

# FROGSTAR



SENIOR CAPSTONE PROJECT 2013-2014  
05/01/14

# Team



- **Stockton Ackermann**
  - Documentation Lead and Android Application Developer
- **Nicholas Capurso**
  - Project Lead, Android Application Developer, and Network Engineer
- **Eric Elsken**
  - Technical Lead and Android Application Developer
- **Myrella Garcia**
  - Website Developer and Media Manager
- **Casey Stephens**
  - Android Application Developer
- **David Woodworth**
  - Testing Lead, Network Engineer, and Website Developer

# Overview



- Project Background
- Project Support Environment
- System Architecture
- FrogStar Application
- Networking
- Demos
- Challenges and Lessons Learned
- Future Work

# Project Background



- **Pre-existing systems.**
  - OnStar and other proprietary systems.
    - ✦ Accident detection.
    - ✦ Roadside assistance.
    - ✦ Contacts emergency services.
  - Expensive and not widely available.
- **Project FrogStar**

# Project Description



- **Proof of concept.**
- **Accident detection.**
  - Uses smartphone and TI SensorTag readings to detect accidents.
  - Redundant computations between smartphone and on-board control unit (OBCU).
- **Ability to contact emergency services.**

# Texas Instruments SensorTags



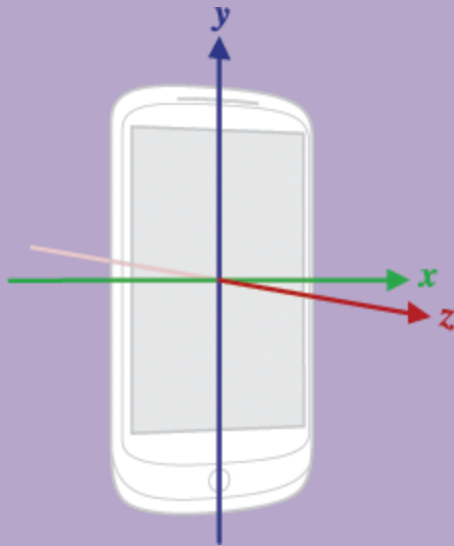
- Portable.
- Inexpensive.
- Bluetooth LE capability.



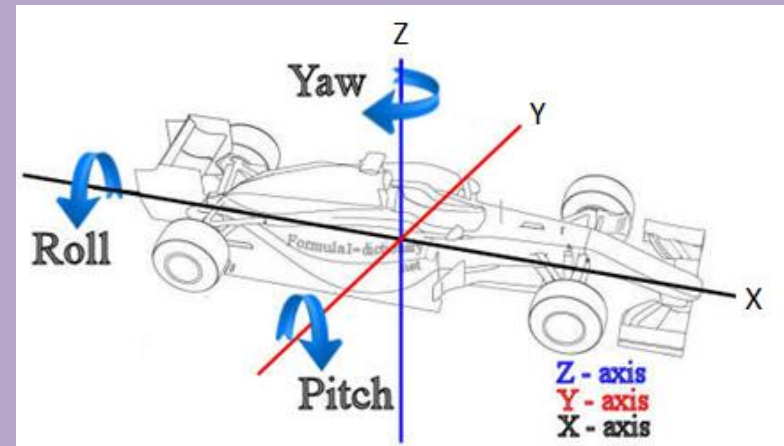
# Texas Instruments SensorTags



- Accelerometer - a device that measures acceleration, or the rate at which speed changes.



- Gyroscope - a device that measures orientation in terms of yaw, roll, and pitch.

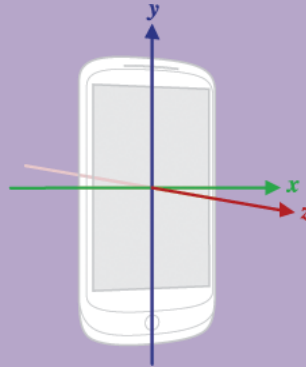


# Types of Accidents



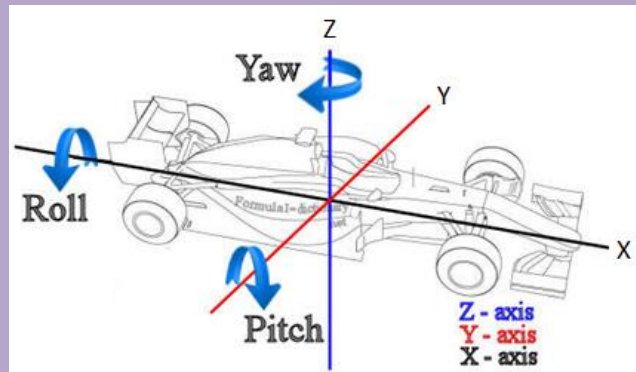
- Accelerometer-based accidents.

- Head-on.
- Rear-end.
- T-Bone.



- Gyroscope-based accidents.

- Spin-out.
- Rollover.





# Why Android smartphones?



- Variety of sensors.
- Bluetooth capability.
- NFC capability.
- Open Source Platform.
- Easy transition for our developers.



# Use of NFC Tags



- Android-compatible method for exchanging small pieces of data.
- Convenient way to start and shutdown the system.
- Re-programmable data storage.
- Holds MAC addresses (networking information).



# OBCU - Raspberry Pi



- On-Board Control Unit (OBCU).
- Advantages:
  - Inexpensive.
  - Storage via SD Card.
  - Expandable via USB.
    - ✦ Bluetooth.
- Offloads SensorTag querying from smartphone.
- Runs Linux.
- Powered by vehicle.



# System Environment



## Programming Environment

- Android Development Toolkit (2.2) Plugin for Eclipse.
- Eclipse: Kepler (4.3.1).
- BlueZ 5.2.
- Raspbian Linux.
- Windows 7 64-bit SP 1.

## Hardware

- Samsung Galaxy S4 (Android 4.2.2).
- Raspberry Pi.
- TI CC2541 Sensor Tags (4).
- USB Bluetooth Adapters (4).
- NFC Tag.

# Iteration Schedule



- Iteration development lifecycle.
- Iteration 1 (Ended 12/02/13).
  - Communication methods.
  - NFC tag functionality.
  - Android training.
  - Considered accident detection methods.
  - Research on car collisions.
- Iteration 2 (Ended 01/31/14).
  - Completed UI for smartphone application.
  - Preliminary networking programs.

# Iteration Schedule

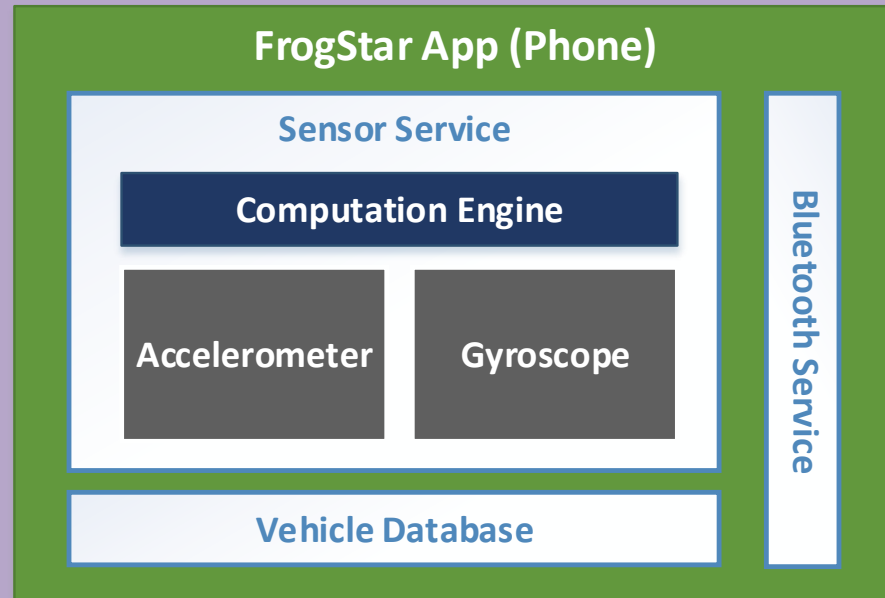


- **Iteration 3 (Ended 03/04/14).**
  - Accident detection on smartphone.
  - Smartphone testing.
- **Iteration 4 (Ended 04/18/14).**
  - Accident detection on OBCU.
  - Finalized networking.
  - System testing.
- **User-acceptance testing.**

# Smartphone Components



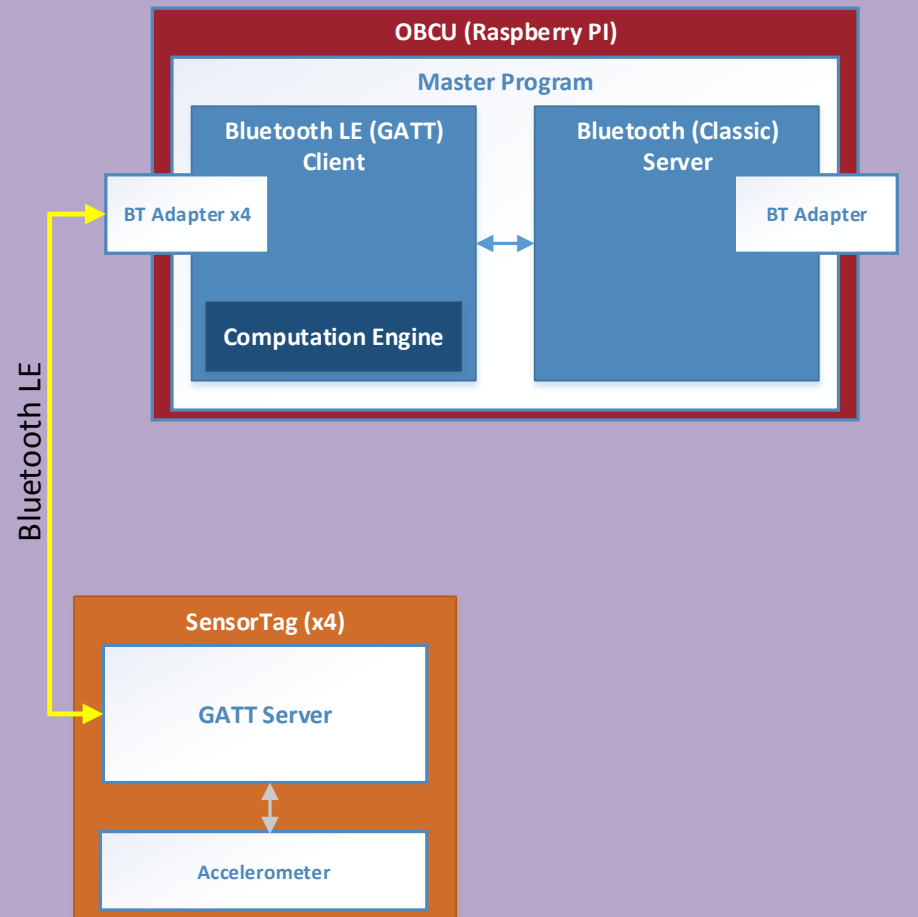
- **FrogStar Application.**
  - User information.
  - Vehicle database.
  - NFC capability.
  - Technician mode.
- **Bluetooth Service.**
  - OBCU communication.
- **Sensor Service.**
  - Computation engine.
  - Sensor querying.



# OBCU (Raspberry Pi) Components



- Master program.
- Bluetooth server.
  - Smartphone communication.
- Bluetooth LE client.
  - General attributes (GATT).
  - SensorTag querying.
  - Computation engine.



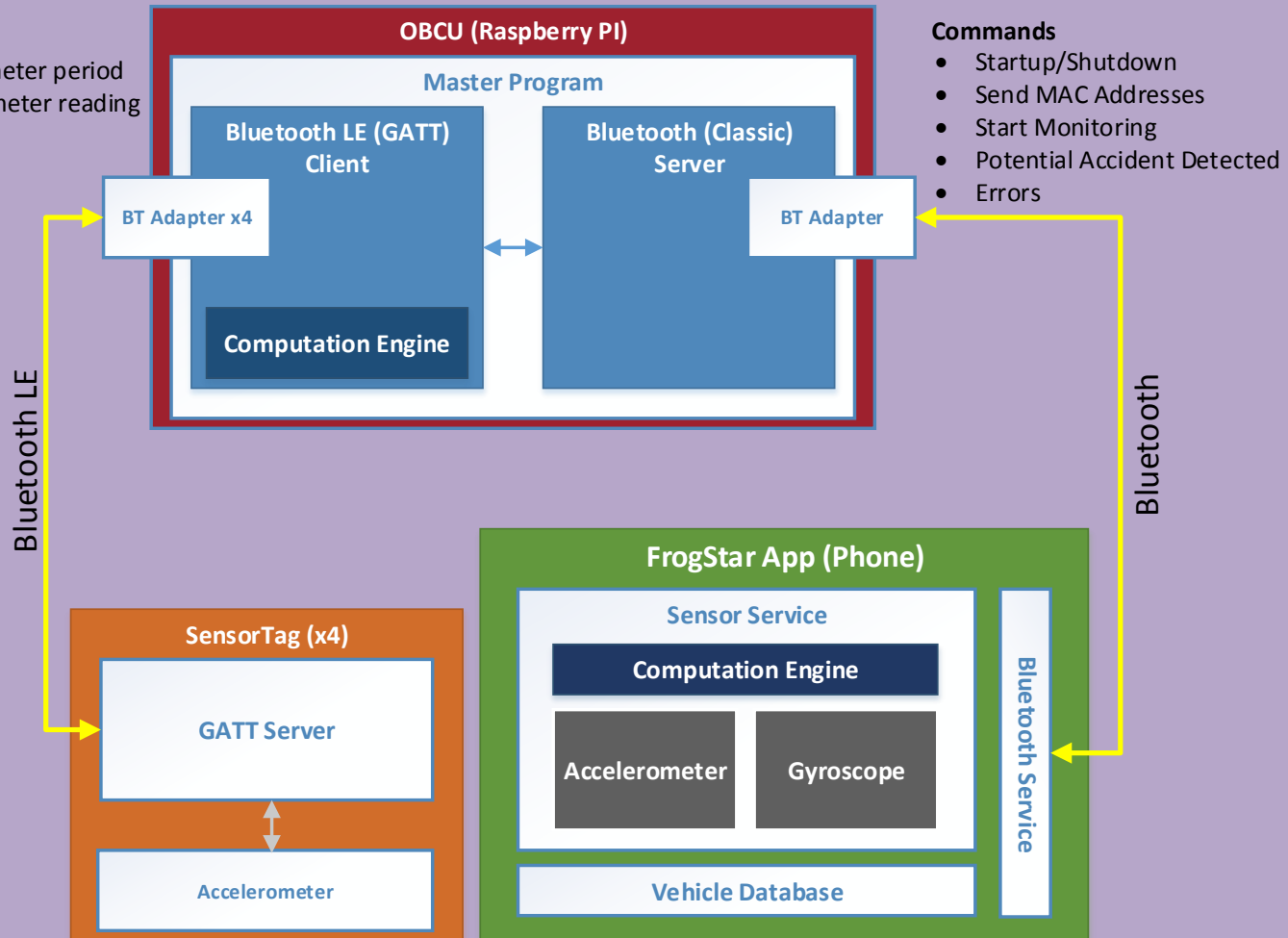


# System Architecture



## Commands

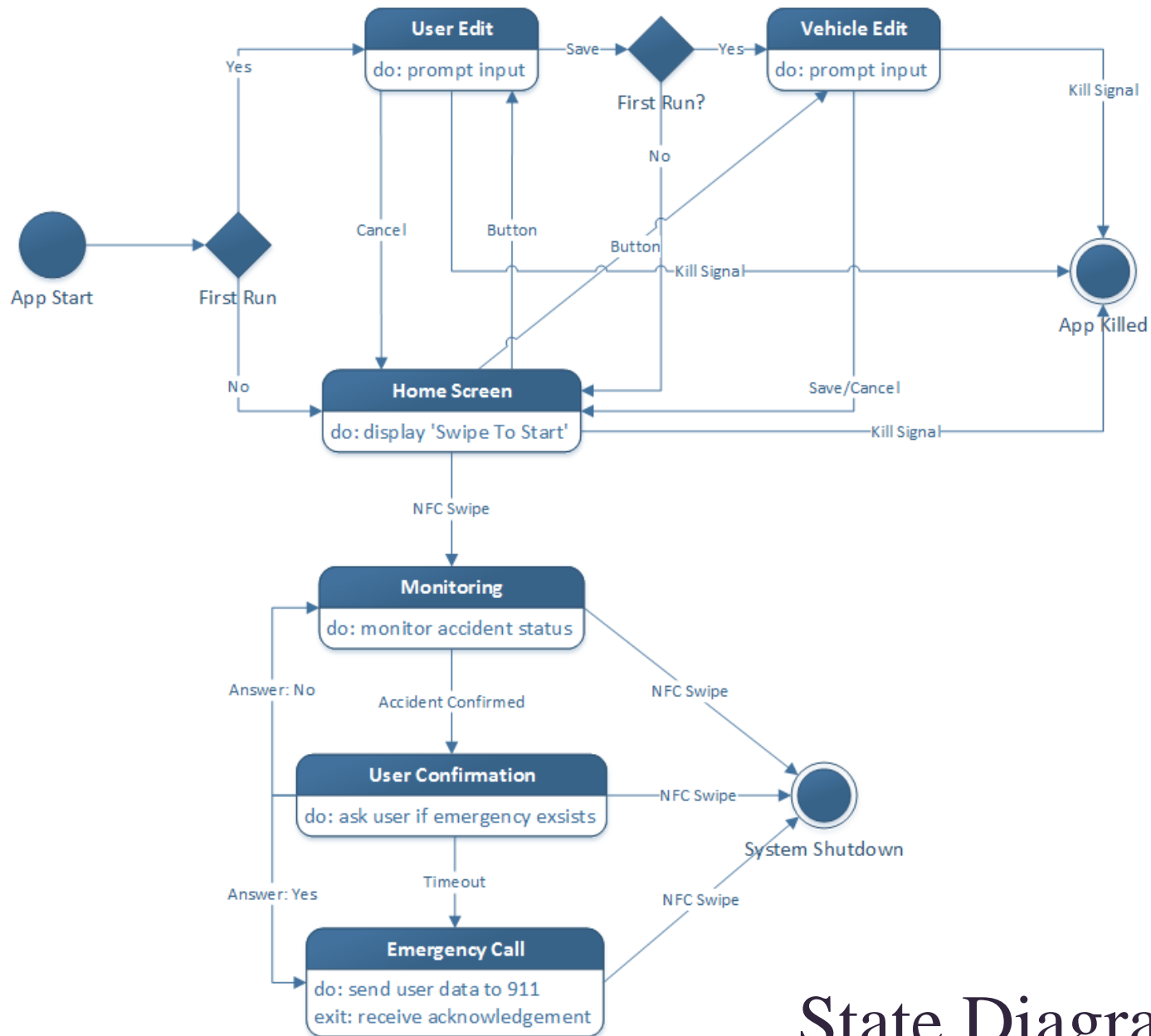
- Set accelerometer period
- Get accelerometer reading



# Key Requirements

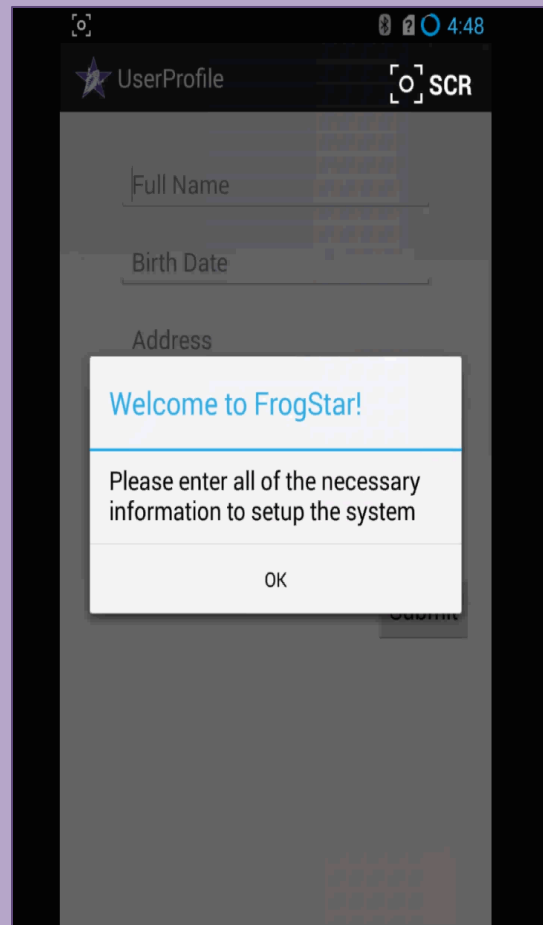


- **NFC tag.**
  - Network information.
  - Toggle the system on and off.
- **Smartphone and OBCU.**
  - Query sensors.
  - Accident detection.
  - Confirm accidents.
- **OBCU.**
  - Provides system status info.
- **On accident confirmation, alert user.**
  - If user confirms or cannot respond, alert proper authorities.



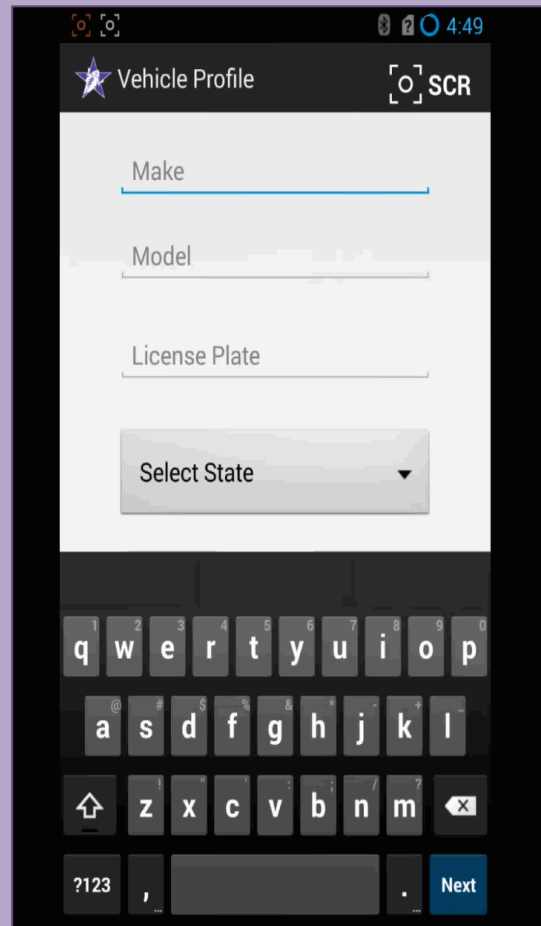
# State Diagram

# FrogStar Application – User Profile



- First-run of application.
- Option to edit from home screen.
- Emergency contact selected from contacts on device.
- Input validation.

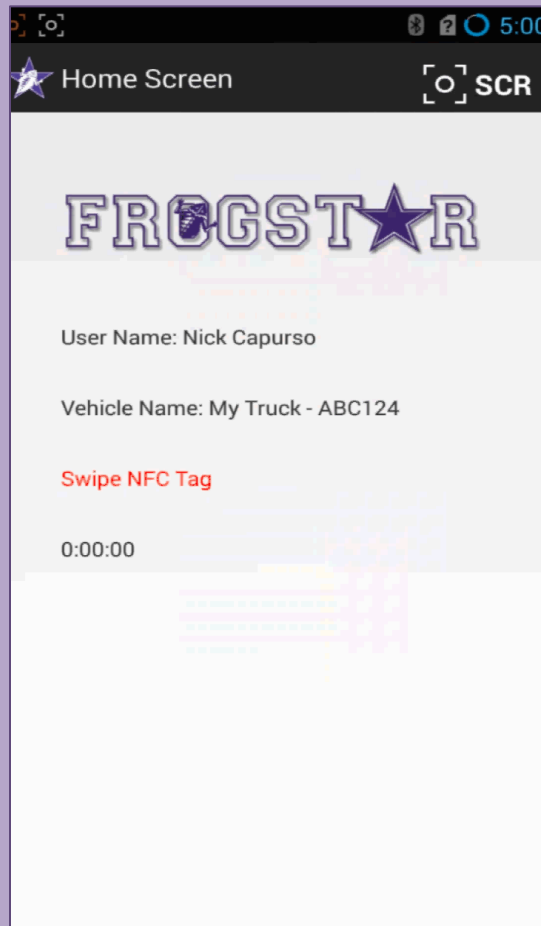
# FrogStar Application – Vehicle Profile



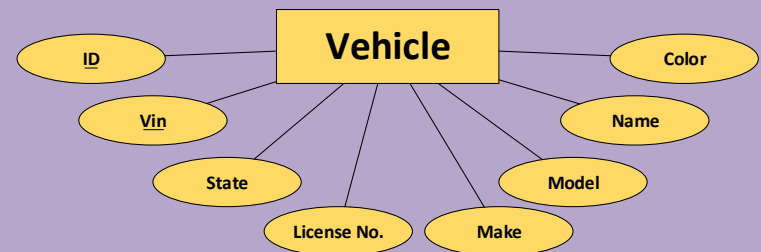
The screenshot shows the 'Vehicle Profile' screen of the FrogStar application. At the top, there is a star icon and the text 'Vehicle Profile' on the left, and a camera icon with 'SCR' on the right. Below this are three text input fields labeled 'Make', 'Model', and 'License Plate'. Underneath the 'License Plate' field is a dropdown menu labeled 'Select State'. A virtual keyboard is visible at the bottom of the screen, with a blue 'Next' button in the bottom right corner.

- First-run of application.
- Option to edit from home screen.
- Multiple vehicles can be entered.

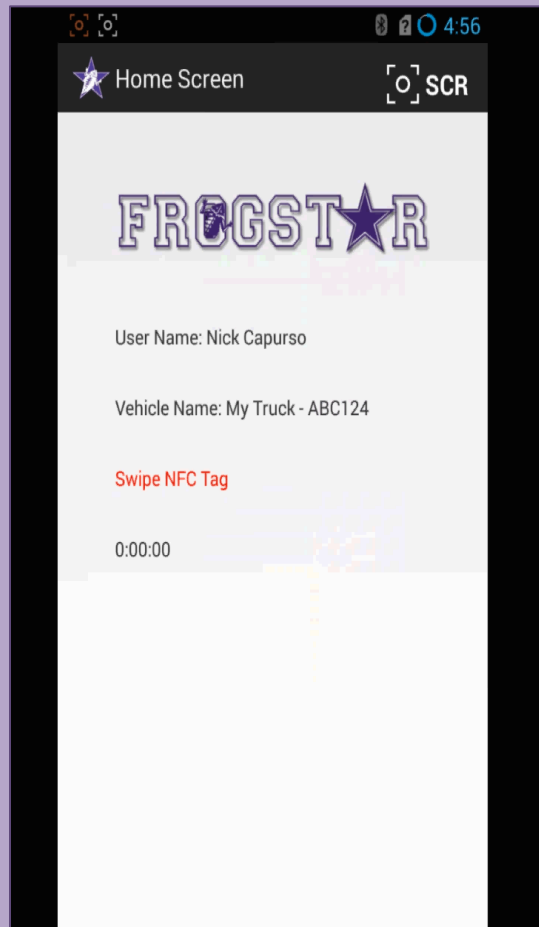
# FrogStar Application – Vehicle Database



- List of all vehicles stored in application.
- Long click to edit or delete.
- Can select default vehicle.

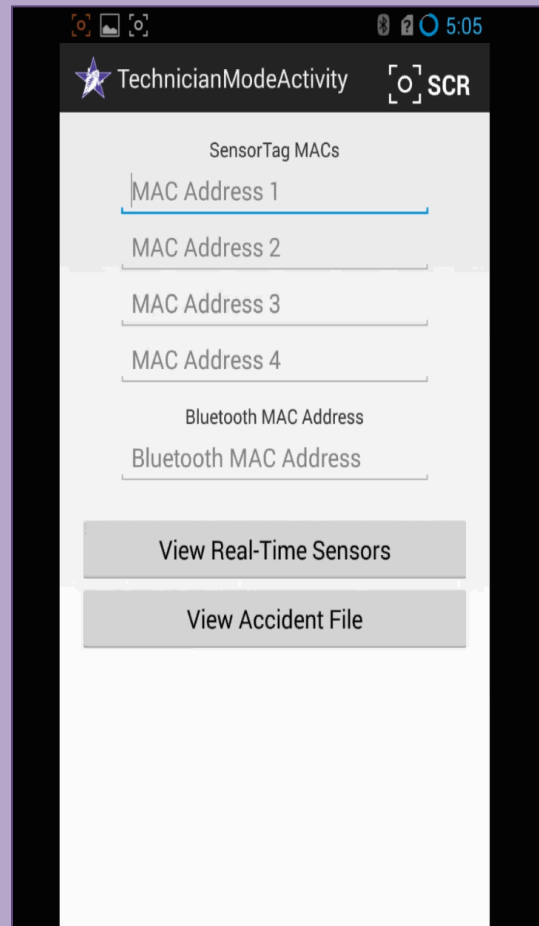


# FrogStar Application – Home Screen



- Prompts user to swipe NFC tag.
- Provides system information.
- Main hub screen.

# FrogStar Application – Technician Mode



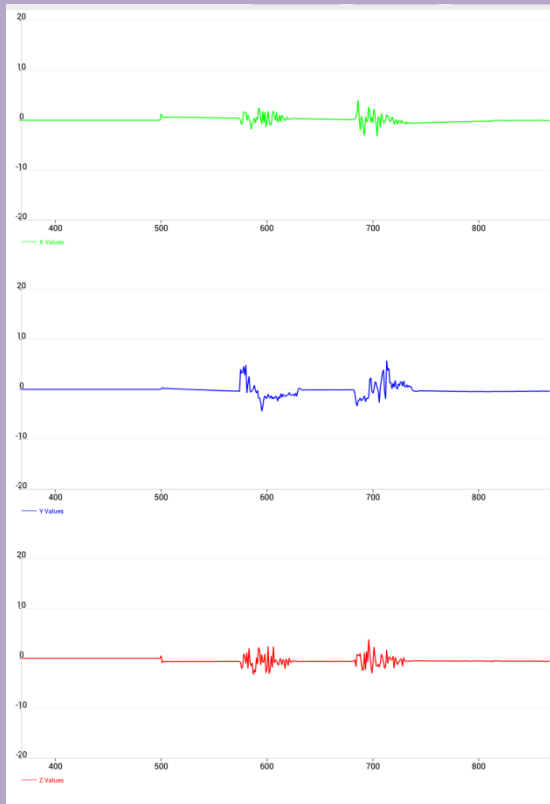
- Write new NFC tags.
- Modify existing tags.
- View real-time sensors.
- Open accident data.



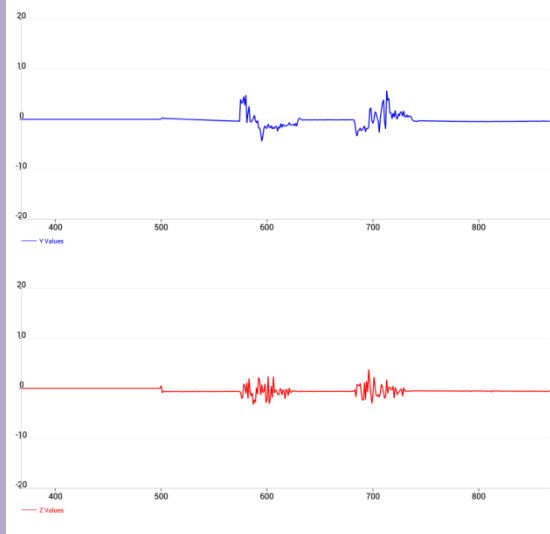
# FrogStar Application – Accelerometer Graph



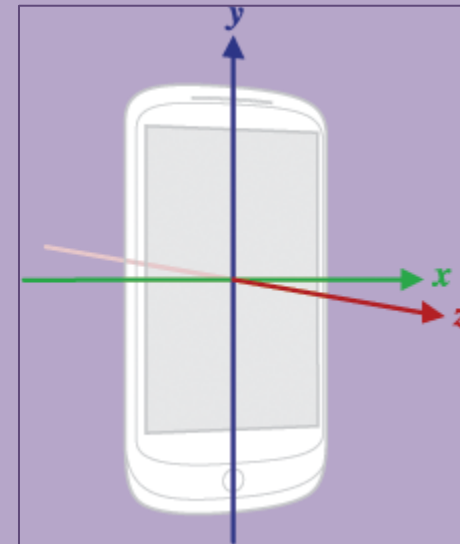
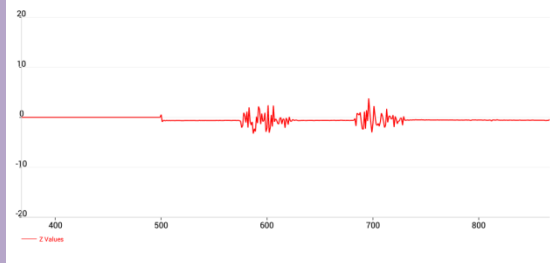
X



Y



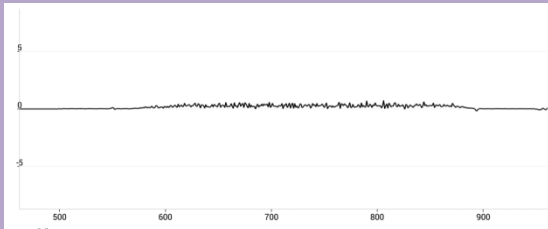
Z



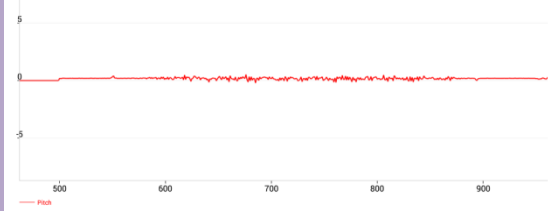
# FrogStar Application – Gyroscope Example



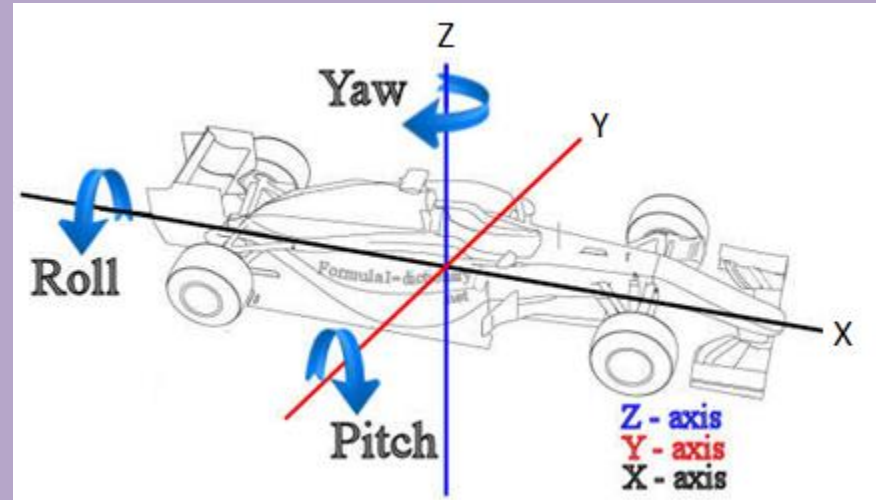
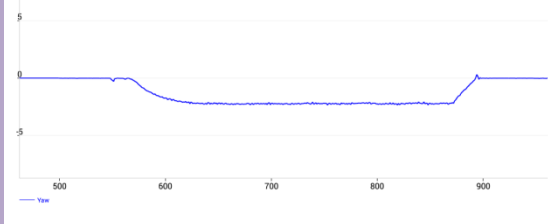
Roll



Pitch



Yaw



# Sensor Readings and Accident Detection

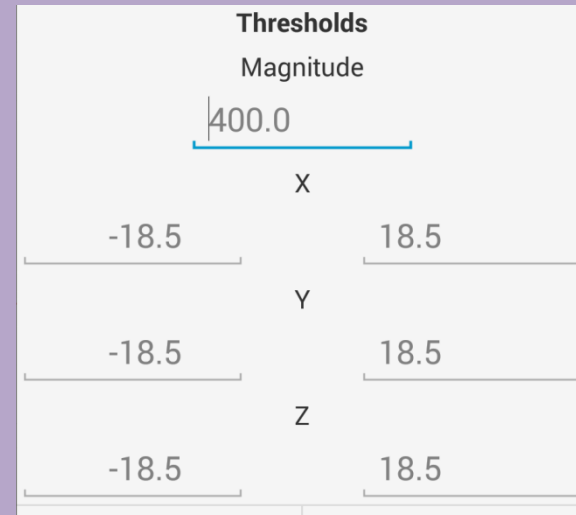


## Sensor Readings

- Vector readings.
- Accelerometer magnitude.
- Gyroscope orientation change.

## Accident Detection

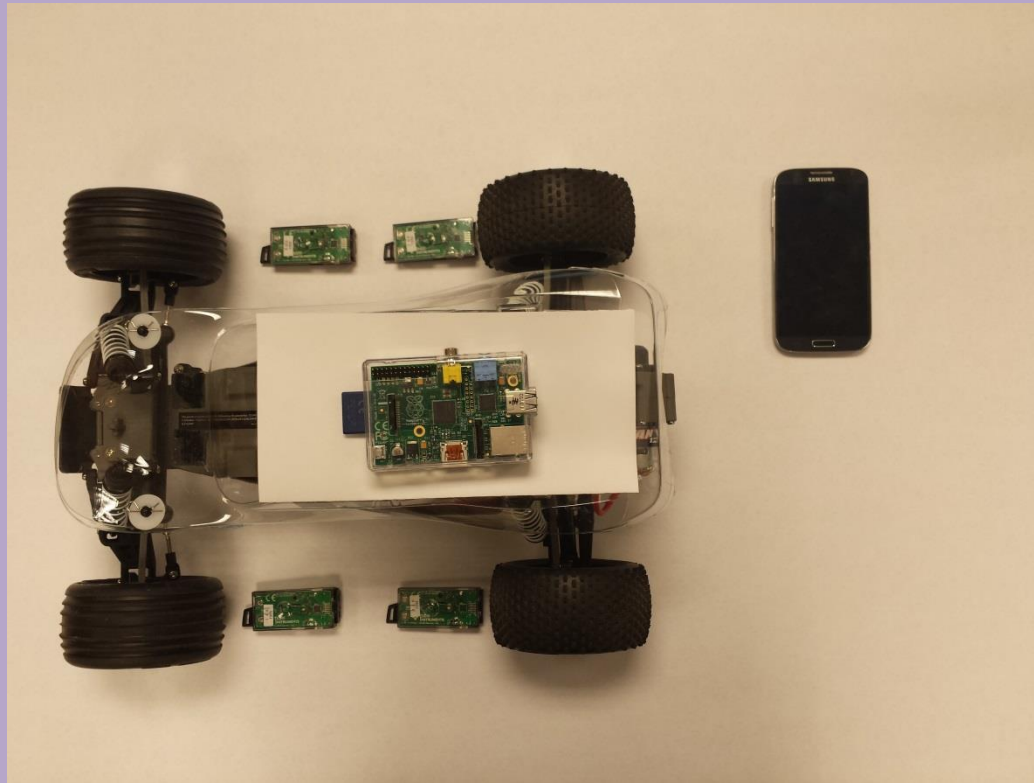
- Threshold-based.
  - Magnitude.
  - Min-Max Values.



# FrogStar Application – Testing Environment



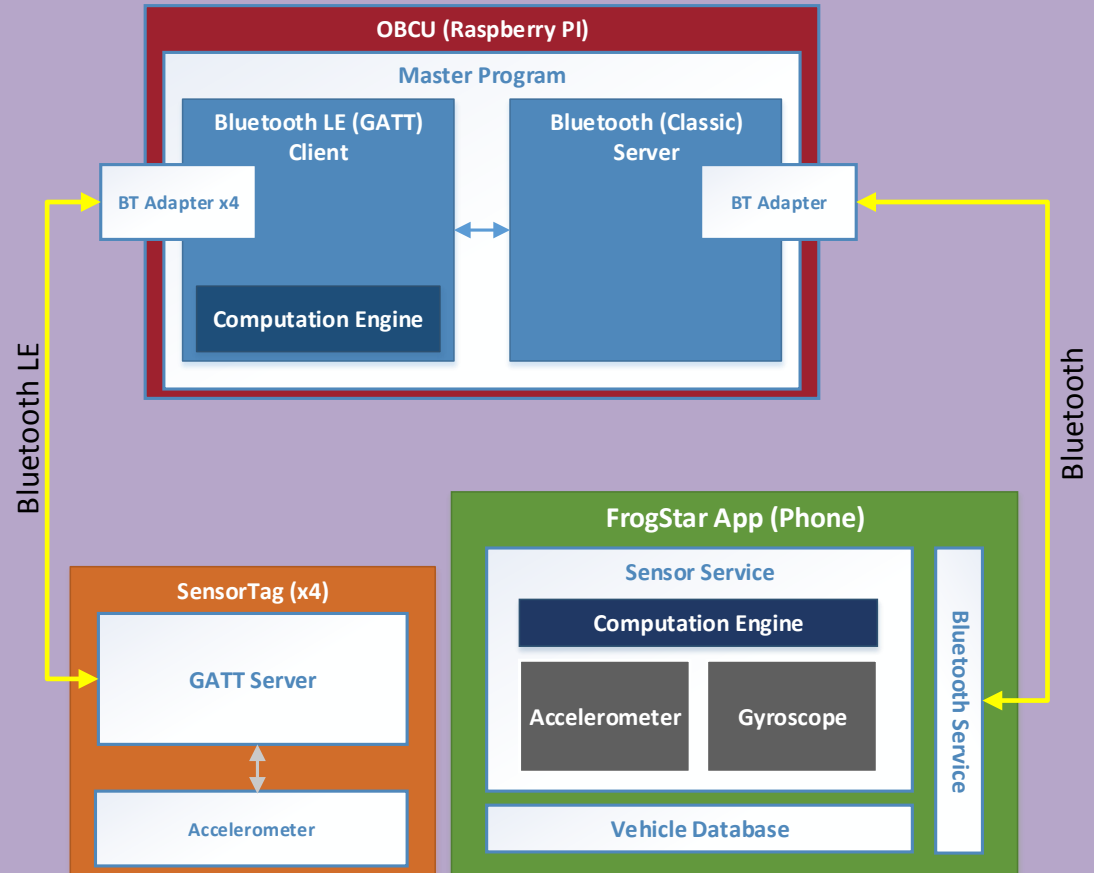
- Prototype Vehicle.

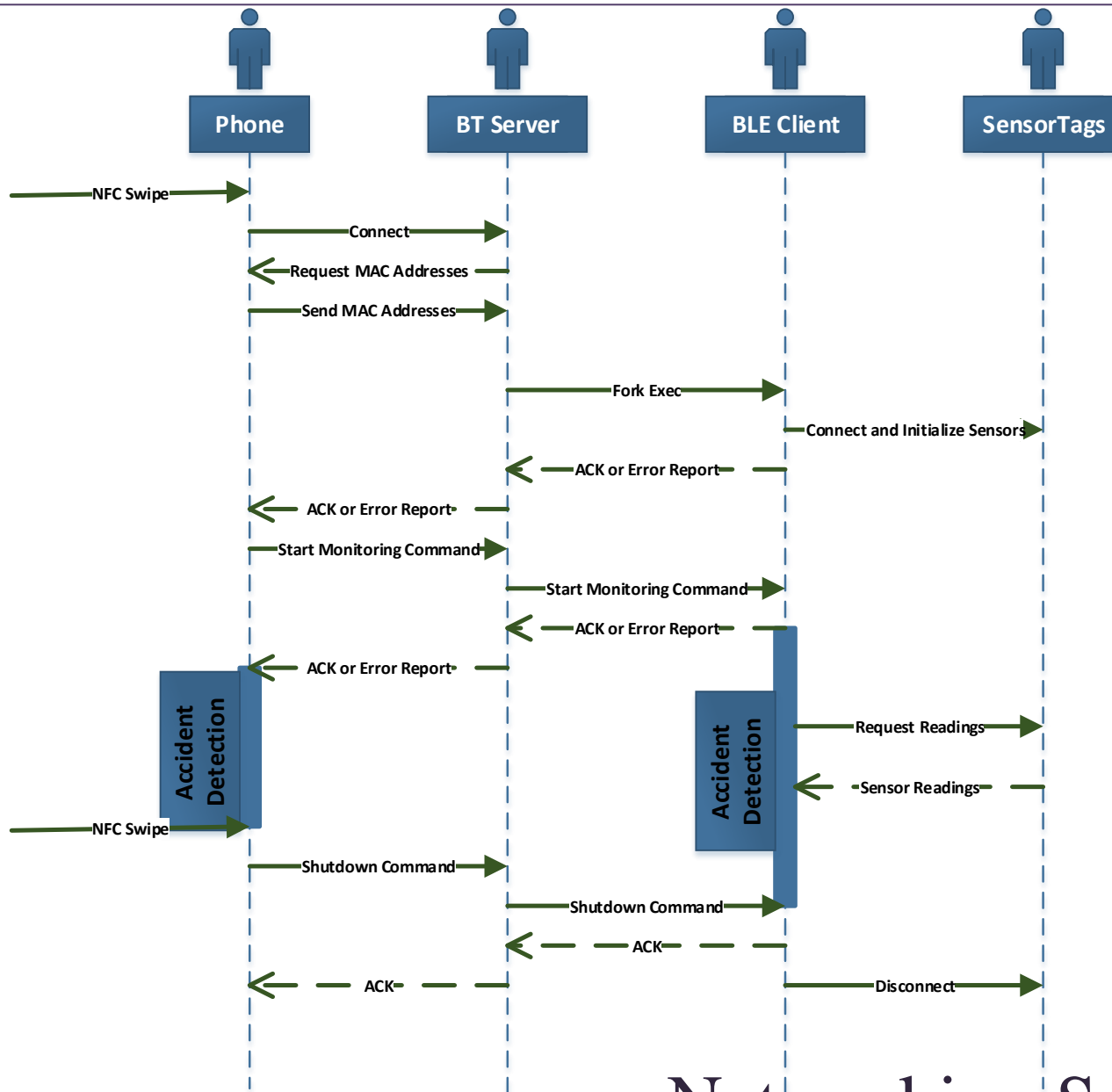


# Networking



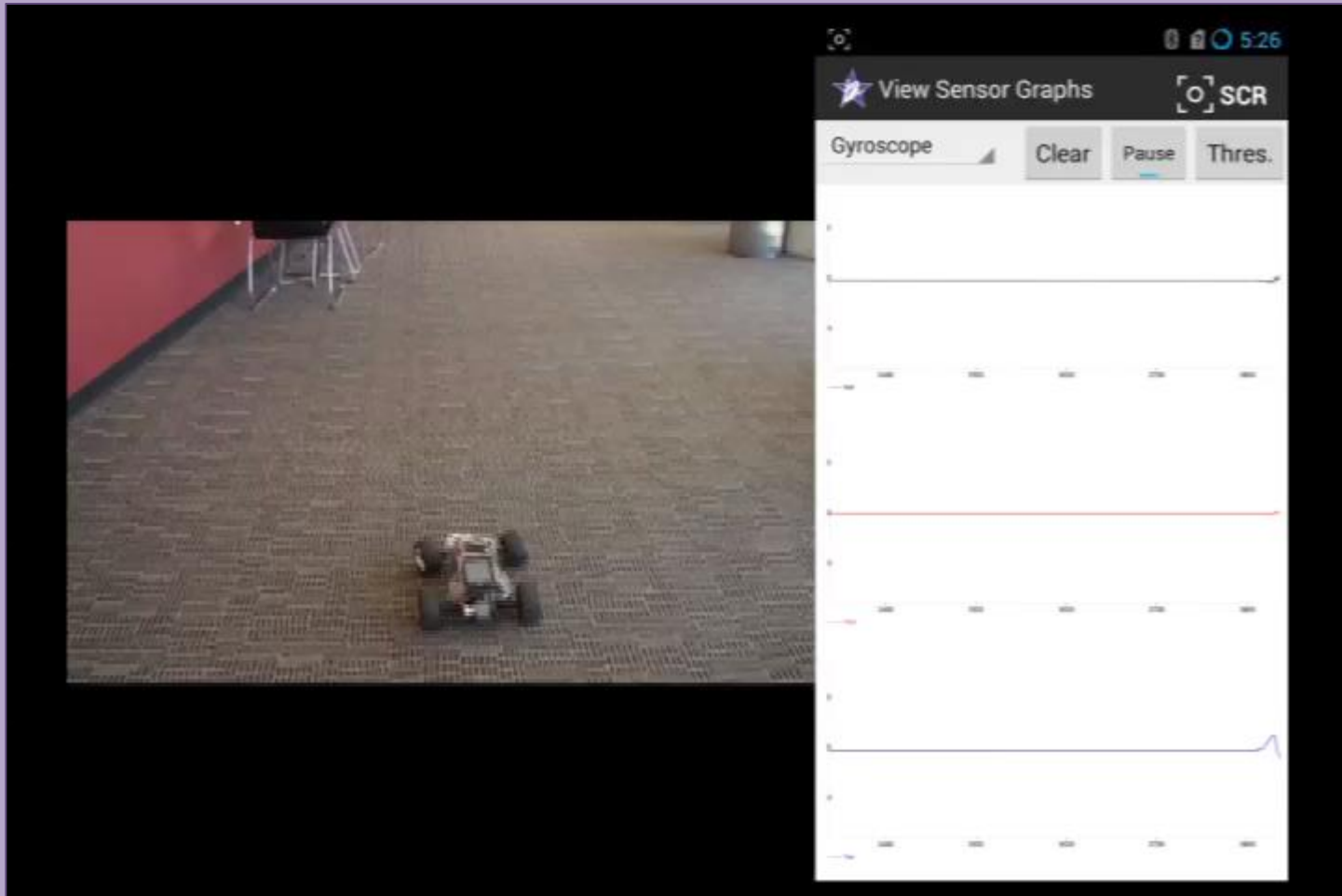
- Smartphone
  - Bluetooth Service
- OBCU
  - Bluetooth Server
  - Bluetooth LE Client
    - ✦ Gatttool



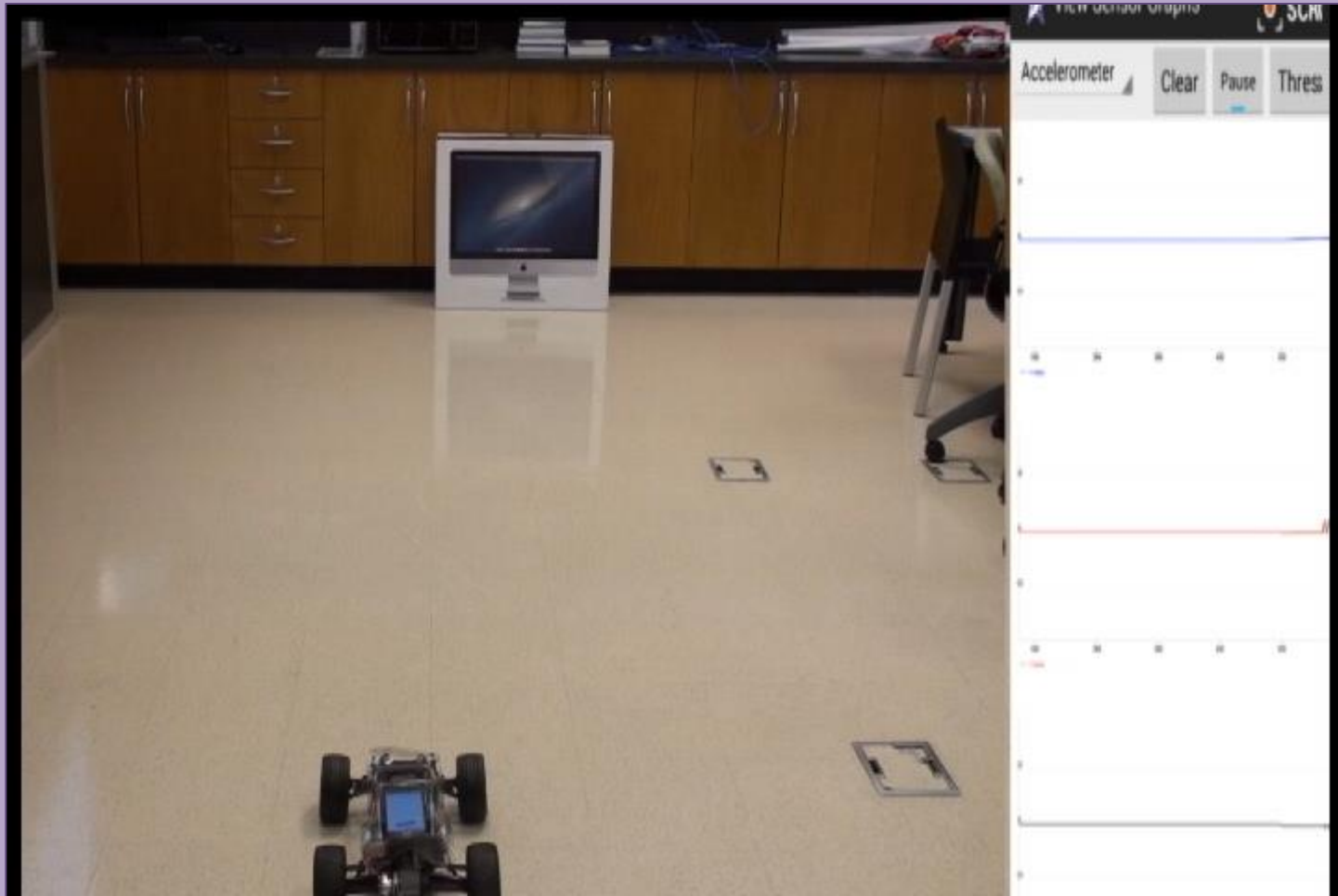


# Networking Sequence

# Demo

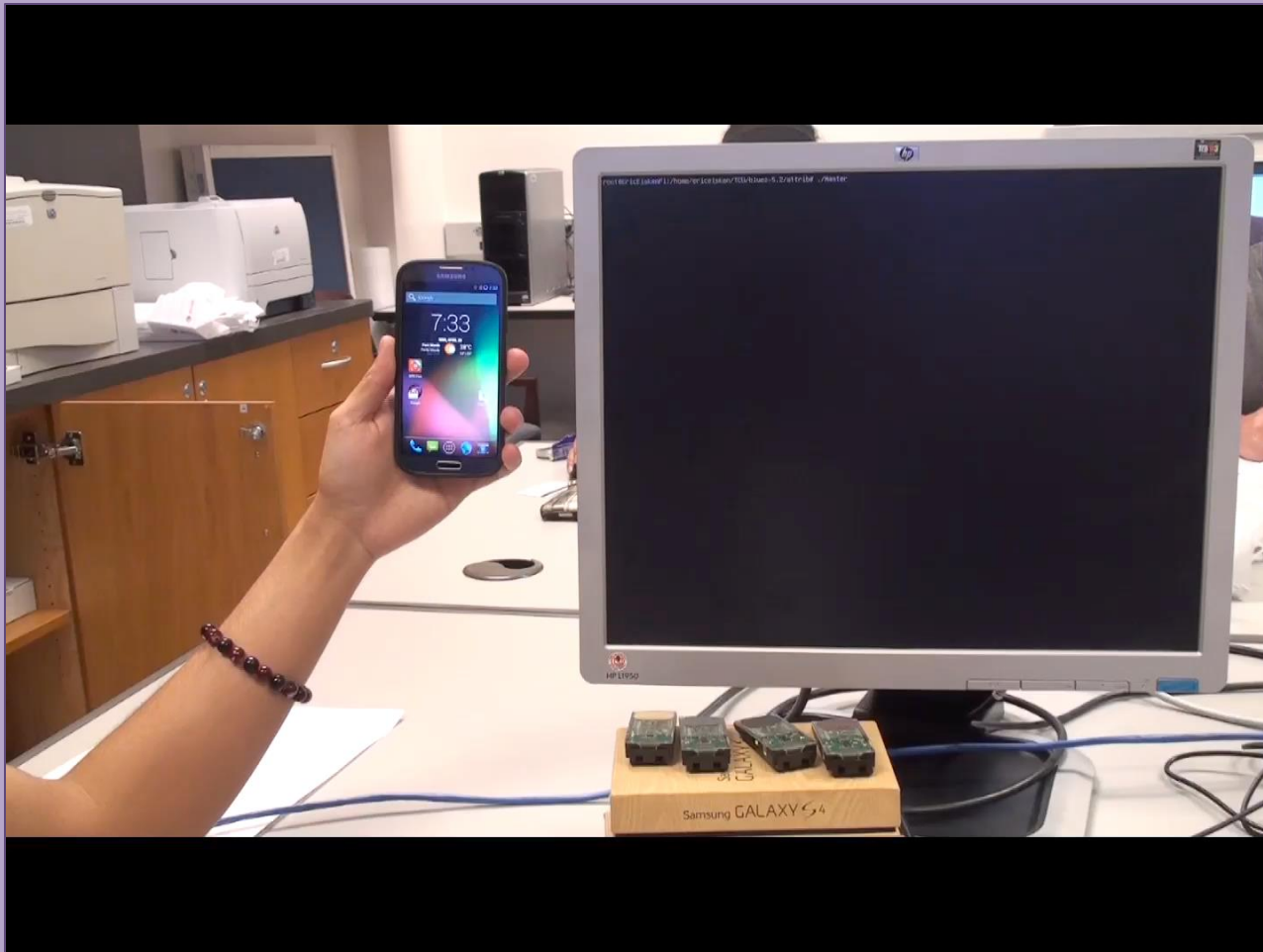


# Demo





# Demo



# Challenges



- Real-time system.
- Lack of Bluetooth programming documentation.
- Networking.
- SensorTag stock sampling rate.
- Smartphone sensor calibrations.
- Linux distribution for OBCU.

# Lessons Learned



- Start early.
- Time management.
- Teamwork.
- Review documents.
- Expect the unexpected.
- Testing.

# Future Work



- Upgrade SensorTag firmware.
- Deployment in a real vehicle.
  - Research on structural differences between RC car and vehicle.
  - Find optimal number and placement of SensorTags per vehicle.
  - OBCU backup power source.
- Storing medical information.
- Sending accident context to emergency responders.

# Acknowledgments



- Dr. Payne – class coordinator.
- Dr. Ma – client.
- TCU Computer Science Department Faculty.

# Questions?

