

# User Guide

## Table of Contents

1 Setup.....	1
1.1 Prerequisites.....	1
1.2 Setup (1) using OpenTCS Binary (recommend).....	1
1.3 Setup (2) using OpenTCS Extensions.....	1
2 Configuration Options.....	2
3 Usage With Example Project.....	2

## 1 Setup

NOTE: The setup has only been tested with Ubuntu 18.04 LTS and sROS Dashing Diademata and with the [TurtleBot3](#) robot.

### 1.1 PREREQUISITES

Install the Turtlebot 3 environment for ROS2 ([link](#)) on your local machine.

### 1.2 SETUP (1) USING OPENTCS BINARY (RECOMMEND)

*NOTE: Skip this setup if you want to set the driver up using extensions (Setup 2).*

With this setup, the complete OpenTCS binary will be used that includes the ROS2 vehicle driver. This setup is recommended if you are not planning to use other vehicle drivers besides the ROS2 vehicle driver.

1. Download binary from the release page.
2. Extract the binary to a location of choice.

OpenTCS can now be started by executing the regular OpenTCS starter (``start_kernel.sh``).

**It is very important that you run OpenTCS in a [ROS 2-sourced environment](#).**

### 1.3 SETUP (2) USING OPENTCS EXTENSIONS

*NOTE: Skip this setup if you ran the setup using an OpenTCS Binary (Setup 1).*

With this setup, an extension will be installed that only contains the ROS2-vehicle driver. You need to have an existing setup of OpenTCS. This setup is recommended if you intend to use other vehicle drivers within OpenTCS besides the ROS2 vehicle driver.

1. Download and extract the extension binary form the release page.
2. Copy ``openTCS-extensions`` to ``<open_tcs_binary>/openTCS-Kernel/lib``.
3. Copy ``openTCS-extensions`` to ``<open_tcs_binary>/openTCS-KernelControlCenter/lib``.
4. Copy ``config`` to ``<open_tcs_binary>/openTCS-Kernel/``.

OpenTCS can now be started by executing the regular OpenTCS starter (``start_kernel.sh``).

**It is very important that you run OpenTCS in a [ROS 2-sourced environment](#).**

## 2 Configuration Options

If you followed the setup steps above, there is no need to explicitly set the configuration parameters unless you want to override them.

Key	Description
<code>ros2.adapter.enable</code>	Whether to disable / enable the ROS2 vehicle driver (`true` `false`)
<code>ros2.adapter.plantModelScale</code>	The scale of the plant model. 1:x where x is the input. Provide a small scale for little plants.

## 3 Usage With Example Project

The ROS2-driver is a multi-purpose driver and can (technically) be used for any ROS2-vehicle. This usage guide is made and tested for a simulated TurtleBot3, but can also be applied to other (similar) robots.

1. Start a gazebo simulation with: TurtleBot3 World ([how-to](#)).
2. Generate SLAM maps in the simulated TurtleBot3 World ([how-to](#)).
3. Start Navigation in RViz with the map you created in Step 2. ([how-to](#)).
4. Set the initial position of the TurtleBot3 ([how-to](#)), this could also be done in the Kernel Control Center Panel.

Now our simulation with Gazebo and RViz with navigation has been started, we are ready to use the OpenTCS.

1. Download the example plant for the TurtleBot3 World. ([link](#)).
2. Start OpenTCS Kernel and the OpenTCS Plant Overview.
3. Load the example plant in the Plant Overview (CTRL + L).
4. Persist the example plant to the Kernel. (ALT + P).
5. Switch the Plant Overview to 'operating mode' (ALT + O).
6. Start OpenTCS Kernel Control Center.
7. Select the first vehicle ('Bus1') and open the 'ROS2 Options' panel.
8. Enable the driver.
9. The AGV is now ready to be used. You can test it by pushing the 'Dispatch to point' button.