

摘要

本技术文档旨在帮助客户快速实现估算 MCU F03X VDDA 电压。在使用锂电池的应用场景下，整个产品生命周期中电池的电压是逐步降低的，如果应用当中没有一个稳定的参考电压源，ADC 采样的值也会有偏差。本文档提供一个估算 VDDA 的方法，可以实现 VDDA 2.0~ 5.5V 电压区间较为准确的估算。

版本

历史版本	修改内容	日期
V1.0	初版生成	2022-06-20

目录

目录.....	2
1 硬件介绍.....	3
2 估算方法.....	3
3 软件参考代码.....	5
4 实际测试.....	6
5 总结.....	7

1 硬件介绍

评估板： CS32F035 EVB
 调试器： Jlink
 电源： 可调直流电源
 串口工具：



图1 CS32F035 评估板



图2 调试器 Jlink

2 估算方法

手册提供了一个参考计算公式，基于芯片内部的参考电压值，可以通过读取 ADIN_17 通道的 ADC 采样值，再通过计算，估算到 VDDA 的实际电压值。计算公式如下：

内部参考电压 Vrefint 为 ADC 提供一个基准电压输出。Vrefint 接到 ADIN_17 输入通道。每颗芯片的 Vrefint 电压在量产测试时被测量并存储在系统存储区，它是只读的。

使用内部参考电压计算实际的 VDDA 电压

芯片的 VDDA 电源电压可能会发生变化或者不能被准确知道。在 VDDA=3.3V 条件下，内部参考电压(V_{REFINT})的校准数据在制造过程中被获取，可以用来计算实际的 VDDA 电压。

下列公式给出了芯片的真实 VDDA 电压：

$$V_{DDA} = \frac{3.3V \times VREFINT_CAL}{VREFINT_DATA}$$

其中：

- VREFINT_CAL 是 V_{REFINT} 的校准值

图 20 ADC 模块框图

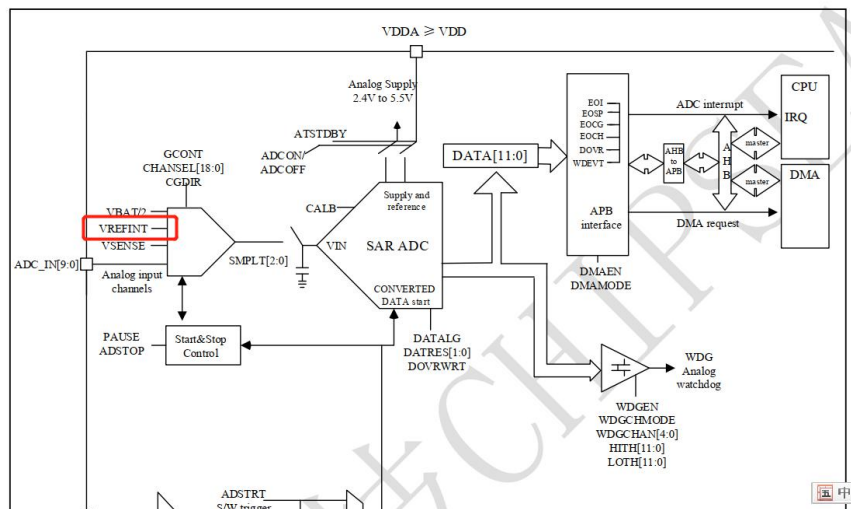


图 3 ADC 模块中 通道 17 连到 Vrefint 通道

- 模拟输入通道
 - 10 个外部模拟输入
 - 1 个内部温度传感器检测通道(V_{SENSE})
 - 1 个内部参考电压检测通道(V_{REFINT})
 - 1 个 VBAT 引脚电池电压检测通道

图 4 MCU 有一个内部参考电压检测通道

通道选择 (CHANSEL, CGDIR)

ADC 采样通道包括:

- 10 个 GPIO 引脚模拟输入 (ADC_IN0~ADC_IN9)
- 3 个内部的模拟输入(温度传感器 ADC_IN16、内部参考电压 ADC_IN17、V_{BAT} 通道 ADC_IN18)
- 6 个通道固定接地(ADC_IN10~ADC_IN15)

3 软件参考代码

代码主要有两个部分，一是配置 ADIN_17 通道，二是计算并输出 VDDA 的估算值。

ADC 配置代码如下：

```
adc_def_init(ADC1);
adc_config_struct_init(&adc_config_struct);

// Configure the ADC1 in continuous mode
adc_config_struct.adc_resolution = ADC_CONV_RES_12BITS;
adc_config_struct.conversion_mode = ENABLE;
adc_config_struct.trigger_mode = ADC_TRIG_MODE_SEL_NONE;
adc_config_struct.data_align = ADC_DATA_ALIGN_RIGHT;
adc_config_struct.scan_direction = ADC_CONV_SEQ_DIR_UPWARD;
adc_init(ADC1, &adc_config_struct);

// Set the ADC1 CH0 with 239.5 Cycles
adc_channel_config(ADC1, ADC_CHANSEL_CHANSEL17, ADC_SAMPLE_TIMES_239_5); // ADC 采样通道配置

ADC1->CTR = ADC1->CTR | 0X01;
ADC->INNCHEN = ADC->INNCHEN | 0x400000; // 0x800000;

adc_calibration_value_get(ADC1); // ADC Calibration.
adc_enable_ctrl(ADC1, ENABLE); // Enable the ADC.

while(!adc_flag_status_get(ADC1, ADC_FLAG_EOI)); // Wait the EOI flag.
adc_conversion_start(ADC1); //ADC1 regular Software Start Conv.
```

VDDA 估算，计算 10 次平均值并打印到串

```
uint32_t get_avg(void)
{
    double vdda;
    uint32_t sample_value;
    uint32_t i;

    for(i=0; i<10; i++)
    {
        while(adc_flag_status_get(ADC1, ADC_FLAG_EOCH) == RESET); // Check EOC flag
        sample_value = adc_conversion_value_get(ADC1);

        delay_simple1();
        vdda = (3.3*ADC_REF)/sample_value;
        ADC_REF1[i] = vdda;
        printf("Avg_VDDA = %f\n\r", vdda);
    }

    VDDA_VALUE = (ADC_REF1[0] + ADC_REF1[1] + ADC_REF1[2] + ADC_REF1[3] + ADC_REF1[4] + ADC_REF1[5] +
    ADC_REF1[6] + ADC_REF1[7] + ADC_REF1[8] + ADC_REF1[9])/10;
    printf("Avg_VDDA_VALUE = %f\n\r", VDDA_VALUE);
}
```

4 实际测试

用一个可调直流电源给开发板 MCU VDD/VDDA 供电，从 2.00V 开始增加电压，观察串口输出信息（9600，PA9）并记录估算数据如图 5。

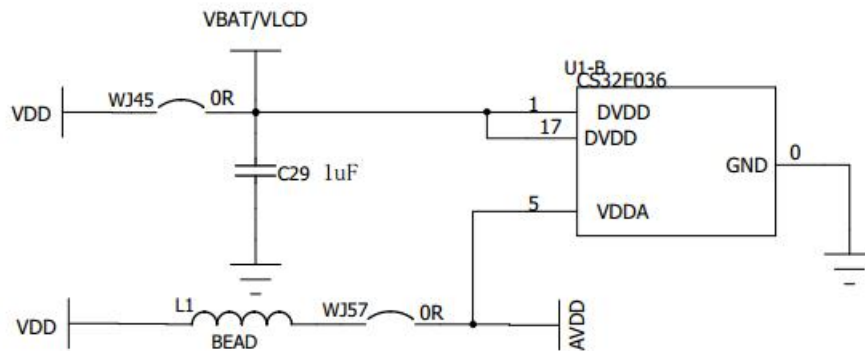


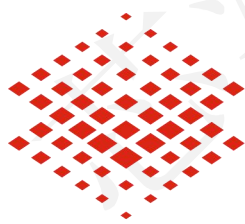
图 4 开发板中 VDDA VDD 是通过一个电感连在一起的

	VDD 电压值 V	估算的 VDDA 电压值 V	精度
1	2.00	2.0029	0.14%
2	2.20	2.2017	0.08%
3	2.40	2.4028	0.12%
4	2.60	2.6096	0.37%
5	2.80	2.8136	0.49%
6	3.00	3.0146	0.49%
7	3.30	3.3177	0.54%
8	3.50	3.5232	0.66%
9	3.80	3.8270	0.71%
10	4.00	4.0339	0.85%
11	4.30	4.3396	0.92%
12	4.50	4.5449	1.00%
13	4.70	4.7509	1.08%
14	5.00	5.0611	1.22%
15	5.20	5.2698	1.34%
16	5.40	5.4481	0.89%
17	5.50	5.5535	0.97%

图 5 VDDA 实际值和估算值的

5 总结

在锂电池供电应用场景下，如果没有一个稳定的参考电压源，可以通过读取芯片内部 Vrefint 值和 ADC 通道 ADIN_17，再经过计算，有效的得到了 VDDA 的估算值，总体精度在 1% 左右。通过 VDDA 的值，可以较为准确的计算出 ADC 实际电压值。



芯海科技
CHIPSEA

股票代码:688595

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。