Luca Criconia

BOARD LCD HD44780

1. INTRODUZIONE

La board mostra un testo di massimo 24 caratteri disposto su due colonne mediante il display lcd HD44780 e viene utilizzata insieme a un microcontrollore che ne comanda l'accensione. Per funzionare la board ha bisogno di essere collegata correttamente alla scheda di sviluppo. Nel caso dell'utilizzo della scheda DAM il **collegamento** avviene tramite un connettore pin to pin a 10 poli a una qualsiasi porta da 8 bit del microcontrollore.
La board non è stata progettata per funzionare solo con un particolare microcontrollore, ma effettuando correttamente i collegamenti descritti prima si può utilizzare un microcontrollore qualsiasi.

2. SCHEMA ELETTRICO

La board LCD è composta dai seguenti **componenti**:

- 1 LCD HD44780;
- 1 resistenza da 10Ω (limitazione corrente retroilluminazione);
- 1 trimmer da 10KΩ (regolazione contrasto).

Presenta 8 **pin di ingresso** di tipo strip-line:

- 4 pin per il bus dati;
- 2 pin di controllo del display;
- 2 pin di alimentazione (VDD, GND).

Lo schematico della basetta è mostrato in Figura 1.

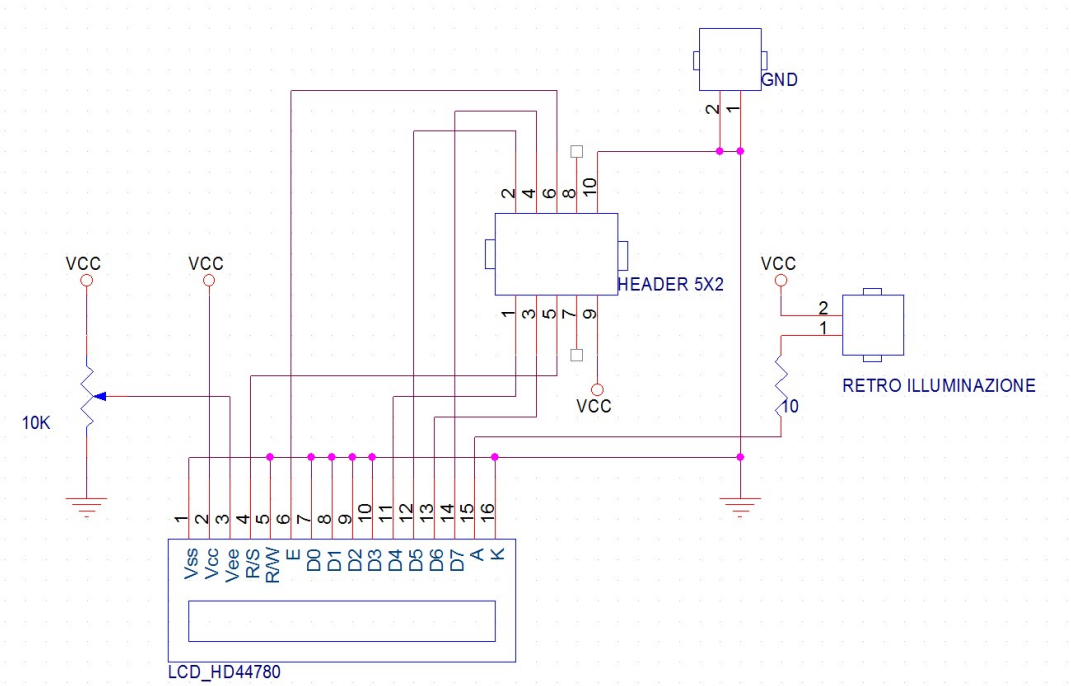

Figura 1. Schematic

Nota:
Nello schema elettrico è presente un jumper denominato: “retro illuminazione”. Esso, se cortocircuitato, serve per poter accendere la retroilluminazione.

3. LAYOUT

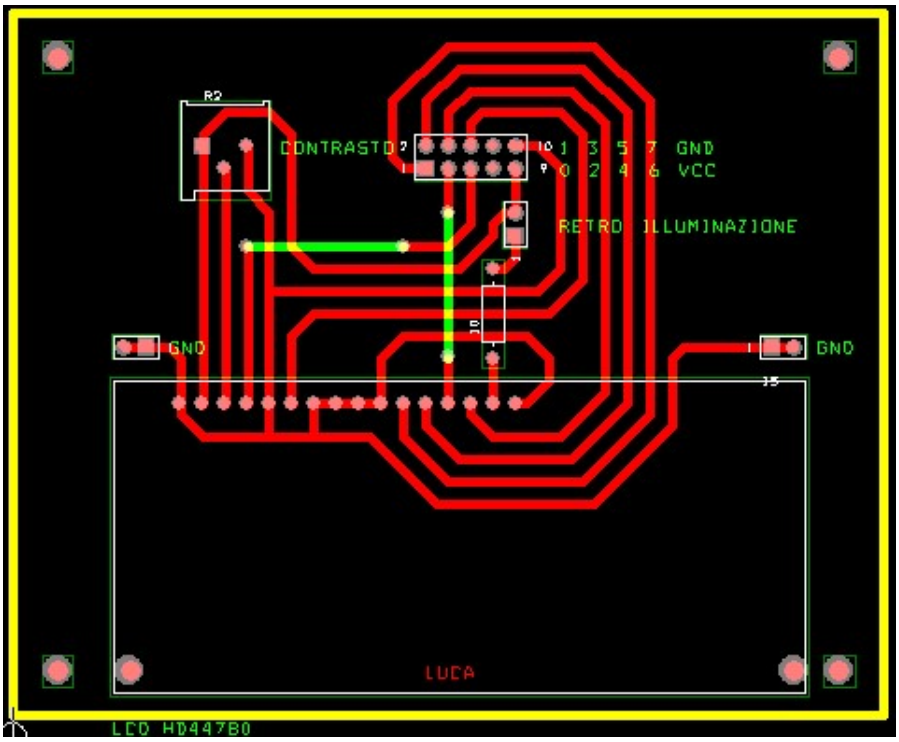

Figura 2. Layout (senza piani di massa): piste e componenti, Top (verde), Bottom (rosso)

La board è dual-layer: i componenti si trovano tutti sul top layer della board, mentre le piste sono state distribuite su entrambe le facciate per ridurre le dimensioni della scheda e cercare di semplificare i collegamenti.
Su tutta la board il **piano di massa** si trova alla tensione di riferimento GND, ma nel caso di un collegamento a coccodrillo è necessario prestare attenzione ad eventuali piste poste sul bottom layer della basetta; per risolvere questo problema sono stati posizionati, sopra al display a sinistra e a destra, due pin collegati alla tensione di massa, a cui è possibile collegarsi.
I **connettori** strip-line di tipo maschio sono divisi in due gruppi: il gruppo in basso (16 pin) serve per la connessione del display, mentre il gruppo di sopra(10 pin)serve per la connessione alla scheda di sviluppo; sulla board è presente la legenda della piedinatura corretta.

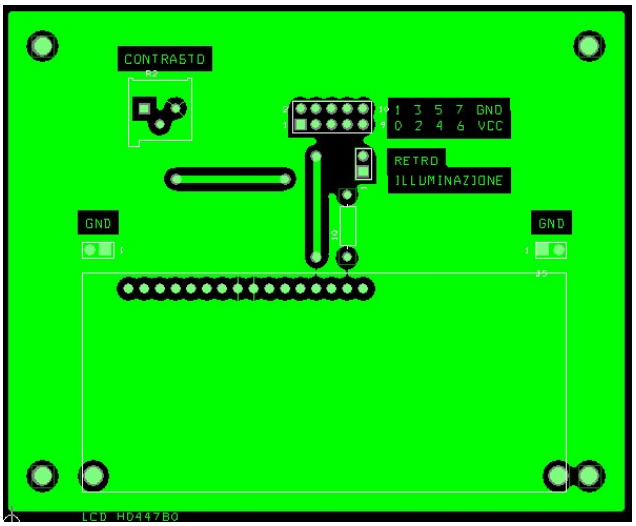

Figura 3. Top Layer

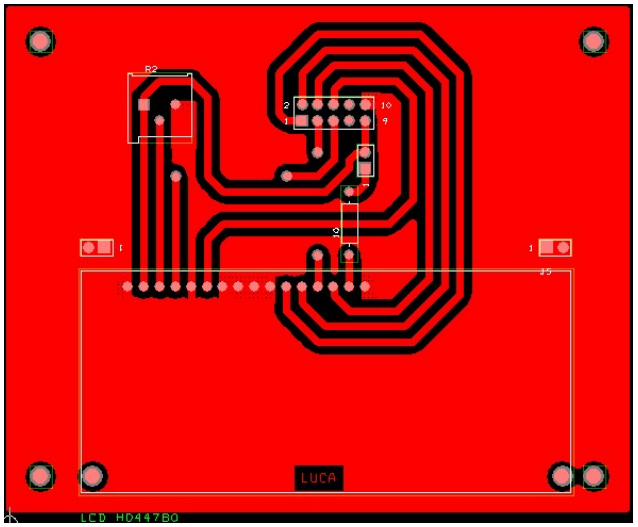

Figura 4. Bottom Layer (Mirrored)

4. PROGRAMMA DI TEST

Il programma è scritto per il compilatore c18 e funziona correttamente per un pic con quarzo a 8MHz. Se eventualmente la frequenza di clock è diversa da quella indicata bisogna ricalcolare i delay della libreria in accordo con i tempi descritti nel datasheet del display. Il codice si compone di un semplice main che fa visualizzare sul display (connesso alla porta D) un conteggio da 0 a 99 sfruttando una libreria non standard di nome LCD_44780_mod.h.

```
#include <stdio.h>
#include <delays.h>
#include <p18f4520.h>

#include "LCD_44780_mod.h"
#include "datatypes.h"

#pragma config OSC = HS   //High Speed Crystal Resonator
#pragma config WDT = OFF  //Watchdog Timer (RC interno)
#pragma config LVP = OFF  //Low Voltage ICSP Programming

void main (void);

void main()
{
	TRISBbits.TRISD0 = 0;
	TRISBbits.TRISD1 = 0;
	TRISBbits.TRISD2 = 0;
	TRISBbits.TRISD3 = 0;
	TRISBbits.TRISD4 = 0;
	TRISBbits.TRISD5 = 0;
	ADCON1bits.PCFG3 = 1;
	ADCON1bits.PCFG2 = 1;
	ADCON1bits.PCFG1 = 1;
	ADCON1bits.PCFG0 = 1;

	OpenLCD();

	InizializzaDisplay();

	while (1)
	{
		unsigned int i;
		char conta[2];
		char contaescrivi[16];

		ClearLCD();
		for (i = 0; i < 100 ; i++)
		{
			HomeLCD();
			sprintf (conta, "%4i", i);
			WriteVarLCD (conta);
			Line2LCD();
			sprintf (contaescrivi, "%4i  aspetta...", i);
			WriteVarLCD (contaescrivi);
		}
	}
}
```

LIBRERIA:

```
#ifndef FLAG_LCD_44780
#define FLAG_LCD_44780

// LCD constants

#define LCD_D0 PORTBbits.RD0// you must set this pin as output
#define LCD_D1 PORTBbits.RD1// you must  set this pin as output
#define LCD_D2 PORTBbits.RD2// you must  set this pin as output
#define LCD_D3 PORTBbits.RD3// you must  set this pin as output
#define LCD_RS PORTBbits.RD4 // you must  set this pin as output
#define LCD_E  PORTBbits.RD5  // you must  set this pin as output

//Other Constants

#define LEFT 0
#define RIGHT 1

/*******************************************************
 Available Functions
*******************************************************/

void   OpenLCD (void);
void   ClearLCD (void);
void   CursorLCD (char,char);        //      1=ONcursor 0=OFF cursor;  1=ON blinking 0=OFF Blinking
void   WriteCharLCD (char);
void   HomeLCD (void);
void   Line2LCD (void);
void   ShiftLCD (char);                        //      0=left    1=right
void   ShiftCursorLCD (char);        //      0=left    1=right
void   WriteStringLCD ();
void   WriteVarLCD(char *buffer);


//*********************************************************************************************
//this function accept in input the number of us for the dalay
//the number of nop inside the loop are trimmed for 20MHz Quartz  //modificato!!! da controllare!!!

void delayLCD (int attesa)
{          int i;
          for (i =0; i<attesa;i++)
                    {
                              _asm
                              nop
                              nop
                              nop
                              nop
                              nop
                              nop
                              _endasm
                    }
}

//*********************************************************************************************
```

```
//This function create a pulse on the Pin Enable of the LCD to allow the sending of the command

void Epulse (void)
{
	LCD_E = 1;
	delayLCD (350);
	LCD_E = 0;
	delayLCD (100);
}

//*****************************************************************************************
//Send a command. The input are the 8 command bits splitted in two 4 bit command
//You have to send the 4 high bit and than the 4 low bit

void SendCommand (unsigned char D3, unsigned char D2, unsigned char D1, unsigned char D0)
{
	LCD_D0 = D0;
	LCD_D1 = D1;
	LCD_D2 = D2;
	LCD_D3 = D3;
	Epulse ();
	delayLCD (20);
}

//*****************************************************************************************

void PosLCD(char row, char pos)
{
	unsigned char D3,D2,D1,D0;
	unsigned char Element;

	row = row << 6; // la seconda riga comincia da 64
	Element = (0b10000000) + row + pos;

		D3 = (Element & 0b10000000) >> 7;	// Splitting of the first nibble
		D2 = (Element & 0b01000000) >> 6;
		D1 = (Element & 0b00100000) >> 5;
		D0 = (Element & 0b00010000) >> 4;

		SendCommand (D3,D2,D1,D0);

		D3 = (Element & 0b00001000) >> 3;	// Splitting of the second nibble
		D2 = (Element & 0b00000100) >> 2;
		D1 = (Element & 0b00000010) >> 1;
		D0 = (Element & 0b00000001);

		SendCommand (D3,D2,D1,D0);

}

//*****************************************************************************************

void Line2LCD()
{
	SendCommand (1,1,0,0);		// the cursor is moved to the second line
```

```
	SendCommand (0,0,0,0);
}

//*****************************************************************************************

void HomeLCD()
{
	SendCommand (0,0,0,0);		// the cursor is moved at the beginning of the LCD
	SendCommand (0,0,1,0);
}

//*****************************************************************************************

void ShiftLCD(char L_R)
{
	SendCommand (0,0,0,1);		// the LCD is shifted on the left or on the right
	SendCommand (1,L_R,0,0);
}

//*****************************************************************************************

void ShiftCursorLCD(char L_R)
{
	SendCommand (0,0,0,1);		// the cursor is shifted on the left or on the right
	SendCommand (0,L_R,0,0);
}

//*****************************************************************************************
// write a character to the LCD

void WriteCharLCD (char Element)
{
	unsigned char D3,D2,D1,D0;

		LCD_RS = 1;

		D3 = (Element & 0b10000000) >> 7;	// Splitting of the first nibble
		D2 = (Element & 0b01000000) >> 6;
		D1 = (Element & 0b00100000) >> 5;
		D0 = (Element & 0b00010000) >> 4;

		SendCommand (D3,D2,D1,D0);

		D3 = (Element & 0b00001000) >> 3;	// Splitting of the second nibble
		D2 = (Element & 0b00000100) >> 2;
		D1 = (Element & 0b00000010) >> 1;
		D0 = (Element & 0b00000001);

		SendCommand (D3,D2,D1,D0);

		LCD_RS = 0;
}

//*****************************************************************************************
// This function allow to write a costant string to the LCD
```

```
void WriteStringLCD(const rom char *buffer)
{
    while(*buffer)              // Write data to LCD up to null
    {
        WriteCharLCD(*buffer);  // Write character to LCD
        buffer++;           // Increment buffer
    }
    return;
}

//*****************************************************************************************
// This function allow to write a string Variable to the LCD

void WriteVarLCD(char *buffer)
{
    while(*buffer)              // Write data to LCD up to null
    {
        WriteCharLCD(*buffer); // Write character to LCD
        buffer++;           // Increment buffer
    }
    return;
}
//*****************************************************************************************
// this function clean the LCD

void ClearLCD ()
{
        SendCommand (0,0,0,0);
        SendCommand (0,0,0,1);
}

//*****************************************************************************************
//This function turn OFF or ON the cursor of the display and its blinking
//0=Turn OFF    1= Turn ON

void CursorLCD(char ON_OFF,char Blink)
{
        SendCommand (0,0,0,0);
        SendCommand (1,1,ON_OFF,Blink);

}

//*****************************************************************************************
//Reset the LCD

void  OpenLCD()
{
  //inizializzazione secondo quanto dice il data sheet fig 24
        LCD_RS = 0x00;
        LCD_E = 0x00;

        delayLCD (1000);
        SendCommand (0,0,1,1);
        delayLCD (1000);                    // delay is more than 4ms
```

```
        SendCommand (0,0,1,1);
        delayLCD (1000);                // delay is more than 4ms
        SendCommand (0,0,1,1);
        delayLCD (1000);                // delay is more than 4ms
        SendCommand (0,0,1,0);
        //IO// e qui finisce la prima parte
        //
        //IO//Function set: dati su 4 bit, display con 2 righe, dimensione carattere 5x8
        SendCommand (0,0,1,0);
        SendCommand (1,0,0,0);
        //IO//display on(/off), cursore on, blinking off
        SendCommand (0,0,0,0);
        SendCommand (1,1,1,0);
        //IO// come quella sopra, ma spegne il cursore
        CursorLCD (0,0);
        //IO//cancella LCD
        ClearLCD ();
}


    #endif
```