Design and development of a very low-power autonomous device and the related mobile app to locate Alzheimer suffering persons

F.N. LOGOTHETIS
Electrical and Computer Engineer
M. Sc. Candidate

Abstract

A common disease of the mind that occurs on many elder people is Alzheimer. It is the cause of 60% to 70% of cases of dementia. The most common early symptom is difficulty in remembering recent events (short-term memory loss). As the disease advances, symptoms can include problems with language, disorientation (including easily getting lost), mood swings, loss of motivation, not managing self-care, and behavioral issues.

The project's main objective was to provide a method for easily locating lost relatives, who suffering from this disease choose to wander outside of their home without informing their close ones. This has led us to the development of a GPS tracking device capable of informing the user about the patient's position using the GSM standard as a means of communication.

The project is also expected to function for a reasonable amount of time since it is a device for emergency situations. This also enables the device to stay on-line for grater time periods which increases the odds of full GPS and GSM coverage.

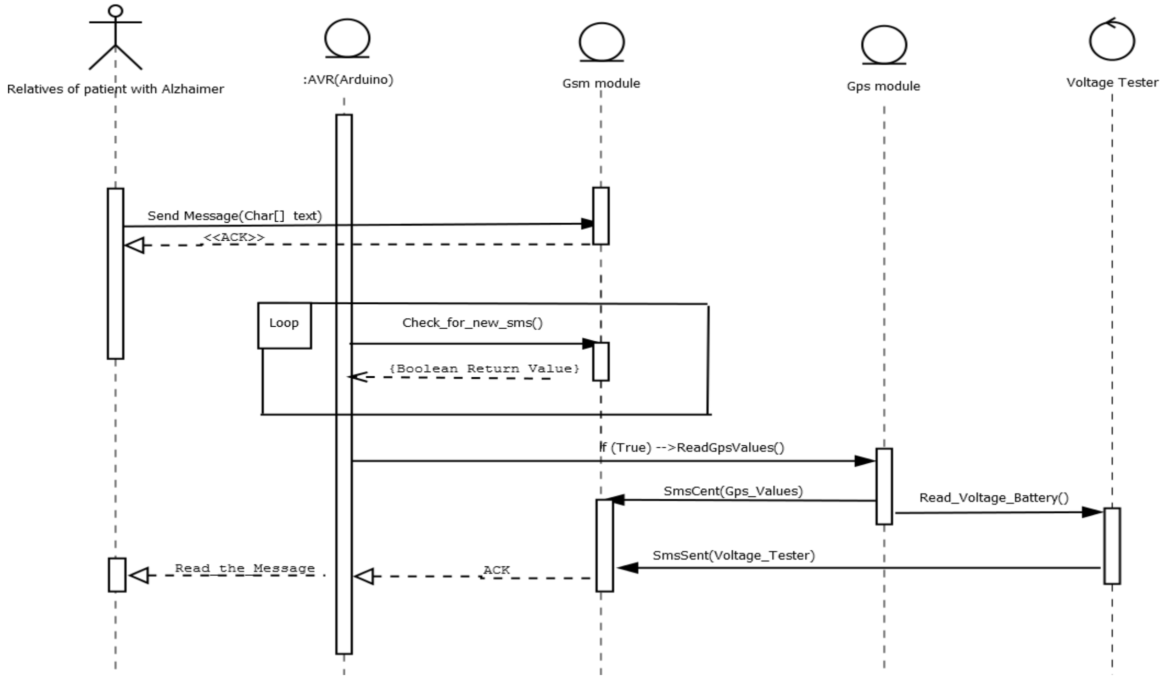
State of the art

GPS tracking is a really famous and quite traditional application. Individuals use to buy such devices to protect their personal belongings (car, motorbikes, etch). These GPS devices are connected on a rechargeable source of energy which is usually a high capacity battery. The recorded location data can either be stored within the tracking unit or transmitted to an Internet-connected device using the cellular (GPRS or SMS), radio, or satellite modem embedded in the unit. This allows location coordinates to be displayed against a map backdrop either in real time or when analyzing the track later, using GPS tracking software. Such software is available for smartphones with GPS capability.

The basic tracking devices are divided into two categories. The most common implementation is comprised by a GPS and a GSM module, which is connected on a server and keeps a database up-to-date each minute about the current "fingerprint" of the tracker. This design demands frequent recharging. The second approach comprises the same modules as the former, but it addresses the energy issue using periodic sampling of the location and utilizing also the GPRS module (mobile signal tracking) for the locating goal. The shortcoming of the aforementioned devices is that cannot be long-term used since they cannot be manually controlled by the user. Given that the GPS is a really power consuming module, in most of the cases the batteries will be run out of power whether is powered on without a specific reason.

The aim of our project is to develop a GPS tracker for people who are suffering from Alzheimer disease. In that case, our devise should be in active condition for a long period without recharging in order to be capable to locate the patient each moment. Hence, the tracking device must consume minimum energy when it samples the current location in order to guarantee the long-life scenario. There is no doubt that the modern hardware modules (GSM, GPS) are fully adjustable, therefore, there is neither software nor hardware limits to cope with the energy consumption issue. In the following sections, both the software and the hardware implementations are analyzed and detailed description about the new Alzheimer device is given.

Real-Life Scenario



Patients suffering from Alzheimer's disease usually decide to wander aimlessly in the streets. Having that in mind we decided to use a GPS module as our locating system. Cases where neither the GPS or GSM signals can be found are usually inside buildings and out of harm's way. The device is also equipped with two lithium-ion batteries (type 18650) providing enough power to function properly for roughly 9 days, ensuring that even if the patient is not found immediately, his relatives will have the option of GPS tracking for days to come, increasing significantly their chances. The tracking device will be on a belt, which will be worn around the patient's waist. This belt could be locked using a PIN number and it cannot be cut or destroyed by the patient. The user is able to communicate with the remote device using an app on his cellphone which makes the whole operation much easier. The android application is really user-friendly and it can locate the tracking device by a single click of a button. The corresponding message from the device contains not only the patients position in a form of URL for immediate redirection on google maps, but also the battery level helping the user decide whether they'd like to conserve battery by lowering the frequency of their communication. The reason why this conserves energy will be explained later on the hardware section.

The whole scenario is depicted in the above UML sequence diagram. After the patient gets lost, the relatives should start the android application in order to begin the locating process. When the GSM module of the tracking device receives the appropriate SMS from the cell phone, the AVR microprocessor starts the tracking procedure. Initially, the GPS module is switched on, until it receives accurately the longitude and latitude coordinates. Furthermore, the voltage of the li-ion batteries is also measured and the final message including all the appropriate information about the location and the battery voltage is sent back to the cell phone. When the relative of the patient received the "fingerprint" message, the google maps will be opened and the position of the tracking device will be illustrated with a pointer on the map.

It should be noted that the tracking device is sampling the location when it receives an appropriate message from the android app but it is able to return useful information about the last patient location when there is no GPS signal (i.e. into a home). Although the GPS signal coverage is available even in the most remote cities, there are indoor places where the GPS module cannot find valid coordinates. To avoid that scenario, our GPS tracker make a periodic sampling of the position as frequent as the battery level allow. The sampling is sparser when the battery voltage is less than 50%, saving the life of the batteries.

Hardware

The device is consisted by four main components. A battery pack, a Teensy 3.2 microcontroller, a GSM SIM900 module and finally a GY-NEO6MV2 module.

Battery Pack:

The battery pack includes two li-ion batteries (type 18650). These are commonly used in laptop battery packs and are designed to cut-off their power output when their voltage has reached a certain level. This is essential since batteries of this technology can behave destructively when discharged below a certain point. The batteries are connected in series giving a total of 8,4 Volts when fully charged and a total of 5,6 Volts before their internal protection circuit disconnects them.

Main Microcontroller:

The project utilizes a Teensy 3.2 microcontroller although its much cheaper brother, the Teensy LC, would be more than enough for this job. The microcontroller was chosen for its already built-in UART buses which are utilized for communicating with the GPS and GSM modules and for the already built-in USB port which in turn is utilized as a programming and debugging port.

GSM Module:

The chosen GPS module was a SIM900 chipset in a shield board designed originally for use with a more common Arduino UNO. This was very helpful as the module had its own voltage regulator which was used to power the rest parts of the device. The power consumption of the extra modules connected to the GSMs is small enough not to excessively overuse its power regulator since the GSM module is lightly used.

GPS Module:

For GPS tracking a GY-NEO6MV2 module was used which comes with an easy to use library for communication, making the whole development process a bit easier.

Power Management:

In order to conserve energy and make the battery last al lot longer the device is capable of disabling completely the GPS module, reenabling it only when it is necessary for location tracking (i.e. when the user asks for the device's location). This technique slows down the response time of the device since the GPS tracker needs about 25 seconds after a cold-start to orient itself. Though unfortunate this drawback is a small price to pay for the enormous amount of battery conserved (an average of 50mA is required by the GPS module when in use). Another key modification done to further lower the power consumption was lowering the clock speed of the main microcontroller. This enabled the microcontroller to work at around 11mA. Finally, the GSM module was put in "low power mode" in which it significantly lowers its power consumption at roughly 4mA without really compromising the performance of the device, since the GSM module is primarily used as a "listening" device, something which is hardly affected by the "low power mode"

Power Scenario:

The above Current measurements where done before the voltage regulators, so the power lost by heat dissipation from the voltage regulators is taken into account. The device is mainly fluctuating between two states, the Active State and the Passive State.

Passive State:

> In the Passive State the only active modules are the GSM module and the Teensy 3.2 consuming a total of 15mA

Active State:

> In the Active state an additional 50 mA are consumed by the GPS module draining a total of 65 mA from the Battery

Each time the device is asked to perform a GPS location check and report back to the user, it changes from the Passive state, to the Active state, in which it stays for about 30 seconds (25 seconds for the GPS module to orient itself + 5 seconds for the rest procedures). For simplicity reasons, we will assume 40 second Active State cycle which is more than enough to accommodate for the extra Current needed by the GSM module to exit Power Save Mode and send a message to the user.

If we assume that in an emergency situation a 1000 Active State cycles occur (something quite overstretched just to prove a point) and that the average voltage provided by the batteries is 7 Volts which means 2300 mAh before they are fully depleted. We end up with 11 hours of the device working in the Active State, and another roughly 109 hours in Passive State before the batteries are fully depleted (11 * 65 + 109 * 15 = 2350). Combined the device provides easily 5 days of continuous usage even when its position is acquired by the user very frequently. If the user decides to track the device less frequently, the battery life can be stretched to roughly 9 days.

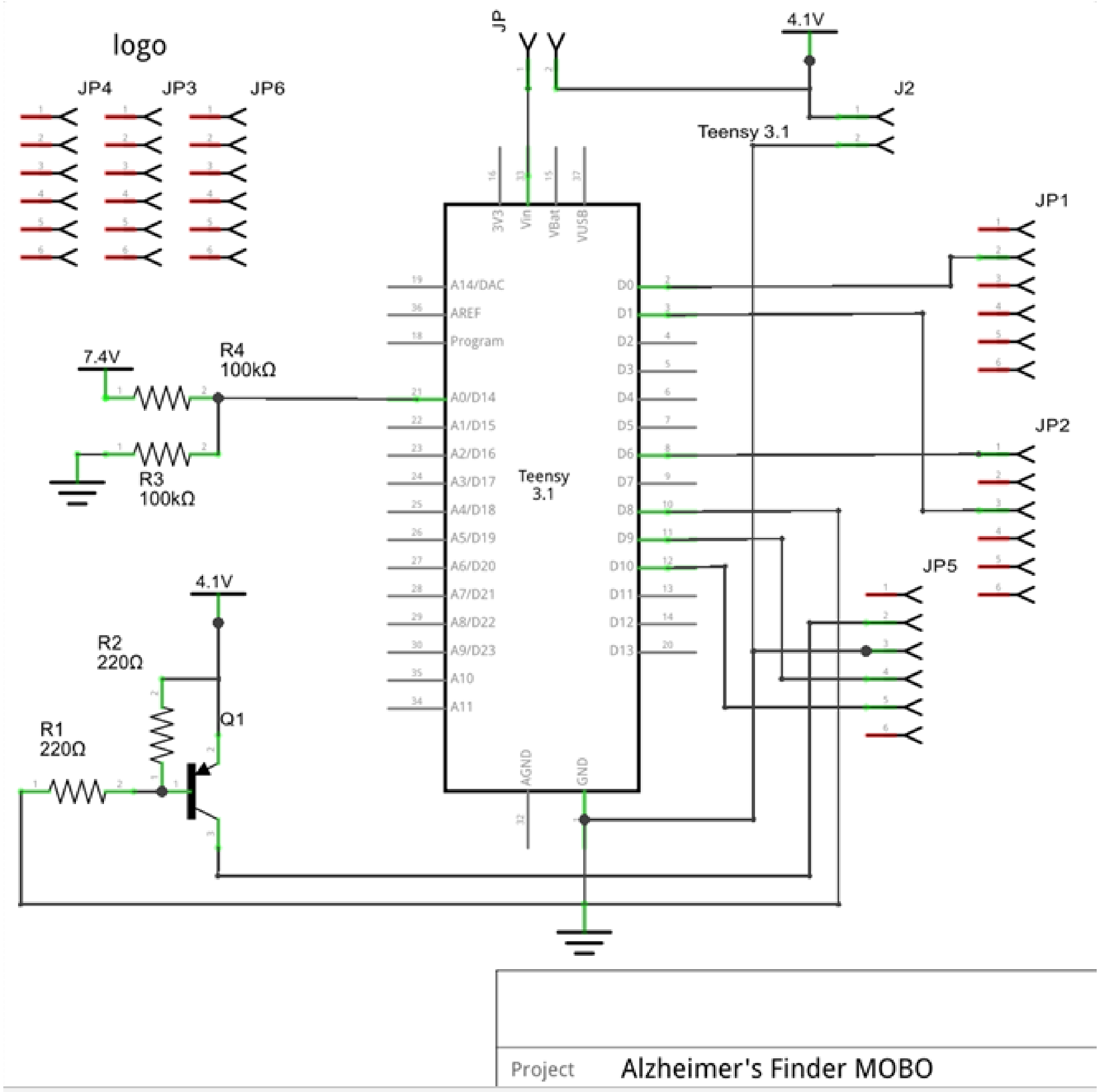
In addition, voltage scaling is one of the techniques that used to reach the lowest power consumption goal. Dynamic voltage scaling is a power management technique in computer architecture, where the voltage used in a component is increased or decreased, depending upon circumstances. Dynamic voltage scaling to increase voltage is known as overvolting; dynamic voltage scaling to decrease voltage is known as undervolting. Undervolting is done in order to conserve power, particularly in laptops and other mobile devices, where energy comes from a battery and thus is limited, or in rare cases, to increase reliability. Overvolting is done in order to increase computer performance.

$$
\begin{aligned}
&P = \alpha\ C_L V_{dd}^2\ f \ \text{ with}\\
&\alpha: \ \text{switching activity}\\
&C_L: \text{load capacitance}\\
&V_{dd}: \text{supply voltage}\\
&f: \ \text{clock frequency}
\end{aligned}
$$

As it is showed in the above equation the power (P) that a CMOS transistor consumes is proportional to the square of supply voltage (V_{dd}). Hence, supplying with the lowest appropriate voltage each module makes our tracker more portable.

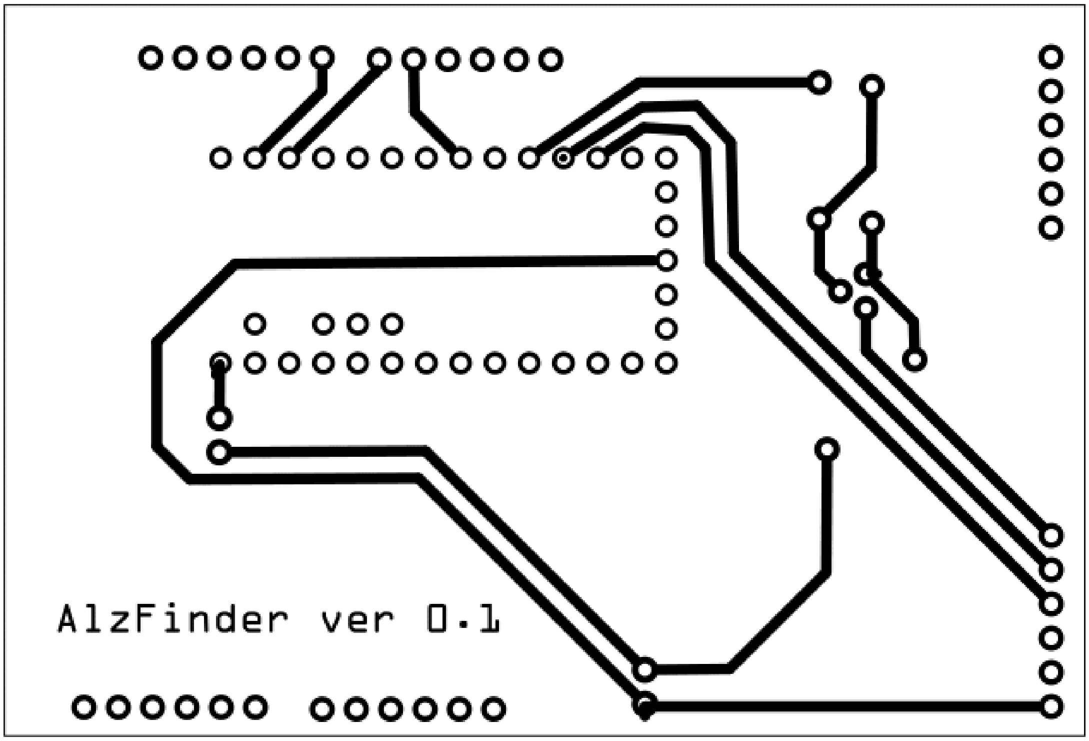
Schematic:

The purpose of this circuit is not only to connect correctly the three modules used for this project, but also to enable the Teensy microcontroller to power-up the GPS module only when it's needed, thus conserving precious energy. The microcontroller and GPS module share the power coming from the GSM board which contains a rather beefy Voltage regulator. A voltage divider is used before the GSM's voltage regulator so the current state of the batteries can be monitored by the microcontroller using one of its analog inputs. This schematic contains only the Teensy microcontroller and no other module, since all the other modules are connected to the main board through external pins.



PCB Design:

The PCB design is based on the outline of a standard "Arduino Uno", the reason for that is the GSM module since it was originally designed to be used as a "shield" in conjunction with an Arduino Uno. This helps the PCB and the rest modules to be connected all together in a tight-spaced and sturdy manner. An L shaped connection has been formed with another board in the far right of the main PCB for the GPS module to be connected. This does not show in the main PCB but is the reason why some extra pins are used for structural support. This L shaped connection can be seen in pictures provided further down this document and was created so that the GPS module will be facing upwards when the hole device is mounted to a patient.

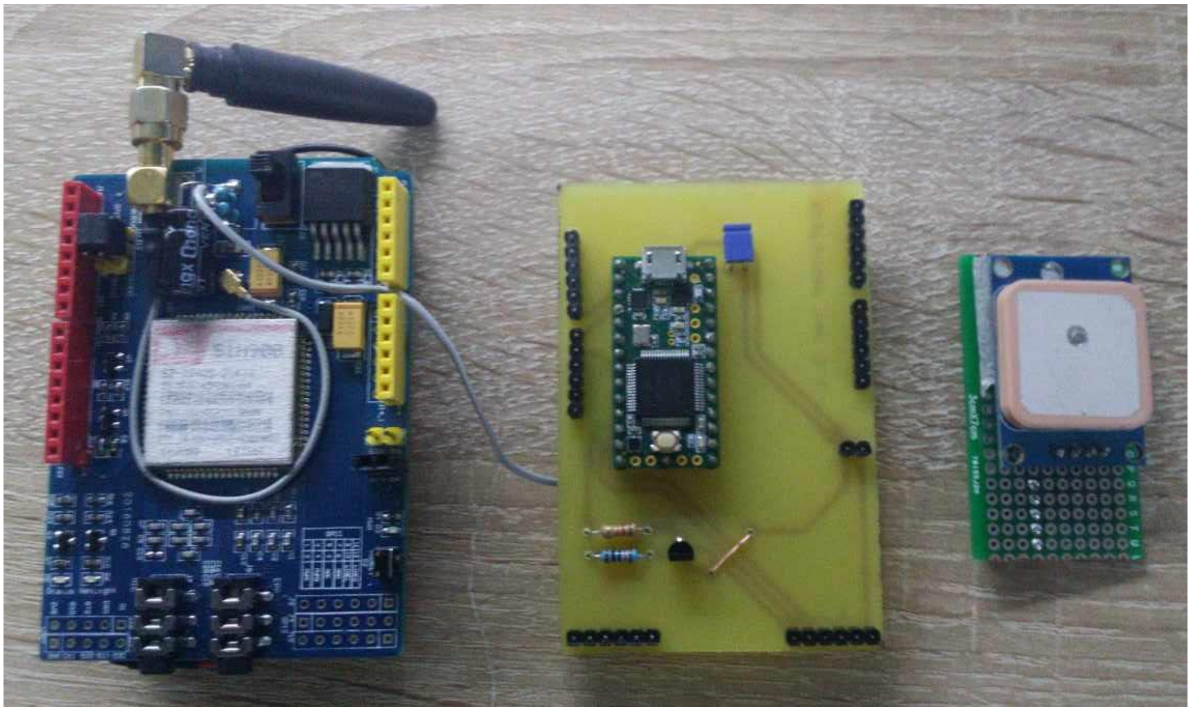


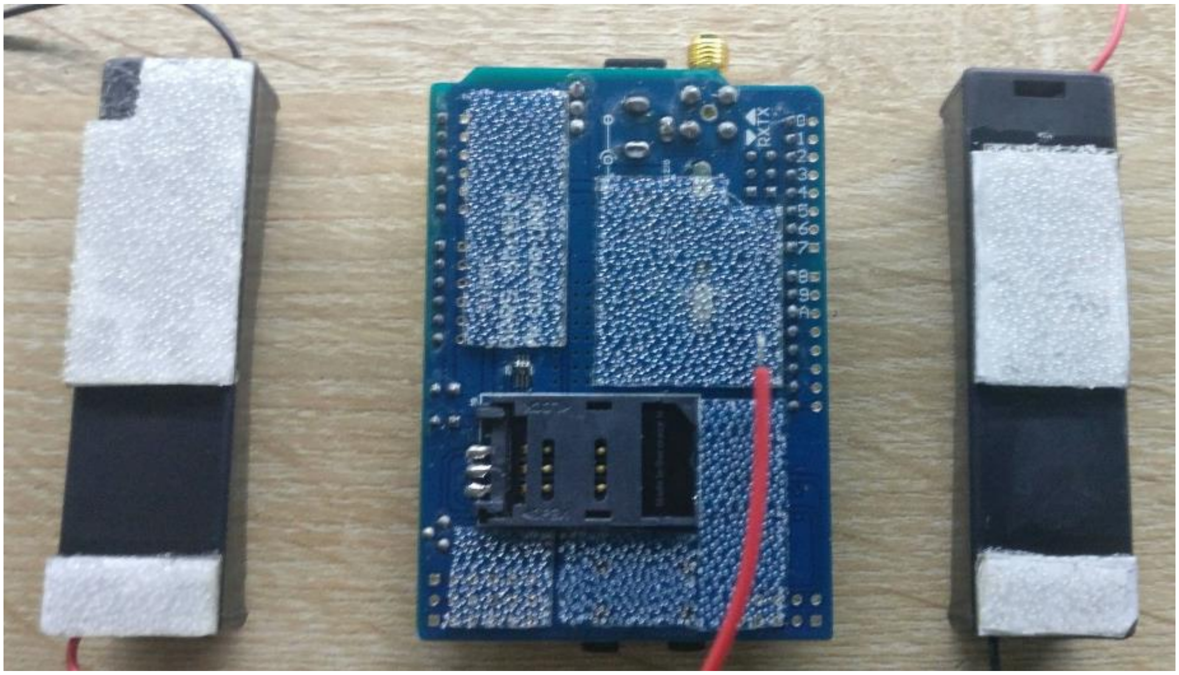
Battery Mounts:

Two battery holders are used for the batteries so that they can be extracted from the device and charged externally. Although this charging method is requiring more effort from the user, it enables the devise be quickly refitted with freshly charged batteries, ensuring it will always be mounted on the patient and function. The battery holders are attached to the GSM module using some "3M Dual Lock" (A more industrial and sturdier version of the commonly used "Velcro") so they can be removed for a SIM card to be inserted.

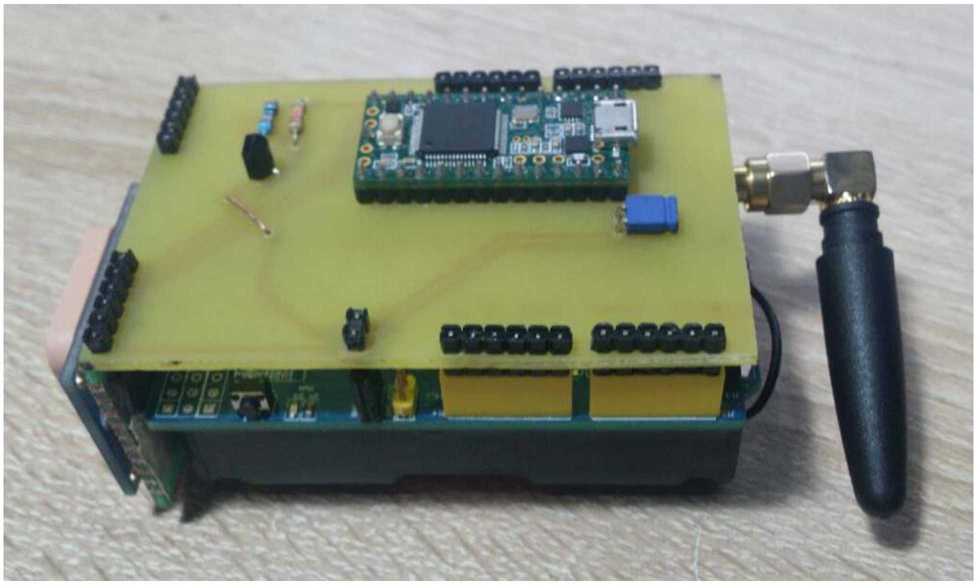
Build Stages and Final Product:

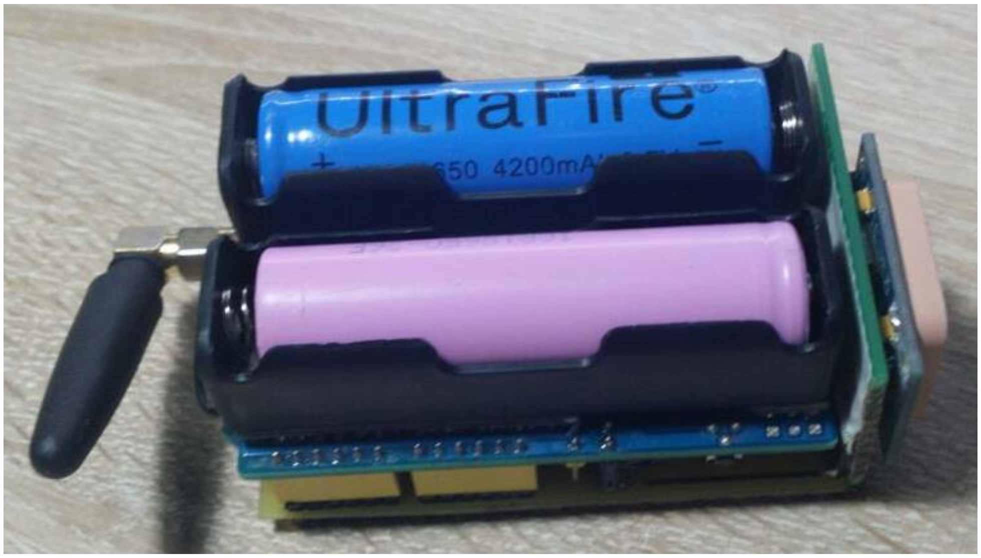
Unconnected Modules:





Connected Modules:



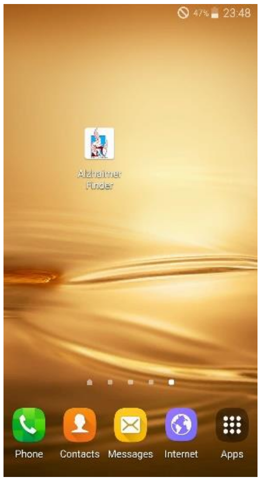


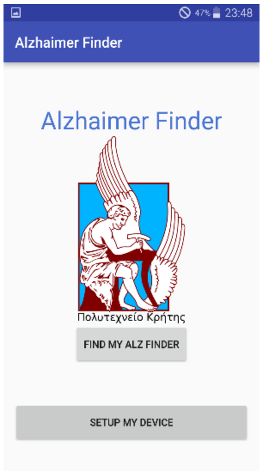
Final Product:

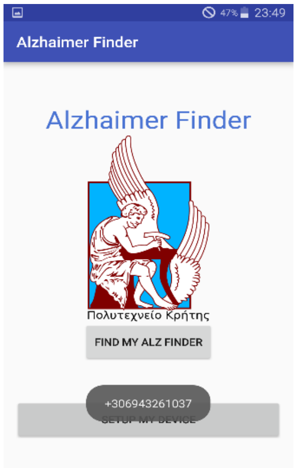
Software:

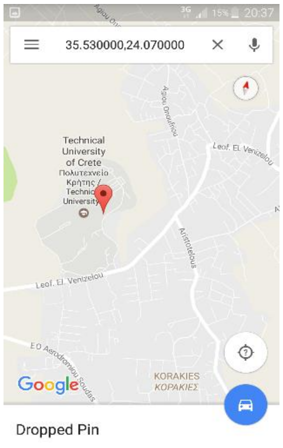
One of the basic parts of this project was the firmware to drive the GPS, GSM, AVR modules. The manufacturer of the GPS module provides the appropriate driver to handle the communication process with the Teensy's UART inputs and outputs. Based on that library, we adjusted the baudrate of the GPS to synchronize it with the GSM and AVR respectively. Furthermore, the appropriate drivers were developed to connect the AVR with the GSM module. Generally, network modules get as input command operations that are written in the serial port and whether the operation successfully finished, a positive acknowledgment is sent to the master module. Taking this into consideration, we developed a driver which comprises that AT Commands in order to send automatically SMS, receive SMS, and change the type of network between 3G/H+/E. Controlling the network type was the key to reduce the power that the GSM consumes, as the "E" network type demands the lowest energy in contrast to the other network interfaces.

Last but to least, the android application was built using the Android Studio platform and Java. The application is consisted for the "Settings" button, that is used to set the initial parameters (Number of the GSM module, the rate of the sampling, and etch). In addition, with the button "FIND MY ALZ FINDER", as it is depicted in the following images, the cell phone sends a message to the tracking device. The returned message comprises the URL with both the longitude and the latitude coordinates. Finally, that message is translated as a "fingerprint" on the google maps.









Future Improvements:

Possible Hardware Improvements:

- Replace type 18650 Li-Ion Batteries with Two Li-ion Cells providing the same power but in a more compatible shape, saving space and making the whole device smaller. In addition, the lack of battery holders provides enough extra room for a Li-Ion battery controller, introducing USB charging capabilities.
- Replace SIM900 module with a similar SIM800 module, which includes a GPS module. This eliminates the rather large GPS module that is currently used.
- Design a multilayer PCB containing a Cortex-M0+ and the SIM800 chipset.

These refinements, although much more time consuming to design, can actually half the overall size of the device.