ROS2 & Webots.

Notes xx,xyz often denote a place holder value you must enter.

Note Try to keep to the same pc each week as your work will be stored locally!

Make notes in the log book as you go along

Apart from the last step, this should all be done in one terminal. Don’t close your terminal

**Step 1)**

Open a terminal (ctrl+alt+t)and type

ros2 topic list

You will likely see a lot of random topics, we don’t want this. So type the following

export ROS\_DOMAIN\_ID=xx

xx = the last two digits on the yellow sticker on your pc. If using your own system you can skip this. (if you are on pc 0-1-2-3 use the domain 215,216,217)

This sets our ros domain name. Currently all pcs are on the same domain, which means any ros node will be shared.

**Step 2)**

In the terminal use cd (change directory) and ls(directory list) to navigate to the following directory

sp2006user@ent-sp2006-32u:~/ros2\_humble/src

i.e., your first command is likely

cd ros2\_humble

Once in ros2\_humble/src

ros2 pkg create --build-type ament\_cmake --license Apache-2.0 --node-name MyRobotDriver my\_package --dependencies rclcpp geometry\_msgs webots\_ros2\_driver pluginlib

This command will create a package for us, set up basic topics and nodes + create our file structure. Here we diffine our package as my\_package and our controller as my\_robot\_driver. These names can be changed but we need to change it across all commands.

cd my\_package  
mkdir launch  
mkdir worlds  
mkdir resource

Mkdir created two folders for us, launch and worlds

touch my\_robot\_driver.xml  
touch include/my\_package/MyRobotDriver.hpp

Touch creates empty files for us.

Your directory should look like this, have a look to double check

src/  
└── my\_package/  
 ├── launch/  
 ├── my\_package/  
 │ ├── \_\_init\_\_.py  
 │ └── my\_robot\_driver.py  
 ├── resource/  
 │ └── my\_package  
 ├── test/  
 │ ├── test\_copyright.py  
 │ ├── test\_flake8.py  
 │ └── test\_pep257.py  
 ├── worlds/  
 ├── package.xml  
 ├── setup.cfg  
 └── setup.py

Step 3

Navigate to my\_package/include/my\_package/

Gedit MyRobotDriver.hpp

This will open Myrobotdriver in gedit (a note pad), copy the content of myrobot.hpp on canvas into gedit and click save and close. MAKE SURE TO CLOSE GEDIT, else we can run into issues later.

Do the same for MyRobotDriver.cpp

Step 4.

Navigate to my\_package/resource. Hint you can use cd .. To go up a level

Once in the direction

Gedit my\_robot.urdf

In the previous example, we used gedit to edit a new file, however we can also use it to create a new file such as in this case.

Copy the content from the urdf file on canvas into your local file

Step 5.

Navigate to the world folder.

Gedit my\_world.wbt

copy and past the world info from the file on canvas.

Step 6.

Navigate to my\_package/launch

Gedit robot\_launch.py

copy and past the launch info from canvas.

Step 7.

Replace the content of my\_package/my\_robot\_driver.xml with the info on canvas

Replace the content of my\_package/CMakeLists.txt with the content from canvas

Step 8.

In your terminal

colcon build  
source install/local\_setup.bash  
ros2 launch my\_package robot\_launch.py

Step 9.

In a new terminal copy all the following at once.

ros2 topic pub /cmd\_vel geometry\_msgs/msg/Twist '

linear:

x: 0.1

y: 0.0

z: 0.0

angular:

x: 0.0

y: 0.0

z: 0.0'