

Laborator 5

Utilizarea unui LCD alfanumeric

Prezentarea instrumentelor interne de tip ceas - timer

Prezentare afișaj de tip LCD alfanumeric

LCD-ul (Liquid Crystal Display) este un modul electronic de afișare utilizabil într-o mare varietate de aplicații. Un modul de afișaj de 16x2 este des întâlnit în diferite aplicații fiind preferat modulelor de afișaj de 7 segmente, motivele fiind: este ușor de programat, nu are limitări la afișarea caracterelor speciale, posibilitatea de realizare de animații și nu în ultimul rând un consum redus (circa 1,5mA + 20mA iluminarea de fundal) față de un afișaj pe leduri / segmente (fiecare segment aprins consumând cel puțin 20mA).

Un modul de afișare de 16x2 poate afișa 16 caractere pe 2 linii. Fiecare caracter este afișat pe o matrice de 5x8 pixeli (Figura 1).

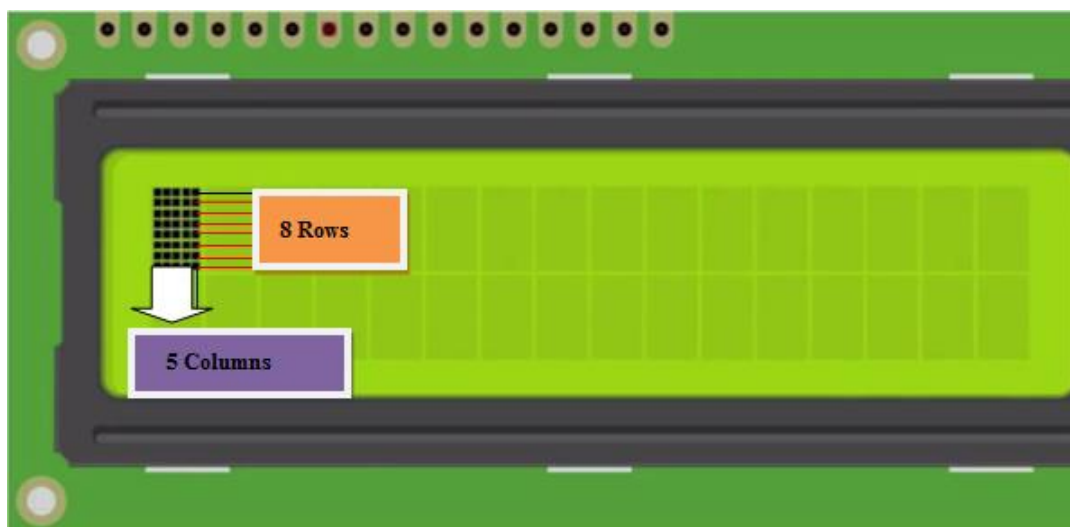


Figura 1. Matricea de pixel pentru un caracter

Acest LCD funcționează cu ajutorul a doi regiștrii: *Command* și *Data*. În regiștrul *Command* sunt stocate comenzile ce sunt trimise LCD-ului. Fiecare LCD are integrat un cip care îl controlează. Acest cip cunoaște un set de comenzi predefinite, precum: inițializare, ștergere, setarea poziției cursorului ș.a.m.d. Regiștrul *Data* înregistrează datele ce vor fi afișate pe ecran. Datele sunt înregistrate în cod ASCII. Semnificația pinilor de interconectare a unui modul LCD se pot observa în imaginea următoare (Figura 2).

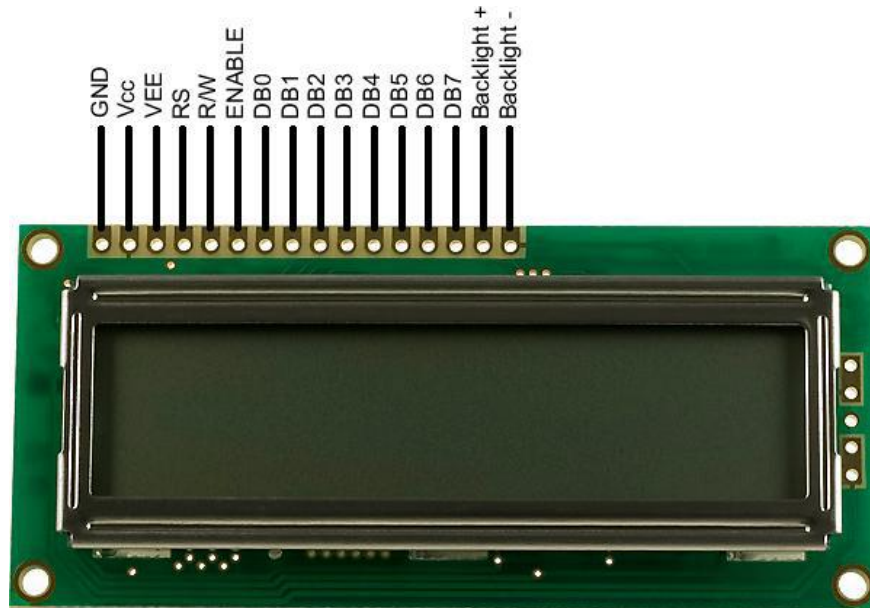


Figura 2. Semnificația Pinilor

Nr. Pin	Nume	Funcție
1	Ground	Închiderea circuitului la masă (0V)
2	VCC	Alimentarea circuitului (5V)
3	VEE	Ajustarea contrastului
4	Register Select	Face diferențierea între date și comenzi în ceea ce privește liniile de date. RS=0 - Instruction input; RS=1 - Data Input
5	Read/Write	Face diferențierea între cicluri de citire și scriere. RW=0 - scriere; RW=1 - citire
6	Enable	Semnal de activare a circuitului
7-14	Data Port	Portul bi-direcțional pe care sunt scrise /citite datele și comenzile
15	Backlight +	Alimentare lumină de fundal
16	Backlight -	Masă circuit lumină de fundal

Tabel 1. Descrierea pinilor

Pentru funcționarea circuitului trebuie conectați cel puțin următorii pinii:

- Ground – pin ce trebuie conectat la masă;
- Vcc – pin ce trebuie conectat la o tensiune între 4.7V și 5.3V;
- VEE – pin ce poate fi conectat la o rezistență variabilă de 10k Ω (potențiometru);

- Register Select – pin ce trebuie conectat la o ieșire digitală a unui microcontroler și controlat de acesta;
- Read/Write – pin ce este conectat la o ieșire digitală a microcontrolerului pentru a controla scrierea sau citirea de valori din registrele interne ale LCD-ului;
- Enable – pin ce trebuie conectat la o ieșire digitală a unui microcontroler;
- Data Port – este compus din 8 pini pe care se pot scrie sau citi datele din regiștrii interni. Se pot utiliza toți 8 pinii acestui port, dar se pot conecta un minim de numai 4 pini (liniile 4-7) pentru a se realiza o conexiune pe o magistrală de 4 biți.

Pentru a se activa lumina de fundal, se pot conecta și pinii 15, 16, pinul 15 la o sursă de minim 3.3V și pinul 16 la masă.

Utilizarea LCD-ului în mediul Keil uVision folosind o placa NucleoF103

Pentru a putea realiza o conexiune între microcontroler și LCD, se va folosi o bibliotecă open source ([link github](#)).

Principalele funcții implementate în bibliotecă și folosite în acest proiect sunt:

- *lcd16x2_gotoxy(uint8_t x, uint8_t y)* - pune cursorul pe poziția specificată de parametrul *x* (coloana) și *y* (linia);
- *lcd16x2_puts(const char* s)* - afișează pe display string-ul trimis prin parametrul *s* începând cu poziția la care a fost setat cursorul;
- *lcd16x2_clrscr()* - șterge toate caracterele de pe ecran și poziționează cursorul la 0,0;
- *lcd16x2_init(uint8_t disp_attr)* – face inițializările necesare, cu set-up pentru magistrala pe 4 biți (pinii de date 4-7) și cu alți parametri definiți în fișierul *lcd.h*.

Pentru a permite utilizatorului să conecteze LCD-ul la microcontroler după necesități, fără a face modificări majore în cod, s-au definit în fișierul *lcd.h* pinii și porturile ce urmează a fi folosite.

```
/** Port and pin definition for 4-bit mode ----- */  
// LCD control lines (must be on the same port)  
#define LCD16X2_RCC_GPIO_CONTROL      RCC_APB2Periph_GPIOB  
#define LCD16X2_GPIO_CONTROL          GPIOB  
#define LCD16X2_GPIO_RS                GPIOB  
#define LCD16X2_GPIO_RW                GPIOB  
#define LCD16X2_GPIO_EN                GPIOB
```

```

// LCD data lines (must be on the same port)
#define LCD16X2_RCC_GPIO_DATA      RCC_APB2Periph_GPIOA
#define LCD16X2_GPIO_DATA          GPIOA
#define LCD16X2_GPIO_D4             GPIOA
#define LCD16X2_GPIO_D5             GPIOA
#define LCD16X2_GPIO_D6             GPIOA
#define LCD16X2_GPIO_D7             GPIOA
// Pin definition
#define LCD16X2_PIN_RS              GPIO_Pin_5
#define LCD16X2_PIN_RW              GPIO_Pin_4
#define LCD16X2_PIN_EN              GPIO_Pin_10
#define LCD16X2_PIN_D4              GPIO_Pin_9           // 4-bit mode LSB
#define LCD16X2_PIN_D5              GPIO_Pin_7
#define LCD16X2_PIN_D6              GPIO_Pin_6
#define LCD16X2_PIN_D7              GPIO_Pin_5           // 4-bit mode MSB

```

- *void lcd16x2_create_custom_char(uint8_t location, uint8_t* data_bytes)* – pe lângă caracterele ASCII recunoscute de LCD, programatorul își poate crea propriile sale caractere sau simboluri. Prin parametrul *data_bytes* se transmite vectorul care descrie caracterul dorit. Este format din 8 numere, pentru fiecare linie a matricei (figura 1), fiecare număr fiind reprezentat pe 5 biți. De exemplu 0x1C (11100) va avea primii trei pixeli de pe linie aprinși, iar ceilalți stinși (figura 3).

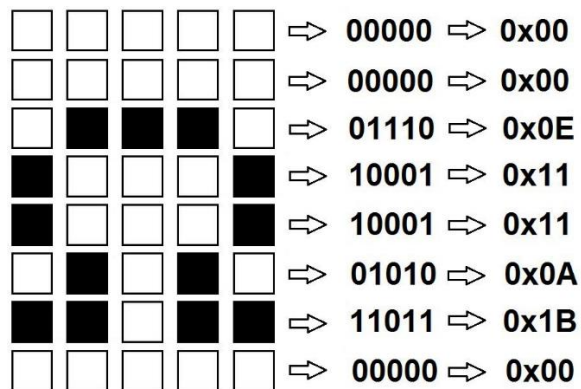


Figura 3. Crearea unui caracter nou

Timere și Watchdog

Watchdog

Pe plăcile STM32 sunt două periferice de tip watchdog (Independent și Window) care oferă un nivel înalt de siguranță, acuratețe și flexibilitate. Ambele servesc la detecția și rezolvarea problemelor apărute în urma unor erori software, prin resetarea sistemului sau prin declanșarea unei întreruperi când counter-ul atinge o anumită valoare (de timeout).

Watchdog-ul IWDG (independent watchdog) primește semnalul de ceas de la un LSI dedicat (low-speed internal clock), ceea ce înseamnă că el va rămâne activ chiar și atunci când există o problemă cu ceasul sistemului. Watchdog-ul WWDG (window watchdog) își ia semnalul de ceas de pe APB1 folosind un prescaler și are o fereastră de timp configurabilă, care poate fi programată să detecteze un comportament anormal în aplicație, fie ca aparut mai devreme fie întârziat. IWDG este mai potrivit pentru aplicațiile care necesită ca watchdog-ul să ruleze ca un proces independent față de aplicația principală, dar va avea o acuratețe mai mică a timpului. WWDG este potrivit pentru aplicațiile cu constrângerea ca watchdog-ul să reacționeze la un interval de timp foarte precis.

Timere

Un timer este o componentă hardware aflată în interiorul microcontrolerului. El are un registru special, numit counter, în cazul plăcii Nucleo F103 pe 16 biți, care dispune de un mecanism hardware de reîncărcare automată a valorii de pornire. Acesta funcționează independent de activitatea microcontrolerului. Odată ce a fost activat, counter-ul va număra automat în sens crescător, descrescător sau crescător-descrescător (up, down, up/down), având ca sursă un semnal provenit de la prescaler. Pentru fiecare front crescător al semnalului provenit de la prescaler, hardware-ul incrementează sau decrementează valoarea din registrul CNT (counter). Semnalul sursă poate să provină de la ceasul intern sau extern, de la un senzor (de exemplu senzor hall), poate să fie generat de un alt timer.

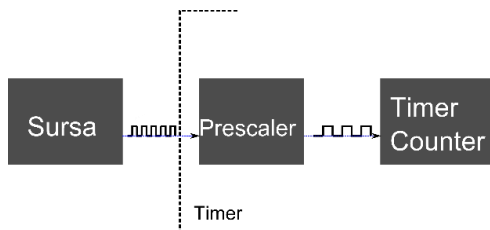


Figura 4. Modul de funcționare a prescaler-ului.

Prescaler-ul este un circuit folosit pentru a diviza frecvența cu orice factor între 1 și 65536. Se bazează pe un counter pe 16 biți, controlat printr-un registru de 16 biți *TIMx_PSC*. Prin configurarea prescalerului se face un compromis între rezoluție și intervalul timerului. O rezoluție bună înseamnă o frecvență mare a semnalului, însă acest lucru mai înseamnă și că valoarea din counter va ajunge foarte rapid la limită (overflow).

Microcontrolerul STM32F103xx are inclus un timer avansat, trei timere de uz general (general purpose), două watchdog-uri și un timer SysTick. În tabelul de mai jos se pot vedea diferențele dintre cele 4 timere ale acestui microcontroler.

Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
TIM1	16-bit	Up, down, up/down	Any integer between 1 and 65536	Yes	4	Yes
TIM2, TIM3, TIM4	16-bit	Up, down, up/down	Any integer between 1 and 65536	Yes	4	No

Tabel 2. Comparatie între timerele microcontrolerului STM32F103 – Datasheet rev.17

pag 17

Timer avansat (TIM1)

Timerul T1 poate să fie folosit pentru a genera PWM pe trei faze, datele fiind multiplexate pe 6 canale. Are ieșiri complementare pentru generare de PWM. De asemenea, poate să fie folosit ca un timer de uz general. Dacă este configurat ca un timer pe 16 biți, acest timer are aceleași funcționalități ca și timerele pentru uz general (TIM2, TIM3 respectiv TIM4). Dacă acest timer este configurat ca generator de PWM, are capacitatea de a genera modulație de la 0 la 100%. Multe funcționalități sunt comune între timere, toate având o aceeași arhitectură.

Timere de uz general

Microcontrolerul STM32F103 are trei astfel de timere. Toate cele trei timere sunt doar niște numărătoare crescătoare/descrescătoare pe 16 biți având un prescaler programabil pe 16 biți și funcție de auto-reload. Fiecare timer dispune de 4 canale pentru intrări, PWM, mod one-pulse etc. Cele trei timere pot să lucreze împreună folosind Timer Link pentru eventuale sincronizări sau evenimente. Counter-ul poate să fie rulat pas cu pas în modul debug. Oricare din aceste trei timere poate fi folosit pentru a genera semnale PWM. De asemenea, ele dispun de un hardware independent care poate să genereze semnale către DMA.

Dacă se alege modul în care se numără crescător (upcounting mode), counter-ul va porni numărătoarea de la valoarea 0 până la valoarea de auto-reload setată de utilizator în registrul *TIMx_ARR*, după care se resetează înapoi la 0 și generează un eveniment de overflow.

Schema de interconectare a afișajului LCD 16x2 cu placa de dezvoltare Nucleof103

