The SmartKart: A Smart Shopping System

Victoria Abreu, Christina Heagney, Lucas Ryan, and Doran Senior

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The SmartKart is a portable system that interfaces with a mobile app to improve the everyday shopping experience. RFID technology will be used to track items that have been added or removed from the customer's shopping cart, while a custom mobile app will display current coupons available, user account information, and handle automatic payment transactions for an easy, no-line checkout process. A GPS system will also be employed to monitor the cart's location at all times, to help prevent theft. The SmartKart system will also assist retail stores in decreasing overhead, and collecting customer and product data, which is critical for financial decisions.**

***Index Terms* — RFID tags, Global Positioning System, Wireless communication, Regulators, Databases, User interfaces.**

I. Introduction

Shopping has become a regular routine and a necessity in our everyday lives, however, there are many issues that may come along with it. Currently when a person goes shopping, they may encounter many hassles that can be averted.

Regardless of the store, a customer can typically expect to wait in long lines before cashing out. The average customer spends time searching for and collecting coupons, and if forgotten, are rendered useless. Additionally, customers with a budget must continuously check that they have haven't exceeded their allotted funds. With all these factors, a simple task like shopping can become hectic and time-consuming.

With our system, shoppers will be able to greatly cut the time spent at the grocery store, by eliminating the need for standing in line. Our system will also allow users to save more time by eliminating the coupon-cutting process. This will not only be helpful to customers who routinely cut coupons, but it will allow other shoppers who traditionally do not cut coupons to save money as well.

The SmartKart will employ an RFID system, which will keep track of all items a shopper places in their shopping cart. From there, this information will be sent to the user's smart phone, where they will be able to easily view the total cost of the items in their shopping cart, and any coupons relevant to their items that are currently offered at that grocery store. In addition to this, users will be able to view all previous purchases made from their account, and all receipts will be sent to their email account.

The SmartKart shopping system will also employ a GPS and locking mechanism, to help the grocery stores prevent theft of the carts. The locking mechanism will be attached to the wheels of the cart, so that when the GPS system detects that the cart has traveled outside of the store's perimeter, the lock will engage and the cart will lose mobility.

II. System Components

The SmartKart shopping system may be broken down into a system of units. Each individual unit performs its own task, while also interfacing with the other components in the system. This section provides an introduction to each of the components used.

A. Microcontroller

For this project, a microcontroller is needed to perform essential functions, allowing different units to interface with each other. The components that the microcontroller will operate are the RFID sensor, GPS module, WiFi module, and locking mechanism on the wheels.

When the RFID sensor reads an RFID tag, the microcontroller will need to pull the unique tag number from the sensor and send that value to the application's database via the WiFi module. Additionally, the GPS module will constantly need to update the microcontroller on the location of the shopping cart. If the condition for valid locations has been broken, then the microcontroller will activate the locking mechanism to halt the wheels.

B. Radio Frequency Identification

The SmartKart shopping system will employ the use of RFID technology in order to keep track of the items placed in the user's cart. All items within the store will be encoded with a unique RFID tag. The RFID sensors will be affixed to the interior of the shopping cart, so that when the user adds a new item into their cart, our system will be able to detect what the item is, its price, and eventually scan the store's database to identify any potentially available coupons associated with that product.

C. GPS Locking System

For this product, a GPS module will be needed to track the shopping cart's position relative to the grocery store.

The software within the microcontroller will use the GPS module to constantly locate the position of the cart. The location of the grocery store is fixed, and its coordinates will be known. Thus, if the GPS module's latitude/longitude coordinates are detected outside of the set range of the store's boundary, the microcontroller will lock the wheels of the shopping cart. This procedure would be used to help in the prevention of shopping cart theft.

D. Wireless Communication

The WiFi module will be used to communicate with the progressive web application. Any information gathered by the hardware components, such as the RFID tags on items, or the GPS location of the shopping cart, will be transferred to the mobile application to be further processed and displayed to the customer's account, via a WiFi connection.

E. Power System

To power the hardware components of our system, the Power Patrol Sealed Lead Acid Battery will be used. This battery provides 12V, a 5A current, and will last approximately 20 hours before needing to be recharged. Power from the battery will be pushed directly to the lock, but will be fed through a 3.3V voltage regulator before reaching the GPS, WiFi, and RFID components of the system. The overall setup of the power system is displayed below in Figure 1.

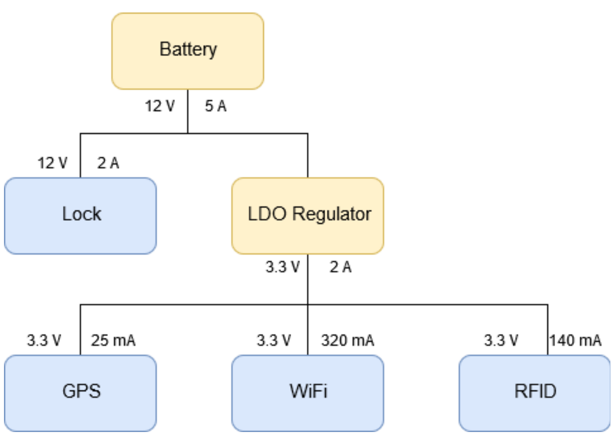


Fig 1. Power system design

III. SYSTEM CONCEPT

To display the overall system concept, a block diagram is provided. This diagram offers a high-level overview of the SmartKart, and how each component interfaces with one another.

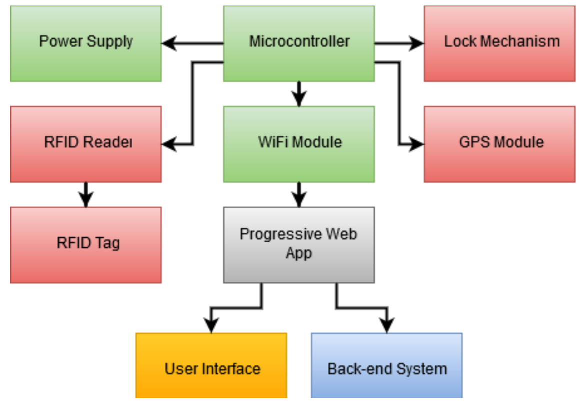


Fig 2. System block diagram

As pictured, the microcontroller unit interfaces with all hardware components of the system. Data is transported between the RFID reader and RFID tags to determine which items are inside the shopping cart, while the GPS module interfaces with the lock to add an anti-theft mechanism to the cart. The WiFi module is used to link the progressive web application to our system, so that data read from any of the hardware components can be transmitted to the backend of mobile application, processed, and relevant information will be displayed to the user on the frontend.

IV. SYSTEM HARDWARE CONCEPT

The hardware setup of the SmartKart consists of multiple units, including the microcontroller, wireless communication module, and RFID system. The RFID sensor and the WiFi module are directly related, as the RFID sensor will read in a unique tag for each store item, and will pass that information to the WiFi module. The WiFi module will then send the RFID values to the database, so that the progressive web application can then update the items and overall pricing.

In addition to these hardware components are the GPS module and locking mechanism, and our overall PCB design. The GPS module will receive data from satellites to generate the shopping cart's location, and based on the detected position, the cart's locking mechanism will be enabled. If the GPS detects that the shopping cart is off of store property, it will send the signal to lock the cart.

A. ATmega328P

For this project, a microcontroller is needed to perform all essential hardware functions. The ATmega328P microcontroller was chosen due to its 32 KB flash memory capacity, 20MHz speed, and adequate amount of I/O pins provided. Components of this project that the microcontroller will operate are the RFID sensor/tag system, wireless communication module, and GPS module, as well as the locking mechanism for the cart's wheels.

The ATmega328P has an 8-bit architecture, 32KB memory, three timers, a maximum clock frequency of 20MHz, and can operate between 1.8V to 5.5V.

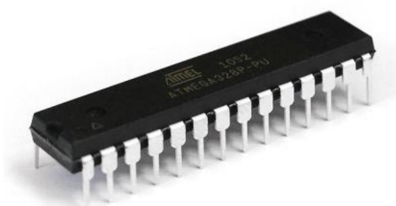


Fig 3. ATmega328P

B. PN532 NFC/RFID

Adafruit's PN532 NFC/RFID kit will be employed to provide RFID functionality for this project. With 2.1”x4.7” dimensions, and a 13.56 MHz frequency read range of up to 4”, this product supports all capabilities that are needed for the RFID portion of the SmartKart system. RFID sensors and tags will keep track of all items a shopper adds/removes from their shopping cart. Each item will have a unique RFID tag. The RFID sensor will be placed on the inside railing of the shopping cart, for easy scanning during shopping.

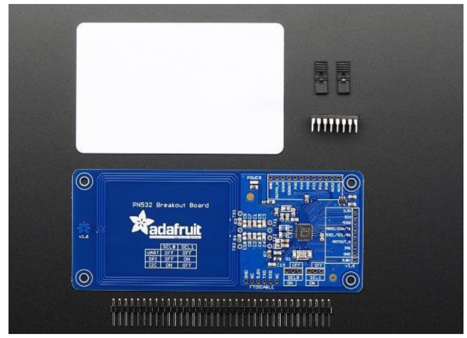


Fig 4. PN532 NFC/RFID

C. Ultimate GPS Breakout

The Adafruit Ultimate GPS Breakout was chosen as it can operate at a maximum input voltage of 5.5V, while the NEO-6M can only operate at 3.6V max. This allows more flexibility for the voltage regulator when supplying an input voltage, and would ultimately dissipate less power. Additionally, the module is NMEA 0183 compliant, has a 9600 baud rate, and is DGPS/WAAS/EGNOS supported. This module may track, at a maximum, 22 satellites on 66 channels, which will allow for better location accuracy.

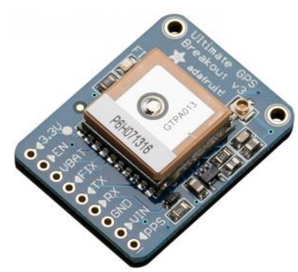


Fig 5. Ultimate GPS Breakout

D. ESP8266-12e

For wireless communication, the component of choice was the ESP8266-12e module. This module is ideal, as it has a connectivity range of approximately 1,000 feet, Serial and SPI interfacing, and is also HTTPS enabled. The module is displayed in the figure bellow.

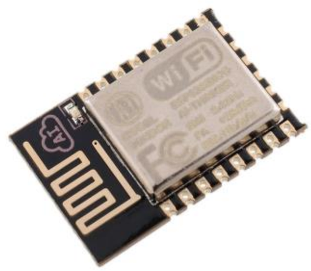


Fig. 6 ESP8266-12e

E. Voltage Regulator

The TSR1-24XX series is a family of DC/DC step-down buck converters that take in 4.75 to 32VDC and step it down to a specific voltage level. One voltage regulator is used to step down 12V to 5V and another is used to step down 12V to 3.3V. It has very good power dissipation, with 91% efficiency.

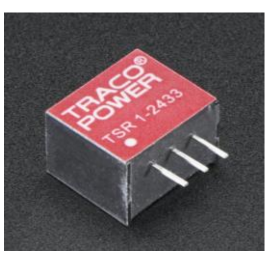


Fig. 7 Voltage regulator

F. PCB Design & Implementation

When implementing the actual hardware components of the project, some considerations were taken. Only one Atmega328P microcontroller was used initially to control the electromagnetic locks, RFID scanner, WIFI module, and GPS module. During testing, however, it was observed that the GPS module and RFID scanner would not function correctly while communicating with the one MCU. The GPS module communicates with the MCU via serial, while the RFID scanner communicates via the I2C interface.

The exact reason as to why the two components could not function correctly on the same MCU was never found. It is suspected that the C++ libraries used to create the embedded software somehow conflicted with one another. Another suspicion is that the 16MHz clock rate the external crystal supplies is not enough to execute the code to control the RFID scanner and GPS module in an efficient manner.

The workaround was to modularize the components. To accomplish this, two Atmega328P MCUs were used. Since the RFID scanner needs to pass a 16-byte RFID tag to the WIFI module, one MCU is used to pass the RFID tag to the WIFI module via serial. The WIFI module then sends the RFID tag to a server via a POST request.

The GPS module is only specifically used to activate the electromagnetic locks when the shopping cart goes out of the range of acceptable LLA boundaries. Because of this, the other MCU communicates with the GPS module via serial and constantly checks to see if the LLA coordinates received are within acceptable boundaries.

The electromagnetic locks are activated by a single pin double throw (SPDT) relay, which is driven by GPIO pins on each MCU. When the circuit is powered on, the MCU that the controls the RFID scanner and WIFI module a GPIO pin to a logical high, which activates the relay. The WIFI module also has GPIO16 set to high in the power on stage, which is connected to another GPIO pin on the MCU. Once the WIFI module connects to WIFI, GPIO16 goes to low, which makes MCU's GPIO pin connected to the relay go to low, thus deactivating the relay.

Additionally, the relay is activated by the other MCU when it receives LLA coordinates that are out of the acceptable range. Each of the MCU's relay GPIO pins have diodes so that the current each one produces only travels to the relay. The RFID scanner, GPS module, SPDT relay, and both Atmega328P MCUs require 5V to function. The sealed lead acid battery to power the system provides 12V, so a 12V to 5V DC/DC step-down buck converter is used provide the correct voltage to these components. Additionally, the WIFI module needs 3.3V to function, so a 12V to 3.3V DC/DC step-down buck converter is used. Figure 8, shown below, displays the final PCB schematic design.

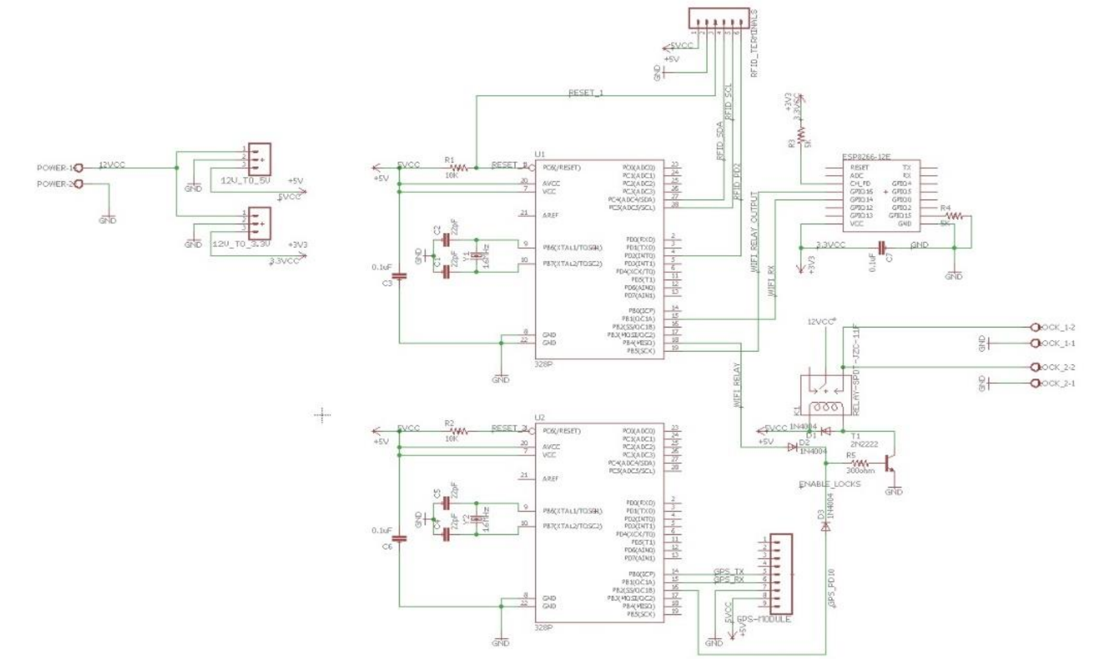


Fig. 8 PCB Schematic

The copper traces on the printed circuit board are two varying widths. The traces carrying 12V are 36 mil wide and the traces for routing signals are 36 mil. The two electromagnetic locks are powered with 12V and each lock has a 1 amp load current. Due to this, the traces that will be carrying 2 amps of current to the locks have a large enough width to carry more than the minimum current. The copper surface thickness of the board is 2 ounces to allow higher current conduction. The PCB is a two layered and the extra space on the board is filled with a ground plane on both layers. Each ground plane is connected with through hole vias. Every component on the PCB is through hole except for the ESP8266-12E.

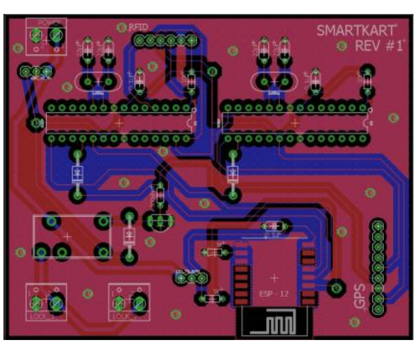


Fig. 9 PCB Final Design

V. SYSTEM SOFTWARE CONCEPT

The following graphic, Figure 10, is the logic diagram representation of our SmartKart shopping system. This diagram displays the software functionality of our system, and details the actions that all components perform.

The system will lock when it turns on and will stay so until we successfully configure the GPS, RFID Scanner, and WiFi module, as well as connect to the network. While shopping, we check to see if we are still within the store bounds every 2 minutes; if the cart sees that it's out of bounds, it will lock itself. While shopping through the store, once a user places an item in to or removes an item from their shopping cart, the system will read 16 bytes from the RFID tag and parse it for the first 12 bytes that map to the item's product in the database, and the last 4 bytes to represent it as a unique product. We then check to see if we already have that item in our shopping cart, to determine if it is being added or removed. The shopping cart table is then updated on the MCU to reflect the action. Next, we send the 12 bytes along with the cart's unique id and the store id to the server to update the user's shopping cart. However, if the connection to the server cannot be made, we halt and physically lock the cart.

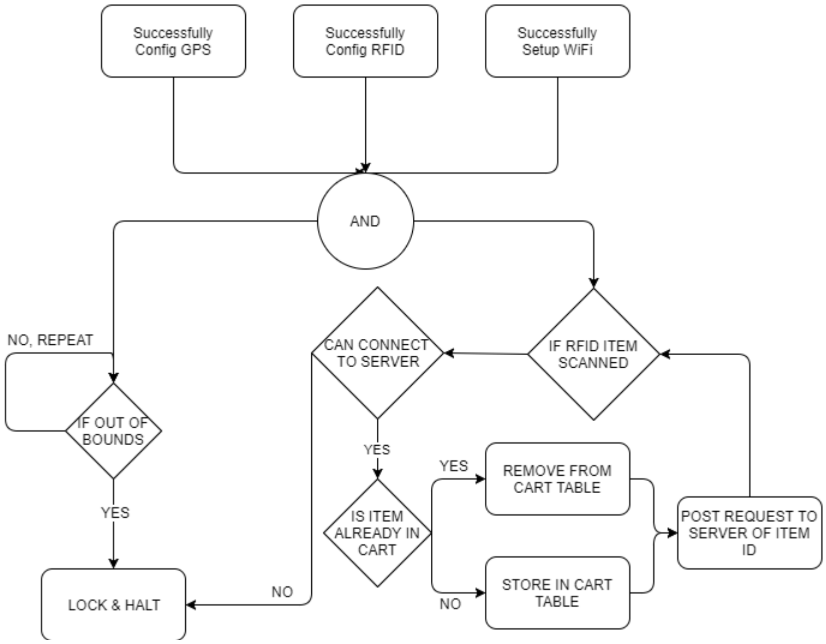


Fig. 10 Logic diagram

A. Socket.io

With an application that requires real-time, low latency, and high throughput data transmission standard HTTP protocol principles are not sufficient. For HTTP to simulate real-time communication, it requires client devices to continuously poll the server for updating information, thus resulting in an increased overhead on the server, increased latency, and high network traffic. A simple tweak is to communicate over UDP to decrease latency times, however, for our application, which involves finances, we need to use TCP as it is a safeguard for lost or malformed packets.

An alternative to this dilemma is to use the Websocket protocol, an application-level protocol on top of TCP that inherits benefits of existing Web infrastructure. Once a client and server complete the TCP handshake the client sends a Websocket Upgrade Request to the server, which the server responds with a Websocket Upgrade Response and initiates an open full-duplex channel to send data.

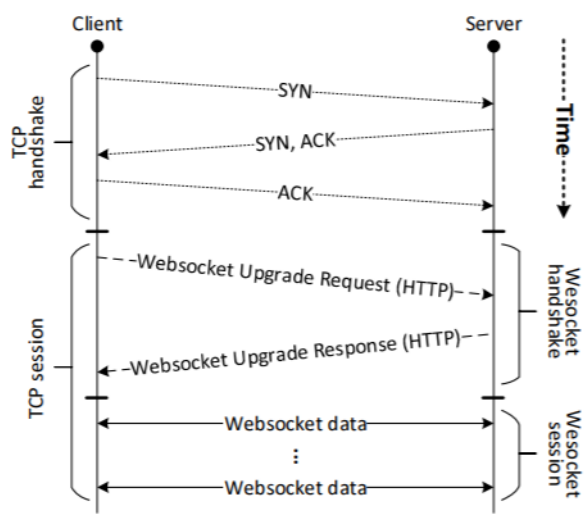


Fig. 11 Socket.io

Using Socket.io we were able to build a scalable, real-time, event driven application that will allow us to notify and update client devices only when they add items into their shopping cart, while allowing them to make their own requests for an update. Latency and overhead would greatly diminish in cases when users in the store for a substantial amount of time, but purchase “many” items. For instance, if a user is at the store and on our app for a total of 30 minutes and we poll every 20 seconds (3 polls per minute), but only purchase 5 items, meaning we make:

$$30\ min * \frac{3\ requests}{1\ min} = 90\ requests$$

Arguably, making 85 unnecessary requests, we could decrease the frequency we poll the server; however, we could no longer guarantee we are displaying the most recent data.

B. PCI credit card

Any software application involved in processing, or storing credit card data it must comply the Payment Card Industry Data Security Standards (PCI DSS). The legal standard is to not store any cardholder data on your system, protect private networks, protect data storage and transmission. If one doesn’t meet government standards, they will be heavily fined and be a potential victim of hackers.

If companies hosting e-commerce sites are unable to put in place the necessary safeguards, they may use third party PCI Service Providers, such as PayPal and Stripe, to handle the payment process and save customer payment sources (bank account, credit/debit cards) to transfer responsibility of securing user financial information to these parties. If the payment source server never touches your server and data is served with TLS for encryption you meet the general requirements to stay PCI compliant.

VI. CONCLUSION

Our team's motivation and goals served as the primary factor in envisioning the SmartKart shopping system. In addition, the requirements and specifications that were shaped in our design process helped bring the project to life.

The SmartKart’s infrastructure is made up of a multitude of different components and modules, including technology such as wireless communication, GPS, and RFID sensors. The integration of these modules into a single, uniform system will allow for the creation of a useful and marketable product for the retail industry and its consumers.

Our team has taken extra measures to ensure that the user’s safety is our top priority. This includes building a hardware system that is safe to integrate into shopping carts, as well as ensuring the software portion of our system is safe for shoppers in regards to the security of their personal information, such as their credit card numbers. The other top concern was the cost of the system; our team was determined to create this product on a reasonable budget, in order to help ensure its marketability potential in the industry.

ACKNOWLEDGEMENT

The authors wish to acknowledge Dr. Mark Heinrich and Dr. Lei Wei, as well as the CECS faculty on the Senior Design Review Committee, for their time and support of this project.

REFERENCES

[1] Understanding Boost Power Stages in Switchmode Power Supplies (n.d.): n. pag. Ti.com. Texas Instruments. Web. 20 Mar. 2017. <http://www.ti.com/lit/an/slva061/slva061.pdf>.

[2] Zhang, Henry J. "Basic Concepts of Linear Regulator and Switching Mode Power Supplies."Cds.linear.com. Linear Technology, Oct. 2013. Web. 20 Mar. 2017. <http://cds.linear.com/docs/en/application-note/AN140fa.pdf+>.

[3] Industries, Adafruit. "Adafruit ATWINC1500 WiFi Breakout." Adafruit industries blog RSS. N.p., n.d. Web. 25 Apr. 2017.

[4] "WiFi Module - ESP8266." WRL-13678 - SparkFun Electronics. N.p., n.d. Web. 25 Apr. 2017. <https://www.sparkfun.com/products/13678>.

[5] "PN532/C1 NFC Controller." (n.d.): n. pag. Cdn-shop.adafruit.com. Adafruit, 31 Mar. 2011. Web. 30 Mar. 2017.<https://cdn-shop.adafruit.com/datasheets/ pn532ds.pdf>.

[6] (n.d.): n. pag. Atmel.com. Atmel Corporation. Web. 30 Mar. 2017. <http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>.

[7] (n.d.): n. pag. Sparkfun.com. Texas Instruments, May 2003. Web. 30 Mar. 2017. <https://www.sparkfun.com/ datasheets/Components/LM7805.pdf>.

PROJECT MEMBERS

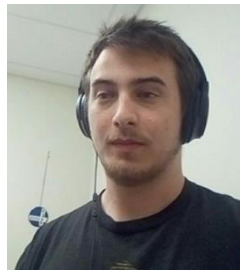
A. Victoria Abreu

Victoria Abreu is a Computer Science major at the University of Central Florida. She is currently interning at Northrop Grumman, and intends to work there as a full-time employee following graduation.

B. Christina Heagney

Christina Heagney is a Computer Engineering student, with experience in frontend development and systems engineering. After graduation, she intends to work full-time, while also pursuing a master's degree in the near future.

C. Lucas Ryan



Lucas Ryan is a Computer Science student with an Interest in robotics and IoT, with internship experience and projects in web development and programming micro-controllers. After graduation, he intends on going to graduate school for Computer Engineering with a focus in Networking and Robotics.

D. Doran Senior

Doran Senior is a Computer Engineering student at the University of Central Florida, with a minor in Mathematics. His field of experience includes developing image processing software for C++ embedded applications and graphical user interface. After graduation, he hopes to develop software in the defense industry.