PROJECT RUNWAY

Making Treadmills Smart Again

10th Feb 2017

18-549 Team 13
Hyeryung Helen Cho (hyeryung)
Amy Lin (awlin1)
Brian Ho (tzeyewbh)

Table of Contents

Project Description

The objective of this project is to build a device that can be easily mounted on a generic, conventional treadmill, providing integration and services similar to a smart, internet-connected treadmill. Internet-connected treadmills currently offer various services, such as speed and incline tracking, course tracking and calorie tracking. However, they are significantly more expensive than conventional treadmills. In addition, many of these advanced treadmills have difficult and complicated user interfaces.

Our goal is to develop a device and a companion smartphone application that would become an affordable alternative to these costly treadmills. By simply mounting this device to a conventional treadmill, users would be able to access most of the services that smart treadmills offer. It is our goal to make the treadmill integrate more closely with external libraries and APIs so that users can gain more actionable insights from their indoor workouts.

Design Requirements

Explicit Requirements

1. On hardware side, the device is able to:
    1. Use the tilt sensor and accelerometer to measure incline of treadmill
    2. Ingest raw vibration data from treadmill; and
    3. Filter and smooth the data to get step count data from it
    4. Use a mechanical wheel and sensor to measure speed of treadmill; and
    5. Convert this kinetic energy to power the device.
    6. Encapsulate each data type with a timestamp in a packet of arbitrary protocol
    7. Send this data via Bluetooth to a smartphone application
2. The smartphone is able to:
    1. Present the treadmill's velocity, distance, incline and step count to the user.
    2. Provides recommended exercise habits
    3. Provides user with insights on correlation (such as incline on drop off in speed)
    4. Has open integration with external apps or APIs wishing to build off it
3. In the general form, we want the device to be:
    1. Easily fixed on to any conventional treadmill
    2. Easy to set up and use
    3. Robust and durable

Implicit Requirements

1. Poses no physical threat to the treadmill user
2. Uses relatively less power than the treadmill
3. Does not affect long term durability of treadmill

Functional Architecture

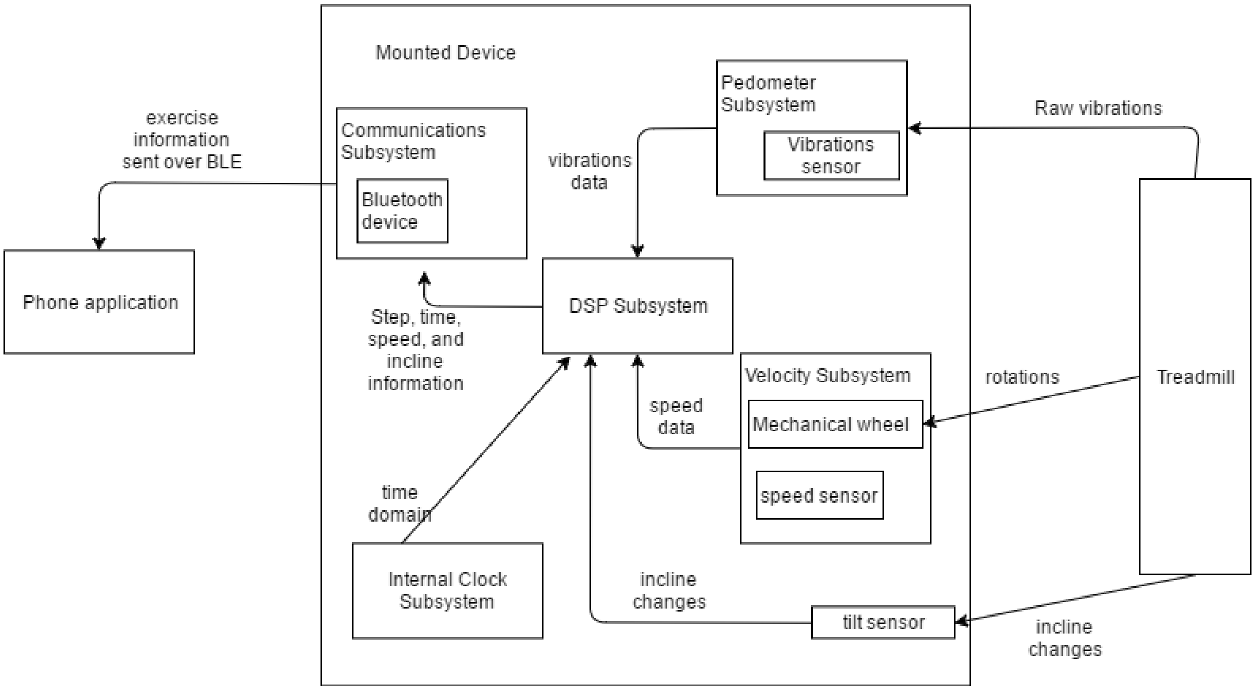


Figure 1 - System Block Diagram

The primary function of this project is to provide information to the user concerning their experience on the conventional treadmill. This information, including the treadmill speed(s), the treadmill incline, and recommended exercise habits, will all be delivered to them through a smartphone application. The user will be able to easily interface with the application to keep their exercise experience recorded. The application ingests this information by processing the data sent via Bluetooth from the mounted module. The mounted module will gather the information on the treadmill experience via numerous embedded sensors. A vibrations sensor will measure the step count. The velocity subsystem, including a mechanical wheel and a speed sensor, will record the speed of the treadmill. Lastly, a tilt sensor will gauge the treadmill height and provide the incline information.

Design Trade Studies

Measurement of Treadmill Speed

To accurately measure the velocity the treadmill is running at, there were two main options that we considered for accurately – optical or mechanical. Each method has its own pros and cons. The optical approach is likely to be more durable as there no moving parts in its operation. An RGB color sensor (e.g. Adafruit TCS34725) can be elevated over the track in a treadmill tracking the frequency of a contrasting colored dot that is affixed on the treadmill. However, this entails some user calibration and setup, since treadmill tracks vary in length and width. Also, this method is susceptible to sudden changes in ambient lightning which might affect measurements. The treadmill also needs to be used in a relatively bright area so that the RGB color sensor can function well, though this problem can be solved by building in a soft glow LED light together with the color sensor.

On the other hand, a mechanical approach seems more intuitive. In this approach a rubberize wheel would be placed onto the track. As the treadmill runs, the wheel would spin and the velocity can be inferred from the rate of revolutions. Unlike the previous approach, there is no need for an external power source as we can integrate a DC motor/generator that converts kinetic energy to electricity needed to power the circuit. The end user experience is also better as there is no user calibration required – said wheels should be pre-calibrated at factory. However, one concern is that the wheels would experience wear and tear. Dirt and grime on the track might also affect the accuracy of results. Also, it is noteworthy that initial measurements might not be accurate due to static and kinetic friction differences – hence the is a need for the wheel to be weighted so that it presses on the treadmill and not just 'skip' along it.

At this moment, we are more inclined to go with the mechanical approach because it makes the resulting device self-contained without the need for external power sources, though we need to work around its deficiencies.

External Device Communication

There are a variety of communication protocols to pick from but the ones that stand out from IoT devices on the market right now are Wi-Fi and Bluetooth. In the former, a hypothetical device sends its data to a central server directly while a companion smartphone app pulls that data from the server to be displayed to the user. The latter entails the device sending its data to a smartphone app which would be opened, with the latter periodically sending data to a central server for storage. The final decision choice is driven by several factors – the need for computational power, energy efficiency, data rate as well as liveliness of information.

We have decided on Bluetooth as our main external communications protocol as the primary goal for our device is to complement a vanilla treadmill while the user is exercising. In this case, there would be a companion smartphone app that the user opens while exercising, giving them a heads-up display of their live activity. Hence, it makes less sense to communicate to a central server, then push real-time updates back to the phone; it is more intuitive to talk to the phone app directly, then push periodic updates to a central server as necessary. In this case, Bluetooth is a better protocol for communicating with smartphones. Even though Bluetooth is expected to have a lower data rate than Wi-Fi, we do not foresee it being a bottleneck since we believe the updates/heartbeats are very small, in magnitudes of single Kilobytes per second.

Accurate Measurement of Step Count

One useful information to get while exercising is the step count. Many of the current implementations involve a wearable that tracks step count but an alternative and more convenient way of recording steps could involve putting one or more gyroscopes, accelerometers or vibration sensors inside the device to detect the rhythmic movement of the feet as they hit the ground with each stride. From this, we are able to infer stride length in relation to other variables such as incline and distance to provide actionable feedback to users. Of course, the easier way would be to get a Fitbit and use their API to pull step data out of it but this might cause unnecessary friction in the user experience. Our method allows the user to simply turn on the treadmill and start running without needing to strap on accessories. This method is also a problem in digital signal processing – how to convert raw telemetry data from aforementioned sensors to a concrete number – step count.

Device Placement

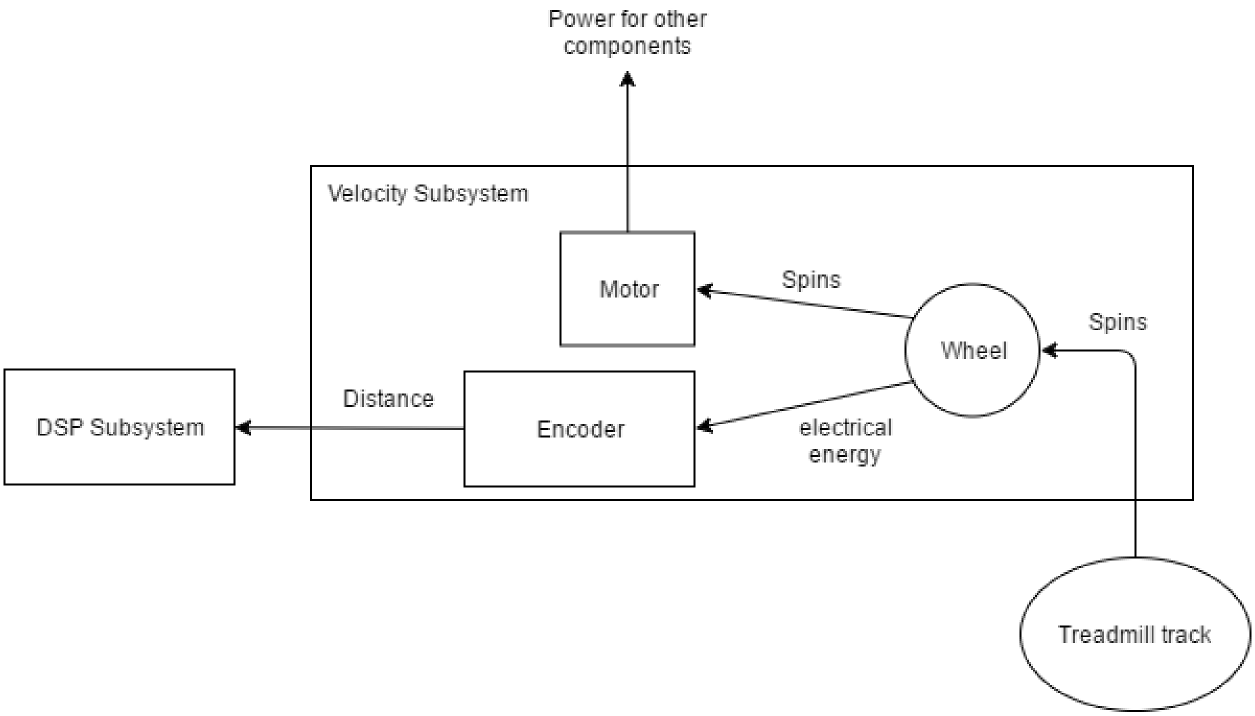
One seemingly trivial problem concerns the placement of the Device. However, this has wider implications in its safe operation. Off the bat, there are three possible positions that we might place the device, as seen below. From our experience and analysis of running gait and stride, we can rule out position 3 as that can seriously trip the runner if their heels hit it. Position 2 is a better choice if we are using an optical method to track the contrasting white dots (shown here). However, since we are using a mechanical approach to track speed, the most out-of-sight position would be at position 1, where it is least likely to obstruct the user and safest.

Figure 2 - Image courtesy of Lifespan Fitness (https://www.lifespanfitness.com/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/9/0/900x900_tr1200-dt3-top-down.jpg)

System Description

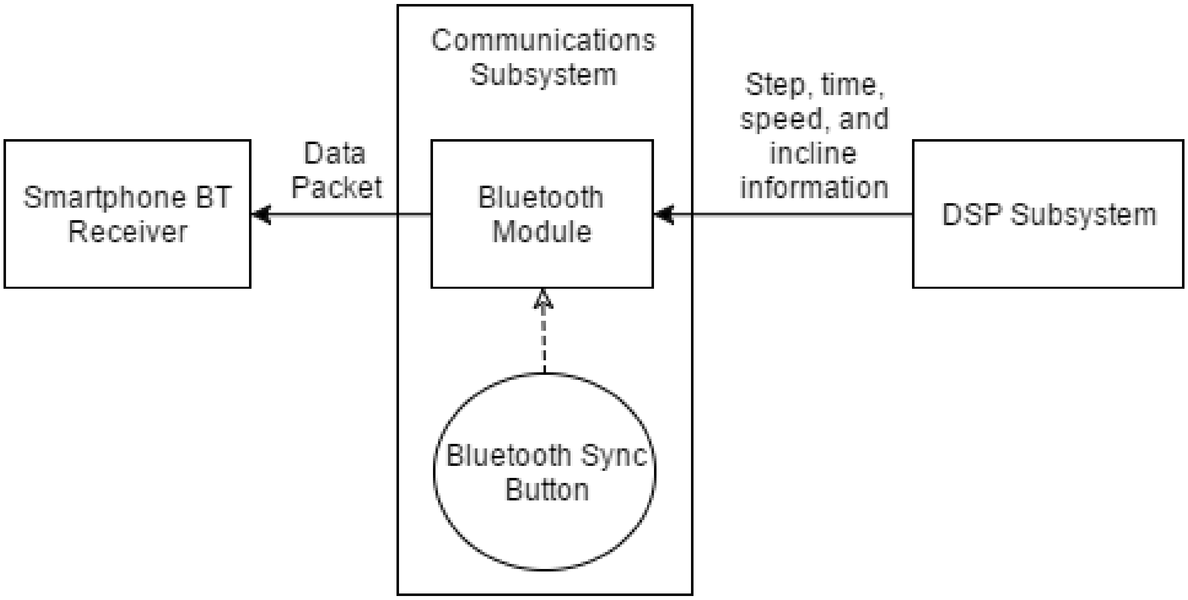
There are various subsystems in play that would make this device a reality. One of the main subsystems involve tracking the speed of the treadmill. To do so, we need time provided by a quartz oscillator or equivalent, as well as the distance moved in that time period. This distance value is an output the Velocity Subsystem as shown below.

Velocity Subsystem



In this set up, a wheel rests on top of the treadmill track. When the track moves, the wheel spins as well. This angular movement rotates a DC motor/generator which provides electricity for the rest of the system. Also, wheel encoders embedded in the motor tracks the revolutions that the wheel has spun. Since we know the circumference of the wheel, we can easily calculate the distance travelled by the track. This value is passed to the DSP subsystem for further processing – to calculate the real-time speed of the treadmill.

Communications Subsystem



Another major component of our system would be the Communications Subsystem which is responsible for external data communication of the device with the smartphone app. The Bluetooth module is paired with the smartphone via a hardware pushbutton. After pairing, the module is able to send data packets to the smartphone. In order to ensure robust data transmission, we have developed a data transmission protocol as follows.

Each outgoing packet has a timestamp, integer indicating data length, the payload (with an optional opcode indicating a single type of step, time, speed or incline), as well as a single byte checksum. Upon receipt by the smartphone Bluetooth module, the companion app will read each packet and verify its integrity. Upon success, it replies with an ACK with the timestamp as the payload. The device Bluetooth module will retransmit the same packet if an ACK is not received within a certain time period to be determined. This packet will be read from an internal buffer pulled in from the DSP module or MCU. It is expected that we store at least 1 second worth of data in the buffers, noting that the latency for the oldest generation of Bluetooth is 100ms. Modern smartphones have latency of less than 10ms.

Secondary Subsystems

There are several other systems that play a role in the device but have similar functionality to the ones described above. The interested reader can take a look at Figure 1 to see all the subsystems in play. A more interesting subsystem would be the pedometer subsystem which is used for step count and tracking. As mentioned earlier, our method of step tracking uses two or more vibration sensors to sense the rhythmic vibrations as each foot hit the treadmill as a way to infer step data. This allows the user to simply turn on the treadmill and start running without needing to strap on telemetry accessories. This method is also an interesting problem in digital signal processing (DSP) – how to convert raw telemetry data from aforementioned sensors to a concrete number in step count. This processing is done in the MCU and the DSP hardware contained therein – a system of filters and capacitors.

The Incline Subsystem makes use of a tilt sensor and an accelerometer to determine the incline and change in incline. The Clock Subsystem is also a simple 1-element system consisting of a quartz oscillator or equivalent to keep track of time.

Project Management

Schedule

Date	Goal	Note
2/13	Proposal Presentation	Finish fine tuning design specifications and objectives
2/24	Website Check 1	Create basic outlines of the website and upload available information about the project
3/6	System Demo 1	Develop hardware
3/20	System Demo 2	Develop hardware
3/27	Team Update presentation	Present with updated information about the project
4/3	System Demo 3	Update hardware and start developing the mobile application portion of the project
4/7	Website Check 2	Fill the website with updated information
4/10	System Demo 4	Develop mobile application / update hardware
4/17	System Demo 5	Develop mobile application / update hardware
4/24	System Demo 6	Develop mobile application / update hardware
5/3	Final System Demo	Fine tune presentation of the project
5/10	Public Demo	Fine tune presentation of the project
5/15	Final Website Check	Polish the functionality and design of the website

Team Member Responsibilities

Although all members of the team will be collectively working on the hardware component of the project (development and debugging), each member's concentration of responsibilities in other portions of the project would differ as follows.

- Amy Lin
    - Primary: Mobile Application
        - Responsible for the timely development and scheduling of the mobile application portion of the project
    - Secondary: Communication
        - Responsible for communication among team members and between class TA or the professor
- Brian Ho
    - Primary: Hardware Design
        - Responsible of keeping track of development of the hardware portion of the project, and discuss potential problems
    - Secondary: Mobile Application
        - Responsible for debugging and discussion of the mobile application portion of the project
- Helen Cho
    - Primary: Website
        - Responsible for the development of the website and updating it according to schedule
    - Secondary: Documentation
        - Responsible for timely completion of any type of documentation needed for the project, in terms of the public demo and the report

Budget

Name	**Cost**	**Note**
Bluetooth Module HC 05/06	$3.50	
Sparkfun Triple Axis Accelerometer Breakout-MMA8452Q	$9.95	
Piezo Vibration Sensor - Small Horizontal	$2.95	
Raspberry Pi 3	$39.95	Raspberry Pi 2 might be available for free from class
2x DG02S Mini DC Gear Motor	$19.95 as a set	
2x Wheel - 65mm		
Wheel Encoder Kit		
Tilt Sensor - T407	$1.95	
Crystal 16MHz	$0.95	
Total	$76.15	

Risk Management

The possible risks in this projects can be broadly categorized into schedule, design and resource. In this section, we will elucidate the risk for each and develop risk mitigation and disaster recovery measures.

Schedule

- Risk: There are possible risks of development schedule being pushed back, resulting in incomplete development of either the hardware or mobile application component of the project.

- Risk Reduction Measure: Each member will keep track of schedules for each component, according to their concentration of responsibilities.

Design

- Risk: Better options for hardware design may emerge during or after development
- Risk Reduction Measure: make note of the option and fall back to it when the hardware being developed fails to meet requirement

Resources

- Risk: Some of the materials for the hardware may fail and need to be replaced
- Risk Reduction Measure: Since we should have enough budget left over from the allowed amount of money to spend on the project, we can reorder the parts as soon as possible. Other approach would be to purchase a slightly bigger quantity of small parts that may be easier to lose.

Related Work

- Milestone Pod[1] ($24.95)

  Milestone Pod is a lightweight wearable device that provides detailed data about the user's foot movement such as gait, cadence, foot strike, and step count. This kind of data is obtained by strapping the device onto the user's shoes. Although this device offers detailed information about foot movement, it does not take into account the incline of the ground the user is stepping on. The absence of incline information affects calorie count and analysis of the focused area of workout.

- Fitbit Blaze[2] ($149.95)

  As Fitbit's one of the most advanced devices, Fitbit Blaze offers data such as heart rate, multi-sport tracking, and step counts. Much like Milestone Pod, it also does not collect incline information.

- Life Fitness Platinum Club Treadmill with Discover SE Console[3] ($8999.00)

  This internet connected treadmill offers most of the functionalities that our device has as goals. Although its price is on the expensive end ($8999), it offers functionalities such as on-screen entertainment, calorie tracking from speed and incline, machine controlled courses, and internet connectivity that also allows for saving data via cloud.

The wearable devices mentioned above already achieve some of the functionalities we have as part of our goal, but our objective is not to build a device that requires the users to wear or put on. The approach is entirely different, as we aspire to develop a device that would immediately improve a conventional workout machine by simply mounting the device onto it. We envision it to be an instant solution to upgrading a pre-existing workout machine, and an affordable alternative to expensive smart treadmills. More importantly, we want to make the treadmill integrate more closely with external libraries and APIs so that users can gain more actionable insights from their indoor workouts, making it part of the Internet of Things ecosystem.

References

1. "Home." *MilestonePod*. N.p., n.d. Web. 10 Feb. 2017.
2. "Designer." *Fitbit Blaze™ Smart Fitness Watch*. N.p., n.d. Web. 10 Feb. 2017.
3. “Life Fitness Platinum Club Treadmill with Discover SE Tablet COnsole.” *FitnessZone.* N.p., n.d. Web. 10 Feb.