

Merging

This short chapter aims to provide information on the current status of the project in respect to the main repository from ruuvi.

The latest version the project has been updated to is version 3.31.1 in march 2022. There has not been a complete alignment as we have made independent changes to some files. These would have been partially overwritten by a strict update. Therefore, when merging, it was decided that changes we made to the project would always take precedence over changes from the root repository.

Caveates

This project is heavily reliant on sub-modules. If you do not know, how to work with these it is highly recommended to get acquainted with them. Especially when changes need to be made to them.

A great startingpoint can be found at these links:

<https://git-scm.com/book/en/v2/Git-Tools-Submodules>

<https://www.atlassian.com/git/tutorials/git-submodule>

Basically a sub-module is a reference to any other repository at a certain commit in that repository. Or to put in other words it is a snapshot of this repository.

Sub-module workflow

Sub-modules do have their own branches and commit histories. When you are working on a sub-module you need to do two things, to make the changes permanent.

- Commit and **push** the changes **in** the sub-module.
- Update the parent repository to point to the latest commit of the sub-module.

A Common error is forgetting the first step. If you just push changes made to the parent repository, any other developer who would pull this changed code will get an error. Their newly pulled code would point to a commit in the sub-module which it can't pull because we didn't push it.

A general advice on merging in a project like this: When you are ready to merge your feature into the master branch, it is best practice to merge the master into your projecdt branch. This way if the code is failing to build the error is contained to your feature branch and does not affect the master on which the other participants are relying on. When you have fixed the error resulting from the merging in your branch, you can then merge your branch into the master branch.

DONT'T DO : feature => master

DO : master => feature => master

Fixing large errors on master is no fun and should be avoided. The level of panic is much lower when you are working on your own branch.

But don't panic even if things break.