上海大学　计算机学院

《计算机组成原理实验》报告十

姓名 赵文宇　　**学号** 23121769

时间 周三 9-11　　**机位** 8　　**指导教师** 刘跃军

实验名称: 中断机制和应用

一、实验目的

1、学习实验箱感知中断的硬件结构和工作原理
2、学习使用中断系统。
3、学习使用扩展外设

二、实验原理

1、程序中断:因“随机性”原因，使一个程序暂停执行，转而执行另一个程序，以处理随机事件，然后再返回原程序继续执行的过程成为“中断”中断同子程序调用有共同点:执行另一个程序，然后返回。所以在调用另一个程序(中断服务子程序)时必须保存断点。

中断与子程序调用有一个根本区别:中断发生的时间是随机的(不可预知,但发生后应该如何处理是安排好的)，而子程序调用时间是安排好的，由程序员写下的调用指令决定。中断发生的“随机性”决定了“必须用硬件感知中断请求”、“不仅要保存断点，还必须保存现场”。中断发生时间与正在运行的程序的无关性，使得整个系统在运行一个程序的同时，还能感知其它事件的发生!这是实时监控的技术基础、是多用户、多任务、多线程技术的关键点，因此是操作系统工作的前提，是计算机系统的“点睛”之笔!深刻理解中断系统是计算机专业人员用好计算机的必备知识!

2、实验箱的中断感知硬件:

只有“中断返回”指令和复位操作使 EINT 为低电平,这个低电平作用到 IREQ 的 SD 端，使上面这个 D 触发器的 Q 端为 1，作用到 IACK 的 CD 端使下面这个 D 触发器的 Q 端输出 0。本课程称其为状态 0。

系统复位结束或执行其他指令时,EINT 为无效的高电平，这时在时钟 CK 驱动下，IREQ 的 Q 端输出 D 端的 INT 状态。当有中断请求时 INT 为 0，则一个 CK 后 Q 端输出 0，但这个 0 能否被 CPU 感知却要看①号“或门”是否允许它通过。而“非取指”微指令有 IREN=1，则②号“或门输出 1，于是 IREQ 的 Q 端无论输出 0 或 1,①号“或门”总输出 1，即不允许中断请求通过。同时这个 1 又送入 IACK 的 SD 端;于是下触发器的 SD 和 CD 端的输入都是无效状态，这个触发器保持稳定。

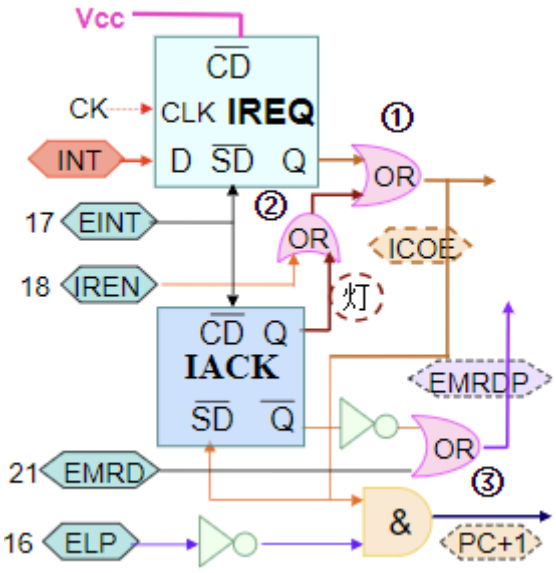

图 1. 实验箱硬件的逻辑结构

当执行取指微指令时，IREN-0，于是②号或门输出 0，这时号“或门”对 IREQ 的 Q 端开放，若有中断请求就会在这时被 CPU 感知。所以无论中断请求在何时提出，都只能在取指阶段被感知!

当①号“或门”输出 0 时中断被感知，同时这个低电平使 IACK 的 SD 有效，迫使其 Q 端输出 1，ACK 灯亮，并使号“或门”对 IREN 关闭并输出 1:子程序的这个 1 又使①号“或门”对中断请求关闭并输出 1，这个 1 又返回 IACK 的 SD 端，使 IACK 保持 Q-1 的状态。所以系统进入中断服务子程序后，ACK 灯保持亮，且不响应新的中断请求(仅一级中断)

直到中断服务程序执行“中断返回”指令 RETI 时，EINT 为 0，使系统再次进入状态 0:

在中断服务期间中断源若不及时撤销中断请求(使 INT 为 1)，则中断返回后系统会再次进入中断状态，造成“中断未返回”的假象。

ICOE 向下经“与门”控制 PC+1 信号，如图。当感知中断请求(ICOE-0)时，使 PC 值不变(确定断点)，进入服务程序后(ICOE 三 1)，PC 恢复自动加 1 功能，保证服务程序的顺序执行。

ICOE 有效时，使 EM 的锁存器输出中断指令 B8H 到 PC，uPC 再把 uM 中的中断微指令(INT)的控制信号输出。

INT 指令在第一个周期保存 PC 值(断点)到 ST 堆寄存器。在第二个周期把 IA 寄存器的内容送 PC，可见:IA 中应事先存入中断服务子程序的入口地址。所以 IA 称为中断向量寄存器。IA 的值由红色的码盘输入:

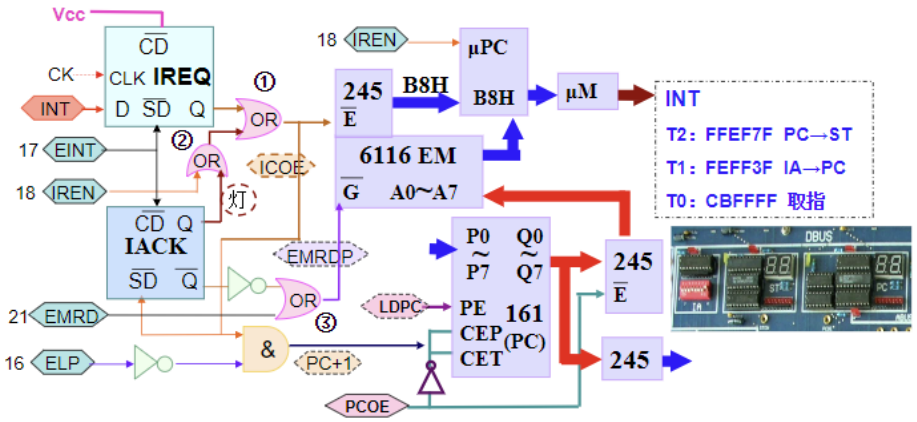

图 2. 实验箱硬件的示意图

3、实验箱外扩系统

如下图。

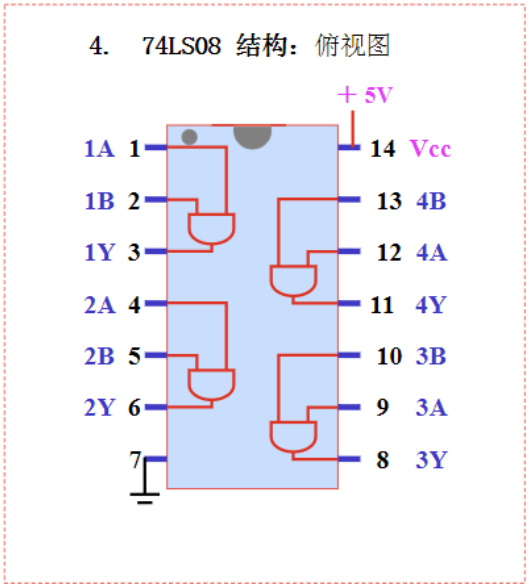

图 3. 实验箱外扩系统示意图与俯视图

三、实验内容

1．实验任务一（用 74LS 08 芯片搭建当电键 K1 和 K2 都为 1 时不产生中断请求信号的外部电路。）

(1) 实验步骤

先对任务进行分析：

因为 74LS08 芯片是很多个与门组合而成的扩展单元，如果 1、2 号插孔的输入电平都为高电平 1，那么 3 号插孔的输出就为 1。反之，如果有一个插孔输入为低电平 0，那么输出就为低电平 0。

题目要求 K1 与 K2 都为 1 时不产生中断请求信号的外部电路（**我在实验的时候替换成了 K14、K15 开关**）。就可以将 K14 K15 与该芯片的 1、2 号插孔相连，3

号插孔与 INT 相连，用于判断是否产生中断请求信号。7 号插孔与 GND 相连，接地，40 插孔与+5V 相连，连接电源。

通过控制 K14 或 K15 开关的高低电平状态（只需要将某一个从高电平->低电平，低电平->高电平），给出一个低电平的下降沿，就可以实现中断请求信号的产生。

至此，即可完成该操作。

下面是具体的连线及操作过程：

将扩展单元上的 1、2、3 号插孔用导线与 K15、K14、INT 相连，将 7 号插孔与 GND 相连，用于接地，将 40 号插孔与+5V 相连。

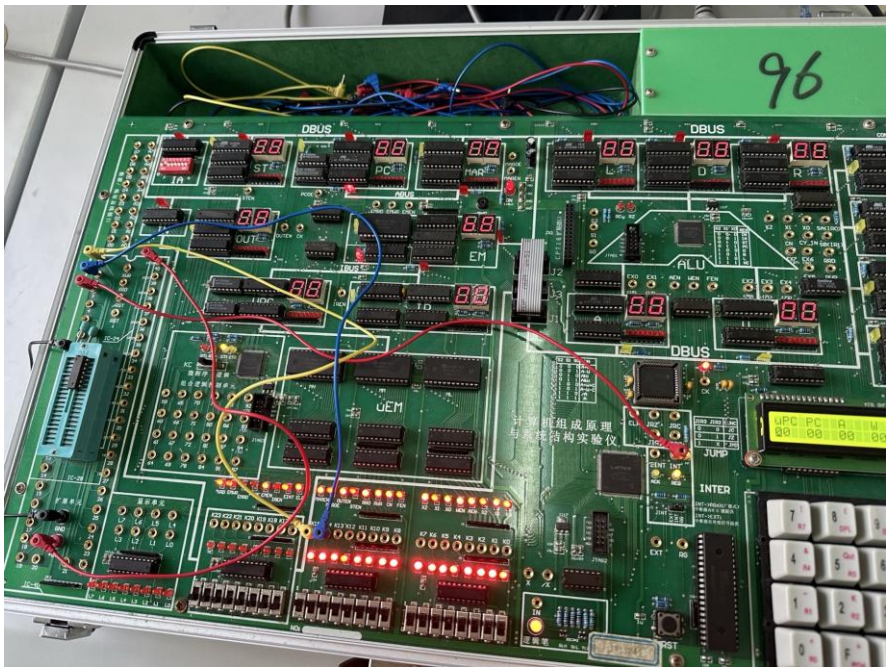

图 4.实验任务一连线

操作如下：

当开关 k14、k15 均高电平为 1 时无任何反应，INT 指示灯亮，表明中端被遏制，不产生中断请求信号，当 K15 开关置为低电平 0 时，INT 指示灯不亮，产生了中断请求信号。

(2)实验现象

在各实验分任务操作过程中已经进行记录，在此不过多赘述。

本实验无数据分析与处理，数据记录同实验现象。

(3)实验结论

可以认为，本次实验正确地完成了用 74LS 08 芯片搭建当电键 K1 和 K2 都为 1 时不产生中断请求信号的外部电路。

2．实验任务二（编制中断服务子程序使 OUT 交替显示 AA、BB 三次后返回源程序。源程序为上次实验完成的交替显示 11 和 55 的程序。

(1)运行上述程序，在完成 AA、BB 交替显示三次之前恢复 K1K2 都为 1 的状态。

记录 OUT 显示的现象、REO 灯和 ACK 灯的情况以及 ST 寄存器的值及改变情况。
(2)运行上述程序，在完成 AA、BB 交替显示时不恢复 K1K2 都为 1 的状态记录 OUT 显示的现象、REO 灯和 ACK 灯的情况以及 ST 寄存器的值及改变情况。
分析上述二种显示现象的原因。）

内容要求如下：R0:中断时 AA、BB 显示的次数 R1:跟踪主程序 OUT 显示的值 R2:跟踪主程序延时的值

先对任务进行分析，可以把任务分解为：

(1)将 A 寄存器中数据改为 11H，并送到 OUT 寄存器。(2)延时。(3)将 A 寄存器中数据改为 55H，并送到 OUT 寄存器。(交替显示 11 与 55)

(4)将 R0 寄存器中数字改为 03H，用于记录中断循环次数。(5) 将 A 寄存器中数据改为 AAH，并送到 OUT 寄存器。(6)延时。(7)将 A 寄存器中数据改为 BBH，并送到 OUT 寄存器。(8)R0 自减一，检查是否为 0，如果为 0，则回到步骤(1)，否则回到步骤(5)。

根据上述分析，本任务汇编语言代码如下：

调试窗口 PAIDAIXNG22.asm | EPRom

```
ORG 00H
LOOP: MOV A,#11H
      MOV R1,A
      OUT
      MOV A,#10H
x1:   SUB A,#01H
      MOV R2,A
      JZ x2
      JMP x1
x2:   MOV A,#55H
      MOV R1,A
      OUT
      MOV A,#10H
x3:   SUB A,#01H
      MOV R2,A
      JZ x4
      JMP x3
x4:   JMP LOOP

ORG 40H
      MOV R0,#03H
LOOP1: MOV A,#AAH
       OUT
       MOV A,#10H
y1:    SUB A,#01H
       JZ y2
       JMP y1
y2:    MOV A,#BBH
       OUT
       MOV A,#10H
y3:    SUB A,#01H
       JZ y4
       JMP y3
y4:    MOV A,R0
       SUB A,#01H
       JZ EXIT
       MOV R0,A
       JMP LOOP1
EXIT:  MOV A,R1
       OUT
       MOV A,R2
       SUB A,#00H
       RETI
END
```

图 4.实验任务二汇编语言代码

其中，我指定了中断程序从内存地址 40H 开始，同时在实验箱左上 IA 区域改为 01000000，代表 40H 地址。

用 CP226 文件打开含有上述代码的.asm 文件，确定正确打开后，点击快捷操作中的连接实验箱，确保与实验箱相连。点击编译写入，发现实验箱闪烁。按下分步操作的快捷操作，观察实验箱上 A 寄存器、OUT 寄存器、R0、R1、R2 寄存器

中的数值变化。

(2)实验现象

一开始编译下载并运行程序的时候,OUT 寄存器中 11 和 55 交替变化，寄存器 R1 中与 OUT 同步显示 11 和 55。R2 寄存器中为每次交替出现的延迟时间，初始值为 10，每次自减一，当 R2 中的数值减到 0 时，11 和 55 的进行交替。ACK 和 REQ 指示灯均不亮，**代表未发生中断请求**。

当将开关 K15 置为 0 时，**产生了一个中断信号**，ACK 和 REQ 指示灯亮，OUT 寄存器中 AA 和 BB 交替出现。R0 寄存器中的值一开始为 03H，每当进行一次交替周期，R0 中的数值减一，当减到 00H 时，重新复位为 03H。R1 中的数值显示 55H(11H)，代表发生中断前程序执行到 OUT 显示 55H(55H)，R2 上显示的数字代表发生中断前的延迟，R1,R2 记录着发生中断前程序的执行情况。

在以上步骤都执行完后，将开关 K15 置为 1 时，代表**中断请求结束**，程序从原本中断的地方继续开始执行。ACK 指示灯灭表示中断请求程序执行完毕，OUT 寄存　器中初始显示 55H，**随着时间交替显示 11 和 55**，PC 寄存器初始为 16H（ST 寄存器中的数值），R0 寄存器显示 01H，R1 寄存器也会显示中断操作前 OUT 寄存器中的数值，交替显示 11 和 55，OUT 寄存器数值与 R1 寄存器中数值一样，R2 寄存器恢复至上一次中断操作前的数值，随着程序执行不断减 1，当减到 0 时重新打入 10H，OUT 寄存器中 11 和 55 进行一次交替，R1 的值与 OUT 中的值相同。（**但是我测试了一下如果 K15 不拨回 1，仍然保持低电平 0，循环三次 AA BB 后，仍然会恢复到交替显示 11、55 的状态，还不太清楚是为什么**）

(4) 实验结论

正确的编写了程序，实现了“**编制中断服务子程序使 OUT 交替显示 AA、BB 三次后返回源程序**”的任务要求。

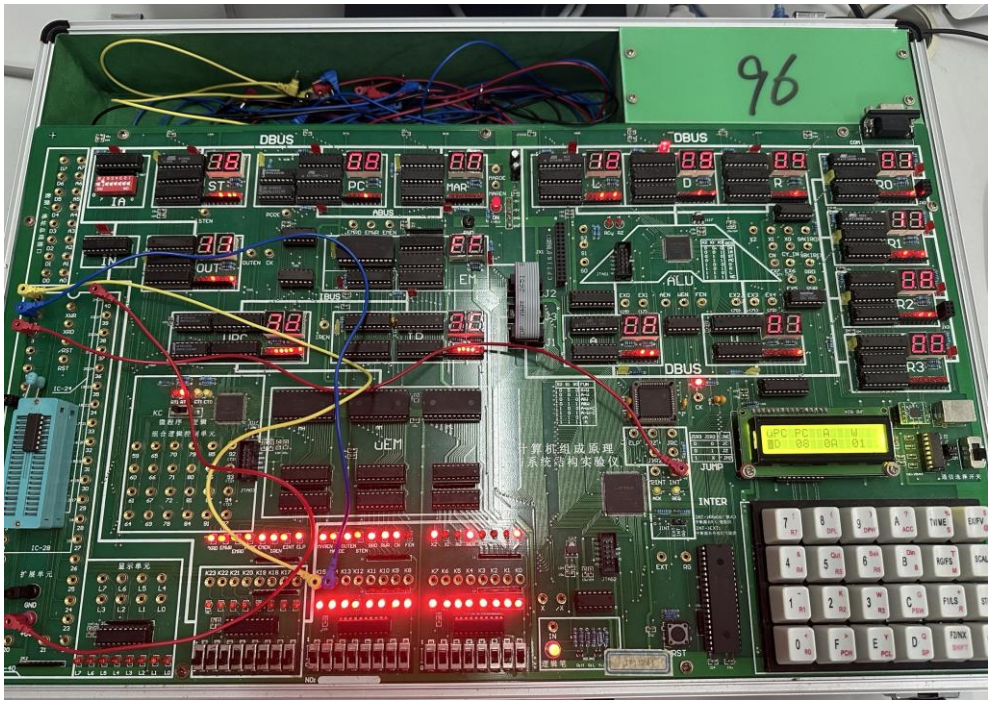

图 5.实验任务二显示 11

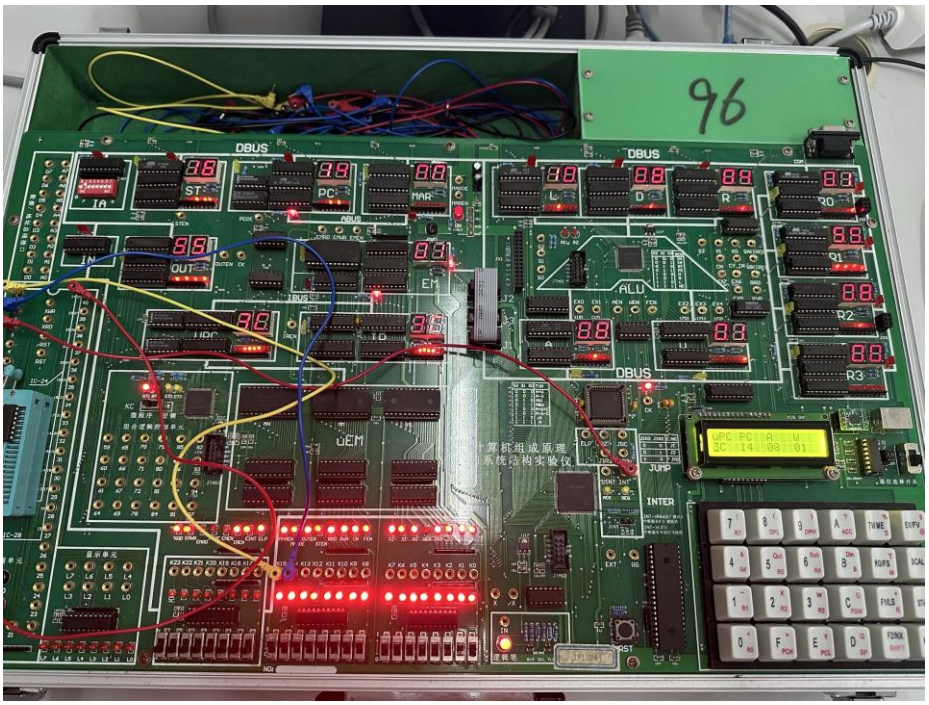

图 6.实验任务二显示 55

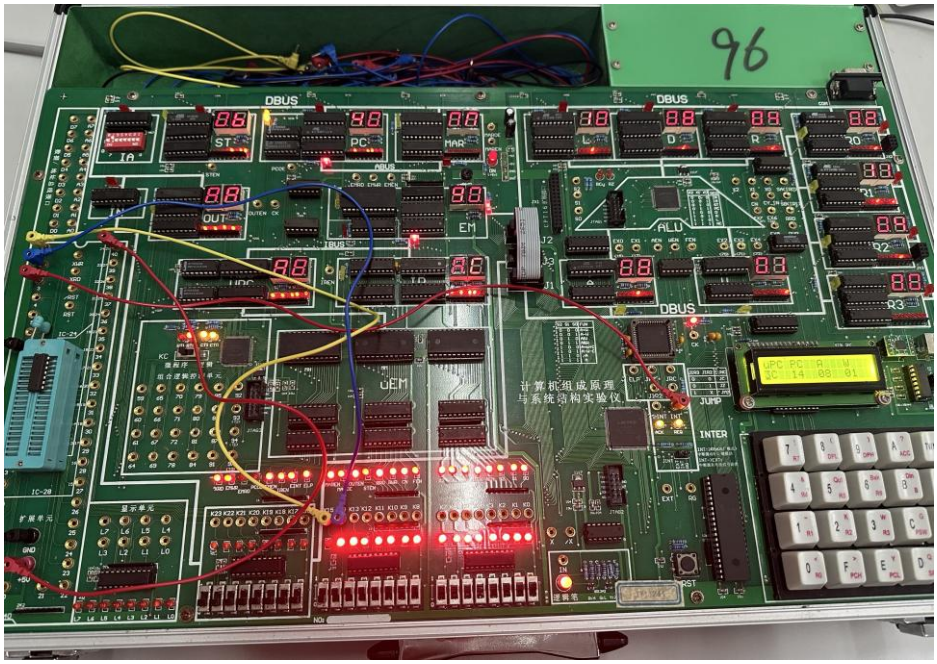

图 7.实验任务二显示 AA

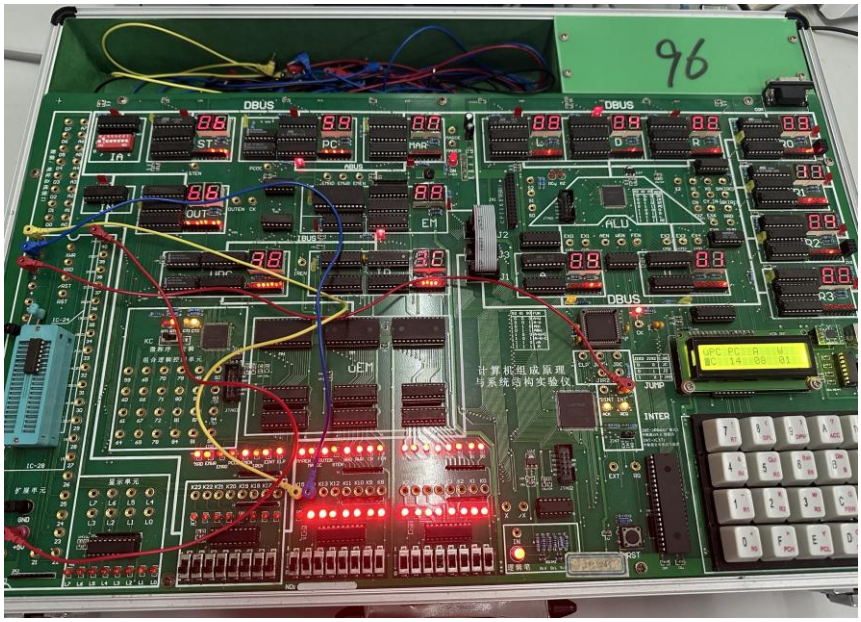

图 8.实验任务二显示 BB

四、建议和体会

1、实验开始前，要先检查一下导线是否功能正常。
2、通过自己的尝试与摸索，自己学会了如何进行 CP226 软件上编写汇编语言与实验箱操作的结合。
3、依然还是老样子，提前预习的重要性，即在进行实验课前，需要对该节课实验的内容对应的理论知识（虽然好像与机组理论课关系不大）进行温故与复习并且对实验步骤进行大框架上的构想与思考，最好能够提前学会要用到的知识点。
（如果不复习、提前构想的话，仅靠课堂上的那两个小时可能不足以完成所有任务，或者说进展的比较慢，没有什么头绪）。

五、思考题

1、实验箱的中断服务程序中可以嵌套一般的子程序吗?

答：不行，因为 CP226 实验中，堆栈寄存器中只能存放 8 位二进制，也就是 1 位，但是一般的子程序嵌套需要两位，所以不可以嵌套一般的子程序。