



**Universidade do Minho**  
Escola de Engenharia

Ariana Mira Bezerra

**Animal location and distribution  
control for an improved management  
system**

Janeiro, 2018







**Universidade do Minho**  
Escola de Engenharia

Ariana Mira Bezerra

**Animal location and distribution  
control for an improved management  
system**

Tese de Mestrado

**Mestrado Integrado em Engenharia  
Eletrónica Industrial e de Computadores**

Trabalho efetuado sob a orientação de  
**Professor Doutor Agostinho Gil Teixeira  
Lopes**

# Declaration

Nome: Ariana Mira Bezerra

Endereço eletrónico: arianamirabezerra@gmail.com

Telefone: 910976169

Número Bilhete de Identidade:14143639

Título Dissertação: Animal location and distribution control for an improved management system

Orientador: Professor Doutor Agostinho Gil Teixeira Lopes

Ano de conclusão: 2017/2018

Mestrado Integrado em Engenharia Electrónica Industrial e Computadores

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, \_\_/ \_\_/ \_\_\_\_

Assinatura: \_\_\_\_\_



# Acknowledgements

This dissertation consecrates the culmination of my academic journey carried out over these seven years. To achieve this would not be possible without the support of those who accompanied me on this journey.

As such, I would like to thank my adviser, Professor Gil Lopes, who has given me the opportunity to develop a master's thesis in an area that I am pleased with.

The support of family and friends is always the most important since they are able to support us at all times, good or bad. To my parents, unconditional supporters who battled and fought for me without a second thought, who provided me with the best education and values. To my companion, Ricardo Roriz who supported me smiling at all times, especially in those of greater stress. To my sister who always contributes to my well being and happiness.

To all who, directly or indirectly, contributed to the realisation of this project, thank you.



## **Abstract**

This dissertation addresses the problem of animal location management and respective distribution control, with special interest on the virtual fence solution based on GPS.

As it is known, animals tend to walk wherever they can find food and safety. This can be a problem since they can interfere with other animals and people in general. To prevent unpleasant encounters physical fences were developed, but since they present high maintenance costs and some questionable efficiency, a new technology was developed: virtual fences.

In addition, since cattle and sheep graze over large areas animals are often rotated in order to achieve pasture management limitations. Virtual fences help in this matter due to the ease of changing boundaries.

The main goal was to develop a virtual fence system where animal's behaviour can be managed. To achieve this purpose, were developed a collar prototype (CP) with GPS location and GSM communication, a software application and an energetic study.

The collar tests revealed good results from GPS localisation and GSM communication, obtaining signal in open fields, house complex areas and inside constructions with different ceiling types. The collar stimuli were applied as expected, taking in consideration the collar localisation in the defined fence. Also, the system efficiency was tested resulting in the expected attribution of time periods accordingly the collar localisation area and stimulus applied. Furthermore, the speed data achieved from the GPS revealed animal behaviour important to deal with control management. The software application revealed efficiency when designing the fence boundaries. The website was considered user-friendly being easy to use, design the fence limits, register the animals and monitor the cattle.

## **Key Words**

Virtual fencing; GPS location; Animal collar; Cattle management; Animal behaviour control; Free-ranging cattle; Prototype virtual fencing device; GSM communication.



# Contents

<b>I</b>	<b>Introduction</b>	<b>1</b>
1	Introduction	3
1.1	Motivation . . . . .	5
1.2	Objectives . . . . .	6
1.3	Document Structure . . . . .	7
2	Literature Review	9
2.1	DVF system . . . . .	10
2.2	eShepherd . . . . .	11
2.3	Real Time Monitoring . . . . .	13
2.4	Discussion . . . . .	15
2.5	Rules and Regulations . . . . .	16
2.6	Theoretical Foundations . . . . .	16
2.6.1	GSM . . . . .	16
<b>II</b>	<b>Methodology</b>	<b>19</b>
3	Research and Design	21
3.1	Design system overview . . . . .	21
3.2	Design structure . . . . .	22
3.3	Design Diagrams . . . . .	23
3.4	Stimuli . . . . .	27
3.4.1	Sound . . . . .	27
3.4.2	Vibration . . . . .	28
3.4.3	Electric Shock . . . . .	28
3.4.4	Discussion . . . . .	31
4	Components	33
4.1	Raspberry Pi 3 Model B . . . . .	33
4.2	GPS . . . . .	34
4.3	GSM . . . . .	36



4.4	USB-TTL Converter . . . . .	37
4.5	Battery . . . . .	38
4.6	LEDs . . . . .	39
4.7	Broadband USB Stick . . . . .	40
<b>5</b>	<b>Software Application</b>	<b>43</b>
5.1	Graphical Environment . . . . .	43
5.1.1	Login and Register . . . . .	44
5.1.2	Define Limits . . . . .	45
5.1.3	Register Animal . . . . .	50
5.1.4	Monitor . . . . .	51
5.2	Google Maps API . . . . .	52
5.2.1	API key . . . . .	53
5.2.2	Map . . . . .	53
5.2.3	Functions and Methods . . . . .	53
5.3	Database . . . . .	54
5.4	Communication . . . . .	57
<b>6</b>	<b>Collar Prototype</b>	<b>59</b>
6.1	Hardware . . . . .	59
6.1.1	Components . . . . .	59
6.1.2	Connections . . . . .	60
6.2	Software . . . . .	62
6.2.1	Management Algorithm . . . . .	62
6.2.2	GPS . . . . .	68
6.2.3	GSM . . . . .	70
6.2.4	Fence Calculus . . . . .	72
<b>7</b>	<b>Communication</b>	<b>77</b>
7.1	SIM800L GSM module . . . . .	78
7.2	Broadband USB stick . . . . .	80
<b>8</b>	<b>Energy Study</b>	<b>83</b>
8.1	Components Choice . . . . .	83
8.2	Processing . . . . .	83
8.3	GPS Frequency . . . . .	85
8.4	Stimuli Frequency . . . . .	86
8.5	Data Exchange Frequency . . . . .	87

<b>9 Tests and Results</b>	<b>89</b>
9.1 GPS . . . . .	89
9.2 GSM . . . . .	90
9.3 Battery . . . . .	91
9.4 Virtual Fence System . . . . .	91
9.5 Data Analysis . . . . .	98
<b>10 Conclusions</b>	<b>101</b>
10.1 Conclusion . . . . .	101
10.2 Future work . . . . .	102
<b>Bibliography</b>	<b>105</b>
<b>Appendix A</b>	<b>111</b>
<b>Annex</b>	<b>119</b>



# List of Figures

1	Schematic representation of how directional virtual fencing (DVF) operates [1] . . . . .	11
2	Components of the eShepherd system [2] . . . . .	12
3	Fitting satellite Collar in elephant [3] . . . . .	14
4	GSM Architecture [4] . . . . .	17
5	System Overview . . . . .	21
6	Login and Register Sequence Diagram . . . . .	24
7	Define Limit Sequence Diagram . . . . .	25
8	Monitor Sequence Diagram . . . . .	25
9	Collar Sequence Diagram . . . . .	26
10	GSM Sequence Diagram . . . . .	27
11	Raspberry Pi 3 Model B . . . . .	33
12	GPS NEO-6M Module . . . . .	35
13	GSM SIM800L Module . . . . .	36
14	USB to serial UART converter module . . . . .	37
15	Battery . . . . .	38
16	LEDs . . . . .	39
17	Broadband . . . . .	40
18	Login Menu . . . . .	44
19	Register Menu . . . . .	45
20	Write Address . . . . .	46
21	Address Map Location . . . . .	46
22	Rectangle option to design on map . . . . .	47
23	Rectangular Field . . . . .	47
24	Circle option to design on map . . . . .	47
25	Circular Field . . . . .	48
26	Polygon option to design on map . . . . .	48
27	Polygon Field . . . . .	48
28	Line . . . . .	49
29	Line Field . . . . .	49
30	Confirm Boundaries . . . . .	49

31	Register Animal . . . . .	50
32	Register Animal - Field(s) . . . . .	51
33	Monitor . . . . .	51
34	Monitor Animal . . . . .	52
35	Database ER . . . . .	54
36	Collar Hardware connections . . . . .	60
37	Raspberry - GPS . . . . .	60
38	Raspberry - GSM . . . . .	61
39	Raspberry - LED . . . . .	61
40	Collar Class Diagram . . . . .	62
41	Processing Information Flow Chart . . . . .	66
42	Stimulus Flow Chart . . . . .	68
43	Polygon Area . . . . .	73
44	Polygon Area . . . . .	73
45	Red, Yellow and Green fences . . . . .	75
46	Communication Scheme . . . . .	79
47	GSM Gateway . . . . .	81
48	HTTP gateway . . . . .	81
49	GSM no ceiling Results . . . . .	89
50	System Overview Hardware . . . . .	92
51	Define Limit Test . . . . .	94
52	Fence Boundaries Database Data Test . . . . .	94
53	System Results 1 . . . . .	95
54	GSM System Results . . . . .	96
55	Monitor Inside Field Result . . . . .	96
56	Monitor Yellow Zone Result . . . . .	97
57	Monitor Outside Field Result . . . . .	97
58	Horse with Collar . . . . .	98
59	Horse with Collar Graphic Result . . . . .	99
60	Walk gait Graphic Result . . . . .	99
61	Trot and Gallop gaits Graphic Result . . . . .	100
62	GPS under wood ceiling Results . . . . .	115
63	GPS under plastic ceiling Results . . . . .	115
64	GPS under aluminium ceiling Results . . . . .	116
65	GPS dense forest Results . . . . .	116
66	GPS inside stone house Results . . . . .	116

67 System Results 2 . . . . . 117



# List of Tables

2	Estimated effects of 60Hz AC currents . . . . .	29
3	Mechanisms of death in electric shock drowning . . . . .	30
1	Raspberry Specifications . . . . .	34
2	GPS Specifications . . . . .	35
3	GSM Specifications . . . . .	36
4	USB/TTL Converter Specifications . . . . .	37
5	Battery Specifications . . . . .	39
6	LEDs Specifications . . . . .	40
7	TMN Broadband Characteristics . . . . .	41
8	Database Entities Relationships . . . . .	55
9	Website receive SMS frame . . . . .	57
10	Collar Prototype Hardware . . . . .	59
11	DMS frame . . . . .	64
12	RMC sentence frame . . . . .	69
13	GGA sentence frame . . . . .	70
14	GLL sentence frame . . . . .	70
15	Collar sent SMS frame . . . . .	71
16	Website send frame . . . . .	72
17	Communication Characteristics . . . . .	78
18	Three vertices fence data frame . . . . .	85
19	GPS Tests results . . . . .	90
20	GSM Tests results . . . . .	90
21	System Event . . . . .	91
22	System Test Case . . . . .	92
23	System Event . . . . .	93





# Acronyms and Abbreviations

API	Application Programming Interface
Async	Asynchronous
CP	Collar Prototype
CS	Communication Station
CTS	Clear To Send
DD	Decimal Degree
DMS	Degree Minute Second
ER	Entity–Relationship Model
GND	Ground
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IoT	Internet of Things
lat	Latitude
lng	Longitude
Mbps	Data rate - Megabit per second
P2P	Peer-to-peer
RF	Radio Frequency
RTS	Request To Send
RX	Receive
SA	Software Application
SMS	Short Message Service

SPI	Serial Peripheral Interface
SQL	Structured Query Language
Sync	Synchronous
TDMA	Time-Division Multiple Access
TTFF	Time-To-First-Fix
TTL	Transistor–Transistor Logic
TX	Transmit
UART	Universal Asynchronous Receiver- Transmitter
UML	Unified Modelling Language
USB	Universal Serial Bus
VCC	Voltage Collector Collector
VF	Virtual Fence

# **Part I**

## **Introduction**



# Chapter 1

## Introduction

This dissertation addresses the problem of animal location management and respective distribution control, surrounding about the virtual fence (VF) solution based on Global Positioning System GPS. As it is known, animals tend to walk wherever they can find food and safety. This can be a problem since they can interfere with other animals and people in general. To prevent unpleasant encounters physical fences were developed, but since they present high maintenance costs and some questionable efficiency, a new technology was developed: virtual fences.

According to Anderson [1], a virtual fencing is a method of controlling animals without ground-based fencing. They provide to animal's owner an animal management and control system as well as 24h/7 monitoring. Umstatter says that virtual fences [5] open up the possibility of managing areas that are not manageable at the moment. In 1971, the first patent about a virtual fence targeting dogs was archived according to Umstatter [5].

In this dissertation, the subject addressed emerged by the need for better animal control both at the level of herds and individual animals (pets). Other needs are the efficiency of the fences, which is intended to increase, and the cost of maintenance, which is intended to decrease. It should be emphasised that there should be careful in the choice of energy's device and the system should be easily understood by animals, Umstatter [5].

The advantages of the virtual fences can be catalogued in three modules: animal advantages, owner advantages and environmental advantages. In terms of animal advantages, since the virtual fence is a flexible fence, it can be changed according to pasture that can better feed the cattle. It also allows a much better problem management related to theft, runs or lost animals once they are monitored 24h/24h. Another advantage has to do with mustering, as it is a stressful situation that can provoke injuries on the animals.

In terms of owner advantages, like the first animal advantage, the pasture

management is much facilitated to the owner. The owner can change the fence limits according to better pasture, providing better aliment to the cattle. Another advantage consists in a permanent monitoring, that associated with others sensors can detect not only the animal location but also temperature and heart rate measurement. As for the mustering, the owner has the advantage of reducing costs with fuel, vehicles and helicopters.

As for the environmental advantages, there could be fewer damages, like in rivers where the cattle can pass and defecate polluting the water for the fishes that live there or even to the people that use that water. Since there is no physical barrier, wild life cannot get stuck on poles or wires allowing fewer injuries on wild life. The virtual fence can also prevent habitat destruction if the owner desires and delimits the fence thinking about it. In times of nidification, birds may not be able to procreate since the cattle can step on the nests or disturb the incubation of the eggs.

In comparative terms, physical mechanical and electric fences are much less flexible because once assembled, it is unlikely that the owner changes them since it implies a lot of time and work. As for the security, in physical and electric fences, animals can break poles or electric wires and then run away. However, virtual fences imply fewer maintenance costs and time. Virtual fences are much more flexible since the owner/client is allowed to change the fence boundaries as it is necessary [6].

Taking into account the advantages mentioned above and the main goal being the substitution of the physical fences to virtual fences with animals using collars with GPS tracking to be controlled, four main modules were developed. A collar prototype (CP) was built, a communication station and a software application were developed, and an energy study was made. The CP shall have a GPS tracking location and should control the animal's behaviour by stimulating with three types of stimuli: sound, vibration and electric shock. The control applied should be that if the animal trespasses the delimited zones, the stimulus is applied. As is known, a cow's movement will be dependent on the location of its herd [7]. Therefore, it may not be necessary to have a collar per animal, reducing the total cost of acquiring the fence.

The CP established communication through GSM with the computer, where the software application is transmitting all the information about the animal using the collar.

The software application with a graphic environment illustrates the field

with the virtual boundaries and animal's behaviour. The application allowed the cattle's owner to delimit the field drawing limit zones. The module responsible for the stimuli consisted on sound, vibration and finally an electric shock stimuli, being represented by LED's.

## 1.1 Motivation

Mankind has a close relationship with animals, especially herds (cows, horses, sheep, pigs, etc.). One of the ways to keep these animals ingress or egress controlled is to use fences. Conventional fences do this job, but they are not flexible and require maintenance care, being based on physical labour. According to Anderson [8], virtual fences promises to change animal management and control through cognitive rather than manual labour.

Physical fences are static tools, although they effectively control the entrances and exits of animals they fail to offer managers the flexibility they need to optimise the physiological requirements of the vegetation with the nutritional needs of foraging animals [1].

Physical fences need to be built regardless the limits size and when finished it requires a lot of maintenance either because cattle or wild animals break poles or wires, either because humans break down the fences to steal cattle or stored food.

Since the fencing costs are one of the most expensive aspects of livestock grazing, Jouven [9], VF could be the solution to the problem. According to Meyer, when building a fence, the type of fence determines the impact on cost such as cost per foot, total and annual ownership cost. Besides the initial construction costs, labour and material, owners have to determine the maintenance costs and taking into consideration the damage of the materials[4].

So like any other thing, fences had to undergo changes. Fences have evolved and in this digital age, virtual fences are being studied, built and improved. According to Anderson [8], the development of virtual fences was possible because of advancements in global navigation satellite systems.

The concept of a virtual fence was firstly patented in 1971 targeting dogs and cats, and used for the first time to control livestock in 1987, according to Umstatter [10].

VFs are non-ground-base fence method to controlling animals [1] instead,



sensors are utilised to manage animal's behaviour without compromising the safety and health of the animal.

VF has the advantage to be flexible so it can be changed according to pasture that can better feed the cattle. It also allows a better problem management related to theft, runs or lost animals once they have a monitoring 24h/7. They also imply fewer maintenance costs and time.

An adequate plan of nutrition became a fundamental task since the cattle domestication occurred. Simultaneously, a sustainable production implied a plan to manage vegetation [8]. According to Provenza [11] and Anderson [8], animal's physiology and morphology, and external factors to the animal affect the behaviour of searching for wild food resources, also known as foraging behaviour. Pasture growth is also important to create certain habitats needed for rare species. One example that Umstatter [12] gives is the plant Early Gentian and Adonis Blue butterfly.

VF easily manage the distribution of livestock, allowing areas to be unclogged and therefore food is available in a way that allows for better pasture growth without jeopardising the feeding of the entire herd or the surrounding animals or plants [13].

## 1.2 Objectives

Taking into account the above said this dissertation focus on developing a method to manage herds called virtual fence through GPS localisation.

There was a list of goals to achieve such as design and build a CP, a communication station, a software application and, finally, an energy study. The CP with GPS localisation provides a behaviour management and control using different stimuli like sound, vibration and electric shock. The communication station makes the communication necessary between the Animal Module (collar) and the User Module (software application). The software application allows the user to manage the herd as well as monitoring it. The energy study detailed solutions to reduce the CP battery power consumption since it was crucial to recharge less often.

This project was divided in five phases, including:

- Dissertation Writing with all the steps included such as the state of art, abstract, introduction, methods, results, and conclusion, as well as the

final power point presentation.

- Software Application process including the website, graphical environment interface development, boundary definition, collar detection and monitoring option.
- Collar Prototype design and building with the assembling of the GPS, GSM module and the stimuli parts: Sound, Vibration, and Electric Shock represented by LEDs. Also, the communication protocols were programmed so the communication is established.
- Communication Station development with the GSM modules adaptation to the collar's structure and computer, and programming the communication protocols.
- Energy Study where the choice of components is well thought in terms of being lower consumption so that the battery has long life without charging. Also, the GPS frequency reading is a mechanism to reduce battery consumption, as well as some algorithms about the control of the animal.

### 1.3 Document Structure

This master's thesis addresses to the cattle management problem with the intention to make fences a more viable, reliable, less expensive way to control animals. The main hypothesis of the proposed system is to track an animal with a collar and manage his behaviour applying stimulus according to the boundaries defined by the user at the software application. The main objectives of the project are the development of a Software Application that allows the user to manage herds, a Collar Prototype design and construction with GPS location and behaviour management and control through different stimuli like sound, vibration and electric shock, the Communication Station development with GSM communication, and an Energy Study.

PART I consists on the problem approach, objectives definition and literature review. In detail, Chapter 1 reveals the problem statement, the motivation and the goals defined for this project. The Chapter 2 mainly focus on the analysis of the mentioned virtual fence systems. Afterwards, in section 2.4, a discussion of the mentioned systems is made, concluding the study of the state of art. This will allow a feature overview of different systems and how one can

complement another. It is visible which systems can be adopted as well as the possibility of mixing them up creating other systems.

Once all the analysis is done the development of the virtual fence system begins, meticulously detailed in the PART II. The project begins with Research and Design Chapter 3 followed by the development of a Software Application, Chapter 5, a Collar Prototype, Chapter 6, a Communication Station, Chapter 7, and an Energy Study, Chapter 8.

The virtual fence system to be developed shall begin with the development of the software application. The website to be programmed with Google Maps API interaction, will allow the user to login in his account and posteriorly choose the address and design the boundaries of the field to fence. Several design options will be available. Next, the CP shall be built resulting on a assembly of various components and algorithm implementation for cattle control through stimuli application.

In subsection 3.4, the stimuli analysis will be approached. There will be three stimuli available: sound, vibration and electric shock, represented by LED switching. The motives for this specific stimuli choice will be later explained as well as the implications of research mentioned.

The communication station is the next step to take. To achieve it, the GSM modules will be wired, one to the CP and the other to the computer, followed by programming both modules to allow communication between collar and computer.

Finally, an energy study will be made with the intuition to improve the virtual fence system.

Results and discussion about the results can be found in the chapter 9, followed by the Conclusions, chapter10 where can be found future work suggestions. Information about AT commands and the Google Map API methods and procedures can be found in the Chapter 10.2.

## Chapter 2

# Literature Review

According to Umstatter [5], the first time a commercial virtual fence system appeared was in 1973 for pets. Later, in 1987 the containment system was used to control livestock. Since then the proof of concept using commercial and designed systems prove that virtual fences are able to maintain and manage livestock.

In 2007, according to Anderson [1], there were still no commercial virtual livestock control systems but research was breaking through the pending needs, in light to the newly available technologies, about this method to control animals.

In these study three different virtual fence systems were selected, all of them with GPS tracking location. The communication is different for the three cases. The list of virtual fence systems' examples is below.

- DVF system
- eShepherd
- RTM - Real Time Monitoring

The focus of the three systems' analysis is the type of boundary definition, the communication between the collar and remaining system (computer station), and the types of stimulus applied. In the following sections, each of the virtual fence systems will be analysed. To conclude this chapter a discussion is performed in sec. 2.4.

## 2.1 DVF system

The Directional Virtual Fencing (DVF) consist on animal's location control method. The technology is patented and trademarked.

The DVF system (Figure 1 [1]) manages the animal's location direction movement by using bilateral applied cues. If the animal attempts to cross the allowed perimeter the cues increase in severity. The perimeter is electronically generated by RF boundaries.

The technology counts on animal's innate behaviour response to reject the annoying cue or stimulus, avoiding the border location direction.

The DVF system has a bilateral cue appliance so that the animal takes the left direction when the cue is applied on the right side of the collar. On the other hand, the animal normally takes the right side when the cue is applied on the left side of the collar.

This system requires a virtual boundary and a virtual centre line. The virtual centre line is thought of as the lines that define one or more side of a polygon where the physical fence would be if it was constructed. The virtual boundary area is the area where the cues are applied to an animal using the DVF device. Both the virtual boundary and the virtual centre line are programmable.

The cues available on DVF system are audible sound and electrical stimulation. Once the cues are applied gradually, that is, if an animal is within the boundary area and remains or advances out of the field, the cue will become stronger. The cue will continue until the animal's direction of movement is at an angle of  $\geq 3^\circ$  away from the virtual centre line.

The system has an angle of approach, determined by an electronic magnetometer, in which the angle information ensures the animal receives cues on the side that will move the animal back into the allowed perimeter.

The frequency of recorded GPS data is approximately every minute while the animal is inside the allowed area. Once the animal enters the virtual boundary zone, the data is recorded every second.

One of the platform of choice for animal containment system remains as collars. However, they have challenges such as the need to the collar be tightly around the animal's neck. Otherwise, the electric stimulation may not work since the electrodes are not in constant contact with the skin. By being tightly

attached to the neck it may cause abrasion to the skin.

On the other hand, if the collar is not tight it can rotate. This means that antennas can move to non-optimal locations to capture the GPS signal and the angle approach can result in a greater jeopardising.

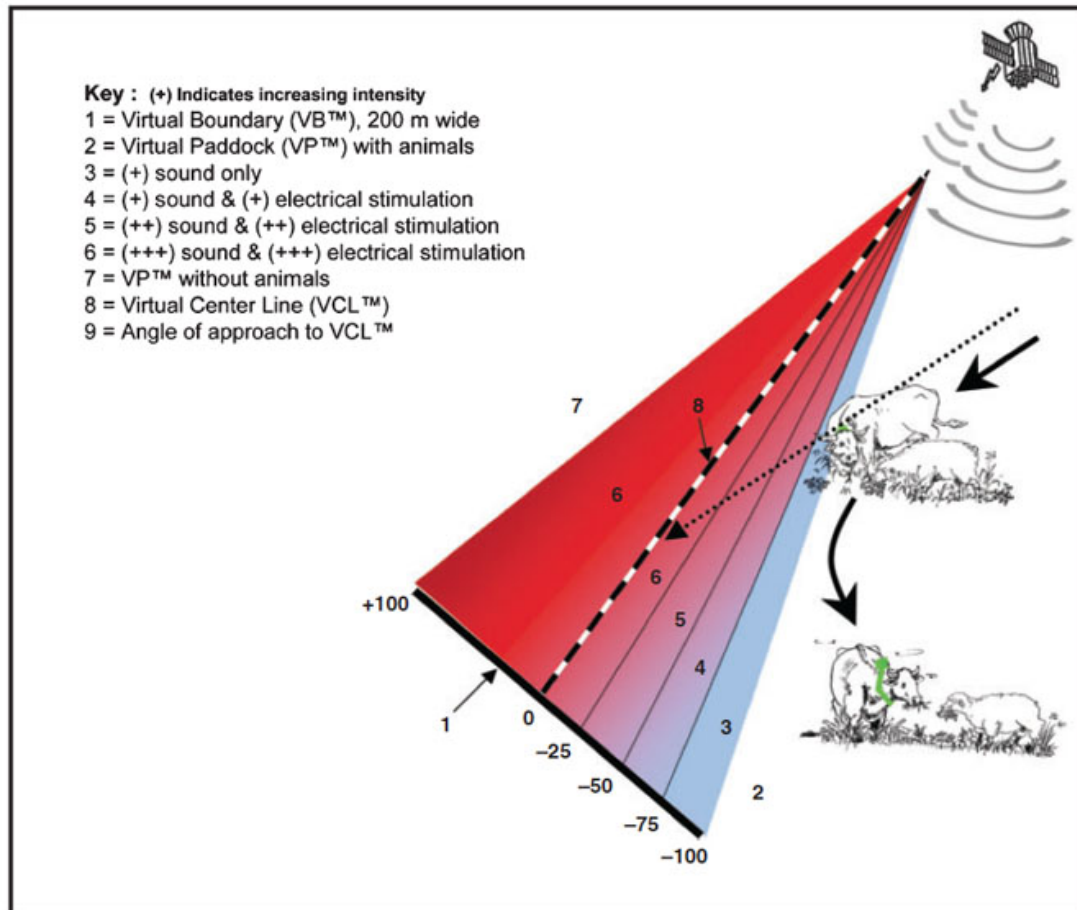


FIGURE 1. Schematic representation of how directional virtual fencing (DVF) operates [1]

## 2.2 eShepherd

The eShepherd [2] virtual livestock fencing product consists of a collar device that has audio stimuli as well as a very mild electric shock to control animal's behaviour. The product, from an existing startup from Melbourne, also has a GPS plotted fence. It is being tested on an on-farm live cattle and is a campaign to raise capital to begin commercialisation.

Currently, the eShepherd is based on all cattle in a herd having a collar.

The objective of the eShepherd virtual fence (Figure 2 [2]) is to address agricultural issues such as farmers getting older, farms getting larger, scarce skilled labour hand, rising costs, land and water sustainability. A study conducted by the Victorian government about virtual fencing be utilised to keep cattle out of waterways was the initial point to an after decision to commercialise the product.

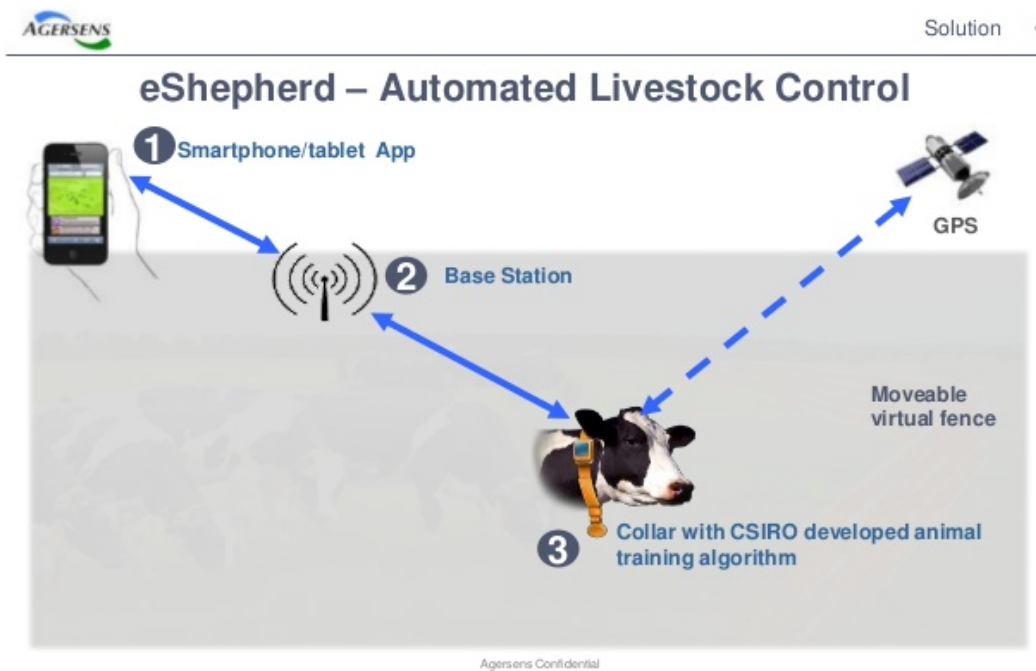


FIGURE 2. Components of the eShepherd system [2]

This device is provided with animal welfare benefits for both livestock and wildlife. The electric shock does not hurt the animal, so training the animal and responding to its behaviour is the right way to do it.

The product concept relies on a smart phone to communicate, via base station on the farm, to the internet. The base station relays instructions to the collar the animal wears and then the collar become autonomous unless the fence is changed.

The main difference of this product being a shepherd instead of a virtual fence is that it does not only establish boundaries on the field. The device keeps track of the animals and guides them along the field, managing and controlling the livestock herd.

eShepherd could be used for mustering so that injuries are avoided both in livestock and wildlife, as well as giving farmers better tools to maintain land

and water assets. The product intends to avoid labour costs without compromising the control of livestock health. Another plus is the virtual fence being half the price of a physical fence. Obtaining information from the animal is possible with the eShepherd and the collar has a solar cell to potential energy refill.

Since sheep solutions still need research the eShepherd needs financing and regulation. Some regulation is still needed because Victorian laws don't allow research on living animals, neither the commercialisation or distribution of the virtual fence collars. However, a trial of a cattle virtual fence was allowed with the approval of CSIRO animal ethics committee.

The trial has proved that the virtual fence collar is effective at keeping cattle away from non-desirable locations.

Two key challenges face by eShepherd are robustness and cost. The main cost is in the GPS power consumption and adding plastic steel rubber so that the collar is robust. The battery size, weight, capacity and price is another factor to take into account. The sound cue device almost adds no cost.

## 2.3 Real Time Monitoring

The real-time monitoring (RTM) system [3] was developed in 2004 and the first algorithm was developed in 2007. In the same year, 2007, in Kenya, a geofencing system was first implemented to solve an animal control problems. The RTM movement' analysis was implemented on wild African elephants in Samburu, Laikipia, Mt. Kenya, Chyulu Hills and Mara ecosystems of Kenya and the Kruger-Limpopo ecosystem in South Africa.

This system is focused on species that support large hardware. The software algorithms are specific to elephants' behaviour but can also be applied to another species. The RTM system applied in elephants is interesting to wildlife managers since the goal is to study crossing points and human-elephant interaction.

The RTM system (Figure 3 [3]) with the Save the Elephants, in Kenya, started to collect elephant movement data with GPS tracking collars. The collar is placed on the elephants' neck and rests on the shoulders with a counter weight at the opposite side to ensure that the device remains facing skyward. It consists of a GPS receiver, a non-volatile memory for on-board data storage, a Very High Frequency (VHF) radio beacon, and either a Global System for



Mobile (GSM) or Satellite communication module. Certain models also have a temperature sensor for recording ambient temperatures. The tracking collars were acquired from African Wildlife Tracking and Savannah Tracking.



FIGURE 3. Fitting satellite Collar in elephant [3]

RTM has enormous potential in the fields of wildlife ecology and conservation, especially for wildlife prone to frequent interactions with humans, at-risk wildlife from poaching, or for studies requiring immediate data retrieval.

Since is a real-time system it is possible to measure, process and access environmental variables including ambient temperature, humidity, light, background noise levels that can reveal information about extreme weather events, earthquakes and climatic changes. There is also the possibility to have information about the health of the animal such as telemetry of heart rate and core body temperature.

This type of system can help to understand animal' movement behaviour that can be complex behaviour patterns. These patterns can be interpreted and the state of the animal understood, and actions may be taken on manage and conservation of species.

The RTM technology uses image processing infrastructures such as Google Earth Engine. It allows near real time access to satellite image data and consequent processing and analysis.

The collars may present some down sides since some units were dropped by animals and other suffered damage. Another variable that can condition

the data acquisition is the battery life and the transmission method. GSM or satellite transmission affect power consumption since it depends on the collar' ability to obtain good connection and number of attempts to establish the communication/transmission.

Environmental variables such as temperature and humidity can affect the capability of the collar to transmit, as well as terrain conditions. Battery capacity, the method of transmission, longevity and quality of data acquisition are still factors that need further study.

## 2.4 Discussion

Taking into account the examples above mentioned, this dissertation stands on a combination between those three examples in addition to an intuitive interface, communication station and the control method.

In this project, the tracking CP shall be designed and developed according to suggestions of the examples above. The CP will have a GPS module implanted to track the animal's behaviour, three stimuli to be activated when the animal tries to cross the boundaries: sound, vibration and electric shock.

The control developed identify if the animal is inside one of the four zones *Green*, *Yellow*, *Orange* or *Red*. According to the zone where the animal is, the GPS data reading frequency will be altered. If the animal is walking inside the denominated *Green Zone*, the GPS data reading' frequency shall be one. If the animal is inside the denominated *Yellow Zone* is possible that it's approaching the fence lines, which means that the stimulus have to be applied with more time accuracy. So the GPS data acquisition' frequency shall be different, and higher. If the animal is inside the denominated *Orange Zone* means it's approaching the fence boundary and possibly will cross it. This means that the GPS data reading' frequency has to be even higher than before. This acts like a prediction system that adds more efficiency to the virtual fence.

The communication between animal' collar and the user shall be by GSM with a GSM module installed both on the Animal Module (collar) and the User Module (software application).

This communication station is important to provide a flawless communication system. This unit will be developed by adapting the GSM module in addition to programming the communication protocols.

Most of the systems developed until now have the ability to delimit the property' virtual fence boundaries by walking on the fence limit.side

One of the goals of this dissertation was to the software application interact with Google Maps API allowing the user to select/draw his property. The user became able to search the intended address and select the field(s) where the animals could walk. The application was developed with an algorithm that easily allows the user to draw the terrain limits no matter the shape. Also, the application provided a fence area suggestion according to the three stimuli application.

## **2.5 Rules and Regulations**

Ethical considerations about animal' rights had to be approached since different stimulus were plan to be used. Sound and electric shock stimuli could imply some verification of the legal laws and animal's rights. Until now, Portugal has no specific laws against experiments on animals relating to scientific techniques such as those based on virtual fences with sound and electric shock stimuli.

The directive n.º 86/609/CEE of the decree-law n.º 113/2013 [14] seeks to ensure that the number of animals used for experimental or other scientific purposes is kept to a minimum, that these animals are properly treated and that they are not inflicted with pain, suffering and distress.

When using animals for scientific purposes, the methods utilised should be likely to provide satisfactory results and cause minimal pain, suffering or distress to the animal. According to the law decree [14], there is a maximum limit of pain, suffer and distress, in which animal can not be submitted to scientific procedures. Therefore, severe procedures are prohibited.

## **2.6 Theoretical Foundations**

### **2.6.1 GSM**

GSM is a mobile communication modem developed as a digital system developed at Bell Laboratories in 1970 [15]. This digital cellular technology is utilised for mobile voice and data services transmission. GSM operates at

850 MHz, 900 MHz, 1800 MHz and 1900 MHz frequency bands. [16] TDMA technique proposes to assign different time slots to each user on the same frequency [17]. Therefore, GSM scans and reduces the data, sending it through a channel with two different streams of customer data, each in its own specific time slot [18]. It is considered the most secure telecommunication currently available although call confidentiality and subscriber secrecy are guaranteed only on the radio channel [19].

GSM communication technology adopts the following architecture, visible in figure 4 [4].

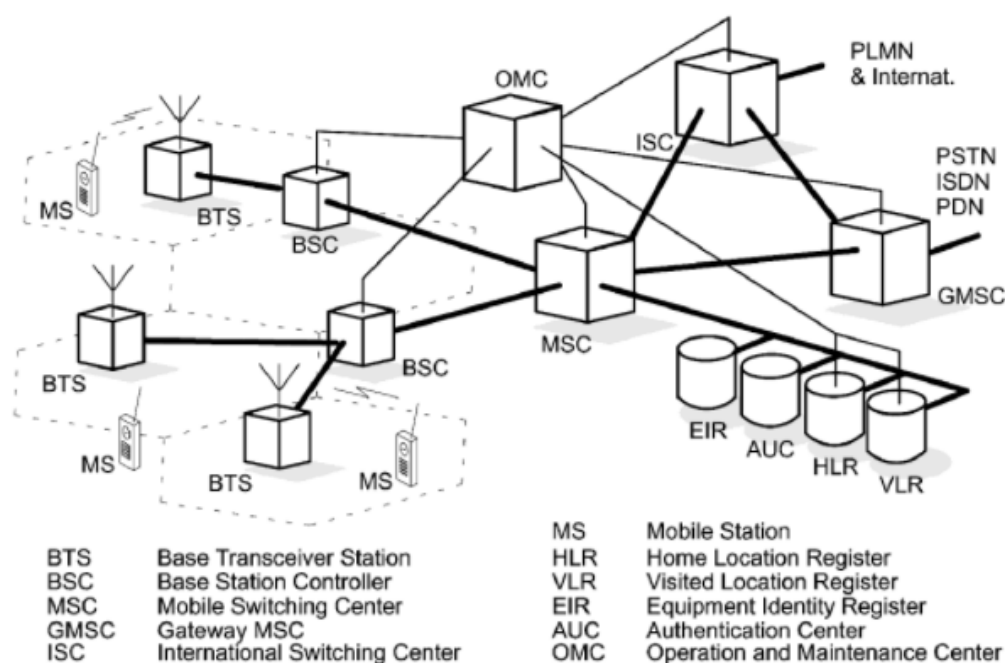


FIGURE 4. GSM Architecture [4]

The GSM network architecture is divided into three major systems: the mobile station (MS), the base station subsystem (BSS), and the Network Subsystem (NS) [20]. The MS consists of the transceiver, display and processor, controlled by the SIM card operating on network.

The BSS consists of an interface between MS and the network subsystem. It also consists of the Base Transceiver Station (BTS) and the Base Station Controller (BSC).

The radio area coverage of a base transceiver station (BTS) forms a cell. The base station controller (BSC) controls several BTS together. The BSS is formed from the BTS and BSC together. The mobile switching center (MSC) routes

the MS combined traffic in the respective cells. The dedicated gateway mobile switching center (GMSC) handles the connections starting from the fixed network or ending in the fixed network. Hierarchically, MSC is the least administrative region composed by at least one location area (LA). LA consists of various cell group, each cell group assigned to a BSC.

The network subsystem (NS) provides the basic network connection to the MSs. The NS consists of the Mobile Service Switching Centre (MSSC) provide access to different networks such as ISDN, PSTN etc. Also, NS consists of the Home Location Register (HLR) the Visitor Location Register (VLR), Authentication Center (AUC) and the Equipment Identity Register (EIR).

The HLR and VLR provide call routing and roaming. HLR is where user's temporary data and permanent data are stored. When calling to a user the HLR is required first to determine the user location. The VLR stores data from the users that belong to the group of location areas that VLR is responsible for. For faster access, the data includes a part of the user's permanent data transmitted from the HLR to the VLR. In addition, VLR can assign and store local data. The AUC generates and stores security-related data. The account which contains all the mobile equipment is maintained by the EIR. The IMEI, which stands for International Mobile Equipment Identity, is the number of each equipment mobile [19].

GSM communication module contains several features:

- Spectrum efficiency improved;
- International roaming;
- Compatibility with integrated services digital network (ISDN);
- New services support;
- SIM management;
- Fixed dialling number (FDN);
- Real time clock with alarm management;
- High-quality speech;
- Encryption for secure phone calls;
- SMS.

## **Part II**

# **Methodology**



## Chapter 3

# Research and Design

This dissertation consists of an empirical study based on experiments and practice. This type of study was the most appropriate to this dissertation since it is a research obtained by observation, induction, deduction and experiment testing [21]. There are examples similar to the current project available in the market [1] [2] [3], so it was crucial to obtain the knowledge needed to produce the experience to which this thesis proposes. Firstly, a meticulous investigation of the subject matter was carried out on the basis of archival documents and libraries related to the virtual fences theme.

### 3.1 Design system overview

Designing the whole project was the first step taken. As mentioned above, the project combines 4 central goals: build a collar prototype (CP), develop a communication station and a software application, and procedure to an energetic study (Figure 5).

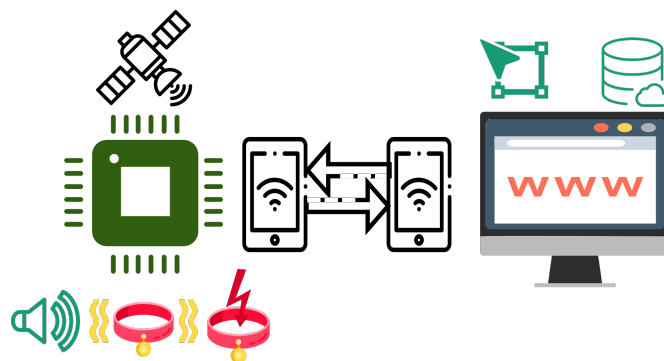


FIGURE 5. System Overview



## 3.2 Design structure

Initially, it was spent some time to carefully choose the components to the CP and CS. Once chosen, there was the need to buy some of the components. Due to project constraints, the software application was defined to be the developed first. The project design structure is demonstrated below.

- Software Application:
  - Build basic website with user login;
  - Get a Google Maps API and interact with it;
  - Develop algorithm to design the virtual boundaries;
  - Create website database;
  - Introduce boundary values into website database;
  - Make database management efficient.

The aim of this piece was to develop a website to be utilised by any user that owns this project's collar prototype. The user had to create an account to obtain access to boundaries definitions and cattle control visualisation.

- Collar Prototype:
  - Choose components;
  - Mount GPS on processor board;
  - Program processors board to receive output values from GPS;
  - Develop algorithm to compare boundaries with GPS data location;
  - Get fence boundaries values from database;
  - Mount GSM on processor board;
  - Program processor board to send SMS with data from GPS;
  - Program processors board to receive SMS with data from website database;
  - Mount LEDs to simulate stimulus output.

This piece aims were to build a CP to be put on the animal(s) to control, as well as to program an algorithm that manages the animal with the CP according to the defined fences.

- Communication Station:
  - Setup broadband to send SMS with data from website database to CP;
  - Setup broadband to receive SMS with data from GPS(CP) to website database.

The CS aim was to establish communication between the CP and the Software Application. This means, the collar obtained the boundaries data sent from the website side, and the website got the animal location data sent from the collar side.

- Energy Study:
  - Careful choice about components;
  - Efficient algorithms;
  - Most of processing made out of CP;
  - Database on website side.

This piece proposed to produce a study where energy consumption is the main point. This study was conducted in different directions such as for the component selection, the algorithms competence and where they were processed, as well as the decision to create a database on the website side.

### 3.3 Design Diagrams

Design includes schemes and diagrams to illustrate different goal achievements.

The website application includes a login and registration panel with database connection to insert and verify the user information (Figure 6).

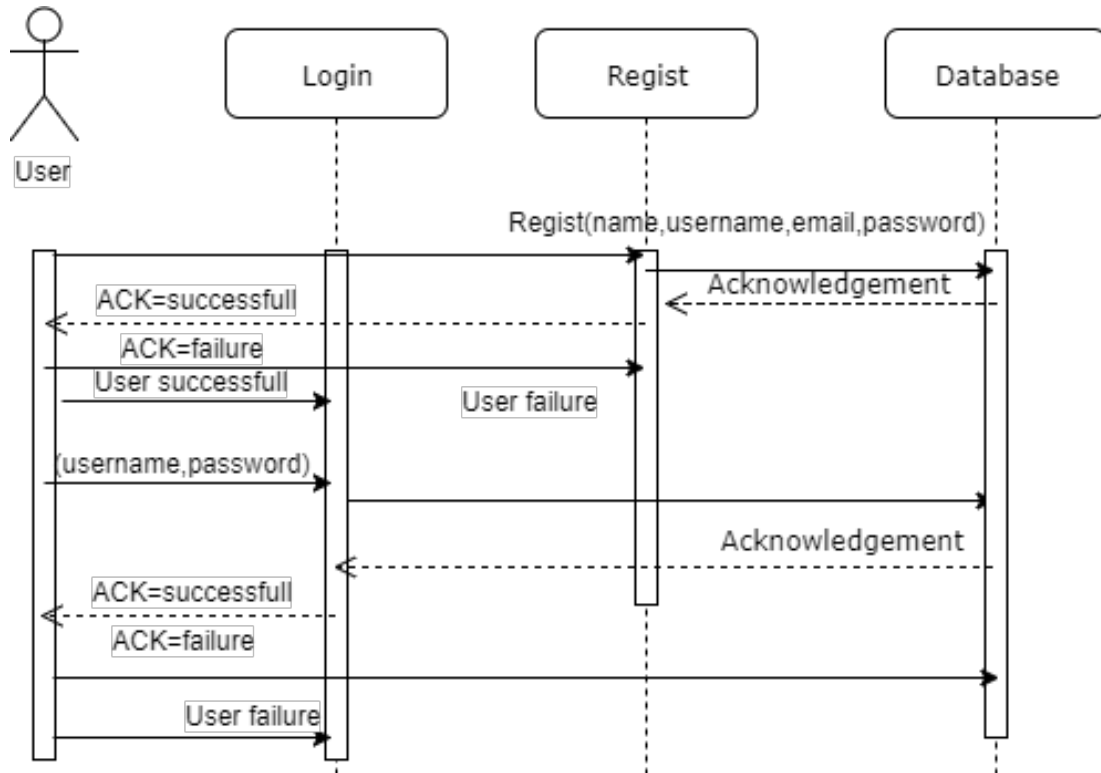


FIGURE 6. Login and Register Sequence Diagram

The SA boundary definition is an available option where the user is able to select the field shape and mark points across the map in order to design the intended field (Figure 7).

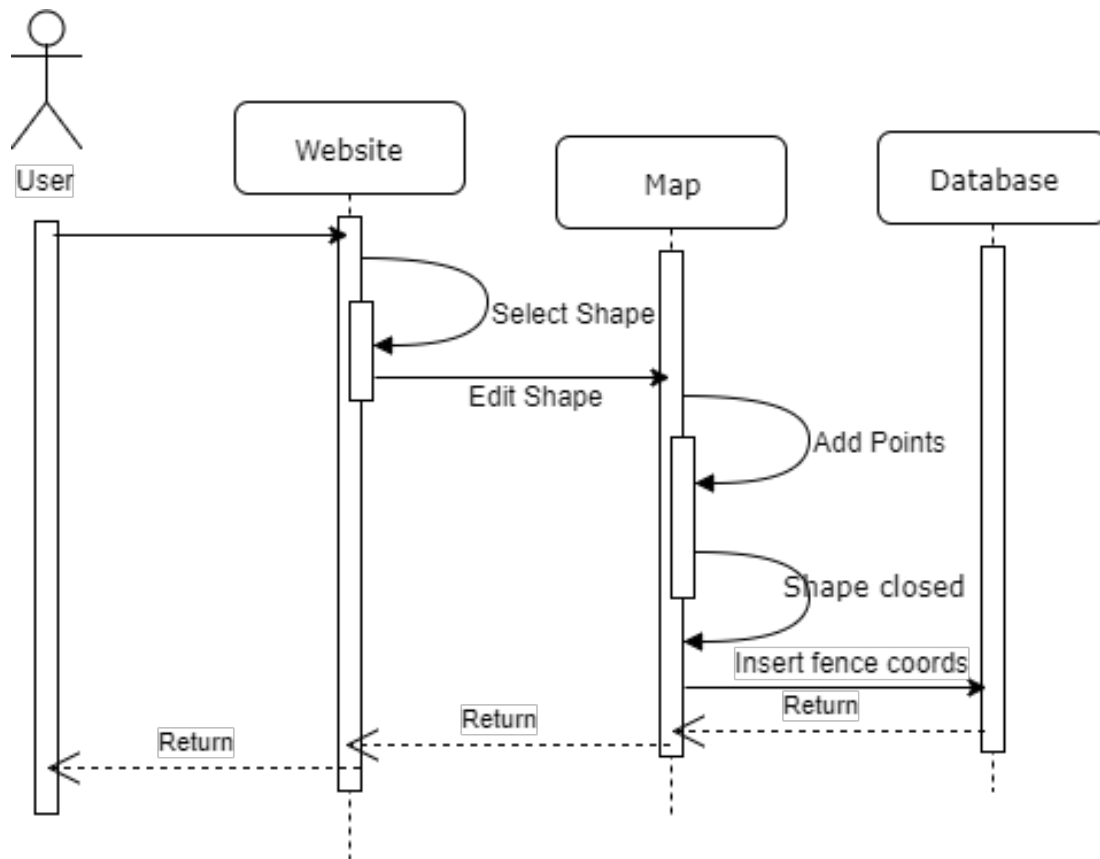


FIGURE 7. Define Limit Sequence Diagram

Monitoring cattle is also an available option. The map request the animal location to the database and the information is placed in the map (Figure 8).

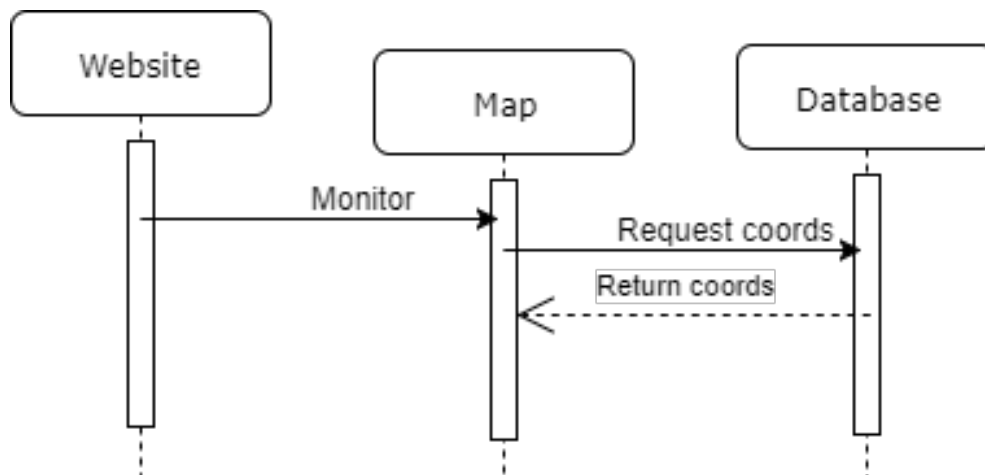


FIGURE 8. Monitor Sequence Diagram

The Collar Prototype reveals an interaction with the GSM communication module, the GPS module and the stimuli. Here the CP keeps checking if there is a new SMS to read. If yes, reads the SMS obtaining the fence coordinates. Then is requested to the GPS to give the animal coordinates. The collar calculates which area the animal is at (green, yellow or red) and applies the stimuli according to the area (Figure 9).

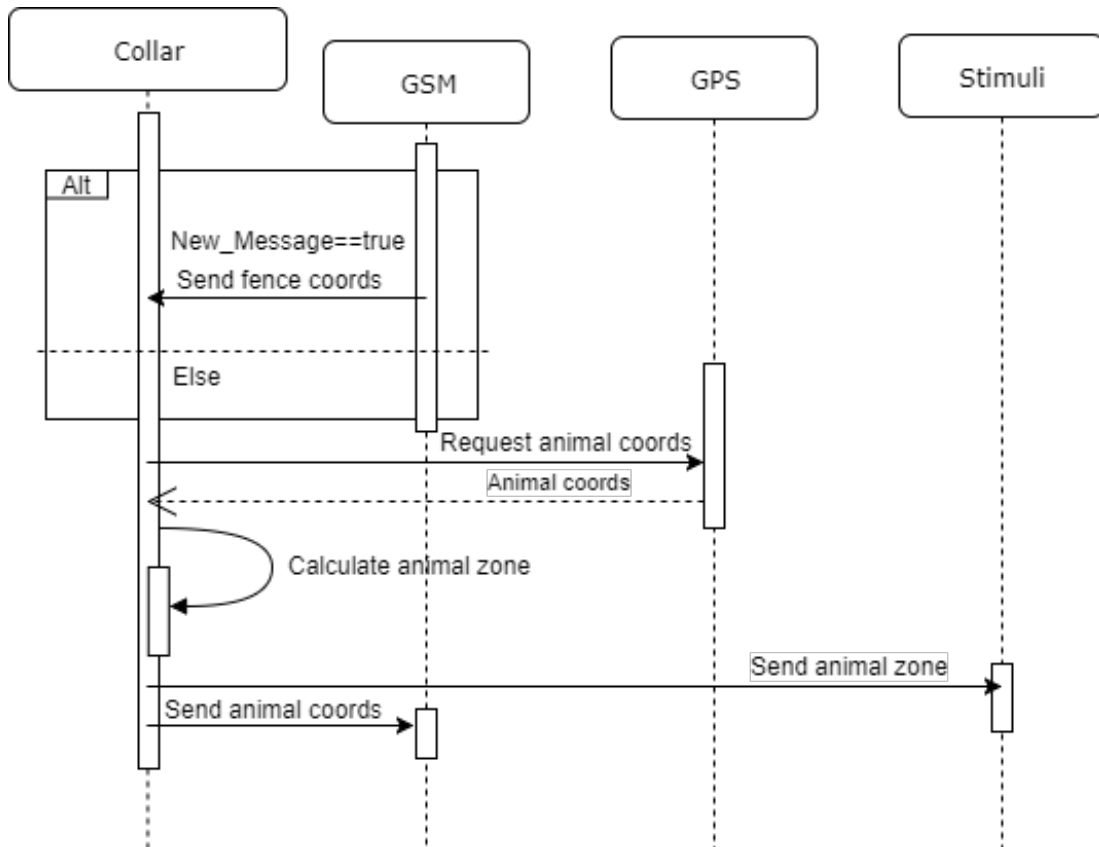


FIGURE 9. Collar Sequence Diagram

In the communication design, the GSM module from the website request the fence coordinates, stored in the database, in order to send them to the GSM collar module. On the other hand, the GSM collar module sends the animal coordinates to the GSM website module and this sends the animal location coordinates to the database (Figure 10).

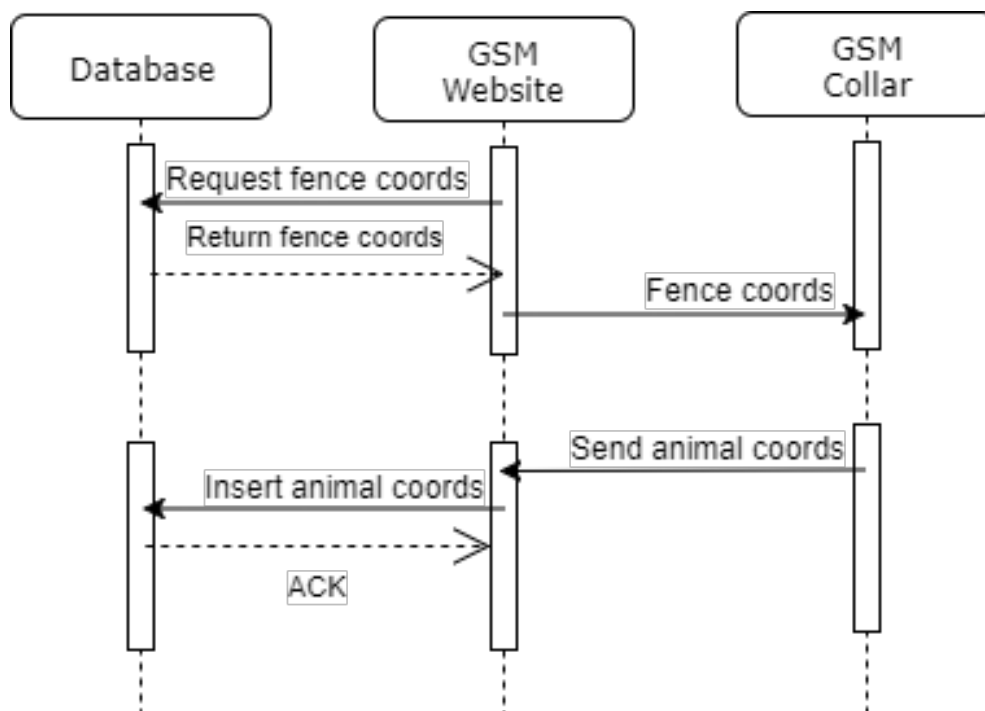


FIGURE 10. GSM Sequence Diagram

### 3.4 Stimuli

Since the animal cannot leave certain boundaries, three stimuli were studied: sound, vibration and electric shock, explained in detail below. To manage and control the animal's behaviour three types of stimulus were proposed to be applied according to the defined stimulus. The number of stimuli was elected due to the advantages that come from being three stimuli and being all different [22]. Several studies show that an animal acquires knowledge associated with some incentive, good or bad [23]. An animal, during a certain type of education training, associates a good action with food [24]. On the other hand, when receiving a negative stimulus, such as an electric fence, the animal associates this barrier with shock, which will associate with an act that should not be repeated. In short, particular stimuli elicit certain patterns of behaviour [25].

#### 3.4.1 Sound

This stimuli will be the first that the animal feels, allowing the animal to be alerted that is near a limit zone and should not cross it. It was chosen due to its harmless characteristics. According to Dr Dean Mauritz Anderson [8], if a cow ears an unpleasant and unknown sound coming from a certain direction, the

animal's tendency is to run/take the opposite direction. This means that the boundaries can be fulfilled with just the sound stimulus. It is an effective way to save energy since the other two stimuli will only be utilised in more remote border areas, considered danger areas. The sound level should be efficient without harming the animal's audition.

### 3.4.2 Vibration

The vibration allows the animal to know that he should not cross that boundary without having the unpleasant electric shock stimuli applied soon. This stimulus is probably the most important because there is a physical feeling that is efficient but does not harm the animal.

### 3.4.3 Electric Shock

The electric shock will be applied as last case scenario. The level of shock intensity used must be sufficient to elicit a reaction in the animal, but not enough to injure or create unnecessary pain or distress [26]. Electric shocks delivered in pulses provide "*shock-free-intervals*" that allow more effective escape attempts by the animal. This means that when the animal is standing at the danger limit the electric shock shall be continuous until the animal goes back into the field.

It will be taken in consideration the animal's rights exposed on Section *Implications of Research*. The important thing to notice is that there is a limit voltage to be applied so that the electric shock won't harm or kill the animal. According to studies, the human body can feel an electric shock of 1 mA (AC) or 5 mA (DC). At 10 mA (AC) occurs muscle contractions and at 30 mA (AC) or 300 mA to 500 mA (DC) debilitation is possible.

Another important aspect is the fact that water decreases the electrical resistance of skin or other tissues. The presence of urine or other sources will increase the shock intensity.

Also to take into account, an electric current delivered to a small area of skin is perceived as more aversive as the same current in a larger area. So, an animal standing on a rough surface may perceive greater shock intensity.

Some notes about the electric shock lethal characteristics can be related to current, voltage, duration, pathway or frequency. These factors in addition to

musculature, bone structure, weight and height vary the current level, Fish [27].

Air puffs, loud noises, bright lights or ultrasonic tones are an alternative for electric shock stimulus. Alternative training methods such as the use of a reward (food) for correct responses instead of punishment (electric shock) for incorrect responses [26] [28].

- Current

The higher the current more lethal the electric shock can be.

The information found at Fish' review [27] talk about effects on skin and body in general related to humans. Below, Table 2 and Table 3 show the possible effects that occur in the human body according to different current values, according to the National Institute for Occupational Safety and Health mentioned by Fish [27].

TABLE 2. Estimated effects of 60Hz AC currents

1 mA	Barely perceptible
16 mA	Maximum current an average man can grasp and "let go"
20 mA	Paralysis of respiratory muscles
100 mA	Ventricular fibrillation threshold
2 A	Cardiac standstill and internal organ damage
15/20 A	Common fuse breaker opens circuit

Alternating current and direct current electric shock cause different effects on the skin. On alternating current (AC), if the current level is high enough, there will be a feeling of electric shock as long as contact is made. With enough current, the result is uncontrolled muscles that are stimulated as rapidly as they can respond. If the shock's duration is longer than one cardiac cycle stimulation will happen [27].

On the other hand, with direct current (DC), the electric shock is only felt when the circuit is made or broken. However, if the voltage is high the electric shock may be always felt.



TABLE 3. Mechanisms of death in electric shock drowning

Mechanism	Current	Voltage(AC)
Electrical stimulation of the heart causing ventricular fibrillation	100 mA	30 V
Tetanic contraction (effectively paralysis) of the muscles of respiration	20 mA	6 V
Loss of muscle control of the extremities: 16mA for an average man	16 mA	4.8 V
Loss of muscle control of the extremities: as little as 10 mA for the most sensitive female	10 mA	3 V

- Voltage

As known by Ohm's law [29], current and voltage are proportional. This means that if the resistance is fixed, high voltage contributes indirectly to a greater risk of the electric shock be lethal due to produce high currents.

If voltage value is over about 600 volts it is considered high voltage, meaning greater current flow. In addition, a high voltage may cause dielectric breakdown at the skin which contributes to lower skin resistance allowing further increased current flow.

According to Fish [27], at 500 V or more, high resistance in the outer layer of the skin breaks down.

- Duration

The longer the animal's exposure to the electric shock, the more likely it is to the limit of current flow' safety be exceeded.

- Pathway

The path that the current follows may be lethal if the flow is through the heart. Another characteristic about the pathway is whether the current passes through the chest or head. If it does there is an increased chance of death.

- Frequency

High-frequency electric current may cause muscle spasm or tissue burning. It is not, however strong enough to cause cardiac arrest since it doesn't penetrate the body far enough.

Taking into consideration all the said above, the selected safety current and voltage shall be studied.

A simulation of the electric shock stimuli module to be developed shall be demonstrated later.

### 3.4.4 Discussion

Since the proposed study control is based in three stimuli the definition of which stimuli will act in which limit zone is already defined.

If the animal crossed the first fence, the sound cue was the elected be activated. If the animal crossed the second fence, the vibration stimulus would be activated. Finally, if the animal crossed the last fence, the electric shock would be applied.

Furthermore, whenever the animal crossed any of the fences the GPS frequency reading would increase. This means that the time period between readings would decrease, more readings per minute. If the next GPS values showed the animal had remain outside the fence the cue would be activated again. However if the animal returned to the permitted zone, the GPS frequency reading would decrease and no stimuli were applied.

Increasing the reading frequency means that the management is more controlled. This management was thought to improve the cattle control efficiency. The frequency would be increased in order to provide a real time control, meaning that the stimuli would be applied at the right moment when the animal is at or stayed at the designated area.

The stimuli application duration was carefully considered. The less negative stimuli, such as sound and vibration, were designated to have a longer duration, 10 seconds. The electric shock stimulus duration time, 5 seconds, was elected from a good response interval [1s-9s] [30]. This time interval revealed a better response from the tested animals. The response rate should decrease as the duration is increased [31]. Longer intervals are more likely to injure the animals and reduce their pressure rate for such stimulation conditions [31].

Increasing the reading frequency means more controlled management as there is a stimulation rate that allows the animal to better understand and respond to those stimuli [32]. If the frequency would to be static, then it would correspond to the RED fence GPS frequency reading, meaning the stimuli application rate would be higher because the stimuli application frequency

would also be higher. The frequency was carefully studied since studies revealed that a negative reinforcement could lead to animals incomprehension over knowledge [33]. In this particular case, if an animal attempts to cross a barrier the stimulus is applied and the animal learns that to go forward is negative and to go back is positive. But if, after the animal perceives where to go (inside the fence), another negative stimulus is applied the animal may get disoriented and even unlearn where to return.

## Chapter 4

# Components

For this dissertation development, careful choices had to be made about the components. According to knowledge obtained previously and deep research, the following components were selected for the project execution.

### 4.1 Raspberry Pi 3 Model B

This processing board (Figure 11) has low power consumption, ideal to prevent high battery consumption.

This processing board was planted on the collar CP to process all the data acquired from the GPS module as well as verify if the animal is inside or outside the fence limits and send signals so that stimuli are executed accordingly.

The Raspberry Pi B+ was chosen due to various reasons. The main reason is the lower consumption associated with great processing performance, ideal for the collar design where it was crucial to save battery. Another motive was the fact that this board contain 40 GPIO pins making it available to connect to a GPS, GSM, sound, vibration and electric shock modules. Lastly, this processing board had already been acquired and the programming was familiar to the project's developer.



FIGURE 11. Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B technical specifications [34] are encountered in the following table 1.

TABLE 1. Raspberry Specifications

Microcontroller	ARMv8-A (64/32-bit)
System-on-chip	Broadcom BCM2837
CPU	1.2 GHz 64-bit quad-core ARM Cortex-A53
Input Voltage (recommended)	5V
Input current (recommended)	2A
Power consumption	200mA
GPIO	40 pin
USB ports	4
Wireless Lan	BCM43438
Bluetooth	Low energy BLE
HDMI	Full size
Stereo output	4 Pole
Storage	MicroSDHC slot
Graphics	Broadcom VideoCore IV 300 MHz/400 MHz
Power (idle)	300mA (1.5W)
Power (maximum under stress)	1.34A (6.7W)
Clock Speed	700MHz
Length	85.60 mm
Width	56.5 mm
Weight	45 g

## 4.2 GPS

The GPS selected was the NEO-6M u-blox 6 GPS Module (Figure 12). It is a cost effective, high-performance GPS with 50 channels which provide more accuracy and less acquisition time. This GPS module, with a strict size, weight and cost, was suitable for the collar fitting.

Since the module has a voltage regulator on board there was no need to conceive one. Due to the 2 million correlators, the GPS module was able to search time-frequency spaces, enabling instant to find satellites. It is ideal for battery operated mobile devices [35].

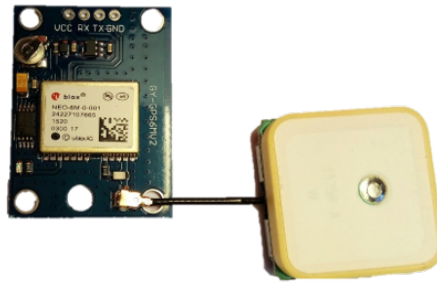


FIGURE 12. GPS NEO-6M Module

The NEO 6M technical specifications can be found in the next table 2.

TABLE 2. GPS Specifications

Channels	50
TTF	$\leq 1s$
Baud rate	9600
Voltage	3.3V-5V
Supply Current (max)	67mA
Update rate	5Hz maximum
Power consumption	39mA
Sensitivity (Tracking)	-161dBm
Sensitivity (Acquisition)	-147dBm
Update rate	max 5Hz
Accuracy	2.5m
Operating temperatures	$[-40^{\circ}C; 85^{\circ}C]$
TXD/RXD impedance	$510\Omega$
SPI timer (Initialisation)	$500\mu s$
SPI timer (Deselect)	1ms
Length	37mm
Width	25mm
Length (antenna)	26mm
Width (antenna)	35m
Weight	12g

### 4.3 GSM

The Quad-band GSM/GPRS SIM800L Module was selected to allow the communication between the collar prototype unit and the software application. <sup>1</sup> This communication module (Figure 13) was elected due to the low energy consumption, good performance, small size and weight, and low price.



FIGURE 13. GSM SIM800L Module

The SIM800L technical specifications [36] can be found in the next table 3.

TABLE 3. GSM Specifications

Frequency bands	GSM850MHz EGSM900MHz DSC1800MHz PCS1900MHz
Power Supply	3.4V - 4.4V
Peak current (TX burst)	2A
Consumption (sleep mode)	0.7mA
Consumption (full mode)	18.7mA
Baud rate	1200bps-57600bps
GPRS connectivity	multi-slot class 12
SIM interface	Support SIM card 1.8V,3V
External Antenna	Yes
Timing functions	AT command set
Length	39mm
Width	28mm
Weight	15g

<sup>1</sup>The election of a GSM communication module will be explained in detail at chapter 7

## 4.4 USB-TTL Converter

This component (Figure 14) served the purpose to extend the serial ports of Raspberry Pi, since the UART GPIO pins were not available to connect to the GSM module. This was due to the GPS module connection being established by the Raspberry UART pins.

The FTDI USB TTL Serial adapter is an USB to serial converter cable that provides connectivity between USB and serial UART interfaces [37]. FTDI FT232R device is integrated within the cable, providing access to UART transmit (TX) and receive (RX), as well as RTS, CTS, VCC and GND connections [38].

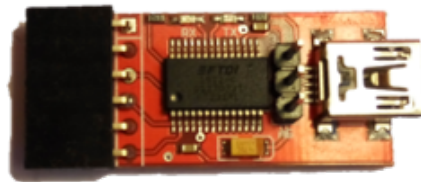


FIGURE 14. USB to serial UART converter module

The USB/TTL Serial adapter technical specifications can be found in the next table 4.

TABLE 4. USB/TTL Converter Specifications

Single Supply Operation	3.3V to 5.25V
USB	v2.0
Connector	Mini USB
Baud rate	300 to 3Mega
Receive Buffer	256 Byte
Transmit Buffer	128 Byte
Level Converter for USB I/O	3.3V
Level Converter on UART and CBUS	interfacing 5V-1.8V Logic
CMOS drive output and TTL input	5V/3.3V/2.8V/1.8V
Operating temperatures	[-40°C; 85°C]
Length	43mm
Width	17mm
Weight	10g



## 4.5 Battery

The battery selected for this prototype was a Xiaomi Pocket 10000mAh Mobile Power Bank that weights 207g (Figure 15). For this prototype, the battery had to be removed from the collar and charged, so this component selection became the most viable since it is easier to remove and charge. The power bank in question was used in the current project since it was already an acquired component and due to its technical characteristics, adequate to the needs of this project.



FIGURE 15. Battery

The Xiaomi power bank [39] automatically adjusts the output power meeting the device needs. With nine layers of world-class circuit chip protection, built by Texas Instruments and Monolithic Power Systems Inc., security is assured. The nine layers of circuit protection can include Temperature, Short-circuit, Reset, Input overvoltage, DC output over current, Overvoltage, Charge-discharge and Batteries PIC protection. Efficiency is enhanced because of the USB smart-control and charging/discharging chips adopted. The Xiaomi Pocket Mobile power bank specifications can be found in the next table 5.

TABLE 5. Battery Specifications

Single Supply Operation	3.3V to 5.25V
USB	v2.0
Connector	Mini USB
Input Voltage (DC)	5V
Input Current	2A
Output Voltage (DC)	5.1V
Output Current	2.1A
Capacity Range	7500-10000mAh
Capacity	10000mAh
Connection Type	1USB Output Interface, 1Micro USB
Battery Type	Li-ion Battery
Material	Aluminium Alloy, Metal
Discharging temperature (Initial)	[-10°C +50°C]
Discharging temperature (Continuous)	[-20°C +60°C]
Length	91mm
Width	60.4mm
Height	22mm
Weight	207g

## 4.6 LEDs

In this collar prototype the stimuli application were visible through the actuation of three LEDs (Figure 16). There were three LEDs present, green, yellow and red, which represent, respectively, the sound stimulus, the vibration stimulus and the electric shock stimulus. Their characteristics are presented due to the information about the energy that they consumed.

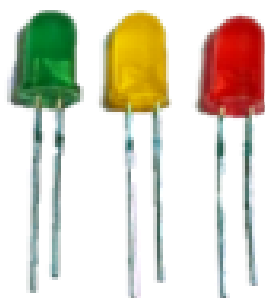


FIGURE 16. LEDs

Some characteristics [40] of the LEDs are mentioned in the table below, Table 6.

TABLE 6. LEDs Specifications

Voltage	1.8V-2.2V
Max current	20mA
Suggested current	16mA-18mA
Consumption	10mA
Cathode	Shorter Lead
Anode	Longer Lead
Intensity	Red 3,600mcd
Intensity	Green 6,700mcd
Intensity	Yellow 5,800mcd
Length	36mm
Width	5mm
Height	5mm

## 4.7 Broadband USB Stick

In this project the communication between the collar prototype and the software application was the GSM. It became necessary from the software application side, to receive information, sent from the collar localisation, and send information about the boundaries values to the collar. To serve this purpose the TMN broadband USB stick was selected since it was previously acquired (Figure 17).



FIGURE 17. Broadband

Some characteristics of the TMN broadband are mentioned in the table below, Table 7.

TABLE 7. TMN Broadband Characteristics

Installation	Plug & Play
Download	7.2Mbps
Upload	5,76Mbps
Requirements	Windows Vista, XP, 200 RAM 128MB Processor >500MHz 50MB free space



## Chapter 5

# Software Application

One of the goals of this dissertation was to develop an interactive and appealing graphical environment application. This goal serves the purpose of the user more easily define the fence boundaries, manage the cattle and monitor it. The application was thought to be user-friendly so every user could utilise it properly without advanced computer knowledge. To achieve such objective, the method chosen was to implement a website, using HTML, PHP, Javascript and CSS languages.

### 5.1 Graphical Environment

The first step taken on the website's graphical environment development was to decide its functionality. It had to have some visual organisation and colours in conformity with the theme - Virtual Fences - so the colour pallet chosen was the green. The website design was thought to be of simple implementation as well as easy perception and interaction.

The website main page would have to consist of a *Login Page* with the option to *Register* a new user in case there was not an account already created. The *Login* page should ask for an username and password to enter on the account, Fig.18. The *Register* page should ask for the user name, an username to use later on *Login* page, the user email and finally password and respective confirmation, Fig.19.

Every page included a menu bar where three options were available: *Define Limit*, *Animal Register* and *Monitor*. The main page, after the user correctly logs in, was the *Define Limit* page. There, the user had the option to search and define new limits/boundaries.

The *Animal Register* page showed a text entry box to register the animal in the desired field. The page had a data fill box and a map below to show the

field(s) available.

When selecting the *Monitor* page, was possible to see the current fence with the boundaries previously defined by the user in *Define Limit* page as well as the collar position changing according to data incoming updates.

### 5.1.1 Login and Register

Whenever the user accessed the software application website, the *Login* page was the initial page presented. In the *Login* option the user was authorised to log in to his account with the respective username and password. If the user was not registered, it was possible to create an account. In Figure 18 is possible to see the main login page. In Figure 19 is possible to see the register page.

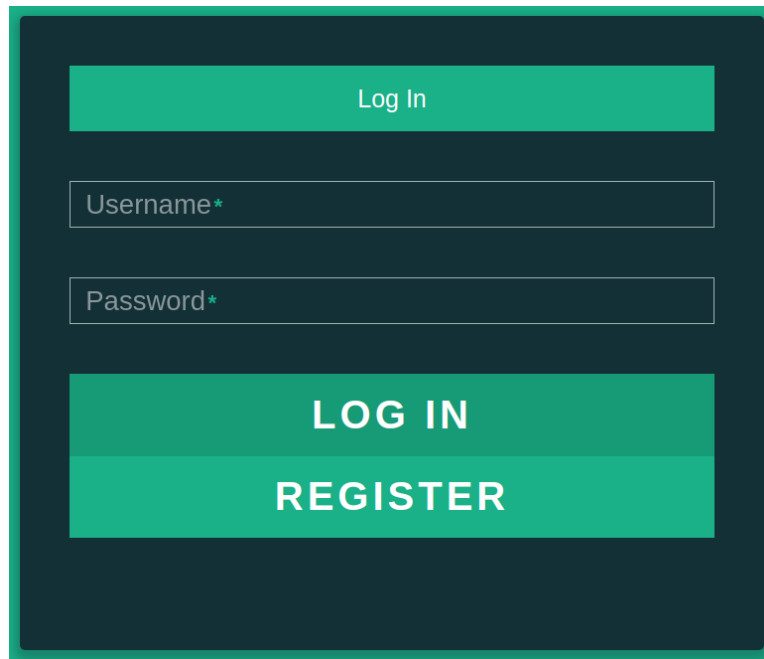
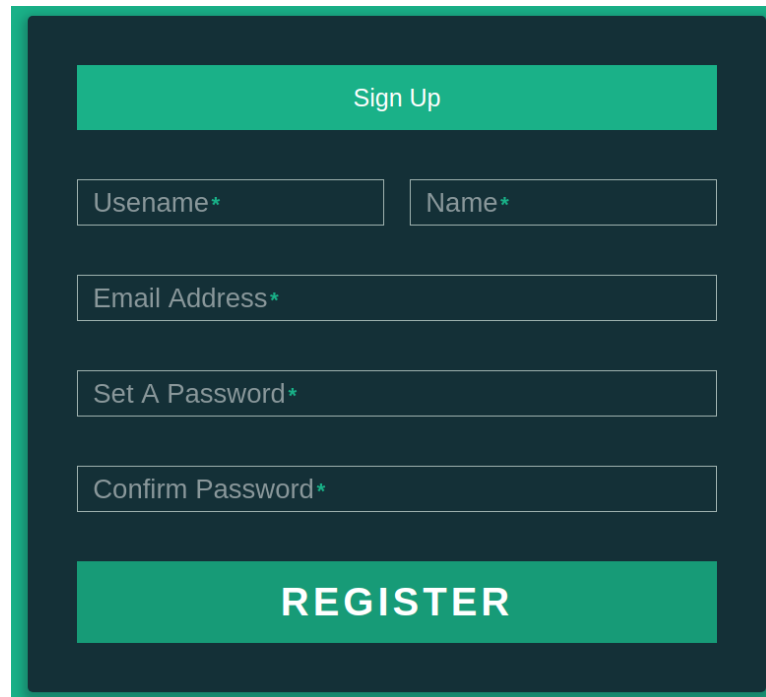
The image shows a dark-themed login menu. At the top is a teal button labeled "Log In". Below it are two input fields: "Username\*" and "Password\*", both with asterisks indicating they are required. At the bottom, there are two stacked teal buttons: the top one says "LOG IN" and the bottom one says "REGISTER".

FIGURE 18. Login Menu

To log in correctly the user needed an account, previously registered in the *Register* page. To register, there were asked personal information such as the user Name, Username, Email, Password and Password Confirmation. When creating the new account, the personal information is verified and it is checked if exists already an user with the introduced email and/or username. Also, the password is compared with the confirmation password to check if they match.



The image shows a registration form titled "Sign Up" on a dark background with teal accents. The form contains five input fields: "Username\*", "Name\*", "Email Address\*", "Set A Password\*", and "Confirm Password\*". At the bottom of the form is a large teal button labeled "REGISTER".

FIGURE 19. Register Menu

Once the account was correctly created, the user was able to log in through the *Login Page*. To log into the right account it was necessary to introduce the username and accordingly password. If the data corresponded to the account information, the user was logged in and was then able to *Define Limits*, do *Animal Register* and/or *Monitor* the herd.

### 5.1.2 Define Limits

Subsequent to the start of the user session, the page *Define Limits* appeared. The user was able to enter the address intended to define boundaries to a field. The address typed appeared in the map with a marker (house icon).

One example of how to search the address is exposed in Figure 20. In Figure 21 is visible the terrain according to the address typed, **Universidade do Minho, Azurém**.





mode will change according to the shape selected (Rectangle, Circle, Line, Polyline).

In case the user has chosen a rectangular shape, he needed to click on the rectangle of the map options as seen in Figure 22.



FIGURE 22. Rectangle option to design on map

The user then had to click on one of the edges of the field that he pretended and drag the rectangle to the other edge (Figure 23).

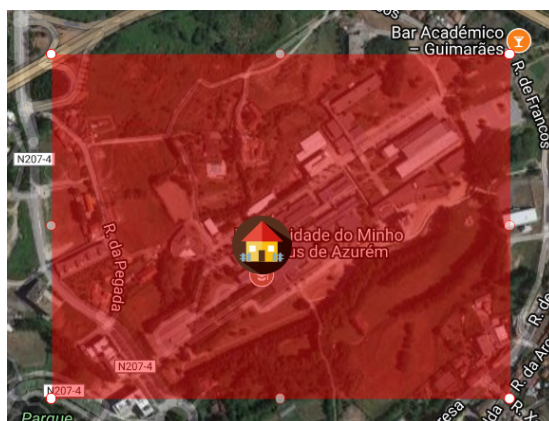


FIGURE 23. Rectangular Field

If the user intended a circular shape, he had to click on the circle of the map options as seen in Figure 24.



FIGURE 24. Circle option to design on map

The user then had to click on the center of the field that he pretended and drag the circle to the edge of the field (Figure 25).







FIGURE 28. Line

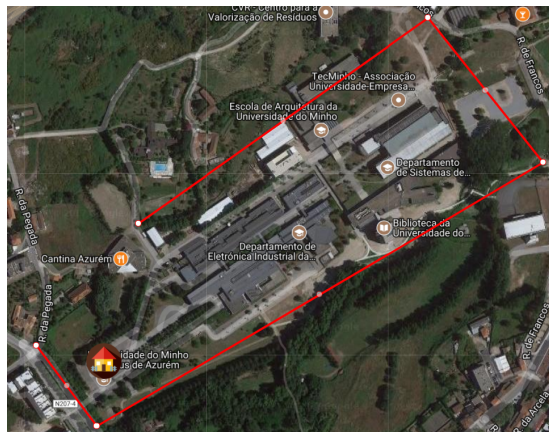


FIGURE 29. Line Field

After the boundaries design the user should click in the designed shape to confirm the boundary definition. When clicking the coloured shape, a pop-up window appeared to ask if the fence boundaries are the previously designed or to cancel (Figure 30).

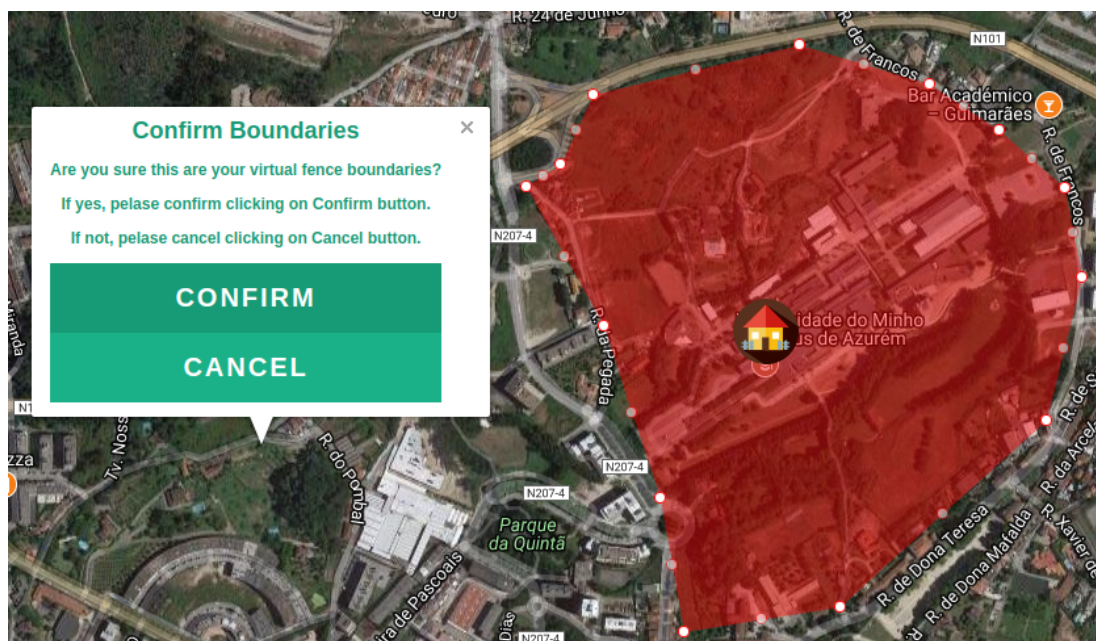


FIGURE 30. Confirm Boundaries

By confirming it is considered that the fence definition is complete. The data acquired is a series of latitudes and longitudes useful for implementation of control on the CP.

After these steps, the user had the option to define new limits, register the animals or monitor the cattle. If the user opts to define new limits, the process of confirming limits, send the SMS with the fence data is realised again. So, the old boundaries are excluded from the computing process of the collar checking and the new ones begin to operate. This means that there was no need for the collar regularly request for updates in regard to the fence boundaries.

### 5.1.3 Register Animal

Whenever the user selected the *Animal Register* option, the user was able to register the animals for later monitoring them. The registration is done by introducing the animal name, the collar number and the field number (Figure 31).

The image shows a mobile application interface for registering an animal. At the top, there is a green navigation bar with three white text options: 'DEFINE LIMITS', 'ANIMAL REGIST', and 'MONITOR'. Below this bar, a white registration form is centered. The form contains three input fields: 'Name\*', 'Collar Number\*', and 'Field Number\*'. Below these fields is a prominent green button with the word 'REGISTER' in white capital letters. Underneath the form, a text instruction reads 'Please, check field number in the map bellow.' followed by a satellite map of a rural landscape with various fields and roads.

FIGURE 31. Register Animal

The collar number is included in each collar. This step was taken to ensure the animal with the accordingly collar number gets the correct boundaries. The field number is visible on the map found after the data fill box (Figure 32). This step was taken to ensure the animal goes to the correct field.

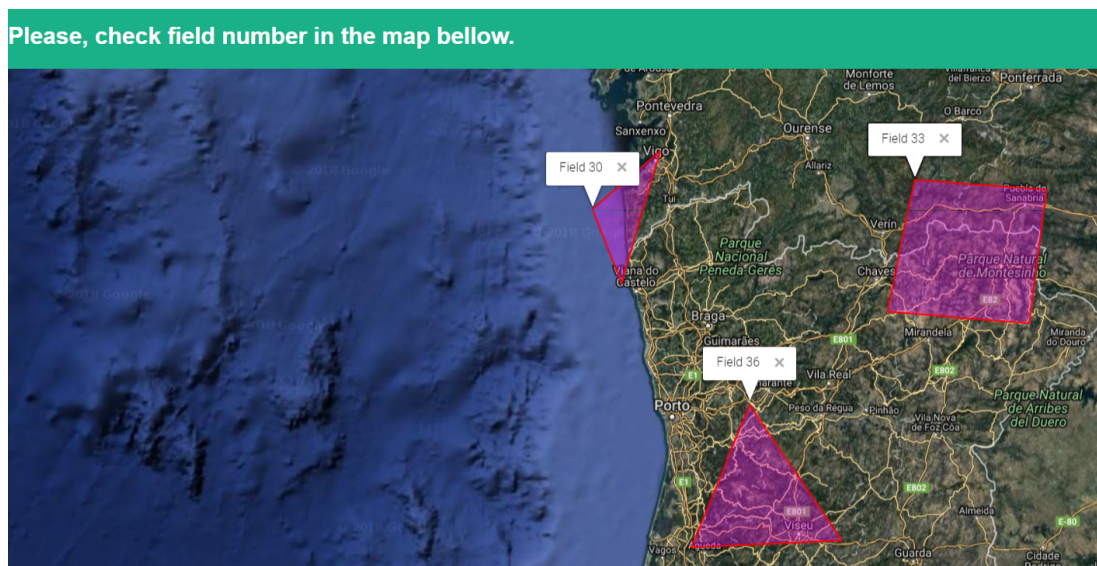


FIGURE 32. Register Animal - Field(s)

### 5.1.4 Monitor

Whenever the user selected the *Monitor* option, the user was able to select the field to monitor and then monitor the herd. The user was allowed to visually check if an animal stayed inside the fence, tried to escape or escaped (or had been stolen). Since the collar frequently sends information to the application, monitoring is an available option. To access to the Monitor option, the user had to first select the designated field to monitor, by clicking on it and confirming if that was the desired field (Figure 33).

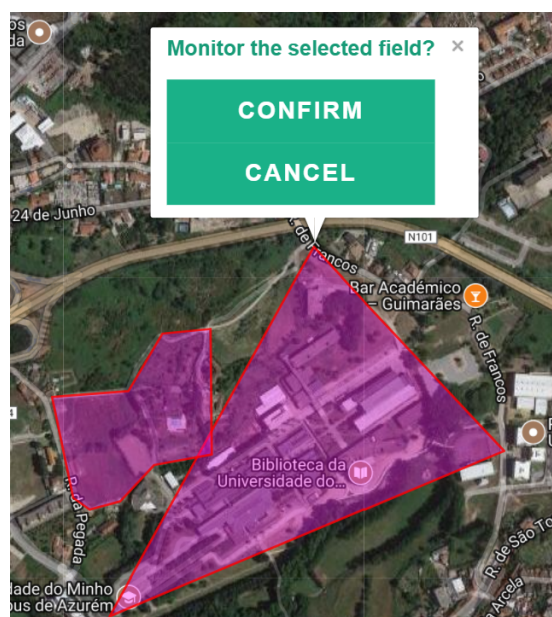


FIGURE 33. Monitor



The CP sent the animal location to the website database through the GSM module. The SMS arrived to the server and was decoded into latitudes, longitudes and the fence type (Green, Yellow or Red). This data was then inserted into the database through MYSQL queries. The file `sms.php` decoded the SMS and inserted the values into the `collar_coord` table.

The *Monitor* option requested the GPS values from the database and placed a marker on the map, which represented the animal. The icon colour varied according to the zone the animal was, green, yellow or red (Figure 34).

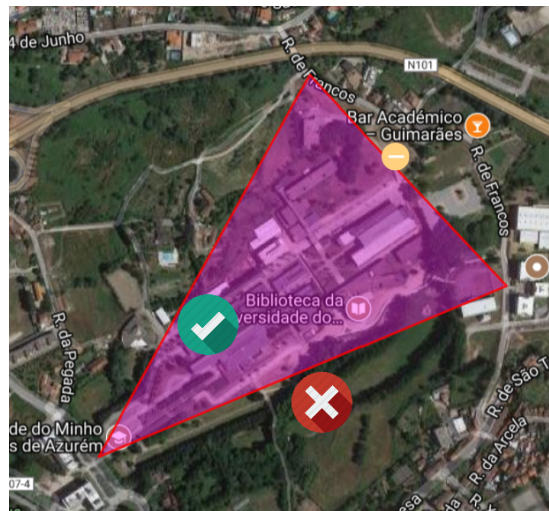


FIGURE 34. Monitor Animal

The first request was executed whenever the user clicked the *Monitor* option and from then, automatically every five minutes. The information is updated in the database every time a message (SMS) is received by the SA, and every five minutes so is the monitor marker.

## 5.2 Google Maps API

Since it was necessary for the website to have an interactive map with multiple options to utilise it, the Google Maps API was the solution. Google APIs (Application Programming Interface) are a set of tools and methods usable to build software applications [41]. It was developed by Google and allows communications with Google Services and integration with other services including Google Maps. They provide functionality like analytic, machine learning as a service and embedded Google map on a website [42].

### 5.2.1 API key

To access the Google map and use it on the website is necessary to create an API key through Google Developers. On their web page, it was necessary to choose the option **Web**, select the type of key according to the functionalities that Google offers. For this project the **Google Maps Geolocation API** was chosen because it was possible to "find a location and accuracy radius-based on information from things like cell towers and WiFi access points that a mobile client can detect with it" [43]. Next it was necessary to get the API key at the **Console** by creating a project and adding credentials to the project.

The key is generated and the next step is to copy it to the *index.html* file adding it to the source after the line `"key=<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCdsnzNm2l3ALRVE42tJAoiNghb4ErZNMg" type="text/javascript"></script>`.

### 5.2.2 Map

As for creating the map, the method called was the `create` with `options` as input. The options are various and for this dissertation project, the map options were as follows so the map follow the Earth curvature, editable and opaque in 45%:

- `geodesic: true`
- `strokeWeight: 0`
- `fillOpacity: 0.45`
- `editable: true`

### 5.2.3 Functions and Methods

The Google Maps Geolocation API was useful not only to create and use the map but also to develop some of the website functionalities. The principal methods used were:

- **Event** - such as `addListener` and `addDomListener`.
- **Size** -to size an icon.



- **InfoWindow** - to show the information windows.
- **Geocoder** - to get the address that the user is searching.
- **Drawing** - Creates a drawing manager attached to the map that allows the user to draw markers, lines, and shapes. This is useful to the user draw the limits of the virtual fence on the website map. This method contains the `Overlay` method that will be used to obtain the latitude and longitude of the drawn fence whatever is shape - `OverlayType` (rectangle, circle, line and polyline).
- **LatLng** - to set a point in map with certain latitude and longitude.

### 5.3 Database

The database tool has a relevant importance in this dissertation project since it was necessary to accommodate and organise data collection. The database-management system (DBMS) used was the MySQL through SQL language, allowing definition, creation, querying, update and administration of databases [44].

Therefore, it was projected a diagram to reveal the entities and respective relationships. In order to reveal the system design, Unified Modeling Language (UML) is utilised. The database Entity–relationship (ER) model is represented below by Figure 35 where the Primary Keys are represented in **bold**.

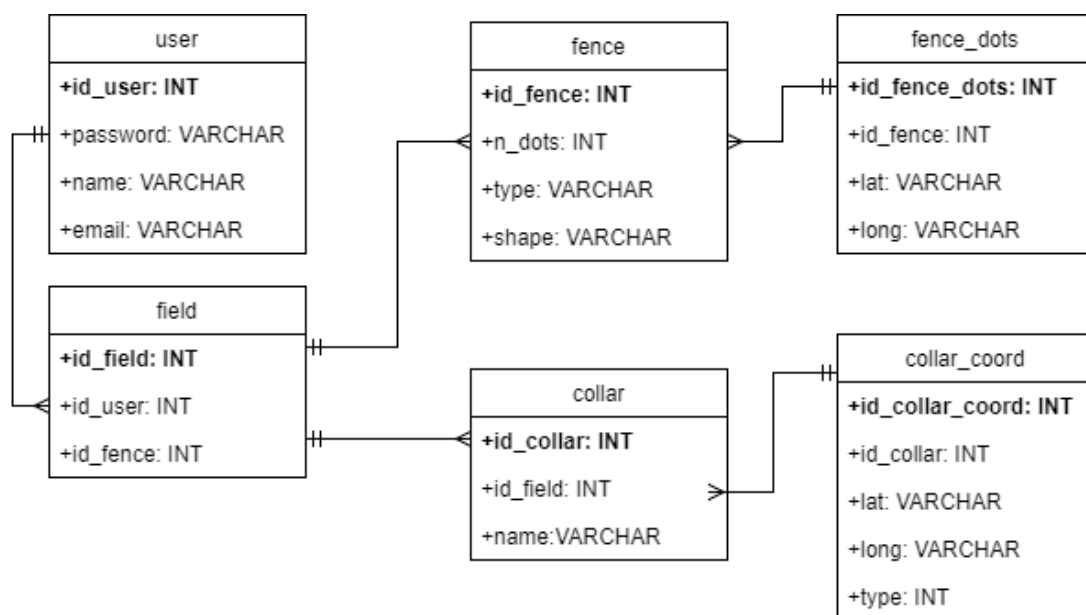


FIGURE 35. Database ER

The relationships (lines) between the tables (entities) are detailed in Table 8.

TABLE 8. Database Entities Relationships

Entity 1	Entity 2	Relation
User	Field	One user must have one or many fields
Field	Fence	One field must have one or many fences
Field	Collar	One field must have one or many collars
Fence	Fence_dots	Many fences must have one coordinate
Collar	Collar_coord	Many collars must have one coordinate

As visible this project database is composed of six tables (boxes/entities):

- Users: **user**

The **users** table was created due to the fact that there will be more than one user utilising the website, so a need urged to manage many users. The user information - Name, Username, Email and Password - is added to the table when registering a new user in the website. When the user tries to login on the website account he has to use the username and password. If the account exists the website redirects to the boundary definition or to monitoring page. If the account does not exist the website returns an error and is necessary to register the user.

- Field: **field**

The **field** table was created to save the various fences associated with an user. One field, that belongs to a certain user identifiable by the **id\_user**, has the original fence designed by the user **id\_fence** associated.

- Fence: **fence**

The **fence** table was created to save data such as the ID (**id\_fence**), the number of dots (**n\_dots**) of the fence design (circle is just 1 dot, rectangular is always 2 dots, and N for the polygon and polyline shapes), the **shape** (circular, rectangular, polygon, polyline) and the **type** (Green fence, Yellow fence or Red fence). The **n\_dots** is important for calculations posterior to the fence design. The **shape** will be important for the CP side when designing the fences. To further arrangements and efficient management, three fence tables will be created: the original fence and two more calculated. The original fence corresponds to the fence the

user designed and it will correspond to the red zone as in the last limit an animal go. If the animal crosses this red zone the electric shock cue is activated. One of the calculated fences will correspond to the yellow fence, meaning that if crossed the vibration cue is activated. The last fence will correspond to the green fence, meaning that if the animal is inside it nothing happens, if the zone is crossed the sound cue is activated. In short, the **type** is important to determine which fence is which, if it is the green, yellow or red fence. After the user design the fence on the map, by clicking on the field defined, an information window appears to confirm the boundaries definition. When confirming the limits, and depending on the design shape selected, coordinates are sent to database.

- Polygon Shape returns an array of coordinates (latitude and longitude), length  $N$  equal to the number of points/dots (**n\_dots**) designed on the map.
- Rectangular returns an array with two coordinates (latitude and longitude) as the two points/dots (**n\_dots**) pressed when designing the rectangle in the map.
- Circular returns the circle center (latitude and longitude) and the circle radius.
- Line return an array of coordinates (latitude and longitude) with length  $N$  equal to the number of points/dots (**n\_dots**) designed on the map.

The main difference between the polygon shape and the line shape is that the polygon is a closed area which is very useful later on programming. The line shape will always be an open area which can lead to inefficient management. Each fence will then have coordinates data visible through the table **fence\_dots**.

- Fence coordinates: **fence\_dots**

This table contains the geographic data, latitude **lat** and longitude **lng**, about the fence as well as the corresponding ID of the fence **id\_fence**.

- Collar: **collar**

The **collar** table was created to save information about the animal that is using it. The collar has an ID, **id\_collar**, which is important to manage the herd, and a name, **name**, to better associate which is the animal with the correspondent collar. Each collar then had coordinates data visible

through the table **collar\_coord**, and was associated with a field through the **id\_field**.

- Collar Coordinates: **collar\_coord**

This table contains the GPS data, latitude **lat** and longitude **lng**, about the animal location, the corresponding ID of the animal collar **id\_collar**, as well as the area **type** which the animal is at with the current **lat** and **lng**.

## 5.4 Communication

Besides the Graphical Environment main functions, there was some important implementation that deserves mention. The code that allowed communication between the website and the database is very important. The communication system is detailed in Chapter 7.

Communication between the SA and the Collar Prototype was crucial, so was necessary to implement it. The GSM server module received information from the CP. In order to obtain the SMS data, was necessary to implement a gateway with the website database. The Diafaan software program served the mentioned purpose. This SMS Server consists on an SMS gateway that eases the send and receive SMS process. With the Diafaan was possible to send and receive SMS from and to a web browser, web server, email, SQL database or C#Visual Basic [45].

The SMS with the animal data localisation followed the frame type detailed in Table 9.

TABLE 9. Website receive SMS frame

ID	Date	Time	Zone	Latitude	Longitude
1byte	10bytes	8bytes	1byte	14bytes	14bytes

*ID* stands for the collar identification number, *Date* and *Time* for, respectively, the date and time of GPS reading. The *Zone* number stands for the area the animal is at. If the animal is inside the field, the *Zone* number equals to zero (0), if crossed the first barrier equals to one (1), if crossed the second barrier equals to two (2), if crossed the third barrier equals to three (3). This information was later utilised to correspond to the accordingly icon in order to reveal the correct icon on the monitor map. *Latitude* and *Longitude* stand for

the animal GPS localisation data. These were the coordinates planted on the monitor map.

Since it was possible, in Diafaan gateway setting, to introduce the website address, the URL contained the path to a PHP file called **Db\_Gsm.php**<sup>1</sup>.

This information is inserted in the database through a MYSQL query that update the latitude, longitude and the zone type (0,1,2 or 3 for, respectively, inside the field, cross first fence, cross second fence, cross third fence) of the corresponding animal collar.

The SA had to send an SMS with the fence information, becoming necessary to the GSM website module acquire that data. The fence information was obtained through a MYSQL query on the website side. It acquired the latitude, longitude and number of the designed fence vertices <sup>2</sup>.

---

<sup>1</sup>Verify Appendix A

<sup>2</sup>Verify Appendix A

## Chapter 6

# Collar Prototype

The development of a collar prototype CP was one of the main goals of this project. So a CP was developed with the final goal to be attached to the animal. Its main function is to track the animal's movements and actuate according to the animal is inside certain boundaries.

To build the collar it was necessary to provide it with hardware components and software algorithms. Next are the details of the CP development.

## 6.1 Hardware

### 6.1.1 Components

As the CP needs to track the animal, a GPS module was attached to the microprocessor selected so that the animal's location can be determined. To exchange information with the server (website and database) the GSM module was attached to the collar. About hardware, the CP is composed of the components listed in Table 10.

TABLE 10. Collar Prototype Hardware

Hardware	Component
Microprocessor	Raspberry Pi 3 Model B (Figure 11)
GPS	ublox Neo 6-M (Figure 12)
GSM	SIM800L (Figure 13)
Serial Adapter	FTDI USB-TTL (Figure 14)
Battery	Xiaomi Pocket Power Bank (Figure 15)
Stimuli	LED's (Figure 16)

### 6.1.2 Connections

The following figure (Figure 36) reveal the collar prototype hardware connections.

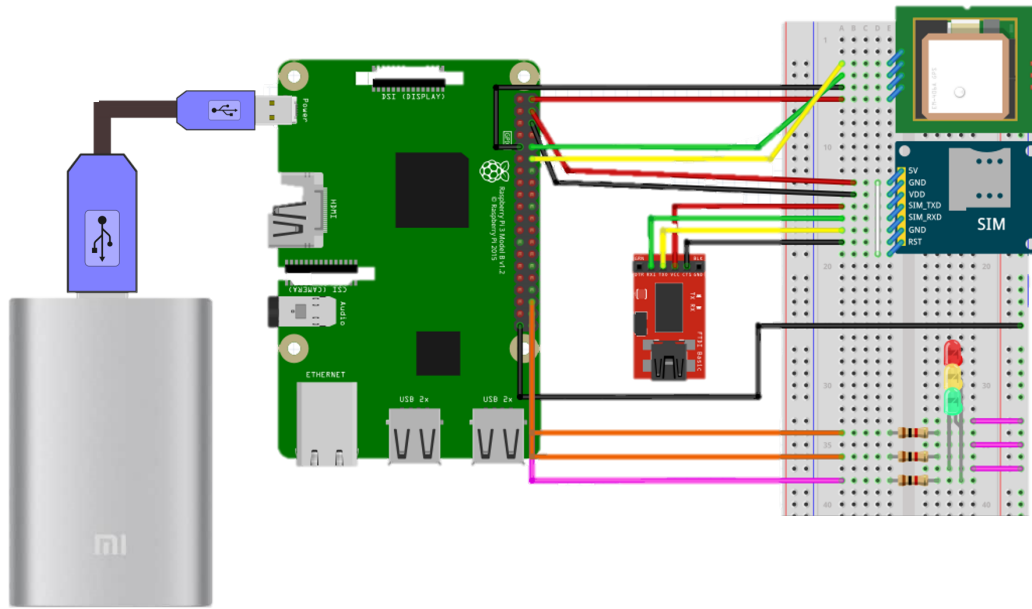


FIGURE 36. Collar Hardware connections

Figure 37 details the connection between the Raspberry Pi and the NEO-6M ublox GPS.

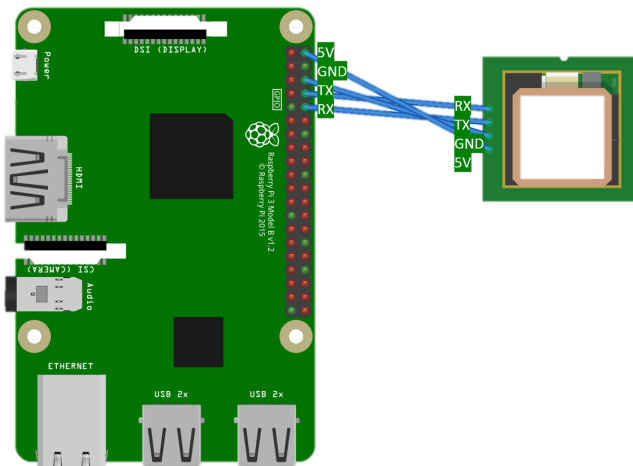


FIGURE 37. Raspberry - GPS

Figure 38 details the connection between the Raspberry Pi and the SIM800L GSM.

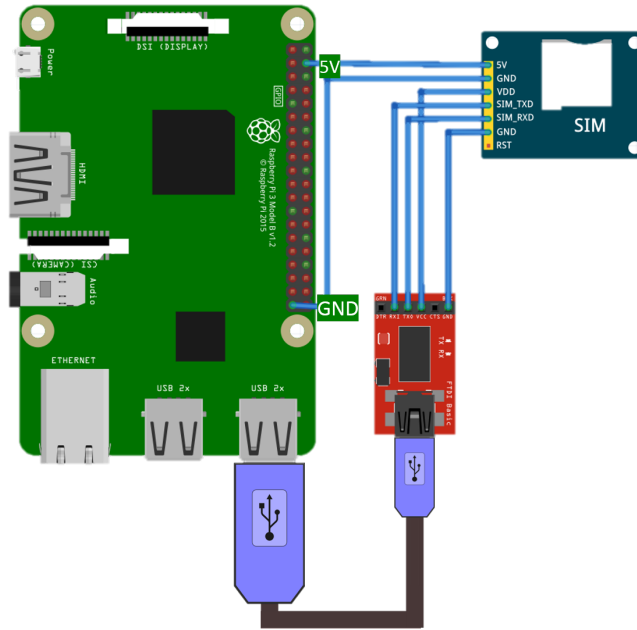


FIGURE 38. Raspberry - GSM

Figure 39 details the connection between the Raspberry Pi and the LEDs serving the purpose of stimuli.

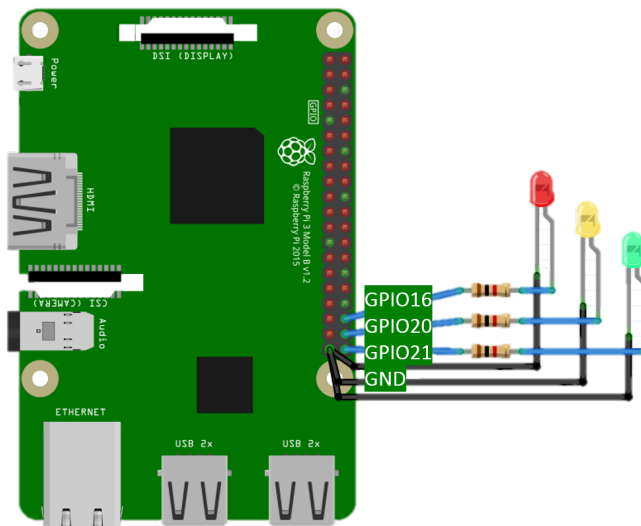


FIGURE 39. Raspberry - LED



## 6.2 Software

### 6.2.1 Management Algorithm

To manage the animal behaviour was necessary to compare the animal localisation with the fence limits. So the GPS module collected data about the animal localisation and the GSM module on the collar received the fence information. In this project development, five class were implemented: *Ccollar*, *Cboundaries*, *Cgps*, *Cgsm* and *Cstimuli* (Figure 40).

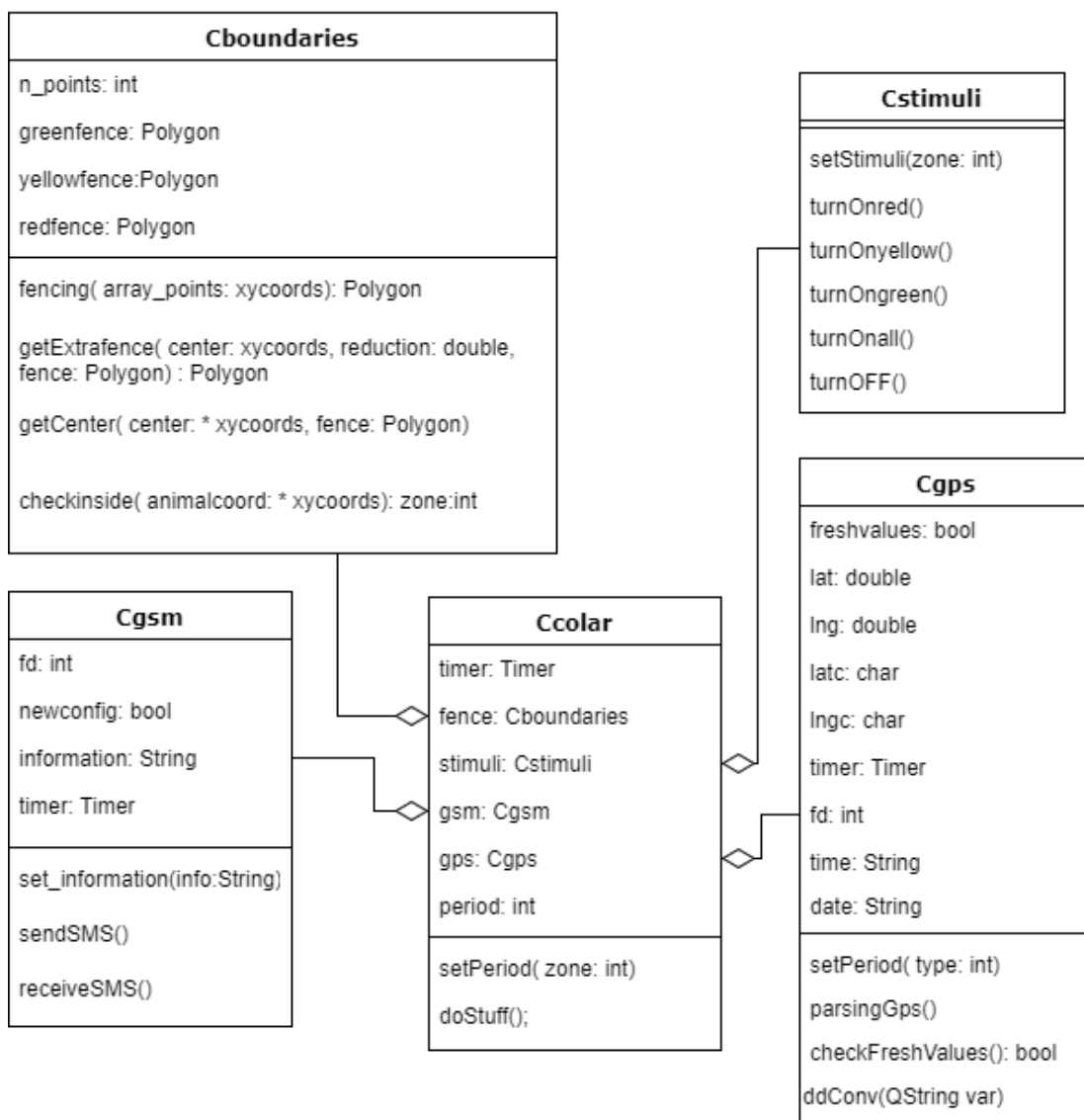


FIGURE 40. Collar Class Diagram

`Ccollar` class includes objects such as a `timer`, `Cboundaries fence`, `Cstimuli stimuli`, `Cgsm gsm`, `Cgps gps` and `int period`. This class also manages `setPeriod(int`

*zone*) and *doStuff()* functions.

- *setPeriod(int zone)* is responsible to attribute a time period, milliseconds, to each zone. Inside field period is equal to 60000ms. Crossing first barrier, period is equal to 20000ms. Crossing second barrier, period is equal to 10000ms. Lastly, crossing last barrier or being outside the field, period is equal to 5000ms.
- *doStuff()* is responsible to execute most part of the management algorithm by calling functions from other classes. It calls the `Cgps parsing_gps` to parse the GPS output data. Stops the current timer and then checks if there are new GPS values to verify the animal location area in the field, `Cboundaries checkinside` function. Then the stimuli are applied according to the zone where the animal is at, `Cstimuli setStimuli` function. Next, all data is appended to be sent by SMS through the GSM, `Cgsm set_information`. Finally a new period is set and the timer starts.

`Cboundaries` class includes objects such as the number of fence vertices *n\_points*, the three fences *greenfence*, *yellowfence* and *redfence*. This class also manages the *fencing*, *getCenter*, *getExtrafence* and *checkinside* functions.

- *fencing* function is responsible to design fences according to the vertices coordinates (x to longitudes and y to latitudes). It is called to transform the original fence coordinates received by SMS into a polygon fence.
- *getCenter* function is responsible to calculate the center of the original fence.
- *getExtrafence* function is responsible to scale a fence according to the reduction percentage (input value). This function is called two times to create the two boundaries fences making a total of three fences (original and two calculated fences).

`Cgsm` class includes objects such as the file descriptor *fd*, the new configuration, *newconfig*, the information to send, *information*, and a timer to send the SMS every few milliseconds. This class also manages the *set\_information*, *sendSMS* and *receiveSMS* functions.

- *set\_information* function is responsible to build the SMS frame to be sent containing the animal latitude, animal longitude and number of fence vertices.

- *sendSMS* function is responsible to execute all processing involved in the action of send a SMS. It is necessary to setup the GSM by calling AT commands specific to the message transmission.
- *receiveSMS* function is responsible to call specific AT commands to read SMSs. This function also decodes the SMS information.

`Cgps` class includes objects such as the file descriptor *fd*, if there are new values, *freshvalues*, and latitude, longitude, time and date. This class also manages the *readGPS*, *parsingGPS*, *checkFreshValues* and *ddConv* functions.

- *readGPS* function is responsible to setup the GPS and read the output.
- *parsingGPS* function is responsible to parse the GPS output in order to read only the necessary sentences.
- *checkFreshValues* function is responsible verify if there are any new GPS values to read.
- *ddConv* function is responsible to convert the geographic coordinates (latitude and longitude) from Degrees, Minutes, Seconds (DMS) to Decimal Degrees (DD). This procedure was done since the website map work with DD coordinates and consequently the fence coordinates were in DD. To take no risks when checking the animal location in the fences, all coordinates are in DD.

When receiving a latitude in DMS coordinates, for example **413308.7**, the frame is as detailed in Table 11.

TABLE 11. DMS frame

(d)Degree °	(min) Minute '	(sec) Second ''
41	33	.95015

The conversion is obtained through the some represented in Equation 1.

$$DD = DMS_d + (DMS_{min}/60) + (DMS_{sec}/3600); \quad (1)$$

*DD* stands for the coordinates in decimal degrees, *DMS<sub>d</sub>* is the DMS degree value, *DMS<sub>min</sub>* stands for the DMS minute value and *DMS<sub>sec</sub>* stands for the DMS second value.

`Cstimuli` class includes functions such as *setStimuli*, *turnOnred*, *turnOnyellow*, *turnOngreen*, *turnOnall* and *turnOFF*.

- *setStimuli* function is responsible to setup the stimuli according to the zone the animal is at.
- *turnOnred* function is responsible to activate the stimulus associated to the crossing of the last fence.
- *turnOnyellow* function is responsible to activate the stimulus associated to the crossing of the second fence.
- *turnOngreen* function is responsible to activate the stimulus associated to the animal being inside the fence.
- *turnOnall* function is responsible to activate the stimulus associated to the crossing of the last fence.
- *turnOFF* function is responsible to deactivate all the stimulus.

The GPS acquired data in a static frequency (5 Hz), but the collar only read the data on a dynamic period according to the animal localisation. By default the period data parsing is minimum, one minute (60000 ms). This procedure was implemented to reduce power consumption, which was a crucial aspect of this project. The following flow chart in Figure 41 reveals how the information processing was handled.

It will wait until the GPS is ready, parsing then the GPS data according to the defined period (default is one minute). Next checks if the animal is inside or outside by analysing if the GPS data value is lower than the minimum and higher than the maximum red fence values.

If it is, it means the animal is in the red zone or outside and the period data parsing became five seconds (5000 ms). The processing trigger is actuated and then returning to process information again.

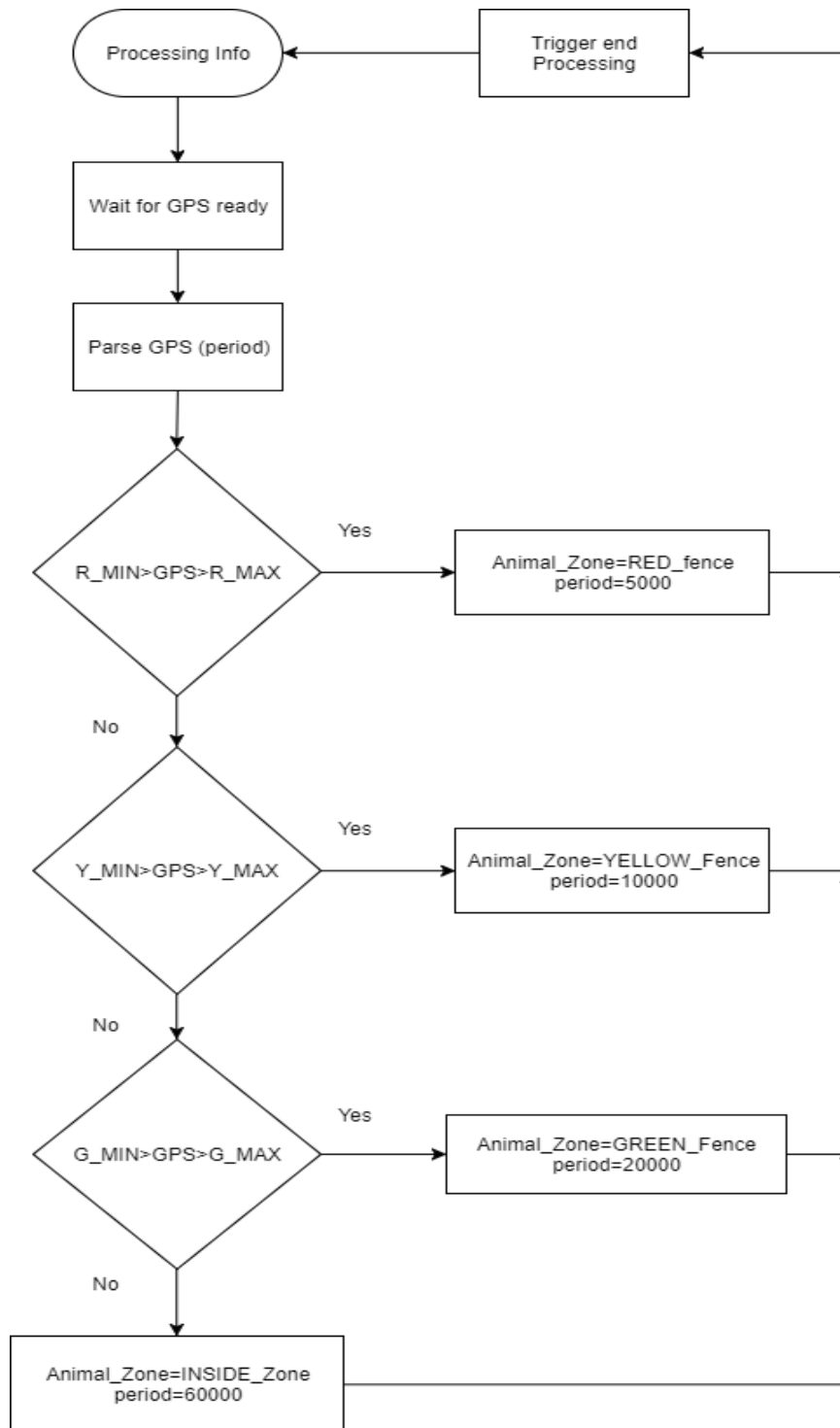


FIGURE 41. Processing Information Flow Chart

However, if the values are within the red fence value's range, a new test is performed. It inquires whether the GPS data value is lower than the minimum and higher than the maximum yellow fence values.

If the GPS values are, the animal is in the yellow zone and the period data parsing became ten seconds (10000 ms). The processing trigger is actuated and

then returning to process information again.

If the GPS values are not, a new check is done. It checks if the GPS data value is lower than the minimum and higher than the maximum green fence values.

If the GPS values are, means that the animal is in the green zone and the period data parsing became twenty seconds (20000 ms). The processing trigger is triggered and then it returns to process information.

If the GPS values are not, the animal is considered inside the field and far from the boundaries, where no stimulus is applied. Here the period data parsing became the minimum established, one minute(60000 ms). The processing trigger is actuated and then returning to process information again.

The animal's position inside the field and consequent stimulus applied shall work according to the following flow chart in Figure 42.

It will wait until the end of the GPS data processing, verifying next if the animal is inside or outside the red fence.

If it the animal is inside the red zone, the electric shock stimulus is applied. The processing then returns to stimuli handler again.

If not, it checks if the animal is inside the yellow zone.

If the animal is inside the yellow zone, the vibration stimulus is applied. Then the processing returns to stimuli handler again.

If the animal is not inside the yellow zone, it checks if the animal is inside the green zone.

If the animal is inside the green zone, the sound cue is applied. Then the processing returns to stimuli handler again.

If the animal is not inside the green zone, it is considered that the animal is inside the field and far from the boundaries, where no stimulus is applied.

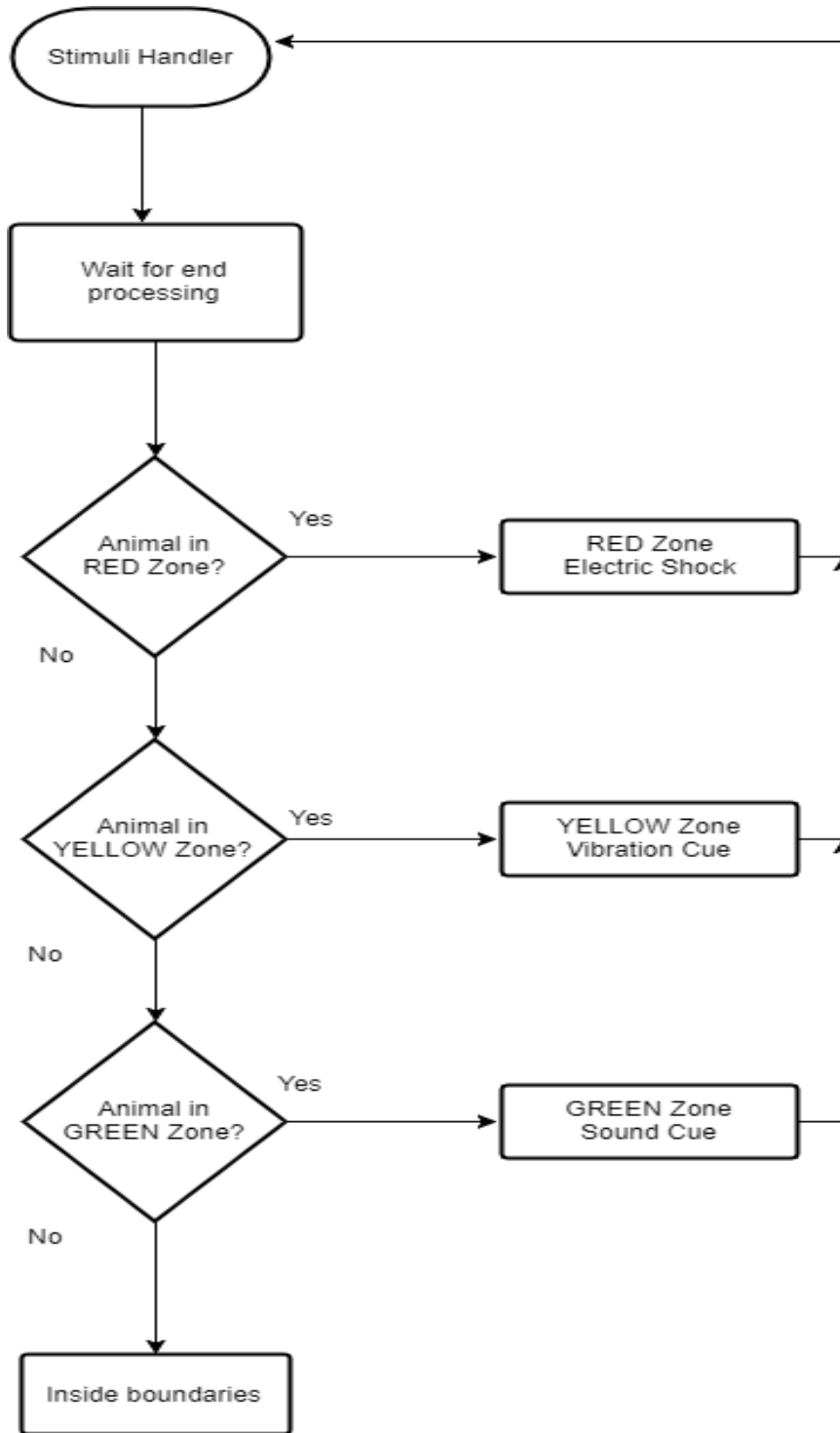


FIGURE 42. Stimulus Flow Chart

## 6.2.2 GPS

The NEO-6M GPS module receives NMEA (National Marine Electronics Association) data from the satellites. It became necessary to parse the received

information in order to obtain the desired information, such as latitude, longitude, date and time. NMEA sentences contain a data type, first word, which defines the interpretation of the rest of the sentence. NMEA standard define the unique interpretation of each data type. Some sentences may repeat the same information but also supply new data. It is possible to ignore sentences that are not an interest to the program. Sentences with GPS receivers applicability begin with "GP", Global Position.

Instead the receiving unit just checks the checksum and ignores the data if the checksum is bad figuring the data will be sent again sometime later.

There are many sentences in the NMEA standard. The ones available in the NEO-6M output are listed below:

- GSV - GPS Satellites in view
- RMC - Recommended minimum specific GPS/Transit data
- GSA - GPS DOP and active satellites
- GGA - Global Positioning System Fix Data
- GLL - Geographic position, latitude / longitude
- VTG - Track made good and ground speed
- TXT - Information on the receiver, such as power-up screen, software version.

This specific project selected three of the above listed NMEA sentences to obtain the intended data. They are RMC, GGA and GLL since all three contain latitude, longitude, date and time among other information. The RMC sentence frame is visible in the table below (Table 12).

TABLE 12. RMC sentence frame

T	S	Lat	N/S	Long	E/W	Sp	A	MV	*
---	---	-----	-----	------	-----	----	---	----	---

T stands for the time in hours, minutes and seconds. S means the status, A for valid and V for invalid. Lat stands for latitude, and the respective direction N/S north or south. Long stands for longitude, and the respective direction E/W east or west. Sp means the speed over ground, in knots. A means the



track angle in degrees. MV stands for the magnetic variation. \* corresponds to the sentence checksum.

The GGA sentence frame is visible in the table below, Table 13.

TABLE 13. GGA sentence frame

T	Lat	N/S	Long	E/W	FQ	NS	HD	A	M	H	M	*
---	-----	-----	------	-----	----	----	----	---	---	---	---	---

T stands for the time in hours, minutes and seconds. Lat stands for latitude, and the respective direction N/S north or south. Long stands for longitude, and the respective direction E/W east or west. FQ means the fix quality, may be [0-8]. NS means the number of satellites being tracked. HD stands for the horizontal dilution of position. A stands for altitude M meters above mean sea. H stands for height M meters above mean sea \* corresponds to the sentence checksum.

The GLL sentence frame is visible in the table below, Table 14.

TABLE 14. GLL sentence frame

Lat	N/S	Long	E/W	T	DS	*
-----	-----	------	-----	---	----	---

Lat stands for latitude, and the respective direction N/S north or south. Long stands for longitude, and the respective direction E/W east or west. T stands for the time in hours, minutes and seconds. DS means data status, A for active and V for void. \* corresponds to the sentence checksum.

### 6.2.3 GSM

To establish communication between the collar prototype and the software application website, the GSM communication needed to be set to send and receive SMSs. The collar GSM module had to be setup in order to send SMS to the website GSM. This was accomplished through out AT commands <sup>1</sup> following the below steps:

- AT + CFUN = 1 : Set to full functionality;
- AT + CREG = 1 : Configure to return unsolicited result code with when network registration is enabled.

---

<sup>1</sup>Verify Annex 10.2

- AT + CMGF = 1 : Set to SMS text mode;
- AT + CMGS = "telephone\_number" : Set phone number where to send the SMS;
- data\_to\_send : Set information/data to send.

The collar's GSM sent an SMS, Table 15, after decoding the GPS data.

TABLE 15. Collar sent SMS frame

ID	Date	Time	Zone	Latitude	Longitude
----	------	------	------	----------	-----------

*ID* stands for the collar identification number, *Date* and *Time* for, respectively, the date and time of GPS reading. The *Zone* number stands for the area the animal is at. If the animal is inside the field, the *Zone* number equals to zero (0), if crossed the first barrier equals to one (1), if crossed the second barrier equals to two (2), if crossed the third barrier equals to three (3). This information was later utilised to correspond to the accordingly icon in order to reveal the correct icon on the monitor map. *Latitude* and *Longitude* stand for the animal GPS localisation data. This were the coordinates planted on the monitor map.

These SMS were sent every five minutes. The five minute period was chosen due to the purpo energy consumption from the collar since the CP needed to use energy every time it sent a message. Regardless of the time chosen for monitoring, the animal was controlled according to the time corresponding to the GPS data reading frequency.

The collar received the fence information through the GSM module. The collar's GSM received a SMS later deconstructed into latitudes, longitudes and number of polygon points. Next, the fence was designed in the corresponding shape, rectangular, circular or polygon, as it was designed originally in the website. This step was taken so that later when examining if the animal was inside or outside the fence the computation became easier and faster.

In order to the CP receive an SMS with the fence information, was necessary to the GSM website module send the boundaries information. The SMS with the fence data consisted on the following frame, Table 16.

TABLE 16. Website send frame

ID	Latitude	Longitude	Number vertices
1byte	14bytes	14bytes	1 byte

*ID* stands for the field identification number, *Number vertices* mean the number of points the original fence is composed by. *Latitude* and *Longitude* stand for the animal GPS localisation data. This were the coordinates planted on the monitor map.

Reading the SMS sent from the website GSM module was accomplished through out AT commands <sup>2</sup> following the below steps:

- AT + CFUN = 1 : Set to full functionality;
- AT + CREG = 1 : Configure to return unsolicited result code with when network registration is enabled.
- AT + CMGR = smsID : Read the SMS correspondent from the smsID;

#### 6.2.4 Fence Calculus

The fence calculation was necessary because three stimuli were utilised, meaning one stimulus per fence. The reason behind the three stimuli choice is detailed in section 3.4. To explain the three fence utility, an example is presented below. If the animal crossed the first fence (GREEN), the sound cue was activated, if the animal crossed the second fence (YELLOW), the vibration cue was activated. Finally, if the animal crossed the drawn fence (third fence - RED), the electric shock stimulus was activated.

The method used to obtain the first and second fences was to calculate each coordinate points  $x$  meters away from the originally drawn fence coordinate. The algorithm base is the same for the three available shapes, rectangle, circle and polygon, with slight differences detailed below.

To implement the method in the polygon, first was necessary to calculate the fence area, corresponding to the coloured part of the map, Figure 43.

---

<sup>2</sup>Verify annex 10.2



Equation 3 and Equation 4 reveal the mathematical expressions to obtain the polygon centroid ( $C_x, C_y$ ). Also  $x_N$  is assumed to be  $x_0$ .

$$C_x = \frac{1}{6A} \sum_{i=1}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (3)$$

$$C_y = \frac{1}{6A} \sum_{i=1}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (4)$$

The following algorithm 1 reveals procedures to obtain the center of the fence polygon, with the mathematical expressions, mentioned above, rightfully implemented<sup>3</sup>.

---

**Algorithm 1** Obtain center

---

- 1: *Clat* ← Center latitude  $C_y$
  - 2: *Clng* ← Center longitude  $C_x$
  - 3: *lats* ← Polygon Latitudes list
  - 4: *lngs* ← Polygon Longitudes list
  - 5: **procedure** GET AREA
  - 6:     *area* ← Polygon area ← Equation 2
  - 7: **procedure** GET CENTER COORDINATE X
  - 8:     \**Clng* ← Equation 3
  - 9: **procedure** GET CENTER COORDINATE Y
  - 10:    \**Clat* ← Equation 4
- 

Later the calculation of the reduced fence is succeeded. The fence reduction is made through four steps execution:

- **Translate fence** - `translated QPolygon` method returns a copy of the polygon that is translated by ( $dx, dy$ ).
- **Transform from scale** - `fromScale QTransform` method creates a matrix which corresponds to a scaling of ( $sx$ ) horizontally and ( $sy$ ) vertically.
- **Map** - `map QTransform` method creates and returns a `QPolygonF` object that is a copy of the given polygon, mapped into the coordinate system defined by this matrix.

---

<sup>3</sup>Verify Appendix A

- `textbfTranslate fence - translated QPolygon` method returns a copy of the polygon that is translated by  $(dx, dy)$ .

The following algorithm 2 reveals the steps to obtain the extra fence(s) from the original fence.

---

**Algorithm 2** Obtain extra fences

---

- 1:  $fence \leftarrow$  Original fence polygon
  - 2:  $Clat \leftarrow$  Center latitude  $C_y$
  - 3:  $Clng \leftarrow$  Center longitude  $C_x$
  - 4:  $reduc \leftarrow$  Fence reduction in percentage
  - 5:  $fence\_reduce \leftarrow$  Reduced fence  $\leftarrow fence.translated(-Clng,-Clat)$
  - 6:  $trans \leftarrow$  Translate from Scale
  - 7:  $fence\_reduce \leftarrow$  Translate Map  $\leftarrow fence.translated(-Clng,-Clat)$
- 

This process is repeated to scale the original fence in order to obtain the YELLOW and GREEN fence. The difference is in the value of the reduction, 90% for the yellow fence and 80% for the green fence. Finally, the two fences are calculated and are now ready for more processing.

The three fences, Original fence, Fence 1 and Fence 2, were not visible to the user, but they were noticeable in the *Monitor* option on the website (Figure 45).



FIGURE 45. Red, Yellow and Green fences

- Original Fence - RED

This fence was the one that the user created at the *Define Limits* option. The original fence will appear in the website map defined by the latitudes and longitudes pointed by the user. This fence will later correspond to the RED fence composed in the CP and where outside this fence the electric shock stimulus is activated.

- Fence 1 - YELLOW

This fence is calculated by the software application through the method detailed. The Fence 1 design is created through the latitudes and longitudes obtained from the original fence. This fence will later correspond to the YELLOW fence composed in the CP and where outside this fence the vibration stimulus is activated.

- Fence 2 - GREEN

This fence is also calculated by the software application through the method. The Fence 2 design is created by the latitudes and longitudes ob from the original fence. This fence will later correspond to the GREEN fence composed in the CP and where outside this fence the sound cue is activated.

## Chapter 7

# Communication

The animal module (CP) and the user module (SA) had to communicate in order to exchange information about the animal GPS location coordinates and the fence boundaries, so the animal stayed contained. Since there were various options available, in terms of communication technologies, it became necessary to deepen the knowledge about some of them. The selected communication technology was the GSM, nevertheless different approaches were studied:

- ZigBee
- Radiofrequency (RF)
- Bluetooth
- GSM

Table 17 shows details from the communications technologies for a better understand of the chosen communication with characteristics from each of the communications mentioned above.

This table allows to conclude that the GSM communication technology is the most adequate for this project due to the extended range, good quality and reliability, low cost, included security protocols. An extensive range was one of the key characteristics of this project's execution. Since the field area could be wide it was crucial not to be restricted by a few meters or hundreds of meters. The quality and reliability were also important aspects taken into account. Mobile communications are immensely used which requires countries to have cell towers. Cell towers provide a good communication signal for the GSM mobile communication. In terms of costs, a mobile communication can be quite inexpensive, especially if the user already owns a monthly fee communication services. As for security, it is a good addition to the GSM characteristics since users can have their data secure.



TABLE 17. Communication Characteristics

Characteristics	Technology			
	Zigbee	Bluetooth	RF	GSM
Range	10m	+10m	10m-100m	35km
Base	Geographic location	Frequency hopping	Frequency	TDMA
Focus	Control and Automation	PC/Mobile phones	Laptop connectivity	Cellular network
Data rate	Low	Medium	Low	Low (2G) Medium (3G)
Power Consumption	Low	Medium	Ultra Low	Medium
Communication Protocols	UART SPI	UART SPI I2C	Unique Proprietary Protocol	UART SPI I2C
Network	Mesh	P2P	P2P	P2P
Advantages	Battery Life Cost Reliability	Cost Convenience	Unlimited number of devices/ transmitter	Range Quality Cost Reliability Security

## 7.1 SIM800L GSM module

The SIM800L GSM module was the selected component to establish the communication between the CP and the SA. Details about this component were already detailed on Section 4.

This project communication adopted the following communication scheme, Figure 46.

As the figure reveals, the Collar with the GPS module, battery and three stimuli, establishes communication through GSM with the software application and vice versa. The collar and the software application communicate directly with each other transferring messages between each other.

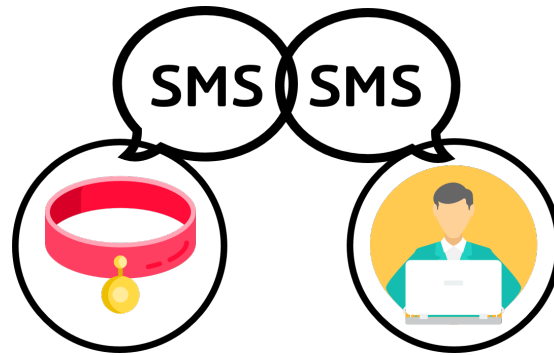


FIGURE 46. Communication Scheme

In order to establish communication between the collar and the software application was necessary to set up some settings.

- Baud Rate Information transference rate in a communication channel, this means, the number of times per second that a signal transitions between states [47]. In Serial Port context, a baud rate of 9600 baud means the serial port can transfer information up to 9600bits per second. Baud rate setting is important since the higher the rate, more sensitive the cable and installation quality becomes.

For this project, the GSM SIM80L module baud rate was established at 9600 baud.

- Data

The actual data of every serial packet. Each packet data amount can be set from 5 to 9 bits.

For this project, the GSM SIM80L module data size is 8 bits.

- Synchronisation bits

Synchronisation bits are two or three special bits transferred with each piece of data. The start bit marks the beginning of a packet. The number of bits of a start bit is always one. The stop bit marks the end of a packet. The number of end bit(s) is configurable to one or two.

For this project, the GSM SIM80L module stop bit was established as 1.

- Parity bits

The parity is a method to check errors and it may be odd or even. It is an optional method not very utilised since it slows down the data transfer and requires error-handling implementation from sender and receiver. If

the parity mode is even the parity bit is set to 1. If the parity mode is odd then the parity bit is set to 0.

For this project, there is no parity bit.

The SIM800L module utilises AT commands<sup>1</sup>. AT stands for ATTENTION and is used as a prefix to other parameters in a string. Combined with other parameters, the AT commands can be set up in the communication package as well as manually typed in a command line instruction [48].

The CP GSM module sent and read the received SMS through AT commands exposed in Chapter 6. The software application GSM module sent and read the received SMS through AT commands detailed in Chapter 5.

## 7.2 Broadband USB stick

To serve the communication purpose, a broadband USB stick was plugged into the server USB slot. Also the software utilised to send SMS with database fence data and receive SMS with animal localisation data was the *Diafaan SMS*.

The *Diafaan SMS* software is an SMS gateway for Windows, making it easy to send and receive SMS from and to a web browser, web server, email, SQL database or C#/Visual Basic .NET script, [45].

Besides the AT commands, there was the need to create two PHP files. One file to read data from the database through MYSQL queries, to get data from the fence designed in the website. The other file was to update the animal localisation inside the field monitor option.

Figures below reveal the steps on how to use the Diafaan software in order to make the connection between the communication module and the SA website.

In Figure 47 is explicit the elected Gateway to use, GSM Modem Gateway. Figure 48 shows the http gateway phases to fill in, such as the web page address, the HTTP query, method (POST) and the maximum length, 160 characters for one SMS.

---

<sup>1</sup>Verify Annex 10.2

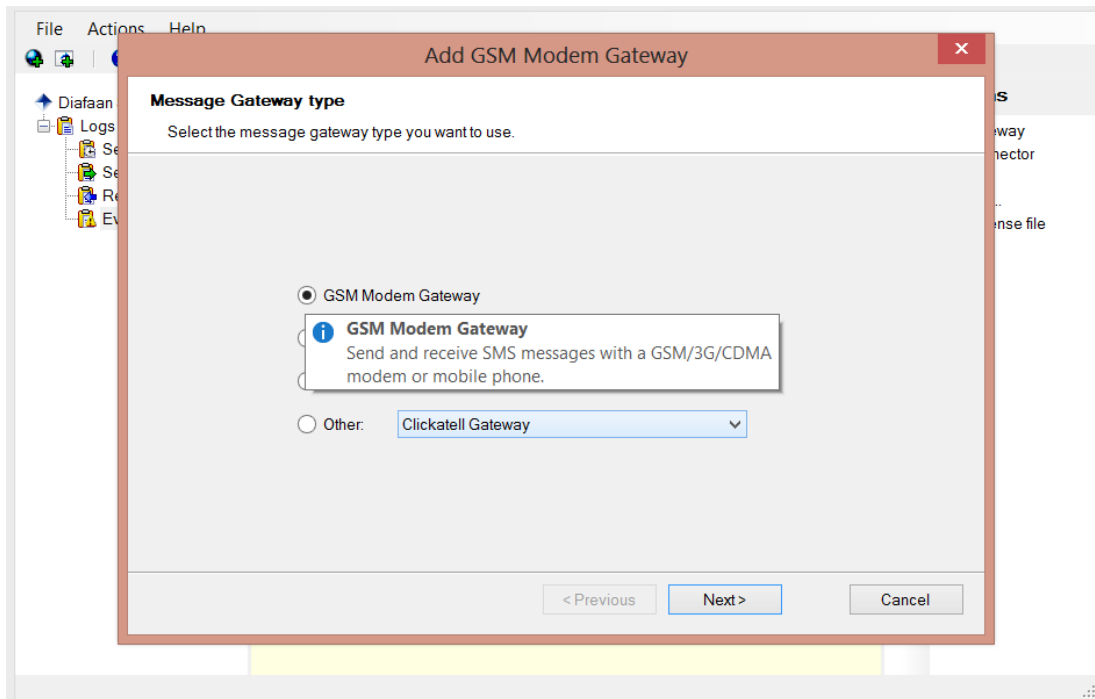


FIGURE 47. GSM Gateway

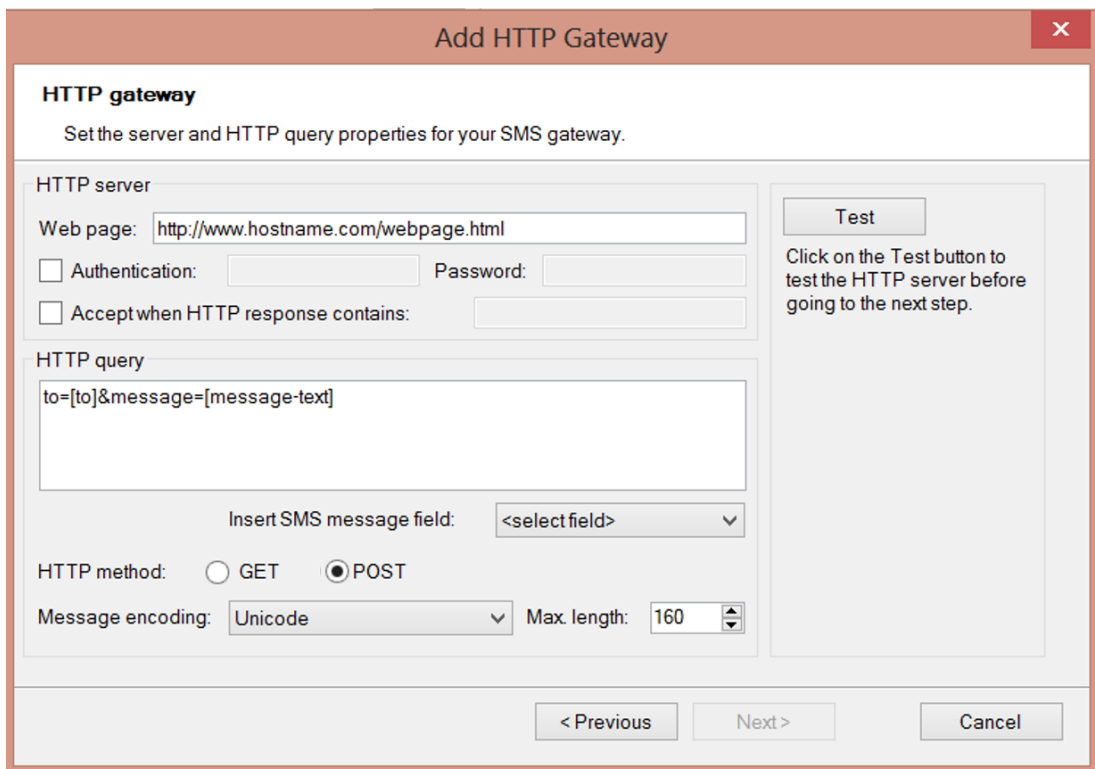


FIGURE 48. HTTP gateway

HTTP GET and POST request methods create an array that holds key/-value pairs, where keys are the names of the form controls and values are the

input data from the user. They are always accessible, regardless of scope, from any function, class or file [49].

POST is an array of variables passed to the current script via the HTTP POST method. With POST method, the information is not visible to others and there is no limit on the quantity of information [50].

On the other hand, GET is an array of variables passed to the current script via the URL parameters. With this method, the information is visible to everyone which means it should not be utilised for non-sensitive data. Also, GET method presents a 2000 characters limitation. However, variables are disposed in the URL which can be useful sometimes [50].

## Chapter 8

# Energy Study

To improve this project efficacy was necessary to conduct an energetic study where various possibilities to reduce energy consumption were scrutinised. The energetic study was not performed at an early stage, nor at a final stage. One of the first steps to execute was to carefully select components to the project execution. Thus, the energetic study was not just about the component choice. This energetic study was carried out through the entire project by making small changes and choices to improve the efficiency in all aspects and modules.

The energetic study was then parted into five main parts, the components choice, the processing algorithms, the GPS reading frequency, the stimuli application frequency and the data exchange frequency.

### 8.1 Components Choice

As mentioned in Section4 all the components were chosen due, mainly, to their low energy consumption. In order to achieve the low consumption hardware goal, was required to first design the project. When designing the project, the key components were determined. Only subsequently, the component specifications were considered in order to choose the component with the best characteristics for this project. It was critical to set the goal of a low consumption system since the collar attached to the animal would eventually require having the battery recharged. So, the fewer times needed to recharge the better, because the low maintenance time is the key.

### 8.2 Processing

When it comes to processing, different possibilities were taken into account.

Considering the software application SA side where the website and database were, some choices were considered. It was defined that designing the fence boundaries, register an animal in a certain field or monitor the cattle were options available on the SA side.

Thus, there was the possibility to execute most of the processing in the software application, for example, the two fences calculus. If the fences were calculated on the software application side, the energy to compute was not being consumed on the collar. This meant the collar prototype CP had to receive the three fences data through the GSM communication modules. This could become a problem since one SMS can only carry 160 characters.

A simpler (polygon) fence, with only three points (vertices) meant an SMS frame such as the described below, Table 18<sup>1</sup>. Where the *ID* is the field identification number, the *N Vertices* is the number of fence vertices (3 in this example). The "O" latitudes and longitudes refer to the original fence designed by the user. The "F1" latitudes and longitudes refer to the first fence calculated by the SA. The "F2" latitudes and longitudes refer to the second fence calculated by the software application. The numbers "1","2" and "3" refer to the correspondent vertex (1,2 or 3) latitude/longitude.

This totals 128 characters which can be inserted into one single SMS. However, if the fence number of vertices is higher than three, the number of SMS will increase. For example, four (4) vertices meant 170 characters, 2 SMS necessary. Whenever the collar read a received SMS, energy was consumed. The energy consumption would be higher to read all the SMS with the fence data, than to the collar calculate itself the first and second fences, through the original fence data. By only receiving the original fence data, user ID and number of vertices, is always ensured that only one SMS is sent for the collar. This is the reason why the fence calculations where defined to be realised in the collar prototype.

Considering the collar prototype CP, where three stimulus were available, was defined, as detailed above, that the fence calculations would be executed there. The three fences were considered necessary to relate one stimulus with only one fence. If only was considered the original fence, every time an animal crossed it, it was necessary to make calculations to apply the rightfully stimulus. So, the three fences definition allow the collar to save energy by not wasting it on calculus every time the animal cross the original fence barrier.

---

<sup>1</sup>For better perception purposes, the frame table disposition (vertical) is different from the previous displayed ones (horizontal).

TABLE 18. Three vertices fence data frame

ID	1 byte
N Vertices	1 byte
O Latitude 1	14 bytes
O Longitude 1	14 bytes
O Latitude 2	14 bytes
O Longitude 2	14 bytes
O Latitude 3	14 bytes
O Longitude 3	14 bytes
F1 Latitude 1	14 bytes
F1 Longitude 1	14 bytes
F1 Latitude 2	14 bytes
F1 Longitude 2	14 bytes
F1 Latitude 3	14 bytes
F1 Longitude 3	14 bytes
F2 Latitude 1	14 bytes
F2 Longitude 1	14 bytes
F2 Latitude 2	14 bytes
F2 Longitude 2	14 bytes
F2 Latitude 3	14 bytes
F2 Longitude 3	14 bytes

### 8.3 GPS Frequency

The ublox Neo-6M GPS has an update frequency interval from 1 Hz to 5 Hz. By default the update frequency is established at 1Hz which means the GPS received information every 1 second. If the collar updated every 1 second, the energy consumption would be high. Since one of the main goals was to have a low consumption system, it became necessary to adopt a strategy to decrease the GPS reading consumption.

Since it is not possible to decrease the update rate from the GPS, the solution became to create a function that reads the GPS data every minute. Independently from the GPS obtained data, the collar would only analyse it every other minute.

To accomplish this, the function that analyses the GPS data was regularly called within a time period of 60000 milliseconds (equivalent to one minute).



The GPS parsing default time period was defined as one minute. Thus, this parsing time period changed according to the area the animal was located. This is detailed in the section below, Section 8.4.

This period definition translated into an energy saving since the frequency reading from the collar is lower than the GPS update frequency.

## 8.4 Stimuli Frequency

This section is related to the above section 8.3 since the update frequency from the collar changes accordingly to the zone the animal is located.

- If the animal is inside the permitted zone, where no stimulus is applied, the update period was one minute (60 seconds).
- If the animal crosses to the GREEN zone, the collar update period decreased to 20 seconds.
- If the animal crosses to the YELLOW zone, the collar update period decreased to 10 seconds.
- If the animal crosses to the RED zone, the collar update period decreased to 5 seconds.
- Whenever the animal travels inwards the update period increases according to the area it is at.

This methodology intends to have a reinforced safety since the verification of the animal location is executed at different time rates, decreasing when the animal is near the authorised area. The GPS data analysis is realised at different period times without jeopardising the animal location control. The management was realised with the intention to give a better response according to the animal localisation on the field.

The approach made was a method to decrease energy consumption since the control had to be carefully realised. Therefore, if the stimuli and accordingly GPS reading were executed every five seconds (lowest defined time to manage the animal achieving good response from stimulus appliance) the energy consumption would be higher. Hence, since the update period varies, within a value range higher than five seconds, it was possible to save up energy.

## 8.5 Data Exchange Frequency

This virtual fence system proposed to send information, about the animal location, from the collar to the website so that the user may monitor the herd (animals with collar) at the **Monitor** option. Also, the software application SA needs to send information of the fences to the collar.

It was then expected a data exchange between the CP and the SA. To decrease the energy consumption, instead of the SA synchronously sending the fence data to the collar, this was only done when the user defined new boundaries to the field. Whenever the use designed new boundaries, a new SMS, with the new fence data, was sent from the website database to the collar prototype. If the system synchronously asked for fence data, either new data or equal data, it would mean a waste of energy consumption from the collar since it had to read the text messages.

Another way to save energy was to, instead of the CP sent the animal GPS location data at the same frequency as the collar update frequency data, the collar only sent data every 5 minutes. The 5 minutes mean 288 SMS/day sent from the collar to the database. One day has 24 hours, 1440 minutes, so 1440 minutes divided by the 5 minutes information exchange mean the 288 SMS per day. If the system sent an SMS every time the animal location was checked, it would mean that the SMS sending rate could be 60 seconds, 20 seconds, 10 seconds or 5 seconds accordingly to the zone the animal was at. If the SMS sending rate was equal to the GPS parsing rate it would translate in a higher energy consumption. This due to the fact that when sending an SMS occurs a peak of current consumed. So, in this scenario, the solution to save energy passed by having a static SMS sending period time of five minutes. Independently from this, the collar realised the necessary checks verifying whether the animal was inside or outside the fences.



## Chapter 9

# Tests and Results

This chapter includes results from the GPS, GSM communication, website database management and animal control. It was also necessary to analyse the energy consumption from the collar system. The efficacy and the efficiency of the project were evaluated since, in the field, the battery and all functionalities had to work.

### 9.1 GPS

To prove the accuracy and efficiency of the GPS NEO-6M tests were realised in different locations and under different ceilings. These tests outputs <sup>1</sup> reveal the raw information from the GPS NEO-6M with different NMEA sentences. Sentences that provided coordinates data (\$GPRMS,\$GPGLL,\$GPGGA) and signal strength data (\$GPGSV) were analysed. The rest NMEA sentences were ignored. Tests were realised, with the GPS held by the project developer, in open field, inside horse boxes with wood, plastic and aluminium ceiling, in a dense tree field, inside a stone house and in a house complex.

Figure 49 reveal the coordinates (latitudes and longitudes) flagged in red boxes and signal strength in yellow boxes.

```

$GPRMC,170844.00,A,4143.92149,N,00837.57774,W,0.316,,050917,,,A*6F
$GPVTG,,T,,M,0.316,N,0.586,K,A*2C
$GPGGA,170844.00,4143.92149,N,00837.57774,W,1,04,4.0837.5,M,50.5,M,,*76
$GPGSA,A,3,02,06,09,19,,,,,6.48,4.08,5.04*00
$GPGSV,2,1,06,02,30,311,36,06,72,287,33,09,83,059,35,7,,,28*43
$GPGSV,2,2,06,19,29,226,39,23,,,30*43
$GPGLL,4143.92149,N,00837.57774,W,170844.00,A,A*7A
$GPRMC,170845.00,A,4143.92097,N,00837.57741,W,0.121,,50917,,,A*6C
$GPVTG,,T,,M,0.121,N,0.223,K,A*22

```

FIGURE 49. GSM no ceiling Results

<sup>1</sup>Verify Appendix A

It was expected to obtain GPS signal since the test locations were in open fields and open areas in home complex. However, the GPS signal inside a house was weaker and sometimes was not possible to obtain localisation values. Also, depending on the hour of the day, inside a house, the signal was weaker or stronger. In dense forest, this GPS provided weak results, failing to localise sometimes. This proves that the GPS module signal strength depends on the location and in weaker conditions the time is also relevant.

TABLE 19. GPS Tests results

No Ceiling	Wood	Plastic	Aluminium	Dense forest	Stone
Great	Good	Good	Good	Marginal	Weak to no signal
36&39	25&21	23&25	25&21	13&16 ; 6&3	4&2

## 9.2 GSM

To prove the accuracy of the selected GSM, as GPS tests, were realised in different locations and below ceilings. These tests results belong to tests realised, with the GSM held by the project developer, in open field, inside horse boxes with wood, plastic or aluminium ceiling, inside a stone house and inside the stone house but near a window.

In order to verify the signal strength, was analysed the output from the GSM collar module when calling the AT command `AT + CSQ`. The signal strength values, in dBm, are mentioned in Table 20. To each response value from the AT command correspond the signal strength value in dBm which is related to the signal evaluation from marginal to excellent. The higher the value from the AT command response, higher the signal strength.

TABLE 20. GSM Tests results

No Ceiling	Wood	Plastic	Aluminium	Window	Stone
Excellent	Good	Good	Ok	Marginal	Weak to no signal
-69 dBm	-77 dBm	-75 dBm	-89 dBm	-99 dBm	-105 dBm

It was expected to obtain signal since the test locations were in cities and villages. However, the signal inside a house was weaker and sometimes was not possible to establish communication. This proves that although GSM communication was the best option, the signal strength depends on the location. In

this particular case if the location was under solid construction material such as stone.

### 9.3 Battery

Taking into account the energy consumption of the CP components visible in Table 21.

- Raspberry boot 250 mA
- NEO-6M 39 mA
- SIM800L in full functionality 18.7 mA + 2 A peaks when sending an SMS
- LED 10 mA

TABLE 21. System Event

Expected	Real
18 h 50	16 h 30

The energy consumption revealed to be higher than the expected probably because of the peak consumption from the SMS sending action, LED consumption and other that may not had been taken into account.

### 9.4 Virtual Fence System

After assembling all the modules that constitute the virtual fence system, Figure 50, tests were realised to evaluate the efficacy of the project.



FIGURE 50. System Overview Hardware

Table 22 shows the system software test cases with the expected and real output from the virtual fence system. Data output belong to the Collar Prototype and Software Application.

TABLE 22. System Test Case

Event	Expected Output	Real Output
Animal Outzone	All LED on	All LED on
Animal on Red fence	Red LED on	Red LED on
Animal on Yellow fence	Yellow LED	Yellow LED on
Animal on Green fence	Green LED	Green LED
Animal Inside fence	All LED off	All LED off
GPS connection error	Retry	Retried connection
GPS no fresh values	Wait	Waited until new values
New GPS data	Check location in field	Animal localisation checked
New GPS data	Send data to DB	Data sent to DB
GSM connection error	Retry	Retried connection
New fence design	Send data to DB	Data sent to database
New boundaries	Send configuration to CP	Configuration sent to collar
Monitor option on	Get animal data from DB	Animal data obtained from DB
Monitor option on	Update animal marker in map	Marker updated

Table 23 shows the system events with details about the source event and the event type. Events source belong to the Collar Prototype and Software Application.

TABLE 23. System Event

Event	System	Source	Type
Read GPS coordinates	Check if animal is inside perimeter	CP	Sync
Read GPS coordinates	Send coordinates to SA	CP	Sync
Outside Green perimeter	Turn green LED on	CP	Async
Outside Yellow perimeter	Turn yellow LED on	CP	Async
Outside Red perimeter	Turn red LED on	CP	Async
Receive data from SA	Design fences	CP	Async
User logs in website	Defines fence     Monitors cattle	SA	Async
User design fence in map	Get design data	SA	Async
Get design data	Send design data to database	SA	Async
Receives design data from website	Obtain latitudes and longitudes	SA	Async
Gets lat and lng	Calculate two larger fences	SA	Async
Gets lat and lng from three fences	Send data to database	SA	Async
Gets lat and lng from three fences	Send data to CP	SA	Async

The following virtual fence system test was realise at a house complex where GPS and GSM signals strength were good. Also, the collar was carried by the project developer through out the tests.

The limit definition was established in the website SA, Figure 51.





FIGURE 51. Define Limit Test

Then, the fence information was sent to the database, Figure 52.

id_fence_dot	id_fence	lat	lng
1	1	41.552911766604	-8.3885657793871
2	1	41.553626330708	-8.3866131312243
3	1	41.551627137113	-8.3853685862414
4	1	41.551145394473	-8.3873534209124

FIGURE 52. Fence Boundaries Database Data Test

The GSM server module obtained the fence data from the database and sent a SMS to the collar with the fence latitude, longitude and number of vertices. The collar received and read the SMS and proceeded to the new fence setting. The collar calculated the two fences and started to manage.

In this test it was visible that the collar was managed in the designated times. This is revealed by the fact that when the collar was inside the field, the control was made every one minute. When the collar crossed the green fence, the control was made every 20 seconds. When the collar crossed the yellow fence, the control was made every 10 seconds. When the collar crossed the red fence, the control was made every 5 seconds. This is proven by the time values found inside the blue box in the test figures below.

Figure 53 reveals the management algorithm output where the blue box contains the time (hours minutes and seconds), the black box contains the area where the collar is at, and the red box contains the collar latitude and longitude.

"0,180118	142631.00	0	41.5524,-8.38709"
parsing			
"\$GPRMC"			
"0,180118	142731.00	0	41.5524,-8.38701"
parsing			
"\$GPRMC"			
"0,180118	142831.00	0	41.5525,-8.38757"
parsing			
"\$GPRMC"			
"0,180118	142931.00	0	41.5524,-8.38778"
parsing			
"\$GPRMC"			
"0,180118	143031.00	1	41.5524,-8.388"
parsing			
"\$GPRMC"			
"0,180118	143051.00	2	41.5524,-8.38802"
parsing			
"\$GPRMC"			
"0,180118	143101.00	2	41.5524,-8.38804"
parsing			
"\$GPRMC"			
"0,180118	143111.00	2	41.5524,-8.38805"
parsing			
"\$GPRMC"			
"0,180118	143121.00	2	41.5524,-8.38806"
parsing			

FIGURE 53. System Results 1

It is possible to visualise that the collar was inside the field, area "0", and passed through the first and second fence getting into area "1" and area "2", respectively. In another test <sup>2</sup> was also visible that the collar went straight from the area "1" to outside the field. This could have happened due to a high-velocity pace what in this case meant that the 20 seconds range (from area "1") was enough time to walk through all the areas, getting outside the field.

As mentioned in partII, Methodology, the collar sent SMS every five minutes. However, for this test, the text messages were sent every two minutes. This is proved by the values inside the blue boxes in Figure 54. Again, red box reveals the collar geographic coordinates.

---

<sup>2</sup>Verify A







## 9.5 Data Analysis

Tests realised with horses revealed peculiar behaviour relative to speed. With the collar attached to the horse (Figure58) was possible to obtain information about the horse speed along time.



FIGURE 58. Horse with Collar

Figure 59 reveals the graphic analysis from the 1000 samples provided by the collar.

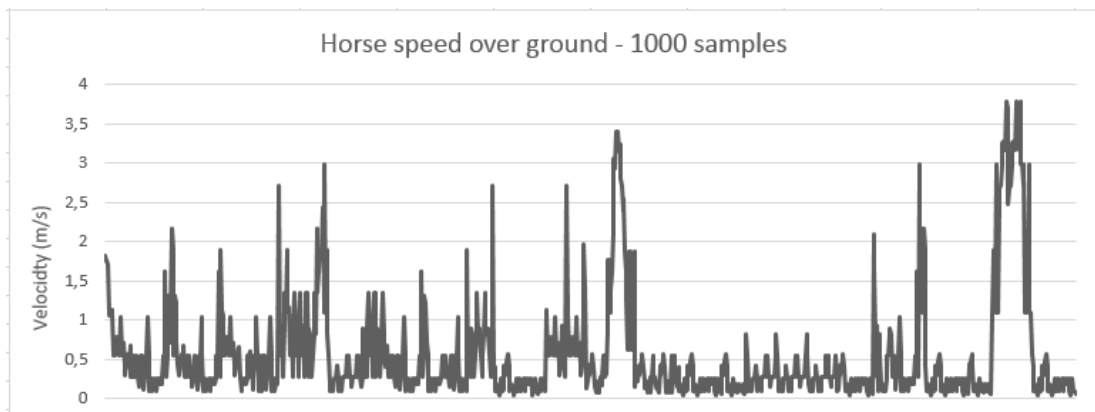


FIGURE 59. Horse with Collar Graphic Result

In the figure mentioned above is possible to see the horse velocity in meter per second (m/s) with values from 0 m/s up to 4 m/s. It is possible to evaluate that the horse spent most time foraging and therefore the average speed over ground was lower than 1.8 m/s.

Figure 60 reveals detail on the graphic analysis from 100 samples provided by the collar.

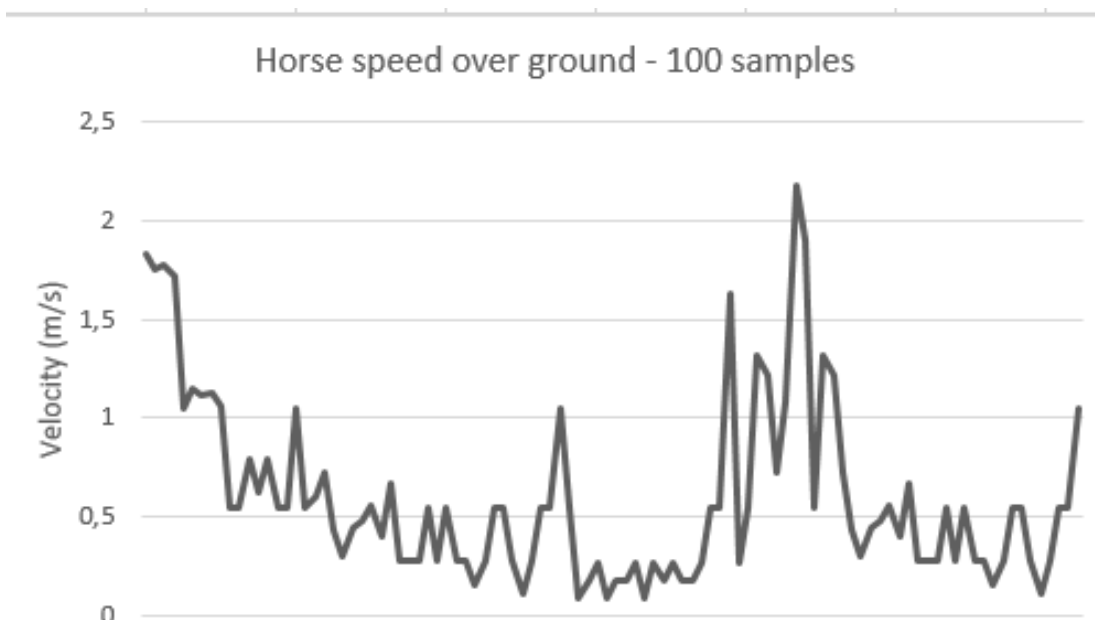


FIGURE 60. Walk gait Graphic Result

In the figure mentioned above it is possible to see the horse velocity varies from 0m/s up to 2.2 m/s. It is possible to evaluate that the horse speed under

0.5 m/s was due to sleep time and/or foraging time. Also, the horse walk gait average speed is under 1.78 m/s.

Figure 61 reveals detail on the graphic analysis from 190 samples provided by the collar.

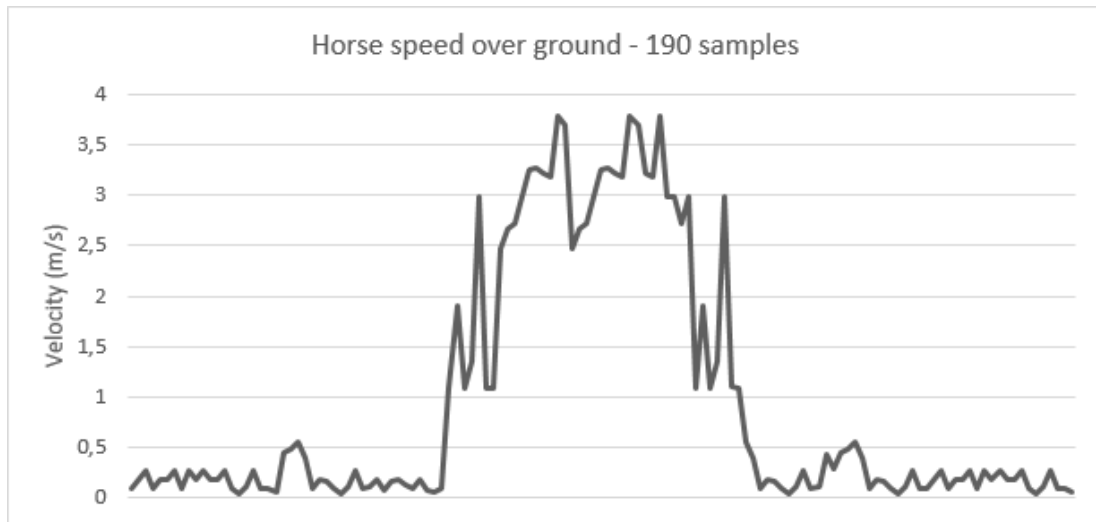


FIGURE 61. Trot and Gallop gaits Graphic Result

In the figure mentioned above it is possible to see the horse velocity varies from 0m/s up to 4 m/s. It is possible to evaluate that the horse speed above 2 m/s was due to trot and gallop. The horse trot gait average speed is under 3 m/s and the gallop gait average speed is above 3m/s and under 4 m/s.



## Chapter 10

# Conclusions

### 10.1 Conclusion

The thesis presented sought to deepen theoretically and empirically the problem of animal location management and respective distribution control. This deepening continued through four main routes: the development of a software application with the management of an efficient database; the development of a prototype collar in which the three stimuli studied were inserted; the choice and implementation of the communication between collar and software application; and finally the energy study related to the system energy consumption.

First, the components that later served to construct the prototype collar were chosen. Characteristics such as energy consumption, dimensions and weight of each component were taken into account in order to provide a more efficient prototype without compromising its robustness. This careful choice is considered as part of the energy study developed throughout the project. The results indicate that the energy consumption was higher than expected. This consumption may be associated with the choice of some specific or several components.

The software application was implemented to provide the user with a simpler fence definition. So, instead of having to go to the field and dial, for example with signposting or someone specialised in design fences, the user was able to draw the fence(s). This was due to the fact that the software application was thought to be user-friendly, so that any user, with or without advanced software knowledge, could use it. The results presented indicate that the whole process of communication between the website and the database was well established contributing to a simple and effective database. With this, the user can draw his fence, register the animal in the corresponding terrain and finally monitor it.



It became pertinent to develop a prototype collar with the ultimate goal of being attached to the animal so as to monitor and control it. In this context, the hardware was assembled with the necessary components, and later the implementation of the control algorithm. This algorithm integrates parameters such as values that come from GPS and GSM. These values were treated in such a way that the animal is constantly controlled, and the appropriate stimuli are applied accordingly the area in which it is found. In this context were chosen different time periods of evaluation and sending of text messages, for a lower consumption energy.

It is important to highlight the contribution that this project can make in solving problems with animal management. With the developed prototype, the solution is to migrate to a new animal management system where the maintenance of fences lowers dramatically, as well as the time spent with it. It is also highlighted the ease of always knowing where the animals are. This can be an important information in the case of an escape, or theft, in which the animal is outside the defined limits. Nevertheless, its location is known through the GPS data and it is possible to visualise on the website.

This virtual fence system with monitoring may still be relevant to behavioural studies of certain species of animals. Studies such as the current one to be developed with Portuguese horse breed, Garrano, in which their movements in certain mountain areas are analysed throughout the days. In this context, this prototype could contribute to the development of the project, providing monitoring data for future analysis and evaluation.

## 10.2 Future work

This thesis stands as an open project where work can be developed in the future. The future work mentioned above evidence the future goals to achieve relative to this virtual fence system. There are improvement goals towards the control algorithm, the communication and stimuli.

The tests results revealed that most of the time animals are foraging and therefore their speed over ground could be taken into consideration in order to develop an effective control algorithm. If speed is taken into consideration, the control can be managed by not only the animal location in the field but also by the animal speed. This could be possible with the implementation of a prediction filter that estimates the animal speed over time, taking into account

the previous speed.

When it comes to communication, sending 288 SMS per day can be expensive. To solve this problem Low-Power Wide Area Networking (LPWAN) technology offers long-range communication, promising communication in outdoor IoT applications [51]. EC-GSM-IoT is an evolution of EGPRS towards IoT, with data rate between 70 and 240 kbps [51] The 3rd Generation Partnership Project (3GPP) approaches intend to reduce the energy consumption and the cost of the devices. Coverage extension, UE complexity reduction, long battery lifetime, and backward compatibility are common objectives [52].

The key requirements for cellular IoT to enable these services and be competitive are [53]:

- Long battery life
- Low device cost
- Low deployment cost
- Extended coverage
- Support for a massive number of devices

There are available IoT SIM cards as Hologram offers [54]. There are various plans and for a free developer, there are options such as pay daily, monthly. The platform offers simple pricing plans to get started, build, and scale the device connectivity seamlessly. In this platform, Portugal is considered a Zone 1 where prices from \$0 for text messages sent to a device, up to \$0.19 for text messages sent from device.

The stimuli applied in this project were visible, LEDs on or off according to the test result. Chapter 3 contains a study about the physical stimuli possibly applicable to this project. In future work, there is the possibility to design the Electric Shock processing board in order to install it to the collar prototype. Also, a buzzer and a vibration motor could be attached to the collar prototype summing this way the three stimuli.



## Bibliography

- [1] D. M. Anderson, "Virtual fencing past, present and future," *Rangeland Journal*, vol. 29, no. 1, pp. 65–78, 2007. [Online]. Available: <http://www.publish.csiro.au/?paper=RJ06036>
- [2] L. B. Production and I. Development, "Lifting the barriers to virtual fencing," *Agersens News*, no. 292, pp. 51–52, 2016.
- [3] J. Wall, G. Wittemyer, B. Klinkenberg, and I. Douglas-Hamilton, "Novel opportunities for wildlife conservation and research with real-time monitoring," *Ecological Applications*, vol. 24, no. 4, pp. 593–601, 2014.
- [4] J. Eberspächer, H.-J. Vögel, and C. Bettstetter, *GSM : architecture, protocols and services*. Wiley, 2009. [Online]. Available: <https://books.google.pt/books?id=v6eN1tt9CEUC{&}dq=gsm+architecture{&}lr={&}source=gbs{&}navlinks{&}s>
- [5] C. Umstatter, "The evolution of virtual fences: A review," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 10–22, 2011.
- [6] V. Rana, A. Vermani, and S. Govil, "Virtual Fencing for Animals Managment Using RF Module," *of the Conference on Advances in ...*, vol. 2013, no. Cac2s, pp. 360–362, 2013. [Online]. Available: <http://www.atlantis-press.com/php/download{&}paper.php?id=6335>
- [7] Z. Butler, P. Corke, R. Peterson, and D. Rus, "From robots to animals: Virtual fences for controlling cattle," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 485–508, may 2006. [Online]. Available: <http://ijr.sagepub.com/content/25/5-6/485.abstract>
- [8] D. M. Anderson, R. E. Estell, J. L. Holechek, S. Ivey, and G. B. Smith, "Virtual herding for flexible livestock management - A review," *Rangeland Journal*, vol. 36, no. 3, pp. 205–221, 2014.
- [9] M. Jouven, H. Leroy, A. Ickowicz, and P. Lapeyronie, "Can virtual fences be used to control grazing sheep?" *Rangeland Journal*, vol. 34, no. 1, pp. 111–123, 2012. [Online]. Available: <http://www.publish.csiro.au/?paper=RJ11044>

- [10] C. Umstatter, "The evolution of virtual fences: A review," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 10–22, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.compag.2010.10.005>
- [11] F. D. Provenza, J. J. Villalba, C. D. Cheney, and S. J. Werner, "Self-organization of foraging behaviour: from simplicity to complexity without goals." *Nutrition research reviews*, vol. 11, no. 2, pp. 199–222, 1998. [Online]. Available: <http://journals.cambridge.org/abstract{ }S095442249800016X>
- [12] C. Umstatter, "The evolution of virtual fences: A review," pp. 10–22, 2011.
- [13] T. Wark, P. Corke, P. Sikka, L. Klingbeil, G. Ying, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 50–57, apr 2007. [Online]. Available: <http://ieeexplore.ieee.org/document/4160605/>
- [14] R. Francisco and N. Dias, "MINISTÉRIO DA AGRICULTURA , DO MAR , Gabinete da Ministra," *Diário da República*, pp. 2011–2012, 2012.
- [15] A. R. Calderbank and A. F. Naguib, "Introduction to space-time codes," in *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, 2006, vol. 9780521851, pp. 133–153. [Online]. Available: <https://pdfs.semanticscholar.org/e25b/c9b91d2819bb1ccc71ecdc6907fdb09c0c88.pdf>
- [16] S. M. Redl, M. K. Weber, and M. W. Oliphant, *An introduction to GSM*. Artech House, 1995. [Online]. Available: <http://cds.cern.ch/record/407371>
- [17] G. Miao, J. Zander, K. Sung, and S. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016. [Online]. Available: <https://books.google.pt/books?id=ImeSCwAAQBAJ>
- [18] H. George D., R. L. Lien, J. M. Lind, and V. H. MacDonald, "High density cellular mobile radio communications," 1983. [Online]. Available: <https://www.google.com/patents/US4475010>
- [19] M. Rahnema, "Overview Of The GSM System and Protocol Architecture," *IEEE Communications Magazine*, vol. 31, no. 4, pp. 92–100, 1993. [Online]. Available: <http://www.ciens.ucv.ve:8080/genasig/sites/redesmov/archivos/GSM.pdf>

- [20] C. E. Lin, C.-C. Li, A.-S. Hou, and C.-C. Wu, "A Real-Time Remote Control Architecture Using Mobile Communication," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 52, no. 4, 2003. [Online]. Available: <https://pdfs.semanticscholar.org/3c35/783dda519b3c5c632a2d65cab2bb456bc3c5.pdf>
- [21] G. Heitink, *Practical Theology: History, Theory, Action Domains: Manual for Practical Theology*. Studies in practical theology, 1999. [Online]. Available: [https://books.google.pt/books?id=X\\_O2d-61pkQC](https://books.google.pt/books?id=X_O2d-61pkQC)
- [22] O. H. Mowrer, "A STIMULUS-RESPONSE ANALYSIS OF ANXIETY AND ITS ROLE AS A REINFORCING AGENT 1," *Psychological Review*, vol. 46, pp. 553–65, 1939. [Online]. Available: <http://s-f-walker.org.uk/2007intranet/7thwk/mowrer39.pdf>
- [23] C. J. Warden and M. Aylesworth, "The relative value of reward and punishment in the formation of a visual discrimination habit in the white rat," *Journal of comparative psychology*, vol. 7, no. 2, pp. 117–127, 1927. [Online]. Available: <http://psycnet.apa.org/journals/com/7/2/117/>
- [24] I. P. Pavlov and I. M. Sechenov, *Physiological Journal of the USSR. , Volume 52, A. o. S. o. t. U. All-Union Physiological Society named after I.P. Pavlova, Ed. "Science" (Leningrad branch) ,, 1966*. [Online]. Available: <https://books.google.pt/books?id=nJlKAAAAYAAJ&num=10>
- [25] N. Mackintosh, *The psychology of animal learning*. Academic Press, 1974, vol. 23. [Online]. Available: <https://books.google.pt/books?id=PfnaAAAAMAAJ>
- [26] J. J. Plaud, *Behavior analysis and the challenge of a scientific psychology: Review of behavior modification in the human services*. CRC Press, 1993, vol. 24, no. 3. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/000579169390032R>
- [27] R. M. Fish and L. a. Geddes, "Conduction of electrical current to and through the human body: a review." *Eplasty*, vol. 9, p. e44, oct 2009. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19907637http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2763825>
- [28] W. E. Crusio, *Mouse behavioral testing. How to use mice in behavioral research - by Douglas Wahlsten*. Academic, 2013, vol. 12, no. 2. [Online]. Available: <http://doi.wiley.com/10.1111/j.1601-183X.2012.00864.x>

- [29] G. S. Ohm, *Die galvanische Kette - Mathematisch bearbeitet*. Berlin: T. H. Riemann, 1827. [Online]. Available: <https://books.google.pt/books?id=0V5dAAAAcAAJ>
- [30] R. Ader, J. A. W. M. Weijnen, and P. Moleman, "Retention of a passive avoidance response as a function of the intensity and duration of electric shock," *Psychonomic Science*, vol. 26, no. 3, pp. 125–128, 1972. [Online]. Available: <https://link.springer.com/content/pdf/10.3758/BF03335453>  
<http://link.springer.com/10.3758/BF03335453>
- [31] G. H. Bower and N. E. Miller, "Rewarding and punishing effects from stimulating the same place in the rat's brain," *Journal of Comparative and Physiological Psychology*, vol. 51, no. 6, pp. 669–674, 1958. [Online]. Available: <http://web.stanford.edu/~gbower/1958/Rewarding{ }punishing{ }brain{ }stimulation.pdf>
- [32] A. Gelfond and U. K. M. F. P. Islam, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*, 1960. [Online]. Available: <https://books.google.pt/books?id=pxpunQAACAAJ>
- [33] M. E. Bouton, "Context, time, and memory retrieval in the inference paradigms of pavlovian learning," *Psychological Bulletin*, vol. 114, no. 1, pp. 80–99, 1993.
- [34] Raspberry Pi Foundation, "Raspberry Pi 3 Model B," 2016. [Online]. Available: <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>
- [35] U-blox, "NEO-6 GPS Modules Data Sheet," p. 25, 2011. [Online]. Available: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6{ }DataSheet{ }\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6{ }DataSheet{ }(GPS.G6-HW-09005).pdf)
- [36] SIMCom, "SIM800L\_Hardware\_Design\_V1.00," SimCom, Tech. Rep., 2013. [Online]. Available: <http://wiki.seeedstudio.com/images/4/46/SIM800L{ }Hardware{ }Design{ }V1.00.pdf>
- [37] C. K. H. Chun-Ping Lin, "USB/UART converter and its control method," oct 1998. [Online]. Available: <https://www.google.com/patents/US6151653>
- [38] FTDI, "Ft232R Usb Uart Ic," *Technology*, pp. 1–40, 2008. [Online]. Available: <http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS{ }FT232R.pdf>

- [39] Xiaomi, "Mi Power Bank 10000mAh," 2017. [Online]. Available: <http://www.mi.com/en/pb10000/{#}params>
- [40] Botnroll.com, "Conjunto de Leds 5mm - botnroll.com," 2017. [Online]. Available: <http://www.botnroll.com/pt/leds/780-conjunto-de-leds-5mm-50-pcs.html?search{ }query=led{&}results=420>
- [41] R. Wilson and M. Patel, "Google Lat Long: Changing your perspective," *Google Imagery Team*, 2009. [Online]. Available: <https://maps.googleblog.com/2009/12/changing-your-perspective.html>
- [42] G. Svennerberg, *Beginning Google Maps API 3*. Apress, 2010. [Online]. Available: <https://books.google.pt/books?id=gHb8m0GSV2MC>
- [43] Google, "Google Maps/Google Earth APIs Terms of Service - Google Maps API Family - Google Code," 2011. [Online]. Available: <https://developers.google.com/maps/terms?csw=1http://code.google.com/intl/en/apis/maps/terms.html>
- [44] System Development Corporation, "Technical Base Program in Information Science," *Burroughs Corporation records. System Development Corporation records., . Charles Babbage Institute Archives.*, 1964. [Online]. Available: <http://archives.lib.umn.edu/repositories/3/archival{ }objects/46939>
- [45] Diafaan, "Diafaan SMS Server | Diafaan SMS Server," 2017. [Online]. Available: <https://www.diafaan.com/diafaan-sms-server/>
- [46] P. Bourke, C. Fuhrman, and P. Dephi, "Calculating The Area And Centroid Of A Polygon," no. C, pp. 3–5, 1988. [Online]. Available: <http://www.seas.upenn.edu/{~}sys502/extra{ }materials/PolygonAreaandCentroid.pdf>
- [47] M. Banks, *The Modem Reference: The Complete Guide to PC Communications*, ser. Communications/Computers. CyberAge Books, 2000. [Online]. Available: <https://books.google.pt/books?id=oeEQAQAAMAAJ>
- [48] Telit Communications S.p.A., "AT Commands Reference Guide," *Telit Communications S.p.A.*, 2012. [Online]. Available: <https://www.sparkfun.com/datasheets/CellularModules/AT{ }Commands{ }Reference{ }Guide{ }r0.pdf>
- [49] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language ( WSDL ) 1 . 1 Web Service



Definition Language ( WSDL ),” W3C, no. March 2001, pp. 1–28, 2009. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.460.467&rep=rep1&type=pdf>

- [50] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext Transfer Protocol – HTTP/1.1,” Tech. Rep., 1999. [Online]. Available: <https://tools.ietf.org/html/rfc2616{#}page-53https://www.rfc-editor.org/info/rfc2616>
- [51] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the Limits of LoRaWAN,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017. [Online]. Available: <https://arxiv.org/pdf/1607.08011.pdf>
- [52] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grövlén, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, “A primer on 3gpp narrowband internet of things (nb-iot),” *arXiv preprint*, 2016.
- [53] Nokia, “LTE evolution for IoT connectivity,” pp. 1–18, 2017. [Online]. Available: [www.nokia.com](http://www.nokia.com)
- [54] Hologram, “Cellular Connectivity No Cellular Connectivity.” [Online]. Available: <https://hologram.io/cellular/>

## Appendix A

The Appendix chapter purpose is to exhibit some of the work realised in this project context. In this chapter appear implementation code from the collar algorithm and the website development revealing the access to the database logs.

PHP code to obtain the SMS string from the Diafaan software and update the animal localisation.

```
<?php
//Get SMS string - ID Date Time Zone Latitude Longitude
$string = $_POST["message"];
$pieces = explode(",", $string);
$id_collar= $pieces[0];
$a=explode(",", $pieces[1]);
$date=$a[0];
$b=explode(",", $a[1]);
$time=$b[0];
$c=explode(",", $b[1]);
$zone=$c[0];
$d=explode(",", $c[1]);
$lat=$d[0];
$e=explode(",", $d[1]);
$lng=$e[0];

//Connect to server and select database
$connection=mysqli_connect('localhost','id2316921_virtualfencefield',
    'virtualfencefield','id2316921_vffield');
if (!$connection) {
    die('Could not connect: ' . mysqli_error($connection));
}

//Update animal coordinates
$update_collarCoord=mysqli_query($connection,"UPDATE collar_coord
    SET lat='$lat',lng='$lng',type=$zone WHERE id_collar=$id_collar")
;
echo mysqli_error($connection);

mysqli_close($connection);
?>
```

PHP code to connect to database and obtain the original fence data (number of vertices/points, latitudes and longitudes).

```

<?php
$connection=mysqli_connect('localhost','id2316921_virtualfencefield',
    'virtualfencefield','id2316921_vffield');
if(!$connection){
    echo "Connection error";
    exit;
}

$result=mysqli_query($connection,"SELECT id_fence FROM field WHERE
    id_user='$id'");

while($row=mysqli_fetch_row($result))
{
    $rows[] = $row[0];
}

$n=mysqli_num_rows($result);
$id_fence =$rows;

$get_ndots=mysqli_query($connection,"SELECT n_dots FROM fence
    WHERE id_fence='$id_fence[$i]'");
while($n_dot=mysqli_fetch_row($get_ndots))
{
    $n_dots[] = $n_dot[0];
}

$get_lat=mysqli_query($connection,"SELECT lat FROM fence_dots
    WHERE id_fence='$id_fence[$i]'");
while($lat=mysqli_fetch_row($get_lat))
{
    $lats[] = $lat[0];
}

$get_lng=mysqli_query($connection,"SELECT lng FROM fence_dots
    WHERE id_fence='$id_fence[$i]'");
while($lng=mysqli_fetch_row($get_lng))
{
    $lngs[] = $lng[0];
}

mysqli_close($connection);
?>

```

Code to obtain the center coordinates from the original fence polygon.

```

void Cboundaries::get_center(double *Clat, double *Clng, QList<
    double> lats, QList<double> lngs)
{
    double area=0;
    for(int i=0;i<count-1;i++)
    {
        area=area + lngs[i]*lats[i+1] - lngs[i+1]*lats[i];
    }
    area=area + lngs[count-1]*lats[0] - lngs[0]*lats[count-1];
    area=area/2;

    for(int i=0;i<count-1;i++)
    {
        *Clng= *Clng + (lngs[i] + lngs[i+1])*(lngs[i]*lats[i+1]-lngs[
            i+1]*lats[i]) ;
    }
    *Clng= *Clng + (lngs[count-1] + lngs[0])*(lngs[count-1]*lats[0]-
        lngs[0]*lats[count-1]) ;
    *Clng=*Clng/(6*area);

    for(int i=0;i<count-1;i++)
    {
        *Clat= *Clat + (lats[i] +lats[i+1])*(lngs[i]*lats[i+1]-lngs[i
            +1]*lats[i]) ;
    }
    *Clat= *Clat + (lats[count-1] +lats[0])*(lngs[count-1]*lats[0]-
        lngs[0]*lats[count-1]) ;
    *Clat=*Clat/(6*area);
}

```

Code to obtain the extra fence(s) polygon data.

```

QPolygonF Cboundaries::getExtrafence(QPolygonF fence, double Clat,
double Clng, double reduc)
{
    QPolygonF fence_reduce;
    fence_reduce= fence.translated(-Clng,-Clat);
    QTransform trans(QTransform::fromScale(reduc,reduc));
    fence_reduce=trans.map(fence_reduce);
    fence_reduce.translate(Clng,Clat);
    return fence_reduce;
}

```

Figures 62 to 66 reveal the GPS output tests realised in different zones when the collar was carried by the project developer. The collar GPS latitudes and longitudes are represented inside the red boxes. The signal strength is represented inside the yellow boxes.

```

$GPRMC,171608.00,A,4143.95625,N,00837.60109,W,2.742,26.94,250917,,A*7B
$GPVTG,216.94,T,,M,2.742,N,5.079,K,A*3D
$GPGGA,171608.00,4143.95625,N,00837.60109,W,1,07,1.2832.1,M,50.5,M,,*76
$GPGSA,A,3,09,06,23,02,07,19,17,,,,,2.42,1.28,2.05*0
$GPGSV,2,1,07,02,32,311,25,06,73,277,28,07,31,149,13,9,79,051,24*7E
$GPGSV,2,2,07,17,14,206,21,19,26,224,17,23,47,052,29*A
$GPGLL,4143.95625,N,00837.0109,W,171608.00,A,A*7F
$GPRMC,171609.00,A,4143.95564,N,00837.60181,W,2.530,28.77,250917,,A*78
$GPVTG,218.77,T,,M,2.530,N,4.685,K,A*3D
$GPGGA,171609.00,4143.95564,N,00837.60181,W,1,07,1.2832.9,M,50.5,M,,*79
$GPGSA,A,3,09,06,23,02,07,19,17,,,,,2.42,1.28,2.05*0

```

FIGURE 62. GPS under wood ceiling Results

```

$GPGLL,4143.95087,N,00837.60803,W,171619.00,A,A*72
$GPRMC,171620.00,A,4143.95015,N,00837.60847,W,2.411,,50917,,A*64
$GPVTG,,T,,M,2.411,N,4.466,K,A*25
$GPGGA,171620.00,4143.95015,N,00837.60847,W,1,06,1.3135.7,M,50.5,M,,*72
$GPGSA,A,3,09,06,23,02,07,9,,,,,2.59,1.31,2.23*0F
$GPGSV,2,1,08,02,33,311,23,03,20,089,25,06,73,277,30,7,31,149,13*7D
$GPGSV,2,2,08,09,79,051,25,17,14,206,,19,26,224,18,2347,052,30*75
$GPGLL,4143.95015,N,00837.60847,W,171620.00,A,A*73
$GPRMC,171621.00,A,4143.94992,N,00837.60910,W,1.965,,50917,,A*6C
$GPVTG,,T,,M,1.965,N,3.638,K,A*26
$GPGGA,171621.00,4143.94992,N,00837.60910,W,1,06,1.3135.8,M,50.5,M,,*78

```

FIGURE 63. GPS under plastic ceiling Results

```

$GPRMC,171608.00,A,4143.95625,N,00837.60109,W,2.742,26.94,250917,,A*7B
$GPVTG,216.94,T,,M,2.742,N,5.079,K,A*3D
$GPGGA,171608.00,4143.95625,N,00837.60109,W,1,07,1.2832.1,M,50.5,M,,*76
$GPGSA,A,3,09,06,23,02,07,19,17,,,,,2.42,1.28,2.05*0
$GPGSV,2,1,07,02,32,311,25,06,73,277,28,07,31,149,13,9,79,051,24*7E
$GPGSV,2,2,07,17,14,206,21,19,26,224,17,23,47,052,29*A
$GPGLL,4143.95625,N,00837.0109,W,171608.00,A,A*7F
$GPRMC,171609.00,A,4143.95564,N,00837.60181,W,2.530,28.77,250917,,A*78
$GPVTG,218.77,T,,M,2.530,N,4.685,K,A*3D
$GPGGA,171609.00,4143.95564,N,00837.60181,W,1,07,1.2832.9,M,50.5,M,,*79
$GPGSA,A,3,09,06,23,02,07,19,17,,,,,2.42,1.28,2.05*0

```

FIGURE 64. GPS under aluminium ceiling Results

```

$GPGLL,4143.95015,N,00837.60847,W,173620.00,A,A*73
$GPRMC,173621.00,A,4143.94992,N,00837.60910,W,1.965,,250917,,A*6C
$GPGSV,2,1,08,02,33,311,13,03,20,089,24,06,73,277,30,7,31,149,15*7A
$GPGSV,2,2,08,09,79,051,16,17,14,206,05,19,26,224,18,3,47,052,29*7B
$GPVTG,,T,,M,1.965,N,3.638,K,A*26
$GPVTG,,,,,N*30
$GPGGA,173621.00,4143.94992,N,00837.60910,W,1,06,1.3135.8,M,50.5,M,,*78
$GPGSV,2,1,08,02,32,311,6,03,20,089,28,06,73,277,29,7,31,149,08*76
$GPGSV,2,2,08,09,79,051,3,17,14,206,,19,26,224,16,2347,052,29*75
$GPGLL,,,,,172248.00,V,A*7
$GPRMC,172249.00,V,,,,,250917,,A*68

```

FIGURE 65. GPS dense forest Results

```

$GPGLL,,,,,172248.00,V,A*7
$GPRMC,172249.00,V,,,,,50917,,A*68
$GPVTG,,T,,M,2.047,N,3.791,K,A*2E
$GPVTG,,,,,N*30
$GPGSA,A,1,,,,,99.99,99.99,99.99,99.99*30
$GPRMC,173249.00,V,,,,,50917,,A*68
$GPGSV,2,1,08,02,33,311,4,03,20,089,13,06,74,276,27,7,31,149,15*7A
$GPGSV,2,2,08,09,79,050,2,17,14,206,11,19,26,224,21,3,47,052,21*73

```

FIGURE 66. GPS inside stone house Results

Figure 67 reveals the system output from a test where the collar was transported by the project developer. It is visible that the collar went from the area "2", crossing the yellow fence to the area "1".

"0,180118,143121.00	2	41.5524,-8.38806"
parsing		
"\$GPRMC"		
"0,180118,143131.00	2	41.5524,-8.38805"
parsing		
"\$GPRMC"		
"0,180118,143141.00	2	41.5524,-8.38805"
parsing		
"\$GPRMC"		
"0,180118,143151.00	1	41.5524,-8.38798"
parsing		
"\$GPRMC"		
"0,180118,143211.00	1	41.5524,-8.38794"
parsing		
"\$GPRMC"		
"0,180118,143231.00	1	41.5523,-8.38794"
parsing		
"\$GPRMC"		
"0,180118,143251.00	1	41.5523,-8.38792"
parsing		
"\$GPRMC"		
"0,180118,143311.00	1	41.5523,-8.38791"
parsing		
"\$GPRMC"		
"0,180118,143331.00	1	41.5524,-8.38792"
parsing		
"\$GPRMC"		

FIGURE 67. System Results 2





# Annex

## AT Commands Reference Guide

### 3.5.2.4.2 +CFUN - Set Phone Functionality

<b>+CFUN - Set Phone Functionality</b>	
<b>AT+CFUN=&lt;fun&gt;</b>	<p>Set command selects the level of functionality in the <b>ME</b>.</p> <p>Parameter:  <b>&lt;fun&gt;</b> - is the power saving function mode            0 - minimum functionality, NON-CYCLIC SLEEP mode: in this mode, the AT interface is not accessible. Consequently, once you have set <b>&lt;fun&gt;</b> level</p>

### AT Commands Reference Guide

80000ST10025a Rev. 0 - 04/08/06

	<p>0, do not send further characters. Otherwise these characters remain in the input buffer and may delay the output of an unsolicited result code. The first wake-up event stops power saving and takes the ME back to full functionality level <b>&lt;fun&gt;=1</b>.</p> <p>1 - mobile full functionality with power saving disabled (factory default)            2 - disable TX            4 - disable either TX and RX            5 - mobile full functionality with power saving enabled</p> <p>Note: if power saving enabled, it reduces the power consumption during the idle time, thus allowing a longer standby time with a given battery capacity.</p> <p>Note: to place the telephone in power saving mode, set the <b>&lt;fun&gt;</b> parameter at value = 5 and the line <b>DTR</b> (RS232) must be set to <b>OFF</b>. Once in power saving, the <b>CTS</b> line switch to the <b>OFF</b> status to signal that the telephone is really in power saving condition.            During the power saving condition, before sending any AT command on the serial line, the <b>DTR</b> must be enabled and it must be waited for the <b>CTS</b> (RS232) line to go in <b>ON</b> status.            Until the <b>DTR</b> line is <b>ON</b>, the telephone will not return back in the power saving condition.</p> <p>Note: the power saving function does not affect the network behavior of the MODULE, even during the power save condition the module remains registered on the network and reachable for incoming calls or SMS. If a call arrives during the power save, then the module will wake up and proceed normally with the unsolicited incoming call code</p>
<b>AT+CFUN?</b>	Read command reports the current level of functionality.
<b>AT+CFUN=?</b>	<p>Test command returns the list of supported values for <b>&lt;fun&gt;</b></p> <p>For compatibility with previous versions, Test command returns <b>+CFUN: (1, 5)</b></p> <p>An enhanced version of Test command has been defined: <b>AT+CFUN=??</b>, that provides the complete range of values for <b>&lt;fun&gt;</b>.</p>
<b>AT+CFUN=??</b>	Enhanced test command returns the list of supported values for <b>&lt;fun&gt;</b>
Reference	GSM 07.07

### 3.5.2.4.4 +CSQ - Signal Quality

<b>+CSQ - Signal Quality</b>	
<b>AT+CSQ</b>	<p>Execution command reports received signal quality indicators in the form:</p> <p><b>+CSQ:&lt;rss&gt;,&lt;ber&gt;</b>            where  <b>&lt;rss&gt;</b> - received signal strength indication            0 - (-113) dBm or less            1 - (-111) dBm            2..30 - (-109)dBm..(-53)dBm / 2 dBm per step            31 - (-51)dBm or greater            99 - not known or not detectable</p> <p><b>&lt;ber&gt;</b> - bit error rate (in percent)            0 - less than 0.2%            1 - 0.2% to 0.4%            2 - 0.4% to 0.8%            3 - 0.8% to 1.6%            4 - 1.6% to 3.2%            5 - 3.2% to 6.4%            6 - 6.4% to 12.8%            7 - more than 12.8%</p>

### AT Commands Reference Guide

80000ST10025a Rev. 0 - 04/08/06

<b>+CSQ - Signal Quality</b>	
	<p>99 - not known or not detectable</p> <p>Note: this command should be used instead of the %Q and %L commands, since GSM relevant parameters are the radio link ones and no line is present, hence %Q %L and have no meaning.</p>
<b>AT+CSQ?</b>	Read command has the same effect as Execution command.
<b>AT+CSQ=?</b>	<p>Test command returns the supported range of values of the parameters <b>&lt;rss&gt;</b> and <b>&lt;ber&gt;</b>.</p> <p>Note: although <b>+CSQ</b> is an execution command, ETSI 07.07 requires the Test command to be defined.</p>
Reference	GSM 07.07

### 3.5.3.1.3 +CMGF - Message Format

<b>+CMGF - Message Format</b>	
<b>AT+CMGF=[&lt;mode&gt;]</b>	<p>Set command selects the format of messages used with send, list, read and write commands.</p> <p>Parameter:  <b>&lt;mode&gt;</b>            0 - PDU mode, as defined in GSM 3.40 and GSM 3.41 (factory default)            1 - text mode</p> <p>Note: issuing <b>AT+CMGF&lt;CR&gt;</b> is the same as issuing the Read command.</p> <p>Note: issuing <b>AT+CMGF=&lt;CR&gt;</b> is the same as issuing the command <b>AT+CMGF=0&lt;CR&gt;</b>.</p>
<b>AT+CMGF?</b>	Read command reports the current value of the parameter <b>&lt;mode&gt;</b> .
<b>AT+CMGF=?</b>	Test command reports the supported value of <b>&lt;mode&gt;</b> parameter.
Reference	GSM 07.05

### 3.5.3.3.2 +CMGL - List Messages

<b>+CMGL - List Messages</b>	
<b>AT+CMGL</b> <b>[=&lt;stat&gt;]</b>	<p>Execution command reports the list of all the messages with status value <b>&lt;stat&gt;</b> stored into <b>&lt;memr&gt;</b> message storage (<b>&lt;memr&gt;</b> is the message storage for read and delete SMS as last settings of command <b>+CPMS</b>).</p> <p>The parameter type and the command output depend on the last settings of command <b>+CMGF</b> (message format to be used)</p> <p style="text-align: center;"><b>(PDU Mode)</b></p> <p>Parameter:  <b>&lt;stat&gt;</b>            0 - new message            1 - read message            2 - stored message not yet sent            3 - stored message already sent            4 - all messages.</p> <p>Each message to be listed is represented in the format:</p> <p><b>+CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b></p> <p>where  <b>&lt;index&gt;</b> - message position in the memory storage list.  <b>&lt;stat&gt;</b> - status of the message  <b>&lt;length&gt;</b> - length of the PDU in bytes  <b>&lt;pdu&gt;</b> - message in PDU format according to GSM 3.40</p> <p style="text-align: center;"><b>(Text Mode)</b></p> <p>Parameter:  <b>&lt;stat&gt;</b></p>

<b>+CMGL - List Messages</b>	
	<p>"REC UNREAD" - new message  "REC READ" - read message  "STO UNSENT" - stored message not yet sent  "STO SENT" - stored message already sent  "ALL" - all messages.</p> <p>Each message to be listed is represented in the format (the information written in italics will be present depending on <b>+CSDH</b> last setting):</p> <p><b>+CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;oa/da&gt;,,[,&lt;tooa/toda&gt;,&lt;length&gt;]  &lt;CR&gt;&lt;LF&gt; &lt;data&gt;</b></p> <p>where  &lt;index&gt; - message position in the storage  &lt;stat&gt; - message status  &lt;oa/da&gt; - originator/destination number  &lt;tooa/toda&gt; - type of number &lt;oa/da&gt;  129 - number in national format  145 - number in international format (contains the "+")  &lt;length&gt; - text length  &lt;data&gt; - TP-User-Data</p> <p>Each message delivery confirm is represented in the format:</p> <p><b>+CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;fo&gt;,&lt;mr&gt;,,,&lt;scts&gt;,&lt;dt&gt;,&lt;st&gt;</b></p> <p>where  &lt;index&gt; - message position in the storage  &lt;stat&gt; - message status  &lt;fo&gt; - first octet of the message PDU  &lt;mr&gt; - message reference number  &lt;scts&gt; - arrival time of the message to the SC  &lt;dt&gt; - sending time of the message  &lt;st&gt; - message status as coded in the PDU</p> <p>Note: <b>OK</b> result code is sent at the end of the listing.</p> <p>Note: If parameter is omitted the command returns the list of sms with "REC UNREAD" status.</p>
<b>AT+CMGL?</b>	Read command has the same effect as Execution command with parameter omitted
<b>AT+CMGL=?</b>	Test command returns a list of supported <b>&lt;stat&gt;s</b>
Note	<p>If Text Mode (<b>+CMGF=1</b>) the Test command output is not included in parenthesis</p> <p><b>AT+CMGL=?  +CMGL: "REC UNREAD","REC READ","STO UNSENT",</b></p>

<b>+CMGL - List Messages</b>	"STO SENT", "ALL"
Note	The improving command <b>@CMGL</b> has been defined
Reference	GSM 07.05

### 3.5.3.3.3 @CMGL - List Messages

<b>@CMGL - List Messages</b>	
<b>AT@CMGL</b> <b>[=&lt;stat&gt;]</b>	<p>Execution command reports the list of all the messages with status value <b>&lt;stat&gt;</b> stored into <b>&lt;memr&gt;</b> message storage (<b>&lt;memr&gt;</b> is the message storage for read and delete SMS as last settings of command <b>+CPMS</b>).</p> <p>The parameter type and the command output depend on the last settings of command <b>+CMGF</b> (message format to be used)</p> <p style="text-align: center;"><b>(PDU Mode)</b></p> <p>Parameter: <b>&lt;stat&gt;</b> 0 - new message 1 - read message 2 - stored message not yet sent 3 - stored message already sent 4 - all messages.</p> <p>Each message to be listed is represented in the format:</p> <p><b>@CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b></p> <p>where <b>&lt;index&gt;</b> - message position in the memory storage list. <b>&lt;stat&gt;</b> - status of the message <b>&lt;length&gt;</b> - length of the PDU in bytes <b>&lt;pdu&gt;</b> - message in PDU format according to GSM 3.40</p> <p style="text-align: center;"><b>(Text Mode)</b></p> <p>Parameter: <b>&lt;stat&gt;</b> "REC UNREAD" - new message "REC READ" - read message "STO UNSENT" - stored message not yet sent "STO SENT" - stored message already sent "ALL" - all messages.</p> <p>Each message to be listed is represented in the format (the information written in italics will be present depending on <b>+CSDH</b> last setting):</p>

<b>@CMGL - List Messages</b>	
	<p><b>@CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;oa/da&gt;,,[,&lt;toa/toda&gt;,&lt;length&gt;]</b>  <b>&lt;CR&gt;&lt;LF&gt; &lt;data&gt;</b></p> <p>where  <b>&lt;index&gt;</b> - message position in the storage  <b>&lt;stat&gt;</b> - message status  <b>&lt;oa/da&gt;</b> - originator/destination number  <b>&lt;toa/toda&gt;</b> - type of number <b>&lt;oa/da&gt;</b>            129 - number in national format            145 - number in international format (contains the "+")  <b>&lt;length&gt;</b> - text length  <b>&lt;data&gt;</b> - TP-User-Data</p> <p>Each message delivery confirm is represented in the format:</p> <p><b>@CMGL: &lt;index&gt;,&lt;stat&gt;,&lt;fo&gt;,&lt;mr&gt;,,,&lt;scts&gt;,&lt;dt&gt;,&lt;st&gt;</b></p> <p>where  <b>&lt;index&gt;</b> - message position in the storage  <b>&lt;stat&gt;</b> - message status  <b>&lt;fo&gt;</b> - first octet of the message PDU  <b>&lt;mr&gt;</b> - message reference number  <b>&lt;scts&gt;</b> - arrival time of the message to the SC  <b>&lt;dt&gt;</b> - sending time of the message  <b>&lt;st&gt;</b> - message status as coded in the PDU</p> <p>Note: The command differs from the <b>+CMGL</b> because at the end of the listing a <b>&lt;CR&gt;&lt;LF&gt;</b> is put before the <b>OK</b> result code.</p> <p>Note: If parameter is omitted the command returns the list of sms with "REC UNREAD" status.</p>
<b>AT@CMGL?</b>	Read command has the same effect as Execution command with parameter omitted
<b>AT@CMGL=?</b>	Test command returns a list of supported <b>&lt;stat&gt;s</b>
Note	<p>If Text Mode (<b>+CMGF=1</b>) the Test command output is not included in parenthesis</p> <p><b>AT@CMGL=?</b>  <b>@CMGL: "REC UNREAD","REC READ","STO UNSENT",</b>  <b>"STO SENT","ALL"</b></p>
Reference	GSM 07.05

## 3.5.3.3.4 +CMGR - Read Message

<b>+CMGR - Read Message</b>	
<b>AT+CMGR= &lt;index&gt;</b>	<p>Execution command reports the message with location value <b>&lt;index&gt;</b> from <b>&lt;memr&gt;</b> message storage (<b>&lt;memr&gt;</b> is the message storage for read and delete SMS as last settings of command <b>+CPMS</b>).</p> <p>Parameter: <b>&lt;index&gt;</b> - message index.</p> <p>The output depends on the last settings of command <b>+CMGF</b> (message format to be used)</p> <p style="text-align: center;"><b>(PDU Mode)</b></p> <p>The output has the following format:</p> <p><b>+CMGR: &lt;stat&gt;,&lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b></p> <p>where  <b>&lt;stat&gt;</b> - status of the message  0 - new message  1 - read message  2 - stored message not yet sent  3 - stored message already sent  <b>&lt;length&gt;</b> - length of the PDU in bytes.  <b>&lt;pdu&gt;</b> - message in PDU format according to GSM 3.40.</p> <p>The status of the message and entire message data unit <b>&lt;pdu&gt;</b> is returned.</p> <p style="text-align: center;"><b>(Text Mode)</b></p> <p>Output format for received messages (the information written in italics will be present depending on <b>+CSDH</b> last setting):</p> <p><b>+CMGR: &lt;stat&gt;,&lt;oa&gt;,,&lt;scts&gt; [,&lt;tooa&gt;,&lt;fo&gt;,&lt;pid&gt;,&lt;dcs&gt;,&lt;sca&gt;,&lt;tosca&gt;,&lt;length&gt;]&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b></p> <p>Output format for sent messages:</p> <p><b>+CMGR: &lt;stat&gt;,&lt;da&gt;,&lt;tda&gt;,&lt;fo&gt;,&lt;pid&gt;,&lt;dcs&gt;,,&lt;sca&gt;,&lt;tosca&gt;,&lt;length&gt;]&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b></p> <p>Output format for message delivery confirm:</p> <p><b>+CMGR: &lt;stat&gt;,&lt;fo&gt;,&lt;mr&gt;,,&lt;scts&gt;,&lt;dt&gt;,&lt;st&gt;</b></p> <p>where:  <b>&lt;stat&gt;</b> - status of the message  "REC UNREAD" - new received message unread  "REC READ" - received message read</p>



<b>+CMGR - Read Message</b>	
	<p>"STO UNSENT" - message stored not yet sent            "STO SENT" - message stored already sent            &lt;fo&gt; - first octet of the message PDU            &lt;mr&gt; - message reference number            &lt;scts&gt; - arrival time of the message to the SC            &lt;dt&gt; - sending time of the message            &lt;st&gt; - message status as coded in the PDU            &lt;pid&gt; - Protocol Identifier            &lt;dcs&gt; - Data Coding Scheme            &lt;oa&gt; - Originator address number            &lt;da&gt; - Destination address number            &lt;sca&gt; - Service Centre number            &lt;toa&gt;, &lt;tda&gt;, &lt;tosca&gt; - type of number &lt;oa&gt;, &lt;da&gt;, &lt;sca&gt;            129 - number in national format            145 - number in international format (contains the "+")            &lt;length&gt; - text length            &lt;data&gt; - TP-User_data</p> <p>Note: in both cases if status of the message is 'received unread', status in the storage changes to 'received read'.</p> <p>Note: an error result code is sent on empty record &lt;index&gt;.</p>
AT+CMGR=?	Test command returns the <b>OK</b> result code.
Note	The improving command <b>@CMGR</b> has been defined
Reference	GSM 07.05

### 3.5.3.3.5 @CMGR - Read Message

<b>@CMGR - Read Message</b>	
AT@CMGR= <index>	<p>Execution command reports the message with location value &lt;index&gt; from &lt;memr&gt; message storage (&lt;memr&gt; is the message storage for read and delete SMS as last settings of command <b>+CPMS</b>).</p> <p>Parameter:            &lt;index&gt; - message index.</p> <p>The output depends on the last settings of command <b>+CMGF</b> (message format to be used)</p> <p style="text-align: center;"><b>(PDU Mode)</b></p> <p>The output has the following format:</p> <p><b>@CMGR: &lt;stat&gt;, &lt;length&gt; &lt;CR&gt; &lt;LF&gt; &lt;pdu&gt;</b></p> <p>where            &lt;stat&gt; - status of the message</p>

<b>@CMGR - Read Message</b>	
	<p>0 - new message 1 - read message 2 - stored message not yet sent 3 - stored message already sent <b>&lt;length&gt;</b> - length of the PDU in bytes. <b>&lt;pdu&gt;</b> - message in PDU format according to GSM 3.40.</p> <p>The status of the message and entire message data unit <b>&lt;pdu&gt;</b> is returned.</p> <p style="text-align: center;"><b>(Text Mode)</b></p> <p>Output format for received messages (the information written in italics will be present depending on <b>+CSDH</b> last setting):</p> <p><b>@CMGR: &lt;stat&gt;,&lt;oa&gt;,,&lt;scts&gt; [,&lt;toa&gt;,&lt;fo&gt;,&lt;pid&gt;,&lt;dcs&gt;,&lt;sca&gt;,&lt;tosca&gt;,&lt;length&gt;]&lt;CR&gt;&lt;LF&gt;&lt;text&gt;</b></p> <p>Output format for sent messages: <b>@CMGR: &lt;stat&gt;,&lt;da&gt;,[,&lt;toa&gt;,&lt;fo&gt;,&lt;pid&gt;,&lt;dcs&gt;,,&lt;sca&gt;,&lt;tosca&gt;,&lt;length&gt;]&lt;CR&gt;&lt;LF&gt;&lt;text&gt;</b></p> <p>Output format for message delivery confirm: <b>@CMGR: &lt;stat&gt;,&lt;fo&gt;,&lt;mr&gt;,,,&lt;scts&gt;,&lt;dt&gt;,&lt;st&gt;</b></p> <p>where:</p> <p><b>&lt;stat&gt;</b> - status of the message "REC UNREAD" - new received message unread "REC READ" - received message read "STO UNSENT" - message stored not yet sent "STO SENT" - message stored already sent <b>&lt;fo&gt;</b> - first octet of the message PDU <b>&lt;mr&gt;</b> - message reference number <b>&lt;scts&gt;</b> - arrival time of the message to the SC <b>&lt;dt&gt;</b> - sending time of the message <b>&lt;st&gt;</b> - message status as coded in the PDU <b>&lt;pid&gt;</b> - Protocol Identifier <b>&lt;dcs&gt;</b> - Data Coding Scheme <b>&lt;oa&gt;</b> - Originator address number <b>&lt;da&gt;</b> - Destination address number <b>&lt;sca&gt;</b> - Service Centre number <b>&lt;toa&gt;,&lt;toa&gt;,&lt;toa&gt;,&lt;toa&gt;</b> - type of number <b>&lt;oa&gt;,&lt;da&gt;,&lt;sca&gt;</b> 129 - number in national format 145 - number in international format (contains the "+") <b>&lt;length&gt;</b> - text length <b>&lt;text&gt;</b> - message text</p>

<b>@CMGR - Read Message</b>	
	<p>Note: the command differs from the <b>+CMGR</b> because after the message <b>&lt;pdu&gt;</b> or <b>&lt;text&gt;</b> a <b>&lt;CR&gt;&lt;LF&gt;</b> is put before the <b>OK</b> result code.</p> <p>Note: in both cases if status of the message is 'received unread', status in the storage changes to 'received read'.</p> <p>Note: an error result code is sent on empty record <b>&lt;index&gt;</b>.</p>
<b>AT@CMGR=?</b>	Test command has no effect; the answer is <b>OK</b>
Reference	GSM 07.05

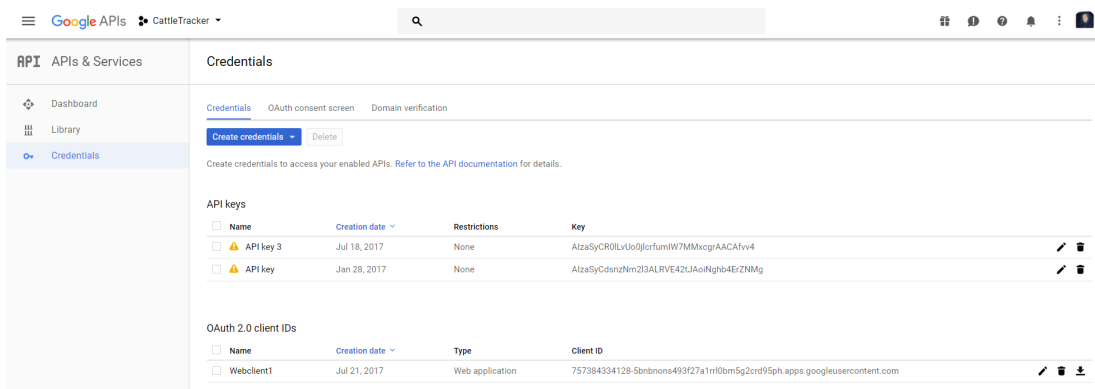
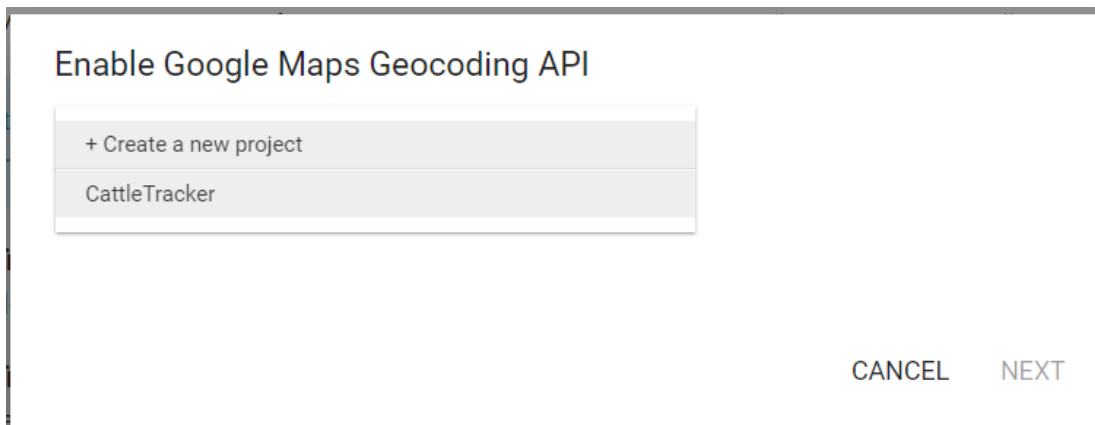
### 3.5.3.4 Message Sending And Writing

#### 3.5.3.4.1 +CMGS - Send Message

<b>+CMGS - Send Message</b>	
<p><i>(PDU Mode)</i> <b>AT+CMGS=</b> <b>&lt;length&gt;</b></p>	<p><b>(PDU Mode)</b> Execution command sends to the network a message.</p> <p>Parameter: <b>&lt;length&gt;</b> - length of the PDU to be sent in bytes. 7..164</p> <p>The device responds to the command with the prompt '&gt;' and waits for the specified number of bytes.</p> <p>To send the message issue <b>Ctrl-Z</b> char (0x1A hex). To exit without sending the message issue <b>ESC</b> char (0x1B hex).</p> <p>If message is successfully sent to the network, then the result is sent in the format:</p> <p><b>+CMGS: &lt;mr&gt;</b> where <b>&lt;mr&gt;</b> - message reference number.</p> <p>Note: if message sending fails for some reason, an error code is reported.</p> <p>Note: care must be taken to ensure that during the command execution, which may take several seconds, no other SIM interacting commands are issued.</p>
<p><i>(Text Mode)</i> <b>AT+CMGS=&lt;da&gt;</b> <b>[,&lt;tda&gt;]</b></p>	<p><b>(Text Mode)</b> Execution command sends to the network a message.</p> <p>Parameters: <b>&lt;da&gt;</b> - destination address number. <b>&lt;tda&gt;</b> - type of destination address 129 - number in national format 145 - number in international format (contains the "+")</p> <p>The device responds to the command with the prompt '&gt;' and waits for message text (max 160 characters).</p> <p>To send the message issue <b>Ctrl-Z</b> char (0x1A hex). To exit without sending the message issue <b>ESC</b> char (0x1B hex).</p> <p>If message is successfully sent to the network, then the result is sent in the format:</p>

<b>+CMGS - Send Message</b>	
	<p><b>+CMGS: &lt;mr&gt;</b> where <b>&lt;mr&gt;</b> - message reference number.</p> <p>Note: if message sending fails for some reason, an error code is reported.</p> <p>Note: care must be taken to ensure that during the command execution, which may take several seconds, no other SIM interacting commands are issued.</p>
Note	To avoid malfunctions is suggested to wait for the <b>+CMGS: &lt;mr&gt;</b> or <b>+CMS ERROR: &lt;err&gt;</b> response before issuing further commands.
Reference	GSM 07.05

Google Maps API credentials.



Google APIs CattleTracker

APIs & Services Client ID for Web application

Client ID: 75738434128-5bnbnons493fz7a1r1r0bmg5zcr95ph.apps.googleusercontent.com  
Client secret: Z10RUSfgJ4Kcn\_SMSHELuG-H  
Creation date: Jul 21, 2017, 11:33:55 PM

Name: Webclient

Restrictions: Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins: https://virtualfenceum.000webhostapp.com, https://www.example.com

Authorized redirect URIs: https://virtualfenceum.000webhostapp.com, https://www.example.com/oauth2/callback

Save Cancel

Google APIs CattleTracker

APIs & Services API key

This API key can be used in this project and with any API that supports it. To use this key in your application, pass it with the `key=API_KEY` parameter.

Creation date: Jul 18, 2017, 7:29:58 PM  
Created by: anianamirabezerra@gmail.com (you)

API key: A1zafyCR8llv08jJlcrfun7n7MycprAACafv4

Name: API key 3

Key restriction: None (selected), HTTP referrers (web sites), IP addresses (web servers, cron jobs, etc.), Android apps, IOS apps

Note: It may take up to 5 minutes for settings to take effect

Save Cancel