OpenWatch getting started guide

Written by Danny Grob

Contents

- Introduction
- Making the PCB
- Soldering tips
- Connecting to the PCB
- Setting up the software
- Copying bitmap files

Introduction

This project is based on the Dialog Semi SmartBond DA14683 microcontroller. It is an ARM-Cortex-M0 processor with BLE 5.0 build-in.

Making the PCB

You can order a pcb from a manufacturer of your choice, like jlcpcb or pcbway. It's recommended to order a stencil too.

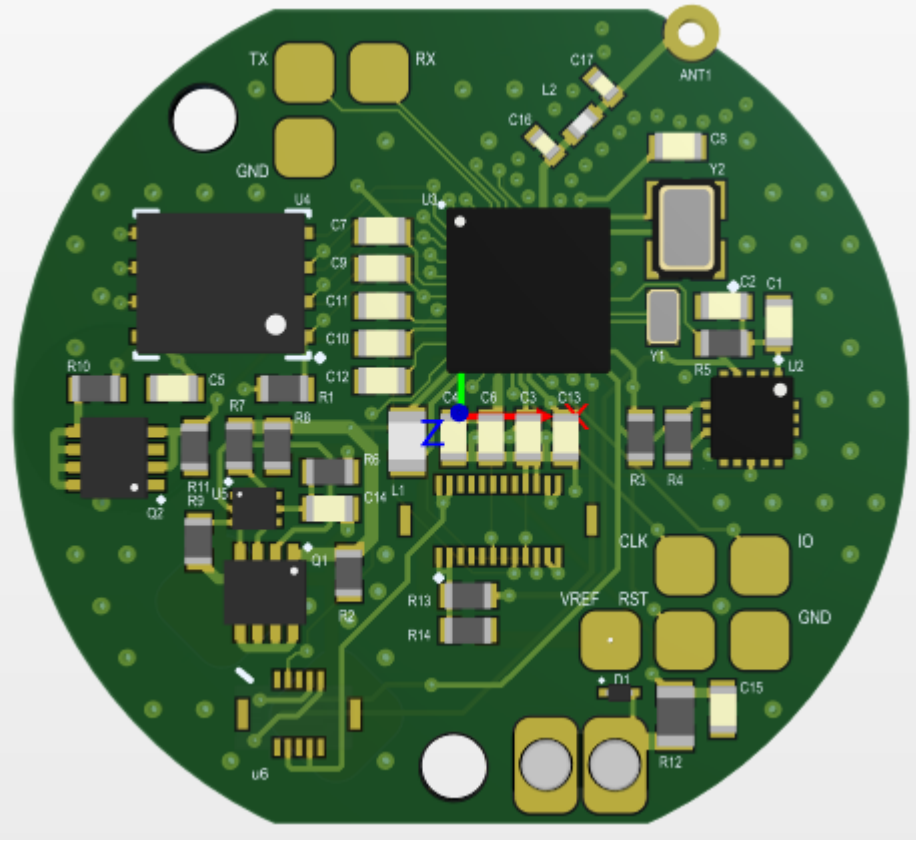
To make the pcb you have to upload a zip file that includes the Gerber and NC Drill folders. You can setup the parameters like the guide below

Board type :	Single PCB	**Panel Way :**			
Size :	33 x 30 mm	**Quantity :**	50	**Layers :**	4 Layers
Material :	FR-4 TG150	**Thickness :**	0.8 mm	**Min Track/Spacing :**	4/4mil
Min Hole Size :	0.2	**Solder Mask :**	Matt black	**Silkscreen :**	White
Gold fingers :	No	**Surface Finish :**	Immersion gold	**Via Process :**	Tenting vias
Finished Copper :	1 oz Cu (Inner Copper:1 oz)	**Additional Options :**			
Create Time :	2019/2/18 11:57:57	**Build Time :**	4-5 days	**Estimated Finish Time :**	2019-02-23 China Time Zone(GMT+8)
Manufacturing :	Layer Order:L1:smarchWatch_V5.GTL,L2:smarchWatch_V5.G1,L3:smarchWatch_V5.G2,L4:smarchWatch_V5.GBL.				

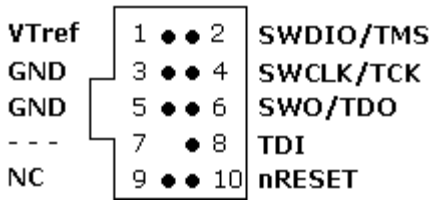
Soldering tips

[todo?]

Connecting to the pcb



Connect the VBUS and GND to a 5v supply. This is needed so the mosfet opens and starts up the power. Also connect your battery. For debugging it's nice to have an on-off switch in line to reset. You connect the JLink pins to the pcb. Either create a custom connector for this to speed up the development time, or just solder it on.

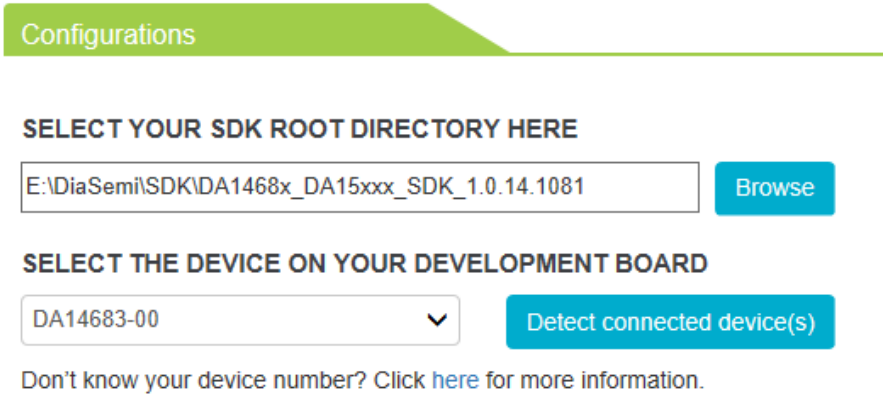
	VTref SWDIO/TMS SWCLK/TCK TDI nRESET GND	VREF IO CLK - RST GND

To have debugger output you can use a serial to usb converter and connect the TX and RX (and GND).

Now the smart snippets studio should recognize the microcontroller.

Setting up the software
Download the Dialog Semi DA14683 SDK Install SmartSnippetsStudio found here, scroll down for your platform. Follow the getting started guide found here

Start the smart snippets studio after you've installed it. Make sure that you have the right SDK selected..



When clicking 'detect connected devices' your pcb should show up if it's connected right. If not, check your connections, and make sure the pcb is powered from battery and 5v on the charge pins.

Project files
Checkout the git: https://github.com/dannygrob/OpenWatch. You can download everything as a zip file, or clone it.

Copy or move the 'Software' folder to (this is for the sdk references):

[install folder]\SDK\DA1468x_DA15xxx_SDK_1.0.14.1081\projects\dk_apps\ble_profiles

Click the 'Import test projects in workspace' button and select the "software" folder you just created. From the guide linked above you should already have imported the 'scripts' folder. This is needed to upload the firmware.

The files in the 'SDK Files' folder should be copied in their right locations. The base folder is: *E:\DiaSemi\SDK\DA1468x_DA15xxx_SDK_1.0.14.1081\sdk\bsp*

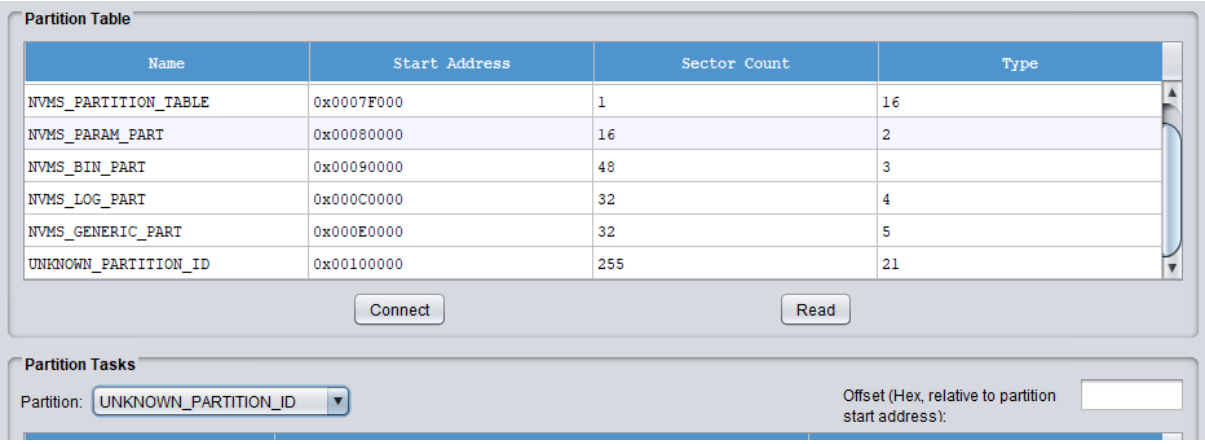
Everything should follow the same structure, so for example platform_devices.h should go into sdk\bsp\adapters\include\. This overwrites the old one, you can make a backup if you want or just transfer the changes (some devices where added)

You are now ready to build the project and upload it to the board. You can upload it using the 'program_qspi_jtag' script. It should upload but the graphics are still missing.

Copying the bitmap files

The bitmap files were generated using Samson March's Bitmap tool. These need to be uploaded into the right partition on the board's flash ic. This can be done using the "SmartSnippets Toolbox". Select your board from the middle column (if it's not there it's not connected) and press 'open'.

Press the 'connect' button below the partitions section. After it connected correctly press 'read'. It should now show something like this:



Make sure that there is an 'UNKNOWN_PARTITION_ID'. If it is not there you either did not copy the "partition_table.h" file to the 16M folder in config in the SDK, or the program did not run before. The code needs to run on the board 1 time to setup the partitions. Select the UNKNOWN_PARTITION_ID in the partition tasks section.

Now select the "pictureFiles.bin" file from the bitmaps folder in the 'select a file to burn or verify against' section on the left of the screen. Press the 'burn' button on the bottom of the middle section to burn it to the flash. It should finish without errors.

You can now connect the display if you did not already and reset it. To reset without a reset button you can use 'Segger Ozone' and select the jlink programmer on SWD and load the 'bin' file that was generated when you compiled the code. This is not needed on the USB dev kit as it has a reset button.