

Enforcement

This page documents how static analysis and autoformatting tools can be used to enforce the Zubax C++ coding conventions.

- 1 clang-format
 - 1.1 CI integration
- 2 clang-tidy
- 3 Legacy
 - 3.1 Eclipse IDE autoformatter

clang-format

Unlike AStyle or Uncrustify, clang-format doesn't tend to break your code or introduce nonsensical changes, so it is the recommended option. The configuration file below requires clang-format version 8 or newer. Note that clang-format can't enforce brace placement and some other things; consider using **clang-tidy** for that.



Put the config file in the sources' root directory, then run from the same directory:

Invoking clang-format

```
clang-format -i -fallback-style=none -style=file src/*.c
```

CI integration

Enforce a consistent coding style using CI by running clang-format against a fresh working copy, and then ask git if there were any changes introduced. If there are no changes, the codebase is conforming, otherwise it's not and the build should be failed.

Example

```
# Clone the repository and cd into its directory; don't forget git
submodule update --init --recursive
all_source_files=$(find $SOURCE_DIRECTORY -name '*.c')
clang-format -i -fallback-style=none -style=file $all_source_files || exit
1
modified=$(git status --porcelain --untracked-files=no)
if [ -n "$modified" ]; then echo "Run enforce_style.sh to reformat the
code."; exit 2; fi
```

clang-tidy

Clang-Tidy can be used to enforce some of the MISRA rules. A compliant configuration is provided below; the provided configuration, however, is

still a work-in-progress, so use with care.

.clang-tidy

```
---
Checks: >-
  boost-*,
  bugprone-*,
  cert-*,
  clang-analyzer-*,
  cppcoreguidelines-*,
  google-*,
  hicpp-*,
  llvm-*,
  misc-*,
  modernize-*,
  performance-*,
  portability-*,
  readability-*,
  -google-readability-todo,
  -readability-avoid-const-params-in-decls,
  -llvm-header-guard,
WarningsAsErrors: '*'
HeaderFilterRegex: '.*'
AnalyzeTemporaryDtors: false
FormatStyle: file
...
```

Legacy

Eclipse IDE autoformatter

Configuration file for the Eclipse autoformatter is attached below. Eclipse's embedded autoformatter is quite limited in its capabilities and tends to break complex C++ code, so one should not rely on it too much. New projects are recommended to use **JetBrains CLion** instead.



eclipse_f...atter.xml