6. QR code instructions

6.1 Introduction to gameplay

This course mainly uses the CSI camera of the RDK-X3 board to obtain the picture of the camera, identify the two-dimensional code information, and control the car movement according to the two-dimensional code information.

6.2 Core content analysis

Import the QR code parsing library pybar

```
import pyzbar.pyzbar as pyzbar
from PIL import Image
```

If pybar is not installed, open the terminal and run the following command to install it.

```
pip3 install pyzbar
sudo apt install libzbar-dev
```

The gray image is analyzed, and the information of two-dimensional code and image position in the image are extracted. If there is no QR code in the image, the message is None.

```
def detect_qrcode(image):
    # 转为灰度图像
    # Convert to grayscale image
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    barcodes = pyzbar.decode(gray)
    for barcode in barcodes:
        # 提取二维码的数据和边界框的位置 The data of the QR code and the position of
the bounding box are extracted
        (x, y, w, h) = barcode.rect
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type
        # print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
        car_control(barcodeData)
        return barcodeData, (x, y, w, h)
    return None, (0, 0, 0, 0)
```

Control the car movement according to the string command of info. car_speed is the speed of the car, which can be adjusted according to actual needs.

```
car_speed = 0.3
```

```
def car_control(info):
    # print(info)
    global g_car_runtime
    if g_car_runtime >= 0:
        return
    g_car_runtime = 100
    if info == "forward": # 前进forward
        g_car.set_car_run(1, car_speed*100)
    elif info == "back": # 后退back
        g_car.set_car_run(2, car_speed*100)
    elif info == "left": # 左平移left
        g_car.set_car_run(3, car_speed*100)
    elif info == "right": # 右平移right
        g_car.set_car_run(1, car_speed*100)
    elif info == "turnleft": # 左旋转turnleft
        g_car.set_car_run(5, car_speed*100)
    elif info == "turnright": # 右旋转turnright
        g_car.set_car_run(6, car_speed*100)
    elif info == "stop": # 停车stop
        g_car.set_car_run(7, 0)
        g_car_runtime = -1
```

Define a task_timeout task to stop the car automatically and not execute other QR code commands during the car movement. g_car_runtime is measured in 10 milliseconds. When g_car_runtime is greater than 0, it automatically decreases by 1 every 10 seconds. When g_CAR_Runtime expires, it automatically stops.

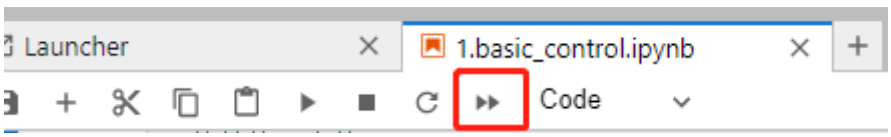
```
g_car_runtime = -1

def task_timeout():
    global g_car_runtime
    while True:
        if g_car_runtime > 0:
            print("g_car_runtime:", g_car_runtime)
            g_car_runtime = g_car_runtime - 1
        elif g_car_runtime == 0:
            print("auto stop")
            g_car.set_car_run(7, 0)
            g_car_runtime = -1
        time.sleep(.01)
```

6.3 Gameplay operations

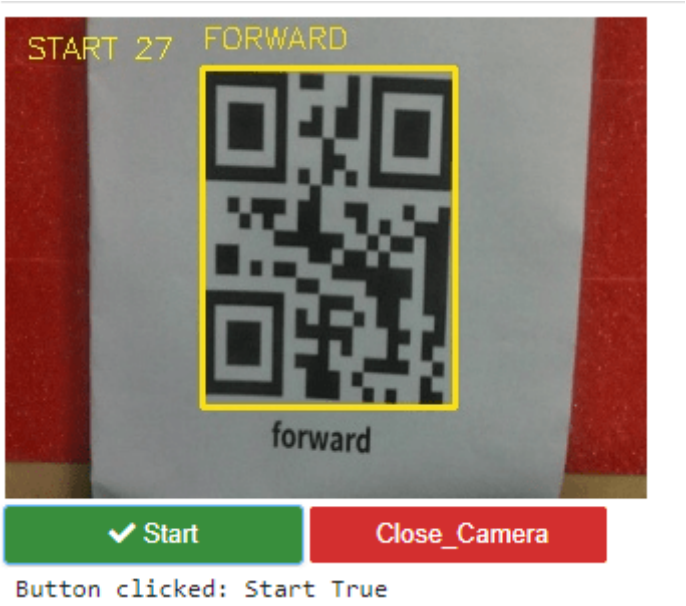
Open the jupyterLab client and find the code path:

```
/root/sunriseRobot/Samples/2_AI/06_qrcode_movement/qrcode_movement.ipynb
```

Click Run All Cells, and then drag to the bottom to see the generated controls.



Click the Start button below to open the QR code and execute the command. At present, the QR code that can be recognized in the routine is QRCode, and the information is "forward" for forward, "back" for backward, "left" for left translation, "right" for right translation, "turnleft" for left rotation, "turnright" for right rotation, and "stop" for stop.



Finally click the Close_Camera button to close the camera.