

《匠人手记》之十二

单个按键多次击键的检测方法

作者：程序匠人

出处：《匠人的百宝箱》

1. 前言

匠人早年曾经写过一篇关于单个按键多次击键的检测方法。现抽空重新整理一下。（这“重新整理”的另一种说法就是“炒冷饭”，呵呵）

在某些设计中，我们可能要用一个按键来输入多种信息。如：单击/双击/三击、短击/长击、还有各种组合击键方式。为了实现这种需要，匠人设计了这个读键子程序。

本文的正式发布版本为 PDF 格式，欢迎转载。匠人唯一的要求是，转载者不可对文件中的任何内容（包括作者和出处信息）进行修改。转载者有义务保证此文档的完整性。

2. 设计思路

在这个读键子程序中，定义了三个常数。

1) 长击时间常数=500ms

用于识别短击/长击。

- 如果按键闭合时间<500ms，判断为一次短击（用“0”代表）；
- 如果按键闭合时间>500ms，判断为一次长击（用“1”代表）。

2) 两次击键时间间隔常数=700ms

用于判别击键动作是否完成。

- 两次击键时间间隔应<700ms，如果按键释放后 700ms 内无键按下，则结束读键。

3) 最多击键次数（ZHBIT）

本程序可以识别的击键次数为 1~7 次。由于每次击键都可以是短击或长击，所以最多可以识别 254 种组合。但并非每个程序中用得上这么多次击键。在大多数程序中，能判断双击即可了，这时可将程序中的 ZHBIT 常数定义为 2。同理，如果要判断 3 次按键，将 ZHBIT 常数定义为 3 即可。

- 当 ZHBIT=1 时，程序仅能识别 1 次击键，那么就有 2 种不同的组合：短击、长击
- 当 ZHBIT=2 时，程序还能识别最多 2 次击键，那么就有 6（2+4）种不同的组合：短击、长击、短击+短击、短击+长击、长击+短击、长击+长击

如果你的‘芯’是一座作坊，我愿做那不知疲倦的程序匠……

- 当 ZHBIT=3 时，程序能识别最多 3 次击键，包括 14 (2+4+8) 种组合。
- 以此类推，就可以得到以下这张表格：

| 最多击键次数 (ZHBIT) | 组合种类 |
|----------------|------------------------|
| 1 | 2 |
| 2 | 2+4=6 |
| 3 | 2+4+8=14 |
| 4 | 2+4+8+16=30 |
| 5 | 2+4+8+16+32=62 |
| 6 | 2+4+8+16+32+64=126 |
| 7 | 2+4+8+16+32+64+128=254 |

通过上面的表格，我们可以看到，这个子程序的潜力巨大，最多可以识别 254 种击键组合序列。现在，我们来对这个序列进行编码。我们可以用一个键号值来表示这些组合序列。这个键值的每一位代表一次击键，其中，0 代表短击，1 代表长击。

为了在这个键值中体现本次操作中用户实际的击键次数，我们需要设立一个引导位“1”。

也就是说，在键值中左边第一个“1”之后的每一位代表一次击键。

我们这个子程序需要做的事情，就是检测用户的击键序列，并在读键完毕返回一个键号值 KEY_NUM。其意义如下：

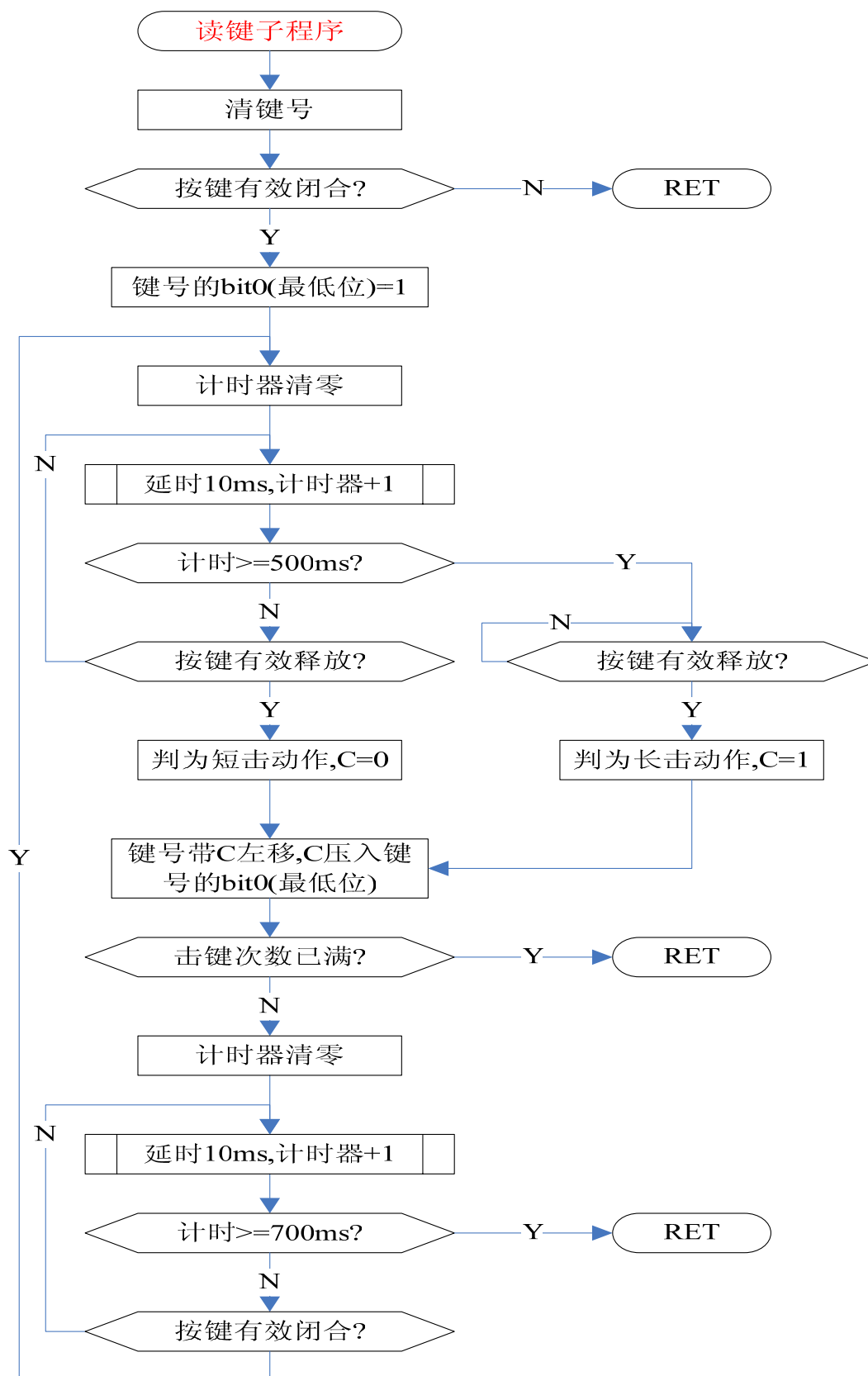
| 键值 (KEY_NUM) | 意义 |
|--------------|---------|
| 00000000 | 无键按下过 |
| 00000001 | 无意义 |
| 00000010 | 单次短击 |
| 00000011 | 单次长击 |
| 00000100 | 短击 + 短击 |
| 00000101 | 短击 + 长击 |
| 00000110 | 长击 + 短击 |
| 00000111 | 长击 + 长击 |
| | |
| 10000000 | 7 次短击 |
| 11111111 | 7 次长击 |

上表中的 KEY_NUM 值的规律是，从左向右看，第一个“1”后面的每一位（红色数字部分）代表一次击键；“0”代表短击，“1”代表长击。

掌握该规律后，我们可将任何一个 8 位的二进制数“翻译”成一种击键组合。例如：01010101，代表的是：短+长+短+长+短+长。

这是一种非常有趣的编码，通过这种方法，我们也许有一天可以做个发报机玩玩。☺

3. 流程图



4. 源程序

(用 EMC 的汇编指令实现):

```
*****
;
;读键子程序
;出口:    KEY_NUM    =键号值
;中间:    KEY_DL     =计数器
;说明:
/*
短击:键按下时间<500MS
长击:键按下时间>500MS
两次按键间隔时间<700MS
键号定义:
    KEY_NUM=00000000:    无键按下
    KEY_NUM=00000001:    无意义
    KEY_NUM=00000010:    单次短击
    KEY_NUM=00000011:    单次长击
    KEY_NUM=00000100:    短击 + 短击
    KEY_NUM=00000101:    短击 + 长击
    KEY_NUM=00000110:    长击 + 短击
    KEY_NUM=00000111:    长击 + 长击
    ...
    KEY_NUM=11111110:    长击 + 长击 + 长击 + 长击 + 长击 + 长击 + 短击
    KEY_NUM=11111111:    长击 + 长击 + 长击 + 长击 + 长击 + 长击 + 长击
*/
    ZHBIT    EQU    2    ;最多击键次数(选择范围 1~7)
*****
READKEY:
    CLR     KEY_NUM        ;清键号
    JKOFF    READKEYF      ;键未按下跳
    BS      KEY_NUM,0      ;"1" -> 键号低位
;=====
READKEYA:
    CLR     KEY_DL        ;清计数器
READKEYB:
    CALL    DL10MS
    INC     KEY_DL
    MOV     A,@50
    SUB     A,KEY_DL
    JBC     R3,C
    JMP     READKEYC      ;计数器溢出跳
    JKON    READKEYB      ;键未释放跳
    BC      R3,C          ;C=0
    JMP     READKEYD
;=====
```

如果你的‘芯’是一座作坊，我愿做那不知疲倦的程序匠……

READKEYC:

```
WDTC          ;喂狗
JKON  READKEYC ;键未释放跳
BS    R3,C     ;C=1
```

READKEYD:

```
RLLC  KEY_NUM ;键号左移一位,C -> 键号低位
JBC   KEY_NUM,ZHBIT ;按键检测未完成继续
RET
```

;=====

```
CLR  KEY_DL ;清计数器
```

READKEYE:

```
CALL  DL10MS
INC   KEY_DL
MOV   A,@70
SUB   A,KEY_DL
JBC   R3,C
```

READKEYF:

```
RET          ;计数器溢出返回
JKOFF READKEYE ;键未按下跳
JMP  READKEYA ;再次检测
```

;=====

;键闭合跳(宏)

;=====

JKON MACRO ADDRESS

```
JBS  R5,KEY ;键断开跳
FJMP ADDRESS ;键闭合跳
CALL  DL10MS ;延时去抖动
JBS  R5,KEY ;键断开跳
FJMP ADDRESS ;键闭合跳
```

ENDM

;=====

;键断开跳(宏)

;=====

JKOFF MACRO ADDRESS

```
JBC  R5,KEY ;键闭合跳
FJMP ADDRESS ;键断开跳
CALL  DL10MS ;延时去抖动
JBC  R5,KEY ;键闭合跳
FJMP ADDRESS ;键断开跳
```

ENDM

5. 参考文章

点击 [《KEY\(按键\)应用》大全](http://blog.21ic.com/more.asp?name=cxjr&id=4449) (<http://blog.21ic.com/more.asp?name=cxjr&id=4449>) 可以访问相应文章。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。

如果你的‘芯’是一座作坊，我愿做那不知疲倦的程序匠……