OnJin
ENGINEERING

87 Alde Ave
Richmond Hill, Ontario
L43 4N2
ensc440-onjin@sfu.ca

Monday, March 9th, 2006

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specifications for the GeoPreserver™

Dear Dr. Rawicz,

The attached document, *Design Specifications for the GeoPreserver™*, specifies the design of the GeoPreserver's components. It covers both the hardware and software aspects, and their interrelationships. With the information in this document, we can begin the prototyping of our device.

Our project team consists of five talented and innovative engineers with solid industrial experience. The team members include Jason Lee, Bryan Friesen, Jason Cho, Will Chan and Jeffrey Huang. Adam Smith of Innovative Technologies will work as a Senior Advisor and Consultant with our project. If you have any questions, please feel free to contact us at ensc440-onjin@sfu.ca.

Sincerely,

Jason Lee
Chief Executive Officer
OnJin Engineering

Enclosure: Design Specifications for the GeoPreserver™

OnJin
ENGINEERING

Design Specifications for the GeoPreserver

Classification: CONFIDENTIAL

Project Team:
Jason Lee
Jeff Huang
Bryan Friesen
Will Chan
Jason Cho

Contact Person:
Jason Lee
ensc440-onjin@sfu.ca

Submitted To:
Dr. Andrew Rawicz
Steve Whitmore
School of Engineering Science
Simon Fraser University

Issuing Date: March 9, 2006

Revision: 1.0

Executive Summary

The commercial fishing industry is one of the most hazardous industries in North America. The Alaskan crab fisheries alone exhibit estimated fatality rates of 111/100,000 workers per year. According to the Alaska Department of Labor, more than 80 percent of these fatalities are due to drowning either from falling overboard or as a result of a boat accident.

The GeoPreserver™ provides a viable tool for combating this fatal problem in the fishing industry. The purpose of the project is to develop a system that monitors and tracks all of the crew on a fishing vessel. The basis of the system is a geofence, which involves defining a boundary that encompasses the boat using GPS navigation. As long as the boat's crew remains within this boundary, they are in the "safe" zone, but once they exit, the system will be alerted.

The development of the prototype will consist of a base station and a personal tracking unit. The prototype's main purpose is to show the functionality of our project. This document outlines, in detail, the inner working of the GeoPreserver™ prototype. The prototype version will focus on the following features:

1. One or Two personal units with the capabilities of tracking an individual and interfacing with the base station.
2. The capabilities to show the functionality of our project such as tracking, alarming when the personnel exits the boundary
3. A base station with an external interface such as a PC
4. An easy to use interface through a PC

This design specification shows in detail the technical makeup of the GeoPreserver™. The document is intended as a reference for the construction of the prototype. The tables, figures and descriptions presented in the following pages will clearly illustrate our methods of meeting our functional specifications.

We hope that this document will show the hard work and thought process involved in designing the GeoPreserver™. By early April, we expect to have a functioning prototype which will demonstrate the product's ability to save the lives of drowning fishermen

Table of Contents

List of Figures

List of Tables

Glossary

DSSS Direct Sequence Spread Spectrum

EEPROM Electronically Erasable Programmable Read Only Memory

EUSART Enhanced USART that included added features such as auto-detecting baud rate.

Geofence A virtual fence or geographical boundary used to trigger events once the fence boundary is crossed.

GPRMC NMEA protocol packet format providing the Recommended Minimum Specific data required for GPS applications.

GPS GPS stands for Global Positioning System. It consists of 24 satellites controlled by the United States Department of Defense for the purpose of navigation and positioning.

IPX7 An IPX7 designation means the GPS case can withstand accidental immersion in one meter of water for up to 30 minutes.

MOB MOB stands for ‘man overboard’, signifying that a person has fallen off the ship into the water.

NMEA National Marine Electronics Association.

SPI Serial Peripheral Interface. A loosely defined serial transmission line that has synchronized data transmission intervals.

UART Universal Asynchronous Receive Transmit. Serial transmission line that is self-synchronizing so that data can be sent at any time interval.

USART A customizable UART that can function as a synchronous transmission line as well as an asynchronous line.

UTC Time Coordinated Universal Time, taking into account the addition or omission of leap seconds by atomic clocks each year to compensate for changes in the rotation of the earth.

Revision History

Revision	Status	Publication/Revision Date	By:
1.0	• Created	Monday, March 9, 2006	Jeff Huang

1. Introduction

The GeoPreserver™ is a new and innovative product intended to help save the lives of potentially drowning fishermen. The system consists of a single base station and multiple personal units. The base station constantly monitors the location of the personal units relative to its own position. When a crew member wearing a personal unit falls overboard, the base station alerts another crew member inside the boat. The base station is used to display the location of the fallen crew member to aide their recovery.

To reduce high fatality rates in the fishing industry, the GeoPreserver™ will provide more efficient and effective method of saving lives. The development of this project consists of multiple stages, including the completion of the prototype to illustrate its functionality by April 2006.

This document will serve as a guide in the design of our prototype. It illustrates the technical details required for the implementation of the prototype as well as the software aspects. Each component of the system is reviewed in detail to reduce implementation time.

2. System Overview

The GeoPreserver™ will consist of two distinct modules, a Base station, and Personal Units. The Base Station will be mounted inside the bridge of the boat and will serve as the reference point for establishing the geofence and determining when a crew member is in the water. The Personal Units will be worn by everyone on the boat at all times. These units will wirelessly transmit their geographical location back to the base station. The base station will then determine if the person is on the boat or over. If over, the system will be alerted, thus indicating a man overboard. The figure below illustrates the relationships.

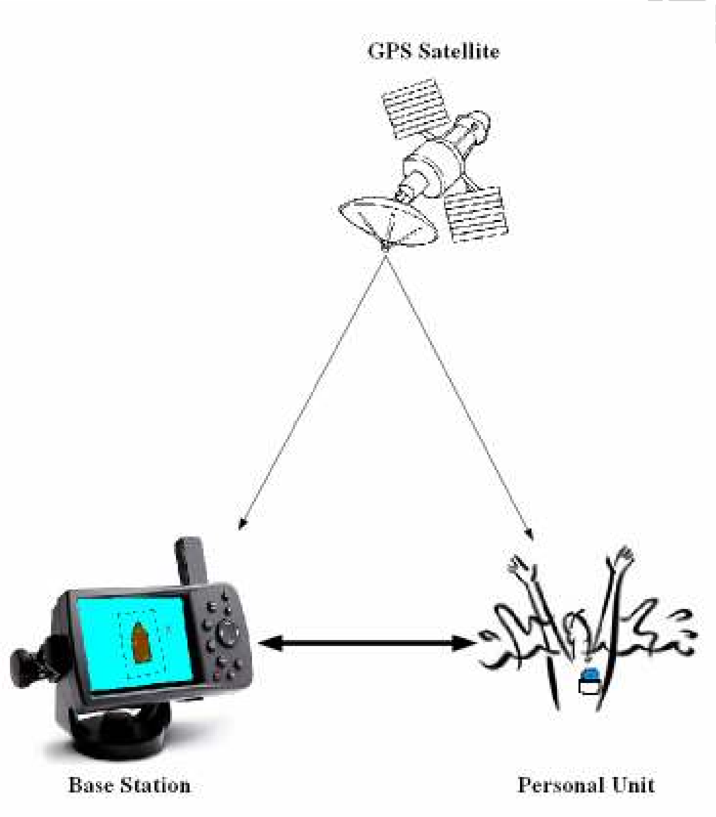


Figure 1: System Overview Diagram.

3. Hardware

3.1. Hardware Overview

The following sections provide a detailed description of our three main hardware components, namely the GPS module, the RF transceiver, and the micro-controller. In each section, we discuss each part's main features, why the part was chosen, its setup requirements, and how it interfaces with our other components.

Our system consists of two distinct components, a Base Station, and Personal Units. A block diagram for each of our system components are shown below.

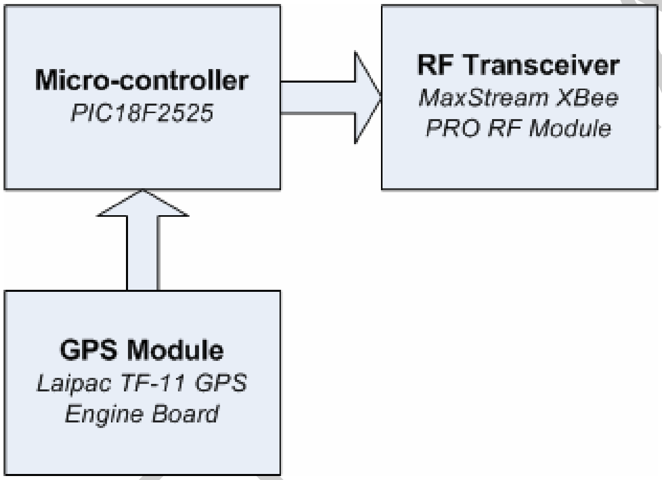


Figure 2: Personal Unit Block Diagram.

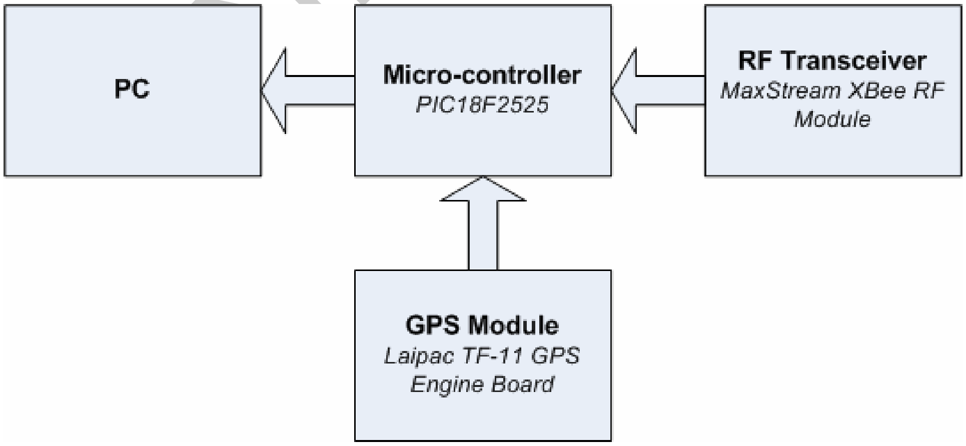


Figure 3: Base Station Block Diagram.

As shown in Figure 2, the Personal Unit is composed of a GPS Module, a micro-controller, and a RF Transceiver. The GPS module will receive data from a satellite and will be passed onto a micro-controller. The micro-controller will then pass on the data to the RF transceiver, which will be transmitted to the base-station. The base-station is composed of a GPS Module, a micro-controller, a RF transceiver, and a PC. The base-station will receive data from the personal unit with its RF transceiver, and pass it onto the base-station's micro-controller. This micro-controller will also take data received by the GPS module. All the data received by the micro-controller will be sent to the PC, where the data is interpreted to update the GUI. In order to maintain global consistency among units, all components will be configured to receive and transmit at 4800 baud. This is to prevent conflicts of configuration between components.

3.2. Laipac TF-11 GPS Engine Board

The Laipac TF-11 GPS Engine Board will supply the GPS data for the GeoPreserver™. The TF-11 is a small, low powered device that has the ability to track up to 12 satellites for increased accuracy and dependability. Communication to and from the TF-11 is will be accomplished using a standard UART connection. This makes it easy for us to interface with the other components in our system, which will be covered in later sections. Other key specifications of this module include:

- Horizontal position accurate to within 10m
- Velocity accurate to 0.1m/sec
- Fast GPS acquisition rate
- Operation temperature from -40 deg. C to +85 deg. C

The GPS module can be configured to output data in many different formats. The TF-11 series is fully compliant with the SiRF binary and the NMEA-0183 protocols. Since our only requirements are the location and speed of each unit, we will use the NMEA-0183 GPRMC protocol. This protocol delivers the recommended minimum specific data that we need for our product, which includes the Latitude, Longitude, and Velocity. A sample string of GPRMC data appears as follows, accompanied by a table explaining its breakdown.

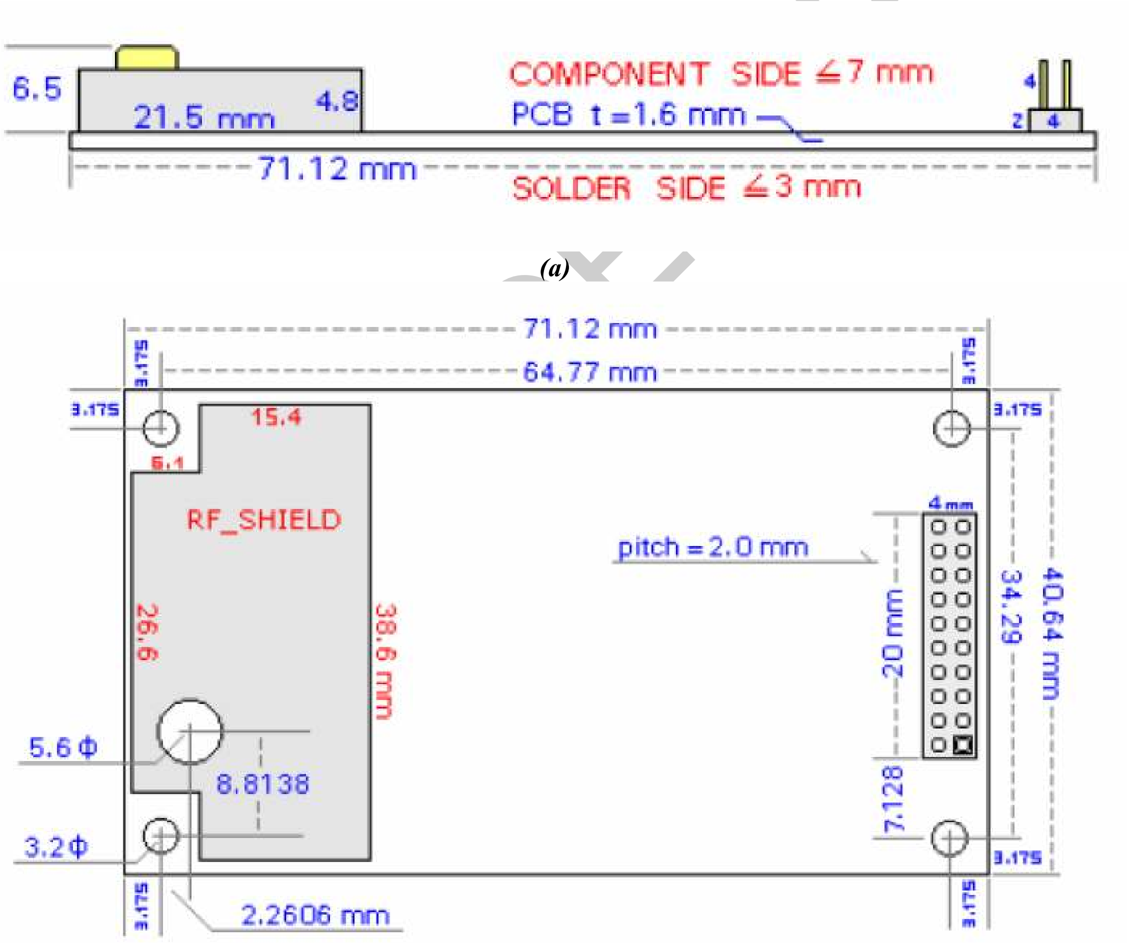
```
$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10
```

Table 1: Breakdown of a NMEA GPRMC String.

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Time	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.mmm.
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]	N/A	Degrees	E = east of W = west
Checksum	*10		Error Check
<CR><LF>			End of message termination

[1] Sirf Technology Inc. does not support magnetic declination. All "course over ground" data are geodetic WGS48 directions

A mechanical layout of the TF-11 engine board is shown in Figure 4 below. One point of interest from these diagrams is that the header has 2mm pitch. Since most breadboards use standard 0.1" spacing, a design solution will need to be developed for prototyping. One possible solution will be to find a DIP connector that can convert from 2mm to 0.1" spacing. So far, we have not been able to locate any sockets that will do this. Another possible solution will be to create a small custom PCB that runs lines from the header out to a standard spacing. The PCB layout would resemble Figure 5 below, where the inner pads are spaced at 2mm and the outer pads are spaced at 0.1". The last solution would be to solder wire directly to the header, and run them onto our protoboard. This solution may appear ugly, but it may end up being the fastest and simplest solution.



(a)

(b)

Figure 4: Mechanical Layout of the Laipac GPS Module.

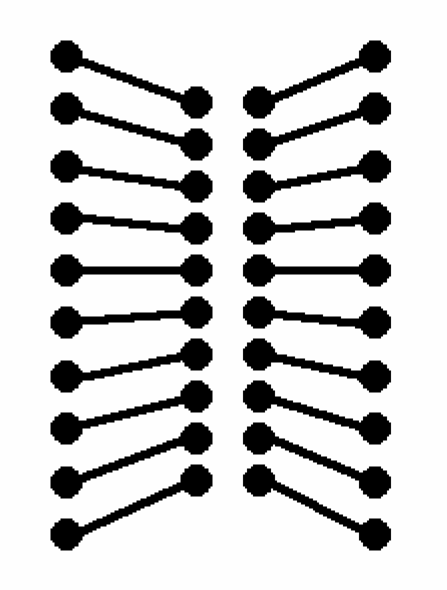
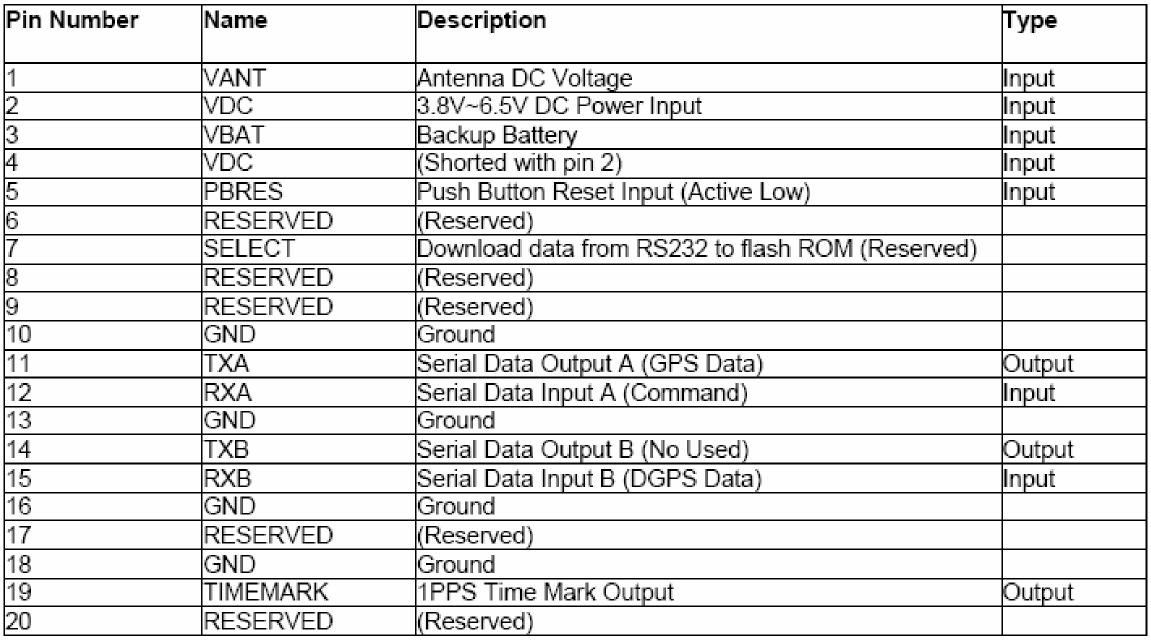


Figure 5: 2mm to 0.1" Converter.

Table 2 below shows the pin-out information for the TF-11. Pins 6, 7, 8, 9, 14, 15, 17, 19, and 20 are not required and will not be connected. The other pins will be connected as required.

Table 2: TF-11 Pinout Description.



Pin Number	Name	Description	Type
1	VANT	Antenna DC Voltage	Input
2	VDC	3.8V~6.5V DC Power Input	Input
3	VBAT	Backup Battery	Input
4	VDC	(Shorted with pin 2)	Input
5	PBRES	Push Button Reset Input (Active Low)	Input
6	RESERVED	(Reserved)	
7	SELECT	Download data from RS232 to flash ROM (Reserved)	
8	RESERVED	(Reserved)	
9	RESERVED	(Reserved)	
10	GND	Ground	
11	TXA	Serial Data Output A (GPS Data)	Output
12	RXA	Serial Data Input A (Command)	Input
13	GND	Ground	
14	TXB	Serial Data Output B (No Used)	Output
15	RXB	Serial Data Input B (DGPS Data)	Input
16	GND	Ground	
17	RESERVED	(Reserved)	
18	GND	Ground	
19	TIMEMARK	1PPS Time Mark Output	Output
20	RESERVED	(Reserved)	

The GPS receiver will be configured using the SiRF Demo software. Using this software, we can change the default setting of the TF-11 to match the requirements specified in the functional specifications. The TF-11 will be configured to operate in trickle power mode.

This special low power mode enables the TF-11 to output data at a configurable interval. Once data has been sent, the TF-11 goes into a low power mode and waits until for the next interval. Using this software, we will also configure the GPS to operate at a baud rate of 4800 bps. While the TF-11 is operating, sending ASCII command instructions to the Rx pin can change the configuration settings. These commands are explained in more detail in the TF-11 data sheet. The SiRF Demo software also will provide a means of verifying that the GPS receivers are functioning correctly and are able to receive GPS coordinates.

3.3. RF Transceiver

A transceiver is required for the base station and each of the personal units. The transceiver is used to transmit and receive data between the base station and the personal units. For our prototype, we will be utilizing both the Maxstream XBee ZigBee OEM RF Module, and the Maxstream XBee ZigBee Pro OEM RF Module. The base station will use the XBee ZigBee RF Module, and the individual units will use the XBee ZigBee Pro RF Modules; however, these modules can be interchangeably used. The XBee ZigBee Pro module has extended range over the non-Pro version, and it operates over a wider temperature range, making it more suitable for the Personal Units.

Major factors which led to our choice of transceiver include its size, cost and power consumption. Our personal units' portability required that our transceiver be small and power efficient with sleep capability. With the XBee modules, we were able to achieve these requirements and more. In Figure 6 and Figure 7 below, we show the mechanical drawings of the modules. We also note that the modules have a 2.00mm pitch.

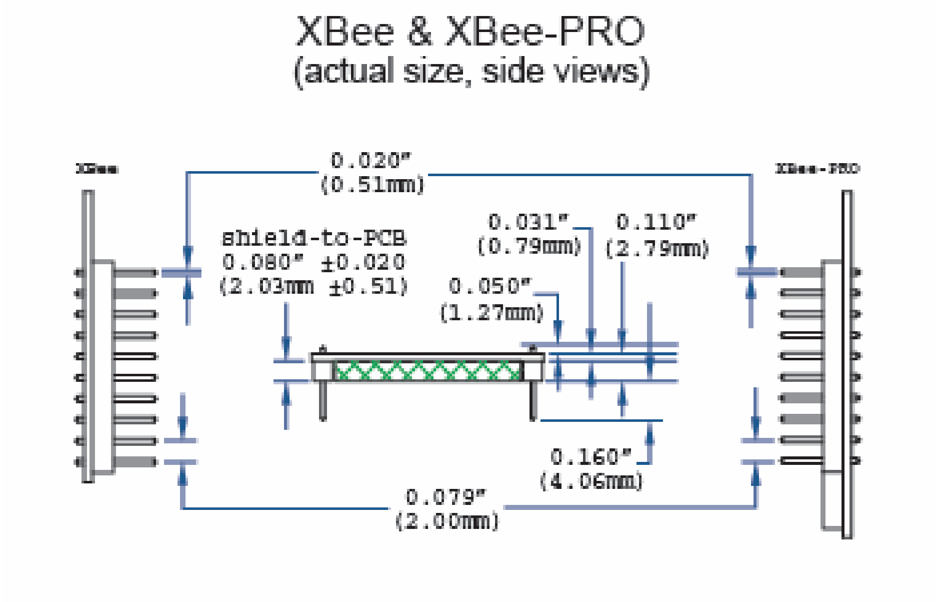


Figure 6: Side Views of the XBee and XBee Pro Modules.

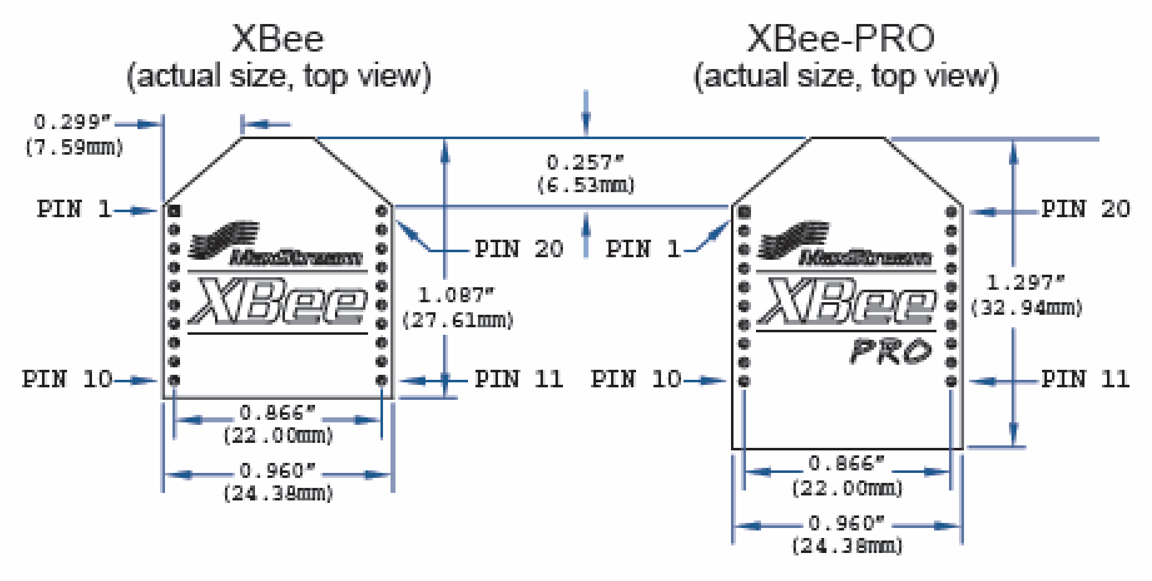


Figure 7: Top View of the XBee and XBee Pro RF Modules.

The XBee Modules operate within the ISM 2.4 GHz frequency. It is advantageous because the 2.4GHz frequency band is an unlicensed radio band worldwide, so the user will be able to operate the units anywhere in the world without a radio license.

Antennas are required to work at the 2.4GHz frequency with our modules. The modules offer three types of antenna options, namely U.FL RF connector, chip antenna, or whip antenna. We chose antennas which use a RPSMA connector. To have this work with our modules, we needed a U.FL to SMA converter.

The XBee modules feature advanced networking capabilities. They offer three topologies: peer-to-peer, point-to-point, and point-to-multipoint. This allows our base station and our personal units to transmit to all the other personal units within range at the same time. Personal units can also transmit the data to another unit, and that unit can relay the message to the base station, which is a huge advantage because it increases our coverage area, and it allows multiple paths for the data to travel in order to reach its destination. The figures below illustrate how the point-to-point and point-to-multipoint topology works.

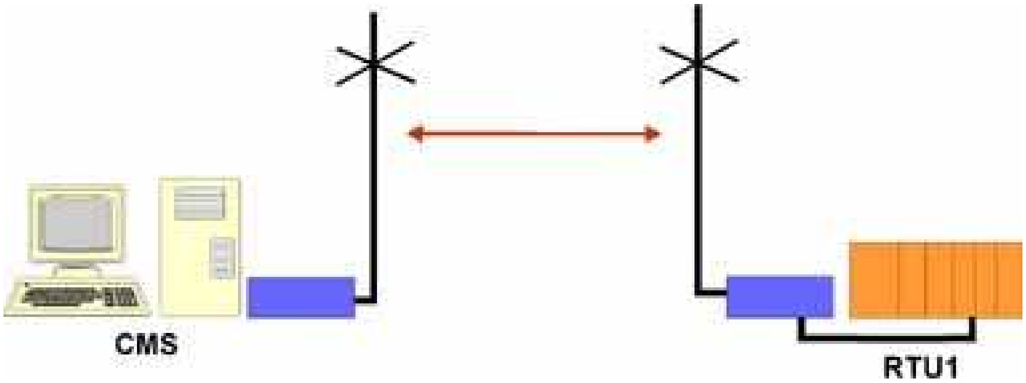


Figure 8: Point-to-Point Topology.

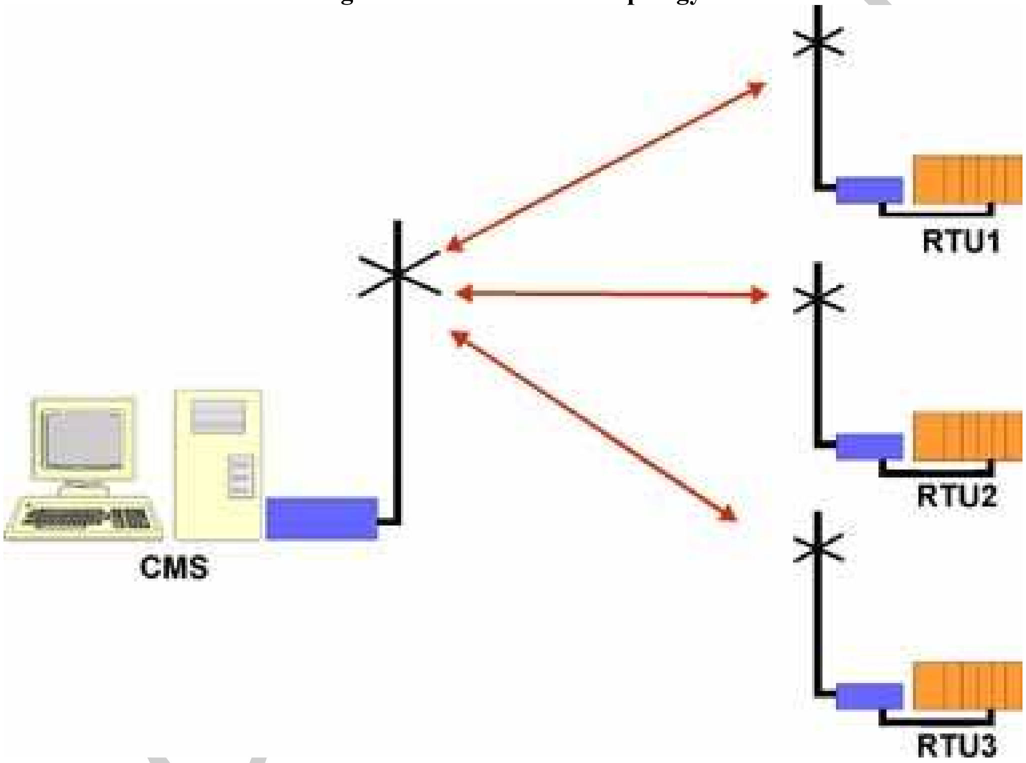


Figure 9: Point-to-Multipoint Topology.

The modules use a modulation approached called Direct Sequence Spread Spectrum (DSSS), which splits the data signal according to a spreading ratio and spreads the pieces along a frequency channel spectrum. This is done by combining the data signal at the point of transmission with a higher data rate bit sequence known as a chipping code. The sequence of bits is just a pattern of repeating bits for each bit of the transmitted signal. This process is sometimes known as multiplying the data signal with a "noise" signal, which has a higher frequency than the data signal. To filter out the data signal at the receiver end, the bit sequence is multiplied by the same bit sequence. This method protects the transmitted signal from interference. If any of the bits transmitted become damaged, recovery of this data can be achieved because of the redundancy of the transmission.

Both the XBee and the XBee Pro have high RF data rate, and interface data rate. RF data is capable of being transmitted at a rate of 250Kbps and interface data is capable of being transmitted up to a rate of 115.2Kbps. The modules allow us to interface with other devices, such as a micro-controller through its UART pins. The interfacing details will be provided later. With the high RF data rates, we will be able to receive and send data out quickly; however, the limiting factor is the interface transfer rate, which is much slower due to the micro-controller's limitations.

The XBee Pro has a 100mW output power, capable of transmitting up to a 1mile range, and the regular XBee has a 1mW output power, capable of transmitting up to a 300meter range.

3.3.1. Setup Requirements

The XBee Modules will be configured using the X-CTU Software provided by MaxStream. This program allows us to modify the default settings of the modules. The modules need to be in command mode, which is a state in which the modules interpret incoming characters as commands. A table of the XBee and XBee-PRO commands is shown below.

Table 3: XBee/XBee-PRO Commands.

AT Command	Command Category	Name and Description	Parameter Range	Default
BD	Serial Interfacing	**Interface Data Rate**. Set/Read the serial interface data rate for communications between the RF module serial port and host.	0 – 7 (custom rates also supported)	3
CC	AT Command Mode Options	**Command Sequence Character**. Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT+CC+GT). The AT Command Mode Sequence enters the RF module to AT Command Mode.	0 – 0xFF	0x2B ('+' ASCII)
CH	Networking & Security	**Channel**. Set/Read the channel number used for transmitting and receiving between RF modules. Uses 802.15.4 protocol channel numbers.	0x0B – 0x1A (XBee) 0x0C – 0x18 (XBee-PRO)	0x0C (12d)
CN	AT Command Mode Options	**Exit Command Mode**. Explicitly exit AT Command Mode.	-	-
CT	AT Command Mode Options	**Command Mode Timeout**. Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	2 – 0xFFFF [x 100 ms]	0x64 (100d)
DB	Diagnostics	**Received Signal Strength**. Read signal level [in dB] of last good packet received (RSSI). Absolute value is reported. (For example: 0x58 = -88 dBm) Reported value is accurate between -40 dBm and RX sensitivity.	0 – 0x64 [read-only]	-
DH	Networking & Security	**Destination Address High**. Set/Read the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 – 0xFFFFFFFF	0
DL	Networking & Security	**Destination Address Low**. Set/Read the lower 32 bits of the 64-bit destination address. When combined with DH, it defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 – 0xFFFFFFFF	0

GT	AT Command Mode Options	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT+CC+GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	0x02 – 0xFFFF [x 1 ms]	0x3E80 (1000d)
ID	Networking & Security	**PAN ID**. Set/Read the PAN (Personal Area Network) ID. 0xFFFF indicates a message for all PANs.	0xFFFF	0x3332 (13106d)
MY	Networking & Security	**16-bit Source Address**. Set/Read the RF module 16-bit source address. Set MY = 0xFFFF to disable reception of packets with 16-bit addresses. 64-bit source address (serial number) and broadcast address (0x000000000000FFFF) is always enabled.	0 – 0xFFFF	0
P0	Diagnostics	**PWM0 Configurations**. Select/Read function for PWM0.	0 – 1	1
PL	RF Interfacing	**Power Level**. Select/Read power level at which the RF module transmits.	0 – 4	4
RE	(Special)	**Restore Defaults**. Restore RF module parameters to factory defaults. Follow with WR command to save values to non-volatile memory.	-	-
RN	Networking & Security	**Random Delay Slots**. Set/Read the minimum value of the back-off exponent in the CSMA-CA algorithm that is used for collision avoidance. If RN = 0, collision avoidance is disabled during the first iteration of the algorithm (802.15.4 - macMinBE).	0 – 3	0
RO	Serial Interfacing	**Packetization Timeout**. Set/Read number of character times of inter-character delay required before transmission. Set to zero to transmit characters as they arrive instead of buffering them into one RF packet.	0 – 0xFF [x character times]	3
RP	Diagnostics	**RSSI PWM Timer**. Enable a PWM (pulse width modulation) output (on pin 6 of the RF modules) which shows RX signal strength.	0 – 0xFF [x 100 ms]	0x28 (40d)
SH	Diagnostics	**Serial Number High**. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 – 0xFFFFFFFF [read-only]	Factory-set
SL	Diagnostics	**Serial Number Low**. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 – 0xFFFFFFFF [read-only]	Factory-set
SM	Sleep (Low Power)	**Sleep Mode**. Set/Read Sleep Mode configurations.	0 – 6	0
SP	Sleep (Low Power)	**Cyclic Sleep Period**. Set/Read sleep period for cyclic sleeping remotes. Maximum sleep period is 268 seconds (0x68B0).	0x01 – 0x68B0 [x 10 ms]	0x64 (100d)
ST	Sleep (Low Power)	**Time before Sleep**. Set/Read time period of inactivity (no serial or RF data is sent or received) before activating Sleep Mode. The ST parameter is only valid with Cyclic Sleep settings (SM = 4 - 6). Set ST on Cyclic Sleep Coordinator to match Cyclic Sleep Remotes.	0x01 – 0xFFFF [x 1 ms]	0x1388 (5000d)
VR	Diagnostics	**Firmware Version**. Read firmware version of the RF module.	0 – 0xFFFF [read-only]	Factory-set
WR	(Special)	**Write**. Write parameter values to RF module's non-volatile memory so that modifications persist through subsequent power-up or reset.	-	-

We need to set up the source address for each module, which is the address we want to have assigned to the device. We also need to set up the destination address, which is the address of the device we want to transmit to. The figure below shows the breakdown of an example command that would be sent to the module.

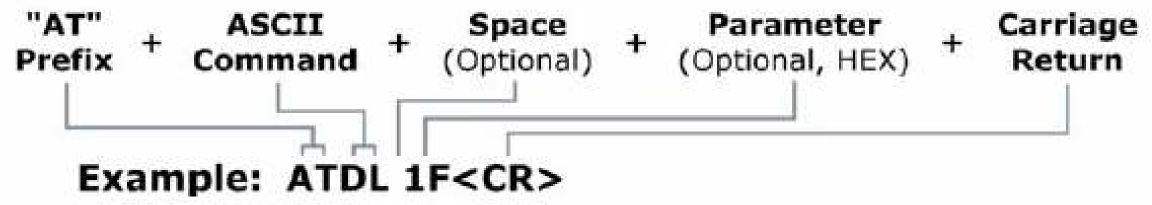


Figure 10: XBee Example Command Breakdown.

To communicate with the microcontroller, the baud rate will be set at 4800 bps.

3.4. PIC18F2525 Micro-controller

Microchip offers a wide variety of micro-controllers for a number of different applications The PIC18F2525 offers a low cost, low power solution to satisfy the needs of our functional requirements.

3.4.1. Features

The PIC highlights some key features that will be extremely useful in our implementation. The micro-controller has two onboard serial connections; one UART and one serial peripheral interface (SPI) port that are multiplexed on Port C. It also includes four timers crucial for synchronizing with our other components and a flexible oscillator structure that can use both an internal or external clock. For the purpose of timing, the PIC will act as a master in synchronizing the devices. The micro-controller offers adjustable baud rates over each serial line to match with the external components. Furthermore the internal oscillator can be configured based on an internal crystal oscillator or an external source. Figure 11 illustrates the pin diagram of our PIC.

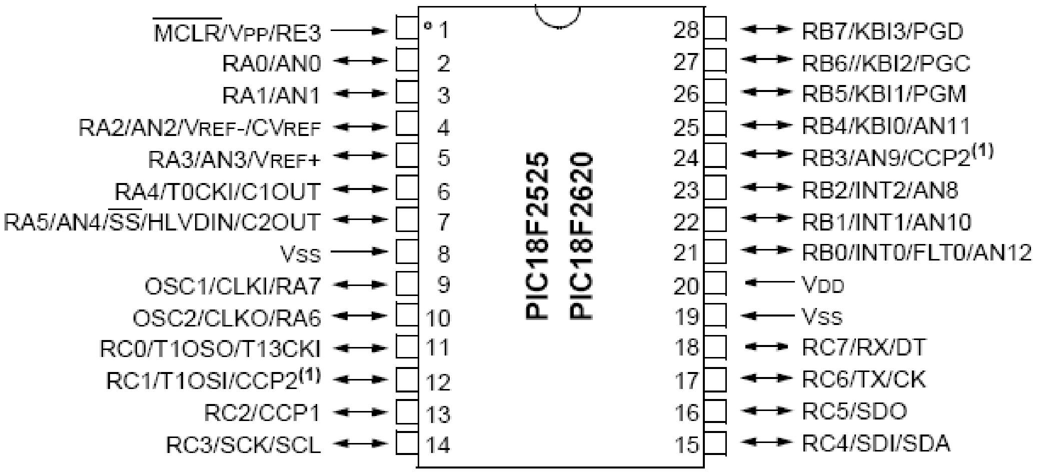


Figure 11: Pinout Diagram for the PIC18F2525 Micro-controller.

The purpose of the micro-controller is to act as a data management unit, which receives and forwards data to individual components.

3.4.2. Interface

In regards to the serial transmission, the PIC has onboard peripherals that have the capabilities of establishing serial connections with external devices.

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripherals. In our application, the serial port will be used to interface with the PC, which will be further explained in a later section. The MSSP will be configured as a SPI bus on Port C using the SDO, and SDI pins. Figure 12 illustrates the SPI port on the micro-controller. By configuring the registers associated with each pin, the SPI can be initialized.

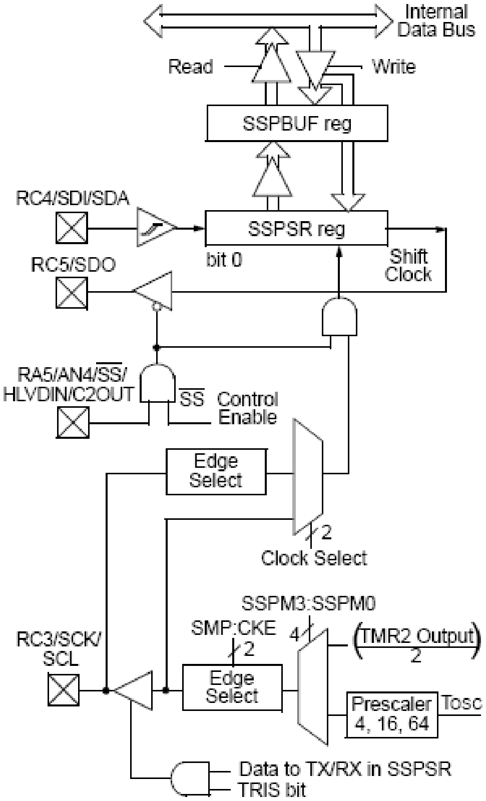


Figure 12: SPI Port Block Diagram.

The Enhanced Addressable USART module is capable of standard RS-232 operation and provides support for LIN bus protocol. Other enhancements include automatic baud rate detection and a 16-bit baud rate generator for improved resolution. The USART provides extremely stable operation with applications using external modules. In our application, the USART will be used to interface with the GPS receiver and RF transceiver. Figure 13 and Figure 14 shows the block diagrams for the EUSART transmit and receive modules, respectively.

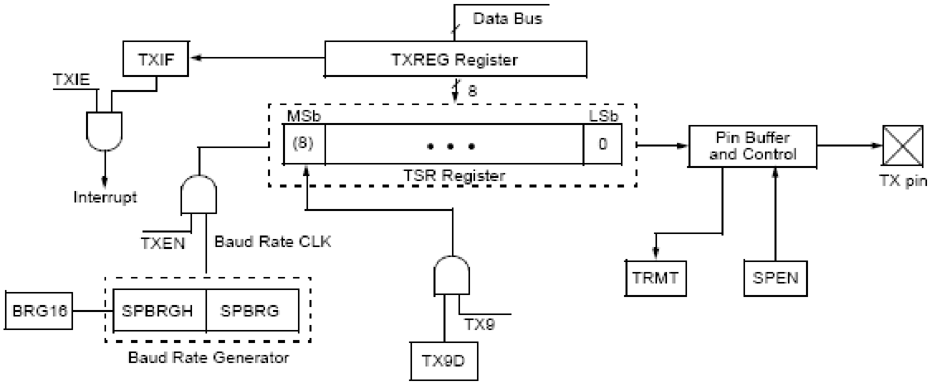


Figure 13: EUSART Transmit Module Block Diagram.

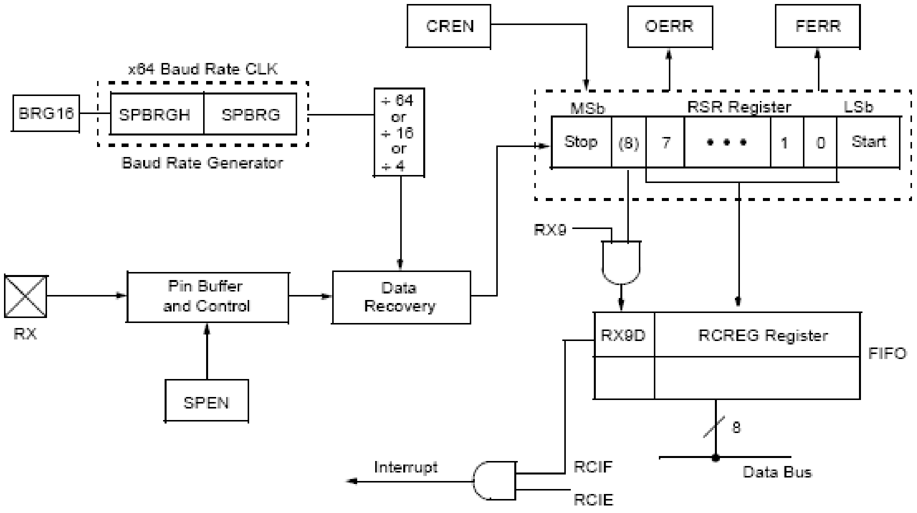


Figure 14: EUSART Receive Module Block Diagram.

The Tx and Rx pins transmit and receive individual bits that are reassembled by the micro-controller. Since there are multiple units sharing the EUSART module, an external multiplexer will be added to the system that will select the GPS and transceiver. The address select line will be tied in with AN0, and AN1 on the micro-controller to select the address of the component. The associated registers are configured for our specific application.

The PIC also offers numerous general I/O pins that will be used as our address select lines for multiplexing, and other various needs that are required by our functionality specified in our functional specifications.

3.4.3. Component Functionality

PIC18F2525 was the micro-controller of choice due to its versatility and capabilities. It offers lower power consumption crucial in our mobile applications and includes all important features that are required of our functionality. It was a requirement that our micro-controller is capable of connecting to multiple serial devices, and the availability of two serial connections satisfies these requirements. Stringent timing requirements are also satisfied by our choice of micro-controller.

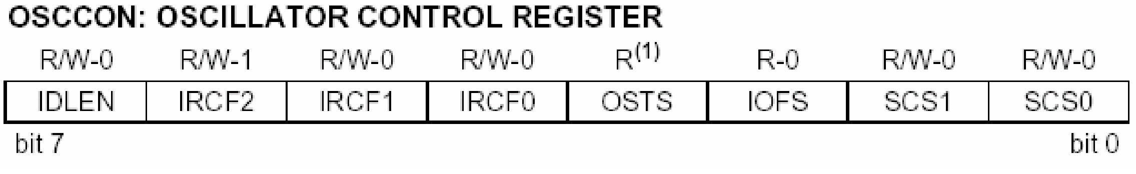
The micro-controller effectively acts as a data management unit, receiving and retransmitting data to their desired location. The specific interfaces will be explained in a later section. Using the available interfaces and peripherals, the PIC18F2525 offers the best solution possible. Furthermore with a large onboard 48Kbyte flash memory, there is ample space to write our desired code. Also, the PIC offers an onboard 1024 bytes of EEPROM to store our buffered serial data that will be further transmitted. With these crucial features, the micro-controller will be capable of satisfying our requirements.

3.4.4. Setup Requirements

The PIC requires minimal setup to operate properly in our system. Due to the simplicity of the micro-controller firmware, there are few algorithms required for the interfacing of individual components. The analysis of the algorithms to act as a data management unit between the modules will be explained in a later section.

MPLAB IDE, and ICD 2 will be used to program and debug the micro-controller. It is possible to use the included c18 compiler or program the micro-controller using assembly. The debugger (ICD 2) is capable of step-by-step progression through code, allowing for real time debugging of the micro-controller.

The clock and serial ports will need to be configured on the micro-controller before it is used. The internal crystal oscillator will be used and set at 8MHz supplied by the main output (INTOSC). The clock is set by setting the OSSCON register with the IRCF2:IRCF0 to be 111 = 8MHz. Figure 15 illustrates the OSSCON register and its associated register.

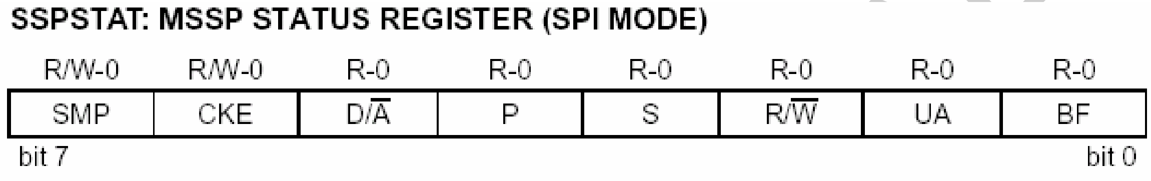


OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R(1)	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

Figure 15: OSSCON Register.

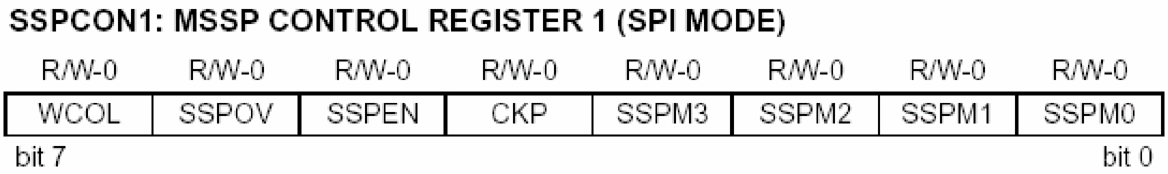
The serial peripheral interface and EUSART module also needs to be configured for our operation. The SPI will be implemented with the Serial Data Out, Serial Data In, Serial Clock and Slave Select pins. There are two associated registers to the serial peripheral interface mode: the status register (SSPSTAT), and control register (SSPCON1). The data interface will be implemented with the serial receive/transmit buffer register (SSPBUF) and MSSP shift register (SSPSR). The SSPSR and SSPBUF create a double-buffered receive/transmit with the SSPSR shifting data, and the SSPBUF directly accessible to read and write data. The registers must be configured appropriately to enable communication between the PC. Figure 16 and Figure 17 illustrate the SSPSTAT and SSPCON1 registers.



SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	$D/\overline{A}$	P	S	$R/\overline{W}$	UA	BF
bit 7							bit 0

Figure 16: SSPSTATE Register.

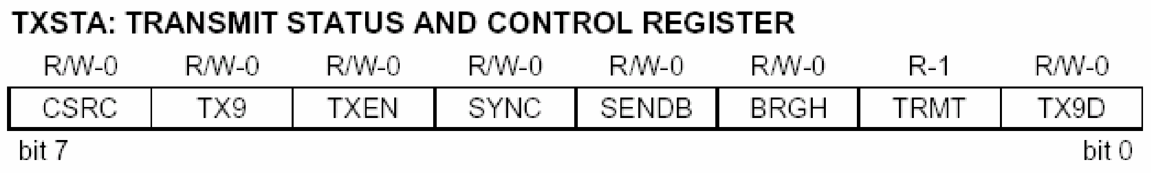


SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

Figure 17: SSPCON1 Register.

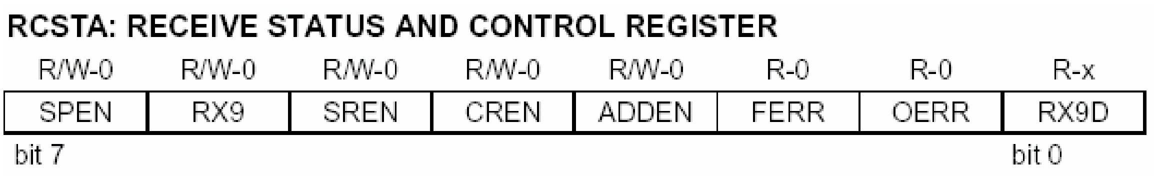
To configure the Enhanced USART, RC6/TX/CK and RC7/RX/DT need to be configured as a USART. This is done by setting the appropriate bits on the SPEN and TRISC configuration registers. The USART module is controlled through three registers. Transmit Status and Control (TXSTA), Receive Status and Control (RCSTA), and Baud Rate Control (BAUDCON). The important setup requirements for the EUSART are to enable the connection to use the internal clock of the micro-controller, start and stop bit for transmission verification, and 4800 baud rate for global consistency. Figure 18**,** Figure 19**,** and Figure 20 illustrates the control registers used in the EUSART.



TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

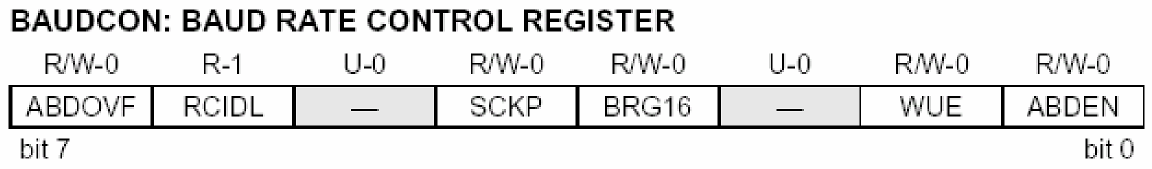
Figure 18: TXSTA Register.



RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Figure 19: RCSTA Register.



BAUDCON: BAUD RATE CONTROL REGISTER

R/W-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

Figure 20: BAUDCON Register.

There are other miscellaneous configurations required for the micro-controller, but these will be dependant on the schematic and layout. The configuration already explained outlines the most important setup requirements needed to run the micro-controller.

3.5. Other Components

3.5.1. Maxim 399 Dual 4-Channel Multiplexer

The Max399 multiplexers will be used to connect the micro-controllers to the GPS and Transceiver. This component is necessary we are limited in the number of serial connections available on the PIC. The Max399 was chosen because samples can be ordered from Maxim's website.

3.5.2. Maxim 232 RS232 Serial Buffers

The Max232 chips will be used as buffers between each component in our system. The buffers are required to convert the data from the components to standard RS232 voltage levels.

3.5.3. Maxim 3100 SPI Microwire-Compatible UART

The Max3100 universal asynchronous receive transmitter is optimized for micro-controller based SPI buses. It allows the micro-controller to communicate to the host (PC) through an asynchronous I/O suitable with RS232. This component is necessary to use the serial peripheral interface on the micro-controller to connect to the RS232 serial port on the PC.

3.5.4. Maxim 3223 RS232 Serial Buffers

The Max3223 chips will be used as a buffer between the SPI/UART chip and the PC. The buffers are much like the Max232 except it includes the clear to send and request to send pins used in the serial peripheral interface chip.

3.5.5. ST Microelectronics Voltage Regulators

Each component on will require a voltage regulator to provide a constant voltage. We will use a 3.3-volt regulator for the transceiver, and 5-volt regulators for the GPS unit, micro-controller, RS232 buffers, and Max399 Multiplexer.

4. Interface

The primary components of the system were explained in the previous section. Each component of the system interfaces with each other based on their available input/output ports. The primary interface types used in the systems are the universal asynchronous receive transmit peripheral (UART) and the serial peripheral interface (SPI). The personal units and base station share some common features, but the base station includes a connection to the PC, which is not included in the personal units.

4.1. SPI

The serial peripheral interface is a loose standard for controlling a clocked serial stream of data. The system is based on a synchronous string of bits that is sent by the transmitter and reassembled by the receiver to form complete bytes. Figure 21 illustrates a simple master and slave system. The two devices are connected via the serial peripheral interface bus and is characteristic of our system.

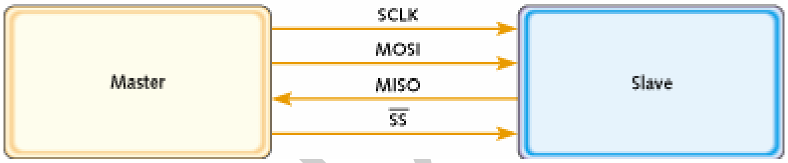


Figure 21: Simple Master-Slave System.

The SPI is implemented on the basis of a serial bus. In operation, there is a clock, “data in”, and “data out” and a chip select. Any serial device can be controlled with this combination of signals. The master selects the slave over the chip select line, and is synchronized with the clock transferring data over the in and out lines. The bus operates on a full duplex mode, capable of transmitting and receiving concurrently. Data goes into a SPI peripheral when the clock goes low, and comes out when the clock goes high. The serial peripheral interface bus is capable of multiplexing numerous devices, but in our case we have a simple master and simple slave system.

The generic serial peripheral interface operating in full duplex mode will be used to interface with the PC. The interface from the microcontroller will be connected to a UART enabled SPI chip to communicate with the PC via RS232 standards. This loosely defined serial standard is capable of interfacing with multiple components through a bus, but due to its generality, it will only be used for the PC.

4.2. UART

Universal Asynchronous Receiver/Transmitter (UART) peripheral is a key component in serial communications between components. It is a more specific serial interface, commonly called the serial communication interface, which is primarily used to communicate between different components. The UART takes bytes of data and transmits each individual bit sequentially. The receiver then reassembles the bits into complete bytes of data.

There are two primary forms of serial transmission: synchronous and asynchronous. Our system primarily deals with UART or universal asynchronous receiver/transmitter. The PIC18F2525 has a configurable USART module (Universal Synchronous-Asynchronous Receiver/Transmitter) that can be configured as an asynchronous or synchronous serial port.

Asynchronous transmission allows the system to transmit data without the transmitter sending a clock signal. Rather, the transmitter and receiver agree on timing parameters in advance by special configuration bytes that synchronize the sending and receiving units. Figure 22 illustrates the operation of a UART transmission line.

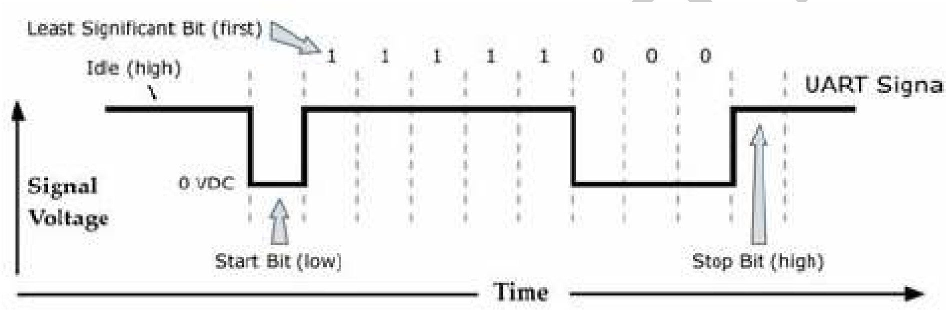


Figure 22: UART Transmission Line Operation.

When data is to be transmitted by the UART, a bit called the “start bit” is added to the beginning of each word. The start bit alerts the receiver that data is to be sent and forces the clock in the receiver to synchronize with the transmitter. These two clocks must be accurate enough to prevent loss of data. Following the start bit, each individual bit of the word is sent. The clock notifies the transmitter to send the next bit at a predefined interval. When the transmitter is done sending bits, the stop bit is used to notify the receiver that it is done.

Asynchronous data is self-synchronizing as previously explained. If there is no data to transmit, the line can idle. This is highly beneficial to our system since we have two components, the GPS and transceiver that will be transmitting over a single UART, thus both can share the same line as one component idles. The specifics behind interfacing the components will be explained later.

4.2.1. RS232

RS232 is a serial communication standard that is used over the UART transmission lines. The three major components operating over the UART is capable of transmitting with the

RS232 protocol. There are nine pins on a typical RS232 connection, although not all have to be used and typically only the Tx, Rx pins are used in a simple serial connection. These allow for simple asynchronous transmission as described above, but additional pins can be used to provide improved features.

Figure 23 illustrates a typical RS232 DB-9 Connector.

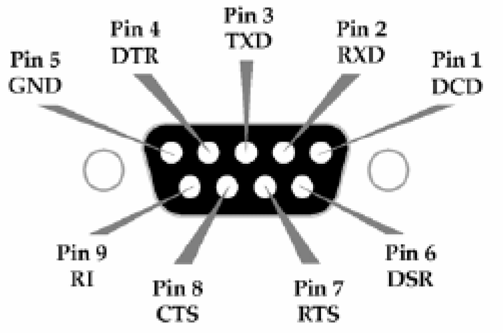


Figure 23: RS232 DB-9 Connector.

Each pin's operation is described in Table 4.

Table 4: RS232 DB-9 Pinout Description.

DB-9 Pin	RS-232 Name	Description
1	DCD	Data-Carrier Detect
2	RCD	Received Data
3	TXD	Transmitted Data
4	DTR	Data-Terminal-Ready
5	GND	Ground Signal
6	DSR	Data-Set Ready
7	RTS/CMD	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

To allow for some handshaking in transmission, RTS and CTS can be used as a request and acknowledgement for the transmitter and receiver. However, in our application, the units are time multiplexed to prevent data loss thus handshaking will not be needed. The Tx/Rx pins will be used to interface our components with each other.

5. Interface Implementation

The transmission standards and protocols were outlined in the previous section. Their application in our system is to interconnect the modules and allow each to communicate with each other. The primary function of the micro-controller will be to act as a data management unit that receives and forwards data according to their destination.

5.1. Personal Unit

The personal units are interfaced with the GPS and transceiver. There are two essential connections made between the three modules as shown in Figure 24.

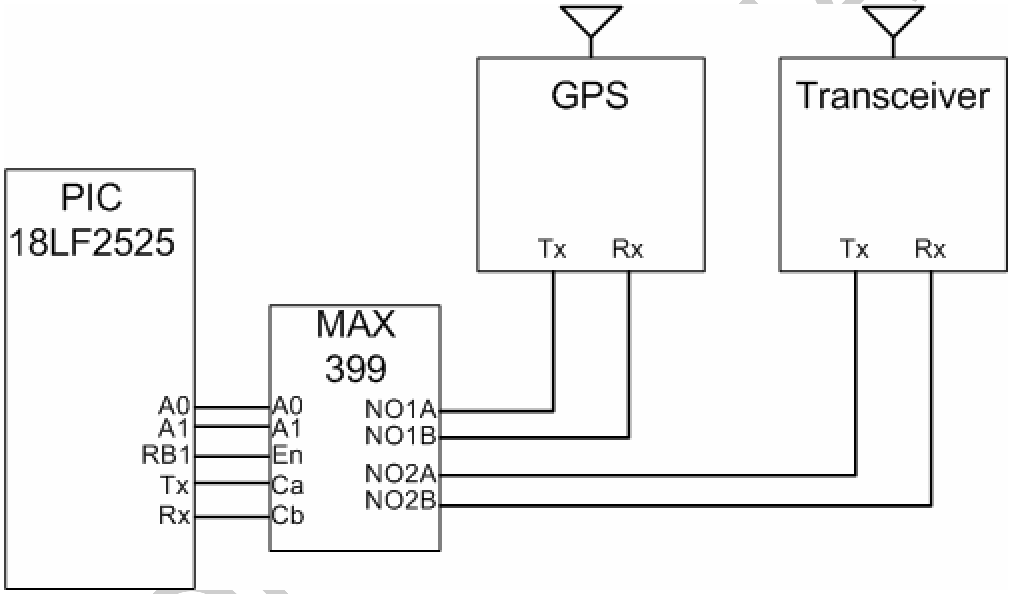


Figure 24: Simplified Personal Unit Interface Diagram.

The GPS and transceiver are connected to a four channel UART multiplexer, with MAX232 buffers (not shown) on either end. The two modules are time multiplexed and address controlled by the micro-controller. The timing scheme for the multiplexer is designed to prevent data loss between the transceiver and the GPS. We have simplified the operation of the personal unit due to the fact that the GPS will always be transmitting data and the transceiver will always be receiving data. The micro-controller will take data from the GPS, buffer and store it, then forward it onto the transceiver to be transmitted. Figure 25 outlines the timing scheme used in the interface.

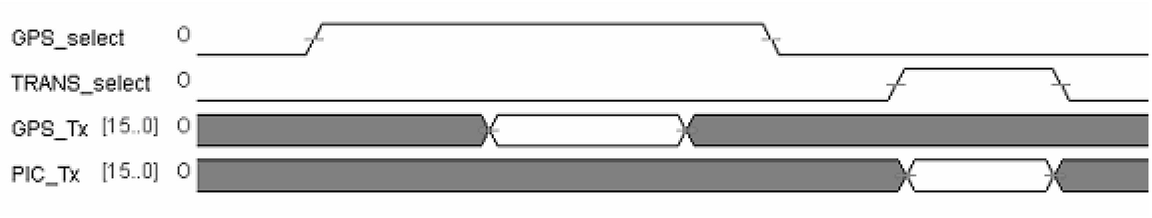


Figure 25: Personal Unit Timing Diagram.

The signals are defined in the table below.

Table 5: Personal Unit Signal Description.

Signal	Description
GPS_select	Select GPS on MUX
TRANS_select	Select Transceiver on MUX
GPS_Tx	GPS transmits data to micro-controller
PIC_Tx	Micro-controller transmits data to the transceiver

The GPS will be configured to transmit data at a certain interval and the micro-controller will receive the data then store it in its EEPROM. The micro-controller will select the GPS by default until the GPS transmits a stop bit indicating the end of the transmission. This will avoid issues of timing between the GPS and the micro-controller since the micro-controller will select the GPS by default until the buffer is full. The micro-controller will then select the transceiver on the multiplexer and then retrieve the data from the EEPROM and forward it to the transceiver. This process is repeated nonstop over the duration of period the personal unit is operating. Before this cycle is initiated, the micro-controller will be responsible for establishing the configuration for the GPS and transceiver such as sampling rate, baud rate, transmission time, etc.

The personal unit has a simple interface and is primarily dependant on the UART capabilities of the micro-controller.

5.2. Base Station

The base station shares many of the same characteristics as the personal units. However, the base station is connected to three different components: GPS, transceiver, and PC. These connections are made via three connections to the micro-controller as illustrated in Figure 26.

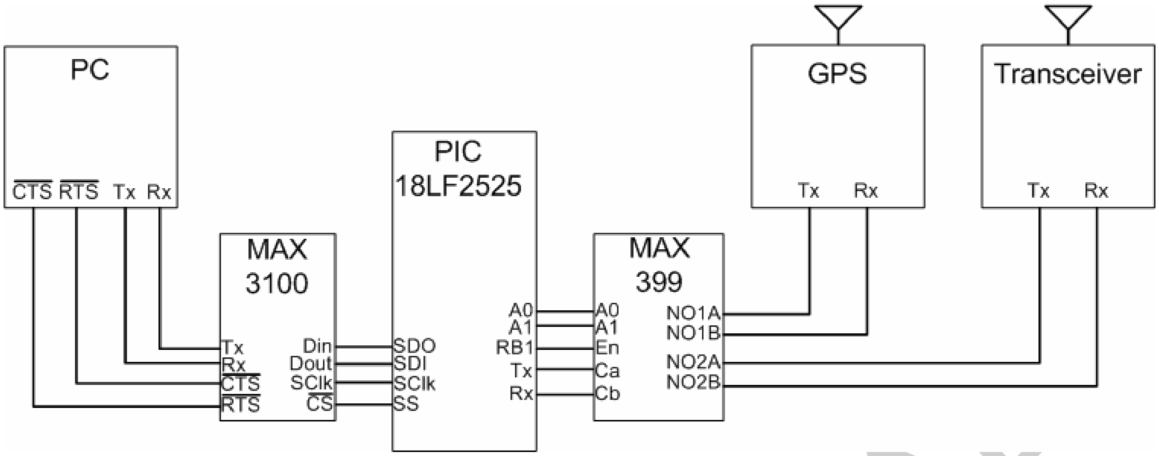


Figure 26: Base Station Interface Diagram.

The connections can be grouped into two major sections: the GPS/transceiver over the UART transmission line and the PC via the serial peripheral interface bus. Much like the personal station, the GPS and the transceiver share the UART over the four channel multiplexer. The PC is independently connected to the micro-controller through a MAX3100 SPI Microwire-compatible UART IC. This allows the serial peripheral interface to connect to the RS232 DB9 port on the PC. Again, the timing scheme is based on preventing as much data loss as possible with priorities set on the receiving data from individual personal units. Figure 27 outlines the timing scheme used by our base station.

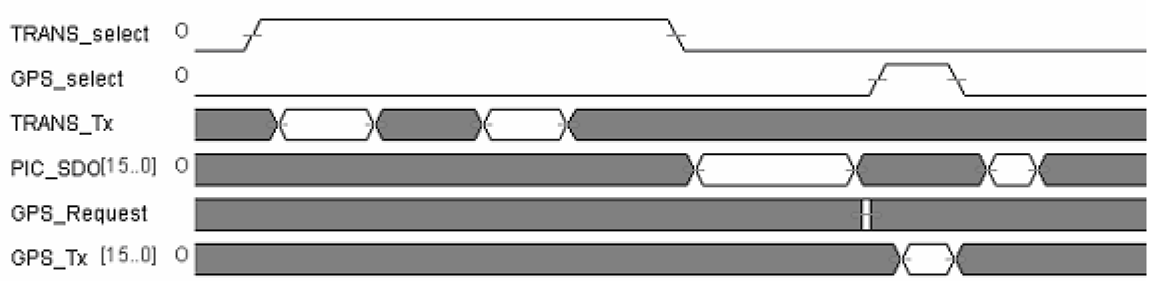


Figure 27: Base Station Timing Diagram.

The signals are defined in the table below.

Table 6: Base Station Signal Description.

Pins	Description
TRANS_select	Select the transceiver on mux
GPS_select	Select GPS on mux
TRANS_Tx	Transceiver transmits data to the micro-controller
PIC_SDO	Micro-controller transmits data to the PC
GPS_Request	Micro-controller requests data from the PC
GPS_Tx	GPS transmits data to the micro-controller

The base station acts as a data management unit by taking data from the GPS and the transceiver and forwarding it to the PC. The information transmitted by the transceiver has the highest priority in the system since it contains information regarding the personal unit. The micro-controller will select the transceiver by default and receive data for a set amount of time. When this occurs, the micro-controller will deselect the transceiver and idle. The micro-controller will then forward the data to the PC, request data from the GPS for an update on the current location and forward the data to the PC, and then switch back to the transceiver. This will allow for maximum amount of time spent receiving data from the transceiver.

This cycle is repeated during the operation of the base station. Much like the personal units, before the data cycle begins, the configuration of the individual components is configured by the micro-controller.

The base station is dependant on two serial connections to establish interfaces between each corresponding module. The GPS and transceiver will both be transmitting data and will be time-multiplexed over the UART in order to receive all data without any loss. All received data will then is forwarded to the PC. As previously explained, the connection to the PC will be done over a UART-enabled SPI microwire.

5.3. Power Requirements

The power requirements of each device are below.

Table 7: Power Requirements of the GeoPreserver™ Components.

Components	Power Supply Range
RF Module (XBEE/XBEE PRO)	2.8V – 3.4V (or 3.3V)
GPS	3.8V – 6.5V (or 5V)
Buffer (MAX232)	3V – 5.5V
Mux (MAX399)	3V – 15V
PIC(PIC18F2525)	2V – 5.5V
Duracell 9V Battery for Personal units	9V
CR2032 Backup battery	3V
12V AC power adapter for base station	12V

It is essential that every component is supplied its proper voltage and current levels to maintain proper supply. For the personal units, the primary supply will come from a 9V battery and for the base station from a 12V AC power adapter.

These supplies will provide a source of power or voltage level that may not be consistent with the component requirements. In order to regulate the voltages to required levels, ST Microelectronics 1086 and 1085 voltage regulators will be used to output a constant 5V and 3.3V voltage for each respective component according to their specifications.

Although the 9V battery may exhibit a linear decay in power due to its specification the voltage regulator will consistently maintain a constant voltage and current output to power the components.

Each major component will be localized and supplied by its own independent regulator to prevent shorting or power conflicts, and other components will be supplied from a common source.

6. Software Design

6.1. Overview

The software application for the GeoPreserver™ is required to provide an interface between the GeoPreserver™ base station and the user. The software designed below is for the prototype of the GeoPreserver™, which will run on a PC. The final product will use a similar interface, but will run off a complete embedded system.

For our interface implementation, we have chosen to use the C++ programming language on Microsoft Visual Studio running on Windows XP. The base station will connect to the PC's serial port.

The main flowchart for our program's fundamental operation is shown in Figure 28. The flowchart outlines the program's behaviour, beginning with the software initialization indicated with "START." The process then proceeds to initialize itself and detect all personal units linked to the system before running a continuous loop. The chief functions of the loop are to update the positions of the base station and the personal units to track for Man Overboard incidences.

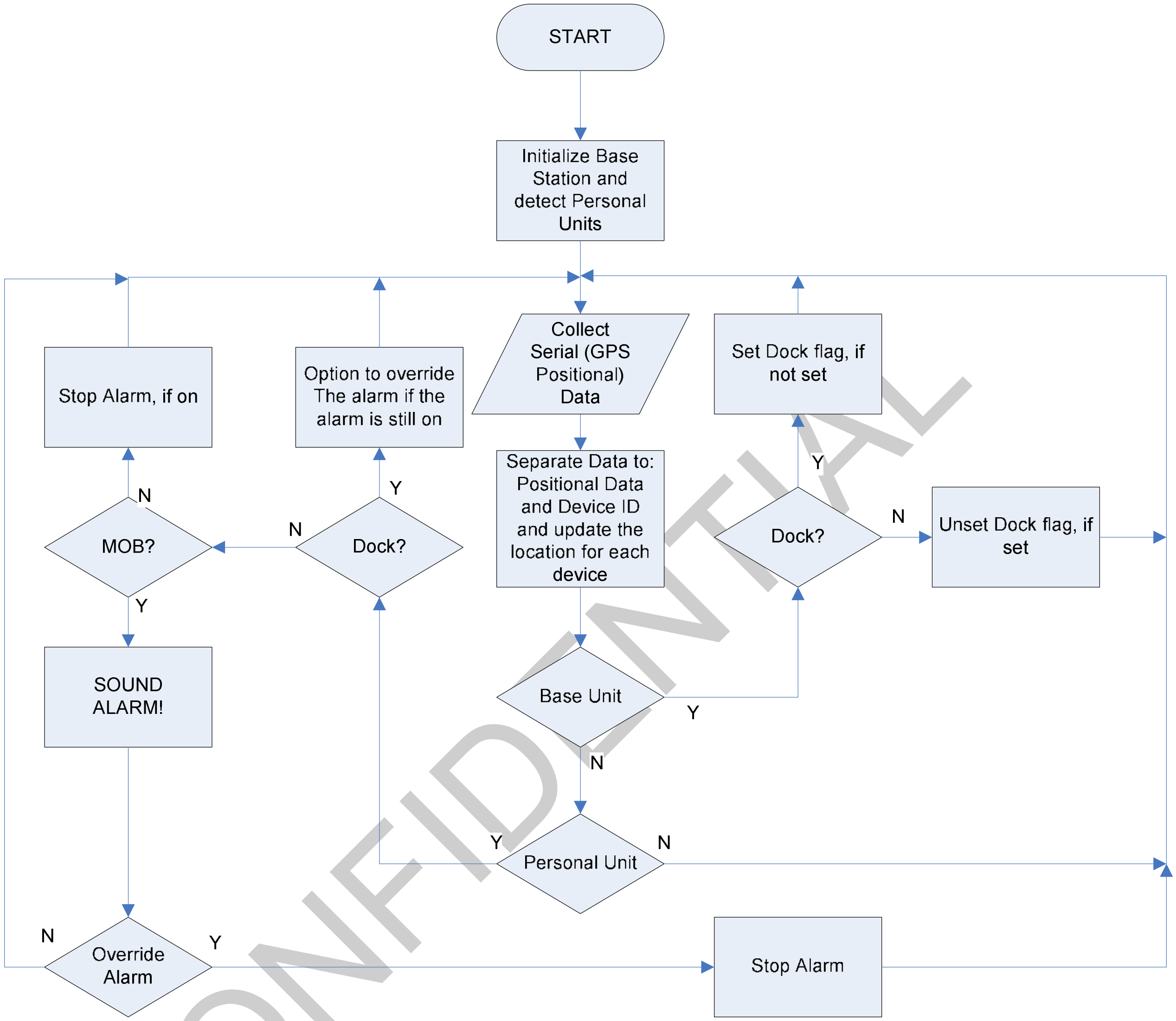

START
Initialize Base Station and detect Personal Units
Collect Serial (GPS Positional) Data
Separate Data to: Positional Data and Device ID and update the location for each device
Base Unit
Y
N
Personal Unit
Y
N
Dock?
Y
N
Set Dock flag, if not set
Unset Dock flag, if set
Dock?
Y
N
Option to override The alarm if the alarm is still on
MOB?
N
Y
Stop Alarm, if on
SOUND ALARM!
Override Alarm
N
Y
Stop Alarm

Figure 28: Main Flow Chart.

6.1.1. Main Flowchart In-Depth

There are two threads which run the functions outlined in the main flowchart. One thread, the Serial Handler thread, is used to collect the GPS data which will arrive as serial data into the PC. The Serial Handler thread will grab the NMEA GPRMC data and separate the individual packet into longitude and latitude values. It will also distinguish where the data came from (base station or personal units). This positional and device data will be placed in a global location for each device as shown in Figure 28.

A second thread, the Data Calculation/GUI[1] thread, is responsible for reading the global data and performing the distance and location calculations, as well as managing the user interface. The Data Calculation/GUI thread begins its operation after the "Separate Data…" process box and, as indicated by the flowchart, will perform different functions depending on where the data arrived from. If the data arrived from the base station, the Data Calculation/GUI thread will check if this location corresponds to a user established dock location. A DOCK FLAG will be set or cleared depending on whether or not the GPS data is within 30 meters of the dock location.

If the data arrived from the personal unit, the Data Calculation/GUI thread will first check to ensure that the DOCK FLAG is not set. If the DOCK FLAG is not set, the Data Calculation/GUI thread will calculate whether a Man Overboard incident has occurred; if so, the alarm will sound.

An alarm override capability is built into the system for cases when,

- The alarm is still active while the ship is proceeding to a dock.
- A man overboard incident has been rectified.
- The alarm has alerted the crew and is deemed no longer necessary by the operator to carry out rescue functions.

6.2. Program Structure

The application will use an object-oriented approach. The objects are defined as the base station, the personal unit, and the crewman that must register with the personal unit.

The program structure will be based on a multi-threaded application involving several components. In total, the program will use a process containing 2 threads. The first thread will manage serial port communications, and the second thread will manage the graphical user interface.

The two threads require some method of exchanging data. The actual data exchanged is a modified NMEA-0183 packets in GPRMC format. The RMC format provides Time, Date, Course, Speed, and Position data in ASCII format, which is exactly what we need for our purposes of tracking the personal units. In addition, the PIC micro-controller will have added an identifier at the beginning of the packet to indicate whether the packet belongs the base station or to a particular personal unit. We have decided to exchange this data using a shared memory construct that is controlled by a mutex. The mutex will control the access rights so that changes are not made to the memory by one thread while the other thread is reading or writing. The serial port thread will receive this data from the PIC microprocessor, and pass it into the shared memory. The GUI thread will read

[1] The Data Calculation/GUI thread also controls the GUI which accepts user commands. This thread will be explained in greater detail in the next section,
6.3. Algorithms.

the data from shared memory and perform computations. The relationship is more clearly illustrated in the figure below.

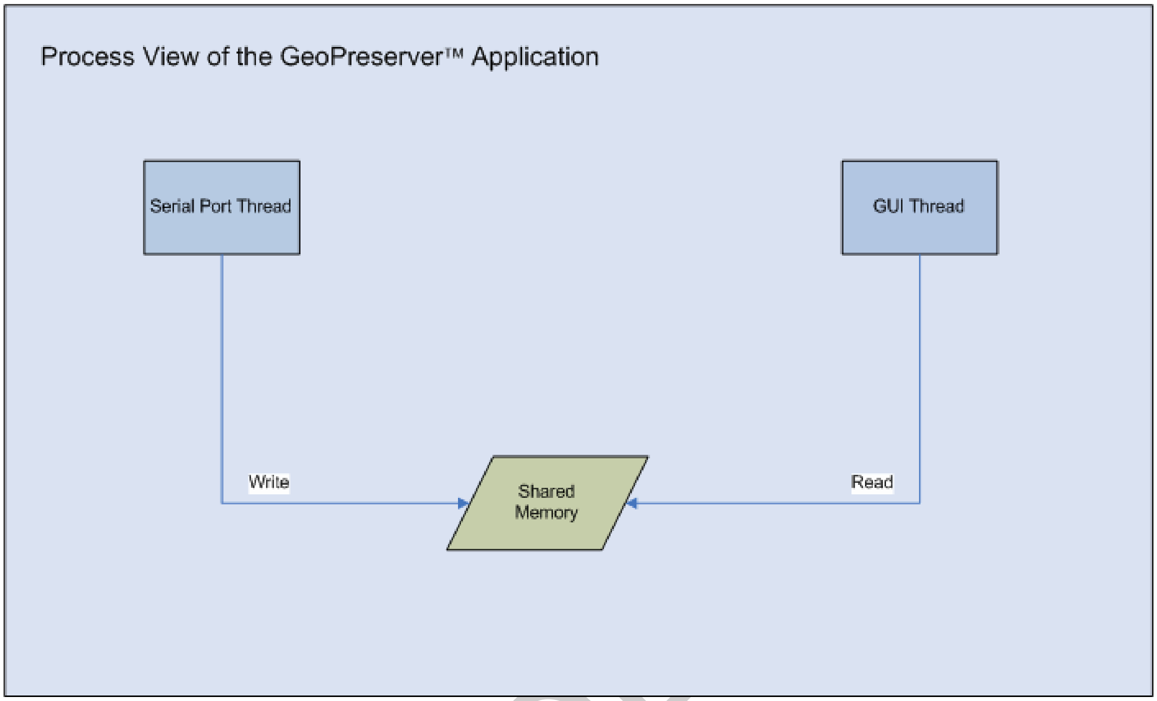


Figure 29: Process View of the GeoPreserver™ Application.

In addition, the GUI thread will require a database containing the users of each Personal Unit. The fields in this database will consist of the following:

- Last Name
- First Name
- GeoPreserver™ ID
- Current Status (Safe or Overboard)
- Current Position
- Last Known Position
- Time of Last Known Position

This database will be loaded from a database file when the application is started, and written to the database file when the application is closed.

The serial port thread will be based on the Serial Library written by Ramon de Klein. Refer to the Reference section for more information.

6.3. Algorithms

To clarify the functionality, this section will cover the algorithms of the main process, and each of the two threads.

6.3.1. Main Process

The Main process is required to setup the resources for the threads, and spawn the threads. The pseudo code for the Main process is shown below.

```
Create mutex
Create shared memory
Initialize shared memory
Spawn Serial Port Thread
Spawn GUI Thread
```

6.3.2. Serial Port Thread

The Serial Port Thread is used to manage serial port communications between the micro-controller and the PC for the base station. The pseudo code for the Serial Port Thread is shown below.

```
Open the COM Port
Setup the connection settings (baud rate, parity, stop bit, flow
control)
Do {
      Wait for incoming data
      Read all the data from the serial port.
      Wait for the mutex
      Write the data to shared memory
      Release the mutex

} while not quit
```

6.3.3. Data Calculation/GUI Thread

The Data Calculation/GUI thread manages the graphical user interface and also performs the distance calculations between the base station, the personal unit, and the defined geofence boundary to determine the status of the personal units. The position data of the personal units and base station will be extracted from the shared memory, which will be updated by the Serial Handler Thread as previously displayed in Figure 29. The pseudo code for Data Calculation/GUI Thread is shown below.

```
Establish connection with shared memory
Set up timers to read from shared memory
RUN the GUI
```

Implicit in running the GUI are the calculations that determine the status of each personal unit. The latitude and longitude of each personal unit will be checked to see if it is outside the geofence. We chose to implement the geofence as a circular perimeter around the ship to allow for a simpler implementation, although this may differ from the final product. The algorithm for the calculation is below.

```
Obtain the Personal Unit’s Latitude and Longitude
Obtain the Base Station’s Latitude and Longitude
Use the Haversine Formula to compute the distance between 2 points
IF the Personal Unit’s distance is less than Geofence Threshold
distance,
      Personal Unit is safe
ELSE
      Personal Unit is NOT safe
```

The Haversine Formula is used in our calculation to translate Latitude and Longitude of 2 points to the distance between 2 points. Refer to the Reference section for more details.

In addition, the thread will determine if the ship is near a dock location. If the Personal Unit leaves the geofence in the dock location, they will be considered safe.

The Dock Algorithm is also performed in this thread and is shown below. The Dock algorithm auto-docks the ship when the distance is within 30 meters of the dock but also allows the user to dock early when approaching a dock location.

```
IF base station’s distance is within 30 meters of the dock THEN
      DOCK FLAG = true
ELSE IF base station’s distance is within 50 meters of the dock THEN
      FIND the difference in distance, delta, between the dock location
      and base station between two time intervals
      IF delta is getting smaller AND DOCK FLAG = false THEN
            DISPLAY dock now?
            IF Dock Button is pressed THEN
                  DOCK FLAG = true
            ELSE
                  DOCK FLAG = false
      ELSE IF delta is getter larger AND DOCK FLAG = true THEN
            DOCK FLAG = false
ELSE
      DOCK FLAG = false
```

The GUI’s functional operation for this thread will be explained in the next section.

6.4 GUI Mockup and Description

The GUI will be created using Microsoft's MFC library. This library was chosen for its quick and simple functions for creating window-based applications. A mockup of the GUI is shown in Figure 30 below.

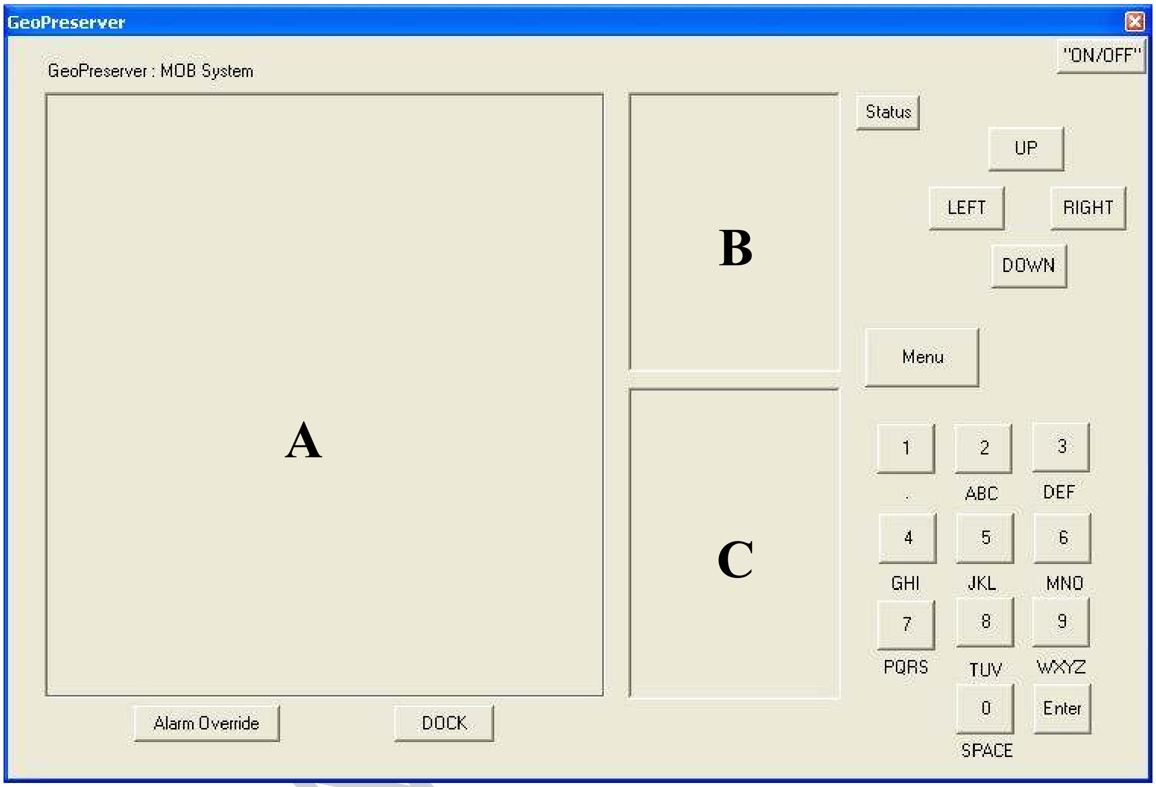


Figure 30 - GUI Mockup.

The larger box, labeled A, is for pictures/drawings and will show the boat, geofence perimeter, and the personal unit's location, similar to Figure 31 below. The two smaller boxes, labeled B and C, to the immediate right of the larger box are text boxes. The box B will display the status of the personal unit or the base station. Text box C will display the menu to access the base stations functions. The UP/DOWN/LEFT/RIGHT navigational buttons will be used to cycle the display to show either the base station or the personal unit information. The "ON/OFF" button is used to turn on or off the GeoPreserver™ while keeping the dialog box shown. . When the GeoPreserver™ is turned off, the data from the base station will be written to the hard drive. The dialog will only close when the [X] button in the title bar is pressed. This ON/OFF functionality is meant to simulate an on/off switch for an actual physical base station unit.

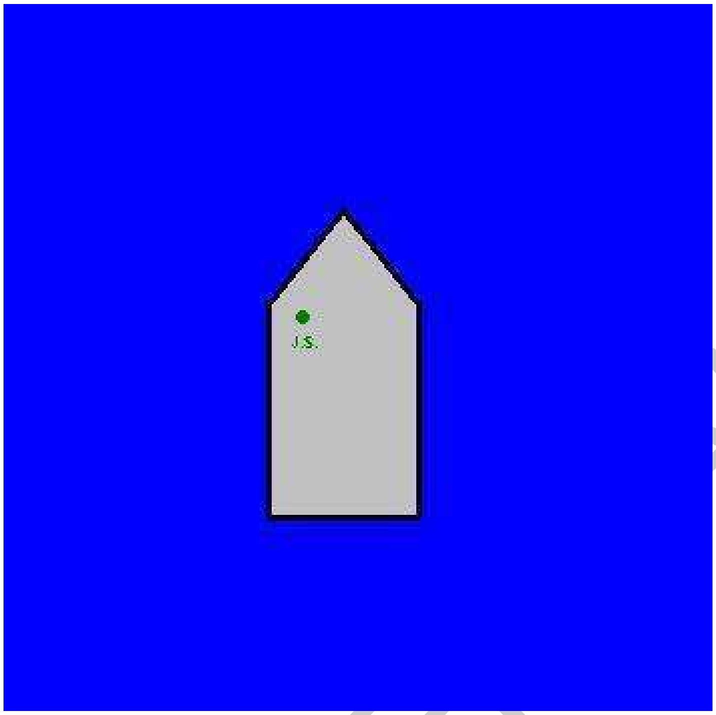


Figure 31 - Ship Display with Personal Unit Location.

The green dot in Figure 31 is the location of the personal unit and will move as the unit moves along the ship.

During a MOB situation the display will change to show the approximate direction as well as continue tracking the personal unit's location. The "MOB!" text at the top of the display and the directional locator (shaded red region) will flash. This is shown in Figure 32 below.

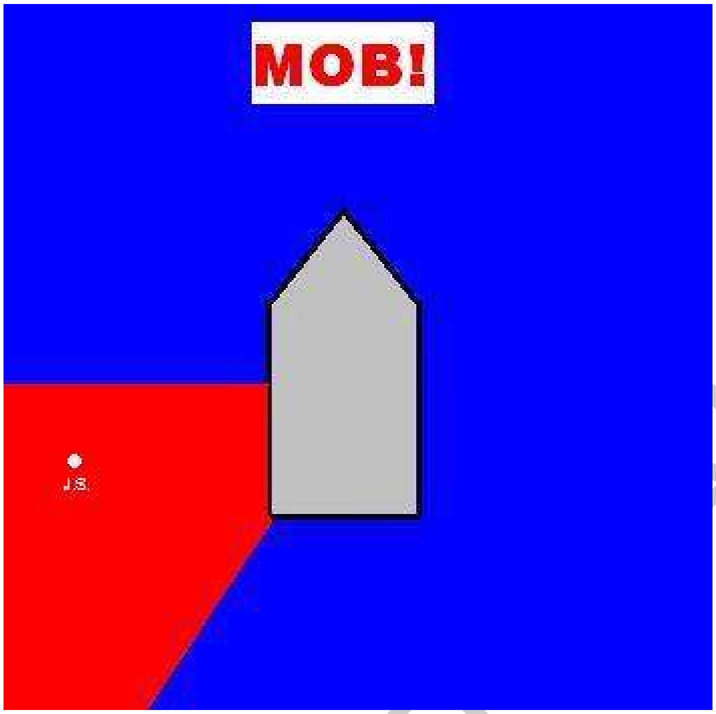


Figure 32 - Ship Display During a MOB.

Box B in Figure 30 will display the GeoPreserver™'s status: Online, Offline, or MOB. The display will also show the status of the base station or the individual personal units and the number of personal units attached. By using the UP/DOWN/LEFT/RIGHT navigational buttons the view can be switched between the base station and the personal unit information. An example of the display for box B is shown below.

```
System Status:          Online
Number of Personal Units:     1
--------------------------------------------
-Base Station-
Longitude:  YY
Latitude:   XX
```

When switched to the personal units by pressing on the Navigational buttons, the display should look like below:

```
System Status:          Online
Number of Personal Units:     1
--------------------------------------------
-Personal Unit #1-
-JOE SOMEBODY-
Longitude:  YY
```

```
Latitude:   XX
Distance from Base Station: X
```

Pressing the Status button will set the navigational buttons to focus on cycling through the status of the base station or personal unit, regardless if the navigational buttons are used for GUI Menu options. The Status button will also bring the default box B display to show the Base Stations status. Pressing the Menu button will refocus the navigational buttons for GUI Menu option use.

The Alarm Override button will allow the user to stop the alarm from sounding during a MOB situation.

The DOCK button allows the user to manually set or clear the GeoPreserver™ in a dock state so that the alarm won't sound if a personal unit crosses the geofence. This is not meant as a permanent setting as the dock status will unset itself if the ship is moving away from the dock as explained in section 6.3.3. Data Calculation/GUI Thread above.

6.4.1. GUI Menu Options

The Menu button will act as a "back" button when navigating through the layers of menus. It will also act as a "cancel" button when the user has already selected menu option. The Main Menu displayed in box C will be:

```
      --- Main Menu ---
   1.    Personal Unit
   2.    Crew
   3.    Dock
   4.    Ship
```

Pressing 1 on the number pad will reveal the following menu

```
         -  Personal Unit –
   1.    Detect Personal Unit
   2.    Assign to Crewman
   3.    Deregister from Crewman
```

The Detect and Register option #1 allows the user to detect a personal unit within 3 meters of the base station and to register or deregister it from the base station.

```
IF personal unit is <= 3 meters from base station THEN
{
      DISPLAY personal unit
      SWITCH (Option)
            Option 1: register with base station
            Option 2: deregister from base station
            Option Menu: Cancel
}
ELSE
      DISPLAY personal unit not found
```

Assign to Crewman option #2 will detect a personal unit within 3 meters of the base station and allow the user to assign it a crew member. Pressing the LEFT and RIGHT navigation buttons allows the user to cycle between the different crew members.

```
IF personal unit is <= 3 meters from base station THEN
{
      DISPLAY personal unit
      DISPLAY unregistered crew man
      WHILE personal unit is unassigned OR cancelled LOOP
      {
         SWITCH (Option)
            Option 1: Assign to Crewman, break from loop
            Option LEFT: Display next crewman
            Option RIGHT: Display previous crewman
            Option Menu: Cancel
      }
}
ELSE
      DISPLAY "personal unit not found"
```

Deregister from Crewman option #3 will detect a personal unit within 3 meters of the base station and allow the user to deregister it from the crewman it's currently registered to.

```
IF personal unit is <= 3 meters from base station THEN
{
      DISPLAY personal unit and registered crewman
      SWITCH (Option)
            Option 1: Deregister crewman
            Option Menu: Cancel
}
ELSE
      DISPLAY personal unit not found
```

Pressing 2 in the Main Menu will show the following menu options:

```
         -  Crew -
   1.    Search (Edit/Delete)
   2.    Add New Crewmember
```

Search option #1 allows the user to search through the crew members stored in the base unit to either edit or delete them from base station. Using the LEFT or RIGHT navigational buttons will cycle between the previous or next crewmember stored in the base station, respectively. The delete command will display an additional warning before deletion if the crew member is currently registered to an active personal unit.

```
DISPLAY first crew member info
WHILE cancel is not selected LOOP
{
      SWITCH (Option)
            Option 1: Edit crew info
            Option 2: Delete crew member
            Option LEFT: Display next crewman
            Option RIGHT: Display previous crewman
            Option Menu: Cancel
}
```

The Add New Crewmember option #2 allows the user to enter a new crew man into the base station. The keypad will be used to enter in the name. The LEFT and RIGHT navigation buttons will be used to move the cursor and the UP / DOWN buttons will used to cycle the input from numeric or alphabetical characters. The Enter button in the keypad is used to end the name input function.

```
WHILE cancel is not selected LOOP
{
      DISPLAY indicated to user to enter full name
      INPUT name  //user can also cancel at this point by pressing menu
      DISPLAY input another user?
            SWITCH (Option)
                  Option 1: Yes
                  Option Menu: No
}
```

Pressing 3 in the Main Menu will show the following menu options:

```
        -  Dock –
   1.    Search Docks
   2.    Add
```

Search Docks option #1 allows the user to search through the dock locations stored in the base unit to either edit or delete them from base station. Using the LEFT or RIGHT navigational buttons will cycle between the previous or next dock location stored in the base station, respectively.

```
DISPLAY first dock location
WHILE cancel is not selected LOOP
{
      SWITCH (Option)
            Option 1: Edit dock info
            Option 2: Delete dock
            Option LEFT: Display next dock location
            Option RIGHT: Display previous dock location
            Option Menu: Cancel
}
```

The Add option #2 allows the user to add the current longitude and latitude location as a dock location so that the GeoPreserver™ will not sound the alarm when nearby the dock. The user will also enter in the name of the of the newly added dock location.

```
WHILE cancel is not selected LOOP
{
      DISPLAY current longitude and latitude location of the base
      station
      DISPLAY ask user if they want to set this location as a new dock
      location
      INPUT name of dock      //user can also cancel at this //point by
                              pressing menu
      DISPLAY input another dock?
            SWITCH (Option)
                  Option 1: Yes
                  Option Menu: No
}
```

Pressing 4 in the Main Menu will show the following menu option:

```
         -  Ship -
   1.    Setup Geofence Dimensions
```

Setup Geofence Dimensions option #1 will allow the user to set the radius of the geofence with respect to the base station GPS module's location on the ship. The position of the GPS module ideally will be centered slightly behind the center point of the ship, closer to the stern.

```
DISPLAY radius of geofence?
INPUT radius
RETURN to previous menu
```

6.4.2. Summary of GUI Button Functions

Table 8: Summary of GUI Button Functions.

GUI Item	*Function*
"ON/OFF"	Turns on or off the GeoPreserver™ system. When the geofence is turned off, stored data in the base station will be written to the hard drive.
Menu	Functions as a Back or Cancel button when navigating through the menus. Also functions to re-focus the navigational buttons for GUI Menu use.
Keypad	Used to selected menu options or to enter in names for crewmembers or dock locations.
Status	When pressed the display will show the Base Station information. It will also focus the navigational buttons to cycle through the base station or personal unit status.
UP/DOWN/LEFT/RIGHT Navigation buttons	Used to cycle between the base station and the personal units in the status screen. The LEFT/RIGHT buttons are also used to move the cursor when the user is entering names for the crewmembers or dock location. UP/DOWN buttons also used to switch between numeric and alphabetical characters when entering in names.
Alarm Override	Overrides the alarm and shuts it off during a MOB situation
DOCK	Allows the user to manually set or unset the GeoPreserver™ to a docked state when approaching a DOCK location so that the alarm will not sound when a personal unit leaves the geofence.

7. Test Plan

The test plan for the system is separated into two sections, the System Test Cases for the overall system, and the Software Test Cases to test the GUI's and serial functions. The System Test Cases will encompass the overall operation of the GeoPreserver™ system and will attempt stress the boundary and regular use case scenarios.

7.1. System Test Cases

7.1.1. Verify Alarm Activation when Personal Unit is Outside the Geofence

The purpose of the tests below is to ensure that the alarm sounds when the tests are complete, and that the display shows the correct MOB information.

Test #	*Test Details*
1	• Turn on the GeoPreserver™ when the personal unit is outside the geofence.
2	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence. • Move the personal unit outside of the geofence.
3	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence • Move the base station away from the personal unit until the personal unit is outside of the geofence.

7.1.2. Verify Alarm Deactivation When Personal Unit Returns Within Geofence

The purpose of the tests below is to ensure that the alarm turns off when the tests are complete, and that the display shows the correct MOB information.

Test #	*Test Details*
4	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence. • Move the personal unit outside the geofence. • Move the personal unit back inside the geofence.
5	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence. • Move the base station away from the personal unit until the personal unit is outside the geofence. • Move the personal unit back inside the geofence.

7.1.3. Verify Acceptable Alarm Limit When Person is at the Edge of the Boat

The purpose of these tests is to ensure that the alarm does not go off prematurely when the person is near the edge of the boat.

Test #	*Test Details*
6	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence. • Move the personal unit along the border of the geofence.

7.1.4. Verify Proper Alarm Operation During Rapid Entry/Exit of the Geofence

The purpose of these tests is to ensure that the alarm will turn on and off as appropriate when rapid transitions into and out of the geofence occur. Also, the test will ensure that the display updates appropriately.

Test #	*Test Details*
7	• Turn on the GeoPreserver™, make sure the personal unit inside the geofence. • Move the personal unit outside of the geofence. • Move the personal unit back inside the geofence. • Repeat the above 2 steps quickly.

7.2. Software Test Cases

Software testing will involve separate tests for both the Serial Port Library, and the GUI functions.

7.2.1. Serial Data Testing:

The serial port library's writing capability will be tested by using a console program to write to the serial port, and the DB-9 connector of the COM port being probed with an oscilloscope to see if the probed output matches the input character.

Test #	*Test Details*
8	• Connect to the COM port. • Setup the connection • Write an ASCII character to the serial port.

Reading will be tested by using the GPS module to transmit data to the serial port, which will be read by the program.

Test #	*Test Details*
9	• Setup the GPS module connection to the PC. • Send a NMEA-0183 GPRMC packet to the PC • Use the console application to read the serial port

7.2.2. General GUI Testing:

A number of tests are required to test the GUI functionality.

7.2.2.1. Adding a User to the System

This test is to verify that a user can be added to the system.

Test #	*Test Details*
10	• Select Add New Crewman from the GUI menu. • Add in a crew member. • Access the Search Menu option. • Verify that the new crew member is in the system.

Retest the scenario above with the Dock Menu option.

7.2.2.2. Editing and Removing a User from the System

This test is to verify that a user can be edited or removed from the system.

Test #	*Test Details*
11	• Select Search for Member from the GUI menu. • Press the navigational buttons to cycle through all crew in both directions. • Try to Edit a user. • Try to remove a user. • Exit menu and reselect the Search option. • Cycle through all the users using the navigational buttons.

Verify that the user’s info has been edited and a user has been removed.

Retest the previous scenario with the Dock Menu

7.2.2.3. Alarm Override Function

This test is to verify the functionality of the Alarm Override button.

Test #	*Test Details*
12	• Cause a MOB situation. • Press the Alarm Override button. • Move the personal unit back in, cause another MOB situation. • Press the Alarm Override button.

Verify that the Alarm turns off at step 2, turns back on at step 3, and turns back off at step 4.

7.2.2.4. Dock Feature

This test is to verify the Dock Button functionality.

Test #	*Test Details*
13	• Press the dock button when near the dock. • Move the personal unit outside of the geofence. • Move the base station significantly away from the dock and away from the personal unit.

Verify that at step 2 that the alarm doesn’t sound, and that at step 3 that the alarm does sound.

8. Conclusion

The design specification discussed in this document outlines and provides a detailed overview of the GeoPreserver™. This document provides a mental picture as to how our main components will be interfaced, and integrated with the GUI application. Block diagrams are provided to illustrate the working conditions of the GeoPreserver™, and flowcharts are provided to illustrate the functionality of the GUI application. Each component is broken down into its features, setup requirements, and how it is interfaced with the other components. Algorithms of the GUI are provided in detail so that the operation of each section of the application can be understood. Diagrams were also provided in describing the possible GUI interface. Having followed these design specifications, OnJin Engineering is confident that our prototype will be completed and functional by our prescribed deadline, April 12, 2006.

References

[1] Introduction to Serial Peripheral Interface. 22 Feb 2006.
< http://embedded.com/showArticle.jhtml?articleID=9900483>

[2] Klein, Ramon. Serial Library. 16 Feb. 2006
<http://home.ict.nl/~ramklein/Projects/Serial.html>

[3] Laipac TF-11 Product specifications, 10 Feb. 2006
<http://laipac.com/gps_tf10_eng.htm>

[4] Maxstream XBee-Pro Product Manual, 22 Feb. 2006.
<http://www.maxstream.net/products/xbee/product-manual_XBee_OEM_RF-Modules.pdf>

[5] Micrologic Systems. Typical system configuration. 18 Feb. 2006.
< www.micrologic-systems.com/primers/scada10.htm>

[6] Microchip PIC18F2525 Data Sheet, 22 Feb. 2006
<http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2057&ty=&dty=Data+Sheets&section=Data+Sheets&ssUserText=PIC18F2525>

[7] Movable Type Scripts. Calculate distance and bearing between two latitude/longitude points using Haversine formula. 5 Mar. 2006
<http://www.movable-type.co.uk/scripts/LatLong.html>.

[8] -----. GIS FAQ Q5.1: Great circle distance between 2 points. 5 Mar. 2006
<http://www.movable-type.co.uk/scripts/GIS-FAQ-5.1.html>.

[9] National Marine Electronics Association (NMEA). 20 Feb. 2006
<http://www.nmea.org/>

[10] NMEA 0183 Protocol. 22 Feb. 2006 <http://www.kh-gps.de/nmea-faq.htm>

RS232 Protocol. 22 Feb, 2006. <http://en.wikipedia.org/wiki/RS232>

[11] Trimble. All about GPS. 15 Jan. 2006 <http://www.trimble.com/gps/>.

[12] Universal Asynchronous Receive Transmitter. Wikipedia.
< http://en.wikipedia.org/wiki/UART>

[13] USGlobalSat Inc. 2005. USGlobalSat GPS Forums. What is IEC 60529? 20 Feb. 2006 <http://www.usglobalsat.com/forum/topic.asp?TOPIC_ID=30>.

[14] Webopedia. What is DSSS? – A Word definition from the Webopedia Computer Dictionary. 18 Feb. 2006 <http://www.webopedia.com/TERM/D/DSSS.html>