

第一百一十九节： 一个完整的人机界面的程序框架的脉络。

【119.1 一个完整的人机界面的程序框架的脉络。】

前面两节例子告诉我们，一个完整的人机界面的程序框架包含两个要素，分别是“支点”与“更新”。“支点”包括“窗口选择”和“局部选择”，“更新”包括“整屏更新”和“局部更新”。

“支点”的作用是把显示函数与按键函数完美无缝的关联起来，两个函数同样的“支点”促使同样的“话语体系”，让“所见即所得”实时同步，确保按键操作的数据就是当前显示被选中的数据。

“静态数据”与“动态数据”的概念。被窗口显示的数据通常有两种：一种是静态数据，比如装饰门面的数据，只能显示不能更改的数据，以及图片图标这类数据；另外一种动态数据，这种数据在窗口显示上是活动的可编辑的，是需要经常修改的，往往也是系统核心的数据，需要保存或者需要跟某些关键运动密切相关的数据。比如，在前面章节中，数码管要显示三个窗口“1-XX”，“2-YY”，“3-ZZ”，其中“1-”、“2-”、“3-”是属于静态数据，它们是起“装饰”作用的。而“XX”、“YY”、“ZZ”则是动态数据，它们是可编辑的，也是单片机系统内部核心的数据。

“整屏更新”与“局部更新”的分工。“整屏更新”主要负责在切换新窗口时，把“静态数据”一次性显示到当前窗口。而“局部更新”主要负责在当前窗口下显示“动态数据”。

下面，我把一个完整的人机界面的程序框架的脉络勾勒出来，让大家有一个整体的观感，这种人机界面的程序框架放之四海而皆准，我已把它应用在各种数码管，单色液晶屏，彩屏，电脑上位机等项目上。假设某个项目中只有两个“窗口”只有两个“局部”，程序框架的脉络如下：

显示部分：

```
void DisplayTask(void) //数码管显示的上层任务函数
{
    switch(Gu8Wd) //以“窗口选择” Gu8Wd 为支点
    {
        case 1:
            Wd1(); //窗口 1 显示函数
            break;
        case 2:
            Wd2(); //窗口 2 显示函数
            break;
    }
}

void Wd1(void) //窗口 1 显示函数
{
    if(1==Gu8WdUpdate) //整屏更新
    {
        Gu8WdUpdate=0; //及时清零，只更新一次显示即可，避免一直进来更新显示

        ..... //此处省略 N 行代码，用来显示静态的数据，比如图片图标，或者装饰的数据
    }
}
```

```

        //以下，“整屏更新”必然是要把所有的“局部更新”都触发一次
        Gu8PartUpdate_1=1; //局部 1 更新显示
        Gu8PartUpdate_2=1 ;//局部 2 更新显示
    }

    if(1==Gu8PartUpdate_1) //局部 1 更新显示
    {
        Gu8PartUpdate_1=0; //及时清零，只更新一次显示即可，避免一直进来更新显示

        ..... //此处省略 N 行代码，用来显示动态的数据。比如可编辑的数据，实时变化的数据
    }

    if(1==Gu8PartUpdate_2) //局部 2 更新显示
    {
        Gu8PartUpdate_2=0; //及时清零，只更新一次显示即可，避免一直进来更新显示

        ..... //此处省略 N 行代码，用来显示动态的数据。比如可编辑的数据，实时变化的数据
    }

    if(0==vGu16BlinkTimerCnt) //跳动的光标，或者动态闪烁的某位被选中的数据
    {
        vGu8BlinkTimerFlag=0;
        vGu16BlinkTimerCnt=BLINK_TIME; //重设定时器的定时时间
        vGu8BlinkTimerFlag=1;

        ..... //此处省略 N 行代码，用来制作跳动的光标或者某位被选中而闪烁的数据
    }
}

void Wd2(void) //窗口 2 显示函数
{
    ..... //此处省略 N 行代码，窗口 2 显示函数的代码跟窗口 1 类似
}

```

按键部分：

```

void KeyTask(void) //按键的任务函数

```

```

{
    if (0==vGu8KeySec)
    {
        return;
    }

    switch(vGu8KeySec)
    {
        case 1:    //1 号按键
            switch(Gu8Wd) //以“窗口选择” Gu8Wd 为支点
            {
                case 1:    //在窗口 1 下
                    switch(Gu8Part) //以“局部选择” Gu8Part 为支点
                    {
                        case 1:

                            .....    //此处省略 N 行代码

                            break;

                        case 2: //局部 2 被选中

                            .....    //此处省略 N 行代码

                            break;

                    }
                    break;

                case 2:    //在窗口 2 下
                    switch(Gu8Part) //以“局部选择” Gu8Part 为支点
                    {
                        case 1:

                            .....    //此处省略 N 行代码

                            break;

                        case 2: //局部 2 被选中

                            .....    //此处省略 N 行代码

                            break;

                    }
                    break;
            }
    }
}

```

```
    }

    vGu8BeepTimerFlag=0;
    vGu16BeepTimerCnt=VOICE_TIME;  //蜂鸣器发出“滴”一声
    vGu8BeepTimerFlag=1;

    vGu8KeySec=0;
    break;

case 2:    //2 号按键

    .....    //此处省略 N 行代码，跟 1 号按键的代码类似

    vGu8BeepTimerFlag=0;
    vGu16BeepTimerCnt=VOICE_TIME;  //蜂鸣器发出“滴”一声
    vGu8BeepTimerFlag=1;

    vGu8KeySec=0;
    break;

}
}
```