# Secure Phone Locker with Integrated Notification Tracking

Ian Flemings, Nicholas Lucas, Alexander Masterson, and Evelin Santana

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

***Abstract* — Secure Phone Locker with Integrated Notification Tracking (SPLINT) is a hardwired, physical storage unit designed for facilities that wish to minimize security risk, hazards, and distractions caused by employee or visitor mobile devices by securing the devices in reusable, fingerprint access controlled, lockers which are able to charge the devices. The Lockers will also detect when a particular phone receives a call or general notification, by way of microphone and vibration sensors, and log that information in a an accessible file on the network which can then be accessed from a computer terminal.**

***Index Terms* — Fingerprint control, I2C communication, phone charging, sound detection, vibration detection.**

## I. INTRODUCTION

The main function of this product is to provide companies with a means to secure devices coming into their facility to prevent potential hazards, such as interference with RF sensitive equipment, theft of a company's intellectual property through photos or video, or simply to improve the productivity of the employees to increase company revenue. This is done in a way that should provide the owners of the devices peace of mind, in that their device is secured until they get back, and they can check if they have missed any calls from a computer terminal. With personal mobile devices getting more advanced every day, high security facilities face a variety of problems, ranging from the interference to sensitive equipment caused by the increasingly powerful RF signals generated by cell phones, to the potential loss of intellectual property caused by an employee or visitor taking a careless selfie, and catching some sensitive information in the background. Many facilities that have these issues are rather large where sending someone back to their vehicle to put their phone away is an inconvenient, time wasting hassle. Some facilities deal with this problem by having people put their phones on a shelf or leaving them with the person at the front desk. But with many people having some of their most sensitive data stored on their phones, our SPLINT system provides a secure means to store a person's phone with their fingerprint to where only they can retrieve it. While the phone is being stored it will then utilize built in microphones and a vibration sensor to detect when a phone has received a notification or a phone call. This information is then relayed and stored into a file that is accessible anywhere on the local network.

## II. MOTIVATION AND GOALS

The initial idea for this project was proposed by a group member who spent several working as an electrician. During his time as an electrician, one of the facilities they did jobs for was Kaman Fuzing and Precision Products, a company that manufactures precision safety elements for penetration bombs for the military. Employees at the facility are strictly forbidden from bringing cellular devices into the testing laboratories unless strictly authorized as the devices in those labs are radio triggered. The only accommodation made for employees in this regard was a small metal shelf nailed to the wall. This shelf was inadequate and our team member watched phones get knocked off, damaged, and accidentally taken by the wrong owners on several occasions. We also have all dealt with a situation where a coworker (or ourselves) were playing on a phone when they should be working. These factors all inspired us to create a solution that would help retain work efficiency and promote the safety of these sometimes very personal devices.

We wanted the system to be compact and sensitive. Phones don't take up a lot of space, and the motors and speakers that phones use to alert users of status changes do not trigger with the highest magnitude in the world. We also wanted power and transmission efficiency. These phones may or not be attached to a charging system, so effective delivery of power helps promote reduced operation cost. This system will also be keeping a change log, so quick reaction to devices the phone uses to alert users of a status change is paramount to keeping the log as accurate as possible. We also support the possibility of modularity. Because the hardware for inter-system communication is relatively simple, we can easily add new modules to the communication network. The device should also be accurate in authentication. It would do the user no good if the device cannot accurately differentiate one users authentication from the wrong users authentication. Lastly, we wanted the device to be accessible. With the high volume of networking and rate of information consumption that happens in the modern world, the ability for a user to see the necessary device information from as many valid sources as possible has become very important.

## III. SYSTEM DESIGN

Our system is reliant on the intercommunication of multiple different parts. For our proof of concept, we will be using three phone chambers linked on the Inter-Integrated Chip (I2C) bus of the main communications controller.

The overall system is composed of five parts, a power circuit, the communication network, the locking system, the sensing system, and the housing. The power circuit is in charge of drawing power from a NEC standard 120v wall outlet and distributing that power to the various components of the system, including the phone charging circuit and the various different electronic and microprocessor components in the system. The communication system connects every sensor unit and the locking system to the main communications controller.

That main controller then sends information off to be read from the terminal. The sensing system is designed to sense for sound and vibrations. When the sensors pick up a large enough signal, they will send data to the main communications controller and the controller will take care of the rest. The locking system handles the image capture, matching, image clearing and other necessary functions needed to power the security implementation of the system. The housing will neatly hold together all of the necessary wiring, sensor components, and user cellular phones.

Figure 1 below shows the overall system overview of the project. The components in all three lockers are exactly the same for ease of development. Elements of the power circuit are shown in blue. We step down power from a 120v wall outlet to 12v and we need both a 12v regulator for the solenoid lock as well as a 5v regulator to feed the charging circuit and operational voltages for the controllers in the circuit. We include a level shifter as the two different controllers we use operate at different voltages, one at 5v, and the other at 3.3v. Elements of the locking circuit are shown in beige. Everything in the locking circuit is controlled by the security controller and the security controller will tell the fingerprint module when to operate. Elements of the sensing system are highlighted in pink. We use piezo disks to sense for vibrations and electret microphones for sensing sound.

*Microcontrollers*

The SPLINT system utilizes a microcontroller in . The microcontroller in the sensing unit will be monitoring the inputs from each of the sensors on the unit and doing whatever necessary operations on the data. If this microcontroller is reading suitable data values, it will start storing a flag for a brief period for it to be inspected, after a short time passes, the microcontroller will then wipe the flag. The microcontroller in the locking unit will handle the acquisition of fingerprint images, the verification and image matching, the clearing in fingerprint images, as well as the trigger signals for the diagnostic implements in this system. The diagnostic implements we are using are

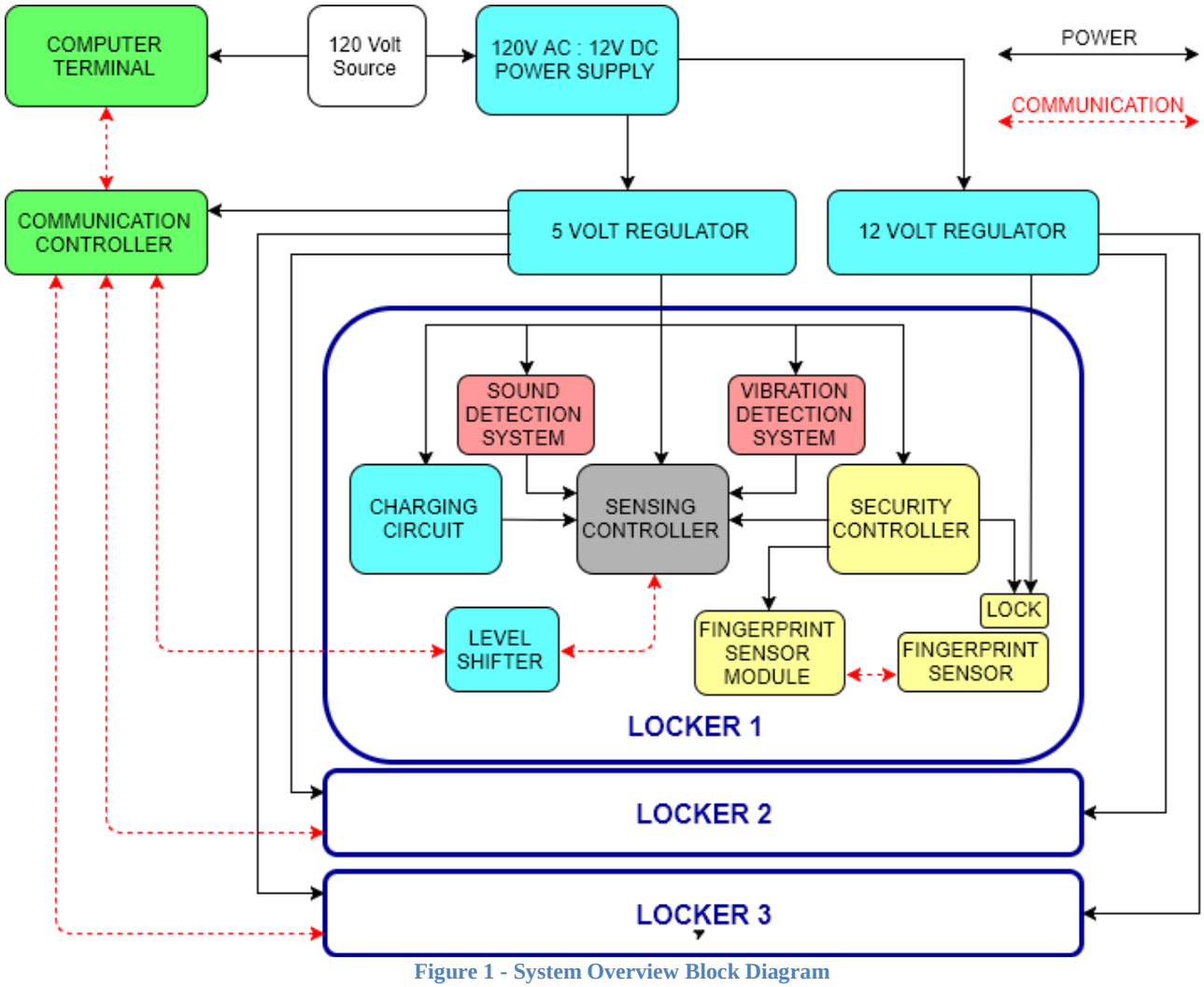


Figure 1 - System Overview Block Diagram

LED's and a small buzzer. The LED's tell the state of the locker and the buzzer plays an audio equivalent of what the LED's are showing.

These microcontrollers all connect to a central microprocessor via a single I2C bus. This microprocessor will continuously be polling the sensor units and the locking unit for specific state data. The microprocessor will then process this state data and transmit the data to be interpreted into the system log which keeps a history of activity in the system for the user to view.

When deciding between the various microcontrollers, we considered the Ardiuno ATMega328 due to its popularity, low cost, and extensive hardware and resource support. We considered the Beaglebone Black due to its Ethernet capabilities, processing speed, and development capabilities. We considered the Rhaspberry Pi for its processing speed and resources. And Finally, we considered the MSP430 due to its low power cost and the groups prior experience with the device. In the end, we went with the Beaglbone Black as the primary communications controller for its Ethernet capabilities, multiple I2C buses, and computational abilities. We also decided on the ATMega328 for microcontroller for both the sensor units and the locking units for their cheap cost since it is the most abundant microcontroller, as well as its relative ease of development and hardware support. This decision also makes development and maintenance easier and cheaper overall.

### *A. Inter-Integrated Circuit Communication*

There are several small subsystems all controlled by their own microcontrollers that all need to communicate to one centralized location. Trying to do all this in such a confined system with wireless would make development more complicated than we felt it should have been, so we opted for a hard wired communication topology. We contemplated between using UART, I2C, or SPI, but in the end we decided to use I2c for inter-device communication. The reason we went with this protocol is that it is the simplest and most modular communication protocol of them all. The wiring network only requires two wires to set up, a serial clock line (SCL), and a serial data line (SDA). One master can easily communicate to any device on the bus by specifying an address, and if in the future there needs to be more modules added to the system, you can accomplish this by just adding the unit to the bus, adding an address in the code, and possibly changing a pullup resister. This protocol requires more power, but the space saved in wiring and increased modularity was worth the investment to the team.

For our system, the master unit for this communication protocol is the Beaglebone Black. The Beaglebone Black has three I2C buses on it, each one capable of housing 105 devices. Two of these buses are hidden from the developer at factory set, but with some extra lines of code they can be revealed and made available for development. This is great for future expansion as we can house a large number of phones, eventually reducing the cost of the Beaglebone Black to less than 20 cents per phone.

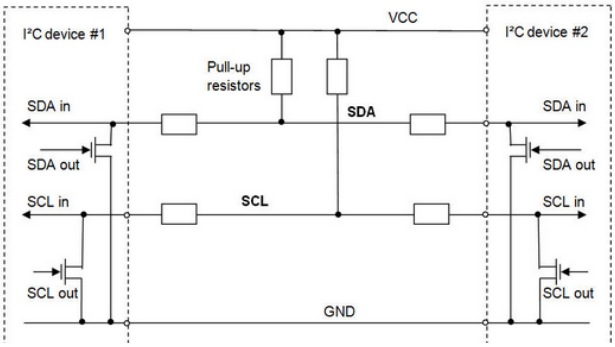


**Figure 2 - I2C Wiring Guide**

## IV. SENSOR SYSTEM

There are sensors in each module to monitor the cell phone inside. When someone places their phone in the module, the sensor will detect if they have a notification by using a vibration and a microphone sensor. The microphone sensor will detect any ringtones from the phone and the vibration sensor will detect any vibrations. The signal that is detected is transmitted to a user so they can determine whether to go out of a restricted area to retrieve their phone and answer their messages or not.

### *A. Microphone Sensor*

The microphone sensor is made of an electret microphone, and an amplification circuit. The microphone by itself gives us a loud enough signal to detect, however the circuit is reducing any unnecessary noise and the transistor is amplifying the signal that we want. The transistor we are using is the 2N3904 which is a NPN silicon transistor designed for general purpose amplifier and switching applications. The transistor is being biased properly by the chosen resistors to provide the appropriate input current and collector voltage conditions so that we have a suitable operating point for the AC signal to the amplified properly. Also, this circuit requires a 5V input for everything to work correctly which is provided by our power supply.

There were other options that we could've picked such as sound sensors with their own built in circuit that could easily be attached to an ATMEGA but for the purpose of our project the electret microphone is a good fit because it is small and light, it is sensitive enough to pick up low frequencies from the phone, its less accurate but more adjustable, so it can be manipulated in the circuit to do what we need, and it converts the sound into a signal that can easily be amplified and sent to the microcontroller. Also, the circuit that we did built takes up a small amount of space in the printed circuit board, so it saves us space and it is cost effective.

We originally had a bigger circuit with an op amp to further amplify the signal, but we determined that it was redundant and unnecessary. It would take the output of the transistor and amplify it by a factor of two, giving us a stronger signal however the ATMEGA could already detect the signal without it so it was removed from the

printed circuit board to save cost and make it more efficient for our purposes.

After we have a clean signal it then goes into the analog pins in an ATMEGA. We have two microphone sensors in each box and the ATMEGA is comparing the values in each one to determine whether to send a flag or not. If both microphones detect a loud sound above a certain threshold then we know it's detecting a ringtone inside of the module it's in so it'll send a flag. If only one of the microphones detects a loud sound, then we know it may be picking up the cell phone ring of a phone in another module so then it does not send a flag. The code in the ATMEGA tells it to sample 100x in a loop for a certain amount of time so that it only sends a signal if the phone is actually ringing rather than a temporary loud noise disturbance. This flag is then sent to the beaglebone which transmits to the user if their phone has been ringing or not and for how long so that the user can determine if they need to check their phone.

*Vibration Sensor*

The vibration sensor consists of a piezo electric disc, resistors and a diode. When you apply a force to the diaphragm of the disc, you are putting stress on the crystal behind it which is a piezo electric material. When the piezo disc is connected to an input, any physical force such as a tap or a change in weight will produce a voltage, so it does not need a voltage input since it already produces its own voltage. The voltage produced is proportional to the change in pressure applied to the disc. The more pressure the higher the voltage, this is essential because then it means the signal does not need to be amplified.

We have the resistors and the diode in the circuit to clamp the output of the disc to 5v, and to limit the current produced by the disc so that it doesn't exceed the maximum analog input for the ATMEGA. This is a good limit so that the ATMEGA can safely work with the signal.

There were other options for sensors that we could have used such as a MEMS accelerometer. While the MEMs accelerometer does work with detecting vibration, they are lower range and high sensitivity devices better suited for structural monitoring and constant acceleration measurements. The piezoelectric disc is better suited for our purposes because it is less sensitive, has a wider range, it is low cost and with our circuit it produces a good voltage with a clean signal.

The signal produced goes into the analog pin in an ATMEGA for it to be monitored. If the cell phone vibrates for a phone call or a text message, then it will send a flag. The code will tell it to sample 100x in a loop and keep looping to see if the voltage produced goes past a certain threshold. For the vibration sensor the threshold is set very low, and it will send a flag with any activity because we want it to be able to detect text messages and a text message is just one quick signal over a period of time. The reduction of false flags is caused by the circuit itself and the discs low sensitivity so that it won't be affected by someone knocking on the door outside of the module. Once the ATMEGA picks up a loud signal, it will send a flag to the beaglebone to then notify the user if they need to step out to check their phone.

## V. Security Subsystem

The primary function of our system is to serve as a secure locker for employee and visitor devices. For users to be comfortable with leaving behind their mobile device, we wanted to ensure that our security system is has a strong means of authentication, and an efficient locking mechanism. In our research, we looked into electromagnetic locks, linear solenoid actuators, motorized electric latches, and linear solenoid latches. The mechanism we selected for our design was a linear solenoid latch. This allows for a secure low profile design with low cost that will be power efficient, easy to implement, and long lasting. While it as heavy duty as an electromagnetic lock, this should not be an issue since the lockers themselves will be in a reasonably secure area. For the authentication there were many available options. We researched integrated circuit cards, numeric keypads, near field communication tokens, and fingerprint scanners and we determined that the fingerprint scanner would be the most secure option, since it part of you and cannot be easily replicated. The other options require something you have, which could be stolen or lost, or they require you to memorize a password, which could be forgotten, or stolen also. The most common readers used for detecting fingerprints are optical, capacitive, ultrasound, and thermal readers. Optical Sensors are typically the cheapest, but fall short in actual optical detection, which means that patterns are very easy to replicate. The sensor we ultimately chose to go with was a capacitive fingerprint module that is manufactured by Grow Technologies. While our system has the potential to store many fingerprints, we only allow it to store 1 person at a time. Our locker is programmed to erase all fingerprints that are stored after each use. We do this to minimize risks posed by potential malicious activity and to alleviate any potential concerns about having one's fingerprints being recorded. The finger prints are also stored on the fingerprint module itself to create an additional layer of separation from any network activity.

*Security Microcontroller*

When selecting which microcontroller we would end up using to handle the security authorization and authentication, we needed to account for all the signals, voltage and current requirements of the parts that are to be controlled and we also need to consider the UART communication that was required to communicate with the fingerprint sensor. The microcontroller we decided to go with for this was the atmega328. This worked out great because it had enough pins for the LEDs, buzzer, and the 5V communication over UART to the fingerprint module. The one complication we had to deal with was to actuate the solenoid lock using the atmega328. Because the lock required 12 volts to operate, we had to come up with a solution to control it with the atmega328. We chose to use an N-channel logic level Mosfet to be able to use the output from the atmega328 as the gate voltage to allow the solenoid to actuate. The physical

## VI. PHYSICAL DESIGN

The physical housing for the SPLINT system is modular by design, meaning we can easily replicate and add new modules to the system if we ever want to expand. The extra benefit of this is that we can easily 3D-print extra modules for lower cost as we can use the same .stl files. Our housing will be printed using high temp PLA as the base material. This material is rigid, light, very inexpensive, and not electro-statically inclined. It is almost heavy duty for our purposes, but its inexpensive enough that it is only a benefit.

There were not too many considerations that needed to be taken in regards to structural integrity. The box itself should ideally be placed in an office/lobby environment where the worst thing that can possibly happen to it is someone misses the storage space and hits the box with their phone. The only thing that really needed to be considered as far as physical design is concerned is leaving spaces for the necessary circuitry components. As far as operational concerns we have as for the structural design, we have to consider sound and vibration bleeding through from one box to the next. To tackle the sound problem physically would require very expensive materials, so instead the sound problem is tackled electrically in the sensor system. Vibration is handled physically with cheap shock isolating pads.

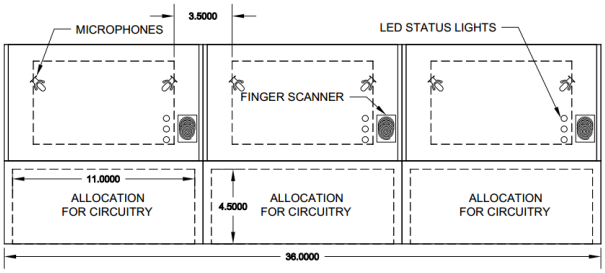


**Figure 3 - Front Demo View**

### *PCB Design*

Our PCB was designed using Altium. For this project, the team went through two iterations of the PCB design. The first iteration was large, and had a lot of poorly used space. We also faced a complication with the first iteration of the design as by the time the fabricated PCB came in, the team had already made design changes to the overall system. We had made more consideration for power efficiency and improved operation of other subsystems, so some of the pins and leads on the original board were now either no longer being used, or completely wrong. The company that made the board also made mistakes in that some of the leads on the board were not properly connected to each other. During the time the team was redesigning the second generation PCB, we had changed some of the parts in the system to more efficient versions. The team got a new microphone component that was better than the prior one, and it rendered part of the amplification circuit redundant and unnecessary. The redundancy added unneeded distortion to the input signal, so the team eventually removed that part of the circuitry entirely. The second generation PCB is overall better as the removal of the unneeded circuitry allowed us to reduce the overall size of the PCB to almost half the footprint of the prior iteration, which made fabrication faster and cheaper, and we now have more space to work with in the confined environment of the project. The second iteration of the PCB is pictured below.

## VII. POWER DESIGN

The power circuit is the heart of the hardware design. Without getting power properly distributed, nothing else will work correctly. Below in Figure 41, we have a layout of how our power will be distributed evenly throughout our design as a whole.

When considering modularity, the system, ideally, will be modular to allow for growth. Since there are a large amount of parts that require relative consistency among the voltage inputs, the design has to keep smooth voltages in mind. There will likely be smoothing capacitors at the input to the circuit, to prevent a large voltage spike when the compartment gets connected. With modularity in mind, the schematic for the individual compartments will be identical within each locker.

### *A. USB Charging Circuit*

USB power standards, for charging, has a requirement of 5V along the power rails, and a limit to the current of 2.5A. It's pretty common practice to limit this to at most 2.4A. For ease of testing and limitations, we will limit the current to around 1.5-2.0A this will also limit the max power consumption and not have to worry as much about energy loss due to heat. For the sake of keeping heat on the chip to a minimum and independent consistency across all USB ports, we will be using separate voltage regulators for each USB charging port.

## VIII. OPERATION AND PROCESS FLOW

The Beaglebone Black was flashed with a 2013 distribution of Angstrom Linux and programmed in the C++ language using a remote development environment on Eclipse. The ATMega328 microcontrollers were both programmed in the Arduino language using the Arduino IDE.

### *Sensor System*

The sensor system begins its operation by dedicating itself to an address on the I2C bus of the Beaglebone Black. After completing that, the sensor system immediately starts setting up a loop function for detecting sound and vibration and the loop function samples the analog pins where the vibration sensors and microphones are attached in sets of 100 samples per cycle. Because phones all ring and vibrate with different cadences, we test for a certain hit percentage in a sample cycle before we throw a flag for a positive hit. When we receive that positive hit, we store that data in a register for the

Beaglebone Black to later pull as it rapidly polls all of the devices logged in to the I2C bus. After the Beaglbone Black polls the device, another flag is thrown to wipe the register in preparation for the next possible positive hit.

*Beaglebone Black*

This device handles the majority of all of the inter-device communication and update log tracking for the device. This device can be plugged into the Ethernet network of a building at any connection point and be found with a port scanner. After you find the device with the port scanner, you can set up a connection via SSH and log into the device from any computer on the network. Once you log into the device you can easily run the entire system with a simple run command. When the Beaglebone Black runs, it will immediately start by setting up the I2C communication with every device on the active I2C bus of the Beaglebone Black. After establishing the communication, the Beaglebone Black is set into Master-Receiver mode and sets up a loop to begin rapidly polling every non-empty and non default occupied address on the bus. During this procedure, the Beaglebone Black will be logging the positive hit data from the sensor systems on the bus. When the Beaglebone Black pulls these positive hits, it will immediately log an instance of contact for the specific address and add that instance to the update log showing that a cellular device at that specific address has been contacted. At the same time, a timer will start. If that same address is contacted a certain number of times within that time frame, the Beaglebone Black will then throw another flag and change the instance added to the update log into one denoting a possible emergency so that the user can either contact the phones owner or the owner can go retrieve their device at their own discretion. The Beaglebone Black can also stop the main functionality of the device simply by logging out of it.

## IX. CONCLUSION

The SPLINT systems core system is the collection of several relatively simple subsystems. The proper integration of these subsystems yields us a functional product that can help alleviate a growing problem in the workplace. Considerations have been made to reduce power consumption, increase portability and modularity, as well as increase accessibility all in the attempt to make our product one that employer will look towards for their employee management needs. The design and implementation workload has been evenly distributed between all four students who worked on this project. The SPLINT system has been designed to improve workplace environments all around.

## X. ACKNOWLEDGEMENTS

The SPLINT team would like to thank Kaman Fuzing and Precision Products for helping inspire the creation of the SPLINT system. They would also like to acknowledge the UCF senior design laboratory and David R. Douglas for providing our group with the facilities and several resources which aided in the development and testing of our project. The team would also like to acknowledge Derek Malloy for his published book and online resources the group used to help them develop on the Beaglebone Black.

## XI. BIOGRAPHIES

Alexander Masterson is currently a senior at the University of Central Florida. He is currently seeking a Double Major with a Bachelors degree in Computer Engineering and a Bachelors in Electrical Engineering. He will graduate in December of 2017. He is pursuing a career in system automation controls. He is currently seeking full time employment for when he graduates. He plans to work in the field for a while and then may pursue a Masters Degree in the future.

Evelin Santana is currently a senior at the University of Central Florida. She is currently completing a Bachelors degree in Electrical Engineering and will graduate in December of 2017. She is currently seeking to accept a position for Controls Systems Engineer with the Disney Imagineering Team when she graduates.

Ian Flemings is currently a senior at the University of Central Florida and will receive Bachelor's of Science in Electrical Engineering. He has held a engineering internship position with AECOM for the last 2 years and in that time he has taken and passed the Fundamentals of Engineering exam for electrical and computer engineering, and currently is planning to move to Grand Rapids, Michigan to take up an open Electrical Engineer I position with AECOM.

Nicholas Lucas is currently a senior in Electrical Engineering at UCF with a minor in mathematics. He passed the fundamentals of engineering exam and his E.I. status is pending graduation. He wishes to go into the industry in either RF, networking or distributed computing, and plans to continue to pursue a masters in Electrical Engineering or Computer Science.