MODULAR GSM GPS TRACKING UNIT

ESKOM EXPO REPORT FILE

DANIEL WYKERD
10A3

10 MEI 2018

Contents

Figure Index

Table Index

Introduction / Problem:

A need was identified in our country for a way to retrieve stolen goods especially vehicles like cars, motorcycles, and bicycles as well as personal property.

This project will attempt to solve this problem by designing and manufacturing a modular GPS module.

The need for this device is substantiated by multiple local governmental statistical releases. In a recent document published by the South African Police Service(SAPS) it is revealed there are 53 307 cases of car and motorcycle theft and 16 717 cases of car hijacking between April 2016 and March 2017 (South African Police Service, n.d.). It is also revealed in a statistical release done by Stats SA that there were 21 051 cases of bicycle theft, 284 252 cases of money/purse/wallet theft, and 18 119 cases of bag theft in this period. (Stats SA, 2017).

During the literature study to motivate the project several pre-existing products that prevent some of these problems by using GPS Tracking or alarm systems were identified, but most of these products are dedicated devices designed to only prevent the theft of one of the above-listed items. Research done on these devices can be found in the journal of this project. This project will attempt to design a GPS unit that can be used locate any kind of stolen goods. The device will not be limited to finding only certain objects.

There are devices like TrackingTheWorld's 'Enduro' series trackers that are multipurpose (TrackTheWorld, n.d.), but most of these devices don't offer any means to hide or attach, they are clunky, have a lot of unnecessary features, are uneconomical as they cost upwards of $289.00 and are not locally available to purchase. The device designed in this project will be customizable to suit the needs of the user and will be cheap to mass produce. The main device will have no unnecessary features and will be very small and compact.

Engineering Goal:

The goal of this project is to ultimately design, test, produce and manufacture a modular electronic device that uses GPS to accurately track stolen objects from anywhere in the country. The reason for designing this device is to give users an easy way to retrieve lost or stolen goods by allowing the owner of the goods to view where their goods are located at all times. The device will be modular meaning that you can connect the core tracking module to different modules to add extra functionality and uses to the device to suit the user's needs. The device/circuity should be as small as possible – preferably smaller than an AA-battery in length – and must be mass producible. The circuit should draw a low current as to prolong the device's battery life. There should be no third-party involved in retrieving the coordinates thus the device should use common communication infrastructure like SMS to send the coordinates to the user.

Background Study on Required Components:

To design the tracker, background knowledge of what components will be required to create the device was required. The required components were narrowed down to a microcontroller unit(MCU), which will be used to communicate with all the other components and process their data; a GSM/GPRS module, which will receive user queries (SMSs) and respond to them; a GPS module, which will get the location of the device. This study will provide the reader with background knowledge on the integrated circuits and modules used in this project. It will also explain the reason behind choosing the components.

The Microcontroller Unit (MCU)

A Microcontroller or MCU is a small computer contained on a single integrated circuit (IC). A microcontroller contains one or more processing cores along with memory and programmable input/output peripherals (Wikipedia, 2018).

The MCU will interface with all the other core components of the device, including the GPS and GSM module. It will interpret user input from the GSM module, process the data from the GPS module and output code back to the GSM module to send the location of the device back to the user.

There are plenty of microcontrollers on the market, but the decision was made to use an Arduino as the microcontroller (MCU) for the first prototype. This decision was made because the project leader (Daniel Wykerd) already had experience using Arduino and its IDE. Another deciding factor was the fact that Arduino is open source hardware. There was also decided to make use of an Atmel MCU with an Arduino bootloader burned on it for later prototypes as their MCUs are Arduino compatible. Research was done on multiple ATMega and ATTiny MCUs (all of which are listed in my journal) but there was decided that the ATMega328p MCU was the best option as it is the same MCU used in most Arduino boards, thereby making the design and testing process much easier as this MCU is very well documented online.

The GSM/GPRS Module

The GSM Module is the part of the device that will receive the user queries and will send back the response to the user. It uses a sim card and gsm antenna to connect to pre-existing mobile providers (e.g. Vodacom).

The prime requirement for the module is to be able to send and receive SMSs. Two companies were found during the study that makes modules that meet this requirement: Quectel and SIMCom. Both these companies were well documented, but further research into local availability revealed that Quectel was not readily available in the form of development boards (boards that already have all the support circuitry), thus SIMCom modules were chosen as they were widely available for purchase on development/breakout boards. It was necessary for the modules to be on breakout boards to create the first prototype on a breadboard for testing. There was decided on 3 possible modules: SIM808, SIM800L/H, SIM800C. The SIM808 seemed like a good choice as according to its datasheet it has GPS functionality built in, but it does not have an integrated patch antenna (SIMCom, 2016). The SIM800L/H was also considered as it is readily available on breakout boards and has all the features required and more, however this chip would not be usable for the final prototype as it comes in an LGA (Land Grid Array) package which can only be reflow soldered. Thus, it was considered to use the SIM800C in the final prototype. The SIM800C has less unnecessary features and more importantly, it has an easier to solder SMT layout.

The GPS Module

According to Wikipedia.org, “The Global Positioning System (GPS), originally Navstar GPS, is a satellite-based radio navigation system owned by the United States government and operated by the United States Air Force. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals.” (Wikipedia, 2018).

There was decided on using the GlobalTop Technology Inc.’s FGPMMOPA6H GPS Module as it has an integrated patch antenna which will result in a smaller PCB footprint in the final prototype and fewer costs. The module is also used in Adafruit’s Ultimate GPS module and is well documented on their website. The module outputs its data as NMEA Sentences that can easily be read and interpreted by the MCU

Main Materials / Core Component list:

Below is a breakdown of the components that can possibly be used as well as their prices and what suppliers can be used to purchase them from:

Table 1 List of possible components:

Component	Options	Suppier	Cost
Microcontroller	Arduino UNO/PRO MINI	Communica	R 75.00
	ATMega328p-au	Mouser	$2.48
GSM/GRSM Module	SIM808	Adafruit	$29.90
	SIM800L/H	Communica	R 236.84
	SIM800C	LCSC	$5.63
GPS Module	FGPMMOPA6H	Communica	R635.95

First Prototype:

1. Introduction and Materials Chosen

The goal of the first prototype was to test the concept of making a tracker and to ensure that all the picked components worked properly, thus the components were brought in the form of breakout boards for easy assembly.

Due to the time constraints of this project, the components for the first prototype had to be purchased locally. Thus, Communica was used as the supplier of parts for this prototype. A SIM800L breakout board and Ultimate GPS module (FGPMMOPA6H) module were purchased from them. An Arduino UNO was already purchased beforehand. A breadboard is also required for this prototype See Appendix A for the total cost of this project.

Figure 1 Image of the main components (Wykerd, 2018)

2. Testing the components

Before creating the first prototype the components were tested to ensure that they were in working order.

To test the GPS Module, Adafruit's provided example code and their provided code library found on their website was used. After installing the library found on their website (Adafruit Industries, 2012), the parsing example code which can be found in the project journal on the 20th of January 2018 was used. Below is an image of the wiring:

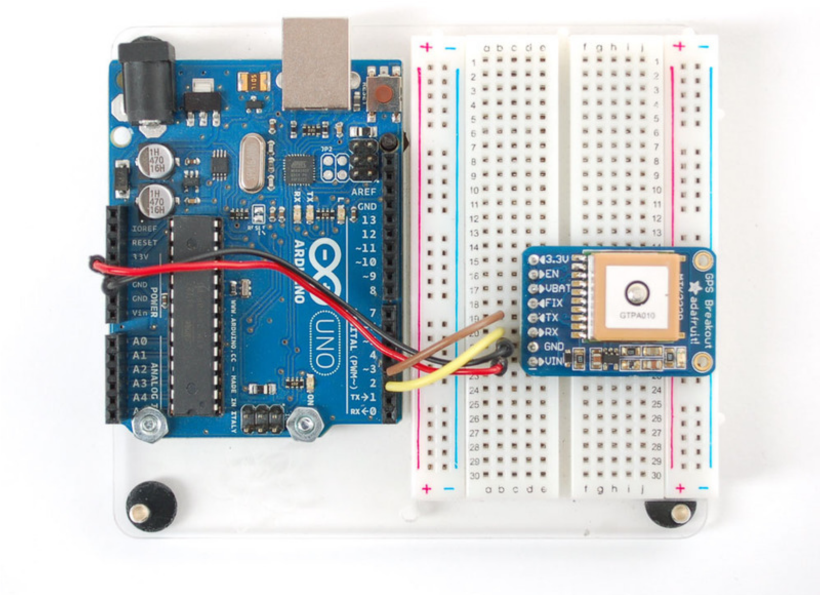


Figure 2 Picture of Ultimate GPS Wiring for testing (Adafruit Industries, 2012)

The test for the GPS was successful and the GPS gave accurate readings.

Next, the GSM module was tested via serial. It was wired like below:

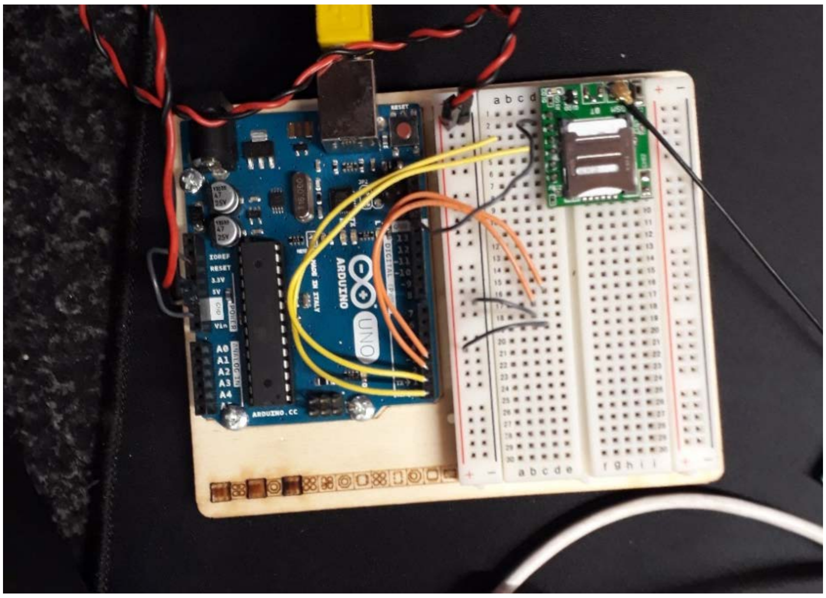


Figure 3 Wiring of the SIM800L module to communicate with the computer (Wykerd, 2018)

This test was also successful. However, the module could not acquire a network connection due to insufficient power. This was solved during the designing of the first prototype by adding a 3,7 V lithium-ion polymer (Li-Po) battery to power the module.

3. Schematic and Breadboard Layout:

The following two figures show the design of the first prototype and was created using Fritzing, a free open-source software that allows you to digitally make your own schematic and breadboard layout.

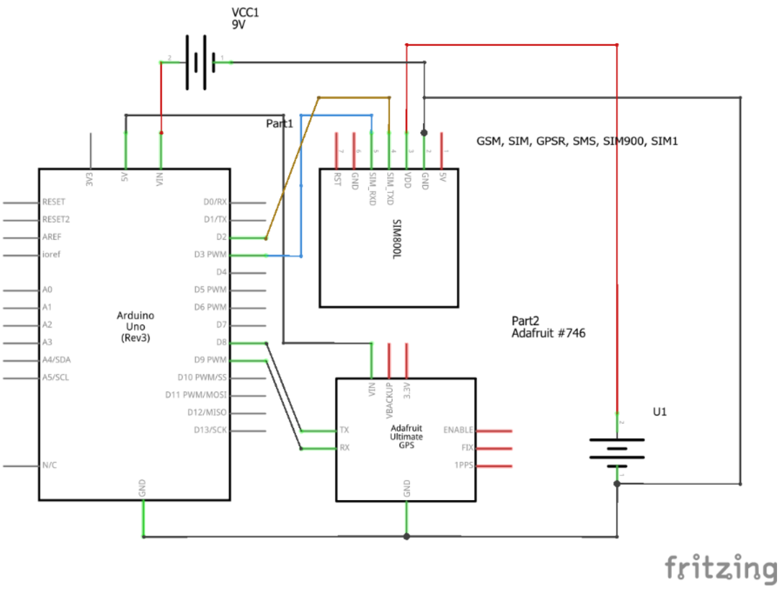


Figure 4 Schematic of first prototype (Wykerd, 2018)

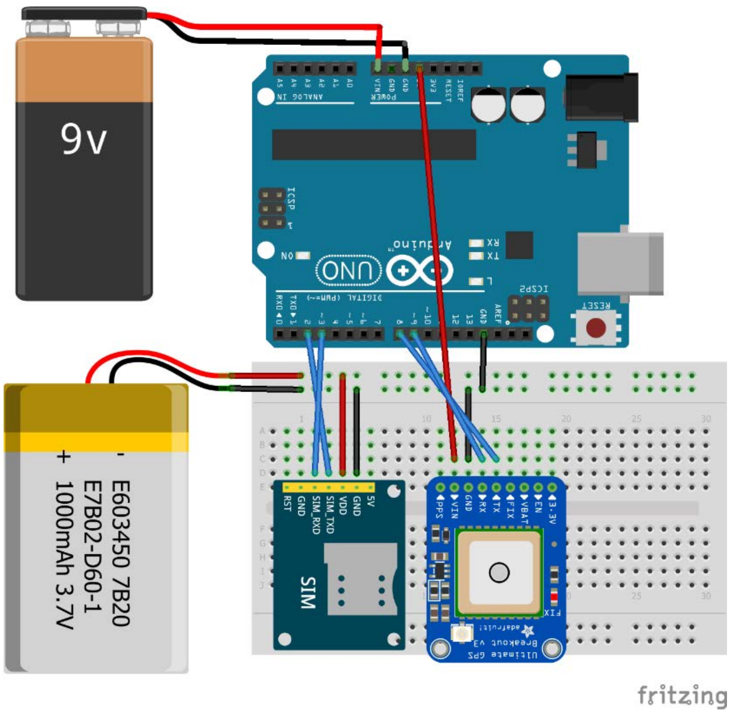


Figure 5 Breadboard layout of first prototype (Wykerd, 2018)

The GSM module will interface with the MCU(Arduino) by wiring the TX (Transmit) and RX (Receive) of the module and digital pin 2, which will act as the MCUs RX (going to TX of module) and 3, which has PWM (Pulse Width Modulation) to transmit commands to the module. The GPS will interface with the module by wiring the module's RX to digital pin 9 of the MCU and the TX to digital pin 8 of the MCU. The interfacing between the modules is required as this is how all the parts will work together.

4. Assembly

The board was assembled on a breadboard using Figure 5 above as reference during the assembly. The hardware was tested beforehand in step 2 on page 7 to decrease the chance that the prototype would fail. The final prototype is shown below in Figure 6.

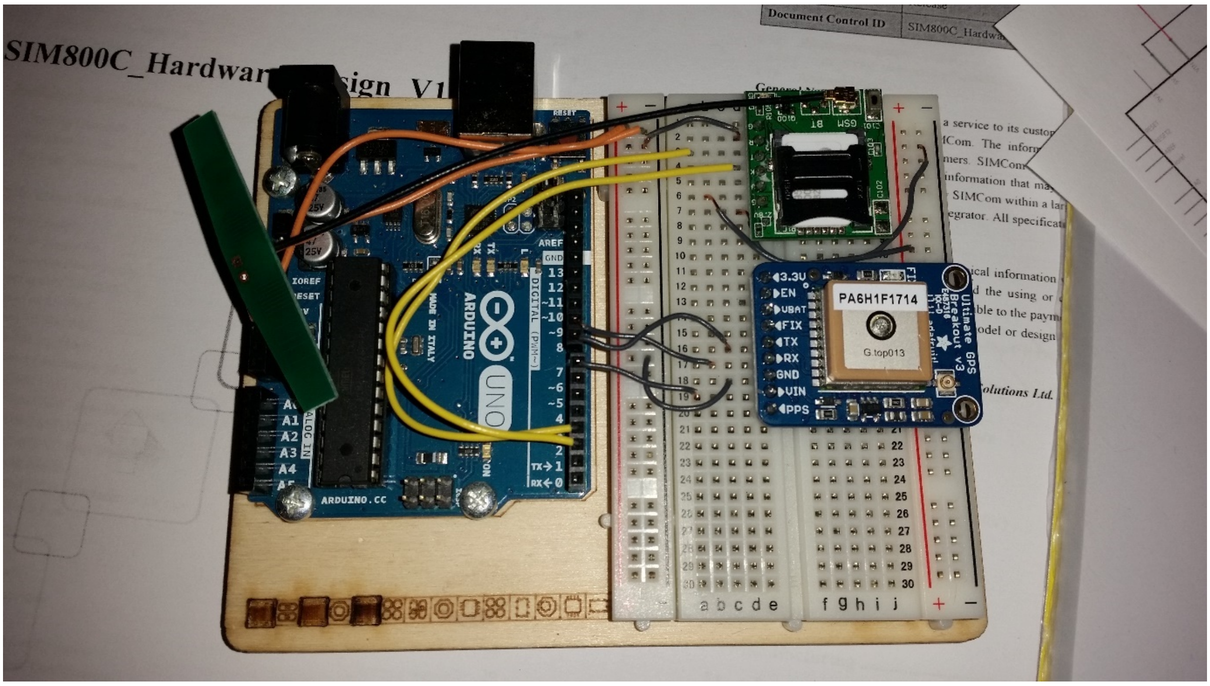


Figure 6 Image of breadboard assembly (Wykerd, 2018)

The wiring of the circuit was done by hand by Daniel Wykerd as was all the other assembly procedures. There was decided on using the power that the Arduino obtained via USB connection to the computer as the power source for the Arduino and GPS Module rather than a 9V-battery.

Another feature was added to the design to allow the user to turn off the GPS module. This is accomplished by connecting the GPS module's VCC to pin 7 of the Arduino instead of directly to the Arduino's 5V output.

5. Software

After the hardware was designed, software had to be created for the MCU so that the device will function. This software was made using the Arduino language which is a set of C/C++ functions.

Firstly, a program that can read and respond to specific SMS commands was created and tested. To communicate with the GSM, Arduino's GSM Library (that was preinstalled on their IDE) was used and their website utilized to gain knowledge of the functions included in the library (Arduino, n.d.). At this point, the GPS was not yet implemented into the code (this code can be found in the project journal on the 6th of February 2018).

Next, it was time to implement the GPS location. To do this Adafruit GPS library was used, but this library clashed with the functions of the GSM library. Thus, an alternative was needed: TinyGPS++, this library was much more lightweight than Adafruit's and AltSoftSerial was also used as an alternative to the IDE's built-in SoftwareSerial library as it also clashed with the GSM library. A revision of the code was created with these libraries and can be found in the journal on the 7th of February 2018 and later the code was further adapted to the more final code available in the journal on the 10th of February 2018.

Lastly, GPS off feature was added (the code of which is available in the journal on the 10th of March). This code was the final code for this prototype. The code can be found in Appendix B.

The software also lets the Arduino output valuable debugging information through the serial port (Rx and Tx) to the computer via USB and can be viewed using a serial monitor application (like the one built into the Arduino's IDE). The debugging output works like follows and an example can be found in figure 7 below.

Upon applying power to the module the line "Boot" is printed to the serial port to verify that the device is indeed booting up. Once the device has registered to a network the line "OK" is printed, this means the device is ready for commands. NOTE: this does not mean the device has a GPS fix. If a message is sent to the device the following lines are printed to serial: The sender's number; the command sent, if the message does not start with a '#' the word "INVALID" will be printed to serial; The word "DEL" meaning that the device has completed the command, deleted the message from the sim card and is ready to receive the next command.

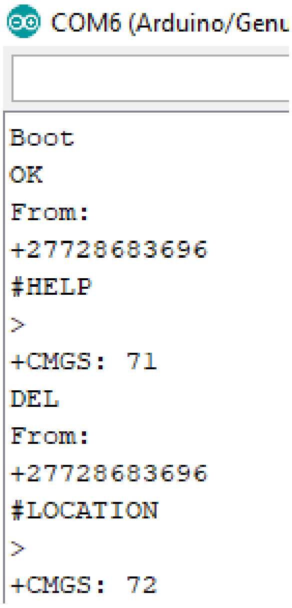


Figure 7 Example of the serial output (Wykerd, 2018)

6. Usage of The Device

In this section, an explanation is given on how the user will retrieve the location of the device.

The user will send a command to the device via SMS, there are three possible commands the user can send to the device. The command "#Location" (not case sensitive) can be sent to the number of the sim card inside the device (the device will require a sim card loaded with airtime to send the response back) to retrieve the device's location. The format of the SMS can be seen in Figure 8 below. The first line of the SMS includes the coordinates, the second includes a link to Google Maps that shows the location of the device on Google Maps. The last line shows the number of satellites that the GPS module has acquired, this is for debugging and to show the quality of the GPS fix. If there is no GPS fix an SMS saying "Acquiring satellites" will be sent.

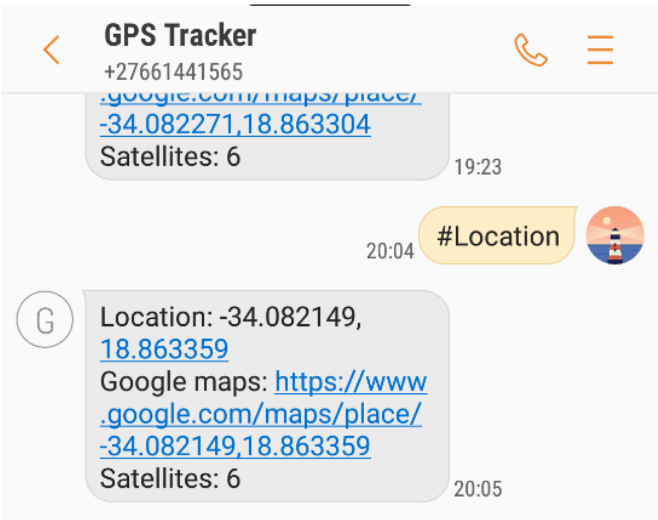


Figure 8 Example of #Location SMS (Wykerd, 2018)

The command "#Help" can be sent to see a list of the possible commands that the device can react to. Lastly, the command "#Off" can be sent to turn off the GPS module to save battery life. The GPS module will automatically turn back on if the user requests the location by sending "#location" again.

7. System Diagram

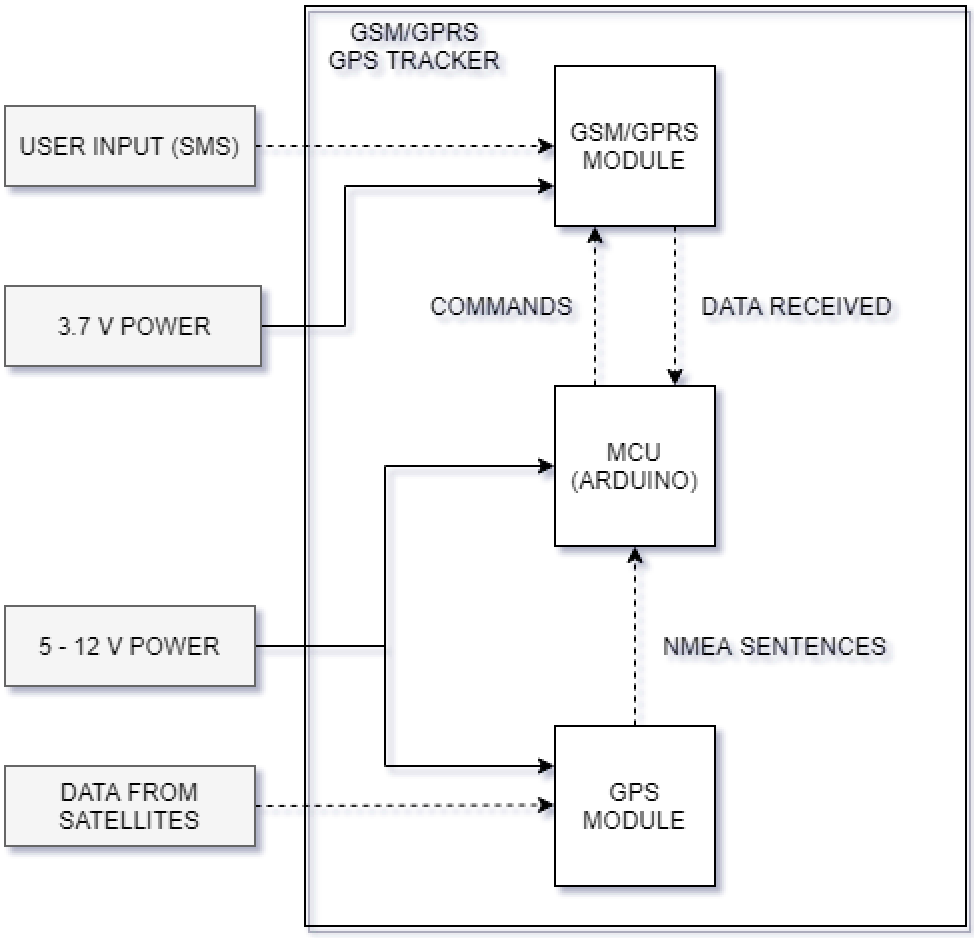


Figure 9 System Diagram of first prototype (Wykerd, 2018).

8. Analysis of prototype:

The prototype was analyzed using the design requirements and criteria outlined in the research plan to determine how well the project succeeds in the engineering goal on page 3.

Requirements

Since the goal of the first prototype was to test the concept of creating a GPS Tracker and not to be reproduced, some of the requirements were not met.

The device is easily manufacturable, but only in small scale, it would be uneconomical to produce this prototype on a mass scale and the electronics of the device is not permanently attached and can easily be disassembled. This easy disassembly was necessary so that debugging can be done fast, and components can be rewired and/or replaced in a matter of seconds.

The device does not get its power from one power source, this issue is addressed in the next prototype by removing the Arduino entirely and opting for a standalone Atmel MCU.

Criteria

Table 2 Analysis of first prototype to design criteria:

Criterion	How well does the device meet the criterion?	Suggested improvements that can be made:
The device should draw a low current to prolong the battery life.	The device is powered via the USB of the computer. It does not use a battery as its primary energy source. There are measures in place to preserve power, this includes the “#Off” command that will turns of the GPS module.	This issue can be eliminated by getting rid of the Arduino and using a standalone MCU instead.
The device should have a small form factor	The device has a footprint of approx. 12cm*9.5cm*4cm. It meets the criterion very badly.	The device’s size can be cut in half by replacing the Arduino with a standalone MCU.
The device should be user-friendly.	The device’s location can be accessed via SMS. The full list of commands can be found under ‘usage of the device’	At this moment no improvements can be made to make the device more user-friendly on the software

	on page 10. This is a very user-friendly way of retrieving the coordinates. The hardware, however, is not at all user-friendly. The goal of this prototype was not to make a product for the end user, but rather to test the components and software	side. The device can be physically made more user-friendly by making the device smaller and self-contained (no computer required).
The environmental impact should be kept low	All the components used in this prototype is RoHS compliant and is thus free of certain hazardous substances such as lead.	Fewer components can be used to decrease steps in production that create hazardous by-products (like soldering)
Cost should be as low as possible	The prototype cost R947.89 to make.	The cost can be minimized by creating a custom PCB with only the required components (no Arduino or breakout boards only the components that are essential for the device to function) on it.
Repair and upgradability	The device can easily be repaired and upgraded as it is built on a breadboard.	No improvements can be thought of at this moment

Second Prototype

1. Introduction

The goal of the second prototype is to get rid of the Arduino and replace it with a standalone MCU, the ATMega328p-pu (the same MCU used in the Arduino UNO). The reason for removing the Arduino is to save space and reduce the number of off the shelf components (that cost expensive) so that it can be easily manufactured. The MCU aspect of this prototype will be used in making the final prototype.

This prototype will not make any changes to the software and thus there is no software part of this prototype, to see how the software works refer to page 10. The usage will also remain the same and can be found on page 11.

2. Materials / Components Chosen

This prototype repurposes the components used in the previous prototype. The same GSM and GPS module is used. The Arduino UNO's microcontroller was removed and reprogramed using a second Arduino UNO as an ISP (in-system programmer).

3. Cost

This prototype would have cost R

Component	Supplier	Cost per unit
Atmega328p-pu	Mouser	R 32.34
SIM800L Breakout	Communica	R 236.84
FGPMMOPA6H Breakout	Communica	R 635.95
	Total:	R 905.13

4. Replacing the Arduino with a Standalone MCU

Getting rid of the Arduino was quite straightforward. Arduino has a great tutorial on their website on how to use an Arduino as an ISP to burn a bootloader onto a new MCU. Although the MCU used in this prototype already had the Arduino bootloader burned to the flash, it required an external 16MHz oscillator (the ATMega328p) which has an 8MHz internal oscillator that can be used instead of an external oscillator by changing some of the MCU's fuse bits. This way 3 more components could be removed from the board (the crystal and 2 18pF ceramic capacitors). The board profile that was used to change the bootloader can be found on Arduino's website (find the link in the references). See the figure below on how the Arduino was used as ISP to burn a new bootloader to the MCU.

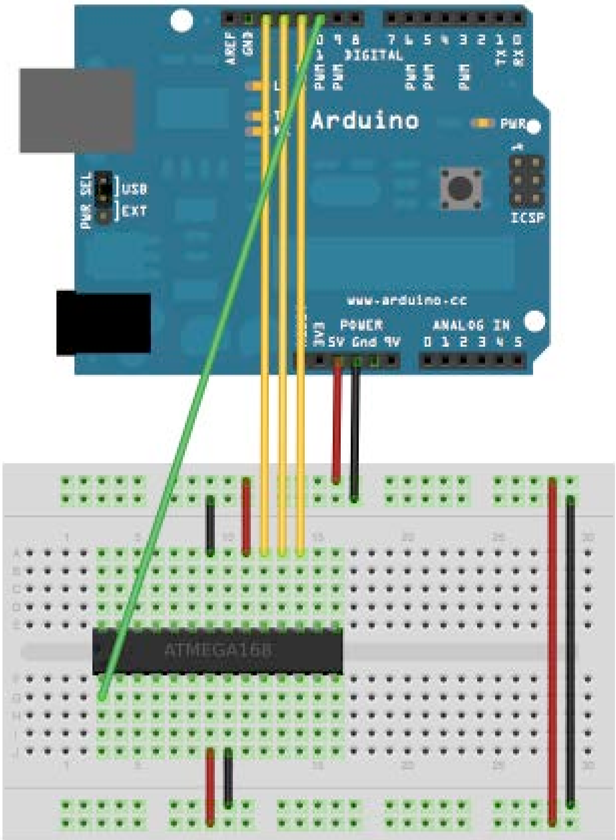


Figure 10 Arduino as ISP (Arduino, n.d.)

After the bootloader was burned onto the MCU, programs could easily be uploaded to the MCU using an USB-to-TTL FTDI chip that is connected to the serial port of the MCU. To save cost the Arduino board's FTDI chip was used to program the ATMega instead of purchasing a new USB-to-TTL board.

The serial connections (RX and TX of the modules) between the MCU and the GSM and GPS modules also had to be rewired as the Arduino has been removed. The bootloader on the MCU works in the exact same way as the bootloader on the Arduino's MCU and rewiring is a simple case of figuring out where which input/output pin connects to the MCU's pins. This could have been done manually by using the continuity tester on a multi-meter and checking to which output the MCU's pins connected or alternatively by using the following diagram seen below of the ATMega168 and ATMega328p's Arduino pinout that has been pre-tested. This was the route followed in this project.

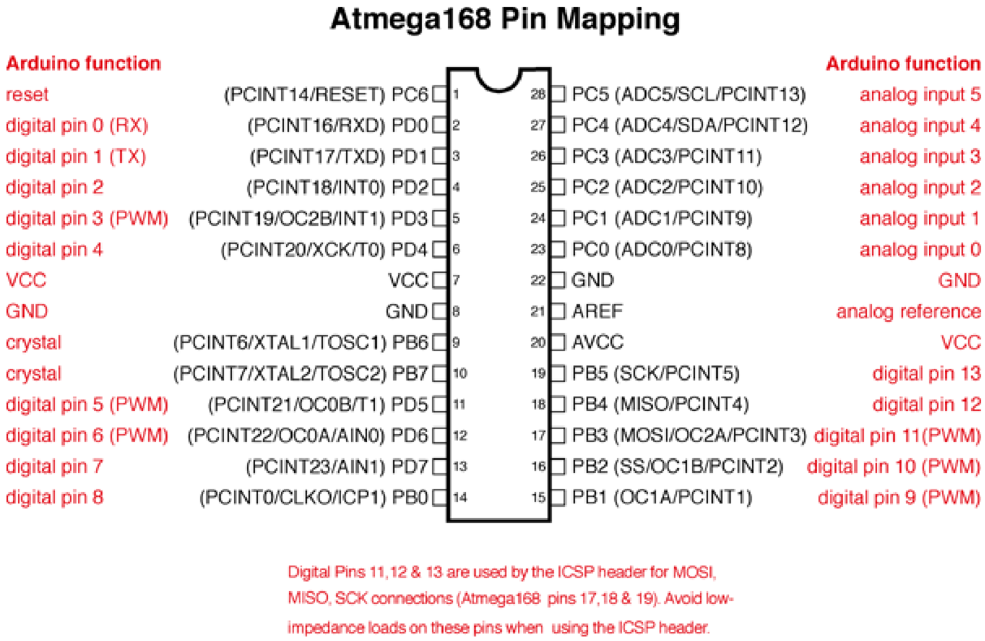


Figure 11 Atmega328p and 168 Arduino Pin Mapping (Arduino, n.d.)

5. Schematic and Breadboard Layout

The following two figures show the design of the second prototype and was created using Fritzing, a free open-source software that allows you to digitally make your own schematic and breadboard layout.

During the design phase, a feature was omitted (the GPS off feature) and was only noticed after the completion of the final prototype and can thus not be changed. This can be reimplemented in a later prototype.

The design does not include the VCC and GND input, but this input can be seen in the figures in the assembly section of the second prototype.

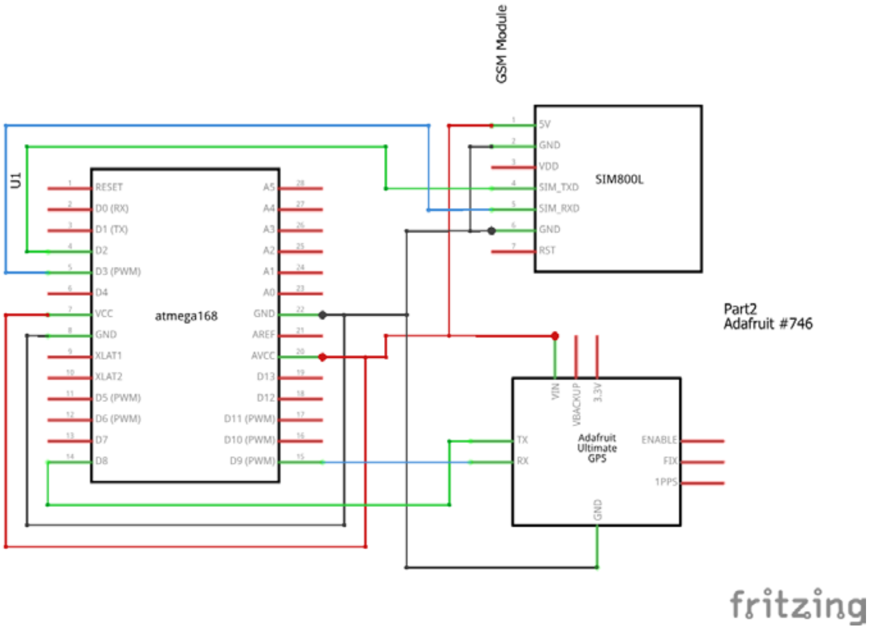


Figure 12 Schematic of second prototype (Wykerd, 2018)

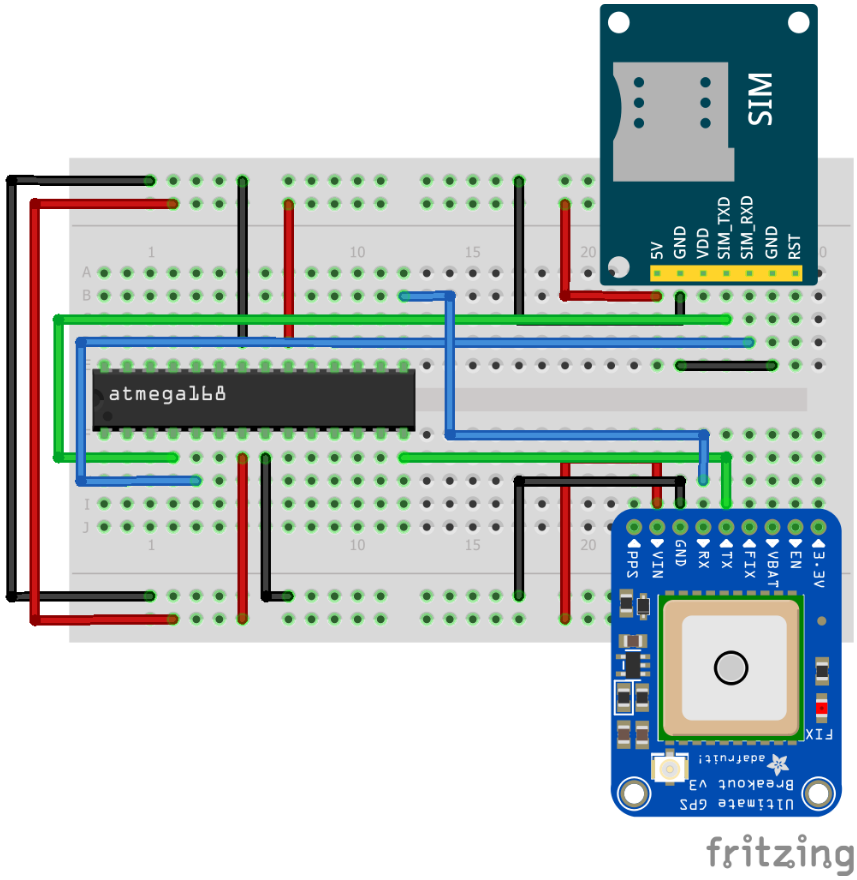


Figure 13 Breadboard layout of second prototype (Wykerd, 2018)

6. Assembly

Assembly was like the first prototype's assembly procedure. All the assembly procedures were done by Daniel Wykerd, including burning of the bootloader to the MCU, programming of the MCU and wiring of the components on the breadboard. The figures below show the results.

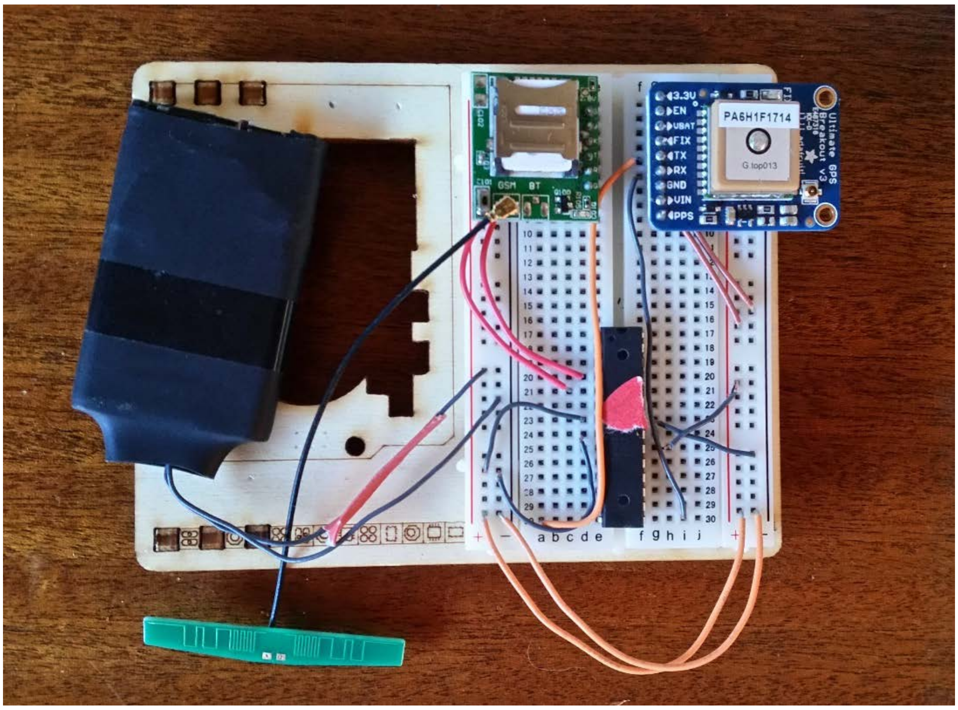


Figure 14 Final assembly of second prototype (Wykerd, 2018)

Power is supplied in parallel to all components. The power source is a 3.7 V Lithium Ion Polymer (Li-Po) battery salvaged from an old cell phone.

7. System diagram

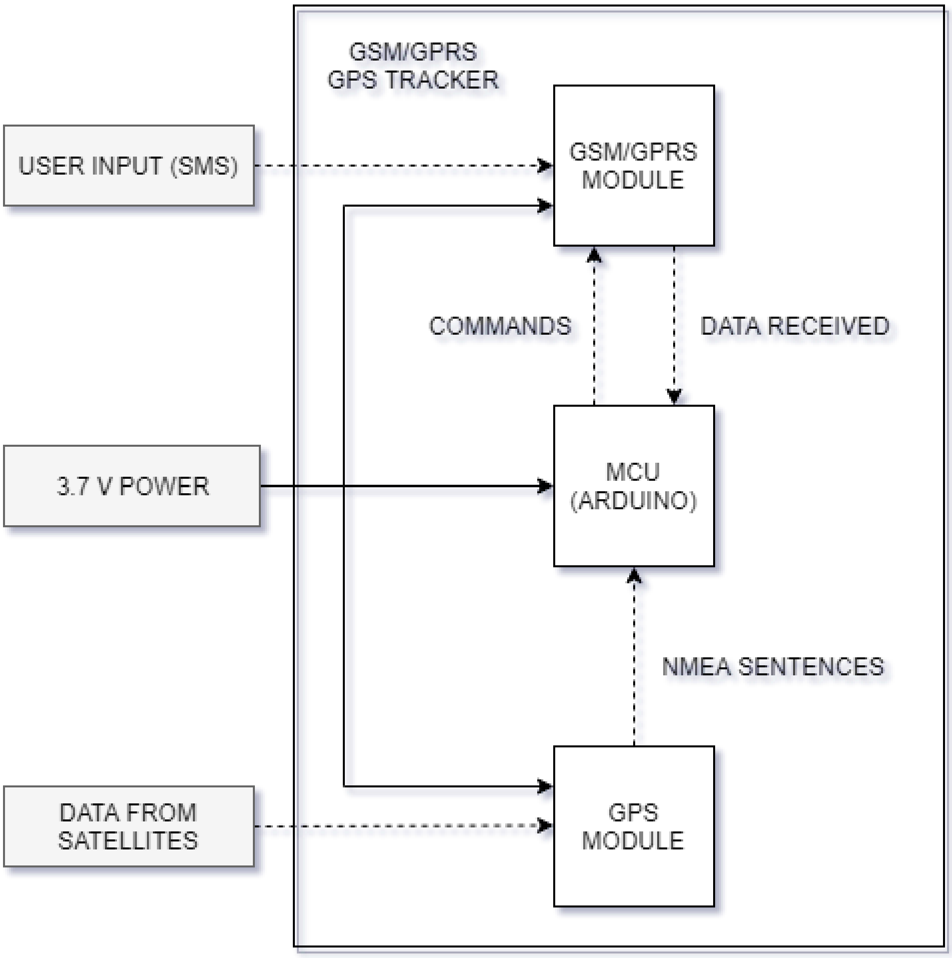


Figure 15 System diagram of second prototype (Wykerd, 2018)

8. Analysis of prototype

The prototype was analyzed using the design requirements and criteria outlined in the research plan to determine how well the project succeeds in the engineering goal on page 3.

Requirements

This prototype met more of the requirements than the first but still has plenty of shortcomings since it is still on a built breadboard and was not meant to be used by an end-user. The device is still not easily mass producible and is still not user-friendly in terms of hardware. The device however now runs from a single battery unlike the first prototype and has fewer parts lowering the cost.

Criteria

Table 3 Analysis of second prototype to design criteria

Criterion	How well does the device meet the criterion?	Suggested improvements that can be made:
The device should draw a low current to prolong the battery life.	The device draws a low current. The average amp draw is around 50 - 60 mA max.	There are not many improvements that can be made except that the '#off' command can be reimplemented.
The device should have a small form factor	The device has a footprint of approx. 5.5cm*8.5cm*2cm. It meets the criterion poorly but better than the first prototype.	The device's size can be diminished several times by creating a custom PCB with only the bare minimum components
The device should be user-friendly.	The device's location can be accessed via SMS. The full list of commands can be found under 'usage of the device' on page 10. This is a very user-friendly way of retrieving the coordinates. The hardware, however, is not at all user-friendly, but the goal of this prototype was not to make a product for the end user but to remove the Arduino in preparation for designing the final PCB	At this moment no improvements can be made to make the device more user-friendly on the software side. The device can be physically made more user-friendly by creating a custom PCB with only the bare minimum components
The environmental impact should be kept low	All the components used in this prototype is RoHS compliant and is thus free of certain hazardous substances such as lead.	Fewer components can be used to decrease steps in production that create hazardous by-products (like soldering)
Cost should be as low as possible	The prototype cost R905.13 to make.	The cost can be minimized by creating a custom PCB with only the required components (no Arduino or breakout boards only the components that are essential for the device to function) on it.
Repair and upgradability	The device can easily be repaired and upgraded as it is built on a breadboard.	No improvements can be thought of at this moment

The Third Prototype

1. Introduction

This prototype will use all the designs made and results obtained in the previous two prototypes to create a fully functional, self-contained Modular GSM/GPRS GPS Tracker.

The device will be contained on a single multi-layered PCB designed from scratch using the chips' datasheets as a reference during the components choosing and design stage. The reason for creating a custom PCB is so that the device can be easily manufactured as PCB fabrication is a very fast process.

There will be no off the shelf components used in the build resulting in a steep drop in the price per unit (especially if the device is made in large quantities where the components are brought in bulk). All the components used will be SMT (Surface Mount Technology) components, because these components are much smaller than their through-hole counterparts and these types of components are the most common type components used in factories. All the items can be purchased straight from the component manufactures and assembled by pick and place machines (a piece of equipment used in factories to automatically apply solder paste and thereafter the SMT components on the unpopulated boards) in factories, minimizing the need for human involvement in production (which will further decrease production cost). It will be possible for the entire device to be assembled in one factory (except for the bare components which will be sourced from suppliers).

A case will also be designed to house the device. The case will offer means of mounting the device to other modules.

2. Choosing the Main Components:

There was decided on using similar components as the components used in the previous two prototypes as these components were already tested to work with the software and hardware configuration.

The MCU chosen is the Atmega328P-AU, the same IC as the Atmega328P-PU but in a different package, TQFP (Quad flat package) instead of DIP package (Dual in-line package). The reason for choosing this component is because it has enough flash memory for the software (as the software is quite large due to the use of libraries) and is the same chip as the one used in the previous two prototypes.

The GSM/GPRS Module chosen is the SIM800C module, this chip is like the SIM800L module used in the previous prototypes but has an easier to solder package, it has its contacts around the chip instead of under the chip in a land grid array (LGA) package. This makes it easier to solder the component. Another advantage of the SIM800C is its size is 17.6*15.7*2.3mm according to its datasheet (SIMCom Wireless Solutions, 2017) in comparison to 15.8*17.8*2.4 mm (SIMCom, 2015). It does not have any unnecessary features like Bluetooth like the SIM800H has. It is thus a better module to use in this case.

The FGPMMOPA6H has been chosen as it has already been used in the other two projects. It served its purpose nicely and is thus used again in this prototype.

3. Breakdown of Schematic

The schematic for this prototype was created in EasyEDA a free online CAD software for schematic and PCB layout creation.

The schematic is broken into three parts: The GPS Module, MCU and GSM Module.

The full schematic can be found in Appendix C.

All resistors, MLCCs (Multi-Layer Ceramic Capacitors) and Tantalum capacitors are 1206 packages.

The GSM Module

The GSM module was designed first using the SIMCOM800C Datasheet listed in the references and added in the journal. Below is an image of this section.

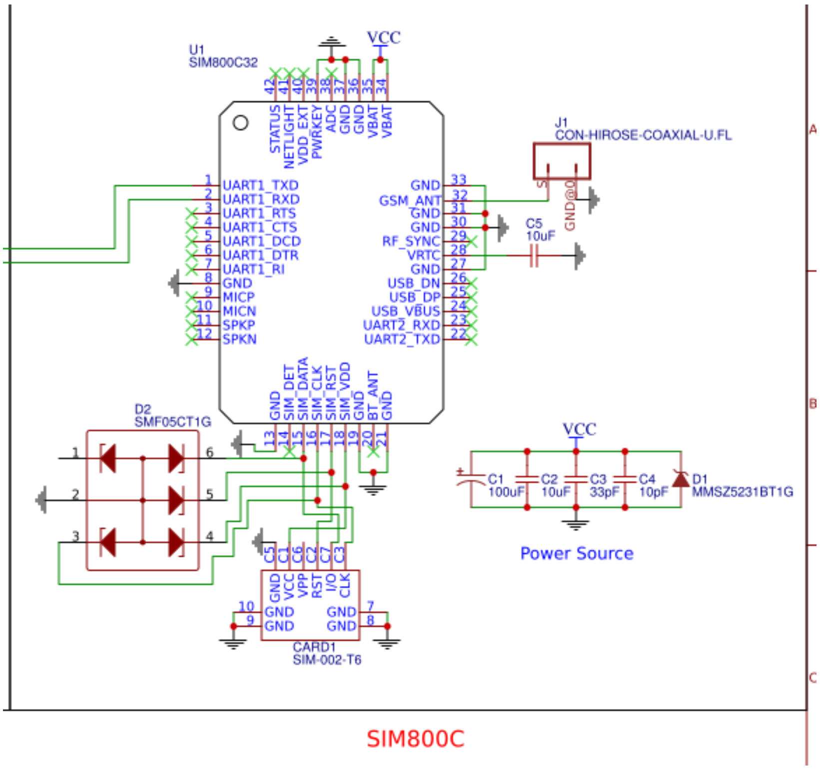


Figure 16 GSM Section of Schematic

The power source was designed using page 17 of the datasheet as reference.

SIM-002-T6 was chosen as the sim card reader (CARD1) as it has a push to insert and push to eject mechanism. This mechanism works the best with the case that was designed to house the PCB as the SIM can easily be ejected.

The purpose of the diode bridge SMF05CT1G (D2) is to prevent electrostatic discharge (ESD) whilst the SIM is inserted. It can be omitted but is there to protect the GSM module.

The GSM antenna will connect to the GSM module via the coaxial jack (J1).

The GPS Module

The GPS module's datasheet can be found in this project journal and in the references. Here is a breakdown of the components used:

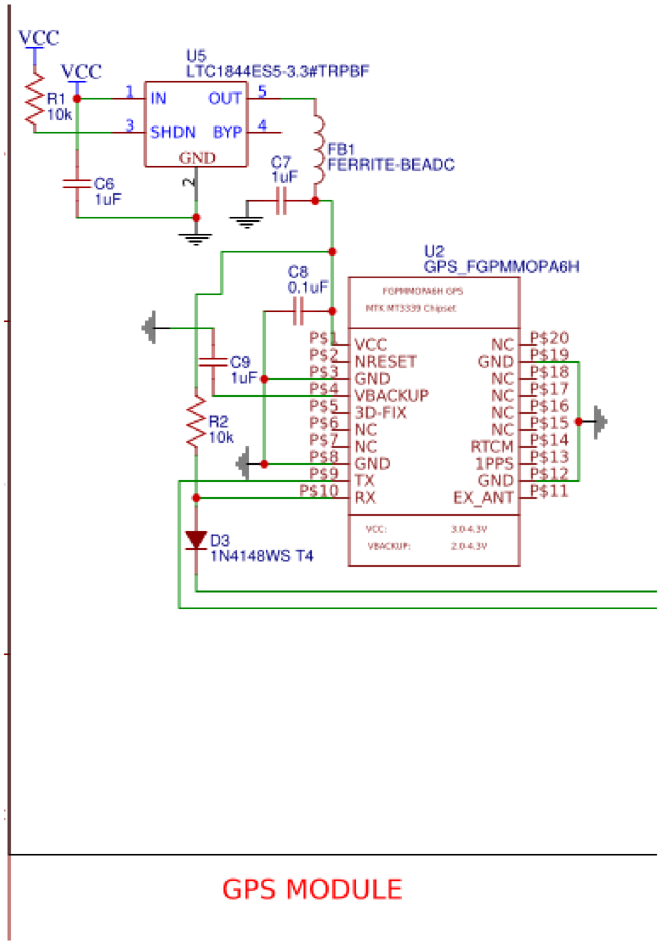


Figure 17 GPS Section of Schematic

U5: The GPS module has a typical voltage input of 3.3 V. The datasheet recommends a voltage regulator to keep the voltage supplied to the module constant. Microchip's MIC2552-3.3 Low Dropout Regulator (LDO) was chosen to provide power to the module – the schematic shows another LDO, but this is because the MIC2552 was not in EasyEDA's library of components.

R1: Keeps the LDO powered on.

FB1: The ferrite bead is used to minimize EMI (Electromagnetic Interference). This is called an RF Choke. It does not let high frequencies through. It has an impendence of 600Ohm at 100MHz according to its datasheet (Murata, n.d.). This is the same as the specs of the ferrite bead recommended by the GPS module's datasheet.

R2 and D2 restrict the Rx input to 3.3V according to Adafruit's reference (Adafruit Industries, 2012).

The MCU

The MCU is the same as the one used in the previous prototypes. Wiring is thus similar. Below is the schematic:

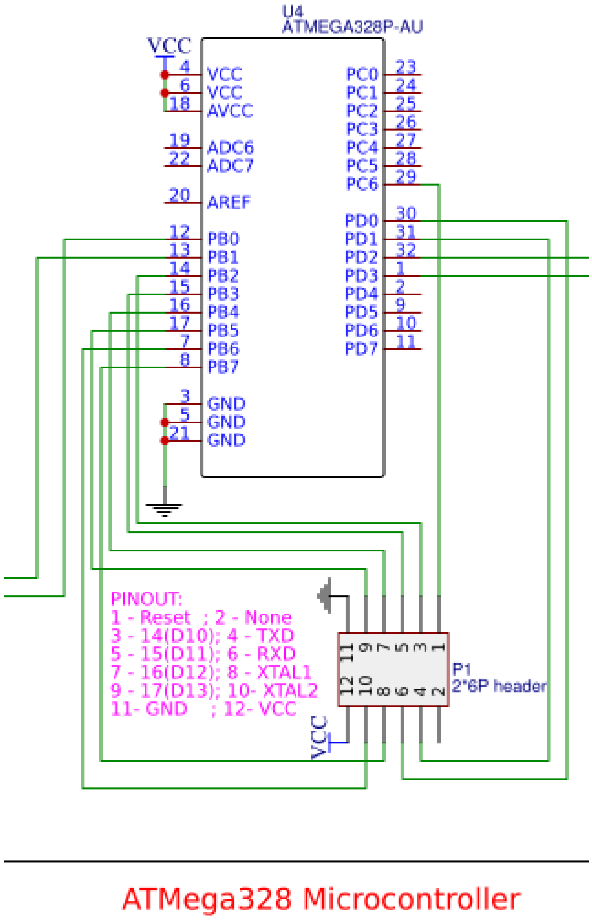


Figure 18 MCU Section of schematic

P1: Male header used for burning bootloader to MCU and programming the MCU. Also used for debugging via serial. The header can also be used by other modules to add functionality to the device as there is 3 extra digital I/O pins available for use by an external device (software update required for modules to take advantage of the MCU's extra pins).

4. Bill of Components (BOM)

Table 4 BOM of third prototype

value	quantity	package	componen	Supplier	Manufactur	Manufacturer Part
SIM800C32	1	SIM800C	U1	LCSC	SIMCOM	SIM800C32
GPS_FGPMMOPA6H	1	FGPMMOPA6H	U2	Adafruit		FGPMMOPA6H
ATMEGA328P-AU	1	TQFP-32_7X7X08P	U4	MOUSER	MICROCHIP	ATMEGA328P-AU
10uF	2	1206	C2,C5	LCSC		?
10pF	1	1206	C4	MOUSER		?
33pF	1	1206	C3	LCSC		?
MMSZ5231BT1G	1	SOD-123	D1	MOUSER	ON	MMSZ5231BT1G
SMF05CT1G	1	SOT-363	D2	LCSC	ON	SMF05CT1G
CON-HIROSE-COAXIAL-U.FL	1	CON-U.FL	J1	MOUSER		?
MIC2552-3.3	1	TSOT-23-5	U5	LCSC	MICROCHIP	MIC2552-3.3
10k	2	1206	R1,R2	LCSC		?
1uF	3	1206	C6,C7,C9	LCSC		?
0.1uF	1	1206	C8	MOUSER		?
1N4148WS T4	1	SOD-323	D3	LCSC	CJ	1N4148WS T4
2*6P header	1	HDR-6X2H/2.54	P1	LCSC	BOOMELE	2*6P header
100uF	1	CASE-A_3216	C1	LCSC		?
FERRITE-BEADC	1	1206	FB1	MOUSER	Murata	?

The total cost of the project can be viewed in Appendix A.

5. PCB Layout

The PCB layout was designed in EasyEDA a free web-based CAD software for creating schematics and PCBs.

The first attempt at designing a PCB was unsuccessful as traces were not matching the right places. This failed attempt is in the journal.

Below is a figure of top PCB layer:

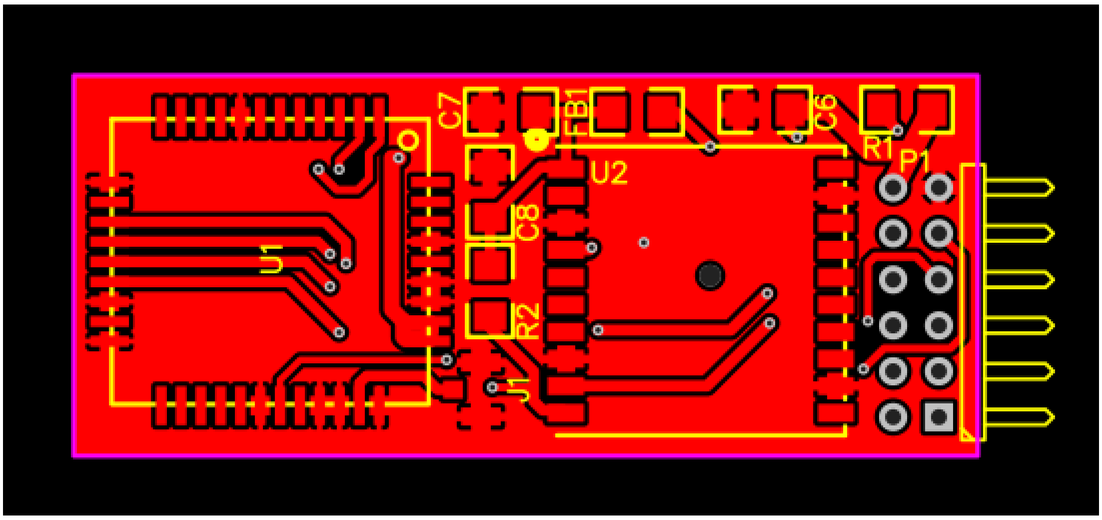


Figure 19 Top Layer of PCB

The GSM module, jack and GPS module will be mounted on the top layer.

Below is a figure of the bottom layer:

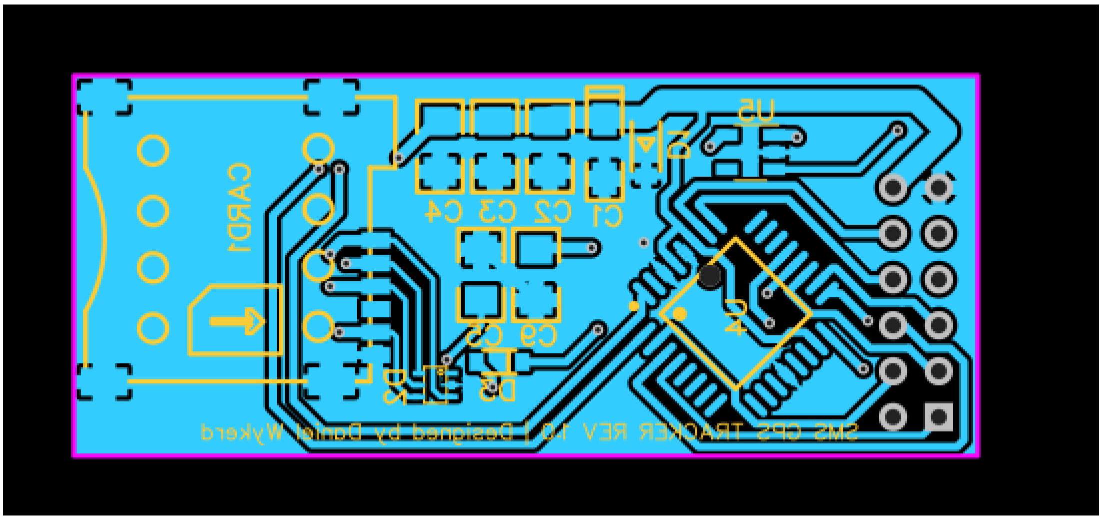


Figure 20 Bottom Layer PCB Layout

The MCU, Card reader, and most VCC lines will be mounted to the bottom layer.

Below is a figure of the board with all layers:

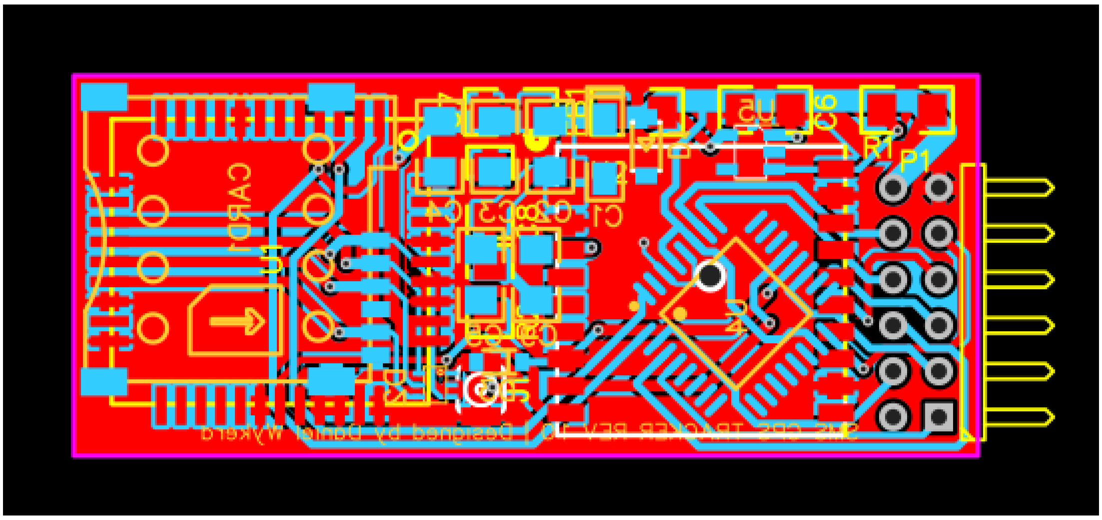


Figure 21 PCB Layout

6. System Diagram

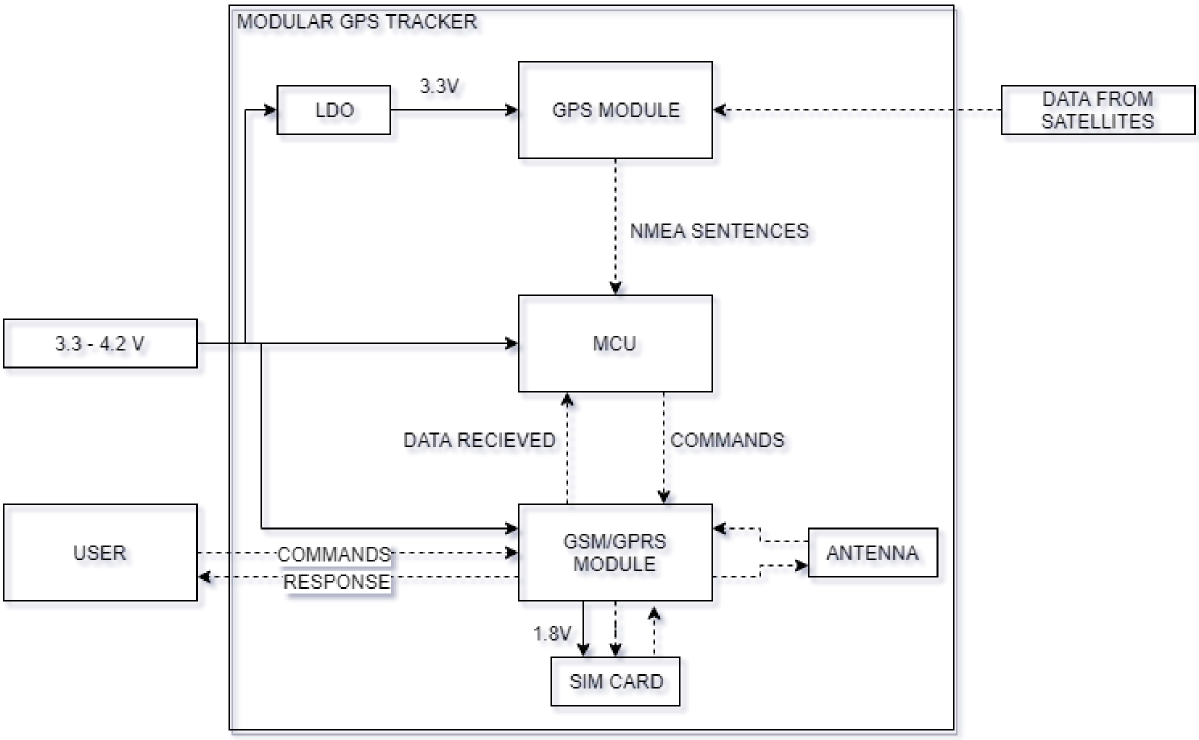


Figure 22 System Diagram of final prototype

7. Case design

A case was designed for the PCB to fit in. The case was designed in 123D Design, a free CAD software.

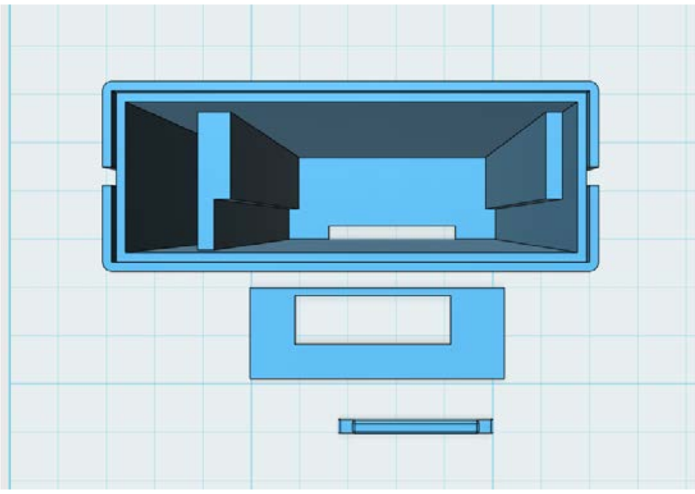


Figure 23 Top view of case

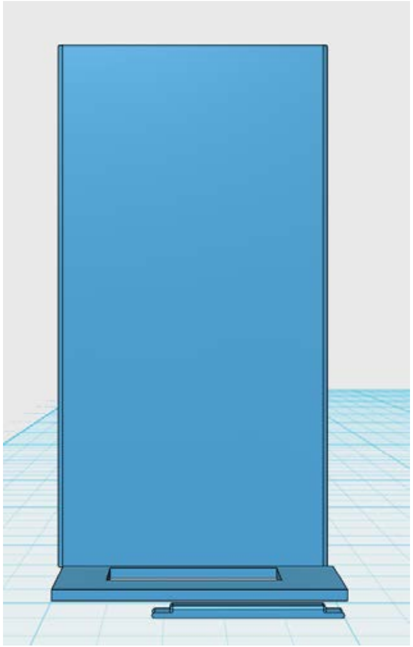


Figure 25 Front view of case

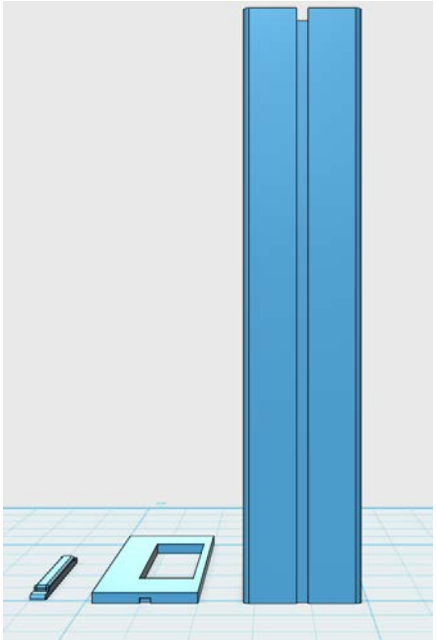


Figure 24 Right view of case

The case has grooves on the sides for other modules to slide onto. The Second rectangle with the hole in goes on top of the case after the PCB and antenna is inserted into the case. The small rectangle in the lower end of the top view is a cap for the SIM slot. It is pushed into the sim hole after the sim card is installed in the slot to prevent dust and other materials from damaging the PCB. The case is not water resistant!

Here are the dimensions of the top view

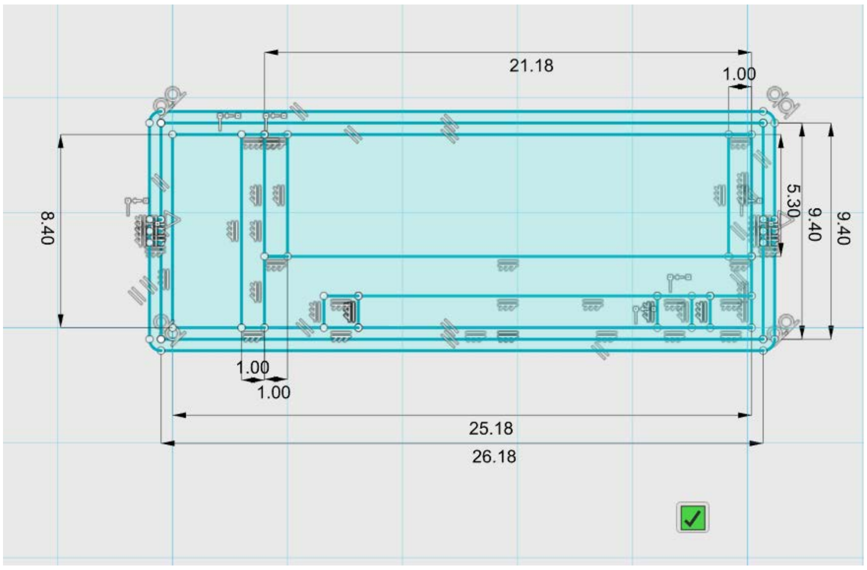


Figure 26 Dimensions of Top view of case

The case is approx. (the headers add height) 22 mm high after the PCB is inserted.

8. Assembly

Assembly required special equipment and knowledge of SMT soldering. Thus, Johann Erasmus was asked to assist with soldering during the first attempt of assembling the prototype. He soldered the board under a microscope using the schematic as a reference. The GPS module was left out to test the MCU and GSM module first. It turned out that the MCU got bricked or fried while attempting to burn the bootloader to the fuses of the MCU. Johann Erasmus and Danie Ludick (PhD) assisted with debugging but no solution was found. Below is a figure of the bricked board:

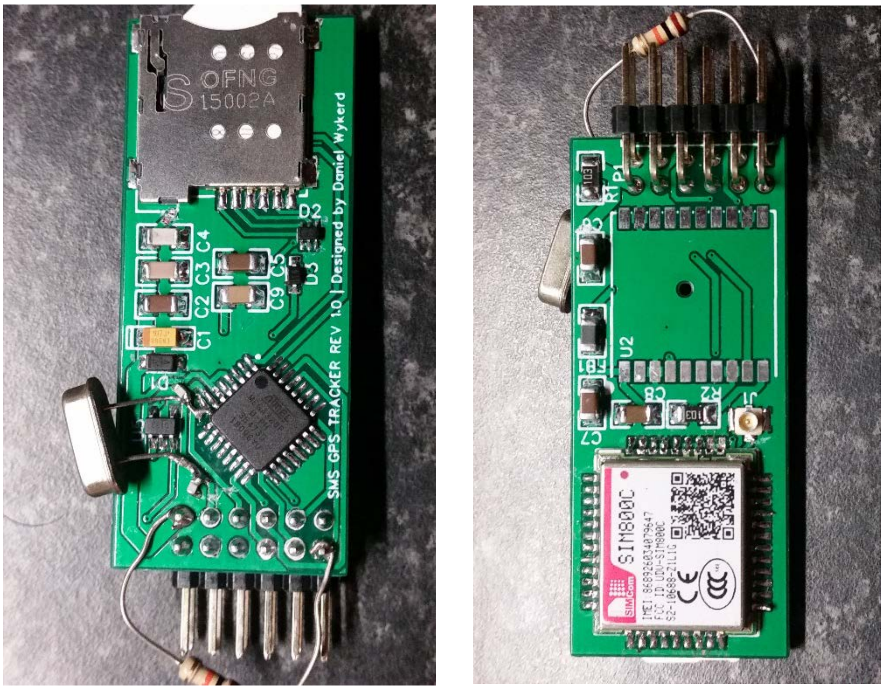


Figure 27 Faulty board

After consulting with Dr. Ludick it was decided to take the PCB and parts to the University of Stellenbosch for them to populate/solder a new board. This time a 10K Ohm pull up resistor was added between VCC and reset to prevent the board from resetting. After this component was added, the bootloader did burn to the MCU. Here is a picture of the final board:

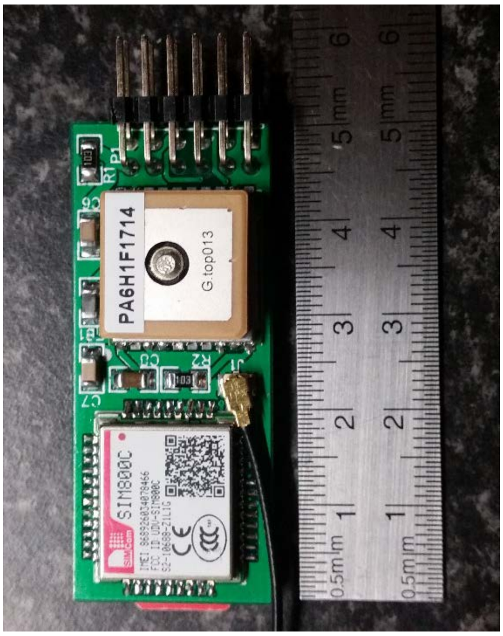


Figure 29 Top of Assembled PCB

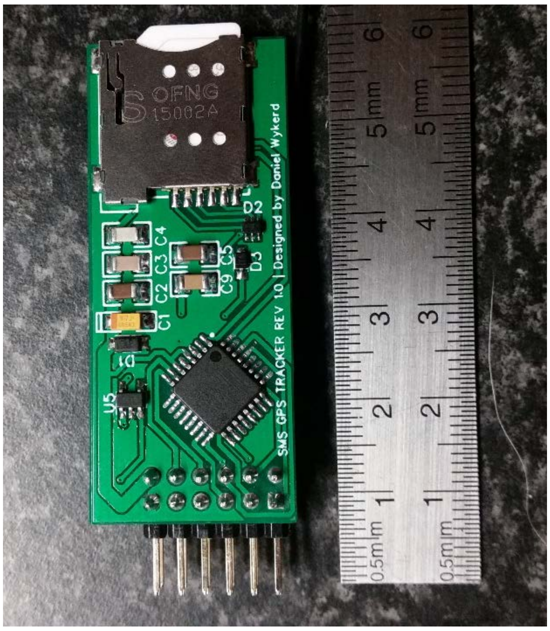


Figure 28 Bottom of Assembled PCB

A makeshift programming station was built by Daniel Wykerd to program and debug the board via serial. Below is an image of the programmer:

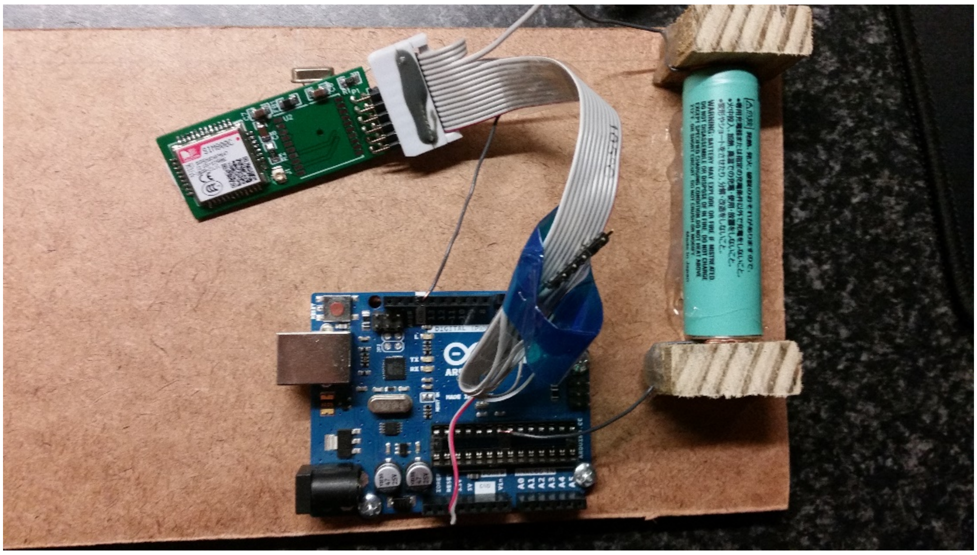


Figure 30 Programming Station

9. Software

The software is the same as the software of the first two prototypes except for losing the ‘#Off’ command. The final software is available in Appendix D.

10. Analysis of results

The prototype was analyzed using the design requirements and criteria outlined in the research plan to determine how well the project succeeds in the engineering goal on page 3.

Requirements

The device meets all the requirements set out in the research plan.

The device is easily manufacturable because the device is contained on a single multi-layered PCB thus the device can be easily manufactured as PCB fabrication is a very fast process.

There is no off the shelf components used in the build resulting in a steep drop in the price per unit (especially if the device is made in large quantities where the components are brought in bulk). All the components are SMT components. These types of components are the most common type components used in factories, it can easily be assembled by pick and place machines minimizing the need for human involvement in production (which further decreases production cost). It is possible for the entire device to be assembled in one factory (except for the bare components which need to be sourced from suppliers).

The device uses pre-existing communication infrastructure, it uses SMS to communicate with the user. Thus, it does not require a subscription to a third party. Airtime can be loaded onto the SIM card and the device will work.

The device is modular. The code can be changed using the serial interface of the MCU and there are 3 digital I/O pins available for use by other modules added later. The case can easily be slid into other modules using the grooves on the sides.

The device has a battery life of over 48 hours on a 2350 mAh according to tests done. The raw data of the test can be found in the journal of the project.

The cost per unit is far below R1000 (R293.47 to be precise).

Criteria

Criterion	How well does the device meet the criterion?	Suggested improvements that can be made:
The device should draw a low current to prolong the battery life.	The device draws a low current. The average amp draw is around 50 mA max. It lasts longer than 48 on a charge of a 2350 mAh battery	There are not many improvements that can be made except that the '#off' command can be reimplemented.
The device should have a small form factor	The device has a footprint of 50.1 mm * 21.1 mm * 8.4 mm. It meets the criterion extremely well.	The GSM antenna can be put on the same PCB
The device should be user-friendly.	The device's location can be accessed via SMS. The full list of commands can be found under 'usage of the device' on page 10. This is a very user-friendly way of retrieving the coordinates. The hardware is very user-friendly especially if it is in a case. The device is sturdy, and it is very easy to add modules	At this moment no improvements can be made to make the device more user-friendly on the software side. The device can be physically made more user-friendly by creating more modules for the device.
The environmental impact should be kept low	The prototype uses lead solder which is a hazardous substance.	The board can be made RoHS compliant by using lead-free solder.
Cost should be as low as possible	The module cost only R293.47 to make. This is a huge improvement above the previous prototypes	The cost can be reduced by buying items in bulk
Repair and upgradability	The device uses SMT components which is not easily replaceable. The case is made of plastic and can be manufactured for cheap and replaced easily Software, however, can easily be upgraded to make the device work with different modules or fix bugs (if any are found)	It can be made easier to replace broken components

Discussion

A lot of results has been obtained in this project. The discussion will point out patterns in the results and explain it. The device's limitations will also be discussed.

Each revision (prototype) of the device met the design requirements and criteria better than the its predecessor following reasons:

With each prototype, a significant decrease in the size of the device occurred. This is because as later prototypes were developed, the components used for testing the device became unnecessary and thus more of the components were omitted with each design revision leading to a smaller form factor.

The current draw of the device also decreased with each prototype as the unnecessary power-hungry components were removed.

The device has several limitations. The first is that the device can only get a GPS fix if it is outside. If the device is switched on inside a building it will not get a GPS fix unless it is placed near a window or somewhere with a clear line of sight with the GPS satellites. The device will function indoors if an initial GPS fix was gained outside.

The other limitation is the battery life that can be improved in later software updates by reimplementing the GPS off feature to preserve power. Another possible solution that could be researched is to add an off switch to the battery module.

Further research that can be done and improvements that can be made in the future includes designing some supporting modules like battery modules or modules that allow attachment of the device to objects. Another version of the software could be created that constantly streams the device's location to a data server like a MySQL server, allowing the user to view a log of where the device was using the web. The device already supports uploading to the internet and has 2G network support thus this feature can easily be added in the future. The GSM antenna can be put onto the PCB making the assembly process easier.

Conclusion

The Modular GSM GPS Module designed and manufactured in this project meets the requirements and criteria set out in the research plan of this project exceptionally well.

Thanks to the fact that a custom PCB was created using industry standard production techniques the module is easily manufacturable, costs only R293.47 to make and does not require a subscription to a third party. The PCB is accompanied by a great case that make the device modular, meaning features can easily be added if other companies make modules for the device that is compatible with the PCB's pin out and case design. The PCB and case were designed in such a way that it is easy to create these modules. It shouldn't take a company to long to produce its own module.

The device draws a low current of around 50 mA and can last over for 48 hours on a 2350 mAh battery. According to all these results obtained, it is clear that the device created meets the engineering goal of this project.

Bibliography

Adafruit Industries, 2012. *Adafruit Ultimate GPS Arduino Wiring.* [Online]
Available at: https://learn.adafruit.com/adafruit-ultimate-gps/arduino-wiring
[Accessed 20 01 2018].

Adafruit Industries, 2012. *Learn Adafruit Ultimate GPS Downloads.* [Online]
Available at: https://learn.adafruit.com/adafruit-ultimate-gps/downloads
[Accessed 5 5 2018].

Arduino, n.d.. *ArduinoToBreadboard.* [Online]
Available at: https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard
[Accessed 13 04 2018].

Arduino, n.d.. *Reference GSM Library.* [Online]
Available at: https://www.arduino.cc/en/Reference/GSM
[Accessed 06 02 2018].

Arduino, n.d.. *Standalone Atmega.* [Online]
Available at: https://www.arduino.cc/en/uploads/Hacking/Atmega168PinMap2.png
[Accessed 13 4 2018].

Murata, n.d.. *Chip Ferrite Bead.* [Online]
Available at: https://www.mouser.co.za/datasheet/2/281/QNFA9105-1130343.pdf
[Accessed 8 5 2018].

SIMCom Wireless Solutions, 2017. *SIM800C Hardware Design.* [Online]
Available at: http://simcom.ee/documents/SIM800C/SIM800C_Hardware_Design_V1.05.pdf
[Accessed 8 5 2018].

SIMCom, 2015. *SIM800H&SIM800L Hardware Design.* [Online]
Available at:
http://simcom.ee/documents/SIM800H/SIM800H%26SIM800L_Hardware%20Design_V2.02.pdf
[Accessed 8 5 2018].

SIMCom, 2016. *SIM808 Hardware Design.* [Online]
Available at: http://simcom.ee/documents/SIM808/SIM808_Hardware%20Design_V1.03.pdf
[Accessed 5 5 2018].

South African Police Service, n.d.. *SAPS Crime Statistics.* [Online]
Available at: https://www.saps.gov.za/services/crimestats.php
[Accessed 28 April 2018].

Stats SA, 2017. *Statistical Release P0341.* [Online]
Available at: https://www.statssa.gov.za/publications/P0341/P03412016.pdf
[Accessed 28 April 2018].

TrackTheWorld, n.d.. *TRACKERS: MULTI-PURPOSE.* [Online]
Available at: http://www.trackingtheworld.com/products/multipurpose.php
[Accessed 28 April 2018].

Wikipedia, 2018. *Global Positioning System.* [Online]
Available at: https://en.wikipedia.org/wiki/Global_Positioning_System
[Accessed 06 May 2018].

Wikipedia, 2018. *Microcontroller.* [Online]
Available at: https://en.wikipedia.org/wiki/Microcontroller
[Accessed 6 May 2018].

Wykerd, D., 2018. *Assembly of first prototype.* [Art].

Wykerd, D., 2018. *Breadboard layout of first prototype.* [Art].

Wykerd, D., 2018. *Case designs.* [Art].

Wykerd, D., 2018. *Example of the serial output.* [Art].

Wykerd, D., 2018. *Format of #Location.* [Art].

Wykerd, D., 2018. *Image of first prototype components.* [Art].

Wykerd, D., 2018. *Photo of second prototype assembly.* [Art].

Wykerd, D., 2018. *Photos of final prototype.* [Art].

Wykerd, D., 2018. *Schematic and breadboard layout of second project.* [Art].

Wykerd, D., 2018. *Schematic of first prototype.* [Art].

Wykerd, D., 2018. *System diagrams.* [Art].

Wykerd, D., 2018. *Test Wiring of SIM800L.* [Art].

Acknowledgments

Thanks to Rochelle and Malan Wykerd for funding this project.

Thanks, Johann Erasmus for soldering the first attempt at assembling the final prototype and helping with the debugging of the final prototype's first attempt.

Thanks to Dr. Danie Ludick (PhD) for organizing with the University of Stellenbosch to populate the final prototype, reading through my designs and helping to debug the first attempt to assemble the first prototype.

Unpopulated PCBs fabricated by JLCPCB.

Appendix A

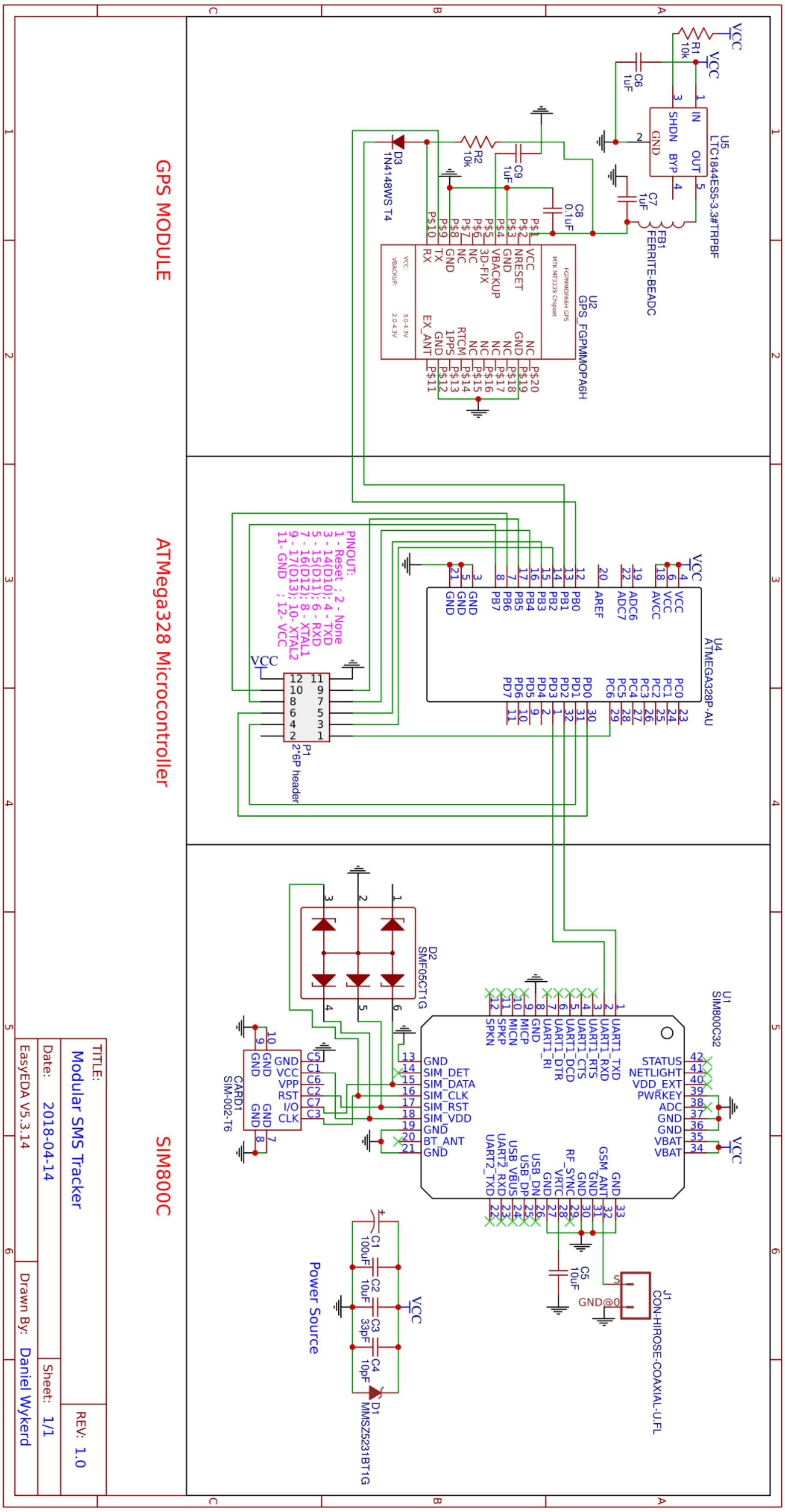
Table 5 Total cost of project

Component/Apperatus	Quantity	SUPPIER	PRICE PER UNIT	TOTAL
GPS MODULE BREAKOUT	1	COMMUNICA	635.96	635.96
GSM MODULE BREAKOUT	1	COMMUNICA	236.84	236.84
PCBS	10	JLCPCB	2.2	22
ATMEGA328P-AU	2	MOUSER	27.28	54.56
MIC2552-3.3YM5TR	2	MOUSER	5.29	10.58
U.FL-R	2	MOUSER	11	22
MLCC 10PF	5	MOUSER	4.51	22.55
MLCC 0.1UF	5	MOUSER	8.24	41.2
FERRITE BEAD	5	MOUSER	3.17	15.85
MLCC 33PF	100	LCSC	0.4	40
1N414WS	50	LCSC	0.16	8
SMF05CT1G	2	LCSC	2.01	4.02
TANTALUM 100UF	5	LCSC	2.93	14.65
HEADER	10	LCSC	0.58	5.8
MMSZ5231BT1G	20	LCSC	0.48	9.6
SIM800C32	2	LCSC	61.92	123.84
MLCC 10UF	10	LCSC	0.69	6.9
MLCC 1UF	20	LCSC	0.84	16.8
SIM-002-T6	2	LCSC	7.238	14.476
CHIP RESISTOR 10K OHM	50	LCSC	0.18	9
		SUB TOTAL	1011.918	1314.626
			SHIPPING COSTS	960
			TOTAL	2274.626

Appendix B

THE CODE WITH THE GPS OFF FEATURE IS ON THE NEXT TWO PAGES:

Appendix C


GPS MODULE
ATMega328 Microcontroller
SIM800C
U5
LTC1844ES5-3.3#TRPBF
FB1
FERRITE-BEADC
U2
GPS_FGPMMOPA6H
D3
1N4148WS T4
U4
ATMEGA328P-AU
PINOUT:
1 - Reset ; 2 - None
3 - 14(D10); 4 - TXD
5 - 15(D11); 6 - RXD
7 - 16(D12); 8 - XTAL1
9 - 17(D13); 10 - XTAL2
11 - GND ; 12 - VCC
P1
2*6P header
U1
SIM800C32
D2
SMF05CT1G
CARD1
SIM-002-T6
J1
CON-HIROSE-COAXIAL-U.FL
Power Source
D1
MMSZ5231BT1G
TITLE:
Modular SMS Tracker
REV: 1.0
Date: 2018-04-14
Sheet: 1/1
EasyEDA V5.3.14
Drawn By: Daniel Wykerd

Appendix D

THE CODE WITHOUT THE GPS OFF FEATURE IS ON THE NEXT TWO PAGES:

Appendix D