# Team 26: Zipcart

Midway Design Review

### Team



Ryan Lagasse



#### Ricardo Henriquez



Jonathan Azevedo





## Block Diagram



## Challenges

- Scan barcodes of items as they enter the cart
  - Shopper doesn't have to orient items for camera
- Process images for barcodes in a timely manner
  - System has real-time requirements and a heavy workload
- Get product information quickly without maintaining a database
  - Request from external API and cache responses
- Use mechanical energy to keep system powered
  - Maintainer doesn't have to charge carts at the end of the day

### Project Requirements

- 1. Recognize barcode as item is placed in cart
- 2. Detect when item is removed from cart
- 3. Display item list and current balance
- 4. Detect an unscanned item to prevent theft
- 5. Sustain power for a full business day

### Specifications

- 1. One item entered or removed per two-second interval
- 2. Barcode surface must be reasonably flat
- 3. Maximum system latency of four seconds
- 4. Eighteen hours of continuous operation

### MDR Deliverables

- 1. Detect barcode around front-face of camera perspective
- 2. Update cloud database with product information
- 3. Successful integration of feedback system
- 4. Demonstrate power generation using stepper motor

# Power

Ricardo

### Power Circuit Diagram



### Stepper Motor + Simulation

- Stepper motors are best at low speeds
- Motor: 2-Phase rated at 6V and 0.8A
- Arduino + LM298 motor driver simulates movement of shopping cart
   ➢ Max. speed 200RPM ~ 3mph
- Plot shows relationship between the current produced versus speed
  - ≻175mA produced at maximum speed using <u>one</u> motor





### Full Wave Rectifier

- Schottky Diode (1N5818) for smaller forward voltage drop (0.55V @ 1A)
- Each phase is rectified which is necessary to convert AC into DC
- Smoothing capacitor helps reduce voltage variations
- DC output voltage is proportional to RPM



## Pololu Voltage Regulator (S7V7F5)

- Capability to step-up/step-down input voltage
- Ideal choice since voltage generated varies
- Efficiently produces 5V from input voltages between 2.7V and 11.8V
- Sources up to 1.6A
- Plot shows regulators efficiency of sourcing current at different input voltages
  - Efficiency depends on voltage produce





### Adafruit Powerboost 1000C

- Built-in load-sharing battery charger circuit allows Raspberry Pi to run while recharging the battery
- Built-in battery protection circuit
- Charger circuitry can recharge at a max rate of 1000mA
- 2A internal switch allows battery to output more than 1000mA if necessary



### Pi Power Consumption & Improvements

- Standby → 2.5W, 5V, 500mA
- All Peripherals  $\rightarrow$  4W, 5V, 800mA
- Increase power delivered to the system
  ➢ Wire 4 motors in parallel, one per wheel, to increase current generated
  ➢ Test with different motors

# Optics & Detection

Jonathan

### Overview

Optics

• Captures item with UPC code in camera frame

Software

• Detects barcode and extracts UPC code to send to AWS database

## Optics

Optic system scans objects as they enter the cart

### 1. Camera

- Main component which captures video of items entering cart
- Barcodes must be in cameras FOV and directed at it to be detected

### 2. Mirrors

• Reflects barcode to direct view of camera independent of orientation

### Camera

### Camera requirements

- Adjustable focus
- Good resolution
- Compatible with software libraries

### Kuman Camera Specs

- 5 MP OV5647 sensor
- 2592 x 1944 pixel static images
- 1080 p @ 30 fps, 720 p @ 60 fps
- Adjustable focus lens



### Software

Important libraries

• OpenCV • PyZBar

#### Procedure

- 1. Creates order in cloud
- 2. Detects barcode in video stream
- 3. Extracts UPC code from barcode





### Status

- Orders are created in database from the Pi
- Can detect barcodes within two-second window
- Sends UPCs to order database
- Detects barcodes at max distance of four inches



### Difficulties

- Focus is a limiting factor in our Pi cameras
  - No autofocus feature
- Tested multiple products, few alternatives for serial interface
  - Most use same OV5647 Sensor

## Remaining Work & Considerations

- USB Cameras
  - More alternatives
  - Better quality cameras
  - Autofocus features
- Intel Movidius Machine Learning
  - VPU- increase vision and complex processing with low-power
  - Speed up detection of items and barcodes
- Taking still images
  - Easier to process then video stream





# Cloud Infrastructure

Ryan

### Overview

- Request passes through **API Gateway**
- Lambda function fulfills the request
  - Interacts with **DynamoDB** database
  - Conditionally requests info from Barcode API



### Procedure: Add or Remove Items in an Order

- 1. Parse HTTP Request. Get Order ID, UPC, Action to Perform
- 2. If UPC in Cache:
- 3. Retrieve product information
- 4. Else:
- 5. Request product information
- 6. Store information in Cache
- 7. Add or Remove UPCs in Order with proper ID

Latency – Not Cached 2.446 seconds



ltem -	UPC	Order	ID
	Name Price		ltem List
			Balance
			Time Created
	Time Updated		Time Updated

### Barcode API



#### Request

GET /products?barcode=778554152253 HTTP/1.1

#### Response

### Remaining Work & Improvements

- Integrate AWS with user interface
  - Update the application on a subscription-basis from AWS
- Automatically update stale, cached item information
- Unit and integration tests with a mock database

# User Interface

Jonathan

### Feedback

- Current State
  - Using Adafruit DotStar Digital LED Strip, 60 LED/Meter
  - LEDs flash green on successful detection

### • Plan

- LEDs flash red on unsuccessful detection
- Signal malicious attempts to steal items



### Android

- The world's most popular mobile OS
- Create applications in Java or C++ with Android Studio IDE





### Interface Specifications

- Start a new order
- View balance and list of items in the order in near-real time
- Process payment
- Complete transaction

### CDR Deliverables

- 1. Mount system on a shopping cart
- 2. Detect barcodes fully around products
- 3. Remove items as they exit the cart
- 4. Increase power delivered to system
- 5. Create PCB for the power circuit
- 6. Make a fully-featured interface

## Budget





LED Strip

## Gantt Chart



# End of Presentation

Questions

# Demonstration

Jon & Ryan: End-to-End Barcode Scanning Ricardo: Power Delivery

