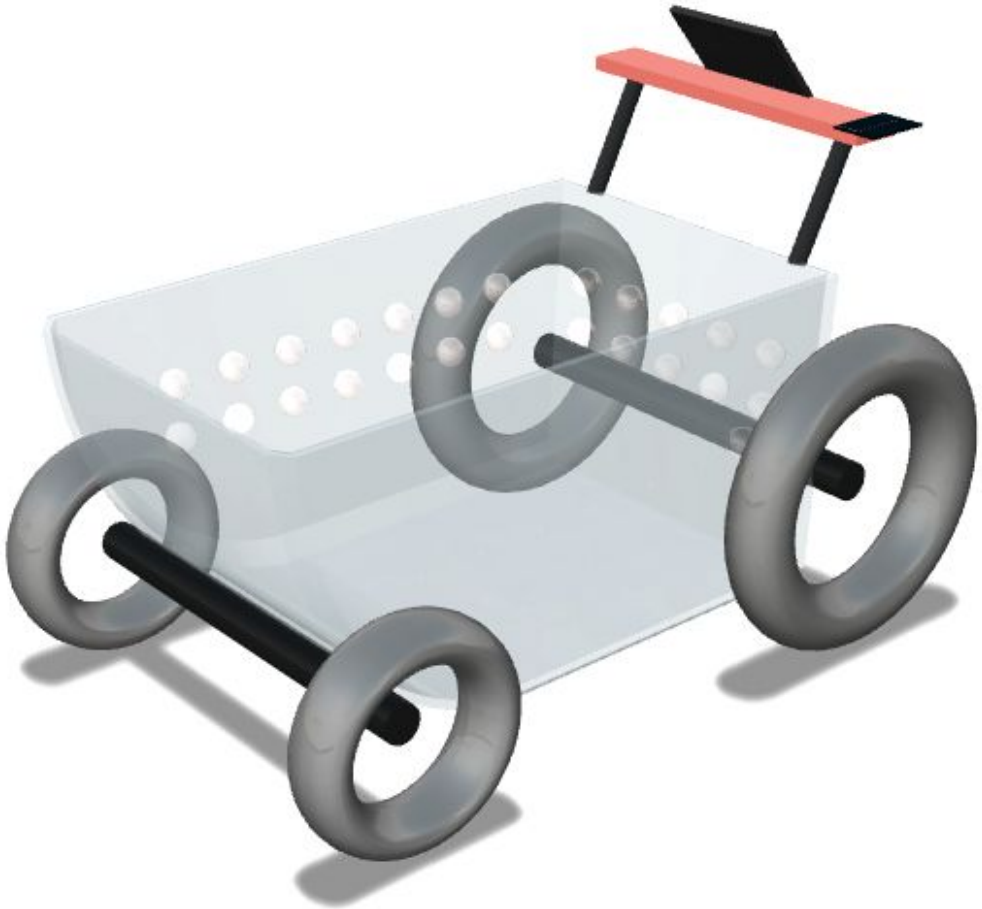
Smart·**Cart**

Bijoy Singh 120050087
Ranveer Aggarwal 120050020
Manik Dhar 120050006
Nishanth Koushik 120050041



Introduction

Shopping is something we spend a lot of our time doing.

How many of us have had the frustrating experience where even when purchases a few items, we have to stand in long queues.

So we created a smart cart, which lets to securely add content which you wish to buy and purchase from your phone itself.

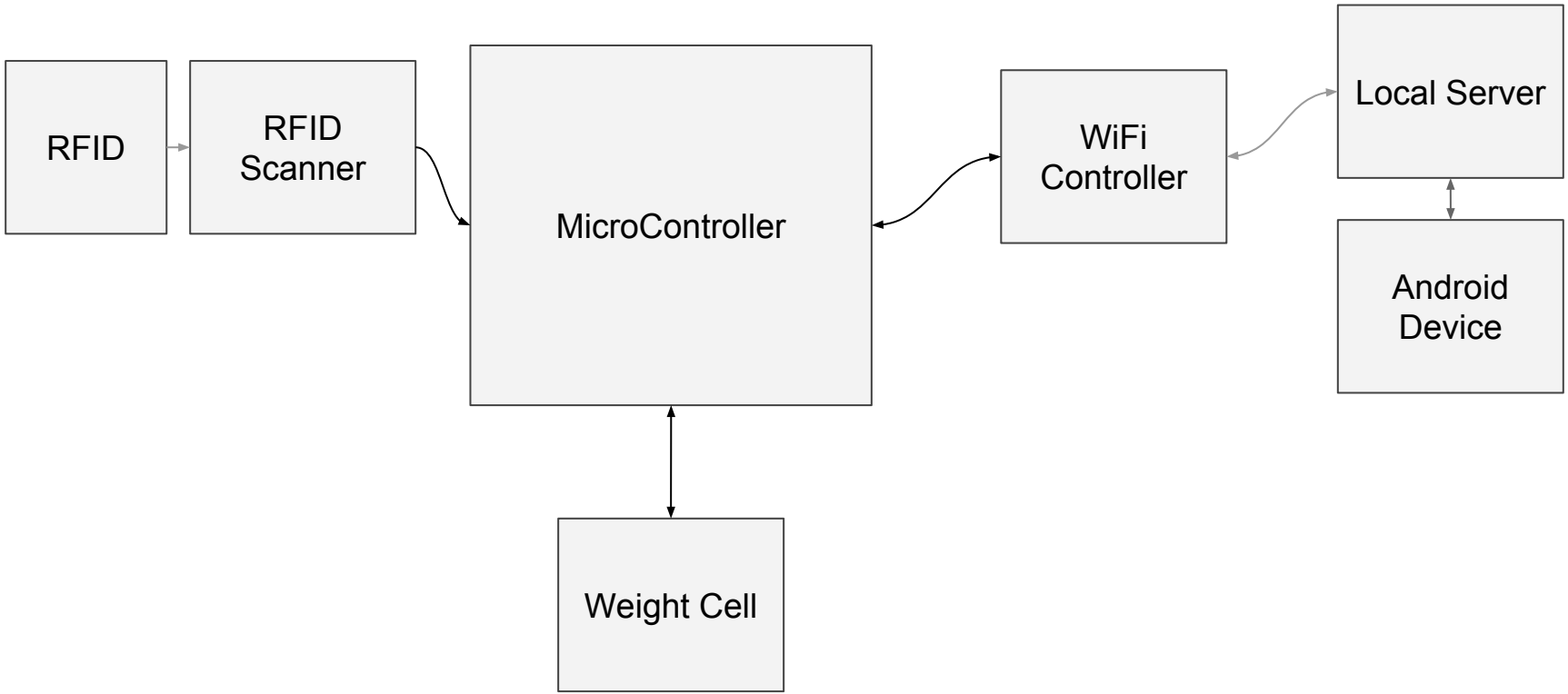
We present the SMART CART project. #NoMoreLines

Problem Statement and Critical Tasks

The SMART CART will let you create a purchase by doing the following:

- Phone scans the RFID card of your cart
- The Microcontroller now gets assigned to your order
- You can order Content my tapping an RFID card to the RFID sensor of the cart
- Your phone should detect this item
- Your cart should now have the weight of the cart and send information till it detects changes.
- The server maintains if the cart weight is indeed correct or not.
- Tapping multiple times increases the count.
- The count can be changed in the application
- You can login using Google Account

System Architecture



Work Division

Android App: Bijoy and Nishanth

Server: Manik and Ranveer

Tiva: All together (RFID, weight cell and WiFi)

Wifi: 22nd March

RFID: 22nd March

Android/Server/TIVA: 4th April (showed partial on 23rd)

Weight Cell: 4th April

Innovation and Challenges

A number of challenges were faced while making this project.

- As the chip has a number of components connected to the same TIVA chip, we realized that the basic configurations of the chip/library lead to conflicts. To tackle this we read through most of the base code of the Energia, and googled. We found and read about MISO and MOSI, and about Chip Select works, hence we modified the switch select pins of the devices to get them to work.

- Further, we realized about how we can optimize the experience of the users, previous iteration was a different control flow, and we fixed it to remove the LCD display thereby increasing the battery life and removing the lag in the initial setup.

- We of course also read through a lot of Energia documentation to understand how to use libraries and utilities.

Task Completed:

Wifi, Tiva, RFID: We used the ENERGIA library. Had to read up a lot of documentation to resolve conflicts caused by the configuration of the chip/library. To tackle this we read through most of the base code of the Energia, and googled. We found and read about MISO and MOSI, and about Chip Select works, hence we modified the switch select pins of the devices to get them to work.

Changing the design: We realized the LCD was not required and that it would be a drain on the battery. We removed the display in our design and instead used an android app.

Testing

We tested our product by using 2 RFID tags to represent two different products. They were calibrated to two different weights. Using the app we brought a combination of these items and changed the weight on the load cell to verify the system is working.

We checked the server to verify that the wifi communication is working.

Performance Metric

The time in which the android app confirms additions and removals to the cart is the most critical metric.

The RFID and the weight cell responds in less than a second and the app is updated every 5 seconds.

Features

The code is written using the Energia library for TIVA. The server is written using Django.

The hardware is easy to assemble.

Future Possibilities

Integrating the hardware on the actual cart.

Charging of the power supply.

Using beacons to help the customer navigate the store.

Improving the weighing cell.

Using a dedicated controller instead of Tiva.

References

Energia : http://energia.nu/

Android Studio : http://developer.android.com/sdk/index.html

Django : https://www.djangoproject.com/

PyCharm : https://www.jetbrains.com/pycharm/

Thank You