AVL ModelCONNECT

Introduction

The Integrated and Open Development Platform is a strategic orientation of AVL. The aim is to connect all elements of the vehicle development process. And this step by step and independent of tools and manufacturers. Neutral connectors interlink simulation models, testbeds, data or analyses to create more flexible environments for the development of a new generation of intelligent and efficient vehicles.

Model.CONNECT™ – Connecting models of different domains, which improves development efficiency by interlinking simulation models into a consistent virtual prototype.

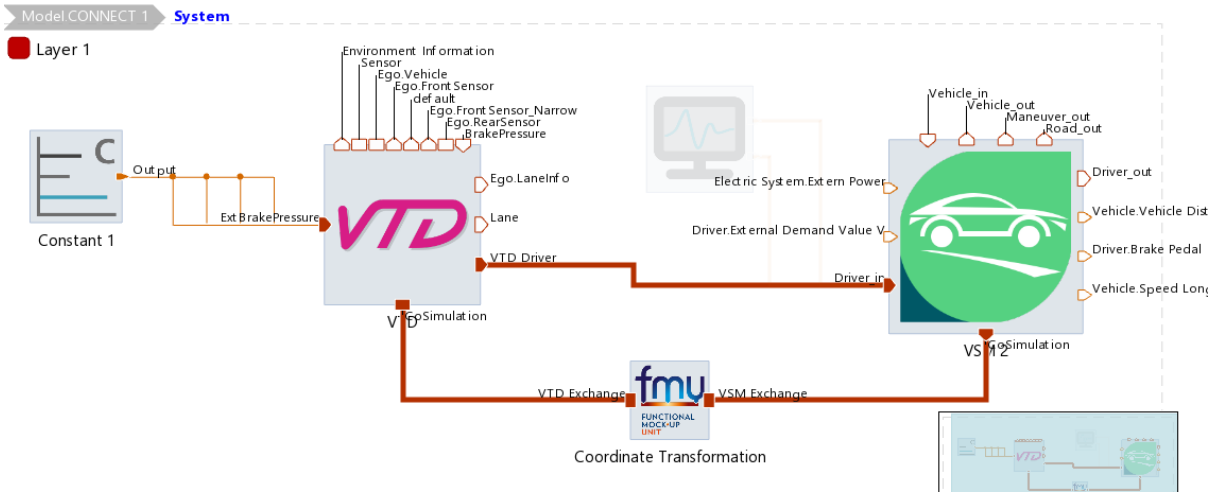
http://bjchepk996.com/web/guest/home

Setting before running

- ROS system in Linux PC
- NoMachine in Linux and Windows PC
- AST Launcher in Windows PC
- AVL VSM 4.3.3 in Windows PC
- Matlab in Windows PC
-

System

- Open OSE3AD_4 system models in **model.CONNECT**



There are three different models which will be discussed below in details.

VSM

Vehicle Simulation Model (in Windows)

VSM stands for vehicle model where we can **select different vehicle model as simulation object**. By clicking properties, you can configure the model input/output ports.

It also connects with **AVL VSM 4.3.3** software, where you can access more details about vehicle, such as motor, tyres, and drivetrains. (Already set, try to not modify)

Meanwhile, you can do testrun in **AVL VSM 4.3.3** to test the basic functionality of vehicle model with no scenior specificed.

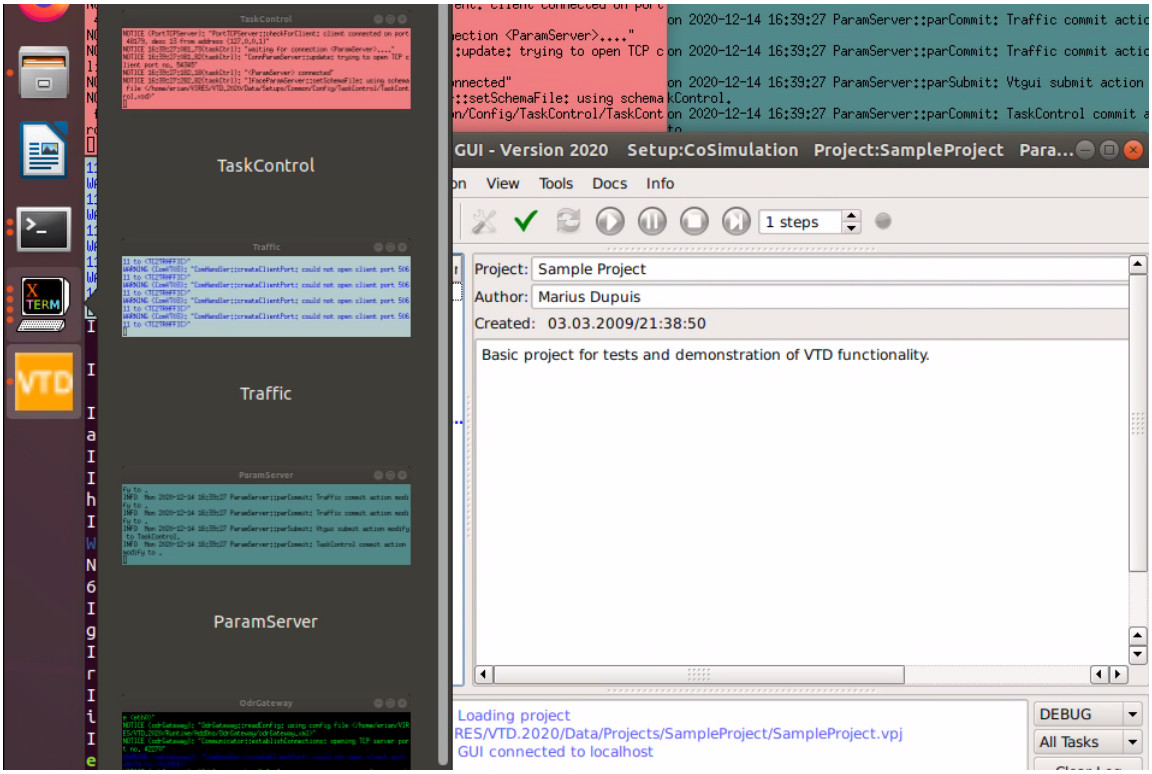
VTD

Virtual Test Drive (in Linux)

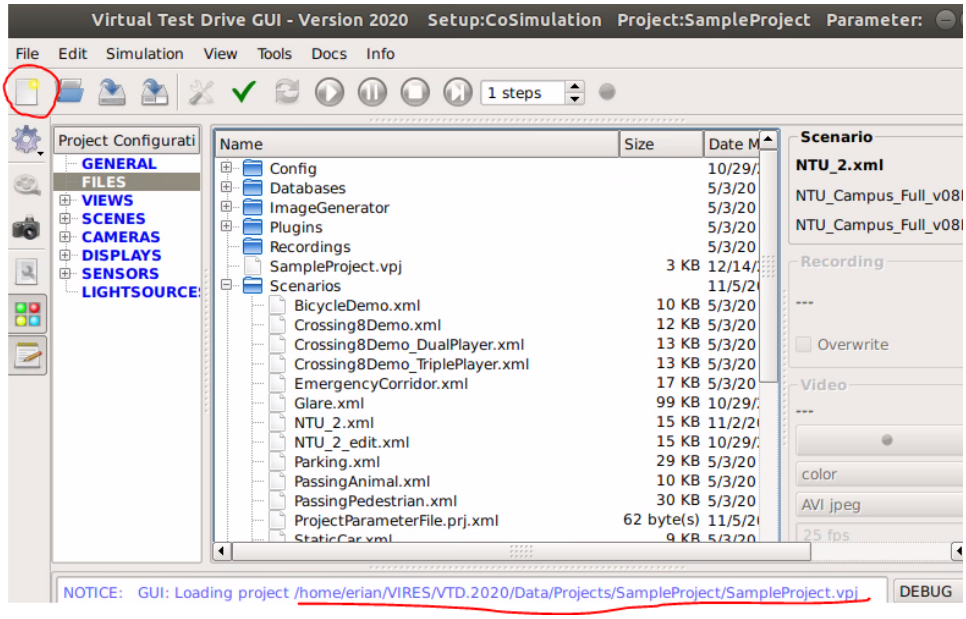
- To access it, the **NoMachine** software is used to connect Linux terminal.
- Follow the instruction in Documents/VTD_Documentary.odt to install and run test VTD. If already installed, skip and contine from next step.
- To launch **VTD (Virtual Test Drive)**, run following commands

```
cd ~/VIRES/VTD.2020/bin
./vtdStart.sh -select
2 (Where 2 is co-simulation mode)
```

- The VTD software will be opened with several process (taskControl, traffic...)

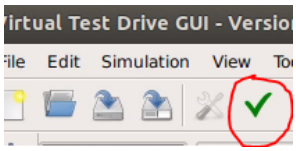


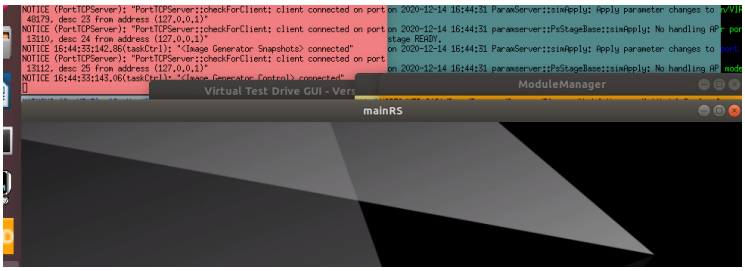
- Select project as **SampleProject.** Click FILES from project configuration, the file under Scenarios are in xml which we need to use in our model building.



- To run VTD, click on the tick icon (apply configuration).

  The mainRS window should comes out, which is visualizer.





- When you want to change Scenarios from ModelCONNECT or want to exit VTD. You must run this command to make sure VTD is properly closed.

```
~/VIRES/VTD.2020/bin$ ./vtdStop.sh
```

VTD Tools — Road Designer

ROD software to build maps

VTD Tools — Scenario Editor

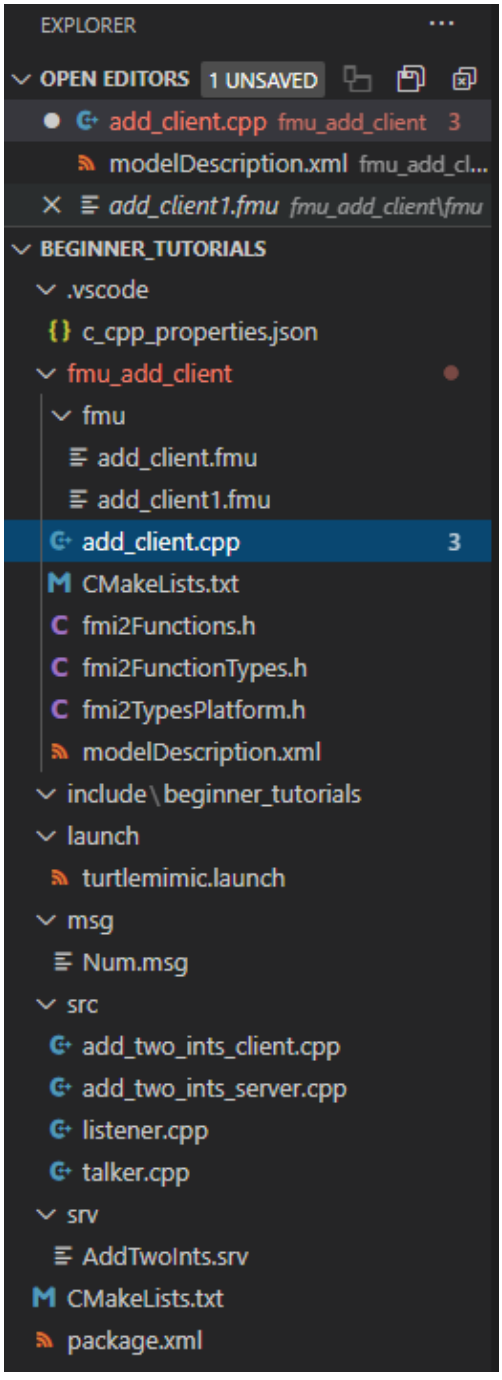
Open drive editor software to modify objects (vehicle, passagers..)

You can edit scenario xml files such as define new waypoint path.

FMU

Functional Mock-up Unit (in Windows)

The basic idea is use C++ to interact with Ros functions (services/nodes ...)



The FMU generated is add_client.fmu

- **add_client.cpp**

  Main C++ file defines basic structure of fmu

  Define client - serviceClient links with srv

- fmi Function.h files

  .h files are header files which define variables and paramenters.

- CMakeList.txt (Inner)

  Define messages, path, and commands

- CMakeList.txt (Outer)

  Main cmake file

  catkin_make will build from here

- modelDescription.xml

  Define fmiModel information

  Keep consistent with other coding

- src directory

  Source files for beginner_tut package to interface with ros functions

  add_two_ints_client.cpp

  add_two_ints_server.cpp

  listener.cpp

  talker.cpp to publish message

```
cd C:\Users\erianguest\Desktop\ros (your workspace)
devel\setup.bat   (Update ros path if there are new packages)
rosrun beginner_tutorials add_two_ints_server
rosrun beginner_tutorials add_two_ints_client
```

Proposed Solution

Instead of connecting Linux terminal through NoMachine, we can set rosbridge between linux and windows.

- However rosbridge only can be achieved in same computer

The better way is set master/slave ros terminal which both are in same network connected through Ethernet cable.

- Linux (Master)
- Windows (Slave)
-

Follow-up Tasks

What needs to be done next for this proposal?

☐