**Coding / GitHub rules for FRAM\_CRAWLER\_2**

* **GitHub rules**

1. The main branch should always be buildable.
2. If a feature is being added, a separate branch from the main branch needs to be developed for this feature.
3. After finishing the development process for a feature, the main branch needs to be merged in the feature branch, and the code must be built after the merge, then merge the feature branch back into the main branch.
4. Make sure a “pull” is done before starting to develop, to ensure the code is updated.
5. Before making a release, a tag needs to be added in the repository with the SW/FW version of the code released, just to be able to come back to it if needed.
6. If GitKraken is being used, do not hit the “Stage all changes” command, until it is confirmed that the venv, and python configurations are not being committed.
7. Warnings should not be merged into main except if it is justified and approved by the team.

* **Coding rules**

1. Code should be cohered not coupled.

* Cohesion: Keeping every file’s content cohered into it, no variables/logic leaking from one file to another file.
* For example, a file that handles the Eddy-Current sensor logic shouldn’t have its variables being directly assigned a value from a file that handles the stepper motors.
* Coupling: No limits to the logic leak from one file to another file, which will result in more bugs and complicated code that is hard to understand.

1. If a file in the FW becomes really complicated, then it needs to be divided into two layers, a layer that handles the logic, and a file that connects the HW with the SW.
2. Variable names should be close to English so that they can be easily understood.
3. Clear comments should be added next to the unclear logic in the code.
4. Add some comments at the beginning of each file to explain its contribution, and how it works.