Head Gesture Controlled UGV

By

SGT HAMMAD AHMED

GC MUHAMMAD ABDULLAH

GC GOHAR ABBAS

Submitted to the Faculty of Electrical Engineering, Military College of Signals, National University of Science and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E. Degree in Electrical (Telecom) Engineering

JUNE 2015

<u>CERTIFICATE OF CORRECTNESS AND APPROVAL</u>

It is certified that the work contained in this thesis "HEAD GESTURE CONTROLLED UGV", was carried out by Hammad Ahmed, Muhammad Abdullah and Gohar Abbas under the supervision of Lec. Moiz Ahmed Pirkani for partial fulfillment of Degree of Bachelor of Telecommunication Engineering, is correct and approved.

Approved by

____

Lec. Moiz Ahmed Pirkani

EE Department

MCS

Dated:____________________

ABSTRACT

Our Project is based on wireless communication between the controller and UGV. We are operating our UGV by user’s head gesture as well as it can be operated in autonomous mode. The whole thing can also be controlled by an Android application when required. We have done the hardware as well as software work for controlling our UGV. We have used Arduino boards for the design of both UGV and the headset controller which detects the head gesture and accordingly move the UGV. The UGV will be able to move in all possible directions. There is a gun mounted on the top of the UGV. This gun can also be controlled by same head gesture detector headset as well as with Android application. The trigger is also controlled by headset and Android app. The gun mounted on the UGV, can move in all 4 directions horizontally and vertically. The gun used is Auto Reloaded.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Gracious, the Most Beneficent,

Dedicated to our parents, our teachers and especially Military College of Signals.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are grateful to our parents who supported us both morally and financially. We thank our friends for their admirable support and their critical reviews.

Due extension of gratitude to our project supervisor Lec. Moiz Ahmed Pirkani for his continuous technical guidance and moral support throughout the project. His calm nature ensured smooth work and his composed manner helped us work around obstacles with perseverance.

Table of Contents

Table of Figures

List of Abbreviation

UGV	Unmanned Ground Vehicle
D.C	Direct Current
IR	Infrared
LED	Light Emitting Diode
I/O	Input/output
RF	Radio Frequency

CHAPTER 1

1.1 Introduction

Our Project is based on wireless communication between the controller and UGV with a gun on it. We have operated our UGV and gun by users head gestures and by an Android application when required. We have worked on making the hardware as well as software for controlling our UGV. We have used Arduino board for the design of both UGV and the headset controller which will detect the head gesture and accordingly move the UGV. The UGV will be able to move in all possible directions. These same head gestures are used to move to gun in 360 degrees on horizontal scale and 0-90 degree on vertical scale.

1.2 Project Motivation

1.2.1 Previous Problems

The UGV's which we had seen in past could not be moved autonomously and to control them people would use remote controlled modules and big joysticks. It is difficult to use and carry big sized modules esp. in battle fields and remote areas. A person having no knowledge of how to use these modules cannot start using them at once and it requires a lot of time to master them.

1.2.2 Problem Solution

Keeping the above problem in our mind we try to introduce Headset controller which is easy to carry, smaller in size and user friendly. Our main goal is to design a UGV that can be controlled through a person's head gestures and secondly can also be moved autonomously or by an Android application. With the help of a simple head set controller and android app controlling a UGV becomes simple and easy.

1.3 Block Diagram

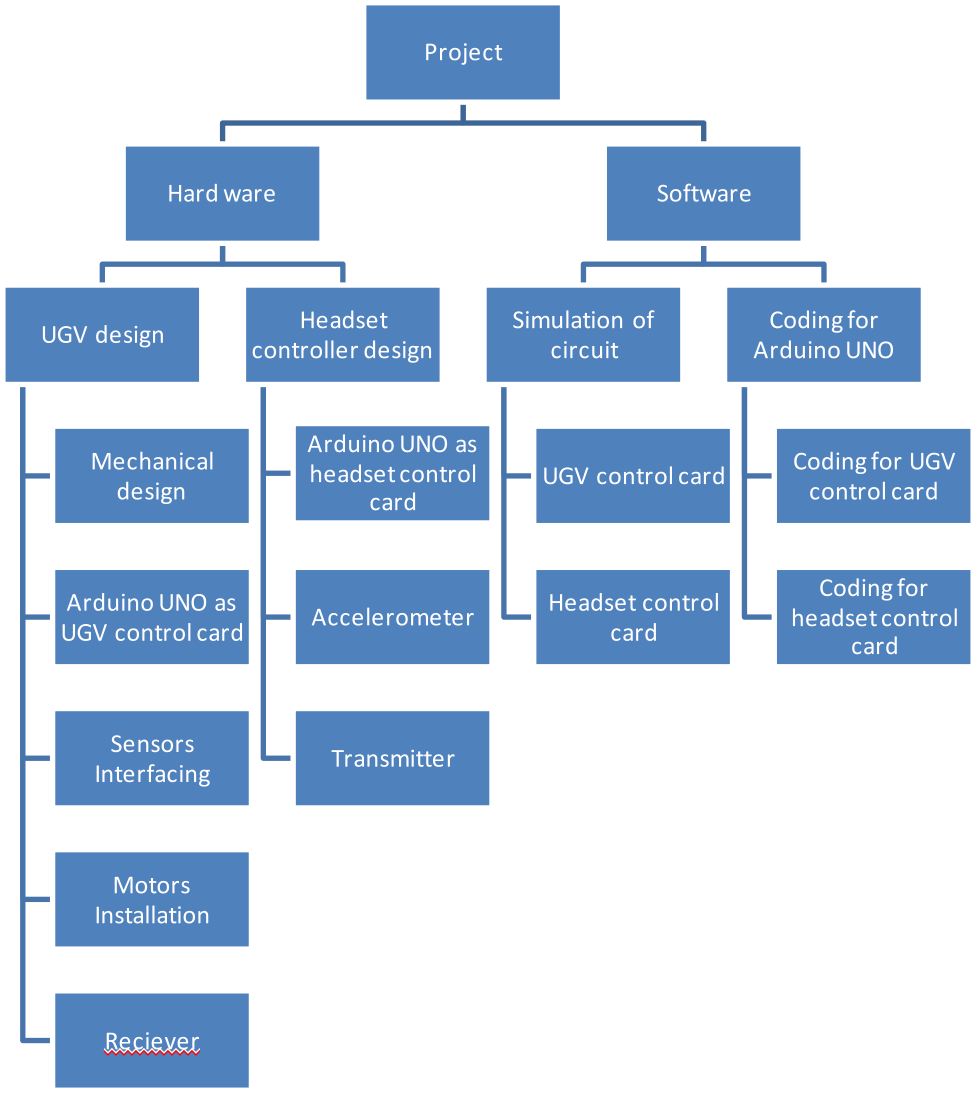

Figure 1: Block Diagram

1.4 SALIENT FEATURES

Our UGV has got following salient features:

1. UGV is mechanical based and light weight. It is cable of moving in rough terrains.
2. There is a gun that is installed on top of UGV.

3. We have installed Arduino UNO, Arduino NANO, Motors, and Batteries, Bluetooth module, RF Transmitter/Receiver and more.
4. Our UGV will move autonomously. We have done this by installing sensors for obstacle detection.
5. A Headset controller is designed which can recognize the head gestures of the person wearing it using Arduino UNO board and an accelerometer.
6. We have established wireless communication between the headset controller and the UGV using RF transmitter / receiver pair or RF transceivers. This enables the two Arduino board to communicate serially.
7. We have written an algorithm which converts the head gestures into commands that are recognized by the UGV and this enables the UGV to be controlled by head gestures.

1.5 Scope, Objectives and Deliverables

1.5.1 Scope

The scope of our project is briefed as follow:

1. Through Head Gestures controlling a UGV becomes easy and simple. This UGV can be mastered through less training.

2. Disable persons or soldiers with a lasting injury which means they're unable to carry out their duties can control this UGV easily using head gestures.

3. This UGV can be used for defense and surveillance purpose.

4. Autonomous mode in this UGV provides full control to UGV with ditch and obstacle detection and avoidance.

5. We can also apply this head gesture technology in other projects e.g. Head Gesture controlled electrical wheel chair can be designed for disabled persons through few modifications to this project.

1.5.2 Objectives

Our objectives are mentioned below:

1. To design a UGV.
2. To make it autonomous.
3. To make a headset controller which can recognize the head gestures.
4. To establish wireless communication between the headset and the UGV.
5. To control UGV movement through this headset.
6. To interface an android application with the UGV.
7. To move the gun mounted on the UGV to move in all directions.
8. To press the trigger of the gun wirelessly.

1.5.3 Deliverables

Our project deliverables are:

1. To make the UGV which can move in difficult terrains.
2. Controlling the UGV by users head gestures using a specially designed headset.
3. Wireless control of UGV using head gestures.
4. The UGV moves in four directions (forward, backward, left and right) at any angle the user wants.
5. UGV will be armed with a gun mounted on top of it which can move in 360 degree in circle and 90 degree up and down and all these motions must be control by headset controller that recognizes the users head gestures.
6. UGV has also autonomous mode the user can put it in autonomous mode with a click of a button on headset controller then the UGV will move automatically in autonomous mode the UGV will be able to avoid obstacles in its way.
7. The whole UGV and gun operations can be controlled by an Android application.

CHAPTER 2

LITERATURE REVIEW

It was necessary to review the literature as then later on we might have problems regarding which module/equipment we have to use for specific purposes .going through many different articles about different component that we will use in the project we summaries them and note all the main points that we think are useful in our project.

The major source of information was the internet as it is the easiest source available for getting large amounts of relevant information just in few seconds.

We also gave a reading to the thesis reports which were written by the old students of our university who had also worked on robotics.

2.1 History of UGV's

The UGV (Unmanned Ground Vehicle) as the name suggests was an unmanned device that operated being in contact with the ground. A UGV can be used for many different applications [1]. A UGV has a set of sensors to see what is going around them or will have the capability of moving autonomously when required.

Figure 2: Old Version of Military UGV

The first UGV's were developed in 1930's in USSR; they developed Teletanks, a tank which is controlled by a radio, which is present in another tank. Such UGV's were used in war of 1939-1940 against Finland. A radio controlled version was developed by some British people in 1941 during World War2. These UGV's were not that reliable and so efficient methods had to be found to bring it up to the standard. During the 1960's, the first reliable and up to the mark UGV was developed which was called Shakey.

2.2 Older Versions of UGV's

Basically, the UGV's are divided into two classes joystick/remote Controlled and autonomous. A joy stick controlled UGV is the one in which a person is controlling its movement through a module. All the actions are it performs depends on the operator while in the autonomous one no operator is needed. The UGV uses its sensors to detect the path it is going to follow plus some control algorithm is needed for the proper and secure movement of the UGV [2].

2.3 Examples of old UGV's

2.3.1 The Pack Bot

It is the kind of UGV which was in use in early 2004. Its main purpose was space exploration but was also used for military purposes. PackBot does not have the capabilities of many robots being used today e.g. backtracking to its operator. Even though this design has some flaws but still it is used for many purposes e.g. exploration etc. It has a Hazmat sensor which also enables it to detect poisonous and toxic chemicals [3]. It is a portable device which could even be carried in your backpack.

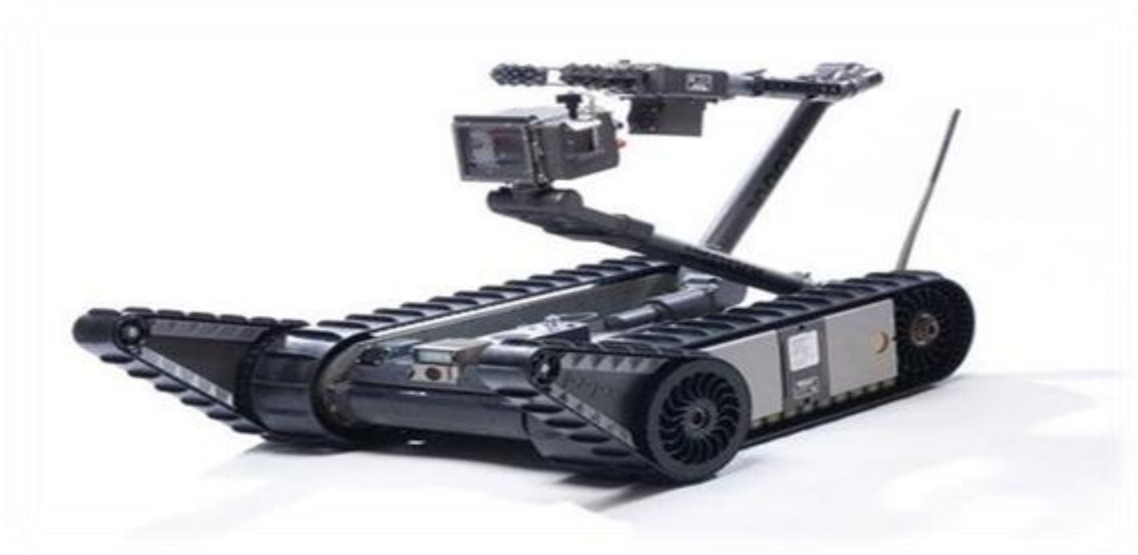

Figure 3: The Packbot UGV

2.3.2 The Gladiator

The size of this system is not very big and its prime purpose is to save Marine Corps. It has the capability of developing direct fire weapons [4]. The best thing about this UGV is that it is extremely durable.

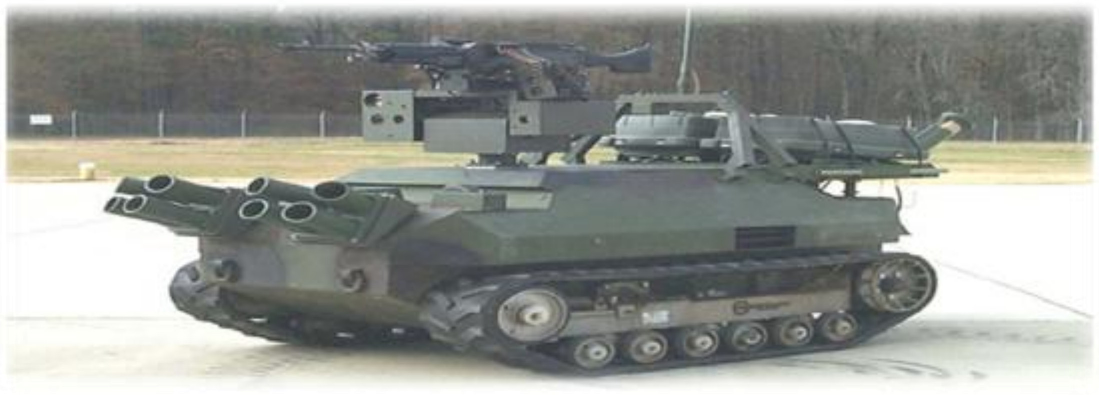

Figure 4: The Gladiator

2.3.3 The Sarge

It is a four wheeled vehicle and has the frame like Yamaha Breeze. Its Prime Purpose is remote surveillance which is sent for the purpose of investigation [5].

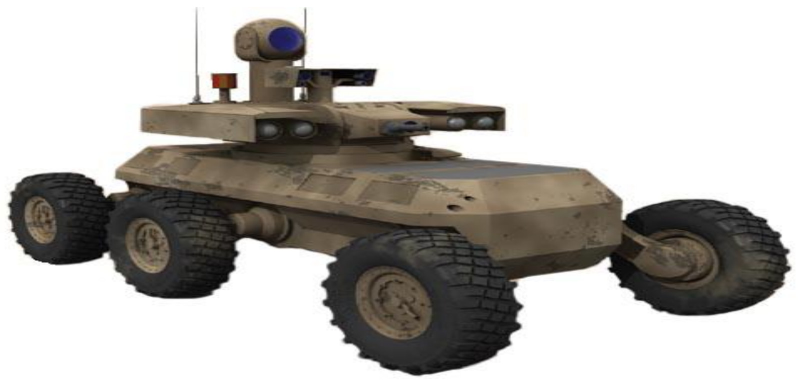

Figure 5: The Sarge UGV

2.4 Gesture Based UGV's

Gesture based UGV's are the need of the day, they can be used for many different purposes e.g. medical, automobile, defense, construction etc.

Previously many people have worked on UGV's which are being controlled through different gestures lots of work is done on controlling a UGV through a person's arm and hand gestures. Two main components were used in majority of the projects e.g. A PIC microcontroller and an accelerometer. We just need to wear the hand set device on which the accelerometer is interfaced with the microcontroller and the UGV is in our control.

Hand gestures can be classified into two approaches

1. Rule based approach.
2. Machine learning based approach

To find an effective way to communicate with the UGV's is extremely important in today's world. As it is becoming more and more common, gestures using fingers is another way of communicating with the UGV.

Head gesture based technology is another increasing interest in people these days and a lot of work is also being done. The recognized gestures are a source of generating motion. It is a human friendly interface and is very easy to use. These are extremely useful systems for the users who have restricted limb movements caused by some diseases such as Parkinson's disease and quadriplegics

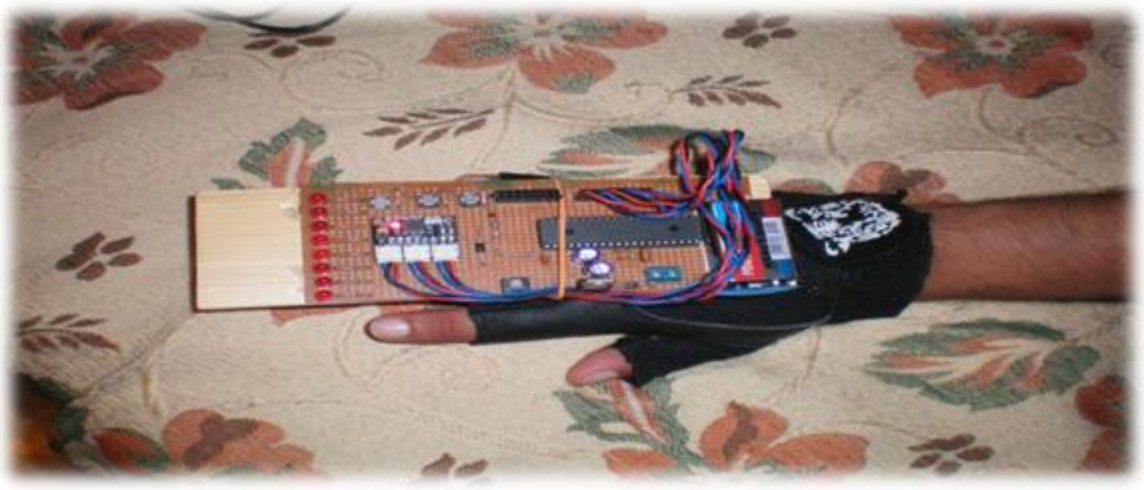

Figure 6: The Handset Controller

2.5 Autonomous Robots

An Autonomous robot performs behaviors or tasks with a high degree of autonomy, which is particularly desirable in fields such as space exploration, cleaning floors, mowing lawns, and water treatments. One important area of robotics research is to enable the robot to cope with its environment whether this be on land, underwater, in the air, underground, or in space.

A fully autonomous robot has the ability to

1. Gain information about the environment (Rule #1)
2. Work for an extended period without human intervention (Rule #2)
3. Move either all or part of itself throughout its operating environment without human assistance (Rule #3)

There are several open problems in autonomous robotics which are special to the field. Problems include things such as making sure the robot is able to function correctly and not run into obstacles autonomously, or fall into a ditch. Secondly if present in a situation where it has to work, it should be reliable enough.

2.6 Android controlled UGV's

Android based robotics project are very powerful, flexible and user friendly. A growing interest in having smart phones interacting with peripheral devices such as motors, servos and sensors led to the recent creation of electronic interface boards that can be purchased online or built at a small cost. These boards serve as communication bridges between Android smart phones and external devices.

Many Android applications have been made which also allow communication and are being used for many such purposes. Smartphone robots and applications are becoming increasingly prevalent in research projects.

2.7 Gesture + Autonomous + Android Controlled UGV's

There are hardly any UGV's made previously which work on the three of the ways. We will find a UGV which either works by detecting gestures or autonomously or by the android application, finding three of the methods to control a single UGV is not that common. The proposed UGV which will operate in three modes which will depend on what the user wants. These three methods of controlling a UGV are also interchangeable, this means that if we want the base of the UGV to be controlled through our head gestures and the gun mounted on the top to be controlled by the android application that is also possible and Vis versa.

The UGV that we have designed is very versatile and can be used for many different purposes where older UGV's were not that reliable. For example If we are in a certain area where controlling the UGV through head gesture's is not possible we will utilize the other two methods of operation that we have.

2.8 Arduino Uno:

2.8.1 Hardware:

(The Arduino Uno),which is a microcontroller board based on the ATmega328. The board consists of 14 digital input/output pins. Out of these 6 pins can be used as PWM outputs and 6 can be used as analog inputs. Other components include a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header plus a reset button. It already consists of everything so no external connections are needed, we can easily connect it to a computer with a USB cable or through an AC-to-DC adapter or even a battery to start it.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with other components to help in programming and incorporation into other circuits. One of the most important aspect of the Arduino uno is the easy way that connectors are exposed, which allows the CPU board to be connected to a variety of interchangeable modules called shields.

The Arduino board exposes almost all the microcontroller's input and output pins for use by other circuits.

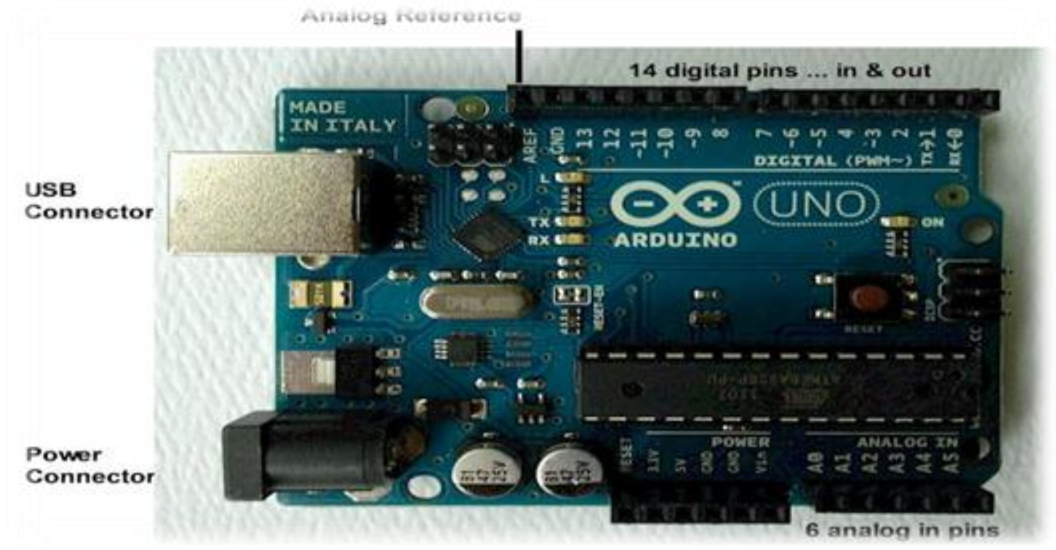

Figure 7: Arduino Uno

2.8.2 Summary:

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 (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

2.8.3 Power:

The Arduino Uno is powered through the USB connection or through some external power supply. Automatically the power source is connected. External it can be powered by an AC-to-DC adapter or a battery.

2.8.4 Input/Output:

All the arduino uno pins (14 pins) can be used as input or output, using functions such as pinMode(), digitalWrite(), and digitalRead() . The operating voltage is 5 V. Each pin can provide or receive a current upto 40 mA and the internal pull-up resistor is 20-50 kOhms.

2.8.5 Communication:

The Arduino Uno has a lot of facilities available for communicating with a computer and other microcontrollers

The Software Serial library allows serial communication on any of arduino uno's digital pins.

2.8.6 Programming:

The Arduino software is available for the programming of arduino uno.

2.8.7 Automatic software reset :

There is no physical button present for resetting it, the Arduino Uno is designed in such a way that by running software on a computer, we can reset it.

2.8.8 USB Over current protection:

On an Arduino Uno a resettable polyfuse button is present that protects the computer's USB ports from getting short and over current conditions. Almost every computer has its own internal protection; the fuse provides an extra protection. If around 525 mA is applied to the USB port, the fuse will break the connection at once until the overload is finished.

2.9 Arduino Nano:

We will be using an arduino nano in our head set controller. we have chosen this over arduino uno to be used in our head set controller because it less in weight and smaller in size.

The Arduino Nano is a much smaller, comprising of everything, and user-friendly board. It is based on the ATmega328 or ATmega168. It does not have a DC power jack, it works with a Mini-B USB cable and not a standard one.

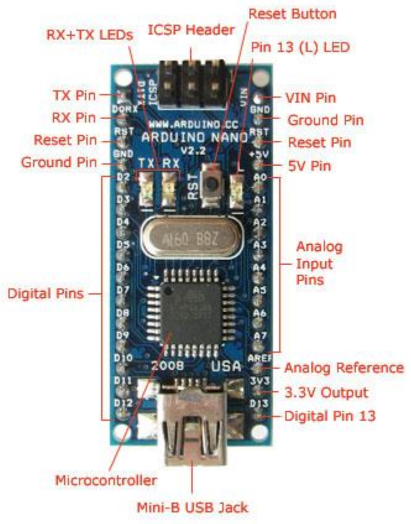

Figure 8: Arduino Nano

2.9.1 Power:

The Arduino Nano is powered by the Mini-B USB connection, An unregulated external power supply at pin no: 30, or with a 5V regulated external power supply at pin no: 27. Automatically the power source is selected to the highest voltage source.

2.9.2 Memory:

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

2.9.3 Input/Output:

There are in all 14 digital pins present on the Nano.They can be used either as input or output, using functions such as pinMode(), digitalRead(), and digitalWrite() . The operating voltage is 5 V. Each pin can provide or receive a current upto 40 mA and the internal pull-up resistor is 20-50 kOhms.

2.9.4 Communication:

The Arduino Nano has a lot of facilities available for communicating with a computer and other microcontrollers.

The Software Serial library allows serial communication on any of arduino Nano's digital pins.

2.9.5 Programming:

The Arduino software is available for the programming of arduino Nano.

2.9.6 Automatic Software Reset:

There is no physical button present for resetting it, the Arduino Nano is designed in such a way that by running software on a computer, we can reset it.

2.10 Accelerometer:

An accelerometer is an electromechanical device that can measure acceleration forces. These forces may be static, like the constant force of gravity or they could be dynamic - caused by moving or vibrating the accelerometer.

We are using ADXL 335 accelerometer for measuring the acceleration produced whenever the head is moved in the 4 directions. The ADXL335 is a very small, thin in size, low power, 3-axis accelerometer with signal conditioned voltage outputs. The product can measure acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in, as well as dynamic acceleration which is due to motion or vibration.

We are using this in our head set controller to measure the acceleration of our head in each direction.

2.10.1 Benefits and Features:

1. 3-axis present for sensing
2. Package with low profile
3. It has Low power around 350 µA (typical)
4. It has Single-supply operation V to 3.6 V
5. Good temperature stability
6. Adjustable BW having a single capacitor per axis
7. RoHS/WEEE lead-free compliant

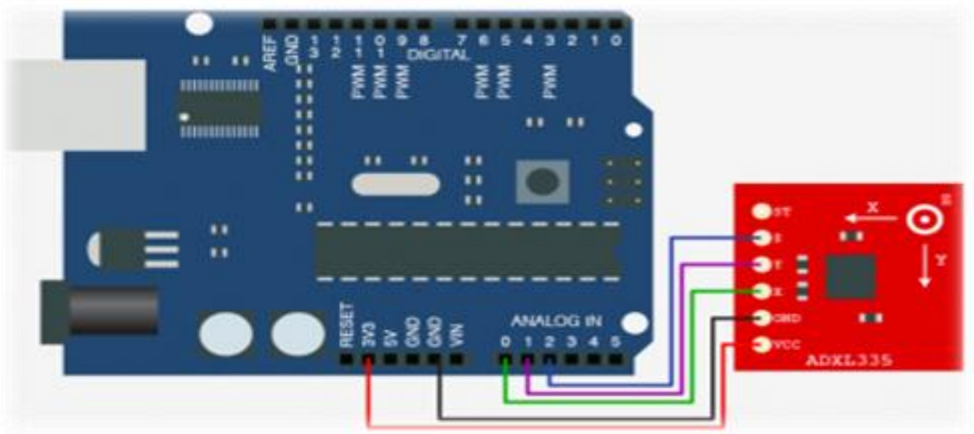

Figure 9: Accelerometer Interfaced with Arduino

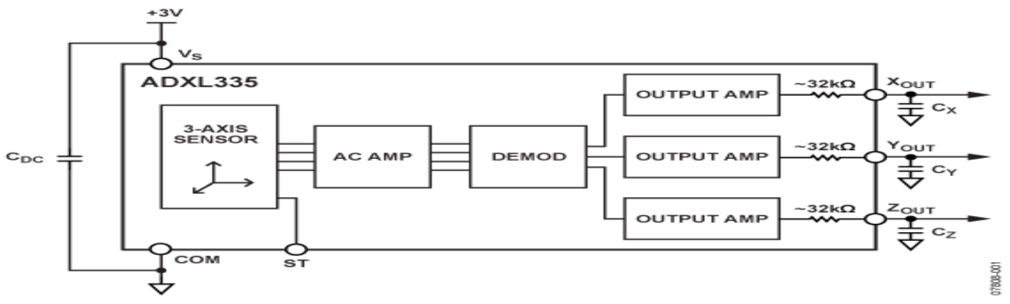

Figure 10: Accelerometer Internal Structure

2.11 Transmitter and Receiver Pair:

We are using two transmitter/receiver pairs. One will be present on the head set controller and the second will be on the UGV. We are using this to establish wireless serial communication between the Headset and the UGV. The pair that we are using is RF 433 Mhz.

As the name suggests, the RF module transmitter/receiver works with Radio Frequency. Using RF as compared to IR is more beneficial. The major advantage of using RF is that signals can travel through long distances so that makes it suitable for a number of applications. The drawback of IR is that it only operates at a line-of-sight, Even if there is an obstruction/hindrance in between, it won't affect the RF signals. RF transmission is very reliable and also very strong. RF pairs also use a specific kind of frequency which IRsensorsdon'thave. The RF module has basically two parts, a transmitter and a receiver. Its operating frequency is 434 MHz. An RF transmitter receives serial data and then transmits it wirelessly with the help of an antenna which is connected at pin no: 4. The rate of transmission is 1Kbps - 10Kbps.The transmitted data will be received by the receiver having the same frequency.

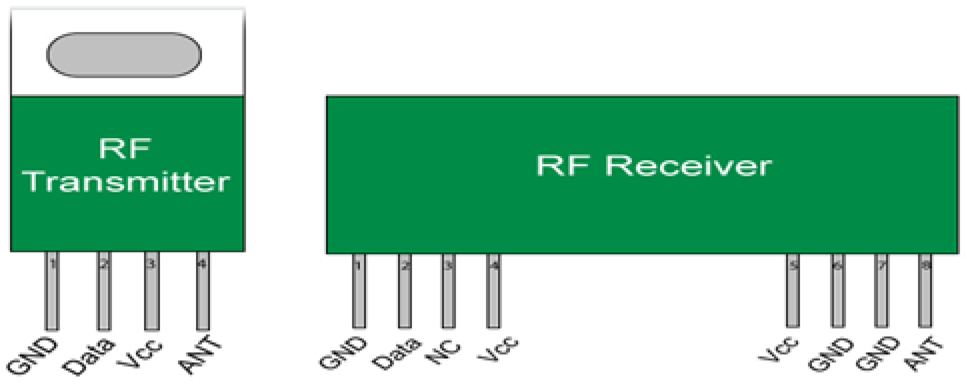

Figure 11: RF Transmitter/Receiver

Pin No	Function	Name
1	Ground (0V)	Ground
2	Serial data input pin	Data
3	Supply voltage; 5V	Vcc
4	Antenna output pin	ANT

Pin No	Function	Name
1	Ground (0V)	Ground
2	Serial data output pin	Data
3	Linear output pin; not connected	NC
4	Supply voltage; 5V	Vcc
5	Supply voltage; 5V	Vcc
6	Ground (0V)	Ground
7	Ground (0V)	Ground
8	Antenna input pin	ANT

2.12 Ultrasonic Sensors:

We have decided to use ultrasonic sensor HC –SR04 which is located in the front of the UGV. We are using this sensor for obstacle detection. Whenever the sensor will detect an obstacle in front, it will change its direction.

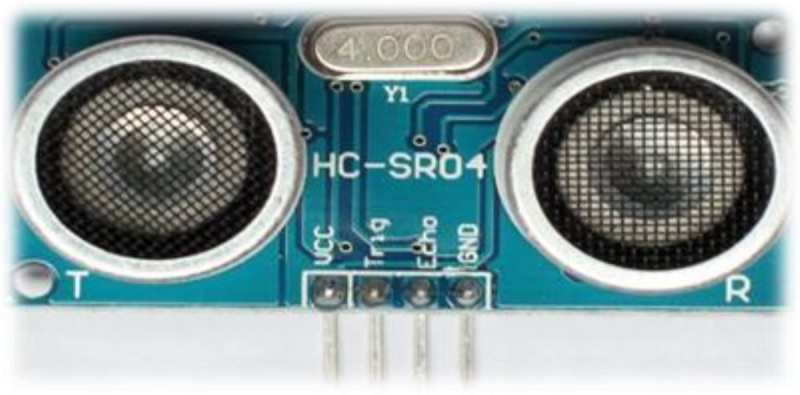

Figure 12: Ultrasonic Sensors

2.12.1 Features:

The Ultrasonic sensor range is around 2cm - 400cm non-contact measurement. The accuracy is around 3mm. The module consists of 3 parts, the transmitter, receiver and its operating circuit. The 3 main steps of its working are:

1. Using input output trigger for a minimum 10us high level signal,

2. The Sensor sends automatically eight 40 kHz and detect whether there is a back signal.

3. IF the signal comes back is through high level, time of high output Input output duration is

Basically the time from sending ultrasonic radiations to returning back.

Test distance = high time × velocity of sound 340M/S / 2

2.12.2 Electric Parameters:

1. Working Voltage DC	5 V
2. The Working Current	15mA
3. The Working Frequency	40Hz
4. The Max Range	4m
5. The Min Range	2cm
6. The Measuring Angle	15 degree
7. Trigger Input Signal	10uS TTL pulse

2.13 Distance Measuring Sensors:

We have used distance measuring sensor GP2Y0A21YK0F which is located at the bottom of the UGV. it is used for ditch detection. Whenever the ditch is detected the UGV changes its direction.

GP2Y0A21YK0F is a distance measuring sensor, which consists of an integrated combination of position sensitive detector known as (PSD), an infrared diode and signal processing circuit. Due to the reflectivity of the object, the temperature and the its duration are not influenced easily to the distance detection ,this is because of the triangulation method. The voltage is outputted in accordance with the distance detected. This sensor can also be used as a proximity sensor.

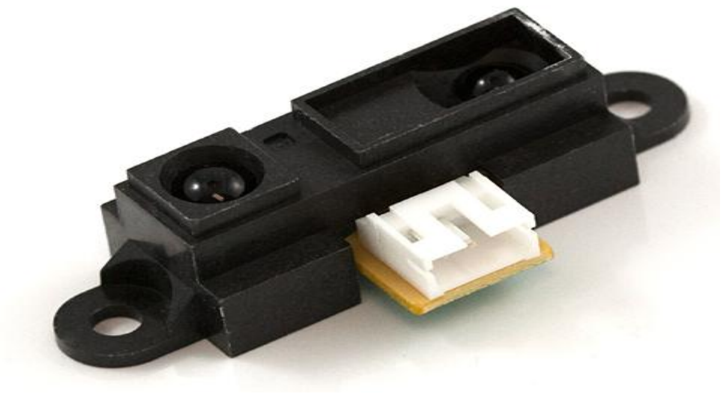

Figure 13: Distance Measuring Sensors

2.13.1 Features:

1. 1. The range of distance measurement : 10 to 80 cm
2. 2. The Package size : 29.5×13×13.5 mm
3. 3. Consumption current is 30 mA
4. 4. The Supply voltage range is 4.5 to 5.5 V
5. The output is analog

2.13.2 Applications:

1. In a Touch-less switch

2. In an energy saving sensor e.g in an ATM

3. In a Robot cleaner

4. Recreation equipment e.g. Arcade game machine

The GP2Y0A21 Sharp distance sensor is a amazing way for detecting ditches. The sensor is extremely easy to use with a detection range of 10 cm to 80 cm.

2.14 Continuous rotation servo and Standard servo:

We have used continuous rotation servo motors and standard servo for the gun movement which is mounted on the top of the UGV. It is the best for robotics and other projects involving movement. It is made for rotating continuously and is very easy to interface with any microcontroller.
Continuous-rotation servos are just like normal servo motors that technically should not be called servos because they do not have the feedback control, which is the present in almost every servo system.

The motor can be divided into such part:

1. Motor
2. Gearbox
3. Position Sensor
4. Motor Control Electronics

available.

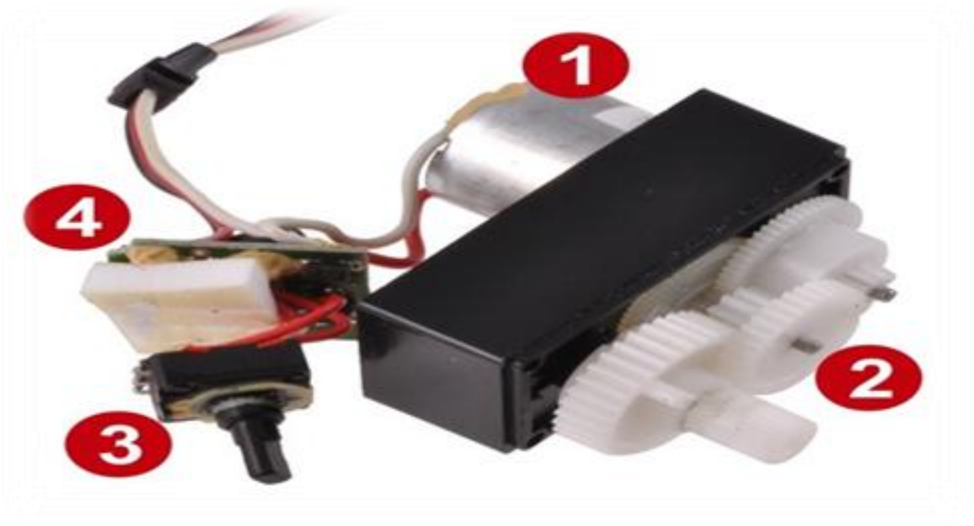

Figure 14: Continuous rotation servo and Standard servo

A continuous-rotation servo does away with its connection to the output and thereby continually drives the motor.

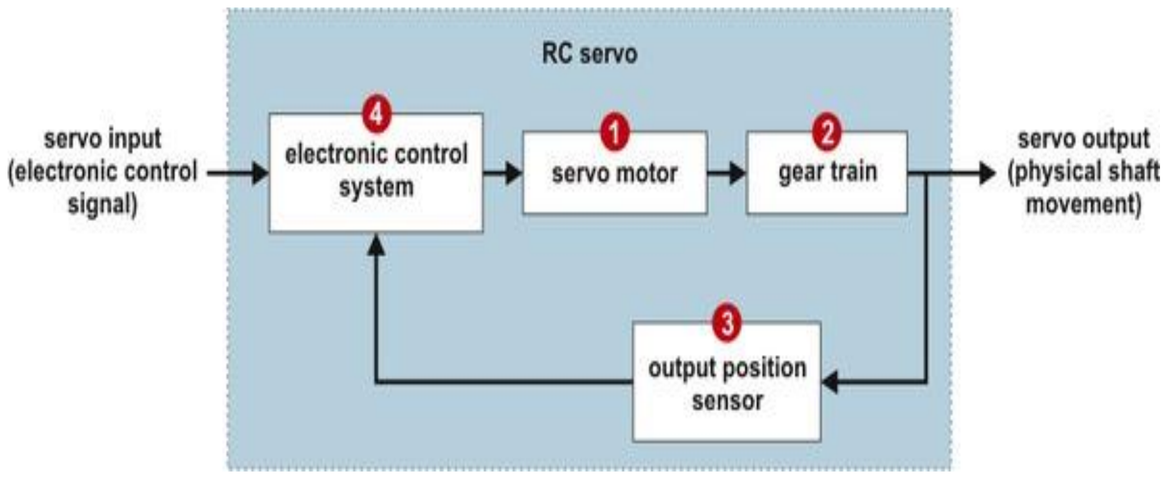
RC servo
servo input (electronic control signal)
4
electronic control system
1
servo motor
2
gear train
servo output (physical shaft movement)
3
output position sensor

Figure 15: RC Servo

The best thing about continuous rotation servos is that a motor, gearbox, and motor controller can be found in a single package and for a low price. A normal servo motor can also be modified for rotating continuously. Therefore a large range with different sizes is available.

2.15 Power Window Motor:

Figure 16: Power Window Motor

We are using the Power window motor for moving the base of the UGV because Power window motors are high torque motors they can handle heavy weights easily as they are made to lift glass windows of cars we are using DENSO motors in our project.

2.16 H-Bridge:

The H-bridge is an accurate and right concept for almost every motor control. With its help it can move forward, backward and with varying speeds. To make a DC motor bidirectional a H-Bridge is a key component it can be made using transistors or relays depending on the current handling requirement. There are also H bridges available in the IC form but they are not capable of handing hugs current a famous L298N H Bridge IC can handle up to 2A current. The basic circuit of an H-Bridge is shown in the figure below.

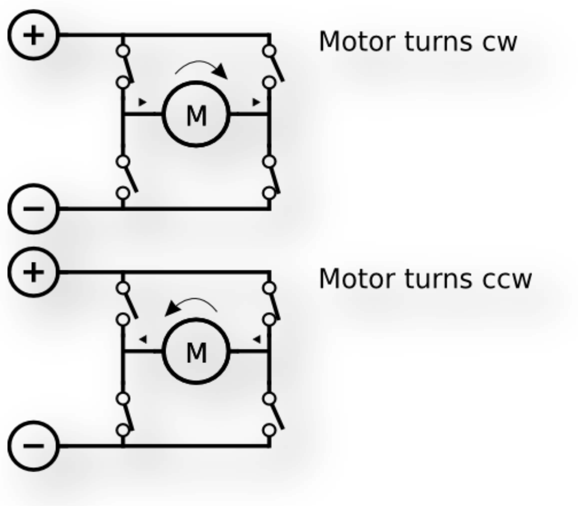

Figure 17: H-Bridge

These switches can be transistors or relays. A relay based H Bridge can handle huge current but its switching speed is slow while a transistor based h bridge can handle less current but its switching speed is very fast. We have decided to make a H-bridge that is transistor based using Darlington transistors which is capable of handling 5A current.

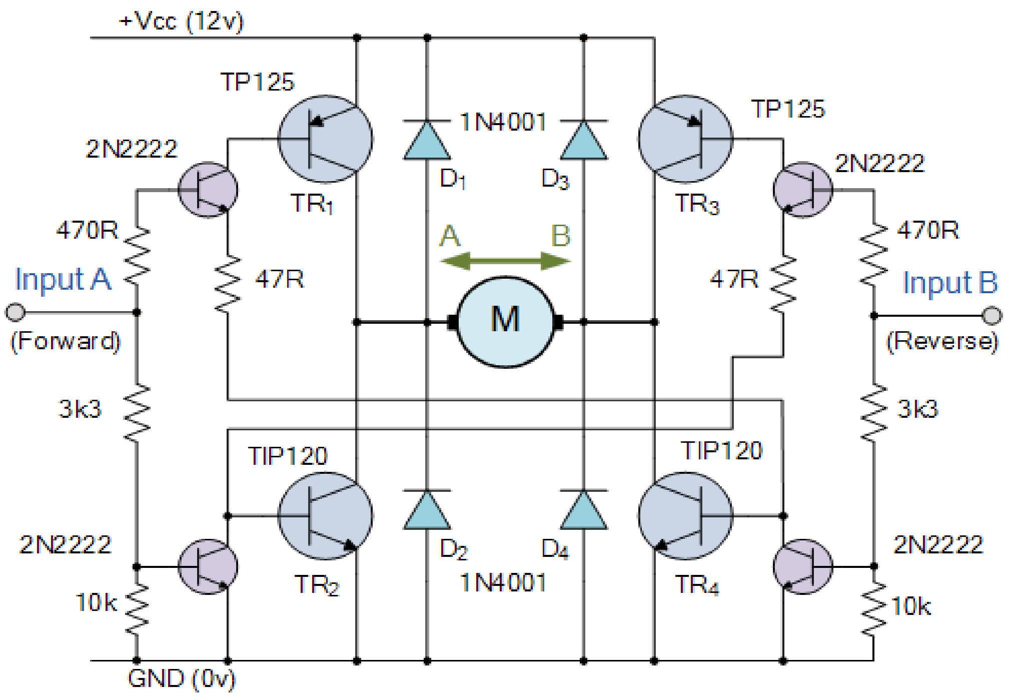

Figure 18: H-Bridge Circuit Diagram

2.17 Bluetooth Module:

Bluetooth is a wireless technology standard used for communication over short distance. There are different Bluetooth modules available for communication master and slave modules some can be made both master and save some can only act as slave module. A master module can build connections with other modules while a slave module can only connect to a master.

We have decided to use **HC-06** Bluetooth module in our project. **HC-06** module is a slave module. It is useful for connecting to a notebook, smart phone as a master to a robot with a slave module e.g. for a wireless serial bridge.

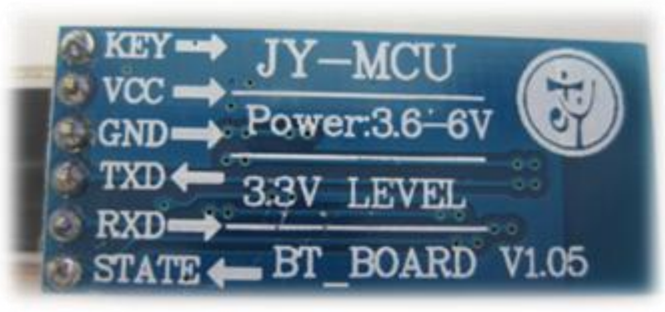

Figure 19: Bluetooth Pin Configuration

Figure 20 Bluetooth Module

2.17.1 Pin configuration:

KEY: No connection
VCC: 3.6V-6V
GND: 0v
TXD: serial output of the module, to be connected to RX of the microcontroller.
RXD: serial input of the module, to be connected to the TX of the microcontroller.
STATE: connected to LED2 (Pin32) of the module

CHAPTER 3

PROJECT DESIGN

3.1 Modules of Design

Our project consists of following eight modules.

1. Unarmed Ground Vehicle (UGV)
2. Headset for Head Gestures
3. Ultrasonic sensor (Obstacles Avoidance)
4. Wireless Electric Gun
5. Bluetooth Module (controlling UGV from mobile application)

3.2 SYSTEM MODEL

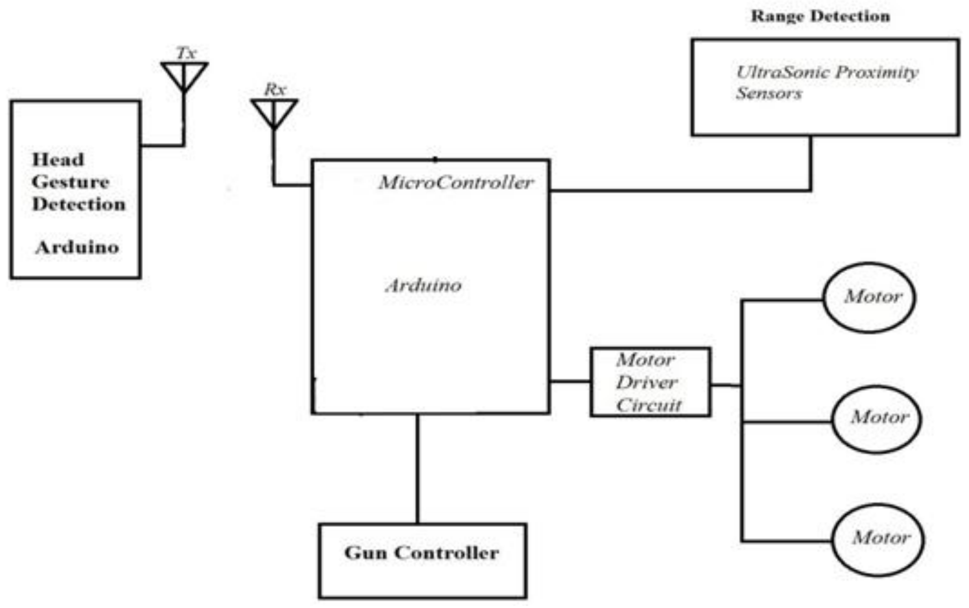

Figure 21: System Model

3.3 ACHIEVEMENTS

1. Successfully Learned Arduino board programming.
2. We have designed the head set controller by interfacing an accelerometer(ADXL 335) with an Arduino Nano.
3. We have also interfaced push buttons which will switch the control to autonomous mode and buttons which will control the gun, which is mounted at top.
4. Successfully Implemented wireless Interface between two Arduinos using 433 MHz Rf Transmitter Receiver
5. We have completed our designing of UGV.
6. We can control the motion of UGV using the headset.
7. The are able to control UGV by an android application.
8. We have mounted a gun on the top of UGV.
9. The gun can be controlled by the Headset and by Android application.

3.4 UGV design:

We started the mechanical design of the UGV by making a simple sketch on paper after discussions with few mechanical engineers we modified our design taking the manufacturing feasibility of the design in consideration we had to design few parts of our UGV on AutoCAD but most of the design of the UGV was done on paper. Our UGV design resembled closely with a tank as we wanted to arm it with a gun and the steering type which we were using to turn it was also similar to a real tank design we were using built instead of separate wheels.

Once our design was completed on paper we had to fabricate it being electrical engineers we lacked the knowledge and equipment required to fabricate the UGV design on our own so we took the help of some professional mechanical engineers to convert our design into real product. We worked with them and learned a lot of mechanical fabrication work. With their help and our effort we successfully completed the mechanical part of our UGV.

3.5 Control Circuit of UGV:

After complicating the UGV design, the next step was to make the control card for our UGV that will act as the brain of the UGV and will enable it to move. We decided to use Arduino Uno microcontroller board to control all the UGV motions. We attached three H Bridges, two ultrasonic sensors , one Sharp distance measuring IR sensor , an RF receiver and a Bluetooth module with Arduino. Before attaching all the components on the UGV we tested each component separation by interfacing with Arduino and studied their working.

We assembled Three H BRIDGE to control three DC motors, two motors for forward and backward motion of UGV and other one motor for the up and down motion of the gun mounted on top of the UGV for clockwise or anticlockwise motion of the gun we used a continuous rotating servo motor so we didn't needed an H bridge for this. Motors which we use to move our UGV are high torque motors, these motors are also use in power window motors.

As we have also put autonomous mode in or UGV. For this purpose we have used ultrasonic sensor to detect hurdles and sharp distance measuring IR sensor for ditch (edge) detection. We install two ultrasonic sensors on front and IR sensor on bottom of UGV.

3.6 Coding logic for UGV Arduino Uno:

Our coding logic was that we received the data through the serial port of the arduino. This data contained the information of the accelerometer which is on top of the user heads based on this reading we have added different check in our code at a particular value of accelerometer different checks will be executed that will control the UGV motors through an H bride and thus move the UGV. The Arduino will also be taking feedback from all the sensors once the UGV has any obstacle or ditch in its way it will move the UGV backward. On this Arduino will be receiving two serial data's one from the accelerometer on users head and one from the android app so we have used two serial ports one serial port have been generated through a software serial library.

3.7 Headset controller circuit:

We made an headset controller by using an Arduino NANO, an accelerometer , RF transmitter and different push buttons for actuations control. We fitted all the circuits in a hollow headphone body so that it becomes easy for a used to wear the circuits on his head. We also designed the power circuit which will run the Arduino Nano and 433 MHz RF transmitter.

3.8 Coding logic for Headset Arduino Nano:

The accelerometer ADXL-335 which is an analog accelerometer is used to capture the users head gestures. The UNO receives the analog data of accelerometer converts it into digital data by its built in A-D converter and the sends it serially to the Arduino UNO which in controlling the UGV.

3.9 Android Control:

Once the basic aim of our project was completed we were able to control the UGV through a person’s head gestures started developing an android based control for our UGV. The first thing we needed was an app that can communicate with Arduino due to shortage of time we wanted to find a premade free app suitable for full filling our requirements. Luckily we found an app in the android marked which was a free app and was suitable for serving our purpose. Now the next task was interfacing this app with the arduino on UGV. We used a Bluetooth to serial module with Arduino for communication with this app. Once the communication was established then we send specific commands from the app based on different push of buttons.

3.10 Coding logic for Android control:

The android app was sending different data based on different push of buttons once we received this data on the arduino UNO using Bluetooth module we added different check which were exsiccated by the data we were receiving each check will provide some kind of motion to the UGV.

CHAPTER 4

PROJECT DEVELOPMENT

4.1 Headset Design And Development

This is the figure of the headset that we have designed. Its each component is explained below with pictures. This headset is used to give commands to the designed UGV which will move accordingly.

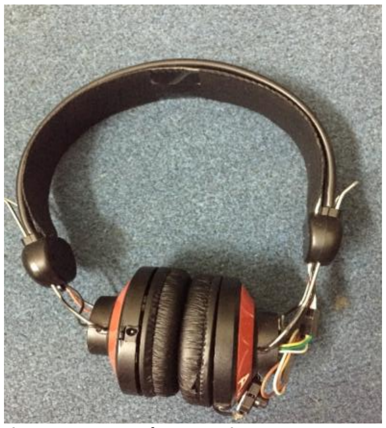
Figure 22: Headset Design

4.2 Accelerometer:

We are using ADXL 335 accelerometer for measuring the acceleration produced whenever the head is moved in the 4 directions. The ADXL335 is a very small, thin in size, low power, and 3-axis accelerometer. We are using this in our head set controller to measure the acceleration of our head in each direction.

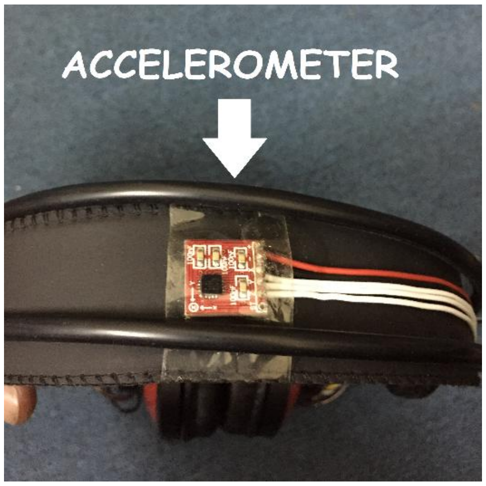

Figure 24: Accelerometer

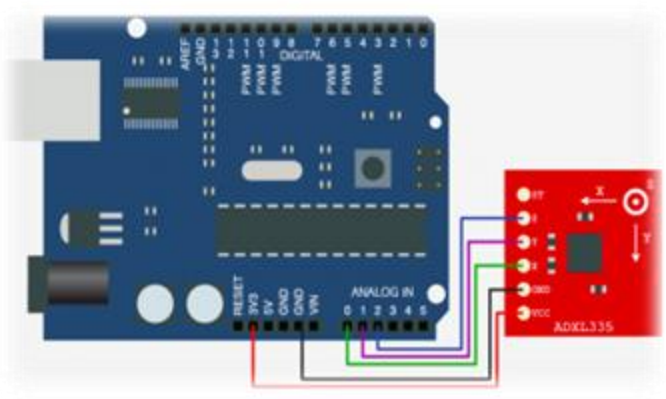

Figure 23: Accelerometer Configuration

4.3 Arduino Nano:

We are using an Arduino Nano in our head set controller. We have chosen this over Arduino Uno to be used in our head set controller because it less in weight and smaller in size. The Arduino Nano is a much smaller, comprising of everything, and user-friendly board. It is based on the ATmega328.

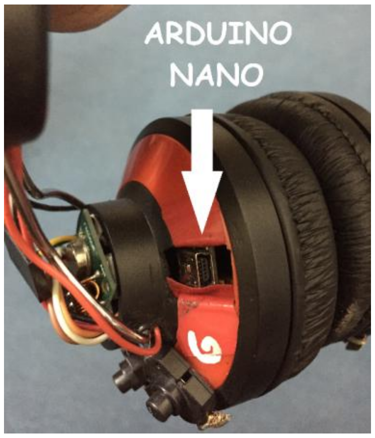

Figure 25: Arduino Nano

4.4 Control Switches:

These are the control switches that enable us to transfer the control from the UGV body to the gun and vice versa. Middle switch is used to press the trigger of the gun. Similarly the last button on left side is used to move UGV autonomously.

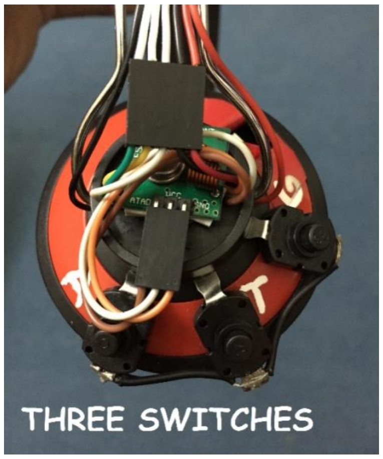

Figure 26: Control Switches of Headset

4.5 RF Transmitter:

We are using two transmitter/receiver pairs. This one is its transmitter part. We are using this to establish wireless serial communication between the Headset and the UGV. The pair that we are using is RF 433 Mhz.

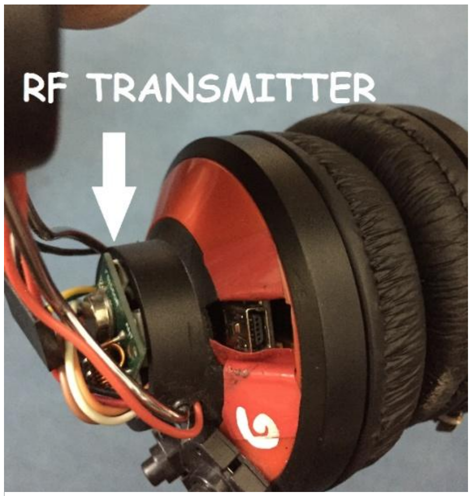

Figure 27: RF Transmitter

4.6 UGV Design And Development

This is the complete picture of our designed UGV. Its different components will be described below.

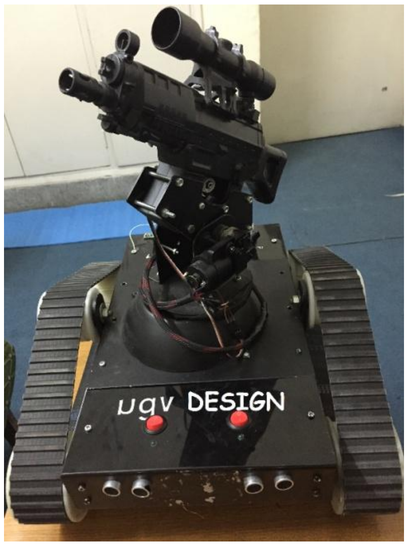

Figure 28: UGV Design

4.7 Mechanical Part :

This is specially designed aluminum body to encapsulate all the structure and wiring of the our UGV. It has got batters, h-bridges and motors inside it.

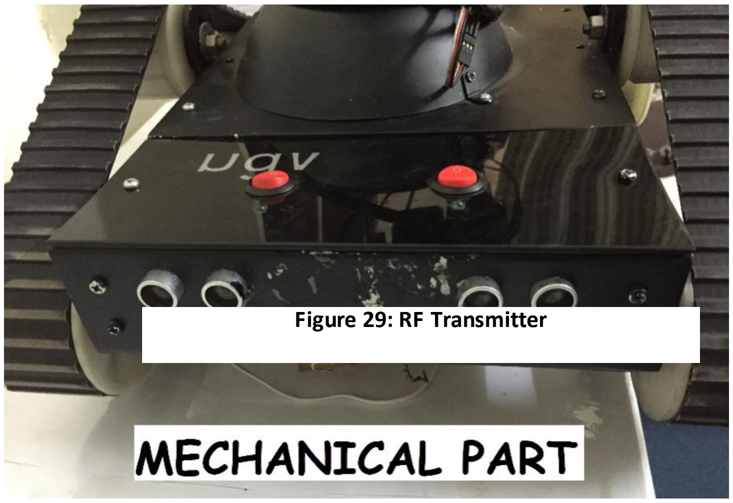

Figure 29: RF Transmitter

Figure 30: Mechanical Part

4.8 Tank Type Structure:

The UGV's movement is based on tank type structure. We have connected motors to the front wheels and they are further connected to rear wheels with the help of the belt.

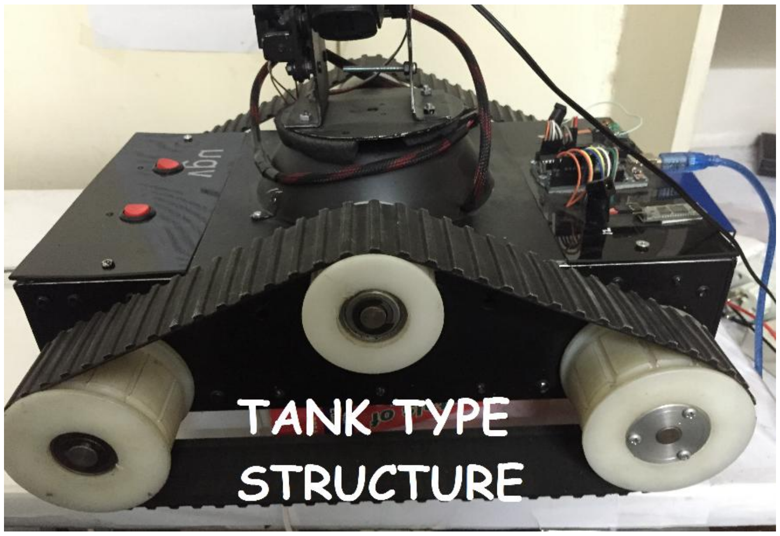

Figure 31: Tank Type Structure

4.9 Arduino Uno:

It is a microcontroller board based on the ATmega328. The board consists of 14 digital input/output pins.. Other components include a 16 MHz ceramic resonator, a USB connection, a power jack plus a reset button. All the motors and modules are connected with Arduino and it gives commands to them.

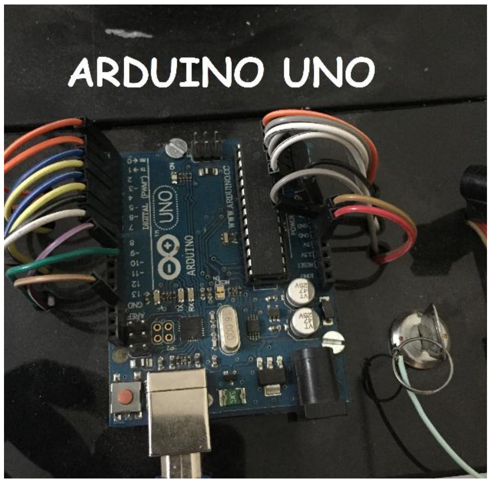

Figure 32: Arduino Uno

4.10 RF Receiver:

We are using two transmitter/receiver pairs. This one is its receiver part. We are using this to establish wireless serial communication between the Headset and the UGV. The pair that we are using is RF 433 Mhz.

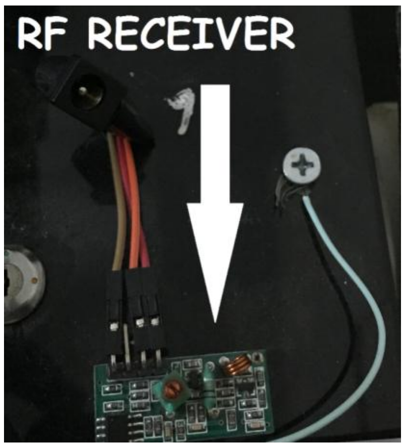

Figure 33: RF Receiver

4.11 Bluetooth Module:

We are using **HC-06** Bluetooth module in our project. **HC-06** module is a slave module. It is useful for connecting to a notebook, smart phone as a master to a robot with a slave module e.g. for a wireless serial bridge.

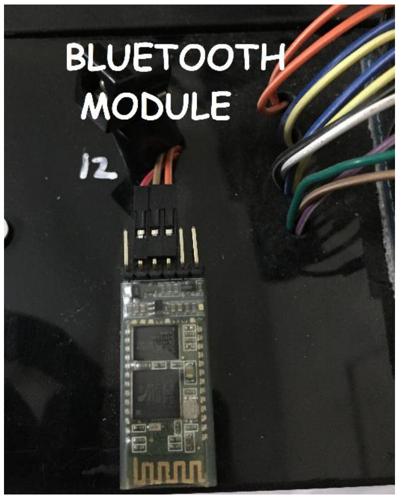

Figure 34: Bluetooth Module

4.12 Battery Switches:

There are two switches of the batteries (12V and 6V). They are used to turn off/on the UGV modules and working.

4.13 Ultrasonic sensor:

Two ultrasonic sensors are planted at the front of the UGV body. Their work is to detect any obstacle present in the path when the UGV is working in autonomous mode.

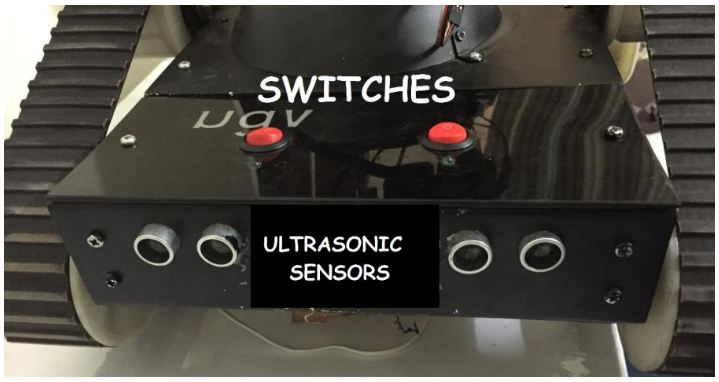

Figure 35: Ultrasonic Switches

4.14 DC and Survo Motors:

We are using 5 motors in our UGV side. 3 are DC motors and 2 are survo. 2 DC motors are used to control the UGV wheels and one is used to move the gun in up/down direction. One survo is used to move the gun in 360 degrees and second is used to move the trigger of the gun.

4.15 Auto Reload Gun:

We have bought this gun from the market, its special features is that reload automatically. The batteries inside it can be recharged.

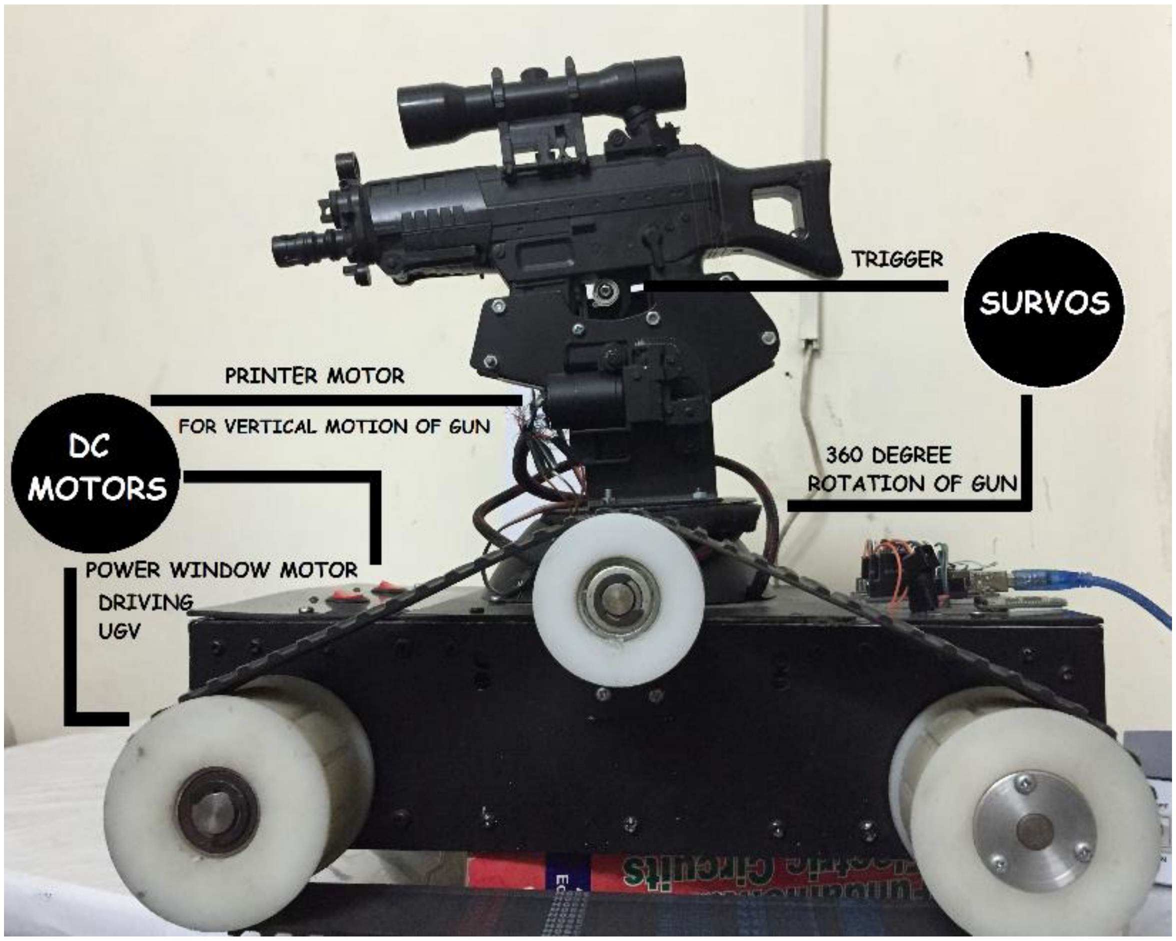

Figure 36: Auto reloaded Gun

4.16 Android App Implementation:

Each part of the UGV and gun can also be controlled by the Android application which is connected via Bluetooth.

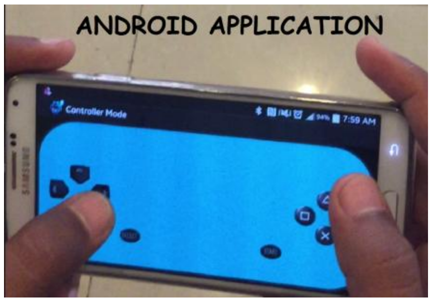

Figure 37: Android Application

4.17 PROJECT ANALYSIS AND EVALUATION

We have checked out project and it is working perfectly. As it is a hardware based projects so its output will be shown in the demo. Coming to analysis we can say:

1. Our designed UGV can move in all four directions.
2. Gun mounted on UGV can move 30-60 degree vertically.
3. The gun can rotate 360 degree on horizontal scale.
4. The gun can fire and can be controlled by Headset and mobile application.
5. Our UGV avoids obstacles when anything come in its way.
6. The way our project has come to the final stages is shown below.

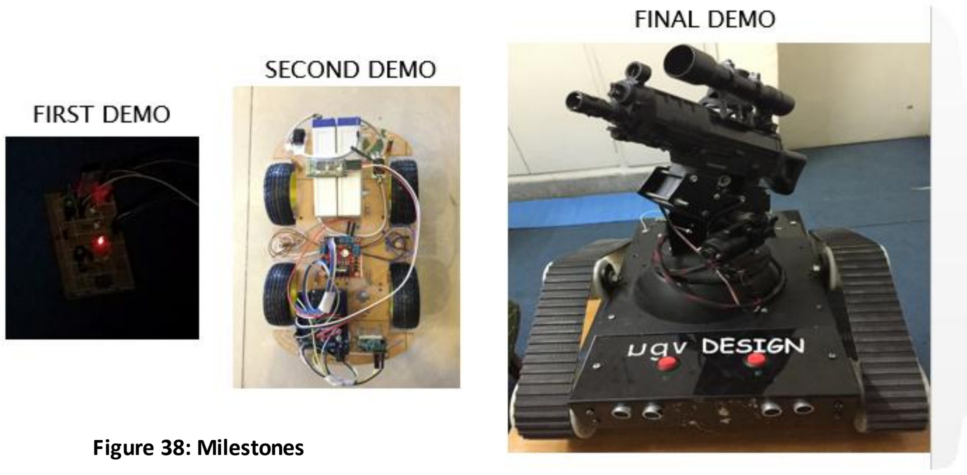

Figure 38: Milestones

CHAPTER 5
RECOMMENDATIONS FOR FUTURE WORK

1. The UGV speed can be increased by using heavy motors.
2. The wireless communication range can be increased by using high power RF Transmitter/Receiver.
3. A camera can be fitted on the UGV and can be controlled when the UGV is out of sight.
4. Ditch detection sensors can be added.
5. Angle of gun movement can also be increased in future.

CHAPTER 6
CONCLUSION

6.1 Overview

Our Project is based on wireless communication between the controller and UGV with a gun on it. We have operated our UGV and gun by users head gestures and by an Android application. We have worked on making the hardware as well as software for controlling our UGV. We have used Arduino board for the design of both UGV and the headset controller which will detect the head gesture and accordingly move the UGV. The UGV will be able to move in all possible directions. These same head gestures are used to move to gun in 360 degrees on horizontal scale and 0-90 degree on vertical scale. The gun can reload automatically

.

6.2 Objective Achieved

1. Successfully Learned Arduino board programming.
2. We have designed the head set controller by interfacing an accelerometer (ADXL 335) with an Arduino Nano.
3. We have also interfaced push buttons which will switch the control to autonomous mode and buttons which will control the gun, which is mounted at top.
4. Successfully Implemented wireless Interface between two Arduinos using 433 MHz Rf Transmitter Receiver
5. We have completed our designing of UGV.
6. We can control the motion of UGV using the headset.
7. They are able to control UGV by an android application.
8. We have mounted a gun on the top of UGV.
9. The gun can be controlled by the Headset and by Android application.

CHAPTER 7

BIBLIOGRAPHY

[1]http://dronecenter.bard.edu/shadows-strange-world-ground-drones Accessed on 15/01/2014

[2]http://en.wikipedia.org/wiki/Unmanned_ground_vehicle Accessed on 15/01/2014

[3]http://science.howstuffworks.com/military-robot3.htm Accessed on 15/01/2014

[4]http://usmilitary.about.com/od/marineweapons/a/gladiator.htm Accessed on 20/01/2014

[5]http://www.gizmag.com/go/4484/ Accessed on 20/01/2014

[6]Bekey, A., Autonomous Robots: From Biological Inspiration to Implementation

And Control, MIT Press, 2005

[7] Brooks, R. and Flynn, A., "Fast, Cheap and Out of Control: A Robot Invasion of

the Solar System," Journal of the British Interplanetary Society, 1989

[8] Murphy, R., An Introduction to AI Robotics, MIT Press, 2000

[9] Arkin, R., Behavior Based Robotics, MIT Press, 1998

APPENDICES

APPENDIX-A

UGV controller code:

```
#include <EasyTransfer.h>
const int gun_leftright=11;
const int gun_updown=3;
const int left_front=5;
const int left_back=6;
const int right_front=9;
const int right_back=10;
const int ir_sensor = A0;
const int echoPin= 2; // Echo Pin
const int trigPin=12; // Trigger Pin

//create object
EasyTransfer ET;

struct RECEIVE_DATA_STRUCTURE{
// put your variable definitions here
// for the data you want to receive
// THIS MUST BE EXACTLY THE SAME ON THE OTHER ARDUINO
int ForBack; // forward backward variable
int LeftRight; // left right variable
int ForBack1; // forward backward variable
int LeftRight1;
int atonomus;
};
//give a name to the group of data
RECEIVE_DATA_STRUCTURE mydata;


void setup()
```

```
{
 pinMode(left_front, OUTPUT);
 pinMode(left_back, OUTPUT);
 pinMode(right_front, OUTPUT);
 pinMode(right_back, OUTPUT);
 pinMode(gun_leftright, OUTPUT);
 pinMode(gun_updown, OUTPUT);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);

Serial.begin(4800); // This is for the computer to check for data
//Serial3.begin(1200); // In Arduino Mega 2560, you have 4 serial
// comm links. Connect the DATA pin of the receiver module
// to RX3 or pin 15 (they are the same). Pay attention to the
// low bandwidth.
//start the library, pass in the data details and
//the name of the serial port. Can be Serial, Serial1, Serial2, etc.
ET.begin(details(mydata), &Serial);


}

void loop(){

//else


//{
//pulseServo(left_front, 1000); //  moves forward
//pulseServo(left_back, 1000);
//pulseServo(right_front, 1000); //  moves backward
//pulseServo(right_back, 1000);
//}
//check and see if a data packet has come in.

if(ET.receiveData()){
```

```
//this is how you access the variables. [name of the
group].[variable name]
// Record Forward/Backward and Left/Right values

int RecForBack = mydata.ForBack;
int RecLeftRight = mydata.LeftRight;
int RecForBack1 = mydata.ForBack1;
int RecLeftRight1 = mydata.LeftRight1;
int RecAtonomus = mydata.atonomus;
//Serial.print(RecForBack1);
//Serial.print("\t");
//Serial.println(RecLeftRight1);
//delay(5);

if (RecForBack1 ==0 && RecLeftRight1 == 0 )
{
if (RecForBack <= 72) // move forward
{
pulseServo(left_front, 1000); //  moves forward
pulseServo(left_back, 1000);
pulseServo(right_front, 2000); //  moves forward
pulseServo(right_back, 2000);
}



  if (RecForBack >= 85) // move backward
  {
pulseServo(left_front, 2000); //  moves backward
pulseServo(left_back, 2000);
pulseServo(right_front, 1000); //  moves backward
pulseServo(right_back, 1000);
  }
  if (RecLeftRight <= 77) // move right
  {
pulseServo(left_front, 1000); //  moves forward
pulseServo(left_back, 1000);
pulseServo(right_front, 1000); //  moves backward
pulseServo(right_back, 1000);
  }
  if (RecLeftRight >= 91) // move left
  {
pulseServo(left_front, 2000); //  moves backward
pulseServo(left_back, 2000);
pulseServo(right_front, 2000); //  moves forward


pulseServo(right_back, 2000);
  }
}



if (RecForBack ==0 && RecLeftRight == 0 )
{
  if (RecForBack1 <= 72) // move up
{
pulseServo(gun_updown, 1350);
}



  if (RecForBack1 >= 85) // move down
  {
pulseServo(gun_updown, 1500);
  }
  if (RecLeftRight1 <= 77) // move right
  {
pulseServo(gun_leftright, 1600);
  }
  if (RecLeftRight1 >= 91) // move left
  {
pulseServo(gun_leftright, 1320);
  }
```

```
}
if (RecForBack ==1 && RecLeftRight == 1 && RecForBack1 ==1 &&
RecLeftRight1 == 1 )
{
  long duration, distance; // Duration used to calculate distance
  /* The following trigPin/echoPin cycle is used to determine the
 distance of the nearest object by bouncing soundwaves off of it. */
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.
 distance = duration/58.2;


  int sensor_value = analogRead(ir_sensor);  //read the sensor value
  int distance_cm = pow(3027.4/sensor_value, 1.2134);
  //Serial.println(distance_cm);
  if ( distance_cm > 9 || distance < 22)
  {
    for (int i=0 ; i<=50 ; i++)
  {
pulseServo(left_front, 2000); //  moves backward


pulseServo(left_back, 2000);
pulseServo(right_front, 1000); //  moves backward
pulseServo(right_back, 1000);
  }
for (int i=0 ; i<=100 ; i++)
  {
 pulseServo(left_front, 1000); //  moves fackward
pulseServo(left_back, 1000);
pulseServo(right_front, 1000); //  moves backward
pulseServo(right_back, 1000);
  }
  }
  else
  {
pulseServo(left_front, 1000); //  moves fackward
pulseServo(left_back, 1000);
pulseServo(right_front, 2000); //  moves fackward
pulseServo(right_back, 2000);
  }
}
// this delay must be lower than the delay in the transmitter code
//delay(5);
}
}
void pulseServo(int pin, int microseconds)
{
digitalWrite(pin, HIGH);
delayMicroseconds(microseconds);
digitalWrite(pin, LOW);
delay(5);
}
```

Headset controller code:

```
#include <EasyTransfer.h>

//create object

EasyTransfer ET;
int button1 = 12; // choose the input pin (for a pushbutton)
int button2 = 10; // choose the input pin (for a pushbutton)
struct SEND_DATA_STRUCTURE{
```

```
// put your variable definitions here
// for the data you want to send
// THIS MUST BE EXACTLY THE SAME ON THE OTHER ARDUINO
//int val = 0;   // variable for reading the pin status
int ForBack; // forward backward variable
int LeftRight; // left right variable
int ForBack1; // forward backward variable
int LeftRight1;
int atonomus;
};
//give a name to the group of data
SEND_DATA_STRUCTURE mydata;
void setup()
{
Serial.begin(4800); // In Arduino Mega 2560, you have 4 serial
// comm links. Connect the DATA pin of the transmitter module
// to TX3 or pin 14 (they are the same). Pay attention to the
// low bandwidth.
//start the library, pass in the data details and
pinMode(button1, INPUT);
pinMode(button2, INPUT);
// the name of the serial port. Can be Serial, Serial1, Serial2,
etc.
ET.begin(details(mydata), &Serial);

}
int xRead ; // forward backward variable
int yRead ; // left right variable

int val1 = 0; // variable for reading the pin status
int val2 = 0;
void loop()
{


  val1 = digitalRead(button1);  // read input value
  val2 = digitalRead(button2);  // read input value
  if (val1 == HIGH) {
  xRead = analogRead(A0)/4;
  yRead = analogRead(A1)/4;
   mydata.ForBack1 = yRead;
   mydata.LeftRight1 = xRead;

    mydata.ForBack = 0;//yRead;


             mydata.LeftRight = 0;//xRead;
    if (val2 == HIGH )
  {
    mydata.ForBack = 1;//yRead;
    mydata.LeftRight = 1;//xRead;
    mydata.ForBack1 = 1;
    mydata.LeftRight1 = 1;

ET.sendData();
  }

    ET.sendData();

  }
  if (val1 == LOW)
  {
  xRead = analogRead(A0)/4;
  yRead = analogRead(A1)/4;
mydata.ForBack = yRead;
mydata.LeftRight = xRead;
mydata.ForBack1 = 0;
mydata.LeftRight1 = 0;
if (val2 == HIGH )
  {
    mydata.ForBack = 1;//yRead;
    mydata.LeftRight = 1;//xRead;
    mydata.ForBack1 = 1;
    mydata.LeftRight1 = 1;

ET.sendData();
  }
ET.sendData();
  }

// Send it off over the air
//ET.sendData();
// this delay must be greater than the delay in the receiver code
//delay(10);
}
```

The Ultrasonic Sensor:

```
#define echoPin 2 // Echo Pin
#define trigPin 12 // Trigger Pin
#define LEDPin 13 // Onboard LED

int maximumRange = 100; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration, distance; // Duration used to calculate distance

void setup() {
 Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}

void loop() {
/* The following trigPin/echoPin cycle is used to determine the
 distance of the nearest object by bouncing soundwaves off of it. */
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.
 distance = duration/58.2;

 if (distance >= maximumRange || distance <= minimumRange){
 /* Send a negative number to computer and Turn LED ON
 to indicate "out of range" */
 Serial.println("-1");
 digitalWrite(LEDPin, HIGH);
 }
 else {
 /* Send the distance to the computer using Serial protocol, and
 turn LED OFF to indicate successful reading. */
 Serial.println(distance);
 digitalWrite(LEDPin, LOW);
 }

 //Delay 50ms before next reading.
 delay(50);
}
```

The Sharp IR Sensor:

```
 int ir_sensor = A0;

void setup() {
  //initialize serial communications at 9600 bps
  Serial.begin(9600);
}

void loop() {
  int sensor_value = analogRead(ir_sensor);  //read the sensor value
  int distance_cm = pow(3027.4/sensor_value, 1.2134); //convert
readings to distance(cm)
  Serial.println(distance_cm); //print the sensor value
  delay(500); //delay 500ms (0.5 second)
}
```

APPENDIX-B

SYNOPSIS

<u>Head Gesture Controlled UGV</u>

Extended Title: Head Gesture Controlled UGV with wireless gun interfaced with android phone .

Brief Description of The Project / Thesis with Salient Specs:

The UGV's which we had seen in past could not be moved autonomously and to control them people would use remote controlled modules and big joysticks. It is difficult to use and carry big sized modules esp. in battle fields and remote areas. A person having no knowledge of how to use these modules cannot start using them at once and it requires a lot of time to master them. Keeping the above problem in our mind we try to introduce Headset controller which is easy to carry, smaller in size and user friendly. Our main goal is to design a UGV that can be controlled through a person's head gestures and secondly can also be moved autonomously or by an Android application. With the help of a simple head set controller and android app controlling a UGV becomes simple and easy

Scope of Work :

1. To make the UGV which can move in difficult terrains.
2. Controlling the UGV by users head gestures using a specially designed headset.
3. Wireless control of UGV using head gestures.
4. The UGV moves in four directions (forward, backward, left and right) at any angle the user wants.
5. UGV will be armed with a gun mounted on top of it which can move in 360 degree in circle and 45 degree up and down and all these motions must be control by headset controller that recognizes the users head gestures.
6. UGV has also autonomous mode. UGV will move automatically in autonomous mode and UGV will be able to avoid obstacles and ditches in its way.

Previous Work Done on The Subject :

Wireless controlled UGV **,**Obstacle avoidance robot [Nust 2012] **,**Remote controlled UGV with gun [MCS 2013]

Material Resources Required :

UGV, Electric Gun, Sensors, Android Phone, Headset

APPENDIX-C

COST BREAKDOWN

Index	Equipment	Quantity	Unit price	Cost
1.	Survo Motors	3	1200	3600
2.	Power Window Motor	2	2000	4000
3.	Printer Motor	1	1500	1500
4.	Ultrasonic Sensors	2	700	1400
5.	Accelerometer	1	2800	2800
6.	H-Bridge	2	1500	3000
7.	Auto Reloader Gun	1	2500	2500
10.	Bluetooth Module	1	1500	1500
11.	Robotic kit(material)	1	2500	2500
12.	Bluetooth Module	1	1000	1000
13.	RF Transmitter	1	1200	1200
14.	Batteries (Li-Ion)	3	800	2400
15.	12V Battery	1	2000	2000
16.	6V Battery	1	1000	1000
17.	Arduino	2	2000	4000
18.	Belts	3	15000	4500
19.	Teflon Wheels	6	150	900

20.	Headphone	1	800	800
21.	Mechanical Design	1	15000	15000
	Total Cost			55600