A PROJECT REPORT ON

Flood Alert and Rescue System

for

Indian Institute of Information Technology

Submitted By

Arpit Jain	**2011028**
Diksha Moolchandani	**2011046**
Jayant Sharma	**2011059**
Mohit Kumar Singh	**2011082**
Suraj Soni	**2011155**

Bachelor of Technology

Under the guidance

of

Mr. Awadesh Kumar Singh
Design Discipline,
Indian Institute of Information Technology Design and Manufacturing Jabalpur

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

ABSTRACT

Flood alert and Rescue System accomplishes two tasks i.e. providing continuous alert information to Disaster Rescue Force and location of victims of flood. It also provides the location of the stuck people in flood. Flood alert system can then be used to provide alert message to people. On the contrary, Rescue System guides the rescue force to the location of those in emergency situation.

For the rescue operation, the people in need will press the switch provided to them by the government. The encoded message signal emitted by it will be received and decoded by the signal processing unit by the rescue force. As such they will know the where-about of the person and rescue him in the shortest time which wouldn’t have been possible by the existing technologies. This is so because our system sets the priorities of the rescue force.

The design of our prototype is produced keeping in mind that the device can be easily inculcated in the chores of people during the time of floods. For sustaining this principle our product has been embedded with a torch that can be appropriately used during the shortage of light in time of floods.

Our project is not only restricted to the flood scenario. It can be put to use during any emergency condition keeping in mind the simplicity of functioning of the device so that it remains very user friendly.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Floods are common and most occurring natural disasters in India. When rivers overflow their banks they cause damage to lives, property, infrastructure and crops. Floods usually are local, short-lived events that can happen suddenly and sometimes with little or no warning. Rivers can also flood its surroundings when the dams fail, when ice or a landslide temporarily block the course of the river channel or when snow melts rapidly.

When rivers overflow their normal channels and flows over the land where people live it causes a flood. Again when the river overflows a land where people do not live then it is said to be in flood. This may be due to heavy rain at the catchment, melting of snow, cloudburst etc. Cloudburst generally takes place in hilly areas and results in flash floods.

1.2 NEED STATEMENT

- Most of the people don’t get prior information of the flood. So there is a need to find out the way to give details of the approaching flood.
- Some people are so optimistic and attached to their homes that they assume the water level won’t go above a certain extent. Thus there is a need to give them continuous updates of the current water level.
- Many people are not located by rescue team due to certain circumstances so there should be an efficient way to find them and locate them.

1.3 OBJECTIVE

The objective of this project is to help people communicate with Rescue Team to provide their location either by telephonic communication or by a private network. It also gives continuous updates of the water level so that people get sufficient time to make all the preparations for the upcoming flood.

1.4 SCOPE

The project aims for a network setup by using our devices with the help of Government authorities by mass production and distribution of this device which would reduce its cost very significantly. It would be a very useful pre-planning step in flood prone areas.
This project aims at minimization of life damage caused floods.

CHAPTER 2
CONCEPTS GENERATION

2.1 CONCEPT GENERATION

Problems with rescue operations:

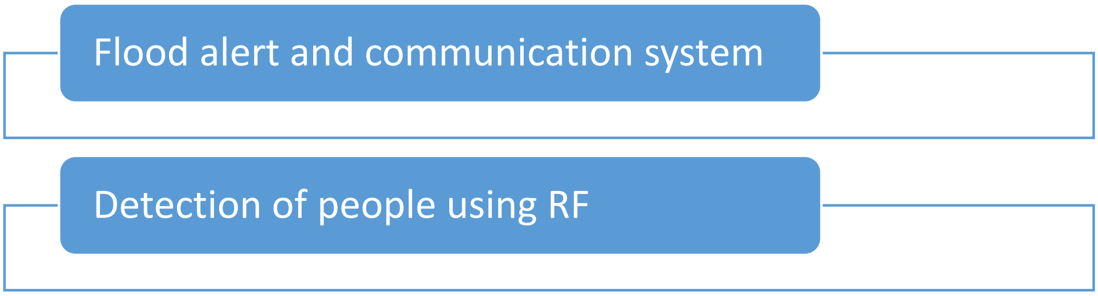
1) Duration of rescue :
    - 1-2 days for low level flood
    - 5-6 days for medium level flood
    - 20-30 days for high level flood
2) Most areas are unreachable areas
3) Limited choppers/boats for food/water supply and evacuation of stranded people
4) Halt in rescue operations due to bad weather
5) Debris of houses make dead bodies searching difficult

Problems with alert systems:

1) Many people reported that they were unaware of the fore coming and happenings of disasters. Even after flood most of them didn't know the safer places and whereabouts of the camps.
2) As reported by National Disaster Rescue Force in survey, most of the times people don't leave their houses because they believe water levels won't go too high. So there is lack of information and communication.

Based on the case study, literature survey and customer survey the following need statements were identified:

- ➢ Alert and information conveying system
- ➢ Efficient search and rescue operations



Flood alert and communication system

Detection of people using RF

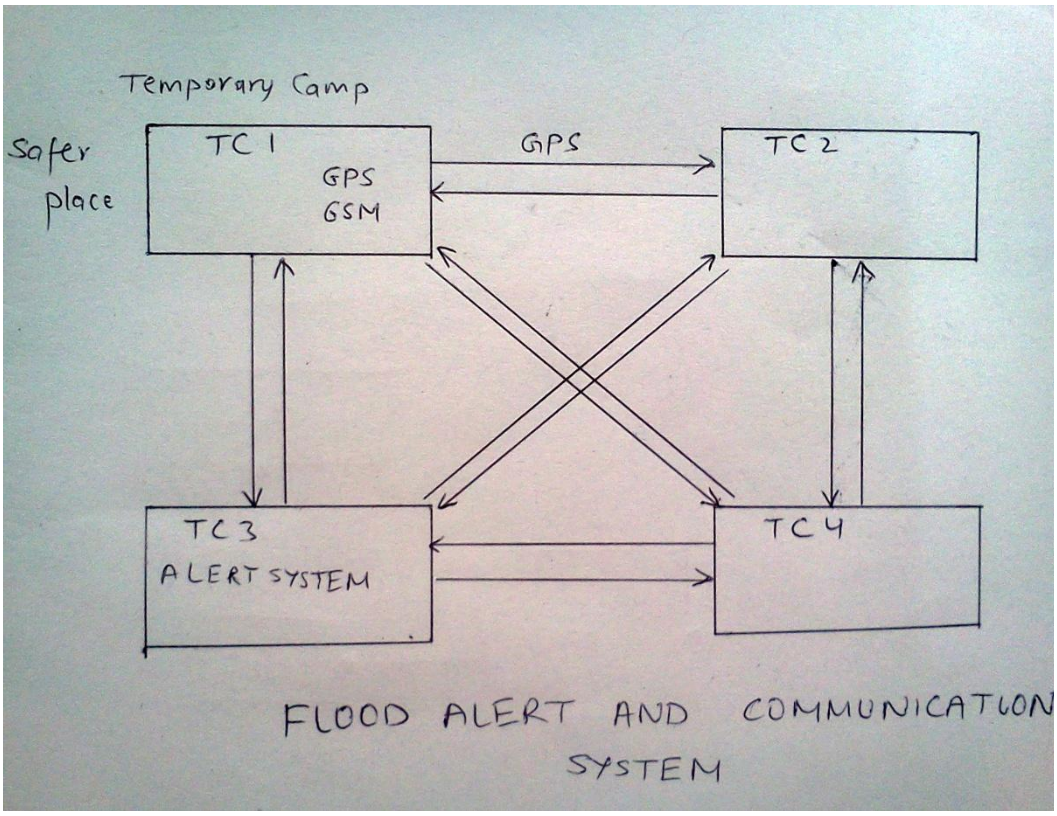


Figure 2.1 Showing communication network to be set at the time of floods

2.2 Feasibility:

- There are some areas at time of flood which are safer where people can gather.
- A personal wireless network can be setup in case all existing communication networks fail.

2.3 Improving concept

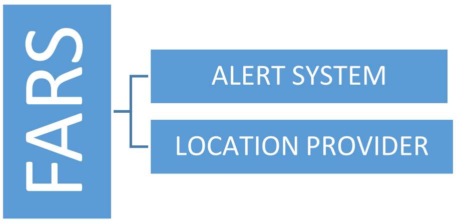
Concept is to make a system that can be used as alert system and also for rescue of stuck people by knowing location of people and communicating with them using phone network and RF communication network for knowing their location.

This system known as **flood alert and rescue system** targets at two major problems –
(i) alert and information conveying
(ii) identifying location of un-rescued victims.

Alert system is being monitored by disaster management team and it provides all details of flood, water levels, location of safer places, shelter camps, medical facilities and other useful information to be conveyed to the mass. It uses GPS based communication system so it requires no cost on people end.

Human detection system can be made implementable and useful if an external device is given to people beforehand in flood prone areas.

Rescue system is a device given pre-flood to a specifically smaller area or houses by the disaster management team. It works if there is no electricity or telephonic communication and people can tell their location using this in case of emergency.



Functions:

1) Alert message just before flood
2) Providing continuous situation report to people during flood
3) Instructions and aid to flood victims
4) Contact with rescue team
5) Knowledge of location of people stuck in flood
6) Torch which can assist during implied situations

CHAPTER 3

EMBODIMENT DESIGN

3.1 AXIOMATIC DESIGN:

Independent functional requirements and minimized functional requirements.

Functional requirements (FR):

1) Alert message just before flood
2) Continuous situation report to people during flood
3) Instructions and aid to flood victims
4) Contact with rescue team
5) Knowledge of location of people stuck in flood
6) Torch assists people and make product a part of their daily chores

Design parameters (DP):

1) Low power design (5V)
2) Mass production cost should be small (approx. Rs 500 for transmitter)
3) Easy to use

Constraints:

1) Low cost
2) Low power consumption
3) Operate without electricity
4) Work without network also
5) No need for refill/reprogram every time
6) Waterproof

3.2 SYSTEM LEVEL DESIGN:

Area affected by flood is divided into several areas of range about 200kms (range of RF). Each of these areas is divided into some sectors, say 8 geographically of same radius from ***center node*** which is receiver end. Each of these sectors have a code allotted to them. Each sector contains some transmitter devices either given to all houses or a street which have this associated code. The code can be configured very easily depending on the location of device.

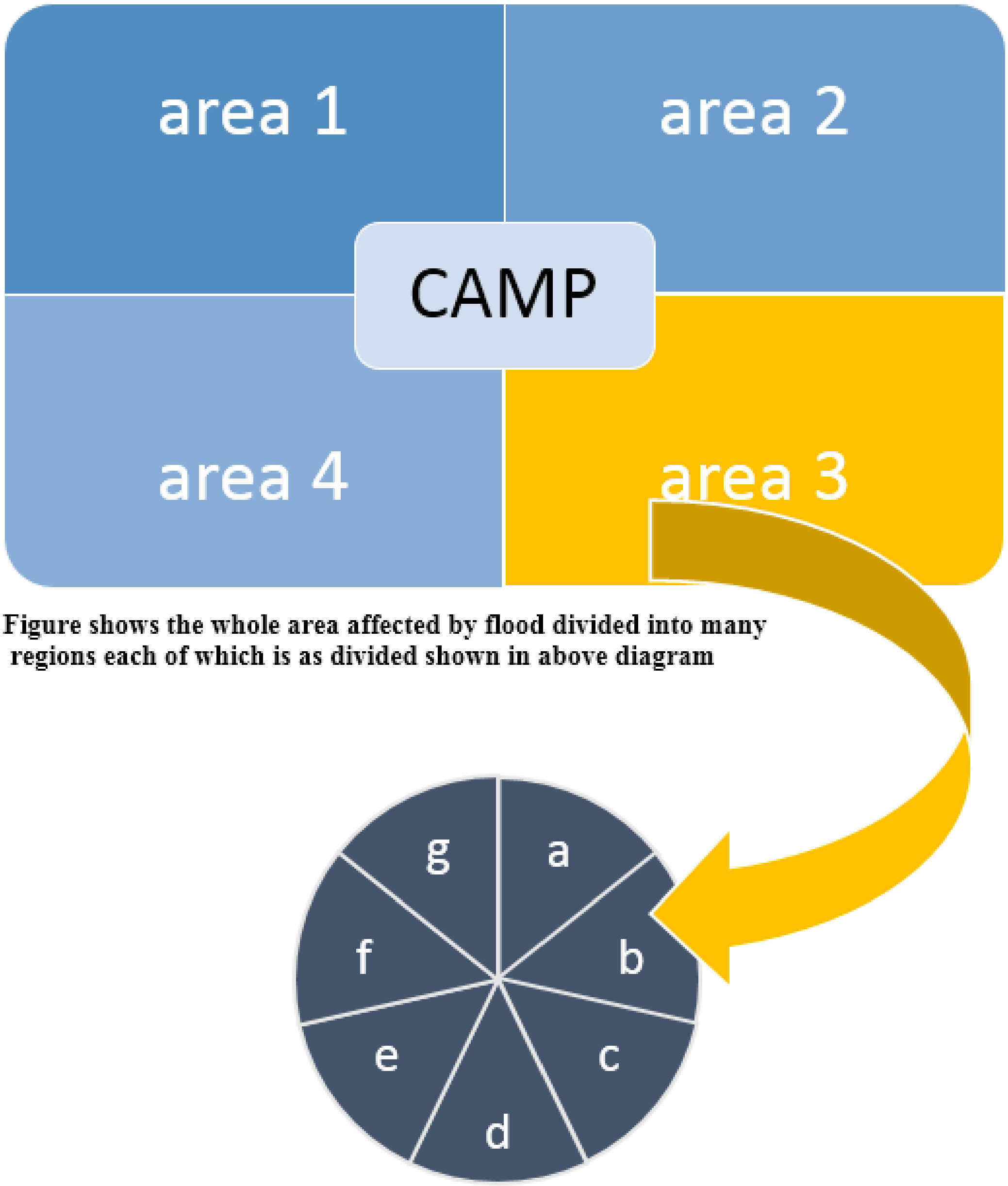


Figure shows the whole area affected by flood divided into many regions each of which is as divided shown in above diagram

Figure 3.1 showing an area 3 divided in sectors.

3.3 ARCHITECTURAL DESIGN:

Modules used are:

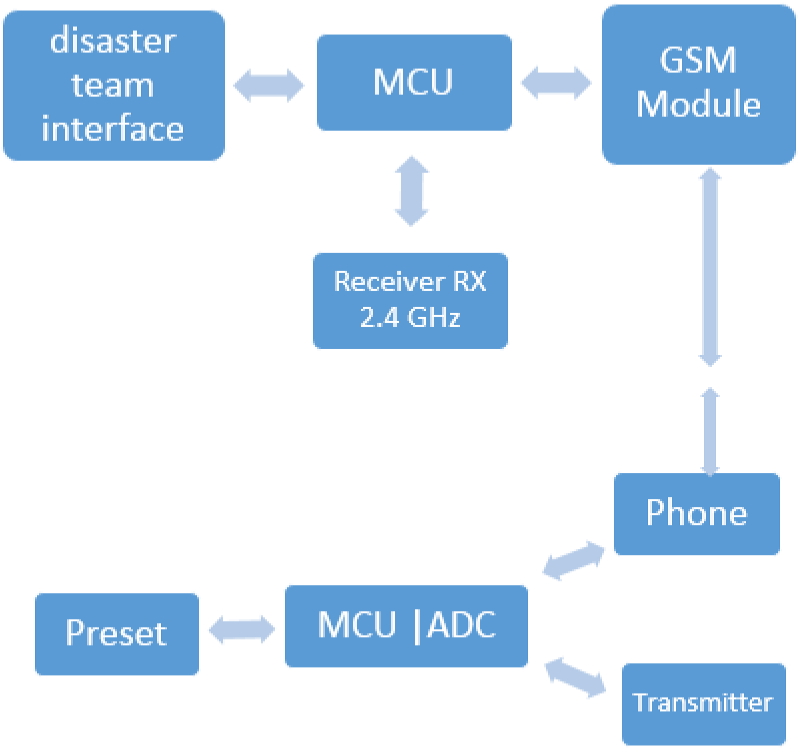


Figure 3.2 showing modules of transmitter sectors and receiver ends

Major functions:

- Transmitting location to central node with phone or without
- Informing people using message.

3.3.1Configuration design:

Modules of design are described as below:

1) **Transmitter :** It is a low powered transmitting device given to people of each sector of area. It sends code of that area to central node. This device has an antenna attached to it and one knob for configuring the code of the area. It contains on PCB a microcontroller unit which has inbuilt Digital to analog convertor and a 5Volts cell.

2) **Receiver :** It receives emergency calls from people either by phone or by rf transmitter. It is accessed by disaster management team which can input its message to be delivered to people during flood and respond to emergency calls. A GSM module and receiver with antenna is used for this purpose.

3) **Controller :** One controller each at receiver and transmitter devices is controlling operations for GSM and transceiver applications.

4) **GSM Module :** It transmits and receives all phone signals for GSM communication. GSM 900 Module is used for this purpose.

5) **PCB :** At transmitter end whole circuit is place don PCB designed using software Kicad to produce standard ***gerber*** files which can be inputted in PCB Design machine.

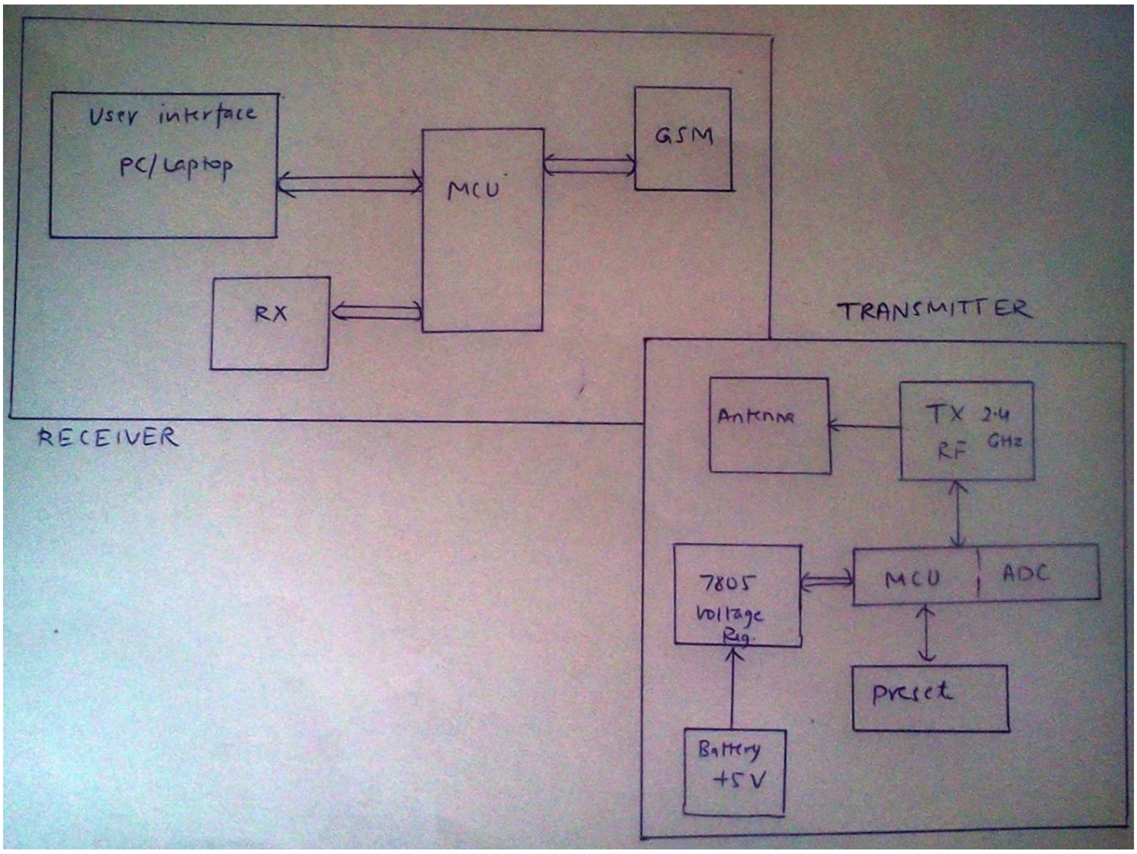


Figure 3.3 showing block diagram of the concept

3.3.2Parametric design :

Transmitter:

Schematic (using EEschema software)

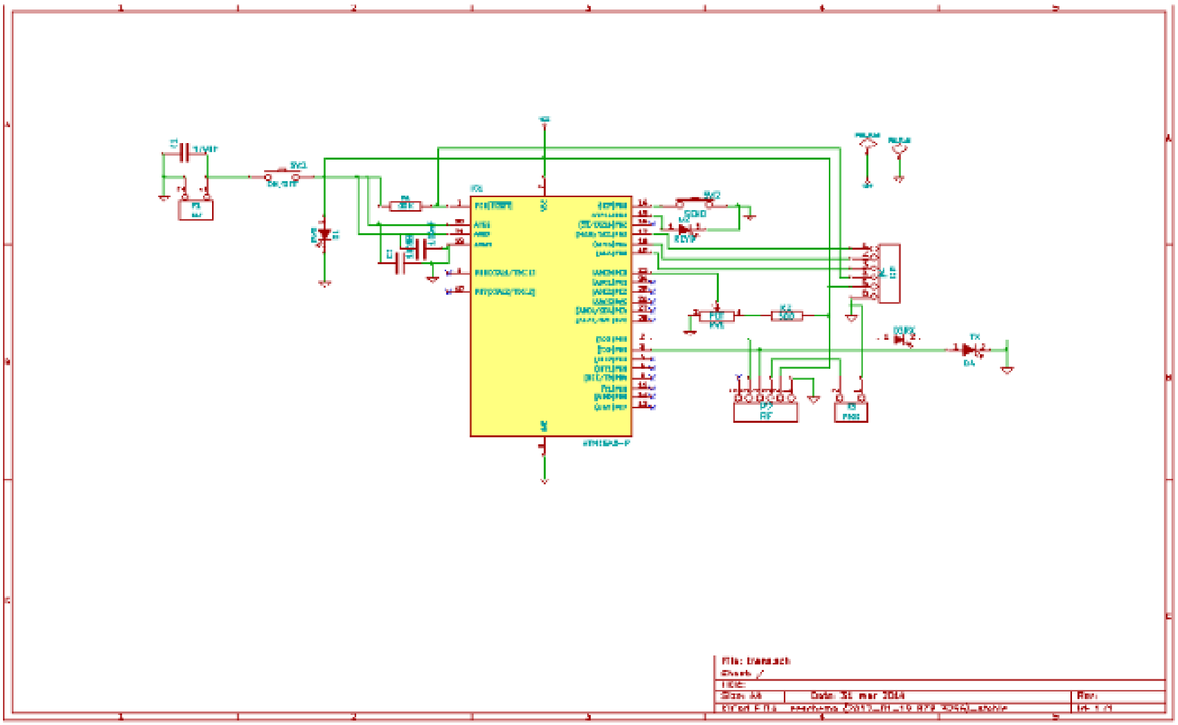


Figure 3.4 showing schematic diagram of transmitter

PCB Layout (using KiCAD software)

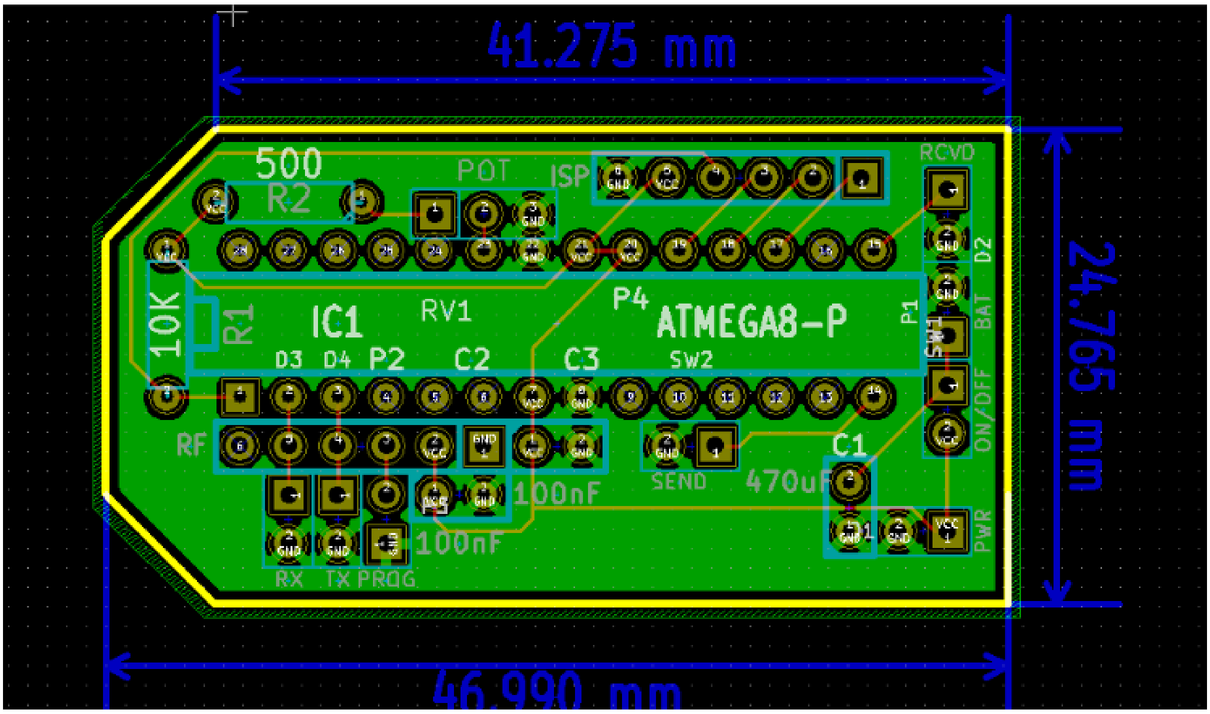


Figure 3.5 showing PCB Layout of Transmitter

CAD Model of circuit (using PCBnew)

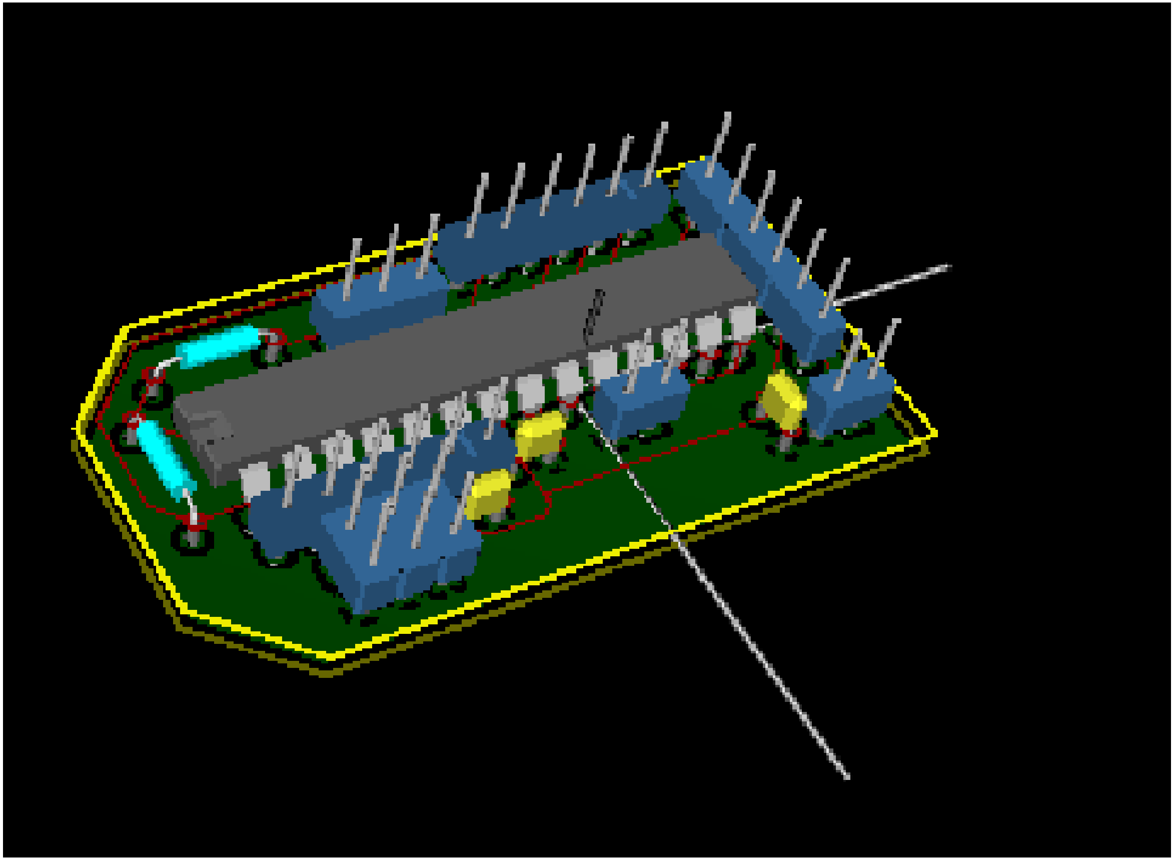


Figure 3.6 showing CAD Model of the circuit of transmitter

Receiver:

Schematic (using EEschema software)

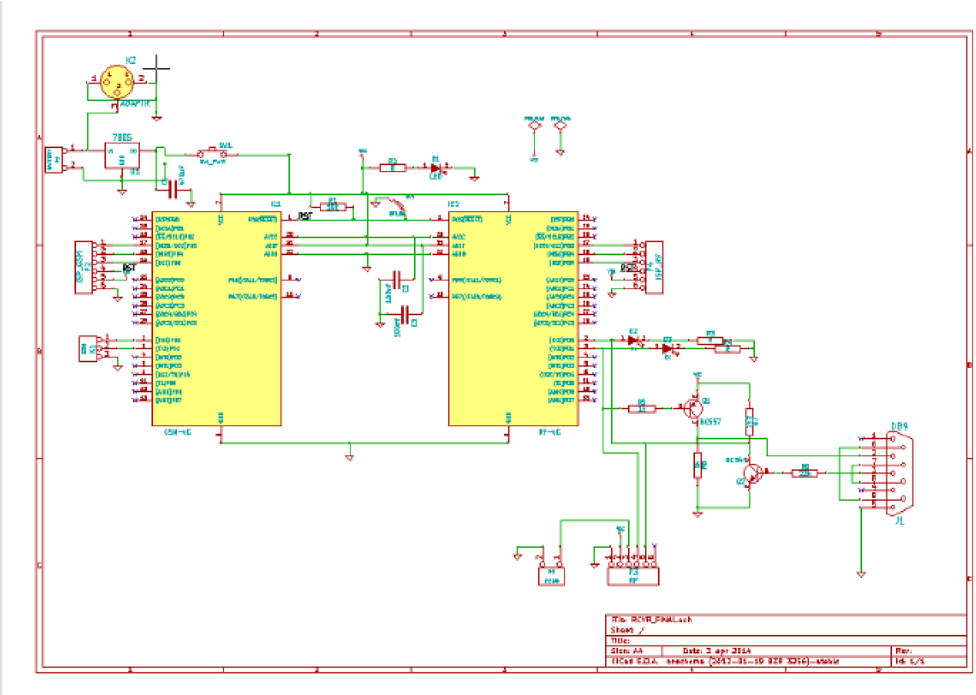


Figure 3.7 showing schematic diagram of receiver

PCB Layout

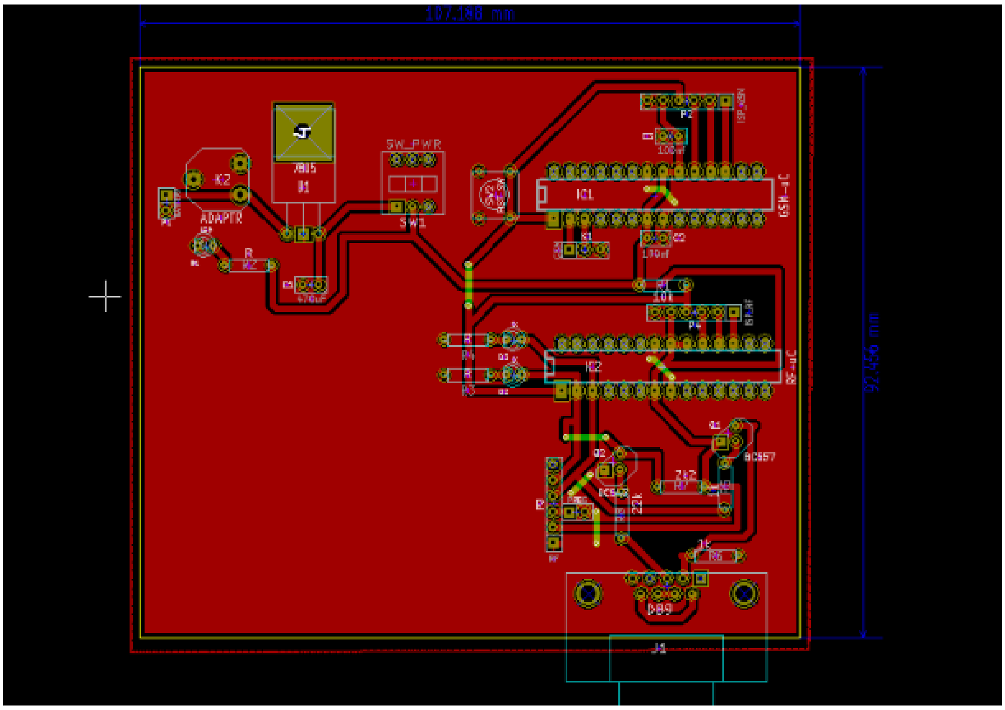


Figure 3.8 showing PCB Layout of receiver

CAD Model of circuit

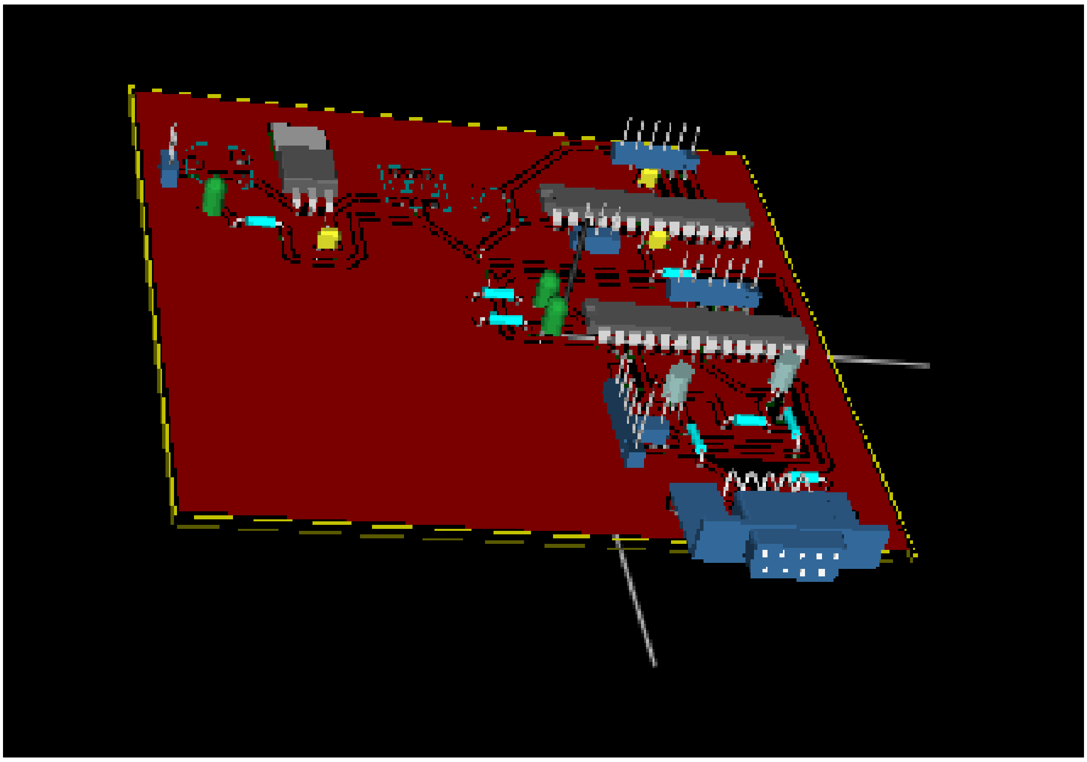


Figure 3.9 showing CAD Model of the circuit of receiver

3.4 MATERIAL SELECTION

3.4.1Transmitter covering

Criteria for material selection:

a) Light-weight
b) Robust
c) Water proof
d) Electrically insulated
e) Ease of attachment
f) Corrosion Resistant
g) Less price

Table 3.1

S.no.	MATERIALS	DENSITY	COST
1.	Polystyrene	1.05 g/cm³	65 Rs/Kg
2.	Neoprene	1.23 g/cm³	100 Rs/Kg
3.	Silicon	1.15 g/cm³	300 Rs/Kg
4.	Aluminium	2.7 g/cm³	115 Rs/kg
5.	Carbon fibre	1.78 g/cm³	1200 Rs/kg
6.	Acrylic	1.18 g/cm³	130 Rs/Kg

Table 3.2

PARAMETER	Polystyrene	Neoprene	Silicon	Carbon fibre	Aluminium	Acrylic
Light Weight	N	Y	Y	N	N	Y
Robustness	Y	Y	Y	Y	N	Y
Waterproof	Y	Y	Y	Y	Y	Y
Corrosion Resistant	Y	Y	Y	Y	Y	Y
Electrically resistant	Y	Y	Y	Y	N	Y
Ease of attachment	N	N	Y	Y	N	Y
Less price	N	Y	Y	Y	N	Y

Hence, according to the following comparison acrylic and neoprene are the most suitable options. For the required transmitter acrylic was devised for chassis and a layer of neoprene was formulated on the chassis for better grip robustness.

3.4.2 Material selection for Printed circuit board

PCB materials are selected by two key factors: how well they meet the needs of an end-use application and what kind of effort is required to fabricate a desired circuit with a particular material.

Typical circuit material types used in the high-frequency PCB industry.

Table 3.3

Material	Ease of circuit fabrication	Electrical performance
PTFE with micro glass fibre	Difficult	Excellent
PTFE with woven glass	Difficult	Good
Ceramic-filled PTFE	Moderate	Excellent
Ceramic-filled PTFE with woven glass	Moderate	Good
Ceramic-filled Hydrocarbon	Difficult	Good
Ceramic-filled Hydrocarbon with woven glass	Easy	Poor
High Performance FR-4	Easy	Good

PCB material FR-4 is easily available and used for high frequency. It is low cost feasible solution. So it is selected as base material for transmitter and receiver PCB.

3.5 COST:

Estimated cost:

Table 3.4

COMPONENTS	SPECIFICATIONS	QUANTITY	COST
Radio Transmitter	2.4GHz frequency	1	800
RF receiver	2.4GHz frequency	1	700
Antenna	Isotropic	2	100
MCU	ATMEGA 8PU	2	260
Preset Resistance	100K	1	10
Resistance	1k	2	10
Capacitance	470uF	6	24
Ceramic capacitor	22pF	4	10
GSM 900 Module	900 MHz frequency	1	1500
PCB	Single sided	1	200
Battery	5V	1	20
ACRYLIC sheet	5mm	50 x 50 mm^2	200
TOTAL COST			Rs 3834

Actual Cost:

COMPONENTS	SPECIFICATIONS	QUANTITY	COST
Radio Transmitter	900MHz frequency	1	600
RF receiver	950MHz frequency	1	600
MCU	ATMEGA 8PU	3	260
Preset Resistance	100K	1	10
Resistance	1k	10	10
Capacitance	470uF	1	24
Ceramic capacitor	100nF	4	10
GSM SIM300 Module	900 MHz frequency	1	1500
PCB	Single sided	1	200
Battery	AAA	4	20
Acrylic sheet	2mm	4*3 feet^2	700
Bolts & nuts	M3, 15 mm length (0.11”,0.59”)	15	30
Screw	M3, 30 mm length (0.11”, 1.2”)	10	10
TOTAL COST			Rs 3974

3.6 ERGONOMICS:

- People end product is water proof and easy to use with just one push button to send signal for location.
- Easily replaceable clock cell.
- Very small portable device and designed according to anthropometric data of human hand given as : Handgrip inside diameter 40-56 mm

Chapter 4

Fabrication Processes

4.1 Moulding:

For the formation of the transmitter through vacuum molding we need to create a die that acts as a base. For this project we have created mold from MDF. Wooden moulds are cheap and easy to fabricate, and have a longer life than plaster moulds. The thermoforming mould can be as simple as a wooden block. They are one of the most important parts of the thermoforming cycle. One of the main advantages of vacuum forming is that the pressures used are significantly less compared to the injection moulding process. The result is that vacuum formed tools can be produced economically and in a wide range of materials to suit different prototype and production requirements.

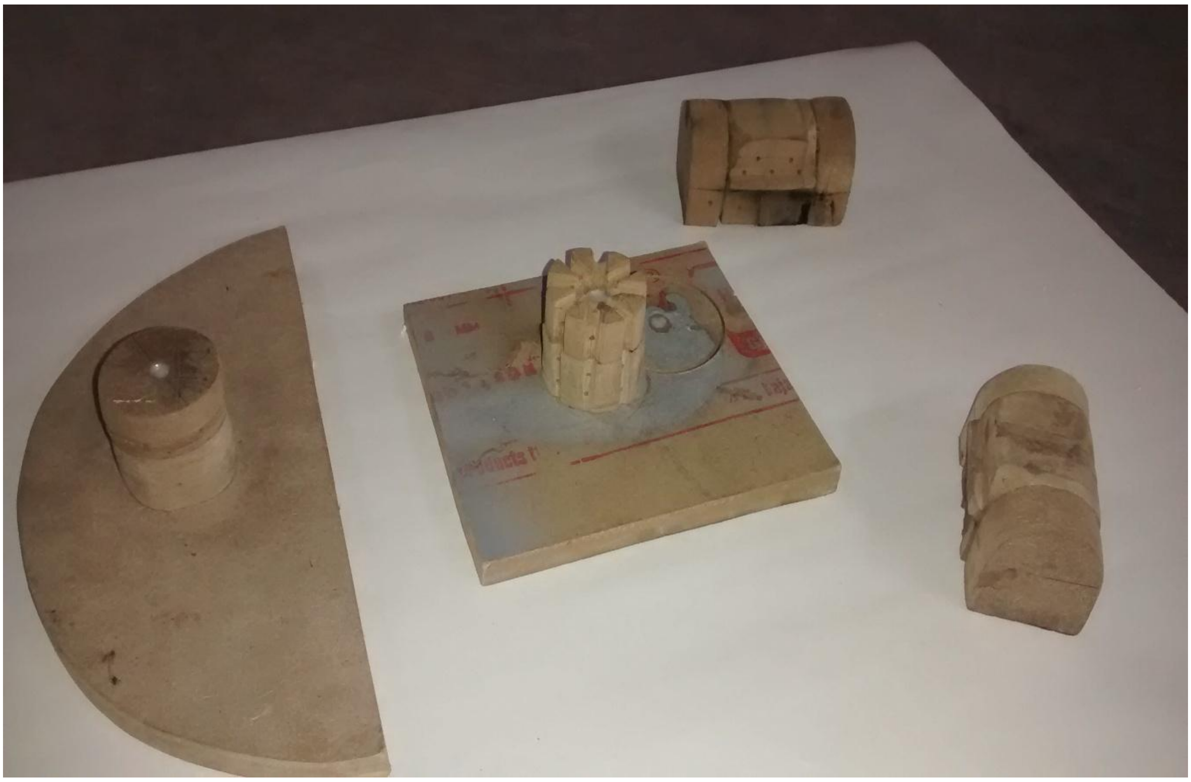


Figure 4.1 Side view of the mold created for transmitter

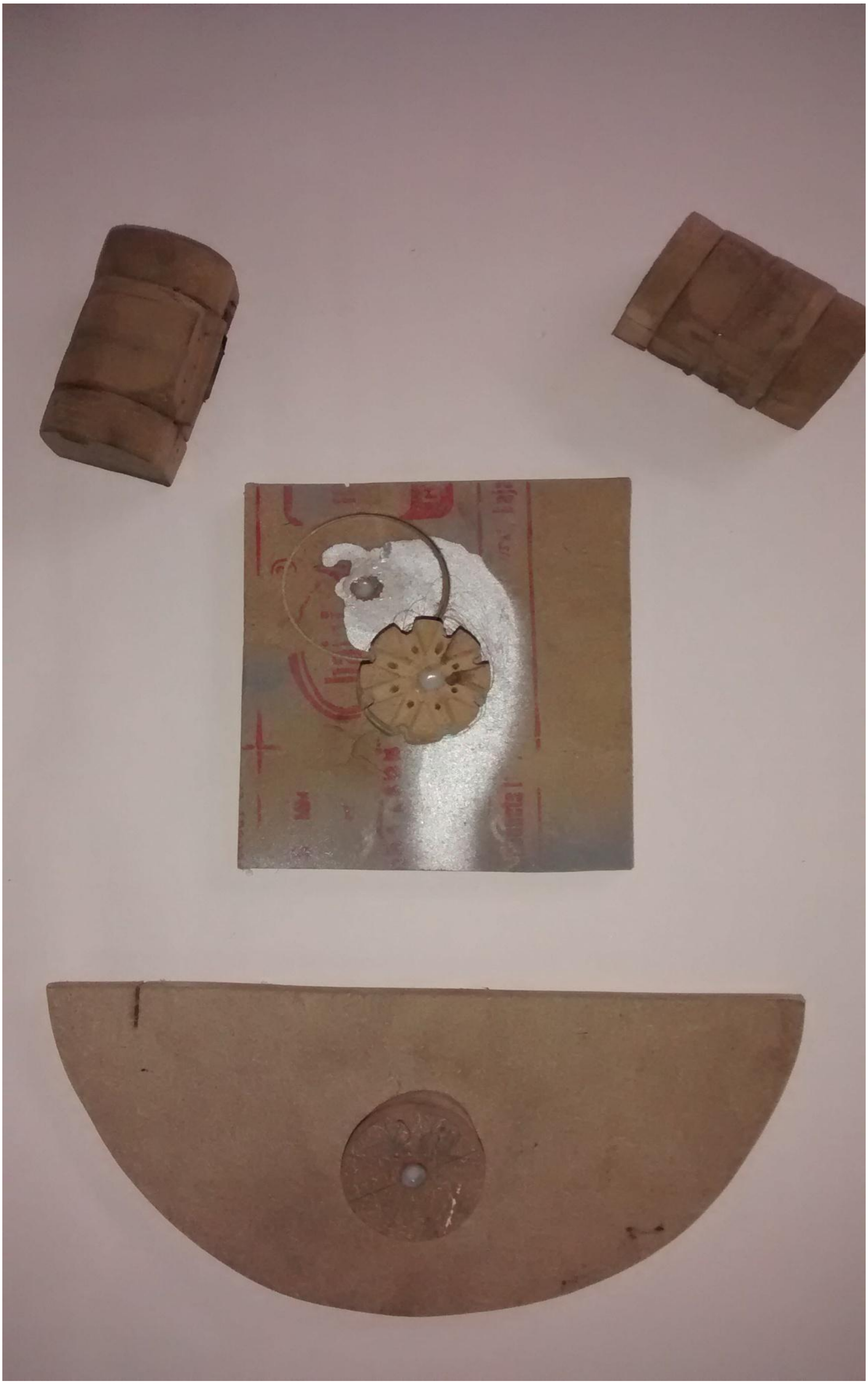


Figure 4.2 Top view of the mold created for transmitter

4.2 Vacuum forming:

The process consisted of inserting a acrylic sheet in a cold state into the forming clamp area, heating it to the desired temperature either with a surface heater and then raising a mould from below. The trapped air is evacuated with the assistance of a vacuum system and once cooled a reverse air supply is activated to release the plastic part from the mould.

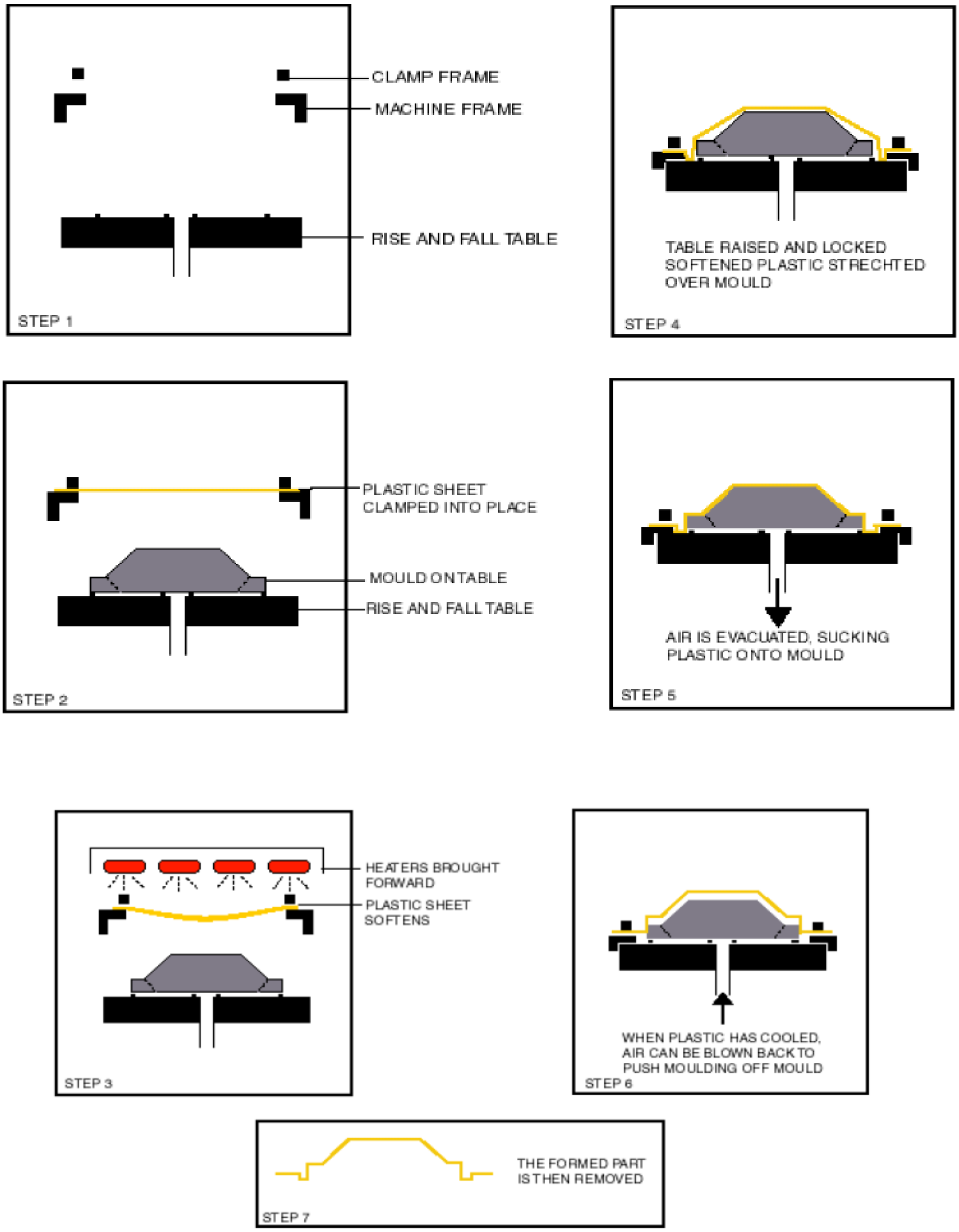


Figure 4.3 Vacuum forming process

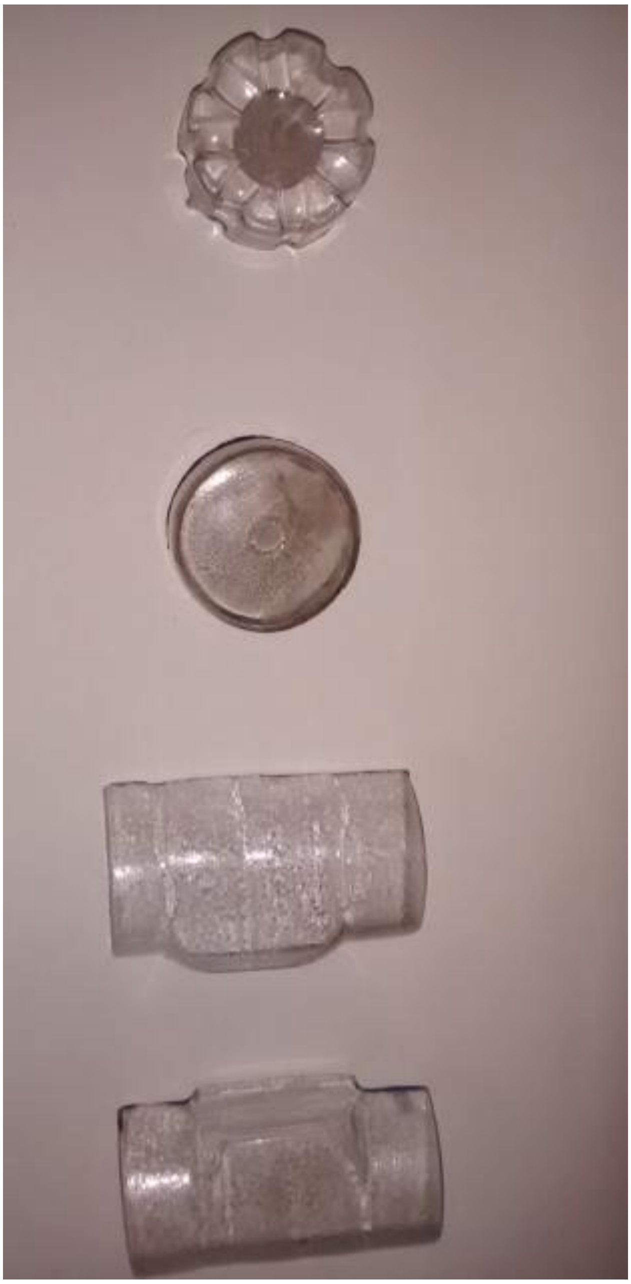


Figure 4.4 Parts formed from vacuum forming

Parts of transmitter formed from vacuum molding

4.3 Manual PCB formation:

- Create the PCB Layout of your circuit. Converting circuit's schematic diagram into a PCB layout using PCB layout software.
- Draw the circuit layout on the copper coated board.
- Etch the board. This process removes any unnecessary copper from the board leaving only wiring of the final circuit.
- Drill mount points. Drilling machine is used for carving out points.
- Mount and solder the electronic components on board.

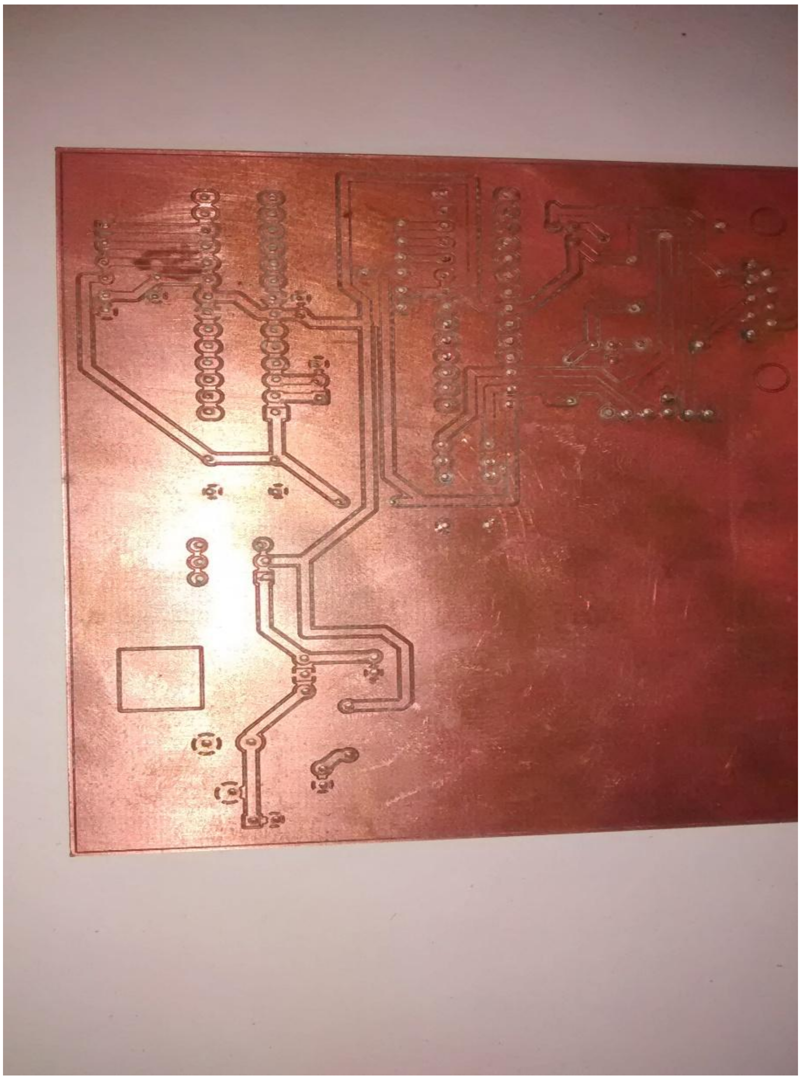


Figure 4.5 Manually formed PCB

Chapter 5
Final parts

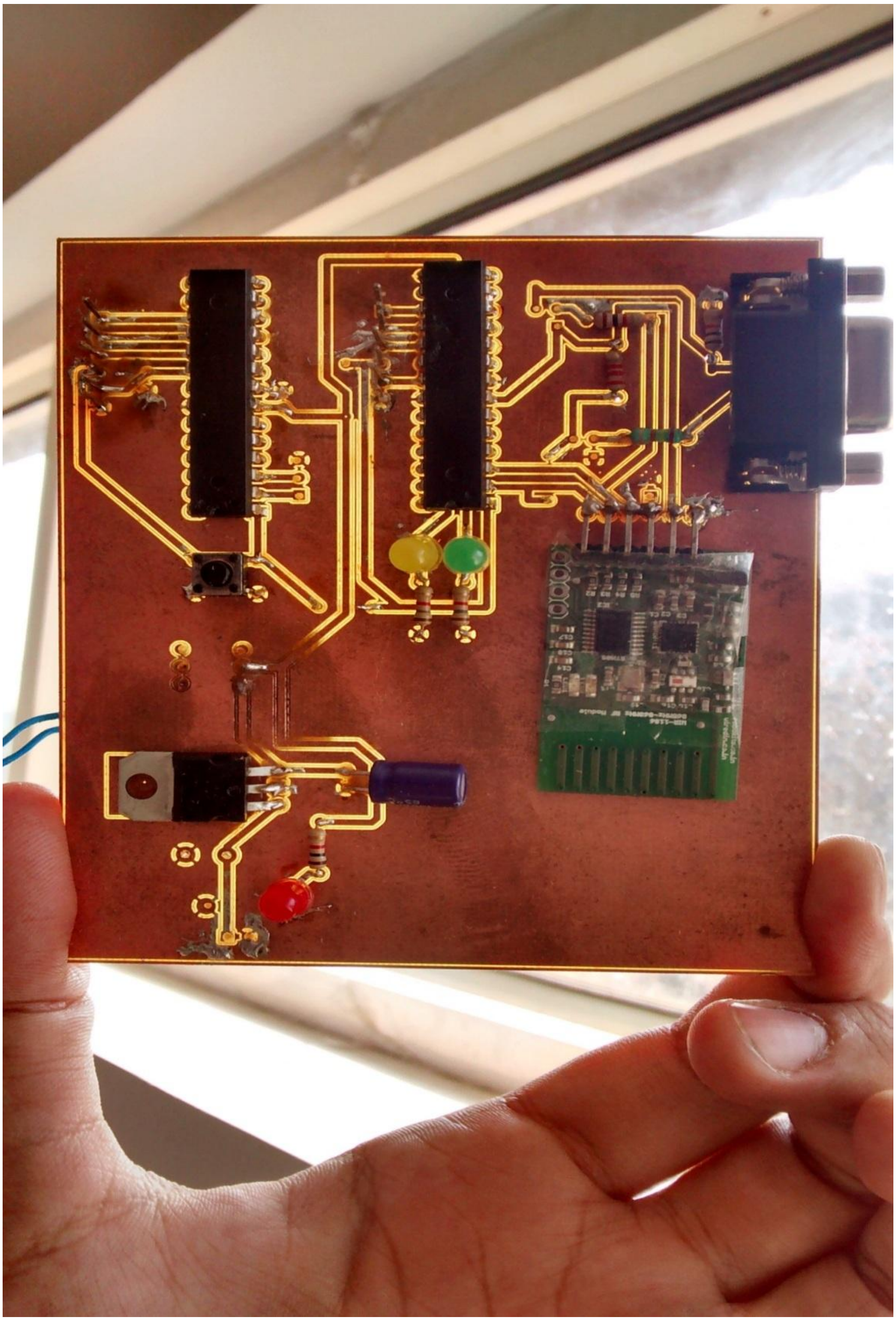


Figure 5.1 PCB of GSM and RF of receiver

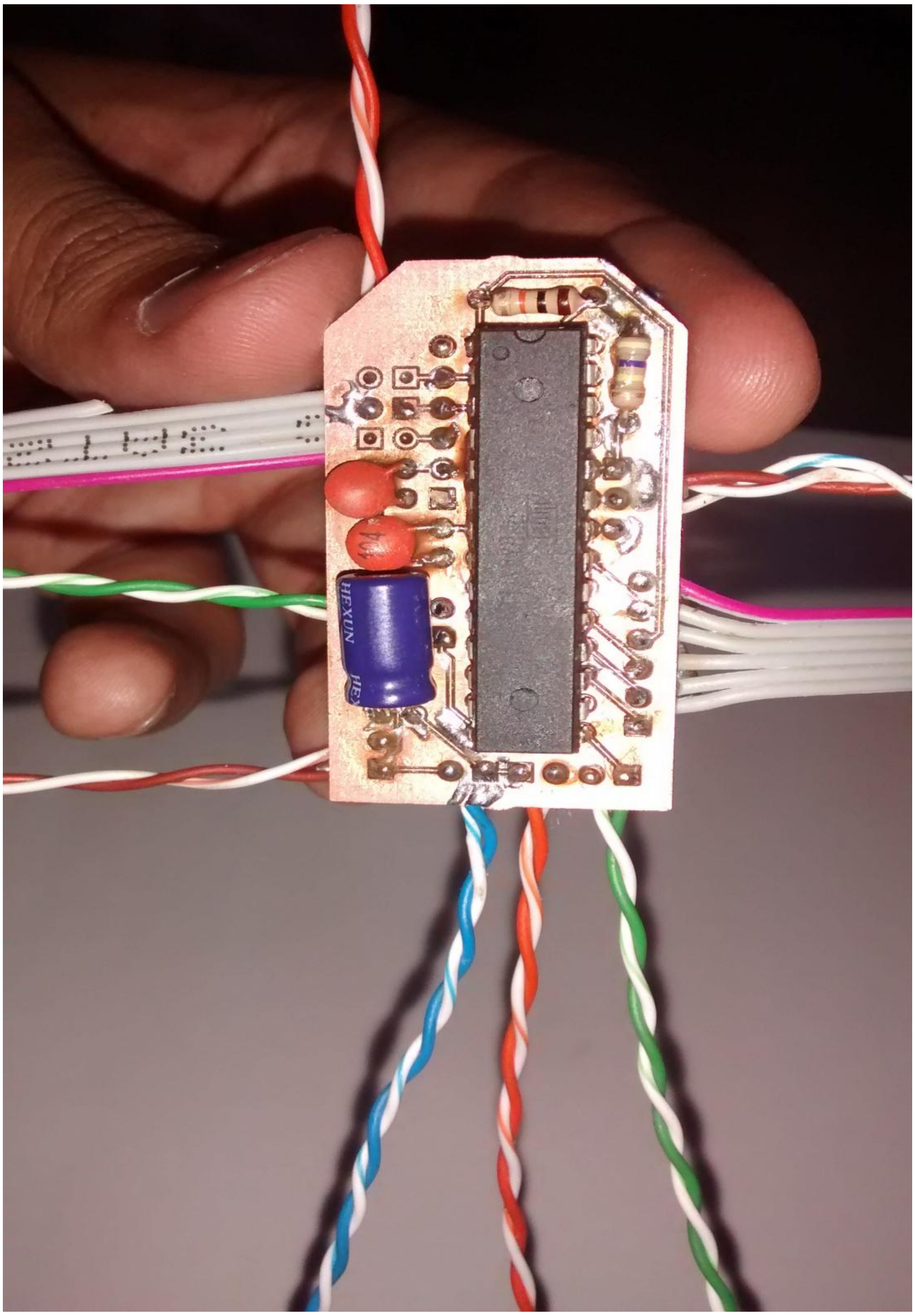


Figure 5.2 Transmitter control unit

Chapter 6
PCB coding:

Coding for control units of RF transmitter, RF receiver and GSM is done via CVAVR. This software is used for the code generation of AVR. The coding used is as follows:

6.1 GSM code:

```
#include <mega8.h>

#include <stdio.h>

void usart_init(void) { UCSRB = (1<<TXEN)|(1<<RXEN);

UCSRC = (1<<UCSZ1)|(1<<UCSZ0)|(1<<URSEL);

UBRRL = 0X33; }

void usart_sendchar(unsigned char ch)

{ while(!(UCSRA & (1<<UDRE)));

UDR = ch;}

void usart_sendstring(unsigned char *str)

{ unsigned char i=0;

while(str[i]!= '\0') {   while(!(UCSRA & (1<<UDRE)));

UDR = str[i];

i++;  }

i=0; }

 char usart_readchar() {   while(!(UCSRA & (1<<RXC)));

 return UDR; }

void sendmsg(unsigned char *msg)

{ char cmd[25];

  unsigned char num1[] = {"\"9424743297\""};

  unsigned char num2[] = {"\"8989828523\""};

  unsigned char num3[] = {"\"8400137397\""};

  unsigned char num4[] = {"\"9479845436\""};

  unsigned char num5[] = {"\"9415557019\""};

  usart_init();

        sprintf(cmd,"AT+CMGS=%s",num1);

        usart_sendstring(cmd);

        usart_sendchar(0x1A);

        usart_sendstring(msg);

        usart_sendchar(0x1A);

        sprintf(cmd,"AT+CMGS=%s",num2);

        usart_sendstring(cmd);

        usart_sendchar(0x1A);
        usart_sendstring(msg);

        usart_sendchar(0x1A);

        sprintf(cmd,"AT+CMGS=%s",num3);

        usart_sendstring(cmd);

        usart_sendchar(0x1A);

        usart_sendstring(msg);

        usart_sendchar(0x1A);

        sprintf(cmd,"AT+CMGS=%s",num4);

        usart_sendstring(cmd);

        usart_sendchar(0x1A);

        usart_sendstring(msg);

        usart_sendchar(0x1A);

        sprintf(cmd,"AT+CMGS=%s",num5);

        usart_sendstring(cmd);

        usart_sendchar(0x1A);

        usart_sendstring(msg);

        usart_sendchar(0x1A);

  }

void main(void)

{

        char cmd[25];

        int send = 0;

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
(0<<PORTB0);

DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);

PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) |
(0<<PORTD0);

TCCR0=(0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;
```

```
OCR1BH=0x00;

OCR1BL=0x00;

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<TOIE0);

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);

UBRRH=0x00;

UBRRL=0x33;

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADFR) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

 while(1)  {send = PORTC && (0b10111111);

   if(send == 0b11111111)   {           sendmsg("\rFLOOD LEVEL INCREASING.PACK YOUR IMPORTANT ITEMS\r");

        send = 0b00000000; }

   if(send == 0b11111111)   {    sendmsg("\rFLOOD LEVEL HIGH\r");

      send = 0b00000000; }

    if(send == 0b11111111)    {    sendmsg("\rFLOOD APPROACHING IN THIS AREA DUE TO RISING WATER LEVEL IN THE YAMUNA RIVER\r"); send = 0b00000000; }

    if(send == 0b11111111) {sendmsg("\rALARMING FLOOD LEVEL.RUN OUT OF YOUR HOUSE AS SOON AS POSSIBLE\r");

     send=0b00000000;  }    }  }
```

6.2 RF Receiver code:

```
#include <mega8.h>
#include <delay.h>
#include <stdio.h>
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))
#ifndef F_CPU
#define F_CPU 4000000
#endif
#define BAUDRATE 1200
#define UBRRVAL ((F_CPU/(BAUDRATE*16UL))-1)
void inituart(void){
   UBRRL = UBRRVAL;
   UBRRH = (UBRRVAL>>8);
   UCSRC|=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);
    UCSRB=(1<<RXEN)|(1<<TXEN);}
void writechar(char ch)
{   while(!(UCSRA & (1<<UDRE)));
```

```
  UDR=ch;}
void writestring(unsigned char *str)
{ unsigned char i=0;
while(str[i]!= '\0')
{   while(!(UCSRA & (1<<UDRE)));
UDR = str[i];
i++;  }
i=0;}
unsigned char readchar()
{   while(!(UCSRA & (1<<RXC)));
  return UDR;}
unsigned char read_adc(unsigned char adc_input)
{ ADMUX=adc_input | ADC_VREF_TYPE;
delay_us(10);
ADCSRA|=(1<<ADSC);
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF);
return ADCH;}
void main(void)
{    char ch;
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
(0<<PORTB0);
DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
(0<<PORTC0);
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) |
(0<<PORTD1) | (0<<PORTD0);
TCCR0=(0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<TOIE0);
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) |
(0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADFR) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ACME);
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
while (1)      {
      inituart();
      ch = readchar();
      delay_us(100);
      writechar(ch);
      delay_us(100);
      if(ch =='a'){  writestring("\rThere is an urgent need of rescue team at \"SADAR\"\r");
        delay_ms(10);}
      if(ch =='b')        {  writestring("\rThere is an urgent need of rescue team at \"BUS-STAND\"\r");
        delay_ms(10); }
      if(ch =='c')
{  writestring("\rThere is an urgent need of rescue team at \"SAMDARIYA\"\r");
   delay_ms(10); }
 if(ch =='d'){  writestring("\rThere is an urgent need of rescue team at \"SOUTH AVENUE MALL\"\r");
   delay_ms(10); }
if(ch =='e')
{  writestring("\rThere is an urgent need of rescue team at \"RUSSEL CHOWK\"\r");
   delay_ms(10); }
if(ch =='f')
{  writestring("\rThere is an urgent need of rescue team at \"GWARI-GHAT\"\r");
   delay_ms(10);}
if(ch =='g')        {  writestring("\rThere is an urgent need of rescue team at \"BHEDAGHAT\"\r");
   delay_ms(10);        }

if(ch =='h')        {  writestring("\rThere is an urgent need of rescue team at \"DUMNA ROAD\"\r");
   delay_ms(10);        }   }}
```

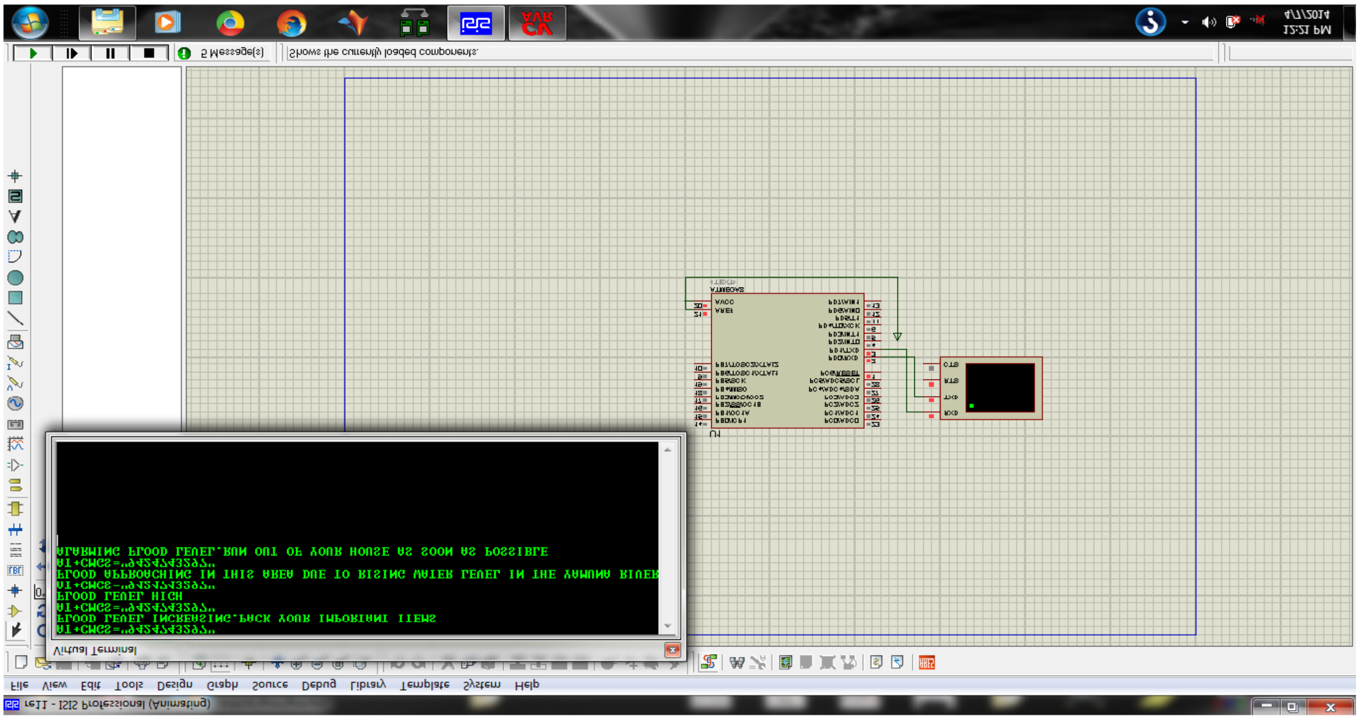
6.3 RF transmitter code:

```
#include <mega8.h>
#include <delay.h>
#include <stdio.h>
#ifndef F_CPU
#define F_CPU 8000000
#endif
#define BAUDRATE 9600
#define UBRRVAL ((F_CPU/(BAUDRATE*16UL))-1)
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))
void usart_init(void){ UCSRB = (1<<TXEN)|(1<<RXEN);
UCSRC = (1<<UCSZ1)|(1<<UCSZ0)|(1<<URSEL);
UBRRL = 0x33;
UBRRH = 0; }
void usart_sendchar(unsigned char ch){while(!(UCSRA & (1<<UDRE)));
UDR = ch;}
void usart_sendstring(unsigned char *str){unsigned char i=0;
while(str[i]!= '\0') {   while(!(UCSRA & (1<<UDRE)));
UDR = str[i];
i++;  }
i=0; }
unsigned char read_adc(unsigned char adc_input){
   ADMUX=adc_input | ADC_VREF_TYPE;
   // Delay needed for the stabilization of the ADC input voltage
   delay_us(10);
   // Start the AD conversion
   ADCSRA|=(1<<ADSC);
   // Wait for the AD conversion to complete
   while ((ADCSRA & (1<<ADIF))==0);
   ADCSRA|=(1<<ADIF);
   return ADCH;  }
void main(void)
{DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
(0<<PORTB0);
DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
(0<<PORTC0);
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) |
(0<<PORTD1) | (0<<PORTD0);
TCCR0=(0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) |
(0<<TOIE0);
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) |
(0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADFR) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ACME);
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
usart_init();
while (1) {         char ad_in;
      char ad_out;
      char reg;
      char send;
      send = PORTB && (0b00000001);
      ad_in = 0;
      ad_out = read_adc(ad_in);
      delay_ms(1);
      if((0 < ad_out) && (ad_out < 108)) //90 {   reg = 'a';
           if(send == 0b00000001) {              {
              usart_sendchar(reg);
              delay_ms(10);  }
            else if(send== 0b00000000)
            break;   }            }
      if((107 < ad_out ) && (ad_out < 126))        {   reg = 'b';
           if(send == 0b00000001)        {
              usart_sendchar(reg);
              delay_ms(10);  d  }                   }           }
      if((125 < ad_out) && (ad_out  < 144))     {   reg = 'c';
           if(send == 0b00000001){         {   usart_sendchar(reg);
              delay_ms(10);   }              }
        if((143 < ad_out) && (ad_out  < 162))                          {   reg = 'd';
           if(send == 0b00000001)                          {
              usart_sendchar(reg);
              delay_ms(10);          }               }
      if((161 < ad_out) && (ad_out  < 180))               {   reg = 'e';
           if(send == 0b00000001)                          {
              usart_sendchar(reg);
              delay_ms(10);          }               }
      if((179 < ad_out) && (ad_out  < 198))               {   reg = 'f';
           if(send == 0b00000001)                          {              usart_sendchar(reg);
              delay_ms(10);     }    }
      if((197 < ad_out) && (ad_out  < 216))  {   reg = 'g';
           if(send == 0b00000001)                          {              usart_sendchar(reg);
              delay_ms(10);          }               }
      if((215 < ad_out) && (ad_out  <= 255))                        {   reg = 'h';
           if(send == 0b00000001)                          {usart_sendchar(reg);
              delay_ms(10);
              }}}}
```

Chapter 7

Simulation

7.1 GSM simulation

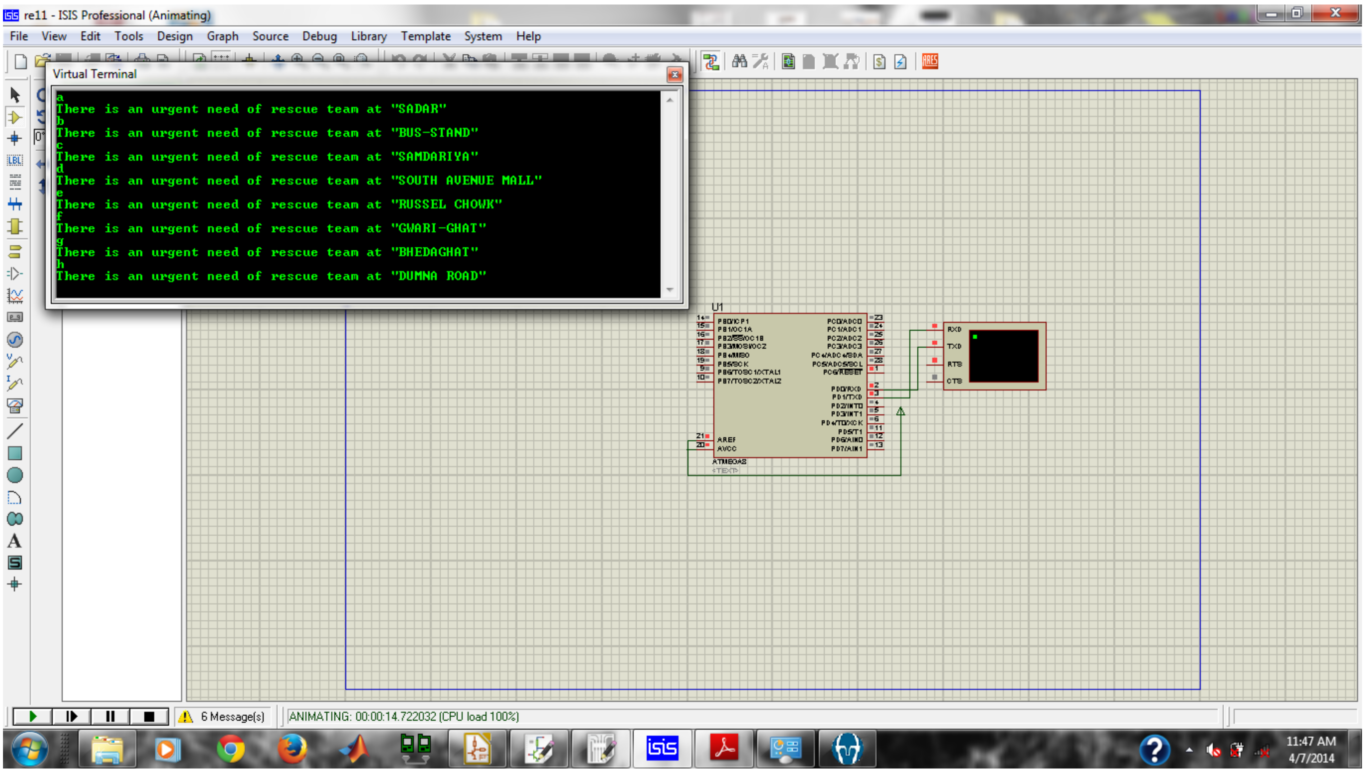


```
AT+CMGS="9424743297"
FLOOD LEVEL INCREASING.PACK YOUR IMPORTANT ITEMS
AT+CMGS="9424743297"
FLOOD LEVEL HIGH
AT+CMGS="9424743297"
FLOOD APPROACHING IN THIS AREA DUE TO RISING WATER LEVEL IN THE YAMUNA RIVER
AT+CMGS="9424743297"
ALARMING FLOOD LEVEL.RUN OUT OF YOUR HOUSE AS SOON AS POSSIBLE
```

rel11 - ISIS Professional (Animating)
File View Edit Tools Design Graph Source Debug Library Template System Help
Virtual Terminal
5 Message(s)
Shows the currently loaded components.

Figure 7.1 GSM simulator

7.2 Receiver simulation


re11 - ISIS Professional (Animating)
File View Edit Tools Design Graph Source Debug Library Template System Help
Virtual Terminal
U1
ATMEGA8
RXD
TXD
RTS
CTS
6 Message(s)
11:47 AM
4/7/2014

```
a
There is an urgent need of rescue team at "SADAR"
b
There is an urgent need of rescue team at "BUS-STAND"
c
There is an urgent need of rescue team at "SAMDARIYA"
d
There is an urgent need of rescue team at "SOUTH AVENUE MALL"
e
There is an urgent need of rescue team at "RUSSEL CHOWK"
f
There is an urgent need of rescue team at "GWARI-GHAT"
g
There is an urgent need of rescue team at "BHEDAGHAT"
h
There is an urgent need of rescue team at "DUMNA ROAD"
```

ANIMATING: 00:00:14.722032 (CPU load 100%)

Figure 7.2 Receiver simulator

Chapter 8

Implementation of this project in real life

This project is constructed keeping in sight its use in real life. Mass production of transmitter can significantly reduce its cost making it cheap to such an extent that it can distributed by government authorities in the flood prone areas. This product will not only help in the flood rescue but also can act as a modern technique to connect people with the concerned authorities. Design of the transmitter is such that it can be easily inculcated into the life of the people in flood prone areas.

RESULTS AND CONCLUSIONS

This product is designed such that it can cater the general public which is not aware of the scenario and need assistance in the time of flood. It does not require any bulky technical knowledge prior to use. Area code knob and frequency generation for signal transmission is done via the help of transmitter. This step generates a wave which has encrypted data for area identification. Receiver receives the data and decrypts the information for area prediction. Processing of this information by disaster management organization. Measures taken to ensure the prevention of further harm to the informed area.

This system will help the management team to identify the victims and affected area. The principal problem of large loss of life and resources caused by the floods is caused due to lack of information.

REFERENCES

1) “High frequency PCB material selection” Microwave Engineering Journal Europe, March-April 2012

2) www.ndrf.org

3) www.ndma.gov.in

4) www.undp.org.in

5) http://en.wikipedia.org/wiki/2013_North_India_floods

6) United Nations Disaster Management Team (UNDMT) and Pragya, which is one of the NGO partners of United Way of India.