















```

WriteSMS(police_Num);
delay(4000);
fona.callPhone(Call_Num);
lcd.setCursor(0, 0);
lcd.print("SMS Sent!");
RedBtn=0;
RedBtnbuttonState = 1;
lcd.clear();
}
if(RedBtnbuttonState == 1 && time > 120000){
timeMinus += time;
lcd.setCursor(0, 0);
lcd.print("Calling & SMS..");
WriteSMS(SMS_Num_1);
delay(5000);
// WriteSMS(police_Num);
// delay(4000);
fona.callPhone(Call_Num);
lcd.setCursor(0, 0);
lcd.print("SMS Sent!");
lcd.clear();
}
}
void GetGPS(){
// if you ask for an altitude reading, getGPS will return false if
there isn't a 3D fix
boolean gps_success = fona.getGPS(&latitude, &longitude, &speed_kph, &heading, &altitude);
if (gps_success) {
lcd.setCursor(0, 0);
lcd.print("GPS OK!");
// Serial.print("GPS lat:");
// Serial.println(latitude, 6);
lcd.setCursor(0, 1);
lcd.print("N:");
lcd.setCursor(2, 1);
lcd.print(latitude,3);
// Serial.print("GPS long:");
// Serial.println(longitude, 6);
lcd.setCursor(8, 1);
lcd.print("E:");
lcd.setCursor(10, 1);
lcd.print(longitude,3);
// Serial.print("GPS speed KPH:");
// Serial.println(speed_kph);
}
}

void WriteSMS (char num[20])
char tempSMS[140];

snprintf(tempSMS,140, "I'm in trouble, Map link :
https://www.google.com.bd/maps/place/%s,%s",f2s(latitude,6),
f2s(longitude,6));
Serial.println(tempSMS);
Serial.println("Sending reponse...");
if (!fona.sendSMS(num, tempSMS)) {
Serial.println(F("Failed"));
} else {
Serial.println(F("Sent!"));
}
}

char *f2s(float f, int p){
char * pBuff; // use to remember which part of
the buffer to use to dostrf
const int iSize = 20; // number of buffers, one for
each float element wrapping around
static char sBuff[iSize][20] // space for 20 characters
including NULL terminator for each float
static int iCount = 0; // keep a tab of next place in
sBuff to use
pBuff = sBuff[iCount]; // use this buffer
if(iCount >= iSize -1){ // check for wrap
iCount = 0; // if wrapping start again and re-
set
}
else{
iCount++; // advance the counter
}
return dtostrf(f, 0, p, pBuff); // call the library function
}

```

IJOART