# Assetto corsa PC setup.

Needed software:

- Assetto corsa
- ROS melodic
- Vjoy (virtual controller)

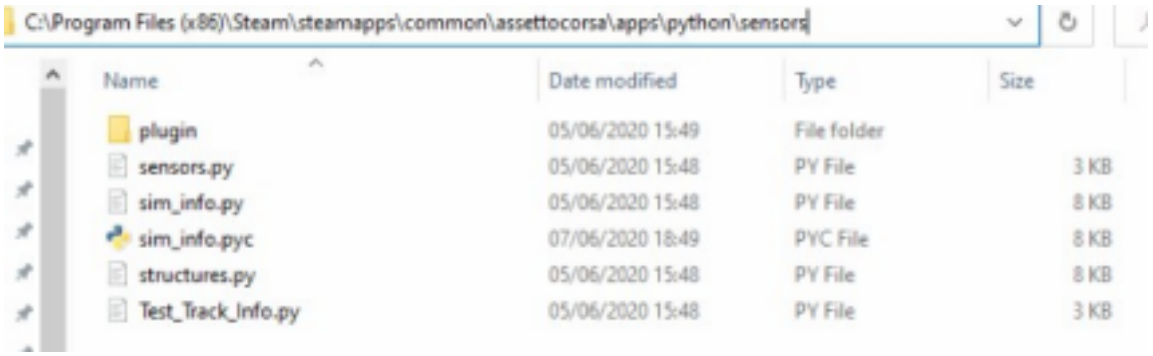
Steps:

1. Install Assetto Corsa
    a. Download and install Steam
    b. Search for Assetto Corsa (not Assetto Corsa Competizione) on Steam c. Buy and install the game

2. Install ROS
    a. To install ROS on Windows, Visual Studio is required
    b. Follow this link (ROS Melodic). I strongly reccomend to install and use the new Windows PowerShell in order to have multiple terminals in a unique window and to exploit other useful tricks.

3. Download and install Vjoy

4. Create a folder named “sensors” under <AC_installation_folder>\assettocorsa\apps\python\
    a. Clone the “link_repo” repository into sensors folder. You should have a similar setting:




The AC_installation_folder could be at Steam\steamapps\common\assettocorsa\

5. Create a ROS workspace
    a. Clone the ROS AC wrapper into the workspace
    b. Add your controller into the workspace

6. Download the embedded python version with sockets library and other configuration files (windows-libs) and follow the readme’s istructions

7. Download the Indianapolis speedway and copy the folders into: <AC_installation_folder>\content\tracks

8. Download the Indy IL-15 and copy the folders into:
   <AC_installation_folder>\content\cars

Steps to follow in order to activate the autonomous mode:

1) Launch the game

2) Open several terminals/window command

3) Start roscore

4) Activate the plugins inside the game, checking gMeter and sensors in options->general:


In options → control, be sure to have Vjoy and WASD and same configuration:


5) Start a race session

6) Select the Indy car oval and the Indianapolis speedway

Car brand logo:

Make sure to select the oval one:


7) In a ROS window command:
   a. change directory to the assetto-corsa-data-wrapper path, source the devel setup using: devel\setup.bat
   b. go to src\receiver\src\ and launch: rosrun receiver recv.py

8) Into another ROS window command:
   a. change directory to the assetto-corsa-data-wrapper path, source the devel setup using: devel\setup.bat
   b. go to src\w_send\src and launch: rosrun w_send windows_sender.py

9) Start your controller receiving the data from control_input_data topic and publishing the outputs to control_output_data

Known issues:

- The trajectory of the pitlane is not available, so it's important to start from the track starting grid using a Race session. In this way is mandatory to include an opponent. Should be found a workaround.

- The windows_sender.py should be restarted after each session.

- The track boundaries aren't accurate in the Indy track, should be checked if there is a better track mod for this issue.