Teardown: Misfit Shine Activity Tracker a learn.sparkfun.com tutorial

Available online at: http://sfe.io/t213

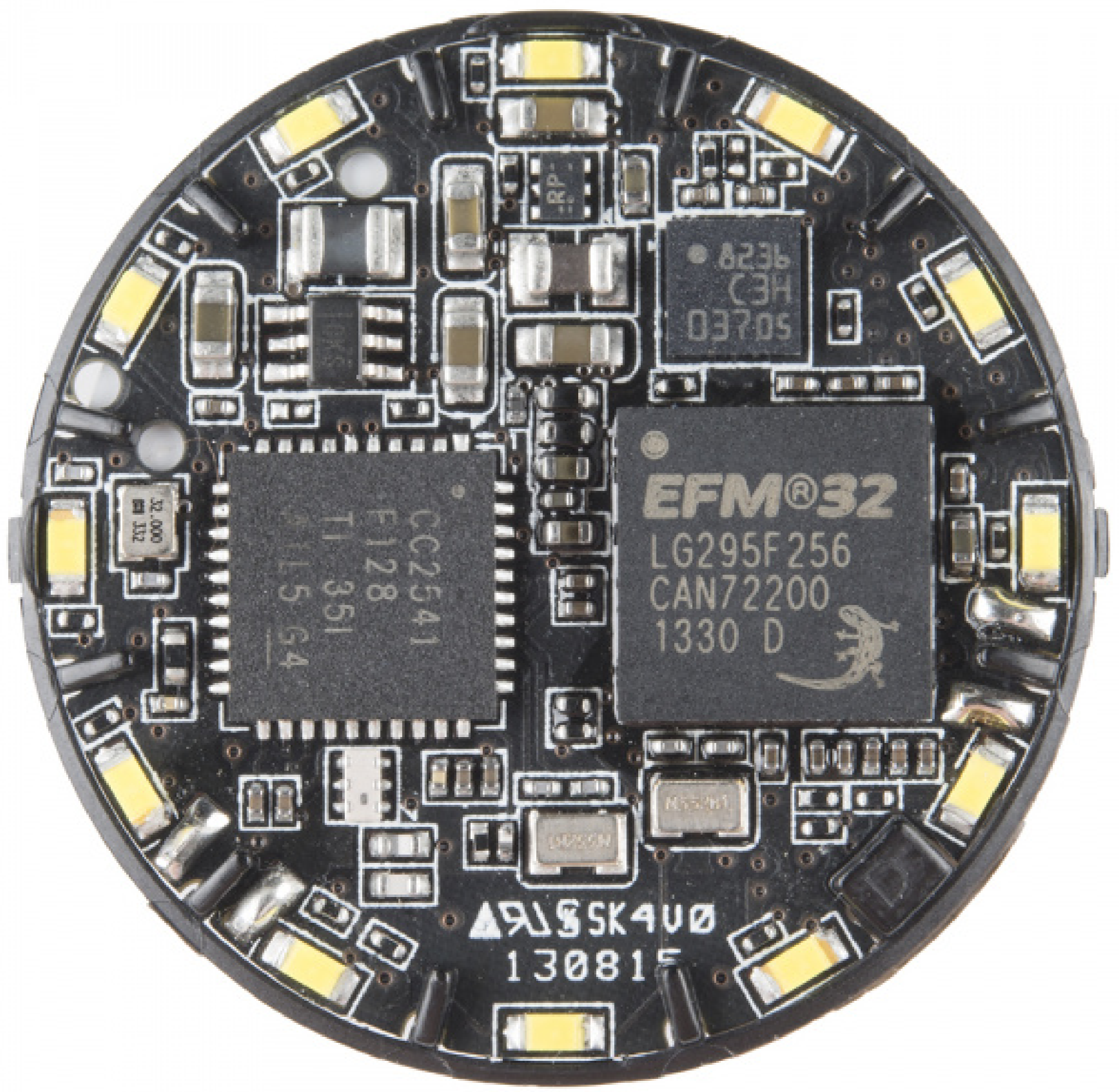
Contents

Shining

More than just fancy pedometers, a whole new class of mobile devices are flooding the market, designed to make us more aware of our own physical activity. And in this emerging market jam-packed with competitors like Nike, Garmin, and Fitbit, the Misfit Shine is easily the... prettiest. Hardly bigger than a US Quarter and barely thicker than a pencil, it can be worn anywhere on the body and looks more like a piece of jewelry than a high tech activity tracker. But can you really fit a fully functional fitness tracker in something that size? There's only one way to find out...

Don't Open 'Til Doomsday

Getting into the Shine turns out not to be that hard. After popping open the case like you're changing the battery, anything sharp will pop the retaining ring off of the pcb and give you access to this:



Wow! That's a lot of stuff in such a small package!

There are really three parts to the Shine, everything else on the board is essentially just there to support these three. Let's examine each part and how it helps the Shine track your fitness:

Brains of the Operation

The EFM32 Leopard Gecko microcontroller from Silicon Labs is the controller that keeps this whole operation in one piece. The Leopard Gecko is a 32-bit microcontroller based on an ARM Cortex-M3 core. It comes packed with energy saving features that make it ideal for this kind of application.

Of particular note is the Leopard Gecko's low-energy sensor interface which plays a big role in the Shine's advertised 4-month battery life. The low-energy sensor interface allows the EFM32's peripherals to communicate independently of the core (gathering accelerometer data, for instance) and allowing the core to stay in energy saving mode.

The Leopard Gecko's 256k of program storage and 48MHz processing speed are also well suited to this application since the Shine not only has to store and process sensor data using complex algorithms, but also has to communicate that processed data using the (somewhat sizable) Bluetooth low energy software stack.

A Sense of Purpose

Of course just wearing a microprocessor on your sleeve won't tell you a lot about your fitness. To get a handle on what you're up to, the Shine relies on the LIS3DH Accelerometer from STMicro, another ultra low-power part. This tiny LGA-16 package saves real estate on the PCB but still serves up acceleration data at speeds up to 5kHz.

Communication is Key

Collecting and processing sensor data is a neat trick, but, if you can't ever get to that data, what good is it? Transmitting the data wirelessly is the perfect way to get it out of the device, but it takes a lot of power to send things over air. To solve this problem, Misfit is taking advantage of the brand-newish Bluetooth Low Enegry (BLE) protocol. The device behind that ability is the CC2541 Bluetooth SoC by Texas Instruments. With its programmable output power and low energy operation, the CC2541 does its part to keep the battery healthy too.

All Together Now

If I had to guess what's going on in there, which I do, I'd say that the Leopard Gecko is spending 98% of its time in power saving mode, only waking up on boot, when syncronizing, and after detecting a double or triple tap event. The accelerometer data probably gets processed either at synchronize time or when some buffer gets filled with raw data... maybe both? Either way, I'm almost sure that your phone never receives raw accelerometer data.

People who have reviewed the Shine online say that it doesn't usually live up to its 4 month battery life promise, but it still lasts impressively long on a standard coin cell battery. Using low energy parts and keeping everything in power-save mode as long as possible has a lot to do with that.

The Unexplained

I tried to find some way of hacking the firmware on this, but I didn't have any luck identifying a point of entry. I probed every test pin and package lead that I could reach, but I couldn't find anything interesting on the logic analyzer. The controller, being a BGA package, wasn't very accessible from a hardware hacking standpoint. Ultimately I gave up hope on trying to figure things out on the device side.

Because the Shine is a Bluetooth Low Energy device, however, I figured it might be possible to get some information on the host side. I downloaded a utility for my smartphone that reads BLE device attributes and connected to the Shine. Unfortunately, there didn't seem to be any way for me to access the sensor data form that end either. For anyone interested, though, here's what I was *was* able to find out using the BLE utility...

After connecting to the device, you have access to 4 services, which are basically categories of attributes that can be read, written, or otherwise manipulated. Generic Access, Generic Attribute, and Device Information are all standard services on the BLE protocol and will give you things like this:

Device Name (UUID 0x2A00): Shine

Appearance (UUID 0x2A01): [1088] Generic: Running Walking Sensor

Firmware Revision String (UUID 0x2A26): 0.0.50r

Not very interesting. You can also get your device serial number and stuff like that. There's another service on the device, and it's a proprietary service with its own UUID that I won't bother typing here. It has writable characteristics, but, without knowing what I'm doing, I'm reluctant to try writing to them. They may not even be related to the primary function of the device.

I guess some things will remain a black box. I threw the device back together, and I'm going wear it for a while to see how it holds up! After all, even if you can't hack it, Misfit has hinted at the release of an API in the future, and we can all look forward to that.

Resources and Going Further

If you enjoyed this teardown (or even if you didn't, heck) then check out these other SparkFun teardowns:

- MetaWatch Teardown
- Nest Thermostat Teardown
- Nest Protect Teardown

And for more information on Bluetooth and accelerometers, check out these tutorials:

- Bluetooth Basics
- Understanding the BC127 Bluetooth Module
- Accelerometer Basics
- LSM9DS0 Hookup Guide

Happy Hacking!