## LCD 16X2 8-BIT MODE DISPLAY USING PIC16F877A

Program 1:



#pragma config FOSC = HS // Oscillator Selection bits

#pragma config WDTE = OFF // Watchdog Timer Enable bit

#pragma config PWRTE = OFF // Power-up Timer Enable bit

#pragma config CP = OFF // Flash Program Memory Code Protection bit

#pragma config BOREN = ON // Brown-out Reset Enable bit

#pragma config LVP = OFF // Low-Voltage ICSP Enable bit

#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit

#pragma config WRT = OFF // Flash Program Memory Write Enable bits

#pragma config DEBUG = OFF // In-Circuit Debugger Mode bit

#include <xc.h>

#define \_XTAL\_FREQ 20000000 // Define clock frequency for delay

#define rs RD6

#define en RD7

void lcd\_init();

void cmd(unsigned char a);

void dat(unsigned char b);

void show(unsigned char \*s);

void lcd\_init() {

cmd(0x38); // 8-bit, 2 line, 5x7 dots

cmd(0x0c); // Display on, cursor off

cmd(0x06); // Increment cursor

cmd(0x01); // Clear display

\_\_delay\_ms(2); // Wait for LCD to clear

}

void main() {

TRISB = 0x00; // Set PORTB as output

TRISDbits.TRISD6 = 0; // Set RD6 as output

TRISDbits.TRISD7 = 0; // Set RD7 as output

lcd\_init();

while(1){

cmd(0x80); // Set cursor position

show((unsigned char \*)"Sanjay Suresh");

cmd(0xc0);

show((unsigned char \*)"22EC100");

}

}

void cmd(unsigned char a) {

PORTB = a;

rs = 0;

en = 1;

\_\_delay\_ms(2); // Delay for command to process

en = 0;

}

void dat(unsigned char b) {

PORTB = b;

rs = 1;

en = 1;

\_\_delay\_ms(2); // Delay for data to process

en = 0;

}

void show(unsigned char \*s) {

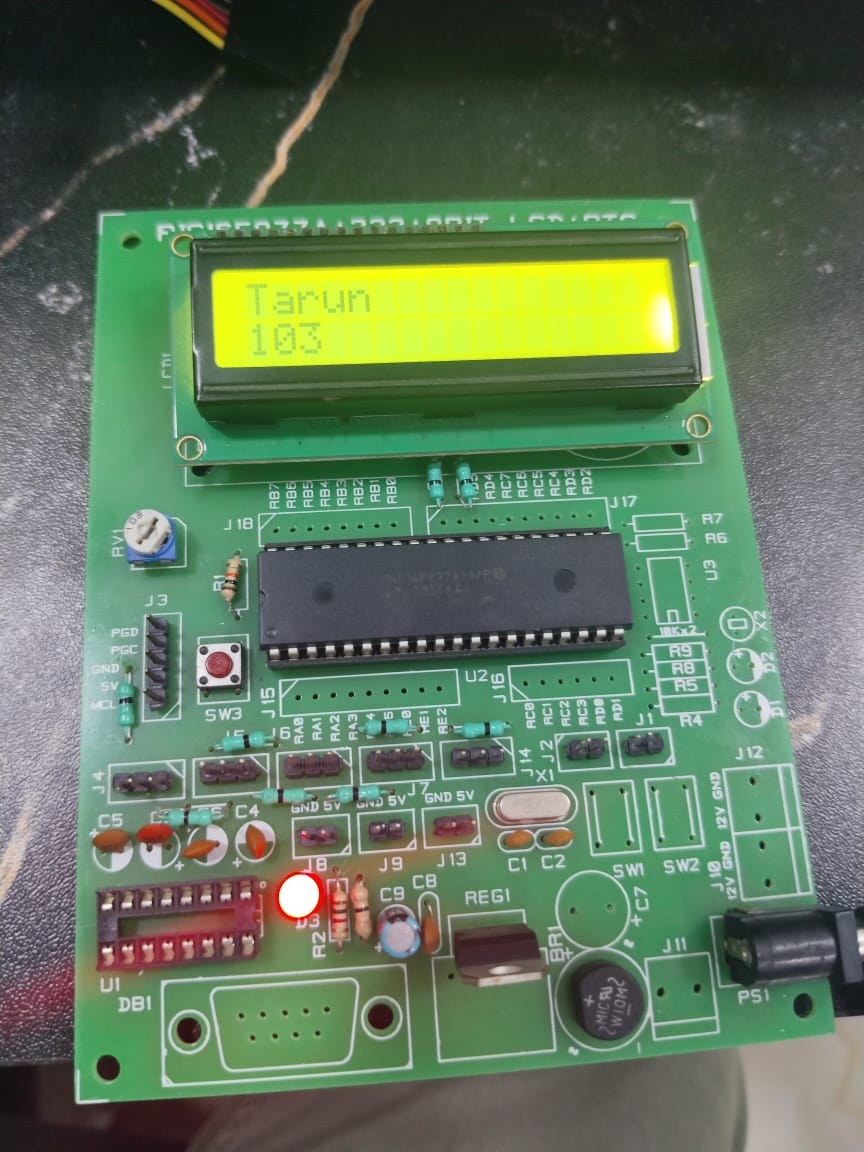
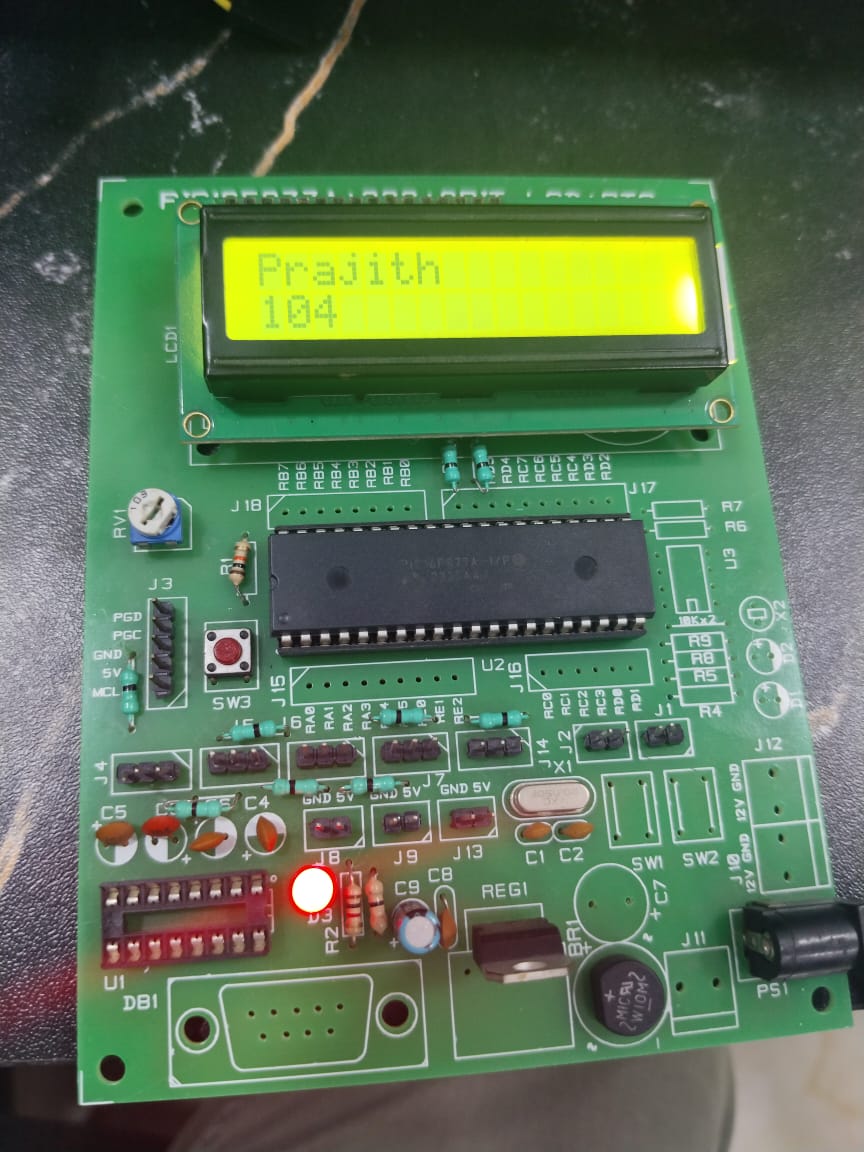
while(\*s) {

dat(\*s++);

}

}

Program 2:



#pragma config FOSC = HS // Oscillator Selection bits

#pragma config WDTE = OFF // Watchdog Timer Enable bit

#pragma config PWRTE = OFF // Power-up Timer Enable bit

#pragma config CP = OFF // Flash Program Memory Code Protection bit

#pragma config BOREN = ON // Brown-out Reset Enable bit

#pragma config LVP = OFF // Low-Voltage ICSP Enable bit

#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit

#pragma config WRT = OFF // Flash Program Memory Write Enable bits

#pragma config DEBUG = OFF // In-Circuit Debugger Mode bit

#include <xc.h>

#include <stdio.h>

#define \_XTAL\_FREQ 20000000 // Define clock frequency for delay

#define rs RD6

#define en RD7

void lcd\_init();

void cmd(unsigned char a);

void dat(unsigned char b);

void show(unsigned char \*s);

void lcd\_init() {

cmd(0x38); // 8-bit, 2 line, 5x7 dots

cmd(0x0c); // Display on, cursor off

cmd(0x06); // Increment cursor

cmd(0x01); // Clear display

\_\_delay\_ms(2); // Wait for LCD to clear

}

void main() {

TRISB = 0x00; // Set PORTB as output

TRISDbits.TRISD6 = 0; // Set RD6 as output

TRISDbits.TRISD7 = 0; // Set RD7 as output

char \*arr1[]={"Sanjay","Rahul","Sri","Tarun","Prajith"};

int r = 100; // Integer to be displayed

char buffer[10];

lcd\_init();

while(1){

int r = 100; // Integer to be displayed

for(int i=0;i<5;i++){

\_\_delay\_ms(1000);

cmd(0x80); // Set cursor position

show((unsigned char \*) arr1[i]);

sprintf(buffer, "%d", r); // Convert integer r to string

cmd(0xc0); // Move to the second line

show((unsigned char \*)buffer); // Display the string version of r

r += 1; // Increment r

\_\_delay\_ms(1000);

cmd(0x01);

}

}

}

void cmd(unsigned char a) {

PORTB = a;

rs = 0;

en = 1;

\_\_delay\_ms(2); // Delay for command to process

en = 0;

}

void dat(unsigned char b) {

PORTB = b;

rs = 1;

en = 1;

\_\_delay\_ms(2); // Delay for data to process

en = 0;

}

void show(unsigned char \*s) {

while(\*s) {

dat(\*s++);

}

}