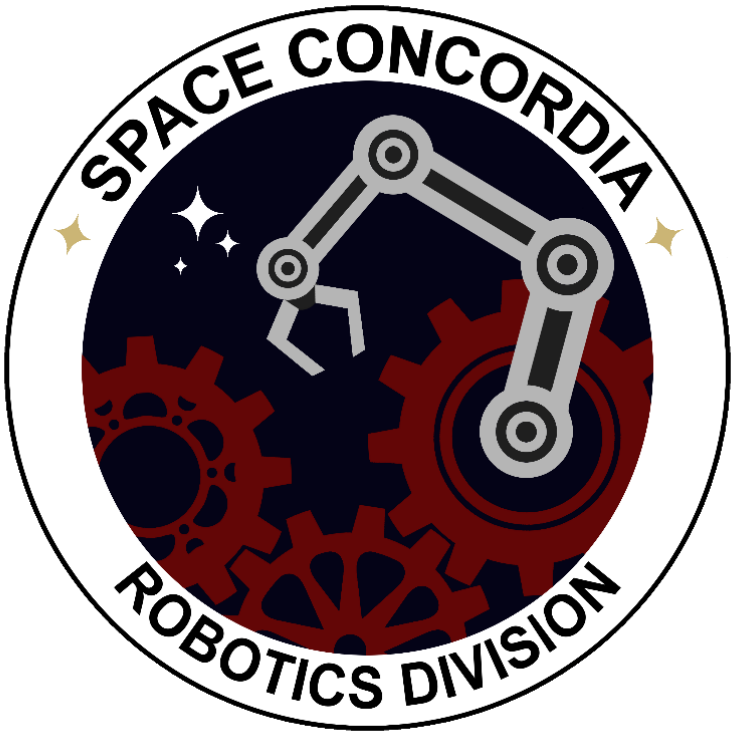

Space Concordia Robotics Division

1455 Boulevard de Maisonneuve Ouest Hall Building
H965
Montreal, QC H3G 2V8

Software Team: Software Training

What technologies, frameworks, programming languages, and hardware do we use?

- Our development environment is Ubuntu 22.04
- ROS Humble for modular robot code
- Python3.8+ and C++ for ROS Nodes
- C++ for high performance and safety critical microcontroller code (Teensy, ATMega328p)
- RQT for front-end web development
- Bash scripts for environment setup and automation
- GitHub for version control
- Nvidia Jetson TX2 as the robot on-board computer, soon to be upgraded

What does the SC Robotics Software Team do?

The software team deals with the software system in the bigger picture. We implement features for the operator GUI (Graphical User Interface), develop the code to give life to motors on board the rover, program sensors to get awareness of the rover's surroundings, work towards improving our architectural efficiency through DevOps, design and develop the AI behind the autonomous traversal tasks at competitions, and more. Ultimately, we concern ourselves with all aspects of the code running both on the base station and on the rover. An important framework we use is called ROS (Robot Operating System) and can be seen as 'glue code' between the different modules of our system. This training will demonstrate some of the fundamental ideas that are behind our system as well as help prepare you in your ability to work independently with us.

The autonomy team deals with anything regarding autonomous operation of the rover: path-planning, computer-vision, surrounding terrain analysis, manipulator operation through inverse kinematics.

How do I join the SC Robotics Software Team?

Every new member must complete a training before joining one of the Space Concordia (SC) divisions. In Robotics we have a different training for our different subdivisions. This document provides the introductory training for the Software Team as well as the steps to take after doing the training. Once you finish and successfully demo it to the designated software recruitment person, you are officially a member of the SC Robotics Software Team and will be given a firstname.lastname@spaceconcordia.ca email account, an invitation to our Slack channel, and access to our shared Google Drive where important documentation can be found.

The Software Training

Please read the instructions carefully before starting the task. You will be required to install Ubuntu 22.04 and use ROS Humble. There is no deadline to submit the training. Take sufficient time to learn and understand how to use the technologies the team is using.

Overview:

Using ROS Humble on Ubuntu 22.04, create two Nodes that can communicate with each other over ROS, using the publisher and subscriber model, this model is built into ROS. One node should be written in Python and the other in C++, there is a ROS library for both languages. You can make it as creative as you wish. As a **bonus**, you can also make the code interact with a microcontroller like an arduino or teensy. You are free to decide how you want to implement it all, and what kind of information you want the nodes to communicate.

Recommended Steps:

- Install Ubuntu 22.04 LTS. You can install your preferred distro, but be prepared to spend some time getting it to work, as it might not be supported, or fully compatible with our codebase. A Virtual Machine is acceptable, but not recommended for autonomy, as it will become **very** resource intensive.
- Install ROS Humble on your fresh linux installation
- Go over the tutorials here. The “beginner level” is more than you will need.
- Using what you learned in the tutorials, create a C++ and a Python node that communicate with each other using the ROS library.
- For your code, create a repository on github/gitlab and use the version control features. We want to see a commit history. Finally, once you are done, share your repository with us (emails at the bottom of the document).

General Tips

- If you are not familiar with Git and GitHub, checkout this video
- New to Linux? Give this a read. Also you may want to watch this.
- Decent Linux friendly editors are Atom and VS Code
- For C++ and Python, the JetBrains suite is great (CLion and PyCharm community editions). You can get free pro versions by getting Github Education (for free).
- If you have an old laptop to spare, backup your files (if needed), and install Ubuntu onto it completely. Otherwise you will have to set up a dual boot alongside your current operating system. Or use a Virtual Machine.

I'm stuck. What can I do?

Make sure to take advantage of google. After all, the point of this training is for you to exercise some autonomy and self learning. That being the case, you should do your best to try and find the proper resources to familiarize yourself with the technologies as you work towards completing the training.

Feel free to ask for assistance on the recruitment slack, or by email. We have members there who would be happy to help in any way they can. They can't do parts of the training for you, but they can help with errors or technical issues and point you in the right direction.

I'm done. What now?

Congratulations! Contact the designated lead to demo your project. Share your git repository, and briefly explain what the code should do. The priority is for the code to work. We might also provide feedback on your work if we feel like it is necessary.

Software Leads

William Wells william.wells@spaceconcordia.ca
Tim Freiman tim.freiman@spaceconcordia.ca