LCD Connections to 8051

KEYBOARD INTERFACING WITH 8051

4X 4 KEYBOARD

Figure 5.4.31 shows a 4 x4 matrix connected to two ports.

- The rows are connected to an output port(Port 1) and the columns are connected toan input port. (Port 2)
- If no key has been pressed, reading the input port will yield 1s for all columns sincethey are all connected to high (Vcc).
- If all the rows are grounded and a key is pressed, one of the columns will have 0since the key pressed provides the path to ground.
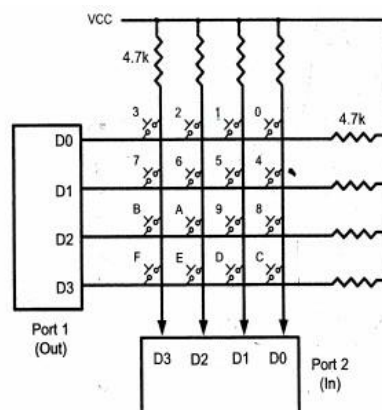- It is the function of the microcontroller to scan the keyboard continuously to detectand identify the key pressed.

Figure 5.4.3 Matrix Keyboard Connections to Ports

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi,Janice Gillispie Mazidi, Rolin McKinlay]

KEY SCAN

To find out the key pressed , the controller grounds a row by sending a '0' on the corresponding line of the output port. It then reads the data at the columns using the inputport. If data from columns is D3-D0=1111, then no key is pressed. If any bit of the columnis '0', it indicates that a key is pressed in that column. In this example, the column is identified by the following values:

1110 – key pressed in column 01101 – key pressed in column 11011 – key pressed in column 2 0111 – key pressed in column 3

STEPS TO FIND OUT KEY PRESSED

Beginning with the row 0, the microcontroller grounds it by providing a low to rowD0 only. It then reads the columns (port2). If the data read is all 1s, then no key in that row is activated and the process is moved to the next row. It then grounds the next row, reads the columns, and checks for any zero. This process continues until a row with a zero is identified. After identification of the row in which the key has been pressed, the column to which the pressed key belongs is identified as discussed above - by looking for a zero in the input values read. Example:

(a)   D3 – D0 = 1101 for the row, D3 – D0 = 1011 for the column, indicate row 1 and column 3 are selected. This indicates that key 6 is pressed.

(b)   D3 – D0 = 1011 for the row, D3 – D0 = 0111 for the column, indicate row 2 and column 3 are selected. Then key 'B' is pressed.

PROGRAM:

The program used for detection and identification of the key activated goes through the following stages:

1. To make sure that the preceding key has been released, 0s are output to all rows atonce, and the columns are read and checked repeatedly until all the columns are high.

●   When all columns are found to be high, the program waits for a short amountof time before it goes to the next stage of waiting for a key to be pressed.

2. To see if any key is pressed, the columns are scanned over and over in an infinite loopuntil one of them has a 0 on it.

- Remember that the output latch is connected to rows, still have their initialzeros (in stage 1), making them grounded.

- After the key press detection, it waits for 20-ms for the bounce and thenscans the columns again.

i) It ensures that the first key press detection was not an erroneous one due to spike noise.

ii) After the 20-ms delay, if the key is still pressed, then it goes to the loop (step 3) to detect the actual key pressed.

3. To detect which row the key pressed belongs to, it grounds one row at a time, readingthe columns each time.

- If it finds that all columns are high, this means that the key press does not belongto that row. Therefore, it grounds the next row and continues until it finds the row, that the key pressed belongs to.

- Upon finding the row that the key pressed belongs to, it sets up the starting address for the lookup table holding the scan codes for that row.

4. To identify the key pressed, it rotates the column bits, one bit at a time, into the carryflag and checks to see if it is low.

- Upon finding the zero, it pulls out the ASCII code for that key from the look-up

  table.

- Otherwise, it increments the pointer to point to the next element of the look-up

  table.

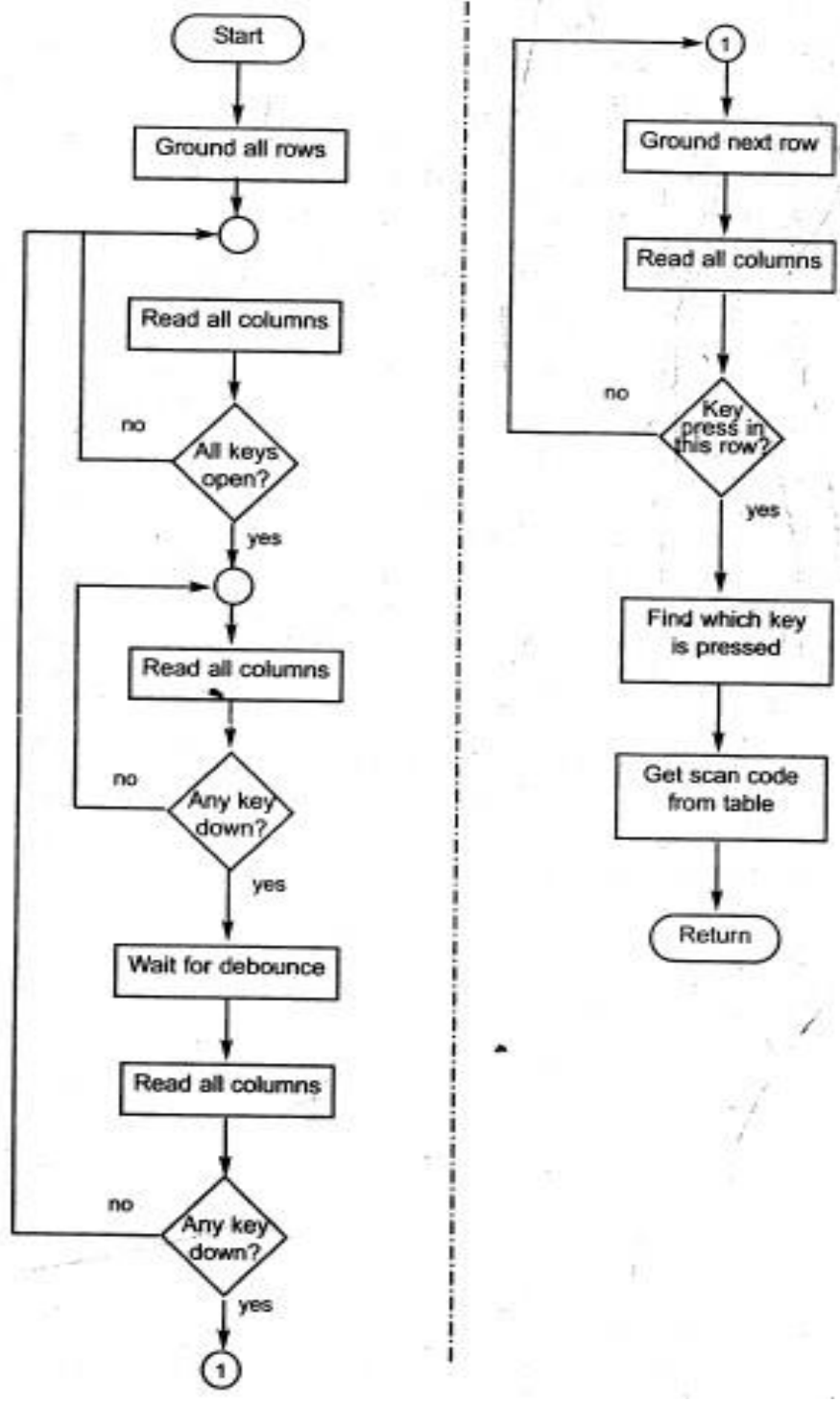Figure 5.4.4 provides the flowchart for keyboard interfacing Program for scanning and identifying the pressed key.

Figure 5.4.4 Flowchart for Keyboard Interfacing