

Date: 09/11/2022

Author: Sanskruti Jadhav

# | HOW TO DOCKER

## 1. INSTALLATION OF DOCKER ON UBUNTU OS

Prerequisites:

OS requirements: To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

Ubuntu Jammy 22.04 (LTS)

Ubuntu Impish 21.10

Ubuntu Focal 20.04 (LTS)

Ubuntu Bionic 18.04 (LTS)

Uninstall old versions

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

It's OK if apt-get reports that none of these packages are installed.

The contents of /var/lib/docker/, including images, containers, volumes, and networks, are preserved.

If you do not need to save your existing data, and want to start with a clean installation, refer to the below site:

[uninstall-docker-engine](#)

This installation guide is the same as the main docker hub installation below:

<https://docs.docker.com/engine/install/ubuntu/#uninstall-docker-engine>

### A. Installation Method: Setup the repository

Although there are many other ways to install docker engine we will use the recommended repository method

- i. Update the apt package index and install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get update
```

```
$ sudo apt-get install \ ca-certificates \ curl \ gnupg \ lsb-release
```

ii. Add Docker's official GPG key:

```
$ sudo mkdir -p /etc/apt/keyrings
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

```
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05: ~  
(base) sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo docker images  
[sudo] password for sanskrj:  
sudo: docker: command not found  
(base) sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ conda deactivate  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo apt-get remove docker docker-engine docker.io containerd runc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
E: Unable to locate package docker-engine  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo apt-get update  
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1,575 B]  
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1,575 B]  
Get:2 file:/var/cudnn-local-repo-ubuntu2004-8.5.0.96 InRelease [1,575 B]  
Get:2 file:/var/cudnn-local-repo-ubuntu2004-8.5.0.96 InRelease [1,575 B]  
Hit:3 http://mirror.ces.clemson.edu/ubuntu focal InRelease  
Get:4 https://mirror.ces.clemson.edu/ubuntu focal-updates InRelease [114 kB]  
Hit:5 http://packages.microsoft.com/repos/code stable InRelease  
Get:6 http://mirror.ces.clemson.edu/ubuntu focal-backports InRelease [108 kB]  
Hit:7 http://archive.canonical.com/ubuntu focal InRelease  
Get:8 https://mirror.ces.clemson.edu/ubuntu focal-security InRelease [114 kB]  
Hit:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64 InRelease  
Hit:10 https://storage.googleapis.com/bazel-apt stable InRelease  
Hit:11 https://packages.ros.org/ros/ubuntu focal InRelease  
Get:12 http://mirror.ces.clemson.edu/ubuntu focal-updates/universe amd64 DEP-11 Metadata [278 kB]  
Get:13 http://mirror.ces.clemson.edu/ubuntu focal-updates/universe amd64 DEP-11 Metadata [391 kB]  
Get:14 http://mirror.ces.clemson.edu/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]  
Get:15 http://mirror.ces.clemson.edu/ubuntu focal-backports/main amd64 DEP-11 Metadata [7,992 B]  
Get:16 http://mirror.ces.clemson.edu/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.5 kB]  
Get:17 http://mirror.ces.clemson.edu/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]  
Get:18 http://mirror.ces.clemson.edu/ubuntu focal-security/universe amd64 DEP-11 Metadata [77.2 kB]  
Get:19 http://mirror.ces.clemson.edu/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]  
Fetched 1,164 kB in 2s (604 kB/s)  
Reading package lists... Done  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo apt-get install \  
> ca-certificates \  
> curl \  
> gnupg \  
> lsb-release  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
lsb-release is already the newest version (11.1.0ubuntu2).  
lsb-release set to manually installed.  
ca-certificates is already the newest version (20211016-20.04.1).  
ca-certificates set to manually installed.  
curl is already the newest version (7.68.0-ubuntu2.13).  
gnupg is already the newest version (2.2.19-3ubuntu2.2).  
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo mkdir -p /etc/apt/keyrings  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ echo \  
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo apt-get update  
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1,575 B]  
Get:1 file:/var/cudnn-local-repo-ubuntu2004-8.5.0.96 InRelease [1,575 B]  
Get:2 file:/var/cudnn-local-repo-ubuntu2004-8.5.0.96 InRelease [1,575 B]  
Hit:3 https://storage.googleapis.com/bazel-apt stable InRelease  
Get:4 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]  
Hit:5 http://mirror.ces.clemson.edu/ubuntu focal InRelease  
Hit:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64 InRelease
```

iii. Use the following command to set up the repository:

```
echo \ "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

## B. Install Docker Engine

- i. Update the apt package index, and install the latest version of Docker Engine, containerd, and Docker Compose, or go to the next step to install a specific version:

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Note: \*\* Receiving a GPG error when running apt-get update?

Your default umask may not be set correctly, causing the public key file for the repo to not be detected. Run the following command and then try to update your repo again: `sudo chmod a+r /etc/apt/keyrings/docker.gpg`.

- ii. To install a specific version of Docker Engine, list the available versions in the repo, then select and install:

- a. List the versions available in your repo:

```
$ apt-cache madison docker-ce
```

- iii. Install a specific version using the version string from the second column, for example, I have replaced <VERSION\_STRING> with 5:20.10.18~3-0~ubuntu-focal in below command

```
$ sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-  
cli=<VERSION_STRING> containerd.io docker-compose-plugin
```

## C. Verify Docker Engine is installed and Setup

Verify that Docker Engine is installed correctly by running the hello-world image.

```
$ sudo service docker start
```

```
$ sudo docker run hello-world
```

```
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo apt-get install docker-ce=5:20.10.18~3-0-ubuntu-focal docker-ce-cli=5:20.10.18~3-0-ubuntu-focal containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
containerd.io is already the newest version (1.6.8-1).
docker-ce-cli is already the newest version (5:20.10.18~3-0-ubuntu-focal).
docker-ce is already the newest version (5:20.10.18~3-0-ubuntu-focal).
docker-compose-plugin is already the newest version (2.10.2-ubuntu-focal).
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo service docker start
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:7d246653d0511db2a0b2e0436cfd0e52ac8c06000264b3ce63331ac66dca625
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Thats It! now you have a running docker engine that can utilized for running docker images

## 2. RUNNING CUSTOMIZED PROJECT CHRONO IMAGE

Let's now call our project chrono image: docker : **sanskirj/ubuntu: psychrono\_projchrono**

### A. Features

Project chrono build with Irrlicht engine

Psychrono in anaconda

POV Ray

Cuda packages

MATLAB R\_2022a linked with project chrono (meaning MATLAB can be called from project chrono c++ script)

### B. Project Chrono Structure

the build directory and the chrono source codes are located in /opt folder,

This folder also has MATLAB install directory. Additionally anaconda3 is installed.

Now that you are in the terminal, you can do ls and you can see below output folders, the main build and folders are found in /opt folder.

It is important to have new features in /opt folder only, by default docker installs in /root folder (e.g. MATLAB, Anaconda etc is installed in /root)

we need to shift these directories to /opt because we need to access these later in singularity images form, they cannot access root folders of docker images.

This is a topic of later discussion, for now just focus on running a demo of project chrono

**\$ ls**

### C. PC Setting

Giving display for our docker to run project chrono irrlicht engine visualization is significant. Run below commands on terminal before running the docker image.

since docker can be run by sudoers only, we need to add access to root user allows the root user to access the running X server.

The current X server is indicated by the DISPLAY environment variable

**\$ xhost +si:localuser:root**

**\$ echo \$DISPLAY**

return argument is :1/0(depending on your system)

Take note of this.

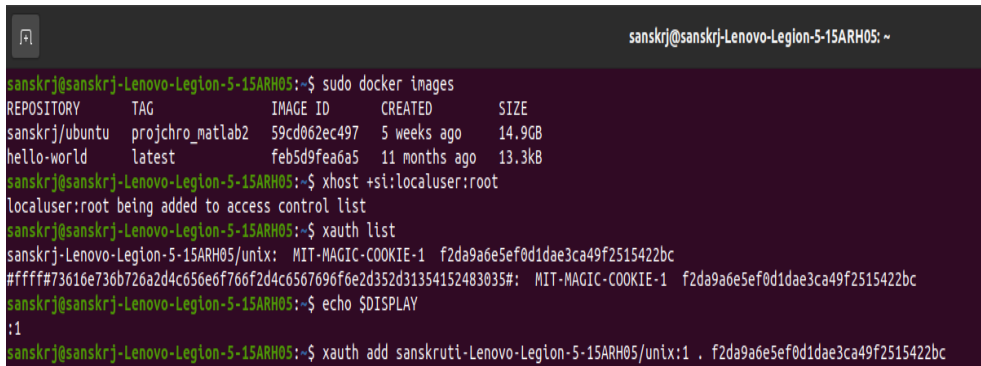
**\$ xauth list**

The xauth command is usually used to edit and display the authorization information used in connecting to the X server.

This will give an MIT magic cookie name of my pc as well a hex key

in above command :1 is my echo \$DISPLAY output additionally after the first part it is necessary to add . before hex key

```
$ xauth add sanskruti-Lenovo-Legion-5-15ARH05/unix:1 .  
06b578f1f32c4921ad7ac0559db142d8
```



```
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05: ~  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo docker images  
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE  
sanskrj/ubuntu       projchro_matlab2 59cd062ec497    5 weeks ago    14.9GB  
hello-world          latest          feb5d9fea6a5    11 months ago  13.3kB  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ xhost +si:localuser:root  
localuser:root being added to access control list  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ xauth list  
sanskrj-Lenovo-Legion-5-15ARH05/unix: MIT-MAGIC-COOKIE-1  f2da9a6e5ef0d1dae3ca49f2515422bc  
#ffff73616e736b726a2d4c656ef766f2d4c6567696f6e2d352d31354152483035#: MIT-MAGIC-COOKIE-1  f2da9a6e5ef0d1dae3ca49f2515422bc  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ echo $DISPLAY  
:1  
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ xauth add sanskruti-Lenovo-Legion-5-15ARH05/unix:1 . f2da9a6e5ef0d1dae3ca49f2515422bc
```

#### D. How to Run the Image

In below detail just replace Image -name as sanskrj/ubuntu:projchro\_matlab2

```
$ sudo docker run -it --rm -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix --  
net=host <image-name >
```

This will bring you to the terminal of the Ubuntu OS , it acts exactly as we type commands in terminal except you need not use sudo.

Now if you want you can visualize this as a normal virtual box OS as well which will display all of it in files and folders just as normal OS, this requires additional processes of using VNC+ docker

But for our purposes let's start with a simple command line interface of docker image.

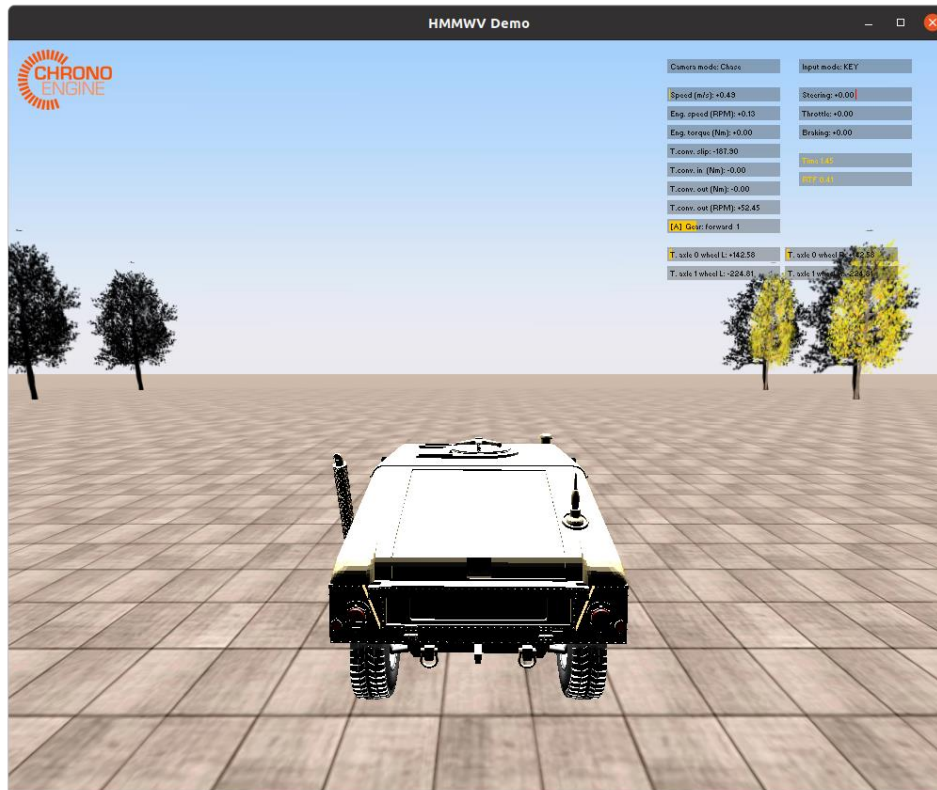
```

docker: Error response from daemon: could not select device driver "" with capabilities: [[gpu]].
sanskri@sanskri-Lenovo-Legion-5-15ARH05:~$ sudo docker run -it --rm -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix --net=host sanskri/ubuntu:projchro_matlab2
root@sanskri-Lenovo-Legion-5-15ARH05:/# ls
NGC-DL-CONTAINER-LICENSE bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp tmpjcef-p978_scheme tmp usr var
root@sanskri-Lenovo-Legion-5-15ARH05:/# cd /opt/
root@sanskri-Lenovo-Legion-5-15ARH05:/opt# ls
CMakeCache.txt CMakeFiles build chrono chrono matlab 2022a
root@sanskri-Lenovo-Legion-5-15ARH05:/opt# cd build_chrono/bin/
root@sanskri-Lenovo-Legion-5-15ARH05:/opt/build_chrono/bin# ls
DEMO_OUTPUT demo_FEA_shellsANCF_3833 demo_IRR_entt_cluster demo_IRR_spring demo_VEH_Cosin_Vlper demo_VEH_Harder
demo_CH_EulerAngles demo_FEA_thermal demo_IRR_entt_processor demo_IRR_suspension demo_VEH_Cosin_WheelRig demo_VEH_RCCar
demo_CH_archive demo_FEA_truss demo_IRR_enttler demo_IRR_tire demo_VEH_Cosin_WheelRig_CustomTerrain demo_VEH_Ride
demo_CH_bulldsystem demo_FEA_visualize demo_IRR_forklift demo_IRR_tracks demo_VEH_Cosin_WheeledVehicle demo_VEH_RigidTerrain
demo_CH_coords demo_IRR_Adams_parser demo_IRR_fourbar demo_IRR_tjoints demo_VEH_DeformableSoil demo_VEH_Sedan
demo_CH_filesystem demo_IRR_OpenSim_parser demo_IRR_friction demo_IRR_visualize_collision demo_VEH_FEDR_Accel demo_VEH_Sedan_AIDriver
demo_CH_functions demo_IRR_assets demo_IRR_gears demo_MTLB_functions_plot demo_VEH_Gator demo_VEH_Sedan_Lockable_Diff
demo_CH_linalg demo_IRR_aux_ref demo_IRR_imposed_motion demo_MTLB_matlab demo_VEH_Gator_Accel demo_VEH_Shock
demo_CH_math demo_IRR_ballSMC demo_IRR_link_bushing demo_POST_entt_creation demo_VEH_HMMWV demo_VEH_SteeringController
demo_CH_matrix_ref demo_IRR_bricks demo_IRR_necanum demo_POST_enttler_asset demo_VEH_HMMWV9 demo_VEH_SuspensionTestRig
demo_CH_powertrain demo_IRR_callbackNSC demo_IRR_motors demo_POST_ghnplot demo_VEH_HMMWV9_YUP demo_VEH_TireTestRig
demo_CH_solver demo_IRR_paths demo_IRRPendulum demo_POST_povray demo_VEH_HMMWV_Accel demo_VEH_TrackTestRig
demo_CH_stream demo_IRR_cohesion demo_IRR_pendulum demo_POST_timestepping demo_VEH_HMMWV_DeFsoil demo_VEH_TrackedJSON
demo_FEA_basic demo_IRR_collisionNSC demo_IRR_plane_plane demo_ROBOT_Curiosity_Rigid demo_VEH_HMMWV_DoubleLaneChange demo_VEH_TrackedJSON_Band
demo_FEA_cables demo_IRR_collisionSMC demo_IRR_prismatic_force demo_ROBOT_Curiosity_SCM demo_VEH_Kraz_ClosedLoop demo_VEH_TwoCars
demo_FEA_cablesMATLAB demo_IRR_collision_2d demo_IRR_prismatic_limits demo_ROBOT_LittleHexy demo_VEH_Kraz_OpenLoop demo_VEH_UAZBUS
demo_FEA_contacts_SMC demo_IRR_collision_object demo_IRR_raycast_test demo_ROBOT_Robosimlan_Rigid demo_VEH_LMTV_ride demo_VEH_UAZBUS_SAE
demo_FEA_Cosmulate_load demo_IRR_collision_trimesh demo_IRR_report_collisions demo_ROBOT_Robosimlan_SCM demo_VEH_M113 demo_VEH_WheeledGeneric
demo_FEA_dynamics demo_IRR_convergence demo_IRR_rev_limits demo_ROBOT_Turtlebot_Rigid demo_VEH_M113_DeFsoil demo_VEH_WheeledJSON
demo_FEA_electrostatics demo_IRR_conveyor demo_IRR_rev_sph demo_ROBOT_Vlper_Rigid demo_VEH_MAN_1ot demo_VEH_paths
demo_FEA_hexaANCF_3813 demo_IRR_crank demo_IRR_rev_trans demo_ROBOT_Vlper_SCM demo_VEH_MAN_St
demo_FEA_loads_dynamics demo_IRR_custom_contact demo_IRR_rot_spring demo_VEH_ArticulatedVehicle demo_VEH_MAN_Zt
demo_FEA_loads_statics demo_IRR_decomposition demo_IRR_sollbin demo_VEH_Citybus demo_VEH_MADLE
demo_FEA_shellsANCF_3423 demo_IRR_earthquake demo_IRR_sph demo_VEH_Cosin_Curiosity demo_VEH_MTV_ride
root@sanskri-Lenovo-Legion-5-15ARH05:/opt/build_chrono/bin# ./demo_VEH_HMMWV
Copyright (c) 2017 projectchrono.org
Chrono version: 7.0.6

Irrlicht Engine version 1.8.4
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using plain X visual
This driver is not available in Linux. Try OpenGL or Software renderer.
Cannot use default video driver - fall back to OpenGL.
Irrlicht Engine version 1.8.4
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using renderer: OpenGL 3.1
llvmpipe (LLVM 12.0.0, 256 bits): Mesa/X.org
OpenGL driver version is 1.2 or better.
GLSL version: 1.4

```





```
sanskrj@sanskrj-Lenovo-Legion-5-15ARH05:~$ sudo docker run -lt --rm -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix --net=host sanskrj/ubuntu:projchro_matlab2
[sudo] password for sanskrj:
root@sanskrj-Lenovo-Legion-5-15ARH05:/# cd /opt/build_chrono/bin/
root@sanskrj-Lenovo-Legion-5-15ARH05:/opt/build_chrono/bin# ./demo_VEH_HMMWV
Copyright (c) 2017 projectchrono.org
Chrono version: 7.0.0

Irrlicht Engine version 1.8.4
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using plain X visual
This driver is not available in Linux. Try OpenGL or Software renderer.
Cannot use default video driver - fall back to OpenGL
Irrlicht Engine version 1.8.4
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using renderer: OpenGL 3.1
llvmpipe (LLVM 12.0.0, 256 bits): Mesa/X.org
OpenGL driver version is 1.2 or better.
GLSL version: 1.4
Loaded texture: /opt/build_chrono/data/skybox/sky_lf.jpg
Loaded texture: /opt/build_chrono/data/skybox/sky_up.jpg
Loaded texture: /opt/build_chrono/data/skybox/sky_dn.jpg
PNG warning: ICCP: known incorrect sRGB profile
Loaded texture: /opt/build_chrono/data/logo/chronoengine_alpha.png
Loaded texture: /opt/build_chrono/data/fonts/arial80.bmp
Loaded texture: /opt/build_chrono/data/vehicle/terrain/textures/tile4.jpg
Loaded texture: /opt/build_chrono/data/models/trees/textures/mapletree.png
Loaded texture: /opt/build_chrono/data/models/trees/textures/mapletreeA.png
Loaded texture: /opt/build_chrono/data/models/trees/textures/mapletree90.png
Loaded texture: /opt/build_chrono/data/models/trees/textures/mapletree90A.png

Subsystem types
Chassis:      RigidChassis
Powertrain:   ShaftsPowertrain
Driveline:    ShaftsDriveline4WD
Steering 0:   PitmanArm
Axle 0
  Suspension: DoubleWishbone
  Brake:       BrakesSimple
  Tire:        TMeasyTire
Axle 1
  Suspension: DoubleWishbone
  Brake:       BrakesSimple
  Tire:        TMeasyTire
```

\$ apt-get update

As I have already build all the project chrono, you have all the executable demos present in this path

\$ cd /opt/build\_chrono/bin

\$ ./demo\_HMMV\_VEH

This will output a HMMV vehicle in an environment and you can interact with using keyboard keys a,w,d,s for right turn, forward, left turn and stop respectively.

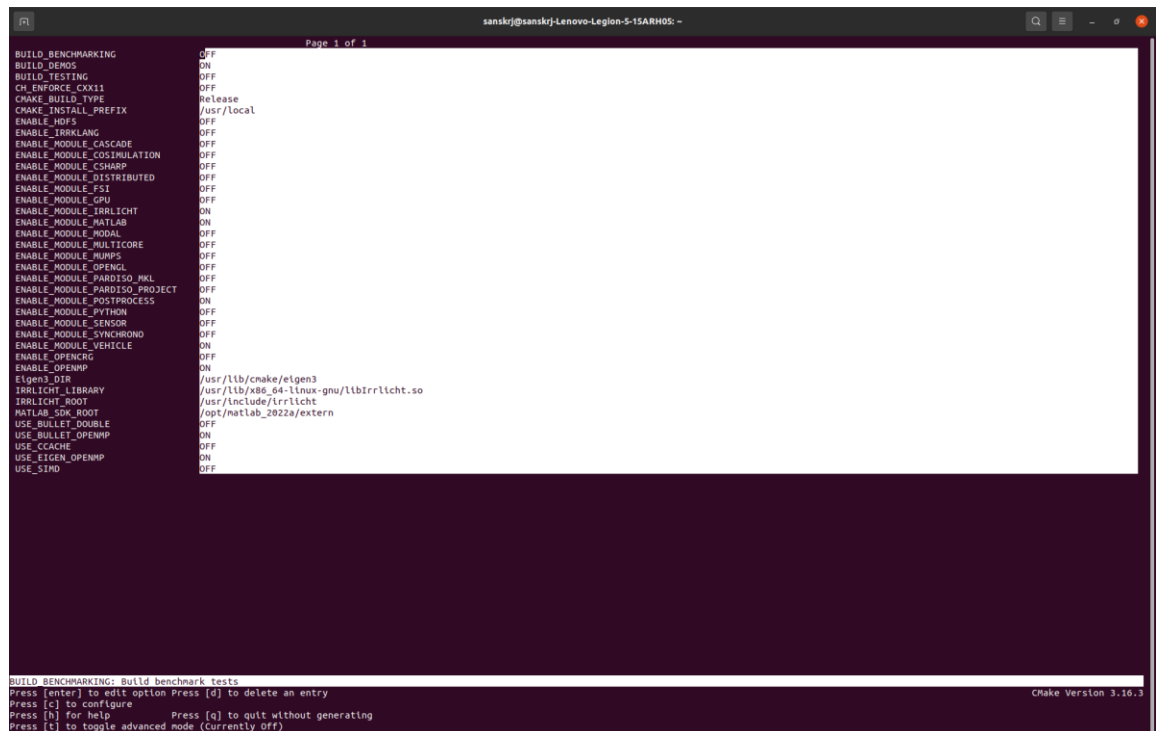
### E. Modify Image

If you make changes in chrono/ src containing the source code in c++ of various modules of vehicle you will need to re-build the chrono as below

```
$ cd /opt/build_chrono
```

```
$ cmake ../chrono
```

This will give a CMAKE GUI as below and you need to configure and generate pressing C and g respectively. It will automatically exit once generation is done .Now we need to make and make install,



```
$ make
```





After having made a lot of changes, adding packages , new apps etc you will need to commit to changes or they will not persist next time you run the image and form a new image by committing to your docker hub a account. You will need your account on docker hub and this image will be stored in repository publicly available to all.

```
$ exit
```

```
$ sudo docker ps -a
```

```
$ sudo docker commit <containerID> {repository:tag}
```

e.g. docker commit c3f279d17e0a sanskrj/ubuntu:projectchrono

```
$ sudo docker login
```

```
$ sudo docker push
```

Details for how to commit is present in docker :

<https://docs.docker.com/engine/reference/commandline/commit/>

### 3. RUNNING CUSTOMIZED PYCHRONO IMAGE

Let's now call our pychrono

docker: **sanskrj/ubuntu: pychrono\_projchrono**

Once in the shell, activate chrono environment with python 3.10 also has irrlicht package installed in this environment.

The environment is build using the below webpage information:

[https://api.projectchrono.org/pychrono\\_installation.html](https://api.projectchrono.org/pychrono_installation.html)

```
$ conda activate chrono
```

```
$ cd /opt/anaconda3/pkgs/pychrono-7.0.0-py310_2116/lib/python3.10/site-packages/pychrono/demos/vehicle/
```

```
$ python demo_VEH_HMMV.py
```

```
(chronon) root@sanskri-Lenovo-Legion-5-15ARH05:/opt/anaconda3# cd pkgs/pychronon-7.0.0-py310_2116/lib/python3.10/site-packages/pychronon/demos/vehicle/
(chronon) root@sanskri-Lenovo-Legion-5-15ARH05:/opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/lib/python3.10/site-packages/pychronon/demos/vehicle# python demo_VEH_HMMWV.py
Irrlicht Engine version 1.8.5
Warning: The library version of the Irrlicht Engine (1.8.5) does not match the version the application was compiled with (1.8.4). This may cause problems.
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using plain x visual
This driver is not available in Linux. Try OpenGL or Software renderer.
Cannot use default video driver - fall back to OpenGL
Irrlicht Engine version 1.8.5
Warning: The library version of the Irrlicht Engine (1.8.5) does not match the version the application was compiled with (1.8.4). This may cause problems.
Linux 5.15.0-46-generic #49-20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Using renderer: OpenGL 3.1
llvmpipe (LLVM 12.0.0, 256 bits): Mesa/X.org
OpenGL driver version is 1.2 or better.
GLSL version: 1.4
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/logo_pychrono_alpha.png
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/skybox/sky_lf.jpg
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/skybox/sky_up.jpg
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/skybox/sky_dn.jpg
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/fonts/arial80.bmp
Loaded texture: /opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/share/chrono/data/vehicle/terrain/textures/tile4.jpg

***** System Information *****
Time = 0.0

Markers

Driver loc: 0.0 0.0 0.0
Chassis COM loc: 0.0 0.0 0.0

Tire forces

Front left: 0.0 0.0 0.0
Front right: 0.0 0.0 0.0
Rear left: 0.0 0.0 0.0
Rear right: 0.0 0.0 0.0

Quit message received.
(chronon) root@sanskri-Lenovo-Legion-5-15ARH05:/opt/anaconda3/pkgs/pychronon-7.0.0-py310_2116/lib/python3.10/site-packages/pychronon/demos/vehicle#
```

