Jaypee University of Information Technology
Waknaghat, Distt. Solan (H.P.)
Learning Resource Center

CLASS NUM: BOOK NUM.:

ACCESSION NO.: SP08085/SP0812088

This book was issued is overdue due on the date stamped below. if the book is kept over due, a fine will be charged as per the library rules.

Due Date	Due Date	Due Date
LRC CHECKED 12 NOV 2014		

GPS AND GSM BASED VEHICLE TRACKER

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Electronics and Communication Engineering

under the Supervision of

Ms Neeru Sharma

By

Kritika Choudhary (081021)

Nipun Agarwal (081094)

Tarang Kumar (081098)

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled " **GPS and GSM based vehicle tracker** ", submitted by Kritika Choudhary (081021), Nipun Agarwal (081094), Tarang Kumar (081098) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 01.06.2012

Supervisor's Name : Neeru Sharma

Designation : Sr. Lecturer

Acknowledgement

We would like to express our gratitude to our project supervisor Ms Neeru Sharma for giving us the opportunity and suggesting us a very useful topic to be taken as the final year B.Tech Project for the academic session 2011-2012. We would also like to show our greatest appreciation to our project review panel Prof. Dr. Sunil Bhooshan, Mr Salman Raju Talluri and Mr Mohammad Wajid for their valuable insights into our project.

Apart from the efforts by us, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the laboratory engineers who have been equally instrumental in the successful completion of this project.

Date: 01/06/2012

Kritika Choudhary

081021

Nipun Agarwal

081094

Tarang Kumar

081098

Table of Content

List of Figures

List of Tables

Abstract

Tracking systems were first developed for the shipping industries to determine the position of ships and boats in the sea. Initially passive systems were developed to support in tracking and navigation for location-based applications. For the applications that require real time location information of the vehicle, these systems cannot be employed, because they store the location information in the internal storage that can only be accessed when vehicle is available. Recently, Automatic Vehicle Location (AVL) systems are developed and deployed in numerous environments. These systems are capable of transmitting vehicle's location information in real time. In these systems, the device installed in the vehicle can transmit the location information in real time to a remote data centre, instead of storing into local storage, using some radio network.

Nowadays, vehicles are essential to transport products or goods in many organizations, but there are numerous problems encountered during the transportation process. E.g. Delay of deliveries, Driving out of paths, Stealing products etc. This project aims to develop an efficient and cost effective way to counter the aforementioned problems. In this project there are two units that were developed:-

1. Vehicle Position Unit
2. Remote Base Unit

The vehicle position unit consists of embedded circuit which is connected to GPS receiver and GSM modem. This unit is fitted on the intended vehicle to be tracked. Now, the GPS receiver receives signals from the satellites, sends the data to the processing circuit consisting of a microcontroller. Now, if the owner of the vehicle at the remote place is interested in knowing the location of it he can send an SMS to the GSM modem of the vehicle base unit via a GSM modem at the remote unit. The SMS will be received and the microcontroller of the vehicle base unit will send the location viz. latitudes and longitudes to the GSM modem connected to it, which will then send an SMS to the remote unit. The remote user after receiving the SMS can use the data to locate the vehicle on the computer which is loaded with software like Google Earth or through the use of GPS data websites on internet.

CHAPTER 1

1.1 INTRODUCTION

Most of the modern vehicle's tracking systems belong to the category of Automatic-Vehicle-Location (AVL) systems. AVL systems aid in determining the geographic positioning information of vehicles and transmitting it to a remotely located server. The vehicle's location is determined using GPS, while the transmission mechanism can be satellite, terrestrial radio or cellular connection from the vehicle to a radio receiver, satellite or nearby cell tower. After collecting positioning data, it is transmitted using some kind of telemetry or wireless communications systems. GSM is the most common used service for this purpose. The design and implementation of the system includes acquisition and transmission of vehicle's location information along with ignition and doors status information to the monitoring station.

This project incorporates two parts:

1. Vehicle Position Unit
2. Base Unit.

The VPU consists of GPS receiver and GSM modem with a microcontroller. It is attached to the vehicle. On the other end i.e. Remote Base Unit one GSM mobile phone is attached to the computer . So the GPS system will send the longitude and latitude values corresponding to the position of vehicle to GSM Modem.

Imagine the vehicle has left in the morning. If the person in charge for that vehicle wants to know where the vehicle is, he will come to the computer and send an SMS to the vehicle number.

The SMS sent would come through the GSM service provider and then reach the vehicle, which is traveling, because the vehicle has a GSM device with sim card. This GSM modem will receive the SMS and send to the microcontroller in the vehicle. The microcontroller will receive this SMS and compare the message with the already fed character in the microcontroller. If this matches then it will perform the request required by the office.

Fig 1.1 highlights the working of GPS and GSM based vehicle tracking system. The GPS receiver receives signals from the GPS satellites orbiting above the earth and sends the data to the microcontroller (ATmega162) for processing. The remote user will send an SMS from the Base Unit to the Vehicle Position Unit via network provider. The Vehicle Position Unit after verifying the remote user's authenticity sends an SMS carrying the position of the intended vehicle to the user.

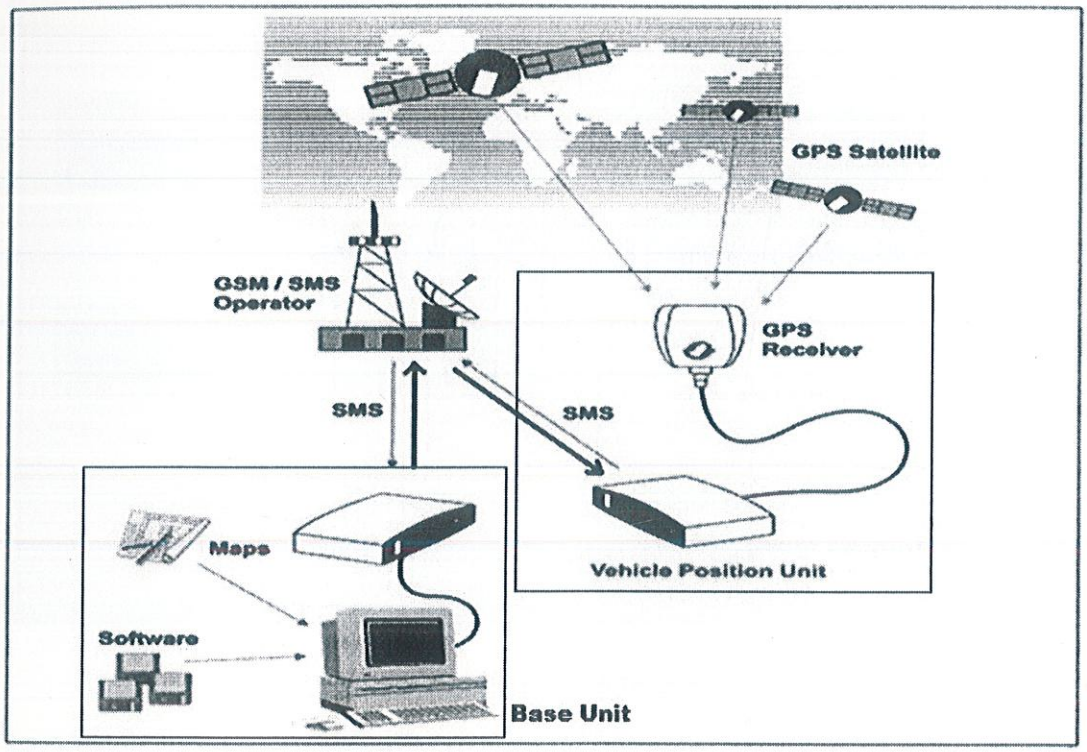

Fig 1.1: Block Diagram of Vehicle Tracking System

1.2 VEHICLE POSITION UNIT

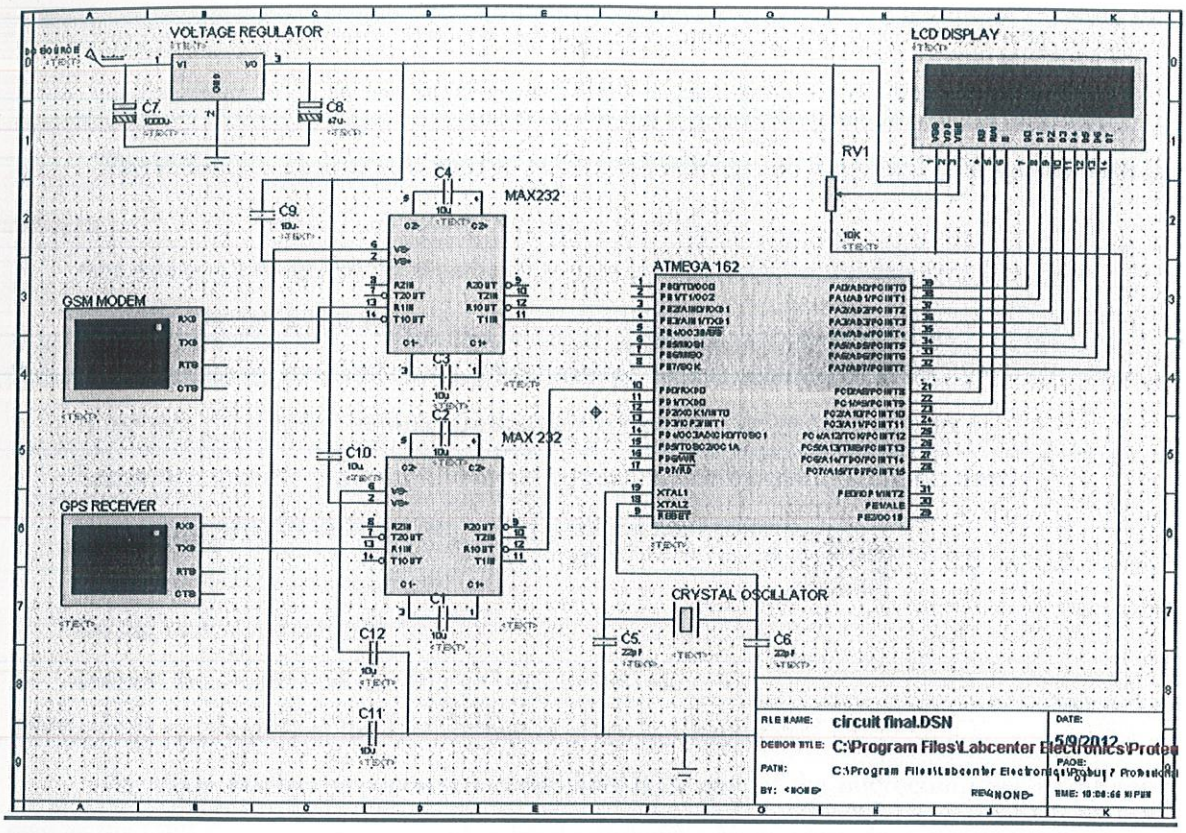

Fig 1.2: Circuit Diagram

The Vehicle Position Unit comprises of the following:

1. GPS Receiver
2. GSM Modem
3. Microcontroller
4. LCD Display
5. Regulated Power Supply

1.3 CIRCUIT FUNCTIONING

As shown in the circuit diagram the GPS Receiver is connected to the microcontroller via a level converter. GPS Receiver gives output in serial form (RS232). The output consists of a series of string. The microcontroller accepts TTL voltage levels. The level converter (MAX232) converts the serial data to TTL levels and sends to the USART0 port of the microcontroller.

The following algorithm is used to extract the latitude and longitude information from the GPS module using $GPGGA string and display it on a LCD:

1. Get data in UDR and check weather that data is equal to $. If the data matches go to Step (2) else get a new data.
2. Get data byte by byte and check if the received byte is equal to GPGGA
3. If the step (2) matches completely then go to step (4) else go back to step(1)
4. Leave first comma and wait till second comma (since we not looking for time).
5. Start taking data in an array lati_value[ ] till the next comma.
6. Get latitude direction in lati_dir.
7. Do the same for longitude
8. Display the values on LCD and go back to step (1).

The detailed functioning of the GPS system is given in Chapter 2.

The GSM Modem is connected to the USART1 port of the microcontroller in the same way as the GPS Receiver i.e. via another level converter. Now, since the microcontroller would be busy in receiving data from the GPS receiver, it needs to be interrupted in case the user demands the location from the Base Unit. In this case, when the GSM Modem receives an SMS requesting the location of the vehicle, it generates an interrupt. The microcontroller is programmed to acknowledge this interrupt and it sends the location of the vehicle to the GSM Modem which then sends this data to the Base Unit through SMS.

The common settings for the GPS Receiver and the GSM Modem are as follows:

Baud Rate: 9600 bps

Data Bits: 8

Stop Bits: 1

Parity Bits: None

1.4 RS-232 SERIAL COMMUNICATIONS

Serial communication involves sending one bit at a time. This is one of the slowest means of data transmission, however, it is still widely used in the scientific community because it is also one of the simplest and most expedient. RS-232 by the books is limited to fifty feet maximum cable length. Serial works by (as much of as possible) a square wave (voltage over time). Period and duty cycle will be decided by the agreed upon baud rate. Good house keeping denotes a positive 2.5 to 15 volts equal a logic 0 and negative 2.5 to 15 volts give you a logic 1. However, with many microcontrollers and embedded systems not equipped with UARTs, a positive 5 volts and a nice pullto- ground is just fine for a logical 0 and 1 respectively. This is inverted as compared to most systems and so makes easy work for testing communication lines with a multimeter. Nine pins are required for the full spectrum of what RS-232 has to offer. For hardware handshaking, five pins are needed; Data Carrier Detect, Data Terminal Ready, Data Set Ready, Request To Send, and Clear To Send. Packet transmission needs two dedicated pins, Serial Data Input, and Serial Data Output. The ninth pin is a ring indicator (for modem use, or to "wake" some device up).

Baud Rate: The speed of serial transmissions is timed in a ratio denoted Baud. In its simplest form 1 baud is 1 bit per second. Therefore, 1200 baud is 1200 bits per second; since the serial protocol is one bit at a time, the baud rate also tells us the sampling rate of the transmission. This means two transceivers communicating at 33,600 baud both have there UART clocks oscillating at 33,600 cycles per second or 33.6 kilohertz. Transmission speed is limited by many variables. Distance limits clock speed mainly because of the internal resistance of the transmission wires, but also because of induction. The hall effect has some small amount of degradation to the transmission. Over vast distances, hardware will start to see an RMS reading of two distinct pulses. In other words, it is possible to turn the transmitter into a pulse width modulator or a digital to analog converter.

Data Bits: Many bits are packaged together in groups of like numbers for data transmission. This could be thought of as people (representing bits) gathering onto a train car. When one car is full the next car pulls forward and loads the same number of people (or bits) to

be carried off to some distant land. These packages of data are called packets. RS-232 will let sender and receiver agree to any number of packet size (so long as both transceivers are in accordance) however, standard packet sizes are 5, 7, and 8 bits. These are the actual raw data elements that needs to be transmitted, not including the start/stop bits, and parity, which will be discussed later. When the number of data bits is being set for packet size, one must take into account what type of data will be predominantly sent.

Stop bits: Stop bits signal the end of a packet. Since the data is essentially represented by a square wave (voltage/ time), and the sender and receiver clocks can slip out of synchronization, stop bits give the computers or routers or whatever is communicating, room for error in clock speed (measured in Parts Per Million of error counts). RS-232 suggests typical values of 1, 1.5, and 2 bits. However, any number of stop bits may be used. The greater the number of stop bits the slower the data rate, this is simply because there is now more data to convey (and more data to process if on a limited system such as a router controller or small robot). The greater the number of stop bits, the greater the modulation error that can be afforded in two separate phase locked loops. This means when using a technique such as bit-banging (software calling hardware clock functions) where clock error will be common, one needs to use a high number of stop bits.

1.5 USART TERMINAL

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection

- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

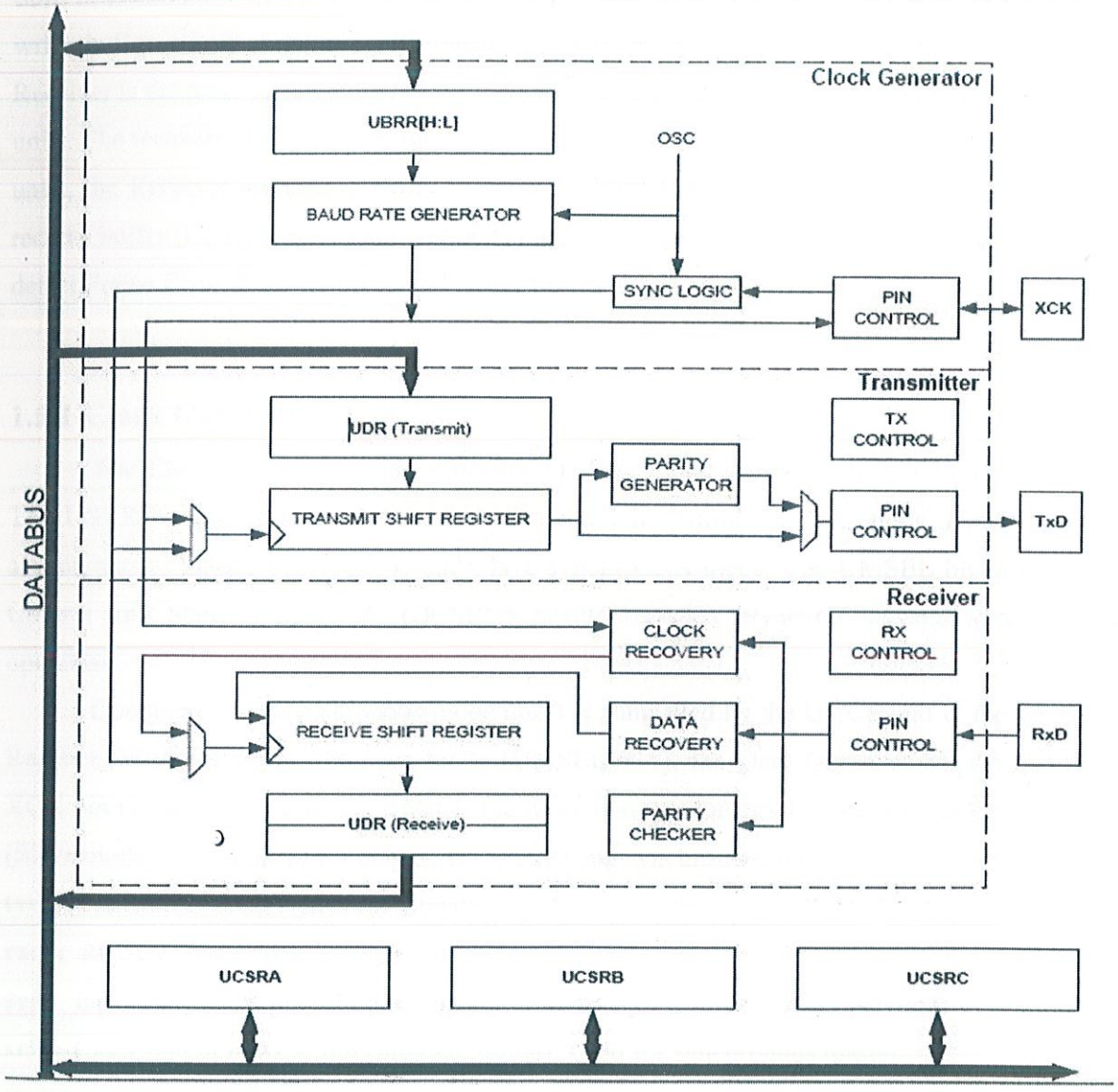

Fig 1.3: USART

The dashed boxes in the Fig 1.3 separate the three main parts of the USART (listed from

the top): Clock Generator, Transmitter and Receiver. Control registers are shared by all units. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by synchronous transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the Receiver includes a Parity Checker, Control logic, a Shift Register and a two level receive buffer (UDR). The receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

1.5.1 Clock Generation

The Clock Generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal asynchronous, Double Speed asynchronous, Master synchronous and Slave synchronous mode. The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation.

Double Speed (asynchronous mode only) is controlled by the U2X found in the UCSRA Register. When using synchronous mode (UMSEL = 1), the Data Direction Register for the XCK pin (DDR_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using synchronous mode.

txclk Transmitter clock. (Internal Signal)

rxclk Receiver base clock. (Internal Signal)

xcki Input from XCK pin (internal Signal). Used for synchronous slave operation.

xcko Clock output to XCK pin (Internal Signal). Used for synchronous master operation.

fosc XTAL pin frequency (System Clock).

1.5.2 Internal Clock Generation – The Baud Rate Generator

Internal clock generation is used for the asynchronous and the synchronous master modes of operation.The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (fosc), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= fosc/(UBRR+1)). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the receiver's clock and data recovery units. However, the recovery units

use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR_XCK bits.

Table 1.1: Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{OSC} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

1.5.3 Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Figure illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

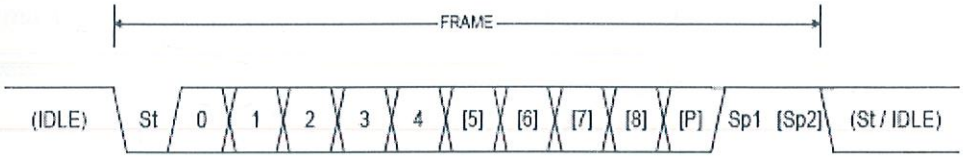

Fig 1.4: Frame Formats

St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

The frame format used by the USART is set by the UCSZ2:0, UPM1:0 and USBS bits in UCSRB and UCSRC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and

Transmitter. The USART Character SiZe (UCSZ2:0) bits select the number of data bits in the frame. The USART Parity mode (UPM1:0) bits enable and set the type of parity bit. The selection between one or two stop bits is done by the USART Stop Bit Select (USBS) bit. The receiver ignores the second stop bit. An FE (Frame Error) will therefore only be detected in the cases where the first stop bit is zero.

1.5.4 USART Initialization

The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization. Before doing a re-initialization with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed. The TXC Flag can be used to check that the Transmitter has completed all transfers, and the RXC Flag can be used to check that there are no unread data in the receive buffer. Note that the TXC Flag must be cleared before each transmission (before UDR is written) if it is used for this purpose.

1.5.5 Data Transmission– The USART Transmitter

The USART Transmitter is enabled by setting the *Transmit Enable* (TXEN) bit in the UCSRB Register. When the Transmitter is enabled, the normal port operation of the TxD pin is overridden by the USART and given the function as the transmitter's serial output. The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If synchronous operation is used, the clock on the XCK pin will be overridden and used as transmission clock.

Sending Frames with 5 to 8 Data Bit

A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDR I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is

ready to send a new frame. The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted. When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the Baud Register, U2X bit or by XCK depending on mode of operation.

1.5.6 Data Reception –The USART Receiver

The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one. When the receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the receiver's serial input. The baud rate, mode of operation and frame format must be set up once before any serial reception can be done. If synchronous operation is used, the clock on the XCK pin will be used as transfer clock.

Receiving Frames with 5 to 8 Data Bits

The Receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the Receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the Receiver. When the first stop bit is received, i.e., a complete serial frame is present in the Receive Shift Register, the contents of the Shift Register will be moved into the receive buffer. The receive buffer can then be read by reading the UDR I/O location.

Receive Compete Flag and Interrupt

The USART Receiver has one flag that indicates the receiver state. The Receive Complete (RXC) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled (RXEN = 0), the receive buffer will be flushed and consequently the RXC bit will become zero.

When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete Interrupt will be executed as long as the RXC Flag is set (provided that global interrupts are enabled). When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

1.5.7 USART Register Description

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig 1.5: USART I/O Data Register – UDR

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDR. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDR Register location. Reading the UDR Register location will return the contents of the Receive Data Buffer Register (RXB). For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the Transmitter and set to zero by the Receiver.

The transmit buffer can only be written when the UDRE Flag in the UCSRA Register is set. Data written to UDR when the UDRE Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the Transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxD pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use read modify write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Fig 1.6: USART Control and Status Register A –UCSRA

• **Bit 7 – RXC: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXC bit will become zero. The RXC Flag can be used to generate a Receive Complete interrupt (see description of the RXCIE bit).

• **Bit 6 – TXC: USART Transmit Complete**

This flag bit is set when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDR). The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXC Flag can generate a Transmit Complete interrupt (see description of the TXCIE bit).

• **Bit 5 – UDRE: USART Data Register Empty**

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written. The UDRE Flag can generate a Data Register Empty interrupt (see description of the UDRIE bit). UDRE is set after a Reset to indicate that the transmitter is ready.

• **Bit 4 – FE: Frame Error**

This bit is set if the next character in the receive buffer had a Frame Error when received. i.e., when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDR) is read. The FE bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRA.

• **Bit 3 – DOR: Data OverRun**

This bit is set if a Data OverRun condition is detected. A data overrun occurs when the receive buffer is full (two characters), it is a new character waiting in the reCeive Shift Register, and a new start bit is detected. This bit is valid until the receive buffer (UDR) is read. Always set this bit to zero when writing to UCSRA.

• **Bit 2 – UPE: Parity Error**

This bit is set if the next character in the receive buffer had a Parity Error when received and the Parity Checking was enabled at that point (UPM1 = 1). This bit is valid until the receive buffer (UDR) is read. Always set this bit to zero when writing to UCSRA.

• **Bit 1 – U2X: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Write this bit to zero when using synchronous operation. Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling the transfer rate for asynchronous communication.

• Bit 0 – MPCM: Multi-processor Communication Mode

This bit enables the Multi-processor Communication mode. When the MPCM bit is written to one, all the incoming frames received by the USART receiver that do not contain address information will be ignored. The transmitter is unaffected by the MPCM setting.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig 1.7: USART Control and Status Register B –UCSRB

• Bit 7 – RXCIE: RX Complete Interrupt Enable

Writing this bit to one enables interrupt on the RXC Flag. A USART Receive Complete interrupt will be generated only if the RXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXC bit in UCSRA is set.

• Bit 6 – TXCIE: TX Complete Interrupt Enable

Writing this bit to one enables interrupt on the TXC Flag. A USART Transmit Complete interrupt will be generated only if the TXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXC bit in UCSRA is set.

• Bit 5 – UDRIE: USART Data Register Empty Interrupt Enable

Writing this bit to one enables interrupt on the UDRE Flag. A Data Register Empty interrupt will be generated only if the UDRIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDRE bit in UCSRA is set.

• Bit 4 – RXEN: Receiver Enable

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxD pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FE, DOR and UPE Flags.

• **Bit 3 – TXEN: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxD pin when enabled. The disabling of the Transmitter (writing TXEN to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxD port.

• **Bit 2 – UCSZ2: Character Size**

The UCSZ2 bits combined with the UCSZ1:0 bit in UCSRC sets the number of data bits (character size) in a frame the Receiver and Transmitter use.

• **Bit 1 – RXB8: Receive Data Bit 8**

RXB8 is the ninth data bit of the received character when operating with serial frames with nine data bits. Must be read before reading the low bits from UDR.

• **Bit 0 – TXB8: Transmit Data Bit 8**

RXB8 is the 9th data bit in the character to be transmitted when operating with serial frames with 9 data bits. Must be written before writing the low bits to UDR.

USART Control and Status Register C –UCSRC

• **Bit 7 – URSEL: Register Select**

This bit selects between accessing the UCSRC or the UBRRH Register. It is read as one when reading UCSRC. The URSEL must be one when writing the UCSRC.

• **Bit 6 – UMSEL: USART Mode Select**

This bit selects between asynchronous and synchronous mode of operation.

Table 1.2: UMSEL Bit Settings

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

• **Bit 5:4 – UPM1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and compare it to the UPM0 setting. If a mismatch is detected, the UPE Flag in UCSRA will be set.

Table 1.3: UPM Bit Settings

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

• **Bit 3 – USBS: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the transmitter. The receiver ignores this setting.

Table 1.4: USBS Bit Settings

USBS	Stop Bit(s)
0	1-bit
1	2-bit

• **Bit 2:1 – UCSZ1:0: Character Size**

The UCSZ1:0 bits combined with the UCSZ2 bit in UCSRB sets the number of data bits (character Size) in a frame the receiver and transmitter use.

Table 1.5: UCSZ Bits Settings

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

- **Bit 0 – UCPOL: Clock Polarity**

This bit is used for synchronous mode only. Write this bit to zero when asynchronous mode is used. The UCPOL bit sets the relationship between data output change and data input sample, and the synchronous clock (XCK).

Table 1.6: UCPOL Bit Settings

UCPOL	Transmitted Data Changed (Output of TxD Pin)	Received Data Sampled (Input on RxD Pin)
0	Rising XCK Edge	Falling XCK Edge
1	Falling XCK Edge	Rising XCK Edge

1.5.8 USART Baud Rate Registers – UBRRL and UBRRH

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Fig 1.8: USART Baud Rate Registers

- **Bit 15 – URSEL: Register Select**

This bit selects between accessing the UBRRH or the UCSRC Register. It is read as zero when reading UBRRH. The URSEL must be zero when writing the UBRRH.

- **Bit 14:12 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, these bit must be written to zero when UBRRH is written.

- **Bit 11:0 – UBRR11:0: USART Baud Rate Register**

This is a 12-bit register which contains the USART baud rate. The UBRRH contains the four most significant bits, and the UBRRL contains the eight least significant bits of the USART baud rate. Ongoing transmissions by the transmitter and receiver will be corrupted if the baud rate is changed. Writing UBRRL will trigger an immediate update of the baud rate prescaler.

1.6 COMPONENTS USED

1.6.1 MICROCONTROLLER ATmega162

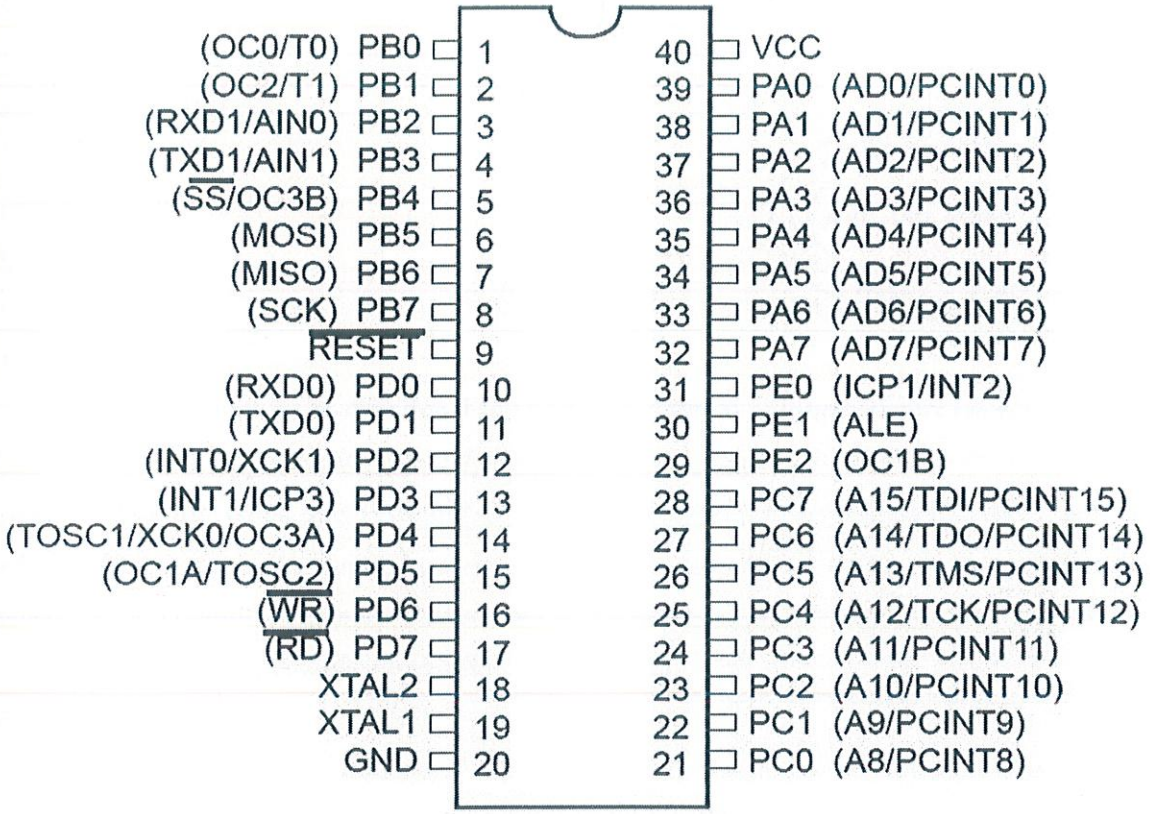

Fig 1.9: ATmega 162

The ATmega162 provides the following features: 16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 1K bytes SRAM, an external memory interface, 35 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, four flexible Timer/Counters with compare modes, internal and external interrupts, two serial programmable USARTs, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip

functions until the next interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

PIN DESCRIPTIONS:

Learning Resource Center
Acc. No. SP08085
SP08/2088
JUIT, Waknaghat, Solan (H.P.)

VCC

Digital supply voltage

GND

Ground

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the ATmega162.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega162.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on

pins PC7 (TDI), PC5(TMS) and PC4(TCK) will be activated even if a Reset occurs. Port C also serves the functions of the JTAG interface and other special features of the ATmega162.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega162.

Port E (PE2..PE0)

Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port E also serves the functions of various special features of the ATmega162.

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the Inverting Oscillator amplifier.

I/O PORTS

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All

port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both VCC and Ground.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write.

1.6.2 MAX 232

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept 30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels.

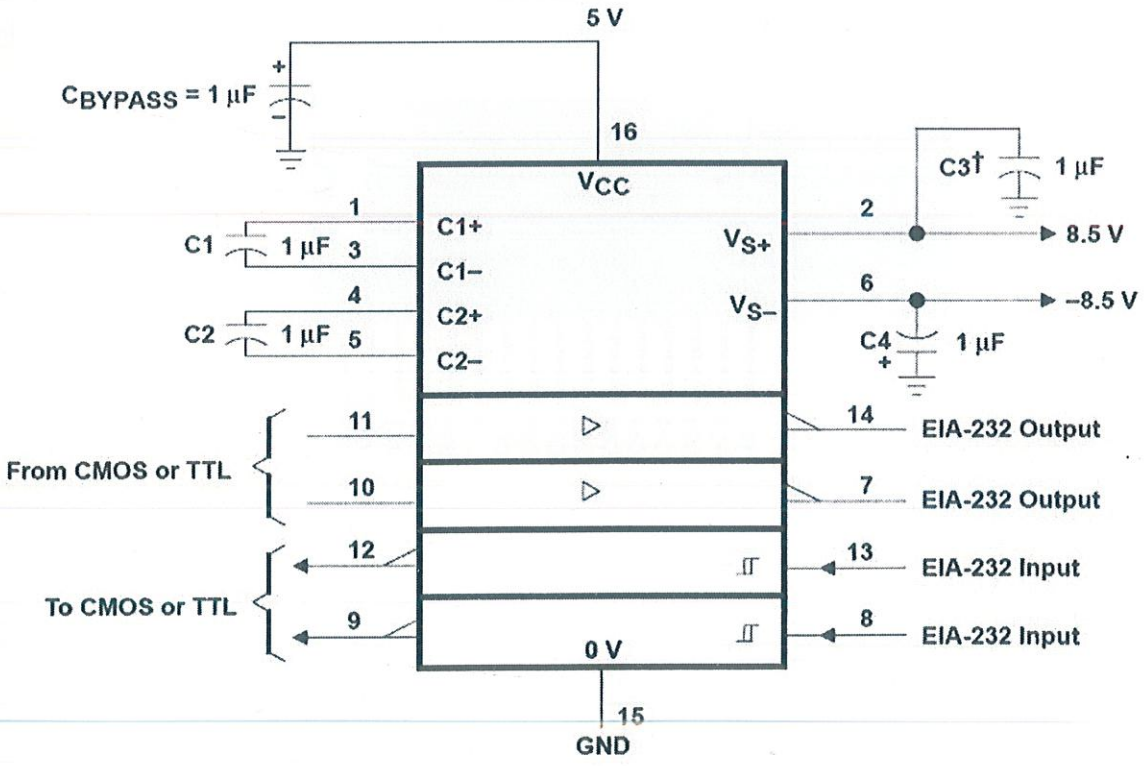

Fig 1.10: MAX 232

1.6.3 LCD

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 preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments) and animations.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

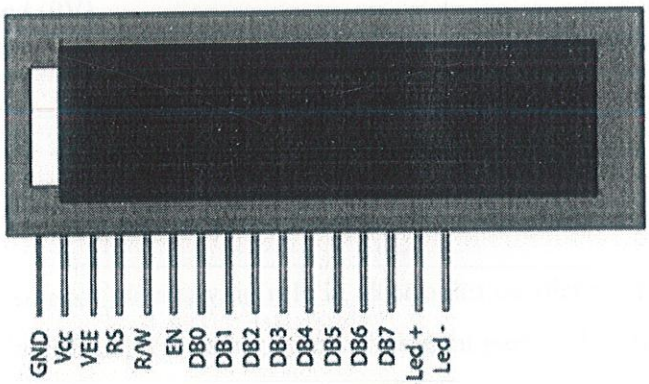

Fig 1.11: LCD Pin Diagram

Table 1.7: LCD Pin Description

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

1.6.4 CRYSTAL

A quartz crystal resonator plays a vital role in electronics oscillator circuitry. Sometimes mispronounced as crystal oscillator, it is rather a very important part of the feedback network of the oscillator circuitry. Electronics oscillators are used in frequency control application finding their usage in almost every industry ranging from small chips to aerospace.

A quartz crystal is the heart of such type of resonators. Their characteristics like high quality factor (Q), stability, small size and low cost make them superior over other resonators like LC circuit, turning forks, ceramic resonator etc.

The basic phenomenon behind working of a quartz crystal oscillator is the inverse piezo electric effect i.e., when electric field is applied across certain materials they start producing mechanical deformation. These mechanical deformation/movements are dependent on the

elementary structure of the quartz crystal. Quartz is one of the naturally occurring materials which show the phenomena of piezo electricity, however for the purpose of resonator it is artificially developed since processing the naturally occurring quartz is difficult and costly process.

1.6.5 LM7805

A LM7805 Voltage Regulator is a voltage regulator that outputs +5 volts. The voltage output by a LM78XX series of voltage regulators is the last two digits of the number. A LM7805 ends with "05"; thus, it outputs 5 volts. The "78" part is just the convention that the chip makers use to denote the series of regulators that output positive voltage. The other series of regulators, the LM79XX, is the series that outputs negative voltage. So:

LM78XX:Voltage regulators that output positive voltage, "XX"=voltage output.
LM79XX: Voltage regulators that output negative voltage, "XX"=voltage output

The LM7805, like most other regulators, is a three-pin IC.
Pin 1 (Input Pin): The Input pin is the pin that accepts the incoming DC voltage, which the voltage regulator will eventually regulate down to 5 volts.
Pin 2 (Ground): Ground pin establishes the ground for the regulator.
Pin 3 (Output Pin): The Output pin is the regulated 5 volts DC.

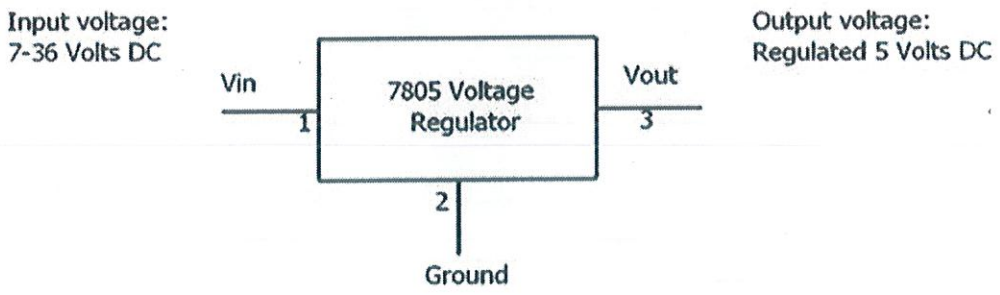

Fig 1.12: LM 7805 Voltage Regulator

CHAPTER 2

2.1 GLOBAL POSITIONING SYSTEM

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible to anyone with a GPS receiver.

The GPS program provides critical capabilities to military, civil and commercial users around the world. In addition, GPS is the backbone for modernizing the global air traffic system.

The GPS project was developed in 1973 to overcome the limitations of previous navigation systems, integrating ideas from several predecessors, including a number of classified engineering design studies from the 1960s. GPS was created and realized by the U.S. Department of Defense and was originally run with 24 satellites. It became fully operational in 1994.

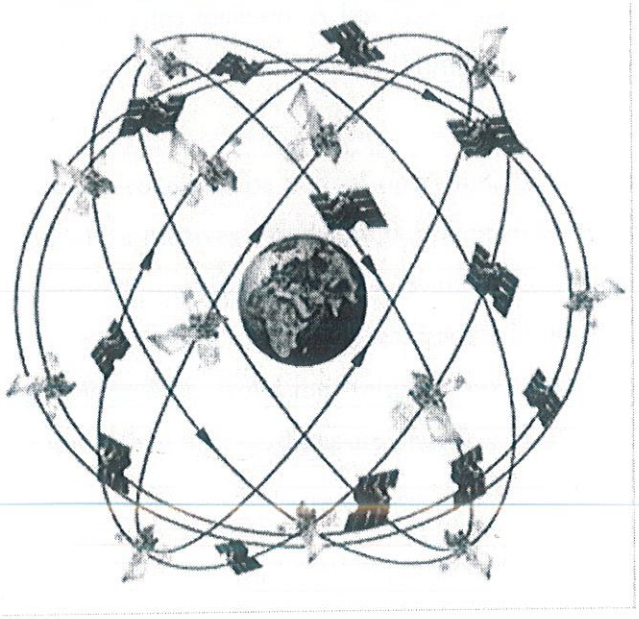

Fig 2.1: GPS System

2.1.1 Basic concept of GPS

A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include

- the time the message was transmitted
- satellite position at time of message transmission

The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite. These distances along with the satellites' locations are used with the possible aid of trilateration, depending on which algorithm is used, to compute the position of the receiver. This position is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS units show derived information such as direction and speed, calculated from position changes.

Three satellites might seem enough to solve for position since space has three dimensions and a position near the Earth's surface can be assumed. However, even a very small clock error multiplied by the very large speed of light- the speed at which satellite signals propagate — results in a large positional error. Therefore receivers use four or more satellites to solve for both the receiver's location and time. The very accurately computed time is effectively hidden by most GPS applications, which use only the location. A few specialized GPS applications do however use the time; these include time transfer, traffic signal timing, and synchronization of cell phone base stations.

Although four satellites are required for normal operation, fewer apply in special cases. If one variable is already known, a receiver can determine its position using only three satellites. For example, a ship or aircraft may have known elevation. Some GPS receivers may use additional clues or assumptions (such as reusing the last known altitude, dead reckoning, inertial navigation, or including information from the vehicle computer) to give a less accurate (degraded) position when fewer than four satellites are visible.

2.2 STRUCTURE OF GPS

The current GPS consists of three major segments. These are the space segment (SS), a control segment (CS), and a user segment (US). The U.S. Air Force develops, maintains, and operates the space and control segments. GPS satellites broadcast signals from space, and each GPS receiver uses these signals to calculate its three-dimensional location (latitude, longitude, and altitude) and the current time.

The space segment is composed of 24 to 32 satellites in medium Earth orbit and also includes the payload adapters to the boosters required to launch them into orbit. The control segment is composed of a master control station, an alternate master control station, and a host of dedicated and shared ground antennas and monitor stations. The user segment is composed of hundreds of thousands of U.S. and allied military users of the secure GPS Precise Positioning Service, and tens of millions of civil, commercial, and scientific users of the Standard Positioning Service.

2.3 GPS Method of Operation

A GPS receiver calculates its position by carefully timing the signals sent by the constellation of GPS satellites high above the Earth. Each satellite continually transmits messages containing the time the message was sent, a precise orbit for the satellite sending the message (the ephemeris), and the general system health and rough orbits of all GPS satellites (the almanac). These signals travel at the speed of light through outer space, and slightly slower through the atmosphere. The receiver uses the arrival time of each message to measure the distance to each satellite thereby establishing that the GPS receiver is approximately on the surfaces of spheres centered at each satellite.

The GPS receiver also uses, when appropriate, the knowledge that the GPS receiver is on (if vehicle altitude is known) or near the surface of a sphere centered at the earth center. This information is then used to estimate the position of the GPS receiver as the intersection of sphere surfaces. The resulting coordinates are converted to a more convenient form for the user such as

latitude and longitude, or location on a map, then displayed. It might seem that three sphere surfaces would be enough to solve for position, since space has three dimensions.

However a fourth condition is needed for two reasons. One has to do with position and the other is to correct the GPS receiver clock. It turns out that three sphere surfaces usually intersect in two points. Thus a fourth sphere surface is needed to determine which intersection is the GPS receiver position. For near earth vehicles, this knowledge that it is near earth is sufficient to determine the GPS receiver position since for this case there is only one intersection which is near earth. A fourth sphere surface is also needed to correct the GPS receiver clock. More precise information is needed for this task. An estimate of the radius of the sphere is required. Therefore an approximation of the earth altitude or radius of the sphere centered at the satellite must be known.

2.4 BLOCK DIAGRAM

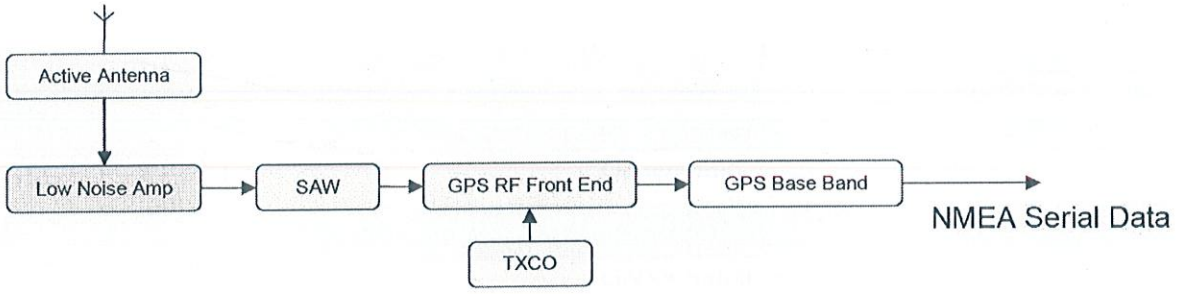

Fig 2.2: GPS Receiver Block Diagram

The GPS Receiver consist of two units, first is active antenna which receives RF signals and amplifies it. The antenna is active in the sense it takes power from the module and amplifies the signal for high sensitivity. The RF signal is filtered and processed to generate NMEA format serial data output.

2.5 NMEA PROTOCOL

This section provides a brief overview of the NMEA 0183 protocol, and describes both the standard and optional messages offered by the GPS Receiver. NMEA 0183 is a simple, yet comprehensive ASCII protocol which defines both the communication interface and the data format. The NMEA 0183 protocol was originally established to allow marine navigation equipment to share information. Since it is a well established industry standard, NMEA 0183 has also gained popularity for use in applications other than marine electronics. The GPS receiver supports the latest release of NMEA 0183, Version 3.0 (July 1, 2000). The primary change in release 3.0 is the addition of the mode indicators in the GLL, RMC, and VTG messages.

For those applications requiring output only from the GPS receiver, the standard NMEA 0183 sentences are a popular choice. Many standard application packages support the standard NMEA output messages. The standard NMEA output only messages are: GGA, GLL, GSA, GSV, RMC, VTC, and ZDA.

Table 2.1: NMEA Messages

NMEA RECORD	DESCRIPTION
GGA	GPS fix data
GLL	Geographic
GSA	GNSS DOP and active satellite
GSV	GNSS Satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course Over Ground and Ground Speed
ZDA	Time&Data

2.5.1 GGA - Global Positioning System Fix Data

Time, position and fix related data for a GPS receiver.

Structure:
*$GPGGA,hhmmss.sss,ddmm.mmmm,a,dddmm.mmmm,a,x,xx,x.x,x.x,M,,,,xxxx*hh<CR><LF>*
Field 1 2 3 4 5 6 7 8 9 10 11

Example:
*$GPGGA,111636.932,2447.0949,N,12100.5223,E,1,11,0.8,118.2,M,,,,0000*02<CR><LF>*
Field 1 2 3 4 5 6 7 8 9 10 11

Table 2.2: Field Description

Field	Name	Example	Description
1	**UTC Time**	111636.932	UTC of position in hhmmss.sss format, (000000.000 ~ 235959.999)
2	Latitude	2447.0949	Latitude in ddmm.mmmm format Leading zeros transmitted
3	N/S Indicator	N	Latitude hemisphere indicator, 'N' = North, 'S' = South
4	Longitude	12100.5223	Longitude in dddmm.mmmm format Leading zeros transmitted
5	E/W Indicator	E	Longitude hemisphere indicator, 'E' = East, 'W' = West
6	GPS quality indicator	1	GPS quality indicator 0: position fix unavailable 1: valid position fix, SPS mode 2: valid position fix, differential GPS mode 3: GPS PPS Mode, fix valid 4: Real Time Kinematic. System used in RTK mode with fixed integers 5: Float RTK. Satellite system used in RTK

			mode. Floating integers 6: Estimated (dead reckoning) Mode 7: Manual Input Mode 8: Simulator Mode
7	Satellites Used	11	Number of satellites in use, (00 ~ 12)
8	HDOP	0.8	Horizontal dilution of precision, (00.0 ~ 99.9)
9	Altitude	108.2	mean sea level (geoid), (-9999.9 ~ 17999.9)
10	DGPS Station ID	0000	Differential reference station ID, 0000 ~ 1023 NULL when DGPS not used
11	Checksum	02	

CHAPTER 3

3.1 Global System for Mobile Communications

The GSM standard (Global System for Mobile Communications) for mobile telephony was introduced in the mid-1980s and is the European initiative for creating a new cellular radio interface . The GSM system uses a TDMA radio access system employed in 135 countries, operating in 200 KHz channels with eight users per channel. Mobile services based on GSM technology were first launched in Finland in 1991. Today, more than 690 mobile networks provide GSM services across 213 countries and GSM represents 82.4% of all global mobile connections. According to GSM World, there are now more than 2 billion GSM mobile phone users worldwide. GSM World references China as "the largest single GSM market, with more than 370 million users, followed by Russia with 145 million, India with 83 million and the USA with 78 million users."

Since many GSM network operators have roaming agreements with foreign operators, users can often continue to use their mobile phones when they travel to other countries. SIM cards (Subscriber Identity Module) holding home network access configurations may be switched to those will metered local access, significantly reducing roaming costs while experiencing no reductions in service. GSM, together with other technologies, is part of the evolution of wireless mobile telecommmunications that includes High-Speed Circuit-Switched Data (HSCSD), General Packet Radio System (GPRS), Enhanced Data GSM Environment (EDGE), and Universal Mobile Telecommunications Service (UMTS).

3.2 GSM Bandwidth Allocation

GSM can operate four distinct frequency bands:

GSM 450: GSM 450 supports very large cells in the 450 MHz band. It was designed for countries with a low user density such as in Africa. It may also replace the original 1981NMT 450 (Nordic Mobile Telephone) analog networks used in the 450 MHz band. NMT is a first generation wireless technology.

GSM 900: When speaking of GSM, the original GSM system was called GSM 900 because the original frequency band was represented by 900 MHz. To provide additional capacity and to enable higher subscriber densities, two other systems were added afterward:
GSM 1800: GSM 1800 (or DCS 1800) is an adapted version of GSM 900 operating in the 1800MHz frequency range. Any GSM system operating in a higher frequency band requires a large number of base stations than for an original GSM system. The availability of a wider band of spectrum and a reduction in cell size will enable GSM 1800 to handle more subscribers thanGSM900. The smaller cells, in fact, give improved indoor coverage and low power requirements.

GSM 1900 (or PCS 1900): PCS 1900 (Personal Communications System) is a GSM 1800 variation designed for use on the North American Continent, which uses the 1900 MHz band.
Since 1993, phase 2 of the specifications has included both the GSM 900 and DCS 1800 (Digital Cellular System) in common documents. The GSM 1900 system has been added to the IS-136 D-AMPS (Digital Advanced Mobile Phone System) and IS-95 Code Division Multiple Access (CDMA) system, both operated at the 1900 MHz band.
The ITU (International Telecommunication Union) has allocated the GSM radio spectrum with the following bands:

- GSM 900: Uplink: 890–915 MHz
  Downlink: 935–960 MHz
- GSM 1800: Uplink: 1710–1785 MHz
  Downlink: 1805–1880 MHz
- GSM 1900: Uplink: 1850–1910 MHz
  Downlink: 1930–1990 MHz

In the above, uplink designates connection from the mobile station to the base station and downlink denotes connection from the base station to the mobile station.

3.3 GSM System Architecture

A cell containing a Mobile Station (MS) is formed by the radio coverage area of a Base Transceiver Station (BTS). Several BTSs together are controlled by one Base Station Controller (BSC). The BTS and BSC form the Base Station Subsystem (BSS). The combined traffic of the MSs in their respective cells is routed through the Mobile Switching Center (MSC). Several databases are required for call control and network management: the Home Location Register (HLR), the Visitor Location Register (VLR), the Authentication Center (AuC), and the Equipment Identity Register (EIR). The GSM system architecture comprised with a set of essential components is illustrated in Figure.

The GSM system network can be divided into three subgroups that are interconnected using standardized interfaces:

Mobile Station (MS)

Base Station Subsystem (BSS), and

Network Subsystem (NSS).

These subgroups are further comprised of the components in the following sections :

3.3.1 Mobile Station (SIM + ME)

The Mobile Station (MS) can refer to a handset or mobile equipment (ME). The Subscriber Identity Module (SIM) card in a GSM handset is a microprocessor smart card that securely stores various critical information such as the subscriber.s identity as well as the authentication and encryption algorithms responsible for providing legitimate access to the GSM network.

Each SIM card has a unique identification number called the International Mobile Subscriber Identity (IMSI). In addition, each MS is assigned to a unique hardware identification called theInternational Mobile Equipment Identity (IMEI).AnMS can also be a terminal (M-ES) that acts as a GSM interface, that is, for a laptop computer.

3.3.2 Base Station Subsystem (BSS)

The Base Station Subsystem (BSS) consists of the Base Transceiver Station (BTS) and the Base Station Controller (BSC). The BSS ensures transmission and management of radio resources.

Base Transceiver Station (BTS): The BTS is responsible for providing the wireless connection between the handset and the wireless network. The GSM uses a series of radio transceivers called BTSs that provide the points of entry to the GSM network. A BTS is comprised of a set of radio transmitters and receivers, and antennas to connect the mobile to a cellular network for pursuing the required call handling tasks. The BTS takes in the calls within its coverage zone and ensures their proper handling.

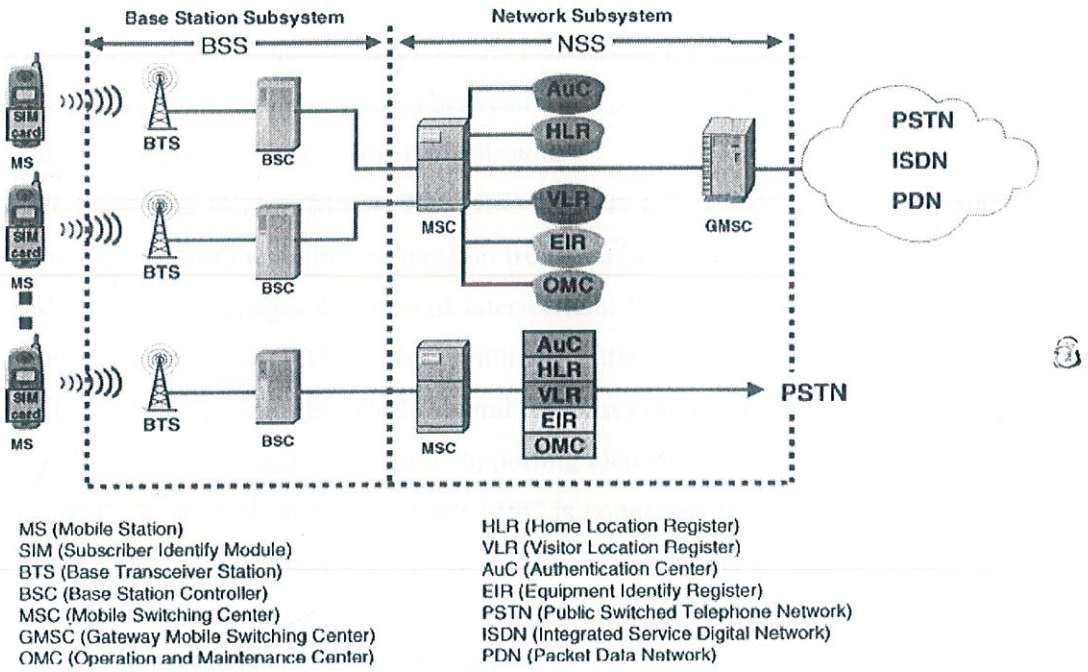

Fig 3.1: GSM System Architecture

Base Station Controller (BSC): The primary function of the BSC is call maintenance. As shown in Figure 1.1, the BSC manages the routing of communications from one or more base stations.ABSC controls a cluster of cell towers. It is responsible for setting up a voice or data call with the mobile terminal and managing handoff when the phone moves from one cell tower boundary to another, without disruption of service. In other words, the BSC manages radio resources and ensures the handover; that is, the passing of a subscriber from one cell to another with no degradation of the quality of the communication. The BSC also serves as the switch for concentration towards the Gateway Mobile Switching Center (GMSC).

3.3.3 Network Subsystem (NSS)

The Network Subsystem (NSS) is made up of the two essential elements, MSC and GMSC, long with its supporting elements: the Home Location Register (HLR), the Visitor Location Register (VLR), the Authentication Center (AuC), and the Equipment Identity Register (EIR). The NSS establishes communications between a cell phone and another MSC, and takes care of the Short Message Services (SMS) transmission.

Mobile Switching Center (MSC): The MSC controls call signaling and processing, and coordinates the handover of the mobile connection from one base station to another as the mobile roams around. The MSC manages the roles of inter-cellular transfer, mobile subscriber visitors, and interconnections with the PSTN. The combined traffic of the mobile stations in their respective cells is routed through the MSC. Several databases mentioned above are available for call control and network management. Those supporting elements include the location registers consisting of HLR, VLR, EIR, and AuC. Each MSC is connected through GMSC to the local Public Switched Telephony Network (PSTN or ISDN) to provide the connectivity between the mobile and the fixed telephone users. The MSC may also connect to the Packet Data Networks (PDN) to provide mobiles with access to data services.

Home Location Register (HLR): The NSS is assisted by HLRs. The HLR is a database used for management of the operator's mobile subscribers. For all users registered with a network operator, permanent data (the user's profile, subscriber's international identity number, and telephone number) and temporary data (the user's current location) are stored in the HLR. In the case of a call to a user, the HLR is always queried first regarding the user's current location.

The main information, stored in the HLR, concerns the location of each mobile station in order to route calls to the mobile subscribers managed by each HLR.

Visitor Location Register (VLR): The VLR is responsible for a group of location areas, and stores the data of those users who are currently in its area of responsibility. This may include the permanent user data that have been transmitted from the HLR to VLR for faster access. But the VLR may also assign and store local data such as a temporary identification. Concerning subscriber mobility, the VLR comes into play by verifying the characteristics of the subscriber and ensuring the transfer of location information. The VLR contains the current location of the MS and selected administrative information from the HLR. It is necessary for call control and provision of the services for each mobile currently located in the zones controlled by the VLR. AVLR is connected to one MSC and normally integrated into the MSC.s hardware.

Authentication Center (AuC): The AuC holds a copy of the 128-bit secret key that is stored in each subscriber's SIM card. These security-related keys are used for authentication and encryption over the radio channel.

Equipment Identification Register (EIR): The GSM distinguishes explicitly between the user and the equipment, and deals with them separately. The EIR registers equipment data rather than subscriber data. It is a database that contains a list of all valid mobile station equipments within the GSM network, where each mobile station is identified by its International Mobile Equipment Identity (IMEI). Thus, the IMEI uniquely identifies a mobile station internationally. The IMEI (a kind of serial number) is allocated by the equipment manufacturer and registered by the network operator who stores it in the EIR. The International Mobile Subscriber Identity (IMSI) identifies uniquely each registered user and is stored in the SIM. A mobile station can only be operated if a SIM with a valid IMSI is inserted into equipment with a valid IMEI.

Operating Subsystem (OSS) : The Operating SubSystem (OSS) constitutes the network Operation and Maintenance Center (OMC) as the operator's network management tool. Network Operation and Maintenance Center (OMC): The OMC is a management system, which oversees the GSM functional blocks. The OMC assists the network operator in maintaining satisfactory operation of the GSM network. The OMC is responsible for controlling and maintaining the MSC, BSC, and BTS.

3.4 GSM Transmission Network Architecture

The GSM transmission network architecture is depicted in Figure 1.2. It is also called the GSM protocol architecture used for the exchange of signaling messages pertaining to various functions of mobility, radio resource, and connection management and interface. The protocol layering consists of the physical layer (Layer 1), the data link layer (Layer 2), and the message management layer (Layer 3). Brief explanations for these items are described in the following sections.

3.4.1 Message Management Layer (Layer 3)

The GSM Layer 3 protocols are used for the communication of radio resource, mobility, code format, and call-related connection management between the various network entities involved. The Layer 3 protocol is made up of three sublayers called the radio resource (RR) (implemented over the link between the MS and the BSS), the mobility management (MM) (connecting between the MS and MSC), and the connection management (CM) (exchanging information with the peer) for providing the communications between the MS and MSC. Layer 3 also implements the message transport part (MTP) and the signaling connection control part (SCCP) of the CCITT SS7, on the link between the BSC and MSC (the A interface), to provide the transport and addressing functions for signaling messages belonging to the various calls routed through the MSC.

Radio Resource (RR) Management Sublayer : The RR management sublayer terminates at the BSS and performs the functions of establishing physical connections over the radio for the purpose of transmitting call-related signaling information such as the establishment of signaling and traffic channels between a specific mobile user and the BSS. The RR management functions are basically implemented in the BSS. The roll of the RR management sublayer is to establish and release stable connection between MSs and an MSC for the duration of a call, and to maintain it despite user movements. RR messages are mapped to the BSS Application Part (BSSAP) in the BSC. The BTS Management (BTSM) is used to transfer all OAM-related information to the BTS. The Message Transfer Part (MTP) and the Signal Connection Control Part (SCCP) are used to support the transfer of signaling messages between the MSC and the BSS. The SCCP is used to provide a referencing mechanism to identify a particular transaction relating to a particular call. The SCCP can also be used to enhance message routing, operation,

and maintenance information. The MTP is used between the BSS and the MSC. The MTP provides a mechanism for reliable transfer of signaling messages. The BSSAP provides the channel switching and aerial function, and performs the RR management and the interworking functions between the data link protocols used on the radio and the BSS-MSC side for transporting signaling-related messages.

Mobility Management (MM) Sublayer: The MM sublayer is terminated at the MSC and the relayed messages from or to the mobile station (MS) are relayed transparently in the BSS using the direct transfer application process (DTAP). These are procedures used to establish, maintain, and release aMM connection between the MS and the MSC, over which an entity of the CM sublayer can change information with its peer. The mobility management (MM) handles the control functions relating to the registration of subscriber location, paging, authentication, handover, and channel allocation. The MM messages are not interpreted by the BTS or the BSC. They are transferred over the A-bis interface as transparent messages and over the A interface using the DTAP.

Connection Management (CM) Sublayer: The CM sublayer terminates at the MSC and contains entities that consist of the call control (CC), the short message services (SMS), and the call-independent supplementary services (SS). An MM connection is initiated by a CM service request message, which identifies the requesting CM entity, and the type of service required of the MM connection. The MM connections provide services to the different entities of the upper CM sublayer. Once aMMconnection has been established, the CM can use it for information transfer, using the DTAP process. The DTAP process is used for the transparent transfer of MM/CM signaling messages between the MS and the MSC. The DTAP function provides the transport level protocol inter-working function for transferring Layer 3 signaling messages from and to the MS and to and from the MSC without any analysis of the message contents.

3.4.2 Data Link Layer (Layer 2)

The data link layer over the radio link, connecting the MS to the BSS, is based on a LAPDm or a modified LAPD (Link Access Protocol for the D channel). LAPDm uses no flags for frame delimitation; instead frame delimitation is done by the physical layer that defines the transmission frame boundaries.

LAPDm uses a length indicator field to distinguish the information carrying field used to fill the transmission frame. LAPDm uses an address field in its frame format to carry the service access point identifier (SAPI, 3 bits). When using command/response frames, the SAPI identifies the user for which a command frame is intended, and the user transmitting a response frame. Figure 1.3 shows the format for the address field. The link protocol discriminator (LPD, 2 bits) is used to specify the use of LAPDm. The C/R (1 bit) specifies a commandor response frame as used in LAPD. An extended address (EA, 1 bit) is used to extend the address field to more than one octet. The 1 bit unused field is reserved for future uses.

LAPDm uses a control field to carry the sequence number, and to specify the type of frame as used in LAPD. LAPDm specifies three types of frames for supervisory functions, unnumbered information transfer, and control functions as defined in LAPD. LAPDm uses no cyclic redundancy check bits for error detection. Error correction and detection schemes are provided by a combination of block and convolutional coding used in conjunction with bit interleaving in the physical layer. Signaling transport between the data link layer (Layer 2) on the radio side and the SS7 on BSS-MSC link is provided by a distribution data unit within the information field of SCCP. There are also parameters known as the discrimination parameter and the data link connection identifier (DLCI) parameter. The discrimination parameter (size: 1 octet) uses a single bit to address a message either to the DTAP or the BSSAP processes. The DLCI parameter (sized one octet) is made up of two subparameters that identify the radio channel type and the SAPI value in the LAPDm protocol used for the message on the radio link.

3.4.3 Physical Layer (Layer 1)

The physical layer on the radio link is based on a TDMA structure that is implemented on multiple access schemes. The multiple access scheme used in the GSM is a combination of FDMA and TDMA. The Conference of European Post and Telecommunications (CEPT) has made available for use two frequency bands for the GSM system: 890–915MHz for the uplink from the mobile station to the base station, and 935–960 MHz for the downlink from the base station to the mobile terminal.

Two types of channels are considered as traffic channels and control channels. The traffic channels are intended to carry encoded voice or user data, whereas the control channels are

intended to carry signaling and synchronization data between the base station and the mobile station. The CCITT SS7 MTPand SCCP protocols are used to implement both the data link and the Layer 3 transport functions for carrying the call control and mobility management (MM) signaling messages on the BSS-MSC link. The MM and CM sublayer signaling information from the mobile station is routed over signaling channels (such as the broadcast control channel (BCCH), the slowassociated control channel (SACCH), and the fast associated control channel (FACCH)) to the BSS from where they are transparently relayed through the DTAP process to an SCCP on the BSS-MSC link for transmission to the peer call control (CC) entity in the MSC for processing. Alternatively, any call signaling information initiated by the MSC on the SCCP connection is relayed through the DTAP process in the BSS to the assigned signaling channel, using the LAPDm data link protocol, for delivery to the mobile station.

3.5 Signaling Channels on the Air Interface

The signaling channels are used for call establishment, paging, call maintenance, and synchronization.

3.5.1 Broadcast Channels (BCHs)

The broadcast channels (downlink only) are mainly responsible for synchronization, frequency correction, and broadcast control.

Synchronization Channel (SCH): The SCH for downlink only will provide the MS with all the information (frame synchronization of the mobiles and identification of the BS) needed to synchronize with a BTS.

Frequency Correction Channel (FCCH): The FCCH for downlink only provides correction of MS frequencies and transmission of the frequency standard to the MS. It is also used for synchronization of an acquisition by providing the boundaries between timeslots and the position of the first time slot of a TDMA fame.

Broadcast Control Channel (BCCH): The BCCH broadcasts to all mobiles general information regarding its own cell as well as the neighboring cells, that is, local area code (LAC), network operator, access parameters, and so on. The MS receives signals via the BCCH from many BTSs within the same network and/or different networks.

3.5.2 Common Control Channels (CCCHs)

The common control channels are used in both downlinks and uplinks between the MS and the BTS. These channels are used to convey information from the network to MSs and provide access to the network.

The CCCHs include the following channels:

Access Grant Channel (AGCH): The AGCH is used on downlinks only for assignment of a dedicated channel (DCH) after a successful random access. The BTS allocates a traffic channel (TCH) or the stand-alone dedicated control channel (SDCCH) to the MS, thus allowing the MS access to the network.

Random Access Channel (RACH): The RACH is used in uplink only for random access attempts by the mobiles. It allows the MS to request an SDCCH in response to a page or due to a call.

Paging Channel (PCH): The PCH is used in downlink only for paging to mobiles. The MS is informed by the BTS for incoming calls via the PCH. Specifically, the paging message for mobiles is sent via the BSSAP to the BSS as a connectionless message through the SCCP/MTP. A single paging message transmitted to the BSS may contain a list of cells in which the page is to be broadcast. The paging messages received from the MSC are stored in the BTS and corresponding paging messages are transmitted over the radio interface at the appropriate time. Each paging message relates to one mobile station only and the BSS has to pack the pages into the relevant paging message. When a paging message is broadcast over the radio channel, if a response message is received from the mobile, the relevant signaling connection is set up towards the MSC and the page response message is passed to the MSC.

3.5.3 Dedicated Control Channel (DCCH)

The dedicated control channels are used on both downlinks and uplinks. The DCCHs are responsible for roaming, handovers, encryption, and so on. The DCCHs include the following channels:

Standalone Dedicated Control Channel (SDCCH): The SDCCH is a communication channel between the mobile (MS) and the BTS. This channel is used for the transfer of call control signaling to and from the mobile during call setup.

Slow Associated Control Channel (SACCH): The 26 multiframe is used to define traffic channels (TCH) and their slow and fast associated control channels (SACCH and FACCH) that carry link control information between the mobile and the base stations. The TCH has been defined to provide the following different forms of services: the full-rate speech or data channels supporting effective bit-rates of 13 Kbps (for speech), 2.4, 4.8, and 9.6Kbps (for data); and the half-rate channels with effective bit-rates of 6.5 Kbps (for speech), 1.2, 2.4, and 4.8Kbps for data. Like the TCHs, the SDCCH has its own slow associated control channel (SACCH) and is released once call setup is complete.

Fast Associated Control Channel (FACCH): The FACCH is obtained on demand by stealing frames from TCH and is used when a very fast exchange of information is needed such as a handover. The FACCH is used by either end for signaling the transfer characteristics of the physical path, or for other purposes such as connection handover control messages. The stealing of a TCH slot for FACCH signaling is indicated through a flag within the TCH slot.

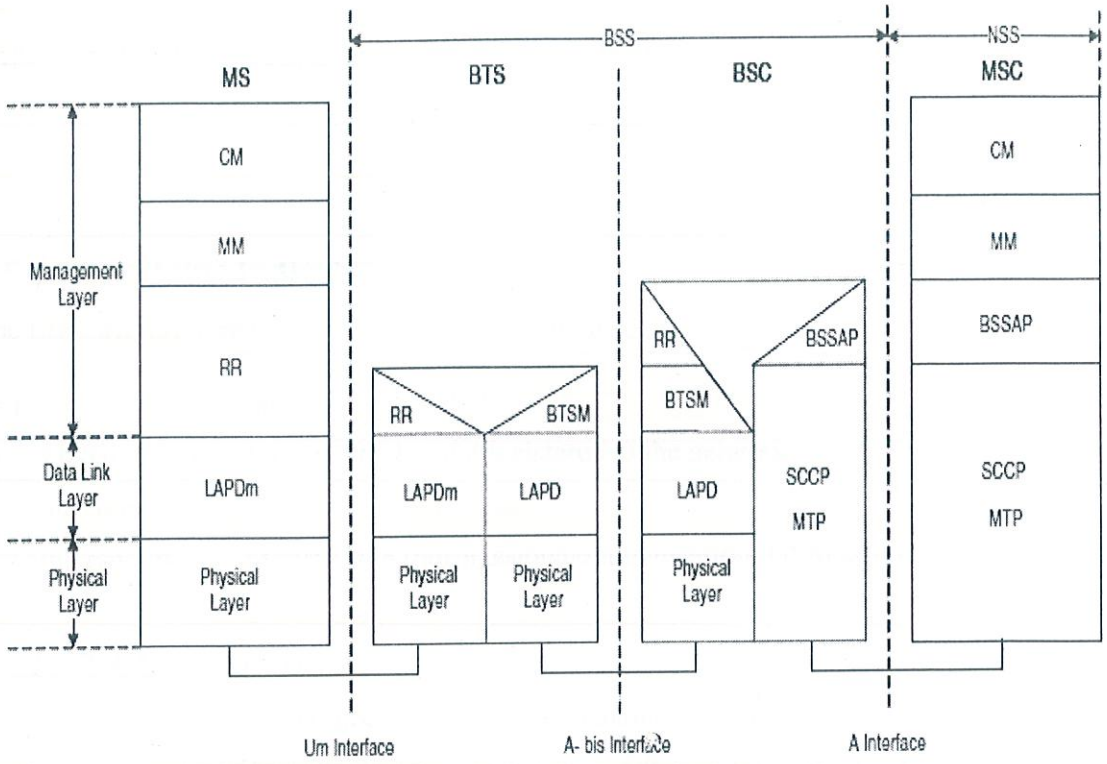

Fig 3.2: GSM Transmission Architecture

3.6 GSM MODEM

The GSM Trainer ST2133 is a modem or mobile equipment for transmission of voice and data calls as well as SMS (Short Message Service) in GSM Network.

To control the GSM modem there is an advanced set of AT commands according to GSM ETSI (European Telecommunications Standards Institute) 07.07 and 07.05 implemented. The GSM standard has established itself across continents. The trainer is well suited for studying AT commands by camping to real networks using SIM card.

Technical Specifications:

GSM capability : GSM 900 / 1800/850/1900 E – GSM

GSM data services : Asynchronous, Transparent & Non Transparent modes. 14.4 kbits / s

SIM Interface : 3 V

3.6.1 LED Status Indicator

The LED will indicate different status of the modem:

OFF	Modem Switched off
CONTINUOUS ON	Modem ON but not registered to the network
Flashing Slowly	Modem is in idle mode
Flashing rapidly	Modem is in transmission/communication (GSM only)

3.6.2 AT COMMANDS

To operate the GSM modem a serial link handler software such as HyperTerminal is needed to communicate with it. The serial link handler is set with the following default values:

Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Commands always start with AT (which means AT Attention) and finish with a <CR> character.

When command "AT" is sent to the GSM Trainer, it every time responses/acknowledges by "OK", can be used to detect connection.

If command syntax is incorrect, the "ERROR" string is returned.

SIM Insertion, SIM Removal

SIM card Insertion and Removal procedures are supported. There are software functions relying on positive reading of the hardware SIM detect pin. This pin state (open/closed) is permanently monitored. When the SIM detect pin indicates that a card is present in the SIM connector, the product tries to set up a logical SIM session. The logical SIM session will be set up or not depending on whether the detected card is a SIM Card or not.

The AT+CPIN? Command delivers the following responses:

If the SIM detect pin indicates "absent", the response to AT+CPIN? Is "+CME ERROR 10" (SIM not inserted).

If the SIM detect pin indicates "present", and the inserted Card is a SIM Card, the response to AT+CPIN? Is "+CPIN: xxx" depending on SIM PIN state.

If the SIM detect pin indicates "present", and the inserted Card is not a SIM Card, the response to AT+CPIN? Is CME ERROR 10.

These last two states are not given immediately due to background initialization. Between the hardware SIM detect pin indicating "present" and the previous results the AT+CPIN? Sends "+CME ERROR: 515" (Please wait, init in progress).

When the SIM detect pin indicates card absence, and if a SIM Card was previously inserted, an IMSI detach procedure is performed, all user data is removed from the product (Phonebooks, SMS etc.). The product then switches to **emergency mode**.

3.6.3 MESSAGE HANDLING COMMANDS

Since this project makes use of message sending and receiving, only message handling commands will be discussed.

Select message service +CSMS

Description: The supported services are originated (SMS-MO) and terminated short message (SMSMT) + Cell Broadcast Message (SMS-CB) services.

Syntax:

Command Syntax: AT+CSMS=<service>

COMMAND	POSSIBLE RESPONSES
AT+CSMS=0	+CSMS: 1,1,1 OK
AT+CSMS=1	+CSMS: 1,1,1

Preferred Message Format +CMGF

Description: The message formats supported are text mode and PDU mode. In PDU mode, a complete SMS Message including all header information is given as a binary string (in hexadecimal format). Therefore, only the following set of characters is allowed: {'0','1','2','3','4','5','6','7','8','9', 'A', 'B','C','D','E','F'}. Each pair or characters is converted to a byte (e.g.: '41' is converted to the ASCII character 'A', whose ASCII code is0x41 or 65). In Text mode, all commands and responses are in ASCII characters.

Syntax:

Command Syntax: AT+CMGF

COMMAND	POSSIBLE RESPONSES
AT+CMGF=0 Set PDU mode	OK

AT+CMGF=1	OK

Set TEXT mode

New message indication +CNMI

Description: This command selects the procedure for message reception from the network.

Syntax:

Command Syntax: AT+CNMI=<mode>,<mt>,<bm>,<ds>,<bfr>

COMMAND	**POSSIBLE RESPONSES**
AT+CNMI=2,1,0,0,0	OK AT+CMTI : "SM",1 Note: message received
AT+CNMI=2,2,0,0,0	OK +CMT"123456","98/10/01, 12:3000+00",129,4,32,240, "15379",129,5<CR><LF>

Read message +CMGR

Description: This command allows the application to read stored messages.

Syntax:

Command Syntax: AT+CMGR=<index>

A message read with status "REC UNREAD" will be updated in memory with the status "REC READ".

COMMAND	POSSIBLE RESPONSES
	AT+CMTI: "SM",1
AT+CMGR=1	+CMGR: "REC UNREAD","0146290800", "98/10/01,18:22 :11+00",<CR><LF> ABCdefGHI OK
	Note: read the message

Send message +CMGS

Description: The <address> field is the address of the terminal to which the message is sent. To send he message, simply type, <ctrl-Z> character (ASCII 26). The text can contain all existing characters except <ctrl-Z> and <ESC> (ASCII 27).

This command can be aborted using the <ESC> character when entering text.

In PDU mode, only hexadecimal characters are used ('0'…'9','A'…'F').

Syntax:

Command syntax in text mode:

AT+CMGS= <da> [,<toda>] <CR>

text is entered <ctrl-Z / ESC >

COMMAND	POSSIBLE RESPONSES
AT+CMGS="+33146290800"<CR>	+CMGS: <mr>
Please call me soon, Fred. <ctrl-Z>	OK
Note: Send a message in text mode	Note: Successful transmission

CHAPTER 4

RESULTS

To verify the results and ensure expected working of our circuit design, we simulated our circuit on PROTEUS ISIS PROFESSIONAL. In this software all the components as mentioned in previous literature were connected as shown in Figure 1.2. Since GPS receiver and GSM modem are also a part of our circuit design, to simulate their behavior ISIS provides a virtual terminal. It basically simulates an actual serial device in which the real settings of it can be made in order for it to behave like the actual serial device.

The settings made for the GPS receiver are shown in the figure 4.1. Similar procedure was adopted for settings of GSM modem.

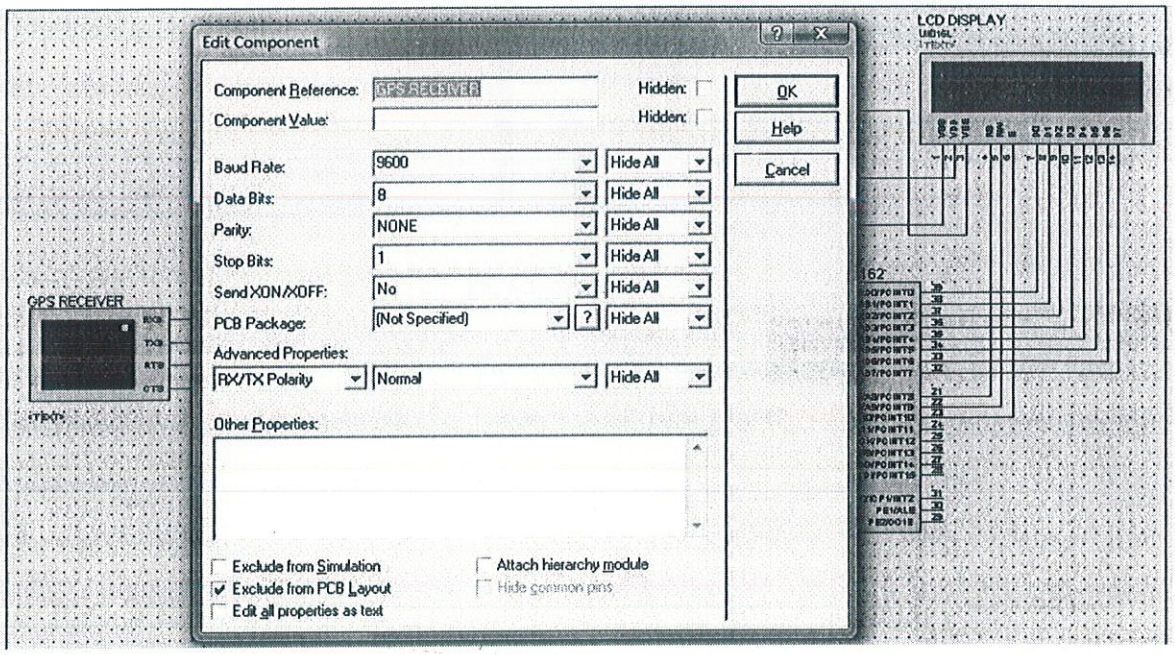

Fig 4.1: Snapshot-1

For the microcontroller ATmega162, the embedded C code was converted into .hex format with the use of AVR Studio 5. The various parameters of the microcontroller were set in ISIS and the .hex file was loaded as shown in the figure 4.2.

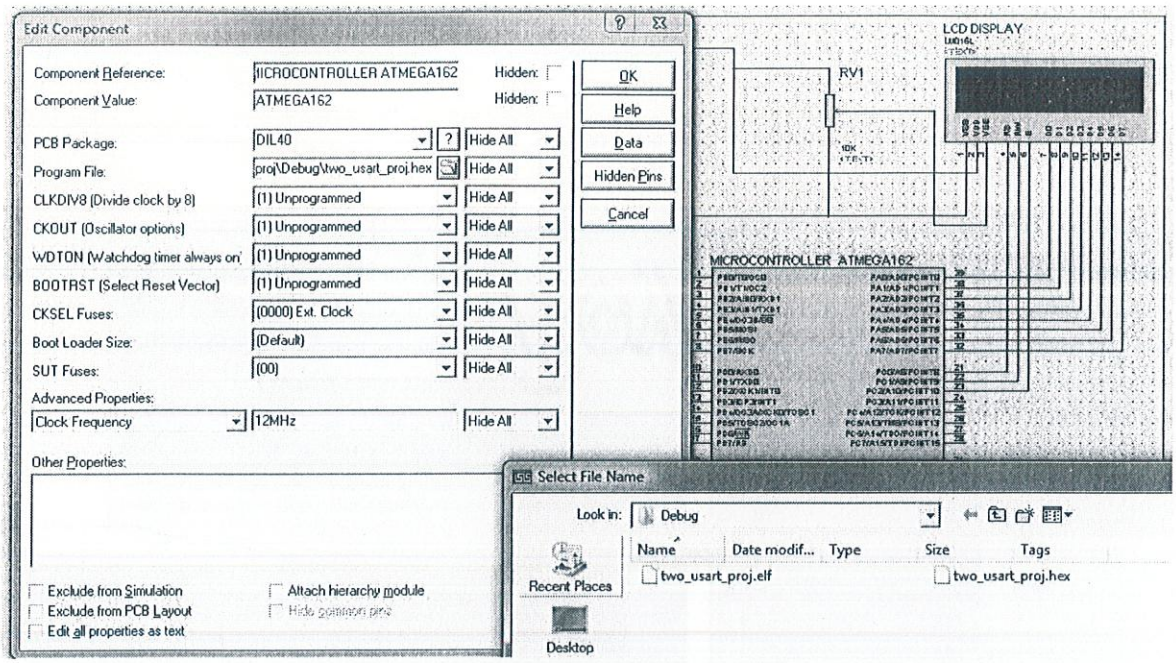

Fig 4.2: Snapshot-2

After doing all the required settings in the components the circuit was ready to be simulated. The various results of the simulation are shown in the next pages.

CIRCUIT SIMULATION

1. The circuit was now simulated. As it can be seen in the figure 4.3 , the LCD display is ON and prints "Loading..." on it as commanded in the C code. Also the virtual terminals for GPS receiver and GSM modem are displayed in which we can feed the necessary inputs and see the desired outputs.

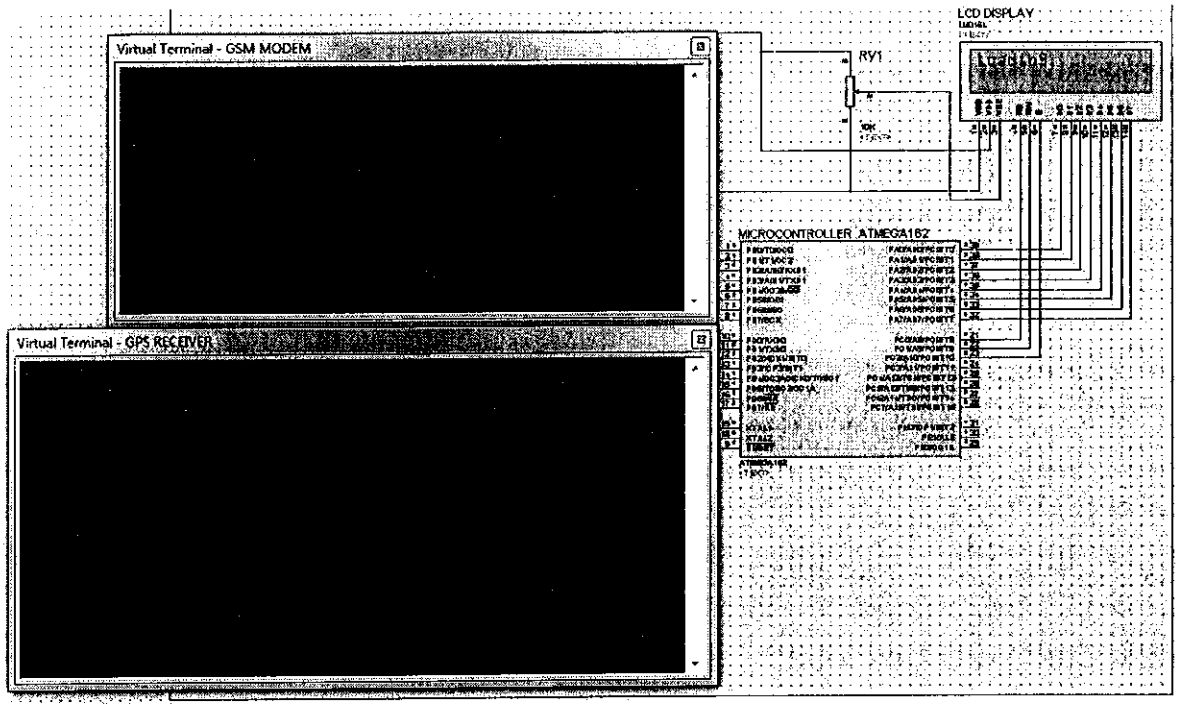

Fig 4.3: Snapshot-3

2. Now, the microcontroller is programmed to read only the $GPGGA sentence as received from the GPS receiver. In the figure 4.4 , it can be seen that $GPGGA sentence was fed into the virtual terminal of GPS receiver. The microcontroller extracts the required information from this sentence and displays the same on LCD display.

The sentence fed into the virtual terminal is:-

$GPGGA,132453.970,2651.0138,N,7547.7054,E,1,03,7.1,42.5,M,-42.5,M,0000*45

The required information as displayed on LCD is:-

2651.0138,N

7547.7054,E

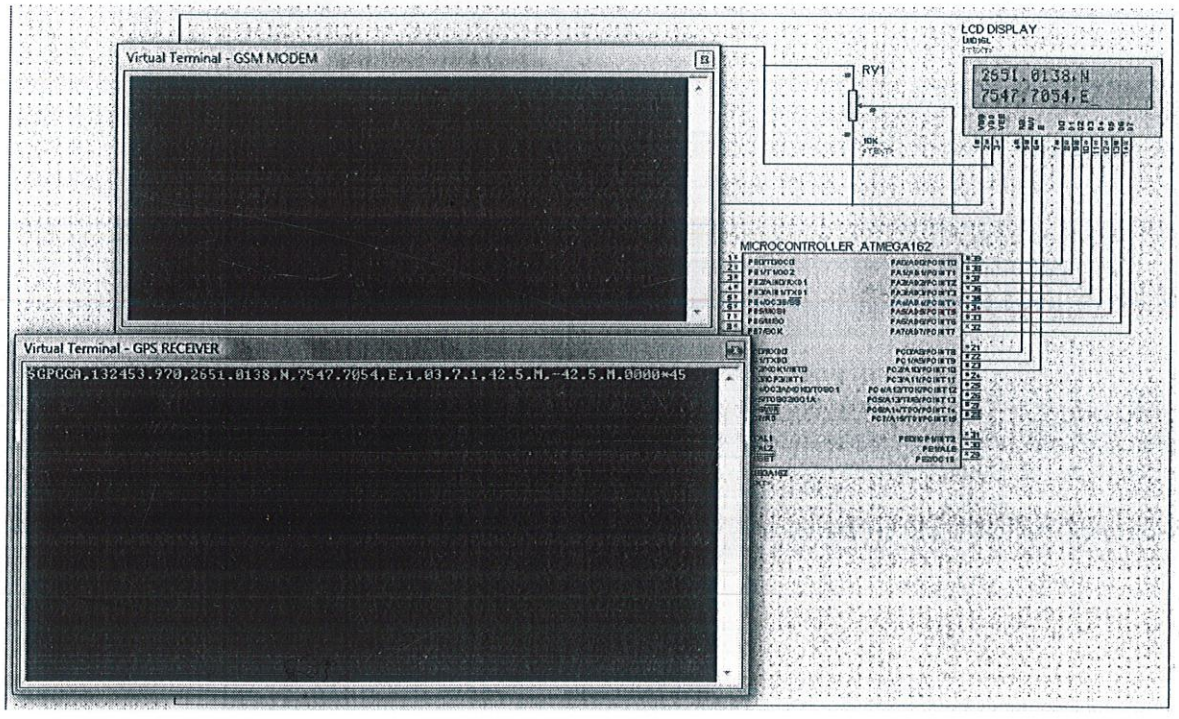

Fig 4.4: Snapshot-4

3. To be able to track a vehicle the user from a remote place can request the location of a vehicle through GSM modem. The microcontroller is programmed to send the position of the vehicle only if the GSM modem receives a valid character. In our code this character is 'r'.

The sending GSM modem is not a part of this simulation so to simulate the receiving process of the GSM modem connected to the microcontroller the character 'r' is fed into the virtual terminal. On receiving this character the microcontroller sends the position of the vehicle to the remote GSM modem in the format as shown in the figure 4.5.

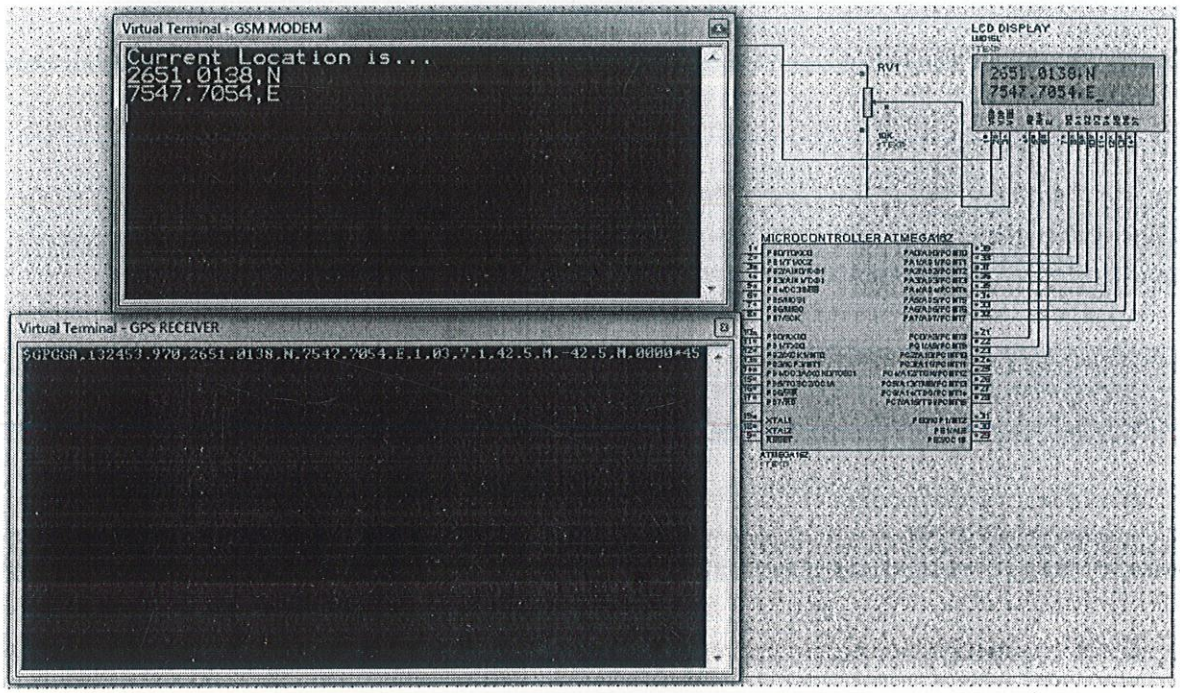

Fig 4.5: Snapshot-5

4. If the GSM modem does not receive the character 'r' and receives any other character the GSM modem sends the message "ERROR" to the remote GSM modem. The process is shown in the figure 4.6.

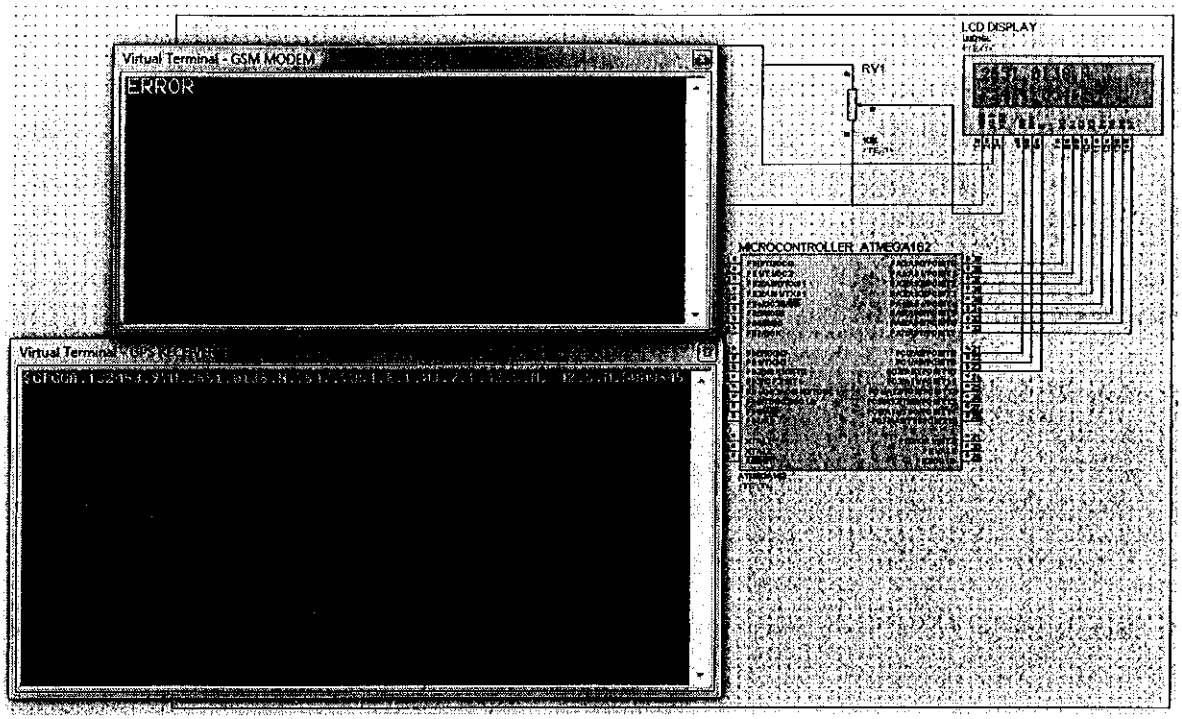

Fig 4.6: Snapshot-6

5. Now, since the vehicle is continuously moving, its position will change. The GPS receiver will then receive other $GPGGA sentence. Based on the new $GPGGA sentence the microcontroller will again extract the required information and send it to LCD which displays the information. Again the remote user can request the position of the vehicle by sending the character 'r' and the GSM modem will send the new information to the remote user as shown in the figure 4.7.

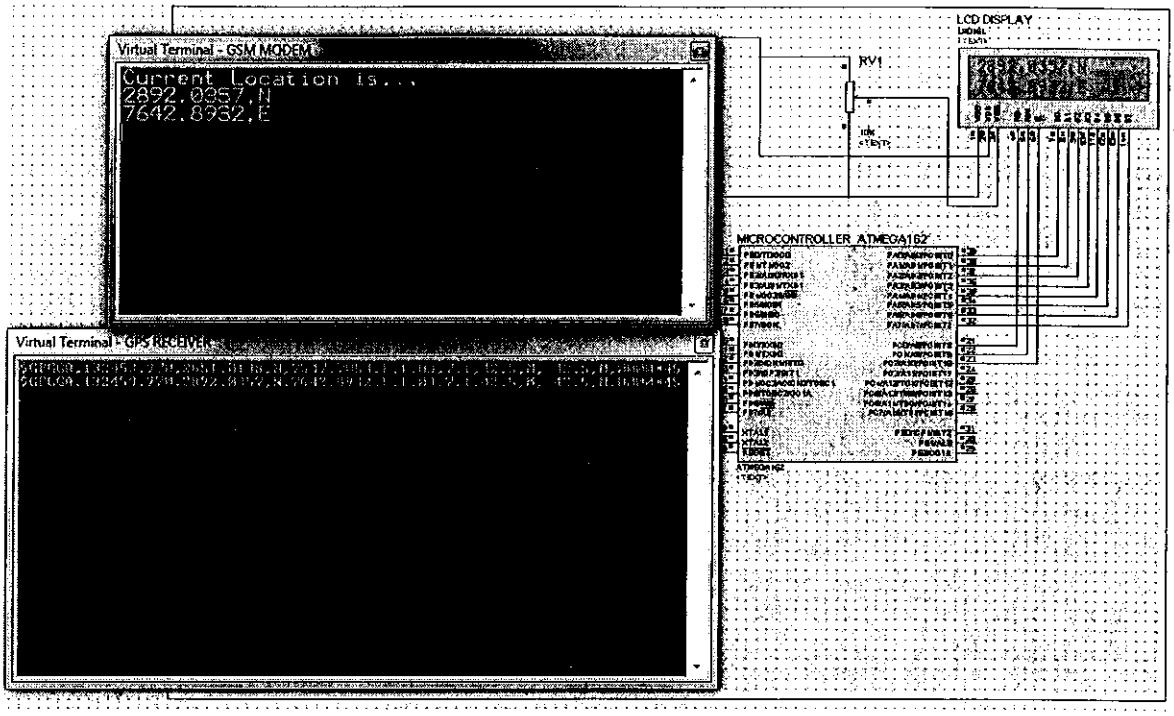

Fig 4.7: Snapshot-7

CHAPTER 5

CONCLUSION AND FUTURE WORK

Real time vehicle tracking system is successfully implemented using GPS and GSM technologies. Currently, the Vehicle Position Unit is implemented using separate GPS Receiver and GSM Modem. It can be replaced by a single GPS/GSM unit. Currently RS232 level converter circuit is used which will not be required when microcontroller will be directly interfaced to a single GPS/GSM unit.

Our future work includes design of tracking server (TS) which will maintain a database of all the vehicles fitted with VPUs. This database will be accessible from internet to authorized users through a web interface. Authorized users can track their vehicle and view all of the legitimate information stored in the database. TS will have a GSM modem attached to it that receives SMS messages from VPUs and sends them to the server through serial port.

Vehicle's information sent by VPU through SMS is received by this modem on TS. TS can also send commands for VPUs using this modem. Server can communicate with modem using AT commands. Database will be designed to store all of the relevant information of vehicles, such as Vehicle Position units and users of the system. To display this information to the end-users a front end application/software is required. End-user is the user of the system who has installed the VPU in his/her vehicle. The administrator of the system, who is responsible for all the activities going on in the tracking system, also requires access to this information. This will be achieved by developing a web interface for end-users and administrator.

Bibliography/References

1. "A Guide To The Global Positioning System (GPS) — GPS Timeline". *Radio Shack*

2. The global positioning system. American Institute of Aeronautics and Astronautics. *Parkinson; Spilker (1996).*

3. An Introduction to GSM. Artech House *Redl, Siegmund M.; Weber, Matthias K. Oliphant, Malcolm W (February 1995).*

4. GPS Engine Board -MR-87(MTK) specification

5.Programming And Customizing The AVR Microcontroller *by Dhananjay Gadre*

6. Avr Microcontroller and Embedded Systems: Using Assembly and C *by Muhammad Ali Mazidi,Janice Mazidi,Sarmad Naimi*

7. ATMEL ATmega162 Datasheet

8. ATMEL Atmega16 Datasheet

9. MAXIM MAX232 Datasheet

10. ST2133 GSM Trainer User Manual

Appendix

EMBEDDED C CODE

```
#define F_CPU 12000000UL
#include<avr/io.h>
#include<avr/interrupt.h>
#include<util/delay.h>

#define USART_BAUDRATE 9600
#define BAUD_PRESCALE 77

#define LCD_DATA PORTA                  //LCD data port

#define ctrl PORTC
#define en DDC2                         //enable signal
#define rw DDC1                         //read/write signal
#define rs DDC0                         //register select signal

void LCD_cmd(unsigned char cmd);
void init_LCD(void);
void LCD_write(unsigned char data);
void LCD_write_string(unsigned char *str);
void usart_init();
unsigned char usart_getch();
unsigned char usart_getch1();

void LCD();
void putdata();
void usart_init1();
void usart_putstr1(char *s);
void usart_putch1(unsigned char send);

unsigned char value,i,lati_value[11],lati_dir, longi_value[15], longi_dir, alti[5],take;

int main(void)
{
        DDRC=0xff;          //ctrl as out put
        0<<DDB2;
        1<<DDB3;
        init_LCD();         //initialization of LCD
        _delay_ms(1);
        LCD_write_string("Loading...");
        LCD_cmd(0xC0);
```

```
        usart_init();
        usart_init1();
        sei();
        while(1)
        {
        LCD();
        }

}

void init_LCD(void)
{
        LCD_cmd(0x38);          //initialization of 16X2 LCD in 8bit mode
        _delay_ms(1);

        LCD_cmd(0x01);          //clear LCD
        _delay_ms(1);

        LCD_cmd(0x0E);          //cursor ON
        _delay_ms(1);

        LCD_cmd(0x80);          // go to first line and 0th position
        _delay_ms(1);
        return;
}


void LCD_cmd(unsigned char cmd)
{
        LCD_DATA=cmd;
        ctrl =(0<<rs)|(0<<rw)|(1<<en);
        _delay_us(40);
        ctrl =(0<<rs)|(0<<rw)|(0<<en);
        _delay_ms(50);
        return;
}


void LCD_write(unsigned char data)
{
        LCD_DATA= data;
        ctrl = (1<<rs)|(0<<rw)|(1<<en);
        _delay_us(40);
        ctrl = (1<<rs)|(0<<rw)|(0<<en);
        _delay_ms(50);
        return ;
```

```
}


void usart_init()
{
UBRR0H =(BAUD_PRESCALE >> 8); // Load upper 8-bits of the baud rate value into the high
                                byte of the UBRR register
UBRR0L = BAUD_PRESCALE;       // Load lower 8-bits of the baud rate value into the low
                                byte of the UBRR register

UCSR0B = (1 << RXEN0) | (1 << TXEN0);  // Turn on the transmission and reception circuitry
UCSR0C = (1 << URSEL0) | (1 << UCSZ00) | (1 << UCSZ01);    // Use 8-bit character sizes

}


unsigned char usart_getch()
{

while (!(UCSR0A & (1 << RXC0))); // Do nothing until data have been received and is ready to
                                   be read from UDR
return(UDR0);                    // return the byte
}

void LCD_write_string(unsigned char *str)        //take address vaue of the string in pionter *str
{
        int i=0;
        while(str[i]!='\0')                       // loop will go on till the NULL characters is
                                                    seen in string
        {
        LCD_write(str[i]);                        // sending data on CD byte by byte
        i++;
        }
        return;
}

void usart_init1()
{
        UBRR1H = (BAUD_PRESCALE >> 8); // Load upper 8-bits of the baud rate value into
                                         the high byte of the UBRR register
        UBRR1L = BAUD_PRESCALE;        // Load lower 8-bits of the baud rate value into
                                         the low byte of the UBRR register
        UCSR1B = (1 << RXEN1) | (1 << TXEN1) | (1<<RXCIE1);  // Turn on the transmission
                                                               and reception circuitry
        UCSR1C = (1 << URSEL1) | (1 << UCSZ10) | (1 << UCSZ11); // Use 8-bit character
                                                                  sizes
```

```
}

void usart_putch1(unsigned char send)
{
        while(!(UCSR1A & (1<<UDRE1)));
        UDR1=send;
}

ISR(USART1_RXC_vect)
{
        i=0;
        take=UDR1;
        if(take=='r')
        {
                usart_putch1('\f');
                char put[] ="Current Location is...";
                while(i!=23)
                {usart_putch1(put[i]);
                i++;}
                putdata();
        }
        else
                puterror();
}

void putdata()
{

        usart_putch1('\r');
        i=0;
        while(lati_value[i]!=0)
        {
                        usart_putch1(lati_value[i]);
                        i++;
        }

        usart_putch1(lati_dir);
        usart_putch1('\r');
        i=0;
        while(longi_value[i]!=0)
        {
                                usart_putch1(longi_value[i]);
                                i++;
        }
        usart_putch1(longi_dir);
        usart_putch1('\r');
```

```
}

void puterror()
{
        usart_putch1('\f');
        i=0;
        char err[]="ERROR";
        while(i!=5)
        {
                usart_putch1(err[i]);
                i++;
        }
}

void LCD()
{

                value=usart_getch();
                if(value=='$')
                {
                        value=usart_getch();
                        if(value=='G')
                        {
                                value=usart_getch();
                                if(value=='P')
                                {
                                        value=usart_getch();
                                        if(value=='G')
                                        {
                                                value=usart_getch();
                                                if(value=='G')
                                                {
                                                        value=usart_getch();
                                                        if(value=='A')
                                                        {
                                                                value=usart_getch();
                                                                if(value==',')
                                                                {
                                                                        value=usart_getch();
                                                                        while(value!=',')
                                                                        {
                                                                                value=usart_getch();
                                                                        }
                                                                        lati_value[0]=usart_getch();
                                                                        value=lati_value[0];
                                                                        for(i=1;value!=',';i++)
```

```
{

  lati_value[i]=usart_getch();
        value=lati_value[i];
}
lati_dir=usart_getch();
value=usart_getch();
while(value!=',')
{
        value=usart_getch();
}

longi_value[0]=usart_getch();
value=longi_value[0];
for(i=1;value!=',';i++)
{

longi_value[i]=usart_getch();
        value=longi_value[i];
}
longi_dir=usart_getch();
LCD_cmd(0x01);
_delay_ms(1);
LCD_cmd(0x80);
_delay_ms(1);
i=0;
while(lati_value[i]!='\0')
{

    LCD_write(lati_value[i]);
    i++;
}
LCD_write(lati_dir);
LCD_cmd(0xC0);
_delay_ms(1);
i=0;
while(longi_value[i]!='\0')
{

  LCD_write(longi_value[i]);
  i++;
}
LCD_write(longi_dir);
_delay_ms(1);
                                        }
}       }       }       }       }       }       }
```