

# k230 face recognition

---

## k230 face recognition

k230 and PICO2 communication

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

## k230 and PICO2 communication

---

### 1. Experimental Prerequisites

This tutorial uses the PICO2 development board, and the corresponding routine path is [14.export\PICO-K230\08\_pico\_k230\_face\_recognition.py].

K230 needs to run the [14.export\CanmvIDE-K230\08.face\_recognition.py] program to start the experiment. It is recommended to download it as an offline program. Since the face recognition function requires the face information to be registered in advance, please refer to the [7. Face Recognition\7. Registering Face Recognition] course first, and then perform this experiment after registering the face.

Things you need:

Windows computer

PICO2 development board

microUSB cable

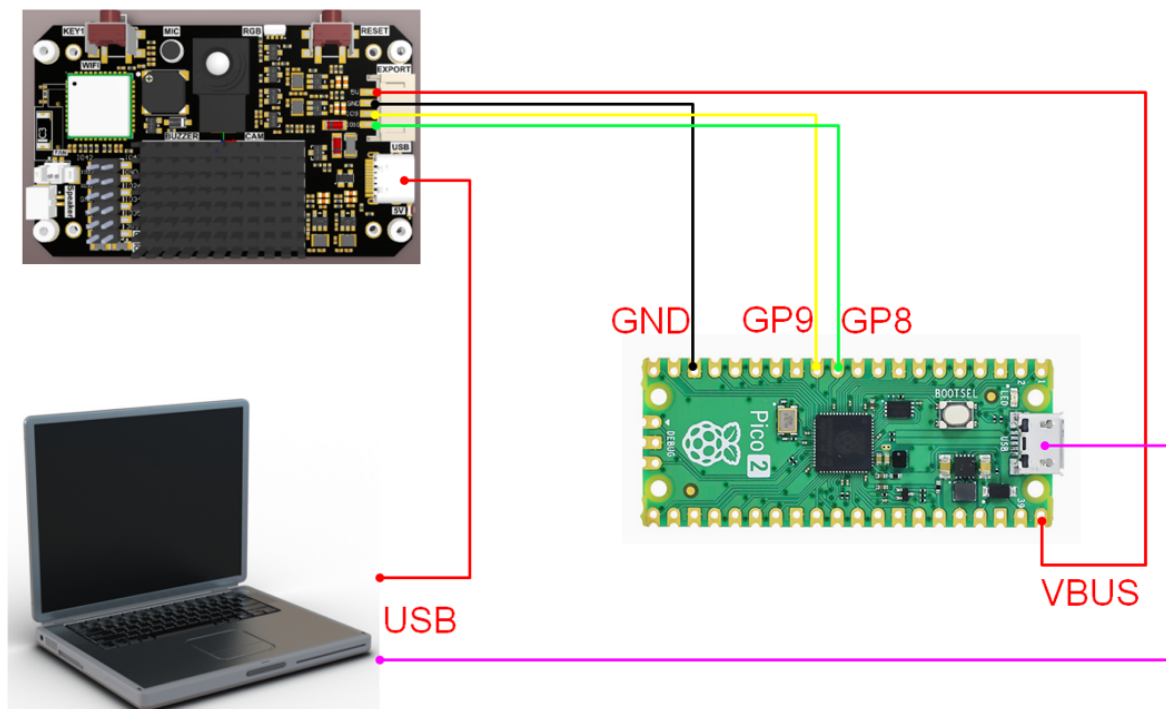
K230 visual module (including TF card with image burned in)

type-C cable

connection cable

### 2. Experimental wiring

k230 vision module	PICO2 Development Board
5V	VCC
GND	GND
TXD(IO9)	RXD(GP9)
RXD(IO10)	TXD(GP8)



### 3. Main code explanation

```
from machine import UART, Pin

FUNC_ID = 8

uart1 = UART(1, baudrate=115200, tx=Pin(8), rx=Pin(9), bits=8, parity=None,
stop=0)

print("hello yahboom")

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)
            x = int(data_list[2])
            y = int(data_list[3])
            w = int(data_list[4])
            h = int(data_list[5])
            msg = data_list[6]
            score = int(data_list[7])
            return x, y, w, h, msg, score
        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)
        return -1, -1, -1, -1, "", -1

last_data = bytearray()
while True:
    if uart1.any() > 0:
```

```

cur_data = uart1.readline()
# print("rx:", cur_data)
if ord('\n') in cur_data:
    # data = bytearray(last_data + cur_data.decode('utf-8'), 'utf-8')
    data = last_data + cur_data
    last_data = bytearray()
    x, y, w, h, msg, score = parse_data(data.rstrip(b'\n'))
    print("face recognition:x:%d, y:%d, w:%d, h:%d, msg:%s, score:%d" %
(x, y, w, h, msg, score))
else:
    last_data = last_data + cur_data

```

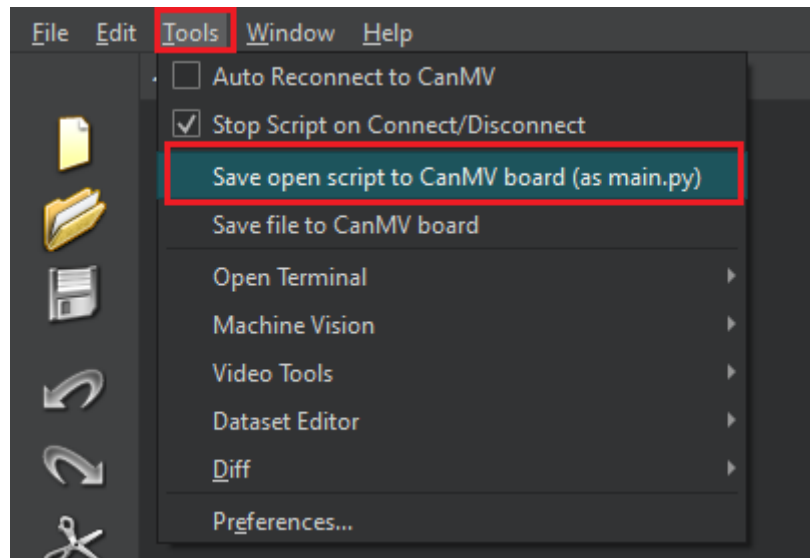
The above program is for parsing K230 data. Only when it complies with specific protocols can the corresponding data be parsed.

in

- x: is the horizontal coordinate of the upper left corner of the recognized box
- y: is the vertical coordinate of the upper left corner of the recognized box
- w: is the width of the recognized frame
- h: is the length of the recognized frame
- name: is the name of the person who is identified, if it is not registered then it is 'unknown'
- score: is the score for identifying people

## 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. Open the Thonny editor, connect the PICO2 mainboard, open the program file and run it.  
Note: The PICO2 mainboard needs to have the microPython firmware downloaded in advance.
3. When the K230 camera image recognizes a face, the terminal will parse and print out the information transmitted by the K230.

in

- x: is the horizontal coordinate of the upper left corner of the recognized box

- y: is the vertical coordinate of the upper left corner of the recognized box
- w: is the width of the recognized frame
- h: is the length of the recognized frame
- name: is the name of the person who is identified, if it is not registered then it is 'unknown'
- score: is the score for identifying people

As shown in the figure below

```
[2025-04-30 11:49:17.394]# RECV ASCII>
face recognition:x:252, y:116, w:92, h:158, name:'peter', score:0.89

[2025-04-30 11:49:17.458]# RECV ASCII>
face recognition:x:261, y:120, w:90, h:157, name:'peter', score:0.86

[2025-04-30 11:49:17.697]# RECV ASCII>
face recognition:x:279, y:120, w:94, h:160, name:'peter', score:0.82
face recognition:x:295, y:117, w:92, h:158, name:'peter', score:0.85
face recognition:x:299, y:114, w:79, h:133, name:'peter', score:0.83
face recognition:x:294, y:100, w:81, h:140, name:'peter', score:0.83

[2025-04-30 11:49:17.760]# RECV ASCII>
face recognition:x:265, y:81, w:76, h:126, name:'peter', score:0.69
```