Device Factory Accelerometer - DF1

Device Factory introduces the DF1, a 3-axis accelerometer open platform. Over a year in the making, DF1 is our first full scale production device assembled in the USA and fully FCC certified. The device communicates to a base station via Bluetooth Low Energy (4.0), and is the only means of interacting with the device. The base station can be a SmartPhone, Tablet, or PC that is Bluetooth 4.0 capable.

The primary features of DF1 are long range antenna, configurable accelerometer to ±2/4/8G range, optimized power consumption, replaceable battery, all in a small form factor.

The function of DF1 is left to the user. Although we provide demo software on our GitHub, you can transform the device into any use case that you can dream up. To give you a few ideas:

* Bike alarm for your fixie
* Door bell
* Impact sports monitor
* Research sensor and data collection

While the elapsed time was over a year, we mostly worked nights and weekends. Often interrupted by beers and off-topic discussions. Device factory started as just a hobby. To give ourselves a goal we had the idea to produce a Bluetooth baby monitor using a remote accelerometer and an iPhone. The idea seemed good enough to create a company… but then reality set in. Making a consumer device for non-technical people requires a lot of refinement and effort in things we didn’t really care about. Ditching the day job to take on the task was maybe not the best path either.

What we really wanted was to work on technical devices with other people that like technology. Doing things on your own is hard… we know. Together we could do more, and we thought it would be fun to be able to provide devices to other people that wanted to learn and write code for them. So the prototype device transformed from a consumer product into a technology platform that can be explored and hacked by the user.

It was not easy. Dealing with testing labs for FCC certification, creating enclosure designs and injection molds, custom battery contacts, ordering up components and working out how to assemble the PCB with the fab house. Not to mention writing all the firmware and software, and creating place for people to play with the code on the internet.

I’m not sure if there are any rational economics in this project. However, what it has given us is the opportunity to meet and interact with interesting people. We hope that you find yourself interested in our first device and will enjoy coming up with new ideas for the platform.

**Specifications:**

* G-Force: 3 axis ±2/4/8G
* Max Range: 250ft
* Battery: CR2032 Lithium-Ion 3.0V
* Battery Life: 3 years standby, 3 months active
* Weight: 22gm
* Diameter: 1.4in (35.56mm)
* Height: 0.38in (9.652mm)

**Components:**

* Main IC: Ti CC2541 System on a Chip
* Sensor: Freescale MMA8451Q Accelerometer
* Primary Clock: XTAL1 32MHz
* Secondary Clock: XTAL2 32.678kHz
* Bandpass Filter: Murata Chip Balun
* Chip Antenna: ANT-2.45-CHP-x

**Base Stations:**

The base station is any computer with a Bluetooth Low Energy (4.0) capability.  Software must be installed to communicate with the DF1 device.  This can be either the demo applications provided on GitHub, or they can be coded by the user.

**⇒**

* iOS devices: iPhone, iPod, iPad
* Android devices: Phones, Tablets
* RaspberryPi with a Bluetooth 4.0 dongle
* Any PC with with Bluetooth 4.0 support

**Hardware:**

DF1 uses a **Ti CC2541** power optimized true system-on-chip (SoC) solution for Bluetooth Low Energy applications.  The CC2541 combines the excellent performance of a leading RF transceiver with an industry-standard enhanced 8051 MCU, in-system programmable flash memory, 8-KB RAM, and many other powerful supporting features and peripherals. The CC2541 requires 2 clock sources for internal timing of application and radio: XTAL1 32MHz provides the main clock source, XTAL2 32.678kHz acts as the auxiliary clock source.  All components are surface mounted to the PCB.  The PCB is double sided, and both sides contain the ground plane. The ground plane not only serves as an electrical ground against the battery, but also as a counterpoise for the monopole chip antenna.

* 2.4-GHz *Bluetooth* Low Energy Compliant RF System-on-Chip
* Supports 250Kbps, 500Kbps, 1Mbps, 2Mbps Data Rates
* High-Performance and Low-Power 8051 Microcontroller Core With Code Prefetch
* In-System-Programmable 256KB Flash
* 8-KB RAM With Retention in All Power Modes

**Antenna:**

The DF1 device includes a monopole chip antenna inside the enclosure.  Radio modulation is done within the Main IC which is then sent though a Murata chip Balun bandpass filter, and then onto the chip antenna.  This antenna is designed to work with Bluetooth devices operating in the 2.4GHz frequency band for radio communication.

**⇒**

* Frequency: 2.45GHz
* Bandwidth: 180MHz
* Wavelength: 1/4-wave
* Impedance: 50Ω
* VSWR at Center: ≤2.0

**Enclosure:**

Injection molded ABS plastic with light pipe for LED emission. A relief cut around the circumference allows for an attachment clip interface.  The flat surface on back of enclosure allows for mounting via 0.75 inch 3M Dual Lock disc.  Enclosure is not waterproof.  Our GitHub features 3D printable accessories for DF1 that you are free to download, modify, and print.

**Operation:**

The DF1 device relies entirely on remote control via Bluetooth Low Energy protocols. The user interacts with the device using Bluetooth capable devices such as SmartPhones, Tablets, or PC’s.  There is no physical on/off switch on the device.  No external power source is needed.  The device is entirely battery operated and is meant to be portable.

Firmware is also upgradable over the air.  Fixes and enhancements for the main firmware will be rolled out in the future.  Alternate firmware will also be available which will unlock more use cases for DF1.  For example,  it will be possible to use DF1 as an iBeacon.

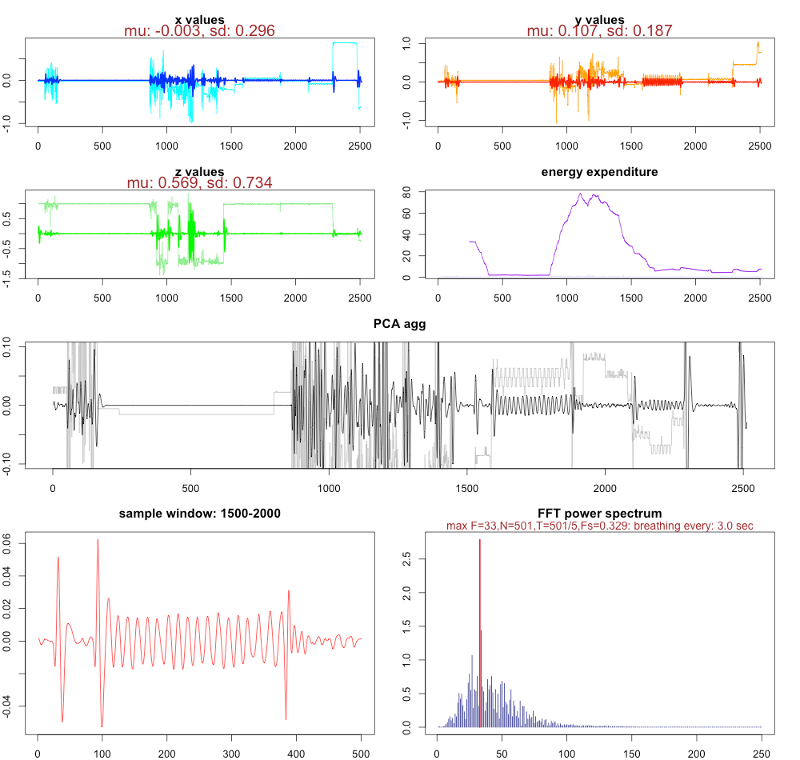
Algorithmic Detection of Breathing Cycle

Our body moves in tandem with breathing. Especially in the upper torso area, the rise and fall of the chest is prominently associated with breathing. When placed near key spots, breathing cycle can be inferred from movement data alone. The xyz values from the accelerometer can be used to derive cyclical human breathing pattern.

The key elements of the algorithm is to filter out as much noise from the data as possible, and apply the appropriate algorithm on the transformed data. In practical implementation, the algorithm also has to run in realtime, and refine its estimation from streaming data. A person normally takes about 2-5 seconds (0.5 – 0.2Hz) to complete a full inhale and exhale cycle. Assuming a single breathing cycle completes within 2-5 seconds, you only need about 1-2 data points per second to detect a frequency in theory. With 5Hz sampling frequency, it becomes feasible to tackle breathing detection using captured xyz acceleration values.

The following plots illustrate this. From left to right, going down each row, you will see:

* x acceleration values and its corresponding high and low pass filtered timeseries
* y acceleration values and its corresponding high and low pass filtered timeseries
* z acceleration values and its corresponding high and low pass filtered timeseries
* plot showing the “activity level”, expressed by the euclidean norm of change in xyz values
* all 3 axis aggregated by their proportional weight in the variances, where their collinearity is taken into account by eigenvalue-based weights
* we zoom into a particular sample patch in the aggregated time series, where clean sinusoidal signal can be extracted
* running discrete fourier transformation on the zoom-ed in sample, and the red bar highlights the prominent frequency



The highlighted frequency coincides with the rate at which the subject was breathing. Our device was placed at various places on the body, while the sample was being collected in realtime. The strongest signal was obtained when our device was placed near the upper torso. The data transmission was done wirelessly at 5 times a second to an iPhone. The iPhone recorded all data received and persisted it into a csv file, which was used for this analysis.

The implementation of the algorithm can be done without much need for complex computation. Also, less than 2 minutes worth of 5 Hz samples (2\*60\*5 = 600 data points), need to be kept in memory to get sufficiently reliable readings. The data should be maintained in a queue, and the most recent data points should replace the older data points. We will be working on sample application to illustrate this algorithm in action.