

这是非常棒的复习思路！将代码抽象为“通用模板”是掌握单片机编程的最快方法，考试时只需要像“填空题”一样把参数填进去即可。

根据你提供的所有代码文件，我为你整理了一份《AT89C51 期末核心编程通用模板》。你可以直接背诵这些框架。

1. 数码管显示模板 (最核心)

核心口诀：

位选 → 段选 → 延时 → 消影。

A. 动态扫描 (同时显示多个数字)

```
// 放在 while(1) 循环中
for(i = 0; i < 8; i++) // 扫描 8 位
{
    P2 = weixuan[i]; // 1. 位选：选中第 i 个位置
    P0 = duanma[数]; // 2. 段选：填入要显示的数字/变量

    delay(1, 40); // 3. 延时：动态扫描必须短（约1-2ms）
                  // 如果太长会闪烁，太短会亮度不够

    P0 = 0x00; // 4. 消影：必须清零，防止重影
}
```

B. 静态/慢速显示 (跑马灯式显示)

```
// 放在 while(1) 或 for 循环中
P2 = weixuan[i]; // 位选
P0 = duanma[i]; // 段选
delay(1000, 40); // 长延时：人眼能看清变化
P0 = 0x00; // 消影
```

2. 独立按键模板 (查询法)

核心口诀：

检测 → 消抖 → 确认 → 执行 → 等待松手。

```
if(P1 != 0xff) // 1. 粗测：是否有键按下？
{
    delay(20, 40); // 2. 消抖：延时约 10-20ms

    if(P1 != 0xff) // 3. 复测：确认真的按下了
    {
        if(k0 == 0) // 4. 判断具体是哪个键
        {
            // --- 在这里写按键要执行的代码（例如 a++） ---
        }
    }
}
```

```
        while(!k0); // 5. 等待松手（死循环，直到按键弹起）
    }

    if(k1 == 0) { ...; while(!k1); } // 其他按键同理
}
}
```

3. 外部中断模板 (INT0 / INT1)

核心口诀： 主函数开开关，中断函数做任务，进出关中断防止重入。

A. 主函数初始化 (Main Setup)

```
void main()
{
    EA = 1;    // 总中断开关：必须开

    // --- INT0 配置 ---
    IT0 = 0;    // 触发方式：0=低电平触发(按住一直触发)，1=下降沿触发(按一下触发一次)
    EX0 = 1;    // INT0 允许开关

    // --- INT1 配置（如果需要） ---
    IT1 = 0;
    EX1 = 1;
    PX1 = 1;    // （可选）优先级：1为高优先级，可打断低优先级中断

    while(1)
    {
        // 主程序逻辑
    }
}
```

B. 中断服务函数 (ISR)

```
// 中断号：INT0 -> 0, INT1 -> 2
void ex0_int() interrupt 0
{
    EX0 = 0;    // 1. 暂时关闭中断（防止按键抖动重复触发）

    // --- 在这里写中断要执行的代码 ---
    // 常用操作 1：翻转变量（暂停/开始） -> TR0 = ~TR0;
    // 常用操作 2：清零变量（复位） -> sum = 0;
    // 常用操作 3：阻塞等待（按住显示） -> while(!k0);

    EX0 = 1;    // 2. 恢复中断
}
```

4. 定时器模板 (Timer0 / Timer1)

这是期末大题最常用的模板，必须背熟初值计算公式。

A. 定时器初始化 (以 50ms 为例)

公式：

$$X = 65536 - (\text{时间}\mu s)。$$

```
void main()
{
    // 1. 模式设置 (TMOD)
    // 0x01: T0模式1; 0x10: T1模式1; 0x11: T0和T1都模式1
    TMOD = 0x01;

    // 2. 装载初值 (例如 50ms = 50000us)
    TH0 = (65536 - 50000) / 256; // 取高 8 位
    TL0 = (65536 - 50000) % 256; // 取低 8 位

    // 3. 开启中断和定时器
    EA = 1; // 总中断
    ET0 = 1; // T0 中断允许
    TR0 = 1; // T0 开始计时 (启动键)

    while(1) { ... }
}
```

B. 定时器中断函数

C

```
// 中断号: Timer0 -> 1, Timer1 -> 3
void et0_int() interrupt 1
{
    // 1. 重新装载初值 (手动重装)
    TH0 = (65536 - 50000) / 256;
    TL0 = (65536 - 50000) % 256;

    // 2. 计数器累加
    count++;

    // 3. 时间判断 (例如 20次 * 50ms = 1秒)
    if(count == 20)
    {
        count = 0; // 清零计数器
        sec++; // 秒加一

        // 4. 进位处理 (如果需要)
        if(sec == 60)
        {
            sec = 0;
            min++;
        }
    }
}
```

```
}
```

5. 蜂鸣器模板

核心口诀： 只有翻转电平才能响，频率靠延时，时长靠循环。

A. 持续响 / 报警

C

```
while(1)
{
    fmq = ~fmq;          // 产生方波
    delay(1, 40);        // 频率控制（延时越小，音调越高）
}
```

B. 节奏响 (滴——滴——)

C

```
// 响阶段
for(i=0; i<500; i++)    // 循环次数决定响多久
{
    fmq = ~fmq;
    delay(1, 40);        // 音调
}

// 停阶段
fmq = 1;                 // 关闭蜂鸣器（根据电路可能是 0 或 1）
delay(500, 40);         // 静音多久
```

6. LCD1602 显示模板

核心口诀： 初始化 \to 定位置 \to 写数据。

C

```
// 包含头文件
#include "lcd.h"

void main()
{
    lcd_ins(); // 1. 初始化 LCD

    while(1)
    {
        // 2. 写指令：设定光标位置
        // 第一行开头：0x80；第二行开头：0x80 + 0x40
        write_com(0x80 + 0x00);

        // 3. 写数据：循环写入字符数组
        for(i=0; i<16; i++)
```

```
{
    write_dat(str[i]);
    // delay(100, 40); // 可选：打字机效果
}

// --- 可选操作：整屏移动 ---
// write_com(0x18); // 左移指令
}
}
```

考试防坑 Checklist (填空前必看)

1. 硬件连接反转：

- 看清题目是 **P0 接段选** 还是 **P2 接段选**？（通常是 P0 段选 P2 位选，但有时会反过来）。

2. 触发电平：

- 蜂鸣器/LED 是 **低电平(0) 触发** 还是 **高电平(1) 触发**？
- 按键是按下变 0 还是变 1？（通常是按下变 0）。

3. 变量范围：

- 计数变量（如 `count`, `i`）不要忘了定义。
- 如果数值超过 255，必须用 `unsigned int`，不能用 `unsigned char`。

4. 头文件：

- 用到 `_cror_` 移位一定要加 `#include <intrins.h>`。
- 用到 LCD 一定要加 `#include "lcd.h"`。