Watchmen/Gesture Watch

Pearak "Derek" Tan, Julian Nguyen

Abstract

We propose adding and improving features on our current gesture watch completed in the Fall of 2019. Currently, our project consists of a simple wearable device that is completely enclosed and can act as a motion controller. The watch contains an MSP430F1611 [6] and be able to connect to either a device via Bluetooth [1]. The main functionality of the watch are 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 have 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 is also designed in a form factor small enough to fit comfortably on a user's arm during use. We propose an independent research in the Spring semester of 2020 to solve some of the shortcomings that we encountered but did not have adequate time to resolve. The four things that we wish to improve upon will fall within 4 categories. These categories are quality of life, new features, battery efficiency changes, and new software modes/functionality.

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 called upon our knowledge of all the Fundamentals Electrical classes [13], mainly in the form of circuits and PCB design. It also called upon 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. We anticipate that the same software will be used if this project were to be continued into the Spring semester.

The project that we undertook in the Fall of 2019 was considered a massive success from our standpoint. We were able to meet all of our deliverables and were able to code all the functionality that was proposed. Although we were able to do all this, some of the functionality was rushed and could use improvement on the software and hardware side. We also propose to make many quality of life changes to help with debugging and ease of use as well as many battery optimization features.

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 works by integrating the various subsystems listed below and soldering all of those subsystems onto a specialized small PCB that we design. The PCB is 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 a computer. The figure below is a block diagram for all of the subsystems and constituent parts of the watch, including the specific components we plan to use.

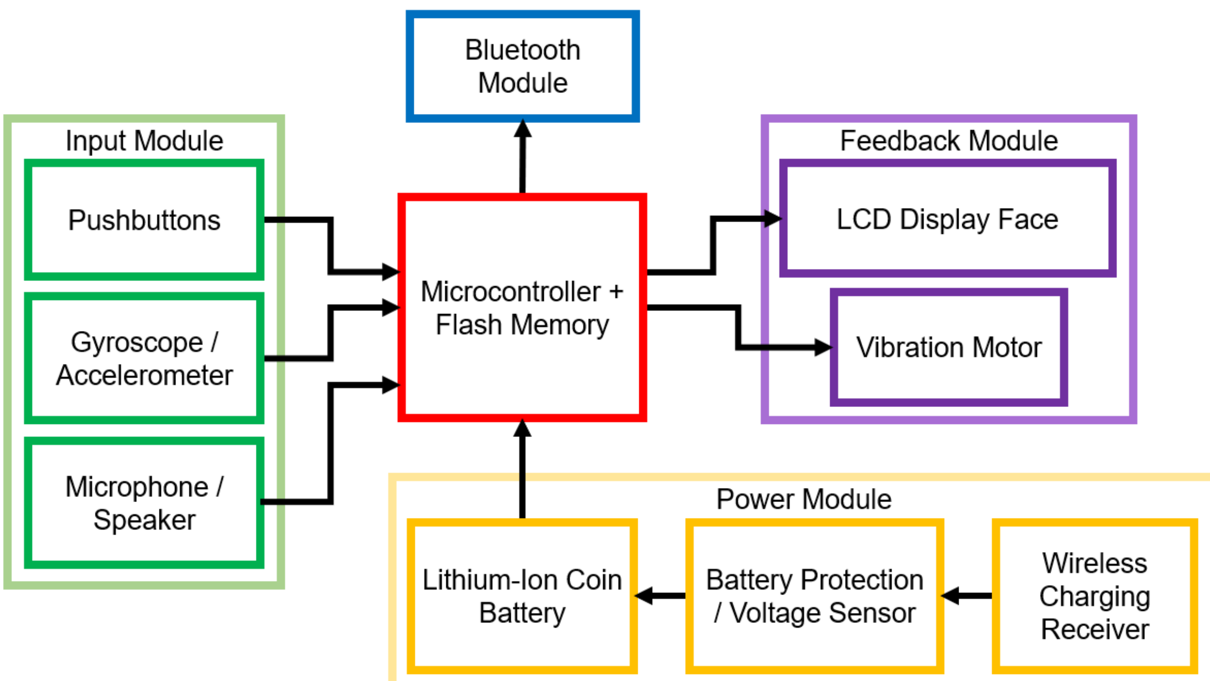

Figure 1: Subsystem Diagram

Quality of Life Changes

The changes that we plan to target first are the quality of life changes that will make debugging easier. The first quality of life change that we propose is adding a small magnet to the bottom chassis of the watch as well as one on the charger chassis. This will provide a more reliable and consistent charging mechanism as it will make sure that the charging transceiver and receiver are in proper contact with each other. Another change would be to have a better way to mount the pushbutton. Currently, the pushbutton is connected by a wire and that wire can sometimes make it very difficult to enclose the top chassis. We believe that adding a direct mounting mechanism to the side of the PCB that eliminates the need for wires would ease the enclosing process. The same could be said with the LED display. A mounting mechanism attaching to the top of the PCB would also greatly ease the process. Another quality of life change would be to have the VCC shunt on a helper board. Currently the VCC shunt to switch between tool and target is on the PCB which is enclosed within a chassis. Having a helper board for the shunt would make it so that the top chassis would have to be removed every time you wanted to change the VCC source and greatly ease debugging.

Entirely New Hardware Features

There are several things that we wished to include in our original proposal but decided to omit in order to scale down the project. An example of this is having a microphone and speaker. Having this would allow for the potential implementation of voice commands. An added LCD screen and flash memory would also make the device possible of fully supporting Android Wear OS [22], an operating system used for wearable devices. Another thing that we wish to explore doing is adding a second button. This second button could be used for extra software functionality and could be combined with gestures to perform an action. It could also be used to turn the watch completely off. One thing that we also want to explore is possibly fabricating a 4 layer PCB. This would make the routing from the Ultiboard side cleaner, but it would also mean that the processing times for the PCB would take significantly longer.

Battery Efficiency Changes

There are many battery efficiency changes that we wish to implement. Our current watch only lasts about 3.5 hours on timekeeping mode and 25 minutes on the gesture mode. Because of this, we believe that the Bluetooth module was a significant factor for the battery drain. We had made some changes that made the watch to only connect to Bluetooth when it was in gesture mode but the battery still proved to be very weak. Another reason why our battery was so poor was because we omitted a reset circuit for the Bluetooth module. This meant that there was no way for us to completely turn off the Bluetooth module. Another thing that we could do is use a newer and more efficient module. The RN4871 is an updated model of the RN42 and should provide more efficiency for the battery. It also has a smaller footprint and would make the router a lot simpler. Another thing that we omitted from the Bluetooth subsystem was the voltage regulator. Initially we overlooked this because of the fact that our lithium ion was rated at 3.7 V which was the same as the maximum operating voltage of the module. This was an erroneous assumption as the lithium ion battery was able to charge all the way up to 4.2 V. This led for the Bluetooth module to be completely dysfunctional at voltages over 3.7 V and would be easily fixed by including a voltage regulator.

We also wish to add a voltage sensor that is able to give the watch user visual or audio feedback when the battery is full or low. This could be added to a GPIO pin on the MSP430. As stated in the previous section, we wish to add a second button that puts the watch into a completely off state. The off state would be used to save battery when the watch is not in use. Another thing that could possibly improve battery life would be to use a more updated microprocessor. Our MSP430F1611 microprocessor came out in 2002 and we believe that using a newer MSP430 chip might result in better battery life, as we had to keep our current microprocessor overclocked to 8 MHz in order to power the LED display.

One thing that we noticed during testing was that our battery protection circuit successfully worked for overvoltage but it did not for undervoltage as our battery drained all the way to 0.5 V. This would be fixed by changing to a battery with a built in protection circuit. We propose using the 3.7V 150 mAh 302030 [21] lithium ion battery. Our current battery is rated for 85 mAh. This would mean changing our chassis to fit the new battery although we believe that it will greatly benefit the battery efficiency. The last thing we wish to possibly add as a way to charge via JTAG. This would help to eliminate the need of switching the VCC shunt.

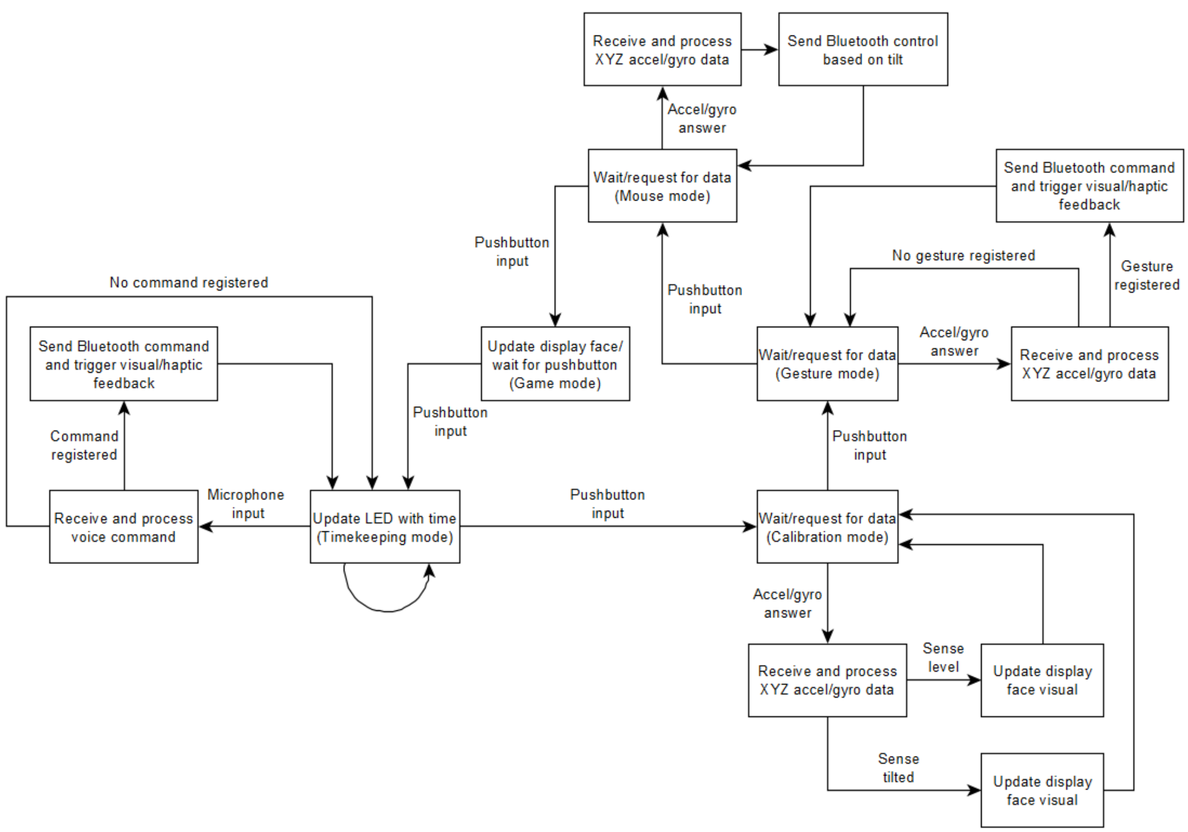

Figure 2: High Level Software Functionality

Figure 2 shows the state diagram for the watch's software behavior, including pushbutton toggle between the various modes and the expected input results for those modes. The initial timekeeping software state is reached by pressing the power pushbutton, which toggles the watch to and from the universal powered-off state (not pictured).

New Modes/ Software Functionality

There are also many new software features that we wish to implement. The one thing that we are really enthusiastic to try out is having a mode where the watch could be used as a mouse for a computer. This could work with the fact that we plan to add a second button and the left and right click could be implemented with both the buttons. Another mode that we wish to implement is a simple reaction time game with the Bluetooth module, pushbutton, and LED display. This would consist of pressing the pushbutton when the LED turns green. The watch would then send a message via Bluetooth to a computer informing the user about the time it took to react. This is a very simple implementation that could be expanded on greatly. With the addition of the speaker and microphone, voice commands could be implemented from the software side. We wish to explore using this with Windows built in voice command application Cortana. Other new things we wish to include are integrated driver support (eliminate AutoIt driver), Android support, and the ability to sync time automatically via the bluetooth module. The time sync would request the time every 10 minutes from the connected computer and update accordingly.

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 of $364. 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. This would also further be a problem if our proposed microphone and speaker were to be implemented.

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 $364. 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. This lists a bill of material if all the features proposed were implemented (see Expectations sections for comprehensive list).

Item	**Quantity**	**Total Price**
MSP430	1	$11
Speaker/Microphone	1	$10
Flash Memory	1	$10

LCD Display	1	$30
RN4871	1	$7
Vibration Motor	1	$3
12 x 5050 RGB LED Ring	1	$8
Lithium-ion Battery	1	$10
Wireless Charging Module 5V/300mA	1	$7
Chip Resistors/Capacitors/Transistors/etc.	-	$15
Pushbutton	2	$1
PLA Plastic	1	$20
PCB Manufacture from Advanced Circuits	1	$33
Watch Band	1	$13
Watch Pin	1	$6
3W Mounting	-	$60 * 3 Hours = $180
Total Price		$364

Figure 3: Preliminary Cost of Materials

Timeline

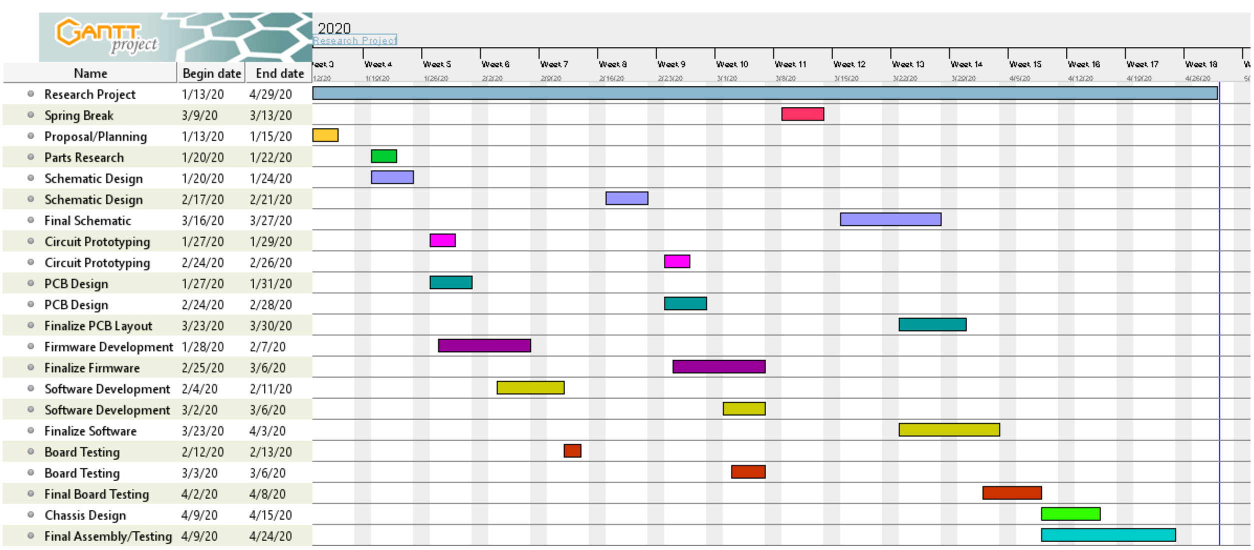

Figure 4: Preliminary Gantt Chart

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 previously tested MSP430 processor, motion sensor, and battery, etc. and the new features such as a possible LCD screen 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. The code will depend entirely on if the hardware is working properly. Although, we will likely have people coding for multiple pieces of software simultaneously.

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 and coding.

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 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.

Expectations

The following lists categories of improvement (labeled 1-4) and components (labeled a-j) that we plan to implement in the final version.

1.) **Quality of Life**
- a.) Magnet for Watch Chassis and Charger Chassis
- b.) LED display mount
- c.) Pushbutton side mount
- d.) Magnet for charger
- e.) Shunt on helper board
- f.) Change chassis for a more secure fit (screw?)

2.) **Entirely New Hardware Features**
- a.) LCD Screen
- b.) Flash Memory
- c.) Second button (Power switch)
- d.) 4 layer PCB
- e.) Speaker
- f.) Microphone

3.) **Battery Efficiency Changes**
- a.) Smaller BLE (rn4871)
- b.) Bluetooth reset / Bluetooth connection pin
- c.) Voltage sensor

d.) New microprocessor
e.) Battery protection undervolting fix
f.) Voltage regulator for Bluetooth
g.) Bigger battery
h.) Completely off state
i.) Charge via JTAG

4.) **New Modes/ Software Functionality**

a.) Reaction time game with Bluetooth
b.) Mouse mode
c.) Integrated driver support (eliminate AutoIt driver)
d.) Android support
e.) Sync time automatically
f.) Voice commands

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

- At least 4 improvements to quality of life category, 4 improvements to battery efficiency category, 6 improvements to the battery, and 3 software functionality additions: A
- At least 3 improvements to quality of category, 3 improvements to battery efficiency category, 5 improvements to the battery, and 2 software functionality additions: B
- At least 2 improvements to quality of category, 2 improvements to battery efficiency category, 4 improvements to the battery, and 1 software functionality additions: C
- At least 2 improvements to quality of category, 2 improvements to battery efficiency category, 2 improvements to the battery, and 1 software functionality additions: 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].

[21] “3.7V lithium battery 302030 032030 160mah MP3 MP4 GPS Bluetooth cell battery toys-in Replacement Batteries from Consumer Electronics on AliExpress,” aliexpress.com. [Online].

Available:
//www.aliexpress.com/item/32346829163.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=. [Accessed: 12-Jan-2020].

[22] “Wear OS by Google Smartwatches,” Wear OS. [Online]. Available: https://wearos.google.com/. [Accessed: 12-Jan-2020].