1. INTRODUCTION

The project is designed to develop a dish positioning system which can be operated which can be operated by using a conventional remote . an electronic dish position control system is made up of two motors that enable the dish to rotate both in horizontal and vertical direction ,that enable the dish to rotate both in horizontal and vertical direction

The invention of the television was the work of many individuals in the late 19th century and early 10th century. Direct to home (DTH) televisions is a method of receiving satellite televisions is a method receiving satellite television by means of signals transmitted from the direct broadcast satellites. In order to receive these signals, it has to be in the direct line of the antenna.

For getting the best signal, following should be done; the line of sight view to the particular satellite is free of obstacles and obstructions. Adjust the antenna reflector or azimuth angle obtained for the help in controlling all types of motors with precise, adaptable intelligence.

Featuring the latest in electronics technology from the best and brightest business, it gives everything from the fundamentals to cutting edge design tips, including real life examples with software code. More more critical in motor design, sophisticated electronic controls, provide greater efficiency,finer speed.

1.2. OBJECTIVE

The system can be implemented by using computer instead of arduino. A satellite locating system is established in the present system. The capability of tracking in a dish positioning system is a very crucial point. Controlling of dish manually is very much time consuming and there can be inaccuracy in the manual process.

The system comprise of a remote for starting the motor and then moving it in required direction. The use of a remote control for monitoring the system makes the advanced and employment of arduino uno moves the motor in required direction. Through for the robotics, control system, automation system this is the initial step to search and is able to assist controls in industries.

CHAPTER 2

EMBEDDED SYSTEM

An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine. We use washing machines almost daily but wouldn't get the idea that it is an embedded system consisting of a Processor (and other hardware as well) and software. Embedded Systems can not only be stand-alone devices like Washing Machines but also be a part of a much larger system.

An example for this is a Car. A modern-day Car has several individual embedded systems that perform their specific tasks with the aim of making a smooth and safe. Some of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor etc.

2.1 PROGRAMMING FOR EMEDDED SYSTEM

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gate Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works. Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA(high level programming languages), Visual Basic, JAVA Script etc.,

2.1 EMBEDDED SYSTEM AND ITS REAL TIME APPLICATION

The World is filled with Embedded Systems. The development of Microcontroller has proved path for several Embedded System application and they play a significant role (and will continue to play in the future as well) in ou rmodern-day life in one way or the other. Starting from consumer electronics like Digital Cameras, DVD Players to high end and advanced systems like Flight Controllers and Missile Systems, embedded systems are omnipresent and became an important part of our life.

The way we live our life has been significantly improved with the utilization of Embedded Systems and they will continue to be an integral part of our lives even tomorrow. Another important concept we are hearing these days is Real — Time Systems. In a real time system, Real Time Computing takes place, where a computer (an Embedded System) must generate response to events within certain time limits.

2.2. REAL TIME APPLICATION OF EMBEDDED SYSTEM

- Latest Smart TV's
- GPS Navigation System
- Almost all modern-day smart phone
- Missile Guidance System
- Space exploration (rovers)
- Automobiles (ABS, Airbags)
- Industries(Assembly Robots)
- Road safety system (traffic monitoring and collision alert systems) and many other

2.3. ADVANTAGE

- Easy to manage embedded system meant for general use are easy to manage
- Fast performance
- They are smaller in size
- Hardware benefits and cost-effectiveness
- It can be used in mobile robots and military application

CHAPTER 3

3.1. HARDWARE DESCRPTION

3.1.1 ARDUINO UNO

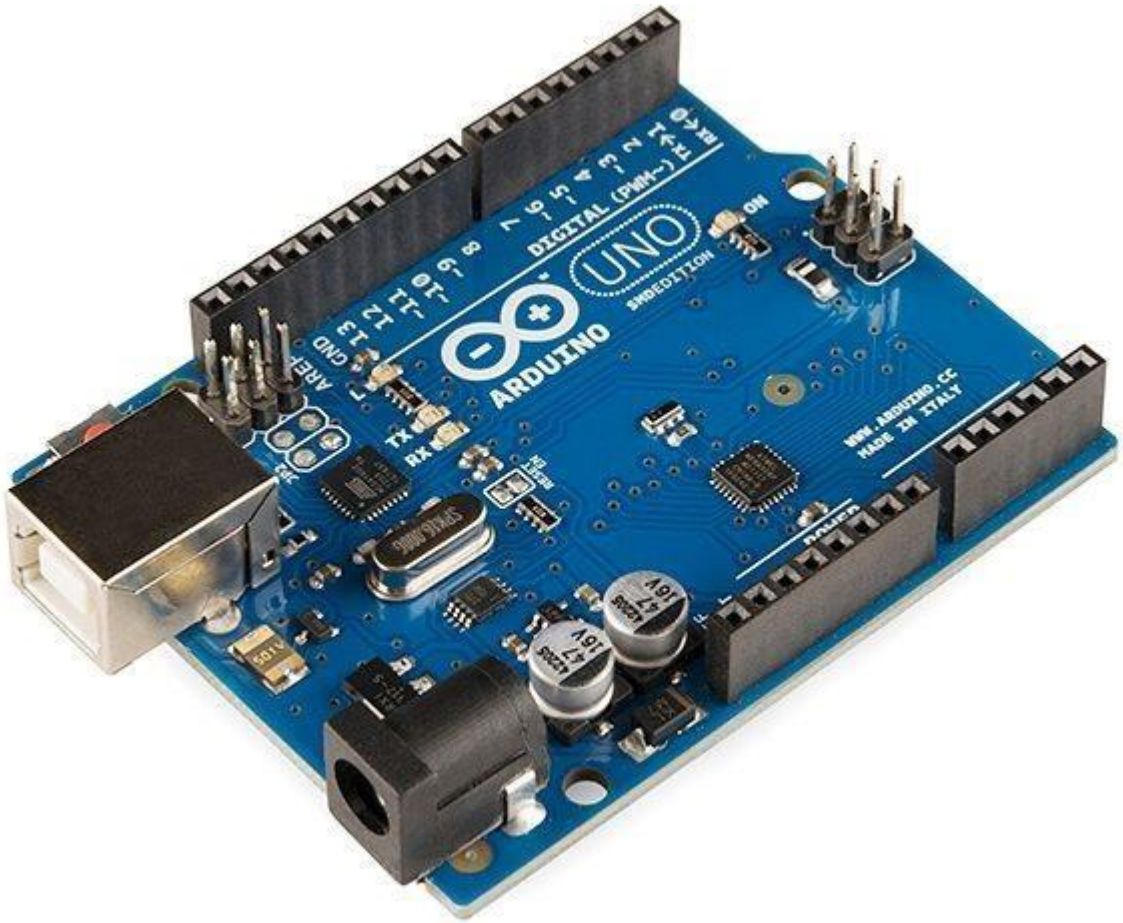


FIGURE 3.1.1

The Arduino Uno is the most popular one among other Arduino development boards. It is based on the ardunio . The factor which make Arduino Uno differ from others is that it does not use FTDI USB-to-serial driver chip. Instead it uses a microcontroller Atmega16U2 (Atmega8U2 up to version R2)

which is programmed as USB-to-serial converter. Its peripheral features includes 14 digital input/output pins (of which 6 can provide PWM outputs), 6 analog inputs, a 16MHz crystal oscillator, a USB connector, a power jack, an ICSP header and a reset button.

We can power Arduino Uno in 2 different ways, either by USB connection or by using an external power supply. AC-DC adaptor and battery will come under the category of external power supply. To use battery, connect its leads to the **Vin** and **GND** pins on the board. To use adaptor, connect a 2.1mm center positive plug in to the board's power jack. The board can operate in the range of 6-20V of the external power source.

But the most suitable range is 7-12V. If the input voltage is less than 7V, the on-board regulator can't produce proper regulated 5V and the board will become unstable. If the input voltage is more than 12V it will cause the voltage regulator to be overheated and may result in damaging the board.

3.1.2.IR RECEIVER

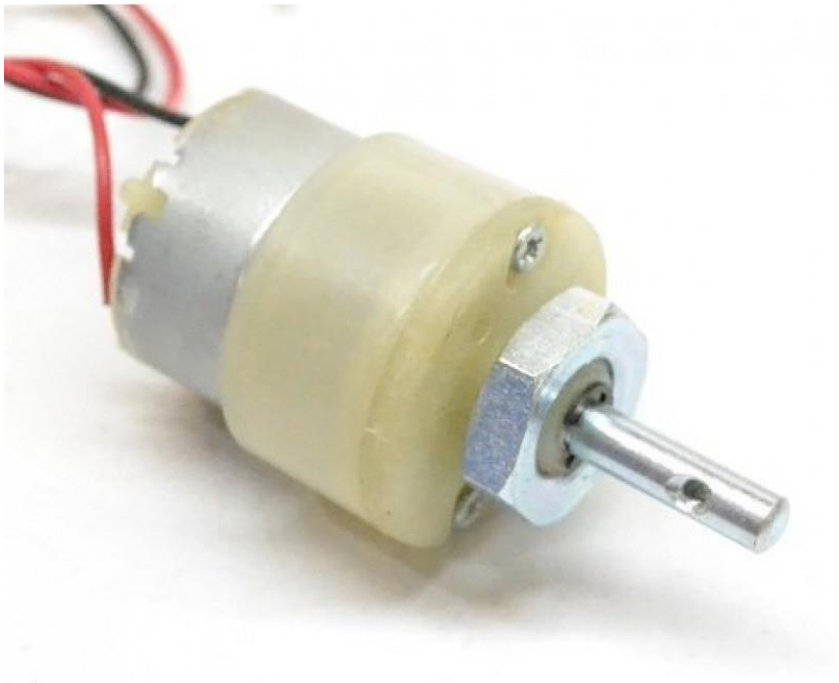
FIGURE 3.1.2

The connections for the IR sensor with the Arduino are as follows: Connect the negative wire on the IR sensor to GND on the Arduino. Connect the middle of the IR sensor which is the VCC to 5V on the Arduino. Connect the signal pin on the IR sensor to pin 8 on the Arduino.

IR Recevier is one of the most common method of wireless communication due to being easy to use and having an affordable price. infrared light, with a wavelength longer than visible light, is not within the range of human vision. That’s good option for wireless communication

.3.1.3.DC MOTORS

DC Motor – 30RPM – 12Volts geared motors are generally a simple DC motor with a gearbox attached to it. This can be used in all-terrain robots and variety of robotic applications. These motors have a 3 mm threaded drill hole in the middle of the shaft thus making it simple to connect it to the wheels or any other mechanical



30 RPM 12V DC geared motors widely use for robotics applications. Very easy to use and available in standard size. Also, you don’t have to spend a lot of money to control motors with an Arduino or compatible board.

The most popular L298N H-bridge module with onboard voltage regulator motor driver can be used with this motor that has a voltage of between 5 and 35V DC or you can choose the most precise motor diver module from the wide range available in our Motor divers Nut and threads on the shaft to easily connect and internally threaded shaft for easily connecting it to the wheel. DC Geared motors with robust metal gearbox for heavy-duty applications, available in the wide RPM range and ideally suited for robotics and industrial applications. Very easy to use and available in standard size. Nut and threads on the shaft to easily connect and internally thread shaft for easily connecting it to the wheel.

3.1.4. SERVO MOTOR

Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is typically black or brown and should be connected to a ground pin on the board. The signal pin is typically yellow or orange and should be connected to PWM pin on the board. In these examples, it is pin number 9.

It uses an encoder or speed sensor to provide speed feedback and position. This feedback signal is compared with the input command position (desired position of the motor corresponding to a load 0 produces the signal

FIGURE 3.1.4

3.1.5. REMOTE

In this project we will learn how to decode IR remote signals with Arduino and to control DC motors depending on the button pressed. For demonstrating the working we are using five buttons on the remote. When the next button on the remote is pressed, motors will rotate in clockwise direction. And if the previous button is pressed motors will rotate in anticlockwise direction. We can also control these two motors individually using left, right arrow buttons and stop button can be used to stop the rotation.

FIGURE 3.1.5

3.2. SOFTWARE DESCRIPTION

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. into the editor has features for cutting/pasting and for searching/replacing text.

The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages text output by the Arduino Software (IDE), including complete error messages text output by the Arduino Software (IDE), including complete error messages board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor

3.1.1 PROGRAMMING

```
#include <IRremote.h>
#include <Servo.h>
int IRpin = 11; // pin for the IR sensor
int led1 = 2;
int itsONled[] = {0, 0, 0, 0};
#define code1 33484815/ // code received from button no. 1
IRrecv irrecv(IRpin);
decode_results results;
Servo myservo;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
  pinMode(led1, OUTPUT);
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}
```

```
void loop()
{
  if  (irrecv.decode(&results))
  {
    irrecv.resume();    // Receive the next value
  }
  if (results.value == 33480735) // change according to your IR remote butt
on number
  {
    myservo.write(0);
    delay(15);
  }
  if (results.value == 33444015) // change according to your IR remote butt
on number
  {
    myservo.write(30);
    delay(15);
  }
  if (results.value == 33478695) // change according to your IR remote butt
on number
  {
    myservo.write(60);
    delay(15);
  }
  if (results.value == 33486855) // change according to your IR remote butt
on number
  {
```

```
  myservo.write(90);
  delay(15);
 }
 if (results. value == 33435855) // change according to your IR remote but
ton number
 {
  My servo. write (120);
  delay(15);
 }
 if (results.value == 33468495) // change according to your IR remote butt
on number
 {
  myservo.write(150);
  delay(15);
 }
 if (results.value == 33452175) // change according to your IR remote butt
on number
 {
  myservo.write(180);
  delay(15);
 }
 if (irrecv.decode(&results)) {
  unsigned int value = results.value;
  switch (value) {
    case code1:
      if (itsONled[1] == 1) {  //  if  first  led  is  on then
        digitalWrite(led1, LOW); // turn it off when button is pressed
```

```
        itsONled[1] = 0; // and set its state as off
      } else { // else if first led is off
        digitalWrite(led1, HIGH);
        // turn it on when the button is pressed
        itsONled[1] = 1; // and set its state as on
      }
      break;
    }
  }
}
```

3.3 WORKING PRINCIPLE

Each sensor circuit is built around MOC7811 having an internal arduino. Both sensor circuits for elevation and azimuth angles are identical.

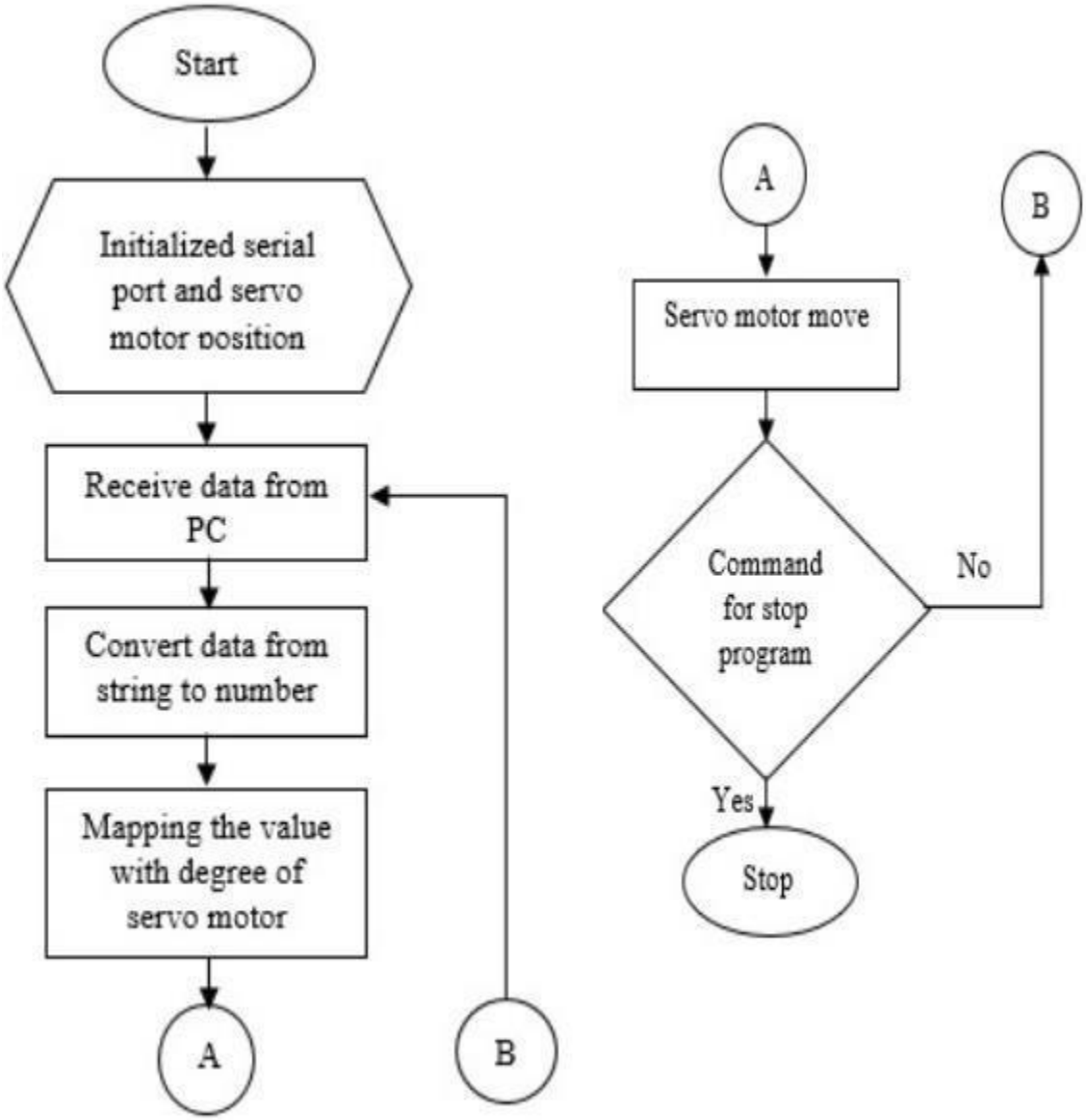
Internal IR LED is given 5V supply through 330-ohm current-limiting resistors (R6 and R8), and photo-transistor is connected to switch configuration using external 1-kilo-ohm resistors (R7 and R9). IR light beams continuously fall on the phototransistors. Sensor outputs are taken from collectors (pin 2) of the phototransistors. Both sensor outputs (from IC1 and IC2) are connected to digital I/O pins 2

and 3 of Arduino Uno, respectively. Pins A0 through A4 are connected to 5V through resistors R1 through R5, respectively.

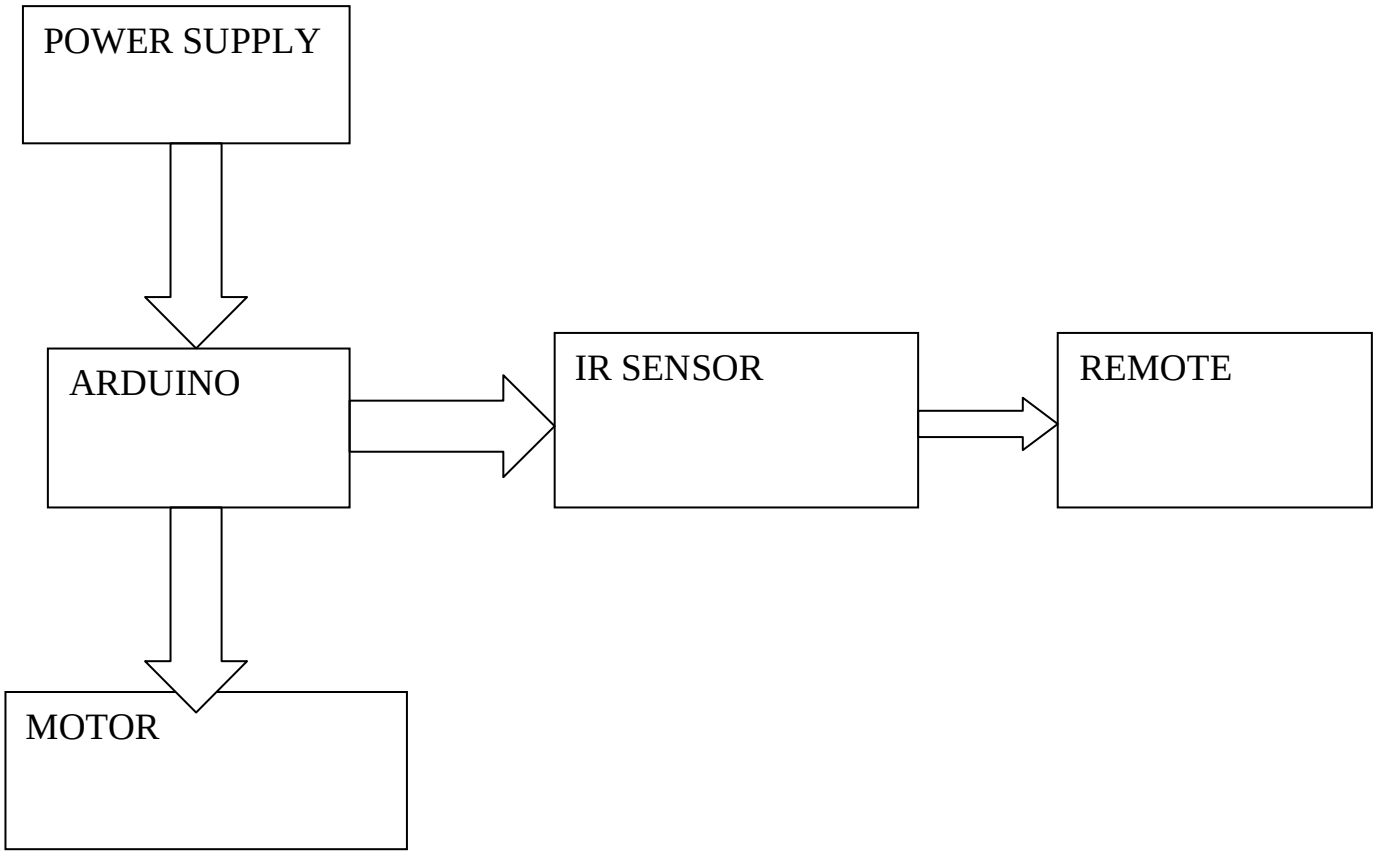
Switches S1 through S5 are connected such that when a switch is pressed, logic 0 is given as input to Arduino Uno. Digital pins 8 and 9 of Arduino Uno are connected to control pins RS and EN, respectively. data pins DB4 through DB7 are connected to digital pins 10, 11, 12 and 13 of Arduino Uno. R/W pin of is connected to ground to make it write enable. Digital pins 4, 5, 6 and 7 of Arduino Uno are connected to input pins of L293D (IC3). The motors are connected to output pins of L293D.

Arduino Uno drives the two DC motors (M1 and M2) The whole circuit works on 5V DC supply. Arduino Uno can be given 5V regulated supply on its 5V pin. A separate 12V supply is given to pin 8 of the motor supply. Actual controlling of the dish antenna angle controller is done using software (dish_contr.ino) written in Arduino programming language and compiled using Arduino IDE software After setting the angles, press S5. Motor 1 will rotate, followed by motor 2. LCD1 will display the values of azimuth and elevation angles. When the dish antenna attains the desired position, will show the set dish position.

3.4 .FLOW CHART


Start
Initialized serial port and servo motor position
Receive data from PC
Convert data from string to number
Mapping the value with degree of servo motor
A
B
A
Servo motor move
Command for stop program
No
Yes
Stop
B

3.5. BLOCK DIAGRAM

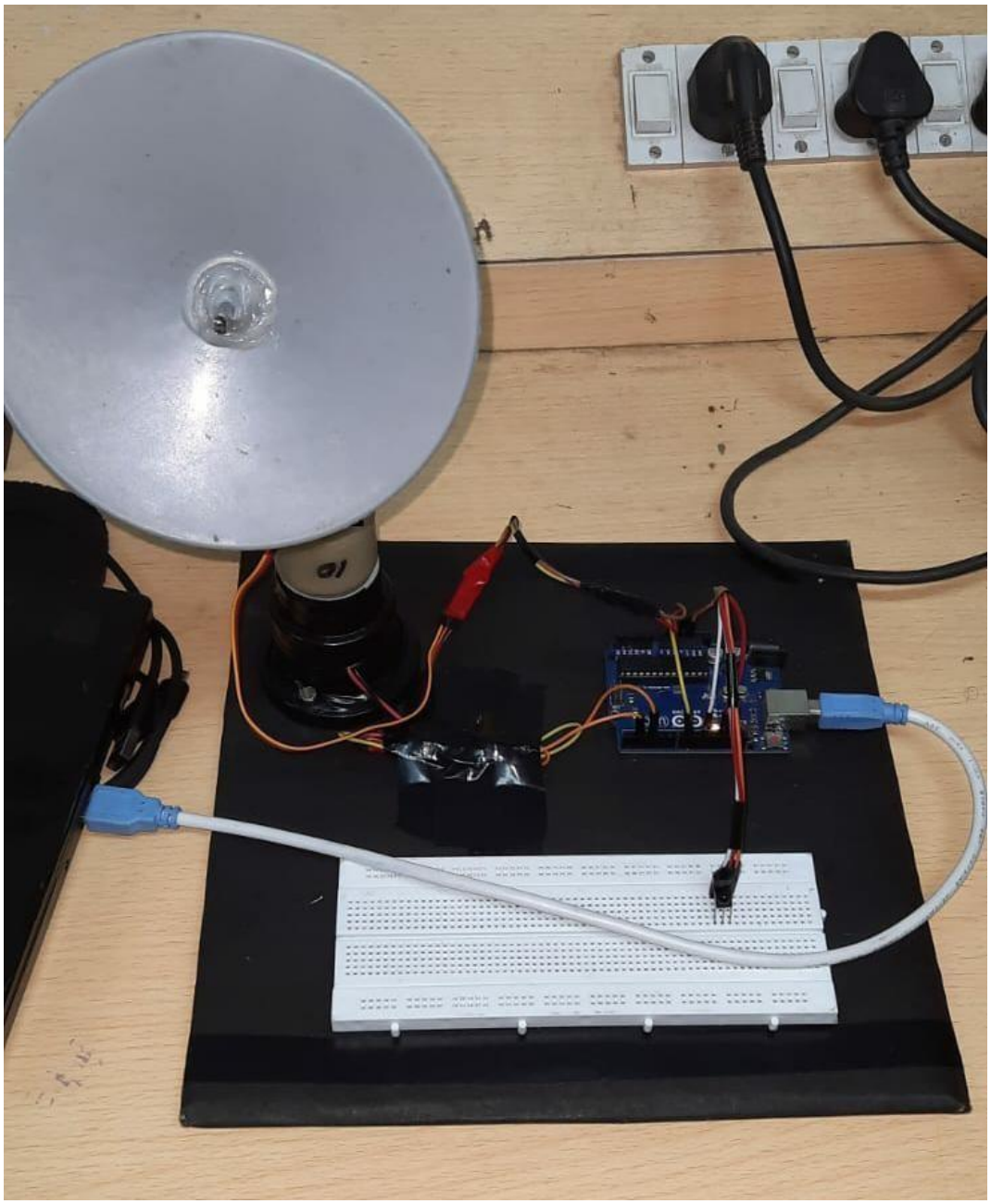


3.6.CONCLUSION

The proposed model of remote controlled dish positioning system can adjust the position of dish remotely by using IR remote . it will reduce the difficulty in adjusting the position of dish manually. If the dish position is adjusted manually, one cannot direct the dish towards the best possible position. This proposed system is intended to adjust the dish position through a simple remote by pressing the button corresponding to the direction of the advantages of the proposed system are less complexity, easy to implement and low cost.

Also, an automated system helps to reduce the human intervention and saves time . This system will help of dish TV by taking the maximum signal strength.

REAL IMAGE



FUTURE SCOPE

The future scope of the system is that positioning of dish can be implemented using RF will improve the efficiency of the system . it will increase the range between the remote and the IR receiver .i.e, using RF transmission the dish position can be controlled from a distance of 200 meters . also ,it can improve by interfacing sensors and control processors. Also to make it very simple it can be used as just a monitoring system and can provide SMS.

REFERENCES

[1] https://nevonprojects.com/ir-based-dish-position-controller

[2] http://repo.uofg.edu.sd/handle/123456789/3163

[3] International Journal of Recent Technology and Engineering (IJRTE) published 2012

[4] https://www.electronicsforu.com/electronics-projects/satellite-dish-antenna-angle-controller

[5] https://www.electronicsforu.com/electronics-projects/satellite-dish-antenna-angle-controll