

# SYSTEM MODELING

Search and rescue robot  
Technobot

Theophile THOMAS  
Alexandre MENSAH  
Alexandre EANG  
Zhipeng ZENG  
Xiaosen CHEN

15/01/2024

## **Primary Goal:**

Develop a robot capable of autonomously searching, finding, and rescuing a specific target.

## **Design Specifications:**

Divided into three levels - Bronze, Silver, and Gold, with increasing complexity.

- ▶ The Bronze level includes basic autonomous exploration and target detection;
- ▶ The Silver level adds target pick-up and placement capabilities;
- ▶ The Gold level incorporates a remote system interface (GUI).

## **System Architecture:**

The robot is based on the EV3 core system, encompassing modules for direction control, grabbing, pick-and-place, and telemetry.

## **Team introduction:**

- ▶ Theophile THOMAS - Project Manager
- ▶ Alexandre MENSAH - Test Chief Engineer
- ▶ Alexandre EANG - Documentation manager
- ▶ Zhipeng ZENG - Software Chief Engineer
- ▶ Xiaosen CHEN - Hardware Chief Engineer

## **Team Advantages:**

Expertise Across the Board

Collaborative Synergy

Flexibility and adaptability

## Main problem

Create a robot that can search, find and rescue a defined object.

## Test conditions (same for all levels)

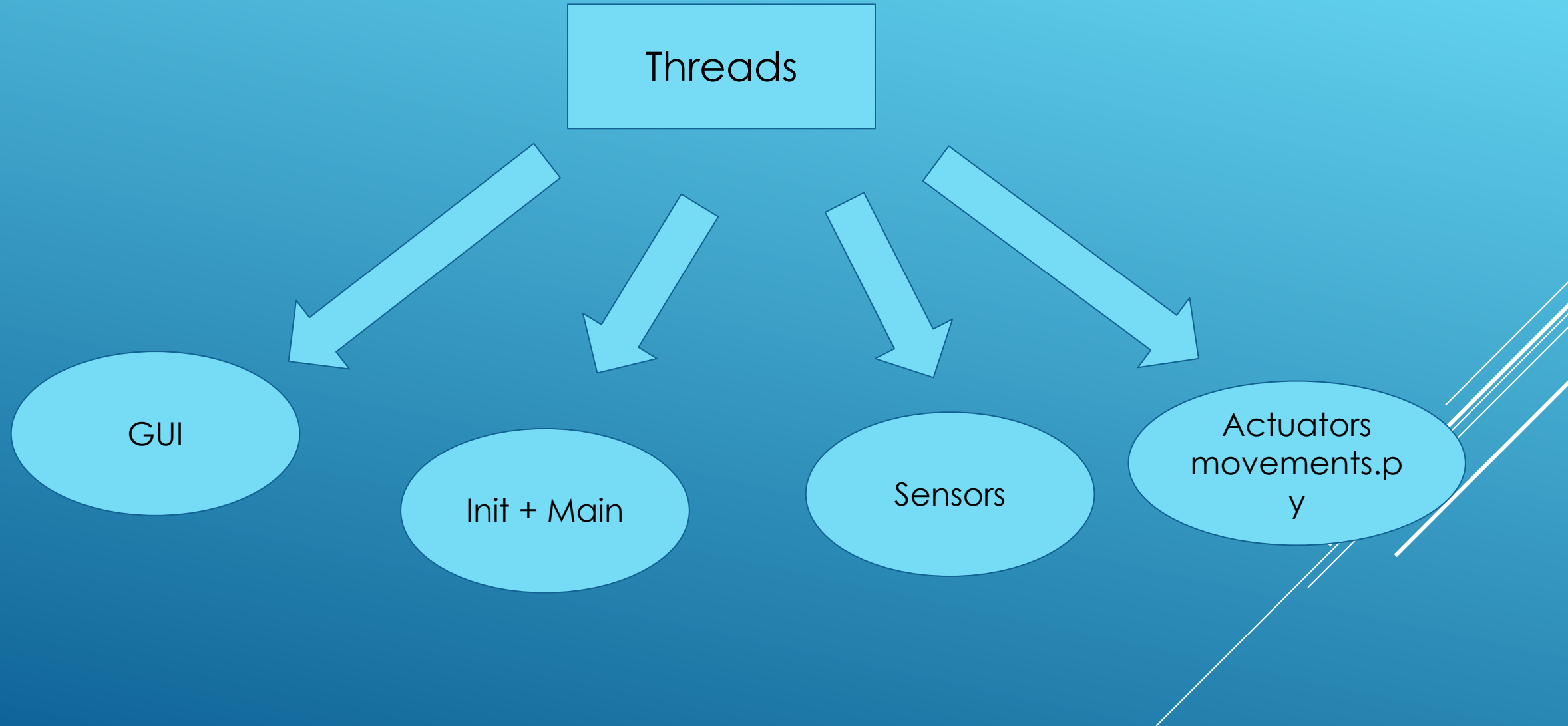
- Ground area 1.5m x 1.5m square
- Area delimited by **black tape** on ground
- 1 target dropped inside area : **cylindrical object**
- Robot starts at one corner of the area

# CODE STRUCTURE

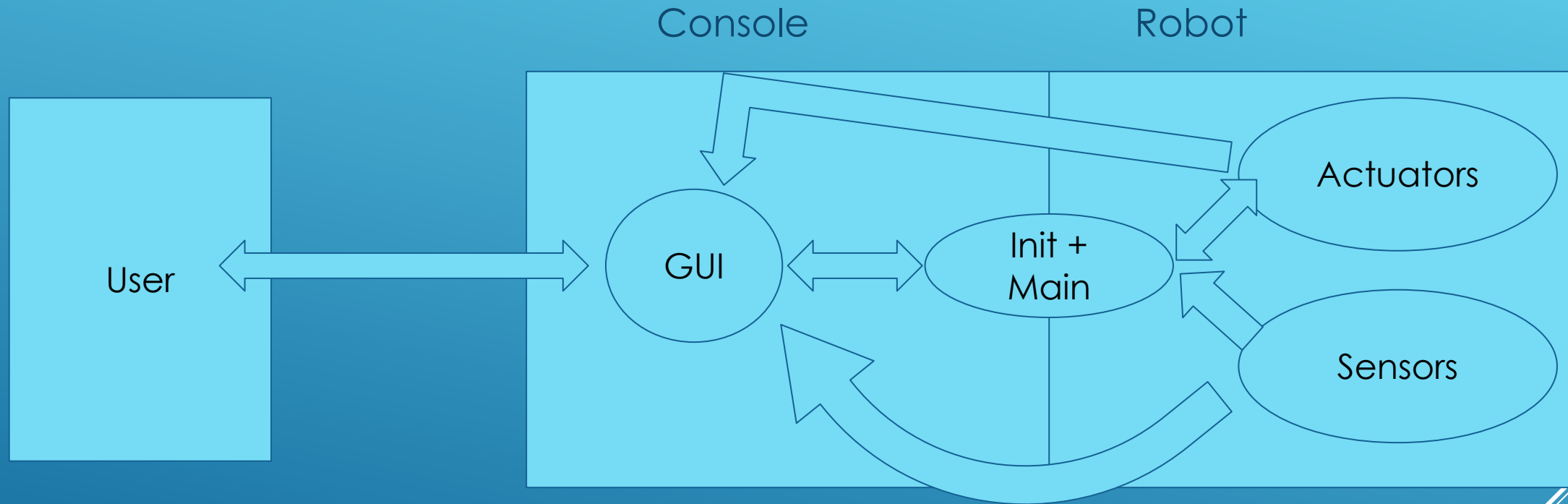
Providing appropriate solutions to the requirements



# Functional block diagram



# Software architecture Diagram



## **Movement.py**

- native functions (from library)
- modified functions
- compound functions
- routine functions (threaded)

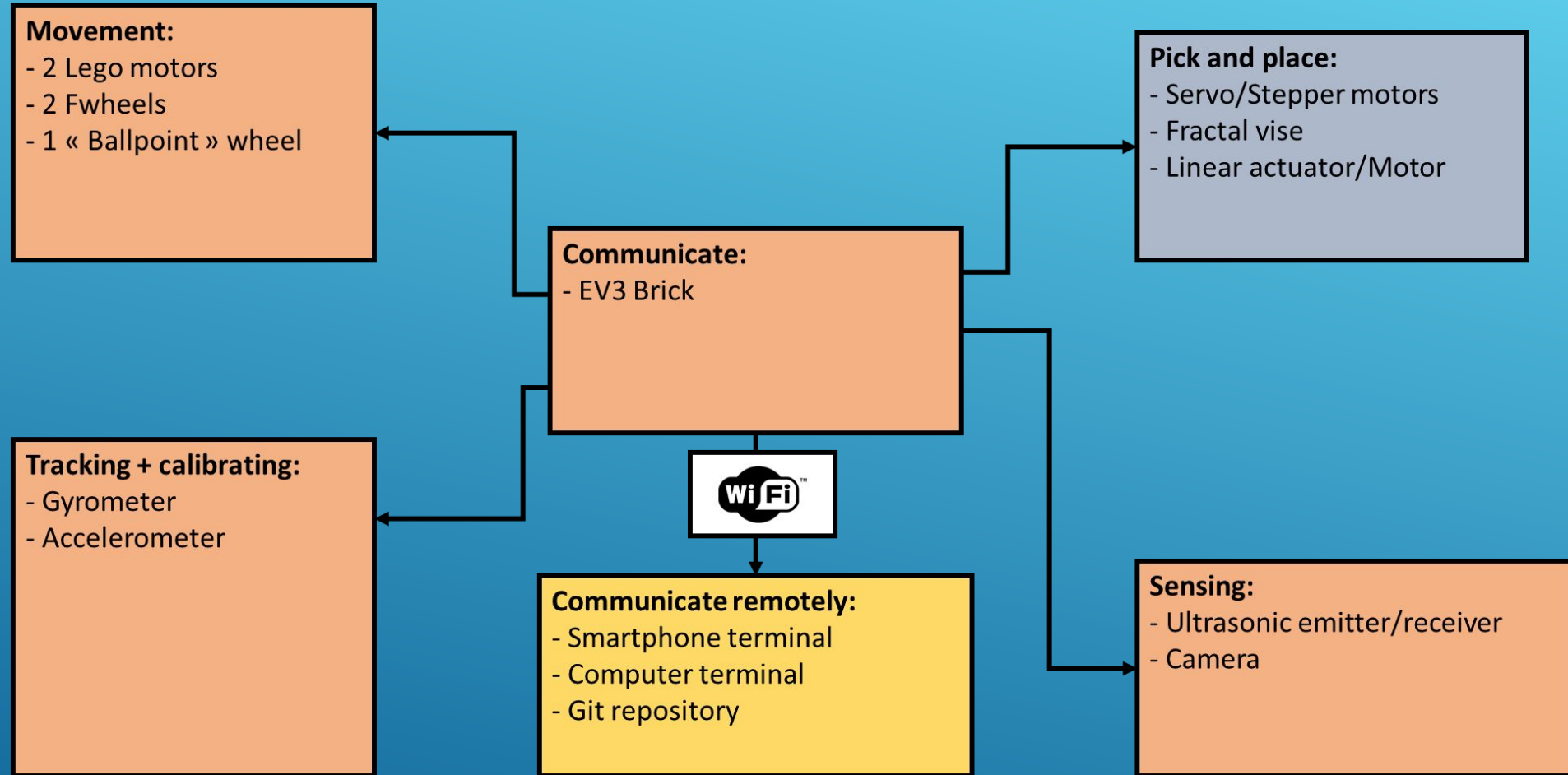


# HARDWARE DESCRIPTION

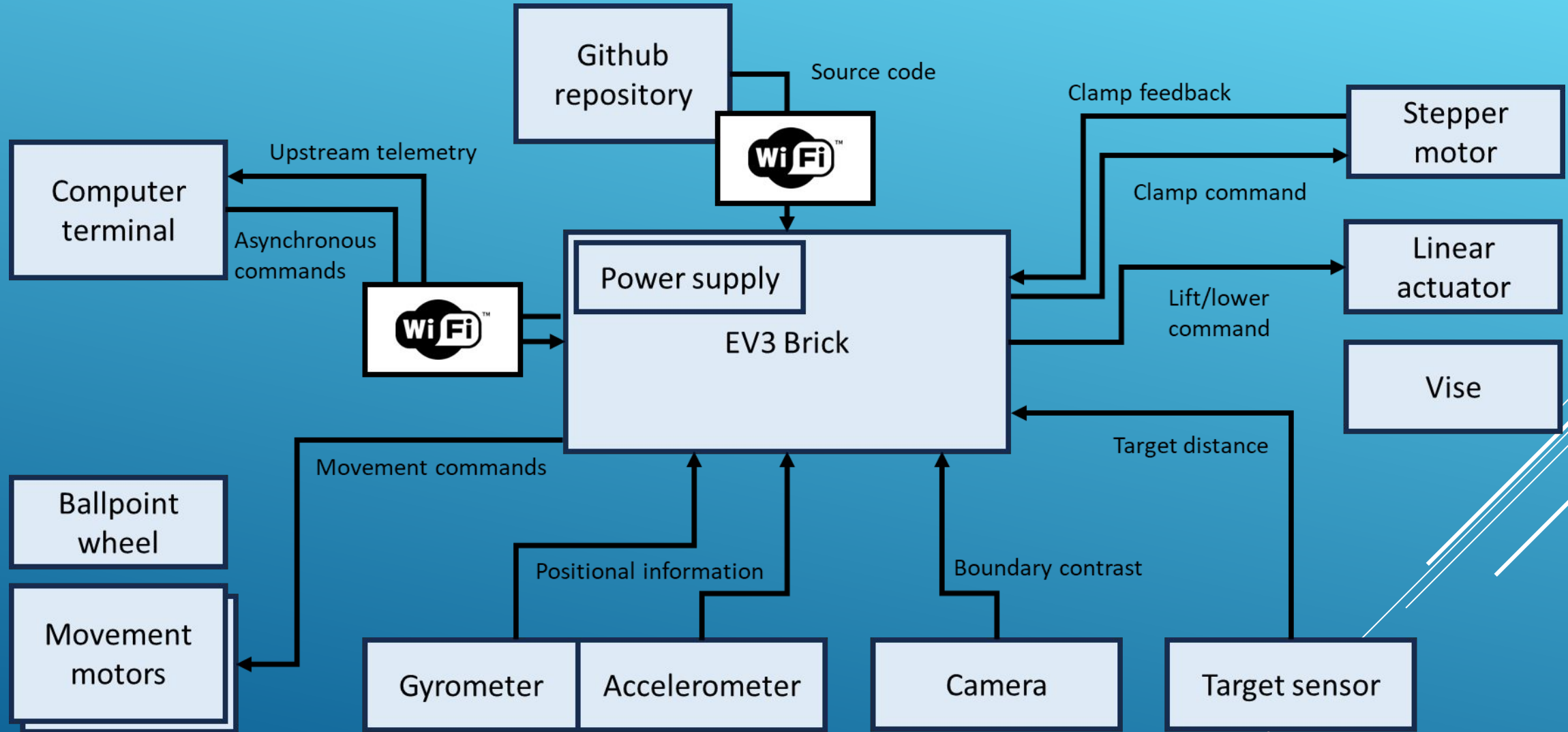
Providing appropriate solutions to the requirements



# General system architecture : functional view



# General system architecture : product view

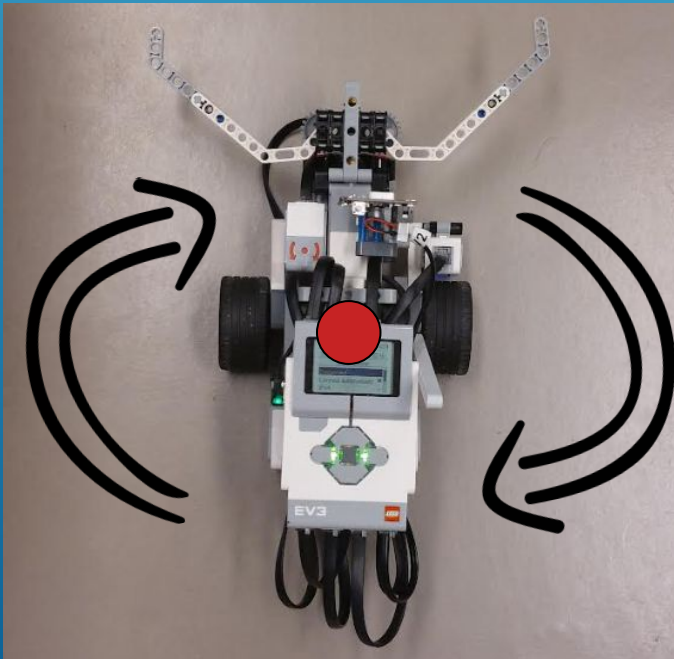


# Robot rotation

2 choices for the rotation

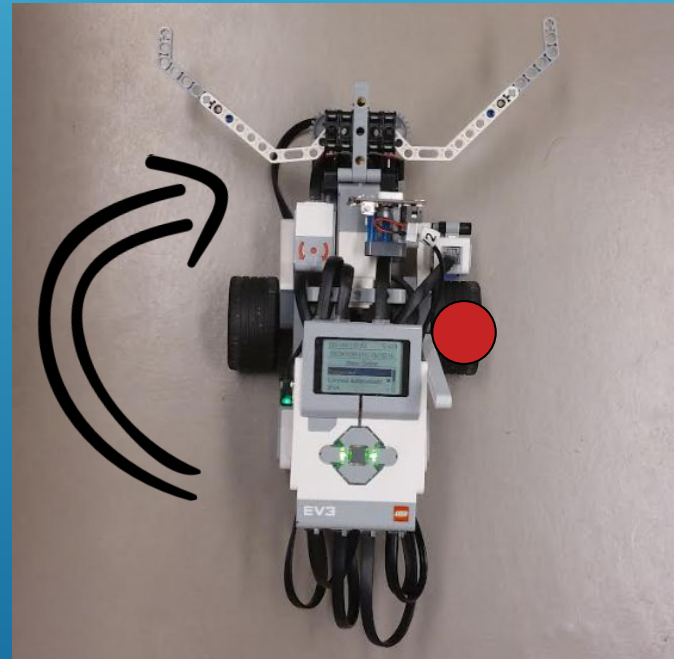
1st choice : Pivot (2 moving wheels)

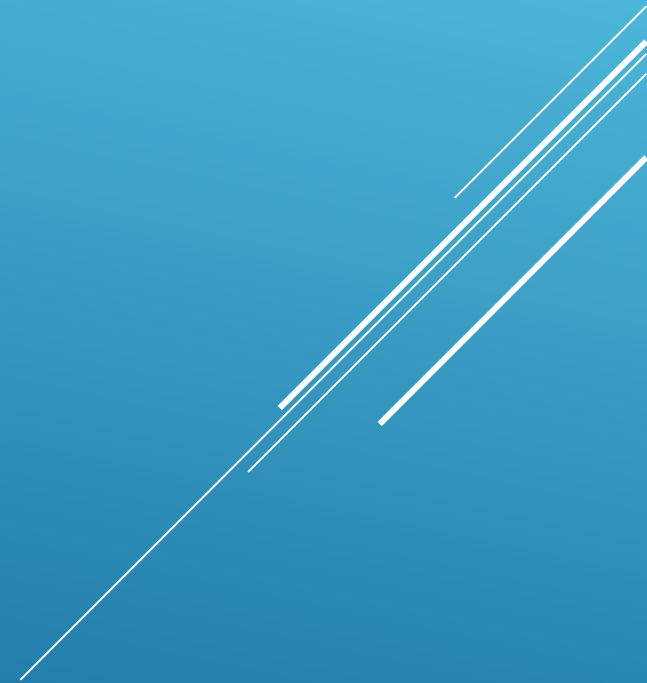
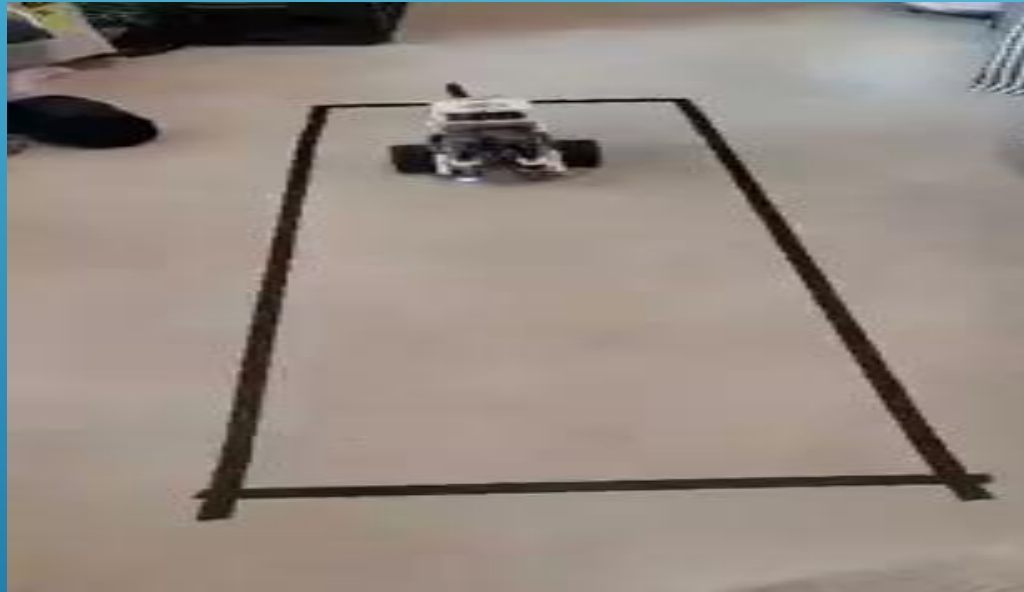
- One wheel will be in indeterminable position (due to servo step jitter)



2nd choice : Turn (1 moving wheel)

- Requires more arithmetics for positional information





# SPECIFICATION TEST REVIEWS

Structuring the end product block by block

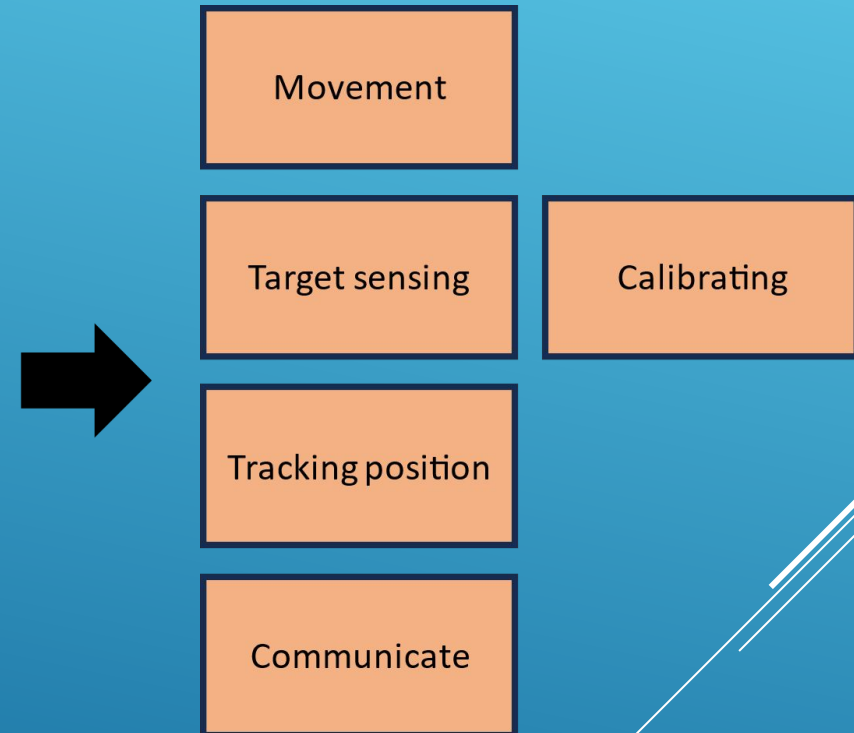


# General system architecture : functional view

## Bronze validation criteria :

- ✓ Robot shall explore area with 1 predefined search pattern ..
- ✓ Robot shall make use of a cartesian coordinate system
- ✓ Robot shall set origin to start point
- ✓ Robot shall detect the target and register its position
- ⊗ Robot shall communicate target position to user through sound, remote interface or built -in display
- ⊗ Robot shall return home after detecting target

## Functions of the bronze system:





# General system architecture : functional view

## Silver validation criteria :

- ✓ Robot shall explore area with 1 predefined search pattern among 3 existing patterns
- ✓ Robot shall make use of a cartesian coordinate system
- ✓ Robot shall set origin to start point
- ✓ Robot shall detect the target and register its position
- ⊗ Robot shall communicate target position to user through sound, remote interface or built-in display
- ⊗ ✓ Robot shall start moving the target home once detected

## Functions of the silver system:





# General system architecture : functional view

## Gold validation criteria :

✓ Robot shall explore area with 1 predefined search pattern among 3 existing patterns remotely

✓ Robot shall make use of a cartesian coordinate system

✓ Robot shall set origin to start point

✓ ⊗ Robot shall detect the target and register its position

⊗ Robot shall communicate target position to user through sound, remote interface or built-in display

✓ Robot shall start moving the target home once detected

⊗ Robot shall communicate its own coordinates to a remote GUI or CLI

✓ ⊗ Robot shall communicate its machine state to a remote GUI or CLI

✓ ⊗ Robot shall asynchronously interpret commands from the remote GUI or CLI

## Functions of the gold system:

Movement

Target sensing

Calibrating

Tracking position

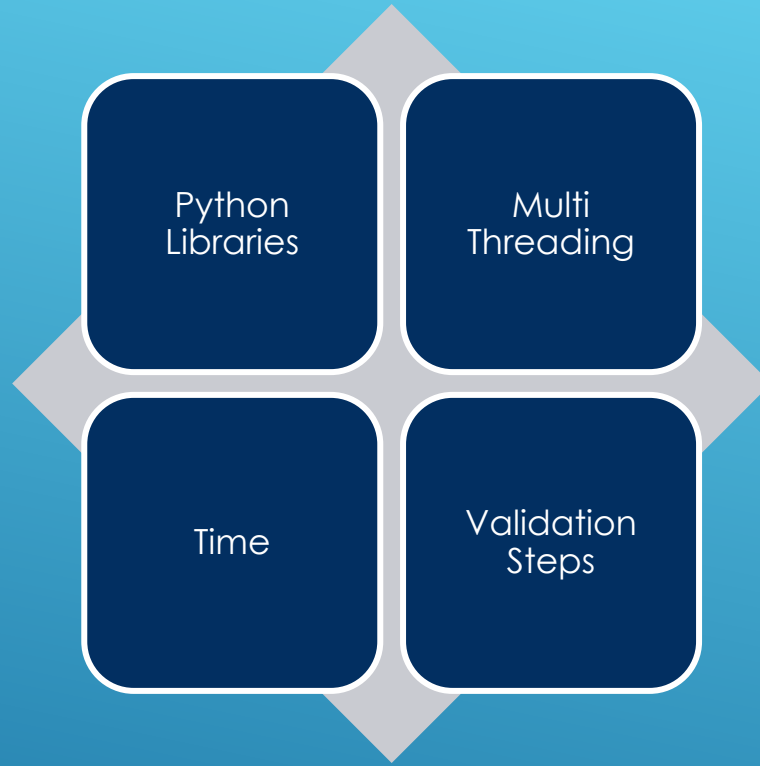
Pick and place

Communicate  
(remotely)

# PROBLEM ENCOUNTERED

Providing appropriate solutions to the requirements





# VARIOUS PROBLEMS

# TEST EXAMPLES

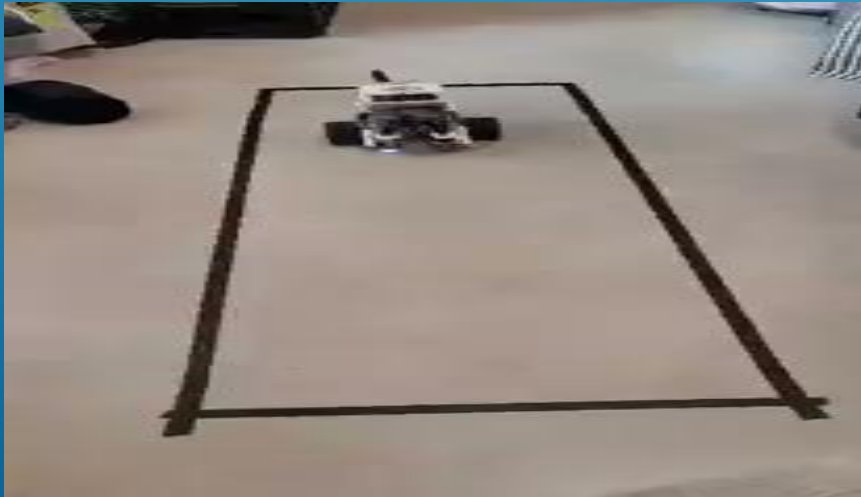
Providing appropriate solutions to the requirements



# Movement - Retracting rectangle

Code:

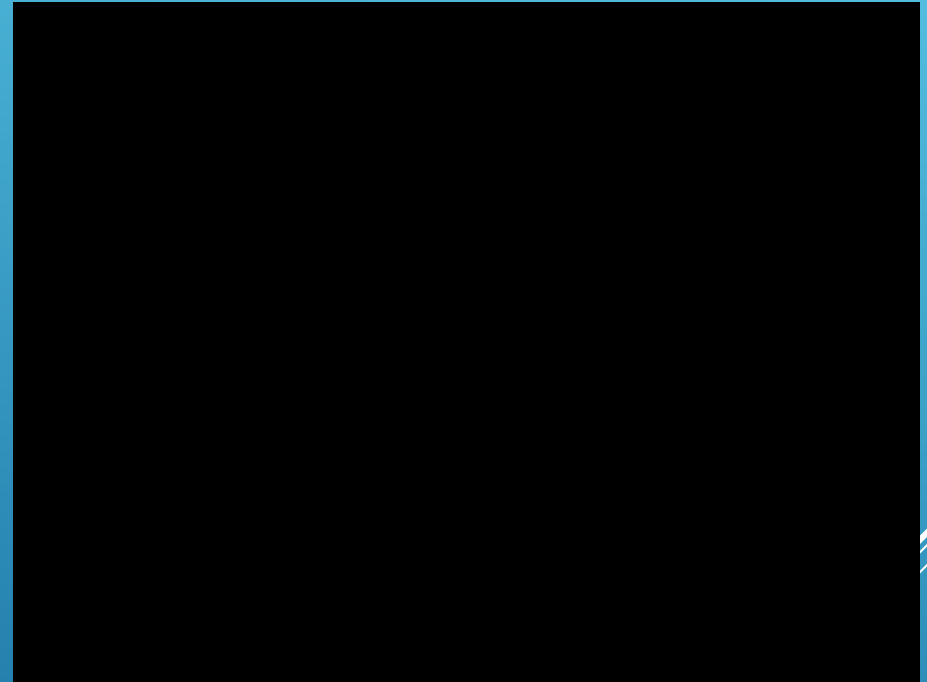
- The robot moves in a retracting square



# Movement - Snake pattern

Code:

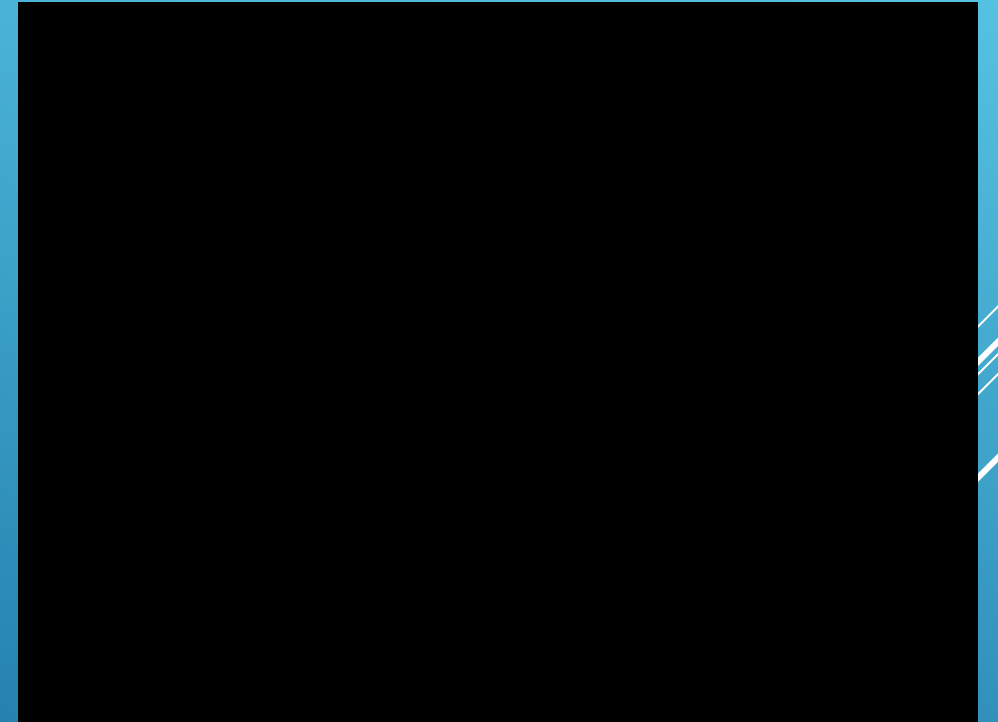
- The robot moves in a snake pattern



# Object detection and grabbing test

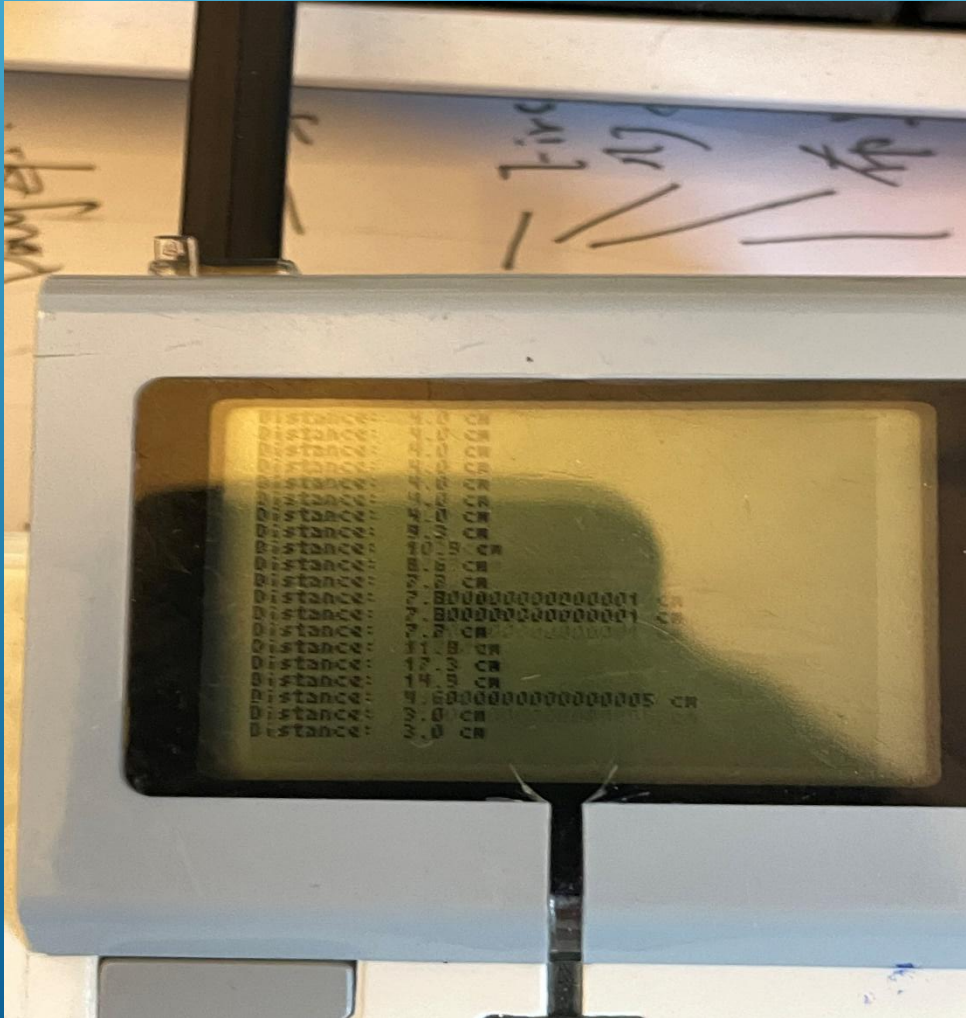
Code:

- Detects the object
- Grabs the object





# Ultrasonic Sensor

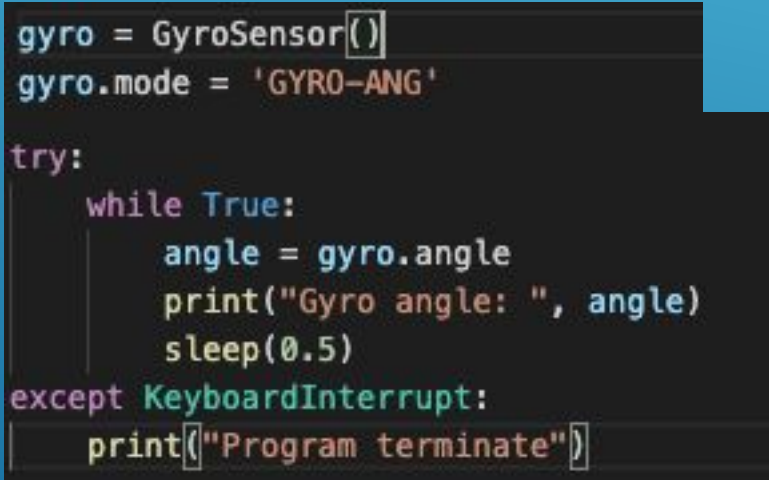


## Ultrasonic Sensor

- Uses reflected sound waves to measure distance between the sensor and any objects in its path.

```
us = UltrasonicSensor()
us.mode = 'US-DIST-CM'
try:
    while True:
        # 读取距离
        distance = us.distance_centimeters
        print("Distance: ", distance, "cm")
        sleep(0.5)
except KeyboardInterrupt:
    print("Program terminated")
```

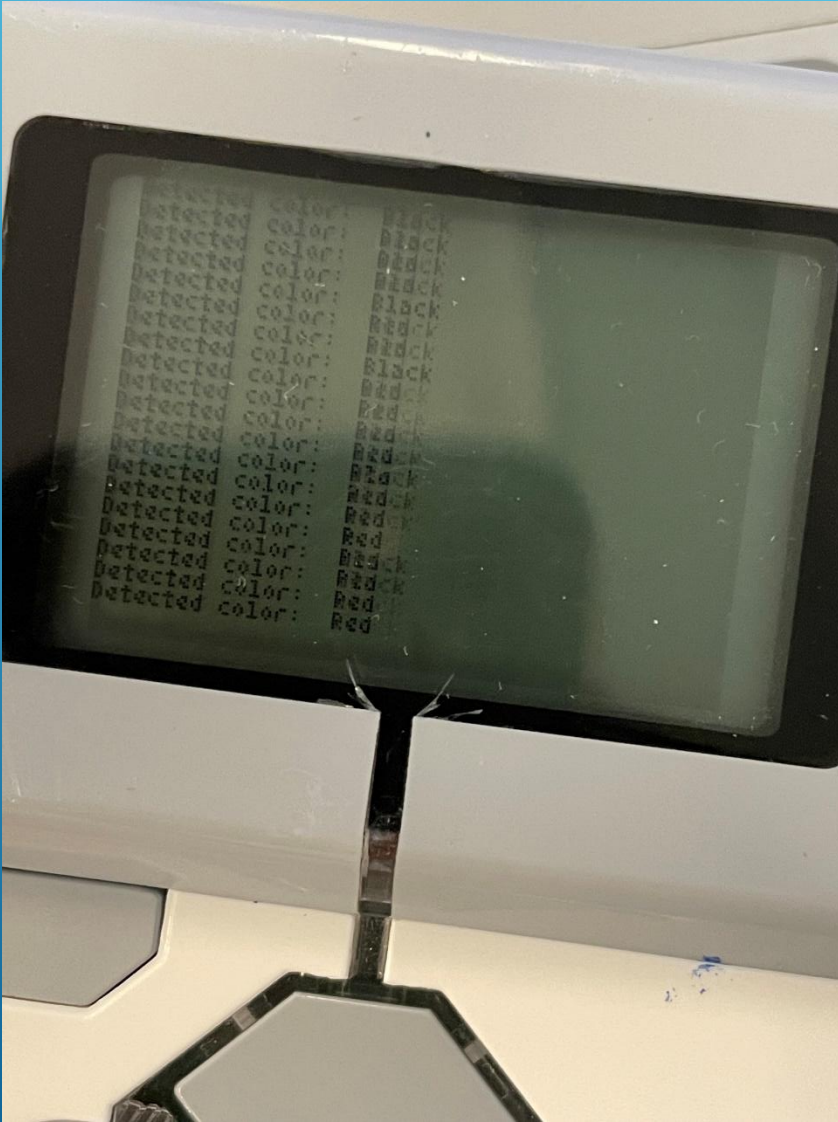


[illegible]

# COLOR SENSOR

```
color_sensor = ColorSensor(INPUT_4)
```

```
while True:  
    color = color_sensor.color_name  
    print("Detected color: ", color)
```



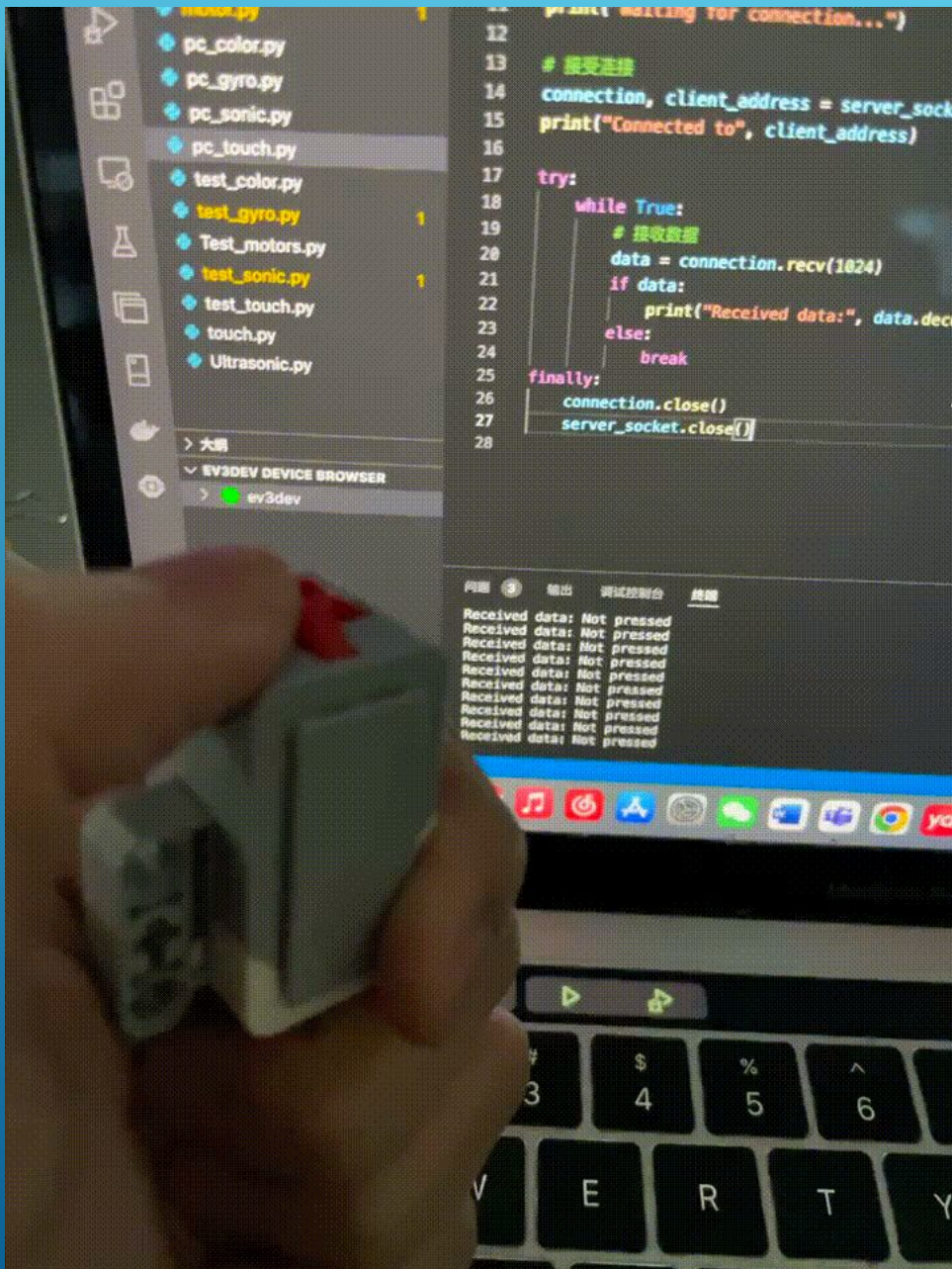


# TOUCH SENSOR

```
touch_sensor = TouchSensor(INPUT_4)
```



```
while True:
    if touch_sensor.is_pressed:
        print("Touch sensor is pressed")
    else:
        print("Touch sensor is not pressed")
```



THANK YOU FOR LISTENING

