Imperial College London

Final Report

EE2-PRJ E2 Project (2016-2017)

Group 2 - Autonomous IoT pet tracker with energy harvesting system

Jacopo Bacchelli (01112853)
Ryan Armiger (01065807)
Valentin Gourmet (01108641)
Harshil Sumaria (01053573)
Eugénie Ducoin (01116843)
Zengyang Pan (01046157)
Sophia Ahmed Ashford (01057326)
Alvaro Robledo Vega (01076364)

Supervisor:
Dr. Adria Junyent-Ferre

Submission date:
13^{th} March 2017

Contents

1 Abstract

Animal tracking devices are attractive as a source of additional safety for pets and as a tool for data collection in the case of wildlife. The recent advances in the Internet of Things has allowed the development of ultra-low power communication technologies. Combining the latter with an energy harvester, an animal could then be tracked indefinitely.

This report describes the progression of the project, discussing the developments since the Interim report. Whilst the power requirements of the device were clearly determined and minimised, identifying a suitable harvesting method and implementing a battery charger in this way proved to be particularly challenging. Research and testing is still required in this area to achieve the specifications set out previously in the Interim Report[1].

2 Introduction / Background

As part of the Second Year EEE curriculum, students are expected to design and build an innovative product to improve a socio-economic problem. A total budget of 100 pounds is available for the students to carry out their project.

The total number of cats and dogs worldwide is estimated to be 491.756 million [2]. One in three of these pets will get lost during their lifetime [3], creating a market for tracking devices.

A pet can go missing at any moment and the tracking products which currently exist are not self-rechargeable, and therefore must be constantly monitored to avoid them having a low battery level when needed the most. The same argument can be applied to wildlife tracking for data collection and behaviour studies: some animals travel large distances, others are dangerous to approach, therefore limiting the number of times such devices need to be replaced, aiding the work of many researchers. These problems justify the idea of an energy-harvesting tracking device, that would render the product autonomous for a long period of time, ideally indefinitely, with the product's life expectancy determined solely by the lifetime of the individual components.

In the Interim report, the product design was divided into three parts: an energy harvesting system, tracking system, and physical design. The latter includes shape, weight, aesthetic and ergonomics of the product.

The group was thus divided in two sub-groups, one concerned with generating power from the animal's movements, the other with tracking the animal. Physical design, depending on many factors and on both sub-systems, would have been carried out in the end, after each group would have identified the constraints linked to their part of the project.

The main criteria that were identified when choosing technologies and components were the creation of a constant and reliable energy source, the development of an ultra-low power localization system, and the design of a product that would not impede a pet in any of its movements.

Several concepts were researched and selected as part of the Interim report [1], but further investigation and decisions have taken place since then and are detailed in this report.

At the time of the Interim Report [1], a prototype for the tracking system was yet to be implemented, and further research had to be conducted in the area of energy harvesting.

The progress made from that point onwards and the outcome of this project are presented in this report.

3 Design Criteria and Concept Designs

The design criteria presented in the Interim report have slightly changed. The original aim was to have a product that fits all sizes of pets but considering the constraints that a small animal poses for the energy harvesting system, the device will be aimed at bigger animals. The other criteria have not changed.

The first criterion is the ease of movement for the animal and the modularity, adjustability and versatility of the device. This is met for bigger pets since the product will have a small size, in the centimetre range. By limiting the size of the pet the product is directed to, it is ensured that the energy harvesting system will be efficient while the device will still be comfortable for the pet.

The energy harvesting system has to be reliable and constantly produce a considerable amount of power. This criterion is achieved with a magnet oscillating in a coil, which generates power that is then stored in the battery. Anytime the pet moves, the magnet will move which is why kinetic energy seemed to be the most appropriate. To generate a larger voltage, more turns in the coils can be added, which does not add too much weight and volume to the device and thus can be easily adjustable during development.

Another criterion is the reliability of both the network and technology used to transmit data. Sigfox's popularity is growing due to its low cost, power and accuracy as its use in many products proves. However, due to the time constraints of the project, the network used for prototyping is GPRS. The GPRS technology has been well established for years and its reliability depends mainly on the quality of the components used. For the energy harvesting system, although this system of a magnet in a coil is not usually used in the context of a natural movement, it is reliable in shake flashlights.

An aesthetically pleasing product is also a key to the client's satisfaction. To meet this demand, several colours and patterns will be offered to the user when they buy the product.

Two final criteria are directly related to the time and budget constraints. Firstly, choosing easy-to-develop and easy-to-implement technologies will facilitate the conception stage and therefore shorten the time needed for manufacturing the device. Secondly, the cost of prototyping cannot exceed the given budget, and the final cost can't be much higher than 80 to 90 pounds, which is what competitors offer on average [1].

To meet these criteria, research has been conducted to choose the ideal concept and to find the most cost-effective components. Thus, as previously discussed in the Interim report, the GPS and Sigfox association have been chosen as well as induction as the method of energy harvesting.

Criteria	**Concept/Solution**
	Physical design
Ease of movement Versatility/Modularity Space available for circuitry	Target has been changed to bigger pets. Product to be placed on a collar or leg band.
Aesthetically pleasing	Options available for the colour and pattern when buying the product.
	Energy Harvesting
Efficiency of the system	Concept chosen (electromagnetic induction) has proven itself in other contexts and after research and testing, its efficiency was determined for this application. Frequency of different body parts at different speeds is studied in detail in section 4 to choose the location of the device on the animal.
Weight and Size	Designed for bigger pets that can handle a larger and heavier device without it being uncomfortable.
Constant energy production	Sufficient energy should be produced by a typical pet's daily movements
Easy to develop Easy to implement	Choice of magnet and coil
	Tracking system
Minimal energy consumption	Achieved with the Sigfox network and by setting the GPS to low power mode when not useful.
Reliable network	Sigfox
Easy to develop	Trackers are common and many existing projects can be used and adapted to this device.
Low cost	Choice of good value for money components and technologies, GPS and Sigfox.

4 Concept Development

The system was divided into two sub systems, each consisting of different blocks which contribute to the completion of the requirements.

4.1 Energy Harvesting

The self-charging device harvests power using the movement of the pet as a source of energy. It then transforms this into useful energy for the GPS tracker to use.

Using Faraday's law and electromagnetic theory, a harvester was designed and built. It consists of a magnet going through a coil wrapped around a cylindrical structure. Initially, it was going to be placed vertically around the pet's neck: It is the most comfortable place for the pet to wear it and is more aesthetically pleasing. The magnet would have been attached to a spring that would enhance any displacement from its rest position (see figure 4.1).

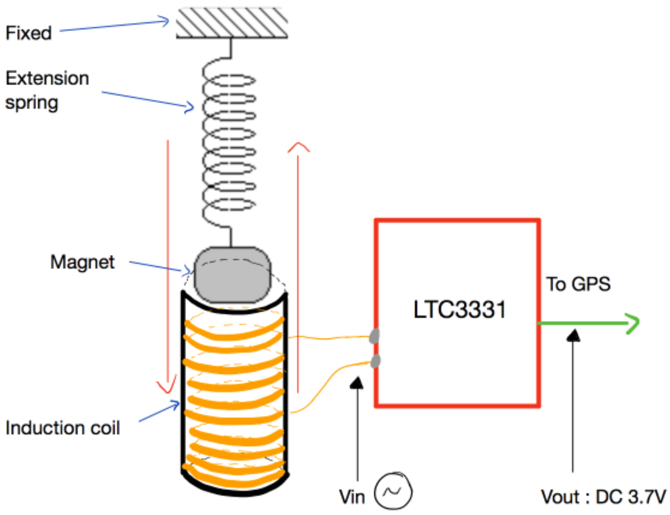

Figure 4.1: Initial schematic for energy generation circuit

However, many complications were encountered. The circuit from figure 4.1 was built, but problems occurred when testing various mechanisms. All the springs had a spring rate that was too high for the circuit's application, and therefore did not stretch at all with the magnet attached to them. This is due to many reasons.

The Force Chart	
More Force	Less Force
Smaller OD	Larger OD
Larger WD	Smaller WD
Less Coils	More Coils

Figure 4.2: Chart indicating how to modify a spring to affect the force applied to it [4]

Figure 4.2 shows how spring rate is affected by different factors related to the spring. It decreases with increasing length and increasing diameter, but too long or too large a spring would have compromised the required small size of the design. Also, the material of the tested springs was too rigid, hence only high tensions applied on the spring led to a significant extension. Therefore, the idea of using a spring was abandoned. Additional research was made to determine if enough energy could be harvested by having the magnet lying just below the induction coil in its rest position without any spring, using only the pet's neck movements to pull it up and down across the cylindrical structure.

An experiment [5] was carried on 6 dogs to measure various data related to their movements as shown in figure 4.3.

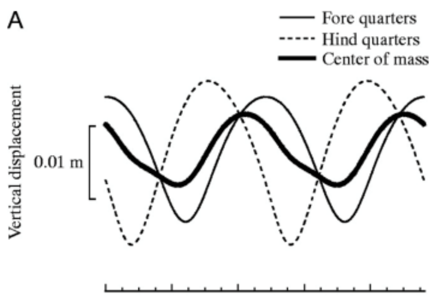

Figure 4.3: Vertical displacement of the centre of mass and hind/fore quarters for a 30-kg dog [5]

The vertical displacement of the animal was found to be very small (order of centimetres) which does not suit the requirements of the initial vertical energy harvesting system positioning. This is confirmed by the measurements of the gravitational potential energy of the centre of mass of the dog (see below). Variations are very low, which directly relates to a low variation in height ΔEp = m*g*Δh).

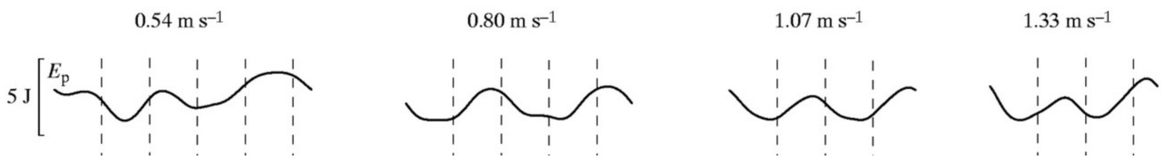

Figure 4.4: Gravitational potential energy (Ep), of the centre of mass vs time for a dog walking at 4 different speeds [5]

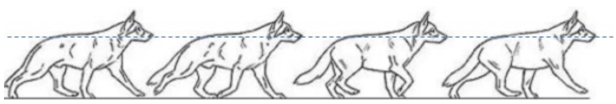

Figure 4.5: Walking pace [6]

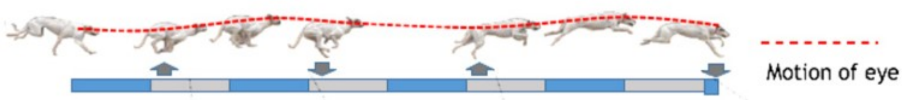

Figure 4.6: Running pace [7]

Looking qualitatively at the displacement of a dog's neck for two different paces, together with the previous observations, it was finally concluded that the final product would not be placed on the dog's neck and that the induction coil would not be vertical.

The legs were therefore taken into consideration. A study on 16 different animal species, including dogs, was realised in 1988 to investigate how speed and stride frequency change with body size [8]. Stride frequency is important as it can be directly related to the magnet's moving frequency if placed horizontally.

The figure 4.7 give the stride frequency versus speed when the dogs were trotting (expected usual pace). Even with the weights varying over a wide range, the range of measured frequency is approximately equal for each dog.

The frequency observed varied between 1 and 5 Hz for both a trotting and a galloping pace. This study is crucial to plan future tests: it shows that a decent frequency for the movement of the magnet can be obtained if the induction coil is placed horizontally on the dog's leg.

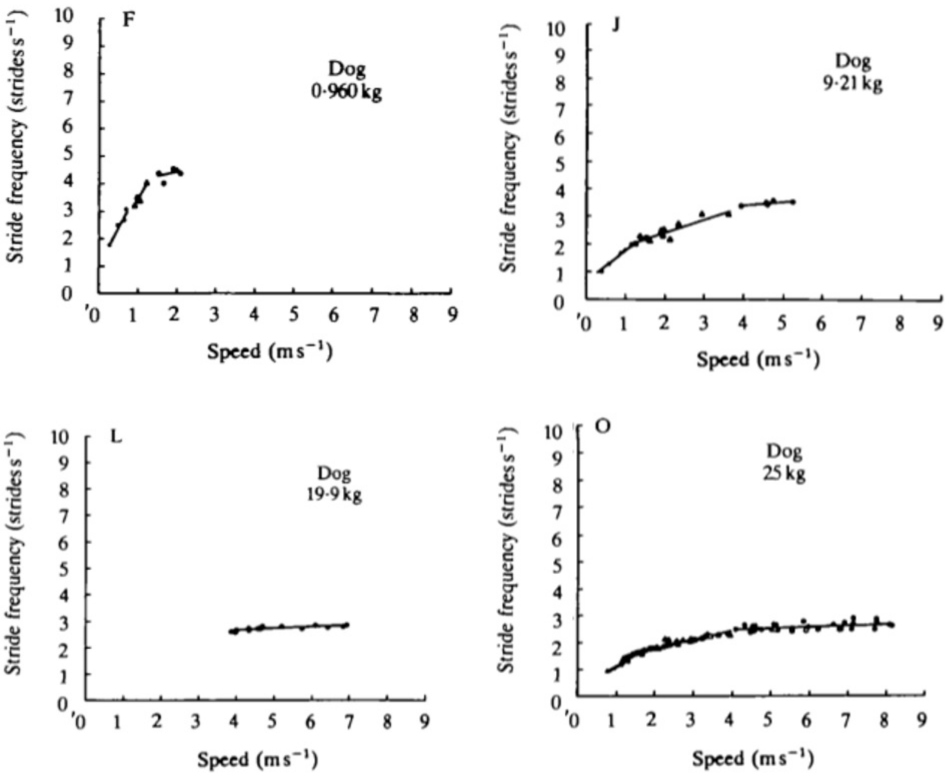

Figure 4.7: Stride frequency during trotting as a function of different speeds [8]

Therefore, such a structure will generate horizontal oscillations of the magnet, producing a time-varying magnetic field over the coil. This will in turn generate a time varying voltage across the coil. But the tracking system requires a DC voltage to operate, thus creating the need for a rectifier.

Two main designs for the Rectifier/Converter have been considered. Initially, the research was focused on a circuit that had to be entirely built. Two very similar designs of this circuit are explained in two different papers [9] [10]. The paper written by Liang Yu and Haoyu Wang [10] was chosen as the one from Suman Dwari and Leila Parsa included a few more components that were harder to obtain (PWM controller, sign detector), while both achieved the same performance.

Both circuit diagrams can be found in Appendix A, Figures A.1 and A.2.

The voltage generated by the coil is represented by an AC voltage source Vin, in figure A.2. The main idea is to manipulate the switching on and off of S1 (P-Channel MOSFET) and S2 (N-channel MOSFET), to change the path of the current iL and charge and discharge the capacitors C2 and C3, which will in turn charge C1 to a constant required output voltage. A quick components price list was established, to estimate if the implementation of this circuit could suit the budget.

However, it was difficult to find the component responsible for the switching of the MOSFETs, and not evident how to use the available charges at the output to recharge the circuit's battery. The team discussed these concerns with the project's supervisor. Looking at the time constraints and the lack of additional information on this design, it was decided that finding an existing chip that performed the same operations was a better alternative.

Such a chip became the new design the team used for rectifying and stepping up the input voltage. The block diagram can be found in Appendix B, Figure B.1. It enables to easily recharge a battery while providing sufficient energy for the GPS module. The datasheet available from the supplier gives enough information on what components should be added and where should the energy harvester be connected.

The LTC3331 chip takes in input an AC voltage, rectifies it to a DC voltage and steps it up at the output for the required 3.7V of the GPS via a Buck-Boost circuit. When enough energy is available, the chip will use it to charge the battery. When the circuit lacks energy at the output, it will draw the required amount of current from the battery.

Table 4. *Stride frequency during galloping, plotted in Fig. 1, is used to calculate least-squares fit linear regression equations relating stride frequency to running speed*

Animal	Body mass (kg)	Intercept (strides s^{-1})	Slope (strides m^{-1})	N	Speed range Minimum ($m s^{-1}$)	Speed range Maximum ($m s^{-1}$)	Minimum Stride Frequency (s^{-1})
White mouse*	0·029	6·86	0·780	3	1·10	1·54	7,718
Chipmunk	0·061	4·58	0·590	1	1·27	2·25	5,3293
Chipmunk	0·107	5·32	0·606	1	1·68	2·25	6,33808
13-lined ground squirrel	0·193	5·77	0·949	3	1·71	3·28	7,39279
White rat*	0·362	4·08	0·532	3	0·77	1·23	45,044
Dog*	0·96	3·81	0·327	1	1·35	1·91	4,25145
Suni	3·5	2·07	0·454	2	1·79	2·92	2,88266
Dog*	3·89	2·84	0·235	1	2·84	4·62	3,5074
Dik-dik	4·35	2·74	0·131	2	2·03	2·72	3,00593
Dog*	9·21	2·92	0·129	1	3·75	5·06	3,40375
Grant's gazelle	11·2	2·06	0·023	1	3·94	5·39	2,15062
Dog	19·9	2·42	0·069	1	3·86	6·94	2,68634
African domestic goat	20	2·49	0·005	1	2·75	3·39	2,50375
Fat-tailed sheep	23	2·34	0·083	2	2·28	3·56	2,52924
Dog*	25	2·27	0·050	1	4·17	8·25	2,4785
Wildebeest	98	1·68	0·073	1	2·44	5·44	1,85812
Pony	110	2·31	−0·005	1	4·33	6·31	2,28835
Waterbuck	114	1·52	0·055	3	4·56	6·33	1,5428
Pony	140	2·30	0·010	1	4·53	6·75	2,3453
Zebu cattle	160	1·71	0·064	2	3·78	5·39	1,95192
Pony	170	2·48	−0·026	1	4·97	6·33	2,35078
Donkey	170	1·80	0·097	1	5·42	7·22	2,32574
Eland	213	2·04	−0·035	1	6·11	6·67	1,82615
Horse*	680	1·81	0·019	31	5·63	11·1	1,91697

The equations are presented in the form: stride frequency = intercept + (slope × speed), where stride frequency is in strides s^{-1} and speed is in $m s^{-1}$; N is the number of animals analysed.

Figure 4.8: Stride frequency during galloping [8]

Product	Link	Quantity wanted	Min. Quantity	Unit price(pounds)	Total price(pounds)	Comments
IRLML2502 (N-Channel)	https://uk.rs-	1	5	0,402	2,01	Out of stock
	http://onecal	1	5	0,392	1,96	Next day delivery
IRLML6401 (P-Channel)	https://uk.rs-	1	5	0,224	1,12	Free next day delivery
Schottky Diode	https://uk.rs-	2	10	0,247	2,47	Free next day delivery
Tantalum capacitors 47uF	https://uk.rs	2	5	1,304	6,52	Free next day delivery
	http://www.r	2	1	0,59	0,59	
Tantalum capacitors 100uF	http://www.r	1	1	0,69	0,69	
				Min. total price	6,83	

Figure 4.9: Components price list

Hence, this chip delivers the required power to the tracking system, while charging the battery to ensure a longer period of autonomy. This will be possible as the GPS circuit will be on standby most of the time, requiring very little energy, so the harvested energy will either be delivered to the battery or dissipated (if the battery is fully charged, to prevent damaging it) by another part of the chip. However, this Linear Technology chip brought additional constraints, that are detailed below.

Amongst the challenges that were encountered, four need to be explicitly mentioned:

- Harvesting enough energy: one of the main problems with these concepts is to implement an efficient circuit subject to numerous constraints such as weight, size, cost and danger for the pet. For this particular system, getting enough energy from the pet's motion is a real concern. First, pets' physical characteristics vary considerably, resulting in a variable frequency and span of oscillation. Second, such energy is still very low, so various methods must be employed to increase it. One of the most straightforward solutions was the introduction of a spring, to enhance any displacement of the magnet from its original position, reducing the energy wasted. This idea was abandoned as the experimental solution did not meet the theoretical expectations. The best place to attach the circuit was then deduced to be the pet's legs using various research. Other ideas to improve the concept included increasing the number of turns in the coil (which would increase the resistance), increasing the size of the magnet (also increasing the weight) or increasing the length of the cylinder which could increase the energy harvested if the magnet has enough time to go through in one period of stride length.

- Many difficulties were encountered while trying to find how to implement the circuit. The primary issue was that the LTC3331 chip is surface mount, and at a size of 5x5mm, it is not possible to solder leads onto it. As a result, it will be soldered to an adapter PCB with header pins allowing it to be easily connected to a breadboard for testing or a stripboard for the final prototype. The drawback is that the adapter board is much bigger than the chip and therefore adds considerable size to the energy harvesting system. Another problem came from the important number of chip pins (32): each of them was either connected to another pin or to passive components: this would have presented difficulty in keeping track of all the required connections. This could have become a problem as operating modes of certain parts of the chip, such as the output voltage, are chosen by making connections between certain pins. Therefore, a diagram of the entire chip was made to avoid messiness and to help the implementation of the circuit (see Appendix B, Figure B.2).

- An unexpected setback also had to be faced: this concerned the ordering of the components. While suppliers offered the "Next day delivery" for most of the components, a lot of them were actually out of stock, and no such information had been provided on the ordering screen. Therefore, the EEStores had to be contacted multiple times to ensure the suppliers had the required articles in stock, and when they did not, possible alternative solutions had to rapidly be investigated. However, this greatly reduced the time available to test the whole circuit.

- Finally, fitting the right values of components to the circuit is also challenging as tests for stride frequency are difficult to realise and not perfectly conclusive. Indeed, the data collected from such tests would not be sufficient for a generalisation to be made, as pets have varying sizes and types of movement. However, it was shown in the study carried by Timothy M. Griffin, Russell P. Main and Claire T. Farley [5] that variations in frequency can be expected to lie in a certain small range for dogs of different sizes. This gives the principal orientation for the Energy Harvesting team, which will follow a strict schedule to have a working prototype ready for the oral presentation.

4.2 Tracking

The aims of development of the tracking system is to produce a device which determines its location and transmits this to the user, whilst minimising power consumption and meeting other constraints, defined in the interim report, such as size.

Hardware

The hardware of the tracking system is comprised of a GPS and a transceiver module, communicating via a microcontroller. Data from several satellites is received by the GPS module, and this transmitted by the transceiver to be accessible to the user.

Selection of specific modules

Given the goal of producing a viable and working prototype, under the strict budget and time constraints, when selecting components to purchase, ease of implementation and availability were factors sometimes chosen over lower power consumption. Due to this compromise, the ideal components may first be highlighted, then the components selected for the prototype and the reason why will be discussed in more detail.

The prices of the following components can be found in appendix D.

GPS

The L80 M39 Quectel GPS module was selected because it is very cost-effective and has several power saving modes, including a back-up mode with a power consumption of 7 μAmps (See appendix D in interim report for a detailed comparison of several GPS modules).

Transceiver

The ideal wireless communication technologies for low power devices such as this are currently Sigfox and LoRa, which are designed for Internet of Things (IoT) applications. Modules using these technologies consume extremely low amounts of power, for example, the TD1208R module from Telecom Design has a sleep-mode power consumption of 1.8 μAmps. Due to the lack of availability and other reasons provided in the interim report, despite the ultra-low power consumption, this option was declined in favour of a GPRS module.

In terms of selecting a GPRS module, the SIM800L was chosen due to its cost-effectiveness, array of useful features and widely available documentation. It also has a lower power consumption than other models in the range, such as the SIM800H [11].

Microcontroller

During prototyping an Arduino has been used to control and test each module, due to the simplicity of programming it and the useful features it provides, such as the 3.3v power supply. This vastly speeds up iterative prototyping, however, in a final product a smaller and lower-cost microcontroller will be programmed.

A wide choice of microcontrollers is available. AVR microcontrollers, such as the AT-Tiny 85 manufactured by Atmel, are well suited for these applications as they support power saving modes and are quite small. The main disadvantage of these components is the limited memory that does not allow the use of complicated libraries. Mixed signal microcontrollers, such as the MSP430F22x2 manufactured by Texas Instruments, are a viable alternative, due to the low supply voltages they support and the ultra-low power consumption. It is in the order of hundreds of μAmps in active mode when operating near the lowest voltages, and 0.7 μAmps in standby [12].

Before development an AT-Tiny 85 was selected based on these factors and previous experience working with the component making prototyping easier. Later in development it was discovered that the 8 kilobytes of memory the AT-Tiny 85 supports may be insufficient to store the program, which in testing totals roughly 15 kilobytes. An AT-Mega 16, with twice as much memory as the AT-Tiny 85, was ordered, but remains to be tested.

Mounting Method

To minimise space, ideally the whole system would be mounted on a printed circuit board (PCB), alongside the power harvesting and supply circuit. Designing a PCB is simple, and certainly this would be done in production, but in prototyping this prevents changes in design. Due to the iterative process of designing a satisfactory energy harvesting circuit, this was implausible and currently the whole system will be soldered on a prototyping board

Software

The microcontroller is connected to both the GPRS and GPS modules and must communicate bi-directionally with both, using serial data communications. Because of this, developing a program which can handle these communications reliably and effectively is vital.

This section will cover the basic connections of the system, from its inputs, the communications between and how they are outputted. The details of exactly how these communications work will be covered individually and in length, later in this report.

The only user controlled input to the system is to the GPRS. This will typically be used to wake the system. The only other input to the system is the location data that the GPS receives from the satellites, while on.

Serial communications from the microcontroller to the GPS contain PMTK commands (a command set which allows configuration of parameters such as baud rate and power mode). The GPS contains default settings and when turned on automatically begins attempting to receive location data (in NMEA protocol format) from the satellites and transmitting this to the microcontroller.

While controlling both the GPS and GPRS modules, the microcontroller also parses the location data received from the GPS into a more intuitive set of co-ordinates. This is done with a library called TinyGPS++.

AT commands (similarly to PMTK, simply a set of functions modems are designed to respond to) are sent from the microcontroller to the GPRS containing settings and commands for the GPRS to complete. Communications from the GPRS to the microcontroller contain responses, usually to those commands, but sometimes to states of the GPRS, such as upon it receiving a text.

The output of the system is the transmission of data from the GPRS to a website called “Carriots”, using the mobile data network to form a TCP connection and requesting an HTTP POST. This data is then logged on the “Carriots” website. This can be viewed directly by logging into our “Carriots” account. Data logged in “Carriots” can also be retrieved via a “GET” API request, which would ideally be done by our website, where the location provided by the response would be plotted on a map, for the user to view.

The code currently implementing this can be found in Appendix C.

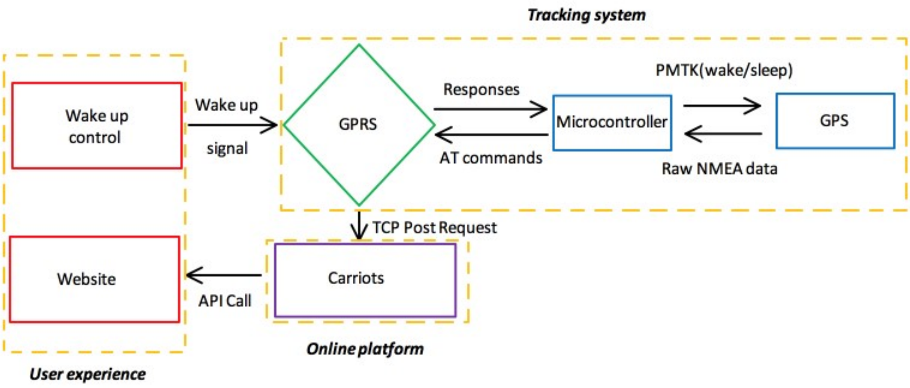

Figure 4.10: Schematic for overall data transmission of tracking system

Serial Communications

Serial communications are the basis for the transmitting and receiving of information between the modules, as both the GPS and GPRS require UART (universal asynchronous receiver/transmitter) connections. The rate at which this data is transmitted is called the baud rate and must be set correctly to transmit and receive usable information. For both the GPS and GPRS the baud rate of operation is 9600 pulses per second. UART converts data into a stream before transmission and rebuilds it after receiving. Due to this a buffer stores the data when received. To read this data it is removed from that buffer an individual bit at a time, by a FIFO (First-in First-out) method. If not read, this data remains in the buffer, so care must be taken to avoid situations in which undesired or unexpectedly received data is read within the program, usually by clearing it before issuing commands whose response must be read.

The official commands within Arduino for communicating serial in this way are only available for pins 0 and 1 on an Arduino and none of the pins on microcontroller such as the AT-Tiny. This is insufficient for communicating with both the GPS and GPRS, so a workaround called “Software Serial” is used. This requires the including of a library, which provides commands that simulate the pulses required to produce serial communication on any pin, by setting it high and then low with an appropriate speed and timing.

In early prototyping an Arduino was to allow direct inputting of AT and PMTK commands from the serial monitor directly to the GPS and GPRS modules, and relaying the responses back to the serial monitor. This was instrumental in testing as it provided quick, legible responses and feedback to commands.

GPS Location

The GPS selected uses an NMEA (National Marine Electronics Association) 0183 protocol as output, encoded in ASCII characters. When the GPS is in "fully on" mode, it begins attempting to receive this information from the satellites. Below is an example of the NMEA protocol data which must be parsed to retrieve the location coordinates. This is done by a library called "TinyGPS++", which provides useful functions to retrieve different information from these streams.

Figure 4.11: example of the NMEA protocol data [13]

An antenna is required to allow the GPS to receive information from the satellites. The GPS selected for this project contains a small patch antenna on the chip, but an external antenna can be attached. While testing the GPS with the patch antenna in the lab, it was incapable of receiving location data. Upon referring to the datasheet, it was discovered that the patch antenna requires a clear view of sky to determine its position. An external antenna may provide a stronger signal but it would consume power and this trade off would require significant testing, which has not been possible given the time constraints.

GPRS Sending

The GPRS must receive the location of the device from the microcontroller and transmit that to a receiver that the user can easily access. This is done by sending the GPRS the required AT commands and settings to allow it to connect to the "Carriots" website via a TCP connection. The GPRS must then send an HTTP POST request, containing the location data.

During initial testing, where commands were sent manually, occasionally the response "ERROR" or a "ÿ" symbol would occur. The "ÿ" and other incorrect symbols are a typical response when receiving serial data using an incorrect baud rate, however in this situation the baud rate was correctly set. The GPRS can be set to produce more useful error responses by sending "AT+CMEE=¡n¿" where n is 1 or 2. While running through the commands and responses required to ensure the correct settings for a TCP connection, a setting accessed by sending "AT+CREG" which refers to the registration with a mobile network, was not appropriately set, when queried. After referring to the data sheet, it was discovered that the GPRS required a voltage between 3.4 - 4.4v (recommended 4.0v), and while being suppled 3.3v, it would respond to commands, but infrequently produce the above errors and would not connect to the network.

Connecting: To create the TCP connection required to send the data to "Carriots" a series of parameters must be set. Some other commands such as "AT+CIPMUX=0" and "AT+CIPMODE=0" can be found in the code. These simply ensure the GPRS is in single connection and non-transparent mode, as set by default, but have been included to prevent any misconfiguration. The table below highlights the remaining parameters that must be set.

Commands	Description	Response
AT+CSTT="apn","user","pass"	Sets the access point settings	OK
AT+CIICR	Brings up wireless connection with GPRS	OK
AT+CIFSR	Get local IP address	local IP address
"AT+CIPSTART="TCP","url","port"	Start TCP connection	OK, CONNECT OK

The APN settings refer to the settings of the mobile network to connect to. The URL and port required are for the Carriots API: "api.carriots.com","80"

Sending: Once connected the data is transmitted by sending an HTTP POST request, followed by the data. To send data to the server the command "AT+CIPSEND" is sent to the GPRS, which should response with a promoting mark "¿". It is important that the program waits for this promoting mark to be received, or nothing will be sent. After receiving this, all required headers data to be sent should be transmitted, followed by "CTRL-Z" to signal the end of the data and command to send. "CTRL-Z" can be sent as ASCII code 26.

```
serialSIM.println("AT+CIPSEND");
delay(DELAY);

serialSIM.println("POST /streams HTTP/1.1"); //Post request and header data
serialSIM.println("Host: api.carriots.com");
serialSIM.println("Accept: application/json");
serialSIM.println("User-Agent: Arduino-Carriots");
serialSIM.println("Content-Type: application/json");
serialSIM.println("carriots.apikey: " + String(APIKEY)); //API key of Carriots account
serialSIM.println("Content-Length: " + String(length)); //Length of data to transmit
serialSIM.println("Connection: close");
serialSIM.println();
serialSIM.println(data);  //Actual data to be transmitted

serialSIM.write(26); //"crtl-z" to send
serialSIM.println();
```

Figure 4.12: Arduino code implementing the Carriots POST request

Arduino code implementing the Carriots POST request.

Power consumption increases significantly when connecting and sending in comparison to at idle, below are average values taken from the data sheet which correlate with experience in testing.

Mode	Average current (mA)
Idle Mode	18.7
Data Mode	212.69

GPRS Receiving

When considering functionality of the tracking system for the user, it's clear that when a pet is lost, recovering it is much easier with frequently up-to-date location information. Providing this constantly would consumer a lot of power and shorten the battery life of the system incredibly, as a result, it seemed intuitive that being able to provide this frequency of information only when necessary would be useful. This would allow the modules to be placed in a mode of reduced functionality and power consumption, when not needed, and this could vastly increase the length of the system's battery life. To control and wake the system in this way, while attached to a pet in an unknown location, requires that the GPRS receives a message of some form from the user.

Several methods for sending a message to the GPRS were considered, due to the variety of functions it provides. Either it could be set up in TCP server mode, allowing a client to connect and transmit data, or an SMS message can be sent directly to the GPRS.

The simplest method is sending an SMS message, as once received, regardless of whether it is in sleep mode, the GPRS sends a response to the microcontroller and the message is stored on the sim card. The sim card can only store 50 SMS messages, so it must be deleted to ensure there is space for future messages, using "AT+CMGD=¡n¿" where n refers to an SMS location on the SIM.

So long as the GPRS is in full functionality mode ("AT+CFUN=1") no settings are required for it to receive an SMS message. Upon receiving an SMS the GPRS will send the response "+CMTI: SM, ¡index¿" where index is the location of the new message in the SIM storage.

Code was developed to parse this response and use it as a condition for resuming the program of the microcontroller.

It is important to begin this loop only when the "+CMTI" command is received as otherwise some unexpected or unwanted serial response may begin the loop, as experienced in testing.

```
//If receives text (+CMTI) begin program
if(serialSIM.available() && CheckWake()){

  if(serialSIM.available()){    //Flush serial
    serialSIM.read();
  }
  serialSIM.println("AT+CMGD=1"); //Delete text (at location 1)

  //**** Remainder of program here ****
}
```

```
bool CheckWake(){
  String tmp;
  if(serialSIM.available()){
    delay(DELAY);
    while(serialSIM.available()){ //Store as string
      tmp = serialSIM.readString();
    }
    int fmarker = (tmp.indexOf(":")); //Mark response
    tmp = tmp.substring(0, fmarker);
    if(tmp == "\r\n+CMTI"){
      return true;
    }
  }
}
```

User Experience

The primary focus on developing the user's experience has been in producing information in an intuitive format and allowing the user a level of control over the system. Ideally in production these methods would be incorporated in a control panel on an app or website.

Displaying location:

Once sent by the GPRS the location coordinates are logged on the Carriots website. This is not a particularly professional nor intuitive way for a user to access this information. A much better way is to retrieve those coordinates and plot them on a map. This could be implemented on the product's website using a small amount of Javascript to send a "GET" request to the Carriots website to retrieve the coordinates.

```
<script>
  function UpdateMap(){
      var location = "51.498870,-0.175992";
      // An API call here would update this variable
      var x = "https://www.google.com/maps/embed/v1/place?key=AIzaSyCgfFk5qkgujECkuabgNrEkN-1BKDEyRVM&q=" +location;
      document.getElementById("demo").src=x;
  }
</script>
<button onclick=UpdateMap()>Map</button>
<iframe id="demo" width="600"  height="450"  frameborder="0" style="border:0" allowfullscreen></iframe>
```

Figure 4.13: HTML uploaded to the website

This is the HTML currently uploaded to the website which produces the map. Within the ¡script¿ ¡/script¿ markers, Javascript code displays how the coordinates can be inserted into the map. Some progress has been made on the code to produce the required API call, however time constraints have prevented this from being completed.

Controlling the system:

When the tracking system is woken via an SMS message, it could read that message and parse the data inside to change the settings within the system, such as duration of wake and interval of location transmission. While much of this code is written, time has prevented its completion.

Another possibility is applying this control to a user-friendly interface on the website. The control panel could adjust the data in a POST request (using similar Javascript as that used to send the GET request) sent from the website to "Carriots", containing the desired settings. Once received by "Carriots", this could trigger what they call "listeners" which would output a text containing this information, directly to the device. This would provide a professional experience, but has not been implement due to time constraints.

Power consumption

As highlighted in our design criteria, a long battery life is important to the feasibility and user experience of the product, and any reduction in power consumption directly impacts this.

Microcontroller power modes

There are several ways to affect the power consumption of the microcontroller, more specifically the ATmega chip that was selected. These include reducing the clock speed, disabling unused functionality and putting the microcontroller into a standby or sleep mode. [14] [15]

Due to the time constraints of the project, and adjustment of microcontroller choice, early testing has focused on the power consumption of the GPS and GPRS modules and no testing has been done on the microcontroller.

GPS power modes

The GPS has three main power modes; fully on, standby and backup. Full on Mode allows acquisition and tracking, the processes of searching for satellites and then tracking them to determine the location.

Standby mode disables these processes, therefore will not produce location data, but will consume less current. The GPS will enter standby mode when sent the PMTK command "$PMTK161,0*28". Any serial communication will wake it. When exiting standby mode, previously stored information with be used to speed up the TTFF (time to first fix).

Backup mode may reduce consumption to as low as $7/mu$A. This mode requires a different power supply circuit to the one currently implemented and so while this possible reduction in power consumption is impressive, given the time constraints, the testing and implementation of this has been left to future work.

In testing, correct time and date information is usually obtained in 20 to 50 seconds after connecting the GPS to the power supply. In the best case, these parameters are obtained almost instantly.

The time taken to converge on a location varies from 30 seconds to over 2 minutes in factory start (when first powering up the module), but is almost instant when waking up from standby mode. Due to limited resources and time, the behaviour of the module when device was or woken in a different location was not tested.

	Datasheet [16]	**Tests**
Power consumption (at 3.3V)	Aquisition: 25mA Tracking: 20mA Standby: 1.0mA	25 to 30mA 22mA 950μA
TTFF (Time to first fix)	Cold start: 35s typ. Hot start: 1s typ.	30s to >2m 1s
Accuracy	<2.5m	15m typ.

GPRS power modes

GPRS power consumption can be reduced by placing it in sleep mode or "Minimum functionality mode".

Sleep Mode

The module can be woken by incoming calls, SMS, data call from network or interrupt. By default, the module wakes itself periodically for about tens of milliseconds, to communicate with the network. This can be seen by the spikes in the below graph of a GPRS in sleep mode

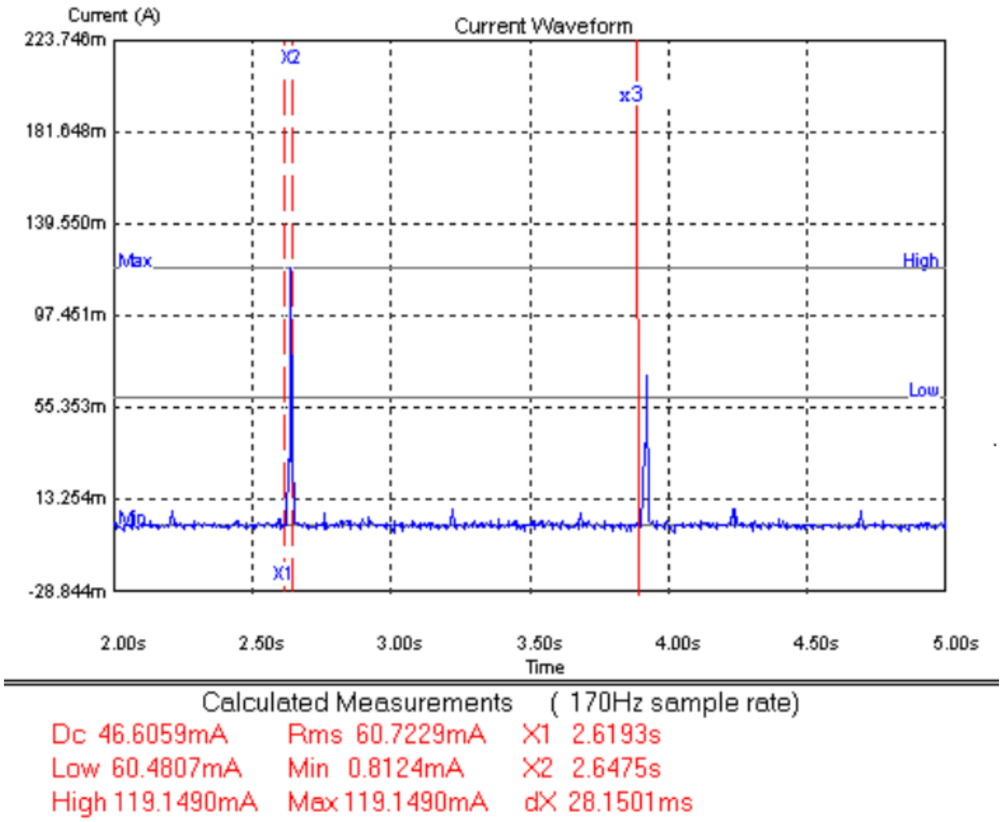

Figure 4.14: Current vs Time of a GPRS in sleep mode [17]

AT Command (sleep Mode)	**Effect**
"AT+CSCLK=0"	Turns off sleep mode
"AT+CSCLK=1"	Turns on sleep mode and disables access to serial port
"AT+CSCLK=2"	Turns on sleep mode while monitoring serial data. If no data is transferred within 5 seconds, the module sleeps. Can be woken by serial transmissions

For early revisions of the system, mode 2 will be used, as automatic sleeping and waking increases power saving quite simply. It is also useful to be able to wake the module via the serial port. When it receives serial transmissions in sleep mode, it deletes them and wakes the module, so any further commands must be delayed for at least 100ms to ensure the entire command is received and not partially deleted. Both wake commands and appropriate delays have been built into the code to make this power saving work. Perhaps after significant prototyping, once exact timings have been determined, manual control may produce more power saving.

Minimum functionality mode:

Functionality mode	**Description**	**Current consumption (mA) in Sleep mode [18]**
"AT+CFUN=0"	Minimum functionality (RF and SIM card function disabled)	0.796
"AT+CFUN=1"	Full functionality	1.02
"AT+CFUN=4"	Flight mode (RF mode disabled)	0.892

Despite the effect on reducing current consumption that adjusting the mode of functionality has we will continue to use it only in full functionality mode to ensure that signal used to wake the system, which will be an SMS, arrives.

Overall power modes

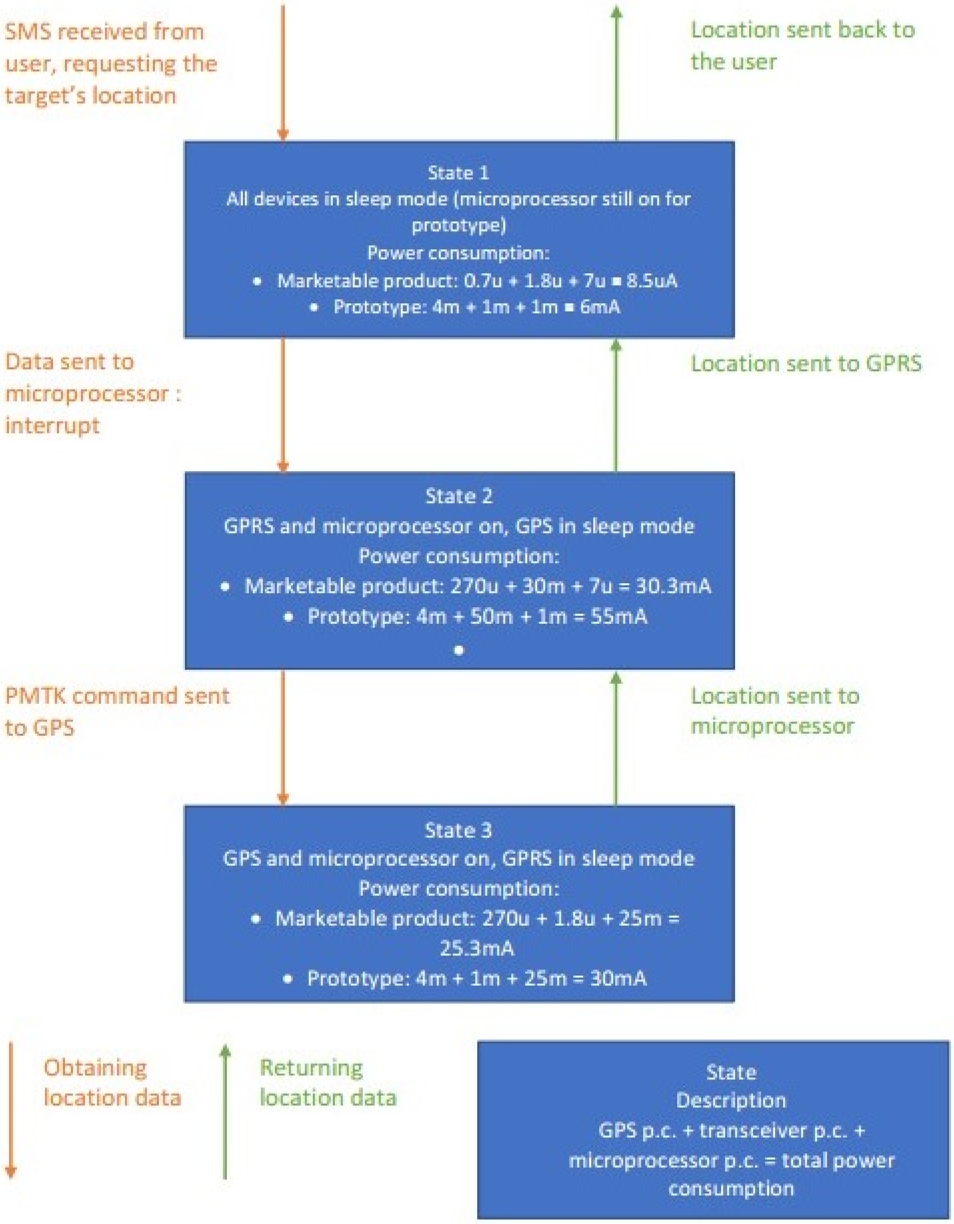

Figure 4.15: Overall Power Consumption Modes State Diagram

5 Project Management

Specific roles were assigned throughout the project to ensure constant progress and that milestones and deliverables were reached in time. The Group Leader (Alvaro Robledo) was responsible for tracking progress. The treasurer (Jacopo Bacchelli) was responsible for approving orders and ensuring budget constraints were respected. The Webmaster (Harshil Sumaria) in charge of the development of the team's website, in terms of both content and design. As mentioned in the interim report, the group was divided into 2 teams and two new roles were designated for coordination of each sub group. The Tracking team coordinator (Ryan Armiger) and the Energy Harvesting team coordinator (Valentin Gourmet) focused the development of the sub-systems on continuity and composability. Technical information and updates were shared between sub-teams to ensure the two technologies were not only fully compatible from a design point of view but also versatile and adaptable, in case unpredicted changes had to be applied to one subsystem without affecting the other.

The communication channels consisted of OneDrive which is used to keep track of the latest developments and make shared contributions easier, as well as a general group chat which is used to organise meetings, make general decisions of every member's concern and ensure fast and effective communication between both teams when needed.

For general group meetings, a library seminar room would be booked for the expected duration of the meeting. For shorter, informal or sub-team gatherings, the chosen location was generally either an empty room in the department or level 1 Electronic Laboratories if practical work was to be carried out.

The deliverables for this project are a final report, a promotional website and a prepared presentation. The planned schedule the team has been following can be found in the Gantt chart (figure 5.1).

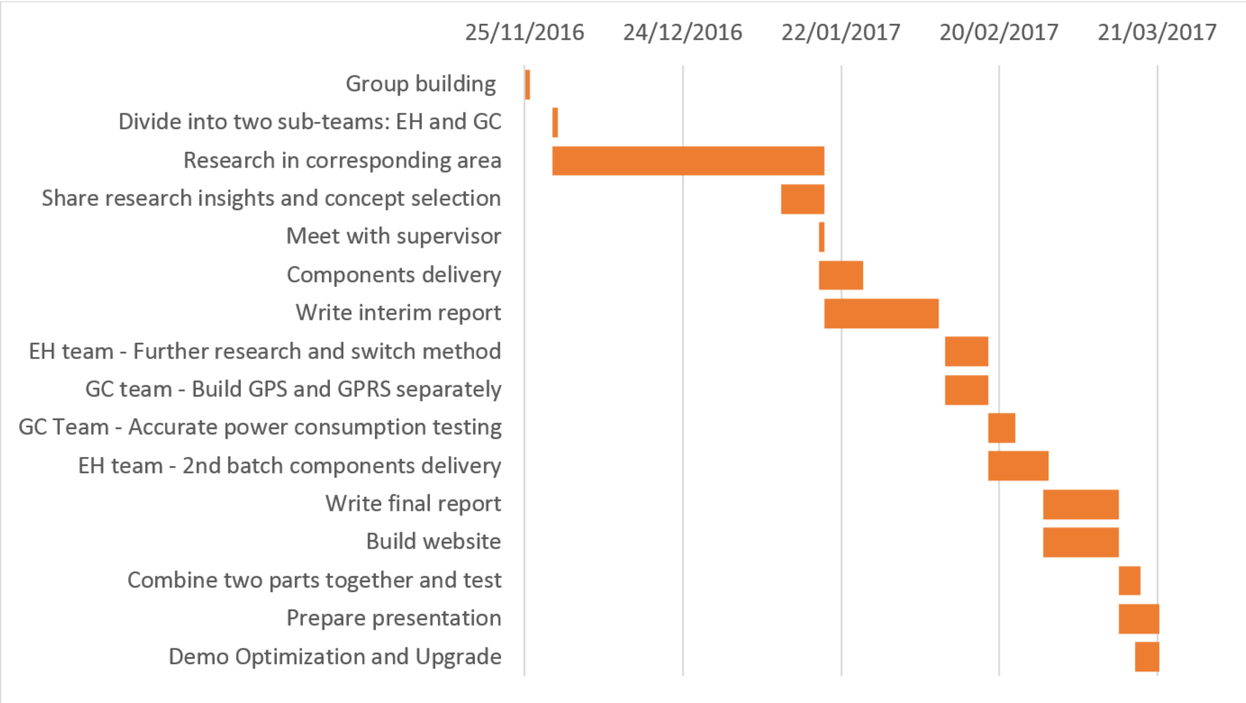

Figure 5.1: Gantt Chart

6 Industrial Design, Ergonomics and Manufacturing Considerations

The final design will be implemented on a PCB, which reduces size and improves the ease of manufacturing. The PCB will have to be designed, and then produced with components soldered onto it. A waterproof case would be developed to provide additional robustness and security to the circuit, as pets tend to walk into puddles. This would be challenging as it would require that all holes for components would need to be tightly sealed. Different colours and patterns of the product would be available for the customer, to enable them to personalize the device they acquire.

The addition of a switch would extend the life of the battery and prevent it from a deep discharge which could damage it. LED indicators could also be used to provide information on the status of the battery.

Ergonomics are concerned with the interactions between the user and the product. In this case, pet and owner are considered. For the customer, it is important that the device remains intuitive to use. For this reason, the website and a mobile application would be available to make sure the information they give ensure the device is easy to use. Their combination will allow the consumer to access some key settings and to control the pet tracker. Regarding the physical design, the product will have a clip allowing it to be easily attached to a leg band.

Key manufacturing considerations include the batch size. The components would be ordered in batches of at least 1000 units to increase the speed of manufacturing and reduce delivery costs.

The final design would be aimed at large animals. Then, similar devices would be implemented for animals of all size by changing key aspects of the product such as its size and weight, to gain access to a wider range of possible consumers. Such aspects are critical as they ensure the animal's movements are not restrained.

A Sigfox-based module would be used instead of the GPRS to reduce the amount of power consumption, as explained in part 4.

Figure 6.1 includes some pictures from the website handicappedpets.com, providing an idea of what the manufactured leg band would look like on an actual dog. This can be imagined in different, more attractive colours and patterns, and with a small box attached to it.

Figure 6.1: Visualisation of leg band attached to a dog's leg [19]

7 Future Work

In its current iteration, the device can send its position to an online platform, Carriots, via the GPRS network and some methods of reducing power consumption in the GPS and GPRS modules have been implemented. Meanwhile, the energy harvesting system can produce energy using electromagnetic and induction properties.

This section is concerned with the future improvements that would ideally be implemented by the 21st of March.

For the tracking system, the website will be developed to request the location data logged in Carriots and plot this on a map to create a more intuitive interface for the user. The hardware will then be rearranged and soldered to fit in small box and produce a prototype which fulfils the criteria of having a convenient device for the pet.

For the energy harvesting part, the energy produced will be connected to a rectifying circuit and intelligent storage system. This will be tested and improvements to component values may need to be made. The whole system will then be assembled into a more compact package in order to demonstrate the viability of size for a tracker.

As soon as the chosen chip, the LTC3331, arrives, it will be connected to the rest of the circuit. Then, the device will be simulated using:

- Various size of magnets: if enough energy is harvested with a smaller one, then it is optimal to choose it above a bigger one, to prevent energy waste.
- Various frequencies: following the above reasoning, frequencies of 1, 2, 3, 4, 5 Hz will be applied to the magnet, and the corresponding relevant data will be measured. It will give information on how much energy can be collected and therefore help the optimisation of the circuit in terms of components values, price and size.

The following cannot be implemented before the final demo and presentation, and are thus left as proposals for future improvements if the project was to be carried out for a longer time period, and if a bigger budget was allocated for its development.

Firstly, the GPRS module would be replaced by a Sigfox module. This network, as presented in the Interim Report, consumes much less power than GPRS and does not need a sim card. As was mentioned earlier, the Sigfox module could not be used in the timeframe of this project because the only component available was surface mount that required to be mounted on a PCB and other components were only available to professionals. Then, the back-up mode and the EASY technology which help to reduce the power consumption could not be implemented because it requires a power transistor.

Besides, the website created would have a section for the user to log into a personal account. From this account, a simpler interface would be set up to only show the map with the position of the pet and a few settings to ask for the position for the location or to check the battery life of the tracker for example. Such an implementation would be useful for potential pet owners that don't have an extensive knowledge of how Carriots work. An easy and practical app would also be created with the same features for users to be able to track their pet anywhere, at any time. Such interfaces would not only enhance the user's experience but would also give more flexibility to add features to the tracker in the future.

On the energy harvesting side, one important change would be to use a printed circuit board as opposed to a stripboard to connect the components to the LTC3331 chip. This would also avoid the need for an adapter to convert the chip from surface mount to through-hole. This would not only be more efficient in terms of space and avoid problems with short circuits while being less sensitive to noise, but it would also be easier to manufacture in mass.

For the generation of energy itself, if sticking to the coil method, additional fixed magnets could be added at either end of the coil to create further oscillations, increasing the magnitude of energy created.

Another useful element would be to use an LED indicator to show when the rechargeable battery is almost out of energy. This would allow the pet owner to possibly manually shake the tracker to create more energy or an emergency backup charger could be included in the product as the energy harvester itself is more to extend the battery life rather than be the sole method of charging the battery. The user could also charge their device if it was not to be used for a long period of time to ensure the battery has enough power for a cold-start of the chip.

A switch could also be added to the circuit, allowing the chip to operate in SHIP mode. In this mode, the chip is completely off with an extremely low leakage current, allowing for charge to be preserved when the tracker isn't to be used for long periods of time. This would allow the user more flexibility with their usage habits, particularly as the product does not have a charger in its current iteration.

A quite important feature to add during future work would be to make the product waterproof. This would ensure that the device could not be damaged by the weather no matter the external conditions. This could be realised through an air-tight compartment on the collar within which the entire circuitry could be contained.

Concerning the size of the hardware, it would need to be smaller to fit a wider range of pets' sizes. A box can be designed to match the users' expectations. Different colours and patterns can be offered for the box and the collar. A special option could be proposed for owners to write their pet's name on the collar and thus make the product more customisable.

8 Conclusion

In this Final report, design criteria were slightly refined, and progress in tracking and energy harvesting subsystems was presented.

The final design criteria that were retained can be summarised as follows: minimisation of power consumption, development of a constant and reliable energy generation method, and creation of an appealing product, harmless for the animal and aesthetically pleasing for the user.

Concerning the tracking system, progress has been made in implementation and technical understanding. The device includes a GPS and GPRS modules for communication with the user and satellites, as well as a microcontroller for internal data transfer and functionalities. A prototype fulfilling the main requirements has been assembled, as a convenient volume and a satisfactory power consumption were reached.

Regarding the energy harvesting circuit, in-depth research has been conducted to determine the most appropriate combination of components, energy generation method and location of the system on the animal. Particular emphasis on the latter engendered the most considerable change in design, by reimagining the product as a leg band.

The physical prototype still needs to be implemented, mainly due to unavailability of components, as the preliminary technical design has been finalised.

Time constraints prevented from focusing on aesthetics and ergonomics. An appropriate industrial design would have constituted the next step towards a hypothetical product launch.

A Step Up Converters

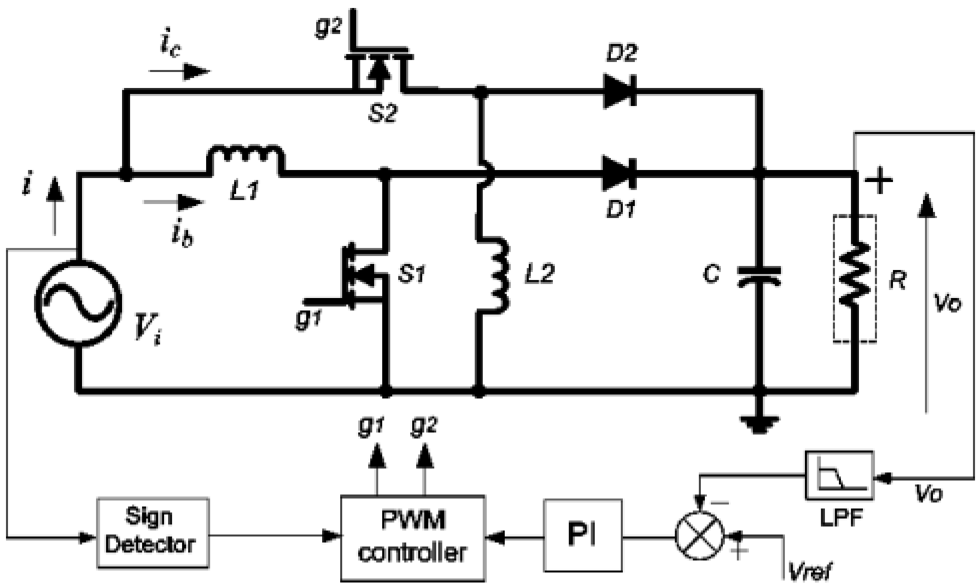

Figure A.1: AD-DC Step-up Converter [9]

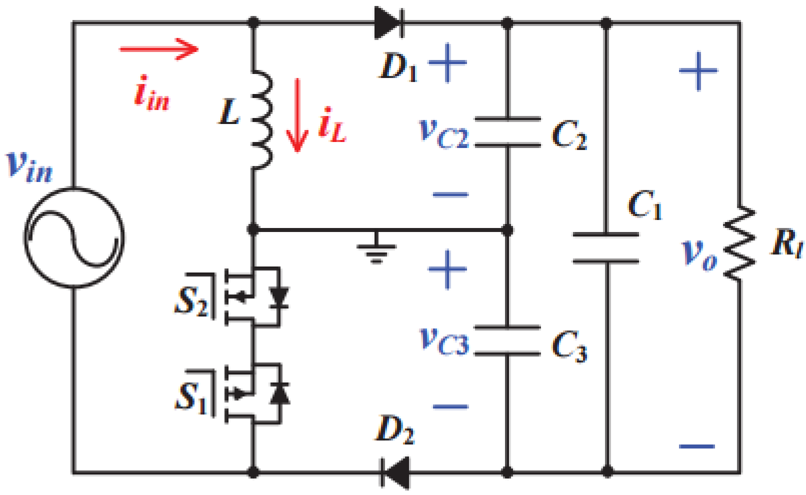

Figure A.2: Single Stage AC-DC Step-up Converter [10]

B LTC3331 chip block diagram

BLOCK DIAGRAM

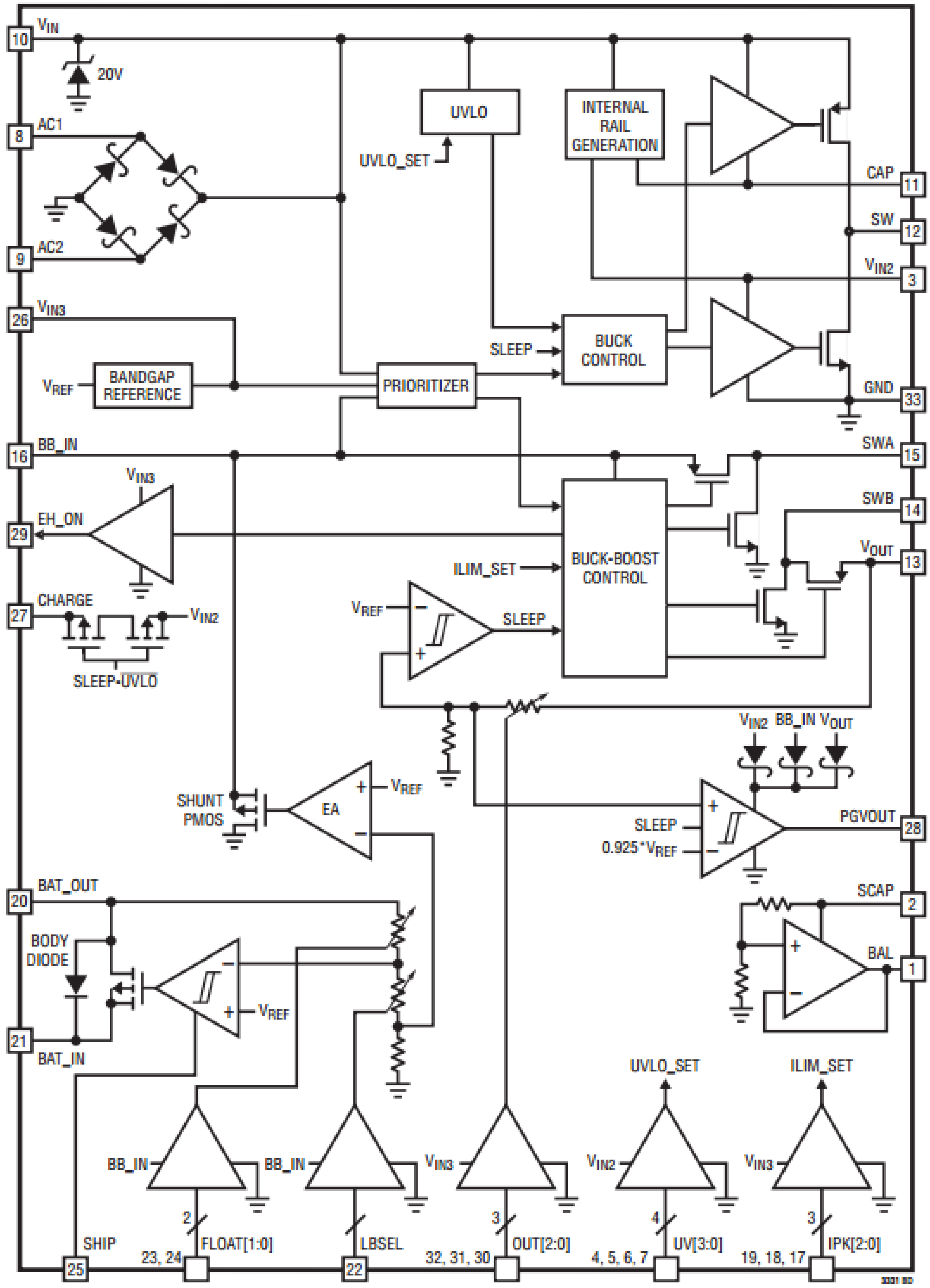

Figure B.1: Block diagram of the LTC3331 chip (used in the final circuit) [20]

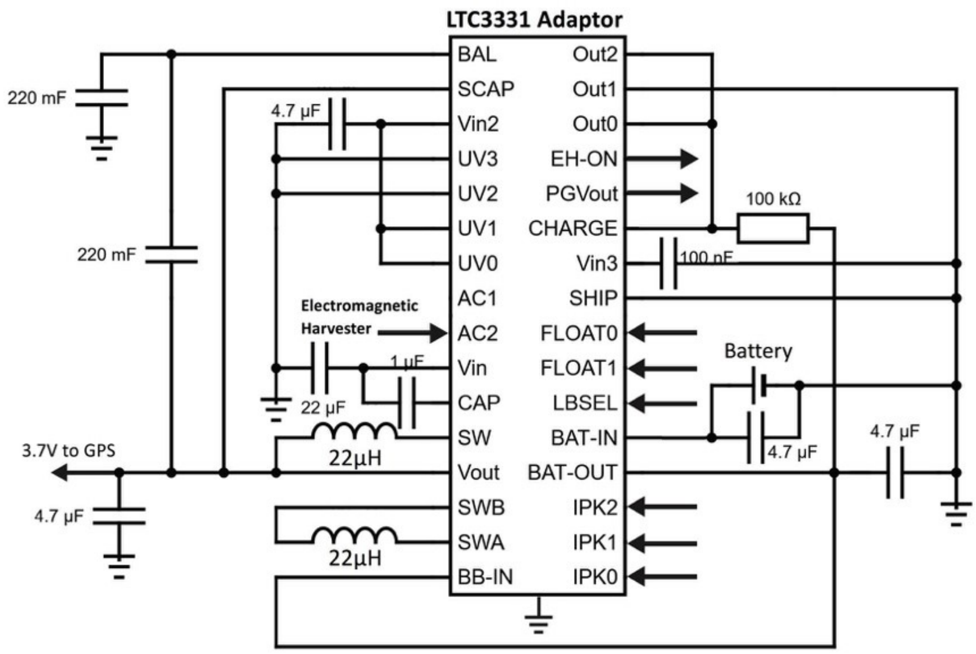

Figure B.2: Diagram of the LTC3331 chip [20]

C Tracking System Code

```
//CODE FOR WORKING SYSTEM:
#include <SPI.h>
#include <SoftwareSerial.h>
#include <TinyGPS++.h>

const String APIKEY="cd92b7ecda256728d3adafa3dc00772558b33553d8317740aadd3557af585eb3";     // Replace with your Carriots apikey
const String DEVICE="defaultDevice@PetTracker.PetTracker";     // Replace with the id_developer of your device

#define SIM_TX_PIN 8
#define SIM_RX_PIN 7
#define GPS_TX_PIN 6
#define GPS_RX_PIN 5
#define DELAY 1000

SoftwareSerial serialSIM(SIM_TX_PIN,SIM_RX_PIN);
SoftwareSerial serialGPS(GPS_TX_PIN,GPS_RX_PIN);
TinyGPSPlus gps; // TinyGPSGplus object


//SIM_command: Inserts AT commands into GPRS, checking for responses. Place above as used in setup
bool SIM_command(String commands[ ], int Array_length){
  int state = 0;
  int attempts = 0;
  int timeout = 5000;
  int reftime = millis();
  int success;

  serialSIM.println("AT+CIPSHUT"); //Ensures no connection
  //Flush serial
  if(serialSIM.available()){
    serialSIM.read();
  }

  while((millis()- reftime) < timeout){
    while(attempts < 10){

      serialSIM.println(commands[state]); //Send command
      reftime = millis(); //Start a timer
      delay(DELAY); //Allow time to respond
      CheckOK(success); //Check response received
      if(commands[state]== "AT+CIFSR"){ //Special case, doesn't respond OK
        success = 1;
      }
      if((success==1) && (state == (Array_length - 1))){ //If all commands successful  ^^^^^^^^^^^UNSURE OF SIZEOF stuff ^^^^^^^^^^^^^
        return true;
      }
      else if(success==1){ //If individual command successful
        state = state + 1;
        attempts = 0;
      }
      else{ //If command unsuccessful
        attempts = attempts + 1;
      }
    }
    reftime = 0;
  }
  serialSIM.println("AT+CIPSHUT");
  //Flush serial
  if(serialSIM.available()){
    serialSIM.read();
  }
  return false;
}

//******************SETUP*************************************************************************

void setup(){
  String SIM_setup[8] = {"AT+CIPMUX=0", "AT+CIPMODE=0", "AT+CGATT=1", "AT+CSTT=\"prepay.tesco-mobile.com\",\"tescowap\",\"password\"", "AT
  int Setup_Command_length = 8;
  Serial.begin(9600);
  while(!Serial);

```

```
  serialGPS.begin(9600);

  serialSIM.begin(9600); //Begin serial communication with Arduino and SIM
  delay(1000);
  //Set GPS to standby mode
  serialGPS.write("$PMTK161,0*28\r\n");
  delay(2000);
  //Set GPRS base settings
  while(!SIM_command(SIM_setup, Setup_Command_length)){
    delay(100);
    }

  //Flush serial
  if(serialSIM.available()){
    serialSIM.read();
  }
}

// ***********LOOP*****************************************************************************************

void loop() {
  String SIM_Connect[3] = {"AT+CIICR", "AT+CIFSR", "AT+CIPSTART=\"TCP\",\"api.carriots.com\",\"80\""};
  int Command_length = 3;
  int Wake_time = 5;
  int Ref_time = 0;
  int Interval_Ref_time = 0;
  int Retries = 0;
  int Send_interval = 30;
  double Longitude = 0;
  double Latitude = 0;
  int Text_location = 0;
  String data;
  int length;

  //If receives text (+CMTI)or (serial available)
  if(serialSIM.available() && CheckWake(Text_location)){
```

```
    Serial.println("WOKEN");
    //Flush serial
    if(serialSIM.available()){
      serialSIM.read();
    }

    serialSIM.println("AT+CMGD=1"); //Delete text
    delay(2000);
    Serial.print("Duration and interval outside");
    Serial.print(Wake_time);
    Serial.println(Send_interval);
    Wake_time = Wake_time*60000; //Convert to millis from minutes
    Send_interval = Send_interval * 1000;
    if(Send_interval < 20000){ //Set a minimum send interval
      Send_interval = 20000;
    }
    if(Wake_time < 300000){ //Set a default minimum wake time
      Wake_time = 300000;
    }
    //Wake GPS
    serialGPS.write("$PMTK000*32");
    delay(2000);
    // Start timer
    Ref_time = millis();
    while((millis()-Ref_time)<Wake_time){ // While awake less than wake time
      //Start send timer
      Interval_Ref_time = millis();
      if((millis()-Interval_Ref_time) > Send_interval){ //If interval has passed
        //Update GPS Location data
        if(get_location(Longitude, Latitude)){
          data = "{\"protocol\":\"v2\",\"device\":\""+String(DEVICE)+"\",\"at\":\"now\",\"data\":{\"Longitude\":\""+String(Longitude)+"\",
          length = data.length();
          //Reset timer now so that time to connect time doesn't affect intervals
          Interval_Ref_time = millis();
          //Connect to Carriots
          serialSIM.println("AT"); //Wake SIM
```

```
          delay(500); //Allow time to wake and delete previous input
          while(!SIM_command(SIM_Connect, Command_length) && (Retries < 10)){ //Connect
            Retries = Retries + 1;
          }
          if(Retries < 10){ //If connected
            Retries = 0;
            while((!SIM_send(data, length) && (Retries < 5))){
              Retries = Retries + 1;
              //Flush serial
              if(serialSIM.available()){
                serialSIM.read();
              }
            }
          }

        }

      }
    }

    //Returning to sleep modes
    serialSIM.println("AT+CIPSHUT"); //Ensure TCP connection closed
    delay(DELAY);
    //Flush serial
    if(serialSIM.available()){
      serialSIM.read();
    }
    serialGPS.write("$PMTK161,0*28\r\n"); //GPS to sleep
    delay(2000);
  }

}


//FUNCTIONS ***********************************************************************************************

```

```

//CheckOK: Used to check commands are successful while connecting (SIM)
int CheckOK(int& success){
    char S1=' ',S2=' ';
    success = 0;
    while(serialSIM.available()){
            S1=S2;
            S2=serialSIM.read();
            if((S1=='O')&&(S2=='K')){
              success = 1;
            }
    }
    return success;
}

bool get_location(double& Longitude, double& Latitude){
  bool Location_found = false;
  int Location_timer = millis();
  while((millis()-Location_timer) < 2000){ //Flush buffer for 2 seconds
    if(serialGPS.available()){
      serialGPS.read();
    }
  }
  Location_timer = millis();
  while((serialGPS.available()) && ((millis()-Location_timer) < 4000)){ //Error handle IF NOT FOUND^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    if(gps.encode(serialGPS.read())){ //Read it
      if(gps.location.isValid()){
        Location_found = true;
        Latitude = gps.location.lat(); //Store it
        Longitude = gps.location.lng();
      }
    }
  }
  return Location_found;
}

```

```
bool SIM_send(String data, int length){
  int sent = 0;

  serialSIM.println("AT+CIPSEND");
  delay(DELAY);

  serialSIM.println("POST /streams HTTP/1.1"); //Post request and header data
  serialSIM.println("Host: api.carriots.com");
  serialSIM.println("Accept: application/json");
  serialSIM.println("User-Agent: Arduino-Carriots");
  serialSIM.println("Content-Type: application/json");
  serialSIM.println("carriots.apikey: " + String(APIKEY));
  serialSIM.println("Content-Length: " + String(length));
  serialSIM.println("Connection: close");
  serialSIM.println();
  serialSIM.println(data);

  serialSIM.write(26); //must send a "crtl-z" to send
  serialSIM.println();

  delay(4000);
  CheckOK(sent);

  serialSIM.println("AT+CIPSHUT");
  delay(1000);
  if(sent==1){

    return true;
  }
  else{
    Serial.println("Send failed");
    return false;
  }
}

bool CheckWake(int& Text_location){
```

```
  String tmp;
  if(serialSIM.available()){
    delay(DELAY);
    while(serialSIM.available()){ //Store as string
      tmp = serialSIM.readString();
    }
    int fmarker = (tmp.indexOf(":")); //Remove response header
    tmp = tmp.substring(0, fmarker);
    if(tmp == "\r\n+CMTI"){
      return true;
    }
  }
  while(serialSIM.available()){ //Flush unused buffer
    serialSIM.read();
  }
  return false;
}

```

D Budgeting

Component	Part Number	Price (£)	Units bought	Supplier	Date
Cylindrical magnet	469-1057-ND	1,86	1	Digikey Electronics	18/01/17
Round magnet	469-1003-ND	0,53	1	Digikey Electronics	18/01/17
Round magnet	469-1002-ND	1	1	Digikey Electronics	18/01/17
Round magnet	469-1036-ND	1,59	1	Digikey Electronics	18/01/17
Microprocessor	ATTINY85-20PU	1,64	1		18/01/17
GPS Tracking Unit : Quectel L80-M39	L80-M39	9,62	1	RS	18/01/17
DIYmall SIM800L GPRS Module with PCB Antenna Bomb Slot Automatic MicroSIM Card		9,99		Amazon	28/02/17
Slim Sticker-type GSM/Cellular Quad-Band Antenna - 3dBi uFL		6,6		Amazon	22/02/17
MAGNET CYLINDRICAL NDFEB AXIAL	469-1057-N	1,86		Digikey Electronics	25/02/17
MAGNET ROUND NDFEB AXIAL	469-1003-N	0,53		Digikey Electronics	25/02/17
MAGNET ROUND NDFEB AXIAL	469-1002-N	1		Digikey Electronics	25/02/17
MAGNET ROUND NDFEB AXIAL	469-1036-N	1,59		Digikey Electronics	25/02/17
IC REG BUCK/LDO SYNC 50MA 32QFN	LTC3330IUH	6,68		Digikey Electronics	25/02/17
Conrad Energy Lithium Polymer Battery 3.7V/1000mAh (10C) with	51-1633	5,9		Rapid	28/02/17
Murata PS 22R223C 22µH ±10% Radial Leaded inductor	88-1610	1,78		Rapid	28/02/17
Murata PS 22R104C 100µH ±10% Radial Leaded inductor	88-1630	0,89		Rapid	28/02/17
Murata PS 22R474C 470µH ±10% Radial Leaded inductor	88-1650	0,89		Rapid	28/02/17
Murata PS 22R105C 1mH ±10% Radial Leaded inductor	88-1660	0,89		Rapid	28/02/17
Panasonic 0.22F Supercapacitor -20 +80% Tolerance SE Series 5.5 \	869-0765	1,25		RS	01/03/17
STRIPBOARD MEDIUM 95mm X 127mm	VB0025	1,5		EEDStores	02/03/17
LINEAR TECHNOLOGY LTC3331IUH#PBF Special Function IC, Buck-E	LTC3331IUH#PBF	14,14		Farnell	02/03/17
SCHMARTBOARD 204-0017-01 IC Adapter, 32-QFN/32-QFP to 32-D	204-0017-01	9,38		Farnell	02/03/17
QFN-32 TO DIP-36 SMT ADAPTER	IPC0020-ND	8,43		Digikey Electronics	06/03/17
ATMEL ATMEGA16-16PU 8 Bit Microcontroller		6,336		Farnell	09/03/17
		Total			
		95,876			

Bibliography

[1] J. Bacchelli, R. Armiger, V. Gourmet, et al. *Group 2 - Autonomous IoT pet tracker with energy harvesting system*. Interim Report. Imperial College London, February 8, 2017.

[2] Steve Dale. *World Pet Population Data a Mixed Bag*. June 13, 2016. URL: http://stevedalepetworld.com/world-pet-population-data-mixed-bag/.

[3] ASPCA. "How Many Pets are Lost? How Many Find Their Way Home? ASPCA Survey Has Answers". In: (June 28, 2012). URL: http://www.aspca.org/about-us/press-releases/how-many-pets-are-lost-how-many-find-their-way-home-aspca-survey-has-answers.

[4] Acxess Spring. *High Tension Springs*. 2017. URL: http://www.acxesspring.com/high-tension-springs.html.

[5] Timothy M. Griffin, Russell P. Main, and Claire T. Farley. "Biomechanics of quadrupedal walking: how do four-legged animals achieve inverted pendulum-like movements?" In: *The Company of Biologists* (July 7, 2004). URL: http://jeb.biologists.org/content/jexbio/207/20/3545.full.pdf.

[6] Linda J. Shaw. *Movement of the Working Dog*. 2017. URL: http://www.gsscc.ca/the-german-shepherd/articles-by-linda-shaw/movement-of-the-working-dog.aspx.

[7] Dr. Wada. *Mammal's Locomotion*. URL: http://mammals-locomotion.com/walking.html.

[8] Norman C. Heglund and C. Richard Taylor. "Speed, stride frequency and energy cost per Stride: How do they change with body size and gait?" In: *The Company of Biologists* (April 14, 1988). URL: http://jeb.biologists.org/content/jexbio/138/1/301.full.pdf.

[9] Suman Dwari and Leila Parsa. "An Efficient AC–DC Step-Up Converter for Low-Voltage Energy Harvesting". In: *IEEE TRANSACTIONS ON POWER ELECTRONICS* (August 2010). URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5420002&tag=1.

[10] Liang Yu and Haoyu Wang. *A Single Stage AC/DC Converter for Low Voltage Energy Harvesting*. School of Information Science and Technology, ShanghaiTech University, 2016. URL: http://ieeexplore.ieee.org/ielx7/7835398/7854636/07854782.pdf?tp&arnumber=7854782&isnumber=7854636&tag=1.

[11] Telecom Design. "TD1208R datasheet". In: (June 2016). URL: https://s3-eu-west-1.amazonaws.com/assetstdnext/download/TD120xR+Datasheet+rev1.2.pdf.

[12] Texas Instrument. "Mixed Signal Microcontroller datasheet". In: (July 2006). URL: http://www.farnell.com/datasheets/2018805.pdf?_ga=1.2482487.666264289.1488973891.

[13] Aleator777. *Intro to GPS With Microcontrollers*. 2016. URL: http://www.instructables.com/id/Intro-to-GPS-with-Microcontrollers/?ALLSTEPS.

[14] Matthew Little. *Sleep Mode on ATTINY85*. November 14, 2013. URL: http://www.re-innovation.co.uk/web12/index.php/en/blog-75/306-sleep-modes-on-attiny85.

[15] David Johnson-Davies. *ATtiny Low Power*. April 3, 2014. URL: http://www.technoblogy.com/show?KX0.

[16] Quectel. "L80 Hardware design datasheet". In: (August 10, 2013). URL: http://www.quectel.com/UploadImage/Downlad/L80_Hardware_Design_V1.1.pdf.

[17] SIMCom. "SIM800 Series$_EmbeddedAT_Sleep_Application_Note_V$1.01". In: (February 10, 2015). URL: https://cdn-shop.adafruit.com/product-files/1946/SIM800+Series+Embedded+AT+Sleep+Application+Note_V1.01.pdf.

[18] SIMCom. "Sim 80L Hardware Design Datasheet". In: (August 20, 2013). URL: http://uamper.com/products/datasheet/SIM800L_datasheet.pdf.

[19] HandicappedPets. *Walkin' Wrist Wrap (Front Only)*. 2000. URL: http://www.handicappedpets.com/walkin-wrist-wrap.

[20] Linear Technology. "LTC3331 datasheet". In: (). URL: http://cds.linear.com/docs/en/datasheet/3331fc.pdf.