智能车图像处理系统

使用说明书

目录

一、 系统简介

1.1 编写目的

本系统是针对全国大学生智能汽车大赛摄像头模块图像处理程序所设计的一款图像处理程序可视化验证系统。使用户能够对自己图像处理程序进行可视化验证与修改，满足在比赛过程中对摄像头模块图像处理程序的一系列要求。

1.2 软件功能

① 支持显示本地储存的 BMP 位图文件。

② 支持对本地 BMP 位图文件进行文件解析，得出其的分辨率和位深度。

③ 支持自动以幻灯片的形式本地的 BMP 位图文件，放映时间间隔可调。

④ 采用动态编译技术，使用户能够自己输入相关的图像处理代码,并运行。

⑤ 支持多语言输入，图像处理。部分可以使用类 c 语言，也可以使用 c 语言。

⑥ 支持对 BMP 位图文件进行颜色修改，使用户能够直观感受到自己图像处理程序运行以后发生的变化。

⑦ 支持对同一张图的原图和修改后的图像进行转换显示。

⑧ 支持二值化的黑白反色显示处理，但像素值逻辑不变。

⑨ 支持监视系统，用户可以输入一定数目的变量，使图像处理程序在动态编译运行中还可以显示出用户自己图像处理程序所想要监视的变量。

⑩ 支持用户自己定义相关设置，并且保存在本地。使用本系统时可自动从本地相关文件读取用户自定义设置。

⑪ 支持灰度图像显示和处理。

⑫ 支持串口传图，保存图像，运行代码等功能。

⑬ 支持 wifi 局域网传图，保存图像，运行代码等功能。

二、软件使用方法

2.1 运行环境

在 V4.0 版本后，系统进行了自动适配调整，绝大多数设备不会出现边界不匹配问题，若有界面相关问题，可采用下列方式尝试解决：

当出现系统界面内容与边框不匹配时如下图所示，可右键点击智能车图像处理系统.exe，点击属性，选择兼容性选项中的“更改高 DPI 设置”。勾选两个选项即可。

2.2 界面介绍

本系统解压后所产生多个文件，如图 2.2.1 所示。

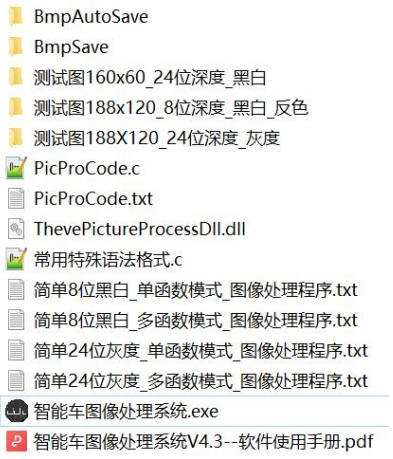


图 2.2.1 解压后所有文件展示图

文件含有软件使用手册、黑白测试图 2 套和灰度测试图 1 套、黑白和灰度相关图像处理算法程序。在系统中编写的程序会保存在 PicProCode.txt 文件中，也可以使用其他软件对此文件修改，启动程序后自动加载或点击代码同步功能实现。

双击“智能车图像处理系统.exe”， 即可以打开本系统的主页面，V4.4 版本如图 2.2.2 所示。

图 2.2.2 V4.4 版本主界面展示图

用户第一次使用时会出现如下界面：

此时需要进行软件的环境匹配验证环节，点击“自动获取 CID”，复制 CID，并联系作者（唯一指定作者，闲鱼用户“落落仪”），索要对应的系列号。一旦匹配成功，以后无需再次填写。

功能介绍：

“保存程序”：能够将用户自己所定义的“图像处理程序”保存在本地的应用程序路径下的文件内，“图像处理程序”的文本将保存在本地的应用程序路径下的 PicProCode.c 文件内。如图 2.2.3 所示。

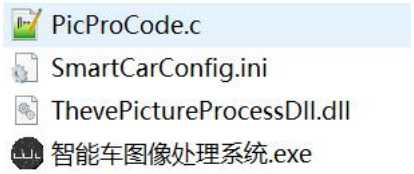


图 2.2.3 保存设置后的文件展示图

“同步代码”：能够将用户自己所定义的“图像处理程序”从在本地的应用程序路径下的文件中读取出来，用来恢复和同步图像处理代码。

“浏览文件夹”：用来打开本地的选择文件夹对话框，选择所需打开的 BMP 位图文件夹的位置，从而自动填充“图片文件夹路径”一栏。

“分辨率”和“位深度”：当用户打开一张本地 BMP 位图文件时，本系统将自动解析获取其分辨率以及位深度，并显示出来。

“参数监视器”：将用户输入的图像处理程序中，调用本系统内置监视语句，可以在程序应运行期间切换“图像处理程序面板”与“监视面板”，用来监视用户所期望看到的数据。“图像处理程序面板”如图 2.2.4 所示；“监视面板”如图 2.2.5 所示。

“帮助说明”：点击将会出现一些相关内容的帮助，如图 2.2.6 所示。

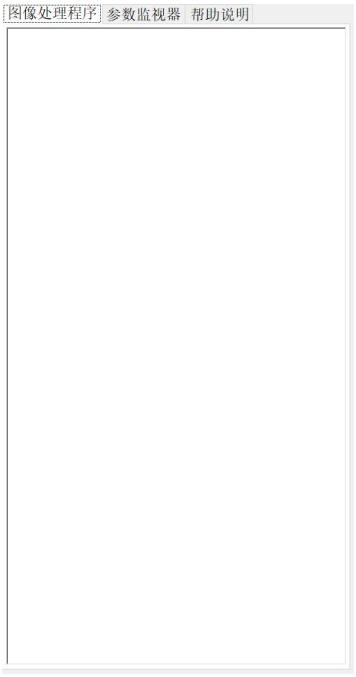


图 2.2.4 图像处理程序面板展示图

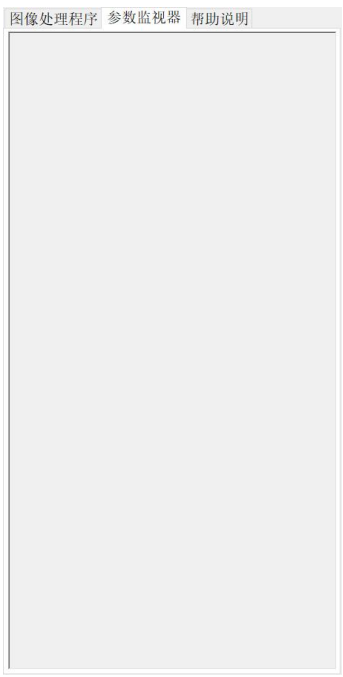


图 2.2.5 图像处理程序面板展示图

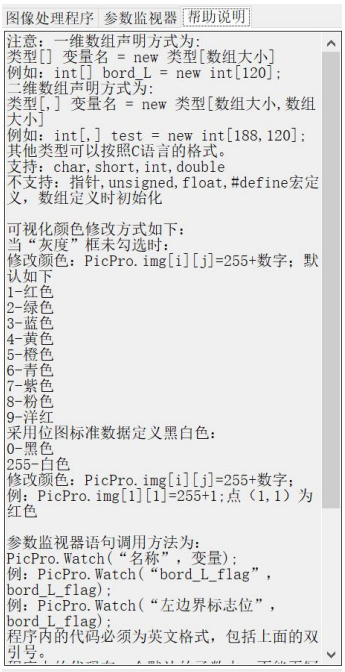


图 2.2.6 帮助界面展示图

“图片文件夹路径”：可点击“浏览文件夹”或手动输入图片文件夹路径，用来确定即将展现 BMP 位图文件所在的绝对路径的文件夹。

“时间间隔”：此功能应用于以幻灯片的形式自动播放选定文件夹目录下的 BMP 位图文件，只选项为幻灯片播放时间间隔，可调至 0.02-0.99 之间。确定以后点击“连续”按钮，即可开始以幻灯片的形式自动播放选定文件夹目录下的 BMP 位图文件。

“黑白反色”：点击后可实现黑白反色当前图片，只是显示反色，逻辑不反，即黑色为 0，白色为 255。

~~“独立程序框”：点击出现独立的程序框，独立程序框始终保持可写状态，当在独立程序框内编辑内容时，主程序框不会立即同步，这时需要手动确认同步，有两种方法，在主程序框内点击解锁=锁定时，独立程序框的内容会同步到主程序框内；或者点击主程序框内的“关闭独立程序框”也可实现同步。~~

“上一张”和“下一张”：用来改变显示相对于此刻此张图片的相邻图片。点击即完成。

“锁定”：需在“图像处理程序面板”内输入用户自定义的图像处理程序，可留空，完成以后点击锁定按钮便能运行。

~~“每次以自定义设置启动”：V4.3 版本后默认以上次的设置信息启动。~~

“隐藏相关信息栏”：勾选后，将隐藏下方的信息栏，使图像框内容最大化。

“全屏启动”：勾选后每次以全屏启动软件。

建议：取消“独立代码栏”功能，统一“保存代码”和“同步代码”功能，使用技巧：在压缩包里有 Notepad++的安装包，这是一款代码编辑器，可以用它编辑 PicProCode.c 文件，在 Notepad++中修改代码，保存后在软件界面点击“同步代码”即可。

串口图传：

“串口号”：进入此页面会自动扫描电脑的已配对串口号，请根据自己电脑的情况选择对应的串口号。

“波特率”：默认为 115200，可以自己设置所需的值。当使用蓝牙模块直接连接电脑时，一般不能超过 115200。当时用电脑的 USB 口对应的串口时，可以更具实际情况设置波特率，一般可以更高。

“图像高/宽”：设置接收图像的宽度和高度，可以参考自己使用芯片的例程中摄像头文件里定义的宽度和高度。

“打开”：即打开串口，提前选择号对应串口。不懂的纯小白可以看作者的蓝牙相关教程。串口传图的第一步，确定串口号和波特率。

“锁定”：填写图像高/宽后，选择项写入图像处理代码（写或不写），锁定时会编译一次你的代码，出错时会提示，请修改你的代码。锁定信息是第二步。

“开始传输图像”：点击后上位机才会开始接收图像，或者接收图像后点击能定格图像，用于对一些特殊图像进行反复修改代码适应性处理。

“清除当前图像”：顾名思义。

“保存当前图像”：会将当前的图像保存在 exe 软件所在文件目录下的 BmpSave 文件内，命名为“img”+数字，其中，数字会根据文件内存在文件序号后续增加，不会覆盖原图。

“自动保存图像”勾选框：勾选后，会在接收图像后，将图像保存在 exe 软件所在文件目录下的 BmpAutoSave 文件内，命名为“img”+数字，其中，数字会根据文件内存在文件序号后续增加，不会覆盖原图。请自动保存所需的图像后将内图像移至另外的文件夹，以防数据丢失。

“代码同步”：本软件自带的代码编辑器功能不强，建议使用其他轻量式的代码编辑器（例如 Notepad++）修改文件夹内的 PicProCode.c 文件，修改保存后，在此软件点击“恢复设置”或“代码同步”即可同步至此软件的代码编辑器中，代码改动后，需“解锁”，“锁定”操作来编译代码。

“显示转换”：完成原图、处理图、标记图三种图像的转换。

“运行”：点击后实现在传输图像上运行用户的代码程序，并能可视化的显示出来。

“自动运行代码”勾选框：勾选后实现自动在每次图像传输成功后的运行一次用户的图像处理代码程序，解放你的双手。

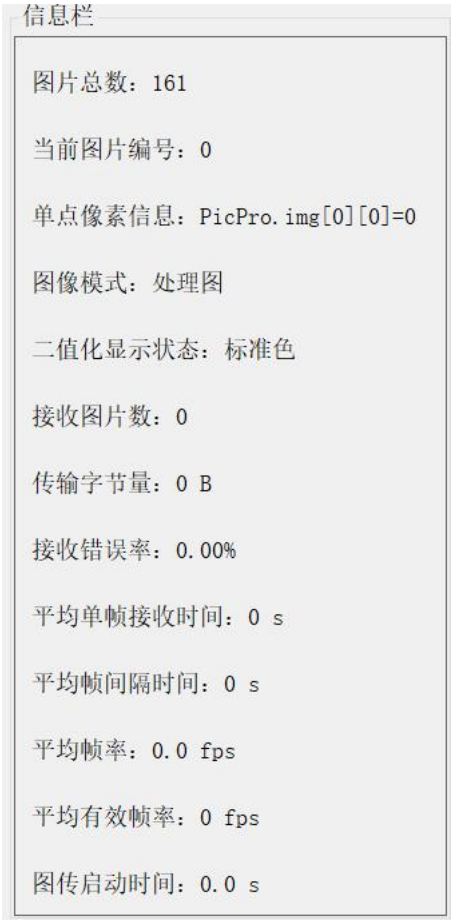
WiFi 图传：本 wifi 图传模式优先配合配套上位机的 SPI-WiFi 图传方案。

“IP 地址”：连接方案中的 wifi 热点后在本机产生的 IP 地址。默认为“192.168.4.2”。

“端口号”：默认为本方案的端口号“51234”。

其他信息与“串口图传”相似。

参数面板：



“图片总数”：所选文件夹内的.bmp 文件的数量。

“当前图片编号”：为兼容性显示，此参数表示软件打开后的加载的第几个图片。

“单点像素系信息”：表示第几行第几列的像素值为多少，该数据和鼠标坐标绑定。

“图像模式”：在采用本地图集功能下显示图像共有三种模式“处理图”、“原图”和“标记图”。

“二值化显示状态”：是否反色，对二值化图像进行反色显示。

“接收图片数”：在串口图传或 WiFi 图传中显示接收到的图片数。

“传输字节量”：在串口图传或 WiFi 图传中显示接收到的字节大小。实时显示上位机接收下位机传输的字节量成功获取有用信息后会移除累计的字节量，是量衡数据是否持续接收的重要参数。当“相关信息栏”出现“此帧数据混乱，已丢弃”字样时，说明数

据出错，考虑是否发送信息有误或波特率过高或重启软件。

“接收错误率”：在串口图传或 WiFi 图传中显示接收到的数据中数据出错或丢失导致的图片数据损坏的数量占总量的比率。

“平均单帧接收时间”：在串口图传或 WiFi 图传中显示当接收第一个字节时到接收完一张图片的数据所用时间。

“平均帧间隔时间”：在串口图传或 WiFi 图传中显示当接收第一个字节时到接收下一张图片第一个字节时的所用的时间。

“平均帧率”：在串口图传或 WiFi 图传中显示随图传启动时间运行时每秒所接收到的图片数。

“平均有效帧率”：在串口图传或 WiFi 图传中显示随平均帧间隔时间之和的运行时每秒所接收到的图片数。

“图传启动时间”：在串口图传或 WiFi 图传中从接收到第一个字节开始计时。

2.3 运行演示

ⅰ，“本地”相关操作

1，点击“浏览文件夹”，在弹出的打开文件夹对话框内选择含有 BMP 位图文件所在的文件夹，点击“确定”，如图 2.3.1 所示。

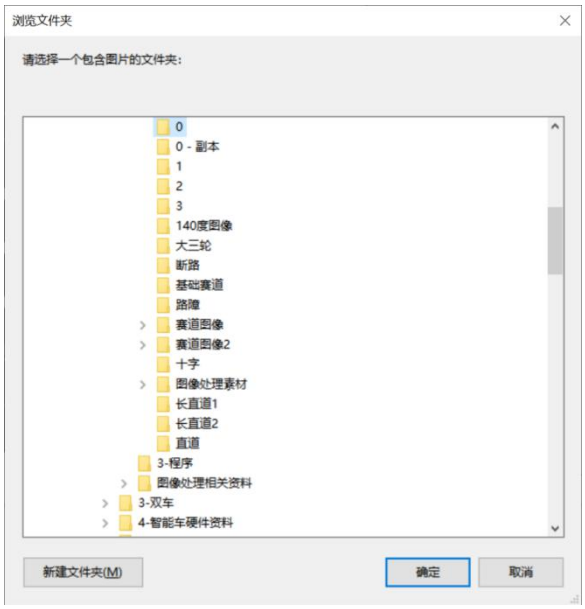


图 2.3.1 打开文件夹对话框展示图

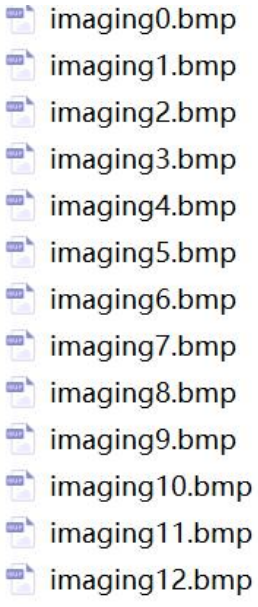


图 2.3.2

2，V4.0 版本后可自动识别文件夹内的 BMP 图片，并按顺序载入显示队列。即需要保证所选文件夹内有 BMP 图片。点击“保存设置”将锁定的目录保存。

3，在“图像处理程序”面板内写入用户自己的图像处理代码，完成以后点击“锁定”按钮。如图 2.3.3 所示。此刻原本只读状态的“上一张”,“下一张”,“连续”等变成读写状态；而“图片文件夹路径”,“图片处理程序”面板被锁定，将由读写状态成只读状态。

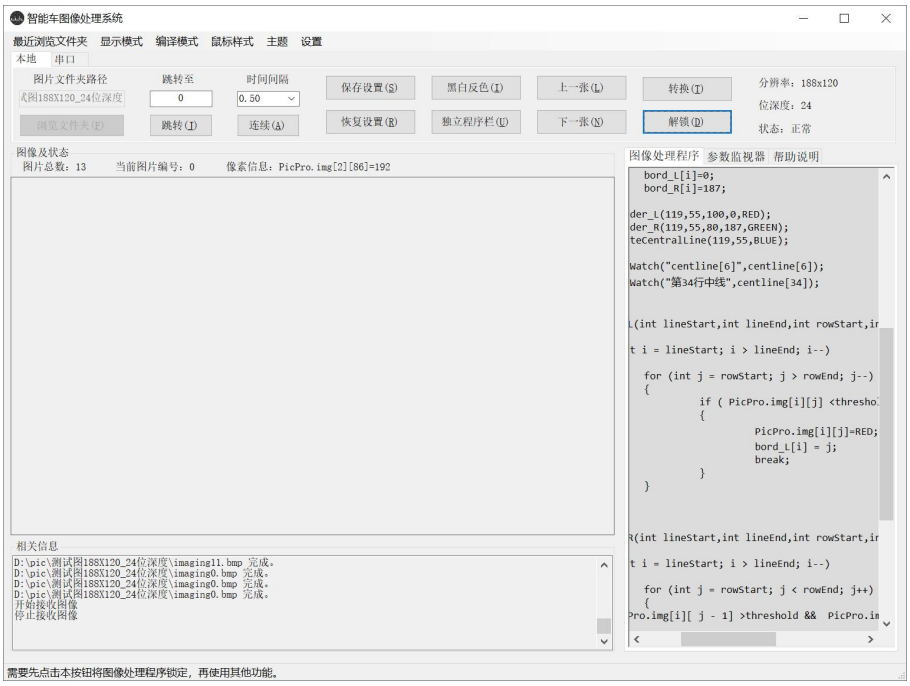


图 2.3.3 准备就绪展示图

4，点击“下一张”或“连续”按钮，“图片显示界面”将改成此图片相邻的图片经过用户图像处理程序修改之后可视化的图片。

5，图像处理程序内容为空时，显示 BMP 的原图。根据所需，判断是否点击“黑白反色”按钮，只是显示反色，逻辑不变。

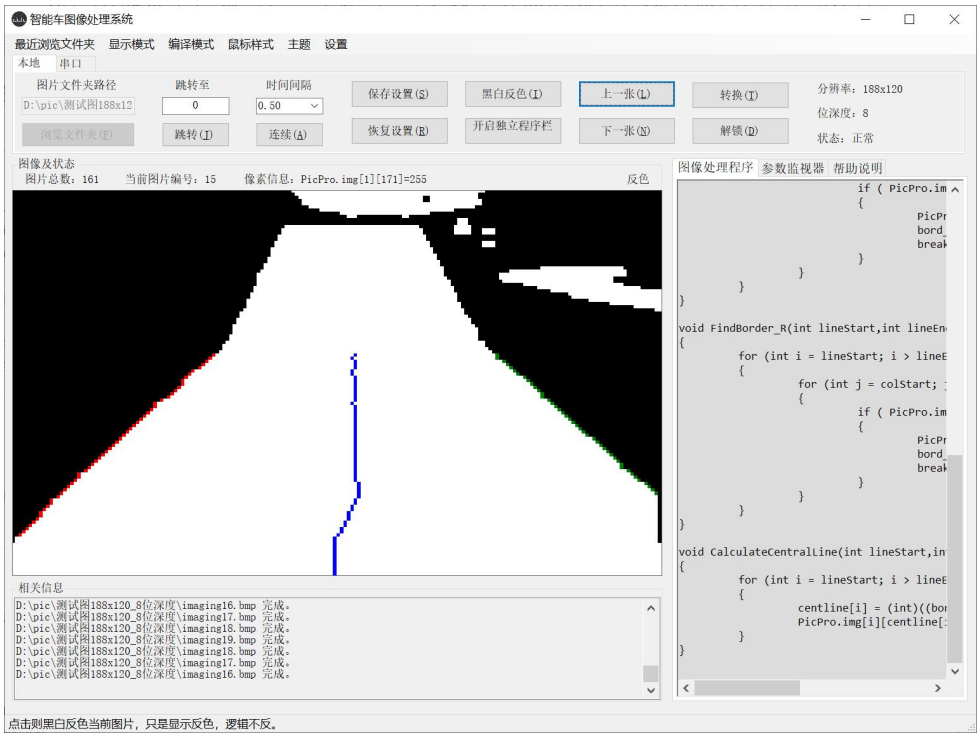


图 2.3.4 经用户图像处理程序修改后的图

6，当用户添加“PicPro.Watch（）” 语句时，可点击“监视”按钮，切换“图像处理程序”界面和“数据监视器”界面，如图 2.3.5 所示。

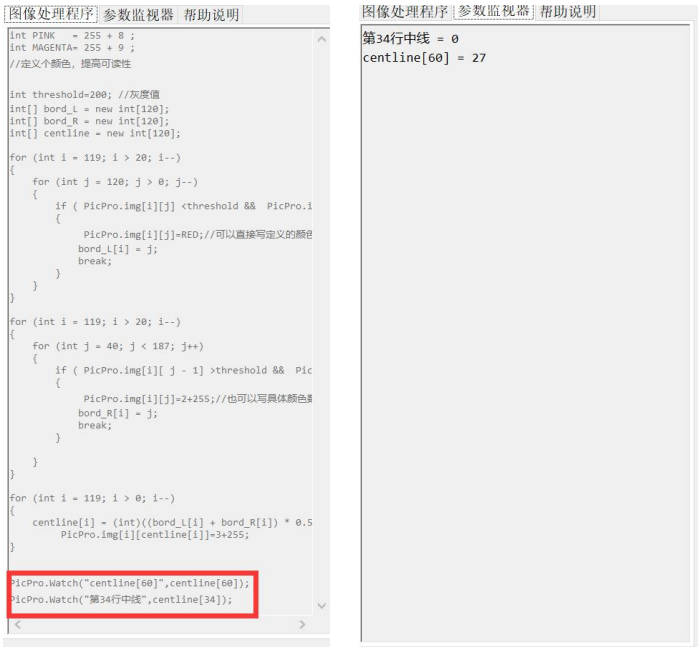


图 2.3.5 数据监视器图

7，对于图像的改色在帮助里看，需对颜色加上 255。

8，V4.0 后统一读取方式，8 位，24 位，均能自动识别，程序无需改变。统一化程序颜色的设置方式，均为 255+对应数据可实现颜色的改变。

9，新增独立窗口功能，如图 2.3.6，能在大小可变的窗口中查看和修改代码。

10，独立程序框始终保持可写状态。当在独立程序框内编辑内容时，主程序框不会立即同步，这时需要手动确认同步，有两种方法，在主程序框内点击解锁-锁定时，独立程序框的内容会同步到主程序框内；或者点击主程序框内的“关闭独立程序框”也可实现同步。

11，V4.2 新增“最近浏览文件夹”菜单栏，点开后能浏览曾经保存过的有效文件夹路径，限量 25 个，多余删除最早保存的路径。

12，增加菜单栏第二项“显示模式”：具有“拉伸模式”和“等宽长比模式”。“拉伸模式”使图片框最大化显示。“等宽长比模式”保证了保持宽长比的情况下最大化图片。

13，增加菜单栏第三项“编译模式”：具有“单函数模式”和“多函数模式”。“单函数模式”代码中不能出现函数，代码默认在一个函数中，只需要写逻辑代码即可，不需用写函数头。“单函数模式”代码中可以出现函数，但需要保留“public void PicProMain()”函数不能修改。代码最后一行建议追加一个新行。

14，增加菜单栏第四项“鼠标样式”：具有“箭头”、“十字”、“圆圈”。其中，所有样式所指的像素点为样式图像的左上角部分得内容，而非绝对的中心像素。

15，增加菜单栏第五栏“主题”：可更改主界面的颜色，现有黑白两色，从白色更改为黑色时能立刻修改，从黑色转为白色主题时，软件会自动重启。黑色主题如图 2.3.7 所示。

16，增加菜单栏第六栏“设置”：具备“隐藏相关信息栏”和“全屏启动”。V4.3 后将默认软件多项设置以上次使用记录为准恢复。

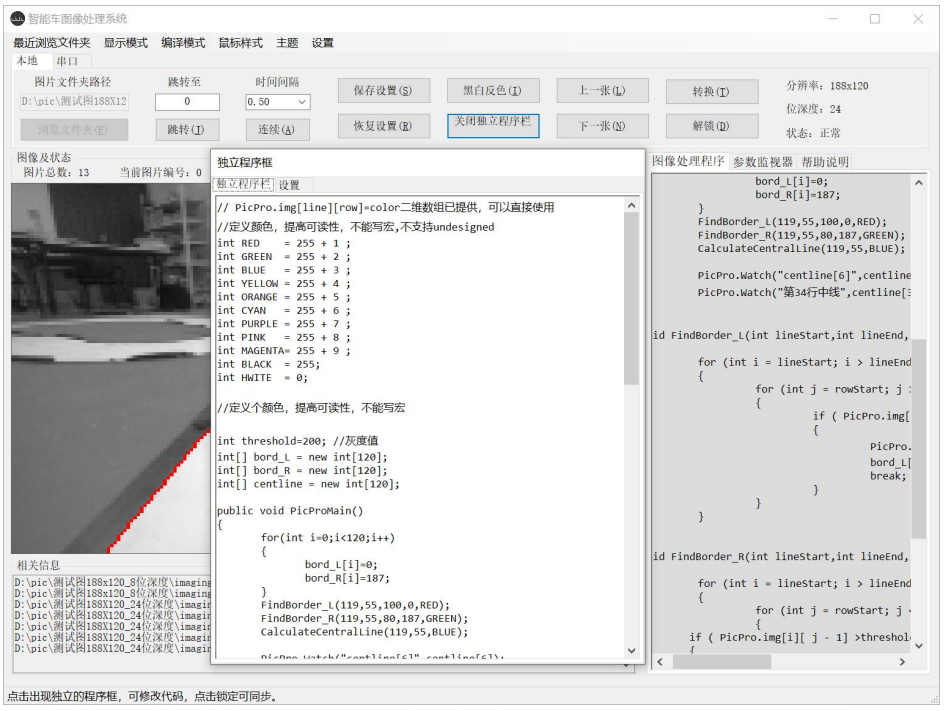


图 2.3.6 灰度图和独立程序框展示图

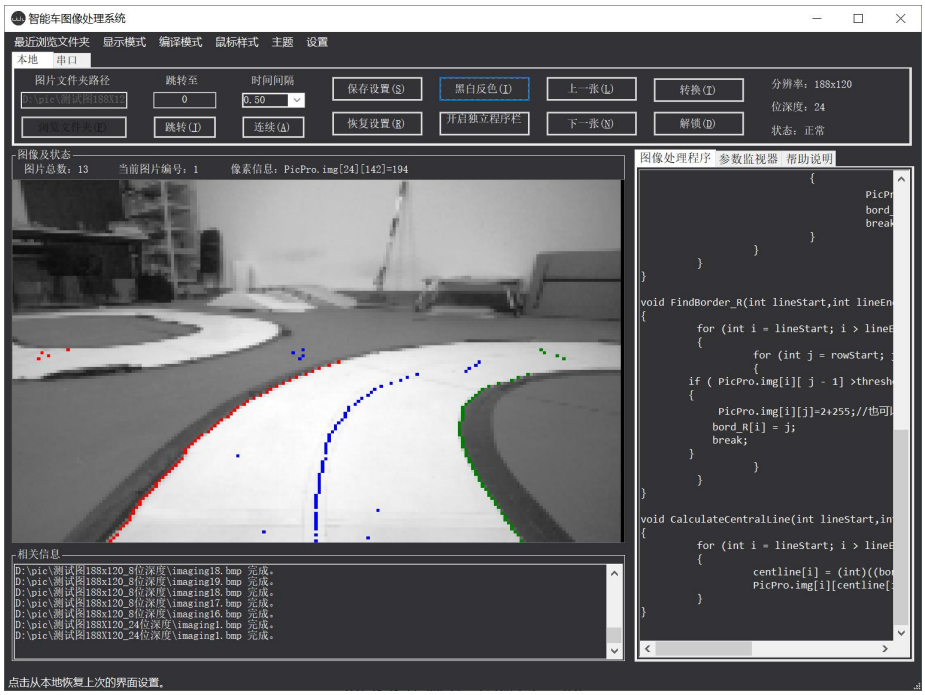


图 2.3.7 黑色主题展示图

17，添加快捷键，对按钮有（x）内容的按钮，采用 Alt+x（对应按键）即可实现快捷键控制。

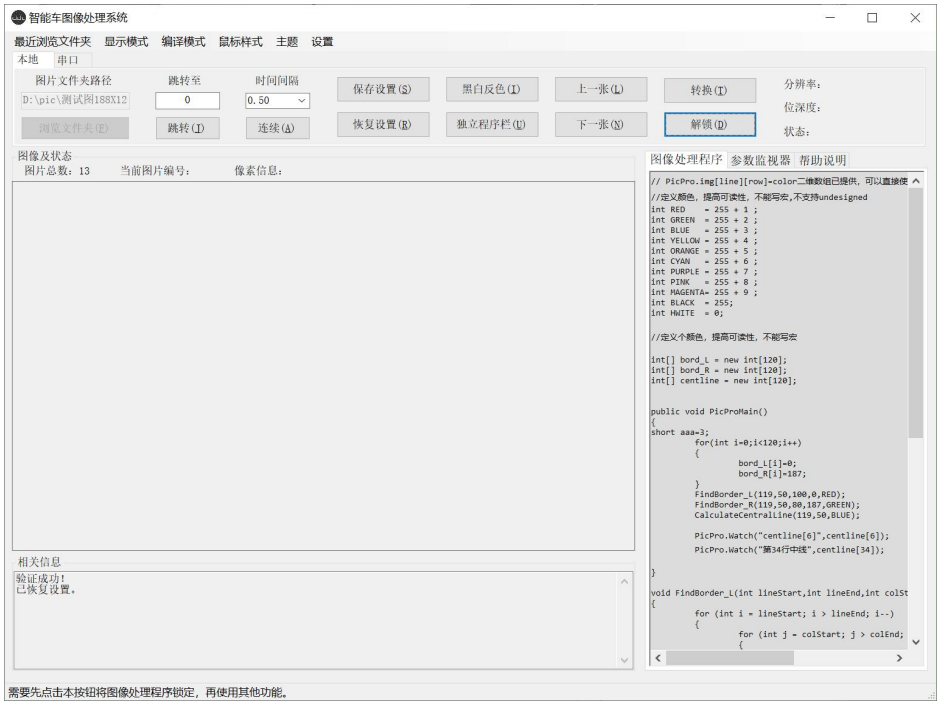


图 2.3.8 按钮快捷键展示图

ii，“串口图传”相关操作

1，点击“串口”任务卡，进入串口图传相关界面。可以使用笔记本电脑自带的蓝牙，所选

波特率最高为 115200。可以使用收发分体蓝牙，最高波特率可以更高。可以使用无线串口模块等。

2，选择对应的串口号和波特率，默认 115200，可选 9600，可自定义具体数据。

3，填写接收图像的宽度和高度，单位像素。

4，打开串口，根据串口不同有不同的情况发生。立刻打开的串口一般为本机虚拟串口，硬件串口或蓝牙配对后两个串口中“没用的”一个。较慢打开的串口，大约需要 3-5s 一般为“有用的”蓝牙串口。超过 10s 还没打开的串口，要么被电脑的其他软件所占用，要么打开的并非你所需的串口，软件可能未响应，这时需调用任务管理器 keill 掉。

5，打开串口后，“锁定”按钮出现可点状态。此时可以修改或同步用户的代码，确定后点击“锁定”按钮，会编译代码，错误时会提示。

6，点击“开始接收图像”，此时会接收到下位机传输的图像了。当“传输字节量”开始变化时，说明传输正常，每当收到 2+图像宽度*图像高度+2 字节时会判定图像数据是否正常，不正常则显示“此帧数据混乱，已丢弃”，正常则记录清零，“当前图片编号”加 1，图像框显示图像。接收过程中再次点击可以暂停接收，修改代码，“运行”图像处理，保存图像等操作。

7，传图速度与波特率和图像分辨率相关，实测 115200 波特率 188*120 分辨率 2s 一张图，115200 波特率 160*60 分辨率 1.6s 一张图，其他情况自测。使用电脑蓝牙时波特率不能超过 115200，想使用更高波特率需用硬件接收，例如主从蓝牙、无线串口等。高波特率当出现下位机持续发送数据且“传输字节量”不再变化时，请重启软件或联系作者。

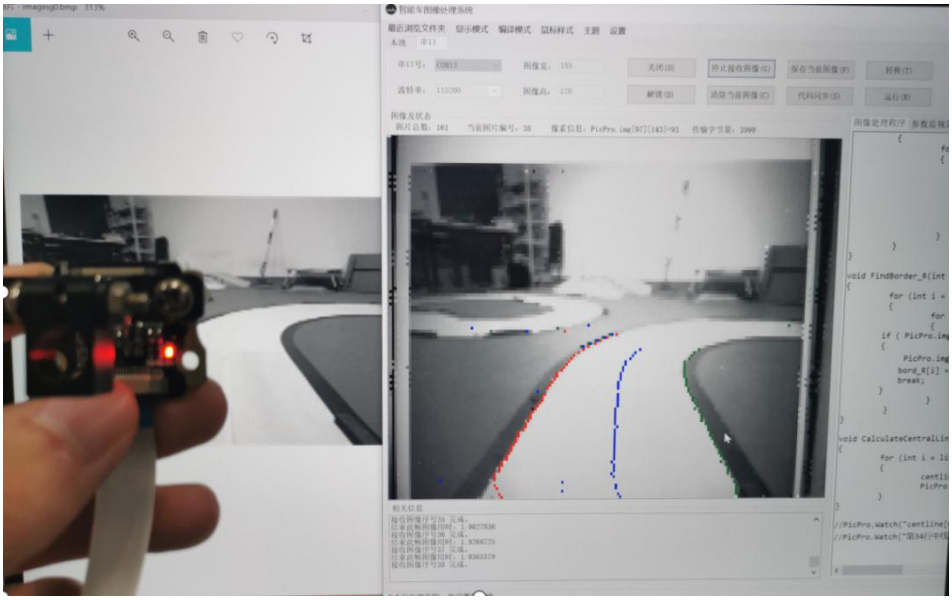


图 2.3.9 摄像头采集传输处理展示图

iii，“WiFi 图传”相关操作

图 2.3.10 WiFi 图传显示界面

本演示采用配套的 SPI-WiFi 方案。

硬件选择 Nodemcu，已搭好外围电路，方便使用，也可采用 esp8266-12F，搭建外围电路也能使用本方案。

1，给 Nodemcu 烧录固件，在文件路径：智能车图像处理系统 V4.4\4-其他工具\2-Nodemcu 固件烧录器中，双击“Nodemcu 固件烧录器.exe”，如图 2.3.11 所示。

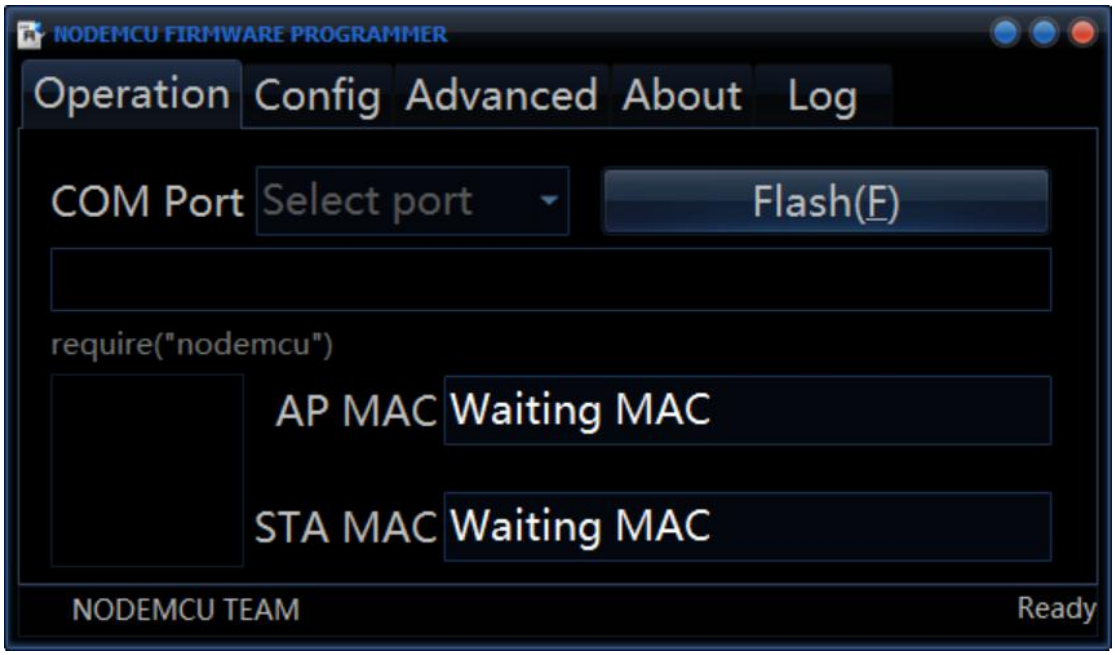


图 2.3.11 Nodemcu 固件烧录器界面

在“Advanced”中配置选项，如图 2.3.12 所示。

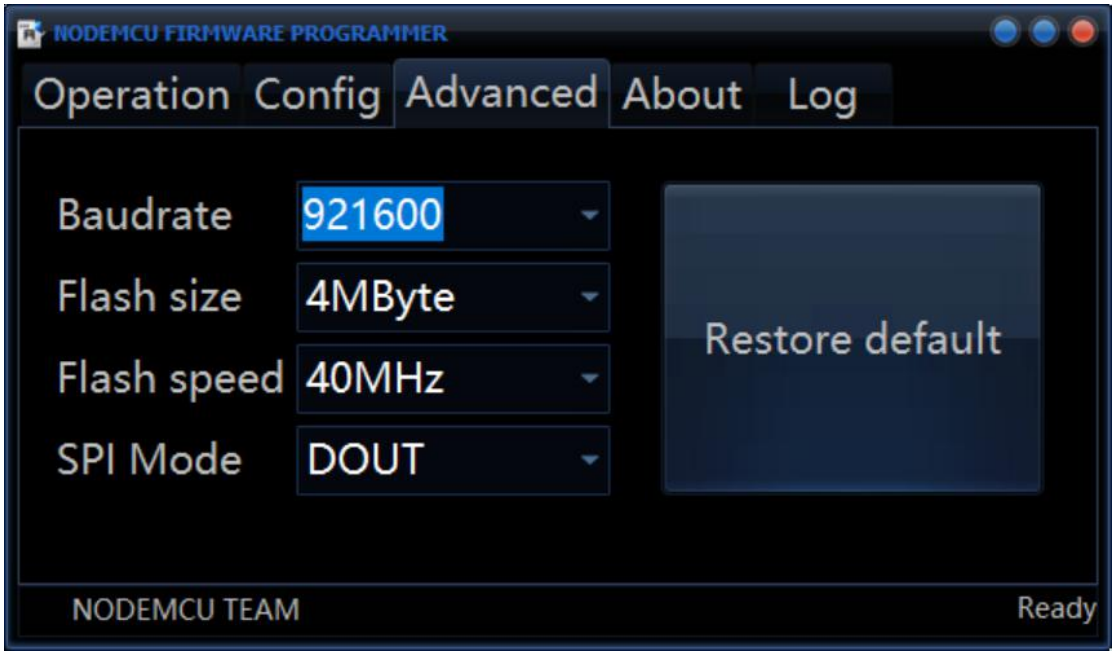


图 2.3.12 Nodemcu 固件烧录器界面

在“Config”中选择同目录下的“Nodemcu_spi_wifi_V1.bin”文件，地址写入 0x00000，如图 2.3.13 所示。

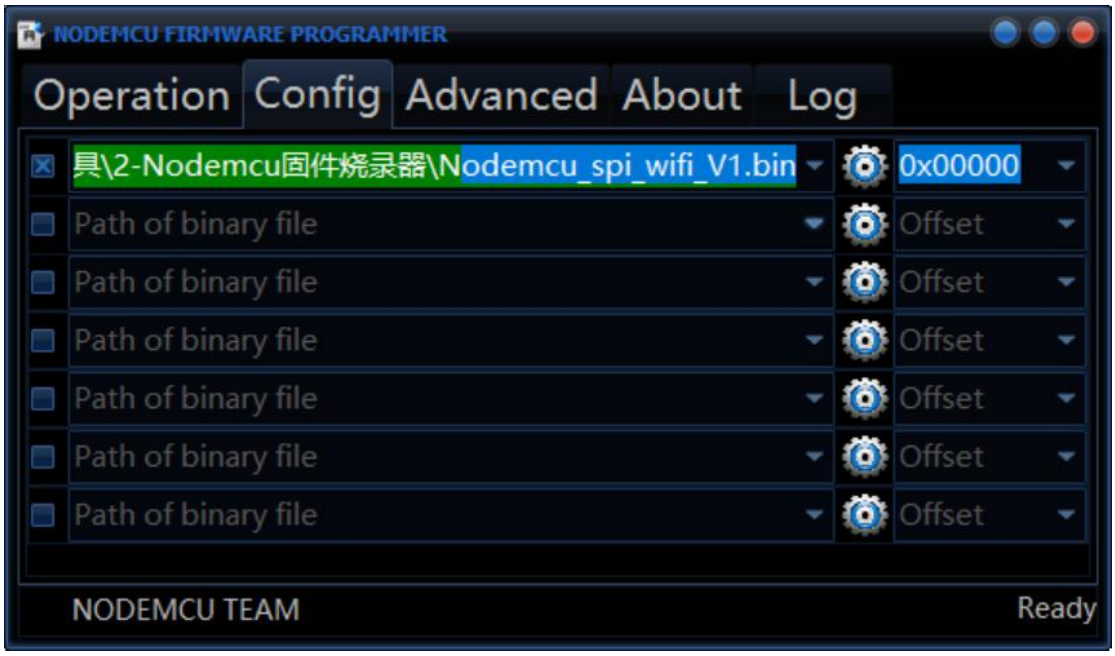


图 2.3.13 Nodemcu 固件烧录器界面

在“Operation”选择中选择正确的对应端口号，点击“Flash”即可，自动下载，无需认为操作。等到进度条完成可以关闭软件。

2，按动 Nodemcu 上的“Rst”按钮以复位硬件。

3，给 Nodemcu 配置固件，在文件路径：智能车图像处理系统 V4.4\4-其他工具\3-Nodemcu

固件配置器中，双击“Nodemcu 固件配置器.exe”，如图 2.3.14 所示。

图 2.3.14 Nodemcu 固件配置器界面

配置器界面介绍：

“自动识别”：能自动识别电脑端口中的“CH340”或“CP2X”的转换芯片，即对应 COM 口为 Nodemcu 或其他硬件的占用端口。

“打开”：打开端口。

“开始配置”：区别上位机配置和手动配置。

“自动获取”：推荐点击此按钮，一键识别+打开端口+获取信息。

“获取信息”：点击后会获取当前的 Nodemcu 相关配置信息。

“重置固件”：返回初始化的信息，点击后需要复位 Nodemcu 硬件。

“WiFi 名称”:考虑到一个实验室内可能会出现不止一个使用本方案的用户，为了区分各自的网络，可以自定义 WIFI 名称，复位后生效。

“用户 IP 地址”：同上，修改此 IP 地址要保证前三个小数点的值不变。

例如：可修改为 192.168.4.4 - 192.168.4.254；

“用户端口号”：一般情况无需修改，如有占用打不开的情况再修改。

“图像高度”：传输的图像高度，默认 60。

“图像宽度”：传输的图像宽度，默认 160。

设置完成后点击“关闭”后关闭软件。

4，连接 MCU。

管脚功能	D5	D6	D7	D8	D4	D3	5V	GND
管脚名	SCK	MISO	MOSI	SS	RTS	CLR	5V	GND
GPIO	GPI	GPO	GPI	GPI	GPO	GPI	-	-
默认值	-	-	-	HIGH	HIGH	HIGH	-	-

Nodemcu 采用 SPI 从机模式，D5-D8 为硬件 SPI 相关引脚。

D4 为 RTS 流控，当接收完一帧图像数据后采用 WiFi 发送时为低电平，否则为高电平。

D3 为 CLR 接收数据清零控制端，当读取到低电平时，清除接收的缓存数据，建议当存在 MUC 断电/复位而 Nodemcu 未断电/复位的情况时，每次初始化 MCU 时先向 Nodemcu 的 D3 引脚拉低一次，延时 1ms 再拉高。

5，MCU 程序。

例程见文件路径：智能车图像处理系统 V4.4\3-程序例程\MCUSpiWifi 使用程序.txt

考虑到文件移植性，未写成.c.h 库文件，采用面向过程设计，使用宏定义，用户只需更改部分接口函数即可使用。

这里做简要介绍，所述内容用户需要修改，未提及内容用户无需关心和修改。

①宏

```
#define IMG_H MT9V03X_CSI_H //此处可直接修改为所需的分辨率 60 也可以对接摄像头文件的宏
#define IMG_W MT9V03X_CSI_W //此处可直接修改为所需的分辨率 160 也可以对接摄像头文件的宏
#define IMG_SIZE (IMG_H*IMG_W)
#define IMG_EXCEPT ((IMG_SIZE+4)%32)
#define IMG_EXTRA (IMG_EXCEPT?1:0)
#define IMG_PACKET (IMG_SIZE/32+IMG_EXTRA)
// 下行需替换为用户使用MCU的SPI传输函数，将发送数组、接收数组、发送字节数改为sendBuff,recvBuff,bytes即可
#define IMG_SPI_SEND(sendBuff,recvBuff,bytes)  spi_mosi(SPI_1, SPI1_CS0_D13, sendBuff, recvBuff, bytes, 1)
#define BUSY GPIO_PORT //换成用户MCU上的某一GPIO，用来对接NodeMCU的RTS
#define KEY GPIO_PORT   //按键GPIO，用来测试传图使用，可取消
#define GET_FREE  gpio_get(BUSY)//获取Nodemcu的RTS状态
#define GET_KEY  gpio_get(KEY)//获取按键GPIO值，用来测试传图使用，可取消
uint8_t spiSendImgArray[IMG_SIZE + 4];
void SpiWiFiImgInit();
void SpiWifiImgSend(uint8_t (*imgPtr)[IMG_W]);
```

使用时可按图示定义 IMG_H 和 IMG_W，后面内容替换成用户 MCU 的宏也可以直接定义为（例）#define IMG_H 60　　#define IMG_W 160

建议：由于摄像头采样分辨率、MCU 处理效率和传图方案的综合考虑，强

烈建议采集 60*160（及以下）的图像传输和使用。如果希望使用此分辨率，由高分辨率（120*188）转换为此分辨率，只需按上文所述修改为数字即可。

#define BUSY GPIO_PORT

#define GET_FREE　gpio_get(BUSY)

两行为定义一个 GPIO 为 Nodemcu 的 RTS 流控的获取端，用户将 GPIO_PORT 替换为 MCU 的某一 GPIO 即可，初始化内容在后文，将 gpio_get(BUSY)替换为 GPIO 的读取函数，此处实在不懂可联系作者。

②初始化

初始化函数，内部需要大量替换为用户的函数。

spi_init()为 SPI 初始化函数，替换为用户函数库内的相关函数，设置参数为 SPI 主机，四线，CPOL=0，CPHA=0 (一般为模式 0)，波特率 4M-10MHz

gpio_init(BUSY, GPI, 1, GPIO_PIN_CONFIG);

GPIO 初始化，对 RTS 流控引脚的初始化，设置为输出，默认高电平，其他参数没有可不写，有其他参数按用户例程设置。

```
//spi传图初始化函数
void SpiWiFiImgInit()
{
  //SPI初始化，选择硬件SPI，波特率默认10M，根据单片机性能调节
  spi_init(SPI_1, SPI1_SCK_D12, SPI1_MOSI_D14, SPI1_MISO_D15, SPI1_CS0_D13, 0, 10 * 1000 * 1000);
  //额外GPIO初始化 设置为输入，默认为高电平（上拉）
  gpio_init(BUSY, GPI, 1, GPIO_PIN_CONFIG);
  gpio_init(KEY, GPI, 1, GPIO_PIN_CONFIG);
  //传输协议初始化
  spiSendImgArray[0] = 0xfc;
  spiSendImgArray[1] = 0xcf;
  spiSendImgArray[IMG_SIZE + 2] = 0xcf;
  spiSendImgArray[IMG_SIZE + 3] = 0xfc;
}
```

③传输函数

```
//spi传图给esp的函数
void SpiWifiImgSend(uint8_t (*imgPtr)[IMG_W])
{
  static uint8_t spiSendBuf32[32] = {0x02, 0x00};
  static uint8_t spiRecvBuf32[32] = {0};
  if ( !GET_FREE)
  {
    return;
  }
  for (int i = 0; i < IMG_SIZE; i++)
  {
    spiSendImgArray[2 + i] = *(*imgPtr + i);
  }
  for (int i = 0; i < IMG_PACKET; i++)
  {
    for (uint8_t j = 2; j < 34; j++)
    {
      spiSendBuf32[j] = spiSendImgArray[i * 32 + j - 2];
    }
    if (!GET_FREE)
    {
      return;
    }
    if (IMG_EXTRA && i == IMG_PACKET - 1)
    {
      IMG_SPI_SEND(spiSendBuf32, spiRecvBuf32, 2 + IMG_EXCEPT);
    }
    else
    {
      IMG_SPI_SEND(spiSendBuf32, spiRecvBuf32, 34);
    }
    for (int i = 0; i < 100; i++)
      asm("nop");//如果这行报错，换成分号即可
  }
}
```

次函数无需关系，若在最后一行报错，需修改为分号即可。

④调用案例

测试实例：采用逐飞库的 RT1064 芯片测试。

无需看懂整个过程，基本过程为：

关中断（可选）-板级初始化-摄像头初始化-图传方案初始化-开中断（可选）-主循环-判断按键是否按下（可选）-判断是否采集到一帧图像（可选）-调用传输函数传输图像-清除采集图像标志位（可选）-循环。

```c
int main(void)
{
  DisableGlobalIRQ();
  board_init();   //初始化MPU 时钟 调试串口
  systick_delay_ms(300);  //延时300ms，等待主板其他外设上电成功
  mt9v03x_csi_init();//初始化摄像头
  SpiWiFiImgInit();
  EnableGlobalIRQ(0);

  while (1)
  {
    if (GET_KEY == 0)
    {
      if (mt9v03x_csi_finish_flag )
      {
        SpiWifiImgSend(mt9v03x_csi_image);
        mt9v03x_csi_finish_flag = 0;
      }
    }
  }
}
```

目前 SPI-WIFI 图传方案适合 60*160=9600 字节以下的图像传输，更大字节尚未统合，需更换固件包。

注：有任何建议或者意见咨询闲鱼用户“落落仪”。

附录 1：

使用此上位机时需对代码进行一点点变动，还请谅解，具体规则为：

1，定义一维数组

类型[] 变量名 = new 类型[数组大小]

例如：int[] bord_L = new int[120];

调用：bord_L[3]=23;

定义时不初始化：

```
int[] array1 = new int[6];
```

定义时初始化：

```
int[] array1 = new int[6] { 1, 2, 3, 4, 5, 6 };
int[] array2 = new int[] { 1, 2, 3, 4, 5, 6 };
int[] array3 = { 1, 2, 3, 4, 5, 6 };
```

2，定义二维数组

类型[,] 变量名 = new 类型[数组大小,数组大小]

例如：int[,] array2D = new int[188,120];

调用：array2D[3][3]=23;或 test[3,3]=23;

定义时不初始化：

```
int[,] array2D = new int[188,120];
```

定义时初始化：

```
int[,] array2D1 = new int[3, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 } };
int[,] array2D2 = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 } };
int[,] array2D3 = { { 1, 2 }, { 3, 4 }, { 5, 6 } };
```

3，修改图像像素颜色

采用位图标准数据定义黑白色：

PicPro.img[i][j]=0 黑色

PicPro.img[i][j]=255 白色

PicPro.img[i][j]=255+数字;

(1-红色 2-绿色 3-蓝色 4-黄色 5-橙色 6-青色 7-紫色 8-粉色 9-洋红)

其中，PicPro.img[i][j]数组无需定义，在加载图片或收到图片后自动产生。

4，参数监视器使用

PicPro.Watch("名称", 变量);

例：PicPro.Watch("bord_L_flag", bord_L_flag);

例：PicPro.Watch("左边界标志位", bord_L_flag);

例：PicPro.Watch("bord_L[4]", bord_L[4]);

例：PicPro.Watch("左边界第 4 行值", bord_L[4]);

第一个参数是将打印出来用来提示用户的信息，第二个参数是具体参数。

5，内置数学函数的使用

取绝对值：

类型 = Math.Abs(类型/变量);

支持大多数变量，byte,short,int,float,double 等

函数返回值和变量类型一致。

例如：double a=-3; double b =Math.Abs(a);

取最小值/最大值：

类型 = Math.Min(类型 a1,类型 a2);

支持大多数变量，byte,short,int,float,double 等

函数返回值和变量类型一致。

例如：double a=3; double b =4;double c = Math.Min(a,b);

同理:Math.Max

取根号：

double a = Math.Sqrt (double b);

只能用 double

次方：

double a = Math.Pow(double b,double c);

a=b 的 c 次方

三角函数：

只能用 double

double a = Math.Sin(double b);

double b = Math.Asin(double a);

double a = Math.Cos(double b);

double b = Math.Acos(double a);

double a = Math.Tan(double b);

double b = Math.Atan(double a);

a 为值 b 为弧度制角度

6，串口收发数据的通信协议

默认串口：115200,0,0

数据传输协议：0XFC 0XCF 图像数据 0XCF 0XFC

例如 1：

```
//UART_PutChar()换成你的串口字节发送函数，串口号改成你的，其他参数得对应上
void UartSendReport(int img[][],int height,int width)
{
    UART_PutChar(UART4,0xFC);
    UART_PutChar(UART4,0xCF);
    for(int i = 0;i<height;i++)
    for(int j = 0;j<width;j++)
    {
         //传灰度图时防接错，二值化图不用担心
         if(img[i][j]==0xCF)
         {
             img[i][j]=0xCE;
         }
      UART_PutChar(UART4,img[i][j]);
    }
    UART_PutChar(UART4,0xCF);
    UART_PutChar(UART4,0xFC);
}
```

例如 2：

```
//    @brief          总钻风摄像头图像发送至上位机查看图像
//    @param          uartn 使用的串口号
//    @param          image 需要发送的图像地址
//    @param          width 图像的列
```

```
//    @param            height 图像的行

void  seekfree_sendimg_03x(UARTN_enum  uartn,  uint8  *image,  uint16  width,
uint16  height)
{
         uart_putchar(uartn,0xFC);//发送命令
        uart_putchar(uartn,0xCF);
        uart_putbuff(uartn,  image,  width*height);    //发送图像
        uart_putchar(uartn,0xCF);
        uart_putchar(uartn,0xFC);
}
```