

Index

- Why this ?
- Introduction + Objective
- Components
- Circuit diagram
- Working
- Applications
- Future enhancements
- References



Why this ?

- What is the most common question you ask everyday ?
- What time is it right now ?
- Some common might also involve, what is the todays date?, what is the current temperature ? Or if you are a health conscious person then it might be what is my heart rate ? Or you current body temperature?
- Setting of alarms etc
- Some smart watch's don't keep the time once the entire battery of the watch is dead, to overcome this we make sometime that never lose its time.



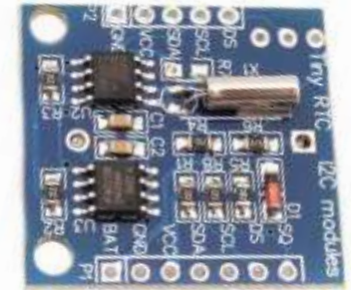
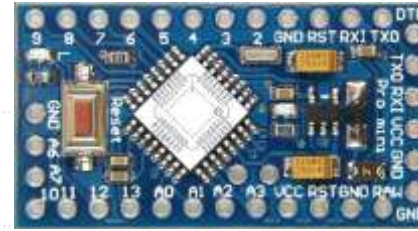
Introduction

- A problem with most of the watches digital or analog is that once the battery gets over we need to charge them and manually set the time again.
- **My objective is to build a smart watch that can display the current accurate time and the current date with leap year date correction. And to display the body temperature as well as this watch would also display the current heart rate of the person wearing it and feature an alarm that turns a haptic feedback motor when the alarm turns on..(including some additional settings)**
- The project focus more on the software rather than the hardware.
- (Also trying to make a longer lasting battery life device than traditional smart watches)

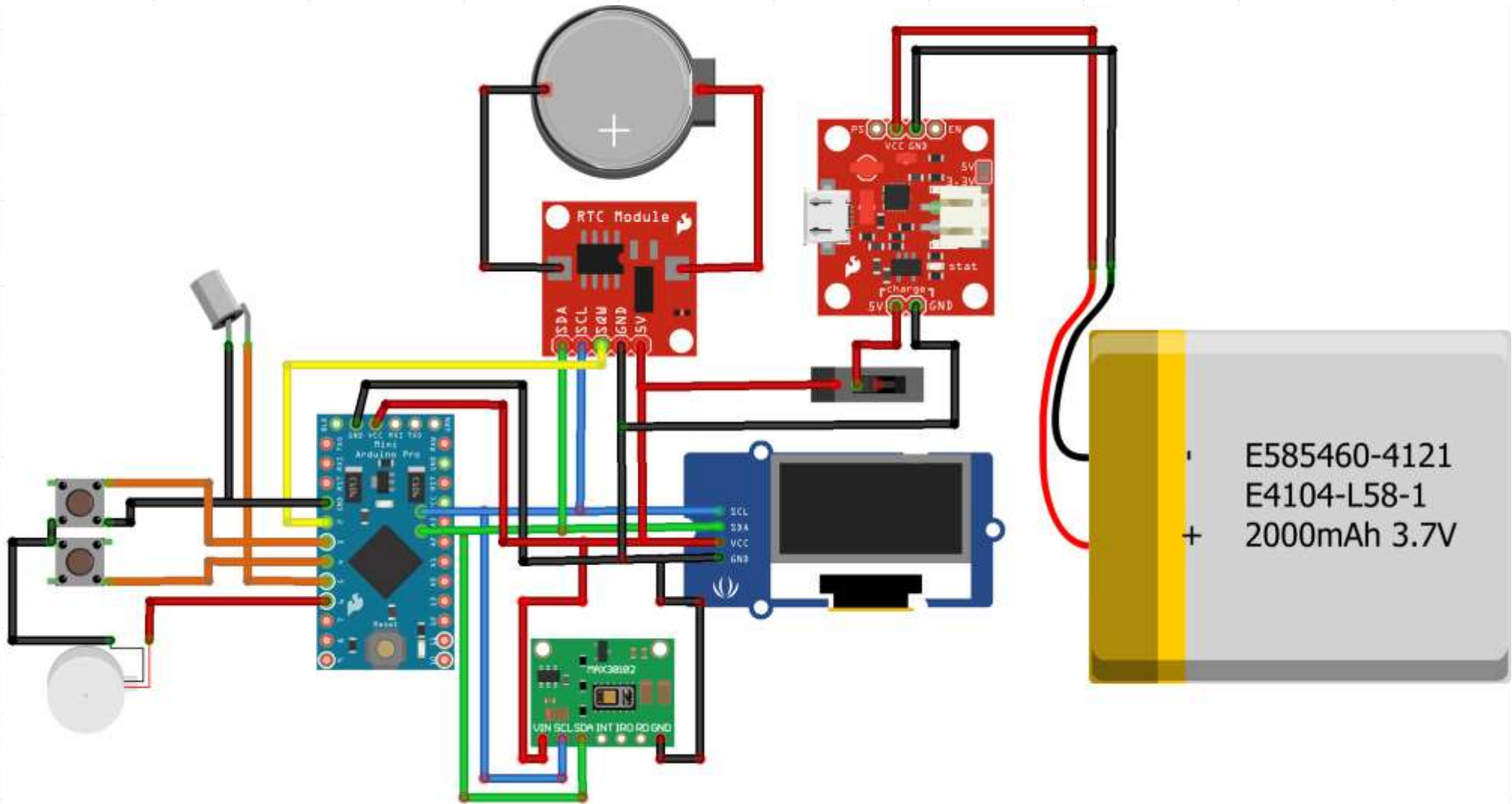


Components

- Arduino pro mini 5v
- DS1307 RTC
- Max30100 Heart rate sensor
- Lithium polymer battery
- TP4056 LiPo voltage regulator (charging module)
- Tactile buttons
- Haptic feedback motor
- 128*32 Oled display
- Cmos (3v battery) CR3032
- SW520D Tilt sensor



Circuit Diagram



Working

- Components are stacked on top of each other to save space.
- Smartwatch.h library made by me which helps, in the working of the project.
- The library has the all the functions which would be needed for the working. Uses time date class and updates every second through an interrupt.
- The software on the smartwatch can be updatable.
- The DS1307 will be working even when the battery dead. Consumes 200 micro amps
- The battery of the smart watch will work on always ON mode for 50+ hours because of its large 1500 mah battery
- The display is divided into segments (I call them pages)
- The user can configure upto 6 alarms, and apply a snooze when required.

Pages

- Basic functionalities



Time



Date

- Menu



Menu

- Alarm
- Set T/D
- Settings
- NA
- BMP/SPO
- Exit

Alarm pages



6 configurable alarms
(pre save alarm)



Setting the alarm



Alarm buzzes and the alarm number



Snooze feature
(+5 mins from alarm)

- Set time/date



Selecting time or date for setting it



Set time



Set date

■ Settings



Settings

- Motor (silent mode)
- Gesture
- Always on (display)

■ BPM\SPO2



BPM* - Beats per minute

SPO2 – Oxygen saturation level

Applications

- The user can use the watch to keep accurate time without worrying about discrepancies in the time or date, as the current time won't be lost and the auto leap year correction will be done automatically.
- The user can set alarms and snooze them when necessary.
- The user can track his or her, heart rate and blood oxygen level.
- The user doesn't have to worry about the battery life as the battery life is unmatched for any watch found in the market.
- There is a slot where the user can attach the FTDI adapter directly to the watch and update the software of the watch. (which is directly connected to the pro mini)

Future enhancements

- A Battery level indicator that tells how much battery is left. (working on soon will be updated)
- A Wi-Fi or a Bluetooth module that can connect with the smart phone and so that it can establish communication between it.
- Once the communication is established, the code can be made such that when the smart phone receives a notification or a call, the watch responds to it.
- The size of the watch could be made smaller so that the device looks aesthetic.
- Capacitive touch sensor could be used instead of using tactile buttons so that its easier to handle the watch.



References and libraries

- <https://www.arduino.cc>
- <https://www.google.com>
- GroverOled.h Library (small and lite library to handle the display operations of the oled screen)
- DS1307RTC.h library (library to get and put data into the registers of ds1307 rtc module)
- MAX30100_PulseOximeter.h (library to get the data from the on board pulse oximeter)