

# k230 license plate recognition

---

## k230 license plate recognition

K230 and Raspberry Pi communication

1. Experimental Prerequisites
2. Experimental wiring
3. Main code explanation
4. Experimental Phenomenon

## K230 and Raspberry Pi communication

---

### 1. Experimental Prerequisites

This tutorial uses the Raspberry Pi 5 development board, and the corresponding routine path is [14.export\Raspberrypi-K230\17\_k230\_licence\_rec.py].

K230 needs to run the [14.export\CanmvIDE-K230\17.licence\_rec.py] program to start the experiment. It is recommended to download it as an offline program.

Things you need:

Windows computer

Raspberry Pi 5 development board

microUSB cable

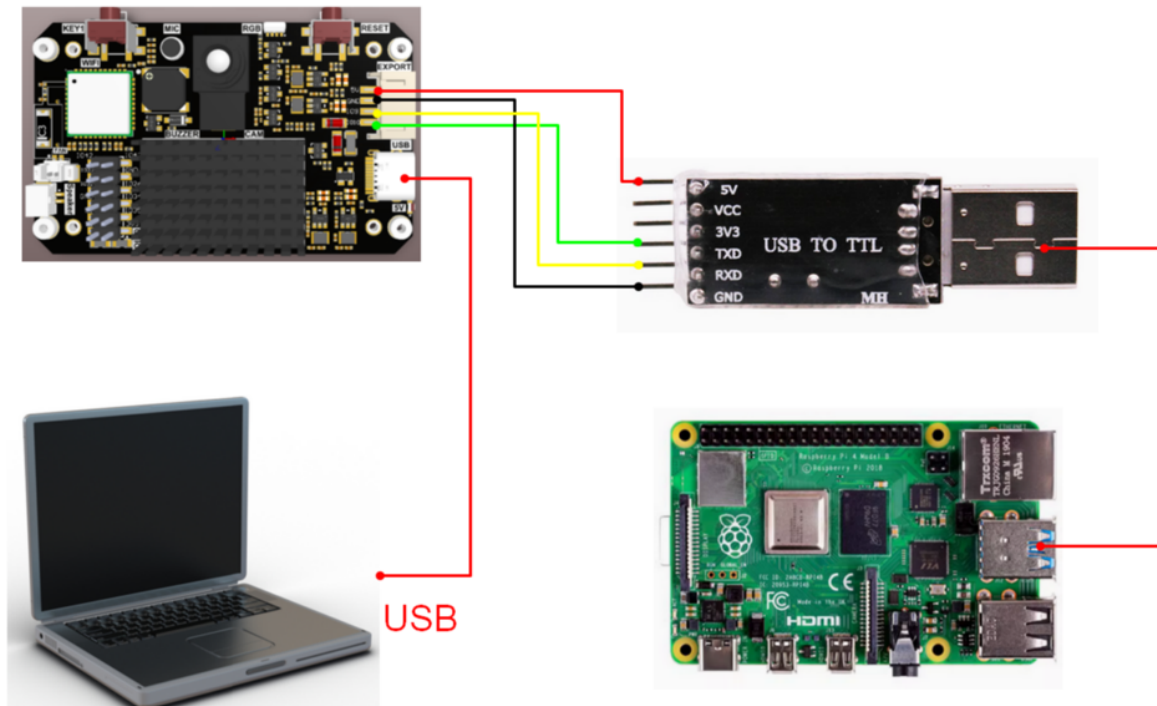
K230 visual module (including TF card with image burned in)

type-C cable

connection cable

### 2. Experimental wiring

| k230 vision module | USB to TTL module |
|--------------------|-------------------|
| 5V                 | VCC               |
| GND                | GND               |
| TXD(IO9)           | RxD               |
| RXD(IO10)          | TXD               |



### 3. Main code explanation

```
import serial

com="/dev/ttyUSB0"
ser = serial.Serial(com, 115200)

FUNC_ID = 17

def parse_data(data):
    if data[0] == ord('$') and data[len(data)-1] == ord('#'):
        data_list = data[1:len(data)-1].decode('utf-8').split(',')
        data_len = int(data_list[0])
        data_id = int(data_list[1])
        if data_len == len(data) and data_id == FUNC_ID:
            # print(data_list)
            result = data_list[2]
            return result
        elif (data_len != len(data)):
            print("data len error:", data_len, len(data))
        elif (data_id != FUNC_ID):
            print("func id error:", data_id, FUNC_ID)
        else:
            print("pto error", data)
    return ""

while True:
    if ser.in_waiting:
        data = ser.readline()
        # print("rx:", data)
        result = parse_data(data.rstrip(b'\n'))
        print("licence_rec:", result)
```

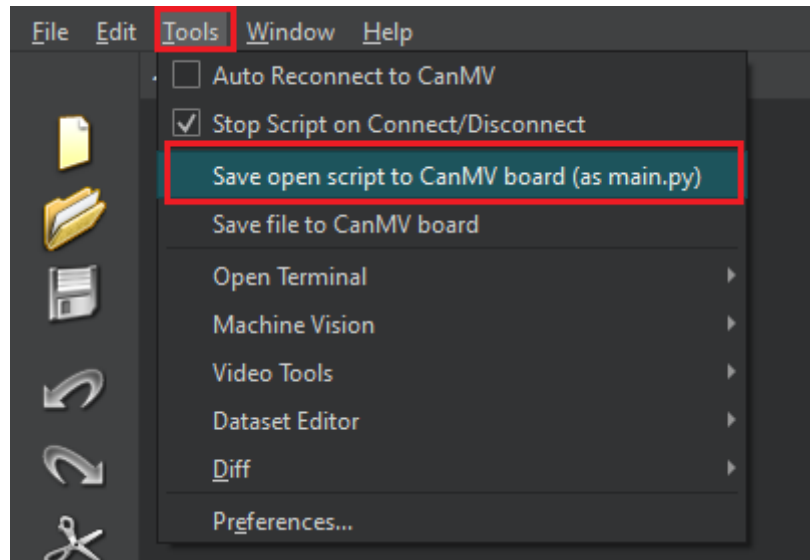
The above program is for parsing K230 data. Only when it complies with specific protocols can the corresponding data be parsed.

in

- licence\_rec: msg is the license plate character information.

## 4. Experimental Phenomenon

1. After connecting the cables, the k230 visual module runs offline. After K230 is connected to Canmv IDE, open the corresponding program, click [Save open script to CanMV board (as main.py)] on the toolbar, and then restart K230.



2. Transfer the program file to the system, open the terminal and enter the corresponding directory, then run the following command to start the program.

```
python3 17_k230_licence_rec.py
```

3. When the K230 camera image recognizes characters, the terminal will parse and print out the information transmitted by the K230.

in

- licence\_rec: msg is the license plate character information.

As shown in the figure below

[2025-04-30 12:21:23.917]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:23.980]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:24.042]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:24.104]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:24.152]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:24.232]# RECV ASCII>

licence\_rec:' 粤A88888'

[2025-04-30 12:21:24.278]# RECV ASCII>

licence\_rec:' 粤A88888'