Automated Cart Checkout

Fall 2019 ECE 445 Design Document

Team Members: Timothy Lan, Lucas McDonald, Andrew Wang

TA: Yangge Li

Table of Contents

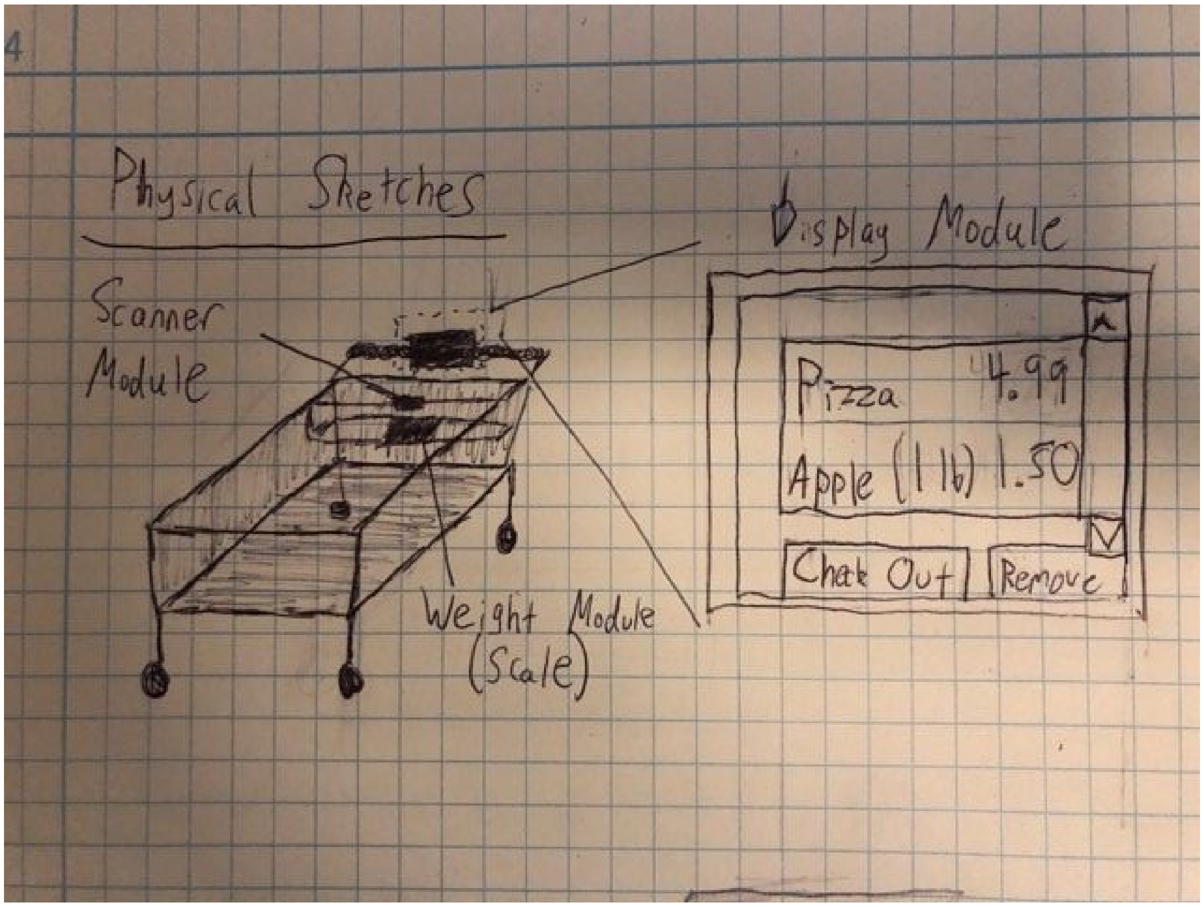
1. Introduction

1.1. Problem and Solution Overview

People want to have a seamless shopping experience, but they are often met with various parts of shopping that delay and otherwise frustrate the customer. Most current shopping systems employ manual checkout, either via a cashier or through self-checkout. These systems both lead to long checkout lines when a store is crowded, creating a point of “friction” in the shopping experience. Many companies are aware of this issue and have tried to address it using new shopping models. Amazon Go is by far the most popular of these, utilizing a sophisticated computer vision system to track customers through the store. However, the Amazon Go model is prohibitively expensive, costing around $1 million to set up each store [6]. In addition, its computer vision system is extremely complex, taking up years of development time at one of the world’s largest companies and incurring heavy capital sunk costs. Thus, the need for a solution to the checkout experience is ongoing.

We propose a solution that would reduce friction in the shopping experience at a significantly lower cost than Amazon Go’s solution. We propose adding a modified portable self-checkout system to a store’s shopping card. The system would include a barcode scanner and weight sensors to handle all items in the store that the consumer might buy. For most products, the customer would pick up a product off of the shelf, scan its barcode with their in-cart system, and add it to their cart. For products that are priced by weight, like tomatoes or other produce, the customer would put it on the weight sensor to get the correct price.

1.2. Visual Aid



We present a rough sketch of the device mounted to a shopping cart and a mockup of the touch screen UI.

The device would be installed in the back area of a shopping cart. The display module would face the customer as they push the cart. The scanner would face inside the cart, so a customer would scan the product's barcode as they drop the product in the cart. The scale would be put on or near the child seat.

Note that we do not intend to outfit a shopping cart for demo purposes. We will develop the device to be installed in a shopping cart as shown, but will likely demo in something like a basket or a bag.

The touch screen UI would display all of the products in a user's cart. The user would be able to scroll through a list of the products in their cart and remove products if they wish. When the user is ready to check out, they would tap "Check Out" and select a payment method.

1.3. High-level requirements list

- Customer must be able to use the device to add and remove items priced by weight and by quantity in a store to their cart.
- Scanning a product’s barcode must be within tolerance and succeed with little error.
- All system modules should be as cheap as possible; ideally, the total cost of all system modules should be under $200.

2. Design

2.1. Block Diagram

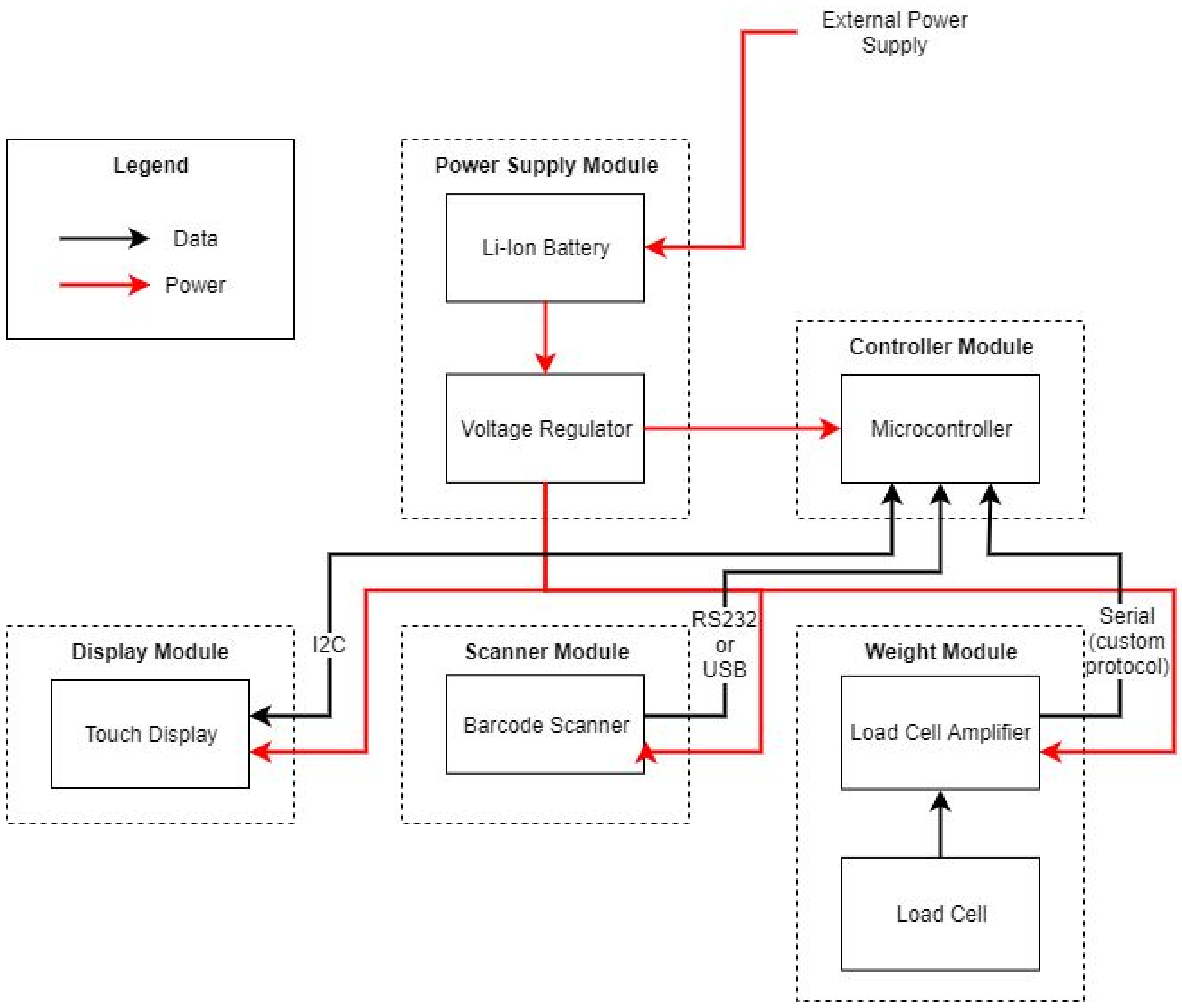


Figure 1: High-Level Block Diagram. The External Power Supply is used to charge the battery and is not part of the overall design.

2.2. Functional Overview

2.2.1. Scanner Module

The scanner module will scan barcodes placed in front of it and report the scanned value to the controller module. This module will allow us to scan the barcodes of products to identify them.

We will use the YHD-M800 barcode scanner. It will communicate with the controller module using the USB protocol. It will send the controller data about the barcode that was scanned. It will be connected to the 5V power supply.

2.2.2. Weight Module

The weight module will measure the weight of objects placed on it and report the weight to the controller module. This module will allow us to determine the correct cost for objects that are priced by weight.

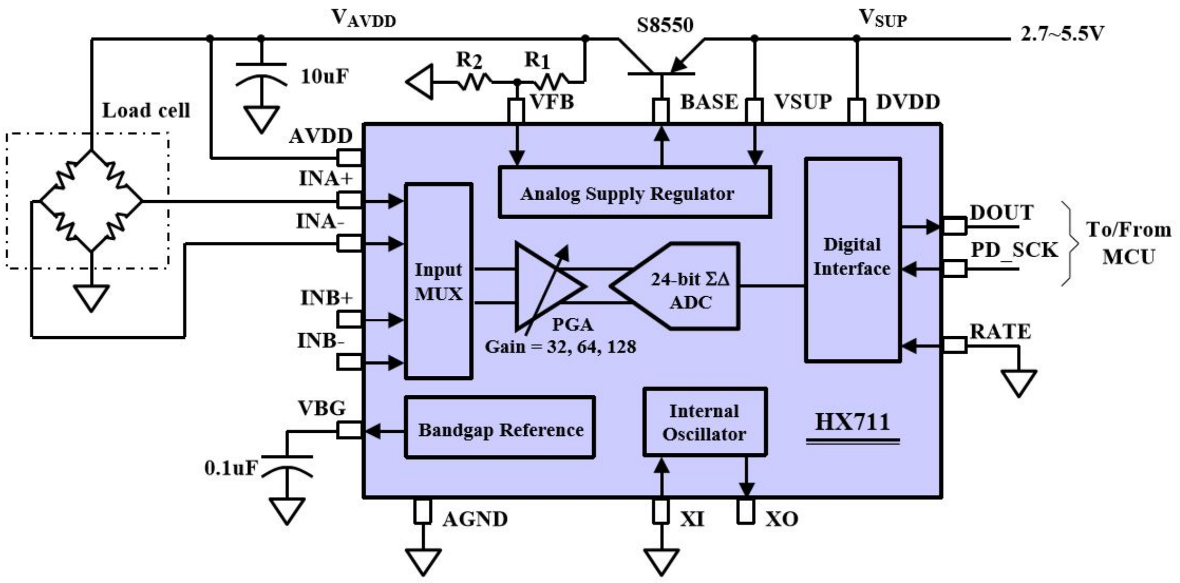


Figure 2.1. Schematic for Load Cell + Load Cell Amplifier (as provided in datasheet)

2.2.2.1. Load Cell

The load cell is able to report the weight of objects that are placed on it. This will allow us to measure the weight of objects.

We will use the SEN-10245 load cell. It has a 50kg weight limit. It connects directly to the load cell amplifier with no additional connections.

2.2.2.2. Load Cell Amplifier

The load cell amplifier interfaces the load cell with the controller module. It allows the weight measured by the load cell to be reported to the controller module.

We will use the HX711 load cell amplifier. It will be connected to the load cell and the controller module to pass data between the two using a custom serial protocol specific to the chip. It will also be connected to the 5V power supply.

2.2.3. Display Module

The price display function will be performed by a touchscreen that will display the total price of objects that have been scanned so far. This will allow the customer to see their current total. It will also have the ability to take multiple payment methods (such as Credit Card or Apply Pay) in software.

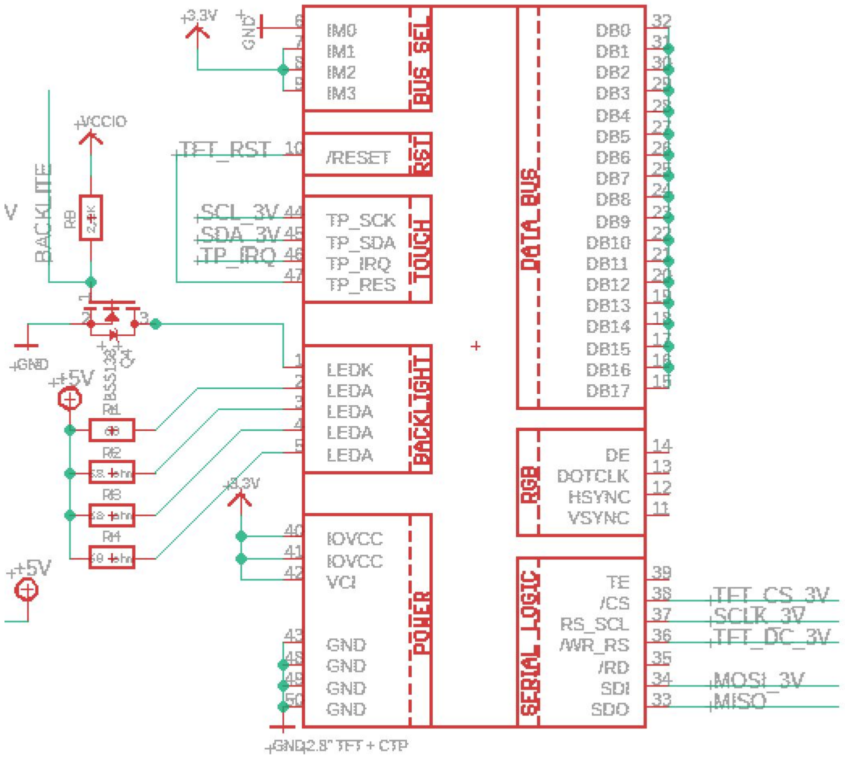


Figure 2.2. Schematic for Touch Display (as provided in datasheet)

2.2.3.1. Touch Display

We would use the Adafruit 1947 Touch Shield with Capacitive Touch as our touch display. It will interact with the microcontroller to display a running total, allow users to remove items from their cart, and give users checkout capabilities. When the checkout feature is selected, it will send a reset signal to the microcontroller.

2.2.4. Controller Module

The controller will provide an interface for all other subsystems. This will allow the components of the device to function together properly. It would also keep track of the items that the customer scanned and the total price of all items scanned so far. Initially, prices for objects will be hardcoded into the controller; however, given time, a database can be built to replace the hardcoded values once initial functionality is achieved.

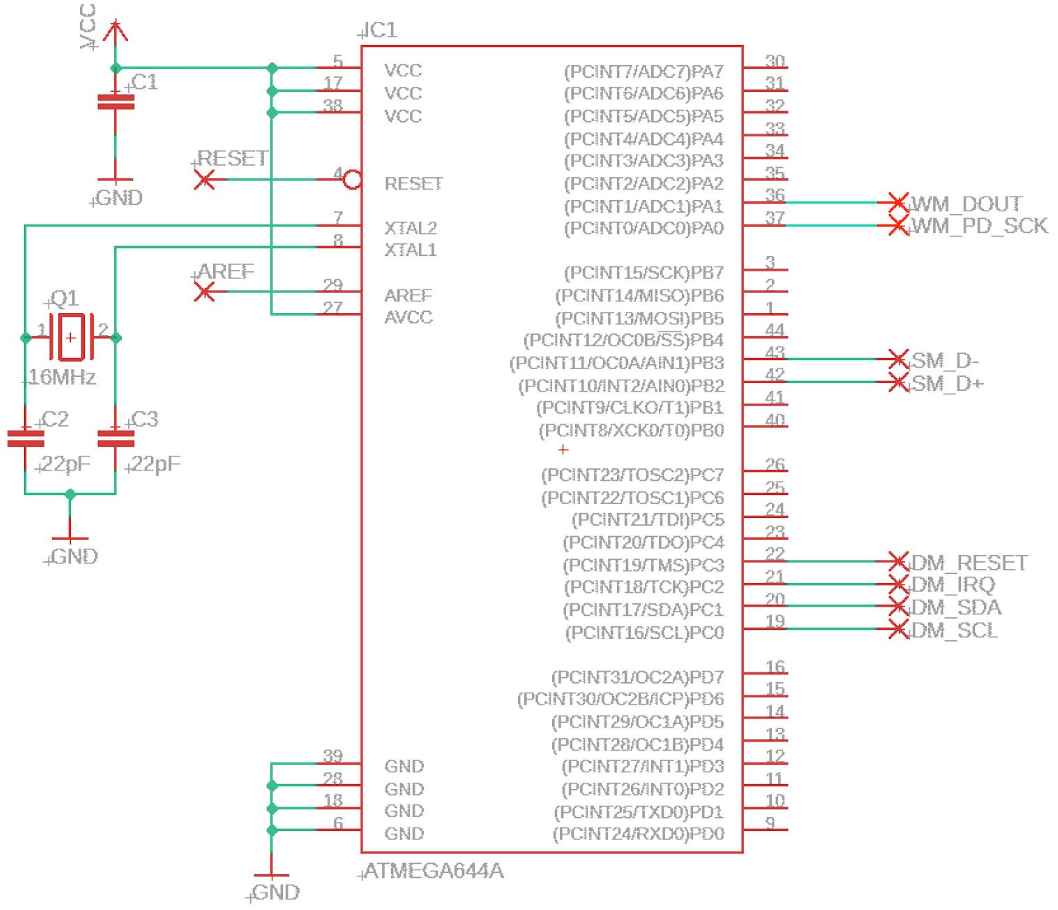


Figure 2.3. Schematic for Microcontroller

2.2.4.1. Microcontroller

We would use the ATMEGA644A-AUR (or ATMEGA328P) or a similar controller.

2.2.5. Power Supply Module

The power supply module will provide power for all device components. This will ensure all components get the power they require.

2.2.5.1. Li-ion Battery Charger

The Li-ion battery charger will ensure that the Li-ion battery capacity will be ready to go for the next user of the shopping cart.

2.2.5.2. Li-ion Battery

The Li-ion battery will store power for the system. Using a battery will allow the device to be portable and able to be installed in a shopping cart.

We will use a 5V battery.

2.2.5.3. Voltage Regulator

The voltage regulator will convert the battery voltage to the voltage needed by each module. The voltage regulator must output 5V. We will use a generic L7805 5V linear voltage regulator.

2.2.5. Device Software

2.2.5.1. Microcontroller

The device's microcontroller will be programmed in software. A flowchart for the proposed software design is below.

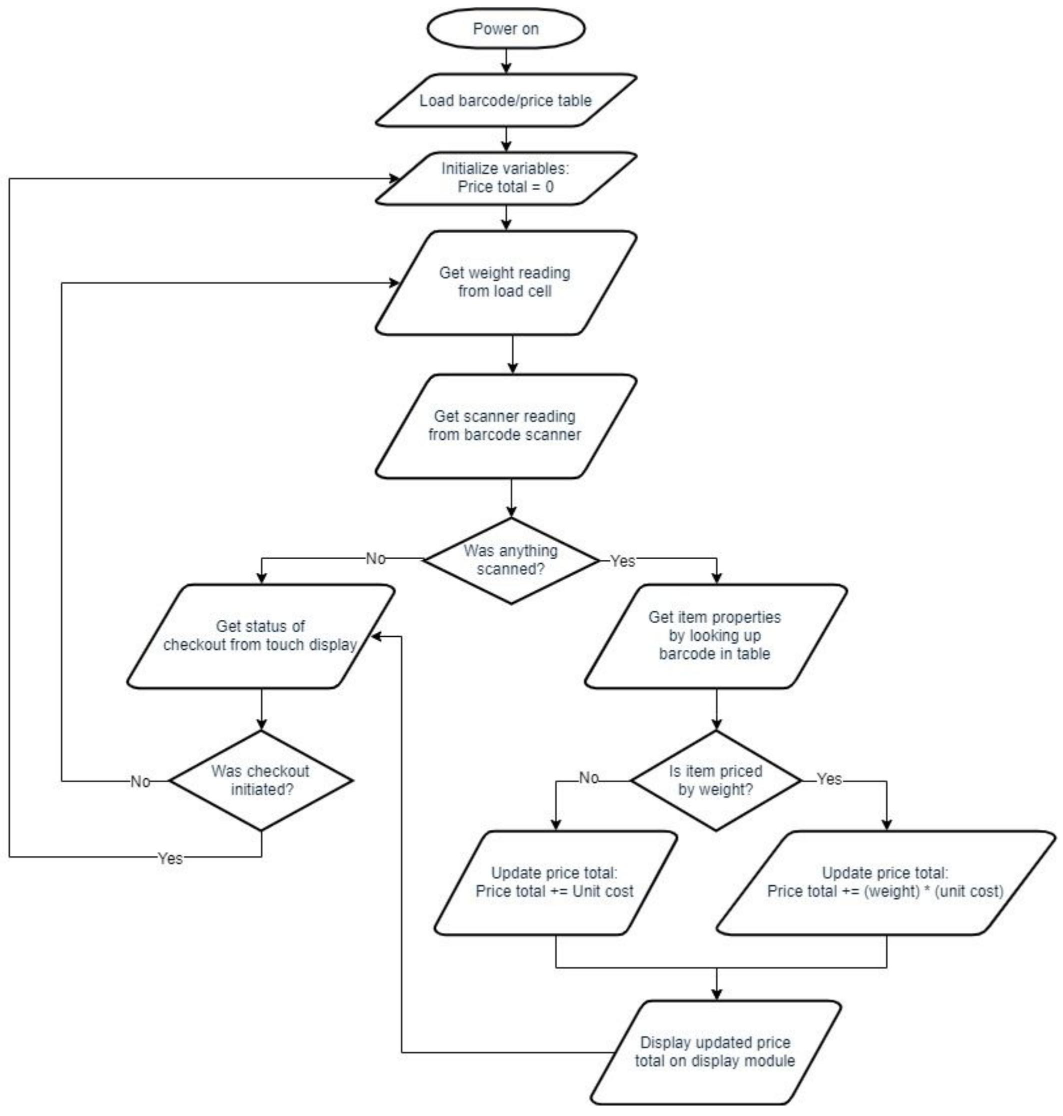

Power on
Load barcode/price table
Initialize variables:
Price total = 0
Get weight reading from load cell
Get scanner reading from barcode scanner
Was anything scanned?
No
Yes
Get status of checkout from touch display
Get item properties by looking up barcode in table
Was checkout initiated?
No
Yes
Is item priced by weight?
No
Yes
Update price total:
Price total += Unit cost
Update price total:
Price total += (weight) * (unit cost)
Display updated price total on display module

Figure 2.4. Flowchart for device software.

At a high level, the software will continuously check the weight and barcode modules, then update the display module when a product has been scanned.

2.2.5.2. Display Module

The touch screen features will be programmed in software. A flowchart for the proposed software design is below

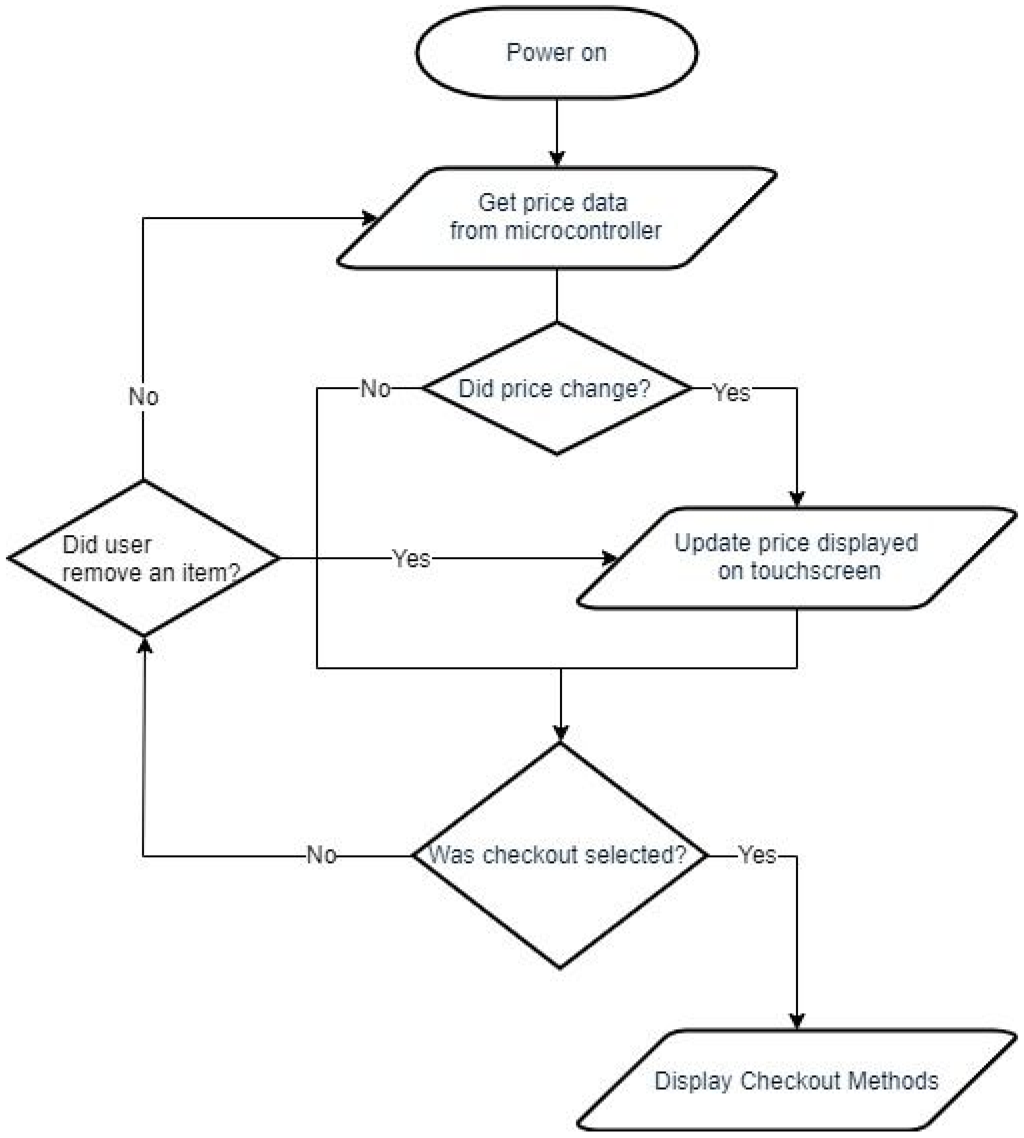


Figure 2.5. Flowchart for touch screen software.

At a high level, the touch screen software will continuously check the price data sent from the microcontrollers, update the price when new price data is received, and handle both removing and checking out components based on removal of an item and if checking out was selected.

2.3. Block Requirements

Module		Requirements	Verification
Scanner Module		1) Must correctly scan barcodes placed in front of it at least 9 times out of 10. 2) Scanning the object must take less than 3 seconds.	Place a barcode in front of the scanner. The scanner should detect the barcode in less than 3 seconds. Repeat 10 times. 1) The scanner should successfully decode the barcode at least 9 times out of 10. 2) The scanner should successfully decode the barcode in under 3 seconds on every trial.
Weight Module	**Load cell**	1) Must correctly report weight of object placed on it within +/-5% accuracy. 2) Reported weight value should fall within acceptable accuracy range in under 5 seconds. 3) Must correctly report weights of objects up to 10 lbs (~4.5 kg).	Place an object of known weight on the load cell. Examine the output of the weight module. 1) The object's weight should be reported within the required accuracy range. 2) The object's weight should be reported within the required time frame. 3) This process should work for objects weighing up to and including 10 lbs.
	Load cell amplifier	1) Must be able to interface the load cell's output with the controller module inputs.	1) Place an object of known weight on the load cell. Verify that the weight information is communicated to the microcontroller by examining an output feed from the weight module.
Display Module (Touch Display)		1) Must be able to receive output from and send input to the controller module. 2) Must be able to display the price and weight of the item being scanned/weighed 3) Must allow users to see products in their virtual cart and remove products as desired 4) Must be able to allow users to select their payment method (credit card, mobile payment, etc)	1) Turn on the touch display screen, and test communication between the screen and the microcontroller 2) Scan two items that the microcontroller recognizes; one to verify price by weight, and one to verify price by quantity. Both tasks must be able to be completed within 3 seconds 3) Check that the displayed price and weight is correct on the display. 4) Confirm that the microcontroller recognizes adding and removal via touch and a valid response to user payment is returned under two seconds.
Control-	**Micro-controller**	1) Must be able to connect to and support communications	1) Interact with the scanner module, weight module, and display module in a

ler Module		with the scanner module, weight module, and display module concurrently 2) Must be able to perform basic arithmetic, like keeping a running sum for price totalling 3) Must be able to keep data in memory, like a map from barcodes to prices	way that would verify they communicate with the microcontroller. (e.g. Perform module-specific verification steps and verify using microcontroller). This process must occur while all devices are connected to the microcontroller (e.g. Device has enough open pins to connect to all modules) 2) Perform a simple sum computation (e.g. price total from scanner module). Verify the output is expected. 3) Load the desired data structure into the device's memory. Verify that all members of the data structure are accessible.
Power Supply Module	**Battery**	1) Must provide enough power to support the power requirements of all components simultaneously, estimated at 3200 mW 2) Fully-charged battery must provide power for at least 4 hours when system is on standby	1) Turn on the corresponding modules within the system 2) Ensure that no module initially fails due to lack of power being supplied to the system 3) Perform actions with the system such as removing and adding the item and check if any system fails due to lack of power 4) Check if the system remains powered on for at least 4 hours on standby
	Voltage Regulator	1) Must take in battery voltage and convert if necessary 2) Must output 4.5 to 5.5 volts (0.5 volts tolerance)	1) Test the voltage regulator by supplying a variety of inputs (5-12 Volts) 2) Measure voltage output using an oscilloscope for the desired range of input voltages. Verify that, in all cases, the voltage output function lies strictly within the range of 4.5 to 5.5 V.

2.4. Tolerance Analysis

The purpose of the project is to be able to scan items and output price change and be able to checkout. Thus, the interface critical to the success of the project is the scanner module. Without the ability to scan, the project would be unable to function.

The benefits of the YHD-M800 scanner is that it has the capability to read incomplete and fuzzy code. Our requirements dictate that the scanner should have a low failure rate to provide a good customer experience. We quantify this as the scanner succeeding at scanning a product at least 9 out of 10 times (with the other 1 out of 10 being the scenario where it is unable to read the barcode). We will analyze this process mathematically and show that the scanner we selected meets these criteria.

First, let us analyze the probability that at least 9 out of 10 scans succeed. We derive an expression for this in Equation 2.4.1. We let $p_{>=9/10\ succeed}$ represent the probability that at least 9 out of 10 scans succeed, and $p_{succeed}$ represent the probability that a single scan succeeds. Then we have

$$p_{>=9/10\ succeed} = ((10\ C\ 9)\ (p_{succeed})^{9}(1-p_{succeed})^{1}\ + (10\ C\ 10)(p_{succeed})^{10}(1-p_{succeed})^{0})$$

Equation 2.4.1

We want to be virtually certain that 9 out of 10 scans will succeed, so we set $p_{9/10\ succeed}\ =\ .99999.$ Substituting this in to Equation 2.4.1, we solve for $p_{succeed}$, and demonstrate the result in Equation 2.4.2:

$$p_{succeed} = .999528$$

Equation 2.4.2

Interpreting this result, this means that a single 3-second scan must succeed roughly 99.9528% of the time to meet our desired success rate. We interpret to mean that we must be virtually certain that a single 3-second scan will succeed. Below, we will show that the device meets this criterion.

The YHD-M800 scanner scans 300 times per second. Assuming we scan for 3 seconds, there will be 900 frames in a single scan. For a single 3-second scan to succeed, only 1 out of 900 frames must be successfully scanned. We present Equation 2.4.3 below, relating the probability that at least 1 out of 900 frames is scanned ($p_{succeed} = p_{>=1/900\ succeed}$) to the probability that a single frame is successfully scanned (p_{frame}):

$$p_{succeed} = p_{>=1/900\ succeed} = 1 - p_{0\ scans\ succeed} = 1 - ((900\ C\ 0)\ (p_{frame})^0 (1 - p_{frame})^{900})$$
Equation 2.4.3

Substituting $p_{succeed} = .999528$ into Equation 2.4.3 and solving for p_{frame} yields Equation 2.4.4:

$$p_{frame} = .00847$$
Equation 2.4.4

Interpreting this result, this means that to get our desired result of 99.85% success on a single 3-second scan, the device success rate per frame must be at least 0.847%.

Now, the success rate per frame for our specific device is not provided. However, we see no reason to suspect that the success rate for the device we selected deviates substantially from the industry standard for barcode scanner error rates. A survey of barcode scanner devices from the same manufacturer and other manufacturers suggests that a typical error rate per frame for a barcode scanner is on the order of 1 in 1,000,000. This corresponds to an industry standard success rate of 99.9999%. This is substantially larger than the 0.847% success rate required to meet our objective. Barring unprecedented deviation from the industry standard, we conclude that the device will meet our success criteria.

To summarize this analysis, we wanted to analyze how likely it is that we would meet the scanner module requirement that scanning a product would succeed 9 times out of 10 at a rate of 99.999%. Under these constraints, we computed the necessary success rate for a single 3-second scan to be $p_{succeed} = .999528.$ Based on this value and the scanner specifications, we determined for 1 out of 900 frames must be successfully scanned, the probability that a single frame would be scanned successfully must be at least $p_{frame} = .00847.$ Finally, we determined that any barcode scanner device has a success rate per frame immensely larger than this value. We conclude that we are well within our tolerance for verifying this requirement.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor Cost

We present estimates for the cost of labor below.

$$\frac{\$77{,}653}{year} \cdot \frac{1\ year}{52\ weeks} \cdot \frac{1\ week}{40\ hours} = \$37.33 \text{ hourly salary}$$

Equation 3.1

Equation 3.1 computes the hourly salary we would expect to pay an engineer working on the project. The yearly salary indicates the average starting salary for a Computer Engineering graduate from the University of Illinois [1].

$$\frac{\$37.33}{hr} \cdot 2.5 \cdot \frac{10\ hr}{week} \cdot 16\ weeks = \$14,932 \text{ per engineer}$$

Equation 3.2

Equation 3.2 gives the total cost of labor for an engineer working on the project for 10 hours every week for the entire semester.

$$\$14,932 \cdot 3\ engineers = \$44,796 \text{ total cost of labor}$$

Equation 3.3

Equation 3.3 gives the total cost of labor for the project. This is based on us having 3 engineers on the team.

3.1.2 Parts Cost

Description	Manufacturer	Part Number	Quantity	Unit Cost ($)	Total Cost ($)
Barcode Scanner	YHDAA	YHD-M800	1	23.00	23.00
Load Cell	SparkFun Electronics	SEN-10245	1	10.95	10.95
Load Cell Amplifier	SparkFun Electronics	HX711	1	12.95	12.95
Touch Screen Display	Adafruit	TFT Controller: ILI9341 TFT: DT280QV10-CT	1	50.00	50.00
Microcontroller	Microchip	ATMEGA644A-AUR	1	4.42	4.42
Battery	TalentCell	PB240B1	1	33.99	33.99
Voltage Regulator	SparkFun Electronics	L7805	1	0.95	0.95
Total					136.26

3.1.3 Total Cost

$$Total = Labor + Parts = \$44,796 + \$104.80 = \$44,900.80$$

Equation 3.4

Equation 3.4 gives the total cost for the project, computed as the value computed by Equation 3.3 plus the total cost of parts to develop the project.

3.2 Schedule

Week	Timothy	Lucas	Andrew
9/30 - 10/6	Prepare Design Document	Prepare Design Document	Prepare Design Document + PCB Schematic
10/7 - 10/13	Quality test power supply	Create physical scale for weighing objects	Prepare PCB layout
10/14 - 10/20	Install power supply onto final board	Develop software to interact with weight module	Prepare PCB layout
10/21 - 10/27	Wire up touch screen display device	Create physical scanner device	Develop software for microcontroller to interface with other components
10/28 - 11/3	Develop software to checkout, remove, and add objects	Develop software to scan objects	Develop software for microcontroller to interface with other components
11/4 - 11/10	Soldering	Soldering; Develop software to integrate scale/scanner	Soldering
11/11 - 11/17	Final touches on touch screen integration	Final touches on scale/scanner integration	Final touches on microcontroller integration
11/18 - 11/24	Prepare for mock demo	Prepare for mock demo	Prepare for mock demo
11/25 - 12/1	Relaxing	Relaxing	Relaxing
12/2 - 12/8	Prepare for final demo	Prepare for final demo	Prepare for final demo
12/9 - 12/15	Prepare for final demo + paper	Prepare for final demo + paper	Prepare for final demo + paper

4. Ethics and Safety

During the development of the project, we shall follow #7 on the IEEE Code of Ethics - we will listen to instructors, classmates, and other individuals who give advice and criticism on the details of our technical implementation. This feedback ensures that the project brings quality to the problem at hand and does not degrade [4]. this transparency ensures that #9 on the IEEE Code of Ethics will be followed and consumer reputation will be protected.

There are a number of ethical issues concerning the completed project. We must correctly report the price to the consumers and not misconstrue the price in light of #3 on the IEEE Code of Ethics [4].

One major ethical concern involves theft associated with self-checkout systems. People could misuse the product to commit theft, either intentionally or absentmindedly. While this is an ethical issue, it has financial implications for any store using this technology. A study by criminologists at the University of Leicester found that self-checkout services led to a loss rate of 4% of the total value of purchases, or a 122% increase over the average loss rate [1, 5]. We note these concerns to anyone seeking to use this product in accordance with the IEEE Code of Ethics, “to be honest and realistic in static claims or estimates based on available data.”

This project shares a number of safety concerns with all consumer electronics. We need to take steps to ensure the device doesn’t fail in a catastrophic way and consequently injuring customers using it. This concern means we need to be extremely cautious in every step of the design. We need to make sure the battery isn’t being overdrawn. We also need to make sure the user can’t injure themselves, either through an electrical fault or a sharp edge on the device.

Developing this device also poses safety issues to us as its developers as well. Generally, these safety concerns are mitigated by our years of experience working in ECE labs, but we still have to take precautions. For instance, soldering the PCB can be dangerous if it is not done properly. We would lessen this risk by reviewing proper soldering procedures before performing any soldering. Holistically, we understand that following defined safety procedures is the best way to develop this product effectively.

Extra supplemental files:

Battery safety: https://www.ehs.washington.edu/system/files/resources/lithium-battery-safety.pdf

5. References

[1] Beck, A. and Hopkins, M. (2016). Mobile Scan and Pay Technology could promote supermarket theft, study suggests. [online] University of Leicester. Available at: https://www2.le.ac.uk/offices/press/press-releases/2016/august/mobile-scan-and-pay-technology-could-promote-supermarket-theft-study-suggests [Accessed 17 Sep. 2019].

[2] “Become an Electrical Computer Engineer”. [online] University of Illinois. Available at: https://ece.illinois.edu/admissions/ece-majors.asp [Accessed 25 Sep. 2019]

[3] Day, M. and Morley, K. (2018). Amazon's new store without checkouts doesn't quite Go to plan. [online] Stuff. Available at: https://www.stuff.co.nz/business/world/100796913/amazon-go-store-with-no-checkouts-opens-to-the-public-in-seattle [Accessed 14 Sep. 2019].

[4] Hern, A. (2017). Amazon's checkout-free physical shop 'can't cope with more than 20 people'. [online] The Guardian. Available at: https://www.theguardian.com/technology/2017/mar/29/amazon-go-checkout-free-physical-shop-delayed-camera-sensor-people [Accessed 14 Sep. 2019].

[5] Ieee.org. (2019). IEEE Code of Ethics. [online] Available at: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 14 Sep. 2019].

[6] Mele, C. (2019). Self-Service Checkouts Can Turn Customers Into Shoplifters, Study Says. [online] Nytimes.com. Available at: https://www.nytimes.com/2016/08/11/business/self-service-checkouts-can-turn-customers-into-shoplifters-study-says.html [Accessed 17 Sep. 2019].

[7] Verhage and S. Soper, "Amazon Could Spend $3 Billion on ‘Go’ Stores, Analyst Says", Bloomberg.com, 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2018-09-20/amazon-could-spend-3-billion-on-go-stores-analyst-says. [Accessed: 14-Sep-2019]