EH100按键菜单控制流程图

本文档有侯东良编辑，联系邮箱：[dlhou@163.com](mailto:dlhou@163.com)

2020.11.13





1. 设计思路
2. 定义全局变量 char MenuPlace[4] ={1,0,0,0}; 此全局变量记录当前菜单的位置，初始值为一级菜单。全局变量TopLevelOption 表示上一级选项，在进入下一级菜单时需要记录上一级菜单选项。全局变量CurrentLevelOption用来记录当前选项值，在进入下一级菜单时将当前选项值赋值给TopLevelOption，同时赋值给MenuPlace对应位。 全局变量 MenuLevel 表示当前所在的菜单级别，初始值为1，表示初始在一级菜单。

|  |  |  |  |
| --- | --- | --- | --- |
| MenuPlace菜单的位置变量说明 | | | |
| 位3 一级菜单标识 | 位2 二级菜单标识 | 位1 三级菜单标识 | 位0 四级菜单标识 |
| 取值范围1-7；表明所处一级菜单的位置 | 取值范围0~16；具体看实际情况，表明处于二级菜单的哪一项 | 取值范围0~16；具体看实际情况，表明处于三级菜单的哪一项 | 取值范围0~16；具体看实际情况，表明处于四级菜单的哪一项 |
| 默认值1，表明处于一级菜单的第一项 | 默认值0，表明当前没有处于二级菜单 | 默认值0，表明当前没有处于三级菜单 | 默认值0，表明当前没有处于四级菜单 |
|  |  |  |  |

1. 设计步骤
2. 首次开机，进入一级菜单，此时 MenuPlace 等于char MenuPlace[4] ={1,0,0,0}，表示在一级菜单，并默认选中第一个选项。完成第一级菜单显示的函数是showMenu(char \* MenuPlace). 普通菜单背景绿色，字体色为暗红色。液晶屏为横向即宽 480，高320，如果字体为30，则只能显示10行。
3. 在第一级显示状态下:
   1. 如果按“向上箭头”，则对“一级菜单标识”进行减1，如果减的结果小于1，则无效；刷新菜单，高亮显示当前CurrentLevelOption所对应的菜单项。（背景蓝色，前景暗红色）。如果显示项在屏幕外边，则执行滚屏操作。一级菜单不存在此情况。只有大于10项的菜单会涉及此情况。
   2. 如果按“向下箭头”，则对“一级菜单标识”进行加1。在显示第一级菜单时，“一级菜单标识”小于等于应菜单的长度（7）。如果显示项在屏幕外边，则执行滚屏操作。
   3. 如果按下“向左”键，则返回上级菜单。对当前级别菜单标识清零。如果目前是一级菜单，则“向左”键无效。
   4. 如果按下“向右”键或“确认”键，则进入下一级菜单。在下一级菜单标识位填入“1”，即默认是第一项。同时将CurrentLevelOption(选项值)赋值给变量TopLevelOption，以备在菜单返回时使用。
4. 在第二级显示状态下:
   1. 如果按“向上箭头”，则对“二级菜单标识”进行减1，如果减的结果小于1，则无效；如果显示项在屏幕外边，则执行滚屏操作。
   2. 如果按“向下箭头”，则对“二级菜单标识”进行加1。在显示第二级菜单时，“二级菜单标识”小于等于对应菜单的长度。如果显示项在屏幕外边，则执行滚屏操作。
   3. 如果按下“向左”键，则返回上级菜单。对当前级别菜单标识清零。如果目前是二级菜单，则“向左”键会回到一级菜单，并选中下级菜单的上级菜单，可以由全局变量TopLevelOption提供上一级菜单的选项。
   4. 如果按下“向右”键或“确认”键，则进入下一级菜单。在下一级菜单标识位填入“1”，即默认是第一项。在第二级菜单下，有些菜单没有第三级菜单。对于那些没有第三级菜单的项，则当按下“向右”键时无效。当按下“确认”键时，调用对应按键的处理程序。同时将CurrentLevelOption(选项值)赋值给变量TopLevelOption，以备在菜单返回时使用。
5. 在第三级显示状态下:
   1. 如果按“向上箭头”，则对“三级菜单标识”进行减1，如果减的结果小于1，则无效；
   2. 如果按“向下箭头”，则对“三级菜单标识”进行加1。在显示第三级菜单时，“三级菜单标识”小于等于对应菜单的长度。
   3. 如果按下“向左”键，则返回上级菜单，对当前级别菜单标识清零(MenuPlace对应位)，MenuLevel值减1，如果减1后，值小于1，则无效，MenuLevel值重新恢复为1。如果目前是三级菜单，则“向左”键会回到二级菜单，并选中MenuPlace中由MenuLevel对应位的值。
   4. 如果按下“向右”键或“确认”键，则进入下一级菜单。在下一级菜单标识位填入“1”，即默认是第一项。在第三级菜单下，有些菜单没有第四级菜单。对于那些没有第三级菜单的项，则当按下“向右”键时无效。当按下“确认”键时，调用对应按键的处理程序。同时将CurrentLevelOption(选项值)赋值给变量TopLevelOption，以备在菜单返回时使用。
6. 在第四级显示状态下:
   1. 如果按“向上箭头”，则对“四级菜单标识”进行减1，如果减的结果小于1，则无效；
   2. 如果按“向下箭头”，则对“四级菜单标识”进行加1。在显示第四级菜单时，“四级菜单标识”小于等于对应菜单的长度。
   3. 如果按下“向左”键，则返回上级菜单，对当前级别菜单标识清零(MenuPlace对应位)，MenuLevel值减1，如果减1后，值小于1，则无效，MenuLevel值重新恢复为1。如果目前是四级菜单，则“向左”键会回到三级菜单，并选中MenuPlace中由MenuLevel对应位值的选项。
   4. 如果按下“向右”键，无效。当按下“确认”键时，调用对应按键的处理程序。
7. showMenu(char \* menuPlace) 函数说明
8. 全局变量MenuLevel，TopLevelOption，CurrentLevelOption，i,分别记录当前菜单级别，上级菜单选项，当前级菜单选项。默认值三者都为0.
9. 给MenuLevel，CurrentLevelOption赋值。

For(i=3;i>0;i--)

{

If(menuPlace[i]>0)

{

MenuLevel=i+1;

CurrentLevelOption=menuLevel[i];

Break;

}

}

1. 显示MenuLevel级菜单。

如果是第一级，则利用for循环把第一级的7个选项都显示出来，同时加亮显示当先选中的选项。If(MenuLevel==1) 执行如下语句：

设置背景色绿色，设置前景色暗红色

For(i=0;i<7;i++)

{

显示第i行。MENU0[7][6]

}

设置加亮背景色为蓝色

显示第CurrentLevelOption行。

1. 如果是第二级，先利用switch 语句，判断是第一级的那个菜单项。

Switch(TopLevelOption)

{

Case 1:

显示MENU1[7][12]菜单;

Case 2:

显示MENU2[11][13]菜单;

Case 3:

显示MENU3[9][12]菜单;

Case 4:

显示MENU4[8][8]菜单;

Case 5:

显示MENU5[5][14]菜单;

Case 6:

显示MENU6[9][20]]菜单;

Case 7:

显示MENU7[4][18]菜单;

}

1. 如果是第三级，先利用switch 语句，判断是第二级的那个菜单项。

Switch(MenuPlace[0])

{

Case 1: //第1级为1.控制

Switch(MenuPlace[1])

{

Case 1: //第2级 为1.1

显示1.1 MENU11[6][14];

Case 2: //第2级 为1.2

显示1.2 MENU12[3][14];

Case 3: //第2级 为1.3

显示1.3 MENU13[6][14];

Case 4: //第2级 为1.4

显示1.4 MENU14[7][18];

Case 5: //第2级 为1.5

显示1.5 MENU15[16][15];

Case 6: //第2级 为1.6

显示1.6 MENU16[2][10];

Case 7: //第2级 为1.7

显示1.7 MENU17[9][16];

}

Case 2: //第1级为2.设备

Switch(MenuPlace[1])

{

Case 1: //第2级 为2.1

显示2.1 目前该菜单未定];

Case 4: //第2级 为2.4

显示2.2 MENU12[3][14];

Case 5: //第2级 为2.5

显示2.5 MENU13[6][14];

Case 6: //第2级 为2.6

显示2.6 MENU14[7][18];

Case 7: //第2级 为2.7

显示2.7 MENU15[16][15];

Case 8: //第2级 为2.8

显示2.8 MENU16[2][10];

Case 9: //第2级 为2.9

显示2.9 MENU17[9][16];

Case 10: //第2级 为2.7

显示2.10 MENU17[9][16];

Case 11: //第2级 为2.7

显示2.11 MENU17[9][16];

}

Case 3: //第1级为3.管理

Switch(MenuPlace[1])

{

Case 1: //第2级 为3.1

显示3.1 MENU31[5][14];

Case 2: //第2级 为3.2 动物日龄

3.2 没有子菜单; 直接执行“动物日龄”对应函数

........

Case 11: //第2级 为2.11

显示2.11 MENU2B[3][16];

}

Case 4: //第1级为4.历史

Switch(MenuPlace[1])

{

Case 1: //第2级 为3.1

显示3.1 MENU31[5][14];

Case 2: //第2级 为3.2 动物日龄

3.2 没有子菜单; 直接执行“动物日龄”对应函数

........

Case 11: //第2级 为2.11

显示2.11 MENU2B[3][16];

}

Case 5: //第1级为5.校准

Switch(MenuPlace[1])

{

Case 1: //第2级 为3.1

显示3.1 MENU31[5][14];

Case 2: //第2级 为3.2 动物日龄

3.2 没有子菜单; 直接执行“动物日龄”对应函数

........

Case 11: //第2级 为2.11

显示2.11 MENU2B[3][16];

}

Case 6: //第1级为6.配置

Switch(MenuPlace[1])

{

Case 1: //第2级 为3.1

显示3.1 MENU31[5][14];

Break;

Case 2: //第2级 为3.2 动物日龄

3.2 没有子菜单; 直接执行“动物日龄”对应函数

........

Case 11: //第2级 为2.11

显示2.11 MENU2B[3][16];

}

Case 7: //第1级为7.测试

Switch(MenuPlace[1])

{

Case 1: //第2级 为3.1

显示7.1 没有子菜单; 直接执行“继电器测试”对应函数

Case 2: //第2级 为7.2 模拟量输出测试

3.2 没有子菜单; 直接执行“模拟量输出测试”对应函数

........

Case 11: //第2级 为2.11

显示2.11 MENU2B[3][16];

}

}

1. 如果是第四级，先利用switch 语句，判断是第一、二、三级是哪个菜单项。

Switch(MenuPlace[0])

{

Case 2: //只有2. 设备菜单有四级子菜单

Switch(MenuPlace[1])

{

Case 6: //只有2.6 有四级子菜单

Switch(MenuPlace[])

{

Case 1: //

//显示2.6.1 MENU261[4][16]

Break;

Case 2: //

//显示2.6.2 MENU262[4][16]

Break;

Case 3: //

//显示2.6.3 MENU263[3][16]

Break;

}

}

}

1. 按键处理函数DealwithKey(char keyNum)说明
2. 该参数是在按键中断中由ch450\_ReadKey()读取键值函数完成。返回值送给KeyNumber。
3. 并且在按键中断中我们要做一个标识NewKey，表示有新的按键（当NewKey为1时）。
4. 当调用处理按键函数后，对NewKey值清零。
5. 当按下“确定”键时，判断当前菜单是否还有子菜单，如果有进入子菜单。如果当前为叶子节点，则调用对应函数。设计思路是当按键移动菜单叶子节点是对全局变量LeafFlag置1；如果非叶子节点，LeafFlag=0.
6. 利用switch对按键进行处理

Void DealwithKey(char keyNum)

{ int MenuOption=0,i;

Switch(keyNum)

{

Case: 0x42

//SK17（确定）

If(LeafFlag==0) //不是叶子节点

{ 进入下一级菜单；相当于按下了“向右”键}

Else

{ MenuOption=MenuPlace[0];

For(i=1;i<4;i++)

{

MenuOption<<=4;

MenuOption+=MenuPlace[i];

}

}

Switch(MenuOption)

{

Case 0x1110:

//执行叶子节点函数1.1.1 日龄

Break;

Case 0x1120:

//执行叶子节点函数1.1.2 目标温度

Break;

}

Break;

Case: 0x43

//SK9（+/-）

Break;

Case: 0x44

//SK1（翻页）

Break;

Case: 0x4A

//SK18（上）

Break;

Case: 0x4B

//SK10（7）

Break;

Case: 0x4C

//SK2（9）

Break;

Case: 0x52

//SK19（菜单）

showMenu(MenuPlace) ; //调用showMenu(char \* menuPlace) 函数显示菜单 Break;

Case: 0x53

//SK11（4）

Break;

Case: 0x54

//SK3（6）

Break;

Case: 0x5A

//SK20（左）

Break;

Case: 0x5B

//SK12（1）

Break;

Case: 0x5C

//SK4（3）

Break;

Case: 0x63

//SK13（修改）

Break;

Case: 0x64

//SK5（0）

Break;

Case: 0x6B

//SK14（右）

Break;

Case: 0x6C

//SK6（8）

Break;

Case: 0x73

//SK15（删除）

Break;

Case: 0x74

//SK7（5）

Break;

Case: 0x7B

//SK16（下）

Break;

Case: 0x7C

//SK8（2）

Break;

}

}