Brenton Chasse  CSE
Alex Nichols  CSE
Zach Boynton  EE
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 closer to the wearer’s lumbar, relieving pressure from the shoulders

Project Overview - Summary

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

Part 1.) Create a “smart” backpack (Equipack) 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 collecting and transmitting sensor data to the wearer’s Android device

Part 2.) Teach users how to properly wear their Equipack backpack by:
- Using the wearer’s phone to host a dialog between the user and their Equipack
- Modeling Equipack’s contents as a point mass - determine how Equipack 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 Equipack

Image source: dreamtime.com stockImages/VectorDrawings
Advisor: Prof. Salthouse

Advisor: Prof. Salthouse
Block Diagram

Weight Sensors
- Left Shoulder
- Right Shoulder
- Lower Back
- Top Loop

ADCs
- nRF9001 Drivers
- ACI Drivers

μController
Interrupt

BLE Module
- Command Service Pipe
- Response Service Pipe

Power System / Battery

BackPack
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
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.
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
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

<table>
<thead>
<tr>
<th></th>
<th>December</th>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zach</td>
<td>Sensor Housing</td>
<td>Finish Sensor Module</td>
<td>2nd Pass PCB</td>
<td>Power Systems</td>
<td></td>
</tr>
<tr>
<td>Colin</td>
<td>Verify Models For Physical Sensors</td>
<td>Design Curve Fitting for input Data</td>
<td>Develop Java API for use in App</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alex</td>
<td>PCB Design</td>
<td>Security</td>
<td>Selective Pairing Support</td>
<td>Power Saving Modes</td>
<td>2nd Pass PCB</td>
</tr>
<tr>
<td>Brenton</td>
<td>UI Features</td>
<td>Send Alert Text</td>
<td>Develop API</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration</td>
<td>Weight Sensors with MCU</td>
<td>Security Protocol between BLE module and App</td>
<td>PCB Designs</td>
<td>App with W.A. API</td>
<td>All Subsystems into Back Pack</td>
</tr>
</tbody>
</table>

Advisor: Prof. Salthouse
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
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

Frequency with 0.001uF cap in parallel with a 5pF cap

Zach Boynton

Advisor: Prof. Salthouse
Weight Analytics: Load Cell

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

Colin Morriseau

Advisor: Prof. Salthouse
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

ADC Measurements of Ringing Load Sensor (Unloaded)

Voltage (V)

Time (us)

V vs. t

Alex Nichols
Advisor: Prof. Salthouse
μController and Broadcast

ADC Measurements of Ringing Load Sensor (Loaded)

Voltage (V)

Time (us)

V vs. t

Alex Nichols

Advisor: Prof. Salthouse
PCB Layout

Advisor: Prof. Salthouse

Alex Nichols
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

Alex Nichols

Advisor: Prof. Salthouse
**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
Conclusion

- Questions?