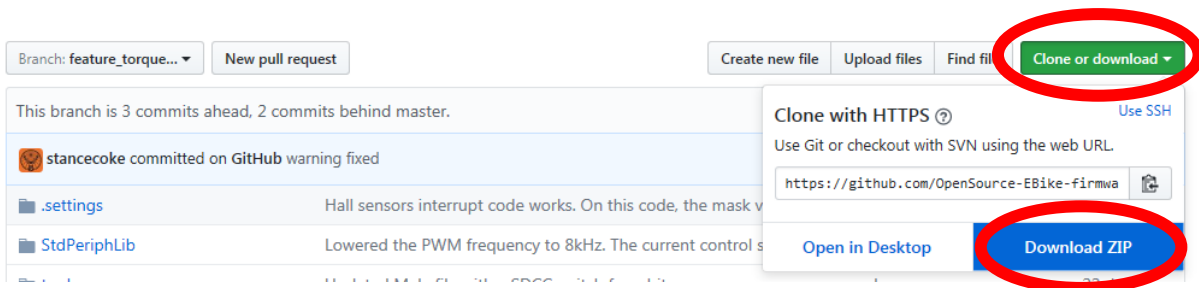




Tutorial for setup toolchain in windows environment

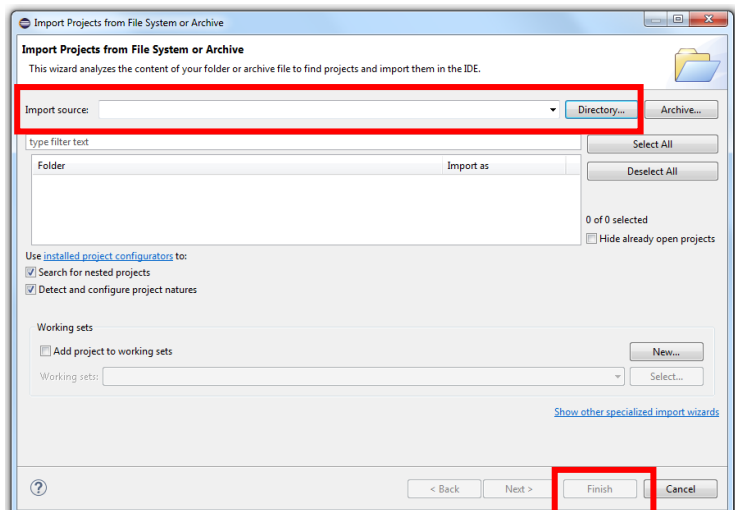
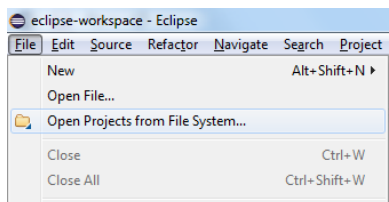
1. Download sdcc for windows according to your system (32 or 64 bit) and install it.
<https://sourceforge.net/projects/sdcc/files/>
2. Download the branch of the repository by clicking "Clone or download" → Download ZIP
https://github.com/OpenSource-EBike-firmware/BMSBattery_S_controllers_firmware/tree/feature_torque_sensor




3. Extract zip file to a suitable folder e.g. directly to c:\
In this stage you are already able to compile the project:
 - open an explorer window (press the windows key and "e" simultaneously)
 - navigate to the project folder, you just created
 - double click "Start_Compiling"
 You can edit the project using any editor like Wordpad
I recommend installing eclipse for editing.

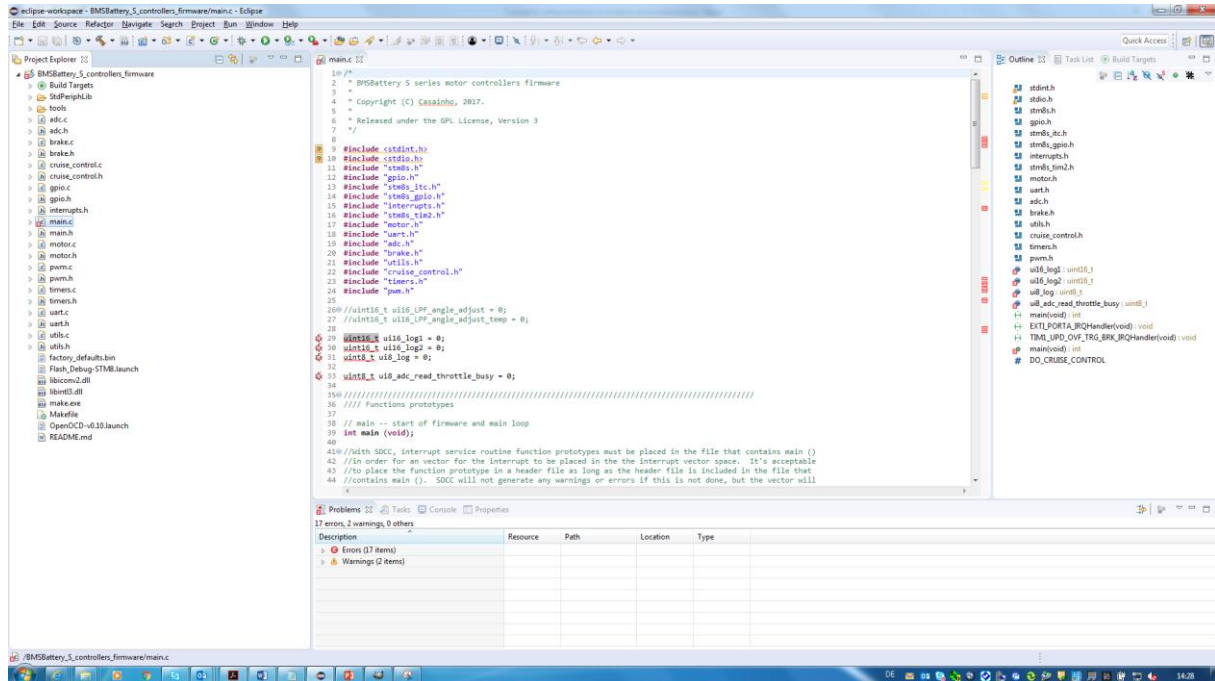



4. Download and Install the java runtime environment (if not already installed)
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
5. Download eclipse for c/c++ developers
<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/oxygenr>
Extract zip file to a suitable folder e.g. directly to c:\
Note: there's no installer, just unpack and double click on "eclipse.exe" in an explorer window.
6. Start eclipse and set workspace directory (you can use the default setting)
Click on "File" → "Open Projects from File System..."
Choose the directory, you extracted the files from github to and hit "Finish"

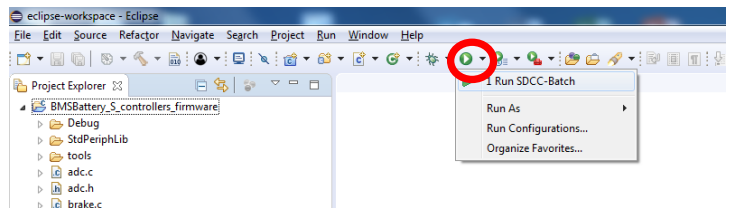




7. Open the project explorer by clicking the small symbol  in the upper left corner.

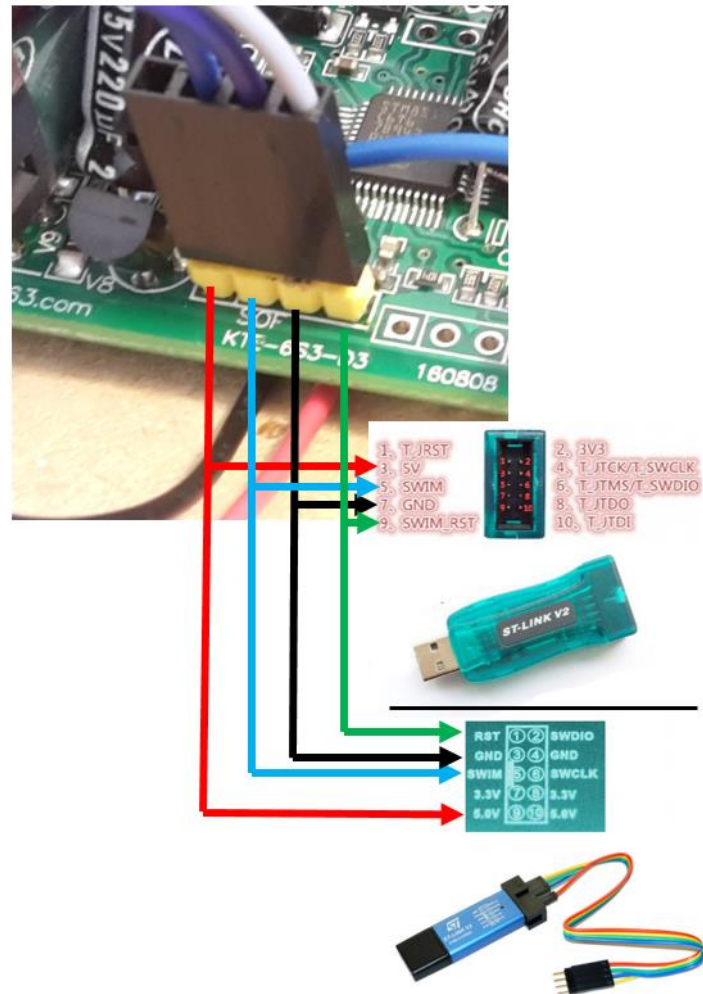


To start compiling press the “Run” button.  Starting the first time, you have to choose “Run SDCC-Batch” from the “Run”-dropdown menu. At my installation, there was an error message, just ignore and hit “Run” again. You will see the batch running in the console-window of eclipse.

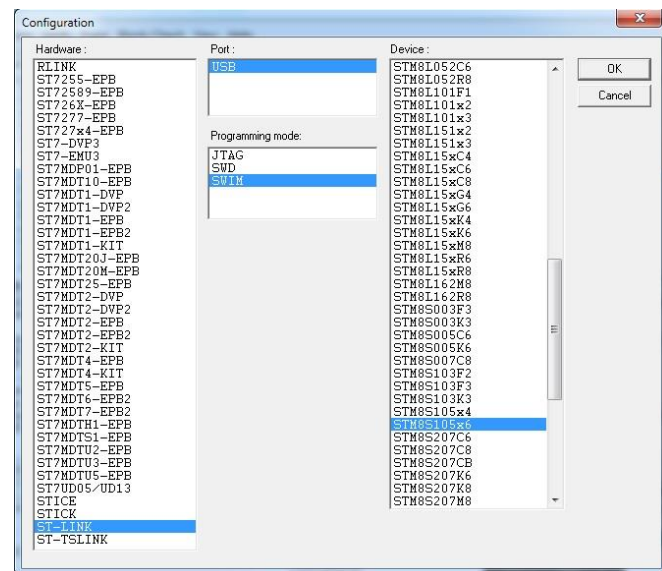




8. Prepare the wiring for flashing the controller as shown in the picture (wiring for two different clones of ST-Link V2 easily available at ebay/amazon/....)
I connected all 4 lines to the ST-Link V2s associated pins. You don't have to power the controller with 36V for flashing in this case.



9. For flashing the controller download and install the ST visual programmer
<http://www.st.com/en/development-tools/stvp-stm8.html>
Start ST visual programmer. In hardware settings choose ST-Link as programmer and STM8S105x6 as target.





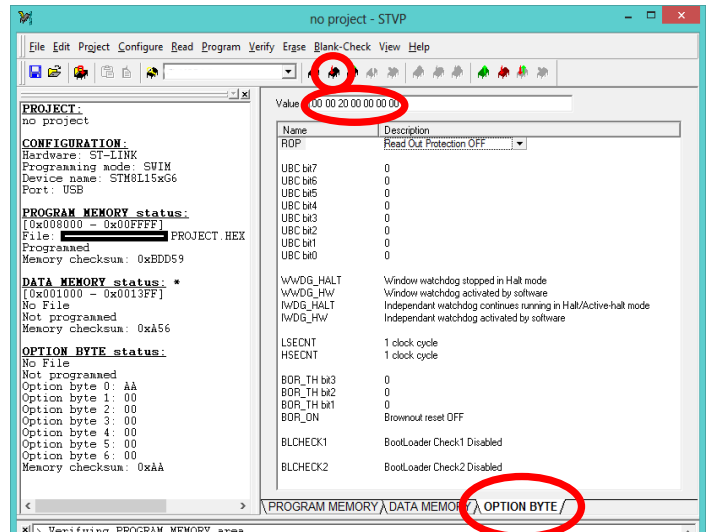
10. For the first time flashing the controller you have to clear the write protection byte and set the right timer options:

Go to tab "OPTION BYTE"

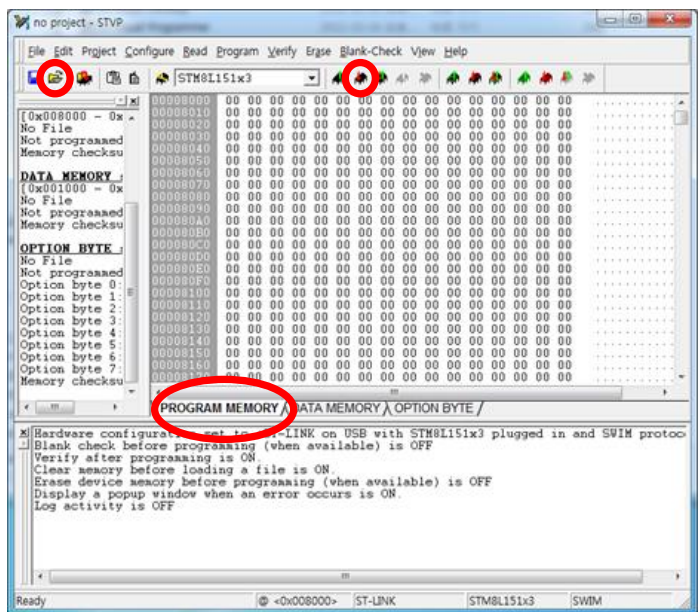
manually edit the "Value" line to

"00 00 20 00 00 00 00 00" and hit the icon

Caution: this will delete the original firmware of the controller and can't be undone



11. For flashing the open source firmware go to the "PROGRAM MEMORY" tab, "File" → "Open" and navigate to your source directory. Choose the file main.hex and hit the icon





Notes:

It is not easy for a newbie to deal with a command-line based toolchain. Here are some basics (as far as I understood after several hours of reading and testing, as I'm a newbie in this myself):

The code itself is stored in simple text files, you can edit them with any text-editor.

To get the file (in this case a "[hex-file](#)") that you can flash to the controller, you have to compile each single part (each .c file) and then link them together. Don't worry, this work is done automated☺.

To tell the automated "build"-process what to do, there is the file "makefile". You don't have to edit this file, as long as you don't add new .h or .c files to your project. The makefile in the downloaded branch just creates the hex-file (main.ihx renamed by the "Start_Compiling"-batch to main.hex to make it visible for the ST visual programmer in default settings). There is no support of debugging tools and flashing in this setting.

To create the hex-file you just have to open a cmd-window and go to your project folder, you need the old fashioned dos-command "cd" to navigate through the file system. Then just type "make" and hit return. I copied the necessary make.exe and .dlls to the repo (taken from Cygwin), so you don't have to install them manually. Instead of this you can just double click the "Start_Compiling.bat" or hit the "Run"-Button in eclipse.

The launch information for calling the Start_Compiling.bat in eclipse is located in the .settings folder.

For flashing via command-line, you can use the open source command-line tool [stm8flash.exe](#) (doesn't work with 32bit Windows), or the one from ST that comes with the visual programmer installation, but I've not tried it yet.

For flashing with stm8flash you can reactivate the line in the makefile by deleting the #. I prefer the graphic tool for flashing.