

1.7 CC2530 Day-7 ADC转换

1.7 CC2530 Day-7 ADC转换

ADC转换基础知识

ADC转换相关寄存器

1-APCFG 模拟I/O配置寄存器

2- ADCH ADC数据高八位

3-ADCL ADC数据低八位

4-ADCCON1 ADC 控制寄存器1

5-ADCCON2 ADC 控制寄存器2 (序列转换)

6-ADCCON3 ADC控制寄存器3 (单次转换)

ADC转换基础知识

CC2530 的ADC 支持多达14 位的模拟数字转换，具有多达12 位的ENOB（有效数字位），有序列转换和单次转换2 种方式。它有16 个输入通道，包括8 路外部模拟输入通道AIN0 ~ AIN7、4 路外部差分输入、1 路内部温度传感器电压、1 路三分之一AVDD5 电压、正参考电压和GND 接地。

它有一个参考电压发生器，可提供4 路参考电压，分别是：

- AVDD 引脚
- AIN7通道
- AIN6 和AIN7 的差分电压
- 内部的1.25V 电压

ADC转换有四个步骤：

- 采样
- 保持
- 量化
- 编码

```
1 //ADC转换用法
2
3 //=====以查询方式获取数据=====
4 unsigned int adc_value = 0;
5 float adc_volt = 0;
6 unsigned char str[64];
7
8 //初始化ADC
9 void Init_ADC()
10 {
11     APCFG |= 0x01; //选择AIN0作为模拟I/O端口
12 }
13
14 //获取采样值函数
15 void Get_Value()
16 {
17     ADCCON3 = (0x80 | 0x30 | 0x00); //AVDD5引脚 512抽取率 AIN0通道
18     //ADCCON3 = 0xB0; 功能与上面的一样
```

```

19     if((ADCCON1 & 0x80) == 1)
20     {
21         adc_value = ADCH;    //读取高八位
22         adc_value = (adc_value << 8) | ADCL;    //左移腾位置，并载入低八位
23         adc_value = adc_value >> 5; //取十位有效位（ADCL前两位无效，所以最少右移2
    位）
24
25         //电压换算方法：(3.3/2的有效位次方-1)x采样值
26         //例如取了10为有效位电压为（3.3/（1024-1））*adc_value
27
28         adc_volt = (3.3/1023) * adc_value; //换算电压
29         sprintf((char *)str, "AIN0的采样结果是： %d,电压为：
    %.2fV\r\n", adc_value, adc_volt);
30
31         //这时候就可以用串口发送函数把数据发出去了
32         UR0_SendString(str);
33     }
34 }
35
36 //=====以中断方式获取数据=====//
37 unsigned int adc_value = 0;
38 float adc_volt = 0;
39 unsigned char str[64];
40
41 //初始化ADC
42 void Init_ADC()
43 {
44     APCFG |= 0x01; //把P0端口设置成模拟信号功能
45     ADCIE = 1; //打开ADC中断
46     EA = 1; //打开总中断
47 }
48
49 //中断服务函数
50 #pragma vector = ADC_VECTOR
51 __interrupt void Service_ADC()
52 {
53     adc_value = ADCH; //读取高八位
54     adc_value = (adc_value << 8) | ADCL; //向左移8位，给ADCL腾位置并载入低八
    位
55
56     adc_value = adc_value >> 5; //取十位有效位
57     adc_volt = (3.3/1023) * adc_value; //换算电压
58
59     sprintf((char *)str, "AIN0的采样结果是： %d,电压为：
    %.2fV\r\n", adc_value, adc_volt); //格式化输出
60     UR0_SendString(str); //发送数据
61 }
62
63 //设置ADCCON3的参数来启动中断
64 //主函数
65 void main()
66 {
67     Init_ADC();
68     while(1)
69     {
70         ADCCON3(0x80 | 0x30 | 0x00);
71     }
72 }

```

ADC转换相关寄存器

1-APCFG 模拟I/O配置寄存器

【58】APCFG 模拟 I/O 配置寄存器

位	位名称	复位值	操作	描述
7: 0	APCFG[7:0]	0x00	R/W	模拟外设 I/O 配置。 用于选择 P0_7~P0_0 作为模拟 I/O 端口。 8 路模拟量输入来自 P0 端口组的 8 个 I/O 引脚,不必通过编程将这些引脚变为模拟输入,当相应的模拟输入引脚在 APCFG 寄存器中被禁用时,该通道将被跳过。 0: 模拟 I/O 禁用 1: 模拟 I/O 使用

2- ADCH ADC数据高八位

【59】ADCH ADC 数据低位寄存器

位	位名称	复位值	操作	描述
7: 0	ADC[13:6]	000000	R	ADC 转换结果的高位部分。

3-ADCL ADC数据低八位

【60】ADCL ADC 数据低位寄存器

位	位名称	复位值	操作	描述
7: 2	ADC[5:0]	000000	R	ADC 转换结果的低位部分。
1: 0	----	00	RO	没有使用,读出来一直都是 0。

注意: ADCL数据前两位无效

4-ADCCON1 ADC 控制寄存器1

【61】ADCCON1 ADC 控制寄存器 1

位	位名称	复位值	操作	描述
7	EOC	0	R/HO	转换结束。当 ADCH 被读取的时候被清除。如果已读取前一数据之前，完成一个新的转换，EOC 位仍然为高。 0: 转换没有完成。 1: 转换完成。
6	ST	0	R1/W	开始转换。读为 1，直到转换完成。 0: 没有转换正在进行。 1: 如果 ADCCON1.STSEL=11，并且没有序列正在运行，就启动一个序列转换。
5: 4	STSEL[1:0]	11	R/W1	启动选择。选择该事件，将启动一个新的转换序列。 00: P2.0 引脚的外部触发。 01: 全速，不等待触发器。 10: 定时器 1 通道 0 比较事件。 11: ADCCON1.ST = 1。
3:2	RCTRL[1:0]	00	R/W	控制 16 位随机数发生器。当写 01 时，操作完成时设置将自动返回到 00。 00: 正常运行。 01: LFSR 的时钟一次。 10: 保留。 11: 停止，关闭随机数发生器。
1:0	----	11	RO	没有使用，一直设为 11。

5-ADCCON2 ADC 控制寄存器2（序列转换）

【62】ADCCON2 ADC 控制寄存器 2

位	位名称	复位值	操作	描述
7: 6	SREF[1:0]	00	R/W	选择参考电压，用于序列转换。 00: 内部参考电压。 01: AIN7 引脚上的外部参考电压。 10: AVDD5 引脚。 11: AIN6-AIN7 差分输入外部参考电压。
5: 4	SDIV[1:0]	00	R/W	为包含在转换序列内的通道设置抽取率。抽取率也决定完成转换需要的时间和分辨率。 00: 64 抽取率（7 位 ENOB）。 01: 128 抽取率（9 位 ENOB）。 10: 256 抽取率（10 位 ENOB）。 11: 512 抽取率（12 位 ENOB）。
3: 0	SCH[3:0]	0000	R/W	序列通道选择。 0000: AIN0。 0001: AIN1。 0010: AIN2。 0011: AIN3。 0100: AIN4。 0101: AIN5。

				0110: AIN6。	0111: AIN7。
				1000: AIN0-AIN1。	1001: AIN2-AIN3。
				1010: AIN4-AIN5。	1011: AIN6-AIN7。
				1100: GND。	1101: 正电压参考。
				1110: 温度传感器。	1111: AVDD/3。

6-ADCCON₃ ADC控制寄存器₃（单次转换）

【63】ADCCON₃ ADC 控制寄存器 3

位	位名称	复位值	操作	描述
7: 6	EREF[1:0]	00	R/W	选择用于单通道转换的参考电压。 00: 内部参考电压。 01: AIN7 引脚上的外部参考电压。 10: AVDD5 引脚。 11: AIN6-AIN7 差分输入外部参考电压。
5: 4	EDIV[1:0]	00	R/W	设置用于额外转换的抽取率。抽取率也决定完成转换需要的时间和分辨率。 00: 64 抽取率（7 位 ENOB）。 01: 128 抽取率（9 位 ENOB）。 10: 256 抽取率（10 位 ENOB）。 11: 512 抽取率（12 位 ENOB）。
3: 0	ECH[3:0]	0000	R/W	单个通道选择。 选择写 ADCCON3 触发单个转换所在的通道号码。当单个转换完成，该位自动清除。 0000: AIN0。0001: AIN1。 0010: AIN2。0011: AIN3。 0100: AIN4。0101: AIN5。 0110: AIN6。0111: AIN7。 1000: AIN0-AIN1。1001: AIN2-AIN3。 1010: AIN4-AIN5。1011: AIN6-AIN7。 1100: GND。1101: 正电压参考。 1110: 温度传感器。1111: AVDD/3。
设计参考		启动一次 A/D 转换，参考电压选择 AVDD5 引脚，256 抽取率，AIN3。 ADCCON3 = (0x80 0x20 0x03); //1010 0011 或者: ADCCON3 = 0xA3;		