Zipcart

2019 Cornell Cup Semifinalists

Ryan Lagasse, Ricardo Henriquez, Jonathan Azevedo

Professor Tilman Wolf

Abstract

Checkout aisles in grocery stores are inefficient.

There are better ways to process transactions that do not require shoppers to wait in line for service.

Zipcart speeds up this process using a self-service "scan as you go" approach.

A mounted camera processes the barcodes of items as they enter the cart.

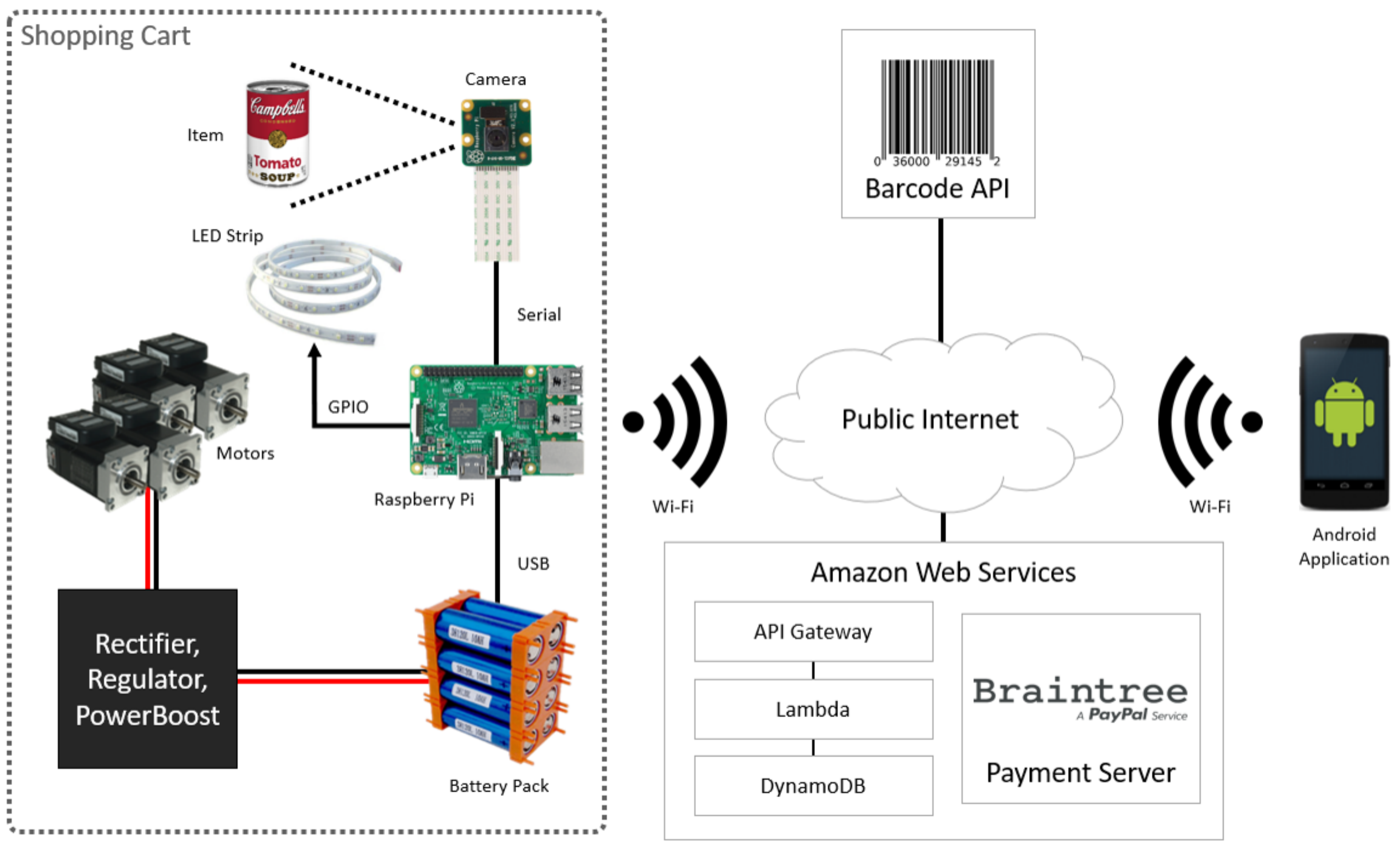
Shoppers can audit their orders, balance, and complete their purchase from our smartphone app.

Zipcart's self-service approach empowers shoppers, decreases wait times, and provides cost savings to store proprietors.

Specifications

Specification	Goal	Actual
Power Generation	2.4 W	≈ 3.1 W
Continuous Operation	18 hours	24 hours
Maximum Detection Range	32"	20"
Scan-to-UI Latency	4 sec	≈ 2.5 sec (with cache)
Display Order Info to User	✓	✓

Block Diagram

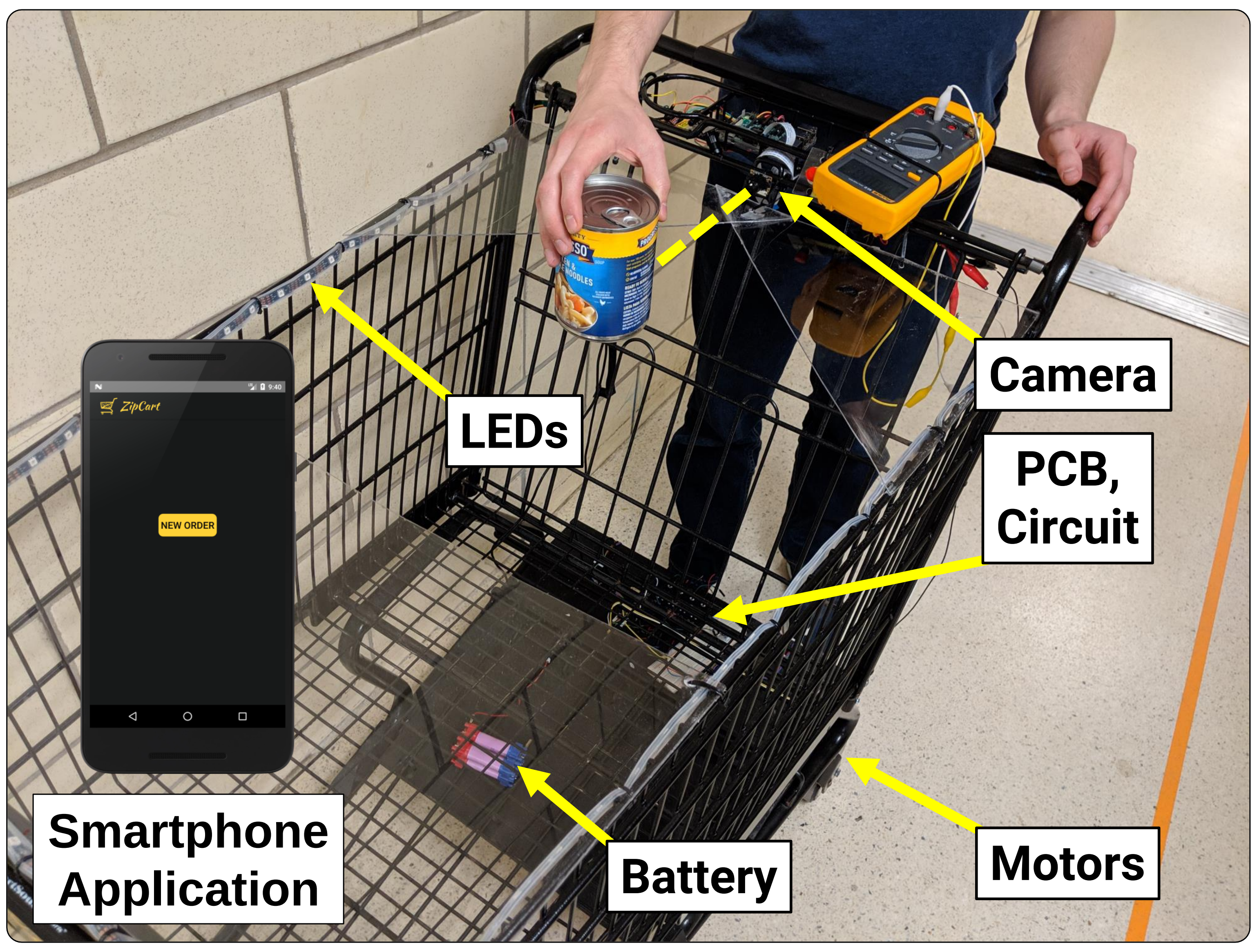


Lambda: "serverless" code platform; manages database

DynamoDB: database platform; stores orders, inventory

Barcode API: to dynamically generate product inventory

System Overview



Results

Detection Status vs. Item Entry Speed			
	Still	Slow	Normal
Correct Barcode	98%	64%	26%
Incorrect Barcode	2%	0%	0%
No Detection	0%	36%	74%
Total Trials	50	50	50

Acknowledgements

Advisor

Professor Tilman Wolf

Evaluators

Professor Zlatan Akšamija

Professor C. Mani Krishna

SDP Faculty

Dean Christopher Hollot

Professor T. Baird Soules

Shira Epstein

ECE Department

Professor Robert Jackson

Professor Dennis Goeckel

Professor David Irwin

MIE Department

Professor Bernd Schliemann

SDP Laboratory Technicians

Francis Caron

Keith Shimeld

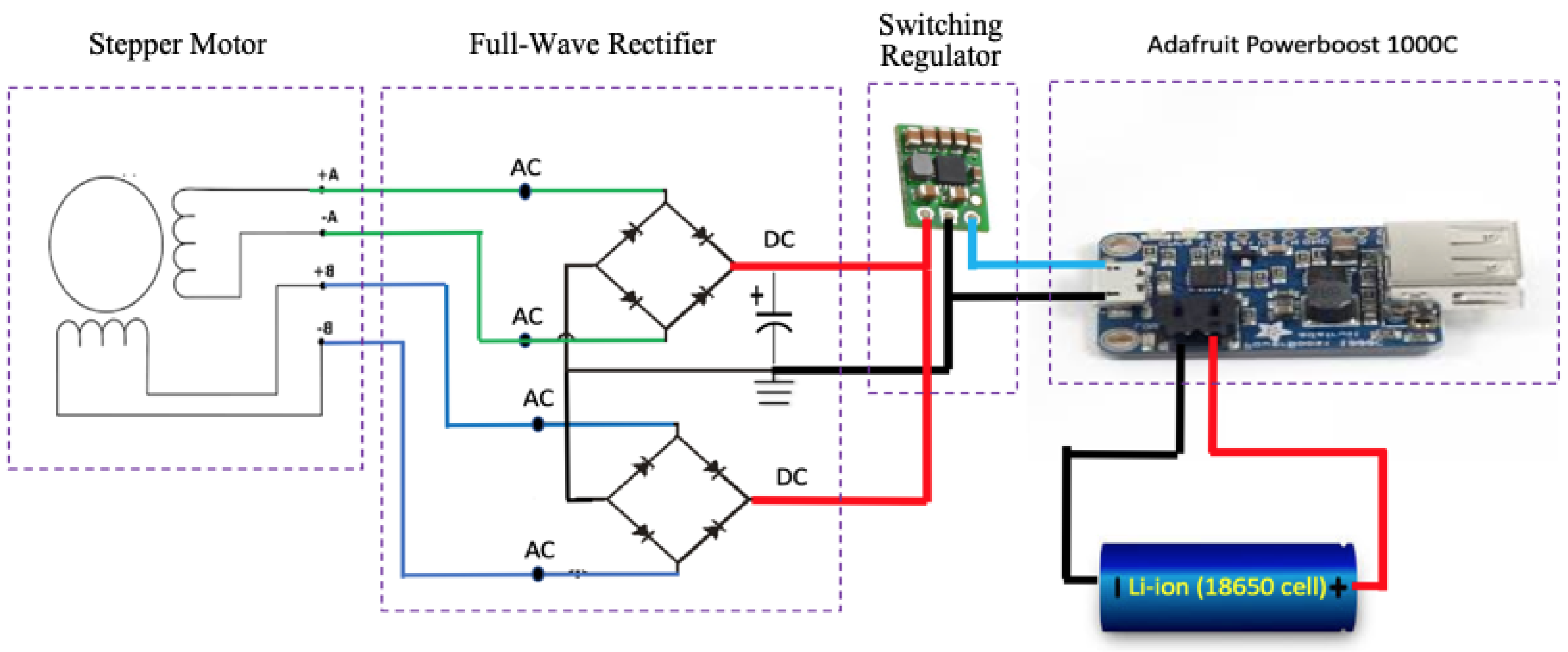
Power Generation

REQUIREMENTS

- Minimum power generation of 2.4W

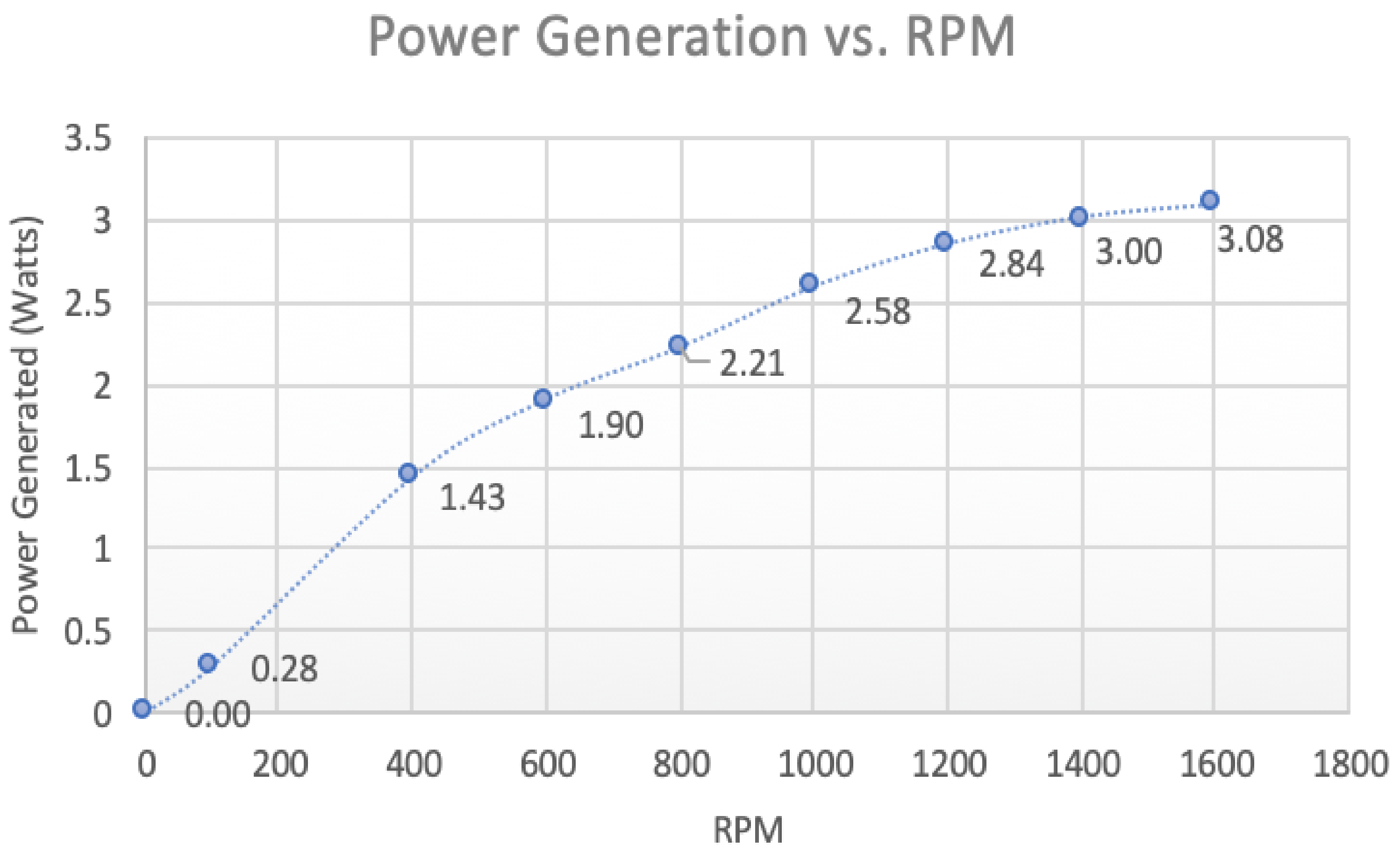
DESIGN

- Custom internal and spur gear (8:1 gear ratio)
- Stepper motor driven by wheel
- Mechanical motion charges battery
- Battery for continuous operation
- PCB can wire up to four motors in parallel



RESULTS

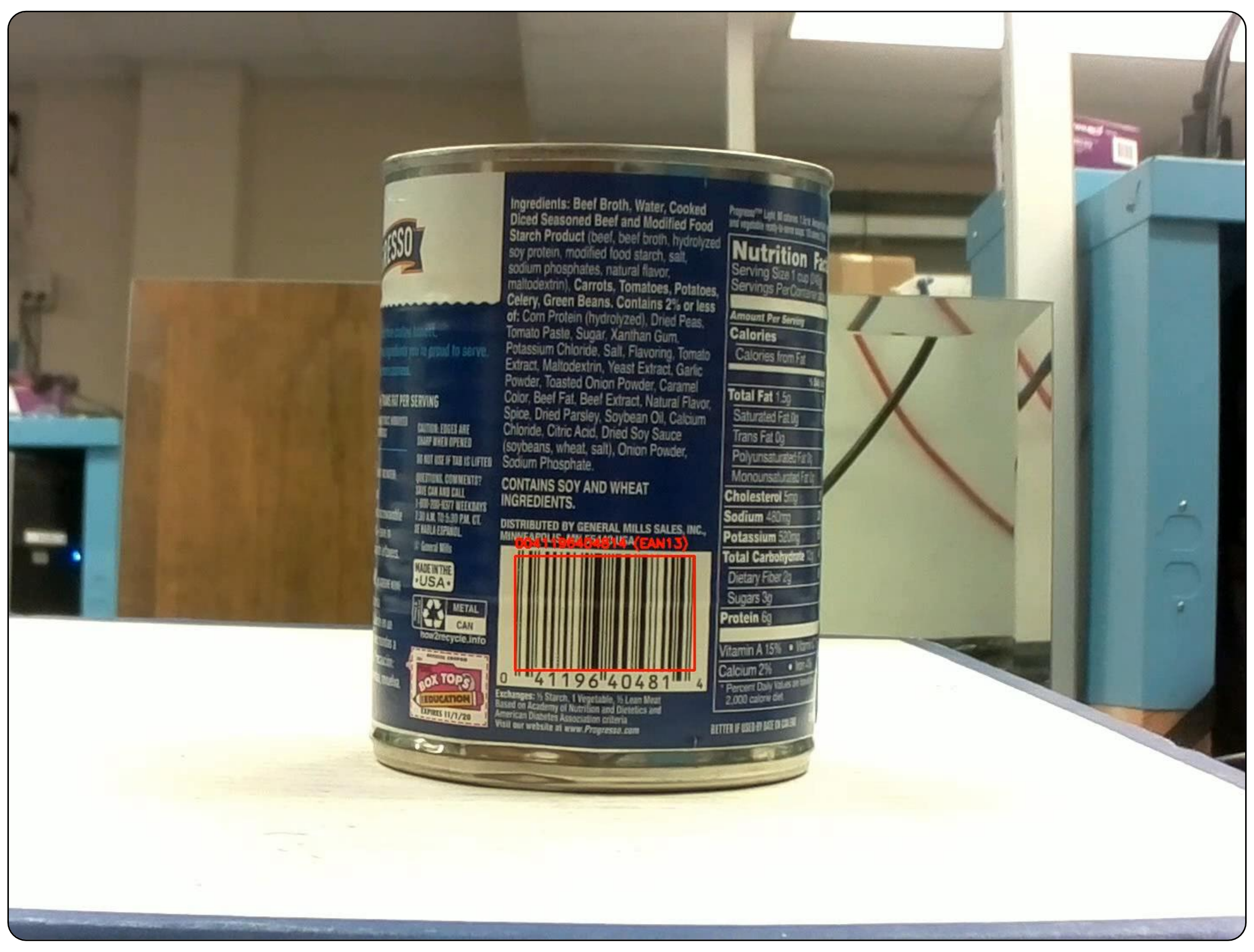
- Amount of power generated depends on speed
- 3W generated at average walking speed



Cost

Part	QTY	Development	Production
Shopping Cart	1	$62.99	$58.49
Raspberry Pi 3	1	$35.68	$35.00
Camera	1	$25.00	$22.50
Ribbon Cable	1	$3.95	$3.16
Stepper Motor	2	$58.62	$44.00
Adafruit Powerboost 1000C	1	$19.95	$15.96
Switching Regulator	1	$14.95	$12.93
Samsung Li-Ion 18650 Cells	4	$15.96	$11.00
PCB	1	$1.00	$0.77
Schottky Diode	16	$7.68	$2.72
Push Buttons	2	$2.18	$1.36
Voltage Level Shifter	1	$2.95	$2.51
Total		$250.91	$210.40

Detection

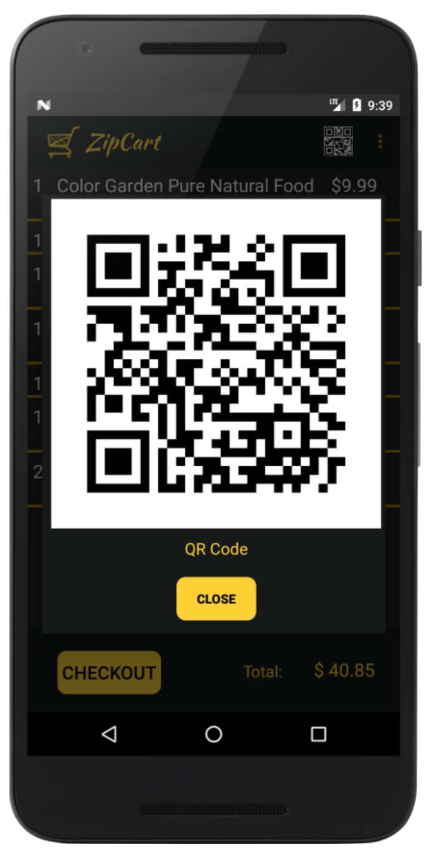


- Detection module implemented in C++ using the OpenCV and ZBar libraries
- Reads the barcodes of items to be sent to AWS for order processing

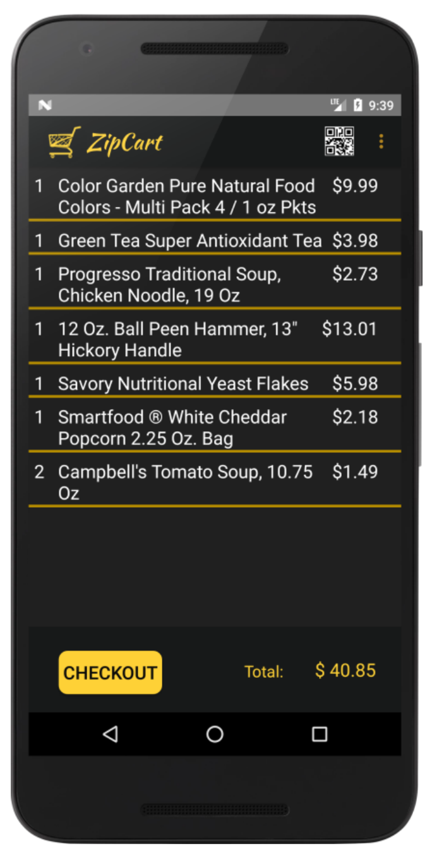
OpenCV

Max Frames Processed/Second by Platform	
Platform	Maximum FPS
Raspberry Pi 3 Model B	5.35
Dell Inspiron i5 Laptop	15.51

Android Application

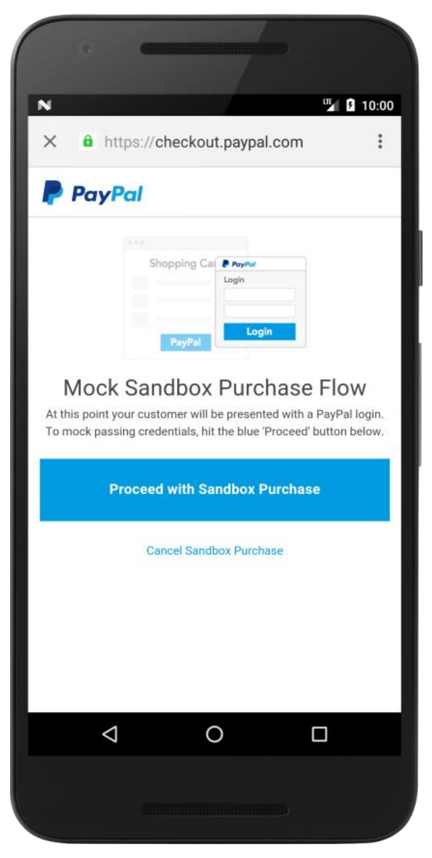


QR Popup Screen



1 Color Garden Pure Natural Food Colors - Multi Pack 4 / 1 oz Pkts $9.99
1 Green Tea Super Antioxidant Tea $3.98
1 Progresso Traditional Soup, Chicken Noodle, 19 Oz $2.73
1 12 Oz. Ball Peen Hammer, 13" Hickory Handle $13.01
1 Savory Nutritional Yeast Flakes $5.98
1 Smartfood ® White Cheddar Popcorn 2.25 Oz. Bag $2.18
2 Campbell's Tomato Soup, 10.75 Oz $1.49

Item View Screen


10:00
https://checkout.paypal.com
PayPal
Mock Sandbox Purchase Flow
Proceed with Sandbox Purchase
Cancel Sandbox Purchase

At this point your customer will be presented with a PayPal login. To mock passing credentials, hit the blue 'Proceed' button below.

App Payment Screen

App Usage

- App generates a QR code encoding an order ID, per order
- Camera scans QR code to synchronize the systems
- Shopper can see the state of the order in near-real time as they add and remove items
- To complete the transaction the user selects "Checkout" and payment option

Payment Server

We use a service called Braintree to process payments.

Supported options include:

- Credit Card
- Apple Pay
- Google Pay
- PayPal
- Venmo

Our payment server accepts and processes all user payment requests; it is hosted on AWS as an EC2 instance.