

# BLDC 方波速度控制

实验指导书

2024.4.11

## 目录

第一章 BLDC 方波速度开环控制实验	3
1.1 实验目的	3
1.2 预习要求	3
1.3 实验设备	3
1.4 实验原理	3
1.4.1 无刷直流电机基础	3
1.4.2 六步换相原理	4
1.4.3 BLDC 换相电路	5
1.4.4 BLDC 换相控制逻辑	6
1.4.5 BLDC 速度控制	9
1.4.6 转速测量	10
1.4.7 电路分析	11
1.5 实验前检查	13
1.6 实验内容	13
1.7 实验步骤	15
1.7.1 配置 CubeMX	15
1.7.2 程序编写与调试	26
1.8 实验注意事项	28
1.9 实验报告要求	29
第 2 章 BLDC 方波速度闭环控制实验	30
2.1 实验目的	30
2.2 预习要求	30
2.3 实验设备	30
2.4 实验原理	30
2.4.1 模拟 PID 控制原理	30
2.4.2 位置式 PID 算法	32
2.4.3 增量式 PID 算法	33
2.4.4 控制器参数整定	34
2.4.5 参数调整规则的探索	36
2.5 实验内容	36
2.6 实验步骤	37
2.7 实验报告要求	38

# 第一章 BLDC 方波速度开环控制实验

## 1.1 实验目的

- 1) 掌握无刷直流电机的组成、工作原理及特点。
- 2) 初步了解STM32F407x 高级定时器的工作原理。
- 3) 熟悉逆变器的工作原理和控制方法。
- 4) 了解STM32F407x 控制无刷直流电机的方法。

## 1.2 预习要求

- 1) 分析掌握无刷直流电机的运行原理。
- 2) 掌握实验用电机参数。
- 3) 了解无刷直流电机的控制方法。
- 4) 熟悉逆变器工作原理。
- 5) 熟悉STM32F407x 高级定时器PWM 输出模式工作原理。

## 1.3 实验设备

- 1) 实验箱 1 台
- 2) 计算机 1 台
- 3) 示波器 1 台
- 4) 万用表 1 个

## 1.4 实验原理

### 1.4.1 无刷直流电机基础

普通直流电动机的电枢在转子上，而定子产生固定不动的磁场。为了使直流电动机旋转，需要通过换向器和电刷不断改变电枢绕组中的电流方向，使定、转子两个磁场的方向始终保持相互垂直，从而产生恒定的转矩驱动电动机不断旋转。无刷直流电机为了取消电刷，将电枢绕组移至定子上，而转子由永磁体构成，结构与有刷直流电机正好相反；然而即使这样改变还不够，因为定子通入直流电后，只能产生不变的磁场，电机依然转不起来。为了使电机转子旋转起来，必须使定

子电枢各相绕组不断地依次换相通电，这样才能使定子磁场随着转子位置不断变化，产生转矩推动转子旋转。

无刷直流电机利用了通电线圈和永磁体的相互作用原理，BLDC 内部结构实物图和逻辑结构图见图 1-1。

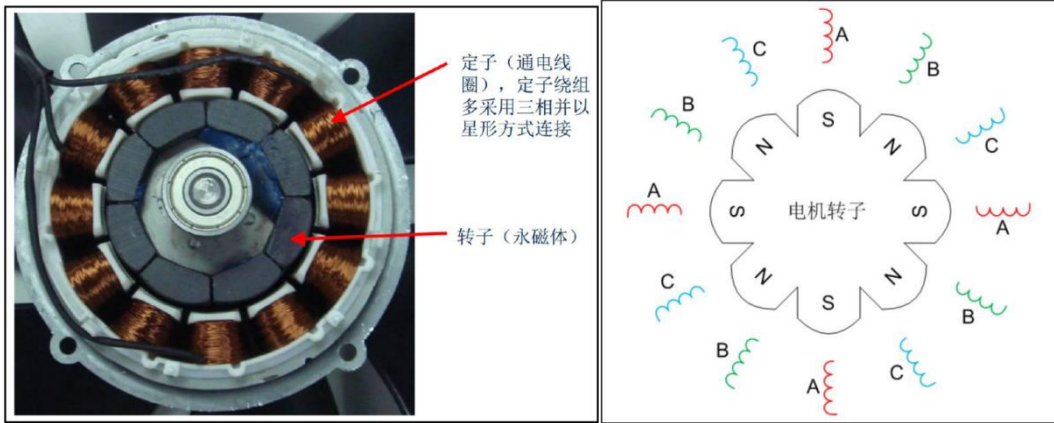


图 1-1 电机实物图与逻辑结构图

### 1.4.2 六步换相原理

在 Y 型连接的 BLDC 电机中，电机具有三条引线，每条引线与一个绕组相连。每个绕组（或串联绕组）与一个公共点相连，该公共点为所有三个绕组共同使用。在图 6.1 中，R 为定义为红色，G 定义为绿色，B 定义为蓝色。为简化分析，可以做成直流无刷电机的简化逻辑结构，见图 1-2。电机外层是定子，包含电机绕组。电机内部是转子，转子由围绕电机圆周的极性相反的磁极组成，图中显示了仅带有两个磁极（南北磁极）的转子。在实际应用中，大多数电机的转子具有多对磁极。

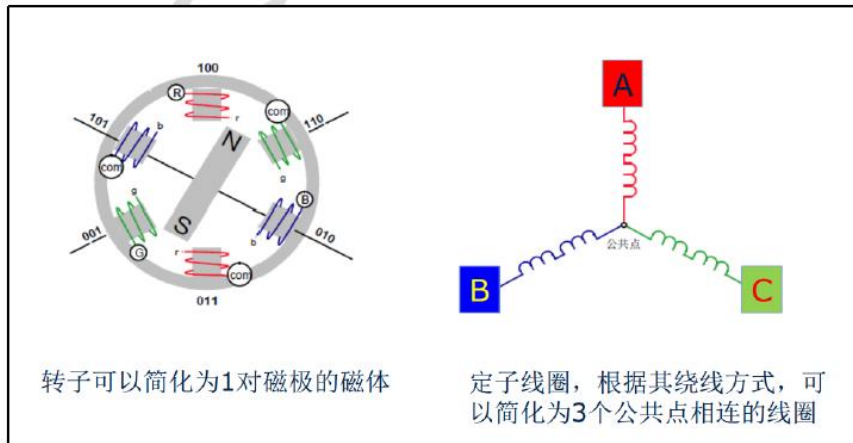


图 1-2 简化逻辑结构图

这样，当电流流过电机绕组时，通电的线圈会产生各自的磁场，他们的合成磁场满足矢量合成的原则，由 N 和 S 标记指示。然后转子旋转，以使转子的北磁极与定子磁场的南磁极对齐。同样的，转子的南磁极与定子磁场的北磁极对齐。直流无刷电机的 6 拍工作方式，线圈产生旋转磁场。图 1-3 说明了六步换相的工

作原理。每一步或每一区间，相当于 60 个电角度。六个区间组成了 360 个电角度或一次电气旋转。

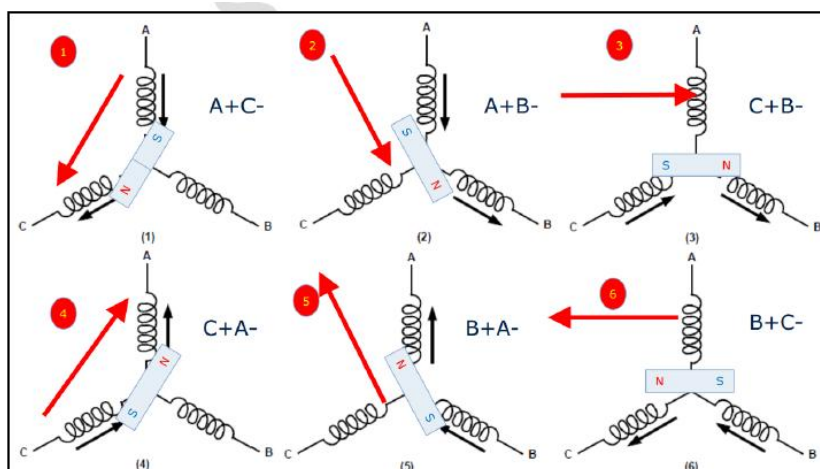


图 1-3 六步换相原理

每一区间上都有两个绕组通电，一个绕组未通电。每一区间均有一个绕组未通电，是六步控制的重要特征。六步循环加电，可完成电机按指定方向旋转。要保持定子中的磁场超前于转子磁场，那么一区间到另一区间的转变必须发生在转子处于特定位置的时刻，从而获得最佳转矩。

### 1.4.3 BLDC 换相电路

有了上面的原理分析，现在想让 BLDC 旋转起来的一个问题就是如何任意的控制 A、B、C 线的电压。答案是可以用一个三相逆变器电路来实现，这里的每个桥臂都有两个电子开关，电子开关可以选择是功率 MOSFET 或者 IGBT，IGBT 用于超大功率电机驱动。最终搭建起来的电路见图 1-4:

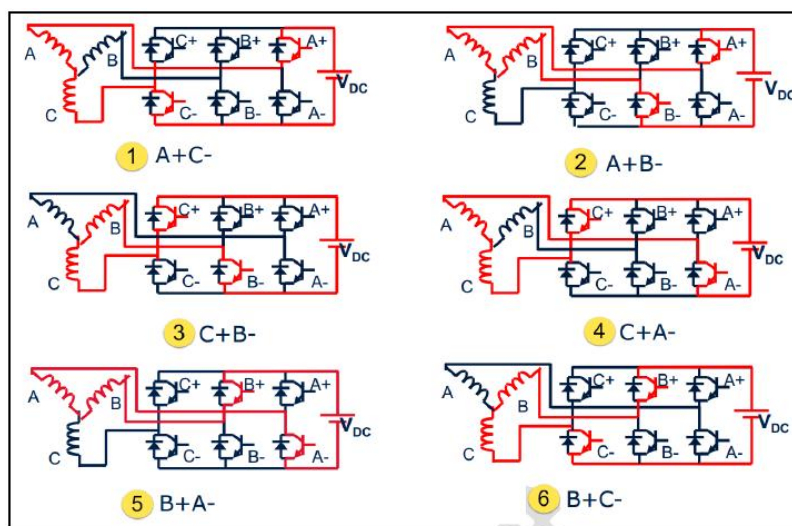


图 1-4 BLDC 换相电路

通过控制此六个开关管的开关顺序可实现的的不同绕组加电，完成六步换相要求。当开关管 T1 和 T4 导通，其它开关管截止时，电流将从绕组 A 端流入 B 端

流出；当开关管 T1、T6 导通，其它开关管截止时，电流将从绕组 A 端流入 C 端流出；当 T3、T6 导通，其它开关管截止时，电流将从绕组 B 端注入 C 端流出；以此类推，可按要求实现不同绕组加电，当 T5、T4 导通，其它开关管截止时，电流将从绕组 C 端注入 B 端流出。这样，最终，我们可以让 STM32 控制 A+、A-、B+、B-、C+以及 C-这六个 MOS 管的通断情况就可以让电机旋转起来。

### 1.4.4 BLDC 换相控制逻辑

上面是解决了绕组电压控制，还有一个问题就是究竟什么时候要给哪个绕组正电压、给哪个绕组负电压以及哪个绕组悬空，就是具体当前时刻要选择图 1-4 中“哪一步”？所以，BLDC 驱动还需要一个非常重要的参数，只有知道了这个参数信息我们才有可能正常的控制 BLDC 旋转，这个重要参数就是转子的位置信息，只有知道了当前转子所处位置，我们才能很好的控制电机旋转，如果毫无目的根据图 1-4 中顺序为绕组给电，最终只能看到电机乱转。

转子的位置信息可以通过三个霍尔传感器获取。

当霍尔在和电机的转子做相对运动时，会随着转子下磁密度的变化，产生变化的信号，见图 1-5。

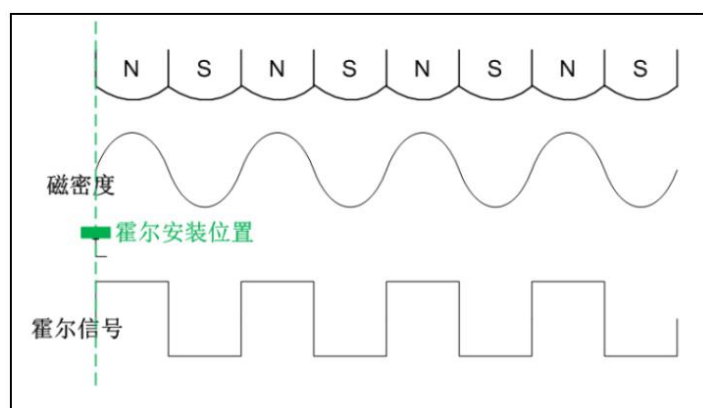


图 1-5 BLDC 霍尔信号

在有霍尔传感器的无刷直流电机中，一般安装 3 个霍尔传感器，可以有效的反应转子的位置。一般采用间隔  $60^\circ$  和  $120^\circ$  的方式按圆周分布。如果间隔  $60^\circ$ ，则输出波形相差  $60^\circ$  电角度。如果间隔  $120^\circ$ ，则 3 个霍尔传感器的输出波形相差  $120^\circ$  电角度，输出信号中高、低电平各占  $180^\circ$  电角度。如图 1-6 所示。

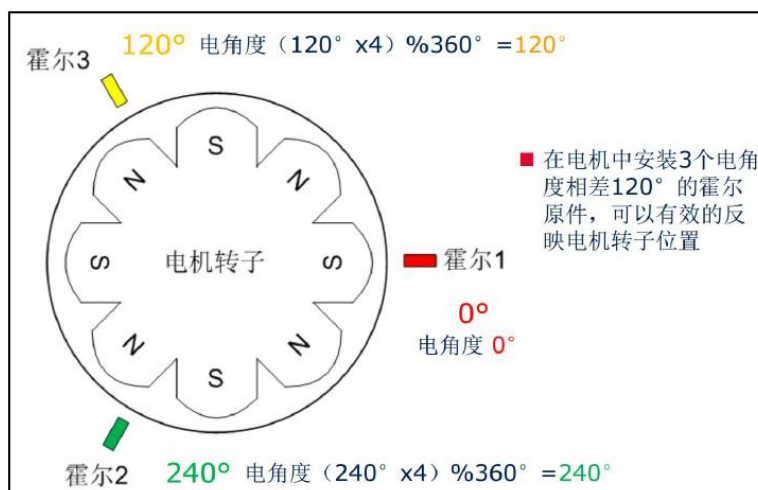


图 1-6 霍尔安装示例

间隔  $60^\circ$  和  $120^\circ$  按圆周分布。如果间隔  $60^\circ$ ，则输出波形相差  $60^\circ$  电角度。如果间隔  $120^\circ$ ，则 3 个霍尔传感器的输出波形相差  $120^\circ$  电角度，输出信号中高、低电平各占  $180^\circ$  电角度。

以  $120^\circ$  霍尔式位置传感器为例，三相无刷直流电机反电势和传感器输出信号间相位关系见图 1-7。

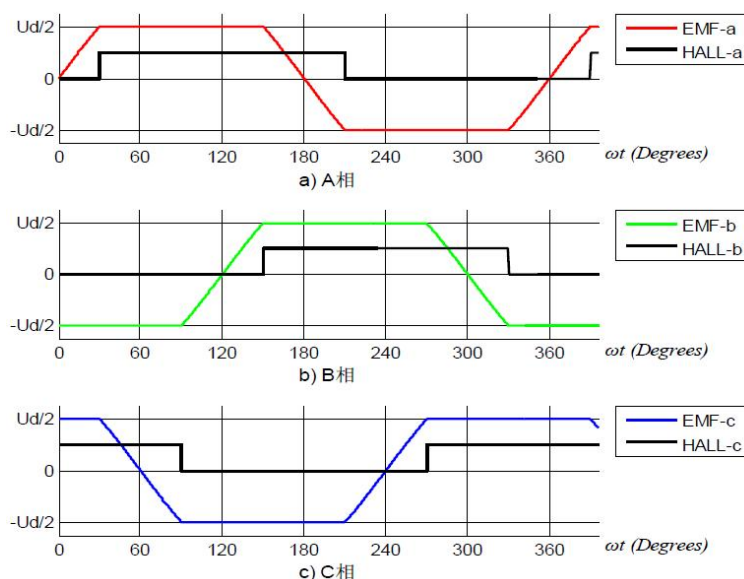


图 1-7 BLDC 反电势和传感器信号

图中，HALL-a、HALL-b、HALL-c 为三个霍尔传感器在电机运转中的波形。EMF-a、EMF-b、EMF-c 为电机反电势电压波形。由图可知，无刷直流电机反电势为正负半波皆有  $120^\circ$  平台的梯形波，三相间相差为  $120^\circ$ ；3 个位置传感器（电角度间距  $120^\circ$  传感器）相差为  $120^\circ$ ，其上升或下降沿位置即对应定子电枢绕组导通时刻。

如此，可通过对霍尔传感器信号的上升或下降沿进行检测，经由功率 MOSFET 或 IGBT 功率开关器件构成的电子换相电路实现换相，使电枢绕组依次通电，从而在定子上产生跳跃式的旋转磁场，驱动永磁转子旋转。随着转子的

旋转，位置传感器输出信号不断变化，电枢绕组的通电状态随之改变，使得在某一磁极下导体的电流方向始终保持不变，这就是无刷直流电机的无接触换相过程。

结合之前介绍的 BLDC 六步控制，在每个霍尔信号都对应一个 BLDC 控制步，霍尔信号和定子通电情况有一一对应的关系，如图 1-8 所示。

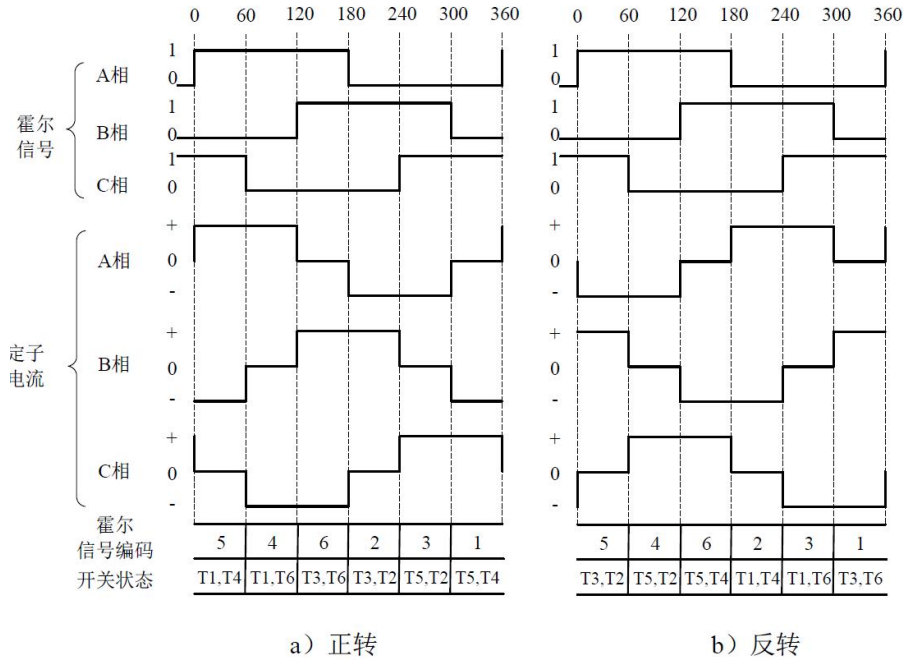


图 1-8 HALL 无刷电机正反转换相

根据该图我们可以得到六步换相法控制 BLDC 电机中的最重要的参数表格——换相控制真值表，在我们的实验中，BLDC 的换相控制真值表如表 1-1 所示：

表 1-1 HALL 换相真值表

	霍尔 C	霍尔 B	霍尔 A	通电方向	A+	A-	B+	B-	C+	C-
	正转	1	0	1	AB	开通	关断	关断	开通	关断
0		0	1	AC	开通	关断	关断	关断	关断	开通
0		1	1	BC	关断	关断	开通	关断	关断	开通
0		1	0	BA	关断	开通	开通	关断	关断	关断
1		1	0	CA	关断	开通	关断	关断	开通	关断
1		0	0	CB	关断	关断	关断	开通	开通	关断

	霍尔 C	霍尔 B	霍尔 A	通电方向	A+	A-	B+	B-	C+	C-
	反转	1	0	1	BA	关断	开通	开通	关断	关断
0		0	1	CA	关断	开通	关断	关断	开通	关断
0		1	1	CB	关断	关断	关断	开通	开通	关断
0		1	0	AB	开通	关断	关断	开通	关断	关断
1		1	0	AC	开通	关断	关断	关断	关断	开通
1		0	0	BC	关断	关断	开通	关断	关断	开通

特别注意，一般 BLDC 厂家都会给出一个霍尔传感器和绕组得电情况对应



关系表，不一定跟上面两个表都完全对应一致，但是原理分析都是一致的。

上面两个表的对应意思就是：当检测到霍尔传感器信号为某个值时，控制六个桥臂对应的开关状态。例如，我们想让电机正转，就用表格 10-1 中的对应信息，假设 STM32 检测到当前的霍尔信号为：霍尔#1、霍尔#2、霍尔#3 分别对应为 1、0、1，那么此时我们应该让 STM32 控制 A+和 B-桥臂导通，而其他桥臂都关断，在 A-和 C+桥臂导通情况下，电机的转子会向着一个位置旋转；在旋转到达目标位置之前，霍尔传感器信号就会发生改变，此时变为：霍尔#1、霍尔#2、霍尔#3 分别对应为 1、0、0，好了，此时我们马上把 B-桥臂关断，而把 C-桥臂开通，即此时 A+和 C-桥臂导通，其他桥臂关断，电机就向旋转一个角度。这样，如此循环下去，电机就可以不停的旋转了。

此时，有些人肯定想问：如果我不管霍尔信号变化，就按中任一种其中给电，电机会怎样的呢？电机会固定在一个位置，实际上，这种情况是很危险的，我们知道，绕组都是漆包线铜丝，电阻非常小，**当总是给电时候，电路中电流就非常大，严重情况，烧毁电机或者电源！**因此，在六步换相控制逻辑编程过程中，**要谨慎使用单步调试！**

特别的，如果直接导通 A+和 A-这两桥臂，或者 B+和 B-这两桥臂，或者 C+和 C-这两桥臂会出现什么情况呢？**结果就是电源必烧无疑，这些情况相当于电源正负极直接短路，所以我们在编程控制开关管导通的过程要非常小心!!!**

在本实验中，我们采用 TIM2 的 CH1、CH2、CH3 三个输入捕获通道作为霍尔信号输入通道，采用定时 2 的捕获输入中断进行换相。当三个霍尔信号的任何一个产生上升或者下降沿时，触发定时器 TIM2 的输入中断（中断函数为 void HAL\_TIM\_IC\_CaptureCallback(TIM\_HandleTypeDef \*htim)），在中断中根据正反转 HALL 换相表完成换相操作。

### 1.4.5 BLDC 速度控制

一般情况下，我们不仅仅要让电机可以旋转，还必须控制它的速度，BLDC 电机调速采用的是脉冲宽度调制技术（PWM），见图 1-9。

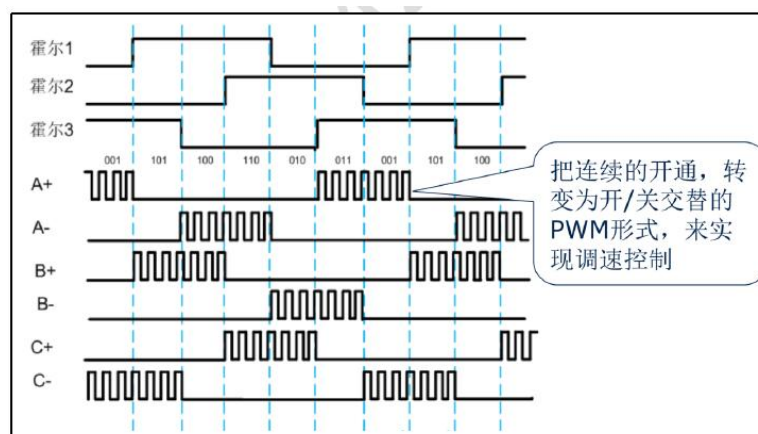


图 1-9 BLDC PWM 调制调速

以霍尔#1、霍尔#2、霍尔#3 分别对应 1、0、1 情况为例（其他五种情况都是相同原理），此时应该是 B+和 C-导通，其他桥臂关断。若此时是 100%导通 B+和 C-这两个个桥臂，不用 PWM 控制其导通时间，那么此时绕组中电流非常

大，转子转动速度非常高。显然这种情况不是我们在实际上应用用常常用到的。在实际生产中，我们总是需要控制电机的转速。现在引入 PWM 控制后什么情况呢？PWM 中一个常用到的名词就是占空比，就是高电平时间占总周期时间的比例，这样我们将 B+和 C-直接导通代换成高频率（一般 10 几 KHz 或者几十 KHz）的 PWM，保证 B+和 C-的 PWM 频率相等，并且周期起始位置相同，这样我们可以非常方便地通过调整占空比的大小，来控制 B+和 C-这两臂的实际导通时间，加上电机绕组本身是感性负载，这样整体加在电机绕组上的电压就是 0V 到电源正电压（24V）之间，最终实现控制电机转动速度。综上，通过控制导通的两个桥臂的 PWM 占空比我们可以非常方便控制转速了。

同时使用 PWM 控制 B+和 C-桥臂，称之为 pwm-pwm 型调制方式，实际上，还是有另外几种调制方式，比如 H\_on-L\_pwm 型、H\_pwm-L\_on 型、pwm\_on 型和 on\_pwm 型等等不同的调制方式，如图 1-10。不同控制方式在性能上有不同的效果，针对实际的应用场合可以尝试多种调制方式，然后选择最优方式。

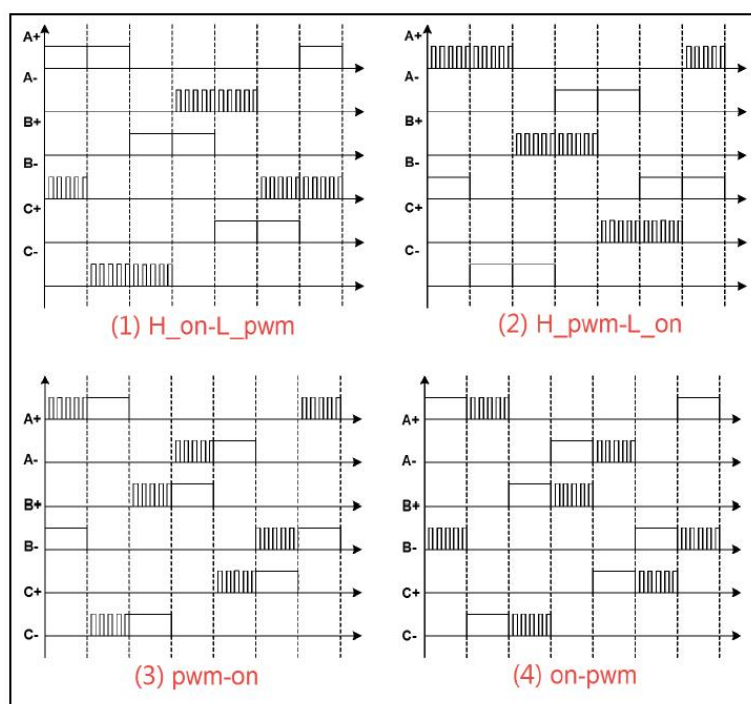


图 1-10 PWM 调制方式

在本实验中，我们采用的是第二种，H\_pwm-L\_on 型。上臂桥采用 TIM8 的 CH1 CH2 CH3 输出 PWM 控制电压，下桥臂使用 GPIO\_Output 控制通断。

### 1.4.6 转速测量

无论是做速度开环控制还是闭环控制，电机转速是必须要获取的。转速测量有很多种方式，可以使用 HALL 传感器、编码器或者无传感器的速度估计算法来实现，本实验中我们使用 HALL 传感器测量速度。

由上面的分析可知，霍尔传感器每产生一个跳变沿，则换相一次，电机旋转 60° 电角度。霍尔信号变化和电角度变化关系如图 1-11 所示。

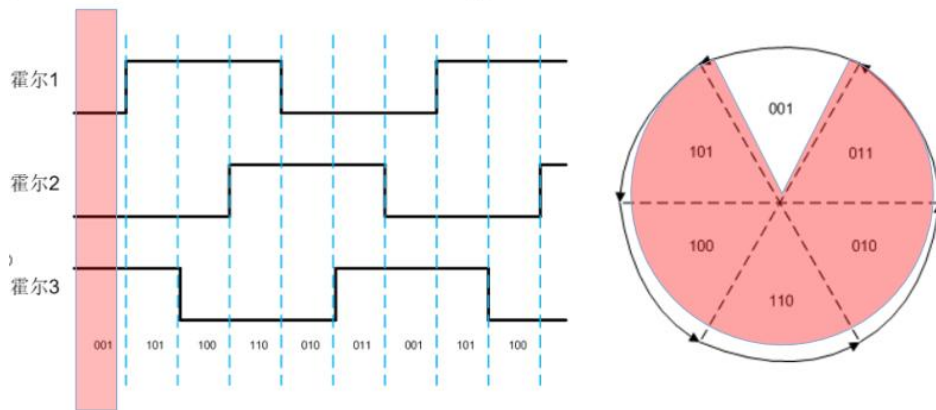


图 1-11 Hall 信号与电角度

由此我们可以得到如下的转速计算公式，其中  $p$  为电机极对数， $t$  为两次换相的时间间隔。

$$\text{speed} = 60/p/t/360 \text{ (转每秒)}$$

因此，若我们得到了两次霍尔跳变沿的时间间隔  $t$ ，即可得到转速。

在本实验中，我们使用 TIM3 作为 1ms 定时器。可以用 TIM3 测量两次进入 TIM2 输入捕获中断的时间间隔，这个时间间隔即为  $t$ 。

### 1.4.7 电路分析

根据以上的讲解，我们可以得到如图 1-12 的 BLDC 霍尔传感器控制框图。如图所示，在实验中，我们采用 TIM2 的 CH1、CH2、CH3 三个输入捕获通道作为霍尔信号输入通道，采用 TIM8 的 CH1 CH2 CH3 输出 PWM 控制电压，下桥臂使用 GPIO\_Output 控制通断。

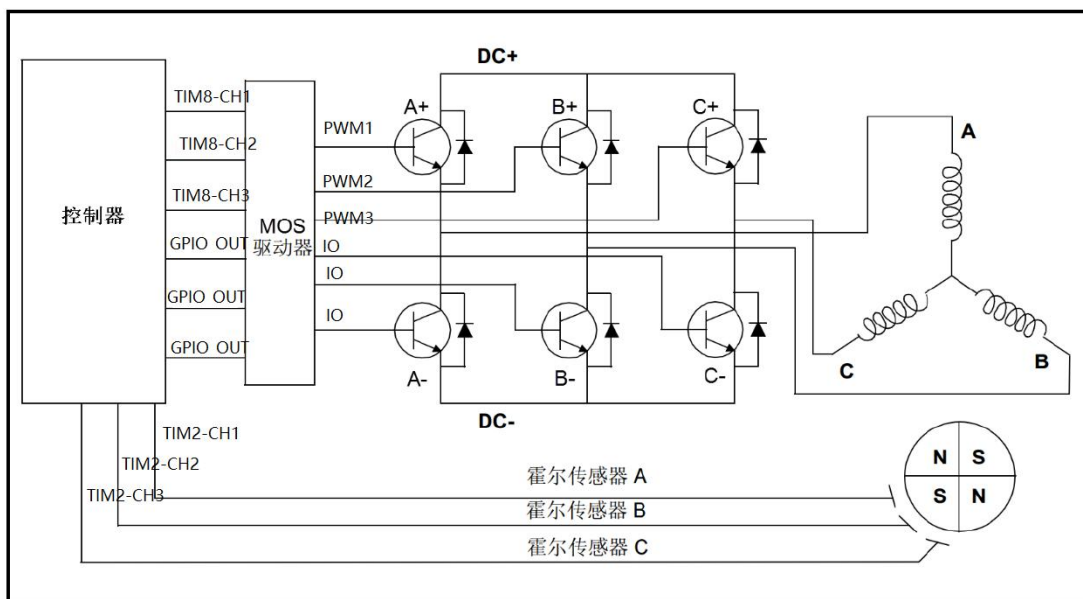


图 1-12 Hall 传感器模式控制框图

为了实现 STM32 的控制功能，接下来我们关心的就是，这些控制引脚的具体位置。这需要通过查看功率板和主控板的电路原理图得知，功率板和主控板的接口处的电路原理图如图 1-13、1-14 所示。

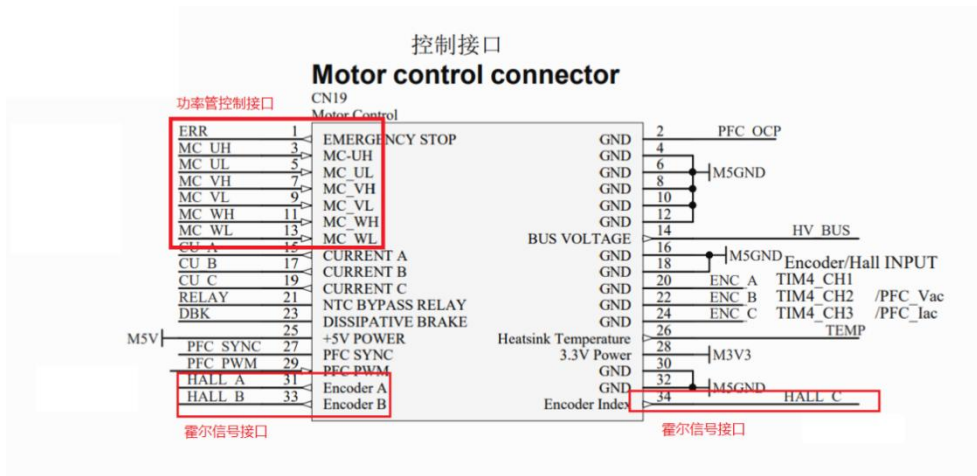


图 1-13 功率板接口电路原理图

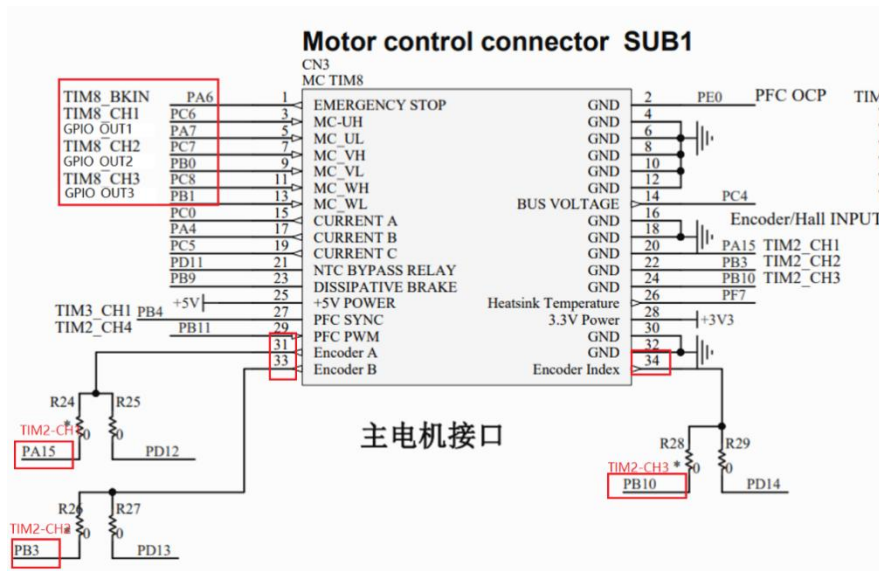


图 1-14 主控板接口电路原理图

由图 1-13 我们可以得知功率板上的逆变器控制接口和 HALL 信号接口的引脚号，功率板上的引脚号和控制板上的引脚号是一一对应的，控制板上的引脚号都标注了所连接的 STM32 的引脚号，因此，最终我们可以得到各个控制引脚的具体位置，可以得到如下表 1-3。

表 1-3 控制接口引脚对应关系

驱动接口	MC_UH	MC_VH	MC_WH	MC_UL	MC_VL	MC_WL	Hall_A	Hall_B	Hall_C
真值表	A+	B+	C+	A-	B-	C-	霍尔 A	霍尔 B	霍尔 C

控制接口	TIM8 _CH1	TIM8 _CH2	TIM8 _CH3	GPIO _OUT 1	GPIO _OUT 2	GPIO _OUT 3	TIM2 _CH1	TIM2 _CH2	TIM2 _CH3
驱动引脚	3	7	11	5	9	13	31	33	34
STM 引脚	PC6	PC7	PC8	PA7	PB0	PB1	PA15	PB3	PB10

可以根据以上表格在 CubeMX 里对 STM32 管脚进行配置。

## 1.5 实验前检查

- 1) 逆变器硬件检查：打开试验箱上电开关，通电后电机驱动电路板5V指示LED点亮。
- 2) 电机检查：检查电机接线是否完好，有无安装松动的现象。

## 1.6 实验内容

根据以上实验原理的介绍，采用六步换相法实现直流无刷电机旋转的开环控制。读取电位器的值，作为 PWM 波的占空比，实现旋转电位器调节转速的效果。按 SEL 键（PG10）启停电机控制。在 LCD 上实时显示电位器数值、PWM 占空比、实测速度、母线电压等参数。主要功能的实现在原理部分已经做了提示，总结如下。

- 1) TIM3为1MS定时器，用于定时更新中断计时功能。
- 2) 上臂桥采用TIM8的CH1 CH2 CH3输出PWM控制电压，下桥臂使用GPIO\_OUT驱动方式驱动电机。
- 3) TIM2的CH1 CH2 CH3作霍尔输入通道，采用定时2的捕获输入中断进行换相。
- 4) 测量TIM2两次中断时间间隔，计算转速。
- 5) 用AD采集电位器，使用PF9读取电位器电压值并设定PWM波占空比，使用DMA自动采集传输，主程序中禁止DMA中断，旋转电位器进行调速。
- 6) 按SEL键启停电机控制，在LCD上实时显示电位器数值、PWM占空比、实测速度等、母线电压参数。母线电压的测量需要查看功率板电路原理，如图1-15所示。

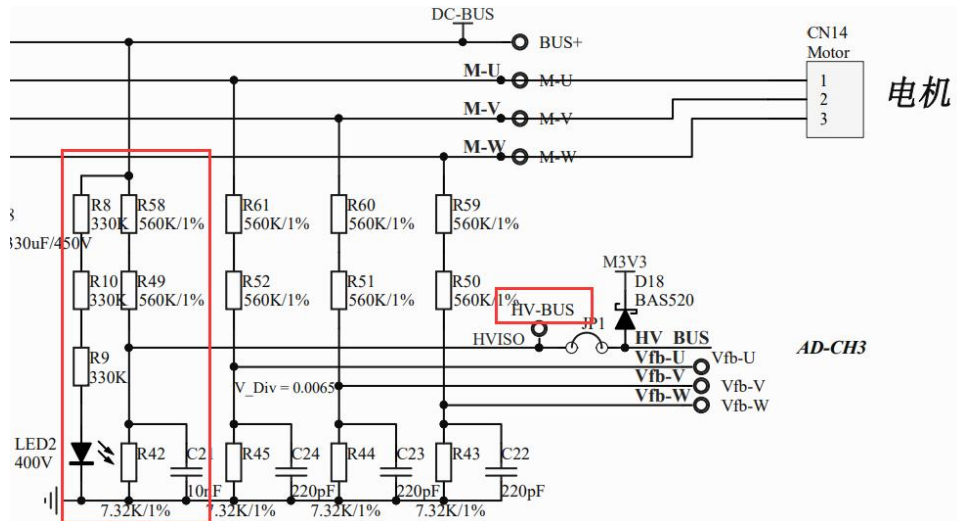


图 1-15 功率板电路原理图（部分）

- 7) 使能TIM8 BKIN，在程序中添加错误处理机制，使电机在换相错误等紧急状态下可以快速制动。
- 8) 采集 U、V、W 三相中任何一相的反电动势波形，以及 Hall\_A、Hall\_B、Hall\_C 中任何一相霍尔传感器波形，一同显示在示波器中。在测量时，注意示波器的负极一定要与测量信号共地。测量时时我们使用探针的针式接头，需要将探针帽摘下来，如图 1-16 所示，



图 1-16 将探针帽摘掉

将探针负极鳄鱼夹通过杜邦线连接到功率板的 GND 端子上（**切勿直接夹取信号!!!**），**找准测点位置**，正极探针点在测点处（**小心操作!**），如图 1-17 和图 1-18 所示，

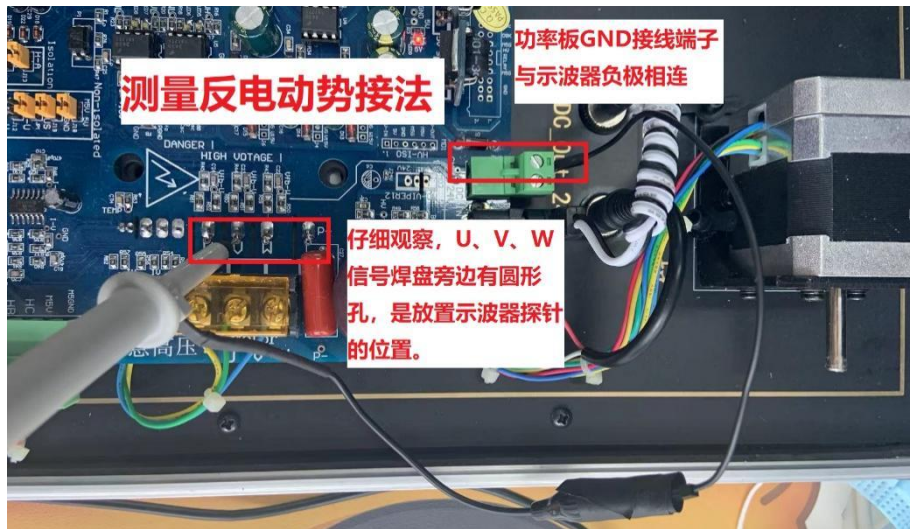


图 1-17 U、V、W 三相反电动势波形测量方法

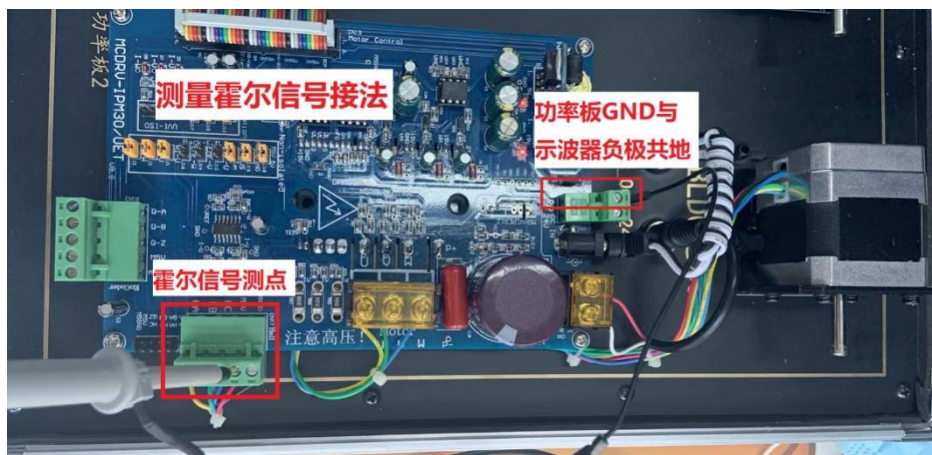


图 1-18 测量霍尔信号接法

- 9) 调节PWM波载波频率，观察不同载波频率下的反电动势波形和霍尔传感器波形，记录参数和波形，结合波形思考PWM载波频率对电机转动的影响。

**注意：PWM 波载波频率 $\leq 20$  KHz。**

## 1.7 实验步骤

### 1.7.1 配置 CubeMX

- 1) 新建 CubeMX 工程，参考前述实验。
- 2) 配置时钟，参照前述实验。
- 3) 配置调试引脚，如图 1-19 所示。

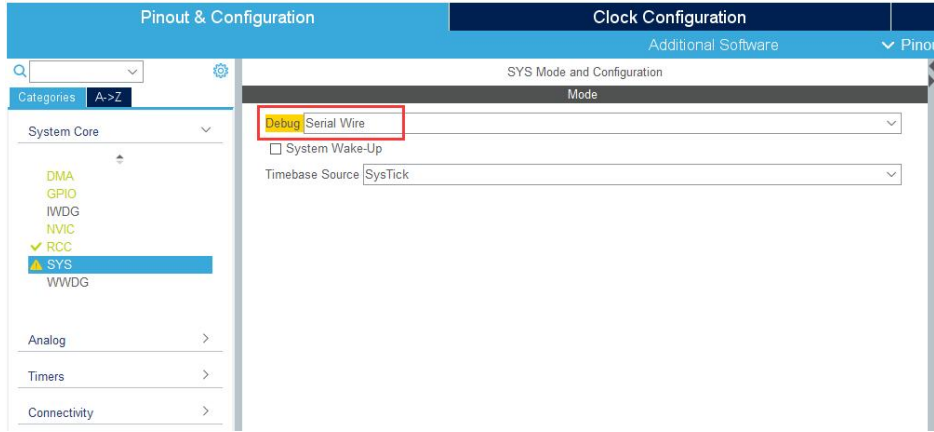


图 1-19 Debug 引脚配置

4) 芯片引脚配置全局图如图 1-20

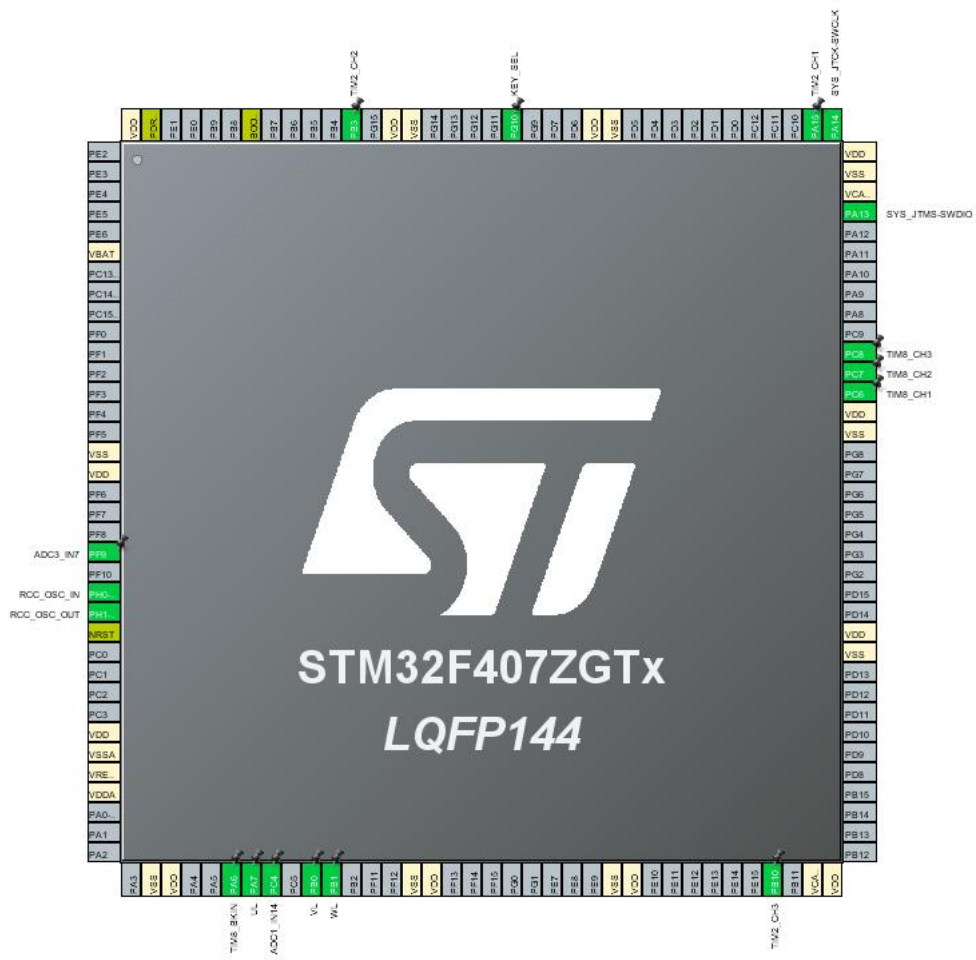


图 1-20 芯片引脚配置

5) GPIO 配置如图 1-21





图 1-21 GPIO 引脚配置

6) ADC3 配置如图 1-22~1-24.

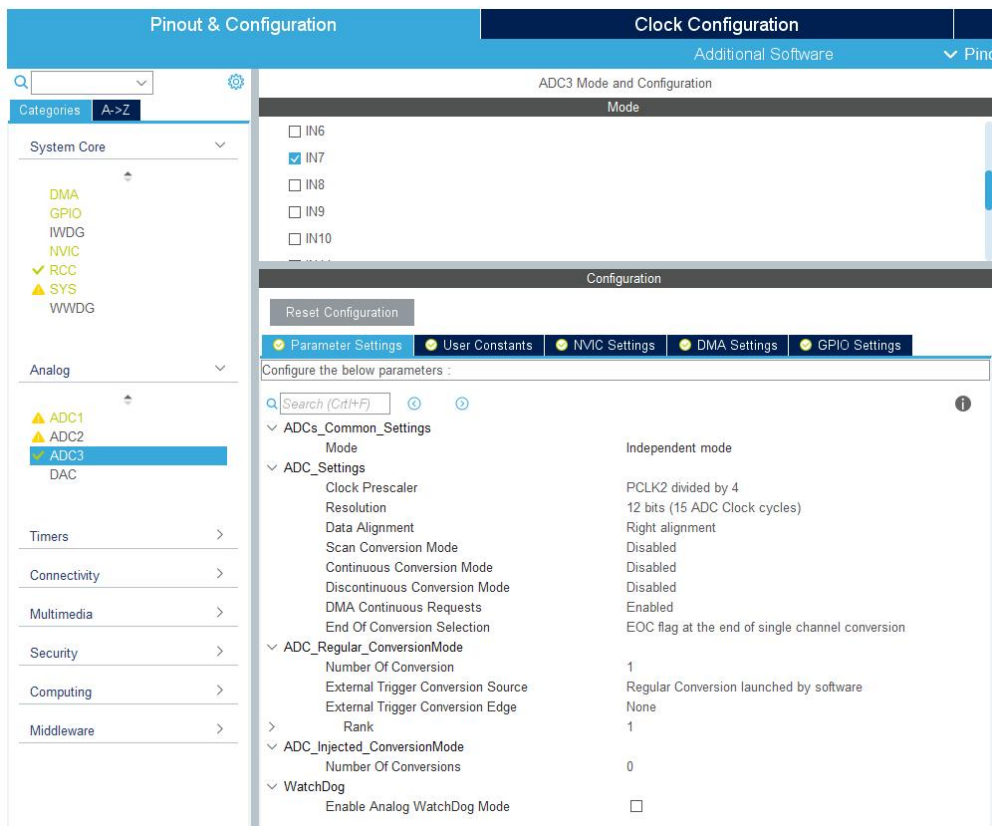


图 1-22 ADC3 配置 1

Pinout & Configuration | Clock Configuration | Additional Software | Pinout

ADC3 Mode and Configuration

Mode

- IN6
- IN7
- IN8
- IN9
- IN10
- IN11
- IN12
- IN13
- IN14

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | **DMA Settings** | GPIO Settings

DMA Request	Stream	Direction	Priority
ADC3	DMA2 Stream 1	Peripheral To Memory	High

Add | Delete

DMA Request Settings

Mode	Circular	Increment Address	<input type="checkbox"/>	Peripheral	<input type="checkbox"/>	Memory	<input checked="" type="checkbox"/>
Use Fifo	<input type="checkbox"/>	Threshold		Data Width	Half Word	Half Word	
				Burst Size			

图 1-23 ADC3 配置 2

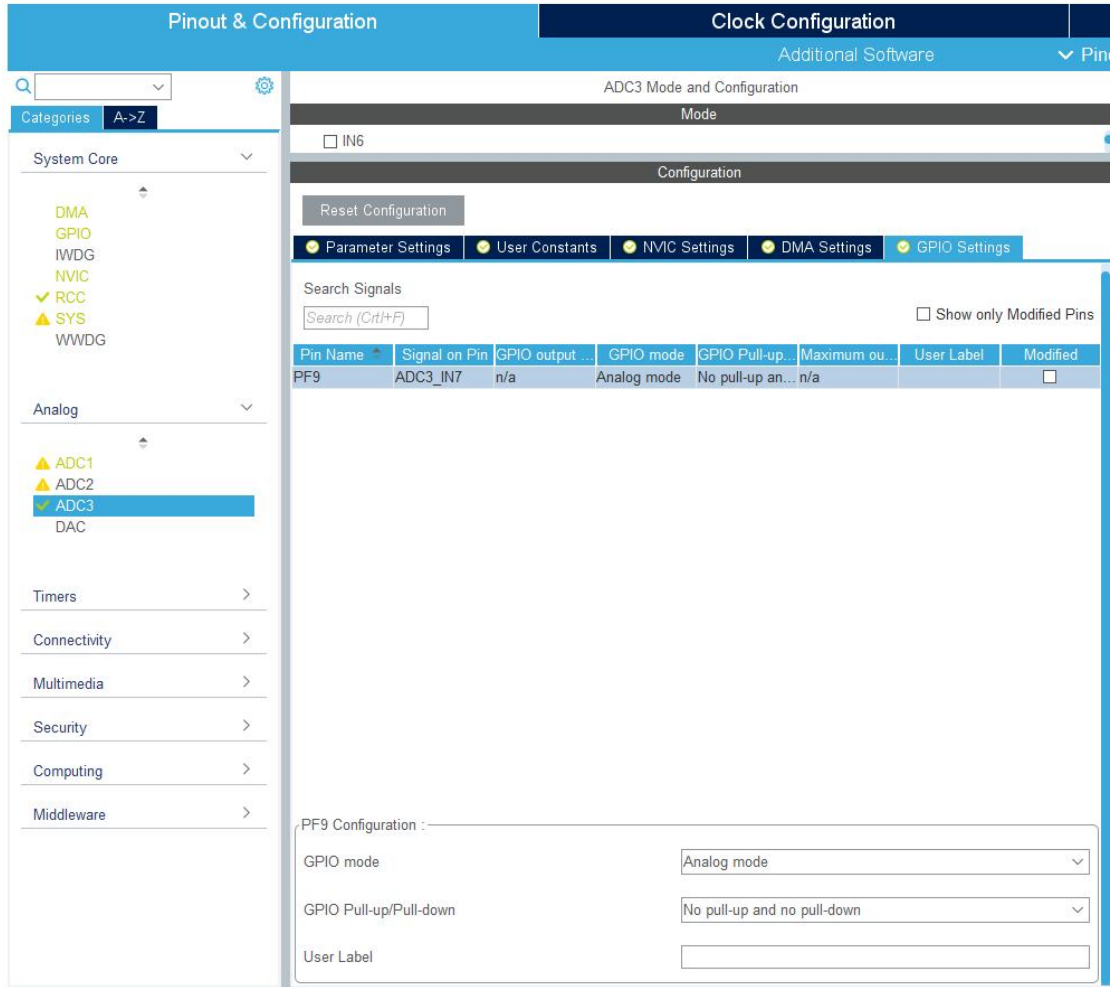


图 1-24 ADC3 配置 3

7) ADC1 配置如图 1-25~1-27。

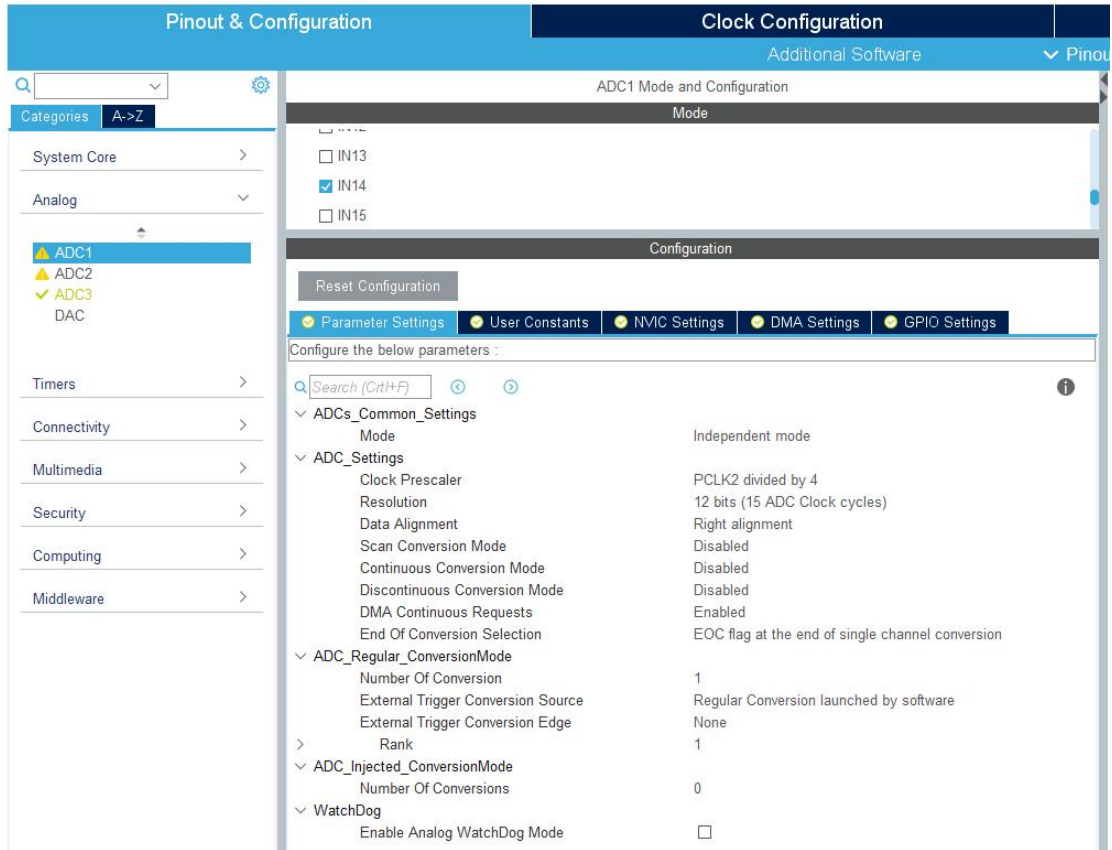


图 1-25 ADC1 配置 1

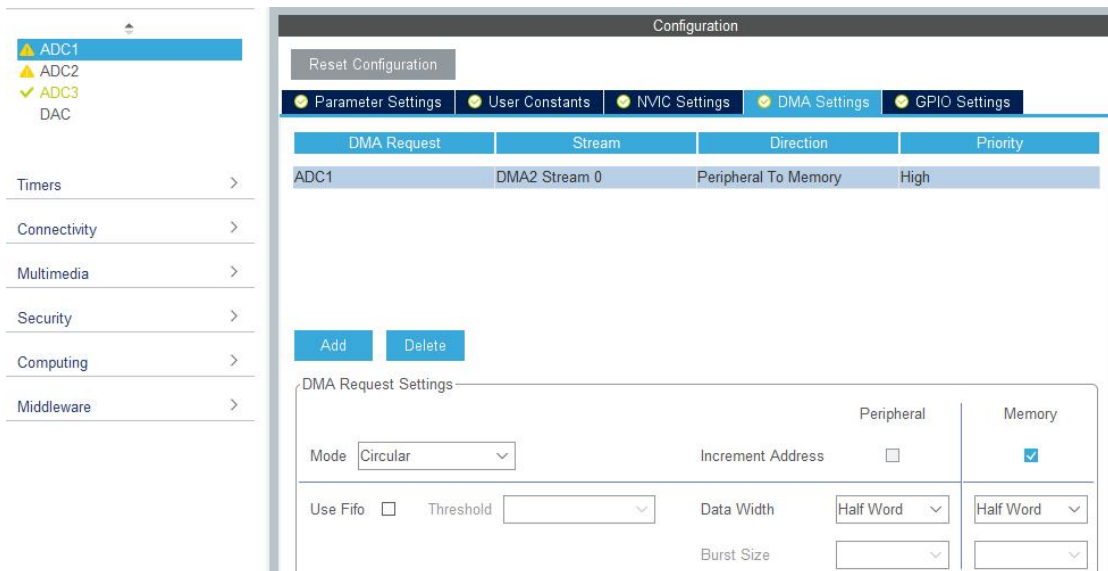


图 1-26 ADC1 配置 2

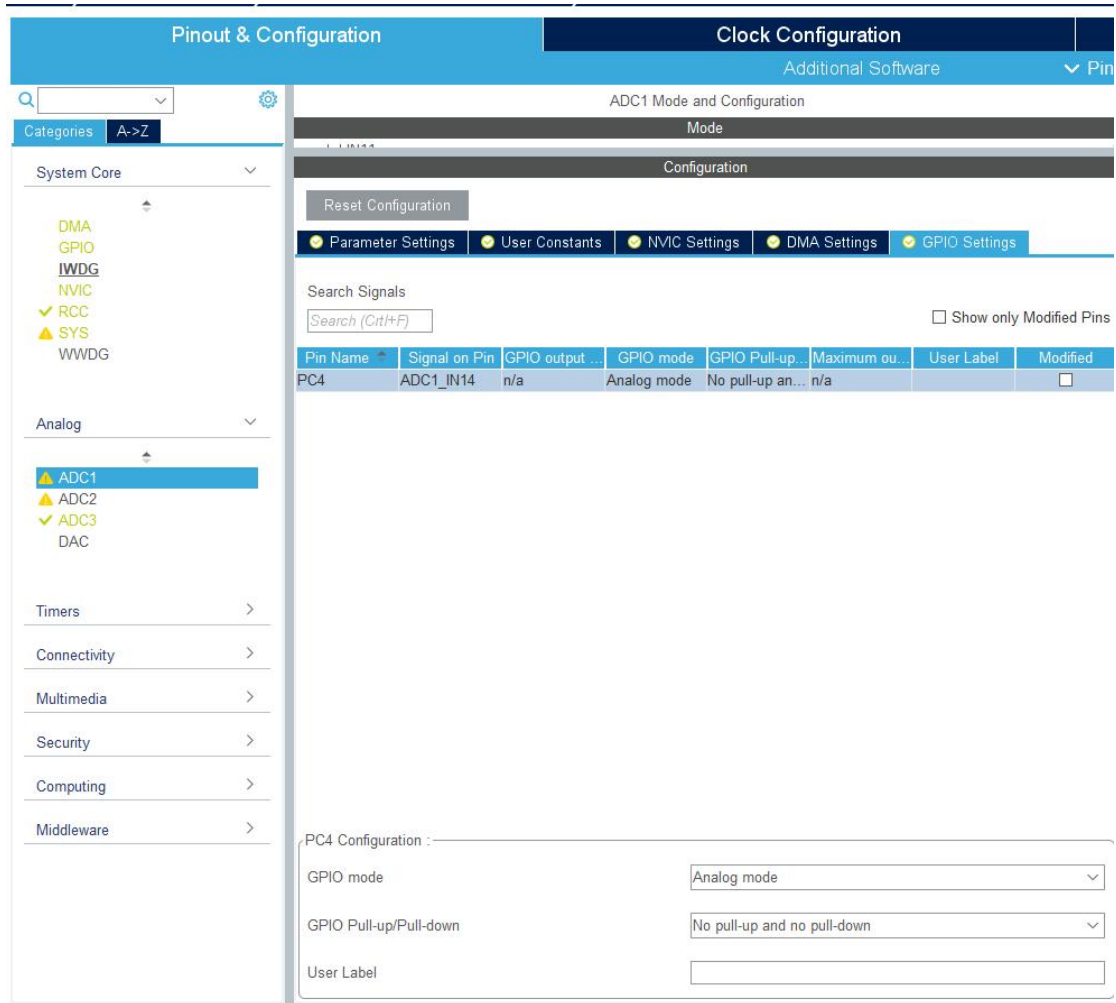


图 1-27 ADC1 配置 3

8) TIM2 配置如图 1-28~1-31。

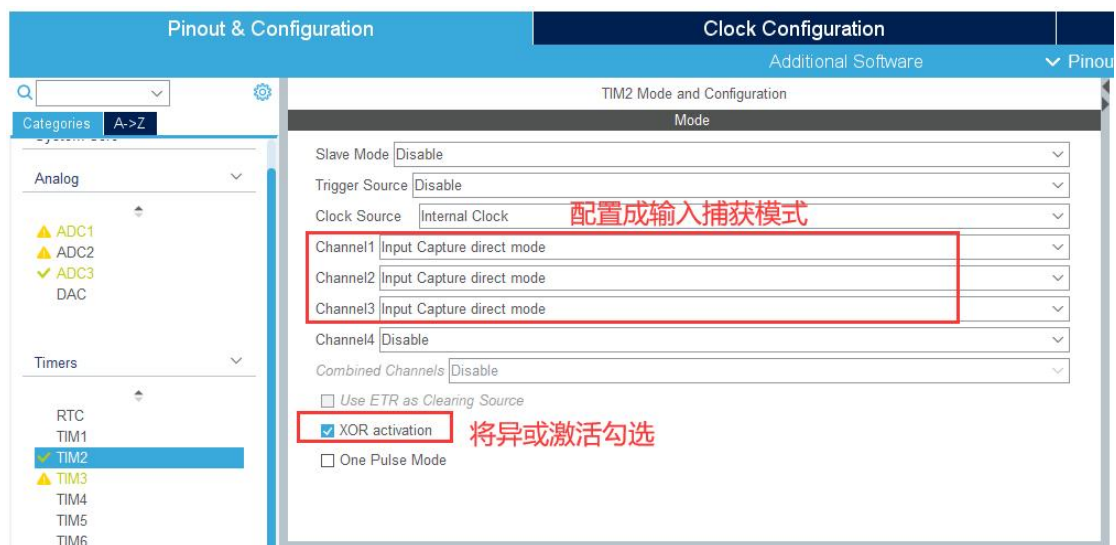


图 1-28 TIM2 配置 1

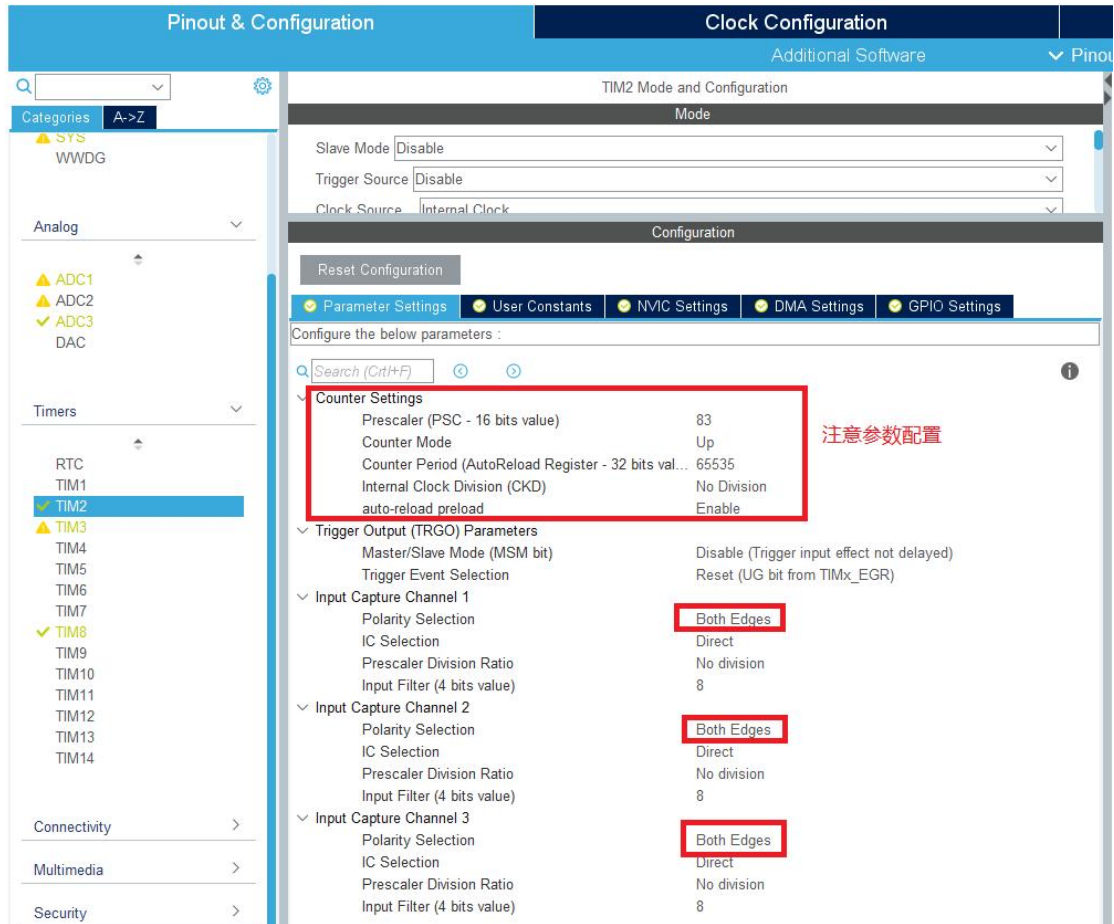


图 1-29 TIM2 配置 2

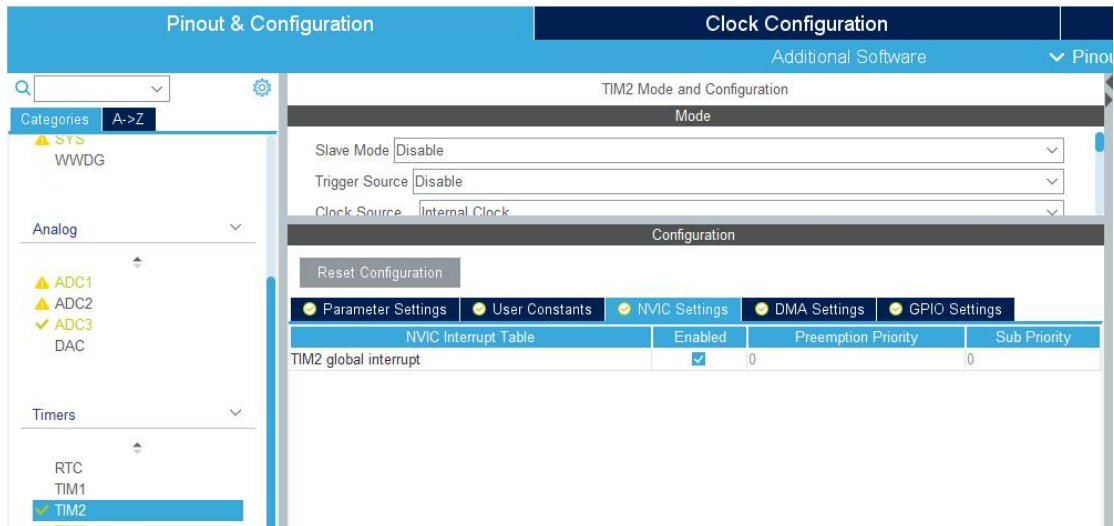


图 1-30 TIM2 配置 3

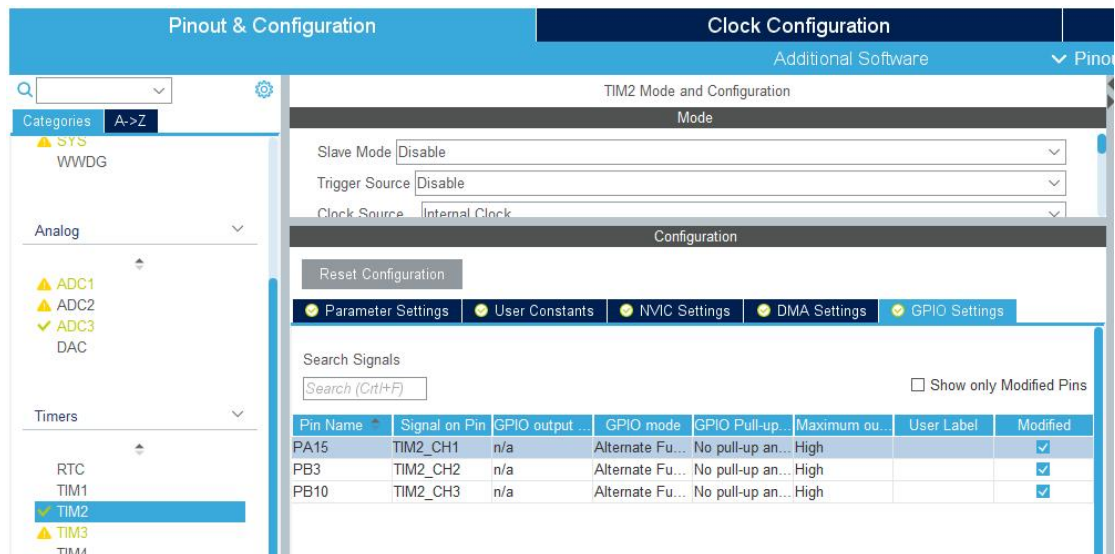


图 1-31 TIM2 配置 4

9) TIM8 配置如图 1-32~1-36。除了生成 PWM，TIM8 还使能了 BKIN，可以使电机在紧急情况下制动。

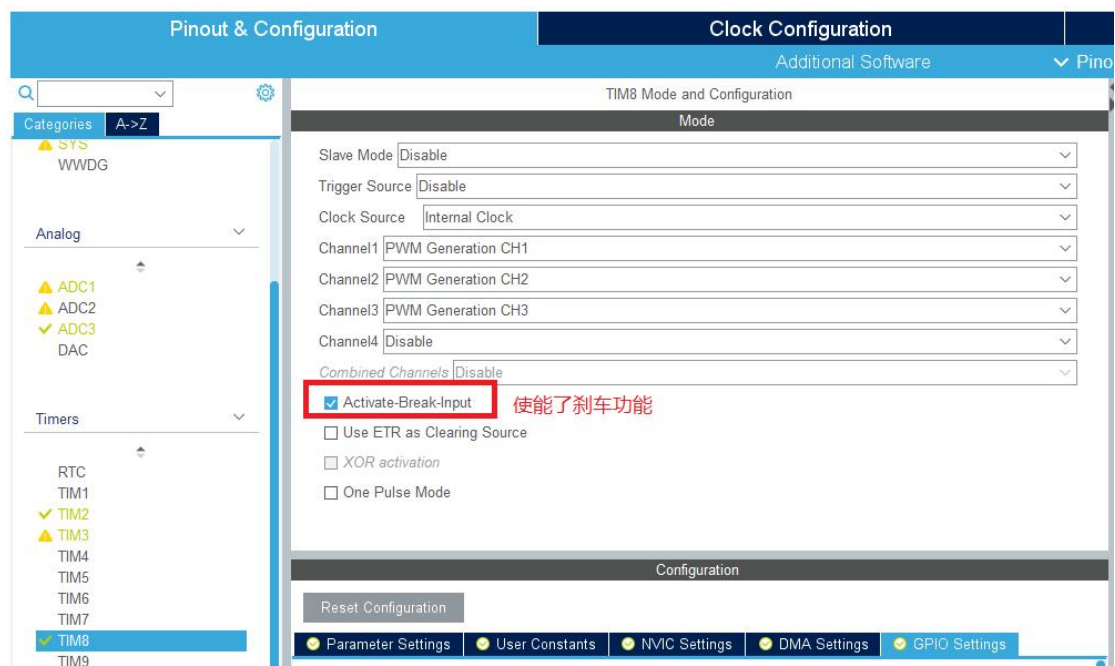


图 1-32 TIM8 配置 1

Pinout & Configuration | Clock Configuration | Additional Software

Categories A->Z

- ▲ SYS
- WWDG
- Analog
  - ▲ ADC1
  - ▲ ADC2
  - ▼ ADC3
  - DAC
- Timers
  - RTC
  - TIM1
  - ▼ TIM2
  - ▲ TIM3
  - TIM4
  - TIM5
  - TIM6
  - TIM7
  - ▼ TIM8
  - TIM9
  - TIM10
  - TIM11
  - TIM12
  - TIM13
  - TIM14
- Connectivity >

TIM8 Mode and Configuration

Mode

Slave Mode: Disable

Trigger Source: Disable

Clock Source: Internal Clock

Channel1: PWM Generation CH1

Channel2: PWM Generation CH2

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings	
Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits val...)	8400
Internal Clock Division (CKD)	No Division
Repetition Counter (RCR - 8 bits value)	0
auto-reload preload	Enable

PWM载波频率不超过 20KHz

Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Break And Dead Time management - BRK Configuration	
BRK State	Enable
BRK Polarity	Low

Break And Dead Time management - Output Configuration	
Automatic Output State	Disable
Off State Selection for Run Mode (OSSR)	Enable
Off State Selection for Idle Mode (OSSl)	Enable
Lock Configuration	Off

注意使能状态

图 1-33 TIM8 配置 2



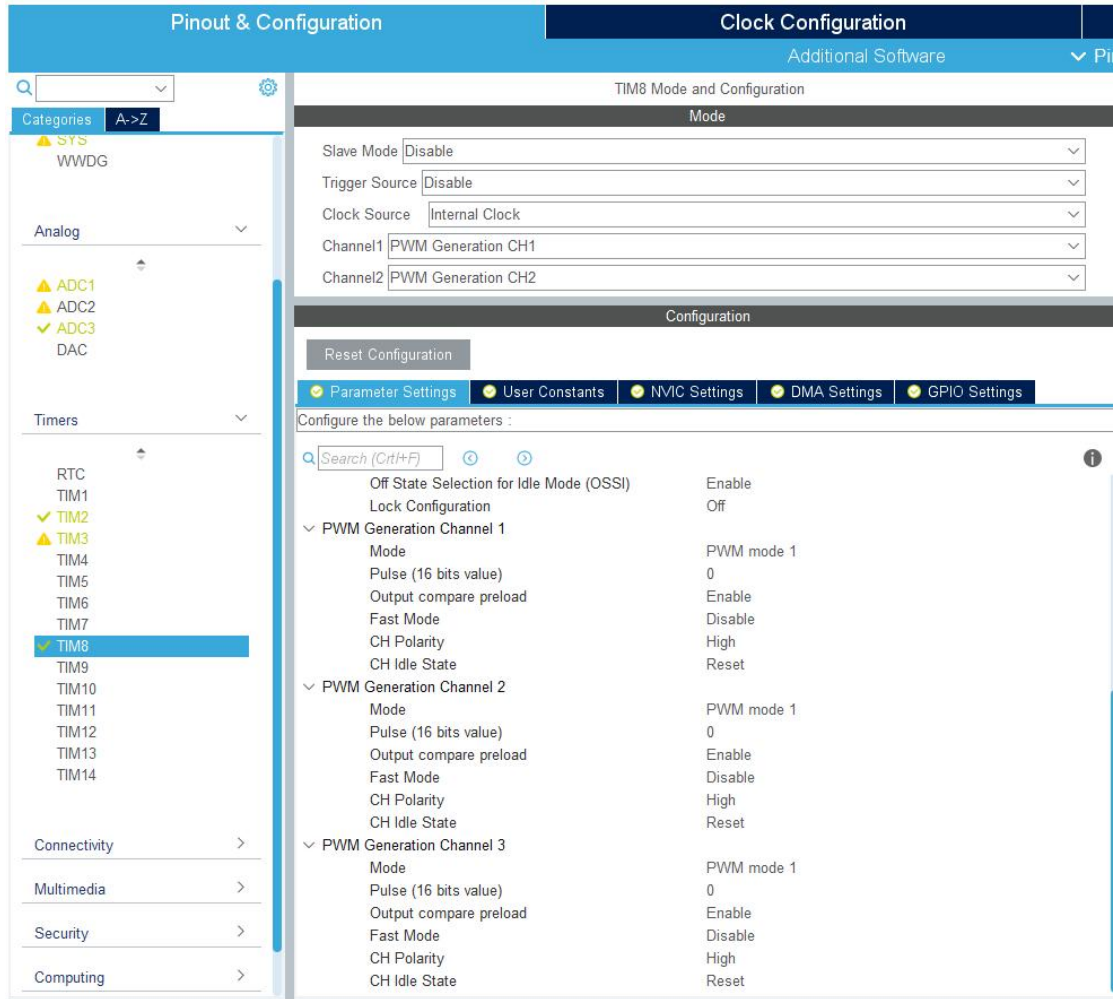


图 1-34 TIM8 配置 3

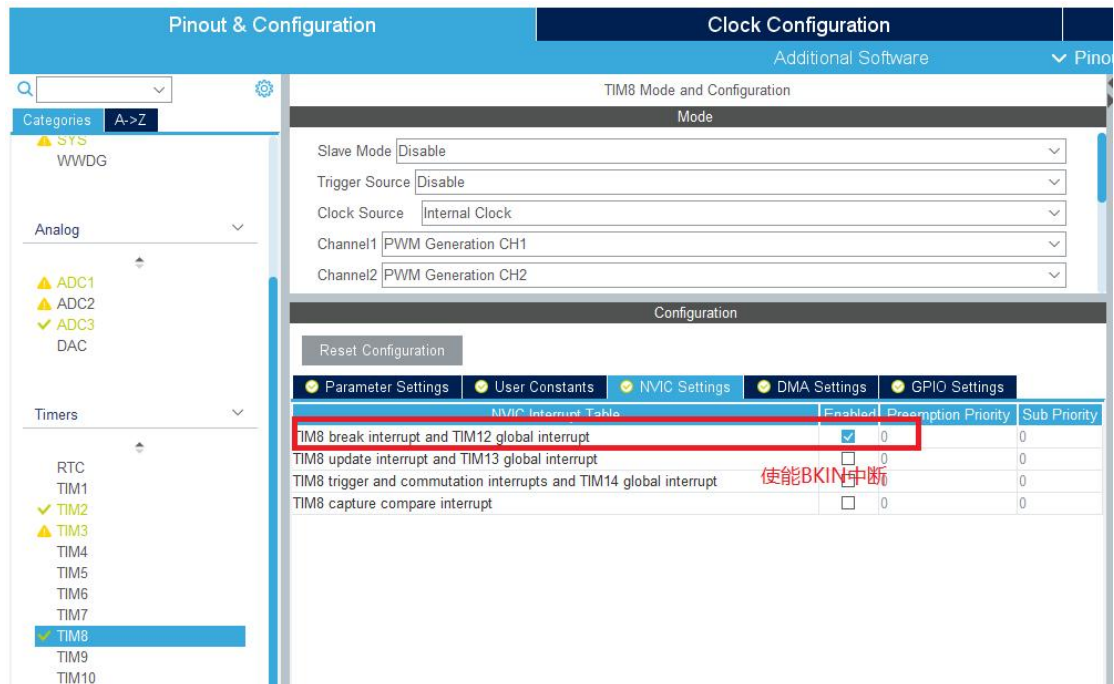


图 1-35 TIM8 配置 4

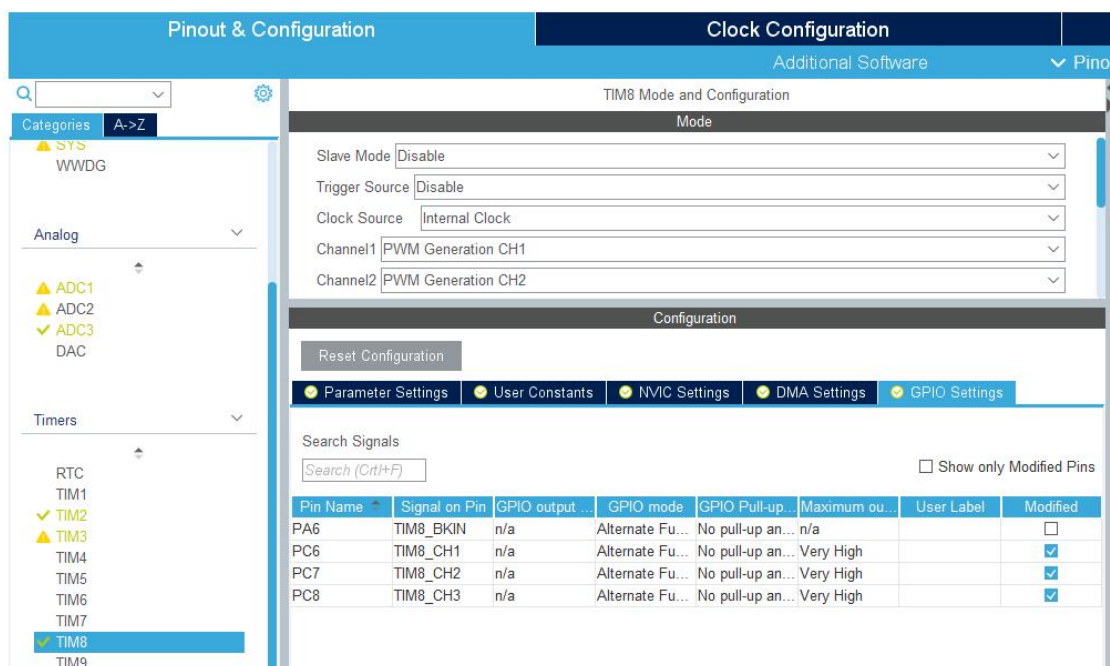


图 1-36 TIM8 配置 5

- 10) TIM3 定时器配置请同学们自行完成。
- 11) 配置工程文件，并生成相关代码，参考前述实验。

## 1.7.2 程序编写与调试

**(这部分需要同学们自己完成)**

编程提示：

- 1) 检测PA15、PB3、PB10三个引脚的电平，确定当前霍尔编码。伪代码如下，

```

if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_10)==GPIO_PIN_SET)
    hall_status =...
if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_3)==GPIO_PIN_SET)
    hall_status =...
if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_15)==GPIO_PIN_SET)
    hall_status =...
    
```

- 2) 确定霍尔编码后，**严格按照**霍尔换相表进行换相。**切勿将A+、A-、B+、B-、C+、C-同时开启，这些情况相当于电源正负极直接短路，结果就是电源必烧无疑!!!** 正转伪代码如下，

```

if(hall_a=1 & hall_b=0 & hall_c=1)
{
    TIM8_CH1_ON;  TIM8_CH2_OFF; TIM8_CH3_OFF;
    GPIO_OUT1_OFF;  GPIO_OUT2_ON; GPIO_OUT3_OFF;
}
    
```

```

}
if(hall_a=1 & hall_b=0 & hall_c=0)
{
    TIM8_CH1_ON;  TIM8_CH2_OFF; TIM8_CH3_OFF;
    GPIO_OUT1_OFF;  GPIO_OUT2_OFF; GPIO_OUT3_ON;
}
if(hall_a=1 & hall_b=1 & hall_c=0)
{
    TIM8_CH1_OFF;  TIM8_CH2_ON; TIM8_CH3_OFF;
    GPIO_OUT1_OFF;  GPIO_OUT2_OFF; GPIO_OUT3_ON;
}
if(hall_a=0 & hall_b=1 & hall_c=0)
{
    TIM8_CH1_OFF;  TIM8_CH2_ON; TIM8_CH3_OFF;
    GPIO_OUT1_ON;   GPIO_OUT2_OFF; GPIO_OUT3_OFF;
}
if(hall_a=0 & hall_b=1 & hall_c=1)
{
    TIM8_CH1_OFF;  TIM8_CH2_OFF; TIM8_CH3_ON;
    GPIO_OUT1_ON;   GPIO_OUT2_OFF; GPIO_OUT3_OFF;
}
if(hall_a=0 & hall_b=0 & hall_c=1)
{
    TIM8_CH1_OFF;  TIM8_CH2_OFF; TIM8_CH3_ON;
    GPIO_OUT1_OFF;  GPIO_OUT2_ON; GPIO_OUT3_OFF;
}
}

```

- 3) **务必添加紧急制动功能！** BKIN中断在电机三相电路电流过大时（比如A+、A-同时加电导致短路的情况）产生，检测到中断后应当立即停止电机转动，将A+、A-、B+、B-、C+、C-断开，可以保护电机，紧急刹车。伪代码如下，在BKIN中断函数中实现。

```

void HAL_TIMEx_BreakCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance==TIM8)
    {
        TIM8_CH1_OFF;  TIM8_CH2_OFF; TIM8_CH3_OFF;
        GPIO_OUT1_OFF;  GPIO_OUT2_OFF; GPIO_OUT3_OFF;
        __HAL_TIM_DISABLE_IT(&htim8, TIM_IT_BREAK);
    }
}

```

- 4) 根据BLDC电机结构，想要电机转起来必须要给定子绕组通电，产生磁场。因此，在电机启动时，需要调用TIM2捕获中断，给绕组换相加电。通常需要调用两次，第一次是初始的输出配置。延迟一段时间，是为了确保TIM8的输出配置已经生效。此时电机可能已经开始转动了，需要再

调用一次确保输出是对的，参考代码如下，

```
HAL_TIM_IC_CaptureCallback(&htim2);
HAL_Delay(4);
HAL_TIM_IC_CaptureCallback(&htim2);
```

5) 编程中可能用到的函数列表，

表 1-4 常用函数列表

函数名称	功能
__HAL_TIM_ENABLE_IT(&htim8, TIM_IT_BREAK)	使能定时器8的BKIN中断，CUBE生成的初始化并没有开启中断，所以调用一次函数。
HAL_TIM_IC_Start_IT(TIM_HandleTypeDef *htim, uint32_t Channel)	启动定时器输入捕获中断函数
HAL_TIMEx_BreakCallback(TIM_HandleTypeDef *htim)	高级定时器BKIN中断函数
HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)	定时器输入捕获中断函数

## 1.8 实验注意事项

- 1) **重要的事情说三遍!!!** 切勿将A+、A-、B+、B-、C+、C-同时开启，这些情况相当于电源正负极直接短路，结果就是电源必烧无疑！
- 2) 添加紧急制动功能，可以再绕组电流过大时急停，保护电机。
- 3) 在六步换相控制逻辑编程过程中，**要慎用单步调试!** 单步调试时，若电机固定在一个位置，三相绕组加电方向固定不变，这种情况是很危险的，绕组都是漆包线铜丝，电阻非常小，**当总是给电时候，电路中电流就非常大，严重情况，烧毁电机或者电源!**
- 4) 电机启动后，**要注意观察实验箱的电流表的示数**，若出现迅速增加的情况，说明出现了短路，应当**立即切断电源!**



- 5) 在电机运行时，**不要用手随意触摸功率板**，以免发生短路或者静电损害。

## 1.9 实验报告要求

- 1) 画出程序流程图，尽量详细的描述程序设计。
- 2) 将自己编写的代码粘贴到实验报告中，写好代码注释。
- 3) 将不同 PWM 载波频率下的反电动势和 HALL 信号波形记录在实验报告中，并说说 PWM 载波频率对电机转动的影响。
- 4) 阐述一下自己在开发过程中遇到的主要问题，及最终解决方法。

## 第 2 章 BLDC 方波速度闭环控制实验

### 2.1 实验目的

- 1) 掌握利用无刷直流电机HALL传感器信号换相的方法。
- 2) 熟悉AD转换器模拟信号的编程及使用。
- 3) 掌握无刷直流电机转速PID闭环的设计、调试方法。

### 2.2 预习要求

- 1) 回顾六步换相法开环控制BLDC电机原理和方法
- 2) 熟悉PID参数的调试、整定的方法和步骤。

### 2.3 实验设备

- 1) 数字式电机调速实验箱 1台
- 2) 计算机 1台
- 3) 示波器 1台
- 4) 万用表 1个

### 2.4 实验原理

自动控制系统可分为开环控制系统和闭环控制系统。前面已经介绍了 BLDC 的旋转驱动，霍尔传感器信号不仅可以检测当前转子位置信息，我们还可以利用它来检测电机旋转速度，而有了速度反馈信号我们就可以使用 PID 算法来实现控速。

将偏差的比例（Proportion）、积分（Integral）和微分（Differential）通过线性组合构成控制量，用这一控制量对被控对象进行控制，这样的控制器称 PID 控制器。

#### 2.4.1 模拟 PID 控制原理

在模拟控制系统中，控制器最常用的控制规律是 PID 控制。为了说明控制器的工作原理，先看一个例子。如图所示是一个小功率直流电机的调速原理图。给定速度 $n_0(t)$ 与实际转速 $n(t)$ 进行比较，其差值 $e(t) = n_0(t) - n(t)$ ，经过 PID

控制器调整后输出电压控制信号  $u(t)$ ， $u(t)$  经过功率放大后，驱动直流电动机改变其转速。



图 2-1 小功率直流电机调速系统

常规的模拟 PID 控制系统原理框图如图所示。该系统由模拟 PID 控制器和被控对象组成。图中， $r(t)$  是给定值， $y(t)$  是系统的实际输出值，给定值与实际输出值构成控制偏差  $e(t)$ ，所以  $e(t) = r(t) - y(t)$ 。

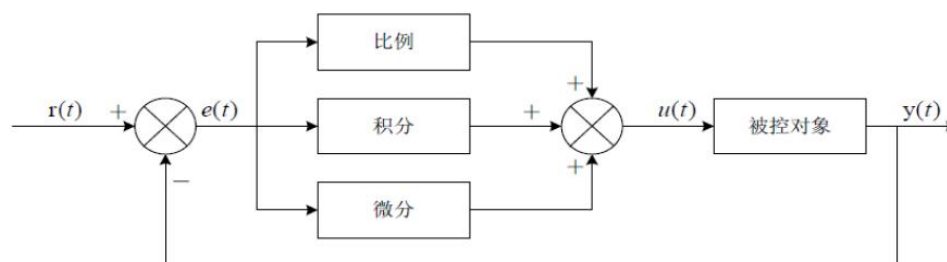


图 2-2 模拟 PID 控制系统原理图

$e(t)$  作为 PID 控制的输入， $u(t)$  作为 PID 控制器的输出和被控对象的输入。所以模拟 PID 控制器的控制规律为：

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (\text{式 2-1})$$

其中： $K_p$  为控制器的比例系数； $T_i$  为控制器的积分时间，也称积分系数； $T_d$  为控制器的微分时间，也称微分系数。

比例部分的数学式表示是：

$$K_p * e(t)。$$

在模拟 PID 控制器中，比例环节的作用是对偏差瞬间作出反应。偏差一旦产生控制器立即产生控制作用，使控制量向减少偏差的方向变化。控制作用的强弱取决于比例系数，比例系数越大，控制作用越强，则过渡过程越快，控制过程的静态偏差也就越小；但是越大，也越容易产生振荡，破坏系统的稳定性。故而，比例系数选择必须恰当，才能过渡时间少，静差小而又稳定的效果。

积分部分的数学式表示是：

$$\frac{K_p}{T_i} \int_0^t e(t) dt$$

从积分部分的数学表达式可以知道，只要存在偏差，则它的控制作用就不断的增加；只有在偏差为零时，它的积分才能是一个常数，控制作用才是一个不会增加的常数。可见，积分部分可以消除系统的偏差。积分环节的调节作用虽然会消除静态误差，但也会降低系统的响应速度，增加系统的超调量。积分常数越

大，积分的积累作用越弱，这时系统在过渡时不会产生振荡；但是增大积分常数会减慢静态误差的消除过程，消除偏差所需的时间也较长，但可以减少超调量，提高系统的稳定性。当积分常数较小时，则积分的作用较强，这时系统过渡时间中有可能产生振荡，不过消除偏差所需的时间较短。所以必须根据实际控制的具体要求来确定积分系统。

微分部分的数学式表示是：

$$K_p * T_d \frac{de(t)}{dt}$$

实际的控制系统除了希望消除静态误差外，还要求加快调节过程。在偏差出现的瞬间，或在偏差变化的瞬间，不但要对偏差量做出立即响应（比例环节的作用），而且要根据偏差的变化趋势预先给出适当的纠正。为了实现这一作用，可在PI控制器的基础上加入微分环节，形成PID控制器。微分环节的作用是阻止偏差的变化。它是根据偏差的变化趋势（变化速度）进行控制。偏差变化的越快，微分控制器的输出就越大，并能在偏差值变大之前进行修正。微分作用的引入，将有助于减小超调量，克服振荡，使系统趋于稳定，特别对高阶系统非常有利，它加快了系统的跟踪速度。但微分的作用对输入信号的噪声很敏感，对那些噪声较大的系统一般不用微分，或在微分起作用之前先对输入信号进行滤波。微分部分的作用由微分时间常数决定。微分系统越大时，则它抑制偏差变化的作用越强；越小时，则它反抗偏差变化的作用越弱。微分部分显然对系统稳定有很大的作用。适当地选择微分常数，可以使微分作用达到最优。由于计算机的出现，计算机进入了控制领域。人们将模拟PID控制规律引入到计算机中来。对式2-1的PID控制规律进行适当的变换，就可以用软件实现PID控制，即数字PID控制。数字式PID控制算法可以分为位置式PID和增量式PID控制算法。

## 2.4.2 位置式PID算法

由于计算机控制是一种采样控制，它只能根据采样时刻的偏差计算控制量，而不能像模拟控制那样连续输出控制量，进行连续控制。由于这一特点，式2-1中的积分项和微分项不能直接使用，必须进行离散化处理。离散化处理的方法为：以T作为采样周期，k作为采样序号，则离散采样时间kT对应着连续时间t，用矩形法数值积分近似代替积分，用一阶后向差分近似代替微分，可作如下近似变换：

$$\left. \begin{aligned} t &\approx kT \\ \int_0^t e(t)dt &\approx T \sum_{j=0}^k e(jT) = T \sum_{j=0}^k e_j \\ \frac{de(t)}{dt} &\approx \frac{e(kT) - e[(k-1)T]}{T} = \frac{e_k - e_{k-1}}{T} \end{aligned} \right\} \quad (\text{式 2-2})$$

上式中，为了表示的方便，将类似于e(kT)简化成e(k)等。



将式 2-2 代入式 2-1, 就可以得到离散的 PID 表达式为:

$$u_k = Kp[e_k + \frac{T}{Ti} \sum_{j=0}^k e_j + Td \frac{e_k - e_{k-1}}{T}] \quad (\text{式 2-3})$$

$$u_k = Kp * e_k + Ki \sum_{j=0}^k e_j + Kd(e_k - e_{k-1}) \quad (\text{式 2-4})$$

其中,  $k$  为采样序号,  $k = 0, 1, 2, \dots$ ;

$u_k$  为第  $k$  次采样时刻的计算机输出值;

$e_k$  为第  $k$  次采样时刻输入的偏差值;

$e_{k-1}$  为第  $k-1$  次采样时刻输入的偏差值;

$Ki$  为积分系数,  $Ki = Kp * T / Ti$ ;

$Kd$  为微分系数,  $Kd = Kp * Td / T$ ;

如果采样周期足够小, 则式 2-3 或式 2-4 的近似计算可以获得足够精确的结果, 离散控制过程与连续过程十分接近。

式 2-3 或式 2-4 表示的控制算法直接按式 2-1 所给出的 PID 控制规律定义进行计算的, 所以它给出了全部控制量的大小, 因此被称为全量式或位置式 PID 控制算法。

这种算法的缺点是: 由于全量输出, 所以每次输出均与过去状态有关, 计算时要对  $e_k$  进行累加, 工作量大; 并且, 因为计算机输出的  $u_k$  对应的是执行机构的实际位置, 如果计算机出现故障, 输出的  $u_k$  将大幅度变化, 会引起执行机构的大幅度变化, 有可能因此造成严重的生产事故, 这在实生产中是不允许的。

增量式 PID 控制算法可以避免这种现象发生。

### 2.4.3 增量式 PID 算法

所谓增量式 PID 是指数字控制器的输出只是控制量的增量  $\Delta u_k$ 。当执行机构需要的控制量是增量, 而不是位置量的绝对数值时, 可以使用增量式 PID 控制算法进行控制。增量式 PID 控制算法可以通过式 2-3 推导出。由式 2-3 可以得到控制器的第  $k-1$  个采样时刻的输出值为:

$$u_{k-1} = Kp[e_{k-1} + \frac{T}{Ti} \sum_{j=0}^{k-1} e_j + Td \frac{e_{k-1} - e_{k-2}}{T}] \quad (\text{式 6-6})$$

(式 2-5)

可以得到增量式 PID 控制算法公式为:

$$\begin{aligned} \Delta u &= u_k - u_{k-1} = Kp[e_k - e_{k-1} + \frac{T}{Ti} e_k + Td \frac{e_k - 2e_{k-1} + e_{k-2}}{T}] \\ &= Kp(1 + \frac{T}{Ti} + \frac{Td}{T})e_k - Kp(1 + \frac{2Td}{T})e_{k-1} + Kp \frac{Td}{T} e_{k-2} \\ &= Ae_k + Be_{k-1} + Ce_{k-2} \end{aligned} \quad (\text{式 2-6})$$

其中，

$$A = K_p \left( 1 + \frac{T}{T_i} + \frac{T_d}{T} \right);$$

$$B = K_p \left( 1 + \frac{2T_d}{T} \right);$$

$$C = K_p \frac{T_d}{T};$$

由式 2-6 可以看出，如果计算机控制系统采用恒定的采样周期  $T$ ，一旦确定  $A$ 、 $B$ 、 $C$ ，只要使用前后三次测量的偏差值，就可以由 6-7 求出控制量。

增量式 PID 控制算法与位置式 PID 算法(如式 2-3) 相比，计算量小的多，因此在实际中得到广泛的应用。

而位置式 PID 控制算法也可以通过增量式控制算法推出递推计算公式：

$$u_k = u_{k-1} + \Delta u_k \quad (\text{式 2-7})$$

## 2.4.4 控制器参数整定

**控制器参数整定：**指决定调节器的比例系数  $K_p$ 、积分时间  $T_i$ 、微分时间  $T_d$  和采样周期  $T_s$  的具体数值。整定的实质是通过改变调节器的参数，使其特性和过程特性相匹配，以改善系统的动态和静态指标，取得最佳的控制效果。

整定调节器参数的方法很多，归纳起来可分为两大类，即理论计算整定法和工程整定法。理论计算整定法有对数频率特性法和根轨迹法等；工程整定法有凑试法、临界比例法、经验法、衰减曲线法和响应曲线法等。工程整定法特点不需要事先知道过程的数学模型，直接在过程控制系统中进行现场整定，方法简单、计算简便、易于掌握。

### 1) 试凑法

按照先比例 (P)、再积分 (I)、最后微分 (D) 的顺序。

置调节器积分时间  $T_i = \infty$ ，微分时间  $T_d = 0$ ，在比例系数按经验设置的初值条件下，将系统投入运行，由小到大整定比例系数  $K_p$ 。求得满意的 1/4 衰减度过渡过程曲线。

引入积分作用(此时应将上述比例系数  $K_p$  设置为  $5/6 K_p$ )。将  $T_i$  由大到小进行整定。

若需引入微分作用时，则将  $T_d$  按经验值或按  $T_d = (1/3 \sim 1/4) T_i$  设置，并由小到大加入。

### 2) 临界比例法

在闭环控制系统里，将调节器置于纯比例作用下，从小到大逐渐改变调节器的比例系数，得到等幅振荡的过渡过程。此时的比例系数称为临界比例系数  $K_u$ ，相邻两个波峰间的时间间隔，称为临界振荡周期  $T_u$ 。

临界比例度法步骤：

1、将调节器的积分时间  $T_i$  置于最大 ( $T_i = \infty$ )，微分时间置零 ( $T_d = 0$ )，比例系数  $K_p$  适当，平衡操作一段时间，把系统投入自动运行。

2、将比例系数  $K_p$  逐渐增大，得到等幅振荡过程，记下临界比例系数  $K_u$  和临界振荡周期  $T_u$  值。

3、根据  $K_u$  和  $T_u$  值，采用经验公式，计算出调节器各个参数，即  $K_p$ 、 $T_i$  和  $T_d$  的值。

按“先 P 再 I 最后 D”的操作程序将调节器整定参数调到计算值上。若还不够满意，可再作进一步调整。

临界比例度法整定注意事项：

有的过程控制系统，临界比例系数很大，使系统接近两式控制，调节阀不是全关就是全开，对工业生产不利。

有的过程控制系统，当调节器比例系数  $K_p$  调到最大刻度值时，系统仍不产生等幅振荡，对此，就把最大刻度的比例度作为临界比例度  $K_u$  进行调节器参数整定。

### 3) 经验法

用凑试法确定 PID 参数需要经过多次反复的实验，为了减少凑试次数，提高工作效率，可以借鉴他人的经验，并根据一定的要求，事先作少量的实验，以得到若干基准参数，然后按照经验公式，用这些基准参数导出 PID 控制参数，这就是经验法。

临界比例法就是一种经验法。这种方法首先将控制器选为纯比例控制器，并形成闭环，改变比例系数，使系统对阶跃输入的响应达到临界状态，这时记下比例系数  $K_u$ 、临界振荡周期为  $T_u$ ，根据 Z-N 提供的经验公式，就可以由这两个基准参数得到不同类型控制器的参数，如下表所示。

控制器类型	$K_p$	$T_i$	$T_d$
P	$0.5 K_u$		
PI	$0.45 K_u$	$0.85 T_u$	
PID	$0.6 K_u$	$0.5 T_u$	$0.12 T_u$

这种临界比例法是针对模拟 PID 控制器，对于数字 PID 控制器，只要采样周期取的较小，原则上也同样使用。在电动机的控制中，可以先采用临界比例法，然后在采用临界比例法求得结果的基础上，用凑试法进一步完善。

上表中的控制参数，实际上是按衰减度为 1/4 时得到的。通常认为 1/4 的衰减度能兼顾到稳定性和快速性。如果要求更大的衰减，则必须用凑试法对参数作进一步的调整。

### 4) 采样周期的选择

**Shannon 采样定律：**为不失真地复现信号的变化，采样频率至少应大于或等于连续信号最高频率分量的二倍。根据采样定律可以确定采样周期的上限值。实际采样周期的选择还要受到多方面因素的影响，不同的系统采样周期应根据具体情况来选择。

采样周期的选择，通常按照过程特性与干扰大小适当来选取采样周期：即对于响应快、（如流量、压力）波动大、易受干扰的过程，应选取较短的采样周期；反之，当过程响应慢（如温度、成份）、滞后大时，可选取较长的采样周期。

采样周期的选取应与 PID 参数的整定进行综合考虑，采样周期应远小于过程的扰动信号的周期，在执行器的响应速度比较慢时，过小的采样周期将失去意义，因此可适当选大一点；在计算机运算速度允许的条件下，采样周期短，则控制品质好；当过程的纯滞后时间较长时，一般选取采样周期为纯滞后时间的  $1/4 \sim 1/8$ 。

## 2.4.5 参数调整规则的探索

人们通过对 PID 控制理论的认识和长期人工操作经验的总结，可知 PID 参数应依据以下几点来适应系统的动态过程。

- 1) 在偏差比较大时，为使尽快消除偏差，提高响应速度，同时为了避免系统响应出现超调， $K_p$  取大值， $K_i$  取零；在偏差比较小时，为继续减小偏差，并防止超调过大、产生振荡、稳定性变坏， $K_p$  值要减小， $K_i$  取小值；在偏差很小时，为消除静差，克服超调，使系统尽快稳定， $K_p$  值继续减小， $K_i$  值不变或稍取大。
- 2) 当偏差与偏差变化率同号时，被控量是朝偏离既定值方向变化。因此，当被控量接近定值时，反号的比列作用阻碍积分作用，避免积分超调及随之而来的振荡，有利于控制；而当被控量远未接近各定值并向定值变化时，则由于这两项反向，将会减慢控制过程。在偏差比较大时，偏差变化率与偏差异号时， $K_p$  值取零或负值，以加快控制的动态过程。
- 3) 偏差变化率的大小表明偏差变化的速率， $e_k - e_{k-1}$  越大， $K_p$  取值越小， $K_i$  取值越大，反之亦然。同时，要结合偏差大小来考虑。
- 4) 微分作用可改善系统的动态特性，阻止偏差的变化，有助于减小超调量，消除振荡，缩短调节时间  $t_s$ ，允许加大  $K_p$ ，使系统稳态误差减小，提高控制精度，达到满意的控制效果。所以，在  $e_k$  比较大时， $K_d$  取零，实际为 PI 控制；在  $e_k$  比较小时， $K_d$  取一正值，实现 PID 控制。

## 2.5 实验内容

本实验重点是在实验十一的基础上增加 PID 的速度闭环功能。在 BLDC 开环实验完成的基础上，添加代码，实现闭环控制。主要内容和要求如下：

- 1) 采用位置式 PID 或增量式 PID 实现闭环速度控制功能。
- 2) 在 LCD 上实时显示设定转速、实测速度、电位器 ADC 采样数值、PWM 占空比、母线电压等参数。
- 3) 将设定转速、实测转速通过串口打印出来，在上位机接收数据，并将数据绘制成曲线。最终实现，能够在上位机实时显示设定速度与实测速度曲线的功能。
- 4) 调节 PID 参数与采样周期  $T_s$ ，观察不同参数配置下实际转速对设定转速的跟踪效果。记录不同参数配置下，曲线的变化。选择最好的一组曲线

对应的参数作为调参结果。

## 2.6 实验步骤

- 1) 完成 BLDC 开环控制，并经过课堂验收通过。
- 2) 在 BLDC 开环控制代码基础上，按照实验内容要求添加闭环控制功能。  
(同学们独立完成)
- 3) 添加串口打印与上位机显示功能。  
串口打印需要重定向 `Printf` 函数，具体实现方法，请大家参考“基础实验”。  
上位机显示可以有多种方式实现。可以用 `matlab` 实现或者下载带有曲线显示功能的串口调试助手实现。  
用 `matlab` 实现的可以参考“电机调速实验参考文件”中的“`MatLab_上位机.rar`”，压缩包提供了 `MatLab` 串口采集与波形显示的脚本文件。  
用串口显示软件的可以参考如下链接下载并使用。

[串口波形显示软件 SerialChart 的使用](#)

`SrialChart` 脚本文件参考如下，同学们可以根据需要在此基础上修改：

```
[_setup_]
port=COM5
baudrate=19200
width=1000
height=201
background_color = white
grid_h_origin = 0
grid_h_step = 10
grid_h_color = #EEE
grid_h_origin_color = #CCC
grid_v_origin = 0
grid_v_step = 10
grid_v_color = #EEE
grid_v_origin_color = #CCC
[_default_]
min=0
max=4095
[ch1]
color=red
[ch2]
color=green
```

- 4) 调节 `PID` 参数与采样周期 `Ts`，按照实验报告模板要求记录调参过程中的曲线，选择跟踪效果最好的一组曲线完成课堂验收。

## 2.7 实验报告要求

- 1) 画出闭环控制部分程序流程图，尽量详细的描述程序设计。
- 2) 将自己编写的代码粘贴到实验报告中，写好代码注释。
- 3) 调节 PID 参数与采样周期  $T_s$ ，记录不同参数配置下，曲线的变化。选择最好的一组曲线对应的参数作为调参结果。
- 4) 阐述一下自己在开发过程中遇到的主要问题，及最终解决方法。