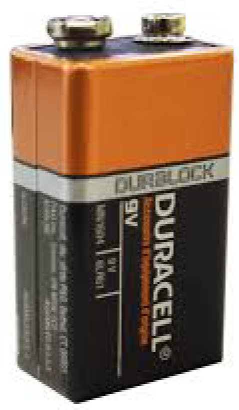

Figure 3.8: Battery.

3.3.8 Wires

These Male/Male Jumper Wires are handy for making wire harnesses or

jumpering between headers on PCB's. These premium jumper wires are long (150mm) and come in a 'strip' of 20 (2 pieces of each of ten rainbow colors). The best part is they come in a 20-pin ribbon cable. You can always pull the ribbon wires off to make individual jumpers, or keep them together to make neatly organized wire harnesses

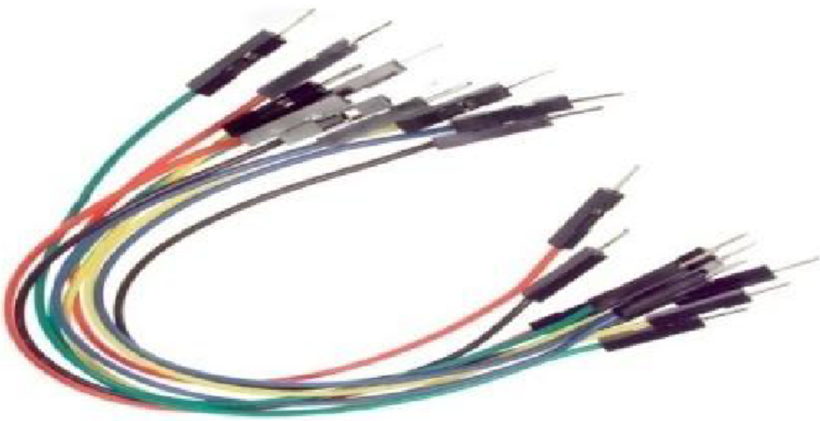

Figure 3.9: Wires.

3.4 Block Diagram of Circuit

The block diagram is shown in figure below.

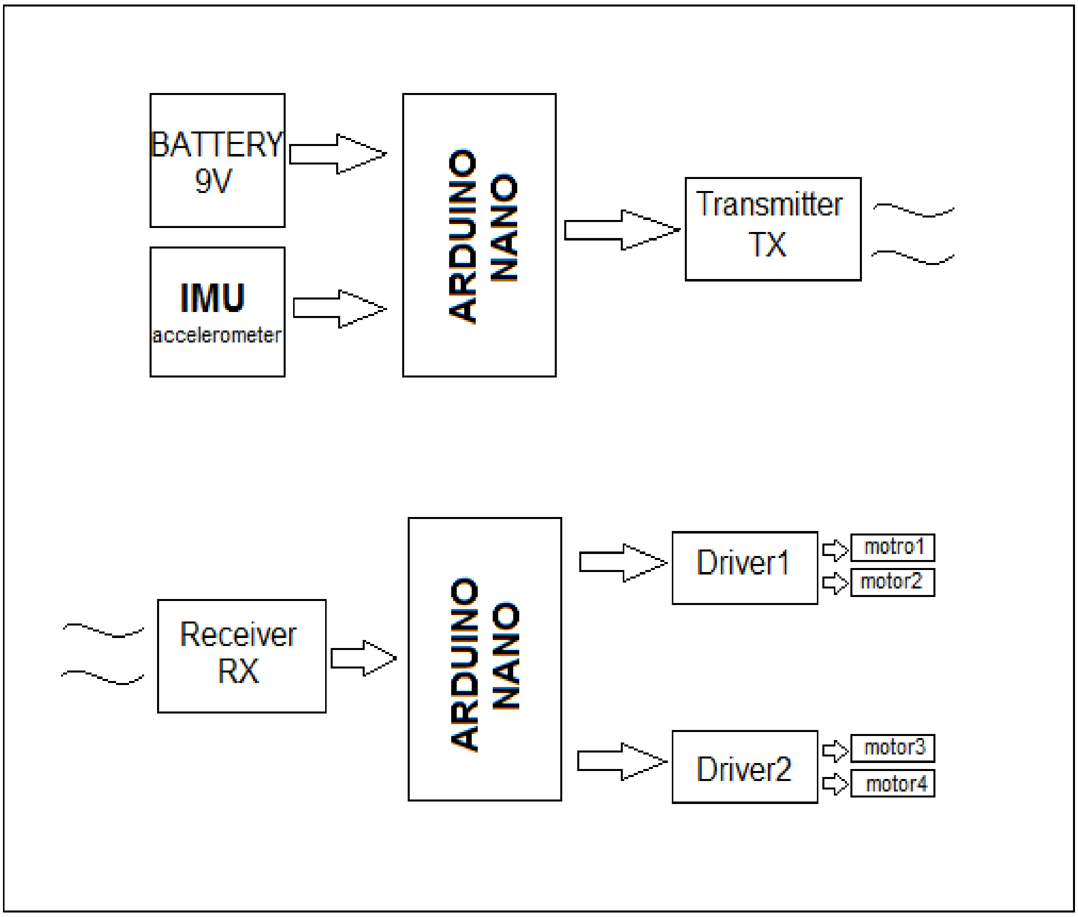

Figure 3.10: Block Diagram.

CHAPTER FOUR
SIMULATION AND OPERATION

4.1 Introduction

To run the circuit that have been designed in chapter three we used microcontroller requires code to do the work, the code have been tested by simulation. Computer simulations have become a useful part of mathematical modeling of many natural systems to observe their behavior. It allows the engineer to test the design before it is built in the real situation. As mentioned earlier, the simulations for this research were performed in PROTEUS program. These software applications are widely used in control engineering for both simulation and design.

4.2 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

It was developed in Yorkshire, England by Lab center Electronics Ltd and is available in English, French, Spanish and Chinese languages.

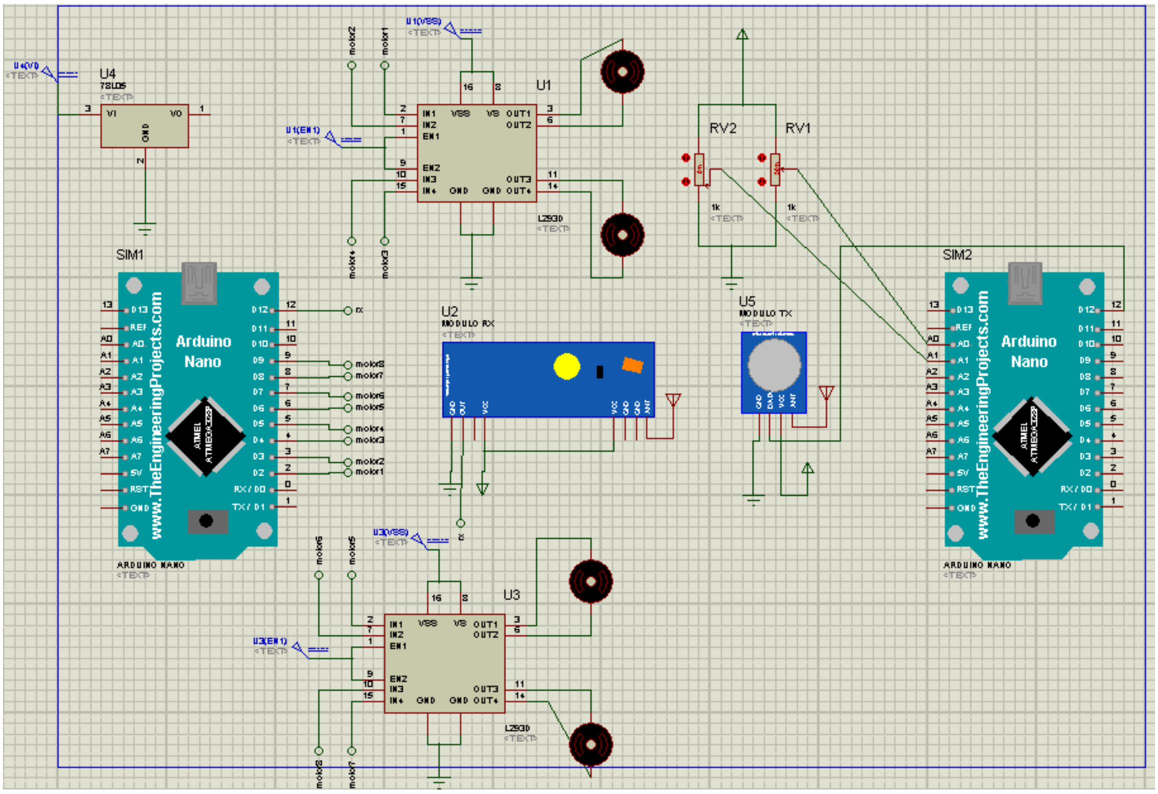

Figure 4.1: Proteus Simulation.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017. Feature led product releases are typically biannual, while maintenance based service packs are released as required.

4.2.1Product module

The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation. All PCB Design products include an auto router and basic mixed mode SPICE simulation capabilities.

4.2.2Schematic Capture

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

4.2.3Microcontroller Simulation

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool. Support is available for co-simulation of:

- ❖ Microchip Technologies PIC10, PIC12, PIC16,PIC18,PIC24,dsPIC33 Microcontrollers.
- ❖ Atmel AVR (and Arduino), 8051 and ARM Cortex-

M3 Microcontrollers

- ❖ NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- ❖ Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- ❖ Parallax Basic Stamp, Freescale HC11, 8086 Microcontrollers.

4.2.4Arduino language

the Arduino language is merely a set of C/C++ functions that can be called from your code. Your sketch undergoes minor changes (e.g. automatic generation of function prototypes) and then is passed directly to a C/C++ compiler (avr-g++). The Four Steps to Writing an Arduino Program are:

- ❖ Define Your Program
- ❖ Declare Each Input/Output (I/O)
- ❖ Create Appropriate Method(s)
- ❖ Call Your Methods in loop()

4.3 Simulation

The simulation was divided into three stages:

4.3.1 Stage one (transmitter)

We has designed the transmitter circuit as shown in fig 4.2. In this circuit we have two potentiometers RV1 and RV2 to represent the motion of hand gestures.

- ❖ RV1 represent the move on forward and backward.
- ❖ RV2 represent the move on left and right.

Then arduino pull this information and send it by module TX to receive circuit.

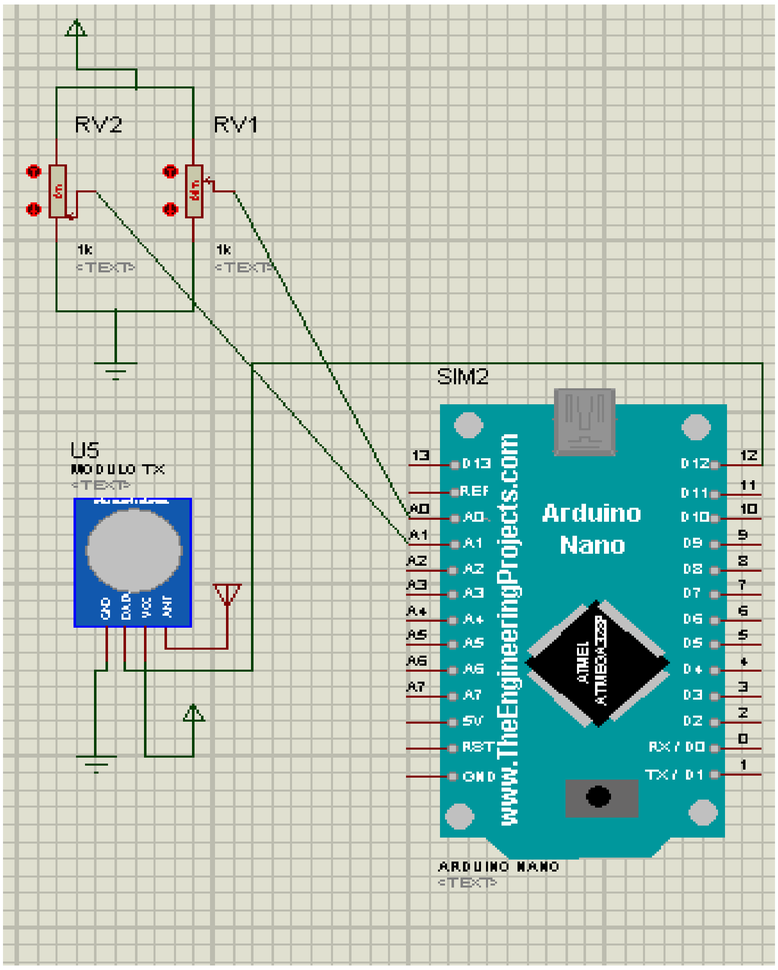

Figure 4.2: Transmitter Circuit.

4.3.2 Stage two (receiver)

We have designed the receiver circuit as shown in fig 4.3. In this circuit we have the module RX to receive the signal from transmitter circuit, arduino to convert the signal of RX to command and give it drivers.

So we have two motor drivers (L293D) to receive command from arduino nano and let the motors move, each one of them drive two motors.

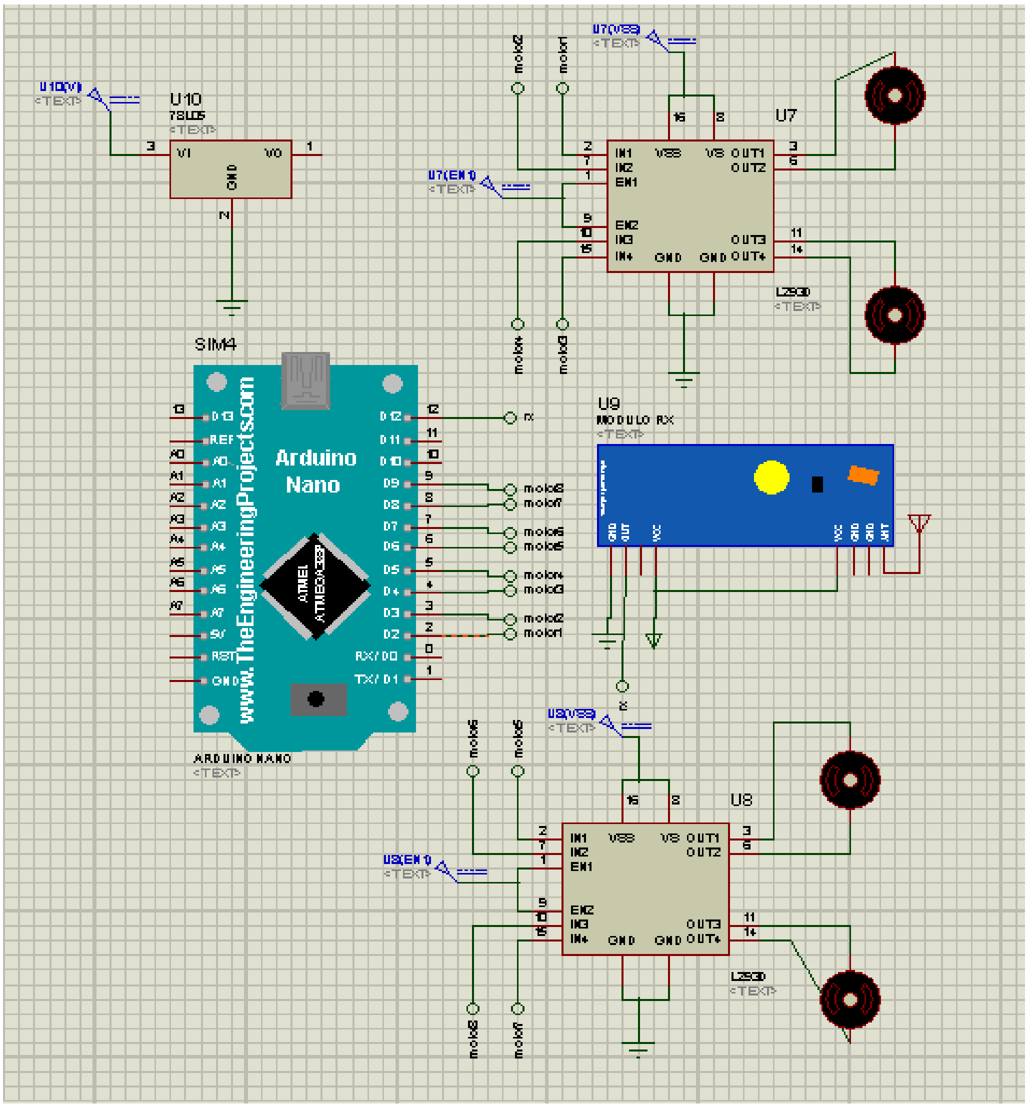

Figure 4.3: Receiver Circuit.

4.3.3 Stage three (operation)

We used potentiometer Instead of accelerometer because we cannot use the IMU in simulation.

4.3.3.1Case one (forward)

In this case we increase the value of RV1 because of that the arduino1 will send a signal to receiver circuit by RF module, the arduino2 receive this signal and convert it to command to driver, then the motor will move forward.

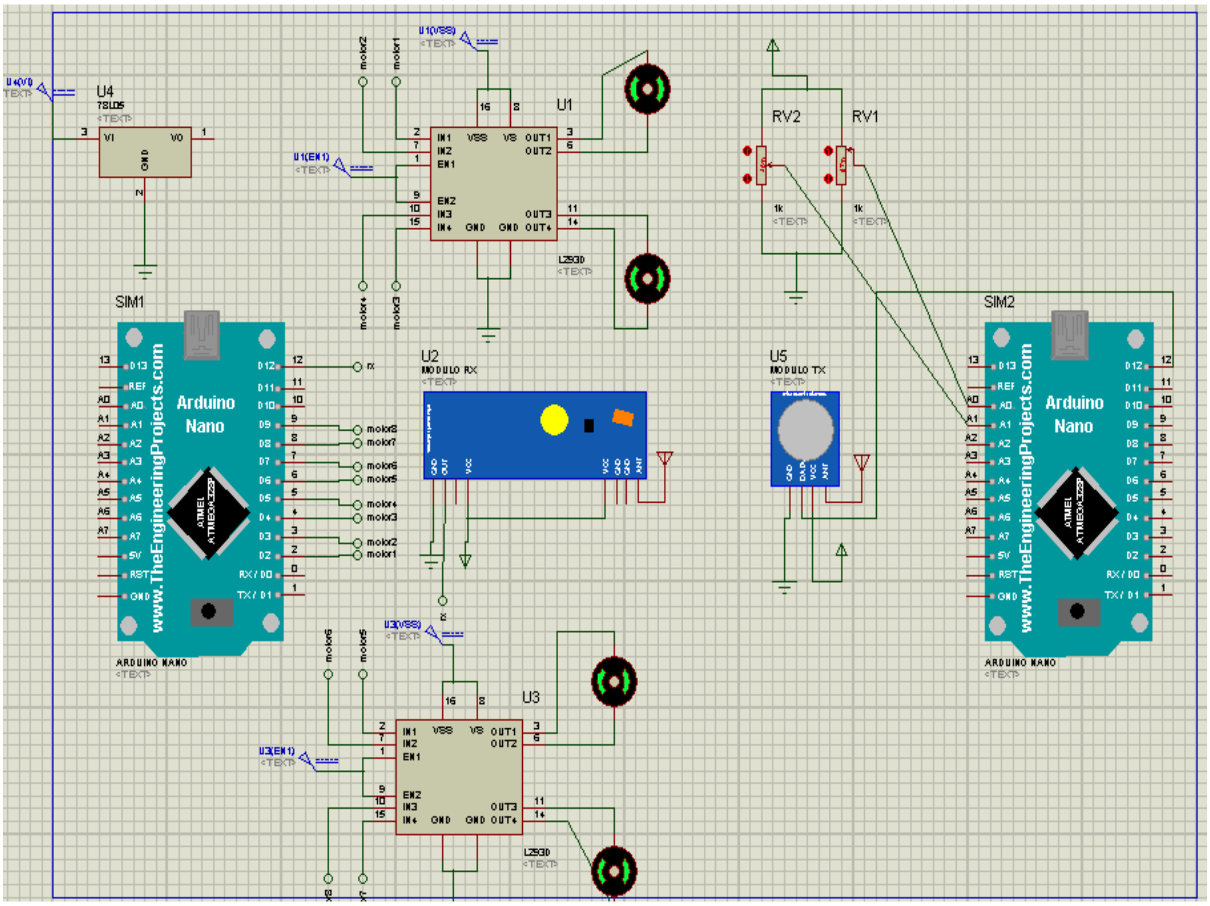

Figure 4.4: Forward Motion.

4.3.3.2Case two (backward)

In this case we decrease the value of RV1 because of that the arduino1 will send a signal to receiver circuit by RF module, the arduino2 receive this signal and convert it to command to driver, then the motor will move backward.

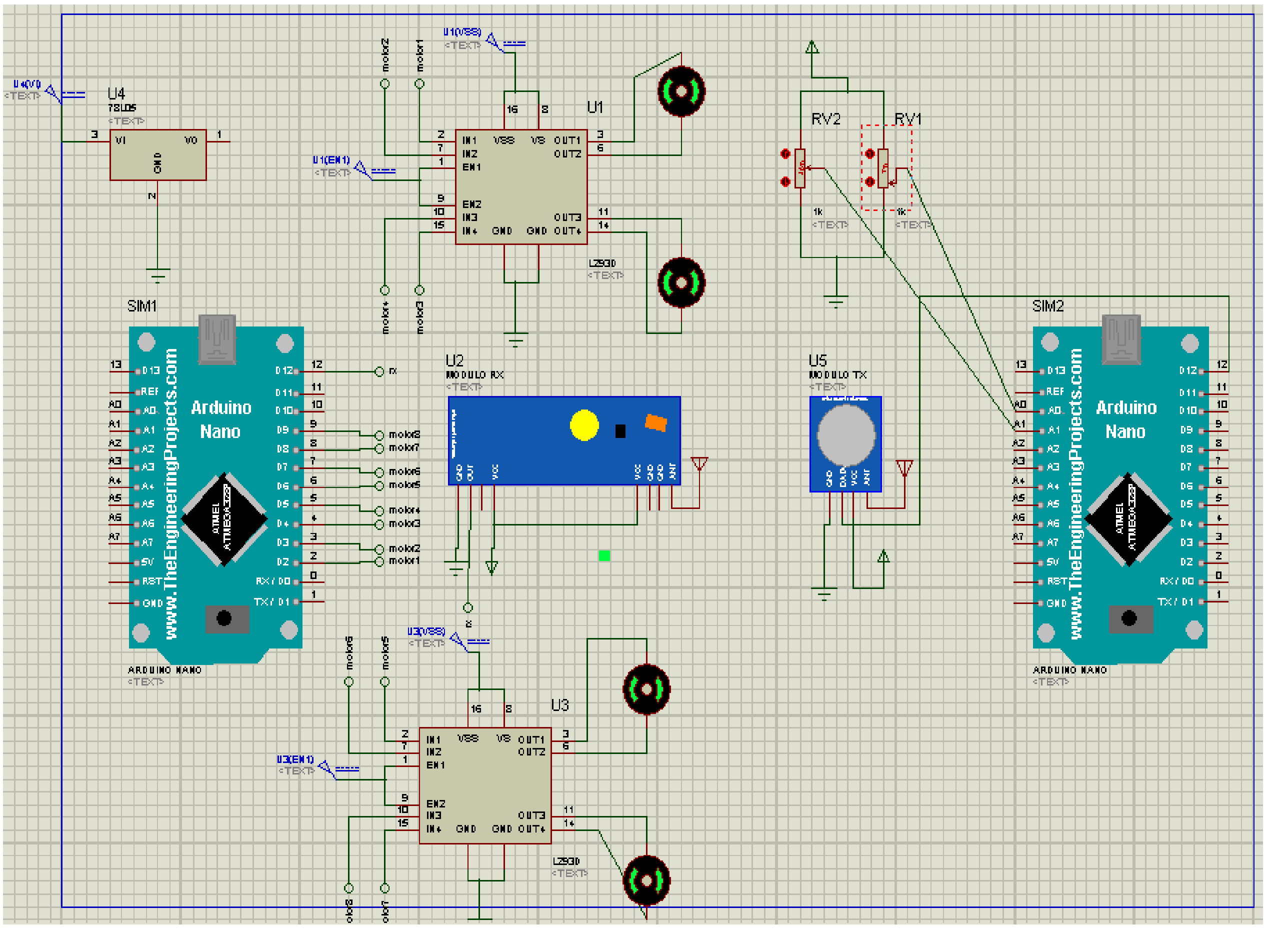

Figure 4.5: Backward Motion.

4.3.3.3Case three (right)

In this case we increase the value of RV2 because of that the arduino1 will send a signal to receiver circuit by RF module, the arduino2 receive this signal and convert it to command to driver, then the motor will move right.

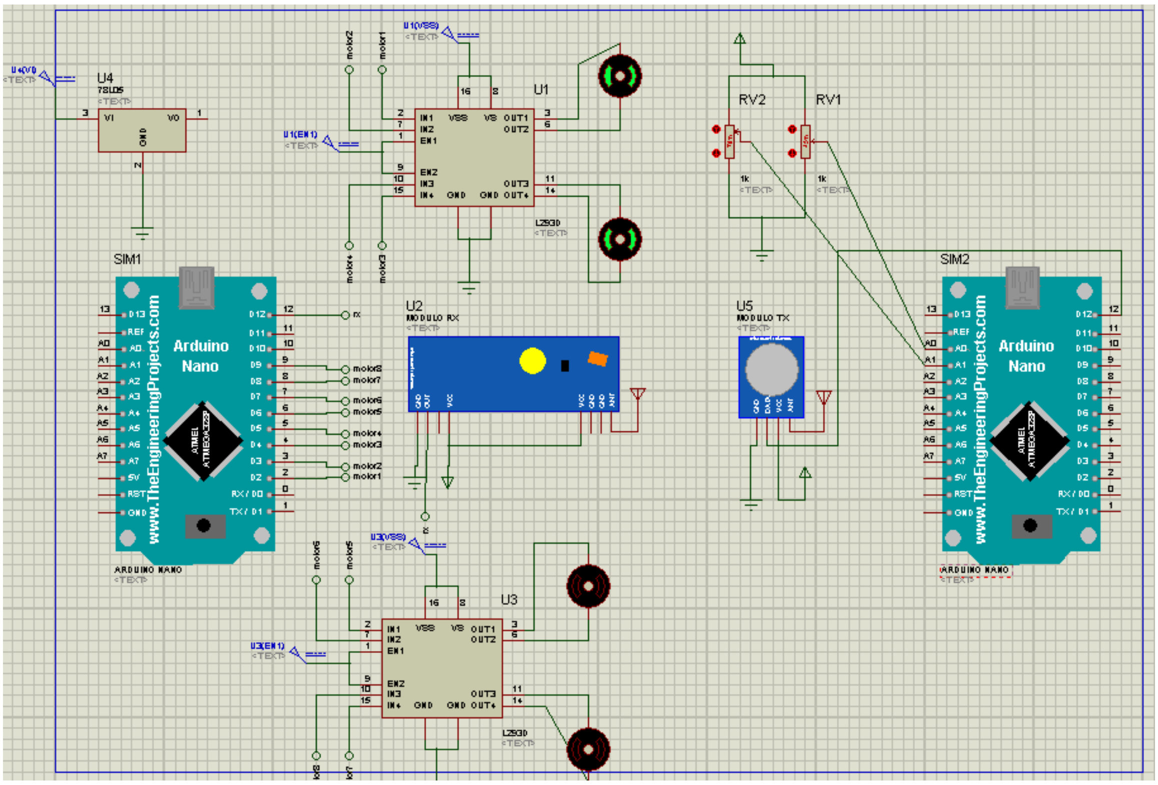

Figure 4.6: Right Move.

4.3.3.4Case four (left)

In this case we decrease the value of RV2 because of that the arduino1 will send a signal to receiver circuit by RF module, the arduino2 receive this signal and convert it to command to driver, then the motor will move left.

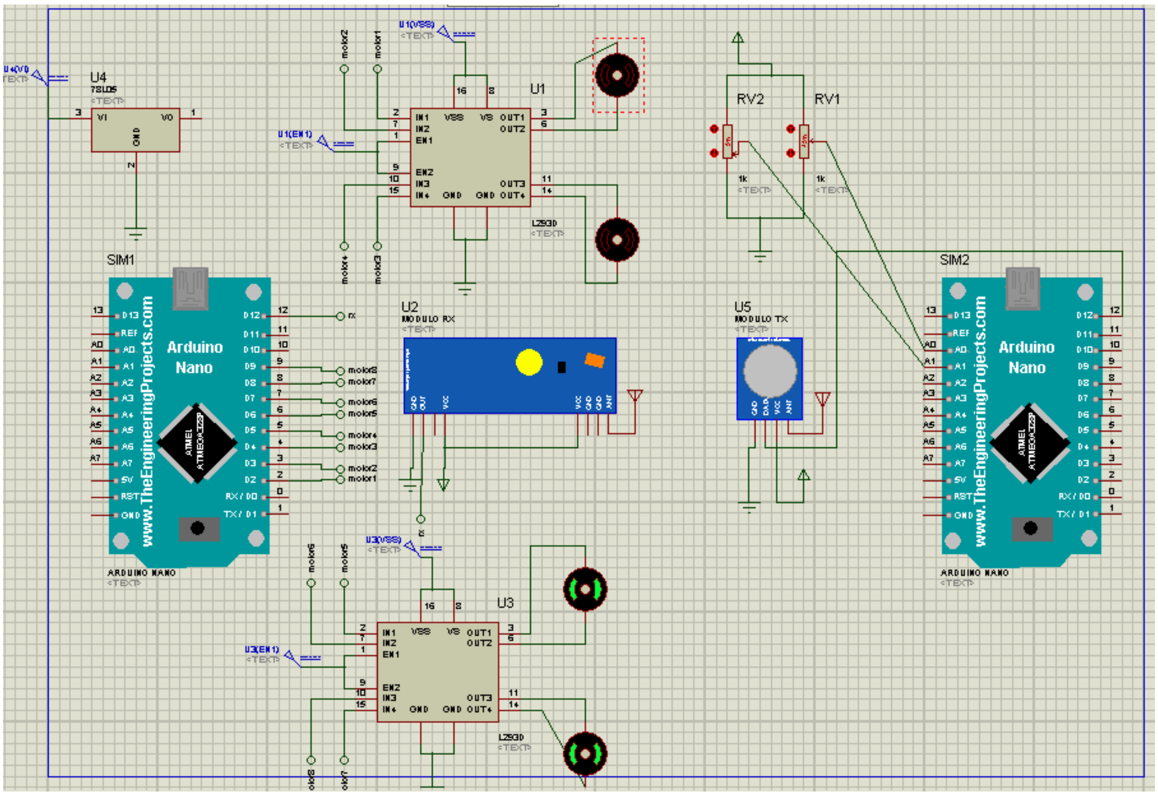

Figure 4.7: Left Move.

4.4 System Flowchart

A flowchart is a type of diagram that represents an algorithm, workflow or process. The steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields, in this thesis the flowchart illustrates the principle of system code as shown in fig 4.8.

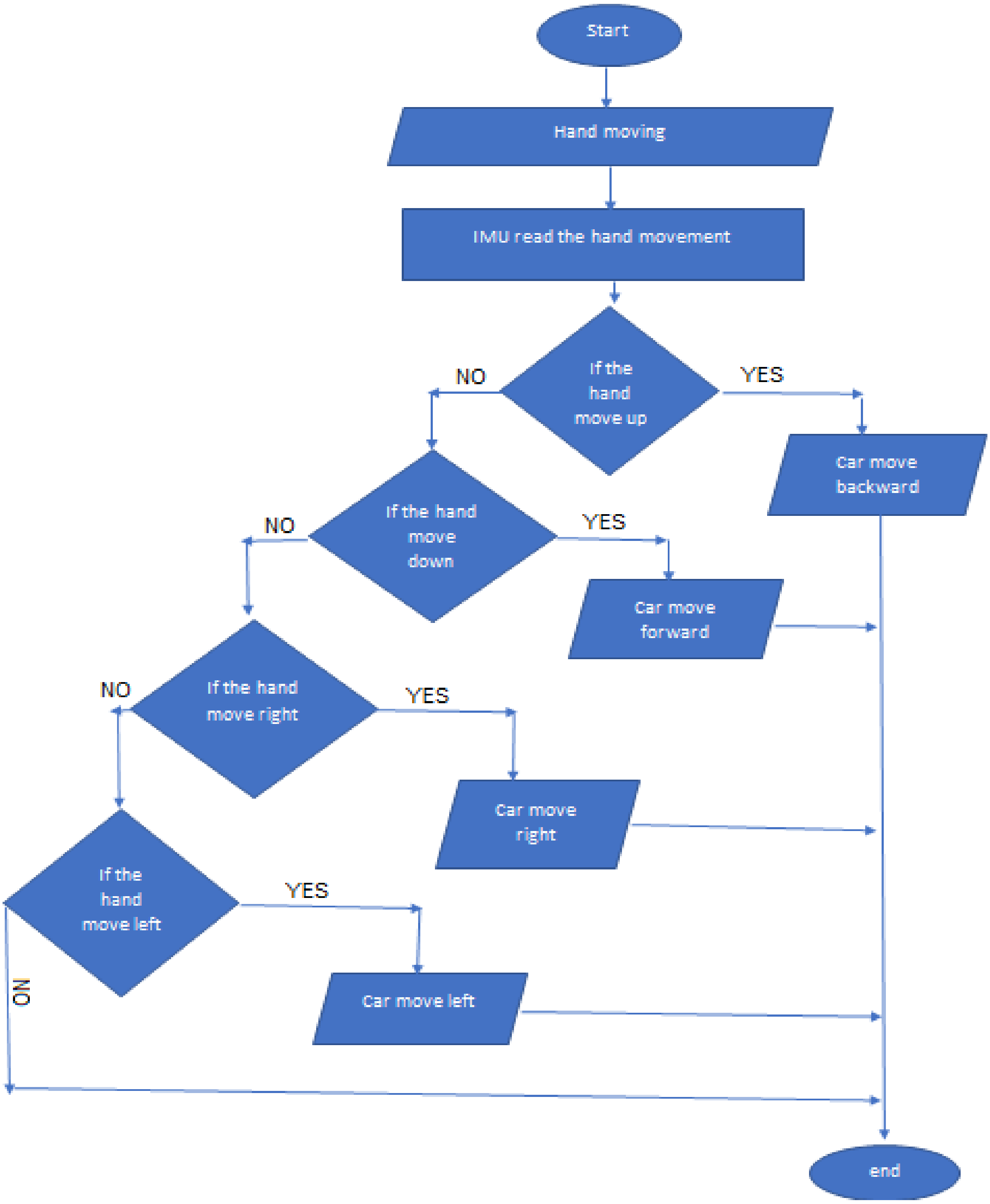

Figure 4.8: Flowchart of project.

4.5 Operation

The operation consists of two parts:

4.5.1 Transmit part

- ❖ IMU unit

The accelerometer measures the acceleration and angular position of hand.

Here we measure pitch to move forward or backward and roll to move right or left.

The connection with the controller by INTER INTEGRATED CIRCUIT protocol

- ❖ Microcontroller unit

Receive the data from IMU by wires to analog pin and process this data and send it to transmitter by wires in digital pin.

- ❖ TX unit

Send data to receiver circuit by frequency that has been equaled to the frequency of RX in receiver circuit.

4.5.2 Receiver part

- ❖ RX unit

Receive the by same frequency data and sent it to microcontroller.

- ❖ Microcontroller unit

Process the data that was receive from RX and inter it in four if loop (forward, backward, right, left).

At each case the microcontroller give the driver command.

❖ Driver (L293D) unit

Here we have two drivers each one of them controlled two motor, instate of that cases the L293D run the suit motor to do right move.

The cases of operation:

❖ Forward

In this case IMU measure pitch and send this data to microcontroller, if the value is greater than specific positive number the controller give command to the drivers to run motors in clockwise.

❖ Backward

In this case IMU measure pitch and send this data to microcontroller, if the value is less than specific negative number the controller give command to the drivers to run motors in anticlockwise.

❖ Right move

In this case IMU measure roll and send this data to microcontroller, if the value is greater than specific positive number the controller give command to the driver 1 to run tow motors in clockwise.

❖ Left move

In this case IMU measure roll and send this data to microcontroller, either value is less than specific negative number the controller give command to the driver 2 to run two motors in clockwise.

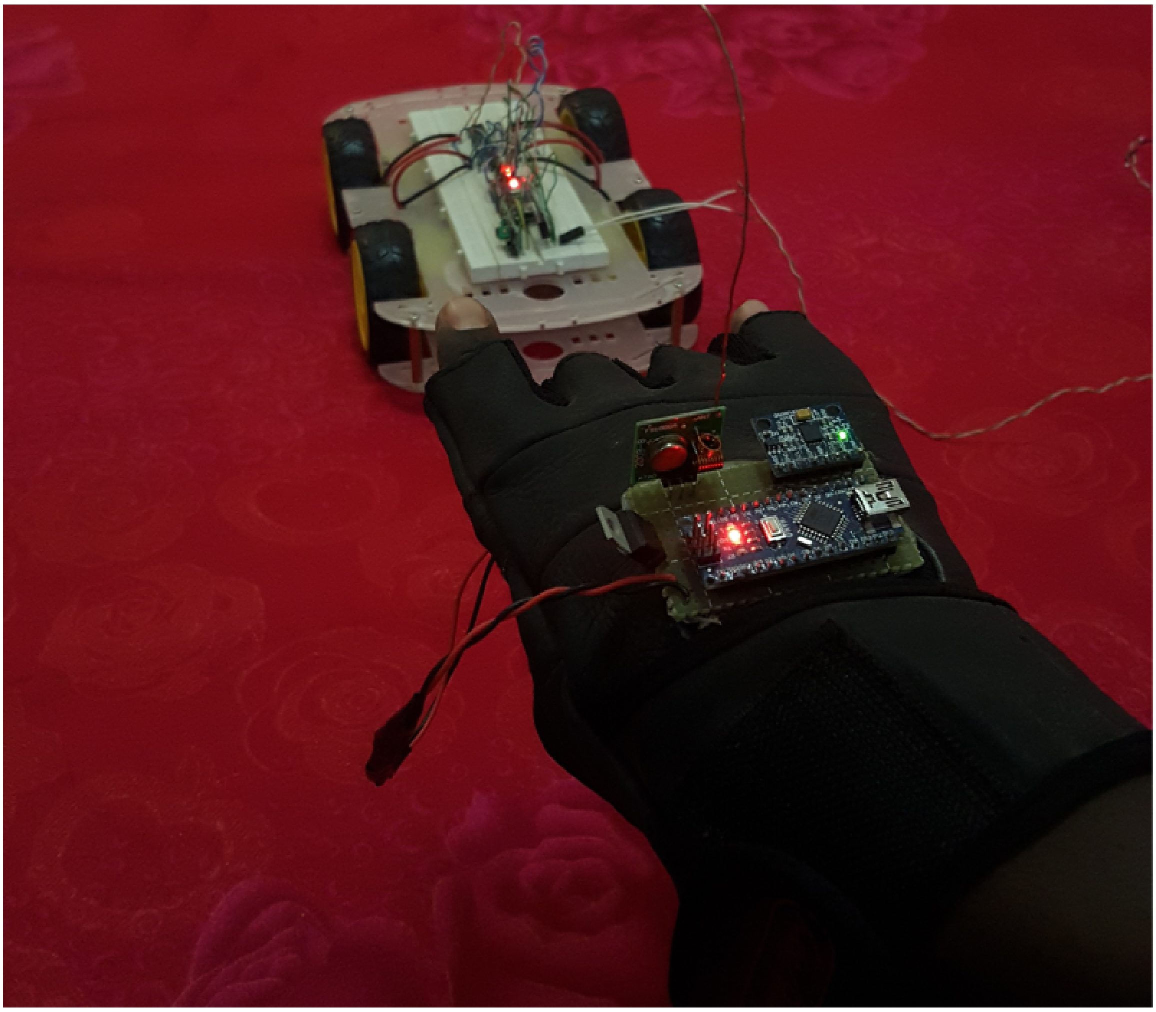

Figure 4.9: The Project.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The model has been designed and the software as well as the control circuit has been implemented successfully. It demonstrates the working of the controlled by hand gesture. Arduino NANO is the most suitable controller whence the cost, case of use, reliable in the work and available in the markets.

5.2 Recommendations

We recommend that:

- ❖ Improve this control system and use it to design motorized wheelchair.
- ❖ The performers can also be improved by installing a camera for exploration and military espionage.

References

[1] Christopher kilian, “Modern control technology”, Cengage Learning, 2005

[2] N. S. Nise, “Control systems engineering”, Hoboken, NJ, 2004.

[3] K. Ogata,” Modern control engineering”, Englewood Cliffs, N.J Prentice-Hall, 1970.

[4] Gunther Gridling, Bettina Weiss, ”Introduction to microcontroller” , in Vienna University of Technology, 2007.

[5] Hai.H.et, “Control Architecture of SY-II Ship Inspection Remote Operated Vehicle”, Computer Science & Service System (CSSS), 2012 International Conference on. 201s2.

[6] Thomas L. Floyd, „„Electronic Devices"", prentice hall, Turin, 2004.

[7] B.L.Theraja, a.k.Theraja, „„A Textbook of Electrical Technology"", S Chand
and Co.Ltd. New Delhi, 2008.

APPENDIX (A)

```
#include <VirtualWire.h>

char *controller;

#include <Wire.h>

const int MPU6050_addr=0x68;


int16_t AccX,AccY,AccZ,Temp,GyroX,GyroY,GyroZ;



void setup()

{

  pinMode(13,OUTPUT);

  vw_set_ptt_inverted(true); //

  vw_set_tx_pin(12);

  vw_setup(4000);// speed of data transfer Kbps

  Serial.begin(9600);


  Serial.println("Initialize MPU6050");
```

```
  Wire.begin();
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
}
void loop()
{
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_addr,14,true);
  AccX=Wire.read()<<8|Wire.read();
  AccY=Wire.read()<<8|Wire.read();
  AccZ=Wire.read()<<8|Wire.read();
  Temp=Wire.read()<<8|Wire.read();
  GyroX=Wire.read()<<8|Wire.read();
  GyroY=Wire.read()<<8|Wire.read();
  GyroZ=Wire.read()<<8|Wire.read();
  // Calculate Pitch & Roll
```

```
  //int pitch = -(atan2(AccX, sqrt(AccY*AccZ +
AccZ*AccY))*100.0)/M_PI;

  int pitch = -(atan2(AccX , AccY)*180.0)/M_PI;

  int roll = (atan2(AccY, AccZ)*180.0)/M_PI;


   Serial.print(" \nPitch = ");

   Serial.print(pitch);

   Serial.print(" Roll = ");

   Serial.print(roll);


  if( pitch > 80  &&  pitch < 100 ){

   controller="f" ;

   vw_send((uint8_t *)controller, strlen(controller));

   vw_wait_tx(); // Wait until the whole message is gone

   digitalWrite(13, HIGH);

   delay(100);

   pitch = 0;

   roll = 0 ;

    }

    else if( pitch < -72 && pitch > -82){

   controller="b" ;
```

```
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(13, LOW);
delay(100);
  pitch = 0;
roll = 0 ;
 }
 else if(roll > 60 &&  roll < 80){
controller="r" ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(13, HIGH);
delay(100);
  pitch = 0;
roll = 0 ;
 }
 else if(roll < -30 && roll > -40){
controller="l" ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx(); // Wait until the whole message is gone
```

```
    digitalWrite(13, LOW);

    delay(100);

      pitch = 0;

    roll = 0 ;

     }



else

{


   controller="s" ;

      vw_send((uint8_t *)controller, strlen(controller));

      vw_wait_tx(); // Wait until the whole message is gone

      delay(100);

      digitalWrite(13, LOW);

        pitch = 0;

    roll = 0 ;


}

  delay(10);
```

```
}
```

APPENDIX (B)

```
//connection for rf receiver
//GND = GND
//DATA = 12
//VCC = 5V
#include <VirtualWire.h>
//connection for motors
void setup()
{
vw_set_ptt_inverted(true); // Required for DR3100
vw_set_rx_pin(12);
vw_setup(4000); // Bits per sec
//pinMode(13, OUTPUT);
  // set all the motor control pins to outputs
  Serial.begin(9600);
vw_rx_start(); // Start the receiver PLL running
```

```
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
   pinMode(6,OUTPUT);
   pinMode(7,OUTPUT);
   pinMode(8,OUTPUT);
   pinMode(9,OUTPUT);
}
void loop()
{
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = 2;
if (vw_get_message(buf, &buflen)) // Non-blocking
{
  Serial.print(buf[0]) ;
  Serial.println(buf[1]);
  if(buf[0]=='f'){
  Serial.print("forward");
  Serial.println(buf[1]);
```

```
digitalWrite(2,1);
 digitalWrite(3,0);
  digitalWrite(4,1);
   digitalWrite(5,0);
    digitalWrite(6,1);
     digitalWrite(7,0);
      digitalWrite(8,1);
       digitalWrite(9,0);
  }
  if(buf[0]=='b')  {
  Serial.print("backward");
  Serial.println(buf[1]);
digitalWrite(2,0);
 digitalWrite(3,1);
  digitalWrite(4,0);
   digitalWrite(5,1);


    digitalWrite(6,0);
     digitalWrite(7,1);
```

```
      digitalWrite(8,0);
       digitalWrite(9,1);


}
  if(buf[0]=='s'){
  Serial.print("stop");
  Serial.println(buf[1]);
digitalWrite(2,0);
 digitalWrite(3,0);
  digitalWrite(4,0);
   digitalWrite(5,0);
    digitalWrite(6,0);
     digitalWrite(7,0);
      digitalWrite(8,0);
       digitalWrite(9,0);


}
if(buf[0]=='l'){
  Serial.print("left");
  Serial.println(buf[1]);
```

```
digitalWrite(2,0);
 digitalWrite(3,0);
  digitalWrite(4,1);
   digitalWrite(5,0);
    digitalWrite(6,1);
     digitalWrite(7,0);
      digitalWrite(8,0);
       digitalWrite(9,0);
}
if(buf[0]=='r'){
  Serial.print("right");
  Serial.println(buf[1]);


digitalWrite(2,1);
 digitalWrite(3,0);
  digitalWrite(4,0);
   digitalWrite(5,0);
    digitalWrite(6,0);
     digitalWrite(7,0);
```

```
digitalWrite(8,1);
 digitalWrite(9,0)}}}
```