Bluetooth Smartwatch for Android - Bryan Smith

From MCS Wiki

Contents

Helios Watch Logo

Helios Watch Completed (No Case)

Project Description

This project, dubbed Helios (God of Day/Night Cycle or God of the Sun), is based off of Tortuga's RetroWatch (http://www.hardcopyworld.com/ngine/aduino/index.php/archives/670) to create a Bluetooth enabled smartwatch using an Arduino that will pair with your Android device. Once paired, the smart watch can not only display the time and date, but notifications from your Android device as well. Additionally, the the watch can also display RSS feeds and system information, all of which can be filtered from the Android app designed for the watch. Modifications from Tortuga's RetroWatch (http://www.hardcopyworld.com/ngine/aduino/index.php/archives/670) include, a larger screen, larger battery, vibration (haptic feedback), and internal battery charging while maintaining serial connection.

Features / Specifications

- **Processor :** ATmega328 – 3.3v(8MHz.
- **Memory :** 32KB Flash (2KB is shared for Bootloader), 2KB RAM, 1KB EEPROM.
- **Size :** Width x Height x Depth = 45mm x 43mm x 26mm (Without case).
- **Battery :** LiPo 500mAh(Around 24 to 25 hours of battery life.
- Pairs via Bluetooth to Android Device running custom App.
- Supports notifications, system info, RSS feed, and message filtering.
- Multiple clock faces.
- Haptic Feedback (Vibration)
- Open Source

Parts

Required

1. Monochrome 1.3" 128x64 OLED graphic display (http://www.adafruit.com/products/938)
2. Lithium Ion Polymer Battery - 3.7v 500mAh (http://www.adafruit.com/products/1578)
3. USB LiIon/LiPoly charger - v1.2 (http://www.adafruit.com/products/259)
4. FTDI Friend + extras (http://www.adafruit.com/products/284)
5. Vibrating Mini Motor Disc (http://www.adafruit.com/products/1201) (Optional)
6. JST-PH Battery Extension Cable (http://www.adafruit.com/products/1131) (Optional. Needed for the Male JST end. However, LiPo Charger may come with one.)
7. Arduino Pro Mini 328 - 3.3V/8MHz (https://www.sparkfun.com/products/11114)
8. HC-06 Bluetooth Module (http://www.amazon.com/gp/product/B00DQ4A7O6/ref=pd_lpo_sbs_dp_ss_2?pf_rd_p=1535523722&pf_rd_s=lpo-top-stripe-1&pf_rd_t=201&pf_rd_i=B008AW3KRK&pf_rd_m=ATVPDKIKX0DER&pf_rd_r=0386W6836PX92QNP24V1) or Bluetooth SMD Module - RN-42 (https://www.sparkfun.com/products/10253)
9. Resistor 10K Ohm 1/6th Watt PTH - 20 pack (https://www.sparkfun.com/products/11508)
10. Momentary Reset Switch SMD (https://www.sparkfun.com/products/8229)
11. USB Cable A/Mini B (http://www.adafruit.com/products/260)

Tools / Misc

- Soldering Iron
- Wire (Stranded 22 Gauge is what I used.)
- 3D Printer and ABS Filament in order to print the watch case. (Optional)
- Electrical tape.
- A solder removal device such as a Solder Sucker. (optional)

- A velcro strip or other material for the watch band.

Assembly

Part Preparations

Some of the parts will need to be prepared before we begin assembling. First is the display. The display from AdaFruit is able to perform in two modes, SPI and I2C. By default, the display is set to use SPI. However, for this project it is recommended to use I2C. The difference between SPI and I2C for simplicity sake, is that I2C requires less cables (but is slower). The speed is not an issue for this project and the less cables the better for space efficiency. Next is the battery. The cable attached to the battery from the factory is much longer than we will need. So I removed some of the cable to save room and make everything cleaner. Lastly, the FTDI Friend may come with a header installed on the board. This takes up a lot of room, so I removed it.

- **Display**
  - To set the display to I2C solder both SJ1 and SJ2 pads on the back of the display.
  - For more info, checkout the Adafruit page for this display - Here (http://www.adafruit.com/products/938)


Before.


After.

- **Battery**
  - I recommend removing some of the battery cable as it takes up precious space. I left about an inch or an inch and a half of cable left, however I could have taken off more and been okay. I still had some I had to wrap under the battery when assembling. Use the pictures below for reference.


Before.


After.

- **FTDI Friend**
  - I recommend removing the header from the FTDI friend as it takes up precious space. Use a Solder Sucker other method to remove the solder from the header to take it off. This guide will assume that you complete this step. It has not been tested with the header on. If the header is left on it may not fit in the case.


Before.


After.

- **Bluetooth Module**
  - Depending on the Bluetooth module you use, there may be a header that needs to be removed similar to the FTDI Friend above. If using the HC-06 it will more than likely come with a header already attached that will need removing. Use a Solder Sucker other method to remove the solder from the header to take it off.

Now that you have all of the parts prepared, we will begin assembling.

Power Supply Assembly

First we will build the Power supply. These steps are quite detailed in comparison to most of the guide as the placement and construction of this section is important. A wire diagram is below that you can use for reference.

- **Power Supply**
  - The power supply will handle most of the power related functions of the watch. The battery will be housed here and power will leave the power supply to go power the

Arduino, the display, etc. Also, battery recharging and serial data connection will be combined into one USB port for convenience.

1. First, solder five wires to the FTDI Friend with the wire stemming from the top. One for power, ground, transfer, receive, and reset. I cut these wires to be about 2 or 3 inches, they will be trimmed to the correct length later in the guide. Trim excess solder/wire from the bottom of the FTDI Friend. See image 1 below.
2. Set the FTDI Friend on top of the LiPo charger in an orientation that allows the LOAD,BAT,and STAT pads on the LiPo charger to still be roughly seen, while keeping the FTDI Friends USB port pointing out. This should put the pins of the FTDI Friend roughly across from the DCIN pins on the LiPo charger. Then bend the positive and ground wires over toward the DCIN pins on the LiPo charger. See image 2 below.
3. Next, wrap the FTDI friend in electrical tape. This will protect from short circuits because the FTDI friend will be setting on top of the LiPo charger next. Make sure to put electrical tape on the bottom side of the FTDI friend covering the pins you just soldered. See image 3 and 4 below.
4. Trim the positive and ground wires about a quarter of an inch after they pass the DCIN pins on the LiPo charger, then solder them into the corresponding DCIN pins on the LiPo Charger. See image 5 below.
5. Use some electrical tape to hold the FTDI Friend in place. In doing so you can also cover up the USB port of the LiPo charger (NOT the FTDI Friend) so you won't get confused. Power will be sent to the LiPo charger through the FTDI Friend now. Make sure to not cover up the JST battery ports on the other side of the LiPo charger though. See image 6 below.
6. Finally, plug in the battery to the LiPo charger (the left JST port if you have the ports facing you) and place the battery on top of the FTDI Friend, curling up any extra cable (hopefully not much if any extra left from the prep step) underneath the battery. Use electrical tape to secure the battery in place. See image 7 below.

- This completes the power supply for now.

1) Solder on the wires. (Power, ground, transfer, receive, reset)

2) Place on top of LiPo charger and position power wires.

3) Top view of electrical tape wrapped FTDI Friend.

4) Bottom view of electrical tape wrapped FTDI Friend.

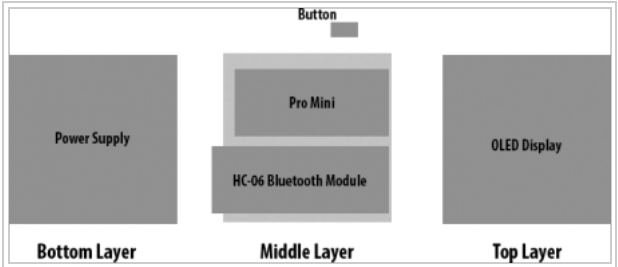
5) Trim power wires and solder to DCIN pins on LiPo Charger.

6) Tape FTDI Friend to secure it to the LiPo Charger.

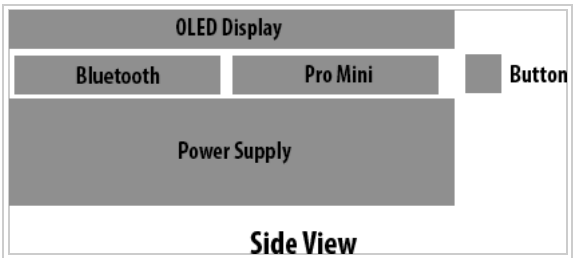
7) Plug in the battery, then secure it to the top of the LiPo charger and FTDI Friend with electrical tape.

Watch Assembly

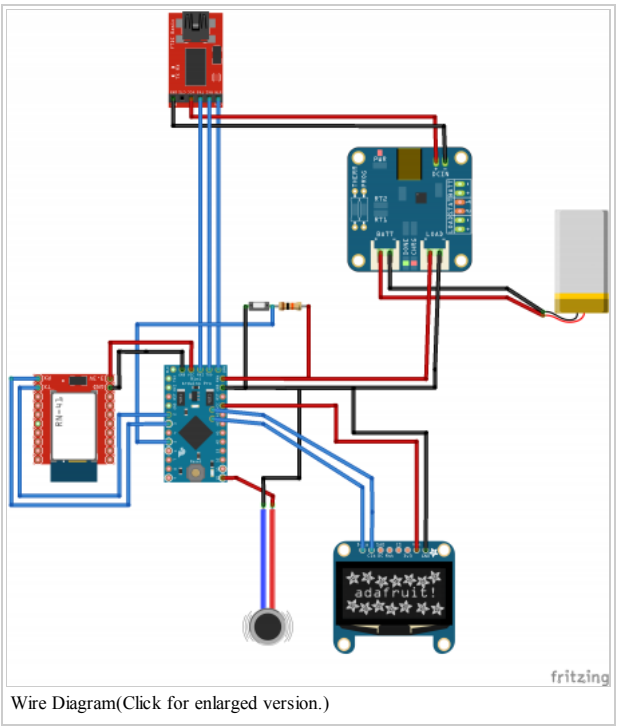
Now that the power supply is done, the rest of the watch can be put together. The majority of the work that is left is the straight forward soldering of wires to the correct pins that are laid out in the diagrams below.



A Top Down View of the 3 Watch Layers



A Side View of the 3 Watch Layers



Wire Diagram(Click for enlarged version.)

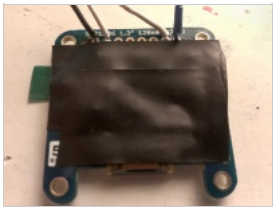
- **Recommended Steps to Assemble (Use diagrams for reference):**
  1. Read the entire guide. Especially the Additional Notes below. Get familiar with the diagrams and layouts.
  2. Power Supply (See above.)
  3. Connect Bluetooth to Arduino
  4. Connect Button to Arduino (Only its digital pin)
  5. Connect the Display to the Arduino
  6. Set the components on top of the power supply. **Make sure the USB port for the power supply is facing you. This also means the battery ports are on the right.**
  7. Combine the Display, Button, and vibration motor power and ground wires with the batteries power and ground.
  8. Connect the combined power and ground to the Arduino.
  9. Place the vibration motor anywhere it will fit. Such as on the side or under the display.
  10. Use electrical tape to hold down the display to hold all of the parts together with the power supply.

- **Additional Notes**
  - Make sure to use tape on the top and/or bottom of some parts to keep them from shorting each other out. Such as the bottom of the display.

-
  - **Connecting Arduino Pro Mini and Bluetooth Module**
    - BT -> Arduino Pro Mini
    - VCC -> VCC

- GND -> GND
- TX -> Digital 2
- RX -> Digital 3
- (Must use pins digital 2 & 3. Do not change. These are special external interrupt pins for the Arduino)

- **Connecting Arduino Pro Mini to OLED Display (I2C)**
  - OLED -> Arduino Pro mini :
  - GND -> GND
  - VIN -> VCC
  - DATA -> Analog 4
  - CLK -> Analog 5
  - (Must use pins analog 4 & 5. Do not change. These are special SDA and SCL I2C communication pins for the Arduino)



Make sure to cover parts with tape to keep them from shorting out since they sit atop one another. As seen here with the back of the OLED Display

Code Base

You can download Helios Watch Arduino source on GitHub.

Helios Code (Github) (https://github.com/BryanSmithDev/HeliosWatch)

Click "Download ZIP" to download and unzip it. The Source Code is divided into folders, one for Arduino and the other for the Android App. Before you will be able to compile the Arduino source you may need to install some drivers and libraries.

Installing FTDI Drivers (Windows & Mac Only)

Windows and Mac OS required drivers in order to communicate to the Arduino through the FTDI friend. Linux has the drivers built in.

Download the drivers here (http://www.ftdichip.com/Drivers/VCP.htm) .

If you need more help installing the drivers, go here (https://learn.adafruit.com/ftdi-friend/installing-ftdi-drivers) .

Installing Graphics & Display Libraries

In order to draw the images, shapes and fonts on the OLED Display you need a graphic library. You must install Adafruit_SSD1306 display library (if using the display listed in Parts) and the Adafruit-GFX-Library. Click "Download ZIP" button on the down-right in the link to download and copy the library folder to libraries folder for your Arduino IDE. If you need help installing Arduino IDE Libraries, click here (http://arduino.cc/en/Guide/Libraries) . (You may need to rename the folders when placing them in the Libraries folder. characters like '-' are not allowed in the folder name. Simple replace them with '_')

(According to your development environment, Adafruit library conflicts with Robot_xxx library. In this case, backup and delete Robot_xxx libraries from Arduino library folder.)

- Adafruit_SSD1306 Library (https://github.com/adafruit/Adafruit_SSD1306)
- Adafruit-GFX Library (https://github.com/adafruit/Adafruit-GFX-Library)

Compile / Upload

If you completed the steps above correctly, you should now be able to compile and upload the source code to the Arduino.

- Steps to Compile and Upload.
  1. Open HeliosWatchArduino.ino in the Arduino IDE
  2. Plug in the watch.
  3. Set the Board to Arduino Pro Mini 3.3v 8Mhz ATmega328 (May vary in different IDE versions)
  4. Select the Port number representing the watch.
  5. Press Upload

If everything worked correctly, you should soon seen the display come on and show a clock at 0:00. It is now ready for pairing with the Android App.

Android App

Most of this information is derived from Tortuga's RetroWatch Guide (http://www.hardcopyworld.com/ngine/aduino/index.php/archives/670) and modified to fit Helios Watch's modifications and for improved English readability.

Helios Watch Android Source code is included on the GitHub along with the Arduino Source code. If you downloaded the source code here then you already have the Android App source code. It is located in the HeliosWatch_Android folder. However the source code is not needed unless you need to modify the app due to changes you made in the watch assembly process (e.x. used a different pin for the Vibration motor.). You only need to download the app to you phone from below.

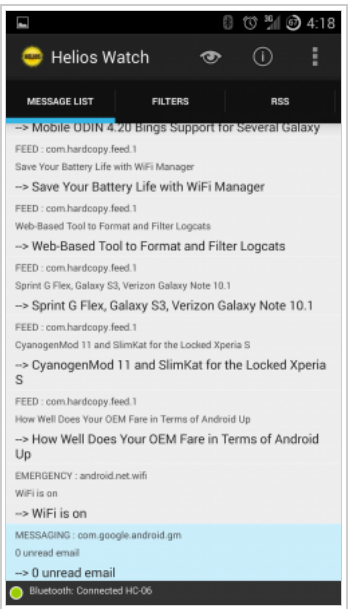
- **Download the Helios Watch App here:**
  - Helios Watch (Android 4.3+) (https://github.com/BryanSmithDev/HeliosWatch/raw/Helios/HeliosWatch_Android/HeliosWatch.apk)
  - Helios Watch LE (Android versions lower than 4.3) (https://github.com/BryanSmithDev/HeliosWatch/raw/Helios/HeliosWatch_Android/HeliosWatchLE.apk)

- **App Tabs:**

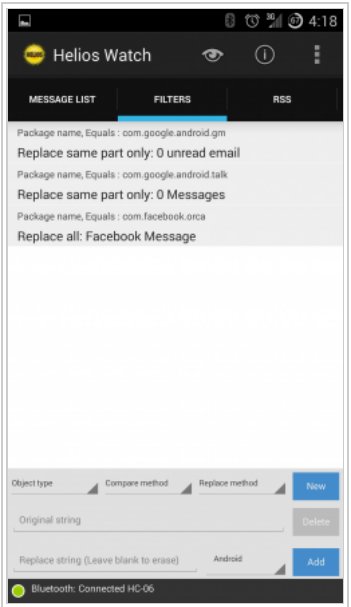
- **Message List Tab:** Messages are the information collected from the app. Every message is inactivated except for emergency messages. Inactivated messages are not sent to the watch. By touching a message, you can create a filter for that message or messages from the same package.
- **Filter Tab:** The Helios Watch app controls all notifications sent to the watch by filtering. You can add, edit, and delete filters in this tab. it also allows string manipulation to edit what shows up on the watch compared to what was given from the notification.
- **RSS Tab:** You can add, edit, and remove RSS feeds here. You can show notifications on the watch based off of RSS feeds.
- **Watch Control Tab:** Allows you to change watch faces, and notification indicator viability, etc. If you put your Gmail account here, unread messages are added in the message list.

- **The Helios Watch app collects three kinds of data:**
  - **Notification:** A notification that is registered in the Android Status Bar. The app uses the Notification Service to collect notifications (Only Android 4.3 or higher). If your are using Android versions lower than 4.3 then use the Helios Watch LE app.
  - **System info:** The system info of the phone such as battery status, mobile data connection status, WiFi status, and recharging progress. In addition, if you register a Gmail account, it counts unread e-mails.
  - **RSS feed:** If you set RSS title and URL, it periodically read RSS data and sends it to the Message List.

- **Other minor features:**
  - Up to 65 icons available to use for notifications on the watch.
  - All information is updated to the watch every 30 minutes.
  - The Helios Watch service runs in background even if you shut the app.

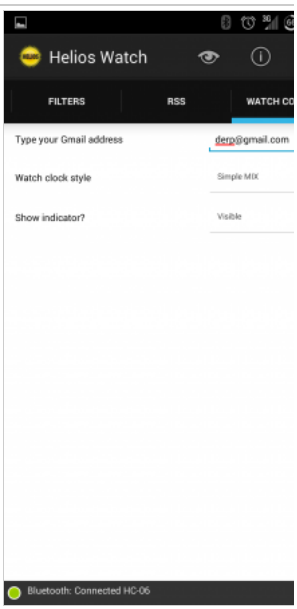


Message List Tab. This allows you to find and select notifications to send to the watch via filters.



Filters Tab. This allows you to create, edit, or delete filters. Filters control what is displayed on the watch and where the data comes from.

RSS Tab. RSS Feeds are added here to allow them to show up in the Message List.



Watch Control Tab. Allows you to ch watch faces, and notification indicato viability, etc.

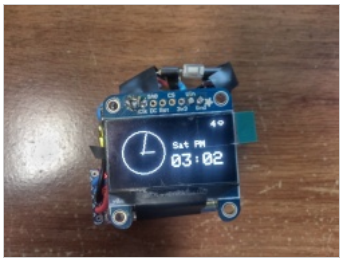
For a long in-depth walkthrough of using the app, read Tortuga's RetroWatch App User Guide (http://www.hardcopyworld.com/ngine/aduino/index.php/archives/722) . It still applies to Helios Watch.

Usage

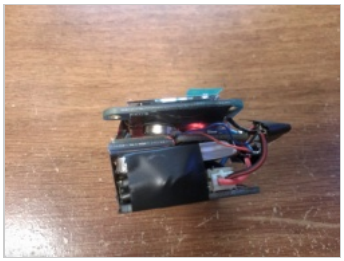
Once you have the code uploaded to the Arduino and the Android App installed on your device, you are ready to use the watch. Turn on bluetooth on your device and pair with the module (If it asks for a pin try 1234 or 0000). Next, start the Helios Watch app. If on Android 4.3+ press the second icon in the action bar, which is the 'i' with a circle around it. Then make sure Helios Watch is selected and then hit the back button. Then choose the first action bar item and select the bluetooth module that is already paired. If you can't find it, check the power and the bluetooth module. Or choose "device scan" menu to scan manually.

If the connection succeeded, you should see "Connected" in connectivity the display area at the bottom. Touch the action bar menu(…) and choose "Force Watch Sync". This will force the phone data to display the time and messages on the watch. You are now all set up and can start making filters to show your notifications.

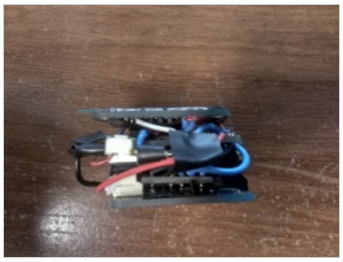
Below are some pictures of my completed watch:



Top View of my Helios Watch


Right Side of my Helios Watch


Left Side of my Helios Watch


Top Side of my Helios Watch

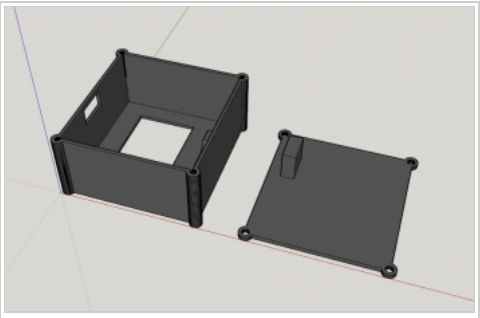

Bottom Side of my Helios Watch

My Helios Watch

3D Printed Case

The case for Helios Watch may be 3D printed and using hot glue or other adhesive to attach velcro to the back for the watch band. Due to the fact that all Helios Watches made will vary from person to person, since we are not robots mass producing these with perfect accuracy, I cannot guarantee that the case will fit your watch. So some modification of the case may be required. The STL file for 3D printing, and the SketchUp 3D file for if you need to make your own modifications, are included in the GitHub Repository (https://github.com/BryanSmithDev/HeliosWatch) in the documentation folder. They are also linked below.

- **Downloads:**
  - Helios Case - STL (https://github.com/BryanSmithDev/HeliosWatch/raw/Helios/HeliosWatch_Documentation/Misc/HeliosCase.stl)
  - Helios Case - SketchUp File (https://github.com/BryanSmithDev/HeliosWatch/raw/Helios/HeliosWatch_Documentation/Misc/HeliosCase.skp)


The Helios Watch Case Model in SketchUp


The Helios Watch Case


The Helios Watch Case

Retrieved from "http://www.mcs.uvawise.edu/wiki/index.php?title=Bluetooth_Smartwatch_for_Android_-_Bryan_Smith&oldid=2711"

- This page was last modified on 6 May 2014, at 22:53.