

# Lesson 19 Self-balancing

## 19.1 Overview

The Self-balancing is developed based on the MPU6050 sensor. Once the self - balancing function is enabled, it ensures that the viewing angle of the robot's camera remains horizontal. When this function is operating normally, the robot is restricted from performing other operations. To disable the feature, simply click the relevant option again.

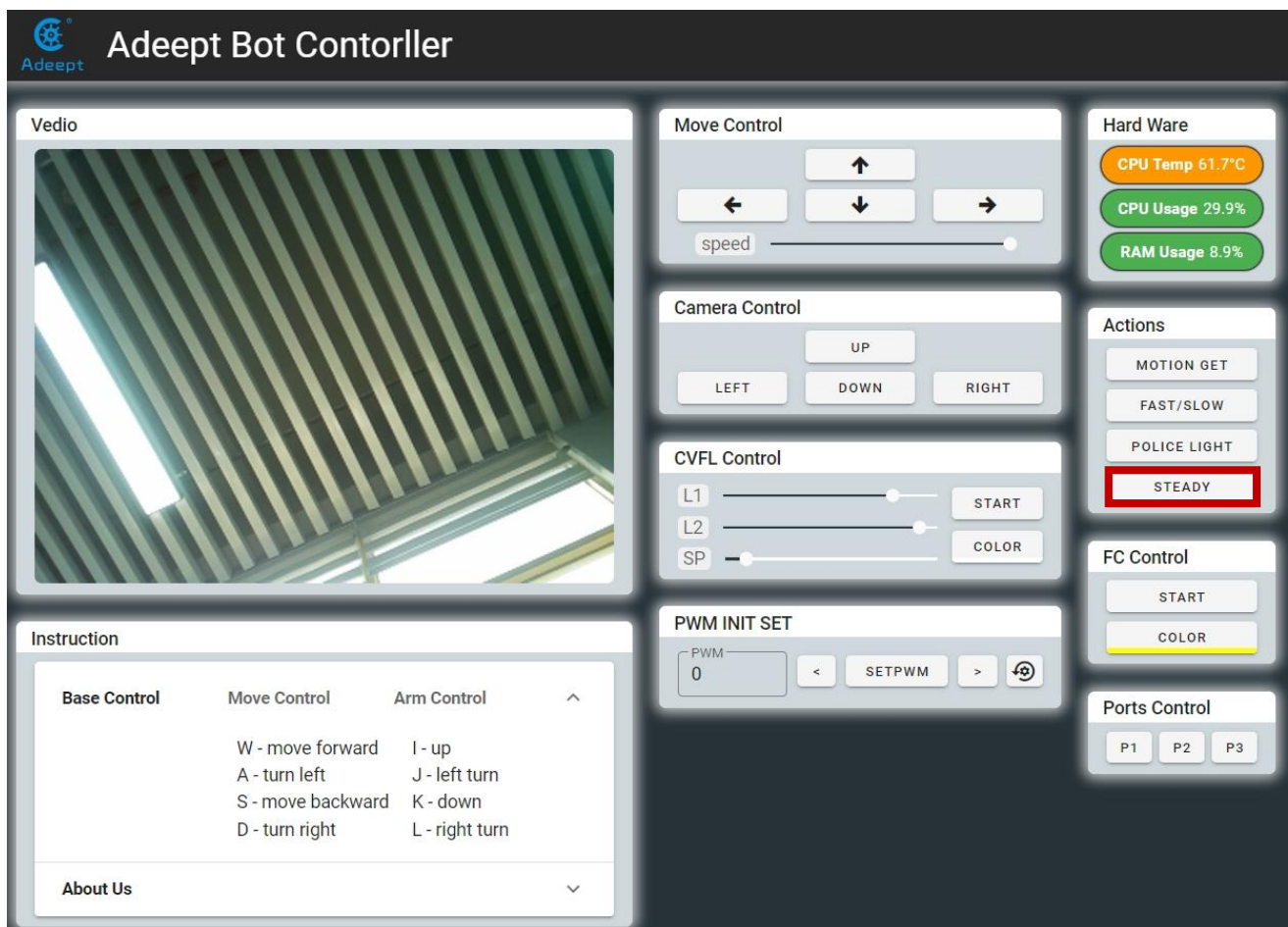
## 19.2 Introduction to Self-balancing

The Self-balancing utilizes the data from the MPU6050 sensor to detect the orientation and movement of the robot. By continuously monitoring the sensor readings, it can adjust the camera's position to maintain a horizontal view. This is achieved through a series of algorithms that calculate the necessary adjustments based on the detected tilt angles. The MPU6050 provides accurate acceleration and gyroscope data, which serves as the foundation for the self - balancing and camera - stabilization mechanisms. This ensures that regardless of the robot's movement or orientation changes, the camera's line of sight remains parallel to the horizon, offering a consistent and stable visual output.

## 19.3 Running the Self-balancing

### Running the Automatic Obstacle Avoidance program

1. Start the Adeept RaspClaws Robot. It may take about 30-50s to boot.
2. After Adeept RaspClaws is turned on, open the Chrome browser on your mobile or computer, enter the IP address of your Raspberry Pi and access port ":5000" into the IP address bar, like this: **192.168.3.31:5000**. The web controller will then be displayed on the browser.



3. After clicking "**STEADY**", the RaspClaws Robot will remain level.

4. Click "**STEADY**" again to disable the -> function.

## 19.4 Code

The main code is as follows. For the complete code, please check [Move.py](#).

```

01 def steady_X():
02     if leftSide_direction:
03         pwm.set_pwm(0,0,pwm0+steady_X_set)
04         pwm.set_pwm(2,0,pwm2)
05         pwm.set_pwm(4,0,pwm4-steady_X_set)
06     else:
07         pwm.set_pwm(0,0,pwm0+steady_X_set)
08         pwm.set_pwm(2,0,pwm2)
09         pwm.set_pwm(4,0,pwm4-steady_X_set)
10
11     if rightSide_direction:
12         pwm.set_pwm(10,0,pwm10+steady_X_set)
13         pwm.set_pwm(8,0,pwm8)
14         pwm.set_pwm(6,0,pwm6-steady_X_set)
15     else:

```

```

16     pwm.set_pwm(10,0,pwm10-steady_X_set)
17     pwm.set_pwm(8,0,pwm8)
18     pwm.set_pwm(6,0,pwm6+steady_X_set)
19
20
21 def steady():
22     global X_fix_output, Y_fix_output
23     if mpu6050_connection:
24         accelerometer_data = sensor.get_accel_data()
25         X = accelerometer_data['x']
26         X = kalman_filter_X.kalman(X)
27         Y = accelerometer_data['y']
28         Y = kalman_filter_Y.kalman(Y)
29
30         X_fix_output += -X_pid.GenOut(X - target_X)
31         X_fix_output = ctrl_range(X_fix_output, steady_range_Max, -steady_range_Max)
32
33         Y_fix_output += -Y_pid.GenOut(Y - target_Y)
34         Y_fix_output = ctrl_range(Y_fix_output, steady_range_Max, -steady_range_Max)
35
36         '''
37         LEFT_I
38         '''
39         left_I_input = ctrl_range((X_fix_output + Y_fix_output), steady_range_Max, steady_range_Min)
40         left_I(0, 35, left_I_input)
41
42         '''
43         LEFT_II
44         '''
45         left_II_input = ctrl_range((abs(X_fix_output*0.5)+Y_fix_output), steady_range_Max,
46 steady_range_Min)
47         left_II(0, 35, left_II_input)
48
49         '''
50         LEFT_III
51         '''
52         left_III_input = ctrl_range((-X_fix_output + Y_fix_output), steady_range_Max, steady_range_Min)
53         left_III(0, 35, left_III_input)
54
55         '''
56         RIGHT_III
57         '''
58         right_III_input = ctrl_range((X_fix_output - Y_fix_output), steady_range_Max, steady_range_Min)
59         right_III(0, 35, right_III_input)
60
61         '''
62         RIGHT_II
63         '''
64         right_II_input = ctrl_range((abs(-X_fix_output*0.5)-Y_fix_output), steady_range_Max,
65 steady_range_Min)
66         right_II(0, 35, right_II_input)
67         '''
68         RIGHT_I
69         '''
70         right_I_input = ctrl_range((-X_fix_output-Y_fix_output), steady_range_Max, steady_range_Min)
71         right_I(0, 35, right_I_input)

```