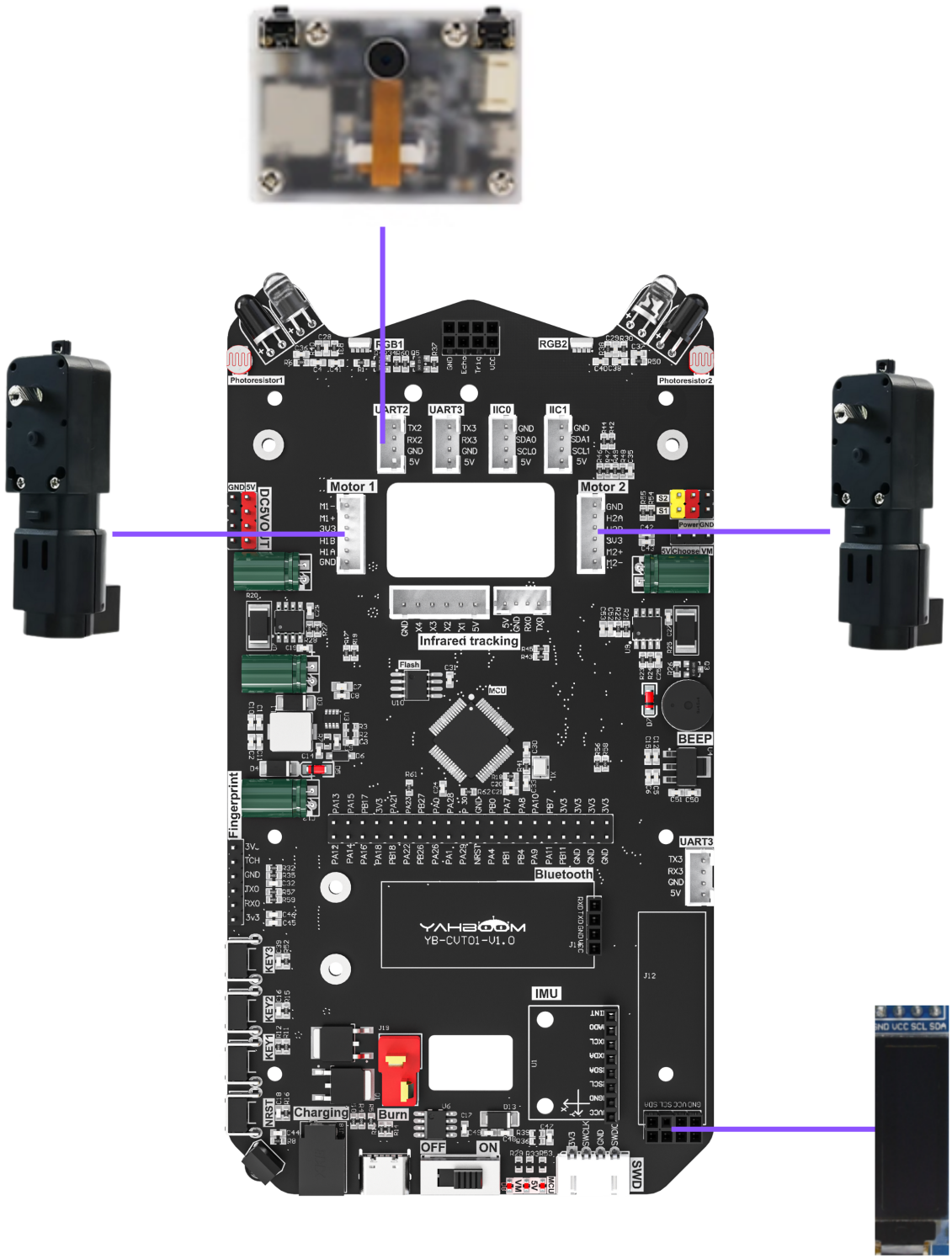
K210 Color Response

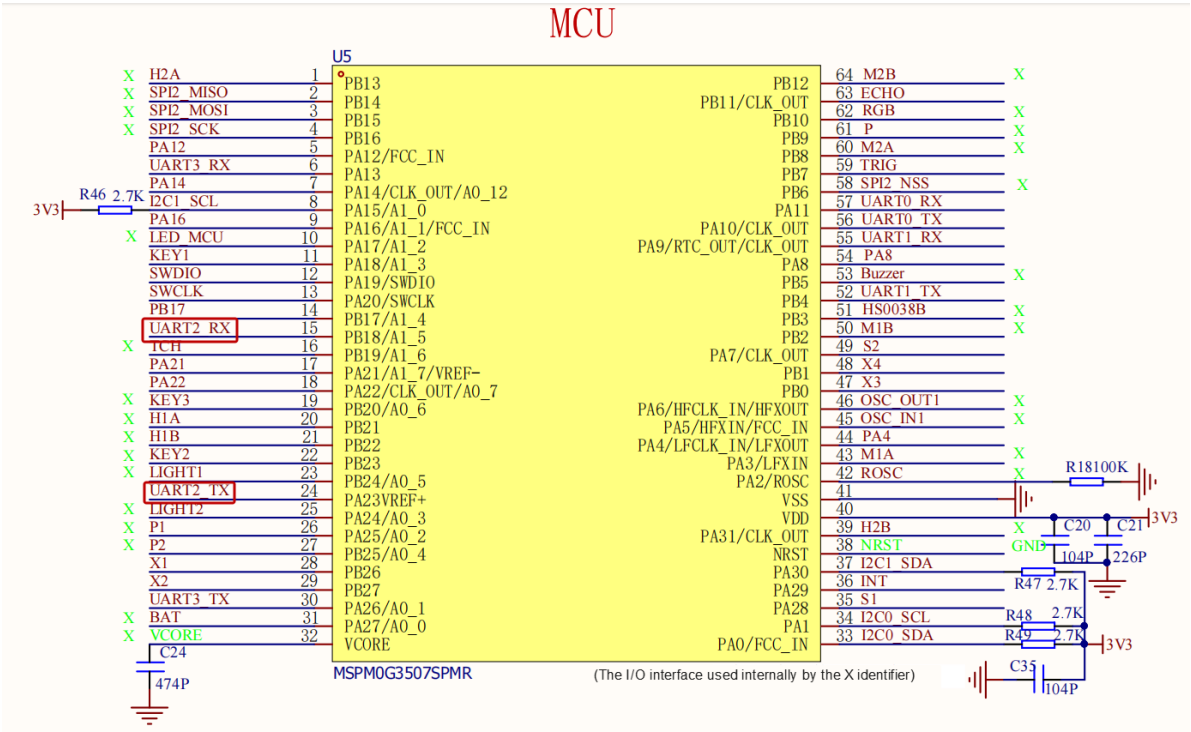
1. Hardware Connection

K210 Vision Module	MSPM0G3507
5V	5V
GND	GND
TX	RX2
RX	TX2

Physical Connection

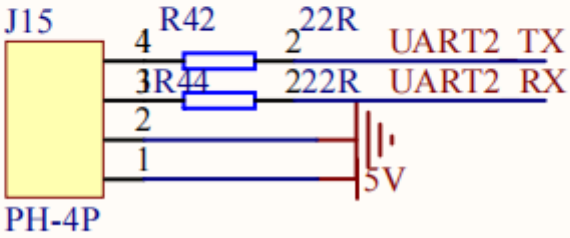


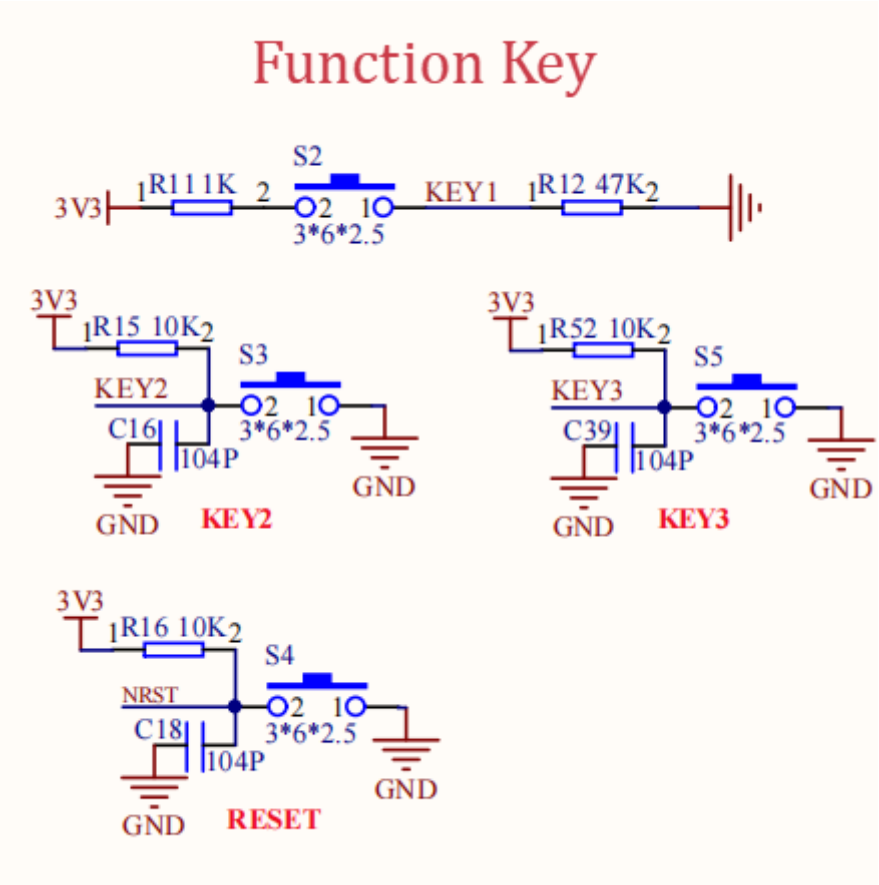
Schematic Diagram



UART 2

Vision module interface
K210/K230 module interface

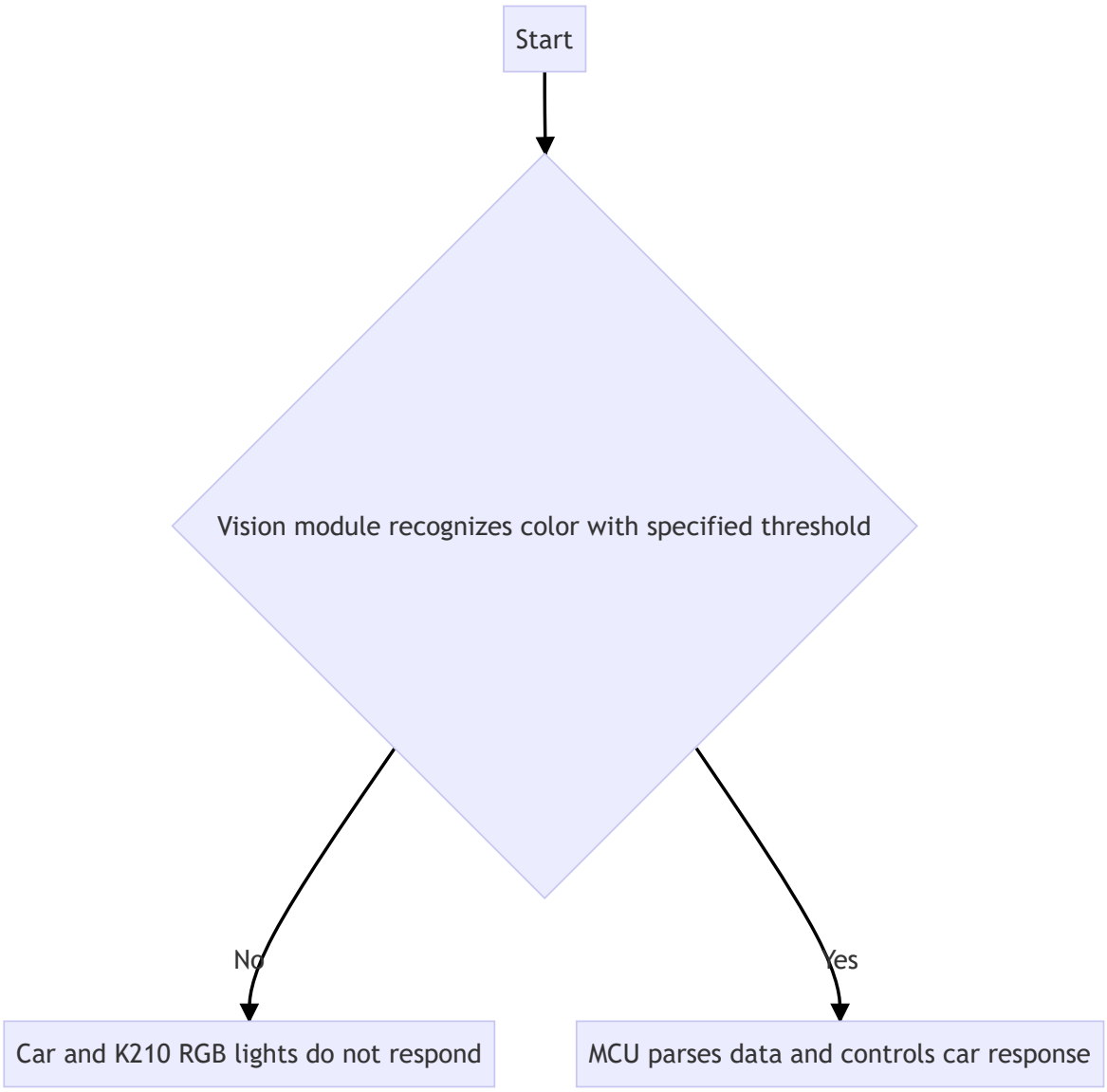




K210 Protocol

Header 1	Header 2	Length	Function Code	ID(0-14|0xff)	R(0-255)	G(0-255)	B(0-255)	checksum
0xFF	0xFC	0x07	0x05	XX ignored not used	XX	XX	XX	XX

Control Flow Chart



2. Code Analysis

revaction.c

```
// 数据分析 / Data analysis
void Upper_Data_Parse(uint8_t *data_buf, uint8_t num)
{
    uint8_t func_id = *(data_buf + 3);
    switch (func_id)
    {
    /* 判断功能字：小车速度设置 / Determine function code: car speed setting */
    case FUNC_MOTION:
    {
        uint8_t parm = (uint8_t) *(data_buf + 4);
```

```
        int16_t Vx_recv = *(data_buf + 6) << 8 | *(data_buf + 5);
        int16_t Vy_recv = *(data_buf + 8) << 8 | *(data_buf + 7);
        int16_t Vz_recv = *(data_buf + 10) << 8 | *(data_buf + 9);
        uint8_t adjust = parm & 0x80;

        if (Vx_recv == 0 && Vy_recv == 0 && Vz_recv == 0)
        {
            Motor_Stop(1);
            g_start_ctrl =0;
        }
        else
        {
            Motion_Car_Control(Vx_recv, Vy_recv, -Vz_recv);
        }
        break;
    }

    /* 判断功能字：彩灯控制 / Determine function code: RGB light control*/
    case FUNC_RGB:
    {
        u8 red = *(data_buf + 5);
        u8 green = *(data_buf + 6);
        u8 blue = *(data_buf + 7);
        rgb_SetRGB(WS2812_MAX, red, green, blue);
        rgb_SendArray();
        break;
    }

    default:
        break;
    }
}
```

K210 Partial Source Code

```
# 各个颜色的阈值以及颜色列表 / Threshold values for each color and color list
#红色 color red
threshold_red = (48, 71, 47, 80, 61, -68)
#绿色 color green
threshold_green = (12, 79, -78, -10, -66, 109)
#蓝色 color blue
threshold_blue = [37, 64, 9, 49, -57, -37]
#黄色 color yellow
threshold_yellow = (98, 78, -66, -6, -36, 121)
#白色 color white
threshold_white = (94, 83, 44, -23, 28, -54)
#紫色 color purple
threshold_purple = [75, 78, 66, 71, -40, -36]
color_list = [threshold_red, threshold_green, threshold_blue,
threshold_yellow,threshold_white,threshold_purple]

while(True):
    clock.tick()
    img = sensor.snapshot()
    fps = clock.fps()
    index = 0
    led = 0
```

```
#遍历各个颜色列表的颜色阈值，如果有找到相近色块，将预设的RGB值发送到小车 / Traverse color thresholds in the color list, if similar color blocks are found, send preset RGB values to the car
    for t in color_list:
        index = index + 1
        for blob in img.find_blobs([t], pixels_threshold=100, area_threshold=100, merge=True, margin=10):
            img.draw_rectangle(blob.rect())
            img.draw_cross(blob.cx(), blob.cy())
            print_color(index)
            led = led + 1
            led_on = 1
    if led == 0:
        if led_on == 1:
            bot.set_colorful_lamps(0xff, 0, 0, 0)
            RGB.set(0, 0, 0)
            led_on = 0
    img.draw_string(0, 0, "%2.1ffps" %(fps), color=(0, 60, 128), scale=2.0)
    lcd.display(img)
```

3. Main Functions

Upper_Data_Parse

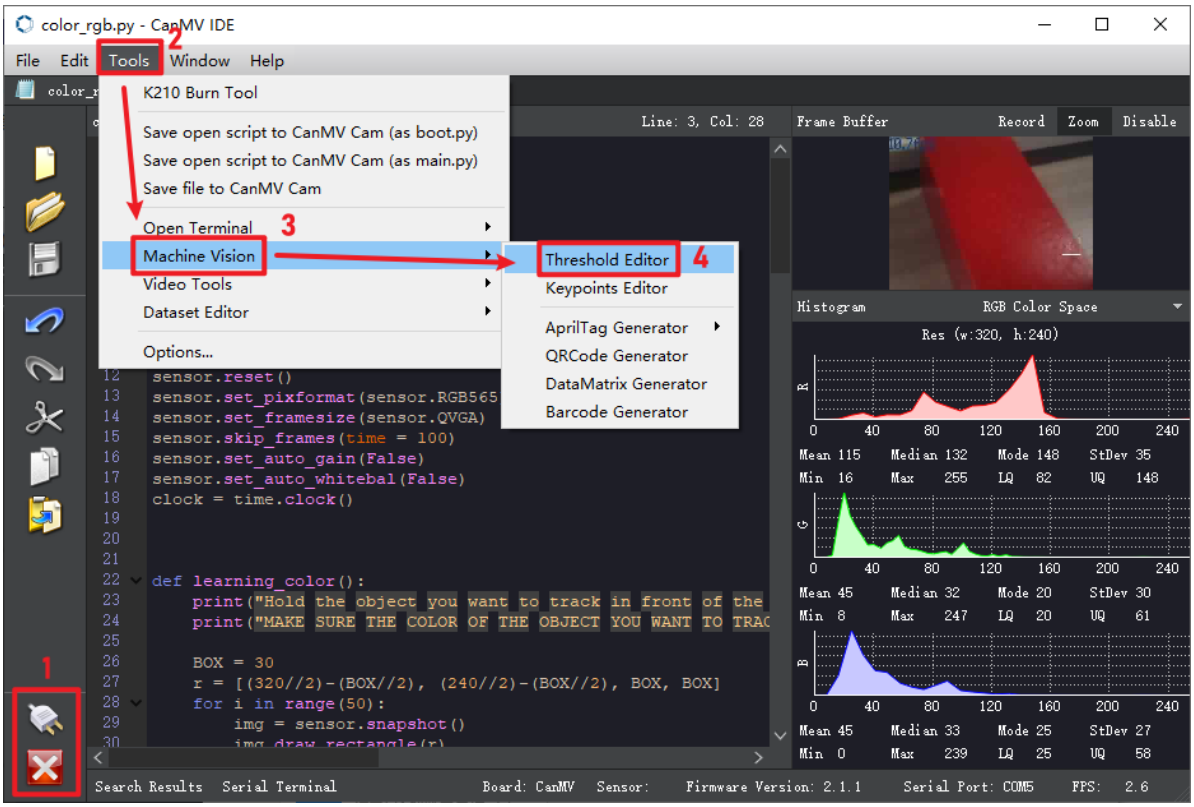
Function Prototype	**void Upper_Data_Parse(uint8_t *data_buf, uint8_t num)**
Function Description	Host computer data parsing function: processes different instructions based on function code (func_id) in data buffer. When function code is FUNC_MOTION, parses speed parameters (Vx_recv, Vy_recv, Vz_recv) and controls car movement (stops motors when speed is 0); when function code is FUNC_RGB, parses RGB color components, sets RGB light color and sends data
Input Parameters	data_buf: pointer to received data buffer num: data length (not used in function)
Return Value	None

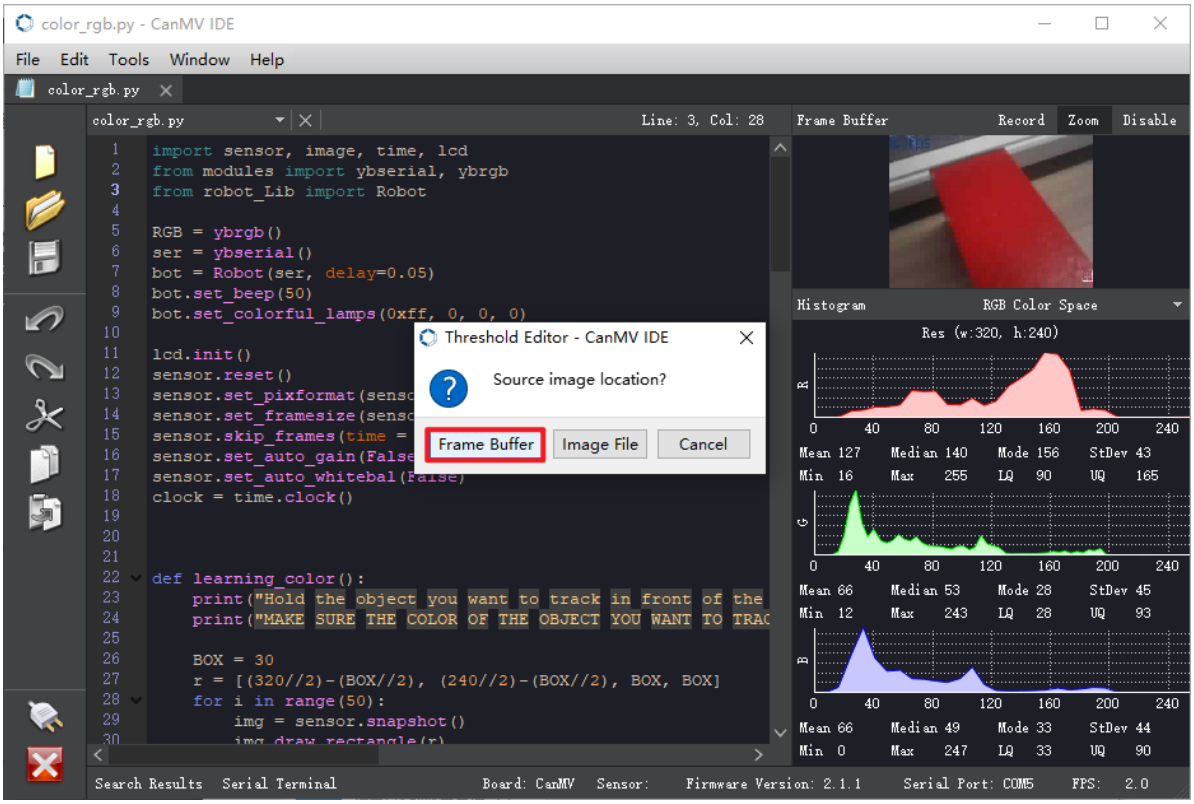
Upper_Data_Receive

Function Prototype	**void Upper_Data_Receive(uint8_t Rx_Temp)**
Function Description	Host computer data receiving function: receives data in fixed format (with PTO_HEAD as header, PTO_DEVICE_ID as device ID), stores it in RxBuffer, and sets New_CMD_flag when reception is complete
Input Parameters	Rx_Temp: received single byte data
Return Value	None

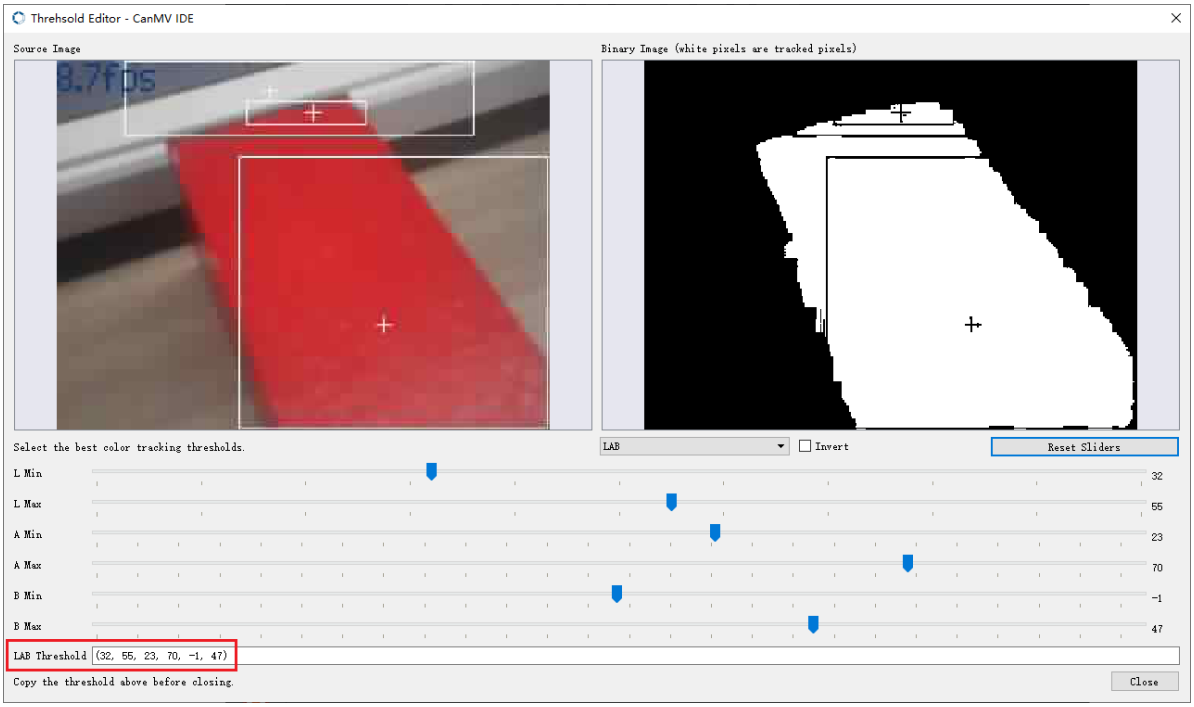
4. K210 Method for Extracting Specified Color LAB Thresholds

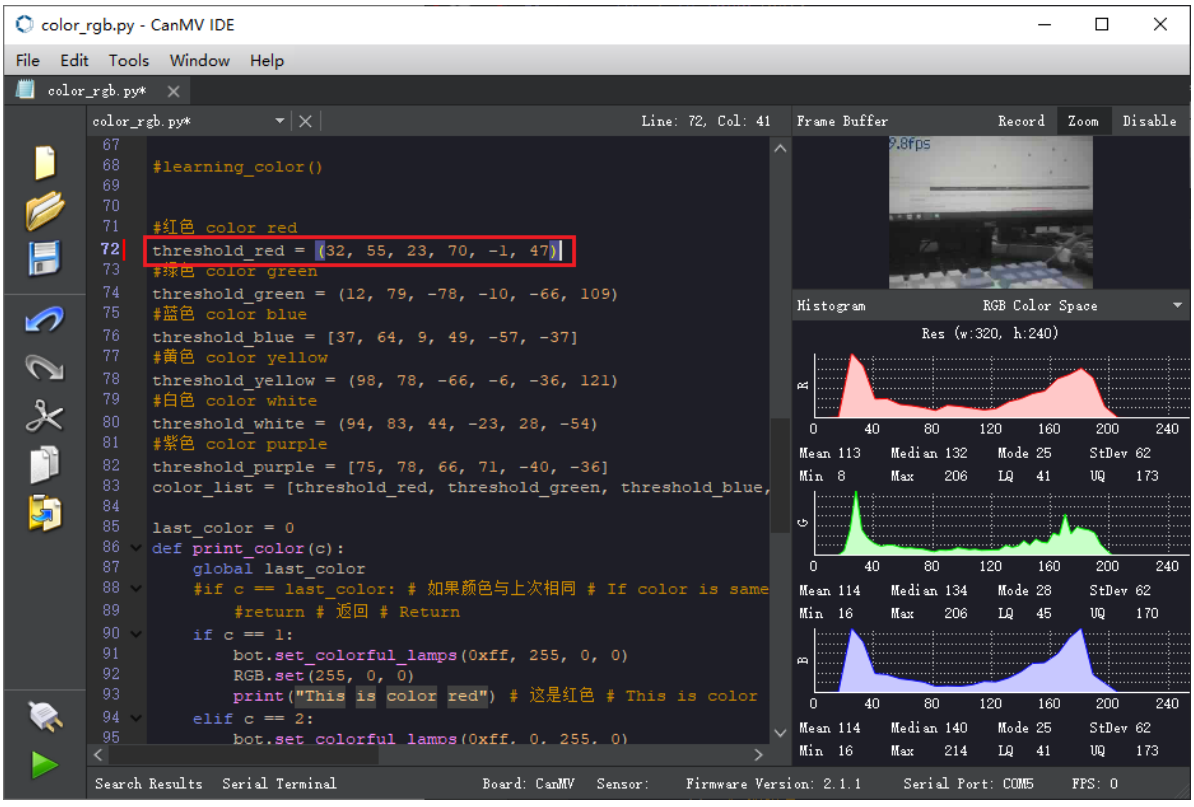
After connecting to CanMV IDE via microUSB, drag follow_color_k210.py to CanMV IDE to open it. First click the run button to ensure the color to be recognized appears in the frame buffer. Here we use red as an example. Click Tools -> Machine Vision -> Threshold Editor in the menu bar to get the image from the frame buffer.





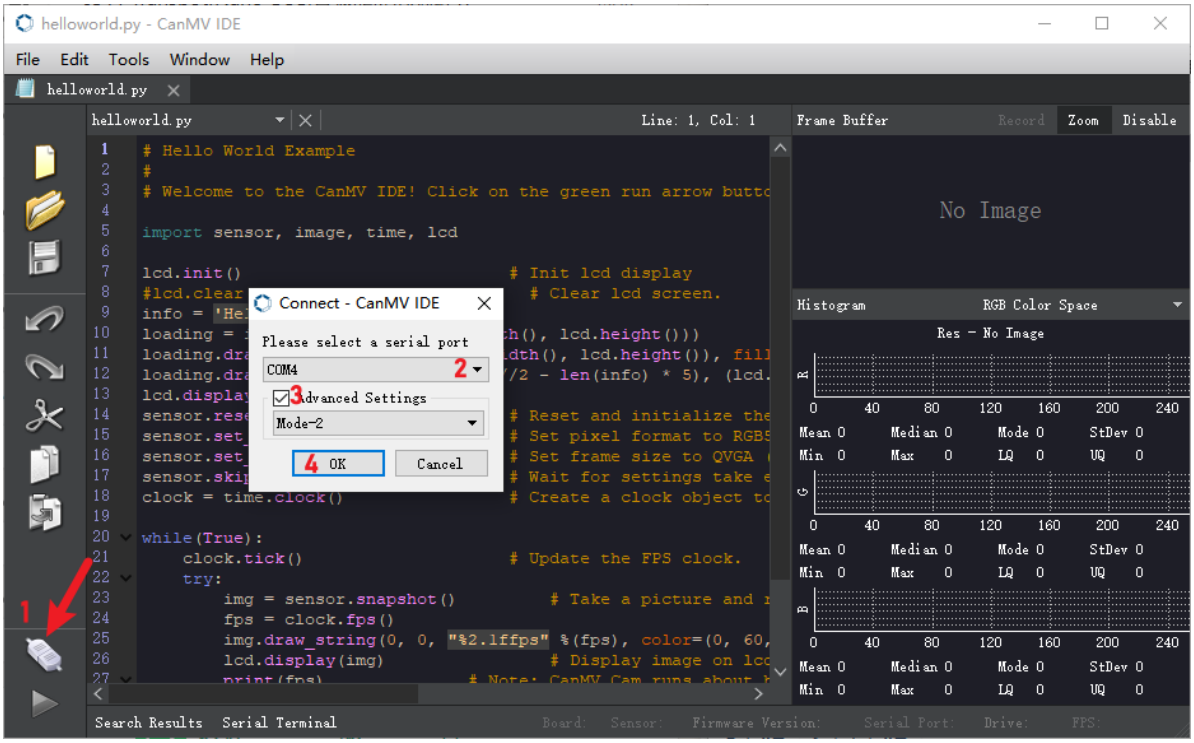
You can drag the slider positions to adjust the threshold. White pixels are the tracked pixels. Finally, fill the LAB threshold values into our color thresholds.



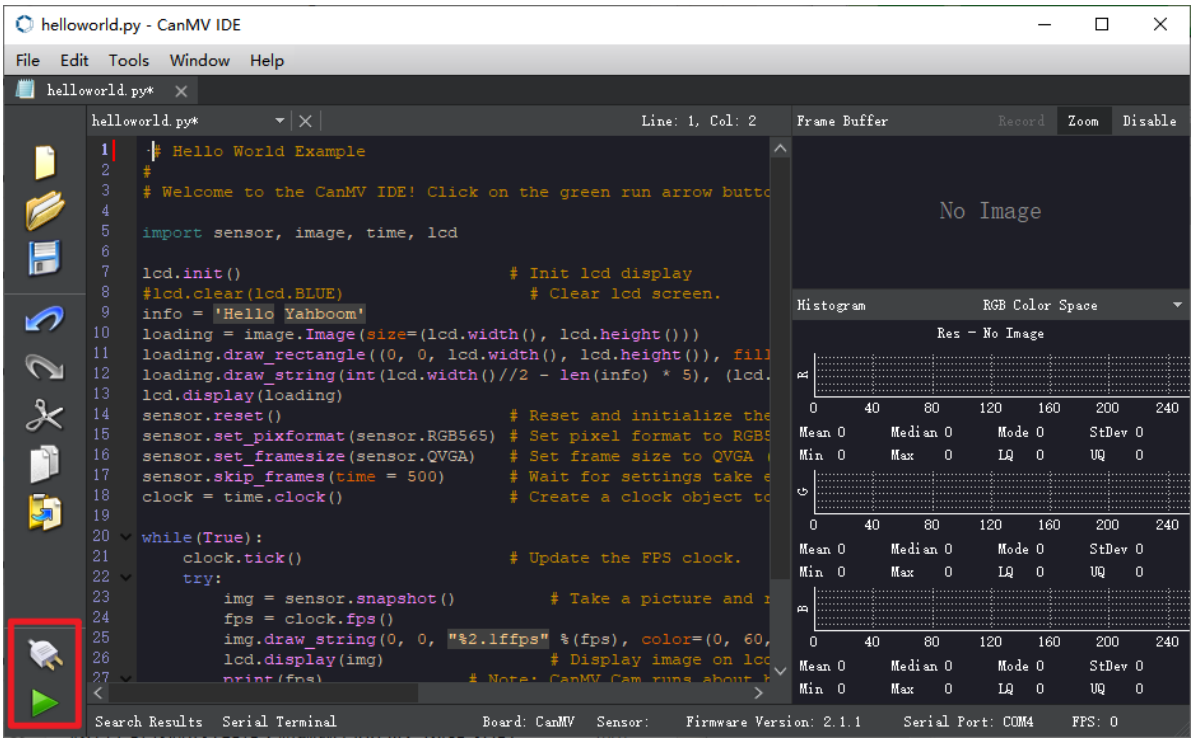


5. K210 Program Burning

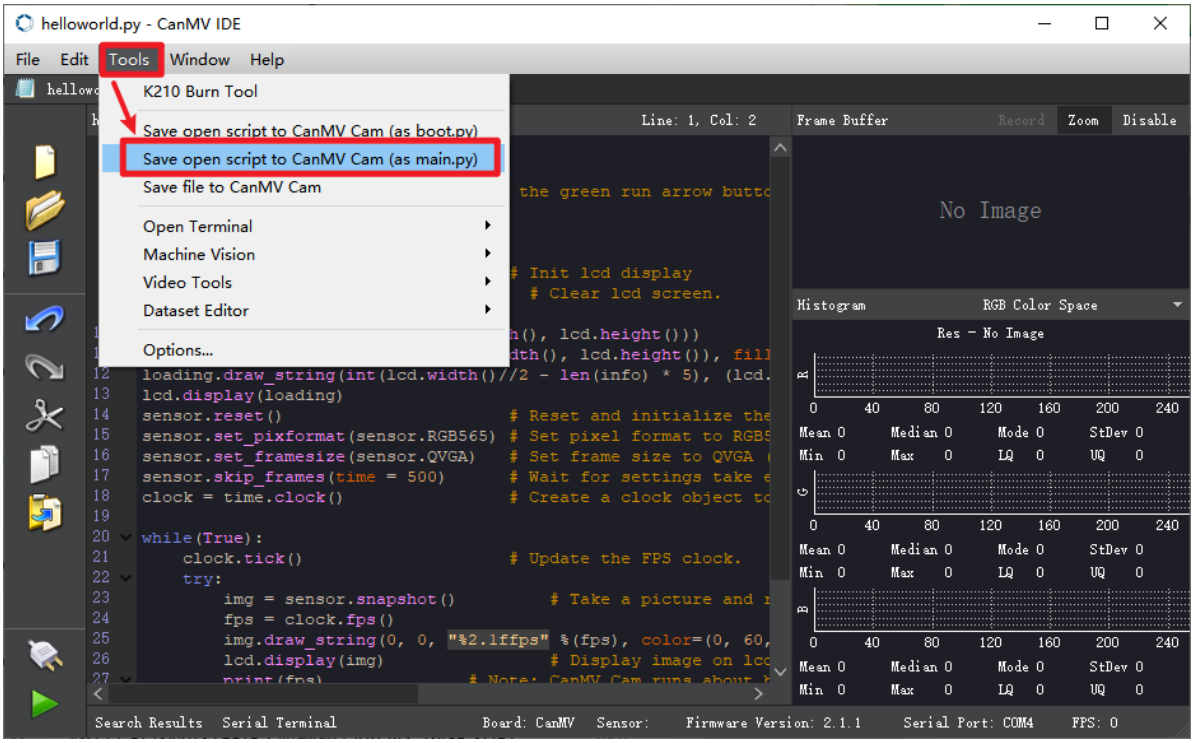
After downloading and opening CanMV IDE, we need to first drag the color_rgb.py file from the k210 source code provided in this course to CanMV IDE to open it, then connect the IDE, **here using helloworld.py as example**

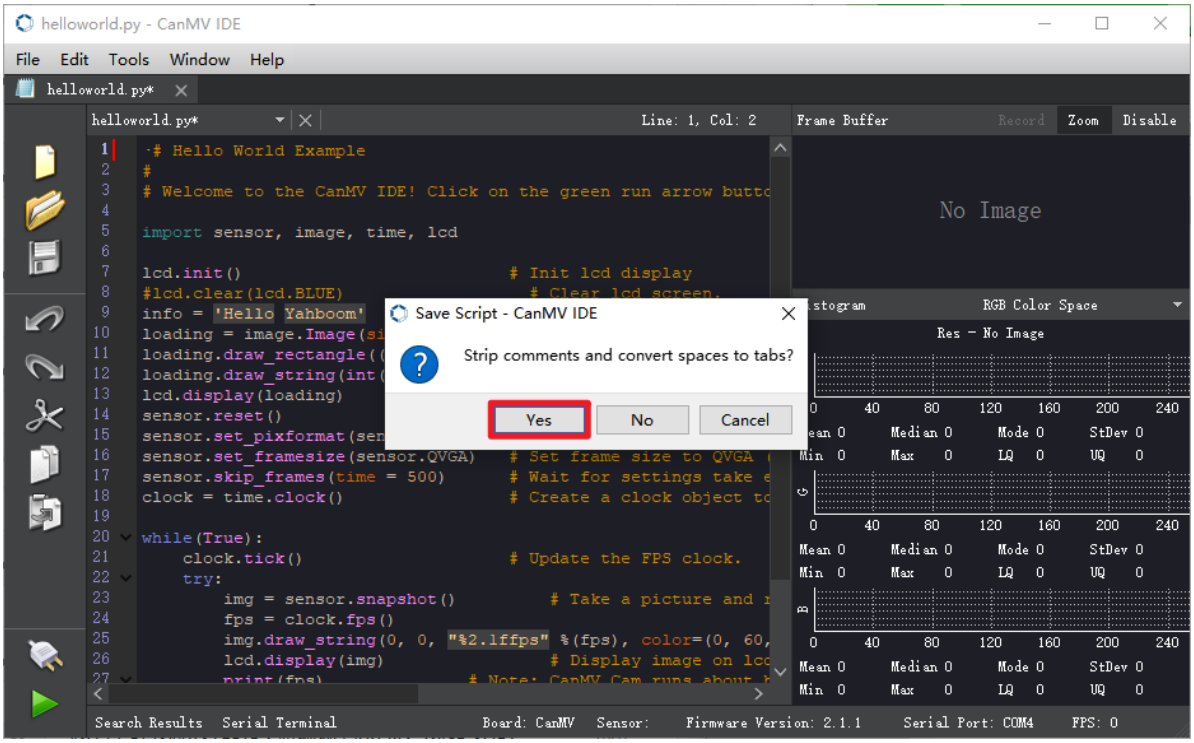


After successful IDE connection, the phenomenon is as follows

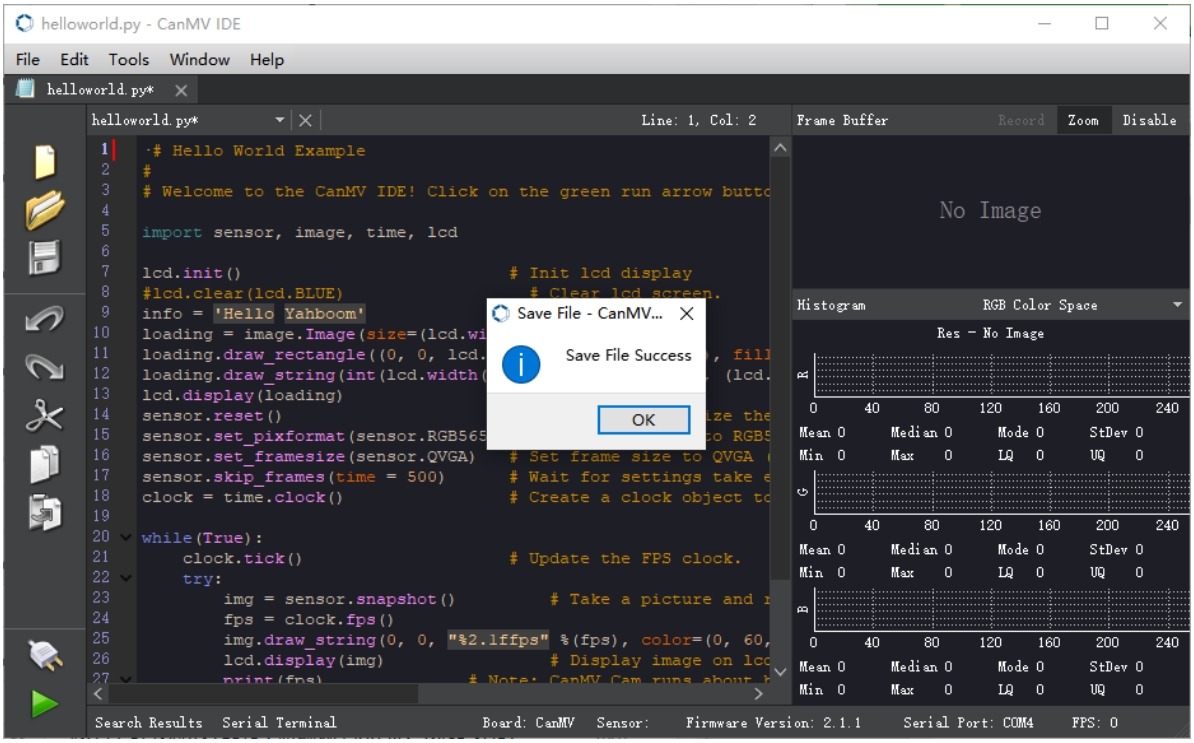


Here we use helloworld.py as an example, open the top menu bar Tools -> Save currently open script as (main.py) to CanMV Cam





Both Yes/No can be selected here. When the following status appears, the write is successful.



6. Experimental Phenomena

After setting the color thresholds and burning the program to K210, place colored objects in front of K210. The car and K210 RGB lights will light up corresponding colors following the object's color.