Open Source, Open Hardware Fitness Watch

EECS 473: Advanced Embedded Systems

Nathan Immerman, Steven Sloboda, Amit Shah
Joshua Kaufman, Tyler Kohan

Overview

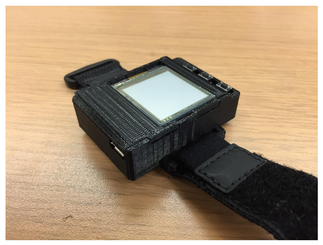
Currently the realm of wearable technology is dominated by closed source, closed hardware, proprietary products. Some include documented APIs so that developers can create apps that interact with the closed systems, but people are prohibited from modifying and redistributing the systems themselves. We believe this limits user freedoms and discourages community development/hacking. Thus, we built an open source, open hardware fitness watch that is meant to help open up the realm of wearable technology to developers, hobbyists, and hackers.

The watch itself is a standard fitness watch—it includes an LCD, a GPS module, and an IMU that we use as a pedometer. The sensors record fitness data such as the number of steps taken and the speed at which the wearer is moving. The fitness data can be wirelessly transmitted via Bluetooth Low Energy (BLE) to a smartphone for further processing. Additionally, we built a sample “dashboard” application to help demonstrate the watch’s functionality and the potential for other applications to be written that use the data provided by the watch.

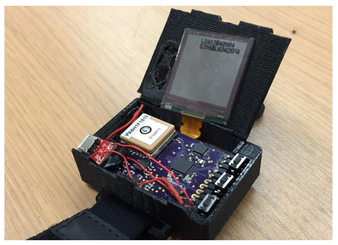
We have published all of our code and circuit board schematics on Github where they are freely available[1]. We have licensed as much of our work as possible under the GNU Public License (GPL) version 2 to ensure that it will always be free and open source.

Watch Case

The watch case was designed using Solidworks and 3D printed using a Makerbot Replicator 2. The top panel of the case can be opened to reveal the printed circuit board inside. A wrist band can be attached to the watch using the loops on both sides of the case.

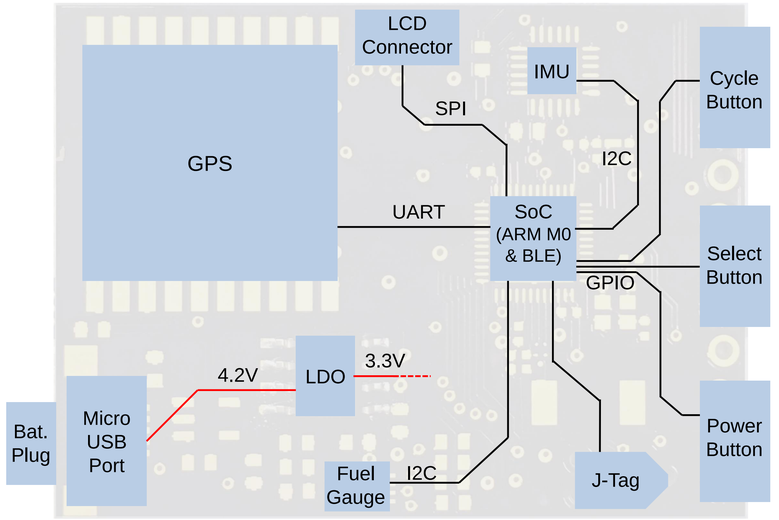


Watch case encloses PCB



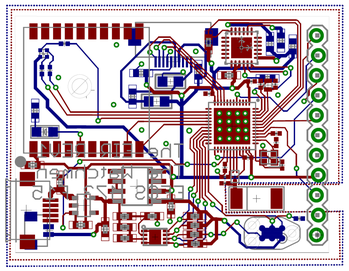
Removing the top panel exposes the PCB

System Block Diagram

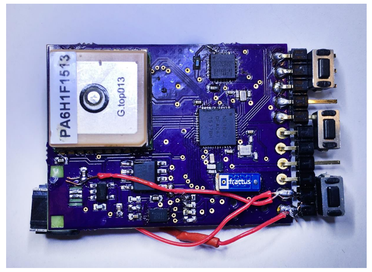


Printed Circuit Board

We created a printed circuit board (PCB) that is 30cm x 40cm to which to affix all of the watch components. We designed a 2 layer board that has incorporated ground planes on both the top and bottom. The majority of the parts, including the GPS, IMU, and SoC, are located on the top side of the board. The bottom side of the board only contains the LCD connector and miscellaneous small parts. Right angle header pins are used as breakouts to attach the mechanical buttons, as well as to allow easy attachment of additional peripherals, such as a heart rate monitor. Parts of the PCB were based upon Lab11’s Nucleum project[2].



EAGLE board layout



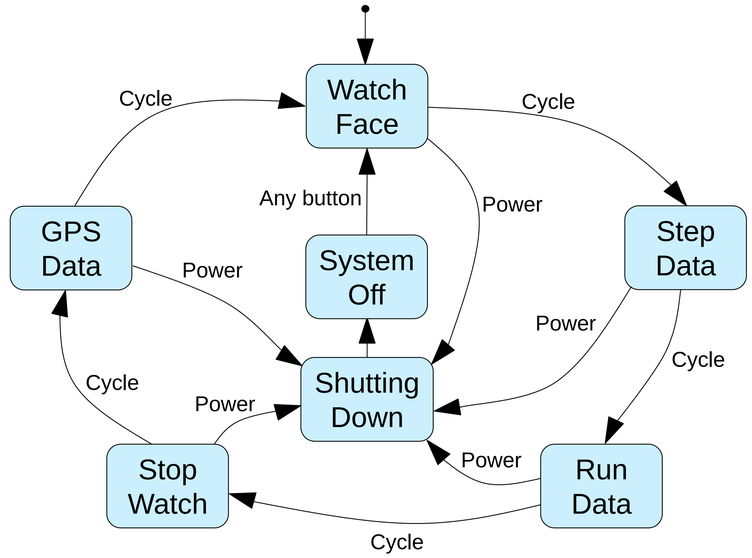
Populated PCB

Embedded Software/Firmware

We developed the software for the watch SoC using our fork of Lab11’s nrf51-pure-gcc-setup[3] repository, which allows for cross compiling of code for the ARM M0 on the nrf51822 SoC using the GNU C Compiler and the GNU ARM embedded toolchain. Also, we used our fork of Lab11’s nrf5x-base[4] repository that contains the Nordic SDK to reduce the time required for software development. Additionally, we wrote drivers for the individual components on the watch including the MPU, fuel gauge Coulomb counter, GPS module, and LCD.

Program State Machine

We created a state machine in software to control the program flow. The states of the state machine roughly correspond to the different screens that are displayed on the watch LCD. The cycle button transitions between the states which update the information displayed on the LCD. The select button can then be used to access additional functionality specific to that screen. Lastly, the power button can be used to enter a low power state in order to preserve battery power and prolong battery life.



Dashboard Mobile Application

We developed a demo dashboard Android app that requests and displays data from the watch. This gives developers the ability to access and store all of the watch’s data on a smart phone. This app shows the potential for developers to build fitness apps using our platform.

ANDROID

ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

1. https://github.com/sloboste/eecs473-fitness-watch
2. https://github.com/lab11/nucleum
3. https://github.com/lab11/nrf51-pure-gcc-setup
4. https://github.com/lab11/nrf5x-base

Special thanks to Lab11 for the development boards, components, and assistance.

Thank you to our sponsor
Ford Motor Company