

CocoPi

AI for STEM Competition

Asia-Pacific STEAM\_AI Robotics Challenge

# Robot Integration (Secondary)





## Integration Process

P

# CONTENT

O

W

E

P

R

# Routes

1

## Supply

Retrieve, Transport and Deliver Supplies

LCD Screen Setup

Set LCD Screen Rotation to 0°

Create Blank Canvas: canvas Size: Width 320 Height 240

AI Module Motor Driver Setup

AI Module Set Motor C's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to 0 (0-255) Rotating Clockwise

Set travel = 44 open

Set junction\_count = 0

Set speed = 105

Set threshold = 150

Repeat forever

Do

Clear Canvas canvas All Content

Aa

On Canvas canvas Draw Text

Set Start Coordinate:

X: 10 Y: 25

Content

Join strings left: IoT Module Get GPIO # 35 Analog Value

Color:

Text Size: (Integer from 1 to 3) 2

Aa

On Canvas canvas Draw Text

Set Start Coordinate:

X: 160 Y: 25

Content

Join strings right: IoT Module Get GPIO # 36 Analog Value

Color:

Text Size: (Integer from 1 to 3) 2

Show Canvas canvas

If

IoT Module Get GPIO # 35 Analog Value < threshold

end

IoT Module Get GPIO # 36 Analog Value < threshold

Do

Change junction\_count by 1

If

junction\_count < 1

Do

Wait 300 milliseconds

Turnleft

Wait 1000 milliseconds

Set travel = 44 close

Else if

junction\_count == 5

Do

Wait 300 milliseconds

Turnright

Wait 1500 milliseconds

Set travel = 44 close

Else if

junction\_count == 2

or

junction\_count == 4

Do

Wait 300 milliseconds

Else if

junction\_count == 3

Do

Stop

Wait 300 milliseconds

back

Wait 1500 milliseconds

Wait 1500 milliseconds

Set travel = 44 close

Else if

IoT Module Get GPIO # 35 Analog Value < threshold

or

IoT Module Get GPIO # 36 Analog Value < threshold

Do

Set travel = 44 open

If

travel == 44 open

Do

If

IoT Module Get GPIO # 35 Analog Value < threshold

or

IoT Module Get GPIO # 36 Analog Value < threshold

Do

Turnright

Else if

IoT Module Get GPIO # 35 Analog Value < threshold

or

IoT Module Get GPIO # 36 Analog Value < threshold

Do

Turnleft

Else

Wait

Deliver front

AI Module Set Motor C's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Anti-Clockwise

Deliver back

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Deliver right

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Anti-Clockwise

Deliver left

AI Module Set Motor C's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Deliver forward

AI Module Set Motor C's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Deliver stop

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Deliver backward

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Anti-Clockwise

# Routes

2

## Rescue

Pickup and transport injured person to the Aid Station

LCD Screen Setup

Set LCD Screen Rotation to 0°

Create Blank Canvas: canvas Size: Width 320 Height 240

AI Module Motor Driver Setup

AI Module Set Motor C's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to 0 (0-255) Rotating Clockwise

Set travel = "open"

Set junction\_count = 0

Set speed = 105

Set threshold = 150

Repeat forever

Do

Clear Canvas canvas All Content

Aa

On Canvas canvas Draw Text

Set Start Coordinate:

X: 10 Y: 25

Content

Join strings "left"

IoT Module Get GPIO # 35 Analog Value

Color:

Text Size: (Integer from 1 to 3)

2

Aa

On Canvas canvas Draw Text

Set Start Coordinate:

X: 120 Y: 25

Content

Join strings "right"

IoT Module Get GPIO # 36 Analog Value

Color:

Text Size: (Integer from 1 to 3)

2

Show Canvas canvas

If

IoT Module Get GPIO # 35 Analog Value <= threshold

and

IoT Module Get GPIO # 36 Analog Value <= threshold

Do

Change junction\_count by 1

If

junction\_count <= 1

or

junction\_count <= 2

or

junction\_count <= 4

or

junction\_count <= 5

Do

front

Wait 300 Milliseconds

turnright

Wait 1000 Milliseconds

Set travel = "close"

Else if

junction\_count <= 3

Do

front

Wait 300 Milliseconds

stop

Wait 1000 Milliseconds

Set travel = "close"

Else if

IoT Module Get GPIO # 35 Analog Value <= threshold

or

IoT Module Get GPIO # 36 Analog Value <= threshold

Do

Set travel = "open"

If

travel <= "open"

Do

If

IoT Module Get GPIO # 35 Analog Value >= threshold

or

IoT Module Get GPIO # 36 Analog Value <= threshold

Do

turnright

Else if

IoT Module Get GPIO # 35 Analog Value <= threshold

or

IoT Module Get GPIO # 36 Analog Value >= threshold

Do

turnleft

Else

front

Define front

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Define back

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Define right

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Define left

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Define turnright

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

Define turnleft

AI Module Set Motor C's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to speed (0-255) Rotating Clockwise

# Routes

**LCD Screen Setup**

Set LCD Screen Rotation to 0°

Create Blank Canvas: **canvas** Size: Width 320 Height 240

**AI Module Motor Driver Setup**

AI Module Set Motor **C**'s Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor **D**'s Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor **E**'s Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor **F**'s Speed to 0 (0-255) Rotating Clockwise

Set **travel** = "open"

Set **junction\_count** = 0

Set **speed** = 105

Set **threshold** = 150

**Repeat forever**

Do

Clear Canvas **canvas** All Content

**Aa**

On Canvas **canvas** Draw Text

Set Start Coordinate:

X: 10 Y: 25

Content

Join strings "left"

IoT Module Get GPIO # 35 Analog Value

Color:

Text Size: (Integer from 1 to 3) 2

**Aa**

On Canvas **canvas** Draw Text

Set Start Coordinate:

X: 120 Y: 25

Content

Join strings "right"

IoT Module Get GPIO # 36 Analog Value

Color:

Text Size: (Integer from 1 to 3) 2

Show Canvas **canvas**

## 3

### Obstacle

Retrieve debris from mechanical robot and transport it to the debris clearing zone

**IoT Module Get GPIO # 35 Analog Value < threshold**

**IoT Module Get GPIO # 36 Analog Value < threshold**

Do

Change **junction\_count** by 1

If

**junction\_count** = 1

Do

Front

Wait 300 Milliseconds

Turnright

Wait 1000 Milliseconds

Set **travel** = "close"

Else if

**junction\_count** = 3

Do

Front

Wait 300 Milliseconds

stop

Wait 1000 Milliseconds

Set **travel** = "close"

Else if

**junction\_count** = 4

Do

Front

Wait 300 Milliseconds

Turnleft

Wait 1000 Milliseconds

Set **travel** = "close"

Else if

**IoT Module Get GPIO # 35 Analog Value < threshold**

or

**IoT Module Get GPIO # 36 Analog Value < threshold**

Do

Set **travel** = "open"

If

**travel** = "open"

Do

If

**IoT Module Get GPIO # 35 Analog Value > threshold**

or

**IoT Module Get GPIO # 36 Analog Value > threshold**

Do

Turnright

Else if

**IoT Module Get GPIO # 35 Analog Value < threshold**

or

**IoT Module Get GPIO # 36 Analog Value > threshold**

Do

Turnleft

Else

Front

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Anti-Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Anti-Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Anti-Clockwise

**Define Event**

AI Module Set Motor **C**'s Speed to speed (0-255) Rotating Clockwise

AI Module Set Motor **D**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **E**'s Speed to speed (0-255) Rotating Anti-Clockwise

AI Module Set Motor **F**'s Speed to speed (0-255) Rotating Clockwise

# Image Identification

\*Main code (See next page for functions)

**Camera Setup**

Set Image Capturing Size: QVGA (320\*240)

Set Image Capturing Color Mode: Colorful

**LCD Screen Setup**

Set LCD Screen Rotation to: 0°

Create Blank Canvas: Canvas Size: Width: 320 Height: 240

Set Canvas Canvas's Starting Coordinate as: X: 48 Y: 8

**Initialize A.I. Model**

Load model file from path /root/preset/model/cocopi\_FloodingMission

Define class name by list:

- Use these values to create an array of Numbers: "Supply\_Station"
- "Rescue\_Station"
- "Cleaning\_Zone"
- "A"
- "B"
- "C"
- "D"

**AI Module Motor Driver Setup**

AI Module Set Motor C's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor D's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor E's Speed to 0 (0-255) Rotating Clockwise

AI Module Set Motor F's Speed to 0 (0-255) Rotating Clockwise

**AI Module Servo Setup**

AI Module Set Servo on GPIO # P0 Rotate to 0 Degree (0°~180°)

AI Module Set Servo on GPIO # P1 Rotate to 95 Degree (0°~180°)

Set travel = "open"

Set junction\_count = 0

Set speed = 105

Set threshold = 150

**Repeat forever**

Do **detection**

If **IoT Module Get GPIO # 35 Analog Value < threshold** and **IoT Module Get GPIO # 36 Analog Value < threshold**

Do **Change junction\_count by 1**

If **junction\_count ≤ 1**

Do **front**

Wait 300 Milliseconds

turnleft

Wait 1000 Milliseconds

Set travel = "close"

Else if **junction\_count = 5**

Do **front**

Wait 300 Milliseconds

turnright

Wait 1500 Milliseconds

Set travel = "close"

Else if **junction\_count = 2** or **junction\_count = 4**

Do **front**

Wait 300 Milliseconds

Else if **junction\_count = 3**

Do **stop**

Wait 300 Milliseconds

back

Wait 1500 Milliseconds

turnright

Wait 1500 Milliseconds

Set travel = "close"

Else if **IoT Module Get GPIO # 35 Analog Value < threshold** or **IoT Module Get GPIO # 36 Analog Value < threshold**

Do **Set travel = "open"**

If **travel = "open"**

Do If **IoT Module Get GPIO # 35 Analog Value > threshold** or **IoT Module Get GPIO # 36 Analog Value < threshold**

Do **turnright**

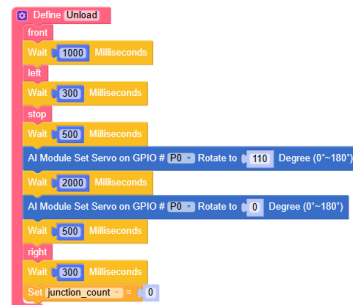
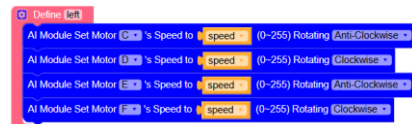
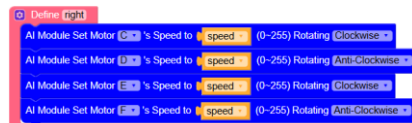
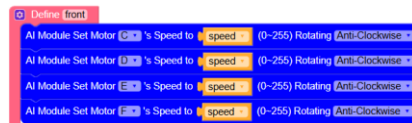
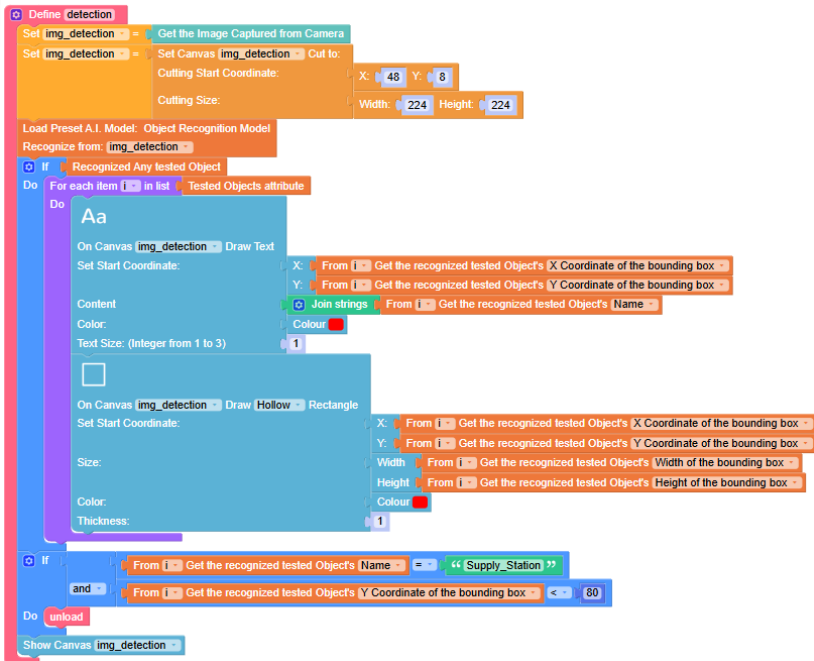
Else if **IoT Module Get GPIO # 35 Analog Value < threshold** or **IoT Module Get GPIO # 36 Analog Value > threshold**

Do **turnleft**

Else **front**

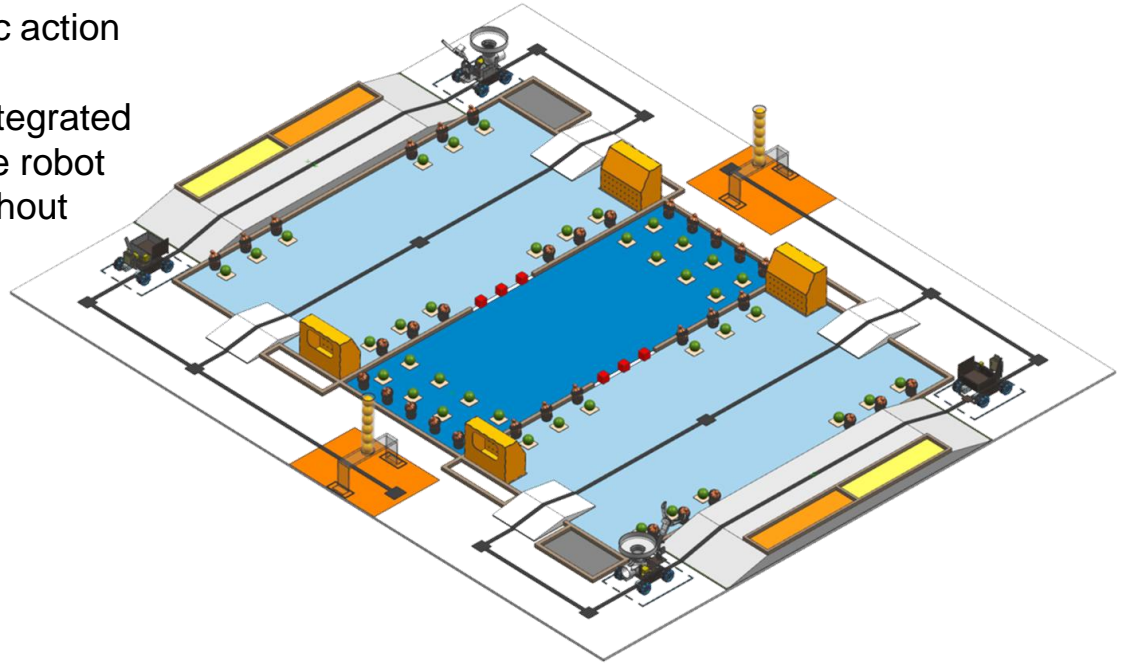
\*Function Definition (See previous page for main code)

# Image Identification



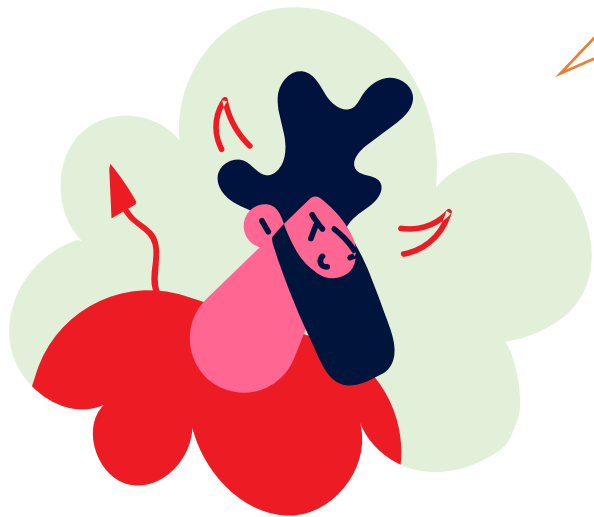
## ● Integrating Sub-sections

- Each program shown on the previous slides correspond to a specific action that the robot performs.
- The different parts must be integrated under a single program for the robot to operate on the playfield without external inputs.





## ● Path Decision Making



Since there are 3 paths, how does the robot decide which and when to take each path



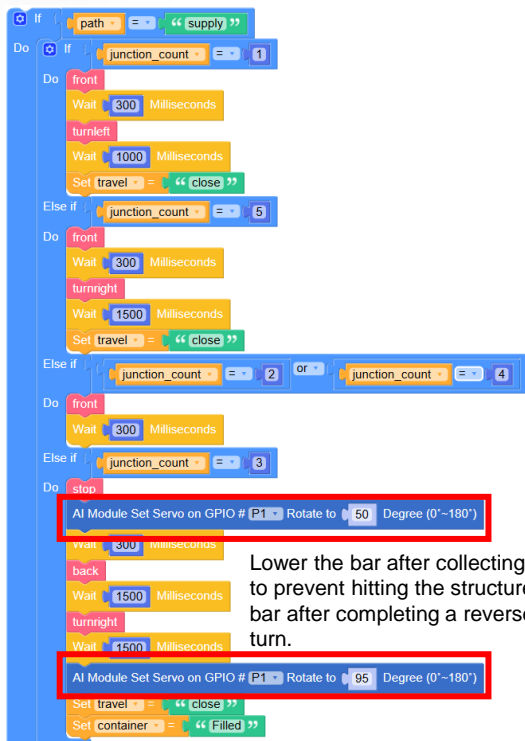
## ● Path Decision Making

Create a variable **Path**, with three states: **Supply, Rescue, Obstacle**

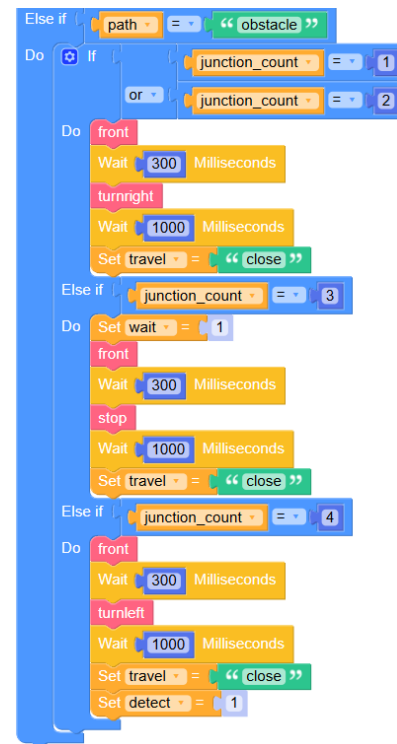
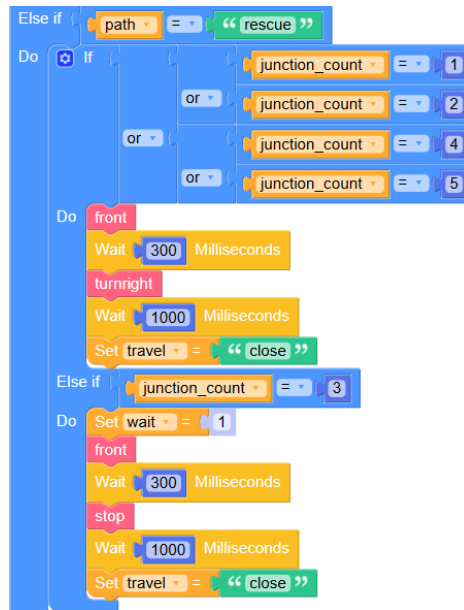
- **Path = Supply:** Robot takes the supply route;
- **Path = Rescue:** Robot takes the rescue route;
- **Path = Obstacle:** Robot takes the obstacle route;

\*The initial state for **Path** to be set as **Supply**

# Path Decision Making

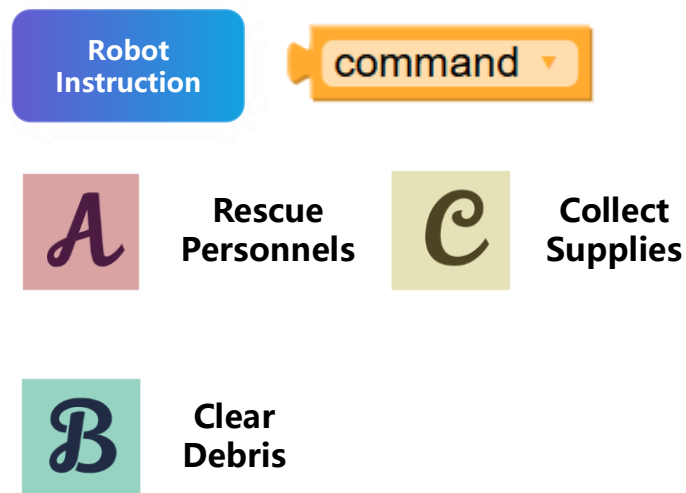
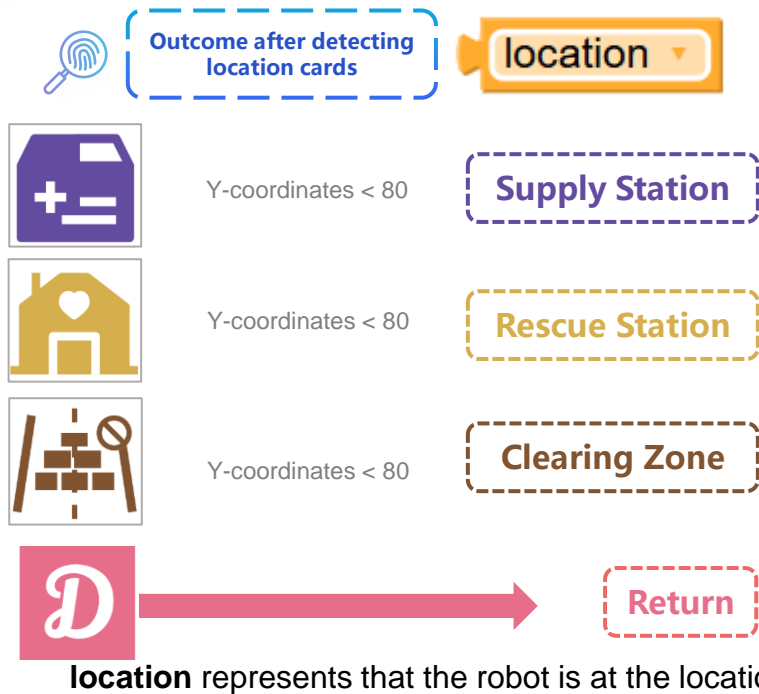


Lower the bar after collecting the supplies to prevent hitting the structure. Raise the bar after completing a reverse and right turn.



## Decision Making

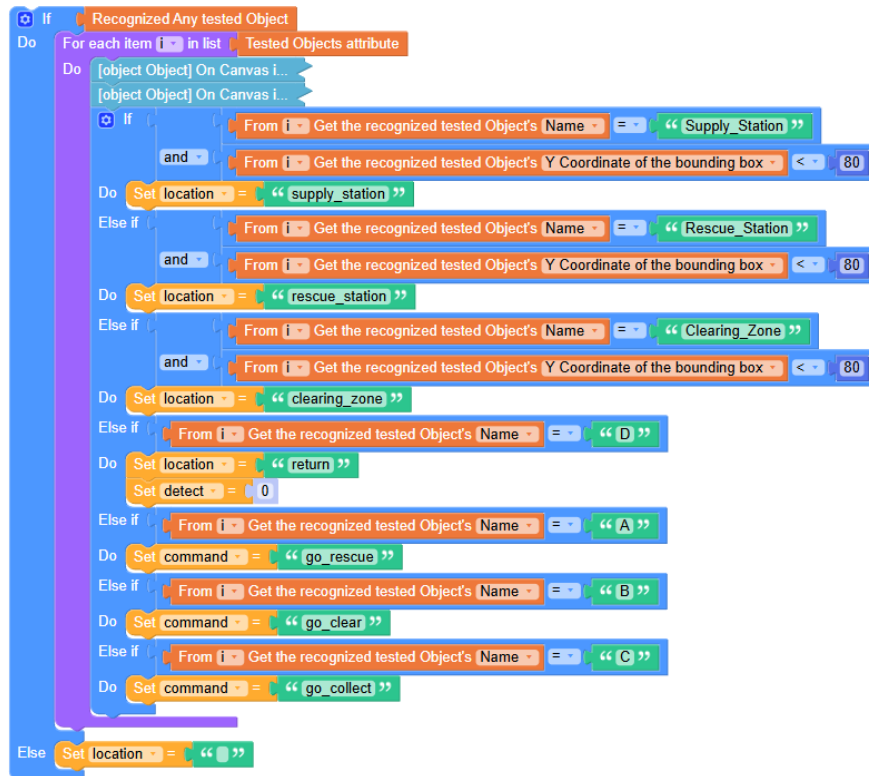
The actions of the robot is dependent on the **location/command card** it detects



**command** stores the instruction the robot receives after it detects the command card

# Decision Making

Detection  
Outcome Code



# ● Object Detection

Does the robot have to constantly do object detection?

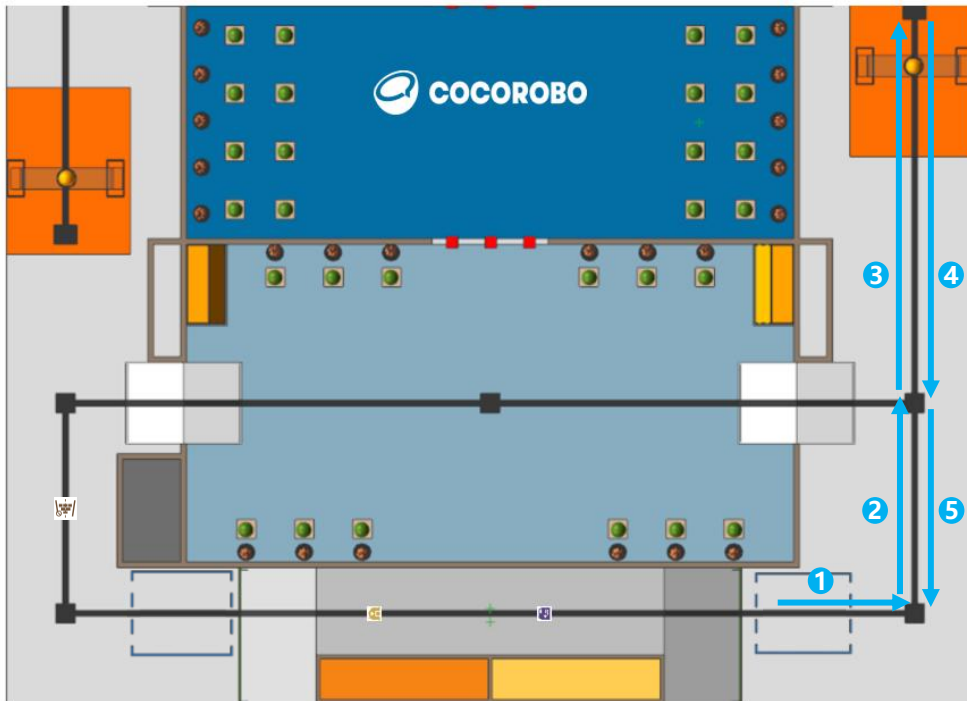
Which path does not require object detection?



- Stop the object detection feature when it is not required
- It will reduce the chances of false detection which leads to errors.

## Analysis

When is the object detection feature needed during the "Supply" route.



Set detect = 0

Detection "Off"

Set detect = 1

Detection "On"

```

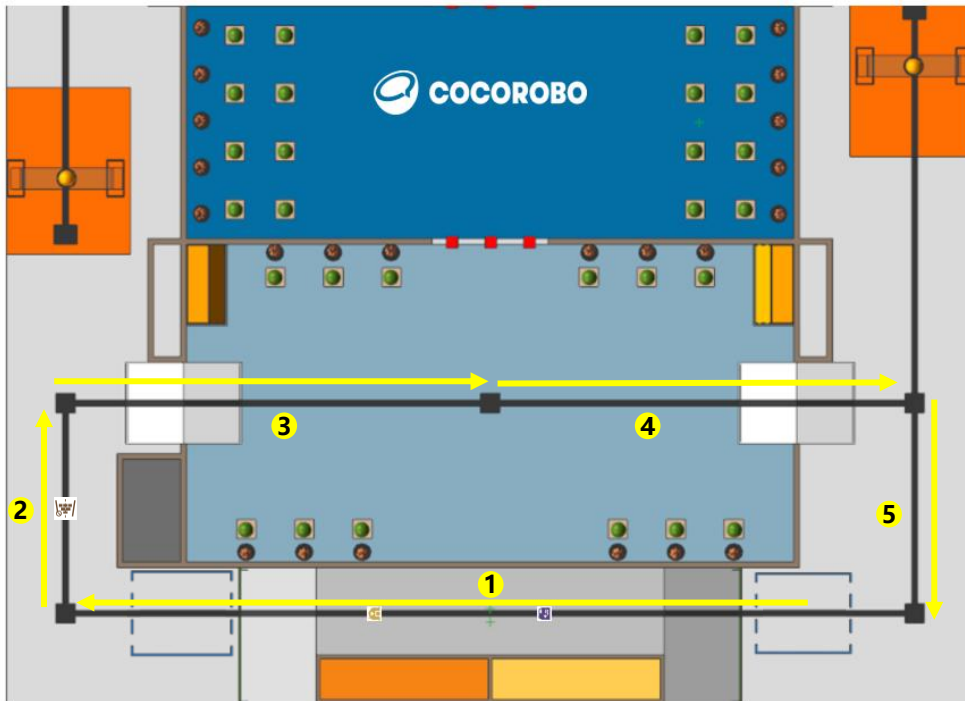
if junction_count <= 1
  Do
    front
    Wait 300 Milliseconds
    turnleft
    Wait 1000 Milliseconds
    Set travel = "close"
    Set detect = 0
  Else if junction_count <= 5
    Do
      front
      Wait 300 Milliseconds
      turnright
      Wait 1500 Milliseconds
      Set travel = "close"
      Set detect = 1
    Else if junction_count <= 2 or junction_count <= 4
      Do
        front
        Wait 300 Milliseconds
      Else if junction_count <= 3
        Do
          stop
          AI Module Set Servo on GPIO # P1 Rotate to 50 Degree (0~180)
          Wait 300 Milliseconds
          back
          Wait 1500 Milliseconds
          turnright
          Wait 1500 Milliseconds
          AI Module Set Servo on GPIO # P1 Rotate to 95 Degree (0~180)
          Set travel = "close"
          Set container = "Filled"
  
```

Turn off detection after passing the first junction

Turn on detection after passing the last junction, robot needs to locate position to drop off supplies

## Analysis

When is the object detection feature needed during the "Rescue" route.

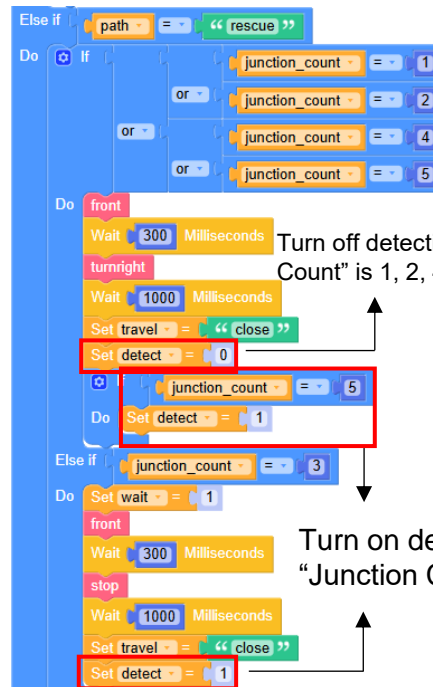


Set detect = 0

Detection "Off"

Set detect = 1

Detection "On"



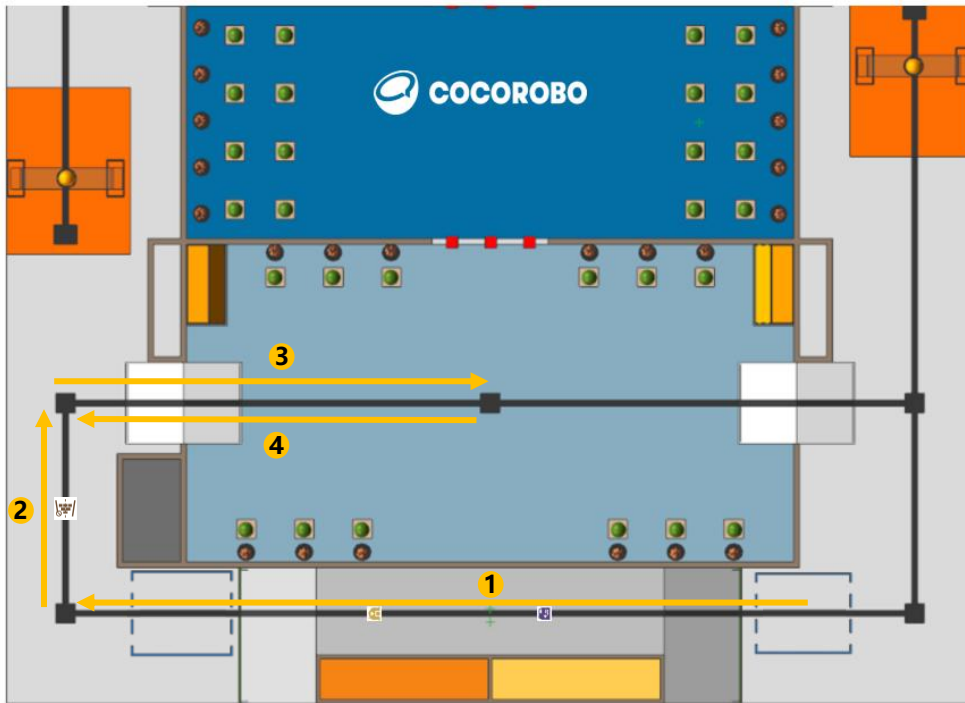
Turn off detection when "Junction Count" is 1, 2, 4 or 5.

Turn on detection when "Junction Count" = 5 or = 3



# Analysis

When is the object detection feature needed during the "Obstacle" route.

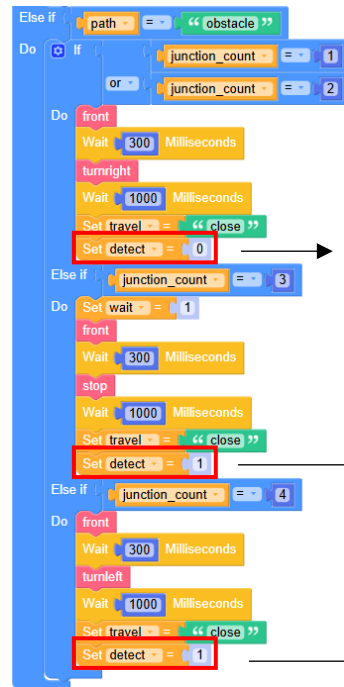


Set detect = 0

Detection "Off"

Set detect = 1

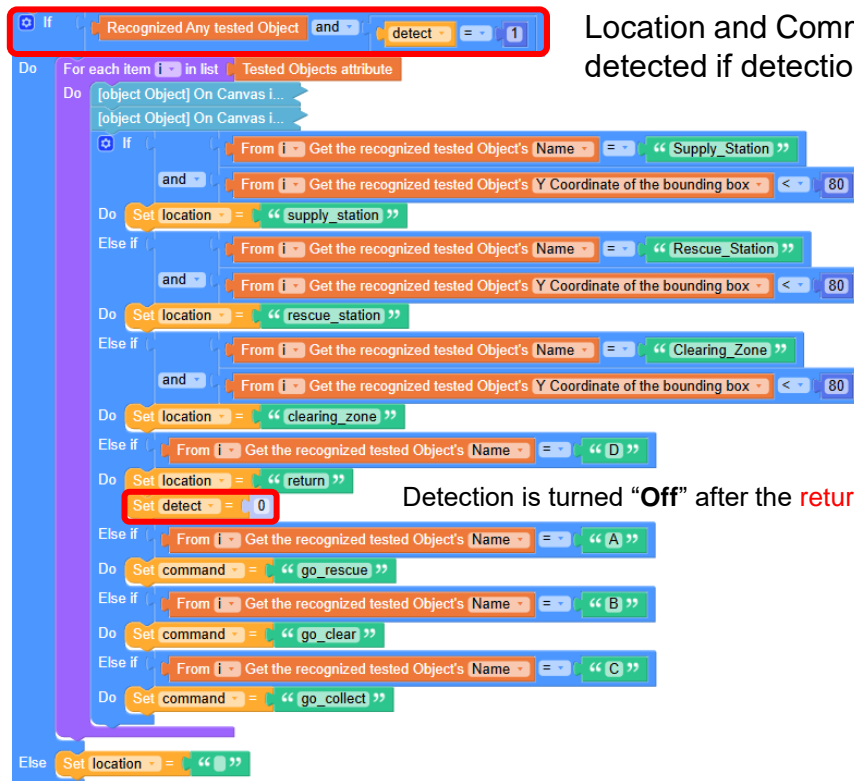
Detection "On"



Turn off detection when  
"Junction Count" = 1 or = 2.

Turn on detection when  
"Junction Count" = 3 or = 4

## Detection (On/Off)



Location and Command cards will only be detected if detection is turned “On”

Detection is turned “Off” after the **return** card on the mechanical robot is detected

## ● Decision Making

Earlier, we discussed how the robot decides which path to take. Now we will work on letting the robot know when to change the path taken.

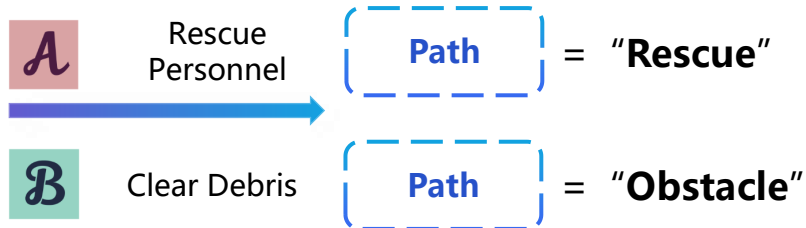
- Create a variable **Container**, with two states: **Filled**, **Empty**. It indicates if the robot is carrying (flood water, personnels or debris)
- The initial state of Container is Empty, the state will changed to filled when:
  - **Collected supplies**
  - **Rescued personnels**
  - **Received return instruction from the mechanical robot**
- The decision to unload or change path is dependent on the **robot's position**, **current route**, **container status** and **command received**.



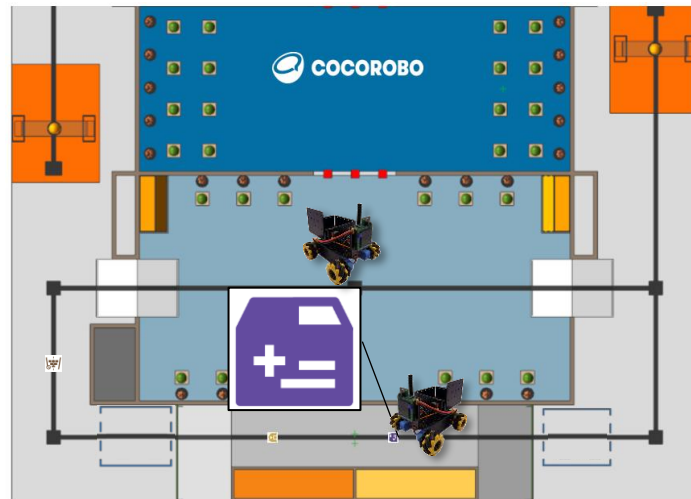
## Decision Making

**Location** set as **Supply Station** & **Path** = "Supply" & **Container** = "Filled"

The robot should go **unload** then follows the **command** to choose the subsequent route.

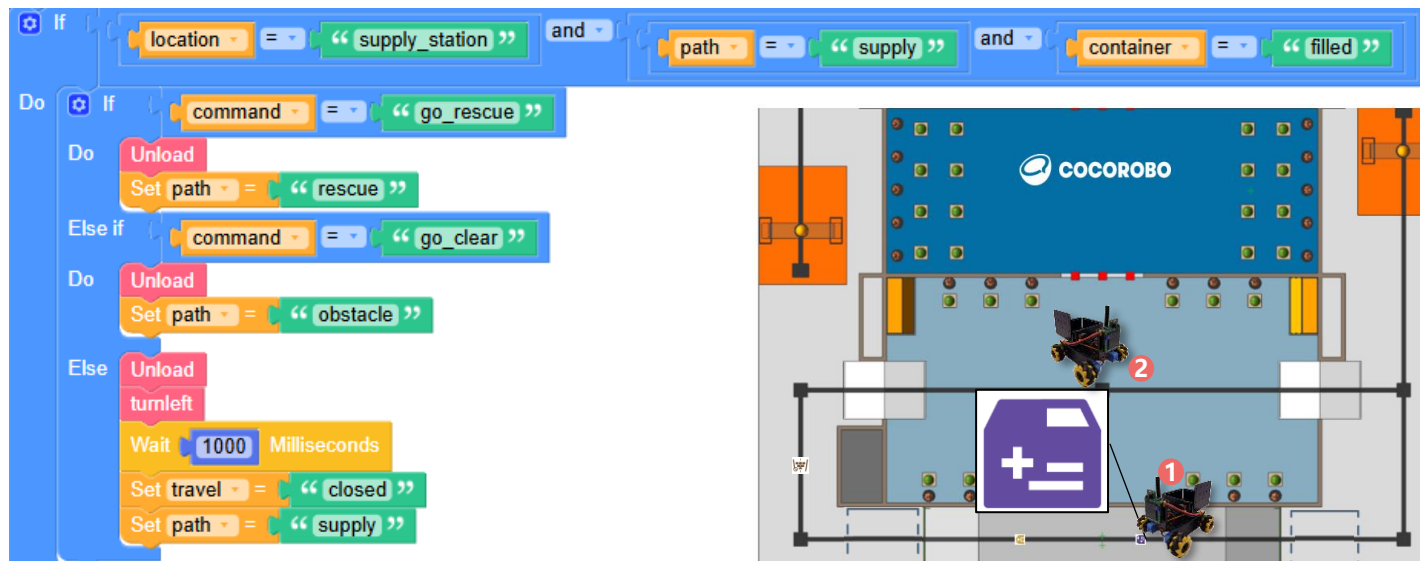


If the command received is neither **A** nor **B**, the robot will continue supply collection. It will turn right and continue down the supply path.



## Decision Making

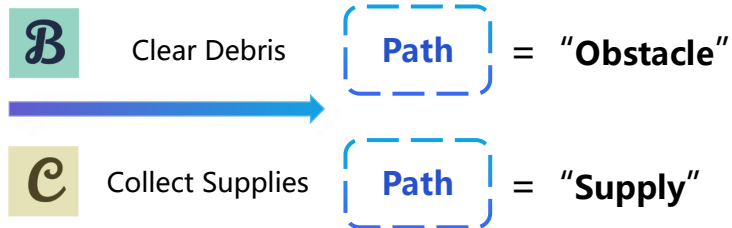
**Location** set as **Supply Station** & **Path** = "Supply" & **Container** = "Filled"



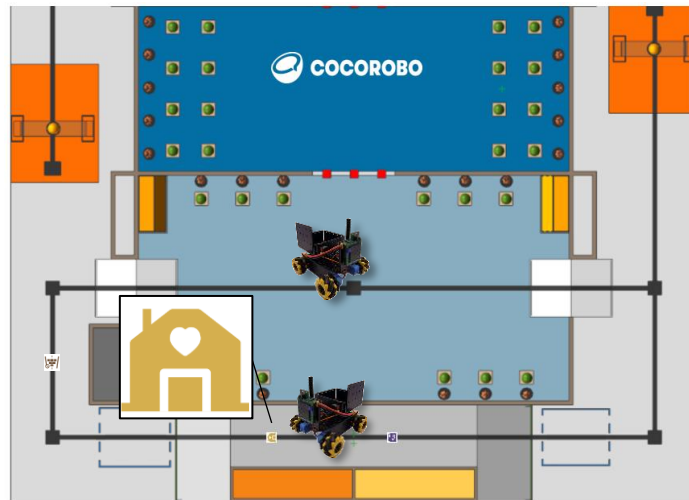
## Decision Making

**Location** set as **Rescue Station** & **Path** = "Rescue" & **Container** = "Filled"

The robot should go **unload** then follows the **command** to choose the subsequent route.

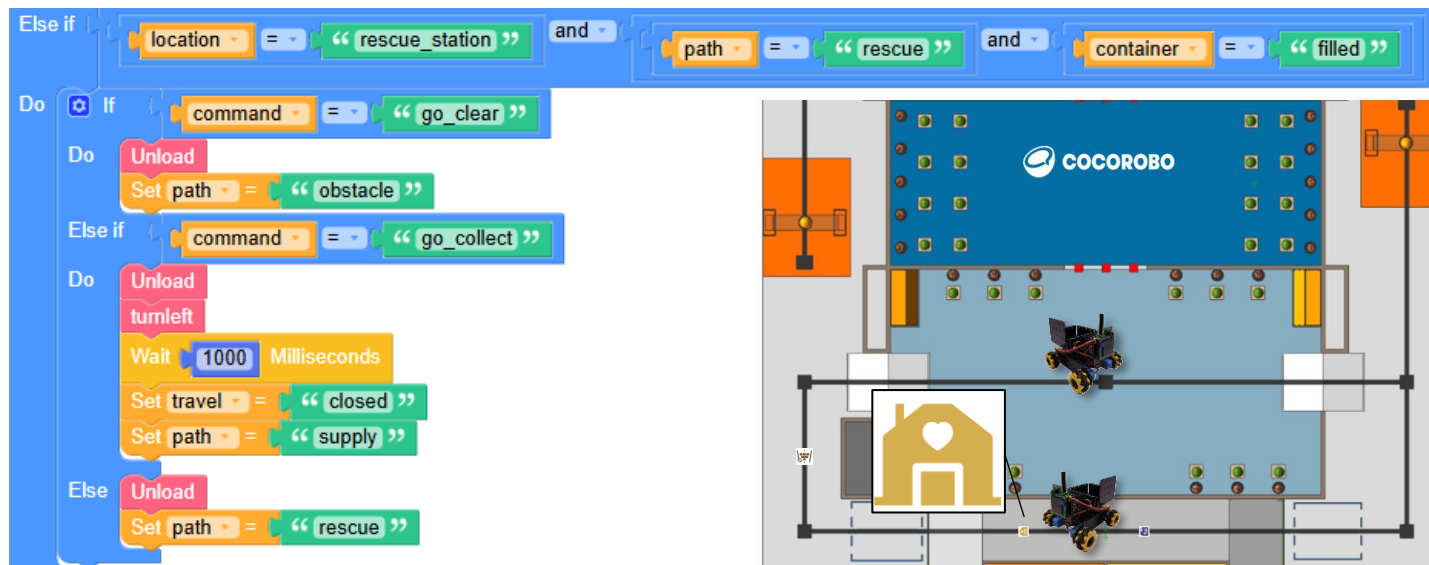


If the command received is neither **B** nor **C**, the robot will continue rescuing personnels. It will continue down the supply path.



## Decision Making

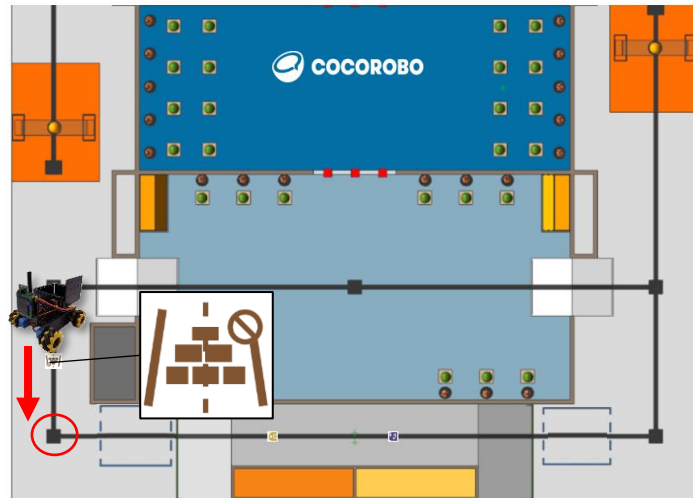
**Location** set as **Rescue Station** & **Path** = "Rescue" & **Container** = "Filled"



## Decision Making

**Location** set as **Clearing Zone** & **Path** = "Obstacle" & **Container** = "Filled"

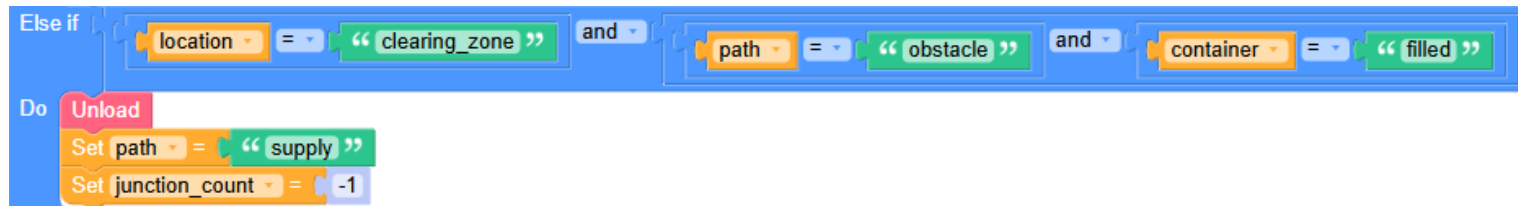
The robot will go **unload** first. Since the robot is currently on the obstacle route, its **movement will be towards the starting point**. It does so through the supplies route. And because there is **still a junction in the road to the starting point**, it is necessary to set the "**Junction Count**" to **-1**, so that when it reaches the starting point it will be 0.





## Decision Making

**Location** set as **Clearing Zone** & **Path** = "Obstacle" & **Container** = "Filled"



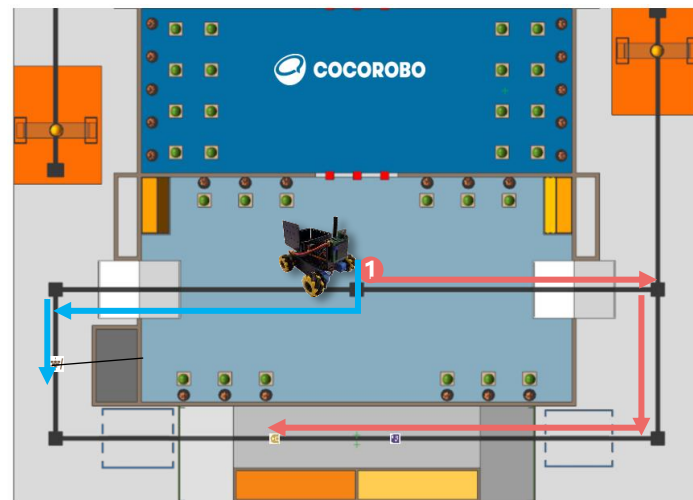
## Decision Making

When the transport robot reaches point ①, it will wait for the mechanical robot to load objects onto its container. After the process is complete, it will detect the command card on the mechanical robot and continue on with its actions

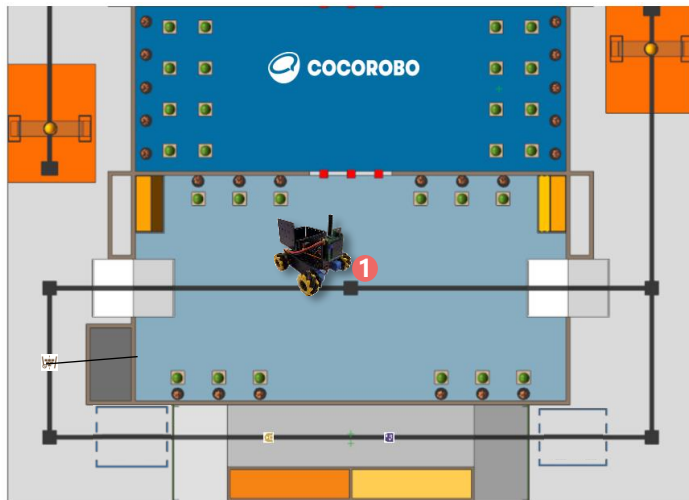
**D** → **Location** set as **Return**, it will operate based on the **path** set

**Path** = "Rescue" →

**Path** = "Obstacle" →



## ● Pause Sequence Analysis

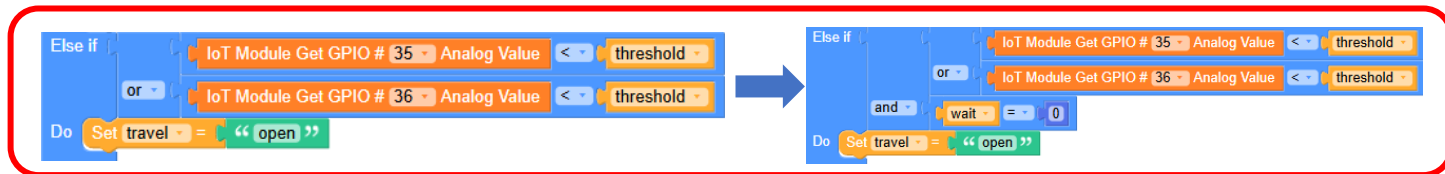


After the transport robot arrives at point ①, it waits there until it detects the **D** instruction from the mechanical robot. When the **D** instruction is received, it performs the return sequence

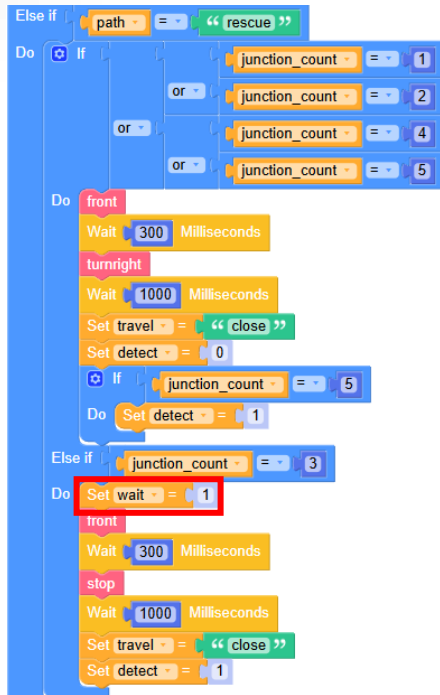
Create a variable **wait** (represents stay at the location), with two states, 0, 1:

- When **wait** = 0 robot will continue with its actions;
- When **wait** = 1 robot will pause and wait;

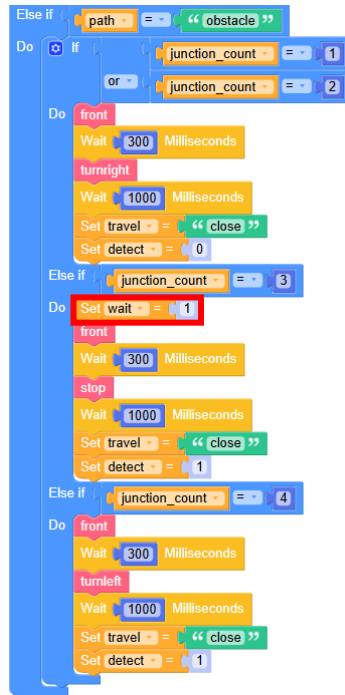
① is a junction. Thus we can decide what to do at the junction by adding a new condition:



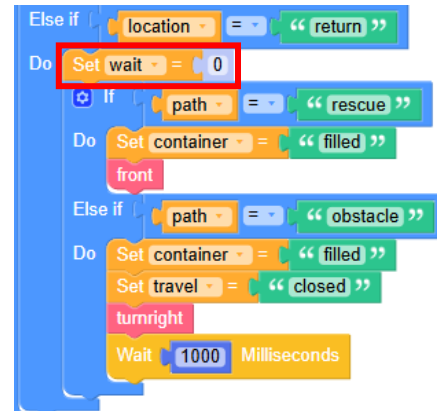
# ● Pause Sequence



A wait at point ① is required during the **rescue** route

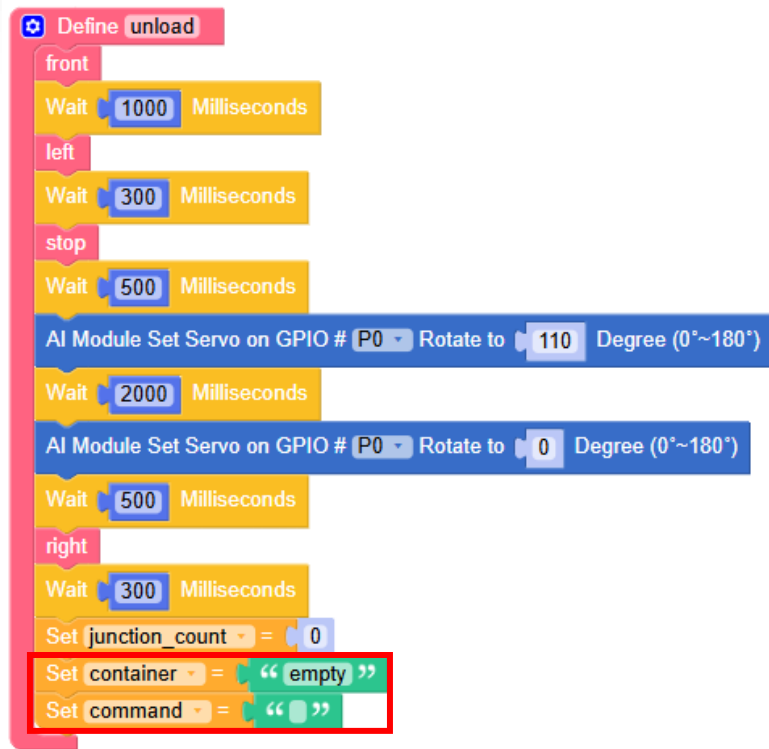


A wait at point ① is required during the **obstacle** route



End the wait and run the return sequence after detecting **D** on the mechanical robot.

## Unloading



The current task is considered complete when the transport vehicle unloads its goods.

- Set status of **container** to **empty**
- Clear the current **command**

# Reference Program (Main Code)

```

Camera Setup
Set Image Capturing Size: [QVGA (320*240)]
Set Image Capturing Color Mode: [Colorful]

LCD Screen Setup
Set LCD Screen Rotation to: [0]

AI Module Motor Driver Setup
AI Module Set Motor [0] % Speed to [0] (0-255) Rotating [Clockwise]
AI Module Set Motor [1] % Speed to [0] (0-255) Rotating [Clockwise]
AI Module Set Motor [2] % Speed to [0] (0-255) Rotating [Clockwise]
AI Module Set Motor [3] % Speed to [0] (0-255) Rotating [Clockwise]

AI Module Servo Setup
AI Module Set Servo on GPIO # [20] Rotate to [0] Degree (0~180°)
AI Module Set Servo on GPIO # [21] Rotate to [95] Degree (0~180°)

Initialize A.I. Model
Load model file from path [/root/preset/model/cocopi_FloodingMission]
Define class name by list:
[0] Use these values to create an array of Numbers:
["Supply_Station"]
["Rescue_Station"]
["Clearing_Zone"]
["A"]
["B"]
["C"]
["D"]

Set wait = [0]
Set travel = ["open"]
Set path = ["supply"]
Set junction_count = [0]
Set speed = [105]
Set location = ["A"]
Set old_result = [""]
Set container = ["empty"]
Set command = [""]
Set threshold = [150]
Set detect = [0]

```

```

Repeat forever
Do Detection
  If
    IoT Module Get GPIO # 35 Analog Value < threshold
    and
    IoT Module Get GPIO # 36 Analog Value < threshold
  Do
    Change junction_count by 1
    If path == "Supply"
      Do Supply
    Else If path == "Rescue"
      Do Rescue
    Else If path == "Obstacle"
      Do Obstacle
  Else If
    IoT Module Get GPIO # 35 Analog Value < threshold
    or
    IoT Module Get GPIO # 36 Analog Value < threshold
    and
    wait = 0
  Do
    Set travel = "open"
  If travel == "open"
  Do
    If
      IoT Module Get GPIO # 35 Analog Value > threshold
      and
      IoT Module Get GPIO # 36 Analog Value < threshold
    Do
      turnright
    Else If
      IoT Module Get GPIO # 35 Analog Value < threshold
      and
      IoT Module Get GPIO # 36 Analog Value > threshold
    Do
      turnleft
    Else
      front

```

# Reference Program (Routes)

```

Define Supply
If junction_count <= 1
Do front
Wait 300 Milliseconds
turnleft
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 0
Else if junction_count == 5
Do front
Wait 300 Milliseconds
turnright
Wait 1500 Milliseconds
Set travel = "close"
Set detect = 1
Else if junction_count <= 2 or junction_count == 4
Do front
Wait 300 Milliseconds
Else if junction_count == 3
Do stop
AI Module Set Servo on GPIO # P1 Rotate to 50 Degree (0~180)
Wait 300 Milliseconds
back
Wait 1500 Milliseconds
turnright
Wait 1500 Milliseconds
AI Module Set Servo on GPIO # P1 Rotate to 95 Degree (0~180)
Set travel = "close"
Set container = "Filled"

```

```

Define Rescue
If
or junction_count == 1
or junction_count == 2
or junction_count == 4
or junction_count == 5
Do front
Wait 300 Milliseconds
turnright
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 0
If junction_count == 5
Do Set detect = 1
Else if junction_count == 3
Do Set wait = 1
front
Wait 300 Milliseconds
stop
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 1

```

```

Define obstacle
If
or junction_count == 1
or junction_count == 2
Do front
Wait 300 Milliseconds
turnright
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 0
Else if junction_count == 3
Do Set wait = 1
front
Wait 300 Milliseconds
stop
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 1
Else if junction_count == 4
Do front
Wait 300 Milliseconds
turnleft
Wait 1000 Milliseconds
Set travel = "close"
Set detect = 1

```

# Reference Program (Object Detection)

```

Define detection
Set img_detection = Get the Image Captured from Camera
Set img_detection = Set Canvas img_detection Out to:
    Cutting Start Coordinate: X: 48 Y: 8
    Cutting Size: Width: 224 Height: 224

Load Preset A.I. Model: Object Recognition Model
Recognize from: img_detection

Do If Recognized Any Tested Object and detect == 1
Do For each item in list Tested Objects attribute
Do (object Object) On Canvas 1...
Do (object Object) On Canvas 1...
Do If From [1] Get the recognized tested Object's Name == "Supply_Station"
and From [1] Get the recognized tested Object's Y Coordinate of the bounding box <= 80
Do Set location = "supply_station"
Else if From [1] Get the recognized tested Object's Name == "Rescue_Station"
and From [1] Get the recognized tested Object's Y Coordinate of the bounding box <= 80
Do Set location = "rescue_station"
Else if From [1] Get the recognized tested Object's Name == "Clearing_Zone"
and From [1] Get the recognized tested Object's Y Coordinate of the bounding box <= 80
Do Set location = "clearing_zone"
Else if From [1] Get the recognized tested Object's Name == "D"
Do Set location = "return"
Set detect = 0
Else if From [1] Get the recognized tested Object's Name == "A"
Do Set command = "go_rescue"
Else if From [1] Get the recognized tested Object's Name == "B"
Do Set command = "go_clear"
Else if From [1] Get the recognized tested Object's Name == "C"
Do Set command = "go_collect"
Else Set location = "return"

```

```

Do If location == "supply_station" and path == "supply" and container == "filled"
Do If command == "go_rescue"
Do Set path = "rescue"
Else If command == "go_clear"
Do Set path = "obstacle"
Else If command == "go_collect"
Do Set path = "supply"
Do Wait 1000 Milliseconds
Set travel = "closed"
Set path = "supply"
Else If location == "rescue_station" and path == "rescue" and container == "filled"
Do If command == "go_clear"
Do Set path = "obstacle"
Else If command == "go_collect"
Do Set path = "supply"
Do Wait 1000 Milliseconds
Set travel = "closed"
Set path = "supply"
Else If location == "clearing_zone" and path == "obstacle" and container == "filled"
Do Set path = "rescue"
Do Set path = "obstacle"
Do Set container = "filled"
Do Set travel = "closed"
Do Wait 1000 Milliseconds
Set function_count = -1
Set location = "return"
Do Set wait = 0
Do If path == "rescue"
Do Set container = "filled"
Do Set travel = "closed"
Do Wait 1000 Milliseconds
Set function_count = -1
Set location = "return"
Set Canvas img_detection = Starting Coordinate as: X: 48 Y: 8
Show Canvas img_detection

```

```

Define unload
front
Wait 1000 Milliseconds
left
Wait 300 Milliseconds
stop
Wait 500 Milliseconds
AI Module Set Servo on GPIO # P0 - Rotate to 110 Degree (0°~180°)
Wait 2000 Milliseconds
AI Module Set Servo on GPIO # P0 - Rotate to 0 Degree (0°~180°)
Wait 500 Milliseconds
right
Wait 300 Milliseconds
Set function_count = 0
Set container = "empty"
Set command = "return"

```



# Reference Program (Movement)

## Define front

AI Module Set Motor C-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor D-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor E-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor F-'s Speed to speed (0~255) Rotating Anti-Clockwise

## Define back

AI Module Set Motor C-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor D-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor E-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor F-'s Speed to speed (0~255) Rotating Clockwise

## Define right

AI Module Set Motor C-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor D-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor E-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor F-'s Speed to speed (0~255) Rotating Anti-Clockwise

## Define left

AI Module Set Motor C-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor D-'s Speed to speed (0~255) Rotating Clockwise  
AI Module Set Motor E-'s Speed to speed (0~255) Rotating Anti-Clockwise  
AI Module Set Motor F-'s Speed to speed (0~255) Rotating Clockwise

## Define stop

AI Module Set Motor C-'s Speed to 0 (0~255) Rotating Clockwise  
AI Module Set Motor D-'s Speed to 0 (0~255) Rotating Clockwise  
AI Module Set Motor E-'s Speed to 0 (0~255) Rotating Clockwise  
AI Module Set Motor F-'s Speed to 0 (0~255) Rotating Clockwise

## Define turnleft

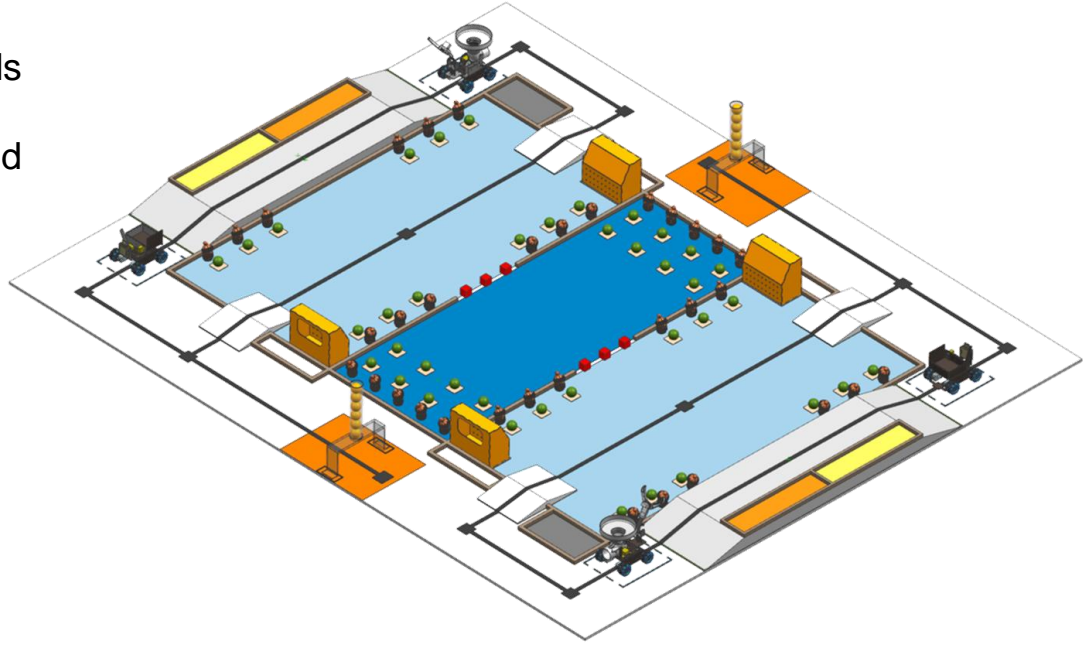
AI Module Set Motor C-'s Speed to speed 25 (0~255) Rotating Anti-Clockwise  
AI Module Set Motor D-'s Speed to speed 30 (0~255) Rotating Anti-Clockwise  
AI Module Set Motor E-'s Speed to speed 30 (0~255) Rotating Clockwise  
AI Module Set Motor F-'s Speed to speed 25 (0~255) Rotating Clockwise

## Define turnright

AI Module Set Motor C-'s Speed to speed 25 (0~255) Rotating Clockwise  
AI Module Set Motor D-'s Speed to speed 30 (0~255) Rotating Clockwise  
AI Module Set Motor E-'s Speed to speed 30 (0~255) Rotating Anti-Clockwise  
AI Module Set Motor F-'s Speed to speed 25 (0~255) Rotating Anti-Clockwise

## ● Test Your Program

- Upload your code onto the robot
- Run it on the playfield
- Observe the program in action
- Note down changes that needs to be made.
- Make necessary changes and repeat



P O W



THANK YOU

cocorobo

P

J U S T L E A V E P R E S E N T A T I O N T O O R I

