

# Synwit-ADC 测试板 应用笔记

V1.0

## 文档说明

|             |                        |
|-------------|------------------------|
| <b>摘要</b>   | 本文档针对 SWM181 系列 ADC 说明 |
| <b>应用场景</b> | 适用于 SWM181 系列          |

# 目 录

|        |                |           |
|--------|----------------|-----------|
| 1      | 简介             | 3         |
| 1.1.   | SD-ADC 说明      | 3         |
| 1.1.1. | 特性             | 3         |
| 1.1.2. | 时钟控制           | 3         |
| 1.1.1. | SD-ADC 模式配置    | 4         |
| 1.1.1. | SD-ADC 输入范围    | 4         |
| 1.1.1. | SD-ADC 偏移校准    | 4         |
| 1.1.1. | SD-ADC 增益校准    | 5         |
| 1.1.1. | SD-ADC 阻抗说明    | 6         |
| 1.1.   | SAR-ADC 功能特性   | 8         |
| 1.2.   | 测试电路板实物图       | 8         |
| 2.     | 电路板组成与使用方法     | 9         |
| 2.1.1. | 串电阻分压电压测量      | 9         |
| 2.1.2. | 串电阻 I-V 转换电流测量 | 10        |
| 2.1.3. | 基准源选择          | 10        |
| 2.1.4. | ADC 选择         | 11        |
| 2.1.5. | 单片机最小系统        | 11        |
| 2.1.6. | OLED 屏幕显示      | 11        |
| 3.     | 操作说明           | 12        |
| 3.1.1. | SARADC         | 12        |
| 3.1.2. | SDADC          | 13        |
| 3.1.3. | 硬件选择           | 14        |
| 4.     | 测试结果           | 15        |
| 5.     | 应用程序示例         | 15        |
| 5.1.1. | 温度测量           | 15        |
| 5.1.2. | 压力测量           | 错误!未定义书签。 |
| 6.     | 修订记录           | 16        |

# 1 简介

测试板演示了 SWM181 的 SD-ADC (Sigma-delta AD) 和 SAR-ADC 使用方法以及精度特性。

由于 SDADC 的特性, SDADC 每次启动采样得到的前八个数据为上一次采样结果的滞后, 使用时建议将前 8 位数据丢弃。

## 1.1.SD-ADC 说明

### 1.1.1. 特性

- 16bit 分辨率
- 6 组差分输入单端输入, 或者它们的组合
- 高性能的数据处理
  - 多通道采样速率 16.6ksps
  - 单通道采样速率 50ksps
- 自校正功能
- 7 种可编程增益:  $\times 0.5$ 、 $\times 1$ 、 $\times 2$ 、 $\times 4$ 、 $\times 8$ 、 $\times 16$  及  $\times 32$
- 支持差分模式和单端模式
- 可配置 3 种配置供不同通道使用
- 输入通道的模拟增益
- 转换模式: 差分模式/单端模式
- 校正偏差
- 参考电压的选择
  - SDADC\_REFP:  $1.1\text{V} \sim 3.6\text{V}$
- 触发源
  - 软件触发
  - 定时器触发
- 转换结束中断, FIFO 数据溢出中断, FIFO 半满中断, FIFO 数据满
- 支持 DMA 读取

### 1.1.2. 时钟控制

SDADC 有两个时钟源:

- 内部高频 RC 振荡器的 8 分频 (RCHF/8)
- 外部高频晶振的 8 分频 (XTAH/8)

SD ADC 最高工作频率为 6MHZ, 最低为 500KHZ, 默认工作在全速模式, 工作时钟最高为 6MHZ。

可切换成低速模式, 获得功耗减半的效果。低速模式下 SDADC 最高工作在 1.5MHZ 下, 最低为 500KHZ, 此时功耗最小, 工作电压可以降到 2.2V。

输入时钟可以和系统时钟使用不同的时钟源，当需要进行精度较高的采样时，建议使用外部时钟源。

### 1.1.1. SD-ADC 模式配置

SD ADC 同时支持三种模式配置：

- 差分输入
- 单端输入
  - 通道的 N 输入脚作为 N 输入端，需要接 GND
  - 公共的 AINREFM 引脚作为 N 输入端，已在片内接 GND

*注：SDADC 无法测量负电压，并且每个通道的输入电压都必须在器件的电气极限之内。*

3 组配置寄存器用于指定通道对应的模式。6 个通道可以选择指定 3 种模式中的任意一种，在 CFGS 寄存器进行配置保存。

### 1.1.1. SD-ADC 输入范围

输入引脚上的模拟电压 (SDADCx\_AIN[5:0]P, SDADCx\_AIN[5:0]N) 必须在 SDADC 电源范围内 (AVSS1, AVDD1)，SDADC 输出数据范围对应的输入模拟电压取决于测量模式的配置 (差分模式和单端模式)，增益通道的选择：

- 差分模式：VIN 范围在 SDADCx\_AIN[5:0]P 和 SDADCx\_AIN[5:0]N 之间

$$VIN = \frac{-SDADC\_REFP}{2*gain} \sim \frac{SDADC\_REFP}{2*gain}$$

输出转换数据为 -32768 ~ 32767

- 单端模式：VIN 范围在 SDADCx\_AIN[5:0]P 和 GND 之间

$$VIN = GND \sim \frac{SDADC\_REFP}{gain}$$

-32768 ~ 32767，软件中执行转换结果+32768，将数据范围转换到 0~65535

### 1.1.1. SD-ADC 偏移校准

SDADC 提供了自动偏移校准，不需增加外部元件。如果使能了校准通道，校准结果产生后将存入到 OFFSET 位。

校正功能可以用来确定当前 SDADC 的配置所产生的 OFFSET 值，即 CFGx 寄存器 BIT0~BIT11。OFFSET 值取决于 CFGx 寄存器的配置情况 (包括 SE、REFM、GAIN)。校正 OFFSET 期间，SDADC 会连接到 CTRL 寄存器 CALIN 位给定的电压，选择 CFGx 寄存器 GAIN 位，并执行一次转换，以确定 OFFSET 的值。

*说明：校正成功后，需要恢复校正之前 CTRL 寄存器的值，因此校正之前需要定义变量存储 CTRL 寄存器原始的数据，以备恢复时使用。*

偏移误差为常量，会加至理想转换值之上。下图显示了偏移误差

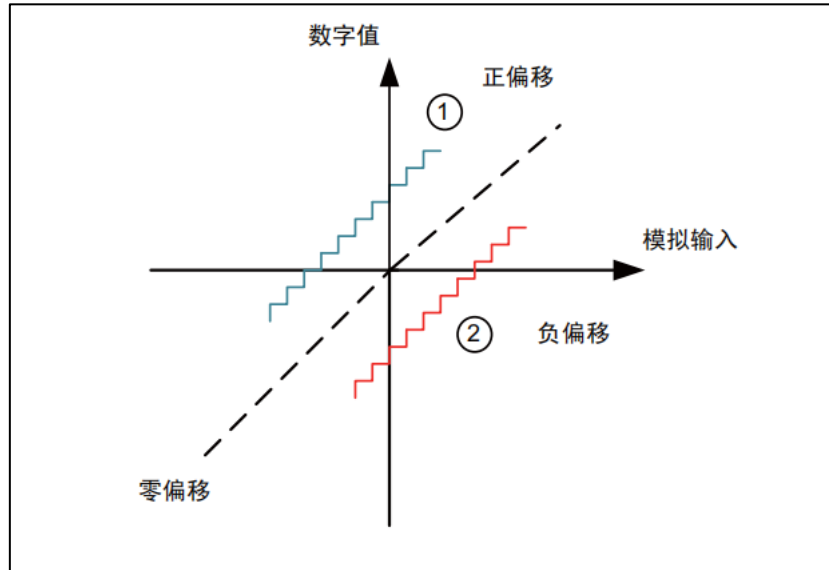


图 1 SDADC 中的偏移误差

### 1.1.1. SD-ADC 增益校准

增益误差为 SDADC 传递函数到与理想直线之间的偏离。它是由内建的可编程放大器引起的。根据器件的数据手册，典型的模拟增益误差为-2.7%，这意味着增益误差可达到 884（最大电压时）。因此，当执行精确测量时，必须降低该增益误差。

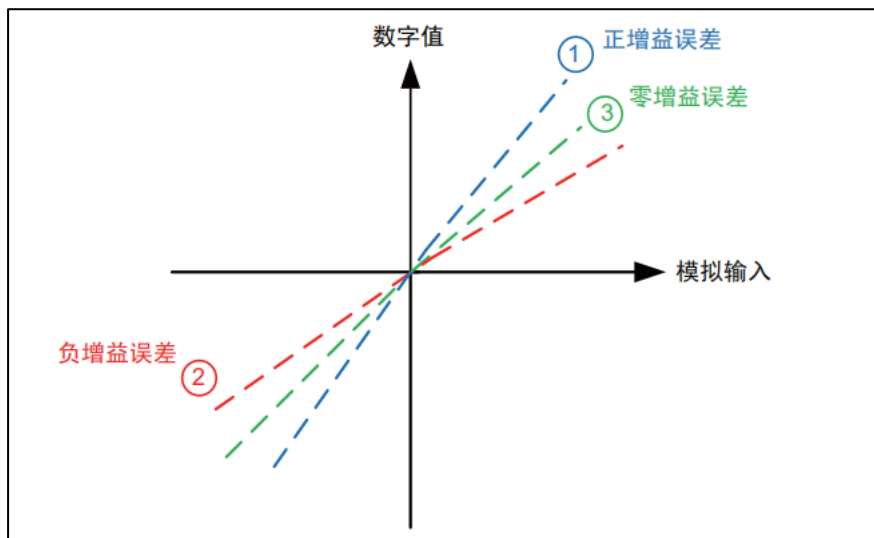


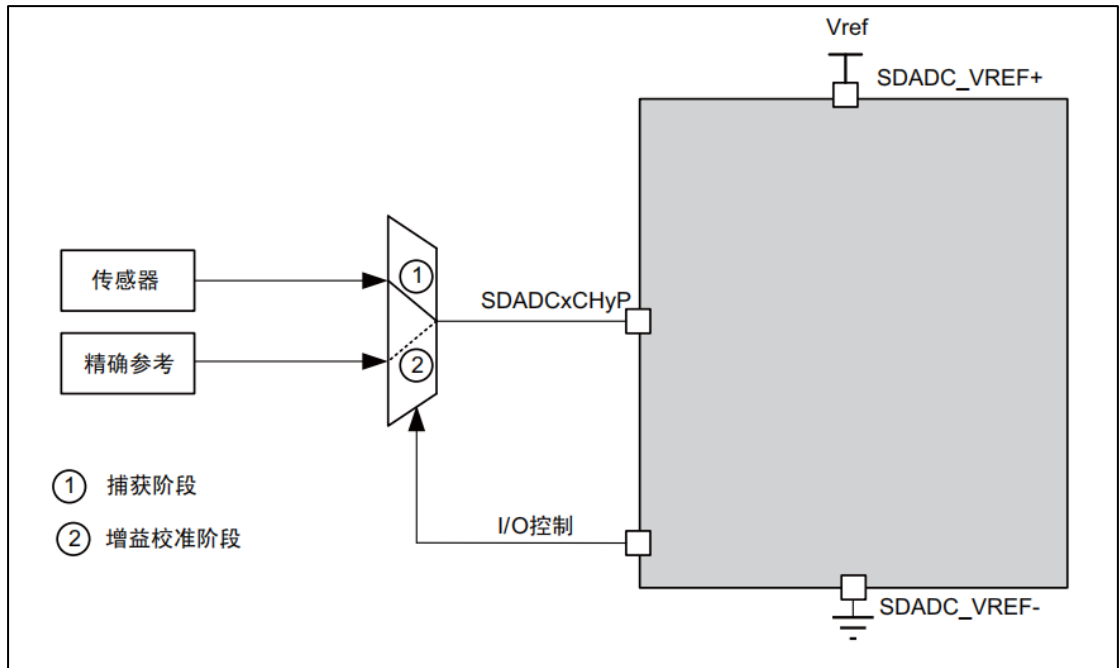
图 2 SDADC 中的增益误差

根据输入信号大小，通过 GAIN 位调节增益值，放大输入信号。包括：x1/2、x1、x2、x4、x8、x16 和 x32。

增益校准时只考虑 x1/2、x1、x2、x4、x8。

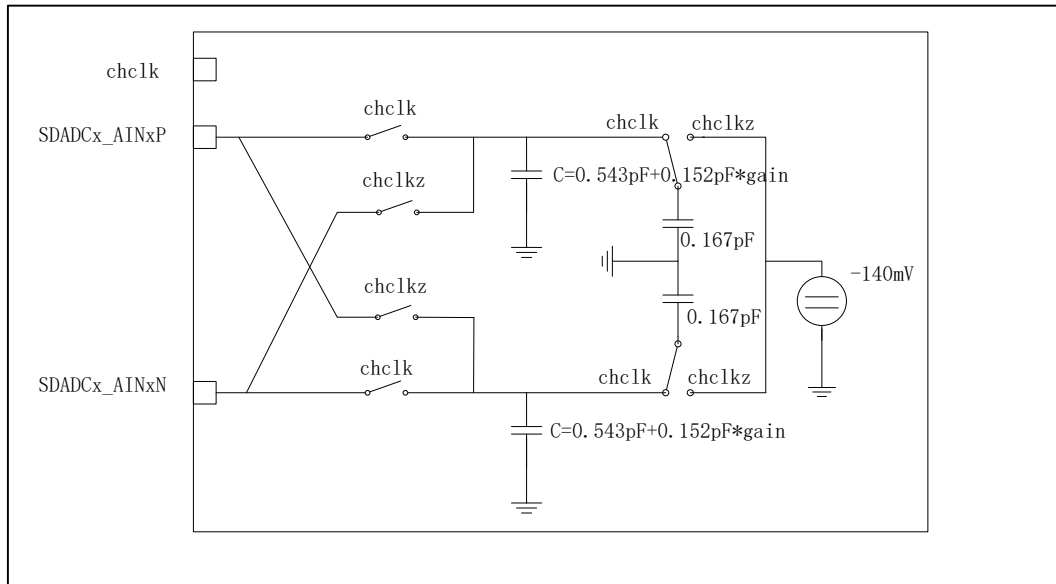
增益校准需要一个外部精确参考，做法是在 SDADC 输入施加精确参考电压 (AccRef)，然后得到 SDADC 输出。增益按下式计算： $\text{AccRef} / (\text{输出} \times V_{\text{ref}} / 65535)$  算出的增益可存储于非易失性存储器（闪存）中，在捕获阶段使用。

注：在运行增益校准前，必须运行偏移校准。



### 1.1.1. SD-ADC 阻抗说明

SDADC 输入阻抗取决于 SDADC 时钟与增益 (1/2x 1x 2x 4x 8x 16x 32x)。输入通道阻抗的等效电路如下：



阻抗计算公式为：
$$R_{in} = \frac{1}{2 \cdot F_{CLK} \cdot C}$$

外部 VREFP 的输入等效电路如图所示，使用外部 VREFP 时需在 VREFP 与 AVSS 间加 1uF 电容。

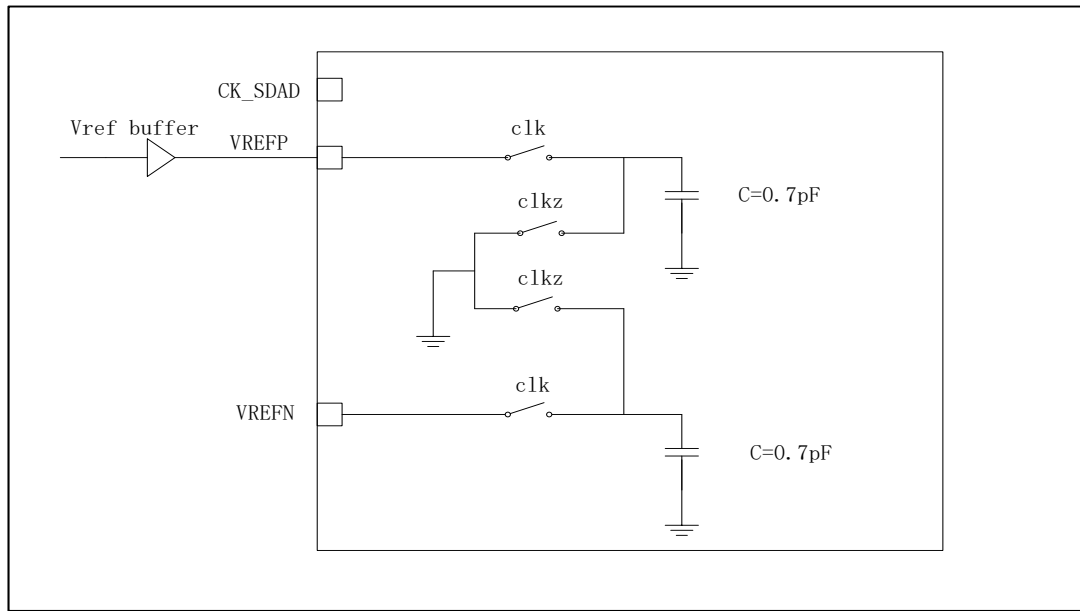


图 5 REFP 阻抗示意图

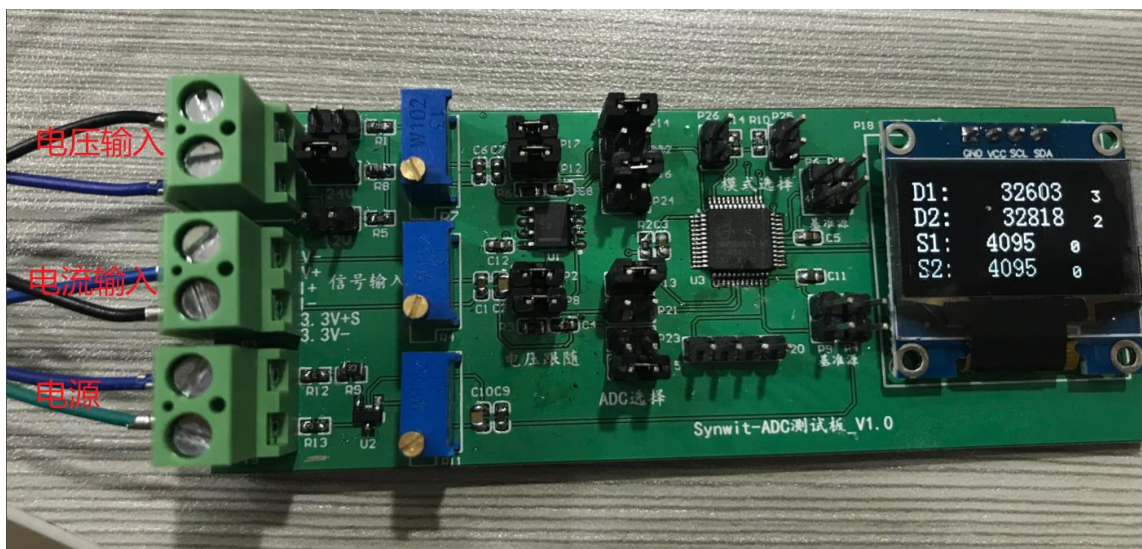
阻抗计算公式为: 
$$R_{in} = \frac{1}{F_{CLK} \cdot C}$$

## 1.1.SAR-ADC 功能特性

- 12-bits 分辨率
- 最多 8 路输入通道
- 最高 1MSPS 转换速率
- 支持单次模式和连续模式
- 具备深度为 8 的 FIFO
- 灵活的转换启动方式，支持软件、PWM、TIMER 启动
- 每个通道都有自己独立的转换结果数据寄存器和转换完成、数据溢出状态寄存器
- 支持 DMA 传输

## 1.2.测试电路板实物图

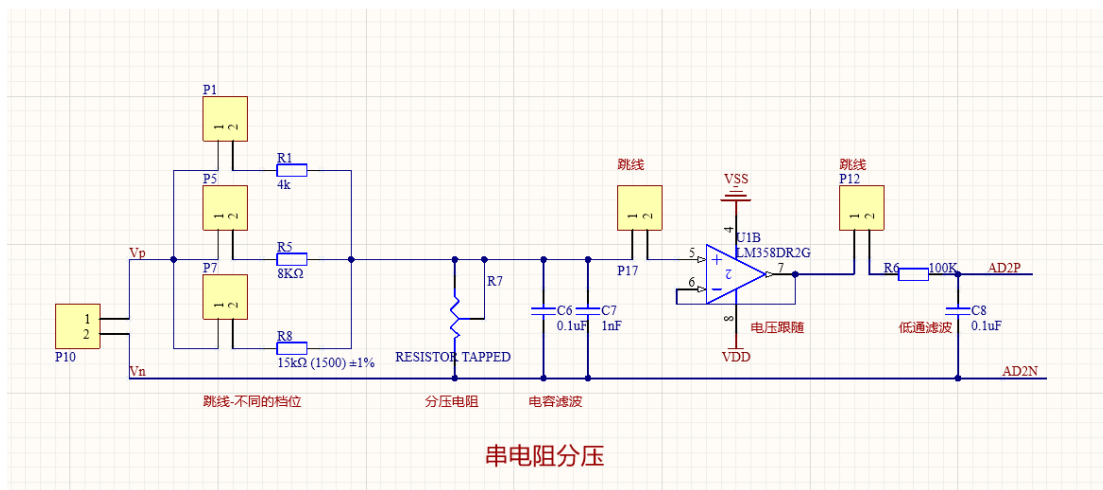
照片如下





## 2. 电路板组成与使用方法

### 2.1.1. 串电阻分压电压测量



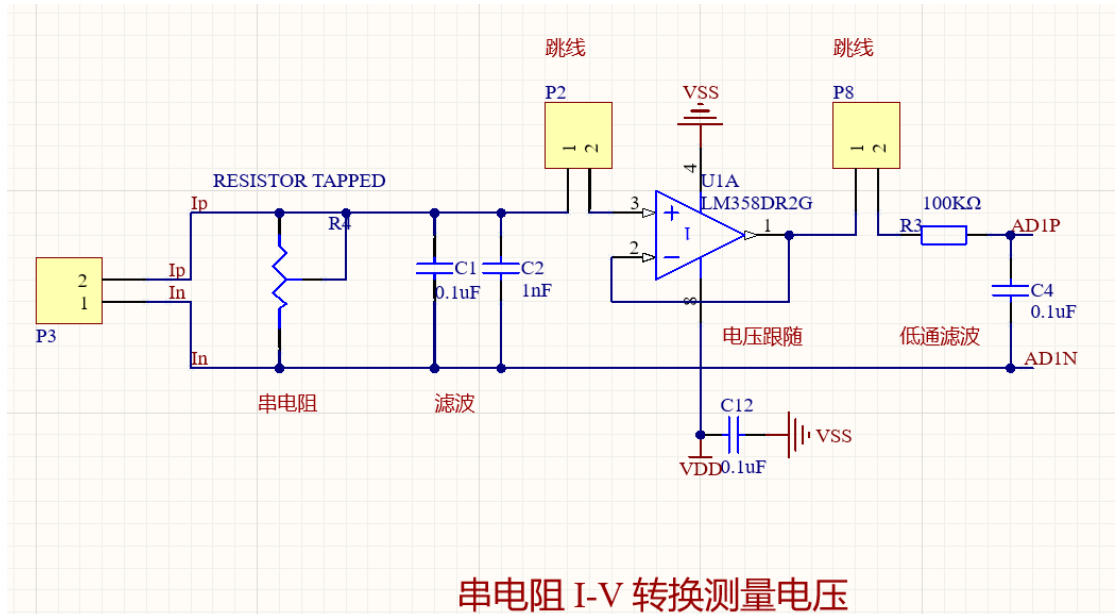
如图，P10 为信号输入；P1、P5、P7 短接端子，短接不同的端子，搭配滑动变阻器 R7 的不同阻值，可选择不同的电压分压比从而选择不同的电压量程。

C6、C7 为 0.1μF 和 1nF 滤波电容，对信号进行初步的滤波。

U1B 为电压跟随电路，短接 P17 的 1 和 2，短接 P12 的 1 和 2，则信号经过电压跟随电路后传到后级电路；若短接 P17 的 1 和 P12 的 2，则信号不经过电压跟随电路直接传给后级电路。

R6 和 C8 组合 RC 低通滤波电路，对信号进行滤波。

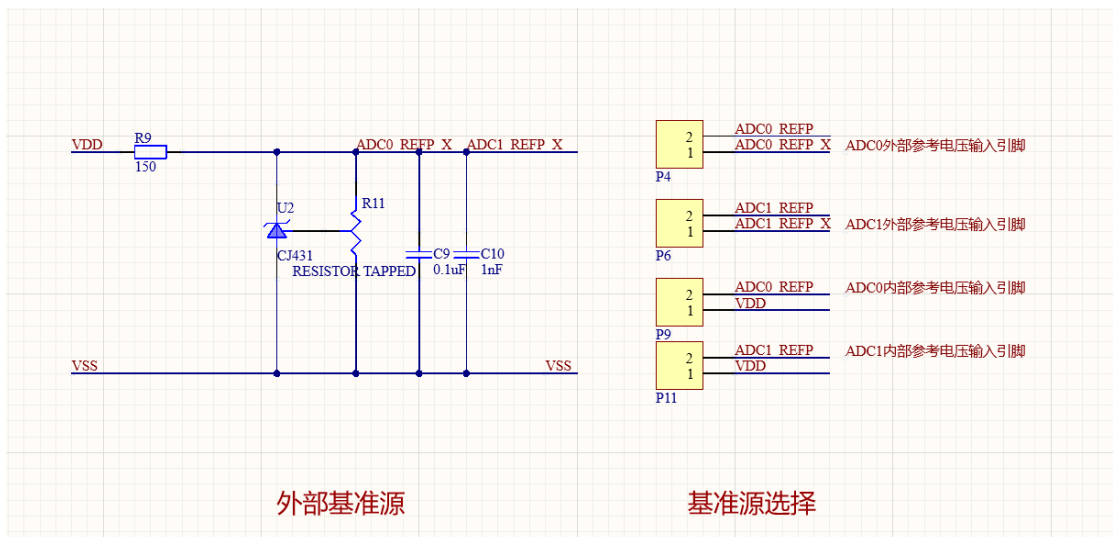
## 2.1.2. 串电阻 I-V 转换电流测量



串电阻 I-V 转换测量电压

如图，P3 为电流信号输入端，信号进入电路板经过 R4 转换为电压信号，然后经过 C1、C2 电容初级滤波，P2、P8、U1A 组成电压跟随电路，用法如上。

## 2.1.3. 基准源选择

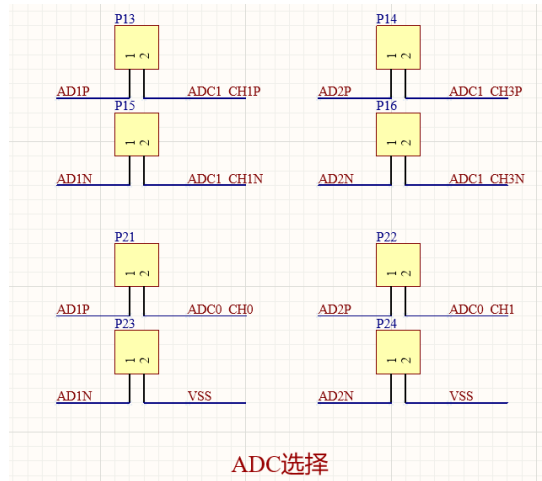


基准源有两种：外部基准、VCC 电压。

如图，外部基准由 CJ431 提供。

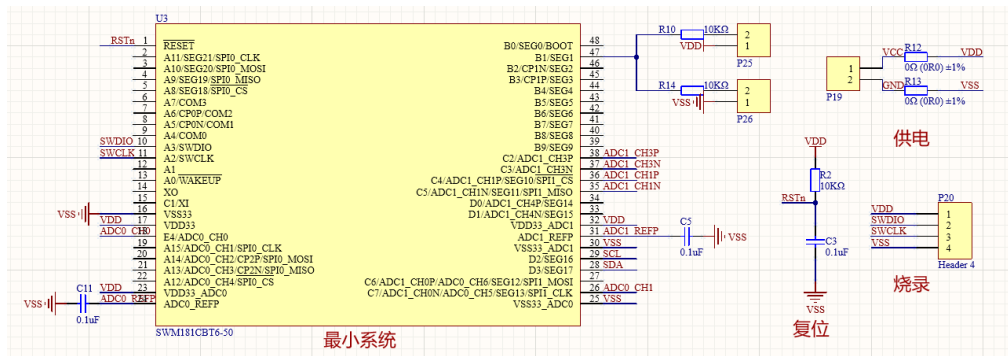
分别短接 P4、P6、P9、P10 可让 SARADC 和 SDADC 选择不同的参考电压。

## 2.1.4. ADC 选择



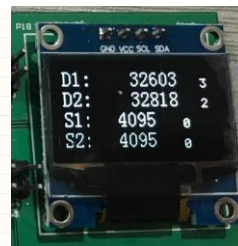
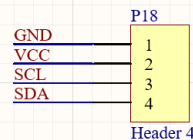
如图，短接不同的端子，可选择信号进入不同的 ADC 测量通道，配合不同的软件配置，可测试不同的 ADC 精度情况。

## 2.1.5. 单片机最小系统



如图，为单片机最小系统，以及单片机的 ADC 接口。

## 2.1.6. OLED 屏幕显示



如图，板上搭载了一个 IIC 接口的 0.96 寸 OLED 显示屏，显示不同 ADC 通道数值，后面小字体为最近 1S 内 adc 数值变化量。

## 3. 操作说明

### 3.1.1. SARADC

```
void My_SARADC_Init(void)
{
    ADC_InitStructure ADC_initStruct;

    PORT_Init(PORTE, PIN4, PORTE_PIN4_ADC_CH0, 0); //PE.4 => ADC.CH0
    PORT_Init(PORTC, PIN7, PORTC_PIN7_ADC_CH5, 0); //PC.7 => ADC.CH5

    ADC_initStruct.clk_src = ADC_CLKSRC_HRC_DIV4; //时钟选择
    ADC_initStruct.channels = ADC_CH0|ADC_CH5; //通道选择
    ADC_initStruct.samplAvg = ADC_AVG_SAMPLE1; //采样平均
    ADC_initStruct.trig_src = ADC_TRIGSRC_SW; //触发方式
    ADC_initStruct.Continue = 0; //非连续模式，即单次模式
    ADC_initStruct.EOC_IEN = 0; //中断
    ADC_initStruct.OVF_IEN = 0; //配置ADC
    ADC_Init(ADC, &ADC_initStruct);

    ADC_Open(ADC); //使能ADC
}
```

如图，为 SARADC 配置代码，芯片的基准源由外部跳线选择。

```
void My_Get_SARADC(void)
{
    /*获取ADC数值*/
    ADC_Start(ADC);
    while((ADC->CH[0].STAT & ADC_STAT_EOC_Msk) == 0);
    ADC->CH[0].STAT = (1 << ADC_STAT_EOC_Pos);
    ADCDATA0 = ADC_Read(ADC, ADC_CH0);
    OLED_ShowNum(20,4,ADCDATA0,7,16);

    while((ADC->CH[5].STAT & ADC_STAT_EOC_Msk) == 0);
    ADC->CH[5].STAT = (1 << ADC_STAT_EOC_Pos);
    ADCDATA1 = ADC_Read(ADC, ADC_CH5);
    OLED_ShowNum(20,6,ADCDATA1,7,16);
    ADC_Stop(ADC);

    /*初始化*/
    if(AdcGetNum == 1)//初始化
    {
        ADC0_MAX = ADC0_MIN = ADCDATA0;
        ADC1_MAX = ADC1_MIN = ADCDATA1;
    }

    /*ADC波动*/
    if(ADCDATA0 > ADC0_MAX) ADC0_MAX = ADCDATA0;
    if(ADCDATA0 < ADC0_MIN) ADC0_MIN = ADCDATA0;

    if(ADCDATA1 > ADC1_MAX) ADC1_MAX = ADCDATA1;
    if(ADCDATA1 < ADC1_MIN) ADC1_MIN = ADCDATA1;

    /*显示出来*/
    if(AdcGetNum == 0xff)//初始化
    {
        OLED_ShowNum(80,5,(ADC0_MAX - ADC0_MIN),4,12);
        OLED_ShowNum(80,7,(ADC1_MAX - ADC1_MIN),4,12);
        AdcGetNum = 0;//清零
    }
    AdcGetNum++;
}
```

如图，为 SARADC 的数据处理。

## 3.1.2. SDADC

```

SDADC_InitStructure SDADC_initStruct;
SDadcGetNum = 0;
GPIO_Init(GPIOC, PIN2, 0, 0, 0, 1);
GPIO_Init(GPIOC, PIN4, 0, 0, 0, 1);

PORT_Init(PORTC, PIN4, PORTC_PIN4_SDADC_CH1P, 0); //PC.4 => SDADC.CH1P
PORT_Init(PORTC, PIN2, PORTC_PIN2_SDADC_CH3P, 0); //PC.2 => SDADC.CH3P

SDADC_initStruct.clk_src = SDADC_CLKSRC_HRC_DIV8; //时钟
SDADC_initStruct.channels = SDADC_CH1|SDADC_CH3; //通道选择
SDADC_initStruct.out_cali = SDADC_OUT_CALIED; //校准
SDADC_initStruct.refp_sel = SDADC_REFP_AVDD; //基准源-内部VDD
// SDADC_initStruct.refp_sel = SDADC_REFP_REFP; //基准源-外部基准
SDADC_initStruct.trig_src = SDADC_TRIGSRC_SW; //触发条件
SDADC_initStruct.Continue = 0; //连续模式
SDADC_initStruct.EOC_IEn = 0; //中断使能
SDADC_initStruct.OVF_IEn = 0;
SDADC_initStruct.HFULL_IEn = 0;
SDADC_initStruct.FULL_IEn = 0;
SDADC_Init(SDADC, &SDADC_initStruct); //配置SDADC

//配置增益、公共端
SDADC_Config_Set(SDADC, SDADC_CFG_A, SDADC_CFG_GAIN_1, 1, 1);
SDADC_Config_Cali(SDADC, SDADC_CFG_A, SDADC_CALI_COM_GND, 0);
SDADC_Config_Sel(SDADC, SDADC_CFG_A, SDADC_CH1);

//配置增益、公共端
SDADC_Config_Set(SDADC, SDADC_CFG_B, SDADC_CFG_GAIN_1, 1, 1);
SDADC_Config_Cali(SDADC, SDADC_CFG_B, SDADC_CALI_COM_GND, 0);
SDADC_Config_Sel(SDADC, SDADC_CFG_B, SDADC_CH3);

SDADC_Open(SDADC); //使能SDADC

```

如图，为配置信息，可选择不同的基准源。

```

void My_Get_SDADC_P(void)
{
    /*获取数据*/
    uint32_t chn;
    SDADC_Start(SDADC); //打开转换
    while((SDADC->STAT & SDADC_STAT_EOC_Msk) == 0); //等待转换完成
    for(uint16_t i=0;i<0x900;i++);
    while((SDADC->STAT & SDADC_STAT_EOC_Msk) == 0); //等待转换完成
    ADCP0 = (SDADC_Read(SDADC,&chn))+0x8000; //得到一个数据
    ADCP1 = (SDADC_Read(SDADC,&chn))+0x8000; //得到一个数据
    SDADC_Stop(SDADC);

    /*OLED显示出来*/
    OLED_ShowNum(28,0,ADCP0,7,16);
    OLED_ShowNum(28,2,ADCP1,7,16);

    /*获取变化量*/
    if(SDAdcGetNum == 1)//初始化
    {
        ADCP0_MAX = ADCP0_MIN = ADCP0;
        ADCP1_MAX = ADCP1_MIN = ADCP1;
    }

    if(ADCP0 > ADCP0_MAX) ADCP0_MAX = ADCP0;
    if(ADCP0 < ADCP0_MIN) ADCP0_MIN = ADCP0;

    if(ADCP1 > ADCP1_MAX) ADCP1_MAX = ADCP1;
    if(ADCP1 < ADCP1_MIN) ADCP1_MIN = ADCP1;

    /*显示变化量*/
    if(SDAdcGetNum == 0xff)//初始化
    {
        OLED_ShowNum(96,1,(ADCP0_MAX - ADCP0_MIN),4,12);
        OLED_ShowNum(96,3,(ADCP1_MAX - ADCP1_MIN),4,12);
    }

    SDAdcGetNum = 0;//清零
}
SDAdcGetNum++;

/*延时*/
for(uint16_t i=0;i<0x800;i++);
}

```

### 3.1.3. 硬件选择

根据第二章的硬件介绍，灵活短接不同的短接端子，配合不同的软件配置，即可组成以下不同的测试条件：

- SARADC 测量电压
  - 非电压跟随、电压跟随
  - 外部基准、内部基准
- SARADC 测量电流
  - 非电压跟随、电压跟随
  - 外部基准、内部基准
- SDADC 测量电压
  - 非电压跟随、电压跟随
  - 外部基准、内部基准
  - 单端模式、差分模式
- SDADC 测量电流
  - 非电压跟随、电压跟随
  - 外部基准、内部基准
  - 单端模式、差分模式

## 4. 测试结果

详情见 Excel



ADC测试板V1.0数  
据测量.xlsx

## 5. 应用程序示例

本节内容说明内 16 位 SDADC 怎样用于温度测量、压力测量等应用程序示例。

### 5.1.1. 温度测量

本示例应用温度传感器 PT100 为例，使用 16 位 sigma-delta 模数转换器进行温度测量。

测量步骤如下：

- 温敏电阻 PT100 通过 1.8K 的参考电阻连至 SDADC 通道电压输入端
- 通过跳线帽调整电路为内部参考电压，单端输入，公共的 AINREFM 引脚作为 N 输入端的模式，输入范围为 0 V 到 VREF/(2\* 增益 )
- SDADC 配置为与硬件配置相同模式
- SDADC 内部增益设为 8，因此范围为 0V-VREF/16，
- 通过跳线帽调整电路为内部参考电压 VREF+，因此测量范围为 0 V 至 VREF/16=0.20625V。
- 转换以连续模式进行
- 软件配置转换结束可产生中断。

温度计数公式如下：

$$R_{pt100} = 100 + 0.385 * T$$

$$\rightarrow T = (R_{pt100} - 100) / 0.385$$

$$V_{pt100} = R_{pt100} * I_{pt100} = R_{pt100} * VDD\_ANA / 2 * Rref$$

$$\rightarrow R_{pt100} = V_{pt100} * 1800 * 2 / VDD\_ANA$$

$$\rightarrow T = ((V_{pt100} * 1800 * 2 / VDD\_ANA) - 100) / 0.385$$

注：Rpt100 为 PT100 传感器的电阻

Vpt100 为 PT100 传感器上测得的电压

Ipt100 为流过 PT100 传感器的电流

VDD\_ANA 为模拟电压

Rref 为 18K 的参考电阻

PT100 温度传感器温度对应电阻值

| 温度/°C | 电阻/Ω |
|-------|------|
|-------|------|

|     |        |
|-----|--------|
| -50 | 80.31  |
| 0   | 100    |
| 20  | 107.79 |
| 200 | 175.86 |

温度测量分两步执行：

**温度传感器校准：**此阶段将 100 Ohm 电阻代替 PT100 温敏电阻（PT100 温度为 0 摄氏度时电阻为 100 欧），算出校正系数。此步骤校准了增益。

**温度测量：**此阶段执行时将 100 欧电阻去掉，用 PT100 代替。

SDADC 转换器测量模拟电压，然后根据下式算出温度：

$$\text{TemperaturePT100} = (((((\text{CoeffCorrection} * (\text{Avrg\_Data}/\text{SDADC\_GAIN}) * \text{REFERENCE\_R} * 2) / \text{SDADCRESOL}) - \text{RESISTANCE\_ZERODEGRE}) / \text{RESISTANCE\_COEFFICIENT});$$

注：

*Avrg\_Data* 为 256 个采样的平均值（舍弃开启后的第一组数据）

*SDADC\_GAIN* 为内部 SDADC 增益。在本例中配置为 8

*CoeffCorrection* 为在温度传感器校准过程中计算得到的校正系数。

*REFERENCE\_R* 为 1.8K 的参考电阻

*SDADCRESOL* 为 sigma delta 转换器： $2e16-1$

*RESISTANCE\_ZERODEGRE* 为 0 °C 时 PT100 的电阻（100 欧）

*RESISTANCE\_COEFFICIENT* 为 PT100 传感器的系数

## 6. 修订记录

| 版本   | 日期 | 说明   |
|------|----|------|
| V1.0 |    | 初版发布 |
|      |    |      |
|      |    |      |
|      |    |      |