单片机开发中，有时会用到屏幕来显示内容，当需要逐级显示内容时，就需要使用多级菜单的形式了。

STM32多级菜单课题研究与探讨

STM32作为一款广泛应用的微控制器，在各种应用场景中都有其身影。在某些应用中，我们可能需要使用多级菜单来逐级显示内容。本篇将对如何在STM32上实现多级菜单进行深入的课题研究与探讨。

一、多级菜单的必要性

在许多应用中，我们需要为用户提供一个友好的界面，让他们能够方便地选择和操作。多级菜单就是一种常见的界面形式，它允许用户通过一系列的菜单项，逐级深入，最终达到他们想要的操作。这种设计不仅提高了界面的友好性，也使得操作更加直观和易于理解。

二、STM32与多级菜单的结合

STM32具有强大的处理能力和丰富的外设资源，这使得在其上实现多级菜单成为可能。通过合理的软件设计，我们可以利用STM32的资源，为用户提供一个高效、稳定的多级菜单系统。

三、实现方法

在STM32上实现多级菜单，主要有以下步骤：

1. 硬件准备：首先，我们需要为STM32配置适当的显示设备，如LCD、LED等。此外，还需要准备相应的按键输入设备，以便用户能够与多级菜单进行交互。
2. 菜单结构定义：我们需要定义多级菜单的结构。这包括各级菜单的显示内容、功能以及它们之间的关系。
3. 程序编写：根据定义的菜单结构，我们需要编写相应的程序。这包括初始化程序、显示程序、按键处理程序等。
4. 调试与优化：在程序编写完成后，我们需要进行详细的调试，确保多级菜单能够正常工作。并根据实际使用情况，对程序进行优化，以提高其性能和稳定性。

四、结论

通过上述的研究与探讨，我们可以得出结论：在STM32上实现多级菜单是完全可行的。这不仅可以提高界面的友好性，还可以使操作更加直观和易于理解。在实际应用中，我们应该根据具体的需求和场景，合理设计多级菜单的结构和功能，使其能够更好地为用户服务。同时，我们也应该注意程序的性能和稳定性，确保多级菜单能够在实际使用中表现出色。

1 多级菜单

多级菜单的实现，大体分为两种设计思路：

通过双向链表实现

通过数组查表实现

总体思路都是把菜单的各个界面联系起来，可以从上级菜单跳到下级菜单，也可从下级菜单返回上级菜单。

数组查表的方式比较简单，易于理解，本篇就来使用数组查表发在STM32上实现多级菜单的显示。

2 代码实现

2.1 数组查表

首先需要定义一个结构体：

typedef struct

{

uchar current;

uchar up;//向上翻索引号

uchar down;//向下翻索引号

uchar enter;//确认索引号

void (\*current\_operation)();

} key\_table;

current：当前页面的索引号

up：按下“向上翻“按钮后要跳转到的页面索引号

down：按下“向下翻“按钮后要跳转到的页面索引号

enter：按下“确认“按钮后要跳转到的页面索引号

current\_operation：当前页面的索引号要执行的显示函数，这是一个函数指针

注意：对于菜单显示的操作，用到了3个按键，分别是向下、向下和确认，如果单片机上的IO资源较为紧张，还可以把“向上翻”按钮省去，只通过“向下翻”按钮来实现循环访问，对应的结构体也可以去掉该成员。

然后定义一个表，用来定义各个页面间如何跳转

key\_table table[30]=

{

//第0层

{0,0,0,1,(\*fun\_0)},

//第1层

{1,4,2, 5,(\*fun\_a1)},

{2,1,3, 9,(\*fun\_b1)},

{3,2,4,13,(\*fun\_c1)},

{4,3,1, 0,(\*fun\_d1)},

//第2层

{5,8,6,17,(\*fun\_a21)},

{6,5,7,18,(\*fun\_a22)},

{7,6,8,19,(\*fun\_a23)},

{8,7,5, 1,(\*fun\_a24)},

{ 9,12,10,20,(\*fun\_b21)},

{10, 9,11,21,(\*fun\_b22)},

{11,10,12,22,(\*fun\_b23)},

{12,11, 9, 2,(\*fun\_b24)},

{13,16,14,23,(\*fun\_c21)},

{14,13,15,24,(\*fun\_c22)},

{15,14,16,25,(\*fun\_c23)},

{16,15,13, 3,(\*fun\_c24)},

//第3层

{17,17,17,5,(\*fun\_a31)},

{18,18,18,6,(\*fun\_a32)},

{19,19,19,7,(\*fun\_a33)},

{20,20,20, 9,(\*fun\_b31)},

{21,21,21,10,(\*fun\_b32)},

{22,22,22,11,(\*fun\_b33)},

{23,23,23,13,(\*fun\_c31)},

{24,24,24,14,(\*fun\_c32)},

{25,25,25,15,(\*fun\_c33)},

};

这里解释一下该表是如何工作的：

此表，表示了4级菜单的显示关系（注意第0层其实只是一个欢迎界面）

第一层菜单，只有4个选项，因此这里只列了4行（注意最后一个选项用作返回上一级，无实际内容含义）

第二层菜单，就是对第一层菜单中的3个实际的选项进行进一步的介绍，每种介绍又有4个子项（注意最后一个选项也是用作返回上一级，无实际内容含义），因此，这里的第二层菜单列了3x4=12行

第三层菜单，又是对第二层菜单中的子项进行进一步的介绍（3个分类，每类有3个子项），所以第三层菜单列了9行

注意数组中每一行的第1个数组，是索引号，先列举一个实际的例子进行分析：

上图就是一个实际的4级菜单要显示的内容，每个条目前，标记了索引号（0~25），即对应数组在定义的索引号。

比如数组关于第0层和第1层的定义:

//第0层

{0,0,0,1,(\*fun\_0)},

//第1层

{1,4,2, 5,(\*fun\_a1)},

{2,1,3, 9,(\*fun\_b1)},

{3,2,4,13,(\*fun\_c1)},

{4,3,1, 0,(\*fun\_d1)},

先看第一行：索引是0，显示欢迎界面；后面的两个0表示此时按“上翻”和“下翻”无效，继续显示欢迎界面；再后面的1表示按下“确认”按钮后，跳转到索引1处（即显示第1级目录，且指向第1级的第1个子项）；最后是此索引要显示的具体内容，fun\_0就是控制屏幕显示欢迎界面

再看第二行：索引是1，显示第1级目录，且指向第1级的第1个子项（天气）；后面的4表示此时按“上翻”跳转到索引4，即显示第1级目录，且指向第1级的第4个子项（Return）；再后面的2表示此时按“下翻”跳转到索引2，即显示第1级目录，且指向第1级的第2个子项（音乐）；再后面的5表示按下“确认”按钮后，跳转到索引5处（即显示第2级目录，且指向第2级的第1个子项-杭州）；最后是此索引要显示的具体内容，fun\_a1就是控制屏幕显示第1级目录，且指向第1级的第1个子项（天气）

其它行的含义与之类似

通过分析，不难发现，这些数组在空间上的关系：

对于菜单的最底层，因为没有上翻和下翻的功能需求，因此每行的前3个数字都是当前的索引号：

//第3层

{17,17,17,5,(\*fun\_a31)},

{18,18,18,6,(\*fun\_a32)},

{19,19,19,7,(\*fun\_a33)},

{20,20,20, 9,(\*fun\_b31)},

{21,21,21,10,(\*fun\_b32)},

{22,22,22,11,(\*fun\_b33)},

{23,23,23,13,(\*fun\_c31)},

{24,24,24,14,(\*fun\_c32)},

{25,25,25,15,(\*fun\_c33)},

2.2 具体的显示函数

对于函数要显示的具体内容，根据自己的实现需要显示即可。

这里我使用的是OLED屏幕，借助U8g2图形库进行内容显示，以下是部分显示示例：

/\*\*\*\*\*\*\*\*\*第1层\*\*\*\*\*\*\*\*\*\*\*/

void fun\_a1()

{

u8g2\_DrawStr(&u8g2,0,16,">");

u8g2\_DrawStr(&u8g2,16,16,"[1]Weather");

u8g2\_DrawStr(&u8g2,16,32,"[2]Music");

u8g2\_DrawStr(&u8g2,16,48,"[3]Device Info");

u8g2\_DrawStr(&u8g2,16,64,"<--");

}

void fun\_b1()

{

u8g2\_DrawStr(&u8g2,0,32,">");

u8g2\_DrawStr(&u8g2,16,16,"[1]Weather");

u8g2\_DrawStr(&u8g2,16,32,"[2]Music");

u8g2\_DrawStr(&u8g2,16,48,"[3]Device Info");

u8g2\_DrawStr(&u8g2,16,64,"<--");

}

void fun\_c1()

{

u8g2\_DrawStr(&u8g2,0,48,">");

u8g2\_DrawStr(&u8g2,16,16,"[1]Weather");

u8g2\_DrawStr(&u8g2,16,32,"[2]Music");

u8g2\_DrawStr(&u8g2,16,48,"[3]Device Info");

u8g2\_DrawStr(&u8g2,16,64,"<--");

}

void fun\_d1()

{

u8g2\_DrawStr(&u8g2,0,64,">");

u8g2\_DrawStr(&u8g2,16,16,"[1]Weather");

u8g2\_DrawStr(&u8g2,16,32,"[2]Music");

u8g2\_DrawStr(&u8g2,16,48,"[3]Device Info");

u8g2\_DrawStr(&u8g2,16,64,"<--");

}

/\*\*\*\*\*\*\*\*\*第2层\*\*\*\*\*\*\*\*\*\*\*/

void fun\_a21()

{

u8g2\_DrawStr(&u8g2,0,16,">");

u8g2\_DrawStr(&u8g2,16,16,"\* HangZhou");

u8g2\_DrawStr(&u8g2,16,32,"\* BeiJing");

u8g2\_DrawStr(&u8g2,16,48,"\* ShangHai");

u8g2\_DrawStr(&u8g2,16,64,"<--");

}

47

2.3 按键切换页面

页面的切换，这里里简单的按钮轮询为例，比如初始显示欢迎界面的状态下，按下不同按键后，通过数组查表，确定要跳转到的索引号，然后根据索引号，通过函数指针执行索引号对应的显示函数，即实现了一次页面切换。

然后，就是在新的页面状态，收到下一个按钮指令，再切换到下一个显示状态。

void (\*current\_operation\_index)(); //定义一个函数指针

//...

while(1)

{

if((KEY1==0)||(KEY2==0)||(KEY3==0))

{

delay\_ms(10);//消抖

if(KEY1==0)

{

func\_index = table[func\_index].up; //向上翻

while(!KEY1);//松手检测

}

if(KEY2==0)

{

func\_index = table[func\_index].down; //向下翻

while(!KEY2);

}

if(KEY3==0)

{

func\_index = table[func\_index].enter; //确认

while(!KEY3);

}

}

if (func\_index != last\_index)

{

current\_operation\_index = table[func\_index].current\_operation;

u8g2\_ClearBuffer(&u8g2);

(\*current\_operation\_index)();//执行当前操作函数

u8g2\_SendBuffer(&u8g2);

last\_index = func\_index;

}

}3 演示

测试效果如附件

4 总结

本篇介绍了一种简易的多级菜单的显示方法，本质是通过数组查表，实现各级菜单的各个页面（状态）的切换（跳转），并在STM32上编程实现，通过OLED屏幕，以及借助U8g2图形库，测试了多级菜单的显示功能。