

3. QR code recognition to control movement

3. QR code recognition to control movement

3.1. Experimental Objectives

3.2. Experimental code

3.3. Experimental Phenomenon

3.1. Experimental Objectives

We have a preliminary understanding of QR code technology and raspbot control hardware, and have realized the decoding of QR codes using OpenCV. Now, we try to combine QR codes with hardware. First, we use free online QR code making software to prepare 5 QR codes, each of which corresponds to a different function to realize the function of QR code control robot.



forward



back



left



right



stop

3.2. Experimental code

Source code path:

/home/pi/project_demo/08.AI_Visual_Interaction_Course/03.QRcode_recog_campaign/03_QRcode_recog_campaign.ipynb

```
#bgr8转jpeg格式 bgr8 to jpeg format
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
import sys
sys.path.append('/home/pi/project_demo/lib')
#导入麦克纳姆小车驱动库 Import Mecanum Car Driver Library
from McLumk_wheel_Sports import *
sys.path.append('/home/pi/software/oled_yahboom/')
from yahboom_oled import *
# 创建oled对象 Create an oled object
oled = Yahboom_OLED(debug=False)
# import the necessary packages
import numpy as np
import pyzbar.pyzbar as pyzbar
from PIL import Image
import ipywidgets.widgets as widgets
# from picamera2 import Picamera2, Preview
# import libcamera

image_widget = widgets.Image(format='jpeg', width=640, height=320)
```

```
def servo_reset():
    bot.Ctrl_Servo(1,90)
    bot.Ctrl_Servo(2,25)
servo_reset()
```

```
speed=60
def detect_control(info):
    if info == "forward":
        move_forward(speed)      #前进 forward
    elif info == "back":
        move_backward(speed)      #后退 back
    elif info == "left":
        rotate_left(speed)      #左旋 Rotate Left
    elif info == "right":
        rotate_right(speed)      #右旋 Rotate right
    elif info == "stop":
        stop_robot()      #停车 stop
```

```
display(image_widget)      #显示摄像头组件 Display
camera components
def decodeDisplay(image):
    barcodes = pyzbar.decode(image)
    for barcode in barcodes:
        # 提取二维码的边界框的位置 Extract the position of the bounding box of the QR
code
        # 画出图像中条形码的边界框 Draw the bounding box of the barcode in the image
```

```

(x, y, w, h) = barcode.rect
cv2.rectangle(image, (x, y), (x + w, y + h), (0, 225, 225), 2)

# 提取二维码数据为字节对象, 所以如果我们在输出图像上 Extract the QR code data
as a byte object and convert it into a string
# 画出来, 就需要先将它转换成字符串
datatemp=0
barcodeData = barcode.data.decode("utf-8")
barcodeType = barcode.type

# 绘出图像上条形码的数据和条形码类型 Draw the barcode data and barcode type on
the image
text = "{} ({}).format(barcodeData, barcodeType)
cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,
225, 225), 2)

# 向终端打印条形码数据和条形码类型 Print barcode data and barcode type to the
terminal
print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
if(barcodeData!=datatemp):
    oled.clear()
    qrtype_str=f'[INFO] Found {barcodeType}'
    qrcode_str=f'barcode: {barcodeData}'
    oled.add_line(qrtype_str, 1)
    oled.add_line(qrcode_str, 3)
    oled.refresh()
    datatemp=barcodeData
    detect_control(barcodeData)
return image

def detect():
    oled.init_oled_process() #初始化oled进程 Initialize oled process
    qrtype_str=f'[INFO] Found '
    qrcode_str=f'barcode: None'
    oled.add_line(qrtype_str, 1)
    oled.add_line(qrcode_str, 3)
    oled.refresh()
    camera = cv2.VideoCapture(0)
    camera.set(3, 320)
    camera.set(4, 240)
    camera.set(5, 30) #设置帧率 Setting the frame rate
    # fourcc = cv2.VideoWriter_fourcc(*"MPEG")
    # camera.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter.fourcc('M', 'J', 'P',
'G'))
    # camera.set(cv2.CAP_PROP_BRIGHTNESS, 40) #设置亮度 -64 - 64 0.0 Set
Brightness -64 - 64 0.0
    # camera.set(cv2.CAP_PROP_CONTRAST, 50) #设置对比度 -64 - 64 2.0 Set Contrast
-64 - 64 2.0
    # camera.set(cv2.CAP_PROP_EXPOSURE, 156) #设置曝光值 1.0 - 5000 156.0 Set the
exposure value 1.0 - 5000 156.0
    ret, frame = camera.read()

    # picam2 = Picamera2()
    # camera_config = picam2.create_preview_configuration(main=
{"format":'RGB888',"size":(320,240)})

```

```

# camera_config["transform"] = libcamera.Transform(hflip=1, vflip=1)
# picam2.configure(camera_config)
# picam2.start()
# frame = picam2.capture_array()
image_widget.value = bgr8_to_jpeg(frame)
try:
    while True:
        # 读取当前帧 Read the current frame
        ret, frame = camera.read()
        #frame = picam2.capture_array()
        # 转为灰度图像 Convert to grayscale image
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        im = decodedisplay(gray)
        image_widget.value = bgr8_to_jpeg(im)
        cv2.waitKey(5)
except:
    #picam2.stop_preview()
    camera.release()
    # 恢复屏幕基础数据显示 Restore basic data display on screen
    os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")

```

```

detect()
stop_robot()
oled.init_oled_process() #初始化oled进程 Initialize oled process
# 恢复屏幕基础数据显示 Restore basic data display on screen
os.system("python3 /home/pi/software/oled_yahboom/yahboom_oled.py &")
print("Ending")

```

3.3. Experimental Phenomenon

After the code block is run, we can put the QR code in front of the camera, and the car will move according to the content of the QR code.