EDL – II

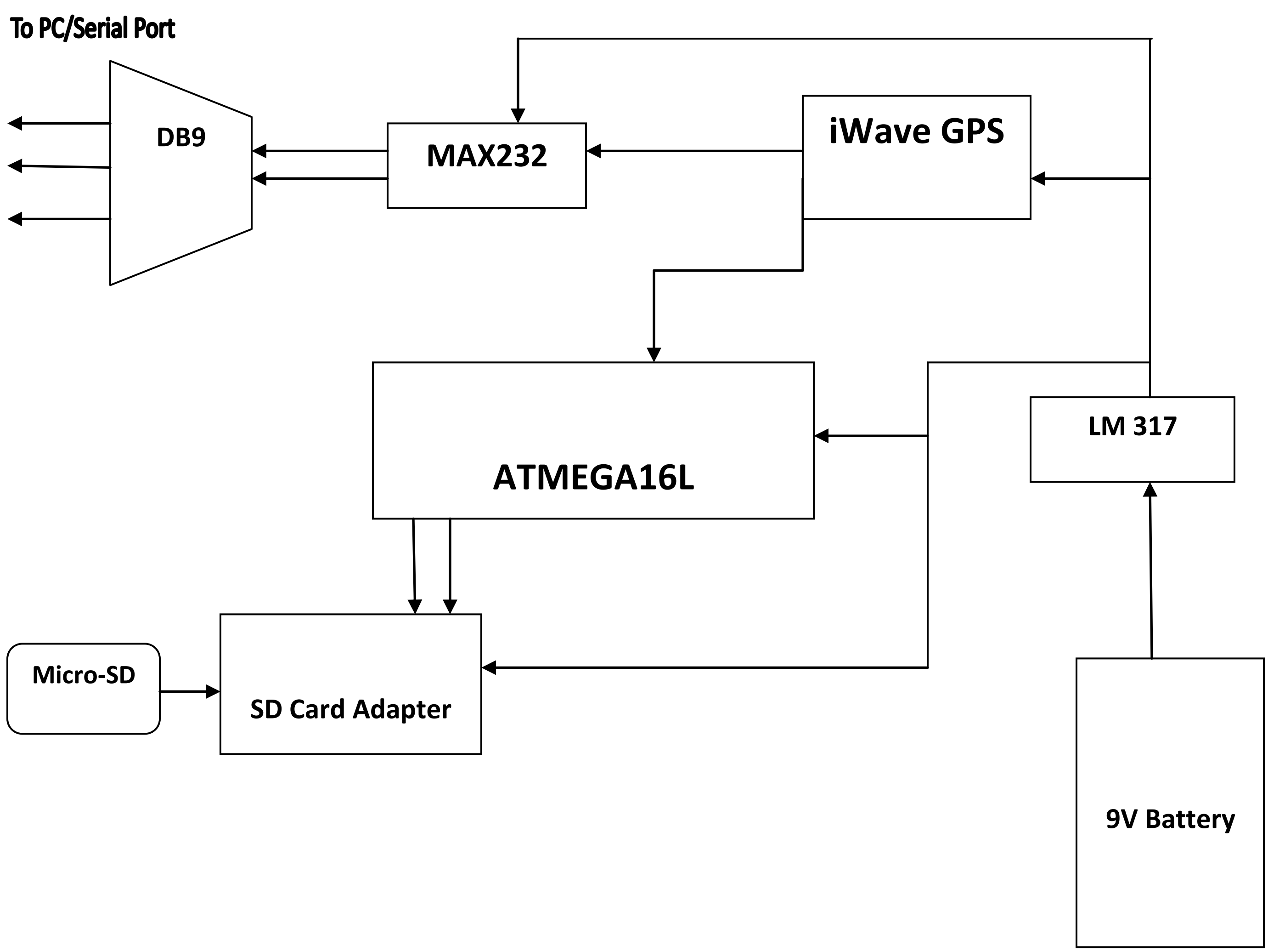
GPS Real-Time Navigation System with Data Storage

Group No. 1:
Shobhraj : 06D07039
Ashok : 06D07040
Santhosh : 06D07041

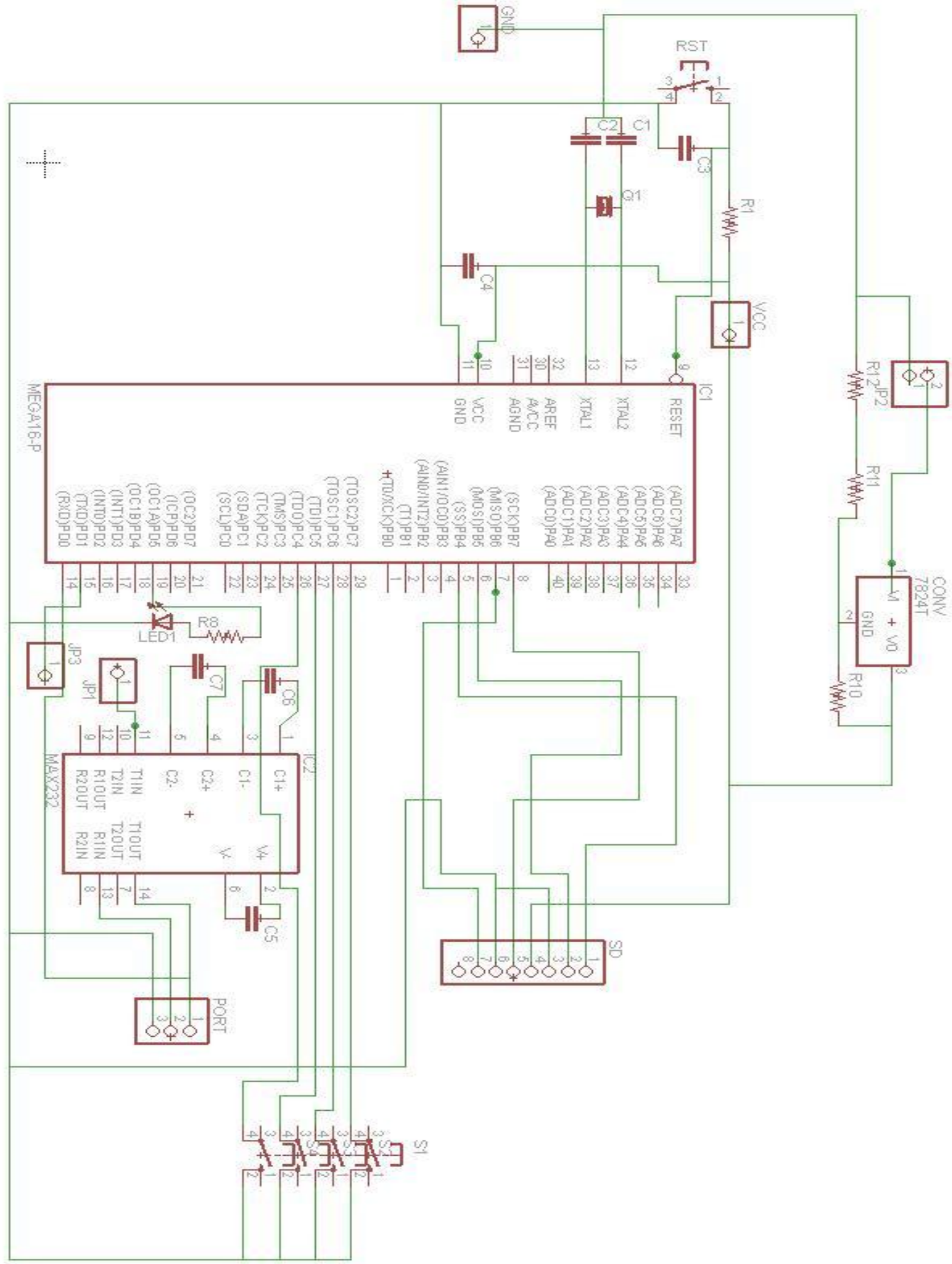
OBJECTIVE: To design a Global Positioning System (GPS) based navigation system with a real-time display using Google Maps Application Programming Interface (API) with real-time path storage into a Micro-SD Card.

MAJOR COMPONENTS: iWave GPS Module, Microcontroller ATMEGA16L, MAX232 Serial Driver/Receiver, Micro-SD Card (1 GB) and Adapter, LM 317 Voltage Regulator.

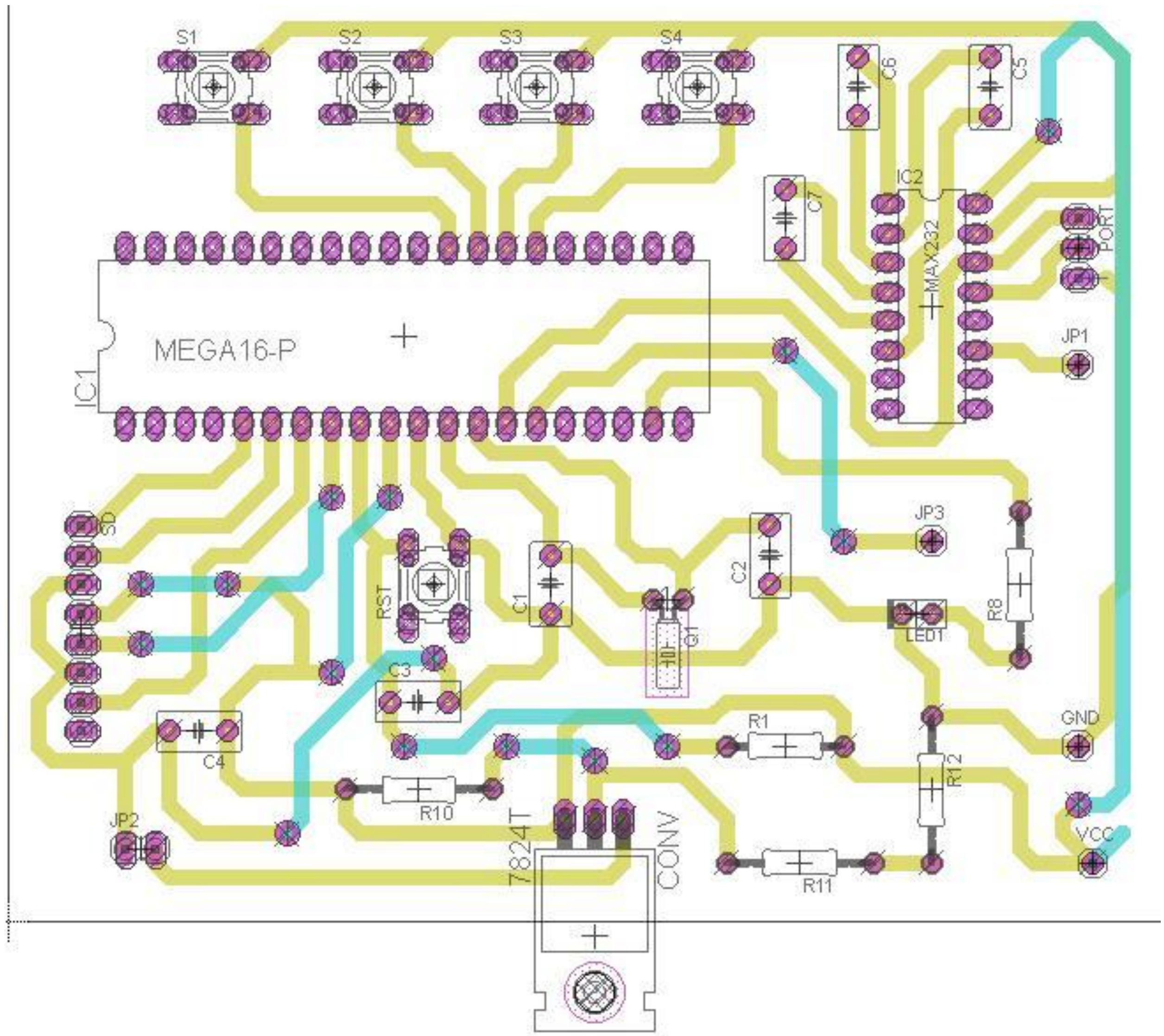
BLOCK DIAGRAM:



SCHEMATIC: LM 317 is wrongly shown as IC7824 in the schematic/layout.



LAYOUT:



BASIC IDEA: The power supply is a 9V battery, which is down converted to 3.3V using an LM 317 voltage regulator IC. As shown in the above block diagram, the GPS is connected to the microcontroller directly. The microcontroller obtains the input from the GPS using the UART and through the serial interface, writes the data into the Micro-SD card. The output of the GPS is also connected as an input to a MAX232 Driver/Receiver IC, whose output is connected using a DB-9 connector to the serial port of the PC. The default serial data transfer software in Windows, 'HyperTerminal' is used to store this data into a text file, which is in turn accessed by the saved Google Maps API and is used to plot the path in real-time. The data which is stored in the Micro-SD card can be used in the PC at a later time to plot the path on Google Maps.

iWAVE GPS MODULE: The Global Positioning System(GPS) refers to a US-based group of space- satellites which provides reliable positioning, navigation and timing service worldwide. The system consists of three parts, the actual satellites, a few control stations on earth and the devices that users own. The iWave GPS module is the navigation device we use to track our position. It has a 20-channel receiver system which enables fast acquisition/reacquisition of data. It has a high sensitivity (-159dBm) and accuracy (~10m) with a low power consumption (~110mW). It operates at 3.3V with a baud rate of 4800/9600Bd. The data is transferred using NMEA protocol. The module has an SMA connector with an active antenna (Gain: 28 +/- 2 dB). A patch antenna could also be connected using the available U.FL connector.

The output of the module looks something like this:

$GPGGA,062129.000,1907.9356,N,07255.0014,E,1,05,1.6,76.1,M,-62.8,M,,0000*47

$GPGSA,A,3,08,19,13,25,28,,,,,,,,3.9,1.6,3.6*35

$GPRMC,062129.000,A,5025.9356,N,09055.0014,E,0.10,170.47,150909,,,A*66

$GPVTG,170.47,T,,M,0.10,N,0.2,K,A*0B

$GPGGA,062130.000,1907.9357,N,07255.0014,E,1,05,1.6,76.1,M,-62.8,M,,0000*4E

$GPGSA,A,3,08,19,13,25,28,,,,,,,,3.9,1.6,3.6*35

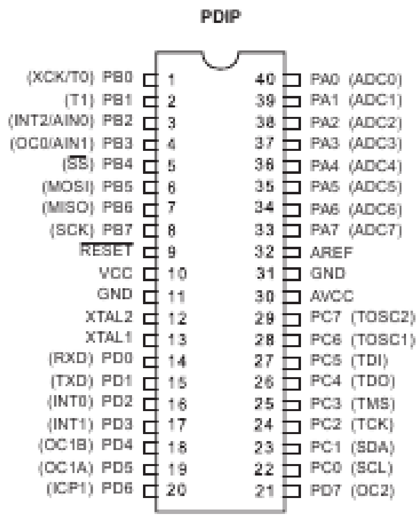
$GPGSV,3,1,12,12,68,319,45,10,61,117,44,05,61,147,43,02,48,013,46*77

$GPRMC,062130.000,A,1907.9357,N,07255.0014,E,0.12,158.57,150909,,,A*66

$GPVTG,158.57,T,,M,0.12,N,0.2,K,A*02

The prefix GP stands for GPS. At each point in the world, at least 6 satellites are accessible. The GGA line gives the fixed data value regarding time and position. The GSA line shows the active satellites. The RMC line refers to the recommended minimum number of satellites, which also has the value of the latitude and longitude of the location. The GSV line gives the satellite number followed by its azimuth and elevation angles. The VTG line refers to the track made good and the ground speed of the device.

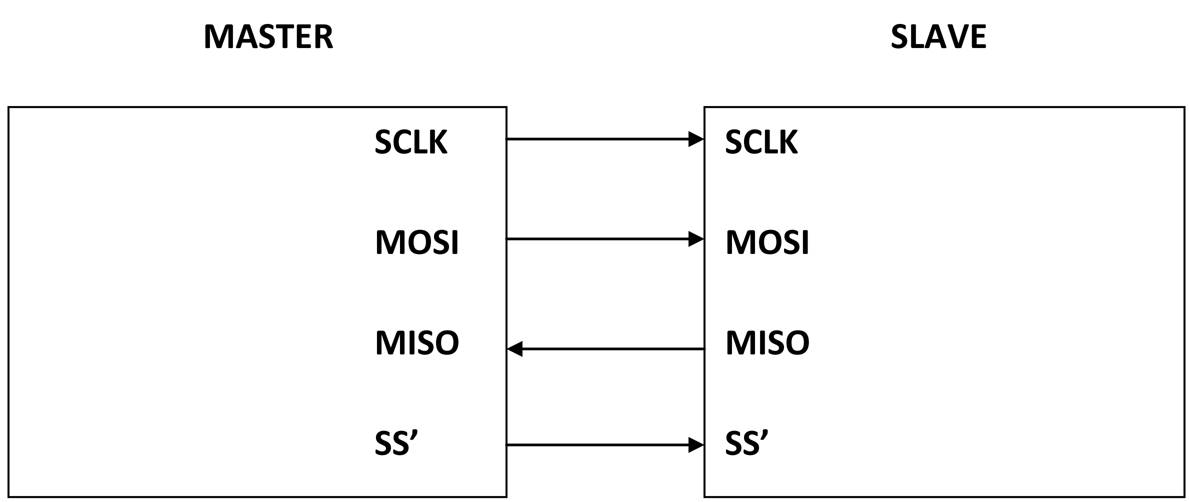
MICROCONTROLLER ATMEGA16L: An ATMEGA16L is preferred over an ATMEGA16 owing to the ability to operate the former at 3.3V, which not only is easily interfaced with the GPS module, but also consumed lesser power.



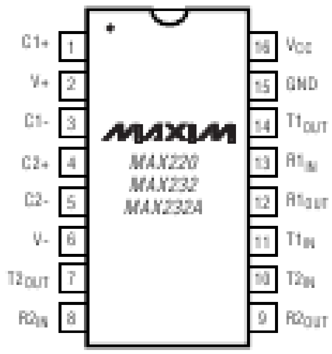
Two special functions of the microcontroller are utilized, the Universal Asynchronous Receiver/Transmitter (UART) and the Serial Peripheral Interfacing (SPI). The data out of the GPS module is connected to the receiver pin of the UART. The Micro-SD card, which is used to store the data from the GPS module, is connected to the SPI pins of the microcontroller.

UART translates data between serial and parallel forms. The UART system consists of shift registers which enable this translation. This serial way of data transfer is much more effective than the parallel data transfer, which uses multiple wires.

The SPI consists of two blocks, the master and the slave, as shown in the figure below. Like the UART, it is a serial data transfer system which transfers data synchronously, unlike the asynchronous UART.



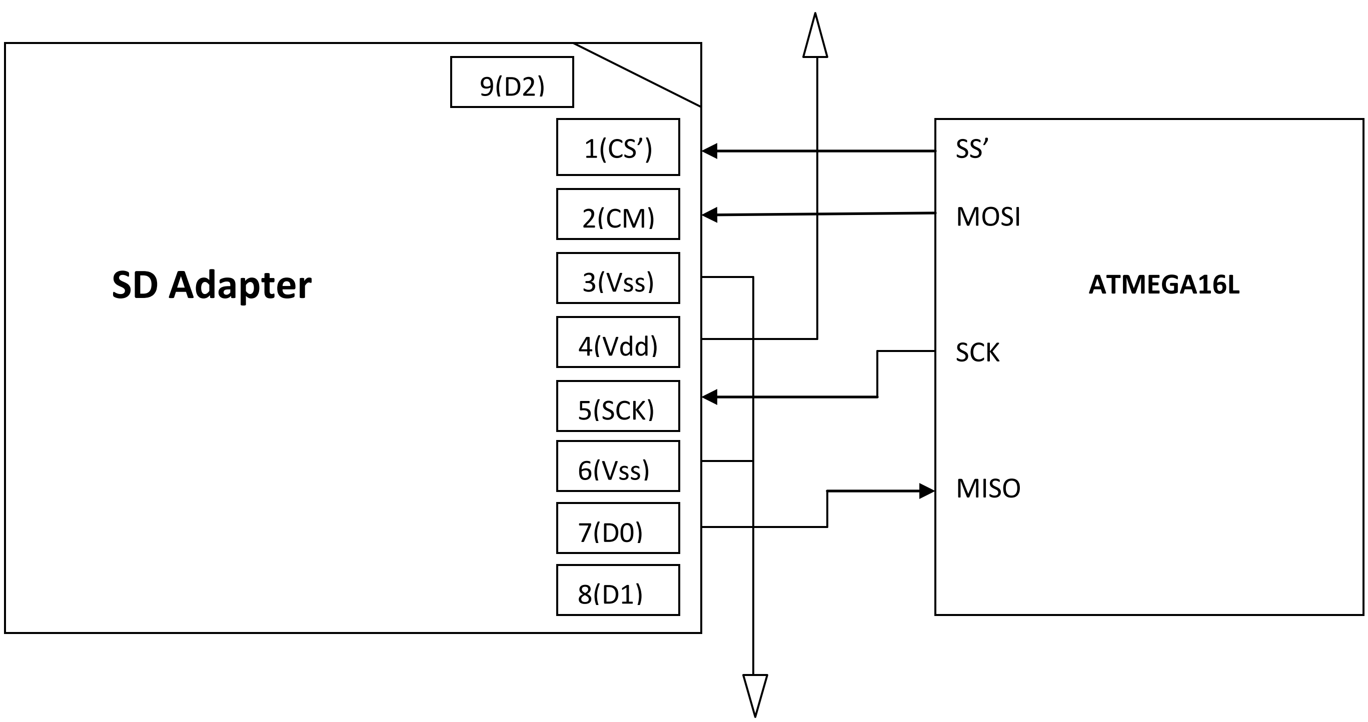
MAX232 Serial Driver/Receiver:



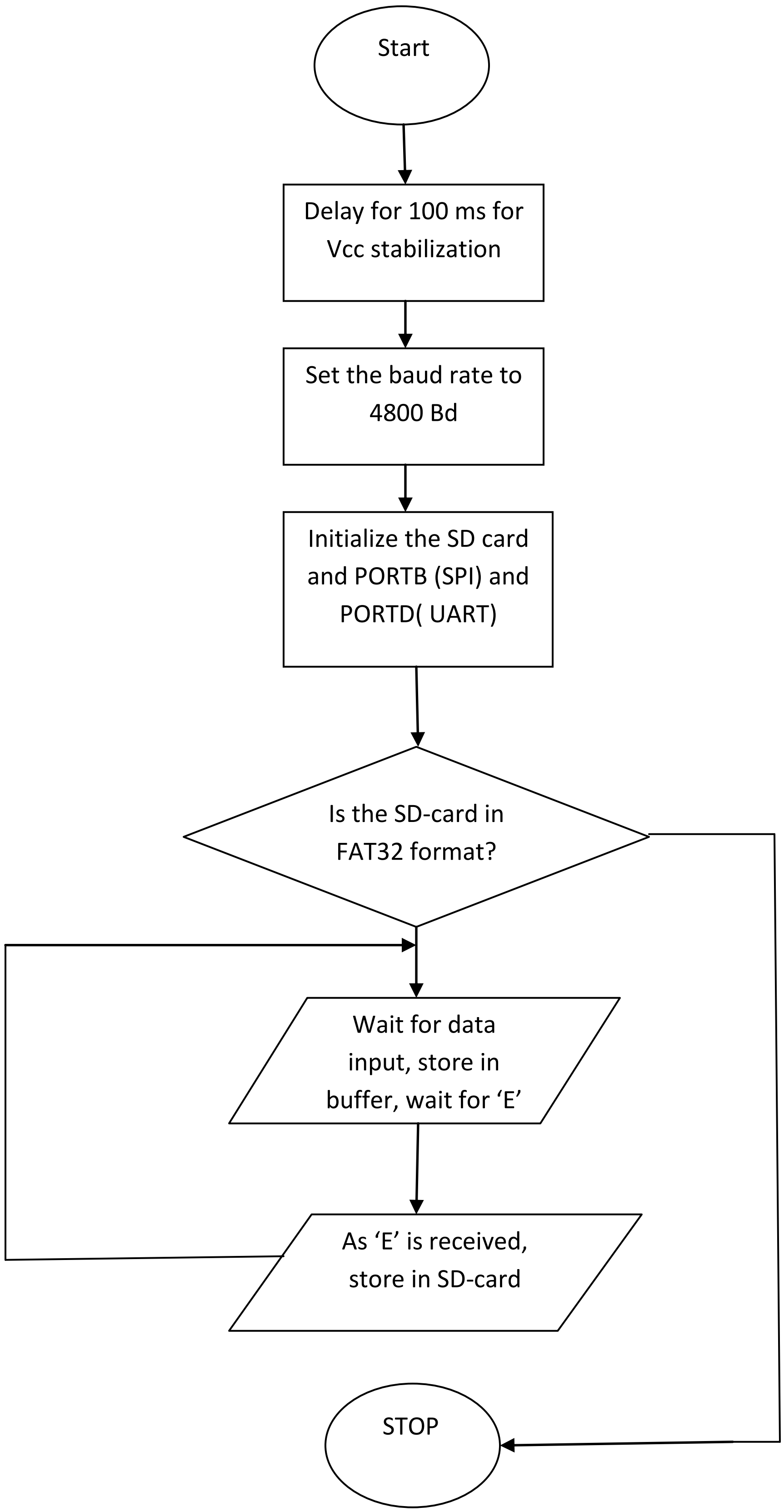
This IC is used to connect the GPS module to the PC. The voltage inverter and doubler in the IC needs three capacitances (10nF each) to function. Vcc is given at 3.3V. The UART output pin of the GPS module is connected to the T1in pin and the T1out pin is connected to the serial port of the PC using a DB-9 connector.

Micro-SD Card: It is the smallest commercially available memory card. An adapter is used to make it compatible with the SD adapter used in the circuit. It operates at 3.3V. A maximum speed of 25Mbits/s can be achieved using an SPI data transfer. Its power consumption varies from 66mW to 330mW. It is formatted in FAT32 format as the used C code is compatible with only this format.

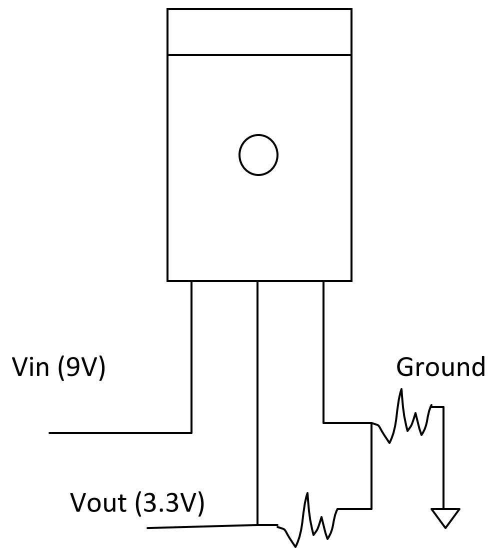
The SPI interfacing is similar to that for an SD card due to the usage of an SD adapter.



The SD card protocol is a simple command-response protocol. The command line of the SD card is connected to the MOSI pin of the SPI in the microcontroller. Before the read/write operation, we need to initialize the card. The command is sent packed as a 6-byte structure. The SD card responds to every command. The flow chart for the SD-Write cycle is given below.



LM 317 Voltage Regulator:



It's a simple three terminal positive voltage regulator capable of producing currents greater than 1.5A. It just needs a couple of resistances to function and requires no capacitors, unlike other voltage regulators.

Google Maps API: Google Maps is entirely coded in XML and JavaScript. The API that it offers enables the user to integrate the service with their websites. This is done by adding the Google Maps JavaScript to the required page and using other functions to suit our needs. We get a key to the Google Maps API on their site which could be used to obtain the JavaScript. Though the latest API has a geocoding compatibility, we just require the plotting using the latitude and longitude values we obtain from the GPS module at regular intervals.

XAMPP is an open source, free cross-platform web server mainly for PHP scripts. It is an acronym for Cross-Platform(X) Apache HTTP Server (A) MySQL (M) PHP (P) and Perl (P). It was used to create an offline webpage of our own, using the JavaScript of the Google Maps API. Though the created page is offline, it needs to access the database, which in turn requires the internet. So, a server is created using XAMPP which stores the Google Maps API as a JavaScript and a PHP based code is used to open the webpage and to regularly refresh the data.

We also use a pointer for the starting point and a 'poly' to plot the path traversed, both of which could be found in the example codes given in the documentation page.

HyperTerminal in Windows has an option of writing the input data values into a text file. The script keeps referring to this file for new data, and regularly plots the values when they get stored in the file.

PROBLEMS FACED: During the course of the project, there were a lot of difficulties we encountered. Some of them were hardware errors and some of them were software issues.

The major issue was with the interfacing of the SD Card. Unless it is formatted properly in FAT32, it does not work. There were serious issues with the coding too. We tried a couple of test codes before we settled with the final code.

Also the initial idea was to work with an MSP series microcontroller. Owing to its novelty and the inherent difficulty associated with the PCB design and programming, we decided to stick to ATMEGA16L.

APPLICATIONS: The idea for the project was conceived keeping in mind the needs of a taxi-driver or a chauffeur. The real-time display acts like a GPS Navigation System for the car, assisting the driver with unknown routes. Also, the SD card can be used to store the locations and the paths the car has been, which can act as a monitoring measure or even a parental control measure!

There exist these kinds of systems which use laptops as display. A more portable device can be made if the GPS module was integrated with a mobile phone, which is not difficult, considering the fact that Google Maps has an API for even mobile phone based systems. Even otherwise, the display is not an absolute necessity in this case. The SD Card can still be used to store the information, which can be displayed at a later time.

We have seen that the maximum power consumption of the SD card read/write and the GPS module are 330mW and 110mW respectively, which adds up to a total of 0.44W. From the measured values for our circuit, we get a power consumption of approximately 0.5W (9V battery with a net current of 55mA) which is not too much higher than the estimated value.

REFERENCES:

I. SD Card interfacing: http://www.captain.at/electronic-atmega-sd-card.php
II. SD Card Software: http://www.dharmanitech.com/2009/01/sd-card-interfacing-with-atmega8-fat32.html
III. Google Maps API: http://code.google.com/apis/maps/documentation/mapsdata/developers_guide_protocol.html
IV. iWave GPS Module: http://iwavesystems.com/iW-GPS.htm
V. GPS NMEA Protocol Description: http://home.mira.net/~gnb/gps/nmea.html