

REFERENCES :

- [1] Carlin, T., Direct broadcast satellites. Communication technology update, 2002.
- [2] Roddy, D., Satellite communications. 2006.
- [3] Inglis, A.F. and A.C. Luther, Satellite technology: an introduction. 1997: Focal Press.
- [4] Maral, G. and M. Bousquet, Satellite communications systems: systems, techniques and technology. 2011: John Wiley & Sons.
- [5] Pelton, J.N., S. Madry, and S. Camacho-Lara, Handbook of satellite applications. 2012: Springer Publishing Company, Incorporated.
- [6] Kenjo, T., Stepping motors and their microprocessor controls. 1984: Clarendon Press.
- [7] Niku, S.B., Introduction to robotics: analysis, systems, applications. Vol. 7. 2001: Prentice Hall New Jersey.
- [8] Jones, D.W., Control of Stepping Motors: A Tutorial. 1998: Douglas W. Jones.
- [9] Valentine, R., Motor control electronics handbook. 1998: McGraw-Hill Professional.
- [10] Stephen, H., Industrial Motor Control. 2010.
- [11] Parkinson, B.W., Progress in Astronautics and Aeronautics: Global Positioning System: Theory and Applications. Vol. 2. 1996: Aiaa.
- [12] Chuong, B. and C. Barry, Satellite dish positioning system. 2000, Google Patents.

- [13] Oo, M.M.K., C.M. Nwe, and H.M. Tun, Satellite Dish Positioning Control by DC Motor Using IR Remote Control .2001.
- [14] Rafael, M.C., J.B. Gonçalves, and P.P.L. do Prado. DEVELOPMENT OF AN AUTOMATED SYSTEM FOR MANEUVERING PARABOLIC DISH ANTENNAS USED IN SATELLITE COMMUNICATION. in ABCM Symposium Series in Mechatronics e Section II e Control Systems. 2012.
- [15] Bhuyan, M.H., WIRELESS CONTROL SYSTEM FOR DC MOTOR TO POSITION A Dish ANTENNA USING MICROCOMPUTER. DAFFODIL INTERNATIONAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY, 2007.
- [16] Alhasan, E. and A. Alzubaidi, A design of software driver for a satellite dish antenna positioning system. IOSRJEN, 2015.
- [17] Alhassan, M. and E.A. Gabbar, Relationship between Stepper Motor Step-angle and Satellite Dish Antenna Diameter and Beam-Width. IOSR Journal of Engineering, 2014.
- [18] Singh, G., Bluetooth Wireless Technology 2002.

APPENDIX A

```
    /*main activity code*/

import android.Manifest;

import android.app.ProgressDialog;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Criteria;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.AsyncTask;

import android.support.v4.app.ActivityCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;
```

```
import android.widget.AdapterView;

import android.widget.Button;

import android.widget.Spinner;

import android.widget.Toast;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.UUID;

public class MainActivity extends AppCompatActivity implements
LocationListener {

    LocationManager locationManager;

    private UUID DEFAULT_UUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");

    String provider;

    String address = null;

    BluetoothAdapter bluetoothAdapter = null;

    BluetoothSocket bluetoothSocket = null;

    Boolean isBtConnected = false;
```

```
ProgressDialog progressDialog ;

Location location;

int currentSat;

int nextSat;

double firstVal;

double secondVal;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    Intent intent = getIntent();

    address = intent.getStringExtra(Main2Activity.EXTRA_ADDRESS);

    Spinner spinner = (Spinner) findViewById(R.id.spinner);

    Spinner spinner2 = (Spinner) findViewById(R.id.spinner2);

    Button button = (Button) findViewById(R.id.button2);

    new ConnectBT().execute();
```

```

button.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        calculations();

        if (bluetoothSocket != null) {

            try {

                bluetoothSocket.getOutputStream().write(String.valueOf(secondVal).getBytes());

                bluetoothSocket.getOutputStream().write(String.valueOf(firstVal).getBytes(
            ));

            } catch (IOException e) {

                Toast.makeText(MainActivity.this , "Error" ,
                Toast.LENGTH_LONG).show();
            }
        }
    }
});

```

```
    }  
    }  
  
    }  
});
```

```
List<String> satList = new ArrayList<>();  
  
satList.add("Default");  
  
satList.add("ArabSat");  
  
satList.add("HotBird");  
  
satList.add("NileSat");
```

```
ArrayAdapter<String> satAdapter = new ArrayAdapter<String>(this,  
android.R.layout.simple_spinner_dropdown_item, satList);
```

```
spinner.setAdapter(satAdapter);  
  
spinner2.setAdapter(satAdapter);
```

```
spinner.setOnItemClickListener(new  
AdapterView.OnItemClickListener() {
```

```
    @Override
```

```
public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
```

```
    switch (position) {
```

```
        case 0:
```

```
            currentSat = 0;
```

```
            break;
```

```
        case 1:
```

```
            currentSat = 26;
```

```
            break;
```

```
        case 2:
```

```
            currentSat = -13;
```

```
            break;
```

```
        case 3 :
```

```
            currentSat = -7;
```

```
            break;
```

```
    }
```

```
String chosenSat = parent.getItemAtPosition(position).toString();
```



```
        Toast.makeText(MainActivity.this, " " + chosenSat + " " +
currentSat, Toast.LENGTH_SHORT).show();
```

```
    }
```

```
    @Override
```

```
    public void onNothingSelected(AdapterView<?> parent) {
```

```
    }
```

```
});
```

```
        spinner2.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
```

```
            @Override
```

```
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
```

```
                switch (position) {
```

```
                    case 0:
```

```
                        currentSat = 0;
```

```

        break;

    case 1:

        currentSat = 26;

        break;

    case 2:

        currentSat = -13;

        break;

    case 3 :

        currentSat = -7;

        break;

    }

    String chosenSat = parent.getItemAtPosition(position).toString();

    Toast.makeText(MainActivity.this, " " + chosenSat + " " +
nextSat, Toast.LENGTH_SHORT).show();

}

```

@Override

```
public void onNothingSelected(AdapterView<?> parent) {
```

```
    }  
});
```

```
    locationManager = (LocationManager)  
getSystemService(Context.LOCATION_SERVICE);  
  
    provider = locationManager.getBestProvider(new Criteria(), false);  
  
    location = locationManager.getLastKnownLocation(provider);  
  
    onLocationChanged(location);  
  
}
```

```
@Override  
  
protected void onResume() {  
  
    super.onResume();  
  
    locationManager.requestLocationUpdates(provider, 1000, 1, this);  
  
}
```

```
@Override  
  
protected void onPause() {  
  
    super.onPause();  
  
    locationManager.removeUpdates(this);  
  
}
```

```
@Override  
  
public void onLocationChanged(Location location) {  
  
    if (location != null) {  
  
        Toast.makeText(this, "location is " + location.getLatitude(),  
Toast.LENGTH_SHORT).show();  
  
        Toast.makeText(this, "location is " + location.getLongitude(),  
Toast.LENGTH_SHORT).show();  
  
    }  
  
}
```

```
@Override  
  
public void onStatusChanged(String provider, int status, Bundle extras) {
```

```
}
```

```
@Override
```

```
public void onProviderEnabled(String provider) {
```

```
}
```

```
@Override
```

```
public void onProviderDisabled(String provider) {
```

```
}
```

```
public void calculations() {
```

```
    double LES ;
```

```
    double LE ;
```

```
    double LS = nextSat;
```

```
    int R = 6371;
```

```
    int a = 42164;
```

```
    double A;
```

```
double b;
```

```
double AZZ;
```

```
double B;
```

```
double d;
```

```
double EL;
```

```
if(currentSat == 0 ) {
```

```
    AZZ = 180;
```

```
    EL = 0;
```

```
    Toast.makeText(this ,"EL"+EL+" AZZ =" +AZZ ,  
    Toast.LENGTH_LONG).show();
```

```
}else {
```

```
    if (location != null) {
```

```
        LES = location.getLatitude();
```

```
        LE = location.getLongitude();
```

```
    } else {
```

```
LES = 15;
```

```
LE = 32;
```

```
}
```

```
B = LE - LS;
```

```
b = Math.toDegrees(Math.acos(Math.cos(Math.toRadians(B)) *  
Math.cos(Math.toRadians(LES))));
```

```
A =
```

```
Math.toDegrees(Math.asin(Math.sin(Math.abs(Math.toRadians(B))) /  
Math.sin(Math.toRadians(b))));
```

```
if (LES < 0 && B < 0) {
```

```
    AZZ = A;
```

```
} else if (LES < 0 && B > 0) {
```

```
    AZZ = 360 - A;
```

```
} else if (LES > 0 && B < 0) {
```

```

        AZZ = 180 - A;
    } else {

        AZZ = 180 + A;
    }

    d = Math.sqrt(R * R + a * a - (2 * R * a *
Math.cos(Math.toRadians(b))));

    EL = Math.toDegrees(Math.acos(Math.sin(Math.toRadians(b))));

    Toast.makeText(this , "B =" + B + " b =" + b + " A =" + A + " LES = " + LES
, Toast.LENGTH_LONG).show();

    Toast.makeText(this , "EL " + EL + " AZZ =" + AZZ + " d= " + d ,
Toast.LENGTH_LONG).show();

    }

    firstVal = (AZZ) / 0.35;

    secondVal = (EL) / 0.35;

}

private class ConnectBT extends AsyncTask<Void, Void, Void> {

```



```

private boolean connectSuccesses = false ;

@Override

protected void onPreExecute() {

    super.onPreExecute();

    progressDialog = progressDialog.show(MainActivity.this , "
Connecting .." , " Please Wait");

}

@Override

protected void doInBackground(Void... devices) {

    try {

        if (bluetoothSocket == null || !isBtConnected) {

            bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

            BluetoothDevice bluetoothDevice =
bluetoothAdapter.getRemoteDevice(address);

```

```

        bluetoothSocket =
bluetoothDevice.createInsecureRfcommSocketToServiceRecord(DEFAULT
_UUID);

        //BluetoothAdapter.getDefaultAdapter().cancelDiscovery();

        bluetoothSocket.connect();

        connectSukses = true;

    }

} catch (IOException e) {

    connectSukses = false ;

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

    super.onPostExecute(aVoid);

    if(!connectSukses) {

        Toast.makeText(MainActivity.this , "Connection Field try again" ,
Toast.LENGTH_LONG).show();

```

```
        finish();

    } else {

        Toast.makeText(MainActivity.this , "Connected" ,
Toast.LENGTH_LONG).show();

        isBtConnected = true;
    }

    progressDialog.dismiss();
}
}
}
```

```
/* Bluetooth enabling activity */

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import java.io.OutputStream;

import java.util.ArrayList;

import java.util.Set;

public class Main2Activity extends AppCompatActivity {
```

```
Button button ;
```

```
ListView deviceList ;
```

```
BluetoothAdapter bluetoothAdapter = null ;
```

```
Set<BluetoothDevice> pairedDevices ;
```

```
OutputStream outputStream = null;
```

```
static String EXTRA_ADDRESS = "device_address";
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main2);
```

```
    button = (Button) findViewById(R.id.button);
```

```
    deviceList = (ListView) findViewById(R.id.listView);
```

```
    bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
```

```
    if(bluetoothAdapter == null) {
```

```
        Toast.makeText(this , "Sorry You Cant use this app" ,  
        Toast.LENGTH_LONG).show();
```

```
    } else {
```

```
        if(blueetoothAdapter.isEnabled()) {
```

```
        } else {
```

```
            Intent turnBTon = new  
            Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
            startActivityForResult(turnBTon , 1);  
        }  
    }
```

```
    button.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
```

```
            pairedDeviceList();
```

```
    }  
});  
  
}  
  
private void pairedDeviceList() {  
  
    pairedDevices = bluetoothAdapter.getBondedDevices();  
  
    ArrayList list = new ArrayList();  
  
    if(pairedDevices.size() > 0) {  
  
        for(BluetoothDevice BTdevice : pairedDevices) {  
  
            list.add(BTdevice.getName()+"\n"+BTdevice.getAddress());  
  
        }  
    } else {
```

```
        Toast.makeText(this , "No Paired Devices Found" ,  
Toast.LENGTH_LONG).show();  
  
    }
```

```
        final ArrayAdapter arrayAdapter = new ArrayAdapter(this ,  
android.R.layout.simple_expandable_list_item_1 , list);  
  
        deviceList.setAdapter(arrayAdapter);  
  
        deviceList.setOnItemClickListener(myLsitClickListener);  
  
    }
```

```
        private AdapterView.OnItemClickListener myLsitClickListener = new  
AdapterView.OnItemClickListener() {
```

```
            @Override
```

```
            public void onItemClick(AdapterView<?> parent, View view, int  
position, long id) {
```

```
                String info = ((TextView ) view).getText().toString();
```

```
                String address = info.substring(info.length() - 17);
```



```
Intent intent = new Intent(Main2Activity.this , MainActivity.class);
```

```
intent.putExtra(EXTRA_ADDRESS , address);
```

```
startActivity(intent);
```

```
}
```

```
};
```

```
}
```

APPENDIX B

```
int inA1 = 4; // input 1 of the stepper
```

```
int inA2 = 5; // input 2 of the stepper
```

```
int inB1 = 6; // input 3 of the stepper
```

```
int inB2 = 7; // input 4 of the stepper
```

```
int inc1 = 8; // input 1 of the stepper
```

```
int inc2 = 9; // input 2 of the stepper
```

```
int ind1 = 10; // input 3 of the stepper
```

```
int ind2 = 11; // input 4 of the stepper
```

```
int stepDelay = 5; // Delay between steps in milliseconds
```

```
String readString, substr1, substr2;
```

```
void setup() {
```

```
    pinMode(inA1, OUTPUT);
```

```
    pinMode(inA2, OUTPUT);
```

```
    pinMode(inB1, OUTPUT);
```

```
    pinMode(inB2, OUTPUT);
```

```
    pinMode(inc1, OUTPUT);
```

```
    pinMode(inc2, OUTPUT);
```

```
pinMode(ind1, OUTPUT);  
pinMode(ind2, OUTPUT);  
Serial.begin(9600);  
  
}
```

```
void step11() {  
    digitalWrite(inA1, LOW);  
    digitalWrite(inA2, HIGH);  
    digitalWrite(inB1, HIGH);  
    digitalWrite(inB2, LOW);  
    delay(stepDelay);  
  
}
```

```
void step12() {  
    digitalWrite(inA1, LOW);  
    digitalWrite(inA2, HIGH);  
    digitalWrite(inB1, LOW);  
    digitalWrite(inB2, HIGH);  
    delay(stepDelay);  
  
}
```

```
}  
  
void step13() {  
    digitalWrite(inA1, HIGH);  
    digitalWrite(inA2, LOW);  
    digitalWrite(inB1, LOW);  
    digitalWrite(inB2, HIGH);  
    delay(stepDelay);  
}
```

```
void step14() {  
    digitalWrite(inA1, HIGH);  
    digitalWrite(inA2, LOW);  
    digitalWrite(inB1, HIGH);  
    digitalWrite(inB2, LOW);  
    delay(stepDelay);  
  
}
```

```
void step21() {  
    digitalWrite(inc1, LOW);  
    digitalWrite(inc2, HIGH);
```

```
digitalWrite(ind1, HIGH);  
  
digitalWrite(ind2, LOW);  
  
delay(stepDelay);  
  
}  
  
void step22() {  
  
    digitalWrite(inc1, LOW);  
  
    digitalWrite(inc2, HIGH);  
  
    digitalWrite(ind1, LOW);  
  
    digitalWrite(ind2, HIGH);  
  
    delay(stepDelay);  
  
}  
  
void step23() {  
  
    digitalWrite(inc1, HIGH);  
  
    digitalWrite(inc2, LOW);  
  
    digitalWrite(ind1, LOW);  
  
    digitalWrite(ind2, HIGH);  
  
    delay(stepDelay);  
  
}  
  
void step24() {
```

```
digitalWrite(inc1, HIGH);  
digitalWrite(inc2, LOW);  
digitalWrite(ind1, HIGH);  
digitalWrite(ind2, LOW);  
delay(stepDelay);  
}  
  
void stopMotor1() {  
    digitalWrite(inA1, LOW);  
    digitalWrite(inA2, LOW);  
    digitalWrite(inB1, LOW);  
    digitalWrite(inB2, LOW);  
}  
  
void stopMotor2() {  
    digitalWrite(inc1, LOW);  
    digitalWrite(inc2, LOW);  
    digitalWrite(ind1, LOW);  
    digitalWrite(ind2, LOW);  
}
```

// the loop routine runs over and over again forever:

```
void loop() {
```

```
  int i=0,j=0;
```

```
  while (Serial.available()) {
```

```
    delay(3); //delay to allow buffer to fill
```

```
    if (Serial.available() >0) {
```

```
      char c = Serial.read(); //gets one byte from serial buffer
```

```
      readString += c; //makes the string readString
```

```
    }
```

```
  }
```

```
  if (readString.length() >0) {
```

```
    Serial.println(readString); //see what was received
```

```
    substr1 = readString.substring(0, 4); //get the first four characters  
    readString.substring(a, b-1)
```

```
    substr2 = readString.substring(4, 8); //get the next four characters
```

```
Serial.println(substr1); //print to serial monitor to see parsed results
```

```
Serial.println(substr2);
```

```
    i = substr1.toInt(); // converting your substrings to integer to use them to position  
your satellite dish !!!
```

```
    j = substr2.toInt();
```

```
readString=""; // emptying the readString to read in it next time
```

```
}
```

```
Serial.println(i); //print to serial monitor to see parsed results
```

```
Serial.println(j);
```

```
if(i>0){
```

```
for(int l=0;l<256;l++)
```

```
{
```



```
if(i>0){  
    step11();  
    i--;  
}  
if(i>0){  
    step12();  
    i--;  
}  
if(i>0){  
    step13();  
    i--;  
}  
if(i>0){  
    step14();  
    i--;  
}  
}  
i=0;  
stopMotor1();  
}
```

```
if(i<0){
for(int l=0;l<256;l++){
    if(i<0){
        step13();
        i++;
    }
    if(i<0){
        step12();
        i++;
    }
    if(i<0){
        step11();
        i++;
    }
    if(i<0){
        step14();
        i++;
    }
}
```

```
}  
  
i=0;  
  
stopMotor1();  
  
}  
  
delay(1000);  
  
  
if(j>0){  
    for(int l=0;l<128;l++)  
    {  
  
        if(j>0){  
            step21();  
            j--;  
        }  
  
        if(j>0){  
            step22();  
            j--;  
        }  
  
        if(j>0){  
            step23();
```

```
j--;  
}  
    if(j>0){  
step24();  
    j--;  
    }  
}  
j=0;  
stopMotor2();  
}  
  
if(j<0){  
    for(int l=0;l<128;l++){  
        if(j<0){  
            step23();  
            j++;  
        }  
        if(j<0){  
            step22();  
            j++;  
        }  
    }  
}
```

```
}  
    if(j<0){  
        step21();  
        j++;  
    }  
    if(j<0){  
        step24();  
        j++;  
    }  
  
}  
j=0;  
stopMotor2();  
  
}  
delay(1000);  
  
}
```