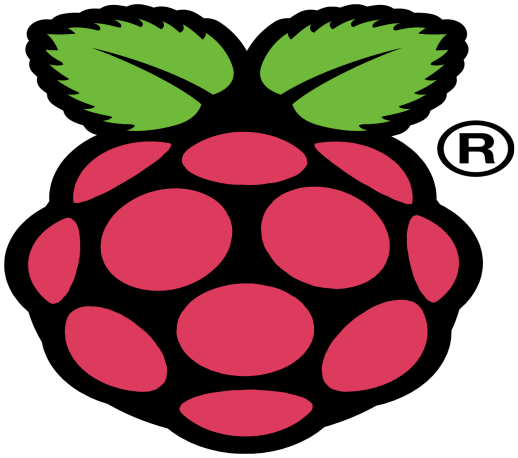
USER INTERFACE DESIGN SPECIFICATION

<PiMyRide>: <David O'Regan & Alan Kehoe>



Version 1.0
January 10th 2012

UI Contacts:
David O'Regan (david.oregan7@mail.dcu.ie) 10331017
Alan Kehoe (alan.kehoe@mail.dcu.ie) 10735389

1 Table of Contents

2 Introduction

- Purpose
- Definitions, acronyms, and abbreviations
- The scope of this spec
- References to other support documents such as research reports, prototypes, etc
- Summary

2.1 Purpose:

Overall PiMyRide has three main goals, be able to gather information from a car, trend the cars performance over time and finally relay that information to the end user. Our target audience would be car enthusiasts or anyone who wanted to monitor their cars performance over a period of time. The product will be internally designed by David & Alan, all testing done will be in house until the end of the product development.

2.2 Definitions, acronyms & abbreviations:

Pi: The **Raspberry Pi** is a credit-card-sized single-board computer developed in theUK by the Raspberry Pi Foundation with the intention of stimulating the teaching of basic computer science in schools.

OBDII port: **On-board diagnostics**, or **OBD**, is an automotive term referring to a vehicle's self-diagnostic and reporting capability. The OBDII port is the standardized version on almost all cars after 2001.

GPS chip: Adafruit Ultimate GPS Breakout – 66 channel w/10 Hz updates – Version 3

Connection: USB - OBD II cables which allows to the computer in this case to power the cable and retrieve codes from the computer, check sensor data and do anything else a hand-held OBD II code reader can do.

LCD Screen: Standard HD44780 LCDs are useful for creating standalone projects. 16 characters wide, 2 rows.

2.3 Scope of spec:

This specification covers the requirements for design, fabrication and coding of the PiMyRide model. It is applicable for cars that contain an ample power supply i.e. a cigarette lighter and an OBDII port. This will include a user interaction section for data display, an operating system overview and a breakdown of the trending software to be built into the hardware.

This system will use a multitude of hardware and software to achieve a seamless interaction between our system and an engine management system.

The two main goals of the system are as follows: 1) to be able to interface with the car's management system and pull information from the cars sensors. 2) To be able to use that information to trend the cars performance over time i.e. from cold on start up to having run the car for half an hour.

2.4 References:

[1]http://www.raspberrypi.org/faqs
[2]http://www.raspberrypi.org/phpBB3/viewtopic.php?f=35&t=20452
[3]http://www.techrepublic.com/blog/european-technology/10-coolest-uses-for-the-raspberry-pi/505
[4]http://adafruit.com/products/181
[5] http://www.adafruit.com/products/746
[6]

2.5 Summary:

The PiMyRide project consists of two main parts, these will be tackled by both members of the team on a equal scale.

The first main part of this project is to get a raspberry Pi minicomputer to successfully interface with a car through its ODBII port. The Pi needs to be able to read data from the car's engine management system and sensors. Things like miles per gallon, speed, temperature and distance traveled are all variables we want the computer to be able to read from the car.

The second main part of this project is once the raspberry Pi can read and record data from the car is to design and implement an algorithm that will trend that cars performance over time. This will be done by getting the starting variables, recording more over a set time frame and then comparing the figures against the ones at the start.

Once both of these components are in place, we relay the information to the user via an LCD screen we will attach to our Pi.

We hope to be able to display all the car's variables to the user as well as a visual representation of the cars trending performance.

3 Overall Description

- Product Perspective
- PP(User characteristics)
- Product Components
- Product Constraints
- Proposed future requirements

3.1 Product Perspective:

Our product PiMyRide, while relatable with other products done using a raspberry Pi is very self-contained. Other Pi-Car projects have been attempted and a few have been successful, our product differs as we not only intend to interface with the car but also provide a trend of the cars performance over time.

This product is not part of a larger system and hence none will be discussed in this functional spec, our product is the full scale of the finished product.

The PiMyRide system will be finished as a stand-alone product; it will include various pieces of extra hardware but will be presented as the one piece.

The building of our hardware choices will be made to ensure the product functions correctly across a multitude of cars, though the OS of the device will stay static. Users will be open to customize the product, though we will have already identified the best possible software and hardware choices for integration.

The PiMyRide system will be designed to rival On-Board car computers & car fault detection computers.

3.2 User Characteristics:

Our user audience will mainly consist of motor enthusiasts and possibly garages.

We will assume our users have some knowledge of cars, hence the interest in the performance trending application of our product. Though it is important to note that only a small amount of car knowledge is needed i.e. OBDII port connection location and most of this information can be found via google.

3.3 Product Components:

The PiMyRide system will contain the following components:

Hardware:

1) 1 raspberry Pi with a Linux application installed
2) 1 Connection cable to OBDII port, 1 connection cable to power source i.e. cigarette lighter
3) 1 GPS chip, connectable via pins that supports altitude
5) 1 LCD screen(3-6 inchs), pin connectable that will support basic display

Software:

6) Information logger : A program that will grab information from a car's sensors at set intervals
7) System display: A program that will display set information onto the LCD screen
8) Trend catcher: A algorithm that will trend the cars performance in certain categories over time.
9) Logger: A program that will store the information "grabbed" from the car and save it into a readable form i.e. spreadsheets/XML tables.

3.4 Product Constraints:

- User Interfaces
- Hardware Interfaces
- Software Interfaces
- Hardware Constraints
- Memory Constraints
- Assumptions & Dependencies

3.4.1 User Interfaces:

The PiMyRide system contains a multitude of programs running on the single Raspberry Pi, these will be automated and will require no user contact to initiate other than turning the device on. The display from the LCD screen for the programs will be consistent, done in the same font style. This is to make the system more user friendly. The output from the screen will also be recordable and transferable to an out-house system i.e. a laptop or PC.

The product consists of one main program and one main algorithm, both of which have will have no DOS driving implementations. The Pi will be running a Linux operating system, this will interface with a laptop or PC in order to communicate the stored tables of information (Most of which will be communicated to the user via LCD screen). Where possible all user interaction will be simple (making good use of plug and play software). This means that our user will deal with a variety of screens, inputs and outputs. Our goal is to constrain these as much as possible for consistency but this may not always be possible.

3.4.2 Hardware Interfaces:

The PiMyRide system is a utility designed to work with two main pieces of hardware, A car engine system and a PC. Both pieces of hardware are fully compatible with the main component of the PiMyRide system (Raspberry Pi) via connection cables (OBDII, USB). The software will be designed in python and so will be compatible with almost all computers available today.

All information gathering algorithms used by the system that rely on hardware (Car connection) must be able to function regardless of the hardware being installed.

3.4.3 Software Interfaces:

The PiMyRide system will require a reasonably up-to-date car engine system i.e. 01 or greater. Being linux based, the PiMyRide system can interface with any software system i.e. A mac, windows or another linux operating system, Windows version must be XP or greater.

3.4.4 Memory Constraints:

The PiMyRide system should be able to function with relatively limited memory as only data is being stored.

However, since the Linux OS for the raspberry Pi must be installed on a external memory stick(smallest size being 4GB), there will always be an abundance of memory for the system to operate on. The Linux OS takes up 1.9 GB approx.

3.4.5 Other Constraints:

- The system will depend on a car's engine system being accurate. Speed measurement, temperature gauge, fuel level and distance traveled must all be accurate within a reasonable margin.
- Being an open system, the raspberry Pi is compatible with a range of LCD screens, the system will depend of a screen big enough to display needed information but small enough to be acceptable.
- The system will depend on access to a GPS chip that can deal with altitude.

3.4.6 Assumptions & Dependencies:

Certain components of the PiMyRide system contain assumptions and dependencies. Should any of these prove not to be the case during development, then this functional specisifaction will be review and edited accordingly.

Known assumptions are:

- It is assumed the car will have an OBDII port connection & a cigarette lighter generating at least 700mA at 5V.
- It is assumed the car owner will have access to a laptop or computer to view logged data
- It is assumed our LCD screen will be able to display all the data we want to communicate including graphing trends.

3.5 Proposed future requirements:

The GPS chip needed for the PiMyRide system needs to be ordered from America and so shipping will delay the process of inclusion into the trending algorithm.

At time of writing this functional specification, we are not aware of any other future requirements that might be delayed.

4 Functional Requirements

- System Start-Up & Instillation
- System Configuration & start
- System Settings
- Display
- Display Options/Interaction
- Data Logs
- PC – PiMyRide operation

4.1 System Instillation:

- **Description:**
  This is the first part of starting the system. The user will receive the product out of the box, all the software will be pre-installed and will be aiming for a "plug and play" system. The user will need to connect the PiMyRide system to their OBDII port and cigarette lighter as will be described in the instructions. Then it's a matter of turning on the system. At the time of writing we are hoping to implement the system switching itself on with the car start.

- **Criticality:**
  While simple, this stage is the crux of our entire system. The system must be connected to the car to run and it must be connected properly. We want a user to be able to install the PiMyRide system with the same ease they would connect an Ipod to their computer.

- **Technical Issues:**
  The system will come with the two connection cables needed to begin the setup process. It will be a matter of locating the OBDII port in the user's car model and then physically integrating the hardware of the system.

- **Dependencies:**
  This stage is dependent on a user having an OBDII port connection and a cigarette lighter providing at least 700mA at 5V power.

4.2 System Configuration & Start:

- **Description:**
  Now that the system hardware has been installed and connected to the car itself we need to boot the machine and let the configuration take place(at the time of writing we are hoping to implement a auto-config which will require virtually no user input). The system will have been powered on via the car start and the software will configure to the OBDII connection to syuscessful recivve information from the cars sensors.
- **Criticality:**
  This part of the system is again utterly critical. If the PiMyRide cannot configure to the OBDII connectiona dn receive the information provided by the car, none of the system will operate.

- **Technical Issues:**
  The system will be connected to the car via the OBDII port to a USB receiver and to the cars power supply via the cigarette lighter. The PiMyRide system will power with the car's engine being engaged and then configure to the OBDII connection.

- **Dependencies:**
  This stage will depend on the user having successfully installed the PiMyRide system via step 1. This stage will also depend on the system powering with the car start.

4.3 System Settings:

- **Description:**
  The system will be available to edit but at the time of writing, this is only available through a PC interface and cannot be done through the systems LCD display. This will allow the user to set certain displays as permanent, which ones to cycle and what type of data to store.
- **Criticality**
  This part of the system is only as critical as a user sets it to be. Which certain users will find this a novelty, the critical component lies with the developer as this allows us to be sure the systems interaction and car sensors are functioning properly during operation.

- **Technical Issues:**
  To achieve this our system will need to be configured to display one type of feedback at a time, either allowing it to hold certain information feeds or cycle them. This will be done via the PC integration where we will be able to edit the display of the variables.

- **Dependencies:**
  This stage will depend on the PiMyRide system being intractable with another PC or having the needed hardware to boot the raspberry Pi from its own memory. This stage also depends on the raspberry pi software allowing us to set static or dynamic displays on incoming data flows.

4.4 Display:

- **Description:**
  Our system will have an LCD screen attached which will be powered from the raspberry Pi. The screen will be small enough to sit on a computer dashboard but big enough to display the variables we want the user to see during operation (3-6 inchs). Until implemented, the main display source will be via laptop or a spare monitor screen for development.
- **Criticality:**
  This stage is utterly critical. The point of our system if the user to be able to gather feedback and the trend of their cars performance over time and so a real time display is something we deem very important to incorporate along-side the PC display for the trends.

- **Technical Issues:**
  The LCD screen needs to be of a compatible type with the raspberry Pi model and it needs to be able to interact with either the pins or an available port. The screen will display information being received via the cars sensors and hopefully the trending stats of the cars performance.
- **Dependencies:**
  This stage is dependent on the LCD screen being able to display the type of information we want to communicate to the user. This will also be dependent of the screen being powered by the raspberry Pi itself.

4.5 Display Options/Interaction:

- **Description:**
  This part of the system continues from the previous stage (System settings) and relates to what the user will actually be shown on their display if its real time(LCD screen). Variables like Miles per gallon, speed, distance traveled, altitude and trending performance will all be possible options. The user will be able to view all of these variables as they are received and review them after via a PC connection with the raspberry Pi memory.
- **Criticality:**
  This part of the system is semi-critical to the developer as it refers more to the visual stimulus as opposed to the trends derived from the data gathered. For the user though we deem this utterly critical as it will be the selling point of the PiMyRide system as users assume a GUI being the standard on any system, thus lacking it would alienate a large amount of users.

- **Technical Issues:**
  The data will be received to the screen on a real time bases(Every second), this data will then be relayed to the raspberry Pi which will re-direct it to both the screen and memory for storage.

- **Dependencies:**
  This stage depends on the raspberry Pi software allowing an operational interaction to be developed; which will allow the user to cycle outputs as they are received. This stage will also depend of the LCD screen being able to handle the high level of output the user will want to receive (Data logged every second).

4.6 Data Logs:

- **Description:**
  During this stage, once the system is in full operation the data taken from the car's sensors will be stored in the Raspberry Pi's memory. This will allow the data to be mined for the performance equations and allow a user to review the entire data sheet; later at their leisure.

- **Criticality:**
  This part of the system is vital. It will be the entire bases for the performance equations and allow us to accurately monitor the cars performance over time. Without storing this information we could not graph anything about the car.

- **Technical Issues:**
  The data will be taken from the cars sensors on a real time bases(every second) and stored in a readable file on the raspberry Pi's memory bank. We are considering either XML tables or a simple spreadsheet format from which data can be taken during system operation.

- **Dependencies:**
  This will depend on the PiMyRide system connection being properly installed and the raspberry Pi's memory bank being big enough to accommodate for the data.

4.7 PC- PiMyRide operation:

- **Description:**
  This stage will be used mostly for development of the system but will also allow a user to interact with the system as a whole, set instructions and view stored data.
- **Criticality:**
  This part is simply needed, without being able to interface with another machine our project will not be able to perform.

- **Technical Issues:**
  The raspberry Pi will have its own operating system preinstalled on its memory. It can then be connected to another machine to be operated on either via connection cable or simply be removing the memory card.

- **Dependencies:**
  This stage depends on the PC in question being able to handle a simple USB connection or more ideally have a memory card slot.

4.8 Trending Algorithm:

- **Description:**
  This will be one of the most important parts of the PiMyRide system. Data will be taken from the car over a period of time, stored and then used in an algorithm/s to compute the cars general performance over a period of time in certain categories. An ideal example would be, on start up what is the fuel consumption of the car's and then again after having driven for a period of time.

- **Criticality:**
  This stage is critical and is a major component of this project.

- **Technical Issues:**
  The data will be taken from the cars sensors, stored in the tables and then an algorithm to compare the variables over time and trend the performance will be invoked, again this information will be stored.

- **Dependencies:**
  This stage depends on the data storage working properly.

5 <System Architecture>

5.1 <System Architecture Diagram(Fig 2.1)>

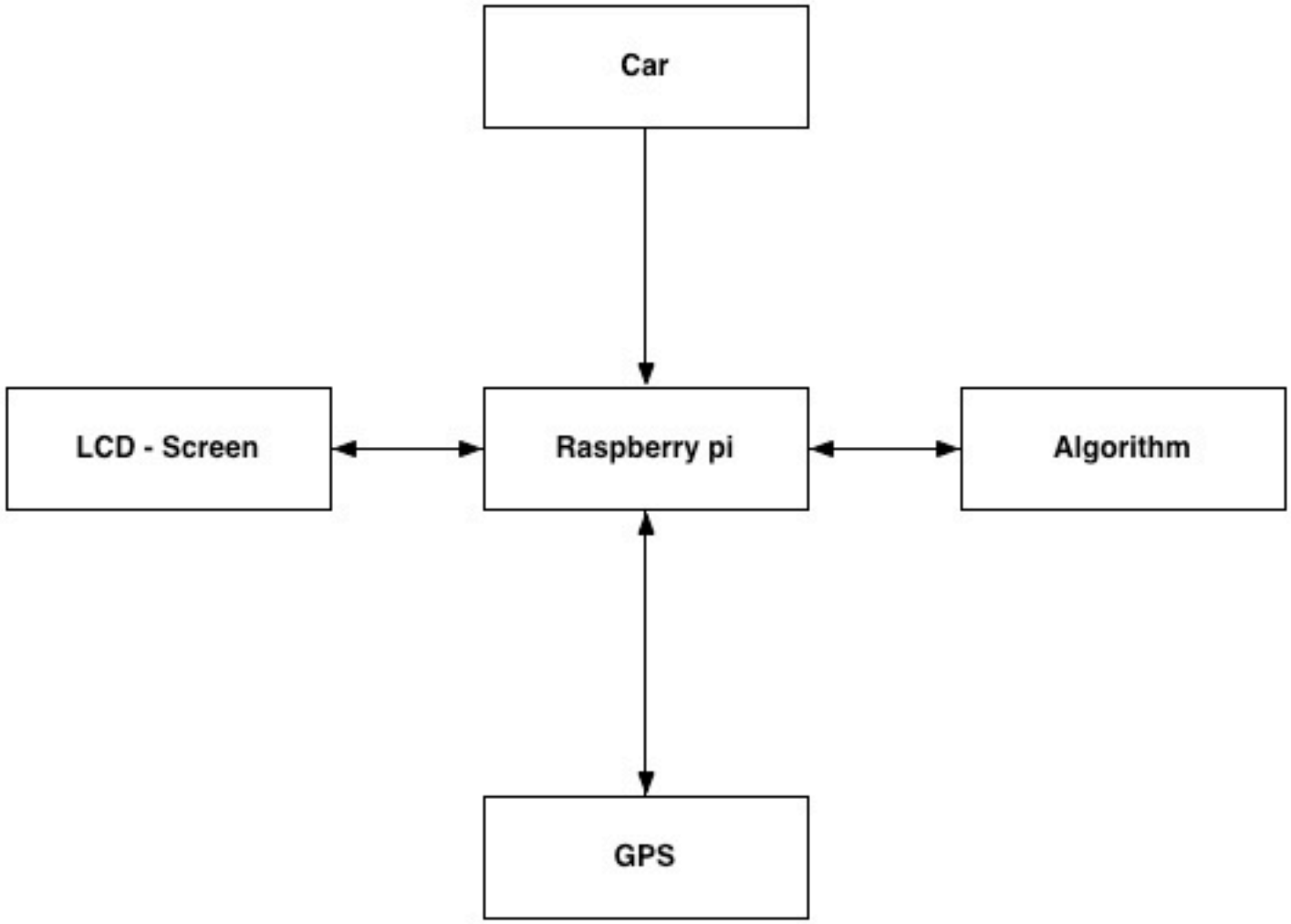


Fig 3.1 above illustrates the system architecture of our entire system. There are 4 main parts as shown in the diagram that all interact with one another to make the system successful. Note that at the time of writing the GPS element is optional depending on progress and the relevant deadlines. Our system begins with the actual raspberry Pi computer, the front of our system. This interacts with both the car via the OBDII and the GPS chip.

5.2 < System Architecture Description>

5.2.1 Raspberry Pi: This is the crux of our entire system. The raspberry Pi mini-computer will be the front end of our system along-side the LCD screen. The goal is to make the system "plug and play" for the users sake, simplicity being the key. We are striving for automatic start up and function on the car's engine being powered on.

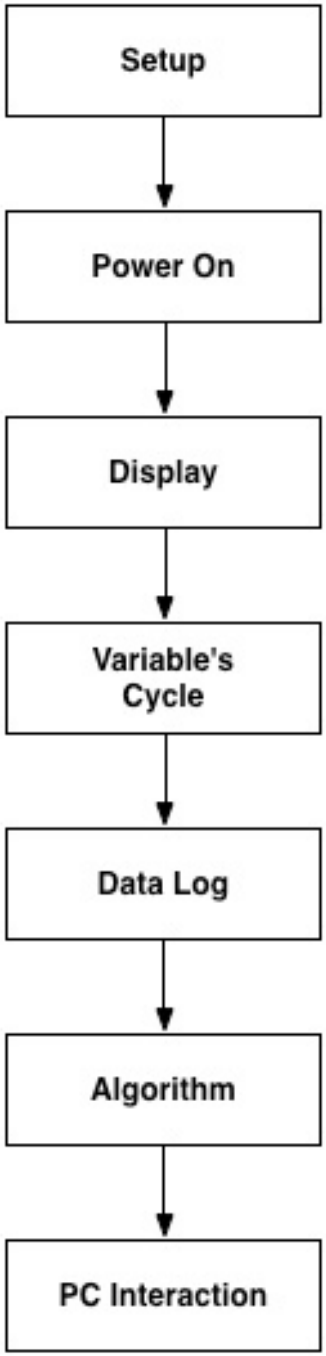
5.2.2 Car(OBDII/POWER): This is not a technical aspect of the system but a needed one. The raspberry Pi will interface with the car's sensors via its OBII port to gather information and via the cigarette lighter to be powered. The user will have virtually no interaction with this stage other than to plug the device in.

5.2.3 Trending Algorithm: This is our backend of the system and the core software part. The Pi will take the information supplied by the car and use the trending algorithm/s to generate how the cars performance is over a period of time. This information will then be given back to the user.

5.2.4 LCD Screen: The LCD screen is our way of displaying information to the user on a real-time basis. It will be connected to the raspberry Pi and communicate relevant information to the user as time goes on. Again the user will have virtually no input in this regard other than to position the screen within the car.

5.2.5 GPS(Optional): The GPS chip is an external element we would like to implement within the system given enough time. It will allow for the tracking of altitude and let the raspberry Pi's trending software take these variables into account. This will give a more accurate over grade of the cars true performance.

6 <High Level Design>


Setup
Power On
Display
Variable's Cycle
Data Log
Algorithm
PC Interaction

6.1 < High Level Description>

6.1.1 Setup:

Our user will connect the PiMyRide system into their car via the OBDII port and cigarette lighter connection.

6.1.2 Power-On:

The deceive will power up when the car has been turned on. The switch will allow the user to decide when to power the system on.

6.1.3 Display:

The LCD display will boot.

6.1.4 Variables cycle:

The LCD screen will start to display the cars variables.

6.1.5 Data Log:

The raspberry Pi will log the information from the car at a regular interval and store the data

6.1.6 Algorithm:

The raspberry pi will use the stored information as variables for the trending algorithm(car performance)

6.1.7 PC interaction:

The user can now view the data log after wards by connecting the Pi's memory to a PC.

7 Appendix

Section	Date	Author/s	Issue	Pages
A	January 10th 2012	Alan Kehoe David O'Regan	Appendix A – Stategy	26 – 26
B	January 10th 2012	Alan Kehoe David O'Regan	Appendix B – Relivent orginisations & people	27 – 27
C	January 10th 2012	Alan Kehoe David O'Regan	Appendix C – ACRYNOM Index	28 – 28

Appendix A - DESCRIPTION OF THE STRATEGY

Applications & Benefits:

- **Direct results:** *Users are provdied instant feedback on their cars performance during a trip and after for review at their lesuire.*
- **Public benefits:** *The common car owner is not forced to spent a large sum of money to investiage their cars performance.*
- **Collateral benefits:** *None*
- **Scope of potentional impact:** *Community and possibilbity for statewide.*

Critical elements:

- **Staff requirments:** *2 people*
- **Other requirments:** *Raspberry Pi, cables, car, GPS chip, PC and designed software*
- **Cost considerations:** *The program is costly to us as students but in the long run very cost effective*
- **Timing issues:** *Dealine allows for 6 weeks of full time work*

Potentional barries/obstecles:

- *Cost*
- *Development time*
- *Skills needed*
- *Hardware intergration*

Appendix B – Relevent orginisations & people

Location	Primary Contact	Building Block
Dublin City University	*Brian Stone*	*Project Supervisor*
Dublin City University	*Robert Hanrahan*	*Software development*
Dublin City University	*Joe Morris*	*Impartial feedback*
Kehoe Brother's	*Piotr Gwitz*	*Vehicle Technician*
N/A	*N/A*	*N/A*
N/A	*N/A*	*N/A*

***As the project continues this area will expand and be updated**

Appendix C – ACRYNOM Index

- **Pi:** *Raspberry Pi minature computer*
- **GPS:** *Global positioning system*
- **OBDII:** *On-Board Diagnostics*
- **USB:** *Universial Serial Bus*
- **PC:** *Personal Computer*
- **OS:** *Operating System*
- **Linux:** *A form of operating systems employed by the raspberry Pi*