ECE 445: Spring 2016

Group #39

Posture Corrector

Vishal GVK
Vignesh Sridhar
Shubham Agarwal

Posture Corrector Introduction

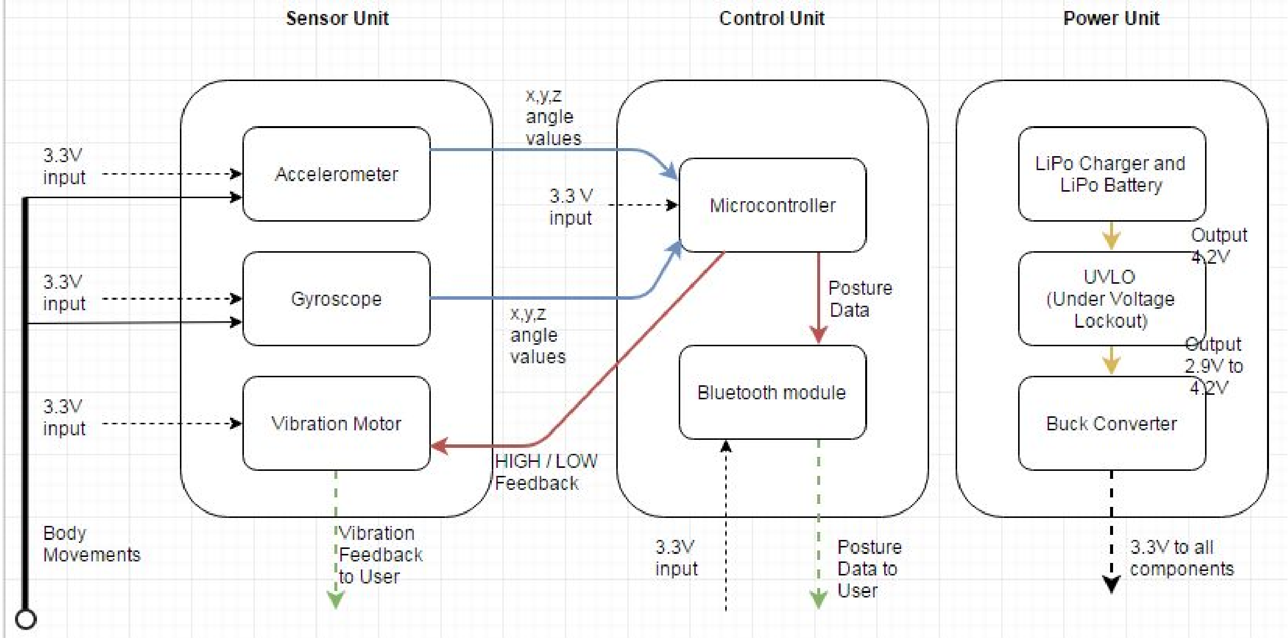
Problem - Back Pain due to Poor Posture

- *National Center for Biotechnology Information* states that acute episodes of back pain are associated with muscle strain.
- Lower back pain is the second most common reason for visits to a doctor, outnumbered only by upper-respiratory infections. Americans spend at least $50 billion each year on back pain.

Cause: People maintain poor posture; strain their back unnecessarily

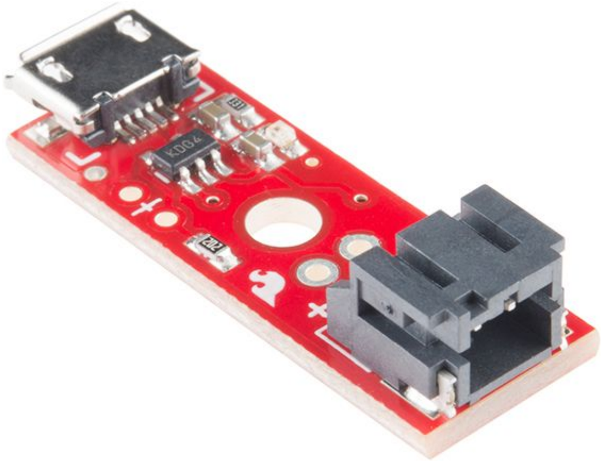
Solution - A wearable platform that provides feedback regarding the wearer's posture

Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

LiPo Charger and Battery

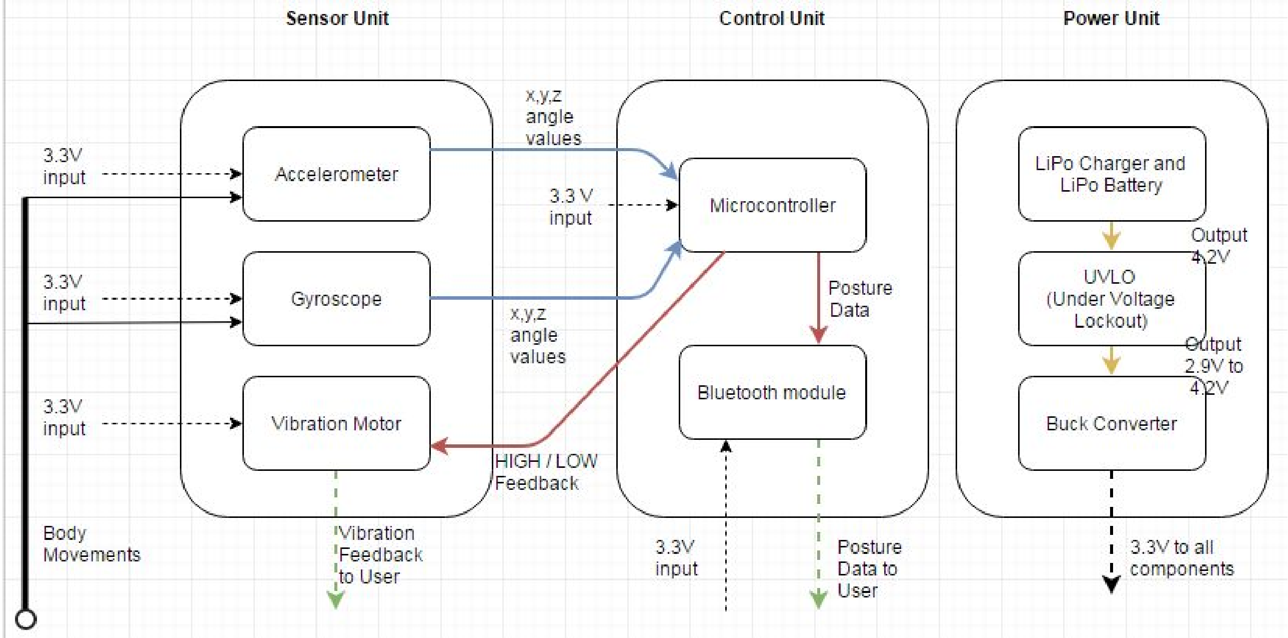


Sparkfun LiPo Charger Basic
Micro-USB
MCP73831/2

Charging circuit with status LED, and a micro-usb connector

Sparkfun Polymer Lithium Ion Battery
GSP652535
400mAh

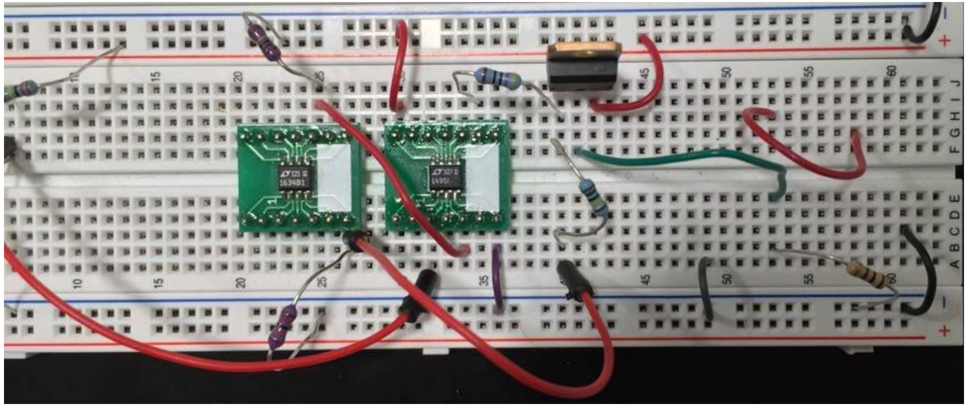
Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

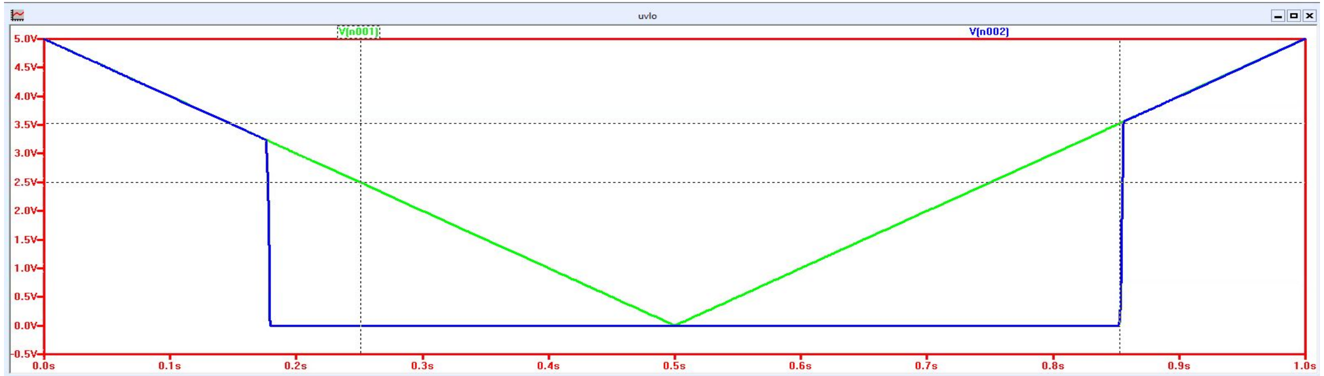
UVLO

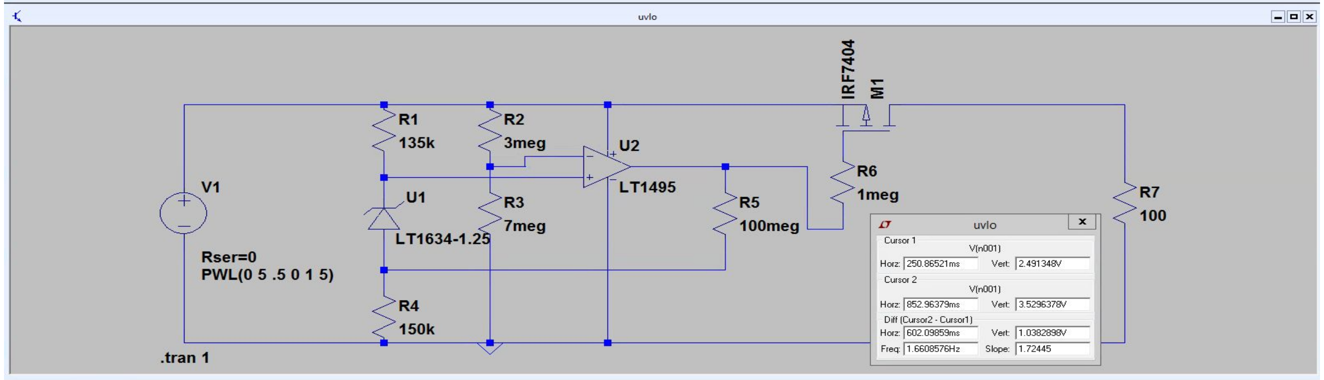
Made a circuit which does the following:



- Turns off the output power to the circuit if the operational voltage of the LiPo battery drops below 3.1 V
- Turns on the output power to the circuit if the operational voltage of the LiPo battery rises above 3.5V, due to hysteresis
- The start voltage is chosen a lot higher than the cut-off voltage to avoid rapid switching on/off when battery reaches 3.1V

UVLO - a LTSpice simulation





Output voltage drops when battery voltage drops slightly below 3.1 V

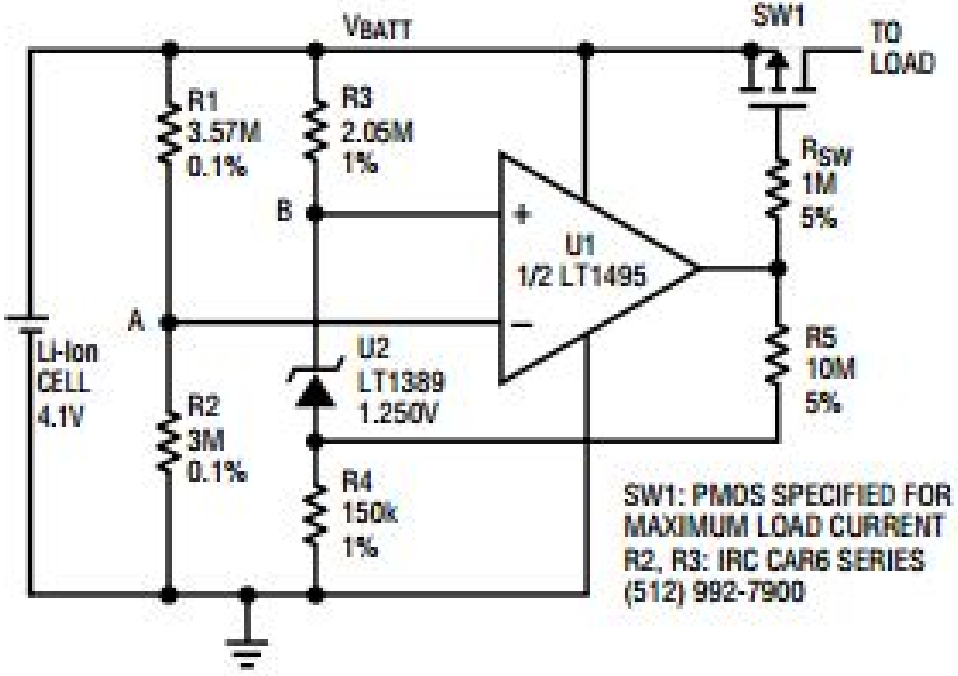
Output voltage only rises after battery output crosses 3.5 V

Circuit composes of:

- Shunt voltage reference
- Op amp comparator
- Pmos

Green line	PWL (piecewise linear) of source voltage
Blue line	Output voltage across R7 (Load)

UVLO - Calculations



Let Vt = lockout voltage

Vb = 1.25V + I.R4 = 1.37V

I = (Vt - 1.25V) / (R3 + R4) = 800 nA

Solving above equations

1.37 = 1.25 + 800 * 10^(-9) . R4

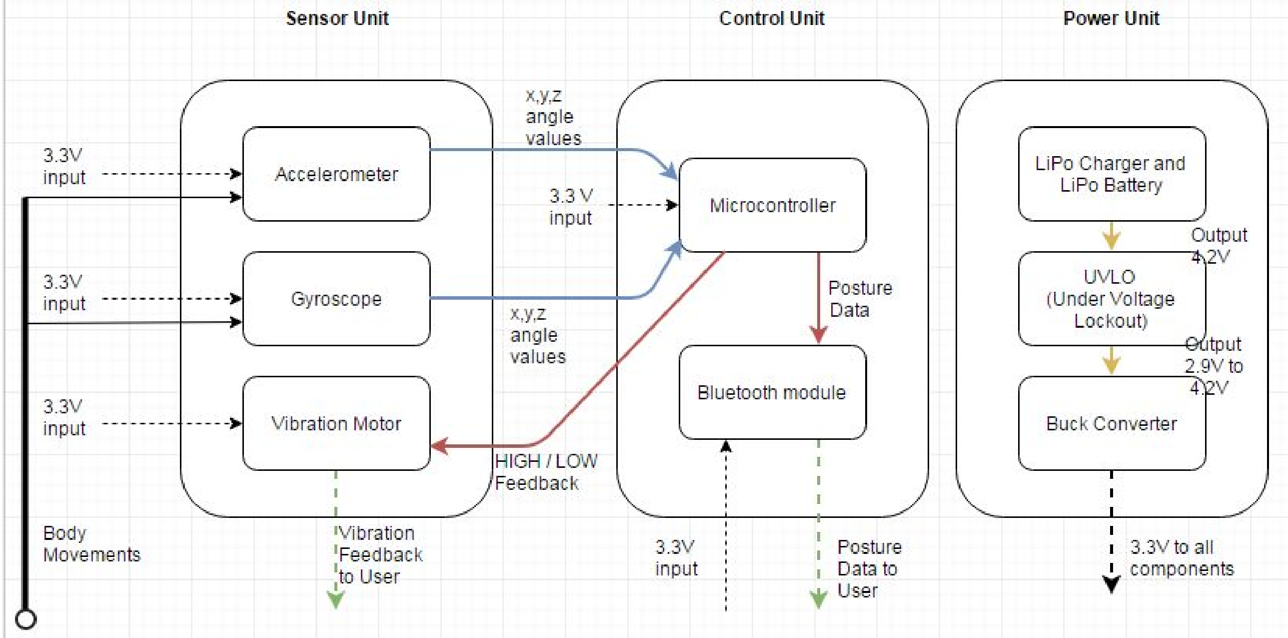
R4 = (0.12 * 10^9) / 800

R4 = 150000 ohm = *<u>150k ohm</u>*

UVLO - Requirements and Verifications

1. For an active (not charging) circuit with an input of 1.55 ± 1.55 V; the output of the UVLO should be 0 V.	• Use a function generator to supply 4.0 ± 0.2 V to the UVLO and slowly drop it by 0.1 V every 5 seconds. • Use an oscilloscope (positive of oscilloscope on positive output of UVLO and negative of oscilloscope on negative output of UVLO. • oscilloscope should show 0 V.
1. For an inactive (charging) circuit with an input of 1.90 ± 1.90 V; the output of the UVLO should be 0 V.	• Use a function generator to supply 1.5 ± 1.5 V to the UVLO and slowly increase it by 0.1 V every 5 seconds. • Use an oscilloscope (positive of oscilloscope on positive output of UVLO and negative of oscilloscope on negative output of UVLO. • oscilloscope should show 0 V till it reaches 3.8 V and input voltage as output voltage after that.

Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Buck Converter

Made a circuit which does the following:

- DC-to-DC power converter
- Steps down voltage from 3.75 ± 0.45V to 3.3 ± 0.1 V

Input: 3.901 V
Output: 3.379 V

Values from our circuit

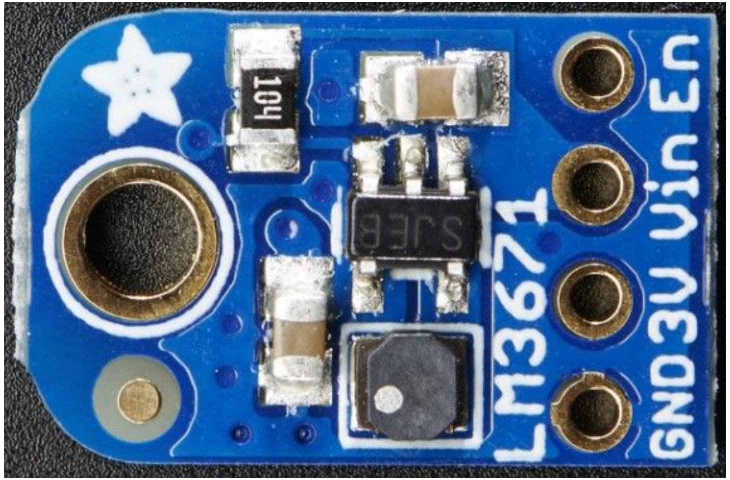
Buck Converter - Calculations

- $P = VI$
- P_{in} = 4.2 [V] * 0.1 [Amps]
- P_{in} = 0.42 [W]
- $P_{out} = V^2/R$
- P_{out} = $(3.379\ [V])^2$ / 30 [ohm]
- P_{out} = 0.3805 [W]
- Efficiency = (P_{out}/P_{in}) * 100 %
- Efficiency = (0.3805 / 0.42) * 100 %
- Efficiency = 90.59 %

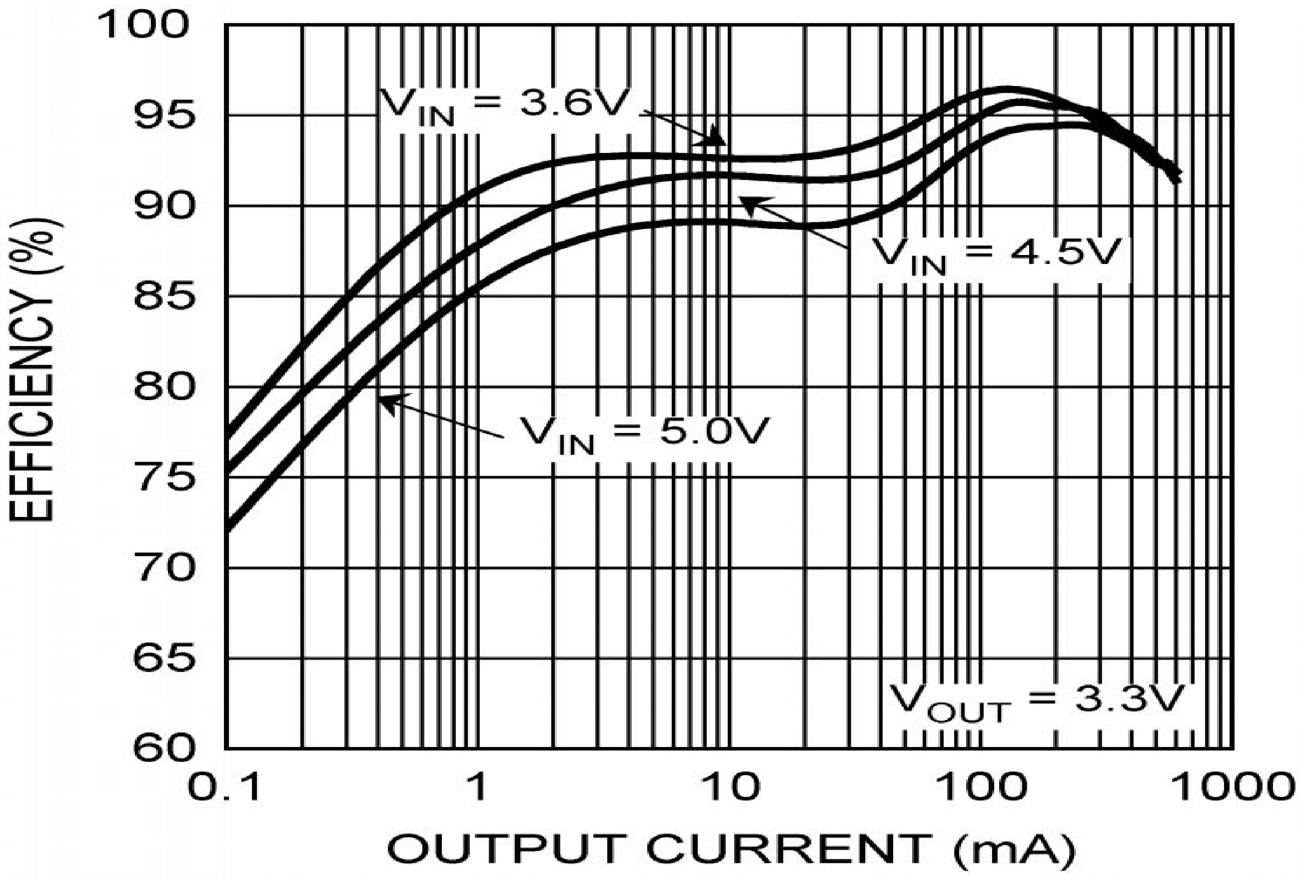
Buck Converter

For the purpose of efficiency we replaced it with a chip:

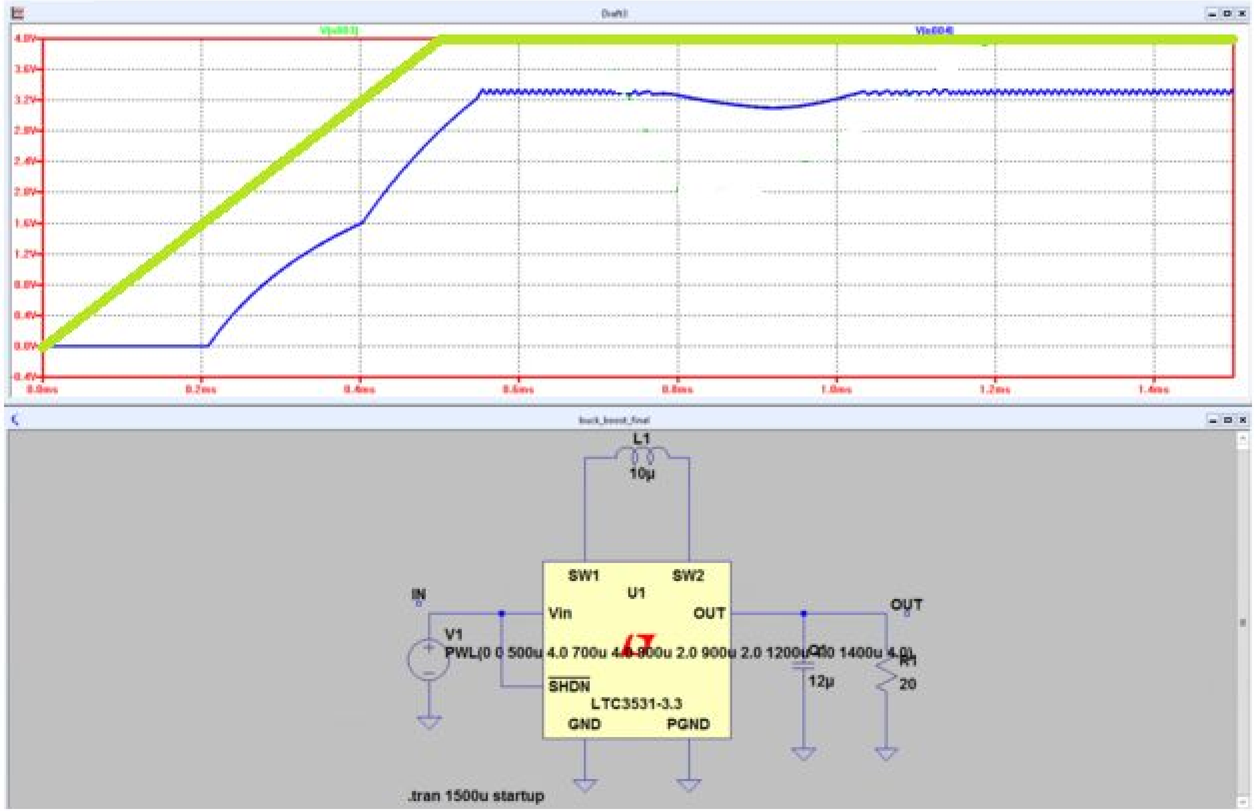
- Part no. LM3671
- Input Voltage: 4.1 ± 1.4 V
- Adjustable output voltage: 2.2 ± 1.1 V



Buck Converter - Graph / Plot



Buck Converter - LTspice simulation

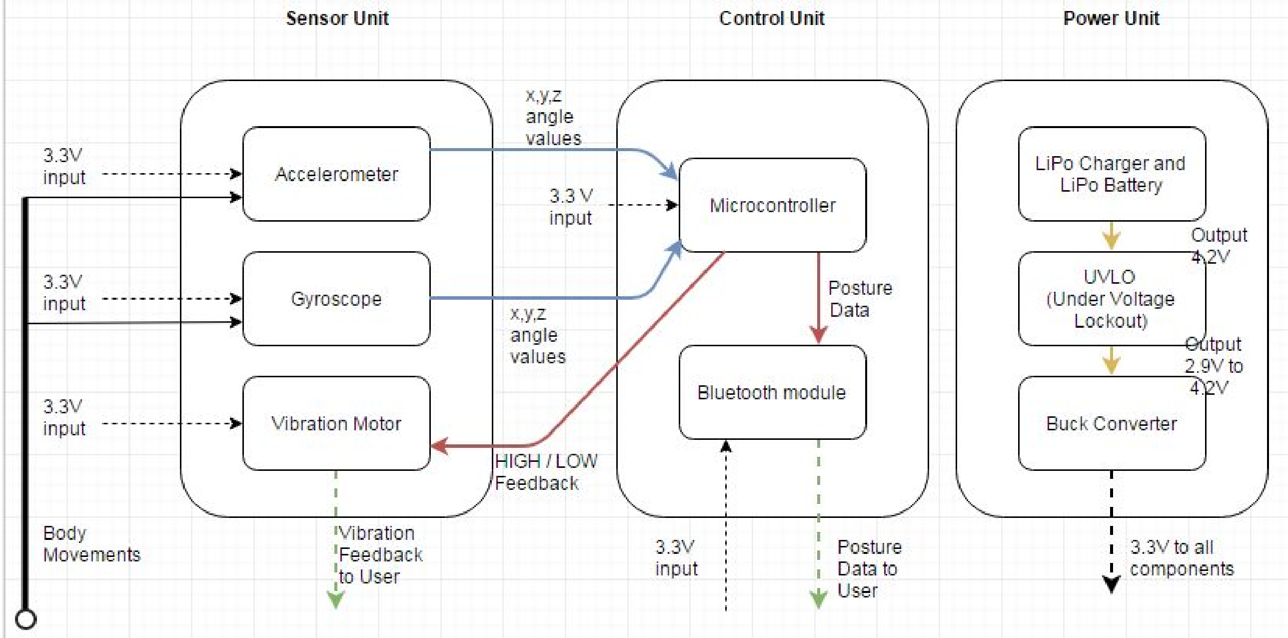


Blue line	Battery input voltage
Green line	Output voltage from buck converter

Buck Converter - Requirements and Verifications

1. For input voltage between 3.75 ± 0.45 V; the buck boost should output a voltage of 3.3 ± 0.05 V.	• Use a function generator to supply 3.75 ± 0.45 V to the Buck. Use an oscilloscope (one terminal of oscilloscope on positive output terminal of Buck-Boost and other terminal of oscilloscope on negative output terminal of Buck-Boost) to measure and graph voltage changes across the Buck-Boost. • The oscilloscope should show 3.3 ± 0.05 V output on the oscilloscope.

Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Microcontroller

ATmega328P - Atmel

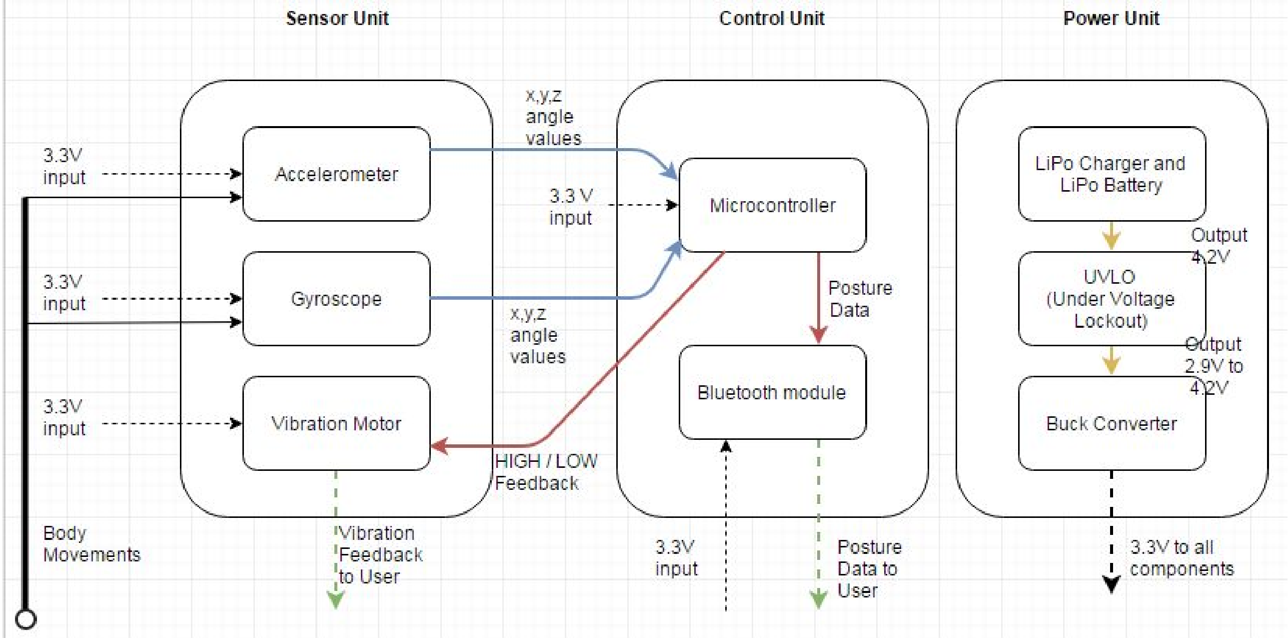
8-bit RISC-based microcontroller

32KB ISP flash memory with read-write capabilities

23 general purpose I/O lines

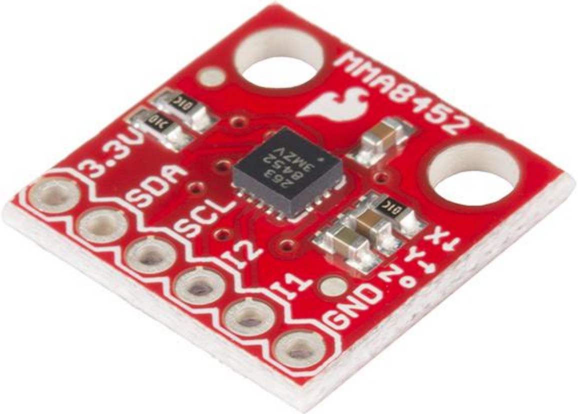
Operates between 1.8 V and 5.5 V

Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Accelerometer



- Low powered, 3-axis
- 1.95 V to 3.6 V supply voltage
- Output Data Range from 1.56 Hz to 800 Hz
- Current consumption 6uA to 165 uA

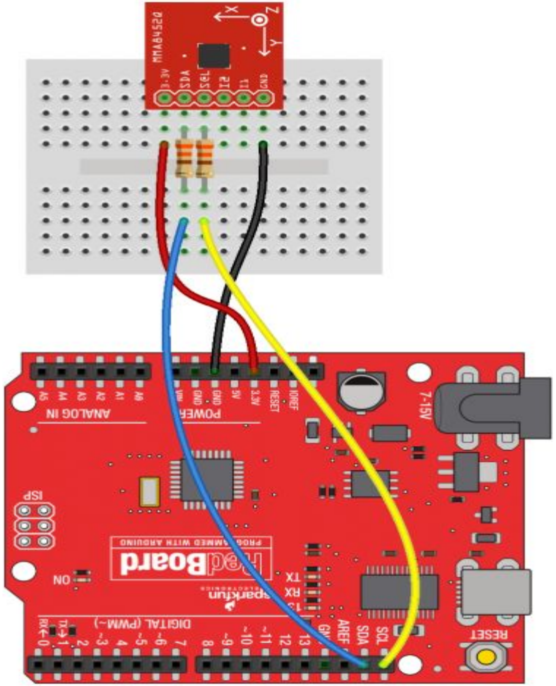
MMA8452Q

This chip uses I2C (Inter-integrated circuit) protocol to allow multiple “slave” digital integrated circuits (“chips”) to communicate with one master, and it requires two signal wires to exchange information

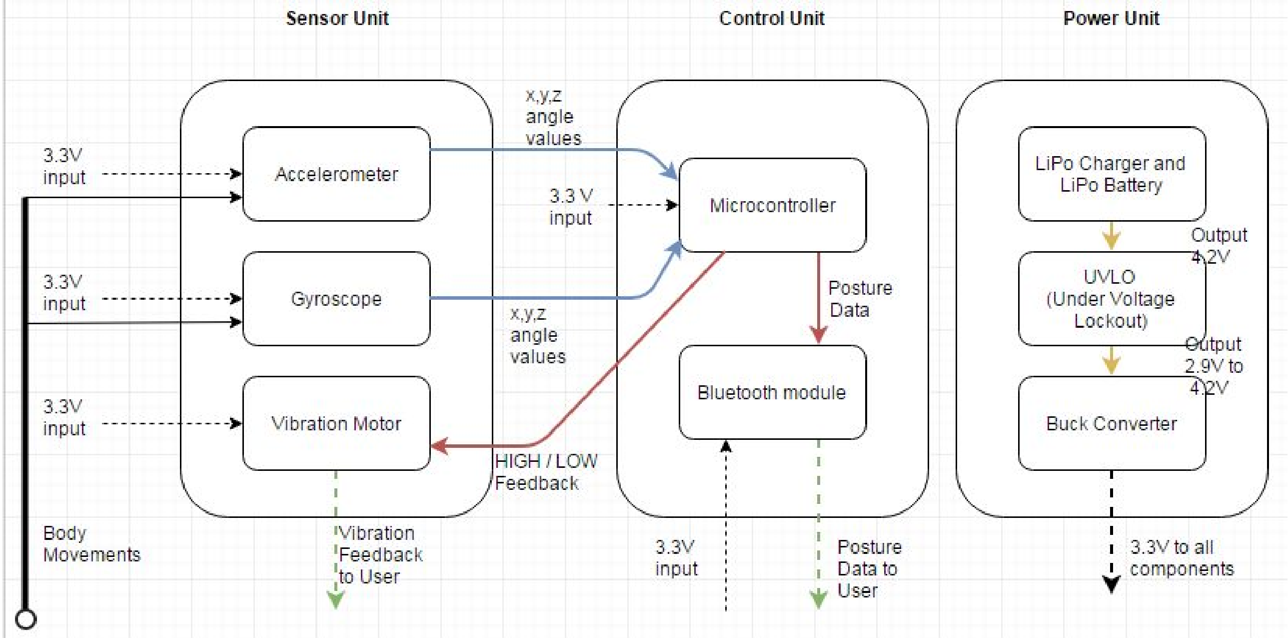
Accelerometer - Hookup

Since our SCL SDL lines do not work at the exact same voltage as the accelerometer, we need some kind of level shifting to protect the SCL and SDL lines.

We use a 330 ohm resistor in series on each I2C line for this.

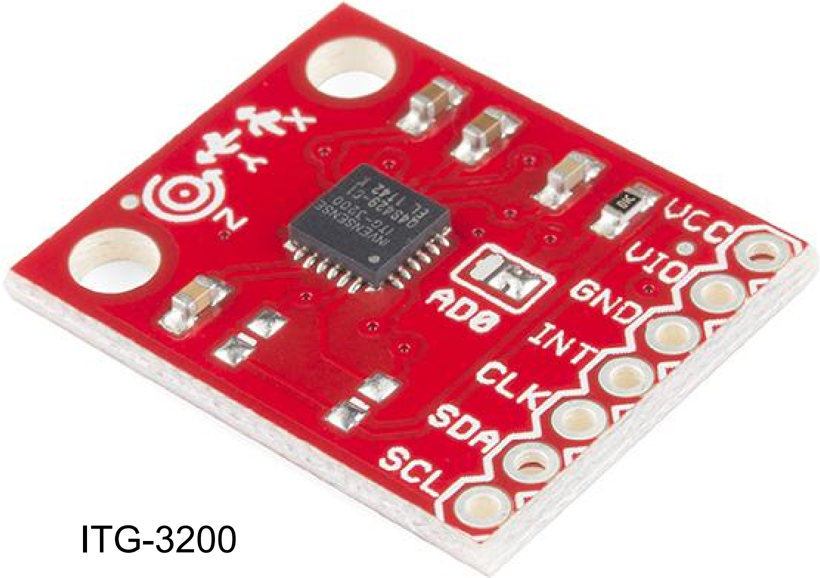


Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Gyroscope



ITG-3200

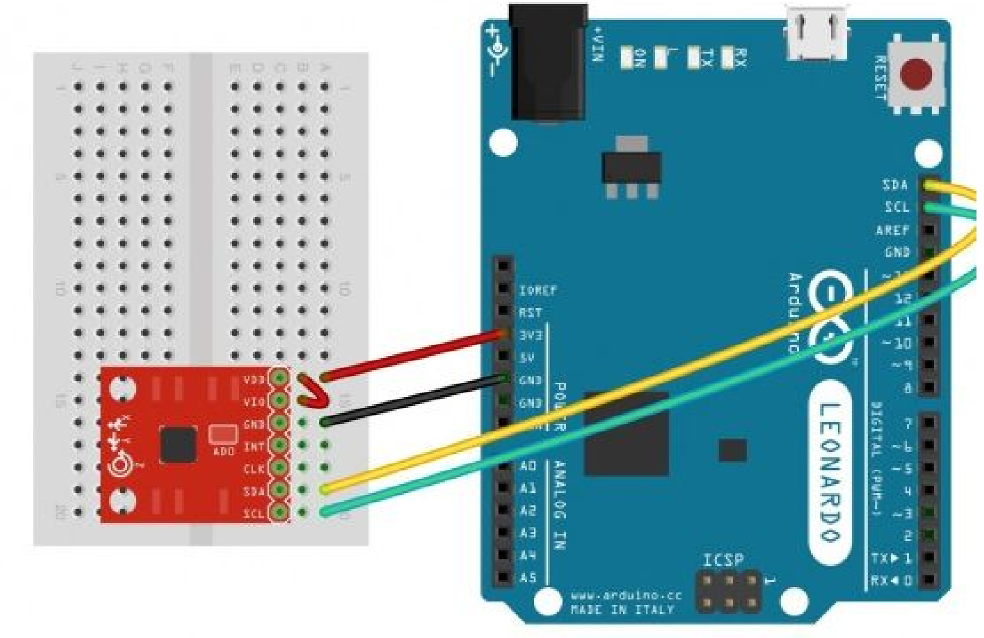
- 3-axis gyroscope
- 2.1 V to 3.6 V supply voltage
- Current consumption 6.5mA

This chip uses I2C (Inter-integrated circuit) protocol to allow multiple “slave” digital integrated circuits (“chips”) to communicate with one master, and it requires two signal wires to exchange information

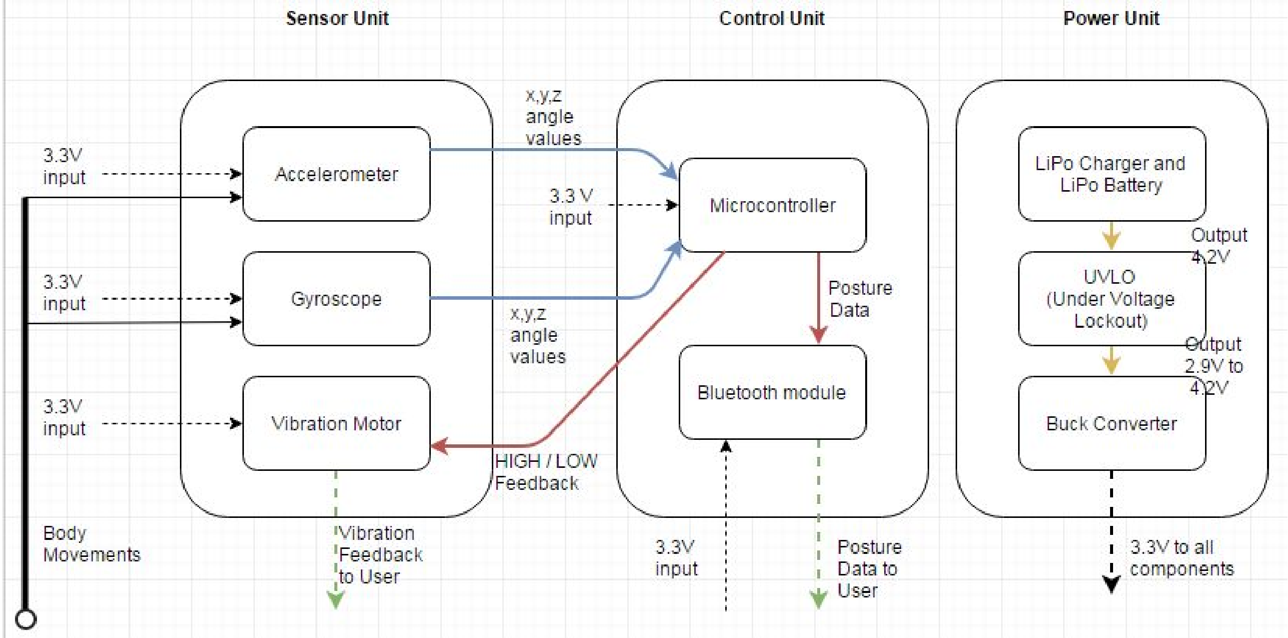
Gyroscope

If there is nothing connected to SCL SDL lines, and the microcontroller reads the state of the pins, it will be difficult to tell if the pin is high (pulled to VCC) or low (pulled to ground)

We use a 10k ohm resistor on each I2C line for this so that the input pin will read a high.



Block Diagram general layout of circuit



Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

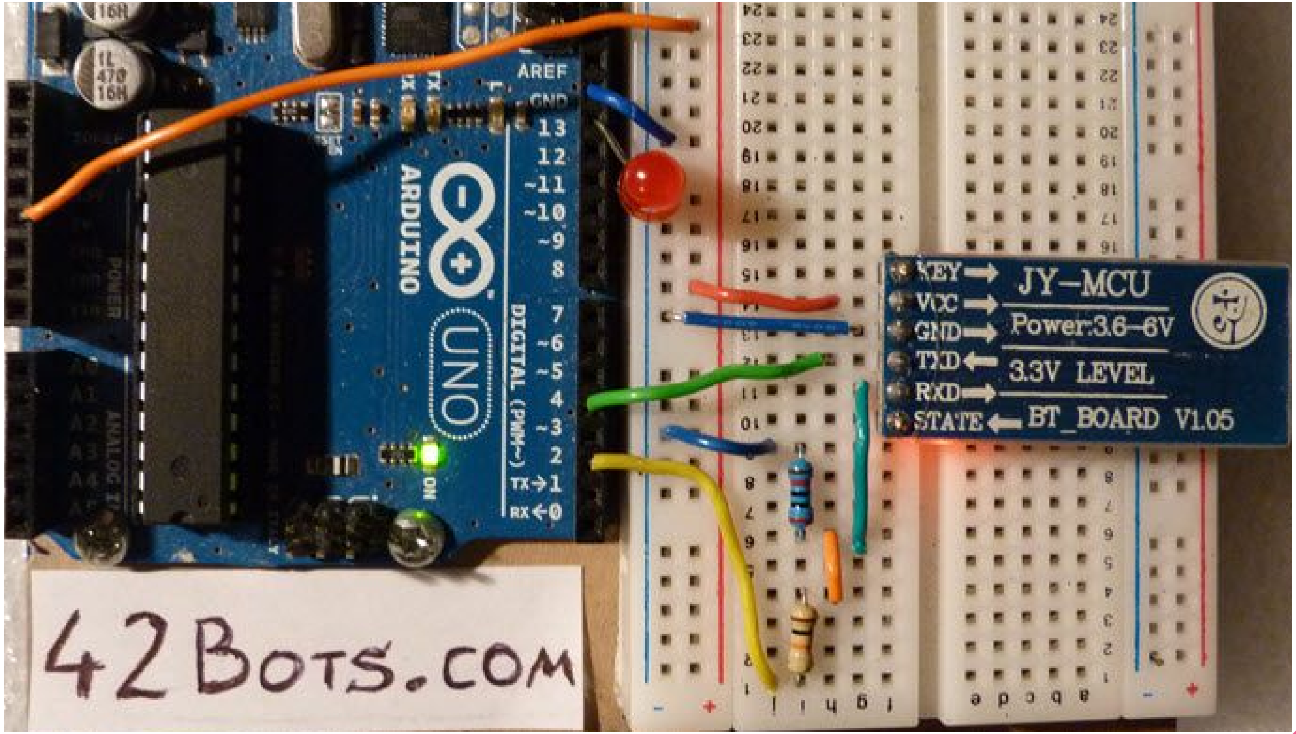
Bluetooth

For the purpose of efficiency we replaced it with a chip:

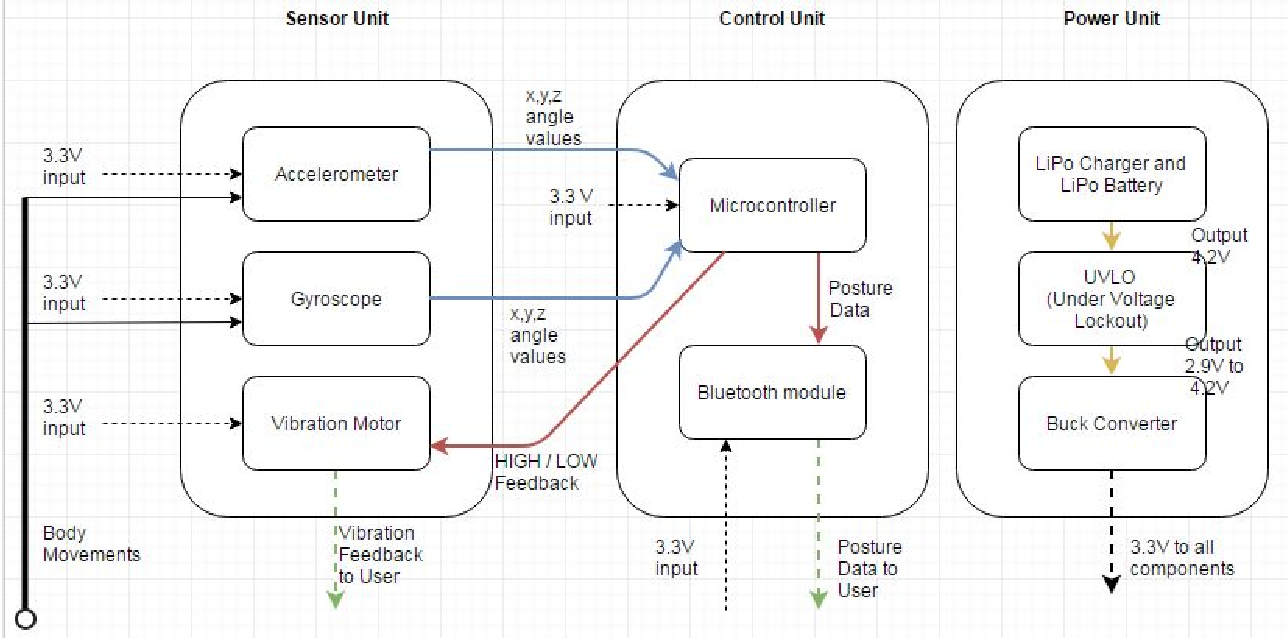
- JY-MCU
- Input Voltage: 3.6 - 6.0 V
- TXD (input data line) takes a 3.3V input from the micro controller
- RXD (output data line) gives out a 3.3V output to the microcontroller

Bluetooth - breakout board

The breakout board setup with a voltage divider using 1k ohm (blue) and 2k ohm (white) resistors.



Block Diagram general layout of circuit

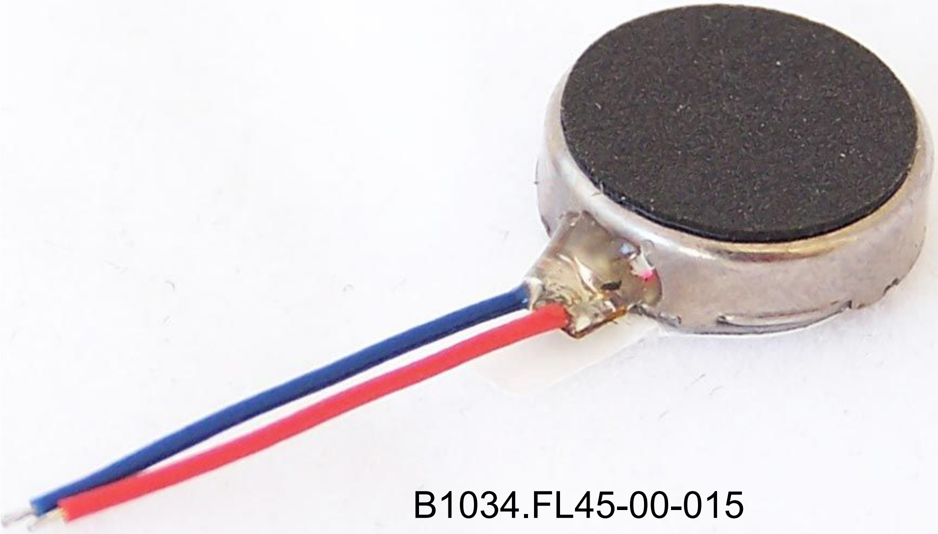


Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Vibration Motor

Specs for the motor from the datasheet:

- Input Voltage range: 2.0 - 3.6 V
- Rated current: 60mA max
- Non audible indicator
- Attach blue end to ground and red end to 60 ohm resistor and then to digital pin 10 (pin 16) of microcontroller.



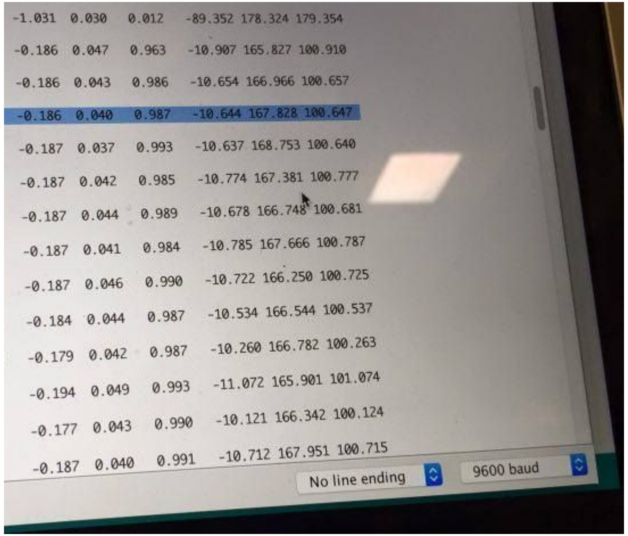
Vibration Motor - Calculations

Calculations for the 60 ohm resistor (R) needed by the motor:

- Input voltage (V) = 3.3 Volts
- Max current to flow (I) = 60mA
- V = I * R
- 3.3 = (60/1000) * R
- R = (3.3 * 1000) / 60
- R = 55 ohm
- Since 60mA is the maximum current, we used a higher resistor of 60 ohms.
- This will give us a current of 55mA.

Bluetooth and vibration motor

Microcontroller reads accelerometer/gyroscope values to calculate angles which are constantly sent to the bluetooth module which can be seen on a computer



If the angle value crosses a certain threshold, the vibration motor gives feedback.

Successes, Challenges, failed verifications

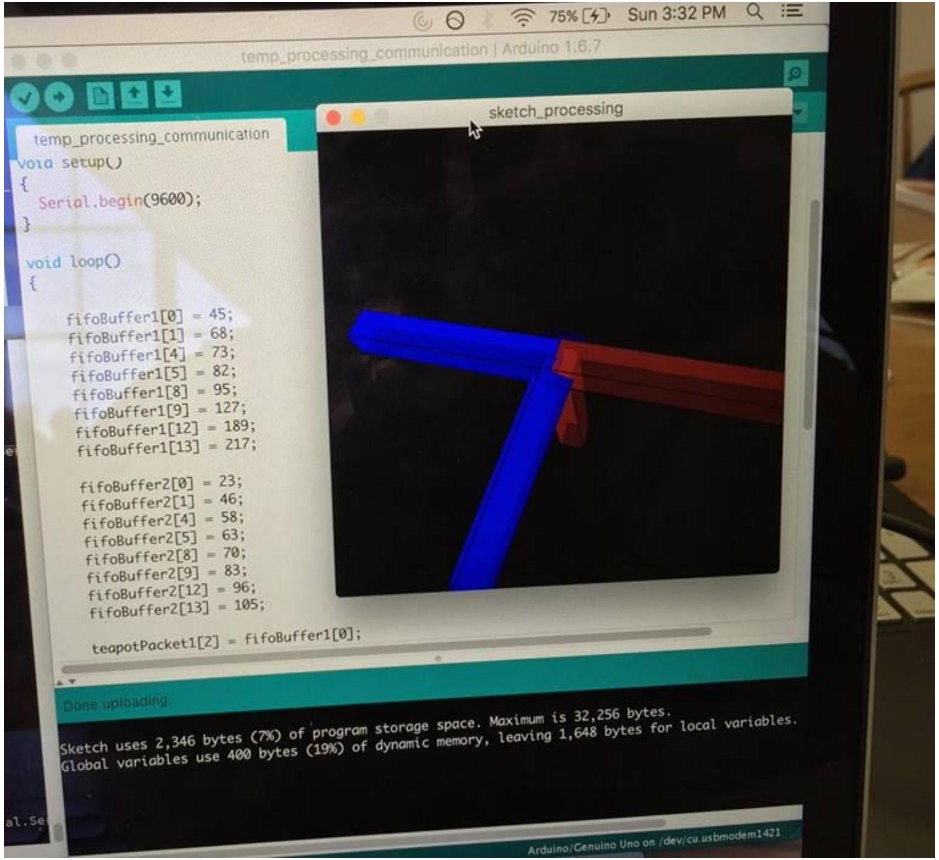
- Good accelerometer data, very responsive motor when threshold angles crossed
- Good design was a challenge, we want a wearable to be as small as possible
- Bluetooth module had low power and was not transmitting data properly; we needed a 4V supply rather than a 3.3V
- Frontend/interaction with the device needs to be perfect. We cannot have a user calibrating the device every time he wears it.

Conclusion

With a lot of people using this, we aim to collect data and use it for medical purposes.

We are also looking at orthopedics recommending our programmable posture corrector to patients to monitor their posture.

Further Work



- Make very good front end. *<u>Interaction with wearable</u>* is a very important part
- Make physical device the *<u>size</u>* of a nail to magnetically stick in shirt or size of a credit card to put in a wallet
- Work on *<u>data storage</u>* and reduce power usage by bluetooth for better *<u>battery life</u>*

Thank you..!

Questions ??