

Continental Learning

Laboratory 3.A SW

Laboratorul 3:

- Prezentarea afisajului driver-ului LCD ILI9341.
- Configurarea și prezentarea driver-ului SPI pentru controlul LCD.
- Aplicații demonstrative.

1. Prezentare sumară a driver-ului ILI9341

Driver-ul ILI9341 este un chip produs de compania ILITEK si este destinat controlului afisajelor TFT bazate pe cristale lichide(LCD) cu o rezolutie de 320x240 de pixeli.

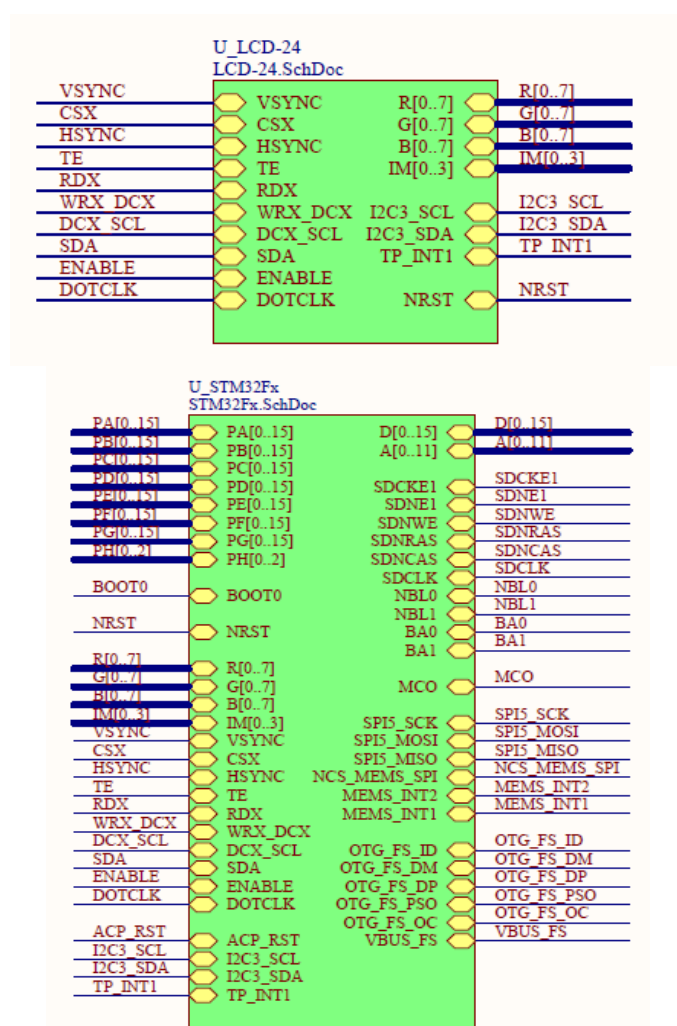
Acest driver deține o magistrala de control și trei magistrale de date: una de tip paralel(de la 8 pana la 18 biți), o magistrala RGB și una serială (SPI).

De asemeni driver-ul are o memorie RAM cu o capacitate de 172800 de octeti, tensiunea de lucru poate fi între 1.65V și 3.3V. Aceste caracteristici fac posibilă utilizarea respectivului driver în producția de dispozitive mobile(smartphone, MP3 player, etc.).

Documentul de specificatii tehnice se poate accesa utilizând link-ul de mai jos:

<https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>

Figurile de mai jos reprezintă schema bloc a modului LCD asa cum este conectat la microcontroller



Pentru o mai bună înțelegere a modului în care modulul LCD este conectat la microcontroller, este necesar un studio asupra documentului atasat acestui laborator, ce conține schema hardware a plăcuței de dezvoltare (fișierul STM32f429I_schematic.pdf).

2. Configurarea și prezentarea driver-ului SPI pentru controlul LCD.

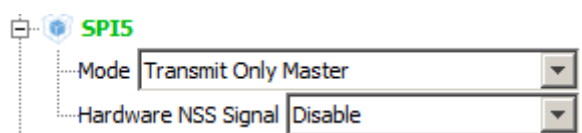
2.1 Configurarea driver-ului SPI

Controlul afișajului LCD utilizat în cadrul laboratorului, este construit sau bazat pe comunicația de tip SPI prin intermediul careia microcontroller-ul va fi capabil să transmită cuvinte de comanda cât și cuvinte de date către chip-ul ILI9341. Așadar microcontroller-ul va trebui să își assume rolul de **master** pe când modulul LCD va fi privit ca fiind **slave**.

Configurarea modului SPI se va face folosind plugin-ul STM32CUBEMX utilizând următorii parametri:

- Baudrate – va fi configurat pentru o rată de transfer de 11,25 Mb/s
- CPOL – starea de nivel a liniei de clock va fi configurată pe front negativ
- CPHA – momentul de eșantionare a bitilor se va configura ca fiind pe front pozitiv
- NSS – linia digitală de selecție a slave-ului va fi controlată software
- Formatul cadrului de transmisie utilizat va fi Motorola
- Numarul de biti transmiși într-un cadru va fi 8
- Primul bit transmis va fi MSB.

Modulul LCD este conectat pe magistrala serială de date SPI5:



Configurația necesară lucrului cu afișajul LCD:

SPI5 Configuration

✓ Parameter Settings | ✓ User Constants | ✓ NVIC Settings | ✓ DMA Settings | ✓ GPIO Settings

Configure the below parameters :

Search : ↕ ↕

Basic Parameters	
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters	
Prescaler (for Baud Rate)	8
Baud Rate	11.25 MBits/s
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

Advanced Parameters	
CRC Calculation	Disabled
NSS Signal Type	Software

Restore Default Apply Ok Cancel

După efectuarea configurării comunicației de tip SPI, se vor importa în cadrul proiectului fișierele: ili9341.c, ili9341.h și font.h acestea fiind accesate accesând link-ul următor:

<https://drive.google.com/open?id=0B4utTsTBoHDDb3MyemRoNWJoZHM>

Fișierele se regăsesc în folder-ul Laboratoare/Exemplu_LCD_ILI9341.

2.2 Prezentarea driver-ului LCD

Pachetul de fișiere sursă ILI9341 importate în proiect conțin toate funcțiile necesare configurării modului LCD, funcții cu ajutorul cărora se pot afișa diferite informații cât și funcții care permit desenarea unor figuri geometrice de bază.

Descrierea sumară a funcțiilor furnizate :

❖ **void** TFT_write(uint8_t value, uint8_t mode)

- funcția transmite un octet(8 biți) prin SPI la modulul LCD
- parametrul **value** reprezintă octetul ce urmează a fi transmis
- parametrul **mode** specifică ceea ce reprezintă acel octet pentru modulul LCD:
 - **mode** = 1 – octetul transmis reprezintă cuvânt de date
 - **mode** = 0 - octetul transmis reprezintă cuvânt de comandă

❖ **void** TFT_write_word(uint16_t value, uint8_t mode)

- funcția transmite un cuvânt de 16 biți prin SPI la modulul LCD
- parametrul **value** reprezintă octetul ce urmează a fi transmis
- parametrul **mode** specifică ceea ce reprezintă acel octet pentru modulul LCD:
 - **mode** = 1 – octetul transmis reprezintă cuvânt de date
 - **mode** = 0 - octetul transmis reprezintă cuvânt de comandă

❖ **void** TFT_on_off(uint8_t mode)

- funcția transmite un cuvânt de comandă de oprire sau activare a afișării pe display.
- parametrul **mode** specific starea în care se dorește a fi pus modulul LCD:
 - **mode** = 0x29 – activare afișare pe display
 - **mode** = 0x28 - oprire afișare pe display

❖ **void** TFT_set_rotation(uint8_t value)

- funcția transmite un cuvânt de comandă ce permite schimbarea modului de afisare pe display, portrait sau landscape cu opțiunea de rotire la 180 de grade în ambele cazuri.

- parametrul **value** specifică tipul de orientare:
 - **value** = 1 – afișare tip portrait
 - **value** = 2 - afișare tip portrait cu rotire la 180 de grade
 - **value** = 3 - afișare tip landscape
 - **value** = 4 - afișare tip landscape cu rotire la 180 de grade

❖ **void TFT_init (void)**

- funcția transmite toti parametrii necesari inițializării driver-ului ILI9341. Această funcție trebuie apelată după ce s-a făcut initializarea SPI-ului.

❖ **void TFT_set_display_window(uint16_t xpos1, uint16_t ypos1, uint16_t xpos2, uint16_t ypos2)**

- funcția setează adresa de inceput(x1,y1) și de sfârșit(x2,y2) zonei din memoria RAM, a driver-ului ILI9341, ce se dorește a fi scrisă.
- parametrii **xpos1** și **ypos1** reprezintă coordonatele in pixeli ale adresei de start.
- parametrii **xpos2** și **ypos2** reprezintă coordonatele in pixeli ale adresei de stop.

❖ **void TFT_fill (uint16_t colour)**

- funcția pune toti pixelii display-ului pe o singură culoare.
- Parametrul **colour** reprezintă valoarea culorii pe 16 biți in format RGB565.

❖ **void TFT_fill_area(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t colour)**

- funcția pune toti pixelii unei anumite zone pe o singură culoare.
- parametrii **x1** și **y1** reprezintă coordonatele in pixeli ale adresei de start.
- parametrii **x2** și **y2** reprezintă coordonatele in pixeli ale adresei de stop:
- parametrul **colour** reprezintă valoarea culorii pe 16 biți in format RGB565.

❖ **void Draw_Pixel (uint16_t x_pos, uint16_t y_pos, uint16_t colour)**

- funcția desenează un pixel pe display cu o anumită culoare.
- parametrii **x_pos** și **y_pos** reprezintă coordonatele pixel-ului
- Parametrul **colour** reprezintă valoarea culorii pe 16 biți în format RGB565.

❖ **void Draw_line(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t colour)**

- funcția desenează o linie pe display cu o anumită culoare specificată.
- parametrii **x1** și **y1** reprezintă coordonatele în pixeli ale adresei de start.
- parametrii **x2** și **y2** reprezintă coordonatele în pixeli ale adresei de stop:
- parametrul **colour** reprezintă valoarea culorii pe 16 biți în format RGB565.

❖ **void Draw_rectangle(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint18_t fill, uint18_t type, uint16_t colour, uint16_t back_colour)**

- funcția desenează un patrulater.
- parametrii **x1** și **y1** reprezintă coordonatele în pixeli coltului din stanga-sus .
- parametrii **x2** și **y2** reprezintă coordonatele în pixeli ale coltului din dreapta-jos.
- Parametrul **fill** specifică dacă patrulaterul o să fie umplut cu o culoare sau nu(1/0).
- Parametrul **type** specifică dacă patrulaterul este patrat sau nu(1/0).
- parametrul **back_colour** reprezintă valoarea culorii de fundal pe 16 biți în format RGB565.
- parametrul **colour** reprezintă valoarea culorii conturului pe 16 biți în format RGB565.

❖ **void Draw_Circle(uint16_t xc, uint16_t yc, uint16_t radius, uint16_t fill, uint16_t colour)**

- funcția desenează un cerc.
- parametri **xc** și **yc** reprezintă coordonatele in pixeli ale centrului cercului.
- parametrul **radius** reprezintă raza in pixeli a cercului.
- Parametrul **fill** specifică dacă cercul o sa fie umplut cu o culoare sau nu(1/0).
- parametrul **colour** reprezintă valoarea culorii pe 16 biți in format RGB565.

❖ **void print_char(uint16_t x_pos, uint16_t y_pos, uint8_t font_size, uint16_t colour, uint16_t back_colour, uint8_t ch)**

- funcția desenează un caracter pe display.
- parametri **x_pos** și **y_pos** reprezintă coordonatele in pixeli ale caracterului.
- parametrul **font_size** reprezintă marimea de font a caracterului.
- parametrul **colour** reprezintă valoarea culorii caracterului pe 16 biți in format RGB565.
- parametrul **back_colour** reprezintă valoarea culorii de fundal pe 16 biți in format RGB565.
- parametrul **ch** reprezintă caracterul ce se dorește afișat.

❖ **void print_str(uint16_t x_pos, uint16_t y_pos, uint8_t font_size, uint16_t colour, uint16_t back_colour, uint8_t ch)**

- funcția desenează un șir de caractere pe display.
- parametri **x_pos** și **y_pos** reprezintă coordonatele in pixeli ale șirului de caractere.
- parametrul **font_size** reprezintă marimea de font a șirului de caractere.

- parametrul **colour** reprezintă valoarea culorii şirului de caractere pe 16 biți în format RGB565.
- parametrul **back_colour** reprezintă valoarea culorii de fundal pe 16 biți în format RGB565.
- parametrul ***ch** reprezintă adresa şirului de caractere ce se dorește afișat.

3. Aplicații demonstrative:

3.1 Să se testeze toate funcțiile relevante afișărilor pe display furnizate de biblioteca sursă ILI9341.

3.2 Folosind butonul de pe placa, să se implementeze o aplicație ce incrementează valoarea unei variabile ce este afișată pe display.