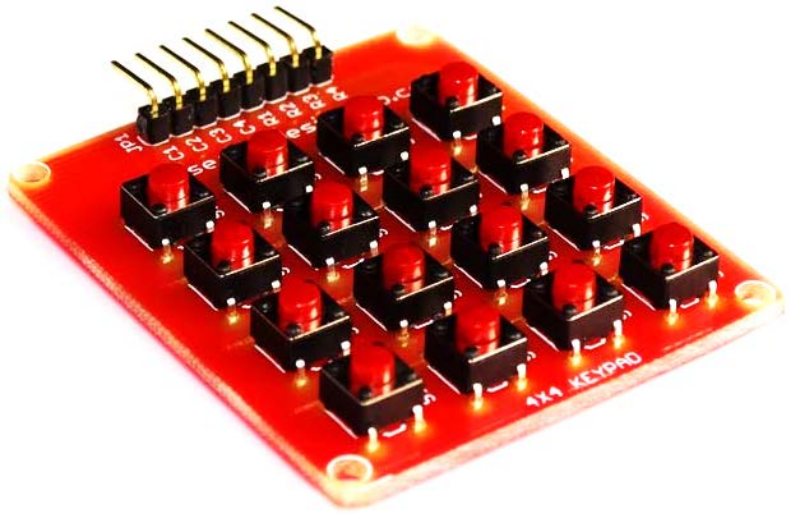
4x4 MATRIX



KEYPAD V 2.0

Table of Contents

OVERVIEW

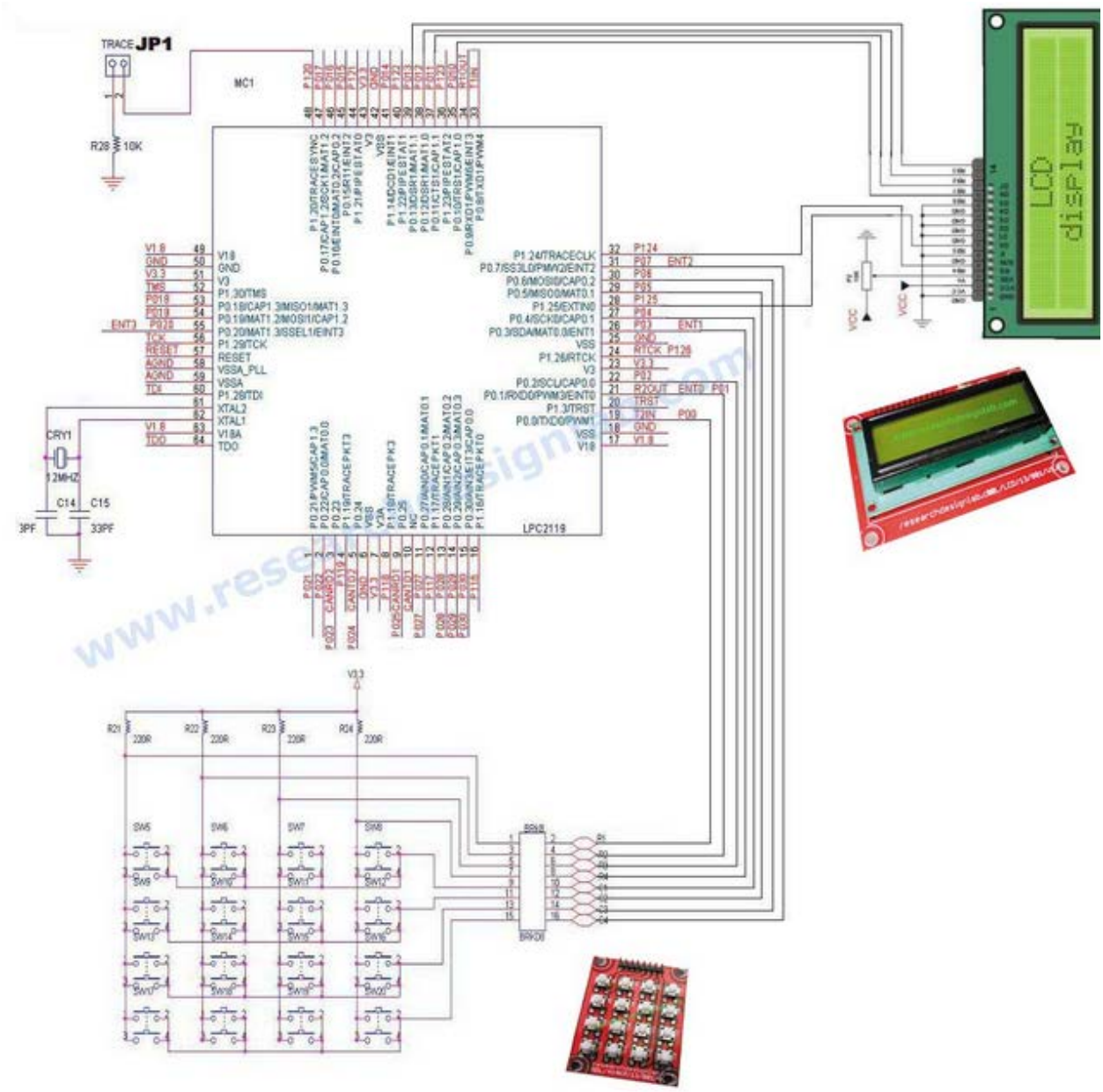
A 4x4 matrix keypad requiring eight Input/Output ports for interfacing is used as an example. Rows are connected to Peripheral Input/Output (PIO) pins configured as output. Columns are connected to PIO pins configured as input with interrupts. In this configuration, four pull-up resistors must be added in order to apply a high level on the corresponding input pins.

FEATURES

- Contact debouncing.
- Easy to interface. Board features 16 push buttons arranged as 4x4 matrix.
- Data Valid output signal for interrupt activation.
- Interfaces to any microcontroller or microprocessor.
- Cost effective for OEM applications.
- Key lifetime: 1 x 109 million operations.
- Low power consumption.
- High quality PCB FR4 Grade with FPT Certified.

CONNECTION DIAGRAM

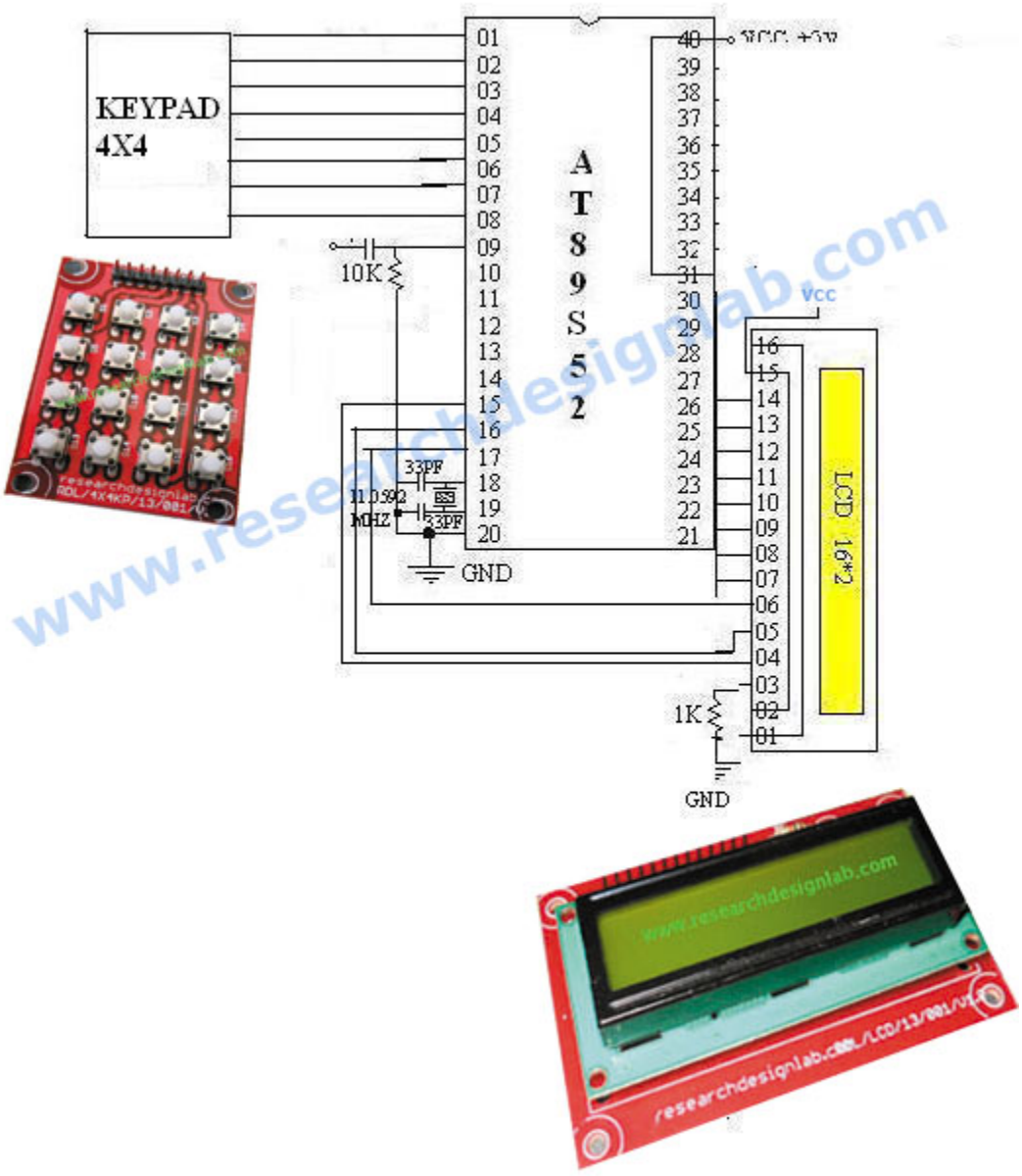
1. ARM



CODE:

http://forum.researchdesignlab.com/KEYPAD/ARM/KEPAD.txt

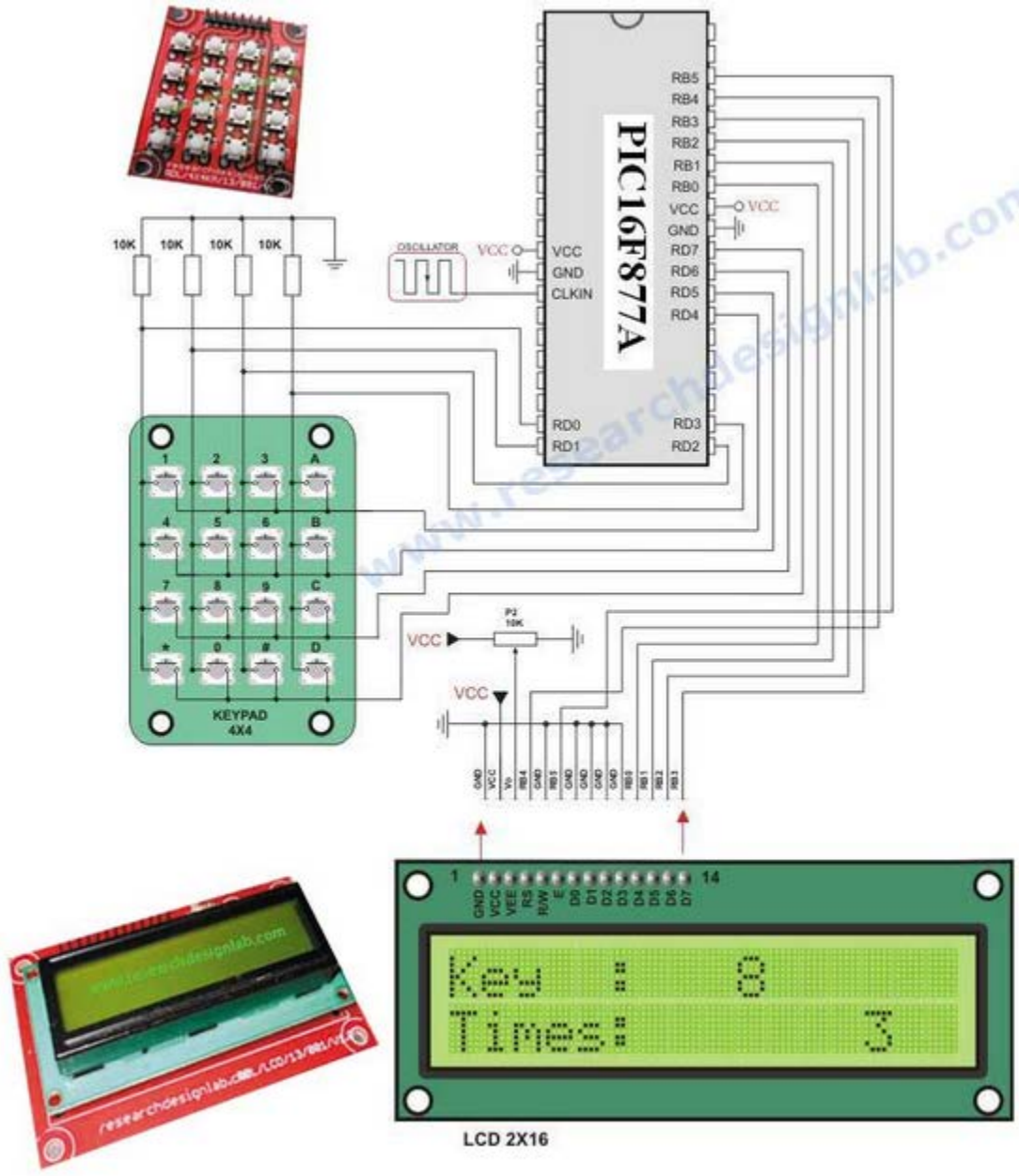
2. ATMEL



CODE:

http://forum.researchdesignlab.com/KEYPAD/ATMEL/KEYPAD.C

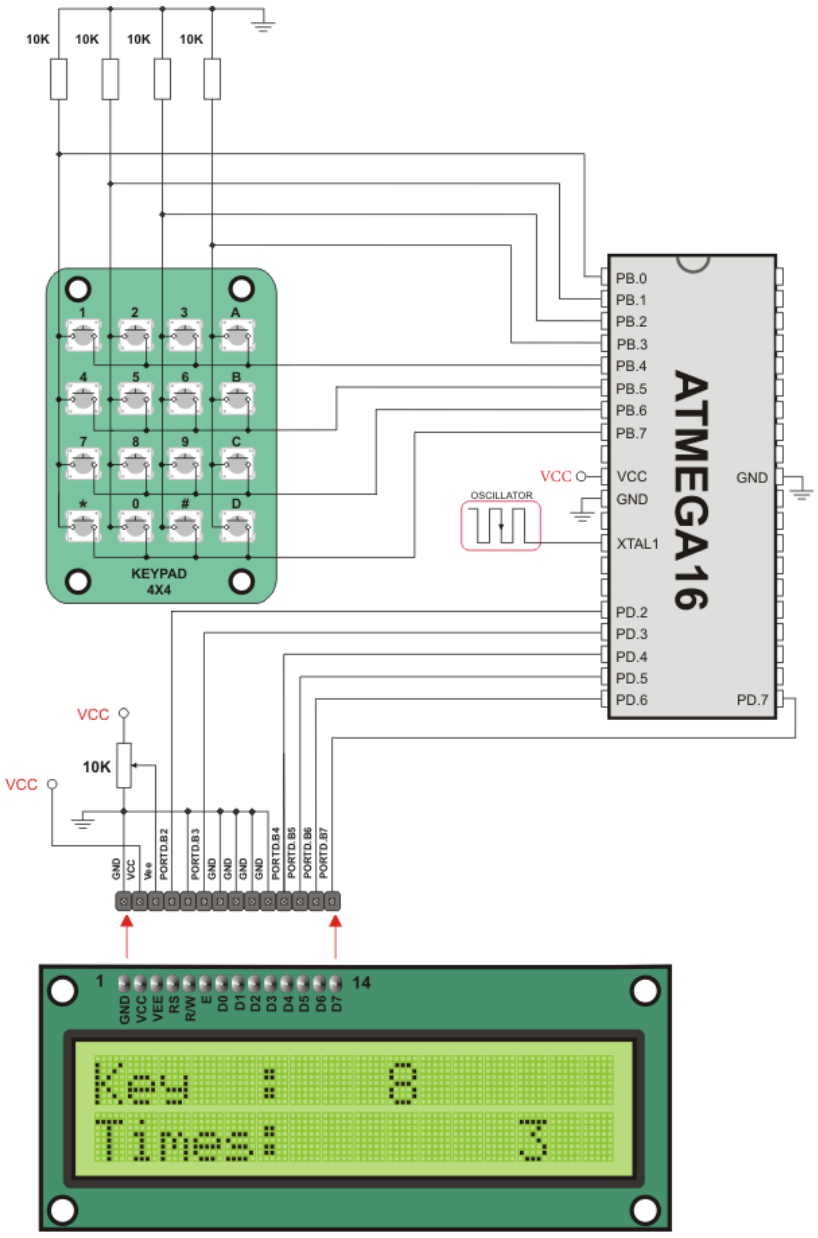
3. PIC



CODE:

http://forum.researchdesignlab.com/KEYPAD/PIC/KEYPAD_MATRIX.c

4. AVR


10K
10K
10K
10K
KEYPAD 4X4
ATMEGA16
PB.0
PB.1
PB.2
PB.3
PB.4
PB.5
PB.6
PB.7
VCC
GND
XTAL1
OSCILLATOR
PD.2
PD.3
PD.4
PD.5
PD.6
PD.7
VCC
10K
VCC
Key : 8
Times: 3

CODE:

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

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

// LCD module connections
sbit LCD_RS at PORTD2_bit;
sbit LCD_EN at PORTD3_bit;
sbit LCD_D4 at PORTD4_bit;
sbit LCD_D5 at PORTD5_bit;
sbit LCD_D6 at PORTD6_bit;
sbit LCD_D7 at PORTD7_bit;

sbit LCD_RS_Direction at DDD2_bit;
sbit LCD_EN_Direction at DDD3_bit;
sbit LCD_D4_Direction at DDD4_bit;
sbit LCD_D5_Direction at DDD5_bit;
sbit LCD_D6_Direction at DDD6_bit;
sbit LCD_D7_Direction at DDD7_bit;
// End LCD module connections

void main() {
  cnt = 0;                                // Reset counter
  Keypad_Init();                          // Initialize Keypad
  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);
}
```