

Author: Leon Kozinakov

Date: 3/14/2019

Aquamote Software Setup Guide

Before you can begin development on Aquamote, you will need to setup your development machine appropriately. There are multiple parts to this setup targeting the different components of the project, both software and hardware. The whole project was done on a Windows 10 machine, so that is the only OS we support. Although other operating systems are not impossible to use for this project, we don't provide the guides for them.

1. **Printed Circuit Board Design:** The entire PCB design was done in Altium Designer 17, however the output files on the Github repo should be portable to newer versions of Altium or other software suites such as Allegro.
2. **CAD Design:** The waterproof enclosure was designed in Autodesk Inventor 2018 Professional, and exported to the .STL format to be 3D printed on a Form 2 machine. Once again, alternate CAD tools such as SolidWorks could be used to modify the design.
3. **Firmware Design**
 - a. **Main Firmware:** The firmware that runs on the CC2650 SoC was written in C using Code Composer Studio (CCS) v7.3 from Texas Instruments (TI). CCS is basically derived from the popular Eclipse IDE, but is customized for TI products.

Step 1: Download the .zip file containing the software installer from TI and extract it anywhere on your computer. [link](#)

If TI has removed the download link above, which happens often when they stop support for products, then use this alternate [link](#)

During the installation process leave the location default (**C:\ti**) and make sure you check “**Simplelink CC13xx and CC26xx Wireless MCUs**” for the supported families. Leave everything else as default.

Step 2: After installing CCS v7.3, you will need to install the TI ARM Compiler version 5.2.6. Launch CCS and go to Help > Install New Software... Then, from the drop down list, select Code Generation Tools Updates and then below check TI Compiler Updates > ARM compiler

Tools 5.2.6

Make sure "Show only the latest versions of available software" is unchecked.

Step 3: Now you need to install BLE SDK 2.2.1 since it is a dependency for our Aquamote application code. [Link](#)

Follow the setup wizard and leave the installation location default ([C:\ti\simplelink](#)) The wizard will also install **TI RTOS 2.20.01.8** which is also a dependency, so leave it's path default as well ([C:\ti](#))

Step 4: Once the SDK has finished installing, copy over the BLE_SDK/src folder from the Aquamote Github repository and overwrite [C:\ti\simplelink\ble_sdk_2_02_01_18\src](#)

Step 5: Run CCS as administrator. On bootup, it will ask you for your workspace. Since you probably don't have one yet, go ahead and create an "Aquamote_Workspace". After the environment opens up, you will go to File > Import and select **Code Composer Studio > CCS Projects**. Hit "Next" and then "Browse" to select a search directory. Navigate to [UnderwaterSensorTag/Aquamote/Firmware3.x/<workspace>/ble_example s.../examples/cc2650lp/<project>/ccs](#)

After selecting that path, you should see two projects pop up, an app and a stack. Check them both and hit "Finish"

You should now be ready to begin development, however to verify the process is correct up to this point, first compile the stack, and then compile the app. They should both build cleanly. You can also try flashing the firmware, once again stack then app, onto a CC2650 launchpad board (if using the client workspace) or onto an Aquamote device (if using the aquamote workspace)

NOTE: The firmware was developed using example code from TI's Github. If you want to browse other examples, or look at the original code, see this [link](#).

NOTE: The projects that are imported in the above steps are all done

through .projectspec files. These are XML style text files that can be modified to control the settings of the project. More info [here](#).

- b. *Sensor Controller:*** The code that goes on the auxiliary sensor controller on the CC2650 SoC was developed using TI's Sensor Controller Studio SCS.
- 4. **Post-Processing Work:** Most of the post-processing work was done in Python scripts and Anaconda via the Jupyter notebook system. See the link below to install Anaconda (which installs Python as well).
<https://docs.anaconda.com/anaconda/install/windows/>