Keypads

HOW THEY ARE BUILT

A keypad, with 12 or 16 keys, is one of the most commonly used input devices in microprocessor applications. The telephone keypad shown below in ***Figure 1*** is a typical example. Like most such keypads, it is wired as an X-Y switch matrix as shown in ***Figure 2***. The normally-open switches connect a row to a column when pressed. Note that the resistors are not part of the keypad. Because this keypad has 12 keys, it is wired as 3 columns by 4 rows. A 16 key pad would have 4 columns by 4 rows.

Figure 1

HOW THEY ARE READ

As shown in ***Figure 2***, the columns are connected to +5 Volts (logic level 1) by pull-up resistors. The other ends of the columns are connected to an input port so that the logic level on each column can be read. The rows are connected to an output port where the software pulls one row at a time low in a repeating cycle. First row 0 is low while rows 1,2, and 3 are kept high. Then row 0 is pulled high, row 1 is pulled low, and rows 2 and 3 are kept high. And so on until each row has been pulled low, at which point the cycle repeats.

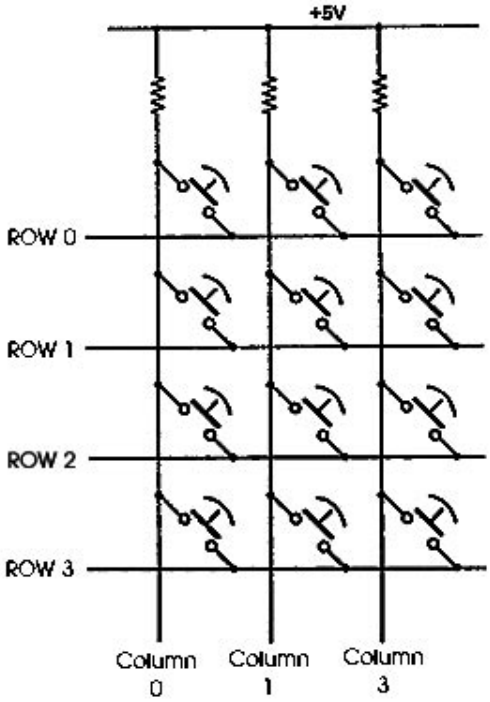

Figure 2

Each time a row is pulled low, the software will read in the columns. If no key is pressed, all the columns will be high. If a key is pressed, one column will be connected to one row. When that row is pulled low, that column will also go low. By knowing the row number and column number that are low, the software knows which key was pressed. The software runs through the scan cycle in a matter of microseconds, so no matter how fast you press the keys the software

will catch it.

Table 1 shows the pin assignments for the Electronix Express keypad (part number 1704627).

WIRING DIAGRAM FOR TELEPHONE-TYPE 3-BY-4 X - Y MATRIX KEYPAD

		COL 0	**COL 1**	**COL 2**
		PIN 4	PIN 2	PIN 6
ROW 0	PIN 3	1	2	3
ROW 1	PIN 8	4	5	6
ROW 2	PIN 7	7	8	9
ROW 3	PIN 5	*	0	#

Table 1

SOFTWARE

The following routine is written in a "generic" assembly language: it's not for any particular microprocessor. Use it for ideas to write your own code for your favorite chip. *Note that this code is for illustrative purposes only. There are no guarantees it will work in your application.*

```
LOOP:   MOV A, 0EH   ;  E = 1110
        MOV B, A     ;  SAVE A COPY
        OUT ROW_P, A ;  OUTPUT TO ROW PORT
        IN  A, COL_P ;  INPUT FROM COLUMN PORT
        AND A, 07    ;  MASK OFF UPPER BITS
        CMP A, 07    ;  ANY COLUMN LOW?
        JNZ KPRESS   ;  IF YES, THEN KEY IS PRESSED
        MOV A, 0DH   ;  D = 1101
        MOV B, A     ;
        OUT ROW_P, A ;
        IN A, COL_P  ;
        AND A, 07    ;
        CMP A, 07    ;
        JNZ KPRESS   ;
        MOV A, 0BH   ;  B = 1011
        MOV B, A     ;
        OUT ROW_P, A ;
        IN A, COL_P  ;
        AND A, 07    ;
        CMP A, 07    ;
        JNZ KPRESS   ;
        MOV A, 07H   ;  7 = 0111
        MOV B, A     ;
        OUT ROW_P, A ;
        IN A, COL_P  ;
        AND A, 07    ;
        CMP A, 07    ;
        JNZ KPRESS   ;
        JMP LOOP     ;

KPRESS: SHL A        ; SHIFT LEFT ONE BIT
        SHL A        ;
        SHL A        ;
        SHL A        ; COLUMN CODE IN UPPER 4 BITS
        OR A, B      ; COMBINE ROW and COLUMN
        MOV PTR, A   ; PUT INTO A POINTER REGISTER
        MOV A, [PTR] ; LOOK UP KEY IN A TABLE
        etc          ; WE DIDN'T SAY IT WAS EFFICIENT
```