

智能车摄像头使用手册（舞台已搭好，精彩靠你来）v1.0

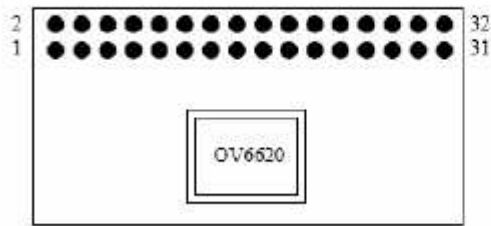
madeby : cumt_liang

一 摄像头的调焦

买回Ov6620后，首先应该把焦距调好，如果焦距调不好，相当于你在眼睛上蒙了一快黑布跑。这个步骤影响到以后图像采集的清晰与否，是以后工作的基础。这个是基础中的基础！！

具体的方法如下：

1 硬件连接



图一 6620 引脚图

各引脚功能如下

PIN1-PIN8 灰度信号输出接口Y0-Y7

PIN11 SCCB数据接口 SDA

PIN12 奇偶场同步信号 FODD

PIN13 SCCB数据时钟 SCL

PIN14 行中断信号 HREF

PIN16 场中断信号 VSYN

PIN18 像素同步信号 PCLK(也叫TCLK)

Pin20 工作电压5V

Pin22 工作电压5V

Pin21 工作电压0V

PIN32 模拟信号输出接口 VTO

这32个引脚中，你需要把pin20接电源的正，pin21接电源的地。这个一定不能反！！否则烧掉摄像头！然后把pin32和电源的地两根线接到电视上，可以用图2所示的莲花头，或者是随便2根线接到电视的视频输入上就可以。这个时候电视就能出图像了。

2 调试方法

硬件连接好后，就要调整焦距了，具体可以对着某书上的字来调，字最清晰了就算调好了，这个时候稍微把6620上的螺丝拧好。固定住焦距。



图2 莲花头

二 摄像头的定标

买回摄像头碰到的第2个问题就是如何准确知道单片机采集的图像信息，这里有2种方法一种是静态的图像，一种是运动捕获的实时图像。智能车大部分的工作比如黑线提取，曲率计算，交叉线，起跑线识别都可以在静态的情况下处理。目前调试图像采集有两种方法：

- 1 用bdm调试,看单片机内部ram，缺点是不直观，麻烦。
- 2 大部分人还是喜欢用labview或者串口输出到电脑上来看。用这种方法目前大部分遇到的问题是图像采集的是好的，你得输出却打乱了行场的中断，导致输出的图像行或场发生错行，我们提供专门的图像采集输出工程，保证所看就是单片机的采集。调试的状态怎么才能无缝的连接到运行状态呢，本模块中只要改动DEBUGE_SHOW_RUN =0就是调试，图像输出状态，DEBUGE_SHOW_RUN =1就是图像不输出，实时运行状态。表示利用此模块你可以调整图像采集的左右宽度，最远前瞻，计算曲率，识别交叉线，起跑线。本模块提供功能有a 图像采集，滤波b 二值化 c 串口输出 d 后期曲率计算，交叉线，起跑线识别接口，也说你只要在三个接口函数写你自己的处理函数就可以了。使用本模块的步骤：

(1) 打开串口调试助手



(2)将串口线连接到电脑的串口 1 和单片机，设置通行波特率为 9600

经过这2步你就可以开始你的智能车之旅了：）！

本部分能完成智能车的绝大部分调试，试验效果如下所示。

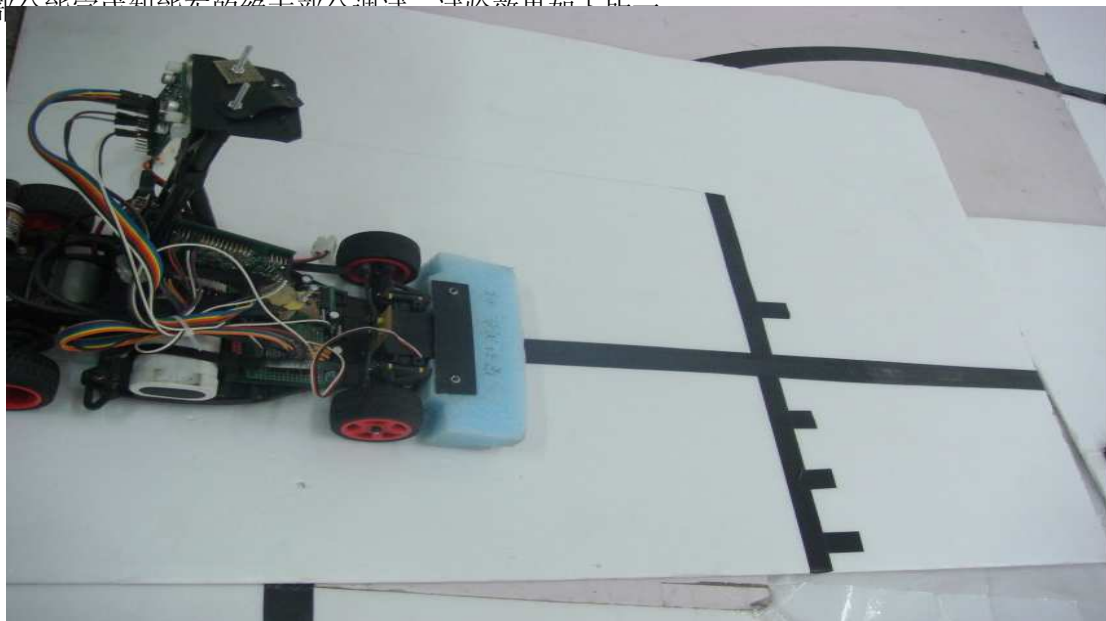


图 3 实物图

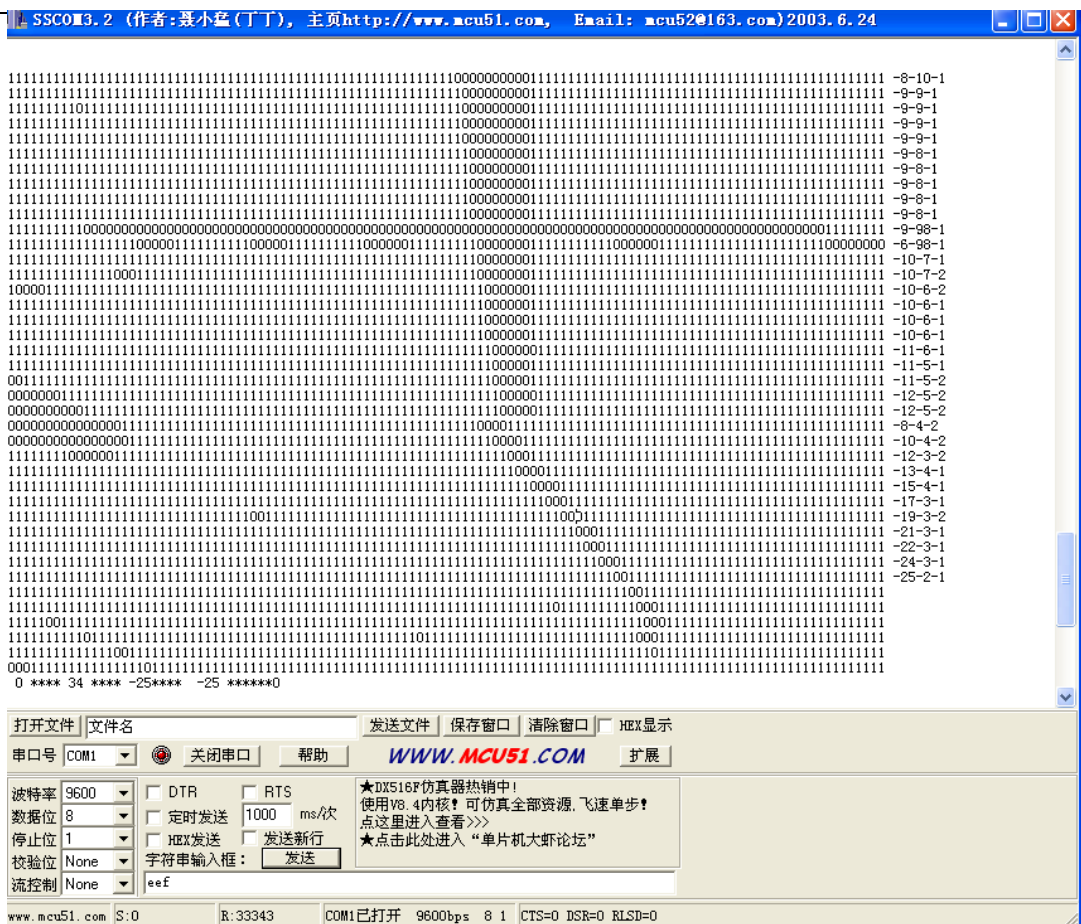


图 4 采集效果图

三 摄像头的定标

图像采集好了，摄像头的左右前后是不是安装好了？对于摄像头的安装也是一个大问题，怎么能准确方便的安装。因为一个微小的角度对摄像头的影响很大。首先你要自己贴个图5这样的板子，专门标定你得摄像头的安装，如果你得摄像头采集的图像是正的和实际的效果有差别你就继续调。直到电脑上看到的是好的就算成功安装好了。板子如下，测试效果如图4所示，每个直线都是直的，就说明你的前后左右都安好了。



图5 调试板

四 让智能车跑起来

在我们提供的工程（提供2种工程9s12和xs12）都是基于嵌入式 uc/os的平台，关于前后台程序和嵌入系统得区别大家可以网上搜搜看看，本平台把基础的电机驱动，舵机驱动，中断，定时器等都作好，并经过了残酷验证。所有的这些并不能让你的车飞速起来，跑的快慢还要有核心算法，但是这里面的控制时序很重要，时序搞不好，车就会跑不好。本工程把时序搞好，剩下的核心算法留出接口，共有转向，速度，停车接口。也就是说你只要在我们留得这三个函数种写你的算法车就按你的想法跑。同时我们提供一套演示用法的工程，你可以以此为平台，也可以再搭个舞台。

五 智能车跑的更好

如果你觉得车跑得不理想，总想看车跑起来的实际信息，有人想到用无线串口来发送信息，可是串口的发送极大的降低了车的实时性，所以上大曾经用sd卡来记录小车运动的时候采集的录像，这个首先要加入Sd卡，额外增加硬件。我们提供一套接口，让你在不做任何硬件的基础上达到这种实时查看信息的目的。你如果想看智能车跑得一圈的路况信息，你只要把写信息的函数打开就可以了，然后把信息记录下来，用我们的配套软件就会放映出车的跑的信息。可以说本方法很好的完成动态看车跑的功能。具体效果如图6所示。

六 测速模块

让车跑的更好一个关键的部分就是你要知道车的速度。没有速度的反馈你的车不会跑的很好，一个简单的例子就是你碰到上坡的时候没有加油，你的车可能会一直在坡上休息。我们提供一个专门的测速模块，这个模块能清晰地显示你的速度，具体事例如下：

1前期准备，测试空载时候占空比和所测的脉冲的比。确定控制的上下限

实验测得数据

占空比 脉冲

15	63-74
20	150-145
30	304-307
40	434-437
45	497-500
50	552--559
55	614--620
60	675-680
65	730-752
99	1118—112

2 调试pid

(1) 确定比例系数Kp

确定比例系数Kp时，首先去掉PID的积分项和微分项，可以令Ti=0、Td=0，使之成为纯比例调节。输入设定为系统允许输出最大值的60%~70%，比例系数Kp由0开始逐渐增大，直至系统出现振荡；再反过来，从此时的比例系数Kp逐渐减小，直至系统振荡消失。记录此时的比例系数Kp，设定PID的比例系数Kp为当前值的60%~70%。

系统出现振荡:本系统中这种现象是指出现两个极端值。比如本测试程序通过串口输出控制的最小10最大100.如果输出的数据仅仅是这两个数，那么说明超调了。出现震荡。如果出现的值在10-100之间比较均匀说明调试成功。下面测试锁速280测的数据。红色数据超调，蓝色没有震荡

213 ----100

205 ----100

266 ----100
319 ----0
305 ----0
210 ----100
261 ----100
316 ----0
300 ----0
212 ----100
201 ----100

233 ----30
208 ----65
246 ----80
306 ----100
295 ----10
299 ----100
297 ----10
301 ----100
295 ----10
239 ----30
213 ----60
245 ----75
302 ----10
303 ----100

这里要注意什么系统出现震荡，及其现象。经确定 $k_p=1.5$

2) 确定微分时间常数 T_d 微分时间常数 T_d 一般不用设定，为0即可，此时PID调节转换为PI调节。如果需要设定，则与确定 K_p 的方法相同，取不振荡时其值的30%。确定 $k_d=2$

3) 系统空载、带载联调

对PID参数进行微调，直到满足性能要求

```
Void Set_Speed_pd()
{
    Int s_ek,s_dek;
    s_ek=(exspeed-speed_vari); //速度偏差
    s_dek=s_ek-lasts_ek;      //速度偏差变化量
    ctrl_motor = pwm_temp;
    ctrl_motor=ctrl_motor+(MP*s_ek)/2+MI*s_dek/10;
    lasts_ek=s_ek;
    if(ctrl_motor< 5) ctrl_motor=5;//0;//
    if(ctrl_motor> 100) ctrl_motor= 100;
    PWMDTY2 = ctrl_motor;
    pwm_temp = PWMDTY2; //正转占空比的影子
}
```

舞台已搭好，精彩靠你来！！！！

cumt_liang 的小店

<http://shop57663993.taobao.com/>