**硬件使用**：

1.大鱼电子的平衡小车系统板

2.大鱼电子平衡小车电机和底盘

3.航模3s电池

**硬件占用的GPIO**的**框架**：

编码器1——PA0/PA1---TIM2

编码器2——PB6/PB7---TIM4

电机1——PB12/PB13

电机2——PB14/PB15

PWM1——PA8

PWM2——PA11

MPU6050中断引脚——PB5

MPU6050所用IIC——PB3/PB4

14151312

**Encoder.c**

GPIO初始化

定时器初始化

设置编码器模式

清除标志位

中断配置

清零计数

开定时器

**PWM.c**

GPIO初始化

定时器初始化

输出比较模式初始化

MOE主输出使能

通道预装载使能

使能定时器在ARR寄存器上的预装载寄存器

**Moto.c**

GPIO初始化

**Mpu6050中断引脚**

**Exit.c**

GPIO初始化

EXTI外部中断初始化

NVIC

PWM赋值函数

PWM限幅

绝对值函数

小车异常关闭函数

PID函数（位置式PID）

直立环（V）

速度环（V）

转向环（X）

平衡小车的控制理论和控制过程的编程。

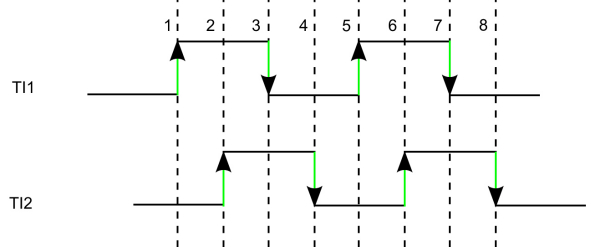
PID调参

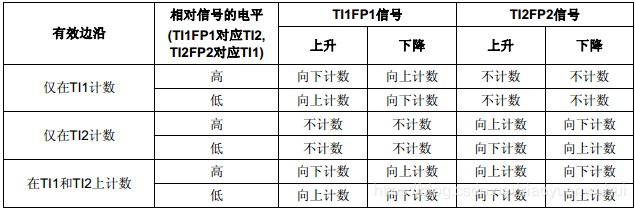
TIM\_EncoderInterfaceConfig

(TIM2,TIM\_EncoderMode\_TI12,TIM\_ICPolarity\_Rising,TIM\_ICPolarity\_Rising);

**TIM2：定时器2**

**TIM\_EncoderMode\_TI12：T1和T2的每个跳变沿均计数。**





**TIM\_ICPolarity\_Rising：不反相。**

**TIM\_ICPolarity\_Rising：不反相。**

根据两个输入信号（TI1&TI2）的跳变顺序，产生了计数脉冲和方向信号。

依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对TIMx\_CR1寄存器的DIR位进行相应的设置。

不管计数器是依靠TI1计数、依靠TI2计数或者同时依靠TI1和TI2计数。

在任一输入端(TI1或者TI2)的跳变都会重新计算DIR位。

【正反转】

正转：T1超前T2相位90度。

反转：T1滞后T2相位90度。

【模式】

TI1模式：在T1的所有边沿 计数。

TI2模式：在T2的所有边沿 计数。

TI12模式：在T1和T2的所有边沿 计数。

**PID控制**

PID控制，就是对偏差进行比例、积分和微分的控制。

PID由3个单元组成，分别是比例（P）单元、积分（I）单元、微分（D）单位。

工程中P必然存在，在P的基础上又有如PD控制器、PI控制器、PID控制器等。

比例项：提高响应速度，减小静差。

积分项：消除稳态误差。只要有偏差，我就积分，有一丁点偏差，我也会积分。积积，就会非常大。直到你偏差变为0.

微分项：减小震荡以及超调。

*Pwm=Kp\*e(k)+Ki\*∑e(k)+Kd\*[e(k)-e(k-1)]*

*Kp\*e(k)*

*Ki\*∑e(k)*

*Kd\*[e(k)-e(k-1)]*

*Ki=Kp\*(1/Ti) \*Ti*

*Kd=Kp\*(1/T)\*Td*

**位置PID**

***1.理论分析***

位置闭环控制就是根据编码器的脉冲累加测量电机的位置信息，并与目标值进行比较，得到控制偏差，

然后通过对偏差的比例、积分、微分进行控制，使偏差趋向于零的过程。

***2.公式***

*Pwm=Kp\*e(k)+Ki\*∑e(k)+Kd\*[e(k)-e(k-1)]*

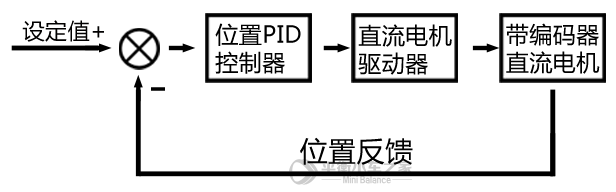
**e(k)**：本次偏差

**e(k-1)**：上一次的偏差

**∑e(k)**：e(k)以及之前的偏差的累积和;其中k为1,2,,k;

**Pwm**代表输出

***3.结构框图***



***4.C语言实现***

int Position\_PID (int Encoder , int Target)

{

static float Bias, Pwm, Integral\_bias, Last\_Bias;

Bias=Encoder-Target; //计算偏差

Integral\_bias+=Bias; //求出偏差的积分

Pwm=Position\_KP\*Bias+Position\_KI\*Integral\_bias+Position\_KD\*(Bias-Last\_Bias); //位置式PID控制器

Last\_Bias=Bias; //保存上一次偏差

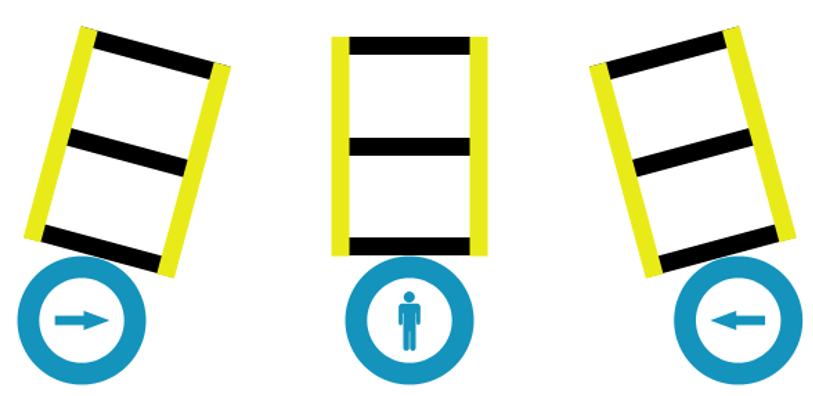
return Pwm; //输出

}

**直立环**

***1.理论***

小车往那边倒，车轮就往哪边开，既可以保持车子的平衡。



***2.公式***

a=b1\*θ+b2\*θ**＇**; ——> 比例微分控制【PDout=Kp\*Angle+Kd\*( Angle-Angle\_last)】

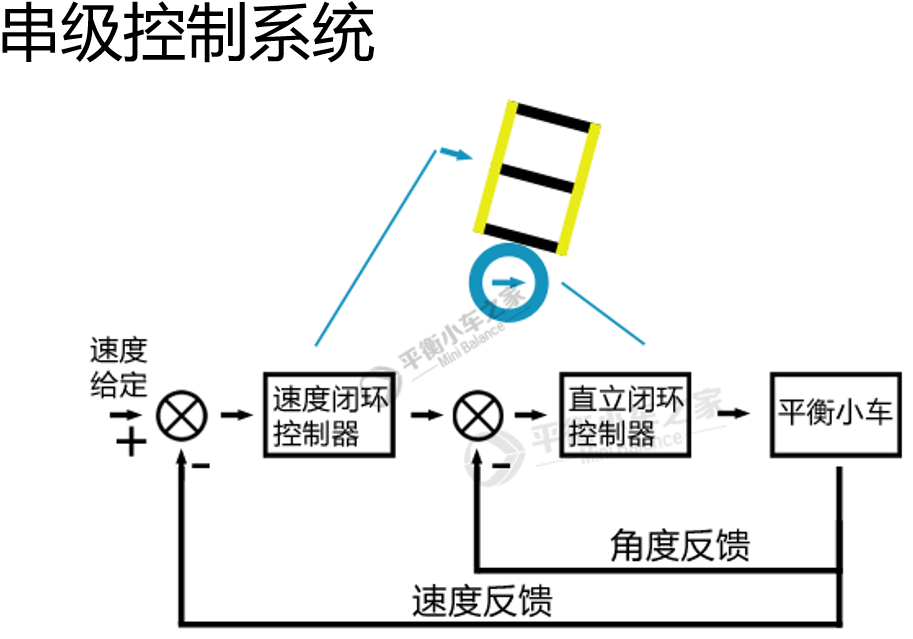
***3.结构框图***



直立环：让小车角度趋近0；

速度环：让电机速度趋近0；

**速度环、串级PID**



速度环输入：1.给定速度。2.速度反馈。

输出：角度值（直立环的期望速度输入）

直立环输入：1.给定角度（速度环输出）。2.角度反馈

输出：PWM（直接控制小车）

Vertical\_out=Kp1\*( real\_Angle- expect\_Angle)+Kd\*D( real\_Angle- expect\_Angle) //直立PD控制器

Velocity\_out =Kp2\*(Encoder\_ real- Encoder\_ expect)+Ki\*S(Encoder\_ real- Encoder\_ expect) //速度PI控制器

（NOTE：（1）Velocity\_out = expect\_Angle.（2）Kp1：Vertical\_Kp.（3）Kp2：Velocity\_Kp.）

【中文】

直立环输出=Kp1\*（真实角度-期望角度+机械中值）+Kd\*角度偏差的微分 //角度偏差=真实角度-期望角度

速度环输出=Kp2\*（反馈编码器值-期望编码器值）+Ki\*编码器偏差的积分 //编码器偏差=反馈编码器值-期望编码器值

（NOTE：（1）速度环输出=直立环的期望角度。（2）Kp1：直立环Kp。（3）Kp2：速度环Kp。）

合并推导：

直立环输出Vertical\_out=Kp1\*(θ\_r-θ\_e)+Kd\* (θ\_r-θ\_e)**＇**

速度环输出Velocity\_out=Kp2\*(E\_r- E\_e)+Ki\*Σ(E\_r- E\_e)

因为：θ\_e = Velocity\_out

所以直立环输出Vertical\_out =

= Kp1\*{ θ\_r-[ Kp2\*(E\_r- E\_e)+Ki\*Σ(E\_r- E\_e) ]}+Kd\*(θ\_r-θ\_e)**＇**

= Kp1\* θ\_r- Kp1\* Kp2\*(E\_r- E\_e)- Kp1\*Ki \*Σ(E\_r- E\_e) +Kd\*(θ\_r-θ\_e)**＇**

= Kp1\* θ\_r+ Kd\*(θ\_r-θ\_e)＇**-** Kp1\* [Kp2\*(E\_r- E\_e)+ Ki \*Σ(E\_r- E\_e)]

【中文】

=Kp1\*真实角度+ Kd\*角度偏差的微分**-**Kp1\* [Kp2\*编码器偏差- Ki \*编码器偏差的积分]

即：

串级输出OUT=

Kp1\* real\_Angle +Kd\* D( real\_Angle- expect\_Angle)

**—**

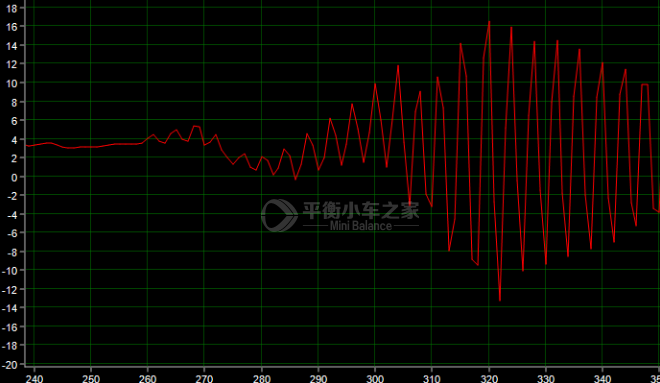
Kp1\*[Kp2\*(Encoder\_ real- Encoder\_ expect)+Ki\* S(Encoder\_ real- Encoder\_ expect)]

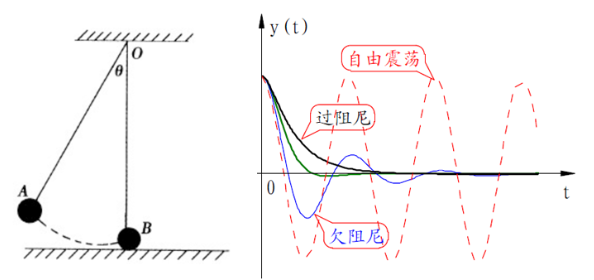
**纯比例控制**

单摆的加速度和单摆的倾角成正比

只存在大小与角度偏差成正比的回复力

a=b1\*θ; ——> 比例控制【Pout=Kp\*Angle】



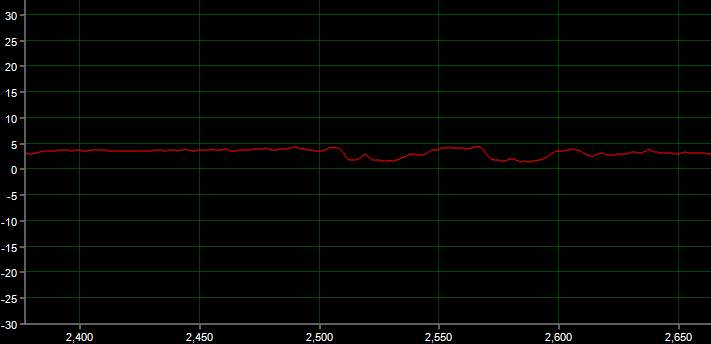


**比例微分控制器**

大小与角度成正比、方向与角速度成正比的回复力

大小与角速度成正比、方向与回复力成反比的阻尼力

a=b1\*θ+b2\*θ**＇**; ——>比例微分控制【PDout=Kp\*Angle+Kd\*( Angle-Angle\_last)】



1. 编写PID相关函数以及平衡小车控制函数。
2. 编写缺少的一个函数。
3. 完善主函数----初始化&变量的定义。

控制函数写在外部中断服务函数里面。

10ms控制周期由MPU6050的INT引脚给定。

保证数据的采集和处理的同步性。

**3.1：**

《修正control.c的遗漏处》、

《速度环的反馈系统》、

《平衡小车硬件结构与组装》

**3.2：**

《PID调参》

**3.2：调参步骤：**

1. 确立机械中值。
2. 直立环（内环）—Kp极性、Kp大小。Kd极性、Kd大小。
3. 速度环（外环）——Kp&Ki极性、Kp&Ki大小。
4. 转向环——系数极性、系数大小。

**机械中值：**把平衡小车放在地面上，从前向后以及从后向前绕电机轴旋转平衡小车，两次的向另一边倒下的角度的中值，就是机械中值。

**直立环**

**Kp极性：**

极性**错误**：小车往哪边倒，车轮就往反方向开，会使得小车加速倒下。

极性**正确**：小车往哪边倒，车轮就往哪边开，以保证小车有直立的趋势。

**Kp大小：**

Kp一直增加，直到出现大幅低频震荡。

**Kd极性：**

极性**错误**：拿起小车绕电机轴旋转，车轮反向转动，无跟随。

极性**正确**：拿起小车绕电机轴旋转，车轮同向转动，有跟随。

**Kd大小：**

Kd一直增加，直到出现高频震荡。

直立环调试完毕后，对所有确立的参数乘以0.6作为最终参数。

原因：之前得到的参数都是Kp、Kd最大值，根据工程经验平衡小车的理想参数为最大参数乘以0.6求得。

结果：乘以0.6后，小车的抖动消失，但同时直立效果也变差。待下面加入速度环就能得到更好的性能。

**速度环**

**速度环参数调节前注意：**

**一、**

在调试【速度环参数极性】时：需要去掉（注释掉）【直立环运算】

在调试【速度环参数大小】时：再次引入（取消注释）【直立环运算】

**二、**

【转向环运算】始终是去掉（注释）的一个状态。若转向环已提前将参数调试好，则未注释也影响不大。

**Kp&Ki：**

线性关系、Ki=(1/200)\*Kp、仅调Kp即可。

**Kp&Ki极性：**

极性**错误**：手动转动其中一个车轮，另一车轮会以同样速度反向旋转——典型负反馈。

极性**正确**：手动转动其中一个车轮，两个车伦会同向加速，直至电机最大速度——典型正反馈。

**Kp&Ki大小：**

增加Kp&Ki，直至：小车保持平衡的同时，速度接近于零。且回位效果较好。

**转向环**

**Kp极性：**

极性**错误**：拿起小车，并将小车绕Z轴旋转，两车轮旋转的趋势与小车旋转趋势一致——典型正反馈。

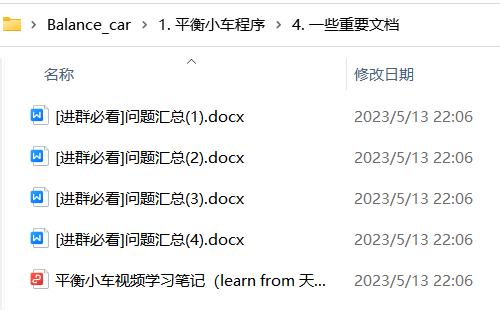
极性**正确**：拿起小车，并将小车绕Z轴旋转，两车轮旋转的趋势与小车旋转趋势相反——典型负反馈。

**Kp大小：**

加大Kp，直至走直线效果较好，且无剧烈抖动。

大家一定要去看下面这几个文档，里面包含了群同学的一些共性问题以及解答。同时也会持续更新。

文件在**\Balance\_car\1. 平衡小车程序\4. 一些重要文档**里面：



**QQ：**398828778

**QQ群：** （学术交流）

**1群：**822263013

**2群：**901509607

**3群：**424492788

**B站：**天下行走ㅍ\_ㅍ （相关视频）

**知乎：**天下行走 （相关文章）

**闲鱼：**抉择之刃银 （卖些好东西）