RFID Refrigerator
Design Document
Group 73

Jeffrey Lee - jilee9

Yuanhao Wang - ywang272

William Mercado - wmercad2

TA: Yuchen He

3/9/2017

Contents

1 Introduction **4**

1.1 Objective 4

1.2 Background 4

1.3 High Level Requirements 4

2 Design **5**

2.1 Block Diagram 5

2.2 Physical Design 6

2.3 Power Module 7

2.3.1 AC/DC Transformer 8

2.3.2 Buck Converter 8

2.4 RFID Module 11

2.4.1 RFID Tags 11

2.4.2 RFID Reader 11

2.4.3 Antenna 12

2.5 Control Module 13

2.5.1 Microcontroller and Wi-Fi Module 13

2.5.2 Status LEDs 16

2.6 Weight Sensor 17

2.6.1 Load Sensor 17

2.6.2 Load Amplifier 18

2.7 Camera Module 20

2.8 MCU Circuit with Camera and LEDs 22

2.9 Overall Power Consumption 23

2.10 Software Module 23

2.10.1 Data Transmission 23

2.10.2 Data Processing 24

2.10.3 User Interface 25

2.10.4 Inventory 25

2.11 Tolerance Analysis 26

3 Cost and Schedule **29**

3.1 Cost Analysis 29

3.2 Schedule 30

4 Ethics and Safety **31**

5 Citations **33**

1 Introduction

1.1 Objective

According to the NIAAA, Americans spent more than $163 billion on alcohol. Indeed, for many college students, beer is an essential part of the college experience. However, living in an apartment with limited fridge space can cause confusion as to who bought what, and may result in some unscrupulous individuals stealing from others. We seek to solve this problem by creating an RFID tagging system so users can keep track of their inventory.

The goal of this project is to be able to alert users when their beer has been removed from the fridge by pushing an alert to their phone. We hope to use modern technology to provide users with peace of mind when using shared space to store their beer. The project will consist of a camera, an RFID reader and tags, and a microcontroller. The goal of our project is to provide a simple inventory system that can be easily fitted to most fridges and provide real time tracking of user inventory.

1.2 Background

Beer is the most popular alcohol in America, commanding 47.7% of all store alcohol purchases in 2012. The average college student spends around $42 per month on alcohol, and male college students report averaging 9 drinks per week. When living with roommates, situations may arise when beer is accidentally or purposely stole from one another. It may be awkward for roommates to confront one another without solid proof, so we seek to rectify that by providing a system in which users will exactly when their beer has been removed.

Currently, a company called Terso Solutions, Inc. makes RFID refrigerators, but they are not for commercial use [3]. Instead, it is more for use in hospitals, research labs, pharmacies, dental offices, and stockrooms where the RFID tag for specimens is extremely important. For commercial use, we do not need features such as remote temperature monitoring. Per their website, the main purpose of their refrigerators is to "eliminate paper work and reduce costs through automation and smarter purchasing." We aim to have our product to not have these expensive features and make it affordable and consumer friendly.

1.3 High Level Requirements

- The RFID reader must be able to send data through the microprocessor to the user device over Wi-Fi and recognize when an item is placed or removed.

- The camera must be able to take an image of a barcode, convert it into a compressed

JPEG format, and be able to interface with our microcontroller so that it can send the picture over WiFi for image processing. This whole process should take no more than 10 seconds.

- The user should be notified when the beer is placed correctly/incorrectly within 5 seconds after placing it through the use of the status lights.

2 Design

2.1 Block Diagram

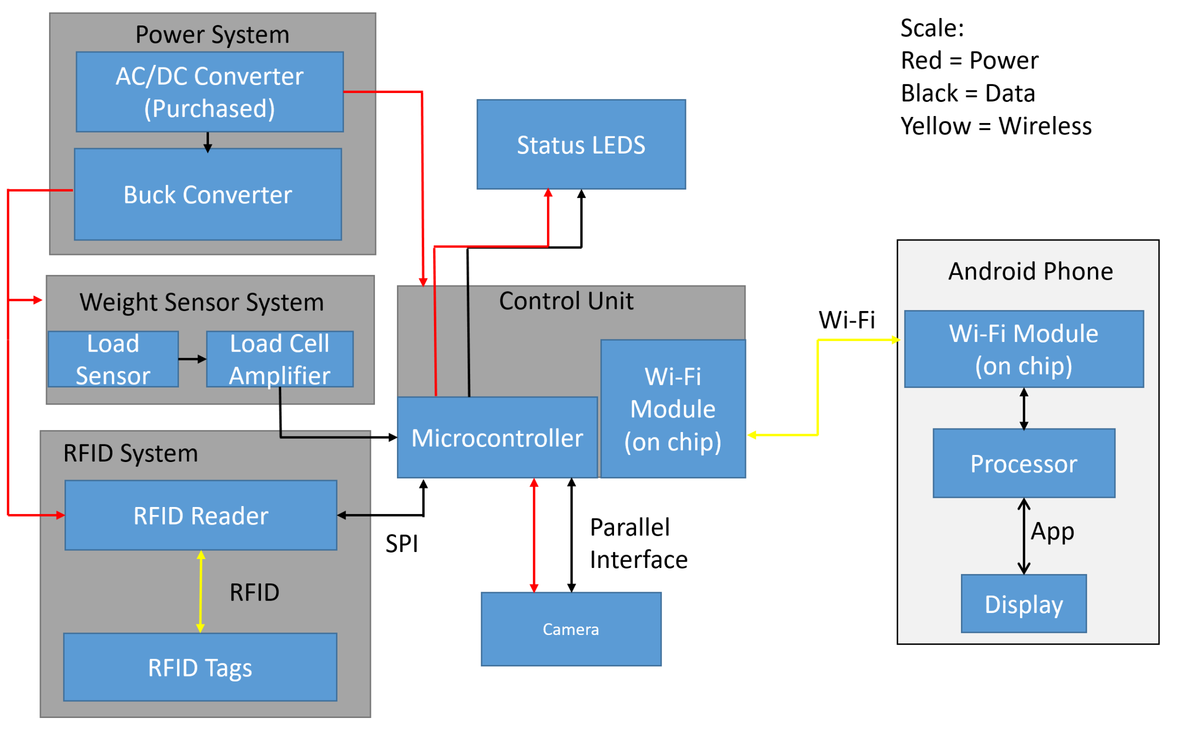

Figure 1: Block Diagram

Figure 1 shows the block diagram of our project. The RFID refrigerator can be broken up into 7 different components: a power system, RFID system, Weight Sensor System, LED Array, Microprocessor, Camera, and Software. The power system is going to be comprised of an AC/DC converter that is plugged in directly into the wall that takes in a 120V AC input and outputs a 5V DC output. It will then be attached to a buck converter that will take in the 5V DC input and output 3.3V DC. The power system will power up the RFID circuit, microcontroller and the camera.

The RFID system will consist of an RFID reader, along with passive RFID tags that we will place on the beer.

The camera will use the built in Camera interface in our microcontroller that will read the barcodes from the beer labels and then use the built-in WiFi Module in our microcontroller to send the image to our phone, where the processing will occur.

The weight sensor system will be comprised of a load sensor and a load cell amplifier, which will convert the electric signals from the load sensors and convert it into weight. In addition, there will be a plate on top of the the load sensor and the total weight will be recorded by the microcontroller. By doing this, the system will detect whether beer is removed from the fridge by the microcontroller detecting a change in weight on the weight sensor.

The LED Array will be used as a confirmation that the RFID tag was scanned correctly and whether or not it is removed or just placed in. That is, when we first put in the beer and scan the RFID tag with the reader, the LED array will display a green light, indicating that the beer is about to be placed in the fridge. It will only display green after the RFID tag is scanned and the beer is placed on the shelf (weight increases). When we remove the beer from the fridge and scan it with the RFID reader, the LED array will display red (the sensor will detect a decreased weight).

The microprocessor we use must be able to take in I/O from a camera, RFID reader, our weight sensor system, and send data over WiFi. This is why we chose the TI-CC3200 which has a built-in WiFi module.

Finally, on the software side, our device must be able to receive data from the microprocessor over the WiFi Module and store the total weight, RFID tag data, and barcode data after processing the barcode image. That is, the device must be able to take in an image from an outside source and then use a Barcode Scanner SDK with Android to process that image. After knowing what type of beer the barcode is, we will store the type of beer and quantity in an inventory. Finally, we will write a simple phone application that will allow the user to view his inventory and will alert the user whenever a beer is taken out incorrectly.

2.2 Physical Design

A 3.3 cubic foot refrigerator will be used. It is large enough to hold the weight sensor on one of the shelves. In Figure 2 it is shown the location of the weight sensor and the location of the circuitry. The circuit is mounted in the interior of the fridge door.

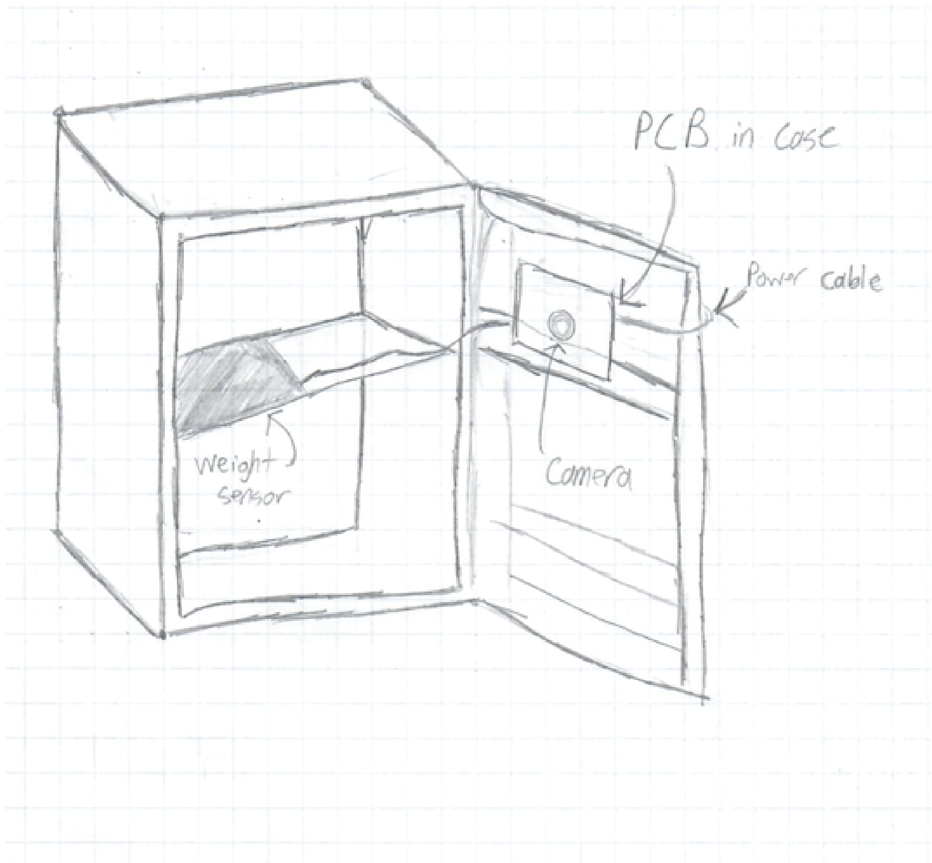

Figure 2: Physical Design for our RFID Fridge

2.3 Power Module

The purpose of our power system is to output a 3.3V DC voltage from a 120V AC Wall Power source. To do this, we must first use an AC/DC transformer that will convert our 120V AC voltage into a 5V DC output. That will directly power the weight sensor. The other components in the circuit require 3.3V so a buck converter will be used to efficiently lower the DC voltage from 5V to 3.3V. LXDC3EP micro DC-DC will be used because it can have 4- 5.5 V input and output 3.3V. The input range is important because the buck converter will take its input directly from AC/DC converter. Table 1 shows the voltage and/or current requirements for our different components.

Table 1: Voltage and/or Current Requirements for Components

Device	Voltage/Current Requirements
MCU	3.3V/250mA
RFID Reader	3.3V
Weight Sensor	5V
Status LEDS	3.3V
Camera	3.0V

2.3.1 AC/DC Transformer

First step is to take the 120VAC and convert it to 5VDC. The AC/DC converter we plan on purchasing is RAC04-05SC/W from Recom [10]. It takes an input voltage of 120VAC and outputs regulated 5VDC with a typical tolerance of 2%. Table **??** shows the requirements and verifications for the AC/DC converter.

2.3.2 Buck Converter

The AC/DC converter outputs 5V but in order to power MCU, and RFID system 3.3V is required. A Buck(step-down) converter will be used to step down 5V to a regulated 3.3V. The converter is required to have an input range that includes 5V in the acceptable range. The output will be traced to the MCU and RFID reader in order to power the components. LXDC3EP [11]micro DC-DC will be used because it can have 4- 5.5 V input and output 3.3V. The input range is important because the buck converter will take its input directly from AC/DC converter. Figure 3 shows our circuit for converting a 5V input DC into a 3.3V DC output. Table 2 shows the requirements and verifications for our buck converter. In addition, Figure 4 shows the load regulation for an input of 5V and an output of 3.3V.

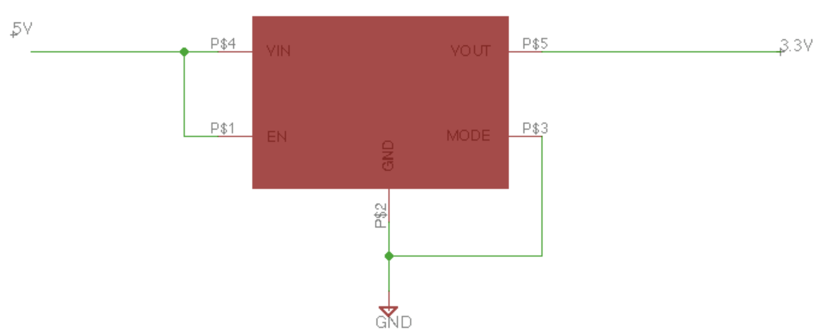

Figure 3: Buck Converter Circuit from 5V to 3.3V

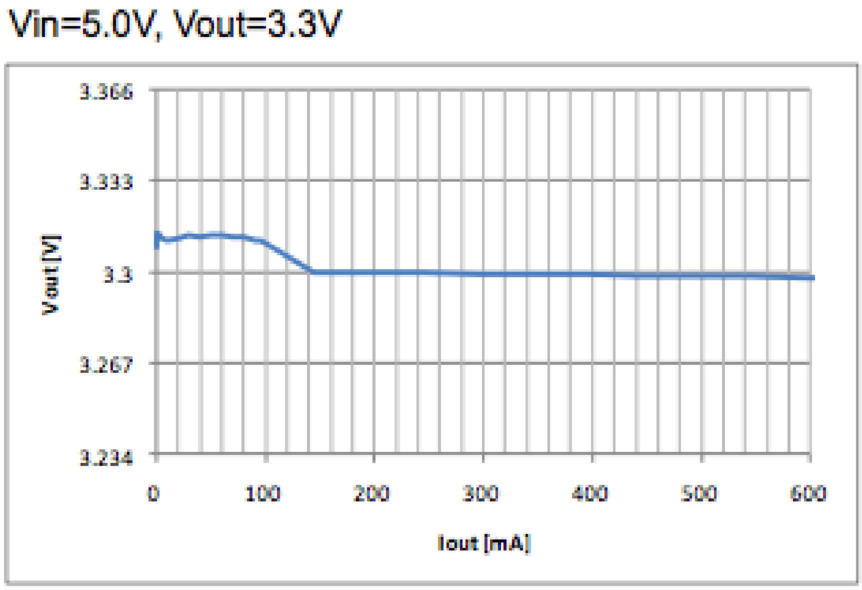

FIGURE 4: Load Regulation for VIN = 5V and VOUT = 3.3V [11]

TABLE 2: Requirements and Verification for the Buck Converter

Requirements	Verification
1. Output must be 3.3 with 10% tolerance with 5±5% V input.	Signal generator to create the min and max voltage input, and use oscilloscope to verify output.
2. Output current must be able to reach rated current max as according to data sheet of buck converter	Connect a resistive load with appropriate load that would lead to max current, verify with oscilloscope.

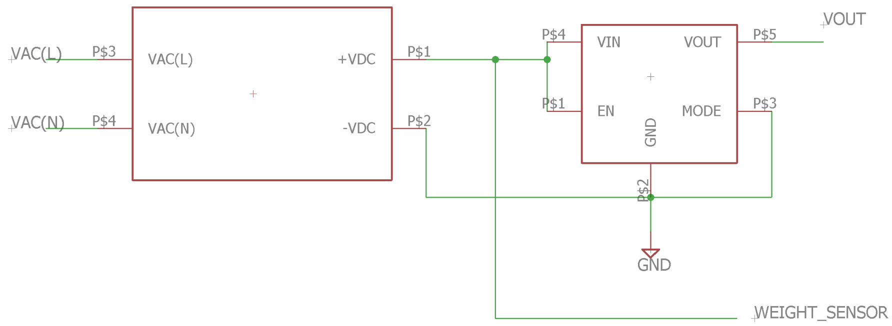

FIGURE 5: Overall Power Circuit

2.4 RFID Module

The RFID module consists of an RFID reader and passive RFID tags. The purpose of the RFID module is to tag the beer with one of the RFID tags and then scan the beer with the RFID reader, which then transmits the signal to the microprocessor that a bottle/can of beer was scanned and will be placed in the fridge. In addition, if the same tag is scanned twice (when you want to remove the beer), then the RFID reader transmits the signal that the beer was removed.

2.4.1 RFID Tags

The RFID tags will be placed on the items in the fridge and will store information. The type of tag we will implement is a passive tag. Passive tags do not require a power source because it will use the energy used from the RF waves from the RFID reader to power itself while also transferring information. Tags must be able to hold at least 512 bytes so we can store enough information on the item. The RFID tags only directly interact with the reader through RF waves at the 13.553 – 13.567 MHz range. [9] The frequency range is in the allowable range under FCC regulation and most passive, low range RFID systems operate in the range. RFID systems are generally cheaper than their UHF counterparts.

Table 3: Requirements and Verification for the RFID Tags

Requirements	Verification
1. Must be able to write new RFID data after each scan.	1. Through attempt to scan new data to tag (place tag next to RFID reader). 2. Verify that data is either removed or written on microcontroller

2.4.2 RFID Reader

A 4 wire Serial Peripheral Interface (SPI) is used to communicate between reader and microcontroller. The frequency must be within range in according to FCC regulations. The RFID reader will run on 3.3V with 10% tolerance so it can use power directly from voltage regulator. TRF7960 from TI has an input range of 2.7V to 5.5V which can be powered from the buck converter. The frequency used is 13.56 MHz.

2.4.3 Antenna

An antenna will be designed. It will be connected to the RFID reader IC and will communicate with the RFID tag. Will design a PCB antenna because it is cheap to create and can tune by changing the shape. The antenna will be attached to a matching network that will be connected to the RFID reader IC.

Table 4: Requirements and Verification for the RFID Antenna

Requirements	Verification
1. The antenna must transmit waves with a peak frequency in the 13.553- 13.567 MHz range	1. Attach the matching network circuit to the spectrum analyzer. . 2. Use spectrum analyzer with the antenna and verify the peak frequency is in the 13.553-13.567 MHz range.
2. Must be able to transfer data at 848kbps	1. Attach the matching network and antenna to a spectrum analyzer. 2. Verify that the bandwidth is 848 kbps.
3. Must be able to read data from reasonable distance, 10- 30 cm	1. Program the reader to read tags. 2. Wave a tag from 10 cm and 30 cm. 3. Verify on microcontroller that data was read.

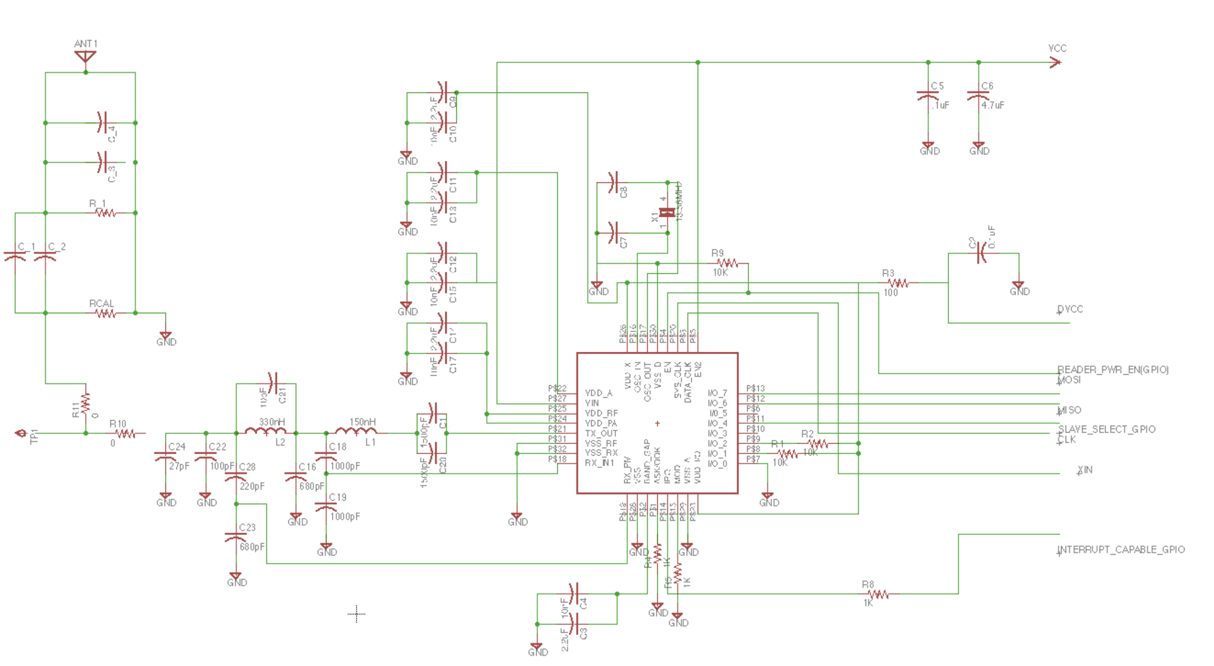

FIGURE 6: Circuit for Antenna and RFID Reader

2.5 Control Module

The control module handles the processing of the data from the RFID reader, camera, and weight scale through GPIOs and a built in Parallel Interface and is able to send this data over Wi-Fi to a user device. In addition, there will be an array of status LEDS that will display green if the beer is properly placed and red if it is improperly placed. It is powered by the power supply using the 3.3V DC output from our power system.

2.5.1 Microcontroller and Wi-Fi Module

The microcontroller, a Texas Instruments CC3200, handles all of the data from the weight sensor module, RFID module, and camera module. We chose the CC3200 because it conveniently has an 8 bit parallel interface for a camera with speeds up to 80 Mbps, along with 1 SPI port, and many GPIO ports. In addition, it has a build-in Wi-Fi module that is designed to send pictures over Wi-Fi via a web browser. In addition it must be able to communicate with the weight sensor via GPIO, an RFID reader via SPI, and a camera via a built in 8-bit parallel camera interface and send that data to a user device via a built-in Wi-Fi module. The microcontroller was chosen for its low cost and built-in Wi-Fi module with the "standard 802.11b/g/n radio, baseband, and MAC with a powerful crypto engine for fast, secure Internet connections with 256-bit encryption" [6]. In addition, this microcontroller is powered by a ARM Cortex-M4 core which has an 80MHz base clock so data transfers and processing can occur as quickly as possible. Table 6 shows the Requirements and Verifications for the Microcontroller.

TABLE 5: Requirements and Verification for the Microcontroller

Requirements	Verification
1. Must be able to receive data from the 8-bit parallel interface (camera)at 80 Mbps.	1. Take a picture of a timestamp with the camera. 2. When the picture is received at the MCU, stop the timestamp. 3. Calculate the time difference between the two time stamps. Repeat a few times and average the results.
2. Must be able to take a 16 bit digital input via SPI.	1. Use the ADC for our weight sensor module and an AC voltage with 0.1V, 1.5, and 2.5V amplitudes. 2. Verify that on our MCU that the input data is 16 bits by printing out the data. 3. The 0.1V input should have the lowest binary value, while the 2.5V input should have the highest binary value.
3. The MCU must be able to perform as a master device for the RFID reader (slave) via SPI. That is, the MCU must be able to initiate all communications with the reader.	1. Program the MCU to send a start condition to the RFID reader. 2. Verify that the RFID reader is able to receive data by scanning an RFID tag next to it. 3. Check on MCU that the RFID tag data is stored.

Table 6: Requirements and Verification for Wi-Fi Module

Requirements	Verification
1. The built-in Wi-Fi subsystem must be able to communicate over IEEE 802.11b/g/n at 4.5Mbps.	1. Setup the Wifi and register the IP Address on the MCU. 2. Write a program to send a 4.5MB packet of data with a timestamp and use a computer or other device to receive it and get the timestamp. 3. Check the time difference between the two time stamps. 4. Repeat a few times and average out the results
2. Wi-Fi module must be able to transmit and receive at 800ft with at least 18.2 dBm transmit power and -95.7dBm receiver sensitivity.	1. Place MCU 800 feet away from router in an open-field environment. 2. Measure the RSSI (Received Signal Strength Indication) by using an identical WiFi Module and given an input of 18.2 dBm, the received signal strength of the second module must be >-95.7 dBm

2.5.2 Status LEDs

The purpose of the status LEDs is to alert the user when the process of placing a beer is complete or incomplete. We would use an array of 5mm Bi-Polar Red/Green LEDs that would change colors based on the polarity. The LEDS would be attached on the door and be powered by the GPIO pins of the microcontroller. The LEDs would turn green whenever a beer is tagged, the barcode scanned, and placed on the weight sensor. In addition, the LEDs would also turn green if a beer is removed correctly. That is, removed from the weight sensor and scanned with the RFID tag. In any other situation where the beer is placed or removed from the weight sensors improperly, the LEDs would turn red. In addition, the LEDs would be off if no action has been taken. Table 7 shows the Requirements and Verifications for the Microcontroller.

Table 7: Requirements and Verification for Status LEDs

Requirements	Verification
1. The LEDs must turn on with a drive current of 2,4, or 6 mA (the drive strength of the GPIO for MCU)	Attach LED to breadboard with the pins programmed with a drive strength of 2, 4, or 6 mA and then see which option provides the brightest LED without burnout
2. Microcontroller must be able to change the polarity of its two ports in order to change LEDs from Red to Green	Program the microcontroller to change polarities every few seconds and use a multimeter to test if the voltage is reversed

2.6 Weight Sensor

The purpose of the weight sensor is to place it on a shelf and then the microcontroller would be alerted whenever there was a beer placed or removed. The system is composed of two components: load sensor and a load amplifier. In addition, there would be a plate on top of the load sensors. We would be using 1 load sensor in the middle of the plate and the four corners would have some stubs on the bottom so that the plate does not tip on one side. This plate would be on top of the already existing shelf of the refridgerator. The beer would be placed on the shelf and then the analog signal would go through a load amplifier into a digital signal which would go into the microcontroller. According to General Electric, the dimensions on a 3.3 cubic foot refrigerator is around 16 inches wide and 14 inches deep, and the average size of a can of beer is 2.6 inches in diameter, and for a bottle it is 2.4 inches. So on average, we can put

$$\frac{16 \quad inches \quad wide}{2.6 \quad inches \quad per \quad beer} = 6 \quad beers \tag{1}$$

wide and

$$\frac{14 \quad inches \quad deep}{2.6 \quad inches \quad per \quad beer} = 5 \quad beers \tag{2}$$

deep. So that means we can store up to 30 beers in our fridge. On average, a bottle of beer would be heavier than a can of beer, so we will use a bottle of beer as the upper bound. A bottle of beer tends to weigh around 1 pound, so the maximum capacity of our fridge would be 30 pounds.

2.6.1 Load Sensor

The load sensor that we will use is the SEN-10245 made by Sparkfun. This particular load sensor can measure up to 50kgs, or 110 pounds. This particular type of load sensor uses a strain gauge load cell, where the force is being sensed by the deformation of strain gauges on the element. A strain gauge is a device that measures electrical resistance changes in response to force that is applied to the device. Each different type of gauge has a different sensitivity to strain and the output is generally measured by resistance. However, the base resistance of each gauge isn't that high, and most microcontrollers would not be able to detect the weight, so we must also use an amplifier to do so [7].

For our design, we will use 1 load sensor, which can detect up to 110 pounds. We would choose sensors with a lower range of weight for more accuracy, but this was the cheapest option.

2.6.2 Load Amplifier

For our load cell amplifier, we will use the HX711 IC that will allow us to easily read load cells to measure weight. The purpose of a load cell amplifier is to be able to more easily read the changes of the resistance of the load cell, and by calibrating with the microcontroller, we will be able to get very accurate weight measurements. This particular Load Amplifier is a 24-bit analog-to-digital converter designed for weight scales. By using a 24-bit ADC, we can achieve higher resolution for weight. That is, we can more easily detect small changes in weight. It will take in a power supply of 2.7-5.5V, so our PSU should be able to provide enough power for our load amplifier. I The HX711 uses clock and data pins, so we should be able to connect it with the GPIO pins of our microcontroller. The microcontroller and the load cell amplifier both use a clock of 80Hz. The HX711 Load Cell Amplifier accepts 5 wires from our load sensor. Four of the wires are hooked up in a Wheatstone Bridge Formation as shown in Figure 7.

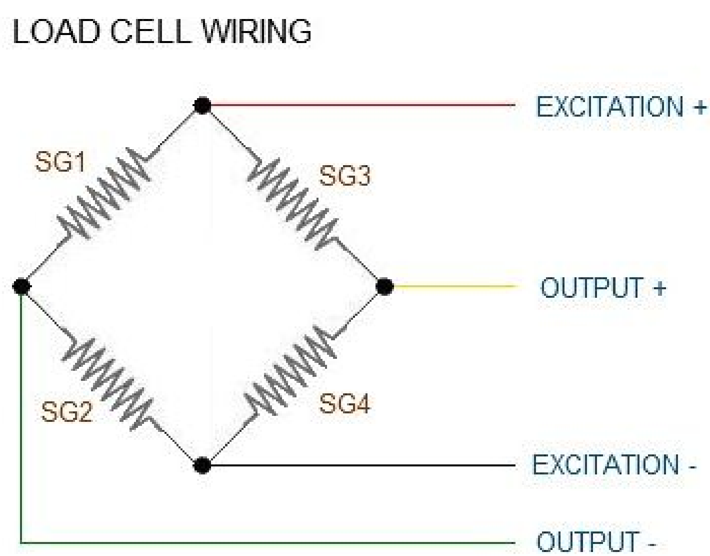

FIGURE 7: Load Cell Wiring [7]

How a wheatstone bridge works is that it is a configuration of four resistors with a known applied voltage. Essentially, it is a voltage divider circuit where the output voltage is dependent on the input. Figure 8 is an example of a generic Wheatstone configuration. The output voltage can be calculated by using the equation 3:

$$V_{out} = (\frac{R3}{R3 + R4} - \frac{R2}{R1 + R2})V_{in} \tag{3}$$

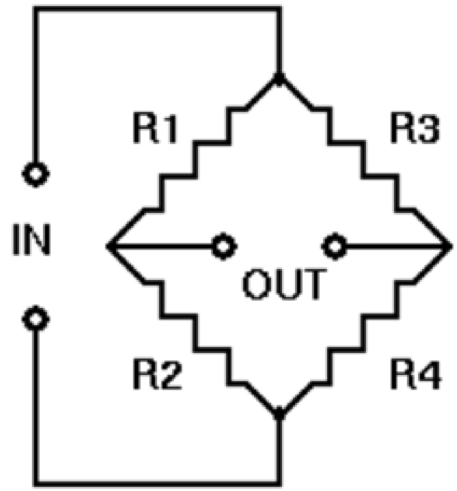

FIGURE 8: Wheatstone Configuration [7]

By replacing one of the resistors in a wheatstone bridge with a strain gauge, we can measure the change in V_{out}. Figure 9 shows how the Wheatstone Bridge is connected to the Circuit.

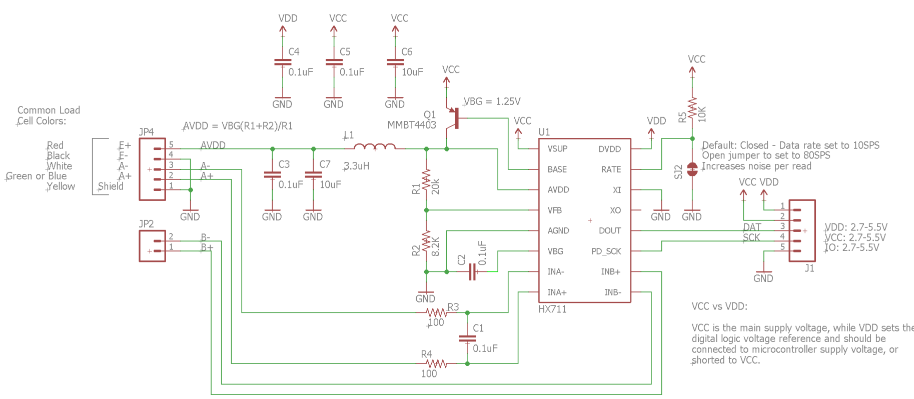

FIGURE 9: Circuit of Load Cell Amplifier [14]

Table 8: Requirements and Verification for Weight Sensor System

Requirements	Verification
1. Each plate must have a maximum weight of 30 pounds. The load sensor should output 0V for no weight weight and 3.3V for the highest weight with ±5% tolerance for the voltage.	1. Acquire three different weights of 1lb, 10lbs, and 30 lbs. 2. Place three weights on plate separately 3. Use a multimeter to measure the output voltage of the weight sensor system. 4. The 1lb weight should output the lowest voltage, whereas the 50lbs should output 2.5V. Also, with no weight, the load sensor should output 0V.
2. Each shelf must have a sensitivity of ±0.5 pounds.	1. Acquire three different weights of 0, 10, and 30 pounds. 2. Place three weights on the shelf separately. 3. Place a 0.6 pound weight and verify on the MCU that the weight changed.
3. The load amplifier must be able to take in an input of 0-2.5V with a tolerance of ±5% and convert it to a digital signal, where it will be processed by the microcontroller and interpreted as weight	1. Use an AC power source and use a standard sine wave in High-Z mode with amplitudes of 0.1V, 1.5V, and 3.5V. 2. Connect the output of the AC power source into the ADC and connect the digital output of the ADC into the MCU via SPI. 3. Record the value that is outputted by the ADC. 4. Make sure that the 0.1V has the lowest number, 1.5V has the middle number, and 3.5V has the highest number.

2.7 Camera Module

The purpose of the camera module is to take a picture of a barcode with a high resolution. The camera that we are using is the MT9D111 because it interfaces really well with our microcontroller. That is, it connects with our microcontroller using the 8-bit parallel interface at 80MBps. In addition, the MT9D111 has an auto focus mode, which is important in getting a high resolution image of our barcode. To scan a barcode, however, the resolution itself does not matter. Instead, we must take a picture such that the barcode is wide enough. That is, the smallest bar or space in the barcode must take at least be two pixels wide. Figure 9 shows the requirements and verification for the camera module.

Table 9: Requirements and Verification for Camera Module

Requirements	Verification
The camera must be able to take a picture of a barcode from 1 to 5 inches away and the smallest space or bar must be at least 2 pixels wide.	1. Program the camera to take pictures with the MCU. 2. After programming, toggle the camera to take a picture of a barcode that ideally has perfect resolution (take the barcode from a commercial product). 3. Transfer the image to the computer. 4. Verify that each the minimum size of the smallest bar or space is 2 pixels wide. 5.Repeat for 1, 3, and 5 inches away from camera
The camera must be able to capture an uncompressed QVGA image and convert it into a JPEG format and then send it to a MCU.	1. Capture image with camera with MCU. 2. Use the MCU to send the picture over Wi-Fi to a user device (computer or phone). 3. Check whether the sent image is in a compressed JPEG format.

2.8 MCU Circuit with Camera and LEDs

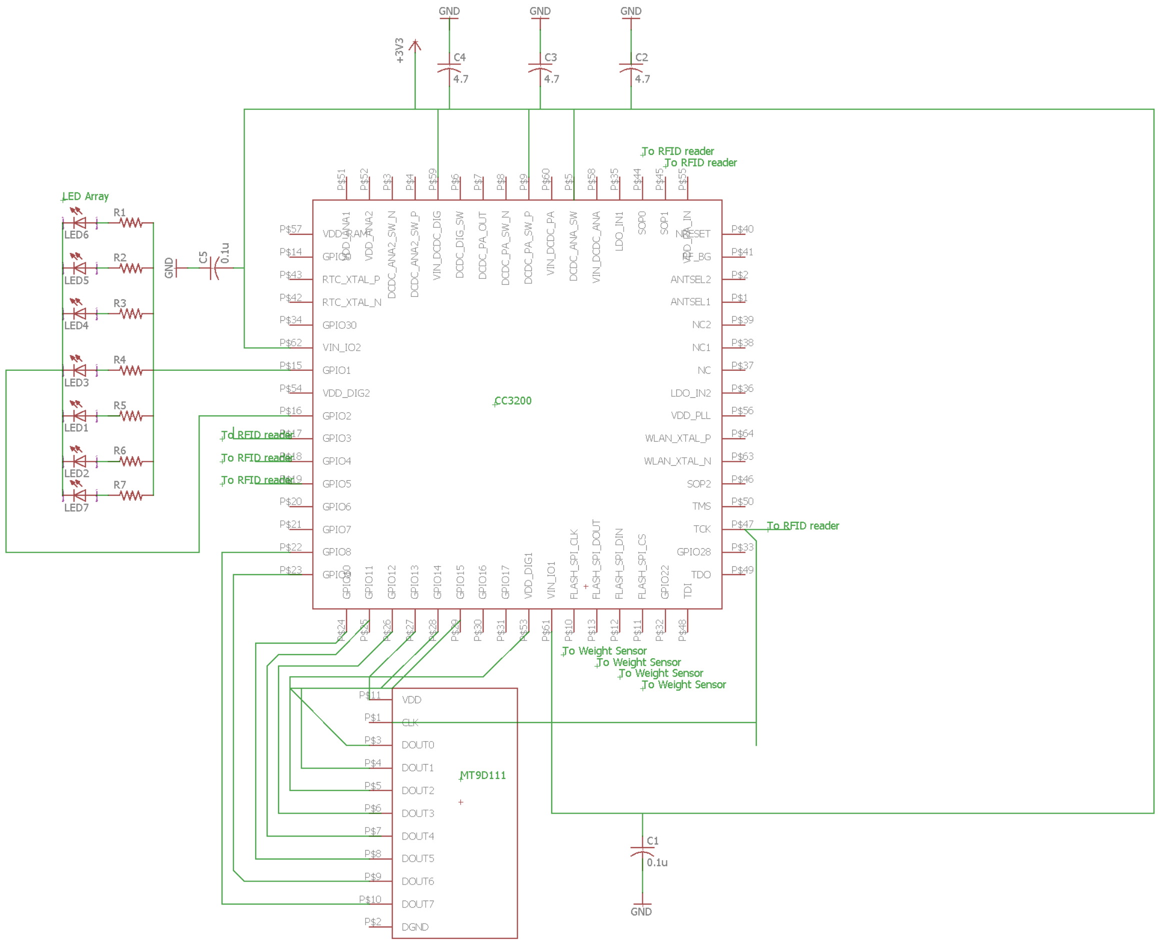

FIGURE 10: General Circuit of MCU with Camera and LED Circuit

2.9 Overall Power Consumption

The power supply should be able drive 300 – 400mA (average) at 3.3V. The microcontroller on average requires 275mA, the RFID reader IC requires 10mA and the Force sensor requires 5 mA. The camera is powered through MCU and is taken into account. The LEDs will take 10 mA each. The 400mA requirement falls below the 1000mA load current drive max of the buck converter and the 800mA max current output of the AC/DC converter.

2.10 Software Module

The software is responsible for processing the barcode, information retrieval, and both sending and receiving the data from the user. The processing will occur mostly on the user device (i.e. a phone), and the final inventory should be available to all users on the same Wi-Fi network.

2.10.1 Data Transmission

The CC3200 is a single-chip microcontroller unit with built-in Wi-Fi connectivity, and it will be the main point of communication between the fridge and the user. After receiving the image from the cameras, it will then send the image to the user device for processing. If the image is correctly processed the microcontroller will then receive the retrieved product information for the beer from the user device. In the event the image was not recognized, the microcontroller will receive an error signal from the device which it can then use to alert the user. Figure 11 shows the flowchart for this process

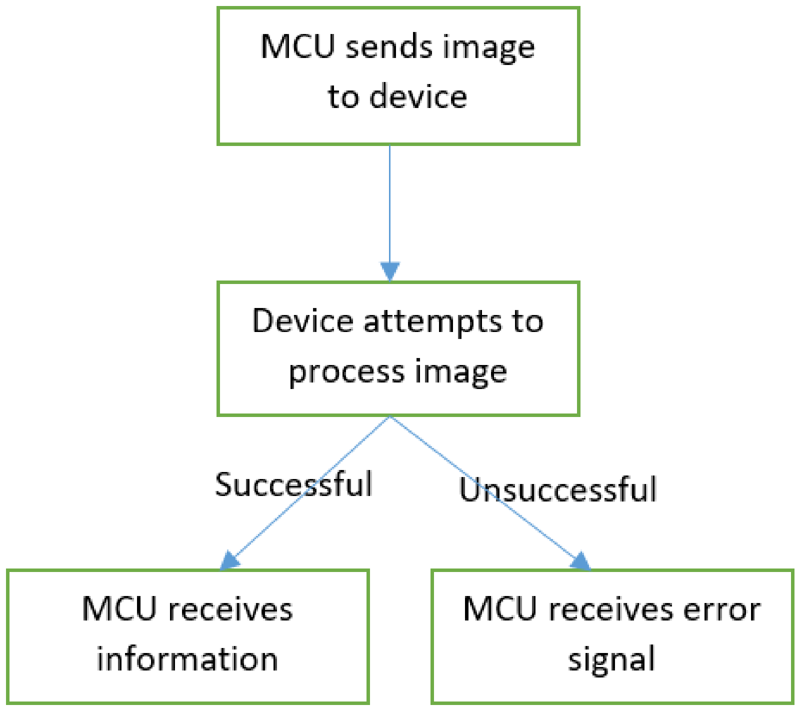

FIGURE 11: Microcontroller to User Device Flowchart

2.10.2 Data Processing

The core of the data processing will occur on the user device itself. We will attempt to use ZXing, an open source project that allows a user to scan 1-D or 2-D graphical barcodes. Upon receiving an image from the microcontroller, the user device will then use ZXing to analyze the image and retrieve the product name from a barcode database.

TABLE 10: Requirements and Verification for data processing

Requirements	Verification
The software must be able to send received barcode images to the SDK and retrieve information.	1. Take a clear, unobstructed picture of a barcode and ensure that existing applications can recognize it. 2. Use the same photo and send it to the application. 3. Verify that the application can recognize the image as a barcode. 4. Verify the accuracy of the information retrieved by comparing it with manual lookup.
User must be able to able to modify all inventory entries associated with each item (i.e. warning date and amount), including creating and deleting items in the inventory.	1. Upon ensuring the software is able to recognize the image as barcode, attempt to have it search up the barcode information. 2. Verify the accuracy of the information retrieved by comparing it with manual lookup.

2.10.3 User Interface

There will be an inventory that contains a list of all the types of beer and how long they have been within the fridge. The user will be alerted if products are removed from the fridge incorrectly. This inventory should be available to all users on the same Wi-Fi network.

2.10.4 Inventory

We will use an inventory to keep track of all items in the fridge. There will be two entries for each item: beer name, and amount. By default, the user will be alerted if products remain within the fridge for more than a week. However, the user will also have the ability to manually change the expiration date for products that may have been purchased near the end of their shelf lives. Users should also have ability to manually add and delete entries from the inventory. This inventory should be available to all users on the same Wi-Fi network

Table 11: Requirements and Verification for the Inventory

Requirements	Verification
All users on the same Wi-Fi network must be able to access the inventory	Have all group members attempt to access the inventory while on same network.
User must be able to manually modify the inventory	1. Have user input a custom entry and verify that the inventory has been updated. 2. Have user change amount of item and verify that the inventory has been updated. 3. Have user change expiration date warning and verify that the inventory has been updated.
Inventory must be able to store at least 50 items	Manually enter 50 test points and verify that the inventory is able to successfully hold all points

2.11 Tolerance Analysis

One important risk we must consider is the readability of the barcode pictures. One important aspect for unreadable barcodes is that the resolution is too low. However, for a barcode reader SDK on Android to work, resolution itself does not matter: what matters is how wide the barcode is, measured in pixels [12]. The number of pixels for the barcode depends on the number of characters encoded and the type of barcode. In addition, a barcode can only be decoded if every bar and space in the original barcode is preserved in the scanned image. That is, if the bars and spaces somehow blur into each other, then the barcode would be extremely difficult to decode. A barcode module is defined as the smallest width of a bar or a space in a barcode, and all other bars and spaces are merely multiples of a module. A perfect scan of a barcode would only need 1 pixel. However, this will most likely result in a barcode with mostly shades of grey. Thus, we must allow at least 2 pixels for each module. Additionally, different barcodes have different pixel widths, depending on the type of barcode being used and the number of characters being encoded in the barcode. For a UPC-A/EAN-13 barcode, which are generally found on beer, the pixel width is 166 Pixels. Next, to calculate the resolution that we need to use, we use Equation 4:

$$Resolution(dpi) = \frac{Pixel \quad width}{Width \quad of \quad a \quad barcode \quad in \quad inches} \tag{4}$$

So if we had a barcode width of 2 inches, the minimum resolution that we would need is shown in equation 5.

$$Resolution = \frac{166}{2} = 83ppi \tag{5}$$

Note that the scanner SDK normally has resolutions of 100, 200, etc ppi so if our Resolution were 83 ppi in this case, then we would use a 100 ppi resolution.

Another issue we may encounter is the low contrast of the barcode. That is, during when a user wants to scan in an item, the lighting in the room may be too low for the phone to process the image. This is because the barcode scanner must be able to differentiate between the light and dark elements of the symbol. If there is not enough contrast between these two barcode elements, then the barcode SDK might be unable to distinguish the barcode from the substrate (the black from the white) and the result will most likely end in a read failure. Figure 12 shows a couple of examples of unreadable barcodes.

Figure 12: Examples of Unreadable Barcodes [13]

Ensuring distinct and uniform barcode elements is the first step to preventing unreadable codes due to low contrast [13]. To do this, we can use lighting equipment that is tailored to produce the most uniform and highest-contrast images of barcodes. If our beer has a glossy label, we can use diffused lighting. Figure 13 shows the difference between proper lighting on a glossy label. We can also use a dark field lighting to enhance the white-parts.

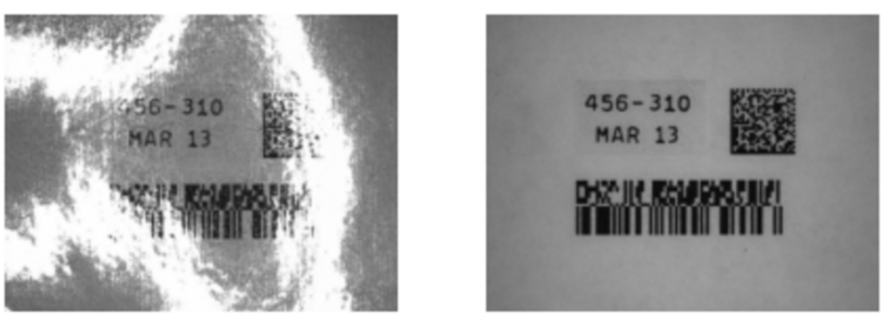

Figure 13: Effects of Proper Lighting on Glossy Label [13]

Another big issue we may encounter is with the read angle of a barcode. That is, a barcode sometimes may fail to be read due to the physical position of the camera relative to the barcode. Thus, it is important for our Android SDK that we use that it is able to auto-adjust the barcode to a right angle. That is, if it is slightly angled, the SDK must adjust the code so that it is level.

3 Cost and Schedule

3.1 Cost Analysis

Our labor costs are estimated to be $40/hour, 250 hours total, for three people. This brings our total estimate to $30,000.

Name	Hourly Rate	Hours	Total	Total x 2.5
Jeffrey Lee	$40	250	$10000	$25000
Yuanhao Wang	$40	250	$10000	$25000
William Mercado	$40	250	$10000	$25000
Total				$30000

Description	QTY	Manufacturer	Part #	Cost/Unit	Total
RFID Reader Module	1	TI	TRF7960RHBR	$2.80	$2.80
AC/DC-Converter	1	Recom	RAC04-05SC/W	$15.30	$15.30
Buck Converter	1	Murata	LXDC3EP33F-204	$2.35	$2.35
RFID Tags	5	Avery Dennison RFID	600368	$0.68	$3.40
MCU	1	TI	CC3200R1M1RGCR	$11.63	$11.63
2MP Camera	1	Micron	MT9D111	$18.99	$18.99
Load Cell Amplifier	1	Sparkfun	SEN-13879	$9.95	$9.95
Status LED	4	Lite-On Inc.	LTL-293SJW	$0.50	$2.00
Load Sensor - 50kg	1	Sparkfun	SEN-10245	$9.95	$9.95
Shelf	1	N/A	N/A	$5.00	$5.00
PCB	5	ECEB Parts Shop		$4.00	$20.00
Total					$101.37

Section	Total
Labor	$30000
Parts	$101.37
Total	$30101.37

3.2 Schedule

Week of	Jeffrey	Yuanhao	William
2/27	Purchase the MCU, camera, and weight sensor modules. Begin work on LED array (make PCB)	Begin work on Barcode SDK for Android	Purchase the RFID System and Power Module Parts and Acquire Refrigerator
3/6	Run the Wi-Fi Tests for the MCU and determine how many FSRs needed for each shelf	Start work on interface between Android and MCU	Assemble Power Supply Module on Breadboard for testing and come up with PCB for PSU
3/13	Start work on camera tests for MCU and come up with Physical Design for Fridge (Group needs to submit final design to Machine Shop before spring break)	Confirm that Android device can receive pictures over Wi-FI	Assemble RFID circuit on Breadboard for testing and come up with PCB
3/27	Make sure Camera is able to take pictures of barcode at desired resolution. Work with Yuanhao to perfect Barcode Scanner interface	Begin work on Android app	Make sure RFID module and MCU are interfaced correctly and come up with PCB for FSRs
4/3	Help with Assembly of Prototype and make sure camera and weight sensors are accurate (test RV for both)	Make sure RFID, weight, and barcode are able to be read by Android application	Begin assembly for Prototype (get a working prototype before end of week so we can debug the next week) and test RV for RFID system.

4/10	Begin final presentation and report. Debug whole system	Begin final presentation and report. Debug whole system	Begin final presentation and report. Debug whole system
4/17	Mock Demo Feedback revisions	Mock Demo Feedback revisions	Mock Demo Feedback revisions
4/24	Work on Final Report after Final Demo	Work on Final Report after Final Demo	Work on Final Report after Final Demo
5/1	Final Report	Final Report	Final Report

4 Ethics and Safety

The main safety we might encounter with our project is the camera and regulator circuity. These cameras will be placed in the fridge, where the moisture could lead to short-circuits and damage components. While the microcontroller itself doesn't need to be in the fridge, it will share connections with cameras, which are inside the fridge. We will need to ensure that our camera and microcontrollers and their connections are cased so that the moisture will not cause any safety hazards.

As engineers, we must be sure to "credit properly the contribution of others" as stated in #7 of the IEEE Code of Ethics [4]. We acknowledge that we will be building off of or incorporating many already existing components into our project, such as the ZXing library. We will ensure to always cite the authors wherever necessary to avoid accidentally representing any of those components as proprietary.

As engineers, we are called "to improve the understanding of technology; its appropriate application, and potential consequences", as stated in #5 of the IEEE Code of Ethics [4]. We believe that our project will have a positive impact on the application of RFID technology, and will serve to provide a way for an owner to protect his items. However, we are responsible for all information sent through our technology, and it is known that RFID tags can be compromised, and have their information extracted, deleted, or rewritten. In event that the tags are compromised, a user could lose track on items in the fridge, which could result in potential false alerts. However, we believe that the likelihood of a dedicated attack on our system is small, simply because of the low value of goods that our system is intended to deal with. Some users may attempt to use our project to facilitate in their storing of items that may be deemed illegal, or could be used to harm others. While we do not have a method to ensure that our project is used solely for legal purposes, we do not believe that limiting the functionality of our system is the right course of action, as they can easily use any number of alternatives. Overall, we believe that our project will have successfully used technology to improve the life of others.

5 Citations

[1] "Americans waste $2,200 worth of this every year ...", CNBC, 2017. [Online]. Available: http://www.cnbc.com/2015/04/22/americas-165-billion-food-waste-problem.html. [Accessed: 08- Feb- 2017].

[2] "FAO - News Article: Cutting food waste to feed the world", Fao.org, 2017. [Online]. Available: http://www.fao.org/news/story/en/item/74192/icode/. [Accessed: 08- Feb- 2017].

[3] "Terso Solutions", Terso Solutions, 2017. [Online]. Available: https://www.tersosolutions.com/solutions/rfid-refrigerators/ts074-rfid-refrigerator/. [Accessed: 08- Feb- 2017].

[4] "IEEE IEEE Code of Ethics", Ieee.org, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08- Feb- 2017].

[5] Texas Instruments, "SIMPLE SWITCHER Power Converter 150-kHz-A Step-Down Voltage Regulator," LM2596 datasheet, Nov. 1999 [Revised May 2016].

[6] Texas Instruments, "CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things Solution, a Single-Chip Wireless MCU," CC3200 datasheet, July 2013 [Revised Feb. 2015].

[7] Load Cell - 200kg, Disc (TAS606) et al. "Getting Started With Load Cells - Learn.Sparkfun.Com". Learn.sparkfun.com. N.p., 2017. Web. 10 Mar. 2017.

[8] Micron, "1/3.2-Inch System-On-A-Chip (SOC) CMOS Digital Image Sensor," MT9D111 datasheet, July 2005 [Revised Jan. 2006].

[8] Interlink Electronics, "FSR Force Sensing Resistors," FSR datasheet

[9] Ian Poole, "RFID frequency bands & spectrum," in Radio-Electronics.com, Google+. [Online]. Available: http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/low-high-frequency-bands-frequencies.php. Accessed: Feb. 24, 2017.

[10] Recom, "Powerline AC/DC Converter," RAC04-C/W datasheet, 2011 [Revised 2015].

[11] muRata, "Micro DC-DC Converter," LXDC3EP TypeF series datasheet, Feb. 2016.

[12] admin, ”Image resolution and Barcode reader,” in Image, bardecode.com, 2012. [Online]. Available: http://www.bardecode.com/en1/image-resolution-and-barcode-reading/. Accessed: Mar. 1, 2017.

[13] M. Systems, ”The most common causes of unreadable Barcodes - Barcoding news,” in Solutions, Barcoding News, 2015. [Online]. Available: https://www.barcodesinc.com/news/?p=11689. Accessed: Mar. 1, 2017.

[14] HX711, SparkFun et al. ”Sparkfun Load Cell Amplifier - HX711 - SEN-13879 - Sparkfun Electronics”. Sparkfun.com. N.p., 2017. Web. 10 Mar. 2017.