Watchmen/Gesture Watch

Pearak "Derek" Tan, Edward Ryan, Julian Nguyen

Abstract

Our project consists of making a simple wearable device that is completely enclosed and can act as a motion controller. The watch will contain an MSP430F1611 [6] and be able to connect to either an iOS [4] or Android [5] device via Bluetooth [1]. The main functionality of the watch will be gesture-based controls, a feedback system that notifies the user when an action is registered, and a rechargeable battery that allows the watch to last throughout the day. To achieve this, we will require 4 main component subsystems, the MCU/communications module (i.e. MSP430 and Bluetooth adapter), the feedback system (i.e. vibration motor and LED display), an actuator input (pushbutton), and a power module (i.e. battery and charging module). This watch will also be designed in a form factor small enough to fit comfortably on a user's arm during use.

Background

We chose this project because we wanted to create a project that was challenging as well as usable as an everyday wearable device with a unique feature. Many companies have produced their own smartwatches with their own features, and some hobby kits and personal smartwatch projects exist such as one created by S. March [17]. This project created by March had very limited functionality, with no gesture controls and no interface with Android [5] devices. This project also utilizes a DA14683 [3] microcontroller, which for our purposes will not be feasible due to the form factor being too small. Advanced Circuits [2] is unable to create a PCB that accommodates a microcontroller this small while still meeting the limitations of the student special boards. The main differentiating feature of our smartwatch is the ability to interpret complex gesture controls in an intuitive and responsive way. For example, making a sweeping motion in some direction while wearing the watch will cause music tracks to skip or a presentation to advance slides. Other projects incorporate basic accelerometer logging, button input, and a clock. We took inspiration from past smartwatch projects such as the one done by S. March [17] and will use it as a resource going forward. This project will call upon our knowledge of all the Fundamentals Electrical classes [13], mainly in the form of circuits and PCB design. It will also call upon all our knowledge from our previous coding classes, mainly CS 2150 [16] and ECE 3430 [15] for their teachings in C/C++ [14] and lower/embedded systems.

Description of Project

Our project is a wearable that has the aforementioned functionality of having thorough motion/orientation tracking, and the ability to implement software defined gesture controls by processing the sensor data. Our project will work by integrating the various subsystems listed below and soldering all of those subsystems onto a specialized small PCB that we design. The PCB will be powered by a lithium ion battery [12] which is chargeable by wireless inductive charging. The software/firmware will be developed to allow the user to pair their personal iOS or Android device to the watch. The figure below is a preliminary block diagram for all of the subsystems and constituent parts of the watch, including the specific components we plan to use.

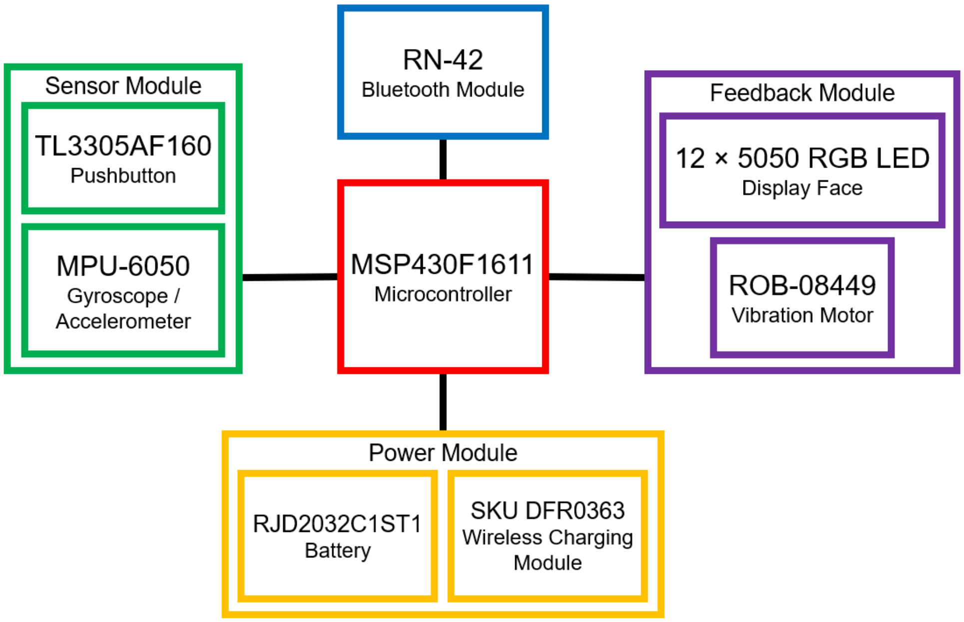


Figure 1.1

One of the biggest problems that we face is the power module system as well as the ability to fit all of the components into a wearable form factor. We expect to create a functional wearable with the ability to connect to a smartphone via Bluetooth [18]. It should have enough battery power to last throughout a regular day's use. We do not anticipate using any external materials beyond those provided by the ECE lab. The project's main subsystems are laid out in Figure 1.1 above. Those 4 systems include the sensors (green), processor and communications (red/blue), power (yellow), and feedback module (purple).

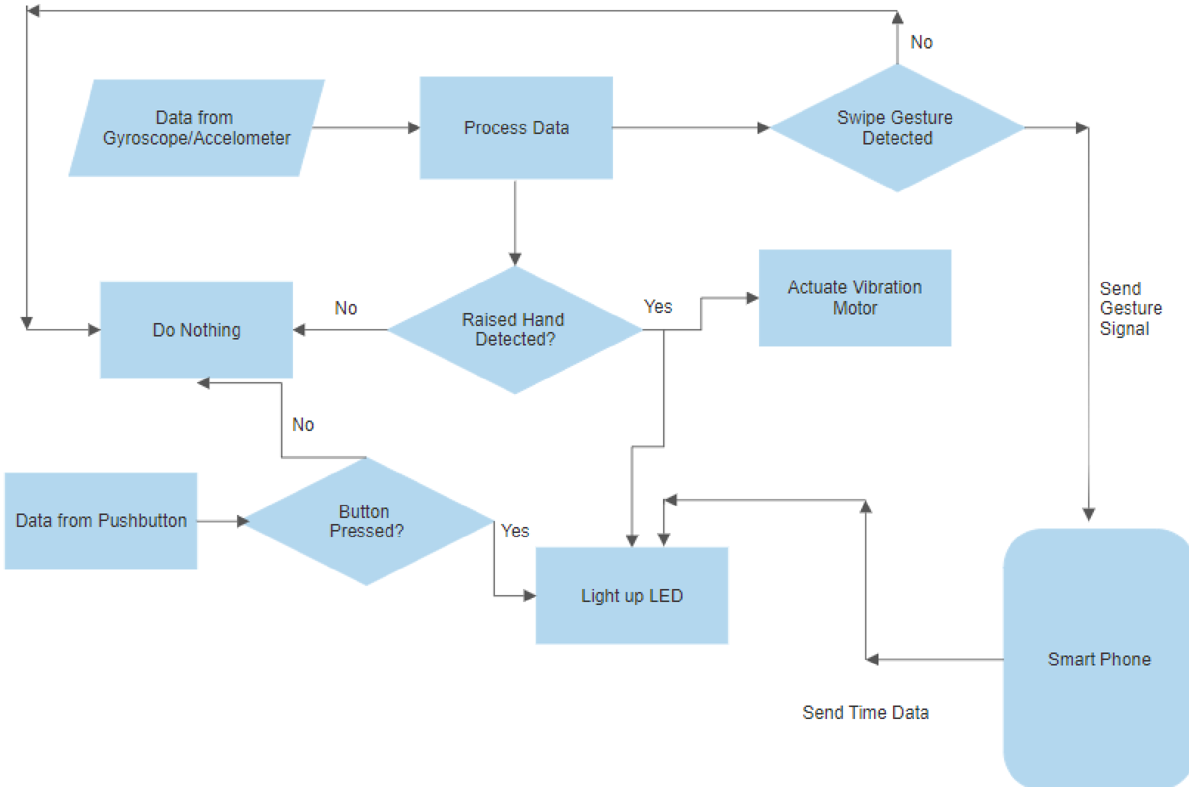


Figure 1.2

Figure 1.2 shows the planned functional behavior of the watch. Motion data will be taken in from the gyroscope/accelerometer whenever a gesture/hand raise is detected. The vibration motor will actuate, and the LED ring display will light up to show the time on the watch. An alternative feedback from the pushbutton may also be used to light up the LED display. In the event that a swipe gesture is detected, data will be sent to the smartphone or other connected device and perform a corresponding command. An example of a successful gesture control command that we want to implement is the advancement of a presentation slide when a swipe is detected. Another example would be to skip a song that is currently playing on the paired device.

The software requirements anticipated will be Code Composer [11] (used to upload code to the MSP430), Multisim [10] (used for modeling and simulation), Ultiboard [1] (used to design PCB layout), the C programming language [14] (language used in Code Composer), Cura [19] (used to slice the 3D model into a printable format), and Inventor [20] (used to 3D model the chassis).

External Considerations

Manufacturing, Parts Availability, and Cost

Due to the wearable device form factor, assembly will be a primary challenge. However, for the same reason, the demand for excessive or expensive components is limited to a manageable scale. The design will mainly consist of common electronic components that are readily available from DigiKey [8] or other popular hardware suppliers. Non-electronic components are equally accessible, with the watch housing being 3D-printable and watch bands being a standardized and interchangeable product. The restricted scale of components, PLA filament [9], and other parts estimates a total cost no larger than $150. The preliminary bill of materials is listed below under the deliverables.

Energy and Sustainability

The primary environmental concern of our project is the power module of the device, as lithium ion batteries used by mobile/wearable devices have a significant energy and pollutant footprint. We will need to ensure that our design efficiently manages power, such that power is effectively drawn when charging and the watch is able to fully function for a significant time before draining the battery.

Health and Safety

The power module of the watch also poses various safety concerns. The charger contact and battery are potential threats of exposure of electrical faults to the wearer of the watch. We will have to make sure that the device is fully enclosed so that human skin will never come into contact with exposed circuitry. In addition to ensuring that the watch's power management is efficient, we will also need to ensure that it has no major risk of injuring the user. A battery protection circuit will be necessary to keep the battery from overcharging and thus overheating and harming the wearer. The feedback system parts may also pose a minor health risk if vibration or LED brightness intensity are not regulated properly. We must also make sure that the plastic filament used to build the watch housing poses no health risk.

Ethical Issues

There could be some concerns in regards to the fact that the watch could potentially be used to cheat on an exam. Since the watch has Bluetooth capabilities, it is possible for a user to program the watch's visual and tactile feedback system to aid in discreet communication.

Standards

The appropriate safety methods will be taken into account when soldering the PCB. Appropriate safety wear will be used. The watch will use Bluetooth 2.1 [18] to connect to a smartphone. It is commonly used for wireless headphones and other audio hardware, as well as wireless keyboards, mice, and game controllers. Bluetooth is also used for communication between various smart home and Internet of Things (IoT) devices [18]. The lithium ion battery [12] that is used to provide power to the watch is also a standard that is used to ensure power retention and high energy density. Standard PLA [9] plastic is used to 3D print the chassis due to its affordability and finish quality, the Windows 10 operating system [7] will be used to run Code Composer [11] as the primary development environment for our MSP430 [6] chip.

Deliverables

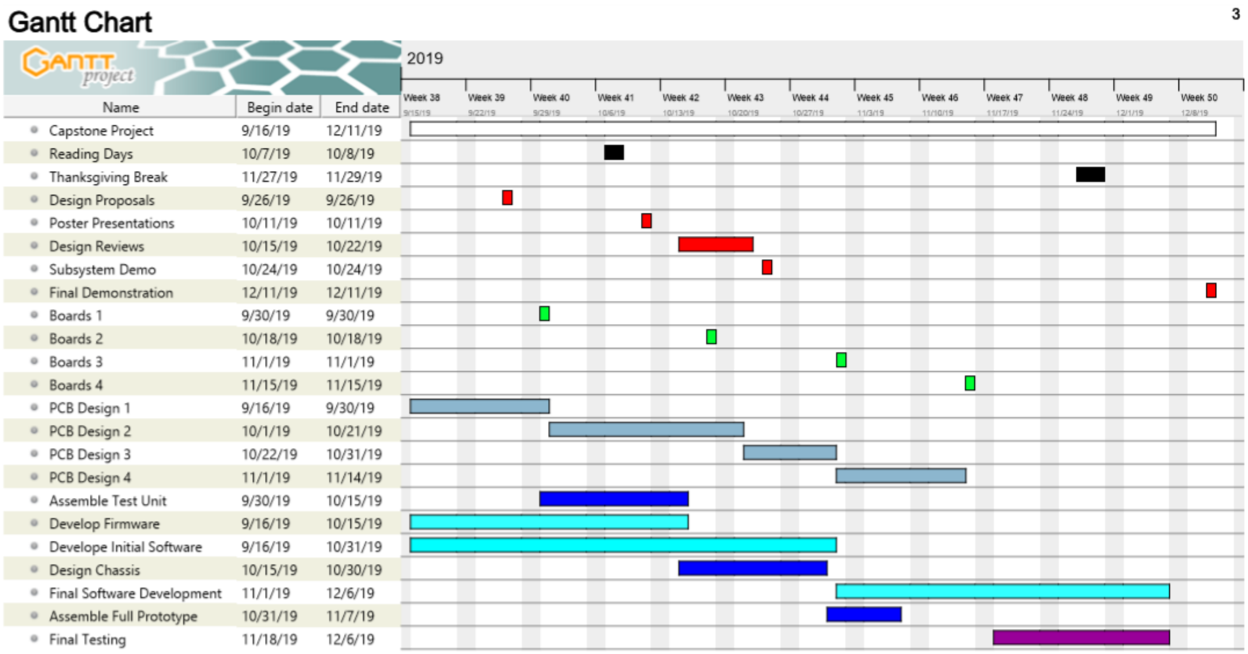
Our project's deliverables will consist of 1) the assembled physical watch components themselves 2) the printed circuit board that fits inside the watch and connects all physical components together 3) the watch firmware which allows interaction with/between the connected components. The physical watch will be put together by a 3D printed chassis. The chassis will be designed in Inventor [20] and 3D sliced in Cura [19]. The circuit and PCB will be designed in

Multisim [10] and Ultiboard [1] and sent out to Advanced Circuits [2] to manufacture. Code Composer [11] will allow us to load C [14] coded programs to the MSP430 [6] launchpad. The total cost expected to produce a single watch is $138.50. Although the project might end up costing much more after testing and trial and error of various components. A list of our preliminary bill of materials is found in Figure 1.3.

Item	Quantity	Price
MSP430F1611	1	$11
MSP430 LaunchPad	1	$25
Gyroscope/Accelerometer	1	$6
Vibration Motor	1	$3
12 x 5050 RGB LED Ring	1	$8
Lithium Ion Battery	1	$10
Wireless Charging Module 5V/300mA	1	$7
Resistors/Capacitors/etc.	-	$15
Pushbutton	1	$0.5
PLA Plastic	1	$20
PCB Manufacture	1	$33
Total Price		$138.5

Figure 1.3

Timeline



The chart above shows our rough outline of our project schedule, deadlines and holidays. Much of this will depend on when exactly we acquire the parts we need. Assembling a test unit with the MSP430 processor, motion sensor, and battery wired together without a chassis will be done early. However, it does depend on when we receive all of the necessary components. Assemblies with future PCB designs will, of course, depend on when the PCBs arrive. Coding software for final features will depend on when the software which handles the basics is completely functional. Although, we will likely have people coding for multiple pieces of software simultaneously. If our first (or more likely second) PCB works perfectly, and we add no additional features, then we will no longer need to work on PCB design. If our initial 3D printed chassis doesn't fit right, then we may need to add additional chassis design time. Basically, we plan to try to get things done as quickly as possible, because we never know what obstacles we might run into along the way, or how much time they may take.

Julian will be the primary person working on the Multisim schematic/simulations and Ultiboard/PCB layout. He will use his Electrical Engineering expertise to analyze the proper parts and do the necessary testing to ensure that all the electrical components work correctly. Julian will primarily focus on anything pertaining to electronics and hardware. He will also be the secondary person working on the 3D modeling.

Derek will be the primary person working on developing the chassis for the watch. He will use his previous experience in 3D modeling and printing and be responsible for ensuring that all components of the watch are properly enclosed. He will serve as the secondary programmer in the project.

Edward will be the primary programmer in the project. He will use his prior experience in C and Code Composer to develop proper firmware and ensure a smooth gateway between the hardware and software components. Edward will also serve as the secondary hardware debugger/tester.

Expectations

For the subsystem functionality demo, we plan to have our power system functional, with an induction-rechargeable lithium-ion battery. We also plan to be able to demonstrate that our accelerometer and gyroscope sensor can be powered (hopefully by the battery, but this could be tested with a virtual bench power supply), detect motion, and output sensor data.

The following lists the subsystems (labeled 1-4) and components (labeled a-d) that we plan to implement in the final version.

1.) Communication Subsystem
    a.) Bluetooth Connectivity
2.) Sensor Subsystem
    a.) Accelerometer Sensing
    b.) Gyroscope Sensing
    c.) Gesture Interpretation Controls
    d.) Pushbutton Control
3.) Feedback/Actuator Subsystem
    a.) Vibration Motor Feedback
    b.) LED Array Feedback
    c.) LED Clock/Timekeeping
4.) Power Module Subsystem
    a.) Battery
    b.) Charging Dock

For our final project demo, we suggest the following grading scale:

- Watch unit exists and fits into a chassis, battery charges, and watch powers on. At least 6 components in the above list can be demonstrated to be at least partially functional: A
- Watch unit exists and fits into a chassis, battery charges, and watch powers on. At least 5 components in the above list can be demonstrated to be at least partially functional: B
- Watch unit exists and fits into a chassis, and watch powers on. At least 4 components in the above list can be demonstrated to be at least partially functional: C
- Watch unit powers on. At least 3 components in the above list can be demonstrated to be at least partially functional: D

References

[1] “Ultiboard - National Instruments.” [Online]. Available: http://www.ni.com/en-us/shop/select/ultiboard. [Accessed: 24-Sep-2019].

[2] “Printed Circuit Board Manufacturer & PCB Assembly | Advanced Circuits.” [Online]. Available: https://www.4pcb.com/. [Accessed: 24-Sep-2019].

[3] james.johnson, “SmartBondTM DA14682 and DA14683,” Dialog Semiconductor, 28-May-2018. [Online]. Available: https://www.dialog-semiconductor.com/products/connectivity/bluetooth-low-energy/smartbond-da14682-and-da14683. [Accessed: 24-Sep-2019].

[4] “iOS 13,” Apple. [Online]. Available: https://www.apple.com/ios/ios-13/. [Accessed: 24-Sep-2019].

[5] “Android | The platform pushing what’s possible.” [Online]. Available: https://www.android.com/. [Accessed: 24-Sep-2019].

[6] “MSP430 Ultra-Low-Power MCUs | Overview | Microcontrollers (MCU) | TI.com.” [Online]. Available: http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html. [Accessed: 24-Sep-2019].

[7] “Windows Documentation | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/. [Accessed: 24-Sep-2019].

[8] “DigiKey Electronics - Electronic Components Distributor.” [Online]. Available: https://www.digikey.com/. [Accessed: 24-Sep-2019].

[9] “PLA - EduTech Wiki.” [Online]. Available: http://edutechwiki.unige.ch/en/PLA. [Accessed: 24-Sep-2019].

[10] “Multisim Live Online Circuit Simulator,” NI Multisim Live. [Online]. Available: https://www.multisim.com/. [Accessed: 24-Sep-2019].

[11] “CCSTUDIO Code Composer Studio (CCS) Integrated Development Environment (IDE) | TI.com.” [Online]. Available: http://www.ti.com/tool/CCSTUDIO. [Accessed: 24-Sep-2019].

[12] “IEEE 1625-2004 - IEEE Standard for Rechargeable Batteries for Portable Computing.” [Online]. Available: https://standards.ieee.org/standard/1625-2004.html. [Accessed: 24-Sep-2019].

[13] “UVa Course Catalog - Complete Catalog for the Electrical and Computer Engineering Department (Unofficial, Lou’s List).” [Online]. Available: https://rabi.phys.virginia.edu/mySIS/CC2/ECE.html. [Accessed: 24-Sep-2019].

[14] “C Language Introduction,” GeeksforGeeks, 12-Feb-2014. [Online]. Available: https://www.geeksforgeeks.org/c-language-set-1-introduction/. [Accessed: 24-Sep-2019].

[15] “ECE 3430 - Introduction to Embedded Computer Systems at the University of Virginia | Coursicle UVA.” [Online]. Available: https://www.coursicle.com/virginia/courses/ECE/3430/. [Accessed: 24-Sep-2019].

[16] “Program and Data Representation: CS 2150 Specific Content.” [Online]. Available: https://aaronbloomfield.github.io/pdr/uva/index.html. [Accessed: 24-Sep-2019].

[17] S-March, S-March/smarchWatch_PUBLIC. 2019.

[18] “Specification of the Bluetooth System”. [Online]. Available: http://grouper.ieee.org/groups/802/15/Bluetooth/core_10_b.pdf. [Accessed: 24-Sep-2019].

[19] “Ultimaker Cura: Powerful, easy-to-use 3D printing software.” [Online]. Available: https://ultimaker.com/software/ultimaker-cura. [Accessed: 26-Sep-2019].

[20] “Inventor | Mechanical Design And 3D CAD Software | Autodesk.” [Online]. Available: https://www.autodesk.com/products/inventor/overview. [Accessed: 26-Sep-2019].