Mini51/2/4 BLDC 控制代码简介

www.nuvoton.com

nuvoTon

议题

- BLDC应用的三大领域：航模，吸尘器，电动工具
- 代码总体框架，三大部分：ADC中断，Timer1中断，主循环
- ADC中断：只负责测控电流
- Timer1中断：只负责检过0换相
- 主循环介绍：负责调整占空比、导通角等
- 启转介绍
- 全局变量介绍
- 启转波形

航模应用特点

- 转速高，启转快
  - 转速100~200Krpm或更高
- 中高端产品测电流
  - 保护功率管不过流
- 低端产品不测电流
  - 硬件上加大功率管电流余量
  - 启转阶段限压限加速，以防过流

吸尘器应用特点

- 中高转速
  - 转速60~150Krpm
  - 加速度无需太快
- 需做电流控制满足以下功能
  - 限最大功率
  - 自动功率或转速调整

电动工具应用特点

nuvoTon 新唐科技

- 转速不高，转矩提升要快
  - 转速20~40Krpm
  - 负载不稳定,要控制转矩
- PWM触发ADC测电流峰值
  - 启转时可能卡住，要保证不烧功率管
  - 扳手类逆转负载变小时(电流变小）停机
  - 出口产品需过IEC-60730B等认证

代码总体框架: 三部分独立工作

nuvoTon 新唐科技

```
main()   控制电压增减
{
    外设、全局变量的初始化
    while(1){
        …… 
        BLDC_Control();   //启转,停转,电压调节
        …… 
        其它任务
    }
}
```

只控制电流

PWM触发的ADC中断
负责测量和控制电流
优先级最高

只控制时间

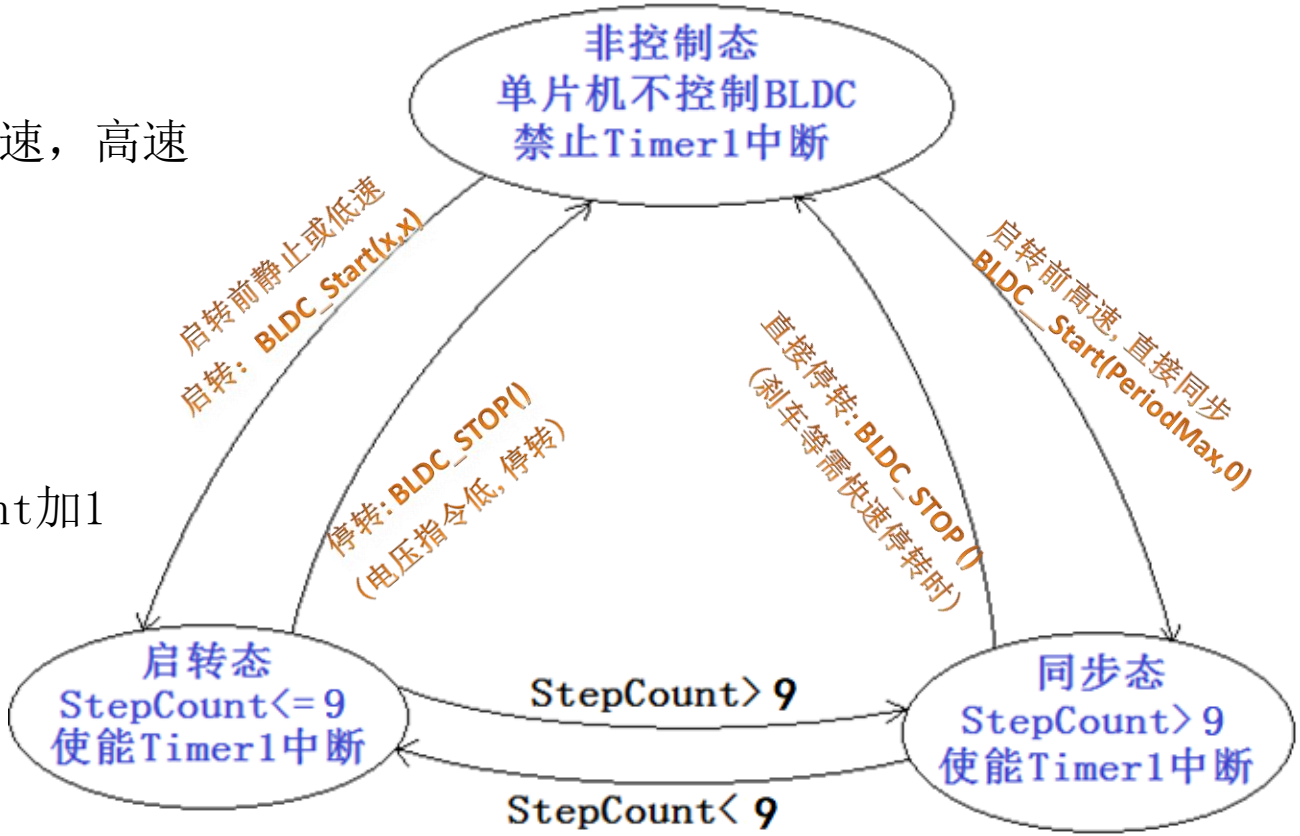
Timer1中断
负责检过0换相
优先级次高

三部分代码各施其职, 主循环执行时间再长，不影响检过0换相亦不影响电流测控，**保证了总是能正常转动，并且功率管不过流**

代码总体框架: 三个状态

三种启转方式：静止，低速，高速

每检到一次过0，StepCount加1



(检过0出错,比如转速太低时)

主循环: 调压调速等总的控制

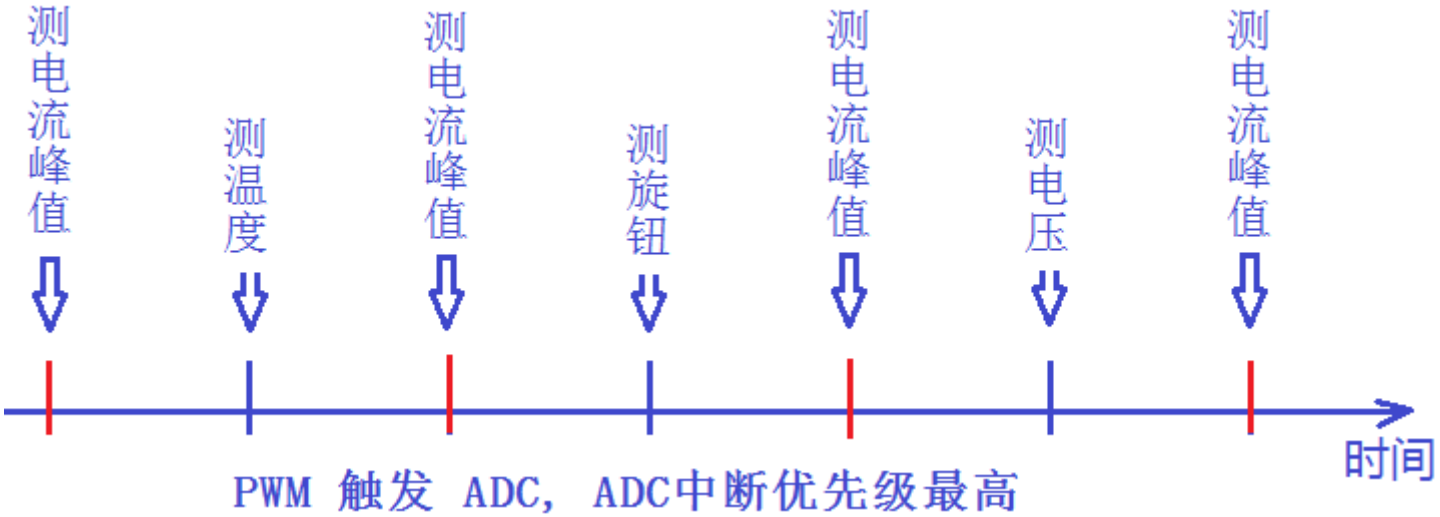

```
main()
{
    外设、全局变量的初始化
    while(1){
        … …
        BLDC_Control();
        … …
        其它任务
    }
}
```

函数内容

```
if(未转){
    if(符合启转条件){ 静止启转，或非静止启转 }
}
else{  BLDC_Modify()      // 在转
    if(过0次数 9 次以内){  // 启转阶段  }
    else{                  // 正常转动
        电压增减, 调超前角, 或按速度调电压
    }
    BLDC_Add_Duty();      //执行电压调整
    占空比更新到PWM寄存器
}
```

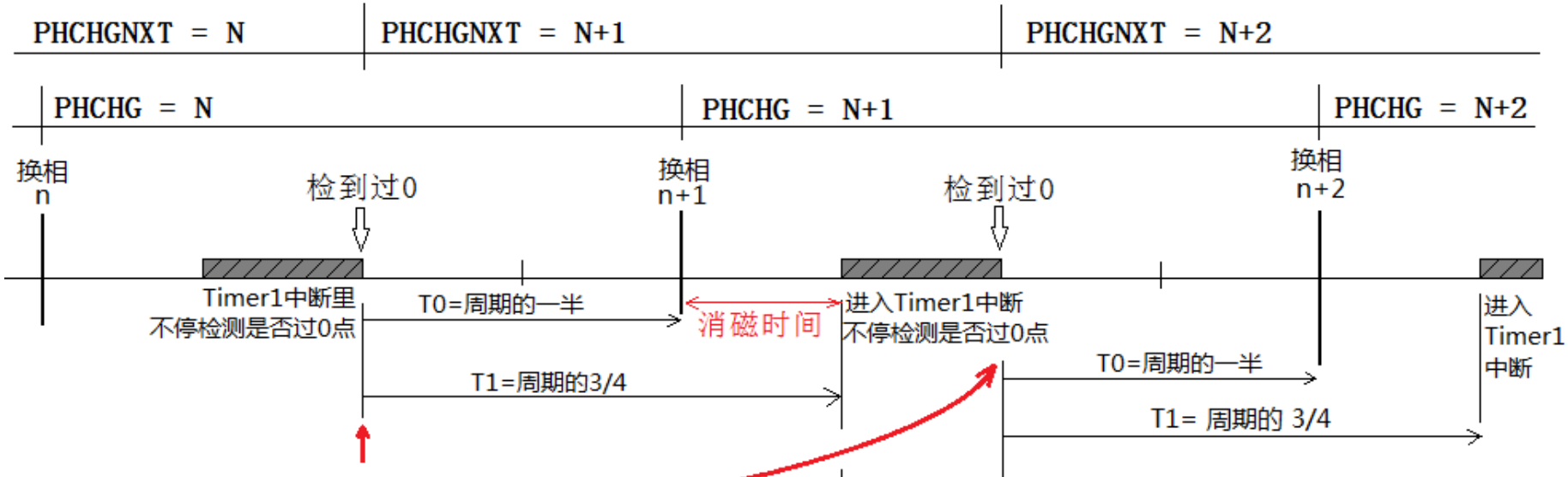
ADC中断：负责测控电流

- PWM触发ADC测电流，电压、温度等ADC软件触发或轮流转换
- 电流PI控制
- ADC中断优先级最高，执行频繁，所以代码要尽量少



Timer1中断: 负责检过0换相

检到过0后PWM->PHCHGNXT改变， 换相后PWM->PHCHG改变



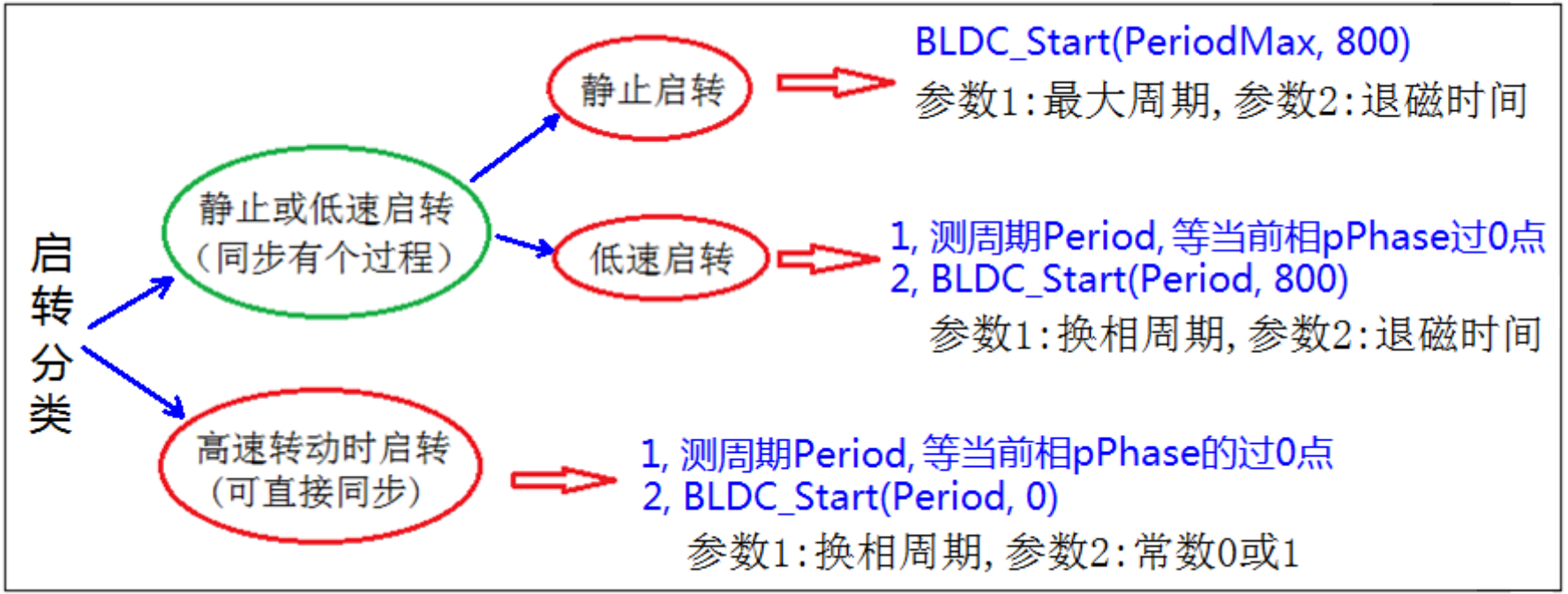
Timer中断代码里检到过0后：
1>更新PeriodNow, ZeroTick等全局变量
2>更新下一相输出寄存器PHCHGNXT
3>确定T0延时后，PHCHGNXT再更新到输出PHCHG
4>确定T1延时后，Timer1中断再次检过0

Timer1中断的优先级为1
仅次于Brake和ADC中断
高于其它中断

Timer1中断: 负责检过0换相

- Timer1配置为连续计数模式，可做通用延时。
- Timer1中断代码：检过0、换相、配置Timer1下次中断时刻，更新PeriodNow,ZeroTick等变量
  - 若使能了Timer1中断，即使电压为0，中断代码仍会周期性执行检过0换相。所以停转后要关Timer1中断。
  - 停转时一定要先禁止Timer1中断——防再次中断改PWM引脚输出，再关功率管，这部分代码见停机函数BLDC_Set_Free()

三种起转方式

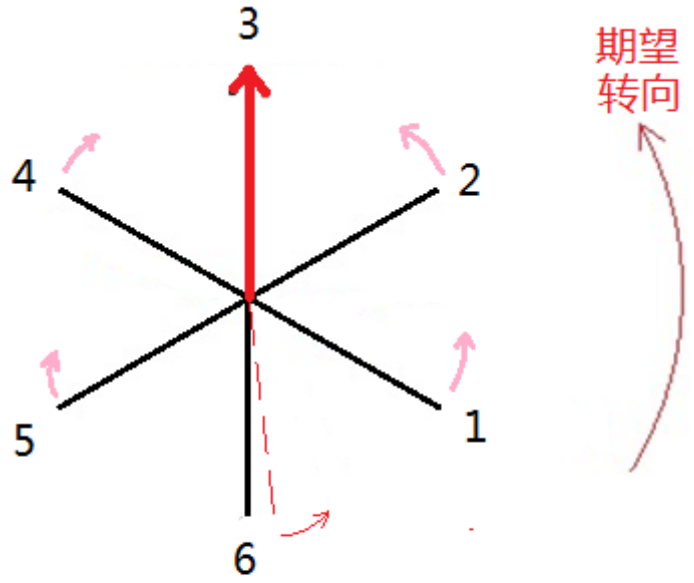

启转分类
静止或低速启转（同步有个过程）
静止启转
BLDC_Start(PeriodMax, 800)
参数1:最大周期, 参数2:退磁时间
低速启转
1, 测周期Period, 等当前相pPhase过0点
2, BLDC_Start(Period, 800)
参数1:换相周期, 参数2:退磁时间
高速转动时启转（可直接同步）
1, 测周期Period, 等当前相pPhase的过0点
2, BLDC_Start(Period, 0)
参数1:换相周期, 参数2:常数0或1

启转

- BLDC_Start()：启转函数，需启转时，在主循环中调用一次即可
  - 函数中做一些配置，函数返回时，状态与刚从Timer1中断返回一样。
  - 函数中，输出第一相,以后由Timer1中断检过0后换相
  - 不同启转方式，由参数决定,见前文
  - 停转时要先关Timer1中断，停止中断里写PHCHG再次让引脚输出
- BLDC_Modify()：转动调整函数,启转后，要在主循环中不停的调用
  - 参数值将在启转成功换相N次后,赋给 PhaseOffset 改变输出方式
  - 为减轻 CPU 负荷，可每次换相后，调用一次
- BLDC_Set_Free()：停转
  - 函数内先关闭Timer1中断,停止中断里改PWM引脚的输出，然后关闭六路功率管

启转

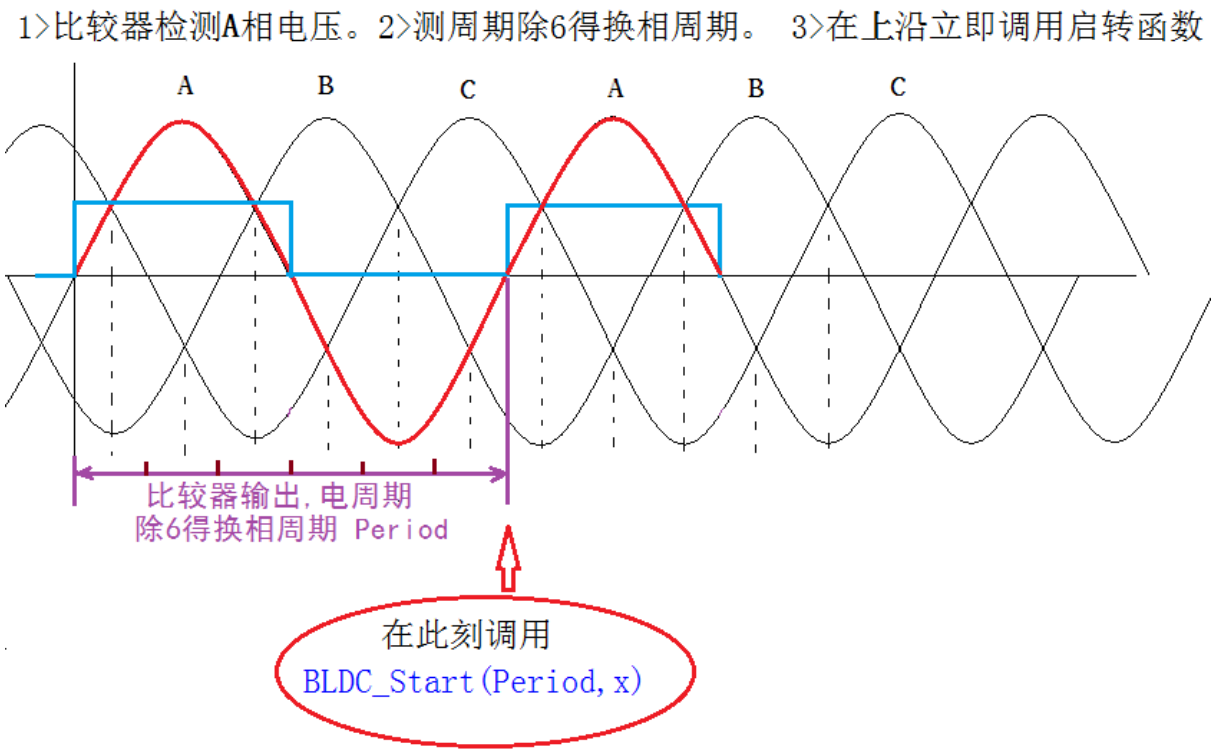
- **静止状态启转：**
  - 最末相*pPhaseEnd，当前相*pPhase，电压占空比CMR0等先赋值
  - 凸极性强的BLDC 可测到磁体位置确定当前相的值
- **First_Delay：静止启转前电力托动时间**
  - 取值参考全速时、换相时间的10倍左右，一般3ms~15ms，此参数不敏感



先给2相通电First_Delay，
若磁体在6相位置，会偏右转一点
再从第3相启转，位置6会顺利启转

◆非静止启转

- 假如启转相为：CB通电，检A相上沿过0



部分全局变量

- PhaseAngle：超前角,越小越超前,值为30不超前
- TabPhase[]：转动数据，6个一组，每组决定一种电机驱动方式
  - 由相偏移变量PhaseOffset=0/6/18决定用哪六个数据转动，改变功率管驱动方式
  - 转动数组有两个,一个正转，另一个反转,具体见代码
- Duty_Command：占空比命令值，一般是调速旋钮的 ADC 值
  - 值由旋钮ADC 获得，或调速脉冲PPM 中断获得。
- Duty_Limit：占空比上限,慢慢增加直到等于Duty_Command
  - 此值写入 PWM 寄存器，直接决定输出占空比。
- Duty_Min：最小启转占空比
  - 小于此值不起转

初始化配置注意事项

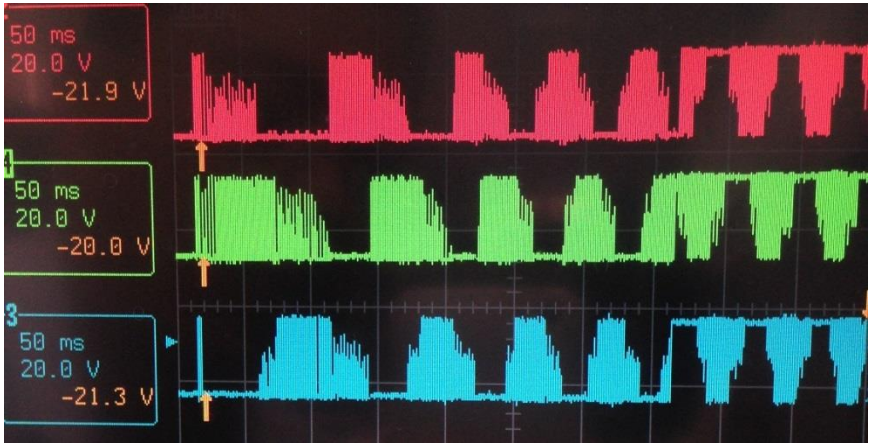
- 尽早配置 ADC，因为50us后才稳定工作
- 先配置PWM工作，再把引脚配成PWM输出，顺序不能错，防止出现短暂的输出电平不对
- 检查BOD复位电压值是否正确,防止Config烧录出错
- 其它自检，全局变量初始化
- 最后测“0电流”ADC值,计算电流

启转波形

StartOffset = 0; //启转时上管做PWM

… …

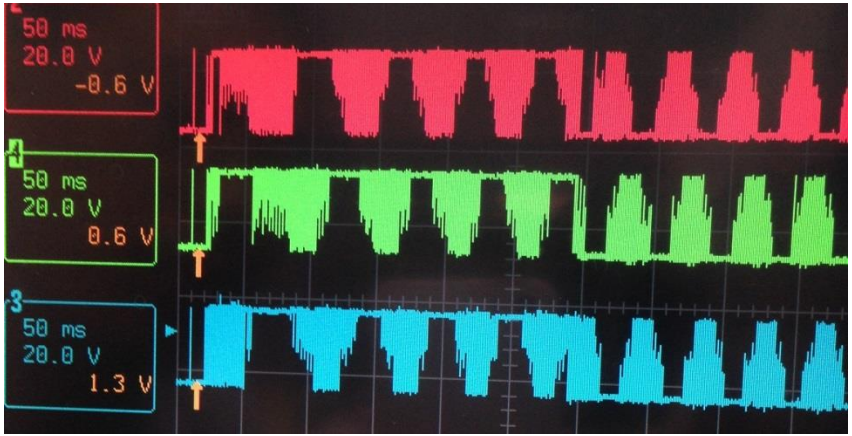
BLDC_Modify(18,30) ; //启转后下管做PWM



StartOffset = 18; //启转时，下管做PWM

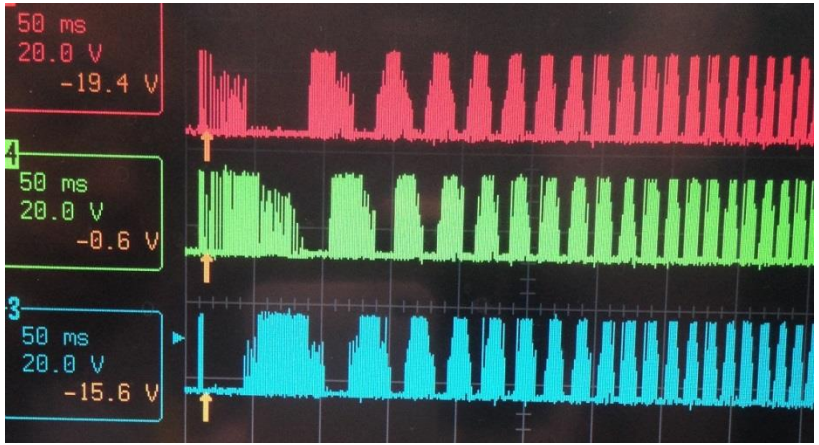
… …

BLDC_Modify(0,30) ; //启转后，上管做PWM

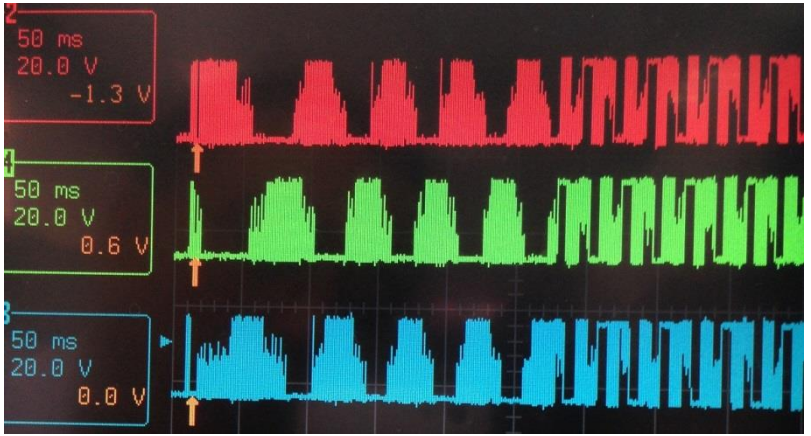


启转波形

//启转与正常转动都是上管做PWM
StartOffset = 0;
… …
BLDC_Modify(0,30) ;



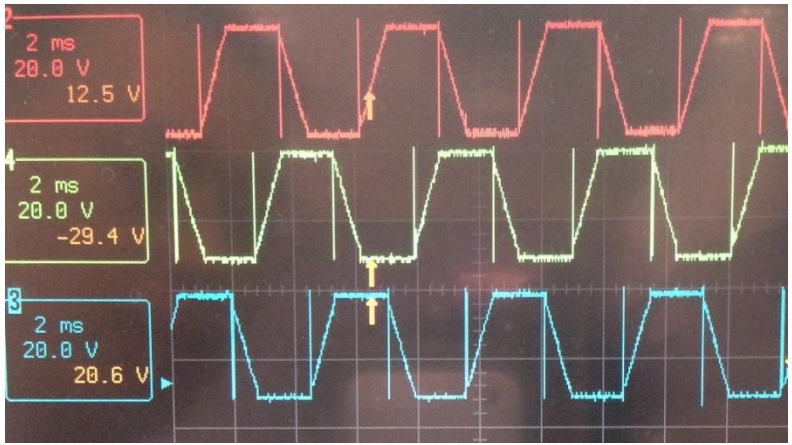
//启转时上管做PWM，正常转上下管轮流做PWM
StartOffset = 0;
… …
BLDC_Modify(18,30) ;



转动波形

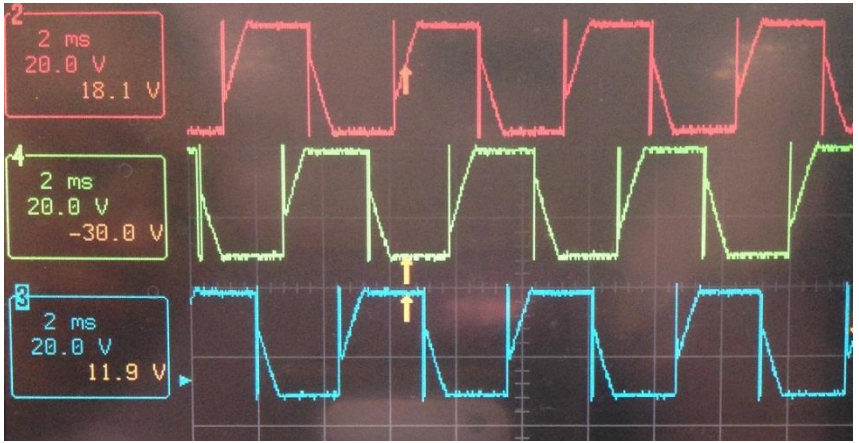
//换相即不超前， 也不滞后

PhaseAngle = 28; //调整此值, 让波形左右对称



//换相滞后, 转矩大

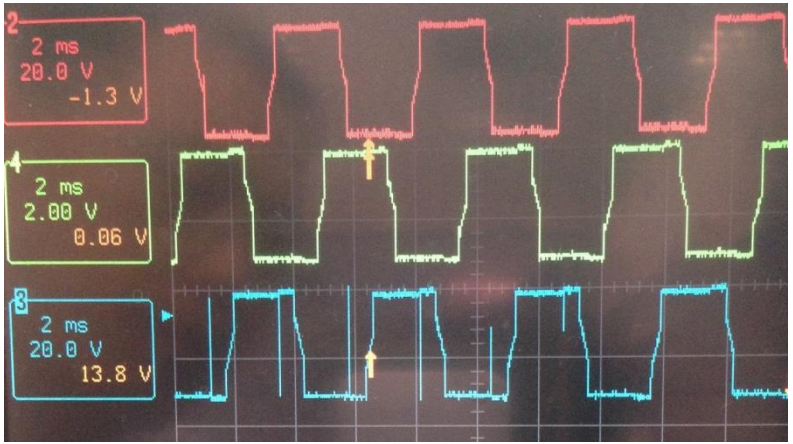
PhaseAngle = 42;



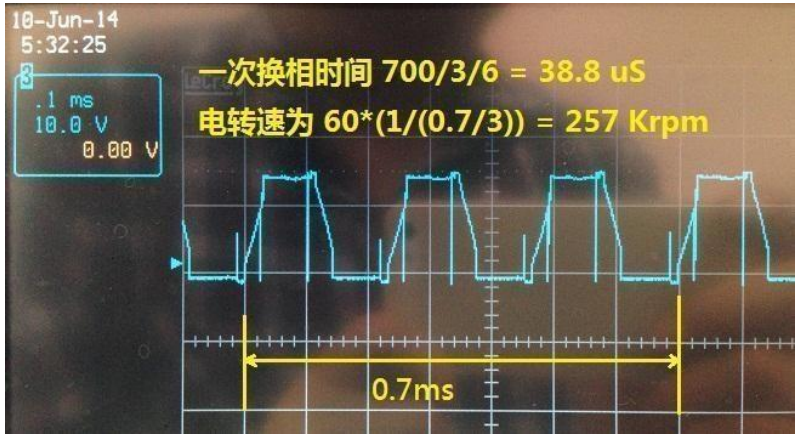
转动波形

//换相超前，转速加快

PhaseAngle = 0;



转速到25.7万转的波形



nuvoTon

Thank You !