



YC31xx ADC 应用说明

V1.1

Yichip Microelectronics

©2014

Revision History

Version	Date	Author	Description
V1.0	2020-02-14	Zhiteng.Yi	Initial version
V1.1	2020-03-01	Zhiteng.Yi	Add lib description

Confidentiality Level:**confidential**

目录

1.	ADC 简介.....	4
2.	库函数.....	4
2.1.	宏.....	4
2.2.	参数类型说明.....	4
2.3.	初始化函数.....	5
2.4.	读 ADC 寄存器值.....	5
2.5.	获取中位平均值.....	5
2.6.	获取电压.....	6
3.	ADC 采集锂电池电压.....	6

1. ADC 简介

最高采样率为 600KHz，最高采样精度为 10 比特，电压校准值在量产芯片中会烧录到 OTP 中。
在 GPIO 模式下，ADC 有 8 个通道，对应 GPIO 为 GPIO37 到 GPIO44，测量电压范围为 0 到 1.2V；
在 HVIN 模式下，测量电压范围为 0-5V，精度平均在 30mV 以内，最大误差不超过 50mV。

2. 库函数

2.1. 宏

```
#define IO_LOW_VOLTAGE_REF      500//GPIO 的参考低电压为 0.5v
#define IO_HIHG_VOLTAGE_REF    1000// GPIO 的参考高电压为 1v

#define HVIN_LOW_VOLTAGE_REF    3300 // HVIN 的参考低电压为 3.3v
#define HVIN_HIHG_VOLTAGE_REF  5000 // HVIN 的参考低电压为 5v

#define ADC_REF_OTP_ADDR 0x012c //参考电压对应值烧录的 otp 位置

#define FILT_NUM      20//中位平均数采样数
#define REMOVE_NUM    6//中位平均数去除个数

#define WAIT_TIME 5000 //采样等待时间
```

2.2. 参数类型说明

```
typedef struct _ADC_InitTypeDef
{
    ADC_ChxTypeDef  ADC_Channel;
    ADC_ModeTypeDef ADC_Mode;
} ADC_InitTypeDef;
ADC_Channel: 通道选择为以下枚举类型
typedef enum
{
    ADC_CHANNEL_0 = 0,
    ADC_CHANNEL_1,
    ADC_CHANNEL_2,
```

```
ADC_CHANNEL_3,  
ADC_CHANNEL_4,  
ADC_CHANNEL_5,  
ADC_CHANNEL_6,  
ADC_CHANNEL_7,  
}ADC_ChxTypeDef;  
typedef enum  
{  
    ADC_GPIO = 0,  
    ADC_DIFF,  
    ADC_HVIN,  
    ADC_VINLPM,  
}ADC_ModeTypeDef;
```

2.3. 初始化函数

- 函数原型：void ADC_Init(ADC_InitTypeDef *ADC_InitStruct)
- 说明：初始化 ADC 模块，配置模式，选择通道
- 参数：ADC_InitStruct
- 返回值：无

2.4. 读 ADC 寄存器值

- 函数原型：uint16_t ADC_GetResult(ADC_ChxTypeDef ADC_Channel)
- 说明：获取当前模式，当前通道电压对应的寄存器读数
- 参数：采电池电压的模式下配置成 ADC_HVIN, 通道选择 ADC_CHANNEL_0; ADC_GPIO 模式下，可选通道 0 到通道 7，对应对应 GPIO 为 GPIO37 到 GPIO44，差分模式(ADC_DIFF)下,选择 ADC_CHANNEL_0 到 ADC_CHANNEL_3, GPIO37 GPIO38 对应 ADC_CHANNEL_0。两个通道一组。
- 返回值：当前通道电压对应的寄存器读数

2.5. 获取中位平均值

- 函数原型：int ADC_Filt(ADC_ChxTypeDef ADC_Channel)
- 说明：获取 ADC_GetResult（）中位平均值；取 FILT_NUM（默认 20）个去掉最大值最小值共 REMOVE_NUM（6）个。
- 参数：采电池电压的模式下配置成 ADC_HVIN, 通道选择 ADC_CHANNEL_0; ADC_GPIO 模式下，可选通道 0 到通道 7，对应对应 GPIO 为 GPIO37 到 GPIO44，差分模式(ADC_DIFF)下,选择 ADC_CHANNEL_0 到 ADC_CHANNEL_3, GPIO37 GPIO38 对应 ADC_CHANNEL_0。两个通道一组。
- 返回值：当前通道电压对应的寄存器读数的中位平均值

2.6. 获取电压

- 函数原型 : `int ADC_GetVoltage(ADC_ChxTypeDef ADC_Channel)`
- 说明: 获取当前模式 (HVIN / GPIO / ADC_DIFF), 当前通道电压(mV).
- 参数: 采电池电压的模式下配置成 ADC_HVIN (芯片 hvin 管脚), 通道选择 ADC_CHANNEL_0; ADC_GPIO 模式下, 可选通道0 到通道7, 对应对应GPIO为GPIO37到GPIO44, 差分模式(ADC_DIFF)下, 选择 ADC_CHANNEL_0 到 ADC_CHANNEL_3, GPIO37 GPIO38 对应 ADC_CHANNEL_0。两个通道一组。
- 返回值:
 - ◆ 当前电压值 (HVIN 模式 : 0 ~ 5500、GPIO / ADC_DIFF 模式 : 0 ~ 1200)
 - ◆ -1 : 模数转换异常

3. ADC 采集锂电池电压

在 Pos 应用中, 电路上 HVIN 脚与锂电池相连, 即可让 ADC 模块在 HVIN 模式下采集电压。

1. ADC 的配置,

```
void ADC_Configuration(void)
{
    //定义 ADC 配置结构体
    ADC_InitTypeDef ADCInitStruct;
    //选择通道, HVIN 模式下只有一个通道
    ADCInitStruct.ADC_Channel = ADC_CHANNEL_0;
    //模式为 HVIN 模式
    ADCInitStruct.ADC_Mode = ADC_HVIN;
    ADC_Init(&ADCInitStruct);
}
```

2. 采集的值读取, 以下接口返回值为采集的 ADC 值

```
ADC_GetResult (ADC_CHANNEL_0);
```

3. 电压计算, 电压是根据采集的值求中位平均值和校准值计算得出, 以下接口返回值为计算的电压值单位 (mV)

```
ADC_GetVoltage (ADC_CHANNEL_0);
```