INTELLIGENT VEHICLE BUILDING BLOCKS

- Actuators
- Control
- Decision making & Planning
- Situation Analysis
- Data Fusion & Localisation

Vehicle sensors  V2X  Maps
CHALLENGES

• How do we avoid writing everything from scratch?
• How to manage different configurations of in-development components?
• How to quickly setup configuration for new computer, PhDs or Master student?
• How to provide workflow to test development code on the real vehicle?
• How to collaborate on code with companies with Non-Disclosure Agreements?
• Tracking code issues, exchanging fixes, having clear installation instructions
ROBOTICS OPERATING SYSTEM

- [http://ros.org/](http://ros.org/)
- ROS is OpenSource middleware layer, runs on Linux (Ubuntu) installation
- In ROS, each robotic module work as a separate programs
- ROS defines standardized communication channels, messages, and monitoring tools
- Record and playback data in ‘rosbag’ files
- Many different implementations of components available on github
INTERNAL 3ME GITLAB

- [https://gitlab.3me.tudelft.nl](https://gitlab.3me.tudelft.nl) is a TU Delft hosted ‘github-like’ git server
- Login with TU Delft credentials, external accounts can be made on request
- group management, issue tracking, code review, CI
INTELLIGENT VEHICLES GROUP ON GITLAB

• ~25 separate projects

• We can clone github repositories, create our own branches with small adaptations

• Also some non-ROS repositories, e.g. with lab assignment code for our courses
WHY USE INTERNAL GITLAB INSTEAD OF GITHUB?

• Login tied to TU Delft user accounts (external accounts added on request)
• Support from TU Delft ICT
• No pricing, or team limit on private repositories
• Not every project is ready to be shared with the world
• Not every project can be shared with the world (NDA)
• Safety check: someone from Delft uploaded it there
• In principle, Continuous Integration (CI) at TU Delft
• Does not exclude use of Github, you can always make more clones!
MANAGING DIFFERENT CODE VERSIONS

• All our ROS packages are in separate repositories on group’s gitlab
• Nested git repositories: One top-level repository, the packages are *git submodules*

When cloning top-level repository:
• Submodules cloned from git URLs stored in special `.gitmodules` file

When checking out specific version of top-level repository:
• Git also records what version of each subrepository should be, so checkout those too
PC in the back of our vehicle running ROS
**TESTING CODE ON THE PRIUS DEMONSTRATOR**

- All code on Demonstrator Vehicle installed via gitlab (no USB sticks)
- All data on Demonstrator Vehicle copied to TU Delft sftp share
- PhDs and staff have accounts on Demonstrator Vehicle
- We have one demo account that keeps latest & greatest working demo to show visitors

- PhD students can push to gitlab, and checkout on their Demonstrator Vehicle account
- Master students push test code on gitlab, PhD supervisor checks out on desktop PC
- Everybody can build their own branches on the Demonstrator, including the demo tag
- Only 1 administrator account (mine!) which can install new dependencies
WISHLIST / FUTURE WORK

• Gitlab subgroups (requires upgrade)
• Continuous Integration (CI)
• Slack-like tool for team chat, collaboration (e.g. Mattermost)
• Transferring repositories from personal to group domain is bit of a hassle
• Use git/gitlab in education, give students experience in code collaboration
• Docker or Singularity setup to facilitate reproducibility on other platforms