

# *PAC52xx BLDC*

## *Ha11 方式电机调试*

**Power Application Controllers**

---

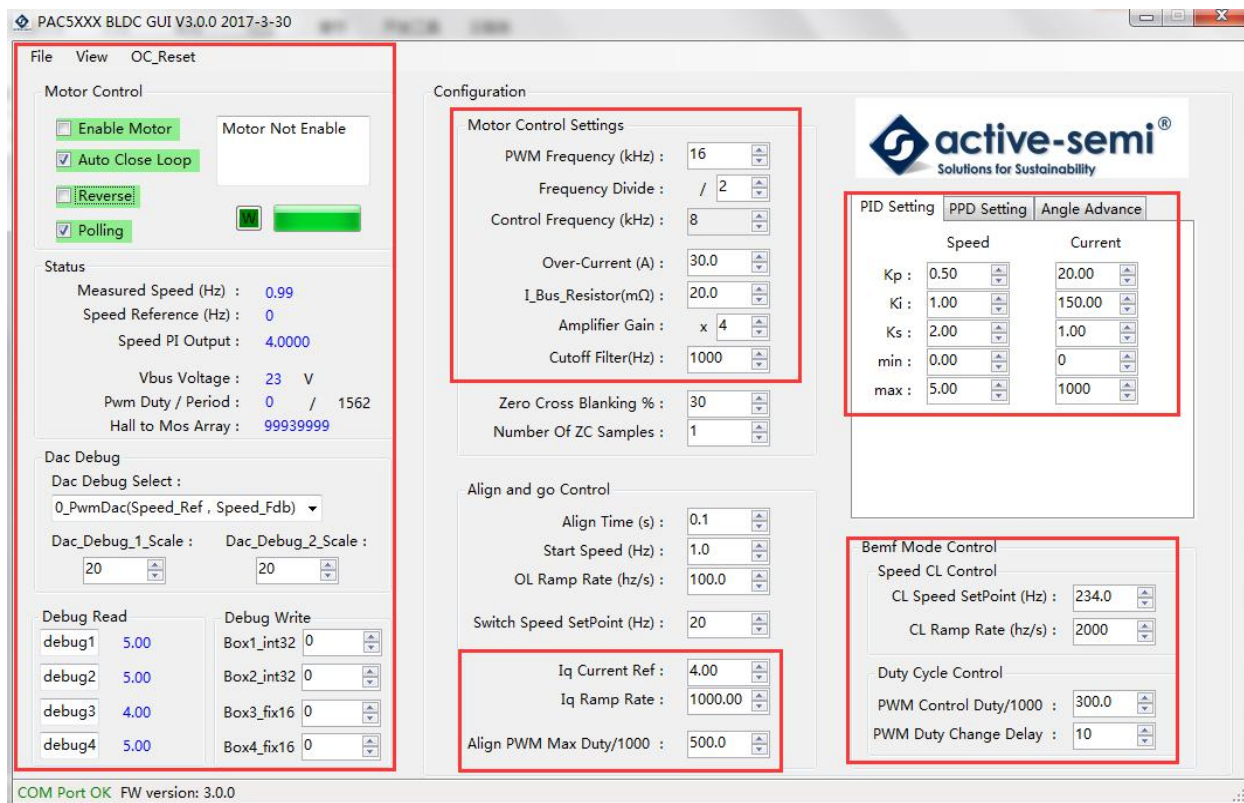
**目录**

1 HALL 模式运行电机.....	3
更改履历.....	8

# 1 HALL 模式运行电机

要用 hall 信号作为换相信号来跑电机，需先校准 hall 与 U/V/W 相序的关系，以下会讲解校准方法；

在 Hall 模式下，GUI 上有用的参数，如下用红色框标记，



在 Hall 模式下 PI 参数的调整与在 BEMF 模式下一样。

有些情况下，无法再 BEMF 模式下校准 hall，还可以观察反电动势波形与霍尔关系，来推导出 hall 值与开管关系的数组，方法如下：

做一个星型接法的电阻，三端接电机 U、V、W，然后外力转动电机，将 Hall 信号线和电源线接在 PCB 板上供电，外力带动电机旋转（此例为反转 reverse），测量 U、V、W 三相相电压（相对中心点），Hall 信号值，找一个参考量，把相电压波形和 Hall 信号波形放在同一时序下，如下图表，可以分析出 Hall 值对应的应该开管的状态，已知开管状态对应的编号，

```

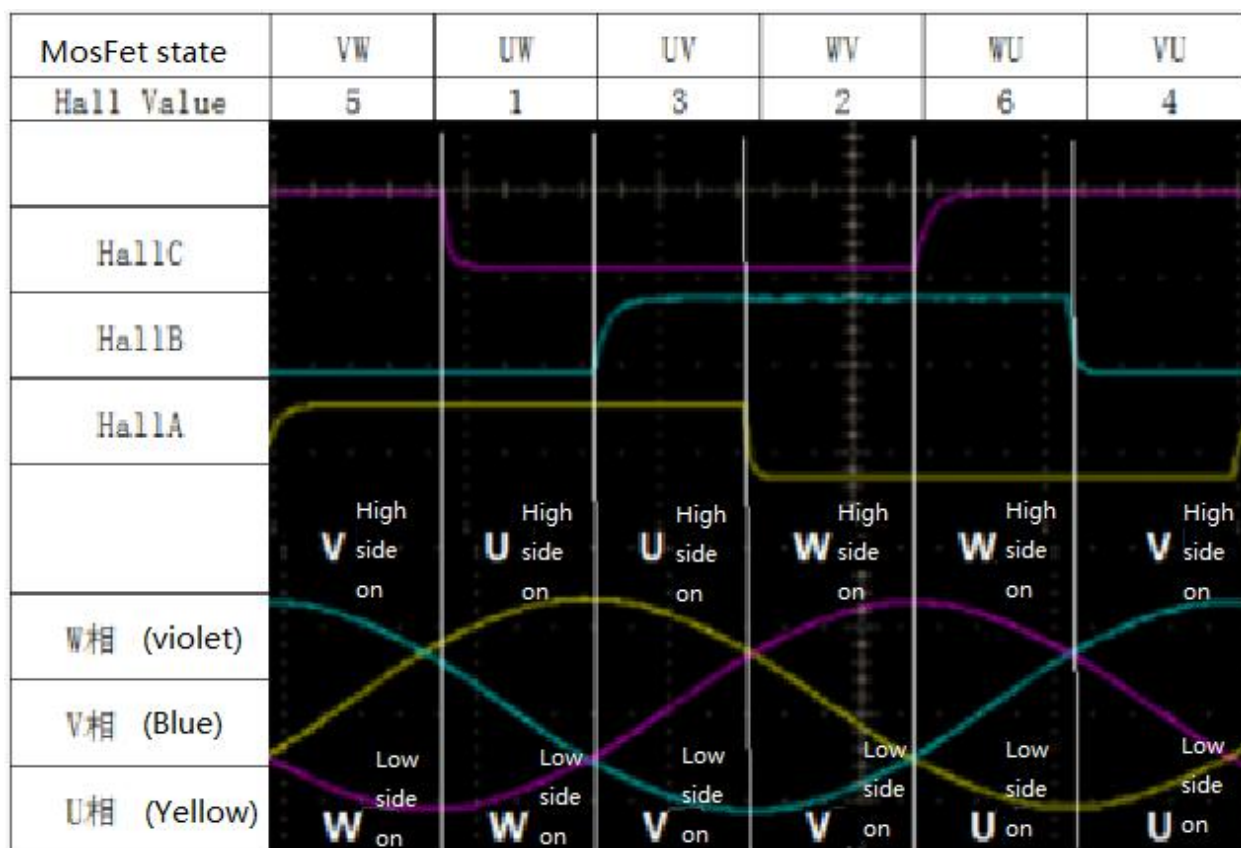
/*      V      W      W      U      U      V      */
uint8_t pwm_io_state[7] = { 0, 0x02, 0x04, 0x04, 0x01, 0x01, 0x02 };

/*      U      U      V      V      W      W      */
uint16_t psel_mask_port_a[7] = { 0, 0x0081, 0x0081, 0x0104, 0x0104, 0x0410, 0x0410 };

```

由此可推出 Hall 值对应的应该开管的状态的数组为:

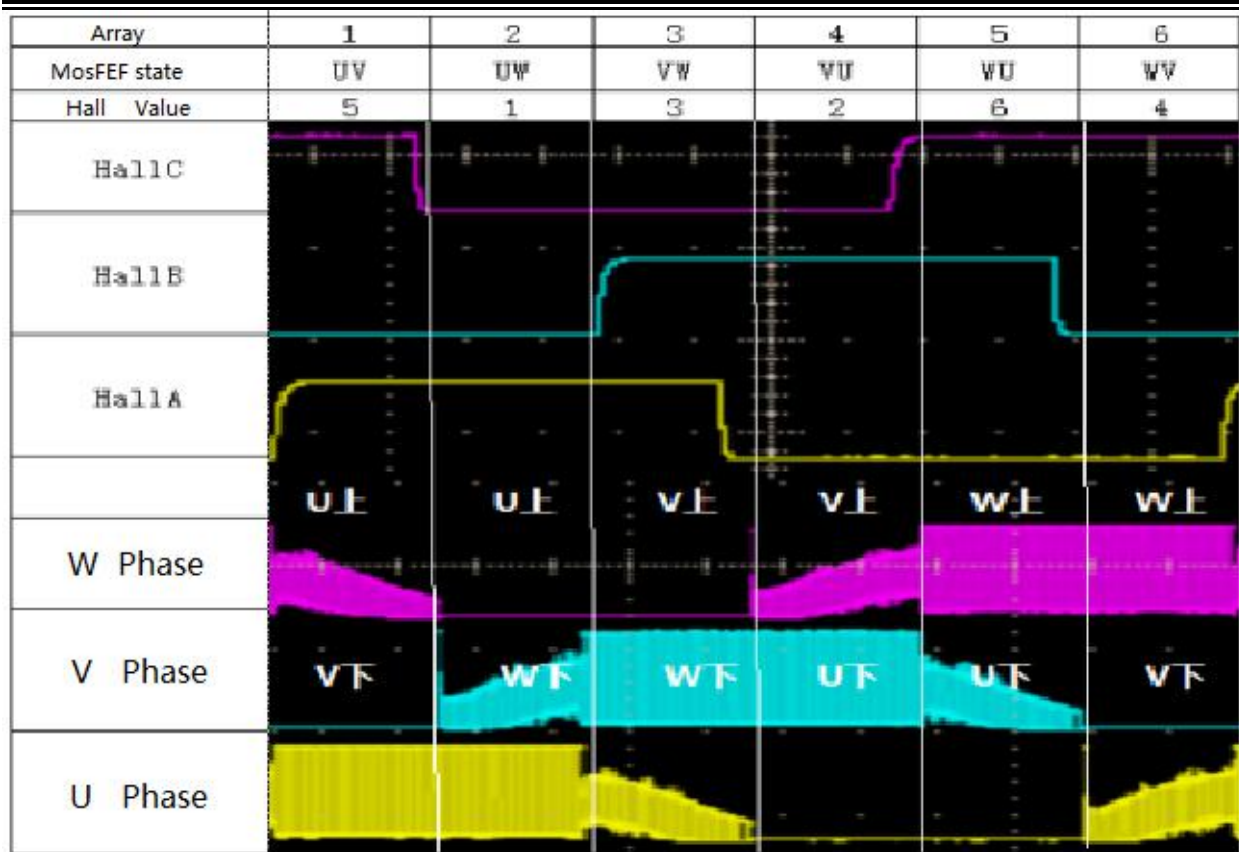
```
hall_to_mos_state_rev[] = {9,2,6,1,4,3,5,9};
```



则另一转向的数组则可由上一数组颠倒得到：

```
hall_to_mos_state[] = {9,5,3,4,1,6,2,9};
```

在 BEMF 模式下，校准 Hall 的方式与这里的原理也是类似的，



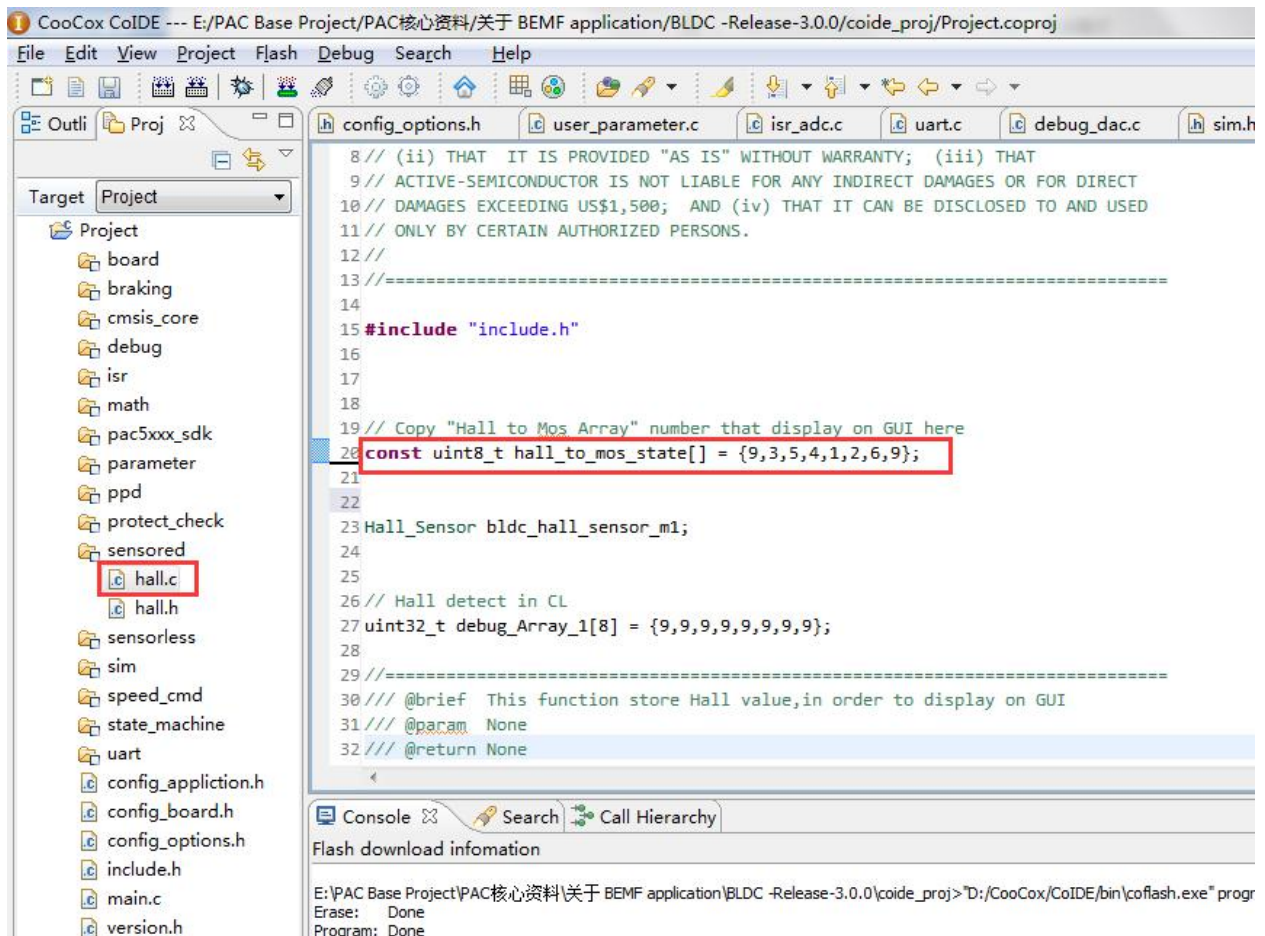
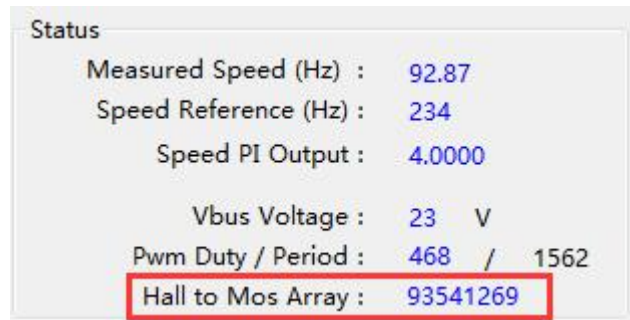
我们在 Firmware 里已经做了 Hall 自校准模式，可以在 BEMF 模式下转电机，再通过 GUI 显示 hall 状态与开管状态对应关系的数组，如下操作，

```

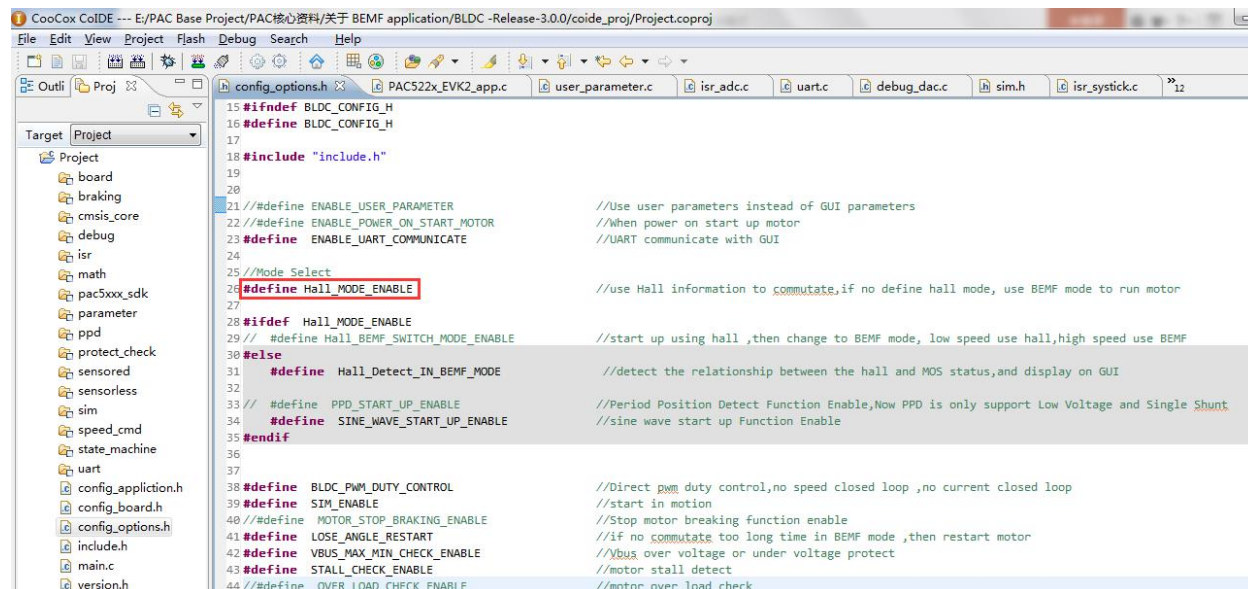
15 #ifndef BLDC_CONFIG_H
16 #define BLDC_CONFIG_H
17
18 #include "include.h"
19
20
21 // #define ENABLE_USER_PARAMETER           //Use user parameters instead of GUI parameters
22 // #define ENABLE_POWER_ON_START_MOTOR     //When power on start up motor
23 #define ENABLE_UART_COMMUNICATE          //UART communicate with GUI
24
25 //Mode Select
26 // #define Hall_MODE_ENABLE                 //use Hall information to commutate, if no define hall mode, use BEMF mode to run motor
27
28 #ifdef Hall_MODE_ENABLE
29 // #define Hall_BEMF_SWITCH_MODE_ENABLE     //start up using hall ,then change to BEMF mode, low speed use hall, high speed use BEMF
30 #else
31 #define Hall_Detect_IN_BEMF_MODE          //detect the relationship between the hall and MOS status, and display on GUI
32
33 // #define PPD_START_UP_ENABLE              //Period Position Detect Function Enable, Now PPD is only support Low Voltage and Single Shunt
34 #define SINE_WAVE_START_UP_ENABLE        //sine wave start up Function Enable
35 #endif
36
37
38 #define BLDC_PWM_DUTY_CONTROL             //Direct PWM duty control, no speed closed loop, no current closed loop
39 #define SIM_ENABLE                        //start in motion
40 // #define MOTOR_STOP_BRAKING_ENABLE        //Stop motor braking function enable
41 #define LOSE_ANGLE_RESTART                //if no commutate too long time in BEMF mode, then restart motor
42 #define VBUS_MAX_MIN_CHECK_ENABLE         //VBUS over voltage or under voltage protect
43 #define STALL_CHECK_ENABLE                //motor stall detect
44 // #define OVER_LOAD_CHECK_ENABLE           //motor over load check
  
```



先配置为 BEMF 模式，并打开 “**#define Hall\_Detect\_IN\_BEMF\_MODE**”，将 BEMF 模式调试好，就可在 GUI 上显示 hall 状态与开管状态对应关系的数组，待此数组值稳定，再记录下来，填写到 firmware 里的 “hall.c” 函数里，



然后，再将模式改为 Hall 模式，编译，Download，就可用 Hall 信息跑电机了，并且可以正反转，在“config\_options.h”里改变模式配置，选择 Hall 模式，如下图，



重新编译烧录程序，连接 GUI，写入参数，启动电机，波形如下，



## 更改履历

( 日期 )

( 更改事项 )