

ard_K210 Autonomous Learning Classification

ard_K210 Autonomous Learning Classification

1.K210 and Arduino communication

1.1 Experimental premises

1.2 Experimental wiring

1.3 Main code explanation

1.4 experimental phenomena

1.K210 and Arduino communication

1.1 Experimental premises

This tutorial uses arduino, and K210 requires running the program in **K210-AI (stm32_pico_arduino)** to start the experiment

arduino *1

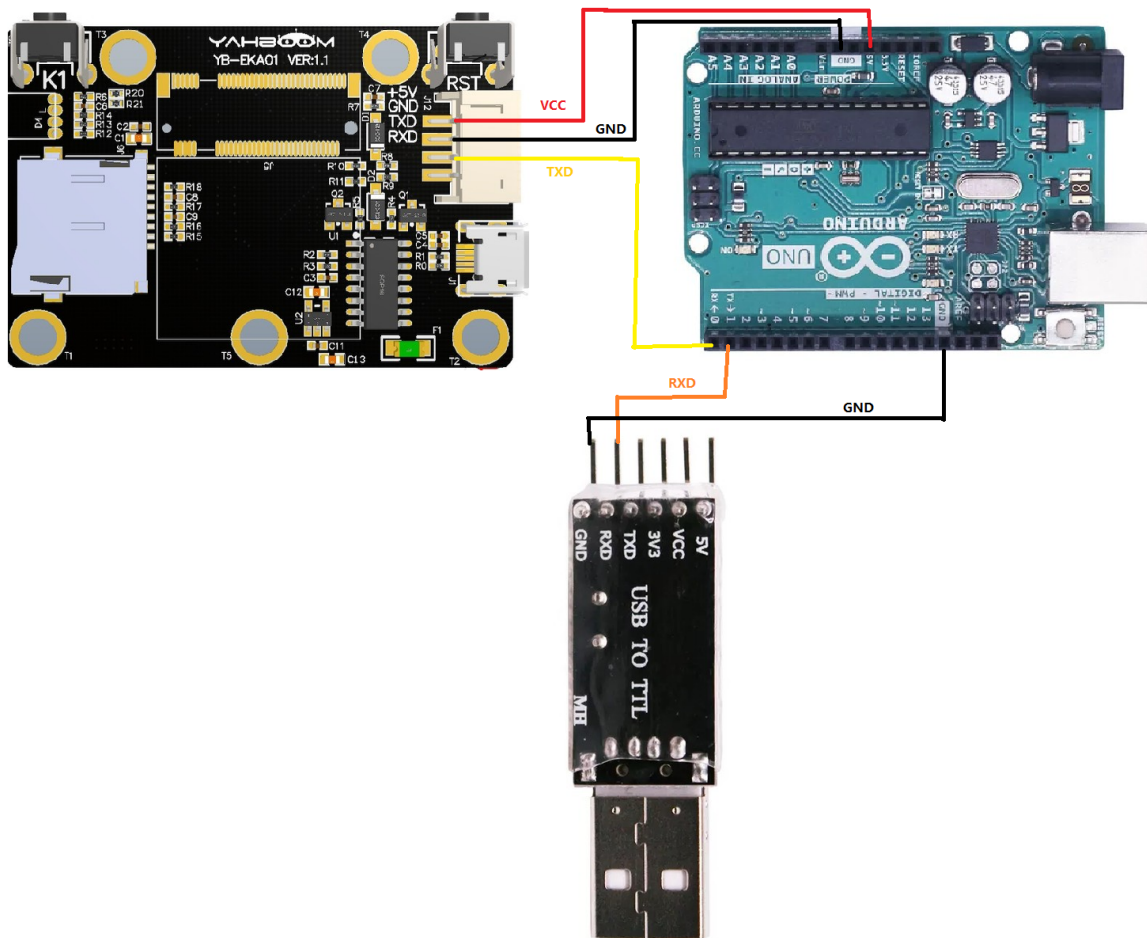
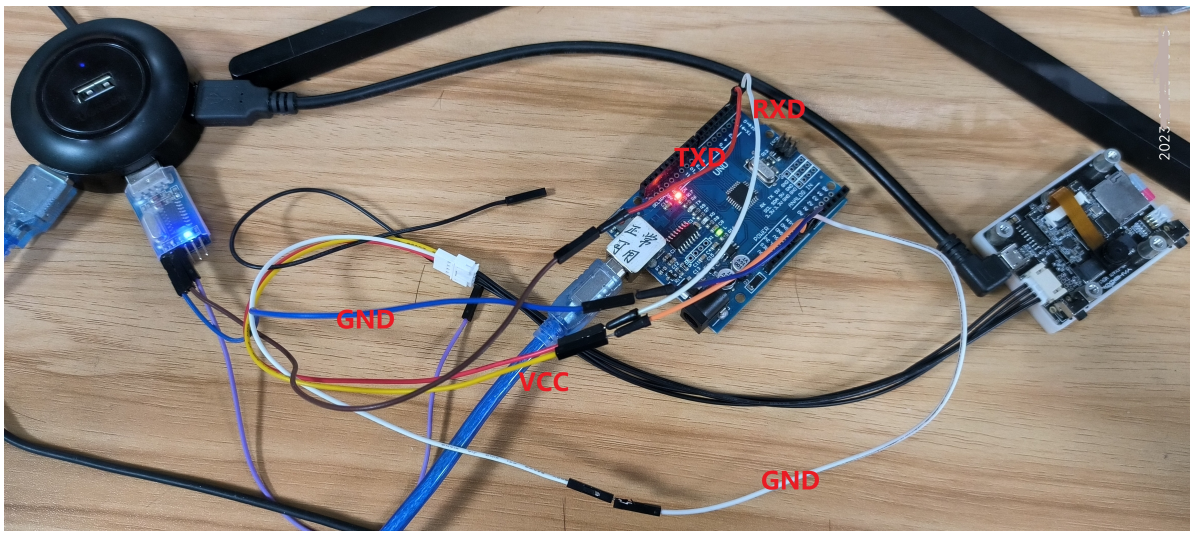
K210 perspective module * 1 (requires SD card (with AI model inside) and camera)

USB to TTL module * 1

1.2 Experimental wiring

arduino	usb to ttl
TXD	RXD
GND	GND

arduino	K210 perspective module
RXD	TXD
GND	GND
VCC	5V
Wiring as shown in the diagram :	



1.3 Main code explanation

```
void loop()
{
  while (K210Serial1.available())
  {
    recv_k210msg(K210Serial1.read());

    if (k210_msg.class_n != 0)
    {
      if(k210_msg.class_n == 10)
      {
```

```

        sprintf(buff_com,"id = %c\r\n",k210_msg.id);
        k210Serial.print(buff_com);

        k210_msg.class_n = 0;
    }

}

}
}

```

After the above program, if you are running this routine, k210_ The members of the msg structure have corresponding values and are processed through serial port printing

K210_Msg: is a structure that receives information, and its main members are

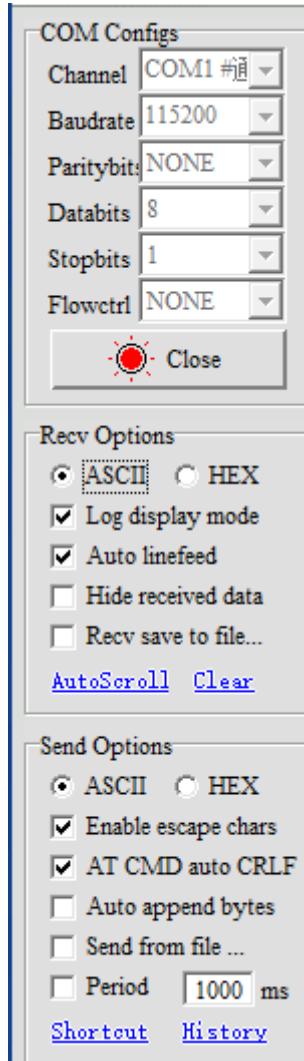
- X: is the horizontal coordinate of the top left corner of the recognized box (range: 0-240)
- Y: is the vertical coordinate of the upper left corner of the identified box (range: 0-320)
- W: is the width of the recognized box (range: 0-240)
- H: The length of the recognized box (range: 0-320)
- ID: is the recognized label
- Class_n: Routine number
- Msg_Msg [20]: Valid data

After receiving and processing data, k210_ Each member of the msg will store valid information. If you want to develop it again, call K210 directly_ Members of msg are sufficient

1.4 experimental phenomena

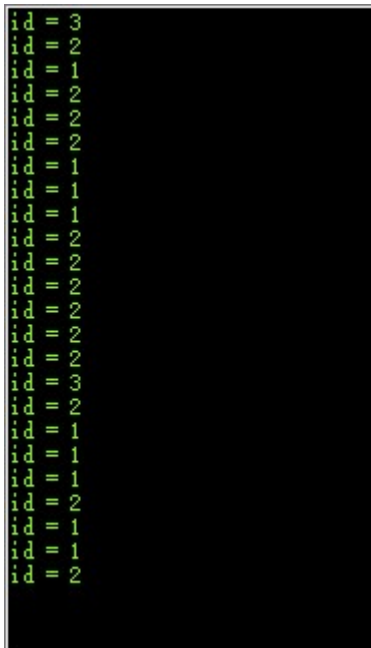
1. After connecting the cable, the K210 perspective module runs offline. Please check 【6.2 K210 as coprocessor】 -- 【ReadMe】

2. Set the serial port assistant to the interface shown in the figure



The image shows a serial port configuration window. The 'COM Configs' section includes dropdown menus for Channel (COM1), Baudrate (115200), Paritybits (NONE), Databits (8), Stopbits (1), and Flowctrl (NONE). Below these is a 'Close' button with a red circular icon. The 'Recv Options' section has radio buttons for ASCII (selected) and HEX, and checkboxes for Log display mode, Auto linefeed, Hide received data, and Recv save to file... (all unchecked). It also features 'AutoScroll' and 'Clear' links. The 'Send Options' section has radio buttons for ASCII (selected) and HEX, and checkboxes for Enable escape chars, AT CMD auto CRLF, Auto append bytes, Send from file ..., and Period (1000 ms). It includes 'Shortcut' and 'History' links.

3. Then run the object detection routine, and the serial assistant will print out the important information transmitted from k210 to stm32, as shown in the following figure



```
id = 3
id = 2
id = 1
id = 2
id = 2
id = 2
id = 1
id = 1
id = 1
id = 2
id = 2
id = 2
id = 2
id = 2
id = 2
id = 2
id = 3
id = 2
id = 1
id = 1
id = 1
id = 2
id = 1
id = 1
id = 2
```

Object detection only transmits k210_ The id of the msg is a Member variable.

ID: Refers to the object ID when taking a photo, and the range of IDs (1-3)

