

K210 Handwritten Digit Action

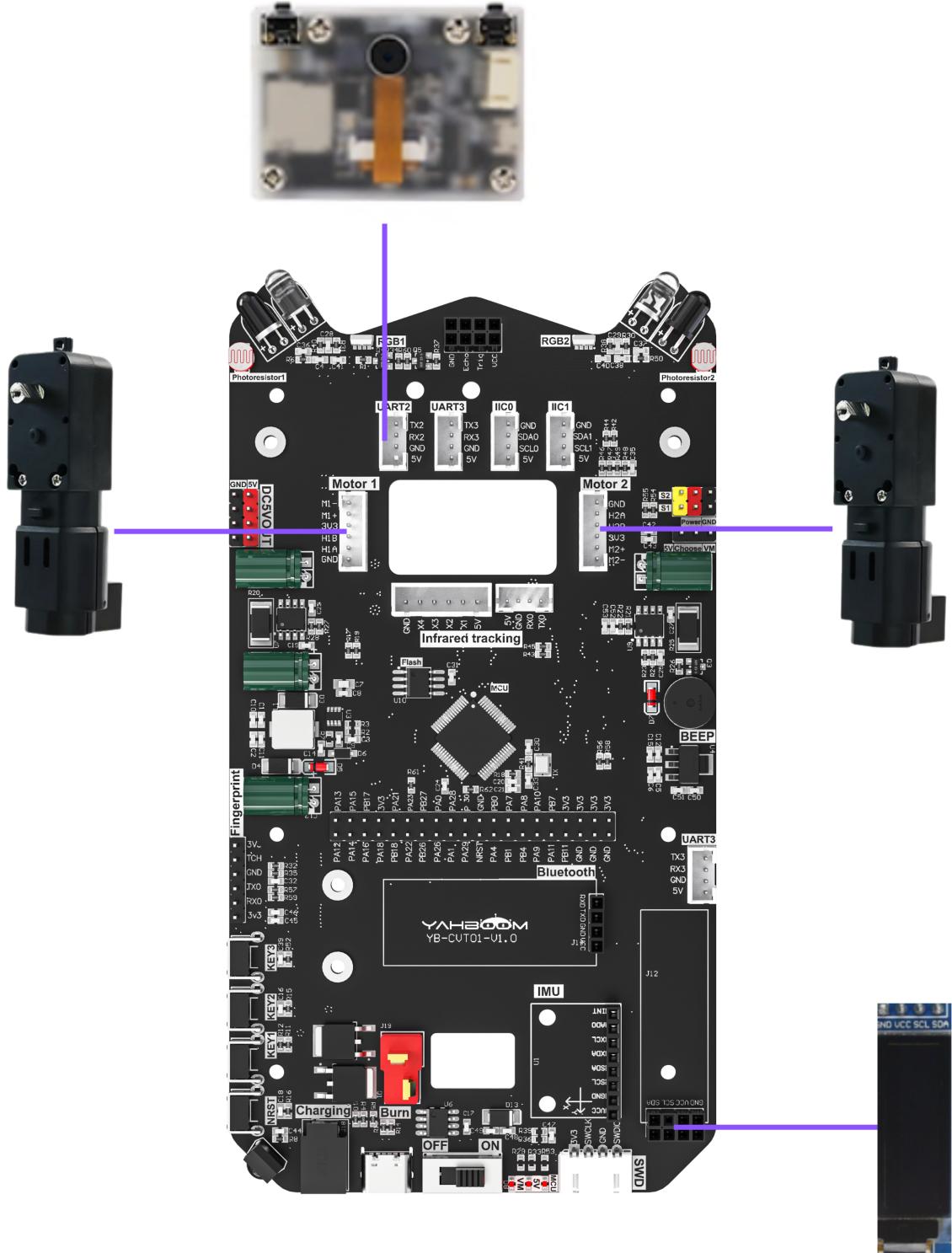
K210 Handwritten Digit Action

1. Hardware Connection
2. Code Analysis
3. Main Functions
4. K210 Program Burning
5. Experimental Phenomena

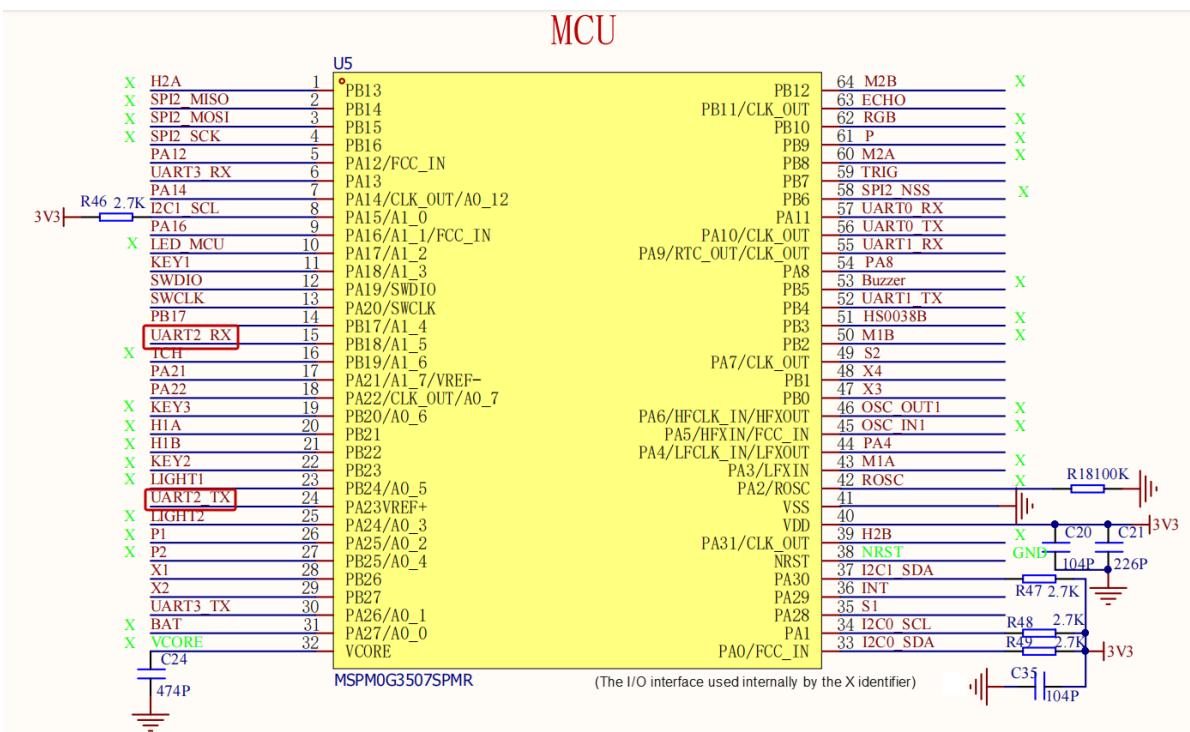
1. Hardware Connection

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

Physical Connection

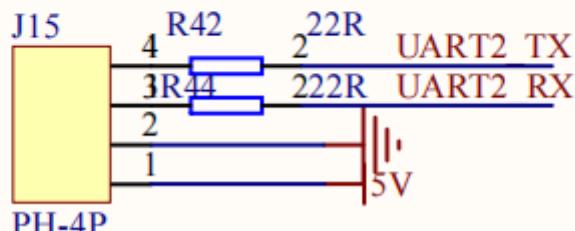


Schematic Diagram

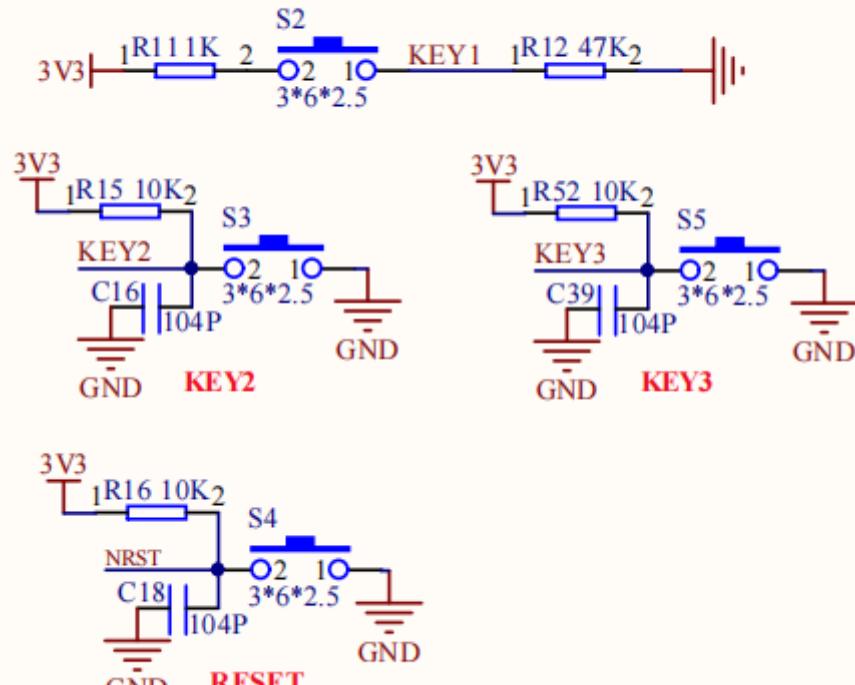


UART 2

Vision module interface K210/K230 module interface



Function Key



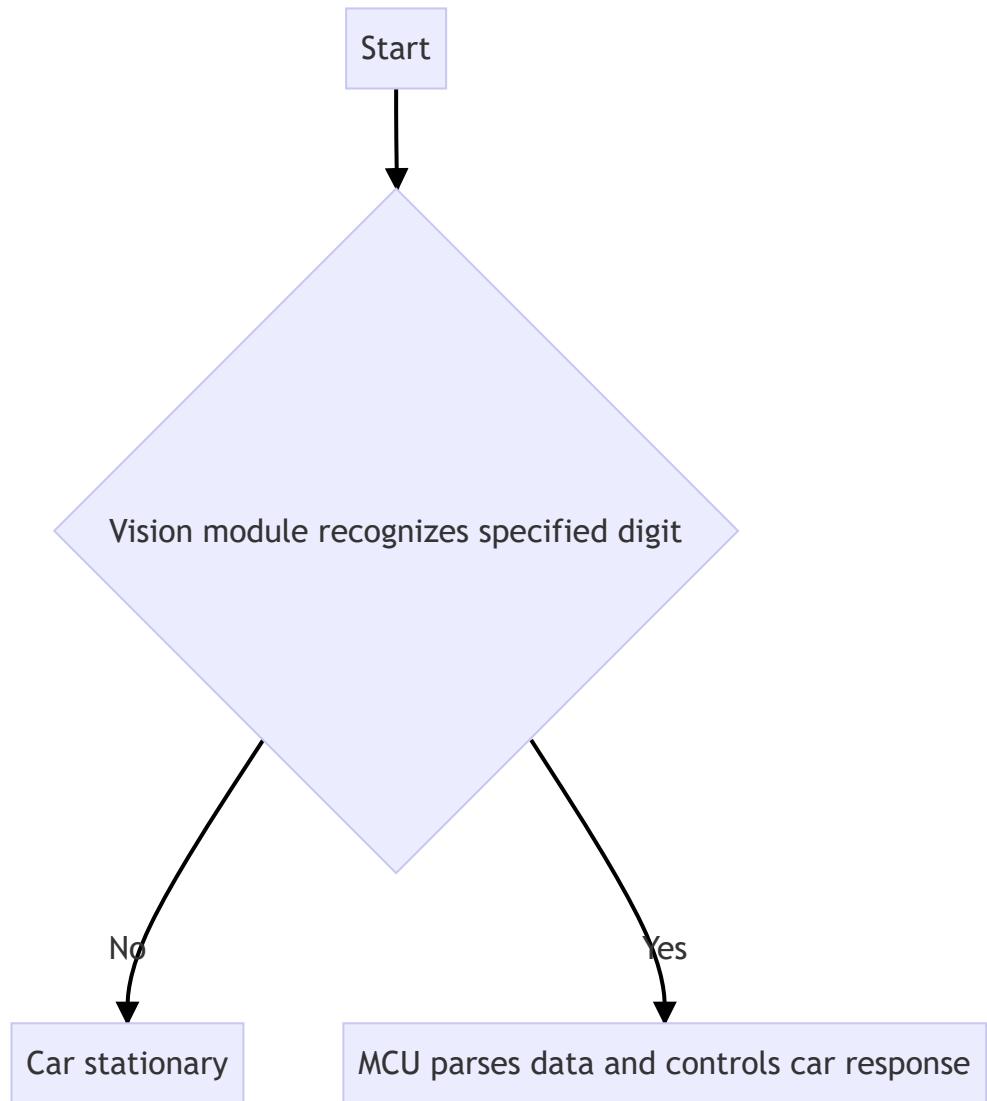
2. Code Analysis

Control Principle

K210 Protocol

Start Symbol	Length	Routine Number	Routine Group	Data Amount	Digit ID	Separator	Checksum	End Symbol
\$	XX	11	BB	01	XXX	,	XX	#

Control Flow Chart



bsp_k210_use.c

```

// 处理K210发来的单字节数据 / Process single byte data from K210
void recv_k210msg(uint8_t recv_msg)
{
    // 检测包头'$', 开始接收新消息 / Detect packet header '$', start receiving new
    // message
    if (recv_msg == '$')
    {
        new_flag = 1;
    }

    // 检测包尾'#', 结束接收并校验 / Detect packet tail '#', end reception and verify
    if(recv_msg == '#')
    {
        if( buf_len == r_index) // 数据长度匹配 / Data length matches
        {
            new_flag = 0;
            tou_flag = 0;
            len_flag = 0;
        }
    }
}
  
```

```

        // 校验和计算: 减去接收的校验和字节 / Checksum calculation: subtract
received checksum byte
    buf_crc -= buf_msg[r_index-1];
    buf_crc %= 256;

    if(buf_crc == buf_msg[r_index-1]) // 校验正确 / Checksum correct
    {
        deal_recvmsg(); // 处理接收数据 / Process received data
    }
    else // 校验失败 / Checksum failed
    {
        r_index = 0;
        buf_crc = 0;
    }
}

// 正在接收新消息时的处理逻辑 / Processing logic when receiving new message
if(new_flag == 1)
{
    // 首次接收包头 / First time receiving packet header
    if(recv_msg == '$' && tou_flag == 0)
    {
        tou_flag = 1;
    }
    else
    {
        // 存储数据并递增索引 / Store data and increment index
        buf_msg[r_index++] = recv_msg;
        buf_crc += recv_msg;
        // 未获取长度时, 从首字节读取长度 / When length not obtained, read length
from first byte
        if(len_flag == 0)
        {
            buf_len = buf_msg[0];
            len_flag = 1;
        }
    }
}
•
// 解析接收的完整消息 / Parse received complete message
void deal_recvmsg(void)
{
    //长度校验 / Length verification
    if(r_index!=buf_len)
    {
        buf_len = 0;
        return ;
    }

    // 提取有效数据 (过滤逗号) / Extract valid data (filter commas)
    for(index = 0 ;index<number;index++)
    {
        if(buf_msg[4+index] == 0x2c && i_duo ==0)//遇到逗号且未过滤 / Encounter
comma and not filtered
        {
            i_duo = 1;

```

```

        continue;
    }
    data[data_i++] = buf_msg[4+index]; // 存储有效数据 / Store valid data
    i_duo = 0;
}

// 重置接收状态 / Reset receive state
buf_crc = 0;
r_index = 0;
memset(buf_msg, 0, sizeof(buf_msg));
// 处理解析后的数据 / Process parsed data
deal_data(eg_num);

```

K210 Partial Source Code

```

def send_data(x,y,w,h,msg):
    start = 0x24
    end = 0x23
    length = 5
    class_num = 0x0B #例程编号 / Routine number
    class_group = 0xBB #例程组 / Routine group
    data_num = 0x00 #数据量 / Data amount
    fenge = 0x2c #逗号 / Comma separator
    crc = 0 #校验位 / Checksum
    data = [] #数据组 / Data array
    •
    #参数都为0 / All parameters are 0
    if x==0 and y==0 and w==0 and h ==0:
        pass
    else:
        #x(小端模式) / x (little-endian mode)
        low = x & 0xFF #低位 / Low byte
        high = x >> 8& 0xFF #高位 / High byte
        data.append(low)
        data.append(fenge) #增加 ","
        data.append(high)
        data.append(fenge) #增加 ","
    ...
    kpu = KPU()
    # 加载MNIST手写数字识别模型 / Load MNIST handwritten digit recognition model
    kpu.load_kmodel("/sd/KPU/mnist/uint8_mnist_cnn_model.kmodel")
    •
    •
    while True:
        gc.collect() # 释放内存 / Free memory
        img = sensor.snapshot() # 捕获图像 / Capture image

        # 图像预处理(适配模型输入要求) / Image preprocessing (adapt to model input
        requirements)
        img_mnist1 = img.to_grayscale(1)
        img_mnist2 = img_mnist1.resize(112, 112)
        img_mnist2.invert()
        img_mnist2.stretch_char(1)
        img_mnist2.pix_to_ai()
    •
    # 运行模型并获取输出 / Run model and get output

```

```

out = kpu.run_with_output(img_mnist2, getlist=True)
max_mnist = max(out) # 最大概率值 / Maximum probability value
index_mnist = out.index(max_mnist) # 识别结果(数字索引) / Recognition result
(digit index)
msg_ = str(index_mnist)
send_data(0,0,0,0,msg_) # 封装数据(仅发送识别结果) / Package data (only send
recognition result)

# 计算置信度(通过sigmoid函数归一化) / calculate confidence (normalize through
sigmoid function)
score = KPU.sigmoid(max_mnist)

# 当识别为数字8且置信度>0.999时, 显示并发送结果 / When recognized as digit 8 and
confidence > 0.999, display and send result
if index_mnist == 8:
    if score > 0.999:
        display_str = "num: %d" % index_mnist
        print(display_str, score)
        img.drawString(4,3,display_str,color=(0,0,0),scale=2)
        serial.send_bytarray(send_buf) # 串口发送数据 / Send data via serial
port
# 识别为数字5且置信度>0.999时, 仅显示结果 / When recognized as digit 5 and
confidence > 0.999, only display result
elif index_mnist == 5:
    if score > 0.999:
        display_str = "num: %d" % index_mnist
        print(display_str, score)
        img.drawString(4,3,display_str,color=(0,0,0),scale=2)

lcd.display(img) # 在LCD上显示图像 / Display image on LCD

```

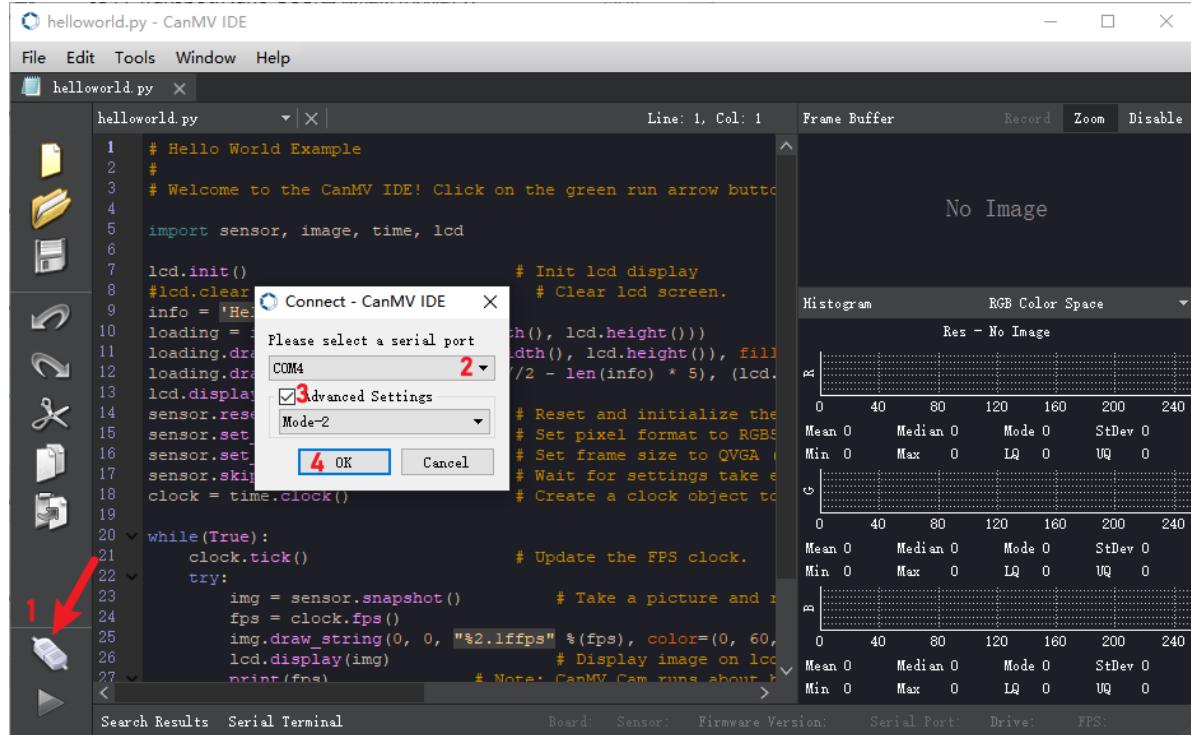
3. Main Functions

BSP_Loop

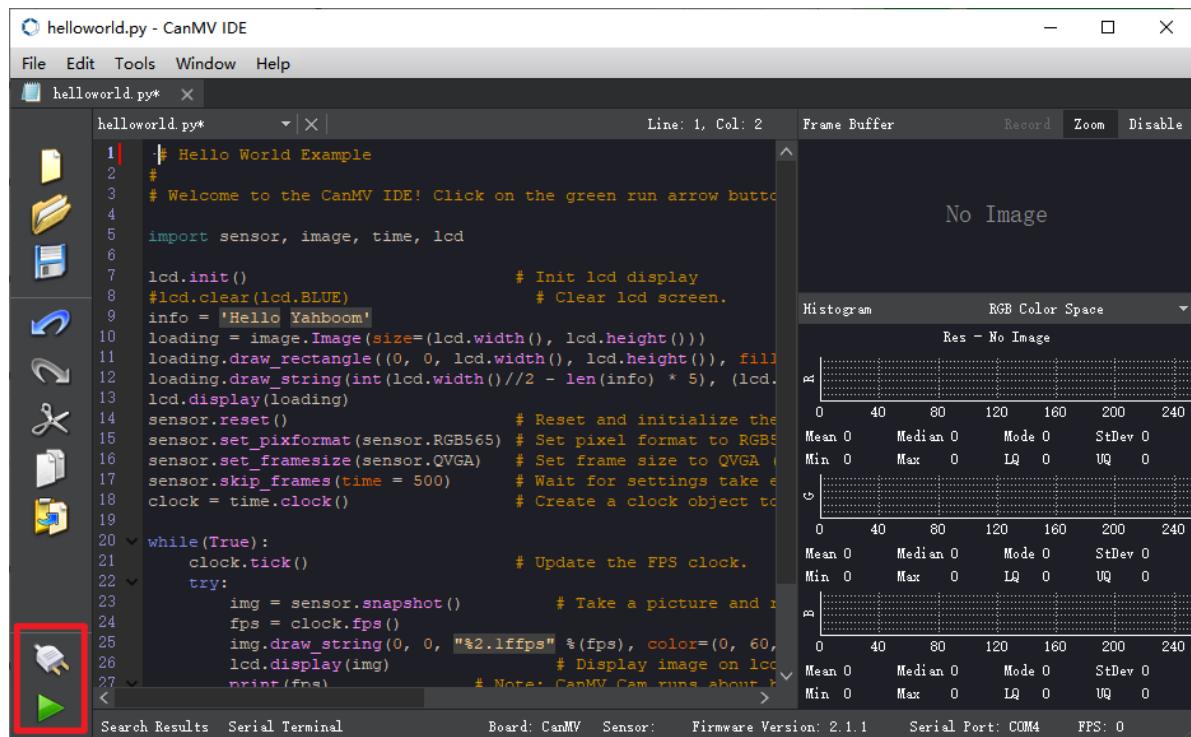
Function Prototype	void BSP_Loop(void)
Function Description	Main loop function: detects key short press events to switch g_key_flag state. When g_key_flag is 0, stops motor control and brakes; when 1, motor moves forward at speed 200, if K210 recognizes target with category 11 and ID '8', after 3 cumulative recognitions, OLED displays recognition result, controls motor to reverse for 2 seconds and resets counter, while displaying currently recognized ID on OLED
Input Parameters	None
Return Value	None

4. K210 Program Burning

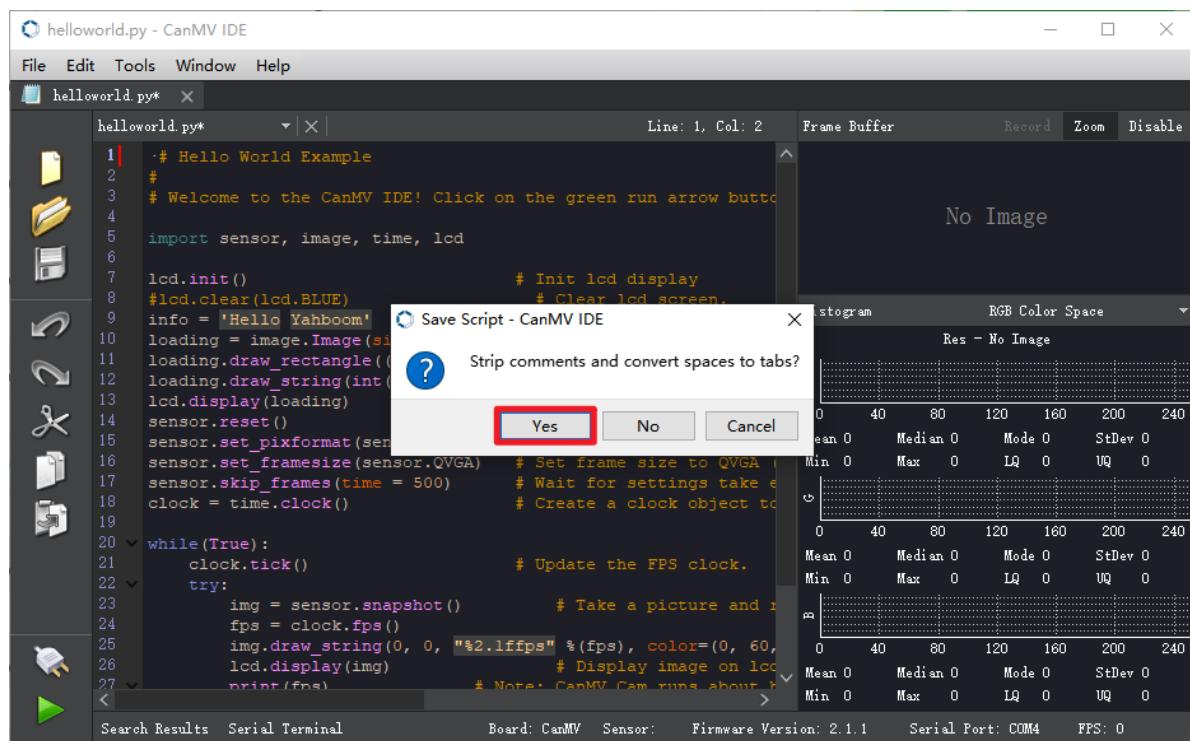
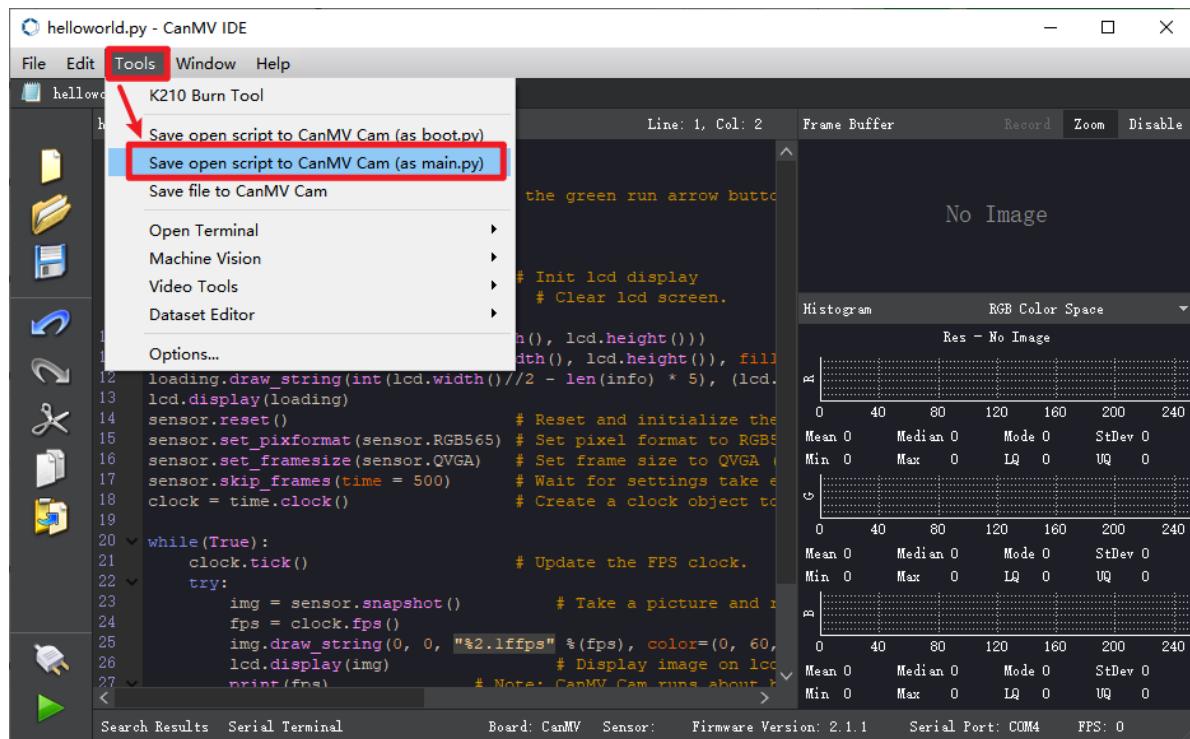
After downloading and opening CanMV IDE, we need to first drag the k210_minist.py file from the k210 source code provided in this course to CanMV IDE to open it, then connect the IDE [here](#) using helloworld.py as example



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.

The screenshot shows the CanMV IDE interface. On the left is a toolbar with icons for file operations like Open, Save, and Run. The main window has a menu bar with File, Edit, Tools, Window, Help. A tab bar at the top shows "helloworld.py*". The code editor contains the following Python script:

```

1 |  # Hello World Example
2 |
3 |  # Welcome to the CanMV IDE! Click on the green run arrow button
4 |
5 |  import sensor, image, time, lcd
6 |
7 |  lcd.init()                                # Init lcd display
8 |  lcd.clear(lcd.BLUE)                         # Clear lcd screen.
9 |  info = 'Hello Yahboom!'                     # Set LCD message
10 | loading = image.Image(size=(lcd.width, lcd.height)) # Create image
11 | loading.draw_rectangle((0, 0, lcd.width, lcd.height), fill=True) # Draw rectangle
12 | loading.draw_string(int(lcd.width/2), int(lcd.height/2), info, color=(0, 60, 160)) # Draw string
13 | lcd.display(loading)                        # Display image on lcd
14 | sensor.reset()                            # Reset sensor
15 | sensor.set_pixformat(sensor.RGB565)        # Set pixel format to RGB565
16 | sensor.set_framesize(sensor.QVGA)           # Set frame size to QVGA
17 | sensor.skip_frames(time = 500)              # Wait for settings take effect
18 | clock = time.clock()                      # Create a clock object to measure FPS
19 |
20 | while(True):
21 |     clock.tick()                          # Update the FPS clock.
22 |     try:
23 |         img = sensor.snapshot()           # Take a picture and return image
24 |         fps = clock.fps()               # Get current FPS
25 |         img.draw_string(0, 0, "%2.1ffps" % (fps), color=(0, 60, 160)) # Display FPS on image
26 |         lcd.display(img)                # Display image on lcd
27 |         print(fps)                   # Note: CanMV Cam runs about 2x slower than RPi Camera
    
```

A modal dialog box titled "Save File Success" is displayed in the center, showing "OK" as the button. To the right of the code editor, there are three histograms labeled "Histogram", "RGB Color Space", and "Res - No Image". Below the histograms, there are summary statistics for each: Mean 0, Median 0, Mode 0, StDev 0, Min 0, Max 0, LQ 0, UQ 0.

5. Experimental Phenomena

After burning the program, press KEY1 button, the car moves forward continuously. If digit 8 is recognized, the car will reverse for a period of time and then resume forward movement, while the recognized digit will be displayed on the OLED.

