COCOROBO

CocoPi

AI for STEM Competition

Asia-Pacific STEAM_AI Robotics Challenge

# Robot Integration (Primary)

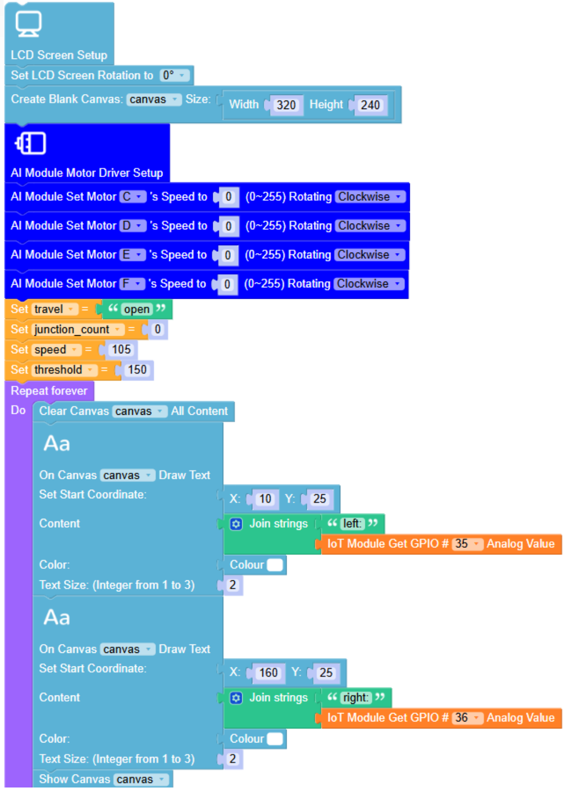
## Integration Process

# Routes

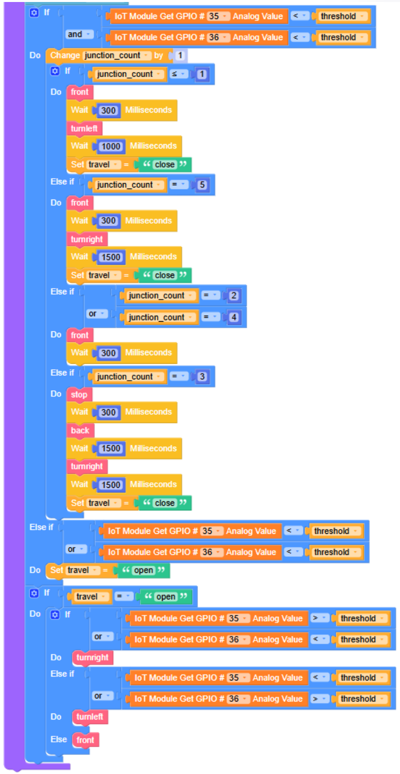
1

## Supply

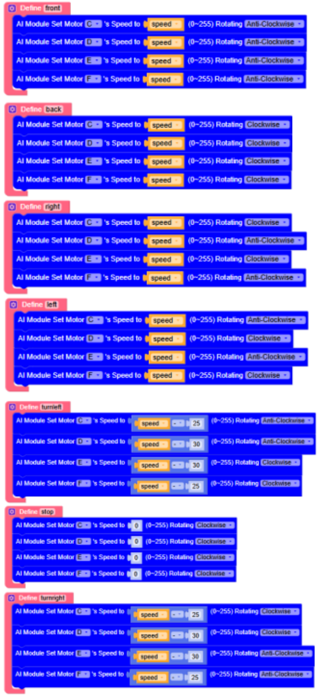
Retrieve, Transport and Deliver Supplies



```
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 = "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: Colour
    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: Colour
    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
    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
```

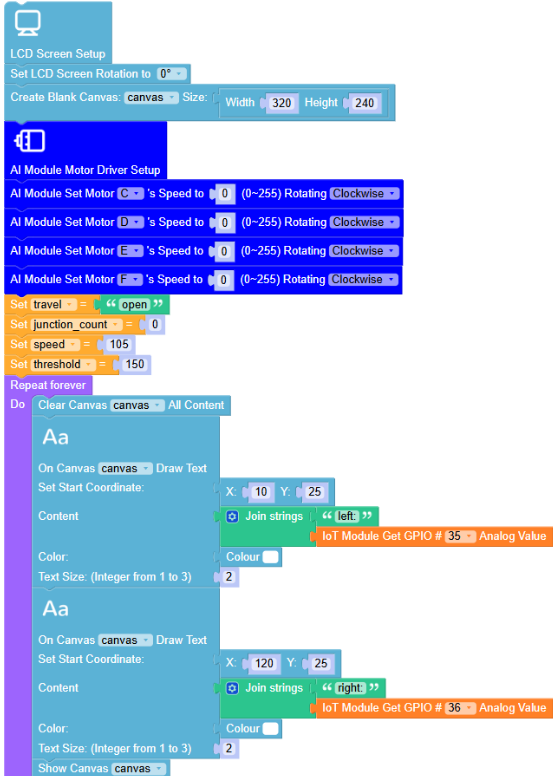


```
Define front
    AI Module Set Motor C 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor D 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor E 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor F 's Speed to speed (0~255) Rotating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

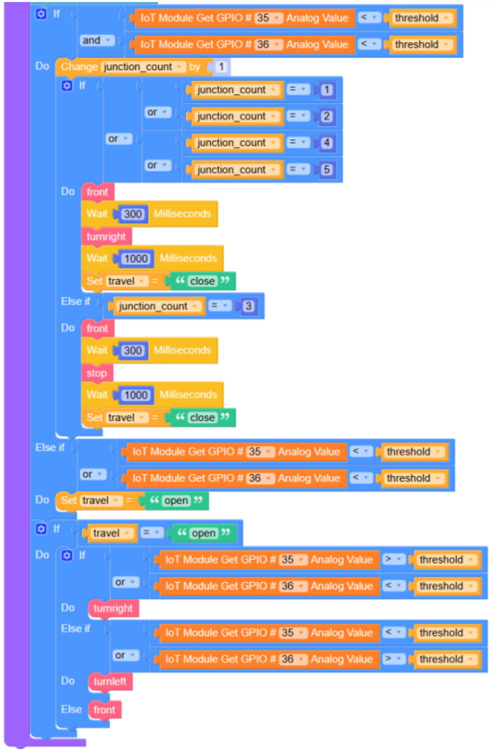
# Routes

## 2 Rescue

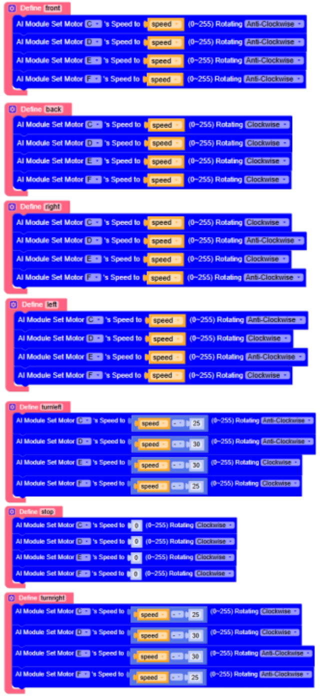
Pickup and transport injured personnels to the Aid Station



```
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:  Colour
    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:  Colour
    Text Size: (Integer from 1 to 3)  2
    Show Canvas canvas
```



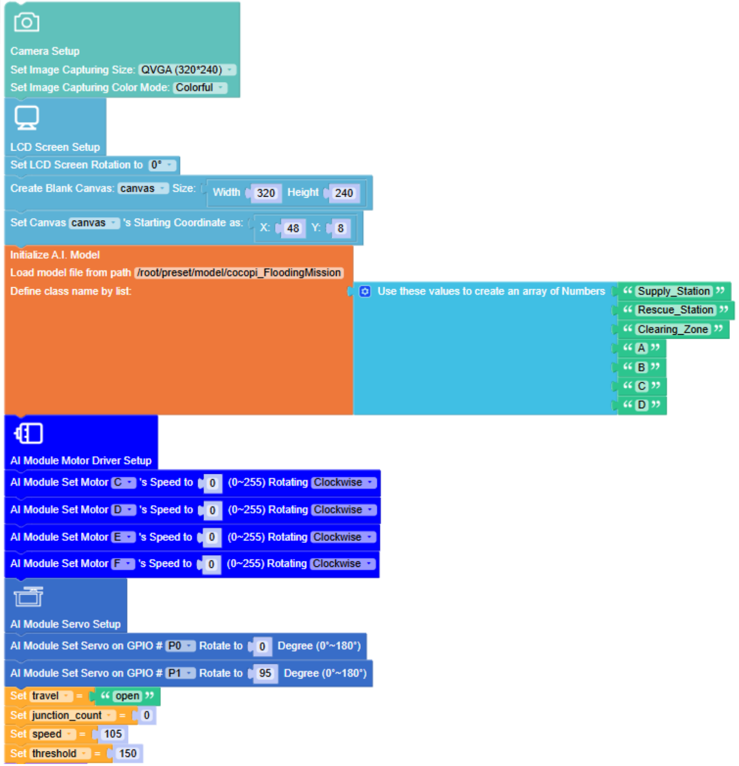
```
If      IoT Module Get GPIO # 35 Analog Value < threshold
    and IoT Module Get GPIO # 36 Analog Value < threshold
Do  Change junction_count by 1
    If      junction_count = 1
         or junction_count = 2
      or    junction_count = 4
         or junction_count = 5
    Do  front
        Wait 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
```



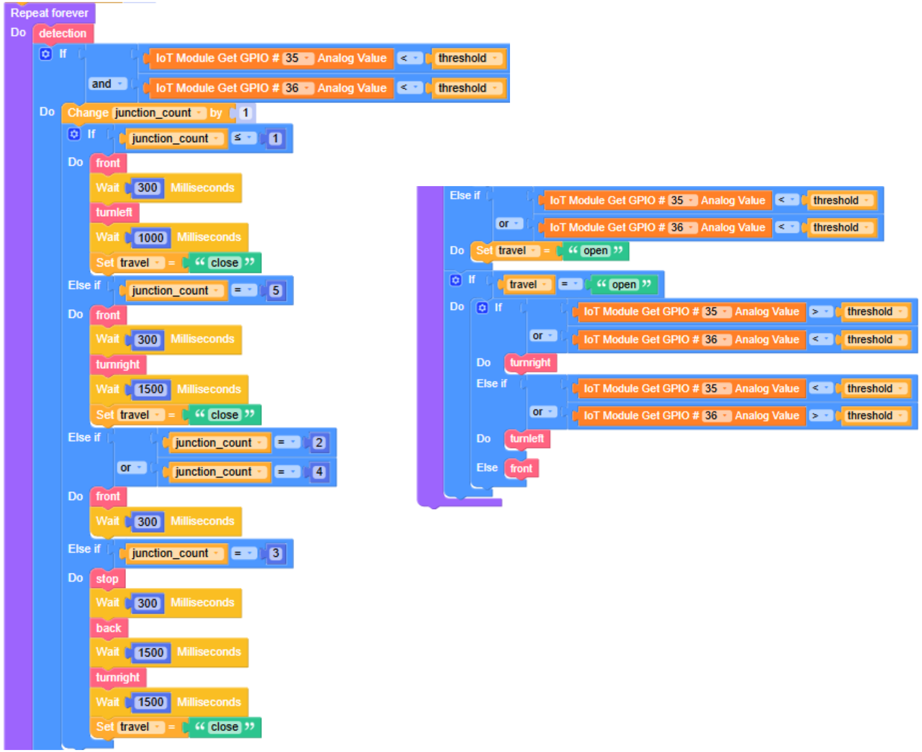
```
Define front
    AI Module Set Motor C 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor D 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor E 's Speed to speed (0~255) Rotating Anti-Clockwise
    AI Module Set Motor F 's Speed to speed (0~255) Rotating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

# Image Identification

*Main code (See next page for functions)



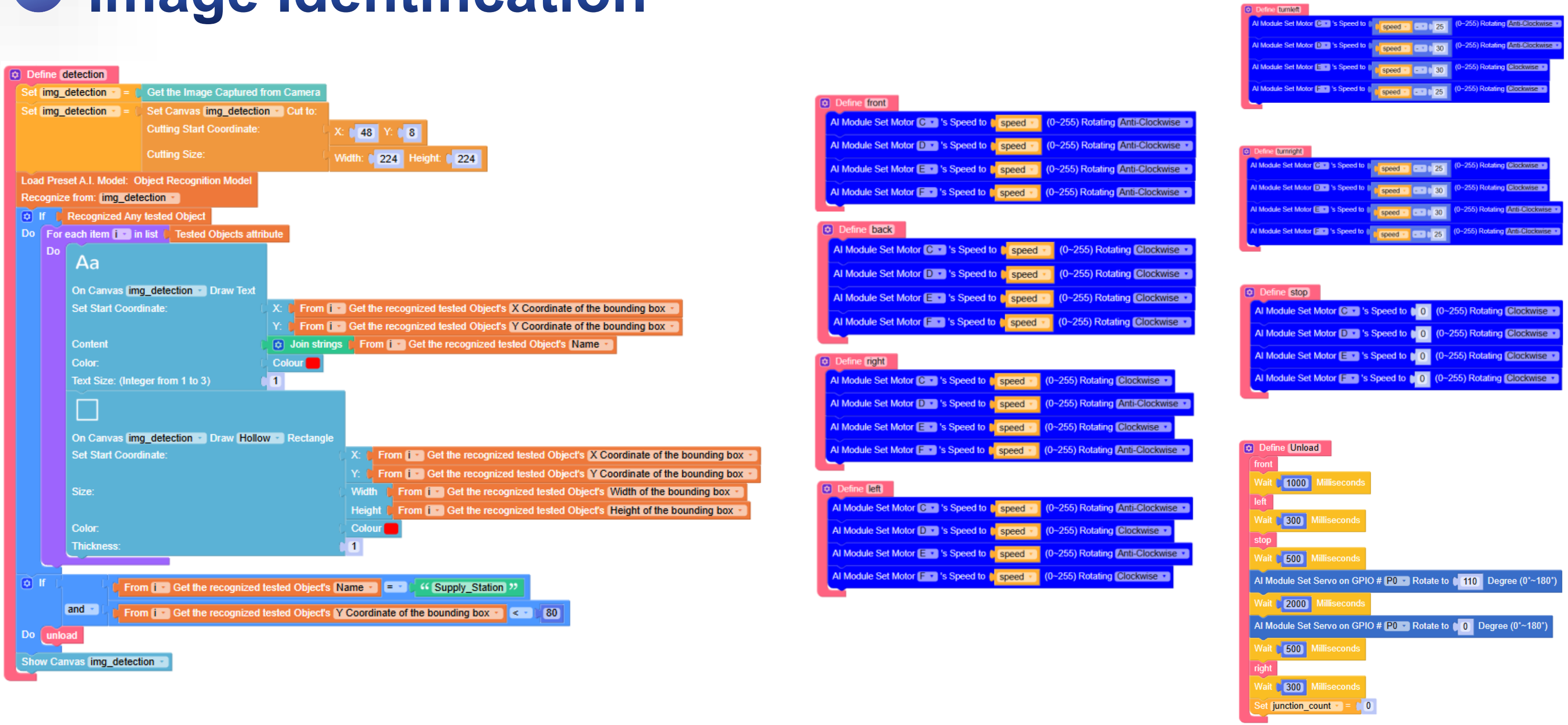
```
Camera Setup
Set Image Capturing Size: QVGA (320*240)
Set Image Capturing Color Mode: Colorful
LCD Screen Setup
Set LCD Screen Rotation to 0°
Create Blank Canvas: canvas Size: Width 320 Height 240
Set Canvas canvas 's Starting Coordinate as: X: 48 Y: 8
Initialize A.I. Model
Load model file from path /root/preset/model/cocopi_FloodingMission
Define class name by list: Use these values to create an array of Numbers
    "Supply_Station"
    "Rescue_Station"
    "Clearing_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)

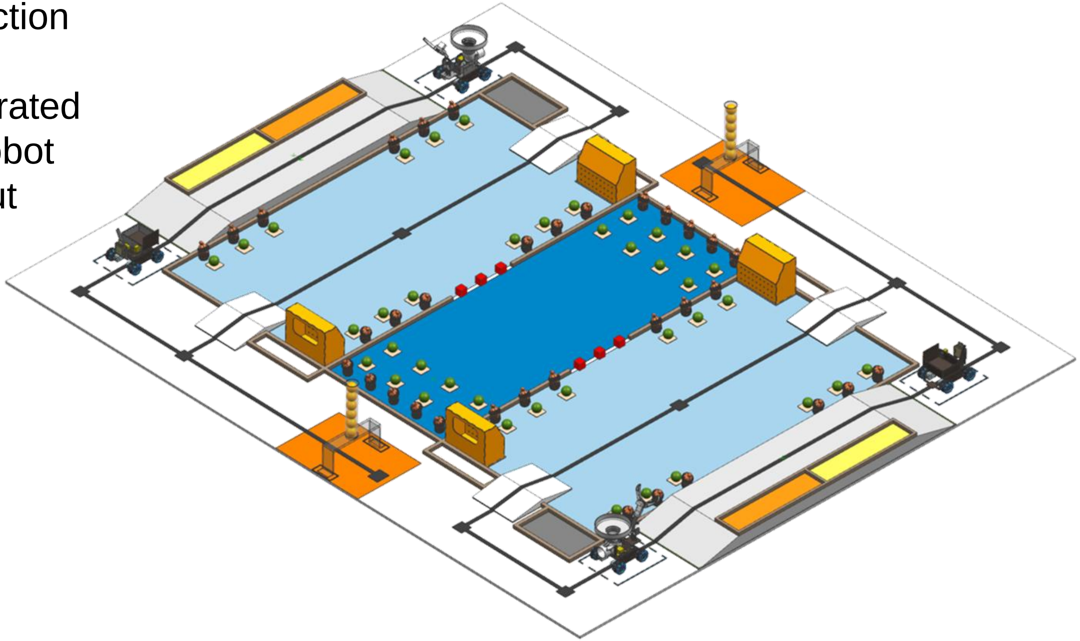
# Image Identification



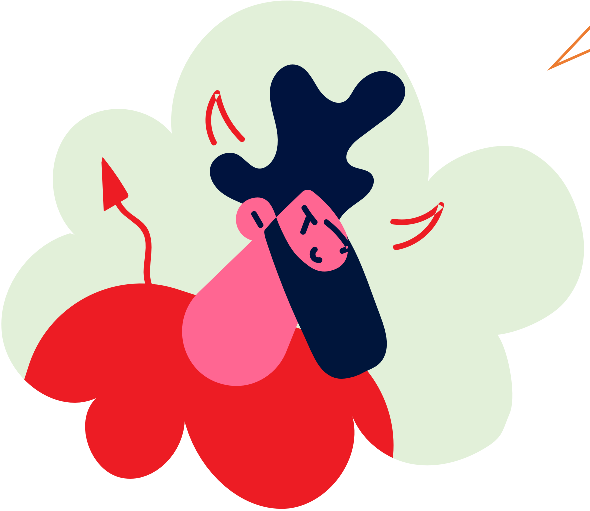
```
Define detection
  Set img_detection = Get the Image Captured from Camera
  Set img_detection = Set Canvas img_detection Cut to:
                      Cutting Start Coordinate:  X: 48  Y: 8
                      Cutting Size:  Width: 224  Height: 224
  Load Preset A.I. Model: Object Recognition Model
  Recognize from: img_detection
  If Recognized Any tested Object
  Do For each item i in list Tested Objects attribute
     Do Aa
        On Canvas img_detection Draw Text
        Set Start Coordinate:  X: From i Get the recognized tested Object's X Coordinate of the bounding box
                               Y: From i Get the recognized tested Object's Y Coordinate of the bounding box
        Content  Join strings  From i Get the recognized tested Object's Name
        Color:  Colour
        Text Size: (Integer from 1 to 3)  1
        □
        On Canvas img_detection Draw Hollow Rectangle
        Set Start Coordinate:  X: From i Get the recognized tested Object's X Coordinate of the bounding box
                               Y: From i Get the recognized tested Object's Y Coordinate of the bounding box
        Size:  Width  From i Get the recognized tested Object's Width of the bounding box
               Height  From i Get the recognized tested Object's Height of the bounding box
        Color:  Colour
        Thickness:  1
  If  From i Get the recognized tested Object's Name = "Supply_Station"
      and  From i Get the recognized tested Object's Y Coordinate of the bounding box < 80
  Do unload
  Show Canvas img_detection

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 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

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 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 junction_count = 0
```

# 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 ?

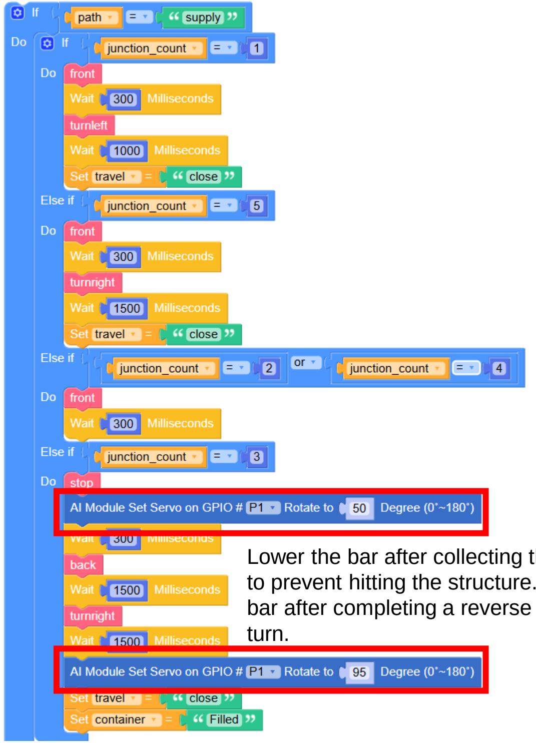
# Path Decision Making

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

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

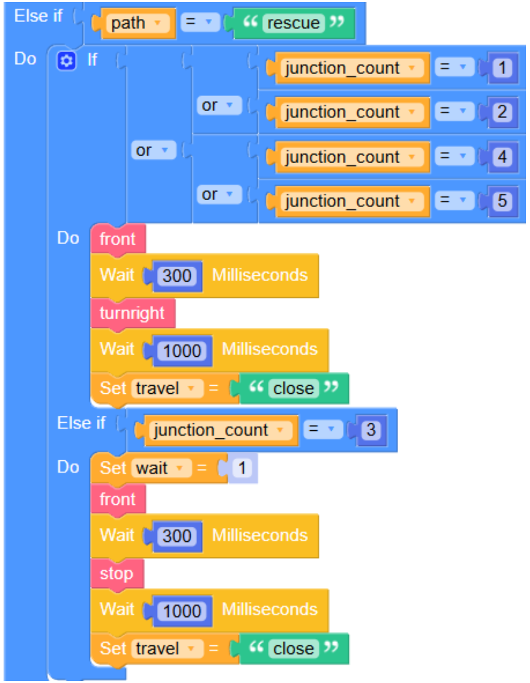
*The initial state for **Path** to be set as **Supply**

# Path Decision Making



```
If  path = "supply"
Do  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
        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"
```

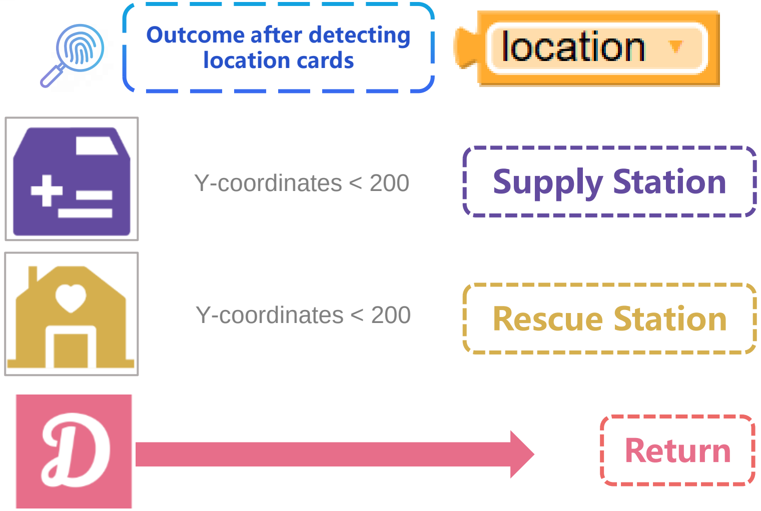
Lower the bar after collecting the supplies to prevent hitting the structure. Raise the bar after completing a reverse and right turn.



```
Else if  path = "rescue"
Do  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  Set wait = 1
        front
        Wait 300 Milliseconds
        stop
        Wait 1000 Milliseconds
        Set travel = "close"
```

# Decision Making

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



Outcome after detecting location cards

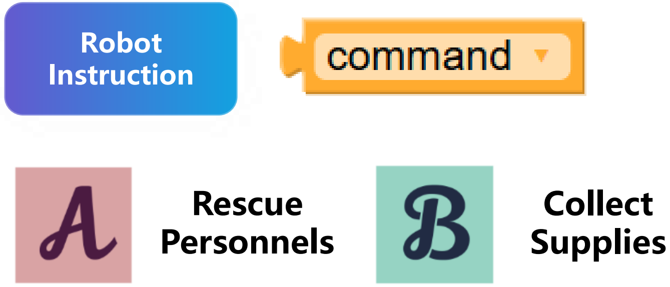
location

Y-coordinates < 200 — Supply Station

Y-coordinates < 200 — Rescue Station

Return

**location** represents that the robot is at the location



Robot Instruction

command

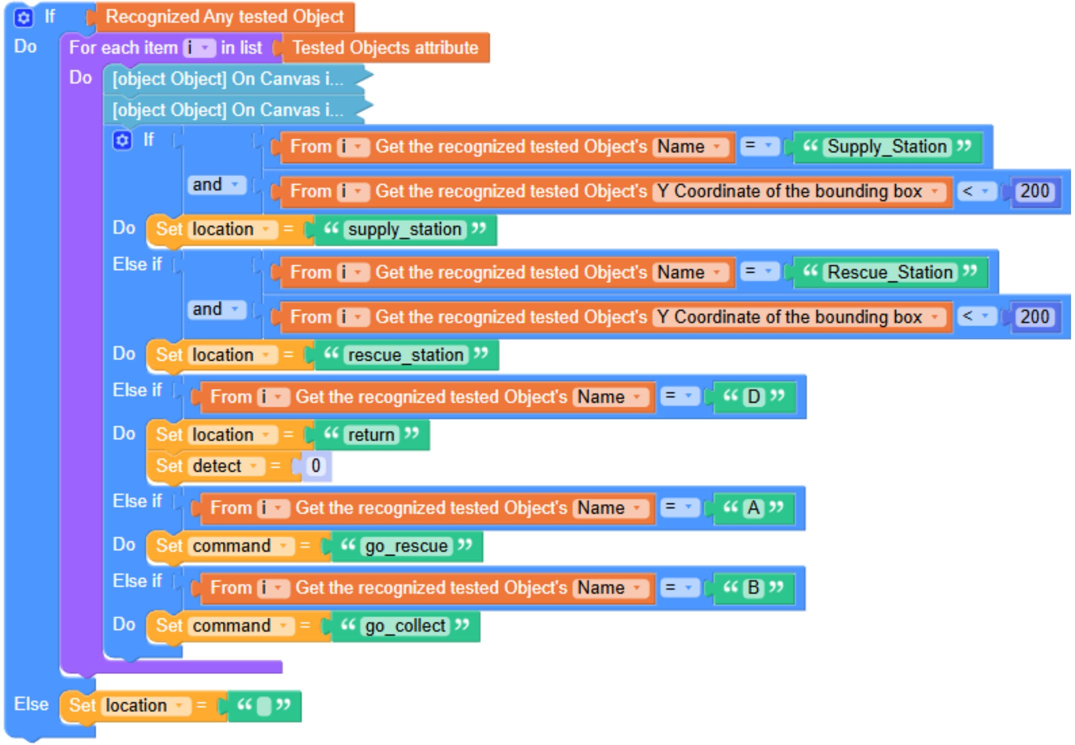
Rescue Personnels

Collect Supplies

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

# Decision Making

Detection
Outcome Code

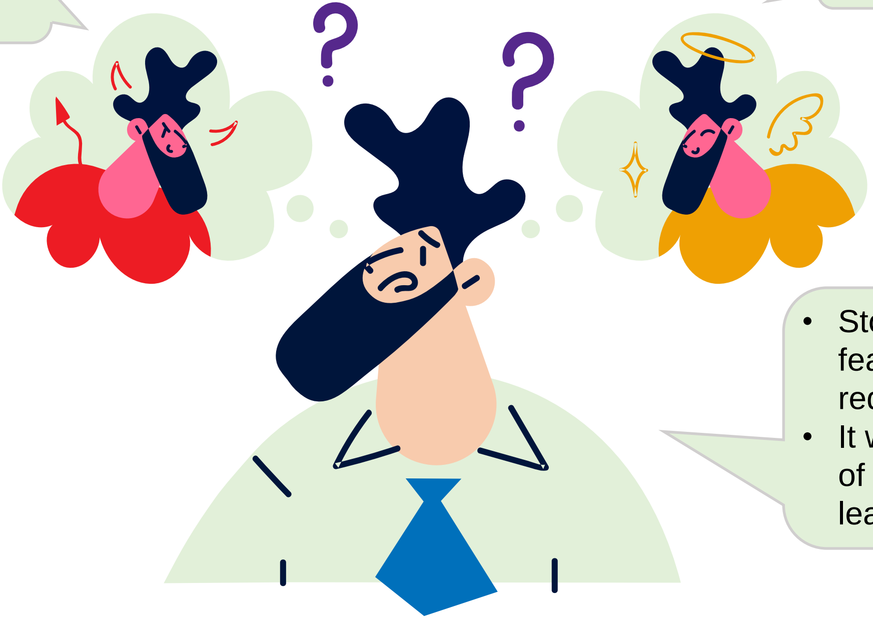


```
If  Recognized Any tested Object
Do  For each item i in list  Tested Objects attribute
    Do  [object Object] On Canvas i...
        [object Object] On Canvas i...
        If  From i Get the recognized tested Object's Name = “ Supply_Station ”
            and  From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
        Do  Set location = “ supply_station ”
        Else if  From i Get the recognized tested Object's Name = “ Rescue_Station ”
            and  From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
        Do  Set location = “ rescue_station ”
        Else if  From i Get the recognized tested Object's Name = “ D ”
        Do  Set location = “ return ”
            Set detect = 0
        Else if  From i Get the recognized tested Object's Name = “ A ”
        Do  Set command = “ go_rescue ”
        Else if  From i Get the recognized tested Object's Name = “ B ”
        Do  Set command = “ go_collect ”
Else  Set location = “ ”
```

# Object Detection

Does the robot have to constantly do object detection?

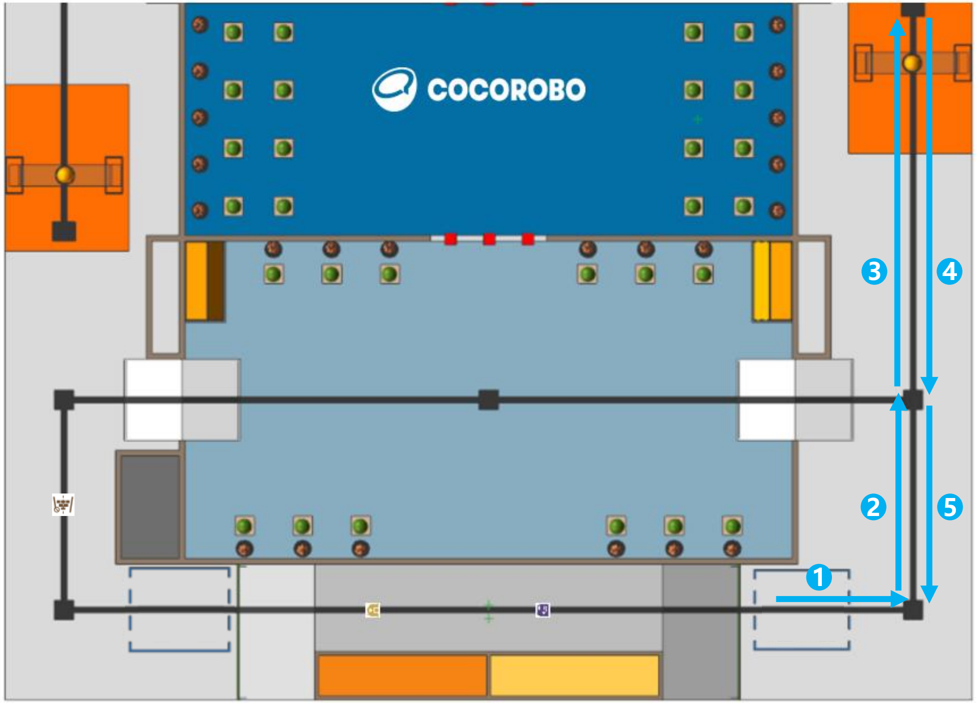
Which path does not require object detection?

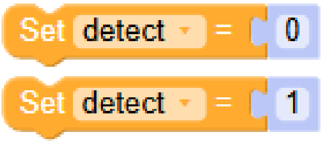


- Stop the object detection feature when it is not required
- It will reduce the chances of false detection which leads to errors.

# Analysis

When is the object detection feature needed during the “Supply” route.

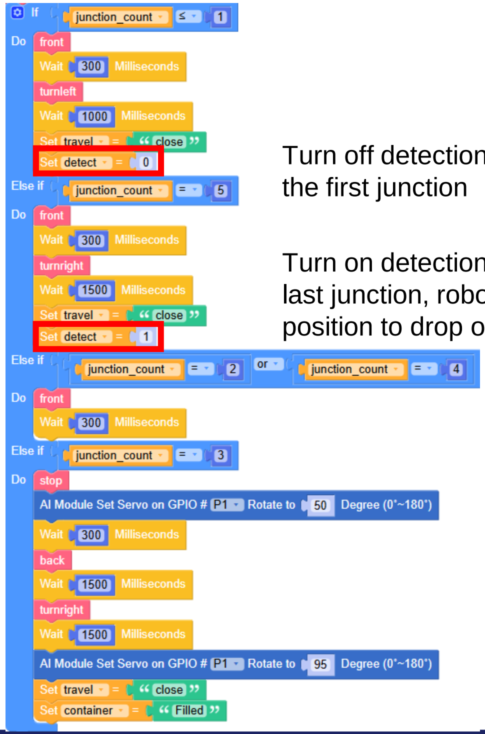




```
Set detect = 0
Set detect = 1
```

Detection “**Off**”

Detection “**On**”



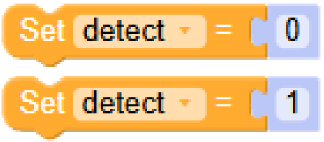
```
If junction_count ≤ 1
Do  front
    Wait 300 Milliseconds
    turnleft
    Wait 1000 Milliseconds
    Set travel = “close”
    Set detect = 0
Else if junction_count = 5
Do  front
    Wait 300 Milliseconds
    turnright
    Wait 1500 Milliseconds
    Set travel = “close”
    Set detect = 1
Else if 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

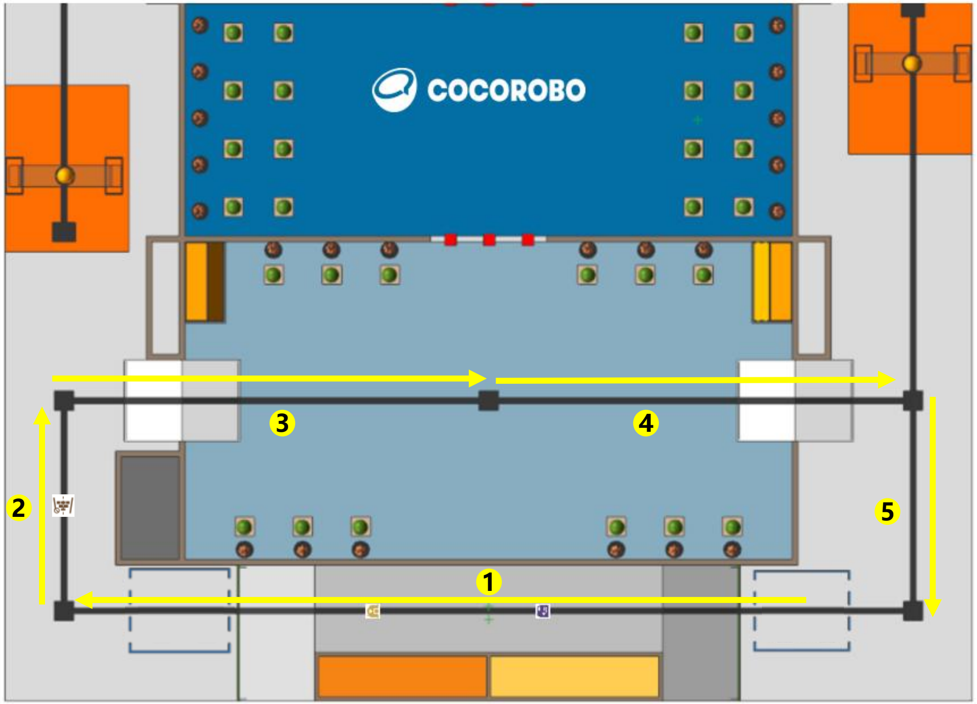
When is the object detection feature needed during the “Rescue” route.

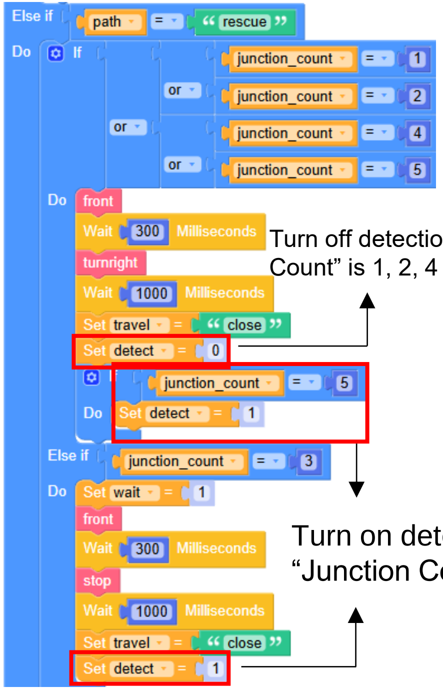


```
Set detect = 0
Set detect = 1
```

Detection “**Off**”

Detection “**On**”





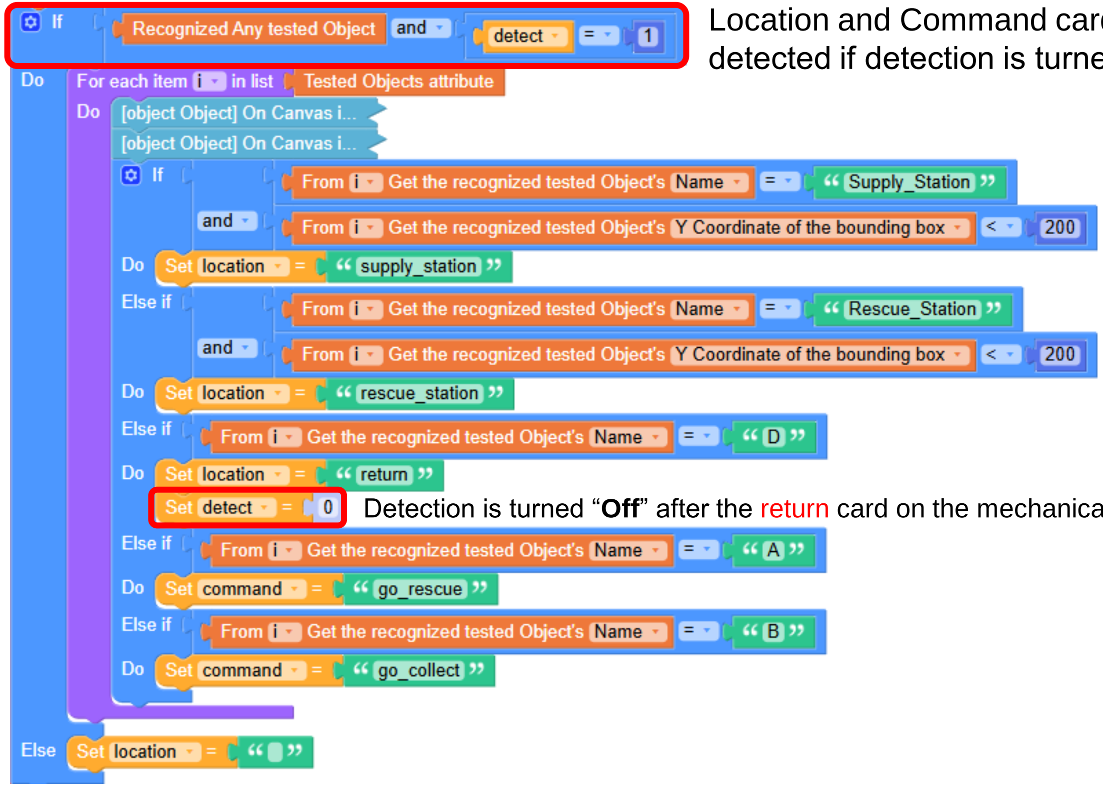
```
Else if  path = “rescue”
Do  If  junction_count = 1  or  junction_count = 2  or  junction_count = 4  or  junction_count = 5
    Do  front
        Wait 300 Milliseconds
        turnright
        Wait 1000 Milliseconds
        Set travel = “close”
        Set detect = 0
        if  junction_count = 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
```

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

Turn on detection when “Junction Count” = 5 or = 3

# Detection (On/Off)

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



```
If Recognized Any tested Object and detect = 1
Do  For each item i in list Tested Objects attribute
    Do  [object Object] On Canvas i...
        [object Object] On Canvas i...
        If      From i Get the recognized tested Object's Name = “Supply_Station”
                and From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
        Do      Set location = “supply_station”
        Else if From i Get the recognized tested Object's Name = “Rescue_Station”
                and From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
        Do      Set location = “rescue_station”
        Else if From i Get the recognized tested Object's Name = “D”
        Do      Set location = “return”
                Set detect = 0
        Else if From i Get the recognized tested Object's Name = “A”
        Do      Set command = “go_rescue”
        Else if From i Get the recognized tested Object's Name = “B”
        Do      Set command = “go_collect”
Else Set location = “”
```

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)
- 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**.

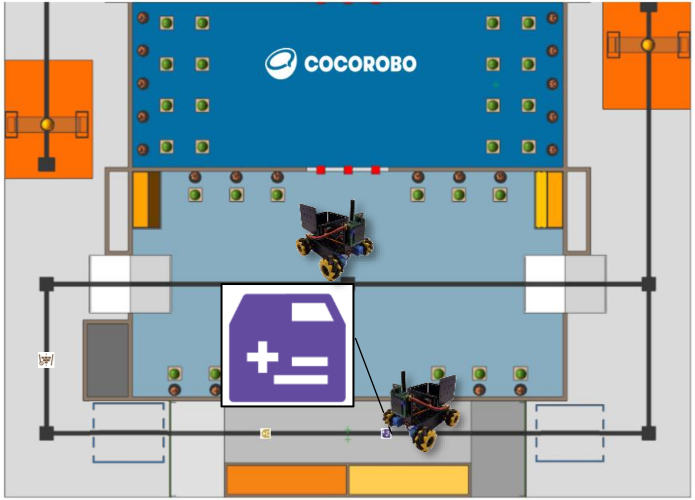
# Decision Making

Location set as Supply Station & Path = “Supply” & Container = “Filled”

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

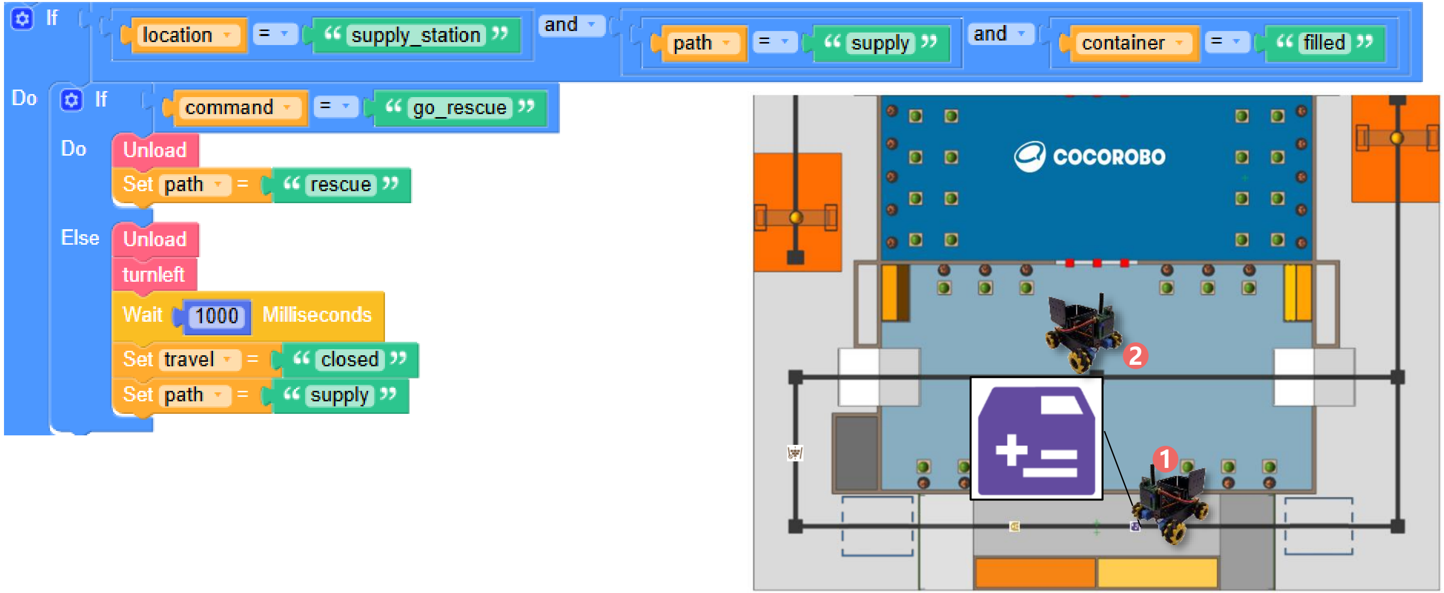
A → Rescue Personnel → Path = “Rescue”

If the command received is not **A,** 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”



```
If  location = “supply_station” and path = “supply” and container = “filled”
Do  If  command = “go_rescue”
    Do   Unload
         Set path = “rescue”
    Else Unload
         turnleft
         Wait 1000 Milliseconds
         Set travel = “closed”
         Set path = “supply”
```

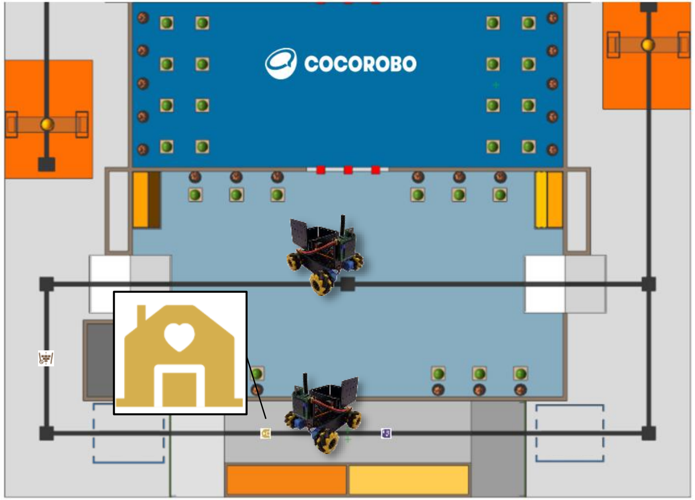
# Decision Making

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

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

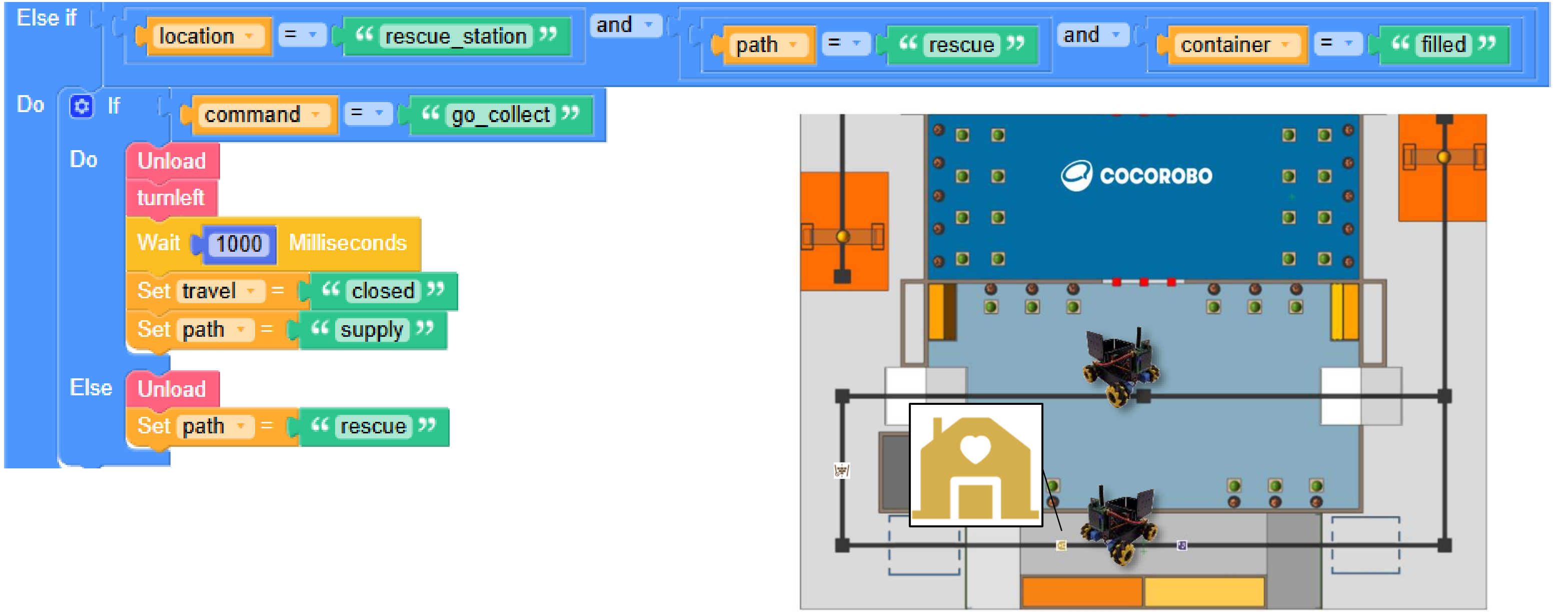
B Collect Supplies → Path = "Supply"

If the command received is not **B,** the robot will continue rescuing personnels. It will continue down the rescue path.



# Decision Making

Location set as Rescue Station & Path = “Rescue” & Container = “Filled”

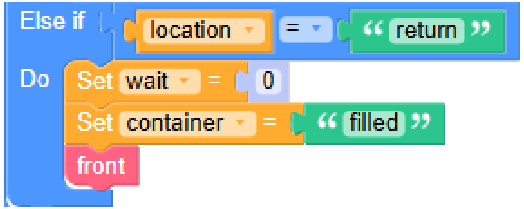


```
Else if  location = "rescue_station"  and  path = "rescue"  and  container = "filled"
Do   If  command = "go_collect"
     Do    Unload
           turnleft
           Wait 1000 Milliseconds
           Set travel = "closed"
           Set path = "supply"
     Else  Unload
           Set path = "rescue"
```

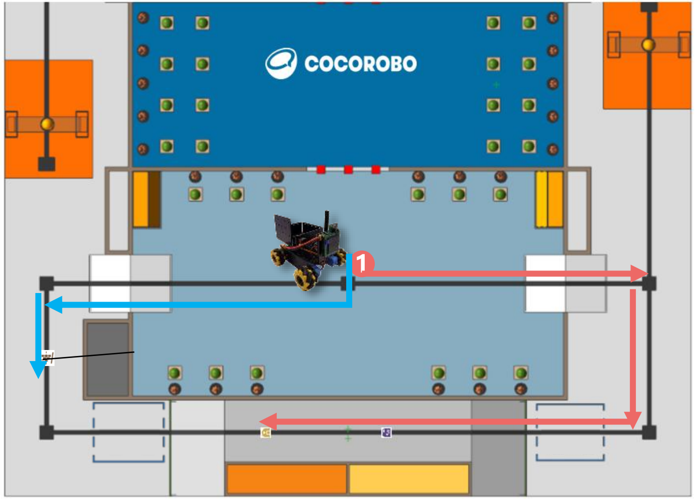
# Decision Making

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

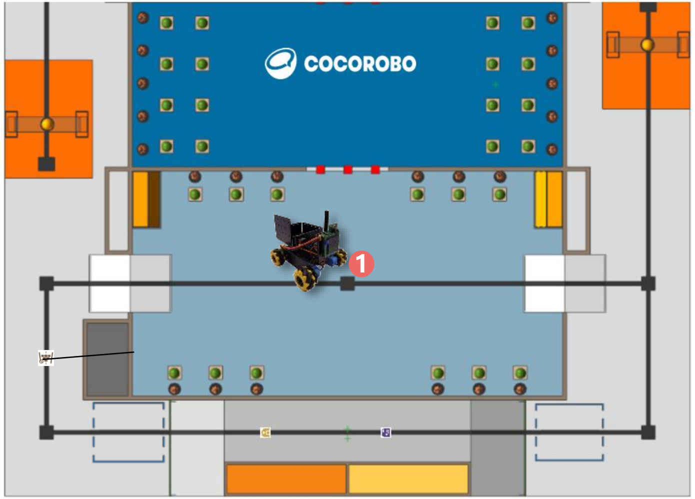
Location set as Return, it will operate based on the **path** set



```
Else if  location = "return"
Do  Set wait = 0
    Set container = "filled"
    front
```



# Pause Sequence Analysis

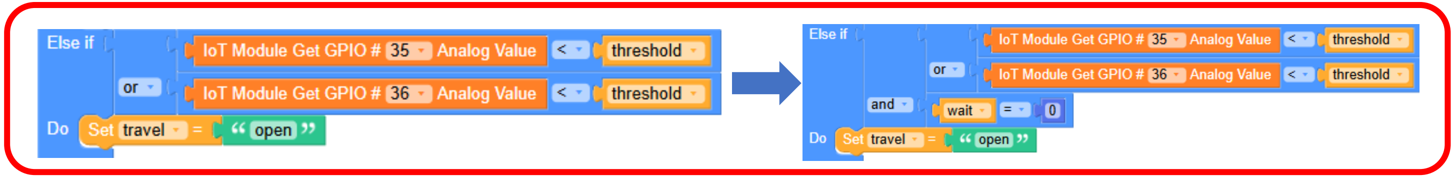


After the transport robot arrives at point ❶, it waits there until it detects the D instruction from the mechanical robot. When the D instruction is received, it performs the return sequence

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;

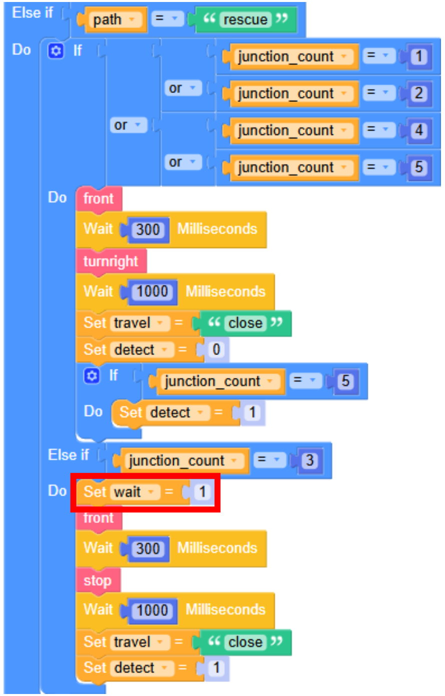
❶ is a junction. Thus, we can decide what to do at the junction by adding a new condition:



```
Else if    IoT Module Get GPIO # 35 Analog Value < threshold
        or IoT Module Get GPIO # 36 Analog Value < threshold
Do  Set travel = " open "
```

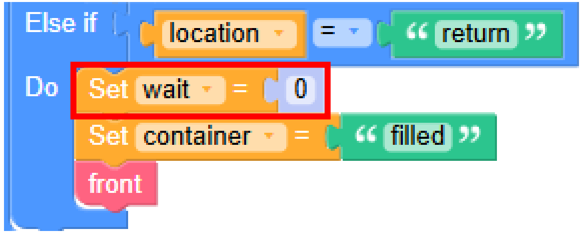
```
Else if        IoT Module Get GPIO # 35 Analog Value < threshold
            or IoT Module Get GPIO # 36 Analog Value < threshold
        and wait = 0
Do  Set travel = " open "
```

# Pause Sequence



```
Else if  path = “ rescue ”
Do  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 ”
        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
```

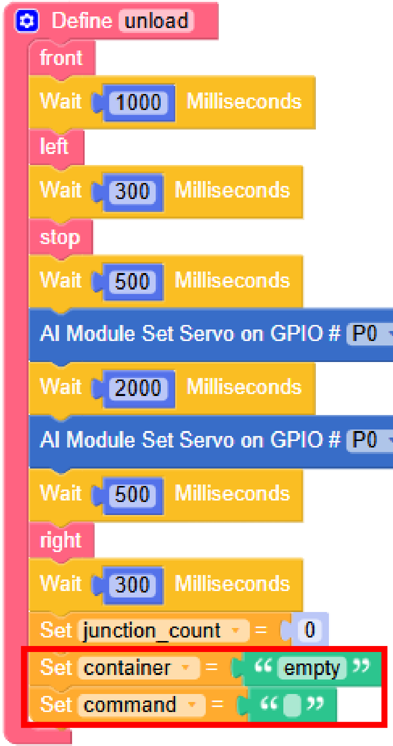
A wait at point ❶ is required during the **rescue** route



```
Else if  location = “ return ”
Do  Set wait = 0
    Set container = “ filled ”
    front
```

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

# Unloading

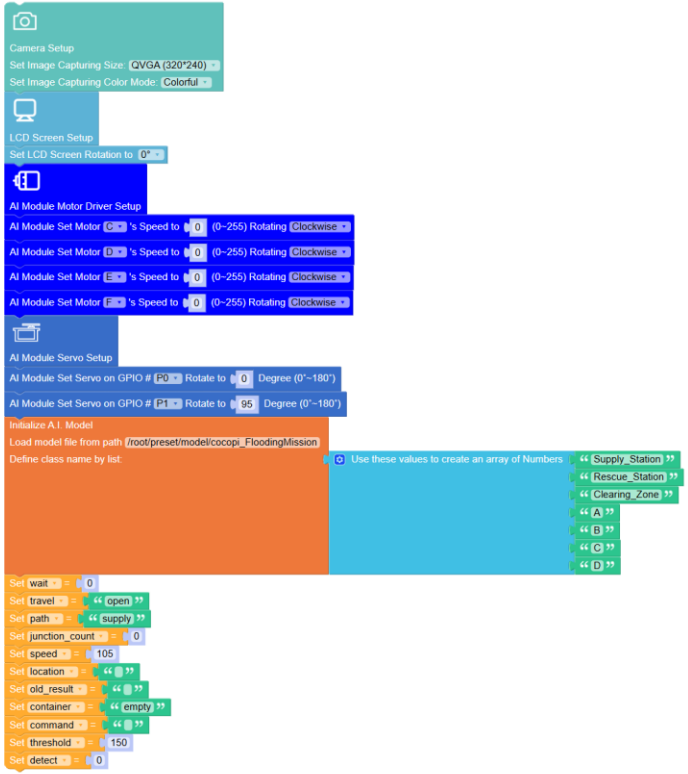


```
Define unload
  front
  Wait 1000 Milliseconds
  left
  Wait 300 Milliseconds
  stop
  Wait 500 Milliseconds
  AI Module Set Servo on GPIO # P0 Rotate to 110 Degree (0°~180°)
  Wait 2000 Milliseconds
  AI Module Set Servo on GPIO # P0 Rotate to 0 Degree (0°~180°)
  Wait 500 Milliseconds
  right
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

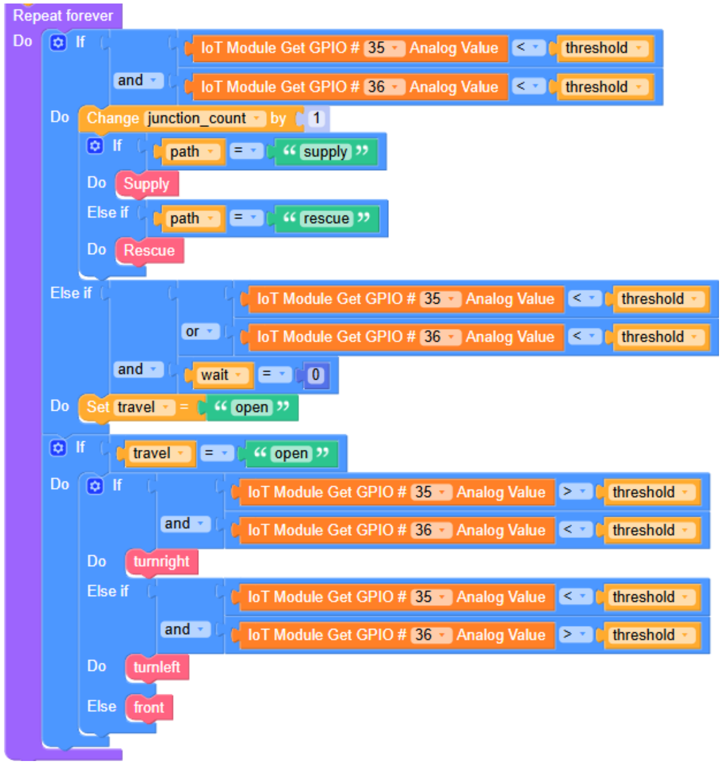
The current task is considered complete when the transport vehicle unloads its goods.

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

# Reference Program (Main Code)

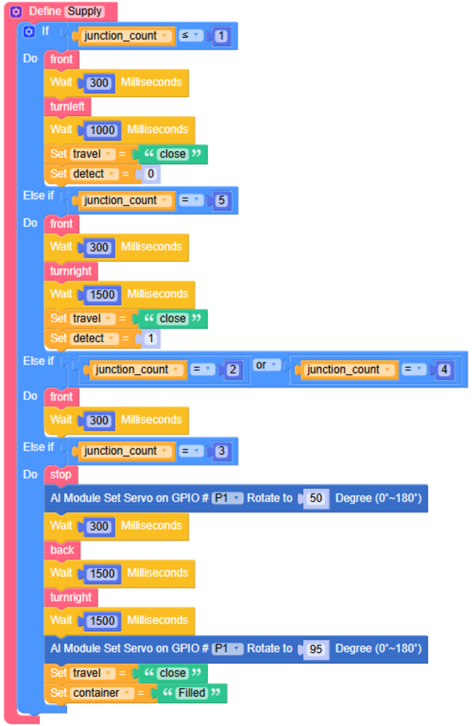


```
Camera Setup
Set Image Capturing Size: QVGA (320*240)
Set Image Capturing Color Mode: Colorful
LCD Screen Setup
Set LCD Screen Rotation to 0°
AI Module Motor Driver Setup
AI Module Set Motor C 's Speed to 0 (0~255) Rotating Clockwise
AI Module Set Motor D 's Speed to 0 (0~255) Rotating Clockwise
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°)
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"
    "Clearing_Zone"
    "A"
    "B"
    "C"
    "D"
Set wait = 0
Set travel = "open"
Set path = "supply"
Set junction_count = 0
Set speed = 105
Set location = " "
Set old_result = " "
Set container = "empty"
Set command = " "
Set threshold = 150
Set detect = 0
```

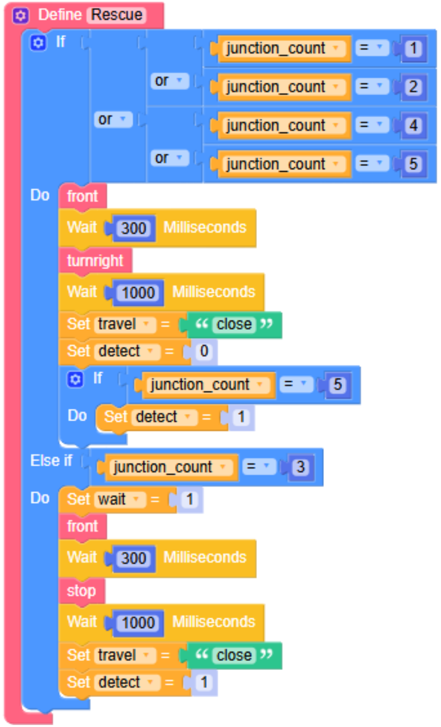


```
Repeat forever
Do  If  IoT Module Get GPIO # 35 Analog Value < threshold
        and IoT Module Get GPIO # 36 Analog Value < threshold
    Do  Change junction_count by 1
        If  path = "supply"
        Do  Supply
        Else if  path = "rescue"
        Do  Rescue
    Else if  IoT Module Get 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
```

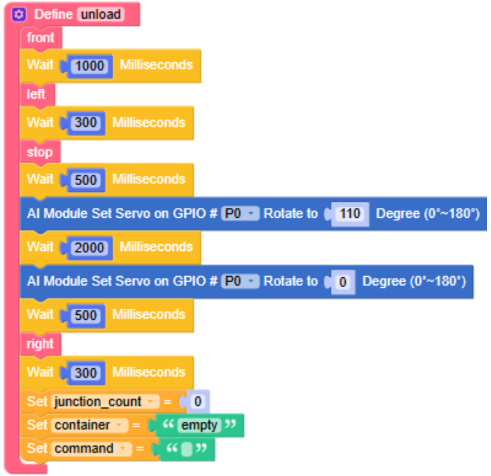
# Reference Program (Routes)



```
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"
```

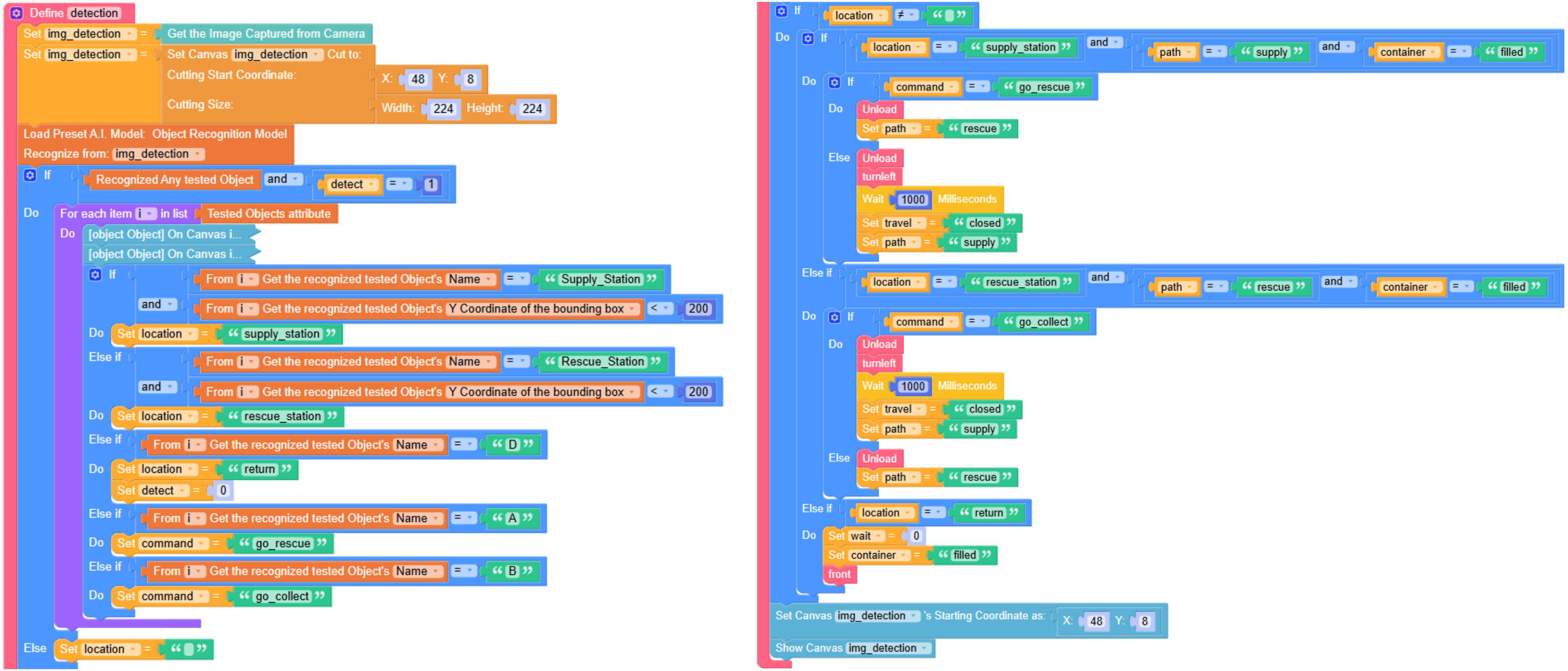


```
Define Rescue
  If junction_count = 1
       or junction_count = 2
     or
       junction_count = 4
       or junction_count = 5
  Do  front
      Wait 300 Milliseconds
      turnright
      Wait 1000 Milliseconds
      Set travel = "close"
      Set detect = 0
      If junction_count = 5
      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 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 junction_count = 0
  Set container = "empty"
  Set command = ""
```

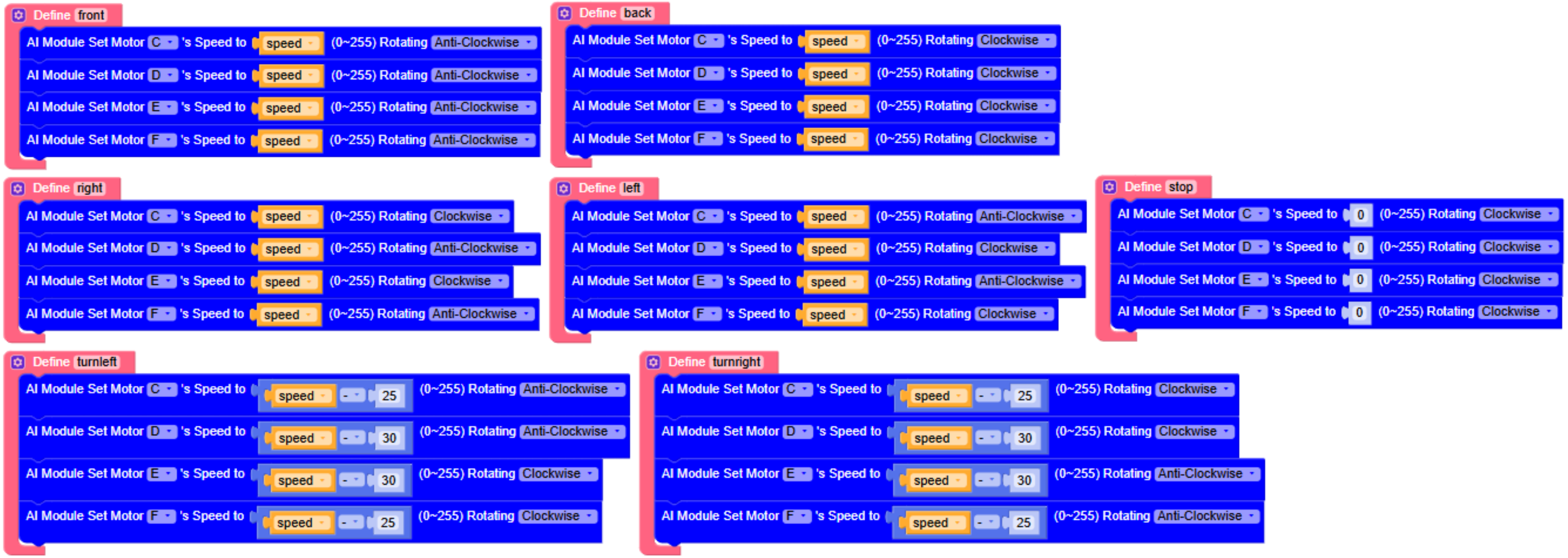
# Reference Program (Object Detection)



```
Define detection
    Set img_detection = Get the Image Captured from Camera
    Set img_detection = Set Canvas img_detection Cut 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
    If Recognized Any tested Object and detect = 1
    Do  For each item i in list Tested Objects attribute
        Do  [object Object] On Canvas i...
            [object Object] On Canvas i...
            If      From i Get the recognized tested Object's Name = “Supply_Station”
                    and From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
            Do      Set location = “supply_station”
            Else if From i Get the recognized tested Object's Name = “Rescue_Station”
                    and From i Get the recognized tested Object's Y Coordinate of the bounding box < 200
            Do      Set location = “rescue_station”
            Else if From i Get the recognized tested Object's Name = “D”
            Do      Set location = “return”
                    Set detect = 0
            Else if From i Get the recognized tested Object's Name = “A”
            Do      Set command = “go_rescue”
            Else if From i Get the recognized tested Object's Name = “B”
            Do      Set command = “go_collect”
    Else Set location = “ ”
```

```
If location ≠ “ ”
Do  If      location = “supply_station” and path = “supply” and container = “filled”
    Do      If command = “go_rescue”
            Do   Unload
                 Set path = “rescue”
            Else Unload
                 turnleft
                 Wait 1000 Milliseconds
                 Set travel = “closed”
                 Set path = “supply”
    Else if location = “rescue_station” and path = “rescue” and container = “filled”
    Do      If command = “go_collect”
            Do   Unload
                 turnleft
                 Wait 1000 Milliseconds
                 Set travel = “closed”
                 Set path = “supply”
            Else Unload
                 Set path = “rescue”
    Else if location = “return”
    Do      Set wait = 0
            Set container = “filled”
            front
Set Canvas img_detection 's Starting Coordinate as: X: 48 Y: 8
Show Canvas img_detection
```

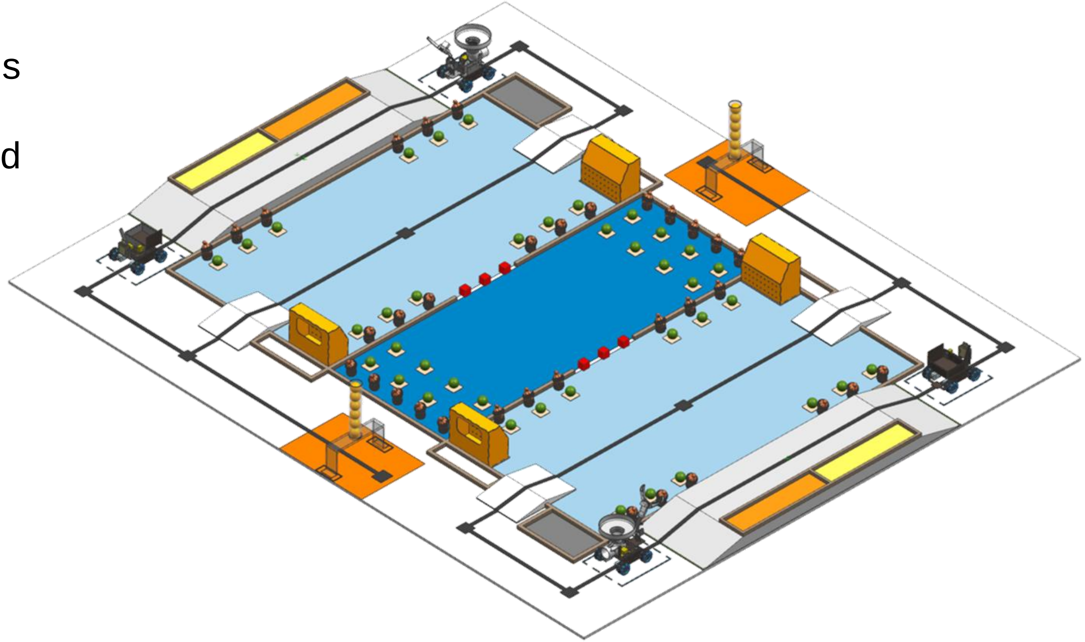
# Reference Program (Movement)



```
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

E

# THANK YOU

cocorobo

P

JUST LEAVE PRESENTATION TO ORI N