To design a smartwatch and evaluate it against commercial alternatives.

Jake M. Drew

Abingdon School

15/1/18

8117 Words

Contents

What is a smartwatch?

A SmartWatch is a wearable computing device that resembles a wristwatch or other timekeeping device, many smartwatches are able to extend the capabilities of the wearer's smartphone to the watch.[1]

What Functions should the smartwatch have?

In order to know what features this smartwatch should have I first need to understand what the commercial alternatives have. Fundamentally all smartwatches must have the ability to tell the time, as well as the date, they all must have a minimum of 1 day battery life such that they can be worn all day without the need to recharge. However, the majority of smartwatches have bluetooth which enables them to connect to the wearers smartphone and to display data retrieved from the phone at a simple glance, such as the local weather, notifications and text messages. Some smartwatches include the ability to track steps and the wearers movement.[2]

What Functions Should this watch have?

It is necessary for this watch to have some means of communicating to the outside world, so it can receive data from either a phone or the internet and display such data to the wearer. As such, I must have either, WiFi or Bluetooth. When considering which to choose, I must take into consideration other factors, such as power consumption, size and cost. In terms of cost, both wifi and bluetooth are relatively cheap with chips under £10, they are relatively small with no bluetooth or wifi devices over a few centimeters in length and several millimeters in height. But in terms of power consumption it is clear the bluetooth is the winner as wifi can consume up to 170ma[3] while in use, even though it has a low powered state the powered on state is too high and it would drain a battery very quickly. The bluetooth has a power consumption of 17ma[4] means that a bluetooth module is 10% that of the wifi. This watch should also have the ability to get data such as the surrounding temperature and air pressure without the need for internet connection or bluetooth connection.

My aims for this project

My intentions for this project is to design a smartwatch which is able to extend the capabilities of my smartphone to my wrist. I want to keep the cost of producing this device below £139.37[5] which is the average cost of a smartwatch in 2014[6]. I want this device if possible to be hand assembled and soldered, such that it is possible for after this project to build a real prototype. I think the watch must last at least a week on a full charge as recharging a watch everyday is inconvenient.

---

[1] IoT Agenda. (n.d.). What is smartwatch? - Definition from WhatIs.com. [online] Available at: http://internetofthingsagenda.techtarget.com/definition/smartwatch [Accessed 6 Oct. 2017].
[2] Silbert, S. (2017). Smartwatches: A Beginner's Guide. [online] Lifewire. Available at: https://www.lifewire.com/an-introduction-to-smart-watches-3441381 [Accessed 9 Oct. 2017].
[3] Bbs.espressif.com. (2014). ESP8266 Power Consumption - ESP8266 Developer Zone. [online] Available at: http://bbs.espressif.com/viewtopic.php?t=133 [Accessed 9 Oct. 2017].
[4] BLE112 Bluetooth Smart Module. (n.d.). 1st ed. [ebook] Silicon Labs. Available at: https://www.silabs.com/documents/login/presentations/BLE112-presentation-v2.pdf [Accessed 10 Oct. 2017].
[5] This price is based of the average cost being $189 which was converted to Pound Sterling on 5/1/18
[6] Green, T. (n.d.). 25 stats on smartwatches, fitness bands, eyewear and smart clothing. [online] Hot Topics. Available at: https://www.hottopics.ht/15985/25-stats-on-smartwatches-fitness-bands-eyewear-wearables-and-smart-clothing/ [Accessed 11 Oct. 2017].

Why Use Arduino

Arduino in their own words “is an open-source electronics platform based on easy-to-use hardware and software”[7]. Arduino has a wide range of hardware projects which vary in cost from under £5 to over £50[8], along with the hardware they build software and it is this software which I believe will be most useful for this project. They have built the Arduino IDE which “makes it easy to write code and upload it to the board” [9], the board in this instance refers to one of their projects which most importantly runs their bootloader. “A bootloader is a program that runs in the microcontroller to be programmed. It receives new program information externally via some communication means and writes that information to the program memory of the processor.”[10] The Arduino IDE is able to program hardware which runs the Arduino bootloader. This limits my board choice to the Atmel AVR family, of which are 8-bit microcontrollers.

The Arduino IDE runs on all major operating systems and the programming language is based off C/C++, this means that the Arduino IDE has a large base of libraries, which are importable code bases which reduce the stress of programming.

[7]Arduino.cc. (n.d.). Arduino - Introduction. [online] Available at: https://www.arduino.cc/en/Guide/Introduction [Accessed 14 Oct. 2017].
[8]Store.arduino.cc. (n.d.). Arduino. [online] Available at: https://store.arduino.cc/ [Accessed 12 Oct. 2017].
[9]Arduino.cc. (n.d.). Arduino - Software. [online] Available at: https://www.arduino.cc/en/Main/Software [Accessed 15 Oct. 2017].
[10]Lathrop, O. (2012). What is a boot loader, and how would I develop one?. [online] Electronics.stackexchange.com. Available at: https://electronics.stackexchange.com/questions/27486/what-is-a-boot-loader-and-how-would-i-develop-one [Accessed 15 Oct. 2017].

Main Processing Unit

Understanding the technical aspects to the main processing unit

In order to decide on what integrated circuit should be used for the main processing unit (microcontroller), I need to first know what technical specification it should meet. As then I will be able to choose a main processing unit which satisfies all the requirements. Firstly, I will discuss the ways for different parts of the watch to communication.

Communication Protocols

There are three main protocols of connecting integrated circuits to other integrated circuits, the first is Universal asynchronous receiver-transmitter or UART, which is also known as serial communication, it allows the integrated circuit to pass data to the other device via two pins a receive (RX/RXD) and transmit (TX/TXD). Both devices must be configured to read at the same speed for this communication to work, UART allows for devices to be programmed via a USB cable and for connecting integrated circuits to Bluetooth modules, RFID chips and GPS modules. The main benefits of this type of communication is that it only requires two wired connections between devices. It also doesn't require for there to be a clock connection as the integrated circuits are programmed prior to be in sync. However it doesn’t fully support more than two devices being connected as the two pins as there is no mechanism to implement a chip selection.

Another method of connecting is through a protocol called SPI or Serial Peripheral Interface which was introduced by motorola, it is designed as a master-slave protocol. Serial Peripheral Interface allows for the communication of multiple devices to the microcontroller as the protocol has a Slave Select of SS pin, this tells the slaves whether or not they should listen to the data being transmitted, this also has two data pins, MOSI (Master Out Slave In) and MISO (Master In Slave Out), the MOSI is the wire in which the microcontroller transmitted data to the slaves, and the MISO is the wire where the slaves transmit data to the master. The SPI protocol is synchronous and this requires a common Serial Clock Pin for the master to tell the slaves on every rising edge to read the data, this allows for the devices to be in sync with each other. Figure 14.38[11] shows a common SPI configuration, where he clock and data lines are shared. As mentioned earlier you can select the chip with the chip select pin which the host (master) can select with the slave select pins.

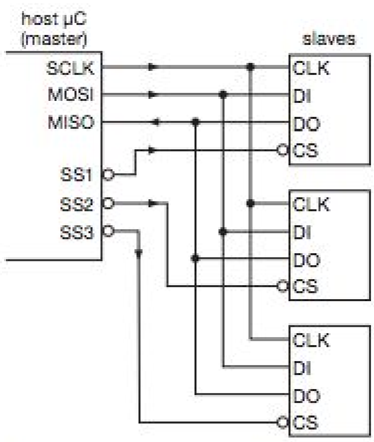


Figure 14.38. The common bused SPI configuration: clock and data lines are shared, with individual SS' (slave-select) lines asserting the chip-select inputs of the corresponding slaves.

A more widely used alternative is I^2C peripheral interface or Inter-integrated Circuit which is a serial interface bus created by Philips, it has several advantages over SPI in that it uses two wires (I^2C on Atmel chips is referred to TWI or Two-wire Interface) rather than 4. Each device has an address in which the address is sent first in the data packet meaning there is no need for a chip select connection as selecting the chip is done with the addressing, this gives a lot more flexibility over SPI in that it can be altered in the software. I^2C is a method of communicating between multiple slaves and masters, upto 112 devices can be connected via this method. It works with a common clock like SPI which allows for it to be synchronous and a data connection, however each device has a unique 7-bit address which allows for 128 addresses but 16 addresses are reserved by the protocol, so only 112 addresses are able to be used by the devices. The I^2C

[11] Horowitz, P. and Hill, W. (2017). The art of electronics. 3rd ed. Cambridge: Cambridge University Press, p.1033.

protocol enables many devices to be connected to a single microcontroller and for them to be selected as you can call devices usings there 7 bit address via software which gives a lot more flexibility for future programming changes. Figure 14.40[12] shows the I^2C data and clock pins with the 7-bit slave address, from A6 (Most Significant Bit) to A0 (Least Significant Bit).

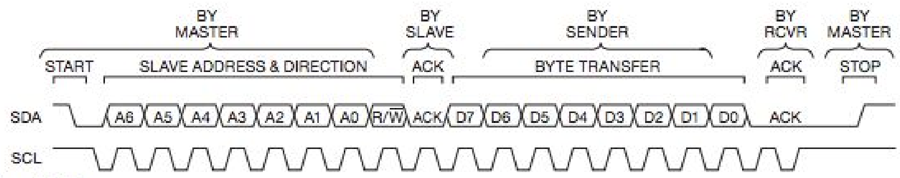


Figure 14.40.

Device Memory

There are several different kinds of memory for the Atmel AVR chips, these are: Flash Memory, SRAM, EEPROM.[13]

Flash memory is the memory where the bootloader goes, in Arduino's case it takes up 2 kilo Bytes of data. Flash memory is non-volatile meaning the data stays there when there is no power, hence why the flash memory is used to store the program memory. Flash can't be modified by the program as it is the program and they can edit themselves

SRAM or Static Random Access Memory, this memory can be written or read from the program stored in the Flash Memory. There are several different purposes of SRAM, firstly there is the static data which is reserved for all the global variables such as the millis() function which counts the number of milliseconds since the program started running. The Heap is for dynamically allocated data. Finally, the stack is were all the local variables and functions are. SRAM is volatile meaning that when the chip stops receiving power this data is erased.

EEPROM[14] or Electrically Erasable Programmable Read-Only Memory is a non-volatiles memory which can be erased and written to from the application. It is like SRAM in that is can be modified from the program but it is non-volatile. EEPROM has to be erased and rewritten every time a value is needed to be changed.

Interrupts

Interrupts[15] are a way in which you can tell the device that some actions is needed to be taken on an important event[16]. In the case of the smartwatch, when a button is pressed there will be an interrupt attached to this and the Atmel AVR microcontroller will have a section of code dedicated for that interrupt, so when that interrupt runs, the program runs that code and the action is taken. A good example is on phones they have a home button when you press that button regardless of what else the phone is doing, it will take you to the home screen.

Comparison of the AVR Microcontrollers

The Atmel AVR microcontroller which I intend to use for this project must be easily programmable, have either a low power consumption or have a low power state - such that the microcontroller can be put into a low power mode (sleep mode) when the device is not in use. There are three main Atmel AVR microcontrollers of which I could use: ATTiny85, ATMega328, ATMega1284.

---

[12] Horowitz, P. and Hill, W. (2017). The art of electronics. 3rd ed. Cambridge: Cambridge University Press, p.1034.
[13] Arduino.cc. (n.d.). Arduino - Memory. [online] Available at: https://www.arduino.cc/en/Tutorial/Memory [Accessed 17 Oct. 2017].
[14] Rouse, M. (2010). What is EEPROM (electrically erasable programmable read-only memory)? - Definition from WhatIs.com. [online] WhatIs.com. Available at: http://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory.
[15] Horowitz, P. and Hill, W. (2017). The art of electronics. 3rd ed. Cambridge: Cambridge University Press, p.1005.
[16] Reilly, N. (2015). Using Interrupts on Arduino. [online] Allaboutcircuits.com. Available at: https://www.allaboutcircuits.com/technical-articles/using-interrupts-on-arduino/ [Accessed 25 Oct. 2017].

The ATTiny85[17] is the smallest of these AVR microcontrollers, it has 8 pins, 8kB of flash memory, 512 bytes of SRAM, 512 of EEPROM. It can interface with components via SPI and I^2C, but they have overlapping pins so you can either use SPI or I2C not both. The ATTIny85 lacks UART which would enable it to communicate with Bluetooth devices.

The ATMega328[18] is most notably used on the Arduino Uno which is Arduino's flagship product. The ATMega328 comes with 32kB of flash, 2kB of SRAM, 1kB of EEPROM. It has both SPI and I^2C on separate pins, and has UART.

The Atmega1284[19] is the larger model which has 128kB of flash and 16kB of SRAM and 4kB of EEPROM, where the ATmega1284 puts itself ahead is with 2 UART connections, 3 SPI and I^2C.

Out of the three Atmel AVR microcontrollers, the ATTiny85 is simply too small for this project, it doesn't have enough flash memory, both the ATMega328 and ATMega1284 are suitable in terms of what functionalities they possess. The ATMega328 has a lower power consumption than the ATMega1284. As shown in Figure 33-2 on the ATMega328 Datasheet (p. 379)[20], for the comparison I will use 5.0V curve with a clock frequency of 20MHz. For the ATMega328 it's active supply current draw is 11.8mA. As shown in Figure 30-2 on the ATMega1284 Datasheet (p. 424) the 5.0V curve at 20MHz has a active current draw of 16.9mA which means that the ATMega1284 has 1.43 times the current draw meaning with the same battery it would last 69.82% as long. In return for a few extra SPI connections and UART connections as well as bigger flash. This would add lot of convenience in the long term, but I don't think that the software will require that much. Therefore, the ATMega328 is the most suitable microcontroller for this job.

Figure 33-2. ATmega328: Active Supply Current vs. Frequency (1MHz - 20MHz)

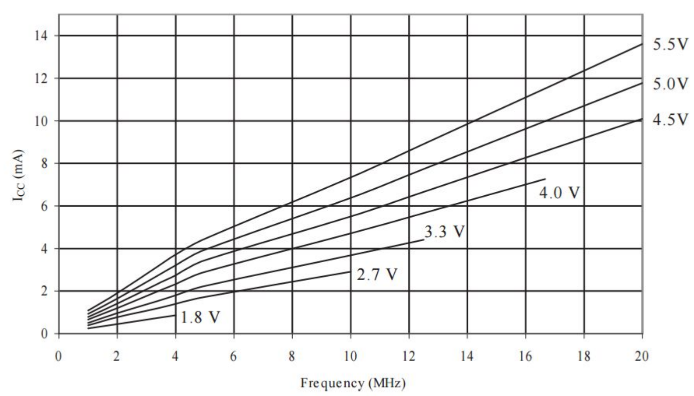
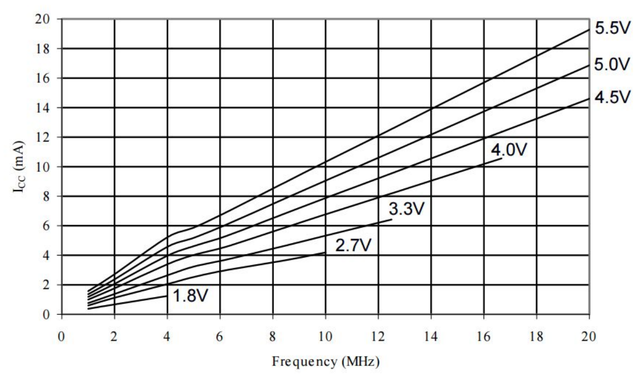


Figure 30-2. Active Supply Current vs. Frequency (1 - 20MHz)



Power supply

A smartwatch power supply must come from a battery for the device to be portable, the battery must be rechargeable as otherwise it would be extremely inconvenient to have to buy AA batteries every week.

In choosing a rechargeable battery for the device, I am limited to several types: nickel metal-hydride, lead-acid or lithium-ion. Nickel metal-hydride are the most common rechargeables and are often found in standardised forms such as AA at 9V. Lead-acid batteries are the oldest rechargeable batteries first

[17] Microchip.com. (n.d.). ATtiny85 - 8-bit AVR Microcontrollers. [online] Available at: https://www.microchip.com/wwwproducts/en/ATtiny85 [Accessed 1 Nov. 2017].
[18] Microchip.com. (n.d.). ATmega328 - 8-bit AVR Microcontrollers. [online] Available at: https://www.microchip.com/wwwproducts/en/ATmega328 [Accessed 1 Nov. 2017].
[19] Microchip.com. (n.d.). ATmega1284 - 8-bit AVR Microcontrollers. [online] Available at: https://www.microchip.com/wwwproducts/en/Atmega1284 [Accessed 1 Nov. 2017].
[20] ATmega328/P Datasheet Complete. (n.d.). 1st ed. [ebook] Microchip. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf [Accessed 1 Nov. 2017].

invented in 1859[21] but they have the lowest energy density and are often found as car batteries. Lithium-ion batteries are extremely lightweight and provide the highest energy density but there are safety issues as due to the high energy density as a small little pierce in the battery can lead to them catching alight. Lithium-ion are most common is smartphones, and laptops. As such, I think despite it small safety precaution, lithium ion represents the best type of rechargeable battery I can get for this project. Lithium-ion batteries tend to have a voltage from 3.3V when depleted to 4.2V when fully charged[22], thus they have a nominal voltage of 3.7V. In terms of how big the battery should be, I think I will leave that to when I roughly know how much the circuit consumes so I can calculate how big the battery should be for it to last a week.

ATMega328 Settings

On the ATMega328 datasheet it specifies certain settings of which the chip is compatible with. Firstly, as mentioned earlier lithium ion batteries have a nominal voltage of 3.7 Volts, I will use a voltage regulator to regulate this down to 3.3V, the use of a voltage regulator will produce a stable output voltage even if the voltage of the battery changes of times as it will from 4.2V when fully charged to 3.3 When depleted. This makes the power supply more reliable and provides stability. Therefore, as shown in ATMega328 datasheet[23] (Figure 5-2) with the 28-pin micro lead frame (MLF), the pins 7 and 8 are XTAL1 and XTAL2 which are the two pins where the external crystals are connected to. The value of the external crystal oscillator varies the speed of the ATMega328, as in the datasheet in the introduction it states that "By executing powerful instructions in a single clock cycle, the ATmega328P achieves throughputs close to 1 MIPS (Million Instructions per Second) per MHz." Therefore by changing the external crystal oscillator, you can change the amount of instructions the ATMega328 can achieve per second. The datasheet (Figure 32-1) shows the Safe Operating Area on a graph of Maximum Frequency vs. VCC, this compares the crystal frequencies to the supply voltage, therefore using the regulated 3.3V supply therefore a safe crystal would have to be less than 12 MHz, so a suitable crystal would be for example 8MHz, but the ATMega328 offers an internal oscillator which goes up to 8MHz, however these are not the most accurate of devices as it states in the datasheet (Table 32-5), it shows that when the chips are manufactured they calibrate it to 8.0MHz at 3.0V with a Calibration Accuracy of ±10% which isn't good. But it is possible to configure and calibrate the internal oscillator to a frequency of 8MHz with an accuracy of ±1%. Even though configuring the internal oscillator would save space on the breadboard, it is much simpler to use an external oscillator with a must greater accuracy than the internal oscillator.

Figure 5-3. 32-pin TQFP Top View

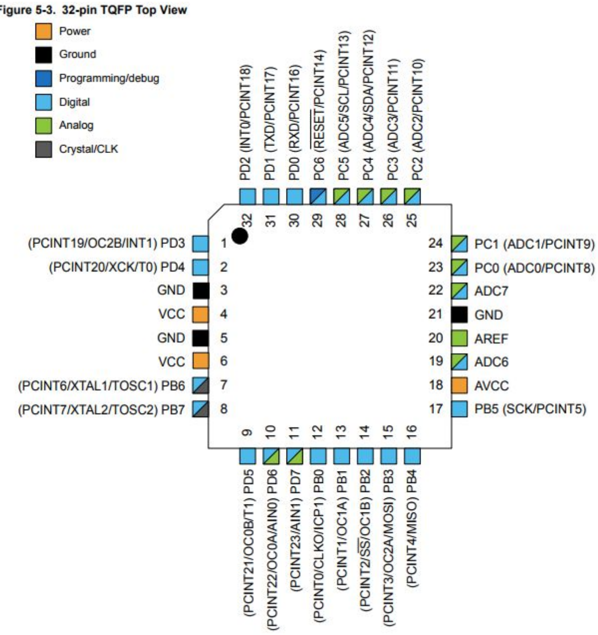


Figure 32-1. Maximum Frequency vs. V_{CC}

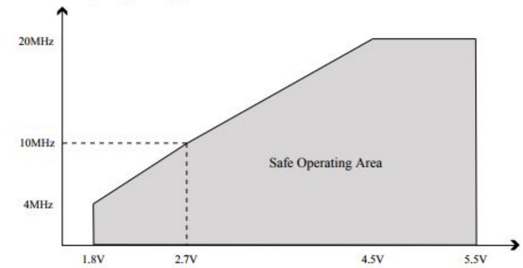


Table 32-5. Calibration Accuracy of Internal RC Oscillator

	Frequency	V_{CC}	Temperature	Calibration Accuracy
Factory Calibration	8.0MHz	3.0V	25°C	±10%
User Calibration	7.3 - 8.1MHz	1.8V - 5.5V	-40°C to - 85°C	±1%

[21]En.wikipedia.org. (2018). Lead–acid battery. [online] Available at: https://en.wikipedia.org/wiki/Lead%E2%80%93acid_battery [Accessed 5 Nov. 2017].
[22]Schneider, B. (2012). A Guide to Understanding LiPo Batteries. [online] Roger's Hobby Center. Available at: https://rogershobbycenter.com/lipoguide/ [Accessed 3 Nov. 2017].
[23]ATmega328/P Datasheet Complete. (n.d.). 1st ed. [ebook] Microchip. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf [Accessed 1 Nov. 2017].

Assuming that I use a 8MHz external oscillator at a regulated 3.3V supply. So by using the datasheet (Figure 33-2) to estimate what the ATMega328 will consume in terms of current draw when the device is in the Active mode. At 8MHz and using the 3.3V line, the ATMega328 should consume ~3mA. Also, when the ATMega328 enters Idle mode which is defined in the datasheet (Section 14.4) as "the SLEEP instruction makes the MCU (Microcontroller Unit) enter Idle mode, stopping the CPU but allowing the SPI, USART (Universal Synchronous/Asynchronous Receiver-Transmitter), Analog Comparator, 2-wire Serial Interface, Timer/Counters, Watchdog and the interrupt systems to continue operating." The Idle mode allows for the ATMega328 to consume less power while still being able to be woken up via the interrupt system. As the datasheet (Figure 33-7) shows the current draw of the ATMega328 with a regulated 3.3V supply and a crystal oscillator of 8MHz consumes a current of ~0.7mA. This is a drastic decrease in the current draw and the Idle mode is perfect for reducing the current draw when the smartwatch is not in use, such as a standby mode when the screen is off. The interrupt system are events either external or internal that require immediate action from the microcontroller, the microcontroller is then paused and a set of code called the Interrupt Service Routine, this runs as soon as the microcontroller interrupt is triggered. There are two types of interrupts, there are internal ones which are triggered by the ATMega328's Timer or Analog Comparator. External interrupts are triggered by the INT or PCINT pins. These external interrupts can be triggered by a rising or falling edge, which type should trigger the interrupt can be sets in the External Interrupt Control Register A. The INT pins are detected asynchronously, this means that they can be used for waking up the ATMega328 from sleep modes.

Figure 33-2. ATmega328: Active Supply Current vs. Frequency (1MHz - 20MHz)

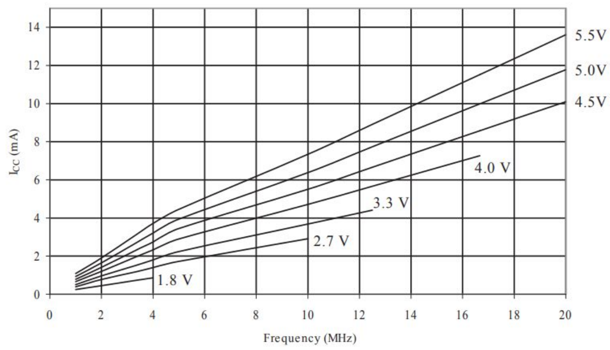
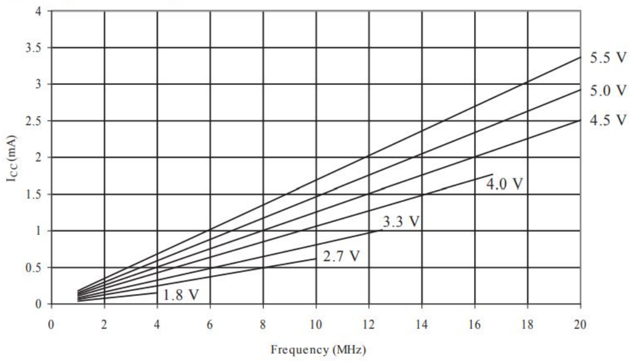


Figure 33-7. ATmega328: Idle Supply Current vs. Frequency (1MHz - 20MHz)



Components

After deciding on using a ATMega328 as the microcontroller, I know need to find components which can work with the microcontroller and communicate with it, to send it data.

Temperature Sensor

The smartwatch's temperature sensor should be low powered and accurate. I have come across two options, of which both use the I2C communication method and both have i2c addresses which can be edited by pulling certain pins high or low. These chips are the MCP9808 and the ADT75ARZ, the MCP9808 has a current consumption of 200uA and has a low powered state of 0.1uA. The ADT75ARZ has a current consumption at the regulated 3.3V supply, by using figure 7 of the datasheet[24], would be about 203uA. The ADT75ARZ has a resolution of ±0.0625°C compared to ±0.5°C on the MCP9808[25], as such the ADT75ARZ is more precise than the MCP9808. In terms of price, the MCP9808 costs for 1 unit £0.936, whereas the ADT75ARZ costs £1.79 for 1 unit. But if you order 8000+ units the cost of the ADT75ARZ goes down to £0.748 and if you order 1000+ of the MCP9808 the cost goes down to £0.651. Overall, even though the ADT75ARZ costs more per unit, that extra cost is justified by the chip being more accurate.

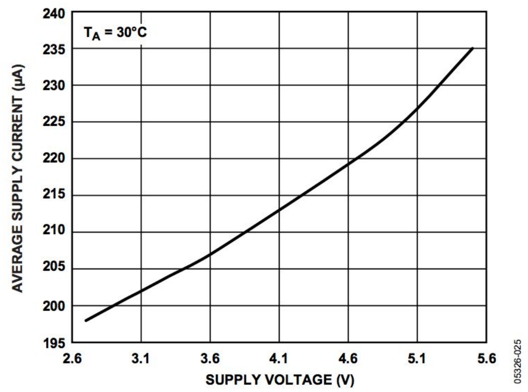


Figure 7. Average Operating Supply Current vs. Supply Voltage at 30°C

Using the ADT75ARZ

The datasheet of the ADT75ARZ on Table 5 - Pin Function Descriptions, it lists each of the 8 pins and their purpose. Pin 1 and Pin 2 are the SDA and SCL respectively and their purpose it to act as the communication

Table 5. Pin Function Descriptions

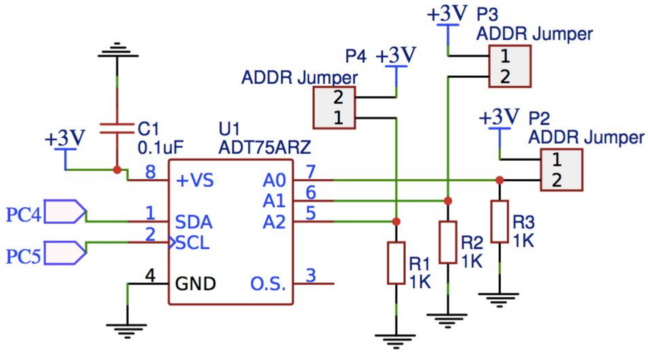
Pin No.	Mnemonic	Description
1	SDA	SMBus/I²C Serial Data Input/Output. Serial data that is loaded into and read from the ADT75 registers is provided on this pin. Open-drain configuration; needs a pull-up resistor.
2	SCL	Serial Clock Input. This is the clock input for the serial port. The serial clock is used to clock in and clock out data to and from any register of the ADT75. Open-drain configuration; needs a pull-up resistor.
3	OS/ALERT	Over- and Undertemperature Indicator. Default power as an OS pin. Open-drain configuration; needs a pull-up resistor.
4	GND	Analog and Digital Ground.
5	A2	SMBus/I²C Serial Bus Address Selection Pin. Logic input. Can be set to GND or V_{DD}.
6	A1	SMBus/I²C Serial Bus Address Selection Pin. Logic input. Can be set to GND or V_{DD}.
7	A0	SMBus/I²C Serial Bus Address Selection Pin. Logic input. Can be set to GND or V_{DD}.
8	V_{DD}	Positive Supply Voltage, 2.7 V to 5.5 V. Decouple the supply to ground.

pins for I2C, these will be connected to PC4 and PC5 of the ATMega328. The OS/Alert pin 3, is not needed for this project. Pin 4 is GND or Ground which must be connected to 0V, Pin 8 is the VDD of Positive Supply voltage, this has a voltage range of 2.7V to 5.5V which means I should connect it to the regulated 3.3V. Decouple the power supply, this means I need a decoupling capacitor between the VDD and GND. "A decoupling capacitor job is to suppress high-frequency noise in power supply signals. They take tiny voltage ripples, which could otherwise be harmful to delicate ICs, out of the voltage supply."[26] This is to protect the circuit. Finally, pins 5,6,7 are the address pins, by changing whether they are set to GND or VDD

[24] Farnell.com. (n.d.). ±1°C Accurate, 12-Bit Digital Temperature Sensor Data Sheet ADT75. [online] Available at: http://www.farnell.com/datasheets/2258695.pdf [Accessed 11 Nov. 2017].
[25] Farnell.com. (n.d.). MCP9808 - +/-0.5°C Maximum Accuracy Digital Temperature Sensor. [online] Available at: http://www.farnell.com/datasheets/1522173.pdf [Accessed 12 Nov. 2017].
[26] Learn.sparkfun.com. (n.d.). Capacitors - learn.sparkfun.com. [online] Available at: https://learn.sparkfun.com/tutorials/capacitors [Accessed 11 Nov. 2017].

changes the ADT75ARZ I2C address, I will by default pull these to GND but I will have a jumper[27] which will enable me to change the address later.

Afterwards I designed the circuit diagram[28] as shown on the right with EasyEDA[29] as shown on the right. I used what I mentioned earlier about the I2C address jumpers these jumpers will enable me to solder across these connections in order to edit the I2C address, I cannot use a definitive address until I have got the chip as different suppliers have different i2c address.



Programming the ADT75ARZ

In order to program this ADT75ARZ, I will use this general ADT75[30] code from github.

<table>
<tr><td>#include <Wire.h></td><td>This enables you to connect to I2C devices from importing this library[31].</td></tr>
<tr><td>#define Addr 0x48</td><td>This address varies with the setup of the address pins, Table 12 from the datasheet shows the addresses with the setups of A2,A1 and A0 which are configurable, by altering whether they are pulled high (binary 1) or pulled low (binary 0)
Table 12. SMBus/I²C Bus Address Options
<table>
<tr><th colspan="7">Binary</th><th></th></tr>
<tr><th>A6</th><th>A5</th><th>A4</th><th>A3</th><th>A2</th><th>A1</th><th>A0</th><th>Hex</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0x48</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0x49</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0x4A</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0x4B</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0x4C</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0x4D</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0x4E</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0x4F</td></tr>
</table></td></tr>
<tr><td>void setup()
{
  Wire.begin();
  Serial.begin(9600);
}</td><td>The void setup is a “function called when a sketch[32] starts”[33]. Wire.begin() initializes[34] the wire.h library imported earlier.Serial.begin(9600) Initialises serial communication, by setting set baud rate to 9600, this enables communications to be set to the computer, that is programming the device in order to debug it.</td></tr>
<tr><td>void loop()
{</td><td>Sets up the loop function[35] which loop continuously till the end of the program.</td></tr>
</table>

[27]Sparkfun.com. (n.d.). Jumper - 2 Pin - PRT-09044 - SparkFun Electronics. [online] Available at: https://www.sparkfun.com/products/9044 [Accessed 18 Dec. 2017].
[28]+3V is used as shorthand for +3.3V
[29]https://easyeda.com/
[30]Singh, Y. (2016). ADT75.ino. [online] GitHub. Available at: https://github.com/ControlEverythingCommunity/ADT75/blob/master/Arduino/ADT75.ino [Accessed 15 Nov. 2017].
[31]Arduino.cc. (n.d.). Arduino - Wire. [online] Available at: https://www.arduino.cc/en/Reference/Wire [Accessed 18 Oct. 2017].
[32]Sketch is another word for program
[33]Arduino.cc. (2017). Arduino Reference. [online] Available at: https://www.arduino.cc/reference/en/language/structure/sketch/setup/ [Accessed 8 Dec. 2017].
[34] Arduino.cc. (n.d.). Arduino - WireBegin. [online] Available at: https://www.arduino.cc/en/Reference/WireBegin [Accessed 15 Dec. 2017].
[35] Arduino.cc. (n.d.). Arduino - Loop. [online] Available at: https://www.arduino.cc/en/Reference/Loop/ [Accessed 7 Oct. 2017].

unsigned int data[2]; *Wire.beginTransmission(Addr);* *Wire.write(0x00);* *Wire.endTransmission();*	Unsigned int data[2] is defining a variable to store 2 bytes of data[36]. Wire.beginTransmission(Addr) this begins the wire communication and uses the Addr variable setup earlier in the program. Wire.write(0x00) selects the data register which stores the temperature data. Wire.endTransmission() ends this current transmission of data.
Wire.requestFrom(Addr, 2); *if (Wire.available() == 2)* *{* *data[0] = Wire.read();* *data[1] = Wire.read();* *}*	Wire.requestFrom(Addr, 2)[37] requesting 2 byte of data from the i2c device at address Addr. if (Wire.available() == 2) starts an if function with the statement of Wire.available() == 2 which checks that the 2 bytes of data was successfully requested. Then data[0] = Wire.read(); reads the first byte of data and data[1] = Wire.read(); reads the second byte.
*int temp = ((data[0] * 256) + data[1]) / 16;* *if(temp > 2047)* *{* *temp -= 4096;* *}*	This sections converts the raw data to 12 bits. This equation is noted in the datasheet as shown below, if(temp>2047) then temp -=4096 accounts for negative temperatures as shown below. **12-Bit Temperature Data Format** • Positive Temperature = ADC Code(d)/16 • Negative Temperature = (ADC Code(d)[1] – 4096)/16,
*float cTemp = temp * 0.0625;*	Float cTemp = temp * 0.0625, is defining a variable cTemp, short for celsius temperature, by multiplying the 12 bits by the resolution of the device.
Serial.print("Temperature in Celsius : "); *Serial.print(cTemp);* *Serial.println(" C");* *}*	*Serial.print()*[38] prints the celcius to the serial monitor which was defined earlier at being at a baud rate of 9600. Then finally close the loop function off with }.

Timekeeping

The SmartWatch must have a reliable and precise clock, which can tell the time, day, month, year and weekday. Such feature is imperative for a watch of any kind, and a smartwatch must be able to go above and beyond what a traditional watch can do. To implement time telling to this device, there are several options for telling the time:

1. The main code loop, could keep track of the time by incrementing variables at set intervals, this is a very impractical method of timekeeping as if the device was to lose power, then the variables would be reset. Therefore this method would be very inaccurate and would require the time to be calibrated every restart.
2. Another method, is the use of GPS as GPS satellites transmit the time in order to calculate the distance to each satellite, so it would be possible to get the time from them. But GPS chips are expensive and relatively large packages

[36] Arduino.cc. (2017). Arduino Reference. [online] Available at: https://www.arduino.cc/reference/en/language/variables/data-types/unsignedint/ [Accessed 23 Nov. 2017].

[37] Arduino.cc. (n.d.). Arduino - WireRequestFrom. [online] Available at: https://www.arduino.cc/en/Reference/WireRequestFrom [Accessed 26 Nov. 2017].

[38] Arduino.cc. (2017). Arduino Reference. [online] Available at: https://www.arduino.cc/reference/en/language/functions/communication/serial/print/ [Accessed 1 Nov. 2017].

3. Real-Time Clocks could be used to keep track of the time, Real-Time Clocks or RTC's are low power integrated circuits with their own power sources, they have the time set once and then they run on their own. The separate power source as well as the fact they consume small power means that RTC's can operate for years without recharging.

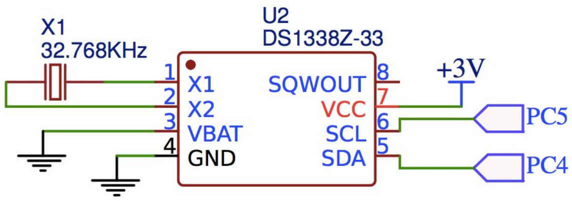
It is clear that the use of RTC[39] would be much more practical and efficient at keeping track of the time for the watch.

Real Time Clock

There are many different RTC chips most of them are either DS1302, DS1307 or the DS1338, the DS1338 is the only one that works at 3.3V as the others work at 5V. The DS1338 is the equivalent of the DS1307 but the DS1307 is only 5V compatible and the DS1338 is 3V compatible, as the regulated power supply is 3.3V thus I need a 3V compatible device. I don't want to have a boost converter in the circuit to boost the voltage from 3.3V to 5v as it wouldn't be necessary when you can find an alternative at 3.3v which is the DS1338Z-33.

Using the DS1338Z-33

On the DS1338Z-33 datasheet[40] the pins descriptions are, pin 1 and 2 are X1 and X2, there is a crystal oscillator connected to these two pins, and it must be 32.768KHz crystal as the internal circuitry is designed to work with this value in order to track the time. Pin 3 is VBat which isn't required for this project and therefore this pin must be grounded. Pin 4 is Ground and therefore like pin 3 must be grounded. Pin 5 and 6 are SDA and SCL respectively and therefore must connect to PC4 and PC5 of the microcontroller. Pin 7 is SQW/OUT which produces a square wave, this is not require and therefore can be left unconnected. Pin8 is VDD where the regulated 3.3V supply is to be connected to.



Programming the RTC

```
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"};

void setup () {
  Serial.begin(57600);
  if (! rtc.isrunning()) {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}

void loop () {
    DateTime now = rtc.now();

    Serial.print(now.year(), DEC);
```

[39] Real Time Clock
[40] Datasheets.maximintegrated.com. (2015). DS1338 IIC RTC with 56-Byte NVRAM. [online] Available at: https://datasheets.maximintegrated.com/en/ds/DS1338-DS1338Z.pdf [Accessed 21 Nov. 2017].

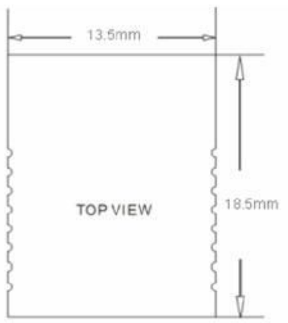
```
        Serial.print('/');
        Serial.print(now.month(), DEC);
        Serial.print('/');
        Serial.print(now.day(), DEC);
        Serial.print(" (");
        Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
        Serial.print(") ");
        Serial.print(now.hour(), DEC);
        Serial.print(':');
        Serial.print(now.minute(), DEC);
        Serial.print(':');
        Serial.print(now.second(), DEC);
        Serial.println();

        delay(2000);
    }
```

The code above every 2 seconds prints out the year, month, day, day of week, hour, minute and second. The rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); adjusts the time when a new code is uploaded, this keeps the RTC uptodate.

Bluetooth

There are a wide range of bluetooth devices, which could be used for this project. But I am limited as I need a device which can be soldered by hand, therefore the HM-11[41] and BLE112[42] are my options. They both are connected via UART and they both run on AT commands[43], all AT commands begin with AT such as "AT+NAME?" would return the Bluetooth device's name. The HM-11 is 18.5mm by 13.5mm as shown in the diagram. The BLE112 on the other hand is 18.10mm by 12.05mm, as shown in Figure 10 [44]. The table below shows the current consumption of these devices in different modes. Both models have a low power state. Overall, the HM-11 has a better current consumption while in modes Transmit and Receive whereas the BLE112 has a lower current consumption while in low powered mode. The HM-11 in low power consumes 14.4ma a day whereas the BLE112 takes 3600 days to consume the 14.4ma in while the BLE112 is in low powered mode. Even though the HM-11 consumes less current while in transmit and/or receive states, I don't intend for the bluetooth modules to be in constant connection with a phone as this would deplete the battery very quickly, most likely in under a day.



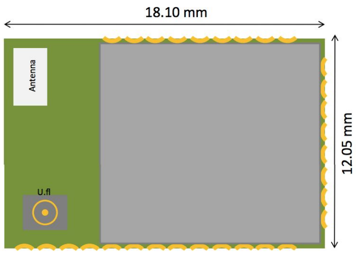


Figure 10: Physical dimensions (top view)

Mode	HM-11 Current Consumption	BLE112 Current Consumption
Transmit	15ma	28ma
Receive	8.5ma	25ma
Low Power	0.6ma	0.004ma

[41]Wiki.seeed.cc. (n.d.). Bluetooth V4.0 HM-11 BLE Module - Seeed Wiki. [online] Available at: http://wiki.seeed.cc/Bluetooth_V4.0_HM_11_BLE_Module/ [Accessed 16 Oct. 2017].

[42]Silabs.com. (n.d.). Bluegiga BLE112 Bluetooth Smart Module | Silicon Labs. [online] Available at: https://www.silabs.com/products/wireless/bluetooth/bluetooth-low-energy-modules/ble112-bluetooth-smart-module [Accessed 22 Oct. 2017].

[43]Itead.cc. (n.d.). Serial Port Bluetooth Module (Master/Slave) : HC-05 - ITEAD Wiki. [online] Available at: https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05 [Accessed 30 Oct. 2017].

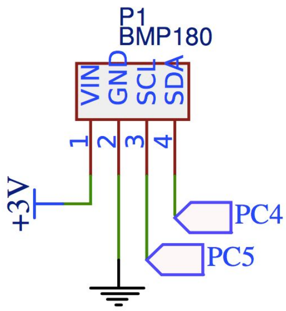
[44]Silabs.com. (2015). BLE112 Datasheet.pdf. [online] Available at: https://www.silabs.com/documents/login/data-sheets/BLE112-DataSheet.pdf [Accessed 23 Dec. 2017].

Pressure Sensor

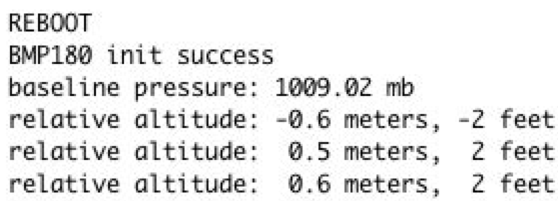
A pressure sensor measures the air pressure[45] they are called barometric pressure sensors. They are very useful as “pressure changes with altitude”[46] and so a pressure sensor can convert the air pressure and thus calculate the altitude of the device. There are a few module thats are compatible with the arduino software such as the SPX3058D but it is a very bulky device, there is the SPD005G which is not as large but still not portable and finally there is the BMP180, which is the successor to the BMP085. The BMP180 is designed for small applications but is still fairly accurate.

Using the BMP180

The BMP180 is a “high performance atmospheric pressure sensor designed for use in applications such as smartphones, tablet devices, sports devices, etc.”[47] and is a “Ultra-small 7-pin LGA package”. As a result of the chip being so small and being high performance, it wouldn’t be possible for this chip to be hand soldered and for that reason, I think that I will use a breakout board, where I can add through holes to the breadboard so I can simply insert the breakout board. Not only will this allow me to insert the BMP180 breakout but if further down the line, I want to add other sensors, the through holes allow for such future modifications. For the breadboard I will use a GY-68 BMP180[48] as it is the smallest breakout board for the BMP180 measuring 10mm by 13mm. The schematic for this is very simple as shown on the right.



Coding the BMP180 is again fairly simple, there are many libraries which enable communications between the microcontroller and the BMP180 chip but Sparkfun library[49] is the smallest at 8496 bytes including code to operate it. I tested the Sparkfun code and the result is as shown. The pressure was 1009.02mb (mb is millibar where bar is equal to 100,000 Pascals[50]), the first relative altitude result was not correct. After a few hundred milliseconds for the sensor to adjust. After that I was getting result of 0.5m which is fairly accurate as it was measures on a desk on the ground floor, which is approximately 0.5m off the floor. The BMP180 also has a temperature sensor in it, but this isn’t as accurate as the ADT75ARZ, so I will leave the ADT75ARZ as the chip which is responsible for the temperature measurements.



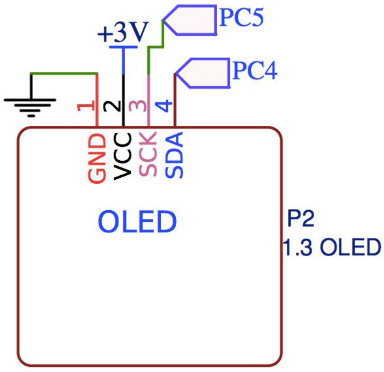
Display

In choosing a display for this device, there are many displays to choose from. Looking down Adafruit's' catalogue[51] of displays shows a lot, but I need one which is compatible and is the right size. There is a Monochrome 1.3" 128x64 OLED graphic display[52] which uses the SPI protocol, there also is OLED Breakout

[45] En.wikipedia.org. (n.d.). Pressure sensor. [online] Available at: https://en.wikipedia.org/wiki/Pressure_sensor [Accessed 2 Jan. 2018].
[46] Industries, A. (n.d.). BMP180 Barometric Pressure/Temperature/Altitude Sensor. [online] Adafruit.com. Available at: https://www.adafruit.com/product/1603 [Accessed 19 Oct. 2017].
[47] Rapidonline.com. (n.d.). Bosch BMP180 Absolute Pressure Sensor | Rapid Online. [online] Available at: https://www.rapidonline.com/bosch-bmp180-absolute-pressure-sensor-35-1630 [Accessed 14 Nov. 2017].
[48] Hotmcu.com. (n.d.). BMP180 Barometer Pressure/Temperature/Altitude Sensor [GY-68] - US $1.80 : HAOYU Electronics : Make Engineers Job Easier. [online] Available at: http://www.hotmcu.com/bmp180-barometer-pressuretemperaturealtitude-sensor-p-162.html [Accessed 8 Dec. 2017].
[49] Learn.sparkfun.com. (n.d.). BMP180 Barometric Pressure Sensor Hookup - learn.sparkfun.com. [online] Available at: https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup- [Accessed 20 Dec. 2017].
[50] The bar is a metric unit of pressure
[51] Industries, A. (n.d.). LCDs & Displays. [online] Adafruit.com. Available at: https://www.adafruit.com/category/63 [Accessed 21 Nov. 2017].
[52] Industries, A. (n.d.). Monochrome 1.3 128x64 OLED graphic display. [online] Adafruit.com. Available at: https://www.adafruit.com/product/938 [Accessed 11 Nov. 2018].

Board - 16-bit Color 1.5" w/microSD holder[53] which uses SPI, Adafruit SHARP Memory Display Breakout - 1.3" 168x144 Monochrome[54] this is a E-Paper display which is the best for low power but it is $24.95 and that is very steep price for a display. The 1.3" display is by far the best option, I am not looking for a multicolored display, this also comes in the i2c protocol which is much simpler to setup and for that reason, I will use the 1.3" monochrome OLED. They aren't cheap but compared to other models they are under £10, They can be bought for around £8.95 from eBay[55]

Using the 1.3" OLED



The 1.3" OLED is very simple to use just like all other i2c connections it only requires connections to the Micronrols PC4 and PC5 pins, and to VCC and ground. Coding the screen is a little more complicated and it would be very impractical to try to code it without a library, as there are two libraries, the U8GLIB and the Adafruit SSD1306 Library, the U8GLIB has many for functionalities than the SSD1306 as the U8GLIB has the ability to use many different fonts and functions. A simple hello world program is done as shown below.

```
#include "U8glib.h"

U8GLIB_SH1106_128X64 u8g(U8G_I2C_OPT_NONE);

void draw(void) {
  u8g.setFont(u8g_font_unifont);
  u8g.drawStr( 0, 22, "Hello World!");
}

void setup(void) {}

void loop(void) {
  u8g.firstPage();
  do {
    draw();
  } while( u8g.nextPage() );
}
```

Power management

In order to regulate power I need to use a linear voltage regulator, these drop the voltage to a predefined value, as LiPo Batteries have a nominal voltage of 3.7V. It can be regulated down to 3.3V. This provides a constant voltage as long as the input voltage is above the output.

The TP4056[56] is a linear charger for single cell LiPo Batteries, this chip enables safe charging for the battery. Combining this chip with the DW01+ - which is a battery protection integrated circuit, it protects against overcharge, overdischarge and overcurrent. Both of these chips provide typical applications circuit

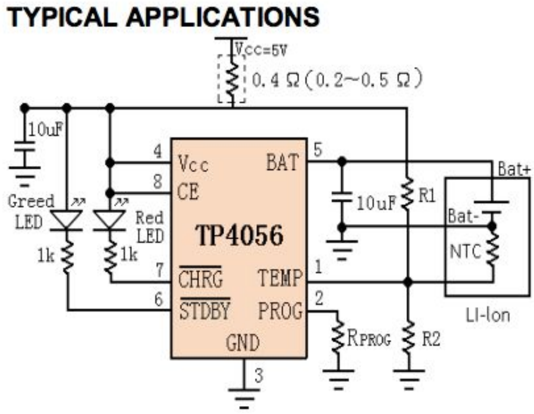
[53] Industries, A. (2018). OLED Breakout Board - 16-bit Color 1.5 w/microSD holder. [online] Adafruit.com. Available at: https://www.adafruit.com/product/1431 [Accessed 11 Nov. 2017].

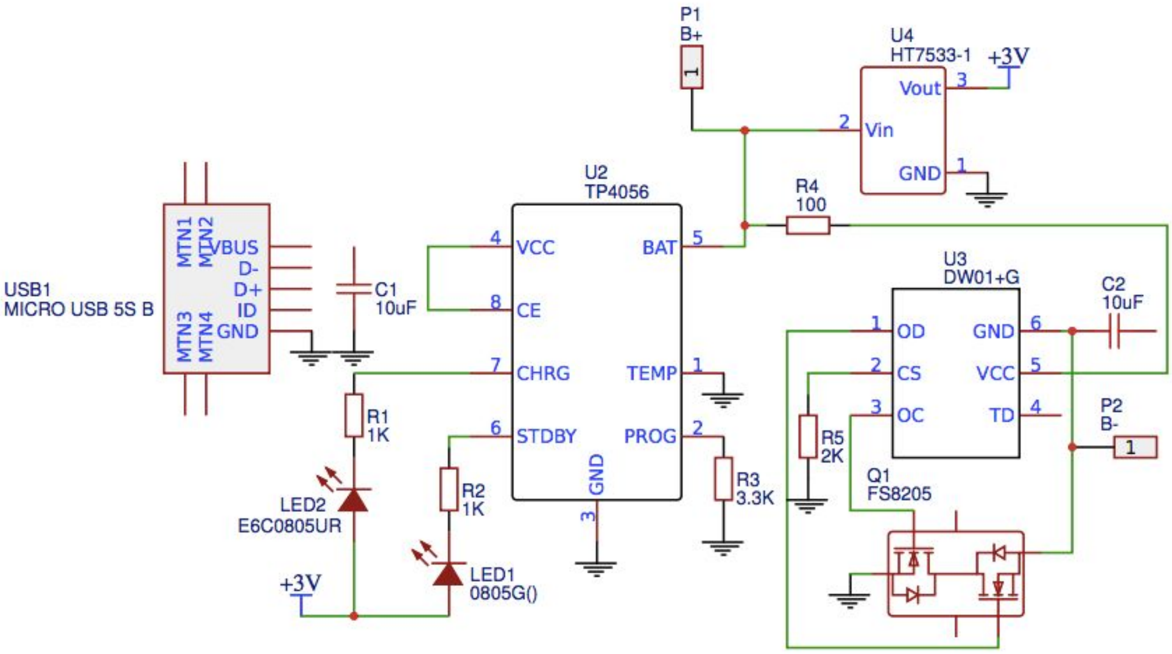
[54] Industries, A. (n.d.). Adafruit SHARP Memory Display Breakout - 1.3 168x144 Monochrome. [online] Adafruit.com. Available at: https://www.adafruit.com/product/3502 [Accessed 11 Nov. 2017].

[55] eBay. (n.d.). 1.3" 128x64 IIC I2C SPI OLED Display Module for Arduino White/Blue. [online] Available at: https://www.ebay.co.uk/itm/1-3-128x64-IIC-I2C-SPI-OLED-Display-Module-for-Arduino-White-Blue/282358769160 [Accessed 11 Nov. 2017].

[56] Dlnmh9ip6v2uc.cloudfront.net. (n.d.). TP4056 1A Standalone Linear Li-lon Battery Charger. [online] Available at: https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf [Accessed 31 Nov. 2017].

diagrams, which helps a lot when designing the schematic. By using the Typical application diagram from each of the chips I was able to produce a complete schematic, this includes a microUSB whereby the device could be charged from. The DW01+G in its datasheet it has two pins “MOSFET gate connection for charge control” and “MOSFET gate connection for discharge control” which means I require two MOSFET. The FS8205 offers two MOSFET gates which is convenient as this will simplify the circuit. Below is the complete power management circuit diagram, this also includes two Light Emitting Diodes which are using the charge and standby pins to give tell the user when the device is fully charged as the green Light Emitting Diode will be on, and when it is charging it will have a red Light Emitting Diode.





Interacting with the Device

The User Experience with this device should be efficient and seamless, nothing should seem unnecessary. As such the user's interaction with the device should be simple.

Buttons

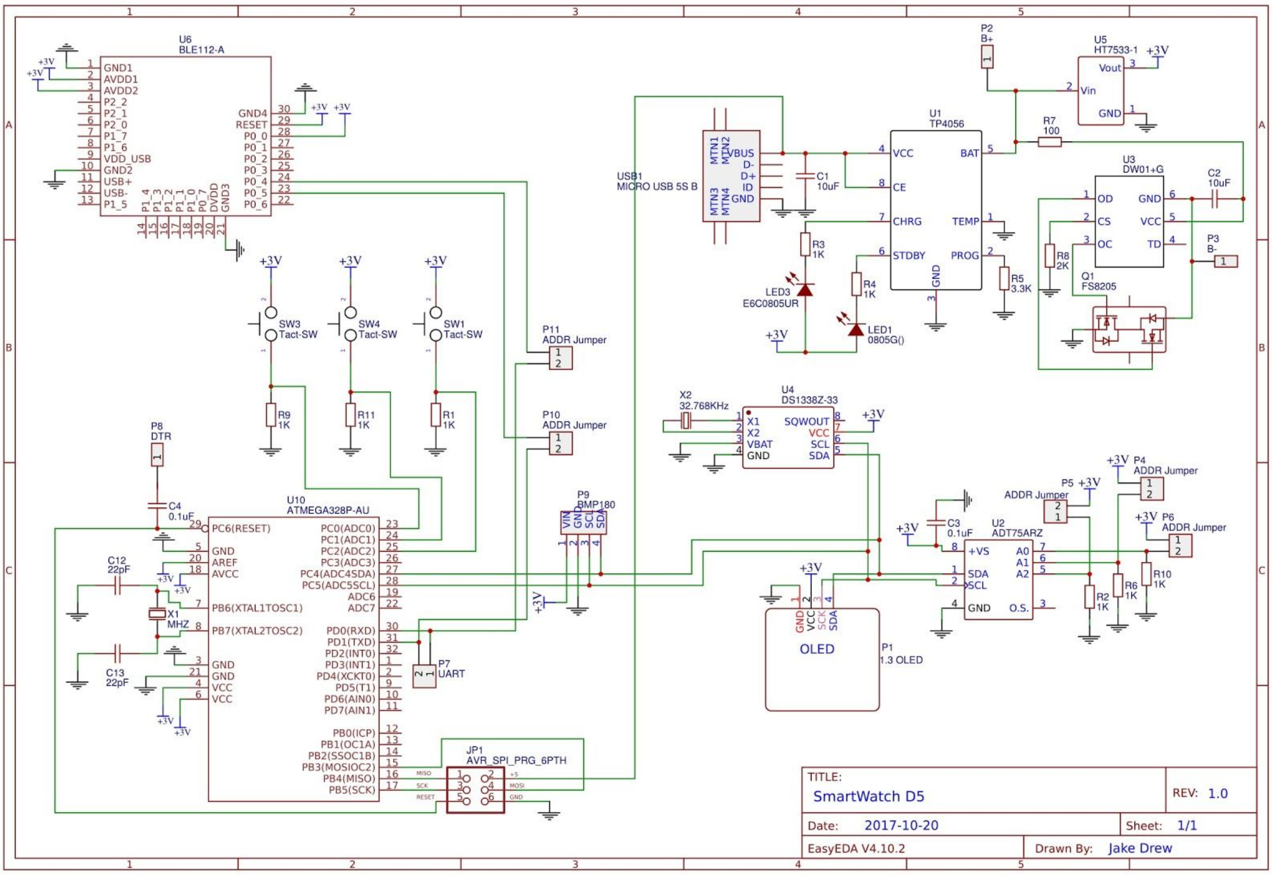
The device will certainly need an efficient way of interacting with the device, and I think that as mentioned earlier about saving power, a low power mode would need firstly a button to activate but also a way to wake-up the device. I would also need some other method of controlling the device when changing on board settings, moving through different screens. There are a few possible options:

1. Firstly, I could use 3 buttons, with two of them being for moving up or down, and the 3rd as an enter button to confirm selections. This would be very simple to implement as they would only require 3 digital pins.
2. I could use a button and a digital encoder which is a device which converts rotational movement to digital signals, this way, I would be able to replace the buttons for moving up and down with a encoder which would act as a scroller. This would be a very similar way to interact with the device as the Apple watch uses.
3. Thirdly, I could use a digital encoder as mentioned above, but instead of a button, I could use a force resistor, which changes it resistance depending on the force acted upon it, this way I could create a touch button

A digital encoder is a very smooth way of interacting with a device if the device has some scroll function, it works on the apple watch because the watch face zooms in and out, but for this device a digital encoder is unnecessary. The 3 buttons, is a much simpler method to use as adding buttons to the ATMega328 is simple and requires only 1 connections. In terms of what the buttons should do I think that the top buttons should reboot the device, the second should wake up the device and maybe change to the next screen and finally the third button at this stage in time in unnecessary, I will still have this on the watch as it could be programmed later if I add some features in the future.

The Completed Circuit Diagram

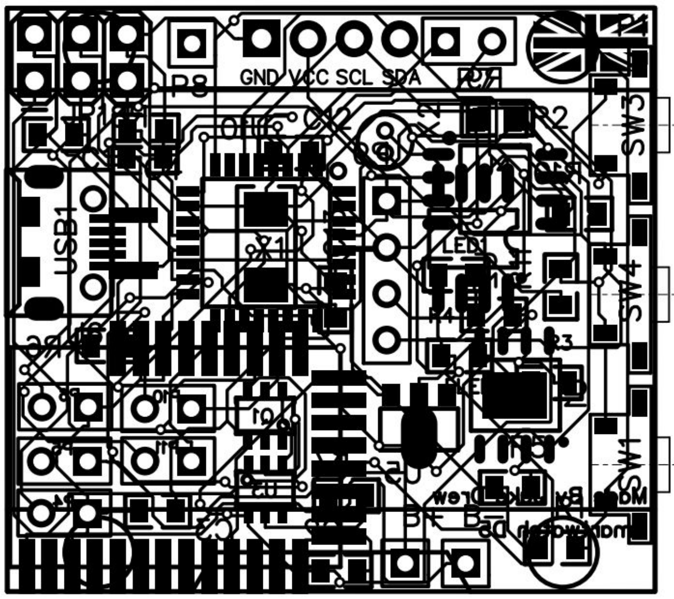
By combining all the circuit diagrams of each component and adding some tactile switches the completed circuit diagram[57] is shown below.

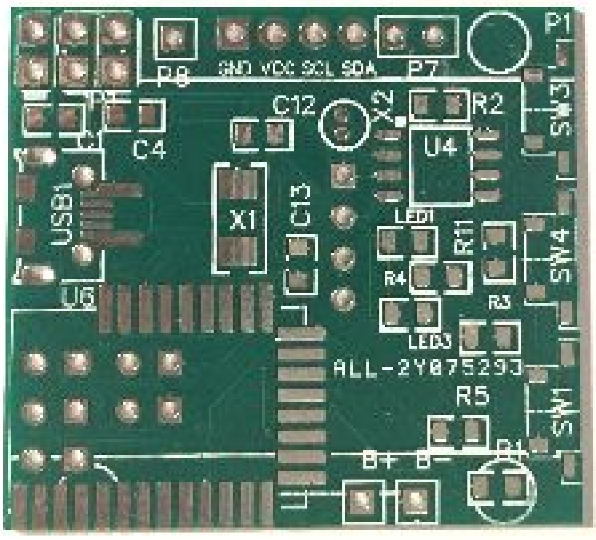


[57]Drew, J. (2018). Schematic.pdf. [online] GitHub. Available at: https://github.com/jakedrew/SmartWatch/blob/D1/Version/D5/Schematic.pdf [Accessed 8 Jan. 2018].

Printing the Board

I converted this circuit diagram to a Gerber file which is the file used for PCBs (Printed Circuit Boards). This file as shown below-left, was printed using a chinese company called allpcb[58] I was able to print 8 of these boards for £4.63 with free shipping with TNT. The boards arrived within 5 days. The Printed Circuit Board on the right is the final version which is the physical version of the design on the left.





58 allpcb.com

Power Consumption

To calculate power consumption, I must use every chip's datasheet and get their typical power consumption. This is the average of their power consumption and these chips will likely consumer above and are just as likely to consume less. This is just an estimate for what its battery life could be like.

The TP4056[59] doesn't consume any power from the battery as this is connected to the micro usb port as this charges the battery and so this value can be excluded. The DW01+G[60] consumes 0.006ma of current at its max, on average it consumes 0.003ma. The HT7533-1[61] which is the linear voltage regulator consumes 0.004ma. The temperature sensor the ADT75ARZ[62] consumes 0.350ma of current. The DS1338[63] has a current consumption of 0.110ma while it is at 3.3V. The Bluetooth BLE112[64] has a current consumption of 25ma which is the largest. The barometric pressure sensor the BMP180[65] consumes only 0.032ma. The microcontroller at 3.3V consumes 7.94ma[66] . The OLED display with every pixel lit up consumes 24.25ma. This comes to a total of 57.692ma which is not good, over the course of a day it would account to 1384.608 which is about as much as a smartphone battery holds. However this is assuming a lot of things which in practise wouldn't happen, it assumes that the OLED display has every pixel lit up at every point in the day. The implementation of a display which turns off after 30s of inactivity would greatly reduce the consumption. A 2016 report[67] from google on how people use their devices showed that on average people used their phones for 170 minutes a day, this is 11.8% of the day, even if the watch's display was on 11.8% of the time the current consumption would decrease from 24.25ma to 2.863ma. Using the Watch for 170 is unlikely. So if I assume that the display stays on for 30s after the user checks it, an article from Time[68], the average person checks their phone 46 times a day. This would account for 23 minutes a day or 1.6% of the day, this would decrease the OLED consumption from 24.25ma to 0.387ma. Not only that but this assumes every pixel is on, when in reality to read information off it, only some pixels would need to be on. If 25% of the display is on then the consumption reduces to 0.097ma a day. If I take the 1.6% figure and apply it to the ATMega328 then I get a figure of 3.05ma a day. If I also apply this same 1.6% figure to the BLE112 then I get a consumption of 9.6ma. Overall the consumption is now at 24.317ma a day. This means that I can get a week's worth of usage with a 240mAh battery, or 9.86 days to be exact. A 500mAh battery would get 20.56 days but that size is unnecessary.

[59] Dlnmh9ip6v2uc.cloudfront.net. (n.d.). TP4056 1A Standalone Linear Li-lon Battery Charger. [online] Available at: https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf [Accessed 31 Nov. 2017].

[60] Ic-fortune.com. (n.d.). DW01-G. [online] Available at: http://www.ic-fortune.com/upload/Download/DW01-G-DS-10_EN.pdf [Accessed 25 Nov. 2017].

[61] Docs-europe.electrocomponents.com. (2014). HT75xx-1. [online] Available at: http://docs-europe.electrocomponents.com/webdocs/140c/0900766b8140c21f.pdf [Accessed 24 Nov. 2017].

[62] Analog.com. (n.d.). ±1°C Accurate, 12-Bit Digital Temperature Sensor Data Sheet ADT75. [online] Available at: http://www.analog.com/media/en/technical-documentation/data-sheets/ADT75.pdf [Accessed 20 Nov. 2017].

[63] Datasheets.maximintegrated.com. (2015). DS1338 IIC RTC with 56-Byte NVRAM. [online] Available at: https://datasheets.maximintegrated.com/en/ds/DS1338-DS1338Z.pdf [Accessed 21 Nov. 2017].

[64] Silabs.com. (2015). BLE112 Datasheet.pdf. [online] Available at: https://www.silabs.com/documents/login/data-sheets/BLE112-DataSheet.pdf [Accessed 23 Dec. 2017].

[65] Cdn-shop.adafruit.com. (n.d.). BMP180 Digital pressure sensor. [online] Available at: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf [Accessed 19 Nov. 2017].

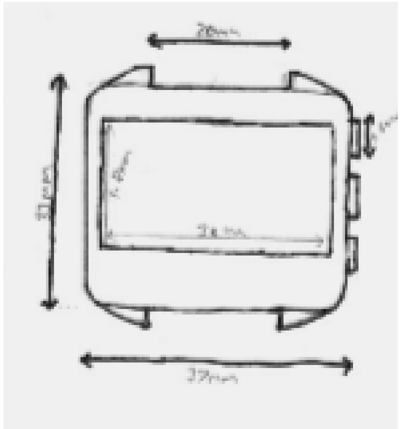
[66] Avrprogrammers.com. (n.d.). ATmega328p Power Consumption | avrProgrammers. [online] Available at: https://www.avrprogrammers.com/howto/atmega328-power [Accessed 18 Sep. 2017].

[67] Storage.googleapis.com. (2016). How People Use Their Devices. [online] Available at: https://storage.googleapis.com/think/docs/twg-how-people-use-their-devices-2016.pdf [Accessed 13 Oct. 2017].

[68] Eadicicco, L. (2015). Americans Check Their Phones 8 Billion Times a Day. [online] Time. Available at: http://time.com/4147614/smartphone-usage-us-2015/ [Accessed 4 Jan. 2018].

Designing the External

This watch, will definitely be a square shape, as it is very hard for me to get a circular screen and by making the device square you get a larger area to work with with the same width.



The watch will support a 20mm spring bar meaning that any 20mm watch strap will be able to fit in it. Therefore allowing the user to swap watch strap from a leather to a metal one within a few minutes. The left hand size will be a micro USB port for charging, the data pins on the micro USB port aren't connected to anything so you can upload or download any data from the micro USB port.

Comparison with alternatives

For my comparison with alternatives, I will compare to 3 other devices on the market, I have chosen 1 from each main operating system. The three devices are the LG G Watch on the Android operating system, the Apple Watch Series 3 from the WatchOS, and the Pebble Time running Pebble OS. By comparing three watchs from three operating systems, this allows me to get a good picture of how my watch compares. I will compare these watches on a few different categories, firstly on features - this compares how the watches perform and what they are capable of. Next I will compare the size of the watches - the size is important when comparing portable devices, a large watch will not be comfortable to wear. Afterwards, I will compare the battery life of these watches as a longer battery life is far more convenient for a watch. Finally, the cost of the watch is important, I will compare the cost to produce several thousand of my watches as this enables the cost to be reduced as otherwise it would be an unfair comparison.

Features

In regards to the smartwatches features, the clock speed is an important thing to note as the greater the clock speed the more operations a watch can perform a second. The ATMega328 was clocked at 8MHz which compared to the other devices it is extremely slow, the Pebble Time has a clock speed of 144MHz[69] and the Apple Watch Series 3[70] and the LG G Watch[71] both have a clock speed of 1200MHz which is 150 times faster than mine. This does make some sense in that those devices have to drive touch screens and both the pebble time and mine don't. My smartwatch has bluetooth low energy version 4.0 just like the LG G Watch, but the Pebble Time has version 4.1 and the Apple Watch Series 3 has version 4.2. These differences make nearly no difference with compatibility. But the higher the version, they do possess some more features but no major ones and they tend to be improvements in the advertising interval[72].

The Apple Watch Series 3 has 16GB of storage which is comparable to some phones a few years ago, but my smart watch has 32KB of storage, thees storage figures are where the software is run and the Apple watch has 536875 times more storage than mine. The LG G Watch has 4GB and the Pebble Time has 2MB, the Pebble Time is most compatible with 62.5 times larger storage. The differences in storage is a result of what the purpose of the smartwatch is to do, the apple watch and the LG G watch both have the ability to download and play music, with the average song being 3.5MB[73] you can understand why the Apple watch and the LG G watch both have large storage.

[69] SmartWatch Specifications. (n.d.). Pebble Time 2 - SmartWatch Specifications. [online] Available at: http://www.smartwatchspecifications.com/Device/pebble-time-2/ [Accessed 27 Nov. 2017].

[70] Support.apple.com. (2018). Apple Watch Series 3 - Technical Specifications. [online] Available at: https://support.apple.com/kb/SP766?locale=en_GB [Accessed 2 Jan. 2018].

[71] Gsmarena.com. (2017). LG G Watch W100 - Full phone specifications. [online] Available at: https://www.gsmarena.com/lg_g_watch_w100-7718.php [Accessed 2 Jan. 2018].

[72] En.wikipedia.org. (n.d.). Bluetooth. [online] Available at: https://en.wikipedia.org/wiki/Bluetooth#Bluetooth_4.1 [Accessed 20 Oct. 2017].

[73] FileCatalyst. (2013). How to transfer large video files using acceleration. [online] Available at: http://filecatalyst.com/todays-media-file-sizes-whats-average/ [Accessed 23 Nov. 2017].

Size

The PCB is 35.5mm wide and 32mm in length, if there was to be a case it would add a few millimeters in both wide and length, say the case was 2mm thick, such that the total watch came to 37.5mm by 34mm this would be similar to the Apple Watch Series 3 with a length of 38.6mm and a width of 33.3mm. The Pebble TIme is 40.5mm by 37.5mm which is relatively bigger, the LG G Watch is 46.5mm by 37.9mm. The size of these watches are all very similar with at most them being a few millimeters larger. The reasons for my size of the board is so the board isn't any bigger than the OLED screen.

Cost

Costing Up the Watch

The Atmel AVR Microcontroller the ATMega328P-AU costs £1.28[74], the DS1338Z-33 costs £1.44[75], The ADT75arz costs £0.748[76]. The TP4056 is unavailable from any United Kingdom suppliers, however it can be found in China. In China it can be brought for £0.17[77] each. The DW01+G costs £0.04[78] and the HT7533-1 costs £0.06[79]. The MicroUSB port costs £0.04[80], the Green and Red LED cost costing £0.013[81] and £0.01[82] respectively the FS8205 costs £0.08[83]. The tactile switches cost 0.173[84] and there are three of them so £0.519. There are a total of 8 0805 1KΩ resistors which cost £0.017[85] each so that's £0.13 for all of them, there is one 2.0KΩ 0805 resistor which costs £0.062[86]. The 100Ω 0805 resistor costs £0.09[87],and the 3.3k ohm resistor costs £0.10[88]. The PCB costs £3.7 from allpcb[89] for 5 but if ordered in bulk of 8000 it would reduce per unit cost from £0.74 to £0.17. The OLED display can bought for £2.96[90]. Finally the BLE112 module can be brought from Silicon Labs for £8.81. Therefore the total cost of this watch not including the

[74] Farnell. (2016). ATMEGA328P-AU - 8 Bit Microcontroller. [online] Available at: http://uk.farnell.com/microchip/atmega328p-au/mcu-8bit-atmega-20mhz-tqfp-32/dp/1715486 [Accessed 22 Sep. 2017].

[75] Farnell. (2018). DS1338Z-33+ - RTC IC. [online] Available at: http://uk.farnell.com/maxim-integrated-products/ds1338z-33/rtc-56b-nvram-i2c-1338-soic8/dp/2518690 [Accessed 17 Nov. 2017].

[76] Farnell. (2017). ADT75ARZ - Temperature Sensor IC. [online] Available at: http://uk.farnell.com/analog-devices/adt75arz/ic-12bit-temp-sensor-smd-soic8/dp/1117116 [Accessed 20 Dec. 2017].

[77] LCSC. (n.d.). TP4056. [online] Available at: https://lcsc.com/product-detail/PMIC-Battery-Management_TP4056_C16581.html [Accessed 15 Nov. 2017].

[78] LCSC. (n.d.). DW01+G - PDF Datasheet - Battery Protection ICs. [online] Available at: https://lcsc.com/product-detail/Battery-Protection-ICs_DW01-G_C14213.html [Accessed 21 Nov. 2017].

[79] LCSC. (n.d.). HT7533-1 - HT7533-1 - PDF Datasheet - Low Dropout Regulators. [online] Available at: https://lcsc.com/product-detail/Low-Dropout-Regulators-LDO_HT7533-1_C14289.html [Accessed 21 Nov. 2017].

[80] LCSC. (n.d.). MICRO USB 5S B Type horns. [online] Available at: https://lcsc.com/product-detail/Micro-USB-Connectors_MICRO-USB-5S-B-Type-horns-High-temperature_C10418.html [Accessed 21 Nov. 2017].

[81] LCSC. (n.d.). Green 0805 Light Emitting Diodes. [online] Available at: https://lcsc.com/product-detail/Light-Emitting-Diodes-LED_Green-0805-Highlighted_C2297.html [Accessed 21 Nov. 2017].

[82] LCSC. (n.d.). Red LED 0805 Light Emitting Diodes. [online] Available at: https://lcsc.com/product-detail/Light-Emitting-Diodes-LED_Red-LED-0805-Highlighted_C2295.html [Accessed 21 Nov. 2017].

[83] LCSC. (n.d.). FS8205 MOSFET. [online] Available at: https://lcsc.com/product-detail/MOSFET_FS8205_C32254.html [Accessed 21 Nov. 2017].

[84] Mouser Electronics. (2018). TL3330AF260QG E-Switch. [online] Available at: https://www.mouser.co.uk/productdetail/612-tl3330af260qg [Accessed 21 Nov. 2017].

[85] Farnell. (n.d.). SMD Chip Resistor 1kohm. [online] Available at: http://uk.farnell.com/yageo-phycomp/rc0805fr-071kl/res-thick-film-1k-1-0-125w-0805/dp/9237496 [Accessed 21 Nov. 2017].

[86] Mouser Electronics. (n.d.). ERJ-P6WF2001V Panasonic. [online] Available at: https://www.mouser.co.uk/productdetail/panasonic/erj-p6wf2001v?qs=sGAEpiMZZMtlubZbdhIBINqS%252bTkaCzO9LGeK1Zee4Co%3D [Accessed 21 Nov. 2017].

[87] Components, V. (n.d.). MCU0805PD1000DP500 Vishay. [online] Mouser Electronics. Available at: https://www.mouser.co.uk/productdetail/vishay-bc-components/mcu0805pd1000dp500?qs=sGAEpiMZZMtlubZbdhIBIEYCvaBm%252bsTP2IFqrUij%252bkg%3D [Accessed 21 Nov. 2017].

[88] Mouser Electronics. (n.d.). RS2012P-332-D-T5-3 Susumu. [online] Available at: https://www.mouser.co.uk/productdetail/susumu/rs2012p-332-d-t5-3?qs=sGAEpiMZZMtlubZbdhIBIBJoyJrfFkBdnAnSTDRYxTQ%3D [Accessed 21 Nov. 2017].

[89] http://www.allpcb.com/

[90] www.alibaba.com. (n.d.). 1.3 Inch 128x64 Resolution OLED Display. [online] Available at: https://wholesaler.alibaba.com/product-detail/1-3-inch-128-64-resolution_60717210502.html [Accessed 21 Nov. 2017].

case or strap is £17.09. A watch strap from the watch shop could cost £10.90[91] and the case if 3d printed costs £0.18 per cm^3 therefore for a 37.2mm by 34mm by 10.8mm would have a volume of 13.67 which would cost £2.46 to print. But if it was made out of steel it would cost £4.44 per part plus £3.7[92] per cm^3 therefore costing £54.98 to print out of steel. Steel would be a better material to make this out of as it would be less prone to breaking from drops but it would be scratched very easily. The cost with steel case and a 20mm strap would be £82.97.

Cost Comparison

The cost of this Smartwatch including the strap and the case is £82.97 compared to the three other watches, it is by far the cheapest. The Pebble Time costs 179[93] and the LG G Watch cost 159.99[94], these are definitely very expensive compared to mine, but this is not as expensive as the apple watch series 3 which is priced from £329[95]. So my watch is the cheapest, and has nearly all the same features as the other ones, however I have compared my watch at cost of producing it rather than retail price, The original apple watch was found to only cost 24%[96] of its retail price to produce. Using the same margin as apple would price mine at £345.7 which is definitely not worth it. I would think a reasonable price would be just over £100, maybe £109.99 as a retail price for mine. But I think that this watch doesn't possess the brank and so using the metal case would be too expensive. SO the

Building the Watch

Programming the Device

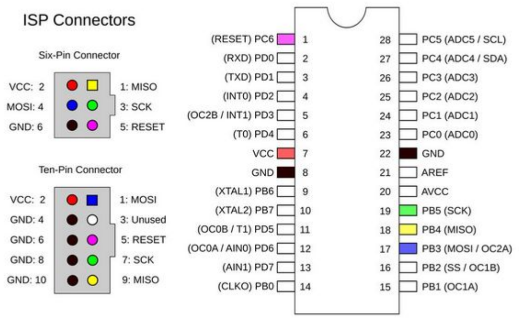
All of this projects code and files are available on this project's Github repository, github.com/jakedrew/SmartWatch/ .

Bootloading

A bootloader is a piece of software which runs before the main code, this sets up the chip with its basic settings such as clock frequency. To bootload it with a computer, I used a bootloader installer program[97] which was written by Nick Gammon, it installs the Optiboot bootloader onto the target device. The Optiboot isn't and official arduino bootloader but it is half the size at 512 bytes.

ISP

Once the bootloader in installed onto the chip there needs to be a way in which I can program the chip while it is has been soldered down, there are two method of programming UART (Serial) and ISP[98]. ISP or In-system programming uses 6 pins to program the chip. All it requires is 6 pins connect as shown here. There is also a 10 pin connector but for a device where size matters, it is necessary to have a larger connector. UART



[91] Watch Shop. (n.d.). Men's Morellato Stainless Steel Sprint Napa Leather Black 20mm Leather Strap. [online] Available at: http://www.watchshop.com/parts/mens-morellato-stainless-steel-sprint-napa-leather-black-20mm-leather-strap-a01x2619875019cr20-p100008505.html [Accessed 4 Jan. 2018].

[92] Shapeways.com. (n.d.). Stainless Steel 3D Printing Material Information - Shapeways. [online] Available at: https://www.shapeways.com/materials/steel [Accessed 2 Jan. 2018].

[93] Trusted Reviews. (2016). Pebble Time Review | Trusted Reviews. [online] Available at: http://www.trustedreviews.com/reviews/pebble-time [Accessed 2 Jan. 2018].

[94] Trusted Reviews. (2016). LG G Watch Review | Trusted Reviews. [online] Available at: http://www.trustedreviews.com/reviews/lg-g-watch [Accessed 2 Jan. 2018].

[95] Apple (UK). (2018). Apple Watch. [online] Available at: https://www.apple.com/uk/shop/buy-watch/apple-watch [Accessed 3 Jan. 2018].

[96] Tilley, A. (2015). The Apple Watch Sport Only Costs $83.70 To Make. [online] Forbes.com. Available at: https://www.forbes.com/sites/aarontilley/2015/04/30/the-apple-watch-only-costs-83-70-to-make/ [Accessed 4 Jan. 2018].

[97] Gammon, N. (n.d.). Microprocessors : Atmega bootloader programmer. [online] Gammon.com.au. Available at: http://www.gammon.com.au/bootloader [Accessed 18 Nov. 2017].

[98] En.wikipedia.org. (n.d.). In-system programming. [online] Available at: https://en.wikipedia.org/wiki/In-system_programming [Accessed 25 Nov. 2017].

is also another way in programming the device, in both methods they perfectly suitable for this application and for that reason, I will have the ability to use both on the device, whether they get used depends on the simplicity of the uploading process. Also, I need to bear in mind that I won't be able to use UART when the BLE112 is installed as it communicates via UART. I can then upload the Arduino ISP code[99], then I can upload any sketch by uploading using programmer.

[99]Arduino.cc. (n.d.). Arduino - ArduinoISP. [online] Available at: https://www.arduino.cc/en/Guide/ArduinoISP [Accessed 22 Nov. 2017].