Microcontroller Diss.

keypad

Eng. Mahmoud A. El-tatar
INTRODUCTION TO KEYPAD

- keypad is a commonly used device to get user input. Although simple push switches can be used to get user input, as we have done so, this would require 1 I/O line per switch.
INTRODUCTION TO KEYPAD CONT.

Keypads are collection of push switches however arranged in the form of a matrix. So there are rows and columns of switches. The two connections of a switch are also connected in the matrix, so that the row has common connection and column has a common connection. Thus when a button is pressed a row and a column, where the button is pressed gets connected internally. The keypads are usually available as telephone type 3 x 4 keypad. This one has three columns and 4 rows, or a 4 x 4 keypad having 4 rows and 4 columns.
CONNECTING THE KEYPAD

- First you identify the various pins of keypad as to which are rows and columns. They are usually grouped together.

- Now when the keypad has been connected it is important to detect which key is being pressed. The trick lies in scanning all the rows one by one. Since columns have pull-up resistors these pins are all at logical 1.
Detecting Key Press

- The first step is to make the row 1 line low, logical 0. Then to scan all the column lines for a logical 0. If all the column lines are high the no key in this row is being pressed. Let's say key 2 was being pressed, the column 2 pin of microcontroller would go low and other column pins would remain high. The same process is repeated for row 2 and row 3 and row 4. Every time one row is taken low and all columns are scanned. The key being pressed depends upon the column which gets low, and the row being scanned.
The mikroC PRO for PIC provides a library for working with 4x4 keypad. The library routines can also be used with 4x1, 4x2, or 4x3 keypad. For connections explanation see schematic at the bottom of this page.
Library Routines (Pro Ver. Only)

- Keypad_Init
- Keypad_Key_Press
- Keypad_Key_Click
- External dependencies of Keypad Library

The following variable must be defined in all projects using Keypad Library:

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>extern sfr char keypadPort;</td>
<td>Keypad Port. char keypadPort at PORTD;</td>
</tr>
</tbody>
</table>
LIBRARY ROUTINES (MICROC)

- Keypad_Init
- Keypad_Read
- Keypad_Released
**Keypad_Init**

- **Prototype:** `void Keypad_Init(void);`
- **Description:** Initializes port for working with keypad.
- **Example:** `Keypad_Init();`(pro ver. only)
- **Example:** `Keypad_Init(&PORTB);`
Keypad_Key_Press or Keypad_Read

- Prototype:
  - `unsigned short Keypad_Read(void);`

- Returns:
  - 1..16, depending on the key pressed, or 0 if no key is pressed.

- Description:
  Checks if any key is pressed. Function returns 1 to 16, depending on the key pressed, or 0 if no key is pressed.

- Example:
  - `kp = Keypad_Read();`
  - `kp = Keypad_Key_Press();`
Keypad_Key_Click OR Keypad_Released

- **Prototype**: 
  - `unsigned short Keypad_Read(void);`
- **Returns**: 
  - `1..16`, depending on the key pressed.
- **Description**: 
  - It is a blocking call: the function waits until some key is pressed and released. When released, the function returns `1 to 16`, depending on the key. If more than one key is pressed simultaneously the function will wait until all pressed keys are released. After that the function will return the code of the first pressed key.
- **Example**
  - `kp = Keypad_Released();`
  - `kp = Keypad_Key_Click();`
# Keypad Connection

<table>
<thead>
<tr>
<th>Pic</th>
<th></th>
<th></th>
<th>R0</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pic 84a</td>
<td></td>
<td></td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>proteus</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Keypad</td>
<td></td>
<td></td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>R4</td>
<td>10</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>R5</td>
<td>11</td>
<td>B</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>R6</td>
<td>12</td>
<td>C</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>R7</td>
<td>13</td>
<td>d</td>
<td>4</td>
<td>*</td>
<td>0</td>
<td>#</td>
</tr>
</tbody>
</table>
LAB WORK
```c
unsigned short kp, cnt, oldstate = 0;
char txt[6];

// Keypad module connections
char keypadPort at PORTB;
///// End Keypad module connections

void main() {
    cnt = 0; // Reset counter
    Keypad_Init(); // Initialize Keypad
    trisa=0x26;
    porta=4;

    do {
        kp = 0; // Reset key code variable

        // Wait for key to be pressed and released
        do
            // kp = Keypad_Key_Press(); // Store key code in kp variable
            kp = Keypad_Key_Click(); // Store key code in kp variable
        while (!kp);

        // Prepare value for output, transform key to it's ASCII value
        switch (kp) {
            //case 10: kp = 42; break; // '0' // Uncomment this block for keypad1
            //case 11: kp = 43; break; // '1'
            //case 12: kp = 45; break; // '2'
            //default: kp += 48;

            case 1: kp = 49; break; // 1 // Uncomment this block for keypad4
            case 2: kp = 50; break; // 2
            case 3: kp = 51; break; // 3
            case 4: kp = 65; break; // A
            case 5: kp = 52; break; // 4
            case 6: kp = 53; break; // 5
            case 7: kp = 54; break; // 6
            case 8: kp = 66; break; // 8
            case 9: kp = 55; break; // 7
            case 10: kp = 56; break; // 8
            case 11: kp = 57; break; // 9
            case 12: kp = 67; break; // C
            case 13: kp = 42; break; // *
            case 14: kp = 48; break; // 6
            case 15: kp = 35; break; // #
            case 16: kp = 68; break; // E
        }
    }

    if (kp==57)
    porta.f0=1;
    if (kp==56)
    porta.f0=0;

    // Print key ASCII value on LCD
    // Display counter value on LCD
    while (1);
}
```
THANK YOU