

# M3X8645 - BEng Major Project P144246

*by* Ian John Charles Trick

---

|                |                                                                                                     |                 |       |
|----------------|-----------------------------------------------------------------------------------------------------|-----------------|-------|
| FILE           | 21111_IAN_JOHN_CHARLES_TRICK_M3X8645_-<br>_BENG_MAJOR_PROJECT_P144246_78196_517984984.DOCX (11.35M) |                 |       |
| TIME SUBMITTED | 21-APR-2016 12:09PM                                                                                 | WORD COUNT      | 15365 |
| SUBMISSION ID  | 56191440                                                                                            | CHARACTER COUNT | 82137 |

## **GPS Vehicle Mileage Tracking System**

by

Ian Trick – P144246

Supervisor: Gareth Jones

Second Marker: Gary Jones

Project submitted as part of the requirements for the award of  
BEng (Hons) Electrical and Electronic Engineering

April 2016

## DECLARATION

I, .....(FULL NAME) ..... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

**Signed:** .....

**Date:** .....

## FORM OF CONSENT

I .....(FULL NAME) ..... hereby consent that my Project, submitted in candidature for the BEng (Hons) Electrical and Electronic Engineering degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

*Signed:* .....

*Date:* .....

## Table of Contents

|                                                              |               |
|--------------------------------------------------------------|---------------|
| <b>ABSTRACT .....</b>                                        | <b>- 5 -</b>  |
| <b>INTRODUCTION .....</b>                                    | <b>- 6 -</b>  |
| <b>RATIONAL .....</b>                                        | <b>- 7 -</b>  |
| <i>Projected Time/Planning .....</i>                         | <i>- 9 -</i>  |
| <b>RESEARCH .....</b>                                        | <b>- 12 -</b> |
| <i>Existing Technology Review .....</i>                      | <i>- 14 -</i> |
| <i>Comparison .....</i>                                      | <i>- 18 -</i> |
| <i>GPS .....</i>                                             | <i>- 19 -</i> |
| <b>RESEARCH CONCLUSION .....</b>                             | <b>- 29 -</b> |
| <i>Scope .....</i>                                           | <i>- 30 -</i> |
| <i>GPS .....</i>                                             | <i>- 30 -</i> |
| <i>Proposed solution .....</i>                               | <i>- 30 -</i> |
| <i>Projected Cost .....</i>                                  | <i>- 32 -</i> |
| <b>DESIGN .....</b>                                          | <b>- 33 -</b> |
| <i>Specify Hardware .....</i>                                | <i>- 33 -</i> |
| <i>Structure drawing .....</i>                               | <i>- 46 -</i> |
| <i>Flowchart .....</i>                                       | <i>- 47 -</i> |
| <i>Pseudo Code .....</i>                                     | <i>- 47 -</i> |
| <i>Schematic design .....</i>                                | <i>- 48 -</i> |
| <i>PCB Design .....</i>                                      | <i>- 51 -</i> |
| <i>Software design .....</i>                                 | <i>- 52 -</i> |
| <i>Software testing .....</i>                                | <i>- 72 -</i> |
| <b>COSTINGS .....</b>                                        | <b>- 73 -</b> |
| <i>Hardware cost .....</i>                                   | <i>- 73 -</i> |
| <b>IMPLEMENTATION .....</b>                                  | <b>- 74 -</b> |
| <i>Purchase hardware .....</i>                               | <i>- 74 -</i> |
| <i>Build .....</i>                                           | <i>- 75 -</i> |
| <i>Testing .....</i>                                         | <i>- 77 -</i> |
| <b>ANALYSIS .....</b>                                        | <b>- 86 -</b> |
| <i>Analysis methods .....</i>                                | <i>- 86 -</i> |
| <i>Record data .....</i>                                     | <i>- 91 -</i> |
| <b>FUTURE ENHANCEMENTS .....</b>                             | <b>- 92 -</b> |
| <i>Hardware Improvements .....</i>                           | <i>- 92 -</i> |
| <i>Software Improvements .....</i>                           | <i>- 93 -</i> |
| <i>Other Improvements .....</i>                              | <i>- 93 -</i> |
| <b>APPENDICES .....</b>                                      | <b>- 95 -</b> |
| <i>Appendix A – Researched technologies .....</i>            | <i>- 95 -</i> |
| <i>Appendix B - Feasibility Documents .....</i>              | <i>- 95 -</i> |
| <i>Appendix C – Design Documents .....</i>                   | <i>- 95 -</i> |
| <i>Appendix D – Costings &amp; Hardware datasheets .....</i> | <i>- 95 -</i> |
| <i>Appendix E – Test &amp; Analysis documentation .....</i>  | <i>- 95 -</i> |
| <i>Appendix F – Future Enhancement documentation .....</i>   | <i>- 95 -</i> |
| <b>REFERENCES .....</b>                                      | <b>- 96 -</b> |

## Abstract

An insight of how to design and build a mobile GPS vehicle mileage tracker, that will automatically record the route of the vehicle has travelled and take the worry away from manually imputing those business miles.

The brief covers digital communications, C programming, the use of software simulation tools such as Proteus VSM and Easy PC.

The authors motivation to design and build such a device was to try an improve the efficiency of the way they recorded business miles as sitting down at the end of each month for a number of hours manually imputing the data was frustrating for the author.

No matter how you look at it, for HMRC purposes business expenses have to accurate and readily available for audit by a tax inspector therefore you can get away with not having mileage records, the purpose of this brief is provide an automatic way of recording mileage to reduce user input. The authors approach was to look at existing technology to see if there was room for development to improve these existing technologies. From Researching the existing technologies this lead the author to technical research around digital communications.

The development of the proposed device suffered pit falls due to the authors level of knowledge and experience around programing, therefore changes had to be made at later stages of the design to resolve the potential failure of the project, however the changes were implement swiftly and the project was deemed as a success for the author.

To conclude the author has gained value knowledge and skills they can take forward into future projects, where they will need to think more strategically on how likely are the outcomes of the project going to be successful, measure the potential of the outcomes and consider if they need a backup plan.

## Introduction

The author of this report is a part time student studying an Electronic & Electronic Engineering degree, as part of the degree program the student has produce a major project that demonstrates their knowledge of Electronic and Electrical Engineering in particular demonstrating the academic skills and knowledge they have gain on the degree program.

The author of this report, has taken time to consider various different Electrical/Electronic engineering projects that have enough scope to satisfy the requirements of the Electronic engineering to a Bachelor of Engineering degree. Initially the author was considering using a work based project that they would able to continue as part of their normal day job that would potentially produce a higher quality project, the work based project that the author was considering as a major project would satisfy the PLCs & Controls module. The project being considered was to produce a design Automation and Electrical project around the replacement of an aeration plant in a waste water treatment works, however due to internal business changes project milestones were at risk of slipping, the author could foresee considerable risk to the project delivery timescales that have the potential to go out of sync with the academic timetable. Considering the risk of program slippage, the author made a decision to develop a project outside of work to mitigate against the risk of time constraints and ultimately an uncompleted project.

The non-worked based project to be taken forward to be developed was an electronics project that was likely to require the use of a microcontroller, if this project was developed it would satisfy the Electronic Systems Design Module which is part of the Level 6 modules on the Electronic & Electronic Engineering degree.

## Rational

As briefly discussed in the introduction the author of this document has taken time to consider potential projects to develop and has chosen to develop an electronics basis project which is the design of a vehicle mileage tracker it is thought this would satisfy the Electronic System Design Module of the degree program.

The rationale behind the device is the author has to accurately record business mileage as part of their job due to traveling around site to site for work purposes. The author has tried a number of methods to try to reduce the time spent recording and inputting mileage, however they have found these methods to be time consuming or not accurate therefore the author has a keen interest in developing an automated system which could save them time post project.

Recording mileage accurately is a requirement of the HMRC for all business expenses, inaccurate mileage records can lead to fines from the HMRC. Where large fleets are being managed there is a business case to keep accurate records apart from the mitigation from potential fines.

- A reduction in claimed mileage expenses are mileage is recorded electronically and will be more accurate than manual input.
- As above a reduction will also see a carbon reduction.
- Time/admin for drivers inputting information into an expense system.
- Mileage record audits will be quicker saving time.

The proposal for the design of a vehicle mileage tracker, will be to record GPS information on any type of journey taken by a vehicle, therefore it is expected the device will have a requirement to start up automatically as soon as the vehicle ignition is switched on and then to start recording data as the vehicle moves from its original position. Consideration will need to be taken on how long the device takes to initialize and to lock onto the GPS Satellite signals.

As the vehicle changes position the tracker will receive new information on the vehicles location and time, at derived intervals this information will have a requirement to be stored locally on the devices data storage facility. The location and time information will be continued to be stored throughout the journey until the vehicle has stopped and the ignition switched off. It is thought retrieving the data stored on the device could be done by using serial communication via a USB port to a PC, if the data stored on the device is in CSV format (comma separated values) it could be viewed in a Microsoft Excel spreadsheet.

Initial thoughts, considering the project brief the design will be a type of electronic device that will require a microcontroller, the vehicle mileage tracker will require the ability to record GPS data for a period of up to possibly 1000 miles to avoid over writing the stored data. The vehicle tracker must be portable and adaptable for any type of road legal vehicle in the UK, with this in mind the device will certainly be battery powered but may need a backup power supply derived from the vehicles 12VDC plug or USB port on modern vehicles.

### Projected Time/Planning

The planning stage outlines a program detailed with activities that will required to be complete in sequential order, the activities will be grouped into stages that are known as milestones. The milestones that are expected to be met to complete this project have been summarized below, the activities below are include within the project program found in appendix B

### Research Activity

Critical to any innovation, without research the engineer could spend significant time in development to find that all the time spent has been in vain, therefore research is key activity to the project without focused research the project could be delayed or even worse mothballed at a later stage in the project. e.g. the technology already exists or a key element of the design has become obsolete.

The type of research and method of research has got to be considered, for this design the type of research will be;

- Research Philosophy – Positivism
- Research Approach – Deductive
- Research Strategy – Case Study's on existing technologies, Archival research through online resources on GPS, electronic theories such as control & communication.
- Research Choices – Mixed Methods
- Research Time Horizon – Longitudinal & Cross Sectional.
- Research Techniques and Procedures – Data Collection

### Feasibility Activity

The research will identify what can be done, however the available time, costs and skills will provide the answer to what type of project is feasible.

### Design Activity

The purpose of the design stage is to pull together the information found in the research and feasibility stage, to produce a legible design for the device, this activity will be split into sub-sections to define the design, these stages will be;

- Specification
- Hardware Identification
- Structure drawings & Flow charts
- Schematics drawings
- PCB Design
- Pseudo Code design
- Program coding
- Software testing

### Purchasing activity

Purchasing the hardware can only to be carried out when the design has been complete or when the hardware identification stage has been complete. The key to purchasing is firstly the cost, but most important the availability therefore during the feasibility stage hardware lead times will be identified.

### Implementation/Build Activity

As it forms part of the later analysis stage the implementation activity may be split into 2 sub-sections, prototype build to confirm the design works on the selected hardware and the final build where the final designed device is built.

### Testing Activity

Testing the device will be a staged activity that will run in parallel with the build of the device, testing at each stage of the build recording relevant information that will be able to be found within the Appendixes within this brief.

### Analysis Activity

The analysis will follow testing of the device; this activity will be to prove the device can record data at different locations.

### Conclusion Activity

The conclusion will be summary of the all the activities included in this design brief, highlighting what went well what didn't go well and how it could be done differently in the future. Possibly an explanation on what the author has gained in experience and knowledge by carrying out this project, and how the skills could be implemented elsewhere. Finally, a Statement on if the project was a success or not, and the reasons justify this statement.

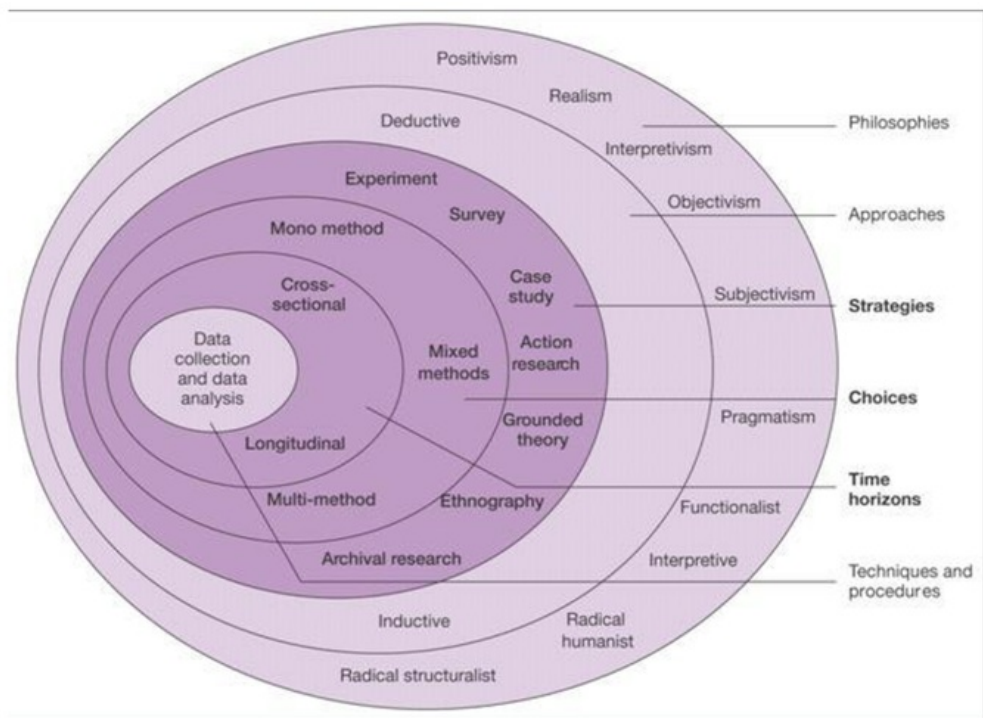
## Research

The type of research required to deliver the project has been identified by the report author, reviewing the research onion in fig R1. the author has identified the following research methods that have been used throughout the report.

- Research Philosophy – Positivism as this research philosophy will identify areas of information that already exist, where data can be collected, observed and quantified.
- Research Approach – A deductive approach will be used as the research subject is known and the end result is to confirm how the project is going to be designed & built, also the research approach will continue through the life of the project until completion.
- Research Strategy – Case Study's as the author will be reviewing other projects that have been published through online resources to gain knowledge on the technical aspects of the subject, Archival research where the author will be using library and online resources to gain academic knowledge on different subject methods.
- Research Choices – Mixed Methods
- Research Time Horizon – Longitudinal & Cross Sectional.
- Research Techniques and Procedures – Data Collection

The author is said to use a positivism research philosophy, the research will be a deductive approach using online archives and case studies to research existing mileage recording technology, GPS what it is and how it works, which is likely to lead to other research such as digital communication methods, programing, etc....

Fig R1 [1]



## Existing Technology Review

Researching archival information through online and library resources existing GPS technology is found to be a broad subject that would take extensive time to fully understand all types of GPS devices and their different functions, therefore the research choice has had to be through mixed methods longitudinally to get an understanding of more than one technology researching over a longer time period.

Online resources have shown that various type of products and devices have GPS capabilities for recording distance travelled, for the purpose of relevant research around the proposed device the author of the document has categorized the types of GPS devices into 3 different groups;

- Professionally installed trackers used for fleet management, where real time information is critical to the fleet manager or the client/customer.
- Plug and Play devices where real time information is not important or required, but mileage data is required to be recorded with minimal user input.
- Smart Phone apps where functionality is limited, requiring user input and duplication if expenses have to be inputted into a corporate system.

### **Fixed installation trackers & data logging systems**

Researching online archival resources such as product manufacturer marketing websites have found fixed installation trackers are Real Time devices, that record data for basically everything from where the vehicle is at a specific time to how economical the vehicle is being driven or by whom.

The mileage accumulated is recorded with the speed the vehicle has been travelling using GPS Data, location messages are sent when requested or at specific locations to notify a Manager/ client/customer their delivery is in transit. The devices have also been found to be able to record speed, driving style and also the functionality to report this or other relevant information via a messaging system [2]

### **Vehicle tracking systems**

As discussed in the previous paragraph vehicle tracking systems are used to manage vehicle fleets the system provides automatic vehicle location to the controller, client, customer, Etc... These tracking systems have been found to use GPS or GLONASS technology, the location of the vehicle can be seen using an electronic mapping system which imports the GPS and or GLONASS information from the Internet.

Vehicle tracking systems are commonly used in cities to provide users of public transport information for their planned journeys, expected arrival and departure times etc...

Active/passive tracking - Active tracking is a means of providing up-to-date almost real-time tracking information to the user/s, passive systems normally store information then send the information remotely at the time of an event for example a status update at particular location that has an importance to the user/client/customer/etc... Passive devices normally require the tracker to be removed and connected to a PC to download the stored data.

History—vehicle tracking systems were introduced by the French in the 1978 with the system called telematics, as a system developed it used GPS.

*"Telematics is a translation of the French word "télématique" which was first coined by Simon Nora and Alain Minc in 1978 report to the French government on the computerization of society. It referred to the transfer of information over telecommunications and was a merging of the French words "telecommunications" and "informatique". The original broad meaning of telematics continues to be used in academic fields but in commerce it now generally means vehicle telematics.*

*The convergence of telecommunications and information processing, the term later evolved to refer to automation in automobiles, such as the invention of the emergency warning system for vehicles. GPS navigation, integrated hands-free cell phones, wireless safety communications and automatic driving assistance systems all are covered under the telematics umbrella.*

*The science of telecommunications and informatics applied in wireless technologies and computational systems. 802.11p, the IEEE standard in the 802.11 family and also referred to as Wireless Access for the Vehicular Environment (WAVE), is the primary standard that addresses and enhances Intelligent Transportation System.*

*Vehicle tracking is monitoring the location, movements, status and behaviour of a vehicle or fleet of vehicles. This is achieved through a combination of a GPS(GNSS) receiver and an electronic device (usually comprising a GSM GPRS modem or SMS sender) installed in each vehicle, communicating with the user (dispatching, emergency or co-ordinating unit) and PC-based or web-based software. The data is turned into information by management reporting tools in conjunction with a visual display on computerised mapping software. Vehicle tracking systems may also use odometry or dead reckoning as an alternative or complementary means of navigation.*

*GPS tracking is usually accurate to around 10–20 metres, but the European Space Agency has developed the EGNOS technology to provide accuracy to 1.5 metres "[3].*

### **Plug & Play Tracker**

The Plug & Play devices researched as part of this brief, focus is around recording mileage and GPS data only;

- Mileage tracking
- Location details
- Cloud services
- Google maps

### **Smart Phone apps**

Using GPRS (General Packet Radio Service) e.g. mobile data, using GPRS mobile data with a mapping application on a mobile device to record miles travelled.

- Mileage tracking only?

## Comparison

### GETOTAB Go5 [3]

- Coms GPS & GPRS
- Plug & Play device
- Small & compact
- Precise tracking
- Live tracking

### Blue water TE2[3]

- Coms GPS only
- Plug & Play device
- Small & compact
- Precise tracking
- Windows compatible software

### XX Manufacturer

### Comparison table

| Manufacturer | Geotab                      | Bluewater                   |  |  |
|--------------|-----------------------------|-----------------------------|--|--|
| Type         | P&P Tracker                 | P&P Tracker                 |  |  |
| Size /weight | 5 cmx 2.5 cm x 7.5cm<br>65g | 3.38" x 2 x 0.98"<br>2.8 oz |  |  |
| Construction | ABS                         |                             |  |  |
| Memory       | 32mb Flash                  | 512k                        |  |  |
| Voltage      | 11 – 16v                    | 12v                         |  |  |
| Power        | 150mA                       | 180mA                       |  |  |
| Coms         | GPS & GPRS                  | GPS                         |  |  |
| Cost         | POA                         | \$349.95                    |  |  |

## GPS

### GPS Overview

Global Positioning System, originally developed for intended use with U.S. Department of Defence (DoD) applications in the 1970's, to provide continuous real-time 3-dimensional positioning, navigation and timing data worldwide, however in the 1980's the government introduced the use of GPS for civilian applications. As we know these days GPS technology can be found in many different Devices such as GPS navigation system, fleet management systems for Parcel delivery notification, smartphones, etc....

Since the launch of GPS technology in the 1970's in total 70 GPS Satellites have been developed and brought into service and replaced as each satellite has come to its end of life, at any time a minimum of 24 GPS satellites are in orbit above the earth, 24 is the number required to get full GPS coverage around the world. The target number of satellites for GPS use is 31 to allow for redundancy's; at the time of writing this report currently there are 24 GPS Satellites in service.

The GPS system is seen to be made up of three key elements;

1. Space the GPS Satellites
2. The control system, that is operated by the U.S. military, and finally
3. The user interfaces the GPS equipment i.e. the GPS receiver, Satellite Navigation systems, Sports watches, etc.

#### The space element

Power is provided to the GPS Satellites by 2 means, the primary power source is provided by sun seeking solar panels and NiCad batteries that provide the secondary power source, to ensure the GPS satellite provides accurate GPS information they have 4 on board atomic clocks, with only one being used for time.

### Satellite orbits

See Fig R2, in 6 orbital planes around the Earth there are 4 satellites in each, each plane is inclined 55-degrees relative to the equator, each GPS satellite set at the 55-degree angle maintains full operational capability with a redundancy of up to two satellite failures.

GPS satellites orbit the earth twice every 24 hours they are stationed in very precise orbit and at an approximate altitude of 11,000 nautical miles above the earth. The GPS Satellites transmits signal information back to earth, the GPS receiver uses this information to perform a calculation called trilateration where it compares the time a signal was transmitted by a satellite with the time it was received to determine its position within a 12,000-mile radius.

### Satellite signals

GPS Satellites broadcast radio signals continuously these are broadcast on two frequencies (L1 and L2), the broadcasted radio signals contain position and timing data, the signals are travelling at the speed of light and will take approximately 6/100ths of seconds to reach earth.

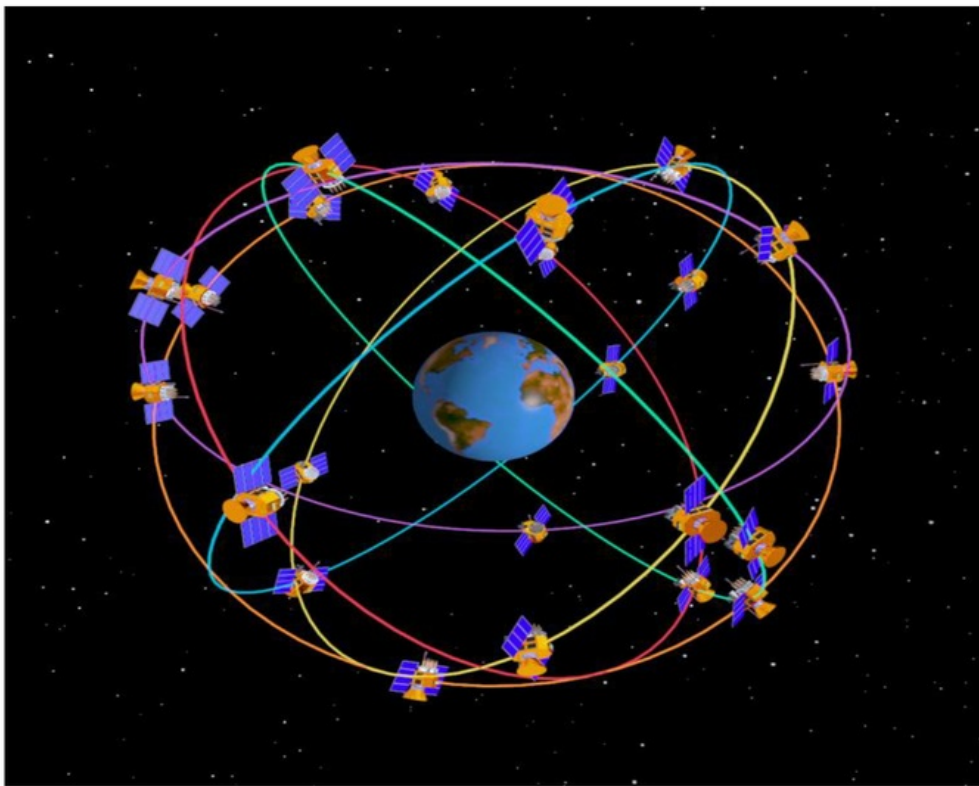
Without direct line of sight between the GPS satellite and the GPS receiver GPS signals are unlikely to be received as GPS signals find it difficult to penetrate obstacles. The two frequencies broadcast by the GPS satellites are L1 frequency (C/A code and P code), C/A (course acquisition) is the code that is available for civilian use that provides Standard Positioning Service (SPS), this service (SPS) the user can expect to achieve 15-meter horizontal accuracy for 95% of the time.

P code (precise code), is broadcast on both frequencies L1 and L2, P as in (PPS) is used for precise positioning Service that is only used by the military. The accuracy is significantly increased using (PPS) oppose to (SPS) hence the precise positioning service.

Control segment: U.S. DoD monitoring

*" The U.S. Department of Defence maintains a master control station at Falcon Air Force Base in Colorado Springs, CO. There are four other monitor stations located in Hawaii, Ascension Island, Diego Garcia and Kwajalein. The DoD stations measure the satellite orbits precisely. Any discrepancies between predicted orbits and actual orbits are transmitted back to the satellites. The satellites can then broadcast these corrections, along with the other position and timing data, so that a GPS receiver on the earth can precisely establish the location of each satellite it is tracking. "[4]*

Fig R2. [4]



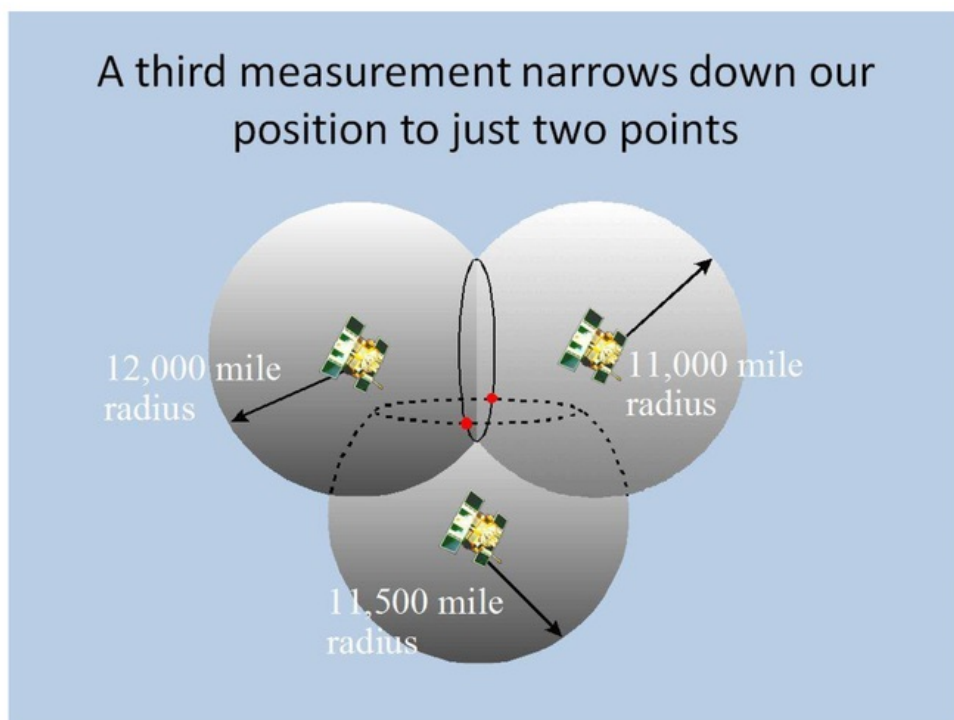
### How GPS works

A device that is intended for use to provide GPS data to a system or user will include a GPS receiver module, the basic function of a GPS Receiver module is to receive satellite data from a number of satellites to calculate its position on Earth within a small radius, following the receiver completing this action it will output a serial data stream in a format set by the user.

The GPS Receiver outputs a 2 dimensional positional on earth (time and velocity) using 3 satellites in view, a receiver can also provide 3 dimensional data (elevation) but this will require a fourth satellite in view.

See fig R3, each satellite provides a position within a 11,000 – 12,000-mile radius, using 3 satellites where the 3 circular radii intercept, there are 2 positions where the GPS is possibly located, 1 coordinates are in a position in outer space, therefore the other coordinates are the position of the GPS receiver.

Fig R3 [4]



Satellites to GPS Receiver

Satellites transmit GPS data at around the speed of light

\*The Binary Phase Shift Keying Data Select Sequence Streaming explanation

GPS signal processing—the GPS receiver module when considered as a sensor has three primary functions, demodulating the GPS navigation data, measurement of range and rate of range. The GPS signal contains a modulated 50-bits a second navigation data stream, the navigation data contains information on the orbital parameters and the satellite clock which is used to calculate position the receiver. The format of the GPS signal is known as direct sequence spread spectrum; the spread spectrum has three characteristics:

Referenced directly from paper – GPS RECEIVER ARCHITECTURES & MEASUREMENTS

*“1) the data are modulated onto the carrier such that the transmitted signal has a larger (and usually much larger) bandwidth than the information rate of the data, hence the name “spread spectrum”;*

*2) a deterministic signal, known a priori to the receiver, is used by the transmitter to modulate the information signal and spread the spectrum of the transmitted signal; and*

*3) the receiver cross correlates the received signal with a copy of the deterministic signal in the process of demodulating the data. By so doing, the receiver can recover the transmitted data. “ [5]*

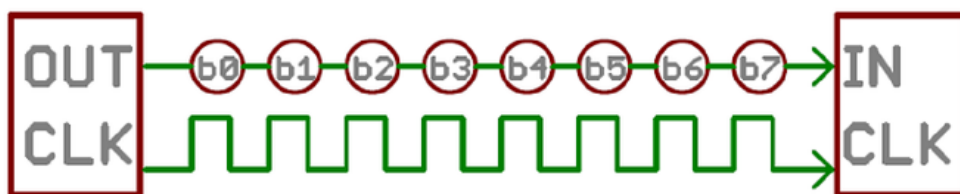
#### **GPS Receiver Module Function**

- Demodulates signal
- Calculates position from demodulated data
- Transmits a serial data stream through a number of different protocols
  - Synchronous serial data stream such as SPI & I<sup>2</sup>C
  - Asynchronous serial data stream, NMEA
  - National Marine Electrical Association Protocol; data is presented as 8-bit binary format each bit representing an ASCII character.

### Serial Communication Bus

Serial communication is a method of digital communication which uses what is known as a serial data stream, basically transmitting or receiving one bit of information at a time, which can be on a clock pulse using 2 wires as seen in Fig R4 below. Serial communication can be done by using between 1 wire and 4 wires dependent of the communication method, serial communication advantage over parallel is a reduced pin count parallel communication requires between 5 and 9 pins.

Fig R4. [6]



Communicating with a GPS receiver, through research the author has found the means of communication from a GPS receiver is commonly is serial data stream, a GPS Receiver will output a serial data stream at its serial ports, interfacing this Serial Bus will be dependent on the selected device capability as many IC's have more than one serial port. The types of serial ports that have been commonly found on GPS receiver's devices are I2C, Jtag, SPI and UART, each interface has been researched below.

**I2c** – Is a trade name the Inter-Integrated-Circuit (IIC, I see, I2C), the serial Interface bus for communication between chips was originated by Phillips (now NXP). I2C only uses two wires addressing is sent first on the same line as data is sent or received data can only move one direction at the time this is known as half duplex Direction of the data is specified by the address I squared C is a master slave architecture however any device on the bus can become Master when the current master stops transmitting data. I2C consists of a clock line (SCL) and a data line (SDA), both lines have resistive pull-ups to a positive voltage.

I2C can manage up to 1008 slaves data rate started at 100kbits/s, then full 400kbits/s, fast 1mbits/s, high speed 3.1mbits/s

### Jtag

A useful chip interface, with their English and history is the Jtag (Joint test action group) standard, also known as “boundary scan,” or IEEE 1149.1. It was devised in the 1980s to deal with the thorny problem of testing for components or connection failures in Circuit boards using the then-new Surface-Mount and multi-layered technologies; these made it increasingly difficult to get at the AC pen connections, not to mention that in heart of the chips themselves. As such it provides a way to look at the registers and detail parts within the chips, usually simple for wire serial bus; that lets you pinpoint what is broken without having to unsolder the multiple parts to get at the problem.

J tag turned out to have additional utility as a general-purpose serial interface, akin to SPI and a I2c, because of the way it accommodates an internal state machine in the target device for controlling data transfers. Among other applications, it is now widely used to program and debug microcontrollers in circuit (e.g., ARM and AVR series), and to program cPLDs, FPGAs, and non-volatile memory (Flash, EEPROM), as well as other chips that include non-volatile memory (e.g., the Maxim DS 4550, where coexists with an I2C port).

The basic scheme is shown in figure 14.43. A clock line (TCK) and a mode select line(TMS) are Bused to all devices; the other two lines are daisy-chained through the chips, and are called data in (TDI) and data out (TDO). TDI bits are clocked in on the rising edge of TCLK, and TDO bits are clocked out on the falling edge. The mode bits, which define what TDI and TDO will be doing, are clocked in on TCLK 's rising edge. (There is also an optional bused reset line, not surprisingly called TRST.) The bus runs at speeds in the range of 1–100 Mb/s, depending on the particular target chip(s). [7]

### SPI

Motorola introduced Serial peripheral interface (SPI), its is commonly used as a communication interface between ICs

SPI requires 4 wires, Clock Pulse, 2 Data lines an Output n& Input, also a Slave Select pin When specifying a device pin count can be critical, SPI requires Too many pins, the high date transfer rate of SPI is not required, master requires additional pin on the Microcontroller for every slave.

#### UART – (RS232, RS-485)

Universal Asynchronous Receive Transmit, does not have a clock reference therefore to ensure the best free data transfers it requires different mechanisms to transmit data.

These are a start bit to confirm the start of a Data stream, data bits containing the data, parity bits for error-checking, a stop bit confirming the data stream is complete and the baud rate (how fast the data is sent of serial line). A Data stream is illustrated in figure R5 below

Fig R5 [8]



UART Baud Rate is expressed in bits per second(Bps), standard baud rates are 1200, 2400, 4800, 9600, 19200, 30400, 57600, and 115200. Baud rates have to be the setup the same at each ends of the bus to ensure effective communication between the devices

has a limited data transfer rate compared to other technology, but through reviewing the at max I will only need to transfer approx. 84Bytes second at the standard baud rate 4800 this data can be transferred at 5 times a second **RS232 & Uart** only requires 3 cables receive, transmit and ground.

#### Comparison

SPI/I2C - When we look at comparing SPI with I2C, the latter is a more complex protocol, that is not as well suited to steady high-rate streaming of data oppose to SPI.

*“The flexibility of multi bus mastership brings with the problems of contention and arbitration. With I2C you have to allocate multiple devices unique addresses, which is commonly handled by including some dedicated pins to select among a built-in set, this somewhat defeats some of the advantages of low pin count. And the flexibility of addressing, bidirectional data and bus mastership complicate debugging, as compared with the dead-simple SPI protocol.*

*Which to use*

*This can be determined by reviewing the peripheral chip Cu datasheet, which usually supports only one protocol all the other, however most microcontrollers include hardware support for both SPI and I2C, if not software can be written to reroute Microcontroller pins to utilize as serial pins but it does create a software overhead that will use up memory and slow down the operation of the device in comparison to dedicated hardware pins". [7]*

#### Programing Languages

Programming language, microcontroller instruction sets is an area of research that is required for the design brief as in order to build a GPS vehicle tracker the author will need to implement a piece of code for the device to work, therefore different languages and instructions have been researched to Identify requirements and suitability for this project.

#### Assembly language

The author has had some experience in using assemble language on previous projects within the last 3 years however at the time the author and peers found it difficult to understand when each function is required. What the author has taken from there previous experience is assembly language is not fit for all applications in reference to this project assemble language would be an overkill as precise control is not required. With this experience and the assumed competence in assembly language required to complete a major project without problems the author has decided to research other options.

### C Programming Language

Archival researching various online sources about programing in the author found one very useful resource called tech on the net, the resource outlines the programming language and provides examples, which author found it easy to understand after running through the tutorials. The author understands in the absence of experience in programming through previous work or academic study's it will be a step learning curve, however out of all the options for programing a microcontroller C is said to be the most intuitive.

## Research Conclusion

The authors research has found various ways that get the same result in respect to recording mileage, the less user input required the more the functionality the device has therefore the more it costs, however the more you spend you get a lot more data.

All devices and technologies researched are fit for purpose with the correct application, however to meet the requirement of this brief the author has indicated an outline specification.

- A functionality to Record mileage.
- Start/stop recording without intervention.
- Assumed to be portable as it is to be located within a vehicle.
- Capability to record data on the device
- USB Compatibility

Considering this outline specification indicated by the author within the introduction to this brief and repeated for clarity above, the type of device that is required for development is a plug & play device.

The fixed type GPS tracker technology capability goes beyond the requirements of the authors indicated specification therefore this will not be considered as a design to be developed at this stage however the features of this type of technology may be considered as a future development.

The smartphone mobile application to record mileage requires significant user input, and it has limitations in comparison to the other technologies research as part of this brief. Also significant programming would be required to interface the smartphones GPS technology and also graphics that would be required for the Mobile Application where the author has little programming knowledge or skills this would prove to be difficult, therefore it is considered as not suitable due to not meeting the indicated specification.

## Feasibility

### Scope

The scope of the feasibility is to identify the requirements and outline specification of a GPS tracker electronic device, what's the purpose of the device, how it should work and what information in what type of format Will be required from the device.

### GPS

As discussed earlier in the design brief, Global positioning system is a system that uses a number of satellites in orbit above the earth atmosphere used to determine a position anywhere on planet Earth. From anywhere on Earth the maximum number of satellites that can be seen at any one time by a GPS receiver is 12 due the positioning of each satellites. For the purpose of this project the electronic device will be a vehicle mileage tracker that will determined the distance travelled by using a GPS Receiver module.

### Proposed solution

The proposed solution will identify the preferred solution and the rational for the solution selection. As detail within the research section of this brief there are various methods of recording vehicle mileage from place to place accurately using mobile technology. Considering the 3 different types of mileage tracking researched the only one that could be developed that would provide enough scope to satisfy sufficient learning that meets all aspects of Electronic systems design would be the plug and play type device.

#### Solution Rational

The solution rational has be considered through the following methods

1. Does the project cover >75% of the learning outcomes of the ESD module?
  - a. Fixed tracker Yes, Plug & Play Yes, Mobile App No
2. Is there sufficient time to develop the projects?
  - a. Fixed tracker No, Plug & Play Yes, Mobile App Yes
3. Will the project require other skill sets that that are not included in the BEng degree program?
  - a. Fixed tracker Yes, Plug & Play No, Mobile App Yes

To summarise the rationale, the fixed tracker has sufficient scope for a project however considering the amount of scope there isn't enough time to complete it within the academic timetable and will also require skill sets are mobile communications that is not included within the BEng program. The mobile App technology project does fit well into the ESD module as it would be a purely software based project that would require a high level of competence in programming to complete therefore this solution is not being considered for development.

Finally, the plug & play vehicle tracker satisfies >75% of the ESD module there is sufficient time to develop this project and the skill set required are all included within the BEng degree program.

### Projected Cost

From reviewing other technologies, the projected outturn costs of the project are expected to be less than the technology researched considering this design material costs will be the only measurable. However, the manufacturers of the existing technology researched earlier will have access to cheaper material cost through bulk purchasing for mass production therefore the overall cost of this proposed build could be higher due to purchasing materials in low quantities.

Reviewing the table F1 below the project build will cost in the region of £100 - £120

Table F1

| Hardware                | Cost (£)               |
|-------------------------|------------------------|
| GPS Module with Antenna | Average cost £30 - £40 |
| Microcontroller         | £2.50                  |
| PCB                     | Est £15.00             |
| LCD Display             | Est £15.00             |
| SD Card Holder          | Est £5.00              |
| Voltage regulator       | Est £2.00              |
| PCB Components          | Est £10.00             |
| Connectors              | Est £5.00              |
| Enclosure               | Est £20.00             |
| Total                   | £114.50                |

## Design

### Specify Hardware

The type of hardware that is required to build this electronic device has been identified during the research and feasibility stage of this project, the device will require a number of key components as detailed below;

1. A GPS Receiver module to provide the device with navigation data, the key component and the primary purpose of the project, the GPS minimum specification;
  - a. Antenna internal or external
  - b. Dimensions – no issue
  - c. Start up time – 30 seconds or less
  - d. No. of Channels – 12 +
  - e. GPS Protocol – NMEA
  - f. Update rate minimum 1 HZ +
  - g. Operating voltage – 3V – 5VDC
  - h. Breakout board – Required
2. A Microcontroller the device used to represent the data to the user interface in format that can be understood easily, the minimum specification for a microcontroller
  - a. Operating voltage 3V – 5VDC
  - b. UART Coms
  - c. SPI Coms
  - d. USB Interface
  - e. PDIP Technology for through hole mounting
3. A PCB, that will be the platform to interlink the relevant hardware, the PCB will be design when all the hardware has been selected and a schematic diagram of the circuit has been complete. If the device was being design to be manufactured for mass production the size of the PCB would be critical, for the purposes of this design brief the PCB size is not critical.

4. LCD Display, the device that will provide a visual indication of the data being received by the GPS Module, the minimum specification for the LCD display is a 16 x1 Alphanumeric display.
5. Removable memory, this interface will be the means of storing the GPS data, this interface is seen as a module or a card holder on a breakout board connected to the microcontroller via the SPI communication port.
6. Power supply & Regulator, the regulator will need to regulate the backup power supply from the vehicle, therefore the regulator will have to be designed to withstand a maximum input voltage of 14.2VDC from the vehicle when the engine is running.
7. Other components, as the hardware is identified other components such as resistors, capacitors, diodes etc. will be derived from carrying out archival research of the hardware datasheets.

## Hardware Identification

### GPS Receiver Module

Archival researching various online resources for different types of GPS Receiver Modules, a suitable GPS module has been identified as an efficient method of gathering the GPS data being transmitted by the GPS navigation satellites orbiting the earth. The GPS module will decode the GPS satellite information to determine its position, following locking onto 3 or more satellites. The GPS receiver will then output usable data via its com port, for the purpose of this project the GPS data required was NMEA. The specification of the selected GPS module is detailed below referenced directly from the manufactures website.

Fig H1.[9]



**Sparkfun Venus GPS with SMA Connector**

**Features:**

- Up to 20Hz update rate
- -148dBm cold start sensitivity
- -165dBm tracking sensitivity
- 29 second cold start TTFF
- 3.5 second TTFF with AGPS
- 1 second hot start
- 2.5m accuracy
- Multipath detection and suppression
- Jamming detection and mitigation
- SBAS (WAAS / EGNOS) support
- 67mW full power navigation
- Works directly with active or passive antenna
- Internal flash for optional 75K point data logging
- Supports external SPI flash memory data logging
- Complete receiver in 10mm x 10mm x 1.3mm size
- Contains LNA, SAW Filter, TCXO, RTC Xtal, LDO
- Single 2.7-3.3V supply

**Dimensions:** 1.15 x 0.7 inches

### Microcontroller Justification

A microcontroller has been specified over a PLC, as the device is required to be portable and inexpensive therefore ticking the boxes for size and cost a microcontroller is the obvious choice for this application. Through researching an online resource [www.microchip.com](http://www.microchip.com) initially a large numbers of microcontrollers were found to be available that satisfy the functions required of this application therefore to avoid an unsuitable microcontroller being selected a minimum specification has been derived to meet the needs of this application.

Following identifying the GPS Receiver to be used in this design, the next logical process was to design structure diagram as seen in fig H3, from this diagram a hardware list was derived refer table H1. below, the table details the hardware type, manufacture, I/O, Coms, power requirements and number of pins.

Table H1.

| Hardware Type   | Manufacturer | Communication | Voltage   | Power | Pins |
|-----------------|--------------|---------------|-----------|-------|------|
| Microcontroller | Microchip    | 2 x UASRT     | 2v – 5.5v | TBC   | 28   |
| GPS Receiver    | Venus        | 2 x UASRT     | 3.3VDC    | 90mA  | 2    |
| LCD Display     | Fordata      | Digital       | 5VDC      | 220mA | 7    |
| SD Card Module  | TBC          | SPI           | 3.3VDC    | n/a   | 4    |
| USB Connections | On Board     | USB           | 3.3VDC    | n/a   | 3    |
| LEDS 3No.       | TBC          | Digital       | 3.3VDC    | 90mA  | 1    |
| Switches        | TBC          | Digital       | >5VDC     | n/a   | 1    |
| Buttons         | TBC          | Digital       | >5VDC     | n/a   | 1    |

Following the rational to identify the minimum number of pins required and the type of communications, the CPU speed, RAM size and program memory need to be defined, again this information has been tabulated for the purpose of the report.

## CPU Speed

To determine the required CPU speed and the memory requirement the author had to identify how much information was going to be processed by the microcontroller for a specific duration. The duration of a second was used as typically a CPU clock is measured in HZ or Clock cycles per second, therefore the selected GPS receiver maximum update rate is 20 Hz, the data output is selectable for type of string and number of strings. For the purpose of this project only one string type is required, therefore for any given GPS string the total number of Bytes per second can be calculated. Each character of a GPS string is ASCII format 8 Bit, 1 byte of information per character with the maximum number of characters in one GPS string 74 Bytes.

- Max GPS Update Rate = 20Hz
- Max length of one GPS string = 74 Characters or 74 bytes
- Therefore  $74 \times 20 = 1480$ , 1480 Bytes per second can be seen at the coms port with only one type of GPS string selected.
- Estimated number of code instructions required to read and write data from GPS to LCD = 40
- 4 clock pluses to read one instruction
- CPU Minimum speed = No. Bytes/sec x 40 instruction x 4 clock pulse per instruction
- CPU Minimum Speed =  $1480 \times 40 \times 4 = 236,800$
- Therefore, without any overhead the minimum CPU speed required is 236.8KHZ, allowing for a 10 fold overhead the minimum CPU speed specification is 2.4MHZ

## RAM

The RAM memory will be used to read and write the GPS data, therefore the estimate amount of RAM required has been calculated below.

With reference to the application of this design the RAM is based on the following, only one type of GPS sentence is required for GPS tracking mileage and an update rate of once per second fast enough. Taking this into consideration the number of Bytes seen at the COM port at any one time will not be more than 70 Bytes as discussed previously for the GPS String. An allowance of 70 Bytes each is being made for SD read/writing and for USB, therefore the total RAM required has been estimated at  $3 \times 70 = 210$  Bytes without overhead multiply by 10 = 2.1k Bytes

## Program memory

Program memory, reviewing the PIC18F24j50 datasheet page 545 the program memory is 16 kB, in the datasheet 16k has been defined as 8192 instructions; these instructions are based on assembly language therefore each instruction equals 2 Bytes. Writing a program in Assembly language requires many instructions to perform a part of a task very precise control, where an application requires precise control Assembly language would be best code to use, for this project precise control is not required therefore assembly language will not be used to write the code.

An estimate of the number of instructions to complete what is to be expected a fairly simple code is less than 1000 instructions therefore the PIC18F24j50 has significant program memory overhead, considering this factor the other PIC's in table H2 based on program memory would be an overkill.

### Pins

Number of pins required referencing table H1. Is 19 for all the hardware plus 1 for power, 2 for ground totals 21 pins. The other consideration was the type of connection; the author planned to prototype the design prior to the full build therefore all hardware if available will be selected as through hole technology, archival researching online resources microchip.com the data sheet have confirmed PDIP technology compatible with through hole mounting are available on up to 40 pin microcontrollers , considering the specification has defined a minimum requirement of 21 pins allowing for future enhancements the number of pins required is going to be 28 pins or 40 pins.

### Communication ports

As identified earlier in the report the number and type of coms ports are 1No. UART for communication between the microcontroller and the GPS Module, 1No. SPI for the SD card coms and 1 USB for updating the firmware following the completion of the device.

Through archival research the author has found that microcontrollers that are specified with a more than 1 coms ports, in reference to this proposed device SPI & UART, the microcontroller shares some of the pin allocation therefore even though the datasheet high level specification states various coms are available in reality they don't if you require to you all of them in your application. Therefore, the author has specified 2No. SPI and 2No. UART when using the microcontroller selector.

### Microcontroller Selection

To summarise the microcontroller justification, the minimum specification required for the microcontroller to be used in the proposed device is list below. This minimum specification was used to select the microcontroller as seen in table H2.

- Minimum CPU Speed – 2.4MHZ
- Minimum RAM – 2.1 k Bytes
- Minimum Program Memory – less than 2k
- PDIP Technology (28 & 40 Pin)
- 2No. UART hardware pins
- 2No. SPI Hardware pins
- USB Module

Table H2.

| Product<br>(PIC18F) | Cost<br>(\$) | Program<br>Memory<br>(Kbytes) | RAM<br>(Bytes) | PINS | Max<br>CPU<br>Speed<br>MHz | CPU<br>MHz | A/D<br>Channels | A/D<br>Resolution<br>(bits) | 8-Bit<br>timers | UART<br>SPI<br>I2C | USB |
|---------------------|--------------|-------------------------------|----------------|------|----------------------------|------------|-----------------|-----------------------------|-----------------|--------------------|-----|
| 24J50               | 1.86         | 16                            | 3800           | 28   | 48                         | 8          | 10              | 10                          | 2               | 2                  | Y   |
| 25J50               | 2.00         | 32                            | 3800           | 28   | 48                         | 8          | 10              | 10                          | 2               | 2                  | Y   |
| 26J50               | 2.28         | 64                            | 3800           | 28   | 48                         | 8          | 10              | 10                          | 2               | 2                  | Y   |
| 26J53               | 2.42         | 64                            | 3800           | 28   | 48                         | 8          | 13              | 12                          | 4               | 2                  | Y   |
| 27J53               | 2.66         | 128                           | 3800           | 28   | 48                         | 8          | 10              | 12                          | 4               | 2                  | 2   |

#### Review Microcontroller Selection

As all the above satisfied the minimum requirements of the specification as seen in table H2.

The differences in each microcontroller has been included in table H2. The first difference is the costs which is insignificant in this case but for mass production this could be one of the factors that's included in the selection. The other differences that might prove to be of significance when selecting the suitable device are program memory, Number of analogue to digital (A/D) channels, the A/D resolution and the number of 8-bit timers.

Reviewing the differences in reference to this project

#### A/D Channels & Resolution

The specification of the device has no requirement for analogue inputs therefore a comparison against number of A/D channels and the resolution is not a requirement of the final design of this device.

#### 8-Bit Timers

Without going further ahead into the project and detail design around the programming the number of timers required is unknown, however effectively with only one sensor input the GPS module data stream the author anticipates they will not require more than two 8-bit timers.

## Summary

Following comparing the microcontrollers in table H2. The microcontroller selection is narrowed down to either the PIC18F24J50 or the PIC18F26J53, referencing the datasheet the PIN outs for both these devices are almost identical see fig H2 & H3, the PIC18F26J53 does have more hardware functionality where it has more analogue comparators and also has capture and compare hardware pins for Pulse with modulation which either is not a requirement of this design.

Therefore, following the comparison of all the microcontrollers in table H2 the PIC18F24J50 is the only remaining device and has been selected as the microcontroller for this project.

Fig H2 [10] [11]

### 28-Pin SPDIP/SOIC/SSOP<sup>(1)</sup>

■ = Pins are up to 5.5V tolerant

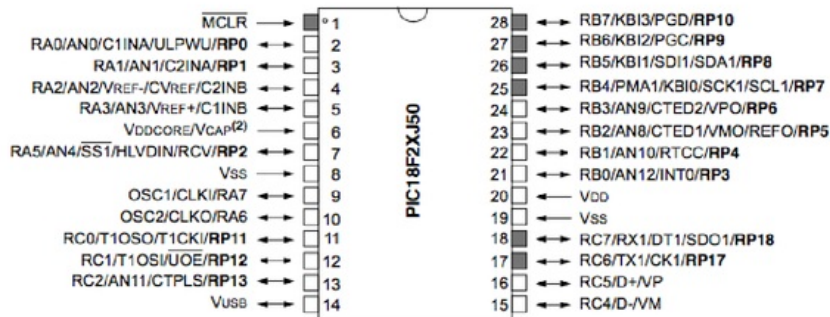
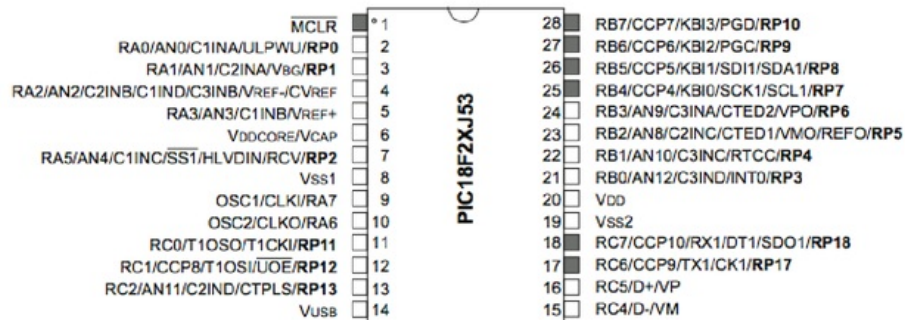


Fig H3 [12]

### 28-Pin SPDIP/SOIC/SSOP



#### PIC18F24J50

The full specification of the PIC18F24J50 has been referenced below directly from the Microchip datasheet for the PIC18F46J50 Family;

#### Features [11]

- Featuring nanoWatt XLP™ Technology ideal for battery applications Typical nanoWatt XLP™ specifications include:
  - 15nA Deep Sleep mode (Watch Webinar)
  - 60nA Sleep mode (RAM retention)
  - 830nA Real Time Clock & Calendar operation in Sleep modes
  - 780nA Watch Dog Timer operation in Sleep modes
- Other Features:
  - Integrated full speed USB 2.0
  - Deep Sleep mode ideal for battery applications
  - Peripheral Pin Select for mapping digital peripherals to various I/O for design flexibility
  - Hardware RTCC provides clock, calendar & alarm functions
  - Charge Time Measurement Unit (CTMU) supports capacitive touch screens
  - 2 Enhanced Capture / Compare / PWM modules
  - 2 MSSP serial ports for SPI or I2C™ communication
  - 2 Enhanced USART modules
  - Dual analogue comparators
  - 10 Ch., 10-bit ADC
  - Self programming Flash supports 10k erase/write cycles & 20 years' retention
  - Operating voltage 2.0 - 3.6V, 5.5V tolerant digital inputs

#### LCD Display Justification [13]

The function of the LCD display is echo data being transmitted by the GPS receiver, as a minimum the design the LCD requires a minimum of 7 Pins available on the microcontroller, the pins required standard I/O pins the allocation of the pins are for, 4 data pins, an Enable pin, RW and RS?

For the purposes of this project the LCD will have a requirement to display a minimum of the longitudinal and latitudinal coordinates of the GPS position, therefore a minimum of a dual line LCD display will be suitable to meet this requirement.

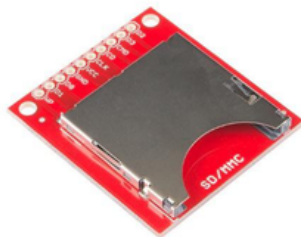
The LCD selected for this project was an Alphanumeric 20 x 4 display to meet the current minimum specification of dual line display plus allowing for future improvements, the device datasheet can be found in Appendix D.

#### Non-volatile Memory

Non-volatile memory (NVM), on power failure the data stored in the non-volatile memory is retained with reference to this project the program memory and the data captured will be required to be stored on non-volatile memory.

The data volume capacity required for the removable memory has been determined as part of the microcontroller selection, therefore the only thing that is left to determined is how is the memory hardware going to interface with the Microcontroller. Archival researching online resources and case studies of similar projects the author has found that the method of communicating between the microcontroller and an SD card is using the SPI hardware of the microcontroller connect to a SD card breakout board, see fig H4 [14].

Fig H4 [15]



### Power Requirements

The power requirements have been calculated using a crude method of totalising the power consumption of all the hardware and components that have been identified in the design.

Table H3.

| Sensor           | Supply Voltage | Load (mA)    |
|------------------|----------------|--------------|
| Micro Controller | 3.3VDC         | 20mA         |
| GPS Module       | 3.3VDC         | 90mA         |
| LCD Display      | 5VDC           | 220mA        |
| 5VDC Regulator   | 6VDC           | 100mA        |
| 3.3VDC Regulator | 5VDC           | 150mA        |
| 3No. LED's       | 3.3 – 5VDC     | 90mA (Est)   |
| 8No. Resistors   | 3.3 – 5VDC     | 200mA (Est)  |
| 10No. Capacitors | 3.3 – 5VDC     | 50mA (Est)   |
| 4.7k Pot         | 5VDC           | 100 mA (Est) |
|                  |                |              |
| Total            | N/A            | 1.02Amps     |

The battery requirements are only to provide backup power, as the device will be powered by the vehicle when in use, also the battery backup will provide power for testing purposes. Typically, the batteries will need to be small, however to get 6VDC we will need to use 4No. 1.5VDC cell batteries therefore the smaller the better to save cost and to reduce weight reviewing an online source battery capacities have been reviewed. [16]

Therefore, the following calculation has been derived to determine the battery life based on the estimate total power consumption of the device;

Battery Capacity / Device consumption x 0.7 (derating factor for external influences)

Therefore, the calculation has been based initially on AA (NiCad) 1.5VDC Cells maximum capacity.

$$1000\text{mAh} / 1020\text{mA} \times 0.7 = 0.69 \text{ hrs}$$

Reviewing the calculation, the batteries should last a maximum of 40 – 45 minutes, for the purpose of this design this is enough power capacity however if these figures are accurate for future development power savings would have to be part of the development.

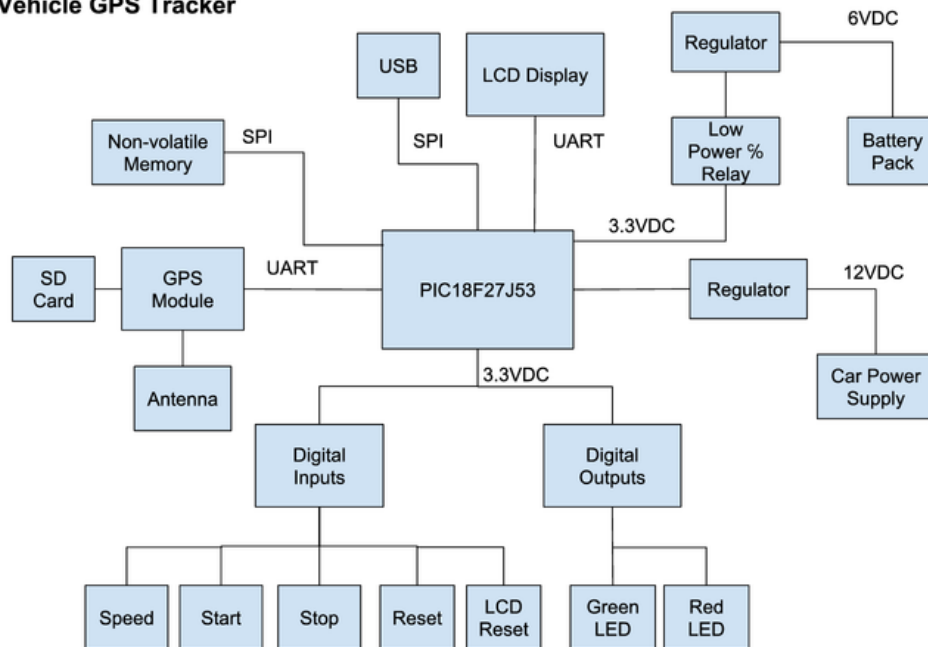
## Design deliverables

The structure diagram for the proposed project has been defined using Google draw, the tool has been used to draw boxes that defined each part of the design structure, the microcontroller is the main part of the design therefore it takes centre stage in the middle the diagram, with all the other components interconnected to microcontroller. The structure diagram is useful to present the hardware requirements in a quick reference format that is user friendly.

## Structure drawing

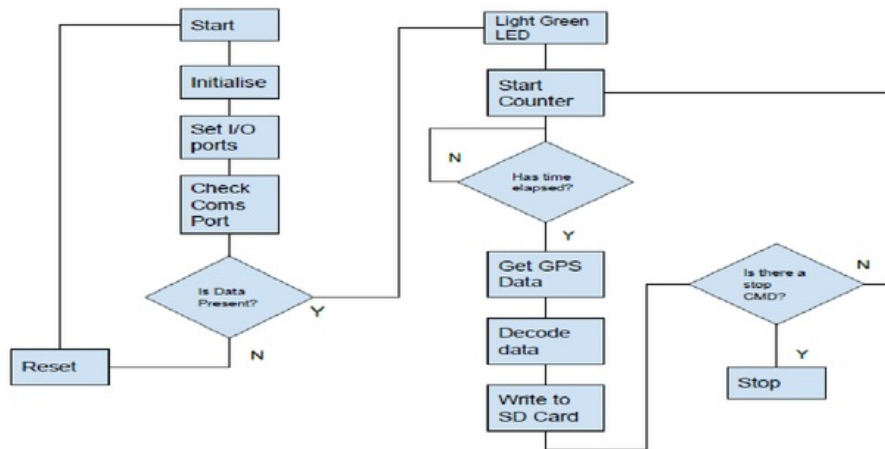
Fig H3.

### Vehicle GPS Tracker



## Flowchart

Fig H4.



## Pseudo Code

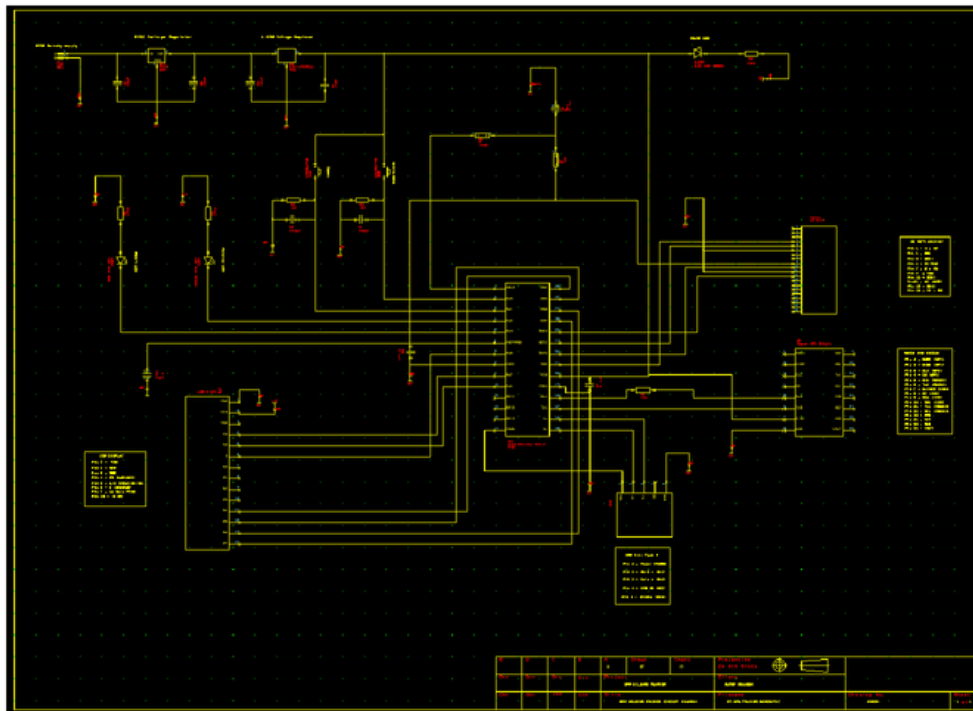
- Device power up sequence
- Setup input/outs & ports
- External interrupt Start/stop Button pressed
- if external interrupt is not true wait until true
- Interrupt is true go to function main
- Main program loop
- LCD to display start message
- Program running indication LED (GREEN) ON, flashes when program completes loop.
- Wait 5 seconds for data at coms port
- Return to wait if no data at coms port
- Read data at coms port
- Format string
- Print string to LCD
- Data receiving LED (RED) Flashing
- Write data to SD card
- Return Main function

### Schematic design

The schematic has been drawn using two CAD packages, Proteus so the design can be simulated using the VSM tool and CCS Compiler, also Easy PC has been used so the schematic can be transferred into a PCB. The Proteus software also has a PCB design tool however the Easy PC PCB tool has more functionality and it is more user-friendly.

The circuit design seen in the CAD screenshot Fig SD1 has been derived by selecting the components previously identified within the design section of this project from the software library where available. When components have not existed in the Library the wizard has been used to design a new component, even though it is a wizard it does require having knowledge of the component that is not available in the software library, knowledge is gained by experience using different components all in the authors case by Reading the datasheet. Reviewing the hardware datasheets, the connection requirements of the components were derived and then were translated into the schematic. Following completion of interconnecting the hardware components the schematic has been labelled accordingly and dropped into a standard A2 CAD drawing template.

Fig SD1.



#### Schematic description

Starting with the Power supply requirements, referencing the microcontroller data sheet for the PIC18F24J50 its operating voltage range is 2.2 – 3.6VDC so the power supply to the microcontroller has been regulated to 3.3VDC using a TS1084A voltage regulator from TS, datasheet seen in Appendix D. The regulator must have 10uf capacitors connected at both input and output pins down to ground. [17]

Reviewing the microcontroller datasheet for the PIC18F46J50 Family Page number 29 and figure 2-1 [11] the recommended minimum connections are found, these are the minimum pins required to operate the microcontroller, the pin allocations are listed below which are connected using various RCs.

- VDD/Vss - referencing in the data sheet a 0.1uf 20v Ceramic capacitor is required in parallel between VDD and VSS. The purpose of the capacitors across every pair of Vdd and Vss pins is for decoupling
- VCAP – reference in the datasheet F type PIC's a capacitor is required to connect to the VCAP/VDDCORE pin to stabilize the voltage regulator output voltage, the capacitor must be 10 uf connected to ground it must be either the ceramic or tantalum capacitor, see page 429 PIC18F24j50 datasheet. [11]
- MCLR - referencing to the datasheet the MCLR pin requires the connection of a resistor in series and parallel, and then a capacitor in series with the parallel resistor. The series resistor can be between 100–470 ohms and the parallel resistor must be 10 K ohms, finally the capacitor must be 0.1uf. For programing and debugging its recommended the capacitor is isolated this could be something as simple as a jumper. See fig 2-2 from datasheet page 30 [11]

#### Hardware connections

Researching relevant datasheets and the 'The Art of Electronics' page 1080 figure 15.20 [18] connecting devices to a microcontroller, the method of connecting the hardware was found as detailed below

- The USB Connector,
- The MMC Card,
- The LED pins – Page 132 data sheet port B & C are designed for high loads such as LEDs!
- The Pushbutton pins – art of electronics page 802 figure 12.13 (driving digital logic inputs) [19] **citation** 10k resistor in parallel with a 100 – 1000pf capacitor to suppress capacitive coupled noise.
- The LCD pins are connected directly to the input pins of the microcontroller, as a minimum requirement reviewing the datasheet the microcontroller will require 7 free pins to connect to the following LCD pins, Enable, Read/write, address and 4 data pins, the LCD also needs to be connect to power and ground.

*PIC18F24J50 Datasheet page 33 - 2.7 Unused I/Os*

*Unused I/O pins should be configured as outputs and driven to a logic low state.*

*Alternatively, connect a 1 k $\Omega$  to 10 k $\Omega$  resistor to VSS on unused pins and drive the output to logic low. [11]*

Reviewing the Venus GPS datasheet [20], the connection details for the GPS module can be found see Appendix D, for the purpose of this design the only pins are being used are the RS-232, Power and ground pins.

## PCB Design

The PCB has been designed using the schematic to PCB transfer tool in the easy PC software package, after the schematic design drawing was completed the document was saved, then using the design rules check located within the tools tab the design was checked to ensure it could be transferred to PCB. Following carrying out the design rules check a report is generated listing any errors, no errors were found on this occasion therefore the translate to PCB option was selected and a new file was created for the PCB design linked to the schematic.

The purpose of the PCB design is to arrange all the hardware and components as close together as possible within the design parameters to ensure the PCB is as small as possible to reduce costs of producing the PCB.

Within the PCB design window, the components were transferred from the schematic design will be found in the component bin, each component can then be selected from the bin and dragged onto the PCB design drawing when all the components are on the drawing they can then arranged. The arrangement of the PCB components can be done automatically when all the components are in the design window you really function in the xx tab, for the purpose of this design the auto routing function has been used to save time.

**Due to time constraints around the build the PCB wasn't develop therefore no completed PCB design is available to be presented.**

## Software design

Software writing is a new found skill for the author therefore to improve the authors competence the code described below was written and simulated on a trial and error basis, with significant time spent researching how they can be implemented.

The software design required a number of design functions in order to produce the final software design for the device, the first part of the software design used is the structure drawing and flowchart produced earlier as part of the design pack. The structure drawing and flowchart have been used to design the high level code Pseudo code (pronounced as seudo code), the Pseudo code is a method of writing software code in plain English so the software developer understands the indented function of the program that has to been written.

As discussed throughout this report the Vehicle Mileage Tracker is a fairly simple design that does not require complex software routines, referencing the earlier research a high level programing language was seen to be suitable for this device, therefore the programing language C was used. The code for this device was C and the basic code was derived using CCS complier, the CCS wizard was used in conjunction with the Pseudo code to specify the basic code and then it was developed writing functions within the code that were relevant to the operation of the device.

## CCS Compiler

To write the basic C code the CCS Wizard was run, to run the wizard the author opened the CCS software and selected new within the File tab and then selected Project Wizard see Fig C1. The following steps were then taken to produce the final wizard code;

1. The project wizard required a place to file the project, the project folder created earlier for the Proteus design was used to save the project wizard file called main in a sub folder called CCS, then the project wizard was launched see Fig C1 & 2 below.

Fig C1.

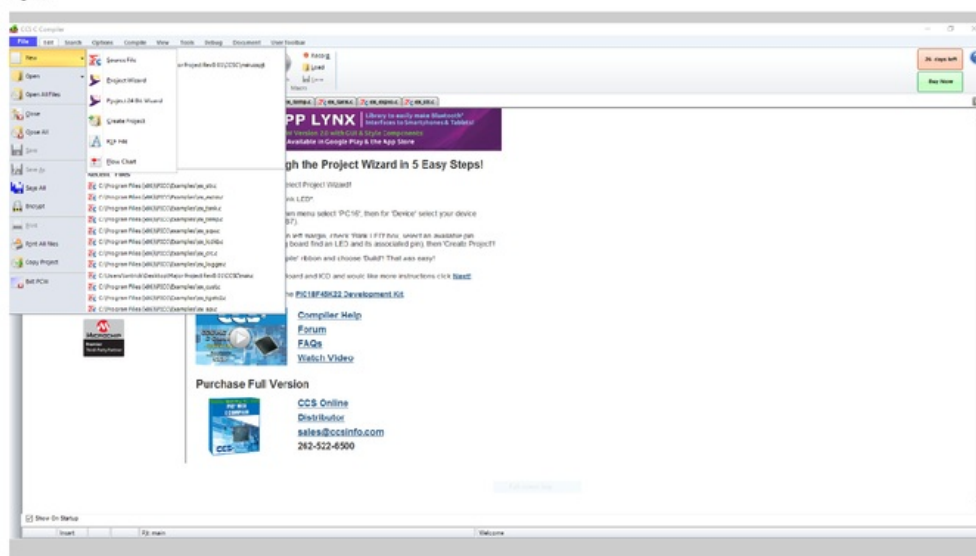
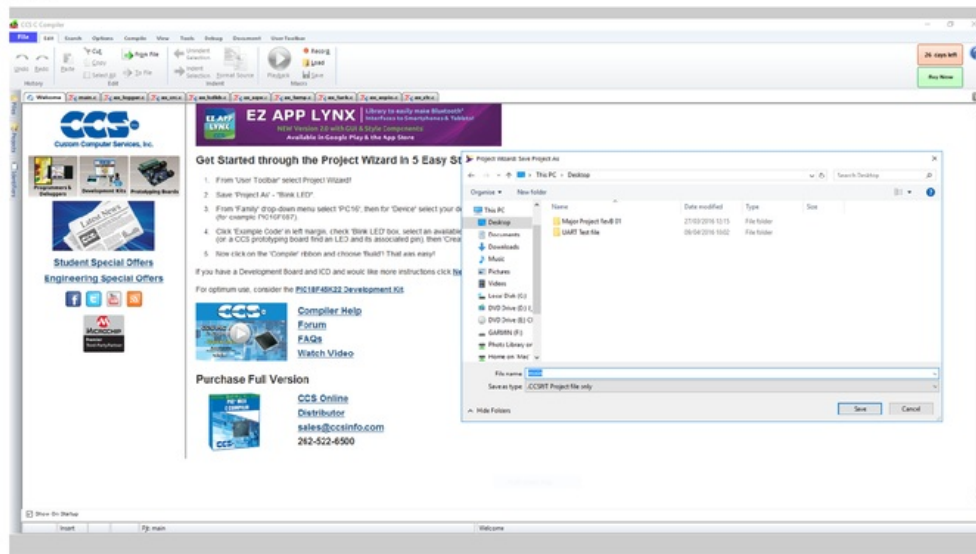
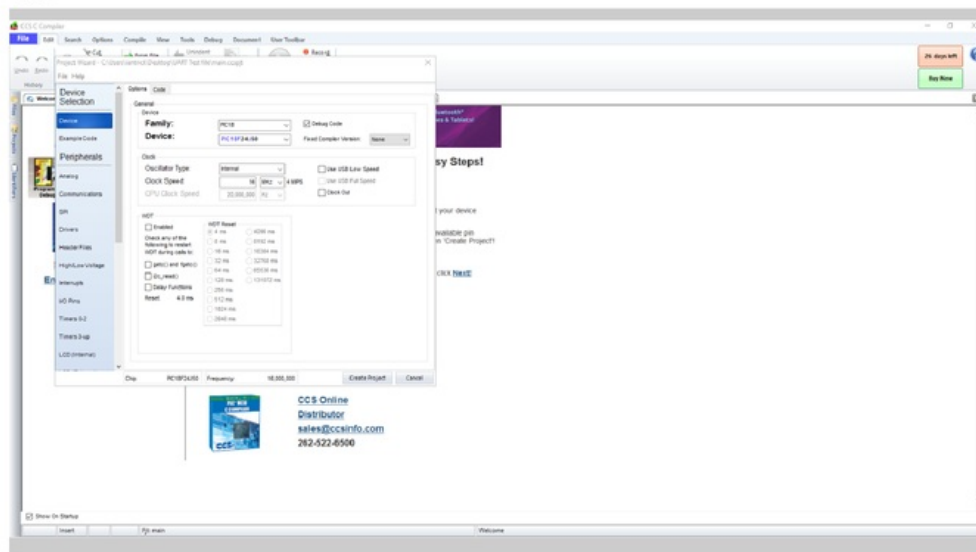


Fig C2.

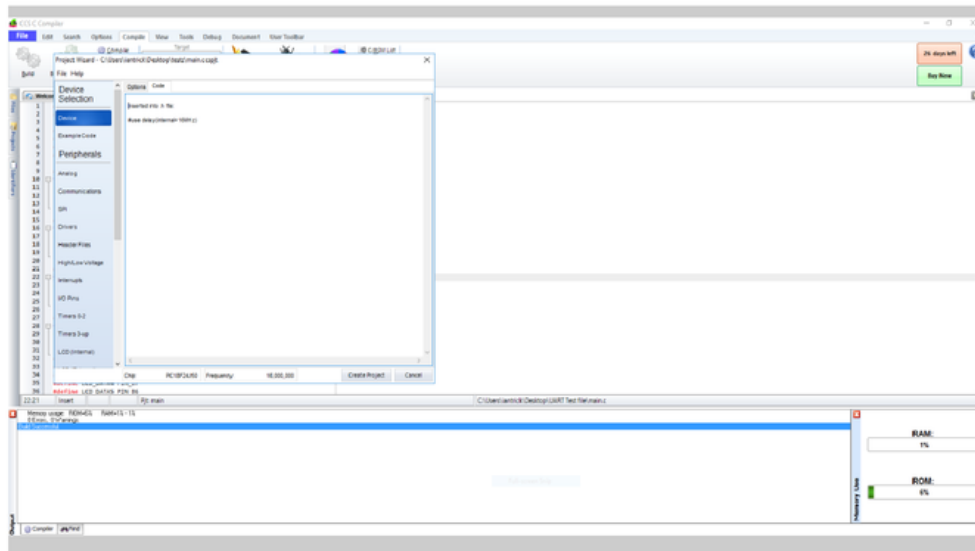


2. The Project Wizard has a tab called 'Device selection' where the microcontroller relevant to this project was selected, the Project Wizard will not let you proceed until you have selected your microcontroller, see Fig C3.

Fig C3.



3. Within the device tab the microcontroller selection and setup is under the sub-tab called options, the other sub-tab is called code, this section produces a C code based on the selection in the options sub-tab, see code Fig C4.



4. The next section under the device selection tab is example code see Fig C5, for the purpose of this project the author has selected a flashing LED on Pin A2 with a delay of 1000ms, the code can be seen for this function by selecting the code tab, see fig C6.

Fig C5.

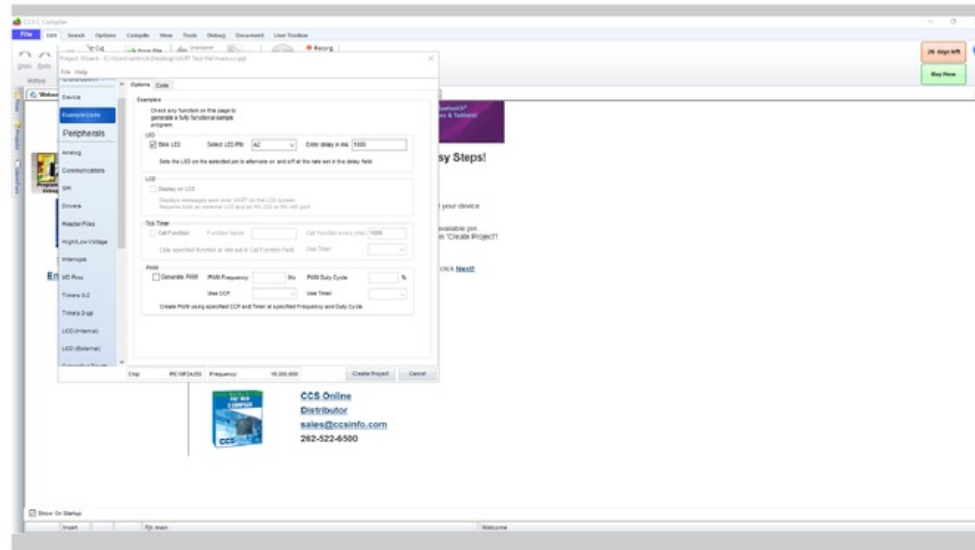
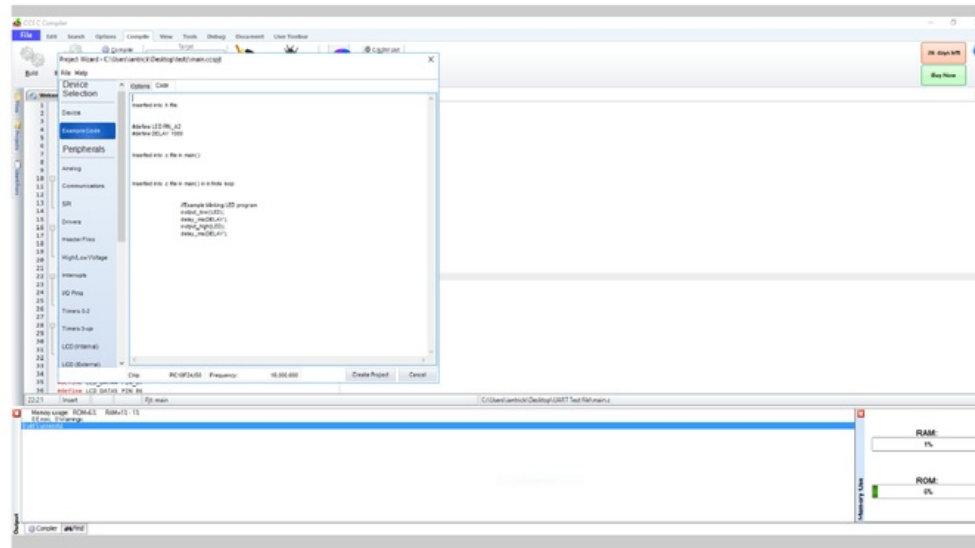


Fig C6.



5. The next section relevant to this project is the 'Peripherals Tab' where for the purpose of testing the code an Analogue Peripheral was selected PIN C2 was setup as an analogue input pin and the analogue to digital resolution was set to 8-bit, see fig C7 and C8 for the code.

Fig C7.

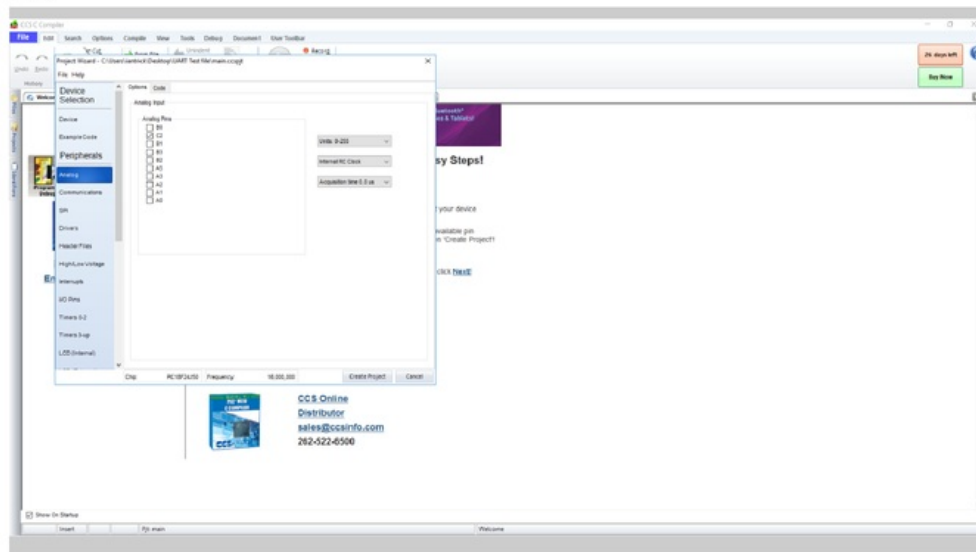
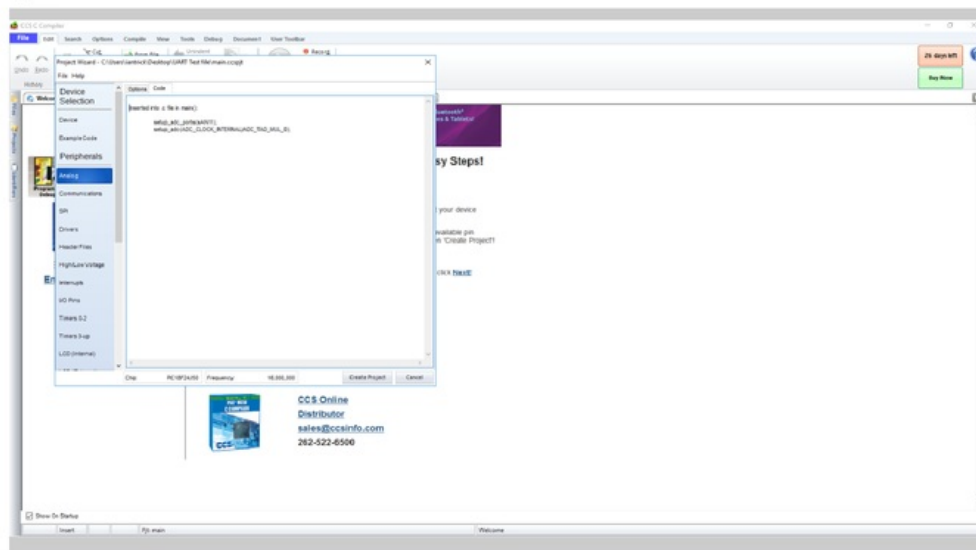


Fig C8.



6. Communications was the next Peripheral selected, with the hardware RS232 being setup for communications between the microcontroller and the GPS module, see fig C9 and Fig C10.

**Fig C9.**

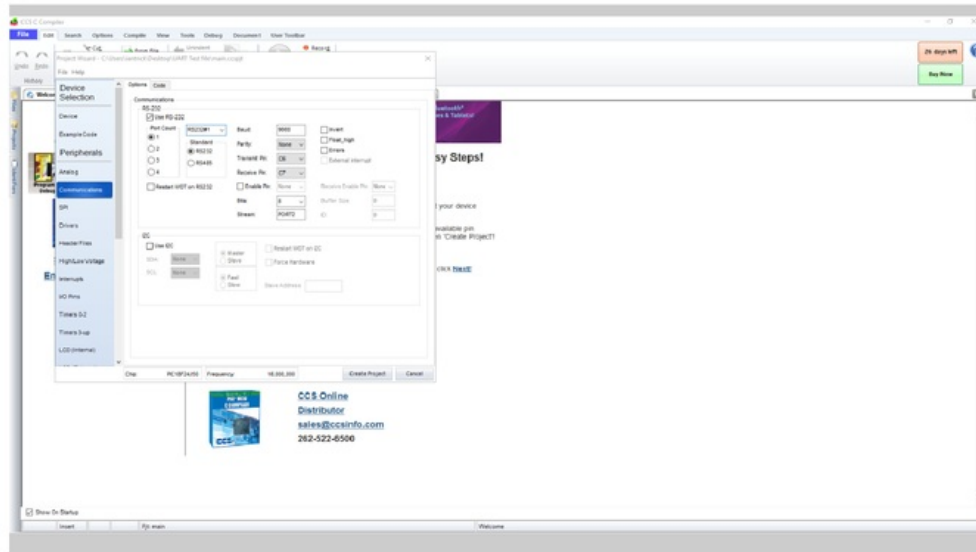
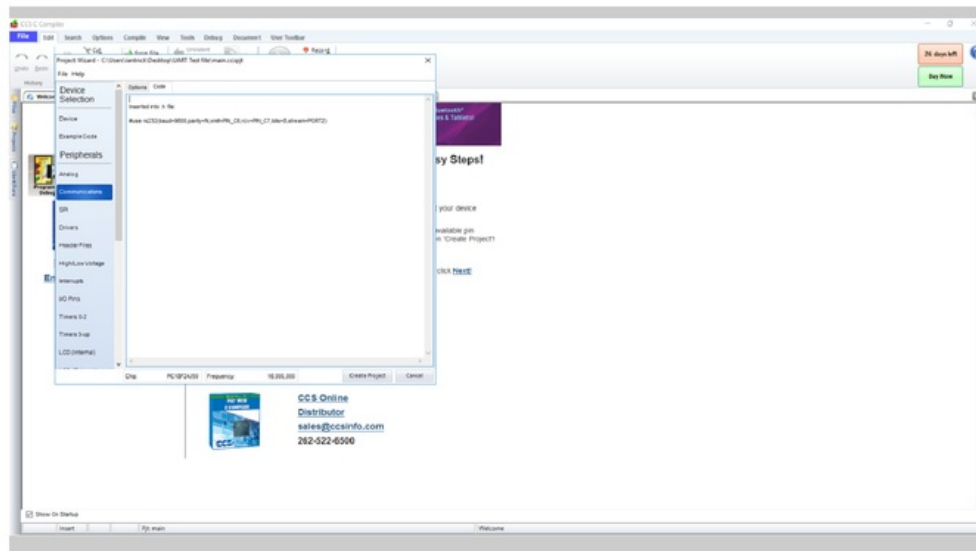


**Fig C10.**



7. SPI was next peripheral setup using the remap-able pins B0 – B3, the SPI port is used for communication between the microcontroller and the memory card module, see fig C11 & C12.

Fig C11.

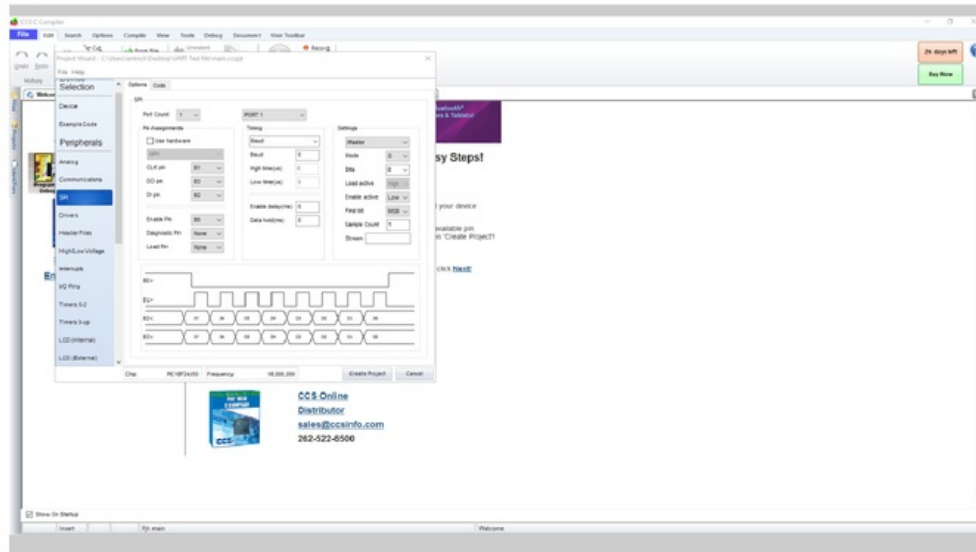
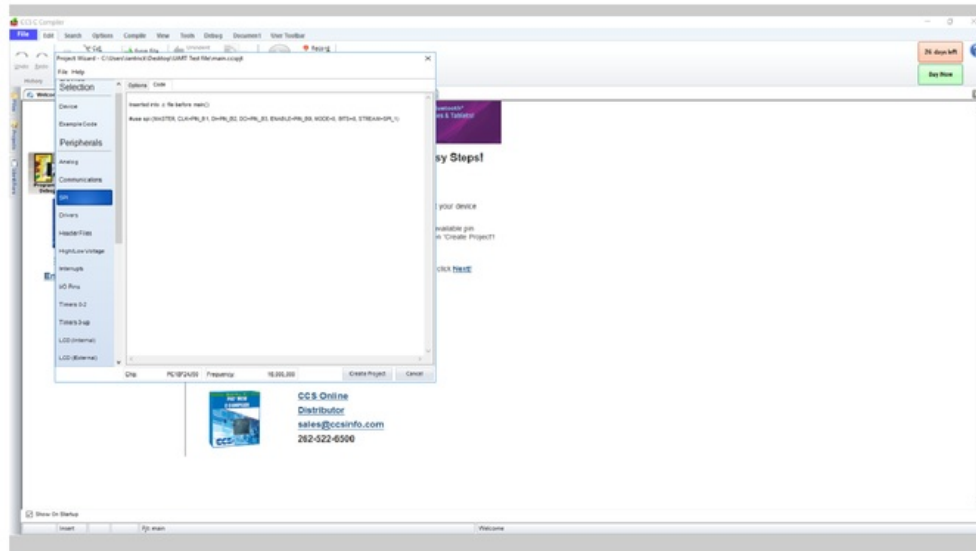


Fig C12.





9. Next the program interrupt routines were added, so functions can operate in the background without effecting the main program. The following interrupts selected Timer 2, RS232 Receive, RS232 Transmit and SPI activity, See Fig C15 and C16.

Fig C15.

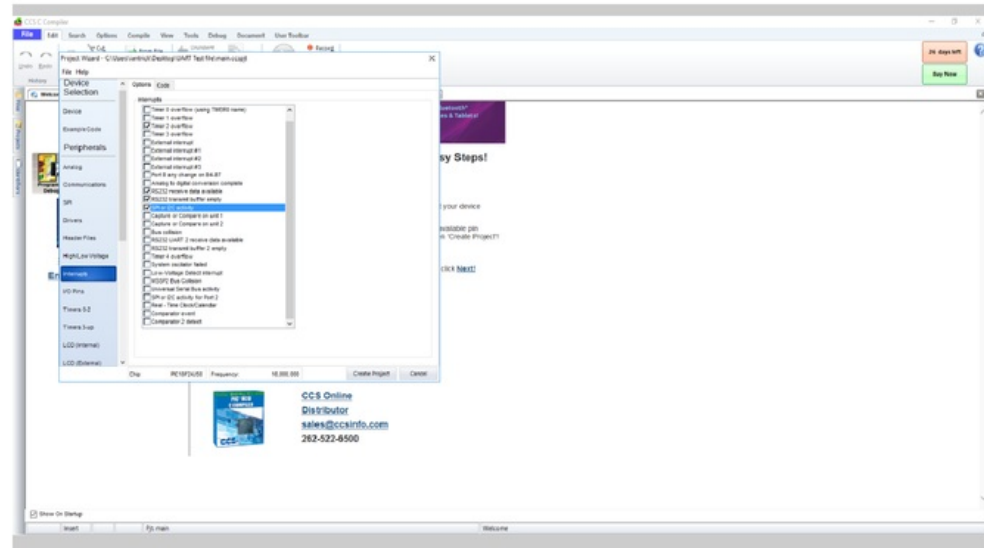
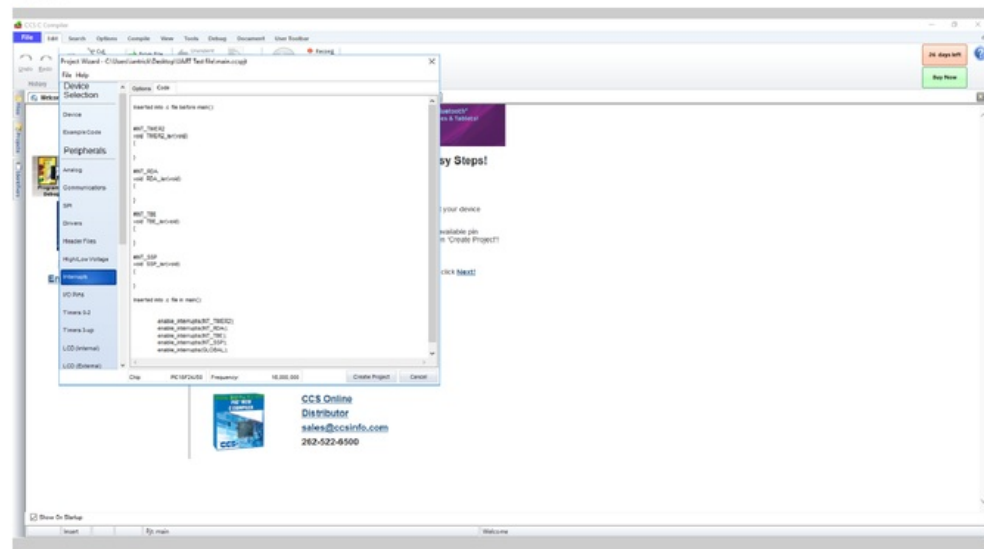


Fig C16.



10. The next peripheral required was a timer function, timer 2 was selected to get the delay time required this can be changed later if required, see Fig C17 and C18.

Fig C17.

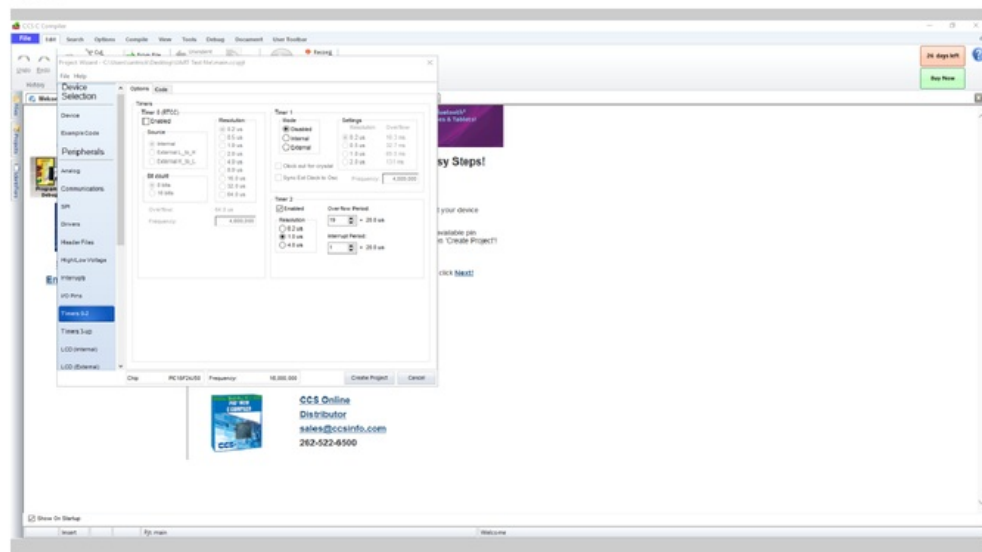
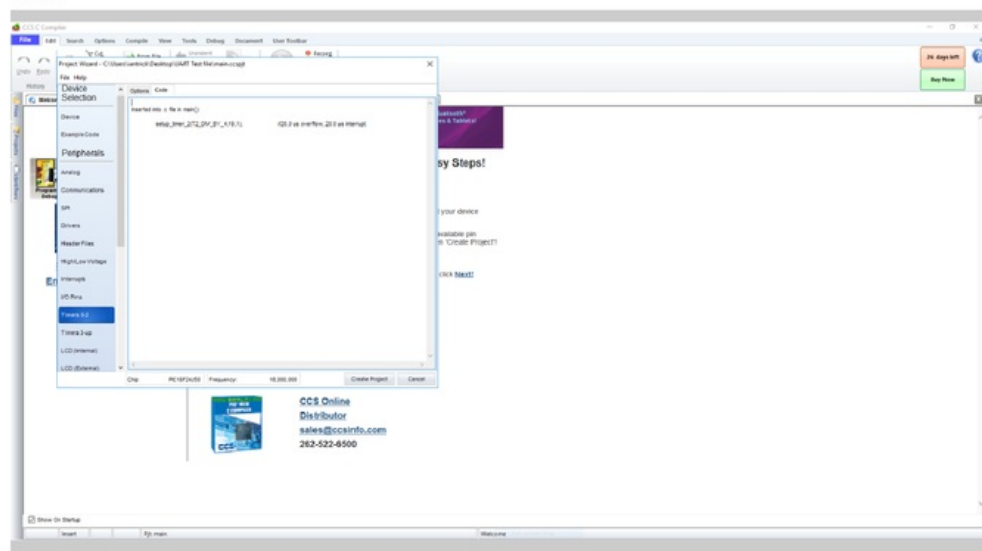


Fig C18.



11. The LCD setup was required next; the USE LCD box was selected then the drop down boxes were used selecting the relevant pin allocation to each function of the LCD by referencing the schematic in the design section of this brief, see Fig C19 and C20.

Fig C19.

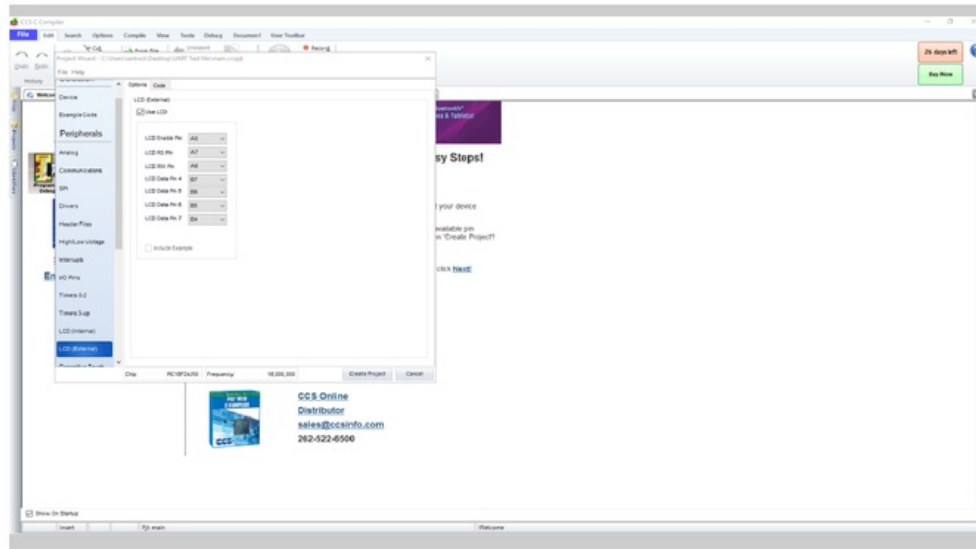
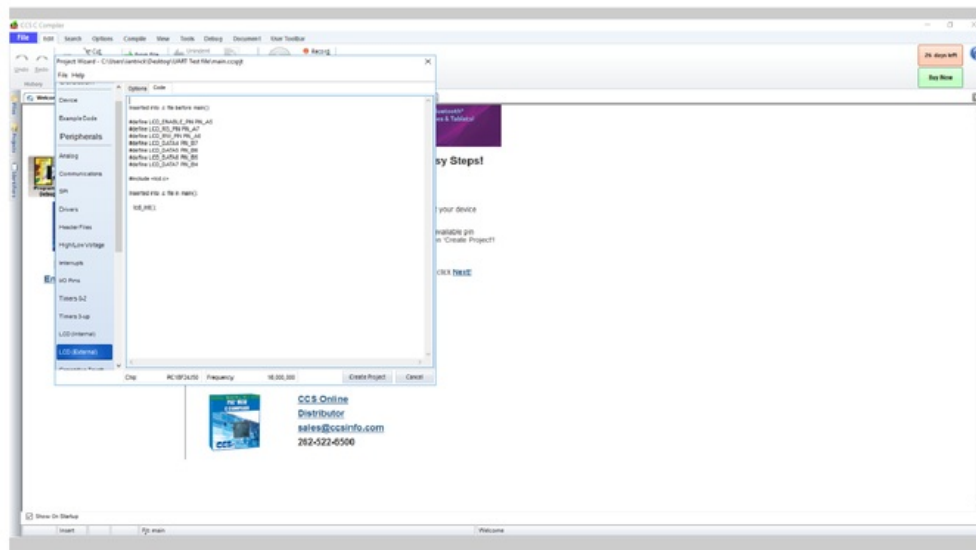


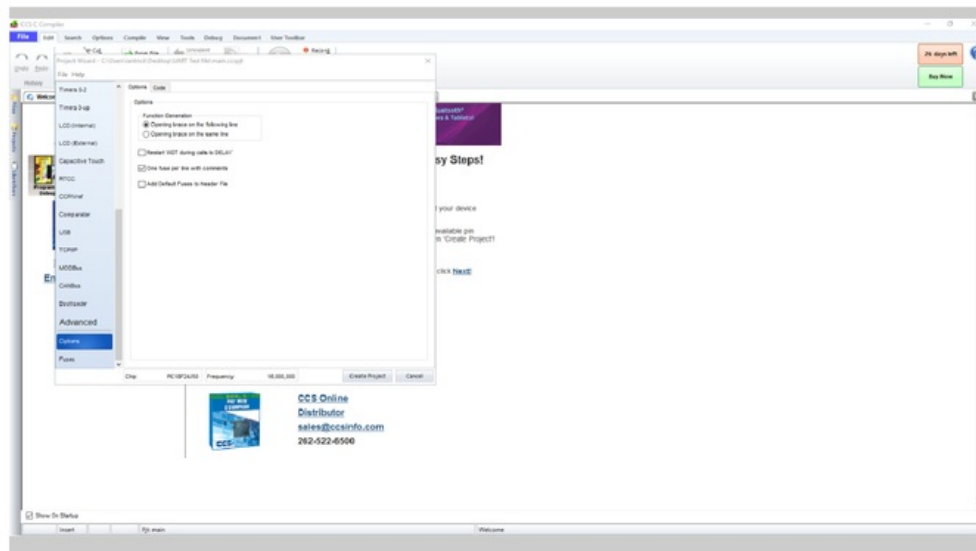
Fig C20.



12. NO other peripherals were required so the next stage was the Advanced tab.

13. Under options in the Advanced tab, the Add default fuses to header files box was selected, see Fig C21.

Fig C21.



14. Finally, the fuses were selected that were relevant to the project see Fig C22 & C23.

Fig C22.

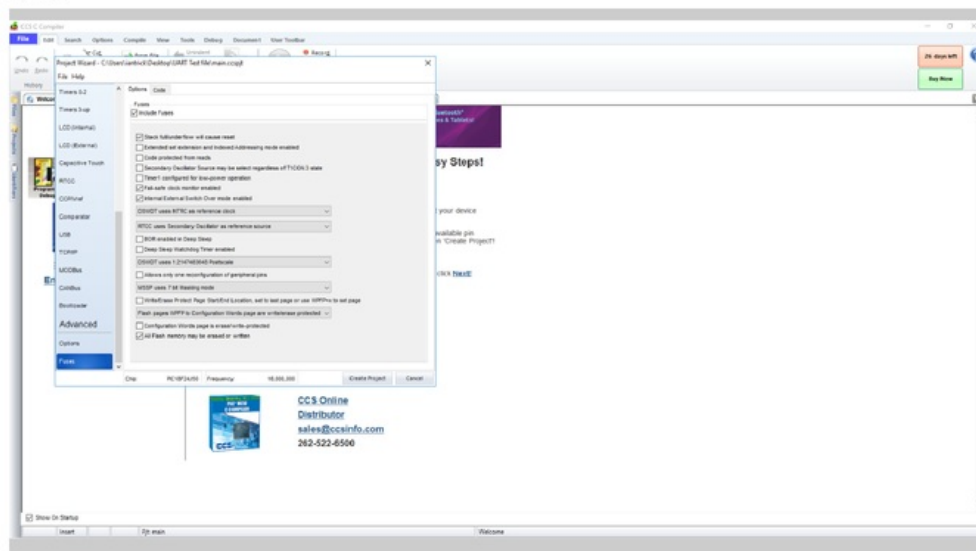
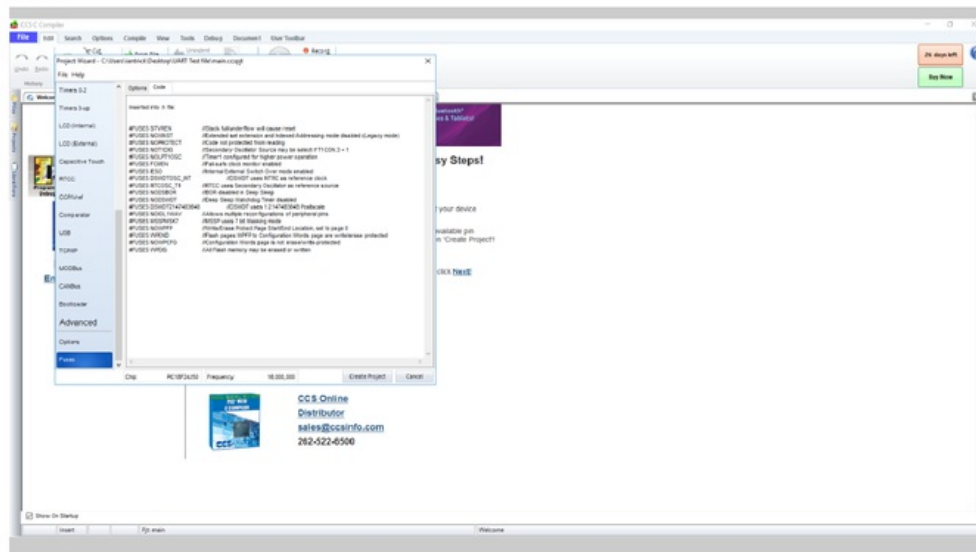
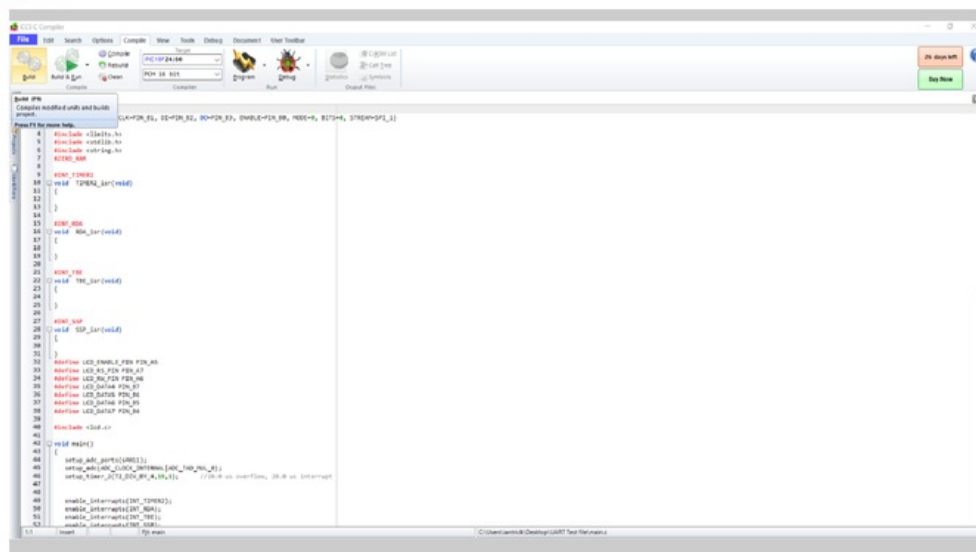


Fig C23.



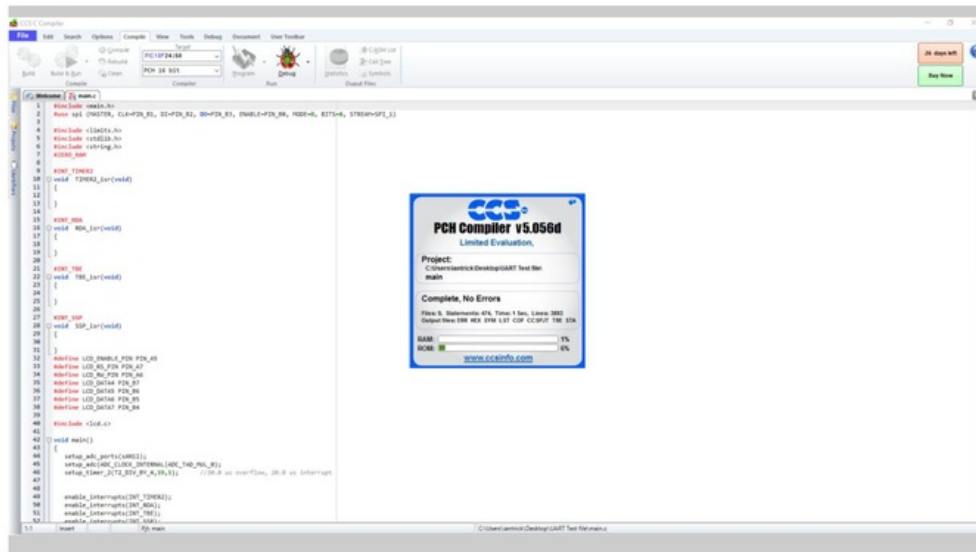
15. Within the the fuses tab create project was selected and the code was generated based on the selections made in the wizard, see Fig C24.

Fig C24.



16. The code was then compiled see fig C25.

Fig C25.



17. The following files associated with the code were found in the file tab on the left hand side of the compiler, each file was copied from its source location to the project folder within the sub-file called Proteus, see fig C26 and C27. The file names were main.c, main.h, 18f24j50.h, ctype.h, limits.h, stddef.h, stdlib.h, string.h and lcd.c.

Fig C26.

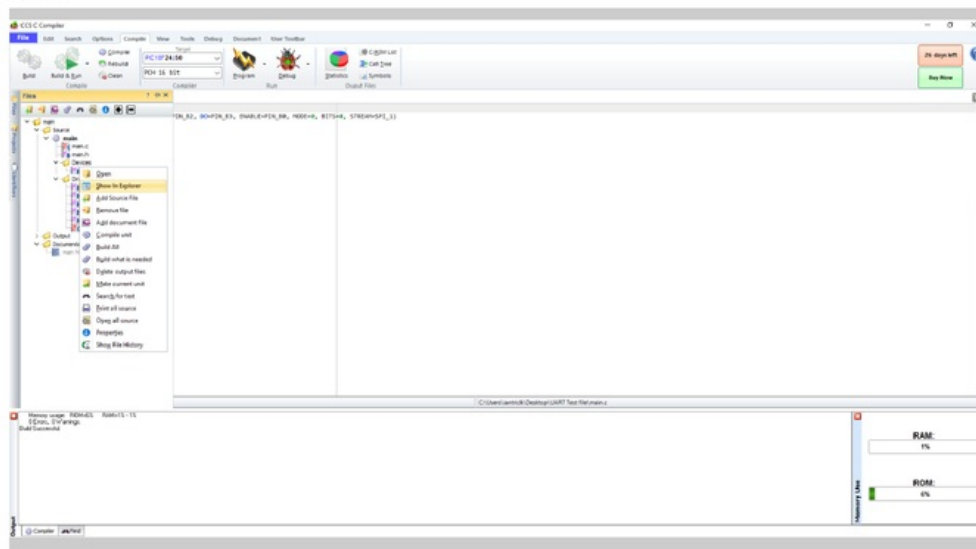
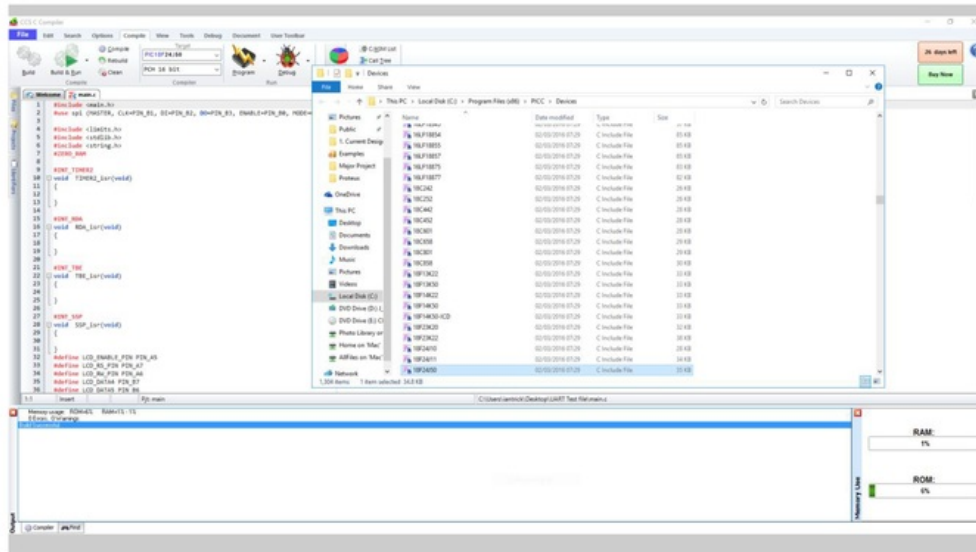


Fig C27.



Proteus VSM

Following completion of the wizard code, the code was loaded into Proteus and then developed to meet the specification of the vehicle mileage tracker. The code was again derived by referencing using trial and error methods.

The functions of the code have been extracted for the program and explained below;

***unsigned int value,q***

***char c;***

***char k;***

Integer and characters have to be assigned so they can be used throughout the program as function, if they are not assigned you will receive an error when complying the program. The integer has been given a value of q so it can be used as a counter and the characters have been defined as c and k to represent characters of a serial data stream.

***lcd\_putc("\f SkyTraq, Venus6");***

***delay\_ms(2000);***

***lcd\_putc("\n\r Ver 011023");***

***delay\_ms(2000);***

***lcd\_putc("\f GPS Mileage");***

***delay\_ms(2000);***

***lcd\_putc("\n\r Tracker");***

***delay\_ms(2000);***

Lcd\_putc is used to put the information defined in the brackets onto the LCD display, the \f clears the display and puts the cursor back to the first position of the LCD display, therefore "Skytraq, Venus 6" will be the message displayed on the LCD. The function repeats its self with different message being displayed each time, the \n\r is the new line carriage return function which takes the cursor to the start of the second line of the LCD display.

```

        if(input (PIN_A0))
        {
            output_low(LED1); //Green LED OFF
            delay_ms(250);
            output_high(LED1); //GREEN LED ON Main prgram initialised, LED will Flash
when the program starts to cycle
            delay_ms(250);

```

The if function is waiting for the PIN\_A0 to be true, pin A0 is the start pin. Following the latching start button being pressed the Green program running LED will go low (OFF) then after a delay it will go high and go to the next function, when the whole program has been executed the LED loop will start again so the LED will flash off and on while the program is running.

Explaining the purpose of the next function first before looking at the code may make it easier to understand the reasoning. The data being received at the microcontroller Rx pin is a 70 byte GPS string at a rate of 1HZ, the purpose of this function is to dump the data that is not required and display the data what is required in this instances it's the latitude and the longitude data and their directions. Therefore, we know the type of data stream being received at the Rx and we know the format, so the function basically will count in the bytes then dump after it meets the counter setting, then counts and display the Latitude then the same for the longitude.

```

k=fgetc(port1); //Get data from Rx pin of RS232 port
if(k)          // True if character is at Rx Pin
    if(k=='$')  //Start reading all bytes following seeing $
    {
        for(q=0;q<19;q++) //Add 1 to q until the count has reached 19
        {
            getch(); //Get Characters 0 - 19 from Rx pin and clear

```

That completes the function to count the data that's not required then it dumps it after its reached its count, see below the code is similar however because the function is already talking to the Rx the code follows on reading the string, but the first function is to tell the cursor to go to an exact position on the LCD display, then using `lcd_putc` combined with `printf` because we are asking the device to print the Rx information on the LCD display after the word "LAT - ". `getch()`; clears anything left on the LCD after LAT -

The for function then counts the next 12 characters then prints to LCD display using the `Lcd_putc` function. The code repeats its self for longitude and drops into the loop when the function has fully executed.

```
lcd_gotoxy(1,1);    //Put Curosr at LCD position 1, Line 1
printf(lcd_putc,"LAT-"); // Display "LAT - " on LCD display

getch();
for(q=0;q<12;q++)    // Add 1 to q until the count has reached 12
{
    k=fgetc(port1);    //Get Character from Rx pin port 1
    lcd_putc(k);    //Display characters from Byte count 0- 12 on LCD
}
```

The final bit of code used to confirm the microcontroller is seeing the data is the red LED that flashes each time the program loop has complete following immediately after the GPS data loop has completed.

```
output_low(LED2); //RED LED OFF
delay_ms(250);
output_high(LED2); // RED LED ON, AND THEN FLASHING WHEN DATA IS
STREAMING
delay_ms(250);
```

The code below has not used but could be utilised for future development if you want to calculate the value of the checksum and compare it against the checksum of the GPS string.

C Code for checking the GPS string, where the checksum is the bitwise exclusive xor of the all the string sentence between \$ and \* XOR truth tale below

| Input 1 | Input 2 | Output |
|---------|---------|--------|
| 0       | 0       | 0      |
| 0       | 1       | 1      |
| 1       | 0       | 1      |
| 1       | 1       | 0      |

[21] In C#:

```
int checksum = 0;  
for (inti = 0; i < stringToCalculateTheChecksumOver.length; i++)  
{  
checksum ^= Convert.ToByte(sentence[i]);  
}
```

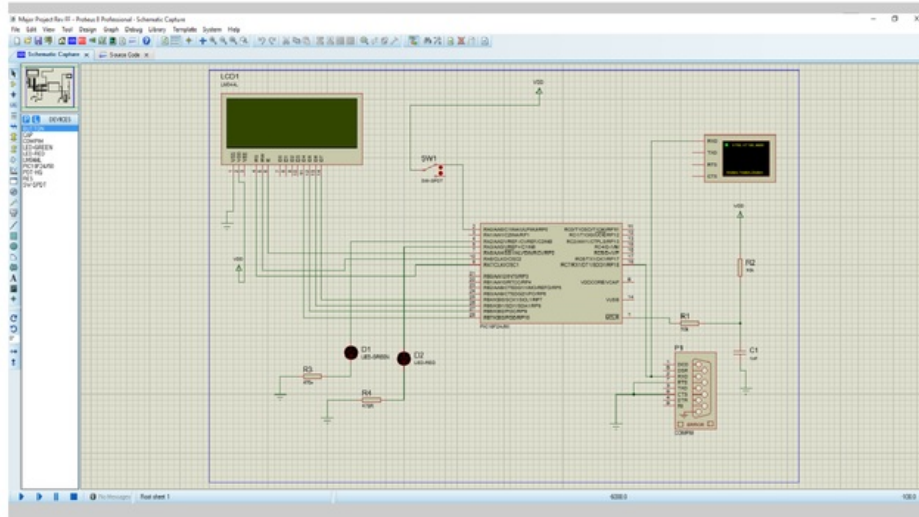
Proteus VSM Simulation error messages

Linker file not found, this message is received because CCS Complier doesn't support linker files therefore every time Microchip update a piece of hardware, the CCS complier is out of date and flags up an error, the way to resolve this would be to update CCS each time you receive the error message however the upgrade is chargeable reference [22]

## Software testing

To carry out a software test the report author has had to derive a method of simulating the hardware to test the software functionality, this was carried out by using Proteus VSM that allows the user to interact with active components within the Proteus schematic, see fig ST1.

Fig ST1.



The active components used in the Proteus VSM software were the Communication port COMPM that was connected to the PC's USB Coms port, the virtual terminal which was connected to the Rx pin of the com port to see data on the pin. The LCD Display, 2No. LEDs a green LED to confirm the program has started and a RED LED to confirm data has been received. Finally, a latching on/off switch was put into the design to start/stop the program.

Initially the Tx pin of the virtual terminal was also connected to the coms port so keyboard hits could be seen on the virtual terminal and being displayed on the LCD. The keyboard hit was being used to prove the software was functioning correctly without requiring additional hardware.

## Costings

The estimated cost that were included in Table F1 in the feasibility section of this brief have been compared with the actual costs seen in the table below,

### Hardware cost

The following hardware was purchased online and at a local electronics store, where possible spare components were purchased to mitigate against loss or damage or both. Due to time constraints the remaining hardware had to be purchased locally which increased the cost in respect the estimate costs for some components.

Table C1.

| Hardware                     | Estimated cost | Actual Cost                  |
|------------------------------|----------------|------------------------------|
| GPS Module with Antenna      | £30.00         | £45.70                       |
| PIC18f24J50                  | £2.50          | £10.40 (2 plus handling fee) |
| PCB                          | £15.00         | Not purchased                |
| LCD                          | £15.00         | Free issue from University   |
| 1 GB SD Card & Holder        | £10.00         | Not purchased                |
| 3.3V Voltage Regulator       | £2.00          | £1.80                        |
| Resistors (7 various sizes)  | £4.50          | £3.46 (10 resistors)         |
| Capacitors (8 various sizes) | £5.00          | £7.31 (10 Capacitors)        |
| LEDS (2 )                    | £1.00          | £1.50                        |
| Switches (2)                 | £3.00          | £4.73                        |
| Fuses & holder (1)           | £3.00          | £2.74 (10 fuses)             |
| Battery Holder               | £3.00          | £1.70                        |
| Total                        | £95.00         | £79.34                       |

## Implementation

### Purchase hardware

Due to longer than expect lead times some of the hardware, an alternative supplier had to be identified by the document author, hardware delivery delays would have had significant impact on the completion date of this project build.

The bill of materials can be extracted from the software tool as seen below;

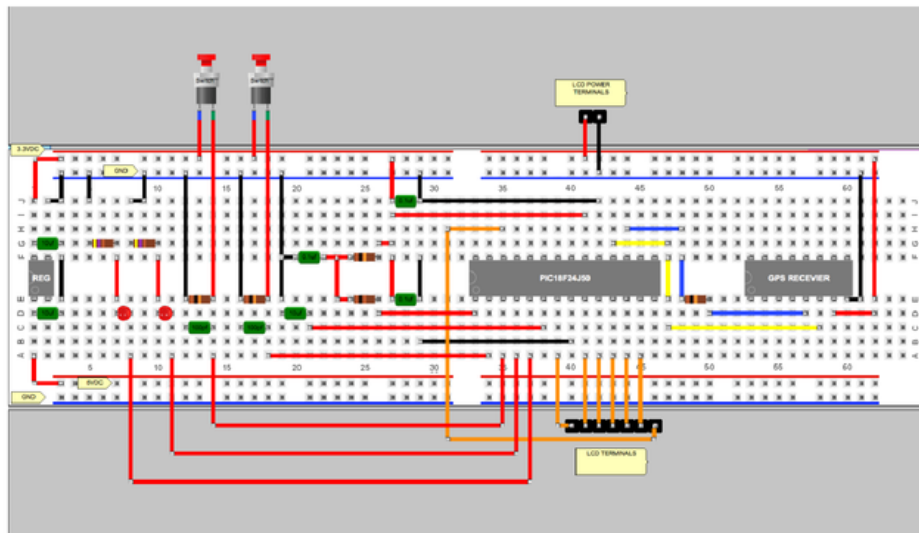
| Hardware              | Supplier  | Manufacturer | No.         | Unit cost                          | Total cost |
|-----------------------|-----------|--------------|-------------|------------------------------------|------------|
| Project Box           | Maplin    | Unknown      | 1           | £6.79                              | £6.79      |
| GPS Module            | Proto-PIC | Venus        | 1           | £36.00                             | £36.00     |
| GPS Antenna           | Proto-PIC | Venus        | 1           | £9.70                              | £9.70      |
| Microcontroller       | Microchip | Microchip    | 2 (1 Spare) | £2.43 plus carriage & handling fee | £10.48     |
| LCD Display           | RS Comp   | Winstar      | 1           | £9.97                              | £11.48     |
| USB Con               | n/a       | n/a          | n/a         | n/a                                | n/a        |
| MMC/SD Board          | n/a       | n/a          | n/a         | n/a                                | n/a        |
| Voltage Reg           | Maplin    | TS           | 1           | £2.49                              | £2.49      |
| Voltage Reg           | UWTSD     | Unknown      |             | Free Issue                         | Free Issue |
| Battery case          | Maplin    | Unknown      | 1           | £1.59                              | £1.59      |
| 4.7k Pot              | UWTSD     | Unknown      | 1           | Free Issue                         | Free Issue |
| Components & Switches | UWTSD     | Various      | N/A         | Free Issue                         | Free Issue |
| Total                 | n/a       | n/a          | n/a         | n/a                                | £78.53     |

## Build

At the time of writing this report the author had identified there wasn't enough time to complete the project as detailed earlier in the design section of this brief, therefore to complete the project the author accepted the project was going to build as a prototype and tested on a bench top using a breadboard in place of a printed circuit board. The other parts of the design not to be developed further were the SD Card and the USB functionality.

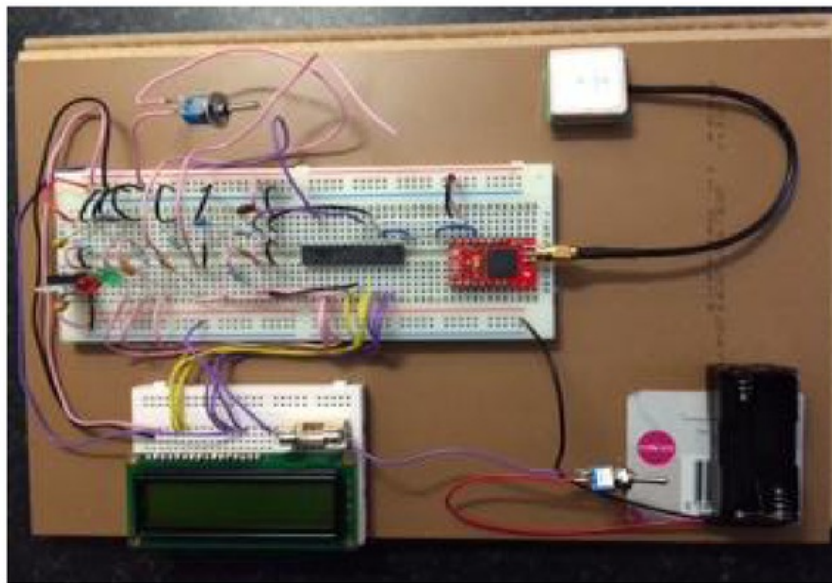
The Author used a breadboard design tool Pebble to derived the circuit layout, the tool used is available online from PICAXE [23], the purpose of this tool is the provide a circuit layout that can be used to build project efficiently, the breadboard design can be seen in figure I1.

Fig I1



The components that could be directly pushed into the breadboard were positioned as per the breadboard design, the GPS Module and LCD Display were through hole mounting therefore header pins were soldered to these devices and then positioning into the breadboard, following completing the soldering of the header pins to the hardware these components were also pushed into the breadboard to complete the circuit as seen in Fig 12.

Fig 12



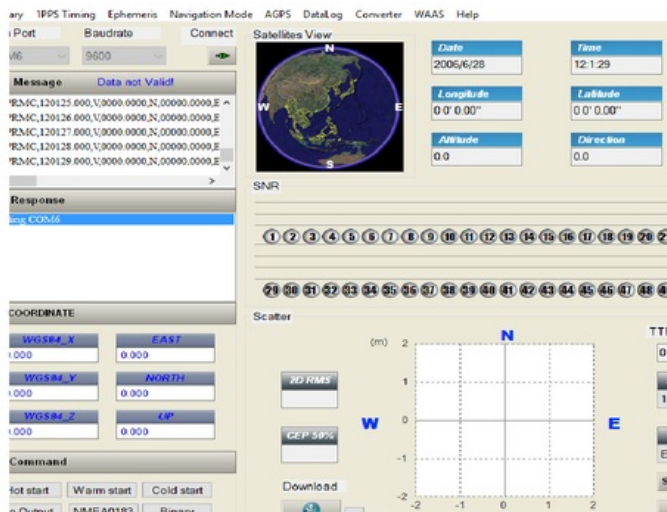
## Testing

Functional testing, the device was powered up in stages, to mitigate against incorrect build of the circuit, that would have the potential to damage critical components. Following the power up the author used a multi meter to measure voltages across the circuit to find incorrect voltages and zero volts in other parts to the circuit. The cause of the faults had been the author referencing the incorrect datasheet for the wiring of the 3.3VDC voltage regulator and the zero-volt fault was the positioning of components on the breadboard were found to be open circuit. At this stage the author made modifications to the circuit and added LED's on all the power lines throughout the circuit to give a visual indication of power being present, then followed up by measuring the voltage either side of the regulators.

When the power cycle had been confirmed, the GPS module was setup using the manufactures software tool to setup the following parameters;

- Number of NMEA Message types, RMC only (Recommended Minimum Settings) selected for navigation (See Fig T1.)
- GPS Update rate, 1Hz selected (1Hz – 20Hz)
- Selected Baud rate set to 9600

Fig T1.



Following setting up the GPS Module parameters the next set was to setup a terminal to read from the serial port of the PC, the GPS module was connected to the PC via an RS232/USB interface connected to the GPS Tx & Rx terminals. The terminal that was selected was Putty that is a free download, a number of version of PuTTY are available the one selected for the testing was PuTTY.exe

Opening PuTTY, the terminal was setup, saved, renamed and loaded, this isn't essential however it does save time if used at a later date. The setting changed in PuTTY were the Baud rate 9600, 8 data bits, no parity bit, 1 stop bit.

With the terminal setup and the GPS connected to the serial port, the device was switched on for testing, following a short delay the data started streaming in the PuTTY terminal, this proved the GPS module was communicating data directly between the device and the PC.

The data seen was complete data but as the test was carried out indoors the GPS coordinates were being seen as zero values and the data valid indicator was displaying XX for invalid data.

The next test was to confirm the GPS module could lock onto satellites and determine its location, the date and time which are part of the RMC NMEA data stream.

The device was taken to a location outside away from tall buildings so the best line of sight could be achieved between the GPS receiver and the satellites.

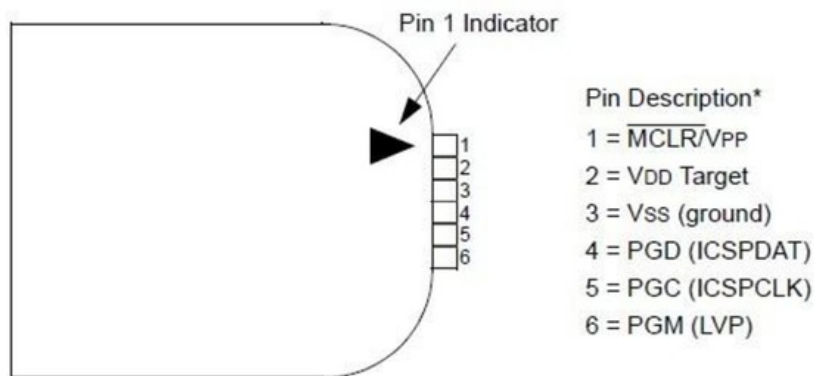
Read GPS data at Hyper terminal

- Test GPS is communicating with microcontroller
- Print to LCD Screen
- Record data to SD card
- Read SD card from PC

Functional testing was carried out in stages, writing a bit code then testing it bit by bit, following getting code operating correctly using the Proteus virtual simulation mode, the staged writing and testing was used to mitigate against potential delays due to debugging the full program if a fault was found.

To download the program from Proteus to the microcontroller, a MATLAB software tool was used MPLABXPIPE (MPLAB Integrated Program Environment) and a device called PICIT 3, MPLABXPIPE is the software tool load that's has a function interface with the microcontroller and the PICIT 3 is a hardware interface that has a USB connection to connect to the PC and 6 no. outputs to connect to the PIC, the PIN OUTS for the PICIT 3 are seen in Fig T1. PICIT Pin 1 is connected to pin 1 (MCLR) of the microcontroller, PICIT pins 2 and 3 are power and ground connected to the power and ground pins on the microcontroller, if the microcontroller has more than one power/gnd terminals they also must be connected to the PICIT3. Pins 5 and 6 are the data transfer pins and are connected to the corresponding pins of the microcontroller at pin 27 and 28. PICIT pin 6 is not used for this microcontroller.

Fig T1.



When the relevant connections had been made the software tool MPLABX was launched see fig T2. to prepare the program download to the microcontroller. From the settings tab advanced setting was selected and the software requested a password to continue, the password was entered and the advanced settings window appeared see fig T3. The device 'PIC18F24J50' was selected from the drop down list, then the power requirements were checked in the power tab button the power settings are set when you select the device therefore the max voltage was already set to 3.25VDC, finally within the same tab check the tick box so the PICIT powers the device.

Following completing the steps above the connect button was pressed the device wasn't seen initially and a message appeared in the output window stating a wiring issue was present, the power and ground connection had been placed on the wrong pins, following rerouting the cables to the correct pins the connect button was pressed again and the software confirmed it had successfully connected to the device. The next step is to locate the Proteus Hex file by using the browse function in the source tab, when the file has been found and selected the program button was pressed see Fig T4.

The first attempt downloading the program didn't work and an error was presented stating there was debug information on the device, the program button was pressed several times with no avail to the point the device couldn't be seen by MPLABX. Through online research a solution to the problem was found, the device already had a program downloaded to its memory that was download previously by the author for testing purposes, the device can only operate one program at a time therefore the original program had to be erased to allow the new program to be downloaded to the device.

Fig T2.

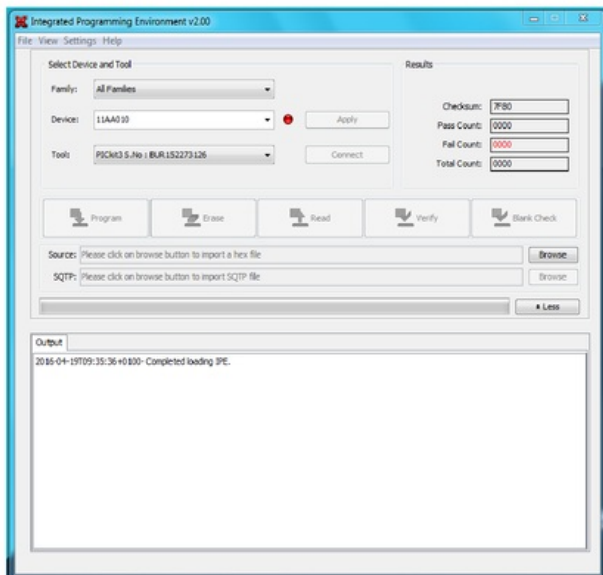


Fig T3.

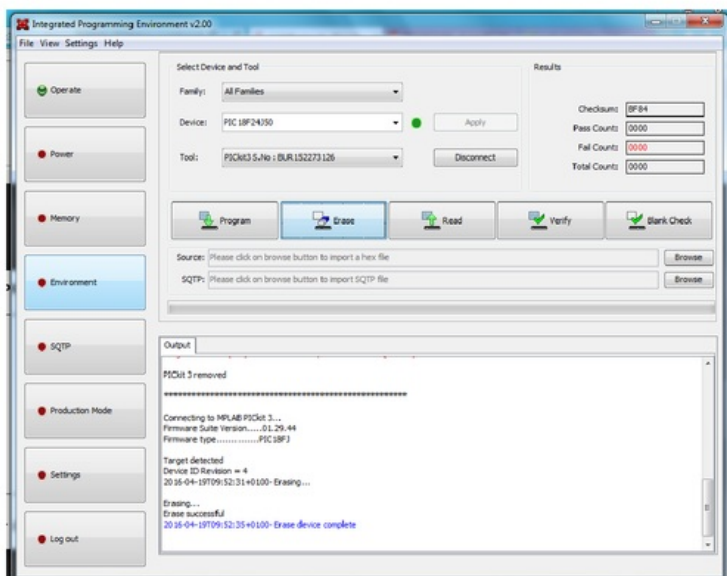
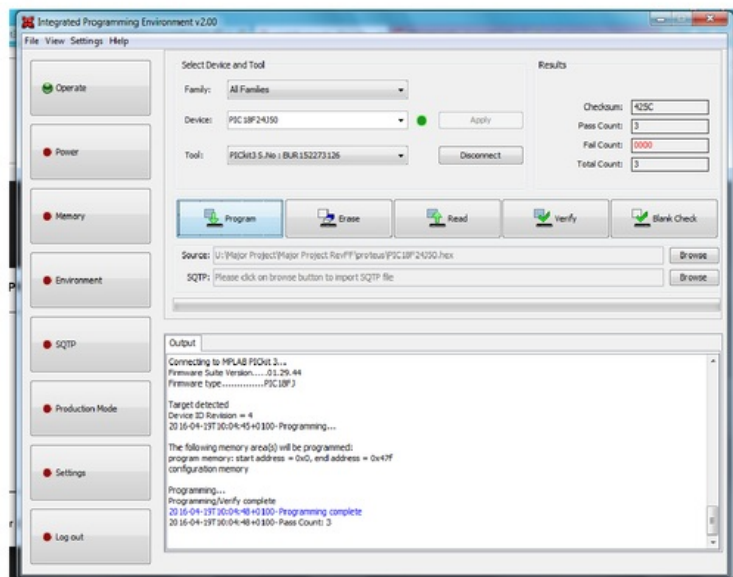


Fig T4.



## Hardware Testing

After the program was downloaded the device was tested and was successful first time see Fig T5. – Fig Tx. The first figure shows the start-up display confirming the GPS manufacturer and the firmware version, the second figure show the next message following the start button being depressed the display confirms the device is a GPS mileage Tracker, the final figure shows the latitude & Longitudinal coordinates because the device is inside no GPS signal is being received therefore zeros are shown in place of the coordinate's.

Fig T5.

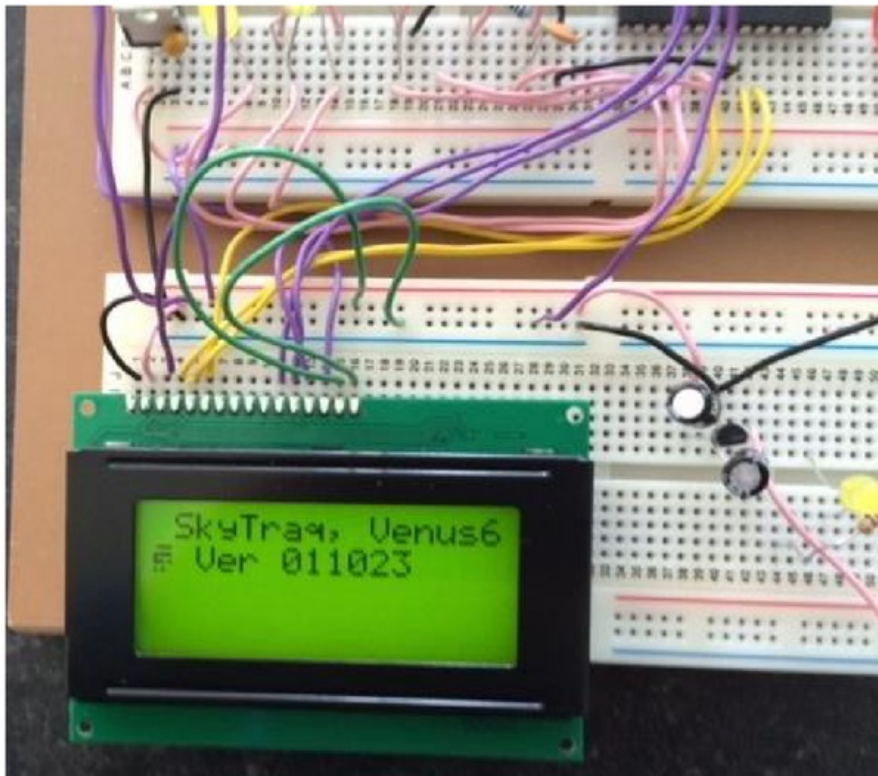


Fig T6.

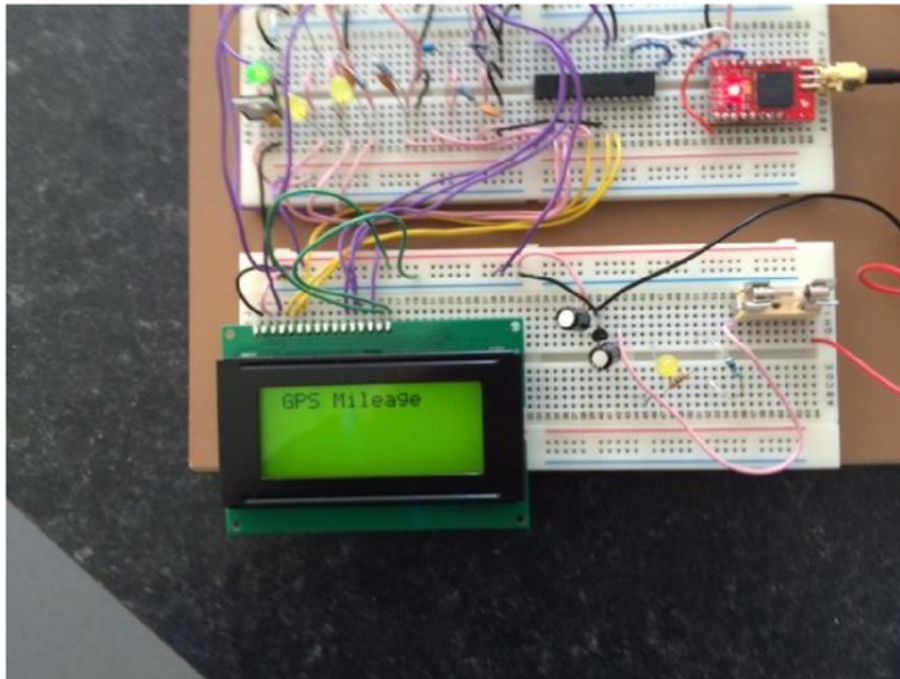


Fig T7.

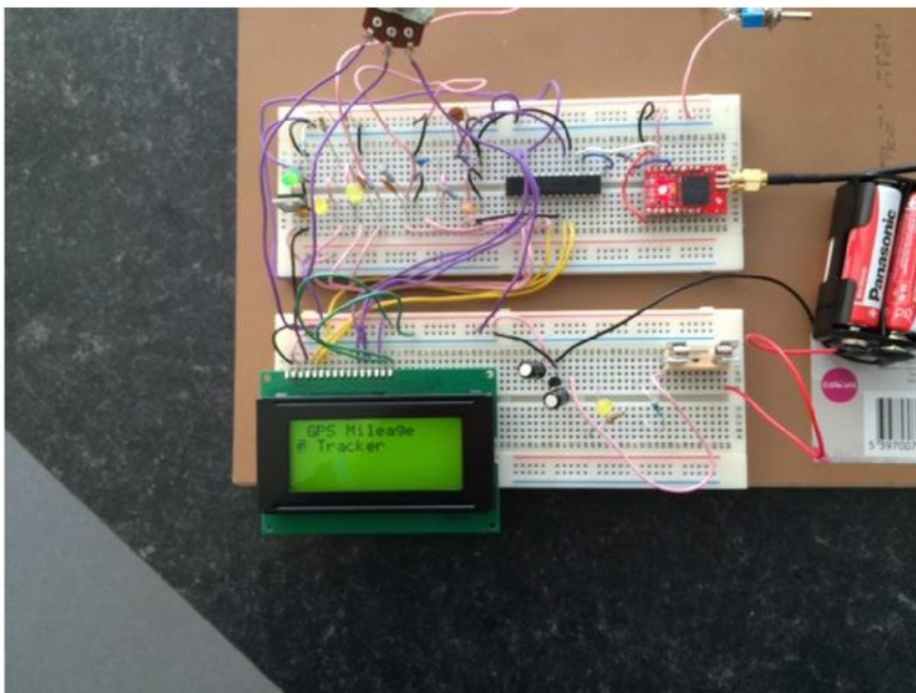
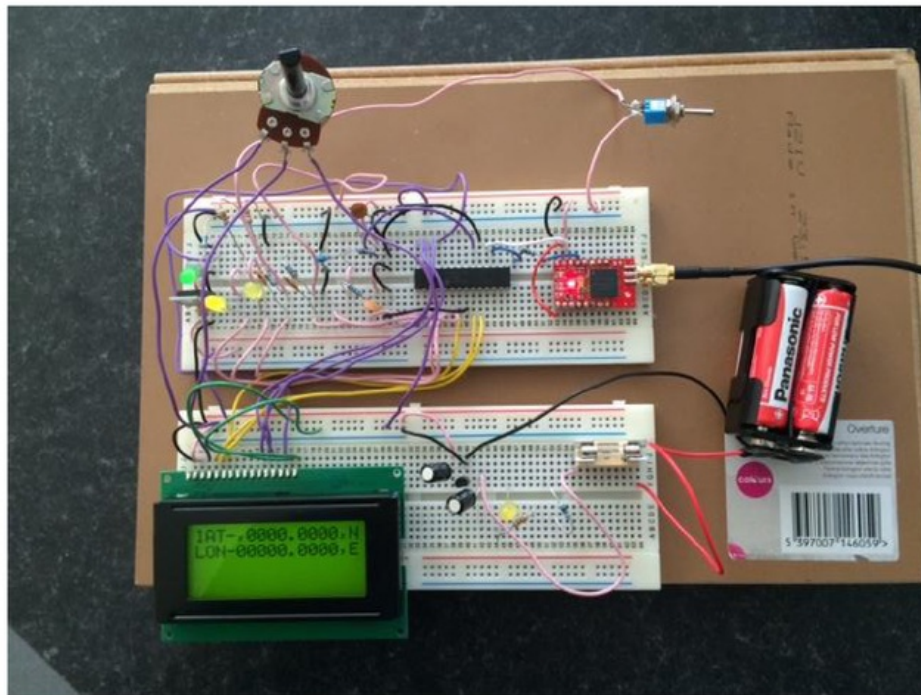


Fig T8.



## Analysis

### Analysis methods

#### Logging

Logging coordinates in different locations, recording the coordinates seen on the LCD Display then moving to new location, referencing the text files of the NEMA data seen Appendix F Table A1 has been completed to present this data. Analysing the data, the UTC time is updating every second, the Status field is V which referencing the GPS datasheet in appendix D 'Navigation receiver warning'. The remainder of the fields up to the mode field are not changing because the GPS receiver was stationary position at the time of data recording, finally the checksum value is changing because the time is incrementing in value, the character value change will see the binary ASCII value change that represents the seconds incrementing, the checksum is the XOR value of the full GPS string therefore character change will result in a different checksum value as seen in table A1. The coordinates were checked on mapping software and the position was of the coordinates were in Holland near Antwerp not in the UK where the GPS receiver was being tested, see fig A1.

The NMEA RMC data recorded using the Skytraq software tool, the data was converted from text file to KML using an online tool [24], KML file format is one type of file type that is used for Google Maps My Maps [25]

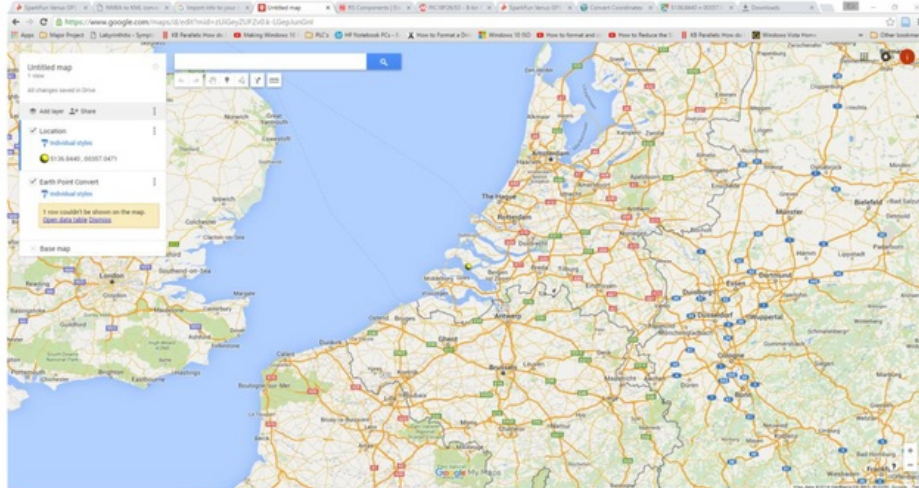
Table A1.

| NEMA | UTC Time | Status | Latitude  | N/S | Longitude  |
|------|----------|--------|-----------|-----|------------|
| RMC  | 21.04.39 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.40 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.41 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.42 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.43 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.44 | V      | 5136.8440 | N   | 00357.0471 |
| RMC  | 21.04.45 | V      | 5136.8440 | N   | 00357.0471 |

Table A1 (continued)

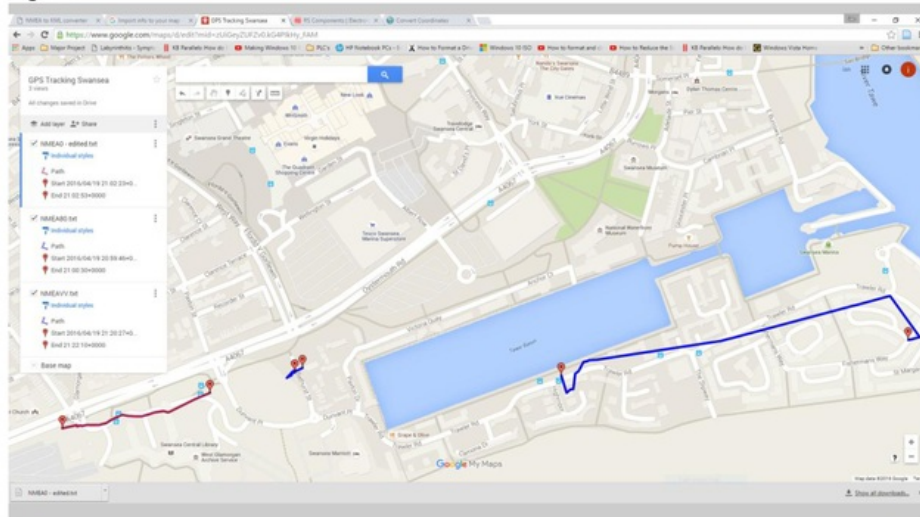
| E/W | Speed | Course | UTC Date | Mode | Checksum |
|-----|-------|--------|----------|------|----------|
| W   | 011.4 | 250.7  | 190416   | N    | 62       |
| W   | 011.4 | 250.7  | 190416   | N    | 6C       |
| W   | 011.4 | 250.7  | 190416   | N    | 6D       |
| W   | 011.4 | 250.7  | 190416   | N    | 6E       |
| W   | 011.4 | 250.7  | 190416   | N    | 6F       |
| W   | 011.4 | 250.7  | 190416   | N    | 68       |
| W   | 011.4 | 250.7  | 190416   | N    | 69       |

Fig A1.



The log was recorded again see fig A2. 3 individual logging events were recorded this time and then transferred onto one map, more routes were recorded using the Skytraq software tool however the author found it difficult to get a good GPS signal, reviewing the GPS data files seen in appendix F the recording did not get any satellite data.

Fig A2.



Screen shots were also taken of the Proteus simulator during receiving GPS data as see below in Fig A3 – Fig A5

Fig A3.

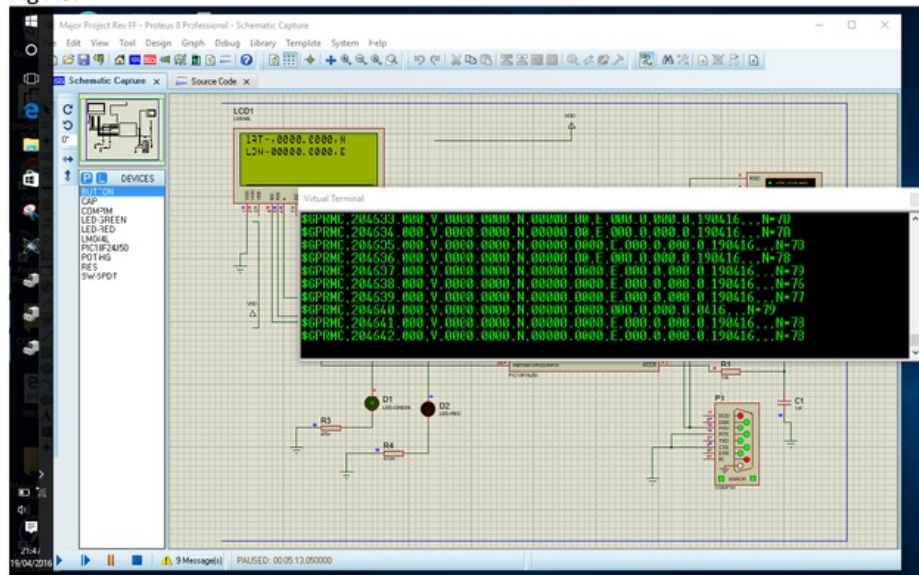


Fig A4.

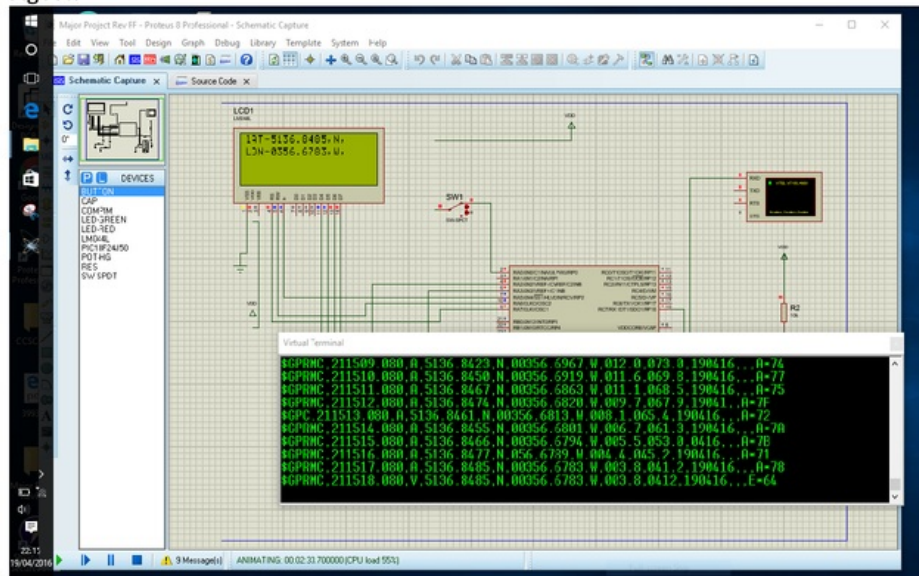
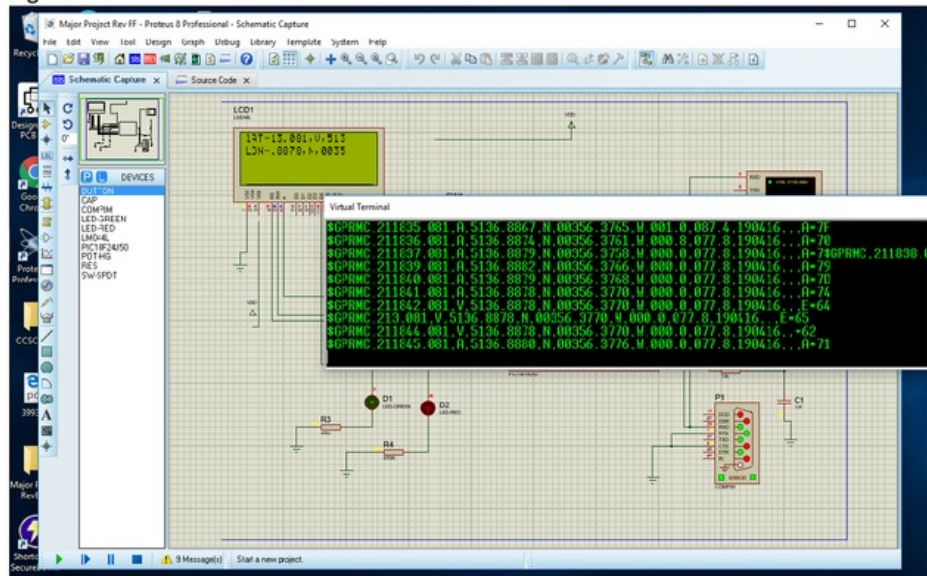


Fig A5.



### Sentence type

Change GPS sentence type and record the information seen on the LCD display, the sentence type is changed by reprogramming GPS module to change the sentence output.

| NEMA | UTC Time | Status | Latitude | N/S | Longitude |
|------|----------|--------|----------|-----|-----------|
|      |          |        |          |     |           |
|      |          |        |          |     |           |
|      |          |        |          |     |           |
|      |          |        |          |     |           |
|      |          |        |          |     |           |
|      |          |        |          |     |           |

| E/W | Speed | Course | UTC Date | Mode | Checksum |
|-----|-------|--------|----------|------|----------|
|     |       |        |          |      |          |
|     |       |        |          |      |          |
|     |       |        |          |      |          |
|     |       |        |          |      |          |
|     |       |        |          |      |          |
|     |       |        |          |      |          |

#### Record data

The method of recording data Will be through manually taking the longitude and latitude c coordinates from the LCD Display and Recording this information in table former. As back evidence photographs will be taken of the LCD Display as the coordinates change, also the screenshot of the location Will be taken from Google maps. Photos showing the device operating can be seen in Figures T5 – T8 however additional photos have been taken and can be seen in Appendix E.

## Future Enhancements

Future enhancements have been considered to show how the design could be improved or to give the user more functionality, in this case it would be both as the final design was a prototype. The Future enhancements have been categorized into hardware improvements, software improvements, other improvements, future design development and future design costs.

### Hardware Improvements

Bluetooth interface - Bluetooth compatibility could provide the user with a function to Interface the device to download the stored data directly to a mobile device such as smart phone or to upgrade firmware etc..., it could be also used to pair with other devices with Bluetooth technology.

Wi-Fi interface - The Wi-Fi interface could provide the user the Function that would almost provide almost real-time data aware the device is within a certain period using Wi-Fi hotspots connectivity, the device could be set to synchronize every time it is in range of Wi-Fi this would certainly reduce the storage requirement on the device.

Removable storage - Removable memory card option could provide users with an option to upgrade to larger memory storage capacity to enable them the store more data for a longer period.

Power consumption – Due to the significant power consumption of the device efficiencies would have to be made, an easy improvement to reduce the power drain would be to consider using MOSFETs to switch high load such as LEDs. Another power saving method could be reducing the size of the LCD display or removing it completely, has the potential to reduce the load by almost 25%.

### Software Improvements

Auto logging, an auto logging function would start to save data as soon as the position changed without the need to provide an intervention to start recording. With the GPS update needing to be as quick as possible so the change of position is identified as soon as possible be the microcontroller, the processing power and storage of the existing device design would be the limiting factor therefore the microcontroller would possibly need more processing power.

Auto synchronization – Auto synchronization would be a function that would synchronize the device with another interface (smartphone or directly to the cloud) when within range. Additional hardware and software requirements would be needed for this improvement and, therefore it's expected the microcontroller would like to require upgrading.

### Other Improvements

Cloud storage - Cloud storage option would give user access to data from the device whenever they could connect to the Internet, therefore the storage requirement of the data on the device would be less.

Smart phone application – A smart phone Application could be used with Bluetooth connectivity to extract Data from the device, to add to Journey information and possibly show your route on mapping software.

### Future design development

The future design that has already been considered for the development, is USB connectivity and removable memory storage, USB connectivity will allow the user to download the data stored on an SD card. The author has developed the following design documentation a structure diagram, a schematic diagram of the circuit, a PCB design and Pseudo can be seen in appendices listed below.

### Future design Costs

The future design discussed in the previous section Will require minimum costs as the only hardware to purchase would be a SD card breakout board, USB socket and some consumables to connect to existing circuit design. If this if this design was to be manufactured there would be also additional some design development costs.

## Appendices

### Appendix A – Researched technologies

- Appendix A - Capturing Serial Data
- Appendix A - Configuration Management Database ODBC
- Appendix A - Embedded C programming
- Appendix A - GO5 Spec Sheet
- Appendix A - GPS Buying Guide - SparkFun Electronics
- Appendix A - GPS Receiver Architectures and Measurements
- Appendix A - GPS Tracking paper IEEE resource
- Appendix A - GPS Receiver Module data sheet
- Appendix A - GPS signals - Wikipedia, the free encyclopedia
- Appendix A - GPS tracking server - Wikipedia, the free encyclopedia
- Appendix A - GPS Vehicle Tracker Manufacturer Kolkata India

### Appendix B - Feasibility Documents

- Appendix B – Major Project Program & Gannt Chart

### Appendix C – Design Documents

- Appendix C – Major Project Rev X01 Easy PC Schematic
- Appendix C – Major Project Rev X01 Easy PC Project File
- Appendix C – Proteus Design file
- Appendix C – Pebble Breadboard design
- Appendix C – C Program code

### Appendix D – Costings & Hardware datasheets

- Appendix D -PIC18F24J50 datasheet
- Appendix D -Venus GPS Module datasheet
- Appendix D -LCD Display datasheet

### Appendix E – Test & Analysis documentation

- Appendix E – Photos

### Appendix F – Future Enhancement documentation

- Appendix F – Data capture files

## References

- [1] 'Global Positioning System', *Wikipedia, the free encyclopedia*. 14-Apr-2016.
- [2] 'Vehicle tracking system', *Wikipedia, the free encyclopedia*. 01-Apr-2016.
- [3] 'TE2 GPS Vehicle Tracking System - Applications' . .
- [4] 'Understanding the Global Positioning System' . .
- [5] 'IEEE Xplore Abstract Record' . .
- [6] 'Serial Communication - learn.sparkfun.com' . .
- [7] P. H. W Hill, in *The Art of Electronics*, 3rd ed., Cambridge: Cambridge University Press, 2015, p. 1036.
- [8] 'Serial Communication - learn.sparkfun.com' . .
- [9] 'SparkFun Venus GPS with SMA Connector - GPS-11058 - SparkFun Electronics' . .
- [10] '39931d.pdf' . .
- [11] 'PIC18F24J50 - 8-bit PIC Microcontrollers' . .
- [12] 'PIC18F26J53 - 8-bit PIC Microcontrollers' . .
- [13] 'FDCC1604A-FLYYBW-51SE | Fordata FDCC1604A-FLYYBW-51SE Alphanumeric Transflective LCD Monochrome Display Green, LED Backlit | Fordata' . .
- [14] 'SparkFun SD/MMC Card Breakout - BOB-12941 - SparkFun Electronics' . .
- [15] 'SparkFun SD/MMC Card Breakout - BOB-12941 - SparkFun Electronics' . .
- [16] 'AA battery', *Wikipedia, the free encyclopedia*. 07-Apr-2016.
- [17] 'TS1084CZ 3.3V 5A Low Dropout Positive Voltage Regulator | Maplin' . .
- [18] P. H. W Hill, in *The Art of Electronics*, 3rd ed., Cambridge: Cambridge University Press, 2015, p. 1080.
- [19] P. H. W Hill, in *The Art of Electronics*, 3rd ed., Cambridge: Cambridge University Press, 2015, p. 802.
- [20] 'Venus\_GPS-v21.pdf' . .
- [21] 'Snapshot' . .
- [22] 'PIC vs. AVR smackdown' . .
- [23] 'PEBBLE - Software - PICAXE' . .
- [24] 'NMEA to KML converter' . .
- [25] 'Import info to your map - My Maps Help' . .