

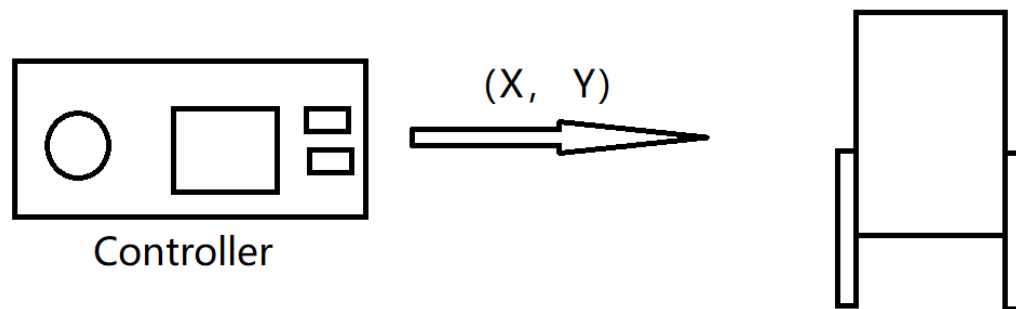
## Lab 4

Xiaoxin Zhou

Introduction of direction:

From the coding, there is seven directions, in order by 1(forward), 2(backward), 3(left Front), 4(right Front), 5(back Left), 6(back Right), 7(stop). The app controls the direction. However, we are not able to proportional control. I came out an algorithm for the proportional control.

Assume we can get the (X, Y) value from the Bluetooth APP.



From our Controller, there is a circle to represent the joystick, and all the rectangles represent the different functions as line following and object detection.

Assume that the maximum number range of X and Y is  $[-100, 100]$ . We need to set the number of speeds for motors to  $[0, 100]$ . If the X value is positive, the robot moves forward. If the X value is negative, the robot moves backward. If the Y value is positive, the robot moves right, and the Y value is negative, the robot moves left. If X is zero, the robot should not move forward or backward. If Y is zero, the robot should not turn right or left. If the user sends  $(-15, 50)$  to the robot, the robot should be slightly left with 15 speed and 50 speed forward. This is the cheapest way to solve Lab 4 task and without adding any extra sensor.

Suppose the maximum number range of X and Y is  $[-100, 100]$ . But the number of speeds is not  $[-100, 100]$ . We need to know the max number of the signal sent. Assume the number of the max is 100. We can divide by the number of users pushed to find the proportional to the speed. When the user moved the joystick at a 250 value (some value from the X or Y, we do  $(250/100) * 100 = 25\%$ . And set the number of speeds for the motor to 100 or higher if necessary. The speed of the robot will be  $100 * 25\%$ . For example, if the user sends  $(-15, 50)$  to the robot, the robot should be slightly left with 15% speed and 50% speed forward. It will work the something as the previous one and without calculating the arc.