



成 绩 _____

北京航空航天大学
B E I H A N G U N I V E R S I T Y

微机原理及接口技术 实验报告

院（系）名称	高等理工学院
专 业 名 称	自动化
学 生 学 号	16231235
学 生 姓 名	李 谨 杰
指 导 教 师	林 新

2018 年 11 月

本实验的硬件连接线较为简单，除了电源线、与电脑连接的 USB 线外，只需要连接两处：

- 1) ADC0809 的模拟量输入端 IN0 与 电位计 RW1 的电压输出端 U1
- 2) ADC0809 的片选信号 CS 与 计算机端口控制端 290H~298H

2. 编程提示

1) ADC0809 转换过程

启动转换和输入转换后的数据，都是将端口号写入 DX 后直接使用 OUT 指令和 IN 指令即可。

等待转换结束有两种方法。一种是查询 ADC0809 的 EOC 状态，另一种是长时间延时。由于实验硬件连线不支持 EOC 状态查询，因此本实验采用后一种方法，通过编写和调用延时子程序 DELAY 等待 ADC0809 转换结束。

2) 十进制显示算法

数据显示允许采用多种方式。直接显示十进制数值的原理如下：

将待显示的字节数据（0~255）使用 DIV 指令除以 100，商（保存在 AL 中）即为十进制的百位数；余数（保存在 AH 中）复制到 AL 寄存器，AH 清零，再次使用 DIV 指令除以 10，得到的商即为十位数，余数为个位数。

然后将得到的百位、十位、个位数分别加上 30H，转换为 ASCII 码值进行显示输出。

3) 循环方法：使用键盘按键来控制循环

由于电压值需要反复读取和输出，因此程序需要循环操作。为了避免循环过快，导致数据显示快速刷新无法看清，因此需要对循环的速度和节奏加以控制。最简单的方法是使用 INT 21H 的 1 号功能调用。

每当一轮数据采集和显示结束后，调用 INT 21H 的 1 号功能，程序暂停等待键盘的输入。

如果输入的按键是 ESC 键则退出程序。否则重新开始一轮模数转换操作。

1) 延时程序

延时程序主要用于启动 ADC0809 转换后，等待转换结束。延时的原理是，每条指令的执行都需要时间，当反复执行某些指令时即可占用相当多的 CPU 时间，从而达到延时的目的。

为了加长延时时间，可以使用嵌套循环结构。

2) 有精力的同学可以增加其它数据显方法

✧ 显示电压值（精确到小数点后两位）

电压值显示算法的原理如下：采集到的数据 X 为 8 位自然二进制编码，范围 0~255，对应电压为 0~+5V。

因此，根据除法计算规则，需要显示的个位数为： $I = X * 5 / 255$ ；此外还需要保持余数 $J = X * 5 \% 255$ ，以便计算小数点后面的电压数据。

小数点后面第一位数据： $K = J * 10 / 255$ ；另有余数 $M = J * 10 \% 255$

小数点后面第二位数： $N = M * 10 / 255$

✧ 显示 16 进制数

模数转换后的数据直接显示其 16 进制数值，原理如下：

由于转换后的数据为 8 位，需要针对高 4 位和低 4 位分别进行显示，另外也需要针对 0~9 和 A~F 做出不同的处理。

首先显示高 4 位。将模数转换后的数据逻辑右移 4 位，只保留高 4 位。然后判断此高 4 位是否大于 9。如果大于 9，则表明其数值在 A~F 之间，需要加上 37H，转换为 ASCII 码值进行输出；否则加上 30H，转换为 ASCII 码值进行输出。

✧ 其显示二进制数

模数转换后的数据显示二进制数值的原理如下：

显示数据的二进制位顺序是从左到右，即先显示高位，再显示低位。

将待显示的数据进行逻辑左移。最高位被移出后进入到了标志寄存器的 CF 位中，因此判断 CF 位，如果为 1 则跳转到显示字符 '1' 的代码处，负责显示字符 '0'。

由于数据共有 8 位，因此上述移位、判断和显示操作需要循环 8 次。

✧ 其它循环方法

① 按一定时间间隔显示采集到的数据。使用延时程序，精确调整参数，使得延时时间在可接受的范围内。

② 仅显示变化的数据。在此方式下每采集一次数据，都和上次的采集结果 DTIN 进行比较，如果相同则不显示，不同才进行显示。

2. 流程图

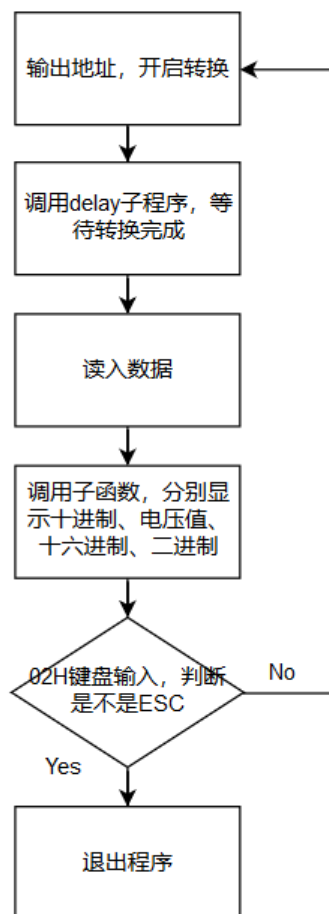


图 1. 主流程图

3. 原理图

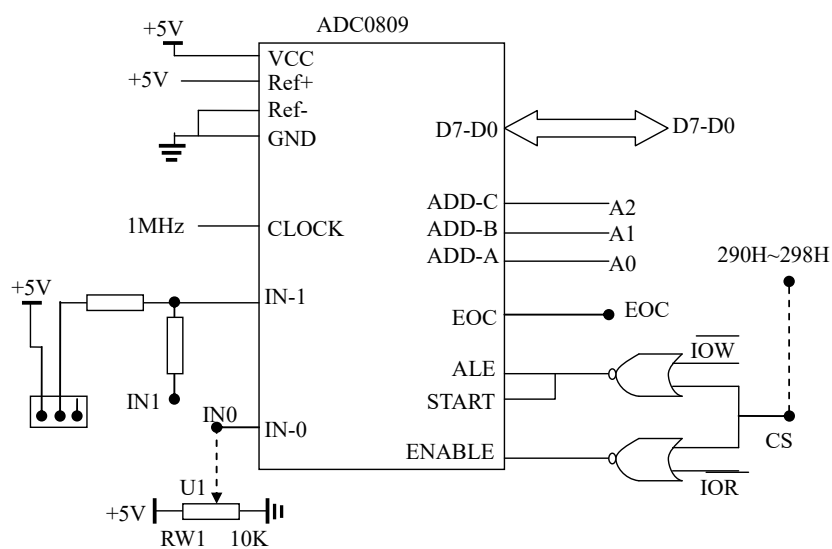


图 2. 实验硬件原理图

三、预习思考题及实验验证

1) 显示 ASCII 码字符的 DOS 功能调用有哪些，它们的区别是什么？

02H: 显示单个字符。

06H: 直接控制台 I/O。不会停下来等待用户输入，而是会直接跳过，因而往往和条件判断相结合，可以用于在不断循环的进程中根据输入跳出。

09H: 显示以\$结尾的字符串。区别在于输出的字符个数。

2) 如果试图将一个字节的二进制数分别以二进制数和十六进制方式显示在计算机屏幕上，需要做哪些处理？试写出处理思路并画出流程图。(提示：使用移位或循环移位指令)

二进制：

显示数据的二进制位顺序是从左到右，即先显示高位，再显示低位。

将待显示的数据进行逻辑左移。最高位被移出后进入到了标志寄存器的 CF 位中，因此判断 CF 位，如果为 1 则跳转到显示字符‘1’的代码处，负责显示字符‘0’。

由于数据共有 8 位，因此上述移位、判断和显示操作需要循环 8 次。

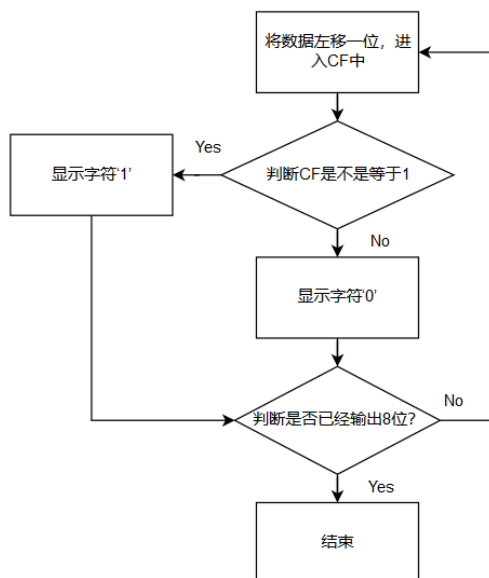


图 3. 二进制流程图

16 进制：

由于转换后的数据为 8 位，需要针对高 4 位和低 4 位分别进行显示，另外也需要针对 0~9 和 A~F 做出不同的处理。

首先显示高 4 位。将模数转换后的数据逻辑右移 4 位，只保留高 4 位。然后判断此高 4 位是否大于 9。如果大于 9，则表明其数值在 A~F 之间，需要加上 37H，转换为 ASCII 码值进行输出；否则加上 30H，转换为 ASCII 码值进行输出。

之后显示前四位，操作与上边类似。

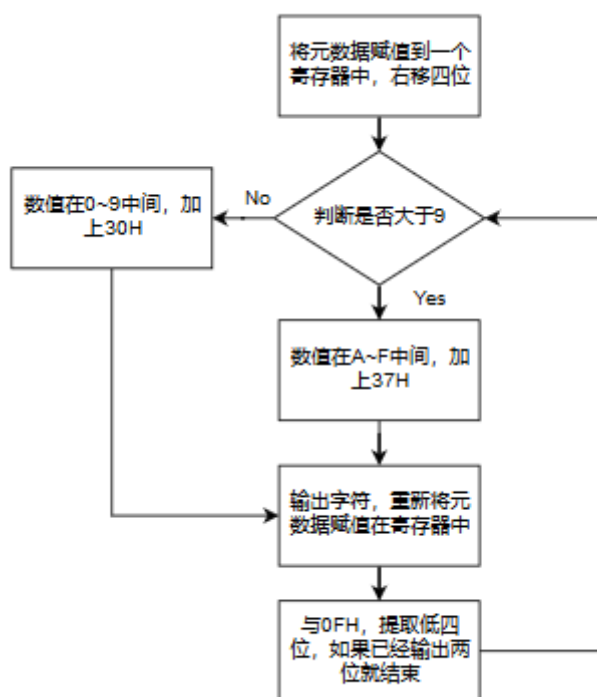


图 4. 十六进制流程图

3) 如何将一个字节的二进制数转换为十进制数显示出来？试写出处理思路并画出流程图。（提示：可以使用除法指令）

将待显示的字节数据（0~255）使用 DIV 指令除以 100，商（保存在 AL 中）即为十进制的百位数；余数（保存在 AH 中）复制到 AL 寄存器，AH 清零，再次使用 DIV 指令除以 10，得到的商即为十位数，余数为个位数。

然后将得到的百位、十位、个位数分别加上 30H，转换为 ASCII 码值进行显示输出。

这个题文字叙述得很清楚了，画流程图并不会将逻辑过程变简洁，因此省略流程图。

4) 设数据段中定义变量：DT1 DB 5CH。试编程在屏幕上显示其对应的二进制数、十进制数和十六进制数。

已完成，由于该程序大部分和主程序相同而且篇幅较长，暂不粘贴。

四、实验源程序

<pre> DATAS SEGMENT VOL DB 1 DUP (0) TEN DB 3 DUP(0) DOT DB 10 DUP(0) DOTNUM DW 5 ;控制显示的小数 位数 MESS1 DB 0AH,0DH,"Decimal: ",'\$' MESS2 DB 0AH,0DH,"Voltage: ",'\$' MESS3 DB 0AH,0DH,"Binary: ",'\$' MESS4 DB 0AH,0DH,"Hexadecimal: ",'\$' MSEE5 DB 0AH,0DH,"Same value",'\$' DATAS ENDS STACKS SEGMENT STACK 'STACKS' STAPN DB 100 DUP(0) TOP EQU LENGTH STAPN STACKS ENDS CODES SEGMENT ASSUME CS:CODES,DS:DATAS,SS:STACKS </pre>	<pre> START: MOV AX,STACKS MOV SS,AX MOV SP,TOP MOV AX,DATAS MOV DS,AX AGA: MOV DX,0290H ;选择 000 通道输出 OUT DX,AL ;启动一次转换 CALL DELAY ;延迟 IN AL,DX ;读入转换之后的 数据 CMP AL,VOL JNE AGA2 MOV DX, OFFSET MESS5 MOV AH, 09H INT 21H JMP AGA3 AGA2: MOV VOL,AL CALL DECIMAL ;十进制输出 </pre>
--	---



CALL DOTP	;实际的电压	LOOP LOOP2
输出		RET
CALL BINP		DELAY ENDP
CALL HEXP		;=====
		=====
AGA3: MOV AH,08H ; 不回显		
INT 21HS		;===== 十 进 制 输 出
		=====
CMP AL,1BH ;与 esc 比较,是则		DECIMAL PROC NEAR
退出, 不是则开启下一次循环		MOV DX,OFFSET MESS1
JNE AGA		MOV AH,09H
		INT 21H
MOV AH,4CH		
INT 21H		MOV AH,00H ;求各位数字,
		保存在 TEN 中
;===== 延 迟		MOV AL,VOL
=====		MOV BL,100
DELAY1 PROC NEAR		DIV BL
MOV CX,0		MOV TEN[0],AL
LOOP1: LOOP LOOP1		MOV AL,AH
RET		MOV AH,0
DELAY1 ENDP		MOV BL,10
		DIV BL
DELAY PROC NEAR		MOV TEN[1],AL
MOV CX,0		MOV TEN[2],AH
LOOP2:		
PUSH CX		MOV CX,3 ; 转
CALL DELAY1 ;循环嵌套		变为字符并输出
POP CX ;这里一定要注意 CX 的改变		MOV SI,0



```
LOOP TEN:    MOV DL,TEN[SI]                MOV SI,1
            ADD DL,30H                      MOV BH,10
            MOV AH,2                        LOOP DOT1:  MUL BH    ;小数
            INT 21H                        DIV BL
            INC SI                          MOV DOT[SI],AL
            LOOP LOOP TEN                  MOV AL,AH
            RET                            MOV AH,0
DECIMAL ENDP                                INC SI
;=====                                LOOP LOOP DOT1
=====

;===== 输出实际电压                    MOV DL,DOT[0] ;输出部分
=====                                ADD DL,30H
                                MOV AH,2
                                INT 21H
                                MOV DL,'!'
                                MOV AH,2
                                INT 21H
                                MOV CX,DOTNUM
                                MOV SI,1
                                LOOP DOT2:  MOV DL,DOT[SI]
                                ADD DL,30H
                                MOV AH,2
                                INT 21H
                                INC SI
                                LOOP LOOP DOT2

MOV DX,OFFSET MESS2
MOV AH,09H
INT 21H

MOV AL,5 ;个位数
MUL VOL
MOV BL,0FFH
DIV BL

MOV DOT[0],AL
MOV AL,AH
MOV AH,0

MOV CX,DOTNUM
```



MOV DL,'V'	
MOV AH,2	MOV DL,'B'
INT 21H	MOV AH,2
	INT 21H
RET	
DOTP ENDP	POP AX
;=====	MOV VOL,AL
=====	
	RET
;===== 二 进 制 显 示	BINP ENDP
=====	;=====
BINP PROC NEAR	==
MOV DX,OFFSET MESS3	
MOV AH,09H	;===== 十 六 进 制 显 示
INT 21H	=====
	HEXP PROC NEAR
MOV AX,0	MOV DX,OFFSET MESS4
MOV AL,VOL	MOV AH,09H
PUSH AX	INT 21H
MOV CX,8	MOV DL,VOL
BIN: SHL VOL,1	MOV CL,4
JC S1	SHR DL,CL ;逻辑右移的计数值
MOV DL,'0'	不能用立即数
JMP O1	CMP DL,9
S1: MOV DL,'1'	JA ST1
O1: MOV AH,2	ADD DL,30H
INT 21H	JMP ST_1
LOOP BIN	ST1: ADD DL,37H

ST_1: MOV AH,02H

INT 21H

MOV DL,VOL

AND DL,0FH

CMP DL,9

JA ST2

ADD DL,30H

JMP ST_2

ST2: ADD DL,37H

ST_2: MOV AH,02H

INT 21H

MOV DL,'H'

MOV AH,2

INT 21H

RET

HEXP ENDP

CODES ENDS

END START

数。按下除 ESC 外任意键进行下一次采样，如果结果相同，屏幕上会显示“Same value”字样；如果结果不相同，会显示这次采样的数值。如果按下 ESC 键，程序会退出。

六、结果分析与实验结论

1. 对实验所得结果进行分析处理，总结实验结论。

本程序可以很好地完成将采样值转换为十进制、电压值、二进制和十六进制的任务，还能检查每次采样值是否相等，如果不等才会输出。

2. 实验过程中遇到的问题分析总结。

实验中遇到的主要问题是，在子程序中使用循环嵌套进行延时，无法结束循环。我一开始以为是汇编不允许子程序嵌套，后来经老师点拨才发现是计数值 CX 在调用过程中被重新赋值了，之后改为 PUSH 和 POP 保存一下就改好了。后来反思，我这是用 C 语言的思路写了这个循环嵌套，忽略了汇编语言中寄存器与 C 语言子程序变量的差异，从而导致错误。我以后一定要注意这一点！

七、实验后思考题

1. 如果通过查询 ADC0809 的 EOC 引脚状态来判断转换是否结束，则硬件应当如何连接，相应的程序如何编写？

硬件：EOC 引脚通过三态门连到 D7 线上，三态门的控制口连到一个 IO 口上，用以查询 EOC 的状态。假设 IO 口地址为 295H。

程序：

```
OUT  DX, AL                                NO_END: IN  AL, DX ; 检查
PUSH DX ; 保存 ADC0809 端 上升沿
口地址                                     TEST AL, 80H
MOV  DX, 295H                             JZ  NO_END
POLL: IN  AL, DX ; 检查下                 POPDX ; 转换完成,
降沿                                     开始读入数据
                                     TEST AL, 80H
                                     IN  AL, DX
                                     JNZ POLL
```

2. 是否有必要将采集后的自然二进制编码转换为对应的电压值进行存储？

没有必要。精确度上，因为电压值是近似值，永远不如源码准确；程序冗余度上，无论是二进制存储还是电压值存储，总流程都是采样、转码和输出三个过程，所以并没有减少运算和代码量。因此没有必要转换为对应的电压值存储。

八、收获、体会及建议

通过这次实验，我重温了 DIV 和 MUL 两个函数，复习了 ADC0809 的输出原理。我觉得之后可以将实验内容改为用 8255A 控制 ADC0809，不要把重点放在进制转换上，而应该放在对 ADC 和 8255A 工作原理的理解上，这样才能体现“Learning by doing”的实验思想。