

## Embedded Software Lab

# Lab 4 - Project Documentation

Christine Jakobs, Martin Richter

In this lab, you will not produce new code, but establish a sustainable documentation for your coding project.

### Initial Remarks

The tasks should be solved in the given order. If you do not manage to solve all tasks in preparation of the next lab meeting, make sure that you solve them afterwards. Your final grade at the end of the semester depends on the availability of *all* solutions in your Git repository.

The presentation of your results during the lab slot is oral. There is *no need for preparing slides* or any other kind of written material.

Be ready to answer questions. Have positive and negatives experiences and according documentation available.

## Task 1

Modularize your software parts and prepare header files for them in the project. A good way would be to have one module for each of the following parts:

- UART communication
- CRC calculation
- IRQ handling
- Layer 1
- Layer 2
- Layer 3

## Task 2

The folder structure of your project should reflect the module structure. Have a look at other software projects on GitHub that have C as main language. Integrate your code from old lab sheets and adjust your Makefile accordingly, so that you form a complete software project that can be built and deployed as one.

## Task 3

Documentation is the key of an sustainable software project. Besides code documentation, a written architecture description is needed for making your work understandable by others. For our project, use Doxygen for the following tasks:

- Add inline documentation comments in your code (see <https://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>).
- Add a general explanation of your architecture in a separate README.md file. It should be formatted in Markdown syntax (see <https://www.stack.nl/~dimitri/doxygen/manual/markdown.html>).
- Generate an HTML version of your overall documentation with the Doxygen command-line tool (see [https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg\\_generate\\_html](https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_generate_html)).

## Task 4

Implement a new target in your Makefile. A call of `make docs` should trigger Doxygen to generate the overall HTML documentation from your code and the README.md file. Make sure that you adjust your `.gitignore` file appropriately. A call of `make clean` should remove all generated files.

Feel free to add more details to your project documentation, such as table of content, pictures, performance results or hints for installation.