Keypad Library

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 key
see schematic at the bottom of this page.

External dependencies of Keypad Library

The following variable must be defined in all projects using Keypad Library:	Description :	Example :
`extern sfr char keypadPort;`	Keypad Port.	`char keypadPort at PORTD;`

Library Routines

- Keypad_Init
- Keypad_Key_Press
- Keypad_Key_Click

Keypad_Init

Prototype	`void Keypad_Init(void);`
Returns	Nothing.
Description	Initializes port for working with keypad.
Requires	Global variable :

	▪ keypadPort - Keypad port must be defined before using this function.

Example	`// Keypad module connections`
	`char keypadPort at PORTD;`
	`// End of keypad module connections`
	`...`
	`Keypad_Init();`

Keypad_Key_Press

Prototype	`char Keypad_Key_Press(void);`
Returns	The code of a pressed key (1..16). If no key is pressed, returns 0.
Description	Reads the key from keypad when key gets pressed.
Requires	Port needs to be initialized for working with the Keypad library, see Keypad_Init.

Example	`char kp;`
	`...`
	`kp = Keypad_Key_Press();`

Keypad_Key_Click

Prototype	`char Keypad_Key_Click(void);`
Returns	The code of a clicked key (1..16).

	If no key is clicked, returns 0.
Description	Call to `Keypad_Key_Click` is a blocking call: the function waits until some key is pressed and released. When released depending on the key. If more than one key is pressed simultaneously the function will wait until all pressed keys are will return the code of the first pressed key.
Requires	Port needs to be initialized for working with the Keypad library, see Keypad_Init.

Example	`char kp;`
	`...`
	`kp = Keypad_Key_Click();`

Library Example

This is a simple example of using the Keypad Library. It supports keypads with 1..4 rows and 1..4 columns. The code being returned by range from 1..16. In this example, the code returned is transformed into ASCII codes [0..9,A..F] and displayed on Lcd. In addition, a sr the second Lcd row number of key presses.

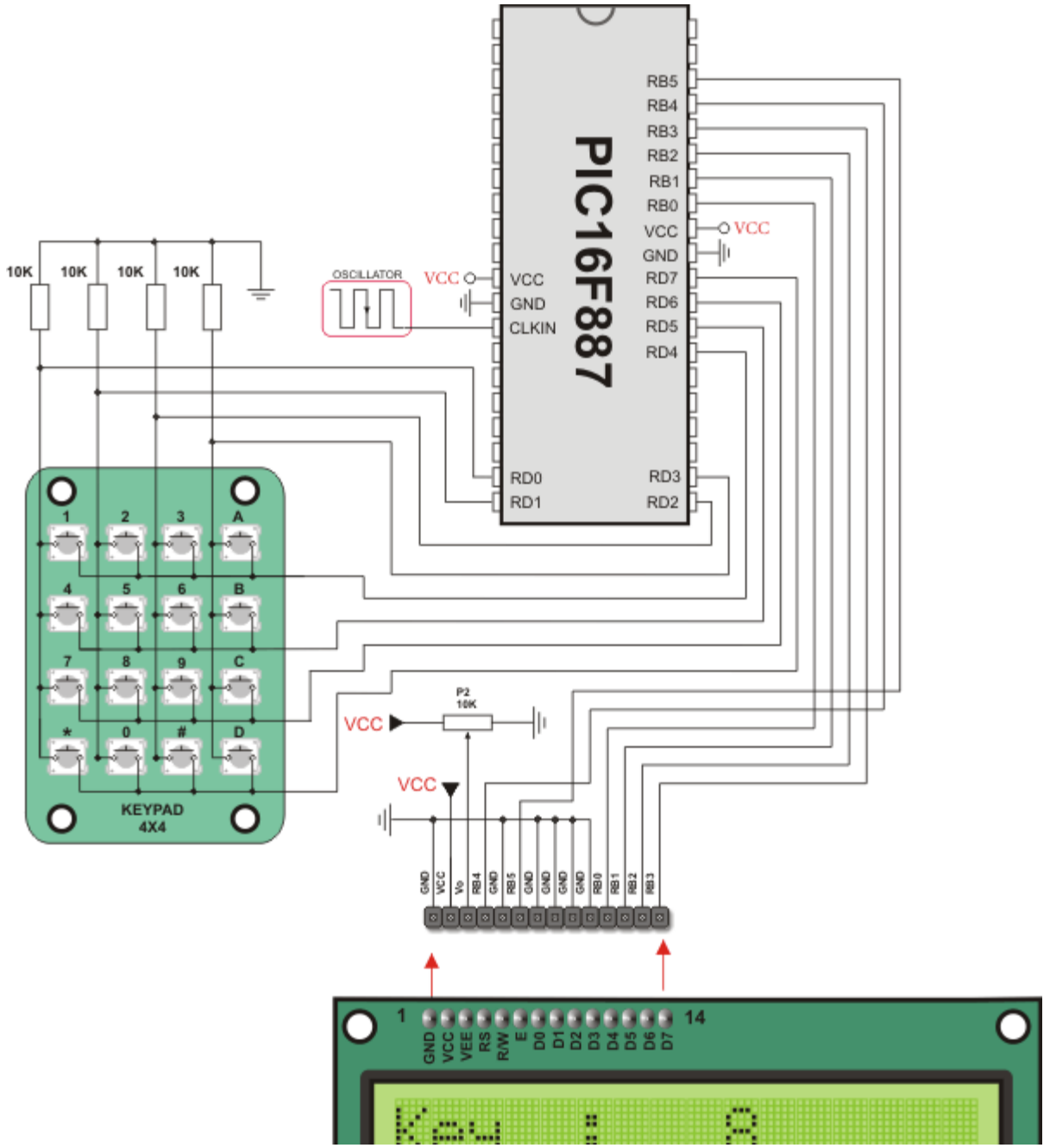
Copy Code To Clipboard

```
unsigned short kp, cnt, oldstate = 0;
char txt[6];

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

// LCD module connections
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
```

```
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections

void main() {
  cnt = 0;                               // Reset counter
  Keypad_Init();                         // Initialize Keypad
  ANSEL  = 0;                            // Configure AN pins as digital I/O
  ANSELH = 0;
  Lcd_Init();                            // Initialize LCD
  Lcd_Cmd(_LCD_CLEAR);                   // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);              // Cursor off
  Lcd_Out(1, 1, "1");
  Lcd_Out(1, 1, "Key  :");               // Write message text on LCD
  Lcd_Out(2, 1, "Times:");

  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;  // '*'   // Uncomment this block for keypad4x3
      //case 11: kp = 48; break;  // '0'
      //case 12: kp = 35; break;  // '#'
      //default: kp += 48;

      case  1: kp = 49; break; // 1        // Uncomment this block for keypad4x4
      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; // B
      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; // 0
      case 15: kp = 35; break; // #
      case 16: kp = 68; break; // D

    }

    if (kp != oldstate) {                 // Pressed key differs from previous
      cnt = 1;
      oldstate = kp;
      }
    else {                                // Pressed key is same as previous
      cnt++;
      }

    Lcd_Chr(1, 10, kp);                   // Print key ASCII value on LCD

    if (cnt == 255) {                     // If counter varialble overflow
      cnt = 0;
      Lcd_Out(2, 10, "   ");
      }

    WordToStr(cnt, txt);                  // Transform counter value to string
    Lcd_Out(2, 10, txt);                  // Display counter value on LCD
  } while (1);
}
```

HW Connection


PIC16F887
RB5
RB4
RB3
RB2
RB1
RB0
VCC
GND
RD7
RD6
RD5
RD4
RD3
RD2
RD1
RD0
CLKIN
OSCILLATOR
10K
10K
10K
10K
KEYPAD
4X4
P2
10K
1
14
GND
VCC
VEE
RS
R/W
E
D0
D1
D2
D3
D4
D5
D6
D7

4x4 Keypad connection scheme

What do you think about this topic ? Send us feedback!