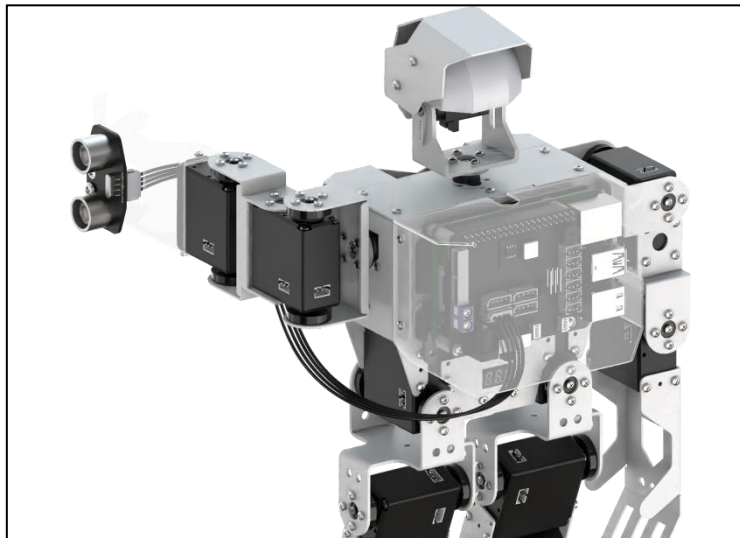# Lesson 3 Obstacle Avoidance

1. Getting Ready

Prepare an ultrasonic sensor and install it on the expansion hole of robot's left hand. The specific installation method can refer to the file "Sensor Installing and Wiring" under the same directory.

2. Working Principle

Let's look at the working principle:

Firstly, the distance to the obstacle ahead is detected by the ultrasonic sensor, and then set the ultrasonic measurement threshold. When the conditions are met, the robot automatically avoids the obstacle, otherwise it moves forward directly.

The source code of the program is located in /home/pi/TonyPi/Extend/Sonar_barrier.py

```
21        GPIO.output(31, 1)          #设置引脚输出高电平
22        time.sleep(sleeptime)       #设置延时
23        GPIO.output(31, 0)
24
25   # 抬起左手
26 ┌def hand_up():
27        Board.setBusServoPulse(8,330,1000)
28        time.sleep(0.3)
29        Board.setBusServoPulse(7,860,1000)
30        Board.setBusServoPulse(6,860,1000)
31        time.sleep(1)
32   # 放下左手
33 ┌def hand_down():
34        Board.setBusServoPulse(7,800,1000)
35        Board.setBusServoPulse(6,575,1000)
36        time.sleep(0.3)
37        Board.setBusServoPulse(8,725,1000)
38        time.sleep(1)
39   # 向左边伸手
40 ┌def hand_left():
41        Board.setBusServoPulse(8,330,1000)
```

3. Operation Steps

ℹ The entered command must pay attention to case sensitivity and space.

1) Turn on the robot and connect to Raspberry Pi desktop with VNC.

2) Click � or press "Ctrl+Alt+T" to open LX terminal.
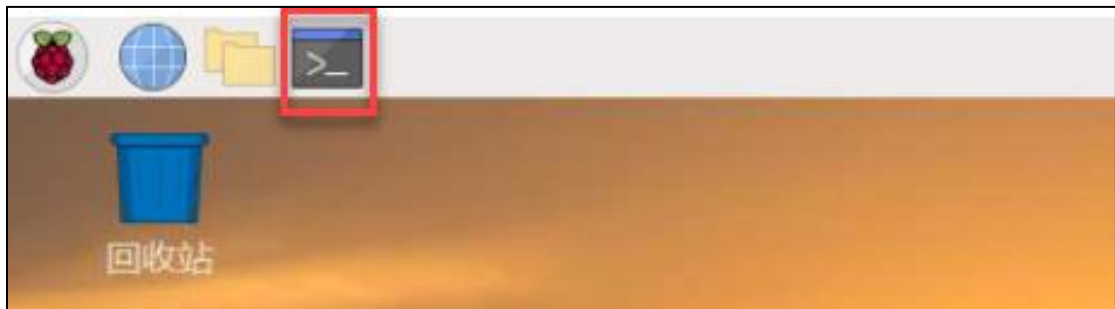
3) Enter "cd TonyPi/Extend/" command and press "Enter" to come to the directory of the game programmings.


```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

4) Enter "sudo python3 Sonar_barrier.py" command, and then press "Enter" to start the game

```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ sudo python3 Sonar_barrier.py
```

5) If want to exit the game, press "Ctrl+C" in the LX terminal. Please try multiple times if fail to exit.

4. Project Outcome

After the program is started, if the distance between the ultrasonic wave and the object ahead is less than or equal to 30cm, it is determined to be an obstacle. TonyPi Pro will raise its left hand to detect whether there is an obstacle ahead. If not, the robot will move forward; If there is an obstacle, the ultrasonic sensor will detect the robot of the left and right sides to control the movement of the robot.

5. Function Extension

The default distance of obstacle avoidance is 300mm. If want to modify the distance value (this section takes setting the distance as 100mm as example), please refer to the following steps:

1) Enter "cd TonyPi/Extend/" command and press "Enter" to come to the directory of the game programmings.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

2) Enter "sudo vim Sonar_barrier.py" command, and then press "Enter" to open the game programming file.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ sudo vim Sonar_barrier.py
```

3) Find the code in the program interface as the figure shown below.

```
57      while True:
58          if distance != 99999:
59              if distance <= 300: #detect the obstacle ahead
60                  distance = 99999
61                  hand_left() #reach to the left
62                  time.sleep(1)
63                  #detect the left three times
64                  for i in range(3):
65                      dist_left.append(distance)
66                      time.sleep(0.05)
67                  #take a average
68                  distance_left = round(np.mean(np.array(dist_left)))
69                  dist_left = []
70                  hand_up()
71
72              if distance_left <= 300: #Detect obstacles on the left
73                  distance_left = 99999
74                  hand_down() # Put down the left hand
75                  for i in range(5): #Turn right
76                      AGC.runActionGroup('turn_right')
77                      time.sleep(0.2)
78
79                  hand_up()
80                  time.sleep(1)
81                  # detect the left three times
82                  for i in range(3):
83                      dist_right.append(distance)
84                      time.sleep(0.05)
85
86                  distance_right = round(np.mean(np.array(dist_right)))
87                  dist_right = []
88
89              if distance_right <= 300: #Detect obstacles on the right
90                  distance_right = 99999
91                  hand_down()
                                                        71,1        52%
```

4) Press "i" on keyboard. When "Insert" appears in the lower left corner, which means it has entered the editing mode.

```
86                  distance_right = round(np.mean(np.array(dist
87                  dist_right = []
88
89              if distance_right <= 300: #Detect obstacles
90                  distance_right = 99999
91                  hand_down()
-- INSERT --                                            71,1
```

5) Modify "300" in "if distance <= 300" to "100", as the figure shown below:

```
57      while True:
58          if distance != 99999:
59              if distance <= 100: #detect the obstacle ahead
60                  distance = 99999
61                  hand_left() #reach to the left
62                  time.sleep(1)
63                  #detect the left three times
64                  for i in range(3):
65                      dist_left.append(distance)
66                      time.sleep(0.05)
67                  #take a average
68                  distance_left = round(np.mean(np.array(dist_left)))
69                  dist_left = []
70                  hand_up()
71
72              if distance_left <= 100: #Detect obstacles on the left
73                  distance_left = 99999
74                  hand_down() # Put down the left hand
75                  for i in range(5): #Turn right
76                      AGC.runActionGroup('turn_right')
77                      time.sleep(0.2)
78
79                  hand_up()
80                  time.sleep(1)
81                  # detect the left three times
82                  for i in range(3):
83                      dist_right.append(distance)
84                      time.sleep(0.05)
85
86                  distance_right = round(np.mean(np.array(dist_right)))
87                  dist_right = []
88
89              if distance_right <= 100: #Detect obstacles on the right
90                  distance_right = 99999
91                  hand_down()
-- INSERT --                                          78,25              52%
```

6) After modification, press "Esc" and then enter ":wq" (Please note that the colon is in front of wq). Then press "Enter" to save and exit the modified content.

```
87                      dist_right = []
88
89              if distance_right <= 100: #Detect
90                  distance_right = 99999
91                  hand_down()
:wq
```