WIRELESS SOURCE DIRECTION FINDER

By

Sumair Amin Rana

Umer Tariq

Hasnain Shafiq

Abdus Salam Wanis

Project Supervisor

Assistant Professor Zeeshan Zahid

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

JUNE 2014

ABSTRACT

The project is able to detect and specify the direction of any wireless source within a certain range and sector. Using specific frequency modules, the wireless source is detected with maximum accuracy in least time. The distance from the source is calculated based on difference in power levels of the two receivers. The resultant relative position of the source will be displayed on screen after the processed signals are coded through programming tools. Geometrical analysis technique of trilateration is employed in our coding. A rotating mechanism may also be employed to make the device rotate and detect from all direction within the specific sector. For our radio frequency direction finding, we designed three wireless modules operating at frequency of 2.4 GHz. The two used receivers give us two different received power levels. Based on that difference, our IC will calculate the location of the source relative to the receiving antennas. The final results (coordinates/angle) will be displayed on LCD. This project can be enhanced by adding a GPS interface for worldwide usage. Also, this project’s concept can be employed with other frequency bands and increase the range as per the requirements.

DECLARATAION

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.

DEDICATAION

Allah, beginning with the name of – the most gracious, the most merciful

Dedicated to the ones who have always been praying for us, rather than for themselves

ACKNOWLEDGEMENTS

Allah has showered more blessings upon us than we can ever realize. We are grateful to our parents whose support kept us going in the bleakest of hours. We thank our friends for their help and encouragement.

We extend our gratitude to our project supervisor Asst Prof Zeeshan Zahid for his continuous technical guidance and backing throughout the project. His calm nature ensured smooth work and his technical expertise in the field always kept us on track

We would like to extend special thanks to all the people who helped us out during the course of this project, whether in limited capacity or much.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATION LIST

GUI	Graphical User Interface
IC	Integrated Circuit
RF	Radio Frequency
DF	Direction Finding
dB	Deci-bells
LT	Linear Technologies
GHz	Giga Hertz
Wi-Fi	Wireless Fidelity
RDF	Radio Direction Finding
USB	Universal Serial Bus
EDR	Enhanced Data Rate
A/D	Analog/ Digital
I/O	Input/ Output
LCD	Liquid Crystal Display
SPP	Serial Port Profile
UART	Universally Asynchronous Receiver/ Transmitter
PCBA	Printed Circuit Board Assembly

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

Direction finding is an important technique in modern security surveillance technologies. It is an imperative part of all telecom related equipment and devices. The implementation of direction finding can be used for hunting down interfering radio signals, both accidental and malicious interference to repeaters. Helping to locate downed aircraft by Direction Finding their emergency locator beacons (ELT) is also a utilization of this technique.

The project has its domain in antenna design as well which is a very tricky task to undertake. The received signal levels are very hard to maintain at a sufficient level. Following are some of the application areas of our project:

- Locating Harmful Interference
- Jammers
- Stuck transmitters
- Local noise sources
- Search and Rescue
- FRS/ham radios
- Wildlife location
- Navigation

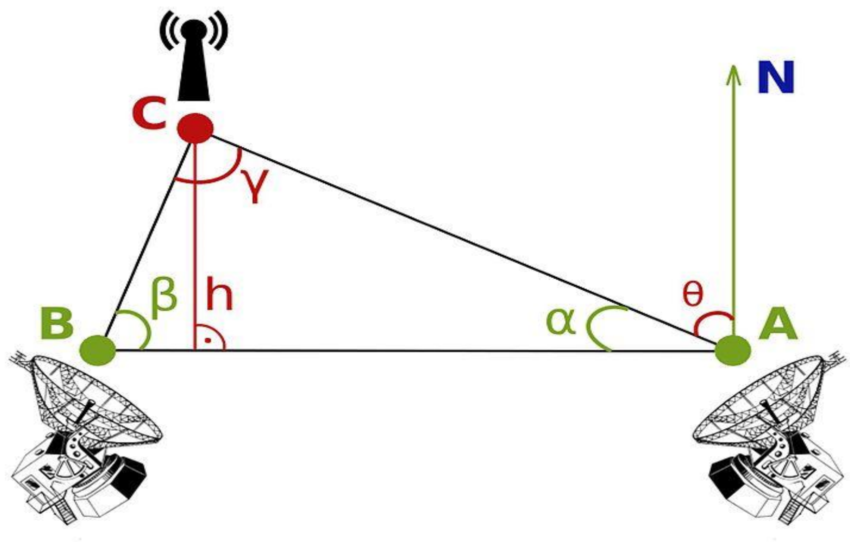


Figure 1-1 General realization

1.2 PROJECT OVERVIEW

Our aim is to design a wireless source detector which will accurately specify the direction of the source. The project was undertaken with a different approach initially and later on, the revamped strategy was implemented.

1.3 OBJECTIVES

- Design a transceiver using ZigBee module for transmission and reception of signals.
- With help of Arduino UNO board develop an algorithm for computation of direction.

- To show final results using visual programming software to make the desired GUI.

1.4 OUTLINE OF TASKS

Firstly, literature research was carried out where various components were studied to understand the architecture, working and different configurations in which they can be integrated together. Various sensors along with specifications and features were studied. This enabled us to select suitable sensors to meet our project requirement. Further study was conducted on the selected hardware. Then the hardware was appropriately coded and configured using hardware specific operating systems and programming languages.

Finally, all the modules and hardware elements were individually optimized and then integrated. Every stage of integration was tested and thoroughly debugged. A user friendly GUI was developed for ease of operation and to make the system more interactive.

1.5 ORGANIZATION OF THE DOCUMENT

Chapter 1 contains the Introduction to the project. Chapter 2 highlights the literature review while Chapter 3 and 4 entails the Hardware design of the project and its development respectively. Chapter 5 explains the software development and programming details. Chapter 6 gives the conclusion and future work that can be taken upon for further improvements in the project.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION

This chapter aims to present all the necessary background knowledge regarding the components used in the project development.

2.1.1 Different Methods of Direction Finding

Direction finding (DF) systems provide several important functions in modern EW systems. Measuring the emitter's bearing or angle of arrival (AOA) is an important and an invariant sorting parameter in the de interleaving of radar signals and in separating closely spaced communication emitters. In addition, the conservation of jamming power in power-managed ECM systems depends on the ability of the associated ESM system to measure the direction to the victim emitter. A function which is becoming increasingly important in defense suppression and weapon delivery systems involves locating the emitter's position passively. This can be accomplished from a single moving platform through successive measurements of the emitter's angular direction, or from multiple platforms which make simultaneous angular measurements.

The emitter identification function requires identifying and associating consecutive pulses produced by the same emitter in angle of arrival (AOA) and frequency. The AOA is a parameter which a hostile emitter cannot change on a pulse-to-pulse basis. However, to measure the AOA of pulses which overlap in the time domain first requires them to be separated in the frequency domain. The advanced ESM receivers which accomplish this function must operate over several octaves of bandwidth while providing RMS bearing accuracies on the order of at least 2 degrees with high POI and fast reaction time in dense signal environments.

2.2 METHODS

There are basically three methods, depicted in Figure 3, which allow the passive location of stationary ground-based emitters from airborne platforms. These are:

1. The azimuth triangulation method where the intersection of successive spatially displaced bearing measurements provides the emitter location;

2. The azimuth/elevation location technique, which provides a single-pulse instantaneous emitter location from the intersection of the measured azimuth/elevation line with the earth's surface; and

3. The time difference of arrival (TDOA), or precision emitter location system (PELS) method, which measures the difference in time of arrival of a single pulse at three spatially remote locations.

Additional methods include:

4. Phase rate of change, which is similar to triangulation, except it makes calculations using the phase derivative.

5. Angle distance techniques, where the distance from the emitter is derived from the signal strength (with known "threat" characteristics).

6. RF Doppler processing, which measures Doppler changes as the aircraft varies direction with respect to the "target" radar.

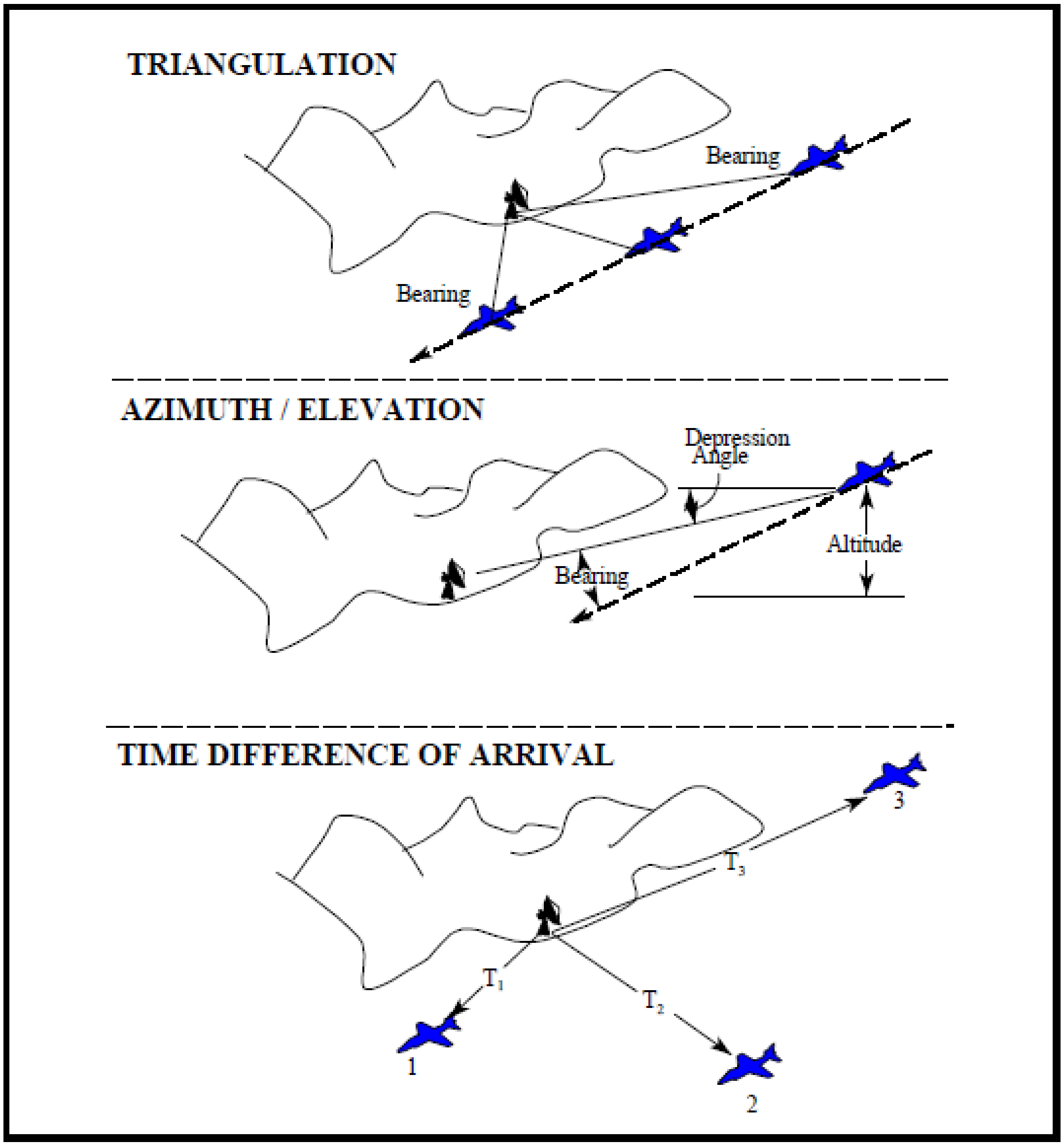


Figure 2.2 Triangulation

Measurement Technique	Advantages	Disadvantages
Triangulation	Single Aircraft	Non-Instantaneous Location; Inadequate Accuracy for Remote Targeting; Not Forward Looking
Azimuth/Elevation	Single Aircraft; Instantaneous Location Possible	Accuracy Degrades Rapidly at Low Altitude; Function of Range
Time Difference of Arrival (Pulsed Signals)	Very High Precision, Can Support Weapon Delivery Position Requirements Very Rapid, Can Handle Short On-Time Threat	Very Complex, At Least 3 Aircraft; High Quality Receivers; DME (3 Sites); Very Wideband Data Link; Very High Performance Control Processor; Requires Very High Reliability Subsystems. Requires common time reference and correlation operation for non-pulse signals.

2.2.1 Angle-Of-Arrival (AoA) Measurements

Several of the above DF measurements require AOA determination. Threat AOA measurements are also required to inform the aircrew in order to position the aircraft for optimal defense.

As shown in Figure 4, angle-of-arrival measuring systems fall into three main system categories of:

1. Scanning beam

2. Amplitude comparison or Simultaneous-multiple-beam

3. Phased Interferometer techniques

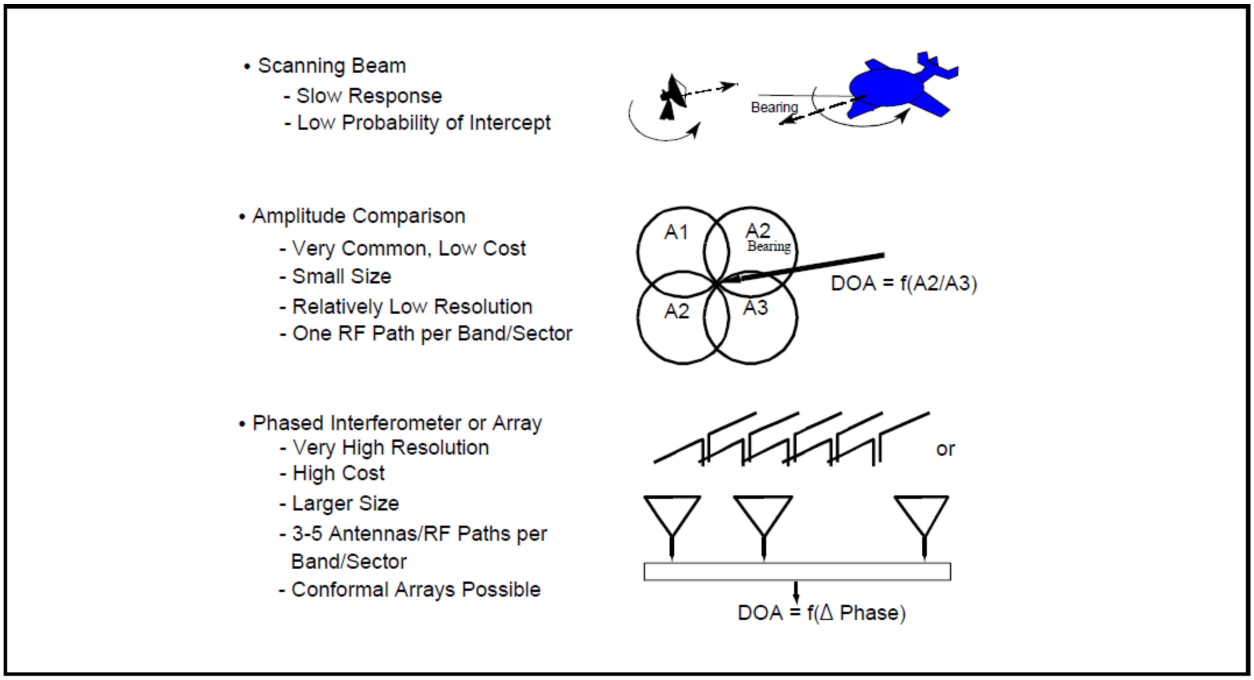


Figure 2.2.1 Direction finding methods

2.2.2 Scanning Beam

The mechanically scanning beam, or "spinner," requires only a single receiver and also exhibits high sensitivity due to the use of a directive antenna. The disadvantage is that the "spinner" usually exhibits slow response because it must rotate through the coverage angle (e.g., 360 degrees) to ensure that it intercepts an emitter. Also, if the emitter uses a scanning directional antenna, both beams must point at each other for maximum sensitivity, which is a low probability occurrence. Both of these effects cause the mechanically scanning beam technique to have a low probability of intercept (POI).

2.2.3 Amplitude Comparison

The two primary techniques used for direction finding are the amplitude-comparison method and the interferometer or phase-comparison method. The phase-comparison method generally has the advantage of greater accuracy, but the amplitude-comparison method is used extensively due to its lower complexity and cost.

Thus, phase interferometers that typically use very wide beam antennas require high signal-to-noise ratios to achieve accurate angle-of-arrival measurements. Alternately, a multi-element array antenna can be used to provide relatively narrow interferometer lobes, which require modest signal-to-noise ratios.

Virtually all currently deployed radar warning receiving (RWR) systems use amplitude-comparison direction finding (DF). A basic amplitude-comparison receiver derives a ratio, and ultimately angle-of-arrival or bearing, from a pair of independent receiving channels, which utilize squinted antenna elements that are usually equidistantly spaced to provide instantaneous 360E coverage.

Typically, four or six antenna elements and receiver channels are used in such 5-8.7 systems, and wideband logarithmic video detectors provide the signals for bearing-angle determination. The mono pulse ratio is obtained by subtraction of the detected logarithmic signals, and the bearing is computed from the value of the ratio.

2.2.4 Simultaneous-Multiple-Beam (Amplitude Comparison)

The simultaneous-multiple-beam system uses an antenna, or several antennas, forming a number of simultaneous beams (e.g., Butler matrix or Rotman lens), thereby retaining the high sensitivity of the scanning antenna approach while providing fast response. However, it requires many parallel receiving channels, each with full frequency coverage. This approach is compatible with amplitude-mono pulse angular measuring techniques which are capable of providing high angular accuracy.

A typical example of a multiple-beam antenna is a 16-element circular array developed as part of a digital ESM receiver. This system covers the range from 2 to 18 GHz with two antenna arrays (2 to 7.5 GHz and 7.5 to 18 GHz), has a sensitivity of -55 to -60 dBm and provides an RMS bearing accuracy of better than 1.7 degrees on pulse widths down to 100 ns.

2.2.5 Phased Interferometer Techniques

The term interferometer generally refers to an array type antenna in which large element spacing occurs and grating lobes appear. Phase interferometer DF systems are utilized when accurate angle-of-arrival information is required. They have the advantage of fast response, but require relatively complex microwave circuitry, which must maintain a precise phase match over a wide frequency band under extreme environmental conditions. When high accuracy is required (on the order of 0.1 to 1E), wide baseline interferometers are utilized with ambiguity resolving circuitry. The basic geometry is depicted in Figure 5, whereby a plane wave arriving at an angle is received by one antenna earlier than the other due to the difference in path length.

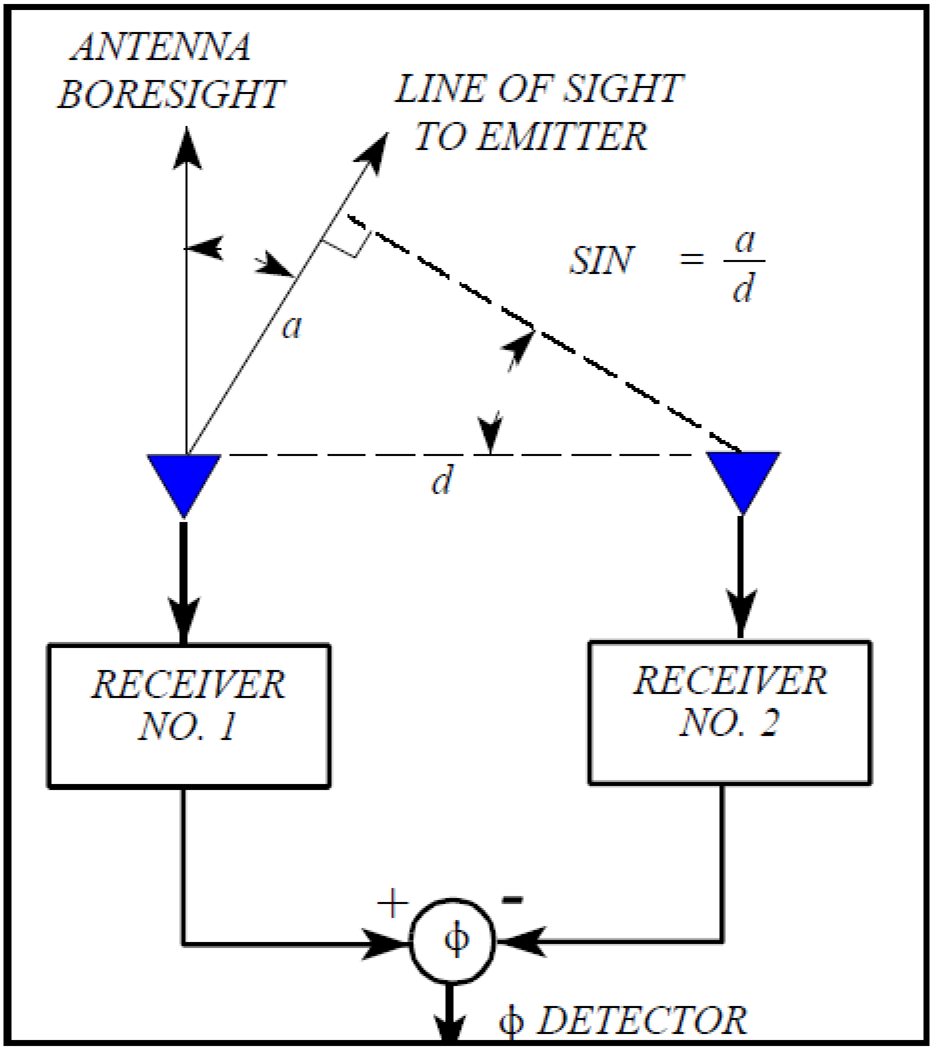


Figure 2.2.5 Direction of Arrival Measurement Techniques

	Amplitude Comparison	Phase Interferometer
Sensor Configuration	Typically 4 to 6 Equispaced Antenna Elements for 360° Coverage	2 or more RHC or LHC Spirals in Fixed Array
DF Accuracy	$DF_{ACC} \approx \frac{\bullet^{2}_{BW} \bullet C_{dB}}{24S}$ (Gaussian Shape)	$DF_{ACC} = \frac{\bullet}{2 \bullet d \cos \bullet} \bullet \bullet$
DF Accuracy Improvement	Decrease Antenna BW; Decrease Amplitude Mistrack; Increase Squint Angle	Increase Spacing of Outer Antennas; Decrease Phase Mistrack
Typical DF Accuracy	3° to 10° rms	0.1° to 3° rms
Sensitivity to Multipath/ Reflections	High Sensitivity; Mistrack of Several dB Can Cause Large DF Errors	Relatively Insensitive; Interferometer Can Be Made to Tolerate Large Phase Errors
Platform Constraints	Locate in Reflection Free Area	Reflection Free Area; Real Estate For Array; Prefers Flat Radome
Applicable Receivers	Crystal Video; Channelizer; Acousto-Optic; Compressive; Superheterodyne	Superheterodyne

2.3 HARDWARE SPECIFICATIONS

Two main hardware components were used in the direction finder.

- ATMEGA 8A Processing Board
- ZigBee Module ETRX35X

2.3.1 Technical Specifications: ATMEGA 8A

The detailed schematic of the transceivers used in the above mentioned design is as shown:

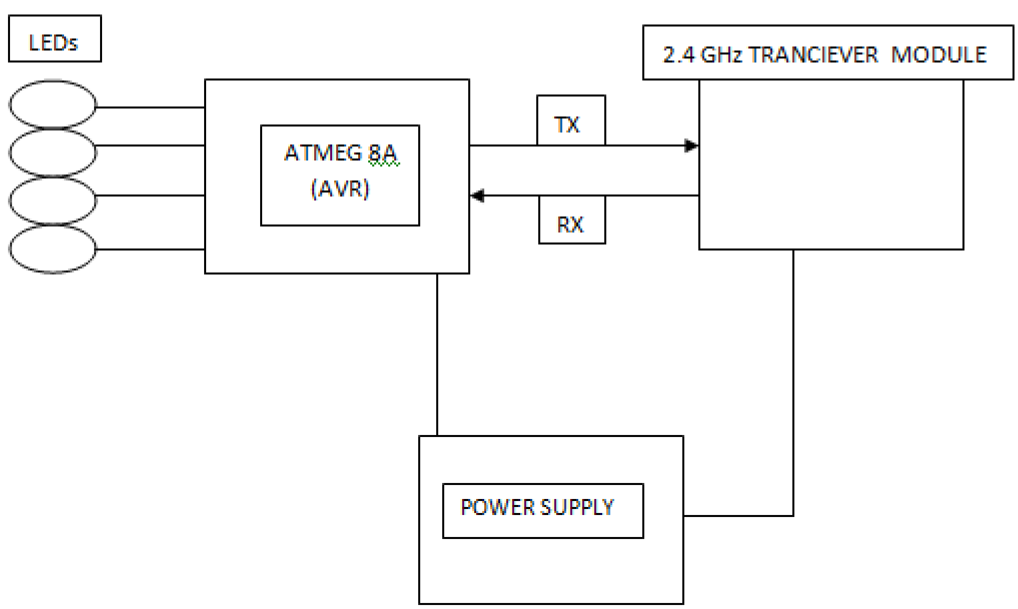


Figure 2.3.1 Transceivers schematic

The components used in the circuit are

- Advanced RISC Architecture
- 130 Powerful Instructions – Most Single-clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16MIPS Throughput at 16MHz
- On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
- 8KBytes of In-System Self-programmable Flash program memory
- 512Bytes EEPROM
- 1KByte Internal SRAM
- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85C/100 years at 25C
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- Programming Lock for Software Security
- Atmel QTouch® library support
- Capacitive touch buttons, sliders and wheels
- Atmel QTouch and QMatrix acquisition
- Up to 64 sense channels
- Peripheral Features
- Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode

- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Three PWM Channels
- 8-channel ADC in TQFP and QFN/MLF package
- Eight Channels 10-bit Accuracy
- 6-channel ADC in PDIP package
- Six Channels 10-bit Accuracy
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Special Microcontroller Features
- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
- 23 Programmable I/O Lines
- 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages

- 2.7 - 5.5V
- 0 - 16MHz
- Power Consumption at 4MHz, 3V, 25□C
- Active: 3.6mA
- Idle Mode: 1.0mA
- Power-down Mode: 0.5µA

The Proteus simulation for the above mentioned transceiver is as follows:

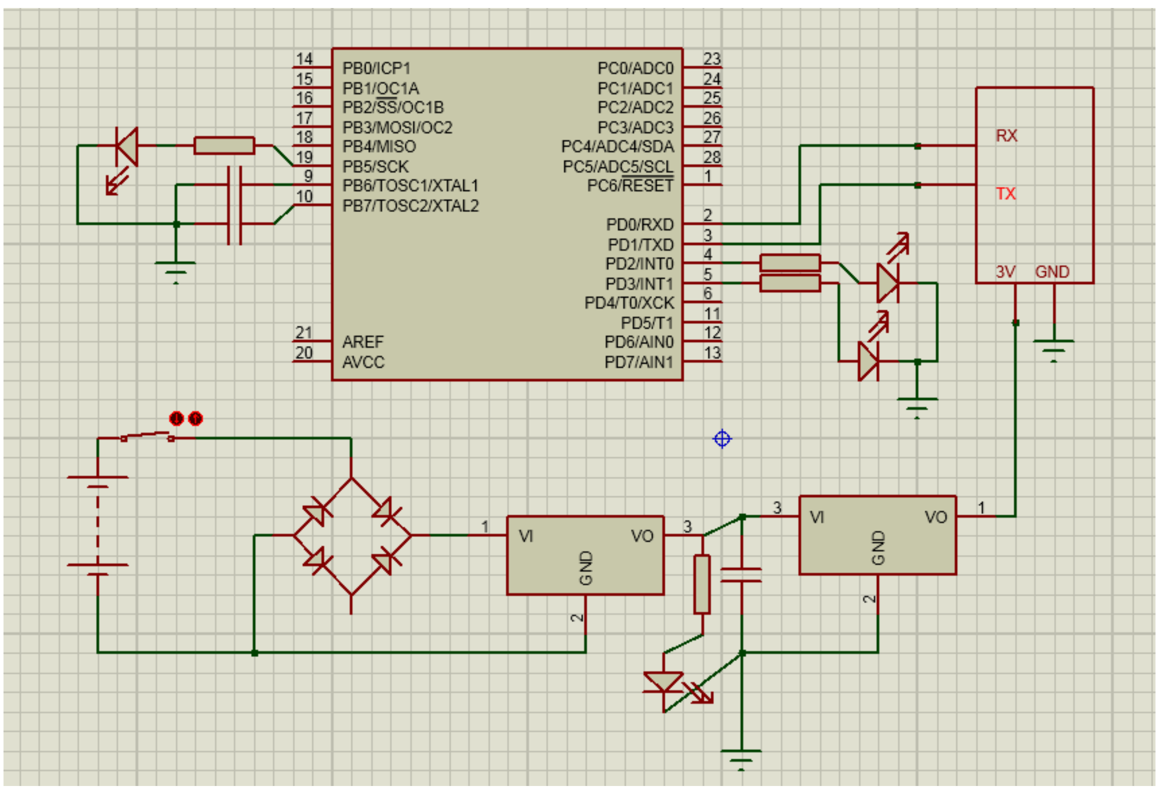


Figure 2.3.2 Proteus simulation

2.3.2 ZigBee Module ETRX35x:

The other major component is **ZIGBEE** module **ETRX35X**. This module is of two types (LRS & HRS) one with antenna and other without antenna but antenna can be mounted if required.

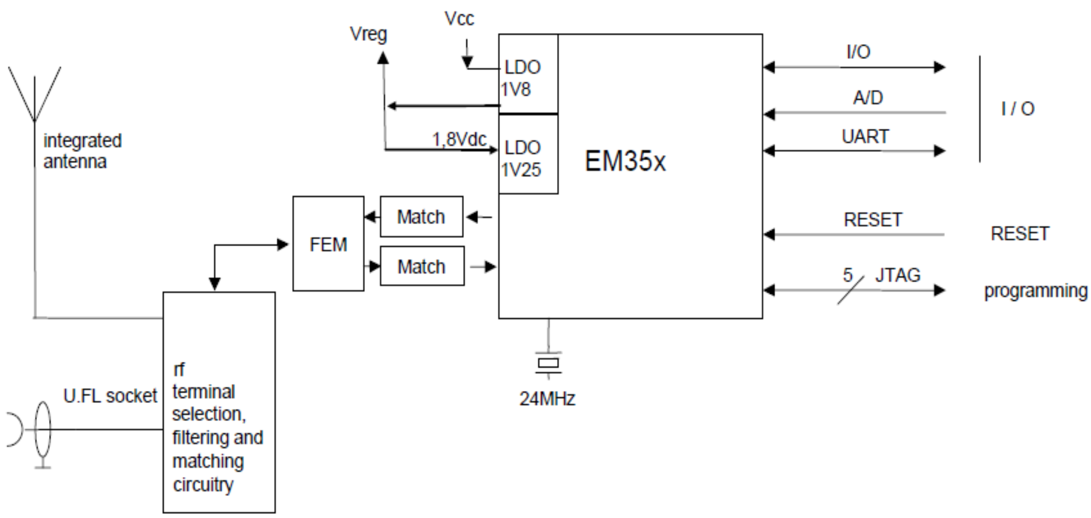


Figure 2.3.3 Zigbee module circuitry

ZigBee long range modules which have been designed to be easily integrated into another device provide a fast, simple and low cost wireless mesh networking interface. AT Command set often allows the ETRX3 series ZigBee modules to be used without an additional host microcontroller saving even more integration time and costs. The main building blocks of the ETRX351-LRS and ETRX357-LRS are the single chip EM351 and EM357 from Ember, a SiGe SE2432L frontend module combining a Power Amplifier with a Low Noise Amplifier, a 24MHz reference crystal and RF front-end circuitry optimized for best RF performance. The modules are available with on-board antenna or alternatively a U.FL connector for attaching external antennae. Modules with the U.FL connector are identified by the “HR” suffix.

The ETRX351-LRS and ETRX357-LRS are based on the Ember EM351 and EM357 respectively in addition to a frontend module containing a PA, LNA and RF switch in addition to the RF-Frontend. The EM351 and EM357 are fully integrated 2.4GHz ZigBee transceivers with a 32-bit ARM® Cortex M3TM microprocessor, flash and RAM memory, and peripherals.

The EM35x has fully integrated voltage regulators for both required 1.8V and 1.25V supply voltages. The voltages are monitored (brown-out detection) and the built in power-on-reset circuit eliminates the need for any external monitoring circuitry. A 32.768 kHz watch crystal can be connected externally to pads 3 and 4 in case more accurate timing is required.

Rest is minor components like:

- Capacitor
- Resistor LED
- Voltage regulator
- Bridge rectifier

2.4 DISPLAY

For the resulting graphical interface, a regular LCD screen will be utilized. Our GUI is based in Visual Studio which provides on-screen output and results.

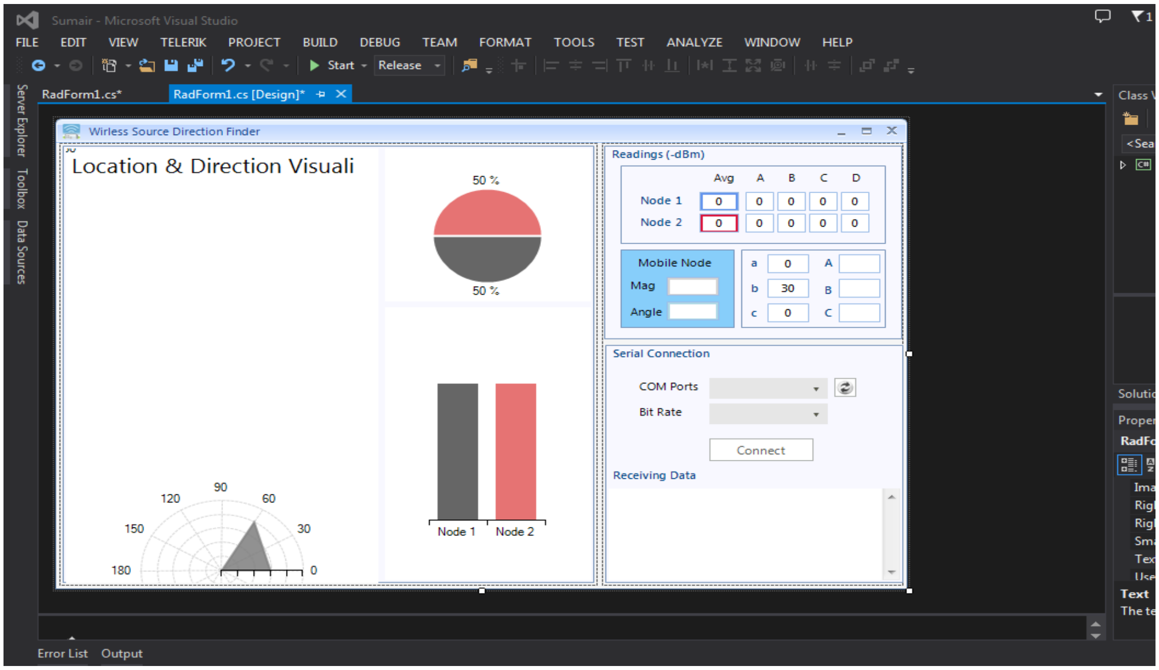


Figure 2.4 Display

2.5 WI-FI FREQUENCY RANGE- BUT NOT WI-FI

All the operations of the ZigBee modules are taking place at 2.4 GHz, yet Wi-Fi is not the technology being used. As mentioned in the component description, ZigBee operates at 900 MHz and 2.4 GHz frequencies using its own specifications. Using Wi-Fi (802.11) was our initial approach but due to extremely noisy environment, we had to change our approach. The desired power levels could not be achieved using Wi-Fi, and singling out signals for a unique source was impossible.

ZigBee is the only standards-based wireless technology designed to address the unique needs of low-cost, low-power wireless sensor and control networks in just about any market. Since ZigBee can be used almost anywhere, is easy to implement and needs little power to operate, the opportunity for growth into new markets, as well as innovation in existing markets, is limitless. Here are some facts about ZigBee:

With hundreds of members around the globe, ZigBee uses the 2.4 GHz radio frequency to deliver a variety of reliable and easy-to-use standards anywhere in the world.

Consumer, business, government and industrial users rely on a variety of smart and easy-to-use ZigBee standards to gain greater control of everyday activities.

With reliable wireless performance and battery operation, ZigBee gives you the freedom and flexibility to do more.

ZigBee offers a variety of innovative standards smartly designed to help you be green and save money.

2.6 COMMUNICATION SCHEME

There was bound to be some interval or delay between the communications. Propagation delay is negligible, though the results will need to be good enough for a real-time environment.

2.7 POWER MANAGEMENT

Our low power system had requirements between 4 and volts and 300milli-Amps. Multiple power sources were considered for this purpose based on their efficiency and fluctuations.

CHAPTER 3 SYSTEM DESIGN AND MODULES

3.1 OVERVIEW

Three transceivers are being utilized, two of them are acting as receivers and one is working as transmitter. Transmitter will first originate a broadcast message to communicate with the remaining two receivers. Receivers will calculate power level from the transmitter using UNICAST commands. These power levels will later be sending to the computing unit for processing. Processing includes calculation of the distance of receivers from the transmitter and the direction of the source from where it is originating.

3.2 TRANSCIEVER UNITS

Omni-directional antennas of 0dBi gain are connected on the two transceivers. Their mounting is such that they are screwed onto the modules, and can be removed if required. Voltage regulation is performed through the use of transistors and clock is used for timing purpose. The blinking of LEDs indicates that reception is taking place.

Following figures show the overall component and nodes design:

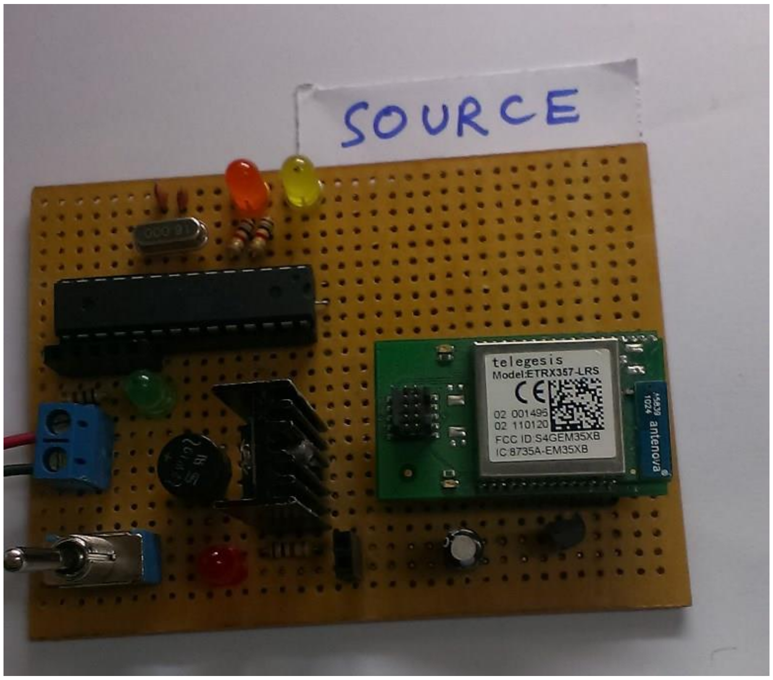


Figure 3.1 Source Transceiver

Mobile node sends broadcast messages in the form of packets.

2 transceivers detect these packets.

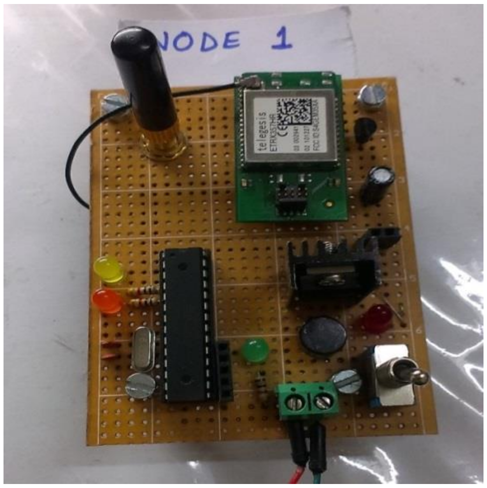


Figure 3.2 Node 1

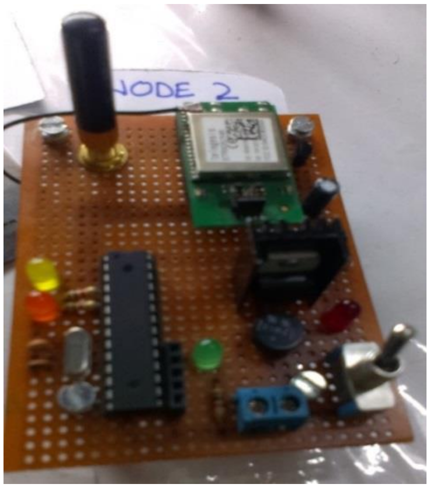
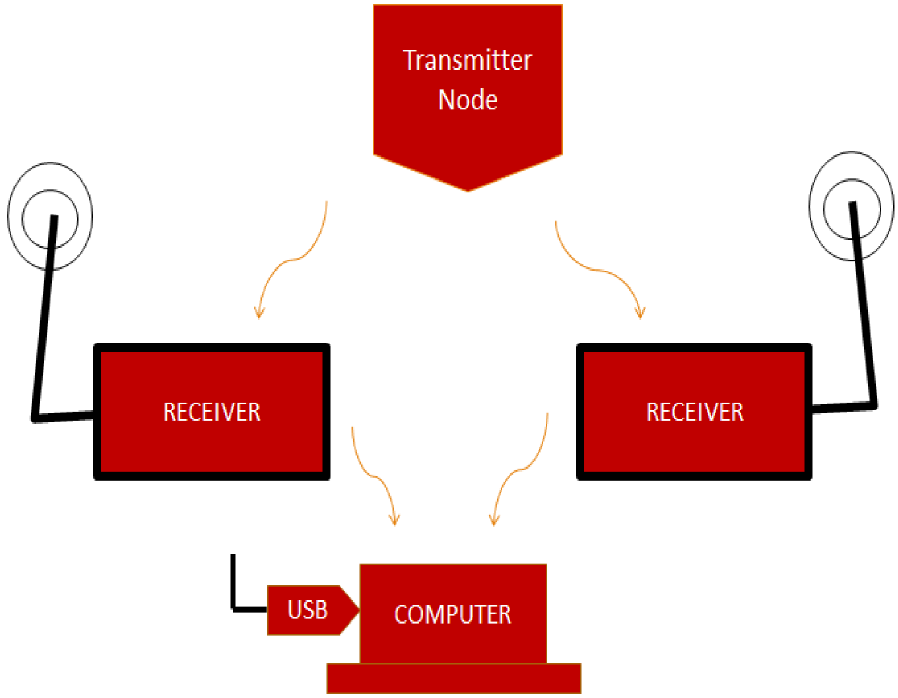


Figure 3.3 Node 2

3.3 SYSTEM DESIGN BLOCK DIAGRAM



’;

Figure 3.4 System Block Diagram

3.4 PROJECT OVERVIEW

In the above block diagram, after the transceivers detect the packets, the microcontroller checks the Received Signal Strength Indicator (RSSI) from packets sent every second. Transceivers send calculated RSSI values to computer in a unicast message. The Control Sys (USB dongle) detects unicast packets and the software application monitors the serial port for received packets using the dongle. The application program monitors the packets to calculate average power after every 4 values and the algorithm developed uses the Power level to visualize the direction. The direction is displayed on GUI on polar coordinate plane.

3.5 CONCLUSION

We managed to achieve the primary goal which was to ascertain the **direction** of a wireless source. The direction has been found in terms of degrees, with reference to both nodes 1 and 2. But this will only be shown through a GUI, which is discussed in the later chapters.

CHAPTER 4 SOFTWARE DEVELOPMENT

4.1 INTRODUCTION

Software is any set of machine-readable instructions (most often in the form of a computer program) that directs a computer's processor to perform specific operations. The term is used to contrast with computer hardware, the physical objects (processor and related devices) that carry out the instructions. Hardware and software require each other; neither has any value without the other. It is the software program which enables the modules to perform their respective functions.

Programming for the project was done in **Visual Studio.** With the help of library functions and some professional help, the required GUI was developed and interfaced with the modules.

4.2 PROGRAMMING THE MICROCONTROLLERS

4.2.1 Programming the transmitter

The transceivers were programed such that it controls LED switching interval, Transmitter announce itself (AT+ANNCE) and this loop repeats every 1100 milliseconds.

Transmitter code:

```
#define led1 13

#define led2 3

#define led3 2
```

```
boolean ZigbeeConnected=false;

void setup()

{

  pinMode(led1,OUTPUT);

  pinMode(led2,OUTPUT);

  pinMode(led3,OUTPUT);

  Serial.begin(19200);

  ledTest();

  SetupZigbee();

}

void loop()

{

  digitalWrite(led1,HIGH);

  digitalWrite(led2,LOW);

  delay(550);

  Serial.println("AT+ANNCE");

  digitalWrite(led1,LOW);
```

```
  digitalWrite(led2,HIGH);
  delay(550);
}
void SetupZigbee()
{
  Serial.println("AT&F");
  delay(1000);
  Serial.println("AT+EN");
  delay(2000);
  Serial.println("ATS10C=1");
  delay(200);
  Serial.println("AT+JN");
}
void ledTest()
{
  digitalWrite(led1,HIGH);
  delay(300);
```

```
digitalWrite(led2,HIGH);

delay(300);

digitalWrite(led3,HIGH);

delay(500);

for(int i=0;i<2;i++)

{

  digitalWrite(led1,LOW);

  digitalWrite(led2,LOW);

  digitalWrite(led3,LOW);

  delay(500);

  digitalWrite(led1,HIGH);

  digitalWrite(led2,HIGH);

  digitalWrite(led3,HIGH);

  delay(500);

}

digitalWrite(led2,LOW);

digitalWrite(led3,LOW);
```

```
}
```

4.2.2 Programming the Transceiver

```
#define led1 13

#define led2 3

#define led3 2

unsigned long ledOnTime=0;

boolean flag=true;

void setup()

{

  pinMode(led1,OUTPUT);

  pinMode(led2,OUTPUT);

  pinMode(led3,OUTPUT);

  Serial.begin(19200);

  ledTest();

  //SetupZigbee();

}
```

```
void loop()
{
 String inputString="";
 x
 while(Serial.available())
 {
  char c= (char) Serial.read();
  inputString += c;
  if(c=='\n')
  {
   if(inputString.startsWith("FFD"))
   {
    Serial.println("AT+SCAST: id_2 "+inputString.substring(27,29));
    if(flag)
    {
     digitalWrite(led2,HIGH);
     digitalWrite(led3,LOW);
```

```
      flag=false;
    }
    else
    {
      digitalWrite(led2,LOW);
      digitalWrite(led3,HIGH);
      flag=true;
    }
    ledOnTime=millis();
  }
  inputString="";
 }
}
delay(50);
 if(millis()>(ledOnTime+4000))
{
 digitalWrite(led2,LOW);
```

```
    digitalWrite(led3,LOW);
    ledOnTime=millis();
  }
}
void SetupZigbee()
{
  //Serial.println("AT&F");
  //Serial.println("AT+DASSL");
  delay(100);
  Serial.println("ATS10C=1"); //show rssi for annce msgs
  delay(200);
  Serial.println("ATS0FE=1"); //show rssi for unicast msg
  delay(200);
  Serial.println("ATS108=1"); //Actively search for a sink if none is known
  delay(200);
  Serial.println("ATS0AD=1"); //Set if router do not route any messages
  delay(200);
```

```
  //Set it as end device
  Serial.println("ATS0AE=0"); delay(200);
  Serial.println("ATS0AF=1");
  delay(200);
  Serial.println("AT+JN");
  delay(2000);
}
void ledTest()
{
  digitalWrite(led1,HIGH);
  delay(300);
  digitalWrite(led2,HIGH);
  delay(300);
  digitalWrite(led3,HIGH);
  delay(500);
  for (int i=0;i<2;i++)
  {
```

```
    digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
    delay(500);
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    delay(500);
  }
  digitalWrite(led2,LOW);
  digitalWrite(led3,LOW);
}
```

4.3 PROGRAMMING THE APPLICATION SOFTWARE

The application software developed in Visual Studio reads the serial port for any changes. It then takes average of 4 RSSI values, to cater for noise and fluctuations in received signal levels. This loop continues after 1100 milliseconds. The packets sent at serial port contain the power information of the two receiving nodes. This information is used to calculate direction of the source using the law of triangulation.

4.3.1 Reading the Serial Port

```
{
    private SerialPort sp;
    private Queue<int> Node1, Node2;
    private float Node1_RSSI = 0, Node2_RSSI = 0;
    private float Node1_x = -15, Node2_x = 15;
    public RadForm1()
    {
        InitializeComponent();
        sp = new SerialPort();
        btn_RefreshPorts_Click(null, EventArgs.Empty);
        cb_bitrate.Items.Add("9600");
        cb_bitrate.Items.Add("19200");
        cb_bitrate.Items.Add("115200");
        cb_bitrate.SelectedIndex = 1;
        if (cb_ports.Items.Contains("COM13"))
```

```
{
    cb_ports.SelectedIndex = cb_ports.Items.IndexOf("COM13");
}
else
    cb_ports.SelectedIndex = 0;
Node1 = new Queue<int>(4);
Node2 = new Queue<int>(4);
for(int i=0;i<4;i++)
{
    Node1.Enqueue(0);
    Node2.Enqueue(0);
}
radChartView1.GetSeries<ChartSeries>(0).BackColor = Color.CornflowerBlue;
radChartView1.GetSeries<ChartSeries>(1).BackColor = Color.Crimson;
PolarSeries ps = radChartView3.GetSeries<PolarSeries>(0);
ps.PointSize = new SizeF(10, 10);
```

```
    ps.DataPoints.Get<PolarDataPoint>(0).Label = "Node 1";
    ps.DataPoints.Get<PolarDataPoint>(1).Label = "Node 2";
    ps.DataPoints.Get<PolarDataPoint>(2).Label = "Mobile Node";
    ps.LabelDistanceToPoint = 20;
    ps.ForeColor = Color.Brown;
    //ps.LabelAngle = 270;
    //ps.BackColor = Color.SandyBrown;
    //ps.ShowLabels = true;
    //radChartView3.ShowSmartLabels = true;
    PolarAxis px = (PolarAxis) radChartView3.Axes[0];
    px.MajorStep = 5;
    px.LineWidth = 0;
    PolarArea pa = radChartView3.GetArea<PolarArea>();
    pa.BorderDashStyle = System.Drawing.Drawing2D.DashStyle.Dot;
}
void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
```

```
try
{
this.Invoke(new EventHandler(
  delegate
  {
      try
      {
          String[] SerialData = sp.ReadExisting().Split('\n');
          foreach (string data in SerialData)
          {
              int id = 0;
              if (data.Contains("id_1"))
              {
                  tb_SerialData.SelectionColor = Color.CornflowerBlue;
                  id = 1;
              }
              else if (data.Contains("id_2"))
```

```
            {
                tb_SerialData.SelectionColor = Color.Crimson;
                id = 2;
            }
            else
                tb_SerialData.SelectionColor = Color.Black;

            tb_SerialData.AppendText(data);
            tb_SerialData.ScrollToCaret();
            if (id != 0)
                processReading(id, data);
        }
    }
    catch(Exception ee)
    {
    }
}));
```

```
}

catch(Exception ee)

{

    RadMessageBox.Show(ee.Message);

}
```

4.3.1 Takes average of 4 RSSI values

```
private void processReading(int id, string data)

  {

     try

     {

        int value;

                    if (int.TryParse(data.Substring(32), out value))

        {

           if (id == 1)

           {

              Node1.Dequeue();
```

```
Node1.Enqueue(value);

int[] values = Node1.ToArray();

float avg=0;

foreach(int v in values)

{

    avg += v;

}

avg = avg / 4;

Node1_RSSI = avg;

side_a.Text = tb_Node1Avg.Text = Node1_RSSI.ToString();

tb_Node1A.Text = values[3].ToString();

tb_Node1B.Text = values[2].ToString();

tb_Node1C.Text = values[1].ToString();

tb_Node1D.Text = values[0].ToString();

lock(this)

radChartView1.GetSeries<ChartSeries>(0).DataPoints.Get<CategoricalDataPoint>(0).Va
lue = Node1_RSSI;

}
```

```
else
{
    Node2.Dequeue();
    Node2.Enqueue(value);
    int[] values = Node2.ToArray();
    float avg = 0;
    foreach (int v in values)
    {
        avg += v;
    }
    avg = avg / 4;
    Node2_RSSI = avg;
    side_c.Text = tb_Node2Avg.Text = Node2_RSSI.ToString();
    tb_Node2A.Text = values[3].ToString();
    tb_Node2B.Text = values[2].ToString();
    tb_Node2C.Text = values[1].ToString();
    tb_Node2D.Text = values[0].ToString();
```

```
                lock(this)
radChartView1.GetSeries<ChartSeries>(1).DataPoints.Get<CategoricalDataPoint>(0).Va
lue = Node2_RSSI;
            }
            ChartSeries ps1 = radChartView2.GetSeries<ChartSeries>(0);
            ps1.DataPoints.Get<PieDataPoint>(0).Value = Node1_RSSI;
            ps1.DataPoints.Get<PieDataPoint>(1).Value = Node2_RSSI;
           CalculateDirection();
        }
        else
            tb_SerialData.AppendText("Unable to Parse Readings\n\n");
    }
    catch (Exception ee)
    {
        RadMessageBox.Show(ee.Message);
    }
}
```

4.3.2 Calculating Direction

```
private void processReading(int id, string data)

    {

        try

        {

            int value;

                        if (int.TryParse(data.Substring(32), out value))

            {

                if (id == 1)

                {

                    Node1.Dequeue();

                    Node1.Enqueue(value);

                    int[] values = Node1.ToArray();

                    float avg=0;

                    foreach(int v in values)

                    {

                        avg += v;

                    }
```

```
avg = avg / 4;

Node1_RSSI = avg;

side_a.Text = tb_Node1Avg.Text = Node1_RSSI.ToString();

tb_Node1A.Text = values[3].ToString();

tb_Node1B.Text = values[2].ToString();

tb_Node1C.Text = values[1].ToString();

tb_Node1D.Text = values[0].ToString();

lock(this)
radChartView1.GetSeries<ChartSeries>(0).DataPoints.Get<CategoricalDataPoint>(0).Va
lue = Node1_RSSI;

}

else

{

Node2.Dequeue();

Node2.Enqueue(value);

int[] values = Node2.ToArray();

float avg = 0;

foreach (int v in values)
```

```
                {
                    avg += v;
                }
                avg = avg / 4;
                Node2_RSSI = avg;
                side_c.Text = tb_Node2Avg.Text = Node2_RSSI.ToString();
                tb_Node2A.Text = values[3].ToString();
                tb_Node2B.Text = values[2].ToString();
                tb_Node2C.Text = values[1].ToString();
                tb_Node2D.Text = values[0].ToString();
                lock(this)
radChartView1.GetSeries<ChartSeries>(1).DataPoints.Get<CategoricalDataPoint>(0).Va
lue = Node2_RSSI;
            }
            ChartSeries ps1 = radChartView2.GetSeries<ChartSeries>(0);
            ps1.DataPoints.Get<PieDataPoint>(0).Value = Node1_RSSI;
            ps1.DataPoints.Get<PieDataPoint>(1).Value = Node2_RSSI;
            CalculateDirection();
```

```
        }
        else
            tb_SerialData.AppendText("Unable to Parse Readings\n\n");
    }
    catch (Exception ee)
    {
        RadMessageBox.Show(ee.Message);
    }
}
private void CalculateDirection()
{
    try
    {
        bool flag = true;
        double a, b, c, A, B, C, x = 25, y = 25;
        a = Node1_RSSI;
        c = Node2_RSSI;
```

```
b = int.Parse(side_b.Text);
if (a+b<c || a+c<b || b+c<a)
{
    tb_SerialData.SelectionColor = Color.OrangeRed;
    tb_SerialData.AppendText("\nError! Triangle Un-Constructable\n\n");
}
else
{
    A = Math.Acos(Math.Abs(b * b + c * c - a * a) / (2 * b * c)) * 180 / Math.PI;
    B = Math.Acos(Math.Abs(a * a + c * c - b * b) / (2 * a * c)) * 180 / Math.PI;
    C = Math.Acos(Math.Abs(b * b + a * a - c * c) / (2 * b * a)) * 180 / Math.PI;
    Angle_A.Text = A.ToString();
    Angle_B.Text = B.ToString();
    Angle_C.Text = C.ToString();
    lock (this)
    {
        PolarDataPointpdp=
radChartView3.GetSeries<PolarSeries>(0).DataPoints.Get<PolarDataPoint>(2);
```

```
                    pdp.Value = Node1_RSSI;

                    pdp.Angle = C;

                    tb_MobileNode_X.Text = Node1_RSSI.ToString();

                    tb_MobileNode_Y.Text = Math.Round(C).ToString();

                }

            }

        }

        catch (Exception ee)

        {

            MessageBox.Show(ee.Message, "Size of One Side can not be greater than sum
of other two sides", MessageBoxButtons.OK, MessageBoxIcon.Stop);

        }

    }
```

4.4 GRAPHICAL USER INTERFACE

A user friendly GUI was created for ease of use. Separate buttons enable control of various functions for easy understanding. It consists of various buttons and fields for information processed by the application and its calculations for direction. Direction is displayed in graphical (angular) form.

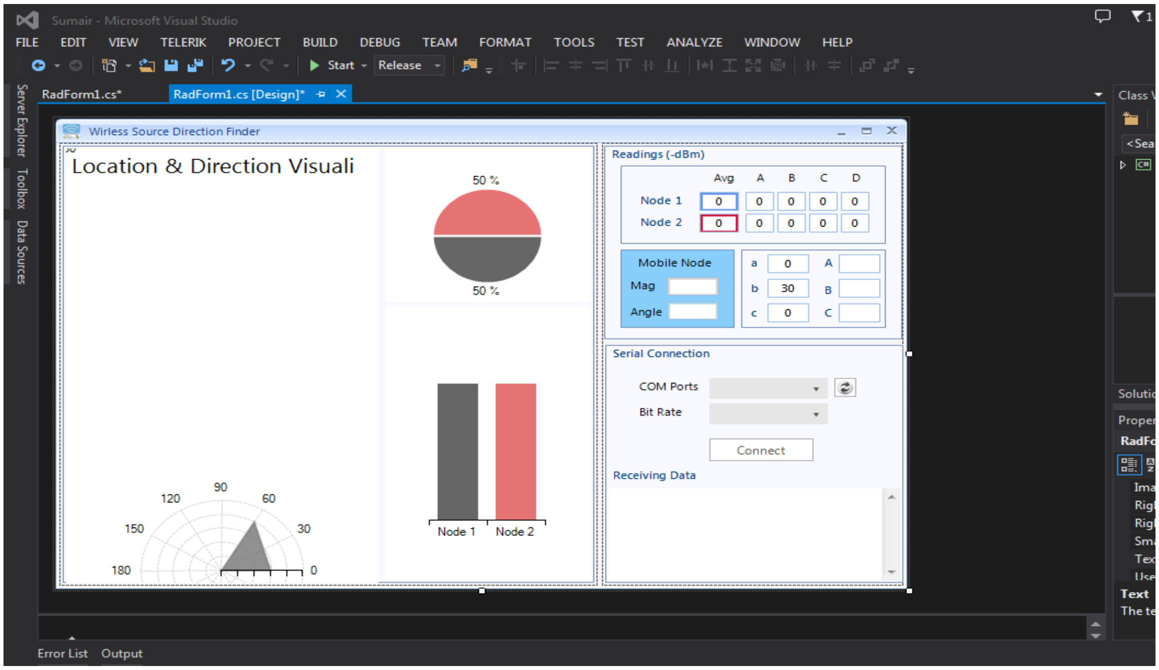


Figure 4.1 GUI

4.5 CONCLUSION

The programming for modules was completed and the programs were fed into the microcontrollers. The graphical application displaying the direction of source was developed. This was our main objective while conceiving the project, which was after this stage.

CHAPTER 5 FUTURE DEVELOPMENT

5.1 POSSIBLE FUTURE DEVELOPMENTS

There is a wide scope of enhancements that can be applied to make the project more accurate and efficient. Some of the possible additions to the project are mentioned below.

5.1.1 Antenna type

The project has been tested in different environments and has been found prone to interferences despite the ZigBee protocol. The signal from the transmitter/source may bounce off the ceiling or nearby appliances and give variations in the location of the source. For the rectification of this issue, **directional antennas** can be employed in place of the Omni directional antenna. This would not only increase the range of our project, but would also provide us with stronger signal strength.

Furthermore, by the use of directional antenna, a given sector can be covered with greater accuracy and range.

5.1.1 Rotating mechanism

A rotating mechanism may also be employed to make the device rotate and detect from all directions. This will require more precise and quick measurements of the RSSI.

5.1.2 GPS Interfacing

The project can be interfaced with GPS to provide worldwide usage and tracking. By knowing the coordinates of the two transceivers, the location of the source signal can be pinpointed. It can also be attached to a GSM Base station which can be then used for a number of applications.

5.1.3 Frequency

The project can be modified to suit the required frequency band. ZigBee also functions in 906 to 924 MHz (10 channels), apart from the 2.4 GHz band (16 channels). This would increase the range and accuracy of the project. Furthermore, this concept can be applied to any desired frequency band by using appropriate transceivers and protocol.

5.1.4 Authentication

User authentication can also be provided as an option. The packets which are being received have the source device name in their headers. These headers can be decoded and unknown sources which are transmitting in the region can be identified.

CHAPTER 6 FUTURE WORK AND CONCLUSION

6.1 INTRODUCTION

Direction finding is an important tool in modern security surveillance technologies. It is an imperative part of all telecom related equipment and devices. The implementation of direction finding can be used for hunting down interfering radio signals, both accidental and malicious interference to repeaters. Helping to locate downed aircraft by Direction Finding their emergency locator beacons (ELT) is also a utilization of this technique.

6.2 PROJECT LIMITATIONS

Batteries were used to operate the two transceivers. The size of the batteries used was 9 volt. This created an unprecedented hindrance due to the fact that the batteries keep draining as time progresses. This, in turn, causes the received signal strengths to vary. This variation resulted in the inaccurate determination of the source's location. The output would show the source to be constantly moving right and left. Hence, our prognosis to this issue is to avoid the use of batteries and rely on permanent sources instead. This will give us a fixed point location of the source without variation.

Wi-Fi environment could not be used, as mentioned earlier.

The use of omni-directional patch antenna limits our range to a great extent. High gain directional antennas can be used to increase range and improve results.

6.3 ADVANTAGES AND APPLICATIONS

The implementation of direction finding can be used for hunting down interfering radio signals, both accidental and malicious interference to repeaters.

Helping to locate downed aircraft by Direction Finding their emergency locator beacons (ELT) is also a utilization of this technique.

Wild-life tracking in large open areas is an application of the project which is already under implementation.

Navigation can be performed around the world, by simply interfacing it with a GPS module.

6.3.1 Surveillance Network

The main task of our project is detection and surveillance. A surveillance network can be created with the help of employing a number of such transceivers working in coordination with each other to monitor a given area and locate any intruder or detect any suspicious activity in that area.

6.3.2. Armored Personnel Carrier (APC) MODE

It can also be fixed on vehicle tops which will make it similar to the APCs having a radar on top. The only difference will be that it can be operated while sitting inside the vehicle.

The initial approach to the project was ambitious, yet unfruitful. Although our revised approach was successful in providing us the answer to our problem statement, there can be certain improvements in the current design. The biggest outcome of this project has been the wide scope of the study involved in it. As a result of the obstacles in our initial approach and the subsequent change in methodology, we achieved substantial know how of both aspects. Hence the biggest achievement has been our exposure to these fields.

6.3.3 Avalanche rescue

Avalanche transceivers operate on a standard 457 kHz, and are designed to help locate people and equipment buried by avalanches. Since the power of the beacon is so low the directionality of the radio signal is dominated by small scale field effects[8] and can be quite complicated to locate.

6.3.4 Wildlife tracking

Location of radio-tagged animals by triangulation is a widely applied research technique for studying the movement of animals. The technique was first used in the early 1960s, when the technology used in radio transmitters and batteries made them small enough to attach to wild animals, and is now widely deployed for a variety of wildlife studies. Most tracking of wild animals that have been affixed with radio transmitter equipment is done by a field researcher using a handheld radio direction finding device. When the researcher wants to locate a particular animal, the location of the animal can be triangulated by determining the direction to the transmitter from several locations.

CHAPTER 7 BIBLIOGRAPHY

- Watson Watt, R.A., and Herd, J.F., "An Instantaneous Direct-Reading Radio goniometer" , lEEE, Vol. 64, pp. 611-617, 1926

- Earp, C.W., and Godfrey, R.M., "Radio Direction-Finding by the Cyclical Differential Measurement of Phase", lEEE, Vol. 94, Part Ilia, pp. 705-721, 1947.

- Rainer, R., and Burwasser, A.J., "An Approach to HF Tactical Radio Direction Finding and Signal Monitoring", Journal of Electronic Defense, pp. 173-184, October, 1983.

- Mumford, W.W., and Scheibe, E.H., "Noise Performance Factors in Communication Systems, Horizon House, Dedham, Massachusetts, 1968.

- Stremler, F.G., "Introduction to Communication Systems", Adison-Wesley Publishing Company, Toronto, Canada, pp. 159-161, 1977.

- "Mobile VHF-UHF Broadband Doppler Direction Finder PA 555", Technical Data Sheet, Rohde and Schwarz, Germany.

- "E210 System Direction Finding Equipment - 20 to 500 MHz", Technical Data Sheet, EM Systems Inc., California

- Review of conventional tactical radio direction finding systems by W. Read communications electronic warfare section electronic warfare division.

- A novel approach to radio direction finding and detecting the number of sources simultaneously: DMSAE algorithm by M. Movahhedi IEEE.

- Hf radio direction finding by dr. David Sadler

- Ivor Horton's beginning visual C++ 2010 by Ivor Horton

- Power supply design basics by P. Antoniazzi