SMART VEHICLE DATA LOGGER USING GLOBAL POSITIONING SYSTEM AND GLOBAL SYSTEM FOR MOBILE COMMUNICATION

WAN MOHD AMIR HARIS BIN WAN SALLEHUDDIN

UNIVERSITY MALAYSIA PAHANG

SMART VEHICLE DATA LOGGER USING GLOBAL POSITIONING SYSTEM AND GLOBAL SYSTEM FOR MOBILE COMMUNICATION

WAN MOHD AMIR HARIS BIN WAN SALLEHUDDIN

A thesis submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Electrical Engineering (Electronic)

Faculty of Electrical and Electronics Engineering

Universiti Malaysia Pahang

NOVEMBER 2008

I have declared “Smart Vehicle Data Logger using GPS and GSM interface” is the result of my own research except as cited in the references. This thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree

Signature: ______________________________

Name: WAN MOHD AMIR HARIS BIN WAN SALLEHUDDIN

Date: 9 NOVEMBER 2008

Special dedicated to

my beloved parents and my brothers for guiding me in journey of education.

ACKNOWLEDGEMENT

During doing this thesis, I founded myself around with a lot of great people. Those people have helped me a lot in doing this research directly or in directly. Their contribution has helped me in understanding about my project thoroughly.

I would like to say thank to my supervisor, Mr Mohd Zamri bin Ibrahim for his support, guide, advices and determination in guiding me to finish my final year project and this thesis to. I would also like to express my gratitude to my colleagues, for their intention on helping me in any sort of way.

I, myself are fully in debt with Faculty of Electrical and Electronics (FKEE), for providing me necessary components, information and funding my project. Without their helped, this project was deeming to be unfinished.

Last but not least, I would like to say my gratitude toward to my family member for their support and encouragement. I am grateful to have them all.

ABSTRACT

Nowadays, there are lots of technologies growing in this world. Technologies such as Global Positioning System (GPS), Global System for Mobile communications (GSM) and Multimedia card (MMC) can make our lives more comfortable. The Smart Vehicle Data Logger can track speed and location of the vehicle travel using Global Positioning System (GPS) and store the information into Multimedia Card. User is allowed to trace the location of the vehicle by SMS the Data Logger. The system will send back the location of the vehicle through GSM module. The controller for this prototype design is microcontroller system using PIC 18F452 and programmed using PIC C Compiler (C Programming) software. So, with these technologies, we can create an excellent way for fleet owners and managers to monitor their cars, trucks, or vehicles efficiently.

ABSTRAK

Pada masa kini, pelbagai teknologi telah berkembang di dunia ini. Teknologi seperti "Global Positioning System (GPS), Global System for Mobile communications (GSM) dan Multimedia card (MMC)" membuatkan hidup kita lebih selesa. "Smart Vehicle Data Logger" ini boleh mengesan kelajuan dan lokasi kenderaan dan menyimpan data tersebut ke dalam "Multimedia kad". Pengguna juga boleh mengesan lokasi kenderaan dengan menerusi perkhidmatan "SMS". Sistem alat tersebut akan membalas kembali lokasi alat tersebut menerusi "GSM" sistem. Sistem pengawal untuk prototaip ini adalah mikro pengawal PIC18F452 dan ia diprogramkan menggunakan C program. Oleh itu, dengan adanya teknologi-teknologi ini, pengguna kenderaaan lebih mudah mengawal system kenderaan mereka.

TABLE OF CONTENTS

2 LITERATURE REVIEW

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

MCU	=	Microcontroller Unit
GPS	=	Global Positioning System
GSM	=	Global System for Mobile communications
MMC	=	Multimedia Card
SMS	=	Short Messaging System
I/0	=	Input / Output
V	=	Volts
LCD	=	Liquid Crystal Display
RAM	=	Random Access Memory
ROM	=	Read Only Memory
MHz	=	Megahertz
CS	=	Chip Select
s	=	second
SMS	=	Short Message System

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Nowadays, there are lots of technologies growing in this world. Technologies such as Global Positioning System (GPS), Global System for Mobile communications (GSM) and Multimedia card (MMC) make our lives more comfortable.

Global Positioning System (GPS): A typical GPS receiver calculates its position using the signals from four or more GPS satellites. Four satellites are needed since the process needs a very accurate local time, more accurate than any normal clock can provide, so the receiver internally solves for time as well as position. In other words, the receiver uses four measurements to solve for 4 variables - *x*, *y*, *z*, and *t*. These values are then turned into more user-friendly forms, such as latitude/longitude or location on a map, then displayed to the user.

Global System for Mobile communications (**GSM**) is the most popular standard for mobile phones in the world. GSM is used by over 2 billion people across more than 212 countries and territories. Its ubiquity makes international roaming very common between mobile phone operators, enabling subscribers to use their phones in many parts of the world.

The **MultiMediaCard** (**MMC**) is a flash memory card standard. MMC is used as storage media for a portable device, in a form that can easily be removed for access by a PC. For example, a digital camera would use an MMC for storing image files. With an MMC reader (typically a small box that connects via USB or some other serial connection, although some can be found integrated into the computer itself), a user could copy the pictures taken with the digital camera off to his or her computer. MMCs are currently available in sizes up to and including 4 GB with 8 GB models announced but not yet available.

As we all know, there are lots of car that produced by local company, but, customer cannot detect the vehicle's location. Customers also cannot log the history of the vehicle travel. So, a product had been design which uses all these technologies in order to make costumer more comfortable with the car.

1.2 Problem statement

As we all know, there are lots of vehicles that produced by local company. But, there are something not available yet:

1. Owners cannot trace their vehicle's location.
2. Owners also cannot log the history of the vehicle travel.

In order to these problem, a vehicle data logger will be design which can log the history of the vehicle travel and trace the vehicle's location.

1.3 Project Objective

The main objective of this project is to design a data logger that has ability to log data of a vehicle. There are three others objectives to be achieved beside the main objective stated above. The three objectives are discussed in the following paragraph.

First of all, design a controller that can read the data from GPS receiver and analysis the data till get the certain protocol only. The protocol that had been analyzed will show the time , longitude, latitude, date and speed of the receiver. To show all the data is valid or not, the four characteristics will be display to user.

The second objective is to store all the data that had been analyzed into a suitable storage. The storage must be large capacity because the data from GPS always receive at every second.

Last but not lease, to make the system is more friendly user , there will be a controller that can make two way communication between user and the system. With this system, user can track the vehicle either send a SMS or make a phone call to the system.

1.4 Project Scope

Few scopes and guidelines are listed to ensure the project is conducted within its intended boundary. This is to ensure the project is heading to the right direction to achieve its intended objectives.

The first scope is to design a controller that can control all the works in the system. The controller can read the GPS data, store the data into a storage card, display it at LCD and do transmit / receive to GSM module. In the other hand, the controller is the brain of the system.

Second scope of this project is to analysis the data from GPS receiver which can get the time , longitude, latitude, date and speed of the receiver. The data will be display at LCD Display.

The third scope is to store the data from GPS receiver into a Multimedia Card. All the time , longitude, latitude, date and speed of the GPS receiver will be store into MMC card through the microcontroller.

The last scope of this project is to trace the location of the vehicle using Global System for Mobile communications (GSM) *(two way communication between user and GSM Module)* and Global Positioning System (GPS).

1.5 Thesis Overview

This thesis consist of seven chapters. The first chapter will give an overview of the project as well as the objective of the project.

Chapter 2 covers the literature review or discuss about the research of the data logger current project and the component that will be used for this project. From this chapter, we can know all the protocol and characteristic of GPS Receiver, GPS Modem and MMC Configuration system.

Chapter 3 covers the electronic overview of the project. It describes the various modules developed, basic operation of each module. All five module are describe briefly in this chapter.

The elaboration of the software development will be discuss in Chapter 4. All the software used for this project will be describe briefly in this chapter.

The system implementation of the project will be discuss on Chapter 5. In this chapter, the software and hardware combination will be explained briefly. The hardware will be elaborates from the circuits and the programming of each modules will be discussed by using flow charts.

Chapter 6 explains the testing and result of each module. The result system effectiveness is also discussed.

Chapter 7 summarized the project outcome. A few suggestions are proposed to enhance the current design.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, we will discuss about the research of the data logger current project and the component that will be used for this project. From this literature review, we can get the idea of the function and description of the modules such as GPS module, GSM module and Multimedia card system.

This chapter is divided by two sections. One section is for current project and the another section is about the research of module use in this project. The sub section are listed as below :

i. Current Project
   - Shadow Tracker® Expert
   - CarChip Pro Automotive Data Logger

ii. Module research.
   - PIC18F452 Microchip Controller
   - Global Positioning System (GPS)
   - SanDisk MultiMediaCard and Reduced-Size MultiMediaCard
   - LCD module
   - GSM Module

2.2 Review of current project

In this project, a research had been done about the system that already exist that related in my project. There are two kind of product that had been create in order to log the data of the vehicle. There are :

i. Shadow Tracker® Expert
ii. CarChip Pro Automotive Data Logger

2.2.1 Shadow Tracker® Expert

The Shadow Tracker® Expert serves as protection for your company against employee time sheet fraud and unauthorized use of your vehicles. Using the latest Shadow Tracker® mapping software, the Shadow Tracker® Expert displays actual routes driven, number of stops, and amount of time spent at each site. [1]

Figure 2.1 : Shadow Tracker® Expert

Features and Benefits :

i. USB connector with 921kbs download rate
ii. The 1MB of memory can store approximately 1851.9 hours of drive time*
iii. GPS antenna sensing and Ignition on/off sensing for detection of tampering
iv. Battery backup power off sensing
v. Down to 2 mph speed recording for slow moving tracking
vi. Optional rechargeable internal Lithium-Ion battery provides approximately 25 hours of tracking without main power
vii. Actively manage field personnel

viii. Know where your mobile workforce is every day
ix. Prevent incidents of employee time sheet fraud
x. Cut mobile fleet costs
xi. Head off customer service complaints

* Based on a 120 second collection rate of actual drive time

According to this product, it do not have two way communication using GSM. So, owner cannot trace the car on time.

2.2.2 CarChip Pro Automotive Data Logger

The Car Chip Pro Logger is capable of recording and logging driving, trip and engine performance data. By attaching the CarChip to your vehicle's On-board diagnostics (OBDII) port you get a detailed look at how the vehicle is driven, trip details, emissions status and engine diagnostic codes.[2]

Figure 2.2 : CarChip Pro Automotive Data Logger

CarChip Pro Logger Features :

i. Record up to 300 Hours of Trip Details
ii. Trip Details includes: Date/Time, Distance Traveled and Speed
iii. Provides Individual Graphs and Summary Reports
iv. Records Extreme Acceleration and Braking
v. Calculate Gas Mileage
vi. Includes Software and USB Cable

CarChip Pro Logger Specifications :

Table 2.1 : Carchip pro logger specification

Operating Temperature Range	-40°F to 185°F (-40°C to 85°C)
Primary Power (Connected to Vehicle)	9 to 16 VDC, 80 mA with Vehicle Running, 17mA with Vehicle's Power Off
Primary Power (Connected to Computer)	USB Powered
Backup Power	Internal Battery, Minimum of 5 Years Total, with Data Logger not Powered by Vehicle or Computer; 10-15 Year Life in Normal Use
Memory	512KB
Data Storage Capacity	300 Hours Maximum (Dependant on Interval and Environment)
Vehicle Interface	16-pin OBDII Connector
Alarm	Adjustable, Audible Alarm for Exceeding Speed, Acceleration, and Deceleration Limits
Supported OBDII Protocols	J1859-41.6, J1850-10.4, ISO9141, KWP2000 (ISO 14230), CAN (Control Area Network ISO 11898)
Vehicle Speed Sampling Interval	1, 5, 10, 30 or 60 Seconds
Dimensions	4.3" x 3.9 " x 0.78 " (109mm x 99mm x 20mm)
Weight	6oz. (169g)

According to this product, it cannot log the location of vehicle. It just can log the speed and the performance of engine of the vehicle. Otherwise, the data storage capacity is too small. Only 300 hours storage.

2.3 Research of project module

List of research :

- PIC18F452 Microchip Controller
- Global Positioning System (GPS)
- SanDisk MultiMediaCard and Reduced-Size MultiMediaCard

- LCD module
- GSM Module

2.3.1 PIC18F452 Microchip Controller

This powerful 10 MIPS (100 nanosecond instruction execution) yet easy-to-program (only 77 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40- or 44-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX, PIC16CXX and PIC17CXX devices and thus providing a seamless migration path of software code to higher levels of hardware integration. The PIC18F452 features a 'C' compiler friendly development environment, 256 bytes of EEPROM, Self-programming, an ICD, 2 capture/compare/PWM functions, 8 channels of 10-bit Analog-to-Digital (A/D) converter, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus and Addressable Universal Asynchronous Receiver Transmitter (AUSART). All of these features make it ideal for manufacturing equipment, instrumentation and monitoring, data acquisition, power conditioning, environmental monitoring, telecom and consumer audio/video applications.[3]

Parameter Name	Value
Program Memory Type	Flash
Program Memory Size (Kbytes)	32
RAM	1,536
Data EEPROM (bytes)	256
I/O	34

Figure 2.3 : PIC18F452 Feature

2.3.2 Global Positioning System (GPS)

The GPS Receiver Module provides standard, raw NMEA0183 (National Marine Electronics Association) strings or specific user-requested data via the serial command interface, tracking of up to 12 satellites, and WAAS/EGNOS (Wide Area Augmentation System/European Geostationary Navigation Overlay Service) functionality for more accurate positioning results.

The Module provides current time, date, latitude, longitude, altitude, speed, and travel direction/heading, among other data, and can be used in a wide variety of hobbyist and commercial applications, including navigation, tracking systems, mapping, fleet management, auto-pilot, and robotics. [4]

2.3.2.1 GPS Technology Brief

GPS (Global Positioning System) is a worldwide radio-navigation system formed by a constellation of 24 satellites and their ground stations. With an unobstructed, clear view of the sky, GPS works anywhere in the world, 24 hours a day, seven days a week.

The Global Positioning System consists of three interacting components:
1) The Space Segment -- satellites orbiting the earth.
2) The Control Segment -- the control and monitoring stations run by the United States Department of Defense (not discussed in this documentation).
3) The User Segment -- the GPS signal receivers owned by civilians and military.

The space segment consists of a constellation of 24 active satellites (and one or more in-orbit spares) orbiting the earth every 12 hours. Four satellites are located in each of six orbits and will be visible from any location on each 95 percent of the time. The orbits are distributed evenly around the earth, and are inclined 55 degrees from the equator. The satellites orbit at an altitude of about 11,000 nautical miles.

Each satellite transmits two signals: L1 (1575.42 MHz) and L2 (1227.60 MHz). The L1 signal is modulated with two pseudo-random noise signals - the protected (P) code, and the course/acquisition (C/A) code. The L2 signal only carries the P code. Civilian navigation receivers only use the C/A code on the L1 frequency. Each signal from each satellite contains a repeating message, indicating the position and orbital parameters of itself and the other satellites (almanac), a bill of health for the satellites (health bit), and the precise atomic time.

The receiver measures the time required for the signal to travel from the satellite to the receiver, by knowing the time that the signal left the satellite, and observing the time it receives the signal, based on its internal clock. If the receiver had a perfect clock, exactly in sync with those on the satellites, three measurements, from three satellites, would be sufficient to determine position in three dimensions via triangulation. However, that is not the case, so a fourth satellite is needed to resolve the receiver clock error. With four satellites, a GPS receiver can provide very accurate clock (time, date) and position information (latitude, longitude, altitude, speed, travel direction/heading).

Note that position data and accuracy are affected or degraded by the satellite geometry, electromagnetic interference, and multipath, an unpredictable set of reflections and/or direct waves each with its own degree of attenuation and delay. Primarily due to satellite geometry, measuring altitude using GPS may introduce an accuracy error of 1.5 times the receiver's position accuracy (in the case of our GPS Receiver Module, this corresponds to about +/-20 meters in the vertical direction).

GPS signals work in the microwave radio band. They can pass through glass, but are absorbed by water molecules (wood, heavy foliage) and reflect off concrete, steel, and rock. This means that GPS units have trouble operating in rain forests, urban jungles, deep canyons, inside automobiles and boats, and in heavy snowfall - among other things. These environmental obstacles degrade positional accuracy or make it impossible to get a fix on your location.

Most GPS receivers output a stream of data so that it can be used and interpreted by other devices. The most common format (and used by our GPS Receiver Module in "Raw Mode") is NMEA0183 (National Marine Electronics

Association, http://www.nmea.org/), developed for data communications between marine instruments. Some receivers also have proprietary data formats which are used (in the case of navigation receivers) to transfer waypoint lists, track logs, and other data between the GPS and a computer. Such proprietary formats are not covered by the NMEA standard.

The NMEA0183 is provided as a series of comma-delimited ASCII strings, each preceded with an identifying header. The data is transmitted as a 4800bps string of 8-bit ASCII characters. Thus, any microcontroller with a serial port can extract data from a GPS module. But, modules do not produce "plain text" location information. Instead, they create standardized "sentences," such as:

$GPGGA,170834,4124.8963,N,08151.6838,W,1,05,1.5,280.2,M,-34.0,M,,,*75
$GPGSA,A,3,19,28,14,18,27,22,31,39,,,,,1.7,1.0,1.3*34
$GPGSV,3,2,11,14,25,170,00,16,57,208,39,18,67,296,40,19,40,246,00*74
$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70

Programmers can parse these strings to obtain their desired information, including time, date, latitude, longitude, speed, and altitude.

There are three standard notations for displaying longitude and latitude data:

- **GPS Coordinates** (degrees, minutes, and fractional minutes), ex: 36 degrees, 35.9159 minutes
- **DDMMSS** (degrees, minutes, seconds), ex: 36 degrees, 35 minutes, 55.3 seconds
- **Decimal Degrees**, ex: 36.5986 degrees

In “Smart Mode,” the Parallax GPS Receiver Module transmits latitude and longitude data to the user in GPS Coordinate format (degrees, minutes, and fractional minutes, see the “Communication Protocol” section for more details). Conversion to the two other notations, DDMMSS (degrees, minutes, second) and Decimal Degrees, is trivial and demonstrated in the example code below.

To graphically display your GPS position using Google Maps (in map, satellite, or hybrid view), simply go to http://maps.google.com/ and enter in your decimal coordinates in the "Search" field (for example, "36.5986, -118.0599" without the quotes).

To graphically display a track or series of waypoints, GPS Visualizer (http://www.gpsvisualizer.com/) is a free, easy-to-use online utility that creates maps and profiles from GPS data. GPS Visualizer can read data files from many different sources, including raw NMEA strings (such as those captured directly from the Parallax GPS Receiver Module in “Raw Mode”) or tab-delimited or comma-separated text of relevant GPS data.

2.3.2.2 GPS Electronic Connections

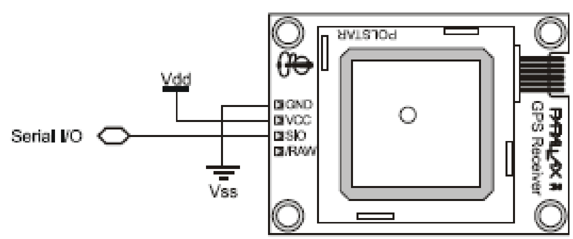


Figure 2.4 : GPS Module

Table 2.2 : GPS Electronic Description

Pin	Pin Name	Type	Function
1	GND	G	System ground. Connect to power supply’s ground (GND) terminal.
2	VCC	P	System power, +5V DC input.
3	SIO	I/O	Serial communication (commands sent TO the Module and data received FROM the Module). Asynchronous, TTL-level interface, 4800bps, 8 data bits, no parity, 1 stop bit, non-inverted.

4	/RAW	I	Mode select pin. Active LOW digital input. Internally pulled HIGH by default. When the /RAW pin is unconnected, the default "Smart Mode" is enabled, wherein commands for specific GPS data can be requested and the results will be returned (see the "Command Structure" section for more details). When /RAW is pulled LOW, the Module will enter "Raw Mode" and will transmit standard strings, allowing advanced users to use the raw GPS data directly.

2.3.2.3 GPS Status Indicators

The GPS Receiver Module contains a single red LED (light-emitting diode) to denote system status. The LED is located in the lower-right corner of the Module. A white overlay on the Module's printed circuit board is used to reflect the light from the LED, making it easier for the user to see. The LED denotes two states of the Module:

1) **Blinking** (both fast and slow): Searching for satellites or no satellite fix acquired

2) **Solid**: Satellites successfully acquired (a minimum of three satellites is required before the Module will begin to transmit valid GPS data)

Upon power up of the GPS Receiver Module in a new location, the Module may take up to five minutes or more to acquire a fix on the necessary minimum number of four satellites. During this time, the red LED on the Module will blink. When enough satellites are acquired for the Module to function properly, the red LED will remain solid red. Due to a variety of conditions, the number of satellites may vary at any given time. If the LED is OFF, there may be a problem. Please check your wiring and configuration of the Module.

2.3.2.4 GPS Mode Selection

The /RAW pin allows user selection of the GPS Receiver Module's two operating modes:

- **Smart Mode:** When the /RAW pin is pulled HIGH or simply left unconnected (the pin is internally pulled HIGH), the default "Smart Mode" is enabled, wherein commands for specific GPS data can be requested and the results will be returned. See the "Communication Protocol" section for more details.

- **Raw Mode:** When the /RAW pin is pulled LOW, "Raw Mode" is enabled in which the Module will transmit standard NMEA0183 v2.2 strings (GGA, GSV, GSA, and RMC), allowing advanced users to use the raw GPS data directly. For more information on NMEA0183 data, see the "GPS Technology Brief" section.

In either mode, data is transmitted at 4800bps, 8 data bits, no parity, 1 stop bit, non-inverted, TTL-level.

2.3.2.5 GPS Electrical Characteristics

Absolute Maximum Ratings

Table 2.3 : GPS Electrical Characteristic

Condition	**Value**
Operating Temperature	-40ºC to +85ºC
Storage Temperature	-55ºC to +100ºC
Supply Voltage (V_{CC})	+4.5V to +5.5V
Ground Voltage (V_{SS})	0V
Voltage on any pin with respect to V_{SS}	-0.6V to +(Vcc+0.6)V

2.3.3 SanDisk MultiMediaCard and Reduced-Size MultiMediaCard

2.3.3.1 Introduction

The SanDisk MultiMediaCard and Reduced-Size MultiMediaCard (RS-MMC) are very small, removable flash storage devices, designed specifically for storage applications that put a premium on small form factor, low power and low cost. Flash is the ideal storage medium for portable, battery-powered devices. It features low power consumption and is non-volatile, requiring no power to maintain the stored data. It also has a wide operating range for temperature, shock and vibration.

The MultiMediaCard and RS-MultiMediaCard are well suited to meet the needs of small, low power, electronic devices. With form factors of 32 mm x 24 mm and 1.4 mm thick for the MultiMediaCard and 18 mm x 24 mm x1.4 mm for the RS-MultiMediaCard, these cards can be used in a wide variety of portable devices like mobile phones, and voicerecorders.

To support this wide range of applications, the MultiMediaCard Protocol, a simple sevenpinserial interface, is designed for maximum scalability and configurability. All device andinterface configuration data (such as maximum frequency, card identification, etc.) are stored on the card.

The SanDisk MultiMediaCard/RS-MultiMediaCard interface allows for easy integration into any design, regardless of microprocessor used. For compatibility with existing controllers, the card offers, in addition to the card interface, an alternate communication protocol, which is based on the Serial Peripheral Interface (SPI) standard.

The MultiMediaCard/RS-MultiMediaCard provides up to 256 million bytes of memory using SanDisk Flash memory chips, which were designed by SanDisk especially for use in mass storage applications. In addition to the mass storage specific flash memory chip, the MultiMediaCard/RS-MultiMediaCard includes an on-card intelligent controller which manages interface protocols and data storage and retrieval, as well as Error Correction Code (ECC) algorithms, defect handling and diagnostics, power management and clock control.

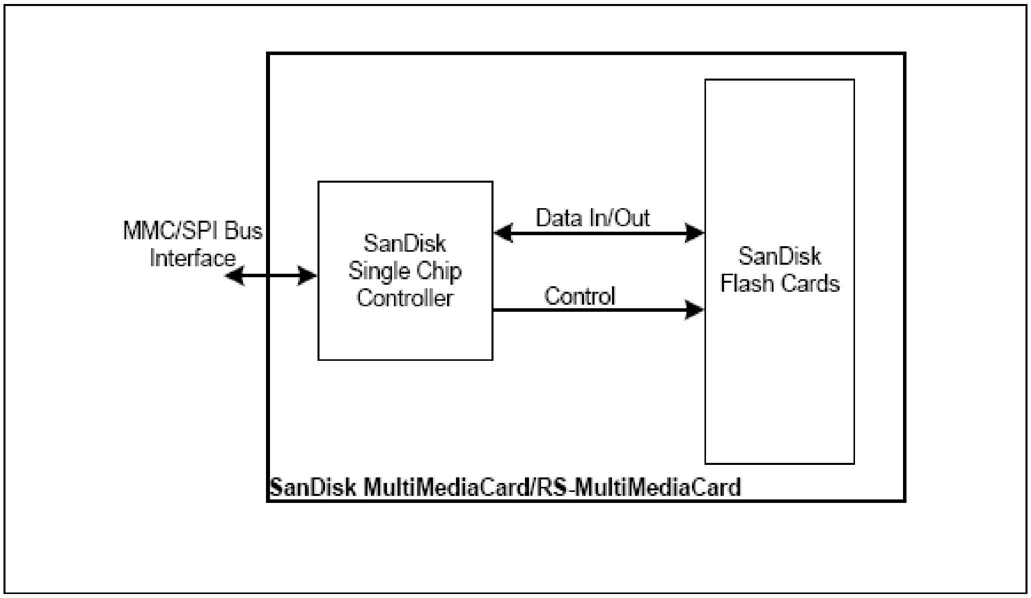


Figure 2.5 : Multimedia Card Block Diagram

2.3.3.2 MMC Features

The SanDisk MultiMediaCard/RS-MultiMediaCard features include:

i. MultiMediaCard Protocol compatible
ii. SPI Mode supported
iii. Targeted for portable and stationary applications
iv. Voltage range
    a. Basic communication: 2.7 to 3.6 V
    b. Memory access: 2.7 to 3.6 V
v. Maximum data rate with up to 10 cards
vi. Correction of memory field errors
vii. Variable clock rate 0 - 20 Mhz
viii. Multiple cards stackable on a single physical bus

2.3.3.3 MMC Functional Description

The MultiMediaCard and RS-MultiMediaCard contain a high level, intelligent subsystem as shown by the block diagram in Figure 1-1. This intelligent (microprocessor) subsystem provides many capabilities not found in other types of memory cards. These capabilities include:

• Host independence from details of erasing and programming flash memory
• Sophisticated system for managing defects (analogous to systems found in magnetic
disk drives)
• Sophisticated system for error recovery including a powerful error correction code
• Power management for low power operation

2.3.3.4 Flash-Independent Technology

The 512-byte sector size of the MultiMediaCard and RS-MultiMediaCard is the same as that in an IDE magnetic disk drive. To write or read a sector (or multiple sectors), the host computer software simply issues a read or write command to the card. This command contains the address. The host software then waits for the command to complete. The host software does not get involved in the details of how the flash memory is erased, programmed or read. This is extremely important as flash devices are expected to get more and more complex in the future. Because the MultiMediaCard and RS-MultiMediaCard uses an intelligent on-board controller, the host system software will not require changing as new flash memory evolves. In other words, systems that support the SanDisk MultiMediaCard/RS-MultiMediaCard today will be able to access future cards built with new flash technology without having to update or change host software.

2.3.4 LCD module

Liquid crystal display (LCD) is another common output device. There are various type and model of LCD available in market. The type of JHD 162A LCD will be used in this project. The role of LCD here will display the user information such as username, time and date.

Some features of LCD are:

- 16 character x 2 row
- 7 x 5 dots
- Reflective with EL and LED backlight
- LED/4.2 VDC

LCD is divided into two register bits which are Control Register and Data Register. The Control Register is used to control the operation of LCD while the data register is used to display the character. The figure 2.6 show the LCD display device.

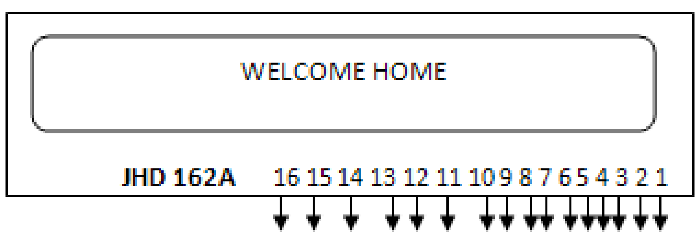


Figure 2.6 : LCD Display

2.3.5 GSM Module

2.3.5.1 Introduction

This part describes the hardware of TMAS GSM/GPRS Terminal. The information is intended for users or developers who design and build wireless cellular, M2M or other data telemetry applications. The scope of this part includes interface specifications, mechanical characteristics of TMAS GSM/GPRS Terminal and power supply issues. TMAS GSM/GPRS Terminal is a compact GSM modem terminal for the transfer of data, SMS and faxes in the GSM networks. Cinterion/Siemens MC52i/MC55i/MC39i/TC35i GSM engine is embedded to provide good quality and reliability of data transfer. Terminal design is also based the industrial standard interfaces and an integrated SIM card reader to allow ease of use to the users.[6]

Module benefit :

i. Low cost
ii. CE Certified, Reliable and Quality with approved GSM/GPRS modules from Cinterion (a.k.a. Siemens)
iii. Non Proprietary Accessories, i.e. standard power adapter and standard modem cable
iv. (You can have the option to purchase these standard accessories in your country or from TCAM or from any cheaper source)
v. Compact and nice stylish silver casing
vi. Flexible mounting: Rail fixing or side mounting plates.

2.3.5.2 GSM Interface Description

TMAS GSM/GPRS Terminal provides the following connectors for power supply, interfacing and antenna:

• 2.1mm DC power connector (center/inner pin is positive)
• 9-pin (female) D-SUB plug for RS-232 serial interface
• SMA connector for antenna (radio interface)
• SIM card holder

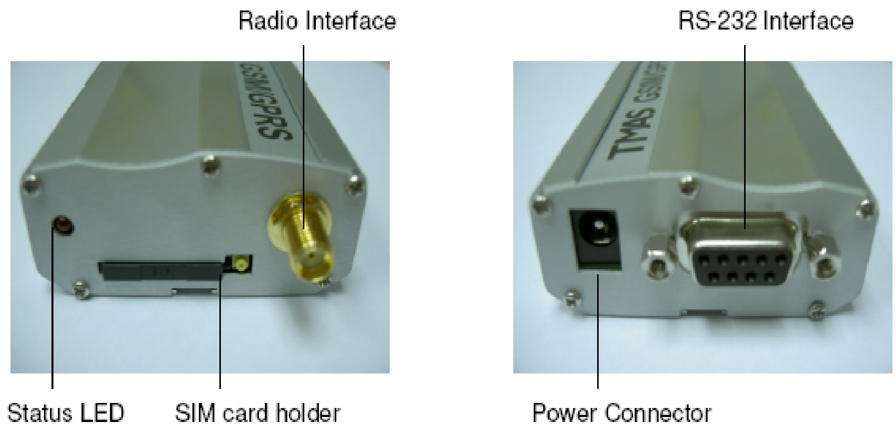


Figure 2.7: GSM Interface

2.3.5.3 GSM Operating Modes

The table below briefly summarizes the various operating modes of TMAS GSM/GPRS Terminal.

Table 2.4: GSM Operating Mode

Mode	Function
SLEEP	Various power saving modes set by AT+CFUN command. Software is active to minimum extent. If the Terminal was registered to the GSM network in IDLE mode, it remains, in SLEEP mode, registered and pageable from the BTS. Power saving can be chosen at different levels. The NON-CYCLIC SLEEP mode (AT+CFUN=0) disables the AT interface. The CYCLIC SLEEP mode AT+CFUN=5, 6, 7 and 8 alternatively activate and deactivate the AT interface to allow permanent access to all AT commands.
GSM IDLE	Software is active. Once registered to the GSM network, paging with BTS is carried out. The Terminal is ready to send and receive.
GSM TALK	Connection between two subscribers is in progress. Power consumption depends on network coverage individual settings, such as DTX off/on, FR/EFR/HR, hopping sequences, antenna.
GPRS IDLE	Module is ready for GPRS data transfer, but no data is currently sent or received. Power consumption depends on network settings and GPRS configuration (e.g. DRX settings).
GPRS DATA	GPRS data transfer in progress. Power consumption depends on network settings (e.g. power control level), uplink / downlink data rates and GPRS configuration (e.g. used multi-slot settings).

2.3.5.4 GSM RS-232 Interface

Via RS-232 interface, the host controller controls the TMAS GSM/GPRS Terminal and transports data. The figure below shows the pin assignment of RS-232 (D-SUB 9-pin female).

Figure 2.8 : GSM RS232 interface

The table below illustrates pin assignment of 9-pole D-SUB (female) RS-232.

Table 2.5 : GSM Rs232 Pin Assignment

Pin no.	Signal name	I/O	Function
1	/DCD	O	Data Carrier Detected
2	/RXD	O	Receive Data
3	/TXD	I	Transmit Data
4	/DTR	I	Data Terminal Ready
5	GND	-	Ground
6	/DSR	O	Data Set Ready
7	/RTS	I	Request To Send
8	/CTS	O	Clear To Send
9	/RI	O	Ring Indication

• Pin TxD @ application sends data to TxD of TMAS GSM/GPRS Terminal

• Pin RxD @ application receives data from RxD of TMAS GSM/GPRS Terminal

The RS-232 interface is implemented as a serial asynchronous transmitter and receiver conforming to ITU-T V.24 Interchange Circuits DCE. It is configured for 8 data bits, no parity and 1 stop bit, and can be operated at bit rates from 300bps to 115Kbps. Autobauding supports bit rates from 4.8Kbps to 115Kbps. Hardware handshake using the /RTS and /CTS signals and XON/XOFF software flow control are supported. In addition, the modem control signals /DTR, /DSR, /DCD and /RING are available. The modem control signal RING (Ring Indication) can be used to indicate to the cellular device application, that a call or Unsolicited Result Code (URC) is received. There are different modes of operation, which can be set with AT commands.

2.3.5.5 SIM Interface

The SIM interface is intended for 3V SIM cards in accordance with GSM 11.12 Phase 2. The card holder is a 5- wire interface according to GSM 11.11. A sixth pin has been added to detect whether or not a SIM card is inserted.

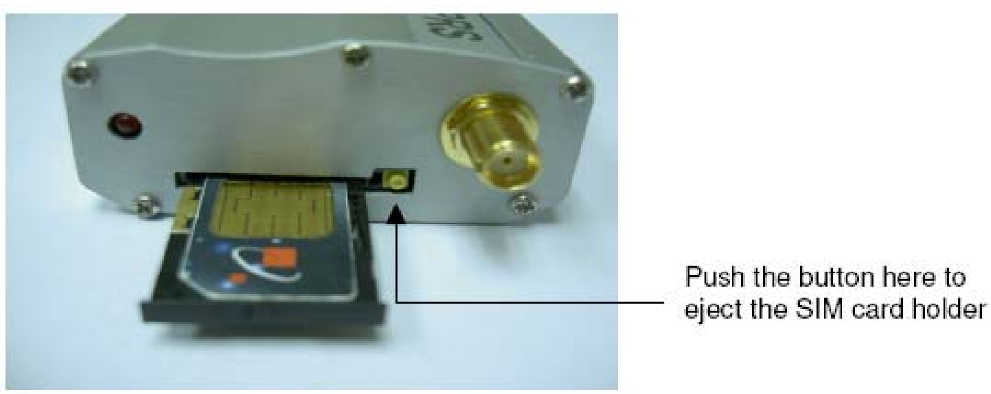


Figure 2.9 : SIM Interface

2.3.5.6 Mechanical Characteristics and Mounting Advice

The table below illustrates the mechanical characteristics:

Table 2.5: GSM Mechanical Charateristic

Weight	82g
Dimension	86mmx54mmx25mm
Temperature range	-20℃ to +55℃
Air humidity	Maximum 80% relative humidity
Casing material	Aluminum alloy

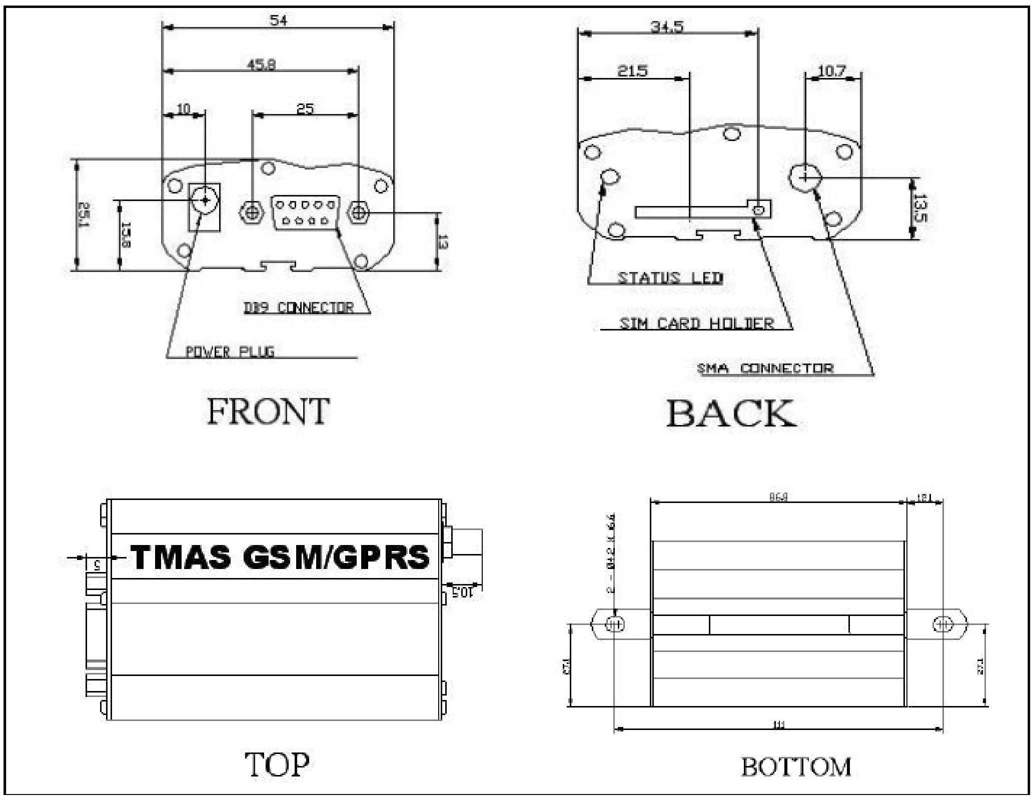


Figure 2.10 : GSM Schematic

CHAPTER 3

ELECTRONIC OVERVIEW

3.1 Microcontroller Unit (MCU)

The microcontroller chip that has been selected for the purpose of controlling the data logger is PIC18F452 manufactured by MicroChip . This chip is selected based on several reasons:

i. It has internal FLASH and RAM and its memory capacity is large enough for programming to control the data logger.
ii. Its size is small and equipped with 34 general-purpose I/O lines.
iii. It is portability and current consumption.
iv. It supports serial interfacing with the GPS module and GSM Module.

3.1.1 Key features of the microcontroller

The PIC18F452 features a 'C' compiler friendly development environment, 256 bytes of EEPROM, Self-programming, an ICD, 2 capture/compare/PWM functions, 8 channels of 10-bit Analog-to-Digital (A/D) converter, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the2-wireInter-

Integrated Circuit (I²C™) bus and Addressable Universal Asynchronous Receiver Transmitter (AUSART).

Table 3.1 : Key features for PIC18F452

Features	PIC18F242	PIC18F252	PIC18F442	PIC18F452
Operating Frequency	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz
Program Memory (Bytes)	16K	32K	16K	32K
Program Memory (Instructions)	8192	16384	8192	16384
Data Memory (Bytes)	768	1536	768	1536
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	17	17	18	18
I/O Ports	Ports A, B, C	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART
Parallel Communications	—	—	PSP	PSP
10-bit Analog-to-Digital Module	5 input channels	5 input channels	8 input channels	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions	75 Instructions	75 Instructions	75 Instructions
Packages	28-pin DIP 28-pin SOIC	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin PLCC 44-pin TQFP	40-pin DIP 44-pin PLCC 44-pin TQFP

3.1.2 MCU Block Diagram

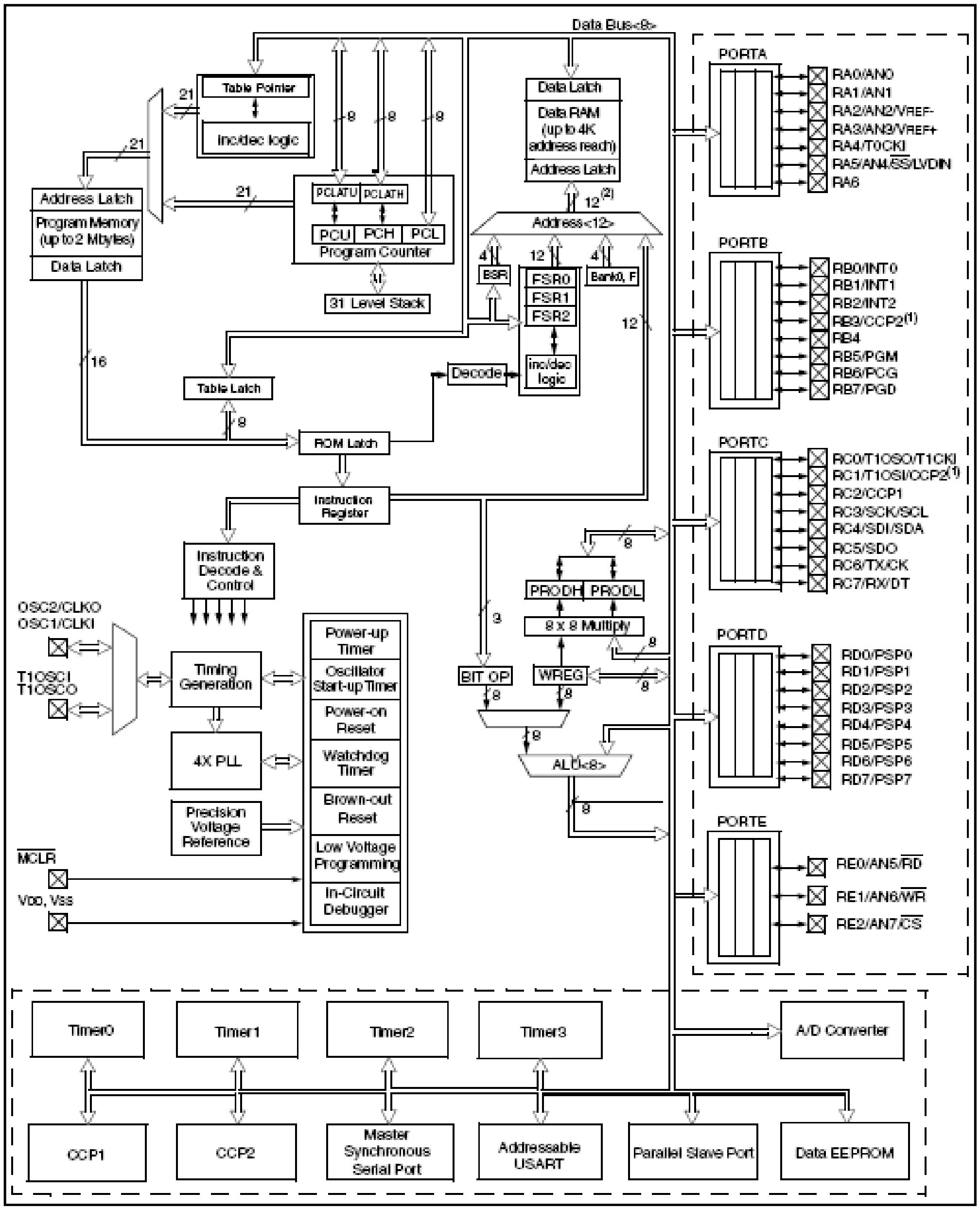


Figure 3.1 : PIC18F452 Block Diagram

3.1.3 MCU Pin Diagram

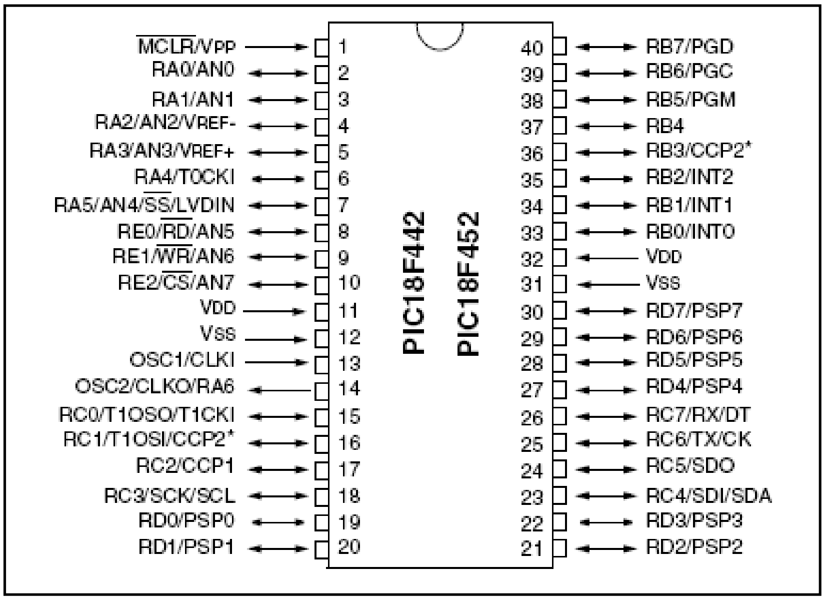


Figure 3.2 : PIC18F452 Pin Diagram

3.1.3 Memory Organization

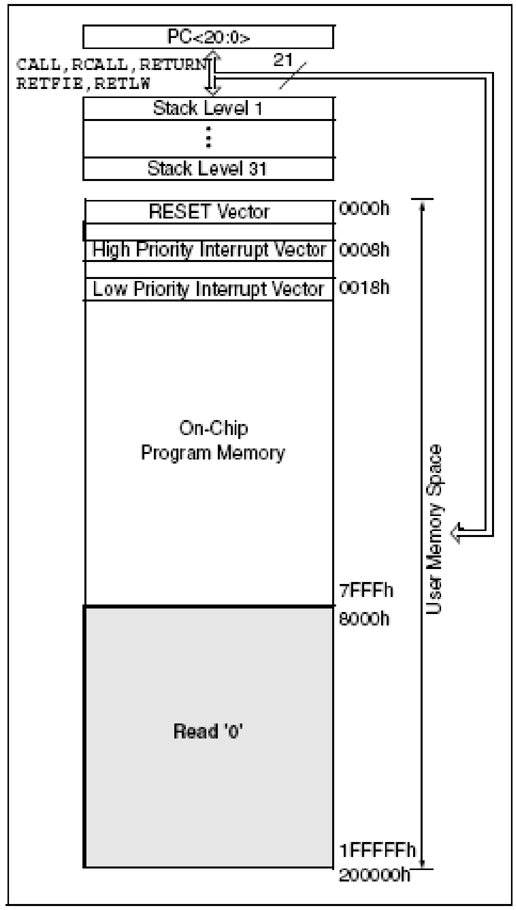


Figure 3.3 : PIC18F452 Memory Diagram

There are three memory blocks in Enhanced MCU devices. These memory blocks are:

- Program Memory
- Data RAM
- Data EEPROM

Data and program memory use separate busses, which allows for concurrent access of these blocks.

3.2 Module Interface

For this project, the connection between microcontroller unit (MCU) and modules are using three type of interface features in PIC18F452. There are:

i. Serial Communication Interface (SCI) for interfacing GSM Module and GPS Module.
ii. Serial Peripheral Interface (SPI) for interfacing Multimedia card holder to the MCU
iii. Parallel Communication Interface for connection to the LCD.

3.2.1 Serial Communication Interface (SCI)

One of important features available in this MCU which is useful in this project is Serial Communication Interface (SCI). With this function, communication between MCU and the GSM and GPS module can be established. Therefore, the command can be delivered to microcontroller trough IC namely MAX 233. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate.

The PIC18F452 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The USART Asynchronous module consists of the following important elements:

i. Baud Rate Generator
ii. Sampling Circuit
iii. Asynchronous Transmitter
iv. Asynchronous Receiver

3.2.1.1 Serial Communication Interface (SCI) between MCU and GPS module

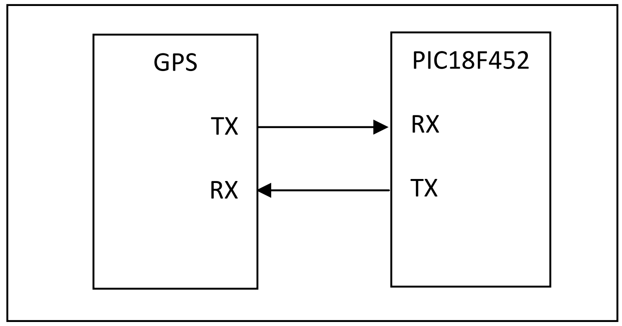


Figure 3.4 : Serial Communication between GPS Module and MCU

3.2.1.2 Serial Communication Interface (SCI) between MCU and GSM module

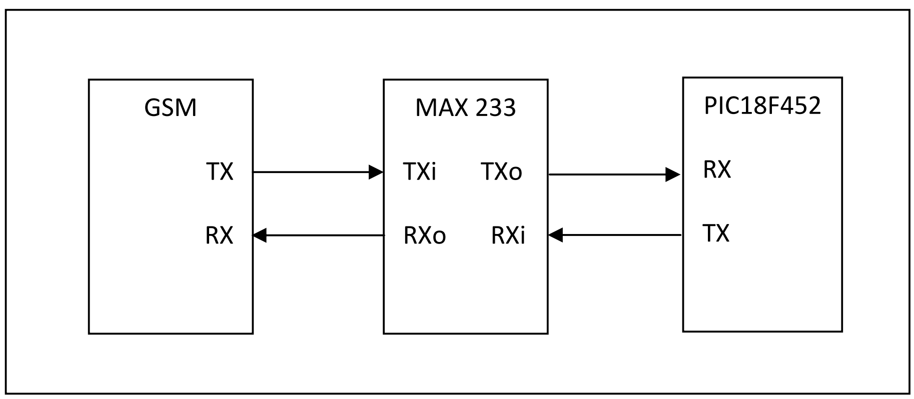


Figure 3.5 : Serial Communication between GSM Module and MCU trough MAX 233

3.2.2 Serial Peripheral Interface (SPI)

The second communication features that used in this project is Serial Peripheral Interface (SPI). This type of interface used to communicate between MCU and Multimedia Card.

The SPI mode allows 8-bits of data to be synchronously transmitted and received, simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

i. Serial Data Out (SDO) - RC5/SDO
ii. Serial Data In (SDI) - RC4/SDI/SDA
iii. Serial Clock (SCK) - RC3/SCK/SCL/LVDIN

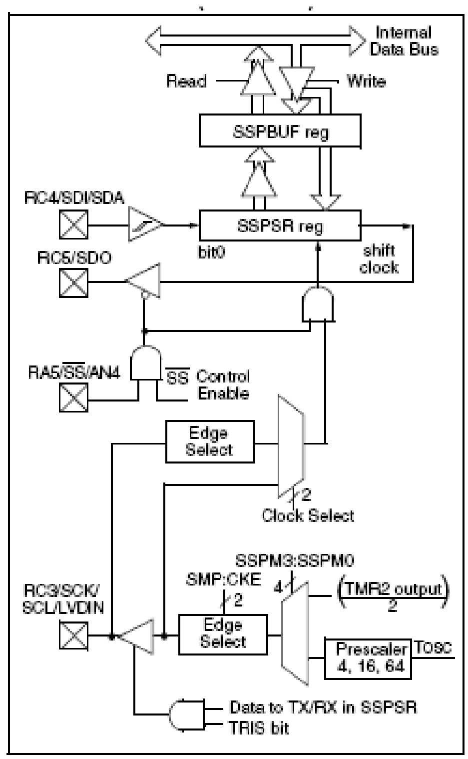


Figure 3.6 : Internal SPI Block Diagram in PIC18F452

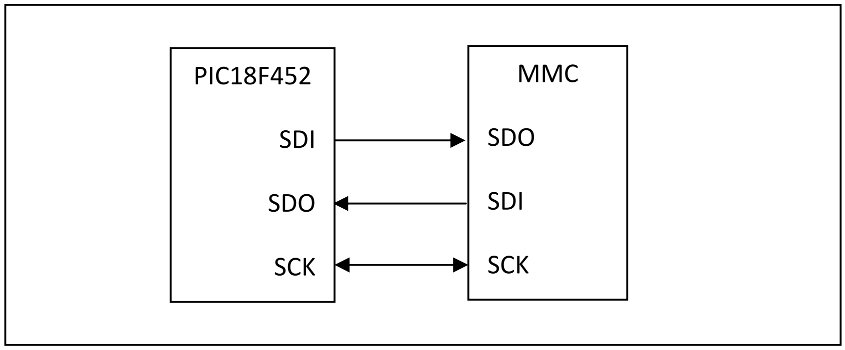


Figure 3.7 : Serial Peripheral Interface between PIC18F452 and Multimedia Card

3.2.3 Parallel I/O Interface

The last communication features that used in this project is Parallel Communication Interface. In this project, I used parallel I/O port from PIC18F452 to send data to the LCD Module. The data is to be transfer by 4 bits parallel port. The circuit for this connection will be discussed on the next chapter.

3.3 Voltage Regulator

One method to reduce voltage of a power supply is to run it trough a linear voltage regulator, such as LM 7805. For all voltages within the regulator’s pre-scribed limits (a minimum of 7.5 volts to a maximum of 25 volts), the output of a 7805 will be even 5 volts.

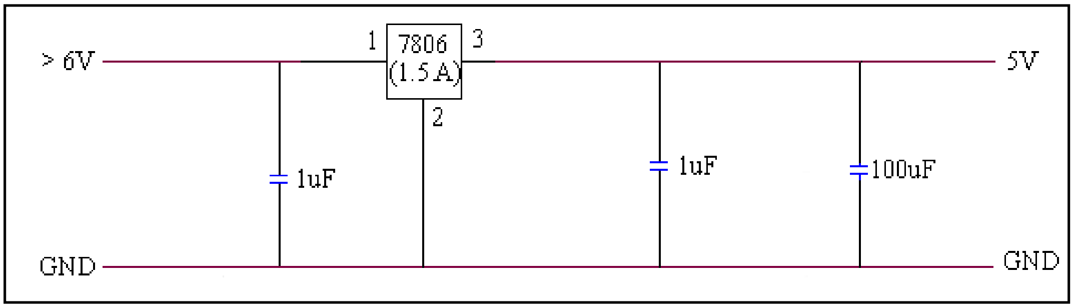


Figure 3.8 : 5 Volts voltage regulator

CHAPTER 4

SOFTWARE OVERVIEW

4.1 Introduction

This project has utilized C programming language. The whole program to the microcontroller unit (MCU) is compiled under basic c language. This chapter will discuss about all the software that used in order to make this project successful.

All the software has its own description such as for simulation, setting the GPS module and programmed the MCU. There are list of the software and its description:

i. PIC C Compiler Software (Write the High Level Language for MCU)
ii. Proteus 7 Professional (Simulation for the whole project)
iii. Mini GPS (software that use to read data from GPS and to separate the GPS NMEA Protocol)
iv. PIC kit 2 (software to burn data from PC to MCU)

4.2 PIC C Compiler Software

This software is used to write the c programming language to the MCU. The compiler contains Standard C operators and built-in libraries that are specific to the PIC registers. Access to hardware features from C. Features include:

i. 1, 8, 16 and 32 bit integer types and 32 bit floating point.
ii. Standard one bit type (Short Int) permits the compiler to generate very efficient Bit oriented code.
iii. #BIT and #BYTE will allow C variables to be placed at absolute addresses to map registers to C variables.
iv. Bit Arrays
v. Fixed Point Decimal
vi. Constants (including strings and arrays) are saved in program memory.
vii. Flexible Handling Of Constant Data
viii. Variable Length Constant Strings
ix. Address mod Capability To Create User Defined Address Spaces In Memory Device

The CCS C Compiler for PIC10, PIC12, PIC14, PIC16, and PIC18 microcontrollers has over 180 Built-in Functions to access PIC® MCU hardware is easy and produces efficient and highly optimized code.

The compiler can handle inline or separate functions, as well as parameter passing in re-usable registers. Transparent to the user, the compiler handles calls across pages automatically and analyzes program structure and call tree processes to optimize RAM and ROM Usage.

The compiler runs under Windows 95, 98, ME, NT4, 2000, XP, Vista, or Linux. It outputs hex and debug files that are selectable and compatible with popular emulators and programmers including the MPLAB® IDE for source level debugging.

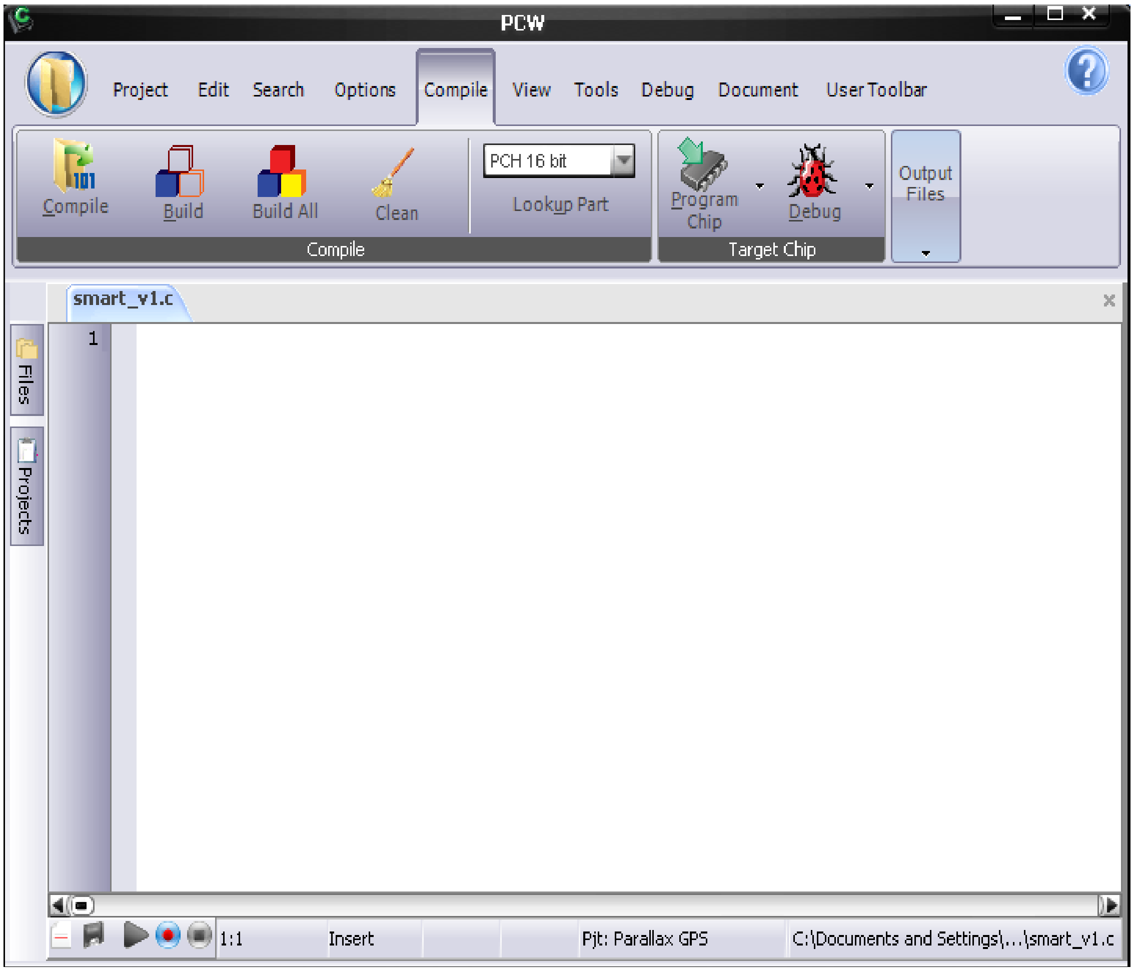


Figure 4.1 : Interface of PIC C Compiler software

4.3 Proteus 7 Professional Simulator

For this project, this Proteus 7 Professional software is used to do the simulation of the project. To make sure that all the system is running based on the theoretical side, the simulation is very important to this project.

According to that, all kind of connection will be design in this software first before assemble them in electronic design. The programmings for the MCU also will be simulated in this software before apply it to the real MCU to make sure that the programming is correct.

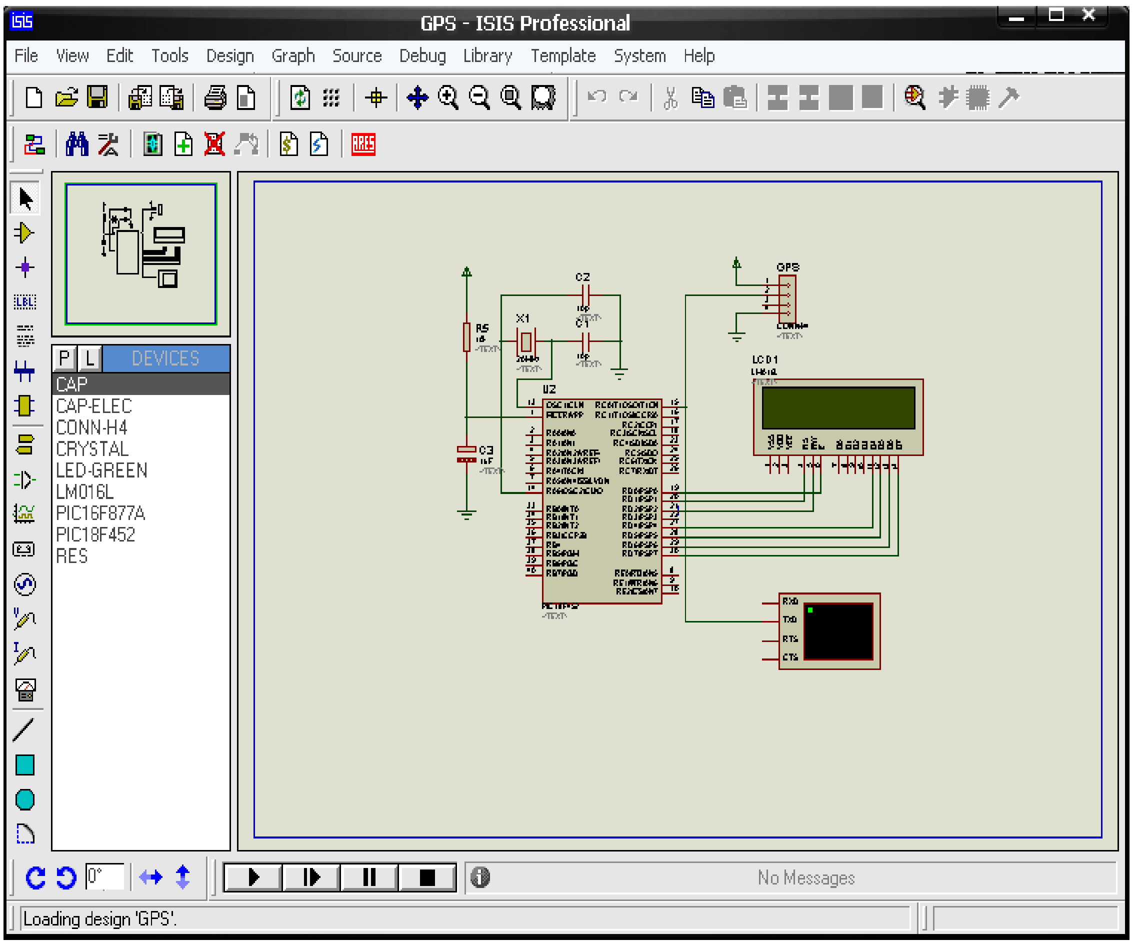


Figure 4.2 : Interface of Proteus 7 Professional Simulator

4.3 Mini GPS

This software is used to test the GPS receiver either it valid or not. From this software, we can change the protocol of the GPS receiver that we want to use. For this project, I used this software to get the RMC protocol only from the GPS receiver.

This software also available to test the signal of GPS receiver. This software can be download from the Sparkfun web site at:

http://www.sparkfun.com/datasheets/GPS/MiniGPS_1.32.zip

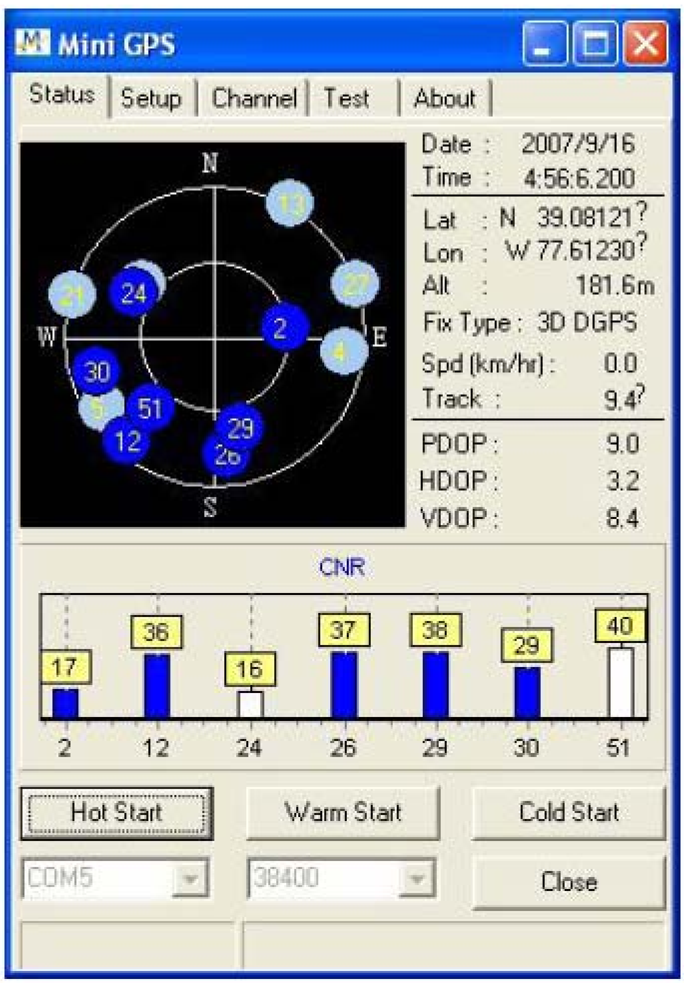


Figure 4.3 : Mini GPS Module Status

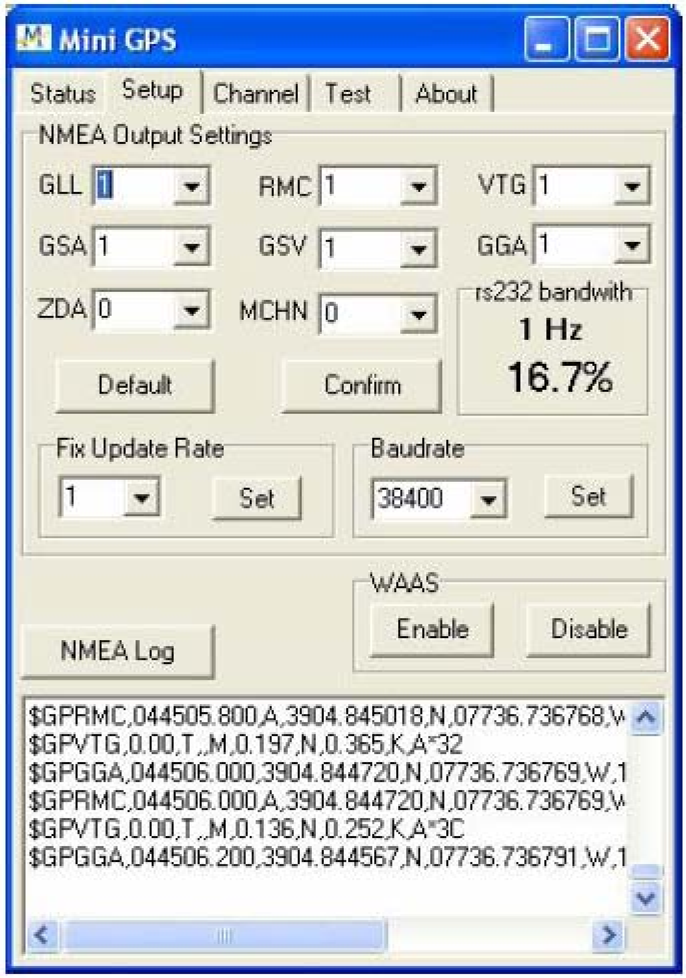


Figure 4.4 : Mini GPS Data Status

Figure show that the protocol from GPS Receiver output using the GPS receiver. We can choose any protocol that we want.

4.4 PIC Kit 2 Programmer

UIC00A offers low cost yet reliable and user friendly PIC USB programmer solutions for developer, hobbyist and students. It is designed to program popular Flash PIC MCU which includes PIC12F, PIC16F and PIC18F family. It can also program 16bit PIC MCU. On board ICSPTM (In Circuit Serial Programming) connector offers flexible method to load program. It supports on board programming which eliminate the frustration of plug-in and plug-out PIC MCU.

This also allow user to quickly program and debug the source code while the target PIC is on the development board. Since USB port have become a popular and widely used on Laptop and Desktop PC, UIC00A is designed to be plug and play with USB connection. This programmer obtained it power directly from USB connection, thus O external power supply is required, making it a truly portable programmer. This programmer is ideal for field and general usage. UIC00A offers reliable, high speed programming and free windows interface software.

For this project, PIC Kit 2 Programmer is used to program the .hex file into MCU Unit. Figure 4.4 shows the PIC kit 2 Programmer interface in Microsoft Windows XP operating system.

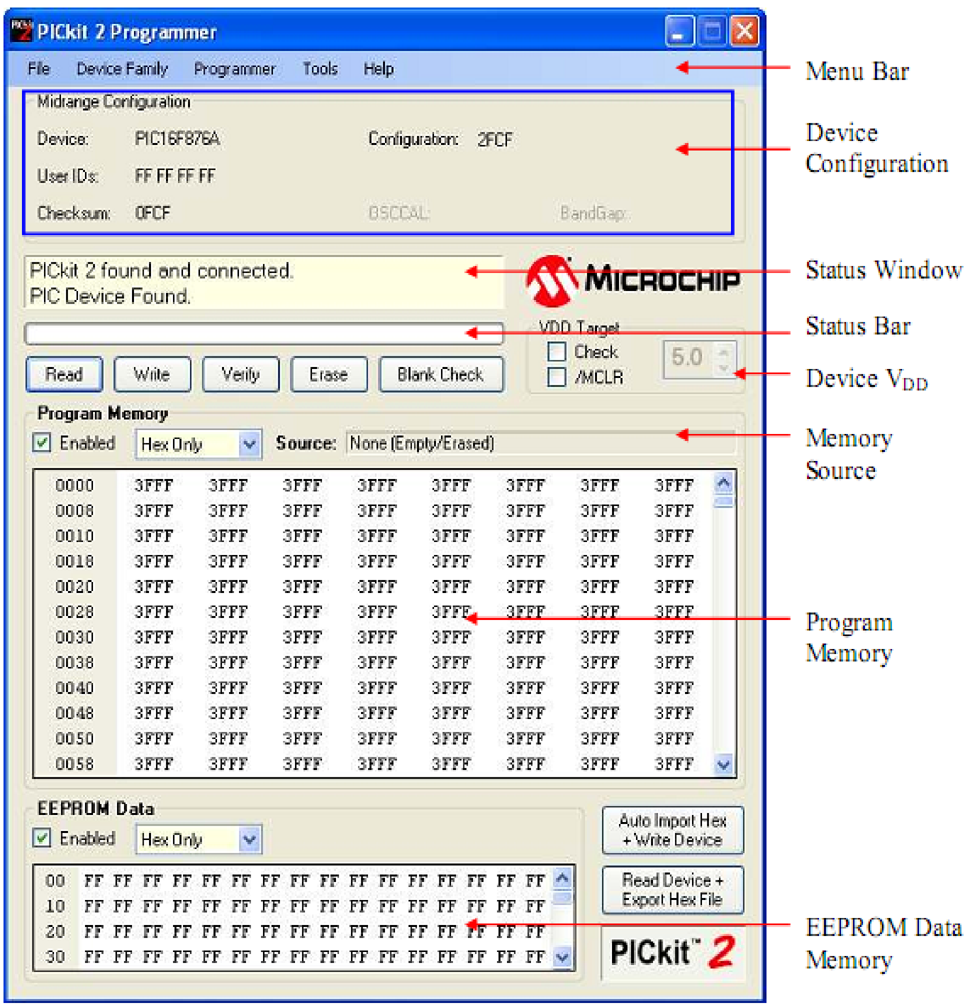


Figure 4.5 : PIC kit 2 Programmer interface

CHAPTER 5

SMART VEHICLE DATA LOGGER SYSTEM
(PROJECT IMPLEMENTATION)

5.1 Introduction

This chapter will explain how this project is implemented using hardware and software. The hardware is consist of microcontroller PIC18F452, GPS receiver, GSM Modem, Multimedia card and LCD Module.

5.2 Hardware setup

Microcontroller PIC18F452 is the brain of the hardware, used to communicate GPS data to MMC, LCD and GSM Modem. The system can be describe trough the block diagram below on the next page.

i. GPS to PIC

All data from GPS Module will be extracted by PIC. The speed and the coordinate can be detected by GPS module through the command from PIC.

ii. PIC to LCD

The LCD will be display the speed and location of the vehicle by the data from GPS Module.

iii. PIC to Multimedia Card (MMC)

The data from GPS will be store in MMC card through the command from PIC. It can be write by using MMC data writer.

iv. Two way communication using GSM

Input from user, SMS , that had been sent to the GSM module will be read in PIC. PIC will take the data (location and coordinate) from GPS and send the data back to user’s cell phone using GSM Module.

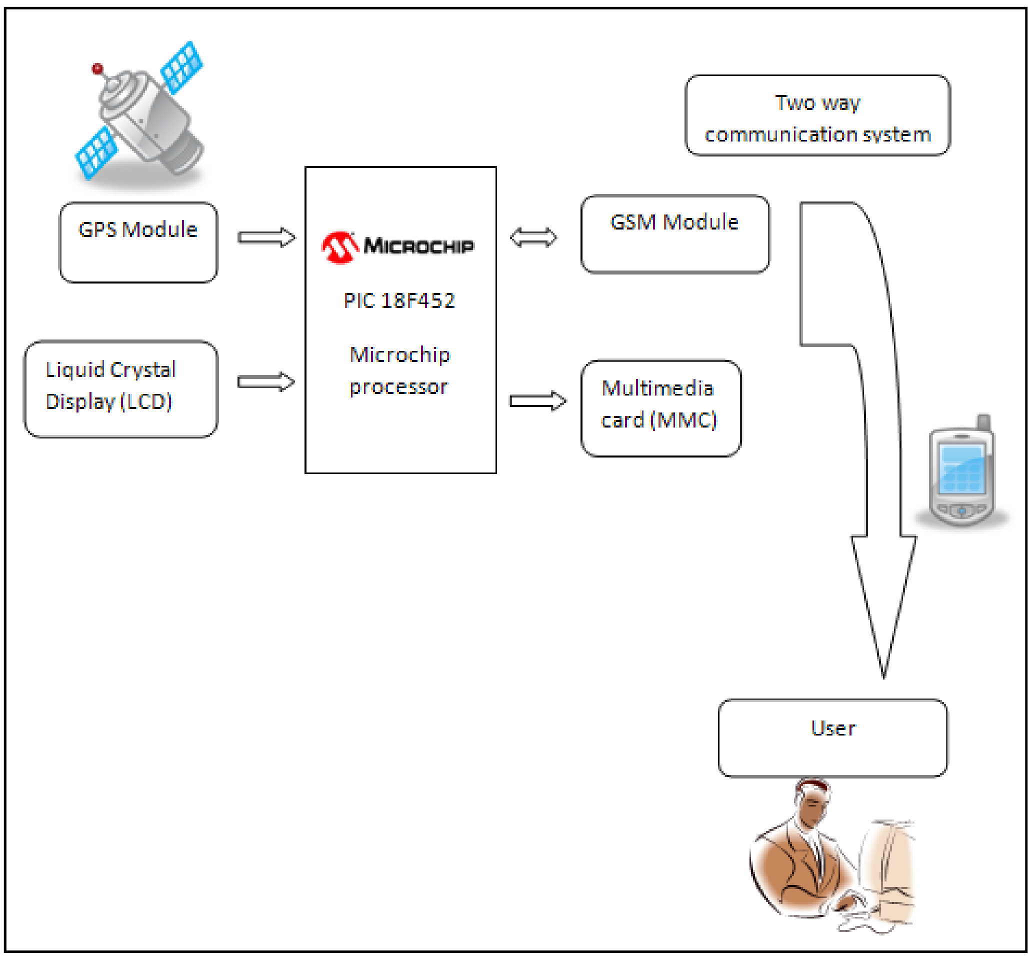


Figure 5.1a : System Block diagram

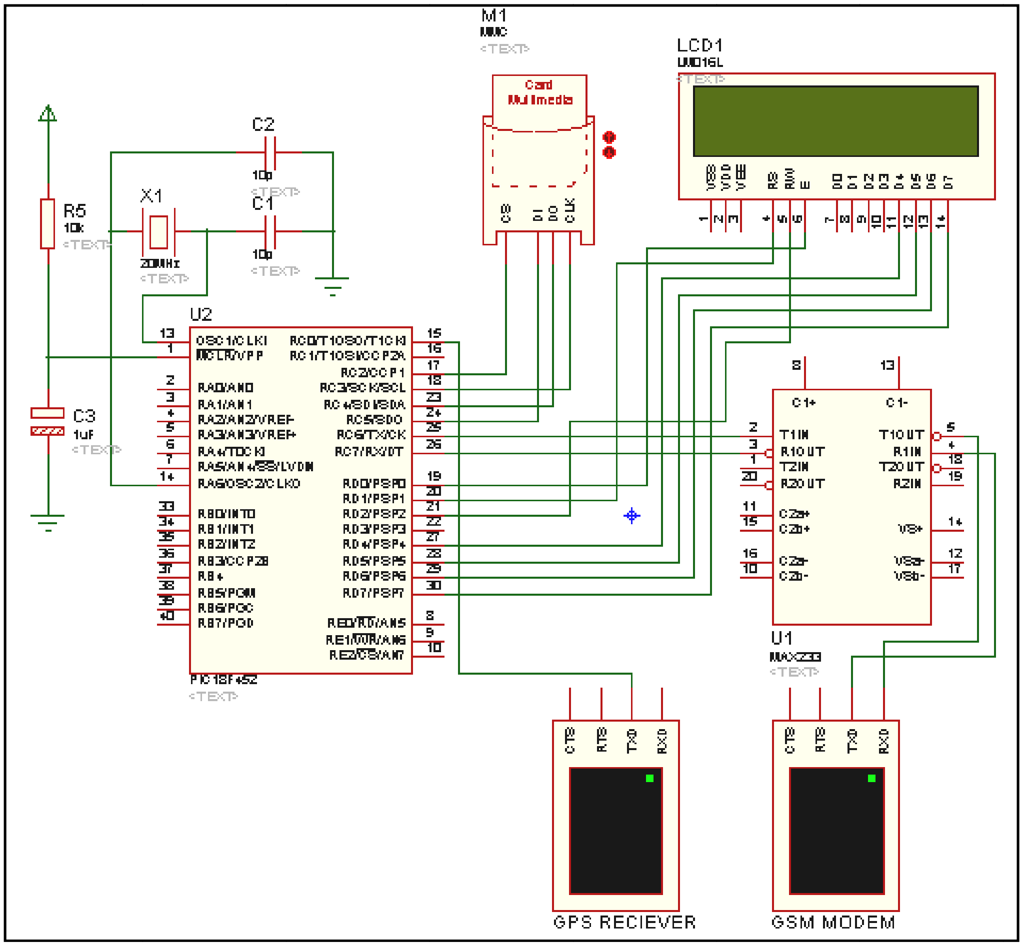


Figure 5.1b : System Overall Circuit

Figure 5.1b show the overall circuit for the system. The connection of the circuit are designed based on the configuration of MCU and the modules connection types such as Serial Communication interface (SCI) and Serial Peripheral Intrface (SPI). The pin connection of each module will be discuss on the next sub topic.

5.2.1 Microcontroller Setup

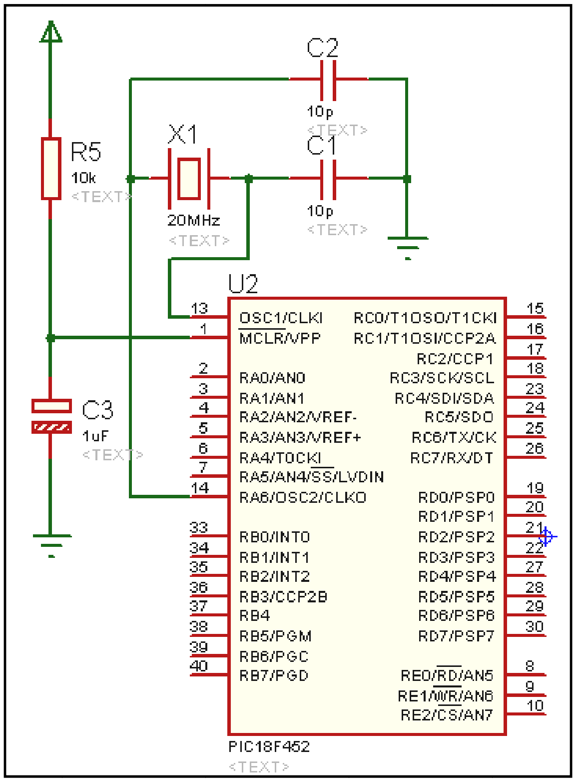


Figure 5.2 : The PIC18F452 schematic

Figure 5.2 shows the microcontroller PIC18F452 schematic. The schematic include the crystal circuit and reset circuit. Port D is used to connect the LCD Module. Pin C0 and Pin C1 are for serial communication to the GPS Module. Pin C6 and Pin C7 are for serial communication to the GSM Modem.

5.2.2 GPS receiver setup

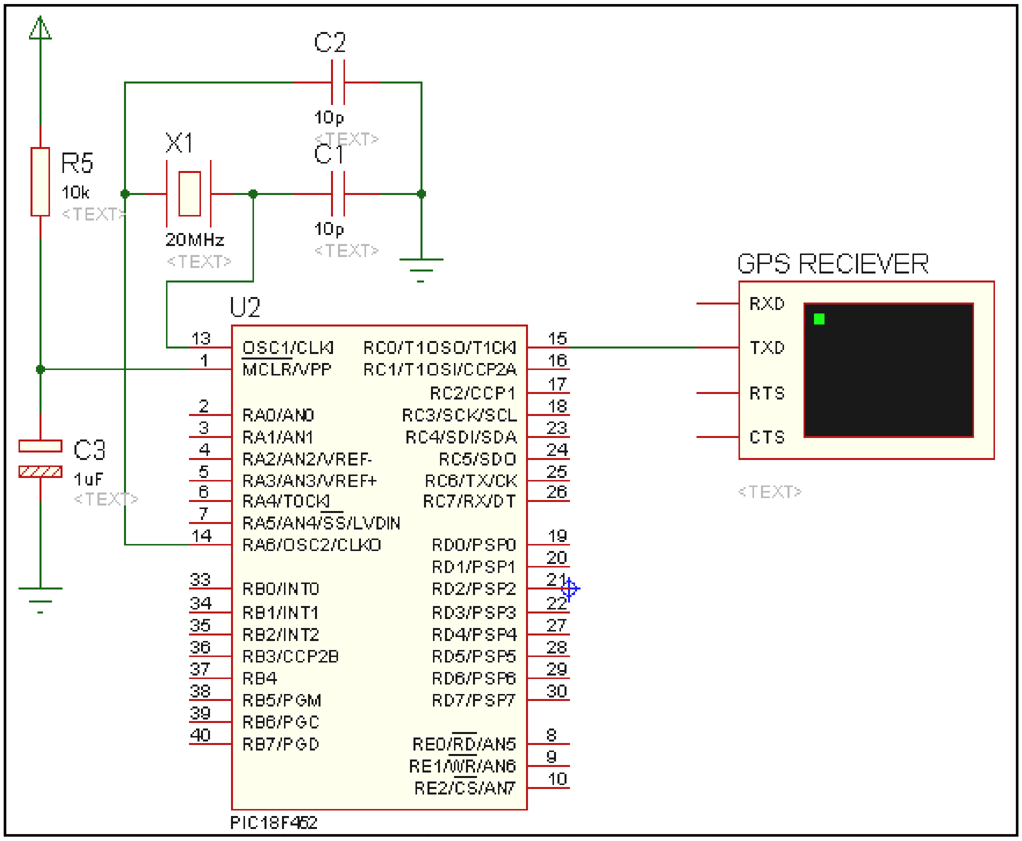


Figure 5.3 : GPS receiver circuit

Figure 5.3 shows that the connection of GPS receiver to the MCU. Pin C0 from PIC18F452 had been set to be RX pin for serial communication. Then, TX pin from GPS receiver is connected to it. The output system from GPS receiver is 5 volts, so it is compatible to the MCU. According to this, the serial connections between them are directly connected without using RS232 circuit.

5.2.3 GSM Modem setup

For this system, the GSM modem and MCU are connected using serial communication. The connection from GSM modem to MCU is trough IC namely MAX233. The function of MAX233 IC is to convert voltage from GSM Modem which is 12 volts data to become 5 volts data that can be read by microcontroller unit.

The figure 5.4 shows that the connections of GSM modem and MCU through MAX233 system. We also can use MAX232 IC for this system, but for this IC , there are no built in capacitor.

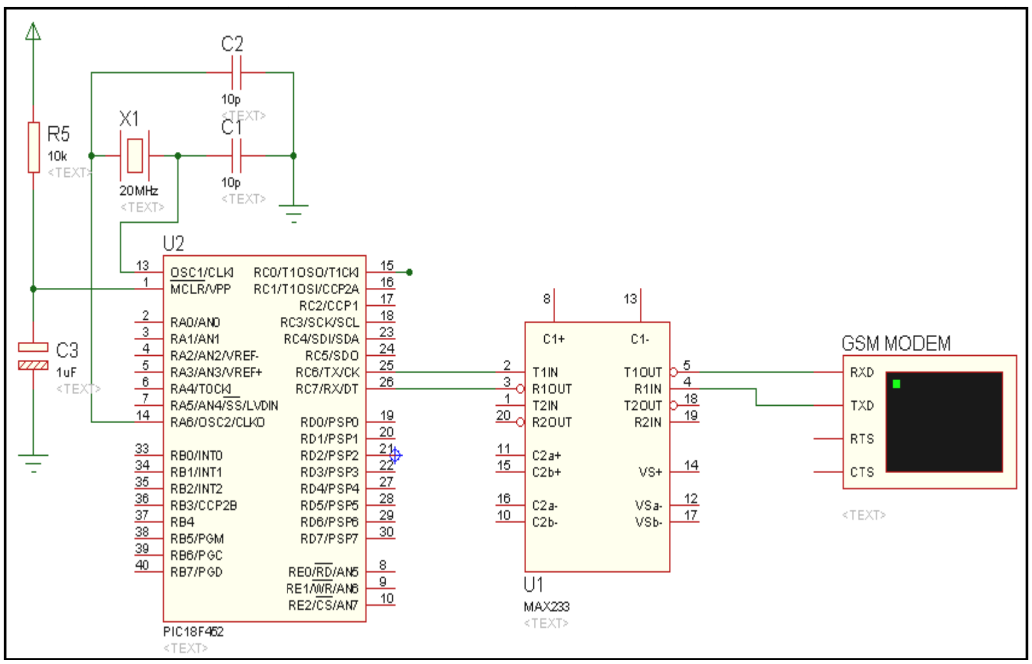


Figure 5.4 : GSM Modem circuit

5.2.3 Multimedia card setup

For Multimedia card, the connection is using Serial Peripheral Interface (SPI). Pin RC2 from MCU is connected to the CS pin in MMC. Pin RC3 is connected to CLK, Pin RC4 is connected to D0 and pin RC5 is connected to DI in MMC. The table below show that the SPI connection.

Table 5.1 : SPI connection

MCU PIC18F452 Pin	**Multimedia Card Pin**
RC2	CS (Chip Select)
RC3/SCK	CLK
RC4/SDI	DO
RC5/SDO	DI

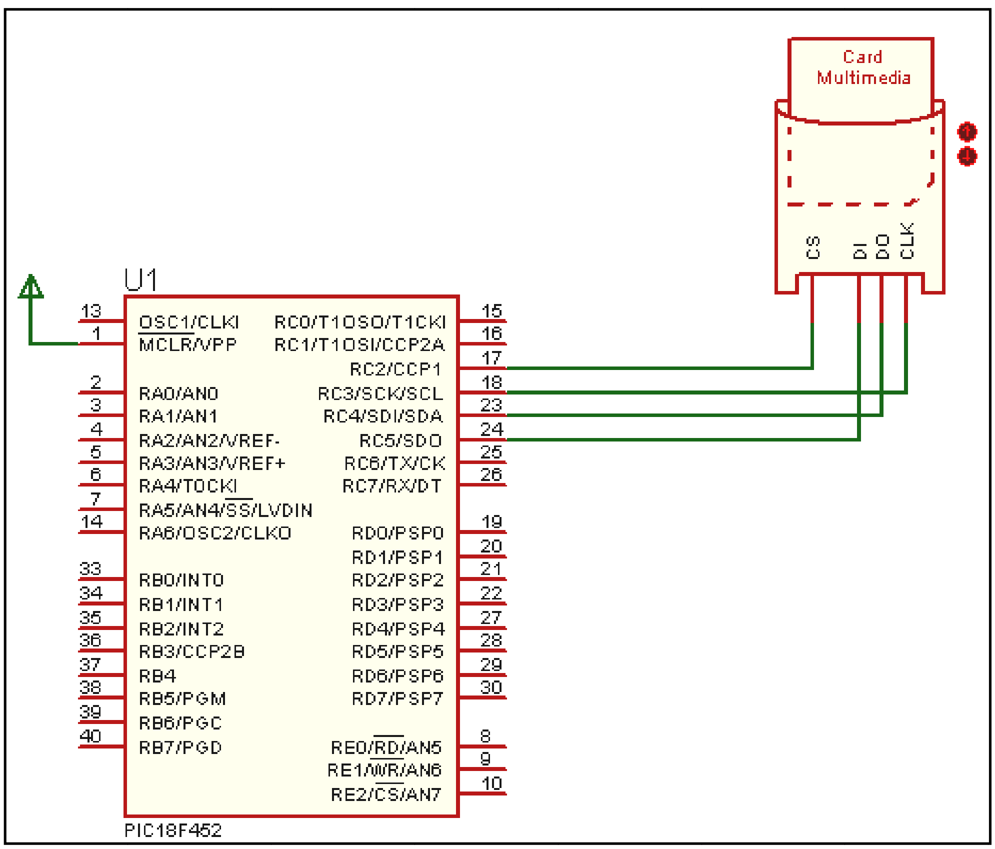


Figure 5.5 : SPI communication between MCU and MMC

5.2.4 LCD Module setup

In this project, Parallel I/O port is used to connect MCU to the LCD Module. Port D parallel port is used to connect it. This LCD module just used 4 data bit line for the data transfer. The data pins are RD4 to RD7. The connection is shown at Figure 5.6.

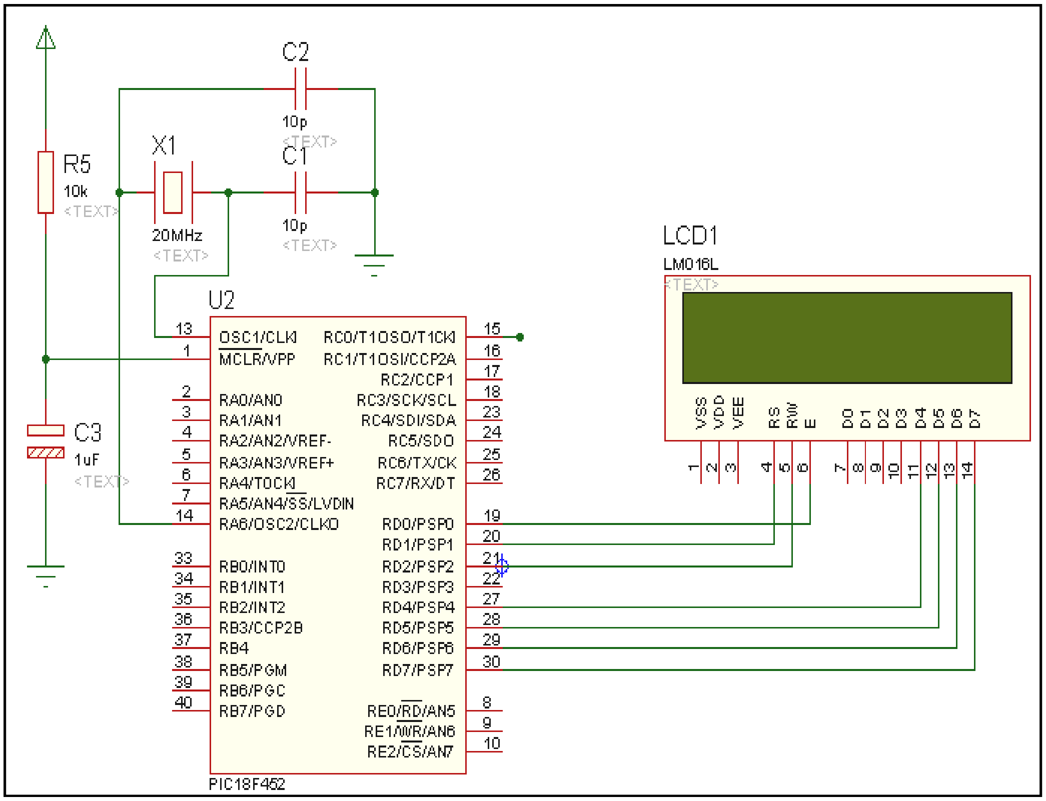


Figure 5.6 : LCD connection

5.3 Software setup

For microcontroller to function, Program Flash Memory has to be loaded with program. The program is basically used to analysis data from GPS receiver , store the data in MMC module and last but not least control the GSM module. The overall source code for MCU will be shown at Appendix A.

5.3.1 GPS receiver source code

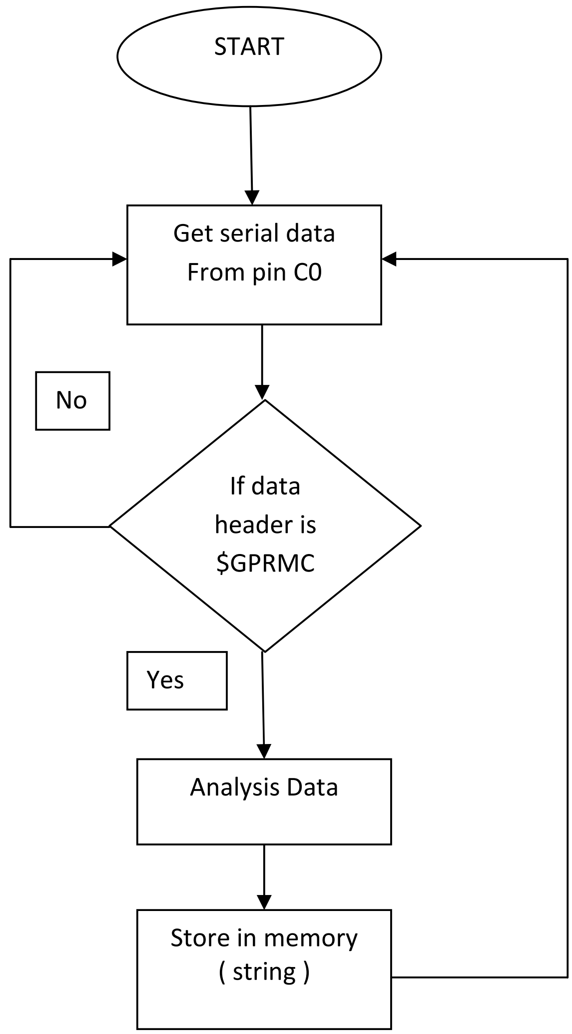


Figure 5.7 : GPS Module Flow Chart

Data from GPS receiver will be sent to MCU at Pin C0 using serial communication. The program will scan the $GPRMC header and analysis the data. Then, the data will be store in memory in string type.

5.3.2 GSM Modem source code

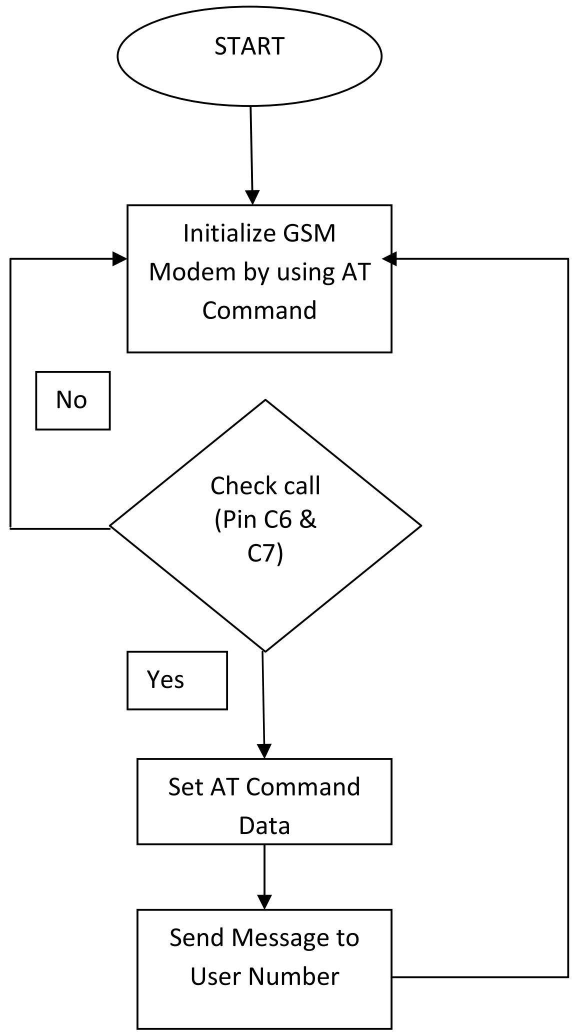


Figure 5.8 : GSM Modem Flow Chart

The GSM Modem system is using AT Command to operate it. There are two step to initialize the GSM Modem system using AT command:

AT+CMGF=1

AT+CSCA=”+60120000015”

After initialize the GSM Modem, the system will check the string ‘*RING*’ from Pin C6 of the MCU to check the call from GSM Modem. If there is a call, the system will send two AT command to the GSM Modem to send SMS. The At command for sending SMS are :

AT+CMGF=1

AT+CMGS=”(Number Reciever)”

5.3.3 MMC source code

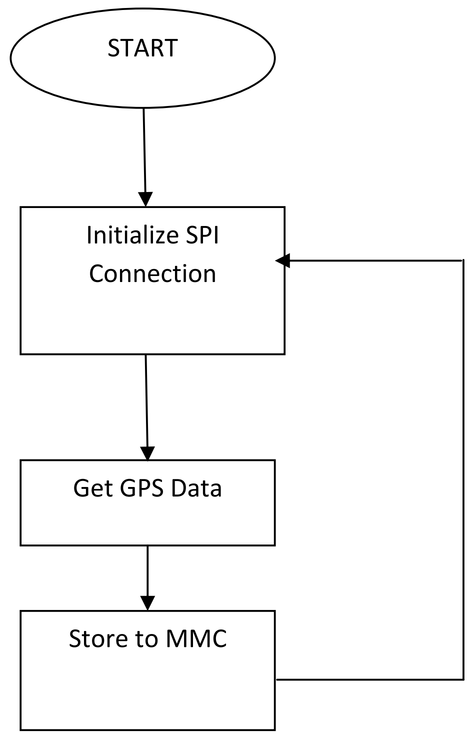


Figure 5.8 : MMC Source Code Flow Chart

Serial Peripheral Interface (SPI) is used to communicate the MMC system to the MCU system. The system will store 512 bytes of data at one time into the memory card. Figure 5.8 show how MCU operation step to get the 512 bytes data from GPS and store it into MMC. The system will always store the GPS data when there are any data received from GPS reciver.

CHAPTER 6

RESULT AND ANALYSIS

6.1 Introduction

In this project, there are several stages that will be construct. For the first stage , the GPS Evaluation Board had been design to get the data from GPS Receiver. For the second stage, the real hardware development.

First stage :

i. GPS Evaluation Board (*Interfacing GPS to Hyper Terminal using RS232 DB9 connection.*)
ii. Simulation connection of GPS, PIC 18F452 Microcontroller and LCD Module using ISIS 7 Professional Software.

6.2 GPS Evaluation Board (*Interfacing GPS to Hyper Terminal using RS232 DB9 connection*)

This circuit is using MAX232 IC to interfacing data from GPS / Rx and Tx input to DB9 connector. Please refer to Appendix B for the Max232 Datasheet.

The GPS receiver module is Parallax GPS receiver Module. Please refer to Appendix C for the Parallax GPS receiver Module Datasheet.

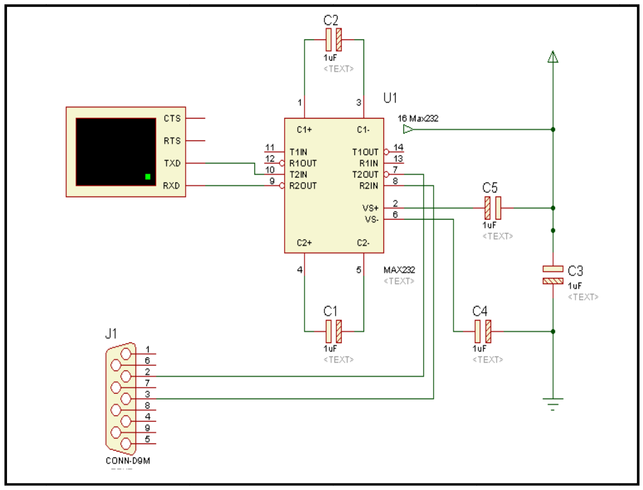


Figure 6.1 : GPS Evaluation Board Diagram for Interfacing Circuit

Figure 6.5 : GPS Evaluation Board Hardware Design

6.2.1 Result from Hyper Terminal

Table 6.1 : Output from Hyper Terminal

```
$GPGSV,3,1,12,24,77,238,37,23,66,148,40,13,64,160,,20,56,084,38*7A
$GPGSV,3,2,12,04,40,296,39,07,30,232,,25,25,052,32,11,15,143,*7F
$GPGSV,3,3,12,01,09,042,,02,09,306,33,30,03,013,,05,02,317,*7A
$GPRMC,143252.99,A,5459.8219,N,00141.4886,W,000.0,010.8,020406,003.5,W,A
*30
$GPGGA,143252.99,5459.8219,N,00141.4886,W,1,06,03.1,00096.5,M,0048.2,M,,*
44
$GPGSA,A,3,24,23,,20,04,,25,,,02,,,04.8,03.1,03.7*0C
$GPGSV,3,1,12,24,77,238,37,23,66,148,40,13,64,160,,20,56,084,38*7A
$GPGSV,3,2,12,04,40,296,39,07,30,232,,25,25,052,32,11,15,143,*7F
$GPGSV,3,3,12,01,09,042,,02,09,306,33,30,03,013,,05,02,317,*7A
$GPRMC,143253.99,A,5459.8220,N,00141.4886,W,000.0,010.8,020406,003.5,W,A
*3B
$GPGGA,143253.99,5459.8220,N,00141.4886,W,1,06,03.1,00096.8,M,0048.2,M,,*
42
$GPGSA,A,3,24,23,,20,04,,25,,,02,,,04.8,03.1,03.7*0C
$GPGSV,3,1,12,24,77,238,37,23,66,148,40,13,64,160,,20,56,084,38*7A
$GPGSV,3,2,12,04,40,296,39,07,30,232,,25,25,052,32,11,15,143,*7F
$GPGSV,3,3,12,01,09,042,,02,09,306,33,30,03,013,,05,02,317,*7A
$GPRMC,143254.99,A,5459.8221,N,00141.4886,W,000.1,010.8,020406,003.5,W,A
*3C
$GPGGA,143254.99,5459.8221,N,00141.4886,W,1,05,04.6,00097.2,M,0048.2,M,,*
4C
$GPGSA,A,3,24,23,,20,04,,25,,,,,,08.7,04.6,07.4*0A
$GPGSV,3,1,12,24,77,238,37,23,66,148,40,13,64,160,,20,56,084,38*7A
$GPGSV,3,2,12,04,40,296,39,07,30,232,,25,25,052,32,11,15,143,*7F
$GPGSV,3,3,12,01,09,042,,02,09,306,32,30,03,013,,05,02,317,*7B
$GPRMC,143255.99,A,5459.8222,N,00141.4887,W,000.1,010.8,020406,003.5,W,A
*3F
$GPGGA,143255.99,5459.8222,N,00141.4887,W,1,06,03.1,00097.5,M,0048.2,M,,*
4B
```

6.2.2 GPS Data Analysis

By using hyper terminal, the output from GPS receiver can be display.

$GPGSV,3,3,12,01,09,042,,02,09,306,32,30,03,013,,05,02,317,*7B
$GPRMC,143255.99,A,5459.8222,N,0141.4887,W,000.1,010.8,250108,003.5,W,A*
$GPGGA,143255.99,5459.8222,N,00141.4887,W,1,06,03.1,00097.5,M,0048.2,M,,*
$GPGSA,A,3,24,23,,20,04,,25,,,,,,08.7,04.6,07.4*0A

In this project, two protocols were used to get the coordinate, speed, real time and date. Table 6.2 and Table 6.3 show RMC Protocol Data Format and GGA Protocol Data Format.

Table 6.2 : RMC Protocol Data Format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Position	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	Knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation[1]		degrees	E=east or W=west
Checksum	*10		
<CR> <LF>			End of message termination

[1] SiRF does not support magnetic declination. All "course over ground" data are geodetic WGS-84 directions.

Table 6.3 : GGA Protocol Data Format

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-3
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude[1]	9.0	meters	
Units	M	meters	
Geoid Separation[1]		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

[1] SiRF does not support geoid correction. Values are WGS-84 ellipsoid heights.

From RMC protocol

a) Coordinate 5459.8222,N,0141.4887,W	Latitude : ddmm.mmmm N/S Indicator : N=north or S=south Longitude : dddmm.mmmm E/W Indicator : E=east or W=west ***So, the location is 54 degrees 59.8222 minutes to the North and 01 degrees 141.4887 to the West from GMT***
b) Speed 0.1	***Means the speed is 0.1 knots (0.1 miles per hour)***
c) Date 250108	Date : ddmmyy ***Means the date is 25 January 2008***

From GGA protocol

a)Real Time 143255.99	UTC Position : hhmmss.sss ***So, the time is 14 : 32 : 55.99 but, in Malaysia is +8 hours from GMT***
b) Satellites Used 06	***Means that there are 6 satellites used to get the data.***

6.3 Simulation connection of GPS, PIC 18F452 Microcontroller and LCD Module using ISIS 7 Professional Software.

The simulation is using ISIS 7 Professional Software. The overview of the software had been discussed on the Chapter 4. This topic will discuss about the result of the simulation.

6.3.1 Circuit Diagram

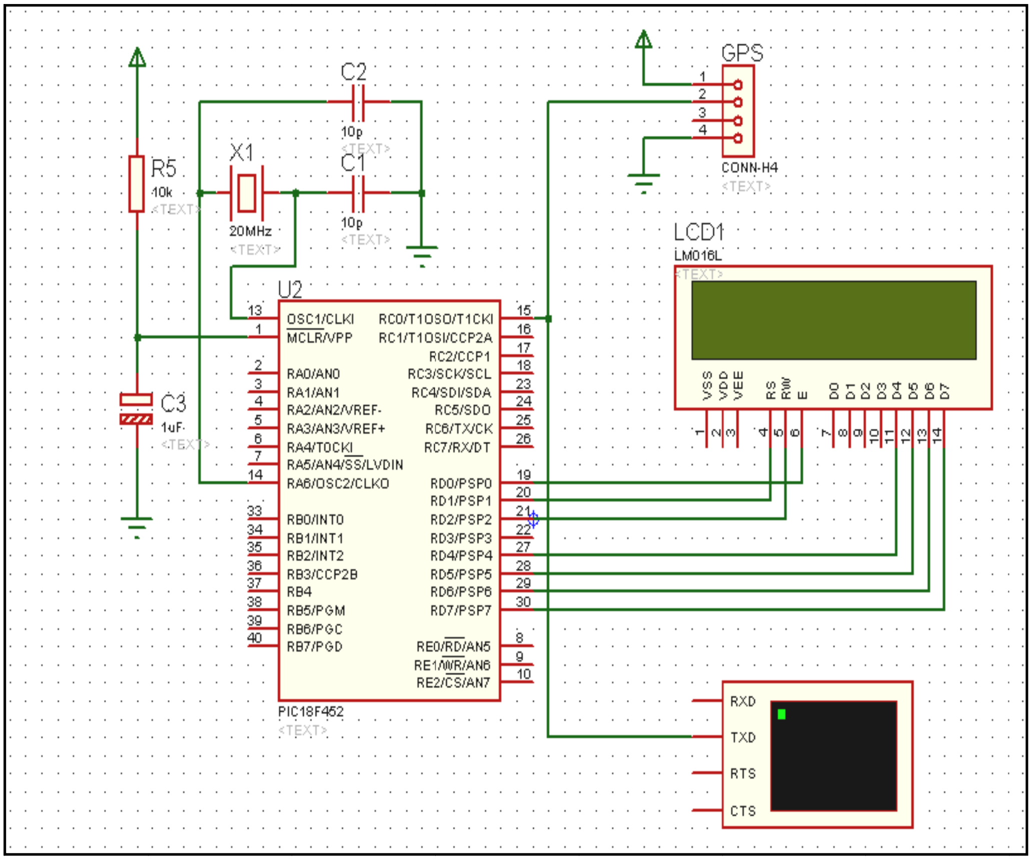


Figure 6.3 : Circuit diagram for connection of GPS, PIC 18F452 Microcontroller and LCD Module in ISIS 7 Professional Software.

6.3.2 Source Code for PIC18F452 Microcontroller / C Programming for PIC18F452

In this stage, we use high level language for programming the PIC18F452 Microcontroller. Refer Appendix A for the source code for the microcontroller for this simulation.

6.3.3 Input From Virtual Memory in the Simulation

In this simulation, there is virtual memory can be put as GPS receiver input. The input had been paste as below to the virtual memory.

Table 6.4 : Input for Virtual Memory in ISIS Professional 7 Simulation

```
$GPRMC,165208.000,A,0345.445734,N,10311.309269,E,1.284,0.00,200108,,,A*50
$PMTKCHN,30382,01302,22332,31312,32302,14001,29001,02001,16001,05001,20001,28031,1103
1,18001,06001,19001,07001,09001,25001,08001,13001,10001,03001,04001,27001,26001,15001,170
01,12001,00000,00000,00000*49
,17001,12001,00000,00000,00000*49
$GPVTG,0.00,T,,M,1.155,N,2.140,K,A*3A
$GPGGA,165212.000,0345.446176,N,10311.309491,E,1,4,2.11,244.074,M,-4.132,M,,*41
$GPGSA,A,3,30,01,22,31,,,,,,,,,2.33,2.11,1.00*03
$GPGSV,1,1,04,22,65,137,33,31,37,343,31,01,28,003,30,30,15,037,38*78
$GPRMC,165212.000,A,0345.446176,N,10311.309491,E,1.079,0.00,200108,,,A*59
$PMTKCHN,30382,01302,22332,31312,14001,29001,02001,11001,16001,28001,05001,20031,1803
1,06001,19001,07001,09001,25001,08001,13001,10001,03001,04001,27001,26001,15001,17001,120
01,00000,00000,00000,00000*49
$GPVTG,0.00,T,,M,1.079,N,2.000,K,A*30
```

6.3.4 Output shown from the circuit in the ISIS 7 Pro software

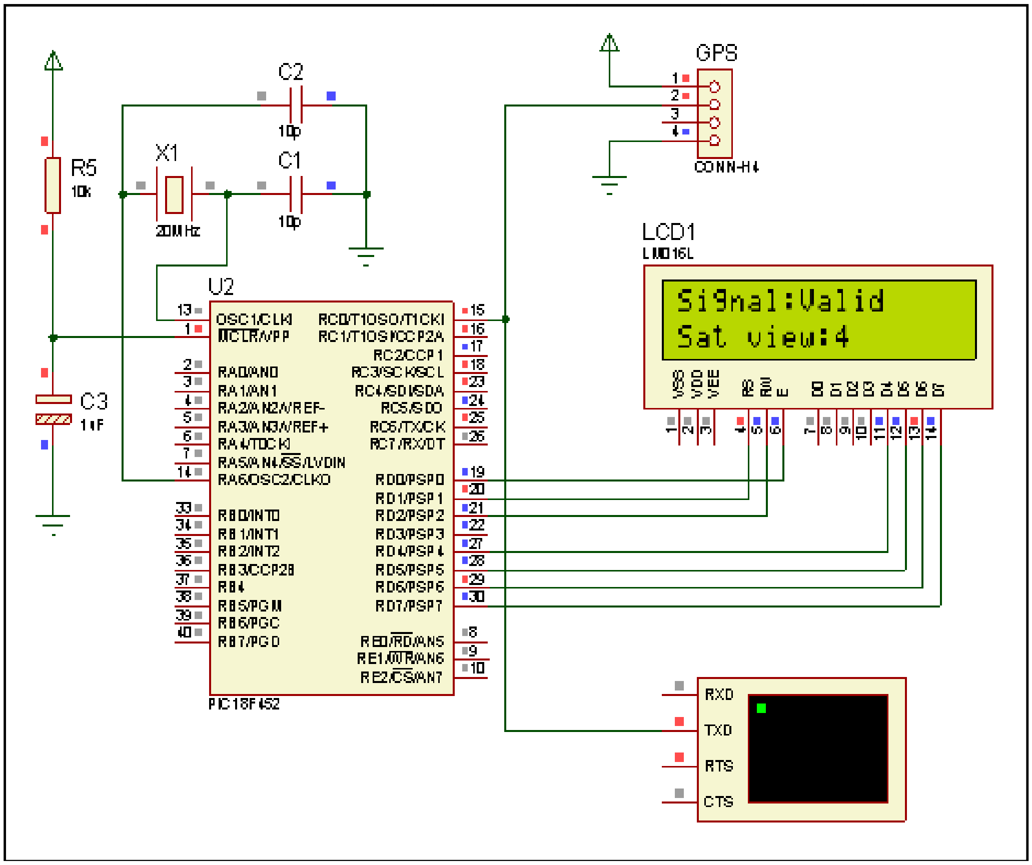


Figure 6.4 : First output from simulation

Figure 6.4 show that the first output for the simulation. It shows that the signal from Virtual Memory is valid and there is four satellites in coverage.

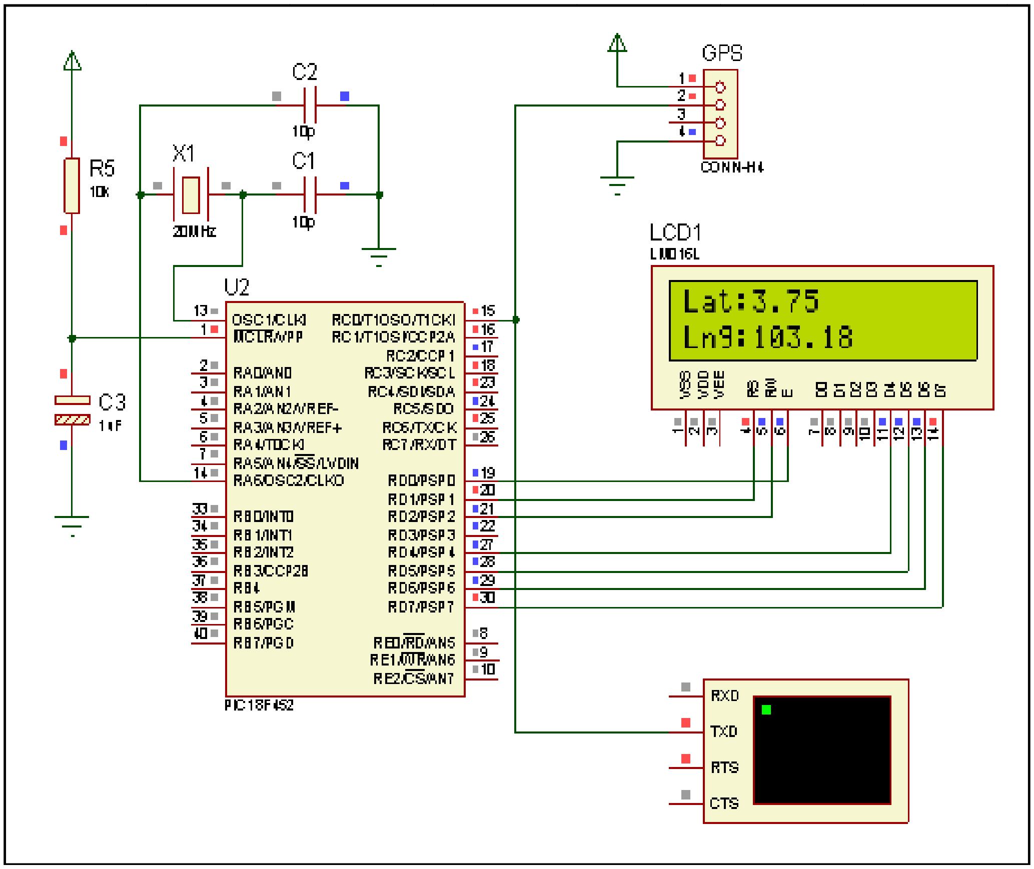


Figure 6.5 : Second output from simulation

Figure 6.5 show that the second output for the simulation. It shows that the latitude and longitude of the GPS receiver.

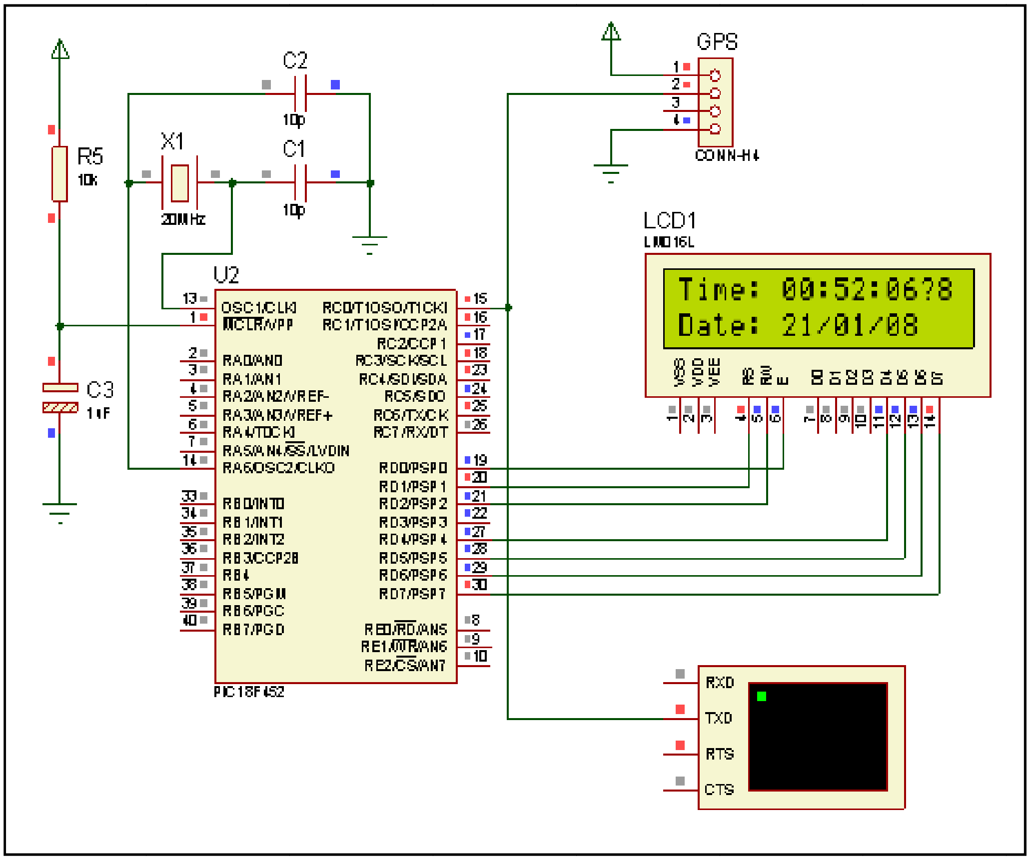


Figure 6.6 : Third output from simulation

Figure 6.6 show that the third output for the simulation. It shows that the real time and date.

6.4 Result on hardware development

According to the whole circuit, all the module had been assembled. The result of the real hardware is same like the simulation. Figure 6.8 show that the real hardware connection.

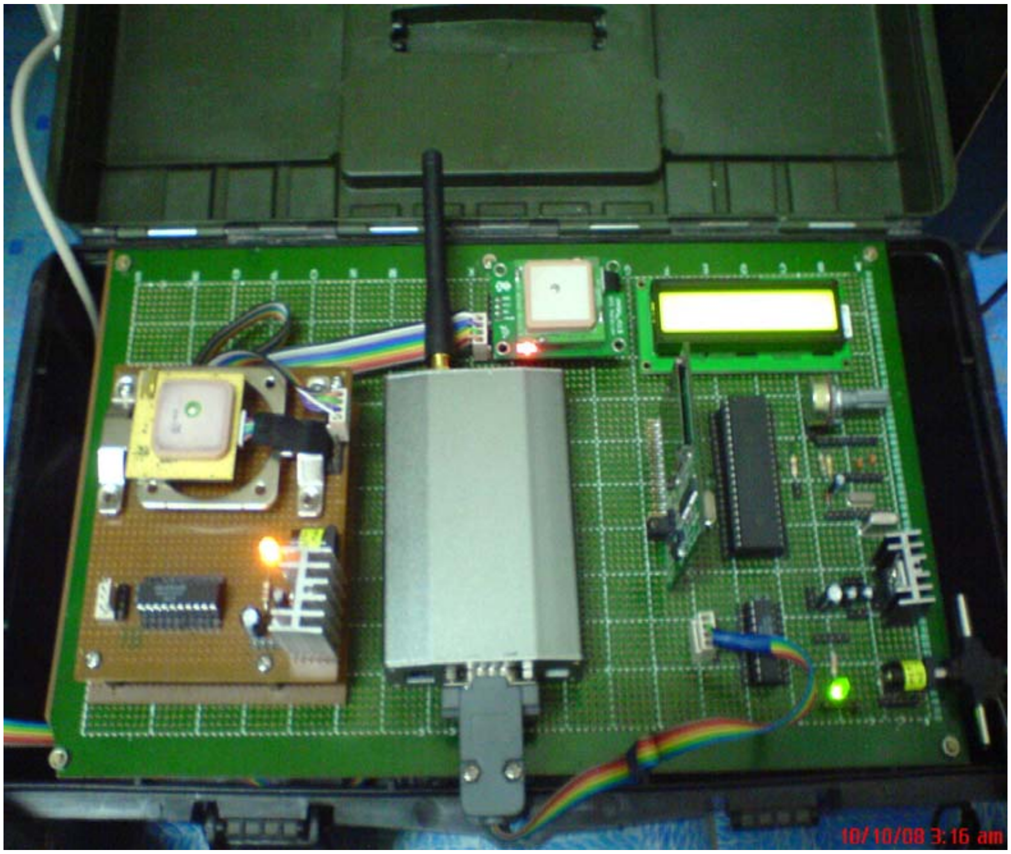


Figure 6.7 : Hardware Design

According to the hardware development, the result are listed below :

i. Result from LCD Module.

Figure 6.8 show the result of LCD Module while the system is running. The result is same as simulation.

Figure 6.8 : LCD Module result

ii. Result from MMC Module

Figure 6.9 show that the file of GPS data on the MMC when open it from the window operating system. The data had been write successfully.

Figure 6.9 : MMC file output

iii. Result from GSM Modem

The system will send a message to the user that want to know the location of the hardware. The SMS will show the latitude and the latitude of the hardware. The programming of the system is show in Appendix A.

CHAPTER 7

CONCLUSION AND FUTURE RECOMMENDATION

7.1 Conclusion

Smart Vehicle Data Logger is a data logger project using GPS Receiver, GSM Modem and Multimedia Card. The Smart Vehicle Data Logger can track speed and location of the vehicle travel using Global Positioning System (GPS) and store the information of the GPS receiver. User is allowed to trace the location of the vehicle by SMS the Data Logger.

The system will send back the location of the vehicle through GSM module. The controller for this prototype design is microcontroller system using PIC 18F452 and programmed using PIC C Compiler (C Programming) software.

Due to the result from Smart Vehicle Data Logger prototype , we can know that kind of information below by using RMC Protocol;

- The real time and date.
- The coordinate
- Speed
- Satellites used

All the data had been shown in MMC Module and had been displayed at LCD Module . Since the project is based on microcontroller system, the development of software plays a major role in develop the system. The high level language (C Programming) is used to programming the MCU.

Last but not least, the system is considered achieved the main point of this project because it was fully functioning. The MCU can communicate with the other device in good condition. The program to control the system can follow the instruction given. The integration of the whole system is successfully working as the objective of the project.

7.1 Future Recommendation

Due to this project , there are a variety of enhancement that could be made to this project to make it more user friendly.

i. Using Double Layer PCB

By using the double layer PCB, the size of circuit board can be reduced. The product can be more smaller than this prototype in order to commercialize it.

ii. Brand of GPS Receiver

The brand of GPS receiver is important to this project. Either the receiver is take time to valid or less time to valid. The Parrallax GPS receiver takes about 10 minutes to 30 minutes to valid. This can interrupt the system stability.

7.3 Costing and Commercialization

The overall of the whole project is based on hardware development. As discussed in previous chapter, the hardware consist of five main module ; GSM Modem, GPS receiver, MultimediaCard , LCD Module and the microcontroller. Therefore the cost of the project nearly depends on type of electronic devices. Table 7.1 shows the list of the components and its price.

Table 7.1 : Price of component

No	Component	Price
1	GPS receiver	RM 780
2	GSM Modem	RM 500
3	MMC Holder	RM 200
4	Multimedia Card 1GB	RM 48
5	PIC18F452 Microcontroller	RM 40
6	LCD Module	RM 32
	Total	**RM 1600**

For this prototype, the estimate cost is about RM1600. The cost is considered compared to the GPS data logger that is available in market. The price will become much lower if the devices are bought in high volume. Since the product are highly on demand nowadays, it is highly recommended that the prototype to be upgraded so that it can be commercialized in near future.

REFERENCES

[1] Shadow Tracker® Expert
14 March 2008
http://www.advantrack.com/hw_Shadow_Tracker_Expert.htm

[2] CarChip Pro Automotive Data Logger
25 Mei 2008
http://www.microdaq.com/davis/automotive/carchip_logger.php

[3] PIC18F452 Microchip Microcontroller
12 Jun 2008
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010296

[4] Global Positioning System
Title Global positioning systems, inertial navigation, and integration /
Author Grewal, Mohinder S.

[5] Multimedia Card
MMC-RS Product manual1.3
Author SanDisk Corporation

[6] GSM Module
TMAS GSM/GPRS Terminal Hardware User Manual Ver 3.0
Author TCAM Technology Pte Ltd

APPENDIX A

MICRO CONTROLLER SOURCE CODE

```
////////////////////////////////////////////////////////////////////
//    SMART VEHICLE DATA LOGGER V1
//
//------------------------------------------------------------------
//    Details: 1. Data from GPS (GPRMC) will be store into MMC card
//
//             2. When recieve a phone call from user, data from GPS
//
//                will be send back to user number.
//
//
//
//    Module : 1. GPS reciever ( ETEK )
//
//             2. MMC Development kit
//
//             3. GSM Tx/Rx Modem
//
//
//
////////////////////////////////////////////////////////////////////

//Microcontroller Configuration
#include <18F452.h>
#include <stdlib.h>
#fuses hs, nowdt, nolvp, noprotect
#use delay(clock=20000000)

#use rs232(baud=4800,xmit=PIN_C1,rcv=PIN_C0,PARITY=N, stream= GPS)
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7,stream=GSM)

//Multimedia Card Configuration
#define MMC_CLK   PIN_C3
#define MMC_DI    PIN_C4
#define MMC_DO    PIN_C5
#define MMC_CS    PIN_C2
#DEFINE _CS PIN_C2
#DEFINE MAX_FILES 2
#DEFINE MMC_BUFF_SIZE 32
#include<mmcdos.c>

//LCD Configuration
#include<lcd.c>
#include<math.h>

const unsigned int Mon[] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
int year, month, day;
```

```
char timed_getc(){

   long timeout=0;

   do{

      timeout++;
   }while(!kbhit() && timeout<50000); //500m second

   if(kbhit()){
      return(getc());
   }else{
      return('?');
   }
}



void MakeSplit(int32 data ,char *p)
{
   int a,b,c,l,m;
   int32 d;

   c = (int)(data - floor(data/100)*100);
   b = (int)(floor(data/100) - floor(data/10000)*100);
   a = (int)(floor(data/10000));

   d = (int32)(a)*10000 + (int32)(b)*100 + (int32)(c);
   itoa(d,10,p);


   for(l=0;l<6;l++)
   {
      if(p[l] == '\0')
      {
          for(m=5;m>0;m--)
          {
            p[m] = p[m-1];
          }
          p[0] = '0';
      }
   }


}



void main(){

   char c,i,j,g;
   char ring[10] = {"RING"};
   int p;
   int8 buf[80],term[3],*ptr;
   int8 _code[7];
   int8 _time[11];
   int8 _latitude[12];
   int8 _longitude[13];
   int8 _dLat[2];
   int8 _dLong[2];
   int8 _valid[2];
   int8 _sv[3];
   int8 _day[3];
   int8 _month[3];
   int8 _year[3];
```

```
   float _lat,_long;
   int _gmt;
   int32 _timefull;
   int32 _datefull;
   int8 _dfull[7];




   i=0;
   _gmt = 0;


   set_tris_c(0b10000000);

   lcd_init();
   lcd_putc("STARTING...");
   delay_ms(1000);
   lcd_putc("\f");
   lcd_putc(" SMART VEHICLE \n  DATA LOGGER");
   delay_ms(1000);
   lcd_putc("\f");
   lcd_putc(" Autorized by \n  Amir Haris");
   delay_ms(1000);
   lcd_putc("\f");
   lcd_putc(" Supervisor : \n  Mohd Zamri");
   delay_ms(1000);
    lcd_putc("\f");
   lcd_putc(" SMART VEHICLE \n  DATA LOGGER");




   do{

      c = timed_getc();
      if(c == '$')
      {
         do{
         c = buf[i++] = timed_getc();

            if((c == ',') && (buf[i-2] == ',') && (i>2))
            {
               buf[i] = ',';
               buf[i-1] = '0';
               i++;
            }

         }while(c != '*' && i < 80);
         i=0;


//Multimedia card Write Operation

   setup_adc_ports(NO_ANALOGS);
   setup_adc(ADC_OFF);
   setup_psp(PSP_DISABLED);
   SETUP_SPI(SPI_MASTER | SPI_CLK_DIV_16 | SPI_H_TO_L
|SPI_XMIT_L_TO_H );
   setup_wdt(WDT_OFF);
   setup_timer_0(RTCC_INTERNAL);
   setup_timer_1(T1_DISABLED);
   setup_timer_2(T2_DISABLED,0,1);
   setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
   setup_oscillator(True);

   set_tris_c(0b10010011);
   output_high(MMC_CS);
   p=init_mmc(10);
```

```
   SET_BLOCKLEN( 512);

//------------------
      MMC_dir_protected=false;
      write_BLOCK( 0, buf, 512);
//------------------


//GPS Data Seperation

         strcpy(term,",;");
         ptr = strtok(buf, term);

         strcpy(_code,"GPRMC");

         if(!strcmp(ptr,_code)) //if msg header = "GPRMC"
         {

            ptr = strtok(0,term);    //time
            strcopy(_time,ptr);
            _timefull = atoi32(_time);

            ptr = strtok(0,term);    //data validity
            strcopy(_valid,ptr);

            ptr = strtok(0,term);    //latitude
            strcopy(_latitude,ptr);

            ptr = strtok(0,term);    //latitude sector
            strcopy(_dLat,ptr);

            ptr = strtok(0,term);    //longitude
            strcopy(_longitude,ptr);

            ptr = strtok(0,term);    //longitude sector
            strcopy(_dLong,ptr);

            ptr = strtok(0,term);    //speed over ground
            ptr = strtok(0,term);    //course over ground

            ptr = strtok(0,term);    //date
            _day[0] = ptr[0];
            _day[1] = ptr[1];
            _month[0] = ptr[2];
            _month[1] = ptr[3];
            _year[0] = ptr[4];
            _year[1] = ptr[5];

           day = atoi(_day);
           month = atoi(_month);
           year = atoi(_year);

           _lat = (atol(_latitude) / 100)*100 ;
           _lat = (atof(_latitude) - (float)_lat)/60;
           _lat = (float)(atol(_latitude) / 100) + _lat;
           if(_dLat[0] == 'S')
           _lat *= -1;

           _long = (atol(_longitude) / 100)*100 ;
           _long = (atof(_longitude) - (float)_long)/60;
           _long = (float)(atol(_longitude) / 100) + _long;
           if(_dLong[0] == 'W')
           _long *= -1;


         //Check validity of signal
         if(_valid[0] == 'A' )
         {
            printf(lcd_putc,"\fSignal:Valid");
```

```
            _gmt = ceil((_long+7.5)/15);                //360deg
divide 24hrs = 15 time band
         }
         else
            printf(lcd_putc,"\fSignal:Not Valid");

         printf(lcd_putc,"\nSat view:%s",_sv);


         delay_ms(1500);

         //Latitude & longitude
         printf(lcd_putc,"\fLat:%f",_lat);
         printf(lcd_putc,"\nLng:%f",_long);

         delay_ms(1500);

         //time & date local time adjustment
         if(_gmt != 0)
         {
            _timefull += (int32)_gmt*10000;
            if(_timefull >= 240000)
            {
               _timefull -= 240000;

               //date adjustment
               if(((year%4==0 && year%100 !=0)||year%400==0) &&
month == 2)
               i = 1;  // <- leap year offset for February
               day++;

               if(day > (Mon[month]+i))
               {
                  day -= (Mon[month]+i);
                  month++;
               }
               i=0;
               if(month > 12)
               {
                  month -= 12;
                  year++;
               }
            }
         }

         _datefull = (int32)(day)*10000 + (int32)(month)*100 +
(int32)(year);


         //time & date
         MakeSplit(_timefull,_dfull);
         printf(lcd_putc,"\fTime:
%c%c:%c%c:%c%c?%d",_dfull[0],_dfull[1],_dfull[2],_dfull[3],_dfull[4]
,_dfull[5],_gmt);
         MakeSplit(_datefull,_dfull);
         printf(lcd_putc,"\nDate:
%c%c/%c%c/%c%c",_dfull[0],_dfull[1],_dfull[2],_dfull[3],_dfull[4],_d
full[5]);

         delay_ms(1500);




//GSM Configuration

    g = fgetc(GSM);

    if(!strcmp(g,ring))
    {
```

```
        fprintf(GSM,"\f");
        delay_ms(3000);
        fprintf(GSM,"AT+CMGF=1 \r");
        delay_ms(3000);
        fprintf(GSM,"AT+CSCA=\"+60120000015\" \r");
        delay_ms(3000);



      //Sending text message SMS
        fprintf(GSM,"AT+CMGF=1 \r");
        delay_ms(3000);
        fprintf(GSM,"AT+CMGS=\"+60179507229\" \r");
        delay_ms(3000);
        fprintf(GSM,"This Message is from Smart Vehicle Data Logger.
Your car  Coordinate is : Latitude:%f and Longitude:%f  %c
",_lat,_long,0x1A);
    }

    else
    {

    }


   //Reading text message SMS
   //fprintf(GSM,"AT+CMGF=1\r");
   //delay_ms(1000);



       }   }

       strcpy(_code,"GPGGA");

       if(!strcmp(ptr,_code)) //if msg header = "GPGGA"
       {
            for(j=0;j<7;j++)
            ptr = strtok(0,term);
            strcopy(_sv,ptr);
       }


       }






   while(1);

}
```

APPENDIX B

PIC 18F452 DATASHEET

MICROCHIP

PIC18FXX2
Data Sheet

High-Performance, Enhanced Flash
Microcontrollers with 10-Bit A/D

28/40-pin High Performance, Enhanced FLASH Microcontrollers with 10-Bit A/D

High Performance RISC CPU:

- C compiler optimized architecture/instruction set
  - Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 32 Kbytes
- Linear data memory addressing to 1.5 Kbytes

Device	On-Chip Program Memory		On-Chip RAM (bytes)	Data EEPROM (bytes)
	FLASH (bytes)	# Single Word Instructions		
PIC18F242	16K	8192	768	256
PIC18F252	32K	16384	1536	256
PIC18F442	16K	8192	768	256
PIC18F452	32K	16384	1536	256

- Up to 10 MIPs operation:
  - DC - 40 MHz osc./clock input
  - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
  - Capture input: capture is 16-bit, max. resolution 6.25 ns ($T_{CY}/16$)
  - Compare is 16-bit, max. resolution 100 ns (T_{CY})
  - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz; 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
  - 3-wire SPI™ (supports all 4 SPI modes)
  - I^2C™ Master and Slave mode

Peripheral Features (Continued):

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

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
  - Fast sampling rate
  - Conversion available during SLEEP
  - Linearity ≤ 1 LSb
- Programmable Low Voltage Detection (PLVD)
  - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Special Microcontroller Features:

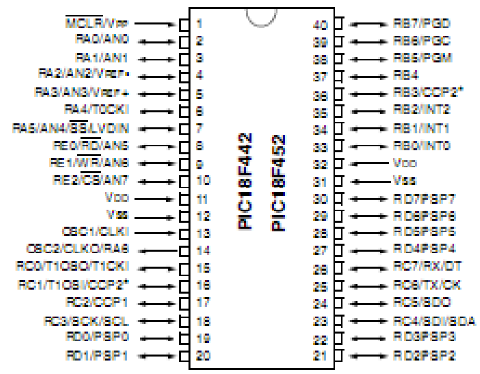
- 100,000 erase/write cycle Enhanced FLASH program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory
- FLASH/Data EEPROM Retention: > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options including:
  - 4X Phase Lock Loop (of primary oscillator)
  - Secondary Oscillator (32 kHz) clock input
- Single supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low power, high speed FLASH/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption:
  - < 1.6 mA typical @ 5V, 4 MHz
  - 25 µA typical @ 3V, 32 kHz
  - < 0.2 µA typical standby current

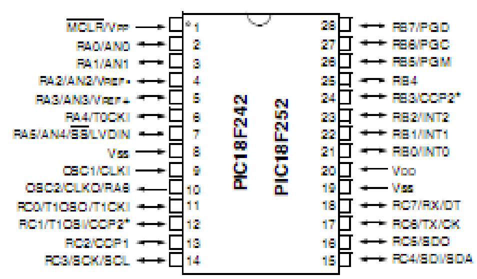
Pin Diagrams (Cont.'d)

DIP



Note: Pin compatible with 40-pin PIC16C7X devices.

DIP, SOIC



*RB3 is the alternate pin for the CCP2 pin multiplexing.

4.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. Figure 4-6 and Figure 4-7 show the data memory organization for the PIC18FXX2 devices.

The data memory map is divided into as many as 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper 4 bits for the BSR are not implemented.

The data memory contains Special Function Registers (SFR) and General Purpose Registers (GPR). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratch pad operations in the user's application. The SFRs start at the last location of Bank 15 (0xFFF) and extend downwards. Any remaining space beyond the SFRs in the Bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as '0's.

The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of a File Select Register (FSRn) and a corresponding Indirect File Operand (INDFn). Each FSR holds a 12-bit address value that can be used to access any location in the Data Memory map without banking.

The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction that moves a value from one register to another.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM. Section 4.10 provides a detailed description of the Access RAM.

4.9.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly. Indirect addressing operates using a File Select Register and corresponding Indirect File Operand. The operation of indirect addressing is shown in Section 4.12.

Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other RESETS.

Data RAM is available for use as GPR registers by all instructions. The top half of Bank 15 (0xF80 to 0xFFF) contains SFRs. All other banks of data memory contain GPR registers, starting with Bank 0.

4.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 4-1 and Table 4-2.

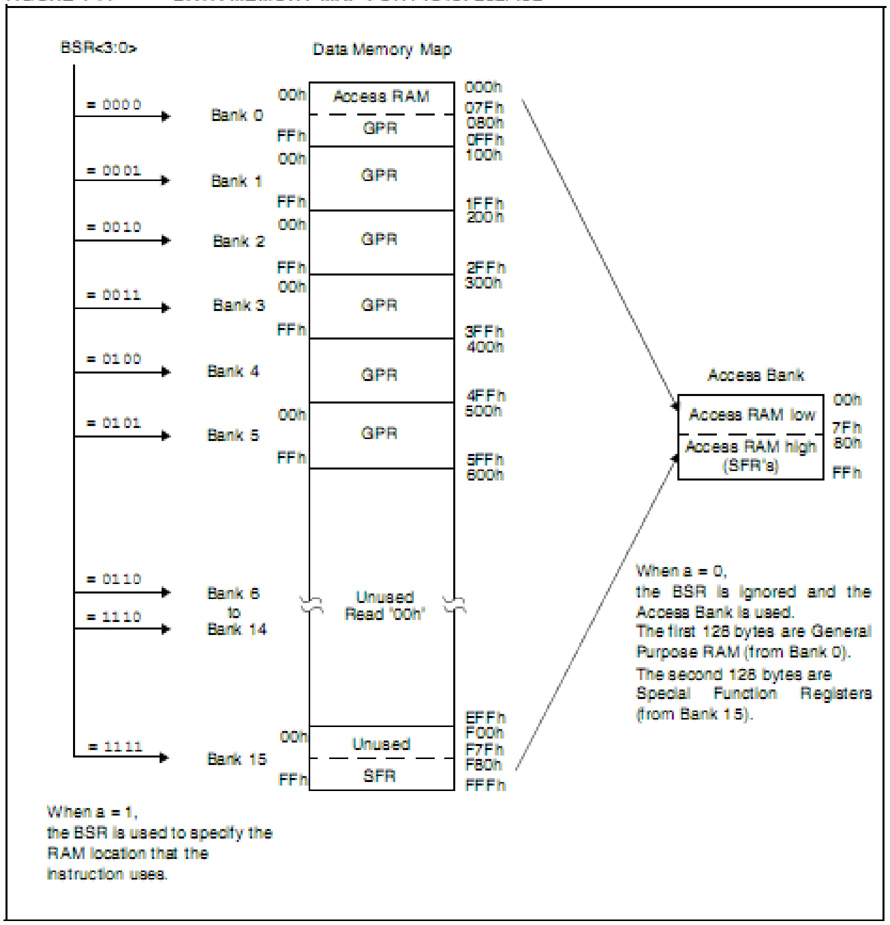
The SFRs can be classified into two sets; those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature.

The SFRs are typically distributed among the peripherals whose functions they control.

The unused SFR locations will be unimplemented and read as '0's. See Table 4-1 for addresses for the SFRs.

PIC18FXX2

FIGURE 4-7: DATA MEMORY MAP FOR PIC18F252/452



5.0 FLASH PROGRAM MEMORY

The FLASH Program Memory is readable, writable, and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

5.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

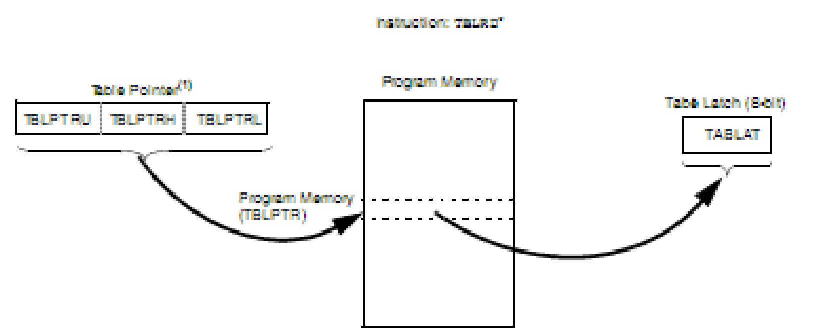
The program memory space is 16-bits wide, while the data RAM space is 8-bits wide. Table Reads and Table Writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table Read operations retrieve data from program memory and places it into the data RAM space. Figure 5-1 shows the operation of a Table Read with program memory and data RAM.

Table Write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in Section 5.5, "Writing to FLASH Program Memory". Figure 5-2 shows the operation of a Table Write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a Table Write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 5-1: TABLE READ OPERATION



Note 1: Table Pointer points to a byte in program memory.

5.5 Writing to FLASH Program Memory

The minimum programming block is 4 words or 8 bytes. Word or byte programming is not supported.

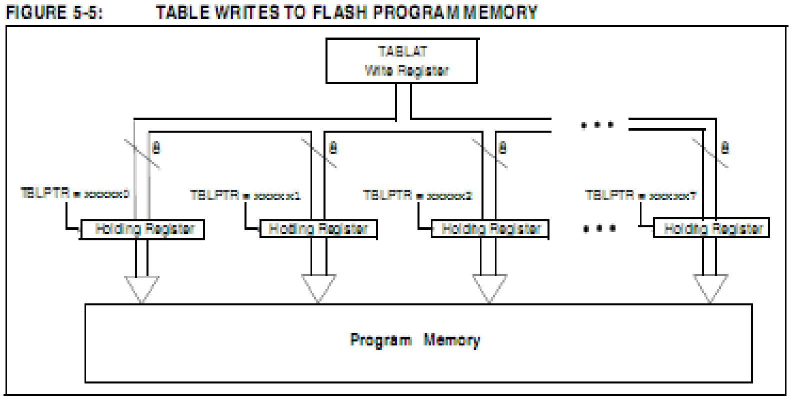
Table Writes are used internally to load the holding registers needed to program the FLASH memory. There are 8 holding registers used by the Table Writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the `TBLWT` instruction has to be executed 8 times for each programming operation. All of the Table Write operations will essentially be short writes, because only the holding registers are written. At the end of updating 8 registers, the EECON1 register must be written to, to start the programming operation with a long write.

The long write is necessary for programming the internal FLASH. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

FIGURE 5-5: TABLE WRITES TO FLASH PROGRAM MEMORY



5.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer with address being erased.
4. Do the row erase procedure.
5. Load Table Pointer with address of first byte being written.
6. Write the first 8 bytes into the holding registers with auto-increment (`TBLWT*+` or `TBLWT+*`).
7. Set EEPGD bit to point to program memory, clear the CFGS bit to access program memory, and set WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Repeat steps 6-14 seven times, to write 64 bytes.
15. Verify the memory (Table Read).

This procedure will require about 18 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 5-3.

Note:	Before setting the WR bit, the table pointer address needs to be within the intended address range of the 8 bytes in the holding registers.

APPENDIX C

GPS PROTOCOL

GPS Protocol Reference Manual

P/N: 980-0330-A
Revision 1.30

We Make Dreams a Reality

Leadtek Research Inc.

Chapter 1 NMEA Input/Output Messages

The unit may also output data in NMEA -0183 format as defined by the National Marine Electronics Association (NMEA), Standard For Interfacing Marine Electronics Devices, Version 2.20, January 1, 1997.

1.1 NMEA Output Messages

The unit outputs the following messages as shown below (Table 1-1):

Table 1-1 NMEA-0183 Output Messages

NMEA Record	Description
GGA	Global positioning system fixed data
GLL	Geographic position – latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

1.1.1 GGA – Global Positioning System Fixed Data

Table 1-2 contains the values for the following example:

$GPGGA, 161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,,,0000*18

Table 1-2 GGA Data Format

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-3
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude[1]	9.0	meters	
Units	M	meters	
Geoid Separation[1]		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

[1] SiRF does not support geoid correction. Values are WGS-84 ellipsoid heights.

Table 1-3 Position Fix Indicator

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3	GPS PPS Mode, fix valid

1.1.2 GLL – Geographic Position – Latitude/Longitude

Table 1-4 contains the values for the following example:

$GPGLL,3723.2475,N,12158.3416,W,161229.487,A*2C

Table 1-4 GLL Data Format

Name	Example	Units	Description
Message ID	$GPGLL		GLL protocol header
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
UTC Position	161229.487		hhmmss.sss
Status	A		A=data balid or V=data not valid
Checksum	*2C		
<CR><LF>			End of message termination

1.1.3 GSA – GNSS DOP and Active Satellites

Table 1-5 contains the values for the following example:

$GPGSA,A,3,07,02,26,27,09,04,15,,,,,,1.8,1.0,1.5*33

Table 1-5 GSA Data Format

Name	Example	Units	Description
Message ID	$GPGSA		GSA protocol header
Mode 1	A		See Table 1-6
Mode 2	3		See Table 1-7
Satellite Used[1]	07		Sv on Channel 1
Satellite Unsed[1]	02		Sv on Channel 2
….			….
Satellite Unsed[1]			Sv on Channel 12
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		Horizontal Dilution of Precision
VDOP	1.5		Vertical Dilution of Precision
Checksum	*33		
<CR> <LF>			End of message termination

[1] Satellite used in solution.

Table 1-6 Mode 1

Value	Description
1	Fix not available
2	2D
3	3D

Table 1-7 Mode 2

Value	Description
M	Manual --- forced to operate in 2D or 3D mode
A	Automatic – allowed to automatically switch 2D/3D

1.1.4 GSV- GNSS Satellites in View

Table 1-8 contains the values for the following example:

$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42*71

$GPGSV2,2,07,09,23,313,42,04,19,159,41,15,12,041,42*41

Table 1-8 GGA Data Format

Name	Example	Units	Description
Message ID	$GPGSV		GSV protocol header
Number of Messages [1]	2		Range 1 to 3
Message Number[1]	1		Range 1 to 3
Satellites in View	07		
Satellite ID	07		Channel 1 (Range 1 to 32)
Elevation	79	degrees	Channel 1 (Maximum 90)
Azimuth	048	degrees	Channel 1 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
….			….
Satellite ID	27		Channel 4 (Range 1 to 32)
Elevation	27	degrees	Channel 4 (Maximum 90)
Azimuth	138	degrees	Channel 4 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
Checksum	*71		
<CR> <LF>			End of message termination

1.1.5 RMC- Recommended Minimum Specific GNSS Data

Table 1-9 contains the values for the following example:

$GPRMC, 161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10

Table 1-9 RMC Data Format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Position	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	Knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation[1]		degrees	E=east or W=west
Checksum	*10		
<CR> <LF>			End of message termination

[1] SiRF does not support magnetic declination. All "course over ground" data are geodetic

1.1.6 VTG- Course Over Ground and Ground Speed

Table 1-10 contains the values for the following example:

$GPVTG, 309.62, T,,M,0.13,N,0.2,K*6E

Table 1-10 VTG Data Format

Name	Example	Units	Description
Message ID	$GPVTG		VTG protocol header
Course	309.62	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic[1]
Speed	0.13	knots	Measured horizontal speed
Units	N		Knots
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometer per hour
Checksum	*6E		
<CR> <LF>			End of message termination

[1] SiRF does not support magnetic declination. All "course over ground" data are geodetic WGS-84 directions.

APPENDIX D

SANDISK MULTIMEDIA CARD USER MANUAL

SanDisk MultiMediaCard and
Reduced-Size MultiMediaCard

Product Manual
Version 1.3
Document No. 80-36-00320
April 2005

SanDisk Corporation
Corporate Headquarters • 140 Caspian Court • Sunnyvale, CA 94089
Phone (408) 542-0500 • Fax (408) 542-0503
www.sandisk.com

Figure 1-2 Memory Array Partitioning

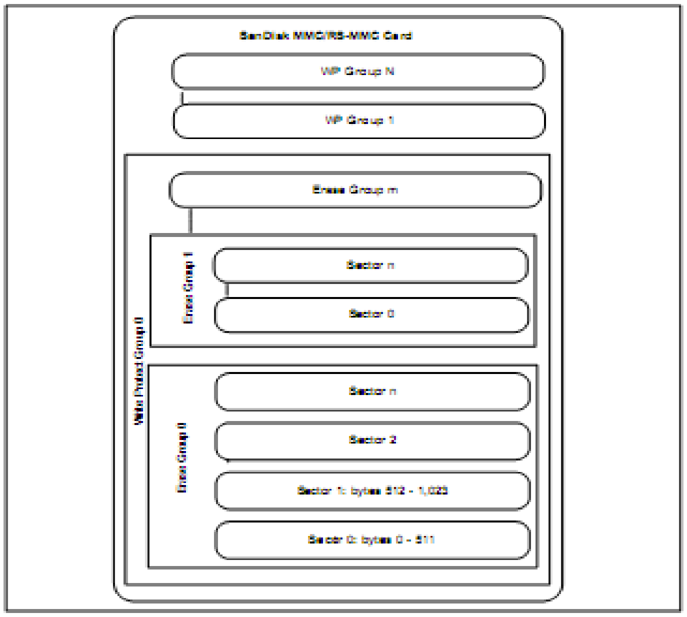


Table 1-1 Memory Array Structures Summary[1]

	Bytes	Sector	Erase Group Size	No. of Erase	WP Group	No. of Write
SDMJ-32 SDMRJ-32	32 MB	62,688	32	1,959	32	62
SDMJ-64 SDMRJ-64	64 MB	125,408	32	3,919	32	123
SDMJ-128 SDMRJ-128	128 MB	250,816	32	7,838	32	245
SDMJ-256 SDMRJ-256	256 MB	501,632	32	15,676	32	490
SDMJ-512 SDMRJ-512	512 MB	1,003,264	32	31,352	32	980
SDMJ-1024 SDMRJ-1024	1 GB	2,006,528	32	62,704	32	1,960

[1] All measurements are units-per-card.

1.12.6 Read and Write Operations

The MultiMediaCard/RS-MultiMediaCard supports two read/write modes as shown in Figure 1-3 and defined in Table 1-2.

Figure 1-3 Data Transfer Formats

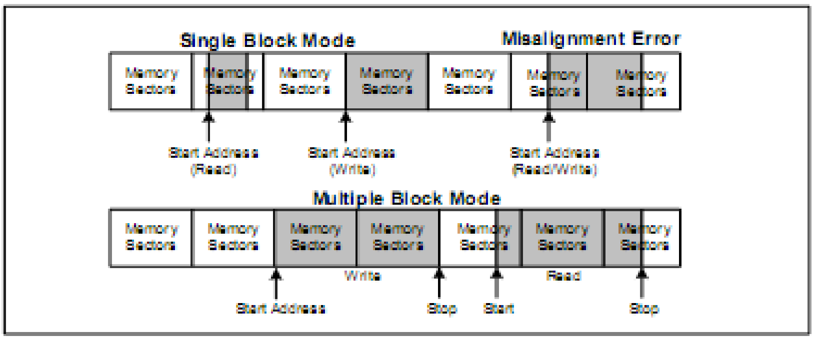


Table 1-2 Mode Definitions

Mode	Description
Single Block	In this mode the host reads or writes one data block in a pre-specified length. The data block transmission is protected with 16-bit CRC that is generated by the sending unit and checked by the receiving unit. The block length for read operations is limited by the device sector size (512 bytes) but can be as small as a single byte. Misalignment is not allowed. Every data block must be contained in a single physical sector. The block length for write operations must be identical to the sector size and the start address aligned to a sector boundary.
Multiple Block	This mode is similar to the single block mode, except for the host can read/write multiple data blocks (all have the same length) that are stored or retrieved from contiguous memory addresses starting at the address specified in the command. The operation is terminated with a stop transmission command. Misalignment and block length restrictions apply to multiple blocks and are identical to the single block read/write operations.

1.12.7 Data Protection in the Flash Card

Every sector is protected with an error correction code. The ECC is generated (in the memory card) when the sectors are written and validated when the data is read. If defects are found, the data is corrected prior to transmission to the host.

1.13 SPI Mode

The SPI mode is a secondary communication protocol for the MultiMediaCard and RS-MultiMediaCard. This mode is a subset of the MultiMediaCard Protocol, designed to communicate with an SPI channel, commonly found in Motorola and other vendors' microcontrollers.

Table 1-4 SPI Mode

Function	Description
Negotiating Operating Conditions	The operating condition negotiation function of the MultiMediaCard/RS-MMC bus is not supported in SPI Mode. The host must work within the valid voltage range, 2.7 to 3.6 V, of the card.
Card Acquisition and Identification	This function is not supported in SPI Mode. The host must know the number of cards currently connected on the bus. Specific card selection is done using the CS signal.
Card Status	In SPI mode, only 16 bits containing errors relevant to SPI mode can be read out of the 32-bit Status Register.
Memory Array Partitioning	Memory partitioning in SPI mode is equivalent to MultiMediaCard mode. All read and write commands are byte addressable.
Read/Write Operations	In SPI mode, single and multiple block data transfers are supported. Stream mode is not supported.
Data Transfer Rate	Same as MultiMediaCard mode.
Data Protection in MultiMediaCard/RS-MultiMediaCard	Same as MultiMediaCard mode.
Erase	Same as MultiMediaCard mode.
Write Protection	Same as MultiMediaCard mode.

3 Interface Description

3.1 Physical Description

The MultiMediaCard and RS-MultiMediaCard has seven exposed contacts on one side. The host is connected to the card using a dedicated seven-pin connector.

3.1.1 Pin Assignments

Table 3-1 MultiMediaCard and RS-MultiMediaCard Pad Assignment

Pin No.	Name	Type[1]	Description
MultiMediaCard Mode			
1	RSV	NC	Not connected or Always "1"
2	CMD	I/O, PP, OD	Command/Response
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DAT0	I/O, PP	Data 0
SPI Mode			
1	CS	I	Chip Select (active low)
2	DataIn	I	Host-to-card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card-to-host Data and Status

3.2 MultiMediaCard/RS-MultiMediaCard Bus Topology

The MultiMediaCard/RS-MultiMediaCard bus has three communication lines and four supply lines.

- CMD
- DAT
- CLK
- VDD
- VSS[1:2]

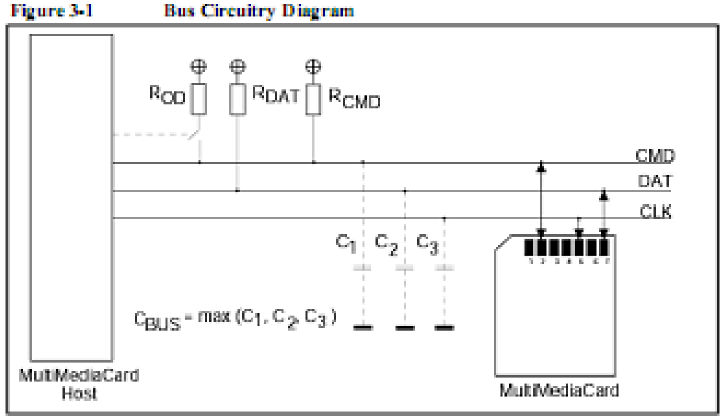
The description of each signal is contained in Table 3-2.

Table 3-2 Bus Signal Descriptions

Name	Description
CMD	Command is a bi-directional signal. Host and card drivers are operating in two modes: open drain and push-pull.
DAT	Data line is a bi-directional signal. Host and card drivers are operating in push-pull mode.
CLK	Clock is a host to card signal. CLK operates in push-pull mode.
VDD	VDD is the power supply line for all cards.
VSS [1:2]	VSS are two ground lines.

Figure 3-1 shows the bus circuitry with one host in MultiMediaCard mode.

Figure 3-1 Bus Circuitry Diagram



The R_{OD} is switched on and off by the host synchronously to the open-drain and push-pull mode transitions. R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and DAT line against bus floating when no card is inserted or all card drivers are in a hi-impedance mode.

A constant current source can replace the R_{OD} in order to achieve better performance (constant slopes for the signal rising and falling edges). If the host does not allow the switchable R_{OD} implementation, a fixed R_{CMD} can be used. Consequently the maximum operating frequency in the open-drain mode has to be reduced in this case.

APPENDIX E

MAX 233 DATASHEET

19-4323; Rev 14; 8/04

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220–MAX249

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5μW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Next-Generation Device Features

- **For Low-Voltage, Integrated ESD Applications MAX3222E/MAX3232E/MAX3237E/MAX3241E/ MAX3246E: +3.0V to +5.5V, Low-Power, Up to 1Mbps, True RS-232 Transceivers Using Four 0.1μF External Capacitors (MAX3246E Available in a UCSP™ Package)**
- **For Low-Cost Applications MAX221E: ±15kV ESD-Protected, +5V, 1μA, Single RS-232 Transceiver with AutoShutdown™**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.

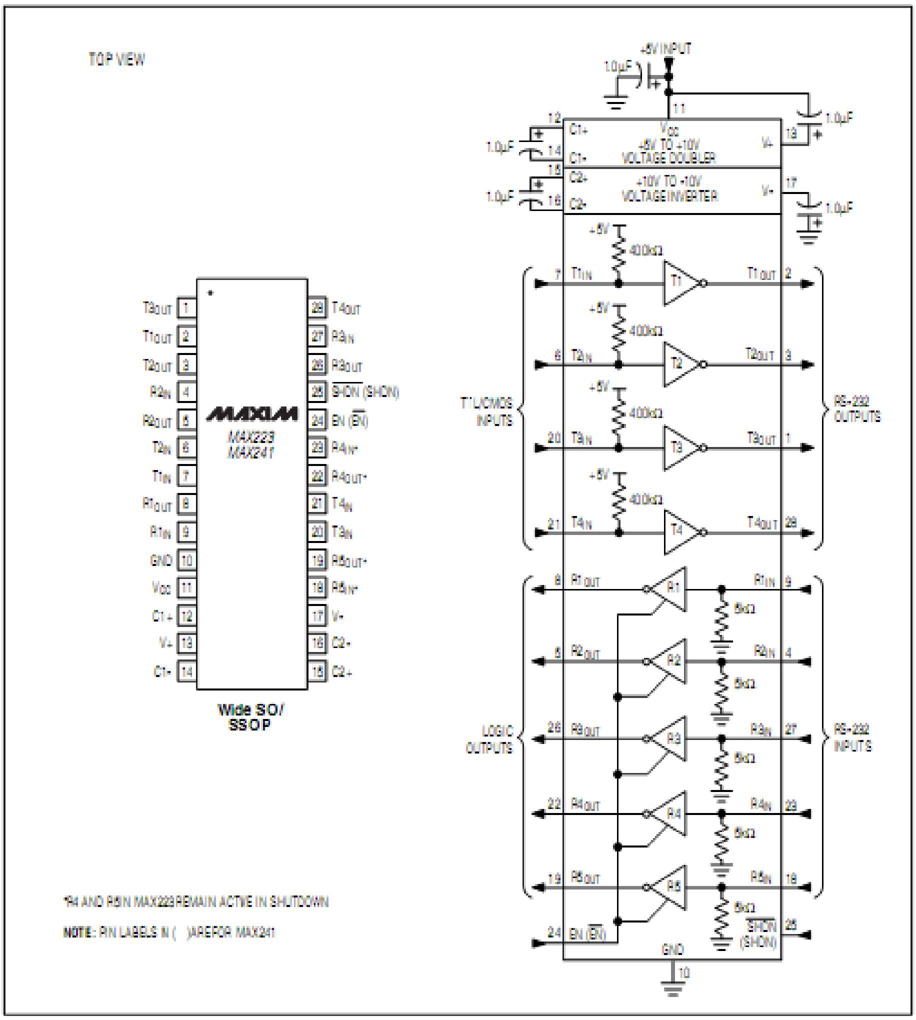
**Contact factory for dice specifications.*

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at


TOP VIEW
MAX223
MAX241
Wide SO/ SSOP
+5V INPUT
+5V TO +10V VOLTAGE DOUBLER
+10V TO -10V VOLTAGE INVERTER
TTL/CMOS INPUTS
RS-232 OUTPUTS
LOGIC OUTPUTS
RS-232 INPUTS
GND
*R4 AND R5 IN MAX223 REMAIN ACTIVE IN SHUTDOWN
NOTE: PIN LABELS IN ( ) ARE FOR MAX241

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		0.8	1.2		V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)		0.6	1.5		
RS-232 Input Threshold High	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)			1.7	2.4	V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)			1.5	2.4	
RS-232 Input Hysteresis	V_{CC} = 5V, no hysteresis in shutdown			0.2	0.5	1.0	V
RS-232 Input Resistance	T_A = +25°C, V_{CC} = 5V			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 1.6mA (MAX231/232/233, I_{OUT} = 3.2mA)					0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1mA			3.5	V_{CC} -0.4		V
TTL/CMOS Output Leakage Current	0V ≤ R_{OUT} ≤ V_{CC}; EN = 0V (MAX223); $\overline{EN}$ = V_{CC} (MAX235–241)				0.05	±10	µA
Receiver Output Enable Time	Normal operation	MAX223			600		ns
		MAX235/236/239/240/241			400		
Receiver Output Disable Time	Normal operation	MAX223			900		ns
		MAX235/236/239/240/241			250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, C_L = 150pF	Normal operation			0.5	10	µs
		$\overline{SHDN}$ = 0V (MAX223)	t_{PHLS}		4	40	
			t_{PLHS}		6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V			3	5.1	30	V/µs
	MAX231/MAX232/MAX233, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V				4	30	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V			300			Ω
Transmitter Output Short-Circuit Current					±10		mA

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ....-0.3V to +6V
V+ (Note 1) ....(V_{CC} - 0.3V) to +14V
V- (Note 1) ....+0.3V to +14V
Input Voltages
  T_{IN} ....-0.3V to (V_{CC} - 0.3V)
  R_{IN} (Except MAX220) ....±30V
  R_{IN} (MAX220) ....±25V
  T_{OUT} (Except MAX220) (Note 2) ....±15V
  T_{OUT} (MAX220) ....±13.2V
Output Voltages
  T_{OUT} ....±15V
  R_{OUT} ....-0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND ....Continuous
Continuous Power Dissipation (T_A = +70°C)
  16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) ..842mW
  16-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ..889mW
  20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) ..440mW
  16-Pin Narrow SO (derate 8.70mW/°C above +70°C) ...696mW
  16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
  18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
  20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
  20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
  16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
  18-Pin CERDIP (derate 10.53mW/°C above +70°C) 842mW
Operating Temperature Ranges
  MAX2_ _AC_ _, MAX2_ _C_ _ ....0°C to +70°C
  MAX2_ _AE_ _, MAX2_ _E_ _ ....-40°C to +85°C
  MAX2_ _AM_ _, MAX2_ _M_ _ ....-55°C to +125°C
Storage Temperature Range ....-65°C to +160°C
Lead Temperature (soldering, 10s) (Note 3) ....+300°C

Note 1: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.
Note 2: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.
Note 3: Maximum reflow temperature for the MAX225_WI and MAX233A_WP is +220°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V_{CC} = 5.0V		2.4			
Logic Pullup/Input Current	All except MAX220, normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/MAX242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/MAX242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V	V_{OUT} = ±15V		±0.01	±10	
		MAX220, V_{OUT} = ±12V			±25	
Data Rate				200	116	kbps
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V	V_{OUT} = 0V	±7	±22		mA
		MAX220			±60	
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
		MAX220			±25	
RS-232 Input Threshold Low	V_{CC} = 5V	All except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 4)	-3			
RS-232 Input Threshold High	V_{CC} = 5V	All except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 4)		-0.5	-0.1	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, V_{CC} = 5V, no hysteresis in SHDN		0.2	0.5	1	V
	MAX220			0.3		
	MAX243			1		
RS-232 Input Resistance	T_A = +25°C (MAX220)		3	5	7	KΩ
			3	5	7	
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
	I_{OUT} = 1.6mA (MAX220)				0.4	
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), figures 5, 6, 11, 19	No load	MAX220		0.5	2	µA
		MAX222/MAX232A/MAX233A/MAX242/MAX243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/MAX232A/MAX233A/MAX242/MAX243		15		
Shutdown Supply Current	MAX222/MAX242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/MAX242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/MAX242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V	MAX222/MAX232A/MAX233/MAX242/MAX243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t_{PHLT}	MAX222/MAX232A/MAX233/MAX242/MAX243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}	MAX222/MAX232A/MAX233/MAX242/MAX243		1.5	3.5	
		MAX220		5	10	

MAX220–MAX249