

总钻风全局快门摄像头 使用说明

目录

目录	1
1. 总钻风优点简介	4
1.1. 全局快门.....	4
1.2. 高动态性能	5
1.3. 用户无需自己编写驱动	5
1.4. 自动和手动曝光可配置	5
1.5. FPS 可调	6
2. 总钻风注意事项	7
3. 总钻风硬件规格参数.....	8
4. 总钻风使用教程	9
4.1. FFC 排线连接	9
4.2. DCMI 或者 DVP 接口采集硬件修改	10
4.2.1. MT9V032	11
4.2.2. MT9V034	12
4.3. 上位机显示总钻风图像	13
4.4. 液晶屏幕显示总钻风图像.....	15
5. 总钻风参数详解	18
5.1. 参数解释.....	18

5.2. 设置参数.....	20
5.3. 单独设置曝光时间.....	21
5.4. 读取摄像头状态	21
6. 总钻风参数调节说明（线下比赛场地）	23
6.1. 一般体育馆	23
6.2. 信标组.....	23
7. 总钻风寄存器操作方法	24
7.1. 使用串口配置摄像头寄存器	24
7.2. 使用 IIC 自行配制摄像头参数.....	25
7.2.1. MT9V032	25
7.2.2. MT9V034	26
8. 总钻风相关推文	28
9. 常见问题.....	29
9.1. 总钻风如何接线	29
9.2. 如何使用图像发送到上位机例程.....	29
9.3. 使用上位机例程看不到图像或者图像出错	29
9.3.1. 硬件检查.....	29
9.3.2. 软件检查.....	30
9.3.3. 图像异常.....	30
9.4. 如何二值化	31

9.5. 图像效果不好、有暗纹	31
9.6. 为什么上位机显示这么慢.....	31
10. 文档版本	32

1.总钻风优点简介

1.1.全局快门

总钻风采用的是 MT9V034/ MT9V032 芯片，CMOS 全局快门 (Global Shutter)：在曝光前整个图像重置，像素可在曝光时间积累电荷；曝光结束后，每个像素积累的电荷同时传送到屏蔽光（对光不敏感）的存储区域；然后信号从此区域读出。因所有像素同时重置，曝光积分同样的间隔，同时传输到光屏蔽存储区域，对移动物体来说没有形变。



用一幅示意图来展示动态模糊、卷帘快门、全局快门拍摄小车高速运行的效果：



对于许多担心和苦恼小车在高速运动下，摄像头采集到的图像会失真的智能车大神，我们给您带来了福音，相信我们的这款全局快门 (Global Shutter) 的高性能摄像头会消除您的疑虑和担忧，会给您带来不一样的感受！

1.2.高动态性能

高动态范围简称 HDR。



高动态范围 (HDR) 图像为我们呈现了一个充满无限可能的世界, 因为它们能够表示现实世界的全部可视动态范围。由于可以在 HDR 图像中按比例表示和存储真实场景中的所有明亮度值, 因此, 调整 HDR 图像的曝光度的方式与在真实环境中拍摄场景时调整曝光度的方式类似。HDR 是目前追求画面逼真度最新最先进的手段

简单通俗来说的话, 就是在能看清暗处的同时, 亮处也不会轻易过曝。

1.3.用户无需自己编写驱动

总钻风摄像头内部集成了驱动配置芯片, 所以用户不需自己写摄像头驱动程序。通过串口即可非常简单的对摄像头参数进行配置。详细介绍参考后面的如何配置摄像头参数配置章节。

1.4.自动和手动曝光可配置

我们的摄像头具有自动曝光功能, 当环境变亮摄像头会自动减小曝光时间, 当环境变暗摄像头会自动增加曝光时间。所以此款摄像头可以适应不同环境。在我们的例程中自动曝光默认

是关闭的,因为在竞赛的时候,开启自动曝光可能会导致图像中赛道和背景的灰度值发生变化,不利于二值化或灰度的计算。

在关闭自动曝光后,可以观察图像的亮度来调整程序中的曝光时间,使得图像不会过亮或者过暗,这样会比自动曝光更利于算法的处理。

这里需要特别注意,有部分同学喜欢使用 OLED 来显示二值化之后的图像,理由是图像已经软件二值化了,如果用 TFT 之类的屏幕显示图像则比较浪费,但是场地光线很强的时候,我们就只能看到二值化后的图片全部是亮的,并且修改二值化阈值也不能取得很好的效果,由于无法查看灰度图从而不知道原始图像是否正常,比赛的时候又非常的短,很容易导致比赛失利。因此显示图像推荐使用 TFT、IPS 之类的屏幕,并且程序能随时调节显示灰度图像。

1.5.FPS 可调

软件可调 FPS,我们的总钻风摄像头可在 1-500 帧之间自由调节。这里需要特别说明,一般用户如果没有特殊需求建议设置为 50 帧或者 100 帧,以此避免工频干扰,有关工频干扰可以自行百度。一般来说 50 帧完全足够控制智能车运行,再快意义不大,因为舵机是一个响应速度较慢的执行机构,如果是直立车想要提高响应速度也可以将帧率设置为 100 帧,但是需要注意自己的处理算法应该在下一次的图像来之前处理完成。否则将毫无意义。

2.总钻风注意事项

1. 总钻风电源是 3.3V 的，切记不要接错。如果电源接错，有可能摄像头板子会发热并可能造成永久损坏，这时请立即断电，并检查电源。
2. 摄像头供电不能直接从系统板或下载器取电，需使用外部电源供电。外部电源需要纹波比较小，建议使用独立的线性稳压芯片为摄像头提供电源，电源的稳定性将会影响图像质量。
3. 总钻风必须与单片机共地（使用同一个接地）。
4. 友情提示

控制器分为以下七个类别。

- (1) 32 位 Kinetis (ARM® Cortex™-M0+), 主要包括 Kinetis E, EA, L, M 等系列;
- (2) 32 位 Kinetis (ARM® Cortex™-M4), 主要包括 Kinetis K, W 等系列;
- (3) 32 位 MPC56xx 系列;
- (4) 16 位 9S12 系列;
- (5) 32 位 ColdFire 系列;
- (6) DSC 系列;
- (7) 8 位单片机系列;

同一所学校在报名的同一组别的两支队伍，必须使用不同系列的处理器芯片。如果在提高类别的两个组别（A1,A2）中，所选择的车型种类不同，则同一组别的两支队伍可以使用相同系列的微处理器。

- 使用微控制器的数量没有限制。
- 如果所选用的传感器或其它电子部件中也包含有微处理器，对此微处理器的种类和数量不做限制，但其不得参与对于赛道信息识别和处理、不参与车模运动决策与控制。

根据 2016 年恩智浦智能车竞赛新规定，我们的总钻风摄像头使用的 51 单片机只是用于摄像头寄存器配置，不参与赛道信息的识别和处理，更不参与车模运动决策与控制，所以总钻风摄像头是符合主委会规定的一款可用于比赛的高性能摄像头，亲们可以放心大胆的使用。

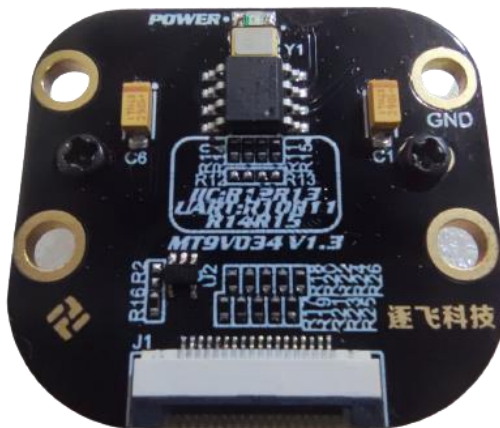
3.总钻风硬件规格参数

1. 供电电压：3.3V
2. 信号输入输出电平：3.3V CMOS 电平
3. 图像信号协议：DVP
4. 控制信号协议：UART
5. 镜座固定孔距：22mm

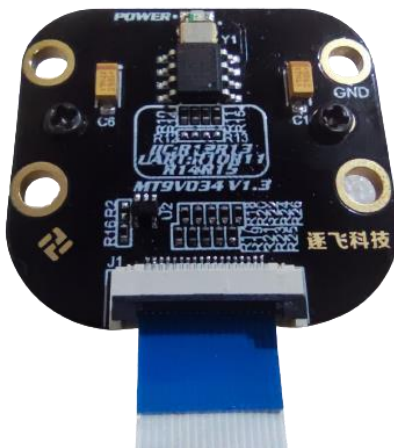
4.总钻风使用教程

4.1.FFC 排线连接

第一步：将 FPC 座子的卡扣向上提如下图所示。



第二步：将 FFC 排线插入 FPC 座子中，如下图所示

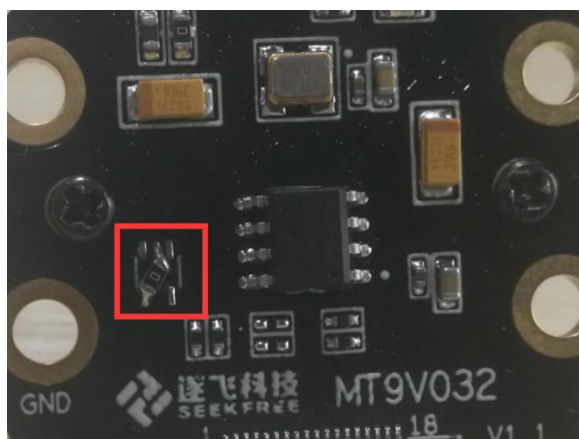


第三步：将 FPC 座子卡扣推入

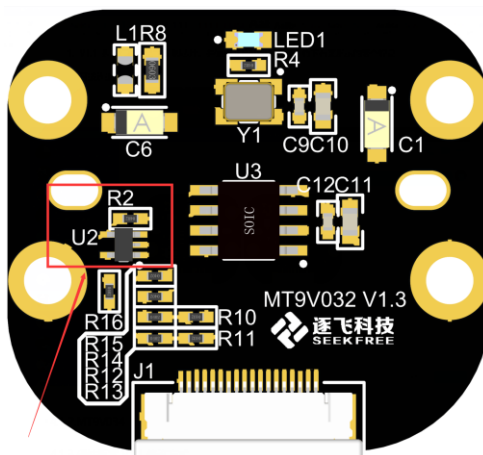
不同硬件版本的总钻风摄像头，修改方式存在一定区别，请参考以下说明。

4.2.1.MT9V032

1. V1.1 拆掉红色框里的 5 脚芯片。并使用 0 欧电阻或导线，将下右图所示的两个焊盘直接连接在一起。



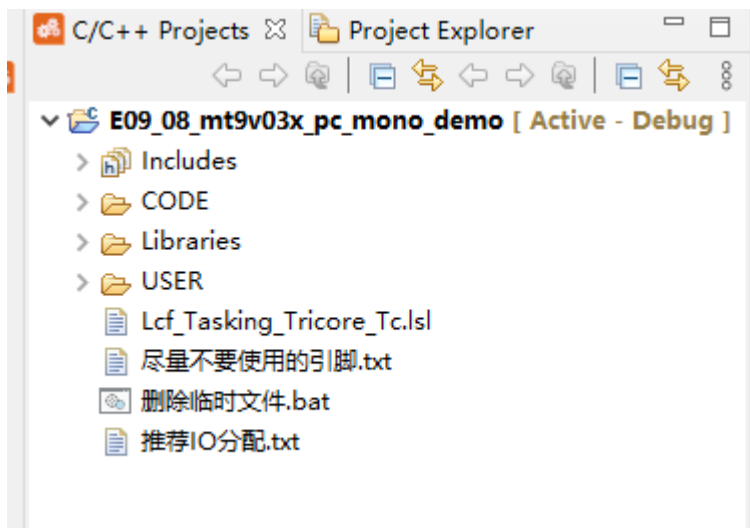
2. V1.3 去掉 U2 芯片，使用 0 欧电阻焊上 R2



3. V1.5 去掉 U2 芯片，使用 0 欧电阻焊上 R2

4.3.上位机显示总钻风图像

1. 打开 AURIX Development Studio 软件，在我们的例程中导入 E09_08_mt9v03x_pc_mono_demo 例程



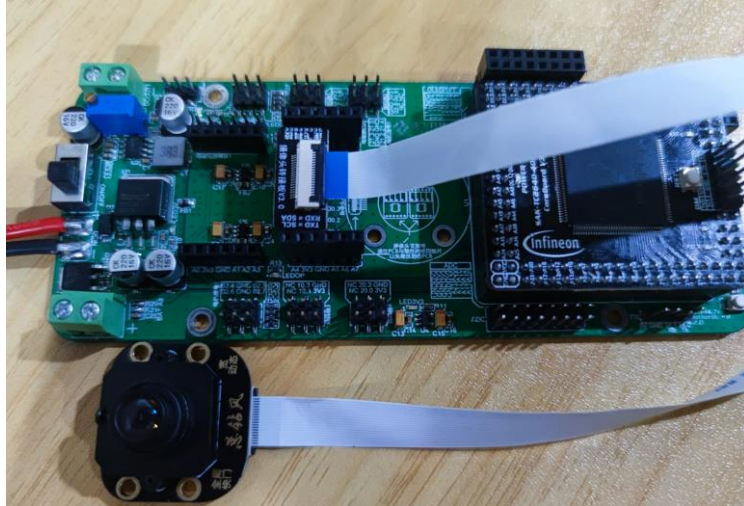
2. 根据 Libraries/zf_device/zf_device_mt9v03x.h 头文件的注释接线

```

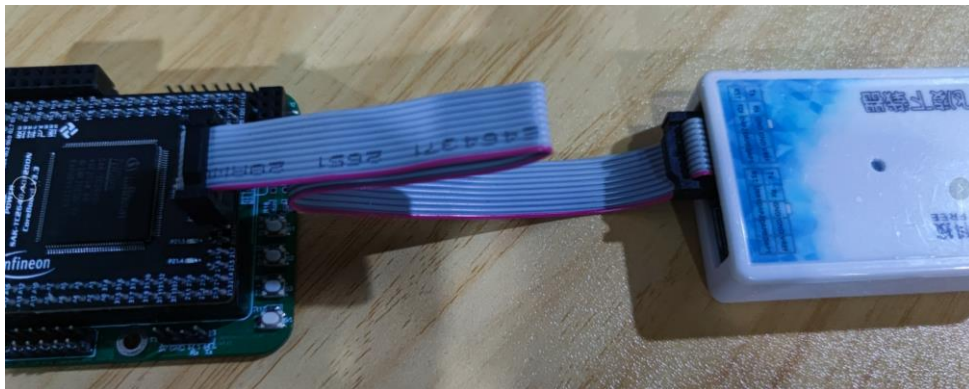
/*****
* 接线定义:
*
* 模块管脚      单片机管脚
* TXD           查看 zf_device_mt9v03x.h 中 MT9V03X_COF_UART_TX 宏定义
* RXD           查看 zf_device_mt9v03x.h 中 MT9V03X_COF_UART_RX 宏定义
* PCLK          查看 zf_device_mt9v03x.h 中 MT9V03X_PCLK_PIN 宏定义
* VSY           查看 zf_device_mt9v03x.h 中 MT9V03X_VSYNC_PIN 宏定义
* D0-D7         查看 zf_device_mt9v03x.h 中 MT9V03X_DATA_PIN 宏定义 从该定义开始的连续八个引脚
* VCC           3.3V电源
* GND           电源地
*
* 其余引脚悬空
*****/

```

这里使用学习板，将摄像头插到对应的接口上



3. 将下载器连接到单片机，注意下载器接口方向

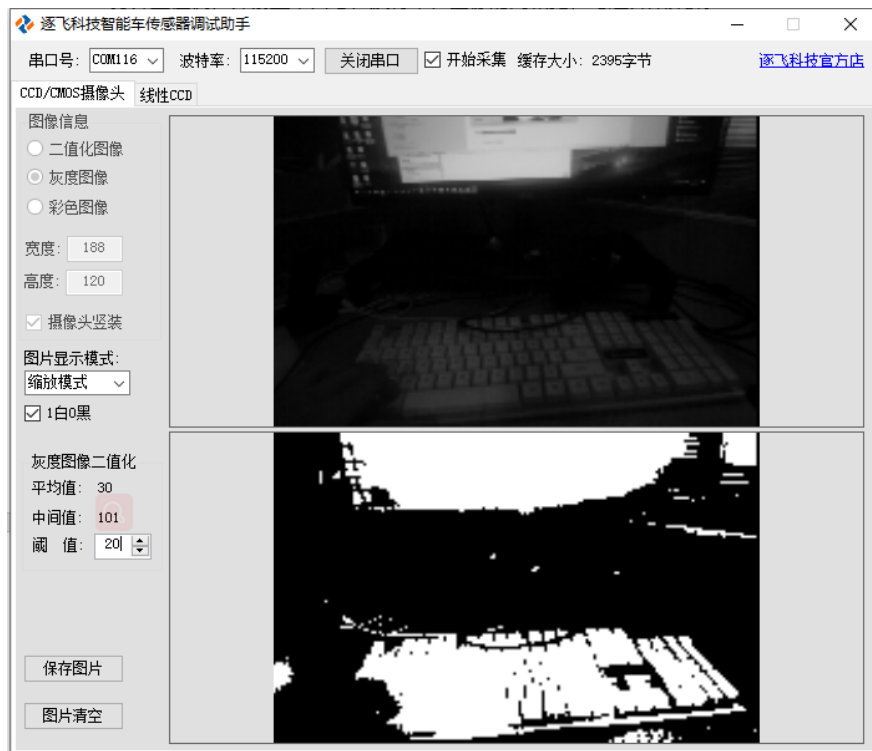


4. 下载程序到单片机

5. 打开上位机，并设置串口号，波特率，图像的高和宽，勾选开始采集

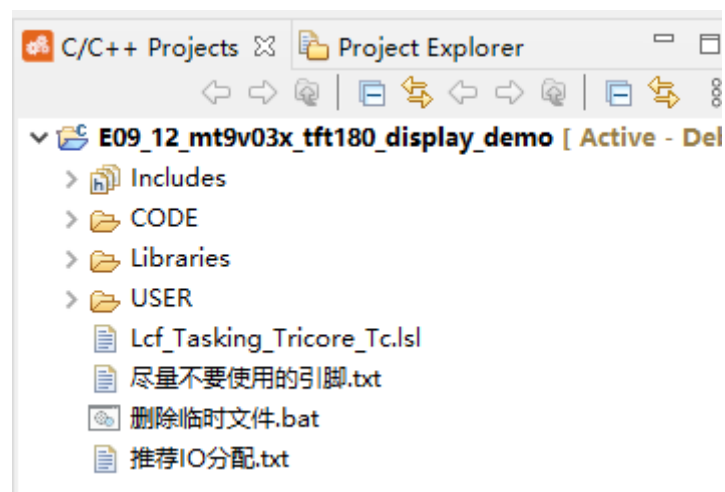


6. 使用电源上电后，上位机打开串口，可以看到图像



4.4.液晶屏幕显示总钻风图像

1. 打开 AURIX Development Studio 软件，在我们的例程中导入 E09_12_mt9v03x_tft180_display_demo 例程



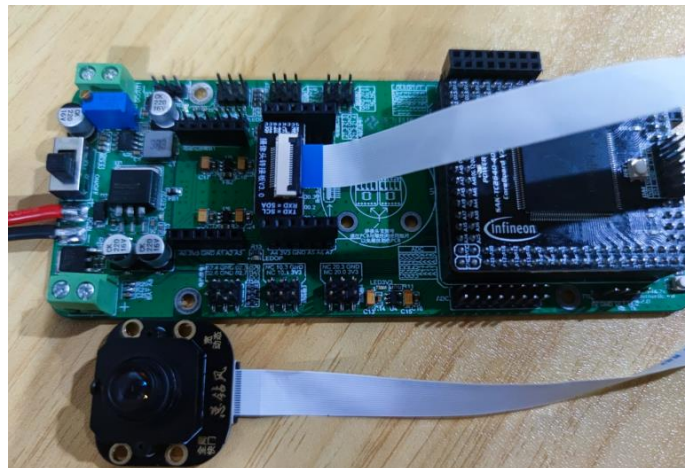
2. 根据 Libraries/zf_device/zf_device_mt9v03x.h 头文件的注释接线


```

/*****
* 接线定义:
*
* 模块管脚      单片机管脚
* TXD            查看 zf_device_mt9v03x.h 中 MT9V03X_COF_UART_TX 宏定义
* RXD            查看 zf_device_mt9v03x.h 中 MT9V03X_COF_UART_RX 宏定义
* PCLK           查看 zf_device_mt9v03x.h 中 MT9V03X_PCLK_PIN 宏定义
* VSY            查看 zf_device_mt9v03x.h 中 MT9V03X_VSYNC_PIN 宏定义
* D0-D7          查看 zf_device_mt9v03x.h 中 MT9V03X_DATA_PIN 宏定义 从该定义开始的连续八个引脚
* VCC            3.3V电源
* GND            电源地
*
* 其余引脚悬空
*
*****/

```

这里使用学习板，将摄像头插到对应的接口上



显示屏也接到主板上，如果没有用学习板，也按照头文件的注释进行接线

```

/*****
* 接线定义:
*
* 模块管脚      单片机管脚
* SCL            查看 zf_device_tft180.h 中 TFT180_SCL_PIN 宏定义
* SDA            查看 zf_device_tft180.h 中 TFT180_SDA_PIN 宏定义
* RES            查看 zf_device_tft180.h 中 TFT180_RES_PIN 宏定义
* DC             查看 zf_device_tft180.h 中 TFT180_DC_PIN 宏定义
* CS             查看 zf_device_tft180.h 中 TFT180_CS_PIN 宏定义
* BL             查看 zf_device_tft180.h 中 TFT180_BL_PIN 宏定义
* VCC            3.3V电源
* GND            电源地
*
* 最大分辨率160*128
*
*****/

```



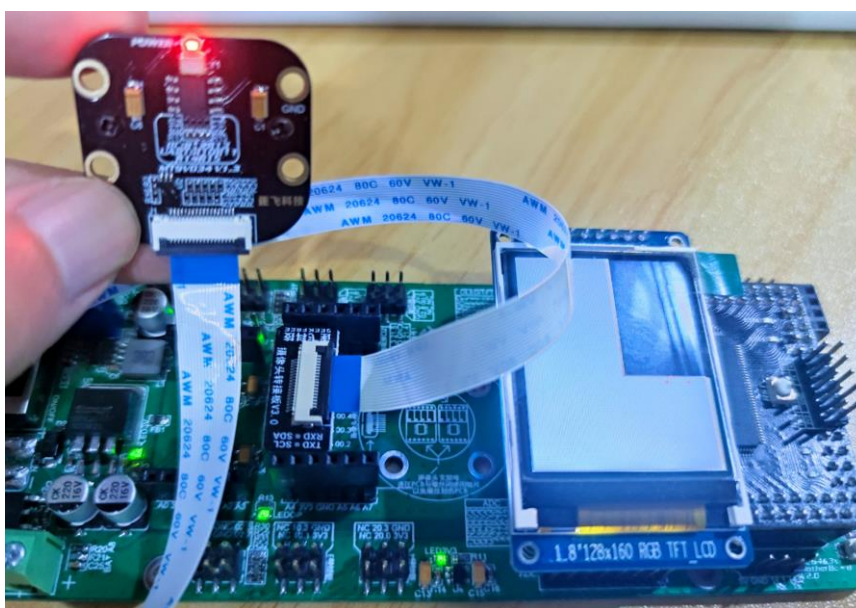
3. 将下载器连接到单片机，注意下载器接口方向



4. 下载程序到单片机，注意代码的位置需要将这两个参数除以 2，因为图像是 188*120 宽度大于了屏幕像素 160*128，会导致显示出错。也可以自定义显示的宽度与高度，但是不能超过屏幕的显示范围即可，例如设置为宽度 160、高度 120。

```
cpu_wait_event_ready();
while (TRUE)
{
    // 此处编写需要循环执行的代码
    if(mt9v03x_finish_flag)
    {
        tft180_displayimage032((const uint8 *)mt9v03x_image, MT9V03X_W/2, MT9V03X_H/2); // 5
        mt9v03x_finish_flag = 0;
    }
    // 此处编写需要循环执行的代码
}
```

5. 上电运行，可以看到屏幕上显示出实时图像



5.总钻风参数详解

5.1.参数解释

1. AUTO_EXP

自动曝光设置：范围：1-63，0 为关闭，如果自动曝光开启 EXP_TIME 命令设置的数据将会变为最大曝光时间，也就是自动曝光时间的上限，建议曝光时间设置为 700。其实在自动曝光开启的时候，曝光时间的作用变成了自动曝光值的最大值，自动调节的时候曝光时间不会大于 700。一般情况是不需要开启这个功能，因为比赛场地光线一般都比较均匀，如果遇到光线非常不均匀的情况可以尝试设置该值，增加图像稳定性。

2. EXP_TIME

曝光时间：摄像头收到后会自动计算出最大曝光时间，如果设置过大则设置为计算出来的最大曝光值。曝光时间越短图像的动态性能越好，因此建议在不影响程序处理图像的前提下越低越好。

3. FPS

图像帧率：摄像头帧率设置，设置 50 即为 50 帧，填写的时候可以按照自己的需求填写，摄像头收到数据后如果超过了能够设置的最大值，摄像头会配置最大值，通过获取配置数据的函数，可以查看具体设置的是多少。建议大家使用 50、100 这两种帧率避免工频干扰。（分辨率在 80*60 时最高 FPS 可达 500 帧）。

4. SET_COL、SET_ROW

图像行列数量：设置图像的分辨率列的范围是 1-752，行的范围是 1-480。最大分辨率需要看单片机是否支持，在提供的示例中也会在驱动头文件中标注最大支持的分辨率。图像分辨

率为 752*480、376*240、188*120，这三种分辨率视野是一样的，因为后面两种分别是通过 752*480 除以 2 和 4 压缩得到的（376*240 就是 752*480 行列除以 2 得到的）。其他分辨率都是通过裁剪得到的，比如 752*400 与 376*240 两个分辨率，376*240 视野反而比 752*400 的视野广，因为 752*400 是通过 752*480 裁剪得到的。

5. LR_OFFSET, UD_OFFSET

在传感器全视野（非裁切图像，如 752*480、376*240、188*120）时无法设置偏移，只有在其他分辨率下才可以是设置偏移。

LR_OFFSET 图像左右偏移量：正值：右偏移，负值：左偏移，列为 188，376，752 时无法设置偏移。

UD_OFFSET 图像上下偏移量：正值：上偏移，负值：下偏移，行为 120，240，480 时无法设置偏移。

摄像头收偏移数据后会自动计算最大偏移，如果超出则设置计算出来的最大偏移。

6. GAIN.

图像增益：范围：16-64，增益可以在曝光时间固定的情况下改变图像亮暗程度。在曝光时间不变的情况下增益设置越大图像越亮。如信标建议将增益设置到最大。

7. PCLK

消隐信号设置：0：不输出消隐信号,1：输出消隐信号，通常都设置为 0，如果使用 CH32V307 的 DVP 接口或 STM32 的 DCMI 接口采集需要设置为 1，**此功能仅在 MT9V034 V2.1 版本及以上的摄像头支持。**

5.2.设置参数

程序通过一个二维数组来保存参数命令和参数数据。每一行的第 1 位是参数命令，第 2 位是参数数据，修改时只需要修改第 2 位即可。

对摄像头进行配置的时候需要按照一定的协议发送数据，数据是一个一个数据包发送的，每一个数据包有 4 个字节，第 0 字节是帧头 0XA5，第 1 字节是命令位，第三字节是数据高八位，第四字节是数据第八位。一个数据包发送后最好延时 2MS。

对摄像头配置是可以采用所有配置数据发送完成然后发送初始化命令，如果想单独修改某个参数可以只修改相关命令的参数，然后发送初始化命令。

```
void set_config(UARTN_enum uartn, uint16 buff[CONFIG_FINISH-1][2])
{
    uint16 temp, i;
    uint8 send_buffer[4];
    uart_receive_flag = 0;
    //设置参数 具体请参看问题锦集手册
    //开始配置摄像头并重新初始化
    for(i=0; i<CONFIG_FINISH; i++)
    {
        send_buffer[0] = 0xA5;
        send_buffer[1] = buff[i][0];
        temp = buff[i][1];
        send_buffer[2] = temp>>8;
        send_buffer[3] = (uint8)temp;
        uart_putbuff(uartn, send_buffer, 4);
        systick_delay_ms(2);
    }
    //等待摄像头初始化成功
    while(!uart_receive_flag);
    uart_receive_flag = 0;
    while((0xff != receive[1]) || (0xff != receive[2]));
    //以上部分对摄像头配置的数据全部都会保存在摄像头上51单片机的eeprom中
    //利用set_exposure_time函数单独配置的曝光数据不存储在eeprom中
}
```

上图所示为先发送全部分配置数据然后发送初始化命令,当摄像头接收到初始化命令就会开始用刚才接收到的配置数据进行初始化,初始化完成后会发送一个数据包回来,数据包有三个字节,第0字节是帧头 0xA5,第1字节是 0xFF,第三字节是 0xFF。

5.3.单独设置曝光时间

使用 `set_exposure_time` 函数可以非常简单的实现单独设置曝光时间。此参数不会被保存,掉电就会丢失。

```
uint16 set_exposure_time(UARTN_enum uartn, uint16 light)
{
    uint16 temp;
    uint8 send_buffer[4];
    send_buffer[0] = 0xA5;
    send_buffer[1] = SET_EXP_TIME;
    temp = light;
    send_buffer[2] = temp>>8;
    send_buffer[3] = (uint8)temp;
    uart_putbuff(uartn,send_buffer,4);
    //等待接受回传数据
    while(!uart_receive_flag);
    uart_receive_flag = 0;
    temp = receive[1]<<8 | receive[2];
    return temp;
}
```

5.4.读取摄像头状态

使用 `get_config` 函数可以轻易的读取摄像头配置信息读取出来的数据保存在 `GET_CFG` 数组里面。

```
void get_config(UARTN_enum uartn, int16 buff[CONFIG_FINISH-1][2])
{
    uint16 temp, i;
    uint8 send_buffer[4];
```

```
for(i=0; i<CONFIG_FINISH-1; i++)
{
    send_buffer[0] = 0xA5;
    send_buffer[1] = GET_STATUS;
    temp = buff[i][0];
    send_buffer[2] = temp>>8;
    send_buffer[3] = (uint8)temp;
    uart_putbuff(uartn,send_buffer,4);
    //等待接受回传数据
    while(!uart_receive_flag);
    uart_receive_flag = 0;
    buff[i][1] = receive[1]<<8 | receive[2];
}
}
```

6.总钻风参数调节说明（线下比赛场地）

6.1.一般体育馆

体育馆的光线会比实验室的亮度更大，可以先将自动曝光设置为 0 关闭自动曝光，然后根据图像降低曝光时间，看到图像颜色的深浅比较合适即可。

6.2.信标组

通常信标组所在的场地都是比较封闭的，并且所用的摄像头都是信标专用的摄像头，因此摄像头只能看见红外光，为了使摄像头看的更远，我们可以在不改变图像帧率的前提将曝光与增益设置到最大，例如帧率 50 的时候，曝光时间设置为 700、增益设置为 64。使得图像尽可能的亮一些，自然也就能看到更远的信标了。

7.总钻风寄存器操作方法

7.1.使用串口配置摄像头寄存器

在默认状态下，使用串口（UART）与总钻风摄像头上的 51 单片机通讯，再由 51 单片机里已有 IIC 的驱动程序对摄像头进行配置。

用户只需要通过串口给出需要操作（操作只能是对寄存器进行写入操作）的寄存器地址与需要写入的数据。使用 `set_mt9v032_reg` 函数可以实现对摄像头内部寄存器进行写入操作，写入完成后摄像头会再次读取该寄存器然后发送回来便于确认是否写入成功。一共发回来一个数据包，一个数据包有三个字节，第 0 字节是帧头 0XA5，第 1 字节是 0XFF，第三字节是 0XFF。

```
uint16 set_mt9v03x_reg(UARTN_enum uartn, uint8 addr, uint16 data)
{
    uint16 temp;
    uint8 send_buffer[4];
    send_buffer[0] = 0xA5;
    send_buffer[1] = SET_ADDR;
    temp = addr;
    send_buffer[2] = temp>>8;
    send_buffer[3] = (uint8)temp;
    uart_putbuff(uartn,send_buffer,4);
    systick_delay_ms(10);
    send_buffer[0] = 0xA5;
    send_buffer[1] = SET_DATA;
    temp = data;
    send_buffer[2] = temp>>8;
    send_buffer[3] = (uint8)temp;
    uart_putbuff(uartn,send_buffer,4);
    //等待接受回传数据
    while(!uart_receive_flag);
    uart_receive_flag = 0;
    temp = receive[1]<<8 | receive[2];
}
```

```
return temp;
```

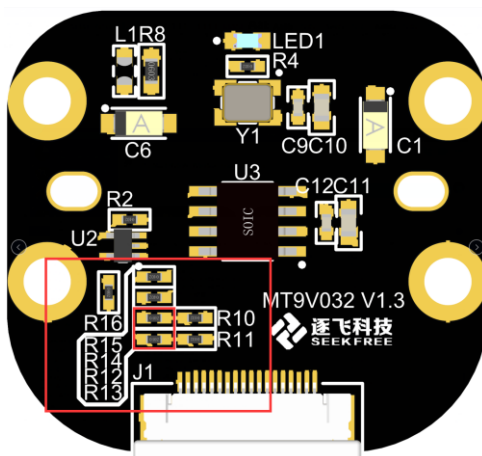
```
}
```

7.2.使用 IIC 自行配制摄像头参数

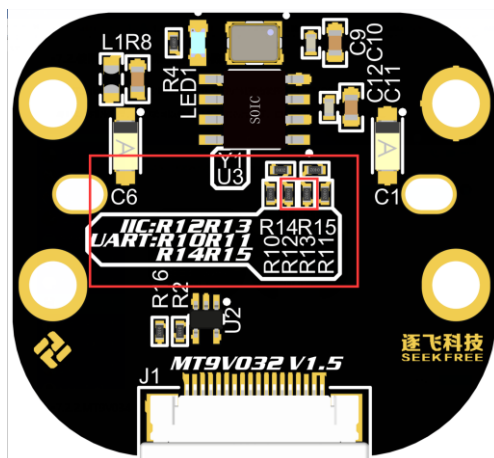
需要修改硬件，修改之后摄像头上的 51 单片机不再发挥作用，CMOS 的 IIC 引脚将直接与 FFC 软排座连通，用户就可以参考 CMOS 的芯片手册，自行使用 IIC 协议配置摄像头。

7.2.1.MT9V032

1. V1.3 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R

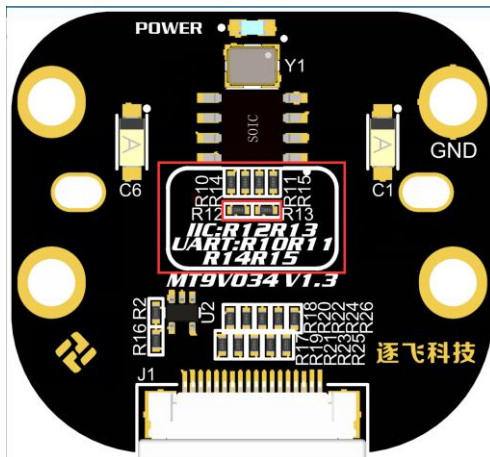


2. V1.5 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R

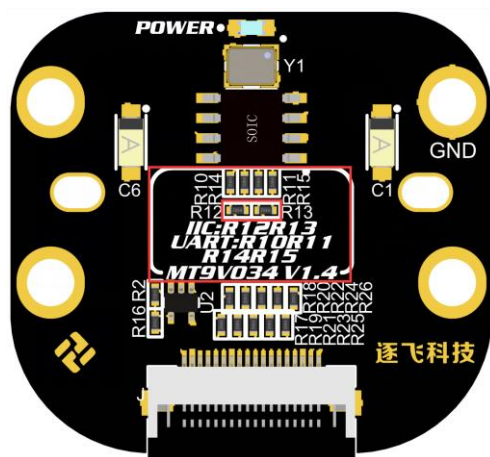


7.2.2.MT9V034

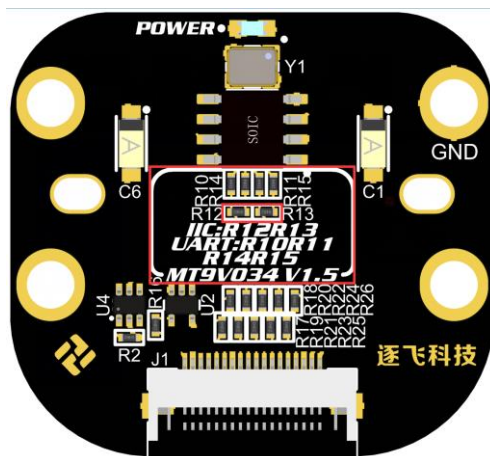
1. V1.3 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R



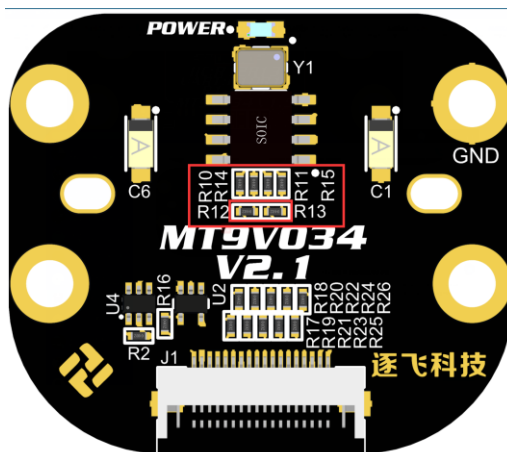
2. V1.4 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R



3. V1.5 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R



4. V2.1 用 IIC 焊 R12 R13 不焊 R10 R11 R14 R15 所需电阻均为 0R



8.总钻风相关推文

[电磁及摄像头（总钻风）寻迹算法浅析--逐飞科技](#)

[基础四轮不基础之总钻风前后双摄--By 逐飞科技](#)

[平衡单车组逐飞演示车模浅析](#)

[四轮摄像头组之逐飞演示车模循迹浅析](#)

[智能视觉组逐飞演示车模浅析-上篇](#)

[逐飞无线充电组演示车模浅析及 LED 电量显示方案开源](#)

[英飞凌基础四轮组浅析--逐飞科技](#)

9.常见问题

9.1.总钻风如何接线

在总钻风的资料中有多个单片机的使用例程，选择自己已有的单片机型号打开例程。在总钻风 MT9V034 的驱动文件中可以看到具体的接线方式。

9.2.如何使用图像发送到上位机例程

1. 先检查是否有例程中的单片机型号，建议使用已有的型号
2. 使用相应的例程，注意一个型号下有多个例程，这里使用连接 USB 转 TTL 发送到上位机的例程。例程如下图所示：
3. 打开驱动文件，根据驱动文件的注释进行接线。
4. 下载例程到单片机中
5. 使用 USB 转 TTL 模块连接单片机和电脑，注意 TTL 的 RX 连接单片机的 TX，TTL 的 TX 连接单片机的 RX。
6. 打开上位机，选择对应串口并打开串口，启动单片机后可以看到图像显示。

9.3.使用上位机例程看不到图像或者图像出错

9.3.1.硬件检查

1. 检查摄像头上的电源指示灯是否常亮
2. 使用万用表的通断档检测接线是否正确
3. 检查 UART 是否连接正确，

- (1) 摄像头的 TX 连接单片机的 RX, 摄像头的 RX 连接单片机的 TX
- (2) 单片机的 TX 连接 USB 转 TTL 的 RX, 单片机的 RX 连接 USB 转 TTL 的 TX

9.3.2.软件检查

1. 上位机是否有接收到数据, 如果接收到数据但没有图像, 看串口是否设置正确, 波特率是否为 115200, 上位机中还需要设置和程序中图像一样的高和宽。如果没有接收到数据则看下一步。
2. 下载器连接单片机后进入调试, 单步运行
3. 如果卡在摄像头初始化的位置无法通过, 需要检查摄像头的 TX 和 RX 是否和单片机连接正确, 总钻风的 RX 引脚需要接单片机的 TX 引脚。总钻风的 TX 引脚需要接单片机的 RX 引脚。
4. 通过了初始化, 但是摄像头的标志位没有置 1, 导致无法执行到串口发送数据的语句。原因是没有进入摄像头中断函数, 检查摄像头的 VSY 和 PCLK 引脚是否连接正确。
5. 如果执行到了串口发送的语句, 但是上位机没有接收到数据, 则检查单片机和上位机的连线是否正确。

9.3.3.图像异常

1. 摄像头的配置串口 (即转接板的 TX、RX 引脚) 没有连接好导致从未对摄像头进行过配置。这个问题主要出现在 V2 版本之前的摄像头中, 因为之前的例程没有接受回传数据校验, 在 V2 版本中如果出现这个问题则会在初始化中持续等待。(这里提示下, 如果在比赛中遇到了配置的串口线突然断了, 这个时候就会造成程序在初始化出不来, 但是比赛又迫在

眉睫不容等待，这个时候我们可以把程序中对摄像头配置的代码屏蔽掉，只保留采集初始化代码，一般也是可以正常工作的，这是为什么呢？因为摄像头上的 51 有 eeprom，每次收到新的配置数据后将会把配置数据写入 eeprom，每次上电的时候加载 eeprom 的配置数据。)

9.4.如何二值化

1. 软件二值化

软件二值化资料包里有一个软件，可以给大家展示效果，比赛应该用大津法的多一点。

2. 硬件二值化

总钻风如何硬件二值化呢，将摄像头的最高位与单片机数据口最低位连接，单片机其他数据口接地。比如例程中单片机用的数据口为 C8-C15，那么摄像头的 D7 与单片机的 C8 连接，单片机 C9-C15 接地。这样采集到的图像不是 0 就是 1。其实这样二值化相当于阈值设置的是 127。然后调节曝光时间来达到调节阈值的效果。效果可能没有软件二值化的好。

9.5.图像效果不好、有暗纹

一般是由于电源的问题引起的，特别是使用电脑 USB 接口供电的最容易引起这个问题，在实际测试中使用 USB 接口供电效果确实不那么好，我们测试摄像头的时候用的电源是 7.2V 电池然后稳压到 3.3V 给摄像头供电的。

9.6.为什么上位机显示这么慢

因为串口传输速度有限，因此上位机刷新的慢。建议购买店铺的 TFT 可以直接配套例程在液晶屏上显示图像，比上位机快的多。

10. 文档版本

版本号	日期	内容变更
V2.1	2019-12-2	CMOS 型号变更为 MT9V034/ MT9V032
V2.2	2020-10-16	订正部分描述错误，增加使用 STM32 采集时 硬件修改方式。
V2.3	2021-3-19	增加修改硬件的提示
V2.4	2021-12-11	增加 CH32V307 DVP 接口使用说明
V2.5	2021-12-15	修复 CH32V307 引脚 D1 和 D4 错误的 问题
V2.6	2022-2-23	修改与门更换的说明
V2.7	2022-9-22	合并调试锦集文档、增加上位机显示图 像说明，重新排版