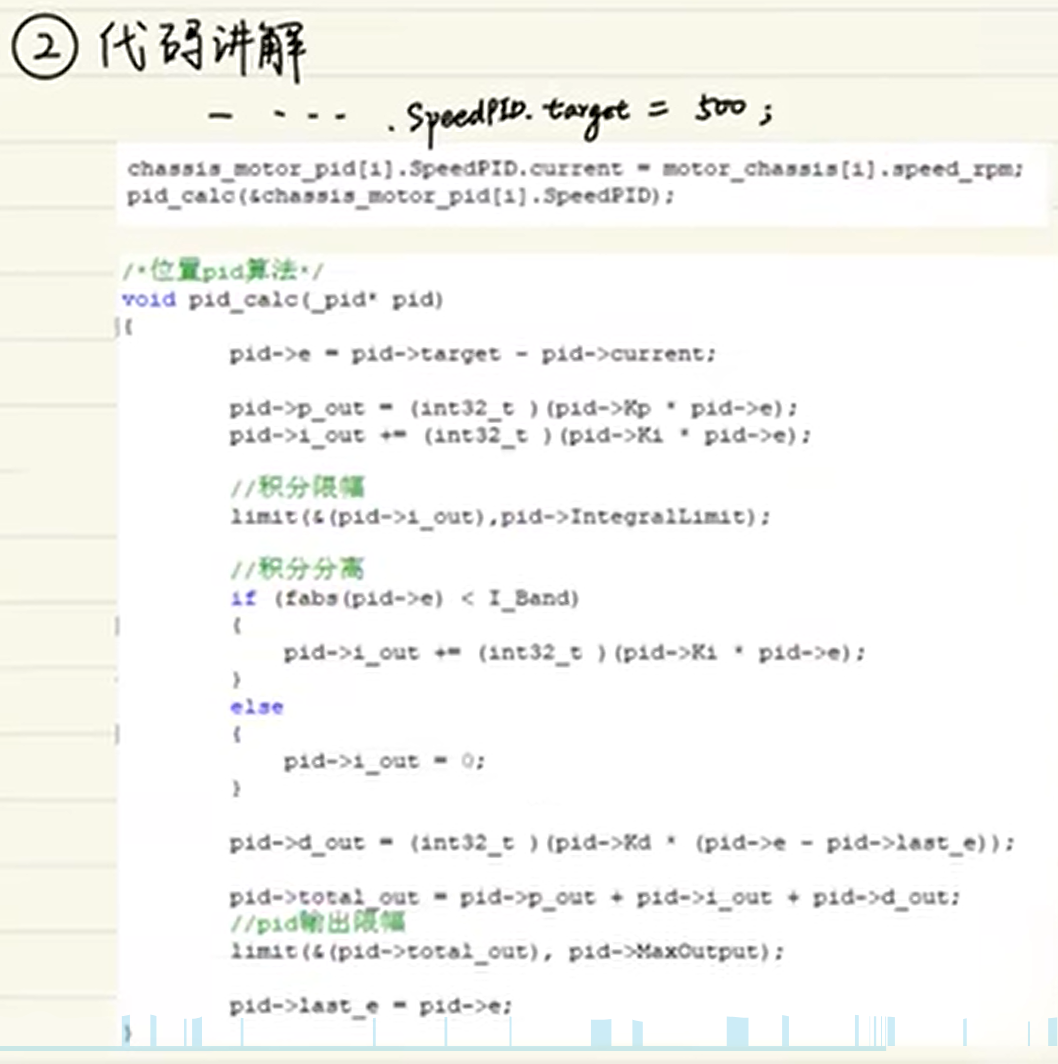
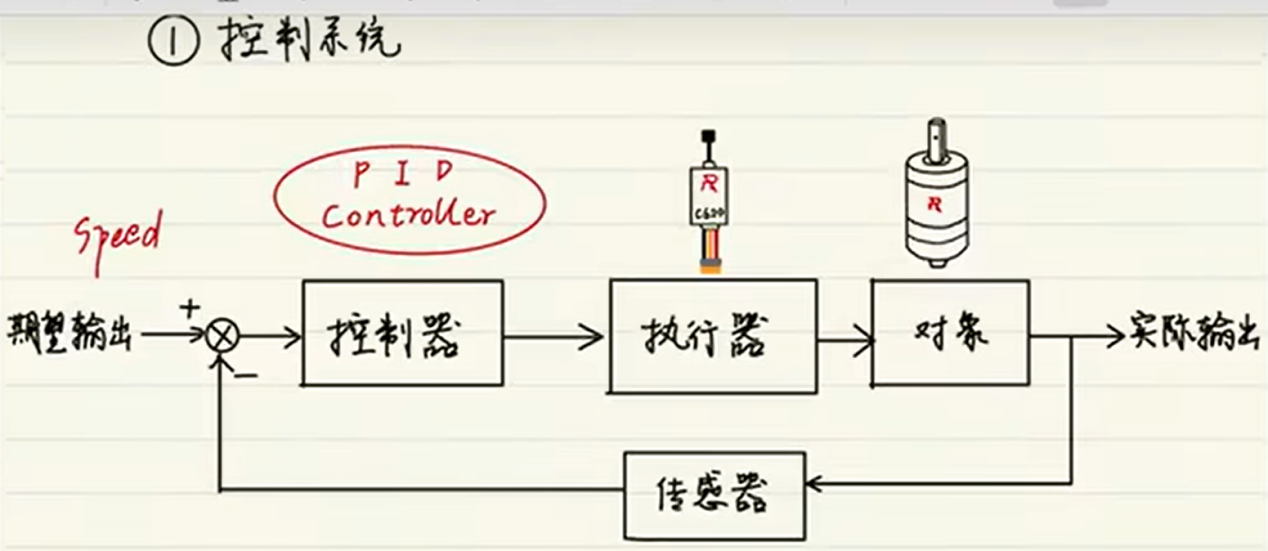
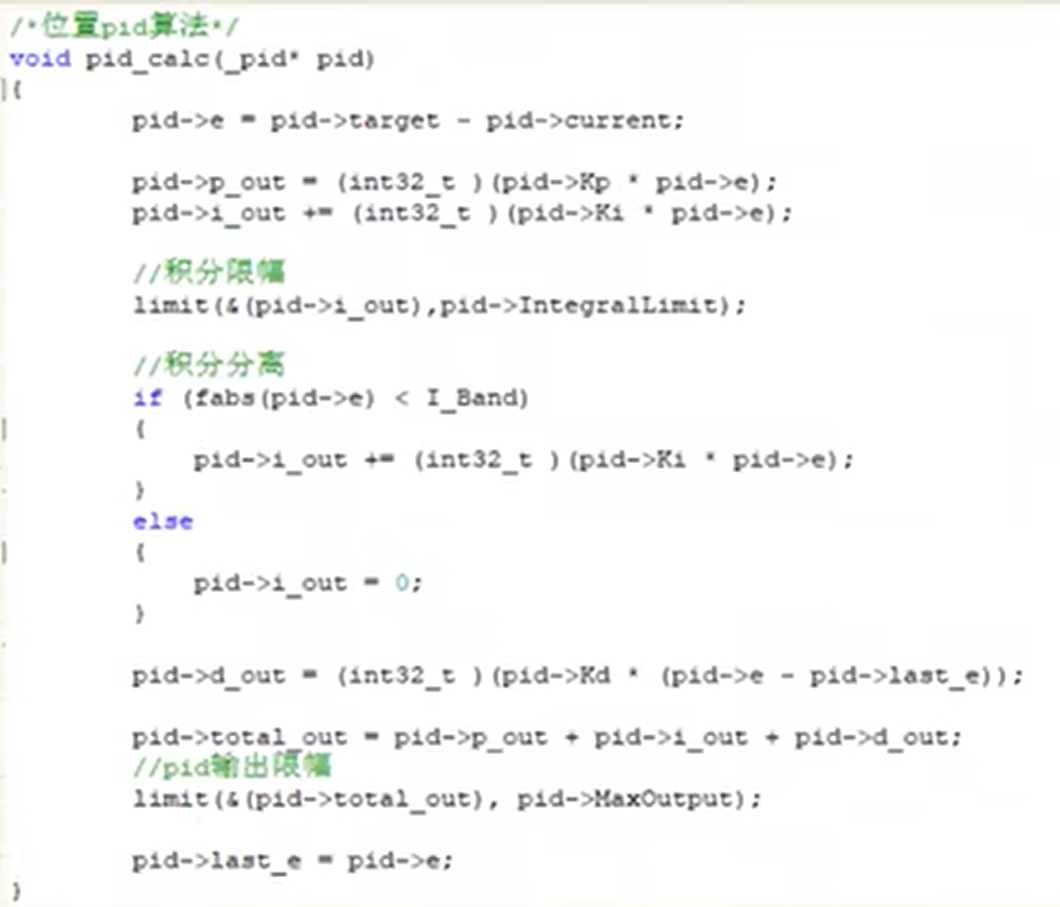
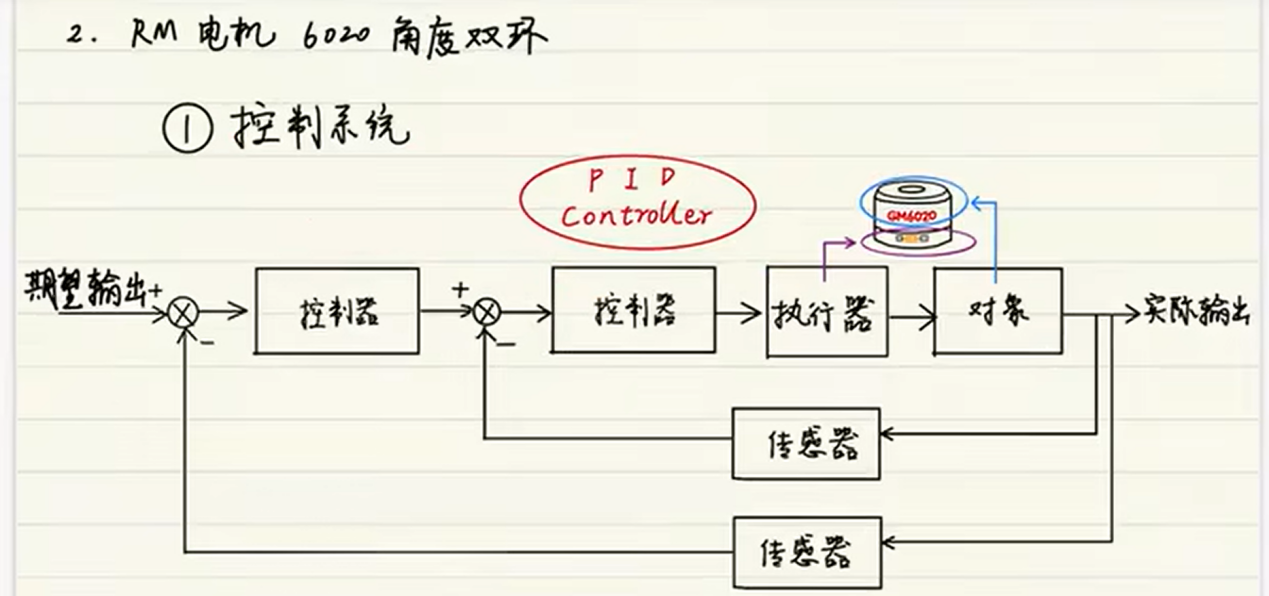
<https://www.bilibili.com/video/BV1B54y1V7hp/?spm_id_from=333.999.0.0>



双环PID

//串级PID

//定义所需要的变量

float Kp\_ex; //外环P

float Kp, Ki, Kd; //内环PID

float Motor1\_speed; //电机当前的速度

float Position\_now; //小车当前的位置，由当前速度积分得出

float target\_speed; //我们需要电机达到的速度,外环的输出，也就是内环的输入

float target\_position; //我们需要小车达到的位置，由摇杆积分得出

float err\_position\_now; //当前小车位置的误差

float err\_position\_last; //上一次小车位置的误差

float err\_speed\_now; //当前速度与电机期望值的差，也就是当前的误差值

float err\_speed\_last; //上一次计算的误差值

float err\_speed\_i; //积分项，将累计时间内所有的误差值

float output; //经过PID算法后输出的数据，其实是一个控制PWM占空比的参数，我们只需要直接放在输出PWM的函数里就好了

//外环PID

err\_position\_now = target\_position - position\_now;

target\_speed = Kp\_ex \*err\_position\_now;//外环的输出就是内环的输入

//速度的限幅处理略

err\_position\_last = err\_position\_now;

//内环PID

err\_speed\_now = target\_speed - Motor1\_speed;

errspeed\_\_i += err\_speed\_now;

if(err\_speed\_i > x) err\_speed\_i = x;//这里需要设定一个极限值，以防积分变量溢出，其实一般来说并不会达到这个极限值，因为误差并不都是正数

//没有对D进行操作

output = Kp \* err\_speed\_now + Ki \* err\_speed\_i +Kd \* (err\_speed\_now - err\_speed\_last);//PID公式

if(output > x) output = x;//这里需要对输出也进行一个限制，这里倒不是限制溢出，而是PWM输出的参数本身就有最大值，超过了就无效了。

TIM\_SetCompare1(TIMx, output);//对产生PWM的定时器通道一进行输出

err\_speed\_last = err\_speed\_now;//将使用完的当前误差值赋给上一次的误差值，进行下一轮循环