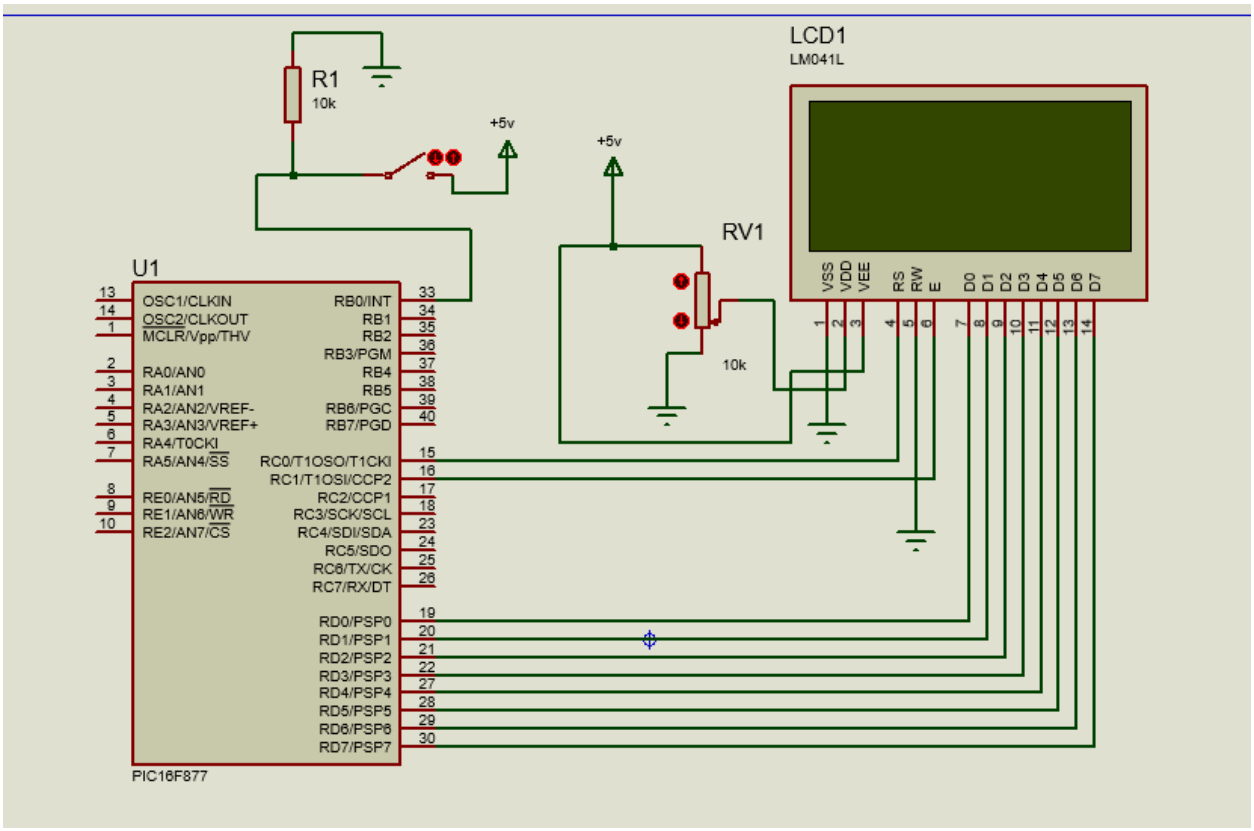
```c
#include <xc.h>

#define _XTAL_FREQ 4000000

// LCD Control Pins
#define RS RC0
#define EN RC1

// ---------- Function Prototypes ----------
void LCD_CMD(unsigned char);
void LCD_DATA(unsigned char);
void LCD_INIT(void);
void LCD_PRINT_NUMBER(unsigned int);
void LCD_DELAY(void);

// ---------- Global Counter ----------
volatile unsigned int count = 0;

// ==================== INTERRUPT SERVICE ROUTINE ====================
void __interrupt() ISR(void)
{
    // ----- Timer0 Overflow Interrupt -----
    if (T0IF)
    {
        count++;               // Increment counter
        LCD_CMD(0xC0);         // Move cursor to 2nd line
        LCD_PRINT_NUMBER(count);

        T0IF = 0;              // Clear Timer0 interrupt flag
    }

    // ----- External Interrupt on RB0/INT -----
    if (INTF)
    {
        count++;               // Increment counter
        LCD_CMD(0xC0);         // Move cursor to 2nd line
        LCD_PRINT_NUMBER(count);

        INTF = 0;              // Clear external interrupt flag
    }
}

// ========================== MAIN PROGRAM ==========================
void main(void)
{
    // -------- Setup I/O --------
    TRISD = 0x00;      // LCD Data port as output
    TRISC = 0x00;      // LCD Control pins as output
    TRISB0 = 1;        // RB0/INT as input

    // -------- Setup Timer0 --------
    OPTION_REG = 0b00000111;  // Prescaler 1:256, TMR0 uses internal clock
                              // PSA=0, PS2:PS0=111
    TMR0 = 0;                 // Initialize Timer0
    T0IE = 1;                 // Enable Timer0 interrupt

    // -------- Setup External Interrupt --------
    INTE = 1;                 // Enable RB0/INT interrupt
    INTEDG = 1;               // Interrupt on rising edge

    // -------- Enable Global Interrupts --------
    GIE = 1;

    // -------- Initialize LCD --------
```

```
    LCD_INIT();
    LCD_CMD(0xC0);              // Start cursor at beginning of 2nd line
    LCD_PRINT_NUMBER(count);

    while (1)
    {
        // Counter increments automatically via interrupts
    }
}

// ======================== LCD COMMAND FUNCTION ========================
void LCD_CMD(unsigned char cmd)
{
    RS = 0;
    PORTD = cmd;
    EN = 1;
    __delay_ms(2);
    EN = 0;
    LCD_DELAY();
}

// ========================== LCD DATA FUNCTION ==========================
void LCD_DATA(unsigned char data)
{
    RS = 1;
    PORTD = data;
    EN = 1;
    __delay_ms(2);
    EN = 0;
    LCD_DELAY();
}

// ============================ LCD INITIALIZATION ============================
void LCD_INIT(void)
{
    __delay_ms(20);

    LCD_CMD(0x38); // 8-bit mode, 2-line display
    LCD_CMD(0x0C); // Display ON, Cursor OFF, Blink OFF
    LCD_CMD(0x06); // Auto-increment cursor
    LCD_CMD(0x01); // Clear display
    __delay_ms(2);
}

// ============================ 4-DIGIT PRINT ROUTINE ============================
void LCD_PRINT_NUMBER(unsigned int num)
{
    unsigned int thousands = num / 1000;
    unsigned int hundreds  = (num / 100) % 10;
    unsigned int tens      = (num / 10) % 10;
    unsigned int ones      = num % 10;

    LCD_DATA(thousands + '0');
    LCD_DATA(hundreds  + '0');
    LCD_DATA(tens      + '0');
    LCD_DATA(ones      + '0');
}

// ============================ SMALL LCD DELAY ============================
void LCD_DELAY(void)
{
    for (unsigned int i = 0; i < 300; i++);
}
```


LCD1
LM041L
R1
10k
+5v
+5v
RV1
10k
U1
PIC16F877
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7