

部分非编程设计作业参考答案

4.29 作业

一、学习例 03 和例 03-2（两例子见群文件），内均有两版源代码，汇编的和 C51 的，请学习掌握至少其中一种。对例 03 和例 03-2 回答问题，

（1）例 03 和例 03-2 中，分别哪个口是段选码输出口，哪个口是位选码输出口？

例 03 P0 口是段选码输出口 P2 口是位选码输出口

例 03-2 8155 的 PB 口是段选码输出口 8155 的 PA 口是位选码输出口

（2）51 单片机输出每一位 LED 的位选码的值分别是。列出对应程序中实现输出位选码操作的语句。

例 03，对于共阳极显示器，输出位选码是选中位给 1，未选中位给 0，比如 00000001、00000010 等。

对应语句：MOV R2, #10000000B ;R2 存放位选码，初始选中 LED 最右位

MOV A, R2

CLR C

RRC A ; 移位指向下一位

C51 版是：P2=1 << i; //输出位选码 00000001、 00000010、 0000010010000000

例 03-2，也是共阳极显示器，输出位选码是选中位给 1，未选中位给 0，比如 00000001、00000010 等。

汇编版

MOV R2, #00100000B

MOV A, R2

MOV DPTR, #7F01H

MOVX @DPTR, A ;输出位选码

.....

MOV A, R2

CLR C

RRC A

JC PASS

MOV R2, A

C51 版

#define wela_data XBYTE[0x7f01]

wela_data=0x20; wela_data=0x10; wela_data=0x01; (这可以在循环中通过移位实现)

（3）程序中段选码输出口是如何实现输出想要显示数字对应的段选码的？对应程序中的哪些语句实现了这一操作？试着解释下在这一过程中显示缓存单元（也就是汇编版中的内存 70H-77H 单元，C51 版中的数组 array[]）起的作用，引入显示缓存单元的好处。

借助段选码表，段选码表是按照数字从小到大存放 0 到 9 的段选码，因此通过以数字值作为索引，查询段选码表即可获得相应数字的段选码。

汇编版

MOV DPTR, #PTRN

```
MOVC    A, @A+DPTR ;查段选码表 PTRN,将显示缓存单元的数字代码转换为对应的
                        段选码
```

```
MOV     PO, A
```

```
;-----共阳极段码表-----
```

```
PTRN:DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H
```

C51 版

//共阳数码管 0~9 的数字段码表

```
code INT8U SEG_CODE[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
```

```
P0=SEG_CODE[array[i]]; //输出段选码
```

显示缓存单元的作用是，将待显示在多位 LED 显示器上的符号（数字）内容先存在显示缓存单元中。在显示子程序/函数中就依次取出显示缓存单元里的符号（数字）转换为段选码，送到段选码输出口实现显示。好处是增加了显示子程序/函数的通用性，方便实现显示内容的灵活改变。想改变显示内容时，就只需要改变显示缓存单元里的值，而不必修改显示子程序/函数。

二、思考如果要把例 03 改为 6 位段式 LED 显示，那将要改哪些地方？（硬件部分、软件部分）

硬件部分：将 8 位 LED 显示器更换为 6 位 LED 显示器；位选码口只需要 6 根引脚。

软件部分：显示缓存单元设置 6 个单元而不需要 8 个单元；位选码输出口程序段只要控制循环选中 6 位，而不是循环选中 8 位。如 `for (i=0; i<8; i++)` //扫描显示 8 位数码管改为 `for (i=0; i<6; i++)`

5.8 作业

一、矩阵式键盘线反转法作业

在例 06-1 基础上改造，实现 4*8 键盘（4 行 8 列）的检测。完成：

（1）模仿课件，写出 4*8 键盘线反转法的思路流程（可以程序流程图形式回答，具体到给各个口线的具体值）

具体略，回答思路：一个 4 位的口作行线口，一个 8 位的口作列线口。先行输出 0000，列输出 11111111；再读回行口和列口，如果列线口读回的仍是 11111111，说明没有按键，返回无键标志；如果列线口读回的是 01111111/10111111/11011111/11101111...../11111110 中的一种，则记录下对应列号（0~7）；接下来确定行号，行输出 1111，列输出 00000000；再读回行口和列口，如果行线口是 0111/1011/1101/1110 中的一种，则赋予对应的行首键号（0/8/16/24）（因为每行 8 个按键）；最后得出按键号：行首键号+列号。（注：步骤中改为先确定行号再确定列号也是可以。）

二、矩阵式键盘行扫描法作业

对例 06-2 和例 06-3（选其中一个例子），进行改造，实现 8*8 键盘的检测（提示：单片机需提供 8 位的扫描口和 8 位的回扫口）。完成：

（1）模仿课件，写出 8*8 键盘行扫描法的思路流程（可以程序流程图形式回答，具体到给各个口线的具体值）

具体略，回答思路：一个 8 位口接行线，一个 8 位口接列线。其中某个方向口固定接上拉电阻，这里假设列线接上拉电阻。那么行线作为扫描口（固定输出口），列线作为回扫口（固

定输入口)。

先开放整个键盘,即所有行口线都输出 0,读回扫口,如果读回的是全 1 则说明没有改变即没有按键(因为这个口是接着上拉电阻或者线值给 1),给无键标志,并退出按键检测程序;如果开放整个键盘,读回扫口读回的是有某一列变为 0 了,说明该列所在按键有按下,则开始逐行扫描;分别依次扫描口输出 01111111/10111111/...../11111110 这些扫描码进行逐行扫描。每输出一行扫描码,就读一次回扫口,看有没有某列变 0,若回扫口是 11111111,则说明不是这一行有键,就切换到下个扫描码继续扫描下一行,当扫描到某一行,回扫口值有一列变 0,则确定了按键所在行号,再结合按键所在列号(变 0 的那列),得出按键号(行首键号+列号)。

补充练习

一、学习理解群上的 ADC0809 例 14-1 和例 14-2

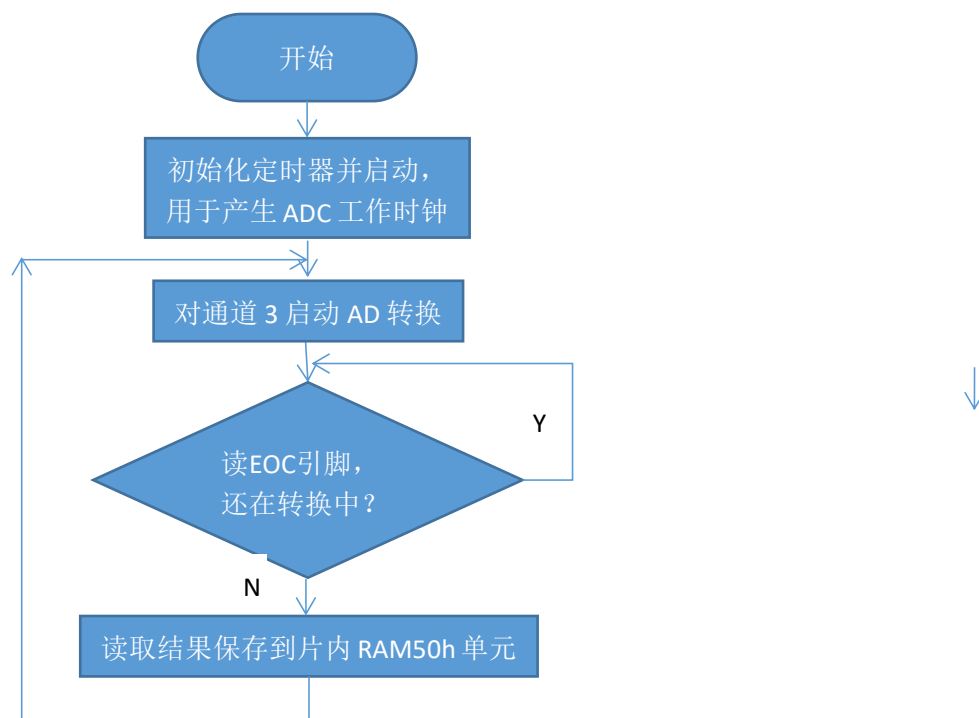
学习理解群上的 ADC0809 例 14-1 和例 14-2

1. 分析例 14-1 单片机是采用哪种方式管理 ADC?
2. 在例 14-1 程序中分别找出单片机启动转换的程序段、读取转换结果的程序段。
3. 分析例 14-2 c 程序是采用哪种方式管理 ADC?
4. 例 14-2 c 程序实现的功能是什么? 画出例 14-2 主程序流程图。
5. 在例 14-2 c 程序中分别找出单片机启动转换的程序段(执行总线写操作,即对应汇编版 MOVX 写)、读取转换结果的程序段(执行总线读操作,即对应汇编版 MOVX 读)。

答:

1. 例 14-1 单片机是采用 延时等待方式管理方式
2. 参考程序注释。
3. 例 14-2 c 程序是采用查询管理方式
4. 例 14-2 c 程序实现的功能是用查询方式管理 ADC,不断循环的对 IN3 通道输入的模拟信号进行 A/D 转换,转换结果(8 位二进制数)放在 51 单片机内存 50h 单元中。

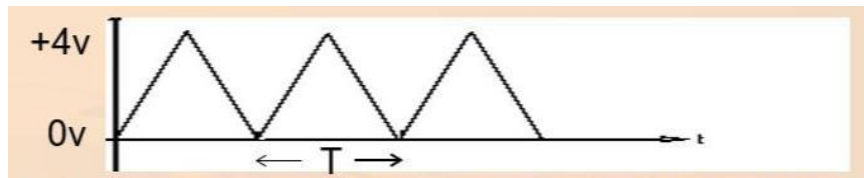
画出例 14-2 主程序流程图。



5. 在例 14-2 c 程序中分别找出单片机启动转换的程序段:ADCADD = 0x00; (执行总线写操作, 即对应汇编版 MOVX 写)、读取转换结果的程序段:DBYTE[0x50] = ADCADD; (执行总线读操作, 即对应汇编版 MOVX 读)。

二、线性标度变换作业。

1. 80C51 单片机想控制 8 位的数模转换器 DAC0832 输出波形, 阐述单片机应输出对应怎样变化的数字量, 才会生成如下图所示的三角波? 设定 DAC0832 的输出范围是 0~5V。



答: 0v 对应数字量是 00h, 4v 对应数字量是 $4 \times 255 / 5 = 204$ (十进制) = 0CCh (十六进制)
应先依次输出从 00h 递增到 0CCh 的数字量, 再输出从 0CCh 递减到 00h 的数字量, 这是一个周期的波形, 然后不断重复输出, 就会产生如图三角波。

2. 某电子秤采用 10 位 ADC, 测量范围 5kg~200kg, 假设被测重量 W 与数字量 D 之间的关系经处理已呈线性,

(1) 请列出该标度变换公式

(2) 试写出实现该标度变换(即实现上一小节的变换公式)的子程序 (汇编或 C51 均可, 建议 C51, 参考例 14-3 中的程序)

(3) 根据公式求当仪器在某一时刻采集得到的数字量为 1C8H (十六进制) 时对应的重量值是多少? 这里 $D = 1C8h$ (十六进制) = 456 (十进制)

(1) 答: $W = D \times 195 / 1023 + 5$ (kg)

(2) 答: $W = D \times 195 / 1023 + 5$; (为保留多位小数字后 2 或 3 位数, 可考虑另乘以 100 或 1000)

(3) 答: 对应重量值: $W = 456 \times 195 / 1023 + 5 = 91.92 \text{ kg}$

