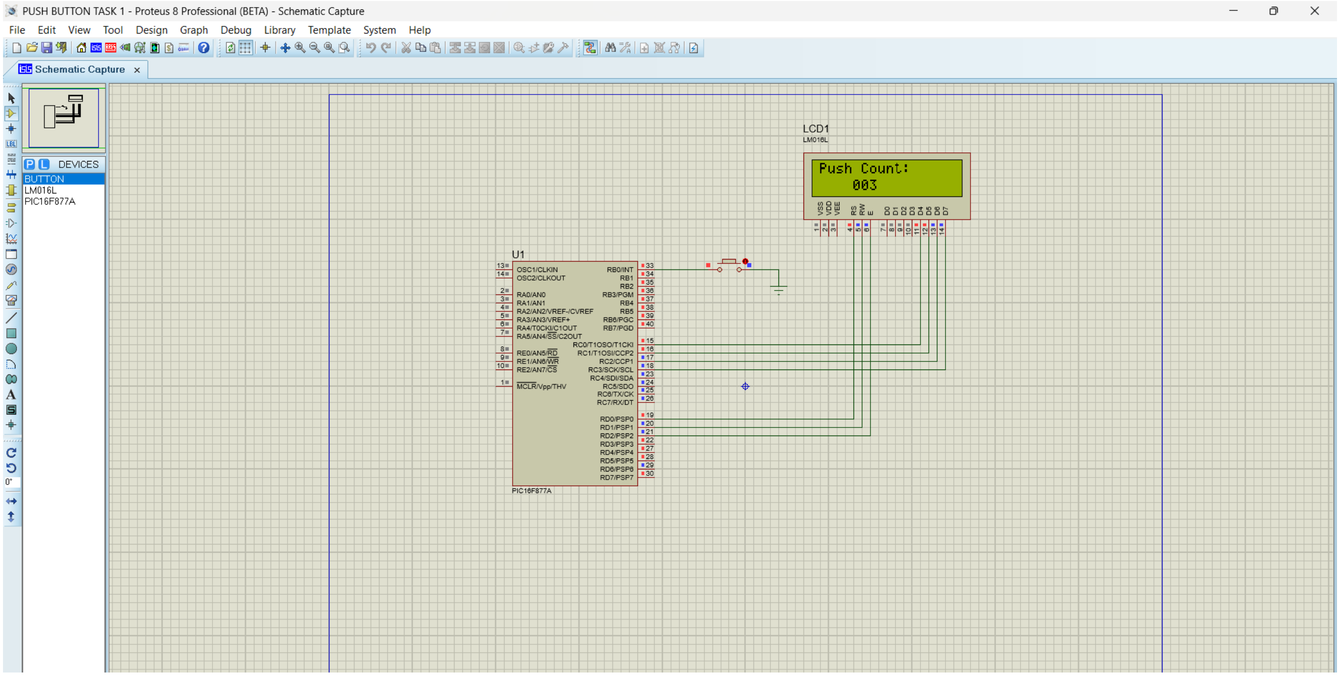
```
// PUSH BUTTON COUNTER

#include <pic.h>

// Configuration Bits
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON & BOREN_OFF & LVP_OFF);

#define _XTAL_FREQ 20000000  // Define crystal frequency

// Define LCD control pins
#define RS RD0
#define RW RD1
#define EN RD2

// Define LCD Data Pins
#define D4 RC0
#define D5 RC1
#define D6 RC2
#define D7 RC3

// Define Push Button
#define BUTTON RB0

void lcd_nibble_write(unsigned char data, unsigned char control) {
  PORTC = (PORTC & 0xF0) | (data & 0x0F); // Send lower nibble
  RS = control;
  RW = 0;
  EN = 1;
  __delay_ms(2);
  EN = 0;
}


void lcd_command(unsigned char cmd) {
  lcd_nibble_write(cmd >> 4, 0);
  lcd_nibble_write(cmd, 0);
}


void lcd_data(unsigned char data) {
  lcd_nibble_write(data >> 4, 1);
  lcd_nibble_write(data, 1);
}


void lcd_string(const char *s) {
  while (*s) {
      lcd_data(*s++);
  }
}


void lcd_init() {
  TRISC = 0x00; // LCD Data Port as Output
  TRISD = 0x00; // LCD Control Port as Output
  __delay_ms(20);
  lcd_nibble_write(0x03, 0);
  __delay_ms(5);
  lcd_nibble_write(0x03, 0);
  __delay_ms(1);
  lcd_nibble_write(0x03, 0);
  lcd_nibble_write(0x02, 0); // 4-bit mode
  lcd_command(0x28); // 4-bit, 2-line, 5x7 dots
  lcd_command(0x0C); // Display ON, Cursor OFF
  lcd_command(0x06); // Entry mode, auto increment
  lcd_command(0x01); // Clear display
  __delay_ms(2);
}


void lcd_set_cursor(unsigned char row, unsigned char col) {
  unsigned char pos;
  if (row == 1) pos = 0x80 + col - 1;
  else if (row == 2) pos = 0xC0 + col - 1;
  lcd_command(pos);
}


void debounceButton() {
  __delay_ms(50);  // Simple debounce delay
}


void main() {
  int count = 0;
  char countStr[4];  // String to hold count value


  TRISB0 = 1;   // RB0 (Button) as Input
  OPTION_REG &= ~(1 << 7); // Enable internal pull-ups
  lcd_init();
  lcd_set_cursor(1, 1);
  lcd_string("Push Count:");

  while (1) {
      if (BUTTON == 0) { // If button is pressed
          debounceButton();
          if (BUTTON == 0) {
              count++;  // Increment count

              // Convert count to string
              countStr[0] = (count / 100) + '0';
              countStr[1] = ((count / 10) % 10) + '0';
              countStr[2] = (count % 10) + '0';
              countStr[3] = '\0';

              // Display count on LCD
              lcd_set_cursor(2, 5);
              lcd_string(countStr);

              while (BUTTON == 0); // Wait for button release
          }
      }
  }
}
```



```
Push Count:
    003
```