KineTrax

Team KineJax

Team Members: Anthony Black, Jack Jenkins,
Cherie Parsons, Grant Swenson, and Chris Whitney

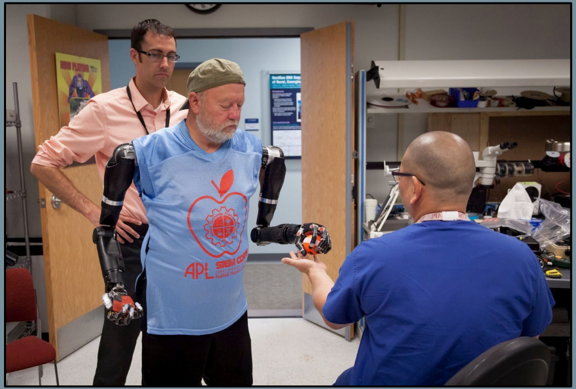
Client/Sponsor: Kyle Winfree

Mentor / Client

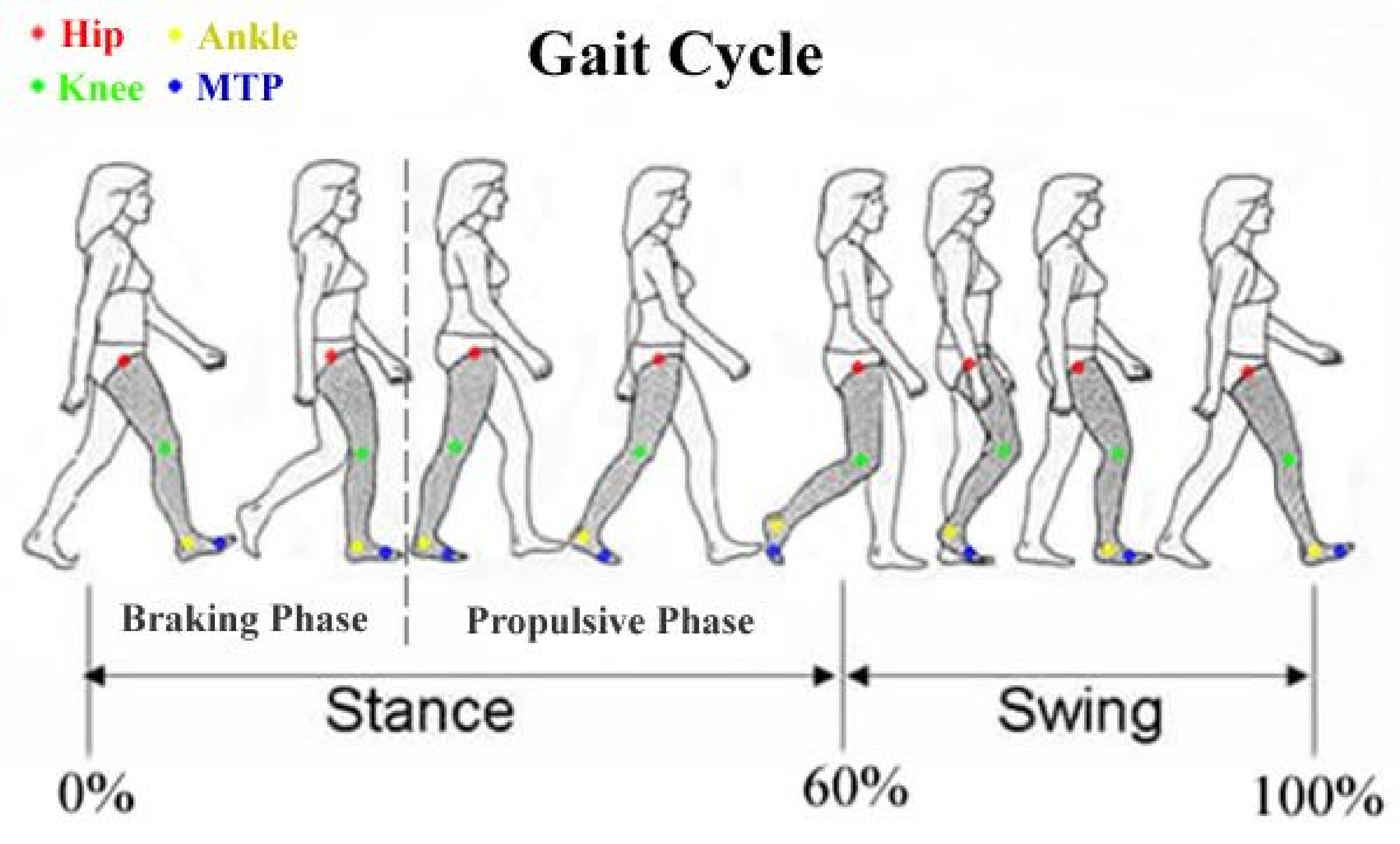
Faculty Mentor / Client

- Kyle Winfree
  - BS Physics
  - MES Robotics
  - PhD Biomechanics and Movement Science
- Part of SICCS department at NAU
- Interested in:
  - Using wearable technology to measure and improve healthcare.

Background - Wearables



Background - Gait Analysis



Problem Statement

Current wearable devices are unable to interface with other devices/sensors and are unable to give the necessary resolution for gait analysis in a community setting.

Limitations

Current wearable devices:

- Unable to interface with other devices
- No synchronization across a distributed network
- Doesn't yield the resolution of measurements as other capturing systems (i.e. Vicon)

	Fitbit	ActiGraph	Vicon
Interface w/ devices	🛇	✓	✓
Sync	🛇	🛇	✓
Resolution	1 min	30-100 Hz	100+ Hz
Cost	Low	Expensive	Expensive

Why is this important?

Information technologies (i.e. data acquisition and storage) is rapidly growing, however, the technologies that sensor and record our movement are not keeping up. This data, if occured could help:

- scientists ask interesting questions
- doctors make more informed and appropriate treatment decisions
- those with movement impairments live normal lives

Solution Overview

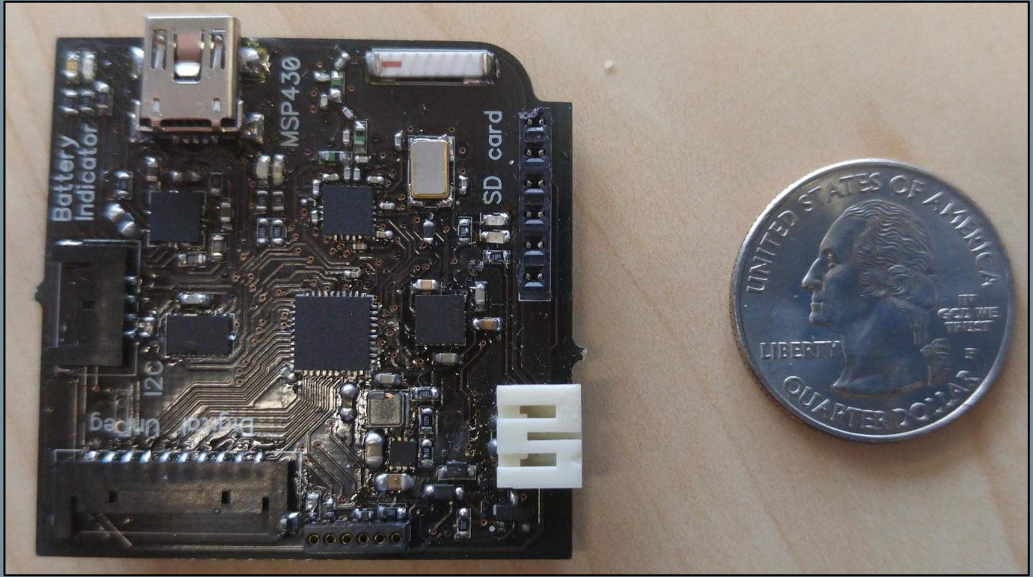
Solution - KineTrax

What the KineTrax offers:

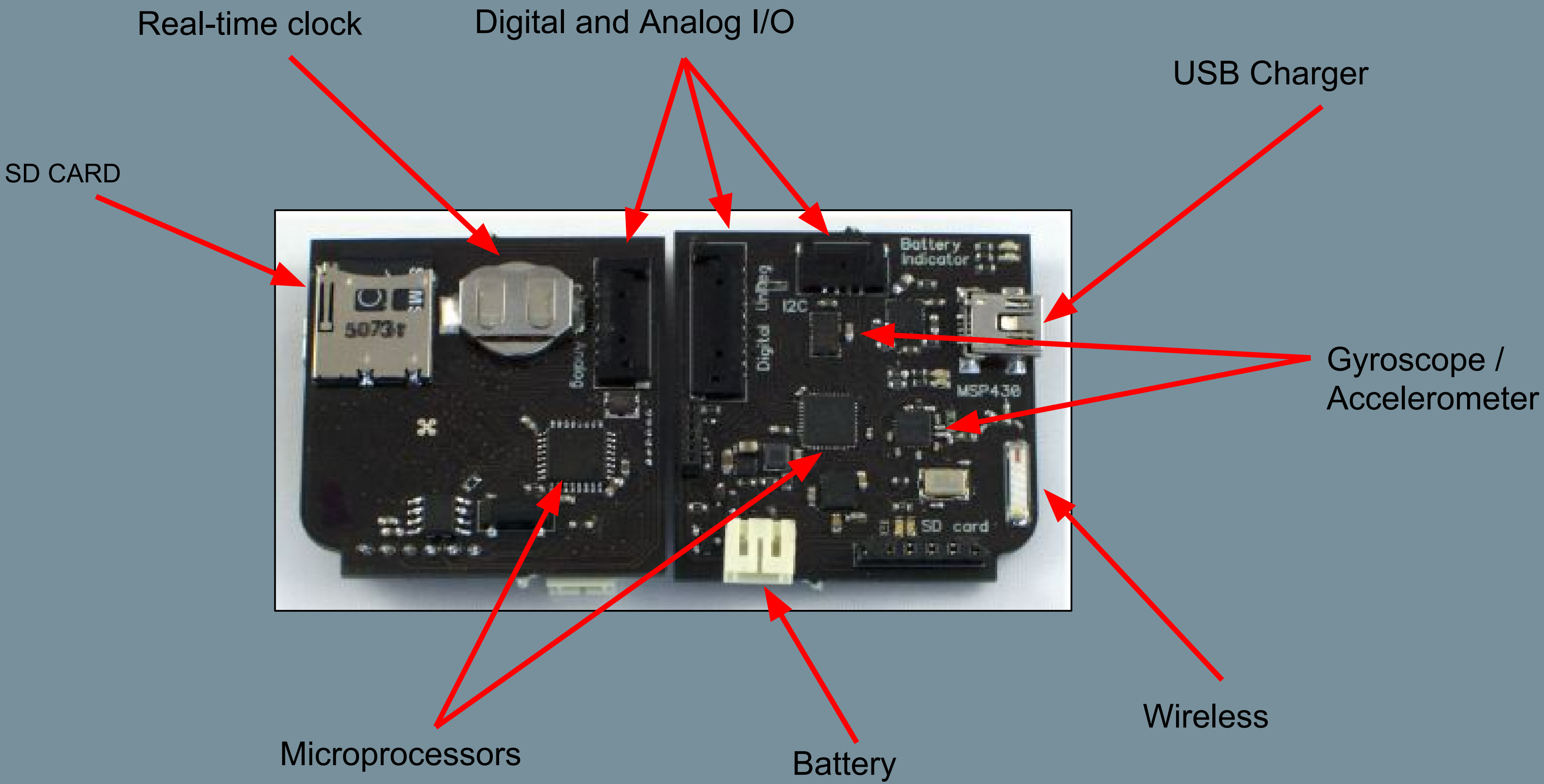
- Digital I/O ports
- Analog I/O ports
- I2C bus, allowing 127 sensors/peripherals

What does this mean?

- Not limited to medical research



Key Components



Real-time clock

Digital and Analog I/O

USB Charger

SD CARD

Gyroscope / Accelerometer

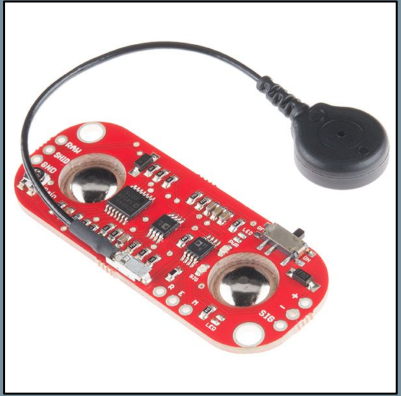
Microprocessors

Battery

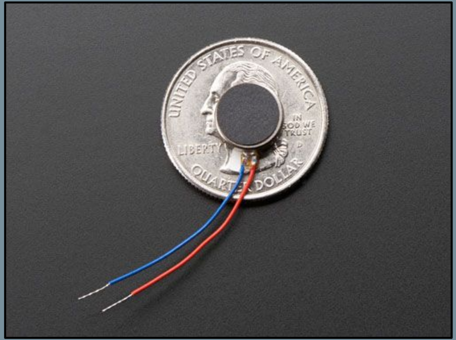
Wireless

Examples of I/O Devices

Input (Sensors, etc.)



Output (Actuators, etc.)



Requirements Review

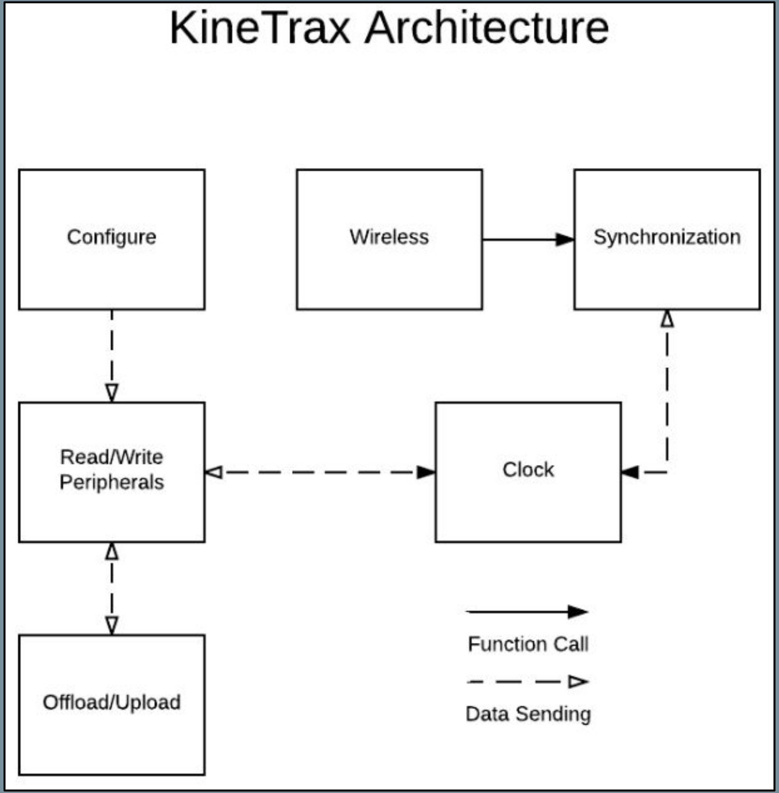
Key Requirements:

- Record/offload timestamped gyroscopic & accelerometer data
- GUI to configure peripherals
- Wireless communication between devices
- Time synchronization between devices

Architecture Overview

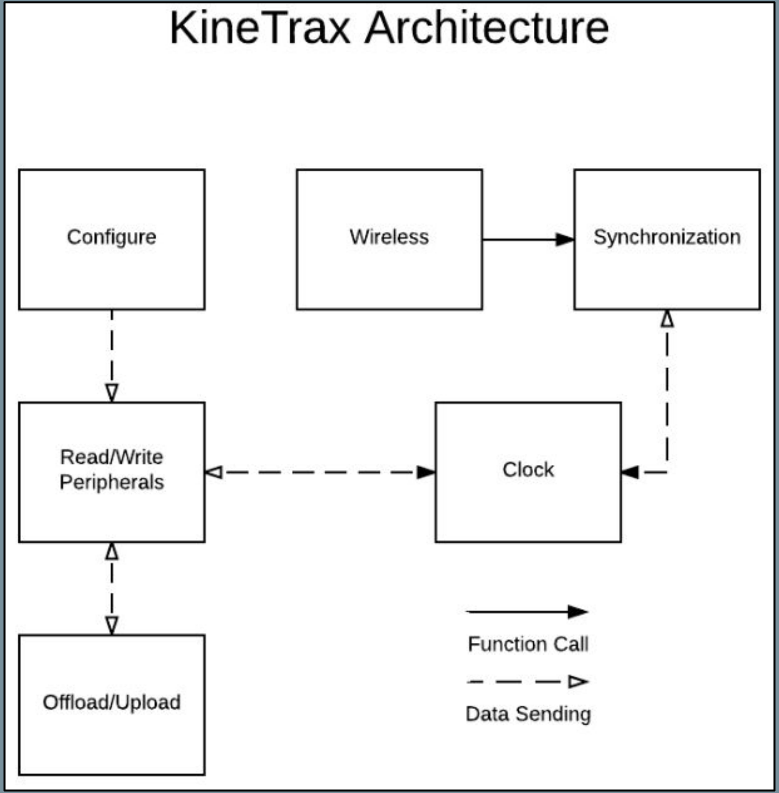
Implementation Overview(Embedded)

- Offload/Upload
  - Reads and writes data from SD card
  - Communicates with connected PC
- Configuration
  - Sets peripheral addresses and sample rates
- Peripherals
  - Peripherals accessed through Inter-Integrated Circuit (I2C) and Universal Asynchronous Receiver/Transmitter (UART)



Implementation Overview(Embedded) (cont.)

- Real-time Clock
  - Timekeeping unit on the device
- Wireless Communication
  - Communicates with other Kinetrax devices in network
- Synchronization
  - Calculates offset between time from Real Time Clock and time received from wireless messages.



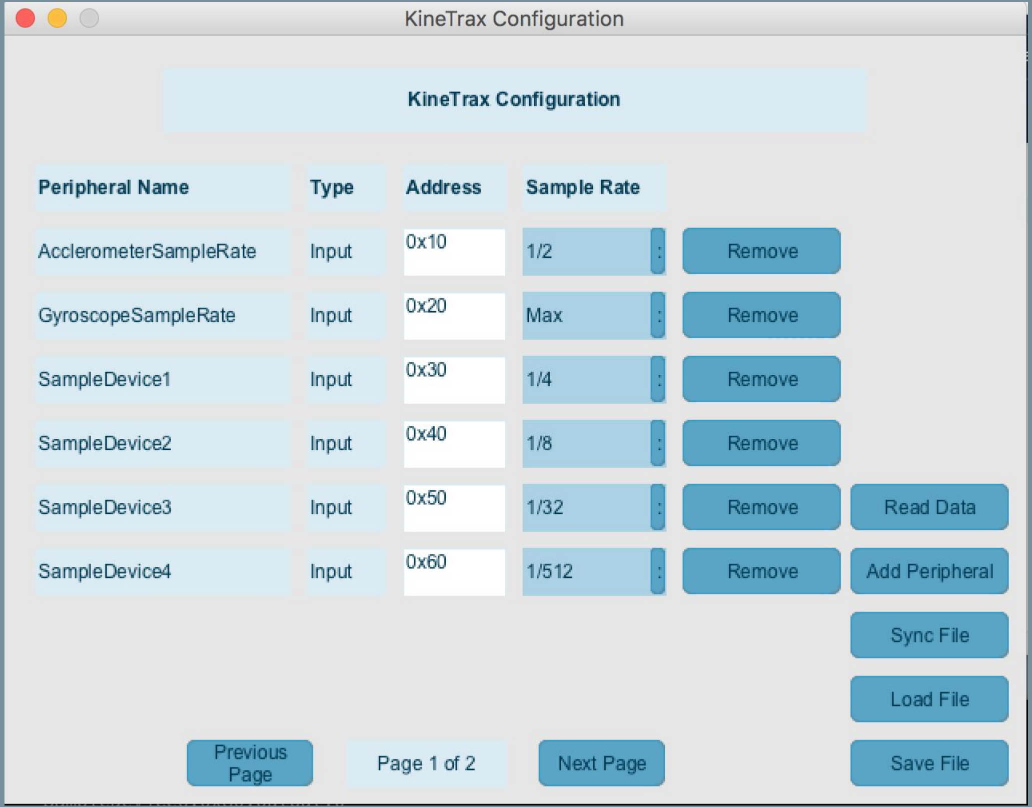
Implementation Overview(PC)

- Offload Data from Device
  - Reading data from device
  - Raw data to CSV

- Configuration
  - Loading from configuration file
  - Saving to configuration file
  - Setting configuration on device

Prototype Video Demo

Prototype - Config GUI



KineTrax Configuration

KineTrax Configuration

Peripheral Name	Type	Address	Sample Rate
AcclerometerSampleRate	Input	0x10	1/2
GyroscopeSampleRate	Input	0x20	Max
SampleDevice1	Input	0x30	1/4
SampleDevice2	Input	0x40	1/8
SampleDevice3	Input	0x50	1/32
SampleDevice4	Input	0x60	1/512

Page 1 of 2

Challenges and Resolutions

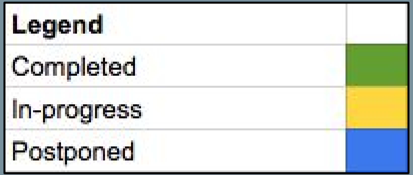
Big Challenges:

- Debugging
- Embedded systems
    - Learning curve for embedded systems
    - Acronyms (UART, I2C, SPI, etc.)
- Hardware issues
    - Sensor issues
    - RTC and wireless can’t work simultaneously
- Existing code has minimal documentation

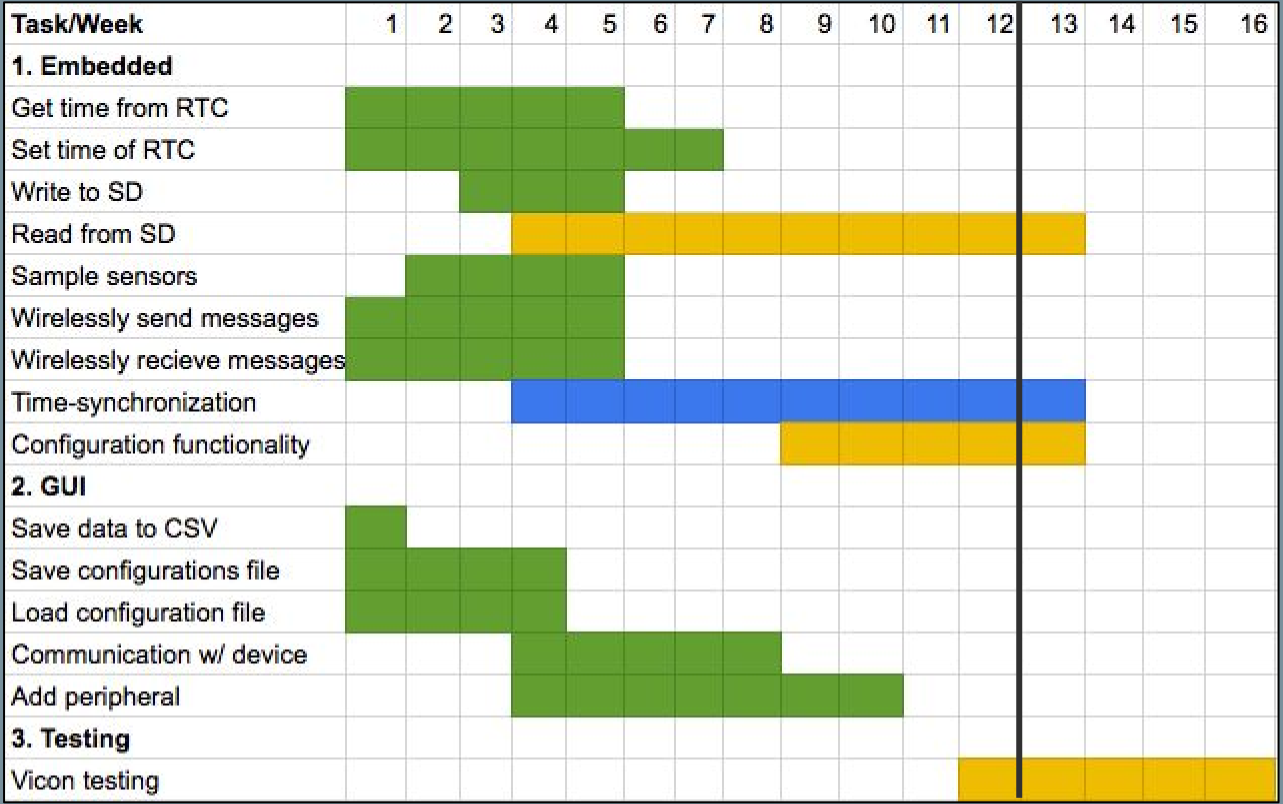
Resolutions:

- Research
    - Existing documentation / sample code
    - Embedded systems books
    - Texas Instruments forums
- External sensors
- Hardware redesign

Schedule



Legend	
Completed	🟩
In-progress	🟨
Postponed	🟦



Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Embedded																
Get time from RTC	🟩	🟩	🟩	🟩	🟩											
Set time of RTC	🟩	🟩	🟩	🟩	🟩	🟩	🟩									
Write to SD			🟩	🟩	🟩											
Read from SD				🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨			
Sample sensors		🟩	🟩	🟩	🟩											
Wirelessly send messages	🟩	🟩	🟩	🟩	🟩											
Wirelessly recieve messages	🟩	🟩	🟩	🟩	🟩											
Time-synchronization				🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦			
Configuration functionality									🟨	🟨	🟨	🟨	🟨			
2. GUI																
Save data to CSV	🟩															
Save configurations file	🟩	🟩	🟩	🟩												
Load configuration file	🟩	🟩	🟩	🟩												
Communication w/ device				🟩	🟩	🟩	🟩	🟩								
Add peripheral				🟩	🟩	🟩	🟩	🟩	🟩	🟩						
3. Testing																
Vicon testing												🟨	🟨	🟨	🟨	🟨

Testing Plan

Unit Testing

- Wireless sending and receiving
- Writing and reading from the SD
- Setting and reading the real-time clock

Integration Testing

- Vicon system at the Human Performance Lab
  - Compare the accelerometer/gyro for different movements

Usability Testing

- Testing GUI with 4 participants

Conclusion

KineTrax has the potential to benefit lots of areas:

- Movement impairments
- Prosthetic limbs
- Sports medicine
- Farm animals

Lots of progress has been made:

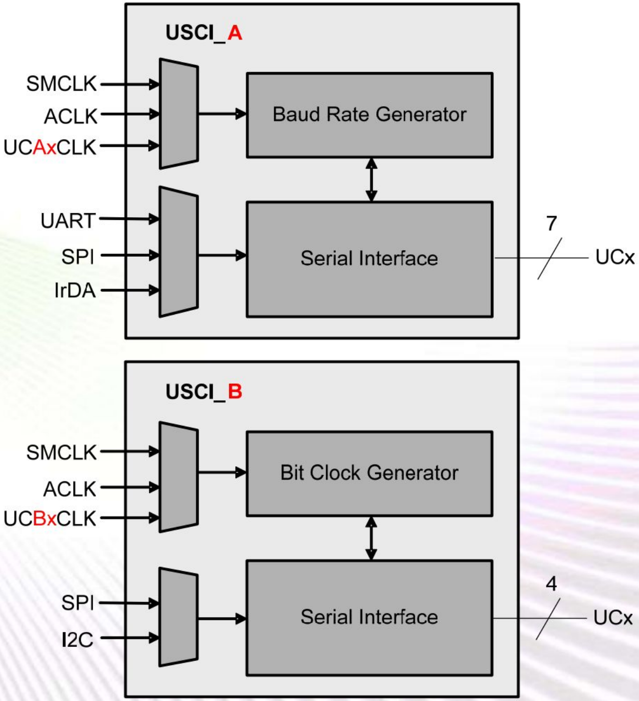
- Device can sample sensors
- Device can get time from RTC
- Device can wirelessly communicate
- Writing samples to SD
- GUI loads configuration file
- GUI can save configuration file

References

1. http://evenamed.com/eyes-on-glasses/
2. http://www.fabricatingandmetalworking.com/2015/02/waterjet-used-make-prosthetic-limb/Winfree
3. http://www.pcclassesonline.com/fitbit-charge-hr-vs-surge-review-and-tutorial/
4. http://www.operationward57.org/2011/03/new-bionic-prosthetic-for-lower-leg-amputees/
5. http://toughasia.com/blog/wearable-tech-vylyv-labs-smart-shorts-enable-men-to-strengthen-pelvic-floor-muscles/
6. https://www.amazon.com/Programming-Embedded-Systems-Development-Tools/dp/0596009836
7. https://www.pearsonhighered.com/program/Gajski-Specification-and-Design-of-Embedded-Systems/PGM30685.html
8. http://www.quintic.com/education/case_studies/Gait%20Analysis.html
9. https://electrosome.com/wp-content/uploads/2014/12/Force-Sensing-Resistor-0.5.jpg

What is the USCI?

- **Universal Serial Communication Interface**
- **Two independent blocks**
- **USCI_A:**
  - UART
  - UART with automatic Baud rate detection (LIN support)
  - IrDA (SIR - Slow InfraRed)
  - SPI (Master & Slave, 3 & 4 wire)
- **USCI_B:**
  - I2C (Master & Slave modes)
  - SPI (Master & Slave, 3 & 4 wire)
- **In high-end 2xx and 4xx devices**


USCI_A
SMCLK
ACLK
UCAxCLK
Baud Rate Generator
UART
SPI
IrDA
Serial Interface
7
UCx
USCI_B
SMCLK
ACLK
UCBxCLK
Bit Clock Generator
SPI
I2C
Serial Interface
4
UCx