第八章 模拟外设

现实世界的物理量信号（如温度、湿度、大气压、重量、电压、电流、光照等）都是模拟量，这些信号的幅值随着时间的变化都是连续的、非跳跃的。在日常生活和工业控制中，人们经常需要对许多物理量进行精确的测量、分析处理。本章主要介绍与 MCU 应用相关的模/数转换器（ADC）、模拟比较器（AC）和数/模转换器（DAC），并结合 Tiva 微控制器的相关模拟外设，说明其特点和使用方法。

1 模数转换（ADC）

1.1 概述

1.1.1 模拟信号和数字信号的区别

在实时控制和智能仪表等应用中，控制或测量的对象往往是一些连续变化的模拟量，如温度、压力、流量、速度等。利用传感器把各种物理量检测出来、并转换为电信号，再经过模/数转换器变成数字量，才能被计算机处理和控制。

对于单片机而言，一些模拟信号经过适当的电路变换，可以直接连接到某些特定的 GPIO 管脚，其输入电压范围为 0~5V。LunchPad 的电源电压为 3.3V，通常只能承受 3.3V 的电压，但也有些管脚可以承受 5V 电压（使用时应查阅数据手册）。经过 A/D 变换变换成数字量，再对这些数字量进行各种数据处理，便能够完成对外部参数（幅度、频率等）进行测量等的功能；或者完成时序上的编程控制，直接经过 DAC 数模转换，输出 PWM 等波形直接控制多种外部设备(如直流电机、步进电机等)。 如此通过对模拟外设的输入输出过程控制，实现与外部世界的信息交互。

1.1.2 A/D 变换过程

ADC 是 Analog-to-Digital Converter 的缩写，意为模/数转换器。顾名思义，ADC 就是完成从模拟量到数字量转化过程的器件。目前很多 MCU 内部都集成了 ADC 部件。TM4C123GH6PM 自身就集成了 2 个 12 位 ADC 转换器，没有 DAC 转换器。德研实验板提供了一个 12 位 DAC 转换芯片 DAC7512 接口电路实现 D/A 功能。

模拟信号转换为数字信号，一般包含四个过程，即采样、保持、量化和编码，形成离散的数字信号。如图 8.1 所示为简化图，把采样和保持放在了一起。

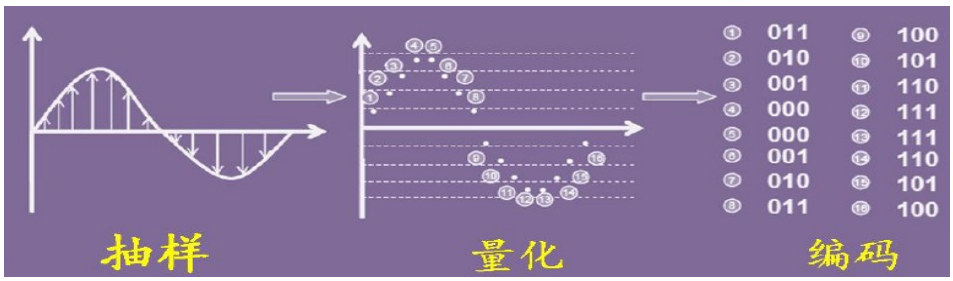


图 8.1 A/D 变换三个过程

采样实现的是对一个时间上和量值上均是连续变化的模拟量，按一定的时间间隔抽取样值。为了保证转换的准确性，要求在转换过程中取样值保持不变，这就是保持过程。一个简单的采样保持电路如图 8.2 所示。对于我们所使用的 TM4C123GH6PM launchpad，可以通过定时一段时间启动 A/D 变换来实现采样，定时时间可以由一个周期拟采样的点数来计算得到。

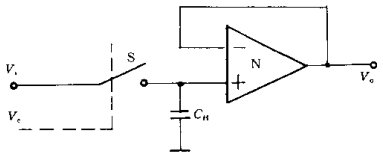


图 8.2 一个简单的采样保持电路

采样保持电路的输出信号仍然是模拟信号（连续信号），通常用数字测量单位去测量并取其整数，然后将这个整数值用一组二进制代码来表示，这就是量化、编码过程。一般把取整量的过程称为量化，把用二进制代码表示量化值的过程称为编码。如果是 3 位数据线（通常定义为 3 位 A/D 变换芯片），则可形成 8 个量化等级（2^3=8）。常用的有逐次逼近型 A/D 变换、双积分式 A/D 变换电路等。而对于 TM4C123GH6PM，是 12 位的 A/D 功能，则可以形成 2^{12}=4096 个量化等级,所以，位数越高，A/D 转换的精度越高，芯片价格也会越高。

1.1.3 A/D 变换技术指标

1.1.3.1 分辨率

分辨率用来反映 ADC 对输入电压微小变化的响应能力。通常用数字输出最低位（LSB）所对应的模拟输入电压来表示，n 位 A/D 转换能反映出 $1/2^n$ 满量程模拟输入电压。分辨率直接与转换器的位数有关，一般也可简单地用数字量的位数来表示分辨率，即 n 为二进制数，最低位所具有的权值就是它的分辨率。常用的 ADC 位数有 8、10、12、16、24 位等。

1.1.3.2 精度

精度有绝对精度和相对精度两种表示方法。一般而言，分辨率（位数）较高的 ADC，精度也相对较高。

（1）绝对精度

在一个转换器中，对应于一个数字量的实际模拟输入电压和理想模拟输入电压之差并非是一个常数，这些差值的最大值定义为“绝对误差”。通常以数字量的最小有效位（LSB）的个数值来表示绝对精度。

（2）相对精度

相对精度是指整个转换范围内，任意数字量所对应的模拟输入量的实际值与理想值之差，用模拟电压满量程的百分比表示。

对于 12 位 A/D 变换器，能够形成 2^{12}=4096 个量化等级，其精度为 $1/2^{12}$=1/4096。所以

位数越高，分辨率越高，精度越高。也就是说，测量精度与 ADC 器件的位数有一定的关系，但没有直接的关系。我们还可以通过软件编程，大大提高模数变换的精度。

1.1.3.3 转换时间

转换时间是指完成一次 A/D 转换所需要的时间，即由发出启动转换命令信号到转换结束信号开始有效的时间间隔。转换时间的倒数称为转换速率。ADC 的转换速率与 ADC 的结构类型、位数有关，主要取决于 ADC 的结构类型。如双积分式 ADC 转换速率都比较低（几 SPS 到几十 SPS,SPS 为每秒采样数），而逐次比较式 ADC 转换速率都比较高（几十 KSPS 到几百 KSPS）。

1.1.3.4 量程

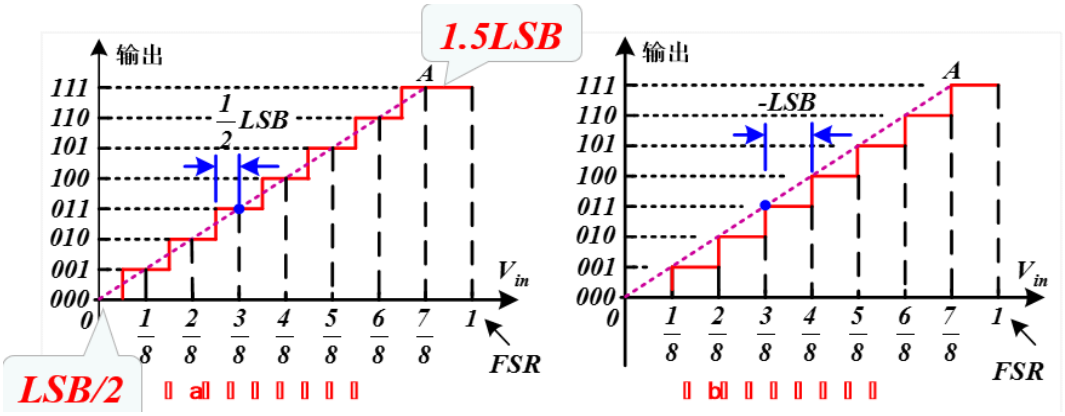
量程是指 ADC 所能转换的模拟输入电压范围，分单极性和双极性两种。例如，单极性的量程为 0~+5V、0~+10V；双极性的量程为-5V~+5V、-10V~+10V。

TM4C123GH6PM 的 VREFP 直接与 VDDA 相连接到了 3.3V 电源上，而另一端 VREFN 接到了 GNDA，接至地上，所以 TM4C123GH6PM 的量程大多为 0~3.3V。

1.1.3.5 量化误差

对任何 AD 来说，量化后输出的数字信号值都是以 1LSB 的电压值步进的，介于 1LSB 之间的电压将按照一定的规则进行入位或舍弃，这个过程中造成的误差被称为“量化误差”，量化误差属于原理性误差，是无法消除的。

例如：满量程 FSR=1V 的 3 位 ADC，其分辨率为 1/8V（1LSB）。分别采用舍入量化和截断量化两种方式，情况如下：



（a）舍入量化　　（b）截断量化

图 8.3 量化误差

如图 8.3 所示，舍入量化误差为 1/2LSB，截断量化误差为 1LSB。

应用开发时，选择 ADC 都会依据具体应用的需求，最主要的指标就是分辨率、精度和转换速率。一般应用中，12 位的逐次比较型 ADC 可以满足大部分的应用需求。ADC 的位数、精度、转换速率越高，价格也越高，应该统筹考虑开发成本。

对于一些精度要求很高、转换速率要求不高的应用，如称重系统、高精度仪表等，可以采用Σ-△型 ADC,这类 ADC 的位数一般都在 16~24 位，转换速率为几十 SPS，而且对 50Hz、60Hz 的工频干扰有很强的抑制能力，如 AD7792、ADS1232 等。

1.2 TM4C123GH6PM 的模数转换

1.2.1 逐次逼近型 ADC

模拟数字转换器 ADC 是一个将连续的模拟电压转换为离散的数字量的模块。常用的方法是逐次逼近型 ADC。TM4C123GH6PM 采用的是逐次逼近型 ADC。

1.2.2 TM4C123GH6PM 的 ADC

TM4C123GH6PM 共包含两个 ADC 模块，如图 8.4 所示。其特点如下：

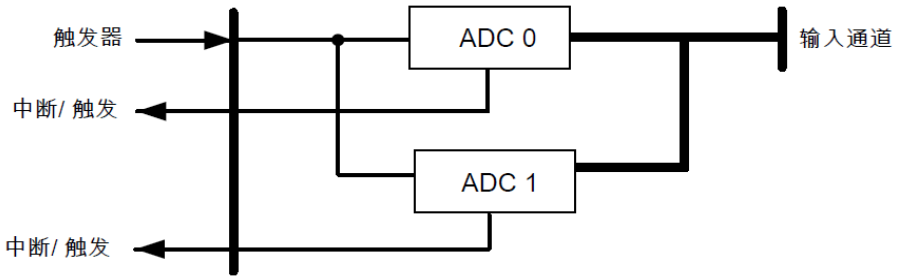


图 8.4 两个 ADC 模块

M4C123GH6PM 微控制器提供两个 ADC 模块，每个模块都具有以下特点：

- 12 个共享模拟输入通道
- 12 位 ADC 精度
- 单端、差分输入配置
- 片内温度传感器
- 一百万样本/秒的最大采样速率
- 在可编程的采样时间中的可选择的相位为从 22.5° 到 337.5°
- 四个可编程的采样转换序列发生器长从 1 到 8 个条目长，相应的转换结果 FIFO
- 灵活的触发控制

  ——控制器（软件）

  ——定时器

  ——模拟比较器

  ——PWM

  ——GPIO
- 硬件平均多达 64 个采样
- 数字比较器单元提供 8 个数字比较器
- 转换器使用 VDDA 和 GNDA 作为参考电压

- 模拟电路的电源和接地与数字电源和接地是分开的
- 高效传输使用微型直接存储器存取（μDMA）

——每个采样序列发生器的专用通道

——ADC 模块使用 DMA 突发请求

1.3 A/D 变换原理

1.3.1 单端输入

ADC 使用内部信号 VREFP 和 VREFN 作为产生来自选择模拟信号输入的转换值的参数。V_{REFP} 连接到 VDDA，V_{REFN} 连接到 GNDA。如图 8.5 所示。对于 TM4C123GH6PM，Vref=V_{REFP}-V_{REFN}=3.3V，通常 V_{REFN} 接到模拟地上，为 0 电平。

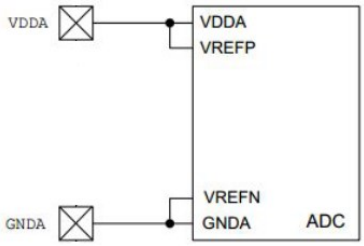


图 8.5 Vref 示意图

对于 12 位的 A/D 转换芯片，该转换值的范围为从 0x000 到 0xFFF。在单端输入模式中，0x000 值与 VREFN 上的电压值一致；0xFFF 值与 VREFP 上的电压值一致。通过这样的配置，分辨率就可以使用(VREFP - VREFN) / 4096 来计算。

12 位输出全 1 的时候（即为 0xFFF）对应输出为 Vref（5V 或 3.3V）。在单端输入模式中，对应于一个变换出来的任意二进制数 X，会对应一个实际电压值，为：

$$V_O = X \times \frac{V_{ref}}{2^{12}-1}\ （V）= X \times \frac{V_{ref}}{4095}(V)$$

例如，信号电压为 1.65 V 时对应的采样值就是：$1.65 \ / \ （\frac{3.3}{4095}$ ）= 2047，即 2047=07FFH，对应输出模拟电压就是 1.65V。

虽然有些 GPIO 管脚最大可以承受 5V 输入，但在 3.3V~5V 之间输出均为 1，失去了变换的精度和意义，所以此时的输入电压范围应该为 0~3.3V，设计前期的模拟变换电路时应予以考虑。如图 8.6 所示，超出 3.3V 的电压，我们称为输入饱和。

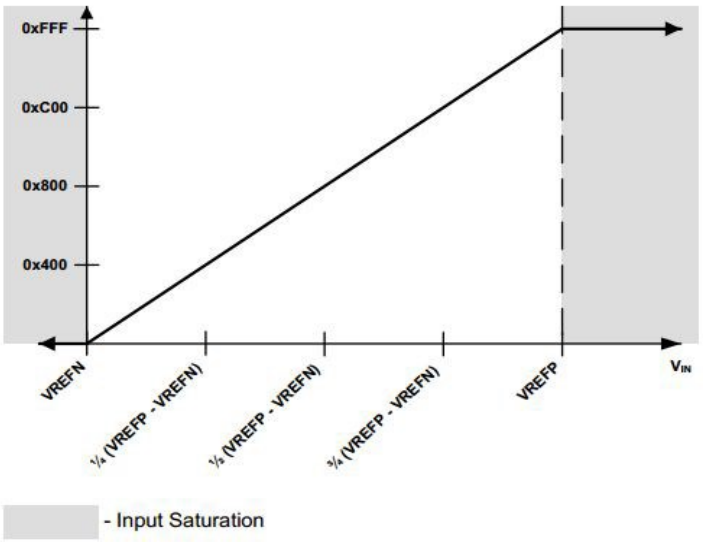


图 8.6 ADC 转换结果

1.3.2 差分输入

除了传统的单端采样，ADC 模块还支持对两个模拟输入通道进行差分采样。要启用差分采样功能，软件必须在某个采样步骤的配置半字节中将 ADCSSCTLOn 寄存器的 Dn 位置位。若采样序列中某个采样动作配置为差分采样，必须在 ADCSSMUXn 寄存器中配置输入的差分信号对。差分模式下采样电压是奇数通道与偶数通道电压的差值。此时，由于 2 个输入端的变化范围均为 VREFN~VREFP，因此 ADC 的采样精度为 2*（VREFP-VREFN）/4096。

由于 Tiva 微控制器 GPIO 口只能输入 0~3.3V 的正电压信号，进行差分变换时需要把信号变换成以 V=1.65 为基准的差分信号，即当差模电压△V=0 时，转换结果为 0x800；当△V>0 时，转换结果为 0x800~0xFFF；当△V<0 时，转换结果为 0x000~0x800。

图 8.7 显示了差分电压，ΔV，是怎样表示 ADC 编码的。

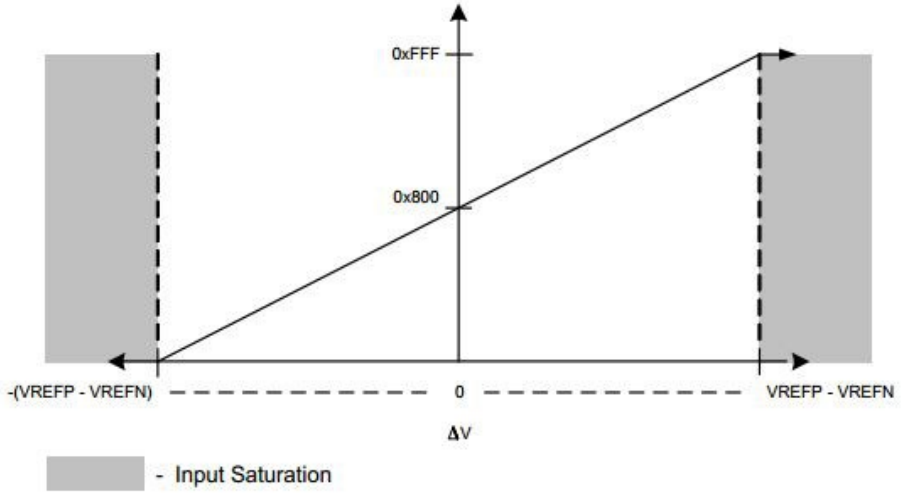


图 8.7 差分电压表示

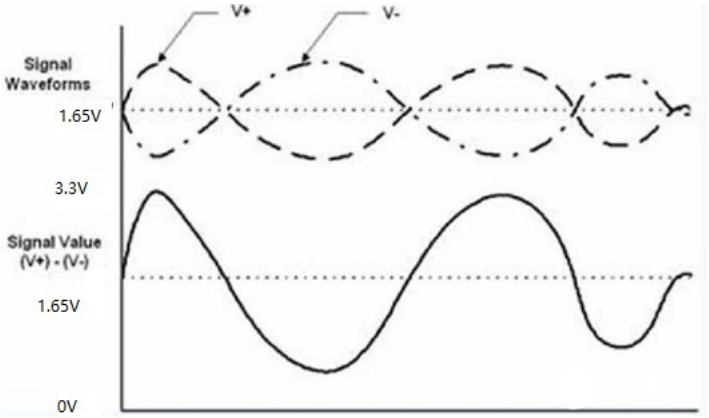


图 8.8 差分变换示意图

差分输入的最大优点是精度高，可以增大共模抑制比，减小干扰。

应用差分输入前，需要经过电路变化把图 8.8 中的 V+信号转换成 V-，（相位相差 180 度，即反相），然后从 TIVA 的差分对输入进行变换。这样 V=（V+）-（V-），减掉共模干扰后，为 V+信号的 2 倍，最后的变换值 V+=1/2V。

当一个序列节拍配置为差分采样，必须在 ADCSSMUXn 寄存器配置采样输入对。差分对 0 采样模拟输入 0 和 1；差分对 1 采样模拟输入 2 和 3；等等（见表 8.1）。ADC 不支持其他差分对，如模拟输入 0 和模拟输入 3 。

表 8.1 差分采样对

差分对	模拟输入
0	0 和 1
1	2 和 3
2	4 和 5
3	6 和 7
4	8 和 9
5	10 和 11

1.4 采样

1.4.1 采样序列发生器

采样序列（*Sample Sequence*）由**一组编程的连续采样**组成，因此 ADC 模块可以自动从多个输入源采集数据，无需处理器对其重新配置或进行干预。

每个采样动作都可灵活编程：可以指定序列中的每个采样动作对哪个 AIN 输入进行采样，也允许序列中的多个采样动作对同一 AIN 输入进行采样。

每个 ADC 模块有 4 个采样序列，如表 8.2 所示。它们的采样数以及 FIFO 深度有所不同。

表 8.2 序列发生器的采样数和 FIFO 深度

序列发生器	采样数	FIFO 深度
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

允许在某个采样动作后结束整个采样序列。

允许在采样序列的每个采样动作后产生中断。

每个 FIFO 单元均为一个 32 位的字，低 12 位包含的是转换结果。

1.4.2 采样控制

（1）具有 4 个可配置的采样序列发生器；

（2）具有硬件平均电路，最多可以做 64 个样本的硬件平均，提高采样精度；

（3）有多种灵活的中断控制，包括 PWM、GPIO、计时器触发、模拟比较器触发等；

（4）内置 8 个数字比较器，可以方便地实现数字信号监测功能；

（5）可以使用 DMA 传输数据；

（6）可配置的参考电压 VREFP(有的用 VREFA+表示)、VREFN（有的用 VREFA-表示）。

1.4.3 采样事件

M4C123GH6PM 有 5 个触发源，所谓触发源是指能触发某个采样序列开始一次序列采样的事件。主要有：

① 控制器触发（默认）

② 模拟比较器触发

③ GPIO 外部信号触发

④ 通用定时器触发

⑤ 持续采样触发

每个采样序列发生器的采样触发条件均通过 ADC 事件多路复用器选择寄存器 (ADCEMUX) 予以定义。

采样序列具有优先级。同时发生多个采样事件（触发条件）时，将按照采样序列的优先级对它们进行排序和依次处理。优先级的有效值为 0~3，其中 0 代表最高优先级、3 代表最低优先级。

配置持续采样触发条件时务必慎重。假如某个采样序列的优先级过高，可能导致其它低优先级采样序列始终无法运行。

1.4.4 ADC 时钟和采样相位控制

ADC 时钟为 16M，可以通过分频的 PLL 输出、PIOSC 或 16M 的 MOSC 时钟源获得。系统

时钟的频率必须与 ADC 时钟相同或更高。

采样频率为 1M，1 次采样需要 16 个 ADC 时钟周期。

ADC0 和 ADC1 是相互独立的两个 ADC 模块，共享 12 个模拟输入引脚因此可以非常灵活地设置触发源、采用的模拟输入端。假如两个转换器以相同的采样率工作，其采样点可以配置为同相，或配置为错开一定的相角（一个周期可以有 16 个点，可实现 15 种离散的相位差）。采样点延后的相位通过 ADC 采样相位控制（ADCSPC）寄存器按 22.5°（16 个时钟为 1 个周期 360°，相位按 360 °/16=22.5°） 逐步递增，最大可递增至 337.5°。

例如图 8.8 所示采样电路。

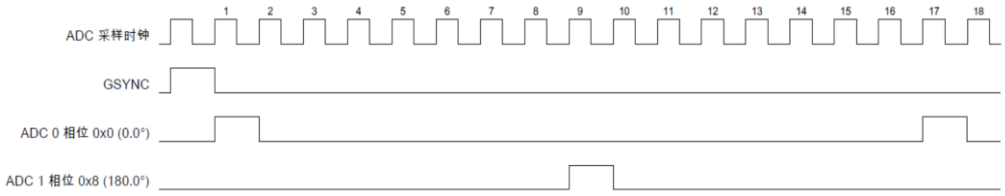


图 8.8 采样相位示例

两个 ADC 采样同一个输入，其中一个相位设为 180 °，就可以实现信号的双倍速率采样。

1.5 数字化比较器

每个 ADC 模块内置有 8 个数字比较器， ADC 转换结果可以保存到采样序列 FIFO 中，也可以发送给数字比较器，与用户编程的门限进行比较，判定 ADC 结果是在低值带、中值带还是高值带运行而生成中断或触发事件。

两组比较门限 COMP0 和 COMP1 可将转换结果划分为 3 个区域：低值带（小于 COMP0）,中值带（大于 COMP0 但小于等于 COMP1）,高值带（大于等于 COMP1）

ADC 通常用于对外部信号采样并监控其数值的变动，确保其保持在给定的范围内。ADC 转换结果可直接发送给数字比较器，与用户编程的门限进行比较。门限通过 ADC 数字比较器范围寄存器（ADCDCCMPn）配置。ADC 可配置为根据 ADC 是在低值带、中值带还是高值带（可在 ADCDCCMPn 位域进行配置）运行而生成中断。另外可将数字比较器的 4 种工作模式（单次触发，持续触发，迟滞单次触发，迟滞持续触发）应用于中断配置。图 8.9 为低值带工作示意图。

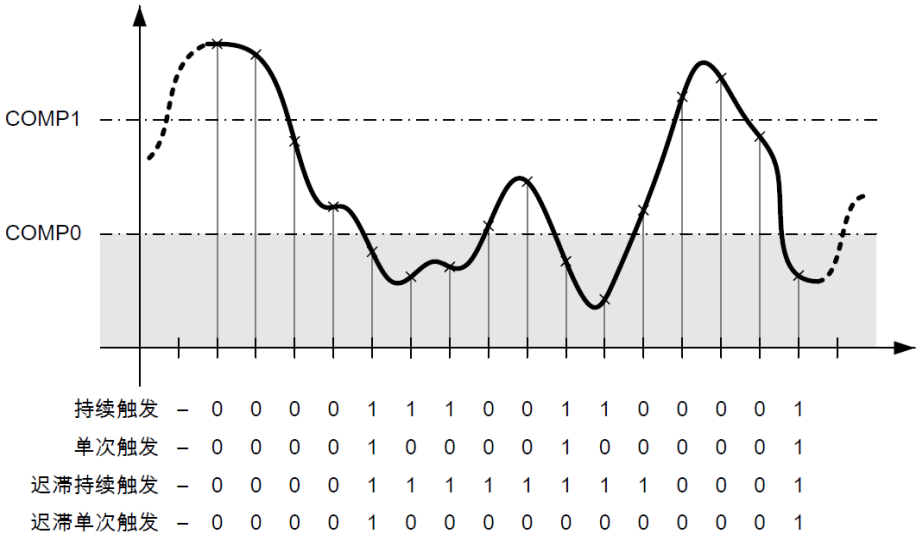


图 8.9 低值带工作示意图

1.6 内部温度传感器和硬件采样平均

M4C123GH6PM 有内部温度传感器，它将温度测量值转换为电压。温度 TEMP （单位 ° C）可以通过以下公式得出：

TEMP = 147.5 - ((75 * (Vref) × ADCCODE) / 4096

有硬件采样平均功能。如果启用，硬件会自动进行多次采样，并以平均值作为单次采样的数据返回 FIFO 中，但吞吐率会下降。最多平均 64 次，相应的吞吐率下降为 1/64。

默认情况下，硬件采样平均电路是关闭的，转换器捕捉的所有数据直接送入序列发生器的 FIFO 中。进行平均计算的硬件由 ADC 采样平均控制寄存器(ADCSAC) 进行控制。每个 ADC 模块只有一个平均电路，不论单端输入还是差分输入都会被执行相同的求平均值操作。

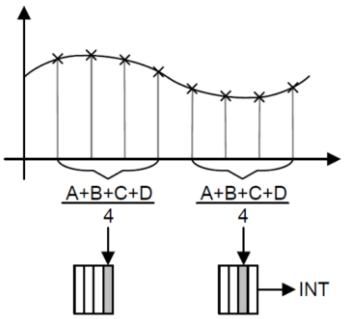


图 8.10 4 次平均示意图

1.7 中断控制

采样完成或是数字比较器检测出异常都可以被配置为中断事件。采样序列发生器和数字比较器的寄存器配置决定哪些事件会生成原始中断，ADC 有 8 个中断相量入口地址（分别为 ADC SS0、SS1、SS2、SS3 和 ADC1 SS0、SS1、SS2、SS3），相较于 GPIO、GPTM 等外设一个端口只有一个中断相量的情况，中断资源还算丰富，因此没有设置中断屏蔽状态寄存器 MIS。

ADC 模块的中断信号由 ADC 中断屏蔽（ADCIM）寄存器的 MASK 位的状态控制。当有中断发生，会置位原始中断状态（ADCRIS）寄存器中的相应位。序列发生器中断通过向 ADCISC 寄存器相应的位写 1 来清除。数字比较器中断通过向 ADC 数字比较器中断状态和清除（ADCDCISC）寄存器写 1 来清除。

1.8 TM4C123GH6PM ADC 模块的结构图

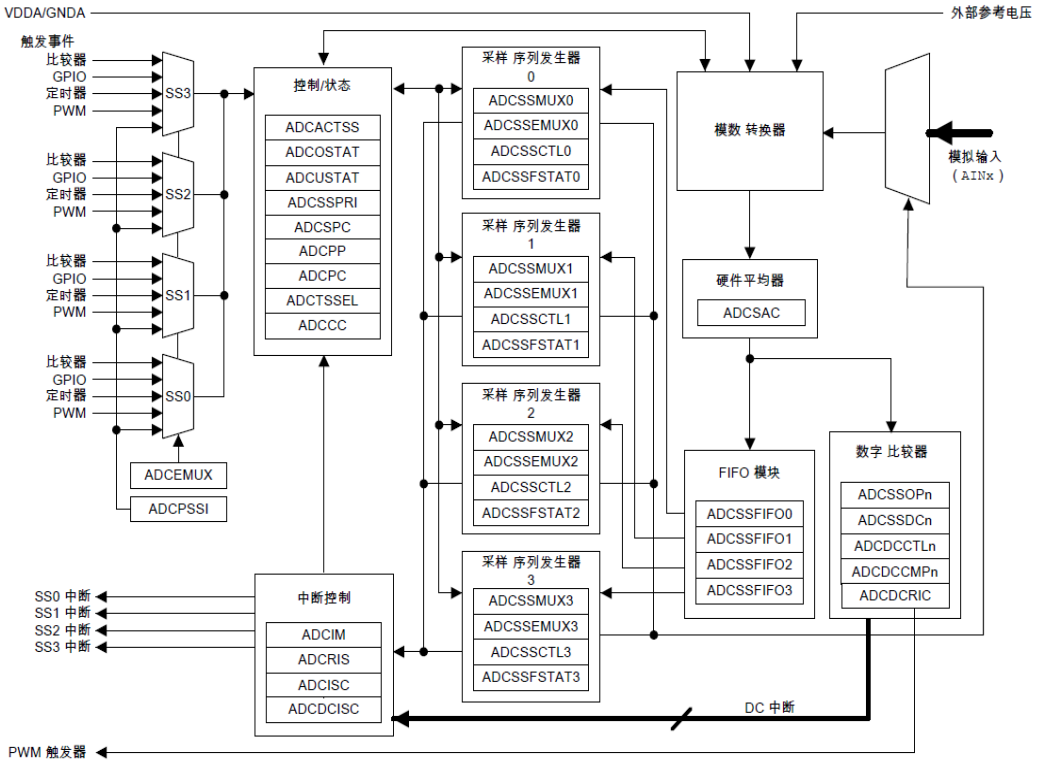


图 8.11 ADC 结构图

每个 ADC 模块包括三个部分，即寄存器、采样序列发生器以及 AD 转换部分。ADC 主要结构图如图 8.11 所示。

主要寄存器介绍：

1、采样序列

ADCSSFIFOn

ADCSSMUXn：序列每个动作的输入通道

ADCSSCTLn：序列每个动作的中断、结束、单端\差分

ADCSSFSTATn：FIFO 满、空标志；上、下指针位置

ADCEMUX、ADCTSSEL：序列触发源

ADCPSSI：控制器启动序列采样

ADCACTSS：序列启用和 busy 标志

ADCOSTAT：FIFO 的上溢状态

ADCUSTAT： FIFO 下溢状态

ADCSSPRI: 序列优先级

2、中断

ADCIM: 四个序列的中断屏蔽（采样和数字比较器中断）

ADCRIS: 四个序列的原始中断状态

ADCISC: 四个序列的屏蔽中断状态（读）和清除（写）

ADCDCISC: 八个数字比较器的中断状态及清除

3、其它

ADCSPC：采样相位控制

ADCSAC: 硬件平均

具体定义大家可以查阅相关寄存器查询。

1.9 TM4C123GH6PM 引脚复用

下表 8.3 列出了 ADC 模块的外部信号，并描述了各自的功能。AINX 信号是 GPIO 信号的模拟功能。下表中“引脚多路复用/引脚分配” 列出的 ADC 信号的 GPIO 管脚布局。这些信号通过清除 GPIO 数字使能（GPIODEN）寄存器对应的 DEN 位和设置 GPIO 模拟数字使能寄存器（GPIOAMSEL）的对应 AMSEL 位来配置。欲了解更多 GPIO 配置信息，请参考“通用输入/输出”。

表 8.3 ADC 信号（64LQFP）

引脚名称	引脚数	引脚复用/分配引脚	类型	缓存类型 a	描述
AIN0	6	PE3	输入	模拟	模数转换输入 0
AIN1	7	PE2	输入	模拟	模数转换输入 1
AIN2	8	PE1	输入	模拟	模数转换输入 2
AIN3	9	PE0	输入	模拟	模数转换输入 3
AIN4	64	PD3	输入	模拟	模数转换输入 4
AIN5	63	PD2	输入	模拟	模数转换输入 5
AIN6	62	PD1	输入	模拟	模数转换输入 6
AIN7	61	PD0	输入	模拟	模数转换输入 7
AIN8	60	PE5	输入	模拟	模数转换输入 8
AIN9	59	PE4	输入	模拟	模数转换输入 9
AIN10	58	PB4	输入	模拟	模数转换输入 10
AIN11	57	PB5	输入	模拟	模数转换输入 11

a TTL 表示引脚具有 TTL 兼容电平。

1.10 初始化配置

初始化配置步骤：

1、调用 SysCtlPeripheralEnable（）函数使能 ADC 时钟；

2、调用 SysCtlPeripheralEnable（）函数使能相应的 GPIO 模块的时钟；

3、通过向相应 GPIO 模块的 GPIOAMSEL 寄存器的相应位写入 1 来配置 AINx 信号为模拟输入。调用 GPIOPinTypeADC（）函数实现。注意：ADC 的 AINx 引脚都是固定的引脚；

4、调用 ADCSequenceConfigure（）函数可以配置 ADC 模块的基地址、采样序列号、启动采样序列的触发源、采样优先级。默认是采样序列发生器 0 为最低优先级和采样序列发生器 3 为最高优先级；

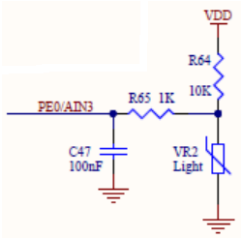
5、调用 ADCSequenceStepConfigure（）函数配置 ADC 模块的基地址、采样序列发生器的编号、被配置的步进序号、步进配置;步进配置必须是 ADC_CTL_TS、ADC_CTL_IE、ADC_CTL_END、ADC_CTL_D 之一，输入通道选择 (ADC_CTL_CH0 到 ADC_CTL_CH11)之一，和数字比较器选择(ADC_CTL_CMP0 到 ADC_CTL_CMP7)之一的逻辑或；

6、中断使能；

7、ADC 使能。

1.11 实验例程

德研 TIVA-PB 资料 V1.0\TIVA-demo\ADC-control-light



1.11.1 程序设计

实验流程步骤：

1) 设置时钟。

2) 配置 ADC 模拟输入引脚 PE0/AIN3 通道和 UART 引脚 PA0 和 PA1 引脚。

3) 使能外设并配置序列产生器 0。

4) 程序循环，持续检测并通过 UART 输出。

实验流程图如图 8.12 所示：

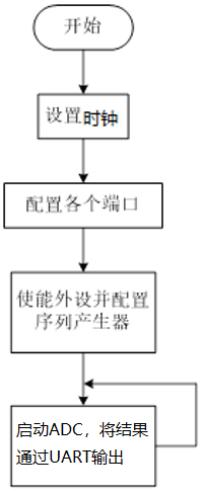


图 8.12 实验流程图

1.11.2 部分代码

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);//使能外设
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
 GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_0); // 引脚复用配置，AIN3 on port E0.
ADCSequenceConfigure(ADC0_BASE, 0, ADC_TRIGGER_PROCESSOR, 0);//配置序列 0 触发源为控制器触发
ADCSequenceStepConfigure(ADC0_BASE, 0, 0, ADC_CTL_CH3 );//配置序列 0 的 8 个动作
ADCSequenceStepConfigure(ADC0_BASE, 0, 1, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 2, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 3, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 4, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 5, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 6, ADC_CTL_CH3 );
ADCSequenceStepConfigure(ADC0_BASE, 0, 7, ADC_CTL_CH3 | ADC_CTL_IE | ADC_CTL_END);//最后 1 个
动作完成后触发中断并结束序列
ADCIntRegister(ADC0_BASE, 0, ADC0Sequence0Handler);//注册中断函数
ADCIntEnable(ADC0_BASE, 0);//打开 ADC0 中 0 序列的屏蔽中断
IntEnable(INT_ADC0SS0);//在 NVIC 中打开 ADC0SS0 的外设中断
ADCSequenceEnable(ADC0_BASE, 0);//使能序列 0
ADCIntClear(ADC0_BASE, 0);  //在采样前清除以前的中断标志
ADCProcessorTrigger(ADC0_BASE, 0);//控制器触发序列 0 采样
uint32_t pui32ADC0Value[8],sum;
void ADC0Sequence0Handler(void)//中断服务函数
{
    uint16_t i;

    ADCIntClear(ADC0_BASE, 0); //清除中断标志
    ADCSequenceDataGet(ADC0_BASE, 0, pui32ADC0Value);//将序列 0 的 8 次采样结果读到数组中
    for(i = 0;i < 8;i ++)
    {
        sum= sum+(pui32ADC0Value[i]*3300/4096);
    }
    flag=1;
}
```

1.11.3　主要库函数

1、GPIOPinTypeADC();//为模数转换输入配置引脚

参数：ADC 模块的基地址、管脚

GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_1); //选择 PE1

2、ADCIntClear();//清除采样序列中断源

ADCIntClear(ADC0_BASE, 0);//清除 ADC0 采样序列中断源

3、ADCSequenceConfigure();//配置采样序列的触发源和优先级

参数：ADC 模块的基地址、采样序列号、触发源、优先级

ADCSequenceConfigure(ADC0_BASE, 0, ADC_TRIGGER_PROCESSOR, 0);//配置 ADC0

采用 SS0 方式（8 个采样数）进行采样，处理器触发，0 优先级

4、ADCSequenceStepConfigure( );//配置采样序列的节拍

参数：ADC 模块的基地址、采样序列号、动作节拍号、该动作节拍的输入通道

ADCSequenceStepConfigure(ADC0_BASE, 0, 0, ADC_CTL_CH2);

ADC0 采样，采样序列 0，第 0 个采样，CH2 通道

5、ADCSequenceDataGet();//获取采样序列捕获的数据

参数：ADC 模块的基地址，采样序列号、数据存储地址

ADCSequenceDataGet(ADC0_BASE, 0, pui32ADC0Value);

6、ADCSequenceEnable(); //使能采样序列

7、ADCProcessorTrigger();//为采样序列产生一个处理器触发

参数：ADC 模块的基地址、采样序列号

8、ADCIntStatus();//获取当前中断状态

9、触发源列表：

```
#define ADC_TRIGGER_PROCESSOR   0x00000000  // Processor event
#define ADC_TRIGGER_COMP0       0x00000001  // Analog comparator 0 event
#define ADC_TRIGGER_COMP1       0x00000002  // Analog comparator 1 event
#define ADC_TRIGGER_COMP2       0x00000003  // Analog comparator 2 event
#define ADC_TRIGGER_EXTERNAL    0x00000004  // External event
#define ADC_TRIGGER_TIMER       0x00000005  // Timer event
#define ADC_TRIGGER_PWM0        0x00000006  // PWM0 event
#define ADC_TRIGGER_PWM1        0x00000007  // PWM1 event
#define ADC_TRIGGER_PWM2        0x00000008  // PWM2 event
#define ADC_TRIGGER_PWM3        0x00000009  // PWM3 event
#define ADC_TRIGGER_NEVER       0x0000000E  // Never Trigger
#define ADC_TRIGGER_ALWAYS      0x0000000F  // Always event
#define ADC_TRIGGER_PWM_MOD0    0x00000000  // PWM triggers from PWM0
#define ADC_TRIGGER_PWM_MOD1    0x00000010  // PWM triggers from PWM1
```

1.11.4　实验现象

实验现象如图 8.13 所示。

图 8.13 实验结果

2　模拟比较器（AC）

在数字电路中，经常需要对两个位数相同的二进制数进行比较，以判断它们的相对大小或者是否相等，用来实现这一功能的逻辑电路就称为数值比较器（也称数字比较器），Tiva C 直接集成到 ADC 单元里了。而在模拟电路中，对两个模拟值进行比较要用到模拟比较器，它能比较两个模拟电压的大小，并提供一个逻辑输出来表示比较结果。主要用来监控系统模拟电路的一些突发情况，也可以集合软件，实现比较简单的模/数转化。

2.1　AC 比较功能

AC 和 ADC 是单片机内部最常见的两种支持模拟信号输入的功能接口。模拟比较器将两个模拟量（电压值）进行比较，当同相端（+端）的输入电压高于反相端（-端）的输入电压时，输出高（或低）电平；反之，则输出低（或高）电平。AC 原理图和逻辑功能图，请分别参看图 8.14 和表 8.4。最终的逻辑输出可以产生一个中断信号，也可以用于触发 ADC 模块的转换。中断信号的产生和 ADC 触发信号的产生时分开且相互独立的，例如，可以在上升沿产生中断信号，在下降沿产生 ADC 触发信号。

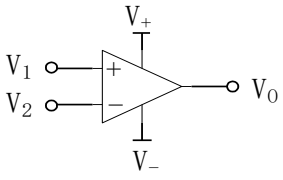


图 8.14 AC 原理图

表 8.4

电压值	输出
V1>V2	VO=1
V1<V2	VO=0

TM4C1230H6PM 的两个模拟比较器均具有以下三种比较方式：

（1）　比较两个外部输入引脚 Cn-和 Cn+的输入电压；

（2）　比较外部输入引脚 Cn-的输入电压和内部参考电压 VREF；

（3） 比较外部输入引脚 Cn-的输入电压和公共外部参考电压即 C0+的输入电压。

设外部输入引脚 Cn-的输入电压为 VIN-（图 8.14 中 V2），被比较的电压为 VIN+图 8.14 中 V1，则模拟比较器的输入输出之间关系见表 8.4。即：

若 VIN-<VIN+，则 VOUT=1；

若 VIN->VIN+，则 VOUT=0。

三种不同比较方式中，负输入都是 Cn-的输入电压，正输入各有不同，通过配置寄存器 ACCTL 的 ASRCP 位来进行选择。

2.2 TM4C1230H6PM AC 的内部结构

TM4C1230H6PM 微控制器提供两个独立集成的模拟比较器。每个模拟比较器模块的结构框图如图8.15所示，主要包括三部分，即寄存器、比较器主模块以及内部参考电压生成装置。

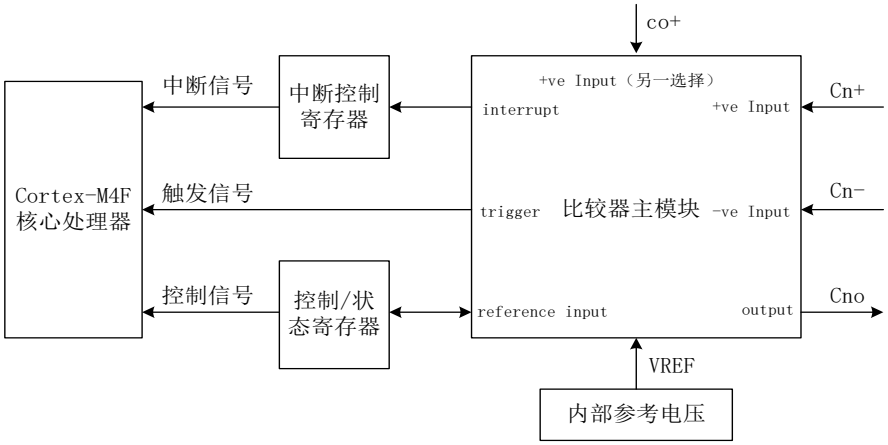


图8.15 模拟比较器结构框图

主要有两种寄存器：中断控制寄存器以及控制/状态寄存器。中断控制寄存器包括3个寄存器ACMIS(屏蔽中断状态)、ACRIS(原始中断状态)和ACINTEN(中断使能)，主要用于模拟比较器模块输出逻辑电平所产生中断信号的查询和控制。而控制/状态寄存器包括控制寄存器ACCTL和状态寄存器ACSTAT，用于模拟比较器的配置和状态的监控。

这些引脚分别对应GPIO的部分能输入模拟量的引脚如表8.5。

表8.5 模拟比较器外部信号（64LQFP）

引脚名称	引脚编号	引脚复用/引脚分配	引脚类型	缓冲类型	描 述
C0+	14	PC6	I	模拟	模拟比较器0正输入
C0-	13	PC7	I	模拟	模拟比较器0负输入
C0O	28	PF0(9)	O	TTL	模拟比较器0输出
C1+	15	PC5	I	模拟	模拟比较器1正输入
C1-	16	PC4	I	模拟	模拟比较器1负输入
C1O	29	PF1(9)	O	TTL	模拟比较器1输出

比较器单元结构如图8.16所示。

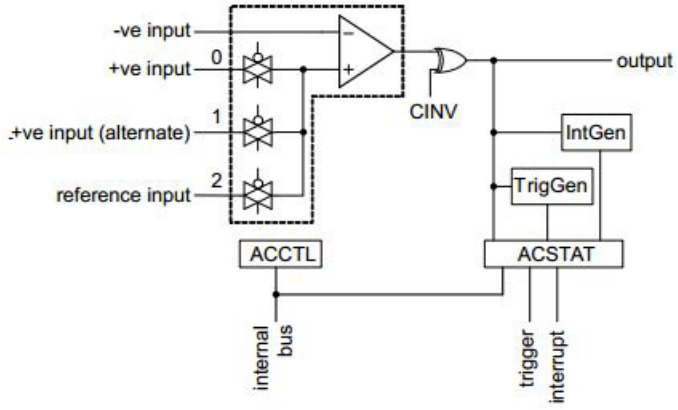


图8.16 比较器单元结构

内部参考电压生成装置用于生成一个用作比较的内部参考电压V_{IREF}，其结构如图8.17所示。Decoder为一4-16译码器结构（只有一个输出有效）。

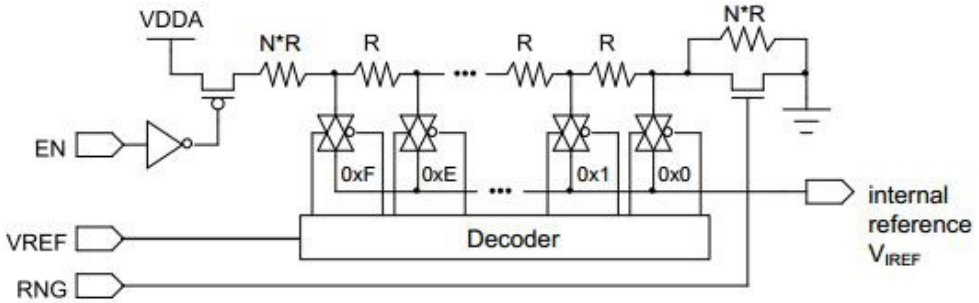


图8.17 内部参考电压生成结构

其中：VREF: 0x0---0xF； VDDA = 3.3v

RNG=0	VIREF = VDDA / 4.2 + VREF * (VDDA/ 29.4)
RNG=1	VIREF = VREF * (VDDA/ 22.12)

2.3 中断和 ADC 触发控制

模拟比较器的逻辑输出可以用于产生一个中断信号或 ADC 触发信号。通过设置寄存器 ACINTEN 的 INn 位可以分别使能每个模拟比较器模块的中断，通过设置寄存器 ACCTL 的 ISLVAL、ISEN 位可以设置中断触发模式。模拟比较器作为 ADC 触发信号的使能和触发模式分别通过寄存器 ACCTL 的 TOEN、TSLVAL、TSEN 位配置。

2.4 初始化和配置

下面的例子给出，如何配置模拟比较器从内部寄存器读出它的输出值。

1.调用SysCtlPeripheralEnable（）函数使能模拟比较器时钟；

2.通过调用SysCtlPeripheralEnable（）函数使能GPIO 模块；

3.调用GPIOPinTypeComparator（）、GPIOPinTypeGPIOInput（）函数使能GPIO 端

口，并配置相关引脚为输入；

4.配置GPIOPCTL 寄存器的PMCn 位，将模拟比较器输出信号分配给相应的引脚。

5.调用ComparatorRefSet（）函数配置内部参考电压为1.65 V；

6.调用ComparatorConfigure（）函数配置比较器来使用内部参考电压，并且不让输出翻转；

7.延迟10μs 。

8.调用ComparatorValueGet（）函数读取比较器输出值。

改变比较器负极输入信号C-的电平以观察OVAL值的变化。

2.5 AC 模块基础实验

以下举例说明模拟比较器的功能和使用。

2.5.1 实验目的和流程图

本实验介绍了应用Tiva C 系列芯片AC 模块进行电压采集，并与已设定的比较电压进行比较。AC配置：无触发模式，双沿触发中断，将参考选为比较器的内部参考电压，不翻转输出，选用模拟比较器0。

该实验流程如图8.18所示：

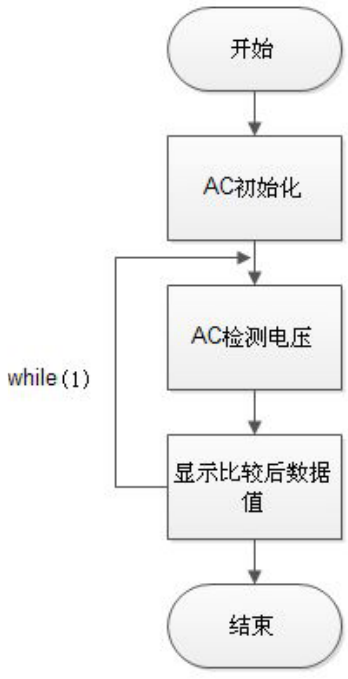


图 8.18 程序流程图

2.5.2 实验代码设计

```
//AC 初始化程序
void AC_Init()
{
```

```
    SysCtlPeripheralEnable(SYSCTL_PERIPH_COMP0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    GPIOPinTypeComparator(GPIO_PORTC_BASE, GPIO_PIN_7);
    ComparatorRefSet(COMP_BASE, COMP_REF_1_65V);
    ComparatorConfigure(COMP_BASE, 0,(COMP_TRIG_NONE | COMP_INT_BOTH |
    COMP_ASRCP_REF | COMP_OUTPUT_NORMAL));
    delay(200000);
}
//读取AC 转换后的值
void AC_detect()
{
    bool temp;
    temp = ComparatorValueGet(COMP_BASE, 0);
    if(temp == false)
    num1 = 0;
    else
    num1 = 1;
}
//AC 测试实验主程序
void AC_TEST()
{
    while(1)
    {
    AC_detect();
    My_printf(4,0,"AC=%d",num1);
    }
    LCD_Clear();
}
```

2.5.3 库函数简述

Tiva 开发板提供了相关的API 函数，方便用户快速开发AC 模块。

2.5.3.1 函数 ComparatorRefSet

功能：设置内部参考电压

原型：void ComparatorRefSet(uint32_t ui32Base,uint32_t ui32Ref)

参数：ui32Base 是比较模块的基地址、ui32Ref 需要的参考电压

参数ui32Config 取以下值之一：

COMP_REF_OFF —— 关闭参考电压

COMP_REF_0V ——设置参考电压为0V

COMP_REF_0_1375V ——设置参考电压为0.1375 V

……

2.5.3.2 函数 ComparatorConfigure

功能：配置比较器

原型：void ComparatorConfigure(uint32_t ui32Base,uint32_t ui32Comp,

uint32_t ui32Config）

参数：ui32Base 是比较模块的基地址、ui32Comp 是比较器的配置索引、ui32Config 是比较器的配置，参数ui32Config 的值为COMP_TRIG_xxx, COMP_INT_xxx,COMP_ASRCP_xxx,和 COMP_OUTPUT_xxx 的值的逻辑与。具体请参考《数据手册》。

2.5.3.3 函数 ComparatorValueGet

功能：获得当前比较器输出值

原型：bool ComparatorValueGet(uint32_t ui32Base,uint32_t ui32Comp)

参数：ui32Base是比较模块的基地址

ui32Comp 是比较器的配置索引

描述：该函数取出当前比较器的输出值

返回：当比较器输出为高时，返回 true；当比较器输出为低时，返回为 false。

3 数模转换（DAC）

数模转换器（DAC）完成数字量到模拟量的转换。Tiva C 系列 MCU 中没有 DAC 外设，DY 实验板上采用了一片 DAC7512 芯片作为扩展。DAC7512 是 T I 公司生产具有内置缓冲放大器低功耗单片 12 位数模转换器。其最高支持 30M 时钟的通用三线串行 SPI 接口。

3.1 D/A 变换原理

将数字量变换成模拟电压输出，直接输出的模拟电压值是离散的，如果想要得到连续的波形输出，应该增加低通滤波环节。

D/A 变换的位数决定了变换的精度，对于 12 位的 DAC7512 来说，12 位全 1（即 0xFFF），应该对应于芯片所接 Vref 电压，所以对任一 12 位的二进制数 X，应该对应一个输出电压 Vo，为：

$$V_O = \frac{X}{2^{12}-1} \times V_{ref} = \frac{X}{4095} \times V_{ref}(V)$$

3.2 DAC7512 芯片

其片内高精度输出放大器可获得满幅（供电电源电压与地电压间）任意输出。DAC7512 带有一个时钟达 30MHz 通用三线串行接口，与 TM4C123GH6PM 连接是一款典型的 SPI 应用。其接口与 SPI、QSPI、Microwire 及 DSP 接口兼容，因而可与 intel 系列单片机、Motorola 系列单片机直接连接而无需任何其它接口电路。

由于 DAC7512 串行数模转换器可选择供电电源来作为参考电压，因而具有很宽动态输出范围，此外，DAC7512 数模转换器还具有三种关断工作模式。正常工作状态下，DAC7512 在 5 V电压下功耗仅为 0．7 mW，而省电状态下功耗为 1 μW。因此，低功耗 DAC7512 无疑是便携式电池供电设备理想器件。

3.2.1 DAC7512 组成框图

DAC7512 组成框图如图 8.19 所示。图中输入控制逻辑用于控制 DAC 寄存器写操作，掉电控制逻辑与电阻网络一起用来设置器件工作模式，即选择正常输出还是把输出端与缓冲放大器断开，而接入固定电阻。芯片内缓冲放大器具有满幅输出特性，可驱动 2 kΩ 及 1000 pF 并联负载。

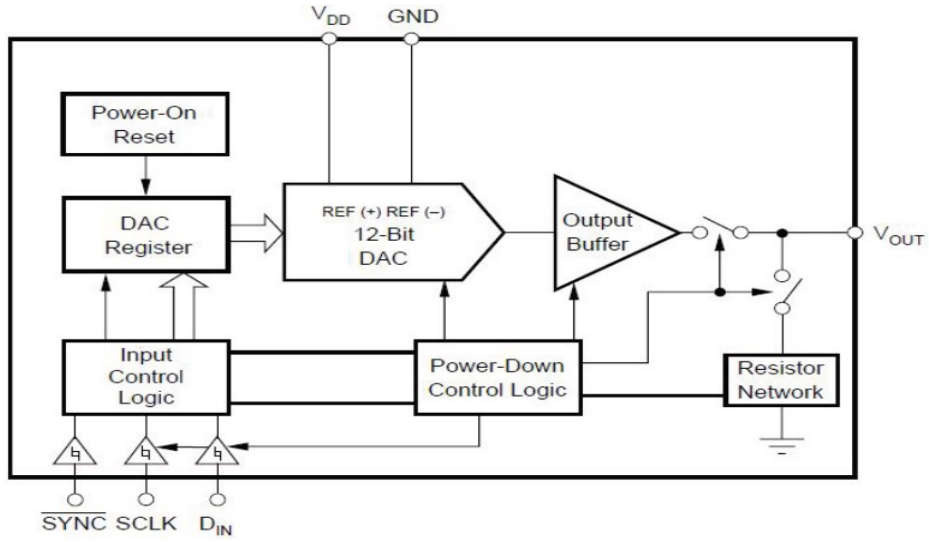


图 8.19 DAC7512 组成框图

3.2.2 DAC7512 主要特点

DAC7512 具有如下特点：

- 微功耗，5 V时工作电流消耗为 135 μA（DAC7512）；
- 在掉电模式时，如果采用 5 V电源供电，其电流消耗为 135 nA，而采用 3 V供电时，其电流消耗仅为 50 nA；
- 供电电压范围为＋2．7 V～＋5．5 V；
- 上电输出复位后输出为 0 V；
- 具有三种关断工作模式可供选择，5 V电压下功耗仅为 0．7 mW；

- 带有低功耗施密特输入串行接口；
- 内置满幅输出缓冲放大器；
- 具有ＳＹＮＣ中断保护机制。

3.2.3 DAC7512 工作特性

3.2.3.1 DAC7512 工作时序

接口工作模式：

DAC7512 采用三线制（/SYNC，SCLK 及 Din）串行接口。其串行写操作时序如图 8.20 所示。写操作开始前，/SYNC 要置低，Din 数据在串行时钟 SCLK 下降沿依次移入１６位寄存器。在串行时钟第１６个下降沿到来时，将最后一位移入寄存器，可实现对工作模式设置及ＤＡＣ内容刷新，从而完成一个写周期操作。此时，/SYNC 可保持低电平或置高，但在下一个写周期开始前，/SYNC 必须转为高电平并至少保持３３ｎｓ 以便/SYNC 有时间产生下降沿来启动下一个写周期。若/SYNC 在一个写周期内转为高电平，则本次写操作失败，寄存器强行复位。由于施密特缓冲器在/SYNC 高电平时电流消耗大于低电平时电流消耗，因此，在两次写操作之间，应把/SYNC 置低以降低功耗。

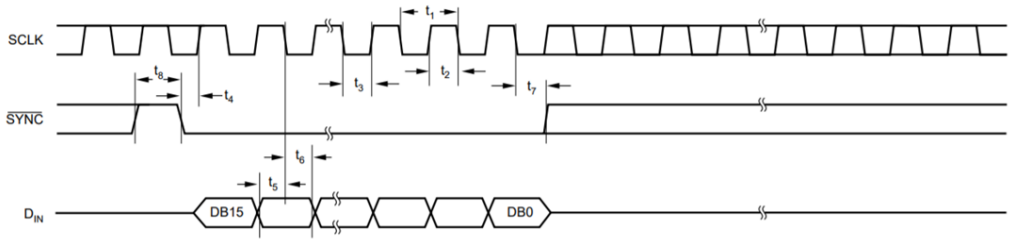


图 8.20 DAC7512 工作时序

DAC7512 片内移位寄存器宽度为１６位，其中 DB１５、DB１４是空闲位，DB１３、DB１２是工作模式选择位、DB１１～DB０是数据位。器件内部带有上电复位电路。上电后，寄存器置０，所以 DAC7512 处于正常工作模式，模拟输出电压为０Ｖ。

DAC7512 四种工作模式可由寄存器内 DB１３、DB１２来控制。其控制关系如下图所示：DAC7512 的工作模式掉电模式下，不仅器件功耗要减小，而且缓冲放大器的输出级通过内部电阻网络接到１ｋΩ、１００ｋΩ或开路。而处于掉电模式时，所有的线性电路都断开，但寄存器内的数据不受影响。

3.2.3.2 DAC7512 与 TM4C123GH6PM 实验板的连接图

德研实验板上使用了 DAC7512 芯片以实现 D/A 功能，如图 8.21 所示。

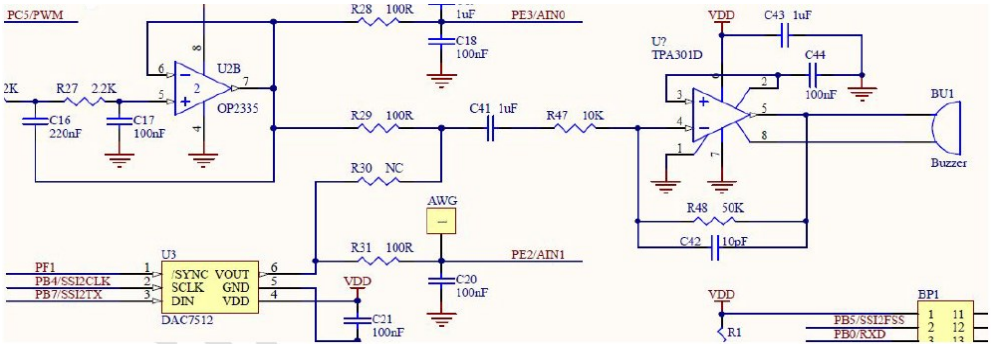


图 8.21 德研实验板 7512 线路图

3.2.3.3 部分程序

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);//SYNC 信号
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1);
GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 1 << 1);
SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI1);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
GPIOPinConfigure(GPIO_PD0_SSI1CLK);
GPIOPinConfigure(GPIO_PD3_SSI1TX);
GPIOPinTypeSSI(GPIO_PORTD_BASE, GPIO_PIN_3 | GPIO_PIN_0);
//配置 SSI1 为主机，工作模式为 FRF_MOTO_MODE_1，比特率 10000，数据位数为 16
SSIConfigSetExpClk(SSI1_BASE, SysCtlClockGet(),
SSI_FRF_MOTO_MODE_1,SSI_MODE_MASTER, 10000, 16);
SSIEnable(SSI1_BASE);
void dac_writedata(uint16_t ui32Data)
{
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0 << 1);//拉低 SYNC 信号
    SSIDataPut(SSI1_BASE, ui32Data);//发送数据
    while (SSIBusy(SSI1_BASE))//等待到发送完
    {
    }
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 1 << 1);//拉高 SYNC 信号
}
```

【课堂练习】

基础任务：

1、测量德研实验板上 ADC 管脚上 0~3.3V 的模拟电压，并将实际测量值显示出来。

2、在德研实验板上用DAC7512芯片编程实现一个三角波信号发生器的输出。要求输出波形频率为500Hz，幅度为0~3.3V。用示波器观察。

下一次任务：与自己项目相关的A/D转换任务（每组需课前提交题目让老师审核）

【课前预习】

1、 熟悉德研实验板ADC的硬件线路，完成TIVA-DY-demo模块里的ADC-control-light例程操作，说明实验现象。
2、 熟悉德研实验板DAC的硬件线路，完成TIVA-DY-demo模块里的DAC例程操作，说明实验现象。
3、 如果想用德研实验板上的模拟比较器做实验，应该注意什么？
4、 请说明ADC采样差分采样的好处。
5、 请说明Tiva C序列发生器的作用。

【作业】

1、 在VREF=3.3V的D/A转换电路中，如果器件是12位的，对应于0x2FD数字量，会转换输出多少V的电压？
2、 简述Tiva C模拟比较器的基本结构，在开发板上编程实现电位器可变电压与设定2V电压进行比较，结果通过UART显示。