

IPD电机调节软件设计

一、规范更新日志

更新日期	更新人	更新说明
2023.6.7	<div><div>李洋</div><div>黄欣然</div></div>	更新了第一阶段实现，通过按键控制ipd，左右屈光度等的电机运动。
2023.6.13	<div><div>李洋</div><div>黄欣然</div></div>	更新了指令模块，通过串口调试发送指令控制电机运动。并完成了ipd部分的功能验证。
2023.6.14	<div><div>李洋</div><div>黄欣然</div></div>	加入hall检测ipd调节是否到达极限位（暂时用了其它磁铁测试）
2023.11.28	<div><div>黄欣然</div></div>	加入i2c发送指令控制ipd，deep sleep mode休眠

二、背景与需求

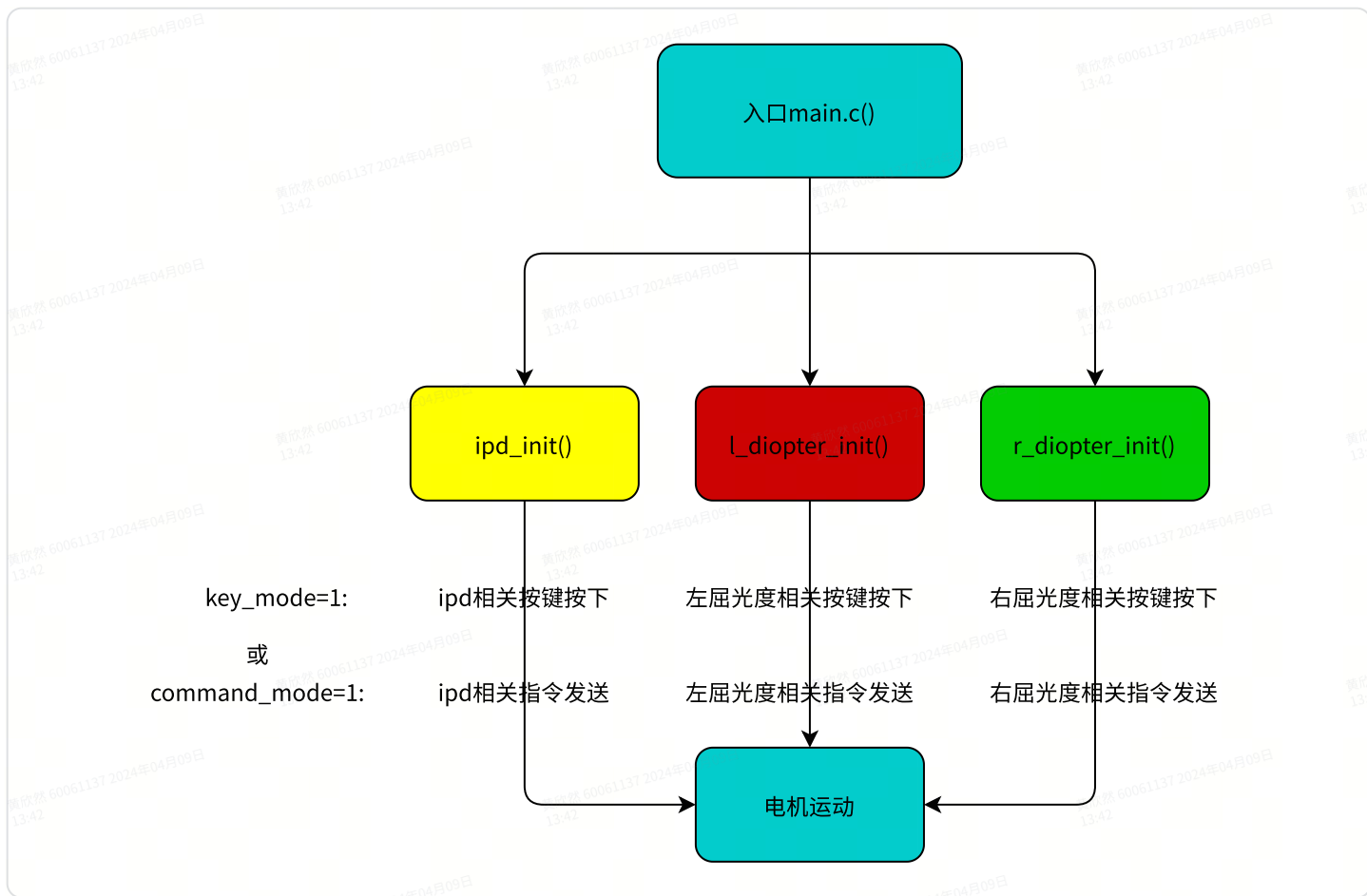
1.背景

2.需求

三、软件设计

3.1 整体设计

3.1.1 整体设计框图



3.1.2 设计思路

系统整体分为3个任务：分别控制ipd电机(aw8646)，左屈光度电机(l_diopter_aw8646)，右屈光度电机(r_diopter_aw8646)。按键控制 IPD、屈光度左、屈光度右共6个按键,按键按下开始运动,系统启动后，弹起停止运动,按下过程中如触发极限位则停止运动。

说明：三个电机的运动不受其他电机影响，完全独立。

3.2 ipd电机运动实现

3.2.1 pwm频率计算

IPD				
需求	行程 (mm)	时间 (s)	方向数	速度 (mm/s)
	16	0.5	2	1.666666667
电机+齿轮箱	步距角 (°)	齿轮箱速比	电机控制电流 (mA) 预设	
	18	23	208.5	
丝杆	螺距 (mm)	头数	外径 (mm)	
	0.4	6	2	
控制	细分	上升沿和下降沿触发	PWM频率 (Hz)	最大脉冲数
	32	1	16000	48000

说明：暂时设置总运动时长为3s，可以通过修改时间来改变pwm频率。

3.2.2 具体实现步骤

ipd调节马达驱动芯片的step控制引脚为PA7，定时器选择TIMER2_CH1。

1. 检测ipd的某个按钮（共两个）按下，电机开始运动，若按下从左到右运动的按键（PB7）则调用aw8646_start_play(AW_POSITIVE)，按下右到左运动的按键（PB8）则调用aw8646_start_play(AW_NEGATIVE)；
2. 检测到未加入hall即adc_hall=0时，使用步数进行检测，步数（cur_step）已达到最大脉冲数或已减少到0，调用aw8646_stop_play()，停止运动；加入hall时，检测到电压大于等于adc_value_max或小于adc_value_min时，停止运动（如果是指令模式仍然需要加上步数的判断）；

注：步数检测和hall检测全部在中断中完成，可能比较耗时，后续可能要试验+修改。

3. 两个按键优先级相同，若先按下一个，另一个无效。代码中int变量ipd_increase_PB7和ipd_decrease_PB8为按键是否按下的标识。

3.2.3 通过HALL元件判读位置

用adc_hall判断，1表示添加了HALL元件。

设置adc_value_max和adc_value_min表示运动过程的极限位置。

3.3 屈光度电机运动实现

3.3.1 屈光度的pwm频率计算

屈光度

需求	运动角度 (°)	时间 (s)	速度 (°/s)	速度 (rpm)
	65	5	13	2.166666667
电机+齿轮箱	步距角 (°)	齿轮箱速比	电机控制电流 (mA) 预设	
	18	350	305	
控制	细分	上升沿和下降沿触发	PWM频率 (Hz)	最大脉冲数
	32	1	8088.888889	40444.44444

说明：暂时设置总运动时长为5s，可以通过修改时间来改变pwm频率。

左屈光度调节马达驱动芯片的step控制引脚为PB0，定时器选择TIMER0_CH1_ON。

右屈光度调节马达驱动芯片的step控制引脚为PB15，定时器选择TIMER14_CH1。

3.3.2 左屈光度电机运动实现

1. 检测左屈光度的某个按钮（共两个）按下，电机开始运动，若按下增加屈光度的按键（PB5）则调用aw8646_start_play_l(AW_POSITIVE)，按下减少屈光度的按键（PB6）则调用aw8646_start_play_l(AW_NEGATIVE);
2. 检测到通过限位开关（两个）：前端（PA11）和后端（PA12）的按下，调用aw8646_stop_play_l()，停止运动;
3. 两个按键优先级相同，若先按下一个，另一个无效。代码中int变量l_increase_PB5和l_decrease_PB6为按键是否按下的标识。

3.3.3 右屈光度电机运动实现

1. 检测右屈光度的某个按钮（共两个）按下，电机开始运动，若按下增加屈光度的按键（PB9）则调用aw8646_start_play_r(AW_POSITIVE)，按下减少屈光度的按键（PB10）则调用aw8646_start_play_r(AW_NEGATIVE);
2. 检测到通过限位开关（两个）：前端（PC13-TAMPER-RTC）和后端（PA15）的按下，调用aw8646_stop_play_r()，停止运动;
3. 两个按键优先级相同，若先按下一个，另一个无效。代码中int变量r_increase_PB9和r_decrease_PB10为按键是否按下的标识。

3.4 指令控制

3.4.1 指令设计

- 代码中key_mode表示按键模式，command_mode表示指令模式，可在代码里设置，1表示开启。
- 暂定13个指令。指令1-6发送1000个对应的pwm脉冲，7-12发送10000脉冲,help指令会打印其它指令。精细调节可使用指令1-6，粗略调节使用7-12。
- 使用方式例如：ins1。然后ipd会从左到右运动1000个pwm。

指令1: ipd increase 1000step---->ins1

指令2: ipd decrease 1000step---->ins2

指令3: l increase 1000step ---->ins3

指令4: l decrease 1000step ---->ins4

指令5: r increase 1000step ---->ins5

指令6: r decrease 1000step ---->ins6

指令7: ipd increase 10000step---->ins7

指令8: ipd decrease 10000step----->ins8

指令9: l increase 10000step ---->ins9

指令10: l decrease 10000step ---->insa

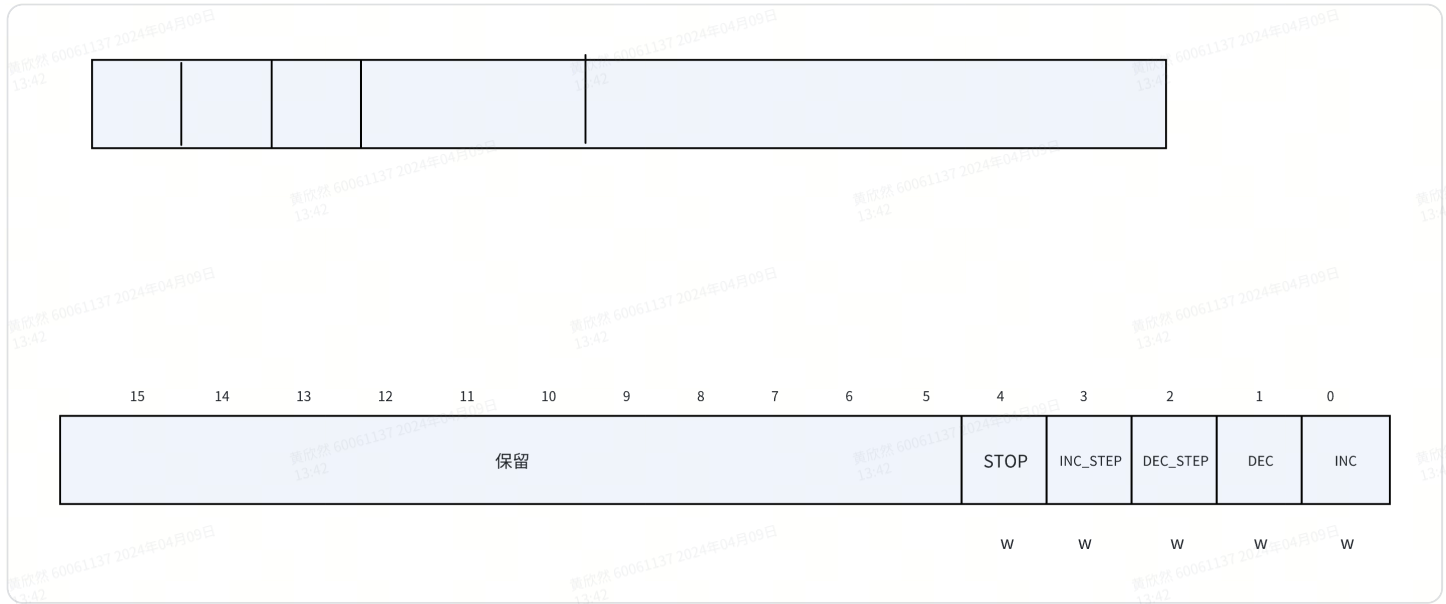
指令11: r increase 10000step ---->insb

指令12: r decrease 10000step ---->insc

注意：若是执行指令7，运动7000个pwm就已到极限位，那么将不会运动10000个pwm。

3.4.2细节实现

1. 在main.c的while(1)中，会判断目前模式，如果command_mode=1，那么将进入指令模式；
2. ins_num[x]=1表示指令x正在执行，在定时器中判断使用的模式和对应的指令。注意，在指令1正在执行的时候，发送命令执行指令2，指令2将不会执行。



3.4 i2c指令控制

3.5 低功耗休眠模式

```
1 /* configure tamper key EXTI */
2 gd_eval_key_init(KEY_USER, KEY_MODE_EXTI);
3
4 rcu_periph_clock_enable(RCU_GPIOA);
5 rcu_periph_clock_enable(RCU_GPIOB);
6 rcu_periph_clock_enable(RCU_GPIOC);
7
8 gpio_mode_set(GPIOA, GPIO_MODE_ANALOG, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);
9 //gpio_mode_set(GPIOB, GPIO_MODE_ANALOG, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);
10 gpio_mode_set(GPIOC, GPIO_MODE_ANALOG, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);
11
12 rcu_periph_clock_enable(RCU_PMU);
13 rtc_pre_config();
14
15 pmu_to_deepsleepmode(PMU_LDO_LOWPOWER, PMU_LOWDRIIVER_DISABLE, WFI_CMD);
16 system_clock_84m_hxtal();
17 rcu_periph_clock_enable(RCU_GPIOA);
18 rcu_periph_clock_enable(RCU_GPIOB);
19 rcu_periph_clock_enable(RCU_GPIOC);
20
21 gpio_mode_set(GPIOA, GPIO_MODE_AF, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);
22 gpio_mode_set(GPIOB, GPIO_MODE_AF, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);
```

```
23 gpio_mode_set(GPIOC, GPIO_MODE_AF, GPIO_PUPD_PULLUP, GPIO_PIN_ALL);  
24  
25 //systick_config();  
26 printf("init start2\r\n");
```

将boot0拉高可以强制下载，如果不小心进入deep sleep mode，可用此法唤醒