

**ABSTRACT:**

A vehicle tracking system combines the installation of an electronic device in a vehicle, or fleet of vehicles, with purpose-designed computer software to enable the owner or a third party to track the vehicle's location, collecting data in the process. Modern vehicle tracking systems commonly use Global Positioning System (GPS) technology for locating the vehicle, but other types of automatic vehicle location technology can also be used. Vehicle information can be viewed on electronic maps via the Internet or specialized software. In the main they are easy to steal, and the average motorist has very little knowledge of what it is all about. To avoid this kind of steal we are going to implement a system it provides more security to the vehicle. Existing System: In the previous system security lock and alarm is implemented in a car. If a burglar can break open the lock, then it becomes easy for the burglar to steal the car. And in old security system if the car is stolen then it is out of the owner control. User doesn’t have any awareness about the current location of the vehicle. The Proposed System: The RF transmitter is attached with the vehicle which has its own identification. This data will be continuously transmitted to the RF receiver connected to the microcontroller. This GPS will be location the position of vehicle and transmit that data to the microcontroller. Suppose the RF

receiver not receiving signal from the transmitting unit, receiver unit send the signal to the microcontroller, from that we can identify the theft. If the vehicle is theft it automatically sends location of the vehicle to its owner as a SMS through GSM modem. This will be a much simpler and low cost technique compared to others. If a password like SMS is sent by the owner, it automatically stops the vehicle Keywords: Global Positioning System (GPS), RF receiver and transmitter, operations and maintenance center (OMC) and Gaussian minimum shift keying (GMSK). A vehicle tracking system combines the installation of an electronic device in a vehicle, or fleet of vehicles, with purposedesigned computer software at least at one operational base to enable the owner or a third party to track the vehicle's location, collecting data in the process from the field and deliver it to the base of operation. Modern vehicle tracking systems commonly use GPS or GLONASS technology for locating the vehicle, but other types of automatic vehicle location technology can also be used. Vehicle information can be viewed on electronic maps via the Internet or specialized software. Vehicle tracking systems are also popular in consumer vehicles as a theft prevention and retrieval device. Police can simply follow the signal emitted by the tracking system and locate the stolen vehicle. When used as a security system, a Vehicle Tracking System may serve as either an addition to or replacement for a traditional Car alarm. Some vehicle tracking systems make it possible to control vehicle remotely, including block doors or engine in case of emergency. The existence of vehicle tracking device then can be used to reduce the insurance cost.

**Working**

The main aim of the project **Accident Detection and Messaging System** is to inform the Ambulance and Police of the accident site and arrange for necessary steps to control the situation. This system is not only efficient but also worthy to be implemented. . The Accident Detection and Messaging System can be fitted in the vehicle (Ambulance or the Police) and they are informed about any such untoward incident at the go. Accident Detection and Messaging System execution is simple as the system makes use of [*GSM*](http://www.engineersgarage.com/articles/gsm-gprs-modules)and GPS technologies. GPS is used for taking the coordinates of the site of the accident while GSM is used for sending the coordinates to cell phones. To make this process all the controls are made using Arduino whereas a LCD is used to display the coordinates.

**Component Explanation**

|  |  |
| --- | --- |
| * 269,219-Reads | [**LCD**](http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet)    **LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are...** |
| * 32,829-Reads | [**LED**](http://www.engineersgarage.com/electronic-components/leds-light-emitting-diode)    **Light emitting diodes.** |
| * 25,957-Reads | [**Resistor**](http://www.engineersgarage.com/electronic-components/resistors)  **Resistor is a passive component used to control current in a circuit. Its resistance is given by the ratio of voltage applied across its terminals to the current passing through it. Thus a particular value of resistor, for fixed voltage, limits the current through it. They are omnipresent in...** |
| * 17,284-Reads | [**Capacitor**](http://www.engineersgarage.com/electronic-components/capacitors)  **Capacitor is a passive component used to store charge. The charge (q) stored in a capacitor is the product of its capacitance (C) value and the voltage (V) applied to it. Capacitors offer infinite reactance to zero frequency so they are used for blocking DC components or bypassing the AC signals. The capacitor...**  **GSM** |

SIM 900 a GSM Module which we are going to use to receive the text message. This module needs 9V DC power supply so we are using adapter for that. Here we are using only two pins of this module to connect with Arduino or our ATmega 328/168/8 board.   
Those are GND to common all grounds and make path complete for data   
TX (TTL Transmitter pin) to receive signal to Microcontroller from GSM Module



**GPS**

The GPS is a system of positioning by satellite to give an accurate position anywhere on the planet within a hundred meters, by day or night. The visible part is a small electronic case, which indicates precisely -and in a split second- the exact place, height, speed and time.

**About GPS**

The Global Positioning System (GPS) is a network of 24 Navistar satellite orbiting Earth at 11,000 miles. Originally established by the U.S. Department of Defence (DOD) at a cost of about US$13 billion, access to GPS is free to all users, including those in other countries. The system’s positioning and timing data are used for a variety of applications, including air, land and sea navigation, vehicle and vessel tracking, surveying and mapping, and asset and natural resource management. With military accuracy restrictions partially lifted in March 1996 and fully lifted in May 2000, GPS can now pinpoint the location of objects as small as a penny anywhere on the earth’s surface.

**Uses of GPS technology**

GPS technology has matured into a resource that goes far beyond its original design goals. These days people from a plethora of professions are using GPS in ways that make their work more productive, safer, and sometimes even easier. There are five main uses of GPS today:

* Location- determining a basic position.
* Navigation - getting from one location to another.
* Tracking - monitoring the movement of people and things.
* Mapping- creating maps.
* Timing - providing precise timing.

**Scope**

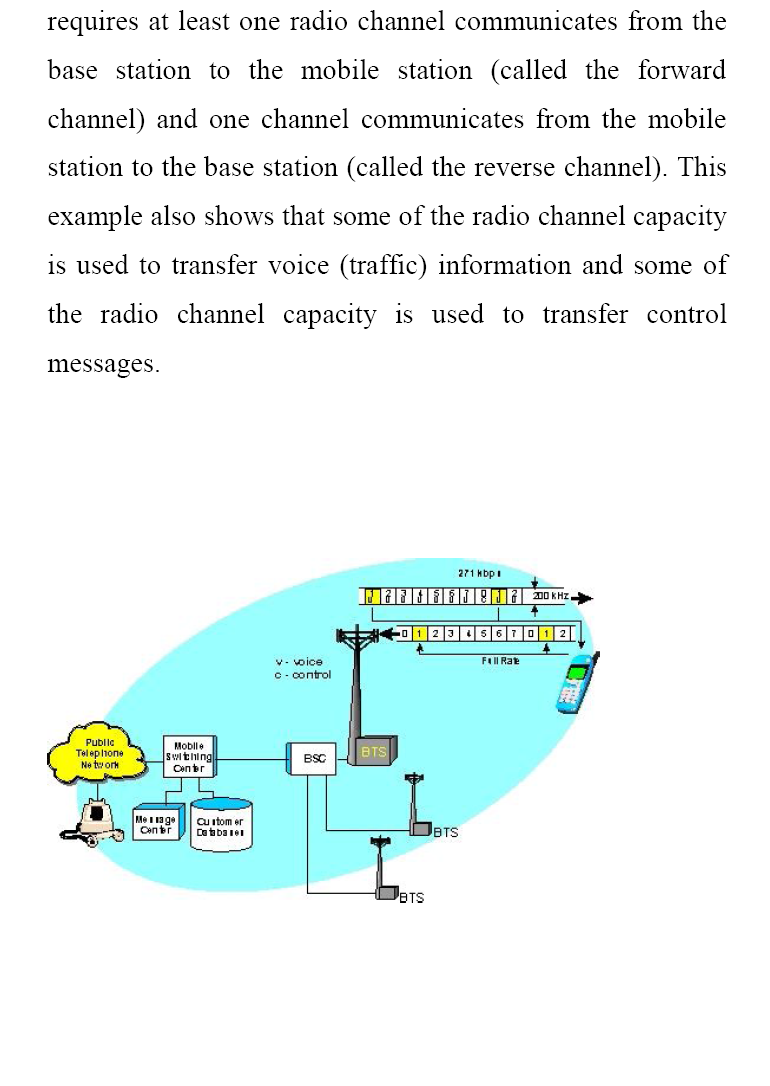
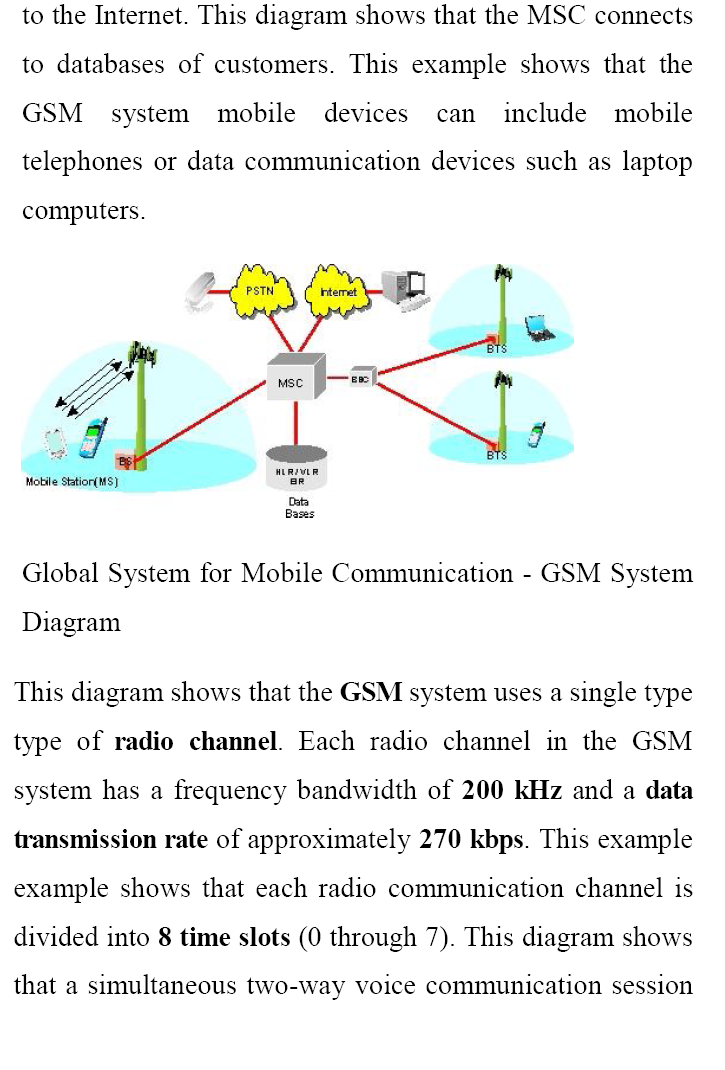
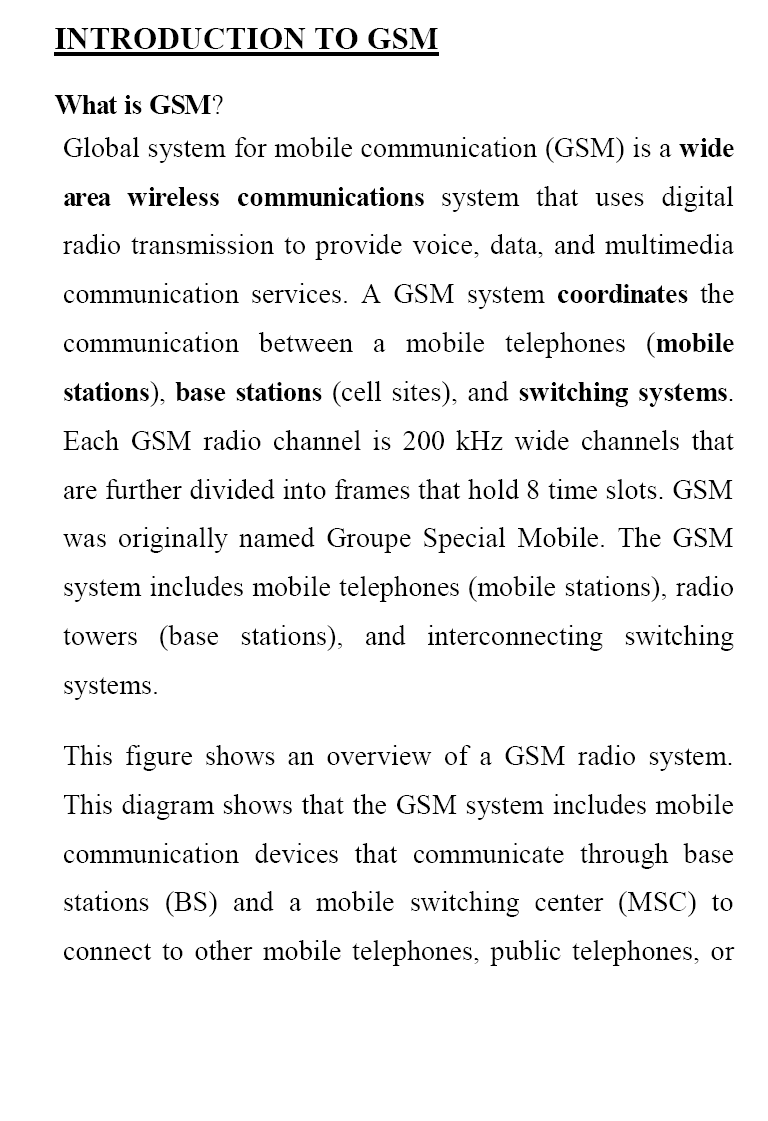
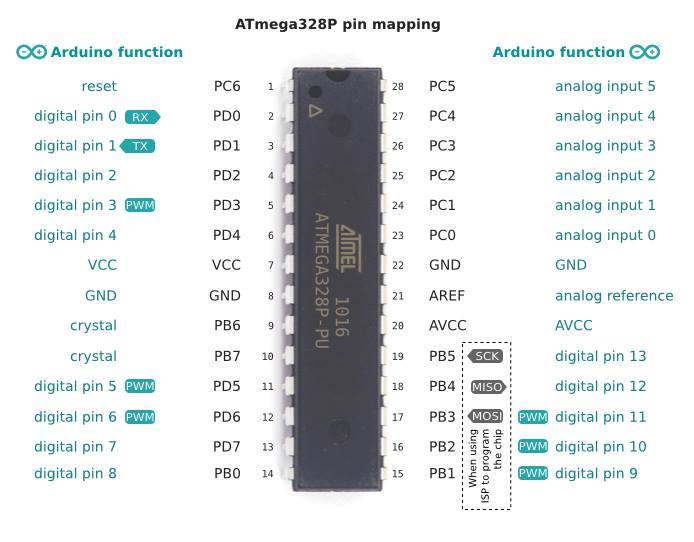
The aims of this project is to design and implement a GPS based Land Surveying system, and find out time, latitude, and longitude of the boundary points and thereby take the survey of the land.

The GPS Land Surveying System includes 24 satellites that provide location awareness by transmitting longitude, latitude, altitude, and time information to GPS receiving and processing devices worldwide. Here the system will track the position of boundary points of a particular land.The micro controller circuitry receive and process the data coming from the GPS receiver. The micro controller will send the needed information to plot the graph of the land to the PC

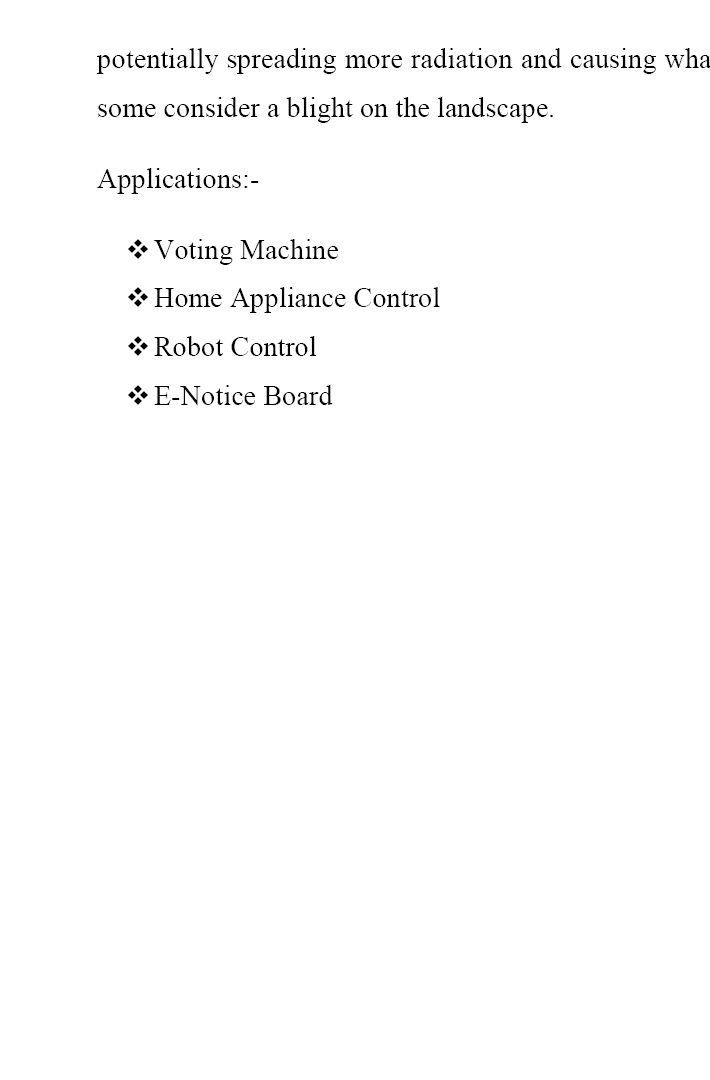
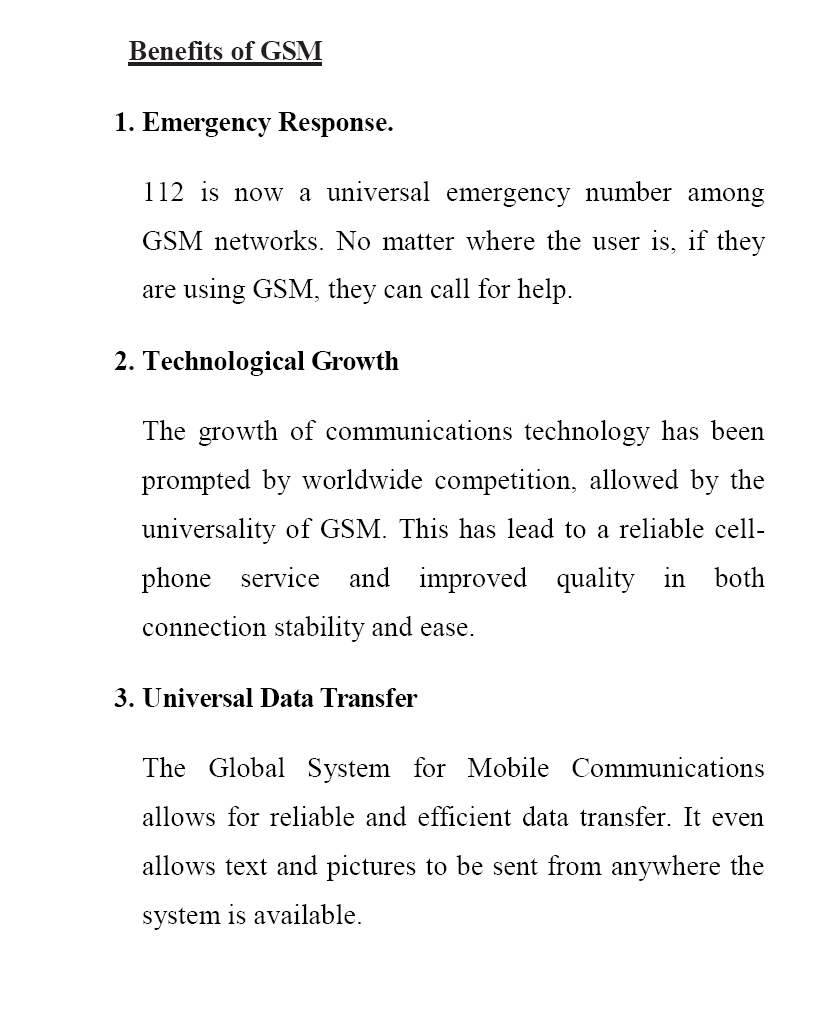
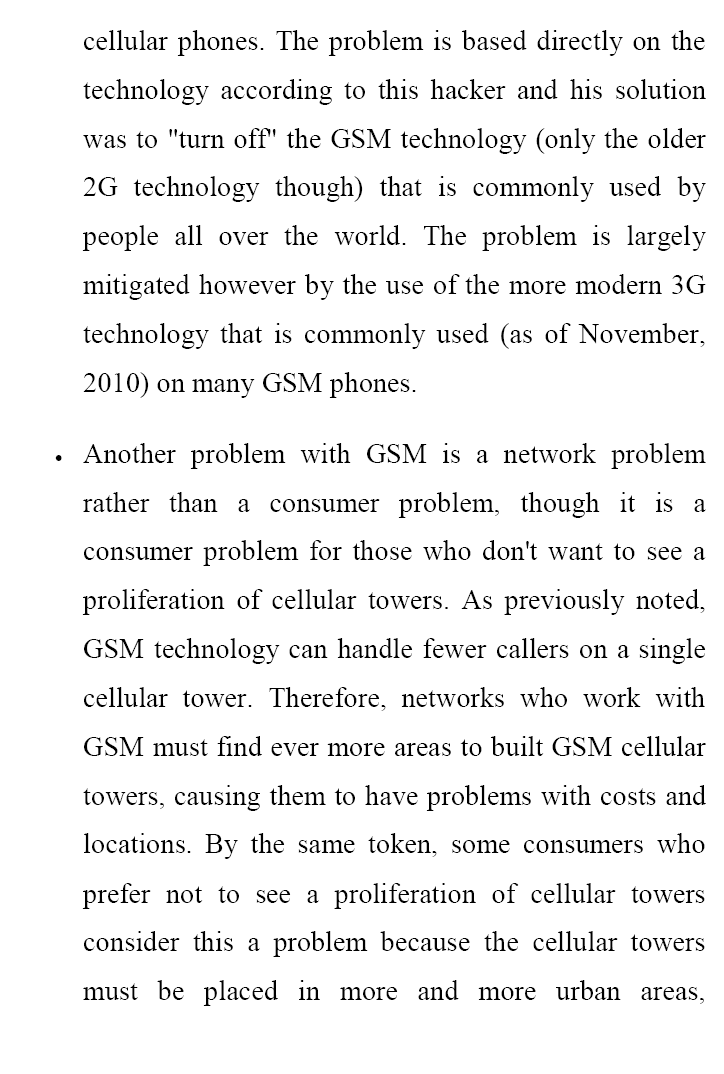
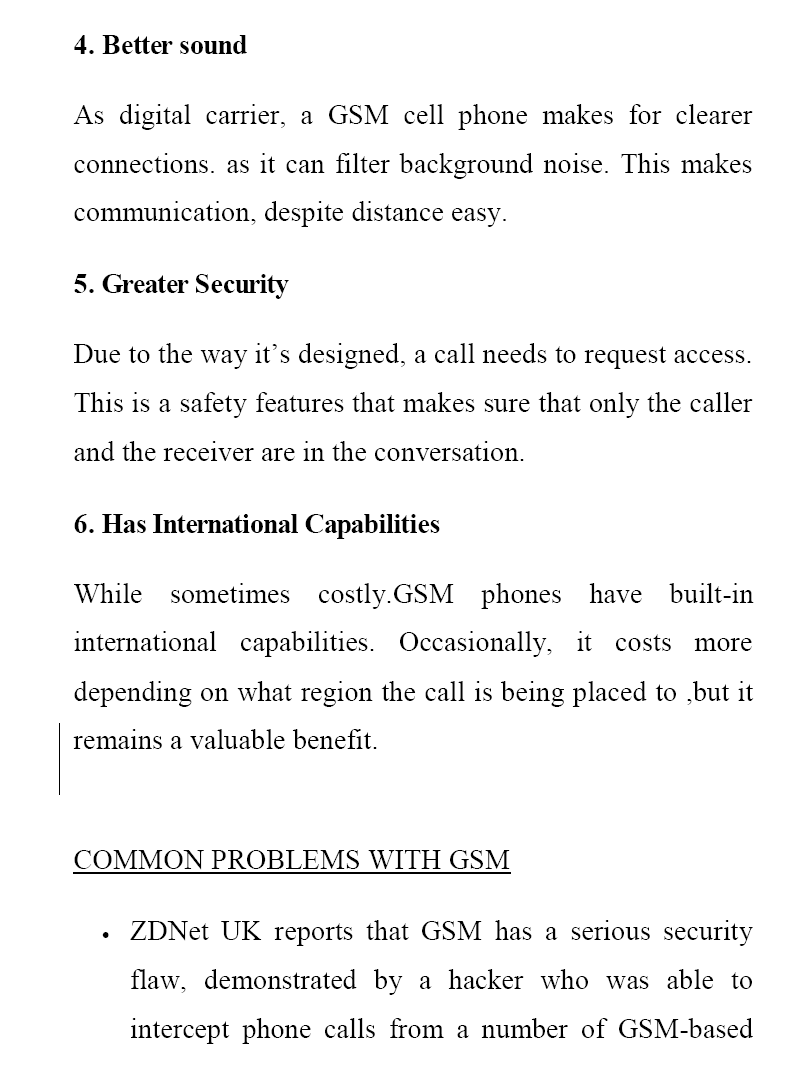
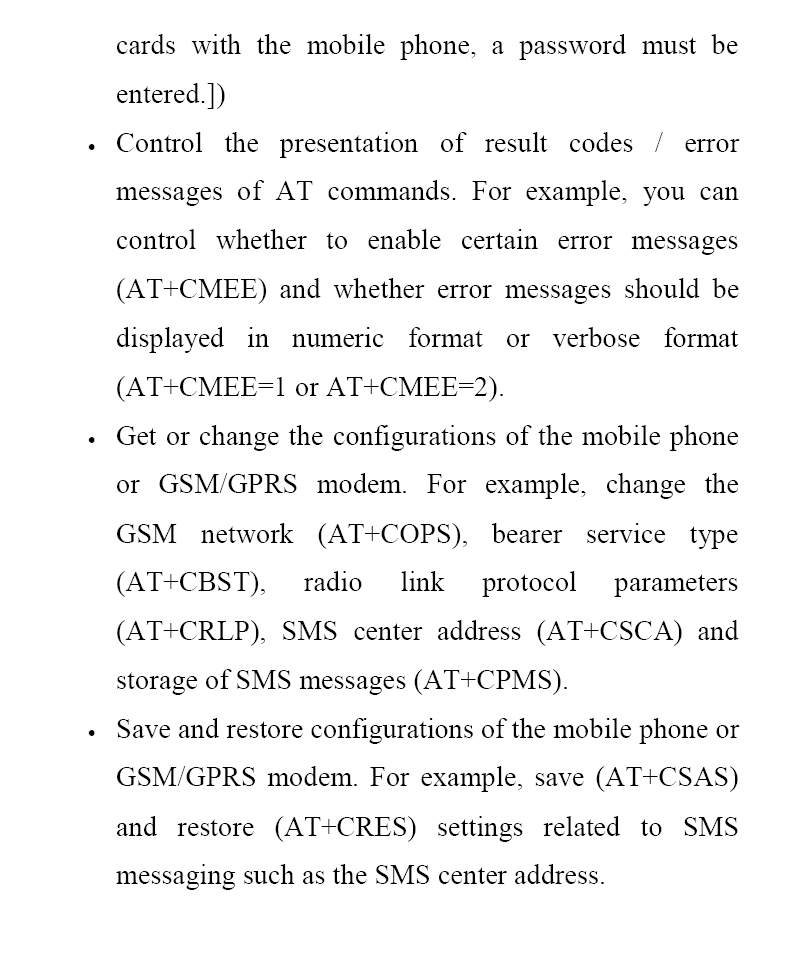
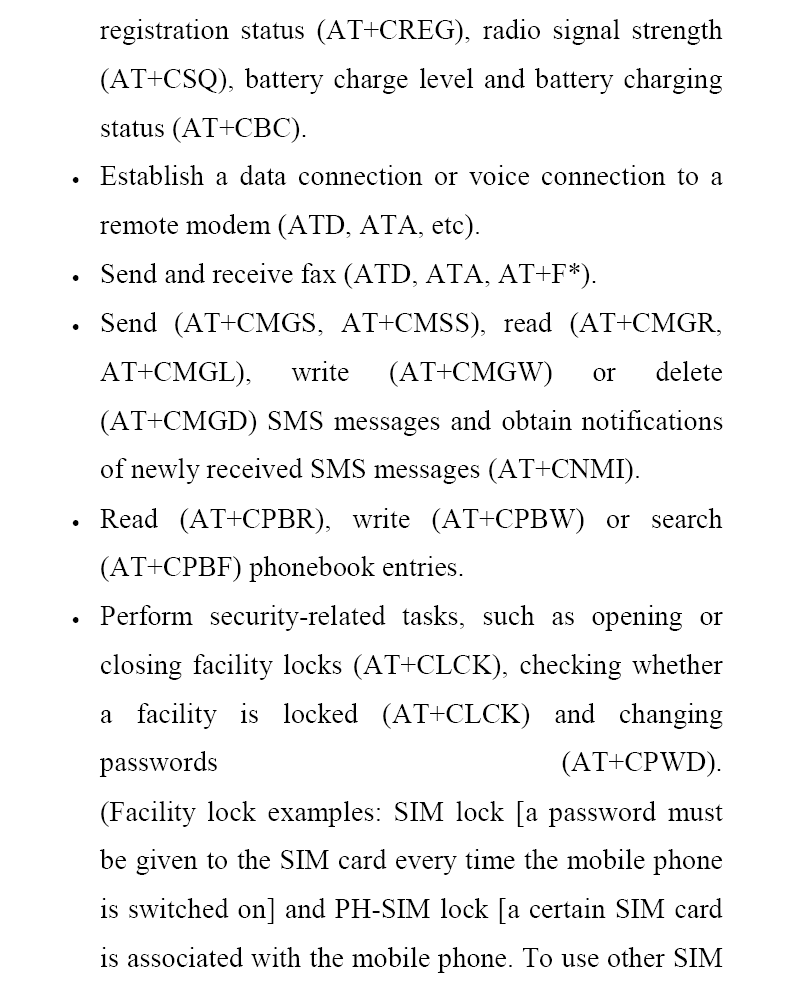
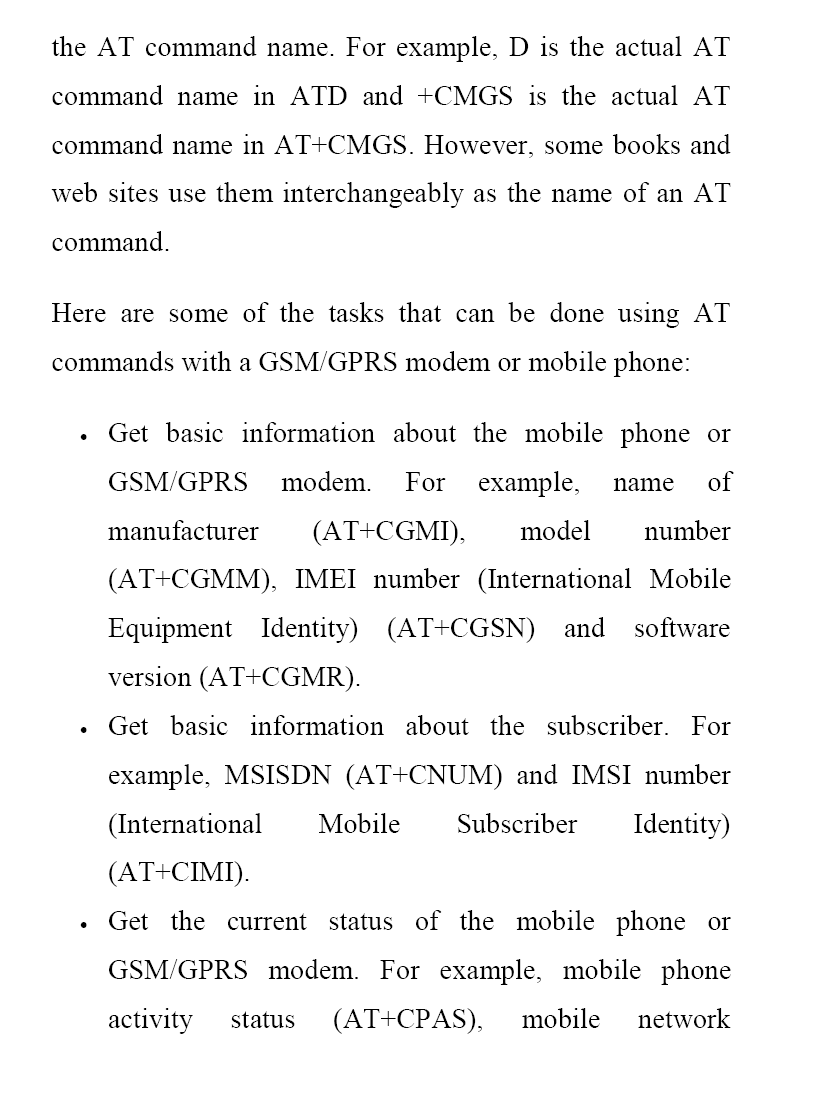
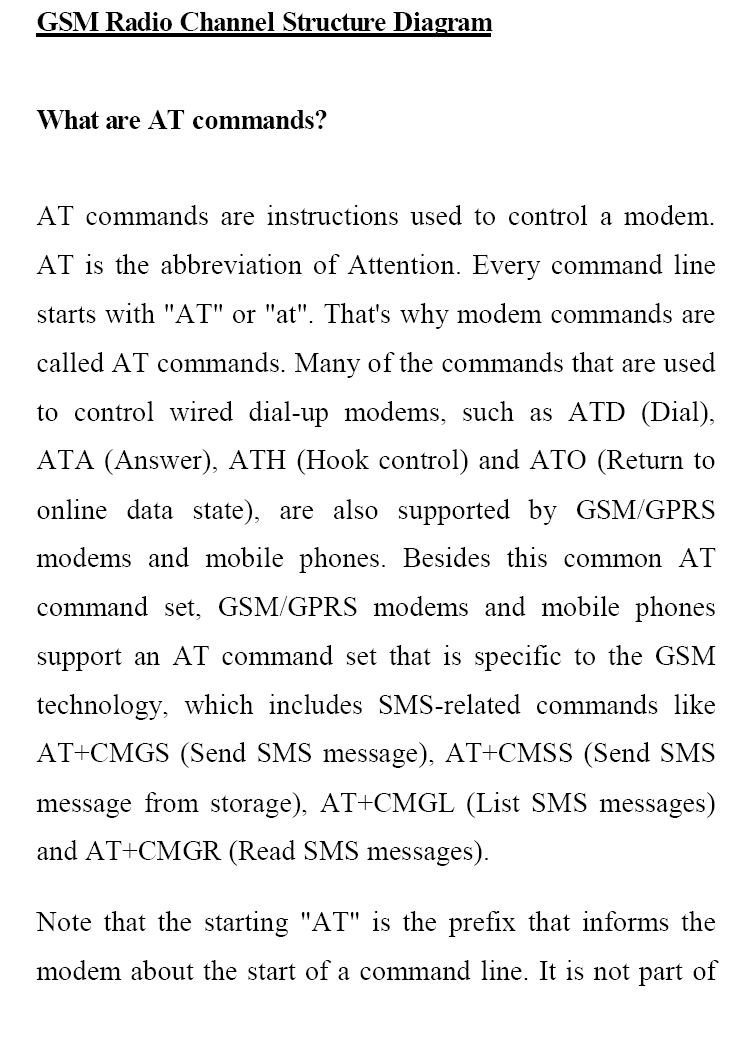
**Vibration Sensor**



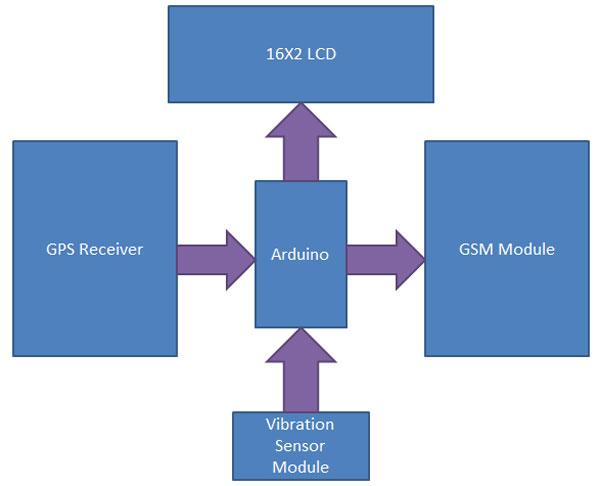
**Arduino**  
  
Here we are using Arduino UNO R3 but guys u have an option of making your own breakout board using Atmega 328, 168, 8 According to your application but don't forgot to burn bootloader without that u never gona burn any code from Arduino board (To burn bootloader in blank chip please follow our link HOW TO BURN BOOTLOADER IN ATMEGA 328) next for connections refer circuit diagram. If you are using Atmega 328/168/8 with own made board then follow this pin mapping and compare it with standard Arduino



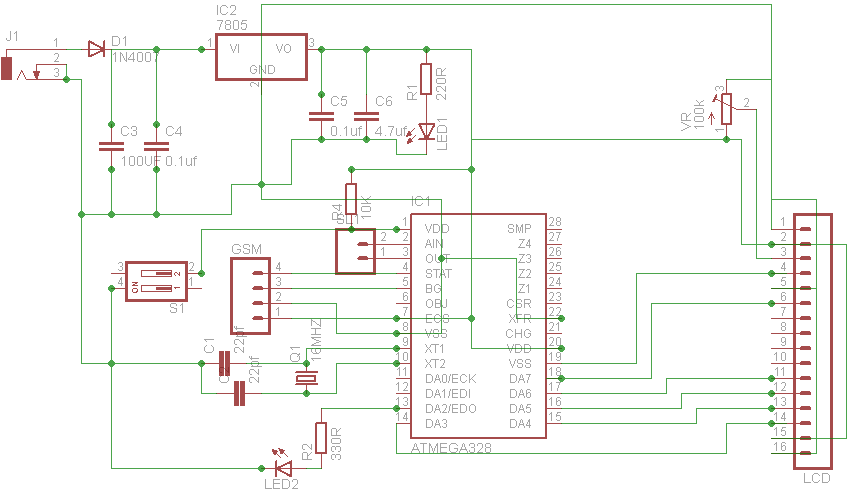
GSM and GPS based vehicle location and tracking system will provide effective, real time vehicle location, mapping and reporting this information value and adds by improving the level of service provided. A GPS-based vehicle tracking system will inform where your vehicle is and where it has been, how long it has been. The system uses geographic position and time information from the Global Positioning Satellites. The system has an "OnBoard Module" which resides in the vehicle to be tracked and a "Base Station" that monitors data from the various vehicles. The On-Board module consists of GPs receiver, a GSM modem both consumers, who may benefit from the ability to roam and switch carriers without replacing phones, and also to network operators, who can choose equipment from many GSM equipment vendors.



**BLOCK DIAGRAM**



**Circuit Diagram**



**Component List**

Part Value Quantity

C1 22pf 2

C2 100uF 1

C3 0.1uf 2

C4 4.7uf 1

D1 1N4007 1

GSM sim900 1

IC1 ATMEGA328 1

IC2 7805 1

LCD 16\*2 1

LED1 3MM 2

CRYSTAL 16MHZ 1

R1 220R 1

R2 330R 1

R3 10K 1

S1 push switch 1

VR 20k 1

GPS

1

**Source Code**

#include <string.h>

#include <ctype.h>

#include <SoftwareSerial.h>

SoftwareSerial SIM900(2, 3);

SoftwareSerial GPS(4,5);

char incoming\_char=0;

int vib\_sensor = A0;

int ledPin = 13; // LED test pin

int byteGPS=-1;

char linea[300] = "";

char comandoGPR[7] = "$GPRMC";

int cont=0;

int bien=0;

int conta=0;

int indices[13];

void setup()

{

SIM900.begin(9600); // for GSM shield

GPS.begin(9600); // for GPS shield

SIM900power(); // turn on shield

delay(20000); // give time to log on to network.

SIM900.print("AT+CMGF=1\r"); // set SMS mode to text

delay(100);

SIM900.print("AT+CNMI=2,2,0,0,0\r");

// blurt out contents of new SMS upon receipt to the GSM shield's serial out

delay(100);

pinMode(ledPin, OUTPUT); // Initialize LED pin

Serial.begin(9600);

for (int i=0;i<300;i++){ // Initialize a buffer for received data

linea[i]=' ';

}

}

void SIM900power()

// software equivalent of pressing the GSM shield "power" button

{

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(7000);

}

char lat[12]={"28.4709161N"};

int lat\_count=0;

char lon[20]={"77.5031764000000E"};;

int lon\_count=0;

int loop\_count=0;

void sendSMS()

{

Serial.println("sending sms");

SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message

delay(100);

SIM900.println("AT+CMGS=\"+917827855025\"\r"); // recipient's mobile number, in international format

delay(100);

SIM900.print("Latitude:"); // message to send

Serial.print("Latitude: ");

delay(100);

SIM900.print(lat); // message to send

Serial.println(lat);

delay(100);

SIM900.print("Longitude:"); // message to send

Serial.print("Longitude: ");

delay(100);

SIM900.print(lon); // message to send

Serial.println(lon);

delay(100);

SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26

delay(2000);

SIM900.println();

delay(5000); // give module time to send SMS

SIM900power(); // turn off module

}

void loop()

{

byteGPS=GPS.read(); // Read a byte of the serial port

if (byteGPS == -1)

{ // See if the port is empty yet

delay(100);

} else {

linea[conta]=byteGPS; // If there is serial port data, it is put in the buffer

conta++;

//Serial.write(byteGPS);

if (byteGPS==13){ // If the received byte is = to 13, end of transmission

// note: the actual end of transmission is <CR><LF> (i.e. 0x13 0x10)

digitalWrite(ledPin, LOW);

cont=0;

bien=0;

// The following for loop starts at 1, because this code is clowny and the first byte is the <LF> (0x10) from the previous transmission.

for (int i=1;i<7;i++){ // Verifies if the received command starts with $GPR

if (linea[i]==comandoGPR[i-1]){

bien++;

}

}

if(bien==6){ // If yes, continue and process the data

for (int i=0;i<300;i++){

if (linea[i]==','){ // check for the position of the "," separator

// note: again, there is a potential buffer overflow here!

indices[cont]=i;

cont++;

}

if (linea[i]=='\*'){ // ... and the "\*"

indices[12]=i;

cont++;

}

}

// Serial.println(""); // ... and write to the serial port

// Serial.println("");

Serial.println("---------------");

Serial.println(loop\_count++);

for (int i=0;i<12;i++){

switch(i){

//case 0 :Serial.print("Time in UTC (HhMmSs): ");break;

//case 1 :Serial.print("Status (A=OK,V=KO): ");break;

case 2 :Serial.print("Latitude: ");

for (int j=indices[i];j<(indices[i+1]-1);j++){

// Serial.print(linea[j+1]);

lat[lat\_count++]=linea[j+1];

}

break;

case 3 :Serial.print("Direction (N/S): ");

for (int j=indices[i];j<(indices[i+1]-1);j++){

// Serial.print(linea[j+1]);

lat[lat\_count++]=linea[j+1];

}

for (int j=0;j<lat\_count;j++){

Serial.print(lat[j]);

}

break;

case 4 :Serial.print("Longitude: ");

for (int j=indices[i];j<(indices[i+1]-1);j++){

// Serial.print(linea[j+1]);

}

break;

case 5 :Serial.print("Direction (E/W): ");

for (int j=indices[i];j<(indices[i+1]-1);j++){

// Serial.print(linea[j+1]);

lon[lon\_count++]=linea[j+1];

}

for (int j=0;j<lat\_count;j++){

Serial.print(lon[j]);

}

lon\_count=0;

delay(10000);

sendSMS();

delay(10000);

break;

//case 6 :Serial.print("Velocity in knots: ");break;

//case 7 :Serial.print("Heading in degrees: ");break;

//case 8 :Serial.print("Date UTC (DdMmAa): ");break;

//case 9 :Serial.print("Magnetic degrees: ");break;

//case 10 :Serial.print("(E/W): ");break;

//case 11 :Serial.print("Mode: ");break;

//case 12 :Serial.print("Checksum: ");break;

}

Serial.println("");

}

Serial.println("---------------");

}

conta=0; // Reset the buffer

for (int i=0;i<300;i++){ //

linea[i]=' ';

}

}

}

}