Deliverable 6a: Final Report

Back Breakers
Keven Fuentes : kfuente@clemson.edu
Joe Hilbert: jhilber@clemson.edu
Wade King: jwk@clemson.edu
Buffet URL: buffet@buffet.cs.clemson.edu:kfuente/BackBreakers-CPSC4820

Project Summary

The goal of our system as of now is a small, accurate, and fast RFID scanner that maintains a database of known cards that are used to track items inside of a backpack. When items are removed or added to the bag they must be scanned by the RFID reader in order to maintain an accurate record of the items. Putting the reader into edit mode allows new and current items to be both added to and removed from the record. A short press of the button puts the reader in check mode which outputs a tune based on if items are missing from the bag.

Project Description and Implementation

- **Changes**

From our initial plans, we only had a couple of changes. One of the main changes came from the trouble that we had building an RFID antenna for our reader. We wanted to have a larger field that would be able to encompass the main compartment of the backpack, and the scanning of objects would be done simply by placing the objects into the backpack. Unfortunately, the antennas that we created did not work properly, so we had to change the approach to be a shorter range scanner that would require the user to explicitly move the items in the area of the scanner in order for scanning to actually occur. We looked for longer range RFID readers with a built-in antenna, however, the cost of such a reader was too high and therefore we decided on a cheaper reader that could still prove the concept that we wanted to achieve. The other change that we had was our power source. We initially decided to use one rechargeable battery pack to power both the microcontroller and the RFID reader, but since we wanted the most out of the RFID reader without the antenna, we needed to make sure that we could supply the maximum current that the RFID reader would draw; therefore we decided to power the RFID reader with a 9v and step it down to 5v instead of powering it from the launchpad. The speaker system and the microcontroller could then be powered by the battery pack.

- **Software**

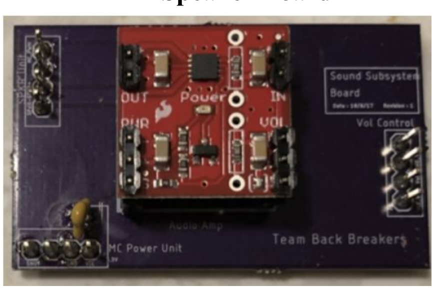
Our software was split into three subsystems or modules: the RFID subsystem, the speaker subsystem, and the microcontroller. This modular design gave us great flexibility in our code development. If we could split the code into three subsystems, then the code could be developed and tested separately, allowing use to work on different parts of the code at once and develop one part of the code without needing to rely on the other portions to work properly. This is especially true if one of the hardware components wasn't working or could not be figured out, allowing us to still make progress on other parts of the project without being held back by one specific part. Both the RFID subsystem and the speaker subsystem then had their own classes with public methods that would allow the other subsystems to interface with it and private members that maintained state information for the subsystem, which would allow the each subsystem of the project to be reusable or adaptable to new projects or later iterations of this same project with minimal changes made to base code. Finally, we had one .ino file that would include these libraries and would be uploaded to the microcontroller. The setup in the microcontroller code was fairly simple and just consisted of initializing the pins that would be used in on the microcontroller, such as the leds and reset pins. Additionally, the reader and the speaker classes were initialized for the device startup, and the reader was reset to start scanning any new incoming tags. The initial RFID code starts with five predefined cards that are registered with the device. The reader, however, has two modes which are defined in the RFID library and are simply toggled by the microcontroller when the button is pressed for a designated time of around three seconds. In order to give the user state information on the mode that they are in, each switch of mode is given a slight tone from the speaker to signify a change, and similar to when pairing a bluetooth device, a red light blinks when it is in "Edit" or "Pairing" Mode, giving the user clear indication of state change and what state the device is currently in. The "Edit" mode was our first stretch goal, and we saw that we could reach this new mode before completing the device, and we had designed the code so that this sort of new mode could be added without large changes to our initial code. Instead, changes could be made in the RFID reader subsystem and its methods and state information members. Pairing mode is used to register a new tag to the system, giving a positive tone signifying a registering of a new tagged object and giving a negative tone to signify that the tag was unregistered when scanned in, allowing for a total of ten objects to be registered to the device. The scanning mode is similar in that when a tag is scanned into the backpack it gives a positive tone to signify that it has been checked in and a negative tone if it has been checked out. This is all maintained by three arrays, one that holds all tags that the system has registered or is aware of, an unchecked array that maintains the indices of unchecked items and a checked array that maintains the indices of the checked in items. Therefore, by checking that the checked array is full it is clear whether a tag is missing or not. Lastly, how a user checks if they are missing something from their backpack they press the button instead of a long press and hold, which would change the reader state. If the button is pressed, the checked list will be checked for completeness and if it is complete, it will delegate to the speaker subsystem to play the positive tone in its library, otherwise it will delegate to the speaker to play the negative tone, signifying to the user that something is missing from the backpack.

- **Hardware:**

    Our project consists of two custom boards, one speaker, one launchpad, and two batteries. When designing our hardware we decided to split it into two main groups, the RFID module and the launchpad. The RFID module controls reading and sending data to the launchpad while the launchpad does all of the logic and signals the speaker. The RFID module drew a significantly higher current than the launchpad could provide, so we used a 9V battery with a step down voltage converter in order to ensure our reader had enough power at all times. In order to power the launchpad, we used a simple 5V phone battery pack with 10000 mAh. This allowed us to easily recharge the battery pack and have it ready at all times. The two main components of the system, the reader and the launchpad, needed to be connected using a logic level converter due to the voltage differences of both system groups. This allowed the reader to send the TTL serial communication signals in a safe manner to the launchpad RX pin. For this project we also designed two custom boards, one for the RFID module and one for the speaker amp. This allowed us to have a solid base for everything related to the RFID module and gave the reader better physical stability in the bag. Breaking the speaker amp into its own board allowed us to wire the speaker through the bag and out on top of one of the straps so that it was as close as possible to the users ears.

- **Board Pictures**

Speaker Board



RFID Board

- **Challenges**

    We ran into quite a few challenges while working on this project. As with all projects our vision has evolved during the development process to be more realistic. By far our biggest challenge was initially trying to design our own RFID antenna. When we started the development process none of us knew anything about RFID. We thought we could make our own antenna, but as the project progress, we learned this was out of our scope of knowledge and every antenna that we created failed to work properly. To fix the problem, we purchased a chip with a built-in antenna and adapted our design slightly to accommodate for the shorter range of the new RFID reader.

Another issue we had was powering our devices. Due to our limited knowledge of electronics we decided to use a 5V battery to power the launchpad and power all the other peripherals from the launchpad. This turned out to not have enough current to power everything and we had debugging issues. We were able to solve these issues by introducing a 9V battery and stepping down the voltage to 5v to the RFID module. We initially also had issues with the speaker, but once we implemented the 9V our speaker worked perfectly.

Resources

Parts (Datasheets included on website):

- ID20LA RFID Reader
  - Website: https://www.sparkfun.com/products/11828
  - This was the product page, including datasheets and tutorials, for our ID20LA RFID Reader.
- Audio Amplifier
  - Website: http://www.ti.com/product/TPA2005D1
  - This was the Texas Instruments product page, including datasheets and other helpful information, for our audio amplifier.
- Logic Level Converter
  - Website: https://www.sparkfun.com/products/12009
  - This was the Sparkfun product page, including datasheets, schematics, and tutorials, for our logic level converter.

Software and Hardware Tutorials:

- RFID Reading (example code + hardware tutorial)
  - http://bildr.org/2011/02/rfid-arduino
  - This website gave helpful information for hooking up our RFID Reader to our system and getting software working with it. We adapted much of the code example included on this site into our project's software to allow it to read RFID tags and store information about the tag during use.
- Playing Sounds
  - https://www.arduino.cc/en/Tutorial/toneMelody
  - This website was used to get us started on playing sounds on our speaker subsystem, especially for playing tones that provided feedback to the user.

From the code provided on this site, we were able to create other tones to implement into our project.

- Logic Level Converter
  - https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide
  - This tutorial was helpful in allowing us to implement our logic level converter into the project to allow our MSP430 microcontroller to interface with our RFID Reader, as their voltage ranges were incompatible.

Class Feedback

- What was the most challenging part of the class?
  - The most challenging parts of the class involved learning how to combine hardware components into a working system. As a group with only computer science students, we had no substantial issues with implementing the software to work in the whole system. In higher level programming languages, functionality is often self-explanatory. However, for hardware, one must use datasheets and other resources to understand the intricacies of individual parts in order to avoid malfunctions. Even worse, malfunctions can often permanently damage the system, unlike in software that includes protections against such permanent issues.
- What would you like to see added to the class in future semesters?
  - Regarding lecture material, we would very much like to see sessions which focus more on how to debug a system before hardware implementation begins. During hardware-implementation, if a major issue is discovered, this could be be fairly harmful to the group's progress, especially monetarily. Sessions such as these would focus on how to reverse engineer schematics and documentation to discover issues in hardware implementations. Outside of lecture material, we hope that the new makerspace in Watt can be improved upon and better implemented into the course, avoiding over-reliance on OSH Park for boards.
- General Feedback?
  - Overall, this class was very enjoyable and it helped provide a good introduction into embedded systems and hardware implementations for students more suited to working with software. There are few criticisms to apply to this course. The most prominent criticism is that some lectures relied too much on hardware which had to be successfully soldered in order to participate. With the low availability of such hardware (accelerometers, etc.), it was difficult to understand how certain information was relevant.