Getting Started

The purpose of this page is to get new CS developers up and going for robosub. If you have any questions, feel free to ask them on the ❌software slack channel!

OS

The sub itself runs Ubuntu 20.04 (focal), as do most members. Any operating system which can run ROS2 foxy should work such as Linux Mint 20. Windows is currently not supported.

Virtual Machine

If you don't feel comfortable installing linux on your machine yet, you can get started quickly with a virtual machine. However, we highly recommend members to install linux on their machine, the virtual machine works very poorly. You can download our robosub virtualbox disk image here. It's a 2.7GB compressed download, after extracting with 7zip it's around 11GB. This already has the robosub code repository downloaded and all dependencies installed, so you can skip some steps. To run the virtual machine image, you'll first need to download Virtualbox. Note however that you cannot run the simulator within a virtual machine, gazebo7 crashes.

Languages

We use primarily C++ for the software on the sub, with a little bit of Python2 (note: not Python3) for the AI. Below are a few links for tutorials for these languages, these are just a few I quickly found to get you started, feel free to search around for other tutorials.

- C++
- Python

Getting Prerequisite Software

This will setup your computer with our custom robosub package repository, the ROS package repository, and the Gazebo package repository. When you install our robosub/robosub-simulator package, all other dependencies will automatically be downloaded. If you experience any issues, please contact the ❌IT slack channel. Depending on your operating system, follow the instructions on one of these pages:

Install ROS

Deprecated: ROS Indigo (for Ubuntu 14.04 trusty or Linux Mint 17)
Deprecated: ROS Kinetic (for Ubuntu 16.04 xenial or Linux Mint 18)

ROS2 Foxy (for Ubuntu 20.04 focal or Linux Mint 20)

Getting our Code

Follow these steps for getting the robosub code on your computer. First, make sure you have ssh authentication setup with gitlab. You only need to do this once on each computer. To check if you already have keys made for your computer, run:

```
$ ls ~/.ssh/id_rsa*
```

If you don't see id_rsa and id_rsa.pub files, you can create new keys by running:

```
$ ssh-keygen
```

Just hit enter through all of the prompts. Don't type in a password, the entire point of this is to eliminate the need to use a password with gitlab! Now, copy the contents of id_rsa.pub and paste the contents in a new key here. To test if this worked, run:

```
$ ssh git@gitlab.com
```

If everything is set up correctly, you should expect to see a message like:

```
PTY allocation request failed on channel 0
Welcome to GitLab, @user!
Connection to gitlab.com closed.
```

Finally, we need to setup our ROS workspace with the robosub code.

```
echo "source /opt/ros/foxy/setup.bash" >> ~/.bashrc
source ~/.bashrc
sudo apt update
sudo apt install python3-rosdep2 python3-colcon-common-extensions ros-foxy-
gazebo-ros ros-foxy-gazebo-plugins socat
mkdir -p ~/ros/src
cd ~/ros/src
git clone https://gitlab.com/PalouseRobosub/Software/robosub.git
git clone https://gitlab.com/PalouseRobosub/Software/robosub_msgs.git
git clone https://gitlab.com/PalouseRobosub/Software/robosub_simulator.git
cd ~/ros
colcon build
echo "source ~/ros/install/setup.bash" >> ~/.bashrc
echo 'alias rsmake="(cd ~/ros && colcon build)"' >> ~/.bashrc
source ~/.bashrc
```

From now on you should be able to compile code using rsmake from any directory.

Now What?

If you were able to successfully compile the software at the end, congrats! Next steps are to learn how to use git, and take a look at the software workflow. After this, head over to the gitlab issues page and see if a task looks interesting!

From:
https://robosub.eecs.wsu.edu/wiki/ - **Palouse RoboSub Technical Documentation**

Permanent link:
https://robosub.eecs.wsu.edu/wiki/cs/getting_started/start?rev=1650010564

Last update: **2022/04/15 01:16**