QR code tracking

1. Program Functionality

> After the program starts, it automatically detects the QR code on the screen, and the servo gimbal will lock onto the center of the QR code. Pressing the `q/Q key` exits the program.

⚠ This function displays the box only after the QR code characters are successfully decoded. The recognition rate and tracking performance will be lower than other visual examples!

2. Program Code Reference Path

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/yahboomcar_astra/qrTracker.py
```

- qrTracker.py

  This function primarily performs QR code recognition, calculates the desired servo rotation angle based on the center coordinates of the QR code, and transmits the servo angle control data to the robot.

3. Program Startup

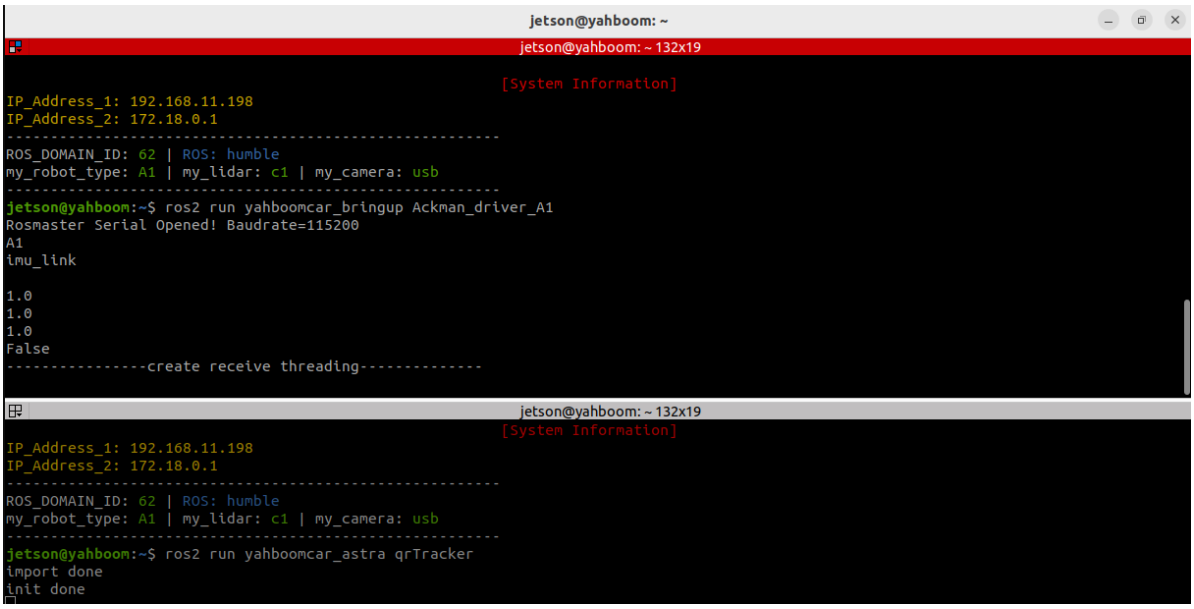
3.1. Startup Commands

For the Raspberry Pi 5 controller, you must first enter the Docker container. This is not necessary for the Orin board.

Enter the Docker container (for steps, see the Docker section [3. Docker Submission and Multi-Terminal Access]).

Enter the terminal:

```
# Start the car chassis
ros2 run yahboomcar_bringup Ackman_driver_A1
# Start the QR code tracking program
ros2 run yahboomcar_astra qrTracker
```



When a QR code is detected, the code will be automatically outlined and its center position will be displayed. The servo gimbal will begin tracking.

After the program starts, the following screen will appear:

In addition, we can enter the following command to print information about the target center coordinates:

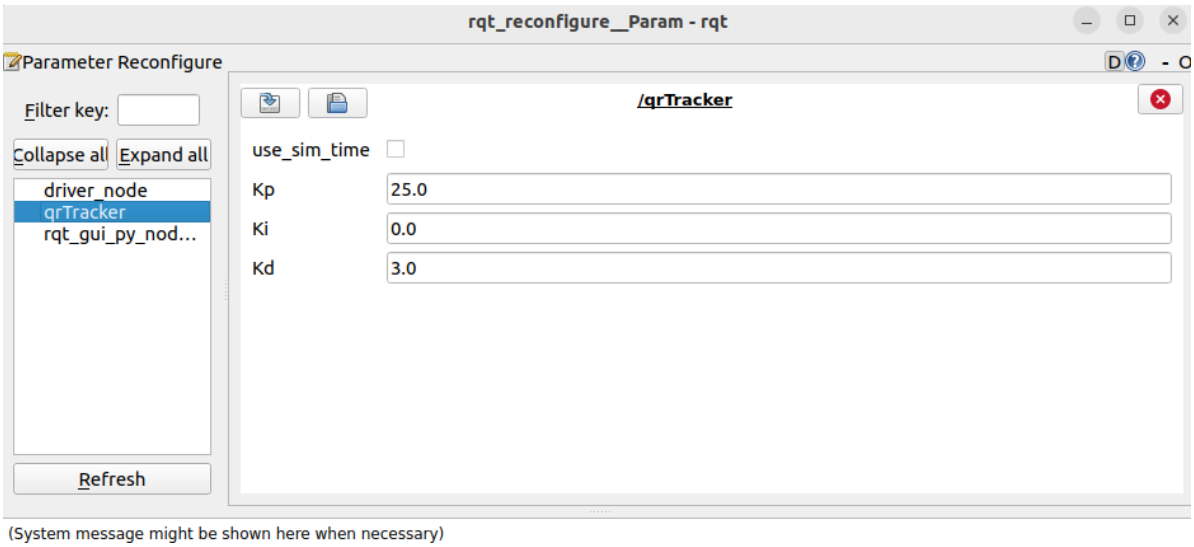
```
ros2 topic echo /Current_point
```

```
jetson@yahboom:~$ ros2 topic echo /Current_point
anglex: 42.0
angley: 220.0
distance: 0.0
---
anglex: 49.0
angley: 222.0
distance: 0.0
---
```

3.2 Dynamic Parameter Adjustment

In the terminal, enter:

```
ros2 run rqt_reconfigure rqt_reconfigure
```



☑ After modifying the parameters, click a blank area in the GUI to write the parameter values. Note that these values will only take effect for the current startup. To permanently implement them, you need to modify the parameters in the source code.

As shown in the figure above,

- qrTracker is primarily responsible for servo gimbal movement, adjusting PID-related parameters to achieve optimal gimbal motion.

🛠 Parameter Analysis:

[Kp], [Ki], [Kd]: PID control of servo gimbal speed during tracking.

4. Core Code

4.1. qrTracker.py

This program has the following main functions:

- Opens the camera and captures an image;
- Recognizes the QR code in the image and obtains its center position;
- Publishes the QR code's center coordinates to the topic /Current_point;
- Calculates the servo angle and publishes the angle data.

Some core code is as follows:

```
# Create a publisher to publish the center coordinates of the QR code
self.pub_position = self.create_publisher(Position, "/Current_point", 10)
# Define the servo data publisher
self.pub_Servo = self.create_publisher(ServoControl, 'Servo', 10)
...
# QR code recognition function
def decodeDisplay(self, image):
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    barcodes = pyzbar.decode(gray)
    for barcode in barcodes:
        (x, y, w, h) = barcode.rect
        cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 255), 3)
        return image, (x, y, w, h)
    return None, (0, 0, 0, 0)
...
# Calculate the center coordinates and publish
if processed_frame is not None:
    frame = processed_frame
    self.Center_x = x + w // 2
    self.Center_y = y + h // 2
    cv.circle(frame, (self.Center_x, self.Center_y), 5, (0, 255, 0), -1)
    threading.Thread(target=self.execute, args=(self.Center_x,
self.Center_y)).start()
...
# According to the center coordinates, use the PID algorithm to calculate the
servo angle
def execute(self, point_x, point_y):
    position = Position()
    position.anglex = point_x * 1.0
    position.angley = point_y * 1.0
    self.pub_position.publish(position)
    [x_Pid, y_Pid] = self.PID_controller.update([point_x - 320, point_y - 240])
    ...
    # Publish the calculated servo angle
    self.pub_Servo.publish(self.servo_angle)
```