T. Y. B. Tech (ECE)

Semester: VI **Subject:** ESD & RTOS

Name: Satej Zunjarrao **Division: B**

Roll No: PB-30 **Batch: B3**

Experiment No: 06

Name of the Experiment: Interfacing GSM/GPS with LPC 2148

Performed on:

Submitted on:

Marks	Teacher's Signature with date

Aim: Write Embedded C program for interfacing GSM/GPS with LPC 2148.

Part List:

- Educational practice board for ARM7 LPC2148
- +9V power supply
- PC
- A to B type USB cable
- DB9 serial cable
- Eclipse IDE
- Flash Magic Utility
- GPS Module

Theory:

GPS receivers are found in most smartphones, many new automobiles, and they are used to track commerce all over the globe. These tiny devices can instantaneously give exact position and time, almost anywhere on the planet. GPS receivers are getting less expensive and smaller every day.

• Dozens of GPS satellites, all containing extremely accurate atomic clocks, have been launched since the late 70's, and launches continue to this day. The satellites continuously send data down to earth over dedicated RF frequencies. Our pocket-sized GPS receivers have tiny processors and antennas that directly receive the data sent by the satellites and compute your position and time on the fly.

• **Working**

GPS receivers use a constellation of satellites and ground stations to compute position and time almost anywhere on earth.

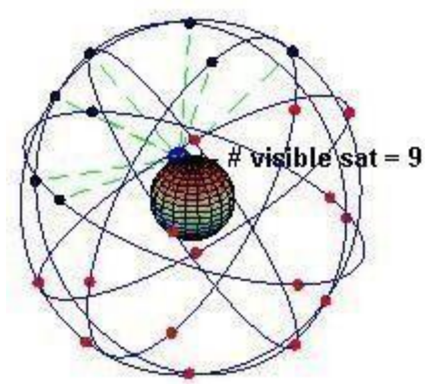


Figure 1: A space based satellite navigation system

• At any given time, there are at least 24 active satellites orbiting over 12,000 miles above earth. The positions of the satellites are constructed in a way that the sky above your location will always contain at most 12 satellites. The primary purpose of the 12 visible satellites is to transmit information back to earth over radio frequency (ranging from 1.1 to 1.5 GHz). With this information and some math, a ground based receiver or GPS module can calculate its position and time.

• **How does a GPS receiver calculate its position and time?**

The data sent down to earth from each satellite contains a few different pieces of information that allows your GPS receiver to accurately calculate its position and time. An important piece of equipment on each GPS satellite is an extremely accurate atomic clock. The time on the atomic clock is sent down to earth along with the satellite's orbital position and arrival times at different points in the sky. In other words, the GPS module receives a time stamp from each of the visible satellites, along with data on where in the sky each one is located (among other pieces of data). From this information, the GPS receiver now knows the distance to each satellite in view. If the GPS receiver's antenna can see at least 4 satellites, it can accurately calculate its position and time. This is also called a lock or a fix.

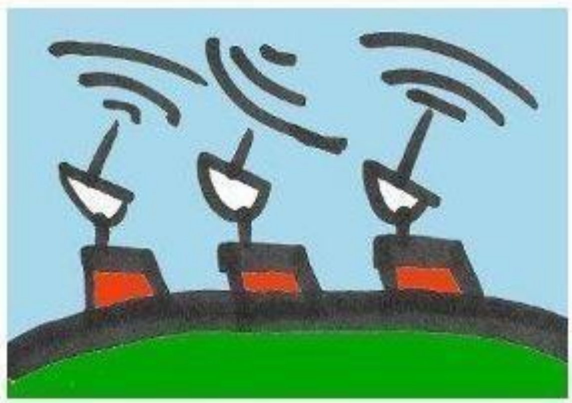


Figure 2: GPS receivers use a constellation of satellites

Along with satellites and GPS receivers, there are ground based stations that can communicate with the satellite network and some GPS receivers. This system is formally called the control segment and increases the accuracy of your GPS receiver. Common systems that use the control segment to improve accuracy are WAAS and DGPS. WAAS is common on most GPS receivers and improves accuracy to about 5 meters. DGPS requires a specific type of GPS receiver and gets centimeter accuracy. DGPS units are also expensive and tend to be larger because they require an additional antenna.

- **GPS Accuracy**

GPS Accuracy depends on a number of variables, most notably signal to noise ratio (noisy reception), satellite position, weather and obstructions such as buildings and mountains. These factors can create errors in your perceived location. Signal noise usually creates an error from around one to ten meters. Mountains, buildings and other things that might obstruct the path between the receiver and the satellite can cause three times as much error as signal noise. A GPS receiver must be able to get a lock on 4 satellites to be able to solve for a position. The first lock it gets allows the receiver to obtain the almanac information and thus what other satellites it should listen for. Although it is possible to get a position from less than 4 satellites, the margin of error of this position can be rather large. Your most accurate read of your location comes when you have a clear view of a clear sky away from any obstructions and under more than four satellites. To combat these errors, a couple of different assistants have been created.

- **Assisted GPS**

One of these ancillaries is Assisted GPS or AGPS. This method uses wireless (ground-based) networks to help relay between the satellite and the receiver when the GPS signal is weak or not able to be picked up. There are two ways AGPS can help out. The first is to provide the receiver with the proper almanac data and the precise time. The second utilizes the higher computing power and good satellite signal of the ground base to interpret the broken or fragmented information the receiver is receiving to provide a more accurate position reading to the receiver. AGPS is mostly accomplished by GPS receivers mounted on cellular towers. When communicating with these receivers, the GPS can acquire a lock on the satellite more quickly as well as receive more accurate information. This method is what is used for GPS in mobile phones and why they're sometimes more accurate than the GPS receivers on their own. But AGPS is present in more devices than just cellphones; it's even available in cameras and some vehicles. It's most beneficial in cities where the GPS signal may have a difficult time making it through the dense maze of the buildings.

Differential GPS

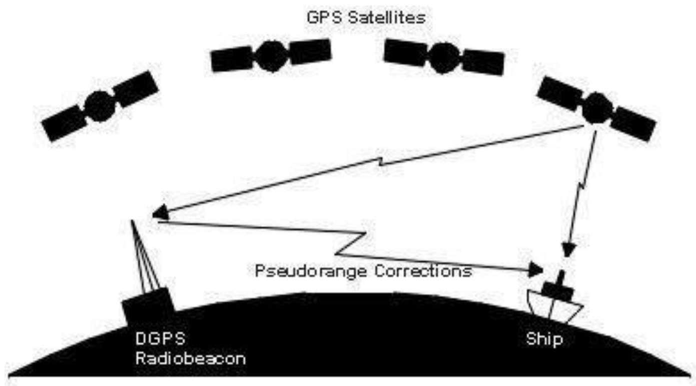


Figure 3: Differential GPS

• Another method is Differential GPS or DGPS. DGPS also uses ground or fixed GPS stations to determine the location, but differs in that it finds the difference between both the satellite and the ground location reading. These ground stations may be up to 200 nautical miles from the receiver, and it is important to note that accuracy deteriorates the further you are from the ground station. DGPS is accomplished by a ground station broadcasting a signal which dictates the error between the actual pseudo range and the measured pseudo range. This value is calculated by multiplying the speed of light by the time it takes the signal to travel from the satellite to the receiver. As an example, one form of DGPS is Wide Area Augmentation System or WAAS.

• Originally developed by the FAA to assist aircraft GPS, WAAS uses a system of specifically built ground stations. WAAS holds a specific set of accuracy standards that ground station measurements must meet. Laterally and vertically, WAAS must be accurate to within 7.6 meters 95% of the time. These ground stations send their measurements to master stations which send the corrections to WAAS satellites every 5 seconds or quicker. From the Satellite, a signal is broadcast back to the receivers on earth where the corrections are used to improve the GPS accuracy. In some locations, WAAS is able to provide an accuracy of 1 meter lateral and 1.5 meters vertically. While WAAS is only present in North America, similar systems are in place in many other parts of the world.

GPS Configuration: Message Formats

GPS data is displayed in different message formats over a serial interface. There are standard and non-standard (proprietary) message formats. Nearly all GPS receivers output National Marine Electronics Association (NMEA) data. The NMEA standard is formatted in lines of data called sentences. Each sentence contains various bits of data organized in comma delimited format (i.e. data separated by commas).

Example NMEA sentences from a GPS receiver with satellite lock (4+ satellites, accurate position):

$GPRMC,235316.000,A,4003.9040,N,10512.5792,W,0.09,144.75,141112,,*19

$GPGGA,235317.000,4003.9039,N,10512.5793,W,1,08,1.6,1577.9,M,-20.7,M,,0000*5F

$GPGSA,A,3,22,18,21,06,03,09,24,15,,,,,2.5,1.6,1.9*3E

In above example, the GPGGA sentence contains the following:

- Time: 235317.000 is 23:53 and 17.000 seconds in Greenwich mean time
- Longitude: 4003.9040, N is latitude in degrees. Decimal minutes, north
- Latitude: 10512.5792, W is longitude in degrees. Decimal minutes, west
- Number of satellites seen: 08
- Altitude: 1577 meters

The data is separated by commas to make it easier to read and parse by computers and microcontrollers. This data is sent out on the serial port at an interval called the update rate. Most receivers update this information once per second (1Hz), but more advanced receivers are capable multiple updates per second. 5 to 20Hz is possible with modern receivers.

Reading GPS data

Most GPS modules have a serial port, which makes them perfect to connect to a microcontroller or computer.

1. Connecting to a Microcontroller

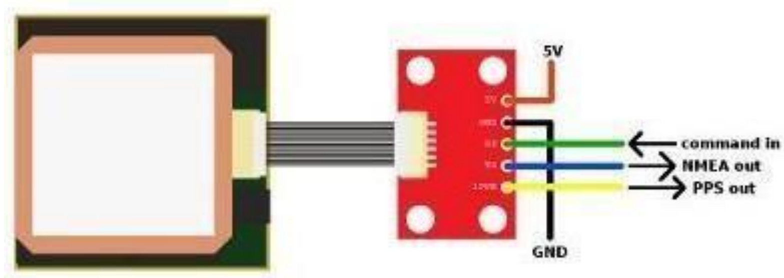


Figure 3: Connecting GPS to Microcontroller

Once a GPS module is powered, NMEA data (or another message format) is sent out of a serial transmit pin (TX) at a specific baud rate and update rate, even if there is no lock. To have your microcontroller read the NMEA data, all that is needed is to connect the TX pin of the GPS to the RX (receive) pin on the microcontroller. To configure the GPS module, you will need to also connect the RX pin of the GPS to the TX pin of the microcontroller.

It is common for the microcontroller to resolve (parse) the NMEA data. Parsing is simply removing the chunks of data from the NMEA sentence so the microcontroller can do something useful with the data.

For example, the microcontroller might need to read only the altitude of your GPS.$GPGGA,235317.000,4003.9039,N,10512.5793,W,1,08,1.6,1577.9,M,-20.7,M,,0000*5F

Instead of dealing with all of this text, the microcontroller can parse the GPGGA sentence and end up with only the altitude (in meters)1577.

Once the microcontroller can grab the data needed, the information can be manipulated to create other interactions on the microcontroller.

2. Connecting to a Computer

A simple way to see the NMEA data directly is to connect the GPS module to a computer. For the connections, all that is needed is to power the GPS with the FTDI basic (in this case 5V and GND), then connect the TX pin of the GPS to the RX pin on the FTDI Basic.

Next, open a serial terminal program at the same baud rate of your GPS module. Even if the GPS does not have a lock, you should see NMEA sentences steaming by.

$GPRMC,235316.000,A,4003.9040,N,10512.5792,W,0.09,144.75,141112,,*19

$GPGGA,235317.000,4003.9039,N,10512.5793,W,1,08,1.6,1577.9,M,-20.7,M,,0000*5F

$GPGSA,A,3,22,18,21,06,03,09,24,15,,,,,2.5,1.6,1.9*3E

Configuring a GPS receiver

To configure a GPS receiver, knowing the type of chipset your GPS is using is very important. The GPS chipset contains a powerful processor that is responsible for the user interface, all of the calculations, as well as analog circuitry for the antenna. The chipset also allows for data to be sent to the GPS receiver to configure parameters like, update rate, baud rate, sentence selection, etc.

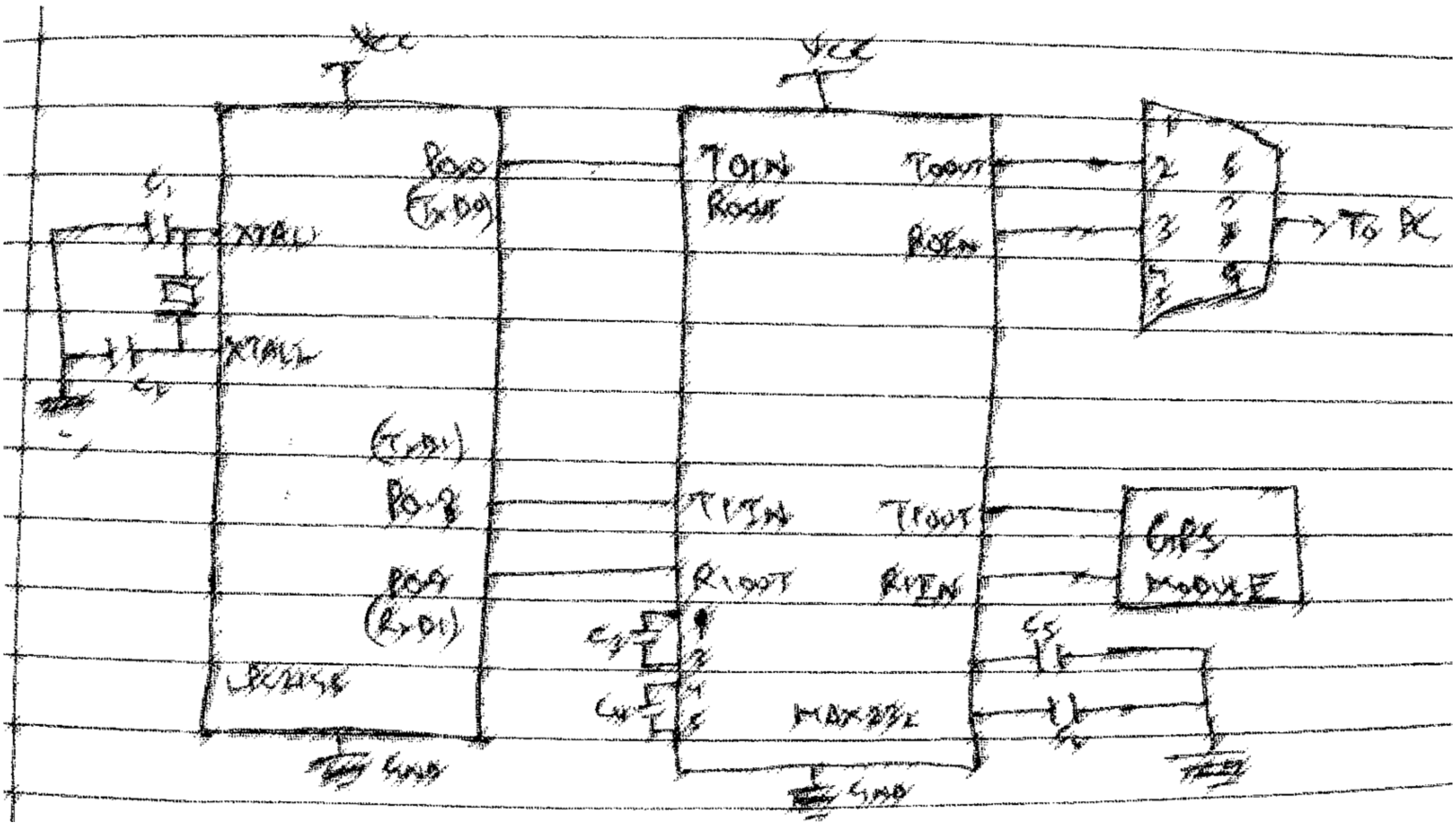
In order to send commands over a serial interface to a GPS receiver, you will need a command set or reference manual. In our case, we have used SkyTraq chipsets.

When communicating with a GPS receiver, most commands need to be terminated by a checksum. In most cases, you need to XOR each of your sentences.

Hardware Connection:

- Connect USB cable between PL3 connector of EPBARM7 and PC.
- Connect GPS modem to J10 connector of EPBARM7 board.

Interfacing Diagram:



Program:

```
//#include "gps.h"
//#include"delay.h"
#include "lpc214x.h"
//#include "uart.h"
void delay_fv(unsigned int x) // delay inducing loop
{
int i,j;
for(i=0;i<=x;i++)
for(j=0;j<=100;j++);

}
int main()
{
      int pclk=12000000, value;
      PINSEL0|=0x05;
      IO0DIR|=0x01;
      VPBDIV|=0x01;
      U0FCR=0x07; //FIFo control register - controls the operation of the uart rx and tx fifo's
      U0LCR=0x83; // line control register - determines the data format of the data character that is to be transmitted and
received
   U0FDR=0xC1; //fractional divider - controls the clk prescaler for thr baud rate genratio
      value=pclk/(16*115200);
      U0DLL=value & 0x0FF; //divisor latch lsb - holds the value used to divide the clk which came from fractional
prescaler
      U0DLM=(value & 0xFF00)>>8; // division latch msb - '' [ lsb hold the lower 8 bits and msb reg holds the upper/higher
8 bits which in total hold 16 bit divisor]
      U0TER=0x80;// transmit enable register
      U0LCR=0x03;// line control register - must be the one to acess the uart divisor latches

      //ConfigUart(Uart1,9600);
      PINSEL0|=0x05<<16;
      IO0DIR|=0x01<<8;
      // VPBDIV|=0x01;
      U1FCR=0x07;
      U1LCR=0x83;
            value=pclk/(16*9600);
      U1DLL=value & 0x0FF;
      U1DLM=(value & 0xFF00)>>8;
                  U1TER=0x80;
                  U1LCR=0x03;
//ConfigUart(Uart1,9600)
//ConfigUart(Uart0,115200);
                  delay_fv(500);
while(1)
          {
while(!(U1LSR & 0x01));
char data1= U1RBR;
U0THR=data1;
while(!(U0LSR & 0x20));

        }

}
```

Result:

```
$GPZDA,064413.000,18,04,2023,00,00*5C
$GPGGA,064414.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,0.0,M,,0000*6E
$GPGLL,0000.0000,N,00000.0000,E,064414.000,V,N*42
$GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,064414.000,V,0000.0000,N,00000.0000,E,000.0,000.0,180423,,,N*79
$GPVTG,000.0,T,,M,000.0,N,000.0,K,N*02
$GPZDA,064414.000,18,04,2023,00,00*5B
$GPGGA,064415.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,0.0,M,,0000*6F
$GPGLL,0000.0000,N,00000.0000,E,064415.000,V,N*43

$GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,064415.000,V,0000.0000,N,00000.0000,E,000.0,000.0,180423,,,N*78
$GPVTG,000.0,T,,M,000.0,N,000.0,K,N*02
$GPZDA,064415.000,18,04,2023,00,00*5A
$GPGGA,064416.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,0.0,M,,0000*6C
$GPGLL,0000.0000,N,00000.0000,E,064416.000,V,N*40
$GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
$GPGSV,1,1,02,14,00,000,17,17,00,000,26*7A
$GPRMC,064416.000,V,0000.0000,N,00000.0000,E,000.0,000.0,180423,,,N*7B
$GPVTG,000.0,T,,M,000.0,N,000.0,K,N*02
$GPZDA,064416.000,18,04,2023,00,00*59
$GPGGA,064417.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,0.0,M,,0000*6D
$GPGLL,0000.0000,N,00000.0000,E,064417.000,V,N*41
$GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,064417.000,V,0000.0000,N,00000.0000,E,000.0,000.0,180423,,,N*7A
$GPVTG,000.0,T,,M,000.0,N,000.0,K,N*02
$GPZDA,064417.000,18,04,2023,00,00*58
$GPGGA,064418.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,0.0,M,,0000*62
$GPGLL,0000.0000,N,00000.0000,E,064418.000,V,N*4E
$GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,064418.000,V,0000.0000,N,00000.0000,E,000.0,000.0,180423,,,N*75
$GPVTG,000.0,T,,M,000.0,N,000.0,K,N*02
$GPZDA,064418.000,18,04,2023,00,00*57
```

Connected 0:01:33 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Conclusion:

Interfaced GPS (S1612V8) module with LPC2148.

Name: Satep Zunjarrao

Div :- B
Batch :- B3
Roll no :- PB-30

Assignment - 06

Post Lab

Q.1] How Position and time is calculated

Ans:] GPS modules use a network of satellites in space to calculate the position and ~~tt~~ time information

- Position calculation

Once the distance at least three ~~sta~~ satellite are knew, the GPS receiver can be use brignometry to determine exat location.

This is done by interferring sphere around each stallite with a radius equal.

- Time calculation:

The addition to position information, GIS module also calculate the precise time. Since satellites transmit their the information in their signal, GPS receiver can use the time stamp in signals to determine accurate.

Q.2] Explain the various field in GPS message structure with suitable

Ans:] i) Message types:

The message type field indicate types of data contained in GPS module. It is typically represented by a 6-bit valued and can specify diffe types of data, such as ephemes data, data or other navigation related.

ii) Satellite ID:-

The satellite ID field identifie the specific GPS satelliete that is ~~th~~ transmitting the message:-

Ex "10" - transmitted at PRIV-10

iii) Subframe ID:-

The subframe ID field identifies the particular subframe within the GPS message structure.

iv) Navigabion Data:-

The navigation data fields contain the actual information related to GPS satellite ephemis, alma, clock correction, health status and ~~nag~~ navigabion.

v) Parity Bit:-

The parity Bit are error-checking bit that are used to ensure the integrity and accuracy of GPS message.