# Midway Design Review

TrueBase
SDP 2021
Team 30

University of Massachusetts Amherst BE REVOLUTIONARY™

# Agenda

- Updated Problem Statement
- Updated System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management

# Problem Statement


- Growing number of close-call plays - nearly impossible to officiate by the naked eye
- Replay and review systems waste a lot of time - human error should not be a part of the game
- What if the need for replay and review was eliminated?
- Even better, what if every close call that was made at first base was almost certainly correct?

# Agenda

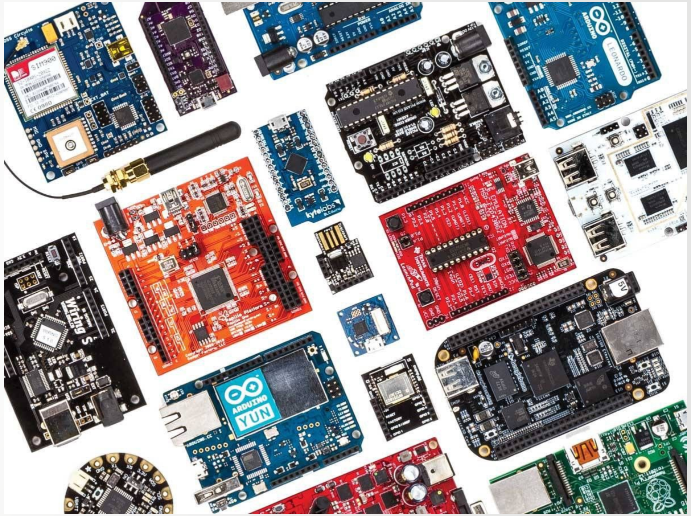
- Updated Problem Statement
- Updated System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management

# Updated System Specifications

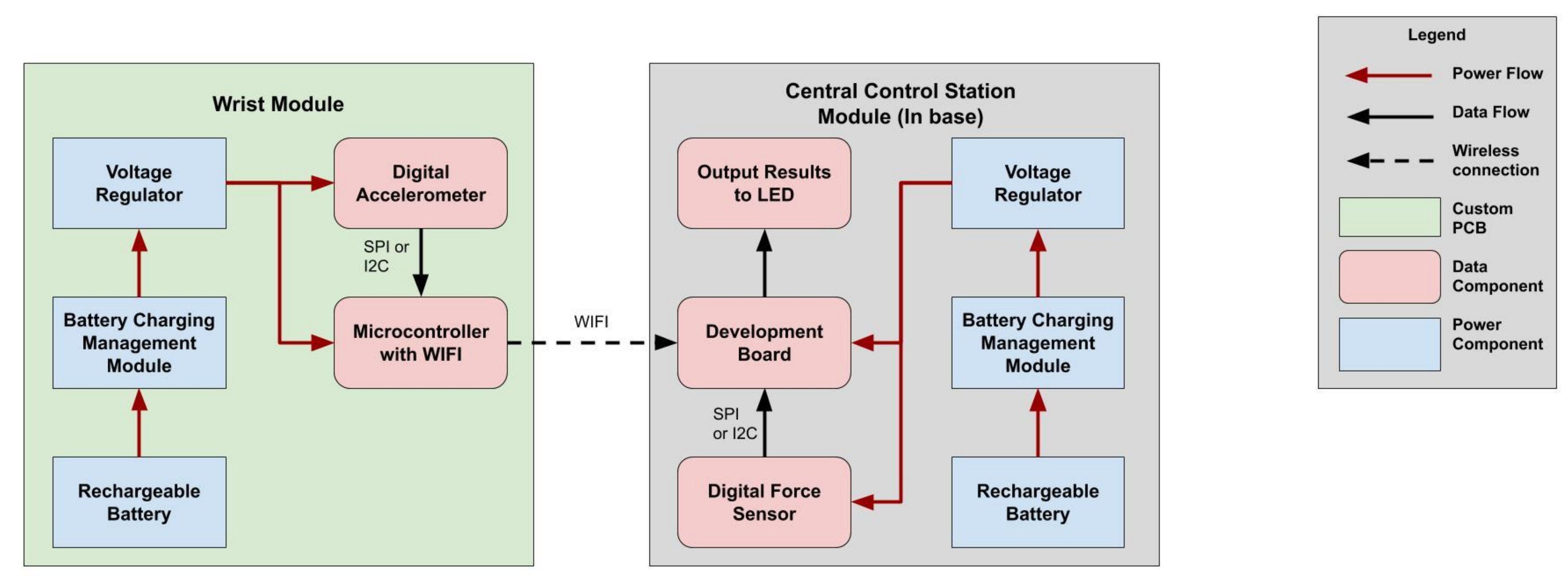
- **35ms accuracy** - enough for “bang-bang” plays
- 5+ hour battery life - enough for extra-inning baseball games
- <150g wrist module weight - as much as the average wristwatch
- System will not interfere with gameplay
- Meaningful and easily interpretable output

# Agenda

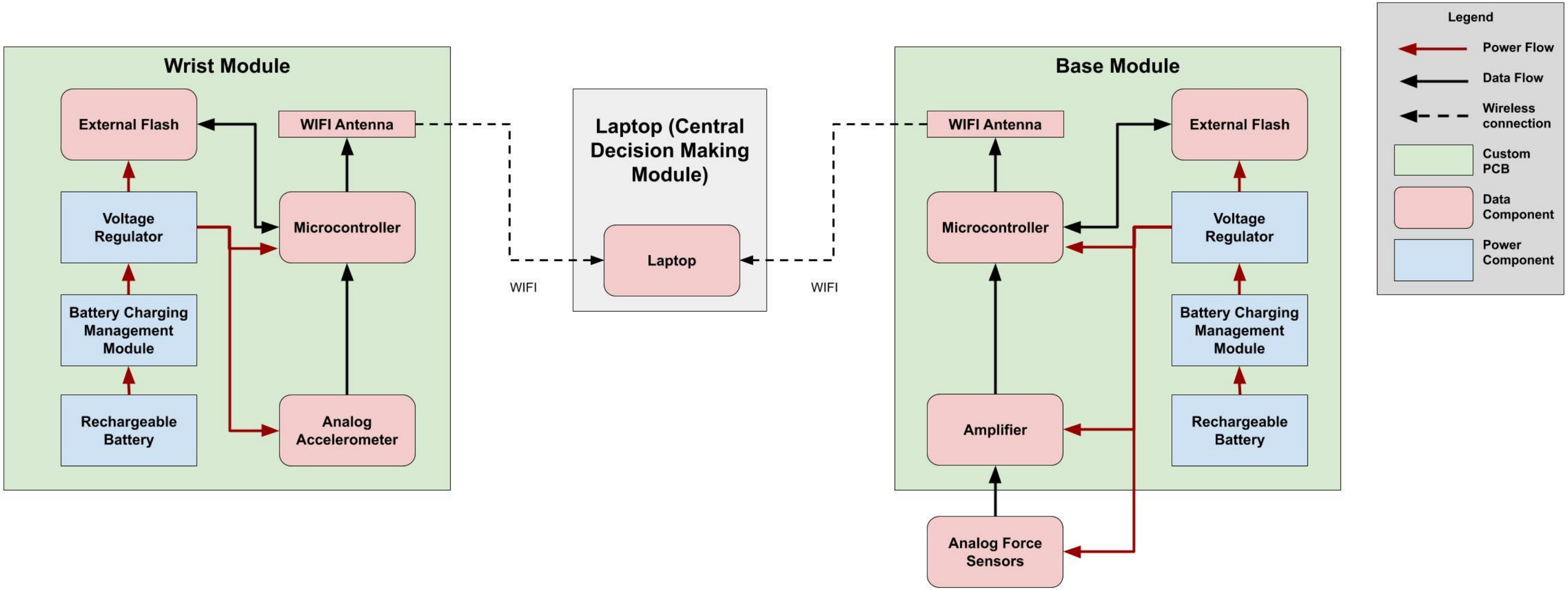
- Updated Problem Statement
- Updated System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management



# Previous Block Diagram



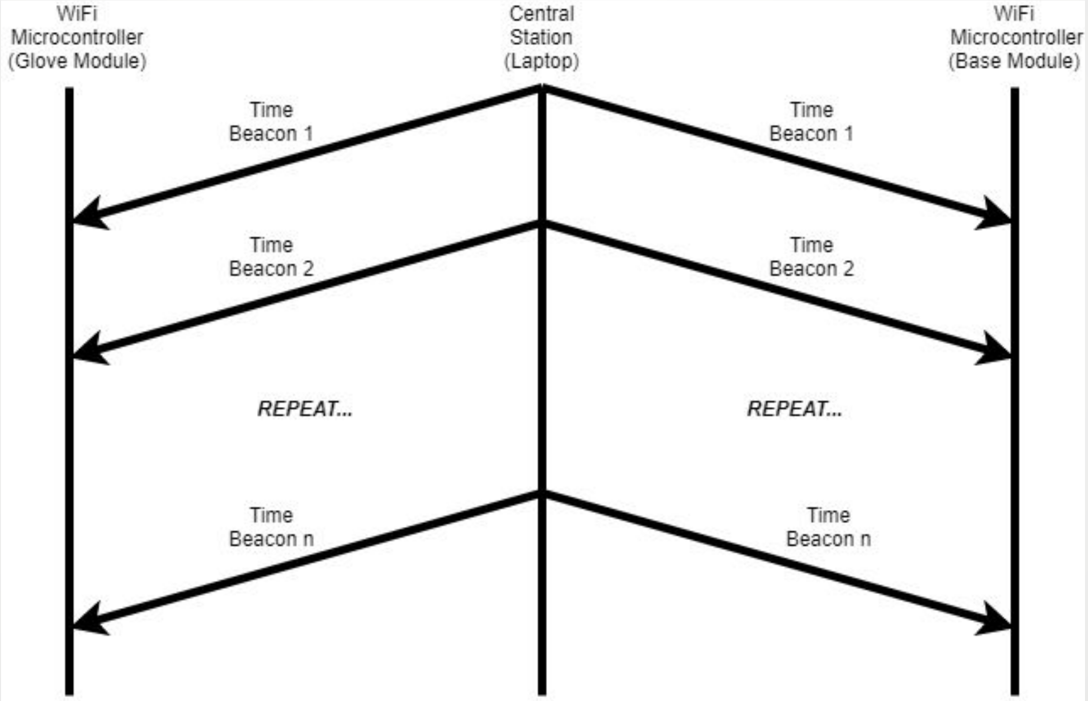
# Overall System Block Diagram



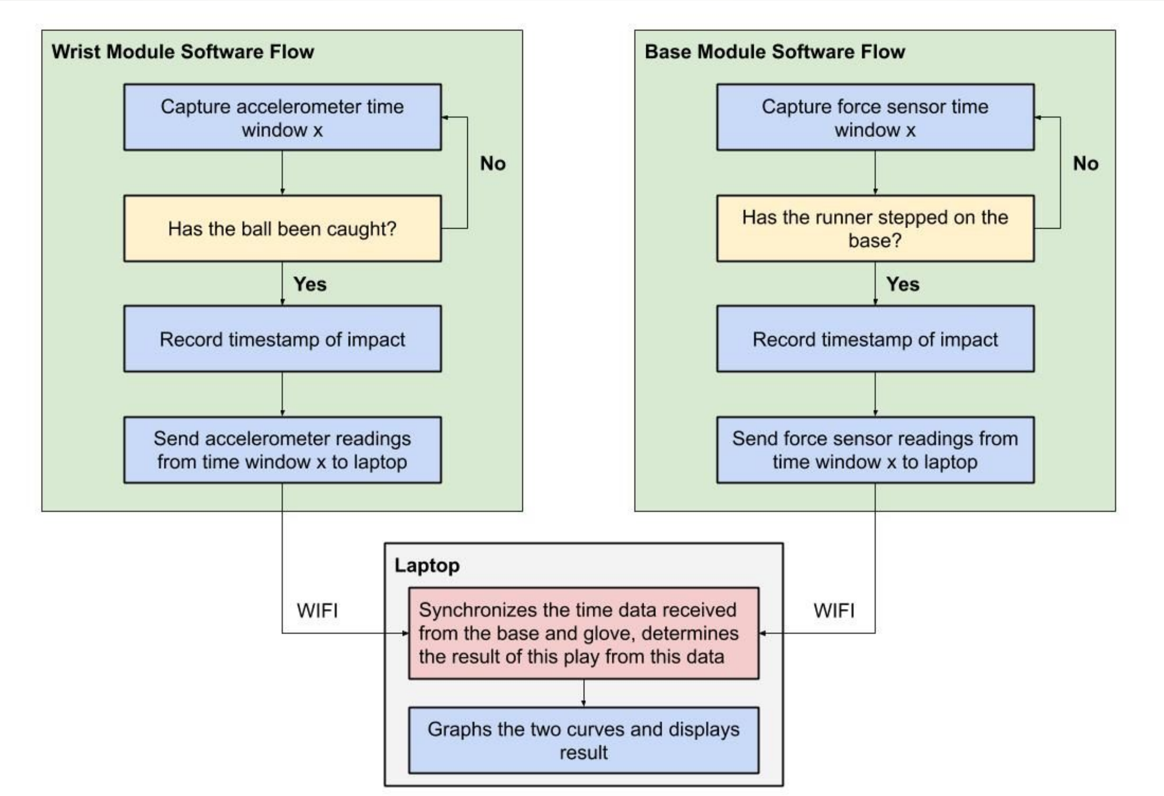

University of Massachusetts Amherst BE REVOLUTIONARY™

TrueBase

# Time Beaconing Diagram



# Sensor Readings Software Diagram



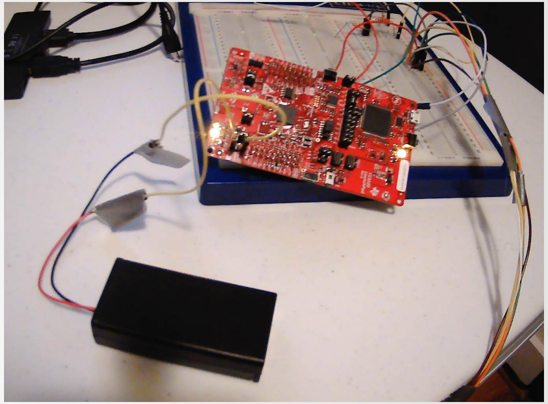
# Agenda

- Problem Statement
- System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management

# Proposed MDR Deliverables

- Core logic design will be completed
- Be able to power force sensor with rechargeable battery and read its output using a development board
- Be able to power an accelerometer with a rechargeable battery and read its output using a development board
- Real Time Clocks (RTC) will sync properly and have negligible drift
- Be able to send timestamp data over WiFi to development board

# Wrist Module - Jonah



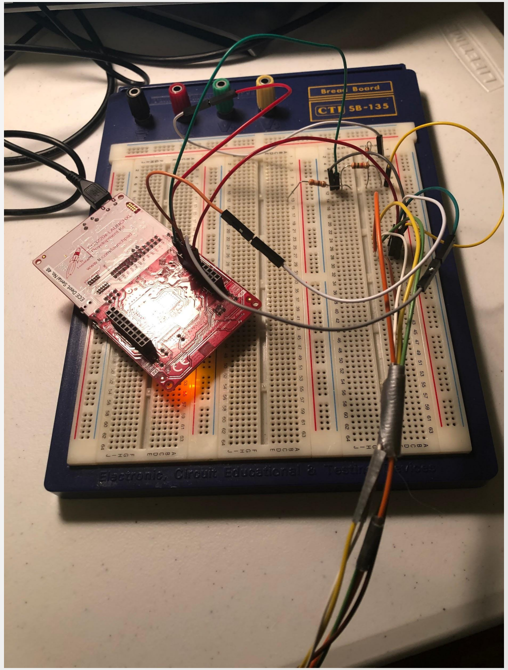
**This semester:**

- Force Sensor and accelerometer research and part selection
- Parts ordering
- Soldering
- Deliverables

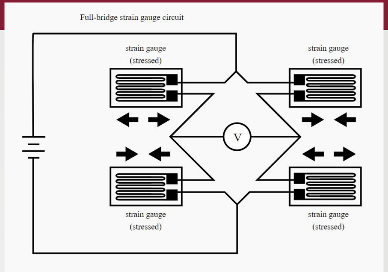
**PDR Deliverable** - power board using a rechargeable battery and get readings from the accelerometer.

**Achieved by MDR** - power board using regular battery, get readings from accelerometer and timestamp the data, aggregate triaxial readings into acceleration vector, and setup initial testing environment

# Wrist Module Setup

# Base Module - Derek



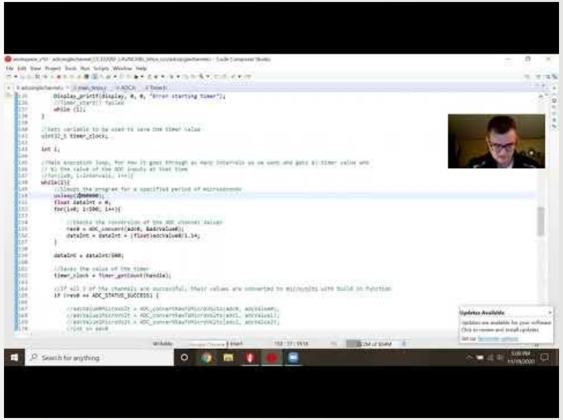

https://medium.com/@dewesoft/measure-weight-with-load-cell-sensors-8ae1dc22926f

## This semester:

- Researching time synchronization protocols, wireless communication microcontrollers
- Soldering training
- Load cell implementation & deliverables

## Load Cell Implementation

- Starting with the Arduino Uno
    - Load cell implementation with the HX711 (Wheatstone Bridge)
    - Weight Scale
- Moving onto the CC3220SF LaunchpadXL
- Configuring different modules onto one project into CCS
    - UART Data Transfer for Serial Terminal
    - Timer
    - Analog to Digital Converter



# WiFi Communication Capabilities - Nnaji

1. **Configure Windows 10 laptop as WiFi AP server**
2. **Connect dev. boards to laptop AP as client**
3. **Send packets between laptop and boards through TCP sockets**
    a. In parallel with multi-threaded processes on laptop side
4. **Send commands from laptop to boards**
    a. “start_counting” - start board’s clock
    b. “get_time” - get board clock time in nanosecond resolution
5. **Receive stream of on-board accelerometer data from board to laptop**
    a. Plotted with Matplotlib on laptop

# Clock Synchronization - Vyom

**Proposed MDR Deliverable:**

*“Real Time Clocks (RTC) will sync properly and have negligible drift”*

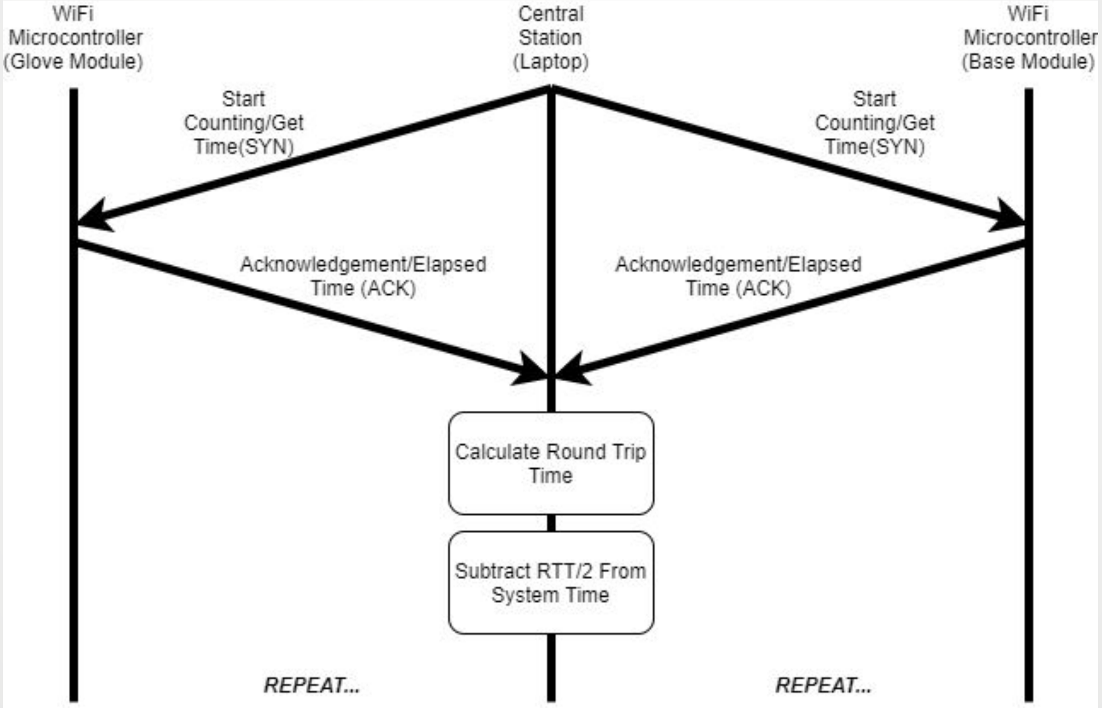
**Achieved by MDR - Current Implementation (with Demo):**

- Central Station signals Module to start counting - initial synchronization
- Central Station will periodically:
    - Ask module for its local time elapsed
    - Restart count on module (to negate drift) - resynchronization
- Account for latency/delay by measuring Round Trip Time and Offset

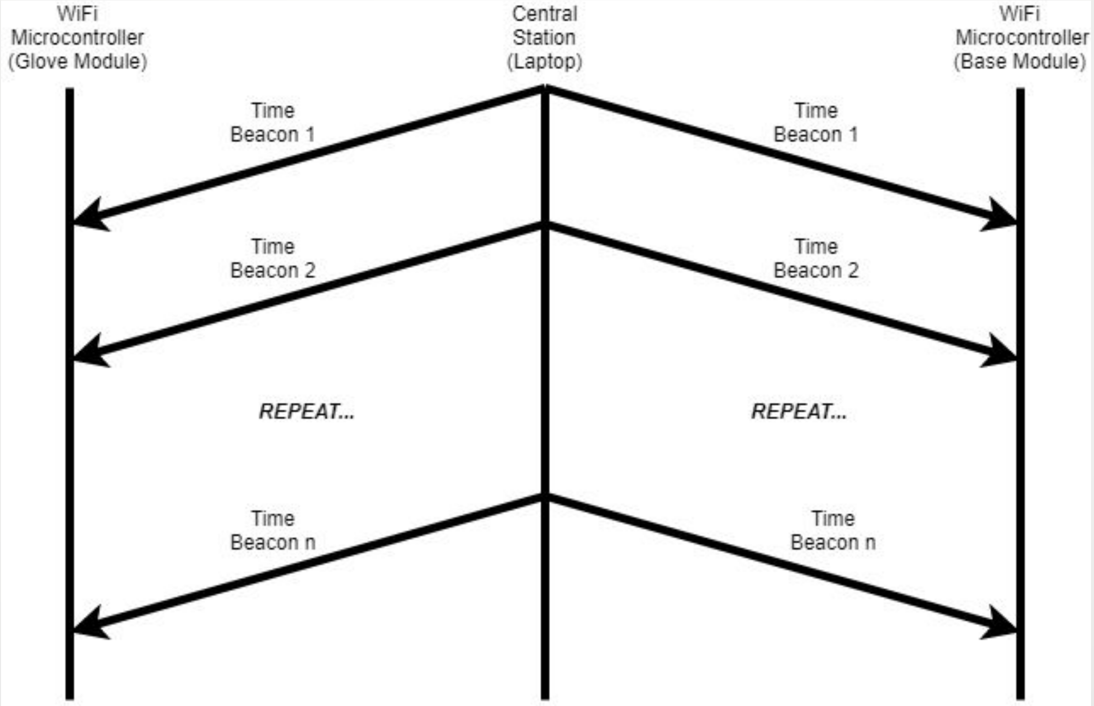
**Planned New Implementation**

- Use Time Beaconing from Central Station to modules
- Send timestamp of Central Station instead of having module count

# Clock Synchronization - Vyom

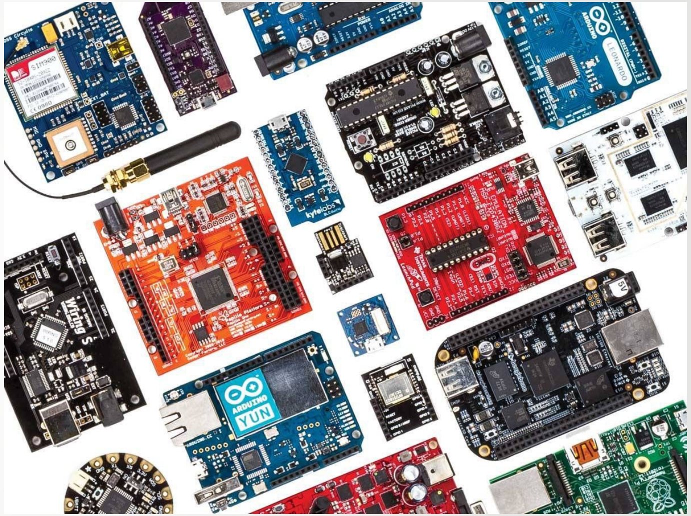


# Clock Synchronization - Vyom



# Agenda

- Problem Statement
- System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management



# Hardware Plan for FPR

## Wrist Module

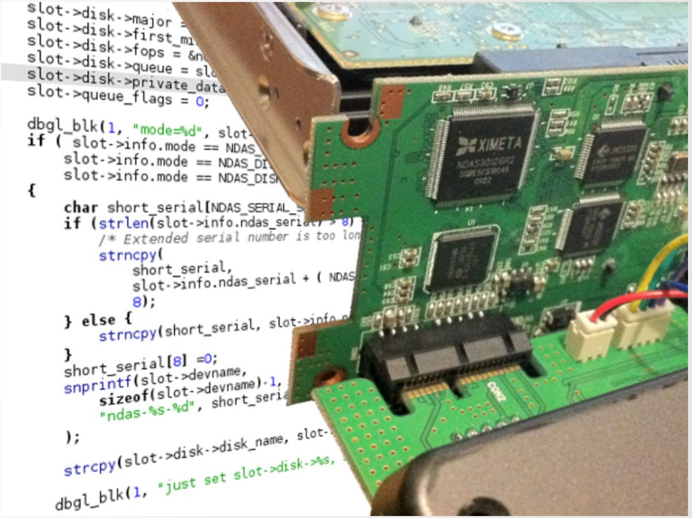
- **Mount CC3220SF, Analog Accelerometer and any other helping parts listed to the PCB**
  - Battery Charging Management Module
  - Voltage Regulator
  - Flash Memory
  - WiFi Antenna
- **Attach the Rechargeable Battery to the PCB**
- **PCB Specifications**
  - As small as possible (under 150 g)
  - Can be worn on the wrist

## Base Module

- **Mount the CC3220SF, Bar Load Cell, and any other helping parts listed to the PCB**
  - Battery Charging Management Module
  - Voltage Regulator
  - Flash Memory
  - Voltage Amplifiers
  - WiFi Antenna
- **Attach the Rechargeable Battery to the PCB**
- **PCB Specifications for the base just require it fits under the base**

# Agenda

- Problem Statement
- System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management



# List of Hardware

## Wrist Module

- **CC3220SF LaunchpadXL**
- **ADXL335 Accelerometer**
- **Lithium Ion Rechargeable Battery and Associated Parts**
  - Battery Charging Management Mod
  - Voltage Regulator
- **Custom PCB**
- **CC3220SF Microcontroller**
- **External Flash for Programming MCU**
- **WIFI Antenna**

## Base Module

- **CC3220SF LaunchpadXL**
- **Arduino Uno**
- **HX711 A2D+Amplifier**
- **Half-Bridge Force Sensors**
- **Bar Load Cells**
- **Voltage Amplifier**
- **Lithium Ion Rechargeable Battery and associated parts**
  - Battery Charging Management Mod
  - Voltage Regulator
- **Custom PCB**
- **CC3220SF Microcontroller**
- **External Flash for Programming MCU**
- **WIFI Antenna**

# List of Software

- **Core Programming Languages:**
  - C  (modules)
  - Python (laptop AP)
- **IDEs**
  - Code Composer Studio - C
  - PyCharm - Python
  - Arduino IDE
- **TI Software**
  - TI CC3220SF SDK
  - TI Sysconfig
  - TI Sensor and Actuator Plug-in for SDK
- **Other**
  - Uniflash
  - Termite/Tera-Term
  - Github
  - Windows Netsh / Mobile Hotspot features

# Agenda

- Problem Statement
- System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management

# Project Cost Estimate

Total: ~$208

## Wrist Module

| | |
|---|---|
| Accelerometer on BB | $31.09 |
| Accelerometer | ~$15 |
| Custom PCB | ~$25 |
| Rechargeable Battery | ~$5 |
| Battery Charging Management Module | ~$2 |
| Voltage Regulator | ~$1 |
| 2x CC3220SF | $15.12 |
| WiFi Antenna | ~$3 |

## Base Module

| | |
|---|---|
| 4x Load Cell | $7.88 |
| Bar Load Cell | ~$10 |
| 10x Voltage Amplifier | ~$3 |
| Custom PCB | ~$25 |
| Rechargeable Battery | ~$13 |
| Battery Charging Management Module | ~$2 |
| Voltage Regulator | ~$1 |
| 2x CC3220SF | $15.12 |
| WiFi Antenna | ~$3 |
| Baseball Base | ~$50 |

# Product Cost Estimate

Total: ~$120.16

## Wrist Module

| | |
|---|---|
| Accelerometer | ~$15 |
| Custom PCB | ~$25 |
| Rechargeable Battery | ~$5 |
| Battery Charging Management Module | ~$2 |
| Voltage Regulator | ~$1 |
| CC3220SF | $7.58 |
| WiFi Antenna | ~$3 |

## Base Module

| | |
|---|---|
| Bar Load Cell | ~$10 |
| Instrumentation Amplifier | ~$1 |
| Custom PCB | ~$25 |
| Rechargeable Battery | ~$13 |
| Battery Charging Management Module | ~$2 |
| Voltage Regulator | ~$1 |
| CC3220SF | $7.58 |
| WiFi Antenna | ~$3 |

# Agenda

- Problem Statement
- System Specifications
- Updated System Design
- MDR Accomplishments
- Hardware Plan for FPR
- List of Hardware and Software
- Project Expenditures
- Project Management

# Gantt Chart Until CDR

| Task | Engineer | 1-February | 8-February | 15-February | 22-February | 1-March | 8-March |
|---|---|---|---|---|---|---|---|
| *Spring Semester Start* | ALL | | | | | | |
| Develop Team Website | Vyom | | | | | | |
| Develop Team Website | Derek | | | | | | |
| Work on PCB Design | Vyom | | | | | | |
| Finalize Accelerometer Output Design | Jonah | | | | | | |
| Finalize Force Sensor Output Design | Derek | | | | | | |
| Finalize Wireless Communication Protocol Design | Nnaji | | | | | | |
| Finalize Time Sync Design | Vyom | | | | | | |
| Finalize Time Sync Design | Nnaji | | | | | | |
| Order + Receive PCB | ALL | | | | | | |
| Develop Fully Working Prototype v1.0 | ALL | | | | | | |
| **CDR** | ALL | | | | | | |

# Gantt Chart Until FPR

| Task | Engineer | 15-March | 22-March | 29-March | 5-April | 12-April | 19-April |
|---|---|---|---|---|---|---|---|
| Finalize PCB Implementation | ALL | | | | | | |
| Final Testing | ALL | | | | | | |
| Finish + Finalize Project | ALL | | | | | | |
| Finish Team Website | Vyom | | | | | | |
| Finish Team Website | Derek | | | | | | |
| Prepare Final Presentation | ALL | | | | | | |
| **FPR** | | | | | | | |

# Project Responsibilities

| | |
|---|---|
| **Jonah** | **Team Coordinator**, responsible for implementation of the PCB on the wrist module, specifically implementing the microcontroller in coordination with the accelerometer including the necessary software |
| **Vyom** | **Altium Lead**, responsible for testing and proving the timeliness of the data from the glove module to the base module, thus ensuring that system specs on response time are met |
| **Derek** | **Budget Management Lead**, responsible for implementation of the PCB on the base module, specifically implementing the microcontroller in coordination with the load cell including the necessary software |
| **Nnaji** | **Software Lead**, responsible for developing the wireless communication between the two modules through an WLAN configuration |


Questions?
TrueBase