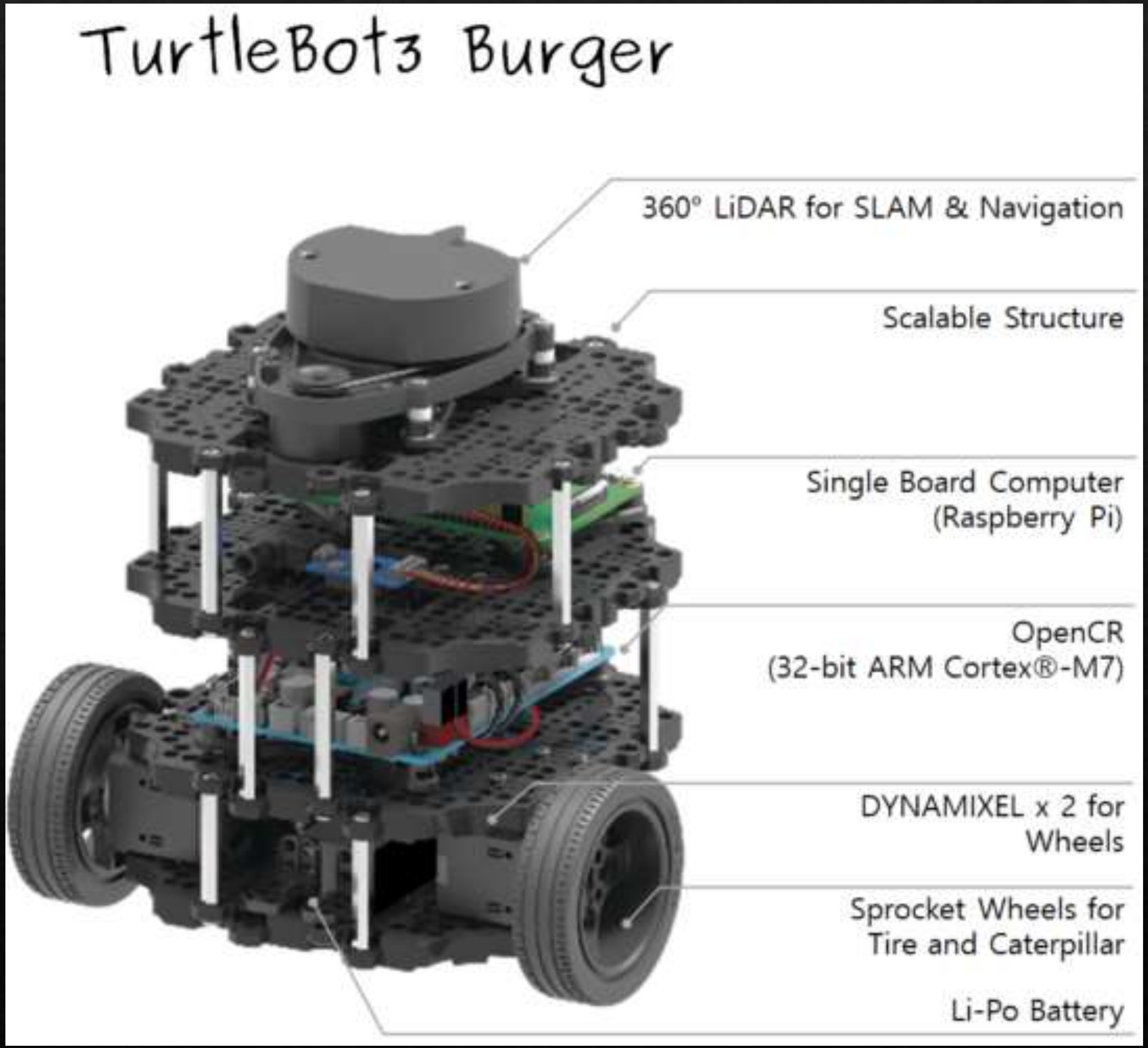
Robotic Arm

Project Proposal : Team A

ISE 학과	Ganiev Mukhammad Abdullokh	12190246
ISE 학과	Kunishev Ozodjon	12190259
ISE 학과	Ibragimov Jasurbek Yunus Ugli	12190263
ISE 학과	Isaeva Gulzhan	12194812
ISE 학과	Khalilov Asadbek Ismoiljon Ugli	12194882
ISE 학과	Yuldashev Mukhammadyusuf Kobuljon Ugli	12194916

Components


TurtleBot3 Burger
360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer
(Raspberry Pi)
OpenCR
(32-bit ARM Cortex®-M7)
DYNAMIXEL x 2 for
Wheels
Sprocket Wheels for
Tire and Caterpillar
Li-Po Battery

2. 1. 1. Hardware Specifications

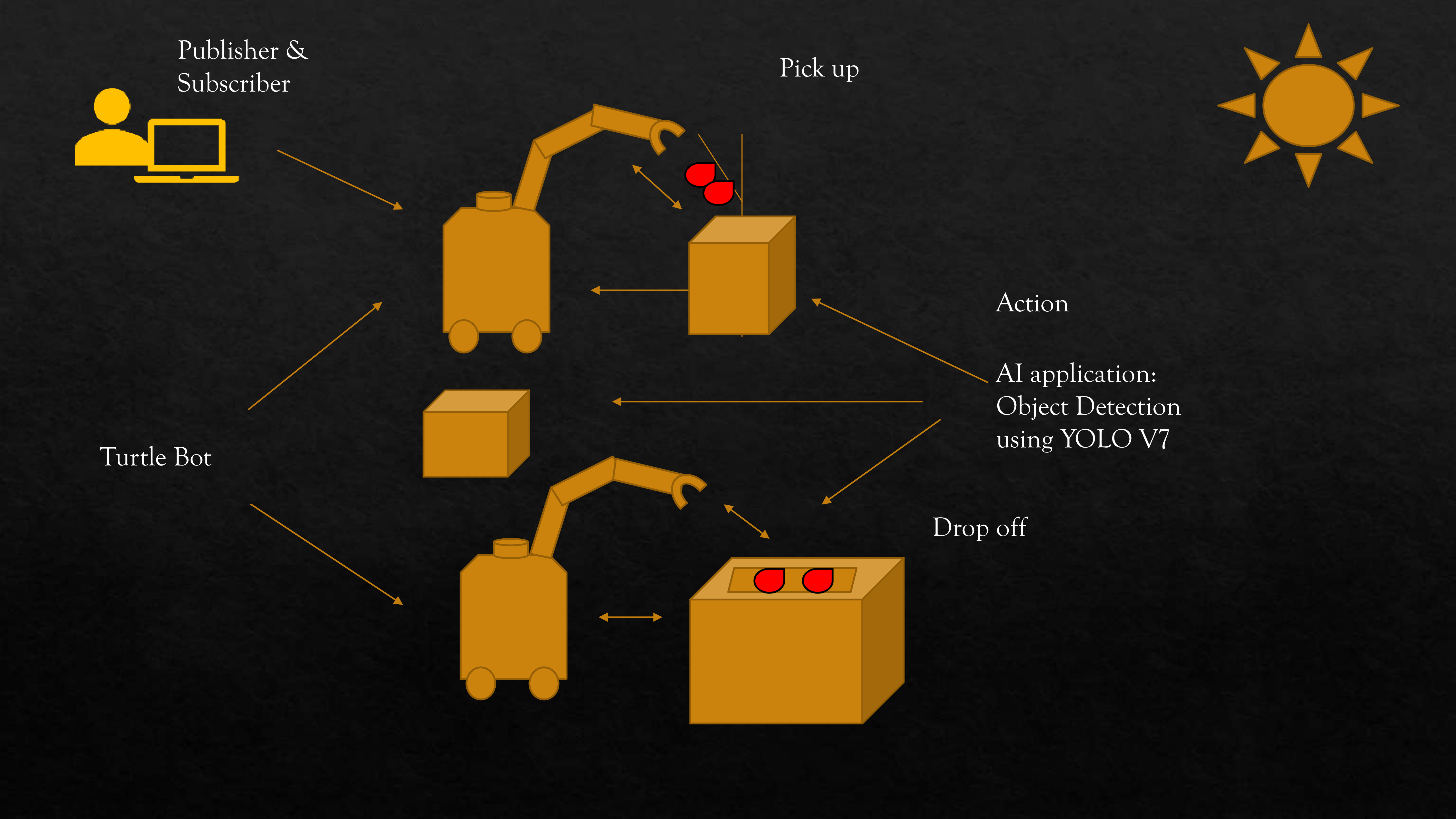
Items	Burger
Maximum translational velocity	0.22 m/s
Maximum rotational velocity	2.84 rad/s (162.72 deg/s)
Maximum payload	15kg
Size (L x W x H)	138mm x 178mm x 192mm
Weight (+ SBC + Battery + Sensors)	1kg
Threshold of climbing	10 mm or lower
Expected operating time	2h 30m
Expected charging time	2h 30m
SBC (Single Board Computers)	Raspberry Pi
MCU	32-bit ARM Cortex®-M7 with FPU (216 MHz, 462 DMIPS)
Remote Controller	-
Actuator	XL430-W250
LDS(Laser Distance Sensor)	360 Laser Distance Sensor LDS-01 or LDS-02
Camera	-
IMU	Gyroscope 3 Axis Accelerometer 3 Axis
Power connectors	3.3V / 800mA 5V / 4A 12V / 1A
Expansion pins	GPIO 18 pins Arduino 32 pin
Peripheral	UART x3, CAN x1, SPI x1, I2C x1, ADC x5, 5pin OLLO x4
DYNAMIXEL ports	RS485 x 3, TTL x 3
Audio	Several programmable beep sequences
Programmable LEDs	User LED x 4
Status LEDs	Board status LED x 1 Arduino LED x 1 Power LED x 1
Buttons and Switches	Push buttons x 2, Reset button x 1, Dip switch x 2
Battery	Lithium polymer 11.1V 1800mAh / 19.98Wh 5C
PC connection	USB
Firmware upgrade	via USB / via JTAG
Power adapter (SMPS)	Input : 100-240V, AC 50/60Hz, 1.5A @max Output : 12V DC, 5A

CPU	Up to AMD Ryzen™ 9 5000 H-Processors
OS	Windows 10 Home (MSI recommends Windows 11 Pro for business.) FREE Upgrade to Windows 11*
DISPLAY	15.6" FHD (1920x1080), 144Hz, IPS-Level 15.6" FHD (1920x1080), 240Hz, IPS-Level
GRAPHICS	AMD Radeon™ RX 6700M with 10GB GDDR6
MEMORY	2 Slots DDR4-3200 Max 64GB
STORAGE CAPABILITY	2x M.2 SSD slot (NVMe PCIe Gen3)
WEBCAM	HD type (30fps@720p)
KEYBOARD	RGB Backlight Keyboard

WEBCAM	HD type (30fps@720p)
KEYBOARD	RGB Backlight Keyboard

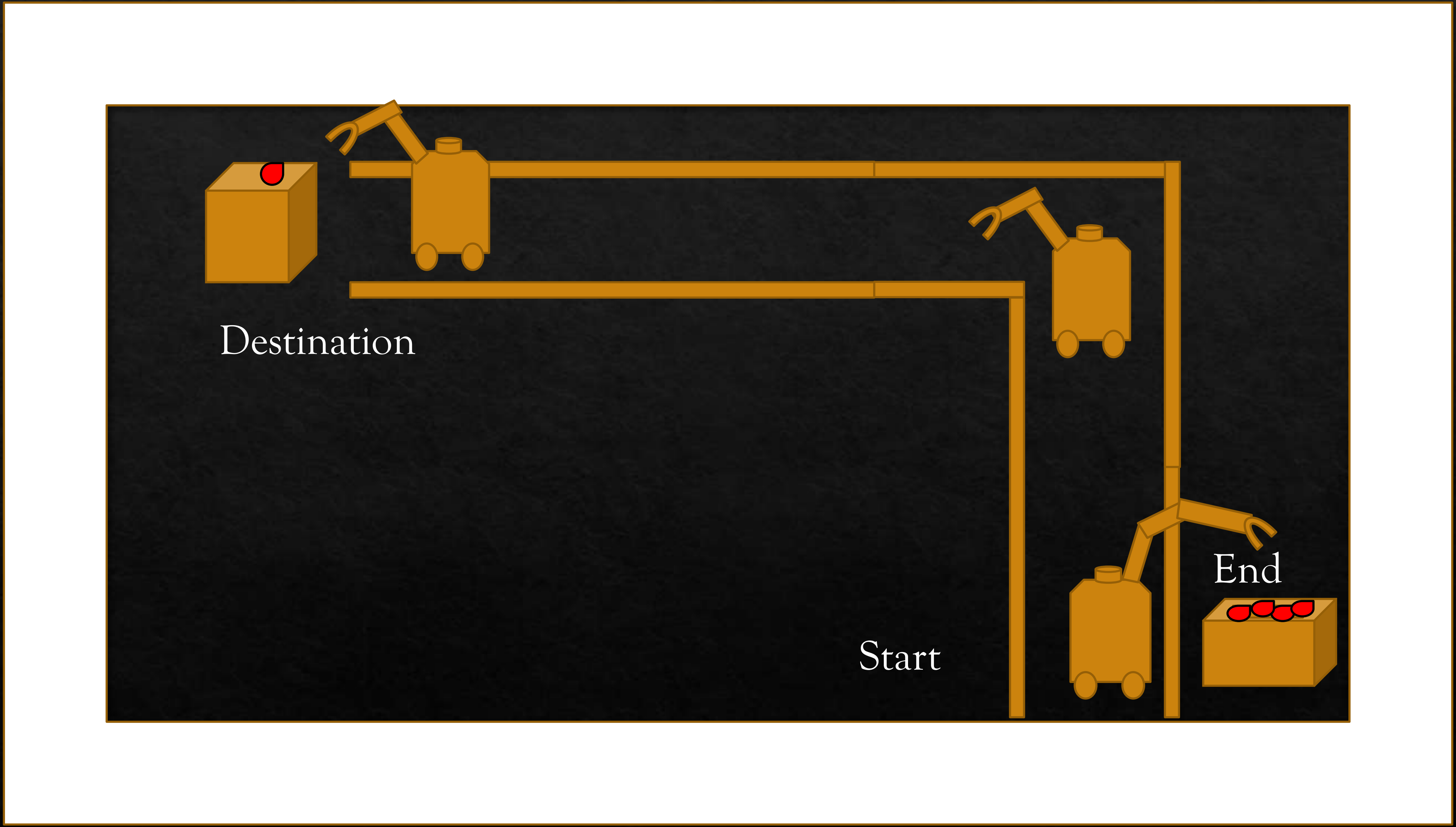
COMMUNICATION	802.11 ax Wi-Fi 6E + Bluetooth v5.2
AUDIO	2x 2W Speaker
WEIGHT (W/ BATTERY)	1.9 kg
AUDIO JACK	1x Mic-in/Headphone-out Combo Jack
I/O PORTS	1x Type-C (USB3.2 Gen2 / DP) 1x Type-C USB3.2 Gen2 2x Type-A USB3.2 Gen2 1x (4K @ 60Hz) HDMI™
BATTERY	82 Battery (Whr) 4-Cell
AC ADAPTER	240W adapter
DIMENSION (WXDXH)	357 x 247 x 19 mm

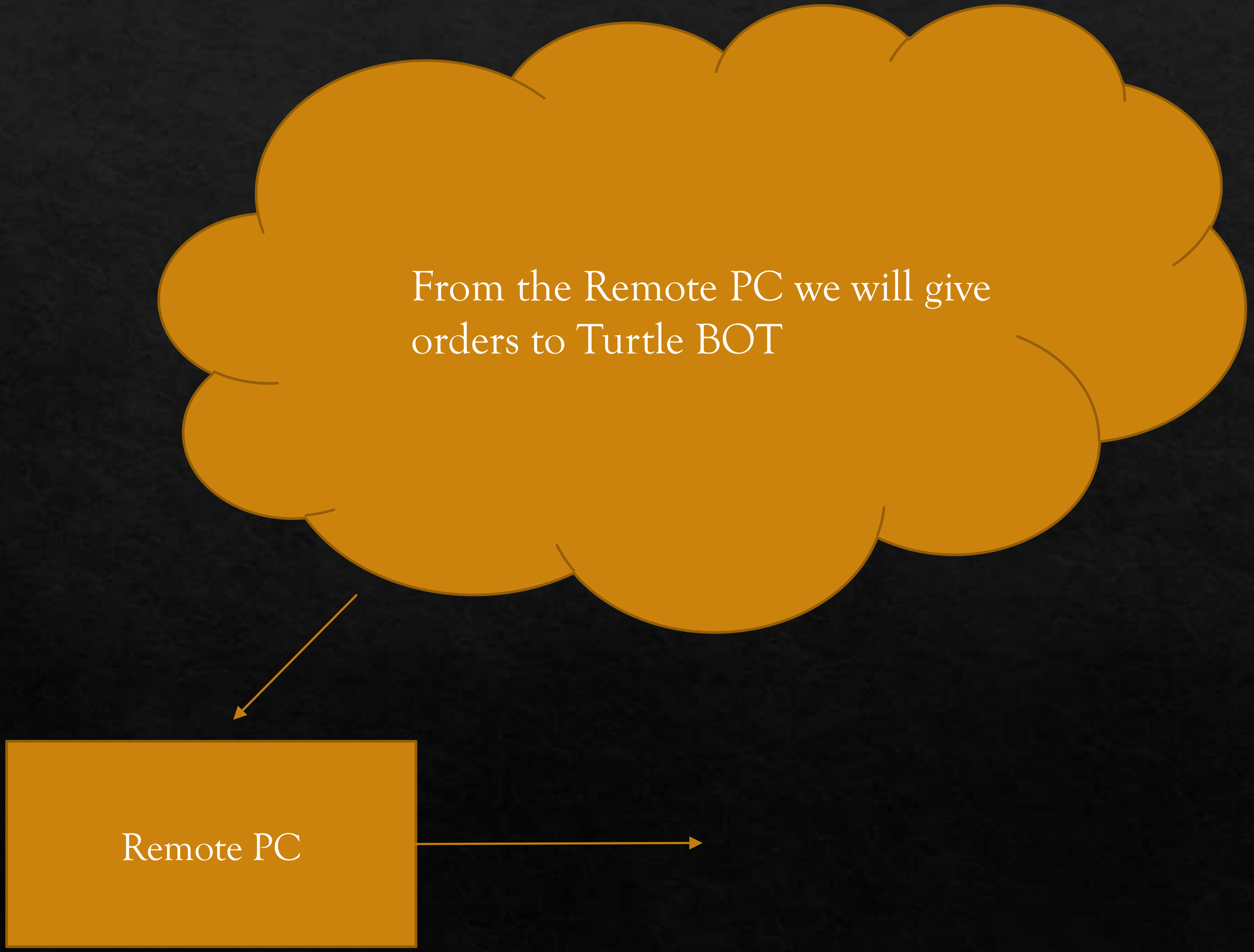
Real-World Application:
Greenhouse Tomato picker smart robot

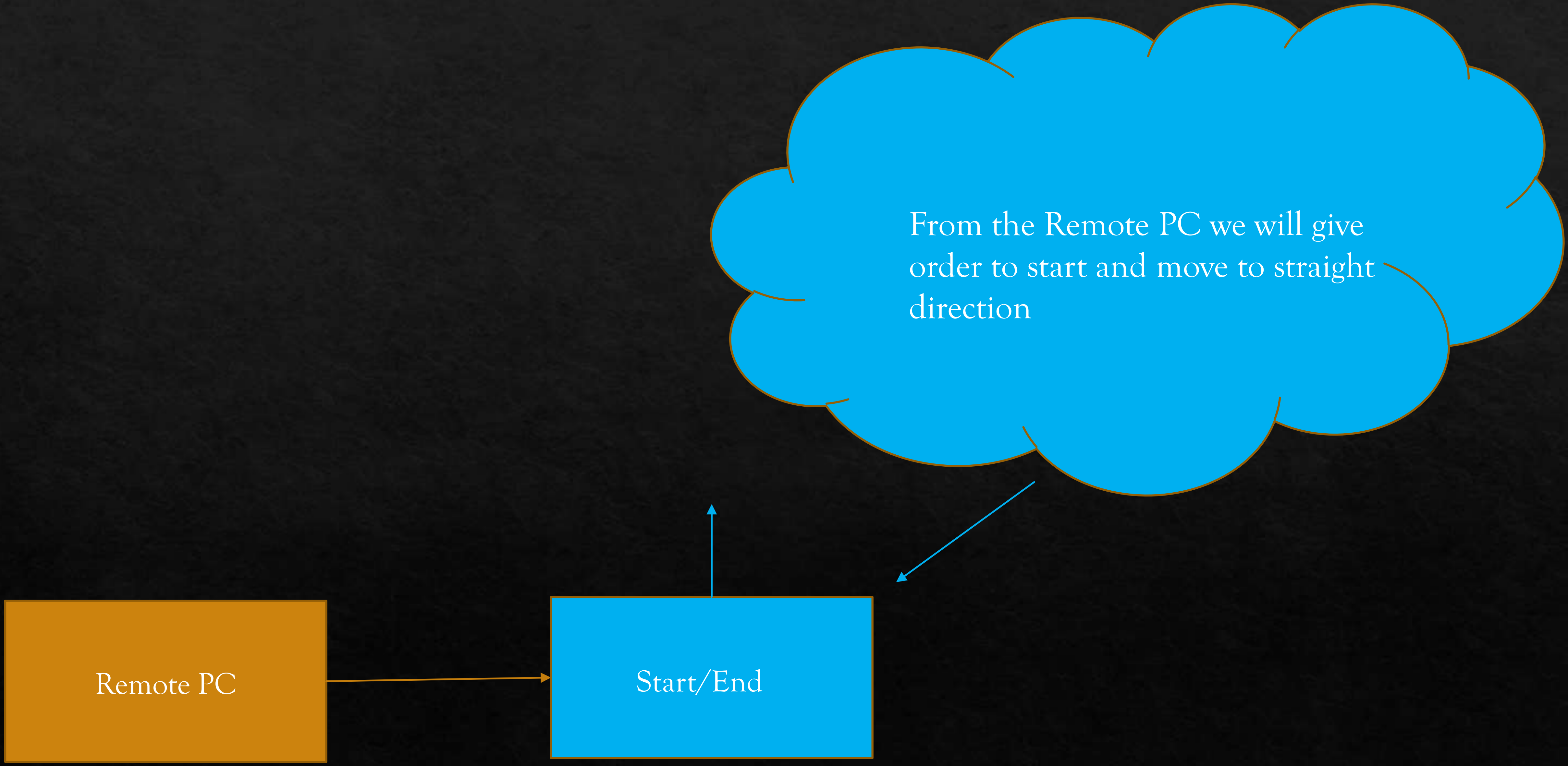


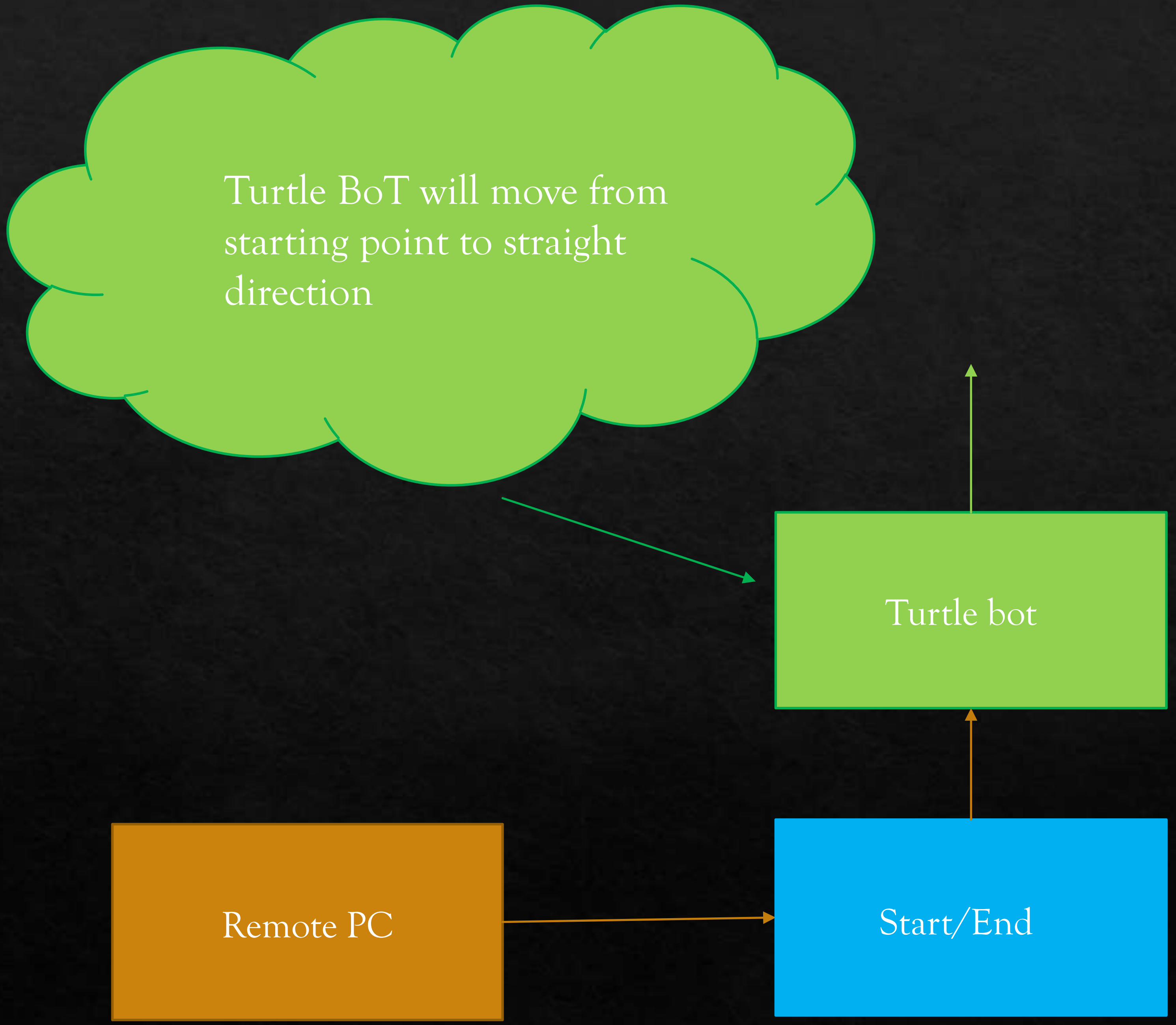
Publisher &
Subscriber
Pick up
Action
AI application:
Object Detection
using YOLO V7
Turtle Bot
Drop off

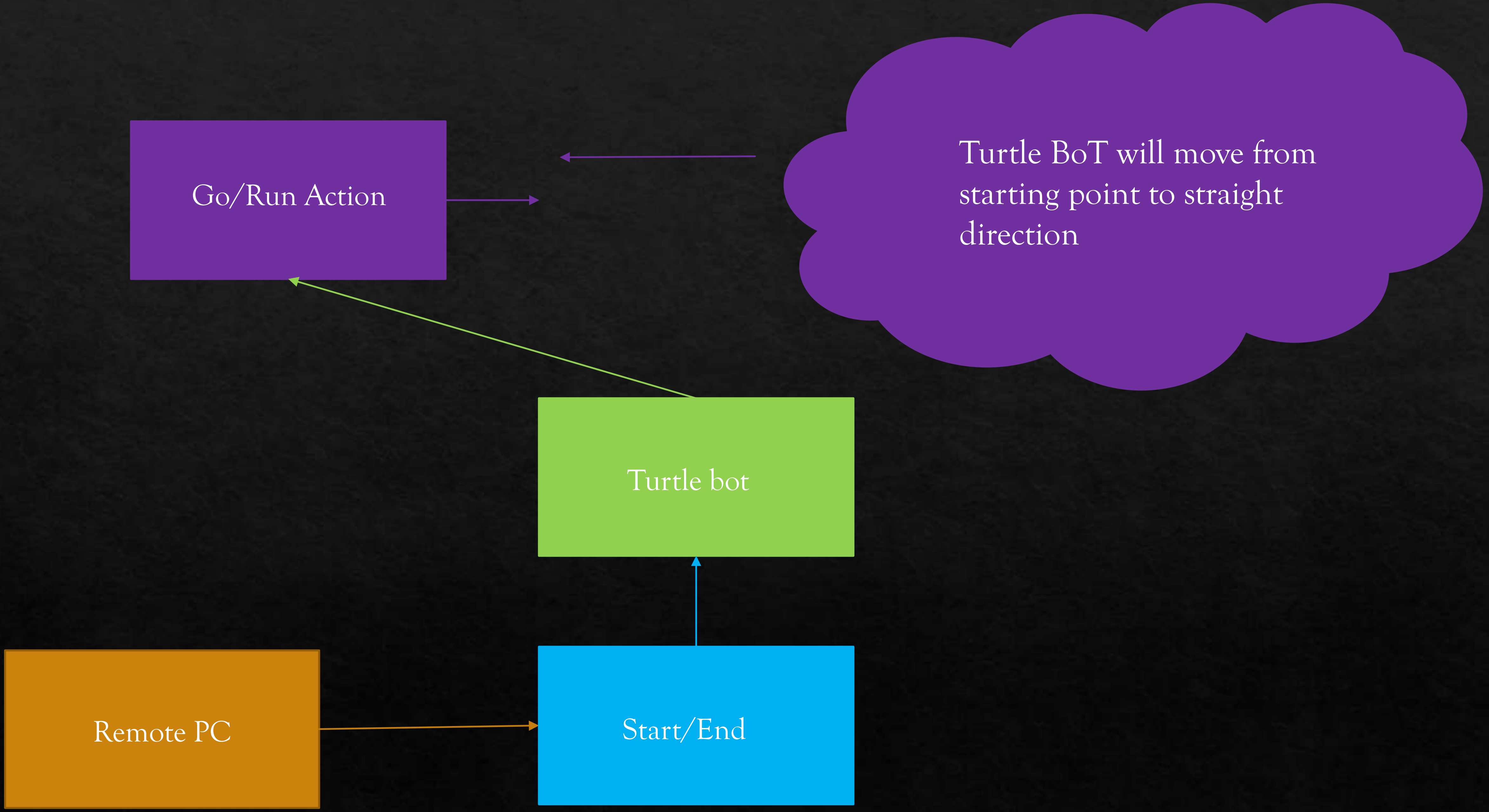
Map

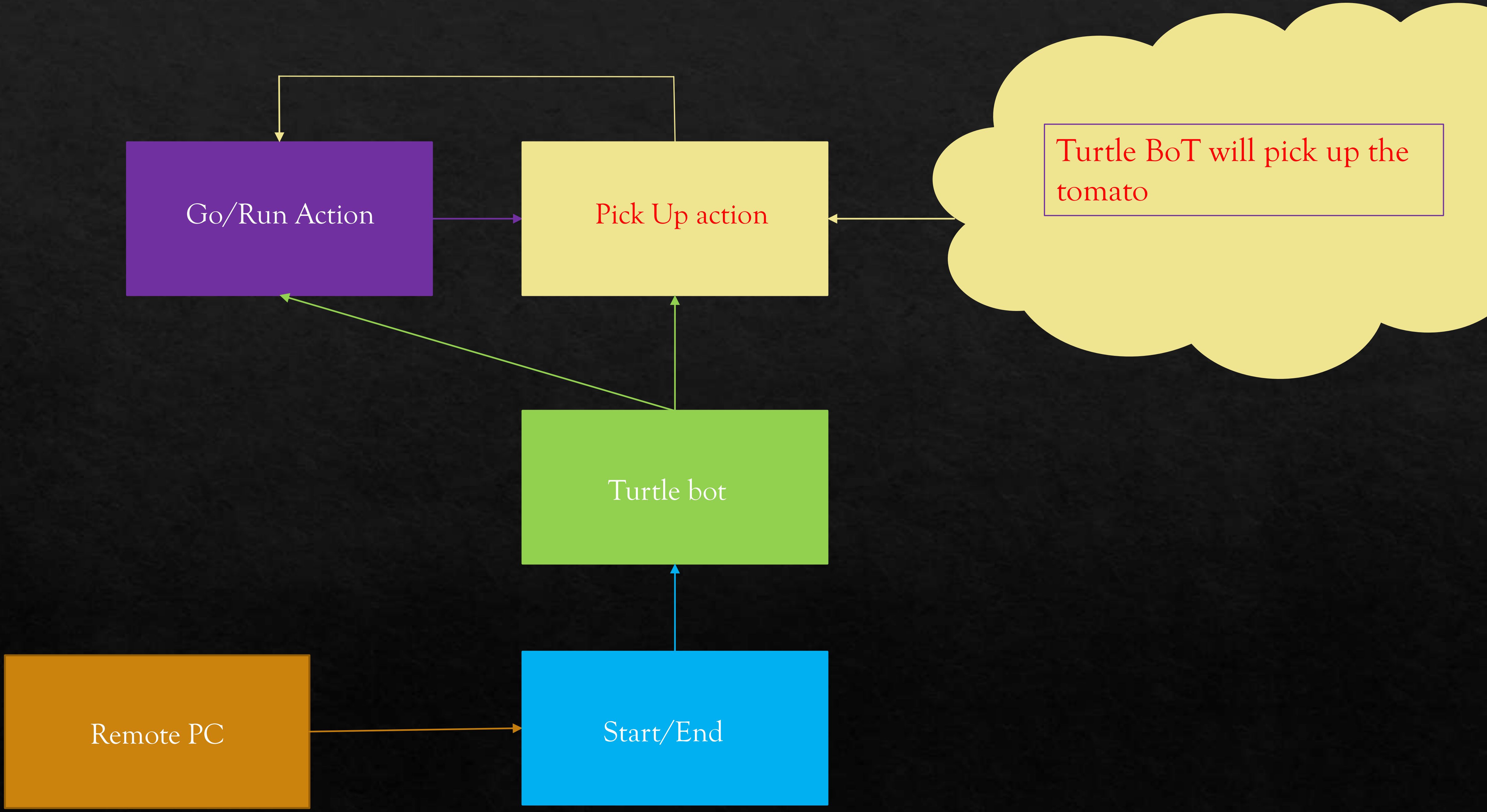



From the Remote PC we will give orders to Turtle BOT
Remote PC


From the Remote PC we will give order to start and move to straight direction
Remote PC
Start/End

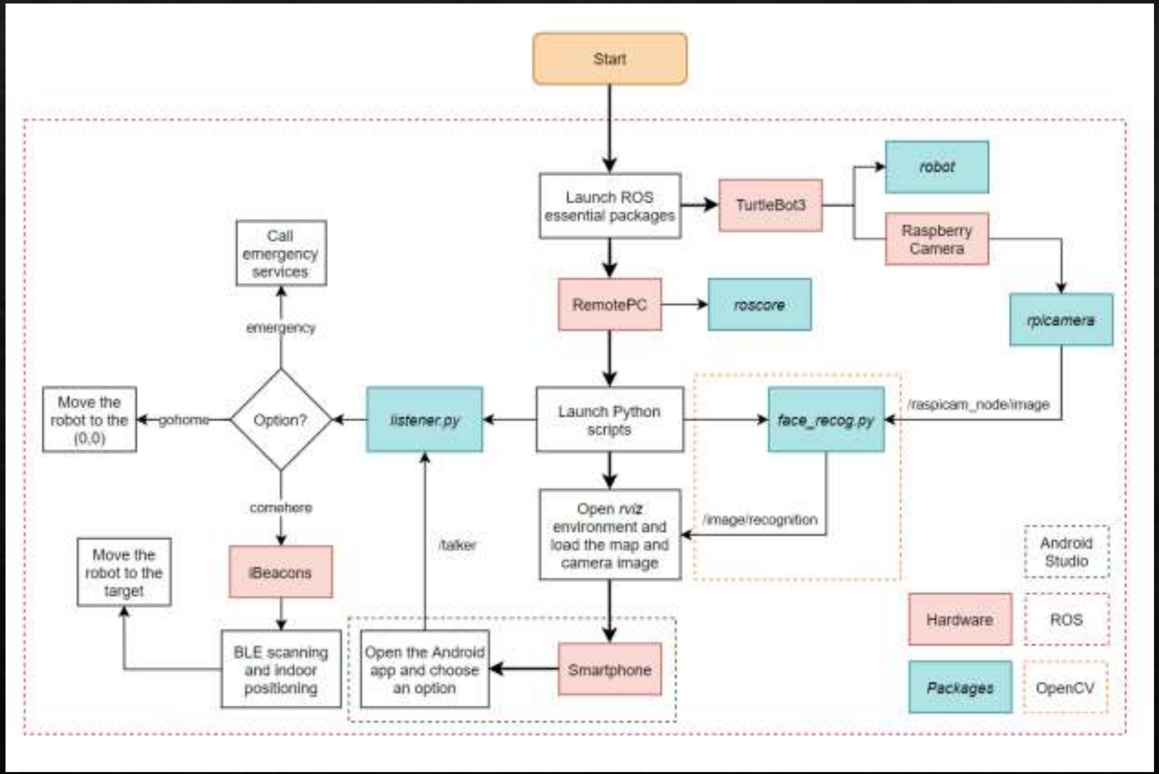

Turtle BoT will move from starting point to straight direction
Turtle bot
Remote PC
Start/End


Go/Run Action
Turtle BoT will move from starting point to straight direction
Turtle bot
Remote PC
Start/End

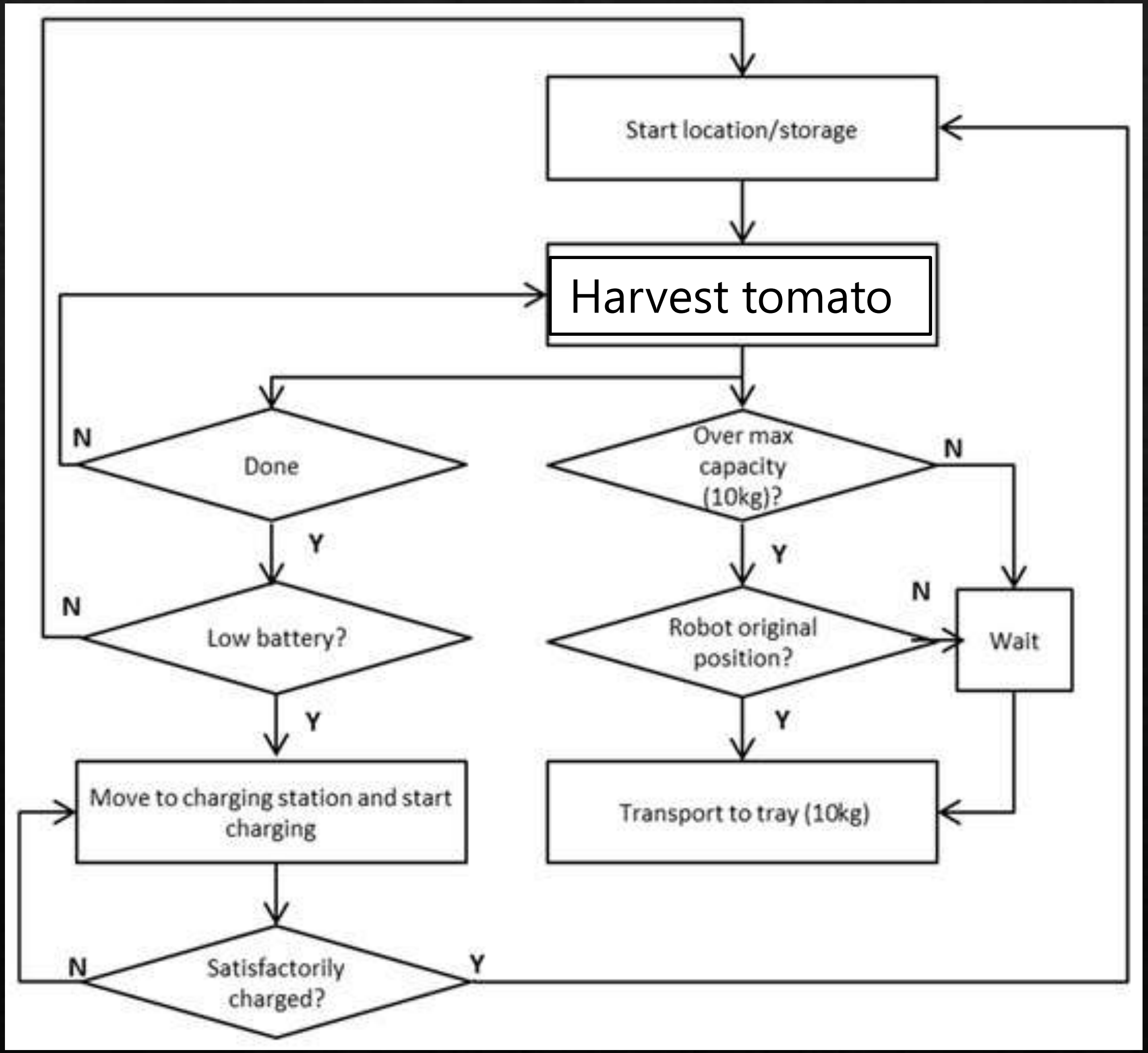

Go/Run Action
Pick Up action
Turtle BoT will pick up the tomato
Turtle bot
Remote PC
Start/End


Go/Run Action
Pick Up action
Drop Off Action
Turtle bot
Turtle bot will drop off the tomato and maintain the default potion
Remote PC
Start/End

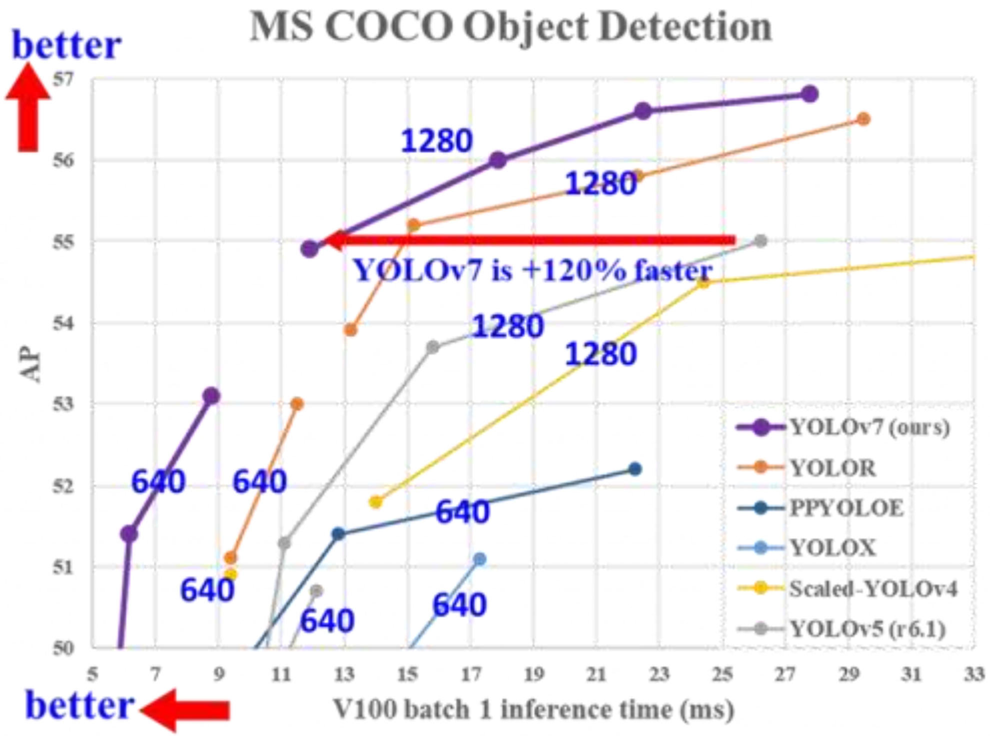
Software Diagram



Flowchart Diagram

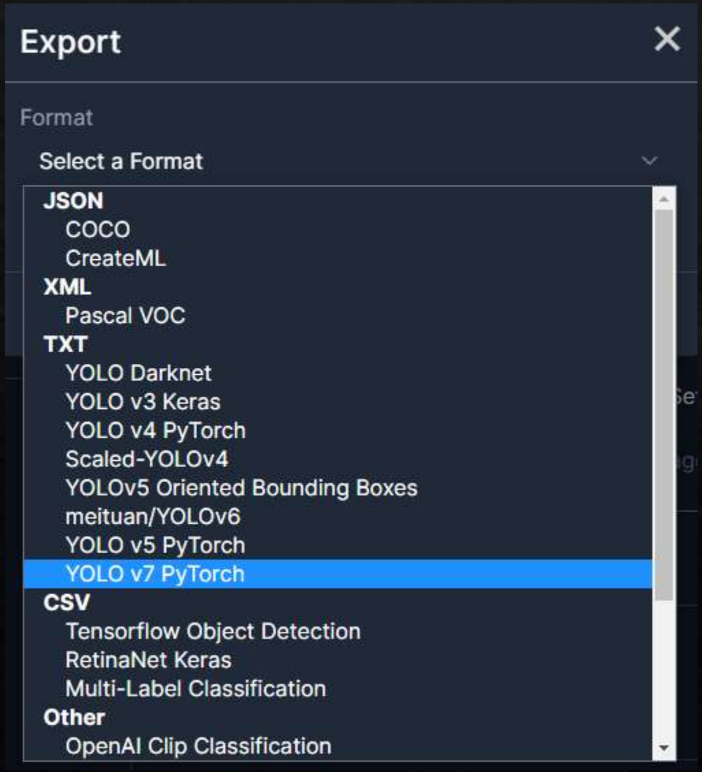


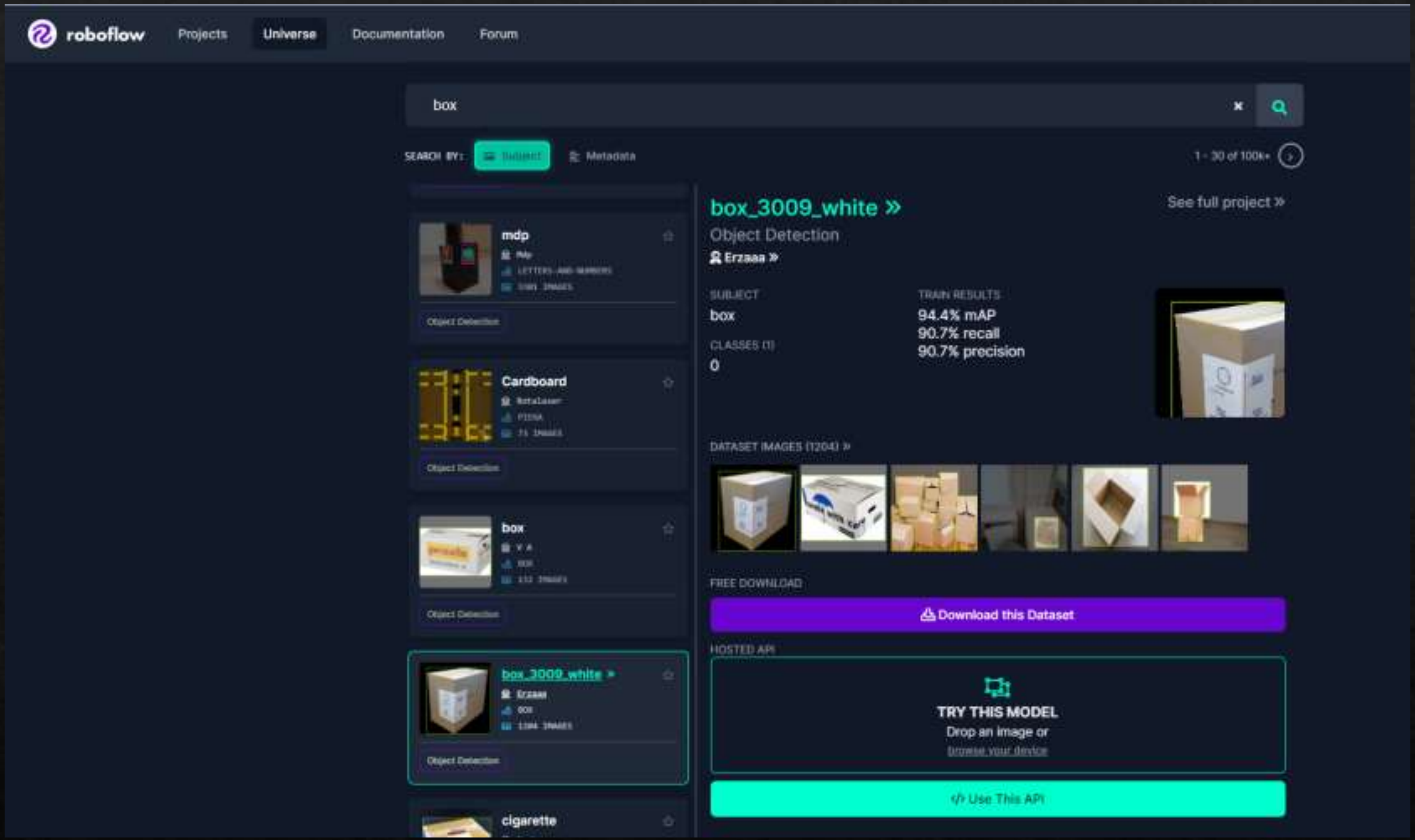
We will be doing object detection using…



For our project we need to detect box as our demo is with box.

- -Ready images
- -Already annotated
- -already YOLOv7 format
- -Can be directly trained.
- -We do custom dataset training for YOLOv7





-We use Google collab's free GPU to train.

-We download YOLOv7 files from its git hub.

-We use google drive as a storage.

Files

- ..
- sample_data
- yolov7
  - cfg
  - data
  - deploy
  - figure
  - inference
  - models
  - paper
  - scripts
  - tools
  - utils
  - LICENSE.md
  - README.md
  - detect.py
  - export.py
  - hubconf.py
  - requirements.txt
  - test.py
  - train.py
  - train_aux.py

```
from google.colab import drive
drive.mount('/content/drive')
```

```
# Download YOLOv7 repository and install requirements
!git clone https://github.com/WongKinYiu/yolov7
%cd yolov7
!pip install -r requirements.txt
```

```
# REPLACE with your custom code snippet generated above

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="YOUR API KEY")
project = rf.workspace("YOUR-WORKSPACE").project("YOUR-PROJECT")
dataset = project.version(1).download("yolov7")
```

```
[ ] # run this cell to begin training
%cd /content/yolov7
!python train.py --batch 16 --epochs 55 --data {dataset.location}/data.yaml --weights 'yolov7_training.pt' --device 0
```

-We use the best epoch weight for detection.

```
opt  = {

    "weights": "/content/gdrive/MyDrive/yolov7/runs/train/exp/weights/best.pt", # Path to weights file default weights are for nano model
    "yaml"   : "Box/data.yaml",
    "img-size": 640, # default image size
    "conf-thres": 0.25, # confidence threshold for inference.
    "iou-thres" : 0.45, # NMS IoU threshold for inference.
    "device" : '0',  # device to run our model i.e. 0 or 0,1,2,3 or cpu
    "classes" : classes_to_filter  # list of classes to filter or None

}
```

-We have some py files that can link our model to web camera..

Webcam Helper Functions

The camera detects it and send signal to turtle bar

\+ Код + Текст

4. Evaluation

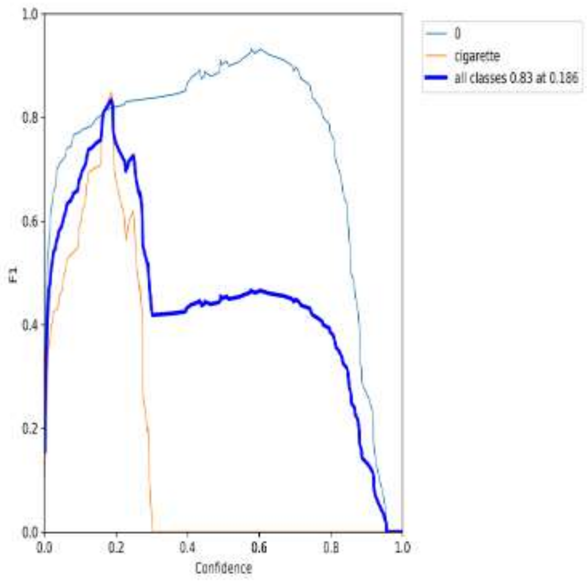
- Note the checkpoints from training will be stored by default in runs/train/exp. Take the path of the latest checkpoint

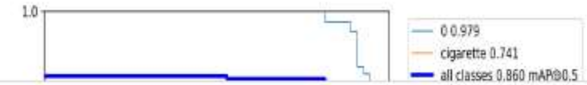
We can evaluate the performance of our custom training using the provided evalution script.

Note we can adjust the below custom arguments. For details, see the arguments accepted by detect.py.

4.1 F1 and Precision Recall Curve

```
from IPython.display import Image
display(Image("/content/gdrive/MyDrive/yolov7/runs/train/exp/F1_curve.png", width=400, height=400))
display(Image("/content/gdrive/MyDrive/yolov7/runs/train/exp/PR_curve.png", width=400, height=400))
display(Image("/content/gdrive/MyDrive/yolov7/runs/train/exp/confusion_matrix.png", width=500, height=500))
```





After training

5.1.2 Display Inference on Folder of Test Images

Note From the above output display copy the full path of folder where test images are stored

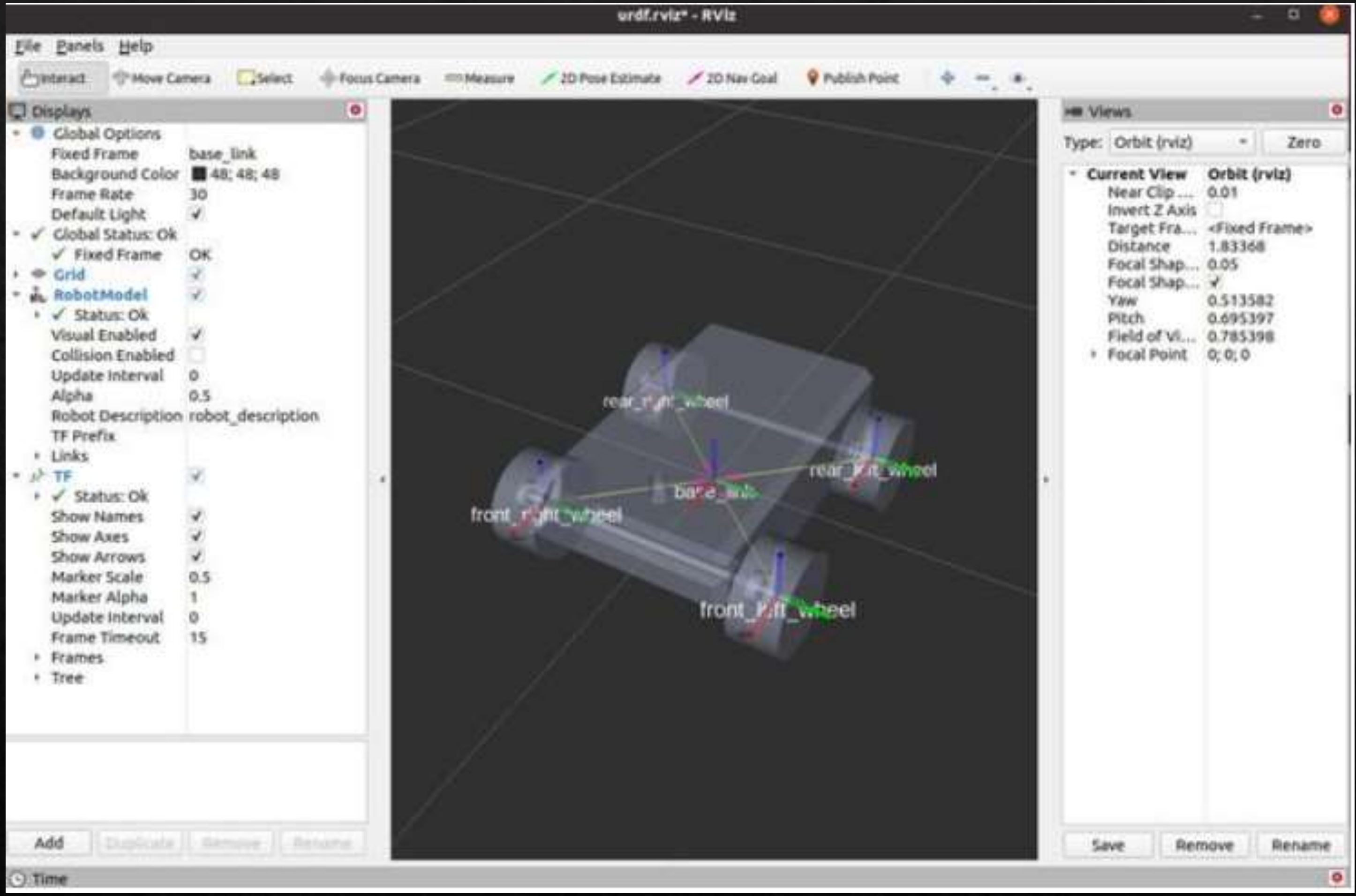
```
#display inference on ALL test images

import glob
from IPython.display import Image, display

i = 0
limit = 10000 # max images to print
for imageName in glob.glob('/content/gdrive/MyDrive/yolov7/runs/detect/exp/depositphotos_25710999-stock-photo-cardboard-boxes.jpg'):
    #Assuming JPG
    if i < limit:
      display(Image(filename=imageName))
      print("\n")
    i = i + 1

display(Image("/content/gdrive/MyDrive/yolov7/runs/detect/exp/depositphotos_25710999-stock-photo-cardboard-boxes.jpg", width=400, height=400))
```

Simulated smart arm installation process:

1Step We installed Ubuntu and some other packages
2Step Then we installed Noetic and world ROS packages.

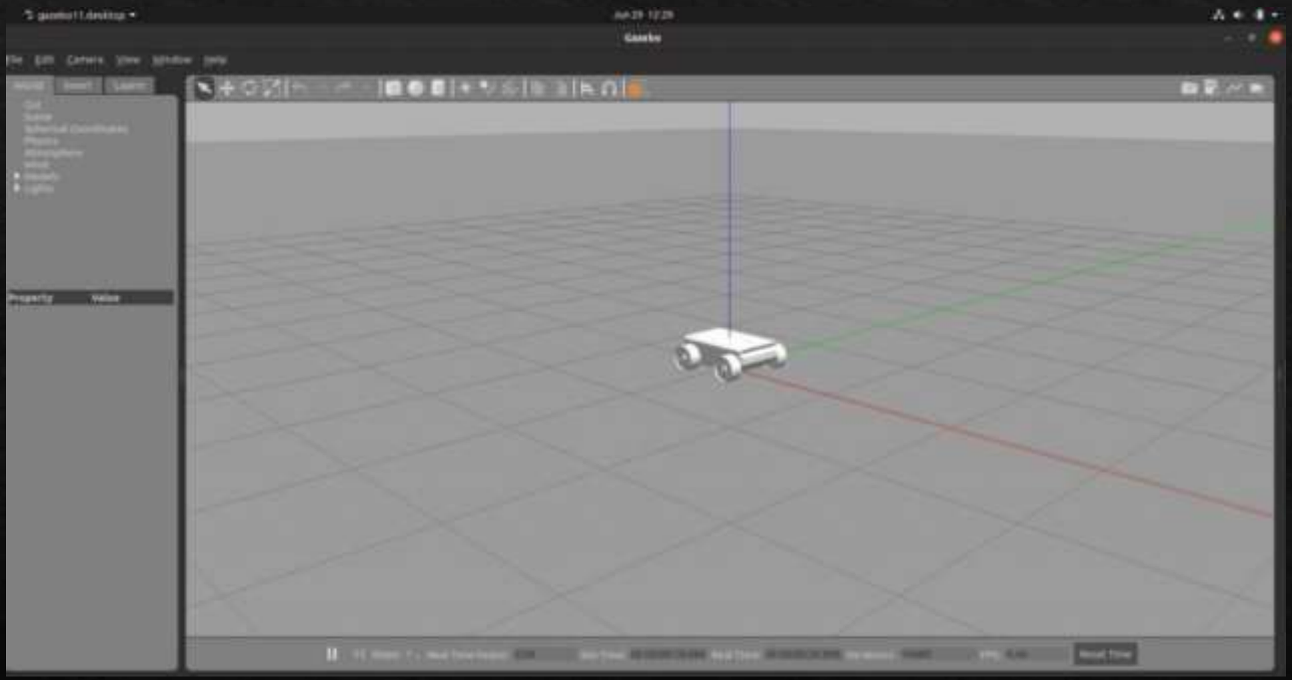
Building a Simulated Mobile Robot Base Using ROS

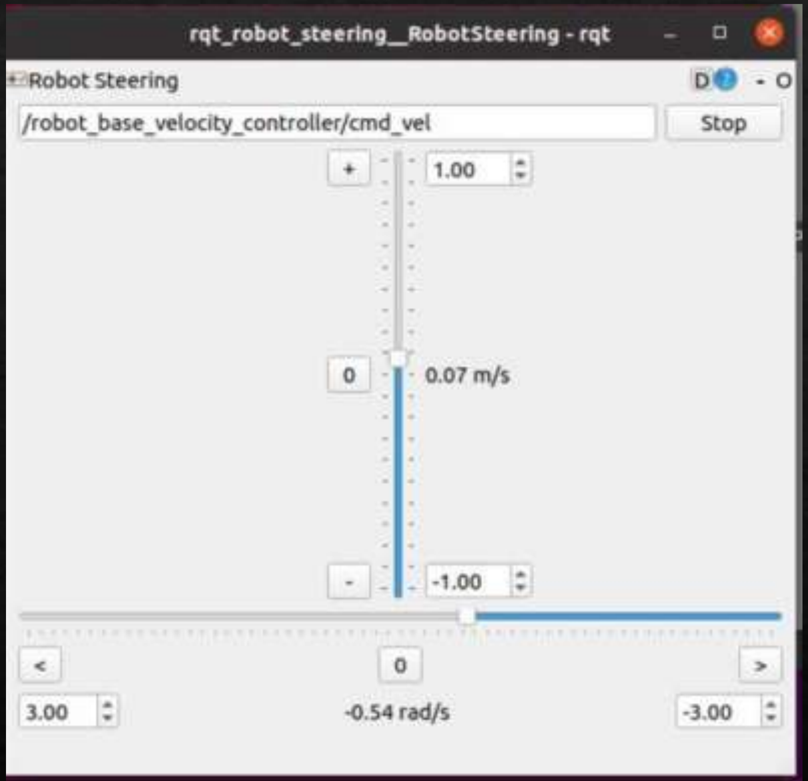
- Install ROS Packages
- Create a ROS Package
- Create Folders
- Build the Base of the Robot
- Launch the Base of the Robot


urdf.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 0.5
Robot Description robot_description
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 0.5
Marker Alpha 1
Update Interval 0
Frame Timeout 15
Frames
Tree
Add
Duplicate
Remove
Rename
Time
Views
Type: Orbit (rviz)
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 1.83368
Focal Shap... 0.05
Focal Shap...
Yaw 0.513582
Pitch 0.695397
Field of Vi... 0.785398
Focal Point 0; 0; 0
Save
Remove
Rename


Node: /joint_state_publis...
front_left_wheel_joint 0.00
front_right_wheel_joint 0.00
rear_left_wheel_joint 0.00
rear_right_wheel_joint 0.00
Randomize

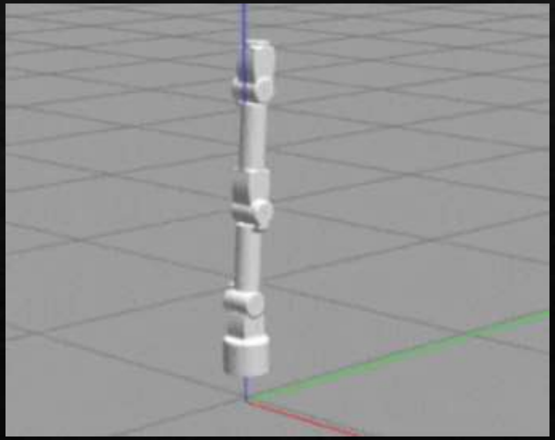
```
rosrun rqt_robot_steering rqt_robot_steering
```

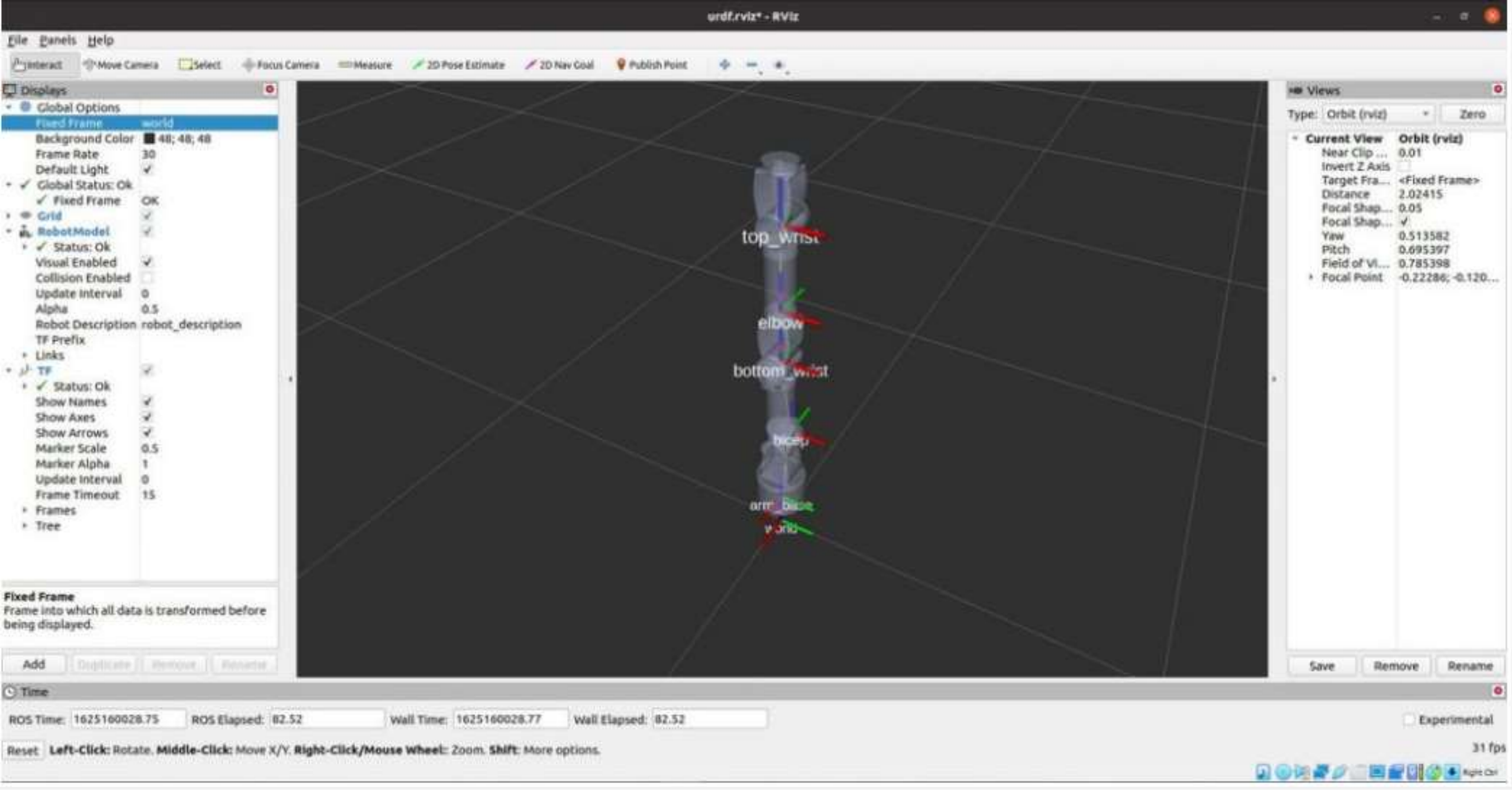


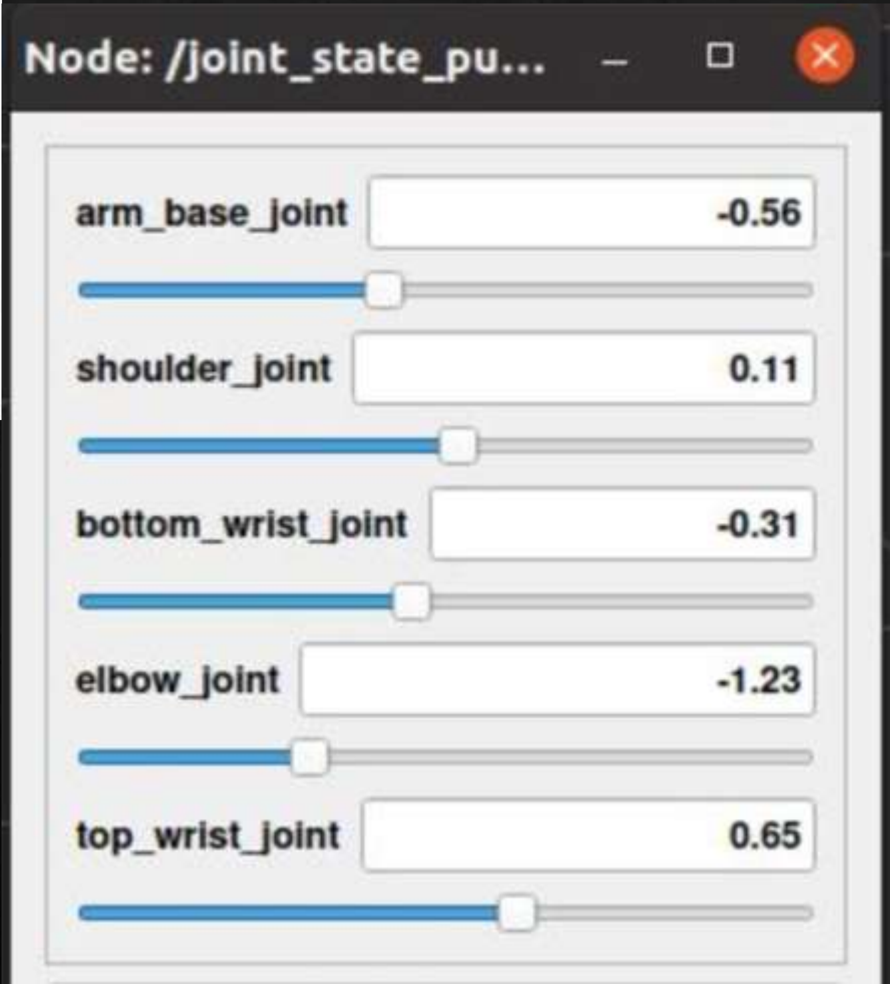


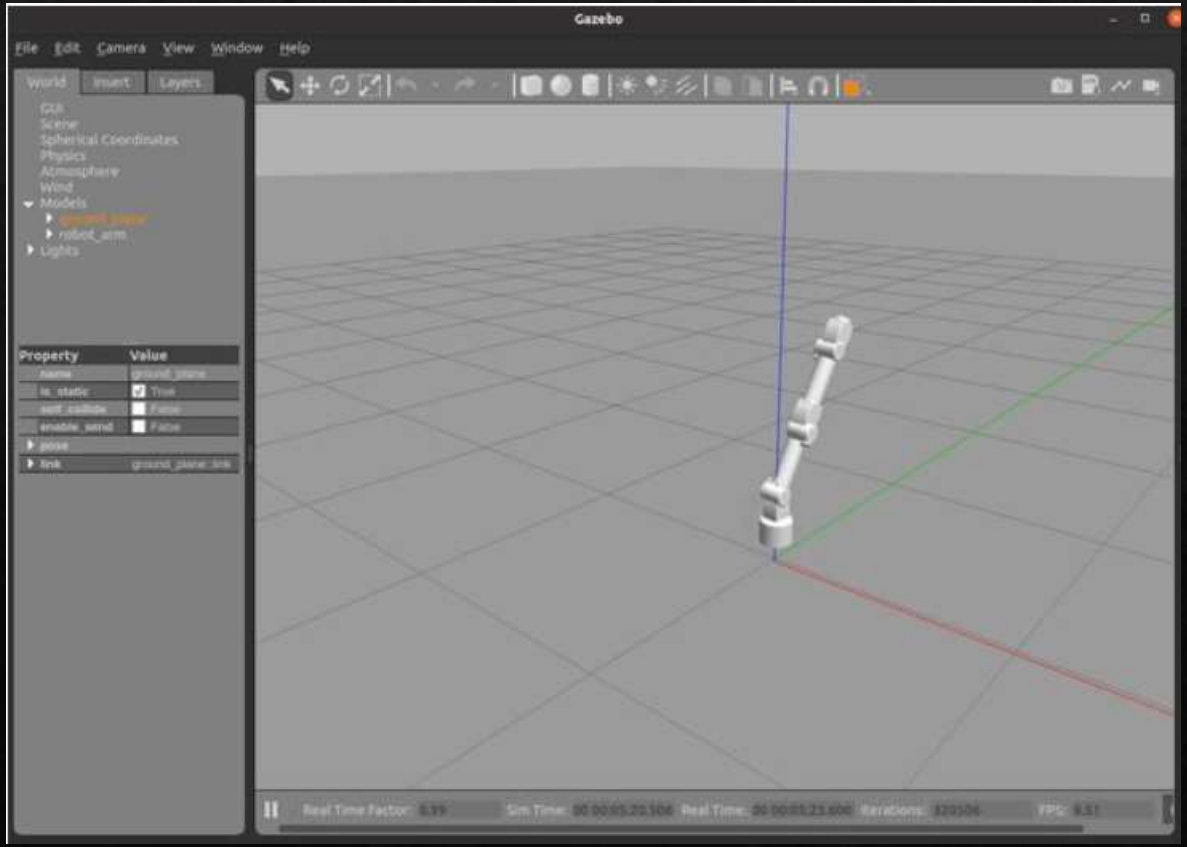
Building a Simulated Robot Arm Using ROS

- Build the Robot Arm
- Test the Robot Arm
- Launch the Robot Arm



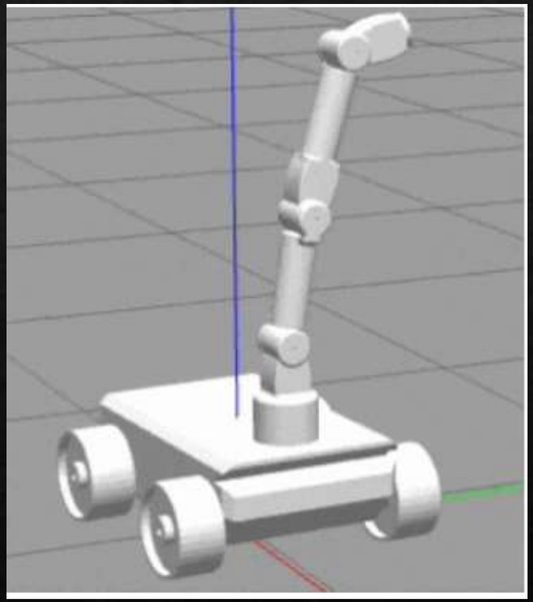

urdf.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Global Options
Fixed Frame world
Background Color 48; 48; 48
Frame Rate 30
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 0.5
Robot Description robot_description
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 0.5
Marker Alpha 1
Update Interval 0
Frame Timeout 15
Frames
Tree
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
top_wrist
elbow
bicep
Views
Type: Orbit (rviz) Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 2.02415
Focal Shap... 0.05
Focal Shap...
Yaw 0.513582
Pitch 0.695397
Field of Vi... 0.785398
Focal Point -0.22286; -0.120...
Save Remove Rename
Time
ROS Time: 1625160028.75 ROS Elapsed: 82.52 Wall Time: 1625160028.77 Wall Elapsed: 82.52
Experimental
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps


Node: /joint_state_pu...
arm_base_joint -0.56
shoulder_joint 0.11
bottom_wrist_joint -0.31
elbow_joint -1.23
top_wrist_joint 0.65

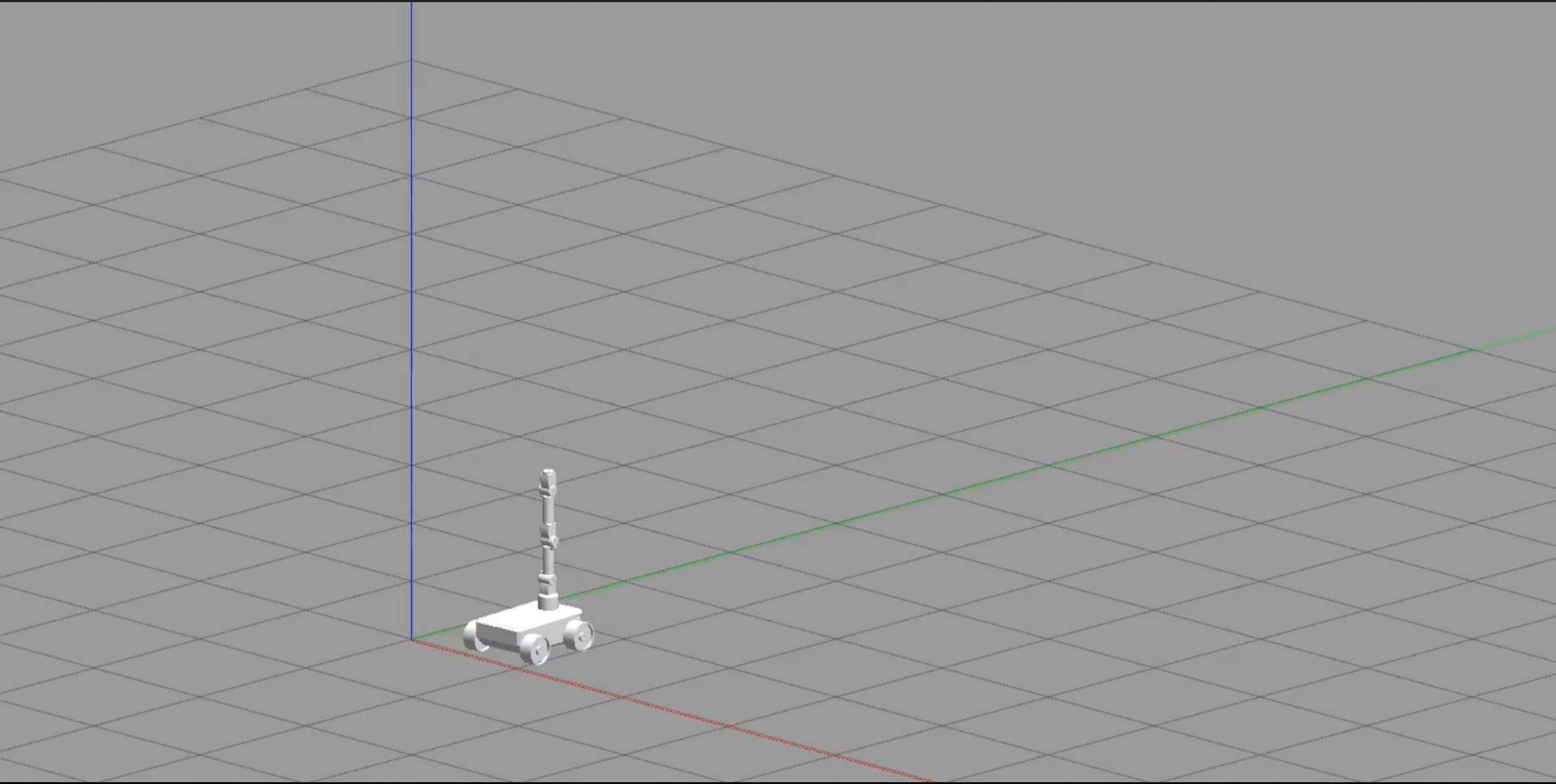

Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
robot_arm
Lights
Property Value
is_static True
enable_wind False
pose
link

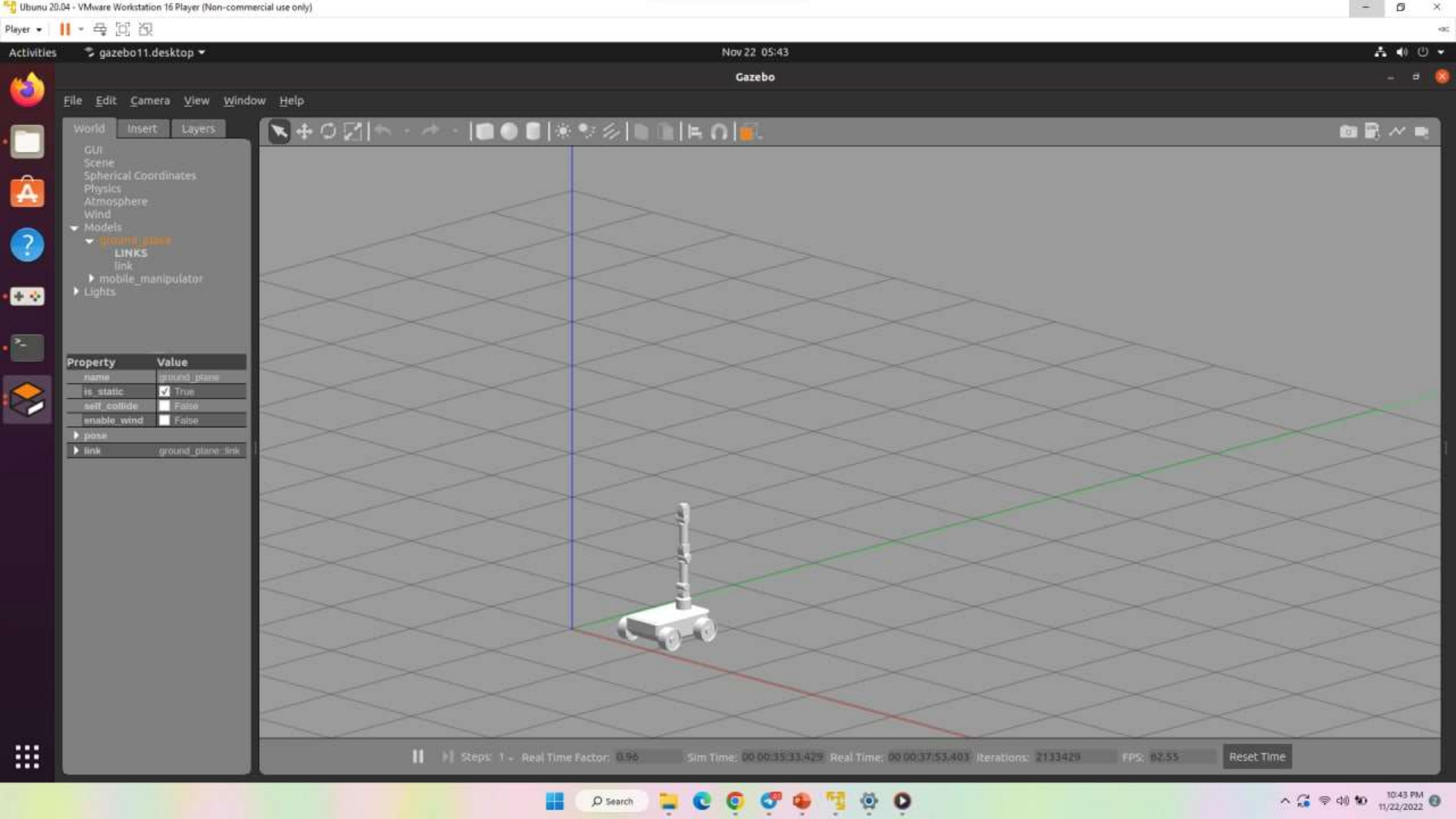
Moving a Simulated Robot Arm to a Goal Using ROS

- Prerequisites
- Create a ROS Package
- Write the Code
- Run the Code



```
rosrun arm_to_goal send_goal_to_arm.py
```




Ubunu 20.04 - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
gazebo11.desktop
Nov 22 05:43
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
mobile_manipulator
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1
Real Time Factor: 0.96
Sim Time: 00 00:35:33.429
Real Time: 00 00:37:53.403
Iterations: 2133429
FPS: 62.55
Reset Time
Search
10:43 PM
11/22/2022


Start

Working on Cartographer part

References:

https://www.semanticscholar.org/paper/Environmental-Measurement-in-Greenhouse-by-Mobile-Kumamoto-Iida/fa91b06d6027db3f58d7601c82047ba8c6570b4e

https://dmexco.com/stories/smart-farming/

https://www.mdpi.com/2073-4395/10/11/1751/htm

https://www.youtube.com/watch?v=BBwEF6WBUQs&ab_channel=WangZheng

https://automaticaddison.com/how-to-move-a-simulated-robot-arm-to-a-goal-using-ros/