TWO AXIS CONTROL SYSTEM FOR A RECEIVING ANTENNA (BSS) OVER A MOVING PLATFORM

By

PC Haider Ali Zaheer
PC Taimoor Siddique
PC Luqman Sajid
ASC Junaid Haider

Supervised By

Col Dr. Amer Gilani, R&D Wing MCS

Submitted to the Faculty of Electrical (Telecomm)Engineering

National University of Sciences and Technology, Islamabad (Military College Of Signals)
in partial fulfillment

For the requirements of a B.E. Degree in Electrical(Telecomm) Engineering

JUNE 2019

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis – “Two axis control system for a receiving antenna (BSS) over a moving platform”carried out by PC Haider Ali Zaheer, PC Taimoor Siddique, PC Luqman Sajid and ASC Junaid Haider in supervision of Col. Dr. Amer Gilani for partial fulfilment of Degree of Bachelor of Electrical(Telecomm)Engineering (MCS,NUST) is correct and approved.

Approved by

Col. Dr. Amer Gilani

EE DEPARTMENT

MCS,NUST

DATED:

ABSTRACT

With the increasing popularity of Television satellite reception systems there has been a tendency to apply the available technology in connection with recreational vehicles. Such vehicles also have achieved considerable popularity and accordingly, a need has developed for practical, inexpensive and readily adaptable systems for dish antenna installation on recreational vehicles so that the vacationer/ traveler can enjoy satellite-transmitted television programs. Satellite television is a miracle of modern technology, delivering clear, crisp picture and sound from satellites thousands of miles away, captured by a relatively small satellite dish.

Over the past decade, TVRO antennas have grown substantially in popularity and are typically found in geographical areas where cable or broadcast television is not prevalent. The advent of such commercially available programming from these satellites has found growing popularity among recreational vehicle (RV) users who would like to tap into this programming during their trips around the country in recreational vehicles. Initial satellite TVRO systems for recreational vehicles were simply comprised of a small TVRO dish antenna placed on the ground near the RV which was then manually adjusted with great care and time to locate and tune into an individual satellite. The tuning process would be repeated for tuning into another satellite. This approach was somewhat effective but resulted in considerable set-up time by the consumer and usually resulted in low quality signals in the television set.

We have designed a two-axis control system for a receiving antenna (BSS) over a moving platform. We have used two microcontrollers (ATmega328P). To one microcontroller, magnetometer (HMC5883) is attached which provides us the azimuthal angle. To the second microcontroller, gyroscope (MPU6050) is attached which gives us the angle of elevation. So, we are using separate controllers for getting azimuthal and elevation angles. To the first microcontroller, GPS is also attached. The controller takes points from GPS and calculate the desired azimuthal and elevation angles.

The ATmega328p will compare these angles with current angles of the dish antenna, taken using HMC5883 and MPU6050 placed along the dish antenna. If the angles are not

aligned, then the controller will adjust current position of our dish antenna by moving it through stepper motors connected. Once the respective angles provided by user are in accordance with dish angles, we will receive proper output from our dish receiver. In this way, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

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 Merciful, the most Beneficent.
To our parents, without whose unflinching support and cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings, which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support, which makes us what we are.

We are extremely grateful to our project supervisor Col. Dr. Amir Gilani from MCS who in addition to providing us with valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all our teachers, library staff, and faculty our Electrical Engineering department of Military College of Signals, NUST. We are extremely thankful and indebted to them for sharing expertise, and sincere and valuable guidance and encouragement.

TABLE OF CONTENTS

TABLE OF FIGURES:

CHAPTER:1

INTRODUCTION

1 INTRODUCTION

1.1 Overview

The two-axis control system for a receiving antenna (BSS) over a moving platform will provide arrangements for changing or varying the orientation or the shape of the directional pattern of the waves radiated from an antenna using mechanical movement of antenna for varying two co-ordinates of the orientation (elevation and azimuth angles). As a result, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

1.2 Problem Statement

Over the past decade, TVRO antennas have grown substantially in popularity and are typically found in geographical areas where cable or broadcast television is not prevalent. The advent of such commercially available programming from these satellites has found growing popularity among recreational vehicle (RV) users who would like to tap into this programming during their trips around the country in recreational vehicles. Initial satellite TVRO systems for recreational vehicles were simply comprised of a small TVRO dish antenna placed on the ground near the RV which was then manually adjusted with great care and time to locate and tune into an individual satellite. The tuning process would be repeated for tuning into another satellite. This approach was somewhat effective but resulted in considerable set-up time by the consumer and usually resulted in low quality signals in the television set.

1.3 Approach

We have designed a two-axis control system for a receiving antenna (BSS) over a moving platform. We have used two microcontrollers (ATmega328P). To one microcontroller, magnetometer (HMC5883) is attached which provides us the azimuthal angle. To the second microcontroller, gyroscope (MPU6050) is attached which gives us the angle of elevation. So, we are using separate controllers for getting azimuthal and elevation angles. To the first microcontroller, GPS is also attached. The controller takes points from GPS and calculate the desired azimuthal and elevation angles.

The ATmega328p will compare these angles with current angles of the dish antenna, taken using HMC5883 and MPU6050 placed along the dish antenna. If the angles are not aligned, then the controller will adjust current position of our dish antenna by moving it through stepper motors connected. Once the respective angles provided by user are in accordance with dish angles, we will receive proper output from our dish receiver. In this way, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

1.4 Scope

With the increasing popularity of Television satellite reception systems; there has been a tendency to apply the available technology in connection with recreational vehicles. Such vehicles also have achieved considerable popularity and accordingly, a need has developed for practical, inexpensive and readily adaptable systems for dish antenna installation on recreational vehicles so that the vacationer/ traveler can enjoy satellite-transmitted television programs

1.5 Aim &Objectives

The main aim and objective of our project is develop a two-axis control system for a receiving antenna (BSS) over a moving platform to provide practical, inexpensive and readily adaptable system for dish antenna installation on recreational vehicles so that the vacationers and travelers can enjoy satellite-transmitted television programs on their moving vehicles.

1.6 Contributions

One of the biggest applications of this project is in Recreational vehicles (RVs).

1.7 Organization

The first part of thesis is the abstract, which describes the main details of our project, followed by the introduction section, which specifies the problem statement, approach, scope and objectives. The literature review section states the various resources read onlineand the research work carried out before the commencement of the project. The design and development part illustrates the modules, blocks and hardware used in the project. The analysis and evaluation part give details of the block diagram, design specifications and detailed plan of the project including codes. The future work gives further improvements and point out additional developments, which can be made to enhance the scope of project.

CHAPTER:2

LITERATURE REVIEW

2 LITERATURE REVIEW

The literature available for this project is explained below:

2.1 GPS signal

Global Positioning System (**GPS**) satellites broadcast microwave **signals** to enable GPS receivers on or near the Earth's surface to determine location and time and derive velocity. The GPS system itself is operated by the U.S. Department of Defense (DoD) for use by both the military and the general public.

GPS signals include ranging signals, used to measure the distance to the satellite, and navigation messages. The navigation messages include *ephemeris* data, used to calculate the position of each satellite in orbit, and information about the time and status of the entire satellite constellation, called the *almanac*.

There are four signals available for civilian use. In order of date of introduction, these are: L1 C/A, L2C, L5 and L1C. L1 C/A is also called the *legacy signal* and is broadcast by all satellites. The other signals are called *modernized signals* and are not broadcast by all satellites. In addition, there are *restricted signals* with published frequencies and chip rates but encrypted coding intended to be used only by authorized parties. Some limited use of restricted signals can still be made by civilians without decryption; this is called *codeless* and *semi-codeless* access, and is officially supported.

Each GPS satellite continuously transmits a microwave radio signal composed of two carriers, two codes, and a navigation message. When a GPS receiver is switched on, it will pick up the GPS signal through the receiver antenna. Once the receiver acquires the GPS signal, it will process it using its built-in software. The partial outcome of the signal processing consists of the distances to the GPS satellites through the digital codes (known as the pseudo ranges) and the satellite coordinates through the navigation message. Theoretically, only three distances to three simultaneously tracked satellites are needed. In this case, the receiver would be located at the intersection of three spheres; each has a radius of one receiver-satellite distance and is centered on that particular satellite. It can be seen in the figure given below.

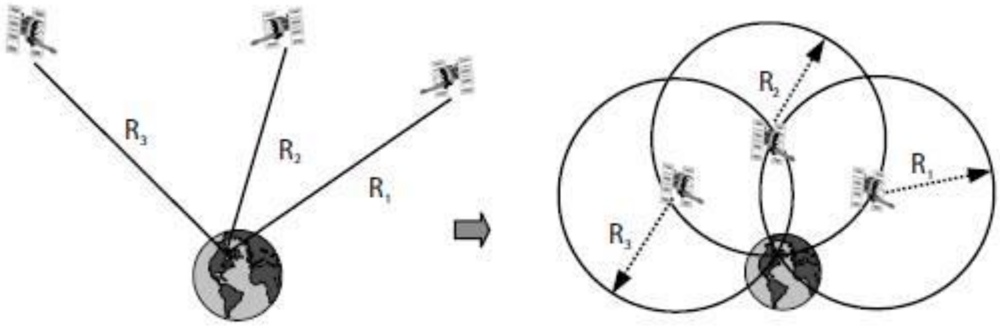


Figure [2. 1]: GPS Positioning

GPS signals are radio signals which use a specific frequency. There are actually two major frequencies that GPS works on, one of which is for public, non-military use (1575.42 MHz) and the other is used solely for the military (1227.6 MHz).

Receiver position is calculated from the position of satellites and the distances to them. Distance is calculated from the time a radio signal travels between satellite and receiver.

Basically, satellite positioning is a trilateration problem. From the known position of three satellites and the measured distances between them and the receiver, coordinates of receiver position can be calculated. The distances are determined by multiplying the travelling time of the radio signals by the speed of light.

In our project, we are using u-blox NEO-6M GPS module to get current position of our dish on the moving vehicle. From the position coordinates, we calculate the desired angles of elevation and azimuth of the dish to get perfect signals from satellite.

2.2 Low noise block (LNB)

The abbreviation LNB stands for **Low Noise Block**. It is the device on the front of a satellite dish that receives the very low level microwave signal from the satellite, amplifies it, changes the signals to a lower frequency band and sends them down the cable to the indoor receiver.

The expression **low noise** refers the quality of the first stage input amplifier transistor. The quality is measured in units called Noise Temperature, Noise Figure or Noise Factor. Both Noise Figure and Noise Factor may be converted into Noise Temperature. The lower the Noise Temperature the better. So an LNB with Noise Temperature = 100K is twice as good as one with 200K. C band LNBs tend have the lowest noise temperature performance while Ka LNBs have the highest (worst).

The expression **Block** refers to the conversion of a block of microwave frequencies as received from the satellite being down-converted to a lower (block) range of frequencies in the cable to the receiver. Satellites broadcast mainly in the range 4 to 12 to 21 GHz.

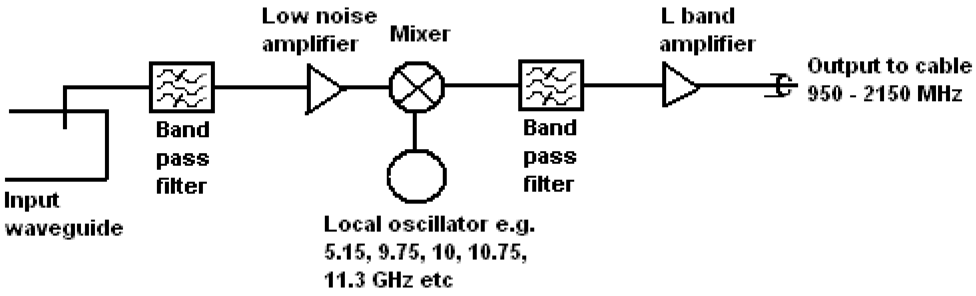


Figure [2. 2]: Low noise block down converter (LNB) diagram

The diagram shows the input waveguide on the left which is connected to the collecting feed or horn. As shown there is a vertical pin through the broad side of the waveguide that extracts the vertical polarization signals as an electrical current. The satellite signals first go through a band pass filter which only allows the intended band of microwave frequencies to pass through. The signals are then amplified by a Low Noise Amplifier and thence to the Mixer.

At the mixer-stage, the signals let through by the input band pass-filter are mixed with the signal generated by the local-oscillator and this process creates a multitude of sum and difference signals.

Amongst the mixer output products are the difference frequencies between the wanted input signal and the local oscillator frequency. These are the ones of interest. The second band pass filter selects these and feeds them to the output L band amplifier and into the cable. Typically the output frequency = input frequency - local oscillator frequency. In some cases it is the other way round so that the output frequency = local oscillator frequency - input frequency. In this case the output spectrum is inverted.

Examples of input receive frequency band, LNB local oscillator frequency and output frequency band are shown below.

C band is 3.4 - 4.8 GHz. Ku band is 10.7 - 12.75 GHz. Ka band is 19.2 - 21.2 GHz.

Compared with C-band, K_u band is not similarly restricted in power to avoid interference with terrestrial microwave systems, and the power of its uplinks and downlinks can be increased. This higher power also translates into smaller receiving dishes and points out a generalization between a satellite's transmission and a dish's size. As the power increases, the size of an antenna's dish will decrease. This is because the purpose of the dish element of the antenna is to collect the incident waves over an area and focus them all onto the antenna's actual receiving element, mounted in front of the dish (and pointed back towards its face); if the waves are more intense, fewer of them need to be collected to achieve the same intensity at the receiving element.

A major attraction of the band over lower frequency microwave bands is that the shorter wavelengths allow sufficient angular resolution to separate the signals of different communication satellites to be achieved with smaller terrestrial parabolic antennas. From the Rayleigh criterion, the diameter of a parabolic dish required to create a radiation pattern with a given angular beamwidth (gain) is proportional to the wavelength, and thus inversely proportional to the frequency. At 12 GHz a 1-meter dish is capable of focusing on one satellite while sufficiently rejecting the signal from another satellite only 2 degrees away. This is important because satellites in FSS (Fixed Satellite Service) service (11.7-12.2 GHz in the U.S.) are only 2 degrees apart. At 4 GHz (C-band) a 3-meter dish is required to achieve this narrow angular resolution. Note the inverse linear correlation between dish size and frequency. For K_u satellites in DBS (Direct Broadcast Satellite) service (12.2-12.7 GHz in the U.S.) dishes much smaller than 1-meter can be used because those satellites are spaced 9 degrees apart. As power levels on both C and K_u band satellites have increased over the years, dish beam-width has become much more critical than gain.

The K_u band also offers a user more flexibility. A smaller dish size and a K_u band system's freedom from terrestrial operations simplifies finding a suitable dish site. For the end users K_u band is generally cheaper and enables smaller antennas (both because of the higher frequency and a more focused beam).[6] K_u band is also less vulnerable to rain fade than the K_a band frequency spectrum. We are using ku-band LNB in our project.

2.3 Azimuth and elevation angles

2.3.1 Azimuth

The angle of rotation (horizontal) that a ground based parabolic antenna must be rotated through to point to a specific satellite in a geosynchronous orbit. The azimuth angle for any particular satellite can be determined for any point on the surface of the earth giver the latitude and longitude of that point. It is defined with respect to due north as a matter of easy convenience. By definition North is 0 deg, East is 90 deg, South is 180 deg and west is 270 deg. North can also be called 360 deg.

Note that you find a satellite by pre-setting the elevation accurately and then swinging the whole antenna boldly in azimuth till the signal locks up - so an approximate azimuth angle is normally sufficient.

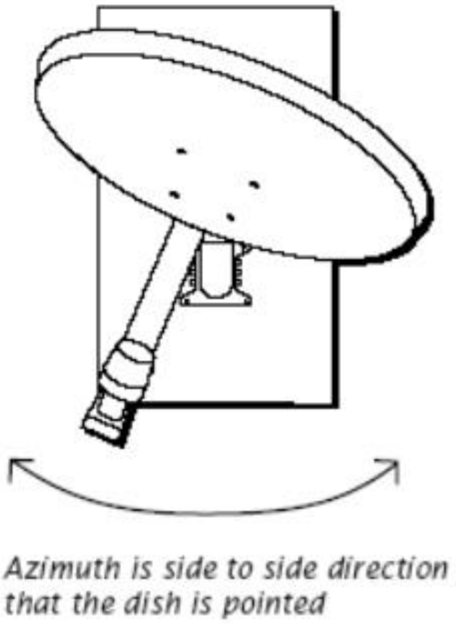


Figure [2. 3]: Azimuth angle

2.3.2 Elevation

The upward tilt to a satellite antenna measured in degrees required to aim the antenna at the communications satellite. When aimed at the horizon, the elevation angle is zero. If it were tilted to a point directly overhead, the satellite antenna would have an elevation of 90 degrees. When your dish is pointed low down near the horizon the elevation angle is only a few degrees. At low elevation angles, below 5 deg at C band and 10 deg at Ku band, the path through the atmosphere is longer and the signals are degraded by rain attenuation and rain thermal noise. Scintillation also occurs, particularly in hot humid weather. This causes increases and decreases in the signal level every several seconds for many hours, like the twinkling of a star. When your dish is pointed almost straight up the elevation angle is nearly 90 degrees. Sites near the equator may require you to point to almost 90 degrees elevation angle when the longitude of the satellite is similar to the longitude of the site location. In high elevation cases watch out for the possibility of rain water collecting in the dish.

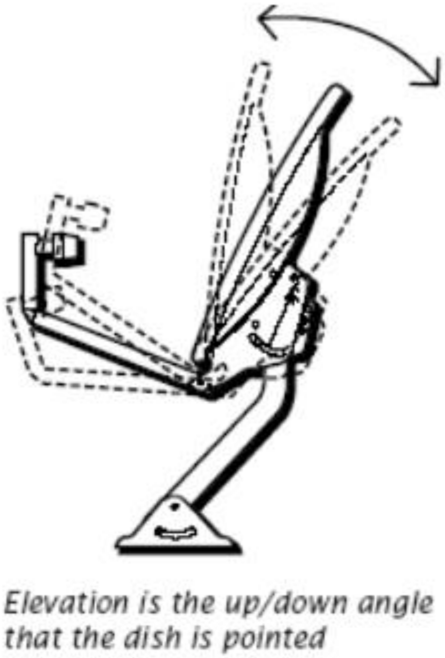


Figure [2. 4]: Elevation angle

CHAPTER:3

DESIGN AND DEVELOPMENT

3 DESIGN AND DEVELOPMENT

3.1 Overview

The two-axis control system for a receiving antenna (BSS) over a moving platform will provide arrangements for changing or varying the orientation or the shape of the directional pattern of the waves radiated from an antenna using mechanical movement of antenna for varying two co-ordinates of the orientation (elevation and azimuth angles). As a result, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

The details for the design of the project are given as below.

3.2 Electrical Design

The project consists of the following modules/blocks:

- Dish Antenna
- LNB
- LNB Mount
- Antenna Mount/Frame with motor placement
- ATmega328Px2
- Stepper Motors (Nema17) x2
- Dual H bridge motor drivers L298N x2
- MPU6050 gyroscope
- HMC5883 magnetometer
- NEO-6M GPS module
- PID controller
- 12V DC battery

3.2.1 Dish antenna

A **satellite dish** is a dish-shaped type of parabolic antenna designed to receive or transmit information by radio waves to or from a communication satellite. The term most commonly means a dish used by consumers to receive direct-broadcast satellite television from a direct broadcast satellite in geostationary orbit. A dish antenna, also known simply as a *dish*, is common in microwave systems. This type of antenna can be used for satellite communication and broadcast reception, space communications, radio astronomy, and radar. A dish antenna consists of an active, or driven, element and a passive parabolic or spherical reflector. The driven element can be a dipole antenna or a horn antenna. If a horn is used, it is aimed back at the center of the reflecting dish. The reflector has a diameter of at least several wavelengths. As the wavelength increases(and the frequency decreases), the minimum required dish diameter becomes larger. When the dipole or horn is properly positioned and aimed, incoming electromagnetic fields bounce off the reflector, and the energy converges on the driven element. If the horn or dipole is connected to a transmitter, the element emits electromagnetic waves that bounce off the reflector and propagate outward in a narrow beam. A dish antenna is usually operated with an unbalanced feed line. For satellite television reception, coaxial cable is used. In applications such as radar where a high-power signal is transmitted, a feed system is preferred.

A 4 feet antenna dish is used in our project.

Figure [3. 1]: Dish Antenna

3.2.2 LNB

Most of a satellite dish is just plain metal. The only part that contains any electronic components at all is in the front of the metal arm. It is generally referred to as an LNB or sometimes LNBF, which stands for Low-Noise Block down converter. (Depending on your satellite TV system, it can also perform other tasks). An LNB has two important functions: it is a low-noise amplifier, meaning that it takes the extremely weak satellite signal and amplifies it. It is also a block down converter, meaning it takes the signals on the super-high satellite frequencies and converts them to lower frequencies. Both functions are important in order to deliver a satellite signal through a regular cable. LNB’s and LNBF’s are both amplifiers used in satellite dishes. As with other signal amplifiers, they take the very faint signal they receive and magnify it so that it is powerful enough to use. This is the first step in taking the microwave signal coming from space and turning it into images and sounds for televisions and computers. A simple LNB attaches to the feedhorn of a satellite dish. A Feedhorn is basically a metal funnel that guides the incoming signal to the actual antenna stub inside the throat of the LNB. As you switch channels, the LNB switches polarity through the use of an exterior motor.

Satellites use comparatively high radio frequencies (microwaves) to transmit their TV signals. As microwave satellite signals do not easily pass through walls, roofs, or even glass windows, it is preferable for satellite antennas to be mounted outdoors. However, plastic glazing is transparent to microwaves and residential satellite dishes have successfully been hidden indoors looking through acrylic or polycarbonate windows to preserve the external aesthetics of the home.

The purpose of the LNB is to use the super heterodyne principle to take a block (or band) of relatively high frequencies and convert them to similar signals carried at a much lower frequency (called the intermediate frequency or IF). These lower frequencies travel through cables with much less attenuation, so there is much more signal left at the satellite receiver end of the cable. It is also much easier and cheaper to design electronic circuits to operate at these lower frequencies, rather than the very high frequencies of satellite transmission.

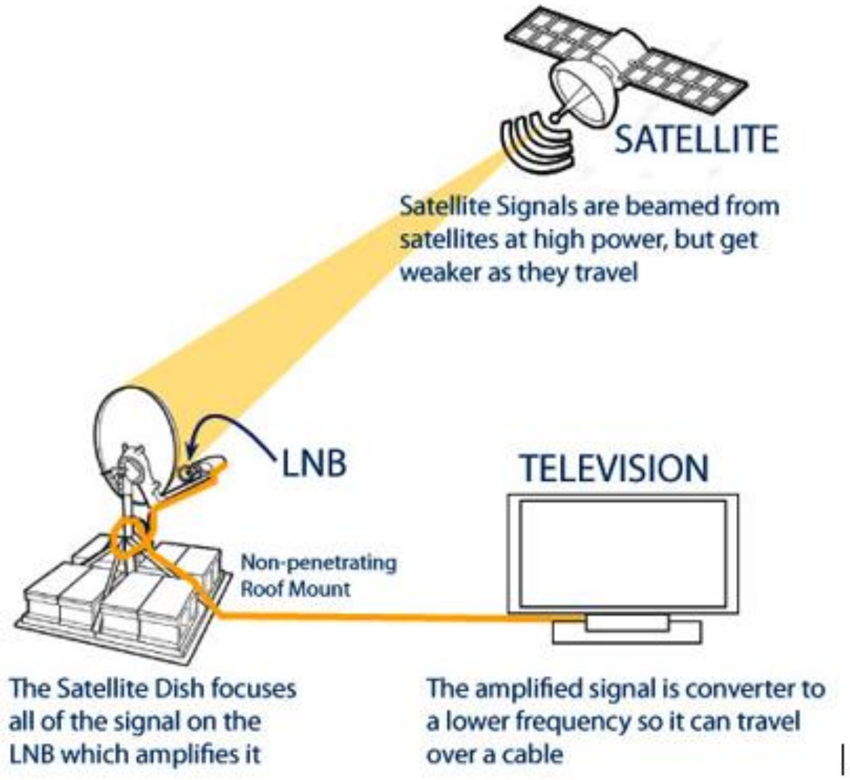


Figure [3. 2]: Working of LNB

We are using K_u-band LNB in our project. Ku band is 10.7 - 12.75 GHz.

The frequency conversion is performed by mixing a fixed frequency produced by a local oscillator inside the LNB with the incoming signal, to generate two signals equal to the sum of their frequencies and the difference. The frequency sum signal is filtered out and the frequency difference signal (the IF) is amplified and sent down the cable to the receiver:

K_u-band:

$$f_{IF} = f_{rec} - f_{LO}$$

where “f”is a frequency. The local oscillator frequency f_{LO} determines what block of incoming frequencies is downconverted to the frequencies expected by the receiver.

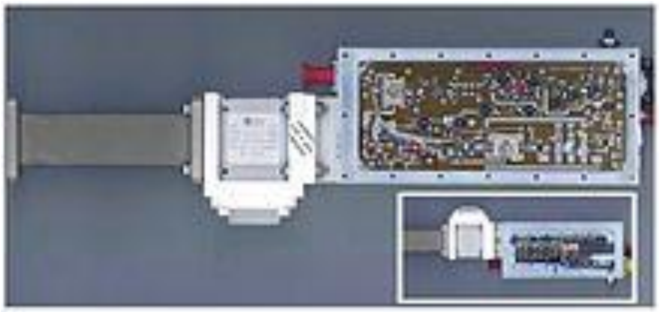


Figure [3. 3]: Ku-band LNB

The following **Block** refers to the conversion of a block of microwave frequencies as received from the satellite being down-converted to a lower (block) range of frequencies in the cable to the receiver. Satellites broadcast mainly in the range 4 to 12 to 21 GHz.

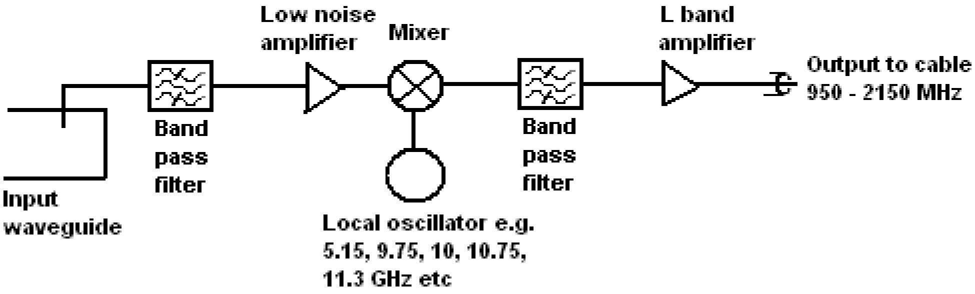


Figure [3. 4]: LNB down converter diagram

This is a listing of common Ku-band LNB frequency ranges and their associated local oscillator frequencies.

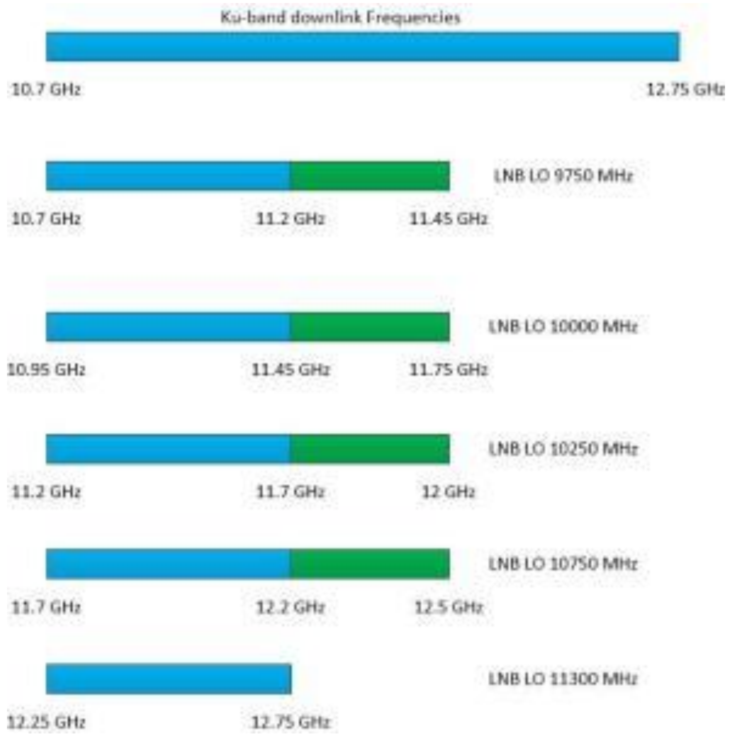


Figure [3. 5]: ku-band downlink frequencies

The Ku-band downlink frequency range is 10.7 to 12.75 GHz. In the diagram above, the blue areas for each LNB are for an input range of 950 to 1450 MHz. Some LNBs have an extended range up to 1700 or 1750 MHz. Here are a few items of note:

- The 10.7 to 10.95 frequency range is very rare. Very few LNBs can receive these low frequencies
- The 11.7 to 12.2 GHz range is used in North America
- Most LNBs will have LO frequencies of 10 GHz, 10.75 GHz, or 11.3 GHz depending on the application.
- This is not an exhaustive list. Other LOs are used based on various factors.

Before implementing any LNB, refer to the specification sheets for the particular LNB. Ensure that the LNB will meet the requirements of the satellite circuit.

3.2.3 LNB mount

LNB mount is used for mounting the LNB.

Figure [3. 6]: ku-band LNB mount

3.2.4 Antenna mount/ Frame with motor placement

Antenna mount is used for mounting the dish antenna on the vehicle.

Figure [3. 7]: Antenna mount

3.2.5 ATmega328P microcontroller

ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards. The Atmel 8-bitAVRRISC-based microcontroller combines 32 kB ISPflash memory with read-while-write capabilities, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz. The ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed

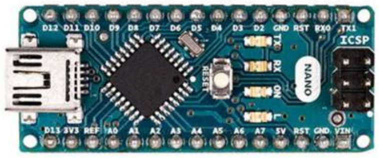


Figure [3. 8]: ATmega328P microcontroller

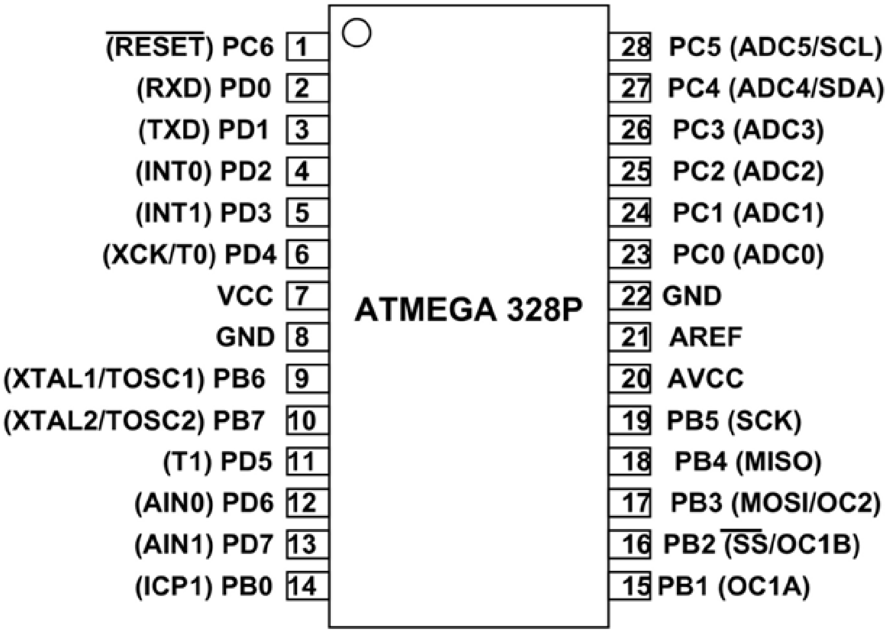


Figure [3. 9]: ATmega328P pinout

3.2.6 Stepper motors (Nema 17)

A stepper motor is an electromechanical device it converts electrical power into mechanical power. Also it is a brushless, synchronous electric motor that can divide a full rotation into an expansive number of steps. The motor’s position can be controlled accurately without any feedback mechanism, as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors.

The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided. The stator has eight poles, and the rotor has six poles. The rotor will require 24 pulses of electricity to move the 24 steps to make one complete revolution. Another way to say this is that the rotor will move precisely 15° for each pulse of electricity that the motor receives.

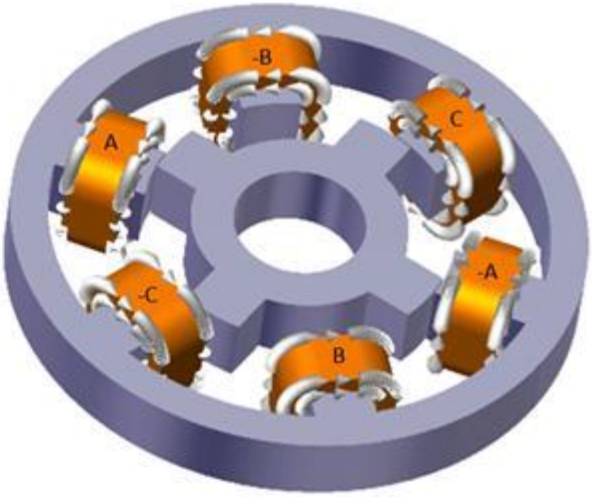


Figure [3. 10]: Stepper motor

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time. With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications. Stepper motors come in many different sizes and styles and electrical characteristics. This guide details what you need to know to pick the right motor for the job.

A **NEMA 17 stepper motor** is a stepper motor with a 1.7 x 1.7 inch (43.18 x 43.18 mm) faceplate. The NEMA 17 is larger and generally heavier than for example a NEMA 14, but this also means it has more room to put a higher torque. However, its size is *not* an indication of its power. We are using two Nema 17 stepper motors in this project.

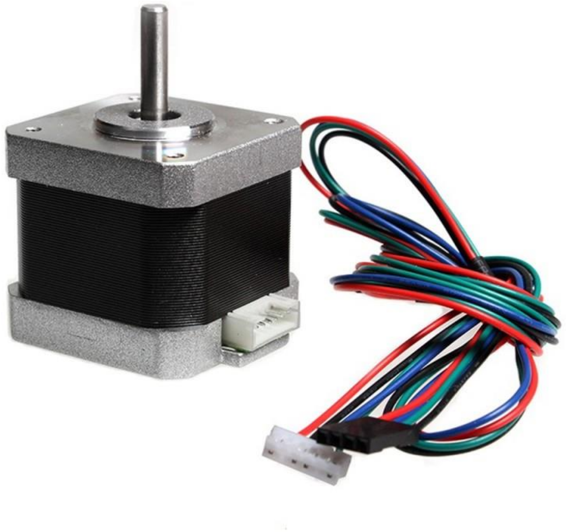


Figure [3. 11]: Stepper motor (Nema 17)

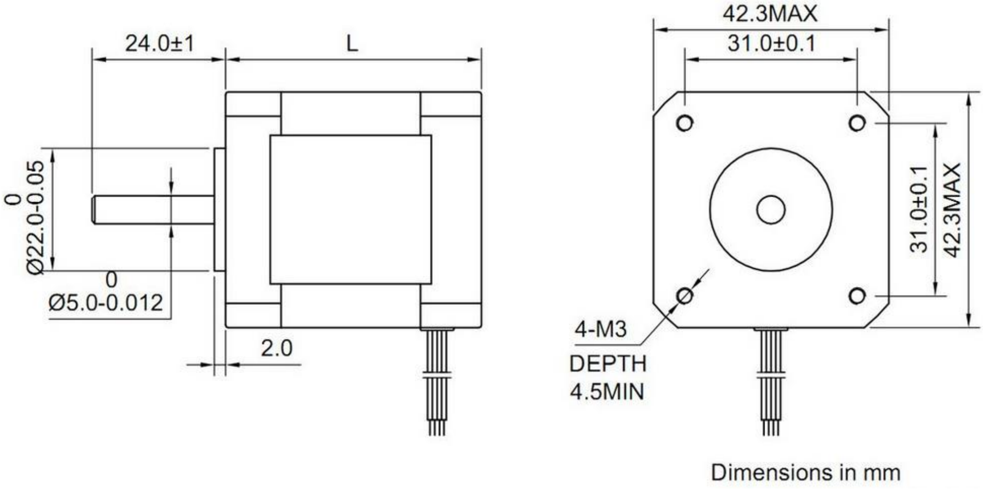


Figure [3. 12]: Nema 17 dimensions

3.2.7 Dual H bridge motor drivers (L298N)

A motor driver is a little current amplifier; the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

This dual bidirectional motor driver is based on the very popular L298 Dual H-Bridge Motor Driver IC. This is a 5 - 35 V, 2A Dual Motor Controller. It can bear larger current due to the increased heat sink dissipation. It is easy to control, using LGS's outstanding high-power motor driver chip, the L298N. This chip allows for direct drive of two bi-directional DC motors or a bipolar stepper. Drive current up to 2A per motor output. The driver uses a broad-brush design to reduce wire resistance.

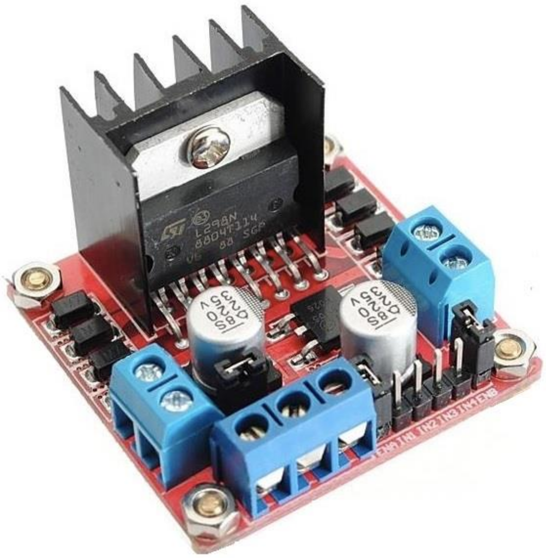
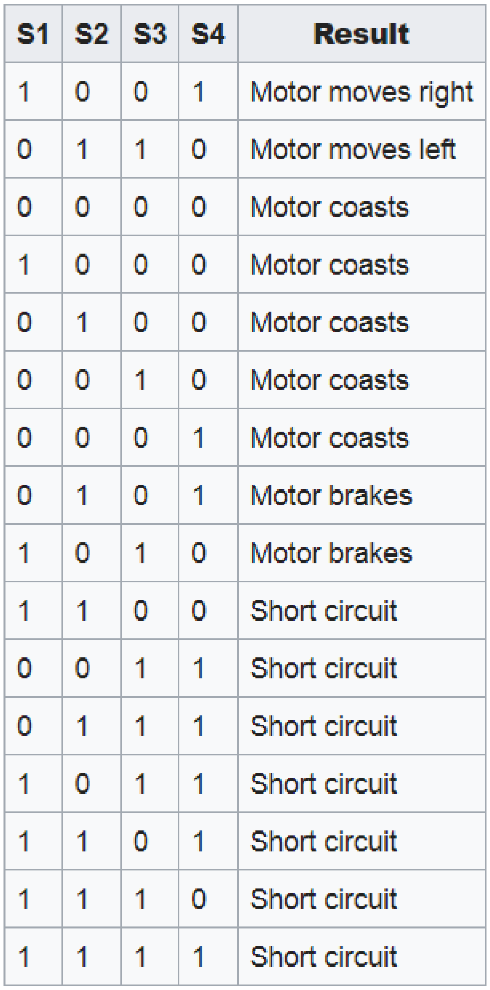


Figure [3. 13]: Motor driver L298N

An **H bridge** is an electronic circuit that enables a voltage to be applied across a load in opposite direction. The H-bridge arrangement is generally used to reverse the polarity/direction of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted, or to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit. The following table summarizes operation, with S1-S4 corresponding to the diagram.



S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor coasts
1	0	0	0	Motor coasts
0	1	0	0	Motor coasts
0	0	1	0	Motor coasts
0	0	0	1	Motor coasts
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Short circuit
0	0	1	1	Short circuit
0	1	1	1	Short circuit
1	0	1	1	Short circuit
1	1	0	1	Short circuit
1	1	1	0	Short circuit
1	1	1	1	Short circuit

Figure [3. 14]: H bridge S1-S4 Operation

3.2.8 MPU6050 Gyroscope

The InvenSense MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefor it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino.

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers. The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System(MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

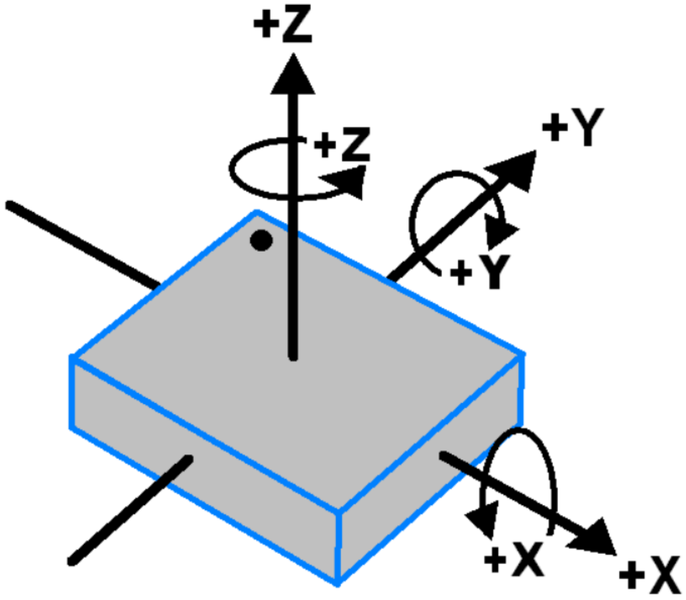


Figure [3. 15]: MPU6050 Orientation and polarity of rotation

When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.

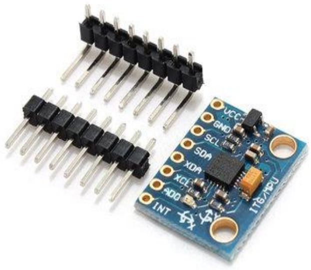


Figure [3. 16]: MPU6050

3.2.9 HMC5883 Magnetometer

The Honeywell HMC5883 is a surface mount multi-chip module designed for low field magnetic sensing with a digital interface for applications such as low cost compassing and magnetometry. The HMC5883 includes our state of the art, high-resolution HMC118X series magneto-resistive sensors plus Honeywell developed ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, 12-bit ADC that enables 1° to 2° compass heading accuracy. The I2C serial bus allows for easy interface. The HMC5883 is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883 include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.

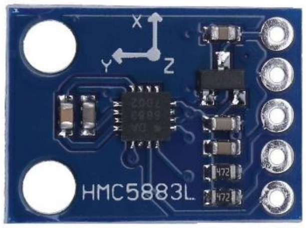


Figure [3. 17]: HMC5883

3.2.10 U-bloxNEO-6M GPS module

This is a complete GPS module that is based on the Ublox NEO-6M. This unit uses the latest technology from Ublox to give the best possible positioning information and includes a larger built-in 25 x 25mm active GPS antenna with a UART TTL socket. A battery is also included so that you can obtain a GPS lock faster. This GPS module gives the best possible position information, allowing for better performance with your Ardupilot or other Multirotor control platform.

The Ublox NEO-6M GPS engine on this board is a quite good one, with the high precision binary output. It has also high sensitivity for indoor applications. UBLOX NEO-6M GPS Module has a battery for power backup and EEPROM for storing configuration settings. The antenna is connected to the module through a ufl cable which allows for flexibility in mounting the GPS such that the antenna will always see the sky for best performance. This makes it powerful to use with cars and other mobile applications.

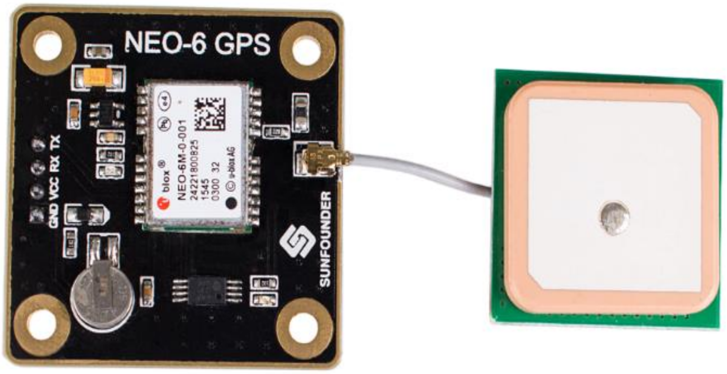


Figure [3. 18]: U-blox NEO-6M GPS module

3.2.11 PID controller

The best-known controllers used in industrial control processes are proportional-integral-derivative (PID) controllers because of their simple structure and robust performance in a wide range of operating conditions. The design of such a controller requires specification of three parameters: proportional gain, integral time constant and derivative time constant.

Setting the PID parameters are called tuning, the classical tuning method is to use the famous Ziegler-Nichols tuning formula. This tuning formula however, is not always effective in the sense that it may completely fail to tune the processes with, for example, relatively large dead time. Moreover, its tuning will have to be supplemented with purely experience-based fine-tuning to meet the specifications. The parameters of the PID controller kp, ki and kd (or kp, Ti and Td) can be manipulated to produce various response curves.

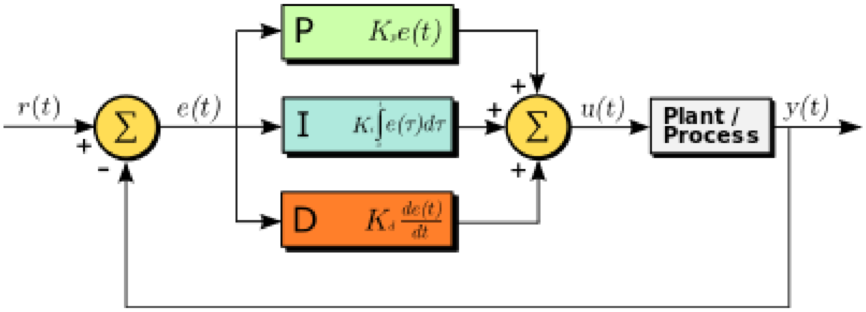


Figure [3. 19]: Block diagram of PID controller

The distinguishing feature of the PID controller is the ability to use the three *control terms* of proportional, integral and derivative influence on the controller output to apply accurate and optimal control. The block diagram on the right shows the principles of how these terms are generated and applied. It shows a PID controller, which continuously calculates an error value as the difference between a desired set point and a measured process variable, and applies a correction based on proportional, integral, and derivative terms. The controller attempts to minimize the error over time by adjustment of a *control variable*, such as the opening of a control valve, to a new value determined by a weighted sum of the control terms.

Term **P** is proportional to the current value of the SP − PV error $e(t)$. For example, if the error is large and positive, the control output will be proportionately large and positive, taking into account the gain factor "K". Using proportional control alone in a process with compensation such as temperature control, will result in an error between the setpoint and the actual process value, because it requires an error to generate the proportional response. If there is no error, there is no corrective response

Term **I** accounts for past values of the SP − PV error and integrates them over time to produce the I term. For example, if there is a residual SP − PV error after the application of proportional control, the integral term seeks to eliminate the residual error by adding a control effect due to the historic cumulative value of the error. When the error is eliminated, the integral term will cease to grow. This will result in the proportional effect diminishing as the error decreases, but this is compensated for by the growing integral effect.

Term **D** is a best estimate of the future trend of the SP − PV error, based on its current rate of change. It is sometimes called "anticipatory control", as it is effectively seeking to reduce the effect of the SP − PV error by exerting a control influence generated by the rate of error change. The rapid the change, the greater the controlling or dampening effect.

Tuning – The balance of these effects is achieved by "loop tuning" (see later) to produce the optimal control function. The tuning constants are shown below as "K" and must be derived for each control application, as they depend on the response characteristics of the complete loop external to the controller. These are dependent on the behavior of the measuring sensor, the final control element (such as a control valve), any control signal delays and the process itself. Approximate values of constants can usually be initially entered knowing the type of application, but they are normally refined, or tuned, by "bumping" the process in practice by introducing a set point change and observing the system response.

Control action – The mathematical model and practical loop above both use a "direct" control action for all the terms, which means an increasing positive error results in an increasing positive control output for the summed terms to apply correction. However, the output is called "reverse" acting if it is necessary to apply negative corrective action. For instance, if the valve in the flow loop was 100–0% valve opening for 0–100% control output – meaning that the controller action has to be reversed. Some process control schemes and final control elements require this reverse action. An example would be a valve for cooling water, where the fail-safe mode, in the case of loss of signal, would be 100% opening of the valve; therefore 0% controller output needs to cause 100% valve opening.

The overall control function can be expressed mathematically as

$$u(t) = k_p e(t) + k_i \int_0^t \quad e(t')d(t') + k_d \frac{de(t)}{dt}$$

where , k_p, k_i, k_dall non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively (sometimes denoted *P*, *I*, and *D*).

The following PID controller model is implemented on Simulink:

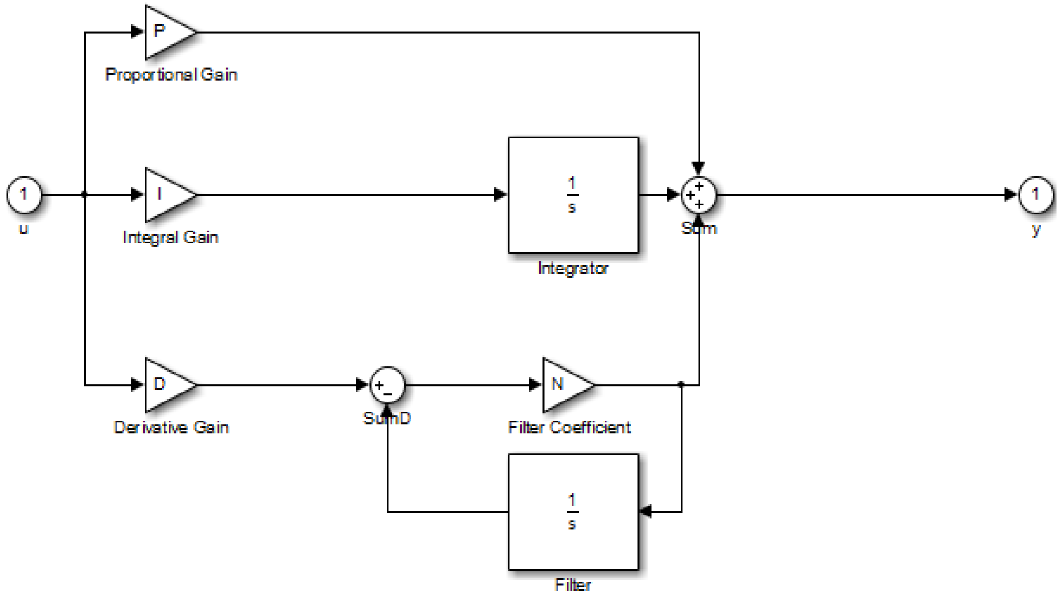


Figure [3. 20]: PID controller simulink model

3.2.1212V DC battery

DC batteries use direct current, which flows in a single direction and is generally used to power small appliances, radios, laptops, mobile phones and other electronic gadgets.

For power supply, we are using a 12V DC battery in our project.

Figure [3. 21]: 12 DC battery

3.3 Mechanical Design

Following are the components of our project which were first designed on solidworks and then fabricated:

3.3.1 V-link

Figure [3. 22]: V-link (solidworks design)

Figure [3. 23]: V-link (fabricated)

3.3.2 Motor

Figure [3. 24]: Motor (solidworks design)

Figure [3. 25]: Motor (fabricated)

3.3.3 Belt

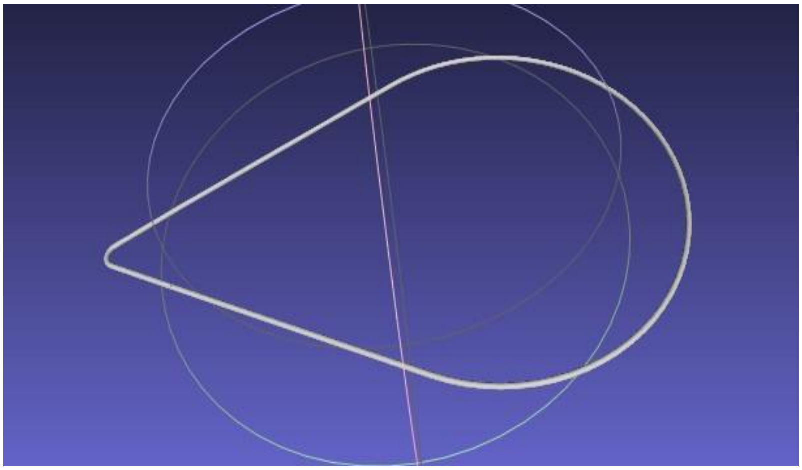


Figure [3. 26]: Belt (solidworks design)

Figure [3. 27]: Belt (fabricated)

3.3.4 Dish

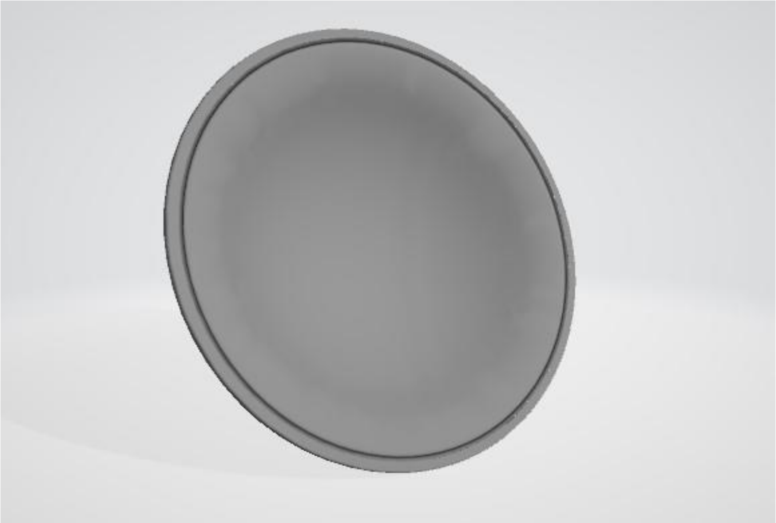


Figure [3. 28]: Dish (solidworks)

Figure [3. 29]: Dish (fabricated)

3.3.5 Dish base

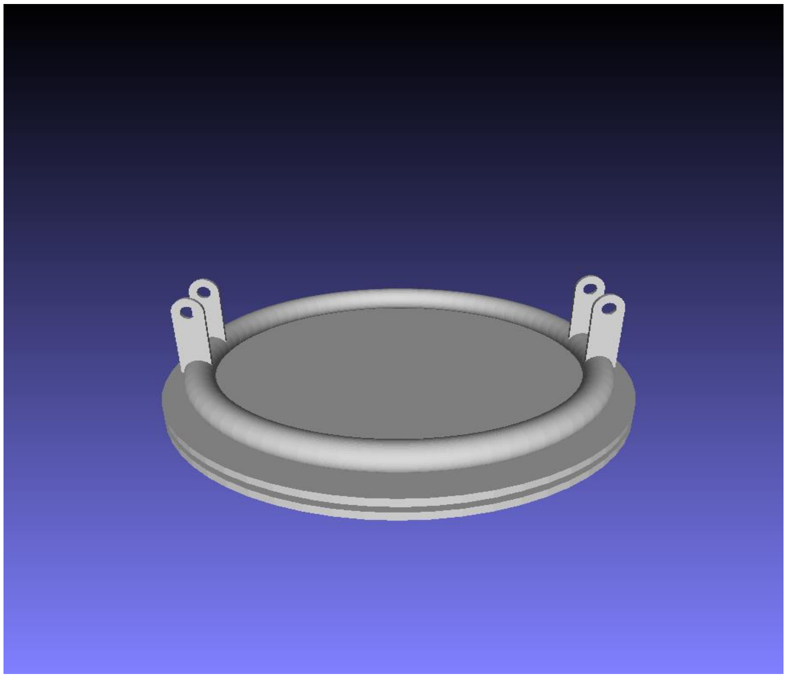


Figure [3. 30]: Dish base (solidworks design)

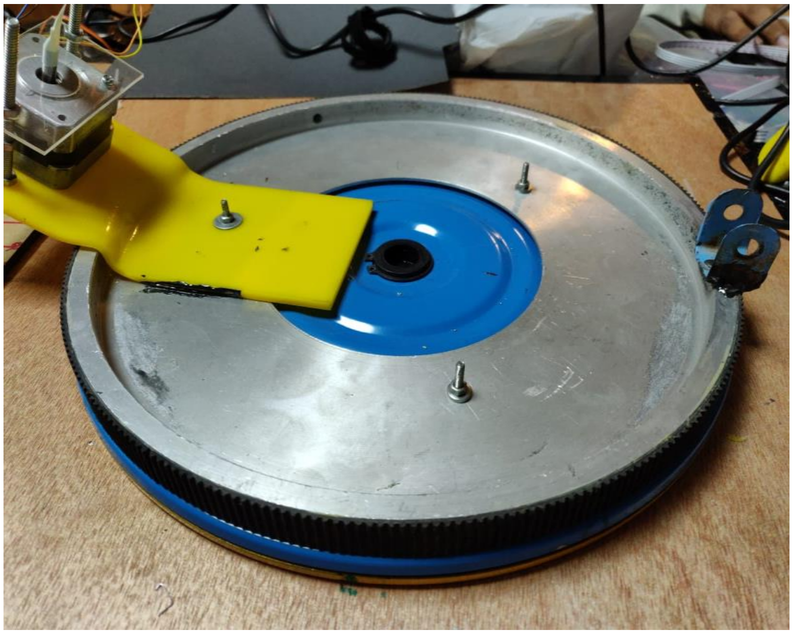


Figure [3. 31]: Dish base (fabricated)

3.3.6 Basic assembly of the project

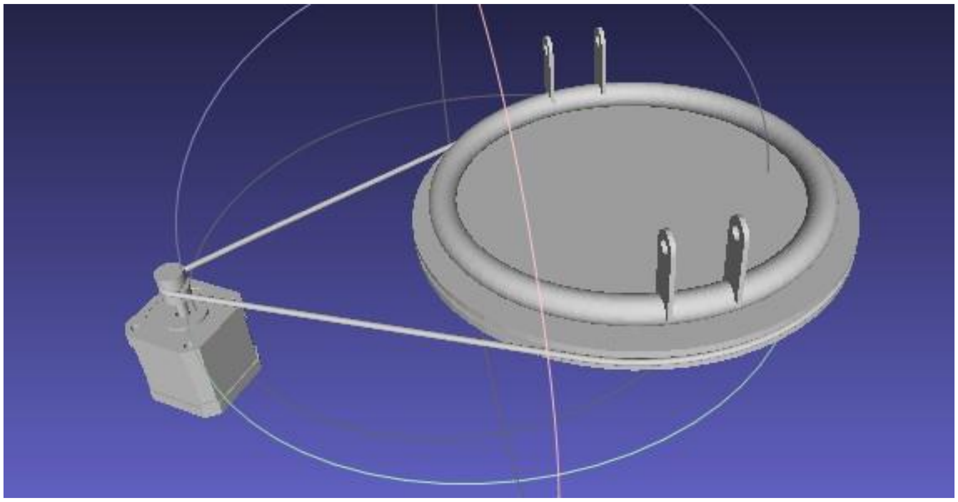


Figure [3. 32]: Basic assembly of the project (solidworks design)

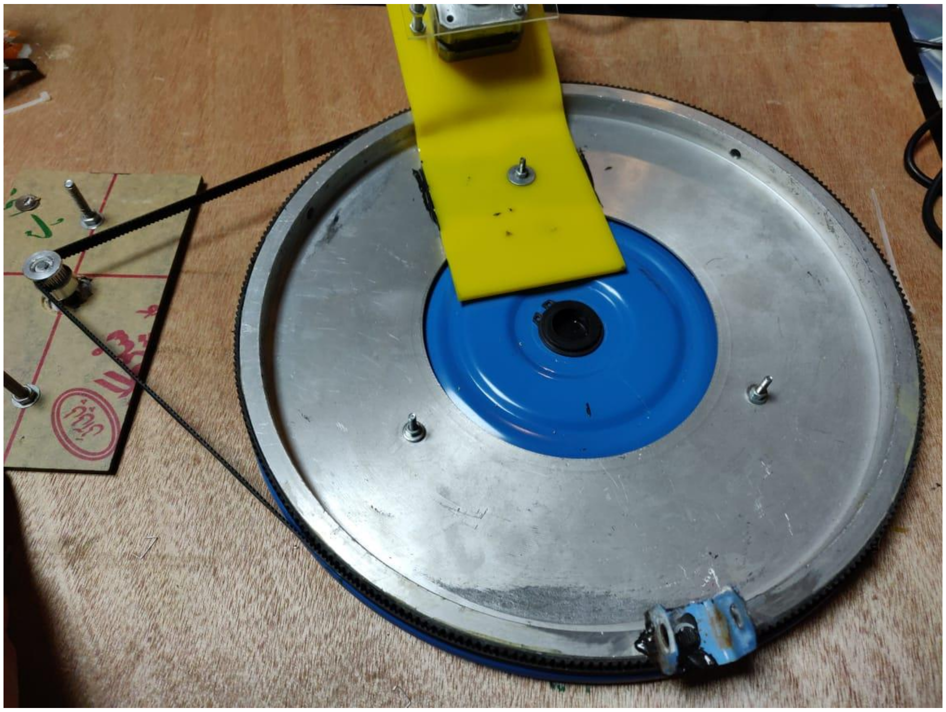


Figure [3. 33]: Basic assembly of the project (fabricated)

CHAPTER:4

ANALYSIS AND EVALUATION

4 ANALYSIS AND EVALUATION

4.1 Overview

The two-axis control system for a receiving antenna (BSS) over a moving platform provides arrangements for changing or varying the orientation or the shape of the directional pattern of the waves radiated from an antenna using mechanical movement of antenna for varying two co-ordinates of the orientation (elevation and azimuth angles). As a result, our dish antenna automatically adjusts to provide dish channels on our moving vehicle.

We have designed a two-axis control system for a receiving antenna (BSS) over a moving platform. We have used two microcontrollers (ATmega328P). To one microcontroller, magnetometer (HMC5883) is attached which provides us the azimuthal angle. To the second microcontroller, gyroscope (MPU6050) is attached which gives us the angle of elevation. So, we are using separate controllers for getting azimuthal and elevation angles. To the first microcontroller, GPS is also attached. The controller takes points from GPS and calculate the desired azimuthal and elevation angles.

The ATmega328p will compare these angles with current angles of the dish antenna, taken using HMC5883 and MPU6050 placed along the dish antenna. If the angles are not aligned, then the controller will adjust current position of our dish antenna by moving it through stepper motors connected. Once the respective angles provided by user are in accordance with dish angles, we will receive proper output from our dish receiver. In this way, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

4.2 Project block diagram

The complete block diagram of our project is as follows:

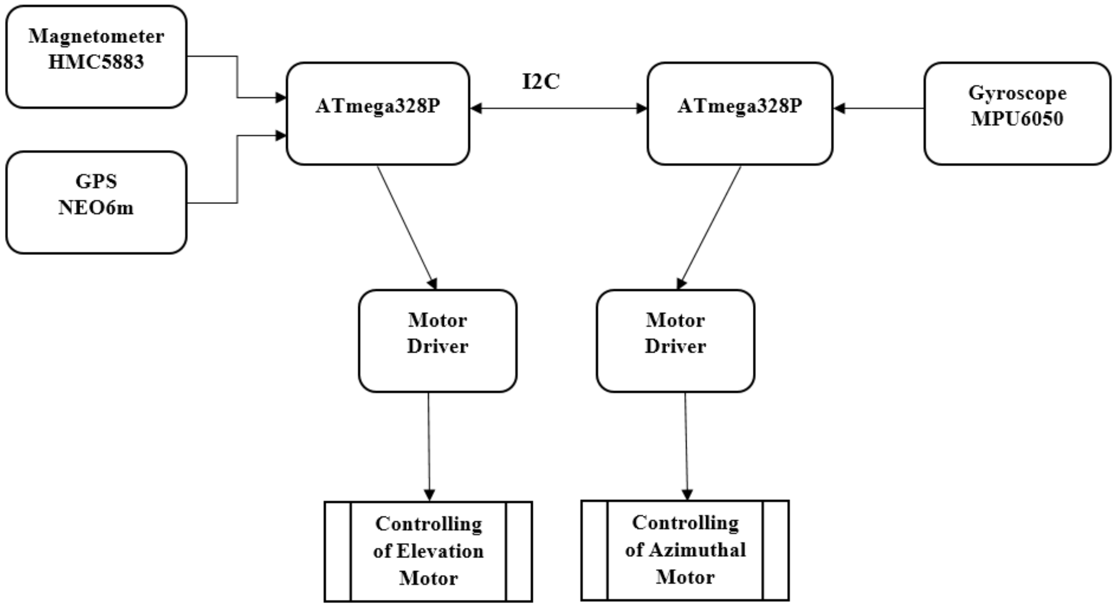


Figure [4.1]: Block diagram of the project

4.3 Design specifications

Our project involves Neo-6m that provides latitude and longitude to ATmega328p to calculate angles (elevation and azimuth). These are the desired angles of elevation and azimuth of the dish. Now to get current angle of elevation and azimuth, we used gyroscope MPU 6050. ATmega328P use calculated values to configure MPU 6050 gyroscope, calculate PID values and control actuators. In this way, ATmega328P controls the stepper motors. Stepper motors will control the dish and move the dish to the desired angle. As a result, the LNB receiver will receive signals and send it to the TV station to provide dish channels on our moving vehicles.

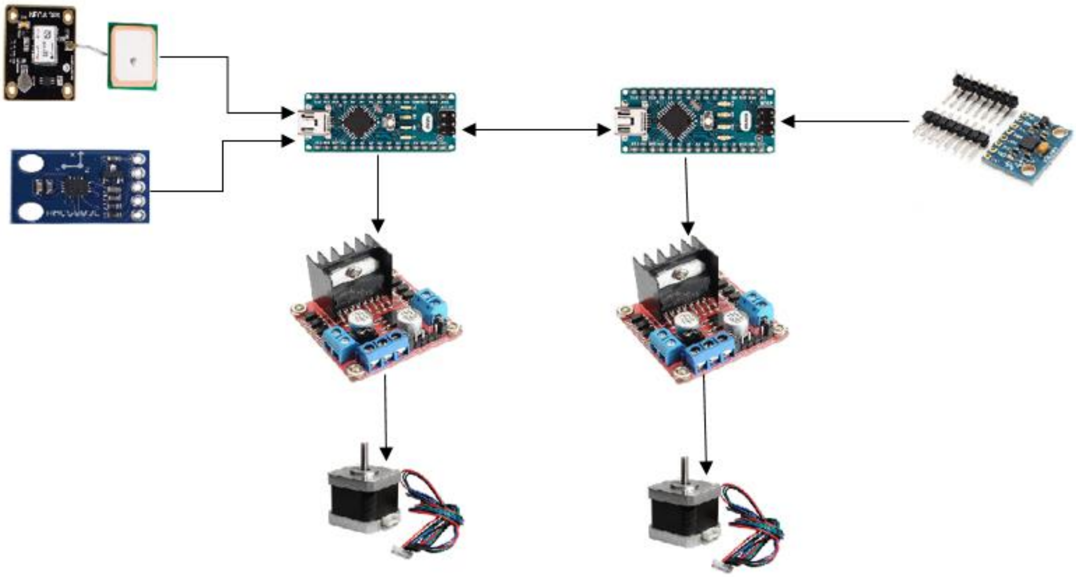


Figure [4.2]: Schematic diagram of the project

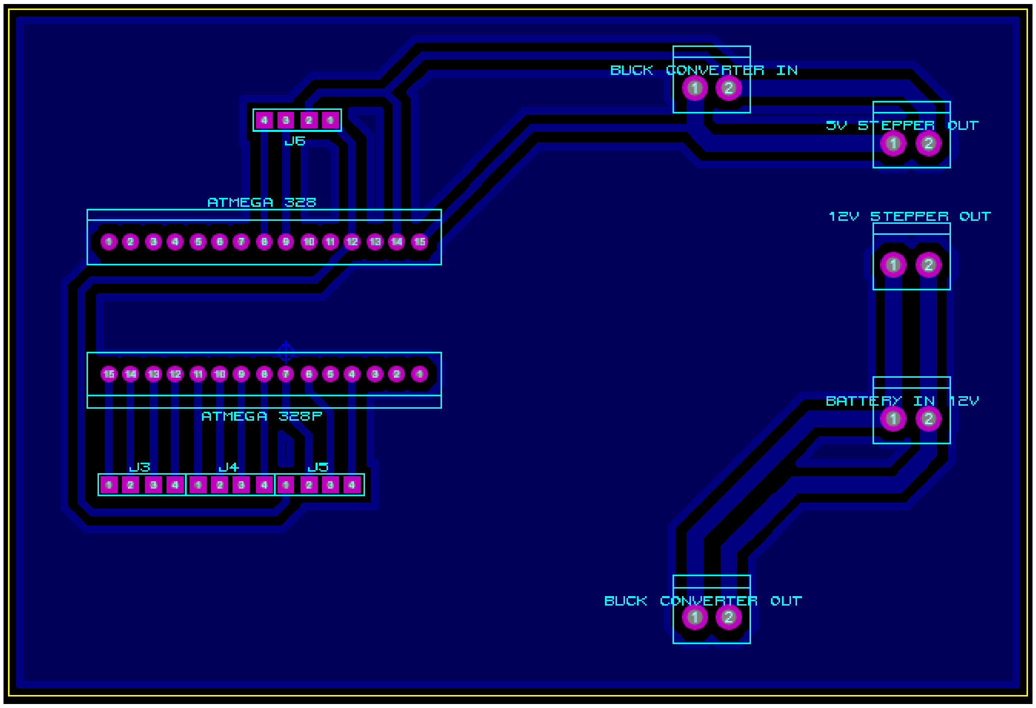


Figure [4.3]: PCB design

4.4 Detail plan

The GPS module u-blox NEO-6M connected to one ATmega328P is used to get the desired position of the dish.From the desired position, angle of elevation and azimuth are calculated. These are our desired angles to get perfect signals from satellite for our TV station. HMC 5883 magnetometer connected to one ATmega328P is used to get the current azimuthal angle while MPU6050 connected to the other ATmega328P is used to get the current elevation angle.The ATmega328p will compare these angles with current angles of the dish antenna, taken using HMC5883 and MPU6050 placed along the dish antenna. If the angles are not aligned, then the controller will adjust current position of our dish antenna by moving it through stepper motors connected. Once the respective angles provided by user are in accordance with dish angles, we will receive proper output from our dish receiver. In this way, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

Current angles – Desired angles = A

Where "**A**" is the difference between desired angle of elevation/ azimuth and the current angle of elevation/ azimuth

The stepper motors moves the dish by a value equal to A to get to the desired position. The LNB receiver will receive signals from the satellite and send it to the TV station. In this way we get clear channels on TV station of a recreational vehicle.

4.5 Final project

Following are the different project components:

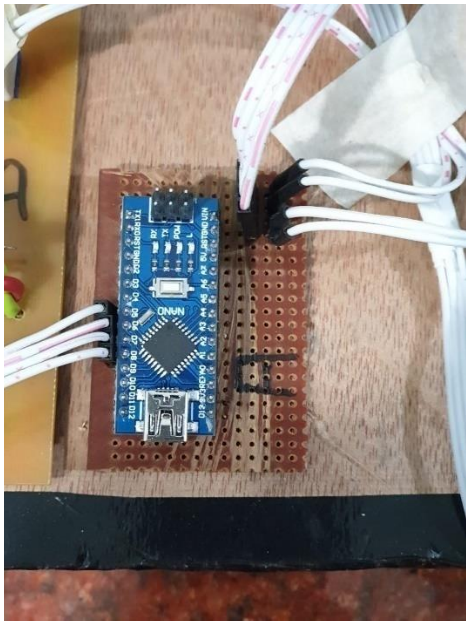


Figure [4.4]: ATmega328P (A)

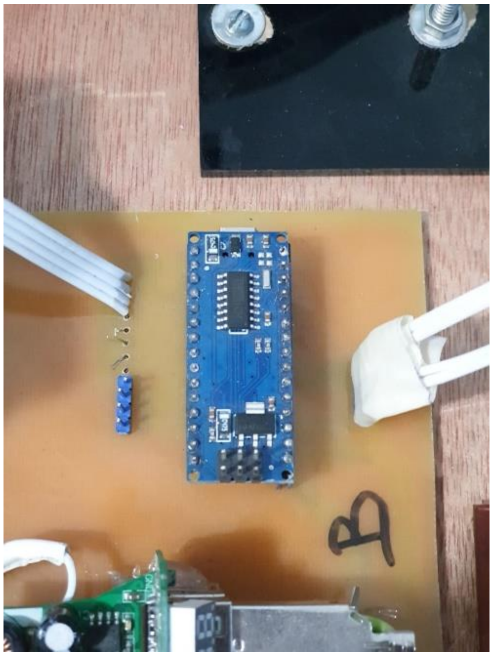


Figure [4.5]: ATmega328P (B)

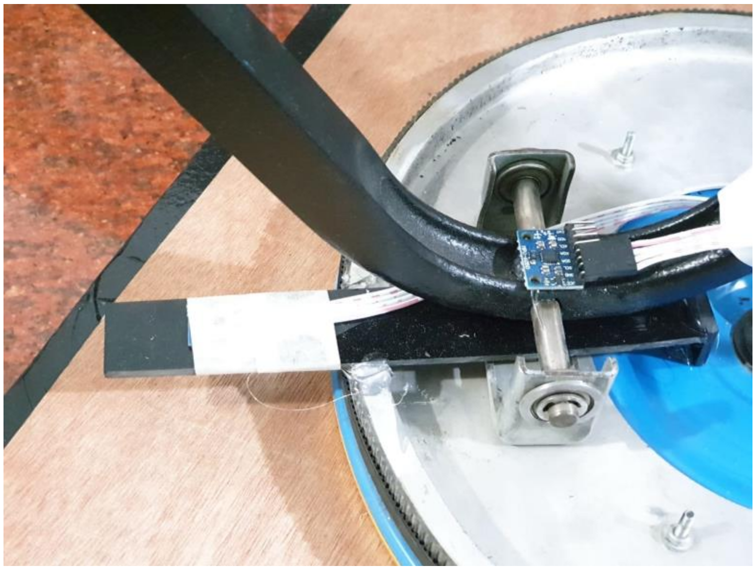


Figure [4.6]: MPU6050

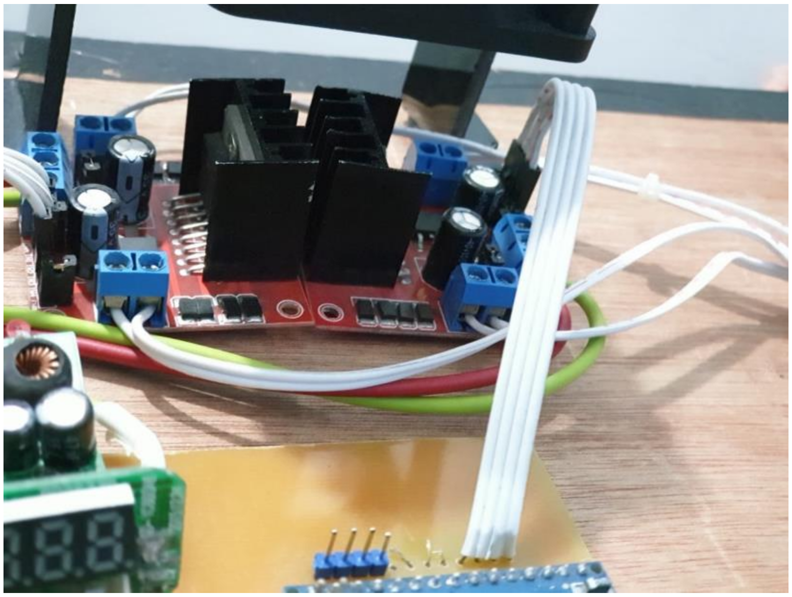


Figure [4.7]: L298 Motor driver

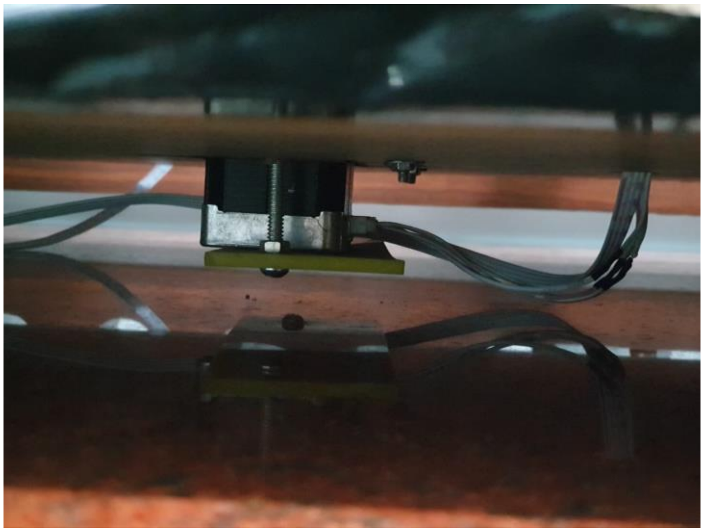


Figure [4.8]: Stepper motor (Azimuth)

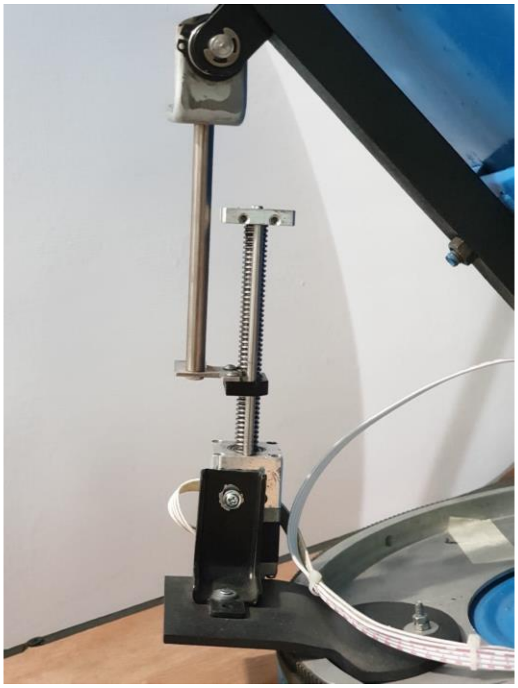


Figure [4.9]: Stepper motor (Elevation)

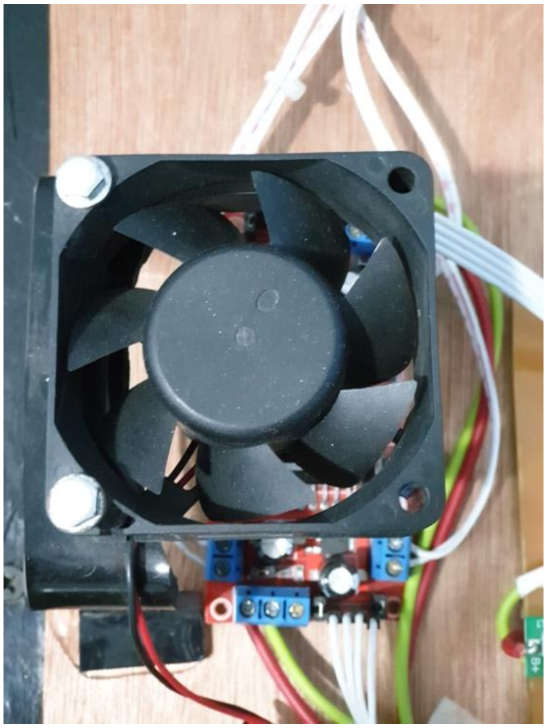


Figure [4.10]: Cooling fan

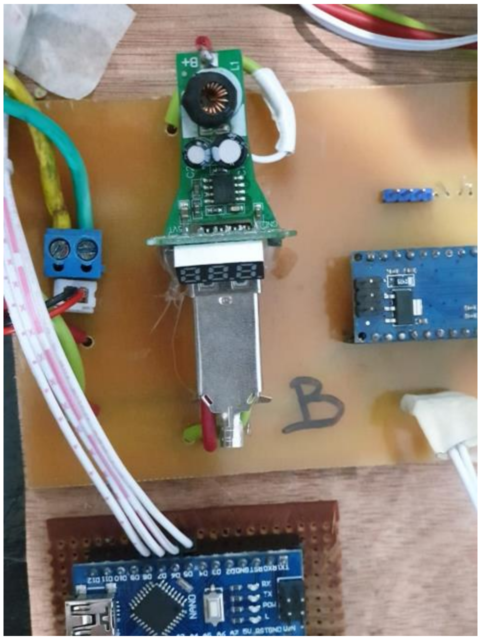


Figure [4.11]: 12V DC to 5V DC buck converter

Our final project looks like:

Figure [4.12]: Final Project (A)

Figure [4.13] Final Project (B)

Figure [4.14]: Final Project (C)

Figure [4.15]: Final Project (D)

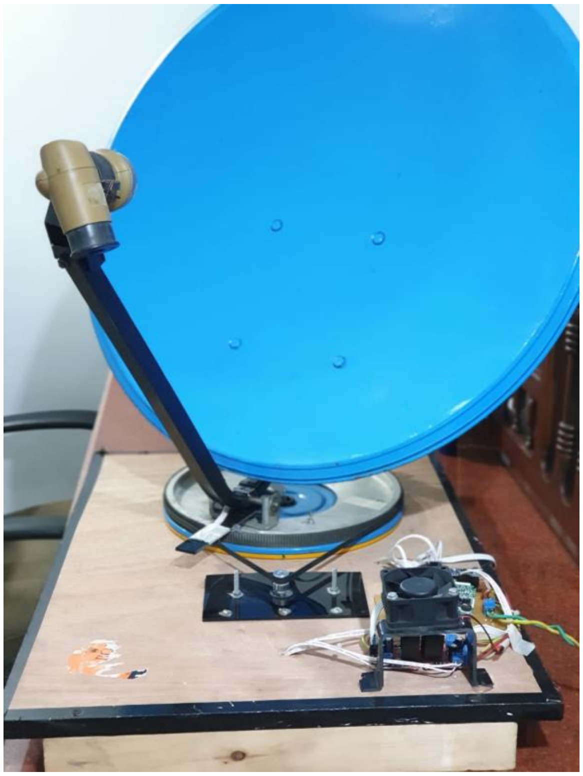


Figure [4.16]: Final Project (E)

CHAPTER:5

CONCLUSION

5 CONCLUSION

4.5 Overview

We have designed a two-axis control system for a receiving antenna (BSS) over a moving platform to provide dish channels on recreation vehicles (RVs). This is a practical, inexpensive and readily adaptable system for dish antenna installation on recreational vehicles so that the vacationers and travelers can enjoy satellite-transmitted television programs on their moving vehicles.

4.6 Objectives/ achievements

The objective of providing dish channels on a recreational vehicle is achieved.The GPS module u-blox NEO-6M connected to one ATmega328P is used to get the desired position of the dish. From the desired position, angle of elevation and azimuth are calculated. These are our desired angles to get perfect signals from satellite for our TV station. HMC 5883 magnetometer connected to one ATmega328P is used to get the current azimuthal angle while MPU6050 connected to the other ATmega328P is used to get the current elevation angle.The ATmega328p will compare these angles with current angles of the dish antenna, taken using HMC5883 and MPU6050 placed along the dish antenna. If the angles are not aligned, then the controller will adjust current position of our dish antenna by moving it through stepper motors connected. Once the respective angles provided by user are in accordance with dish angles, we will receive proper output from our dish receiver. In this way, our dish antenna will automatically adjust to provide dish channels on our moving vehicles.

CHAPTER: 6

FUTURE WORK

5 FUTURE WORK

5.1 Overview

The two-axis control system for a receiving antenna (BSS) over a moving platform provides arrangements for changing or varying the orientation or the shape of the directional pattern of the waves radiated from an antenna using mechanical movement of antenna for varying two co-ordinates of the orientation (elevation and azimuth angles). As a result, our dish antenna automatically adjusts to provide dish channels on our moving vehicle.

5.2 Additional Goals

We have designed a “two-axis” control system for a receiving antenna (BSS) over a moving platform. Additional goal of the project is to do some research and develop a system to control, the third axis i.e., “Skew”.

LNB skew is the rotational position of the **LNB** mounted on a satellite dish. It must be set within certain limits to reduce the number of errors received on both vertically and Horizontally Polarized Transponders. ... The actual “**skew** angle” is dependent on the satellite your dish is facing and on your location.

When you set the **skew** of your Feed/LNB assembly, you are lining up your **dish** exactly to the same horizontal/vertical position as the **satellite** in the **sky**. The satellites all lie in a curve in the sky called the Clarke Belt and so are at different heights. Ku band satellites use horizontal and vertical linear polarization and so the skew has to match the (H/V) position of the satellite for optimum satellite signal strength and quality. So controlling the skew axis, we will be able to get optimum satellite signal strength and quality.

7 BIBLIOGRAPHY

[1]Weerackody, V., and Cuevas, E., “Technical Challenges and Performance of Satellite Communications On-The-Move Systems,” Johns Hopkins APL Tech. Dig. 30(2), 113–121 (2011).

[2] DeBruin, J., “Control Systems for Mobile SATCOM Antennas,” IEEE Contr. Syst. Mag. (1), 86–101 (2008).

[3] International Telecommunication Union, “Statistical Methodology to Assess Time-Varying Interference Produced by a Geostationary Fixed-Satellite Service Network of Earth Stations Operating with MF-TDMA Schemes to Geostationary Fixed-Satellite Service Networks,” ITU-R Recommendation S.2029 (Dec 2012).

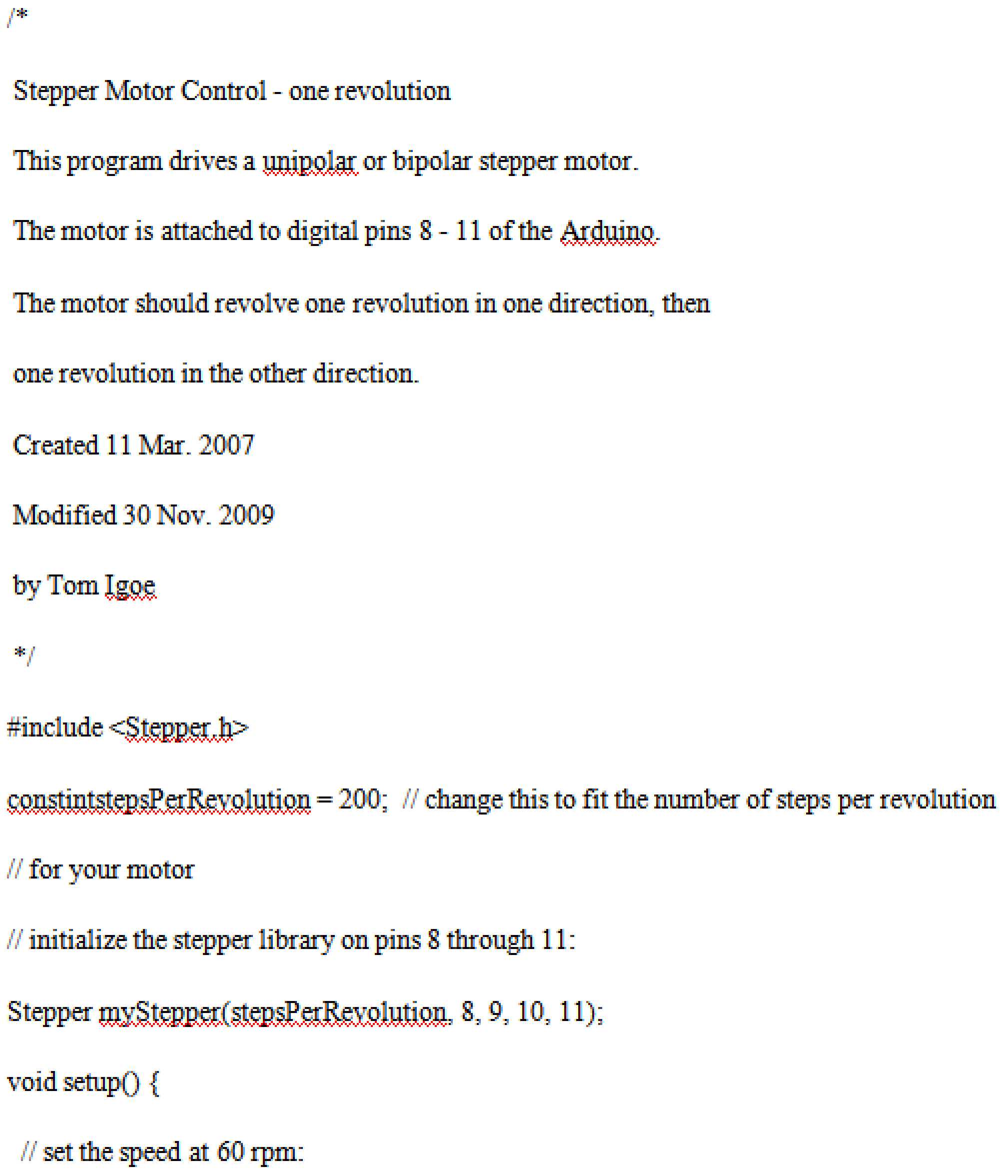
[4]https://patents.google.com/patent/US4126865A/en?q=satellite&q=satellite+dish&q=dish+antenna&q=position&q=antenna&before=priority:19921118&

[5]https://patents.google.com/patent/US4801940A/en?q=satellite&q=satellite+dish&q=dish+antenna&q=position&q=antenna&before=priority:19921118&scholar.

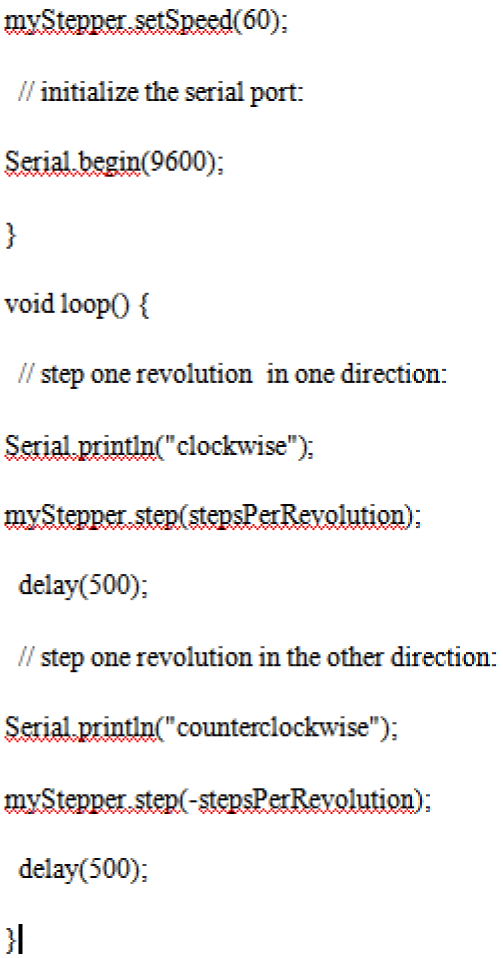
8 APPENDIX A

8.1 Stepper Motor Code

Following is the code for stepper motors:



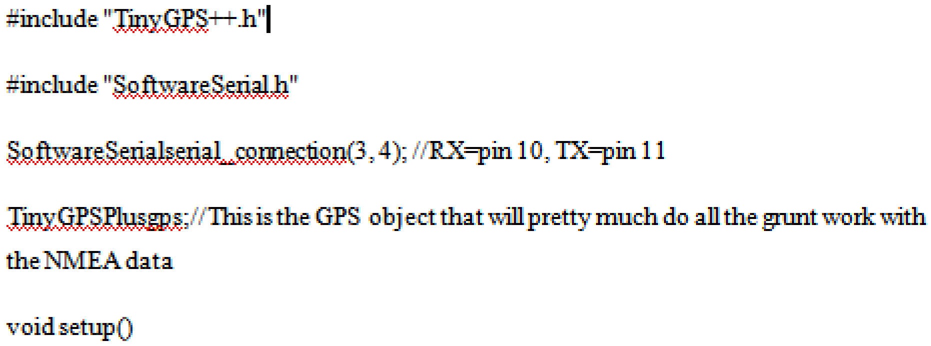
```
 /*

  Stepper Motor Control - one revolution

  This program drives a unipolar or bipolar stepper motor.

  The motor is attached to digital pins 8 - 11 of the Arduino.

  The motor should revolve one revolution in one direction, then

  one revolution in the other direction.

  Created 11 Mar. 2007

  Modified 30 Nov. 2009

  by Tom Igoe

  */

 #include <Stepper.h>

 constintstepsPerRevolution = 200;  // change this to fit the number of steps per revolution

 // for your motor

 // initialize the stepper library on pins 8 through 11:

 Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

 void setup() {

   // set the speed at 60 rpm:
```



```
myStepper.setSpeed(60);

  // initialize the serial port:

Serial.begin(9600);

}

void loop() {

  // step one revolution  in one direction:

Serial.println("clockwise");

myStepper.step(stepsPerRevolution);

  delay(500);

  // step one revolution in the other direction:

Serial.println("counterclockwise");

myStepper.step(-stepsPerRevolution);

  delay(500);

}
```

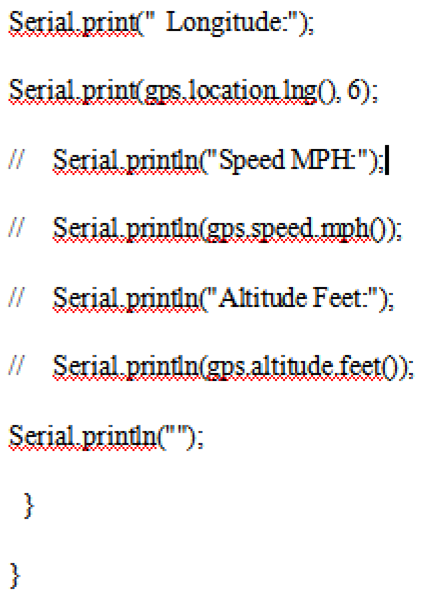
8.2 U-blox NEO-6M GPS Module Code

Following is the code for u-blox NEO-6M GPS module:



```
#include "TinyGPS++.h"

#include "SoftwareSerial.h"

SoftwareSerialserial_connection(3, 4); //RX=pin 10, TX=pin 11

TinyGPSPlusgps;//This is the GPS object that will pretty much do all the grunt work with
the NMEA data

void setup()
```

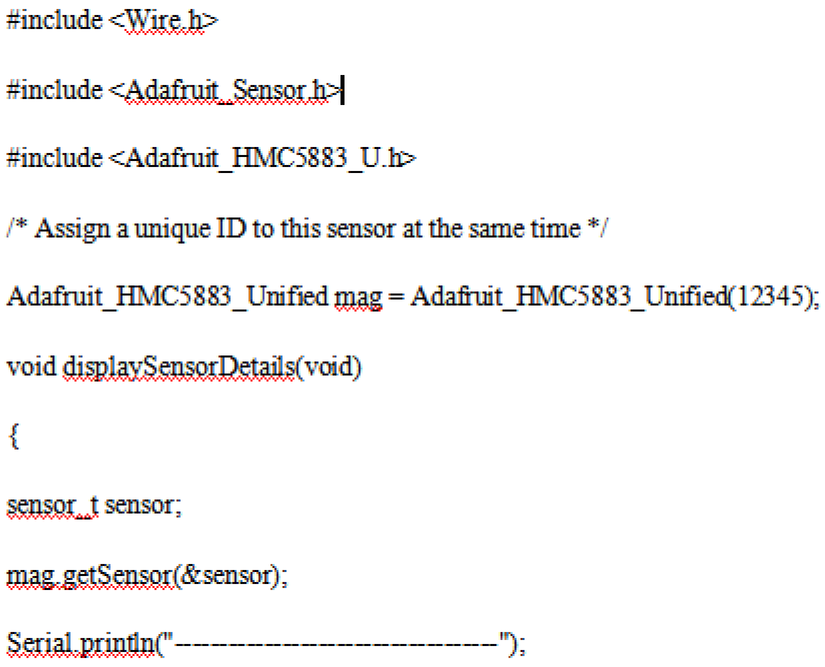
```
{
Serial.begin(9600);//This opens up communications to the Serial monitor in the Arduino IDE
serial_connection.begin(9600);//This opens up communications to the GPS
Serial.println("GPS Start");//Just show to the monitor that the sketch has started
}
void loop()
{
  while(serial_connection.available())//While there are characters to come from the GPS
  {
gps.encode(serial_connection.read());//This feeds the serial NMEA data into the library one char at a time
  }


  if(gps.location.isUpdated())//This will pretty much be fired all the time anyway but will at least reduce it to only after a package of NMEA data comes in
  {
    //Get the latest info from the gps object which it derived from the data sent by the GPS unit
//    Serial.println("Satellite Count:");
//    Serial.println(gps.satellites.value());
Serial.print("Latitude:");
Serial.print(gps.location.lat(), 6);
```



```
Serial.print(" Longitude:");
Serial.print(gps.location.lng(), 6);
//    Serial.println("Speed MPH:");
//    Serial.println(gps.speed.mph());
//    Serial.println("Altitude Feet:");
//    Serial.println(gps.altitude.feet());
Serial.println("");
  }
}
```

8.3 HMC-5883 Magnetometer Code

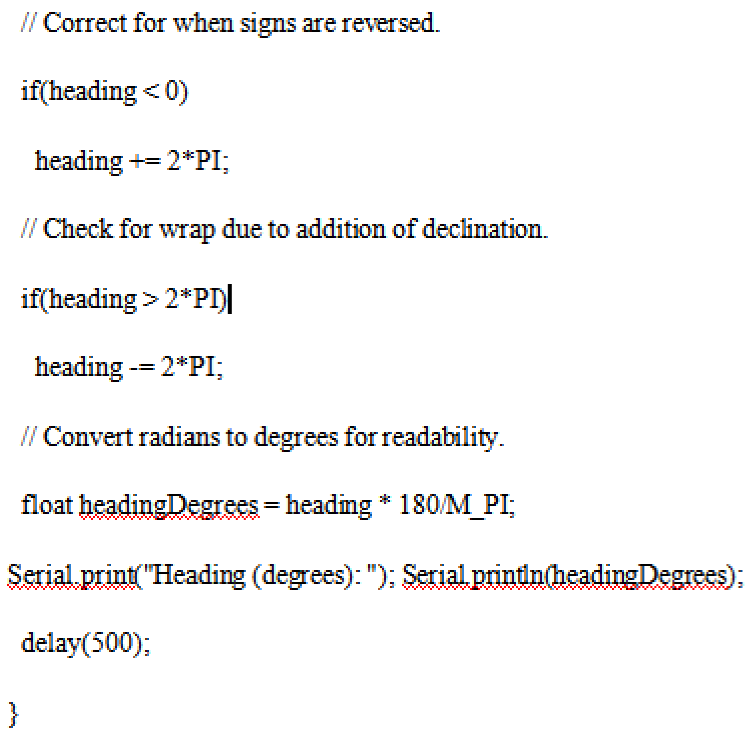
Following is the code for HMC-5883 magnetometer:



```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_HMC5883_U.h>
/* Assign a unique ID to this sensor at the same time */
Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345);
void displaySensorDetails(void)
{
sensor_t sensor;
mag.getSensor(&sensor);
Serial.println("--------------------");
```

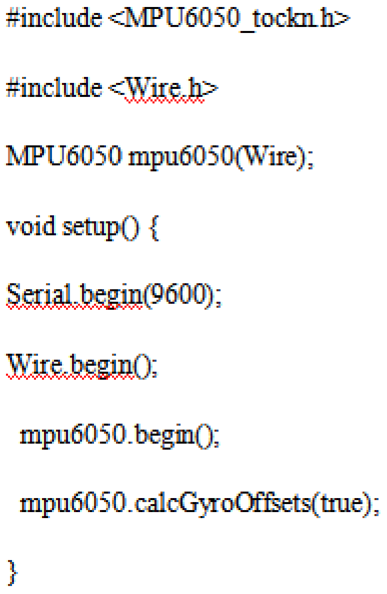
```
 Serial.print ("Sensor:       "); Serial.println(sensor.name);

 Serial.print ("Driver Ver:   "); Serial.println(sensor.version);

 Serial.print ("Unique ID:    "); Serial.println(sensor.sensor_id);

 Serial.print ("Max Value:    "); Serial.print(sensor.max_value); Serial.println(" uT");

 Serial.print ("Min Value:    "); Serial.print(sensor.min_value); Serial.println(" uT");

 Serial.print ("Resolution:   "); Serial.print(sensor.resolution); Serial.println(" uT");

 Serial.println("------------------------------------");

 Serial.println("");

  delay(500);

 }

void setup(void)

{

Serial.begin(9600);

Serial.println("HMC5883 Magnetometer Test"); Serial.println("")

  /* Initialise the sensor */

  if(!mag.begin())

  {

   /* There was a problem detecting the HMC5883 ... check your connections */

Serial.println("Ooops, no HMC5883 detected ... Check your wiring!");

   while(1);

  }
```

```
  /* Display some basic information on this sensor */
displaySensorDetails();
}
void loop(void)
{
  /* Get a new sensor event */
sensors_event_t event;
mag.getEvent(&event);
  /* Display the results (magnetic vector values are in micro-Tesla (uT)) */
Serial.print("X: "); Serial.print(event.magnetic.x); Serial.print("  ");
Serial.print("Y: "); Serial.print(event.magnetic.y); Serial.print("  ");
Serial.print("Z: "); Serial.print(event.magnetic.z); Serial.print("  ");Serial.println("uT");
  // Hold the module so that Z is pointing 'up' and you can measure the heading with x&y
  // Calculate heading when the magnetometer is level, then correct for signs of axis.
  float heading = atan2(event.magnetic.y, event.magnetic.x);

  // Once you have your heading, you must then add your 'Declination Angle', which is the 'Error' of the magnetic field in your location.
  // Find yours here: http://www.magnetic-declination.com/
  // Mine is: -13* 2' W, which is ~13 Degrees, or (which we need) 0.22 radians
  // If you cannot find your Declination, comment out these two lines, your compass will be slightly off.
  float declinationAngle = 0.22;
  heading += declinationAngle;
```



```
  // Correct for when signs are reversed.
  if(heading < 0)
    heading += 2*PI;
  // Check for wrap due to addition of declination.
  if(heading > 2*PI)
    heading -= 2*PI;
  // Convert radians to degrees for readability.
  float headingDegrees = heading * 180/M_PI;
 Serial.print("Heading (degrees): "); Serial.println(headingDegrees);
  delay(500);
 }
```

8.4 MPU6050 Gyroscope Code



```
#include <MPU6050_tockn.h>
#include <Wire.h>
MPU6050 mpu6050(Wire);
void setup() {
Serial.begin(9600);
Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
}
```

```
void loop() {
  mpu6050.update();
Serial.print("angleX : ");
Serial.print(mpu6050.getAngleX());
Serial.print("\tangleY : ");
Serial.print(mpu6050.getAngleY());
Serial.print("\tangleZ : ");
Serial.println(mpu6050.getAngleZ());
}
```

8.5 Complete Code Dish-A

```
#include <Stepper.h>
#include <MPU6050_tockn.h>
#include <Wire.h>
constintstepsPerRevolution = 200;
Stepper elevation(stepsPerRevolution, 9, 10, 11, 12);
MPU6050 mpu6050(Wire);
intx,y,z; //triple axis data
void setup() {
  // set the speed at 60 rpm:
Serial.begin(9600);
elevation.setSpeed(100);
```

```
 // initialize the serial port:
Wire.begin();
 mpu6050.begin();
 mpu6050.calcGyroOffsets(true);
}
void loop() {
 mpu6050.update();
 y = mpu6050.getAngleY();
 while( !(y < -3 && y > -5) )
 {
   mpu6050.update();

   y = mpu6050.getAngleY();
   if( y > -3)
   {movedown();}
   if( y < -5)
   {moveup();}
 }
}
void moveup()
{
elevation.step(200);
```

```
}

void movedown()

{

elevation.step(-200);

}
```

8.6 Complete Code Dish-B

```
#include <Stepper.h>

#include <Wire.h>

#define I2C_TX write

#define I2C_RX read

#define HMC5883_WriteAddress 0x1E //  i.e 0x3C >> 1

#define HMC5883_ModeRegisterAddress 0x02

#define HMC5883_ContinuousModeCommand (uint8_t)0x00     // cast to uint8_t added to get code to compile under Arduino v1.0

#define HMC5883_DataOutputXMSBAddress 0x03

intregb = 0x01;

intregbdata = 0x40;


intoutputData[6];

float azimuth;

constintstepsPerRevolution = 200;

Stepper azimuthal(stepsPerRevolution, 8, 7, 6, 5);
```

```
intx,y,z; //triple axis data
void setup() {
  // set the speed at 60 rpm:
 Serial.begin(9600);
 azimuthal.setSpeed(100);
  // initialize the serial port:
 Wire.begin();  //Initiate the Wire library and join the I2C bus as a master
 }
 void loop() {
 azimuthAngle();
void loop() {
azimuthAngle();
Serial.println("Signals received   Frequency:1150MHz TURKSAT ");
  while( !(azimuth < 85 && azimuth > 83) )
  {
Serial.println("Searching ");
azimuthAngle();
   if(azimuth > 84)
    {moveright();}
     if(azimuth < 83)
    {moveleft();}
```

```
   }

  }

  void moveleft()

  {

  azimuthal.step(1);

  }

  void moveright()

  {

  azimuthal.step(-1);

  }

  float azimuthAngle()

  {

  intx,y,z;

    double angle;

Wire.beginTransmission(HMC5883_WriteAddress); //Initiate a transmission with
HMC5883 (Write address).

  Wire.I2C_TX(HMC5883_ModeRegisterAddress);     //Place the Mode Register Address
in send-buffer.

  Wire.I2C_TX(HMC5883_ContinuousModeCommand);   //Place the command for
Continuous operation Mode in send-buffer.

Wire.endTransmission();                    //Send the send-buffer to HMC5883 and end the
I2C transmission.

Wire.beginTransmission(HMC5883_WriteAddress);  //Initiate a transmission with
HMC5883 (Write address).
```

```
Wire.requestFrom(HMC5883_WriteAddress,6);     //Request 6 bytes of data from the address specified.
 if(Wire.available() <= 6) // If the number of bytes available for reading is <=6
 {
  for(inti=0;i<6;i++)
  {
outputData[i]=Wire.I2C_RX();  //Store the data in outputData buffer
  }
 }
 x=outputData[0] << 8 | outputData[1];  //Combine MSB and LSB of X Data output register
 z=outputData[2] << 8 | outputData[3];  //Combine MSB and LSB of Z Data output register

 y=outputData[4] << 8 | outputData[5];  //Combine MSB and LSB of Y Data output register
 angle = (double)atan2(y, x); // angle in radians
 float declinationAngle = -0.019;
 angle += declinationAngle;
 // Correct for when signs are reversed.
 if (angle < 0)    angle += 2*PI;
 // Check for wrap due to addition of declination.
 if (angle > 2*PI) angle -= 2*PI;
                                                                              //
```

```
Convert radians to degrees for readability.

  float bearing = angle * 180/PI;

  azimuth = bearing;

//  Serial.println();

//  Serial.println("Heading (degrees): " + String(bearing));

//  Serial.print("\nYou are heading ");

//  if((bearing > 337.5) || (bearing < 22.5))    Serial.print("North");

//  if((bearing > 22.5)  && (bearing < 67.5 ))   Serial.print("North-East");

//  if((bearing > 67.5)  && (bearing < 112.5 ))  Serial.print("East");

//  if((bearing > 112.5) && (bearing < 157.5 ))  Serial.print("South-East");

//  if((bearing > 157.5) && (bearing < 202.5 ))  Serial.print("South");

//  if((bearing > 202.5) && (bearing < 247.5 ))  Serial.print("South-West");

//  if((bearing > 247.5) && (bearing < 292.5 ))  Serial.print("West");

//  if((bearing > 292.5) && (bearing < 337.5 ))  Serial.print("North-West");

//  delay(100);

}
```