











#### Supply

Retrieve, Transport and Deliver Supplies

```
Set LCD Screen Rotation to 0°
Create Blank Canvas: canvas - Size:
                                 Width ( 320 Height ( 240
4∏
Al Module Motor Driver Setup
Al Module Set Motor C - 's Speed to 0 (0~255) Rotating Clockwise -
Al Module Set Motor D - 's Speed to 0 (0~255) Rotating Clockwise -
Al Module Set Motor E → 's Speed to 0 (0~255) Rotating Clockwise →
Al Module Set Motor F 's Speed to 0 (0~255) Rotating Clockwise
 et travel = [ " open "
   junction_count - = 0
   speed
   threshold
 Repeat forever
 Oo Clear Canvas canvas 🔻 All Conten
     Aa
    On Canvas canvas Draw Text
                                  X: 10 Y: 25
                                  Join strings | " (left:) "
                                                IoT Module Get GPIO # 35 - Analog Value
                                  Colour
    On Canvas canvas Draw Text
                                   X: 160 Y:
                                               25
                                  Join strings 44 (right:)
                                                IoT Module Get GPIO # 36 - Analog Value
                                  Colour
    Text Size: (Integer from 1 to 3)
    Show Canvas canvas
```

```
IoT Module Get GPIO # 35 · Analog Value ( threshold ·
and In In In In Indian Get GPIO # 36 Analog Value | < 1 | Threshold |
ange junction_count - by 1
     junction_count - ≤ - 1
    travel - = C " close >
   junction_count - = - | 5
     300 Millisecond
    1500
    travel - = ( " close) 2
         junction_count - = - 2
    or junction_count = 4
   junction_count - 3
     1500 Milliseco
      IoT Module Get GPIO # 35 - Analog Value  threshold -
OF TOT Module Get GPIO # (36 To Analog Value Threshold
             IoT Module Get GPIO # 35 · Analog Value    threshold
          IoT Module Get GPIO # 35 - Analog Value    threshold
           loT Module Get GPIO # 36 Analog Value  threshold
```

```
speed - - 1 30 (0-
speed - 25
```





#### Routes



#### Rescue

Pickup and transport injured personnels to the Aid Station

```
Set LCD Screen Rotation to 0°
Create Blank Canvas: canvas Size: Width 320 Height 240
4
Al Module Motor Driver Setup
Al Module Set Motor C ▼ 's Speed to 0 0 (0~255) Rotating Clockwise ▼
Al Module Set Motor D - 's Speed to 0 (0~255) Rotating Clockwise -
Al Module Set Motor E > 's Speed to 0 (0~255) Rotating Clockwise >
Al Module Set Motor F v 's Speed to 0 (0~255) Rotating Clockwise v
  travel = [ " open "
   junction count = 0
             105
   threshold
 Repeat forever
Do Clear Canvas canvas All Content
    Aa
    On Canvas canvas Draw Text
                                  X: 10 Y: 25
                                  Join strings ( 44 left:) >>
                                               IoT Module Get GPIO # 35 - Analog Value
                                  Colour
    On Canvas canvas Draw Text
                                  X: [ 120 Y: [ 25
                                  Join strings ( " right: "
                                               IoT Module Get GPIO # 36 • Analog Value
                                  Colour
    Show Canvas canvas
```

```
IoT Module Get GPIO # 35 v Analog Value < v threshold v
      IoT Module Get GPIO # 36 · Analog Value  threshold
ange (junction_count =
              junction_count · = · 1
        or v junction_count v = v 2
               junction_count = = 4
        or v junction_count v = v 5
    300 Millisecond:
    1000
           Close 2
   junction_count - = - 3
    300 Milliseconds
    1000
           Close 2
     oT Module Get GPIO # 36 
Analog Value 

threshold
        ( open "
travel - = - 6 open
          IoT Module Get GPIO # [35 *] Analog Value
    IoT Module Get GPIO # 35 T Analog Value Threshold T
    or > loT Module Get GPIO # 36 > Analog Value > 1 threshold >
```







#### Routes

```
Set LCD Screen Rotation to 0°
Create Blank Canvas: canvas - Size:
                                 Width ( 320 Height ( 240
Al Module Motor Driver Setup
Al Module Set Motor C ... 's Speed to 0 (0~255) Rotating Clockwise ...
Al Module Set Motor D > 's Speed to 0 (0~255) Rotating Clockwise >
Al Module Set Motor E ▼ 's Speed to 0 (0~255) Rotating Clockwise ▼
Al Module Set Motor F > 's Speed to 0 (0~255) Rotating Clockwise >
  travel - = 6 44 open 2
   junction count = 0
   speed = 105
   threshold
   Clear Canvas canvas All Conten
    Aa
    On Canvas canvas Draw Text
                                   X: [ 10 Y: [ 25
                                  Join strings | " (left:) "
                                                 IoT Module Get GPIO # 35 Analog Value
                                  Colour
    Aa
    On Canyas canyas Draw Text
                                               25
                                   X: 120
                                  Join strings 66 right: 22
                                                 IoT Module Get GPIO # 36 ▼ Analog Value
                                  Colour
    Show Canvas canvas
```



#### Obstacle

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

```
IoT Module Get GPIO # 35 - Analog Value | < 1 threshold -
      and I ToT Module Get GPIO # 36 Analog Value  threshold
Do Change junction_count by 11
                junction_count · = · 1
          or junction_count = 2
           300
           1000
         junction_count - = - 3
           300 Millisecond
          1000 Millisecond
          travel - = ( " close )
   Else if junction_count = 4
           300 Millisecond
           1000
          travel - = ( (close) 22
            IoT Module Get GPIO # 35 • Analog Value  threshold •
      or IoT Module Get GPIO # 36 Analog Value (threshold
Do Set travel - = 5 " open "
      travel - = - | " open "
                 IoT Module Get GPIO # 35 - Analog Value > 1 threshold
                 loT Module Get GPIO # 36 Analog Value  threshold
                 IoT Module Get GPIO # 35 Analog Value  threshold
```





#### **Image Identification**

```
Set Image Capturing Size: QVGA (320*240)
 Set Image Capturing Color Mode: Colorful
 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
 Load model file from path //root/preset/model/cocopi FloodingMission
                                                              Use these values to create an array of Numbers 5 44 Supply Station 2
                                                                                                              W Rescue Station
                                                                                                              " Clearing_Zone 2
                                                                                                              66 B) 2
 41
Al Module Set Motor C * 's Speed to 0 (0~255) Rotating Clockwise *
Al Module Set Motor D : 's Speed to 0 (0~255) Rotating Clockwise :
Al Module Set Motor E → 's Speed to 0 (0~255) Rotating Clockwise →
Al Module Set Motor F → 's Speed to ↑ 0 (0~255) Rotating Clockwise →
 Al Module Servo Setup
 Al Module Set Servo on GPIO # P0 Rotate to 0 Degree (0*~180*)
 Al Module Set Servo on GPIO # P1 Rotate to 95 Degree (0°~180°)
    travel - = 64 open
    junction_count - = 0
    threshold = 150
```

```
*Main code (See next page for functions)
```

```
Repeat foreve
o detection
  O If
                 IoT Module Get GPIO # (35 - Analog Value < - ) threshold
                  IoT Module Get GPIO # (36 - Analog Value < 1 threshold
               junction_count - | ≤ - | 1
              300 Milliseconds
                                                                      IoT Module Get GPIO # 35 Analog Value
                                                                     IoT Module Get GPIO # 36 Analog Value
              1000
                      Close 2
                                                                travel = = " " (open )
               junction_count - = 5
                                                                            300 Millisecond
                                                                            IoT Module Get GPIO # 36 - Analog Value ( threshold
                                                                           IoT Module Get GPIO # 35 - Analog Value ( threshold
              1500
                      Close >
                                                                          loT Module Get GPIO # 36 - Analog Value > 1 threshold
      Else if
                    junction_count - = - 2
                   junction_count = = 4
              300 Millisecond
              junction count - = - 3
              300 Milliseconds
              1500 Millisecond
              1500
                      " close
```



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

## Image Identification

```
efine detection
                     Get the Image Captured from Camera
    img_detection
   img_detection
                      Set Canvas img_detection - Cut t
                                                         X: C 48 Y: C 8
                                                          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 in list Tested Objects attribute
        On Canvas img detection Draw Text
        Set Start Coordinate
                                              From [12] 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 •
                                               Join strings From i Get the recognized tested Object's Name
                                             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 [ - Get the recognized tested Object's Y Coordinate of the bounding box
                                                              Width From i Get the recognized tested Object's Width of the bounding box
                                                              Height From Get the recognized tested Object's Height of the bounding box
                 From [ • Get the recognized tested Object's Name • ] = • | " Supply_Station "
                 From ( Get the recognized tested Object's (Y Coordinate of the bounding box ( 80
    w Canvas img_detection -
```

```
Al Module Set Motor C 

's Speed to speed 

(0~255) Rotating (Anti-Clockwise 

•
 Al Module Set Motor D ▼ 's Speed to speed ▼ (0~255) Rotating (Anti-Clockwise ▼
Al Module Set Motor E ▼ 's Speed to Speed ▼ (0~255) Rotating (Anti-Clockwise ▼
 Al Module Set Motor F : 's Speed to speed : (0~255) Rotating Anti-Clockwise :
 Al Module Set Motor C * 's Speed to speed * (0~255) Rotating Clockwise *
 Al Module Set Motor D : 's Speed to speed (0~255) Rotating Clockwise :
 Al Module Set Motor E 's Speed to speed (0~255) Rotating Clockwise •
 Al Module Set Motor F 's Speed to speed (0~255) Rotating Clockwise
Al Module Set Motor C v 's Speed to speed v (0~255) Rotating Clockwise v
Al Module Set Motor D * 's Speed to speed * (0~255) Rotating (Anti-Clockwise *
 Al Module Set Motor E ▼ 's Speed to speed ▼ (0~255) Rotating Clockwise ▼
Al Module Set Motor F * 's Speed to * speed * (0~255) Rotating Anti-Clockwise *
 Al Module Set Motor C ▼ 's Speed to speed ▼ (0~255) Rotating (Anti-Clockwise ▼
Al Module Set Motor D v 's Speed to speed v (0~255) Rotating Clockwise v
 Al Module Set Motor E • 's Speed to speed • (0~255) Rotating Anti-Clockwise •
 Al Module Set Motor F > 's Speed to speed > (0~255) Rotating Clockwise >
```

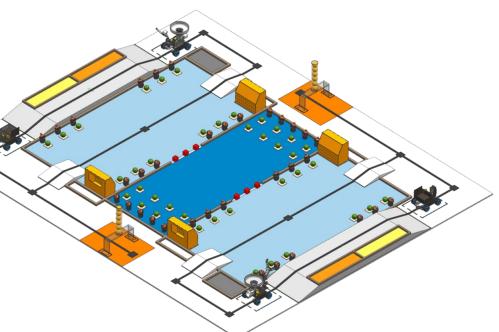
```
Al Module Set Motor C ■ 's Speed to 0 (0~255) Rotating Clockwise ■
Al Module Set Motor D * 's Speed to 0 (0~255) Rotating Clockwise *
Al Module Set Motor E * 's Speed to * 0 (0~255) Rotating Clockwise *
Al Module Set Motor F 's Speed to 0 (0-255) Rotating Clockwise •
     1000 Millisecond
     300 Millisecor
Al Module Set Servo on GPIO # P0 Rotate to 110 Degree (0*~180*)
Al Module Set Servo on GPIO # P0 Rotate to 0 Degree (0'~180')
     500 Millisecond
    t 300 Millisecond
    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, Obstacle

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

<sup>\*</sup>The initial state for Path to be set as Supply



## Path Decision Making

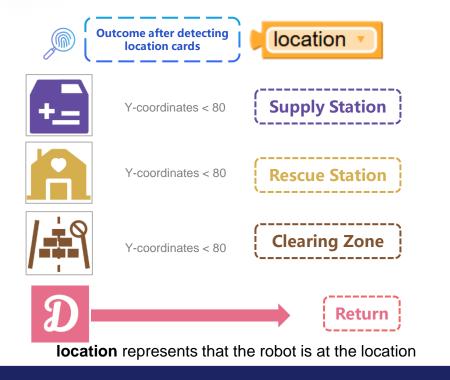
```
path · | = · | " (supply )"
        junction count - = - 1
                 " close "
        junction_count v = v 5
       300 Milliseconds
                 " close ?
Else if
         junction_count = = 2 or junction_count = = 4
       junction_count • = • 3
   Al Module Set Servo on GPIO # P1 Rotate to 50 Degree (0*~180*)
                           Lower the bar after collecting the supplies
                           to prevent hitting the structure. Raise the
        1500 Milliseconds
                           bar after completing a reverse and right
                           turn.
    Al Module Set Servo on GPIO # P1 T Rotate to 95 Degree (0'~180')
                  ( Filled )
```

```
" rescue "
Do 😝 If
                                          = 1 1
                           junction count
                           junction count
                                          = - [2
            or 🔻
                           junction count
                                          = - 4
                           iunction count
                                          = - 5
             300 Milliseconds
         turnriah
                      " close "
                            = - 3
              junction count -
            wait
              300 Milliseconds
                       " close "
```

```
66 obstacle 22
Do 🔯 If
                    junction count .
                    junction count = = 2
             300 Milliseconds
             1000
                      " close "
             junction_count - = - 3
             300 Milliseconds
              1000
                      " close "
    Else if
             junction_count = = 4
    Do front
            300 Milliseconds
            1000
                      " close "
            travel
            detect
```



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



Robot | command v

A Rescue Personnels C

Collect Supplies

**3** Clear Debris

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





Detection
Outcome Code

```
Recognized Any tested Object
 For each item in list Tested Objects attribute
Do [object Object] On Canvas i...

    If

                                                                From (i • Get the recognized tested Object's Name
                                                                                                                                                                                                              = V Supply Station >>
                                                              From 1 To Get the recognized tested Object's Y Coordinate of the bounding box T
                                                                           " supply station "
                                     location
             Else if
                                                               From Get the recognized tested Object's Name
                                                                                                                                                                                                              = V Kescue Station 22
                                                               From ( Get the recognized tested Object's Y Coordinate of the bounding box
                                                                             "rescue station"
                                       location
                                                               From (i T) Get the recognized tested Object's Name T) = T (" Clearing Zone "
                                                                From (i • Get the recognized tested Object's Y Coordinate of the bounding box • ( • ) 80
                                                                              " clearing zone "
             Do
                                      location
                                      From I Get the recognized tested Object's Name From I Get the recognized tested Object's Name From I From I Get the recognized tested Object's Name From I Get the Recognized tested tested Object's Name From I Get the Recognized tested t
                                       location
                                                                               " return "
                                       detect
             Else if
                                      From (i Telegraphics Get the recognized tested Object's Name Telegraphics (A)
             Do
                                                                                  " go rescue "
                                       command
                                      command
                                                                               Go clear "
                                       From (i 🔻 Get the recognized tested Object's Name 🔻 🖃 🥨 🕻 🕻 🗀
                                                                               go collect "
                                       command
            location
                                                 " 🔳 "
```



#### Object Detection

Does the robot have to constantly do object detection?

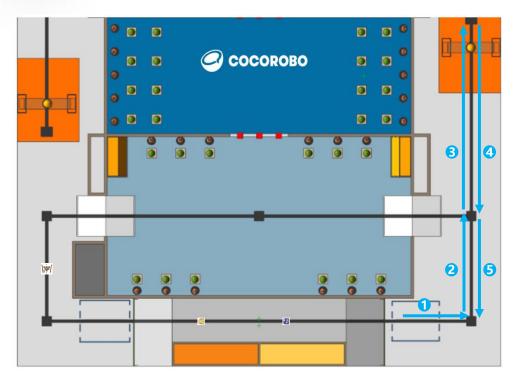
Which path does not require object detection?



- 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.



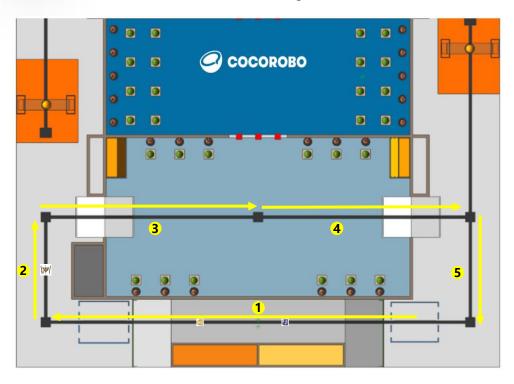
```
Detection "Off"
Set detect
Set detect -
                     Detection "On"
```

```
junction_count - | ≤ - 1
    300 Millisecond
                             Turn off detection after passing
   junction_count - = - 5
                             the first junction
   300 Milliseconds
                             Turn on detection after passing the
   1500
                             last junction, robot needs to locate
                             position to drop off supplies
     junction_count = 2 2 or junction_count = 4
   300 Milliseconds
   junction_count = 3
Al Module Set Servo on GPIO # P1 Rotate to 50 Degree (0°~180°
   300 Millisecond
    1500 Millisecond
   1500
Al Module Set Servo on GPIO # P1 - Rotate to 95 Degree (0°~180°)
         Close 2
   container = [ " Filled "
```



Analysis

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



```
Set detect = 0 Detection "Off"

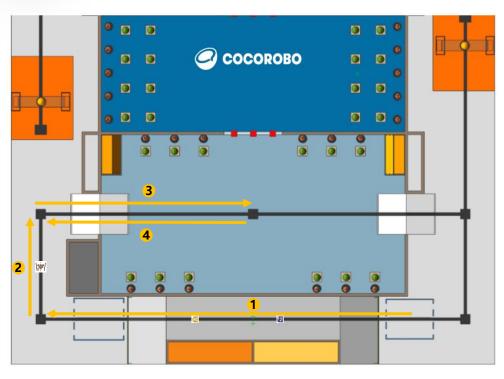
Set detect = 1 Detection "On"
```

```
= - | " (rescue) "
Do 😝 If
                        junction_count
                       junction count
          or -
                        junction count
                      junction_count - = - 5
   Do front
                          Turn off detection when "Junction
                           Count" is 1, 2, 4 or 5.
           1000 Milliseconds
                   Close 2
                   0
               junction count -
            junction_count
                             Turn on detection when
           300 Milliseconds
                             "Junction Count" = 5 or = 3
           1000
                  Close 2
```



Analysis

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



```
Set detect = 0 Detection "Off"

Set detect = 1 Detection "On"
```

```
66 obstacle
      junction count - = 1
     junction_count - = - [2]
300 Millisecond
1000
                         Turn off detection when
      Close 2
                         "Junction Count" = 1 \text{ or } = 2.
junction_count - = - 3
300 Millisecond
1000
       Close 2
junction_count - = - 4
it 300 Milliseconds
                         Turn on detection when
                         "Junction Count" = 3 or = 4
       Close
```





```
Recognized Any tested Object and detect detect and detect 
                                                                                                                                                                                                                                                                                                                                                                                                                  Location and Command cards will only be
                                                                                                                                                                                                                                                                                                                                                                                                                  detected if detection is turned "On"
Do For each item in list Tested Objects attribute
                               Do [object Object] On Canvas i...
                                                       🔯 If
                                                                                                                                                     From [ Get the recognized tested Object's Name = = 1
                                                                                                                                                                                                                                                                                                                                                                                                                                                             Supply Station 22
                                                                                                                                                    From [ Get the recognized tested Object's Y Coordinate of the bounding box | < | 80
                                                                                                                                                                         upply_station >>
                                                       Do Set location
                                                       Else if
                                                                                                                                                        From ( Get the recognized tested Object's Name = Rescue Station >
                                                                                                                                                           location
                                                                                                                                                                               "rescue station 22
                                                       Else if
                                                                                                                                                      From I Get the recognized tested Object's Name = Clearing Zone
                                                                                                                                                         From [ • Get the recognized tested Object's Y Coordinate of the bounding box • < • 80
                                                                                                                                                                         Clearing zone
                                                                                                   location
                                                                                                       From ( Get the recognized tested Object's Name = 5 44 D 22
                                                                                                                                                              = ( " return "
                                                                                                                                                                                                                                                                               Detection is turned "Off" after the return card on the mechanical robot is detected
                                                                                             From [ Tom Get the recognized tested Object's Name Tomas Get the Recognized tested tested Object's Name Tomas Get the Recognized tested tested
                                                                                                                                                                                          " go rescue
                                                                                                       From (i v Get the recognized tested Object's Name v = v (B)
                                                                                                                                                                                          " go clear "
                                                                                                     From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To From To Get the recognized tested Object's Name To Get the Recognized tested test
                                                                                                                                                                                 Go collect 2
```



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.





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.



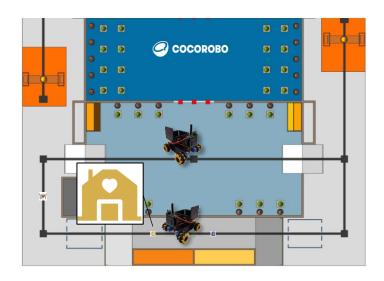


```
Location set as Supply Station
                                                      Path
                                                                                        Container
                                                                   "Supply" &
                                                                                                         = "Filled"
                                              and 🕶
                            " supply_station "
                                                                                and 🔻
                                                                    " (supply "
                                                                                                        66 (filled) >>
                                                                                         container
       🔯 If
                 command
                                 " go_rescue "
            Unload
                                                                                    COCOROBO
                        " rescue "
               path
       Else if
                 command - = -
                                 " go_clear "
            Unload
                        " obstacle "
       Else
            Unload
                         " closed "
            Set travel
                         " supply "
               path
```



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.

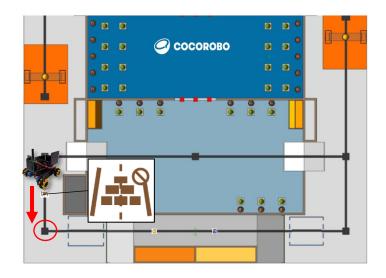




```
= "Rescue" & Container
Location set as Rescue Station
                                                    Path
                                                                                                     = "Filled"
  Else if
                                           and 🔻
                          "rescue station
                                                                 " rescue "
     □ If
               command -
                              " go clear "
           Unload
                                                                                COCOROBO
                       66 obstacle 22
      Else if
                         = v ( go_collect )
                                                                      0 0
           Unload
           turnleft
           Wait (1000)
                        " closed "
                       " supply "
           Set path
      Else
          Unload
                      " rescue "
           Set path
```



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.





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

Else if location = "Clearing zone " and path = "Container = "Filled"

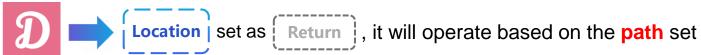
Do Unload Set path = "Supply"

Set junction_count = 1
```





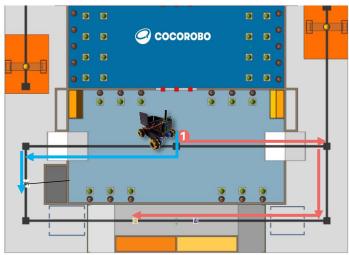
When the transport robot reaches point ①, it will wait for the mechanical robot to load objects onto its container. After the process is complete, it will detect the command card on the mechanical robot and continue on with its actions



```
Path = "Rescue"

Do Set container = "filled "

Font | Filled | Fil
```





#### Pause Sequence Analysis



After the transport robot arrives at point ①, it waits there until it detects the ② instruction from the mechanical robot. When the ③ 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;

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

```
Else if loT Module Get GPIO # 35 Analog Value  threshold or loT Module Get GPIO # 36 Analog Value  threshold or loT Module Get GPIO # 36 Analog Value  threshold or loT Module Get GPIO # 36 Analog Value  wait = 0

Do Set travel =  "open "
```



#### Pause Sequence

```
" rescue "
Do 🔯 If
                           junction count - = 1
                            junction count
            or 🔻
                            junction count -
                           junction_count - = - 5
   Do front
             (300) Milliseconds
                        " close 2
            detect
                  junction_count = 5
            junction count -
            (300) Milliseconds
             1000
            travel
                        " close "
```

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

```
= v ( obstacle »
Do 😝 If
                   junction count - = 1
                 junction_count = = 2 2
   Do front
           300 Milliseconds
                     " close "
            junction count - = - 3
            300 Milliseconds
                     " close
            junction_count - = - 4
   Do fro
            300 Milliseconds
            1000
                     " close "
```

A wait at point **()** is required during the **obstacle** route

```
Do Set wait = 0 0

Set wait = 0 0

Do Set container = "" rescue "

Do Set container = "" filled "

front

Else if path = "" obstacle "

Do Set container = "" filled "

Set travel = "" closed "

turnright

Wait 1000 Milliseconds
```

End the wait and run the return sequence after detecting ① on the mechanical robot.



# Unloading

```
Define unload
  front
        1000 Milliseconds
  left
       300 Milliseconds
      500 Milliseconds
  Al Module Set Servo on GPIO # P0 Rotate to 110 Degree (0°~180°)
  Wait 2000 Milliseconds
 Al Module Set Servo on GPIO # P0 Rotate to 0 Degree (0°~180°)
  Wait 500 Milliseconds
  right
       300 Milliseconds
                       0
  Set junction count
     container
                    " empty "
     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)

```
Set Image Capturing Size: QVGA (320*240)
Set Image Capturing Color Mode: Colorful
Set LCD Screen Rotation to 0° -
4□
Al Module Set Motor C 💌 's Speed to 🕻 0 (0-255) Rotating Clockwise 🔻
Al Module Set Motor D v 's Speed to 0 (0~255) Rotating Clockwise v
Al Module Set Motor E * 's Speed to 0 (0-255) Rotating Clockwise *
Al Module Set Motor F : 's Speed to 0 (0~255) Rotating Clockwise :
Al Module Set Servo on GPIO # P0 • Rotate to 0 Degree (0*~180*
Al Module Set Servo on GPIO # P1 Rotate to 95 Degree (0*~180
 oad model file from path //root/preset/model/cocopi_FloodingMission
                                                                Use these values to create an array of Numbers | "Supply_Station |
                                                                                                                 " Rescue_Station
   detect ·
```

```
Do Detection
   ☐ If
                 IoT Module Get GPIO # (35 ) Analog Value ( ) threshold
                  IoT Module Get GPIO # 36 T Analog Value Threshold
            junction_count v by [ 1
               path = = " " Supply >
                    = V K Rescue 22
                    = V GObstacle
                      IoT Module Get GPIO # 35 v Analog Value
                       wait = 0
         travel
                   " open "
   travel = T 4 open >>
  Do 🔯 If
                      IoT Module Get GPIO # 35 TAnalog Value Threshold
                      IoT Module Get GPIO # 36 v Analog Value < v threshold
                      IoT Module Get GPIO # 35 T Analog Value Threshold T
                     IoT Module Get GPIO # 36 Analog Value > (threshold
```





```
Define Supply
          junction_count - ≤ 1
                  " close "
                 " close "
          junction_count = 2 or junction_count = 14
        junction_count = = 3
    Al Module Set Servo on GPIO # P1 Rotate to 50 Degree (0°~180°)
         1500 Millisecond
     Al Module Set Servo on GPIO # P1 Rotate to 95 Degree (0°~180')
```

```
Define Rescue
                        junction count
                        junction_count
         Or 🔻
                        junction_count - | = - 4
                        junction_count - = - 5
Do front
         Milliseconds
      turnright
          1000
                   " close "
         detect
               junction_count = 1 5
           junction_count = 3
          300 Milliseconds
           1000 Milliseconds
                   " close "
         detect
```

```
junction_count - = - 1
       junction count = 2
300 Milliseconds
1000
        " (close "
junction_count - = - 3
300 Milliseconds
1000
        " (close "
junction_count - = - 4
300 Milliseconds
1000
         " close "
```



#### Reference Program (Object Detection)

```
Set Canvas img detection - Cut
                                                                                                                                        X: 48 Y: 8
                                                                                                                                           Width: 224 Height: 224
Load Preset A.I. Model: Object Recognition Mode
Recognize from: img detection -
                                 ecognized Any tested Object and detect detect and
            For each item i in list Tested Objects attribute
                                                                    and From ( Get the recognized tested Object's Y Coordinate of the bounding box 80
                        Do Set location - = 6 44 supply_station 22
                       Else if
                                                                   From [ • Get the recognized tested Object's Name • From [ • Rescue_Station 2]
                                            and From Get the recognized tested Object's (Y Coordinate of the bounding box 4 1 80
                                       et location = 66 rescue station 22
                                                                   From ( • Get the recognized tested Object's Name • = • 64 Clearing_Zone **
                                             and From From Get the recognized tested Object's Y Coordinate of the bounding box 80
                                            location = C (clearing_zone )
                                             From [ Get the recognized tested Object's Name From [ Set the Recognized
                                             command - = " " go_rescue "
                                             From i • Get the recognized tested Object's (Name • ) = • | " ( B )
                                             command - = 6 44 go clear 22
                                            Set command - = [ " go_collect "
```

```
location · = · ( supply_station ); and ·
                                                     path - = - " supply " and - container - = - " filed "
              command = G (go_rescue)
              command - = - ( go_clear)
               1000 Milliseconds
              travel - = [ " closed "
              path · = ( "(supply)
            location · = · | " rescue_station » and ·
                                                       path · = · " rescue " and · container · = · " filed "
              command - - G Gg_clear 22
              path - - (4 obstacle 22
              command - = - " go_collect "
               1000 Millisecond
              travel - - ( " closed 2
              path - = 6 " supply !
              path · = ( " rescue "
           location - = - | " (clearing_zone ) and - |
                                                     path . = . . " obstacle " and . container . = . " filled "
         junction_count - = ( -1
        location - = - " return "
             path - = - " rescue >
            container - = ( 44 filled) **
            path - = - | " obstacle "
             travel - 66 closed 27
              1000 Milisecond
et Canvas img_detection • 's Starting Coordinate as: X: 48 Y: 8
  Canvas imp detection -
```

```
Define unload front
Wait 1000 Milliseconds
left
Wait 300 Milliseconds
stop
Wait 500 Milliseconds
Al Module Set Servo on GPIO # PO = Rotate to 110 Degree (0*-180*)
Wait 2000 Milliseconds
Al Module Set Servo on GPIO # PO = Rotate to 10 Degree (0*-180*)
Wait 2000 Milliseconds
Al Module Set Servo on GPIO # PO = Rotate to 10 Degree (0*-180*)
Wait 300 Milliseconds
right
Wait 300 Milliseconds
Set junction_count = 10
Set container = 44 empty 27
```



# Reference Program (Movement)

```
Define back
                                                                                     Al Module Set Motor C 3 's Speed to ( speed 5
                                                                                                                                    (0~255) Rotating Clockwise •
 Al Module Set Motor C 3 's Speed to Speed
                                                (0~255) Rotating Anti-Clockwise
                                                                                                                                     (0~255) Rotating Clockwise
 Al Module Set Motor F 's Speed to speed
                                                (0~255) Rotating Anti-Clockwise
                                                                                     Al Module Set Motor F . 's Speed to speed
                                                                                                                                    (0~255) Rotating Clockwise
Define Fright
                                                                                   Define (left)
                                                                                                                                                                      Define Stop
                                                                                                                                                                         Al Module Set Motor C → 's Speed to 0 (0~255) Rotating Clockwise
 Al Module Set Motor C 's Speed to speed
                                                (0~255) Rotating Clockwise •
                                                                                    Al Module Set Motor C 's Speed to speed
                                                                                                                                    (0~255) Rotating Anti-Clockwise
  Al Module Set Motor Fire 's Speed to | speed
                                                                                     Al Module Set Motor F 3 's Speed to speed
Define (turnleft)
                                                                                                 Define (turnright)
 Al Module Set Motor C 3 's Speed to
                                                                                                  Al Module Set Motor C : 's Speed to
  Al Module Set Motor D : 's Speed to (
                                                                                                   Al Module Set Motor D : 's Speed to |
                                                                                                                                                                 0~255) Rotating Clockwise -
  Al Module Set Motor Fire 's Speed to
                                                               0~255) Rotating Clockwise •
                                                                                                   Al Module Set Motor Fire 's Speed to (
                                                                                                                                                                (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



