Generating Documentation

Pieter P

Generating the documentation using Doxygen

As mentioned before in the style guide, the library is documented using Doxygen. It will parse the source code, create class diagrams, calling networks, overviews of classes and functions, based on the comments in the code. All of this results in a folder of HTML files you can open in the web browser.

Generating the documentation in Visual Studio Code

Press **CTRL+P**, and type "task", space, and then "Generate Documentation".

Generating the documentation from the terminal

Navigate to the `Control-Surface/build` directory, and run:

```
$ cmake --build . --target documentation
```

Generating the coverage information using LCOV

It is important that as many lines, special cases, and functions are covered by the unit tests.
GCC has a tool called GCOV to check what lines of code are executed. LCOV is a front-end for GCOV that exports the results as HTML pages.

Generating the coverage information in Visual Studio Code

Press **CTRL+P**, and type "task", space, and then "Generate Coverage Information".

Generating the coverage information from the terminal

Navigate to the `Control-Surface/build/coverage` directory, and run:

```
$ cmake --build . --target coverage
```