REMOTE VEHICLE CONTROLLING
&
TRACKING USING GPRS AND GPS

By

Muhammad Kashif Naveed

Waqar Ali

Muhammad Yousaf Khan

Hassan Ali

Submitted to the Faculty of Electrical Engineering Department

National University of Sciences and Technology, Islamabad

in partial fulfillment for the requirements of a B.E Degree in

Telecom Engineering

JUNE 2014

ABSTRACT

Due to advancement in next generation technologies, humans are being replaced by computers and robots. So, there was a need to automate life so that user can take advantage of the technological advancement in such a way that a person having internet connection can control and track his ground vehicle in real time. Although, satellite communication is the most suitable technology available for wide range of applications but it is not commonly available to ordinary clients, therefore GPRS is the next best option available.

In our project "**Remote vehicle control and tracking using GPRS and GPS**", remote vehicle is being controlled and tracked at the same time. Exact location of the ground vehicle is also being traced out on Google Map.

This project can be best security solution for the areas or perimeters where human monitoring is not possible all the time. This project can potentially be used in the design of the next-generation, secure systems because we can implement ultrasonic sensors for hurdles and also using DSP's, real time video transmission or picture capturing is possible.

ACKNOWLEDGEMENTS

Firstly, thank to Allah with His blessing for our **“Remote vehicle controlling and tracking using GPRS and GPS”** project. We want to say A BIG THANKS to our worthy supervisor Dr Adil Masood Siddique, who helped us from the earliest first design of the project, up to the final design. His guidance and advices throughout the process of this final year project were useful indeed.

We also want to say A BIG THANKS to our beloved parents and siblings for always mentally and financially supporting us while doing this project. Achieving a big target of this FYP was impossible without their prayers.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

RC: Remote Control

TCP/IP: Transmission Control Protocol/Internet Protocol

GPRS: General Packet Radio System

GPS: Global Positioning System

IT: Information Technology

DSP: Digital Signal Processing

SMS: Short Message service

RF: Radio Frequency

GSM: Global System for Mobile Communication

PIC: Peripheral Interface Controller

LCD: Liquid Crystal Display

LED: Light Emitting Diode

PCB: Printed Circuit Board

AT Commands: Attention Commands

CHAPTER 1: INTRODUCTION

1.1 Overview

The project revolves around the remote controlling and tracking of vehicle using GPRS and GPS. The major aim of this project is to build up a prototype by which a user having internet connection can control and monitor his end application via internet. End application can be an air or ground vehicle, but we will implement it on prototype ground vehicle. End application will be connected to user by GPRS. GPRS gives almost instantaneous connection setup and continuous connection to internet, data rates (14.4kbit/s to 115kbit/s). At the same time the vehicle will be tracked using the GPS (Global Positioning System) service. Longitudes and latitudes of the position will be sending to the user via GPRS and hence the vehicle can be tracked in real time on Google maps.

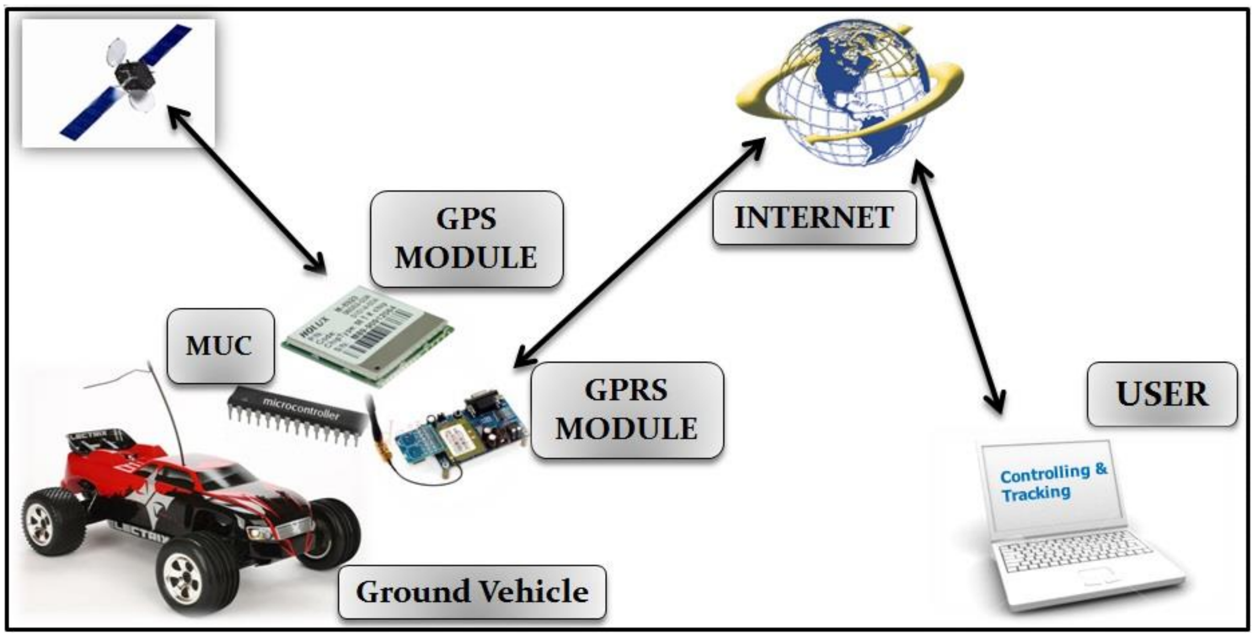


Figure 1.1 Project Overview

1.2 Problem Statement

Objective was to control a vehicle with optimum freedom of range and minimum delay. Satellite communication is the best technology for this purpose as the modern DRONES and tracking system use GPS service but this service is not available in Pakistan. Only

GPS receivers are available which have a very limited bandwidth. GPRS communication can be used as a counterpart for satellite communication as it provides continuous connection to internet and can be used for worldwide communication. GPRS provides us optimum freedom of range that is our vehicle can be controlled from anywhere in the world, you just need an internet connection.[1]

1.3 Background

From the past few years, people are continuously working on vehicle controlling and tracking. For controlling purpose, some of them used Remote Control (RC) technology; many people used SMS based technology; few started work on GPRS (internet) Technology but for tracking purpose, people are still working on GPS technology. Let's discuss these technologies and their drawbacks.

1.3.1 RC Technology

In the early 21st century, 2.4 gigahertz (GHz) transmissions have become increasingly utilized in high-end control of model vehicles and aircraft. This range of frequencies has many advantages. Because the 2.4 GHz wavelengths are so small (around 10 centimeters), the antennas on the receivers do not need to exceed 3 to 5 cm. Electromagnetic noise, for example from electric motors, is not seen by 2.4 GHz receivers due to the noise's frequency (which tends to be around 10 to 150 MHz). The transmitter antenna only needs to be 10 to 20 cm long, and receiver power usage is much lower; batteries can therefore last longer. In addition, no crystals or frequency selection is required as the latter is performed automatically by the transmitter. However, the short wavelengths do not diffract as easily as the longer wavelengths of PCM/PPM, so 'line of sight' is required between the transmitting antenna and the receiver. Also, should the receiver lose power, even for a few milliseconds, or get 'swamped' by 2.4 GHz interference, it can take a few seconds for the receiver - which, in the case of 2.4 GHz, is almost invariably a digital device - to re-sync.

The major **disadvantage** of this technology is limited or short range i.e. round about 300 feet.

1.3.2 SMS Technology

The major **disadvantage** of this technology is delay. Due to this reason, SMS technology is being used for controlling and monitoring of various equipment. The user may get the status of a remote device like pressure, temperature, voltage, current, on/off condition, weight etc. by sending respective message to the remote device.

1.3.3 Satellite Communication

Satellite communication is the best technology for wide range of applications but we can't use this technology in Pakistan. So, GPRS is counterpart for us.

13.4 GPRS and GPS Technology

Due to enhancements in 3G evolutions and the world moving fast to Internet, these technologies are very advance and have many advantages. We have also used GPRS and GPS technology in our project to implement the latest technology and to overcome the issues occurred in previous technologies.

1.3.5 Advantages

a. **World-wide communication:**

   User will control the end application using any PC/Tablet/Mobile with internet connection. That means end application will be accessed from any device having internet anywhere in the world.

b. **Real time controlling of end applications:**

   End application will be controlled with minimum delay.

c. **Real time video transmission or Picture capturing:**

   3G networks provide higher data rates over mobile internet which can be used for remote video transmission or sending pictures over MMS service.

d. GPRS provides user an optimum freedom of range. Radio frequency, infrared, ad-hoc networks have very limited range.

e. GPRS provide continuous connection and this can be used for controlling the end application in real time and for also video and picture transfer.

United States military was the first one to launch GPS for surveillance purposes. GPS access was given to civilian after some years but with a demoted accuracy grade. The US military still holds the GPS access even decades after its invention, however Europe is inventing its own system which will be named as Galileo.[2]

Vehicle-tracking systems were very costly involving state-of-the-art skill and technology. However, as the world is progressing and moving ahead, dramatic drops in prices of vehicle- tracking systems, more companies have an access to it who will profit the most from it.

Total purchase options were controlled to be used by companies. But now hiring and pay options are available. Initial capital expenditure is no more required which restricted many companies from configuring vehicle-tracking systems.

Many developments have taken place regarding the provision of applications of vehicle-tracking systems. Vehicle-tracking systems can be utilized for producing expenses reports, time sheets, reducing fuel consumption and increasing the efficiency of driving. Geofences can be arranged around specific locations providing an alert when the driver crosses them. This will deliver added safety and security and customer service capabilities for the clients.

1.4 Project Significance

This project has many applications such as:

a. A robot controlled over internet that can be used for security purposes.
b. Vehicle can be used for remote video transmission or sending pictures over MMS service.
c. A portable device installed on vehicles that connect to internet using the GPRS Module and uploads current position (using Global Position System) to a server. The server stores that location in a database with the ID of vehicle. Then a client (using a PC) can connect with the server using World Wide Web to see the route of the vehicle.
d. The GPRS provides always-on service. The need to dial up like one has to on a PC instance is no more a requirement.

e. GPRS works on packet switching and speeds 56-118 kbps are attainable which is a great improvement compared to speed 9.6 kbps for circuit switched networks. Speed of 171.2 kbps are also possible when combined with standard GSM time slots.

1.4.1 Advantages over other Methods of implementation

a. GPRS provides user an optimum freedom of range. Radio frequency, infrared, ad-hoc networks have very limited range.
b. GPRS provide continuous connection and this can be used for controlling the end application in real time and for also video and picture transfer.
c. Satellite communication is the best technology for wide range of applications as we cannot use this technology at our level, so GPRS is the counter part for us.

1.5 Organization of Thesis

a. In first Chapter, the introduction of the project is given. For example overview, significance etc.
b. Second chapter presents the literature review of the communication technologies used in our project.
c. Third chapter is related to all the hardware components and modules used in this project.
d. In fourth chapter, methodology is discussed in detail.
e. Fifth chapter is completely describing the software portion of the project i.e. Proteus design, MP LAB, Server control application using MS visual studio.
f. Sixth or Last chapter is related to Final design of our project **“Remote vehicle controlling and tracking using GPRS and GPS”**.

CHAPTER 2: LITERATURE REVIEW

2.1 GPRS Communication

General packet radio service (GPRS) communication is a wireless procedure and practice using packet-switching networks for the transferal of data. It is connected with generation 2.5 (2.5G) wireless networks. GPRS mobile devices can send multimedia messages (MMS), download software, check email etc.

Packet switching is a development over circuit-switching communication technology. In circuit-switching the preset line of communication is required for handover of information. No other transmission can use that line for the duration of the transfer, called dedicated networks. It is good for voice calls and must be transmitted in continuous fashion to make sense.

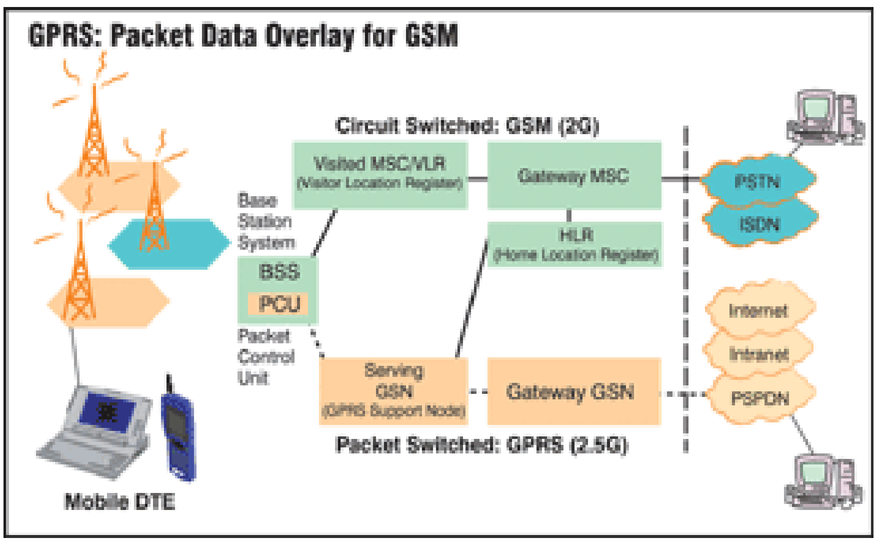


Figure 2.1 GPRS (2.5G)

Source: (Google images)

Data, on the other hand, does not need to be communicated uninterruptedly. GPRS communication breaks down or allocates data into packets and sends it wirelessly on any available / obtainable line. The packets are reconstructed at the destination point. Networks that do not use dedicated lines for each transmission are known as connectionless.

Connectionless networks form a virtual line for wireless devices. Data can be transferred endlessly. Due to this push-to-talk voice calls and instant messaging is possible, which require a continuous exchange of data.

Before GPRS technology, sending and receiving of data was possible, but it was expensive and slow using 2G technology in which only10 short messaging service (SMS) text messages could be sent per minute. In GPRS you can send 30 SMS texts per minute. Third-generation (3G) and fourth-generation (4G) wireless devices also use GPRS communication technology. The improved bandwidth of later generations of wireless networks allow transfer speeds of more than 10 megabytes per second.

GPRS communication actually decreases the cost of use per person. In circuit switching the customer was essential to pay for use of the entire line during voice calls and data transfers. Data charging cost was per minute, regardless of how much data was actually downloaded. In packet-switching indicting method is per megabyte, and the overall rate is much cheaper because multiple customers use the same line.

GPRS technology has been progressive since its first implementation in the year 2000. These developments have made it possible for the customers to use GPRS to make voice calls. Voice over Internet protocol (VOIP) technology uses computers to call telephones and cell phones. VOIP uses packet-switching Internet networks instead of traditional laid telephone cables.

The speed of transfer for GPRS communication is 114 kilobytes per second. This speed is reasonably sufficient which enables fast communications. Without the increased speed, wireless networks would be too slow to exchange picture messages, video files and email or to download software. [3]

2.2 TCP/IP

TCP/IP stands for **T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol.

TCP/IP defines communication protocol for the Internet. A computer communication protocol is a depiction of the rules which must be followed by the computers on far ends to communicate with each other, TCP/IP is the communication protocol for communication between computers on the Internet, TCP/IP defines how electronic devices (like computers) should be connected to the Internet, and how data should be transmitted between them.

Inside the TCP/IP standard there are several protocols for handling data communication:

a. TCP (Transmission Control Protocol) communication between applications
b. UDP (User Datagram Protocol) simple communication between applications
c. IP (Internet Protocol) communication between computers
d. ICMP (Internet Control Message Protocol) for errors and statistics
e. DHCP (Dynamic Host Configuration Protocol) for dynamic addressing

TCP is for communication between applications. If one application wants to communicate with another via TCP, it sends a communication request. This request must be sent to an exact address already defined. After a "handshake" between the two applications, TCP will set up a "full-duplex" communication between the two applications. The "full-duplex" communication will occupy the communication line between the two computers until it is closed by one of the two applications.

UDP is very similar to TCP, but simpler and less reliable.

Internet Protocol is for communication between computers. IP is a "connection-less" communication protocol. IP does not occupy the communication line between two computers. IP reduces the need for network lines. Same line can be used for communication between many different computers at the same time. With IP, messages (or other data) are broken up into small independent "packets" and sent between computers via the Internet. IP is responsible for "routing" each packet to the correct destination.

TCP/IP is TCP and IP working together. TCP provides the communication between your application software (i.e. your browser) and your network software. IP takes care of the communication with other computers. TCP is responsible for breaking data down into IP packets before they are sent, and for assembling the packets when they arrive. IP is responsible for sending the packets to the correct destination.

2.3 Socket Programming (Client-Server Connection)

A socket is one endpoint of a two-way communication association between two programs running on the network. A socket has a port number which helps TCP layer so that it can recognize the application that data is intended to be sent.

A server is supposed to work on a specific computer and consists of a socket which is linked to a specific port number. Meanwhile the server waits and then listens to the socket for a client to make a connection request.

The client knows the hostname of the device on which the server is running and the port number to which the server is listening. The client tries to engage with the server on the server's device and port to make a connection request. The client needs to recognize itself to the server so it binds to a local port number for use during this connection. This is allocated by the system.

If the above stated process goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. A new socket is required for continuation to listen to the original socket for connection requests.

If the connection is accepted on the client side a socket is created and the client use the socket to interconnect with the server. The client and server can now connect by writing to or reading from their sockets as explained in the figure below.

2.3.1 Socket Address

a. Combination of an IP address and a port number is the Socket Address.

b. Internet sockets transport received data packets to the suitable application process.

Following figure demonstrates the steps involved for creating socket-based server and client programs.

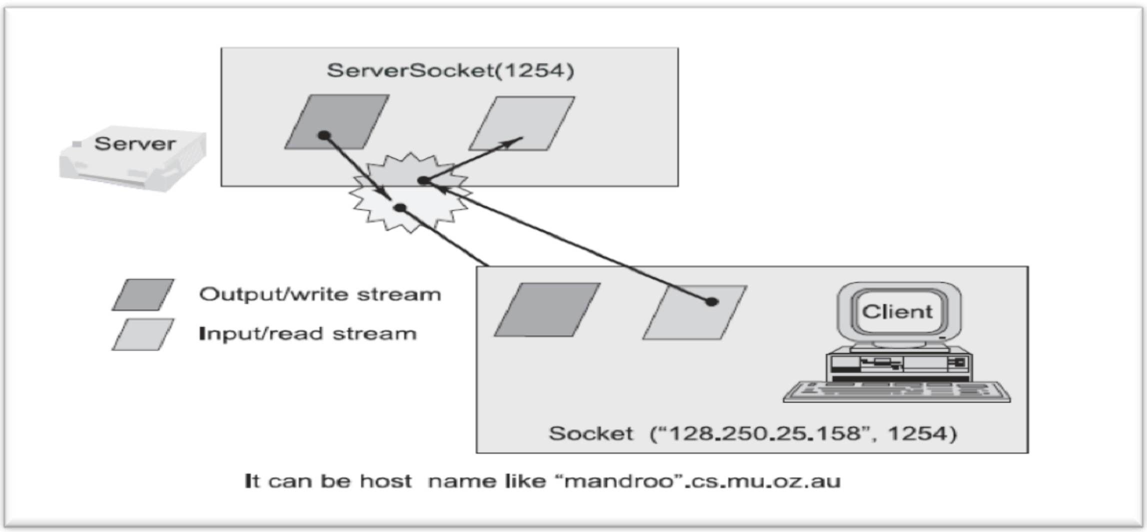


Figure 2.2 Socket-based server and client program

Source: (Google Images)

CHAPTER 3: HARDWARE COMPONENTS AND MODULES

3.1 SIM 300 GPRS/GSM module

The SIM300 is a Quad Band GSM/GPRS in a small form better to be used in university level projects. GSM Modem can use any GSM mobile operator SIM card and act just like a mobile connection. We can use its RS232 port to communicate and develop embedded applications.[4]

We can connect module to PC serial port directly or by using microcontroller. We can send and receive SMS or make/receive voice calls. In GPRS mode we can also connect to any remote FTP server and upload files for data logging. It also supports wide range of certain other features which include Voice, SMS, Data/Fax, GPRS and integrated TCP/IP stack. [5]

3.1.1 SIM300D AT Command Set

The GSM engine is controlled by the controlling device by sending AT commands via the serial interface. The controlling device of the serial line is mentioned as following term:

a. Terminal Equipment (TE)
b. Data Terminal Equipment (DTE)

3.1.2 AT Command syntax

The "AT" or "at" prefix must be set at the beginning of each command line. To terminate a command line enter <CR>. Commands are usually followed by a response that includes.”<CR><LF><response><CR><LF>”. Only the responses are presented, <CR><LF> are omitted intentionally. The AT command set implemented by SIM300D is a combination of GSM07.05, GSM07.07 and ITU-T recommendation V.25ter and the AT commands developed by SIMCOM. Flow control is very important for correct communication between the GSM engine and DTE. For in the case such as a data or fax call, the sending device is transferring data faster than the receiving side is ready to accept. When the receiving buffer reaches its capacity, the receiving device should be capable to cause the sending device to pause until it catches up.

3.1.3 AT Commands for SMS

Command	Description
AT+CMGD	DELETE SMS MESSAGE
AT+CMGF	SMS MESSAGE FORMAT
AT+CMGL	LIST SMS MESSAGES FROM PREFERRED STORE
AT+CMGR	READ SMS MESSAGE
AT+CMGS	SEND SMS MESSAGE
AT+CMGW	WRITE SMS MESSAGE TO MEMORY
AT+CMSS	SEND SMS MESSAGE FROM STORAGE
AT+CMGC	SEND SMS COMMAND
AT+CNMI	NEW SMS MESSAGE INDICATIONS
AT+CPMS	PREFERRED SMS MESSAGE STORAGE
AT+CRES	RESTORE SMS SETTINGS
AT+CSAS	SAVE SMS SETTINGS
AT+CSCA	SMS SERVICE CENTER ADDRESS

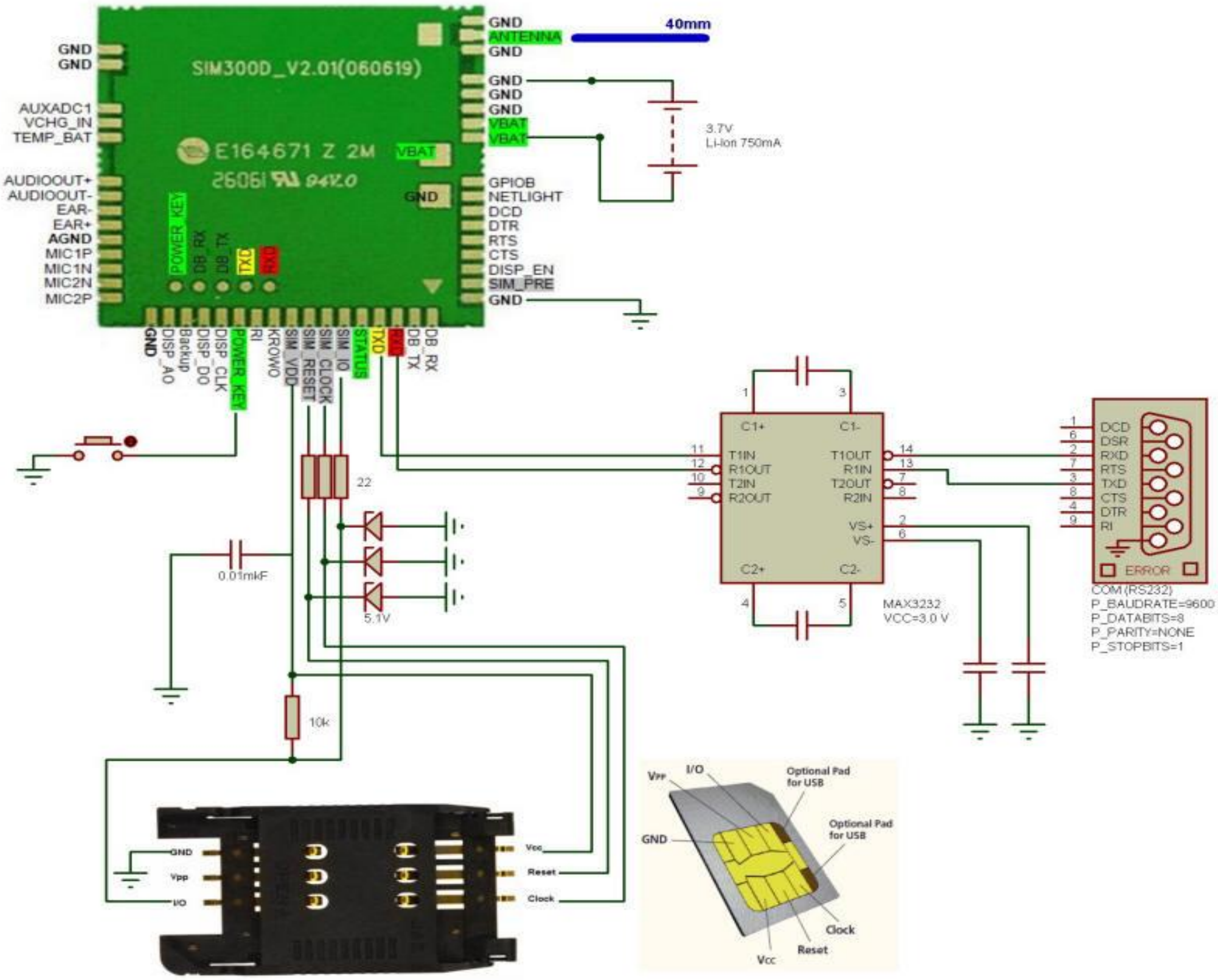


Figure 3.1 SIM300 circuitry

3.1.4 Applications

a. Remote Control and Alerts (SMS)

b. Security Applications

c. Sensor Monitoring

d. GPRS Mode Remote Data Logging

3.1.5 Features

a. Consistent for 24x7 procedure with Matched Antenna

b. Status of Modem Indicated by LED

c. Simple to Use and Low Cost

d. Quad Band Modem supports all GSM operator SIM cards

3.2 Holux M8929 GPS module

M-8929 is an ultra-miniature with a dimensions of 25.4 * 25.4 * 2.5 mm which is designed by low power consumption MTK GPS solution. Using this GPS we can almost accurately find co-ordinates of the device on which it is installed. The value of sensitivity is up to -165dBm and fast Time-To-First-Fix in navigation application. The provision of stable results of M-8929 is best suited to be embedded in prototype devices, like Smart phone, Tablet PC, DVR, PND for GPS service.

Figure 3.2 Holux M-8929

3.2.1 Key Features

a. Small size of 25.4* 25.4 * 2.5 mm
b. High sensitivity -165 dBm
c. 22 tracking channels and upto 66 parallel searching
d. Fast Position Fix
e. Low power consumption

3.2.2 Applications

a. Automotive and Marine Navigation
b. Automotive Navigator Tracking
c. Emergency Locator
d. Geographic Surveying
e. Personal Positioning
f. Smart phone, Tablet PC, DVR, PND etc.

3.2.3 Block Diagram

Following diagram shows the internal view of GPS receiver:

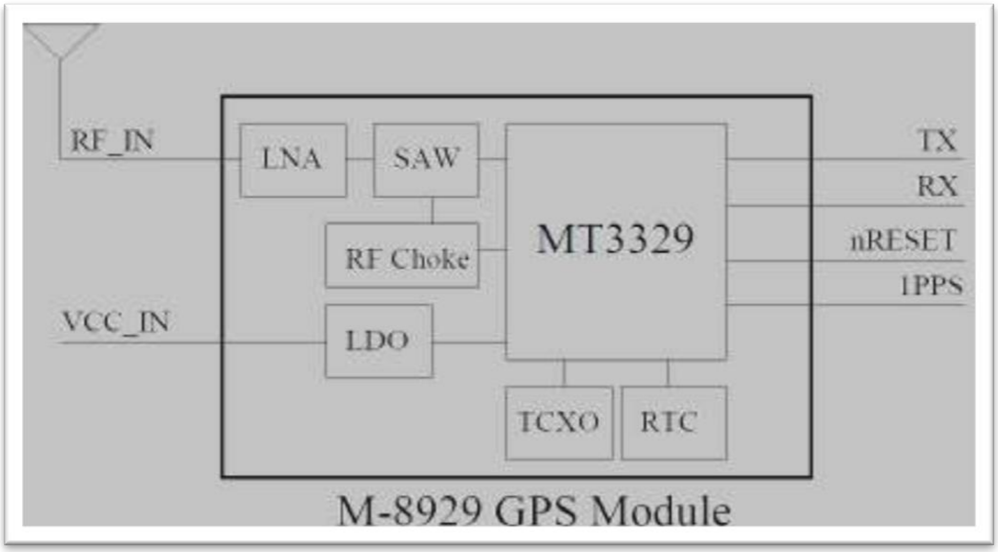


Figure 3.3 Internal Circuit Holux M-8929

M-8929 provides 2-wire digital UART port for communication of GPS position data using NMEA protocol or MTK extension protocol. UART port is capable of 4800 to 115200 baud rate. M-8929 is default to support standard NMEA-0183 protocol.

M-8929 is able to detect the presence of an external antenna. When M-8929 connects to an external active antenna, the active antenna sink a current higher than 3mA, M-8929 will send a signal to indicate that active antenna is functional.

3.3 PIC 18F452

A PIC microcontroller is a single integrated circuit small enough to fit in the palm of a hand. PIC microcontrollers can be used as the ‘brain’ to control a large variety of products. In order to control devices, it is necessary to interface or connect them to the PIC microcontroller.

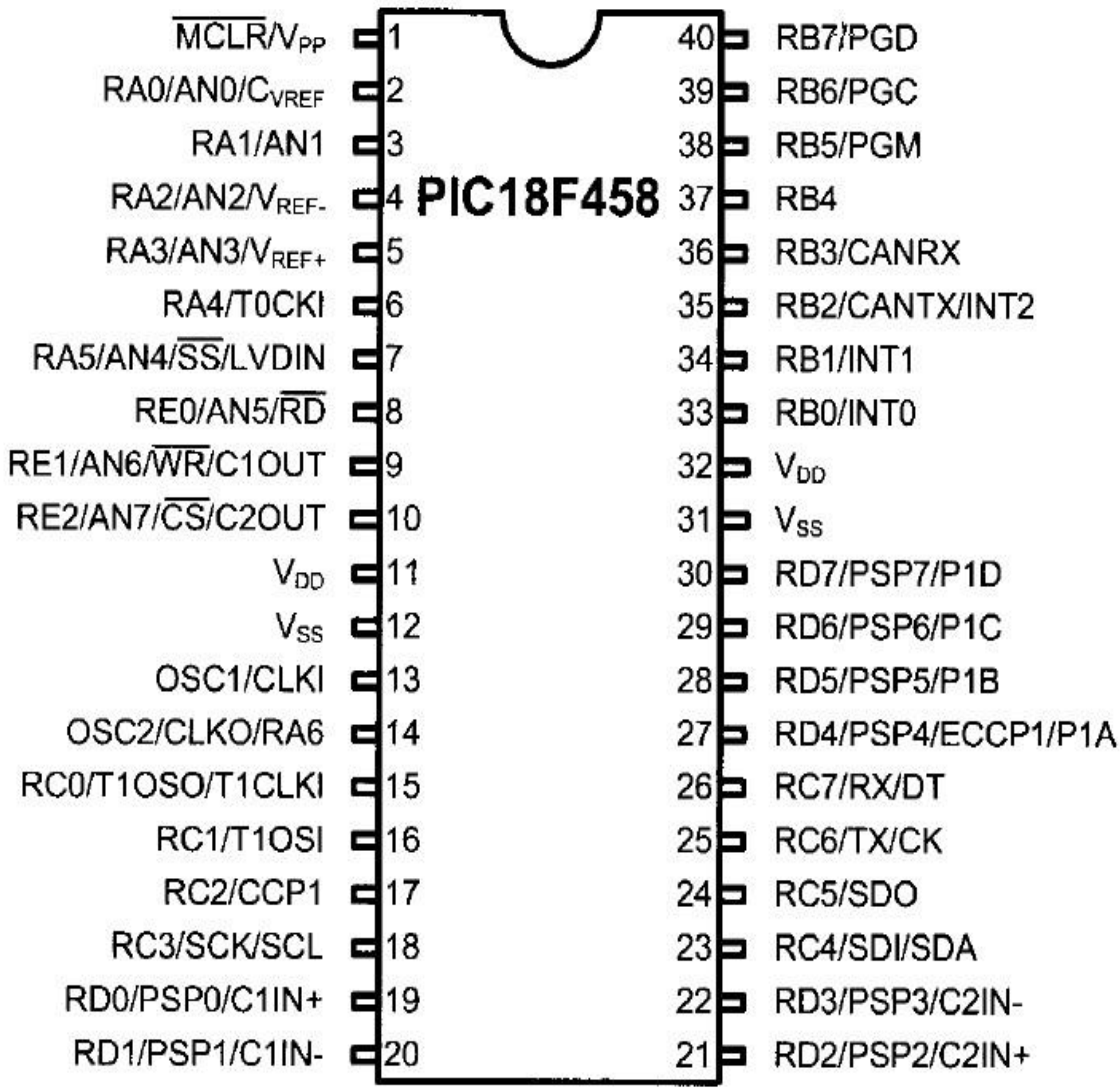


Figure 3.4 PIN configuration of PIC18F452

The PIC18F452 is, as with all the other **18F** series parts, **optimized for using C**.[6] It has a 31 deep hardware stack and **linear memory** (rather than banked memory) so you don't have to make adjustments for the hardware when coding in C.

The 18F452 has **16k of program memory** and **1536 Bytes of RAM**. The RAM is linear and can declare large arrays with no possibility if you use devices with banked memory (16F series).

It consists of **40 pin DIP** and can be programmed in circuit using ICSPI. As with the other 18F devices it has an 8 bit hardware multiplier (8x8).

4x internal phase locked loop oscillator is also present for the supply of an external clock signal of 10MHz. It runs at 40MHz internally but the maximum internal instruction rate is F osc/4 and with the PLL active (or using an external 40MHz clock) only1 instruction every 4 clock cycles is seen. The maximum one can get is **10MIPS**.

3.3.1 Features of PIC18F452

High Performance RISC CPU:

a. C compiler optimized architecture/ instruction set and Source code compatible with the PIC16 and PIC17 instruction sets
b. Linear program memory addressing to 32 Kbytes
c. Linear data memory addressing to 1.5 Kbytes, DC - 40 MHz oscillations /clock input 4 MHz - 10 MHz oscillations /clock input with PLL active
d. 16-bit wide instructions, 8-bit wide data path
e. Priority levels for interrupts
f. 8 x 8 Single Cycle Hardware Multiplier

3.3.2 Peripheral Features

a. Addressable USART module: Supports RS-485 and RS-232
b. Parallel Slave Port (PSP) module

3.3.3 Analog Features

a. Compatible 10-bit Analog-to-Digital Converter module (A/D) with: Fast sampling rate
b. Conversion available during SLEEP
c. Programmable Low Voltage Detection (PLVD) Supports interrupt on-Low Voltage Detection
d. Programmable Brown-out Reset (BOR)

3.3.4 CMOS Technology

a. Low power, high speed FLASH/EEPROM technology
b. Fully static design
c. Wide operating voltage range (2.0V to 5.5V)
d. Industrial and Extended temperature ranges
e. Low power consumption:

   - < 1.6 mA typical @ 5V, 4 MHz

   - 25 μA typical @ 3V, 32 kHz

   - < 0.2 μA typical standby current

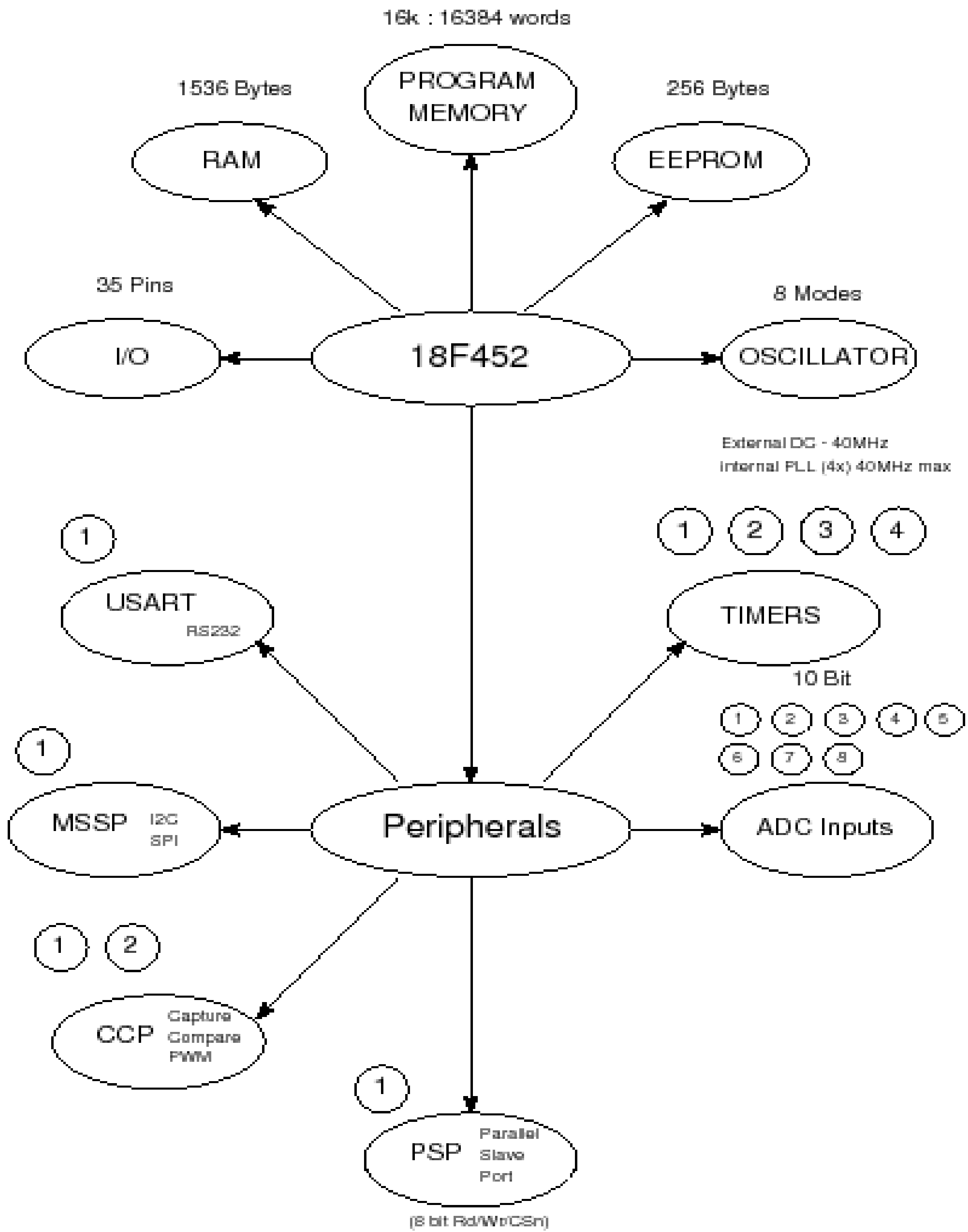


Figure 3.5 PIC18F452 Features

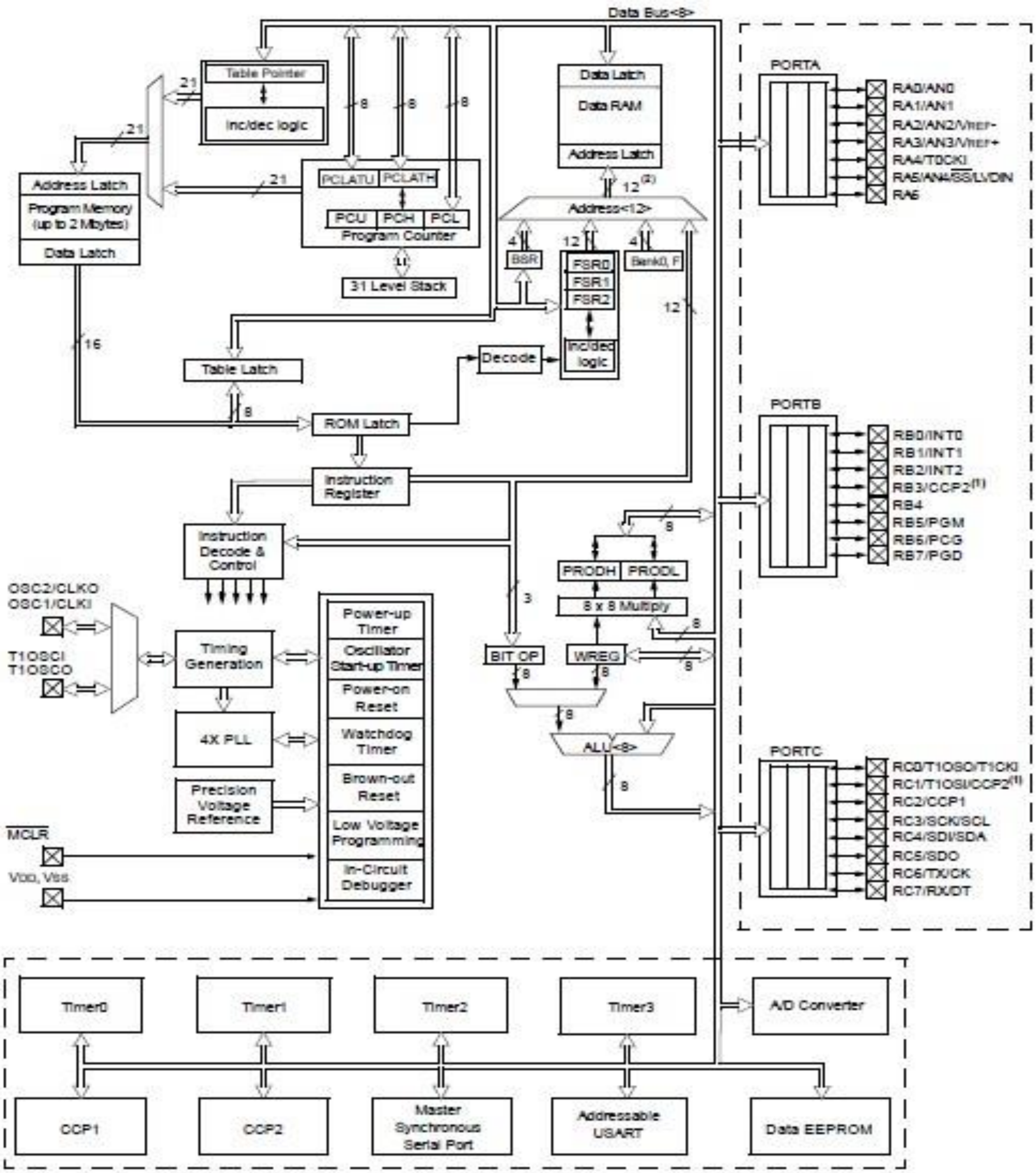


Figure 3.6 PIC18F452 Internal Circuit

3.4 DC Motors

DC motors are designed in many types and sizes, including brush less, servo, and gear motor types. A motor comprises of a rotor and a permanent magnetic field stator. The magnetic field is retained using either permanent magnets or electromagnetic windings. DC motors are most commonly used in variable speed and torque.

Motion and controls cover a wide range of components that in some way are used to generate and/or control motion. Areas within this category include bearings and bushings, clutches and brakes, controls and drives, drive components, encoders and resolves, Integrated motion control, limit switches, linear actuators, linear and rotary motion components, linear position sensing, motors (both AC and DC motors), orientation position sensing, pneumatics and pneumatic components, positioning stages, slides and guides, power transmission (mechanical), seals, slip rings, solenoids, springs.

Motors are the devices that provide the actual speed and torque in a drive system. This family includes AC motor types (single and multiphase motors, universal, servo motors, induction, synchronous, and gear motor) and DC motors (brush less, servo motor, and gear motor) as well as linear, stepper and air motors, and motor contactors and starters.

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

Let's start by looking at a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

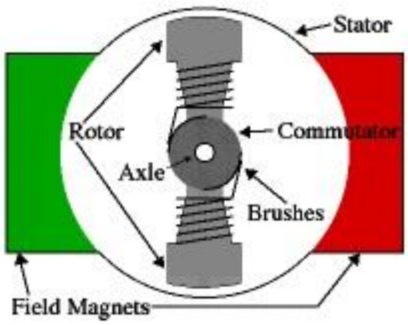


Figure 3.7 2-pole DC Motor

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beamers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, and driving it to continue rotating. The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque” ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

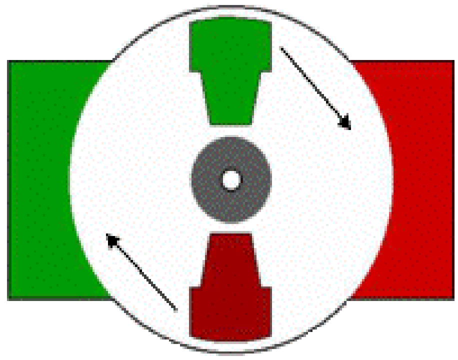


Figure 3.8 2-pole DC Motor

3.5 Opto-couplers

An opto-isolator, also called an optocoupler, photocoupler, or optical isolator is a module that transferals electrical signals between two isolated circuits with the help of light. Opto-isolators avert high voltages from affecting the system for receiving the signal. Opto-isolators endure input-to-output voltages up to 10 kV. Voltage transients with speeds up to 10 kV/μs. A LED and a phototransistor are present in common type of opto-isolator. Opto-isolators are used for broadcast of digital signals. Some techniques allow the use with analog signals.

There are two kinds of optocouplers (a light emitting diode (LED) as an input and a phototransistor as an output) according to the type of output transistor: Single transistor type and Darlington-transistor type.

Optocouplers (optical couplers) are designed to isolate electrical output from input for complete elimination of noise. They have been used conventionally as substitutes for relays and pulse transformers. Today's current technology in the area of microcom-puters creates new applications for optocouplers.

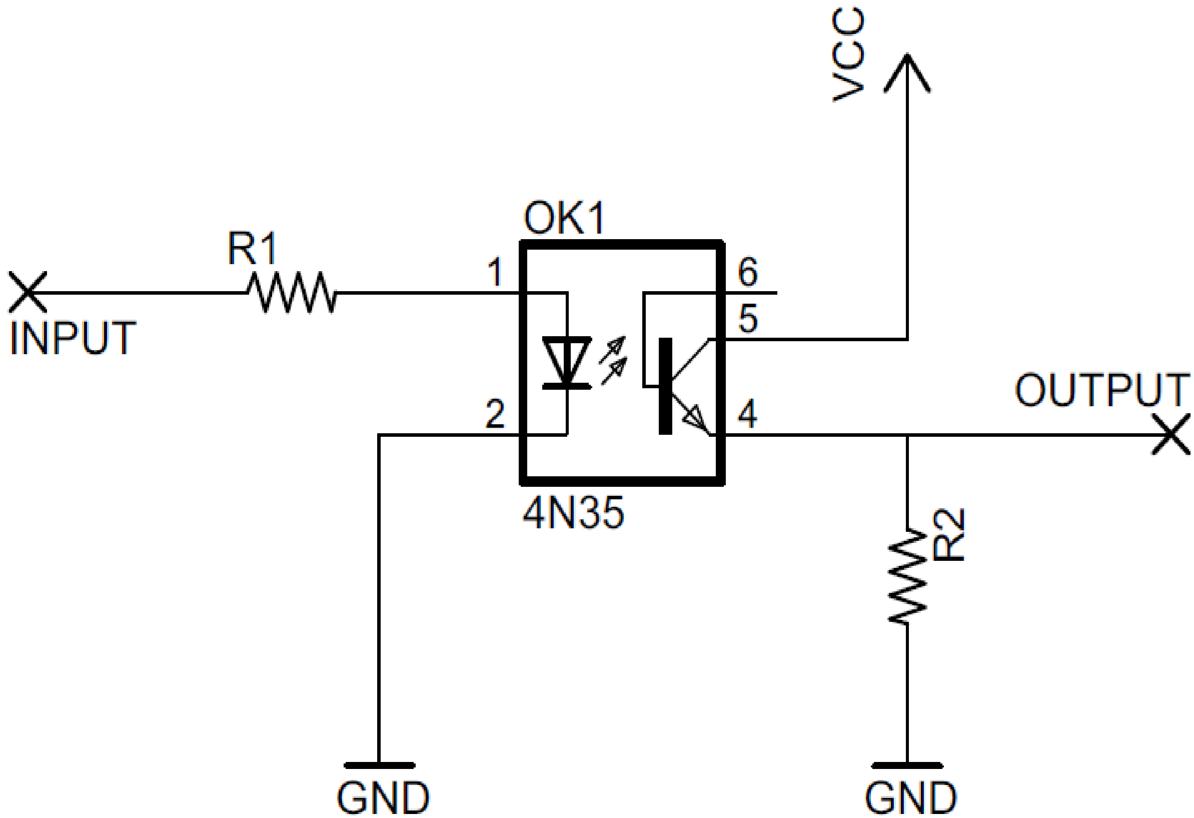


Figure 3.9 Opto Coupler

The single-transistor type optocouplers are used to perform high-speed switching (with high-speed response). The Darlington-transistor type optocouplers are used to obtain a large output current by utilizing a small input current (independently of switching speeds).

An opto-isolator comprises a source of light, a near infrared light-emitting diode (LED), for converting electrical input signal into light, a closed optical channel and a photosensor detecting incoming light and produces electric energy straight and modulates electric current flowing from an external power supply. A photoresistor, a photodiode, a phototransistor, a silicon-controlled rectifier (SCR) or a triac can be used as sensor. LEDs sense light apart from emitting it, construction of symmetrical, bidirectional opto-isolators is possible. An optocoupled solid state relay comprises a photodiode opto-isolator driving a power switch, usually a complementary pair of MOSFETs. A slotted optical switch consists of a light source and a sensor. Its optical channel is open, which allows modulation of light by external objects hindering the path of light or reflecting/ imitating light into the sensor.

3.5.1 Applications

a. Increasing switching speed
b. Stabilizing output levels
c. Elimination of induced noise
d. Elimination of input surges

3.6 L293D (dual H-bridge IC)

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

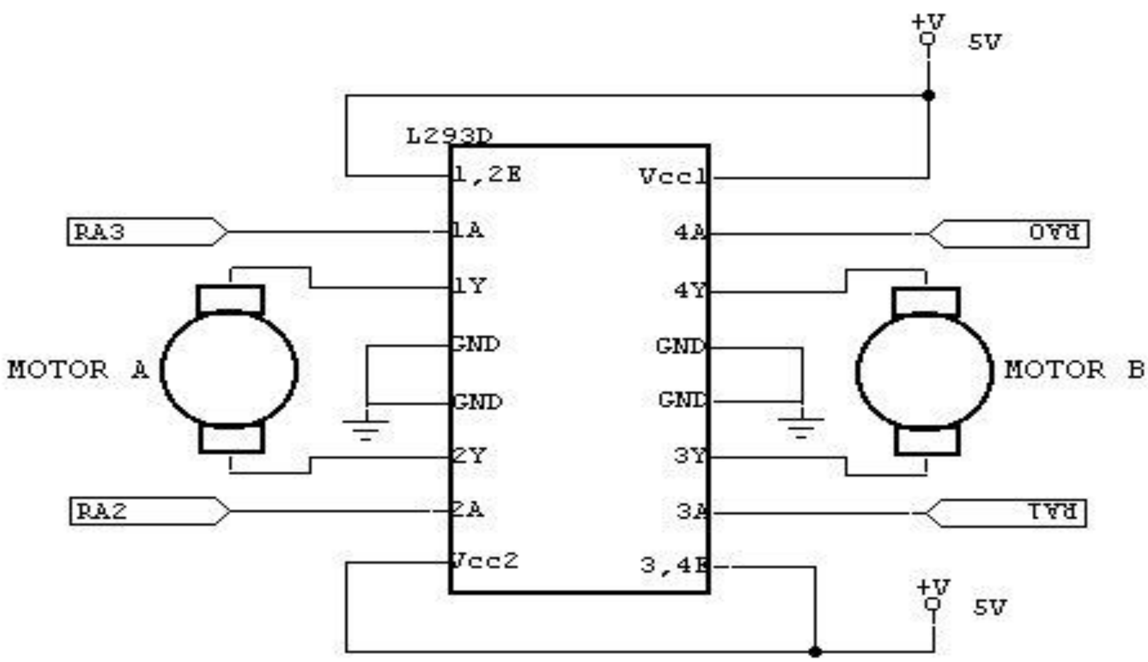


Figure 3.10 L293D operation

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 &

15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

3.6.1 Pin Diagram

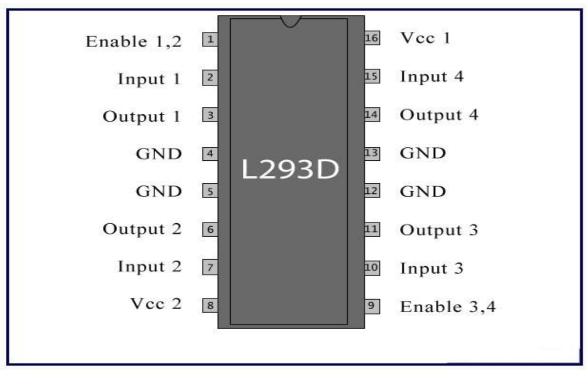


Figure 3.11 L293D Pin configuration

3.7 Max 232

The MAX232 IC is used to convert the TTL/CMOS logic levels to RS232 logic levels during serial communication of microcontrollers with PC. The controller operates at TTL logic level (0-5V) whereas the serial communication in PC works on RS232 standards (-25 V to + 25V). This makes it difficult to establish a direct link between them to communicate with each other.

The intermediate link is provided through MAX232. It is a dual driver/receiver that includes a capacitive voltage generator to supply RS232 voltage levels from a single 5V supply. Each receiver converts RS232 inputs to 5V TTL/CMOS levels. These receivers (R_1 & R_2) can accept ±30V inputs.

The transmitters take input from controller's serial transmission pin and send the output to RS232's receiver. The receivers, on the other hand, take input from transmission pin of

RS232 serial port and give serial output to microcontroller's receiver pin. MAX232 needs four external capacitors whose value ranges from 1μF to 22μF.

3.7.1 Voltage Levels

A MAX232 IC accepts a TTL level to convert, it changes TTL logic 0 to between +3 and +15 V, and changes TTL logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. The RS232 data transmission voltages at a certain logic state are contrary from the RS232 control line voltages at the same logic state. The table illustrates the above mentioned method.

Line type & Logic Level of RS232	RS232 Voltage	TTL voltage to/ from MAX232
Data transmission (Rx/Tx) logic 0	+3 V to +15 V	0 V
Data transmission (Rx/Tx) logic 1	-3 V to -15 V	5 V
Control signals (RTS/CTS/DTR/DSR) logic 0	-3 V to -15 V	5 V
Control signals kjnic 1	+3 V to +15 V	0 V

Figure 3.12 MAX 232 Voltage Levels

3.7.2 Pin Diagram

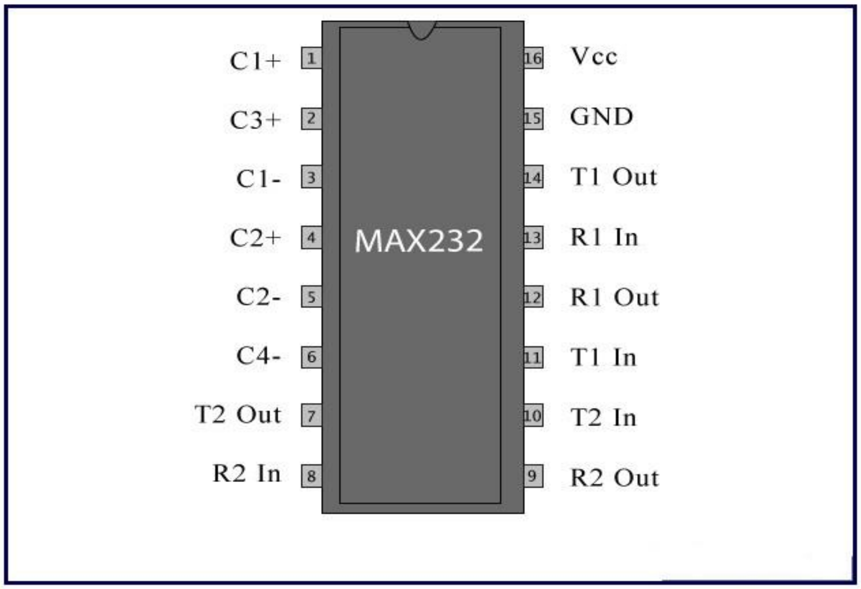


Figure 3.13 MAX 232-Pin Configuration

3.7.3 Pin Description

Pin No	Function	Name
1	Capacitor connection pins	Capacitor 1 +
2		Capacitor 3 +
3		Capacitor 1 -
4		Capacitor 2 +
5		Capacitor 2 -
6		Capacitor 4 -
7	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_2 Out
8	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_2 In
9	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_2 Out
10	Input pins; receive the serial data at TTL logic level; connected to serial transmitter pin of controller.	T_2 In
11		T_1 In
12	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_1 Out
13	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_1 In
14	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_1 Out
15	Ground (0V)	Ground
16	Supply voltage; 5V (4.5V – 5.5V)	Vcc

Figure 3.14 MAX 232-Pin Description

3.8 ULN 2803

A ULN2803 is an Integrated Circuit (IC) chip with a High Voltage/High Current Darlington Transistor Array. It allows you to interface TTL signals with higher voltage/current loads. In English, the chip takes low level signals (TLL, CMOS, PMOS, NMOS - which operate at low voltages and low currents) and acts as a relay of sorts itself, switching on or off a higher level signal on the opposite side.

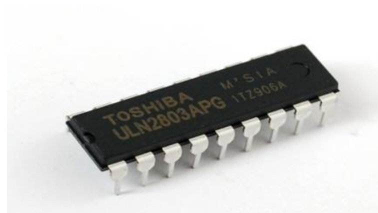


Figure 3.15 ULN 2803

A TTL signal works from 0-5V, and all the things between 0.0 and 0.8V considered "low" or off, and 2.2 to 5.0V being considered "high" or on. The maximum power available on a TTL signal depends on the type,but it doesn't generally moves ahead of 25mW (~5mA @ 5V), so it is not useful for providing power to something like a relay coil. Computers and other electronic devices frequently generate TTL signals. On the output side the ULN2803 is generally rated at 50V/500mA, so it can operate small loads directly. Alternatively, mostly used to power the coil of one or more relays, which in turn allow even higher voltages/currents to be controlled by the low level signal. In electrical terms, the ULN2803 uses the low level (TTL) signal to switch on/turn off the higher voltage/current signal on the output side.

The ULN2803 comes in an 18-pin IC design and consists of (8) transistors. Pins 1-8 receive the low level signals. Pin 9 is grounded (for the low level signal reference). Pin 10 is the common on the high side and would generally be connected to the positive of the voltage you are applying to the relay coil. Pins 11-18 are the outputs (Pin 1 drives Pin 18, Pin 2 drives 17, etc.).

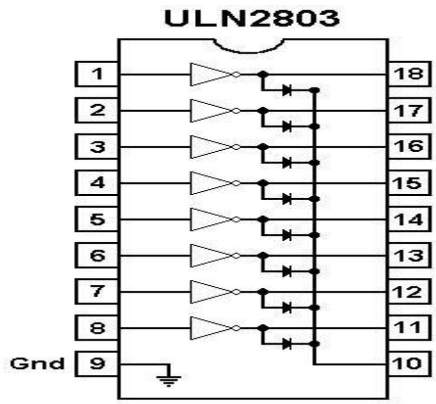


Figure 3.16 ULN 2803-Internal Circuit

3.9 Relay

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

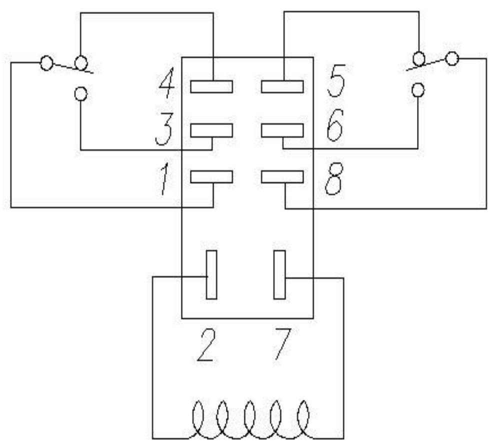


Figure 3.17 Relay Operation

3.9.1 Applications

Relays are used for:

a. Amplifying a digital signal, switching a large amount of power with a small operating power. Some special cases are:
   - A telegraph relay, repeating a weak signal received at the end of a long wire.
   - Controlling a high-voltage circuit with a low-voltage signal, as in some types of modems or audio amplifiers.
   - Controlling a high-current circuit with a low-current signal, as in the starter solenoid of an automobile.

b. Detecting and isolating faults on transmission and distribution lines by opening and closing circuit breakers (protection relays).

c. Isolating the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors to conserve energy.

d. Vehicle battery isolation. A 12v relay is often used to isolate any second battery in cars, 4WDs, RVs and boats.

e. Switching to a standby power supply.

3.10 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), animations and so on.

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. Click to learn more about internal structure of a LCD.

3.10.1 Pin Diagram

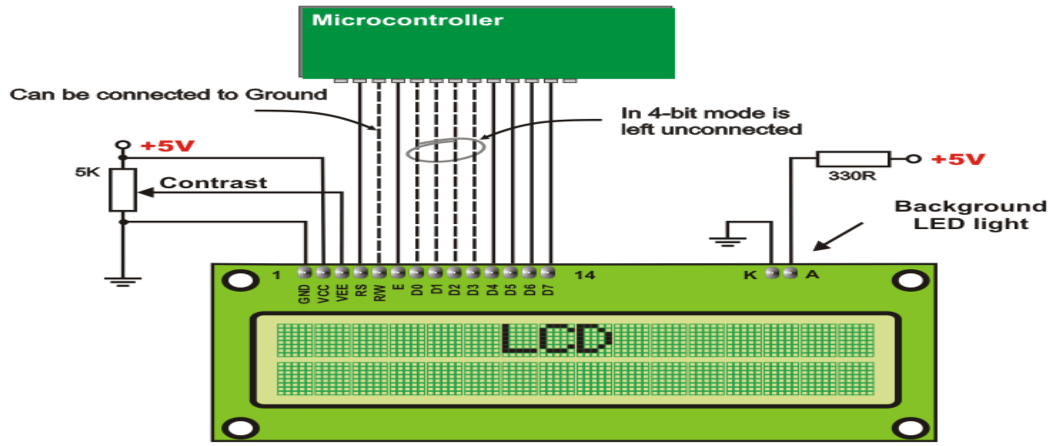


Figure 3.18 LCD-Pin Configuration

3.10.2 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-

Figure 3.19 LCD-Pin Description

CHAPTER 4: METHODOLOGY

4.1 Methodology

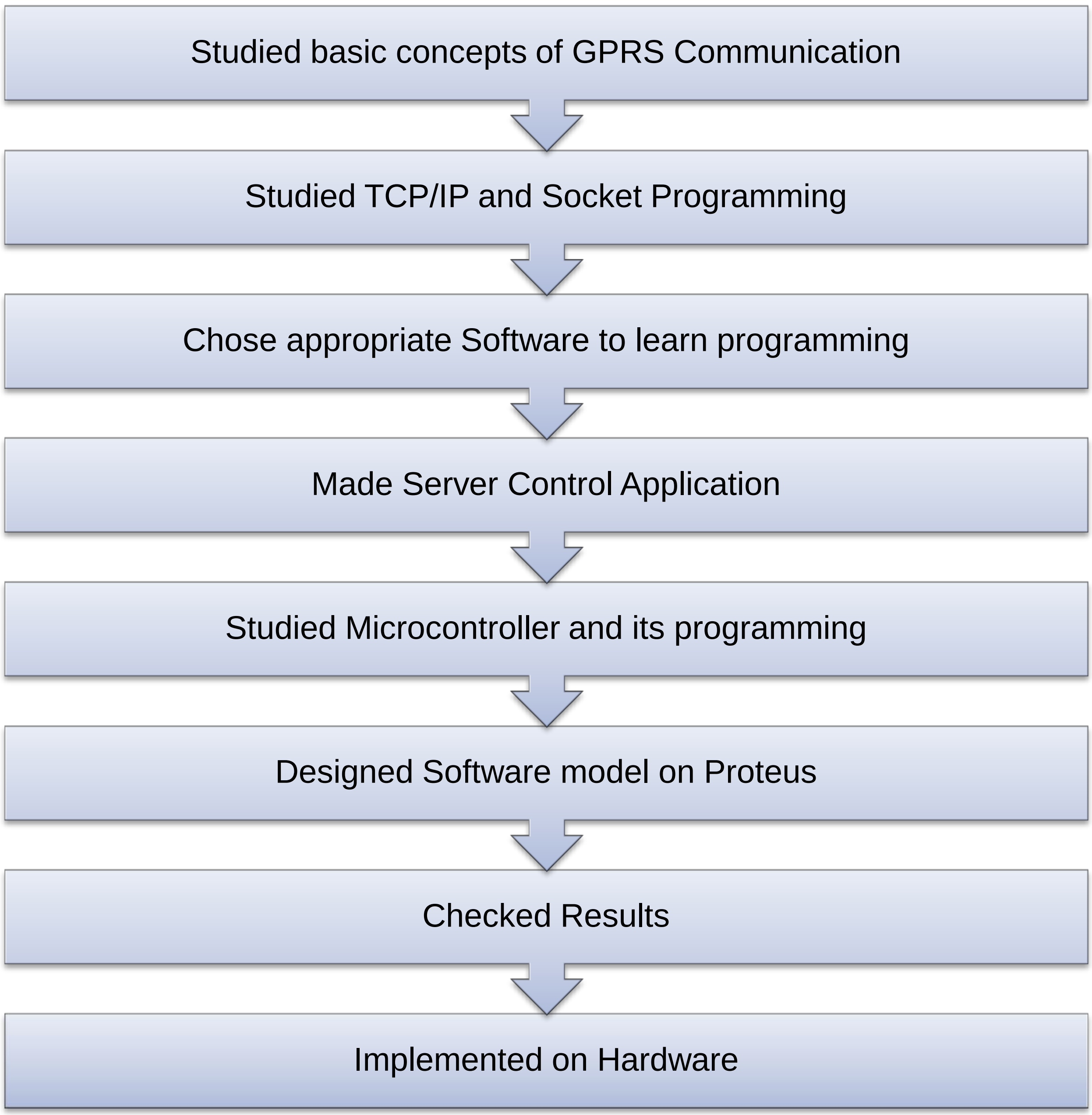


Figure 4.1 Methodology

4.2 GPRS connection setup for internet using HyperTerminal

GPRS module SIM300 uses serial communication and can be connected to computer through serial port. On computer we run HyperTerminal to send AT commands to SIM300 in order to perform various functions. These are the AT commands to establish a connection between GPRS module and a computer with known IP address.

AT

AT+CGATT=1 ..... \\ Attach or detach from GPRS service

AT+CGDCONT=1,"IP""INTERNET" ..... \\ Define Packet data protocol type (Internet protocol)

AT+CSTT="INTERNET" ..... \\ a string parameter which indicates the GPRS access point name, GPRS user name, GPRS password.

AT+CIICR ..... \\ Bring up wireless connection with GPRS or CSD

AT+CIFSR ..... \\ Get local IP address

AT+CDNSORIP=0 ..... \\Connect with IP address or domain name server
0 remote server is an IP address
1 remote server is a domain name

AT+CIPSTART="TCP","182.188.135.205",1000.....\\<mode> a string parameter which indicates the connection type
“TCP” Establish a TCP connection
“UDP” Establish a UDP connection

<IP address> remote server IP address
<port> remote server port
<domain name> remote server domain name
<state> a string parameter which indicates the progress of connecting

4.3 Interfacing GPRS modem with Microcontroller

Now we have to automate the task of connection establishment between the GPRS and end user by the help of microcontroller. Microcontroller is to be programmed is such a way that it will be provided with the IP address of the user and using that IP it will connect to the user. As GPRS does not have its own static IP this is the best method to establish the connection.[7]

SIM300 is a TTL compatible GSM/GPRS module and hence can be directly integrated with microcontroller. SIM300 communicates serially with microcontroller. Serial data communication is of two types, asynchronous and synchronous. The synchronous method transfers a block of data at a time whereas the asynchronous method transfers a single byte at a time. Special IC chips are built in the microcontrollers for serial communication. These chips are called UART and USART.

4.3.1 Asynchronous serial communication

Asynchronous serial communication is widely used for character oriented transmission. In the asynchronous communication each character is placed between start and stop bit. This is called framing. The start bit is always one bit but stop bits can be one or two bits.

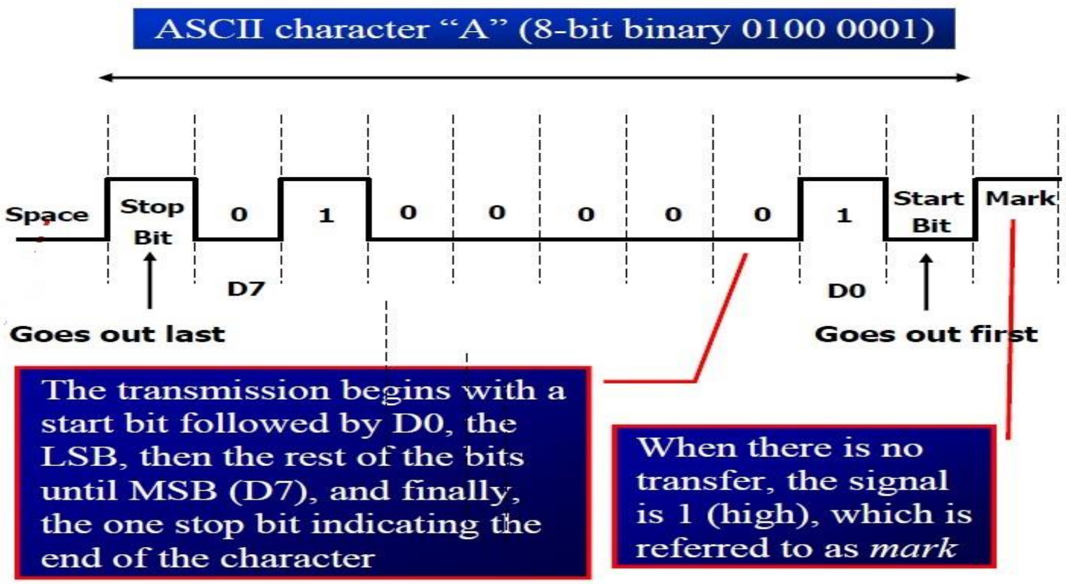


Figure 4.2 Asynchronous serial communication

4.3.2 PIC18 Serial Port Programming in C

USART of PIC18 has both synchronous and Asynchronous communication. To communicate with SIM300 we will be using asynchronous method of communication. There are a number of registers that control serial communication and its features.[8]

4.3.3 RX and TX pin of PIC18

PIC 18 has two pins RX and TX for receiving and transmitting data. These pins belong to PORTC (RC6 and RC7). These pins are connected to the RX and TX of Sim300 module. RX of PIC18 is connected to TX of sim300 and TX pin of PIC18 is connected to RX of Sim300.

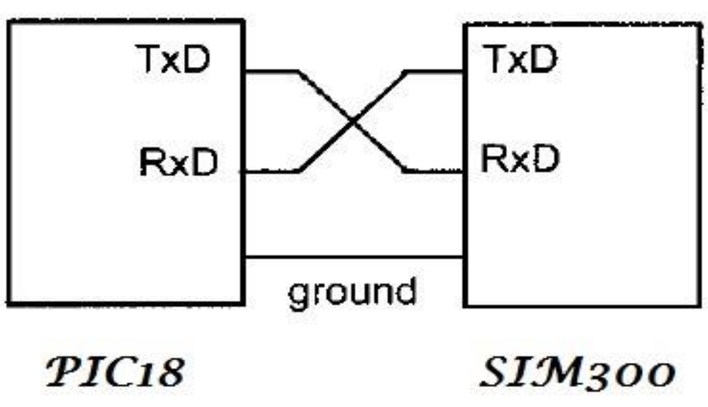


Figure 4.3 PIC18 connection with sim300

4.3.4 SPBRG Register and BAUD RATE setting in PIC18:

Baud rate in PI18 is programmable. This is done by an 8-bit register called SPBRG register. For a given crystal frequency the value loaded into the SPBRG decides the baud rate. The relation between the crystal frequency and baud rate is given as:

$$\text{Desired Baud Rate} = \text{Fosc}/(64X + 64) = \text{Fosc}/64(X + 1)$$

Where X is the value loaded into the SPBRG. Some of the Baud rates supported by PC are given below:

PC Baud Rates in HyperTerminal
1,200
2,400
4,800
9,600
19,200
38,400
57,600
115,200

4.3.5 TXREG Register

TXREG is another 8-bit register in PIC18 used to transmit data. As soon as data is loaded into the TXREG register is serially transmitted by a Transfer Shift Register. 8-bit data is added with one start and one or two stop bits.

4.3.5 RXREG Register

RXREG is another 8-bit register used to receive data serially. As soon as a byte of data is received it is loaded into RXREG register. Data should be fetched from RXREG so that next byte could be received.

4.3.6 TXSTA (Transmit Status and Control Register)

The TXSTA register is an 8-bit register used to select synchronous /asynchronous modes and data framing. 8-bits of TXSTA along with the function of bits is given below.

CSRC	TX9	TXEN	SYNC	0	BRGH	TRMT	TX9D

CSRC	D7	Clock Source Select (not used in asynchronous mode, therefore D7 = 0.)
TX9	D6	9-bit Transmit Enable 1 = Select 9-bit transmission 0 = Select 8-bit transmission (We use this option, therefore D6 = 0.)
TXEN	D5	Transmit Enable 1 = Transmit Enabled 0 = Transmit Disabled We turn "on" and "off" this bit in order to start or stop data transfer.
SYNC	D4	USART mode Select (We use asynchronous mode, therefore D4 = 0.) 1 = Synchronous 0 = Asynchronous
0	D3	
BRGH	D2	High Baud Rate Select 0 = Low Speed (Default) 1 = High Speed We can double the baud rate with the same Fosc. See the end of this section for further discussion on this bit.
TRMT	D1	Transmit Shift Register (TSR) Status 1 = TSR empty 0 = TSR full

4.3.7 RXSTA (Receive Status and Control Register)

RXSTA is another 8-bit register used to enable serial port along other functions. Function of each bit is explained below:

SPEN	RX9	SREN	CREN	ADDE	FERR	OERR	RX9D

SPEN	D7	Serial port enable bit 1 = Serial port enabled, which makes TX and RX pins as serial port pins 0 = Serial port disabled
RX9	D6	9-bit Receive enable bit 1 = Select 9-bit reception 0 = Select 8-bit reception (We use this option; therefore, D6 = 0.)
SREN	D5	Single receive enable bit (not used in asynchronous mode D5 = 0)
CREN	D4	Continuous receive enable bit 1 = Enable continuous Receive (in asynchronous mode) 0 = Disable continuous Receive (in asynchronous mode)
ADDEN	D3	Address delete enable bit (Because used with the 9-bit data frame D3 = 0)
FERR	D2	Framing error bit 1 = Framing error 0 = No Framing error
OERR	D1	Overrun error bit 1 = Overrun error 0 = No overrun error
TXD9	D0	9th bit of Receive data (Because we use the 8-bit option, we make D0 = 0) Can be used as an address/data or a parity bit in some applications.

4.3.8 Programming in C

Function to Initialize USART:

```
void InitUART(int baud)
{
        TRISC6 = 1;                                    // TX Pin
        TRISC7 = 1;                                    // RX Pin
        SPBRG=(int)(_XTAL_FREQ/(64.0*baud)-1);
        BRGH  = 1;                                     // Fast baudrate
        SYNC  = 0;                                     // Asynchronous
        SPEN  = 1;                                     // Enable serial port pins
        CREN  = 1;                                     // Enable reception
```

```
    SREN  = 0;                          // No effect
    TXIE  = 0;                          // Disable tx interrupts
    RCIE  = 1;                          // Enable rx interrupts
    TX9   = 0;                          // 8-bit transmission
    RX9   = 0;                          // 8-bit reception
    TXEN  = 0;                          // Reset transmitter
    TXEN  = 1;                          // Enable the transmitter
}
```

Function To Transmit String:

```
void TransmitString(const unsigned char* st)
{
    while(*st)
        Transmit(*st++);
}
```

Function To Receive Data:

```
unsigned char ReceiveByteSerially(void)   // Reads a character from the serial port
{
    if(OERR) // If over run error, then reset the receiver
    {
        CREN = 0;
        CREN = 1;
    }
    while(!RCIF);  // Wait for transmission to receive
    return RCREG;
}
```

4.3.9 PIC18 Interrupt Service Routine (ISR)

When a TCP/IP connection has been established between end user and the GSM/GPRS modem data can be received or transmitted in both directions. End applications send ASCII characters to GPRS modem. GPRS modem sends these characters serially to microcontroller. Microcontroller on receiving these characters makes decisions. This is done in interrupt service routine ISR. Whenever a byte is received serially an interrupt is generated and CPU leaves its work and handles the interrupt. Upon reading the received character it makes decisions.

Interrupt Service Routine Function:

```
void interrupt isr(void)
{
        if(RCIF==1)
    {       rec=RCREG;
              if (rec==0x01)                 // for Up arrow in C# application
                {       FWD=~FWD;           // move forward
                        RCIF=0;              // clear interrupt flag
                }
                if (rec==0x04)               // for Down Arrow in C# application
                {       BWD=~BWD;            // move backward
                        RCIF=0;              // clear interrupt flag
                }
                if (rec==0x02)               // for Right Arrow in C# application
                {       RT=~RT;              // turn right
                        RCIF=0;              // clear interrupt flag
                }
                if (rec==0x03 )              // for Left Arrow in C# application
                {       LT=~LT;              // turn left
                        RCIF=0;              // clear interrupt flag
```

```
}
if(rec==0x05)                    // Turn OFF All LED's
{
        FWD=0;
        BWD=0;
        RT=0;
        LT=0;
        RCIF=0;             // clear interrupt flag
}
```

4.4 User Application

This project is based on TCP/IP communication between a remote IP (GPRS) and an internet user (server). We have to develop a User application that makes a TCP/IP connection[9] with the GPRS modem. TCP/IP connection uses SOCKET ADDRESSES. Socket addresses consist of IP address and port number. We developed a C# application in which we have defined the port number. We give the same port number to the GPRS modem through microcontroller. The application give the present IP address by which the user is connected to internet.

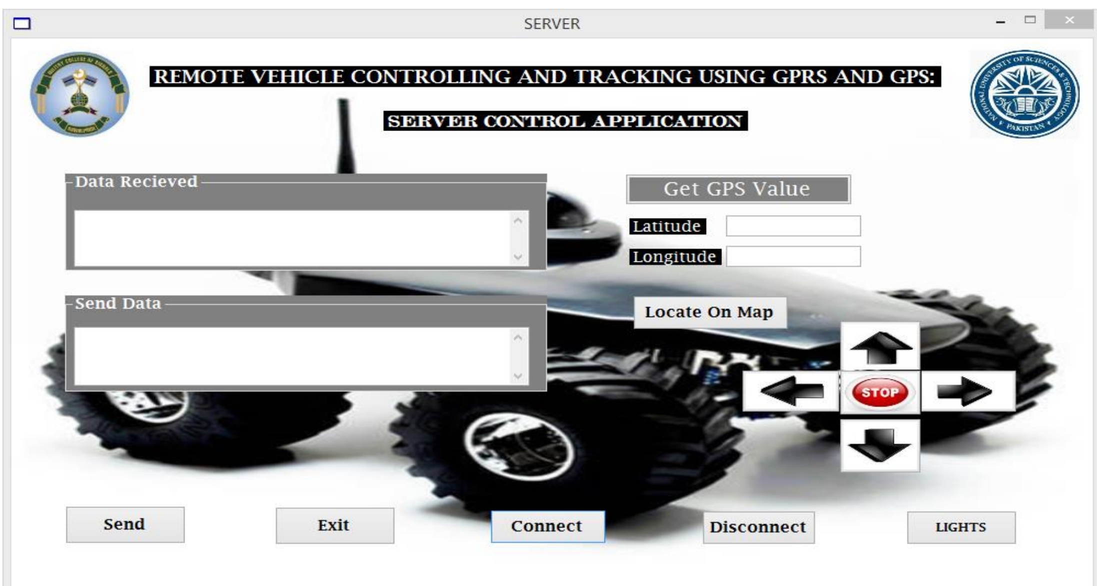


Figure 4.4 Server IP

4.5 GPS Module Interfacing with Microcontroller

Global Positioning system module receives coordinates from satellite and transmits a string serially through TX pin. HOLUX M8929 GPS has a baud rate of 4800. This TX pin of GPS is connected to RX pin of microcontroller. Microcontroller extracts the longitude and latitude from the string.

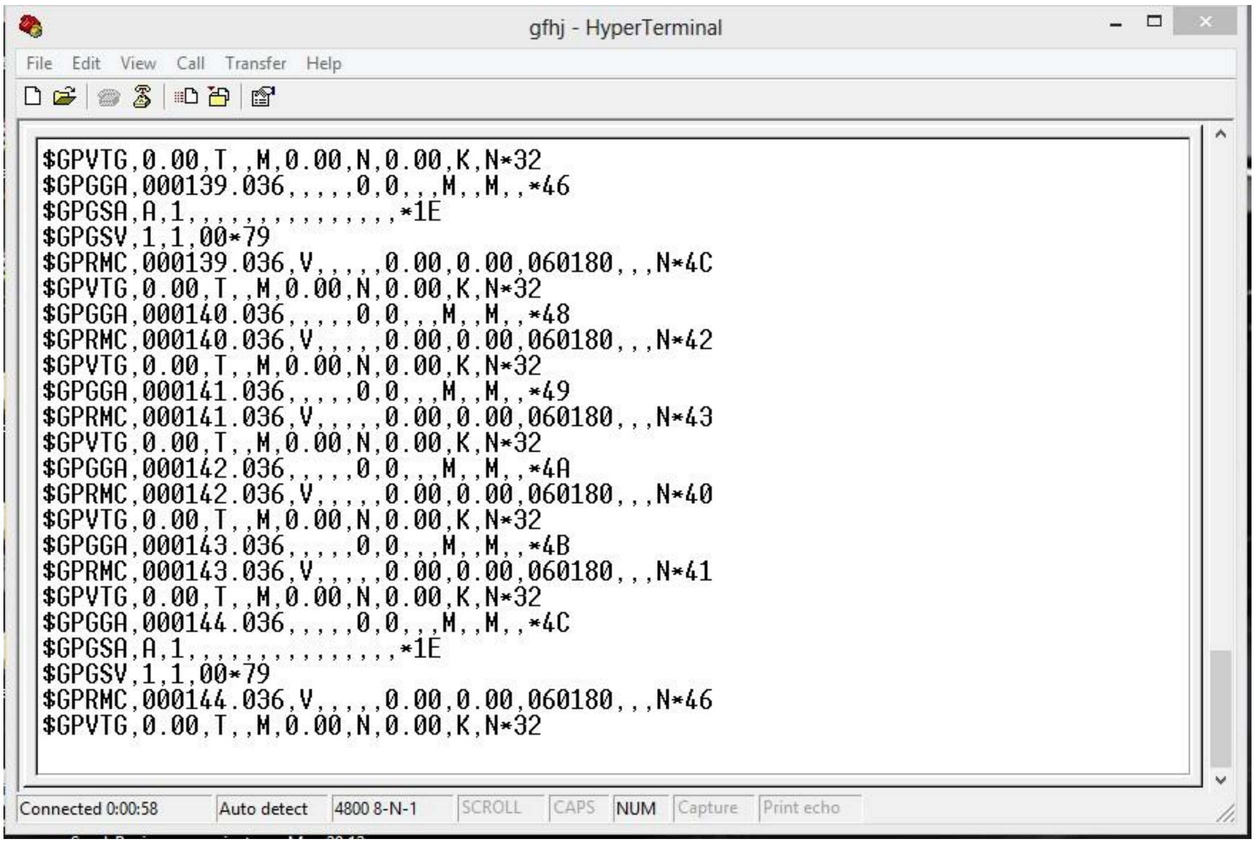


Figure 4.5 GPS connection using HyperTerminal

Coordinates received by microcontroller are sent to user application using GPRS. Sim300 AT commands for sending data after a TCP/IP connection has been established are:

AT+CIPSEND // SEND DATA THROUGH TCP OR UDP CONNECTION

GPS Data sending Function:

```
void send_gps_data()
{
	TransmitString("at\r");
```

```
        TransmitString("at+cipsend\r");
        TransmitString("longitude, latitude");
        Transmit(0x1A);
        RCIF=0;                         // clear interrupt flag
}
```

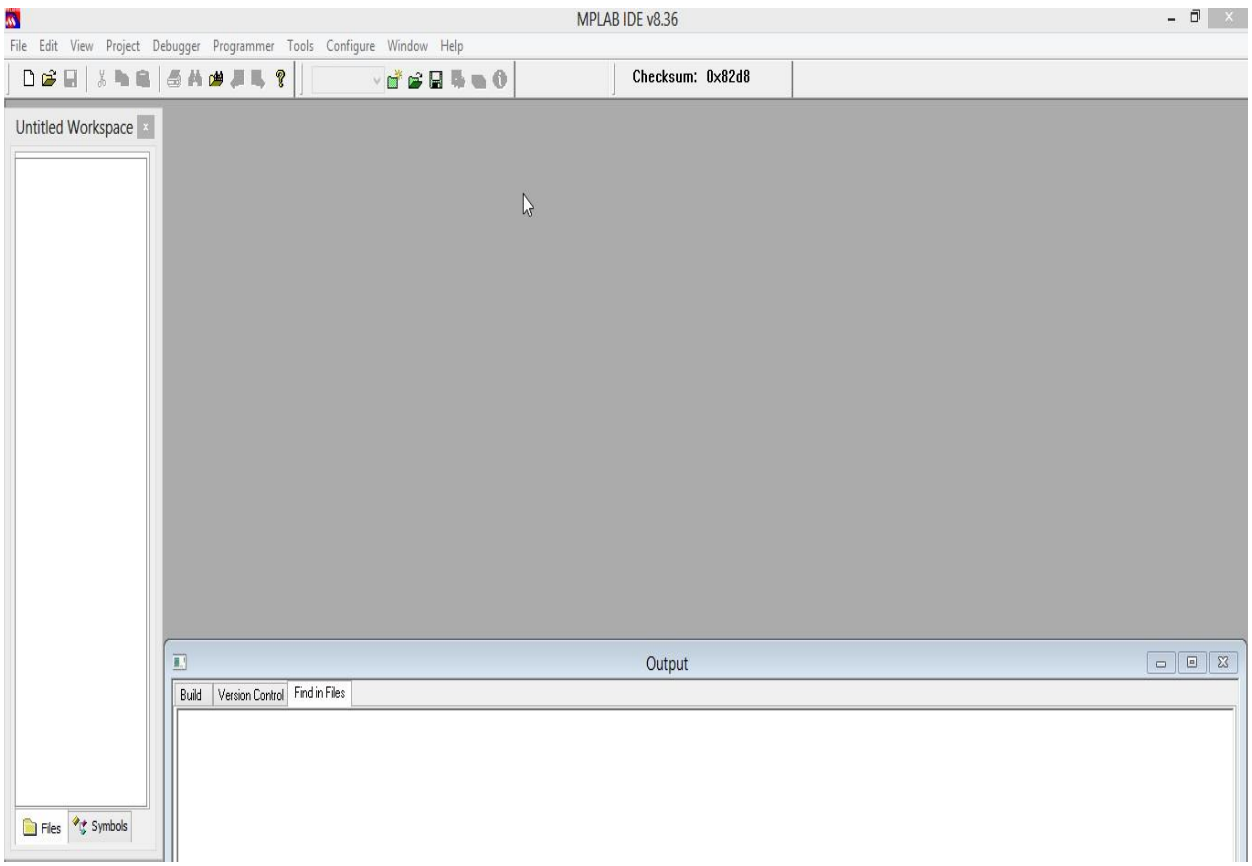
CHAPTER 5: SOFTWARE SIMULATION AND APPLICATION

5.1 MPLAB IDE and HI-TECH C compiler for PIC18:

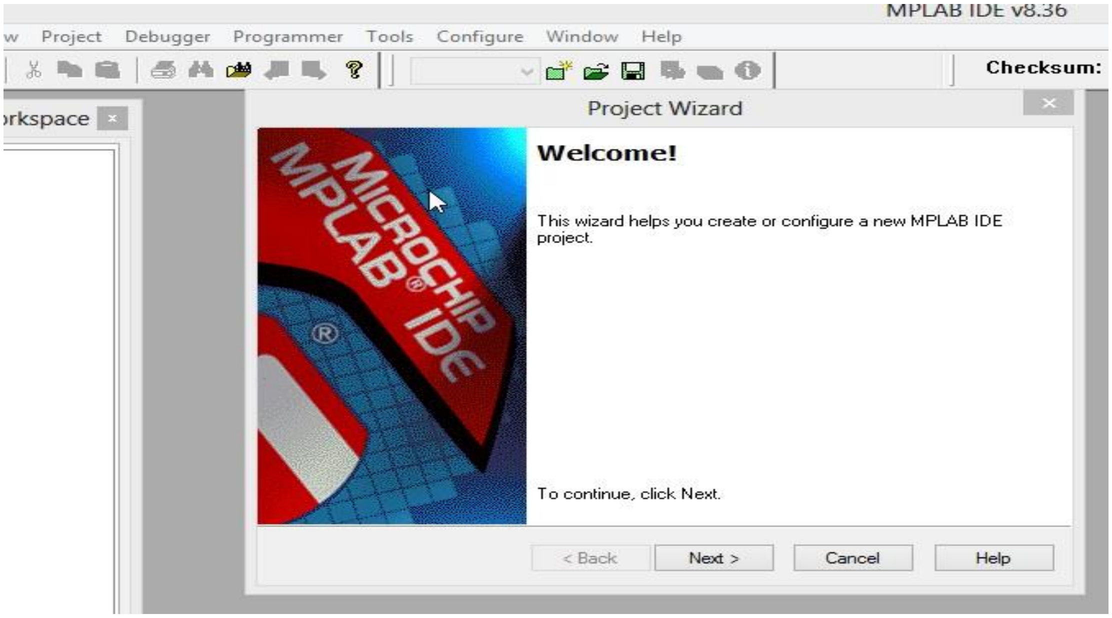
MPLAB IDE is microchip recommended software for compiling codes for Microchip devices. Along with MPLAB IDE we can use C18 compiler or HI-TECH C compiler for simulation of codes written in C language.

HI-TECH C compilers are strong compliers having a huge internal library. For our project we used MPLAB IDE along with HI-TECH C compilers. Here a brief introduction is given using these tools.

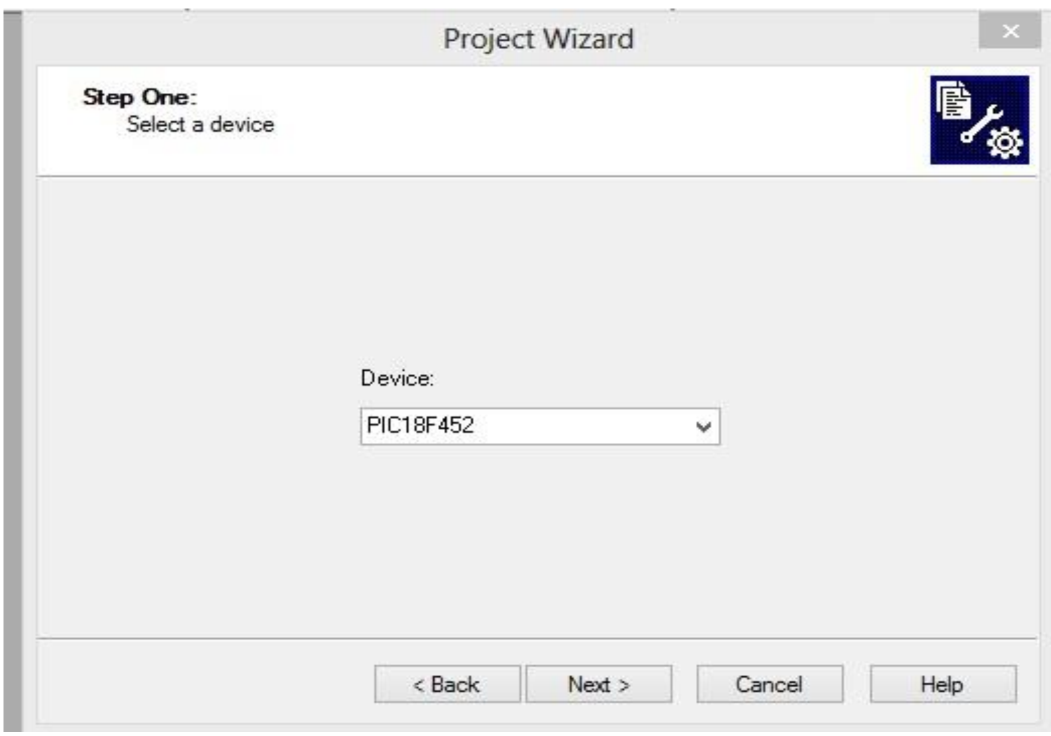
5.1.1 Open MPLAB IDE environment



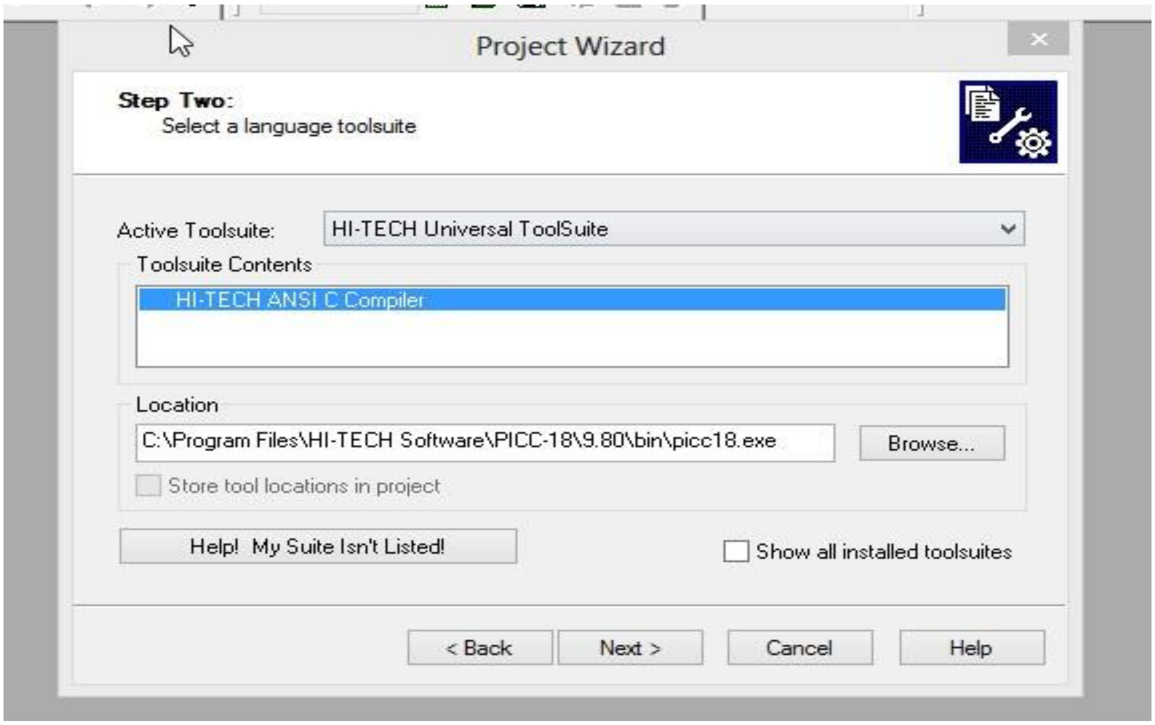
5.1.2 Go to Project > project wizard > create project and a new window will open click next



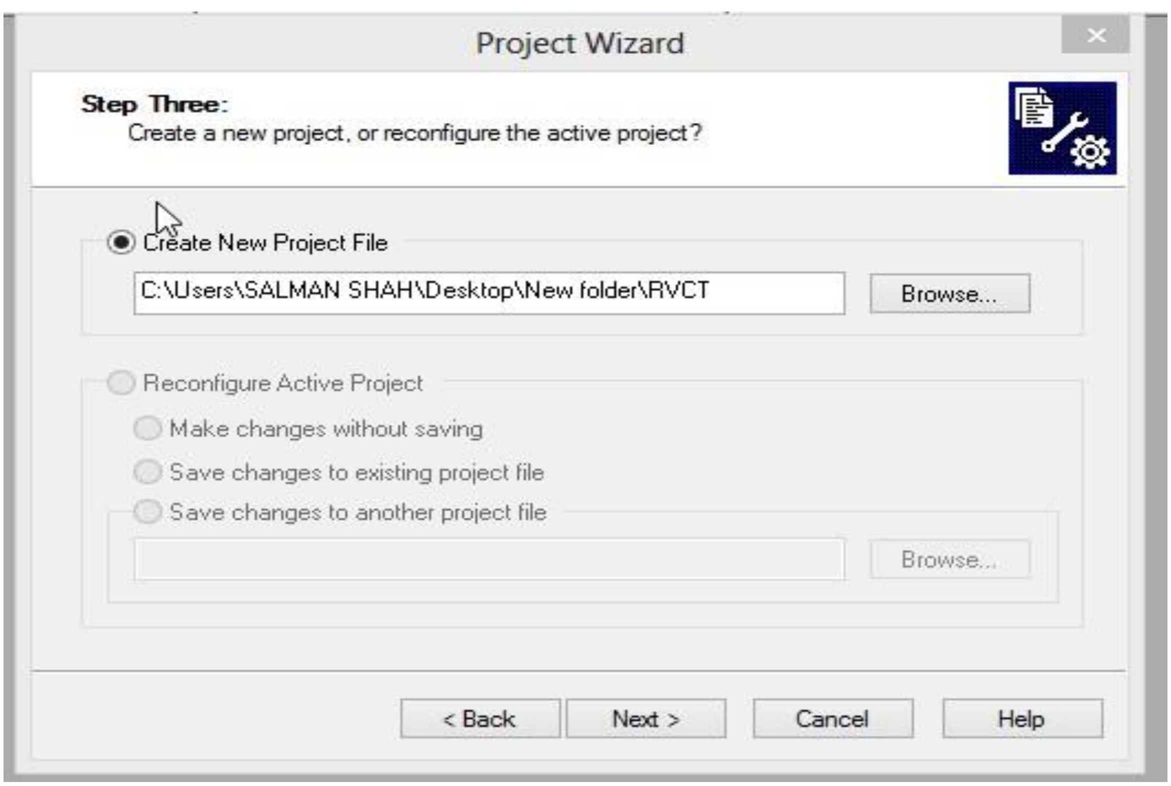
5.1.3 Select device from the list. We are using PIC18F452. Click next.



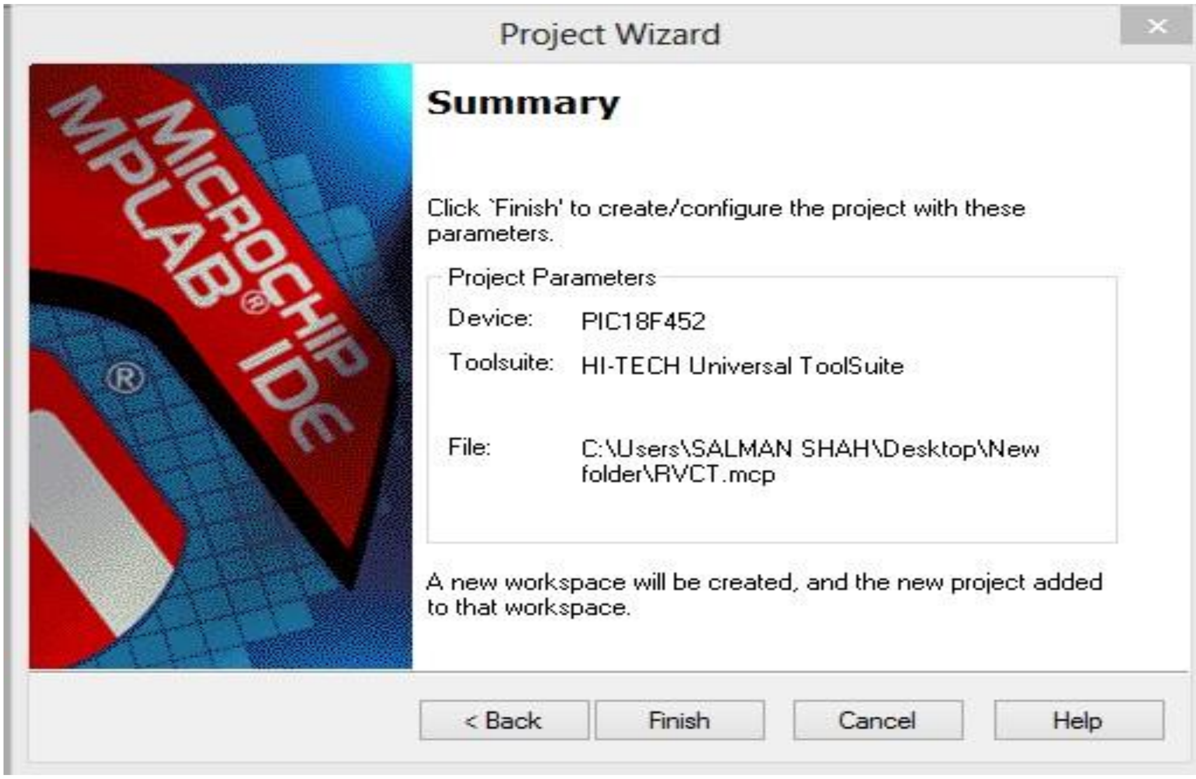
5.1.4 Select language tool. For c language code select HI-TECH C compilers



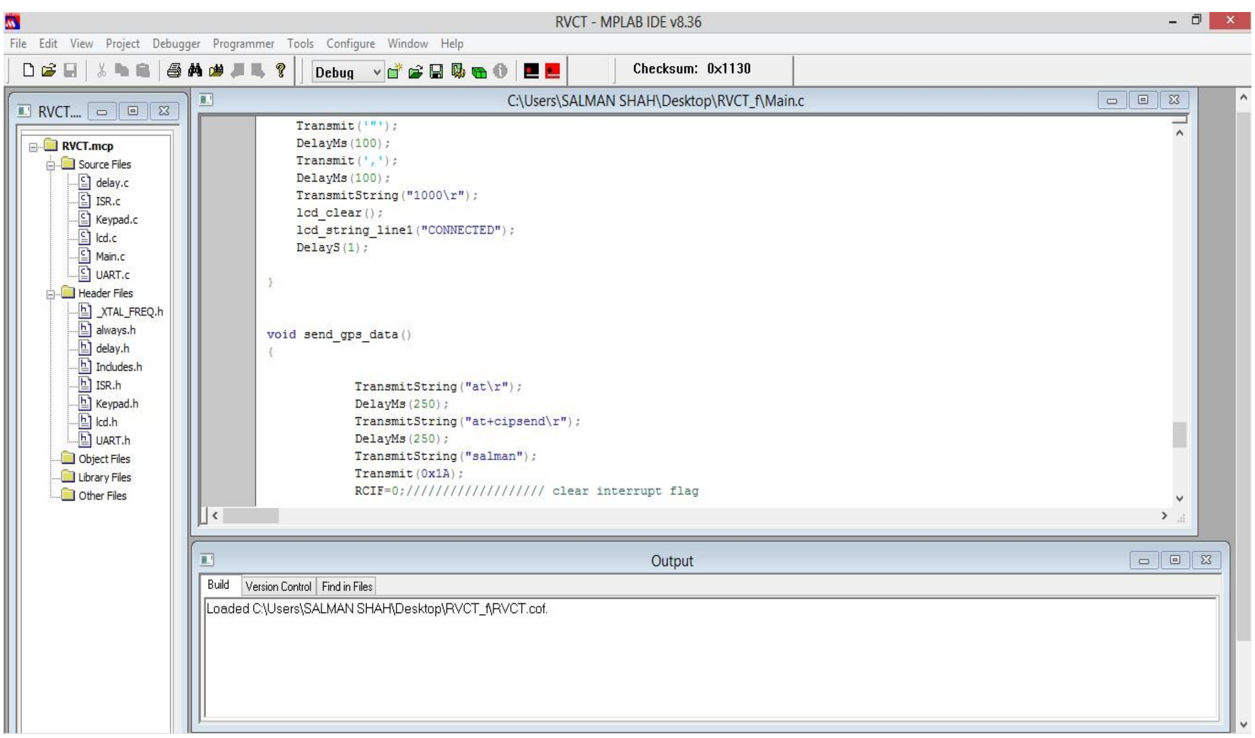
5.1.5 Select location to save the new project files and name your project



5.1.6 Click finish to open your project



5.1.7 Here is your project window



5.2 Proteus Simulation

Once the code has been compiled and a HEX file is created we can check the working of our code in Proteus. Proteus is a highly professional simulator and PCB designer.

In our project we have connected the GSM module to the serial port of our computer. In Proteus we give the path of that physical port. Using virtual terminal we can see what kind of data is send or received serially. A 4x3 keypad is connected with the microcontroller to give IP address. In place of DC motors we have placed LEDs just for indication that the required function is being performed. A 16x2 LCD is connected with microcontroller which gives the display.

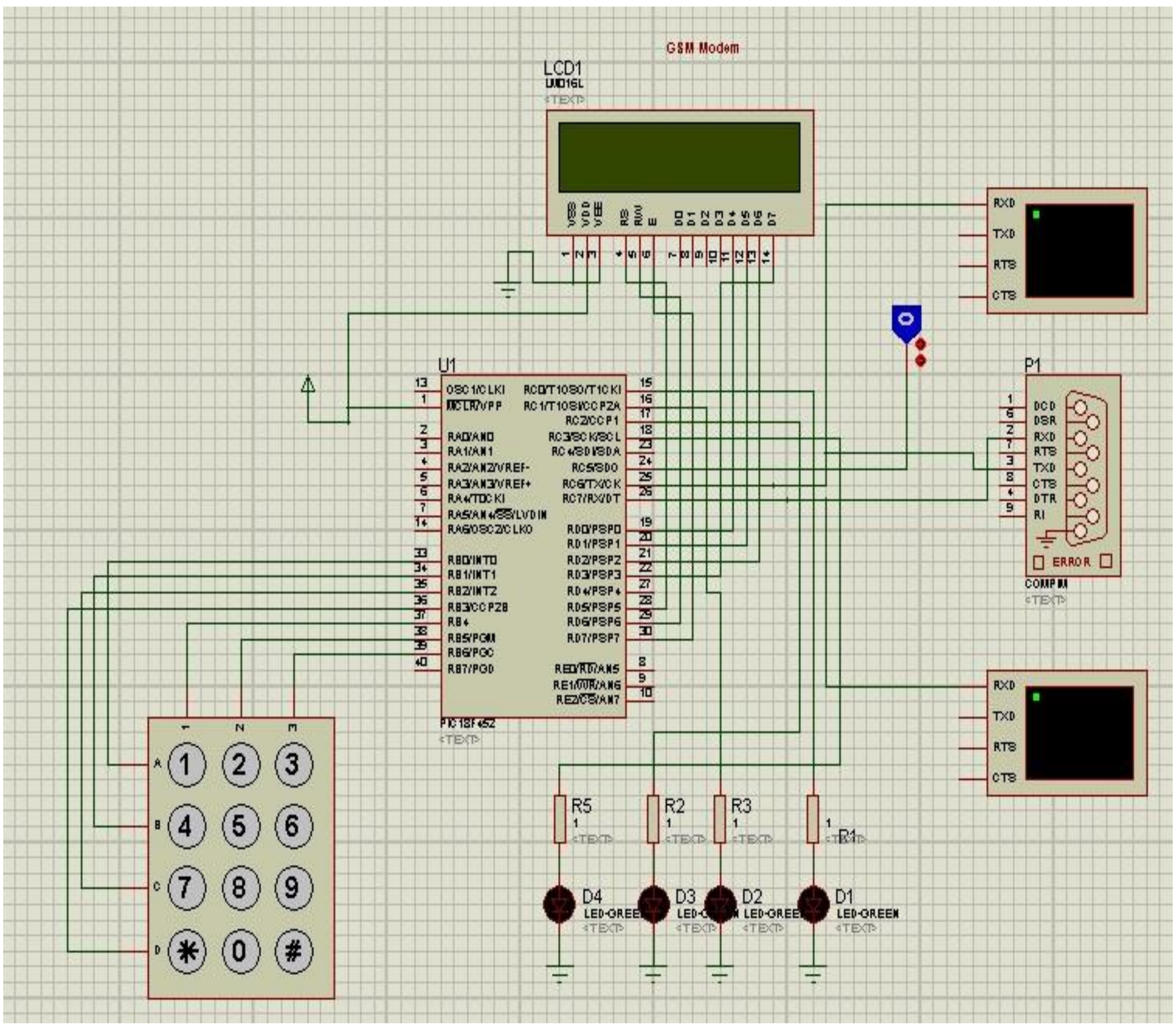
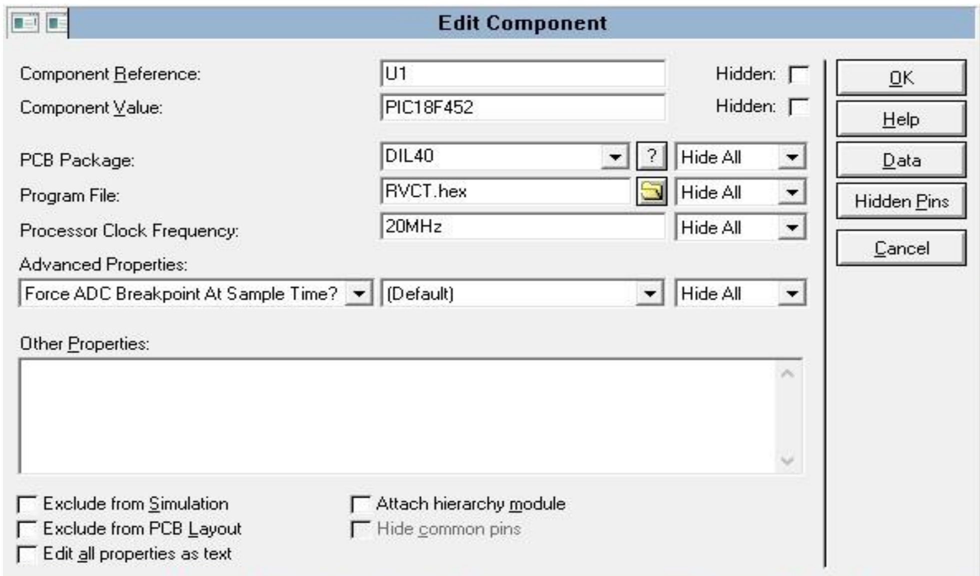
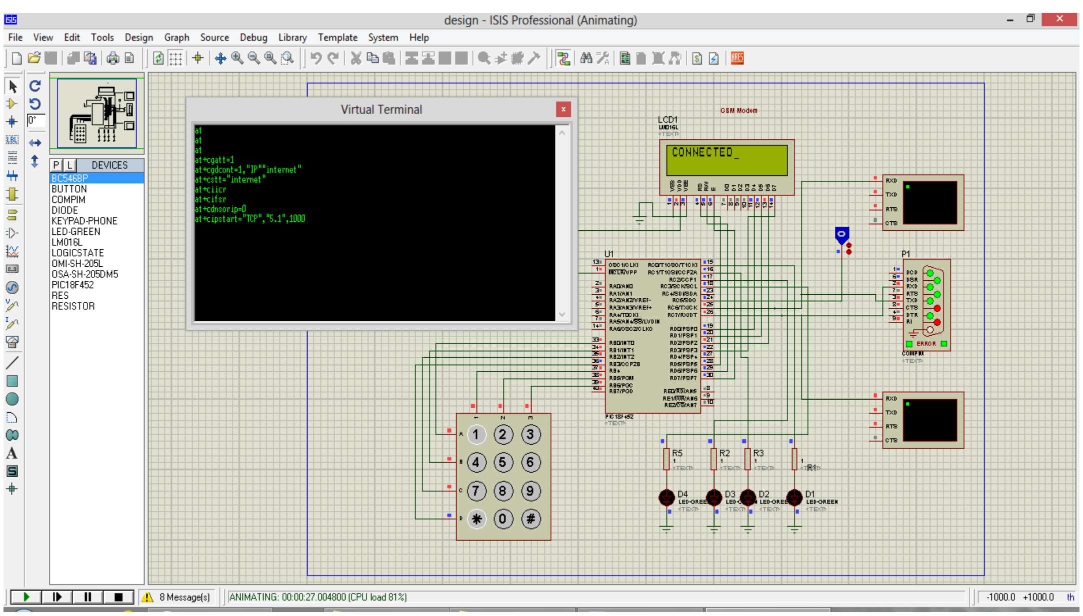


Figure 5.1 Proteus Simulation

After making the right connections of the devices with microcontroller the HEX file is loaded into the microcontroller and the required clock frequency is set as shown in the diagram below.



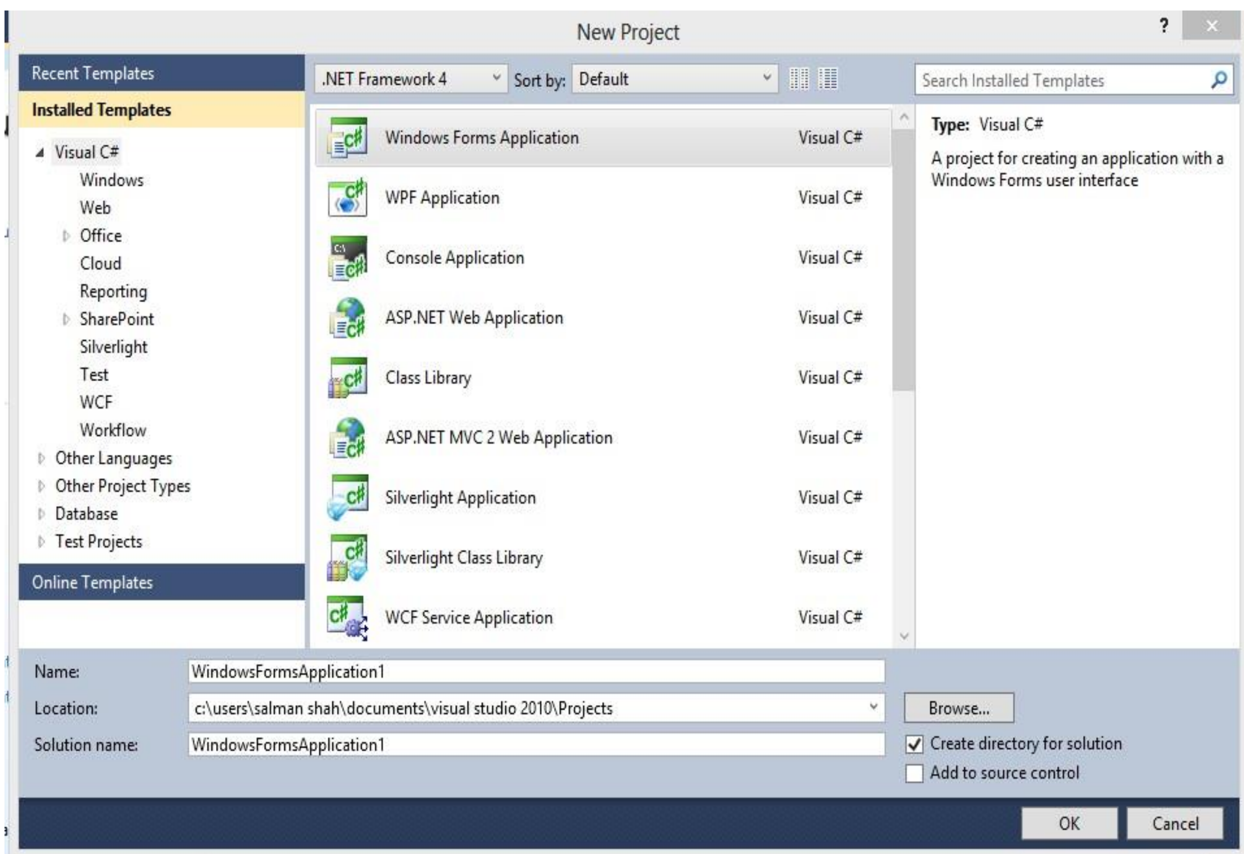
After loading the HEX file into the microcontroller we run the simulation. On receiving the commands the LEDs turn on and off. Once the simulation is working properly the same configuration can run on hardware equally well. The final design of our Proteus simulation is shown below.



5.3 Server Control Application using Microsoft Visual Studio

Microsoft visual studio is an IDE (Integrated Develop Environment) means that coding, compiling, linking and debugging is done with the same development platform. Microsoft Visual Studio supports various programming languages foe example Visual Basic, C/C++, Java, C# / .NET. A number of windows applications can be built using this tool for example windows Form Application, WPF Application ASP.NET MVC 2 Web Application silver light application WCF service application etc.

Application is an implementation of socket programming. Socket programming is related with the socket addresses. Socket addresses consist of an IP address and a port number. There is a need of public IP address to make a connection with a remote IP. If we are using a private IP address we have to NAT it. Port number can be any number between 1 and 65535 but reserved port numbers, like port 80 , 23 etc. should be avoided.



We have made windows Form Application of our project. It establishes a TCP/IP connection with the remote IP and on establishing the connection it can send and receive data.

There is a connect button to connect , on clicking this connect button the application displays the current IP address of user by which it is connected to internet. On establishment of TCP connection with the remote IP the application displays “server started listening”. Here is the code behind the connect button.

```
private void btnstart_Click(object sender, EventArgs e)
  {
      Dns.GetHostName();
      IPHostEntry ihe = Dns.GetHostByName(Dns.GetHostName());
      IPAddress myip = ihe.AddressList[0];
      MessageBox.Show(myip.ToString());

      try
      {
          IPEndPoint myendpoint = new IPEndPoint(myip,1000);
          server.Bind(myendpoint);
          server.Listen(1);
          client = server.Accept();
          MessageBox.Show("Server Started Listening");
          count = 1;
      }
 }
```

We have created six controls for the application: forward, backward, right, left lights 0n/off, horn on/off. On clicking any of these buttons an ASCII character is send behind it.

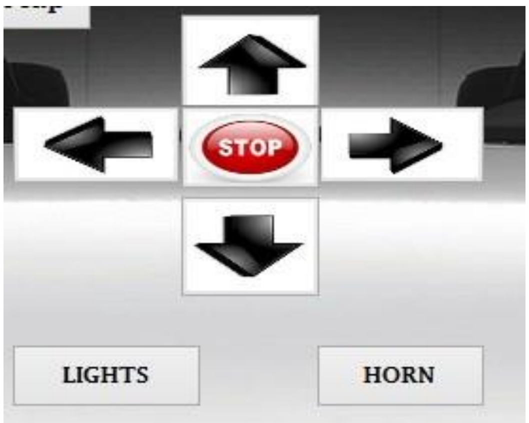


Figure 5.2 6 control buttons

Code behind these buttons is given below:

```
private void button1_Click(object sender, EventArgs e)
{
    byte[] value = {0x01};
    client.Send(value);
}

private void button3_Click(object sender, EventArgs e)
{
    byte[] value = { 0x02 };
    client.Send(value);
}

private void button2_Click_1(object sender, EventArgs e)
{
    byte[] value = { 0x03 };
    client.Send(value);
}

private void button4_Click(object sender, EventArgs e)
{
    byte[] value = { 0x04 };
    client.Send(value);
}

private void button5_Click(object sender, EventArgs e)
{
    byte[] value = { 0x08 };
    client.Send(value);
}

private void button6_Click(object sender, EventArgs e)
{
    byte[] value = { 0x05 };
    client.Send(value);
}
```

Second important function performed by this application is that it receives coordinates from the remote vehicle and locates them on Google maps.

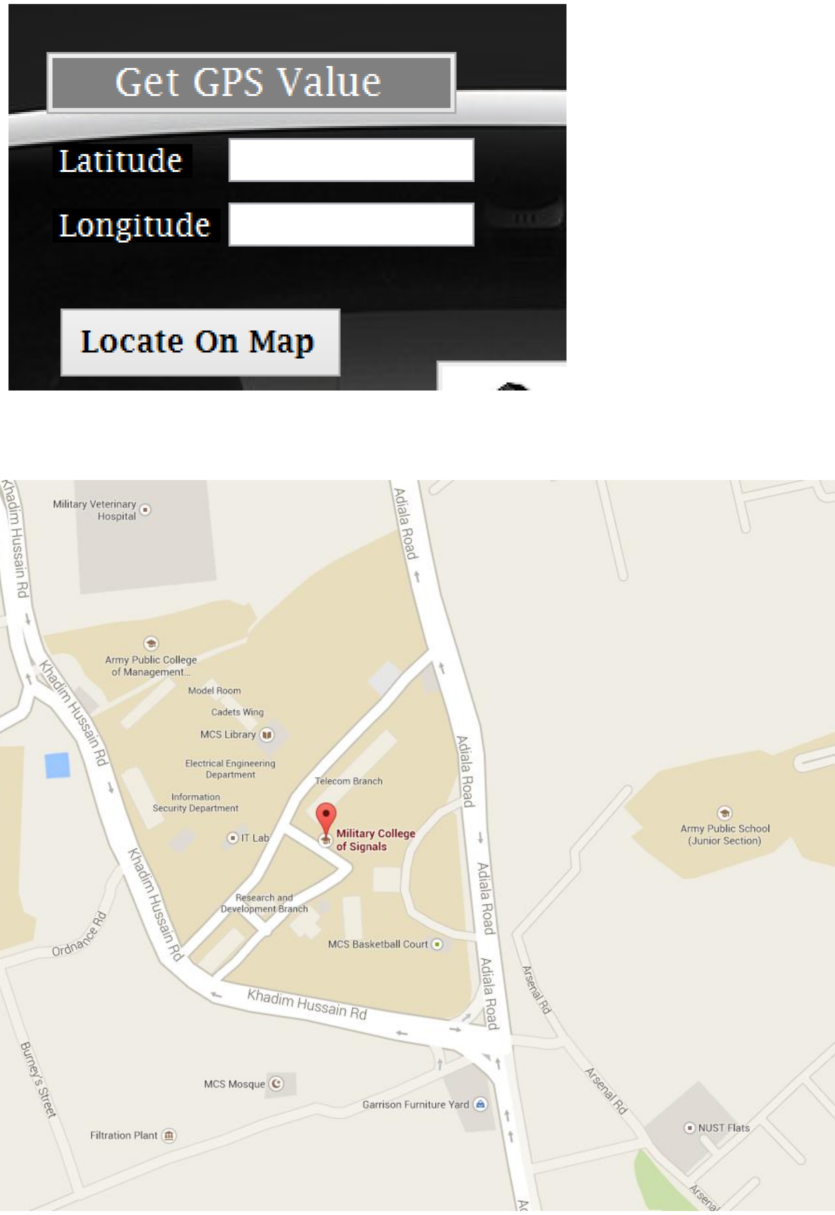


Figure 5.3 Location tracking on Google map

This is the final design of our server control application.

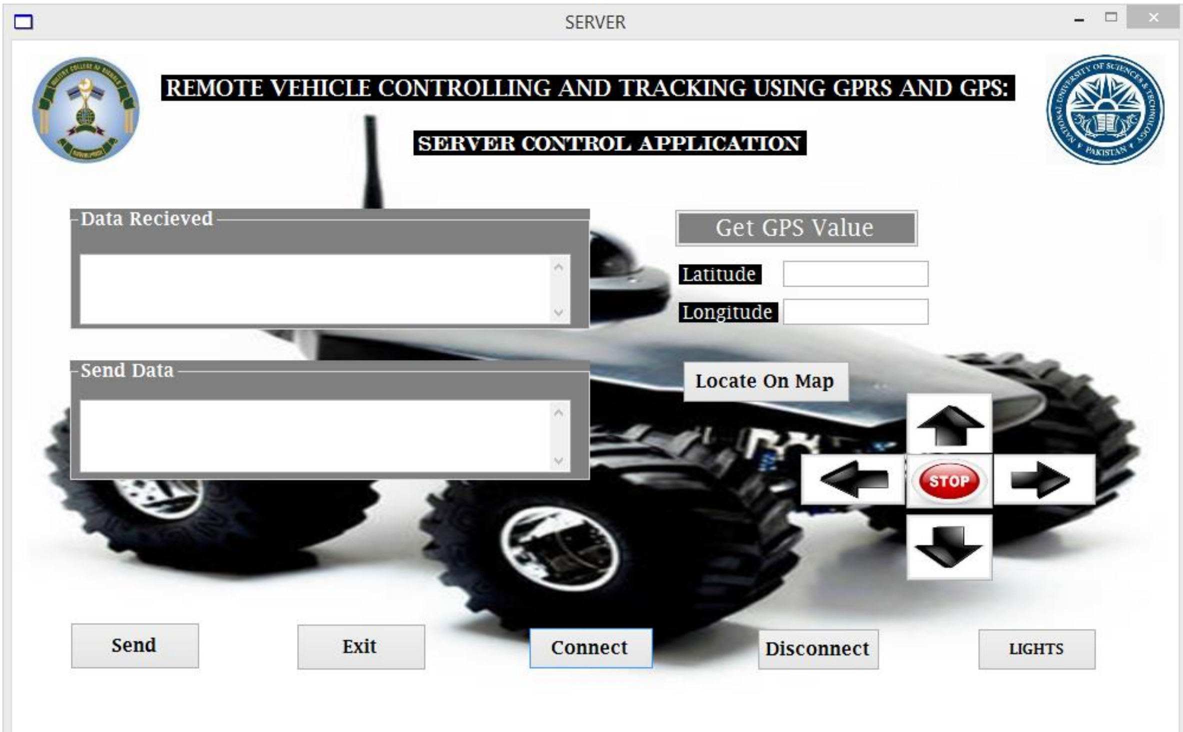


Figure 5.4 Server Control Application

CHAPTER 6: FINAL DESIGN

6.1 LED TEST

After successful run on Proteus first hardware design is for LED test. When we click on any button on application respective port bits turn on and off. Further a motor driver circuit is connected to these port bits.

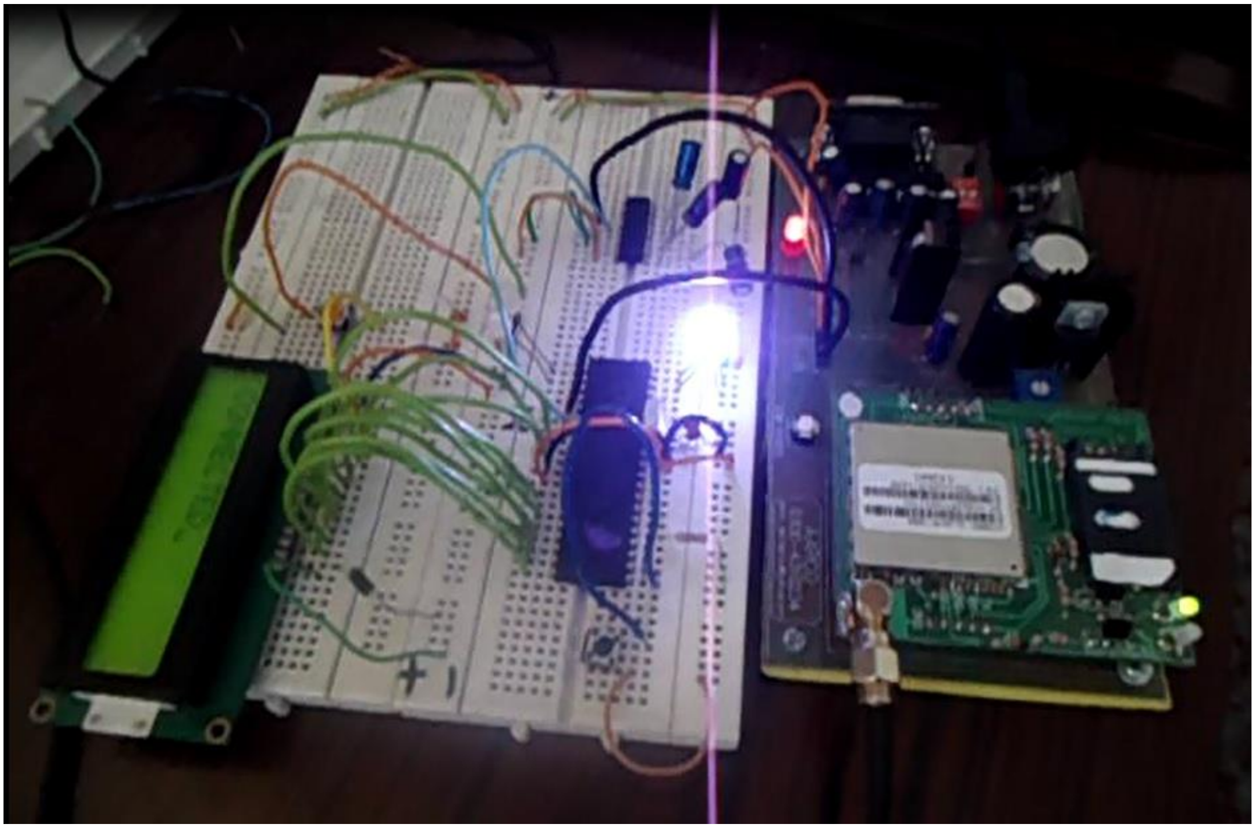


Figure 6.1 LED Test

LED test indicates that the microcontroller is receiving data correctly and making decisions upon reception. PIC18 cannot read the data directly from the TX pin of sim300 therefore there is a need of MAX232 voltage converter. MAX232 coverts the TTL logic to higher voltage level so that microcontroller can read the characters received.

6.2 Motor Driver Circuit

Motor driver circuit consists of:

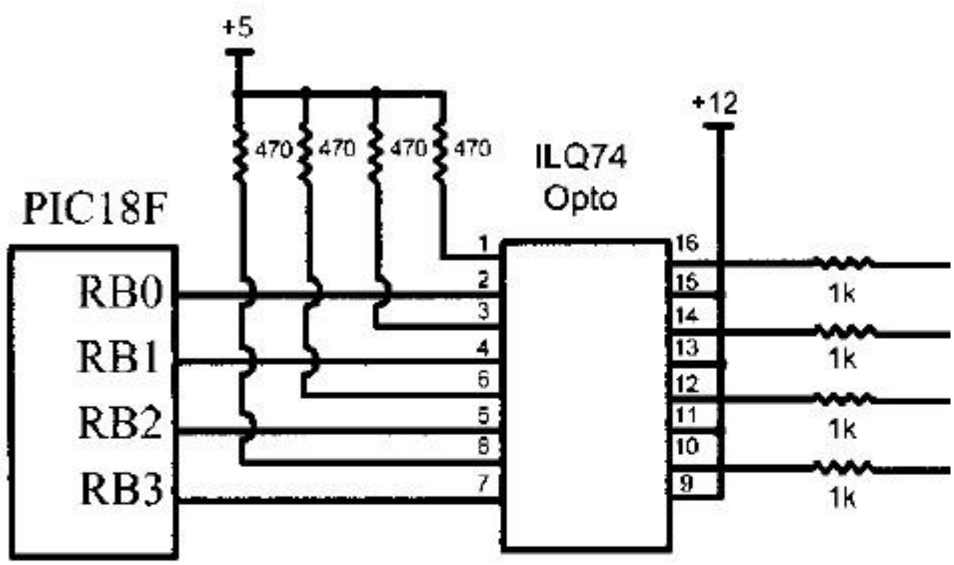
a. Opt coupler

b. L293D (dual H-bridge IC)

c. Voltage regulator

d. Protection Diodes

Opt couplers are used to insulate the microcontroller circuit from the motors because motors produces back EMF which can damage the circuit. Opt couplers also separate the supply voltage for the DC motors and rest of the circuit. By this if large current is drawn by the motors it will not disturb the circuit.



Outputs from the opt coupler are connected to L293D dual H-bridge IC. By using L293D we have a bidirectional control of 2 DC motors.

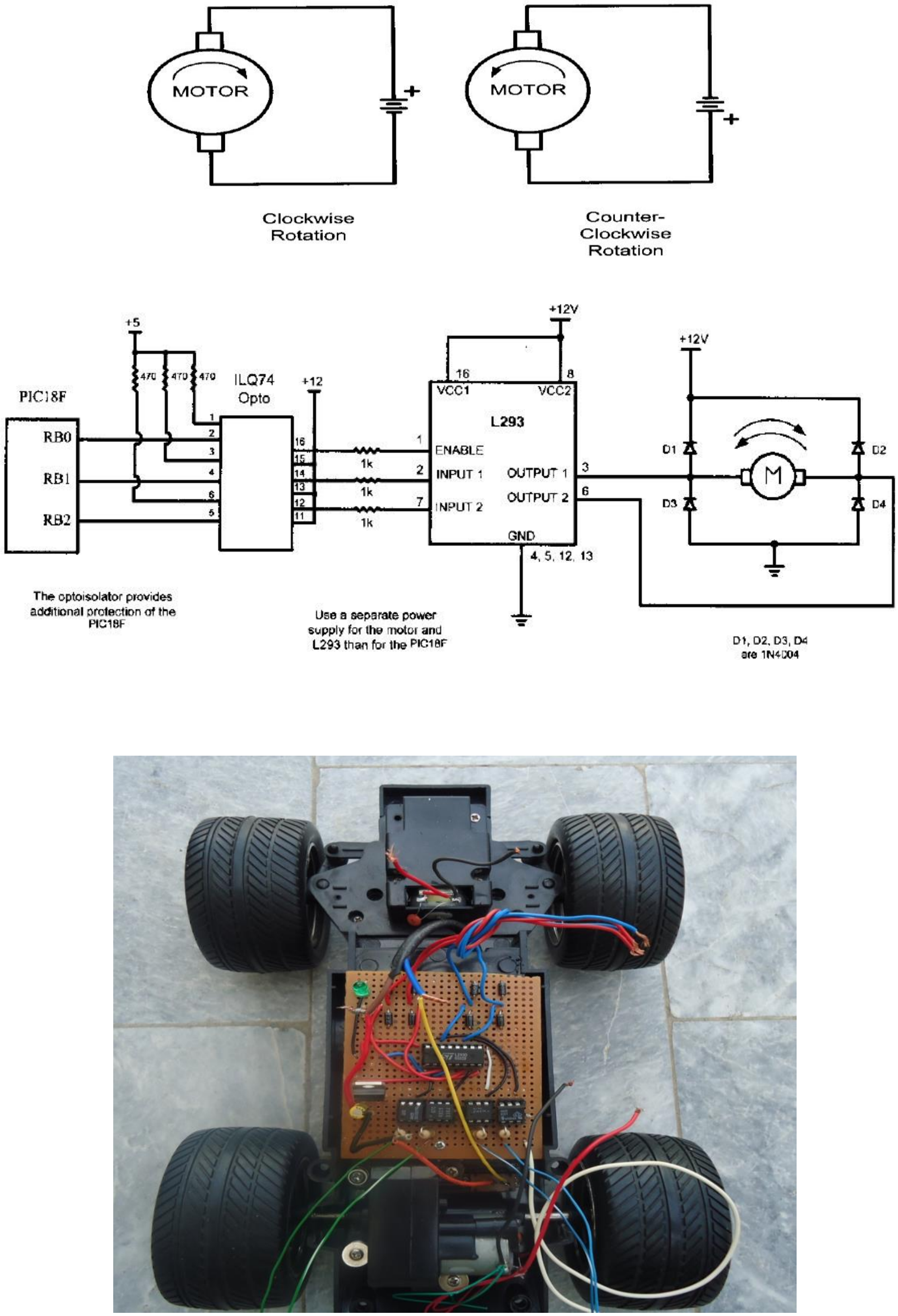


Figure 6.2 Motor Driver circuit

6.3 Final PCB Design

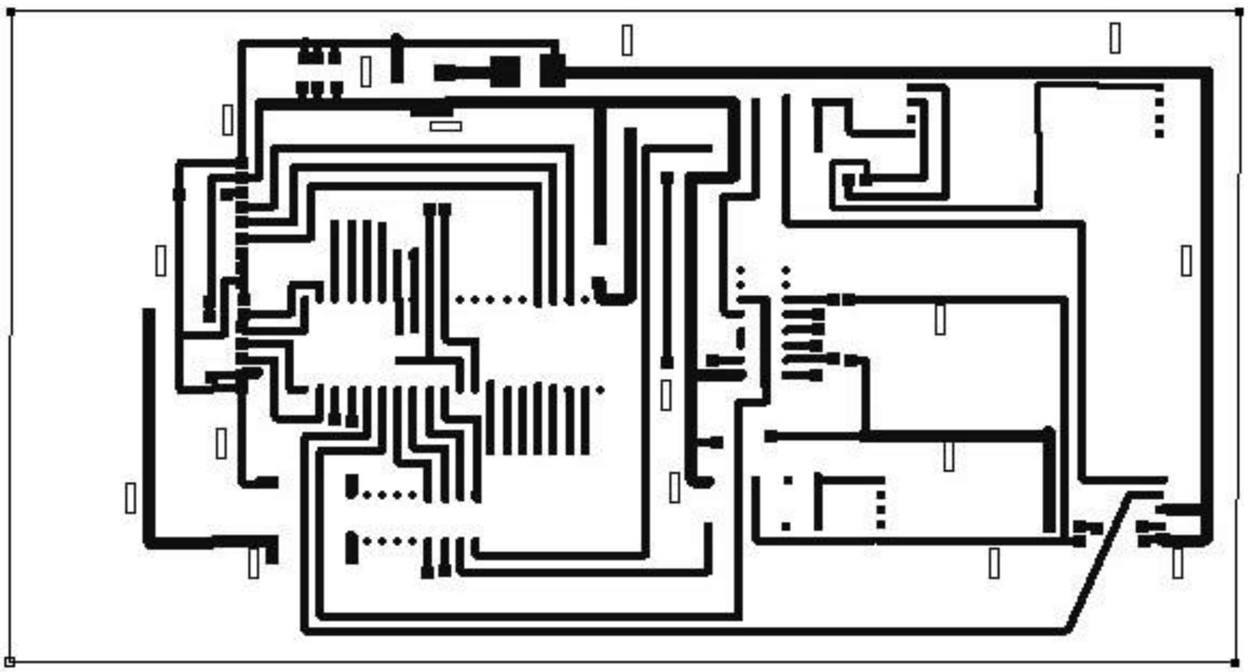


Figure 6.3 PCB print

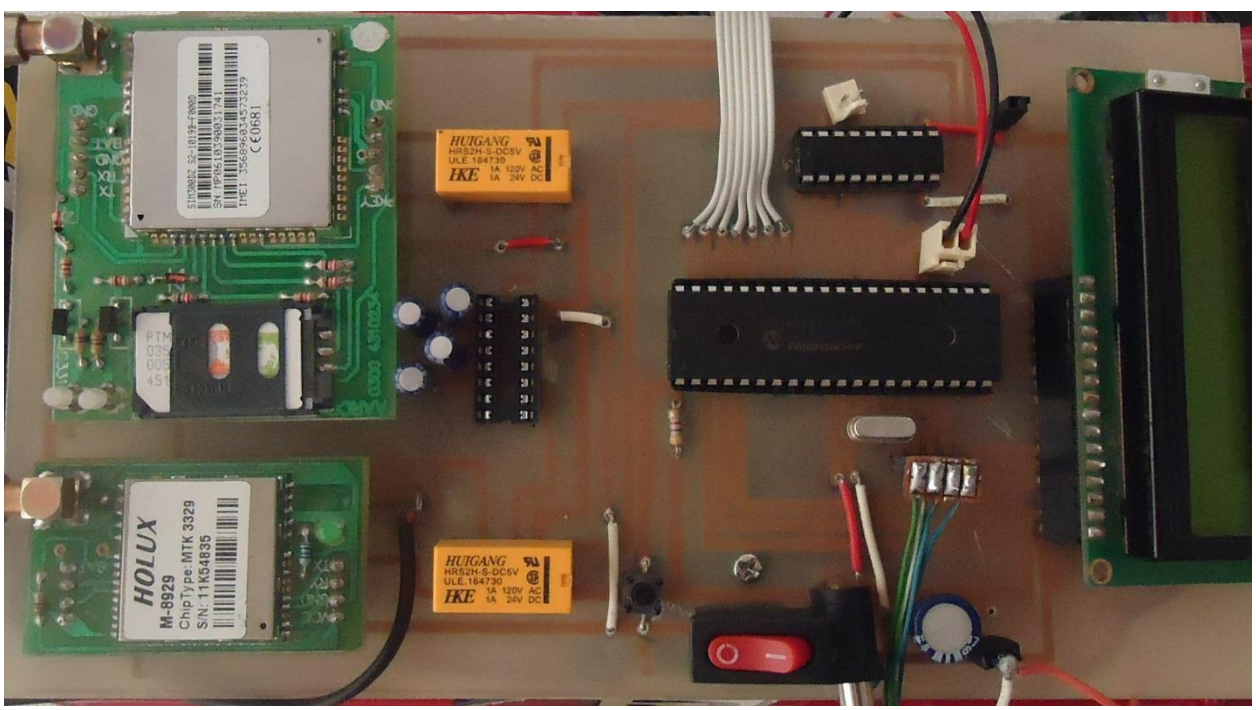


Figure 6.4 PCB design

6.4 Vehicle Design

Figure 6.5 Vehicle Design 1

Figure 6.6 Vehicle Design 2

CONCLUSION & FUTURE WORK

The project "**Remote vehicle controlling and tracking using GPRS and GPS**" has been successfully designed, developed and tested. All the hardware components used have been developed with integrated features. Every module / component has been made well-structured and is benefitting the best working of the unit/ module. The project has been implemented with highly advanced and developed ICs and modern technologies.

Currently, in our project, remote vehicle is being controlled and tracked by GPRS and GPS. In future, we can implement ultrasonic sensors for the hurdles. We can also implement real time video transmission or picture capturing using Digital Signal Processing (DSP). This will be a best security solution for the areas or perimeters where human monitoring is not possible all the time. This project can potentially be used in the design of the next-generation secure systems.

BIBLIOGRAPHY

[1] Rappaport Theodore S., *Wireless Communications: Principles And Practice, 2/E.* Pearson Education India, Sept 1, 2010.

[2] Matt Weatherford, "Transportation District's Automatic Vehicle Location System," *http://ntl.bts.gov/lib/jpodocs/repts_te/13589.html*, Aug 10, 2000.

[3] "What is GPRS," *http://www.s21.com/gprs.htm*, Jan 13, 2012.

[4] Electronics Projects, "GSM GPS based vehicle theft control system," *http://theelectronicsproject.blogspot.com/2013/04/gsm-gps-based-vehicle-theft-control.html*, May 2013.

[5] William C. Y. Lee, *Mobile Cellular Telecommunications*. McGraw-Hill, 1995.

[6] Muhammad Ali Mazidi, Rolin D. McKinlay, Danny Causey, *Pic Microcontroller And Embedded Systems: Using Assembly and C for Pic18*. Pearson Education, Limited, 2008.

[7] Martin P. Bates, *Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation*. Newnes, Sept 13, 2013.

[8] Richard H. Barnett, Larry O'Cull, Sarah Alison Cox, *Embedded C Programming and the Microchip PIC*, vol 1. Cengage Learning, 2004.

[9] Margaret Rouse, "TCP/IP (Transmission Control Protocol/Internet Protocol)," *http://searchnetworking.techtarget.com/definition/TCP-IP*, October 2012.