A/D 转换芯片 ADC0832 的应用

作者：杜洋

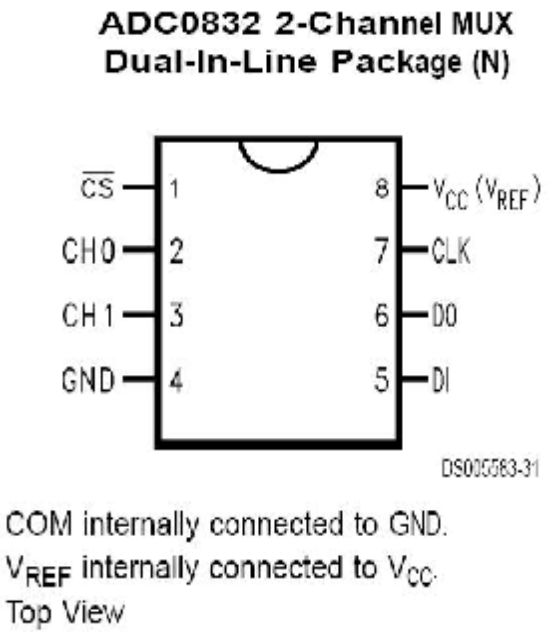
2005 年 10 月 11 日

ADC0832 是美国国家半导体公司生产的一种 **8** 位分辨率、双通道 **A/D** 转换芯片。由于它体积小，兼容性强，性价比高而深受单片机爱好者及企业欢迎，其目前已经有很高的普及率。学习并使用 **ADC0832** 可是使我们了解 **A/D** 转换器的原理，有助于我们单片机技术水平的提高。

ADC0832 具有以下特点：

- **8** 位分辨率；
- 双通道 **A/D** 转换；
- 输入输出电平与 **TTL/CMOS** 相兼容；
- **5V** 电源供电时输入电压在 **0~5V** 之间；
- 工作频率为 **250KHZ**，转换时间为 **32μS**；
- 一般功耗仅为 **15mW**；
- **8P**、**14P**—**DIP**（双列直插）、**PICC** 多种封装；
- 商用级芯片温宽为**0°C to +70°C**，工业级芯片温宽为**−40°C to +85°C**；

芯片顶视图：（图1、图2）



Top View

图 1

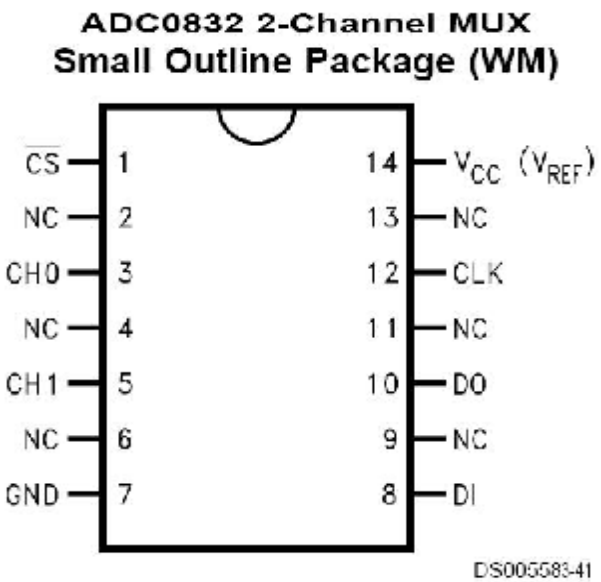


图 2

芯片接口说明：

- CS_　　片选使能，低电平芯片使能。
- CH0　　模拟输入通道 0，或作为 IN+/-使用。
- CH1　　模拟输入通道 1，或作为 IN+/-使用。
- GND　　芯片参考 0 电位（地）。
- DI　　数据信号输入，选择通道控制。
- DO　　数据信号输出，转换数据输出。
- CLK　　芯片时钟输入。
- Vcc/REF　电源输入及参考电压输入（复用）。

ADC0832 与单片机的接口电路：

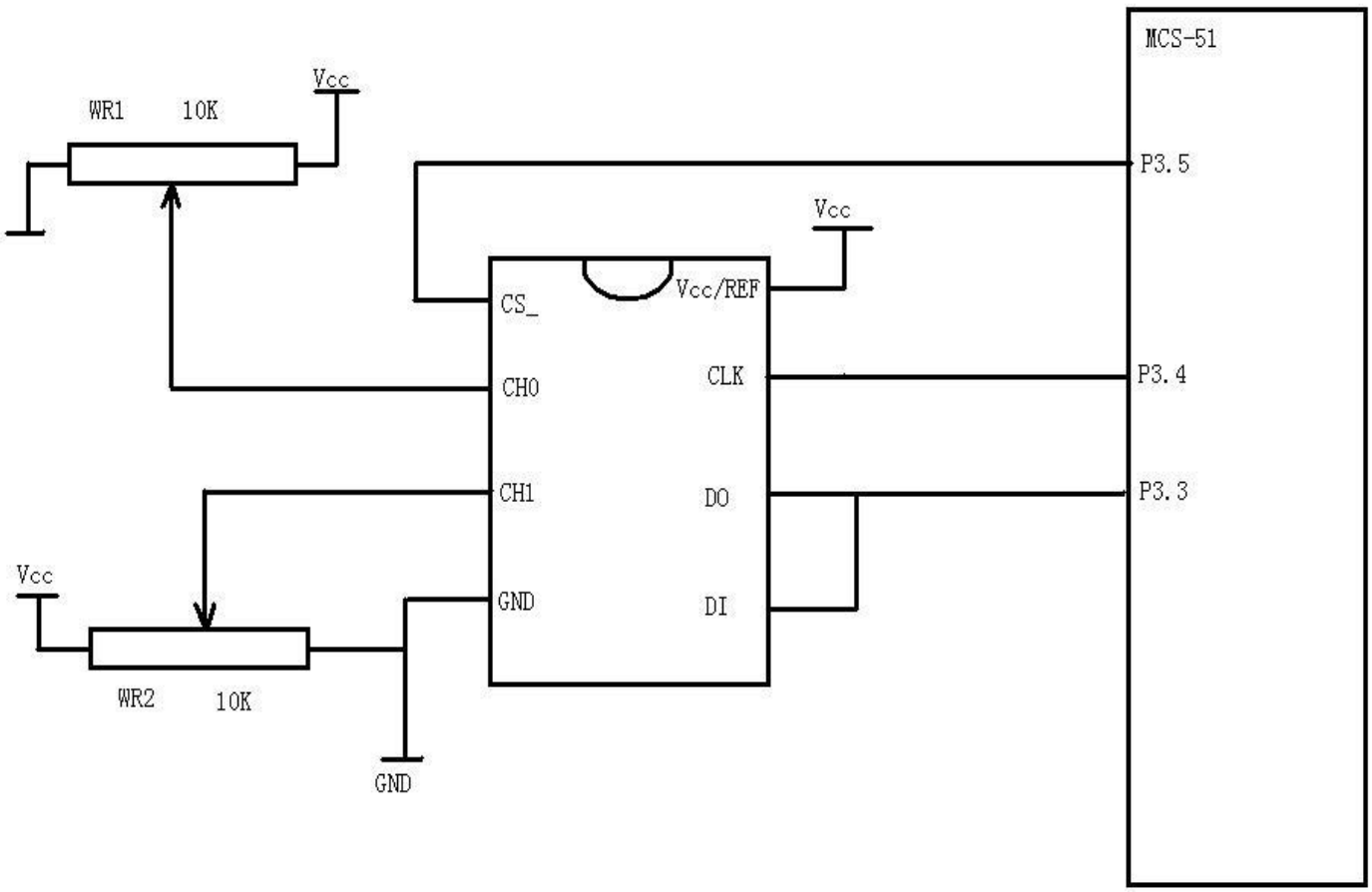


图 3

ADC0832 为 8 位分辨率 A/D 转换芯片，其最高分辨可达 256 级，可以适应一般的模拟量转换要求。其内部电源输入与参考电压的复用，使得芯片的模拟电压输入在 0~5V 之间。芯片转换时间仅为 32μS，据有双数据输出可作为数据校验，以减少数据误差，转换速度快且稳定性能强。独立的芯片使能输入，使多器件挂接和处理器控制变的更加方便。通过 DI 数据输入端，可以轻易的实现通道功能的选择。

单片机对 ADC0832 的控制原理：

正常情况下 ADC0832 与单片机的接口应为 4 条数据线，分别是 CS、CLK、DO、DI。但由于 DO 端与 DI 端在通信时并未同时有效并与单片机的接口是双向的，所以电路设计时可以将 DO 和 DI 并联在一根数据线上使用。（见图 3）

当 ADC0832 未工作时其 CS 输入端应为高电平，此时芯片禁用，CLK 和 DO/DI 的电平可任意。当要进行 A/D 转换时，须先将 CS 使能端置于低电平并且保持低电平直到转换完全结束。此时芯片开始转换工作，同时由处理器向芯片时钟输入端 CLK 输入时钟脉冲，DO/DI 端则使用 DI 端输入通道功能选择的数据信号。在第 1 个时钟脉冲的下沉之前 DI 端必须是高电平，表示启始信号。在第 2、3 个脉冲下沉之前 DI 端应输入 2 位数据用于选择通道功能，其功能项见表 1。

TABLE 6. MUX Addressing: ADC0832 Single-Ended MUX Mode

MUX Address		Channel #	
SGL/$\overline{\text{DIF}}$	ODD/SIGN	0	1
1	0	+	
1	1		+

COM is internally tied to A GND

TABLE 7. MUX Addressing: ADC0832 Differential MUX Mode

MUX Address		Channel #	
SGL/$\overline{\text{DIF}}$	ODD/SIGN	0	1
0	0	+	−
0	1	−	+

表 1

如表 1 所示，当此 2 位数据为“1”、“0”时，只对 CH0 进行单通道转换。当 2 位数据为“1”、“1”时，只对 CH1 进行单通道转换。当 2 位数据为“0”、“0”时，将 CH0 作为正输入端 IN+，CH1 作为负输入端 IN-进行输入。当 2 位数据为“0”、“1”时，将 CH0 作为负输入端 IN-，CH1 作为正输入端 IN+进行输入。

到第 3 个脉冲的下沉之后 DI 端的输入电平就失去输入作用，此后 DO/DI 端则开始利用数据输出 DO 进行转换数据的读取。从第 4 个脉冲下沉开始由 DO 端输出转换数据最高位 DATA7，随后每一个脉冲下沉 DO 端输出下一位数据。直到第 11 个脉冲时发出最低位数据 DATA0，一个字节的数据输出完成。也正是从此位开始输出下一个相反字节的数据，即从第 11 个字节的下沉输出 DATD0。随后输出 8 位数据，到第 19 个脉冲时数据输出完成，也标志着一次 A/D 转换的结束。最后将 CS 置高电平禁用芯片，直接将转换后的数据进行处理就可以了。更详细的时序说明请见表 2。

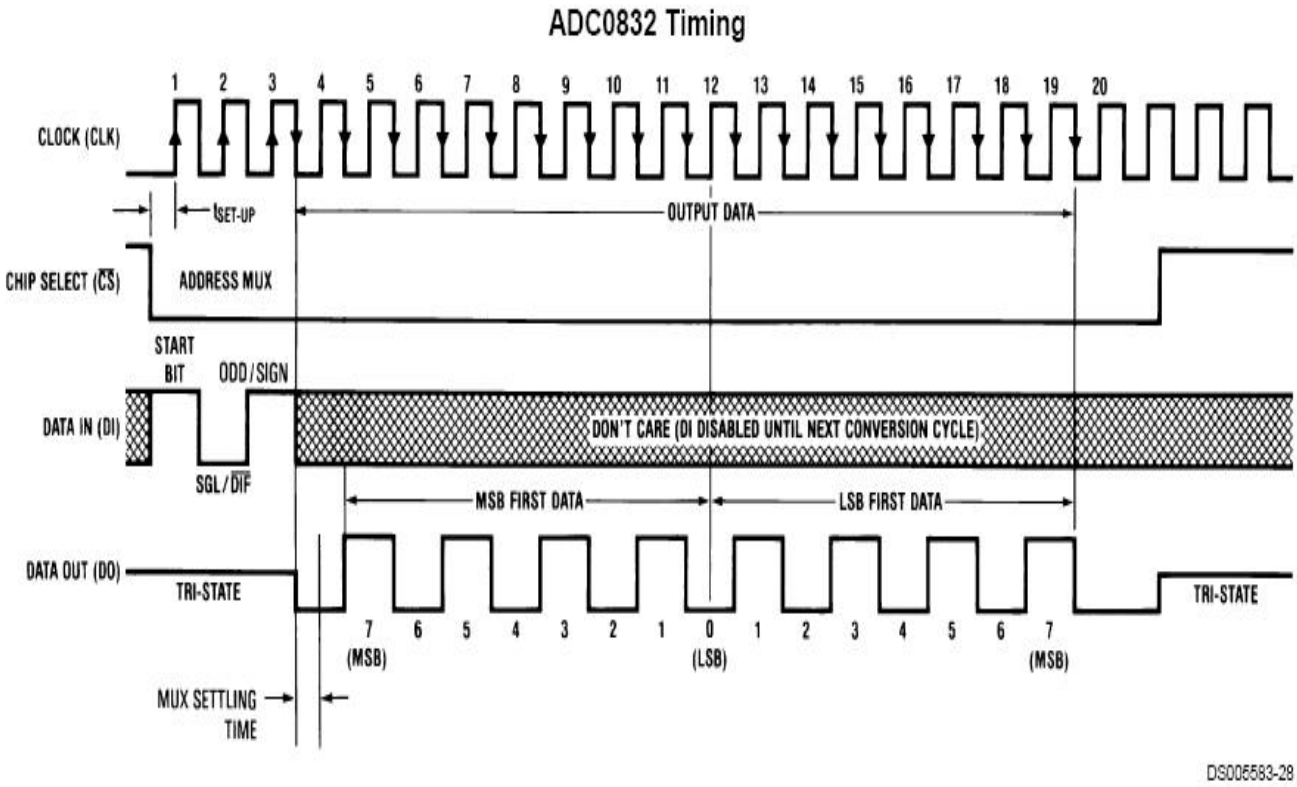


表 2

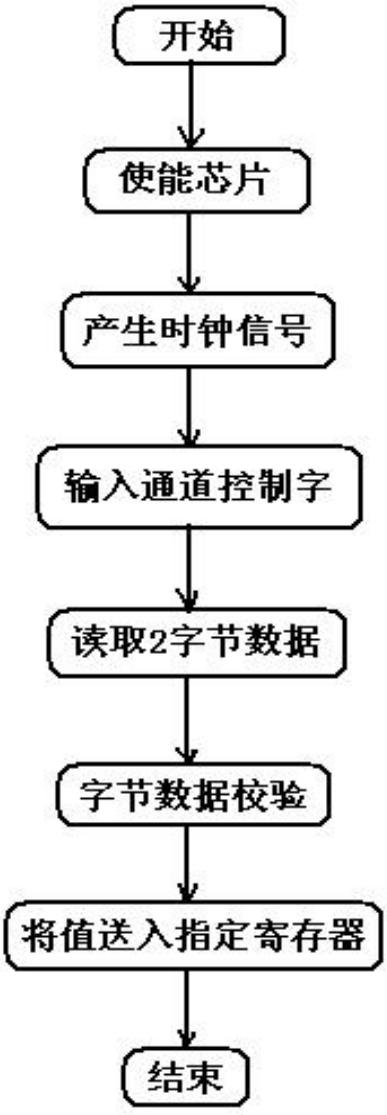
作为单通道模拟信号输入时 ADC0832 的输入电压是 0~5V 且 8 位分辨率时的电压精度为 19.53mV。如果作为由 IN+与 IN-输入的输入时，可是将电压值设定在某一个较大范围之内，从而提高转换的宽度。但值得注意的是，在进行 IN+与 IN-的输入时，如果 IN-的电压大于 IN+的电压则转换后的数据结果始终为 00H。

ADC0832 芯片接口程序的编写：

为了高速有效的实现通信，我们采用汇编语言编写接口程序。由于 ADC0832 的数据转换时间仅为 32μS，所以 A/D 转换的数据采样频率可以很快，从而也保证的某些场合对 A/D 转换数据实时性的要求。数据读取程序以子程序调用的形式出现，方便了程序的移植。

程序占用资源有累加器 A，工作寄存器 R7，通用寄存器 B 和特殊寄存器 CY。通道功能寄存器和转换值共用寄存器 B。在使用转换子程序之前必须确定通道功能寄存器 B 的值，其赋值语句为“MOV　B,#data”（00H~03H）。运行转换子程序后的转换数据值被放入 B 中。子程序退出后即可以对 B 中数据处理。

ADC0832 数据读取程序流程：



ADC0832 芯片接口程序[汇编]：

```
/*-----------------------------------------
子程序名：      ADC0832 子程序
编写人：        杜洋
初写时间：      2005 年 10 月 10 日
程序功能：      将模拟电压量转换成数字量
实现方法：      串行通信。
CPU 说明：      MCS-51
植入说明：      占用 A、B、CY、R7
-----------------------------------------*/
;以下接口定义根据硬件连线更改
ADCS    BIT   P3.5          ;使能接口
ADCLK   BIT   P3.4          ;时钟接口
ADDO    BIT   P3.3          ;数据输出接口（复用）
ADDI    BIT   P3.3          ;数据输入接口

;以下语句在调用转换程序前设定
MOV     B,#00H      ;装入通道功能选择数据值

;以下为 ADC0832 读取数据子程序
;==== ADC0832 读数据子程序====
ADCONV:
    SETB    ADDI            ;初始化通道选择
    NOP
    NOP
    CLR     ADCS            ;拉低/CS 端
    NOP
    NOP
    SETB    ADCLK           ;拉高 CLK 端
    NOP
    NOP
    CLR     ADCLK           ;拉低 CLK 端,形成下降沿
    MOV     A,B
    MOV     C,ACC.1         ;确定取值通道选择
    MOV     ADDI,C
    NOP
    NOP
    SETB    ADCLK           ;拉高 CLK 端
    NOP
    NOP
    CLR     ADCLK           ;拉低 CLK 端,形成下降沿 2
    MOV     A,B
    MOV     C,ACC.0         ;确定取值通道选择
    MOV     ADDI,C
    NOP
```

```
        NOP
        SETB    ADCLK           ;拉高 CLK 端
        NOP
        NOP
        CLR     ADCLK           ;拉低 CLK 端,形成下降沿 3
        SETB    ADDI
        NOP
        NOP
        MOV     R7,#8           ;准备送下后 8 个时钟脉冲
AD_1:
        MOV     C,ADDO          ;接收数据
        MOV     ACC.0,C
        RL      A               ;左移一次
        SETB    ADCLK
        NOP
        NOP
        CLR     ADCLK           ;形成一次时钟脉冲
        NOP
        NOP
        DJNZ    R7,AD_1         ;循环 8 次
        MOV     C,ADDO          ;接收数据
        MOV     ACC.0,C
        MOV     B,A
        MOV     R7,#8
AD_13:
        MOV     C,ADDO          ;接收数据
        MOV     ACC.0,C
        RR      A               ;左移一次
        SETB    ADCLK
        NOP
        NOP
        CLR     ADCLK           ;形成一次时钟脉冲
        NOP
        NOP
        DJNZ    R7,AD_13        ;循环 8 次
        CJNE    A,B,ADCONV      ;数据校验
        SETB    ADCS            ;拉高/CS 端
        CLR     ADCLK           ;拉低 CLK 端
        SETB    ADDO            ;拉高数据端,回到初始状态
        RET
;====子程序结束====
```