

---

# FR801xH ADC 使用说明

版本号: V0.1

日期: 2022.03



## 修订版本

版本	日期	更新内容
V0.1	2022.03.21	首版

FreqChip Confidential

## 目录

1. ADC 配置 .....	4
1.1 采样电压来源(Cfg.src).....	4
1.2 采样基准源选择(Cfg.ref_sel).....	4
1.3 内部基准源选择(Cfg.int_ref_cfg).....	5
1.4 配置采样通道(Cfg.channels).....	5
1.5 配置采样方式(Cfg.route.pad_to_sample).....	5
1.6 时钟源选择(Cfg.clk_sel).....	5
1.7 时钟分频比(Cfg.clk_div).....	6
2. 软件配置 .....	7
2.1 采样 VBAT 脚电压 .....	7
2.2 采样 PAD 脚电压 .....	7
2.2.1 参考电压设置为外部的 LDO_OUT(AVDD)脚电压 .....	7
2.2.2 参考电压设置为内部 1.2v .....	8
2.3 多通道采集 .....	8
2.4 ADC 中断采集 .....	9
3. 硬件参考 .....	11
4. ADC 采集问题分析参考 .....	12
联系方式 .....	1

## 1. ADC 配置

ADC 的主要配置参数如表格 1-1 所示，具体的使用说明在后续小节详细阐述。

表格 1-1 ADC 配置参数表

参数名称	含义
Cfg.src	采样电压来源
Cfg.ref_sel	采样基准源选择
Cfg.int_ref_cfg	内部基准源选择
Cfg.channels	脚采样时配置采样通道
Cfg.route.pad_to_sample	配置采样方式
Cfg.clk_sel	时钟源选择
Cfg.clk_div	时钟分频比

### 1.1 采样电压来源(Cfg.src)

表格 1-2 采样电压来源

参数	含义
Cfg.src	采样电压来源
<b>说明：</b> 可以有 VBAT 和 PAD 两种选择，VBAT 即是直接采样 VBAT 脚电压，PAD 就是采样 PD4-PD7 PAD 脚位的电压。	

### 1.2 采样基准源选择(Cfg.ref\_sel)

表格 1-3 采样基准源选择

参数	含义
Cfg.ref_sel	采样基准源选择
<b>说明：</b> 可以配置为内部基准源 ADC_REFERENCE_INTERNAL，或者是外部 AVDD 基准源 ADC_REFERENCE_AVDD，当基准源是 AVDD 时，LDO_OUT 电压配置建议用默认的 2.9v 配置，该电压在芯片出厂时有做校准，其他电压没有实际做校准，如果电压必须修改，可以等比类推出对应电压的校准值来做计算，或者基准源切换到内部 1.2v。	

### 1.3 内部基准源选择(Cfg.int\_ref\_cfg)

参数	含义
Cfg.int_ref_cfg	内部基准源选择
<b>说明:</b> 这个配置只有当 ref_sel 配置为 ADC_REFERENCE_INTERNAL 时才需要配置，一般配置为 ADC_INTERNAL_REF_1_2，1.2v 我们芯片在出厂时有做校准，其他电压没有实际做校准	

### 1.4 配置采样通道(Cfg.channels)

参数	含义
Cfg.channels	脚采样时配置采样通道
<b>说明:</b> bit0-bit3 对应 PD4-PD7，对应 ch11-ch14。	

### 1.5 配置采样方式(Cfg.route.pad\_to\_sample)

参数	含义
Cfg.route.pad_to_sample	配置采样方式
<b>说明:</b> pad_to_sample: 直接 pad 采样到硬件处理，最终得到数据。 pad_to_div: 采用分压采样，比如参考电压是 1.2v，输入电压是 2.4v 则可以采用 div 的采样方式，这种采样方式需要配合 div_res 和 div_cfg 来做。 div_res: 配置内部分压方式的电阻总大小，有 105 欧、415 欧、7.25K、30K 四种，注意电阻越小需要的驱动能力越强。 div_cfg: 电阻分压配置，有 1/4、1/3、1/2、2/3 几种配置。 pad_to_buffer: 一般用于外围电压输入驱动能力很弱的时候，一般没用。	

### 1.6 时钟源选择(Cfg.clk\_sel)

参数	含义
Cfg.clk_sel	时钟源选择
<b>说明:</b> 单通道采集时可以配置为 24M 分频 ADC_SAMPLE_CLK_24M_DIV13 或者多通道采集时一定要配置成低速的 RC 时钟 64K 分频 ADC_SAMPLE_CLK_64K_DIV13，多通道时采样率高会导致通道之间的相互干扰。另外内部 RC 时钟不是很准，频率计算可以按 56K 计算，DIV13 是配置采样时间，稳定建议按 1/15 去计算	

## 1.7 时钟分频比(Cfg.clk\_div)

参数	含义
Cfg.clk_div	时钟分频比
<p><b>说明:</b></p> <p>高 2bit 是配置 64K 时的分频比, 低 4bit 是配置 24M 时的分频比, 配合 clk_sel 计算采样时钟, 如配置为 0x3f:</p> $\text{clk} = 24\text{M} / (0x0f + 1) / 15 = 100\text{K}$ $\text{Clk} = 56\text{K} / (0x03 + 1) / 15 = 0.93\text{K}$ <p>根据实际情况配置, 多通道采集时, 采样率是低速时钟源, 分频比可以配置成 0x01, 适当提高采样率</p>	

ADC 值采样之后, 做电压转换时, 需要先调用 `adc_get_ref_voltage` 来获取基准源的校准电压, 内部基准源参数就写 `ADC_REFERENCE_INTERNAL`, 外部就写 `ADC_REFERENCE_AVDD`。

## 2. 软件配置

从参考电压，采样点几个配置分别列举如下：

### 2.1 采样 VBAT 脚电压

一般电池直接供电到VBAT脚时，可以通过直接采集VBAT脚的adc值来计算当前电量，此时adc的参考电压只能是内部的1.2v，内部有1/4分压的电路设计，参考代码如下：

```
struct adc_cfg_t cfg;
uint16_t result, ref_vol, vbat_vol;
memset((void*)&cfg, 0, sizeof(cfg));
cfg.src = ADC_TRANS_SOURCE_VBAT;
cfg.ref_sel = ADC_REFERENCE_INTERNAL;
cfg.int_ref_cfg = ADC_INTERNAL_REF_1_2;
cfg.clk_sel = ADC_SAMPLE_CLK_24M_DIV13;
cfg.clk_div = 0x3f;
adc_init(&cfg);
adc_enable(NULL, NULL, 0);
adc_get_result(ADC_TRANS_SOURCE_VBAT, 0, &result);
ref_vol = adc_get_ref_voltage(ADC_REFERENCE_INTERNAL);
vbat_vol = (result * 4 * ref_vol) / 1024;
```

### 2.2 采样 PAD 脚电压

一般用于采样外部电压，电池或者其他电路输入的电压，根据参考电压的配置使用不同的量程。参考代码如下：

#### 2.2.1 参考电压设置为外部的 LDO\_OUT(AVDD)脚电压

```
struct adc_cfg_t cfg;
uint16_t result, ref_vol, voltage_vol;
system_set_port_mux(GPIO_PORT_D, GPIO_BIT_4, PORTD4_FUNC_ADC0);
memset((void*)&cfg, 0, sizeof(cfg));
cfg.src = ADC_TRANS_SOURCE_PAD;
cfg.ref_sel = ADC_REFERENCE_AVDD;
cfg.channels = 0x01;
cfg.route.pad_to_sample = 1;
```

```

cfg.clk_sel = ADC_SAMPLE_CLK_24M_DIV13;
cfg.clk_div = 0x3f;
adc_init(&cfg);
adc_enable(NULL, NULL, 0);
adc_get_result(ADC_TRANS_SOURCE_PAD, 0x01, &result);
ref_vol = adc_get_ref_voltage(ADC_REFERENCE_AVDD);
voltage_vol = (result * ref_vol) / 1024;

```

### 2.2.2 参考电压设置为内部 1.2v

```

struct adc_cfg_t cfg;
uint16_t result, ref_vol, voltage_vol;
system_set_port_mux(GPIO_PORT_D, GPIO_BIT_4, PORTD4_FUNC_ADC0);
memset((void*)&cfg, 0, sizeof(cfg));
cfg.src = ADC_TRANS_SOURCE_PAD;
cfg.ref_sel = ADC_REFERENCE_INTERNAL;
cfg.int_ref_cfg = ADC_INTERNAL_REF_1_2;
cfg.channels = 0x01;
cfg.route.pad_to_sample = 1;
cfg.clk_sel = ADC_SAMPLE_CLK_24M_DIV13;
cfg.clk_div = 0x3f;
adc_init(&cfg);
adc_enable(NULL, NULL, 0);
adc_get_result(ADC_TRANS_SOURCE_PAD, 0x01, &result);
ref_vol = adc_get_ref_voltage(ADC_REFERENCE_INTERNAL);
voltage_vol = (result * ref_vol) / 1024;

```

## 2.3 多通道采集

多通道采集时，一般在adc\_enable之后做个delay，保证采样完成；比如分频配置0x01， $\text{clk} = 56\text{k}/2/15 = 1.86\text{k} = 540\mu\text{s}$ 左右，两个通道转换完需要1ms以上。

```

struct adc_cfg_t cfg;
uint16_t result[2], ref_vol, voltage_vol1, voltage_vol2;
system_set_port_mux(GPIO_PORT_D, GPIO_BIT_4, PORTD4_FUNC_ADC0);
system_set_port_mux(GPIO_PORT_D, GPIO_BIT_5, PORTD5_FUNC_ADC1);
memset((void*)&cfg, 0, sizeof(cfg));
cfg.src = ADC_TRANS_SOURCE_PAD;
cfg.ref_sel = ADC_REFERENCE_AVDD;
cfg.channels = 0x03;
cfg.route.pad_to_sample = 1;
cfg.clk_sel = ADC_SAMPLE_CLK_64K_DIV13;

```



```

cfg.clk_div = 0x1f;
adc_init(&cfg);
adc_enable(NULL, NULL, 0);
co_delay_100us(20); // delay 2ms
adc_get_result(ADC_TRANS_SOURCE_PAD, 0x03, result);
ref_vol = adc_get_ref_voltage(ADC_REFERENCE_AVDD);
voltage_vol1 = (result[0] * ref_vol) / 1024;
voltage_vol2 = (result[1] * ref_vol) / 1024;

```

## 2.4 ADC 中断采集

ADC 中断方式采集数据，只能用于单通道采集，需要设置回调函数和缓存 buff，当采集到设置的个数之后会调用回调。

```

#include "driver_adc.h"
uint16_t adc_buff[5];
__attribute__((section("ram_code"))) void adc_isr_callback(uint16_t * buff, uint32_t len)
{
    NVIC_DisableIRQ(ADC_IRQn);
    co_printf("adc value:\r\n");
    for(uint8_t i = 0; i < len; i++)
    {
        co_printf("%d ", buff[i]);
    }
}
void adc_isr_test_init(void)
{
    struct adc_cfg_t cfg;
    uint16_t result;
    system_set_port_mux(GPIO_PORT_D, GPIO_BIT_4, PORTD4_FUNC_ADC0);
    memset((void*)&cfg, 0, sizeof(cfg));
    cfg.src = ADC_TRANS_SOURCE_PAD;
    cfg.ref_sel = ADC_REFERENCE_AVDD;
    cfg.channels = 0x01;
    cfg.route.pad_to_sample = 1;
    cfg.clk_sel = ADC_SAMPLE_CLK_24M_DIV13;
    cfg.clk_div = 0x3f;
    adc_init(&cfg);
    adc_enable(adc_isr_callback, adc_buff, 5);
    // adc_get_result(ADC_TRANS_SOURCE_PAD, 0x01, &result);
    NVIC_EnableIRQ(ADC_IRQn);
}

```

说明: `adc_isr_callback`为回调接口, 回调是在中断里面调用的, 所以需要加上`ram_code`的修饰。  
`adc_buff`为缓存buff和缓存个数。

FreqChip Confidential

### 3. 硬件参考

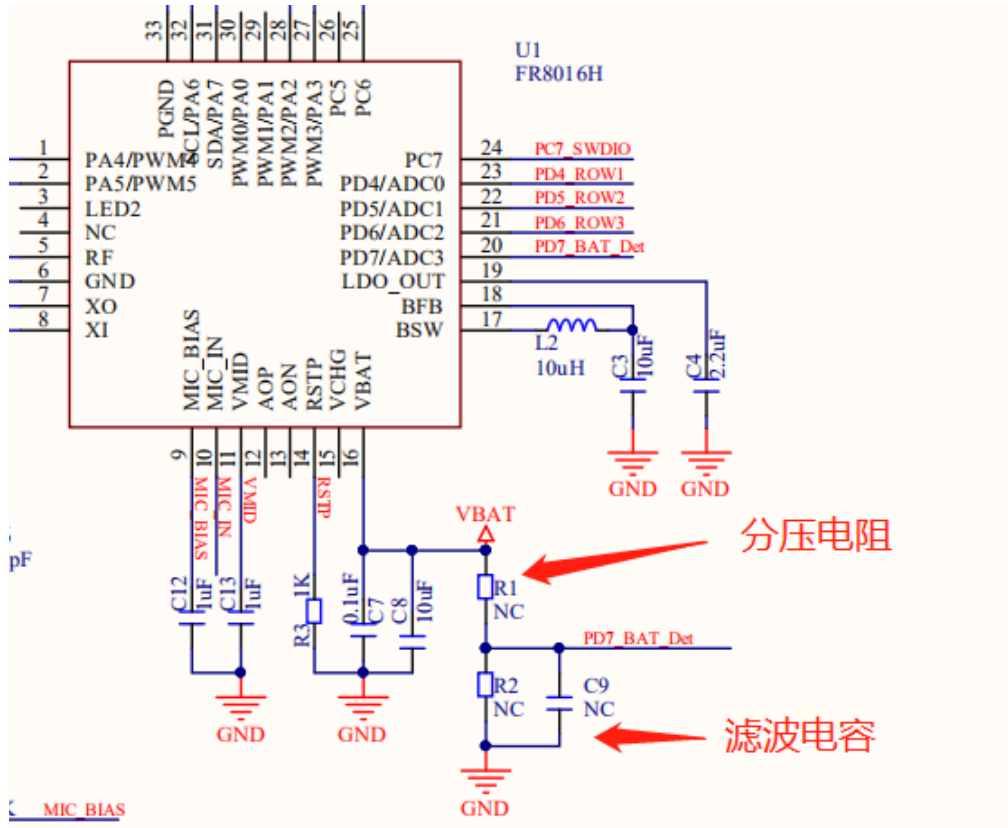


图 3-1 硬件参考图

## 4. ADC 采集问题分析参考

以上软硬件处理之后，如果采样还是不太稳定，则需要分析下采样电压是否稳定，如果配置参考源是 AVDD 时，外围是否大的负载或者干扰，导致 AVDD 波动。一般只要配置没问题，采样有波动的话基本都是采样电压或者采样基准源有波动导致。所有软硬件都没问题的话采样也不会有问题。

FreqChip Confidential

## 联系方式

欢迎大家针对富芮坤产品和文档提出建议。

反馈: [doc@freqchip.com](mailto:doc@freqchip.com)

网站: [www.freqchip.com](http://www.freqchip.com)

销售: [sales@freqchip.com](mailto:sales@freqchip.com)

电话: +86-21-5027-0080

FreqChip Confidential