Murdoch
UNIVERSITY
PERTH WESTERN AUSTRALIA

The School of Engineering and Energy

Differential Global Positioning System

Submitted in Partial Fulfilment of the requirements for the Degree of Bachelor of Engineering.

2010 - 2011

Supervisor:

Dr. Gareth Lee

Executive Summary

Differential Global Positioning System (DGPS) is emerging as a crucial technology within the resource and communications sector, allowing for the development of autonomous control in hazardous environments. This thesis is concerned with the design, construction and evaluation of a low cost and high performance DGPS which can be embedded into robotic systems. The principle behind this thesis is to prove that a DGPS can be designed to meet the same performance of an industrial grade system, while using low end components. The thesis is composed of five major stages: research, design, construction, evaluation and documentation. This thesis is to provide documentation of the development and evaluation of the DGPS. The final product is a fully documented and highly functional DGPS system, for the Murdoch University School of Engineering & Energy to integrate into future projects.

Standard GPS systems are based on a single Global Navigation Satellite System (GNSS) signal receiver to process position co-ordinates and relevant data. These GNSS signals containing pseudoranges, travel through the atmosphere and are susceptible to distortion, which is the predominate cause of error in GPS positioning. The atmospheric distortion and hence positioning error can be significantly reduced through the use of DGPS as it assumes that the signals have travelled the same atmospheric path and hence induced with the same distortion. By adopting this concept, highly accurate relative positioning between receivers can be produced through a series of calculations. DGPS can be extended further to produce corrected positioning based on fixed receiver locations which can compare the error of the fixed position to those received, and then apply the same error to other mobile receiver positions. Currently a large scale network of DGPS base stations exists and can be used for correcting commercial and industrial grade systems, however these can be expensive. The expense of these systems is largely dependent on the accuracy and functionality that the system can produce, and therefore highlights the need for this thesis as a means of a cost effective solution.

The prototype produced in this thesis was the product of thorough engineering design and evaluation of the three major components; communications, hardware and software. Each of these components had sub-objectives allow the main objective of a low cost, high performance system to be achieved. Evaluation of proposed communication wireless transceivers was undertaken to find characteristics of reliability, suitable bandwidth and a sufficient operating distance. The wireless communications is used for the transmission of the base station positioning data to the mobile station for processing. Comparative testing methods of the APPCON APC200 found it to be a suitable alternative to the University proposed HopeRF HM-TR transceivers. The APPCON provides reliable

data transmission, with error correction capabilities while exceeding an operational distances of 850 meters.

Processing the incoming simultaneous stream of pseudoranges from both GPS receiver modules requires a minimum of two hardware universal asynchronous receiver/transmitter (UART) ports, while meeting the system memory and processing requirements. The initial open-source Arduino Duemilanove microcontroller chosen for the project was deemed to be unsuitable after evaluation of two software based UART. The Arduino Mega, having met these requirements, has been used as the prototype microcontroller as it provides four hardware UART ports and largely redundant memory and processing capabilities.

Arduino is based on the open-source software environment called 'wiring' which is a derivative of the C++ software language. The strong online support community for the Arduino has developed an expansive range of open-source software libraries which can be adapted to user projects. Implementing these software libraries, developed as community based projects, means that advanced capabilities and extensively tested code is integrated into the system software code.

The outcome of combining these components has completed the sub objectives of this project and has resulted in a high functioning, reliable, accurate and low cost system. Thorough evaluation of the DGPS prototype has yielded results that substantially exceed the manufacturer rated specifications and prove the benefits of DGPS relative positioning. The average accuracy of the system in an open environment has achieved 0.216 meter averaged distance and an operation range of 882 meters. Testing of the system has been undertaken in a number of environments to evaluate the accuracy and reliability of the system.

The finalized hardware value of the prototype was $467.28, developing an accurate positioning system at a budget price.

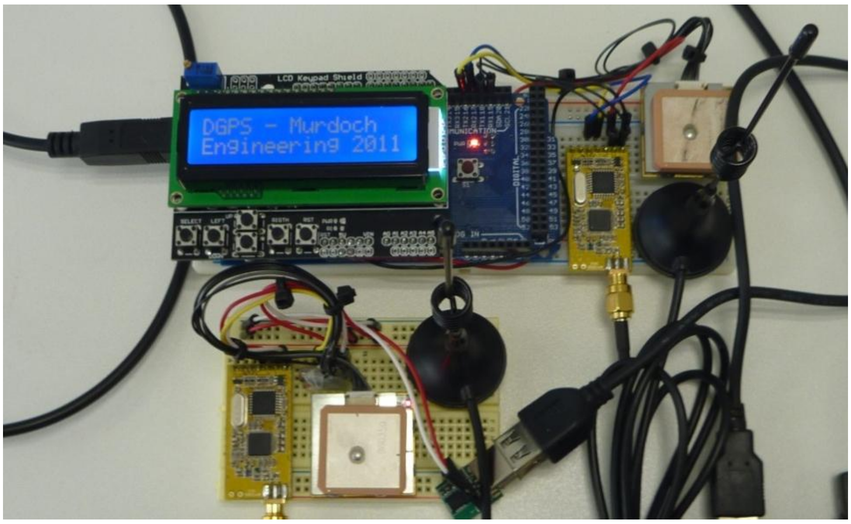


Figure 1: Prototype DGPS

ENG460 Engineering Thesis

Academic Supervisor endorsement pro forma

This is to be signed by your academic supervisor and attached to each report submitted for the thesis.

I am satisfied with the progress of this thesis project and that the attached report is an accurate reflection of the work undertaken.

Signed:

Date:

Acknowledgements

For their assistance and support throughout this project, acknowledgement is given to:

Dr. Gareth Lee

Murdoch University School of Engineering and Energy

Contents

List of Figures

List of Tables

List of Equations

List of Appendices

Terminology and Acronyms

Name / Acronym	Name / Definition
3G	3rd Generation Mobile Telecommunications
4G	4th Generation Mobile Telecommunications
Baud Rate	Data transmission rate
C++	Software Programming Language
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMACA	Carrier Sense Multiple Access Collision
DGPS	Differential Global Positioning System
DoD	Department of Defense
DSSS	Direct Sequence Spread Spectrum
FCS	Frame Checksum
FEC	Forward Error Correction
FPS	Frames Per Second
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HMI	Human Machine Interface
kbps	KiloByte Per Second
L1	GPS SPS frequency - 1575.42MHz
L2	GPS PPS frequency - 1227.60MHz
LCP	Link Control Protocol
MAX232	Converts signals from RS-232 to TTL
mbps	MegaByte Per Second
MD5	Message-Digest Algorithm 5

NATO	North Atlantic Treaty Organization
NMEA	National Martine Electronic Association
O-QPSK	Offset-Quardrature phase-shift keying
OSI	Open Systems Interconnection
PPP	Point-to-Point Protocol
PPS	Precise Positioning System
PRN	Pseudo-Random Noise
PWM	Pulse Width Modulation
RMS	Root Mean Squared
RS232	Recommended Standard 232 - Standard for serial binary single ended-data and control signals connecting between data terminal equipment
RS485	Recommended Standard 485 also known as EIA-485 - Standard for electrical characteristics of drivers and receivers for use in balanced digital multipoint systems
RTO	Research and Technology Organization for NATO
SPS	Standard Positioning System
TCP/IP	Transmission Control Protocol / Internet Protocol
TEC	Total Electron Content
TTL	Transistor-to-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network

CHATPER 1: INTRODUCTION

1.1 Background

1.1.1 Overview of GPS

The Global Positioning System (GPS) has been a project developed by the U.S Department of Defense (DoD) since 1978 and was first operational in 1995. The system was broken into two services: Standard Positioning Service (SPS) using the L1 1575.42MHz frequency and accessible to civilian use. The second service was the Precise Positioning Service (PPS) for authorized DoD use which uses the L2 1227.60Mhz encrypted code frequency (Chaves 2010).

McNeff (2002) presents a thorough and informative paper on the technical aspects of GPS. Initial reading of this paper created a fundamental understanding of GPS and enabled for an in-depth understanding of the objectives for thesis and the reasons as to a why DGPS is required.

1.1.2 GNSS and the Ionosphere

GPS receivers use Global Navigation Satellite System (GNSS) signals to obtain position relevant data. The GNSS signal architecture consists of three components: the carrier, the ranging code and navigation data. The carrier components is the transmission signals through the L1 and L2 frequencies. The ranging code is a set of Pseudo-Random Noise (PRN) sequences, unique to each satellite, for both the SPS and PPS services. Accuracy differences exist between the SPS and PPS services, with PPS producing considerably more accurate data, through a faster update time and a smaller wavelength. Contained within navigation data is information regarding satellite health, clock parameters, satellite orbital parameters and general information about the satellite constellation (Chaves 2010).

This satellite system is the formation of 24 operational satellites plus a number of redundant satellites, which travel at an altitude of 20,200 kilometers above the surface of the earth. The transmitted signals travel through the ionosphere, troposphere and stratosphere which, when combined, form the atmosphere. These aspects of the atmosphere all affect signal transmissions causing refraction of the signal and resulting in changes to the speed of travel. The transmission of the GNSS signals through these layers must account for these affects in the calculations of the positioning process (Garmin 2011).

In-depth research has been completed into the various attributes that cause error in GPS based systems. Jensen and Mitchell (2001) describes in detail atmosphereic affects on GNSS signals. The ionosphere ranges from approximately 80 Kilometers to 1,000 kilometers above the surface of the earth and is a construction of free electrons. The ionosphere is regarded as electrically neutral however, it contains free electrons that are stripped from atoms by ionization. The free electrons are negatively charged causing electromagnetic effects on the GNSS signals in various ways. Predominately the effects cause refraction, which in turn changes the speed of travel 'ionospheric delay' of the signals. The ionosphere delay is caused by the change in refractive index along the path of travel. The refractive index is continuously changing function of the composition of transmission media (Jensen and Mitchell 2011).

Methods of calculating the effects of the ionosphere are used to calculate the signal delay. The delay is modeled by using a constant derived from the values of the electron charge, mass of the electron and the permittivity of free space. This is then divided by the square of the frequency of the transmitted signal from the GNSS satellite (for example 1.57542x10^2mhz for GPS L1). Finally, the Total Electron Content (TEC) is obtained by integrating the number of free electrons along the signal path in a cross section of one square meter. The ionospheric delay is a function of the frequency of the signal as the ionosphere is dispersive for radio waves (Jensen and Mitchell 2011).

Equation 1: Ionospheric Delay

$$d = \frac{40.3}{f^2} TEC$$

The problem with signal frequency receivers is that it is not possible to obtain the actual number of electrons along the signal path. Models to solve this problem have been developed, however they are not as effective as dual frequency GPS receiver or DGPS. If dual frequencies are available, modeling of the ionospheric effect can be more effective. DGPS can achieve accurate results as pseudoranges are calculated using two models with signal transmissions obtained at the same time. This means that technically the signals have travelled very similar paths and the delay experience should be only marginally different if at all.

1.1.3 DGPS Overview

DGPS was developed by the Department of Defense for military applications that required higher accuracy in positioning and distance measuring than standard GPS. (NATO - RTO 2008) The architecture of DGPS uses two or more GPS receivers to simultaneous receive and process signals. These signals are assumed to have travelled the same atmospheric path and hence induced with the same distortion. Using this concept two results can be achieved: differential corrected positioning and relative positioning. Differential corrected positioning is the comparison of received positioning from the GPS to the known location. This difference between the known location is then applied to mobile locations to correct the received positioning data. Differential and relative positioning can also be accurately acquired from the DGPS which assumes the base station is the origin and the mobile station is located on an (x,y) Coordinate.

1.1.4 DGPS Error Sources

There are various sources of error that must be accounted for in the design of GPS. There has been vast research and development on reducing error within GPS and DGPS. Error reduction is an expensive task and will never completely achieve zero. Table 1 is a summary of the possible sources or error effecting standard GPS.

Table 1: GPS & DGPS Comparison (Moore 2002)

Error Source	Standalone (m)	DGPS(m)
Ionosphere	15-20	2-3
Troposphere	3-4	1
Satellite Clock	3	0
Multipath	2	2
Receiver Noise	2	2
Selective Availability	50	0

Note: multipath error is site dependent and cannot be corrected by DGPS therefore the value is an example only. Receiver clock drift is not included as this is corrected with in a standard system and does not add significant error.

Table 2: DGPS Error Source Comparison with Distance (NATO - RTO 2008)

Error Sources	0NM	100NM	500NM	1000NM
Space Segment: Clock Errors	0	0	0	0
Control Segment: Ephemeris Errors	0	0.3	1.5	3
Selective Availability	0	0	0	0
Propagation Errors:				
Ionosphere	0	7.2	16	21
Troposphere	0	6	6	6
Total (RMS)	0	9.4	17	22
User Segment				
Receiver Noise	3	3	3	3
Multipath				
UERE (RMS)	3	9.8	17.4	22.2

It should be noted if DGPS distance separation between mobile and base station receivers is used in excess of 250km, ionospheric and tropospheric delay models should be used. The affects are of separation distance in nautical miles is shown in Table 2.

Communication latency errors could exist in the datalink between the base station and the receiver station referred to ‘age of corrections’. This issue is not included in the above Table 1 and Table 2 as it is not concerned with the GNSS but rather the method and technique of the immediate DGPS design. This issue will only exist if the corrections are out of time and calculated under different conditions. This would result in the pseudoranges being marginally offset.

1.2 Project Objective

The goal for this project is to achieve a low cost, high performance DGPS for inclusion into the Murdoch University School of Engineering & Energy robotic systems. The creation of a DGPS will add significant technical functionality to the robot systems developed by Murdoch University School of Engineering and Energy. Integrating accurate positioning in a robotic system, combined with various other functions, including ultrasonic capability could provide an accurate and reliable platform for autonomous control. Creating such a system could be used as a teaching tool for undergraduate electrical engineering students majoring in Instrumentation and Control and Industrial Computer Systems engineering.

It was encouraged by the university to undertake development using the Arduino platform, an open source microcontroller range. (Arduino. 2011) Arduino is supported by a large developer community and has a selection of evolving open source hardware which can be bought genuine or third party. (Arduino. 2011) The Arduino platform includes a software development suite which comes integrated with open-source software libraries for free use in the development of sketches.

Two other major hardware components will be necessary to make up the system, the EM-406a GPS modules and wireless transceivers. Although these hardware components are proprietary they are to be simply integrated with the chosen Arduino microcontroller and no further development is required. The HopeRF HM-TR wireless RS232 transceivers are to be evaluated for reliability and usability and if they do not meet the benchmark requirements of these components alternate replacements will be sought. Further evaluation of wireless transceivers will need to be undertaken, to find the best suited for implementation into the system.

Upon construction of the hardware prototype, the software provides complete integration of the hardware and to successfully process positioning data received from the receivers. The software will utilize the received positioning data, to accurately positioning through using advanced positioning functionality. The system will provide accuracy to within meters and preferably tens of centimeters. Table 3 lists the minimum requirements of the DGPS hardware for the system to achieve the set outcomes in Table 4.

Table 3: DGPS System Requirements to achieve Outcomes

Component	Requirements
GPS Receivers	1. Accurately produce positioning data which can be communicated to the microcontroller via TTL. 2. Low cost 3. Low power consumption
Communications	1. Reliably transmit and receive communications from the receivers to the system 2. Operational distance of at least 50m 3. Work in any environment and meet distance requirements 4. Error connection is desirable 5. Low power consumption 6. Multiple nodes on transmission medium
Microcontroller	1. Sufficient computational power and memory for calculations 2. Minimum of 3 UART / Serial Communications channels 3. Low Power consumption

Table 4: DGPS System Outcomes

Aspect	Outcomes
Accuracy	1. Within 2 meters 2. If first is satisfied then within 10s of Centimeters 3. Reliable output – Refreshing every one second
Operation	1. Operational distance of minimum 50m 2. Operational environment of universities, commercial and residential 3. Update frequency of 1 second or greater 4. Lower power consumption 5. Flexible power source 6. Operation time of Base station 5 hours on battery
Output	1. Serial Communications output 2. Digital display 3. Positions, distance, speed, bearing, relative positions
Cost	1. Low cost and high efficacy

1.3 Project Structure

Engineering projects require a strong framework and schedule if they are to successfully complete their objectives. Even though the project was to be undertaken by only one person, a management strategy still had to be created to ensure work was completed for the due date and that all quality aspects were observed. This thesis project was broken into five major stages: research, design, construction, evaluation and documentation. Research of the differential system was the initial step and continued throughout the design and construction stages. In-depth research ensured a fundamental base for which the project could be proposed and then developed. Of particular benefit, was the research into the Arduino software community which contained libraries of open-source software used to enhance the system.

The design undertaken was a collaboration of preexisting software libraries that were integrated to operate on a single microcontroller. In the construction phase any changes or issues are redesigned and then reevaluated. During this stage all components for the system were procured to prevent delays in equipment delivery.

Construction is the integration of the design and research phases of the project. It is important that good preparation has been completed to prevent disruptions and delays. Any major design issues are reengineered to meet the project outcomes. During the construction phase the software was written so it could tested concurrently with the prototype hardware. Using the prototype hardware was crucial to identify any flaws or issues with the software. Quality must be guaranteed for the project to meet the objectives through evaluation. Each of the three major parts of the project had different testing strategies that were fulfilled to enable the project to reach the final outcome.

At the completion of the evaluation stage all results and findings were thoroughly documented. This allows for the project to be implemented or expanded by future students. All software should has clear annotation with additional information included on calculations.

CHAPTER 2: HARWARE

2.1 Overview

The project was initially based on two open source Arduino microcontroller designs. The first was a single microcontroller system which used a base station with a GPS receiver connected to a transparent RS232 wireless bridge and a mobile station with the microcontroller for processing. This system sourced as a cost effective design with all processing of the data to be completed through the mobile station. The use of low resource base stations could potentially expand into multiple base stations and make use of a triangulation configuration. The single microcontroller concept system is shown in Figure 2.

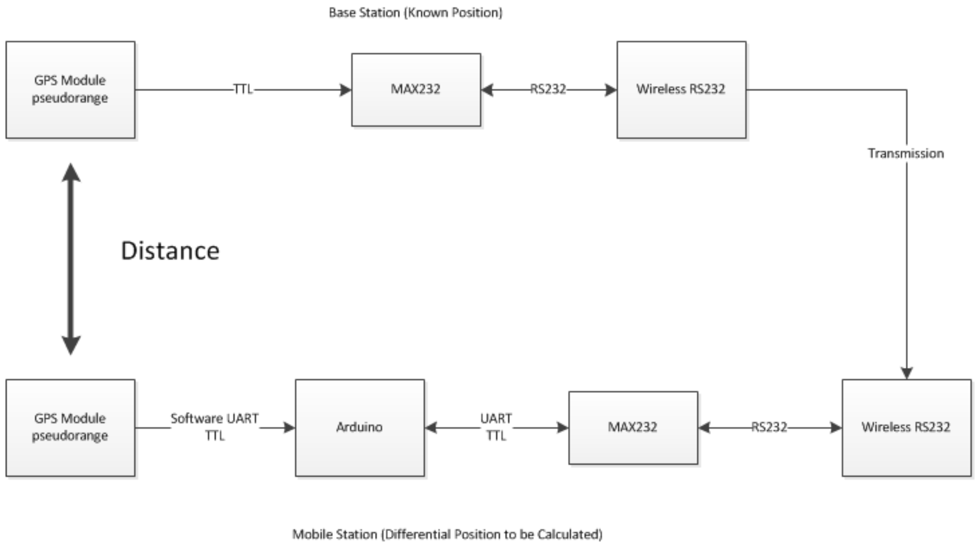


Figure 2: Single Microcontroller Original Design

The second concept configuration used two microcontrollers to allow for bi-directional processing of data. This meant that there would be a microcontroller located at the base station which would calculate the averaging coordinates and then send to the mobile station microcontroller. The differential position could be established and transmitted back to the base station for tracking. This configuration however did not present any major operational advantage and the cost of adding an additional microcontroller could not be justified. The dual microcontroller concept configuration was not built into a prototype or evaluated in this thesis. The dual microcontroller concept system is shown in Figure 3.

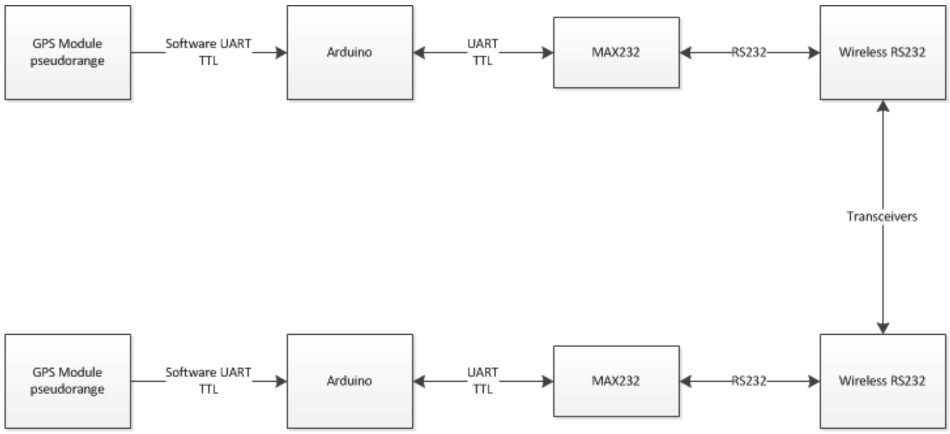


Figure 3: Dual Microcontroller Original Design

The final configuration of the prototype system shown in Figure 4, which deviated from the original single microcontroller with use of hardware UART and TTL and the removal of the RS232. Additional hardware functionality has also been implemented into the system.

This chapter explores the various hardware components and configurations evaluated to meet the requirements of the system. Communications hardware has been included under Chapter 3.

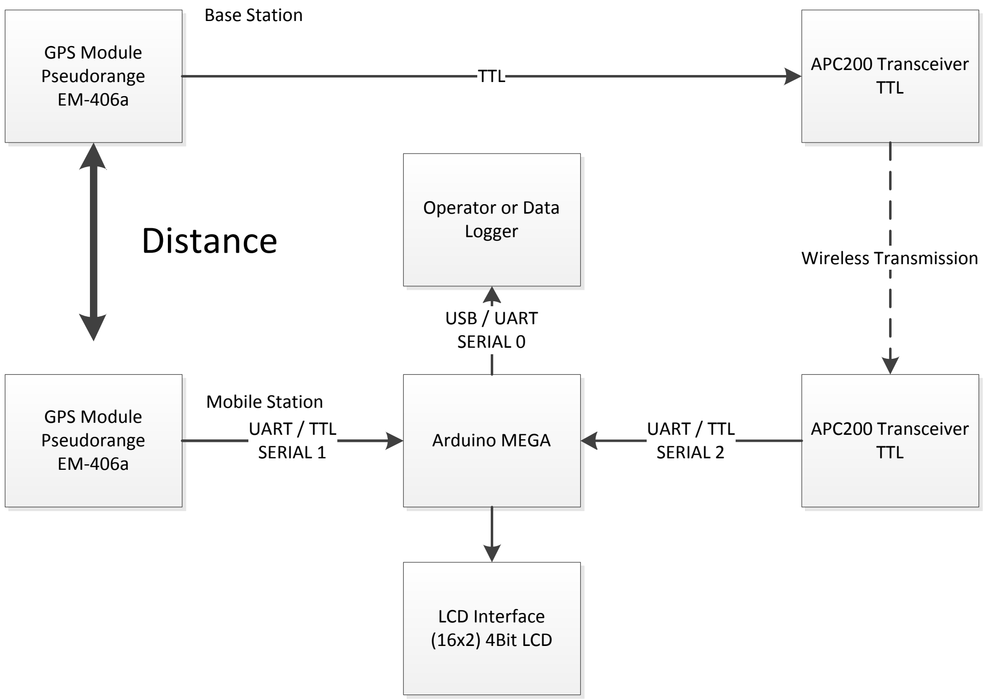


Figure 4: Final System Design

2.2 Design

The design was created by integrating components. This section lists the major hardware components and discusses the relevant specifications of each.

2.2.1 Arduino Duemilanove

Duemilanove in Italians means '2009' and is symbolic to the 2009 model Arduino. The Duemilanove was supplied by Murdoch University as the original microcontroller for the thesis to be based upon. The Duemilanove appeared to have sufficient processing power for the differential algorithms, however lacked communication ports known as UART (Arduino 2009). This is covered further in Section 3.2.3 of this paper.

Table 5: Arduino Duemilanove Specification (Arduino 2009)

Summary	**Value**
Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

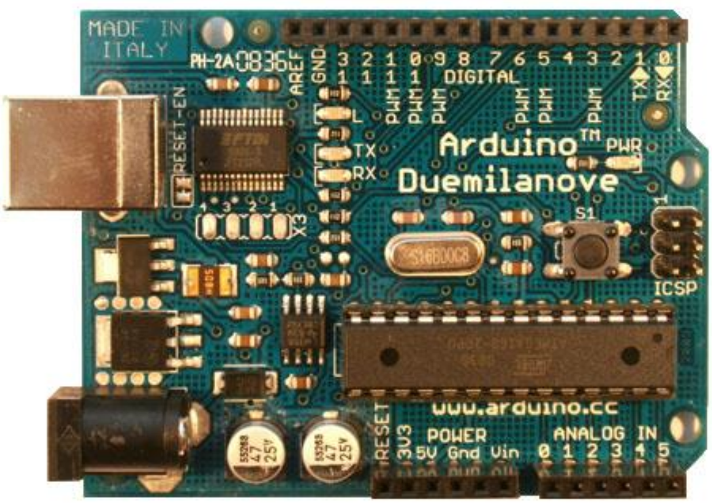


Figure 5: Arduino Duemilanove (Arduino 2009)

2.2.2 Arduino Mega

The Arduino Mega microcontroller uses an ATmega1280 for processing its 54 digital inputs/outs, 16 analog inputs and 4 UART. The Mega has a substantial increase in microcontroller capability from the standard Arduino Duemilanove 168 and 328 with its 16Mhz clock speed and 128kB of flash memory. The Arduino Mega has been used in the thesis prototype system. (Arduino. 2011)

Table 6: Arduino Mega Specifications (Arduino. 2011)

Summary	Value
Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

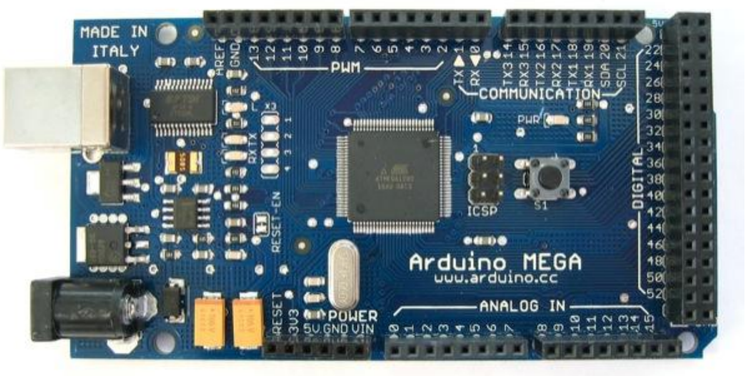


Figure 6: Arduino Mega (Arduino. 2011)

2.2.3 EM-406a GPS Receivers

The EM-406a GPS engine board produced by Global is a low cost, highly accurate system, suitable for integration into robotic systems. On this basis, the module was provided by Murdoch University at the commencement of the project and used on the prototype system. The following is a list of the features of the EM-406a: (GlobalSat Technology Corporation 2007):

1. SiRF Star III high performance GPS chipset
2. Very high sensitivity (Tracking Sensitivity: -159dBm)
3. Extremely fast TTFF (Time To First Fix) at low signal levels
4. Supports the NMEA 0183 data protocol
5. Built-in SuperCap to maintain system data for rapid satellite acquisition
6. Built-in patch antenna
7. Foliage Lock for weak signal tracking
8. Compact in size
9. All-in-view 20-channel parallel processing
10. Snap Lock 100ms re-acquisition time
11. Superior urban canyon performance
12. WAAS / EGNOS /MSAS support
13. RoHS compliant

Figure 7: GlobalSat EM-406a GPS Engine Board (GlobalSat Technology Corporation 2007)

Table 7 includes the important specifications of the EM-406a and Appendix DGPS008 contains the complete list.

Table 7: Important specifications of the EM-406a (GlobalSat Technology Corporation 2007)

Field	Value
Chipset	SiRF Star III
Frequency	L1, 1575.42Mhz
Channels	20 Channel
Sensitivity	-159dBm
Accuracy	10 meters
Velocity	0.1m/s
Main Power	4.5V-6.5V DC
Power Consumption	60mA
Communications Electrical level	TTL 0V-2.85V
Baud Rate	4800 bps
Output Message	NMEA 0183 GGA, GSA, GSV, RMC

The EB-85A GPS engine board is a newer type of receiver using a refresh rate of 5Hz which presents a major advantage for real time systems when a fast update is required. The restriction of this GPS engine board is the baud rate requirement of 38400bps. The use of the EB-85A GPS engine board is recommend in future expansion of the DGPS system as the alternate APC200 & APC220 wireless transceivers will support this higher baud rate (Sparkfun Electronics 2009).

2.2.4 LCD Display & Keypad

Although an LCD screen was not in the original requirements of the system, the low cost and 4 bit communications was an appropriate functional addition to the system. The LCD shield uses a HD44780 controller and schematic drawings are available on the supplier website, Little Bird Electronics. (LCD Keypad Shield 2011) This LCD controller is commonly used in Arduino projects and therefore an open source library exists for integration into projects (LCD Keypad Shield 2011).

Figure 8: DFRobot LCD Keypad Shield (LCD Keypad Shield 2011)

Table 8 contains useful information and links for the DFRobot LCD Keypad.

Table 8: DFRobot LCD Keypad Shield Tools and Links

Tool	Link	Verified
Australian Supplier	http://littlebirdelectronics.com/products/lcd-keypad-shield	10/6/11
User Guide	http://www.robotshop.com/content/PDF/dfrobot-arduino-shields-manual.pdf	10/6/11
Shield Schematic	http://www.robotshop.com/content/PDF/dfrobot-lcd-keypad-shield-schematic.pdf	10/6/11
Arduino Library	http://www.robotshop.com/content/ZIP/dfrobot-lcd4bit-mod.zip	10/6/11

2.2.5 High Gain 433MHz Antenna

To improve the performance of the transceivers the standard 433Mhz antennas were replaced with high gain 433 MHz antennas. A notable increase in performance was seen in the evaluation of the APC200 transceivers.

Table 9: Contact Information for Antennas

Contact details	Link	Cost
3W~DIY-cable~KOM, 深圳市, 广东省, China	http://cgi.ebay.com.au/ws/eBayISAPI.dll?ViewItem&item=270680271502&ssPageName=STRK:MEWNX:IT	$6.00 US incl. Postage

Table 10: Technical Specifications of the Antennas (lm4871n 2011)

Technical Data	Value
Frequency Range (MHz)	433MHz
VSWR	≤1.5
Gain	3.0dBi
Input Impedance	50Ω
Polarization Type	Vertical
Antenna length	155mm
Cable Type	RG174
Cable Length	1.5 meters
Connector Type	SMA male Plug
Mounting	Magnetic mounting base
Magnetic base diameter	30mm
Colour	Black

Figure 9: Black 433MHz High Gain Antenna (lm4871n 2011)

2.2.6 PPS2 Duracell Li-Ion Battery

To maintain reliability throughout the system, Duracell USB Lithium ion batteries were used as a rechargeable battery source. These batteries were purchased from Bunnings Warehouse for approximately $30 AUD each and presented a preferred alternate to disposable alkaline batteries.

Table 11: PPS2 Duracell Battery Technical Specifications

Technical Data	Value
Input	5V, 0.8A max
Input Connector	USB micro
Output	5V, 0.6A, 3W max
Capacity	1150mAh

Figure 10: PPS2 Duracell Rechargeable Li-ion Battery

CHAPTER 3: COMMUNICATIONS

3.1 Overview

Communications was a major component of this thesis as thorough research into RS232 or TTL transceivers had not taken place previously at Murdoch University. The Engineering department had previously purchased RS232 type transceivers to be evaluated for future implementation into the DGPS and other robotics based systems. If evaluation proved that the transceivers were suitable for use by the DGPS then further design would need to address the issue of limited hardware communication channels, known as UARTS. This chapter investigates the various aspects of the communications system used in the DGPS combined with the evaluated hardware and software, to find a suitable system that would meet the defined requirements.

3.2 UART Communications

3.2.1 Software Serial

The Arduino Duemilanove microcontroller originally selected provided one hardware Universal Asynchronous Receiver/Transmitter (UART) to be used for operator communications or GPS communications. UARTs are used to translate data between integrated circuits in both parallel and serial forms. This created a problem, as two communication channels were needed for the GPS EM-406a modules. After further research and correspondence with Dr. Gareth Lee, it was suggested to evaluate the standard Arduino library Software Serial. Using this library would create a defined number of software UART ports through two digital I/O pins per software UART. The software serial library is limited as it lacks some functionality of the hardware UART, such as interrupt based communications and baud rate.

One issue in particular that poses major problems is the lack of the Serial.Available() function resulting in the microcontroller not being flagged when there is new data available in the buffer. The Serial.Available() function indicates whether any bytes are stored in the buffer however the standard software serial does not have buffer. Therefore, the only function that works with this library is Serial.read(). This receives data only when it is being called, also known as polling, and any other data that is received when the function is not called is lost. Other issues that have been noted include timing issues and data corruption when using higher baud rates. Using this library would mean that the microcontroller would only be able to receive the pseudoranges from the GPS receiver and would have limited CPU time to process other functions. The limited processing time would mean the microcontroller was too busy capturing all the NMEA characters and would not be able to parse this data into a usable form. This method was therefore going to fail and further research would be required on how to find suitable code or changing of the hardware.

Further research into methods of software UART emulation found that a third party software programmer had created a library that enhanced the standard software serial. The new library titled 'NewSoftSerial' was developed by Mikal Hart and provided considerable enhancements over the standard method of software UART (Hart, Arduiniana 2011). The NewSoftSerial was based on Ladyada's AFsoftwareSerial which introduced interrupt-driven receives that replaced the standard method of polling the software serial functions. This interrupt structure allowed for the implementation of the Serial.Available() which enabled the microcontroller to read from the software buffer instead of having to constantly poll short regular intervals (Hart, Arduiniana 2011).

Table 12 includes a list of enhancements over the standard software serial library. It is expected that the Arduino community will replace the standard software serial with the NewSoftwareSerial developed by Mikal Hard in the Arduino IDE.

Table 12: List of Additional Support Features in NewSoftSerial

1	Inherits from built-in class *Print*, Eliminating some 4-600bytes of duplicate code
2	It implements circular buffering scheme to make RX processing more efficient
3	Extends support to all Arduino pins 0-19 not just the standard 0-13
4	It supports multiple simultaneous soft serial devices
5	It supports a much wider range of baud rates
6	It provides a Boolean overflow() method to detect buffer overflow.
7	Higher Baud rates have been tuned for better accuracy
8	It supports the ATMega328 and 168
9	It supports 8Mhz processors
10	It uses direct port I/O for faster and more precise operation
11	It supports software signal inversion
12	Supports 20Mhz processors
13	Runs on Teensy and Teensy++
14	Supports an end() to complement begin()

After integration of the NewSoftwareSerial library into the DGPS software it became apparent that CPU and memory would be severely limited by the time taken to process incoming communications. This method of communications would limit the software and system's ability to do advanced differential calculations which use a large number of processor intensive trigonometric algorithms. Using hardware UARTs would remove the issue of processor time on software communications and therefore develop a more advanced positioning system. The Arduino Mega hardware was chosen for this reason.

This research into the software serial functionality for the Arduino was well spent and beneficial to the engineering department which has purchased a number of Arduino Duemilanove controllers. The library 'NewSoftwareSerial' has been included as Appendix DGPS003 for future implementation into less processor intensive projects.

3.3 HopeRF HF-TR RS232 Version

3.3.1 Overview

The original intention was to use the HopeRF transceivers for wireless communications, configured to transmit RS232 signals transparently. The HF-TR transceivers were pre-purchased by the University for use in microcontroller or robotic applications. To implement these transceivers into a project meant that testing and evaluation was necessary. Literature surrounding the topic was reviewed to find the best method of evaluation and the additional components required.

An evaluation board was already provided and was a construct of Dr. Gareth Lee's design. For point to point testing, two of these evaluation boards would need to be used and connected to two separate RS232 communications ports of the testing PC. The evaluation board provides the ability to breakout the pins of the transceiver so RS232 signals can be connected. The evaluation board also has its own power conditioner to regulate the attached 9Volts battery to 5Volts. Table 13 below lists the components and equipment required. A second evaluation board was configured for testing between terminals (in this evaluation, computer to computer).

Table 13: HopeRF HM-TR Equipment List

QTY	Item	Description
2	HF-TR evaluation boards	Originally designed by Dr. G Lee these boards breakout the connections for the HF-TR transceivers allowing connections to be made to the RX and TX pins.
2	3.5mm Socket to RS232 Cable	This cable is used to carry the RS232 signal from the computer or testing equipment to the HF-TR evaluation board.
2	HF-TR Transceivers	Module to be tested
2	9V Battery	Power the evaluation board and using the onboard MOFET to regulate the voltage to 5V VCC.
2	Telnet / PuTTY	This will be used to test RS232 signals.

Table 14: PuTTY Telnet Client Homepage

Tool	Link	Verified
PuTTY	http://www.chiark.greenend.org.uk/~sgtatham/putty/	10/6/11

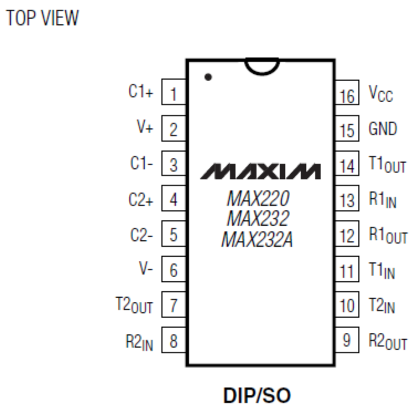
Figure 11: HopeRF HM-TR RS232 (Hope Microelectronics CO. LTD 2008)

Implementing the design shown in Figure 2, research project required the use of wireless RS232 bridges. HF-TR transceivers come in two models, one with an inbuilt max232 integrated circuit to convert the RS232 signals received into TTL protocol which it then transmits and receives. The second processes direct connection TTL signals from the microcontroller and transmits without conversion. Murdoch University has already pre-purchased RS232 HF-TR transceivers which includes an embedded MAX232CPE converter. The Arduino Duemilanove and Mega use TTL communications to the GPS modules, which are also TTL native. The HopeRF HM-TR RS232 modules mismatch communications between the microcontroller and the transceivers and therefore an additional conversion circuit must be used. The difference between TTL and RS232 is the signaling voltage of 0-5 Volts and 0-12 Volts where by the higher RS232 "high" voltage of 12 Volts would damage electronics operating at the TTL 5 Volts "high" voltage. High voltage is the signaling between low and high communications.

The design of this conversion circuit will include a MAX232CPE IC which will convert between the TTL microcontroller and the RS232 MAX232 transceiver. Converting the signals from the microcontroller would be transparent and therefore would not affect the communications. Research was conducted on the various types of MAX232 conversion chips available and it was decided a Maxim MAX232CPE would be best suited. Figure 12 shows the schematic and the configuration that must be used to correctly operate the MAX232. Using this configuration enabled the conversion of TTL to RS232 between the Arduino and the transceivers and vice versa. Further information on the MAX232CPE IC operation is available on the Maxim website show in Table 15.

Table 15: MAX232CPE IC External Reference

Tool	Link	Verified
MAX232CPE Information	http://www.maxim-ic.com/app-notes/index.mvp/id/2020	10/6/11



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

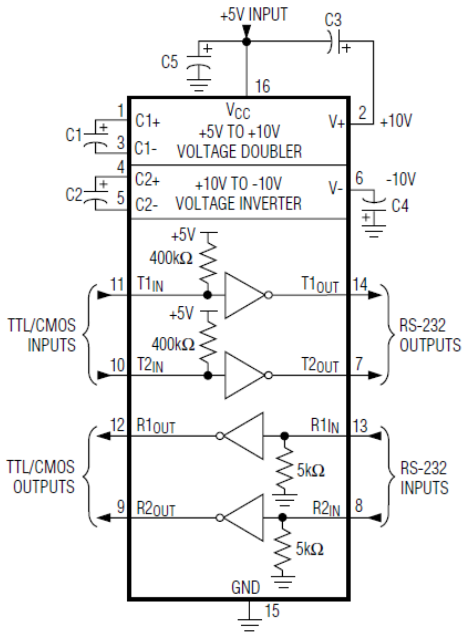


Figure 12: MAXIM MAX232CPE (MAXIM 2006)

Testing these transceivers and the MAX232 conversion circuits resulted in finding a particular problem with the GPS modules. The EM-406a modules were the TTL protocol however only signally an output high of a maximum 2.85 Volts. This did not meet the required MAX232 TTL high of 3.5 Volts. To overcome this limitation a Schmitt trigger (MM74C14) was used feed in the EM-406a voltage in which it would output at the correct “high” voltage. To achieve this however the Schmitt trigger inverts the signal therefore requiring the signal to be fed into a second input of the Schmitt trigger to correct. The Schmitt trigger amplified the signal from 2.85 Volts to approximately 3.8 Volts (FairChild 1999). These voltages were viewed and monitored during testing using an oscilloscope. This circuit is shown below in Figure 13.

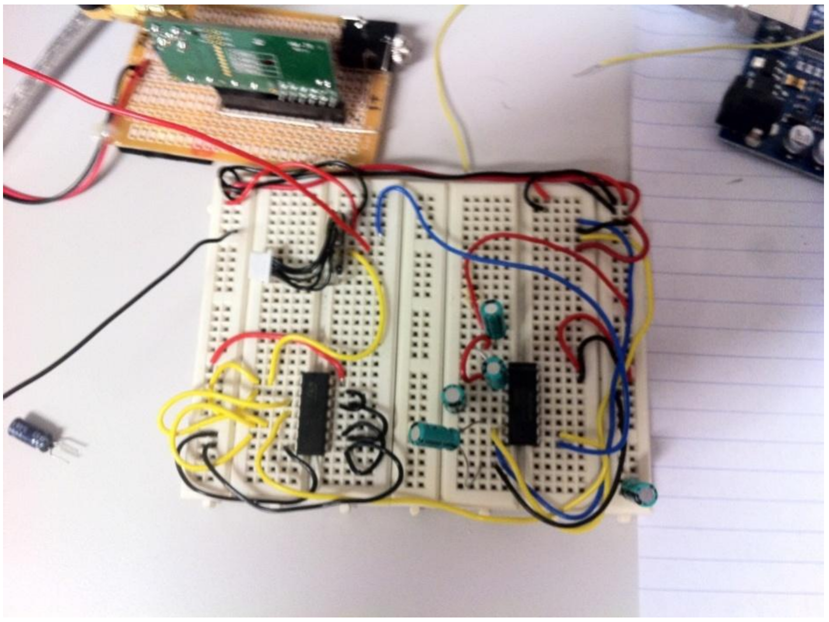


Figure 13: Schmitt Trigger & MAX232CPE Conversion Circuit

3.3.2 Configuration

The HopeRF HM-TR transceivers use propriety software to write configuration parameters to the modules. The configuration is then stored in the EEPROM of the onboard ATmega of the transceiver. Due to the severe lack of technical documentation on the HM-TR transceivers, configuration was achieved through a trial and error method. For the software to function correctly the module must be connected to default communications port 1. The module must also have a high voltage to the 'config' pin at startup for the transceiver to accept configuration parameters.

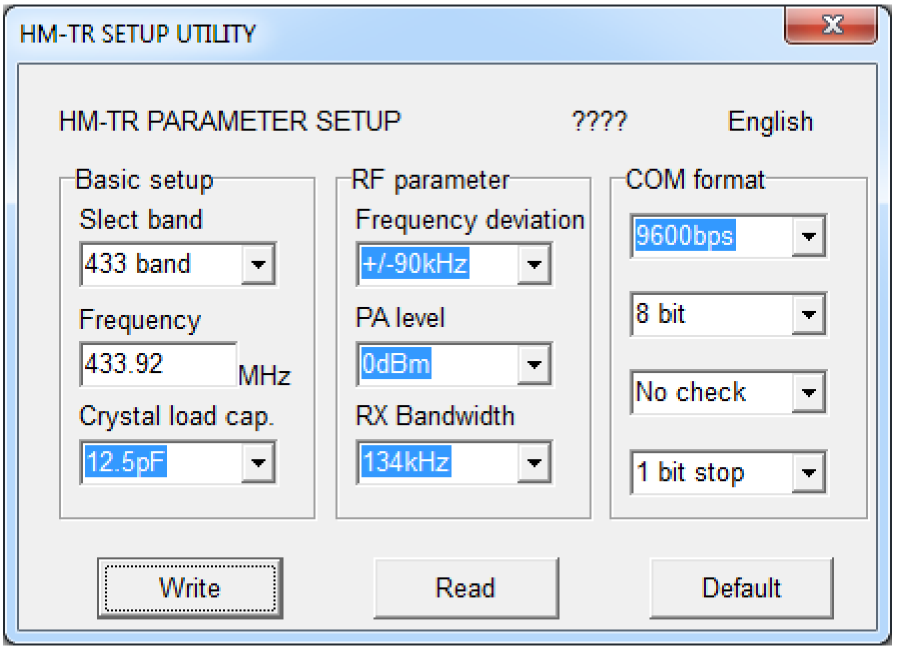


Figure 14: HopeRF HM-TR Configuration Software

3.3.3 Evaluation Method

For the evaluation and testing to be thorough, an evaluation method created and followed. The evaluation method and reason for testing is listed in Table 16.

Table 16: Communications Testing Aspects

Item No.	Component	Description / Notes
1	Evaluation Environment	Technically this affects the results significantly because there were two test environments used outdoors and indoors. It is assumed that the system will be used outdoors where GPS signals can be received by the GPS modules. Other environmental factors can affect the testing such as moisture from the recent rain which has been soaked up by trees. The 433mhz spectrum is also not regulated and freely used therefore interference could be a common issue.
2	Distance	This is a standard test as signal deteriorates over distance, the quality of the connection will assume the same rule.
3	Baud Rate	The rate of transmission is affected by distance. Slower connections are technically more reliable however minimum transmission rates must be met for GPS data. I.e. (4800bps)
4	Stop/Start bits	This can add additional reliability to the data connection
5	Testing Data	Various different forms of information were to be tested on these transceivers from different characters types and different sizes of strings.

Environment is a key factor in where the testing and evaluation are conducted. These environments should be where practical use of transceivers will take place, as this would introduce any external factors not accounted for in the strategy. Two test environments were chosen to help isolate any area specific interference and in particular on the open 433MHz spectrum. The first environment was the Murdoch University Physical Sciences building for indoor tests and Bush court for the outdoor tests. This Campus environment was chosen because this is the most practical place to evaluating these transceivers. However, due to the nature of it being a research institute, additional interference should be considered. The secondary test environment used to isolate any interference was South Yunderup, Western Australia. South Yunderup was chosen as it is classed as a regional area in which there is very limited commercial or industrial significance. Further testing was also to be conducted on the Mandurah estuary, as the manufacturer lists the ideal environment of operation on water, however this was deemed not required due to the evaluation results covered in 3.3.4 (Hope Microelectronics CO. LTD 2008).

Distance was presumed to be the limiting factor in the operation of these transceivers and therefore small increments were chosen to estimate the range. These distance increments were increased in both indoor and outdoor environments to demonstrate the impact of brick and construction materials on the signal. Outdoors the transmitters ideally should be capable of operating more than distances of over 100 meters, however this is still very dependent on interference. If the distances achieved are less than expected, alternate solutions would be researched including higher gain antennas and signal power amplifiers or using an alternate technology.

The baud rate has to meet the minimum requirement of 4800bps to allow the GPS receiver modules to transmit the GPS packets. The testing and evaluation will use 4800bps, however if results are better than expected the baud rate can be increased to allow for future project expansion. If the multiplexing of data is to be used then this baud rate will need to be higher to allow for the increase in data being transmitted.

Using different types of testing data is important to demonstrate any issues with particular characters or strings. Therefore, alphabetic and numeric characters were tested and different strings were used to identify problems. After character testing, the GPS data was sent from an EM-406a module over the transparent RS232 Bridge and checked for inconsistencies or errors. Using the GPS data is the true test as this is the predominant form of data to be transmitted.

Included below in Table 17 is a list of tools used in the testing of data over the communications links.

Table 17: Checksum & CRC Tool List

Tool	HTTP Link	Verified
Random text generator	http://johno.jsmf.net/knowhow/ngrams/index.php?table=en-finnegan-word-2gram&paragraphs=20&length=100	10/6/11
Random number generator	http://www.random.org/integers/	10/6/11
Online CRC generator	http://www.lammertbies.nl/comm/info/crc-calculation.html	10/6/11
C++ CRC generator	http://www.networkdls.com/Software.Asp?Review=22	10/6/11
MD5 Generator	http://www.miraclesalad.com/webtools/md5.php	10/6/11

Using the character text generator, the following output was generated in Table 18 and the random number generator output in Table 20. This data was transmitted over the transparent bridge to test the reliability and to identify if an error occurred. For comprehensive testing, each set of data was associated with a generated CRC-32 and MD5 Checksum then compared at the receiver for error or corruption.

Table 18: Random Character Generator Output

6d0ala397gl9d8p8typ4aryzwlbq8natln7oc9sddoyal564ofivfcq,esv6pgxl7bpwvscrrtsm,9d9b1ujc3avfd38k9wpkoqnertb8euxwrfz3u20mdliquy4im1q8rj9u3lg6,jadhhejsqj709f3rjo73qnv9x1m4c5v9qk9s80a2tkzvyhy,vcsayexcr7y89iotar352gcmc8uyjanjj8jh85jjqh,f34lpetde5v4o6sg1prfk,auey99vcsuq9c81omn27ykf9vr1ery81gws65lx86vxfiozsdfun7sut,hee33kx2t76whxcnlgxu,xcw2h0agacdy41qshz2irqe87,gq6vnwevg54p3vl2t0y7djblj9nih2kos,21eum4wvvn43976fcl4m4id5dfm7hukaujy37lo8it3,qay9,k8yn3w6hpkjx3y,ha0evjl79m6lo00p7cozbi70kvhdxgcgo95,1sdwhtkvefzhklj2rvcqh3,gzxpp5tfir6rhwdpweir3makotpcurpy77eq8o0eft2qc5v2ura795i,l2eloabkc2bqw1g79e62mcsnuxus4e67e2j,egm0q6gkr5hojcmvztcy2mvz12u73hmqi6tcdjusnn9sflb3pofhn00m29bd4321k897jqz57y7k,te5v2s9h626bv33no5p3a5loog2uqr1ae4tv,gqgjhw7mts98q0hsu0qbmxmoqo6mr29sm6ofjrxbs12

Table 19: Character Test Checksums

CRC Type	Checksum
CRC-32	0x5F439683
MD5 Checksum	e4b8e07d12e6c9c5b38d42a9d0cd8dfe

Table 20: Random Number Generator Output

12304713294918251982645918273198234098120348132809412097509817340986172093741894598346587610928749328741093274081623841312041234091273049183240912865087164591871875198275981349861562458971239847198257189346598375320984723947109238470915619237419283570891465091837459134197324089174258971357132984710385612346128576138475628034629479123874912865982635913485917320497102984710298561189456159283749918327441932740189263508913426358917456914322081222389513489571348957349587458758793148718791874491862958914118600232623673567469915157239487129371987349879938475092384750982734591701023401234102394019328641932468173401401234680127364817263048712064612

Table 21: Numeric Test Checksum

CRC Type	Checksum
CRC-32	0x3637E9F1
MD5 Checksum	183157cf59a26d139aa69576b0d612fa

.

3.3.4 Evaluation Results

The results from testing showed mixed performance, with corruption and identified issues that would need to be rectified before implementation. As expected, the testing methods shown in Table 22 highlighted the major issues and the percentage of failed checksums over time.

Table 22: Communications Test Results

No.	Environment	Distance	Baud	Bit	Data	Results
1	Indoors (open space)	0.5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Very Good – 2% Notes: TX data matched RX data
2	Indoor (open space)	5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data. Slight time delay (latency)
3	Indoor (open space)	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data. Slight time delay (latency).
4	Indoor (between rooms – Brick walls)	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Poor – 50% Notes: TX data did not match RX data. With only some alphabetic characters working.
5	Indoor (between rooms – Brick walls)	15m	4800	1	Testing of Alphabetic and Numeric data	Quality: Failure – 100% Notes: No data was communicated.
6	Indoors (open space)	0.5m	4800	1	GPS Data NMEA packets	Quality: Average – 30% Notes: Data was transmitted and received however lines where not received separate and were joined. This is concerning for GPS packet separation
7	Indoors (open space)	5m	4800	1	GPS Data NMEA packets	Quality: Average – 30% Notes: Data was transmitted and received however lines where not received separate and were joined. This is concerning for GPS packet separation
8	Outdoors	0.5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 2% Notes: TX data matched RX data. Slight time delay (latency).
9	Outdoors	5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data. Slight time delay (latency).

10	Outdoors	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data. Slight time delay (latency).
11	Outdoors	50m	4800	1	Testing of Alphabetic and Numeric data	Quality: Good – 8% Notes: TX data matched RX data
12	Outdoors	~90m	4800	1	Testing of Alphabetic and Numeric data	Quality: Average – 30% Notes: TX data did not match RX data. Time delay (latency). There was evidence of interference approx 95% of data was received correctly
13	Outdoors	0.5m	4800	1	GPS Data	Quality: Failure – 0% Notes: Data was transmitted and received however lines were not received separate and were joined. This is concerning for GPS packet separation
14	Outdoors	5m	4800	1	GPS Data	Quality: Failure – 0% Notes: Data was transmitted and received however lines were not received separate and were joined. This is concerning for GPS packet separation
15	Outdoors	10m	4800	1	GPS Data	Quality: Failure – 0% Notes: Data was transmitted and received however lines were not received separate and were joined. This is concerning for GPS packet separation
16	Indoors (open space)	0.5m	9600	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data.
17	Outdoors	10m	9600	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data.
18	Outdoors	50m	9600	1	Testing of Alphabetic and Numeric data	Quality: Good – 5% Notes: TX data matched RX data.

3.3.5 Evaluation Review

The results from the evaluation process were not as expected due to problems with data quality on short distance links. The most significant issue was the corruption of GPS sentences, which would cause the GPS parsing software to fail (discussed in Chapter 4). The distance achieved from the testing and evaluation was limited in all environments including line of site. The baud rate was initially adjusted, however using the final system design, increasing the baud rate was not required and not practical to implement.

These transceivers also had a lack of documentation on their technical aspects, such as the possible use of error-correction. From the testing results above it is concluded that an alternate transceiver needed to be used to achieve the objectives of this project.

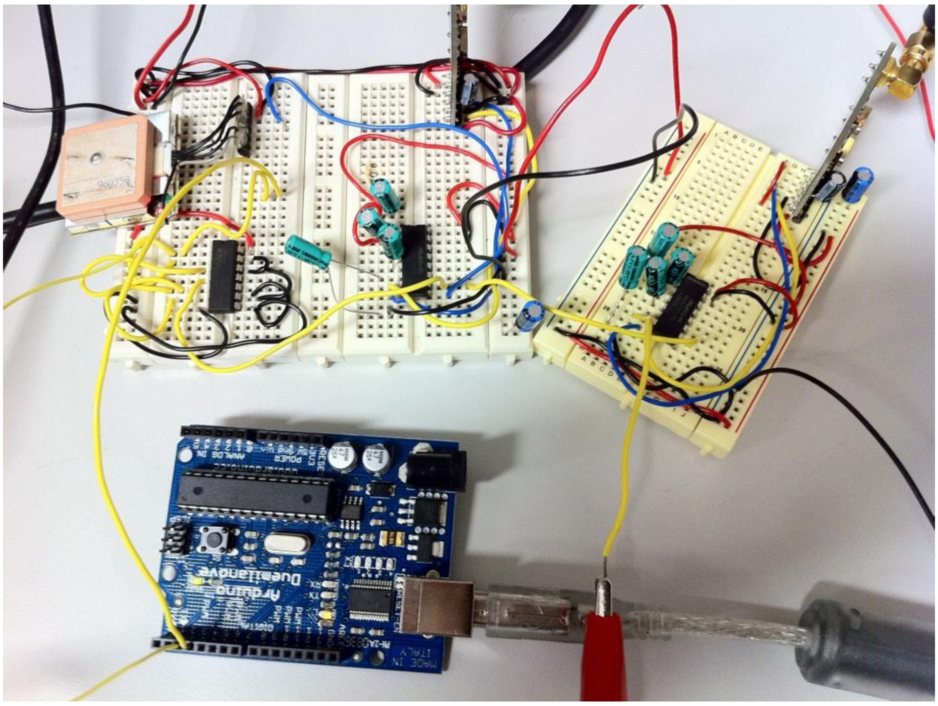


Figure 15: Arduino Duemilanove and Wireless Transceiver Evaluation

3.4 HopeRF HM-TR TTL Version

The HopeRF HM-TR TTL modules were ordered from Futurlec Australia but were not received until late May from Thailand. Due to the issues experienced in the RS232 version it was assumed that these would have consistent in the TTL version. Due to time constraints, evaluation of these modules was restricted to configuration of parameters and brief operational testing.

Figure 16: HopeRF HM-TR RS232

The contact details for Futurlec the supplier for HopeRF have been included below in Table 23.

Table 23: HopeRF Contract Details

Australasia	Futurlec PO Box 24, Broadmeadow, NSW 2292 Australia Fax No. : 02 94 754 051 Fax No. - International: + 61 2 94 754 051
Asia and The Middle East	Futurlec 1108/15-16 Sukhumvit Rd, Phra Kanong, Bangkok 10110 Thailand Fax No.: +66 2 712 1658
US and North America	Futurlec 1133 Broadway, Suite 706, New York, NY 10010 USA Fax No.: +1 212 714 8018

3.5 ZigBee Communications Standard

3.5.1 Overview

After the disappointing evaluation results of the HopeRF modules an alternate working solution was necessary to successfully complete the objectives of building and evaluating a differential global positioning system. The first alternate module to be researched was the ZigBee modules, which would replace the 433MHz frequency with a 2.4GHz alternate. Further research was conducted into these devices however with the desired low resource configuration of the DGPS system this module became less desirable.

Figure 17: XBee Pro (Digi International Inc 2009)

ZigBee is the IEEE802.15.4-2003 standard for low rate wireless personal area network (WPAN) which was designed to be simpler and more cost effective compared to other WPANs. The ZigBee standard provides a secure and low-power consumption networking standard that can be used to create mesh networks for wireless control and monitoring applications. The ZigBee specification uses offset-quadrature phase-shift keying (O-QPSK) and direct sequence spread spectrum (DSSS) to combine and produce a communications method with excellent performance in low signal-to-noise ratio environments. ZigBee also uses technology named Carrier Sense Multiple Access with Collision Avoidance (CSMACA) to listen to the channel before transmitting to avoid talking over another radio. This increases reliability by eliminating the possibility of corrupt data from cross talk. Error detection and correction is achieved through the use of a 16-bit CRC Frame Checksum (FCS) on each packet (Digi International Inc 2009).

The XBee module is produced by MaxStream and designed to the ZigBee IEEE 802.15.4-2003 standard. The module is ideal for use with microcontrollers and in particular the Arduino when used with a shield. Out of box implementation can be achieved when using two Arduinos and shields with only a jumper change to be made. For further implementation on multiple devices configuration and addressing settings have to change. After further research it was discovered that the XBee could not be adapted directly to the EM-406a GPS module without the use of a second Arduino microcontroller (Digi International Inc 2009).

To use the ZigBee specification and in particular the XBee module would require the redesign of the DGPS system. It would create a substantial cost increase due to the requirements of a second microcontroller and the required XBee modules and shields. This would not successfully achieve the objective of the project to produce a low-cost DGPS. No further testing was undertaken until an alternative solution was available. If the Murdoch University School of Engineering was to undertake further development into robotics requiring the use of mesh networks these devices would be ideal. The possibilities of creating number devices on one simple secure and reliable network could provide numerous advantages in future projects.

Figure 18 shows the comparison between the two models of XBee modules available with the major difference being output power. The Murdoch University school of Engineering currently owns two XBee-PRO devices and appropriate shields for Arduino.

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses

Figure 18: XBee Specification (Digi International Inc 2009)

3.6 APPCON Technologies APC200 & APC220

3.6.1 Overview

Further research was conducted into an alternate solution for wireless communications on the DGPS system after it was found the HopeRF and XBee modules would not be appropriate. Two small TTL, RS232 & RS484 based devices the APPCON APC200 & APC220 were procured as a suitable alternate. This device could only be purchased from China and one supplier in England. There are no apparent resellers or suppliers in Australia. The technical documentation supports the device as the best alternate solution for the DGPS. The manufacturer was contacted to check if the higher power model APPCON APC230 was available, however this was restricted to technology partners only. Using the APC230 would have presented a major advantage for its high power output (100mW) which would have allowed for greater distance of use and better error resilience. In practical terms, however the Rover robot would only be operated within tens of meters from the operator and kilometer ranges would not be required at this time of robotics development (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008).

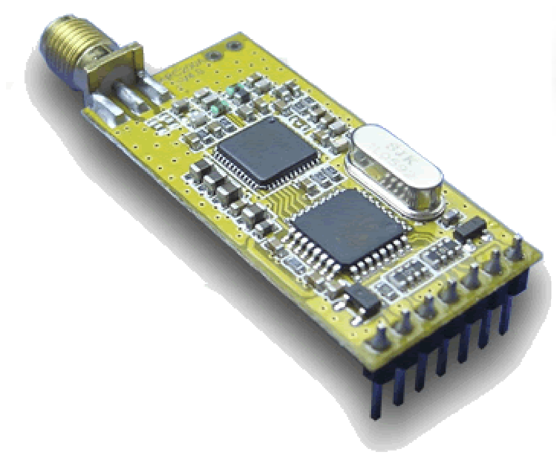


Figure 19: APPCON APC200

(SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

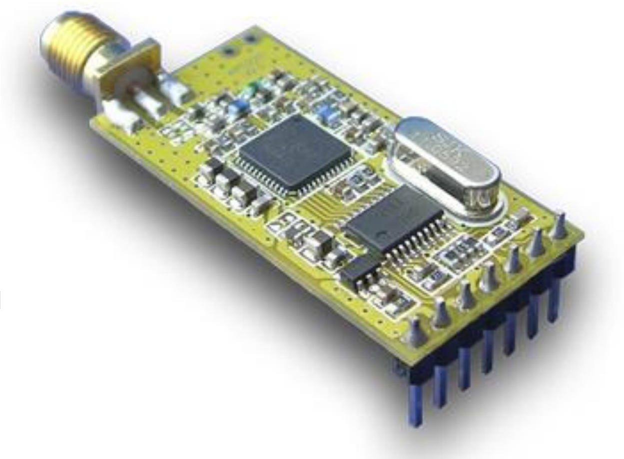


Figure 20: APPCON APC220

(SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

The APPCON technologies APC series APC200 & APC220 are highly integrated TTL level transceivers, providing reliable performance in strong interference environments. The transceiver module uses efficient forward error correction with interleaving encoding technology to overcome interference with high sensitivity. The transceiver module is specified by the manufacturer to an operational distance of 1800 meters at 2400bps. The APC220 allows for extremely flexible configuration as shown in Table 24 below (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008).

Table 24: APC200 & APC220 Configuration (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

Setting	Options	Default
UART Rate	1200, 2400, 4800, 9600, 19200, 38400, 57600	9600bps
Series Parity	Disable, Even Parity, Odd Parity	Disable
Frequency	418MHz – 470MHz	434MHz
Air Rate	2400, 4800, 9600, 19200bps	9600bps
RF Power	20mW	20mW
Comms	TTL or RS232/RS485 (APC200 only)	TTL
Online Configuration	Enabled through passing voltage through pin 7 (APC220 only)	Not enabled

The APC200 & APC220 contains a 256 byte data buffer which allows for large files to be transferred across the GFSK Modulation while maintaining a UART interface. The APC220 supports semi-duplex communications by point to point or point to multi-point configurations. Although there is very limited information on how to implement the point to multipoint, the manufacturer states that it can be designed using one host module while the receiver modules become client modules. Using this configuration, individual unique IDs can be set. The host coordinates the communications through sending the unique ID of the desired client with the data or command across the network of client transceivers. The clients that the ID does not match will discard the data. The manufacturer however does state that to avoid interference with other modules all client modules must be at 'transmitting mode' when the network is working. Unfortunately this configuration would not be useful to the DGPS setup or rover robot as bi-directional communications are required (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008).

Additional notes on issues with the transceivers were included in the manufacture's documentation attached in the Appendix DGPS005, DGPS006 & DGPS007. The major points include the following (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008):

1. Delay of wireless communications – While receiving data the transceiver will wait a tenth to a hundredth of a milli-second once the data is received, to confirm that no additional data follows. Another small delay will be incurred while transmitting between transceivers through wireless.
2. Data flux – The transceiver module has an onboard 256byte buffer zone. If the UART rate is higher than the air rate the buffer will overflow resulting in the loss of data. The manufacturer therefore recommends that the UART rate is to be 60 percent of the air rate therefore air rate of 9600bps but a UART rate of 4800bps should be sufficient.

3. Control of errors – The transceiver has strong anti-interference capabilities due to the efficient error checking and interleaving encoding technology however in severe situations strong electric interference can cause data loss. The reliability of the communications can be improved by using an Open Systems Interconnection model (OSI model) layer three protocol such as TPC/IP or the Point-to-Point Protocol. Using TCP/IP will allow for slip window and repeat transmitting functions to improve error redundancy and functionality.
4. Antenna – This is essential for reliable communications as each antenna is sized based on the frequency it is to transmit. If the working frequency is to be changed the antenna should be matched to the frequency.

3.6.1.1 Forward Error Correction

Communication systems that require accurate and reliable data transmission in environments of noise and interference can use different techniques of error-correction, however the most effective and economical is forward error correction coding. Forward error correction (FEC) also referred to as channel coding, is a form of digital signal processing that generates systematically generated data sequence prior to transmission. This known structured data sequence called the error-correcting code enables the receiver system to detect and correct limited errors caused during transmission. The increased transmission size requires a greater bandwidth to also send the error-correcting code (Wang, Sklar and Johnson 2002).

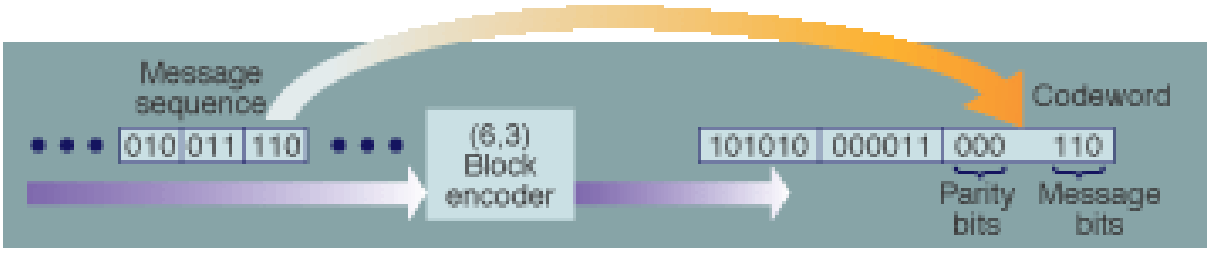


Figure 21: Forward Error Correction Diagram (Wang, Sklar and Johnson 2002)

3.6.1.2 Interleaving Encoding

Interleaved encoding is used to improve the performance of forward error correction to enhance the ability to deal with burst errors. This is achieved by shuffling source symbols across several code words which evenly distributes the errors. Latency increases are the adverse effect of interleaving techniques because the entire block must be received before the packet can be decoded. Technical information on what type of interleaving design the transceiver uses is not available.

Examples below show the use of interleaving (Techie 2011).

Interleaver designs

- Rectangle or uniform interleavers
- Convolutional interleavers
- Random interleavers
- S-random interleavers
- Quadratic permutation polynomial

Example

Transmission without interleaving:

```
Error-free message:                         aaaabbbbccccddddeeeeffffgggg
Transmission with a burst error:            aaaabbbbccc____deeeeffffgggg
```

The codeword dddd is altered in three bits, so either it cannot be decoded at all or it might be decoded incorrectly.

With interleaving:

```
Error-free code words:                      aaaabbbbccccddddeeeeffffgggg
Interleaved:                                abcdefgabcdefgabcdefgabcdefg
Transmission with a burst error:            abcdefgabcd____bcdefgabcdefg
Received code words after deinterleaving:   aa_abbbbccccdddde_eef_ffg_gg
```

In each of the codewords aaaa, eeee, ffff, gggg, only one bit is altered, so one-bit error-correcting-code will decode everything correctly.

Transmission without interleaving:

```
Original transmitted sentence:              ThisIsAnExampleOfInterleaving
Received sentence with a burst error:       ThisIs______pleOfInterleaving
```

The term "AnExample" ends up mostly unintelligible and difficult to correct.

With interleaving:

```
Transmitted sentence:                       ThisIsAnExampleOfInterleaving...
Error-free transmission:                    TIEpfeaghsxlIrv.iAaenli.snmOten.
Received sentence with a burst error:       TIEpfe______Irv.iAaenli.snmOten.
Received sentence after deinterleaving:     T_isI_AnE_amp_eOfInterle_vin_...
```

No word is completely lost and the missing letters can be recovered with minimal guesswork.

Figure 22: Interleaving Diagram (Wikipedia, Unknown 2011)

3.6.1.3 Point-to-Point Protocol

The point-to-point protocol (PPP) is an encapsulation protocol for transporting IP traffic over point-to-point links such as RS232, RS485 or RS422. The protocol uses the benefits of the TCP/IP protocol to include the assignment and management of an addressing system, bit-oriented encapsulation, error detection and multiplexing. For the PPP to establish communications over the APPCON APC220 transceiver the PPP would send an initial set of LCP frames to configure and test the link. A standard PPP Frame is shown in Figure 23 (Cisco Systems Press 1999).

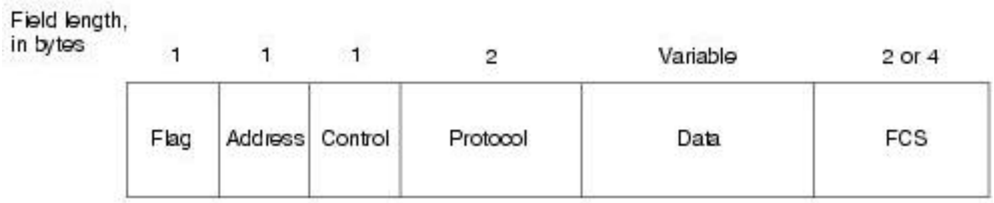


Figure 23: Point to Point Diagram (Cisco Systems Press 1999)

Table 25: PPP Packet Field Description (Cisco Systems Press 1999)

Field	Description
Flag	This is a single byte used to flag the start and end of the frame
Address	Contained in a binary sequence is a single byte for the standard broadcast address
Control	This is contained in a single byte which calls for transmission of user data in an unsequenced frame. Field is very similar to the Logical Link Control (LLC) type 1.
Protocol	This field is used to define the protocol used to encapsulate the data in the in the data field of the frame.
Data	This field contains the datagram for the protocol with a maximum length of 1,500bytes.
Frame Check Sequence	Used for error detection it can be either 16bits or 32 bit depending on PPP configuration

For the PPP to be implemented, a microcontroller is required on both sides of the link. This would add additional cost to the design of the DGPS system and the use of PPP on these microcontrollers would be resource intensive. There is little documentation or available libraries for PPP implementation on Arduino microcontrollers and its implementation could be a separate project. Implementing this system is very dependent on the microcontrollers used, for example for integration into the Rover Robot project would require the Motorola 68HC11 controller to have software written to support PPP.

3.6.2 Pin Configuration

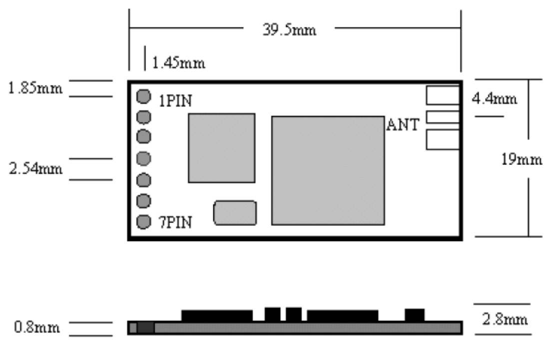


Figure 24: APC200 Pin Configuration (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

Table 26: APPCON APC200 Pin Configuration (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

Pin No.	Pin Name	Description
1	GND	Grounding of Power Supply
2	VCC	Power Supply DC 3.3V-5.5V
3	EN	Power Enable, >1.6V or empty, <0.5V
4	RXD	UART Input, TTL
5	TXD	UART Output, TTL
6	B/RX	RS485- or RS232 RX
7	A/TX	RS485+ or RS232 TX

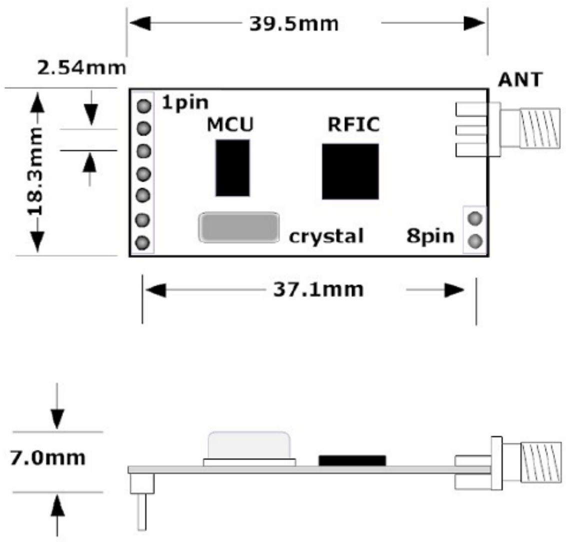


Figure 25: APC220 Pin Configuration

Table 27: APPCON APC220 Pin Configuration (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

Pin No.	Pin Name	Description
1	GND	Grounding of Power Supply
2	VCC	Power Supply DC 3.3V-5.5V
3	EN	Power Enable, >1.6V or empty, <0.5V
4	RXD	UART Input, TTL
5	TXD	UART Output, TTL
6	MUX	The pin is expanded for other functions
7	Set	Setting parameters, setting online supported.

APPCON Technologies do not have a mass distribution network and an almost non-existent retail supplier. Limited information was found within the English language to support these transceivers. Below is the contact information for APPCON Technologies.

APPCON Technologies
SHENZHEN APPCONTECHNOLOGIES CO., LTD.
RMB1-B2,5F,112 Building, JinDiindustry Zone FuTian
District ShenZhen (518048)
Tel: 86-755-83405295
Fax: 86-755-83405660
Email: appcon@126.com
http://www.appcon.com.cn

3.6.2 Configuration

3.6.2.1 APC200 Configuration

The APC200 can be configured using RF-Magic 4.2 software for windows based machines, shown in Figure 26.

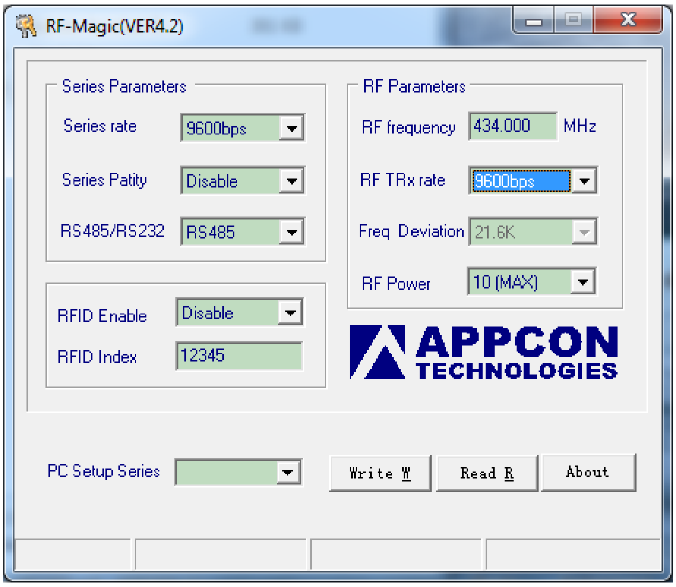


Figure 26: APC200 Configuration Software

3.6.2.2 APC220 Configuration

There are two methods of configuring the parameters of the APC220.

The first method is to use the RF-Magic software for windows based machines, shown in Figure 27.

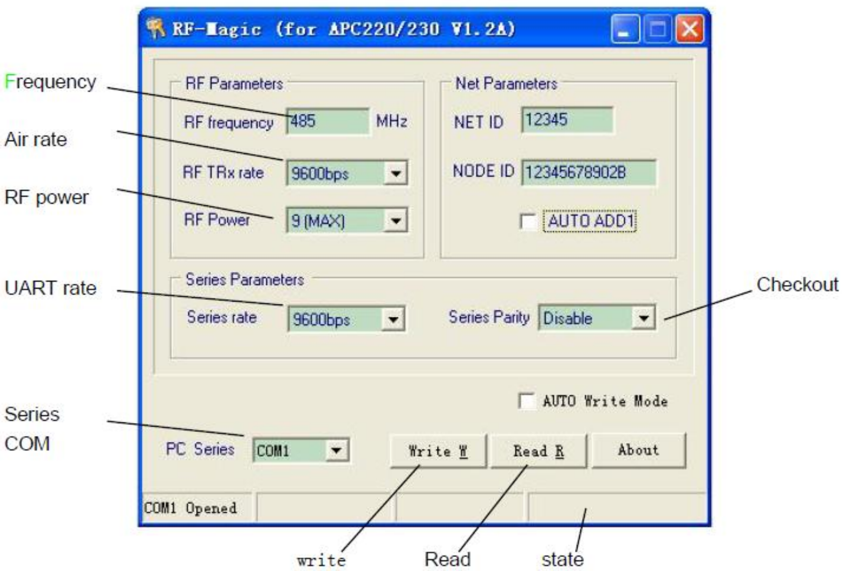


Figure 27: APC220 Configuration Software (SHENZHEN APPCON TECHNOLOGIES CO.LTD 2008)

The second method uses online configuration. For further information on this functionality, see Appendix DGPS006.

3.6.3 Evaluation Method

Evaluation of the APPCON APC200 modules was to be undertaken in the same method as the HopeRF Transceivers, however with an additional stress test for the advanced error detection and correction abilities of the module. The testing strategy used follows that of section 3.3.3.

This additional method of testing used various controlled distances and positions to test the accuracy of the GPS receivers positioning accuracy and the system differential accuracy. The method of this evaluation used the following:

This method of evaluation used a circle park oval to test the relative positioning

Table 28: APPCON 200 Evaluation Strategy

Step	Description and Action	Result	Comment
1	Position the base station in the estimated center of the circle	Base station setup	Location can be found by using Google Earth however it can only be an estimate
2	From the base station move to the outer circle. From this point walk at least one complete circle around the base station keeping an estimated constant radius.		Start point should be marked ensuring that one complete circumference is travelled.
3	Record output from the mobile station using software sketch 'DGPS Logging' – Logging should be completed over a complete lap of the circle which will provide enough data to demonstrate the relative positioning is correct.	This will provide evidence that the relative coordinates are correct and follow those of a unit circle	The unit circle is a good demonstration as to how this aspect of the system works. By providing a relative distance to the base station by x and y coordinates

3.6.4 Evaluation Results

The percentage noted against quality reflects the number of failed instances of either checksum or GPS sentences.

Table 29: APPCON APC200 Evaluation Results

No.	Environment	Distance	Baud	Bit	Data	Results
1	Indoors (open space)	0.5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data
2	Indoor (open space)	5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data
3	Indoor (open space)	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data
4	Indoor (between rooms – Brickwalls)	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data
5	Indoor (between rooms – Brickwalls)	15m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
6	Indoor (between rooms – Brickwalls)	~50m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
7	Indoors (between rooms – Brickwalls)	~50m	4800	1	GPS Data NMEA packets	Quality: Excellent – 0% Notes: TX data matched RX data & passed GPS checksum
8	Outdoors	0.5m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
9	Outdoors	10m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
10	Outdoors	20m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
11	Outdoors	~50m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
12	Outdoors	~150m	4800	1	Testing of Alphabetic and Numeric data	Quality: Excellent – 0% Notes: TX data matched RX data.
13	Outdoors	10m	4800	1	GPS Data	Quality: Excellent – 0% Notes: TX data matched RX data & passed GPS checksum

14	Outdoors	50m	4800	1	GPS Data	Quality: Excellent – 0% Notes: TX data matched RX data & passed GPS checksum
15	Outdoors	150m	4800	1	GPS Data	Quality: Excellent – 0% Notes: TX data matched RX data & passed GPS checksum
16	Outdoors	250m	4800	1	GPS Data	Quality: Very Good – 2% Notes: TX data matched RX data & passed GPS checksum. GPS packets would be dropped due to error correction rarely due to interference
17	Outdoors	400m	4800	1	GPS Data	Quality: Good – 5% Notes: TX data matched RX data & passed GPS checksum. GPS packets would be dropped due to error correction occasionally due to interference
18	Outdoors	800m	4800	1	GPS Data	Quality: Average – 30% Notes: TX data matched RX data & passed GPS checksum. GPS packets would be dropped due to error correction occasionally due to interference. Dependent on interference.

Note: Symbol '~' notes estimated distance

Figure 28 graphs distance over time from point origin and destination in Figure 29 for the testing completed in Table 29. The results shown in these figures demonstrated the effective communications distance of the transceivers. From this testing it can be concluded that the APC200 transceivers are suitable for implementation into the DGPS providing reliable and strong performance.

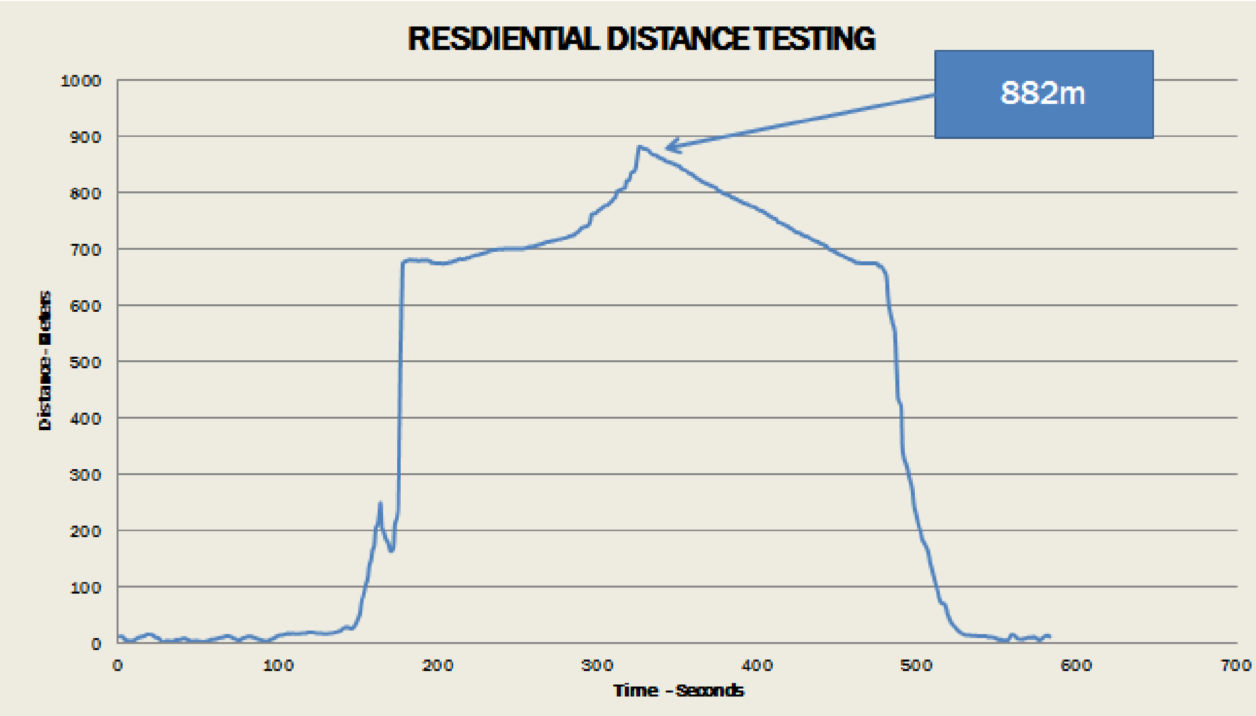


Figure 28: APC200 Distance Testing Over Time

Figure 29: APC200 Distance Testing Maximum Distance Positions

3.6.5 Evaluation Review

The results from the APPCON APC200 substantially exceeded those of the system requirements. Additional error correction capabilities present in the APC200, APC220 and APC230 transceivers enable the error free transmissions in high interference areas. Shown in Figure 28 and Figure 29 are the results of distance testing in a residential environment, achieving 882 meters transmission distance. Residential areas commonly use devices based on the 433MHz frequency for alarm systems, control and smart systems as it is partly unregulated, which can cause substantial interference.

As a secondary test, all GPS data tested were compared against the NMEA 0183 checksum of each sentence transmitted. Results for this secondary test indicated that using the APC200 transceivers enable correct operation of the communications component of the system and has therefore been integrated into the prototype.

The evaluation of the transceivers distinctly shows the APC200 are the best of the transceivers evaluated. A functionality comparison of the APC200 to the other transceivers evaluated is shown in Table 30.

Table 30: Comparison of Transceivers

HopeRF HM-TR RS232 Version	HopeRF HM-TR TTL Version	APPCON APC200	APPCON APC220	XBee
RS232	TTL	232/485/ TTL	TTL	SPI
433mhz	433mhz	431-478Mhz	431-478Mhz	2.4Ghz
3.2mW	3.2mW	20mW	20mW	10mW
5 Volts	5 Volts	5 Volts	5 Volts	5 Volts
300m	300m	1000m@1200	1000m@1200	60m
Murdoch Uni	Futurlec Thailand	BestElectronics China	DROBOTICS UK	Murdoch Uni

3.7 Multiplexing with the Rover Robot

The concept of being able to multiplex communications from the Rover Robot using the DGPS communications is a tedious and time intensive task. It was discussed, if time permits, to research and design a possible method of integration of the communications for the Rover Robot. The following is a summary of how this could best be achieved taking into account the incompatibilities of the 68HC11 with the Arduino.

Firstly, to achieve strong and reliable communications, the APPCON APC220 transceiver would be required for its extensive integrated error detection and correction technologies. The modules do not support the advanced features of the ZigBee specification which includes the ability for the transceiver to first check if any other device is communicating before transmitting, avoiding cross talk and ultimately corrupt data. These modules allow for up to 100 Channels which would present an advantage over other transparent transceivers by eliminating interference from multiple transmissions from communicating at the same time. The system would use two sets of transceivers differentiated by transmission channels for communications between the GPS Base station to the Arduino Mega and the Operator communications to the Arduino Mega. It was decided that the operation of the Rover Robot could be achieved from a different location to the GPS base station which requires access to GPS signals. It is recommended that the GPS base station is placed outside where the GNSS signals received are uninterrupted by buildings and obstacles.

The possibility of video and audio integration could influence the type of communications used between the operator. This would require a high bandwidth connection to support the continuous Data Stream. The high bandwidth frequencies of 2.4GHz would be ideal as a transmission medium however the high power requirements for compatible devices could restrict the Rovers abilities significantly. Using the ZigBee protocol would provide no apparent advantage as the maximum transmission rate is only 250Kbps. IP based video solutions generally transmit anywhere between 12.5fps to 30fps requiring bandwidth above 1Mbps. Currently no set requirements for the Rover Robot communications have been prescribed, however before a recommendation can be made, operational functions must be analyzed.

Figure 30: Rover Robot

Another avenue of communications which has been not been reviewed in this project, is the use of preexisting GSM, 3G or 4G communications networks. A host of operators are available, with the cost of data decreasing rapidly making it economically viable. The possibility of having a metro wide communications link of 3.6mbps and upwards could be very desirable. Latency delays should be expected, however these can be extremely low on a 4G network depending on signal strength and network load. It is recommended if further development of the Rover Robot is to be undertaken, Murdoch university Engineering department should investigate the use of this type of communications systems in robotics.

CHAPTER 4: SOFTWARE

This section of the project brought together all the major components of the system including the hardware, communications and the data received from GNSS. The software follows a very distinct methodology which starts with the processing, called parsing, of the data received from the receivers. Parsing translates the data into a usable format from which position and attributes calculations can take place. Finally, the calculations are sent to the operator and displayed on the embedded LCD screen.

Before the software was written, research was conducted to find and evaluate any preexisting software libraries that could contribute to the functionality of the system. The use of open source software libraries would reduce the complexity of the system design and also provide a significant increase in system functionality and error reduction. The open-source software libraries used in this project are reflected in Table 31. This chapter reviews the software design for the system and includes a review and breakdown of including open source libraries.

Table 31: Custom Software Libraries Used in Software

Custom Software Library	Author	Version
TinyGPS	Mikal Hart	10
FloatToString	Tim Hirzel	March 2008
LCD4Bit_mod	neillzero	0.1

Figure 31 shows a simple breakdown of the software design and the flow of data and information through the system to achieve the required output. It is expected that the software completes one cycle every second to keep in line with the received GPS NMEA packets.

For the functionality of the libraries to be used, the system software had to be constructed with subroutines. The first step of the software written in this thesis was to define the hardware configuration and the operation of the system. The next step was to integrate the libraries and then the output could then be used to write the relative positioning calculations and user interfaces. During the construction of the software thorough testing was conducted ensure the output was correct. Testing included the comparison of distances, angles and known location positioning coved in section 4.4.3 of this thesis.

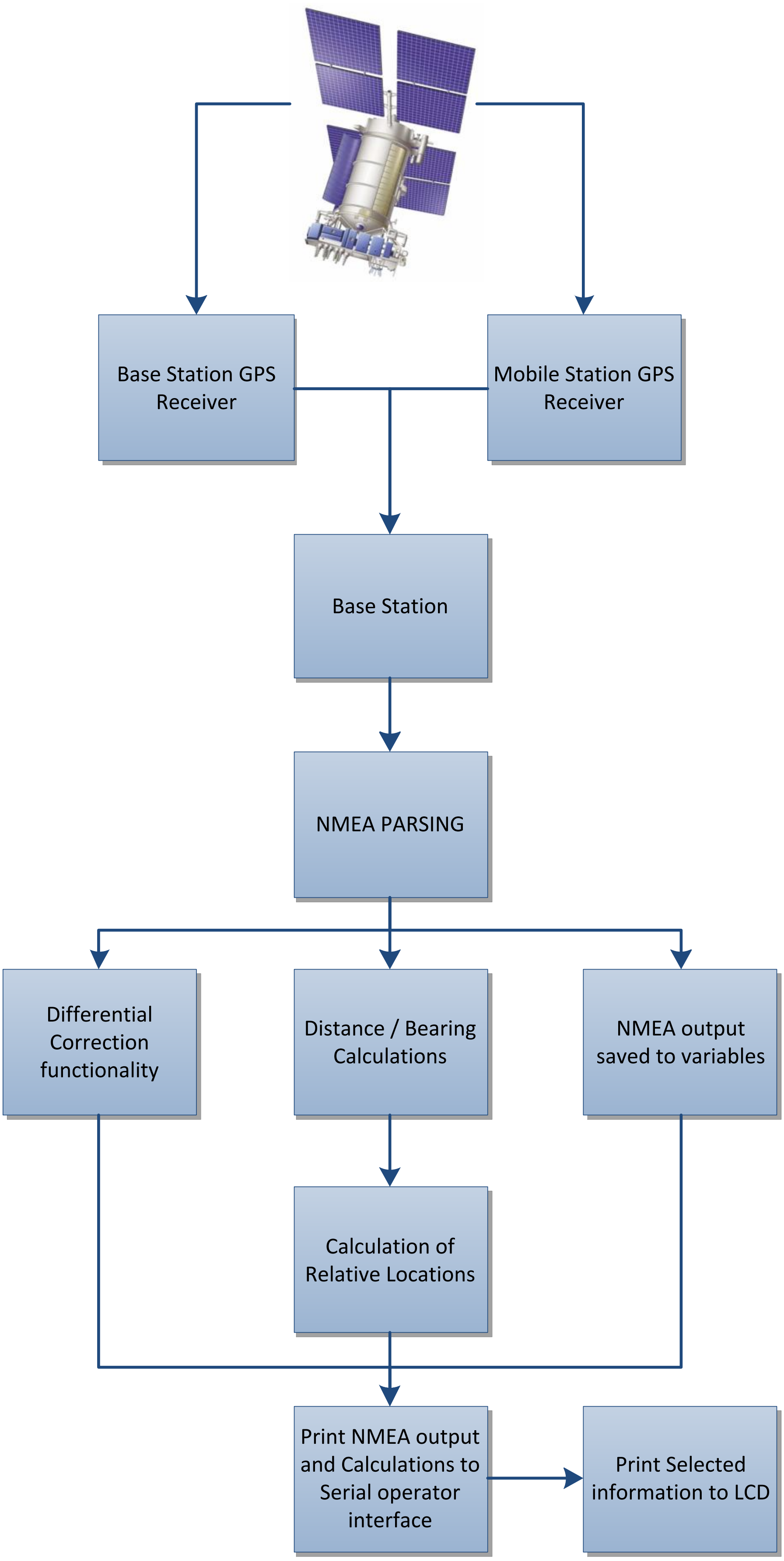


Figure 31: Software State flow

4.1 Parsing NMEA

4.1.1 Concepts

This thesis makes use of an open source library called TinyGPS to parse the received NMEA GPS packets. This library is capable of parsing the collected data into usable formats which include the position, date, time, altitude, speed and course. The TinyGPS library is written and structured such that it has low resource consumption and avoids any floating point dependencies. Created by Mikal Hart the library has been developed for Arduino users to implement into projects requiring various position and speed attributes. (Hart, Arduiniana 2011)

The TinyGPS library is designed for singular GPS use however this project has adapted the code such that two GPS modules can be used. There were various methods researched and tested as to how to restructure the library so that it could be used as part of a DGPS system. The major concern was that the sharing of variables between the modules could corrupt the read data. The library was originally implemented into the system using the 'NewSoftSerial' library which was previously discussed in this thesis. Due to the restraints of CPU time using software serials to receive and process the data, it was not possible and resulted in continued failure. The Mega included four hardware UARTS which would be used between the GPS modules and host operator to receive simultaneous data streams. The received data would then be processed using two individual calls of the TinyGPS library using a different data file for each module. This meant that the issues of corrupt data through sharing variables would be resolved.

4.1.2 Design

Changing the library such that two modules could be used was a complex and tedious task. The library was makes use of an advanced understanding of the wiring command set and programming techniques. Research was performed on understanding how other Arduino users integrated the library and in particular paying attention to the methods of implementation and issues encountered. Focusing on the problems encountered provided an understanding of why it was incorrect and how to resolve the issue.

The first approach to implementing two modules was to rewrite the library so that each GPS module had its own set of variables. This meant the library would be constructed as 'variable.gps.a' or 'variable.gps.b.' Testing was undertaken with this newly altered library. The newly altered 'TinyGPS' library had close to doubled in size using the limited memory available to the Arduino Duemilanove. The major issues in testing were that the software serial would not capture the entire NMEA as the buffer would overflow resulting in incomplete data. This meant that the newly altered Tiny GPS library would not be able to process the data.

After the testing of the newly altered TinyGPS library, the design was taken back to the initial stages where further research was conducted. It was decided that an Arduino Mega would be used for the reasons discussed previously. Using the Mega meant that more processing time would be available and multiple instances of the TinyGPS library could be used. For this approach the TinyGPS library was used unaltered from the original library. The initial setup in the Arduino sketch was structured in such a way that multiple TinyGPS instances were called when data was available from both GPS modules via hardware UARTs.

Once implemented this method produced parsed NMEA data into the usable format of position (degrees), date, time, altitude and speed (knots) from both modules. Table 32 shows the results from two instances of TinyGPS and the result on the system.

Table 32: NMEA Packet Parsing Method Study

Test	Method	Results	Comment
Single Instance	Implementing the TinyGPS library using standard Method	Normal Output	
Single instance Checksum	Using the TinyGPS library inbuilt Checksum function from the NMEA packet	Resulted in ‘0’ indicating that there were no checksum errors	Stable system
Multiple Instances	Implemented TinyGPS library using dual instances for both EM-406a Modules	Normal Ouput	
Multiple Instances Checksum	Using the TinyGPS library inbuilt Checksum function from the NMEA packets	Resulted in ‘0’ indicating that there were no checksum errors	Stable system

There are three major software components to the DGPS software package.

The two GPS modules are connected via serial communications to the Arduino Mega however these modules do not simply pass the coordinates of its position. Instead the GPS modules use the standard National Marine Electronics Association (NMEA) structure which consists of different packets of data. The EM-406a outputs the NMEA 0183 structure with GGA, GSA, GSV, RMC (VTG, GLL optional) packet types (GlobalSat Technology Corporation 2007). These packets are transmitted by the EM-406a at 4800bps which allows the microprocessor to parse the packets into more useful formats. The NMEA standard was originally released in 1983 however since then it has undergone various version changes including 1.5, 2.0 through to 2.3 and 3.01. NMEA 0183, also known as version 2.0, is the most common NMEA structure to be used with navigation equipment in the EM-406a (Bennett and Mehaffey n.d.).

The concept of the NMEA packet structure is to send a single line which is self-contained and independent from other transmitted sentences. Using the standard structure shown below in Figure 32, allows a universal platform for which GPS systems can obtain and read data. There is the ability for companies of proprietary systems to define their own specific packet structure. The beginning of all NMEA sentences contains the character '$' which specifies that it contains an NMEA sentence. The sentence then ends with a Checksum of the sentence transmitted including all characters except the '$' and checksum.

```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Where:
     GGA          Global Positioning System Fix Data
     123519       Fix taken at 12:35:19 UTC
     4807.038,N   Latitude 48 deg 07.038' N
     01131.000,E  Longitude 11 deg 31.000' E
     1            Fix quality: 0 = invalid
                               1 = GPS fix (SPS)
                               2 = DGPS fix
                               3 = PPS fix
                               4 = Real Time Kinematic
                               5 = Float RTK
                               6 = estimated (dead reckoning) (2.3 feature)
                               7 = Manual input mode
                               8 = Simulation mode
     08           Number of satellites being tracked
     0.9          Horizontal dilution of position
     545.4,M      Altitude, Meters, above mean sea level
     46.9,M       Height of geoid (mean sea level) above WGS84 ellipsoid
    (empty field) time in seconds since last DGPS update
    (empty field) DGPS station ID number
     *47          the checksum data, always begins with *
```

Figure 32: Standard NMEA 0183 GPGGA Sentence (Bennett and Mehaffey n.d.)

To use the NMEA packet data the microcontroller needs to parse the packet into a useable format. The Arduino community is active in compiling and sharing libraries that can be used to compliment programs. In particular a library developed by Mikal Hart called TinyGPS is the most commonly used Arduino library to parse NMEA 0183 packets. Developing a library such as TinyGPS is time intensive and large amounts of testing and error detection must be completed. To advance the Murdoch University DGPS system with the best software, the TinyGPS library was chosen to be integrated as the GPS NMEA 0183 packet parser. TinyGPS is an active project with thorough public testing and error correction; the library is currently at version 10.

The TinyGPS library is not contained in the standard Arduino library distribution, however it can be very easily imported. The TinyGPS library is a construction of two files, TinyGPS.h and TinyGPS.cpp, which contain functions and commands to parse the data. The Arduino sketch uses standard programming techniques to call the library and its functions. The standard TinyGPS library makes use of the ‘NewSoftSerial’ library, which is intended to replace the standard Arduino software serial library.

4.3 Differential Correction

4.3.1 Concepts

Using the differential concepts discussed in Chapter 1, DGPS correction can significantly increase the accuracy of the location data. It is assumed that GNSS signals received within milliseconds on two separate receivers within a specific area should have the same amount of error caused by atmospheric distortion. Using differential correction should greatly reduce the effects of the atmosphere and correct the received positioning data. This is achieved by using the known location of the base station and calculating the difference between the received base station coordinates and the known coordinates. This difference is predominately the error caused by GNSS signal diffraction in the atmosphere and can be applied to the mobile station. The difference corrected mobile station coordinates can then be used to obtain further positioning data.

Table 33: GPS & DGPS Atmosphere Comparison

Error Source	Standalone (m)	DGPS(m)
Ephemeris	5-20	0-1
Ionosphere	15-20	2-3
Troposphere	3-4	1

4.3.2 Design

When designing the system, considerations had to be made, including the mobility of the base station and the time required to 'warm' the system up. It was decided that an averaging formula would be used over a period of time to obtain the known location. This would not be as accurate as placing the base station on a known location but would allow flexibility in the operation area of the system while still providing accurate readings. Therefore time dictates how accurate the base station 'known' location is to become.

4.3.3 Evaluation Strategy

The strategy for testing the base station location averaging formula is completed over a defined time period. Looking at the averaged coordinates over selected time intervals the change in co-ordinates at the fixed location can be compared.

More intensive testing could include setting the system up at a known location and recording the averaging over the time intervals. These averaged coordinates could be compared to the true known location. Due to time constraints of the project, Google Earth was used to locate known a location, for testing of the differential correction.

4.3.4 Evaluation Results

Shown in Table 34 are the recorded known location and the base station averaged difference. This difference would be applied to the mobile station to correct the position against the known location of the base station and the received base station GPS positioning data. The averaged location is calculating the distances over the specific time period.

Table 34: Averaging ‘Known’ Location

Time Period	Known Location	Difference between Averaged Location
10 minutes	(-31.0555, 115.8294)	(0.000019, -0.000031)
15 minutes	(-31.0555, 115.8294)	(0.000027, -0.000031)
30 minutes	(-31.0555, 115.8294)	(0.000019, -0.000023)
45 minutes	(-31.0555, 115.8294)	(0.000008, -0.000031)
1 hour	(-31.0555, 115.8294)	(0.000019, -0.000046)
2 hours	(-31.0555, 115.8294)	(0.000017, -0.000036)
3 hours	(-31.0555, 115.8294)	(0.000008, -0.000031)
4 hours	(-31.0555, 115.8294)	(0.000008, -0.000031)

4.3.5 Evaluation Review

The results recorded during evaluation indicated that the software was functioning properly as differences were recorded between the ‘known location’ and the received base station data. This data could then be applied to the mobile station to correct the location with the same error distance of the mobile station. From the review data Table 34, the limited resolution of the GPS receivers will restrict the accuracy of the system. Another issue that could affect the accuracy of the calculations, is the rounding issues caused by floating point numbers which have a precision of 6-7 decimal digits (Arduino Reference 2011).

4.4 Differential positioning

4.4.1 Overview

Relative positioning techniques are vital for autonomous robotic movements, to ensure that the system can precisely track its location based on a base station reference. This system can only be accurately implemented with the use of a differential system with a minimum of one base station location and one GPS receiver for mobile positioning. Using an additional base station reference points could vastly improve the accuracy through triangulation methods to track relative positioning. This mobile position can then be calculated from the known location into x,y,z coordinates.

4.4.2 Design

The differential positioning of the system must separate into three major components from which calculations for each can be undertaken. Using the unit circuit concept can simplify the calculations that need to be performed into what is required to find points on a unit circle. It assumed for this concept to work the z-plane of the GPS is ignored due to the inaccuracies of the altitude data from the GPS.

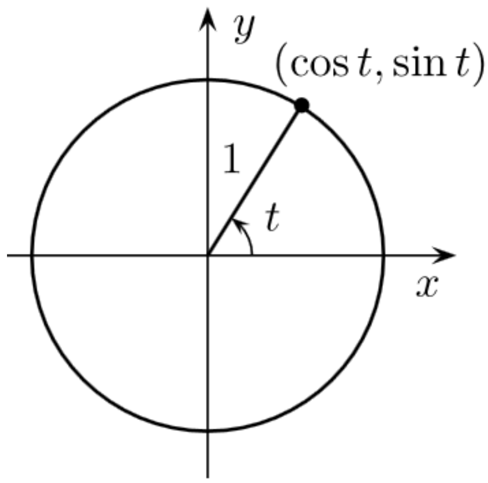


Figure 33: Unit Circle (Wikipedia 2011)

4.4.2.1 Distance Calculation

The first step in calculating the relative position of the mobile station to the base station is to undertake calculations of distance between the two nodes. The haversine formula is used to find the distance between two points on a sphere from the longitude and latitude. It is therefore the relation between the latitudes of the origin and destination nodes and separation of the latitudes and the longitudes. Used in previous times before modern computers, it provided navigators with a method of calculating distances without using long-multiplication and long-division (Longitude Store n.d.).

Common modern navigation formulas such as the 'spherical law of cosines' are inaccurate when using 8 bit microcontrollers with 4 byte floats as the precision of these size floats cause too much error. (aliasmrjones 2009) Therefore using the haversine formula, although slightly more complex, will provide more accuracy using the 4 byte floats at small distances. The haversine formula however is only an approximation, as the Earth is not a perfect sphere and therefore an error of 0.1% is assumed with the earth radius of 6367.45km. (Longitude Store n.d.) The haversine formula is shown in Equation 2 and Figure 34.

$$
\begin{gathered}
a = \sin^2(\delta\mathrm{lat}/2) + \cos(\mathrm{lat}_1)\cos(\mathrm{lat}_2)\sin^2(\delta\mathrm{lon}/2) \\
c = 2 \times \mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)}) \\
d = R \times c
\end{gathered}
$$

Equation 2: Haversine Formula (Python 2011)

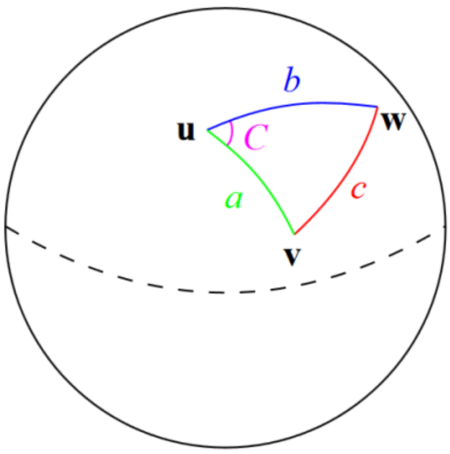


Figure 34: Haversine Formula (Veness 2010)

To program the formula into the wiring language, the calculation has to be broken into a number of steps. These steps are shown in Equation 3 also noting the angles must be in radians for the formula to function correctly.

Equation 3: Haversine Formula Step Calculation (Veness 2010)

R = earth's radius (mean radius = 6,371km)
$\Delta lat = lat_2 - lat_1$
$\Delta long = long_2 - long_1$
$a = \sin^2(\Delta lat/2) + \cos(lat_1).\cos(lat_2).\sin^2(\Delta long/2)$
$c = 2.atan2(\sqrt{a}, \sqrt{(1-a)})$
$d = R.c$

(Note angles need to be in radians)

Equation 4 shows the haversine formula implemented in a wiring subroutine and is best understood by a person competent in the software languages C++ or wiring. The use of variables starting with 'd' refer to degree values and 'dtor' is a call for a calculation from degrees to radians.

Equation 4: Distance Calculation - Two Co-Ordinates Software (Arduino Forum: arbarnhart 2011)

```
float CalcDistance(float lat1, float lon1, float lat2, float lon2)
{
float dlon, dlat, a, c;
float dist = 0.000;
dlon = dtor(lon2 - lon1);
dlat = dtor(lat2 - lat1);
a = pow(sin(dlat/2),2) + cos(dtor(lat1)) * cos(dtor(lat2)) * pow(sin(dlon/2),2);
c = 2 * atan2(sqrt(a), sqrt(1-a));

dist = 6378140 * c;
return( (float) dist + 0.5);
}
```

The complete coding implementation is included in the appendices where step calculations have been annotated for referencing and understanding.

4.4.2.2 Bearing Calculation

The second step in finding the relative distance is the calculation of the bearing between the two points. In the case of the DGPS it is assumed that the base station is the origin and the mobile station is a calculated distance and bearing from the origin. To find this bearing, Equation 5 is used and is the bearing relationship between the origin and destination points.

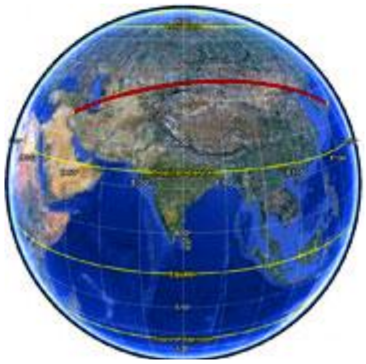


Figure 35: Bearing Calculation Diagram (Veness 2010)

Equation 5: Bearing Formula (Veness 2010)

$$\theta=\operatorname{atan2}(\sin(\Delta long).\cos(lat2),\ \cos(lat1).\sin(lat2) - \sin(lat1).\cos(lat2).\cos(\Delta long))$$

The code from which the relative angle for the system is calculated is included below in Equation 6.

Equation 6: Bearing Calculation Software Code (Arduino Forum: arbarnhart 2011)

```
float CalcBearing(double lat1, double lon1, double lat2, double lon2)
{
lat1 = dtor(lat1);
lon1 = dtor(lon1);
lat2 = dtor(lat2);
lon2 = dtor(lon2);

//determine angle
double bearing = atan2(sin(lon2-lon1)*cos(lat2), (cos(lat1)*sin(lat2))-
(sin(lat1)*cos(lat2)*cos(lon2-lon1)));
double bearing_mobile;
//convert to degrees
bearing = rtod(bearing);
//use mod to turn -90 = 270
bearing = fmod((bearing + 360.000), 360.000);
return (float) bearing;
}
```

4.4.2.3 Relative Positioning Calculation

The final step which utilises the values calculated in 4.4.2.1 and 4.4.2.2 finds the (x,y) coordinates, in meters, from the base station. In terms of autonomous control of robotics, this is a crucial and significant function which can enable the system to accurately position itself in terms of the base station or point of origin. The principle technique in finding these coordinates is based on the unit circle concept and trigonometric functions.

The calculation is included below in Table 35, which differentiates the y coordinate calculation to the x coordinate calculation. From viewing these, a series of 'if' statements are used to ensure no bearing exceeds 90 degrees. The distance and bearing from previous calculations are used in either the y cos relative calculation or the x sin relative calculation.

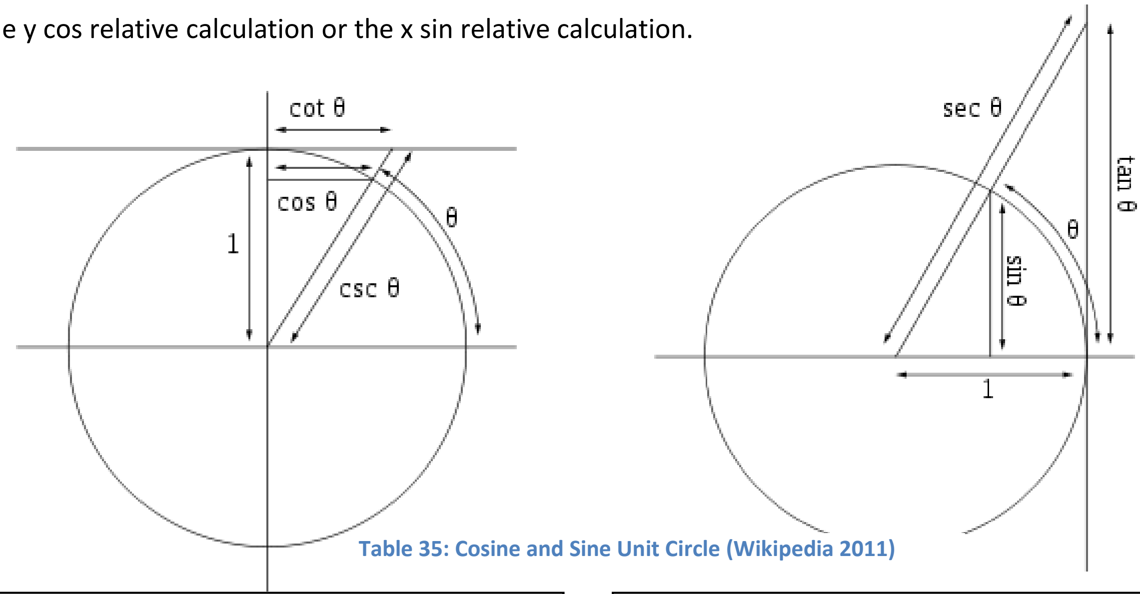


Table 35: Cosine and Sine Unit Circle (Wikipedia 2011)

```
float Y_Rel_Location(float yDist, float yBear)
{
  float  y_rel_coordinate;
  float  y_angle;
  float  y_angle_rad;

  if (90 < yBear){
    y_angle = yBear;
  }
  if (90 < yBear < 180){
    y_angle = yBear - 90;
  }
  if (180 < yBear < 270){
    y_angle = yBear - 180;
  }
  if (270 < yBear < 360){
    y_angle = yBear - 270;
  }

y_angle_rad = dtor(yBear);
return yDist * sin(y_angle_rad);
}
```

```
float X_Rel_Location(float xDist, float xBear)
{
  float  x_rel_coordinate;
  float  x_angle;
  float  x_angle_rad;

  if (90 < xBear){
    x_angle = xBear;
  }
  if (90 < xBear < 180){
    x_angle = xBear - 90;
  }
  if (180 < xBear < 270){
    x_angle = xBear - 180;
  }
  if (270 < xBear < 360){
    x_angle = xBear - 270;
  }

x_angle_rad = dtor(xBear);
return xDist * cos(x_angle_rad);
}
```

4.4.3 Evaluation Method

Evaluation of the system at this stage was to test the capacity and accuracy through challenging tests, in differing environments. This evaluation included distance accuracy testing and unit circle testing. Using these evaluation methods the abilities and accuracy of the system have been established.

4.4.3.1 Distance Precision Testing

To achieve an accurate system, comparable to an industrial grade equivalent, thorough testing and analysis was conducted. This method of evaluation used various controlled distances and positions (Google Earth was used for larger distances) to test the accuracy of the GPS receiver's positioning accuracy and the system differential accuracy. The accuracy of Google Earth could not be found, however it was assumed that it would be equivalent to an industrial grade DPGS system. The method undertaken for precision testing is shown in Table 36.

Table 36: Distance Evaluation Strategy

Step	Description and Action	Result	Comment
1	Two known positions are used and distance between is found. This is achieved through the use of Google Earth or if in close proximity measured.	Accurate known position to test the base station and mobile station.	For Google earthed to be used the known locations must be established before 2008. (Data age)
2	Base station & mobile station are place on the two known locations creating a known distance between.	Accurate known distance from receivers is obtained	Google earth is used to measure the distance between the two locations.
3	Record output from the mobile station using software sketch 'DGPS Logging' – Logging should be completed over a time period which will provide enough data to accurate average the distance	DGPS Distance is recorded	This should be completed over varying distances. The greater the distance less percentage error.

The results for this test are shown in The known distance is known between two points from testing is to occur. The base station and mobile station are separated at these locations and then DGPS output distance data is recorded. This recorded data is taken over a period of time to find the average distance produced by the system. The Standard deviation is also recorded to demonstrate the precision of the prototype system.

Table 38 and compare the known distance to the average distance and standard deviation.

4.4.3.2 Unit Circle Testing

To test the relative positioning code, the unit circle test was developed to evaluate the sign convention of positions in each quadrant of the unit circle. Refer to figure 33 where the unit quadrants are depicted. The method undertaken for relative positioning through the unit circle test is shown in Table 37.

Table 37: Unit Circle Evaluation Strategy

Step	Description and Action	Result	Comment
1	Position the base station in the estimate center of the circle	Base station setup	Location can be found by using Google Earth however it can be an estimate
2	From the base station move to out to the surrounding circle. From this point walk at least one complete circle around the base station keeping an estimated constant radius. The WPS testing oval has a perimeter fence and wall. During testing maintain a 1 meter distance from the perimeter fence and wall.	Positions will have entered every quadrant of the unit circle	Start point should be marked such that one complete circumference can be travelled.
3	Record output from the mobile station using software sketch 'DGPS Logging' – Logging should be completed over a complete lap of the circle which will provide enough data to demonstrate the relative positioning is correct.	This will provide evidence that the relative coordinates are correct and follow those of a unit circle	The unit circle is a good demonstration as to how this aspect of the system works. By providing a relative distance to the base station by x and y coordinates

4.4.4 Evaluation Results

This section will show the results achieved and review particular results which are important to the evaluation of the system.

4.4.4.1 Distance Precision tests

The known distance is known between two points from testing is to occur. The base station and mobile station are separated at these locations and then DGPS output distance data is recorded. This recorded data is taken over a period of time to find the average distance produced by the system. The Standard deviation is also recorded to demonstrate the precision of the prototype system.

Table 38: Distance Testing Results

Area	Inside/Outside	Weather	Known Distance	Av. Distance	St. Deviation
Residential	Outside	Fine	1m	2.50588	1.2638
Residential	Outside	Fine	5m	5.51086	2.5242
Residential	Inside	Fine	1m	5.12671	2.4423
Residential	Inside	Fine	5m	7.25189	4.6877
Oval	Outside	Fine	1m	2.52913	0.5927
Oval	Outside	Fine	20m	19.7845	0.7626
Oval	Outside	Cloudy	20m	19.4665	1.0236
Oval	Outside	Fine	100m	101.2747	1.2465

Included below are sample photos of the setup in which the evaluation of the prototype was undertaken.

Figure 36: Testing Distance of 1 meter Open Space Setup

Figure 37: Testing Distance 5 meters Restricted / Residential Setup

Figure 38: Distance Testing 20m Open Space (Google Earth Tools)

Figure 39: Distance Testing 20 meter Open Space Setup

4.4.4.2 Unit Circle Clockwise Testing Results

Figure 40: Unit Circle Testing Clockwise (Overlay on Google Earth Image)

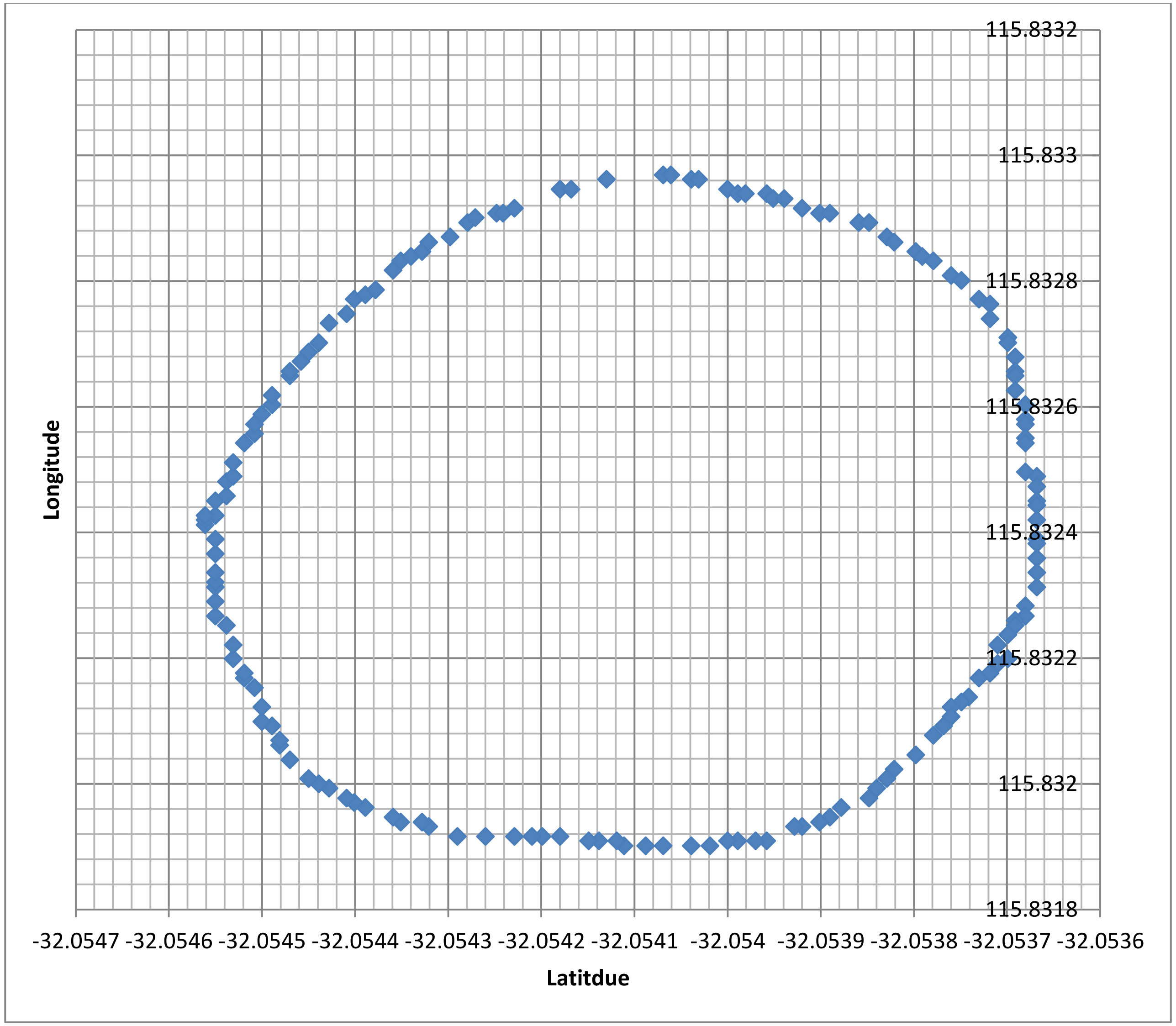


Figure 41: Unit Circle Testing Clockwise (Co-ordinates Graphed)

4.4.4.3 Unit Circle Anti-Clockwise Testing Results

Figure 42: Unit Circle Testing Anti-Clockwise (Overlay on Google Earth Image)

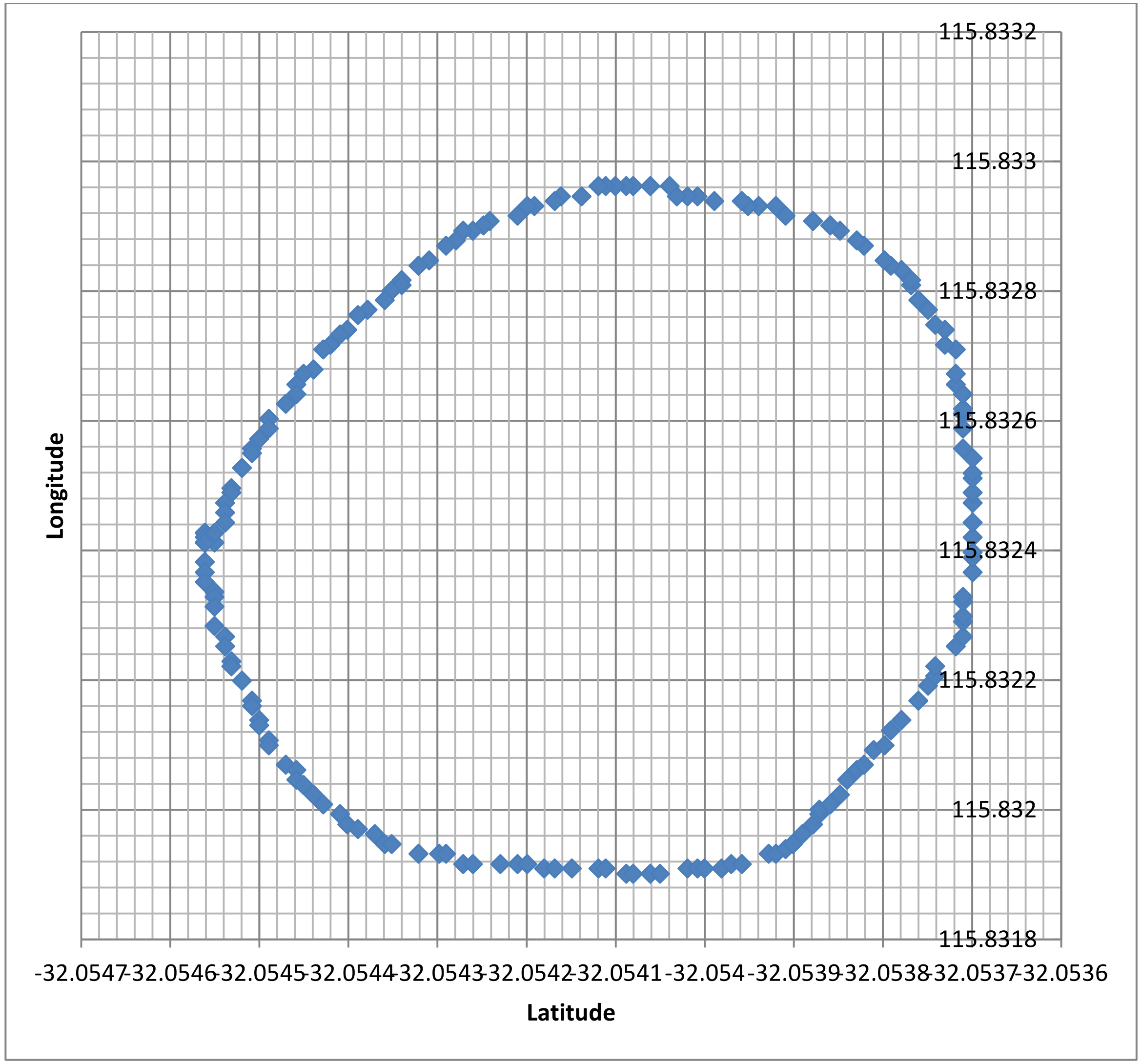


Figure 43: Unit Circle Testing Anti-Clockwise (Co-ordinates Graphed)

4.4.5 Evaluation Review

The results achieved in the differential positioning evaluation, concluded that a number of varying factors influence accuracy. To demonstrate the impact of the environment on accuracy, the prototype was operated in both restricted and open space spaces. The difference between these locations was the ability for the receivers to connect to GNSS satellites. GPS is more accurate when it can connect to a higher number of satellites and to function correctly, a minimum of four satellites must be present. It should be noted that from cited text, using a residential environment restricts the potential to connect to satellites and increases multipath error. Multipath error is when GPS signals reflect off buildings and obstacles and cause increased signal distortion. (NATO - RTO 2008)

Distance testing results indicated that open environment produced higher accuracy (0.216m) and a lower standard deviation (0.763m) compared to testing for the restricted environment, accuracy (0.511m) and standard deviation (2.524m). Further restricted testing was conducted inside a building which results in reduced accuracy (2.252m) and an increased standard deviation (4.688m). These indoor testing results supported the conclusion that the system should only be operated outside. Further results can be viewed in The known distance is known between two points from testing is to occur. The base station and mobile station are separated at these locations and then DGPS output distance data is recorded. This recorded data is taken over a period of time to find the average distance produced by the system. The Standard deviation is also recorded to demonstrate the precision of the prototype system.

Table 38.

Using the unit circle concept, the relative positioning coordinates were recorded and graphed. This graph was then overlaid on the Google Earth image of the oval to confirm the result followed the shape of the oval. This method of evaluation concluded the successful functional capabilities of relative positioning. Both clockwise Figure 40 and anti-clockwise Figure 42 testing was completed to ensure the accuracy of the results. These were compared to the graph Coordinates of each in Figure 41 and Figure 43. Both anti-clockwise and clockwise testing was completed to ensure direction of travel did not impact the results.

The use of an offset should be investigated in future development of the DGPS to overcome any rounding issues apparent with floating point numbers. This offset could reduce the error consistently on the results produced and improve accuracy of the DGPS.

CHAPTER 5: FUTURE RECOMMENDATIONS

5.1 Multiple Base Stations

An industrial grade differential position system uses multiple base stations to surround the operational area of the mobile station to create a highly accurate network of reference points. This concept could be applied to the DGPS design in this project where multiple GPS base stations could be constructed and wireless communications established on different channels using the APPCON APC220s. In a close proximity environment, XBee modules could be used in a mesh network configuration which could be beneficial for the relay of data to the operator. Expansion of the current software would need to include the additional use of the incoming GPS communications using the four UARTS available on the Arudino Mega. The new sketch would need to include a replication of the code for the existing base station differential calculations and output reference point coordinates for the additional base stations.

This progression of the differential positioning system would be a significant advantage for the university if further development of robotic systems and autonomous control was to be undertaken. Additional enhancements to the system could be made by procurement of GPS receiver modules with external antennas to increase signal sensitivity.

5.2 Kalman Filter

Kalman filters are used to improve the performance of GPS systems through estimation and reduction of zero-mean white noise disturbances to predict the state of a linear system. Chaves (2010) research thesis provides a detailed evaluation of such a system. The system researched is based on the sensor fusion technique, which combines measurements from a DGPS, gyro and speed sensor. Through using a discrete-time Kalman filter implementation, the paper investigates the five essential stages that repeat every time interval. These stages are listed below (Chaves 2010):

- State Estimate Extrapolation (Propagation)
- Covariance Estimate Extrapolation (Propagation)
- Filter Gain Computation
- State Estimate Update
- Covariance Estimate Update

In Chaves thesis, he concludes that the introduction of the Kalman filter reduces velocity error by 50% over the raw GPS velocity measurement. Chaves acknowledges ‘without the Kalman filtering, the low-cost GPS-based collision warning system fails to meet performance requirements outlined by previous researchers’ (Chaves 2010).

Chave’s thesis appears to be based on Kobayashi et al. (1998) research of ‘fuzzy-logic-tuned Kalman Filter (FLKF) sensor fusion technique that appropriately combines measurements from a vehicle DGPS with a yaw-rate gyro and speed sensor information. The Kalman filter equation builds on a dynamical relationship among the sensors, while fuzzy logic tunes the parameters of the Kalman filter algorithm.’ The results demonstrated from the experiments undertaken the effectiveness of the FLKF which traced out the path taken by the experimental automobile with consistent accuracy and repeatability (Kobayashi, et al. June 1998).

The DGPS system designed and evaluated in this thesis could largely benefit with the Kalman filter sensor fusing technique, to improve the accuracy and reliability of the system. Due to time constraints and the issues with wireless communications, the Kalman filter technique was not investigated. Using Murdoch University’s extensive capability in the Instrumentation and Control Engineering, this project expansion would prove to be a challenging project for future engineering students. The result would be a highly functional system that could be integrated into future robotics projects.

5.3 Cost Analysis

The DGPS project, although small, required the procurement of various components and parts to build the prototype for evaluation. The cost analysis shown in Table 39 is a breakdown of the market price for the components used in this project as well as any additional research and development costs involved. The EM-406a was supplied by Murdoch University and the market price included is to give the value of the system.

Table 39: Breakdown of Prototype and Research & Development

Part	**Cost**	**Qty**	**Total**	**Supplier**
Arduino Mega	$70.00	1	$70.00	Little Bird Electronics
EM-406a	$79.97	2	$159.74	Little Bird Electronics
EM-406a interface cable	$3.77	2	$7.54	Little Bird Electronics
LCD Keypad Shield	$27.00	1	$27.00	Little Bird Electronics
USB-TTL Module	$12.00	1	$12.60	Little Bird Electronics
Freight	$15.00	1	$15.00	Little Bird Electronics
APC200	$29.90	1	$29.90	Sure-Electronics
Freight	$25.00	1	$25.00	Sure-Electronics
High Gain Antenna	$6.00	2	$12.00	Ebay
Bread Board Large	$15.00	1	$15.00	Ebay
Bread Board Small	$10.00	1	$10.00	Ebay
Bread Board Wiring	$10.00	1	$10.00	Ebay
Duracell Li-Ion Battery	$30.00	2	$60.00	Bunnings Warehouse
Double sided Velcro	$8.50	1	$8.50	Bunnings Warehouse
Solder	$5.00	1	$5.00	Bunnings Warehouse
		Total	**$467.78**	
Research and Development Material				
APC220	$50.93	1	$50.93	DRobotics Online
Freight	$5.51	1	$5.51	DRobotics Online
HopeRF HM-TR TTL	$39.68	1	$39.68	Futurlec
Freight	$15.00	1	$15.00	Futurlec
Arduino Duemilanove	-	1	-	Supplied by Murdoch
HopeRF HM-TR RS232	-	2	-	Supplied by Murdoch
HopeRF HM-TR Evaluation	-	2	-	Supplied by Murdoch
Capacitors	-	10	-	Supplied by Murdoch
Max232	-	2	-	Supplied by Murdoch
		Total for R&D	**$111.12**	

Note: All prices are in AUD and priced at 5/6/2011

5.4 Conclusion

The differential global positioning system project has advanced my personal understanding and knowledge of microcontrollers and the limitless possibilities for design and implementation. To oversee the various stages of the project from the starting concept to the fully functional prototype has not only enabled me to gain valuable engineering experience but also an introduction to project management. The achievements in this project will enable the Murdoch University School of Engineering to grow its crucial understanding and development of robotic systems. Using such a system in robotics can create a reliable and accurate platform for the development of autonomous control.

Using available resources, each component chosen within the system design was evaluated to test if its inclusion would be appropriate to the system outcome, to achieve reliability and accuracy. Various changes were made and further research and procurement of components was required to achieve the present outcomes of the system. Thorough evaluation was conducted to ensure that the system was pushed to its capacity to yield the strong results achieved.

The challenging introduction to the Wiring software language meant that the skills learnt from industrial computer systems study could be adapted to understand and develop software within this language scope. The software was developed through using both open source software libraries and engineered code, additional and advanced functionality was integrated into the system.

The initial aim of the project to produce a low cost and accurate system was not only achieved but demonstrated the power and abilities of the evolving embedded microcontroller open source hardware and software communities. The final prototype results are shown in Table 40 supporting the above conclusion. Through the continued development of projects which utilize these platforms, Murdoch University students specializing in areas of industrial computer systems and instrumentation & control engineering will be ultimately able to enter the workforce with highly adaptive skills.

Table 40: DGPS Prototype Results

Type	Result	Comment
Accuracy	0.216m	Dependent on operation location (Averaged distance testing in open space)
Error	0.216%	Per 100 meters
Operation Range	882m	Tested in Harsh Environment
Battery Operation	Base: 7hrs Mob: 3hrs	Approximation 2x 1150mAhr Li-Ion Battery
Memory Usage	19k	
CPU Run Time	218ms	

Reference List

aliasmrjones. *How to Calculate Distance and Bearing to a Latitude Longitude Waypoint in Excel.* April 30, 2009. http://blog.deathpod3000.com/?p=192 (accessed June 11, 2011).

Arduino. "Arduino Duemilanove." *Arduino.* 2009. http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove (accessed June 10, 2001).

Arduino Forum: arbarnhart . "Distance Calculation Software." 01 07, 2011. http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1294325235 (accessed June 10, 2011).

Arduino Reference. *float.* 2011. http://www.arduino.cc/en/Reference/Float (accessed June 10, 2011).

Arduino. "Arduino Mega." *Arduino.* 2011. http://www.arduino.cc/en/Main/ArduinoBoardMega (accessed June 10, 2011).

Bennett, Peter, and Joe Mehaffey. "NMEA Data." *NMEA Data.* n.d. http://www.gpsinformation.org/dale/nmea.htm (accessed June 10, 2011).

Chaves, Stephen M. *Using Kalman Filtering to improve a low-cost GPS-based collision warning system for vehicle convoys.* Thesis, Pennsylvania: Pennsylvania State University, 2010.

Cisco Systems Press. "Internetworking Technology Handbook." *Point to Point Protocol (PPP).* August 1999. http://www.cisco.com/en/US/docs/internetworking/technology/handbook/PPP.html (accessed November 5th, 2010).

Digi International Inc. *XBee / XBee-PRO RF Modules.* Manual / Specification, Minnetonka, MN: Digi, Inc., 2009.

FairChild. "MM74C14 Hex Schmitt Trigger." *Jaycar.* January 1999. http://www.jaycar.com.au/images_uploaded/MM74C14.PDF (accessed June 10, 2011).

Garmin. "What is GPS?" *GARMIN.* 2011. http://www8.garmin.com/aboutGPS/ (accessed June 10, 2011).

GlobalSat Technology Corporation. *EM-406a: GPS Receiver Engine Board.* Taiwan: GlobalSat Technology Corporation, 2007.

Hart, Mikal. "Arduiniana." *TinyGPS.* 2011. http://arduiniana.org/libraries/tinygps/ (accessed June 10, 2011).

—. "Arduiniana." *NewSoftSerial.* 2011. http://arduiniana.org/libraries/newsoftserial/ (accessed June 10, 2011).

Hope Microelectronics CO. LTD. *HM-TR Series UHF Wireless Transparent Data Transceiver.* China: Hope Microelectronics CO. LTD, 2008.

Jensen, Anna B.O, and Cathryn Mitchell. "GNSS and the Ionosphere." *GPS World*, 2011: 40-48.

Kobayashi, Kazuyuki, Ka C. Cheok, Kajiro Watanabe, and Fumio Munekata. "Accurate Differential Global Positioning System via Fuzzy Logic Kalman Filter Sensor Fusion Technique." *IEEE Transactions on Industrial Electronics, VOL. 45, No. 3*, June 1998: 510-518.

"LCD Keypad Shield." *Little Bird Electronics.* 2011. http://littlebirdelectronics.com/products/lcd-keypad-shield (accessed June 1, 2011).

lm4871n. "SMA male crimp 1.5M RG174 433MHz Antenna cable." *Ebay.* 2011. http://cgi.ebay.com.au/ws/eBayISAPI.dll?ViewItem&item=270680271502&ssPageName=STRK:MEWNX:IT (accessed June 10, 2011).

Longitude Store. *The Haversine Formula.* n.d. http://www.longitudestore.com/haversine-formula.html (accessed June 10, 2011).

MAXIM. "Maxim - Multichannel RS-232 Drivers/Receivers." *Maxim.* Janurary 2006. http://www.maxim-ic.com/datasheet/index.mvp/id/1798.

McNeff, Jules G. "The Global Positioning System." *IEEE Transactions on Microwave Theory and Techniques, VOL. 50, No. 3*, March, 2002: 645-652.

Moore, T. "An Introduction to Differential GPS." *Lecture Notes, Institute of Engineering Surveying and Space Geodesy (IESSG).* University of Nottingham (UK), 2002.

NATO - RTO. "Differential Global Positioning System (DGPS) for Flight Testing." *Systems Concepts and Integration Panel.* Oct 2008. http://ftp.rta.nato.int/public//PubFullText/RTO/AG/RTO-AG-160-V21///AG-160-V21-01.pdf (accessed January 2011).

Python. *Haversine.* 2011. http://packages.python.org/seawater/_images/math/97aa41f6fb395759343a8f98652ed359f9263108.png (accessed June 10, 2011).

SHENZHEN APPCON TECHNOLOGIES CO.LTD. *APC Series Transparent Transceiver Module APC200.* Product Specifications, Shenzhen: APPCON Technologies, 2008.

SHENZHEN APPCON TECHNOLOGIES CO.LTD. *APC Series Transparent Transceiver Module APC220-43.* Product Specifications, Shenzhen: APPCON Technologies, 2008.

SHENZHEN APPCON TECHNOLOGIES CO.LTD. *APC Series Transparent Transceiver Module APC230-43.* Product Specifications, Shenzhen: APPCON Technologies, 2008.

Sparkfun Electronics. "32 Channel San Jose Navigation GPS 5Hz Receiver with Antenna." *Sparkfun Electronics.* 2009. http://www.sparkfun.com/products/8266 (accessed June 10, 2011).

Techie. "Explaining Interleaving." *TechGeeks Blog.* 2011. http://techgeeks-online.com/2010/explaining-interleaving/ (accessed June 10, 2011).

Veness, Chris. "Calculate distance, bearing and more between Latitude/Longitude points." *Movable Type Scripts.* 2010. http://www.movable-type.co.uk/scripts/latlong.html (accessed June 10, 2011).

Wang, Charles, Dean Sklar, and Diana Johnson. "Forward Error-Correction Coding." *AEROSPACE.* 2002. http://www.aero.org/publications/crosslink/winter2002/04.html (accessed June 10, 2011).

Wikipedia. *Unit Circle.* 2011. http://en.wikipedia.org/wiki/File:Unit_circle.svg (accessed June 10, 2011).

Wikipedia, Unknown. "Interleaving." *Wikipedia.* May 31, 2011. http://en.wikipedia.org/wiki/Interleaving (accessed June 10, 2011).

Appendixes

DGPS001: Software Tag Names

DGPS002: Software DGPS

DGPS003: Software Libraries

DGPS004: HopeRF HM-TR Manual

DGPS005: APC200 Manual

DGPS006: APC220 Manual

DGPS007: APC230 Manual

DGPS008: EM-406a Manual

DGPS009: Evaluation Data

DGPS010: Prototype Photo

DGPS011: Open Source License

DGPS012: Australian Maritime Safety Authority - Differential Global Positioning System