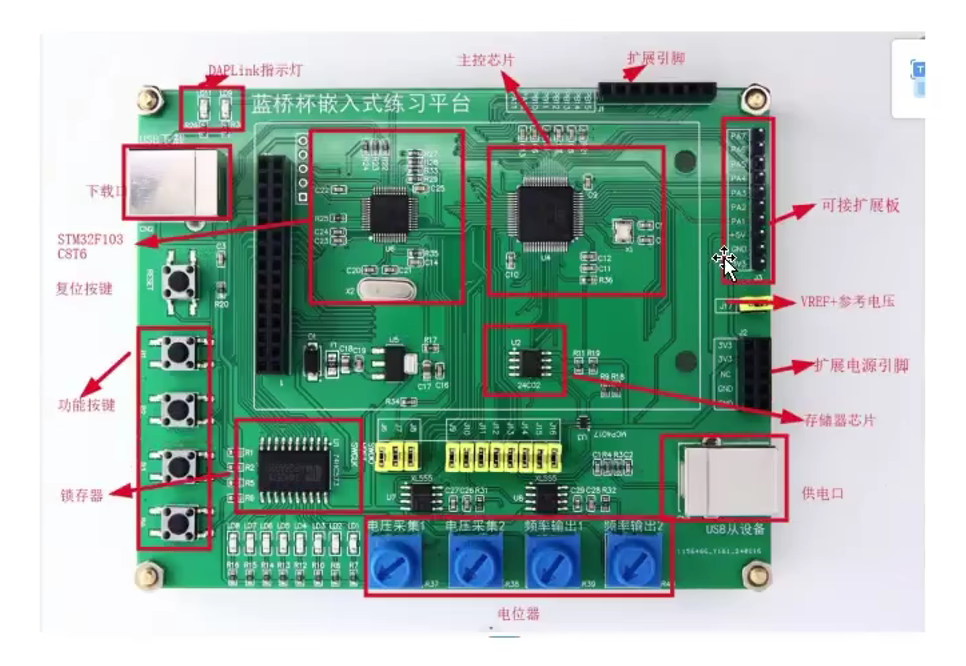
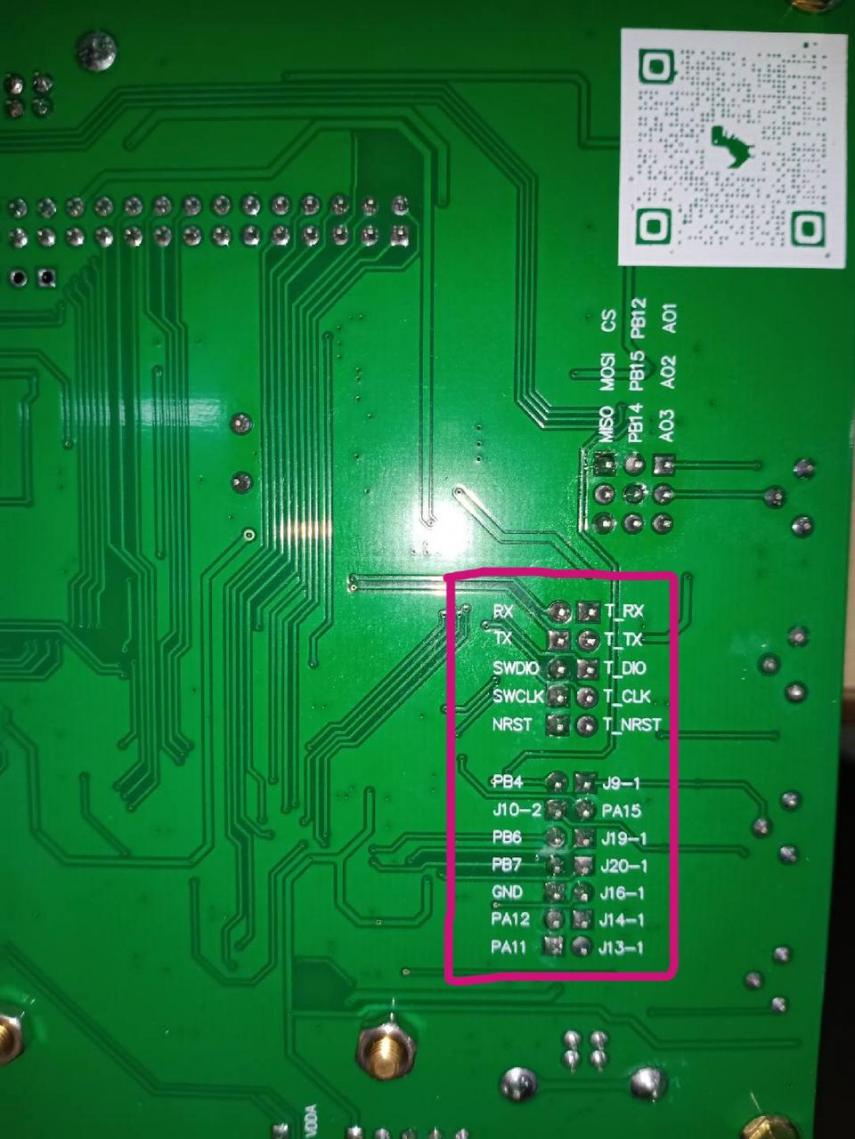
**请打开“导航窗格”以更清晰地查看文件结构**

**00开发板总体介绍**





1.外部晶振**24MHz**

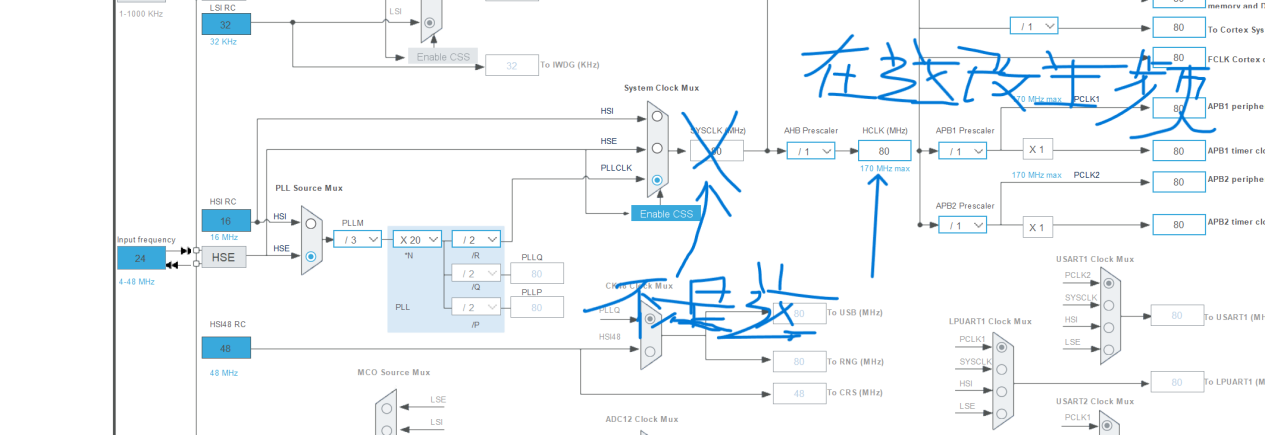
2.LED连有锁存器，使用时需要先对锁存器操作

3.显示屏：20列，10行；在官方资源包的BSP中，行下标从0开始

**01初始化配置**

1.RCC-**High Speed Clock**选择**Crystal/Ceramic Resonator**、SYS-**debug**选择**serial wire（串行线路）**

2.时钟配置里面，将外部晶振输入值改为**24**，选中**HSE、PLLCLK**，总时钟频率设为**80MHz（具体看题目需求）**



3.Code Generator中**Generated files**勾选**生成外设初始化的.c和.h**文件

4.Project Manager-Toolchain/IDE选择**MDK-ARM**

5.魔术棒中**debug**选择**cmsis dap**，进入**settings**选择**reset and run**

6.代码需要写在begin和end之间，特别是在主函数的while循环部分，否则重新生成文件后begin和end之外的代码可能会被清除；**建议在重新生成工程文件前先保存一份现有工程**

7.注意：如果点击生成后没有keil的工程文件，可能是cubemx中的包版本选择错误，更换其他包版本试试

**02LED**

**0201配置**

**基础控制（亮灭、翻转）**

* PC8-PC15（LED）、PD2（锁存器控制）配置为**推挽输出**
* LED正极已连接3.3V，负极接引脚（即**共阳**极接法）
* LED默认电平设为HIGH，锁存器控制引脚的默认电平设为LOW（默认关闭LED和锁存器）

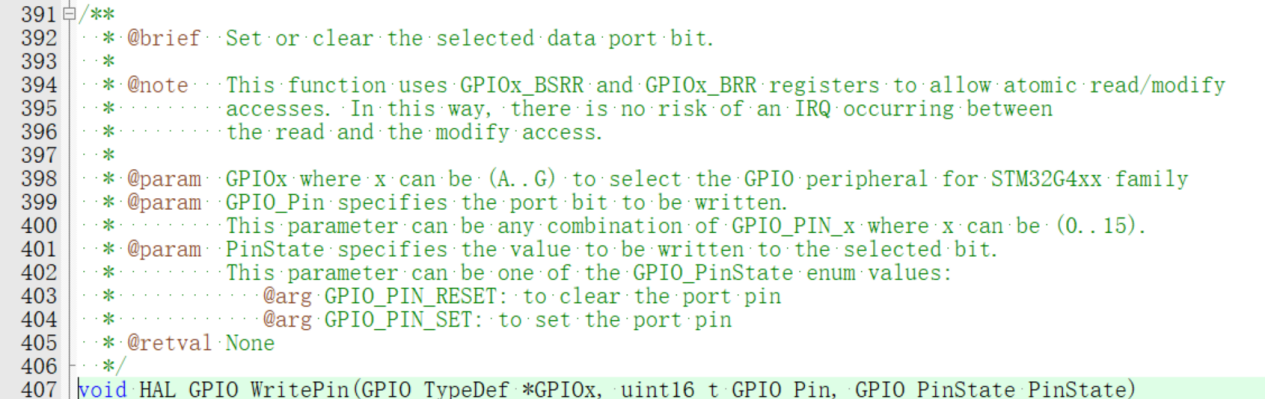
**流水灯**

* 在基础控制的配置上加一个定时器，配置为10ms定时中断

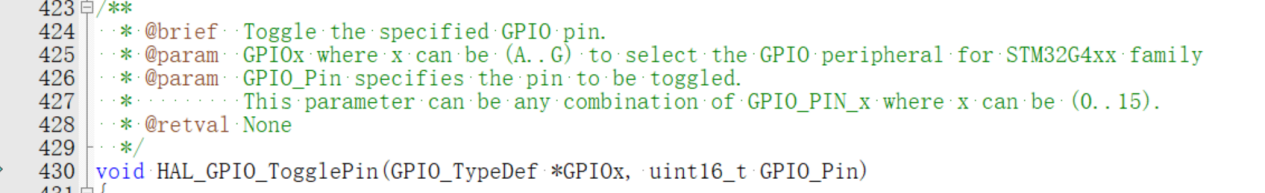
**0202相关函数**

**打开与关闭**

**HAL\_GPIO\_WritePin**



**HAL\_GPIO\_TogglePin**



**GPIO状态（全字母大写）**

@arg **GPIO\_PIN\_RESET**: to clear the port pin

@arg **GPIO\_PIN\_SET**: to set the port pin

**流水灯**

* 这种实现方法代码写起来会比较复杂，可找其他方案代替（比如直接操作寄存器）

**1.结构体：流动方向、当前点亮的LED编号、计数器（用于控制前进时间）、允许流动的标志位**

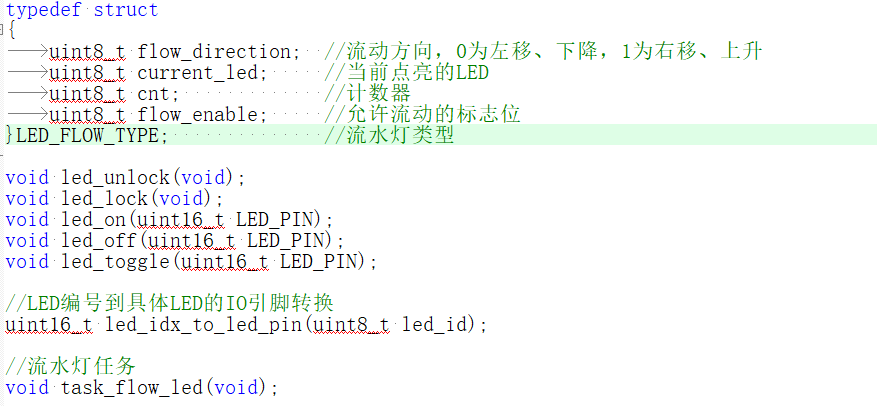
**2.LED编号转换为具体引脚（易忘）：编号从1开始，用switch语句返回对应LED\_Pin**

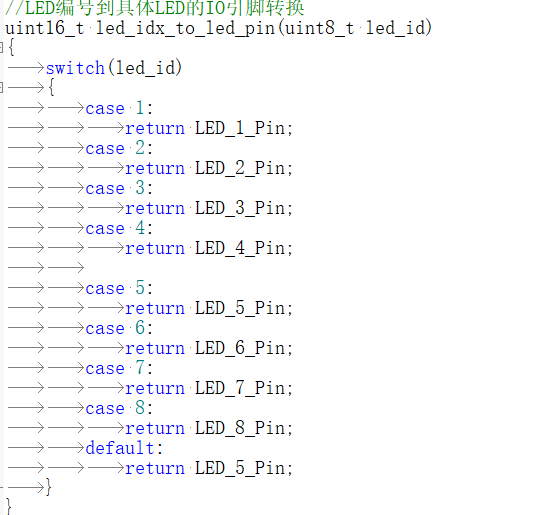
**3.流水灯任务：**

3.1.1 开启流水灯 -> 计数器到达40（具体参数根据实际而定） -> 清空计数器，用for循环找到对应要点亮的LED（编号匹配时点亮，否则熄灭） -> 根据流动方向重新设置当前点亮的LED编号

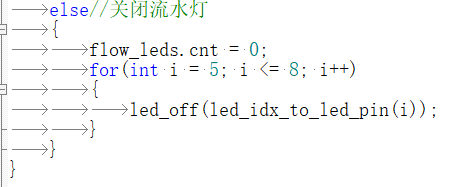
3.1.2 关闭流水灯 -> 计数器清零，所有流水灯关闭

3.2 在10ms定时中断中加入计时：开启流水灯时计数器自增











**0203相关文件**

stm32g4xx\_hal\_gpio.c

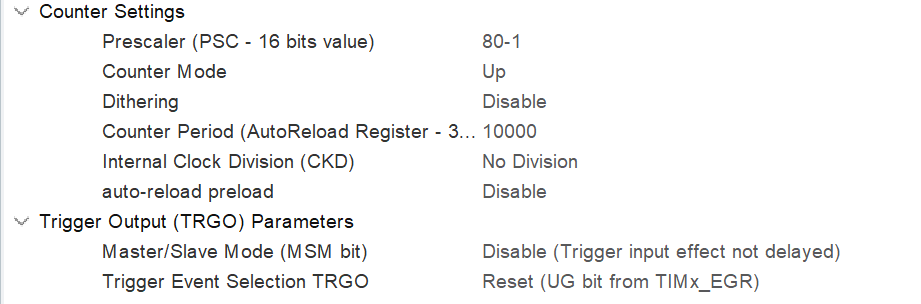
**0204注意事项**

* LED给高电平熄灭，给低电平点亮
* 使用LED时（三种模式中的任意一种），需要先解锁再传入电平
* 初始化后LED仍然打开原因：未处理LCD函数或没有初始化LED引脚
* 如果初始化后LED仍点亮，则先熄灭一个引脚（加一个临时代码并烧录即可）
* 使用LCD显示屏时，需要处理引脚冲突，同时LED函数（包括流水灯任务）不能在中断中使用
* LED引脚可能需要初始化为高电平才熄灭，即使做了引脚冲突处理（因为低电平时会点亮，而引脚默认低电平）

**03KEY**

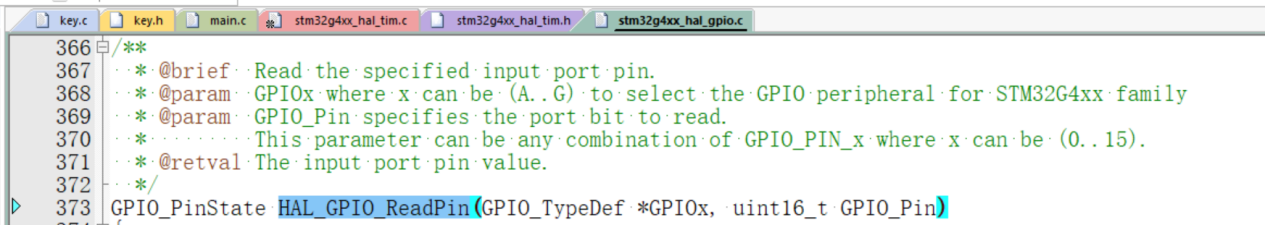
**0301配置**

* PB0-PB2（B1-B3）、PA0（B4）配置为浮空输入
* 设置定时器：**clock source**设为**internal clock**，Prescaler设为**80-1**，计数模式为向上计数，计数周期**（Counter Period）**为**10000-1**，使能更新中断；即**每10ms进入一次中断**

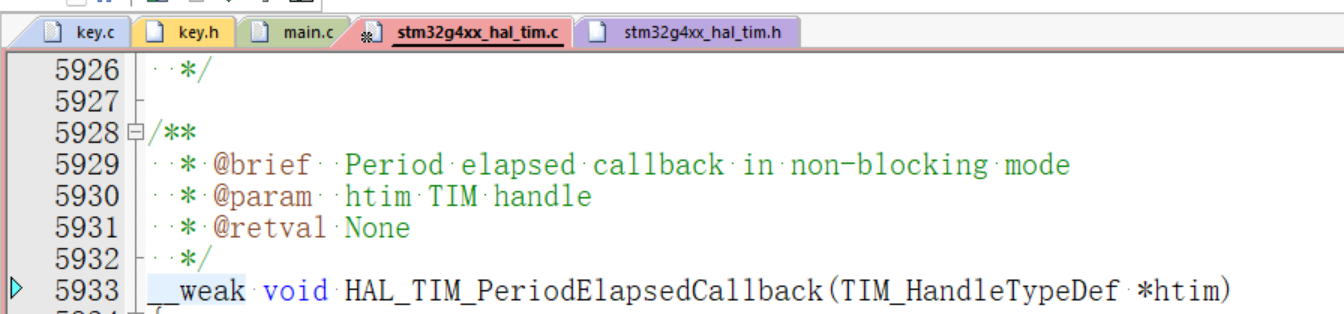


**0302相关函数**

**HAL\_GPIO\_ReadPin**



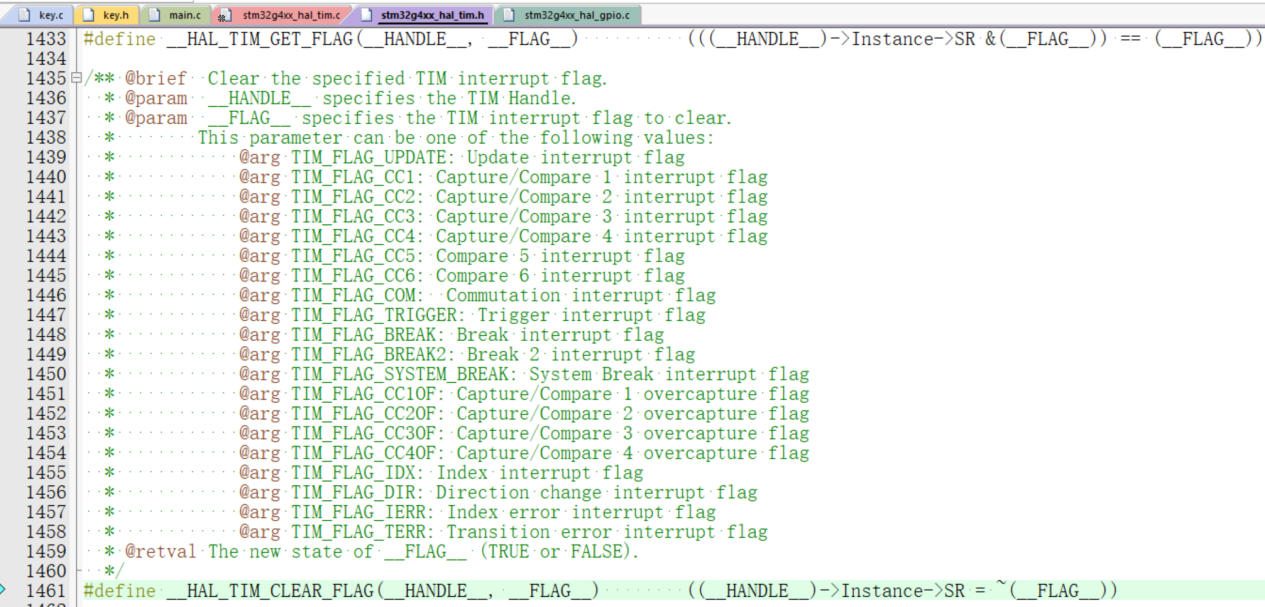
**HAL\_TIM\_PeriodElapsedCallback**



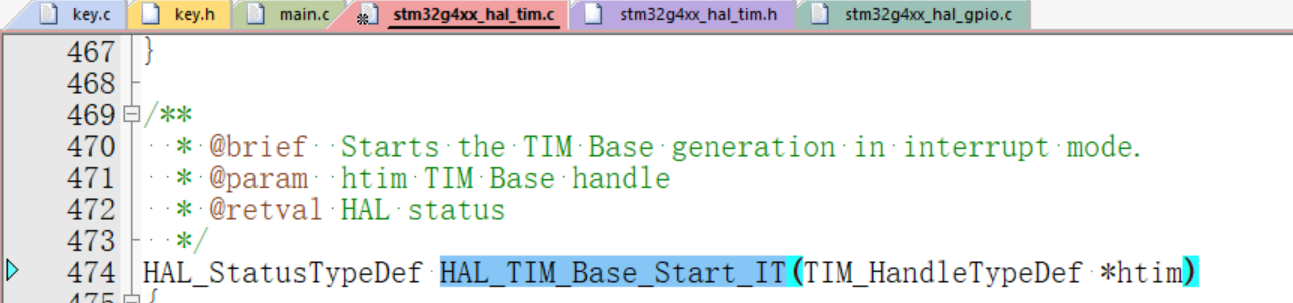
判断具体的定时器：if(htim->Instance == TIM2){ }

**\_\_HAL\_TIM\_CLEAR\_FLAG**

\_\_HAL\_TIM\_CLEAR\_FLAG(&htim2,**TIM\_FLAG\_UPDATE**);



**HAL\_TIM\_Base\_Start\_IT**



**短按松开时响应**

1. 结构体：当前引脚电平、评判状态（当前按键处理所在的步骤）、按键状态（用于外界读取）
2. 按键处理（10ms中断处理）：

* 获取当前按键引脚电平
* 初始状态：当按键按下时（此处为引脚电平是0），转到滤波状态
* 滤波状态：检查到按键还在按下，则进入确定按键按下状态；否则回到初始状态
* 确定按键按下状态：当检测到按键松开时（即引脚电平为1时），设置标志位

1. 读取按键状态：用一个临时变量接收对应按键的状态，然后将按键状态清空（用于继续读取按键），将临时变量传出



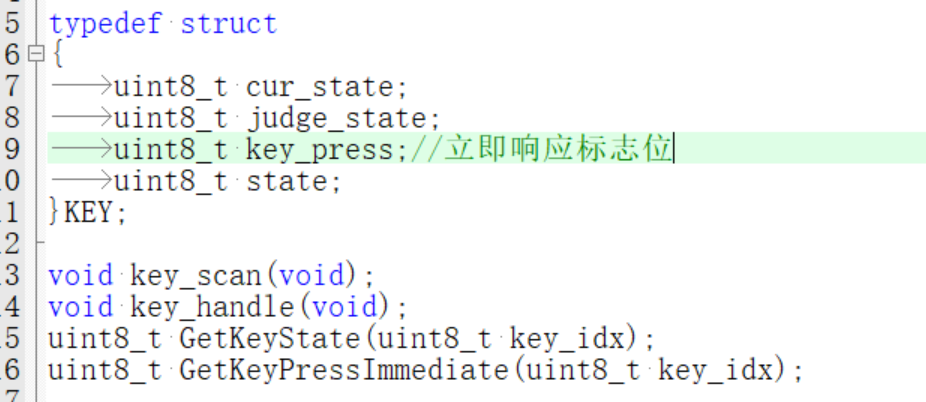




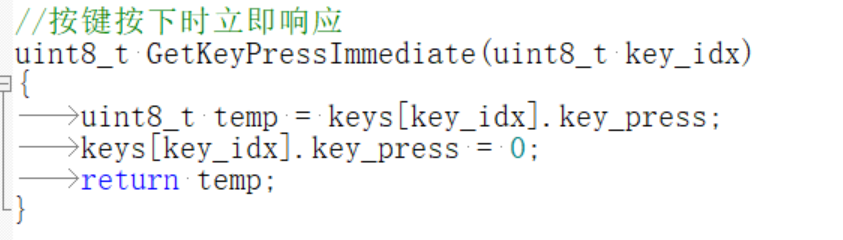


**按下时立即响应**

* 加一个标志位，按键按下时，在滤波阶段置1，否则置0
* 与读取短按一样，获取状态后立即清空标志位
* 注意：置位之后要由读取函数清理标志位，不能在按键处理函数中清空标志位



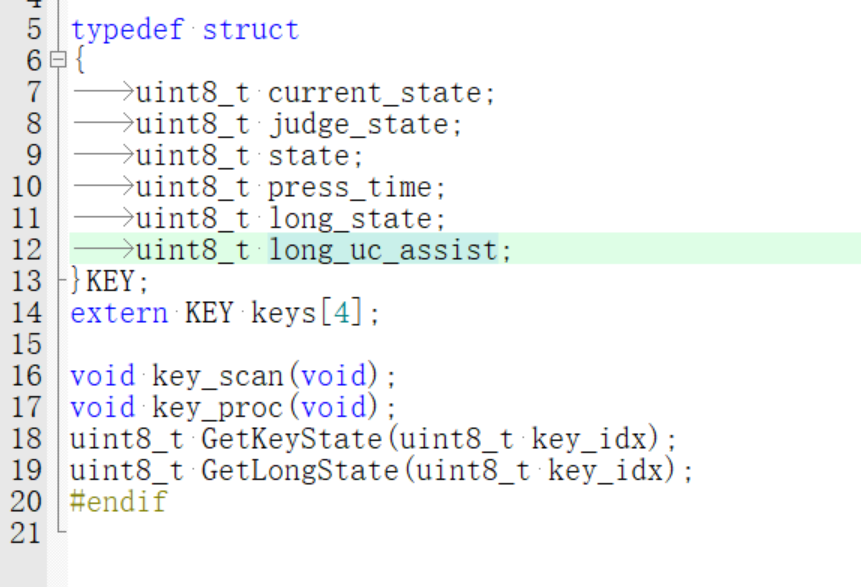


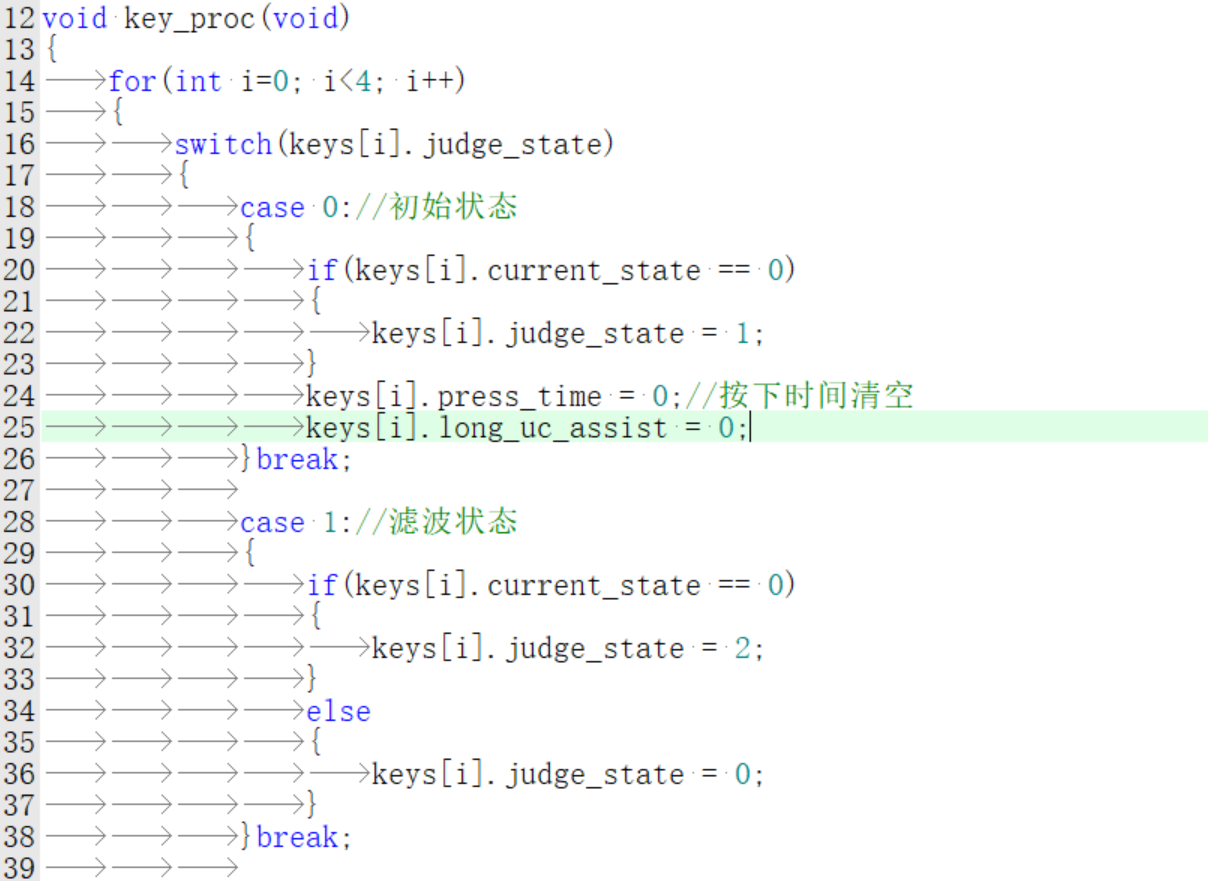


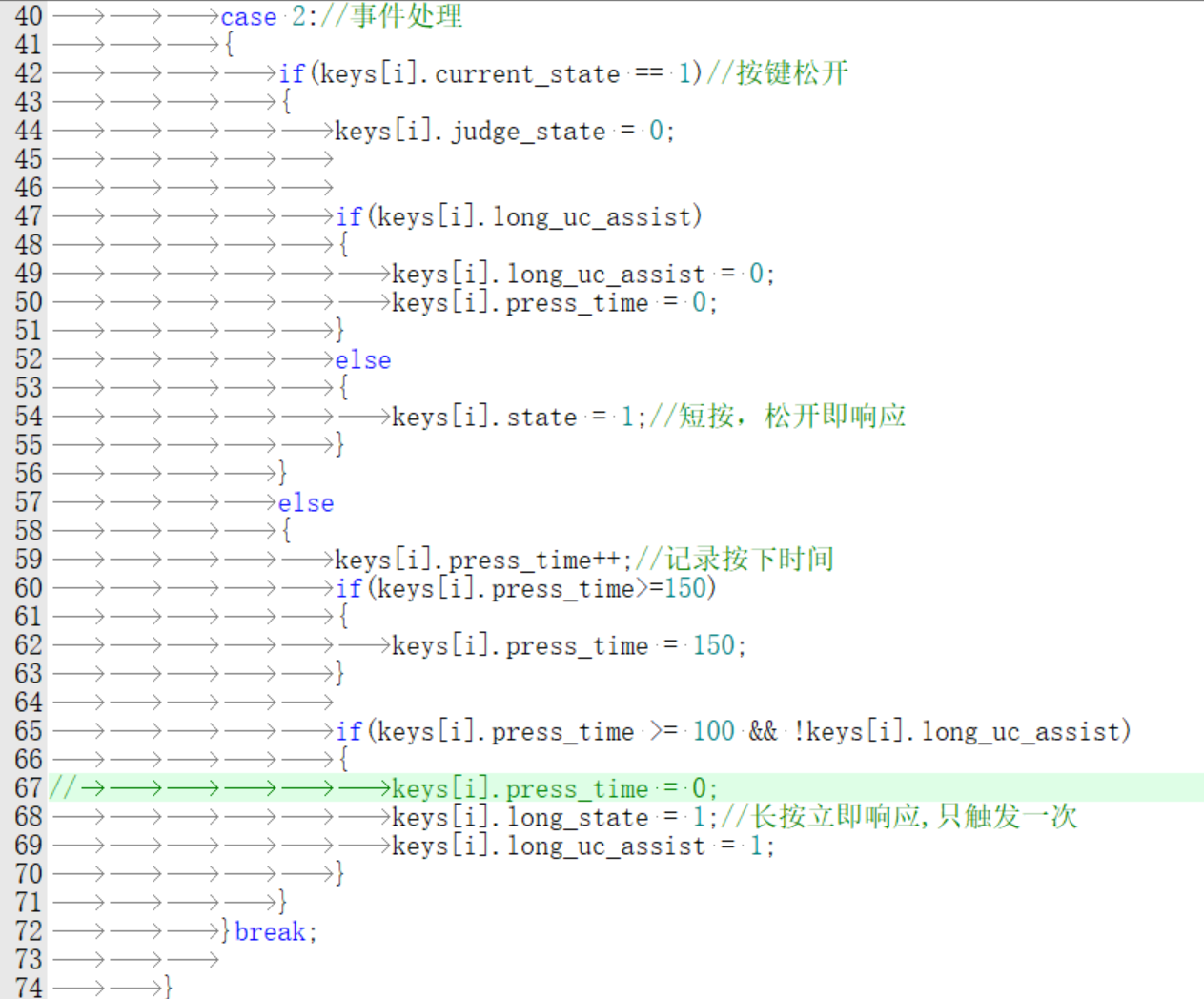
**长按**

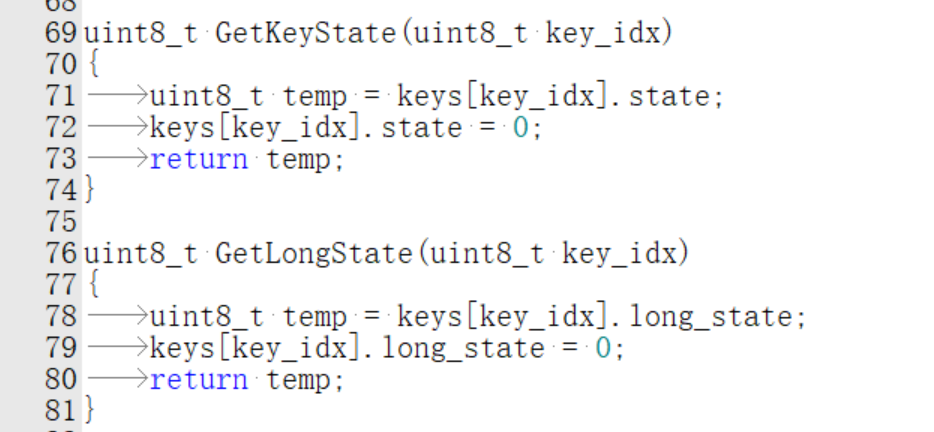
**立即响应+不持续触发（即响应一次）**

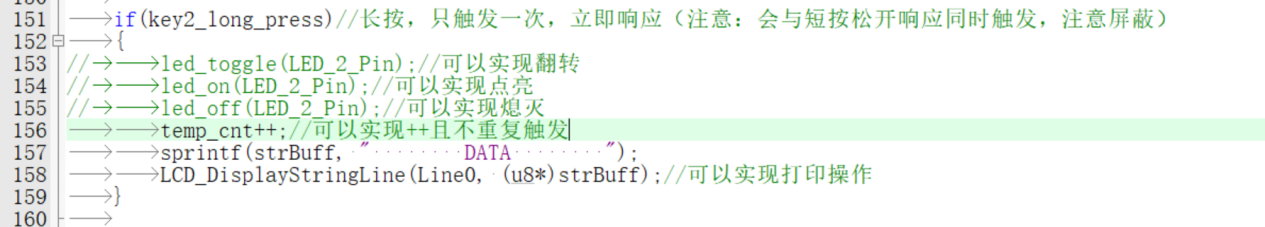
* 增加一个辅助标志位
* 长短按隔离：判断assist置位时不将短按置位
* 触发长按后对press\_time进行限幅处理







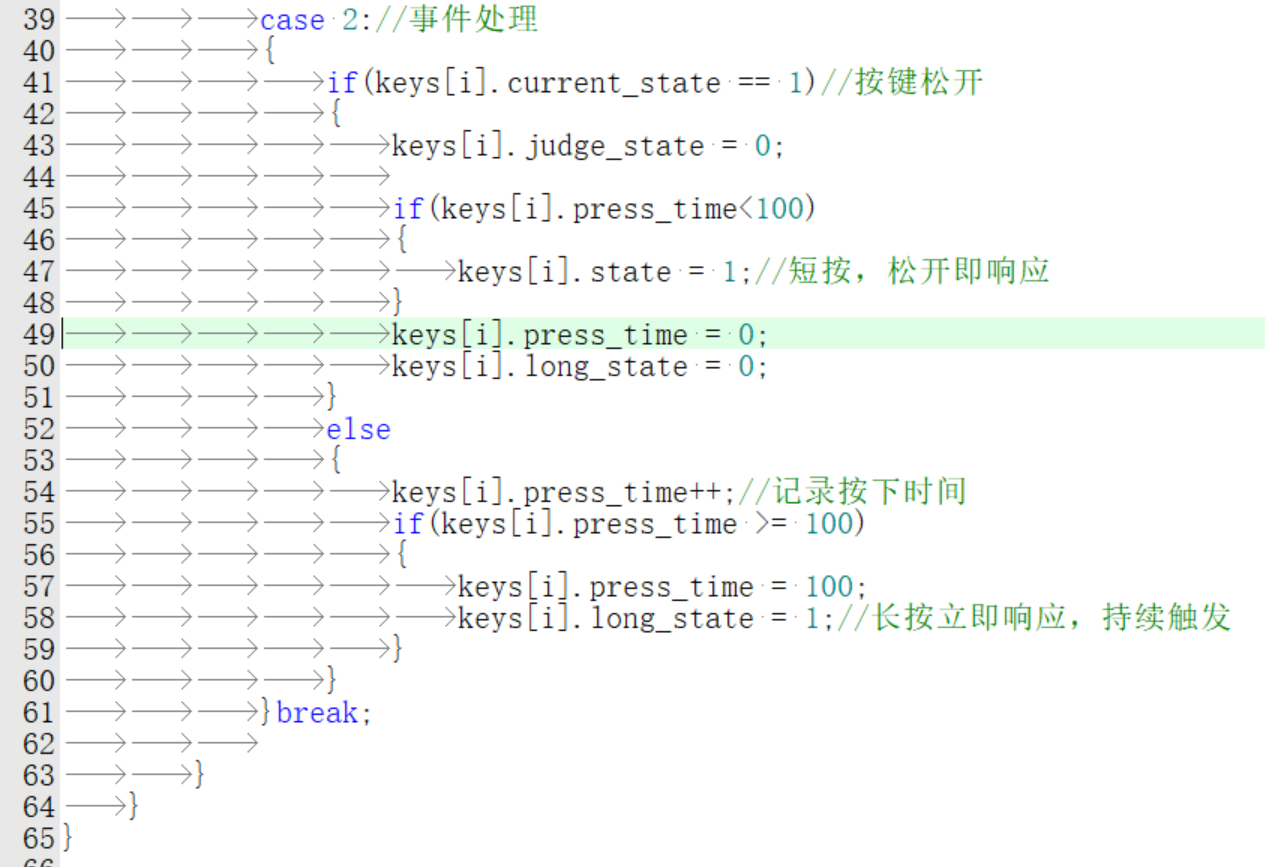




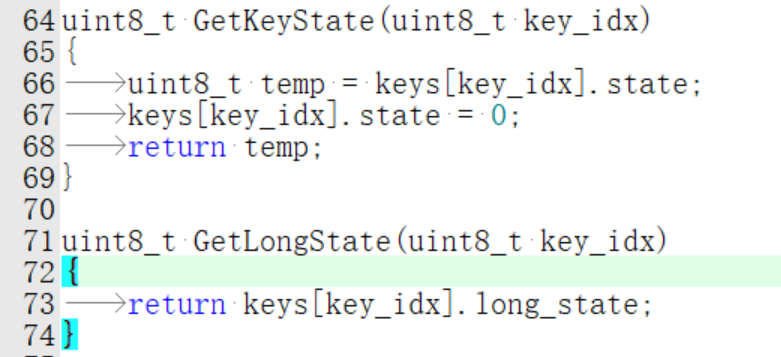
**立即响应+连续触发（即不松手时标志位一直置1）**

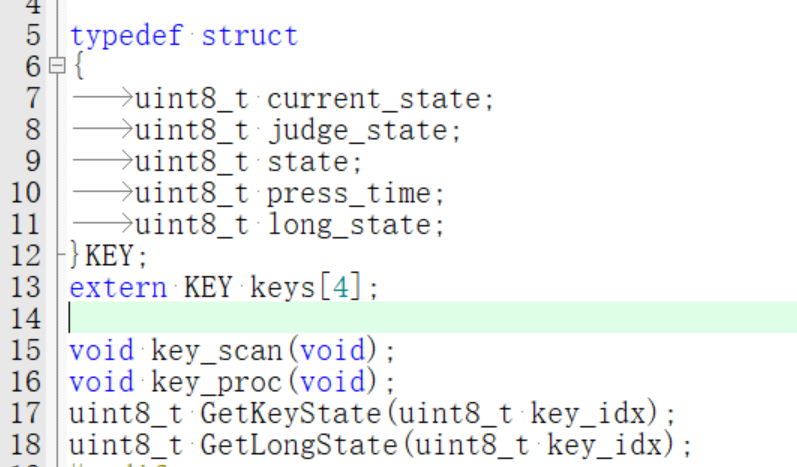
* 长短按隔离：判断长按标志置位时不将短按置位

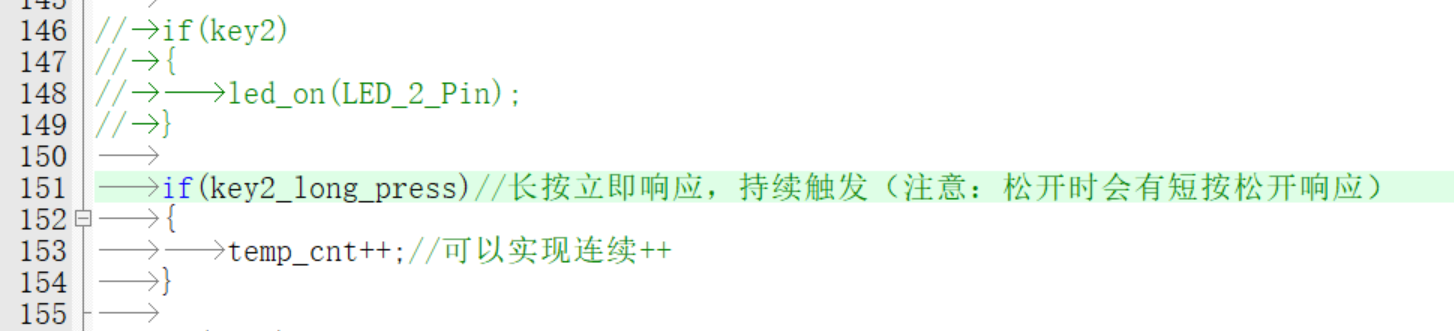




* 注意：此处长按标志位的清除在按键处理函数中，不是在状态读取函数内

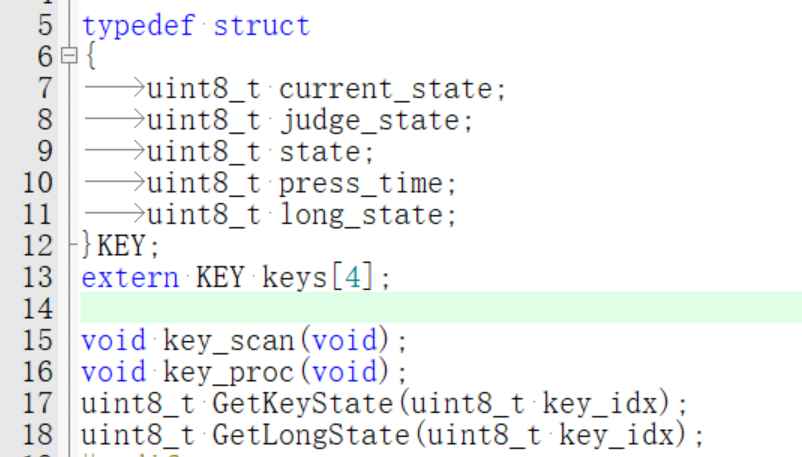






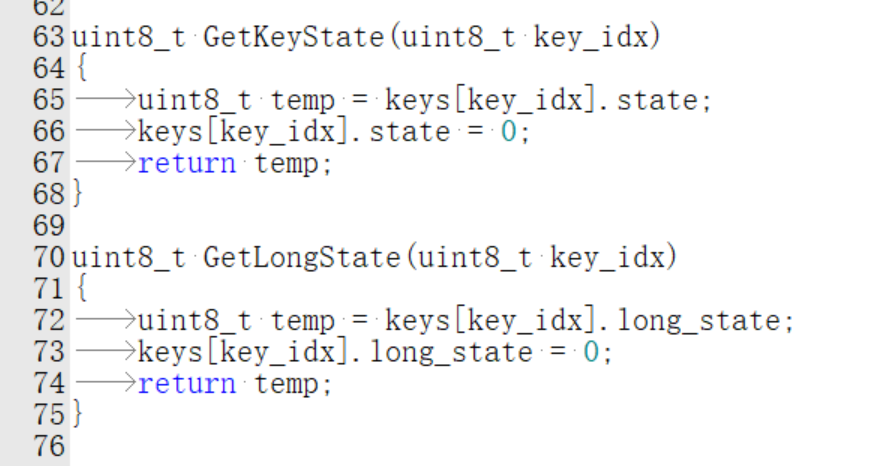
**松开再响应（判断到长按时松手再置位）**

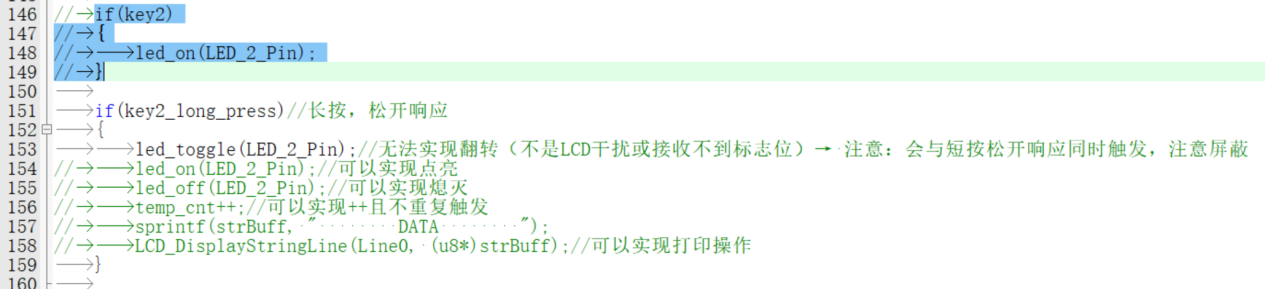
* 对press\_time进行限幅
* 松开时判断press\_time是否超限，设置对应的标志位
* 长短按隔离：判断是长按时短按不置位





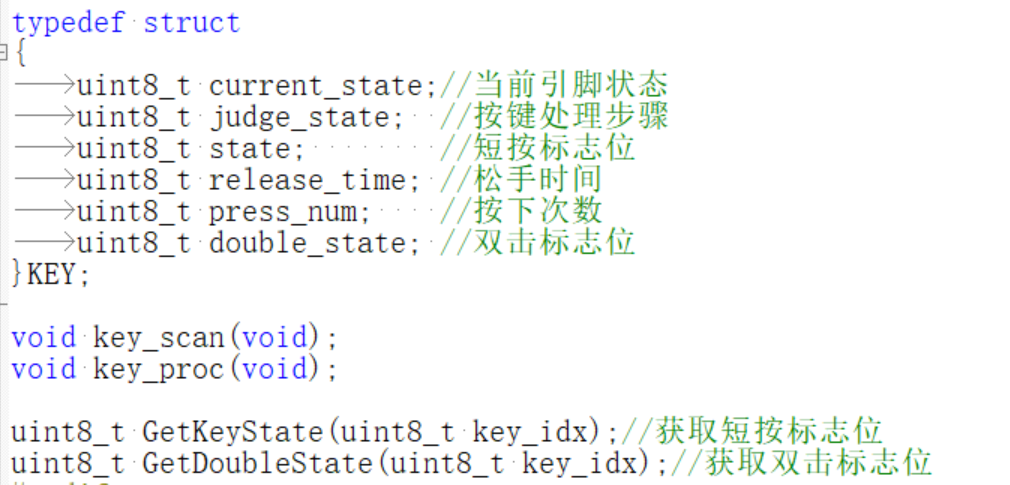




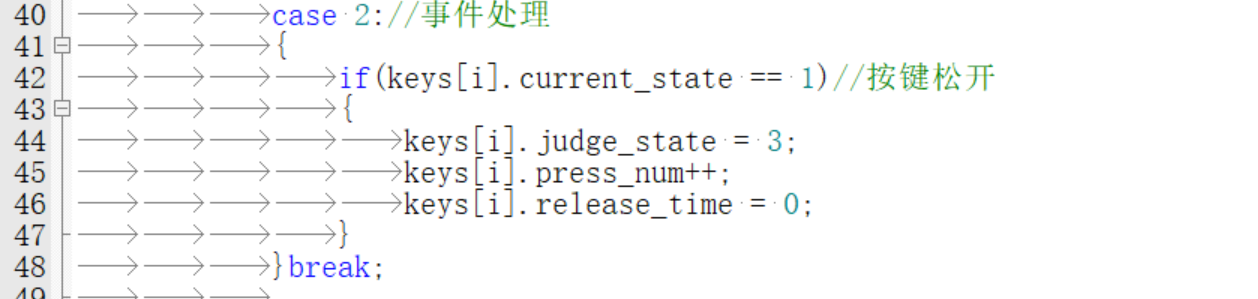


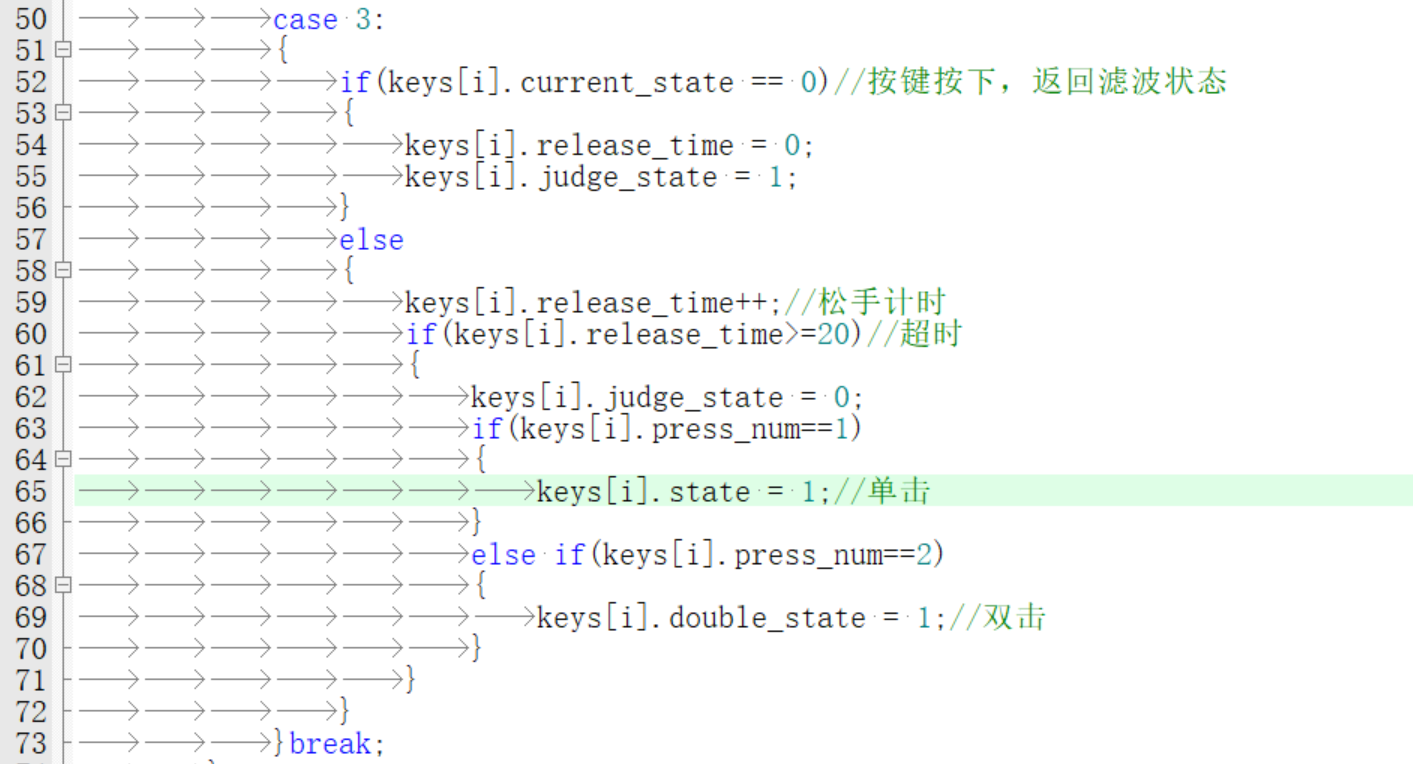
**单击+双击**

* 按键属性：加入松手时间和按下次数、双击标志位

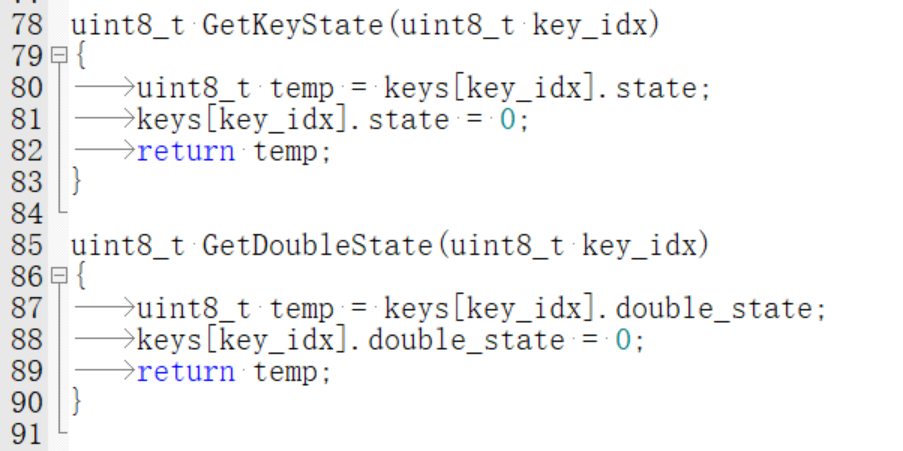


* 按键处理：多加一个case





* 标志位获取



**0303相关文件**

stm32g4xx\_hal\_tim.c

stm32g4xx\_hal\_gpio.c

stm32g4xx\_hal\_tim.h

**0304注意事项**

* 按键引脚外接高电平，按下时为低电平
* 使用定时器前需要清空标志位、开启定时器、**打开中断（CubeMx、代码-打开定时器）**
* 获取按键状态时用变量接收再做判断
* 按键处理函数中，switch每隔10ms切换一个case
* 短按立即响应需要放在滤波中
* 长按分三种：长按立即响应且响应一次、重复响应、松开再响应
* 长按松开响应与短按隔离：用判断语句进行隔离，检测到长按时不将短按置位

**04LCD**

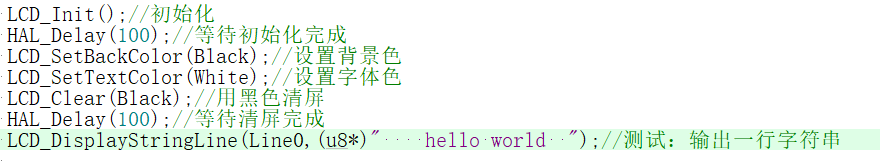
**0401配置**

无需cubemx配置

**0402函数**

**初始化**

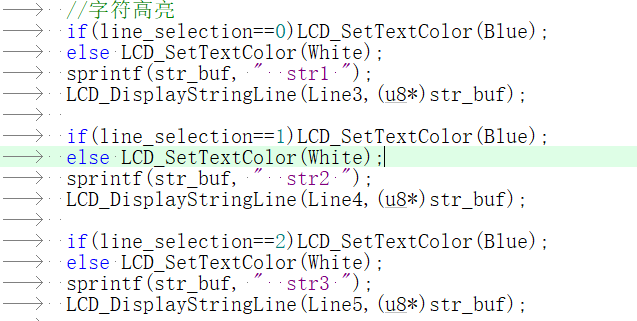
初始化 -> 等待初始化完毕 -> 设置颜色 -> 清屏 -> 等待清屏完成



**高亮显示**

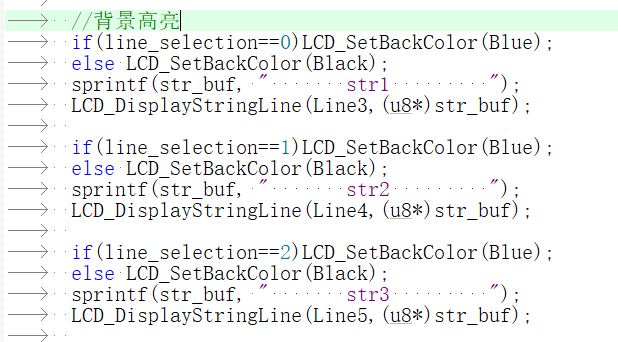
**字符高亮**

* 在字符串显示之前设置**字符**颜色



**背景高亮**

* 在字符串显示之前设置**背景**颜色



**一行内特定的字符串高亮**

* 复制DisplayStringLine函数，修改函数名称
* 增加：到达起始高亮字符、高亮字符范围内设置高亮颜色，其余字符设置默认颜色



**翻转显示**

* forward：正向显示
* inverse：翻转显示



**0403相关文件**

lcd.h

lcd.c

**0404注意事项**

* 显示屏有10行、20列，即一行最多可打印20个字符
* 行数从0开始（Line0）
* 高亮显示一整行时，多余的列需要**用空格补齐**
* 设置完颜色后，需要用背景色清屏
* strBuff大小建议设置为30，防止sprintf打印时**数组越界（数组越界时可能会影响其他变量的值）**