TiRRex

ROBOTERRIUM

Jean Laneurit
jean.laneurit@inrae.fr

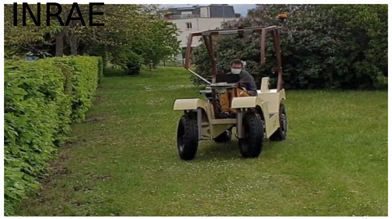
Roland Lenain
roland.lenain@inrae.fr

Goal

- Make available some off-road robotic platforms belonging to French research laboratories like INRAE, INRIA, CEA, LAAS, Xlim …
- Provide codes needed to control them under ROS2 middleware
- Provide an experimental environment located to the INRAE Montoldre site

Robots

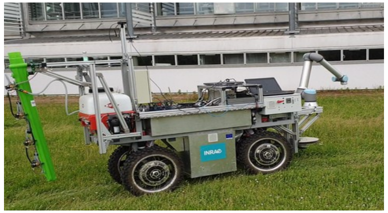
RE1. Robotized electric tractor – Alpo, SabiAgri - INRAE



Locomotion mode	2WS, 4WD
Weigth	800 kg (+500kg)
Size	2,5mx1mx 2,5m
Max Speed	2,5 m/s
Autonomy	4 H
Manipulation type	Not yet equipped?
On-boarded sensors pkg	RTK-GPS [Drotek], IMU [Xsens], Odometry, Lidar sick 1 layer

→ Autonomy, Ag work, implement (Greenhouse, row navigation)

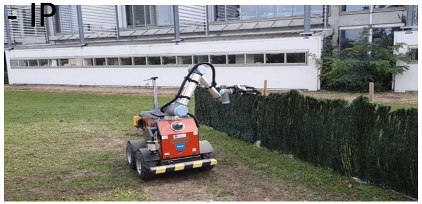
RE2. Electric robot – Adap2E– INRAE



Locomotion mode	4WS-D
Weigth	550 kg (+100)
Size	2,5mx1,5mx1,5m
Max Speed	5m/s tested (th: 8m/s)
Autonomy	6 H
Manipulation type	UR10
On-boarded sensors pkg	RTK-GPS [Drotek], IMU [Xsens], Odometry, Lidar sick 1 layer

→ Mobile manipulation, autonomous navigation

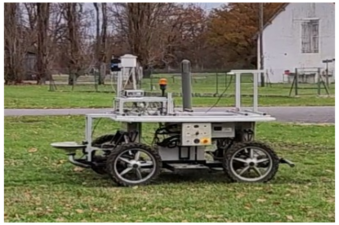
RE3. Skid steer mobile manipulator – Campero platform - IP



Locomotion mode	differential
Weigth	400kg (20kg)
Size	1mx1mx1m
Max Speed	2m/s
Autonomy	1H30
Manipulation type	UR10
On-boarded sensors pkg	RTK-GPS [Drotek], IMU [Xsens], Odometry, 2 Lidar of 1 layer

→ Mobile manipulation, autonomous navigation

RE4. 2WS Robot – RobuFAST platform - INRAE

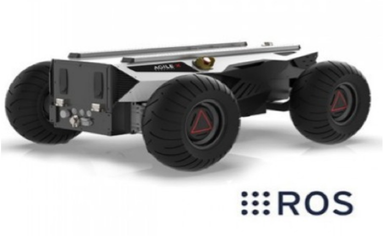


Locomotion mode	4WS-D
Weigth	550kg (10kg)
Size	1,2m
Max Speed	7m/s
Autonomy	2H
Manipulation type	none
On-boarded sensors pkg	RTK-GPS [Drotek], IMU [Xsens], Odometry, 2 Lidar of 1 layer

→ High speed test, stability, regular ground

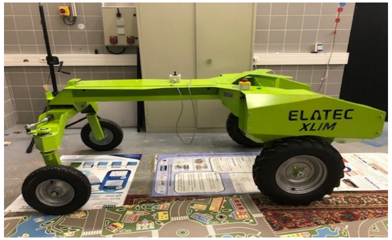
Robots

RE5. 2WS mobile robot – Agilex Platform – LAAS



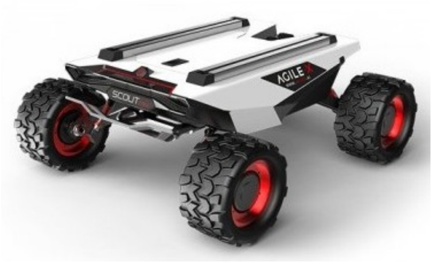
Locomotion mode	2WS
Weigth	150kg
Size	1,2*1*0,8m
Max Speed	2m/s
Autonomy	4H
Manipulation type	none
On-boarded sensors pkg	4 realsense, 1lidar

RE6. 2WS mobile robot – Cinteo –Xlim



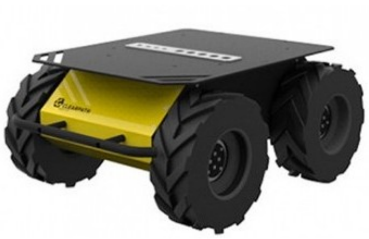
Locomotion mode	2WS
Weigth	50kg (50kg)
Size	0,8→ 1,6*1*1
Max Speed	1m/s
Autonomy	4H
Manipulation type	none
On-boarded sensors pkg	4 realsense, 1lidar

RE7. 4WD mobile platform : Scouts minis – Xlim



Locomotion mode	4WD
Weight	20Kg
Size	0,63*0,55*0,25
Max Speed	5,5m/s
Autonomy	10Km
Manipulation type	none
Onboarded sensor-pkg	none

RE8. Husky –GT Lorraine



Locomotion mode	4WD
Weight	50Kg
Size	0,99*0,67*0,39
Max Speed	1m/s
Autonomy	8h
Manipulation type	none
Onboarded sensor-pkg	none

Base / ROS2

Robots	ROS2 Live	ROS2 simulation	Public
Alpo INRAE	Base OK Under tests	OK	Not yet
Adap2e INRAE	Base OK Under tests	OK	OK
Campero IP	Base OK Under tests Arm not OK	Base OK Arm not OK	No yet
Robufast INRAE	Base OK Under tests	OK	OK

Base / ROS2

Robots	ROS2 Live	ROS2 simulation	Public
Agilex LAAS	2024	2024	
Cinteo XLIM	Under development	Under development	
Scout-mini XLIM	2024	2024	
Husky GT Lorraine	2024	2024	

Sensors / ROS2

Robots	Sensors	ROS2 Live	ROS2 simulation
Alpo INRAE	Ublox RTK GPS Xsens IMU Mti10	OK OK	OK OK
Adap2e INRAE	Ublox RTK GPS Xsens IMU Mti10 Sick LMS 150	OK OK OK	OK OK OK
Campero IP	Ublox RTK GPS Xsens IMU Mti10 Sick LMS 150	OK OK OK	OK OK OK
Robufast INRAE	Ublox RTK GPS Xsens IMU Mti10 Sick TiM31	OK OK OK	OK OK OK

Sensors / ROS2

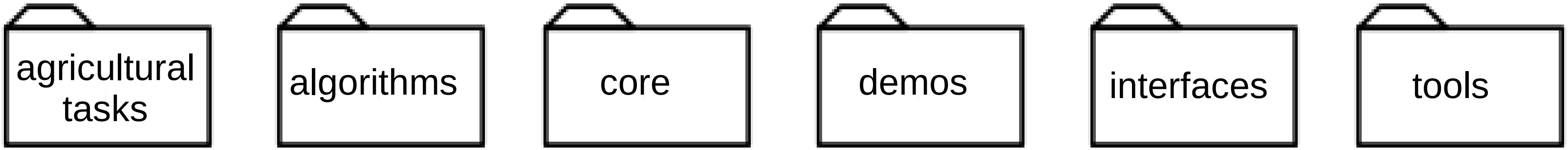
Robots	Sensors	ROS2 Live	ROS2 simulation
Agilex LAAS			
Cinteo XLIM	Ublox RTK GPS Xsens IMU Robotsense 16 Nap 2 cameras RGB, 2 NiR	OK OK ?? ??	OK OK ?? ??
Scout-mini XLIM			
Husky GT Lorraine			

ROS2 developments

- Codes :
  - https://github.com/Romea
    - Core :
      - Low level :
        - Sensor data processing
        - Mobile base kinematics
      - High level :
        - Localisation algorithms
        - Path following algotrithms

    ROS :
      - ROS interface for core codes
      - ROS interfaces for devices (sensors, manipulators, robots …)
      - Simulation (gazebo)
  - https://github.com/Tirrex-Roboterrium
    - Launch packages used to start easily demonstration for each robot
- Entry point :
  - https://github.com/Tirrex-Roboterrium/tirrex_demo/tree/main/doc

Workspace organization



agricultural tasks

algorithms

core

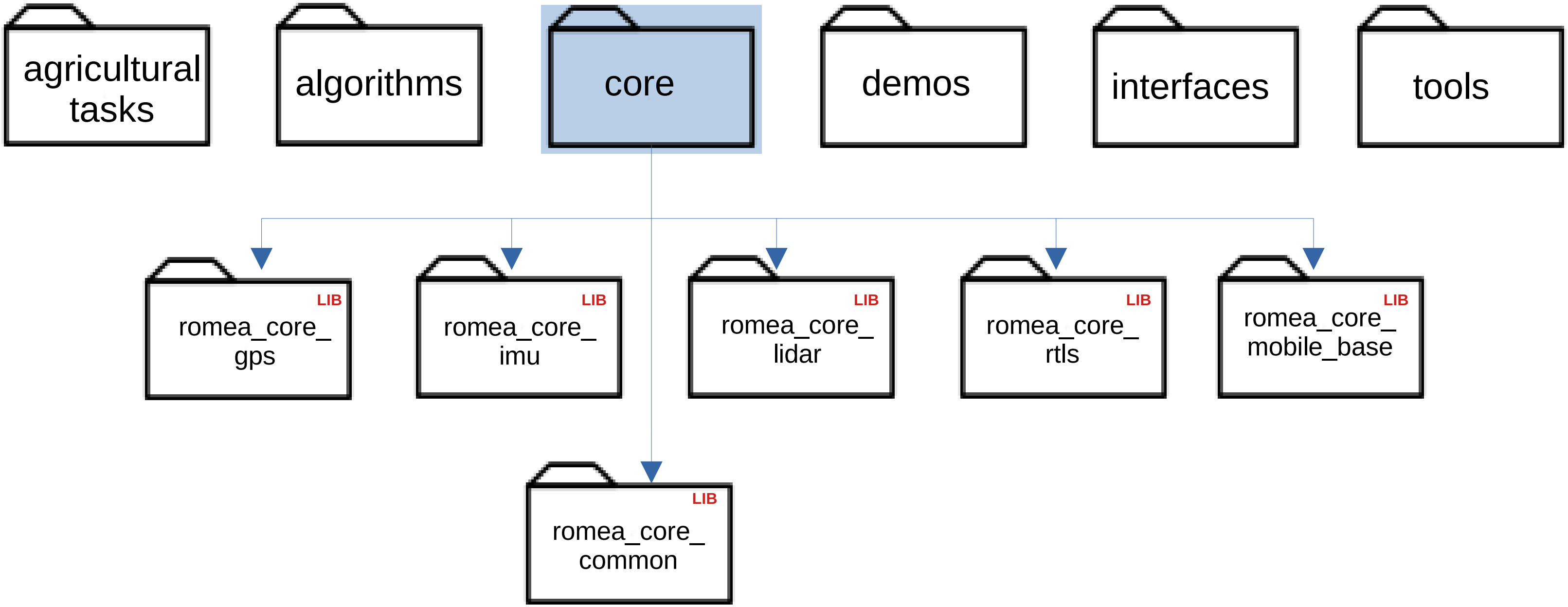
demos

interfaces

tools

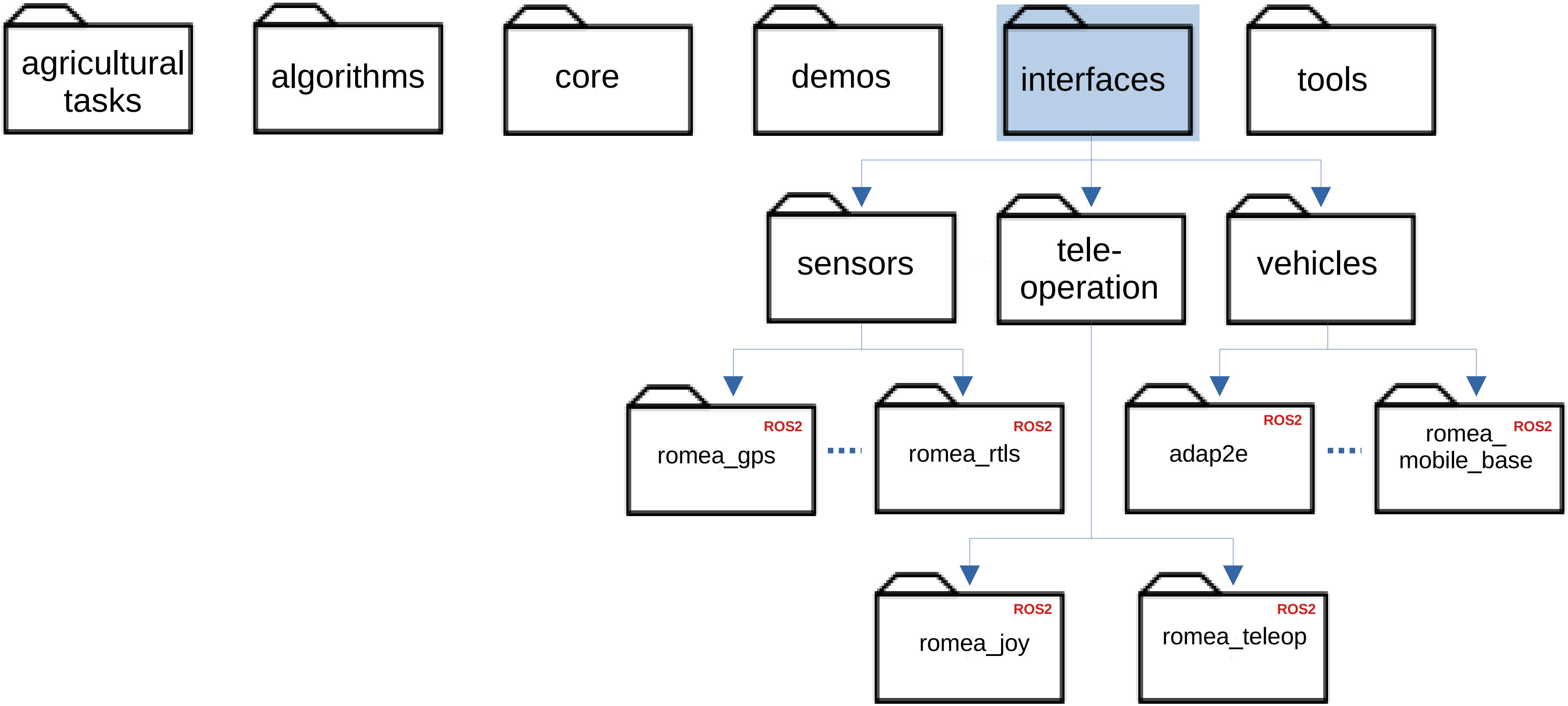
Workspace organization

Core librairies



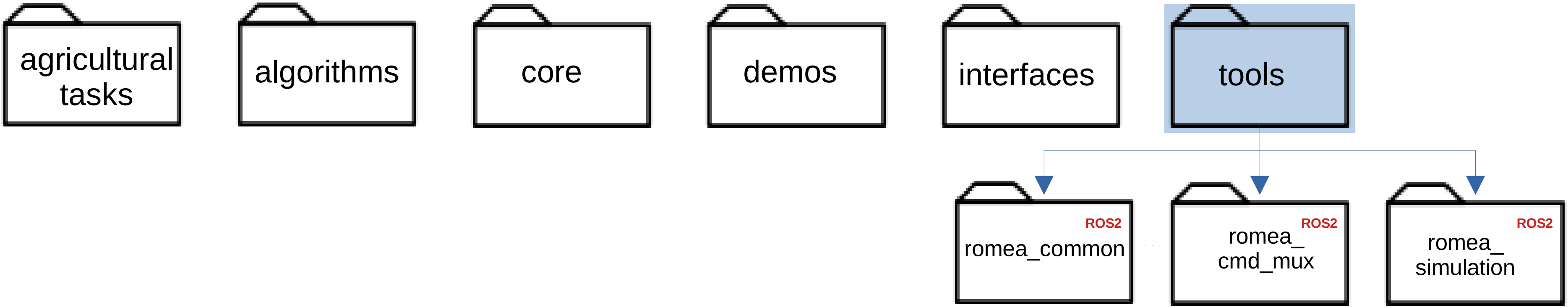
Workspace organization

Device interfaces packages



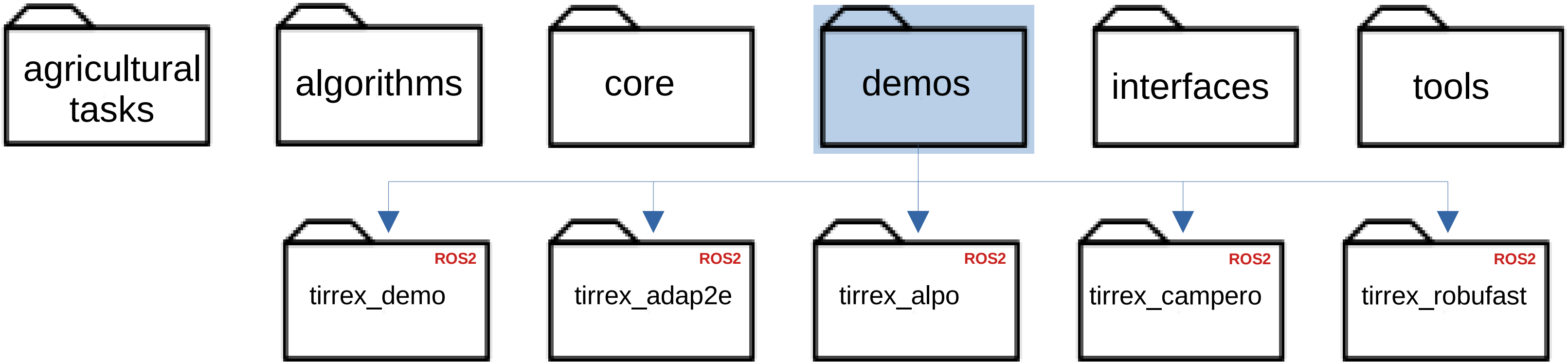
Workspace organization

Tools packages



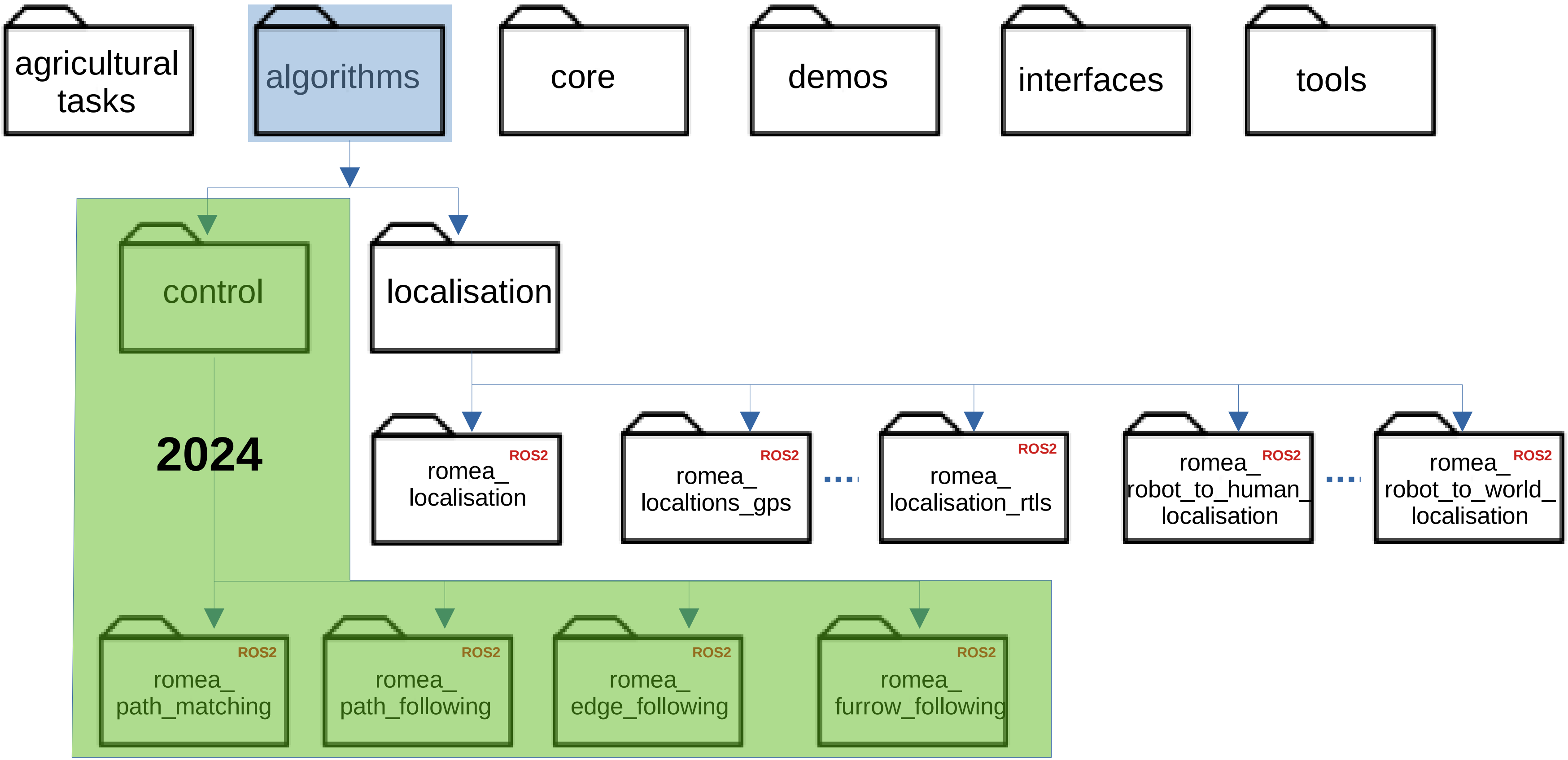
Workspace organization

Demo packages



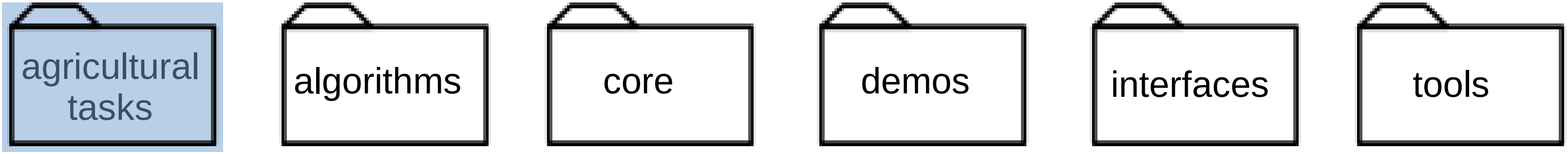
Workspace organization

Algorithms packages



Workspace organization

Agricultural tasks packages



agricultural tasks

algorithms

core

demos

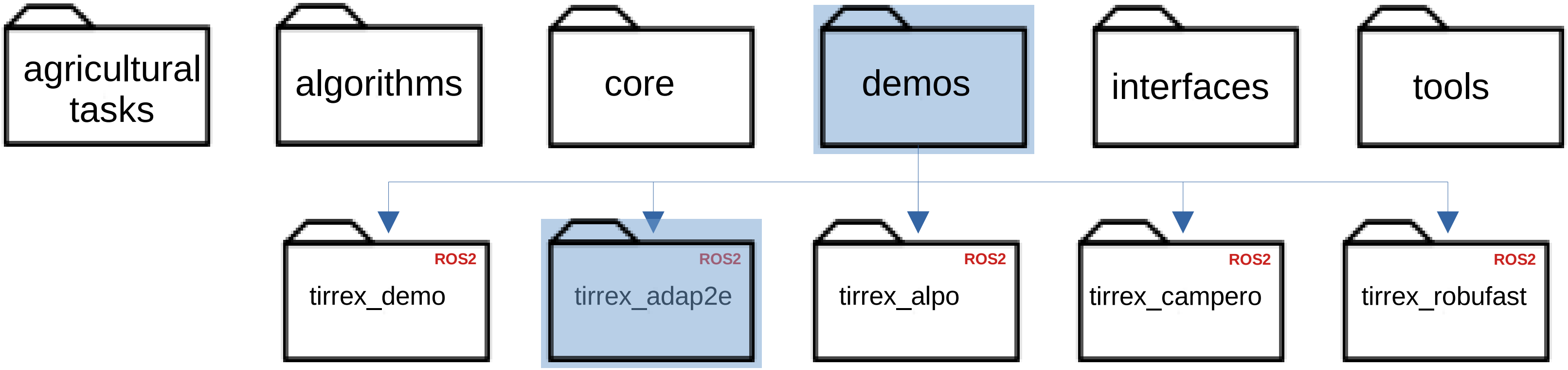
interfaces

tools

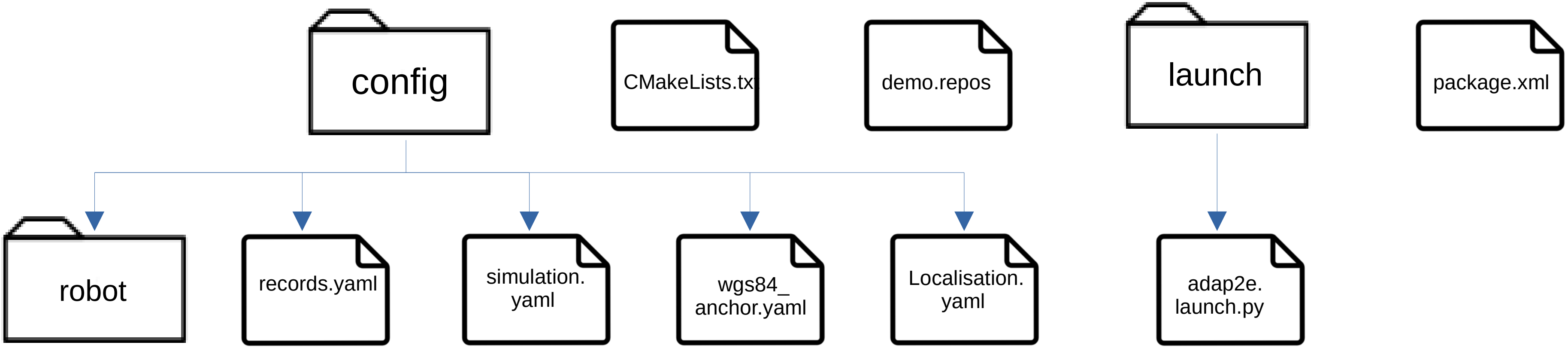
Contributions du projet Ninsar

Workspace organization

Demo packages



Demo organization



config

CMakeLists.txt

demo.repos

launch

package.xml

robot

records.yaml

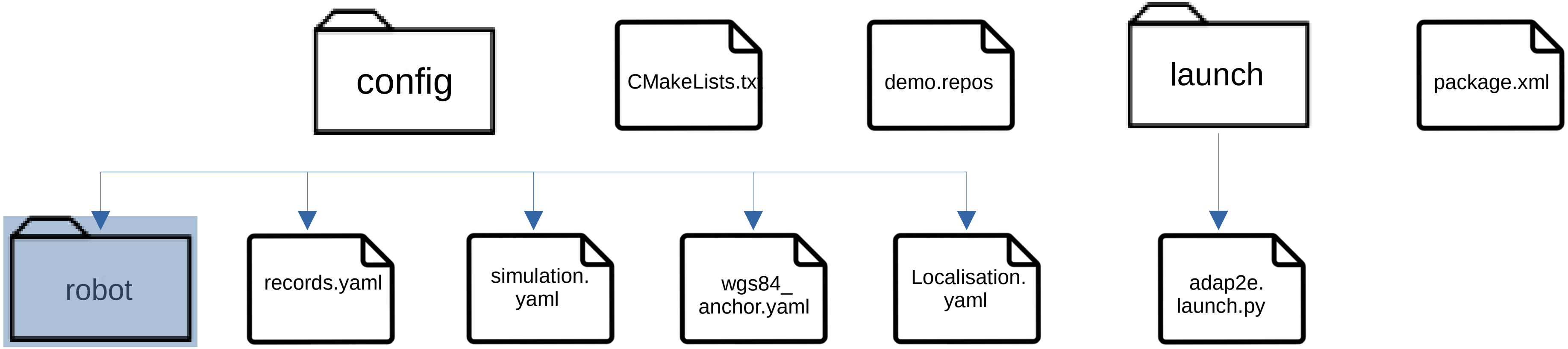
simulation.
yaml

wgs84_
anchor.yaml

Localisation.
yaml

adap2e.
launch.py

Demo organization



config

CMakeLists.txt

demo.repos

launch

package.xml

robot

records.yaml

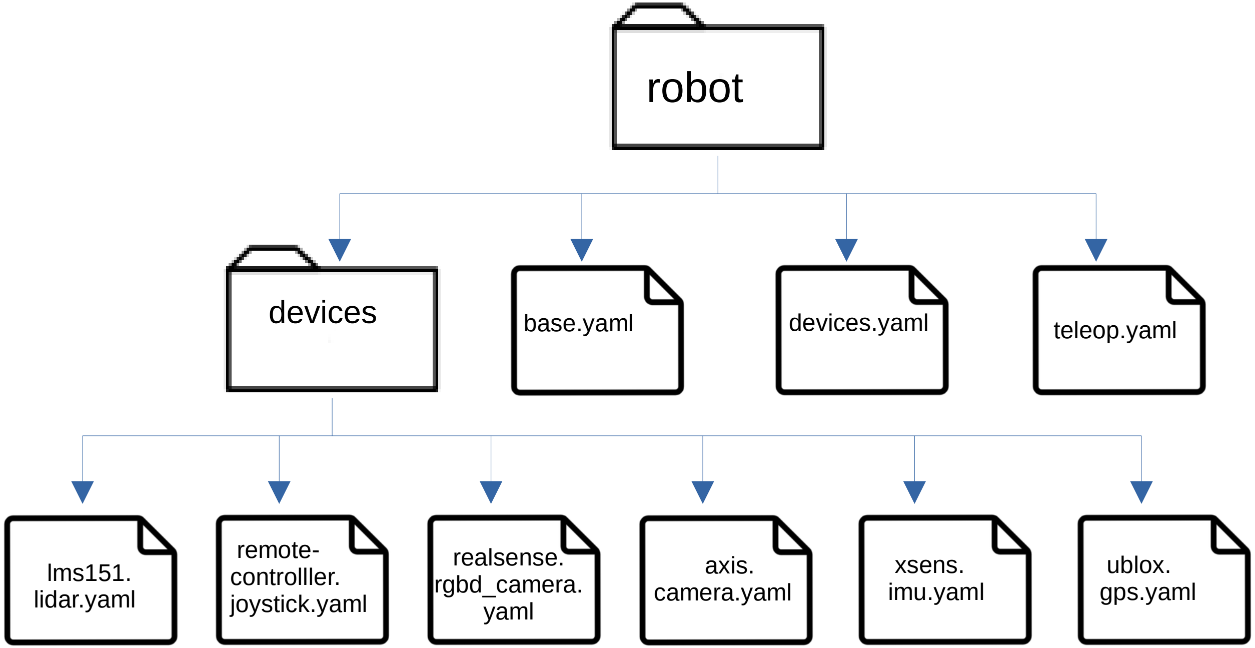
simulation.
yaml

wgs84_
anchor.yaml

Localisation.
yaml

adap2e.
launch.py

Robot configuration



URDF description :

```
ros2 run romea_demo robot_description.py
  mode:live or simulation
  robot_namespace:adap2e
  robot_configuration_directory:robot_directory_path
```

Launch :

```
ros2 launch romea_demo robot.launch.py
  mode:=live or simulation
  robot_namespace:=adap2e
  robot_configuration_directory:=robot_directory_path
  urdf_description:= user_description ( optional, by default
urdf_description parameter is the result of urdf description script)
```

More info :

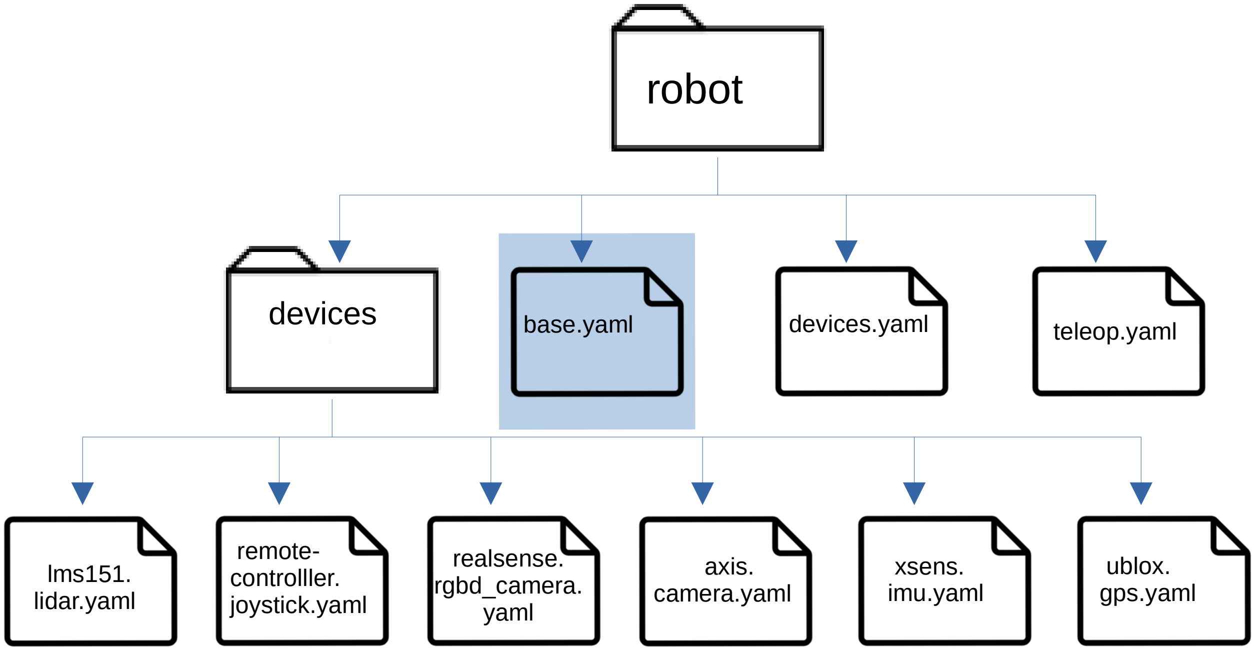
tirrex_demo package

Examples :

```
-ros2 run tirrex_demo robot_description.py mode:live robot_namespace:adap2e robot_configuration_directory:path_to_demo/config/robot  > robot.urdf
-ros2 launch tirrex_demo robot.launch.py mode:=live robot_namespace:=adap2e robot_configuration_directory:=path_to_demo/config/robot
 (warning in simulation mode gazebo must be launch before by using ros2 launch gazebo_ros gazebo.launch.py)
```

Robot configuration

Mobile base



robot

devices

base.yaml

devices.yaml

teleop.yaml

lms151.lidar.yaml

remote-controlller.joystick.yaml

realsense.rgbd_camera.yaml

axis.camera.yaml

xsens.imu.yaml

ublox.gps.yaml

Meta-description:

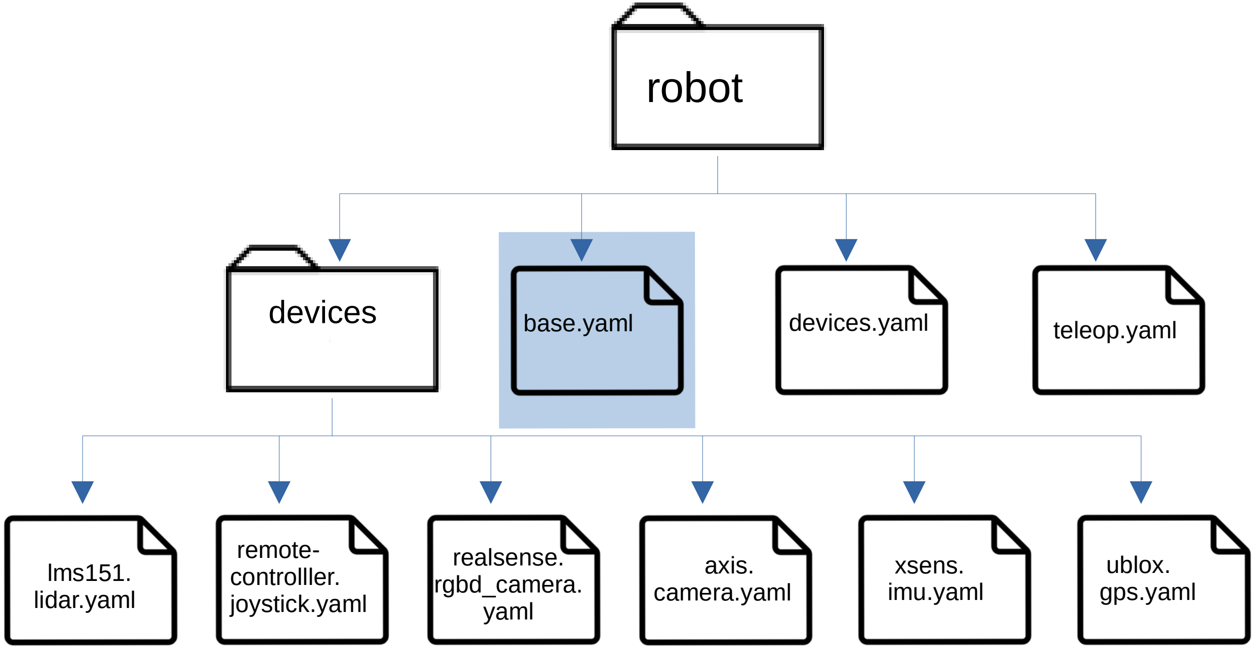
```
name: "adap2e"
configuration:
  type: adap2e
  model: fat
records:
  joint_states : true
  controller/odom : true
  controller/odometry : true
  controller/kinematic : true
simulation:
  initial_xyz: [0.0, 0.0, 0.0]  # meters
  initial_rpy: [0.0, 0.0, 0.0]  # degrees
```

Supported robots :

- adap2e (fat and slim models)
- alpo (pom and 4x4 models)
- aroco
- campero (rubber model)
- ceol
- robucar

Robot configuration

Mobile base



URDF Description:

ros2 run romea_mobile_base_bringup urdf_description.py
mode: live or simulation
robot_namespace:adap2e
meta_description_file_path:path to base meta-description file

Launch :

ros2 run romea_mobile_base_bringup mobile_base.launch.py
mode:=live or simulation
robot_namespace:=adap2e
meta_description_file_path:=path to base meta-description file

More info :

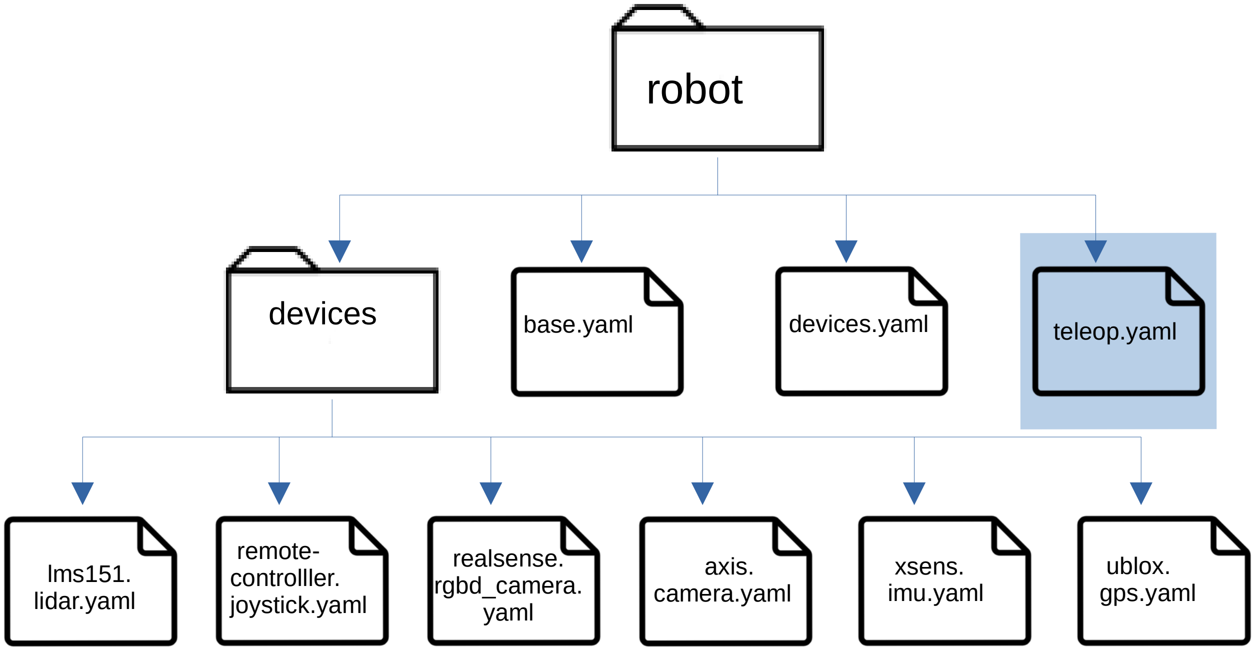
- romea_mobile_base stack

Examples :

-ros2 run romea_mobile_base_bringup urdf_description.py mode:live robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/base.yaml > base.urdf
-ros2 launch romea_mobile_base_bringup robot.launch.py mode:=live robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/base.yaml
(warning in simulation mode gazebo must be launch before by using ros2 launch gazebo_ros gazebo.launch.py)

Robot configuration

Teleoperation



Meta-description:

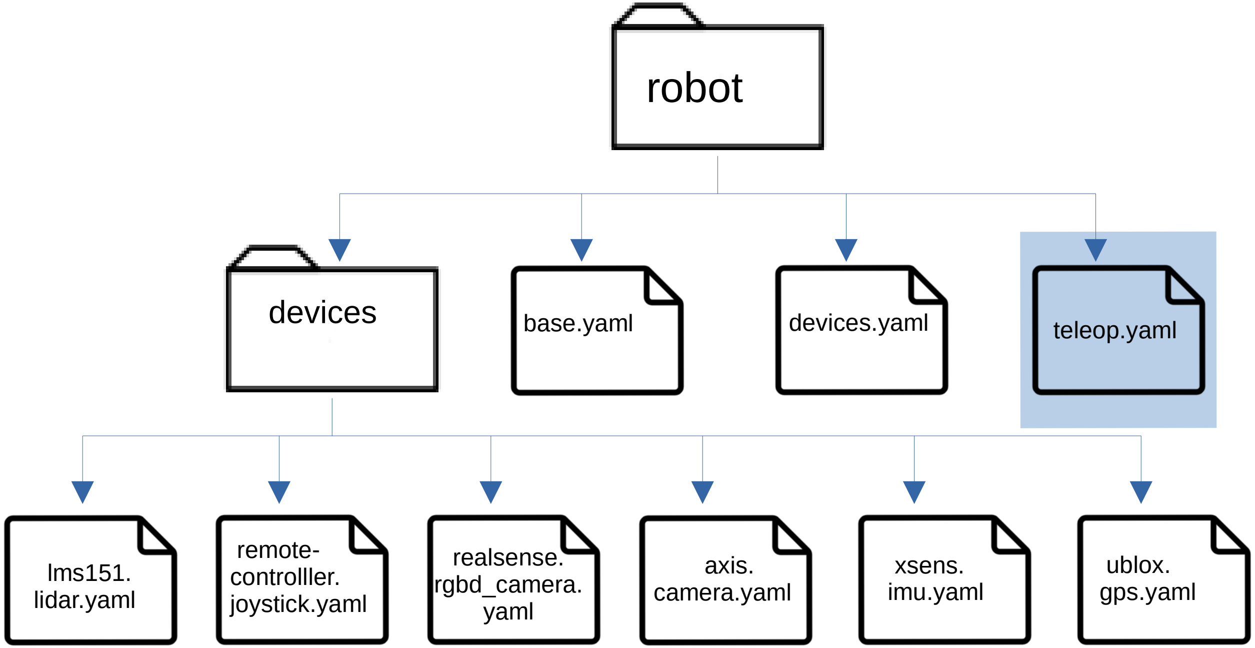
```
 cmd_output:
   message_type:    romea_mobile_base_msgs/
TwoAxleSteeringCommand
   message_priority: 100
 cmd_range:
   maximal_linear_speed:
     slow_mode: 1.0  # m/s
     turbo_mode : 2.0 # m/s
   #maximal_front_steering_angle: 20.0 # degree, optional
   #maximal_rear_steering_angle:  0.0 # degree, optional
```

Supported command type:

- one axle steering (alpo)
- two_alxle_steering (aroco,robucar,adap2e)
- skid_steering (campero rubber, ceol)
- omni_steering (campero mecanum)

Robot configuration

Teleoperation



Joystick mapping (TODO) :

```
ros2 run romea_teleop_bringup joystick_remapping.py
```

base_meta_description_file_path: path to mobile base meta-description file

joystick_meta_description_file_path: path to joystick meta-description file

teleop_configuration_file_path **:** path to teleop configuration file

Launch:

```
ros2 launch romea_teleop_bringup mobile_base_teleop.launch.py
```

robot_namespace:adap2e

base_meta_description_file_path: path to mobile base meta-description file

joystick_meta_description_file_path: path to joystick meta-description file

teleop_configuration_file_path **:** path to teleop configuration file

More info :

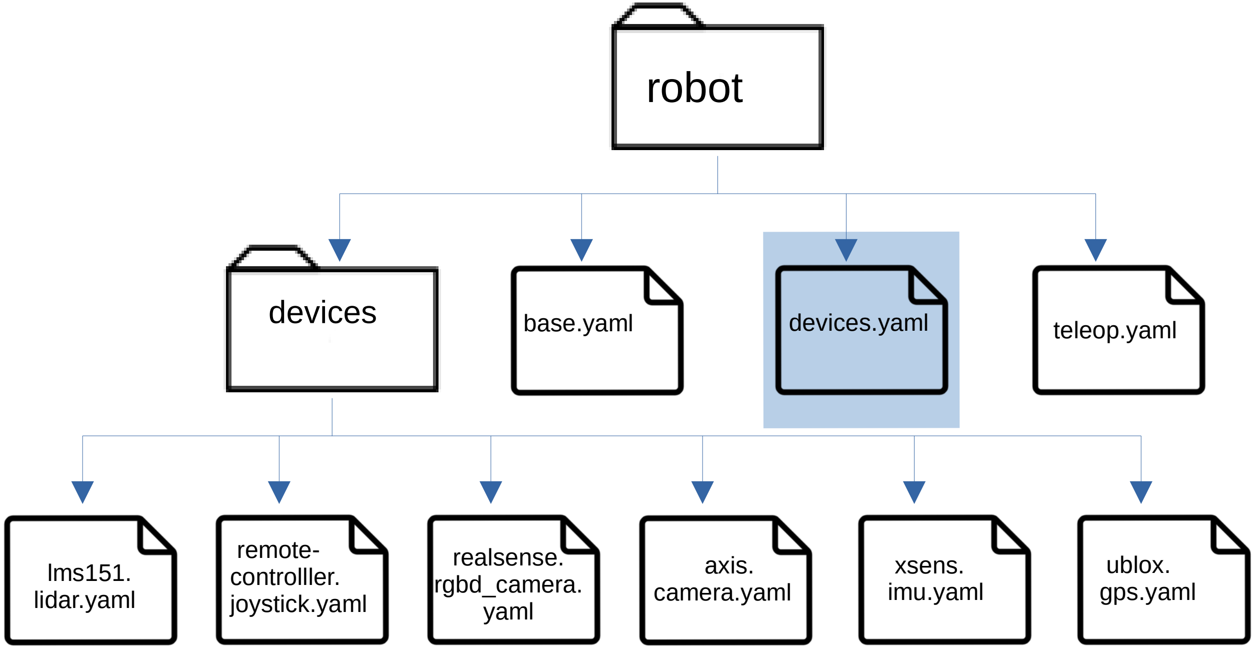
- **romea_teleop** stack

Examples :

```
-ros2 launch romea_teleop_bringup mobile_base_teleop.launch.py robot_namespace:=adap2e base_meta_description_file_path:=path_to_demo/config/robot/base.yaml joystick_meta_description_file_path:=path_to_demo/robot/devices/remote_controller.joytick.yaml  teleop_configuration_file_path:=path_to_demo/config/ robot/teleop.yaml
```

Robot configuration

Devices selection



Devices configuration:

```
remote_controller:
  type: joystick
  available_mode: all
lms151:
  type: lidar
  available_mode: live
ublox:
  type: gps
  available_mode: none
xsens:
  type: imu
  available_mode: none
transceiver0:
  type: rtls
  available_mode: simulation
transceiver1:
  type: rtls
  available_mode: simulation
```

More info :

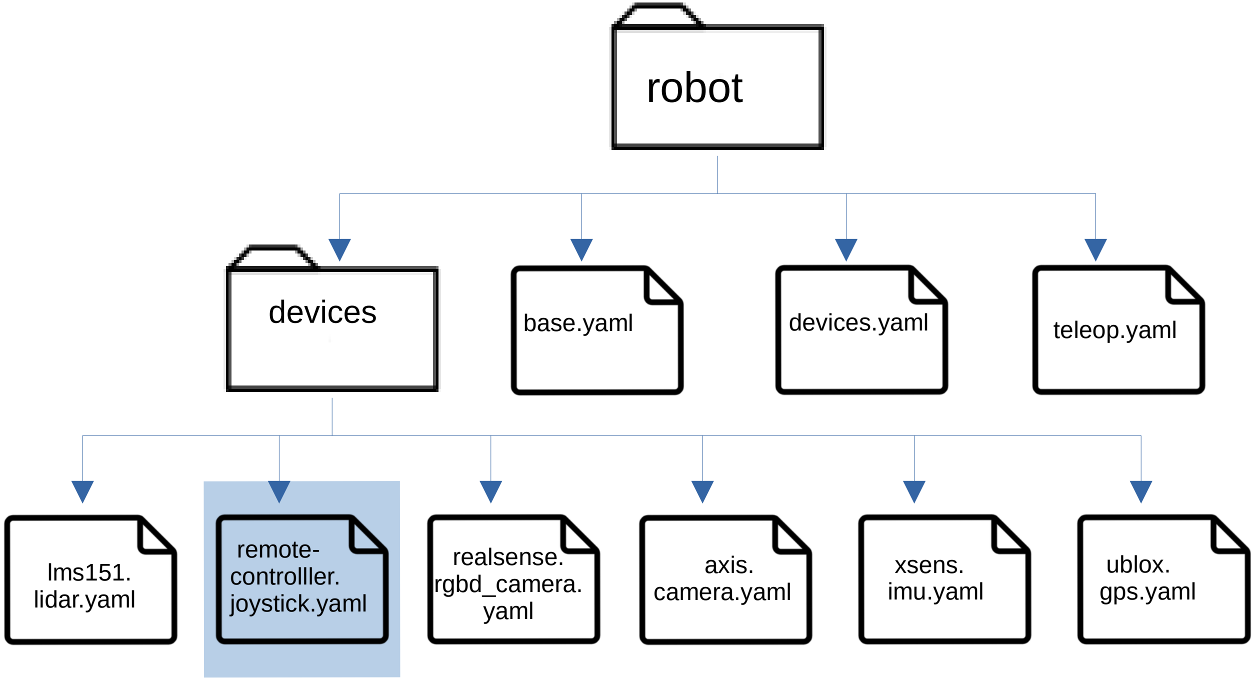
- tirrex_demo package

Examples :

```
-ros2 launch tirrex_demo robot_devices.launch.py configuration_filaneme:=path_to_demo/config/robot/devices.yaml
```

Robot configuration

Joystick



robot

devices

base.yaml

devices.yaml

teleop.yaml

lms151.
lidar.yaml

remote-
controlller.
joystick.yaml

realsense.
rgbd_camera.
yaml

axis.
camera.yaml

xsens.
imu.yaml

ublox.
gps.yaml

Meta-description :

```
name: "joystick"
driver:
  pkg: "joy"
  device: "/dev/input/js0"
  autorepeat_rate: 10.0
  deadzone: 0.1
configuration:
  type: xbox
records:
  joy: true
```

Supported driver packages:

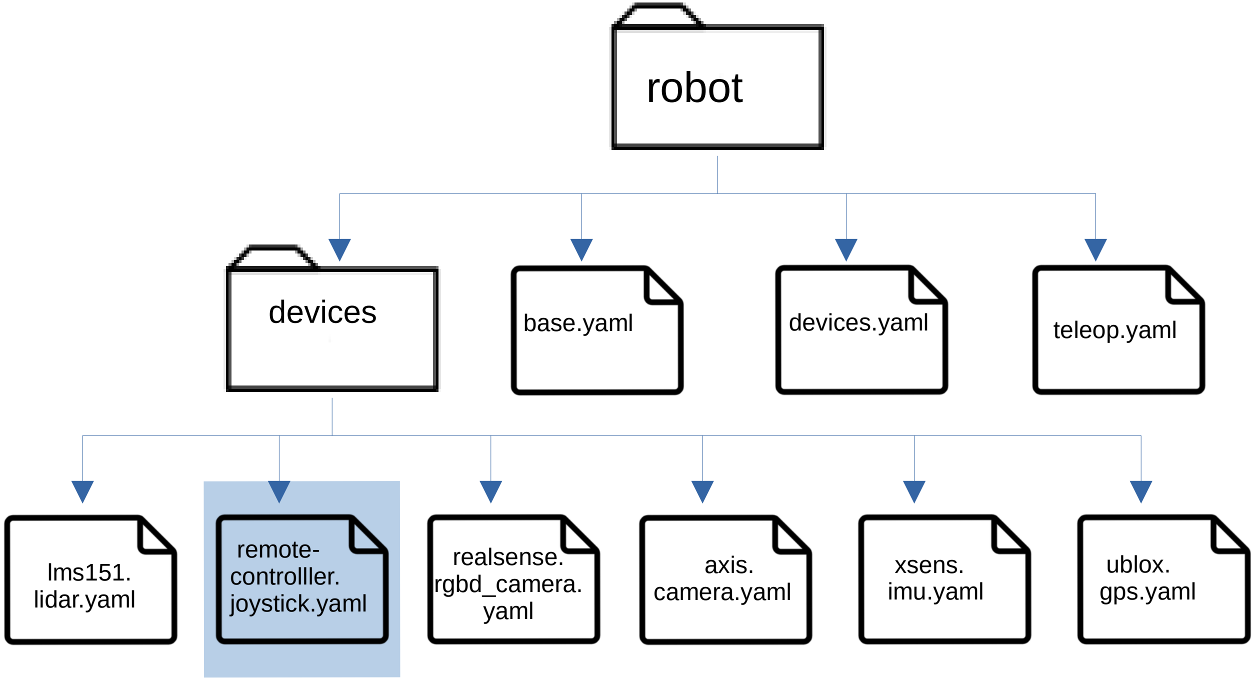
- joy
- ds4_driver

Supported joysticks:

- xbox (360, one)
- dualshock4

Robot configuration

Joystick



Launch :

ros2 launch romea_joy_bringup joystick_driver.launch.py

robot_namespace:=adap2e (optional, default = "")

meta_description_file_path:= path to joystick meta-description file

More infos :

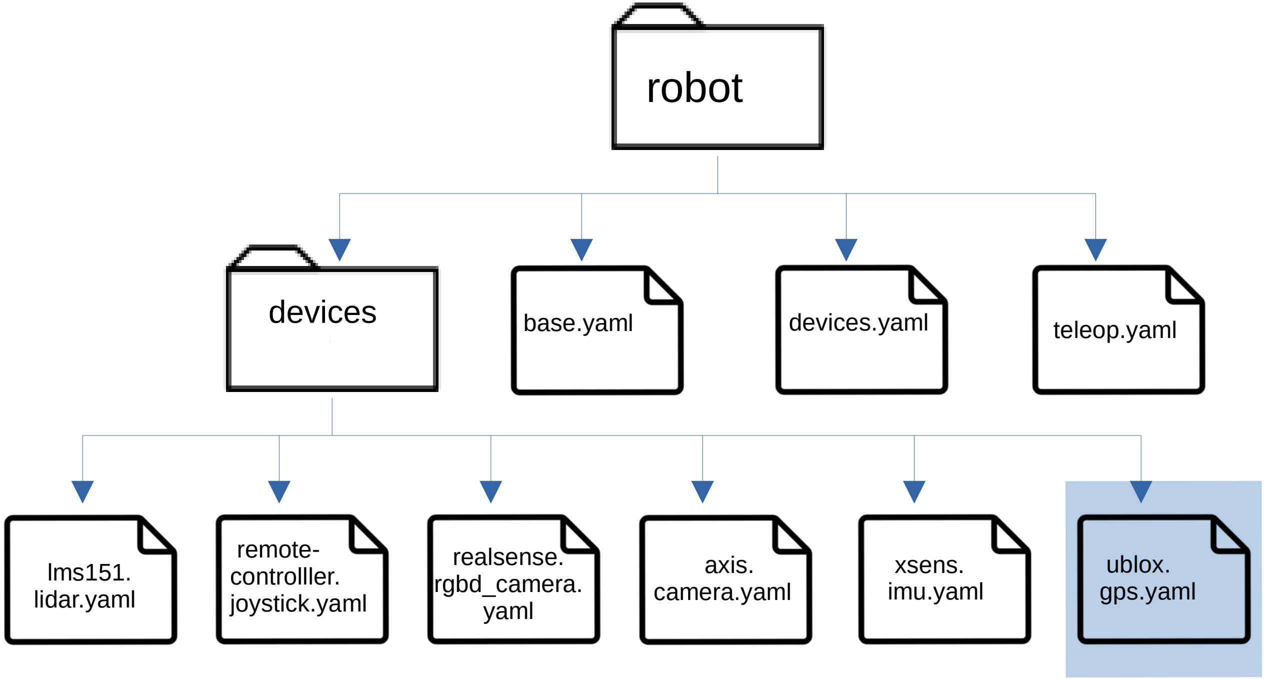
-romea joy stack

Example :

-ros2 launch romea_joystick_bringup joystick_driver.launch.py robot_namespace:=adap2e meta_description_filename:=path_to_demo/robot/config/devices/remote-controller.joystick.yaml

Robot configuration

GPS



Meta-description :

```
  name: "gps"
  driver:
    pkg: "romea_ublox_driver"
    device:  "/dev/ttyACM0"
    baudrate: 115200
  ntrip:  # optional
    pkg: "ntrip_client"
    host: caster.centipede.fr
    port: 2101
    username: centipede # optional
    password: centipede # optional
    mountpoint: MAGC
 configuration:
    type: drotek
    model: f9p
    rate: 10 # hz
geometry:
  parent_link: "base_link"
  xyz: [0.0, 0.0, 1.5]  #meters
records:
  nmea_sentence: true
  gps_fix: false
  vel: false
```

Supported driver packages :

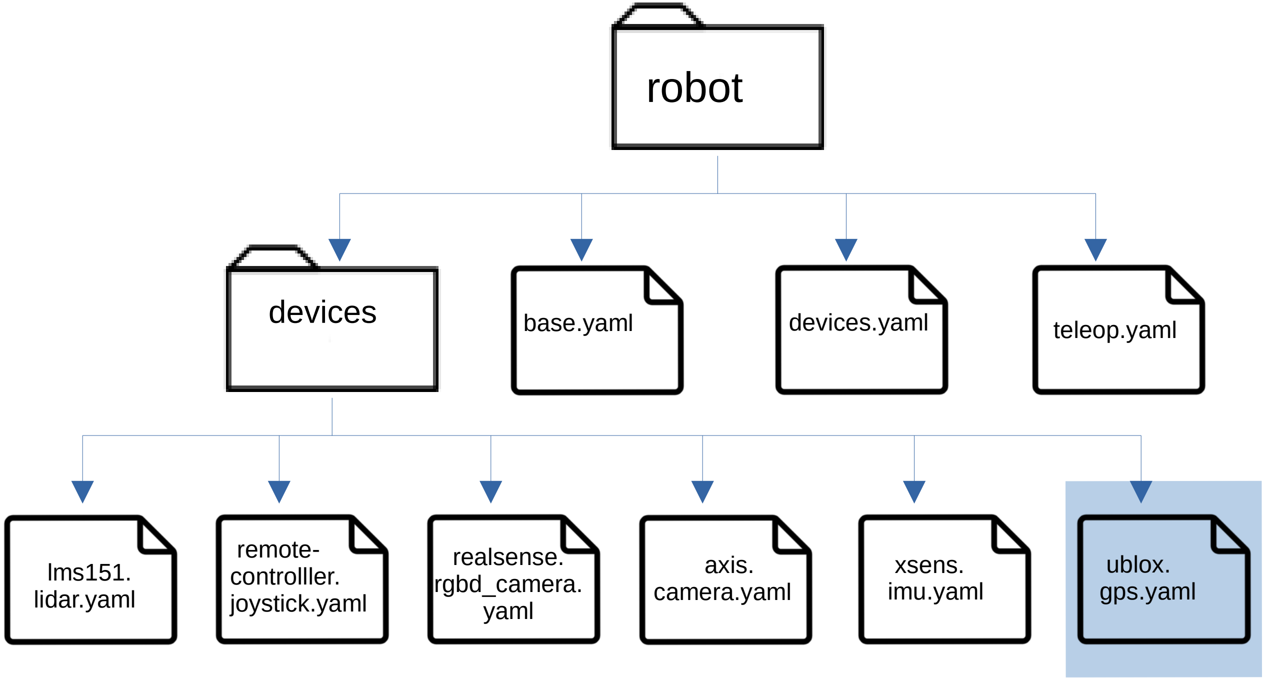
- nmea_navsat_driver
- romea_ublox_driver
- ntrip_client

Supported GPS receivers

- ublox **(** drotek fp9, ublox evk m8)
- ashtech proflex800

Robot configuration

GPS



URDF Description :

ros2 run roma_gps_bringup urdf_description.py
robot_namespace:adap2e
meta_description_file_path: absolute path to gps meta-description file

Launch :

ros2 run roma_gps_bringup gps_driver.launch.py
robot_namespace:=adap2e
meta_description_file_path:= absolute path to gps meta-description file

More info :

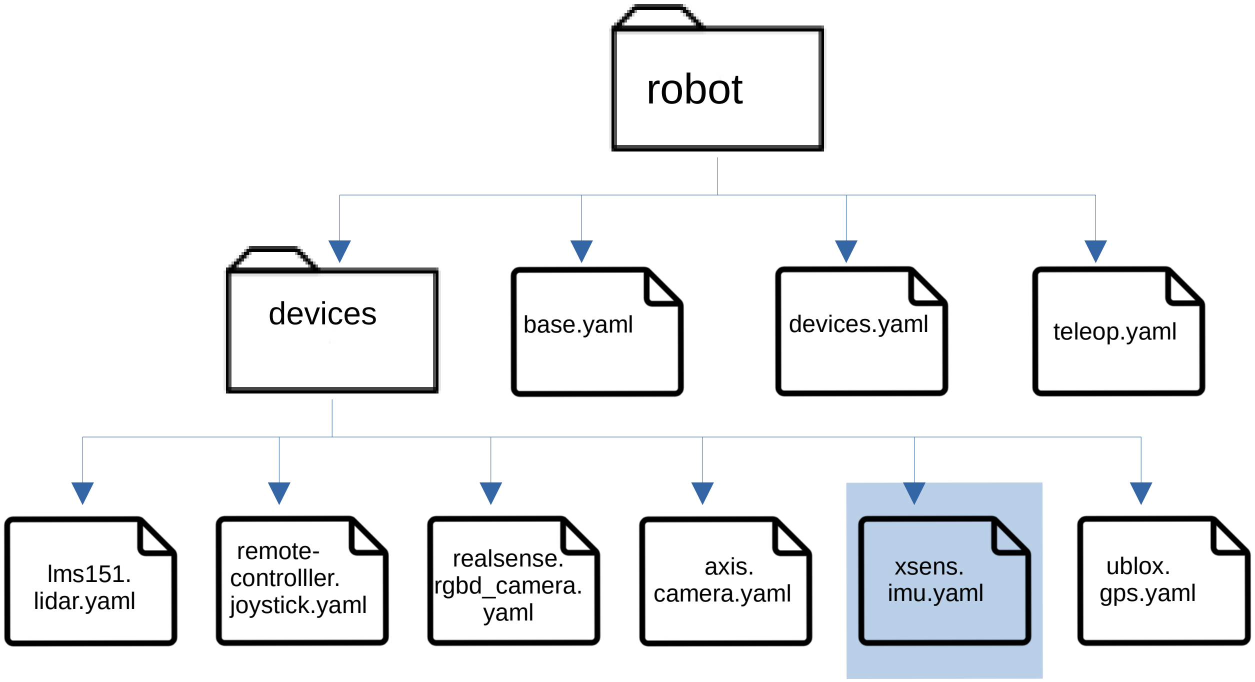
- romea gps stack

Examples :

-ros2 run romea_gps_bringup urdf_description.py robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/ublox.gps.yaml > ublox.urdf
-ros2 launch romea_gps_bringup gps_driver.launch.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/ublox.gps.yaml

Robot configuration

IMU



Meta-description :

```
name: "imu"
driver:
  pkg: "xsens_driver"
  device:  "/dev/ttyUSB0"
  baudrate: 115200
configuration:
  type: xsens
  model: mti
  rate: 100  #hz
geometry:
  parent_link: "base_link"
  xyz: [0.0, 0.0, 1.0]  #meters
  rpy: [0.0, 0.0, 0.0]  #degrees
records:
  data: true
```

Supported driver packages :

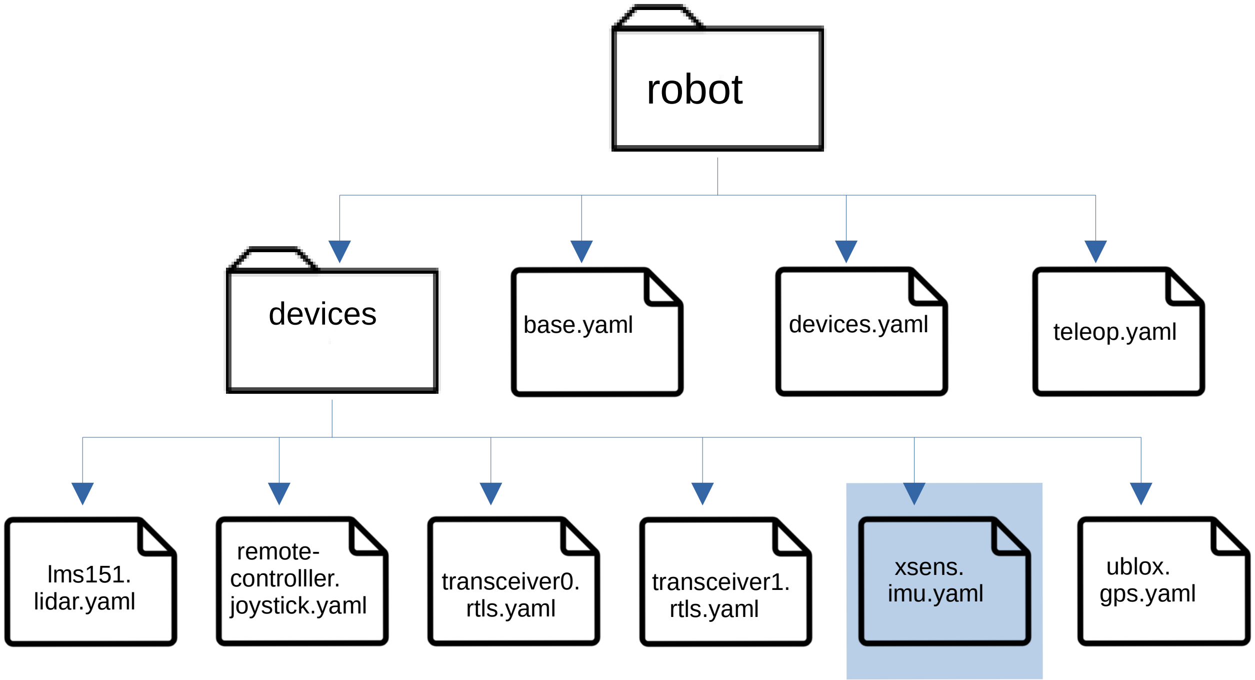
- bluespace_ai_xsens_mti_driver
- xsens_driver

Supported IMU sensors:

- xsens (models mti and mti6xx)

Robot configuration

IMU



URDF Description :

ros2 run roma_imu_bringup urdf_description.py
robot_namespace:adap2e
meta_description_filename:path to imu meta-description file

Launch :

ros2 run roma_imu_bringup imu_driver.launch.py
robot_namespace:=adap2e
meta_description_filename:=path to imu meta-description file

More info :

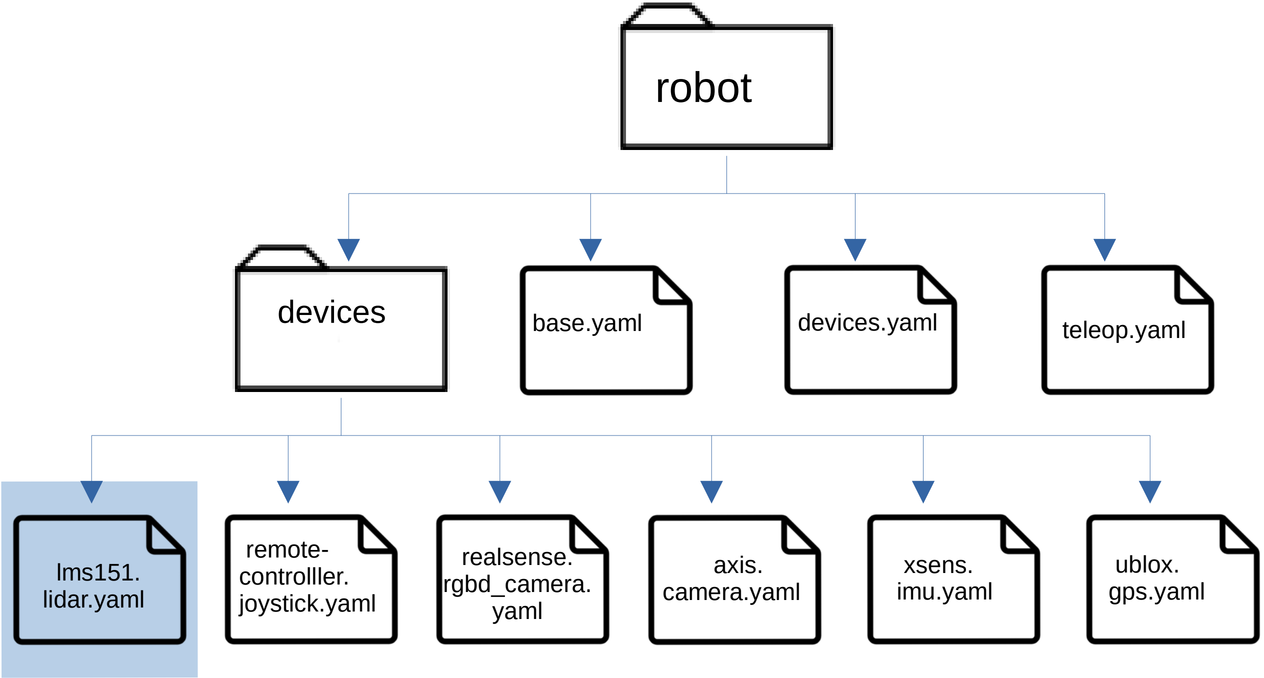
- romea imu stack

Examples :

-ros2 run romea_imu_bringup urdf_description.py robot_namespace:adap2e meta_description_filanem:path_to_demo/config/robot/devices/xsens.imu.yaml > xsens.urdf
-ros2 launch romea_imu_bringup imu_driver.launch.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/devices/xsens.imu.yaml

Robot configuration

LIDAR



Meta-description:

```
 name: "lidar"
 driver:
   pkg: "sick_scan"
   ip:  "192.168.1.112"
   port: 2112
 configuration:
   type: sick
   model: lms151
   rate: 50  # hz
   resolution:  0.5 # degree
 geometry:
   parent_link: "base_link"
   xyz: [2.02, 0.0, 0.34]  # meters
   rpy: [0.0, 0.0, 0.0]  # degrees
 records:
   scan: true
   cloud: false
```

Supported driver packages :

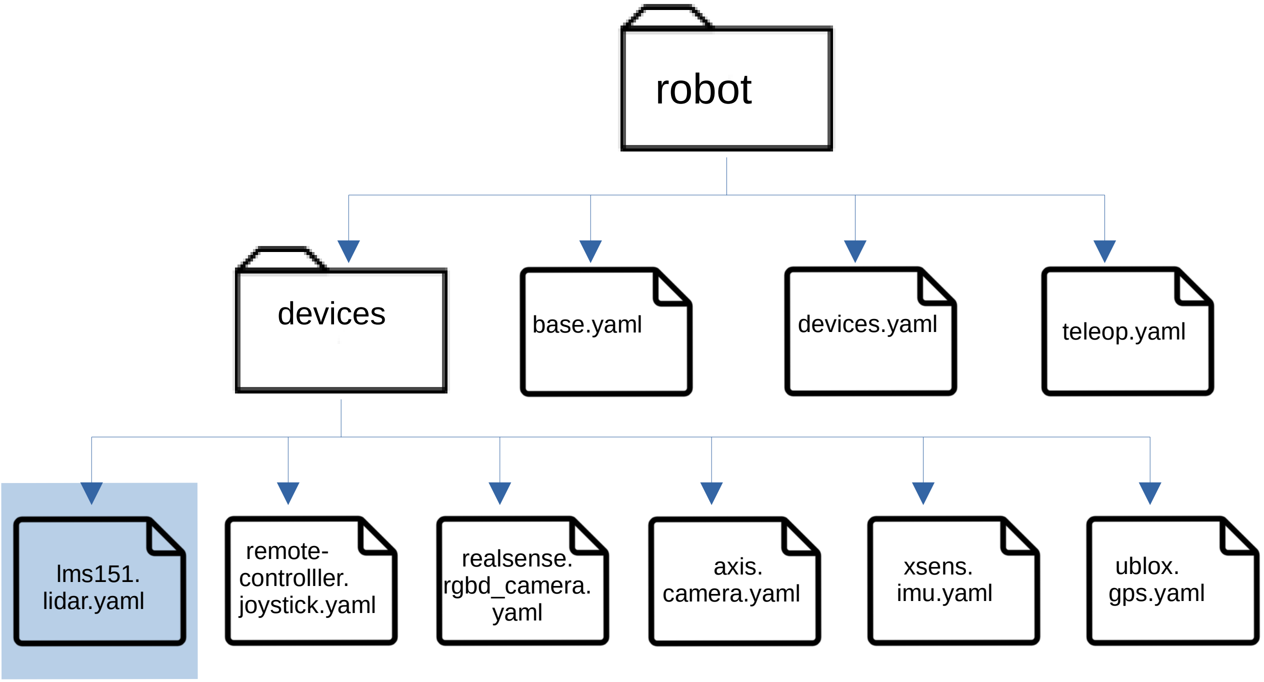
- sick_scan

Supported lidars :

- **sick (lms1xx,tim5xx)**

Robot configuration

LIDAR



robot

devices

base.yaml

devices.yaml

teleop.yaml

lms151.lidar.yaml

remote-controlller.joystick.yaml

realsense.rgbd_camera.yaml

axis.camera.yaml

xsens.imu.yaml

ublox.gps.yaml

URDF Description :

```
ros2 run roma_lidar_bringup urdf_description.py
  robot_namespace:adap2e
  meta_description_filename:path to lidar meta-description file
```

Launch :

```
ros2 run roma_lidar_bringup lidar_driver.launch.py
  robot_namespace:=adap2e
  meta_description_filename:=path to lidar meta-description file
```

More info :

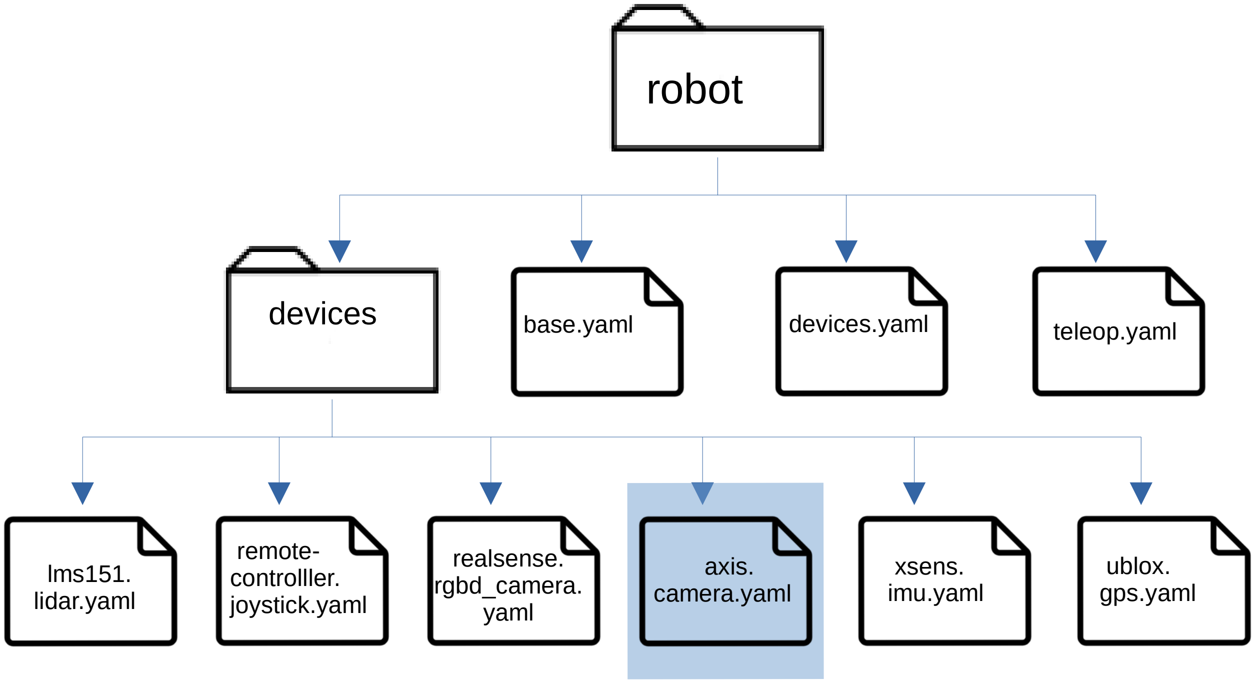
- **romea_lidar_stack**

Examples :

```
-ros2 run romea_lidar_bringup urdf_description.py robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/lms151.lidar.yaml  > lms151.urdf
-ros2 launch romea_lidar_bringup lidar_driver.launch.py robot_namespace:=adap2e meta_description_file_path:path_to_demo/config/robot//lms151.lidar.yaml
```

Robot configuration

Camera



robot

devices

base.yaml

devices.yaml

teleop.yaml

lms151.
lidar.yaml

remote-
controllller.
joystick.yaml

realsense.
rgbd_camera.
yaml

axis.
camera.yaml

xsens.
imu.yaml

ublox.
gps.yaml

Meta-description:

```
 name: camera
 driver: coming soon
 configuration:
   type: axis
   model: p1346
   frame_rate: 30  # hz
   resolution:  1280x720
 geometry:
   parent_link: "base_link"
   xyz: [1.02, 0.0, 1.34]  # meters
   rpy: [0.0, 20.0, 0.0]  # degrees
 records:
   camera_info: false
   image_raw: true
```

Supported driver packages :

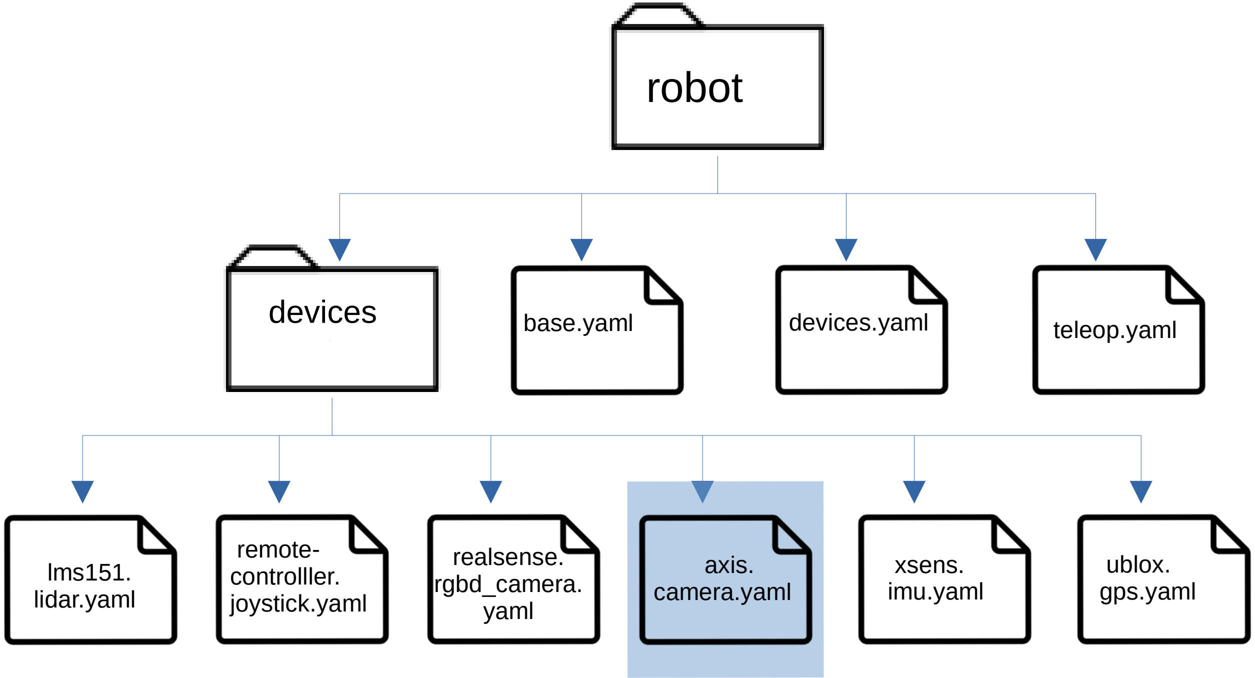
- **coming soon**

Supported cameras:

- **axis 134x**

Robot configuration

Camera



URDF Description :

ros2 run romea_camera_bringup urdf_description.py
robot_namespace:adap2e
meta_description_filename:path to camera meta-description file

Launch :

ros2 run romea_camera_bringup camera_driver.launch.py
robot_namespace:=adap2e
meta_description_filename:=path to camera meta-description file

More info :

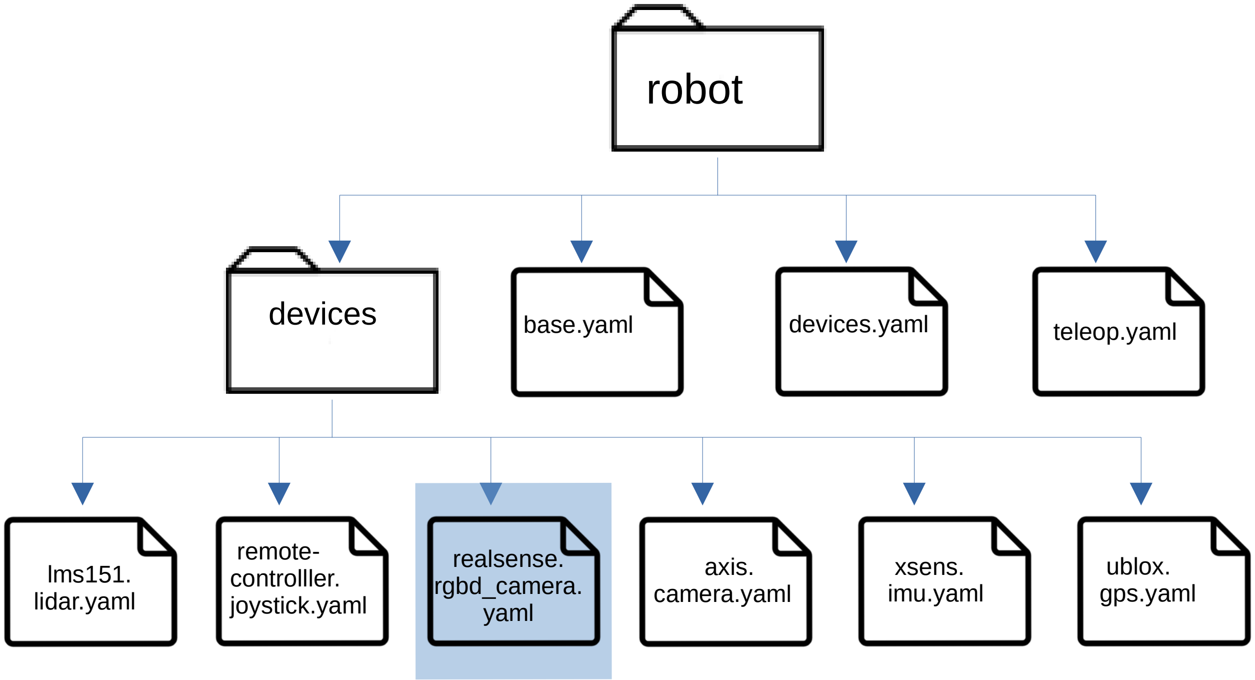
- **romea_camera** stack

Examples :

-ros2 run romea_camera_bringup urdf_description.py robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/axis.camera.yaml > axis.urdf
-ros2 launch romea_camera_bringup camera_driver.launch.py robot_namespace:=adap2e meta_description_file_path:path_to_demo/config/robot/axis.camera.yaml

Robot configuration

RGBD Camera



Meta-description:

```
name: rgbd_camera
driver: coming soon
configuration:
  Type: realsense
  model: d435
  rgb_camera :
      frame_rate: 30  # hz
      resolution:  1280x720
  infrared_camera :
      frame_rate: 30  # hz
      resolution:  1280x720
  depth_camera :
      frame_rate: 30  # hz
      resolution:  1280x720
geometry:
  parent_link: "base_link"
  xyz: [1.02, 0.0, 1.34]  # meters
  rpy: [0.0, 20.0, 0.0]  # degrees
records:
  camera_info: false
  image_raw: true
```

Supported driver packages :

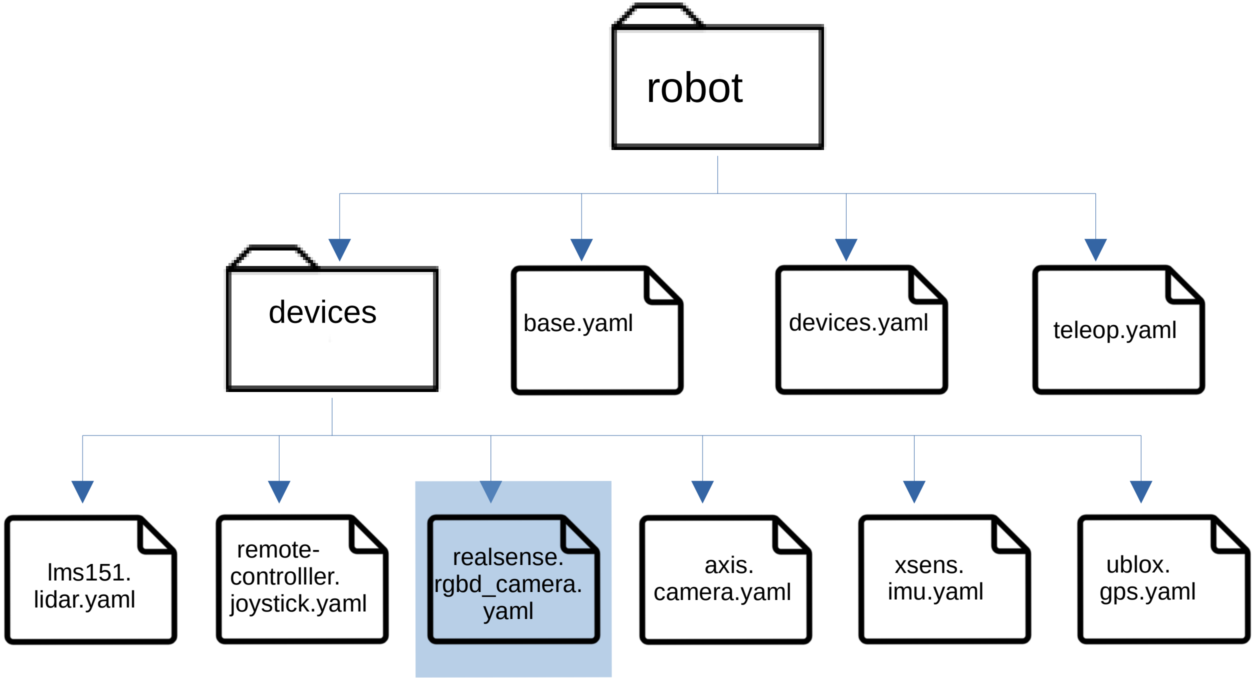
- coming soon

Supported cameras:

- realsense d43x
- zed 1 and 2

Robot configuration

RGBD Camera



URDF Description :

```
ros2 run romea_rgbd_camera_bringup urdf_description.py
  robot_namespace:adap2e
  meta_description_filename:path to camera meta-description file
```

Launch :

```
ros2 run romea_rgbd_camera_bringup  rgbd_camera_driver.launch.py
  robot_namespace:=adap2e
  meta_description_filename:=path to camera meta-description file
```

More info :

- **romea_rgbd_camera** stack

Examples :

```
-ros2 run romea_rgbd_camera_bringup urdf_description.py robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/realsense.yaml  > realsense.urdf
-ros2 launch romea_rgbd_camera_bringup camera_driver.launch.py robot_namespace:=adap2e meta_description_file_path:path_to_demo/config/robot/realsense.yaml
```

Robot configuration

Bridge (Campero, Alpo….)

Adap2e

Meta-description:

```
name: "lidar"
driver:
  pkg: "sick_scan"
  ip:  "192.168.1.112"
  port: 2112
configuration:
  type: sick
  model: lms151
  rate: 50  # hz
  resolution:  0.5 # degree
geometry:
  parent_link: "base_link"
  xyz: [2.02, 0.0, 0.34]  # meters
  rpy: [0.0, 0.0, 0.0]  # degrees
records:
  scan: true
  cloud: false
```

Campero

Meta-description:

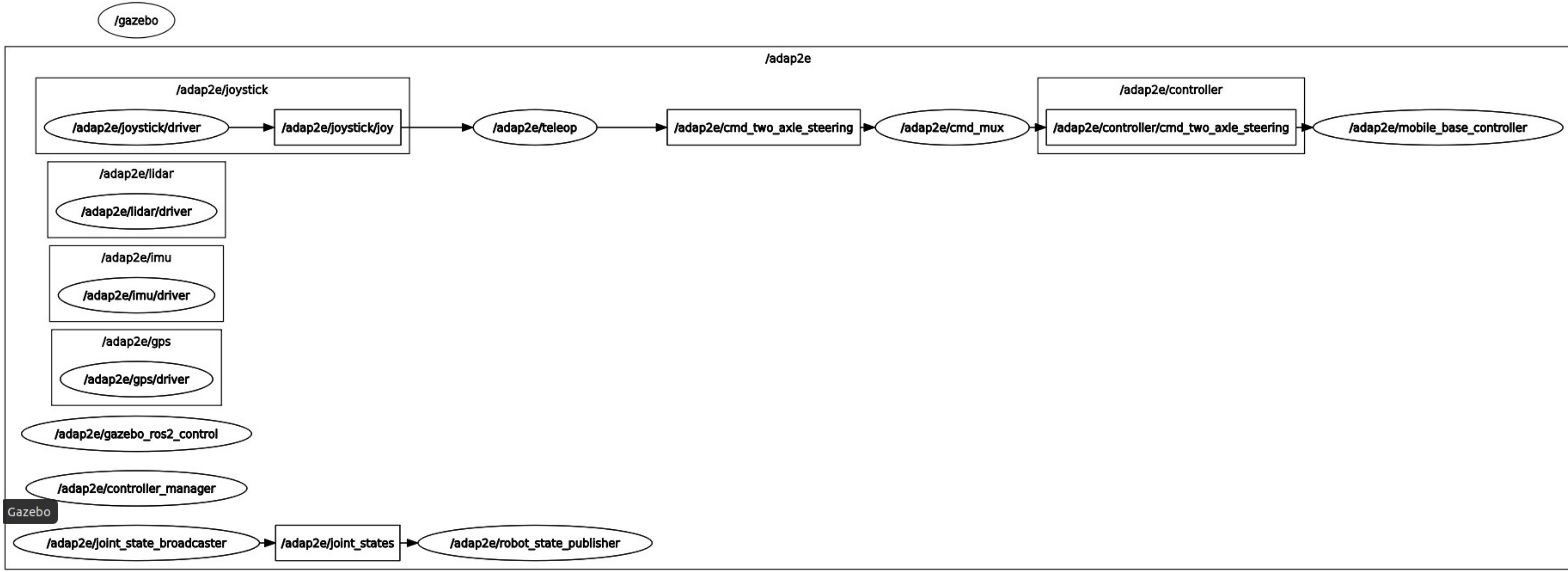
```
name: "front_lidar"
configuration:
  type: sick
  model: lms151
  rate: 50  # hz
  resolution:  0.5 # degree
geometry:
  parent_link: "base_link"
  xyz: [0.490, -0.300, 0.3513]  # meters
  rpy: [180.0, 0.0, -45.0]  # degrees
records:
  scan: true
bridge:
  scan: /campero_bridge/front_laser/scan
```

Complete remapping used by algorithms, applications…
/robot_name/front_lidar/scan : /campero_bridge/front_laser/scan

❗ Only a driver or a bridge into a Meta-description

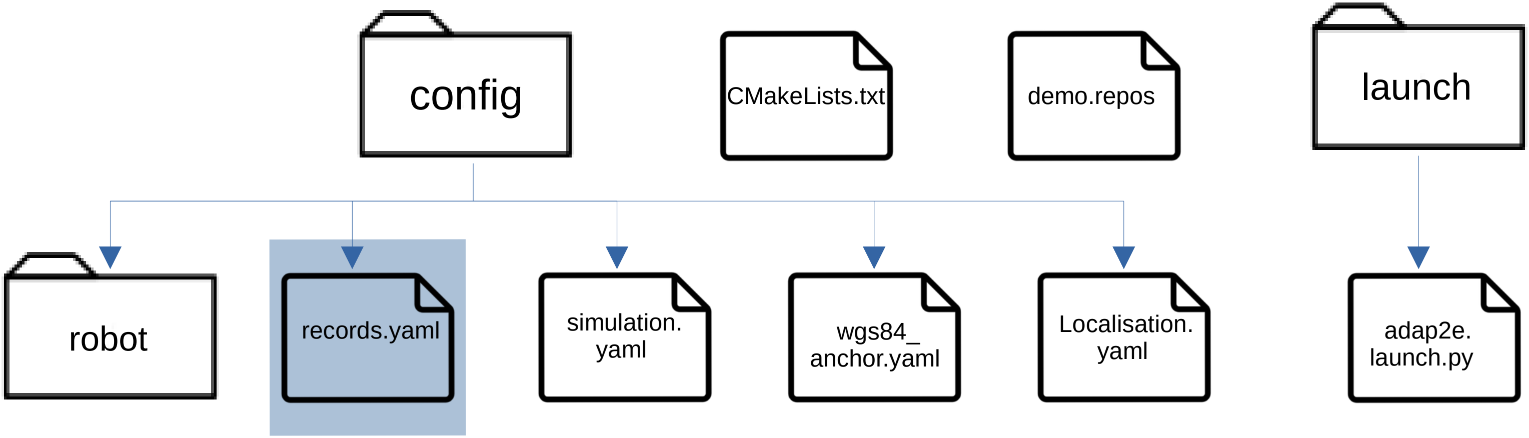
Robot Simulation

Nodes graph



Demo configuration

Record/Replay



Records configuration:

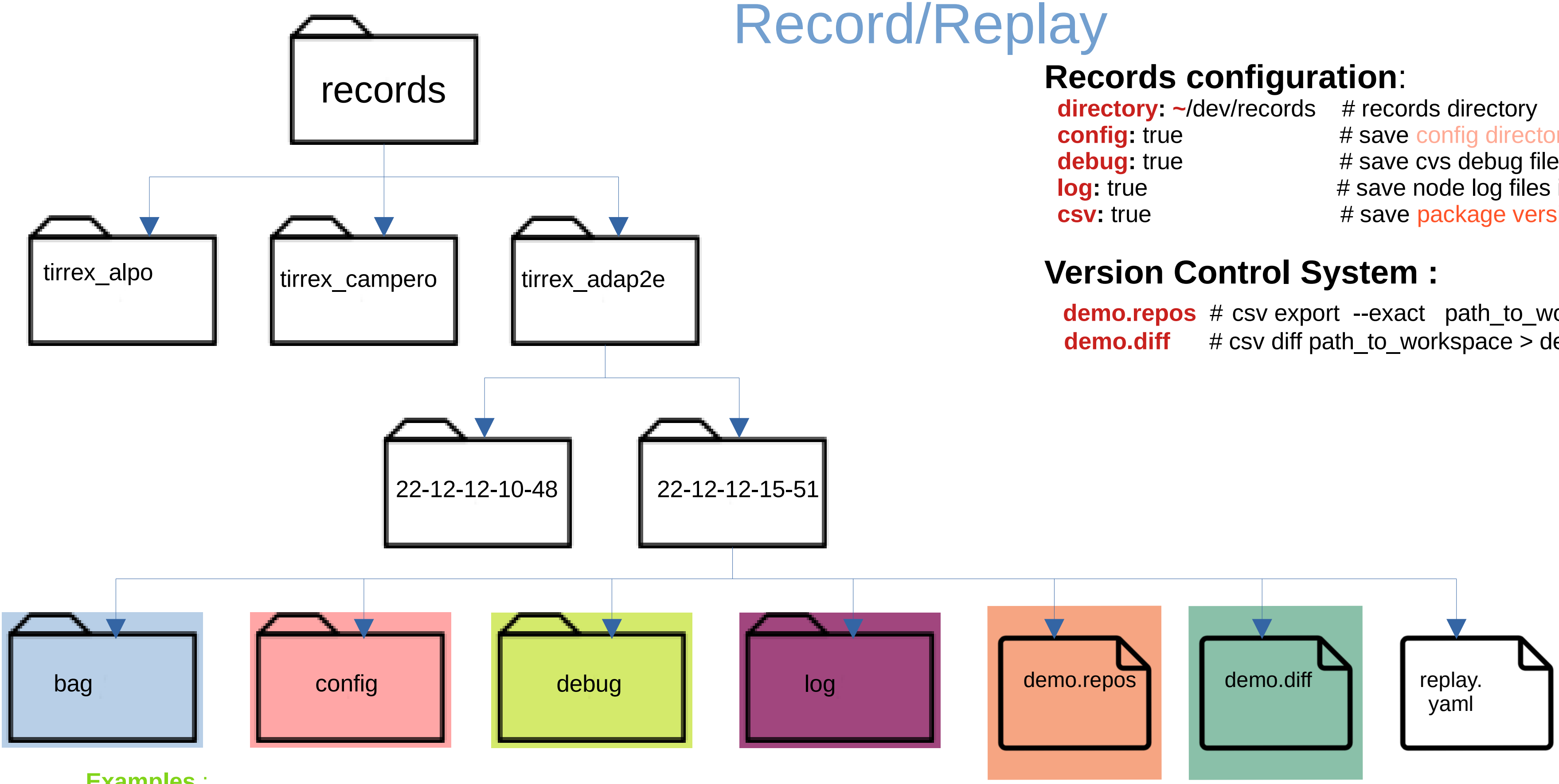
```
directory: ~/dev/records    # records directory
config: true                # save config directory
debug: true                 # save cvs debug files into debug directory
log: true                   # save node log files into log directory
csv: true                   # save package versions and source codes diff
```

Examples :

- ros2 launch tirrex_adap2e adap2e.launch.py mode:=simulation record:= true

Demo configuration

Record/Replay



Records configuration:

```
directory: ~/dev/records    # records directory
config: true                # save config directory
debug: true                 # save cvs debug files into debug director
log: true                   # save node log files into log directory
csv: true                   # save package versions and source codes diff
```

Version Control System :

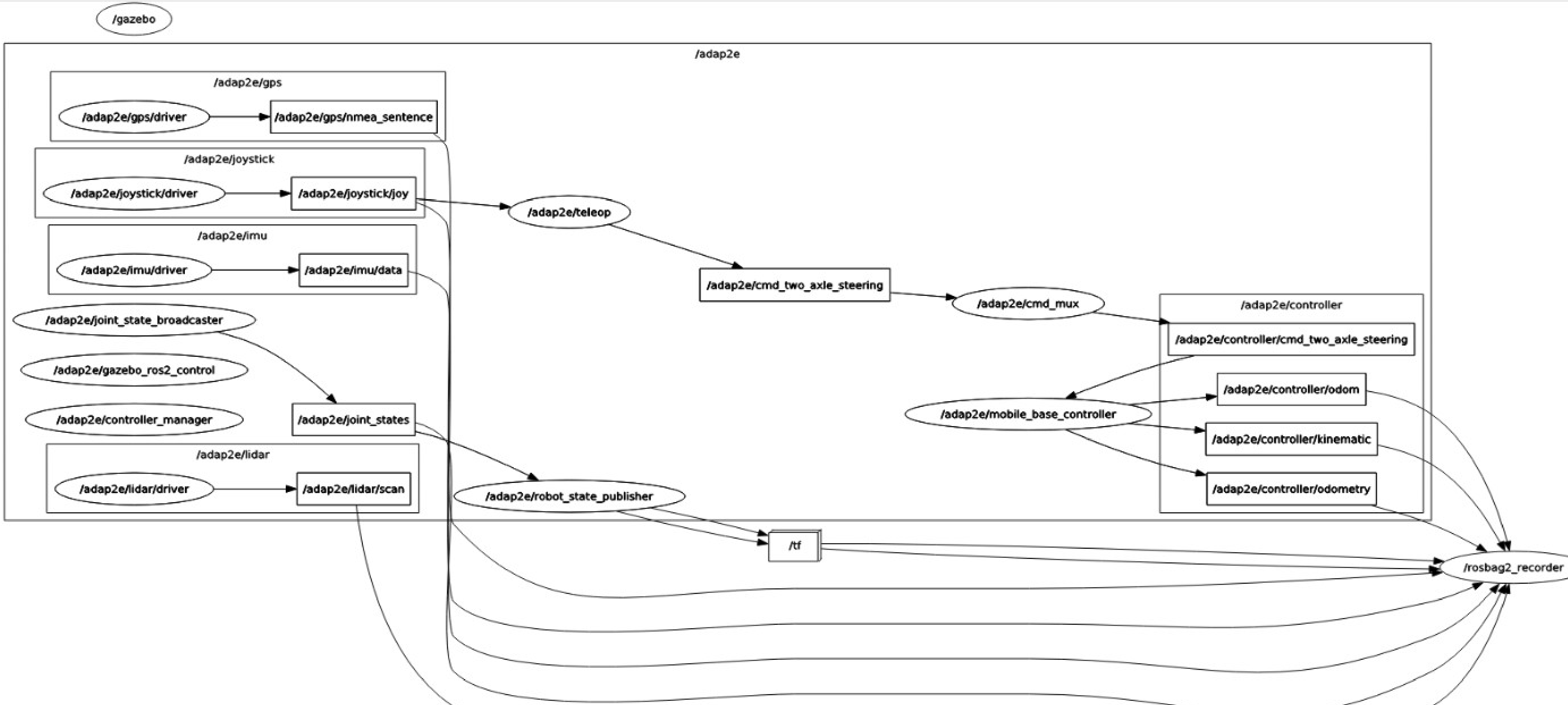
```
demo.repos  # csv export  --exact  path_to_workspace >  demo.repos
demo.diff   # csv diff path_to_workspace > demo.diff
```

Examples :

- ros2 launch tirrex_adap2e adp2e.launch.py mode:=simulation record:= true
- ros2 launch tirrex_demo replay.launch.py   replay_directory:=~dev/records/tirrex_adap2e/22-12-12-15-51

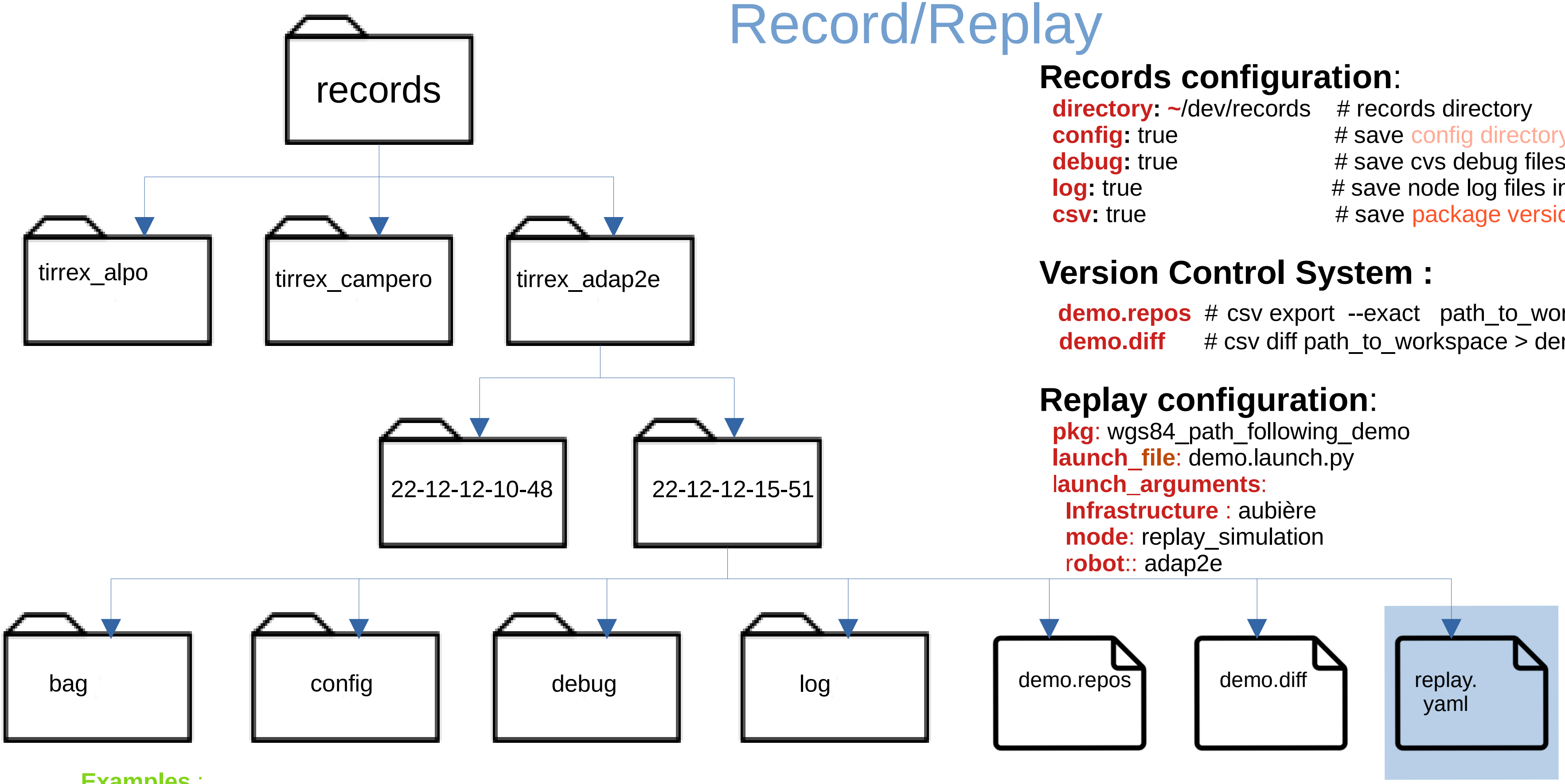
Demo configuration

Record/Replay



Demo configuration

Record/Replay



records

tirrex_alpo

tirrex_campero

tirrex_adap2e

22-12-12-10-48

22-12-12-15-51

bag

config

debug

log

demo.repos

demo.diff

replay.
yaml

Records configuration:

```
directory: ~/dev/records      # records directory
config: true                  # save config directory
debug: true                   # save cvs debug files into debug director
log: true                     # save node log files into log directory
csv: true                     # save package versions and source codes diff
```

Version Control System :

```
demo.repos  # csv export  --exact   path_to_workspace >   demo.repos
demo.diff   # csv diff path_to_workspace > demo.diff
```

Replay configuration:

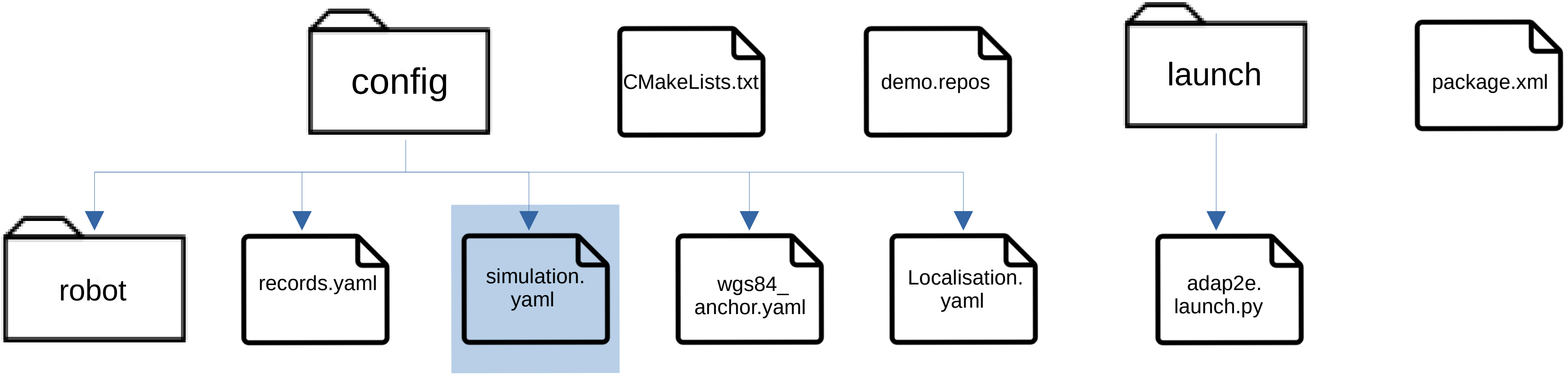
```
pkg: wgs84_path_following_demo
launch_file: demo.launch.py
launch_arguments:
  Infrastructure : aubière
  mode: replay_simulation
  robot:: adap2e
```

Examples :

- ros2 launch tiirex_adap2e adap2e.launch.py mode:=simulation records:= true
- ros2 launch romea_demo replay.launch.py   replay_directory:~dev/records/rtirrex_adap2e/22-12-12-15-51

Demo configuration

Simulation configuration



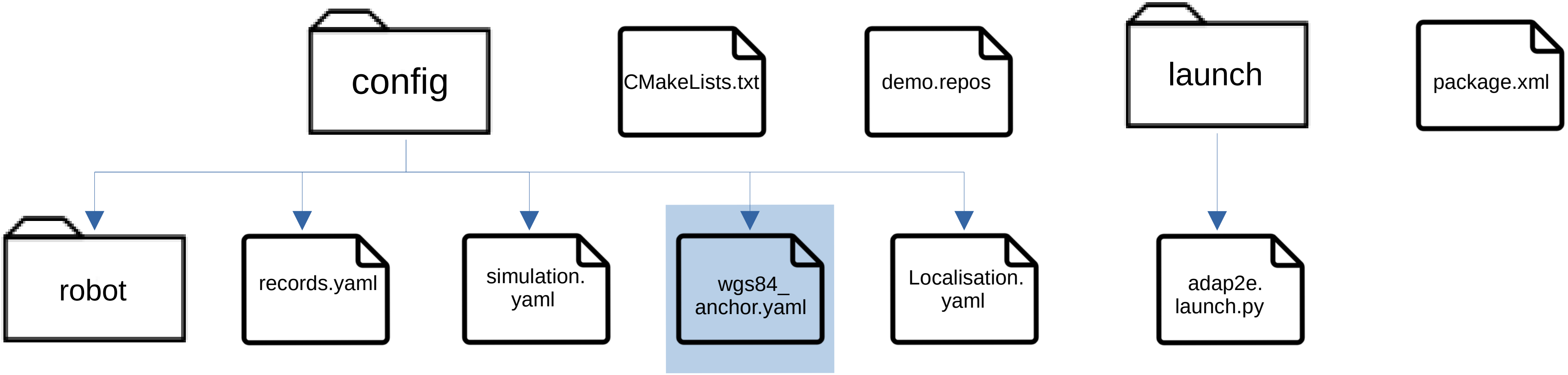
Simulation configuration :
world_package: romea_simulation_gazebo_worlds
world_name: romea_small_vineyard.world

More Info :
romea_simulation stack

Examples :
-ros2 launch romea_simulation_bringup simulator.launch.py simulation_configuration_file_path:=path_to_demo/config/simulation.yaml simulator_type :=gazebo

Demo configuration

WGS84 reference point configuration

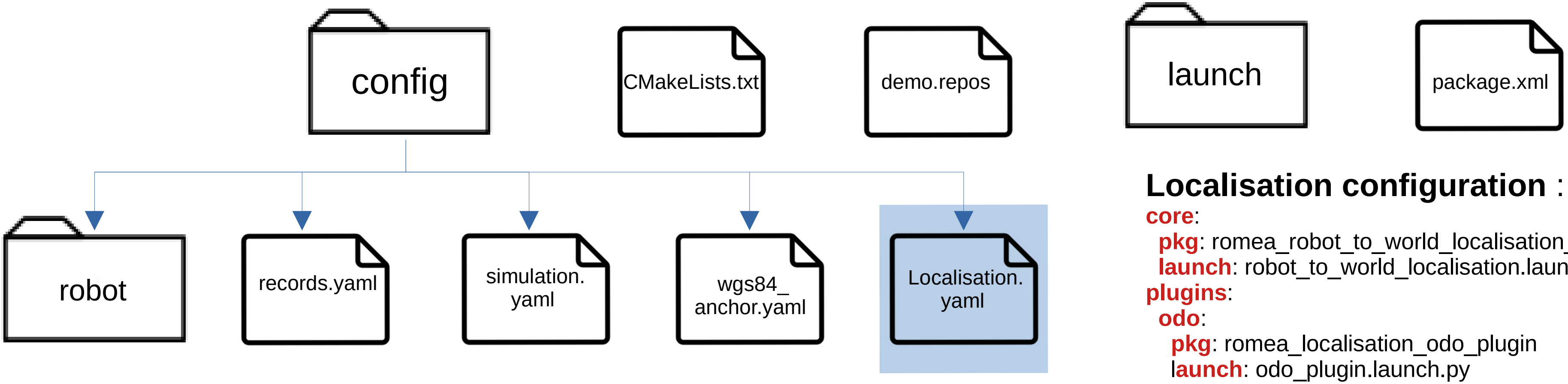


WGS84 anchor :

latitude: 45.76265802
longitude: 3.11000985
Altitude: 405.839

Demo configuration

Localisation configuration



config

CMakeLists.txt

demo.repos

launch

package.xml

robot

records.yaml

simulation.yaml

wgs84_anchor.yaml

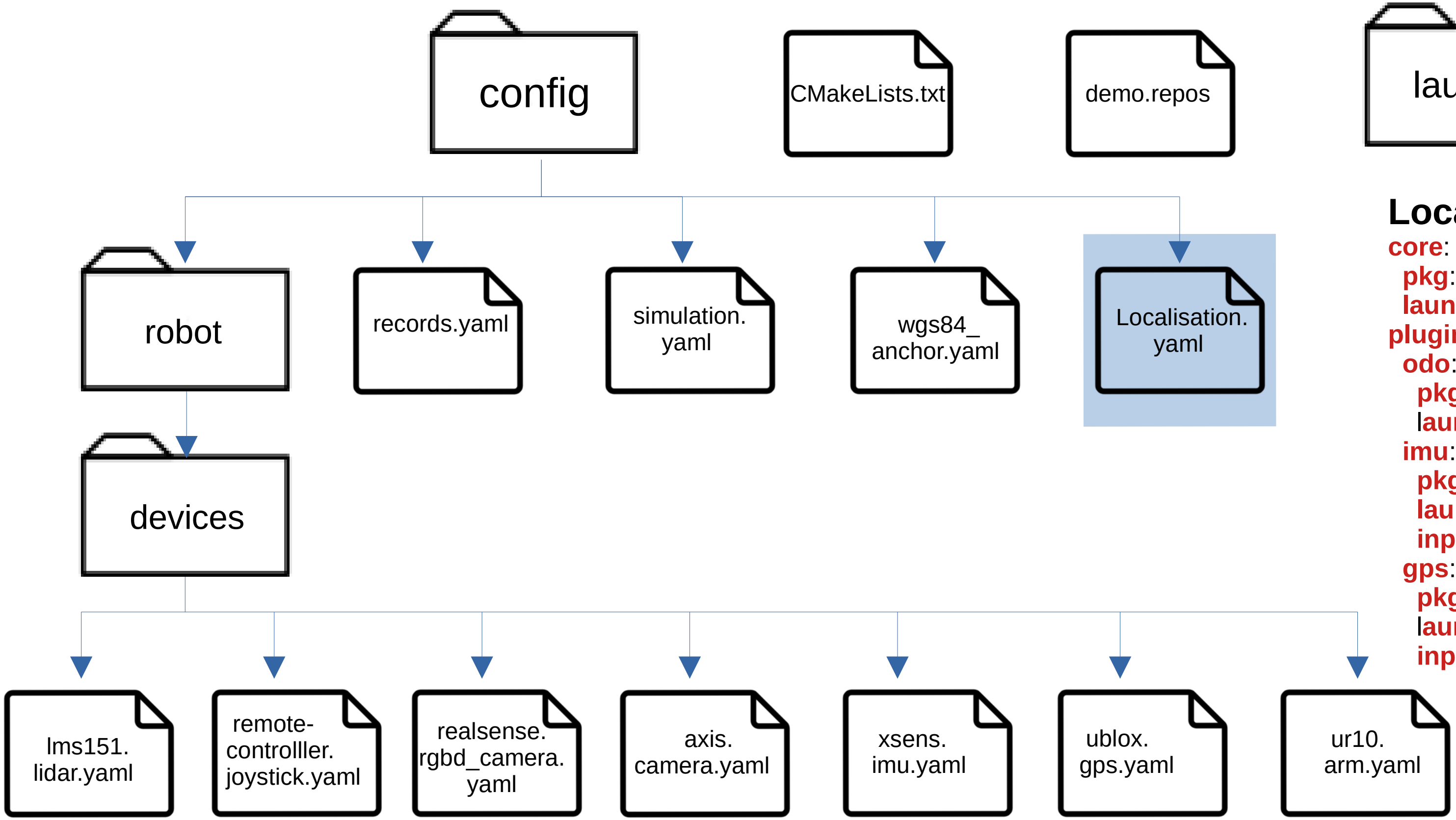
Localisation.yaml

Localisation configuration :

```
core:
  pkg: romea_robot_to_world_localisation_core
  launch: robot_to_world_localisation.launch.py
plugins:
  odo:
    pkg: romea_localisation_odo_plugin
    launch: odo_plugin.launch.py
  imu:
    pkg: romea_localisation_imu_plugin
    launch: imu_plugin.launch.py
    input_sensor_meta_description_filename: xsens
  gps:
    pkg: romea_localisation_gps_plugin
    launch: gps_plugin.launch.py
    input_sensor_meta_description_filename: ublox
```

Demo configuration

Localisation configuration



config

CMakeLists.txt

demo.repos

launch

package.xml

robot

records.yaml

simulation.yaml

wgs84_anchor.yaml

Localisation.yaml

devices

lms151.lidar.yaml

remote-controlller.joystick.yaml

realsense.rgbd_camera.yaml

axis.camera.yaml

xsens.imu.yaml

ublox.gps.yaml

ur10.arm.yaml

Localisation configuration :

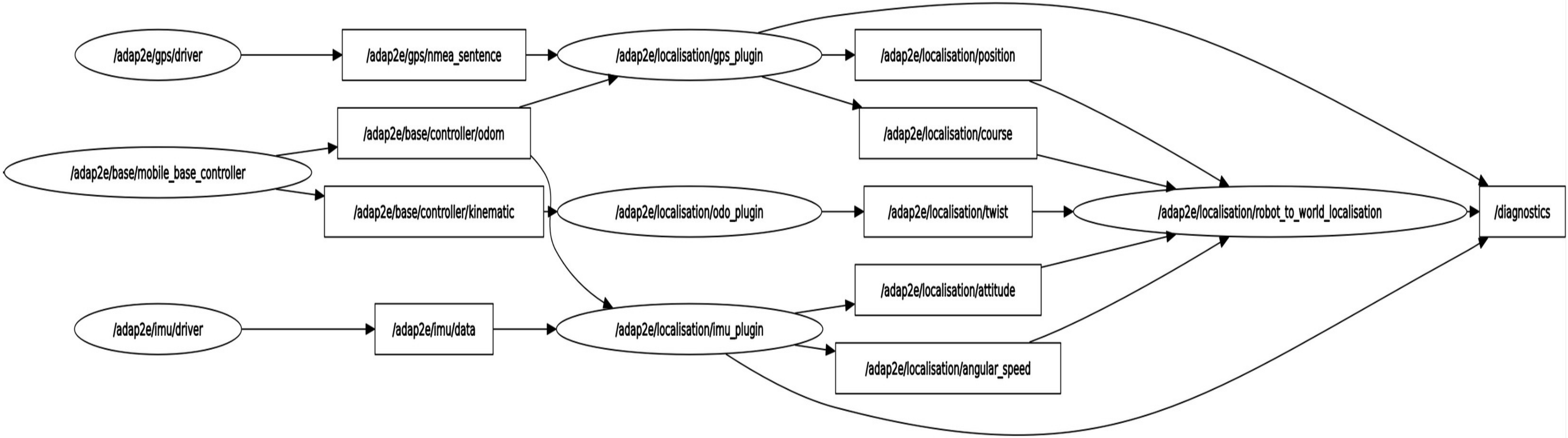
```
core:
  pkg: romea_robot_to_world_localisation_core
  launch: robot_to_world_localisation.launch.py
plugins:
  odo:
    pkg: romea_localisation_odo_plugin
    launch: odo_plugin.launch.py
  imu:
    pkg: romea_localisation_imu_plugin
    launch: imu_plugin.launch.py
    input_sensor_meta_description_filename: xsens
  gps:
    pkg: romea_localisation_gps_plugin
    launch: gps_plugin.launch.py
    input_sensor_meta_description_filename: ublox
```

More Info :

romea_localisation stack

Demo configuration

Localisation configuration


/adap2e/gps/driver
/adap2e/gps/nmea_sentence
/adap2e/localisation/gps_plugin
/adap2e/localisation/position
/adap2e/base/controller/odom
/adap2e/localisation/course
/adap2e/base/mobile_base_controller
/adap2e/base/controller/kinematic
/adap2e/localisation/odo_plugin
/adap2e/localisation/twist
/adap2e/localisation/robot_to_world_localisation
/diagnostics
/adap2e/localisation/attitude
/adap2e/imu/driver
/adap2e/imu/data
/adap2e/localisation/imu_plugin
/adap2e/localisation/angular_speed