



Fremont Micro Devices

FT61F24X 在无刷电机筋膜枪上的应用

日期	版本	描述
2021-06-17	V1.00	初版
2021-06-25	V1.01	修改 LED 指示灯描述部分
2021-07-28	V2.00	修改部分 config 参数

目录

1	硬件介绍	- 3 -
1.1	概述	- 3 -
1.2	主控芯片介绍	- 3 -
1.3	充电模块	- 4 -
1.4	稳压模块	- 5 -
1.5	下载接口	- 5 -
1.6	P+N 半桥驱动系统及母线电流采样电路	- 6 -
1.7	反电动势采集电路	- 7 -
1.8	母线电压检测	- 7 -
1.9	按键及指示灯	- 8 -
2	软件	- 9 -
2.1	概述	- 9 -
2.2	完整程序流程图	- 10 -
2.3	电机启动流程图	- 11 -
2.4	ADC 采集数据时刻说明	- 12 -
3	软件参数配置	- 13 -

1 硬件介绍

1.1 概述

本应用使用的是 TSSOP20 封装的 FT61F245 芯片。使用 3 路 PWM 输出和 3 个通用 IO 口控制 3 个 N+P 半桥、2 个比较器（其中一个比较器有两个输入可以切换）用于采集反电动势、1 路 ADC 采集母线电压、1 路 ADC 采集母线电流、1 个 IO 用于按键检测，3 个 IO 用于驱动 LED 指示，使用 TYPE-C 接口充电，使用 LDO 降压给 MCU 供电。

1.2 主控芯片介绍

下图是主控芯片 FT61F245。主控芯片系统时钟 8MHz，4KROM，256B SRAM。

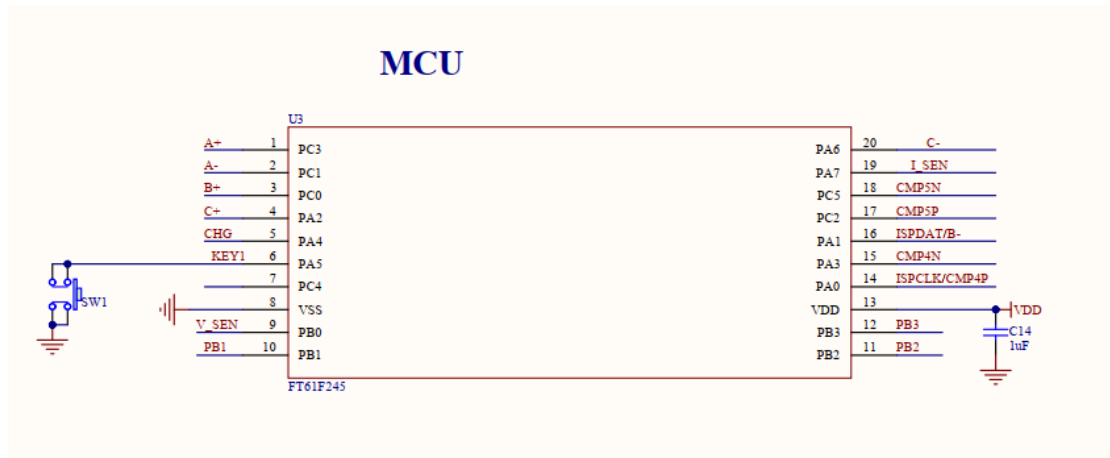


图 1.2-1

详细的引脚分配如下：

PIN	管脚名称	说明
1	PC3	A 相下桥驱动
2	PC1	A 相上桥驱动
3	PC0	B 相下桥驱动
4	PA2	C 相下桥驱动
5	PA4	充电状态检测脚
6	PA5	按键
7	PC4	
8	VSS	芯片 GND

9	PB0	母线电压监测脚
10	PB1	LED 灯驱动 IO1
11	PB2	LED 灯驱动 IO2
12	PB3	LED 灯驱动 IO3
13	VDD	芯片 VDD
14	PA0	程序下载脚/比较器 4 的负极输入
15	PA3	比较器 4 的负极输入
16	PA1	B 相上桥驱动
17	PC2	比较器 4、5 的正极输入
18	PC5	比较器 5 的负极输入
19	PA7	母线电流监测脚
20	PA6	C 相上桥驱动

1.3 充电模块

下图是 TYPE-C-6P 立贴式的 USB 充电口，只使用了 VBUS 和 GND 用于充电。

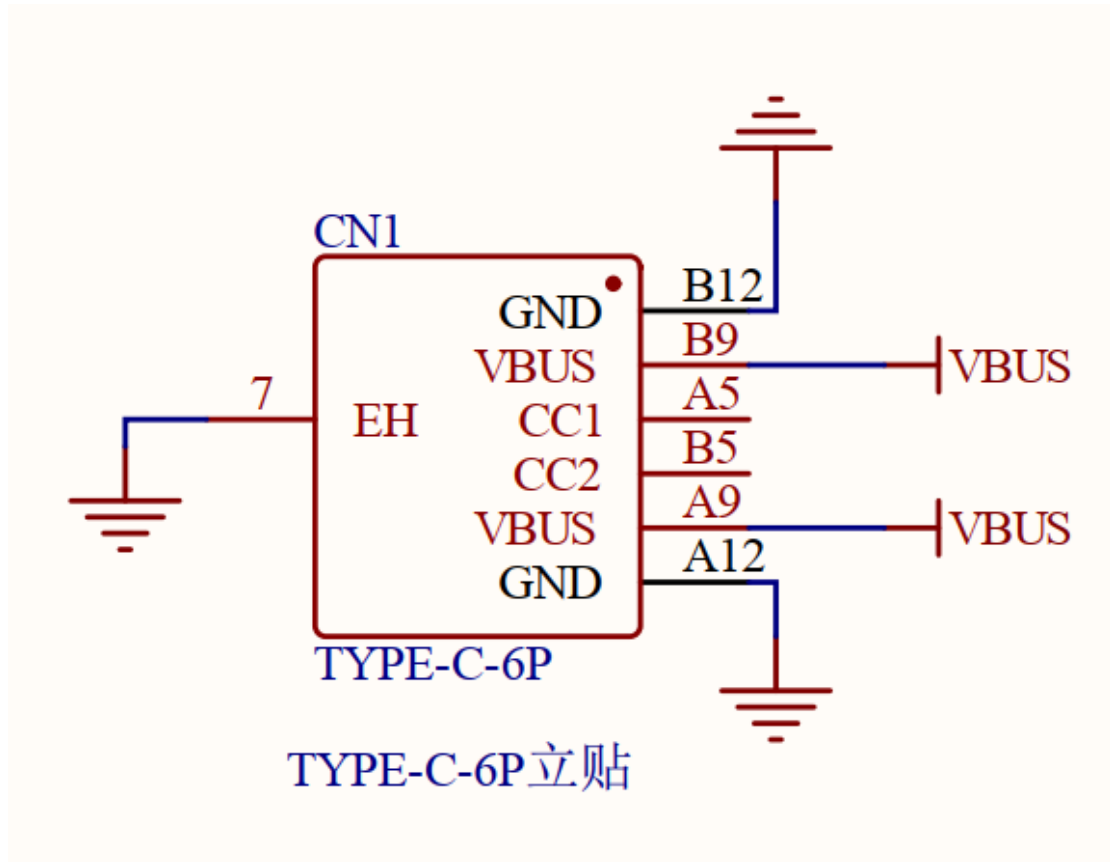


图 1.3-1

通过 TYPEC 的 5V 输入，给 CS5095 供电，充电过程由电源管理芯片控制，充电电流可以通过调整 R5，详细说明请参考“CS5095 芯片手册”。充电的状态由 CS5095 的第 8Pin 输出接入 MCU 的 PA4 Pin，MCU 根据接收到的信号使用两个指示灯指示充电状态。

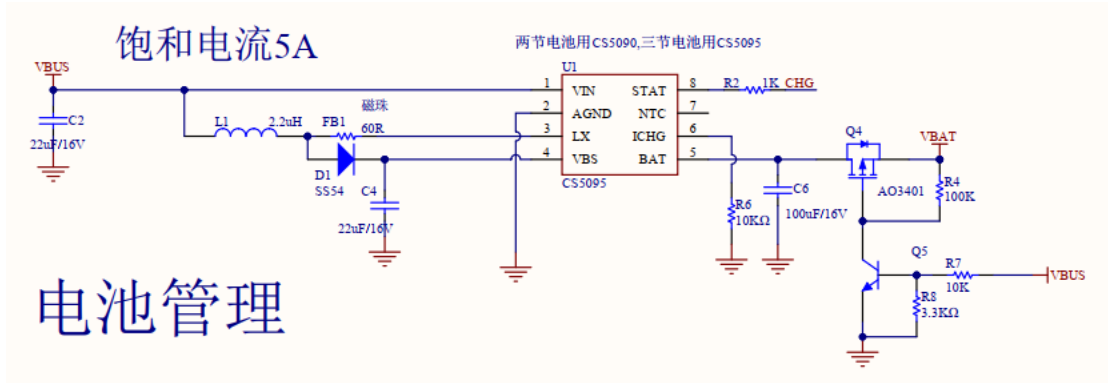


图 1.3-2

1.4 稳压模块

电池的 12V 经过 78L05 之后稳定电压在 5V 给 MCU 供电。

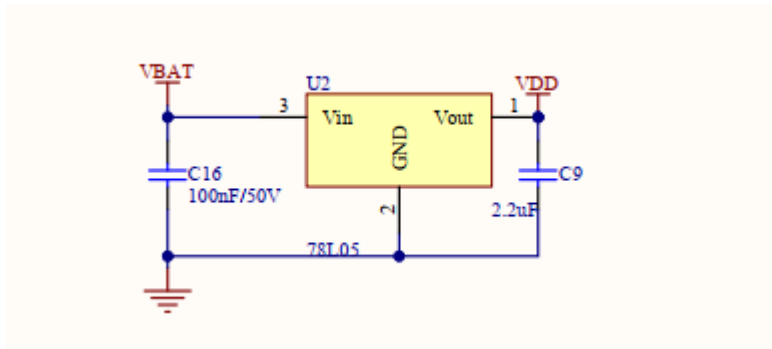


图 1.4-1

1.5 下载接口

由于 debug 口已经在程序中用于控制电机，所以 debug 口现在只能用于程序下载。

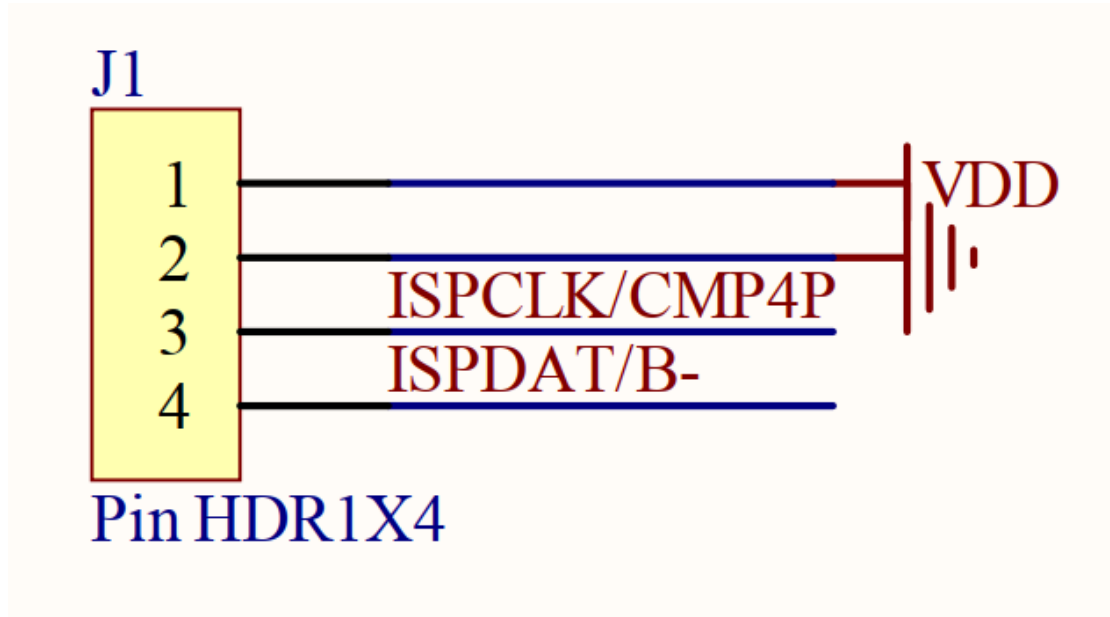


图 1.5-1

1.6 P+N 半桥驱动系统及母线电流采样电路

电机驱动系统中使用了 6 个 IO 驱动，其中上半桥控制 IO 输出高或者低，控制三极管及 MOS 管的开关，下半桥使用 PWM 控制。RS1 是采样电阻，采样信号经过 RS1 和 C15 组成的 RC 滤波电路之后直接输入到 MCU 的 ADC 通道处理。

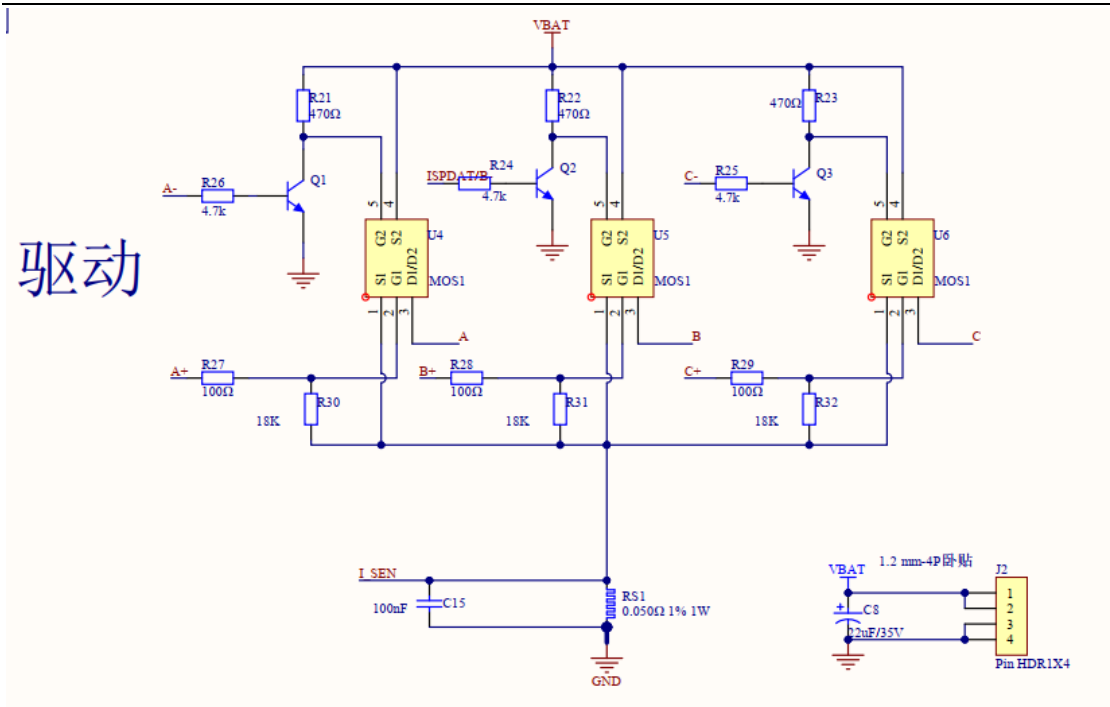


图 1.6-1

1.7 反电动势采集电路

比较器：公共极接 CMP5P，负极分别接 CMP5N/CMP4P/CMP4N。12V 的 VCC 经过分压网络分压至 MCU 可以识别的电压，通过比较器负极和公共极的比较来换相。

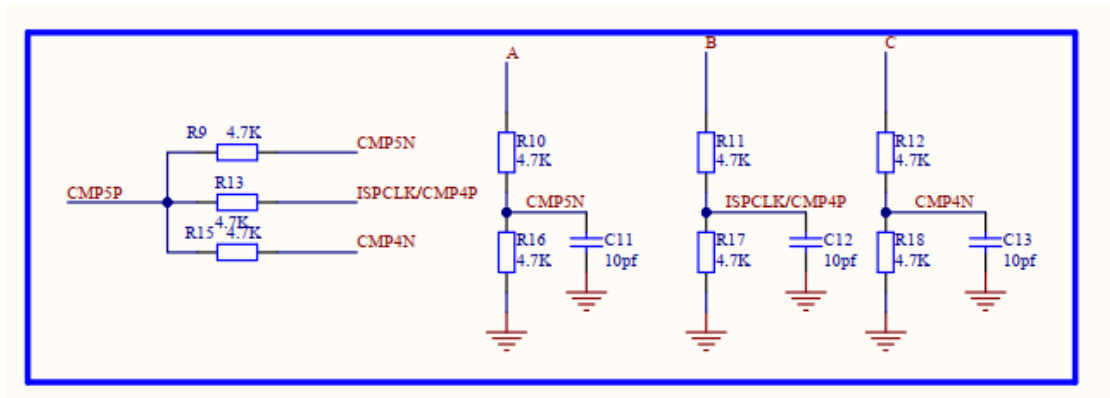


图 1.7-1

1.8 母线电压检测

电池经过分压电阻分压之后送到 MCU 的电压监测脚，时时监测当前的电池电压。在使用

过程中，当电压低于一定的配置值之后，相应指示灯会闪烁提示，表示需要充电。

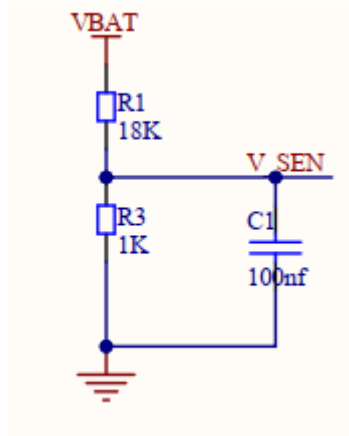


图 1.8-1

1.9 按键及指示灯

1 个 IO 口控制按键的长按、短按功能。现在配置的是长按 1.5 秒开关机，短按循环调节速度。

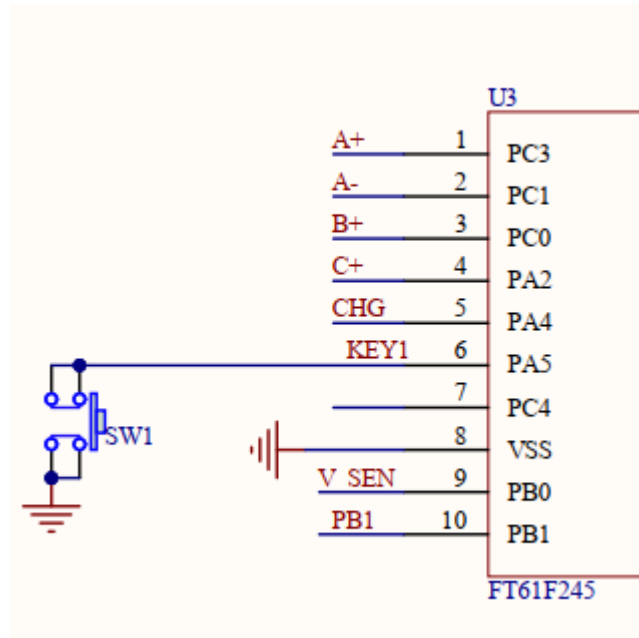


图 1.9-1

3 个 IO 控制 4 个速度档位指示灯及两个电池状态指示灯，D2/3/4/5 用于速度档位指示，同一个颜色的灯（蓝），D6/7 用于电池状态指示，一个蓝灯，一个红灯。

三个 IO 口控制 6 个灯是通过扫描的方式控制的，也就是说，依次查询相应 LED 是否需要点亮，循环查询 6 个灯。比如先查询 D2 是否要亮，亮（PB1 输出高电平，PB2 输出低电平，PB3 高阻态），然后依次查询 D3、D4、D5、D6 和 D7 的状态，完成一次循环，下次重新从 D2 开始查询。

目前 D2 亮表示速度一档，D3 速度二档，D4 速度三档，D5 速度四档，D6 亮表示电机在工作或者充满电，D7 亮表示充电中，D6、D7 闪烁表示充电异常。

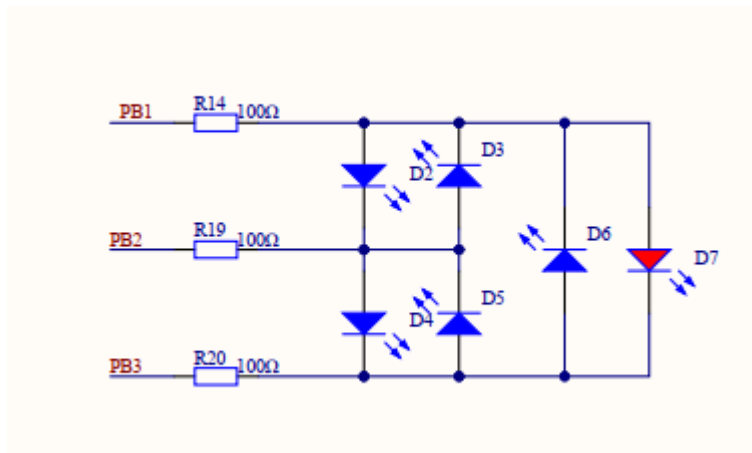


图 1.9-2

2 软件

2.1 概述

整个软件的结构是由主循环及中断系统组成的，主循环处理非紧急的任务，比如按键检测、LED 指示、速度调节和 ADC 数据处理等，中断系统处理紧急的任务，比如定时、电机换相处理和启动 ADC 采样等。

MCU 上电之后，初始化 IO、ADC 等外设，以及初始化变量，之后进入主循环，MCU 一直执行主循环及中断任务。电机没有启动前，MCU 只会执行简单的电压监测和中断定时功能。

当按键触发电机启动之后，MCU 则开始了全任务执行。

2.2 完整程序流程图

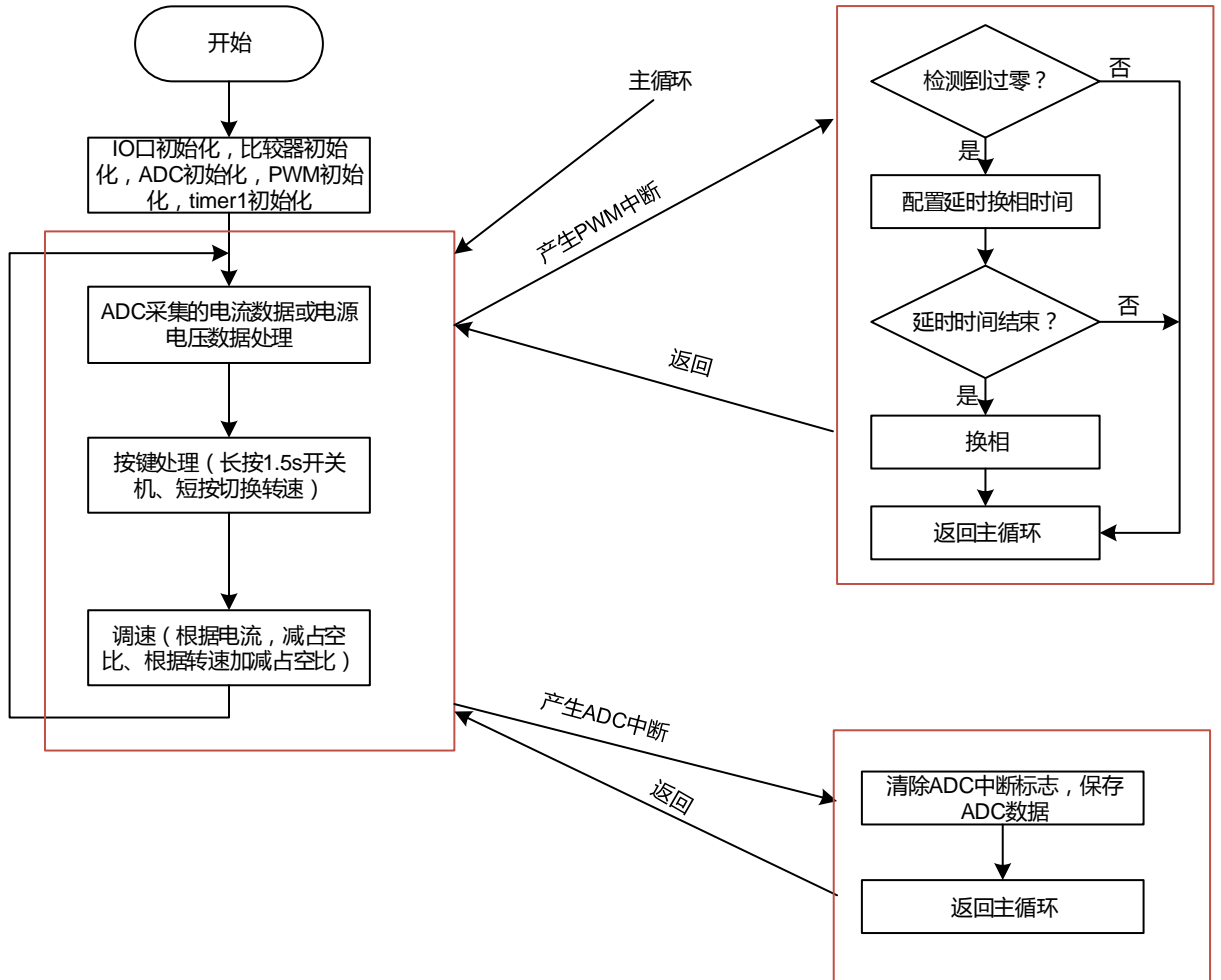


图 2.2-1

2.3 电机启动流程图

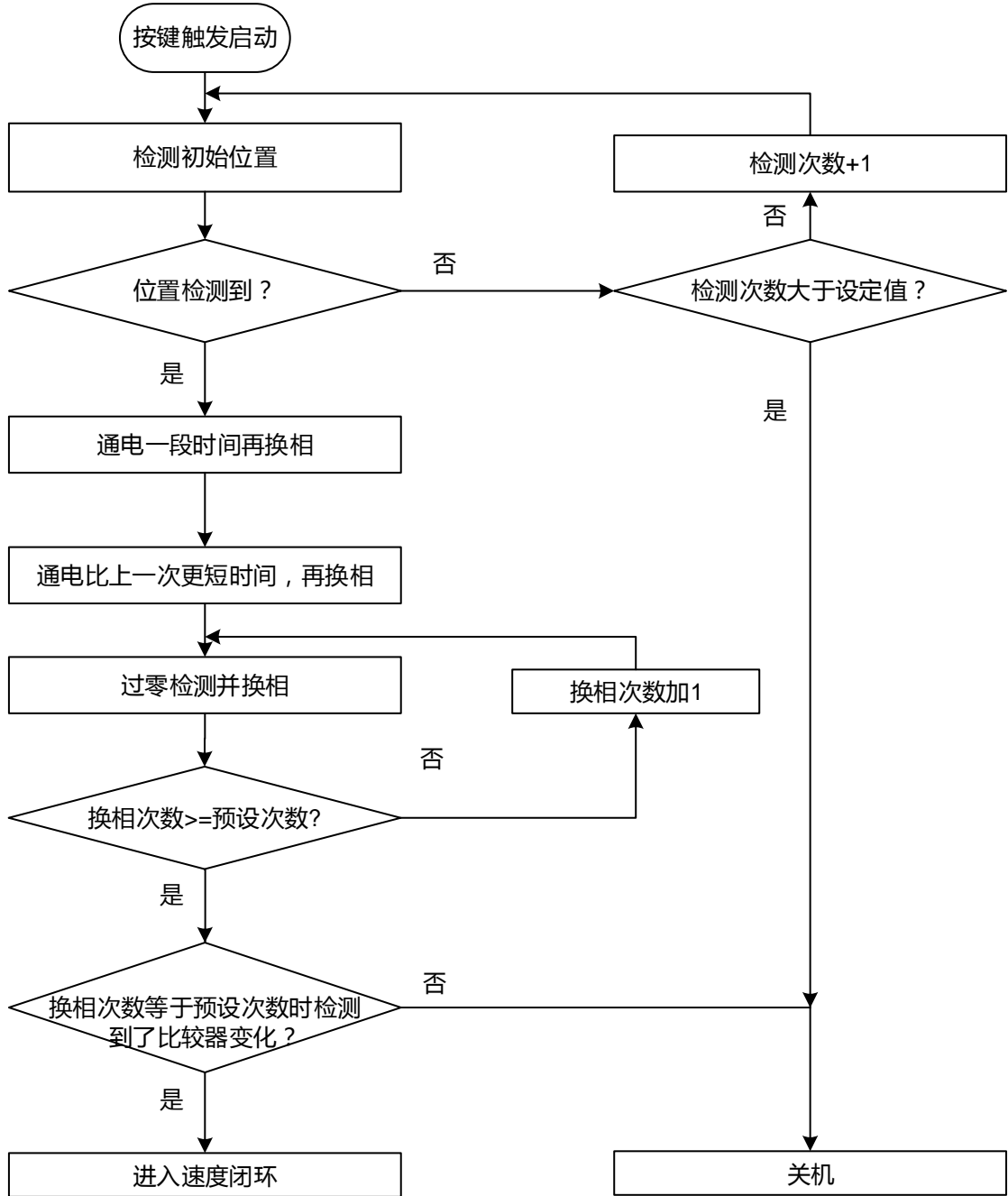


图 2.3-1

2.4 ADC 采集数据时刻说明

本应用使用了两个通道，一个通道采集母线电流，另一个通道采集母线电压，程序中配置了分别采集 16 次之后，切换通道，并且 ADC 采集时刻都是在 PWM 为高电平的时刻。如下图所示，其中黄色线，绿色线和橙色线是 3 路 PWM 输出，蓝色线是表示 ADC 启动采集时的 IO 口翻转。其中图 2.4-2 是图 2.4-1 放大后的波形。



图 2.4-1

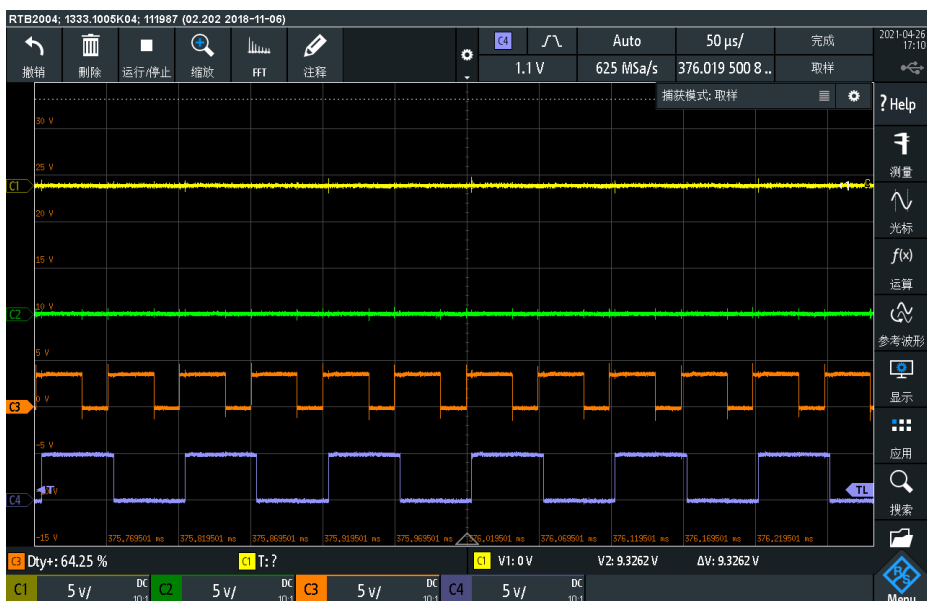


图 2.4-2

3 软件参数配置

所有能配置的参数都在 config.h 文件中，可以按照需要以及按键说明选择合适的参数修改。

```
#define POLE_PAIRS 7 //7 对极

#define FIRST_SPEED 2000 //2000RPM
#define SECOND_SPEED 2500 //2500RPM
#define THIRD_SPEED 3000 //3000RPM
#define FOUR_SPEED 3500 //3500RPM

/*电流、电压相关*/
#define CURRENT_LIMIT_VALUE 4
//16 次(1<<CURRENT_LIMIT_VALUE) ,ADC 采集(1<<CURRENT_LIMIT_VALUE) 次
取平均值
#define CURRENT_ADC_MAX 512
//公式: (X*CURRENT_RES *512),X 表示限制的最大电流, 超过最大电流停机, 这里配置
为 20A
#define POWER_LIMIT_MAX 64
//50:1.9A 64:2.5A, 公式: (X*CURRENT_RES *512), 限制功率的电流

#define VCC_LIMIT_MIN 190
//LIMIT = 1023/(2*19) * VMIN ,VMIN = 7V, 可修改 VMIN, 计算出最小启动电压
#define VCC_LIMIT_MAX 450
//LIMIT = 1023/(2*19) * VMAX ,VMAX = 16.5V, 可修改 VMAX, 计算出最大启动电压

#define VCC_LOW_POWER 229
//8.5V
```

/*PWM 相关*/

```
#define    START_POSITION_PWM          10          //10% , 对齐使用的占空  
比, 建议范围 7~15%  
  
#define    START_LOOP_PWM            35          //35% , 对齐之后的启动占  
空比, 建议 30~60%  
  
#define    START_POSITION_TIME        2000       //2000*50us = 100ms,对齐  
时间配置  
  
#define    START_BLANKING_TIME        10         //10*50us = 500us , 启动时  
的消磁时间配置  
  
#define    START_CHECK_ZERO_TIME      90         //90*50us = 4500us , 强制  
换相时间配置  
  
#define    START_CHECK_TIMES          6          //6 次, 启动阶段检测  
次数配置  
  
#define    START_DRAG_TIME            100        //100*50us = 5ms ,对  
齐之后, 第一相通电的时间配置  
  
#define    DRAG_MAX_TIME              300        //300*10ms = 3s , 连  
续 DRAG_MAX_TIME 秒强制换相之后停机配置  
  
#define    MOTOR_STALL_STOP_TIME      5714      //1500rpm 对应的时间  
5.7ms  
  
#define    MAX_PWM_REG                760        //最大限制占空比寄存
```

器的值: $760/800 = 0.95$

```
#define MIN_PWM_REG 280 //最小限制占空比寄存
```

器的值: $280/800 = 0.35$

```
#define AUTO_CHANGE_PHASE_TIME 1
```

```
// t>>AUTO_CHANGE_PHASE_TIME,配置延时换相时间,此处配置为过零时间(t)的一半
```

```
/*调速快慢控制*/
```

```
#define KP_ERR 2 //配置为 2, 2/800
```

= 0.0025, 表示每次改变 0.25%占空比

```
/*按键去抖和长按时间配置*/
```

```
#define DEBOUNCE_TIME 2 //20ms
```

10ms*2 = 20ms ,去抖时间配置

```
#define HOLD_TIME 150 //150 *10ms =
```

1.5s ,长按时间配置

```
/*指示灯相关*/
```

```
#define LED_TIMING 2000
```

```
// 2000 *50us = 100ms ,启动时的LED亮或者熄灭的快慢配置
```

```
/*进入睡眠的时间配置*/
```

```
#define SLEEP_TIME 500 //500 *10ms = 5s
```