stand

Contributors:
Lisa Yip

Advisor:
Dr. Hugh Smith
Computer Engineering Department

California Polytechnic State University, San Luis Obispo
June 2016

State of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Table of Contents

List of Tables

List of Figures

List of Nomenclature

Accelerometer – an instrument for measuring acceleration, or gravitational force. It keeps track of the movement and orientation of an electronic device.

Application Programming Interface (API) – a set of routines, protocols and tools for building software and applications.

Backend as a Service (BaaS) – a model for providing web and mobile app developers a way to link applications to back end cloud storage and APIs while providing user management, push notifications, and integration with social networking services. These services are usually provided via custom SDK and APIs.

Certificate Signing Request (CSR) – it is a block of encrypted text that can be generated on the server that the certificate will be used on. It contains information such as organization name, common name, and public key. A certificate authority will then use the certificate signing request to create a SSL certificate

Graphical User Interface (GUI) – This digital display program runs on the user's personal computer, and allows the user to input the necessary information into a digital format that can program the Forest Sign Maker accordingly.

JavaScript Object Notation (JSON) – a JSON is a open-standard format that uses human-readable text to transmit data objects consisting of attribute – value pairs. It is a language-independent data format derived from JavaScript.

Integrated Development Environment (IDE) – a software application that provides comprehensive services to computer programmers for software development. Usually consists of a source code editor, automation build tools and a debugger.

Inter-Integrated Circuit (I2C) – a multi-master, multi-slave, single ended, serial computer bus typically used for attached peripherals to processors and controllers.

Pedometer – an electronic or electromechanical portable device, that counts each step a person takes by detecting the motion of a step.

Software Development Kits (SDKs) – a set of software development tools that allows the creation for applications for certain software, framework, or platform.

Serial Peripheral Interface (SPI) – a synchronous serial interface specifically used for short distance communication, usually found in embedded systems.

Secure Sockets Layer (SSL) – a standard security technology for establishing an encrypted link between a web server and browser. It ensures all data passed between the web server an browsers remain private and integral.

Abstract

With busy lives, people sometimes forget to take breaks for multiple hours at a time. The purpose of this project is to remind the user to move around every hour if they have not already. To do so, `stand` will be a light weight fitness tracker that will detect the number of steps made within an hour. If the step count, dependent on goals, is not met then a vibration motor will gently remind them to take a break and walk. The data throughout the day will be shown on an iOS application where the user will be able to interact and modify their goals.

Introduction

Problem Definition

The problem that Stand have set out to resolve is the issue of making sure that the user is walking every hour. Currently, other than Jawbone, none of the low cost fitness devices have a feature that motivates the user hourly if they forget to walk.

Objective

This senior project will attempt to resolve the problem of making sure a user walks hourly by designing a wearable fitness device, `stand`, with the aid of Dr. Hugh Smith for advice and motivation.

Project Motivation

At the moment, most fitness devices in the market has features such as activity tracking, sleep tracking, heart rate monitors and much more. However, none of them offer a feature that makes sure the user walks a recommended amount of minutes per hour [26]. Creating a low-budget wearable prototype device that reminds the user to get up and walk throughout the day can lead to an increase in health benefits.

Scope

The goal of Stand is to keep the user in check hourly in order to meet the recommended daily steps by the National Institute of Health (NIH) [26]. In order to so, the requirement is to build a wearable device that can fit on a human wrist and it will interact with an Apple iOS application to display useful information to the user. This project will help eliminate the issue of forgetting to walk or remembering to take a break hourly.

The scope of this project encompasses the majority of the design process that extends from the initial problem definition to developing a functional prototype of the final design. All immediate steps, such as conceptual ideation, design development, fabrication and prototype testing will be executed in accordance to the requirements outlined by the CPE 460-461 class series. Any derivatives of or improvement made to the final prototype developed by Stand extends beyond the scope of this project.

Stakeholders

A successful prototype will ensure that the people who rely on fitness trackers will be motivated to take breaks often and walk hourly. With that said, the stakeholders involved in this project are: Lisa Yip, Dr. Hugh Smith, and customers who would be interested in using `stand`.

Lisa Yip is a student that studies Computer Engineering at California Polytechnic State University, San Luis Obispo. This project will be design, built and engineered at Cal Poly. Potentially, this project could be advertised to fitness and health related companies to promote a more interactive way of motivating users to take breaks.

Dr. Hugh Smith is the advisor for Stand senior project. He will be mentoring Lisa during the two quarter long project to ensure that she is applying the fundamentals of design process and are progressing the project properly.

The customers who are health conscious but tend to forget about their routine of taking breaks. They will be reminded to take breaks hourly if they haven't done so already.

Customer Requirements

The objective of this endeavor is to design, prototype, qualify and document a wearable device that can be used by any user. The following outlines the customer requirements as perceived by `stand`:

User Interface
- Must have a backend server so data can be stored on the cloud
- Must have an Apple iOS application that interacts and loads data to the user

Device Capabilities
- Must be able to detect step count
- Must notify the user if their hourly step count has not been met
- Must be able to save and transfer data

Operation
- Must be simple to use
- Must be low maintenance

Privacy
- Must have user authentication
- Must have security to protect user's data

Safety
- Must have a protective housing

The final prototype `stand`, must satisfy all of the above customer requirements in order to consider the endeavor a success.

Background

With wearable technology, learning more about oneself has not only become high-tech but also real-time. From fitness devices that help track activities to gadgets that monitor heart rate, wearable devices proliferate in today's market. A fitness tracker relays how active and healthy an individual is on a daily basis. Picking out the best fitness tracker can be a difficult prospect. The right activity tracker may be based on the individual's need; whether it's accurate step counting, reliable sleep tracking, or heart rate monitoring. It is an emerging technology that motivates me to learn about it.

The National Institute of Health (NIH) recommends at least 10,000 steps per day in order to maintain a healthy lifestyle. Studies show that those who increase their walking to 10,000 steps daily experience health benefits such as reduced blood pressure, healthier glucose levels, and reduced stroke risk. Unfortunately, people tend to forget to walk or take a break from work or studies when their schedule becomes busy. Research shows prolonged periods of inactivity, like sitting, decrease your body's production of fat-burning enzymes. The good news: that can be reversed with frequent, short walk breaks. Getting up and walking around for two minutes out every hour can increase your lifespan by 33 percent, compared to those who do not. [27]

The task is to design, develop, build, and test a low-budget wearable prototype that detects if the user is sitting for almost an hour, it reminds them to get up and walk throughout the day. The device will connect through an iOS application to monitor the average steps per hour.

Evaluation of Existing products

Due to the purpose of this project, it is essential to understand the human factors that are involved in the experience of the user. In order to most benefit the user, the wearable device and application must be easy to use without impeding their daily lives.

Each wearable device comes with similar features and they all aim to motivate and improve health. Some of these features include activity tracking, calories burned, sleep tracking, wireless syncing, food logging, sleep tracking, water resistance, and more capabilities. While the market for activity monitors continue to grow, the following examines some of the top activity monitors that can be purchased at most fitness stores.

Table 1 shows the current market cost of the top three devices by its cost, lasting of the battery from a full charge, and the weight of the system. [28]

Table 1 Evaluation of Existing Products

Fitness Devices	Cost ($)	Battery (days)	Weight (g)
Fitbit Charge HR	129.99	5	23
Jawbone UP4	129.99	7	29
Garmin Vivosmart	149.99	7	19

Many of the fitness devices have similar features, as shown in Table 2, such as activity tracking and heart rate. However, each device has features that can increase the cost of the product. Some of them may have more features than others. The next section goes in depth about its features.

Table 2 Continuation Evaluation of Existing Products

Fitness Device	Activity Tracking	Heart Rate Monitor	Sleep Detection	Caller ID	AmEx Payments	Smart Coach	GPS
Fitbit Charge HR	✔	✔	✔	✔	✘	✔	✘
Jawbone UP4	✔	✔	✔	✘	✔	✘	✘
Garmin Vivosmart	✔	✔	✘	✘	✘	✘	✔

Fitbit Charge HR

The Fitbit Charge HR mainly keeps tab on steps and calories. With the device, the user can stay connected with real time stats and Caller ID. It automatically detects when the user goes to sleep. One disadvantage about the Fitbit Charge HR is it does not have a built in GPS, which may result in miscalculated data for cycling and running.

Jawbone UP4

The Jawbone UP4 lacks a screen and active heart rate data, but has a smart coach that monitors the data to learn and recommends activities over time to make better health choices. It has a none month memory, giving a long time before data is overwritten by new data. This is also the only activity tracker offers wireless American Express payment, similar to the Apple Wallet or Google Wallet.

Garmin Vivosmart

The Garmin Vivosmart is a waterproof device that has an easily read display and could be customized to show any metrics desired. It also displays smartphone notifications, current weather along with music player and camera remote controls. This is the only fitness tracker that keeps a record of resting pulse and workout intensively. Some features that

none of the other devices offer is that it has an auto goal that keeps you challenged, GPS tracking as well as an inactivity alert that reminds you to move.

While these devices may be similar to `stand`, it is costly. Although the activity monitors track step count, none of these devices show how many steps were taken or remind the user to take a break. With `stand`, it will actively monitor to give a more comprehensive analysis of walking activity per hour and notify the user throughout the day to move.

Engineering Specifications

From the listed customer requirements in the introduction, a working table of engineering specifications could be made. A more detailed perspective on the testability of each engineering specification can be found on Table 3, which lists the formal engineering requirements. To be able to interpret the information, the concepts of risk and compliance must be explained.

Table 3 Formal Engineering Requirements

Index	Parameter Description	Requirement or Targets	Tolerance	Risk	Compliance
1	System Weight	30 g	10 g	M	T
2	Cost	$200	Max	L	A
3	Step Sensor	Detect Steps	Max	H	T, I
4	Blueooth LE	Interface between Arduino and iOS	--	H	T
5	Graphical User Interface	Mobile Application	--	M	T
6	Backend Database	User Authlentication and Data Store	--	L	T

A few of the requirements outlined above in Table 3 have been quantified in comparison to the other fitness trackers.

The tolerance is assessed by how much wiggle room each specification can have.

The risk for each specification will be assessed. They are assigned high (H), medium (M), and low (L). High risk requirements are typically harder to meet compared to low risk.

The formal engineering specifications include how each design requirement is to be met, also known as the compliance method. These methods are analysis (A), tests (T), similarity to existing designs (S), and inspection (I).

System Weight – The operating location of the wearable device will be located on the user's wrist, which limits the weight of the device so it's not a distraction.

Cost – The cost is limited by how much reimbursement is offered through the Computer Engineering department. In this case, the department can only reimburse up to $200.

Step Sensor – The step sensor will need to detect steps. Failure to do so will prevent the project from moving forward.

Bluetooth LE – This is the main source of communication between the hardware and software devices. Examples of the data being transferred are the hourly step counts and daily goals.

Graphical User Interface – Without the interface, the user will have no idea how many step counts are made throughout their day.

Backend Database – With a database, data can be stored securely without worrying about security issues between multiple users. Using a reliable database, its contents will be stored persistently.

Design Development

If the user does not walk within a certain threshold within an hour, the device would gently remind them to stand up and walk for a minute or until a certain amount of steps is made. The following devices are considered for this project:

Hardware

Microcontroller

A microcontroller is a simple, versatile, low-powered computer processer that can execute programs. It is small in size and inexpensive, enabling it to be a good starting point for various projects. Due to the scope of the project, the two most compact and popular microcontrollers are: Arduino Pro Mini, and Arduino Nano. The Arduino consumes very little power and is great for simple projects. Using any of the two microcontrollers will allow this project to take advantage of the open source software, to reference and build on. Both of the boards offers data receive and transmit through serial, inter-integrated circuit (I2C) and serial peripheral interface (SPI) communication.

Arduino Pro Mini

$9.95

The Arduino Pro Mini is a microcontroller board based on the ATmega328. It operates at 3.3V with a 8MHz clock speed, and has 14 digital input/output pins, 6 analog inputs, as well as a six pin header that can be connected to a FIDI cable or a breakout board to provide USB power and UART TTL serial communication to the board. [4]

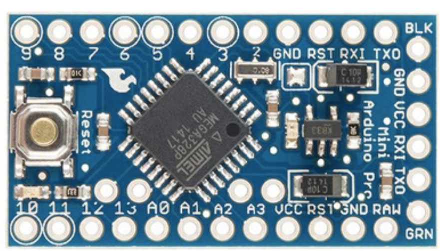


Figure 1 Arduino Pro Mini [3]

Arduino Nano

$6.54

The Arduino Nano is also a microcontroller board based on the ATmega328. It operates at 5V with a 16MHz clock speed, and has 14 digital input/output pins, 8 analog inputs, and will be operated though a Mini-B USB port. [1]

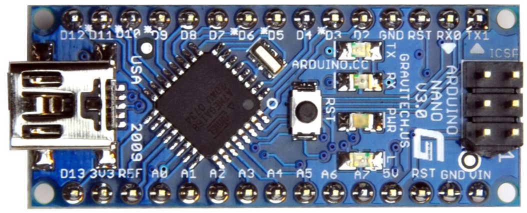


Figure 2 Arduino Nano [2]

Using any of the two can allow additional connections to various sensors and motor components for the project. The main difference about the two boards is the USB connection. Deciding between two boards will depend on how the board will operate and powered.

Wireless Communication

A Bluetooth Low Energy (Bluetooth LE) is a technology that uses wireless communication instead of cables and aims to simplify data synchronization between devices. Bluetooth LE is a technology that is aimed at applications such as healthcare, fitness/sport, security and entertainment. It is small, lightweight, inexpensive and uses little energy, making it a suitable candidate for the project.

The Bluetooth can be a device used to transfer data to and from the Arduino to the iOS application. It is a simple connection where it would connect to the Arduino using an SPI connection and the iOS via UART. These two modules are good candidates for the project because they are compatible with mobile devices such as iOS 7 or 8, Android 4.3 or above. This is a desirable feature as it can provide data syncing between devices.

Bluetooth LE nrf8001

$19.95

This module is designed specifically for applications that operate in Peripheral (Slave) role. It is the best-in-class power consumption enabling a long battery lifetime when running with coin cells. Data can be sent and received from up to 10 meters away and works by simulating a UART device between the two connected devices. Arduino made this Bluetooth easy by providing an already developed iOS app that has the UART connection configured. [6]

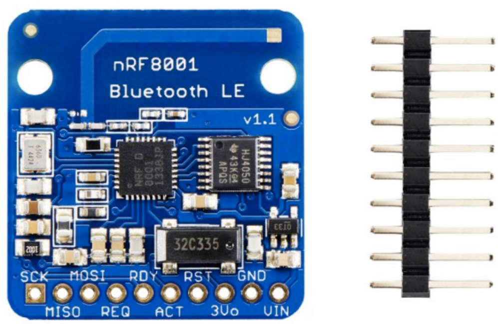


Figure 3 Bluetooth LE nRF8001 [5]

Bluetooth LE Nano

$33

The Bluetooth LE Nano, also known as nRF51822, is the smallest Bluetooth LE System-on-Chip development board. This module also supports Windows Phone 8.1, Windows 8.1, Mac OSX 10.9.2 and Linux with BlueZ with built-in Bluetooth 4.0. There is an already developed library for nRF51822 that will be release to users who register the module. [8]

Figure 4 Bluetooth LE Nano Kit [7]

Sensors

A sensor is a device whose purpose is to detect events or a change in its environment and provide feedback. Sensors will be used in this project to detect steps made by the user. With the possibility of this project, two different types of sensors were evaluated: STP10M pedometer and MPU4060 accelerometer. Both the boards are small and lightweight.

STP10M Pedometer

$5

The STP10M pulse-output interface embedded pedometer is a functional chipset that includes a G-sensor. This low power, small pedometer can give a high precision step count in any direction. The chip can be simply interfaced with other microcontrollers or embedded devices. [23]

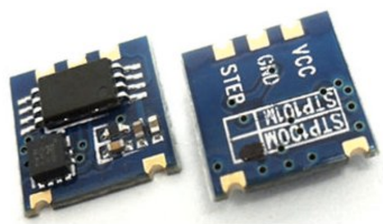


Figure 5 STP10M Pedometer [22]

InvenSense MPU-6050 Accelerometer

$39.95

MPU-6050 contains a MEMS accelerometer and gyro in a single chip. The sensor uses a I2C-bus to interface with the Arduino. It captures the x, y, and z channel at the same time due to its accurate 16-bit analog to digital conversion hardware for each channel. It contains a 1024 byte FIFO buffer and could easily be read by an Arduino. Using an accelerometer will mean that the walking algorithm will need to configured to properly detect a step. [19]

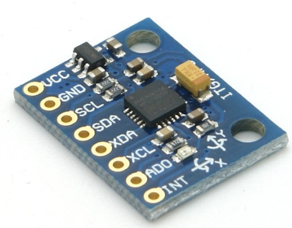


Figure 6 MPU6050 Accelerometer [18]

Vibration Mini Motor Disk

$1.95

A vibration motor will be used to gently remind the user to stand and move around when the number of steps are not met yet. This motor in particular, as shown on Figure 7, is a small and light motor that will be connected directly to the Arduino board to provide direct feedback. The motors are tiny disks with two wires to control the power of the vibration. The vibration of the motor will vary depending on the range of voltage input, between 2V to 5V. [25]

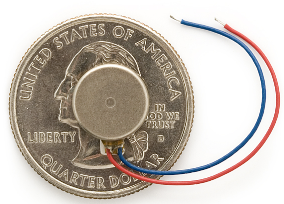


Figure 7 ROB-08449 ROHS Vibration Motor [24]

Battery

Picking a power source that suits the project will be depending on its size in relation to the Arduino Pro Mini. A battery – LiPro, coin cell, alkaline – may be a good choice. Any power supply that us greater than 3.3V and less than 12V will be able to power the Arduino.

Polymer Lithium Ion Battery

$9.95

These batteries are slim, lightweight, and has the highest density currently in production. It is an excellent choice for long-term self-discharge rates, and has robust power source under extreme conditions. Battery includes built-in protection against over voltage, over current, and minimum voltage. [20]

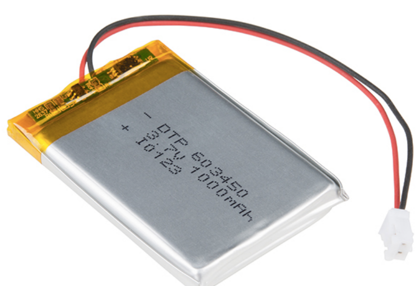


Figure 8 Polymer Lithium Ion Battery [21]

Coin Cell Battery

$1.95

This is a CR2032 Lithium metal 3V 250mAh button cell battery. It is great for powering low power processors or LEDs. [10]

Figure 9 Coin Cell Battery [11]

Alkaline Battery

$0.50

With a standard 1.5V 1500mAh AA battery, it can power the Arduino with just three batteries. These batteries are not rechargeable, however, they are cheap and good for testing purposes. [13]

Figure 10 AA Alkaline Battery [12]

Software

iOS Application

Due to limited time and budget, only one mobile application will be built and tested. Building the application on an iOS platform was chosen at the time, because it's the easiest to develop on. iOS is a mobile operating system created and developed by Apple Inc. and distributed exclusively for Apple hardware. Major versions of the iOS are released annually with current release being iOS 9.3.

The iOS application will be used as a GUI to gain comprehensive insight of their data and steps made throughout the day. Since this project is still a prototype, the application will not be published to the App Store.

Swift 2

This application will be developed and simulated on an Xcode IDE Version 7.3.1 and an iPhone 6. There are three main languages that can be used to develop for iOS: Objective-C, Swift, and Swift 2. Swift 2 is the newest released language and it has been redefined from ground up. It generates faster code in both release and debug builds, compiles fast and supports syntax suggestions. This language is also open source at Swift.org with regular development builds for everyone. Programming in Swift and Objective C can be more challenging than developing in Swift 2.

Data Store

Most modern applications store data and interact with other services on the internet. User accounts, shared content, documents and purchases; these things all need to be communicated and stored somewhere. Building this independently would require knowledge across many disciplines. Fortunately, there are platforms that already supports backend development on the cloud.

Using a platform that already has services such as user authentication and real time database updates is important. With user authentication, it provides a safe and quick way of transferring data, offering security for each user who is using the application. Real time data would mean that data would be serviced as soon as possible.

Parse

Parse is a BaaS company that provides an application development platform in the cloud. This service handles the backend and allows more time investment on the front end to enhance user's experience. With Parse, it abstracts the backend allowing a solo mobile developer to focus on the next great mobile app. Parse hosting services will be fully retired on January 28th, 2017, and recently became open source.

Parse server is an open source version of Parse backend that can be deployed to any infrastructure that can run Node.js. It is a self-hosted application and setting it up can be done in less than a day. It can be hosted anywhere, even the option of running from local instances to multiple instances in different regions to serve a global audience. For data storage, Parse server allows developers to control which database platform and file system they like to use. It offers backups, query performance, gaining raw access to data, free and open source along with all of its client SDKs.

Since it is open source version of Parse backend, setting up the fastest and easiest way is to have MongoDB and Parse Server locally. From there, any of the features such as user

authentication can be added on. However, in order to have this working on a mobile application, the server will need to be deployed to an infrastructure provider.

Firebase

Firebase is also a cloud services provider as a BaaS company. Since Firebase store and sync data with their NoSQL cloud database, data is synced real time and remains available even when the app is offline. Data is stored as JSON synchronized in real time to every client that is connected. Firebase supports iOS, Android, and JavaScript SDKs platforms.
It also offers custom business logic, offline synchronization management, and in app purchases. It is a great option for real time multiple device communication. Firebase features services such as user authentication, database, storage, hosting, remote configuration, test lab and crash reporting.

Data Flow

With raw values being produced by the sensor, the values will be directly transferred, using I2C, to the Arduino. Since the values come in sequential order, it is safe to assume that the raw values read come in X, Y, and Z, accordingly. Once the raw values is on the Arduino board, the magnitude is computed with the following equation:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$ [26]

The Arduino will be outputting this data and with the connected Bluetooth LE, the data will be sent to the iOS application. From there data will then be uploaded to the server.

Final Design Concept

Hardware

One of the subdomains of Stand is the hardware components. The following provides insight on the final design for the electronics region, including explanations for microcontroller design, power supply selection, and interfacing with sensors present in the system.

Arduino Pro Mini 3.3V

The Arduino Pro Mini was chosen for its compact size, and the ease of powering the board using a battery through the use of two pins. The FIDI SPI module was used to communicate between the serial console and board during debugging.

Step Sensor

At first, the pedometer sensor was chosen since it offered a direct way of detecting step counts with just three pins, digital pulse output, voltage. However, after multiple attempts of testing the pedometer, it seemed that the board was dead on arrival. The output of the digital pulse output should be low most the time, and becomes high only when a step is made. However, when using an oscilloscope to test the input and output of the sensor, the pulse digital output remained high the entire time.

Since it took the sensor two weeks to arrive upon ordering, reordering the part was not an option because that will cause significant delays in the project. An accelerometer was used instead to detect step count. Although the step count values are not as accurate as desired, improvements of the algorithm will be in future development.

Batteries

Unfortunately, this project did not use the batteries to test the device. The entire prototype was powered by a serial module USB cable adapter. For future considerations, the polymer lithium ion battery is good candidate due to its size and energy.

Vibration Motor

The vibration motor works as expected. With the ground wire connected to ground and voltage in connected to a pin on the Arduino, it vibrates when the pin is high, otherwise stays idle. The intensity of the vibration could also be dampened by controlling the voltage.

Bluetooth LE

The Bluetooth LE nRF8001 module was chosen due to its simplistic features. Since the prototype just needed to connect and transfer data to the iOS, it didn't require additional features that the Bluetooth LE Nano had. Since there was a provided iOS application to the nRF8001 module, it is hypothesized that development would be more

straightforward. The included application was tightly coupled, making custom use more difficult.

Data Transportation

After the user has produced an export file containing the data specifying the analysis of hourly step count, the next step is to convert the data into an acceptable format in order to upload to Firebase. A web application is used to read files and upload data to Firebase through JavaScript.

Initial Architecture

The initial architecture of the dataflow, shown in Figure X, would start from the pedometer, where every step will be transferred via I2C to the Arduino. The Arduino will have two tasks running concurrently: sending and receiving data via Bluetooth LE and making sure the user is meeting the step count goal per hour. The vibration motor will vibrate if the minimum number of steps is not met within the hour. Bluetooth LE is a form of communication between the iOS application and the Arduino. Once data is transferred to the iOS application, it will be uploaded to the Parse Servers with iOS SDK.

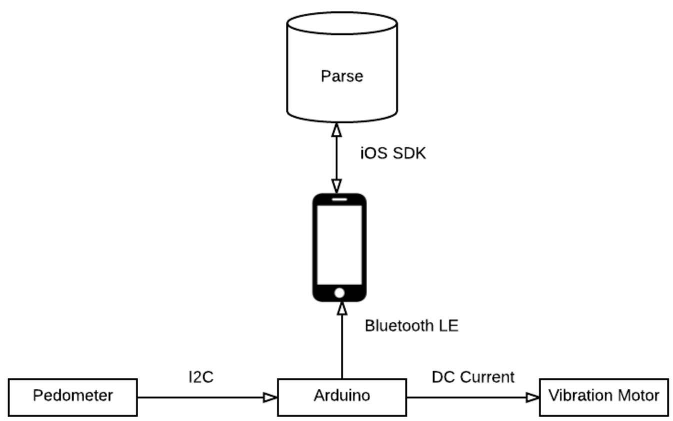


Figure 11 High Level Prospective of Initial Design

Secondary Architecture

After a duration of successful development locally between the Parse Server and iOS device, it was finally time to host the server remotely. Doing so will actually allow the application to functionally work off of an iOS application, instead of the simulator in Xcode. Hosting a Parse Server remotely proved to be more challenging than expected. The CRT had issues authenticating the SSL connection. After a week of lead time, making no progress, time and efforts wasted, it was decided that `stand` should use another BaaS platform. Firebase was decided due to its features and support and the new architecture is shown below, Figure 12.

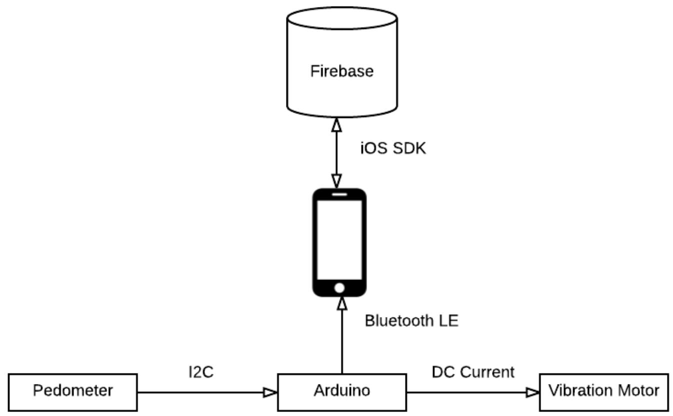


Figure 12 High Level Prospective of Initial Design with Parse Server

Final Architecture

With the successful implementation and migration from Parse Server to Firebase, there were other obstacles that delayed the development of the prototype. After several attempts of debugging through supported open source repositories, Core Bluetooth SDK, and other documentations, the Bluetooth LE was unable to send data to the iOS application.

Even though the Bluetooth LE could receive and send data on the Arduino side, the iOS application could only send data to the Bluetooth LE, but unable to receive. However, there is a workaround to this problem. Instead of communicating via Bluetooth LE to the iOS application, prototype will then use serial output to a computer in order to transfer data to a more capable device than rely on Bluetooth. The data will be uploaded to Firebase via a web page. Once data is successfully uploaded to Firebase, the iOS application will be able to refresh and the uploaded data will be ready to display to the user.

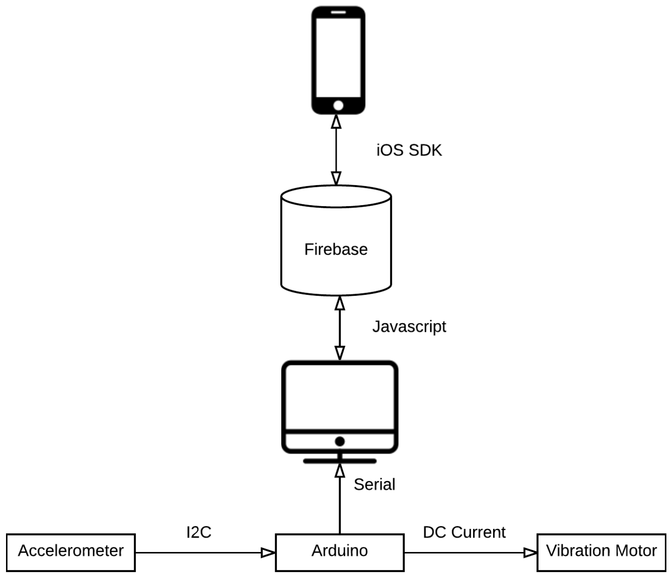


Figure 13 High Level Prospective of Final Design with Firebase

Software

The other subdomain of Stand is the software components. Following the conceptual design of the software section, further research and implementation were done to move the project forward.

iOS Application

The main features that an iOS application needs would be user authentication and the ability to store data. User accounts are stored in Firebase, which collects their email address, username, and hashed password.

Login View Controller

When the user launches the application, the app will automatically check if the user has previously logged in before. If they have, then it redirects them to the home view controller in Figure 18. However, if it is their first time on the application or they have never logged in before, the user will see the login view controller in Figure 14. This view controller gives the user three different choices: login if they already have user credentials authenticated, sign up if they need to create a new account, or reset their password if they have forgotten. Having the user login is vital because it protects data privacy.

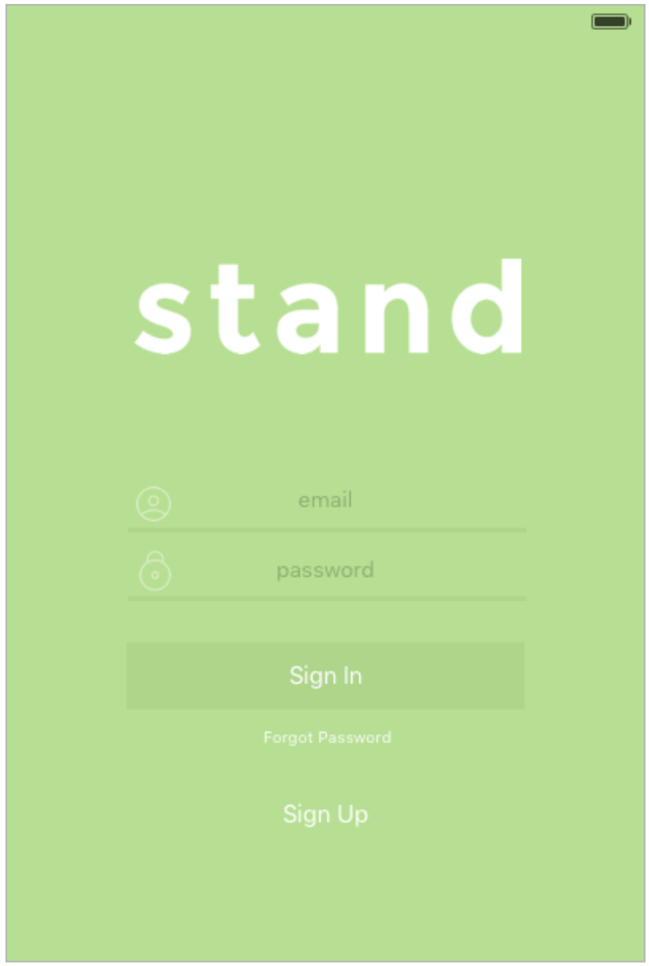


Figure 14 Login View Controller

Sign Up View Controller

If the user decides to create a new user from the login view controller, they will be prompted the sign up view controller, Figure 15, to create a new user. The sign up view controller prompts the user for their full name, email, and password. The application makes sure a proper email is put in using regular expressions, *Figure 16*. If the email is not valid, an error will be alerted and the user will not be created. The password must be greater than eight characters or an error will occur and the user will not be created.

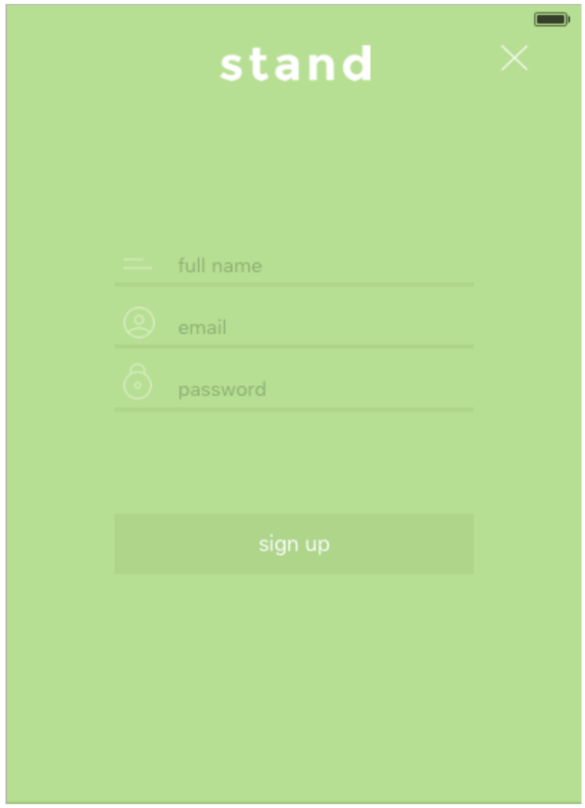


Figure 15 Sign Up View Controller

```
func isValidEamil (email: String Bool) {
let "[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}"
let NSPredicate(format: "SELF MATCHES %@, emailRegEx)

return emailTest.evaluateWithObject(email)
}
```

Figure 16 Email with Regular Expressions Validation

Forgot Password View Controller

In the case that the user forgets their password to the account, a forgot password is featured as a button on the login view controller. Once clicked, the forget password view controller will show up, Figure 17. Once the user types in the email address, an email will be sent to them, the Reset Password button will disappear and a label saying, “Please check your email” will take the place of the button.

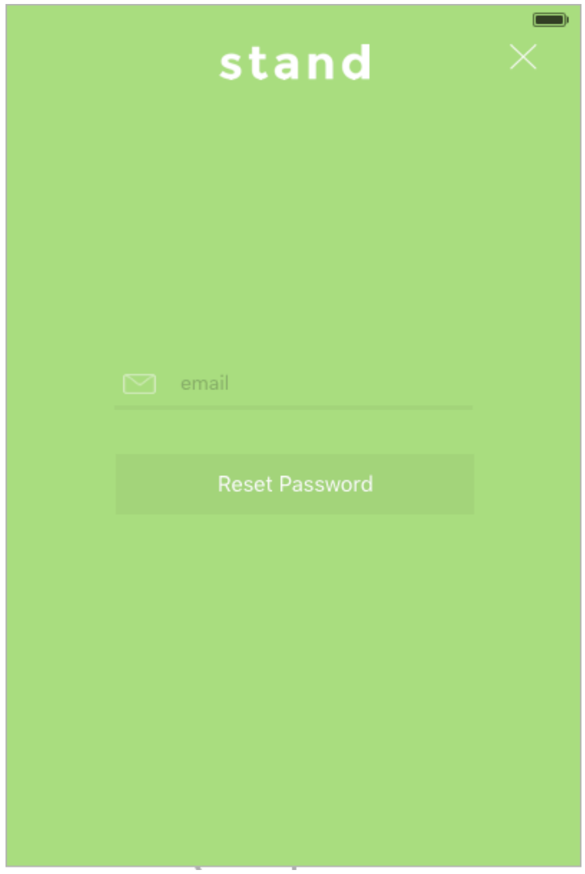


Figure 17 Forgot Password View Controller

Home View Controller

The home view controller is where all the data from the Arduino will be located. It shows the user how many hours out of the day did they reach the step count. On the bottom of the controller, there is a chart that shows a more detailed view of how many steps are taken per hour.

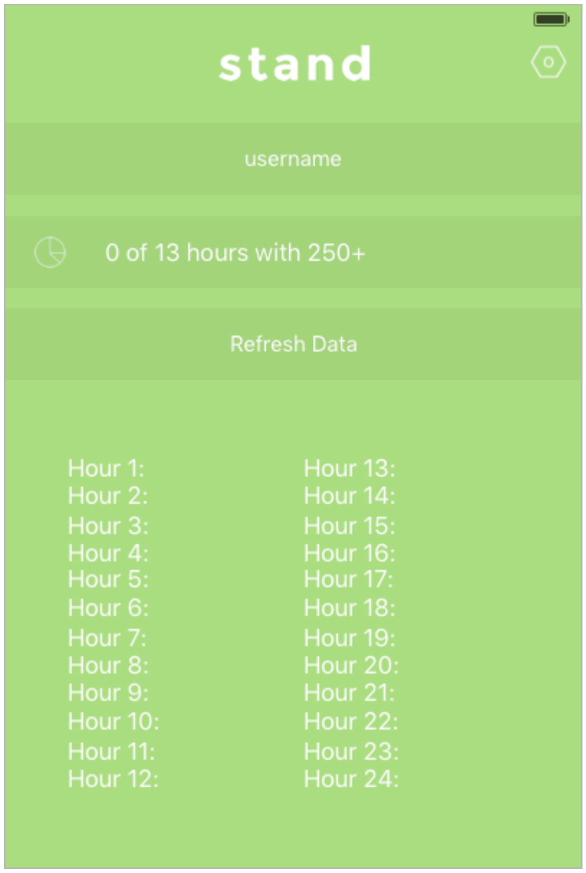


Figure 18 Home View Controller

Settings View Controller

The settings view controller can be access via a small hexagon icon at the top right corner of the home view controller. This controller shows the current logged in user, allows the user to pull or refresh to the most up-to-date data from Firebase. The goal portion of this application is not yet implemented due because the Bluetooth LE did not work. Had it worked, there would be a goal view controller where it can allow the user to set different steps counts per day or per hour. Last but not least, this controller has a log out button that logs the user out of their account and brings them back to the login view controller.

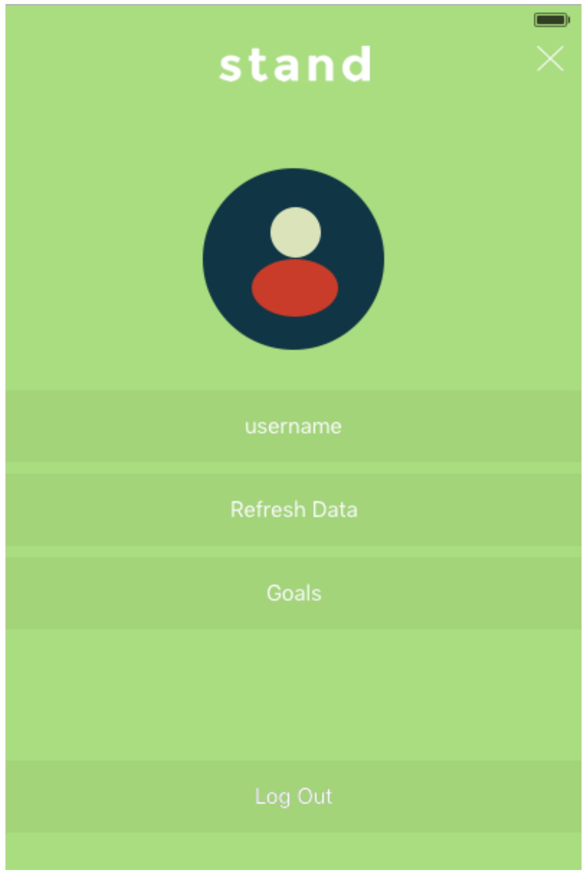


Figure 19 Settings View Controller

Arduino

The Arduino is the main device that does all the computing. With data being sent from the accelerometer to the microcontroller, it is analyzed and then saved.

Accelerometer

Accelerometer does not have built-in pedometer counter inside, so logic using the forces measured by the accelerometer is used to detect a step. Future improvements will create a more define step count analysis.

Saving Data

Data is saved by outputting data from the Arduino into a text dump from the serial module onto the computer. The data is then uploaded to a with a web page that will compute and uploaded into a structured data store.

Data Store

With Parse being a widely used BaaS platform, it was initially chosen because it supported many platforms such as iOS, Arduino, Android, JavaScript, OSX, Unity, much more. All the platform are well documented with different API references and tools.

However, since it started to decommission beginning of the year in 2016, all the services had to be implemented locally. Since all testing can easily be done locally, Parse Server was running locally on a machine where the application was built in Xcode. The application was developed and tested with a simulator, proving that the platform worked. However, when testing the application on an iPhone 6, Parse Server had to be accessed remotely via a server in order to support mobile access. This is because the configurations cannot access the local server but can access any servers that are created for remote access.

Web hosting and domain names were created to support Parse Server, but the process proved to be more challenging than expected. There are Security Sockets Layer (SSL) issues when setting up the server. SSL is needed in order to provide a secure connection, allowing private data to be transmitted cryptographically. To do so, a certificate signing request (CSR) had to be created and installed into the web server. After countless hours of customer support and attempts to create a secure site for data transfer, the efforts were futile.

Even though Parse is a popular platform, its decommissioning created a significant challenge to the project. Since the project's main needs for a BaaS are user authentication and data storage of user configurations and step count, other BaaS platforms were assessed. The main concerns for the project was to have short development time, be able to show data in real time, and allow the application to scale. Firebase was chosen because it satisfied all the concerns.

With Firebase, there are no need to create anything locally. All the set up needed was created through the Firebase dashboard. With its support for iOS SDK's, it opens doors for mobile application development. With the support of documentation and forums, setting up Firebase proved to be less complicated. The database, and user authentication, along with Facebook authentication, was set up quickly and data could be uploaded directly to the server, which gave the user real time data on the iOS application.

Cost Breakdown

In order to determine the cost of a prototype on a manufacturing level, the cost of one prototype will need to be determined. Table 4 shows the build of materials for the entire prototype, assuming the Bluetooth LE was functional. The total cost of this prototype came out to be $84.75. In order to configure the Arduino Pro Mini, a one time purchase of a $7.99 serial adapter module is required, making the entire prototype to be $92.74.

Table 4 Build of Materials Breakdown

Index	Product	Price
1	Arduino Pro Mini	9.95
2	Bluetooth LE nRF8001	19.95
3	InvenSense MPU-6050 Accelerometer	39.95
4	Vibbration Mini Motor	4.95
5	Polymer Lituium Ion Battery	9.95

Design Verification

This section proves the results of the design verification plan.

Table 5 Design Specifications with Testing Results

Index	Parameter Description	Requirement or Targets	Actual Results	Verdict
1	System Weight	30 g	53g	✘
2	Cost	$200	$92.74	✔
3	Step Sensor	Detect Steps	Detect Steps	✔
4	Blueooth LE	Interface between Arduino and iOS	Using serial and Firebase	✘
5	Graphical User Interface	Mobile Application	iOS Mobile Application	✔
6	Backend Database	User Authlentication and Data Store	Firebase	✔

Prototype Evaluation

Cost – The total cost of the prototype turned out to be less than the budget, making it within tolerance.

Step Sensor – The accelerometer sensor was able to detect a step during the test phases, passing the requirements.

Graphical User Interface – An interface is provided to the user as a mobile application to interact with their personal data. This interface is graphical with interactive buttons, therefore passing the requirement of having a graphical user interface.

Backend Database – `stand` was able to use Firebase as a backend database. This allows user authentication and data stored remotely, passing the requirements.

Failed or Untested Specs

System Weight – The weight of the hardware modules was estimated to be about 7g since each module is about 1-2g. However, the Alkaline AA batteries weigh about 23g each. This totals the system weight to 56g, failing the system requirements. Since the main task is to build a proof of concept for the prototype, the alkaline batteries can be later replaced with other batteries.

Bluetooth LE – The Bluetooth LE was not fully implemented due to the lack of time and poor planning.

Recommendations

Upon the completion of the Senior Project effort, it is clear that there is more work that can be done to improve the current state of the prototype. The following recommendations relate to the scope of this project:

1. **Functionally working Bluetooth LE connection**

   Without a working Bluetooth LE connection between the Arduino and iOS application, the current solution requires more manual steps to transfer data, leading to inconvenience in the user's experience.

2. **Change data flow from sensor to iOS application**

   The current flow that the data is acquired and presented to the user is through manually uploading data to Firebase from Arduino then the data will show on the iOS application upon refresh. Due to time constraints, the Bluetooth LE was not completely implemented. Future efforts should focus on transferring data from Arduino to Bluetooth LE to the iOS application, then uploading the data to Firebase.

3. **Find a more structured algorithm for steps using accelerometer**

   Although this prototype is just in its testing phases, the current accelerometer algorithm is not fully functional. It is a proof of concept and it may record uncorrelated data. In future improvements, a more proper step count algorithm should be implemented.

4. **Add Features**

   Adding features to this prototype will be desirable for the customer. Such features may include an alarm clock, custom made goals, and data logging of different activities.

5. **Design a case for the hardware**

   For later iterations, a case encasing all the electrical components should be made. Doing so will protect the user from any sharp edges or static shocks. The device will also become more marketable.

Acknowledgements

Despite the efforts made by Stand, this project could not have been assembled within the set timeframe without the assistance of Dr. Hugh Smith.

Dr. Hugh Smith served as the senior project advisor, and was integral in the design development of Stand. The knowledge he had guided the project in the proper direction, while allowing the myself to make decisions on the project.

References

[1] "Arduino Board Nano." *Arduino - ArduinoBoardNano*. N.p., n.d. Web. 29 May 2016. <https://www.arduino.cc/en/Main/ArduinoBoardNano>.

[2] *Arduino Board Nano*. Digital image. N.p., n.d. Web.

[3] *Arduino Pro Mini*. Digital image. N.p., n.d. Web. 29 May 2016.

[4] "Arduino Pro Mini." N.p., n.d. Web. 29 May 2016. <https://www.sparkfun.com/products/11114>.

[5] *Bluefruit LE - Bluetooth Low Energy (BLE 4.0) - NRF8001 Breakout.* Digital image. N.p., n.d. Web. 29 May 2016.

[6] "Bluefruit LE - Bluetooth Low Energy (BLE 4.0) - NRF8001 Breakout." N.p., n.d. Web. 29 May 2016. <https://www.adafruit.com/product/1697>.

[7] Bluetooth *LE Nano Kit*. Digital image. N.p., n.d. Web.

[8] "Bluetooth LE Nano Kit." N.p., n.d. Web. 29 May 2016. <http://store-usa.arduino.cc/products/e000060>.

[9] "BluetoothKit." *GitHub*. N.p., n.d. Web. 29 May 2016. <https://github.com/rhummelmose/BluetoothKit>.

[10] "Coin Cell Battery - 20mm (CR2032)." N.p., n.d. Web. 31 May 2016. <https://www.sparkfun.com/products/338?_ga=1.21443644.681093268.1460075052>.

[11] *Coin Cell Battery*. Digital image. N.p., n.d. Web.

[12] *1500 MAh Alkaline Battery*. Digital image. N.p., n.d. Web.

[13] "1500 MAh Alkaline Battery." N.p., n.d. Web. <https://www.sparkfun.com/products/9100?_ga=1.21443644.681093268.1460075052>.

[14] FTDI USB to TTL Serial Adapter Module. Digital image. N.p., n.d. Web. 29 May 2016.

[15] "FTDI Usb to TTL Serial Adapter Module for Arduino Mini." N.p., n.d. Web. 29 May 2016. <http://www.amazon.com/Qunqi-FT232RL-Serial-Adapter-Arduino/dp/B014Y1IMNM?ie=UTF8&psc=1&redirect=true&ref_=oh_aui_detailpage_o02_s00>.

[16] "How To Get the Arduino Time Library up and Going." N.p., 11 Apr. 2014. Web. 31 May 2016. <https://oopsohno.wordpress.com/2014/04/11/how-to-get-the-arduino-time-library-up-and-going/>.

[17] "How to Use IOS Charts API to Create Beautiful Charts in Swift." N.p., n.d. Web. 31 May 2016. <http://www.appcoda.com/ios-charts-api-tutorial/>.

[18] "MPU-6050 Accelerometer + Gyro." *Arduino Playground*. N.p., n.d. Web. 29 May 2016. <http://playground.arduino.cc/Main/MPU-6050>.

[19] *MPU-6050 Accelerometer + Gyro.* Digital image. N.p., n.d. Web.

[20] "Polymer Lithium Ion Battery - 1000mAh." N.p., n.d. Web. 31 May 2016. <https://www.sparkfun.com/products/13813>.

[21] *Polymer Lithium Ion Battery*. Digital image. N.p., n.d. Web.

[22] *Pulse-output Interface Embedded Pedometer Module*. Digital image. N.p., n.d. Web. 29 May 2016.

[23] "Pulse-output Interface Embedded Pedometer Module." *Tindie*. N.p., n.d. Web. 29 May 2016. <https://www.tindie.com/products/NiceRF/pulse-output-interface-embedded-pedometer-module/>.

[24] *Vibration Motor*. Digital image. N.p., n.d. Web.

References

[25] "Vibration Motor." N.p., n.d. Web. 29 May 2016. <https://www.adafruit.com/products/1201?gclid=CjwKEAjwpqq6BRC99aKUkaSjuDsSJAC0pNTVvF-8nefhqSX7WQ0GHRtFPlQbpeN7mcMg2kuaga6ncRoCExjw_wcB>.

[26] "3D Vectors." *3D Vectors*. N.p., n.d. Web. 09 June 2016. <http://members.tripod.com/~Paul_Kirby/vector/VLintro.html>.

[27] "Weight Loss Dietitian Blog." *Seriously? You Want Me to Walk How Many Steps per Day?* N.p., n.d. Web. 09 June 2016. <http://www.mynetdiary.com/seriously-you-want-me-to-walk-how-many-steps-per.html>.

[28] "Best Fitness Trackers 2016: Jawbone, Misfit, Fitbit, Garmin and More." N.p., n.d. Web. <http%3A%2F%2Fwww.wareable.com%2Ffitness-trackers%2Fthe-best-fitness-tracker>.

Appendices

Source Code

Vendor Datasheets

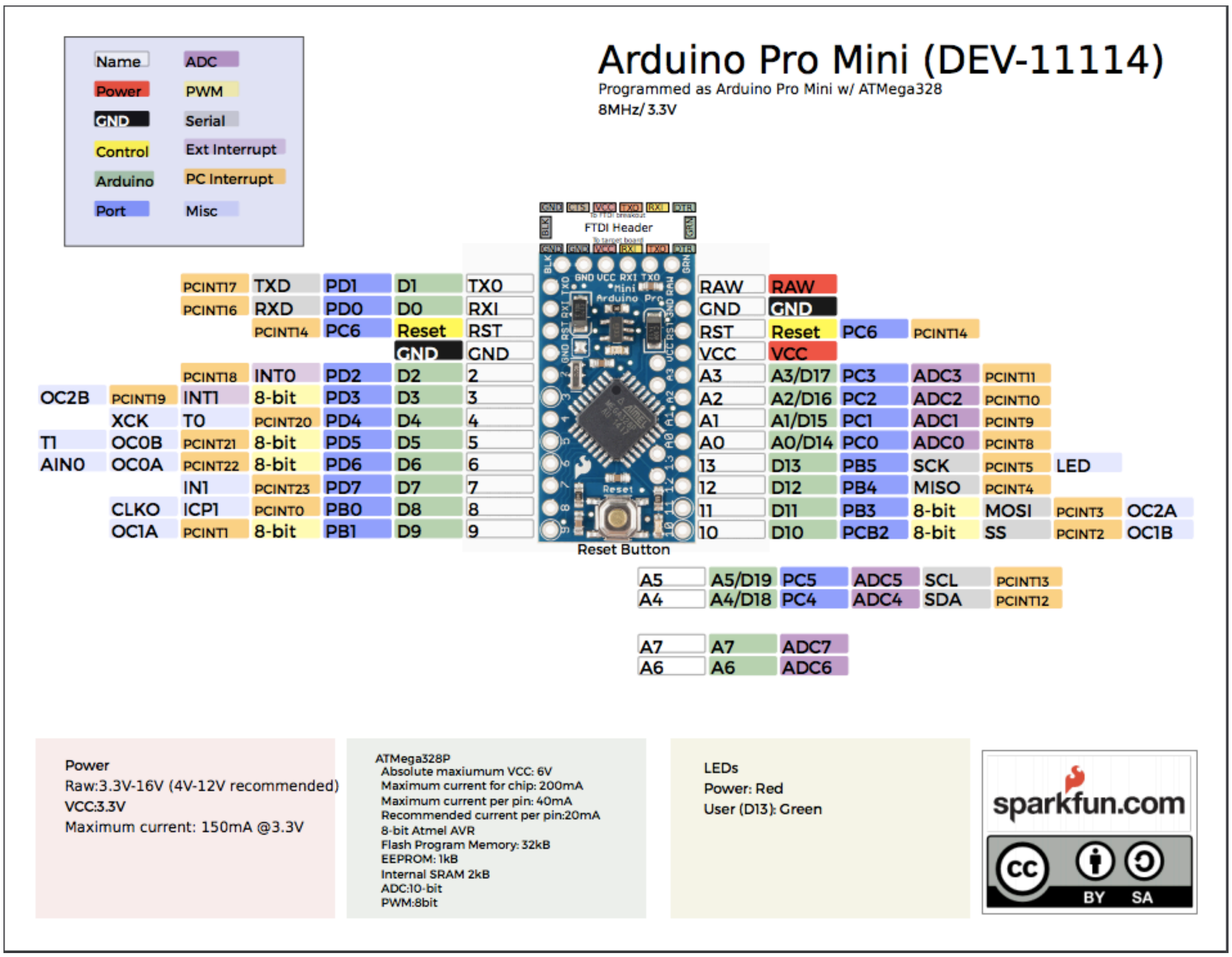
Arduino Pro Mini

Bluetooth nRF8001

MPU 6050 Accelerometer

Pin Out Diagrams

Arduino Pro Mini



Bluetooth nRF8001

MPU 6050 Accelerometer