Good Samaritan: A Mobile Aid to Help Concerned Parents Monitor Their Young Children

Submitted by:

Sumathi Arumugam
Neharika Bhandari
Ama Freeman
Raphaelle Marcial

Team Project Number: 12

Advisor[s]:

Dr. Marco Gruteser
Dr. Hana Godrich

May 1, 2018

Submitted in Partial Fulfillment of the Requirements for Senior Design Project

Electrical and Computer Engineering Department

Rutgers University, Piscataway, NJ 08854

Abstract

Our main goal of this project is to help parents keep track of the safety of their young children. Small children around the age of 2 and 3 are often too young to have their own phones. However, it is essential for them to have some point of contact with their parents or guardians. Our project is a solution to helping parents know that their child is safely at school or any other predetermined location such as school trips or family outings. We have designed a wearable for the child to have on them that tracks their location and pings this location to a database via Internet. Along with this wearable, we have also developed an Android phone application that updates the parent on the location of the child. The wearable is a wrist wrap that uses an Adafruit Feather M0 WiFi to transmit the location and sensor data to the online, real-time database. We are using Firebase and a web server to provide real-time updates between the wearable and the application. Our implementation is much cheaper than a parent purchasing a phone for their toddler age child. It is also quick and efficient as the parent would get to see periodic updates of the location, temperature and pulse readings of their child to know that the status of the child. These additional features on the wearable are so that the parent can keep track of the child's health status, as well as exactly where they are located at all times. The Android application also implements geofencing and calendar integration, which allows the guardian to see the start and end of the event times as well as the marker locations of the child in a surrounding perimeter that the child would be located in. As a result, these added features to our device make it unique when compared with other devices on the market that simply track location through WiFi.

Table of Contents

1. INTRODUCTION 4

2. METHODS / RESULTS / APPROACH 5

2.1. Methods 5

2.2. Use of Standards 6

2.3. Experiment / Product Results 7

3. COST AND SUSTAINABILITY ANALYSIS 10

4. CONCLUSIONS / SUMMARY 13

5. ACKNOWLEDGMENTS 14

6. REFERENCES 15

1. Introduction

In 2017, according to the FBI, there were 464,324 National Crime Information Center entries for missing children (1). In 2016, the total number of missing children was even higher as the entries into National Crime Information Center was 465,676 (1). These statistics do not include all incidents where the missing child is not reported. So, as can be inferred, the numbers are extremely high, and will rise if enough action is not taken. In the United States, missing children reports are annually almost double that of countries such as Australia, Canada, Germany, India, Jamaica, Russia, and Spain. Many of these reports include children being abducted by family members, abducted by non-relatives, or taken by a stranger. As of March 2018, a total of 924 children have been recovered through the AMBER alert system; however, that is a very miniscule number when compared to the total numbers of how many were actually missing to begin with (2). Additionally, when it comes to children with disabilities and other disorders, this a whole some solution is even more significant to be able to provide.

Since this is problem is so imminent, we wanted to provide a solution to help reduce these statistics. Our first approach was to to try to create some sort of tracking device that would simply track the GPS location coordinates of the child which would then be sent to the Android phone application through a database. As we looked at all aspects of this approach, we wanted to broaden the scope and add specific features that would make the device more innovative and useful, especially when it comes to a parent or guardian looking after their child. These features include a temperature sensor and a heart rate sensor which would help update the parent on the health status of the child.

We have planned to create a wearable device for young children (mainly between the ages of two through five who don't have a phone) placed around the wrist that embeds the arduino feather to keep track of their exact location, the heart rate, as well as the temperature of the child. The wearable will track the GPS coordinate location of the child and pings the location to a database via Internet. We have created an Arduino application that updates the parents or the guardians on the location, and health status of the child. The wearable is a wrist wrap that uses an Adafruit Feather M0 WiFi to transmit the location and sensor data to the online, real-time database. Many devices have only one or two main features, however ours uses calendar integration, geofencing, frequent location updates, as well as temperature and heart rate sensors. Most products do not use all of these features into one affordable wearable like ours, and many also do not use our implementation of the GPS M0 Featherwing to be able to see GPS coordinates on to MIcrosoft Azure database, which can be seen on the Android application.

2. Methods / Results / Approach

2.1. Methods

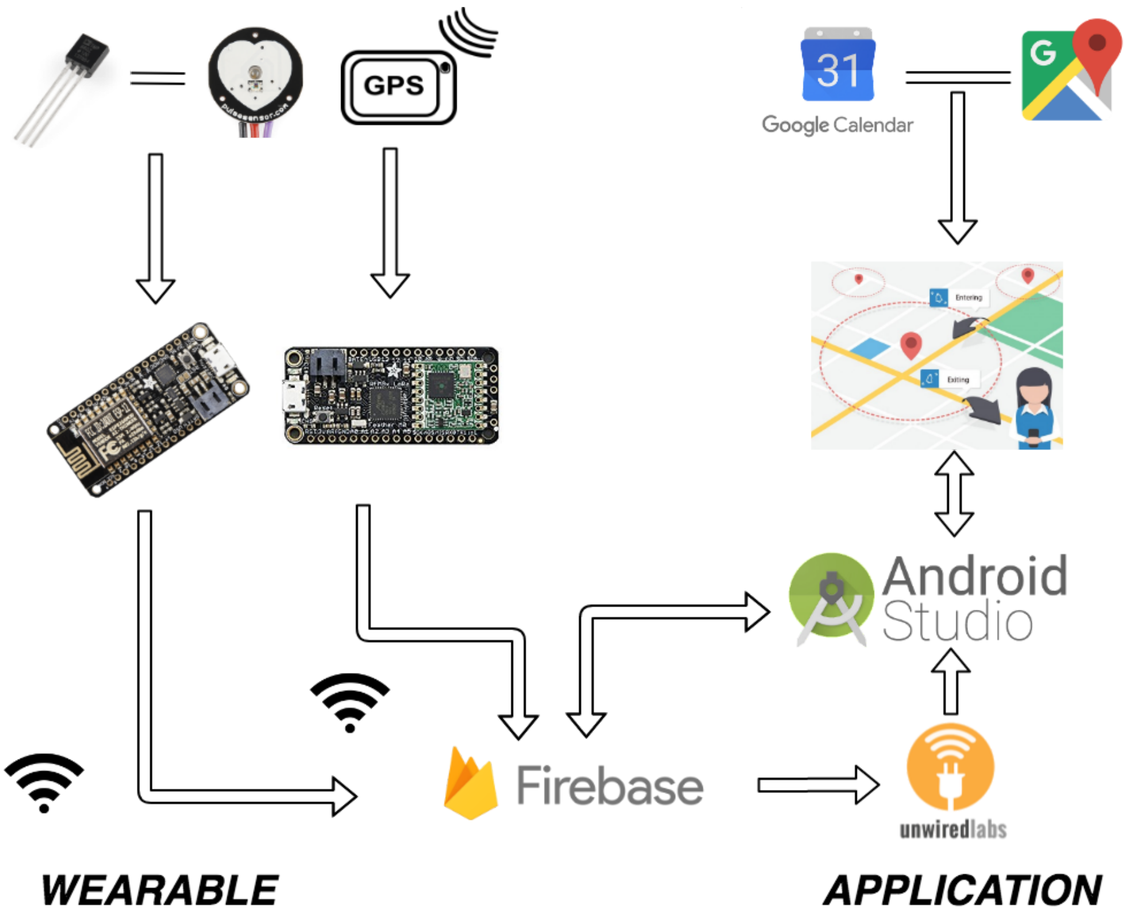

Figure 1: Implementation of the Communication Between the Hardware and Software

The system was divided into two subsystems: wearable (hardware) and application (software) as depicted by the above figure. One of the challenges on the application side was incorporating Google Calendar with Geofencing. Fortunately, there were example applications on the internet which aided in fetching event data and translating the event address into geographical coordinates. These coordinates were used in setting the Geofence while the event duration was used as the duration of the Geofence.

We would have liked to incorporate Bluetooth into our design since this can be useful in close-range location tracking. Also, we would have liked to connect our sensors to one microcontroller. Due to compatibility issues and time constraints, the group was forced to use two in the final design.

This project requires prior programming experience, preferably in Java or C. In order to run the database, knowledge in SQL and JSON is useful. Debugging experience is critical in saving time. Furthermore, programming knowledge in the Android Studio and Arduino IDEs is necessary for this project. Experience in circuit building is beneficial for the hardware portion.

The project also required some basic knowledge of how to connect a temperature sensor and a heart sensor to the Arduino feather. We had to use the latest version of Arduino Genuino 1.8.5, to be able to test as well as run the code to test both temperature and heart rate sensors. Additionally, the project required some knowledge of how to use Microsoft Azure and send data from the FeatherWing into Microsoft Azure.

The hardware involves an Adafruit HUZZAH Feather microcontroller which has a ESP8266 WiFi module to send the wearable information to Firebase. The second microcontroller was a Adafruit M0 Feather which also uses a WiFi module to send GPS coordinates to Firebase. The two sensors involved are a SEN-11574 pulse sensor and a TMP36 temperature sensor. The Adafruit Ultimate GPS Featherwing was the module utilized to obtain GPS coordinates. The software tools utilized were Firebase Real Time Database and Android Studio.

Since we want our device to operate at a low power, we are not able to make requests from our application to the database as frequently as we wanted. Also, this limits the amount of devices which can be connected to our microcontroller.

The GPS was not as accurate as we wanted it to be indoors while WiFi can only transmit data in a certain range since it must be connected to a known WiFi network. A future prototype can include a SIM card in order to allow the wearable to be used when connection to a WiFi network is limited.

2.2. Use of Standards

Many of the modules and software were generally standard for both hobby and project prototype use. There were situations where documentation was abundant and helpful, while in other cases, we had little to no documentation for use cases that we wanted to implement in our project.
On the hardware side, we used a very common model that had many libraries to support the WiFi needs we required: the ESP8266. Not only was this product very well documented, it also catered well to what

we already knew about the C language and microcontroller implementation. The microcontroller was easily programmed using the Arduino IDE and highly responsive to the sensors we attached to it.

On the software side, we had access to many sources in order to accomplish the parts we needed: establish a connection to a database, display data to the user, implementing Google Maps and much more. We utilized the Android Development website, various features from Google's API for location purposes, and a third party API called Unwired labs, whose API was crucial for determining location as well. However, we had to utilize each part seamlessly, without certain parts stepping on top of each other and not working properly. A majority of the work on the application side was combining all of the parts together in order to create one cohesive application. Many bugs were encountered in this process, and it took longing to combine features than to individually test each feature to make sure it would work correctly. However, the parts were able to come together and work correctly, as intended due to the standardized libraries and our own hard work.

2.3. Experiment / Product Results

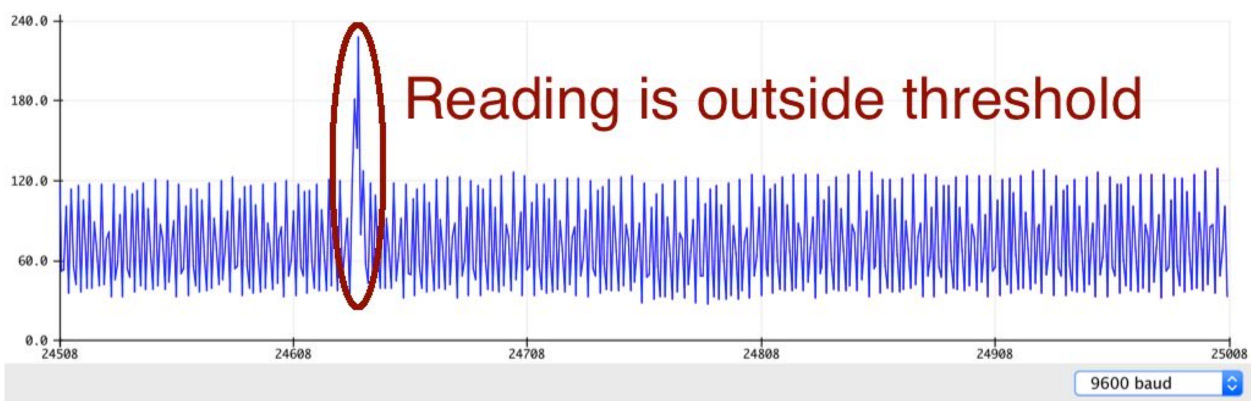

Figure 2: Graph of the Heart Pulse Measurements and a Reading Outside of the Safety Threshold

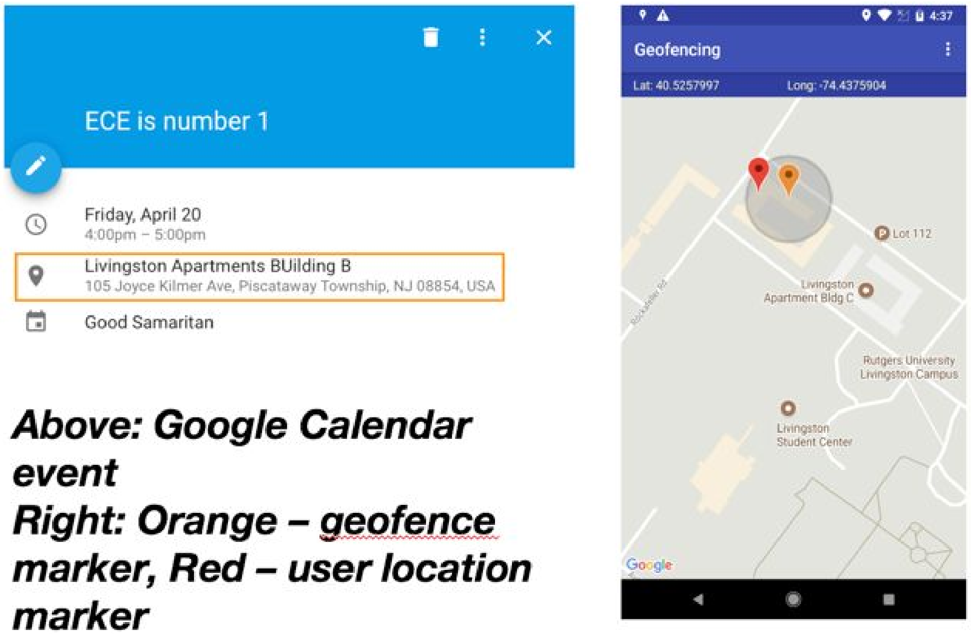

Figure 3: Google Calendar and Geofencing implementations in the Android Application.

```
Satellites: 3

$GPGGA,191441.000,4031.5044,N,07426.1442,W,1,03,3.92,184.1,M,-34.1,M,,*5D

$GPRMC,191441.000,A,4031.5044,N,07426.1442,W,0.28,122.27,230418,,,A*71

$GPGGA,191442.000,4031.5043,N,07426.1441,W,1,03,3.92,184.1,M,-34.1,M,,*5A

$GPRMC,191442.000,A,4031.5043,N,07426.1441,W,0.30,134.27,230418,,,A*78

Time: 19:14:42.0
Date: 23/4/2018
Fix: 1 quality: 1
Location: 4031.5044N, 7426.1440W
Speed (knots): 0.30
Angle: 134.27
Altitude: 184.10
Satellites: 3

$GPGGA,191443.000,4031.5041,N,07426.1438,W,1,03,3.92,184.1,M,-34.1,M,,*57

$GPRMC,191443.000,A,4031.5041,N,07426.1438,W,0.32,137.72,230418,,,A*74

$GPGGA,191444.000,4031.5040,N,07426.1436,W,1,03,3.92,184.1,M,-34.1,M,,*5F

$GPRMC,191444.000,A,4031.5040,N,07426.1436,W,0.33,134.68,230418,,,A*75

$GPGGA,191444.206,4031.5040,N,07426.1434,W,1,03,3.92,184.1,M,-34.1,M,,*59

$GPRMC,191444.206,A,4031.5040,N,07426.1434,W,0.32,134.40,230418,,,A*78

Time: 19:14:44.203
Date: 23/4/2018
```

Figure 4: Readings from the GPS Module displaying both latitude and longitude of the module's location. The data displayed in terms of $GPGGA, GPRMC and so on is the raw data which is correctly extracted and parsed to show the information in a readable format.

3. Cost and Sustainability Analysis

For the first prototype of our wearable, we have many parts that sum up from a variety of costs. To start off, The Adafruit Feather M0 cost us $40.00. This piece allowed us to retrieve the GPS coordinates using Arduino Genuino of the location of the wearable, when connected to the Adafruit Ultimate GPS Featherwing. This Adafruit Ultimate GPS Featherwing cost us $39.99. Our 3.7 Volt Lithium battery was $5.95; these batteries came attached with two pins that we used to connect it to the Arduino board. The full cost of the craft fabric wrap for the design of the wearable was $14.80. This red fabric for the design of the wearable is a weather-proof unique fabric that can be used to cut, glue, or sew with. Next, the two sensors were the temperature sensor and the heart rate temperature. The three-pin temperature sensor cost us $9.99 and has a voltage input from 2.7V to 5.5V in DC. The pulse sensor cost us $24.89, where the pack included Vinyl protective dots, Velcro dots, matching earclip, velcro finger strap, a 24-inch cable with header pins. The total cost for one prototype would then be $135.62 according to the table below. Another cost that must be mentioned is that of the hours spent coding the application and the microcontrollers. Total hours have not been completely counted while working on this project, however the cost should be considered as the average payment that software and electrical engineers would be paid to design this product.

Costs could possibly be lowered for the physical prototype in many ways. Firstly, a new and smaller chip would need to be designed in order to purchase a smaller wristband for the wearable. However, with the design of a custom chip, there would be work hours included in designing this new chip, thereby keeping the cost the same or increasing it even more.

Part List	***Cost***
Adafruit Feather M0	$40.00
Adafruit Ultimate GPS Featherwing	$39.99
3.7 Volt Lithium battery	$5.95
Craft Fabric Wrap	$14.80
Temperature Sensor	$9.99
Pulse Sensor	$24.89
Total Cost	$135.62

Table 1: Accumulated Cost Analysis

The environmental impacts of our project are particularly low. For the first prototype, we utilized the parts by combining them all together into a rather bulky prototype. However, a second prototype would utilize all features but be combined in a smaller custom designed chip. Since we have not created a second chip, we must predict that the smaller size would require far less resources than purchasing separate parts for creating the wearable.

Other factors of energy consumption include the battery power consumed by the wearable and the battery consumption of the user's cellular device. These features are customizable as battery consumption for the wearable can be controlled by setting how often it runs a wifi scan and sends the information to the database. The same can be said for the Android application; the periodic check would need to be customized for a certain time limit.

Our developed product impacts the lives of many families. It provides parents and guardians with a sense of comfort, knowing the location, health, and status of their child at any given point in time. Because this product is catered towards children between age 2 and 5, it definitely poses a positive impact because according to the FBI, in 2017 there were 464,324 NCIC entries for missing children. This product can definitely reduce that number significantly if a parent and guardian can know the status of their young infant.

It addresses community needs as the safety of children for both households and school districts in the community. This accounts for the location of the child whether they are with parents, guardians, teachers, or anyone else in the community. In terms of personal needs, the parent/guardian can keep the child's vitals in check using the temperature and heart rate sensors.

We believe that the product will change consumption patterns. Because we have implemented the Adafruit Feather M0 where as most companies who use these types of trackers don't have this implemented in the wearable. Additionally, our added features such as geofencing and calendar integration within the product including event times and marker locations of the child provide a more wholesome aspect since it integrates many types of tracking on to one device.

The use of this product mirrors that of a caretaker, but cannot fulfill all their responsibilities. Therefore, it is making this responsibility more autonomous yet there is much work to be done on this product to impact the employment of caretakers.

The product would create new jobs in the sense that it would allow software developers to work more on the application. It also would provide more downloads for the Android application.

This product fills a niche in the current economy for both electrical and computer engineers. The wearable itself requires great knowledge of hardware systems and circuitry. On the other hand, the software application requires skill in database usage and mobile application practices. Both groups can work together in creating a product that is highly impactful in today's society but also challenging to develop and maintain.

In terms of safety, a more practically sized wearable should be designed that takes into consideration how the child or user will be able to wear it. In particular, the temperature sensor itself can reach high temperatures and become too hot to wear. A better and more modern sensor would be a high priority for this wearable.

This product can be used for the protection of many, however, it can also be abused by potential hackers if data is not protected well between the wearable and the cellular device. Therefore, a structured and well learned security team should also be utilized in the production of this product.

4. Conclusions / Summary

Overall, we believe that our project will help improve child safety greatly, especially when compared to other products in today's market. Due to our implementation of the M0 Featherwing, we were able to extract the longitude and latitude GPS coordinates which can then be seen in our Android application. Summarizing our activities and results we achieved for the wearable includes achieving accurate sensor readings, establishing real time database between the wearable and the application and periodically updating the database with sensor readings. In terms of the application, we were able to utilize Google and Unwired Labs API's to determine accurate location of the child, implemented geofencing, and retrieved data successfully from database to be displayed in the application.

The significance of our product compared with other existing solutions is that it improves the safety of infants no matter where they are. This is evident in our implementation since we use WiFi to monitor the child when indoors but switch over to GPS coordinates when monitoring outdoors. Because a child's safety is our utmost priority, we ensure that the child is periodically monitored given that we also have sensors attached to the wearable which sends updates to the parent quite often. For future iterations, we hope to ensure that all our modules and software that we plan on using are compatible with each other before attempting to implement the project. In terms of further development of the work done, we hope to make the wearable itself smaller in size where all our sensors along with the GPS module is integrated in a single chip, which perhaps may enable us to design our own printed circuit board.

5. Acknowledgments

We would like to thank our advisors Dr. Hana Godrich and Dr. Marco Gruteser for their guidance along with the ECE Department for their generosity. Thank you to our families who have supported us throughout our college careers. We would also like to thank Google and Unwired Labs for letting us use their respective APIs.

6. REFERENCES

[1] *Google Maps APIs | Google Developers*, Google, developers.google.com/maps/.

[2] API, Unwired Labs Location. “Unwired Labs Location API - Geolocation API, Cell Tower & WiFi Geolocation Database.” *Unwired Labs Location API - Geolocation & Mobile Triangulation API*, unwiredlabs.com/docs#geolocation.

[3] “Documentation | Firebase.” *Google*, Google, firebase.google.com/docs/.

[4] “Free Image on Pixabay - Pedestrian Crossing, Traffic Sign.” *Pedestrian Crossing Traffic Sign · Free Vector Graphic on Pixabay*, pixabay.com/en/pedestrian-crossing-traffic-sign-160672/.