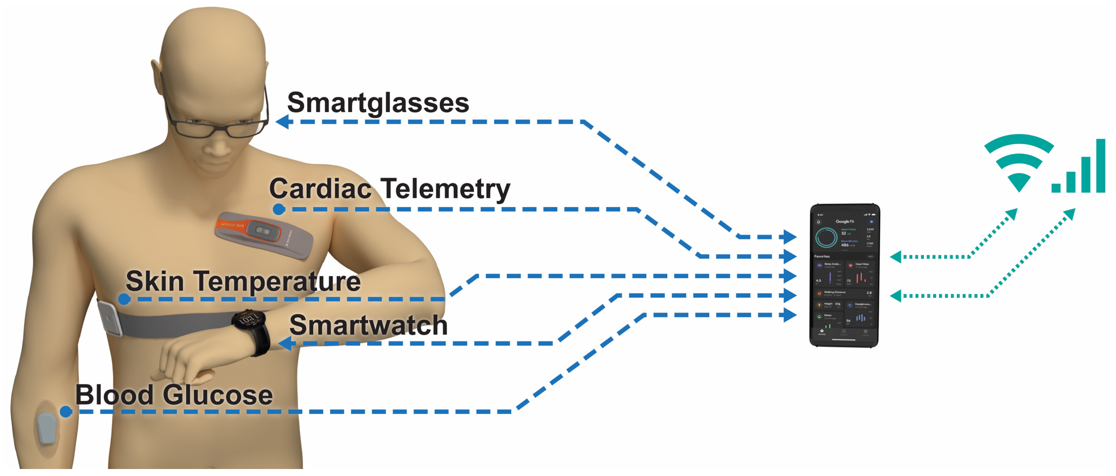
An Extensible Body Sensor Network Platform with Real-Time 3D Visualization

Myles Keller
Dr. Robert Karam

Introduction

A Body Sensor Network (BSN) is a system consisting of low-power, often wireless sensor nodes used to monitor various aspects of the wearer's body and the environment surrounding them

Fig. 1: Typical Body Sensor Network Configuration



Motivation

Most existing BSN research focuses on optimizing individual system components or on healthcare-specific applications

Our focus is on designing an extensible BSN platform that supports research and development of emerging wearable and implantable applications

Novel features include support for:

- Dynamic microcontroller role reconfiguration
- Real-time, multi-dimensional data visualization
- Platform-independent, browser-based front-end for developers
- Support for multi-sensor energy optimization strategies

UNIVERSITY of SOUTH FLORIDA

Browser Front-end

Info Overview ●

Shows unique name and icon, current value(s), and visibility state of all sensors configured in environment

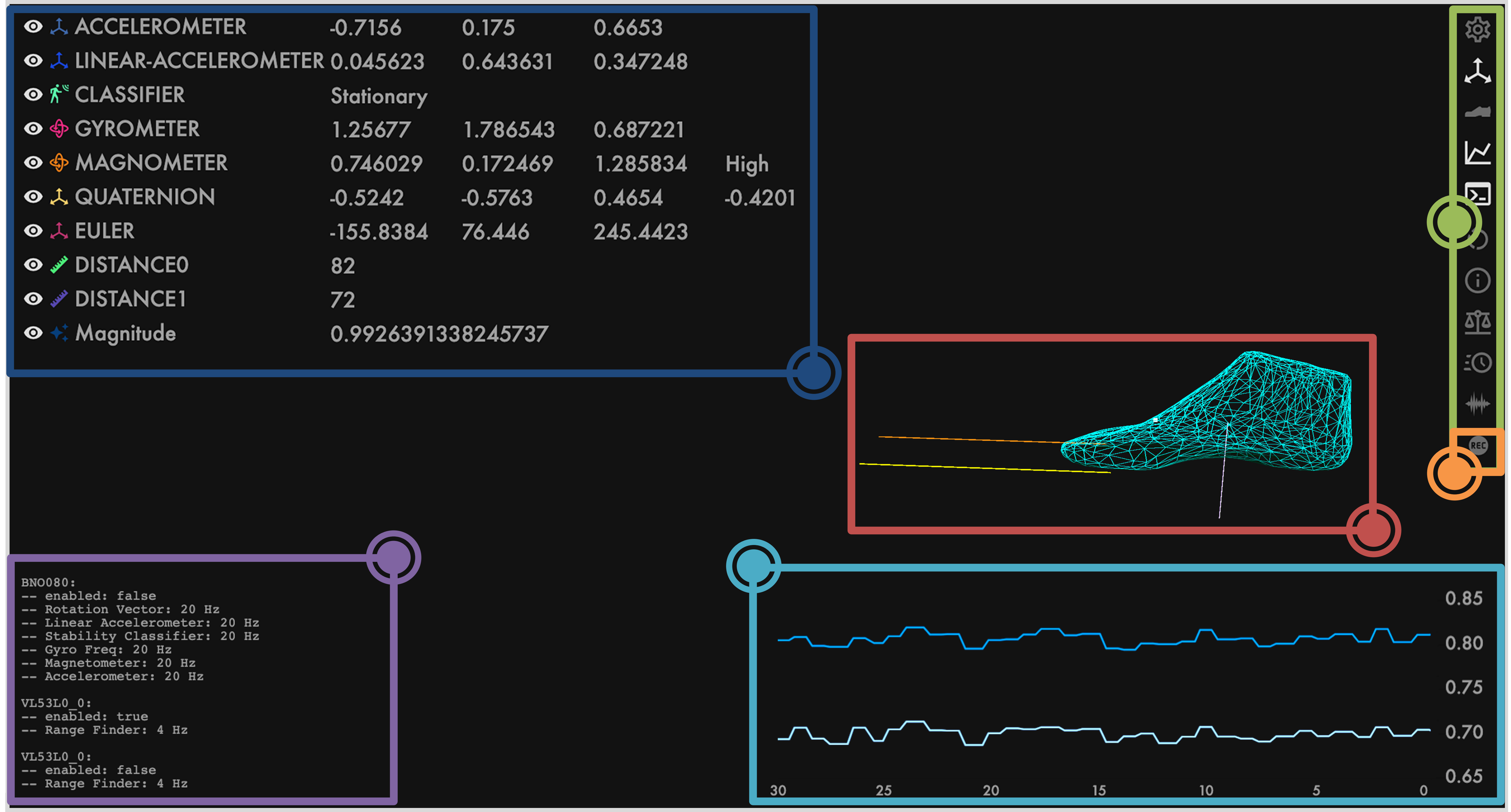
3D Visualizer ●

Applies spatial, location, orientation, and movement sensor data to symbolic objects in 3D space

UI Buttons ●

Allows user quick access to preconfigured or custom JavaScript commands for sensors and the front-end

Fig. 2: Screenshot Displaying Each Component of Browser Front-End



ACCELEROMETER	-0.7156	0.175	0.6653	
LINEAR-ACCELEROMETER	0.045623	0.643631	0.347248	
CLASSIFIER	Stationary			
GYROMETER	1.25677	1.786543	0.687221	
MAGNOMETER	0.746029	0.172469	1.285834	High
QUATERNION	-0.5242	-0.5763	0.4654	-0.4201
EULER	-155.8384	76.446	245.4423	
DISTANCE0	82			
DISTANCE1	72			
Magnitude	0.9926391338245737			

```
BNO080:
-- enabled: false
-- Rotation Vector: 20 Hz
-- Linear Accelerometer: 20 Hz
-- Stability Classifier: 20 Hz
-- Gyro Freq: 20 Hz
-- Magnetometer: 20 Hz
-- Accelerometer: 20 Hz

VL53L0_0:
-- enabled: true
-- Range Finder: 4 Hz

VL53L0_0:
-- enabled: false
-- Range Finder: 4 Hz
```

Virtual Console ●

Provides debugging information from sensors and allows input of simple user commands

2D Graphing ●

Plots streaming sensor data for easier identification of patterns, anomalies, and any correlations

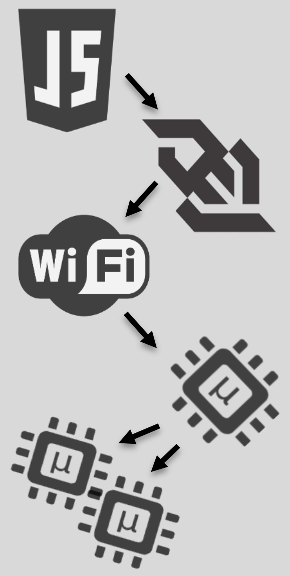
Data Export ●

Saves streaming sensor data to local file system for future offline processing and analysis

Communication Back-end

Sensor Interfacing

- JavaScript constructors are used to initialize sensors for use in BSN
- NODE microcontrollers report data up to the CONTROLLER microcontroller
- PlatformIO used for one-time flash of C++ firmware for initial configuration



Communication Protocol

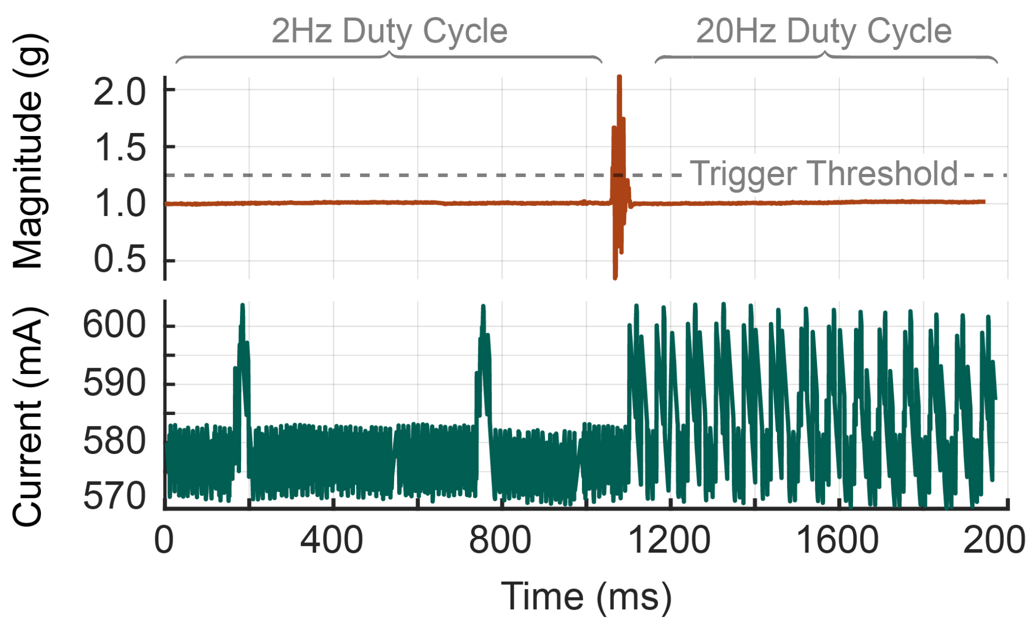
- Browser Front-end, controllers, and nodes communicate via Wi-Fi
- WebSocket parses information between Browser Front-end and Wi-Fi for reliable, bi-directional communication with minimal lag

Results

Sampling Rate Optimization

- Sensor passes ID of another sensor as a reference of how to modulate its internal sampling rate
- Sensor scales back internal duty cycle proportionally to values from reference
- Observed **14% reduction** in overall microcontroller power consumption

Fig. 3: Effect of Sample Rate Modulation on Current Draw

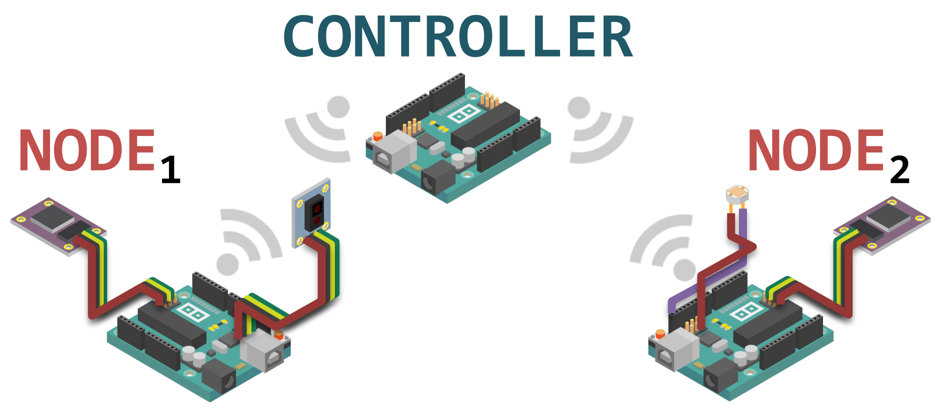


Microcontroller Role Flexibility

We observed appropriate responses from the microcontroller according to each role assignment it can perform in the BSN:

- NODE: microcontroller hub for one or more physical or virtual sensors
- CONTROLLER: microcontroller providing communication link between NODEs and Browser Front-end

Fig. 4: Microcontroller Role Assignment Hierarchy



Wireless Data Streaming

- CONTROLLER hosts ad-hoc Wi-Fi network for sending/receiving data on BSN and hosting Browser Front-end
- Measurements from the BSN rendered in real-time with a **15 Hz** refresh rate when transmitting via Wi-Fi