

T1 (中断法)

```
ORG 0000H
AJMP MAIN
ORG 000BH
AJMP TIME
```

MAIN:

TIME

```
MOV TMOD, #02H
MOV TH0, #00H
MOV TL0, #06H
SETB EA
SETB ET0
SETB TR0
HERE: SJMP HERE
```

CLP P1.0
RETI.

T2 (查询法) (中断法比较烦)

MAIN:

Loop1:

```
MOV TMOD, #50H ; 工作方式1, 计数
MOV TH1, #FCH ;
MOV TL1, #18H ; 初值
MOV IE, #00H ; 不允中断
CLR TFI
SETB TR1 ; 开始计数
```

Loop2:

```
JBC TFI, Loop3
SJMP Loop2 ; 计数完成后进入计时
```

Loop3:

```
MOV TMOD, #10H ; 定时
MOV TH1, #ECH
MOV TL1, #78H
CLR TFI
SETB TR1
```

Loop4:

```
JBC TFI, Loop
SJMP Loop4 ; 定时到了回到计数
```

T3. (也可用查表法)

```
ORG 0000H
AJMP MAIN
ORG 0023H
AJMP SENT
```

MAIN:

```
MOV R0, #50H ; 初始地址
MOV R1, #40H ; 计数
MOV SCN, #C8H ; 初始值, 寄存器
MOV TH1, #E8H ; 设定波特率
```

Loop:

```
CJNE R1, #00, Loop ; 循环条件
```

SENT:

```
CLR TI ; 初始置0
DEC R1 ; 计数+1
INC R0 ; 取下一地址
MOV A, @R0 ; 取值
MOV SBUF, A ; 开启串口
RETI
```

T5 (太麻烦), 不写

T6

```
ORG 0000H
AJMP MAIN
```

MAIN:

```
MOV R0, #30H
MOV R1, #08H
MOV DPTR, #7FFFH
```

Loop: JNB P1.0, Loop

Loop1: MOVX A, @DPTR

INC R0

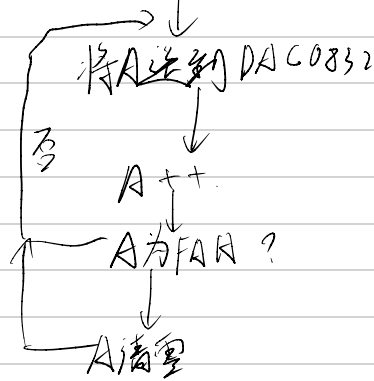
INC DPTR

DJNZ R1, Loop1

SJMP MAIN

T4

(1) 初始化 A, DPTR



(2)

MAIN:

```
MOV A, #00H ; 初始化
```

```
MOV DPTR, #80H
```

```
MOV DPL, #00H ; 地址
```

Loop: MOVX @DPTR, A ; 发送

```
INC A
```

```
LCALL DELAY ; 延时
```

```
CJNE A, #FAH, Loop ; 清零
```

```
MOV A, #00H
```

```
SJMP Loop
```

T7 不会, 没学过

T8 用汇编有点烦

T9

(1) 接收 A/D 输入的 8 字节数据

(2) 7FFFH

(3) 控制/开启数据传输

(4) A0H~A7H 内部 RAM