SMART BUS LOCALE ASCERTAIN TECHNOLOGY

By

ASIF SARVODHIN Reg No: 0710107011

P GOPALAKRISHNAN Reg No: 0710107031

of

KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE - 641 049.
(An Autonomous Institution affiliated to Anna University of Technology, Coir

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree

of

BACHELOR OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING

APRIL 2011

BONAFIDE CERTIFICATE

This is to be certified that this project report "**SMART BUS LOCALE ASCERTAIN TECHNOLOGY**" is the bonafied work of "**ASIF SARVODHIN, P.GOPALAKRISHNAN**" who carried out the project work under my supervision.

SIGNATURE

Dr.(Mrs).Rajeswari Mariappan

Head of the Department

Department of Electronics and

Communication Engineering,

Kumaraguru College of Technology

Coimbatore.

SIGNATURE

Ms. A.Amsaveni

Project Guide

Department of Electronics and

Communication Engineering,

Kumaraguru College of Technology

Coimbatore.

The candidates with University Register numbers **0710107011, 0710107031** were examined by us in the Project Viva voce Examination held on 18/4/11.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We extend our heartfelt gratitude towards our revered chairman **Arutselvar Dr. N. Mahalingam, Chairman** and the **College Management** for having provided us with the necessary infrastructure to undertake this project.

We are extremely grateful to **Dr. J.Shanmugam, Director,** Kumaraguru College of Technology for having given us a golden opportunity to embark on this project.

We are extremely grateful to **Dr.Ramachandran, Principal,** Kumaraguru College of Technology for having given us a golden opportunity to embark on this project

We are deeply obliged to **Dr.Rajeswari Mariappan, M.E.,Ph.D.,** Head of the Department of Electronics& Communication for her concern and implication during the project course.

We extend our heartfelt thanks to our Project Coordinator **V.Jeyasri Arokiamary, M.E.,Associate Prof,** and **Prof.K.Ramprakash, M.E.,** Professor, Department of Electronics and Communication and our respectable Guide **Ms.A.Amsaveni, M.E.,(Ph.d), Asst.Prof.(SRG)** Department of Electronics and Communication for their helpful guidance and valuable support given to us throughout this project. Our thanks to all **Teaching and Non-teaching staffs** of our department for providing us the technical support for our project.

We also thank **our family and friends** who helped us to complete this project with flying colours.

ABSTRACT

Buses constitute of the major portion of the public transport system in a city but passengers opt for private vehicles such as motor cycles and cars. This is due to the unpunctual operation of the buses. In developing country like India increase use of private vehicles causes a huge traffic congestion. By introducing this system the exact location of the bus, a passenger intends to board is obtained and can be accessed by the passenger. This enables the user to plan his commute. Thus the public bus service becomes more user friendly.

The location of the vehicle or bus can be identified with the help of a GPS module. The location value is stored in the microcontroller and sent to the PC with the help of GSM modem. The route of all buses and the bus stops are already fed into the server. The user/passenger sends the required bus no. (Unique id) to a specified number allotted for the smart bus locale ascertain system along with his location i.e. bus stop. The user/passenger will receive a reply mentioning the current position of the bus and time taken by the bus to arrive to the bus stop that he intended to know.

WE DEDICATE THIS PROJECT TO OUR BELOVED PARENTS, GUIDING STAFFS AND TO OUR FRIENDS

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABULAR COLUMNS	viii
	LIST OF FIGURES	ix
	LIST OF ABBRIVATIONS	x
1.	**INTRODUCTION**	1
	1.1 BRIEF OVERVIEW	2
	1.2 BLOCK DIAGRAM	3
	1.3 BLOCK DIAGRAM DESCRIPTION	4
	1.4 MODULES	4
	1.5 PROJECT DESCRIPTION	5
2.	**MICRO CONTROLLER**	6
	2.1 INTRODUCTION	7
	2.2 ARCHITECTURE DIAGRAM	8
	2.3 PIN CONFIGURATION	10
	2.4 INTERFACING THE CONTROLLER	12
3.	**GPS MODULE**	13
	3.1 OVERVIEW	14
	3.2 NMEA PROTOCOL	15
	3.3 GPRMC SENTENCE	16
	3.4 INTERFACING TO CONTROLLER	22
	3.5 HARDWARE DESCRIPTION	22
4.	**GSM MODULE**	24
	4.1 OVERVIEW	25
	4.2 AT COMMANDS	26
	4.3 INTERFACING TO CONTROLLER	28

5.	**RS 232 COMMUNICATION**	29
	5.1 INTRODUCTION	30
	5.2 SCOPE OF THE STANDARD	31
	5.3 CIRCUIT WORKING DESCRIPTION	31
6.	**LCD DISPLAY**	33
	6.1 INTRODUCTION	34
	6.2 INTERFACING LCD	35
	6.3 DESCRIPTION	36
7.	**MICRO CONTROLLER CODING**	37
	7.1 AVR STUDIO	38
	7.2 ALGORITHM	38
	7.3 FLOW CHART	40
8.	**JAVA CODING**	41
	8.1 INTRODUCTION	42
	8.2 DATA BASE CREATION	43
	8.3 AUTO UPDATING CONCEPT	45
	8.4 AUTOREPLY CONCEPT	45
	8.5 ALGORITHM	46
9.	**POWER SUPPLY**	47
	9.1 INTRODUCTION	48
	9.2 BLOCK DIAGRAM DESCRIPTION	48
	9.3 WORKING PRINCIPLE	49
10.	**ADVANTAGES**	51
11.	**CONCLUSION**	53
12.	**FUTURE SCOPE**	55
	APPENDICES	A-1
	REFERENCES	R-1

LIST OF TABULAR COLUMNS

SL NO.	TITLE	PAGE NO.
2.1	Various memory sizes of the μC	8
3.1	Tabulation showing meaning of each part of the GPRMC Sentence	21
4.1	Tabulation showing various AT Commands for GSM mobiles	27
5.1	Functional tables of MAX 232 IC	32

LIST OF FIGURES

SL NO.	TITLE	PAGE NO.
1.1	Interfacing of GPS and GSM modules with µC - Block diagram	3
2.1	Interfacing GSM modules with PC -Block diagram	3
2.1	Architecture diagram of ATmega164p µC	8
2.2	Pin configuration diagram of Atmega164p µC	10
2.3	Circuit diagram showing the interfacing of GPS and GSM modules with µC through RS 232	12
3.1	Circuit diagram showing the interface of GPS module and µC	22
4.1	Circuit diagram showing the interface of GSM Module and µC	28
5.1	Pin Configuration of MAX 232 IC	30
5.2	Circuit Diagram of MAX 232 IC connected to DB9 Connector	30
5.3	Logic Diagram(positive logic) of Tx and Rx in MAX 232 IC	32
6.1	LCD Display	34
6.2	Circuit Diagram showing interface between LCD to µC	35
7.1	Flow Chart for µC Coding	40
8.1	Window created using JAVA Swing program	43
9.1	Block diagram of Power supply	48
9.2	Circuit diagram of Power supply	49
9.3	Simple circuit diagram of IC Voltage Regulator	50

LIST OF ABBRIVATIONS

GPS	Global Positioning System
GSM	Global System for Mobile
GPRMC	General Purpose Remote Machining Center
NMEA	National Marine Electronics Association
LCD	Liquid Crystal Display
SQL	Structured Query Language
AT	Attention Terminal
USART	Universal Synchronous Asynchronous Receiver Transmitter
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPRS	General Packet Radio Service
RS 232	Recommended Standard 232

Introduction

CHAPTER 1

INTRODUCTION

1.1 BRIEF OVERVIEW:

Smart Bus Locale Ascertain Technology is a Tele-monitoring and management system for city transportation vehicles such as buses. This system is based on a stand-alone single-board embedded system that is equipped with GPS and GSM modems that is installed in the vehicle. A personal computer based GSM-Server is installed at transportation head-quarter. The location of the bus and the expected time it will reach the bus stop is obtained using a mobile phone.

Buses constitute of the major portion of the public transport system in a city but passengers opt for private vehicles such as motor cycles and cars. This is due to the unpunctual operation of the buses. By introducing this system the exact location of the bus, a passenger intends to board is obtained and can be accessed by the passenger. This enables the user to plan his commute. Thus the public bus service becomes more user friendly.

1.2 BLOCK DIAGRAM:

Bus Side:

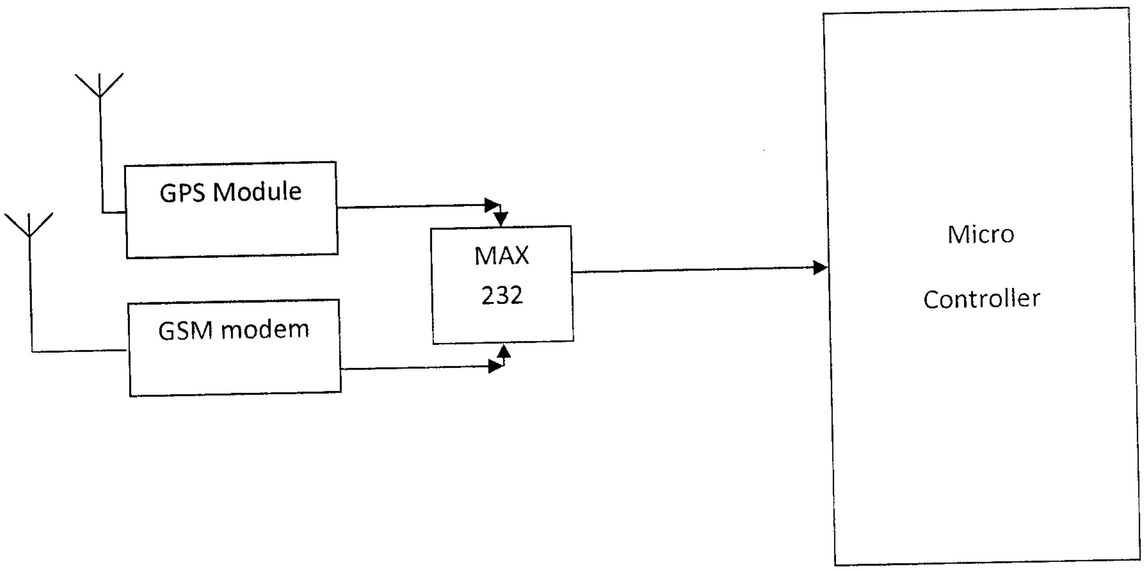

Fig 1.1 Interfacing of GPS and GSM modules with μC - Block diagram

PC Side (Server):

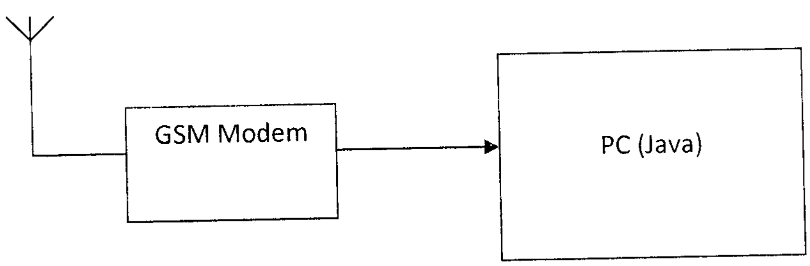

Fig 1.2 Interfacing GSM modules with PC - Block diagram

1.3 BLOCK DIAGRAM DESCRIPTION

On the bus side, the GPS and the GSM modules are interfaced to Microcontroller through MAX 232 IC. MAX 232 IC uses RS-232 standard for serial binary data interconnection between a *DTE* and a *DCE*. The microcontroller used here has two USART ports, which enables it to communicate both the GPS and GSM modules without using a relay.

On the PC side, the GSM modem is interfaces to PC through Hyper Terminal connection. The JAVA program, which acts as front end interface will receive and send messages through this serial port. The received messages from GPS module in bus are updates in back end SQL database and the messages from user are auto replied based on their location and bus location.

1.3 MODULES:

A brief overview of the project modules are explained for better understanding of block diagram.

- Microcontroller – Programmed to receive the GPS coordinates, extract latitude and longitude values and send using GSM mobile.
- LCD Display – Helps to know the location for database creation
- GSM mobile phone (MODEM) – It is the communicating device between Bus module and server PC
- GPS – Gives the exact latitude and longitude co-ordinates
- PC (JAVA) - For data base creation (SQL), auto update and auto reply for the messages from Bus module and user respectively.
- MAX-232 - In telecommunications, MAX 232 uses RS-232 standard for serial binary data interconnection between a *DTE* and a *DCE*. It is commonly used in computer serial ports.

1.4 PROJECT DESCRIPTION:

The location of the vehicle or bus can be identified with the help of a GPS module. The GPS module reads the latitude and longitude of its position. The GPS gives a latitude and longitude value that identifies its current position on earth. This location value is sent to a microcontroller.

The location value is stored in the microcontroller and latitude and longitude values are extracted from the GPRMC string. This value is sent to the PC with the help of GSM modem. In predefined time intervals the above processes are continuously updated.

In the PC (Data base) module, java is used to view the process and the location of bus at all times and the location is stored into PC (server). The PC (Data base) uses a GSM modem to receive the location data. The route of all buses and the bus stops are already fed into the server (SQL Data base).

The user/passenger sends the required bus no. (Unique id) to a specified no. allotted for the smart bus locale ascertain system. The required data such as the position of the bus and the expected time to reach the bus stop is obtained through the mobile phone.

CHAPTER 2

MICRO CONTROLLER

2.1 INTRODUCTION:

Micro controller is a standalone unit, which can perform functions on its own without any requirement for additional hardware like I/O ports and external memory. The heart of the micro controller is the CPU core.

The controller used is **ATMEL ATmega164P.** The ATmega164P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega164P/324P/644P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Microcontroller ATMEGA164p Specifications

- 8 bit micro controller
- Advanced RISC Architecture(*Reduced instruction set computing*)
- ATmega164P/324P/644P also features an extra USART
- Peripheral Features:
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- Number of I/O Pins :32
- Number of Pins :44
- Access Time :20.0 µs
- Clock Speed :20.0 MHz (max)
- Operating Temperature :85.0 °C (max)
- Supply Voltage (DC) :5.50 V (max)
- Special Microcontroller Features

  – Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby.

- **Memory sizes**

Device	Flash	EEPROM	RAM
ATmega164P	16 Kbyte	512 Bytes	1 Kbyte

Table 2.1 Various memory sizes of the µC

2.2 ARCHITECTURE DIAGRAM:

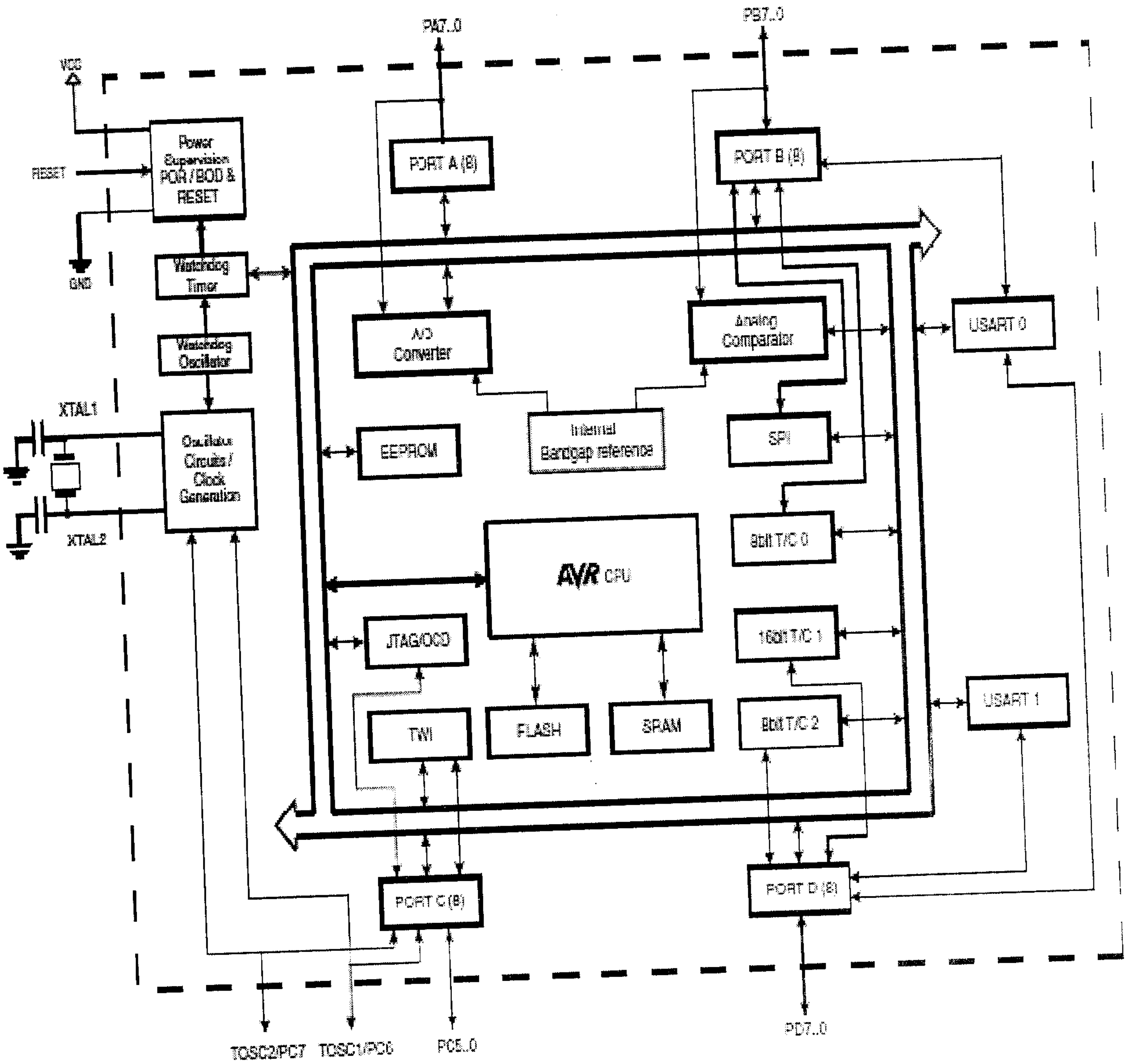

Fig 2.1 Architecture diagram of ATmega164p µC

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs upto ten times faster than conventional CISC microcontrollers.

The Atmega164P/324P/644P provides the following features: 16/32/64K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512B/1K/2K bytes EEPROM, 1/2/4K bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented 2-wire Serial Interface, a 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip

Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

2.3 PIN CONIGURATION:

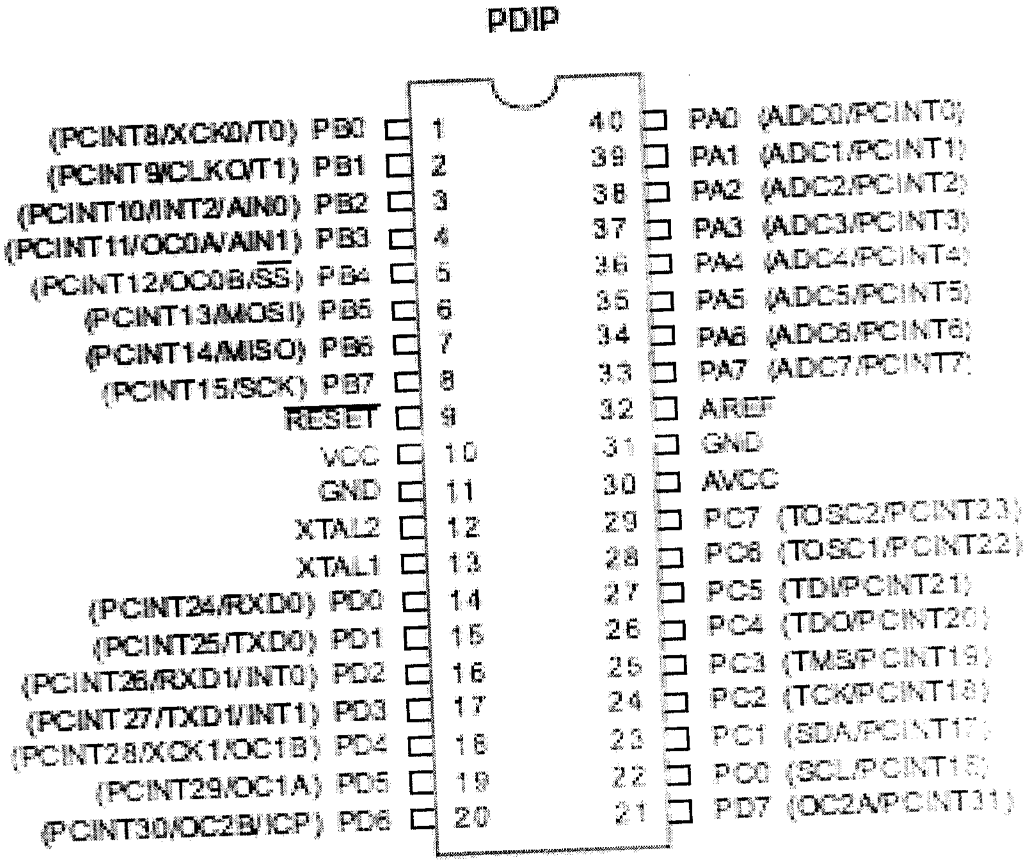

Fig 2.2 Pin configuration diagram of Atmega164p µC

PIN DESCRIPTIONS:

2.3.1 VCC

Digital supply voltage.

2.3.2 GND

Ground.

2.3.3 Port A (PA7:PA0)

Port A serves as analog inputs to the Analog-to-digital Converter. Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the ATmega164P.

2.3.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega164P.

2.3.5 Port C (PC7:PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port C also serves the functions of the JTAG interface, along with special features of the ATmega164P.

2.3.6 Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega164P.

2.3.7 RESET

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

2.3.8 XTAL1

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

2.4 INTERFACING THE CONTROLLER

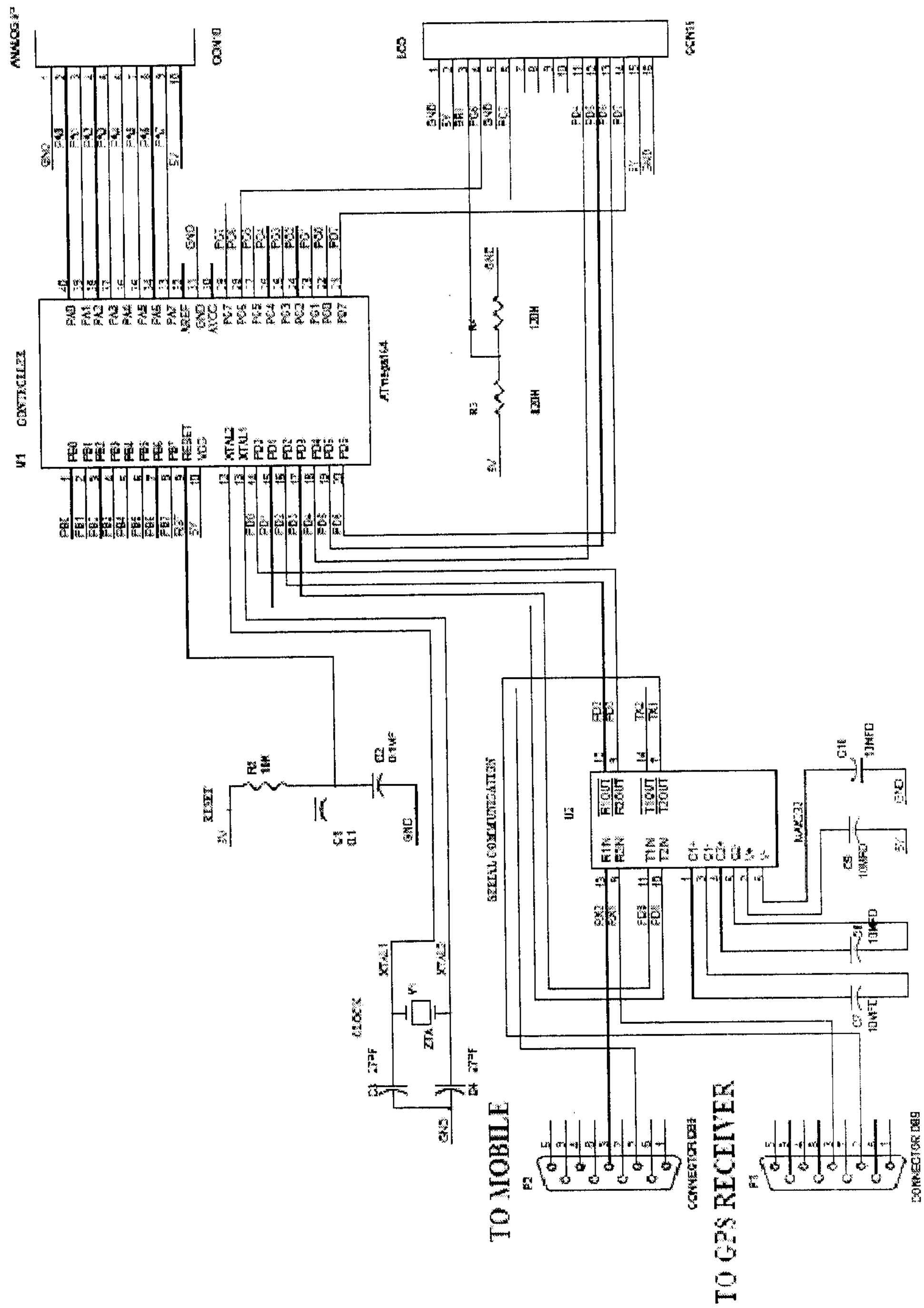

Fig 2.3 Circuit diagram showing the interfacing of GPS and GSM modules with μC through RS 232

GPS Module

CHAPTER 3

GPS MODULE

3.1 OVERVIEW:

The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

HOW IT WORKS:

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position.

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.

THE GPS SATELLITE SYSTEM:

The 24 satellites that make up the GPS space segment are orbiting the earth about 12,000 miles above us. They are constantly moving, making two complete orbits in less than 24 hours. These satellites are travelling at speeds of roughly 7,000 miles an hour.

GPS SIGNAL:

GPS satellites transmit two low power radio signals, designated L1 and L2. Civilian GPS uses the L1 frequency of 1575.42 MHz in the UHF band. The signals travel by line of sight, meaning they will pass through clouds, glass and plastic but will not go through most solid objects such as buildings and mountains. A GPS signal contains three different bits of information — a pseudorandom code, ephemeris data and almanac data. The pseudorandom code is simply an I.D. code that identifies which satellite is transmitting information. this number can be viewed on Garmin GPS unit's satellite page, as it identifies which satellites it's receiving. Ephemeris data tells the GPS receiver where each GPS satellite should be at any time throughout the day. Each satellite transmits ephemeris data showing the orbital information for that satellite and for every other satellite in the system. Almanac data, which is constantly transmitted by each satellite, contains important information about the status of the satellite (healthy or unhealthy), current date and time. This part of the signal is essential for determining a position.

3.2 NMEA PROTOCOL:

The National Marine Electronics Association (NMEA) has developed a specification that defines the interface between various pieces of marine electronic equipment. The standard permits marine electronics to send information to computers and to other marine equipment. A full copy of this standard is available for purchase at their web site. None of the information on this site comes from this standard and I do not have a copy. Anyone attempting to design anything to this standard should obtain an official copy. GPS receiver communication is defined within this specification. Most computer programs that provide real time position information understand and expect data to be in NMEA format. This data includes the complete PVT (position, velocity, time) solution computed by the GPS receiver. The idea of NMEA is to send a line of data called a sentence that is totally self contained and independent from other sentences. There are standard sentences for each device category and there is also the ability to define proprietary sentences for use by the individual company. All of the standard sentences have a two letter prefix that defines the device that uses that sentence type. (For GPS receivers the prefix is GP.) which is followed by a three letter sequence that defines the sentence contents. In addition NMEA permits hardware manufactures to define their own proprietary sentences for whatever purpose they see fit. All proprietary sentences begin with the letter P and are followed with 3 letters that identifies the manufacturer controlling that sentence. For example a Garmin sentence would start with PGRM and Magellan would begin with PMGN.

Each sentence begins with a '$' and ends with a carriage return/line feed sequence and can be no longer than 80 characters of visible text (plus the line terminators). The data is contained within this single line with data items separated by commas. The data itself is just ASCII text and may extend over multiple sentences in certain specialized instances but is normally fully contained in one variable length sentence. The data may vary in the amount of precision contained in the message. For example time might be indicated to decimal parts of a second or location may be show with 3 or even 4 digits after the decimal point. Programs that read the data should only use the commas to determine the field boundaries and not depend on column positions. There is a provision for a checksum at the end of each sentence which may or may not be checked by the unit that reads the data. The checksum field consists of a '*' and two hex digits representing an 8 bit exclusive OR of all characters between, but not including, the '$' and '*'. A checksum is required on some sentences.

There have been several changes to the standard but for GPS use the only ones that are likely to be encountered are 1.5 and 2.0 through 2.3. These just specify some different sentence configurations which may be peculiar to the needs of a particular device thus the GPS may need to be changed to match the devices being interfaced to. Some GPS's provide the ability configure a custom set the sentences while other may offer a set of fixed choices. Many GPS receivers simply output a fixed set of sentences that cannot be changed by the user. The current version of the standard is 3.01. I have no specific information on this version, but I am not aware of any GPS products that require conformance to this version.

3.3 The GPRMC SENTENCE:

This sentence, known as the "Recommended Minimum" sentence, is the most common sentence transmitted by GPS devices. This one sentence contains nearly everything a GPS application needs: latitude, longitude, speed, bearing, satellite-derived time, fix status and magnetic variation.

SENTENCE EXAMPLE:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A

SENTENCE CONTENTS:

The GPRMC sentence consists of twelve comma-delimited words:

THE COMMAND WORD:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A

The command word indicates that the sentence is to be interpreted as a recommended minimum message.

SATELLITE-DERIVED TIME:

$GPRMC,**040302.663**,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A

GPS devices are able to calculate the current date and time using GPS satellites (and not the computer's own clock, making it useful for synchronization). This word stores the current time, in UTC, in a compressed form "HHMMSS.XXX," where HH represents hours, MM represents minutes, SS represents seconds, and XXX represents milliseconds. The above value represents 04:03:02.663 AM UTC.

SATELLITE FIX STATUS:

$GPRMC,040302.663,**A**,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A

When the signals of at least three GPS satellites become stable, the device can use the signals to calculate the current location. The device is said to be "fixed" when calculations of the current location are taking place. Similarly, the phrases "obtaining a fix" or "losing a fix" speak of situations where three signals become stable or obscured, respectively. A value of "A" (for "active") indicates that a fix is currently obtained, whereas a value of "V" (for "invalid") indicates that a fix is not obtained.

LATITUDE DECIMAL DEGREES:

$GPRMC,040302.663,A,**3939.7**,N,10506.6,W,0.27,358.86,200804,,*1A

The latitude represents the current distance north or south of the equator. This word is in the format "HHMM.M" where HH represents hours and MM.M represents minutes. A comma is implied after the second character. This value is used in conjunction with the longitude to mark a specific point on Earth's surface. This sentence says that the current latitude is "39°39.7'N".

LATITUDE HEMISPHERE:

$GPRMC,040302.663,A,3939.7,**N**,10506.6,W,0.27,358.86,200804,,*1A

This word indicates if the latitude is measuring a distance north or south of the equator. A value of "N" indicates north and "S" indicates south. This sentence says that the current latitude is "39°39.7'N".

LONGITUDE DECIMAL DEGREES:

$GPRMC,040302.663,A,3939.7,N,**10506.6**,W,0.27,358.86,200804,,*1A

The longitude represents the current distance east or west of the Prime Meridian. This word is in the format "HHHMM.M" where HHH represents hours and MM.M represents minutes. A comma is implied after the third character. This value is used in conjunction with the latitude to mark a specific point on Earth's surface. This sentence says that the current longitude is "105°06.6'W".

LONGITUDE HEMISPHERE:

$GPRMC,040302.663,A,3939.7,N,10506.6,**W**,0.27,358.86,200804,,*1A

This word indicates if the longitude is measuring a distance east or west of the Prime Meridian. A value of "E" indicates east and "W" indicates west. This sentence says that the current longitude is "105°06.6'W".

SPEED:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,**0.27**,358.86,200804,,*1A

This word indicates the current rate of travel over land, measured in knots.

BEARING:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,**358.86**,200804,,*1A

This word indicates the current direction of travel over, measured as an "azimuth." An azimuth is a horizontal angle around the horizon measure in degrees between 0 and 360, where 0 represents north, 90 represents east, 180 represents south, and 270 represents west. This word indicates that the direction of travel is 358.86°, or close to north.

UTC DATE:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,**200804**,,*1A

GPS devices maintain their own date and time calculated from GPS satellite signals. This makes GPS devices useful for clock synchronization since the date and time are independent of the local machine's internal clock. This word contains two-digit numbers for days, followed by months and years. In the example above, the date is August (08) 20th (20), 2004 (04). The two-digit year is added to 2000 to make a full year value.

THE CHECKSUM:

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,***1A**

The checksum is used to identify errors in the data which may have occurred during transmission.

GPS DATA STRING:

$GPRMC,hhmmss,A,llll.ll,N,yyyyy.yy,W,kk.k,tt.t,ddmmyy,mm.m,W*CS

Where:

- hhmmss is the UTC time
- A is the status: data valid (A) or receiver warning (V)
- llll.ll is latitude (degrees, minutes.m - ddmm.mm)
- N is North or South
- yyyyy.yy is longitude (degrees, minutes.m - dddmm.mm)
- W is West or East
- kk.k is speed over ground in knots
- tt.t is track made good, degrees true
- ddmmyy is the date
- mm.m is magnetic variation, degrees
- W is West or East
- *CS is the check sum

Sentence Example

- $GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A

NAME	DATA	DESCRIPTION
The Command Word	$GPRMC	Indicates that the sentence is to be interpreted as a recommended minimum message.
Satellite-Derived Time	040302.663	The current time.
Satellite Fix Status	A	A value of "A indicates that a fix is currently obtained.
Latitude Decimal Degrees	3939.7	The latitude represents the current distance north or south of the equator.
Latitude Hemisphere	N	This word indicates if the latitude is measuring a distance north or south of the equator.
Longitude Decimal Degrees	10506.6	The longitude represents the current distance east or west of the Prime Meridian.
Longitude Hemisphere	W	This word indicates if the longitude is measuring a distance east or west of the Prime Meridian.
Speed	0.27	This word indicates the current rate of travel over land, measured in knots.
Bearing	358.86	This word indicates the current direction of travel over, measured as an "azimuth
UTC Date	200804	This makes GPS devices useful for clock synchronization since the date and time are independent of the local machine's internal clock.
The Checksum	*1A	The checksum is used to identify errors in the data which may have occurred during transmission.

Table 3.1 Tabulation showing meaning of each part of the GPRMC Sentence

3.4 INTERFACING TO CONTROLLER

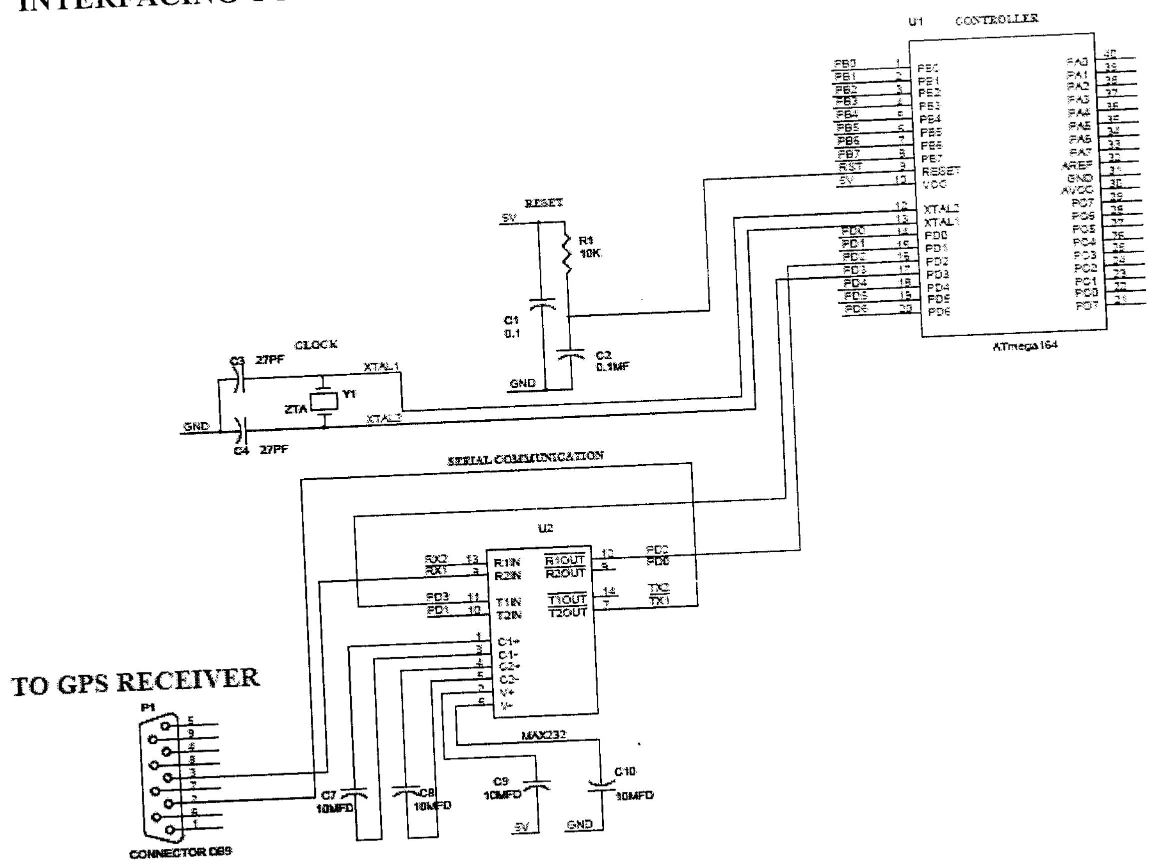

Fig 3.1 Circuit diagram showing the interface of GPS module and μC

3.5 HARDWARE DESCRIPTION:

The hardware interface for GPS units is designed to meet the NMEA requirements. They are also compatible with most computer serial ports using RS232 protocols. however strictly speaking the NMEA standard is not RS232. They recommend conformance to EIA-422. The interface speed can be adjusted on some models but the NMEA standard is 4800 b/s (bit per second rate) with 8 bits of data, no parity, and one stop bit. All units that support NMEA should support this speed. Note that, at a b/s rate of 4800, you can easily send enough data to more than fill a full second of time. For this reason some units only send updates every two seconds or may send some data every second while reserving other data to be sent less often. In addition some units may send data a couple of seconds old while other units may send data that is collected within the second it is sent. Generally time is sent in some field within each second so it is pretty easy to figure out what a particular GPS is doing. Some sentences may be sent only during a particular action of the receiver such as while following a route while other receivers may always send the sentence and just null out the values. Other difference will be noted in the specific data descriptions defined later in the text.

At 4800 b/s you can only send 480 characters in one second. Since an NMEA sentence can be as long as 82 characters you can be limited to less than 6 different sentences. The actual limit is determined by the specific sentences used, but this shows that it is easy to overrun the capabilities if you want rapid sentence response. NMEA is designed to run as a process in the background spitting out sentences which are then captured as needed by the using program. Some programs cannot do this and these programs will sample the data stream, then use the data for screen display, and then sample the data again. Depending on the time needed to use the data there can easily be a lag of 4 seconds in the responsiveness to changed data. This may be fine in some applications but totally unacceptable in others. For example a car traveling at 60 mph will travel 88 feet in one second. Several second delays could make the entire system seem unresponsive and could cause you to miss your turn.

GSM Module

CHAPTER 4

GSM MODULE

4.1 OVERVIEW:

Global System for Mobile Communications or GSM is the world's most popular standard for mobile telephone systems. When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages. A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities.

For the purpose of this document, the term GSM modem is used as a generic term to refer to any modem that supports one or more of the protocols in the GSM evolutionary family, including the 2.5G technologies GPRS and EDGE, as well as the 3G technologies WCDMA, UMTS, HSDPA and HSUPA.

A GSM modem exposes an interface that allows applications such as NowSMS to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. To perform these tasks, a GSM modem must support an "extended AT command set" for sending/receiving SMS messages, as defined in the ETSI GSM 07.05 and and 3GPP TS 27.005 specifications

4.1.1 MODULATION:

The modulation used in GSM is Gaussian minimum-shift keying (GMSK), a kind of continuous-phase frequency shift keying. In GMSK, the signal to be modulated onto the carrier is first smoothed with a Gaussian low-pass filter prior to being fed to a frequency modulator, which greatly reduces the interference to neighboring channels (adjacent-channel interference).

4.1.2 GSM CARRIER FREQUENCIES:

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands.

4.2 AT COMMANDS:

4.2.1 OVERVIEW

AT commands are also known as Hayes AT commands. There are different views to understand the meanings of "AT". Some call it "Attention Telephone", whereas others interpret it as "Attention Terminal" commands.

AT commands allow giving instructions to both mobile devices and ordinary landline telephones. The commands are sent to the phone's modem, which can be a GSM modem or PC modem. This article focuses on AT commands on Nokia's GSM and WCDMA products only. Different manufacturers may have different sets of AT commands. Fortunately, many AT commands are the same. Mobile device manufacturers may also give attention to operators to allow or not to allow some commands on phones.

AT commands can be used for operations that are usually done from the keypad, for instance calling a number, sending, reading, or deleting an SMS, setting the SMSC number, looking for a GPRS access point, reading and deleting phonebook data, reading the battery status, reading the signal strength, and so on. When you want to make a PC-based application to interface a mobile phone using USB, IR, or Bluetooth, these commands are needed to communicate with mobile phones. Basically such commands are the application layer of MBUS or FBUS commands. Nokia provides an AT command set guide, where you can see the basic command syntax and the response of the command in various situations.

4.2.2 SMS COMMANDS:

SMS TEXT MODE:

Command	Description
AT+CSMS	Select message service
AT+CPMS	Preferred message storage
AT+CMGF	Message format
AT+CSCA	Service centre address
AT+CSMP	Set text mode parameters
AT+CSDH	Show text mode parameters
AT+CSCB	Select cell broadcast message types
AT+CSAS	Save settings
AT+CRES	Restore settings
AT+CNMI	New message indications to TE
AT+CMGL	List messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMSS	Send message from storage
AT+CMGW	Write message to memory
AT+CMGD	Delete message

Table 4.1 Tabulation showing various AT Commands for GSM mobiles

RS 232 Communication

CHAPTER 5

RS 232 COMMUNICATION

5.1 INTRODUCTION:

In telecommunications, **RS-232** is a standard for serial binary data interconnection between a *DTE* (Data terminal equipment) and a *DCE* (Data Circuit-terminating Equipment). It is commonly used in computer serial ports. The general circuit diagram of MAX 232 IC connected to DB9 serial connector for PC is shown below.

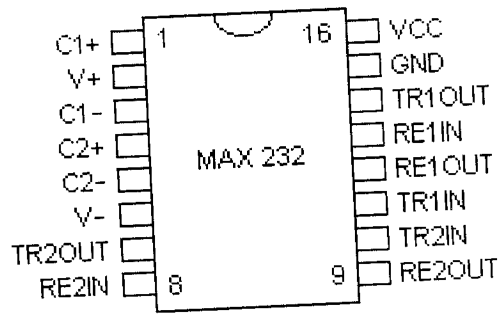

Fig 5.1 Pin Configuration of MAX 232 IC

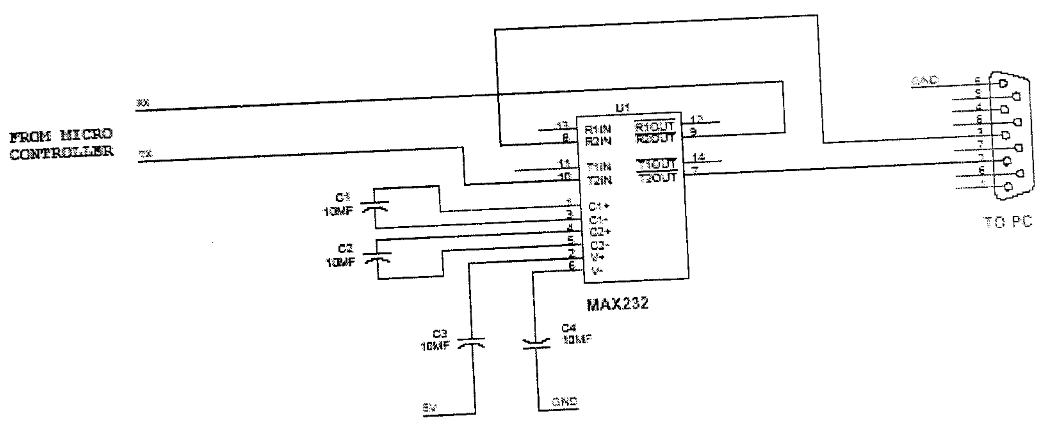

Fig 5.2 Circuit Diagram of MAX 232 IC connected to DB9 Connector

5.2 SCOPE OF THE STANDARD:

The Electronic Industries Alliance (EIA) standard RS-232-C [3] as of 1969 defines:

- Electrical signal characteristics such as voltage levels, signaling rate, timin g and slew-rate of signals, voltage withstand level, short-circuit behavior, maximum stray capacitance and cable length
- Interface mechanical characteristics, pluggable connectors and pin identification
- Functions of each circuit in the interface connector
- Standard subsets of interface circuits for selected telecom applications

The standard does not define such elements as character encoding (for example, ASCII, Baudot or EBCDIC), or the framing of characters in the data stream (bits per character, start/stop bits, parity). The standard does not define protocols for error detection or algorithms for data compression.

The standard does not define bit rates for transmission, although the standard says it is intended for bit rates lower than 20,000 bits per second. Many modern devices can exceed this speed (38,400 and 57,600 bit/s being common, and 115,200 and 230,400 bit/s making occasional appearances) while still using RS-232 compatible signal levels.

Details of character format and transmission bit rate are controlled by the serial port hardware, often a single integrated circuit called a UART that converts data from parallel to serial form. A typical serial port includes specialized driver and receiver integrated circuits to convert between internal logic levels and RS-232 compatible signal levels.

5.3 CIRCUIT WORKING DESCRIPTION:

In this circuit the MAX 232 IC used as level logic converter. The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA 232 voltage levels from a single 5v supply. Each receiver converts EIA-232 to 5v TTL/CMOS levels. Each driver converts TLL/CMOS input levels into EIA-232 levels.

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

Table 5.1 Functional tables of MAX 232 IC

logic diagram (positive logic)

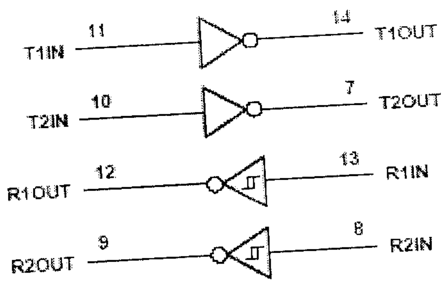

Fig 5.3 Logic Diagram(positive logic) of Tx and Rx in MAX 232 IC

In this circuit the microcontroller transmitter pin is connected in the MAX232 T2IN pin which converts input 5v TTL/CMOS level to RS232 level. Then T2OUT pin is connected to reviver pin of 9 pin D type serial connector which is directly connected to PC.

In PC the transmitting data is given to R2IN of MAX232 through transmitting pin of 9 pin D type connector which converts the RS232 level to 5v TTL/CMOS level. The R2OUT pin is connected to receiver pin of the microcontroller. Likewise the data is transmitted and received between the microcontroller and PC or other device vice versa.

LCD Display

absorbed by the rear analyzer. The observer sees a black character on a sliver gray background. When the electric field is turned off, the molecules relax back to their 90 degree twist structure. This is referred to as a positive image, reflective viewing mode. Carrying this basic technology further, an LCD having multiple selectable electrodes and selectively applying voltage to the electrodes, a variety of patterns can be achieved.

6.2 INTERFACING LCD:

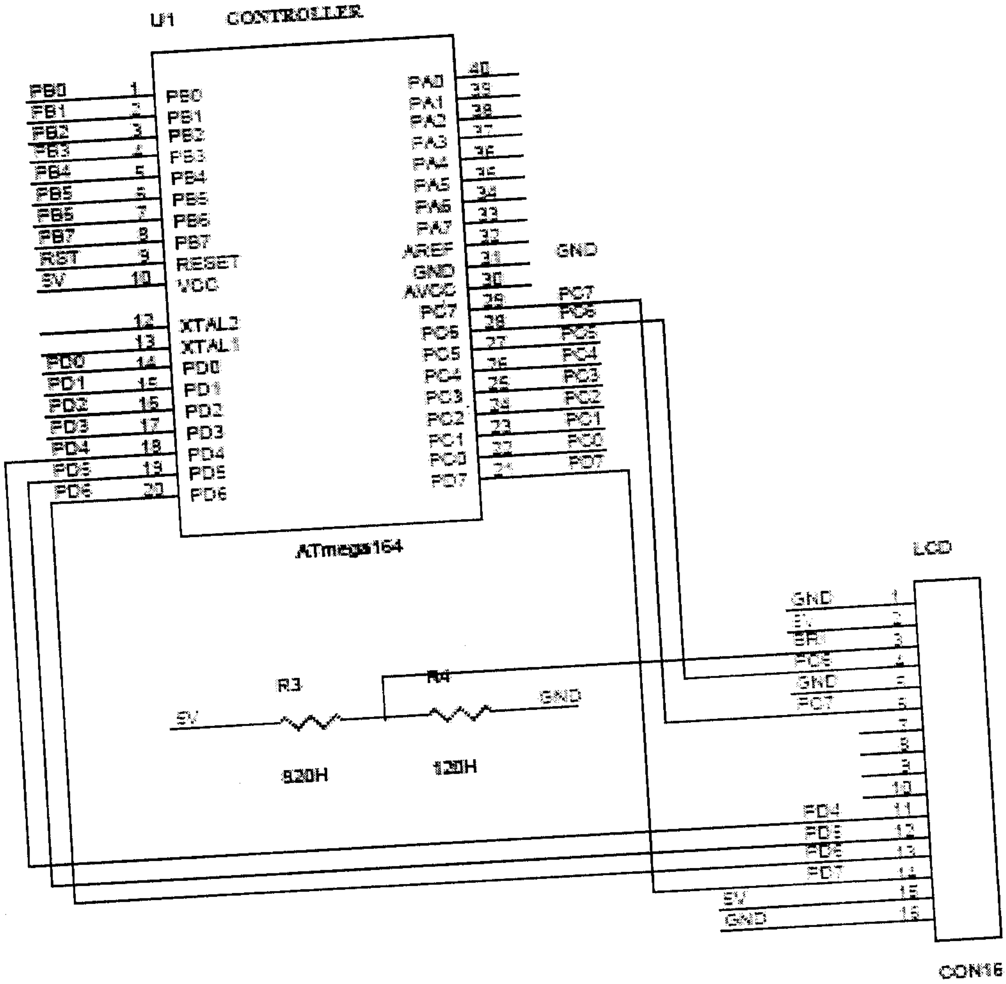

Fig 6.2 Circuit Diagram showing interface between LCD to μC

6.3 DESCRIPTION

The display used here is 16x2 LCD (Liquid Crystal Display); this means 16 characters per line by 2 lines. A very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the Atmega164p) and communicates directly with the LCD. The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. Here we are using 8-bit mode of LCD, i.e., using 8-bit data bus. The LCD display requires a power supply of 5V. It is interfaced with microcontroller through port C and port D. The microcontroller is programmed to display the current location GPS coordinates. This is helpful to preprogram the Database in the server. The three control lines are referred to as EN, RS, and RW.

The EN line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The RS line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Microcontroller Coding

CHAPTER 7
MICROCONTROLLER CODING

7.1 AVR STUDIO:

AVR Studio is used by embedded programmers for programming and debugging for many of the Atmel microprocessors such as the Atmega8 or even the Atmega164p. While it has support for assembly programming for those who prefer to use higher languages, it uses the COFF format for debugging. Beginning with version 4 AVR Studio has now moved to dwarf2, and can be more readily used in conjunction with the open source GCC based compiler WinAVR.

7.2 ALGORITHM:

Step 1

- Include all necessary library functions.
- Declare global functions to send message through the mobile.
- Declare other global variables for iteration, to store GPRMC string, latitude and longitude data, count time.

Step 2

- Start main function.
- Initialize port D as output port.
- Set the timer/counter values as requires from the datasheet.
- Set the baud rate for the GPS and GSM mobile for serial communication.

Step 3

- Get the GPS data from the GPS module through serial port_1, copy it to a one dimensional array.
- Check whether the received data has GPRMC string, which starts with ' $GPRMC '

- Keep checking the data till this string is obtained.
- Separate the latitude and longitude values from the GPRMC string and store them is two separate arrays and show the same on LCD display.

Step 4

- Initiate the microcontroller serial communication with GSM mobile using 'AT commands'.
- Send the 10 digit GSM mobile number (DATA BASE) to which the message should be sent.
- Using 'AT commands' initiate SMS service by 'AT+CMGS' command.
- SMS the latitude and longitude array, obtained from the GPS module.

7.3FLOW CHART:

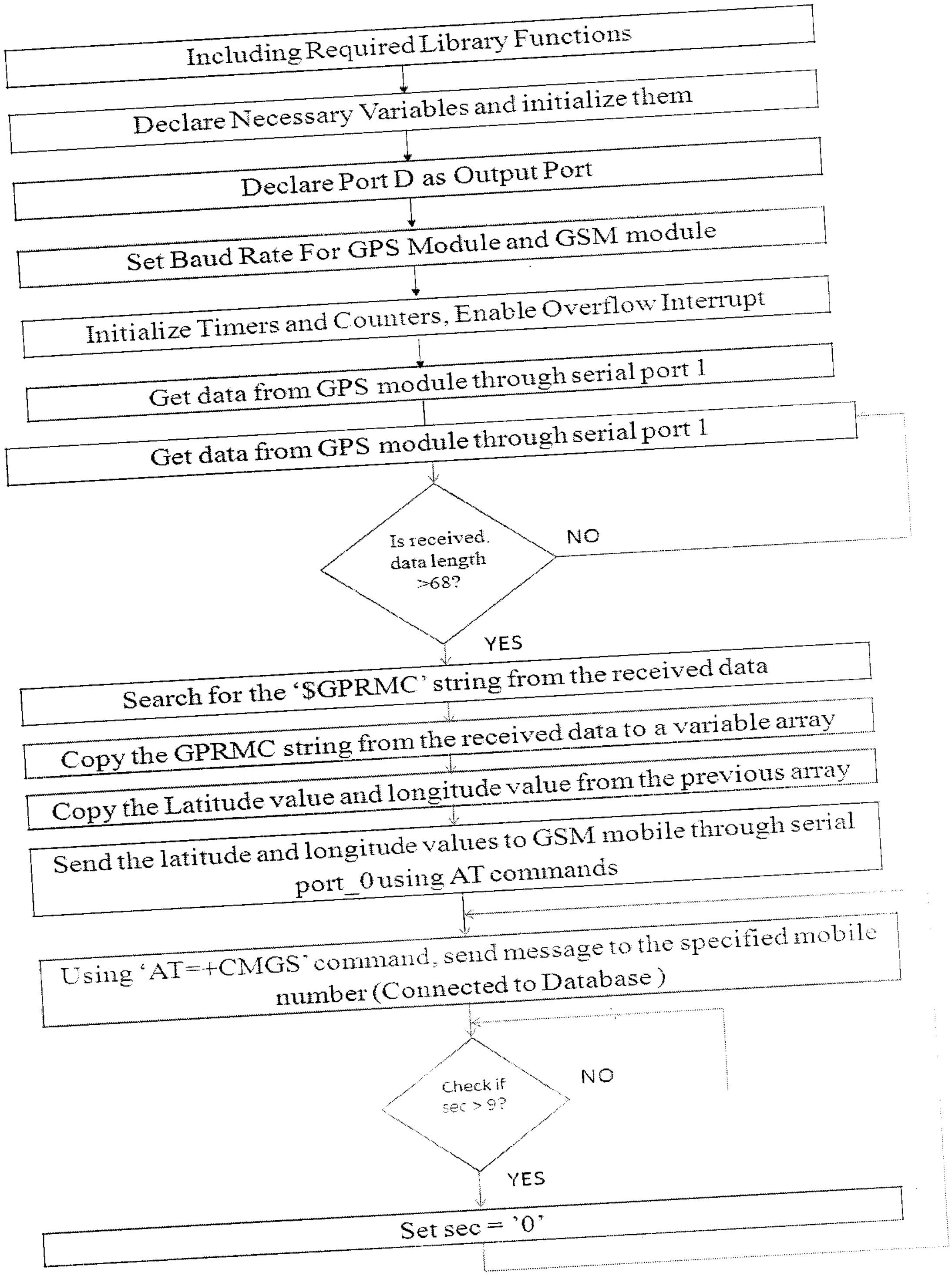

Fig 7.1 Flow Chart for µC Coding

Java Coding

CHAPTER 8

JAVA CODING

8.1 INTRODUCTION

Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. Java is used in mobile phones, Web servers and enterprise applications, and while less common on desktop computers, Java applets are sometimes used to provide improved and secure functionalities while browsing the World Wide Web.

Writing in the Java programming language is the primary way to produce code that will be deployed as Java byte code, though there are byte code compilers available for other languages such as JavaScript, Python, and Ruby. Several new languages have been designed to run natively on the Java Virtual Machine (JVM), such as Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C. Java eliminates certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

The heart of the Java Platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT compiler within the *Java Virtual Machine*, or JVM. The JIT compiler translates the Java byte code into native processor instructions at run-time and caches the native code in memory during execution.

JAVA SWING:

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit. Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform.

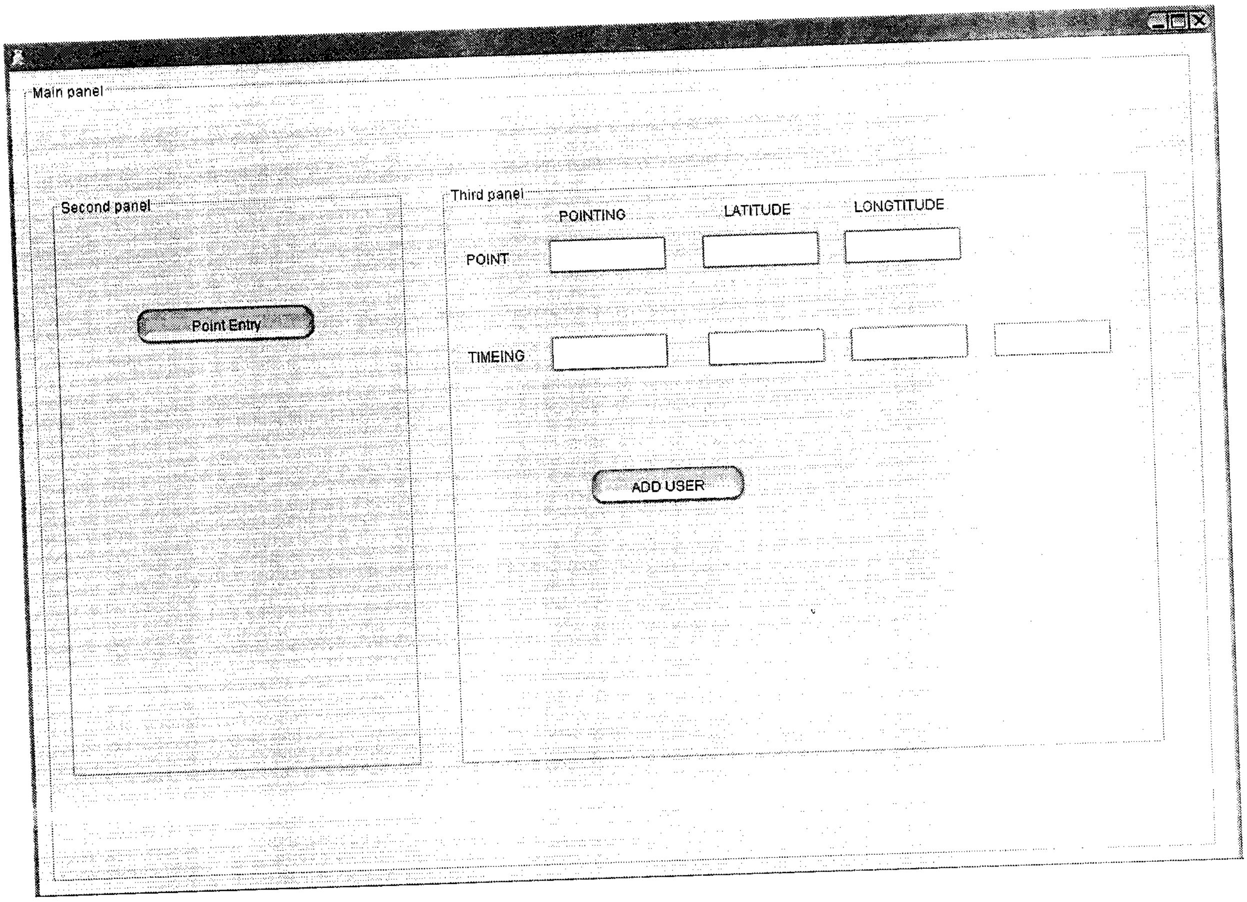

Fig 8.1 Window created using JAVA Swing program

8.2 DATA BASE CREATION:

SQL:

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL is an ANSI (American National Standards Institute) standard

SQL FUNCTIONS:

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Although SQL is an ANSI (American National Standards Institute) standard, there are many different versions of the SQL language.

However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

SQL often referred to as Structured Query Language, is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra and calculus. Its scope includes data insert, query, update and delete, schema creation and modification, and data access control. SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not adhering to the relational model as described by Codd, it became the most widely used database language.

8.3 AUTO UPDATING CONCEPT:

Java program is the front end user interface to the project. It communicates with the GSM mobile through serial communication, using AT Commands. It can read the message and deletes the message using this AT Commands.

SQL is the back end on the PC, which is a data base collection, which is partially filled during the implementation, and the rest is updated during the program run. The bus stop locations are fed into the database initially along with the time taken by the bus to teach there from all other stops. The current location of the bus will be loaded dynamically during the program run period.

When the message is received from a GPS module in a bus, it identifies the message, which starts with a '*' sign and automatically updates the current location of the bus into the database.

8.4 AUTO REPLY CONCEPT:

When the message is received from the user, which mentions his/her current location, the program will find the time that will be required by the bus to reach the mentioned location. This is done by comparing the user's location and the bus's current location. This time will be sent to the user, from which the request has been made.

It is possible to read one message at a time by these AT Commands, which is received at first. Hence once the message is read from the GSM mobile it will be deleted automatically by using AT Command in order to read the next message. This operation is done automatically for every message. Once all the messages in the inbox are read, the program waits for a minute before terminating. In case if any message is received during this time, it'll perform the required operation.

CHAPTER 9

POWER SUPPLY

9.1 INTRODUCTION

The ac voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

9.2 BLOCK DIAGRAM DESCRIPTION:

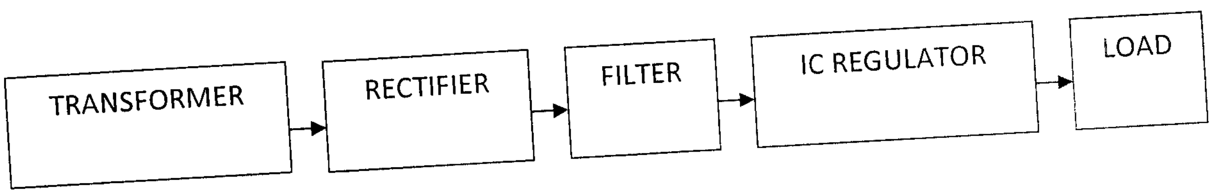

Fig 9.1 Block diagram of Power supply

9.2.1 Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

9.2.2 Bridge rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

8.5 ALGORITHM

Step 1

- Check for the message in Inbox of the mobile using AT commands through the serial connection.
- If message is present, read the message using "AT+CMGR=1" command.

Step 2

- Check whether the message is from the GPS module in the bus or from the user.
- If the message is from GPS module in the bus, call the auto update function.
- If the message is from the user, call the auto reply function.

Step 3

- In the auto update function, store the received location of the bus in a table.
- Find the current location of the bus by comparing coordinates of each bus stop with the received coordinates.
- Store the current bus stop details is another table, where the bus is located.

Step 4

- In the auto reply function, check the location of the user based on the codeword for each bus stop.
- From the auto update function, obtain the details of the current bus location.
- Send message to the user using 'AT Commands' through the serial connection, containing the information about current location of bus and the time taken by it to reach the user location.

Power Supply

9.3 WORKING PRINCIPLE:

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

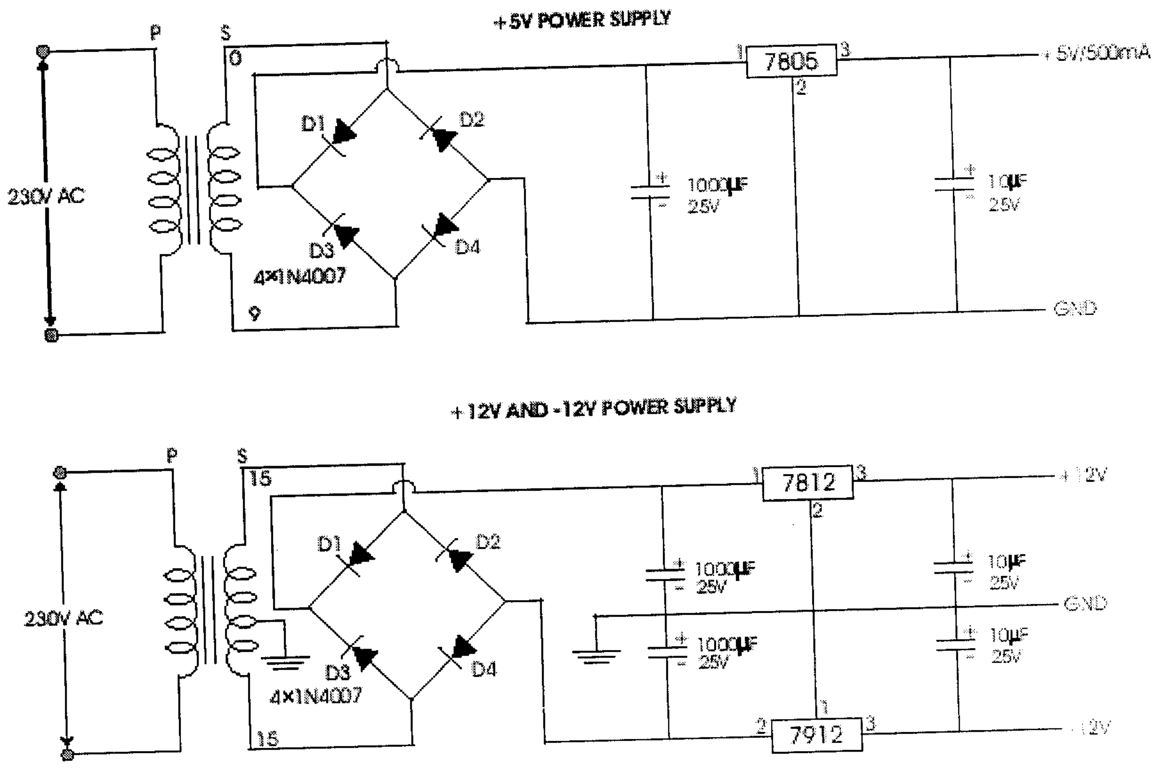

Fig 9.2 Circuit diagram (Power supply)

9.3.1 IC voltage regulators

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

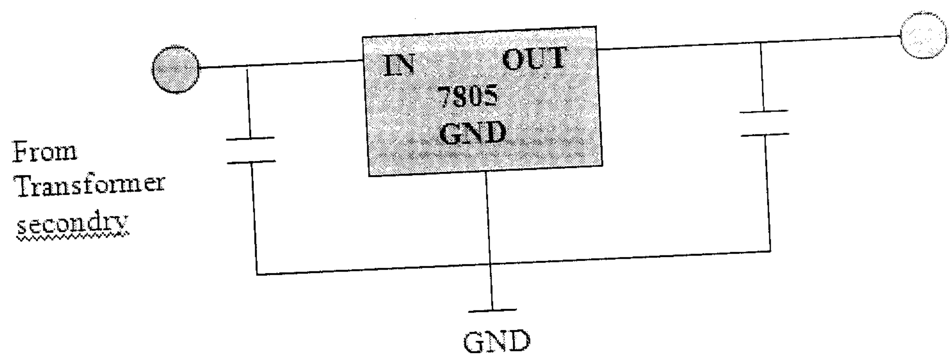

Fig 9.3 Simple circuit diagram of IC Voltage Regulator

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

- For ICs, microcontroller, LCD --------- 5 volts
- For GPS ---------- 12 volts

Advantages

CHAPTER 10

ADVANTAGES

- This project is very useful to passengers (users) because they can easily identify the location of the bus.
- This project helps to bring about an efficient public transport system thus enabling more passengers to opt for busses rather than private vehicles. This brings about a lot more advantages such as less traffic congestion, less usage of fossil fuels, better control over the traffic, etc.
- Low power consumption by the device in vehicle.
- Easy monitoring of the public transport system.
- Enables passenger friendly public transport systems.
- Saves time for public vehicle users.
- Less traffic congestion in the cities.

Conclusion

CHAPTER 11

CONCLUSION

In cities across the globe, the personal automobile is the single greatest polluter, as emissions from a billion vehicles on the road add up to a planet-wide problem. Driving a private car is a typical citizen's most air polluting activity. The negative effects of automotive emissions are maximum when you sit in traffic surrounded by cars, their engines idling. Everyone sitting in a traffic jam is getting poisoned. The study concludes that since pre-industrial times, 15 percent of the RF caused by man-made CO2-emissions has come from the transport sector.

Implementing this project helps to minimize the number of private vehicles such as cars, motor bikes, there by decrease in the pollution due to automobiles. Also it reduces the traffic intensity during the busy hours. The percent of RF can be reduced to 10% from the actual current value of 15%.

Future Scope

CHAPTER 12

FUTURE SCOPE

This project currently uses a front end Java program and back end data base in SQL, which is related to one particular city. Data related to each bus stop is to be fed initially into the data base manually. And also provides only GSM communication with the users.

With the increase in the usage of Internet, GPRS, WAP connections all around the world, in near future this project can provide access to the bus locations through online. User can view his/her required bus information online. This makes this project even more user friendly as all GSM mobiles are now having the access to online, which is quite fast and cheaper than conventional SMS system in GSM. Also the bus routes in all cities throughout India can be fed into one single application on online, so that the new user in the cities can view online which bus goes where, along with finding of the location of the bus and time to reach the bus stop. Also with less effort new bus stops can be added, without editing the whole database.

Appendices

APPENDIX – I

AVR505: Migration between ATmega16/32 and ATmega164P/324P/644P

8-bit AVR Microcontrollers

Application Note

Features

- General porting considerations
- Memories
- I/O Mapping and SRAM
- EEPROM Programming
- External 32 kHz Watch Crystal
- Changes To Register and Bit Names
- Fuse Settings
- Interrupt Vectors
- IEEE 1149.1 (JTAG) Boundary Scan
- Operational Range

1 Introduction

The ATmega164P and the ATmega324P are new and enhanced versions of the ATmega16 and the ATmega32 respectively, and ATmega644P is a new 64kB device with the same features. An important improvement is the possibility for low voltage operation (1.8V) with ATmega164P/324P/644P and decreased power consumption. ATmega164P/324P/644P also features an extra USART and can run at frequencies up to 20 MHz.

This application note summarizes the differences between ATmega16/32 and ATmega164P/324P/644P and is a guide to assist current ATmega16/32 users in converting existing designs to the ATmega164P/324P/644P. Note that electrical differences such as power consumption and I/O driving capabilities are not covered in this document. Refer to the datasheets for detailed information on the devices. Migration from ATmega644 is covered by AVR508: Migration from ATmega644 to ATmega644P.

ATmega164P/324P/644P are pin compatible with ATmega163/323, and migration between these devices is possible but not within the scope of this application note. Refer to AVR083: Replacing ATmega163 by ATmega16 and AVR084: Replacing ATmega323 by ATmega32 for further details.

Rev. 8001C-AVR-06/05

2 General porting considerations

To make the porting process as easy as possible, we recommend to always refer to registers and bit positions using their defined names, as absolute addresses and values may change from device to device. When porting a design it is then often just necessary to include the correct definition file. Some examples are shown below.

```
PORTE |=  (1<<PORTE5);           // Set pin 5 on port E high
DDR   &= ~(1<<PORTE5);           // Set pin 5 on port E as input

// Configure USI
USICR  =  (1<<USISIE)|(0<<USIOIE)|(1<<USIWM1)|(0<<USIWM0)
          (1<<USICS1)|(0<<USICS0)|(0<<USICLK)|(0<<USITC);
```

To avoid conflicts with added features and register functionality, never access registers that are marked as reserved. Reserved bits should always be written to zero if accessed. This ensures forward compatibility, and that added features will stay in their default states if not used.

3 Memories

The memory sizes are the same for the ATmega16 and the ATmega164P as well as for the ATmega32 and ATmega324P, hence this does not imply a considerable difference in migrating between them. Please note the different page size for 644P memories as shown in Table 3-2.

Table 3-1. Memory Sizes.

	ATmega16/164P	ATmega32/324P	ATmega644P
FLASH	16 kb	32 kb	64 kb
SRAM	1 kb	2 kb	4 kb
EEPROM	512 b	1 kb	2 kb

Table 3-2. Page Sizes.

	ATmega16/164P	ATmega32/324P	ATmega644P
FLASH	64 words	64 words	128 words
EEPROM	4 bytes	4 bytes	8 bytes

4 I/O Mapping and SRAM

The I/O memory space contains 64 addresses for CPU peripheral control registers. The ATmega164P/324P/644P I/O space and I/O range are changed and extended compared to ATmega16/32. The extended I/O space goes from 0x60 to 0xFF in data memory space where ST/STS/STD and LD/LDS/LDD instructions must be used.

The memory map is slightly different between the ATmega16/32 and the ATmega164P/324P/644P due to extended I/O space. The ATmega164P/324P/644P internal data SRAM addressing starts at 0x100 as opposed to 0x60 in ATmega16/32.

5 EEPROM Programming

In ATmega16/32 EEPROM data programming is done in one atomic operation, in ATmega164P/324P/644P it is also possible to split the erase and write operations in two different operations. The typical EEPROM programming times are also reduced in ATmega164P/324P/644P.

Table 5-1. EEPROM Programming Time.

	ATmega16/32		ATmega164P/324P/644P	
	Number of Calibrated RC Oscillator Cycles	Typical Programming Time	Number of Calibrated RC Oscillator Cycles	Typical Programming Time
EEPROM write from CPU	8448	8.5 ms	26368	3.3 ms

6 External 32 kHz Watch Crystal

When operating the timer/counter from an external 32.768 kHz watch crystal or an external clock source, external capacitors might be needed to the TOSC1/2 pins when using ATmega164P/324P/644P, as opposed to ATmega16/32 where the crystal can be connected directly between the pins. The internal capacitance of ATmega164P/324P/644P low-frequency oscillator is typically 6pF, but the tracks to the crystal will add some additional capacitance. Refer to the datasheet for details on crystal connections.

The low frequency crystal oscillator of the ATmega164P/324P/644P is optimized for very low power consumption and thus the crystal driver strength is reduced compared to the ATmega16/32. This means that when selecting a crystal, its load capacitance and Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor. Table 6-1 shows the ESR recommendations for ATmega164P/324P/644P.

Table 6-1. ESR recommendation 32.768 kHz crystals with ATmega164P/324P/644P.

Crystal CL [pF]	Max ESR [kΩ][1]
6.5	75
9	65
12.5	30

Note: 1. The values stated are for an oscillator allowance safety margin of 5. Since the oscillator's transconductance is temperature compensated one can use a safety margin of 4, thus giving a max ESR of 90, 80 and 40 kΩ respectively.

For examples of crystals that comply with the requirements see Appendix A.

The startup times are also increased as shown in Table 6-2.

Table 6-2. Startup times with 32.768 kHz crystals.

Crystal CL [pF]	Startup time[2] [ms] ATmega16/32	Startup time[2] [ms] ATmega164P/324P/644P
6.5	-	600
9	300	700
12.5	400	1700

Note: 2. Crystals usually need ~3000ms before they are completely stable with any oscillator design. The time stated is before the crystal is running with a sufficient amplitude and frequency stability.

7 Changes To Register and Bit Names

Several modifications have been done in register and bit naming conventions between ATmega16/32 and ATmega164P/324P/644P. The locations of the registers are changed considerably.

7.1 Registers

Table 7-1. Changes to Register Names and Locations.

Address in ATmega16/32	Register Name in ATmega16/32	Address in ATmega164P/324P/644P	Register Name in ATmega164P/324P/644P
$3C ($5C)	OCR0	$27 ($47)	OCR0A
$3B ($5B)	GICR	$1D ($3D)	EIMSK
$3A ($5A)	GIFR	$1C ($3C)	EIFR
$39 ($59)	TIMSK	($6E) ($6F) ($70)	TIMSK0 TIMSK1 TIMSK2
$38 ($58)	TIFR	$15 ($35) $16 ($36) $17 ($37)	TIFR0 TIFR1 TIFR2
$37 ($57)	SPMCR	$37 ($57)	SPMCSR
$36 ($56)	TWCR	($BC)	TWCR
$35 ($55)	MCUCR	$33 ($53) ($69)	SMCR EICRA
$34 ($54)	MCUCSR	$34 ($54)	MCUSR[3]
$33 ($53)	TCCR0	$24 ($44)	TCCR0A[3]
$32 ($52)	TCNT0	$26 ($46)	TCNT0
$31 ($51)	OSCCAL OCDR	($66) 31 ($51)	OSCCAL OCDR
$30 ($50)	SFIOR	($7B)	ADCSRB[3]
$2F ($4F)	TCCR1A	($80)	TCCR1A[2]
$2E ($4E)	TCCR1B	($81)	TCCR1B
$2D ($4D)	TCNT1H	($85)	TCNT1H
$2C ($4C)	TCNT1L	($84)	TCNT1L

Address in ATmega16/32	Register Name in ATmega16/32	Address in ATmega164P/324P/644P	Register Name in ATmega164P/324P/644P
$2B ($4B)	OCR1AH	($89)	OCR1AH
$2A ($4A)	OCR1AL	($88)	OCR1AL
$29 ($49)	OCR1BH	($8B)	OCR1BH
$28 ($48)	OCR1BL	($8A)	OCR1BL
$27 ($47)	ICR1H	($87)	ICR1H
$26 ($46)	ICR1L	($86)	ICR1L
$25 ($45)	TCCR2	($B0)	TCCR2A[3]
$24 ($44)	TCNT2	($B2)	TCNT2
$23 ($43)	OCR2	($B3)	OCR2A
$22 ($42)	ASSR	($B6)	ASSR[3]
$21 ($41)	WDTCR	($60)	WDTCSR[3]
$20 ($40)	UBRRH UCSRC	($C5) ($C2)	UBRR0H[3] UCSR0C[3]
$1F ($3F)	EEARH	$22 ($42)	EEARH
$1E ($3E)	EEARL	$21 ($41)	EEARL
$1D ($3D)	EEDR	$20 ($40)	EEDR
$1C ($3C)	EECR	$1F ($3F)	EECR
$1B ($3B)	PORTA	$02 ($22)	PORTA
$1A ($3A)	DDRA	$01 ($21)	DDRA
$19 ($39)	PINA	$00 ($20)	PINA
$18 ($38)	PORTB	$05 ($25)	PORTB
$17 ($37)	DDRB	$04 ($24)	DDRB
$16 ($36)	PINB	$03 ($23)	PINB
$15 ($35)	PORTC	$08 ($28)	PORTC
$14 ($34)	DDRC	$07 ($27)	DDRC
$13 ($33)	PINC	$06 ($26)	PINC
$12 ($32)	PORTD	$0B ($2B)	PORTD
$11 ($31)	DDRD	$0A ($2A)	DDRD
$10 ($30)	PIND	$09 ($29)	PIND
$0F ($2F)	SPDR	$2E ($4E)	SPDR
$0E ($2E)	SPSR	$2D ($4D)	SPSR
$0D ($2D)	SPCR	$2C ($4C)	SPCR
$0C ($2C)	UDR	($C6)	UDR0
$0B ($2B)	UCSRA	($C0)	UCSR0A[3]
$0A ($2A)	UCSRB	($C1)	UCSR0B[3]
$09 ($29)	UBRRL	($C4)	UBRR0L
$08 ($28)	ACSR	$30 ($50)	ACSR
$07 ($27)	ADMUX	($7C)	ADMUX
$06 ($26)	ADCSRA	($7A)	ADCSRA

Address in ATmega16/32	Register Name in ATmega16/32	Address in ATmega164P/324P/644P	Register Name in ATmega164P/324P/644P
$05 ($25)	ADCH	($79)	ADCH
$04 ($24)	ADCL	($78)	ADCL
$03 ($23)	TWDR	($BB)	TWDR
$02 ($22)	TWAR	($BA)	TWAR
$01 ($21)	TWSR	($B9)	TWSR
$00 ($20)	TWBR	($B8)	TWBR

Note: 3. Some of the register bits may be located in another register or the bit names are changed, see Table 7-2.

7.2 Bit Definitions

Some bits in ATmega164P/324P/644P have changed name and register location compared to ATmega16/32, other bits have changed location within the register.

Table 7-2. Changes to Bit Names and Locations.

Register in ATmega16/32	Bit Name in ATmega16/32	Register in ATmega164P/324P/644P	Bit Name in ATmega164P/324P/644P
GICR	IVSEL, IVCE	MCUCR	IVSEL, IVCE
TIMSK	OCIE2 TICIE1 OCIE0	TIMSK2 TIMSK1 TIMSK0	OCIE2A ICIE1 OCIE0A
TIFR	TOV2 TOV1 TOV0	TIFR2 TIFR1 TIFR0	TOV2 TOV1 TOV0
TIFR	OCF2 OCF0	TIFR2 TIFR0	OCF2A OCF0A
MCUCSR	JTD ISC2	MCUCR EICRA	JTD ISC20
TCCR0	FOC0 COM01 COM00 CS02 CS01 CS00	TCCR0B TCCR0A TCCR0A TCCR0B TCCR0B TCCR0B	FOC0A COM0A1 COM0A0 CS02 CS01 CS00
SFIOR	PUD PSR2 PSR10	MCUCR GTCCR	PUD PSRASY PSRSYNC
TCCR1A	FOC1A FOC1B	TCCR1C	FOC1A FOC1B
TCCR2	FOC2 COM21 COM20 CS22 CS21 CS20	TCCR2B TCCR2A TCCR2A TCCR2B TCCR2B TCCR2B	FOC2A COM2A1 COM2A0 CS22 CS21 CS20

Register in ATmega16/32	Bit Name in ATmega16/32	Register in ATmega164P/324P/644P	Bit Name in ATmega164P/324P/644P
ASSR	OCR2UB TCR2UB WDTOE	ASSR n/a[1]	OCR2AUB TCR2AUB n/a[1]
UBRRH	URSEL	n/a[1]	n/a[1]
UCRSC	URSEL UMSEL UPM1 UPM0 USBS UCSZ1 UCSZ2 UCPOL	n/a[1] UCSR0C UCSR1C UCSR0C UCSR0C UCSR0C UCSR1C UCSR0C	n/a[1] UMSEL00 UPM10 UPM00 USBS0 UCSZ00 UCSZ10 UCPOL0
UCSRA	RXC TXC UDRE FE DOR PE U2X MPCM	UCSR0A	RXC0 TXC0 UDRE0 FE0 DOR0 UPE0 U2X0 MPCM0
UCSRB	RXCIE TXCIE UDRIE RXEN TXEN UCSZ2 RXB8 TXB8	UCSR0B	RXCIE0 TXCIE0 UDRIE0 RXEN0 TXEN0 UCSZ02 RXB80 TXB80

Note: 1. The function of this bit has a different implementation in ATmega164P/324P/644P. Refer to datasheet for details.

8 Fuse Settings

ATmega164P/324P/644P has four fuse bytes instead of two in ATmega16/32. New features in ATmega164P/324P/644P imply other fuses and fuse settings.

Table 8-1. Comparing Fuses.

	ATmega16/32 Fuse	ATmega16/32 Default Setting	ATmega164P/324P/644P Fuse	ATmega164P/324P/644P Default Setting
Extended Fuse Byte	-	-	BODLEVEL2	1
	-	-	BODLEVEL1	1
			BODLEVEL0	1
Fuse High Byte	-	-	WDTON	1
	CKOPT	1	-	-
Fuse Low Byte	-	-	CKDIV8	0
	BODLEVEL	1	-	-
	BODEN	1	-	-

	ATmega16/32 Fuse	ATmega16/32 Default Setting	ATmega164P/324P/644P Fuse	ATmega164P/324P/644P Default Setting
	-	-	CKOUT	1
	CKSEL1	0	CKSEL1	1
	CKSEL0	1	CKSEL0	0

9 Interrupt Vectors

The ATmega164P/324P/644P has 31 interrupt vectors, located at different addresses than the ATmega16/32, which has 21 interrupt vectors. The ATmega16 and the ATmega32 also have different interrupt tables as shown in Table 9-1.

The additional interrupt vectors are due to extra peripherals not found on ATmega16/32

Table 9-1. Interrupt Table.

Vector #	ATmega16	ATmega32	ATmega164P/324P/644P
1	RESET	RESET	RESET
2	INT0	INT0	INT0
3	INT1	INT1	INT1
4	TIMER2_COMP	INT2	INT2
5	TIMER2_OVF	TIMER2_COMP	PCINT0
6	TIMER1_CAPT	TIMER2_OVF	PCINT1
7	TIMER1_COMPA	TIMER1_CAPT	PCINT2
8	TIMER1_COMPB	TIMER1_COMPA	PCINT3
9	TIMER1_OVF	TIMER1_COMPB	WDT
10	TIMER0_OVF	TIMER1_OVF	TIMER2_COMPA
11	SPI_STC	TIMER0_COMP	TIMER2_COMPB
12	USART_RXC	TIMER0_OVF	TIMER2_OVF
13	USART_UDRE	SPI_STC	TIMER1_CAPT
14	USART_TXC	USART_RXC	TIMER1_COMPA
15	ADC	USART_UDRE	TIMER1_COMPB
16	EE_RDY	USART_TXC	TIMER1_OVF
17	ANA_COMP	ADC	TIMER0_COMPA
18	TWI	EE_RDY	TIMER0_COMPB
19	INT2	ANA_COMP	TIMER0_OVF
20	TIMER0_COMP	TWI	SPI_STC
21	SPM_RDY	SPM_RDY	USART0_RX
22	-	-	USART0_UDRE
23	-	-	USART0_TX
24			ANALOG_COMP
25	-	-	ADC
26	-	-	EE_READY
27	-	-	TWI

Vector #	ATmega16	ATmega32	ATmega164P/324P/644P
28	-	-	SPM_READY
29	-	-	USART1_RX
30	-	-	USART1_UDRE
31	-	-	USART1_TX

10 IEEE 1149.1 (JTAG) Boundary Scan

The boundary scan has changed in ATmega164P/324P/644P where analog circuits no longer constitute a part of the scan chain. The order of the signal names in the boundary scan has also changed. Refer to datasheet for details.

11 Operational Range

Table 11-1. Operating voltage and Speed grades.

	Operating Voltage	Speed Grade
ATmega16/32	4.5-5.5V	0-16 MHz
ATmega16/32L	2.7-5.5V	0-8 MHz
ATmega164P/324P/644P	2.7-5.5V	0-20 MHz
ATmega164PV/324PV/644PV	1.8-5.5V	0-10 MHz

12 Appendix A

Table 12-1 is a selection of crystals that meet the ESR requirements of the ATmega164P/324P/644P. The crystals are listed based on datasheet information and are not tested with the actual device. Any other crystal that complies with the ESR requirements can also be used. Availability and RoHS compliance has not been investigated.

Table 12-1. Examples of crystals compliant with ATmega164P/324P/644P low-frequency Crystal Oscillator.

Vendor	Type	Mounting (SMD/HOLE)	Frequency Tolerance [±ppm]	Load Capacitance [pF]	Equivalent Series Resistance (ESR) [kΩ]
C-MAC	WATCH CRYSTALS	HOLE	20	6	50
C-MAC	85SMX	SMD	20	6	55
C-MAC	90SMX	SMD	20	6	60
ECLIPTEK	E4WC	HOLE	20	6	50
ENDRICH	90SMX	SMD	5	6	50
EPSON	C-001R	HOLE	20	6 -> 12.5 (specify)	35
EPSON	C-002RX	HOLE	20	6 -> 10 (specify)	50
EPSON	C-004R	HOLE	20	6 -> 10 (specify)	50
EPSON	C-005R	HOLE	20	6 -> 10 (specify)	50
EPSON	MC-30A	SMD	20	6 -> 10 (specify)	50
EPSON	MC-306	SMD	20	6 -> 10 (specify)	50
EPSON	MC-405	SMD	20	6 -> 10 (specify)	50

Vendor	Type	Mounting (SMD/HOLE)	Frequency Tolerance [±ppm]	Load Capacitance [pF]	Equivalent Series Resistance (ESR) [kΩ]
EPSON	MC-406	SMD	20	6 -> 10 (specify)	50
GOLLEDGE	GWX	HOLE	5	6, 8 or 12.5	35
GOLLEDGE	GSWX-26	SMD	10	6 , 8 or 12.5	35
GOLLEDGE	GDX1	HOLE	10	6	42
GOLLEDGE	GSX-200	SMD	5	6	50
IQD	WATCH CRYSTALS	HOLE	20	6	50
IQD	90SMX	HOLE	10	6	60
IQD	91SMX	HOLE	10	6	60
MICROCRYSTAL	MS3V-T1R	HOLE	20	7 or 9	65
MICROCRYSTAL	MS2V-T1R	HOLE	20	7 or 9	65
MICROCRYSTAL	CC4V-T1A	SMD	30	9	65
MICROCRYSTAL	CC1V-T1A	SMD	30	9	60
MICROCRYSTAL	CC7V-T1A	SMD	30	9	70
MMD	WC23	HOLE	8	8	35
MMD	WC33	HOLE	8	8	35
MMD	WC155	HOLE	8	8	40
MMD	WCSMC	SMD	20	6	50
OSCILENT	SERIES 111	HOLE	10	6 or 12.5	30
OSCILENT	SERIES 112	HOLE	10	6 or 12.5	40
OSCILENT	SERIES 113	HOLE	10	8	40
OSCILENT	SERIES 223	SMD	20	6	50
RALTRON	SERIES R38	HOLE	5	6 or 12.5	35
RALTRON	SERIES R20	HOLE	5	6 or 12.5	30
RALTRON	SERIES R145	HOLE	5	8	40
RALTRON	SERIES RSE A, B, C, D	SMD	20	6	50
SBTRON	SBX-13	SMD	20	6	50
SBTRON	SBX-20	SMD	20	6	50
SBTRON	SBX-21	SMD	20	6	50
SBTRON	SBX-24	SMD	20	6	50
SBTRON	SBX-23	SMD	20	6	50
SBTRON	SBX-22	SMD	20	6	50
SBTRON	SBX-14	HOLE	20	6	50
SUNTSU	SCT1	HOLE	20	6, 8, 10 or 12.5	40
SUNTSU	SCT2	HOLE	20	6, 8, 10	50
SUNTSU	SCT3	HOLE	20	6, 8, 10	50
SUNTSU	SCP1	SMD	20	6	50
SUNTSU	SCT2G	SMD	20	6 or 10	50

References

REFERENCES

BOOK REFERENCES

1. "Location-based Services Fundamentals and Operation" by Axel Küpper published by John Wiley & Sons, October 2005.
2. "Outdoor Navigation with GPS" by Stephen W. Hinch, April 2004.

IEEE PARER REERENCES

1. GPS-GSM Integration for Enhancing Public Transportation Management Services by Farooq, U.; ul Haq, T.; Amar, M.; Asad, M.U.; Iqbal, A.; Dept. of Electr. Eng., Univ. of The Punjab, Lahore, Pakistan

2. Efficient Cost-Based Tracking of Scheduled Vehicle Journeys
Tiesyte, D.; Jensen, C.S.; Dept. of Comput. Sci., Aalborg Univ., Aalborg

WEBSITE REFERENCES

www.alldatasheet.com/

http://www.gpsinformation.org/dale/nmea.htm

http://en.wikipedia.org/wiki/NMEA_0183

http://www.howstuffworks.com/lcd.htm

http://www.developershome.com/sms/atCommandsIntro.asp

http://wiki.forum.nokia.com/index.php/AT_Commands

http://en.wikipedia.org/wiki/SQL

http://www.w3schools.com/sql/sql_intro.asp

http://en.wikipedia.org/wiki/GSM

http://www.atmel.com/dyn/resources/prod_documents/doc7674.pdf

http://www8.garmin.com/aboutGPS/