

Stacking solutions from Robot Norge

Prepwork for workshop 25.01.2021

Source code and general software quality

All source code should be versioned inside a Version Control system (VCS). A part from the obvious benefit of collaboration and safe keeping of the source code this will also give us good ability to track changes to the code over time. Furthermore it will give us the option to revert to previously known stable versions of the code in case we meet problems. Code should be in english and whenever the purpose of the commands or logical flow is not obvious it should be documented in-line.

The software should have sufficient separation of concern to reduce complexity and increase robustness. One should avoid large amounts of global parameters holding a complex state. Subroutines is a good way to achieve this.

Documentation of the software

The software should be documented as a product meaning all I/O and expected behaviours should be documented. Documentation should also contain all the different states, error codes and configuration possibilities. There should also be documentation on how to use the supplied testing programs to verify a new deploy before we put the robot back in rotation.

Deploy

Deploy should be done through a scripted process that has as few manual steps as possible to prevent potential human mistakes. This will ensure we are comfortable that we run a complete version of code on each robot and that there are no mix & match of new and old code. This also ensures that we run the exact same code on all robots, unless we for testing purposes are running different versions in a controlled fashion. This will increase our trust in the robots, the software and our ability to deploy and revert in a controlled fashion which in turn will cater for smaller more frequent deploys with less risks.

Testability

The software package should also contain tooling to test the robot manually after deployment. With functions like get a box from position X and put it on pallet Y. This program should be runnable directly towards the existing communication protocol that our platforms are communicating through. It should cover as many test cases Robot Norge feel is needed to be certain that the robot logic and / or physical movements are correct and the robot can safely be put in production. This test suite should be run systematically after every deploy of new software to the robots and only when they are successful will the new version be deemed functional.

Communication protocol

The current version was created swiftly and still has some shortcomings. We are hoping we can have one or more workshops when we create the next generation of this interface where we together can make sure we take advantage of what we have learned the past years. As an example we would like to pursue the idea of having less state in the robot and more of a streaming communication that can handle sudden deviations in the physical state of the boxes, pallets and stacking cells.

Support and fixing of bugs

Fixing bugs, adding missing functionality or improving the software with new functionality after the initial installation should always follow the procedures above and live up to the same quality as the initial launch. It is desirable that we keep such changes as small and frequent as possible to reduce risk with each fix / upgrade.