UMass Amherst

Equipack

Brenton Chasse CSE

Zach Boynton EE

Alex Nichols CSE

Colin Morrisseau EE

Advisor: Prof. Salthouse

Project Overview - Problem

- People don't know how to properly wear/load their backpacks
  - +7,000 E.R. visits annually
  - ⅓ of 6th graders carry +30% of weight (+10% above recommended limit)
- Health risks include:
  - Vertebral subluxation including herniation
  - Shoulder/neck stress
- Risks can be significantly reduced by:
  - **Reduce stress and strain on human body parts not meant to bear load**
  - Keeping pressure evenly distributed between both shoulder straps
  - Tighten the pack's straps, raising the pack's center of mass up and clos to the wearer's lumbar, relieving pressure from the shoulders

Image source: myphysiosa.com.au/education/backpack-tips-children/

Project Overview - Summary

- **Problem**: The misuse of backpacks poses health risks
- **Solution**:

Part 1.) Create a “smart” backpack (Equi**pack**) featuring:

- 4 pressure sensors on the shoulder straps
- 4 pressure sensors on the back of the backpack
- One load cell securing one lower strap to the backpack
- Embedded electronic system featuring BLE communication for colating and transmitting sensor data to the wearer’s Android device

Part 2.) Teach users how to properly wear their Equi**pack** backpack by:

- Using the wearer’s phone to host a dialog between the user and their Equi**pack**
- Modeling Equi**pack’**s contents as a point mass - determine how Equi**pack** should be adjusted (Force measurements received from sensors over BLE)
- Providing Android app to graphically aid the user in learning how to properly adjust their Equi**pack**

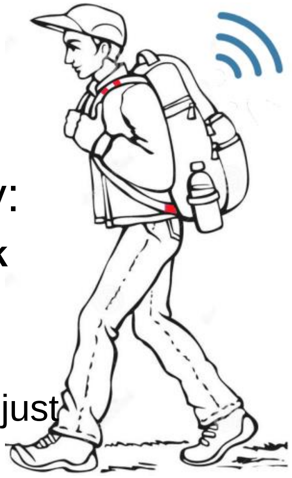
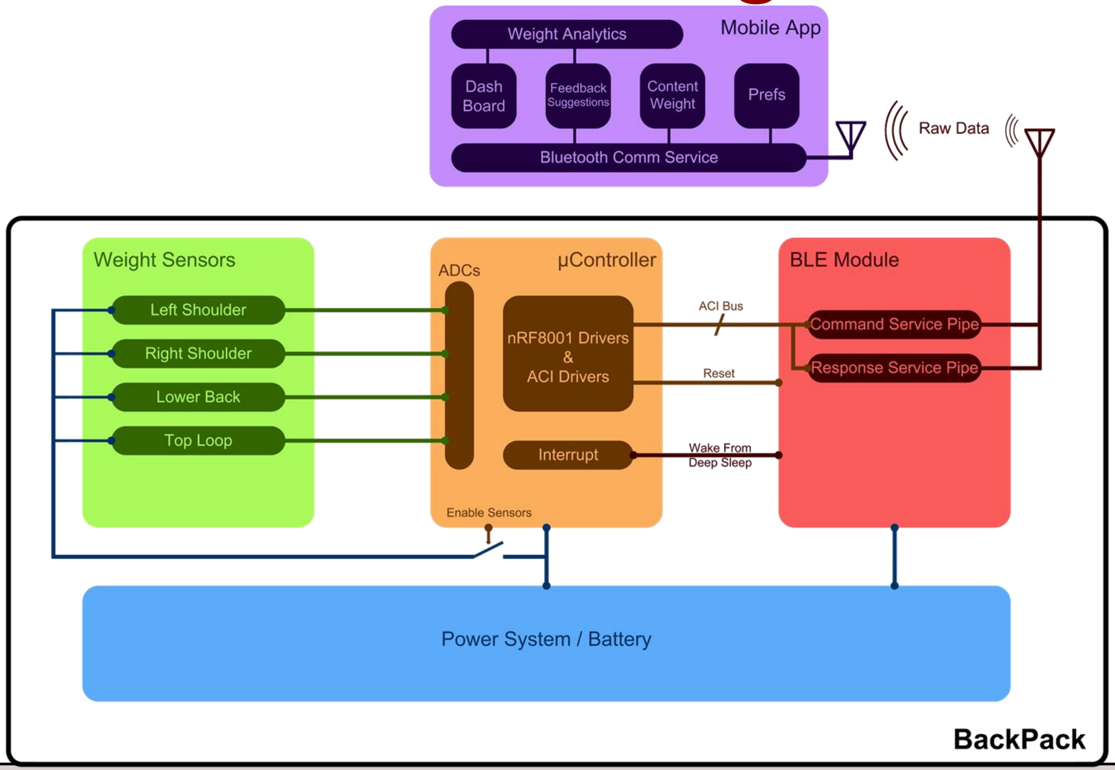


Image source: dreamtime.com stockImages/VectorDrawings

Block Diagram



Proposed CDR Deliverables

- Demonstrate complete system functionality by:

1.) Showing integration between all subsystems

2.) Show implementation of a battery powered system

3.) Having a mobile application with UI elements to display feedback

4.) Show backpack can provide all core functions

CDR Deliverables - Demo Overview

Addressing:

1.) Showing integration between all subsystems

- **(Zach, Colin)** Strain gauge on backpack strap
- **(Alex, Zach)** Embedded system amplifies, samples, colates, and transmits load sensor readings via BLE
- **(Brenton)** Application running on Android device requests and receives strain gauge readings via BLE
- **(Colin)** Analytics library produces weight (in lbs) given strain gauge readings
- **(Brenton)** Application provides user with a simple intuitive interface for controlling the process as well as for visualizing the results.

CDR Deliverables - Demo Overview

Addressing:

2.) Show implementation of a battery powered system

- **(Alex)** Embedded processing system powered off of 4 x 1.5v AA batteries
- **(Colin)** Strap strain gauge and capacitive sensors are powered off of amplification network powered off of 5 volt regulator

CDR Deliverables - Demo Overview

Addressing:

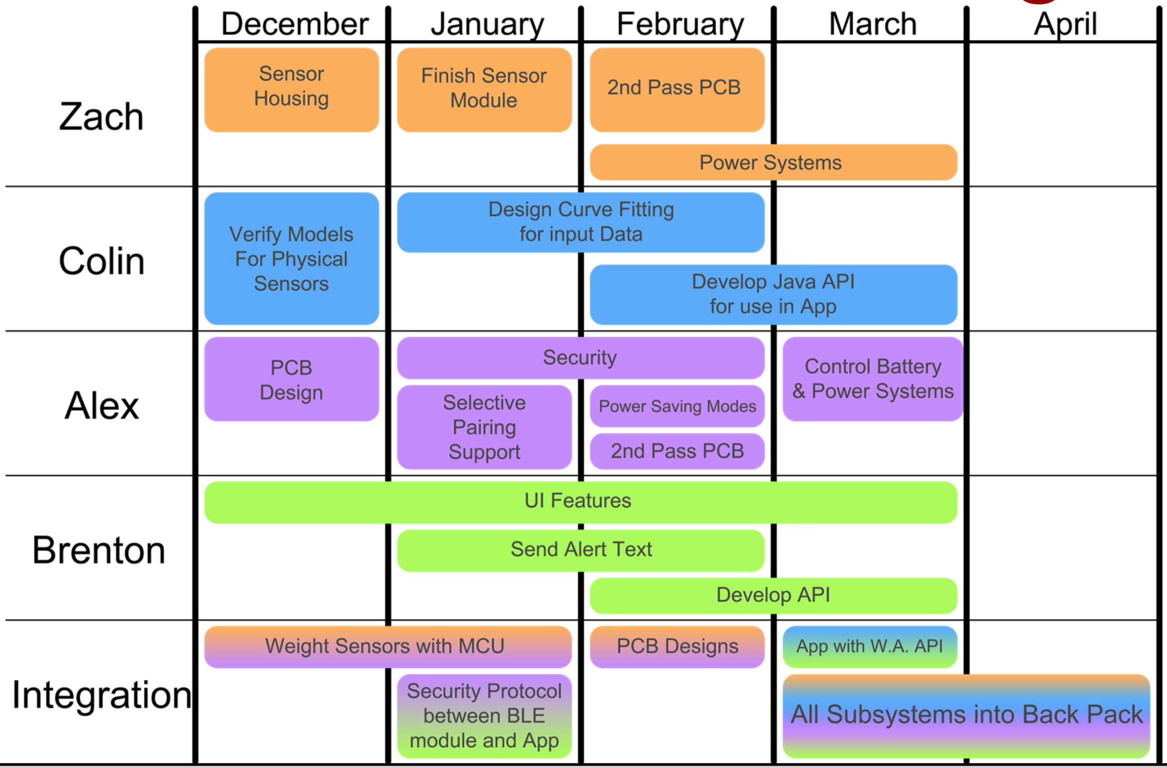
3.) Having a mobile application with UI elements to display feedback

- **(Brenton)** GraphView capable of displaying graph receiving stream of data
- **(Brenton)** TextView capable of displaying a formatted weight
- **(Brenton)** Settings to customize feedback (i.e.: lbs or kg)

4.) Show backpack can provide all core functions

- **(Team)** Refer to "Addressing: 1."

Timeline/Schedule: Integration



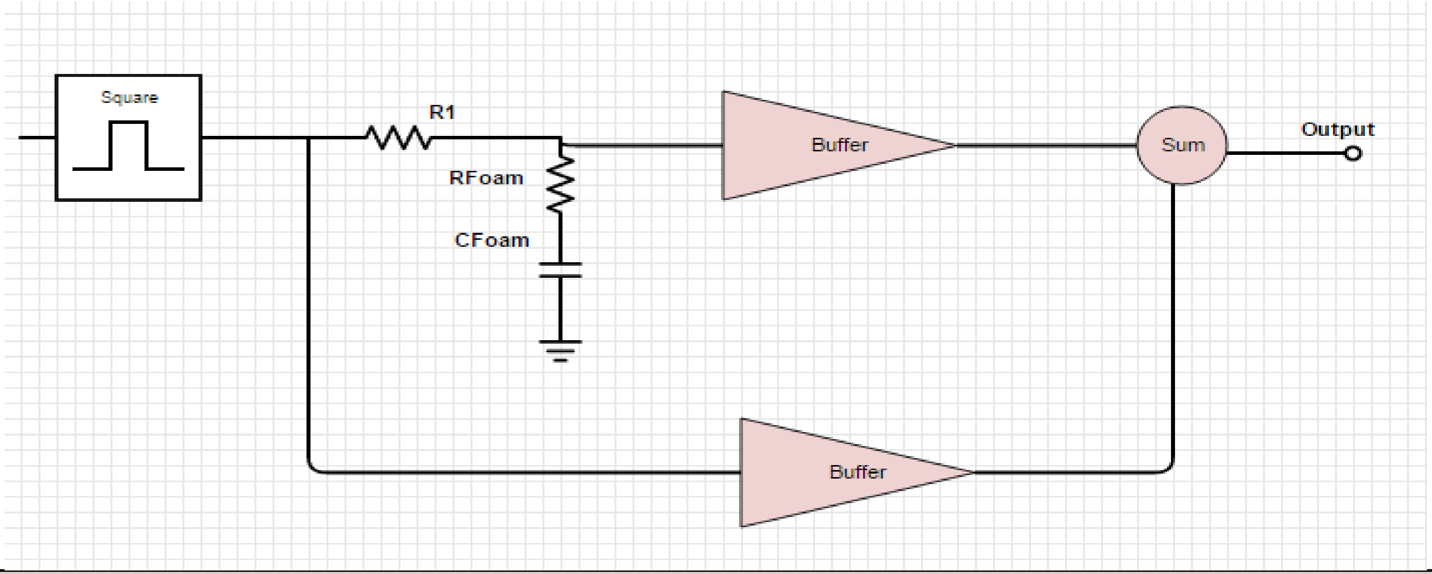
Weight Sensors

- Last time sensors using conductive foam were demonstrated
  - Sensor delay was a serious issue

A few methods were considered to fix this issue
  - RC fitting, other filtering based approaches

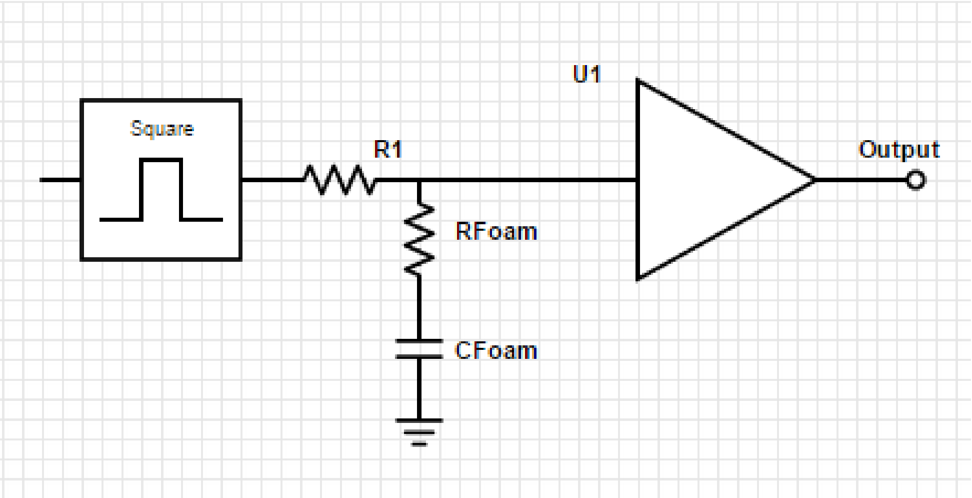
Weight Sensors

- Initially a time domain approach was considered for finding the RC value of the foam



Weight Sensors

- Similarly an approach was taken to detect the pole frequency of the foam



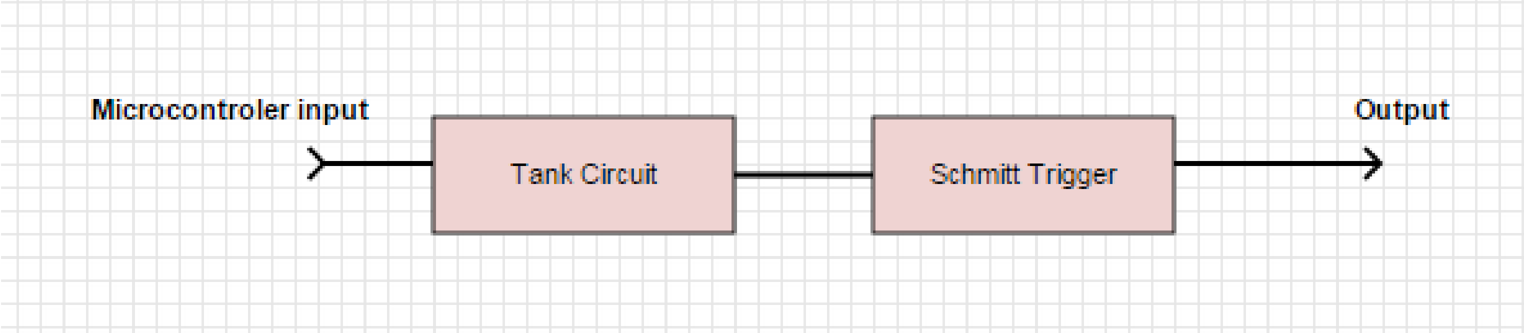
UMassAmherst

Weight Sensors

- Foam was not feasible for taking reasonable measurements
- New ideas were needed
- Capacitance, initially ruled out was reconsidered

Weight Sensors

- By measuring the frequency of a tank circuit we can determine a change in load via a change in frequency



Weight Sensors

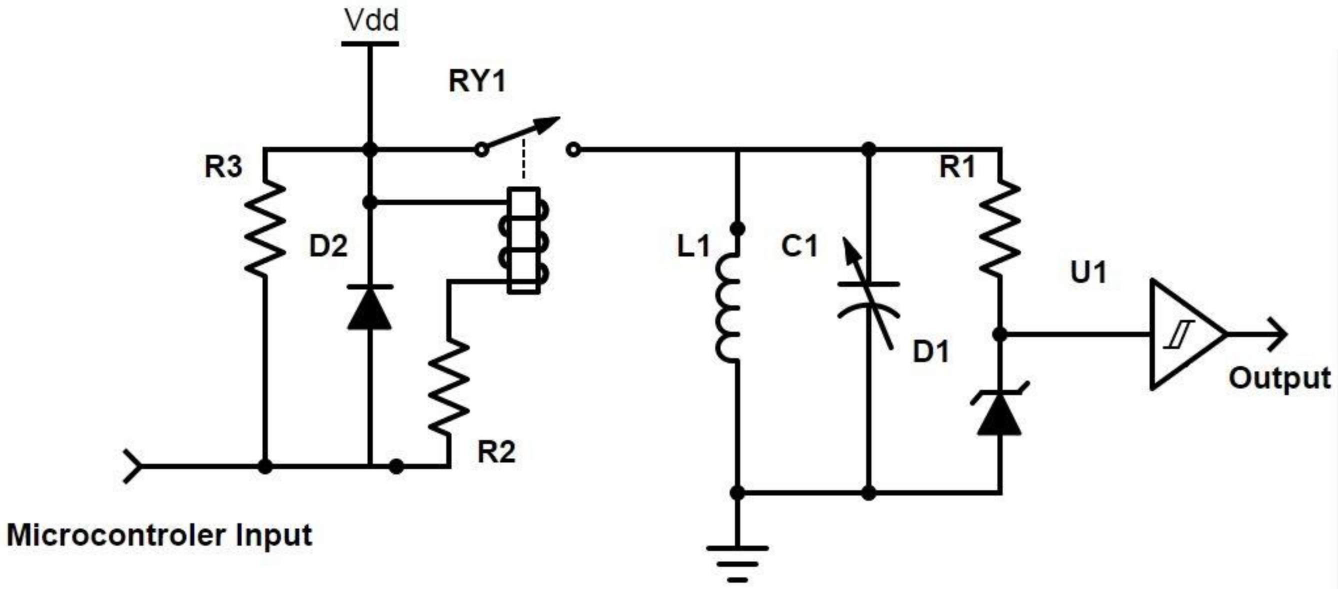
- Schematic diagram

R1=R3=100K

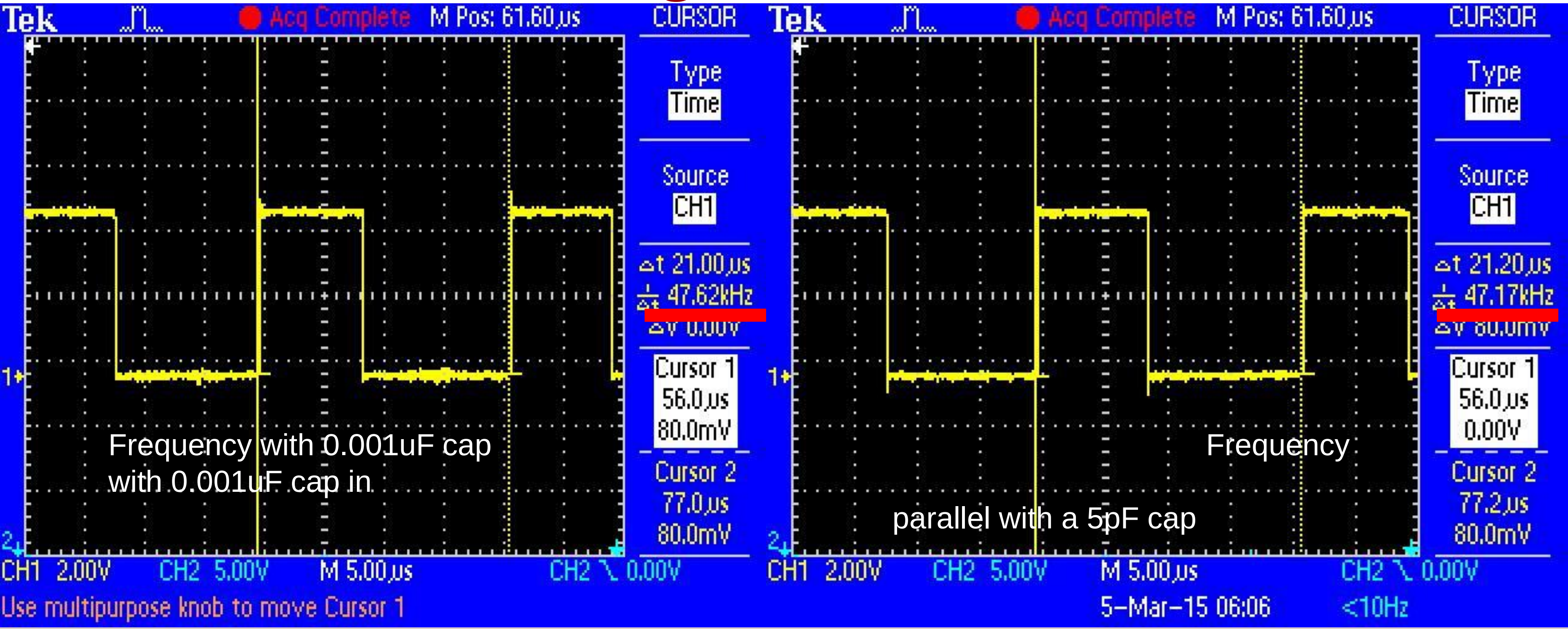
R2=390

L1=10mH

Vdd=5V



Weight Sensors



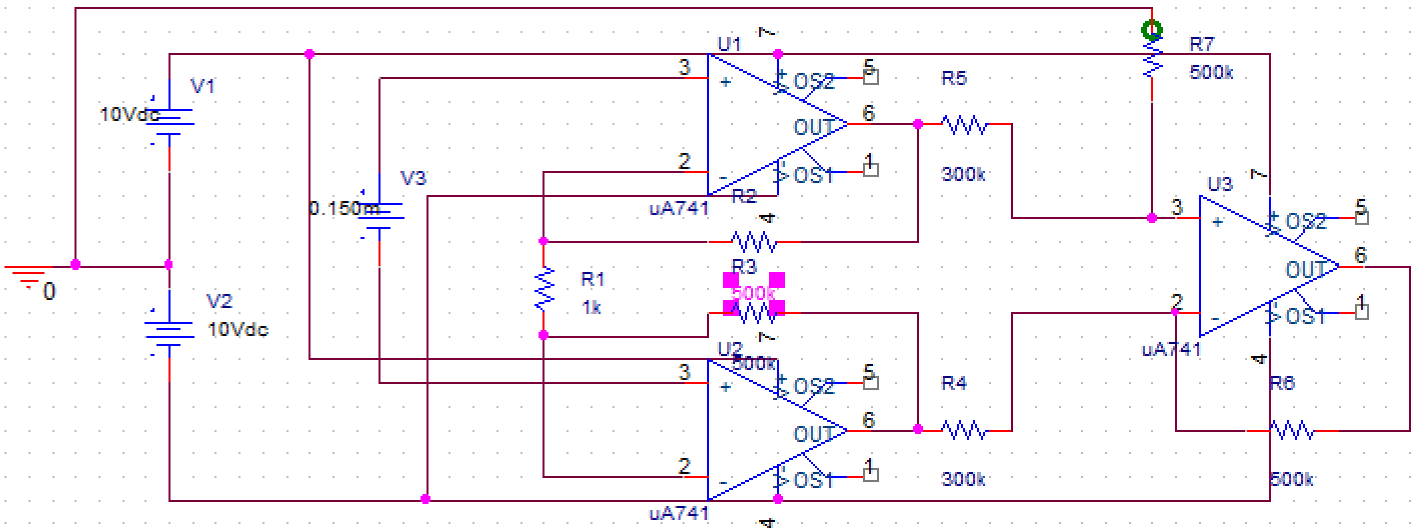
Weight Analytics:Load Cell

- Load Cell provides differential voltage on the scale of microvolts with common mode voltage around 2 volts.
- an Instrumental amplifier is used because input resistance is not a factor in calculating gain
- Amplifier provides a gain of 1665 with extremely high CMRR

Colin Morrisseau

Advisor: Prof. Salthouse

Weight Analytics:Load Cell Schematic



Weight Analytics: Algorithms

- Weight analytics is run by test arrays to simulate static and dynamic conditions
- Weight analytics are run inside the android app as methods in java

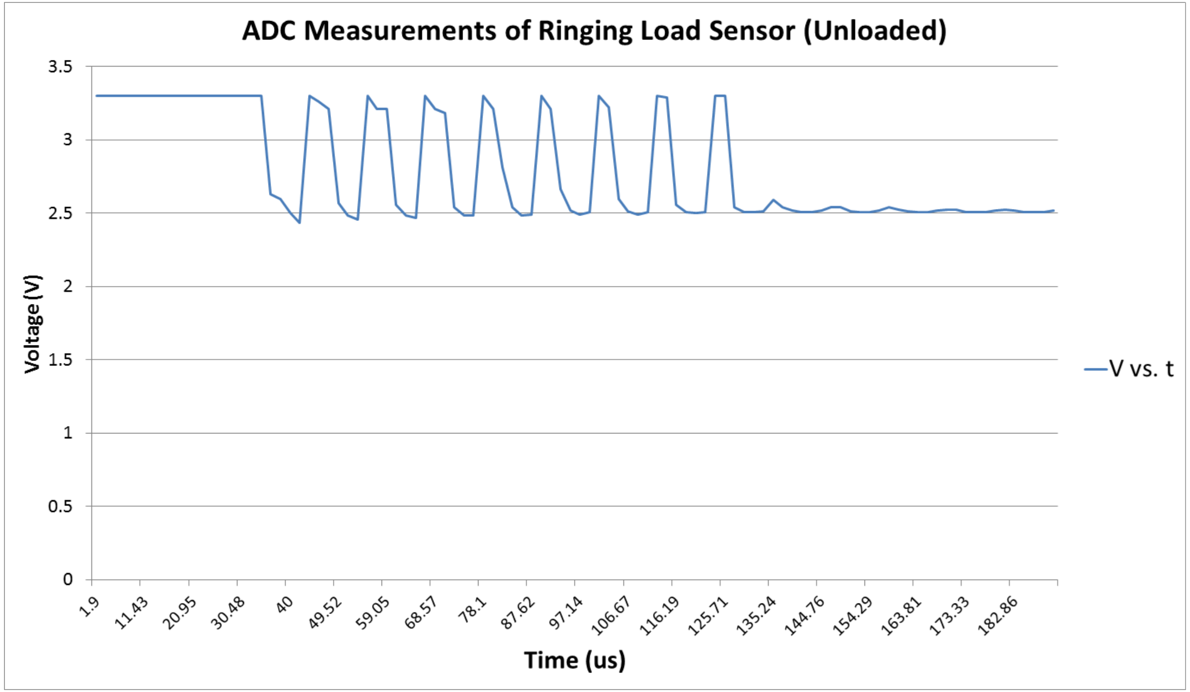
µController and Broadcast

- Implemented 3-bit MUXs to select sensor for ADC input, and to select sensor to excite with pulse
- 4-bit Serial-Parallel IC allows select to occur with one GP Output
- 1 ADC used for Load Cell, 1 for remaining 8 sensors

µController and Broadcast

- Data Collation Methods
  - ADC samples at ~400kHz
  - Load Cell can be read by taking a number of data points and finding mean and variance
  - Foam Load Sensors slightly more complicated: need to read frequency of oscillations. Method: take mean and variance of freq. data; if single-point transition spans reasonable fraction of variance, note as edge. Then take

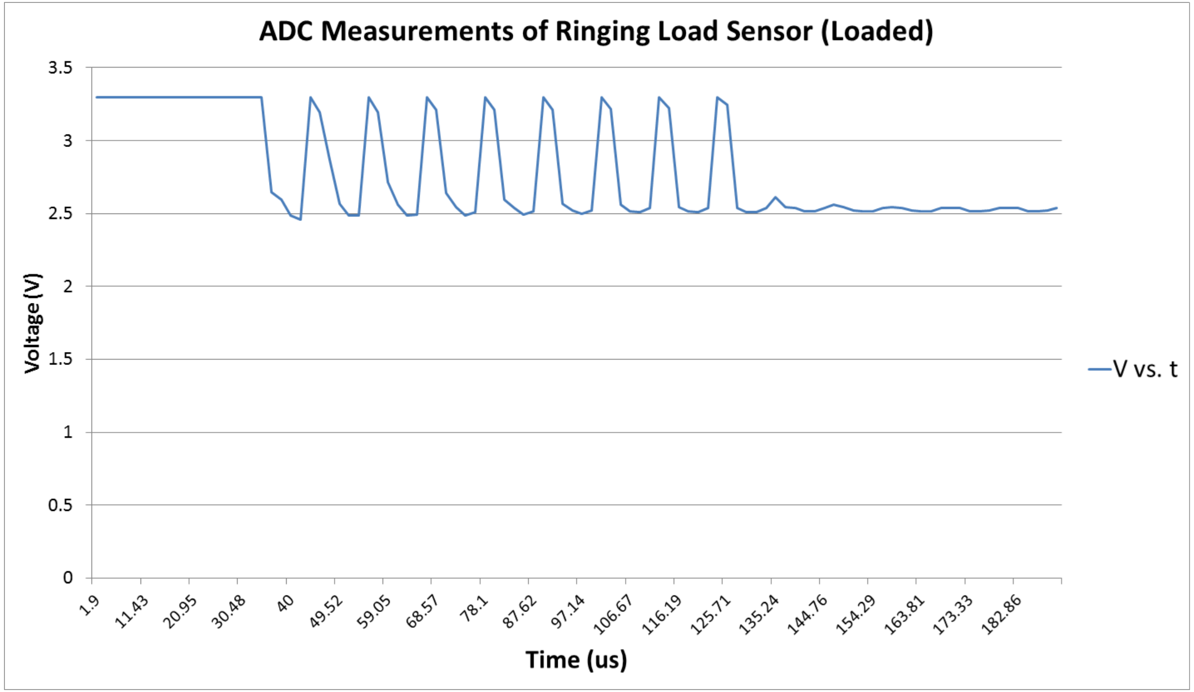
µController and Broadcast



Alex Nichols

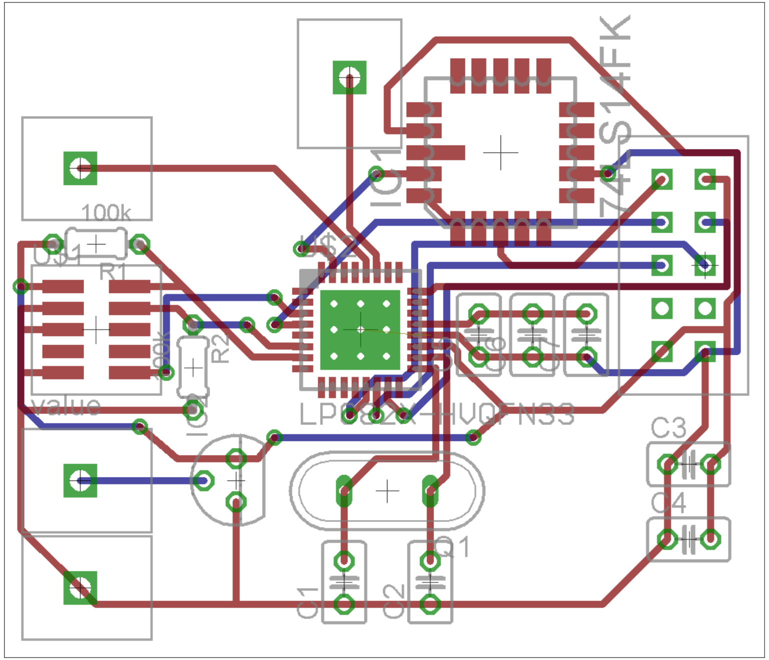
Advisor: Prof. Salthouse

μController and Broadcast



UMassAmherst

PCB Layout



Alex Nichols

Advisor: Prof. Salthouse

µController and Broadcast

- Next Steps
  - Increase ADC sample rate
  - Fabricate and Test PCB

Mobile Application

Addressing Timeline:

January: Implement top-level encrypted communication with µController

Is Bluetooth encryption really necessary?

- BLE data is not truly sensitive (no personal data)
- Already Sent using a one-off boot, command, and response scheme that could deter vandals from spying on transmitted data.

I am: Currently on track.

Timeline/Schedule: Zach

-December: Sensor housing built to handle weight requirement. Start to integrate with microcontroller.

-January: Finish weight sensor module. Continue with microcontroller integration.

-February: Begin power systems work. Begin 2nd pass PCB if required.

-March: Begin integration power systems and sensors into bag.

-April: Final debugging and integration

Timeline/Schedule: Colin

-December: verify models with physical sensors

-January: design curve fitting algorithm to speed up the response time of the sensors

-February: continue previous as necessary

-March: develop API for digital implementation in the smartphone app

-April: final debugging and integration

Timeline/Schedule: Alexander

-December: Integrate µController PCB Design with Weight Sensor PCB design

-January: Keep track of Various Phones, integrate with NVM. Implement top-level encrypted communication with Android Phone

-February: Work On 2nd Pass PCB Design. Start Using Power-saving functionality on µController and BLE module to ensure optimal sleep schedule

-March: Begin integration into Bag; begin using battery for power

-April: Debugging and stability enhancements

Timeline/Schedule: Brenton

-December: More error handling, Start adding basic UI features

-January: Continue adding basic UI features, Sent text message to remote device upon a given condition. Implement top-level encrypted communication with μController

-February: Enhance appearance of UI features, Finish sending text message ensure solid stability of current features. Begin API as required features become defined.

-March: API for interfacing with the UI elements to display equipack calculations

-April: Defect/stability fixes, finish any tasks that have rolled over

Demo

Conclusion

- Questions?