

《匠人手记》之十一

按键漫谈

作者：程序匠人

出处：《匠人的百宝箱》

1. 前言

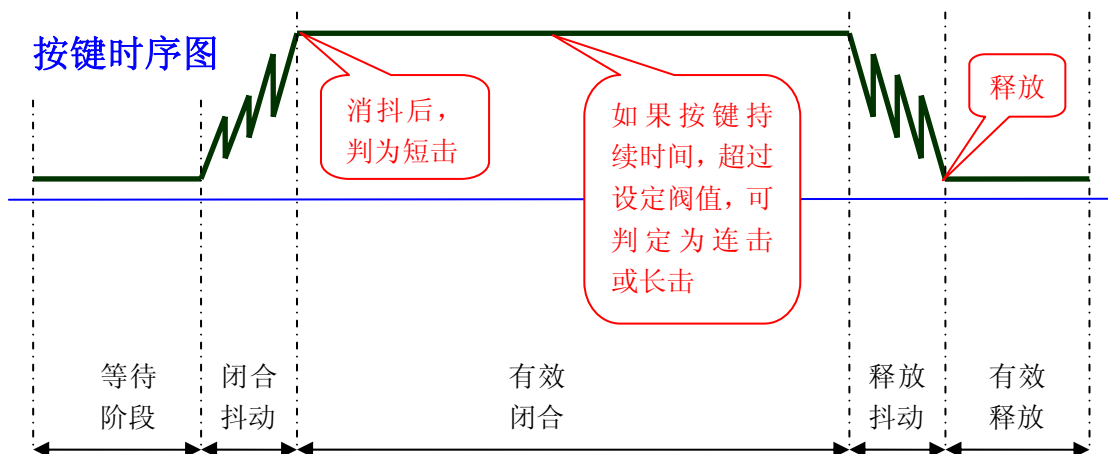
按键处理，可以说，是做单片机的朋友的一项必须掌握的基本功。在本文中，匠人将试着对按键的检测及处理做一些肤浅的分析，权当是给新手扫盲。如果您是高手，请跳过此文，谢谢！如果您自认是高手，也请跳过此文，谢谢！😊

由于文中包含部分图表，所以本文的正式发布版本为 PDF 格式。各位可以在匠人的 BLOG——《匠人的百宝箱》(<http://cxjr.21ic.org>) 中找到。

准备好了吗？那就请跟我来吧。（背景音乐响起来：我踩着不变的步伐，是为了等待你的到来……）（读者：别哼哼唧唧了，等待你个头啊！人都到齐了，快点开讲！）

2. 按键时序分析

一次完整的击键过程，包含以下几个阶段：



1、等待阶段：

此时按键尚未按下，处于空闲阶段。

2、闭合抖动阶段：

此时按键刚刚按下，但信号还处于抖动状态，系统在监测时应该有个消抖的延时。这个延时时间为 **4~20ms**。

消抖动延时的另一个作用是可以剔除信号线上的干扰，防止误动作。

3、有效闭合阶段：

此时抖动已经结束，一个有效的按键动作已经产生。系统应该在此时执行按键功能；或将按键所对应的编号（简称“键号”或“键值”）记录下来，待按键释放时再执行



更多精彩文章，尽在《匠人的百宝箱》。网址：<http://cxjr.21ic.org>

4、释放抖动阶段：

一般来说，考究一点的程序应该在这里做一次消抖延时，以防误动作。
但是，如果前面“闭合抖动阶段”的消抖延时时间取值合适的话，可以忽略此阶段。

5、有效释放阶段：

如果按键是采用释放后再执行功能，则可以在这个阶段进行相关处理。处理完成后转到阶段 1；
如果按键是采用闭合时立即执行功能，则在这个阶段可以直接切换到阶段 1。

3. 常见击键类型分析

击键类型，也就是用户击键的方式。

➤ 击键类型的划分有多种方式，比如：

- a) 按照击键时间来划分，可以分为“短击”和“长击”；
- b) 按照击键后执行的次数来划分，可以分为“单击”和“连击”。
- c) 另外还用一些组合击键方法，如“双击”或“同击”等等。

➤ 一般来说，常用的击键类型包括以下几种：

击键类型	类型说明	应用领域
单键单次短击 (简称“短击”或“单击”)	用户快速按下单个按键， 然后立即释放	基本类型，应用非常广泛，大多数地方都有用到。
单键单次长击 (简称“长击”)	用户按下按键并延时一定 时间再释放	1、用于按键的复用， 2、某些隐藏功能。 3、某些重要功能（如“总清” 键或“复位”键），为了防止用 户误操作，也会采取长击类型
单键连续按下 (简称“连击”或“连接”)	用户按下按键不放，此时 系统要按一定的时间间 隔连续响应	用于调节参数，达到连加或连减 的效果（如“UP”键和“DOWN” 键）
单键连接两次或多次 (简称“双击”或“多击”)	相当于在一定的时间间 隔内两次或多次单击	1、用于按键的复用， 2、某些隐藏功能。
双键或多键同时按下 (简称“同击”或“复合按键”)	用户同时按下两个按键， 然后再同时释放	1、用于按键的复用， 2、某些隐藏功能。
无键按下 (简称“无键”或“无击”)	当用户在一定时间内未 按任何按键时需要执行 某些特殊功能	1、设置模式的“自动退出”功 能； 2、自动进待机或睡眠模式

➤ 针对不同的击键类型，按键响应的时机也是不同的。

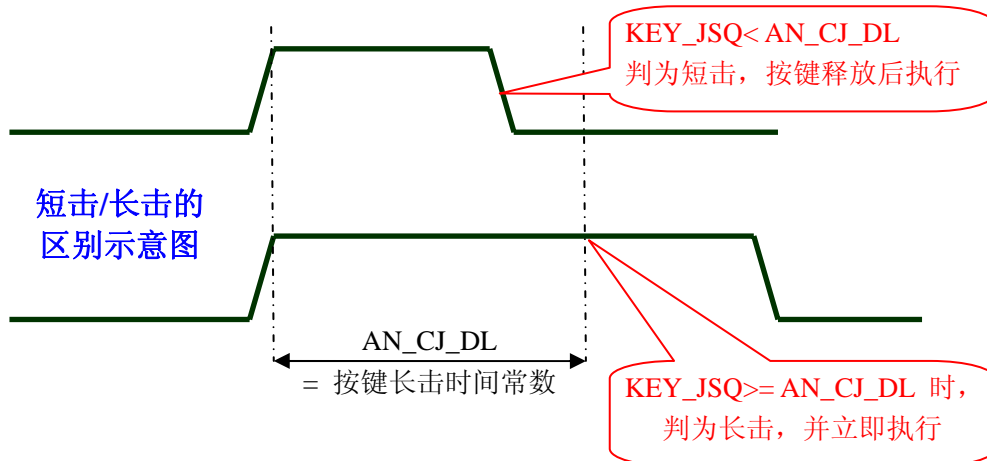
- a) 有些类型必须在按键闭合时立即响应，如：长击、连击
- b) 而有些类型则需要等到按键释放后才执行，如：当某个按键同时支持“短击”和“长击”时，必须等到按键释放，排除了本次击键是“长击”后，才能执行“短击”功能。
- c) 还有些类型必须等到按键释放后再延时一段时间，才能确认。如：

- i. 当某个按键同时支持“单击”和“双击”时，必须等到按键释放后，再延时一段时间，确信没有第二次击键动作，排除了“双击”后，才能执行“单击”功能。
- ii. 而对于“无击”类型的功能，也是要等到键盘停止触发后一段时间才能被响应。

4. 常见击键类型的判别方法

➤ 如何区别“短击”和“长击”？

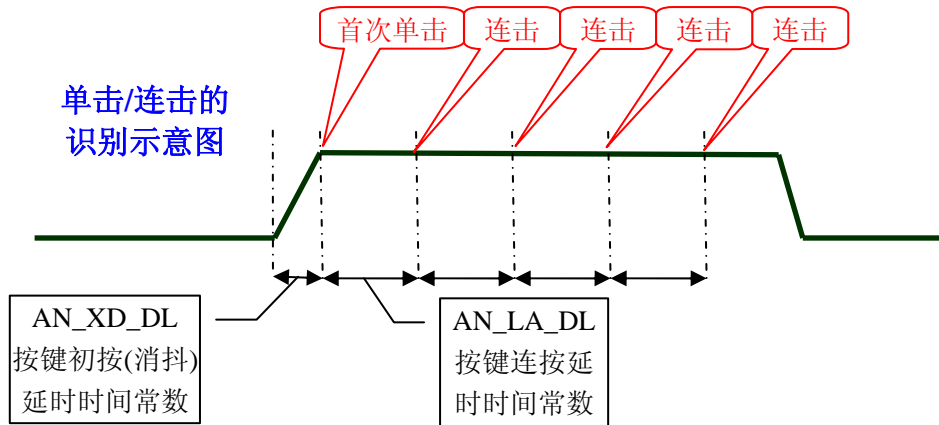
- a) 定义一个变量： KEY_JSQ = 按键闭合计数器
- b) 定义一个常数： AN_CJ_DL = 按键长击时间常数
- c) 定时检测按键，当按键闭合时， KEY_JSQ 按一定的频率递增；
- d) 当 $KEY_JSQ \geq AN_CJ_DL$ 时，确认一次有效长击。
- e) 当按键释放时，再判一次 KEY_JSQ ，如果 $KEY_JSQ < AN_CJ_DL$ ，则说明刚才释放的那次击键为“短击”
- f) 需要指出的是，当一个按键上同时支持“短击”和“长击”时，二者的执行时机是不同的。
 - i. 一般来说，“长击”一旦被检测到就立即执行。
 - ii. 而对于“短击”来说，因为当按键刚被按下时，系统无法预知本次击键的时间长度，所以，“短击”必须在释放后再执行。
- g) 当按键释放后， KEY_JSQ 应当被清零。



➤ 如何识别“单击”和“连击”？

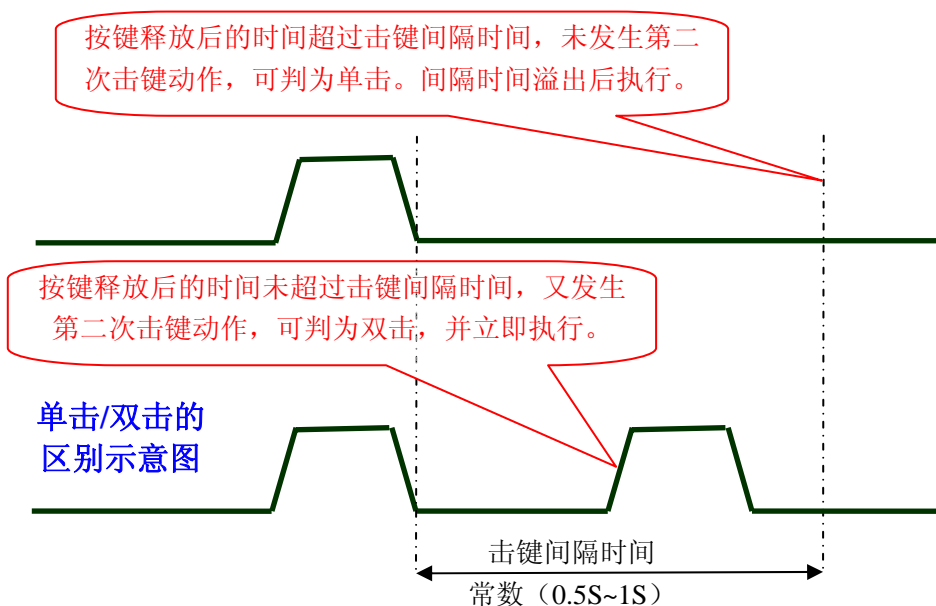
- a) 一般来说，“连击”和“单击”是相伴随的。事实上，“连击”的本质就是多次“单击”。
- b) 定义一个变量： K_DELAY = 按键响应延时时间寄存器
- c) 定义二个常数：
 - i. AN_XD_DL = 按键初按(消抖)延时（用来确定消抖时间，一般取 4~20ms）
 - ii. AN_LA_DL = 按键连按延时（用来确定连击的响应频率。比如，如果要每秒执行 10 次连击，则这个参数=100ms，）
- d) 按键未闭合前，先令 $K_DELAY = AN_XD_DL$
- e) 当按键闭合时， K_DELAY 以一定的频率-1。当 $K_DELAY = 0$ 时，即可先执行一次按键功能。（此为首次单击）。

- f) 执行完按键后令 $K_DELAY = AN_LA_DL$ ，并重新进行-1 倒计时。
- g) 当 K_DELAY 再次=0 时，即可再执行一次按键功能。（此为“连击”）。
- h) 如果按键一直闭合，就重复执行上面的 f 和 g 两个步骤，直到按键释放。



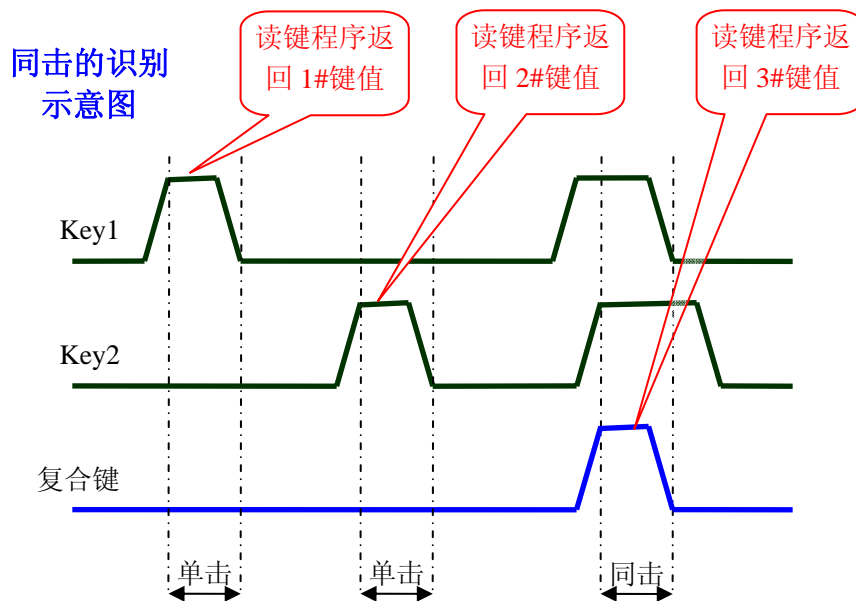
➤ 如何识别“双击”和“多击”？

- a) 识别“双击”的技巧，主要是判断两次击键之间的时间间隔。一般来说这个时间间隔定为 $0.5S \sim 1S$ 。
- b) 每次按键释放后，启动一个计数器对释放时间进行计数。如果计数时间 $>$ 击键间隔时间常数 ($0.5S \sim 1S$)，则判为“单击”。
- c) 如果在计数器还没有到达击键间隔时间常数 ($0.5S \sim 1S$)，又发生了一次击键行为，则判为“双击”。
- d) 需要强调的是：如果一个按键同时支持单击和双击功能，那么，当检测到按键被按下或释放时，不能立即响应。而是应该等待释放时间超过击键间隔时间常数 ($0.5S \sim 1S$) 后，才能判定为单击，此时才能执行单击功能。
- e) “多击”的判断技巧与“双击”类似，只需要增加一个击键次数计数器对击键进行计数即可。



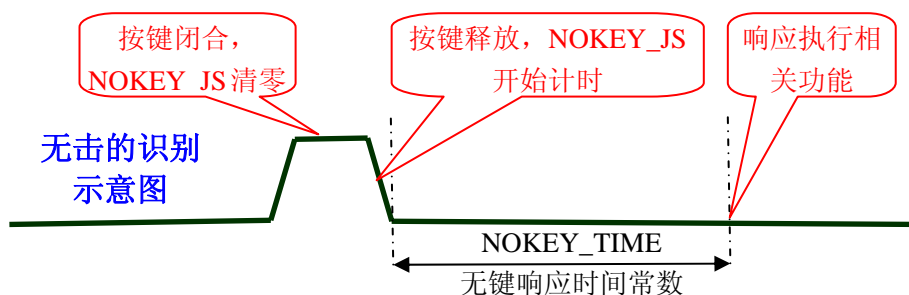
➤ 如何识别“同击”？

- “同击”是指两个或两个以上按键同时被按下时，作为一个“复合键”来单独处理。
- “同击”主要是通过按键扫描检测程序来识别。按键扫描程序（也称为“读键程序”）为每个按键分配一个键号（或称为“键值”），而“复合键”也会被赋予一个键号。比如，有两个按键，当它们分别被触发时，返回的键号分别为 1# 和 2#，当它们同时被触发时，则返回新的键号 3#。
- 在键盘处理程序中，一旦收到键号，只需按不同的键号去分别处理即可。



➤ 如何识别“无击”？

- “无击”指的是当按键连续一定时间未触发后，应该响应的功能。常见的应用如：自动退出设置状态、自动切换到待机模式，等等。
- 定义一个变量：NOKEY_JS=无键计时器。
- 定义一个常数：NOKEY_TIME=无键响应时间常数（一般为 5S 或 10S）
- 当检测到按键释放时，NOKEY_JS 每 1S 自动+1。一旦 NOKEY_JS>=NOKEY_TIME，就执行相关功能。
- 当检测到按键闭合时，NOKEY_JS 清零。



5. 参考文章：

点击以下文字连接，可以访问相应文章。

如果连接无效，可以直接到 <http://blog.21ic.com/more.asp?name=cxjr&id=4449> 找到。

或者，您也可以到《匠人的百宝箱》(<http://cxjr.21ic.org>) 搜索更多相关内容。

[藏经宝阁]《KEY(按键)应用》大全：关键字：键、KEY、

- 支持键盘双击事件的 C 程序设计 <http://blog.21ic.com/more.asp?name=cxjr&id=6920>
- 8 条口线 24 按键的 C51 源程序 <http://blog.21ic.com/more.asp?name=cxjr&id=5572>
- 96 个 key 的零延时采集 <http://blog.21ic.com/more.asp?name=cxjr&id=5574>
- 按键扫描驱动程序 <http://blog.21ic.com/more.asp?name=cxjr&id=5573>
- 键盘接口软件设计 <http://blog.21ic.com/more.asp?name=cxjr&id=5584>
- CH451 的 LED 驱动及 8*8 键盘驱动程序 <http://blog.21ic.com/more.asp?name=cxjr&id=5581>
- 8 个 IO 接 36 个按键（没有用二极管）--键扫描程序 <http://blog.21ic.com/more.asp?name=cxjr&id=4481>
- 《一个按键的多次击键组合判别技巧》大话篇 <http://blog.21ic.com/more.asp?name=cxjr&id=668>
- 《多个按键的连接处理技巧》大话篇 <http://blog.21ic.com/more.asp?name=cxjr&id=669>
- PC 机键盘按键通码与 ASCII 对照表 <http://blog.21ic.com/more.asp?name=cxjr&id=2248>
- 单片机驱动标准 PC 机键盘的 C51 程序 <http://blog.21ic.com/more.asp?name=cxjr&id=2249>
- 一个比较通用的行列键盘程序 (KEY.C) <http://blog.21ic.com/more.asp?name=cxjr&id=2466>
- 一个比较通用的行列键盘程序 (KEY.H) <http://blog.21ic.com/more.asp?name=cxjr&id=2468>
- 支持键盘双击事件的 C 程序设计 <http://blog.21ic.com/more.asp?name=cxjr&id=2481>
- Freescale H08C 键盘中断子函数(c) <http://blog.21ic.com/more.asp?name=cxjr&id=2855>
- Freescale H08C 键盘中断子函数的头文件"KB.h"(c) <http://blog.21ic.com/more.asp?name=cxjr&id=2856>
- 来自网友 xwj 的 按键程序设计方法 <http://blog.21ic.com/more.asp?name=cxjr&id=3186>
- 20 键 PS2 小键盘中断服务程序 <http://blog.21ic.com/more.asp?name=cxjr&id=4218>
- The PC's keyboard. <http://blog.21ic.com/more.asp?name=cxjr&id=4640>