TEXTBOOK DETECTION SYSTEM WITH RADIO-FREQUENCY IDENTIFICATION

By

Xiaohao Wang

Xiaosheng Wu

Zhao Weng

Final Report for ECE 445, Senior Design, Spring 2017

TA: Jose Sanchez Vicarte

May 2017

Project No. 7

Abstract

The purpose of the project is to assist students or faculty members to secure their textbooks. It is common that a student may forget to bring books to his or her classes. We therefore intended to provide a solution by constructing an electronic system that keeps track of the textbook information in the backpack. The project involves the use of microcontroller, sensor, radio-frequency identification (RFID) and Android Development.

With the help of RFID and Bluetooth implementation, the project works well and it can display accurate information on an android application. From this project, we hope to build a tracking system for college students.

Contents

1. Introduction

Our goal is to make a device that can help keep track of the information about textbooks in a student's backpack based on his/her daily schedule. In order to make a user interface application, we developed an Android application to display the textbook information on the phone. We attached an ID tag to each book and used radio-frequency identification (RFID) to keep track of the textbooks.

1.1. Objective

- The RFID antenna should detect tag IDs and transmit the data to the Android application
- The protection circuit can prevent the battery voltage from dropping too low

2. Design

2.1. Block Diagram

The project involves the use of the microcontroller, sensor, RFID, and Android development. We will construct an electronic system that can keep track of the textbook information in the backpack.

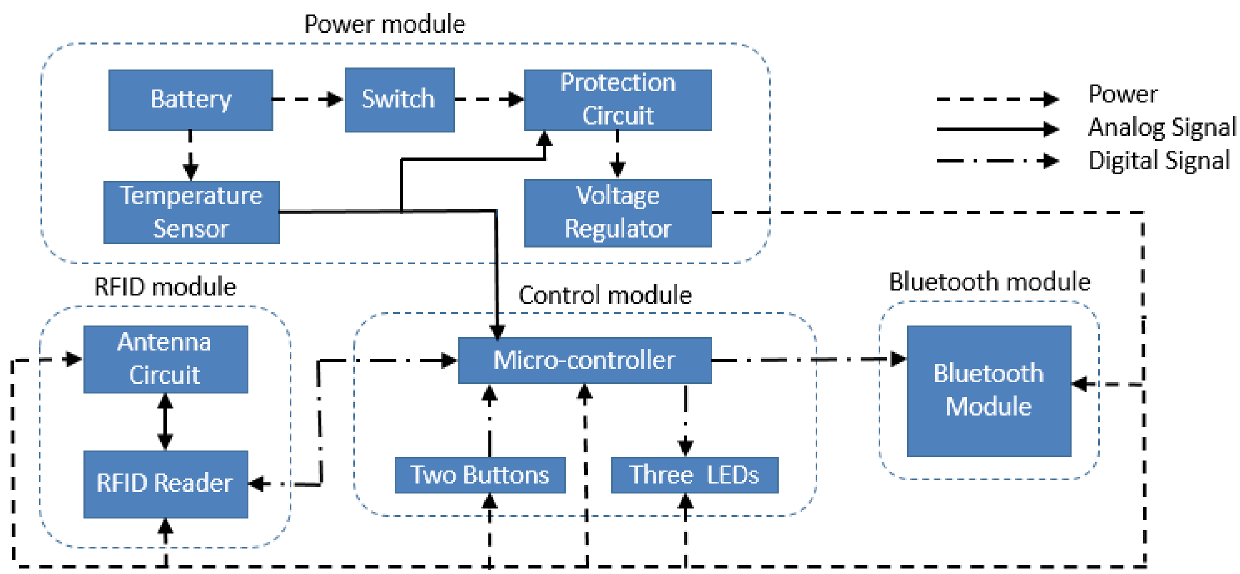

Figure 1. Block Diagram

There are four different modules: the power module, the control module, the Bluetooth module, and the RFID module. In the block diagram, the dashed line represents how battery delivers power to all the modules. The protection circuit protected the circuit by ensuring that the voltage stays within the safety range. After the power goes through the voltage regulator, the power module converts the battery voltage from 3.7 volts to 3.3 volts. The control module communicates with RFID module and Bluetooth module via SPA and UART interface. Once the RFID module detects the ID tags, it passes the IDs to the microcontroller, and the controller transfers the data to the Bluetooth module. Finally, the android application updates and displays the correct textbook checklist for the user.

2.2. Physical Design

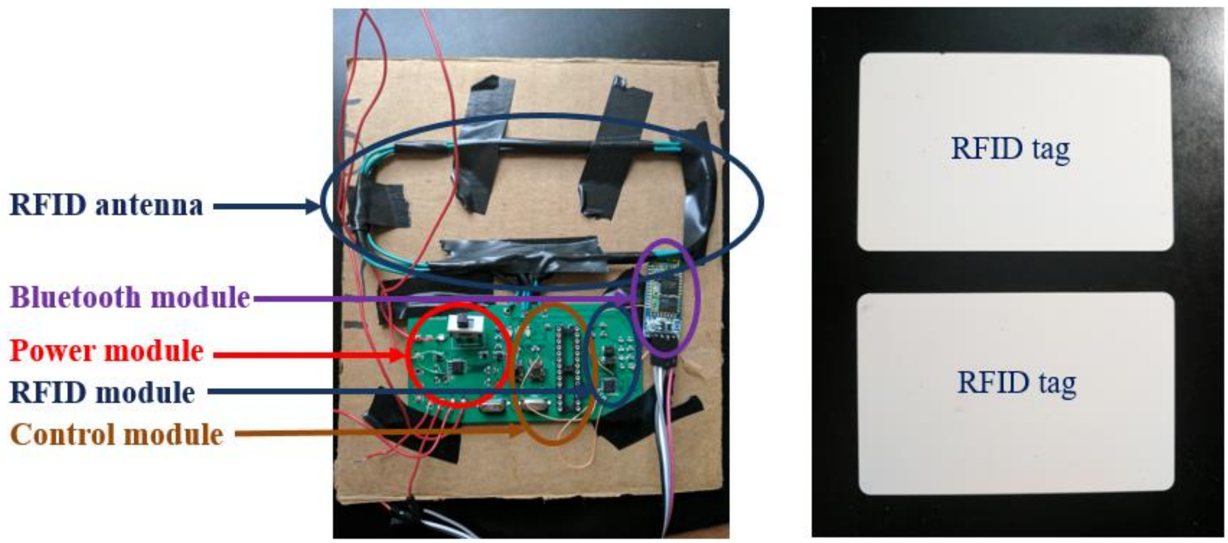

Figure 2. Physical Design(left) and RFID Tags(right)

The image on the left side shows the physical design of our project. Our RFID antenna is at the top of the PCB, and it receive data information from the RFID tags. We attached the PCB to the antenna and it consists of four different modules. On the right side, the image shows the RFID tags we used for the project. Each book has one tag attached to the cover page. We used a cardboard and tapes to fix the dimension of the antenna to maximize the detection area.

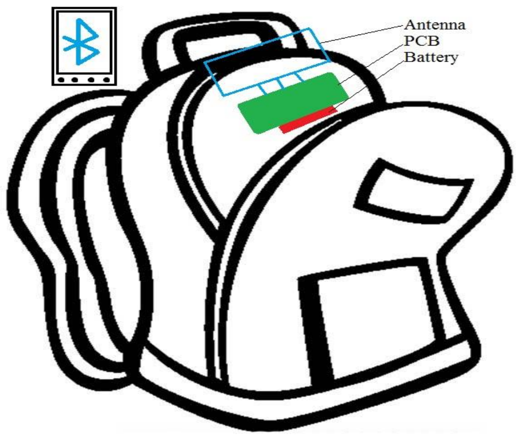

Figure 3. Physical Design Sketch[1]

We planned to install the project at the top of the backpack so that the antenna can easily detect the textbook IDs when the user adds or removes books from the bag.

2.3. Power Module

The power module provides constant 3.3V power to the rest of the circuit. It also ensures the safe operation of the battery. The power module consists of a Lithium Ion battery, a charger, a temperature sensor, the protection circuit, and a voltage regulator.

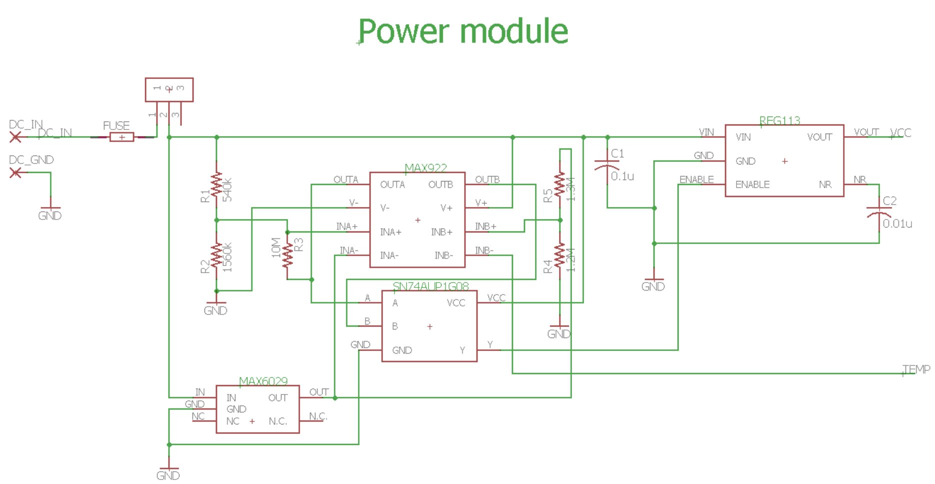

Figure 4. Power module schematics

2.3.1. Battery and charger

For our project, we wanted the battery to be able to last at least 8 hours. We estimated the peak current consumption to be about 270mA and the normal current consumption to be about

160mA. So, we chose to use a 2500mAH battery from Adafruit to ensure we had enough battery life. We were also using the charger from Adafruit to ensure the safety of the charging process.

2.3.2. Temperature sensor

The temperature sensor monitored the temperature of the battery so the protection circuit can cut the power to the rest of the circuit when battery temperature is too high. We chose to use the TMP36 analog temperature sensor for our project. It can provide accurate measurement over the -40°C to +125°C temperature range which is sufficient for our use and has very low power consumption.

2.3.3. Protection circuit

Due to the inherent danger of lithium battery, we added a protection circuit in the design. It is dangerous to overcharge or overdrain the battery. At first, we wanted to add both high voltage protection and low voltage protection, but since we were using the charger from the battery manufacturer, we only needed to worry about the overdrain problem. Adding high voltage protection requires another comparator, which is pointless, since it is safe to drain an overcharged battery. As a result, we decided not to implement the high voltage protection. There is a undervoltage lockout (UVLO) circuit in our protection circuit that disables the voltage regulator when the battery voltage drops below 3.3V and only enables the regulator after the battery voltage rises above 3.5V. Also, the protection circuit will be able to shut down the rest of the circuit when the temperature is above 70°C so the battery doesn't overheat.

We used the application note from MAX922 [2] to design the UVLO circuit. We chose R_3 to be 10MΩ and set the boundaries to be between 3.3V and 3.5V, and V_{HB}=0.2V and V_{THR}=3.5V.

Calculating the value of R_1:

$$R_1 = R_3 \times \frac{V_{HB}}{V+} = 10M\Omega \times \frac{0.2V}{3.7V} \approx 540k\Omega \tag{1}$$

Calculating the value for R_2:

$$R_2 = \frac{1}{\left[\left(\frac{V_{THR}}{V_{ref} \times R_1}\right) - \frac{1}{R_1} - \frac{1}{R_3}\right]} \approx 1560k\Omega \tag{2}$$

For the temperature protection circuit, we only needed to compare the temperature sensor input with the corresponding voltage when temperature is 70°C, which is about 1.23V during testing.

Thus, we chose to use two 1.2MΩ resistors to divide the 2.5V reference voltage to be approximately 1.25V.

2.3.4. Voltage regulator

The function of the voltage regulator is to provide a stable 3.3V to the rest of the circuit. It should also be able to cut the power to the rest of the circuit when enable signal is LOW. Based on our current consumption, we chose to use REG113 low dropout regulator. It can handle 400mA current draw and only has 250mV dropout voltage.

2.4. Control Module

The control module consists of one microcontroller, two buttons, three LEDs and the setup circuit for the microcontroller chip.

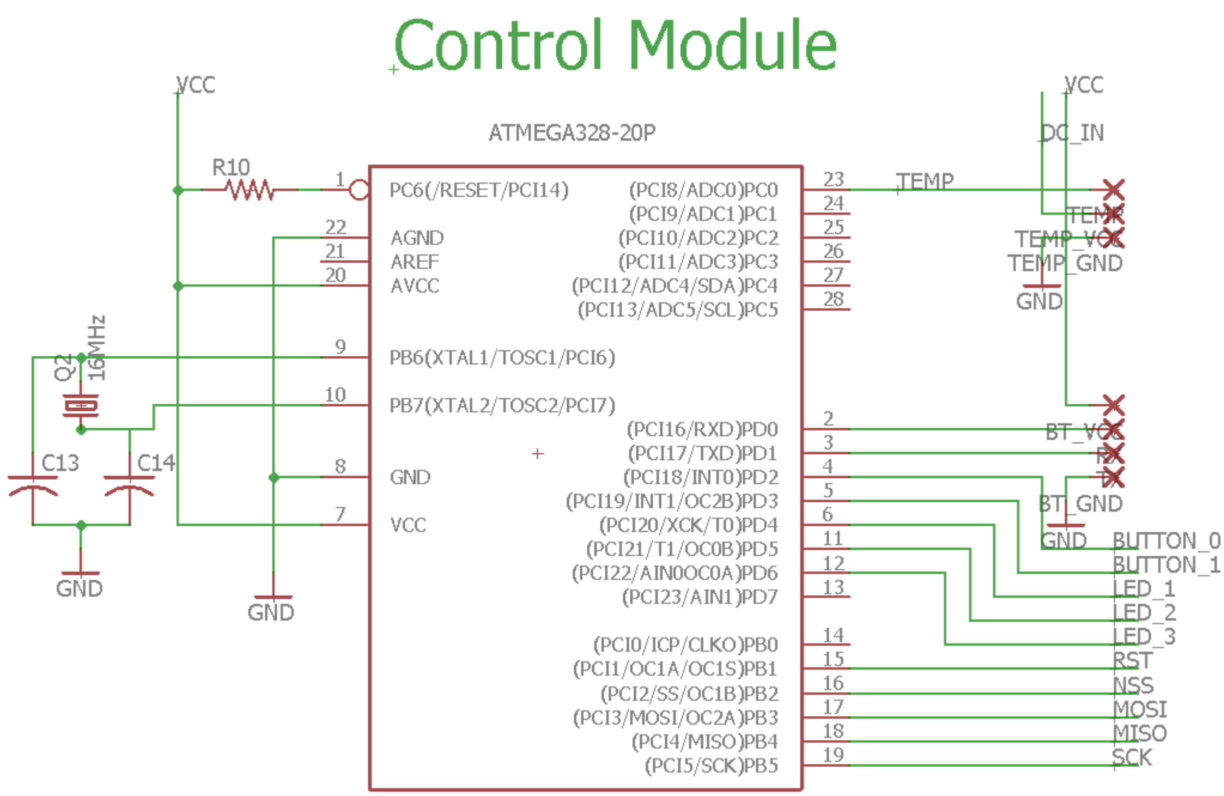

Figure 5. Control Module Schematics

2.4.1. Microcontroller

The programmed ATmega328p chip initialized the connection between different components once the power module provides 3.3 volt for the chip. The controller handles both SPI and

UART communication between the RFID and Bluetooth module for data transmission [3]. It can also modify the register values on the RFID chip to turn on or off the antenna circuit. We used three LEDs to indicate if the controller and the antenna is on or off.

2.4.2. LEDs and Buttons

There are three LEDs to indicate the status of the system. The blue LED is on when the microcontroller is working. The user can turn on the green LED by pressing the button on the left. The button on the right can turn off the green LED. The orange LED glows if the battery temperature exceeds the 70°C.

2.5. RFID Module

The basic function of RFID module is to generate the magnetic field to communicate with the tags and transmit the tag data to our control module. We chose to use high frequency (HF) chip because it provides a reasonable reading distance and is relatively cheap. We wanted the detection range to be around 3 cm to 5 cm for a passive tag. With proper tuning, HF chips should be able to achieve that range.

2.5.1. RFID reader

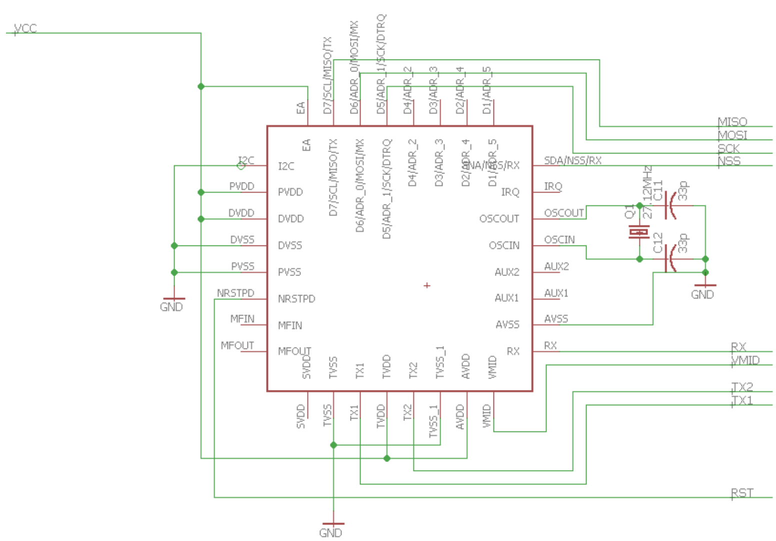

Figure 6. RFID reader circuit schematics

The RFID reader is used to receive tag IDs from the antenna circuit and transmit the data to the control module. We chose to use MFRC522 reader chip [4]. It should be able to transmit data received from antenna via SPI interface to the microcontroller. We also wanted it to be able to turn off antenna circuit so that we can save some battery life when we are not using RFID module.

2.5.2. RFID antenna circuit

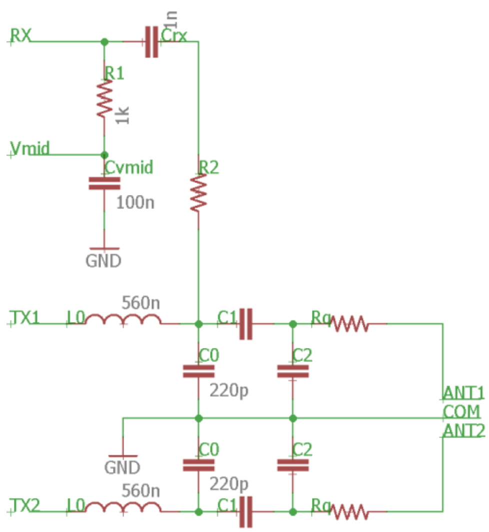

Figure 7. RFID antenna circuit

We followed AN1445 application note [5] to design the antenna circuit. We used the recommended values for L_1, L_2, R_4, C_3, C_4, C_9 and C_{10}. The rest was determined based on the actual antenna we made. We made two-turn 15cm wide and 5cm high antennas with 22 AWG wires. We chose to use two-turn antennas because three-turn antenna inductance is too high. Detailed antenna measurements are in Appendix B. We measured the parameters of the antennas with a network analyzer. From the experiment, the inductance of the antenna is about 3.88μH, the resistance is about 5.5Ω and the capacitance is about 1pF. We can then start to calculate the approximated values for the antenna circuit as a starting point for our tuning process.

First, we need to calculate the Q factor for the antenna. The Q factor should be less than 35. If the Q factor is too high, we need to add two resistor R_Q.

$$Q_{orig} = \frac{\omega \cdot L_a}{R_a} \approx 60.1 \tag{3}$$

$$R_Q = 0.5\left(\frac{\omega \cdot L_a}{35} - R_a\right) \approx 2\Omega \tag{4}$$

We chose to use 3Ω for R_Q. Then we can calculate the corresponding R_{pa} of the parallel representation of the antenna:

$$R_{pa} \approx \frac{(\omega \cdot L_a)^2}{R_a + 2R_Q} \approx 9502\Omega \tag{5}$$

The recommended value for R_{match} is 50Ω. Using that, we can calculate R_{tr} and X_{tr}:

$$R_{tr} = \frac{R_{match}}{(1-\omega^2 L_0 C_0)^2 + \left(\omega \frac{R_{match}}{2} C_0\right)^2} \approx 217\Omega \tag{6}$$

$$X_{tr} = 2\omega \frac{L_0\left(1-\omega^2 L_0 C_0\right) - \frac{R_{match}{}^2}{4} C_0}{(1-\omega^2 L_0 C_0)^2 + \left(\omega \frac{R_{match}}{2} C_0\right)^2} \approx -58\Omega \tag{7}$$

Calculating C_1 and C_2:

$$C_1 \approx \frac{1}{\omega\left(\sqrt{\frac{R_{tr} \cdot R_{pa}}{4}} + \frac{X_{tr}}{2}\right)} \approx 17pF \tag{8}$$

$$C_2 \approx \frac{1}{\omega^2 \frac{L_{pa}}{2}} - \frac{1}{\omega\sqrt{\frac{R_{tr} \cdot R_{pa}}{4}}} - 2C_{pa} \approx 55pF \tag{9}$$

We can use these values as starting point for our tuning process. After tuning process, we chose C_1 to be 13pF and C_2 to be 39pF. We then measured the peak-to-peak voltage across C_0 to choose the resistor R_2 in the receiver circuit. Here is the result:

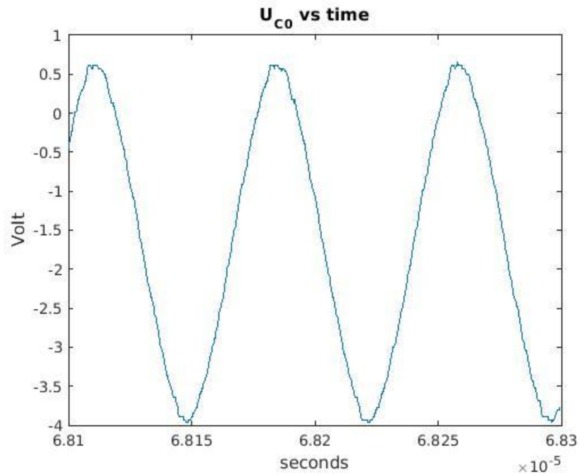

Figure 8. UC0 vs time

The peak-to-peak voltage is about 4.6V.

$$R_2 = R_1 \cdot \left(\frac{U_{C0}}{U_{RX}} - 1\right) = 3.6k\Omega \qquad (10)$$

We chose R_2 to be 4k**Ω** to be safe.

2.6. Bluetooth Module

We wanted to use a simple JY-CMU HC-06 Bluetooth module to ensure stable connection between the phone and our device. JY-MCU was easy to use and it can provide a stable connection between the module and the phone. The module could tolerate a range of around 20 Hz around 9600 Hz [6].

3. Design Verification

3.1. Power Module Verification

For our power module, we did not test the battery and charger, since we did not design any of that. We tested all the other components with stable power supply instead of battery.

3.1.1. Protection circuit verification

The enable output should be 0V when input voltage drops below 3.3V and should stay at 0V until the input voltage rises to 3.5V. We verified this voltage value on the breadboard with the oscilloscope and the function generator. The function generator would simulate battery voltage with a 2V peak-to-peak sine wave with 3.7V offset. We then used oscilloscope to probe the enable output. In Figure 9, the enable signal dropped to 0V when input was below 3.3V and stayed at 0V until input rose to 3.5V.

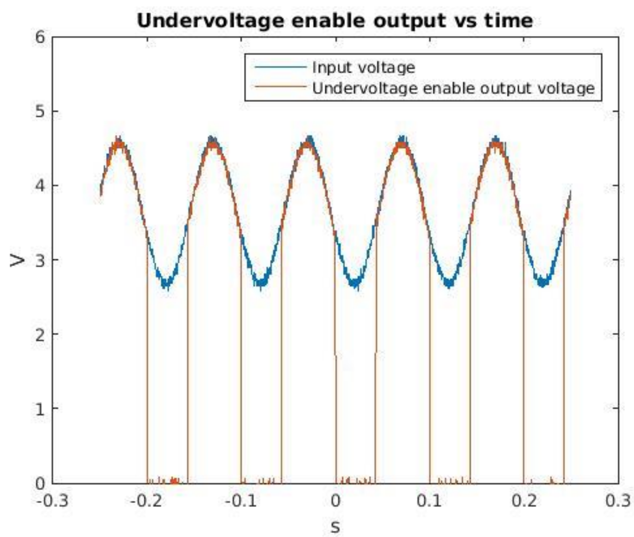

Figure 9. UVLO enable output

On our PCB, since we could not get the ideal resistor values, the results were higher than the results on breadboard. The enable output dropped to 0V when input was below 3.39V and stayed at 0V until input rose to about 3.54V. We also tested temperature sensor. The enable signal dropped to 0V when temperature sensor input was above 1.23V as we wanted.

However, the current consumption was much higher than we expected. The peak current could be 50mA when enable signal is LOW. The voltage regulator output was still 0V and all other modules were turned off. We thought the large current may be caused by the protection circuit or the leakage current of other modules.

3.1.2. Voltage regulator verification

The voltage regulator should provide a stable 3.3V power to the rest of the circuit. Here we tested it using breadboard. The input result was the same sine wave as the result from section 3.1.1. The output was 3.3V only when the enable signal from section 3.1.1 is HIGH.

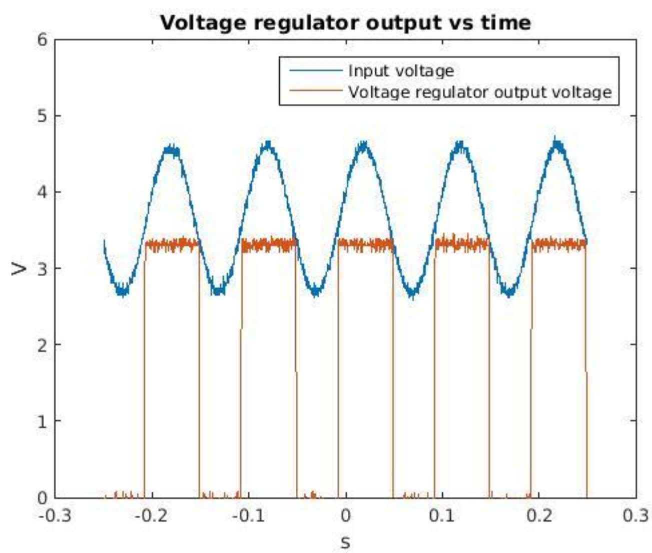

Figure 10. Voltage regulator output

We noticed that the voltage output of the regulator started to fluctuate near the 3.3-volt threshold on our PCB. Due to the varying current consumption, the output was not stable. When the Bluetooth module was in pairing mode, we observed that current consumption varies from 160mA to 200mA. If we had more time, we will use separate regulators for different modules to avoid this problem.

3.2. Control Module Verification

For the control module, we wanted the microcontroller to establish the connection between the Bluetooth module and RFID module. We first connected our RFID module and microcontroller, and then we connected the RX and TX pins with a PC terminal via USB. We can receive data from the RFID module. After we changed the USB cable to our Bluetooth module and applied the Android application, we can verify the tag IDs on a mobile device.

During the modular testing, we noticed that the microcontroller restarted randomly if the input voltage is near the 3.3-volt threshold. To fix the problem, we can use different regulators for different modules.

3.3. RFID Module Verification

For our RFID module, we first wanted to verify that it can receive tag IDs from the antenna and send the data to the control module. By using Arduino serial monitor, we disconnected the Bluetooth module on the PCB and connected the UART interface directly to the Arduino. The tag IDs showed up on our serial monitor when we placed the tag near the antenna which proved that RFID module was working correctly. The RFID reader should also be able to turn off antenna circuit. We verified the reader by probing the voltage at TX1 with low capacitance probe and the oscilloscope. The voltage at TX1 should be 0V when the antenna was turned off. The voltage at TX1 indeed dropped to 0V when we turned off the antenna through our microcontroller. For the 3cm to 5cm range requirement, we verified that by slowly moving the RFID tags toward the antenna from 10cm away, and the detection distance was about 3.5cm to 4.5cm away.

3.4. Bluetooth Module Verification

After we received our Bluetooth module, we followed a tutorial online to test all the functionalities with a PC terminal. The Bluetooth module is able to receive and transmit data from the microcontroller to a mobile device.

4. Costs

4.1 Parts

Table 1 Parts Cost

Part	Quantity	Unit Cost ($)	Actual Cost ($)
MAX922	4	3.320	13.28
MAX6029	4	6.110	24.44
MFRC522	8	4.630	37.04
REG113	8	2.870	22.96
SN74LVC1G08DCKR	4	0.820	3.28
TMP36	4	1.500	6.00
16MHz Crystal	4	0.390	1.56
27.12MHz Crystal	2	0.400	0.80
CS12ANW03 Slide Switch	3	2.000	6.00
1206SFP100F 1A Fuse	4	0.840	3.36
BAT60AE6327HTSA1 Diode	4	0.410	1.64
APT3216LVBC/D Blue LED	10	0.313	3.13
HSMD-C150 Orange LED	10	0.288	2.88
LTST-C150GKT Green LED	10	0.287	2.87
PDIP-28 Socket	5	1.680	8.40
Various Resistors	100	0.100	10.00
Various Capacitors	200	0.050	10.00
Lithium Ion Battery	1	9.950	9.95
Lithium Ion Battery Charger	1	12.50	12.50
SMIT BreakOut PCB for SOIC-8	2	2.950	5.90
SMT BreakOut PCB for 32-QFN or 32-TQFP	1	5.950	5.95
SMT BreakOut PCB Set For SOT-23	1	4.950	4.95
Atmega328 Microcontroller	1	2.990	2.99
JY-MCU HC-06 Bluetooth Module	3	7.390	22.17
0.56μH Inductor	10	0.271	2.71
Total			**224.76**

4.2 Labor

Costs: our development costs were $30/hour, 10 hours/week for three people. We worked for 12 weeks. The cost was:

$$3\times\frac{\$30}{\text{hour}}\times 10\,\frac{\text{hour}}{\text{week}}\times 12\text{ weeks}\times 2.5\ = \$27000 \tag{11}$$

Total cost: $27224.76

5. Conclusion

5.1 Accomplishments

Based on the results from the final demo, our customized RFID antenna and module can receive correct textbook information from the tag IDs. The microcontroller can establish the SPI and UART interface between RFID module and Bluetooth module. Our android application performs well and can display the textbook information for the user.

5.2 Uncertainties

One problem is that the microcontroller restarts randomly when the battery voltage input is close to the protection circuit threshold. We measured the output at the voltage regulator and observed that the voltage is not stable. It is possible that the unstable voltage caused the microcontroller to restart. We could use multiple regulators for different modules.

We noticed that the current was unnaturally high when the voltage regulator was disabled. The rest of the circuit should have no power, but the current consumption was 50mA. One possible source of that current comes from the protection circuit. The other possible source is the leakage current of the other modules.

5.3 Ethical considerations

According to IEEE Ethics, #1: "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment." [7], we, as future electrical engineers, were responsible for designing a safe and reliable application.

Because we used Lithium-ion battery for this project, it was very important to keep the current and voltage within the limitation to prevent explosion [8]. In order to prevent potential hazards, we had a voltage regulator module and a protection circuit. The goal was to guarantee that the overall circuit can operate with low current (below 1 A) and low voltage (about 3.3 V). Furthermore, users are not supposed to expose the power source to water or direct sunshine, which might cause safety issues on our battery. In addition, even if our final product might be able to automatically turn off the power by itself based on different conditions, it was also the user's' responsibilities to check and turn off the power if this application is not in use.

Since the purpose behind our project was to help students with their study, our application was also responsible for sending information for education purpose. According to IEEE Code of Ethics, #5:"to improve the understanding of technology; its appropriate application, and potential consequences; "[7]. We hoped our project could provide students with basic knowledge in engineering design. It was important that our design can influence other engineers in our procession. We decide to make our code open source to "to assist colleagues and co-workers in their professional development and to support them in following this code of ethics." [7]

Although, with the help of internet, it was easy to search for our project online, we hoped people who were passionate about rebuilding or improving our project could respect our property and contribution, based on IEEE Code of Ethics #7:"to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others" [7]. If other project teams were going to use the design or code from our project, the techniques used in this application should be correctly cited.

5.4 Future work

We hope to use force sensor to detect when the user adds or remove books from the bag so that our project can automatically turn on or off the RFID antenna. Additionally, initializing tag IDs at the beginning of the application can improve the user interface application. We also hope to add separate regulators to make our circuit more stable. The final goal is to make our project fully automatic.

References

[1] ClipartKid.com “Open Backpack ClipArt”, 2016 [Online]. Available: http://www.clipartkid.com/open-backpack- cliparts/

[2] mouser.com, “Ultra Low-Power, Single/Dual-Supply Comparators”, 2009. [Online]. Available: http://www.mouser.com/ds/2/256/MAX921-MAX924-108730.pdf

[3] “Microcontroller: ATmega328P (arduino)” [Online]. Available: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA- 168A-168PA-328-328P_datasheet_Complete.pdf

[4] nxp.com, “MFRC522” [Online]. Available: https://www.nxp.com/documents/data_sheet/MFRC522.pdf

[5] eng,Utah.edu,” Antenna design guide for MFRC52x, PN51x and PN53x”[Online]. Available: http://eng.utah.edu/~mlewis/ref/NFC/AN1445.pdf

[6] www.olimex.com, “Product Data Sheet” [Online]. Available: https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf

[7] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016]

[8] "Design Review For: Advanced Electric Vehicle Battery Charger, ECE 445 Senior Design Project". *090521 courses.ece.illinois.edu*. Archived from the original on 4 May 2013.

Appendix A Requirement and Verification Table

Table 2 System Requirements and Verifications

Requirement	Verification	Points
1. The battery voltage should be 4.1V to 4.25V when fully charged by a USB charger	1. A. Connect the discharged battery to the USB charger, limit the charging current to be 500mA B. Measure the output voltage when the battery is fully charged, ensure the voltage is between 4.1V and 4.25V	0
1. Analog temperature output of the temperature sensor should be accurate to +/- 3 degree Celsius (See tolerance analysis)	1. A. Connect a constant 3.7V power to the sensor B. Connect sensor output to one of the analog pins of the microcontroller C. Start reading temperature data from microcontroller. Also use a thermometer to measure the precise temperature D. Use a hot air gun to slowly heat up the temperature sensor. Use about 200 degrees and low fan E. Make sure the readings from microcontroller is within +/- 3 degree Celsius with respect to thermometer readings	0
1. The enable output should be 0V when input voltage is lower than 3.3V and should stay at 0V until input voltage rise to 3.5V. 2. The enable output should be 0V when temperature is above 70 degrees Celsius 3. The circuit current should be less than 100uA when the input voltage is less than 3.3V	1. A. Use a power source to power the circuit B. Sweep the input voltage from 3.7V to 3.1V, then back to 3.9V C. Enable output should be 0V when input voltage drops below 3.3V D. Undervoltage enable output should stay at 0V until input rises to 3.5V 2. A. Connect a constant 3.7V input to the protection circuit B. Use a voltage source and a voltage divider to simulate temperature sensor output C. Enable signal should be 0V when simulated voltage is above 1.2V 3. A. Use a constant voltage of 3.2V B. Use a multimeter to measure the current drawn from battery C. The current should be less than 100uA	35

1. Supply 3.3V +/- 5% from 3.7V to 4.2V source with ~300mA current draw 2. Output less than 0.05V when enable pin is low	1. A. Use signal generator to generate a sawtooth wave from 3.7V to 4.2V B. Connect the output of the signal generator to the input of the regulator and tie enable to high. Use a 10Ω resistor as load, and measure the output voltage with an oscilloscope C. Make sure the voltage is within 3.135V and 3.465V 2. Use the same setup, connect enable pin to ground and use an oscilloscope to measure the output voltage, the voltage should be smaller than 0.05V	0
1. The LEDs should represent the current status of the device 2. The LEDs must be visible from 1 meter away	1. A. connect LEDs based on Fig.5 and change the status of our device B. Ensure each LED works correctly Turn on the LEDs and ensure all 2. LEDs are visible in one meter	10
1. The microcontroller can receive RFID tags from the reader through SPI and transmit the tag information to the Bluetooth module through UART without interference 2. The microcontroller should be able to handle interrupts from two buttons	1. A. Connect the Bluetooth module and RFID reader to the microcontroller B. Connect the phone to the Bluetooth module and put a tag on the reader C. Ensure the phone can receive the correct data from the tag 2. A. Connect two buttons and a LED to the microcontroller B. Write separate handler code for each interrupt, one to turn on a LED, one to turn LED off. Make sure nothing else can change the LED state C. Push one button should make LED light up, push the other button should turn LED off	25
1. Can receive tag ID from antenna circuit and transmit tag ID to microcontroller via SPI interface 2. Antenna have no power when reader chip is in transmitter power-down	1. A. Connect reader to arduino via SPI and connect reader to antenna circuit. Connect microcontroller to PC via USB B. Power on the entire circuit and place a HF tag about 3-5 cm from antenna C. Should be able to see tag ID with serial monitor, 2. A. Use microcontroller to put reader to transmitter power-down mode by setting a flag B. Use an oscilloscope to measure voltage between pin 10 and pin 11 of the RFID reader chip, it should be 0V	0

1. Antenna size should be at least 15 cm wide and 5 cm high 2. Can read 13.56MHz tags that are placed parallel to antenna 3 to 5 cm away 80% of the time	1. A. Measure the size of antenna coil with a ruler 2. A. Connect reader to arduino via SPI and connect reader to antenna circuit. Connect microcontroller to PC via USB B. Power on the entire circuit, place a 13.56MHz tag about 3 to 5 cm away from antenna and see if the tag is detected C. Repeat B 50 times and the tag should be detected at least 40 times	20
1. The module should be able to transmit data with 9600 baud 80% of the time	1. A. Connect bluetooth module to microcontroller B. Create an Android application and establish the Bluetooth connection between the bluetooth module and the phone. C. Send 10 bytes from the microcontroller through the bluetooth module and receive the data on the phone. Make sure the data match D. Repeat step C for 50 times, data should match for at least 40 times	10

Appendix B Antenna Impedance Comparison

Figure 11. Single two-turn antenna impedance

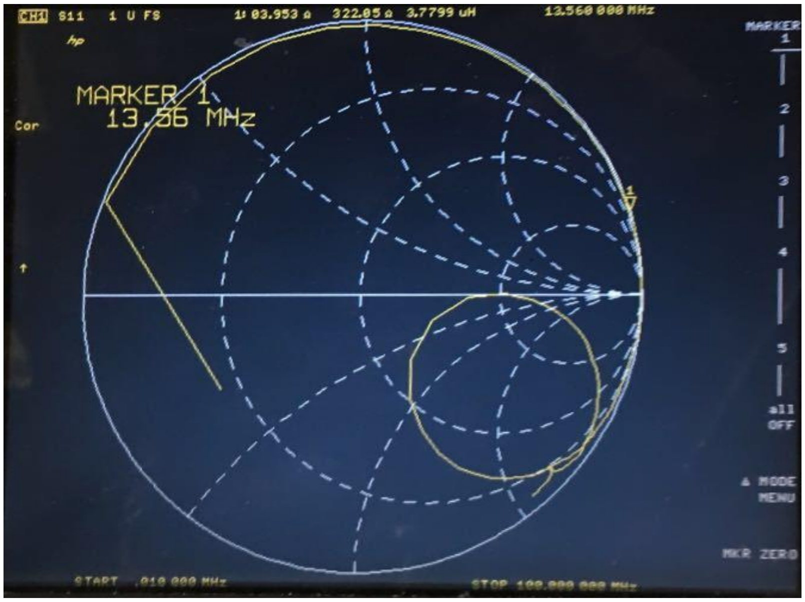

Figure 12. Single three-turn antenna impedance

Note: Recommended combined antenna inductance for two antennas is 0.3µH to 3µH. Here a single three-turn antenna inductance is already 3.95µH. The inductance of two three-turn antennas will be much higher than the recommended value.

Appendix C Additional Graphs for Tuned Antenna Circuit

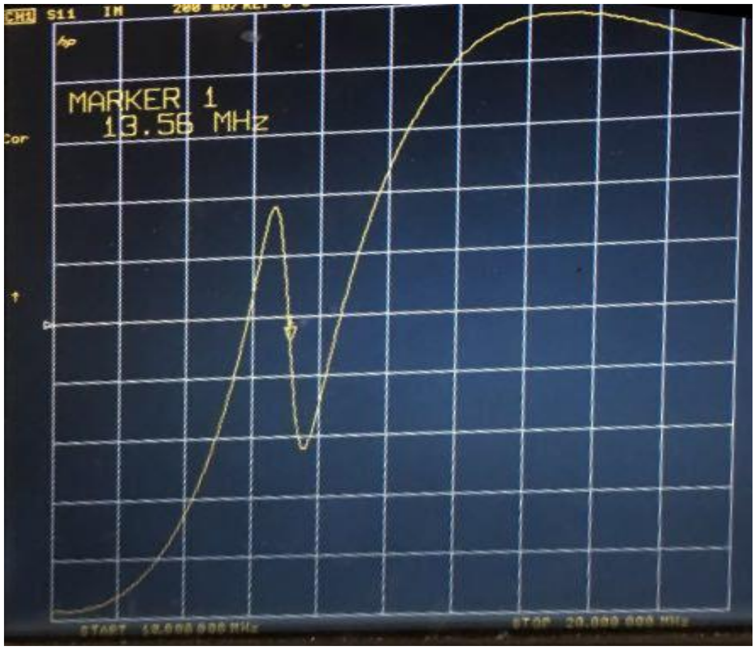

Figure 13. Imaginary part of input impedance of the tuned antenna circuit

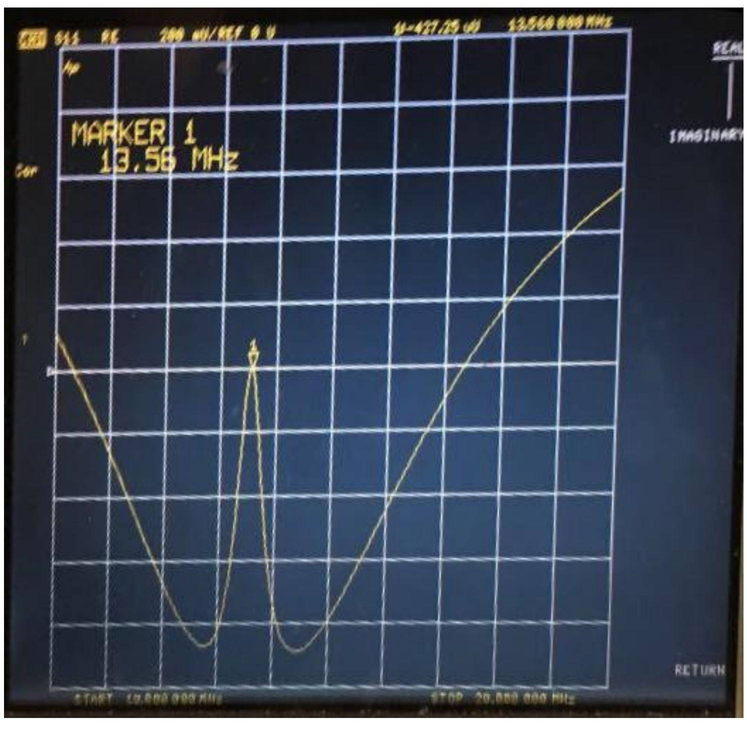

Figure 14. Real part of input impedance of the tuned antenna circuit