

## 两轮自平衡车

算法：和大家的一样，一个倾角环，一个车速环。取得角度、角速度、车速、车位移四个量后经过运算送给 PWM 驱动电机。

硬件：

主控：atmega16；

角度传感器：角速度传感器(陀螺仪)ENC-03MB(直接接 AD 输入，未加硬件滤波)、加速度传感器 MM A7260，二者 kalman 融合取得角度、角速度。PS：抄 z1stone 的，呵呵。

电机速度传感器：每个电机两个霍尔传感器（AB 相）。

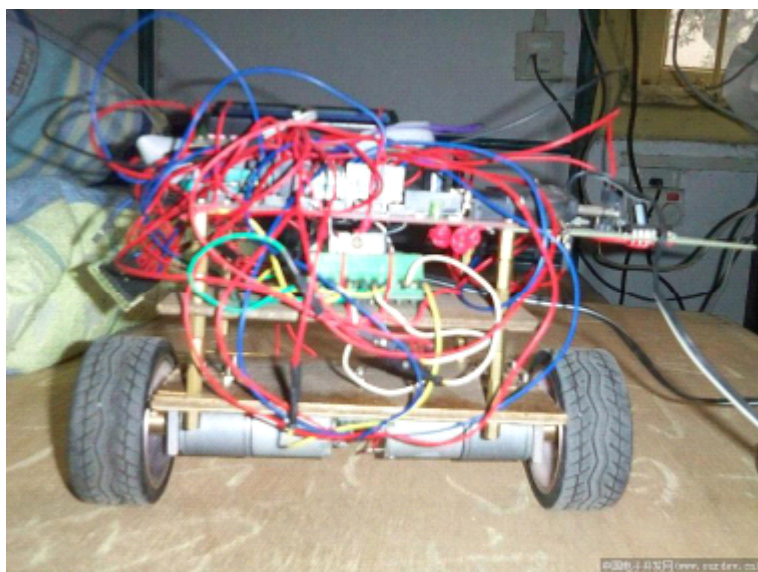
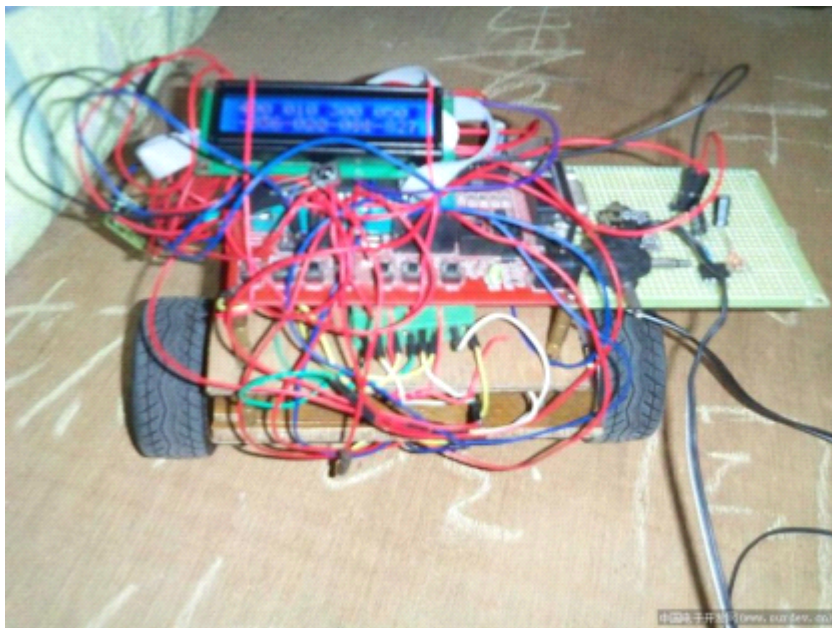
电机：型号不清楚，很常见的减速电机。额定电压 6V，功率 3W。

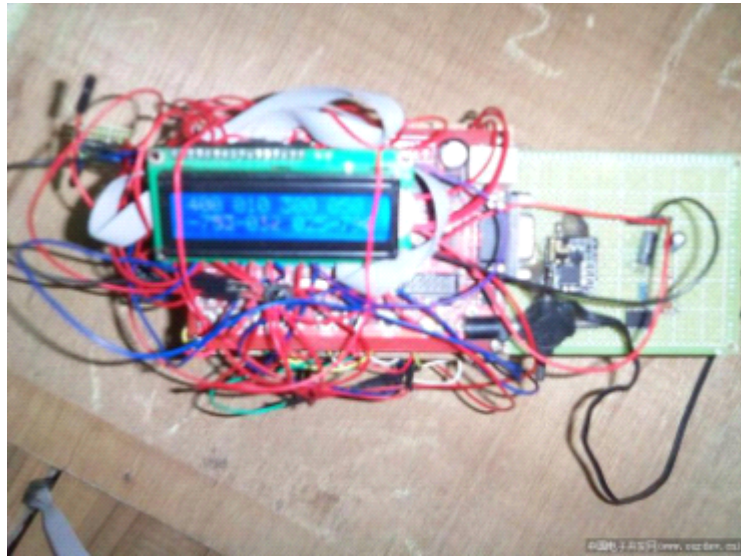
电机驱动:L298N

电源：变压器整流桥那种普通电源，几块钱一个。两个，电机、MCU 分开供电。电机电源电压打到最高不接电机时 15V 多，接了电机 5V 多，汗。。

显示器：LCD1602B

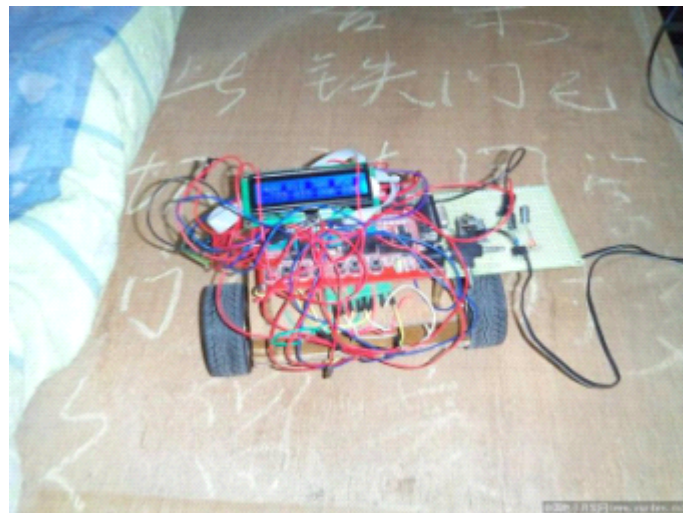
遥控：电视红外遥控器





(原文件名:20110110\_0103. jpg)

引用图片



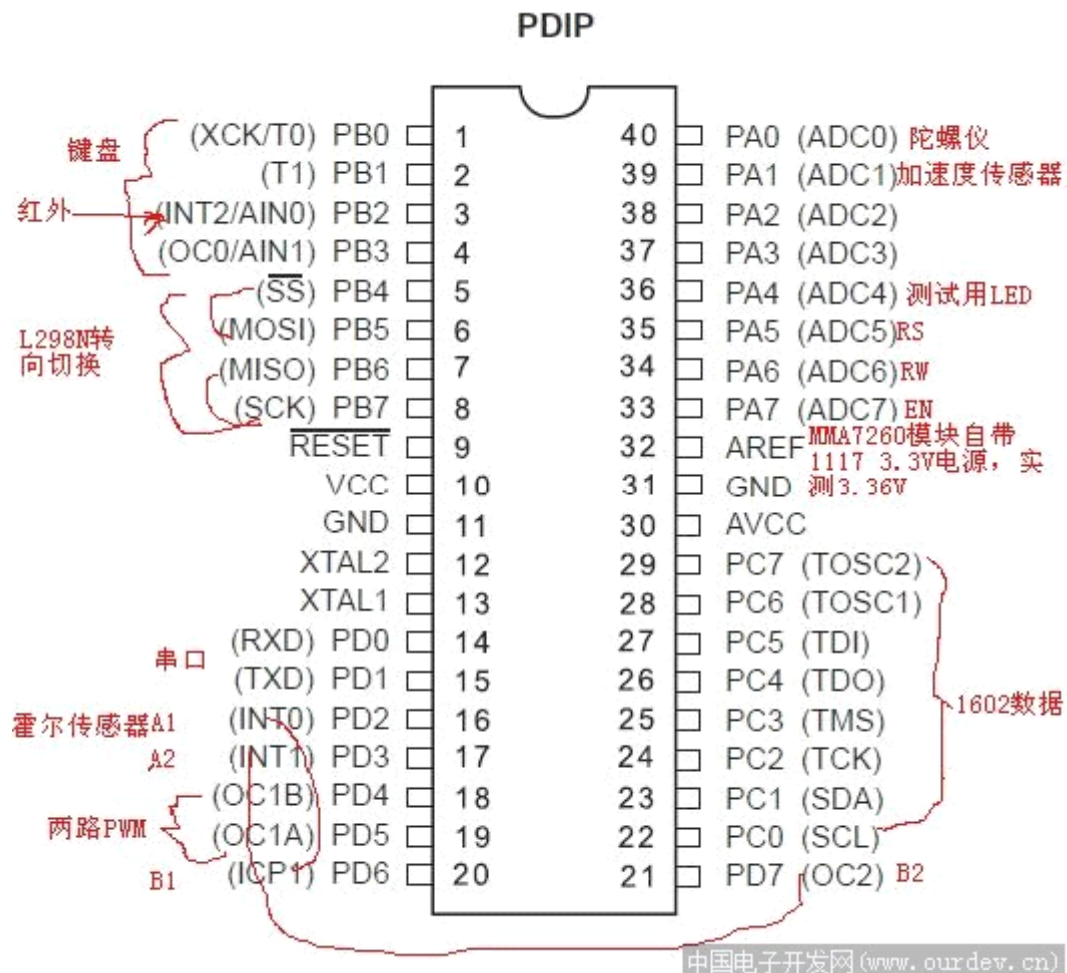
(原文件名:20110110\_0104. jpg)

引用图片

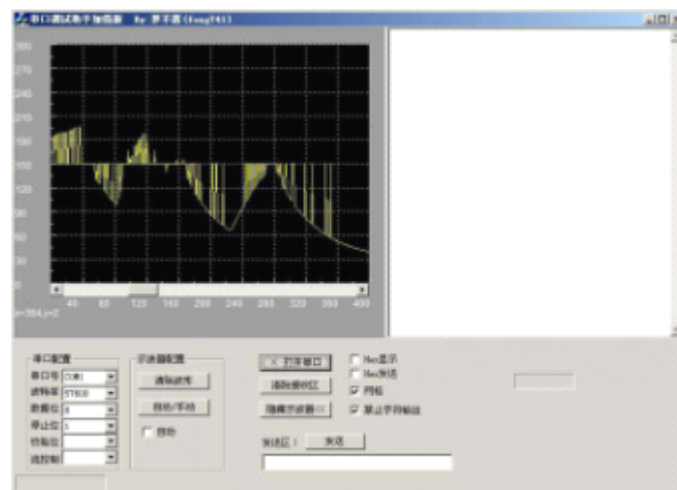
源代码 WINAVR20100110+AVRStudio4.18[ourdev\\_610434C8FD1C.rar](#)(文件大小:104K) (原文件名:Balance.rar)

原理图:

atmega16 最小系统版 [ourdev\\_610214M890EI.pdf](#)(文件大小:30K) (原文件名:M16 迷你板电路图.pdf)

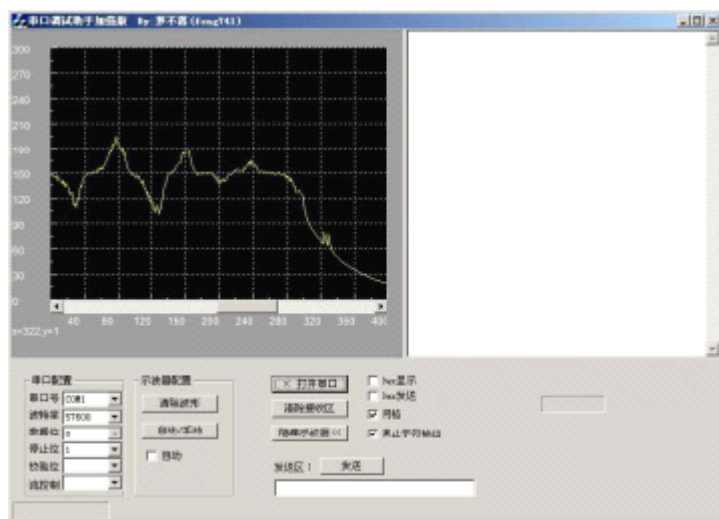


上位机，带波形、数据显示 [ourdev\\_610318TY8G24.rar](#) (文件大小:48K) (原文件名:串口调试.rar)



车速未滤波之前波形 (原文件名:车速未滤波之前波形.JPG)



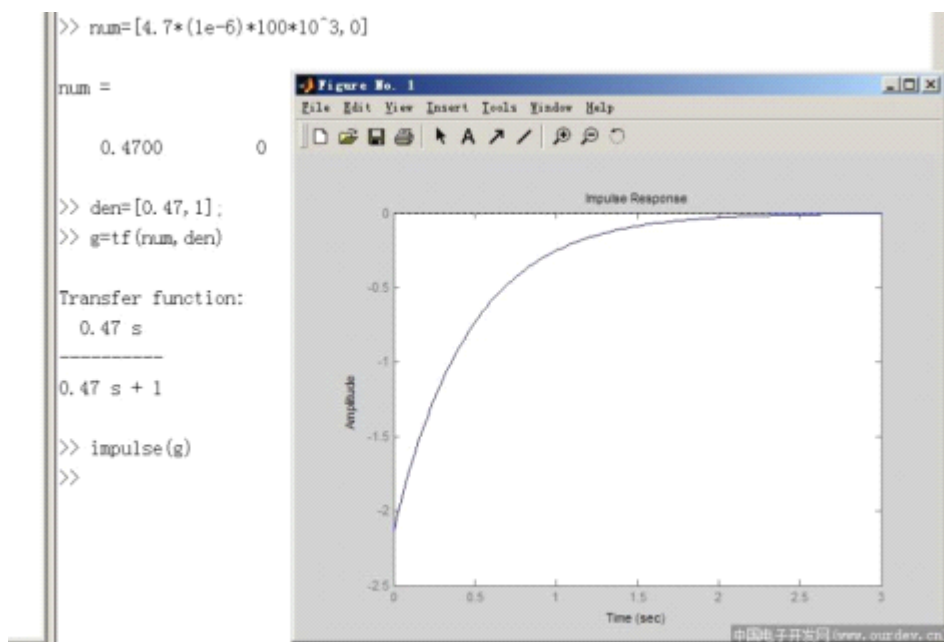


车速 10Hz 低通滤波后波形 (原文件名:车速 10Hz 低通滤波后波形. JPG)

视频在这里 [http://v.youku.com/v\\_show/id\\_XMjM10TQ3NzU2.html](http://v.youku.com/v_show/id_XMjM10TQ3NzU2.html)

现在还不是很稳,我想有两个原因,一个是参数没调到最佳,调了好久,先这样吧。再有就是电源太烂了,电机是额定 6V 的可实际电压空载的时候才打到 5 伏多一点,在平衡的时候没测,肯定更低了。

陀螺仪 ENC-03 是直接接 AD 输入端的,因为按照 datasheet 上边的参考电路有过冲问题,这个问题有个帖子已经讨论过,很多人都是围绕怎么补救这个问题,我来算一下为什么这样子,呵呵~如下:



高通滤波脉冲响应 (原文件名:QQ 截图未命名. jpg)

因为有问题,会给倾角数据造成影响,所以我就去掉了滤波,直接接到 AD。这样 1deg/s 有 0.67mv, 10 位 AD 参考电压是 3.36V,最小才能测到 3.28mv,小于 4.8deg 时就测不到了。本来担心这个问题,但试了下 KALMAN 滤波,真是强啊! 角度很精确,就这么用了。

车体研究了好久，没有用钢化玻璃的设备，就一直没动工。有天去打水突然看到旁边有个大的三合板，呵呵，于是乎。。

感觉车体结构也是个难点啊，要算尺寸，要打孔（先用烙铁头扎，在用螺丝刀来回钻），挺费劲的。还有支架，试用很短的铜柱，一个一个摞起来的，所有能找到的铜柱都用上了，刚好够用。

这种车体的中间不能负重，重了后面俩轮就往外翻，当时是做打水机器人的，后来没有做完就 over 了~~~

不知道 [STC](#) 那种的，我只用过 [STC89C51](#) 那种的，呵呵。这个车要用很多片内外设的。

AIN0:陀螺仪输入

AIN1:加速度计输入

T0: 采样周期定时，算法都在这里完成；

T1: 两路 PWM；

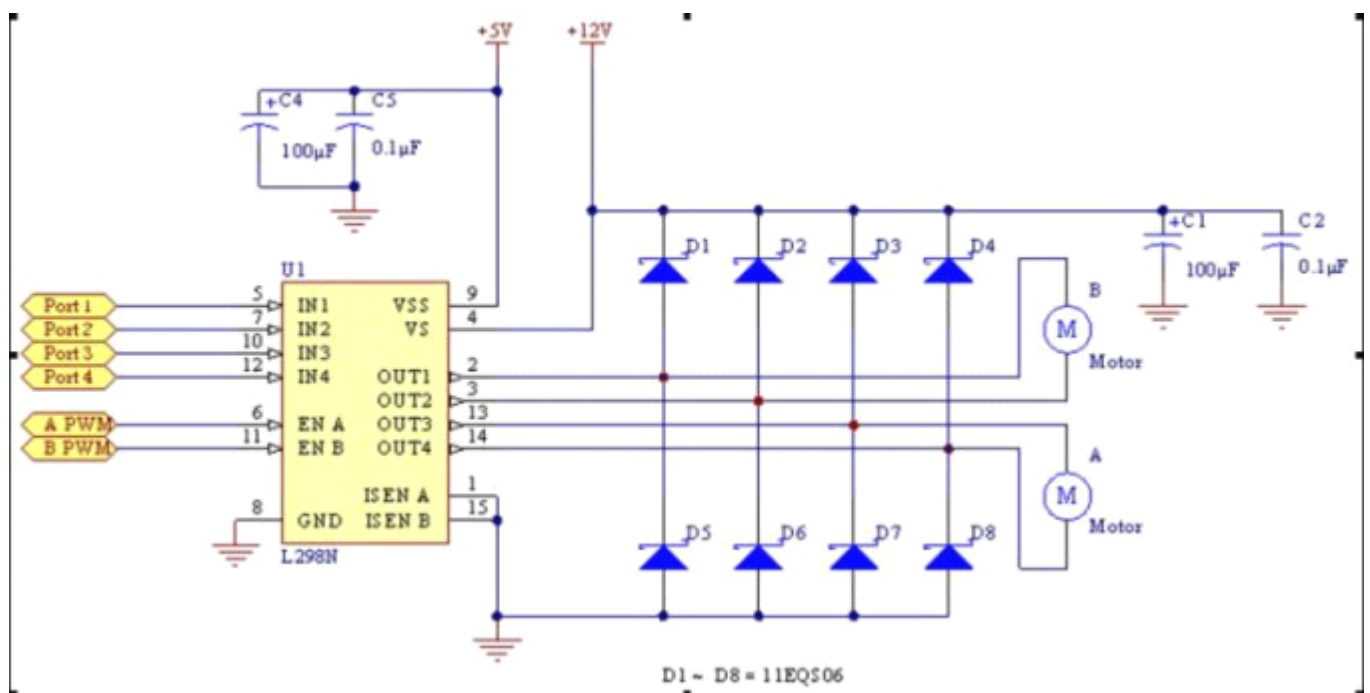
T2: 用作霍尔传感器 T 法测速（要用码盘 M 法测速，不需要用）、红外脉冲定时；

INT0、INT1: 接霍尔传感器，采集速度

INT2: 红外接收头输入信号（遥控要用无线的话，也省了）

电机驱动电路图

驱动我买的模块 20 块，卖家没提供电路图。应该是和这个一样的 [http://www.ourdev.cn/bbs/bbs\\_content.jsp?bbs\\_sn=1344216&bbs\\_page\\_no=1&bbs\\_id=1024](http://www.ourdev.cn/bbs/bbs_content.jsp?bbs_sn=1344216&bbs_page_no=1&bbs_id=1024)。都是没加光耦直接接过去的。



这个图完全能用，我已经焊接好这个电路，驱动小车。就是要大一点的散热片，普通散热片太小太烫了。

L298 偏偏是没有，PCB 上还要额外放 8 个高速二极管是很让人吐血的一件事情.....  
小功率版的 L293 是有二极管的，用起来方便多了。

问题找到了，本来程序是 Debug 版本的，要改成正式发行版的，VC 我只用了不久，是低级错误啦，呵呵。LZ 位软件我重新改过了。

VC 工程在这里。

上位机 VC++ 工程源码 [ourdev\\_610317VUO0LN.rar\(文件大小:4.99M\)](#) (原文件名:Curve1.rar)

写的有点乱，10.1 假期写的，后边就没怎么更新了。两种显示，字符显示和调试助手相同，要用示

波器的话，协议为：先发 0xFD，再连续发两个字节数据，多个数据依次发下去。单片机中此段程序即为发送波形显示数据的格式：

```
if(++ctr_pos==2)
{
ctr_pos=0;

temp1=speed_filter*10+150;
Putc(0xfd);
Putc(temp1/256);
Putc(temp1%256);
}
```

数据发的不能太快（差不多要大于 10ms），太快右侧字符频繁刷新界面，上位机会卡死，所以在界面上设置了禁止字符输出，用示波器时，最好选上。这个软件可以查看历史波形、数据。

自己顶一个，没人关注啦，大家对这个没什么兴趣？所有资料都有啊！~

我说下成本：

AVR 最小系统版 45；

陀螺仪 ENC-30MB: 35；

加速度传感器 MMA7260 模块: 48（5V/3.3V 供电）；

电机+轮子+联轴器两套: 96；

霍尔传感器模块（两对，每对 AB 相）: 34；

电机驱动板: 20；

遥控、显示看个人了。

45+35+48+96+34+20=278 元。

所有的加一起 300 元吧。

此贴的价值：

1.车的成本相对来说很低；

2.所有代码，电路开源（都是模块插线，没有 PCB 的。。）；

3.这个上位机真的很实用。没有示波器或者不方便用示波器时候，用它显示波形很爽的，并且可以保留历史波形哦！在做此车过程中，我先后用它看了角速度、加速度、陀螺仪积分的角度、KALMAN 滤波的角度、车运行过程中的角度、电机空载速度、PI 闭环的空载调速曲线（此车没用速度闭环）、车位移等曲线。很直观！发送数据的协议很简单，上面已经讲过。而且代码开源的，自己可以在此基础上增加、改善一些功能。

知道可不可以贴淘宝地址，有做广告的嫌疑啊。。。



LZ 买的太贵了，MMA7260 才 13，ENC-03 为 18，我也买了许多元件，但 AVR 单片机不太懂，一直没敢开工。正在学习，刚设计了个实验板。另外驱动电路是自己做的，分立元件的，体积稍大，但效果很好。向楼主学下中。。。。。

是有点贵当初买的时候也看了其他店子，有像你说的很便宜的，但是你也要考虑邮费起码 10 元吧，那个 MMA7260 挺好的别看要 48 元，是已经做好的模块，而且有插针，可以直接插在万用板上的，你要是 13 块钱买个芯片的话，要是没风枪的话，不好焊吧，我是不会焊，呵呵。其他店类似的模块也要 30 多吧，再加个邮费也差不多了。但客观的讲，那个店确实有点点贵，但是所有东西一次性都买齐了。

做此车不是非要用 AVR 的，只要速度、外设够用就可以！需要的外设上边都有说。关键要懂算法！其实我也是一知半解，给你推荐篇文章

卡尔曼滤波介绍中文版 [ourdev\\_611837OCLGOY.pdf\(文件大小:540K\)](#) ( 原 文 件 名:kalman\_intro\_chinese.pdf)

卡尔曼滤波介绍英文版 [ourdev\\_611838G0F5Z1.pdf\(文件大小:173K\)](#) (原文件名:kalman\_intro.pdf)  
文章从头开始看，看完离散卡尔曼滤波那一节，就会理解很多。后面扩展卡尔曼我看不懂，最后面举了个例子，好好看下。

卡尔曼 C 语言 [ourdev\\_611839GC1Z3U.txt\(文件大小:7K\)](#) (原文件名:kalman.txt) 再看下这个程序。和我用的结构差不多的（我的是抄 zlstone 的，他貌似也是复制的，呵呵，最难的就是卡尔曼）。下面是我的：

```
void Kalman_Filter(float angle_m,float gyro_m) //gyro_m:gyro_measure
{
angle+=(gyro_m-q_bias) * dt;//先验估计
```

```
Pdot[0]=Q_angle - P[0][1] - P[1][0];// Pk-' 先验估计误差协方差的微分
```

```
Pdot[1]=- P[1][1];
```

```
Pdot[2]=- P[1][1];
```

```
Pdot[3]=Q_gyro;
```

```
P[0][0] += Pdot[0] * dt;// Pk- 先验估计误差协方差微分的积分 = 先验估计误差协方差
```

```
P[0][1] += Pdot[1] * dt;
```

```
P[1][0] += Pdot[2] * dt;
```

```
P[1][1] += Pdot[3] * dt;
```

```
angle_err = angle_m - angle;//zk-先验估计
```

```
PCt_0 = C_0 * P[0][0];
```

```
PCt_1 = C_0 * P[1][0];
```

```
E = R_angle + C_0 * PCt_0;
```

```
K_0 = PCt_0 / E;//Kk
```

```
K_1 = PCt_1 / E;
```

```
t_0 = PCt_0;
```

```
t_1 = C_0 * P[0][1];
```

```
P[0][0] -= K_0 * t_0;//后验估计误差协方差
```

```
P[0][1] -= K_0 * t_1;
```

```
P[1][0] -= K_1 * t_0;
```

```
P[1][1] -= K_1 * t_1;
```

```
angle += K_0 * angle_err;//后验估计
```

```
q_bias += K_1 * angle_err;//后验估计
```

```
angle_dot = gyro_m-q_bias;//输出值（后验估计）的微分 = 角速度
```

```
}
```

我只能帮这些了，我的方向是运动控制方面，**KALMAN** 貌似在信号分析相关专业会学的，所以我也没细看了。上述程序我是直接复制过来的，只改了采样周期，剩下一点没动（还加了点注释，呵呵）。有关 **KALMAN** 的详细解释一般都是外国网站的，中国的都是扯淡。我做这个车的目的就是想搞下 **PID**，怎么获取角度是次要的东西。

需要两个传感器，角速度传感器（**ENC-03MB**）、加速度传感器（**MMA7260**）。**AD** 测量并转换得两者数据，送 **KALMAN** 滤波，输出得到角度（**angle**）、角速度（**angle\_dot**），就可以用了！

两个传感器得到的数据是什么呢??

是角度還是角速度呢??

我用了角速度传感器和加速度传感器得到傾角是沒問題的, 但對車子計算偏航的角度就出現問題了因為加速度传感器沒法對角速度传感器進行誤差補償, 這方面 **lz** 是怎麼解決呢??



就是角速度和加速度啊！要得到角度需要卡尔曼（KALMAN）融合，可以活得很精确的角度。如果用我那段程序需要根据自己的采样周期修改 dt 的值，我的是 6ms，所以 dt=0.006。

用卡尔曼融合角速度传感器和加速度传感器的数据，这里明白了，也可以得到倾角，因为有对地面的加速度是不变的，也是 1g

到了计算偏航的时候，没有了地心引力，就头痛了

偏航的也是角速度和加速度吗？角速度传感器的数据得到 gyro\_m，但怎么把加速度得到的数据算成 angle\_m 呢？

我还是不懂呢!!

不好意思，我搞错了。送给 KALMAN 的数据应该是角速度、加速度计求得的角度而不是加速度。

```
acc=acc-1650;//加速度测量数据-静态值
```

```
acc=acc/800;//这里得到角度的正弦值
```

```
if(acc>1)//算出的值可能大于 1 或小于-1，是错的，所以限下幅
```

```
acc=1;
```

```
else if(acc<-1)
```

```
acc=-1;
```

```
acc=180/3.1415*asin(acc);//asin 求反正弦，得出的单位是弧度，*180/pi 得到角度值
```

```
w=(gyro-1178)/0.67;//角速度值
```

```
Kalman_Filter(acc,w);//角度，角速度。函数原型：Kalman_Filter(float angle_m,float gyro_m)
```

加速度计反正弦求出的角度静止时是很精确的，但是在运动（偏航就是运动吧）时由于有除重力附加的加速度，所以算出的角度将不准。角速度积分求出的角度，因为是积分，所以在累加过程中又慢慢的累计误差，慢慢得出的角度也不准了，角速度传感器本身也有温漂问题，也会造成影响。融合之后很精确。

acc 是加速度计是输出，这是一轴加速度计吗？就是说加速度计的方向轴是与车子的方向垂直，这理解对吗？

```
acc=acc-1650;//加速度测量数据-静态值 <---这是减去 0g 输出的值
```

```
acc=acc/800;//这里得到角度的正弦值 <---800 这值... 不理解...>,<
```

之后的都理解了，asin 后换成角度，w 是处理角速度传感器的输出

还有的是，Iz 主控的 AD 是多少位的？我的是 0-5V 换成 0-1024 的，所以静态值大约 3XX.

其实这也不是问题，主要的是 800 是怎么计算的呢？

刚想到了一个问题，车子向前左或前右转的时间，加速度计不是应该有俩个输出吗？怎么只取 acc 一个呢？

想了想，如果 acc 是与地平面承水平的，在 asin 的时候，不会把别一个值成了 1g(地心引力)吧？那不是也在做倾角吗？

800mv 见数据手册：点击此处下载 [ourdev\\_612056WR1X9S.pdf](#) (文件大小:197K) (原文件名:MMA7260QT\_NoRestriction.pdf)

## g-Select

The g-Select feature allows for the selection among 4 sensitivities present in the device. Depending on the logic input placed on pins 1 and 2, the device internal gain will be changed allowing it to function with a 1.5g, 2g, 4g, or 6g sensitivity (Table 3). This feature is ideal when a product has applications requiring different sensitivities for optimum performance. The sensitivity can be changed at anytime during the operation of the product. The g-Select1 and g-Select2 pins can be left unconnected for applications requiring only a 1.5g sensitivity as the device has an internal pull-down to keep it at that sensitivity (800mV/g).

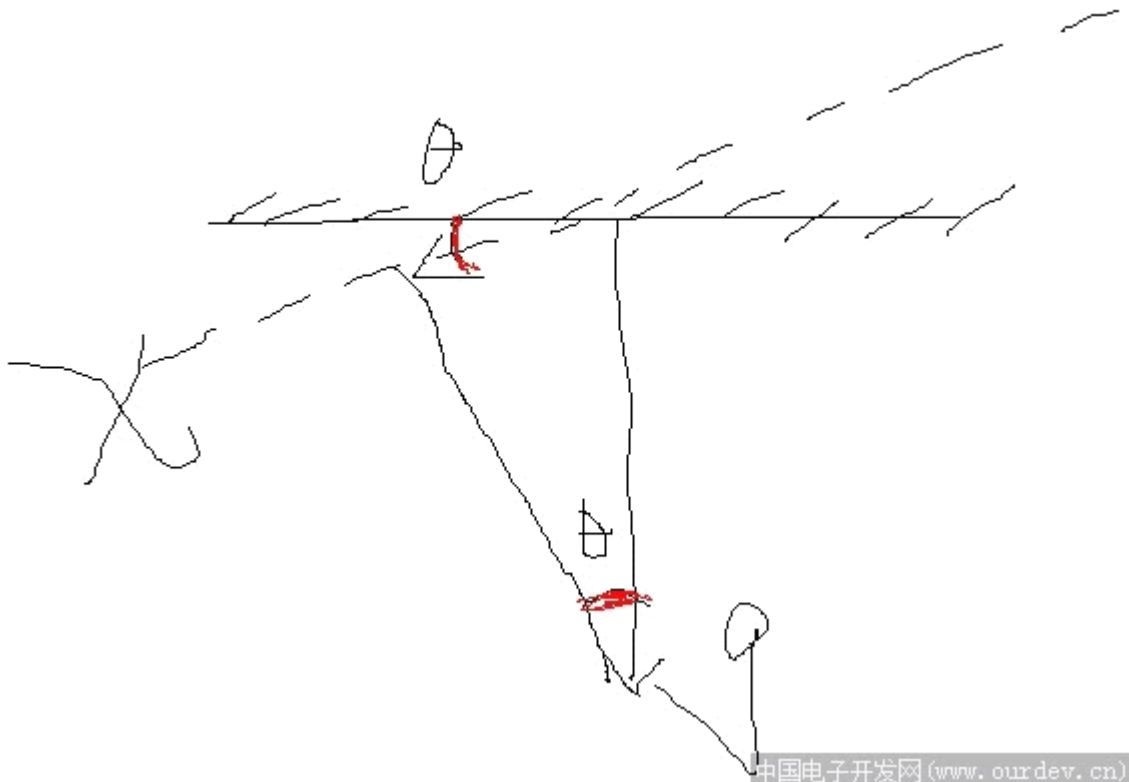
**Table 3. g-Select Pin Descriptions**

g-Select2	g-Select1	g-Range	Sensitivity
0	0	1.5g	800 mV/g
0	1	2g	600 mV/g
1	0	4g	300 mV/g
1	1	6g	200 mV/g

中国电子开发网 (www.ourdev.cn)

(原文件名:QQ 截图未命名 3. jpg)

引用图片



中国电子开发网 (www.ourdev.cn)

(原文件名:QQ 截图未命名 2. jpg)

引用图片

除 800mv 得到的是多少 g，假如 0.2g，那么  $0.2g/g=0.2$  不就是正弦值么？然后再反正弦并把单位转成角度制。

我的 AD 用 10 位，参考电压用 MMA7260 模块自带的 3.3V 稳压电源（3.36v），参考电压越小越精确，但也别太小了。看下最大能测的加速度： $(3360-1650)/800=2.1375$ ，能测到 2g 多点，够用了。你用 5V 参考的话分辨率降低很多，也没有什么好处。MMA7260 都是 3.3V 供电的吧，直接把电源线接到 AD 参考电压就行了。还有要考虑陀螺仪的量程，静态输出 1.35V，由公式：

Output voltage =  $V_0 + S_v \times w$  [V]

$V_0$  :Static output [V] (at angular velocity is 0 [deg/s])

$S_v$  :Scale factor [mV/deg/s]

w : Angular velocity [angular velocity range : -90 to +90 [deg/s]]

最大能测的 w 为 -90 to +90 极限电压为  $1350 + 0.67*90=1410.3$ ,  $1350 - 0.67*90=1289.7$ ，落在参考电压 3.36V 内，没有问题。

另外 AD 最小能分辨  $3360/1024=3.28125\text{mv}$  电压，对于小于此值时的角速度是无法测量的（其实挺小的），用 KALMAN 滤波之后的角度很精确的。

在昨晚想到的, 車子向前左或前右轉的時間, 加速度計不是應該有兩個輸出嗎? 怎么只取 **acc** 一個呢?

想了想, 如果 **acc** 是與地平面承水平的話, 在正弦的時候, 那不是也在做傾角嗎??

這已猜到了是正弦了, 但是...

除 800mv 之前, 不是要把 AD 得到的数据換成電壓值嗎? 例如我的主控 AD 是 0-5V 對應 0-1023 的, 先把 AD 值\*5/1024 再除 800m 才對嗎?

傾角的計算(也就係除 800mv 後得到的是多少 g, 再除 1g 的正弦)已經完成了

回到一直的問題, 是水平面的偏角, 如果 z 軸向天, 水平面就是 xy 面的角度變化, 沒有 1g 的參與, 這所以才令我頭痛了

看懂了, 原來已經換成了電壓,  $(3360\text{mv}-1650\text{mv})/800\text{mv}=2.1375$

但是 xy 面(z 軸向天)的偏角還困擾著....

lz, 傾角的計算已經完成了

水平面的偏角有辦法解決嗎?? 感謝...

不懂你的意思。。。

得到角度就行了呗，还算什么“水平面的偏角”啊？

OUT1、OUT2 间电压就是电机电源经过 PWM 斩波后的电压，你把 ENA 也进程高电平，就是占空比 100%，这是 OUT1、OUT2 电压为电机电源电压回复【74 楼】ZRYXY

7260 是加速度传感器，在静止时测的是重力在相应坐标轴上的加速度分量，但是在频繁变速运动时在对坐标轴上会有运动中的额外加速度，这就给获得真实角度带来干扰了，可能可以通过补偿获得角度吧，但是应该很困难了，呵呵，但是只用陀螺仪貌似可以的，乐高的就是只用的陀螺仪，也需要补偿了。

我用到了 PID,  $PWM=ut=Kp1*角度+Kd1*角速度+Kp2*车速+Ki2*车位移$ 。但是和传统的 PID 有点不太一样, 一般 PID 就是单输入单输出, 但这里给定为角度 0 和车位置 0, 输出为角度和车位置。其实用 PID 是最简洁、最粗糙的方法, 效果好的话都用到了现代控制, 我不怎么懂, 呵呵。

你这个程序有些难啊, 能不能简化下, 程序主要的只留 AD 采集、卡尔曼滤波和 PID 计算 PWM 输出就能跑的, 不要外围的车速检测、红外遥控、液晶显示、和键盘什么的, 只留加速度传感器和角速度传感器以及 PWM 输出还有电机正反转换, 底子薄的人看的好辛苦啊, 看了五天了, 眼花了, 呵呵, 麻烦你啦, LZ!

我的也模仿成功了, 效果还可以。下一步将小车美化一下, 刚购买了十节镍氢电池, 回头加在小车上, 它就可以自由跑了。我调试时, 将加速度获得的角度乘以 3.5 后才送入卡尔曼滤波的, 不过还没理解卡尔曼滤波, 在调试时发现小车对 MMA7260 反应偏小偏慢, 所以就将其放大后才送入卡尔曼滤波的。

回复【102 楼】ZRYXY

这个问题是因为你没有定义 uchar 和 uint 两个宏, 这个怨我! 我图省事把这两个定义直接写在 io.h 中了, 这样以后就直接用这两个宏就可以了, 但这也让代码不可移植了。不要学我这么干, 我以后也不这么干了。用的时候这样定义吧:

```
#ifndef uchar
#define uchar unsigned char
#endif
```

```
#ifndef uint
#define uint unsigned int
#endif
```

此段程序分别加在 Balance\_1602.c 和 Balance\_2motors.c 程序前边。

这样就 OK 了。通过这个错误, 也知道了大家都没有运行过程序啊。。如果我有时间会把程序整理下、加些注释。

回家过了个年, 回来之后有点忘记了。。

首先要确保采集到的角度足够稳定, 要没有毛刺。有微小偏差没关系, 这是因为车平衡时的角度为 0 度, 如果有点偏差, 车顶多就是平衡时倾斜一点, 再靠水平的速度、位移调节也不会让车动弹的! 检查测量到角度的效果可以用我上边 25 楼提到的那段程序稍加修改。

再就是一定要确保你的 PID 各项构成负反馈, 如果其中任意一项构成正反馈都永远不会稳, 或者只能稳定一小段时间。参数共 4 项, 需要依次测试。

调节过程中, 要时刻记住 PWM 量程是多少, 假设为 0~255, 那么在调节过程中就要根据这个参数来设定各个 K 值大概的值。你想想太小的话, 假设角度  $K=2$ , 那么当角度很大的时候, 假设 30 度, 这时控制量才为  $2*30=60 \ll 255$ , 效果就是车都快到了, 你给电机的能量才一点点, 你说车还能站起来么? 另外  $K=10000$  的话, 当车体倾斜 1 度的时候, 控制量为  $10000*1=10000 \gg 255$ , 效果就是只要角度不为精确地 0 度, 那么电机就会全速运行, 以最大力量调整车体, 这时你想想能稳定么, 车体倾斜角度会很快发散, 最后倒下。

整定方法大概如下:

1. 调角度 K 值, 调到基本能够站立;
2. 调角速度 K 值, 从零慢慢增加, 这个 K 很小。太小的话, 车子反应迟钝, 好像没有这项一样, 太

大的话，车子抗扰能力变差，所以要适当。

3.调水平速度  $K$  值的时候，要注意你获得的车速是不是很精确的，如果用很多线的光电码盘就会很精确，但如果码盘只要几线，或者像我用的一个霍尔传感器，获得的速度就会很不精确，这时就要做好车速的低通滤波，很关键！滤波做好了，车速调节效果很好的，算位移的时候都是很精确的，没有太快的漂移。车速  $K$  值也是慢慢增加，增加过程中观察车的反应。

4.位移  $K$  值也应该很小，因为是速度的累计，慢慢会漂移变大，设定一个上下限，是为了防止漂移过大造成控制量过大最后会等于 255。

出现点效果后的调节也不是按照 1234 这样调的，后来我是 1、3、4 反复调节的，2 基本没动。

很多细节问题有点忘了，当时刚做完应该写下来的啊。。

修改问题后的程序，点击此处下载 [ourdev 618995UWBCLG.rar\(文件大小:105K\)](#) (原文件名:程序.rar)

PS: 我上传的这个程序不是我最终烧录的程序，我当初上传之前做了点格式的改动，现在好了。

如果你的程序直接用我的话，那启动系统是需要先按下 PINB0 按钮的。main 函数 512 行 while(1){...}。

我现在没有时间简化程序，只加了注释，等我有空改下。