Mosaic Industries

Implementation of Keypad Buffer to Enter Numbers

APPLICATION
NOTE
MI-AN-001

Summary

The following program shows how to enter numbers through implementation of the keyboard buffer.

```
\ October 5,1993 Mosaic Industries Inc.
\ The information provided herein is believed to be reliable; however,
\ Mosaic Industries assumes no responsibility for inaccuracies or omissions.
\ Mosaic Industries assumes no responsibility for the use of this information
\ and all use of such information shall be entirely at the user's own risk.


DECIMAL 15 WIDTH !
3 MANTISSA.PLACES !
5 LEFT.PLACES !
3 RIGHT.PLACES !

ANEW TENKEY.ENTRY


integer: kpd.debug
FALSE to kpd.debug

VARIABLE KEYPAD.BUFFER          \ MAKE A SCRATCH PAD FOR NUMERIC INTERPRETATION
20 CONSTANT MAX#CHARS           \ 20 characters max on the LCD line
MAX#CHARS VALLOT                \ ALLOCATE 20 MORE BYTES


\ Comment - The numeric string is stored as a counted string.
\ To utilize NUMBER and FNUMBER for numeric conversion, there MUST be a space
\ at the end of the string. VARIABLE allocates 2 bytes. We then VALLOT 20 more.
\ This makes a total of 22 bytes available. One byte, the first, is used to hold
\ the count. One byte, the last, must have the previously mentioned space in it.
\ This leaves 20 bytes for characters.

DECIMAL
: get.NUMBER.from.keypad ( line \1stchar -- | line and column position of 1st
character )
KEYPAD.BUFFER 0 FALSE FALSE
LOCALS{ &RADIX.POINT &NUMBER.COMPLETE &COUNT X&ADR &firstchar &line | &char }
```

```
\ Takes input from the keypad and stores it as a counted string in KEYPAD.BUFFER
\ Does NOT check to see if the number is longer than the allocated buffer
\ The number is echoed starting at line,1stchar
\ This routine looks for the numbers 0-9, the radix point (decimal point)
\ an 'enter' button and a 'clear' button.
\ The 'enter' button signifies the number has been entered
\ The 'clear' button signifies the user has made an entry error
\ and wishes to re-enter the number.
\ The keypad is set up as follows ( the 'x' symbol means a button with
\ no significance:
\       x x 7 8 9
\       x x 4 5 6
\       x x 1 2 3
\       x C 0 . E             \ 'C' stands for clear, 'E' stands for enter
\ The keys are number 0 to 19 starting with the lower right hand corner
\ and ending in the upper left hand corner.

\ Initialization
FALSE to &NUMBER.COMPLETE
&firstchar to &char         \ load &char with the position of the 1st character
                             \ ON THE DISPLAY
\ Start looking at the keypad and process the number
BEGIN
      KEYPAD                 \ ( -- n ) Wait for a keypad button to be pressed
                             \ and return the key number


      CASE
            0 of             \ ENTER has been pressed
                  kpd.debug if cr ." Enter " endif
                  &COUNT 1+ TO &COUNT   \ INCREMENT THE STRING COUNT
                  BL                    \ ASCII CODE FOR SPACE
                  X&ADR &COUNT XN+ C!   \ STORE A SPACE AS THE LAST CHAR
                  &COUNT X&ADR C!       \ STORE THE COUNT IN THE FIRST BYTE
                                        \ OF THE STRING BUFFER
                  TRUE to &NUMBER.COMPLETE
            ENDOF
            1 of                        \ The 3 has been pressed
                  kpd.debug if  ." 3 " endif
                  " 3 " &line &char $>DISPLAY UPDATE.DISPLAY
                   &char 1+ to &char
                  &COUNT 1+ TO &COUNT   \ INCREMENT THE STRING COUNT
                  ASCII 3          \ Push the key representation onto the stack
                  X&ADR &COUNT XN+ C!   \ PUT CHARACTER INTO STRING
            ENDOF
            2 of                        \ The 6 has been pressed
                  kpd.debug if  ." 6 " endif
                  " 6 " &line &char $>DISPLAY UPDATE.DISPLAY
                   &char 1+ to &char
                  &COUNT 1+ TO &COUNT   \ INCREMENT THE STRING COUNT
                  ASCII 6         \ Push the key representation onto the stack
                  X&ADR &COUNT XN+ C!   \ PUT CHARACTER INTO STRING
            ENDOF
```

```
3 of                   \ The 9 has been pressed
    kpd.debug if  ." 9 " endif
    " 9 " &line &char $>DISPLAY UPDATE.DISPLAY
    &char 1+ to &char
    &COUNT 1+ TO &COUNT     \ INCREMENT THE STRING COUNT
    ASCII 9           \ Push the key representation onto the stack
    X&ADR &COUNT XN+ C!     \ PUT CHARACTER INTO STRING
ENDOF
4 of                   \ The decimal point has been pressed
    &RADIX.POINT NOT IF     \ ONLY ONE DECIMAL POINT PLEASE
        kpd.debug if  ." . " endif
        " . " &line &char $>DISPLAY UPDATE.DISPLAY
        &char 1+ to &char
        &COUNT 1+ TO &COUNT \ INCREMENT THE STRING COUNT
        ASCII .             \ PUSH A DECIMAL POINT ONTO THE STACK
        X&ADR &COUNT XN+ C! \ PUT CHARACTER INTO STRING
        TRUE TO &RADIX.POINT
    ENDIF

ENDOF
5 of                            \ The 2 has been pressed
    " 2 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
    kpd.debug if  ." 2 " endif
    &COUNT 1+ TO &COUNT         \ INCREMENT THE STRING COUNT
    ASCII 2                     \ PUSH A DECIMAL POINT ONTO THE STACK
    X&ADR &COUNT XN+ C!         \ PUT CHARACTER INTO STRING
ENDOF
6 of                            \ The 5 has been pressed
    kpd.debug if  ." 5 " endif
    " 5 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
    &COUNT 1+ TO &COUNT         \ INCREMENT THE STRING COUNT
    ASCII 5                     \ PUSH A DECIMAL POINT ONTO THE STACK
    X&ADR &COUNT XN+ C!         \ PUT CHARACTER INTO STRING
ENDOF
7 of                            \ The 8 has been pressed
    kpd.debug if  ." 8 " endif
    " 8 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
    &COUNT 1+ TO &COUNT         \ INCREMENT THE STRING COUNT
    ASCII 8                     \ PUSH A DECIMAL POINT ONTO THE STACK
    X&ADR &COUNT XN+ C!         \ PUT CHARACTER INTO STRING
ENDOF

8 of                            \ The 0 has been pressed
    kpd.debug if  ." 0 " endif
    " 0 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
    &COUNT 1+ TO &COUNT         \ INCREMENT THE STRING COUNT
    ASCII 0                     \ PUSH A DECIMAL POINT ONTO THE STACK
    X&ADR &COUNT XN+ C!         \ PUT CHARACTER INTO STRING
ENDOF
9 of                            \ The 1 has been pressed
    kpd.debug if  ." 1 " endif
    " 1 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
    &COUNT 1+ TO &COUNT         \ INCREMENT THE STRING COUNT
    ASCII 1                     \ PUSH A DECIMAL POINT ONTO THE STACK
    X&ADR &COUNT XN+ C!         \ PUT CHARACTER INTO STRING
ENDOF
```

```
    10 of                              \ The 4 has been pressed
        kpd.debug if  ." 4 " endif
        " 4 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
        &COUNT 1+ TO &COUNT            \ INCREMENT THE STRING COUNT
        ASCII 4                        \ PUSH A DECIMAL POINT ONTO THE STACK
        X&ADR &COUNT XN+ C!            \ PUT CHARACTER INTO STRING
    ENDOF
    11 of                              \ The 7 has been pressed
        kpd.debug if  ." 7 " endif
        " 7 " &line &char $>DISPLAY &char 1+ to &char UPDATE.DISPLAY
        &COUNT 1+ TO &COUNT            \ INCREMENT THE STRING COUNT
        ASCII 7                        \ PUSH A DECIMAL POINT ONTO THE STACK
        X&ADR &COUNT XN+ C!            \ PUT CHARACTER INTO STRING
    ENDOF

    12 of                              \ The C has been pressed
        \ the clr button clears the number for re-entry
        \ The number calc must be reset, the lcd display must be
        \ cleared of the number, and the number must be re-entered

            kpd.debug if  ." Clear " endif

        \ Clear LCD display by putting blanks in the display line
        \ Calculate the position of the 1st character in the display
        \     buffer

            DISPLAY.BUFFER &line &firstchar BUFFER.POSITION XN+

        \ Put BLANKs in the DISPLAY.BUFFER

            &char &firstchar - BLANK
            UPDATE.DISPLAY

        \ Reset display position

            &firstchar to &char

        \ RESET POSITION IN STRING BUFFER

        0 TO &COUNT

        \ RESET TEST FOR DECIMAL POINT

        FALSE TO &RADIX.POINT
    ENDOF
ENDCASE
```

```
\ If the length of the string is = the maximum # of characters,
\ Then set the count back to 1 less than the max#chars
\       &count .
      &count max#chars = IF
            max#chars 1- to &count  \ Back the count up 1
            &char 1- to &char       \ Back the LCD up 1 character
      ENDIF
      &NUMBER.COMPLETE
      UNTIL                         \ When the number is complete, bail out
      ;

: TEST.FOR.NUMBER.TYPE ( -- [R\-1] OR [N\1] OR [D\2] OR [0] )
\ Examines the counted string in keypad.buffer
\ Determines if it is a floating point number, integer, or double number
\ Leaves the numeric value under a flag on the stack
\       NUMBER TYPE                     FLAG
\       FLOAT                           -1
\       INTEGER                         +1
\       DOUBLE                          +2
\       UNABLE TO CONVERT       0

      KEYPAD.BUFFER FNUMBER \ Is the number a floating point number? [R\-1] OR 0
      DUP NOT IF                    \ The number can't be converted to a
                                    \ floating point number
      DROP          \ Drop the flag from the attempted floating point conversion
      KEYPAD.BUFFER NUMBER    \ Try an integer and double number conversion
      ENDIF
      ;

: DEMO ( -- )
\ Demonstrates the use of GET.NUMBER.FROM.KEYPAD
\ First obtain a number from the keypad
\ Then, determine the number type

\ ASSUMES KEYPAD AND DISPLAY HAVE BEEN INITIALIZED !!!!!
\ Set up the display
      clear.display
\ 01234567890123456789
      "                    " 0 0 $>DISPLAY
      "  Enter The Desired " 1 0 $>DISPLAY
      "          Quantity     " 2 0 $>DISPLAY
      "                    " 3 0 $>DISPLAY
      UPDATE.DISPLAY
      3 0 GET.NUMBER.FROM.KEYPAD
      cr ." The string input from the keypad is " keypad.buffer count.type
      cr
      TEST.FOR.NUMBER.TYPE
      CASE
            -1      OF      CR ." Floating Point : " f.     ENDOF
            0       OF  CR ." Can't be converted "  ENDOF
            1       OF      CR ." Integer " .                       ENDOF
            2       OF      CR ." Double number " d.        ENDOF
      ENDCASE
      ;

\ To run the demo, initialize the display with INIT.DISPLAY
\ Then type DEMO
```

The information provided herein is believed to be reliable; however, Mosaic Industries assumes no responsibility for inaccuracies or omissions. Mosaic Industries assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.

Mosaic Industries

5437 Central Ave Suite 1, Newark, CA 94560 Telephone: (510) 790-8222 Fax: (510) 790-0925