

## LCD-EXT2

- Computação Embarcada - Insper
- Rafael Corsi

### Resumo :

Esse exemplo demonstra o controle do módulo de LCD maX Touch X PLAINED PRO via interface SPI (Serial Peripheral Interface Bus).

Periféricos uC:

- Power Managment Controller (PMC)
- USART
- Serial Protocol Interface (SPI)

APIs :

- Driver ili9488

Módulos :

- LCD maX Touch X PLAINED PRO

### Diagrama

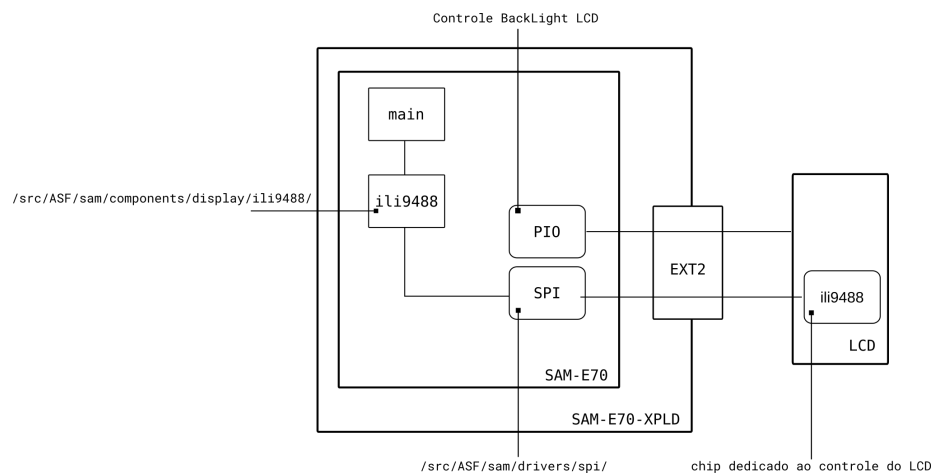


Figure 1: Diagrama de blocos

## Conexão e configuração

Deve-se configurar o módulo do LCD para operar via SPI via o dip switch localizado na parte de traz do módulo, como na imagem a baixo :

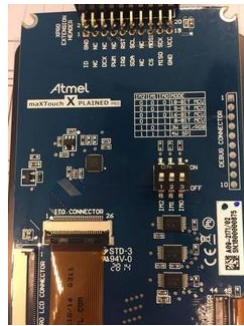


Figure 2: Dip switch

Para conectar o LCD no EXT2, deve-se seguir a referência de polaridade do flat cable a baixo :



Figure 3: Polaridade cabo

## USART

- BaudRate : 115200
- StopBit : 1 bit
- Paridade : 0 bit
- Data : 8 bits

Utilizado como debug do programa, deve-se utilizar um terminal (exe. putty) no computador para acessar o printf realizado no firmware.

## SPI

- BaudRate : 20000000
- 8 bits

O Serial Peripheral Interface Bus (SPI) é uma maneira serial de dois dispositivos (ou chips) se comunicarem, o SPI é amplamente utilizado pela indústria e diversos sensores e atuadores o utilizam com forma de comunicação.

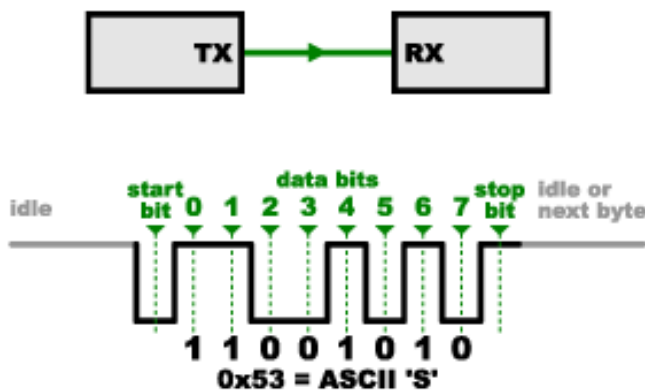


Figure 4: Comunicação SPI - (ref. sparkfun)

O SPI possui as seguintes configurações :

- Síncrono
- Barramento diferenciado de dados de entrada/saída
- Serial

Para mais informações acesse o site da sparkfun : [SPI SparkFun](#)

O SPI é utilizado pela API ili9488 para acessar e configurar o LCD.

## LCD

- Colorido
  - RGB, 8 bits por cor
- Resolução de : 320x480

### ili9488 (controlador LCD)

O ili9488 é um CI dedicado ao controle do LCD integrado no módulo LCD maX Touch X PLAINED PRO, a comunicação do microcontrolador com o LCD é via a interface com o ili9488. Esse dispositivo é responsável por :

- Acionar independente cada pixel (RGB) e fazer o acionamento (analógico)
- fazer o refresh no LCD (atualizar os pixels a uma taxa fixa)
- Receber novos valores para os pixels
- ...

Os comandos de comunicação com o ili9488 são definidos na biblioteca `/src/ASF/sam/components/display/ili9488/`. Essa biblioteca possui também alguns comandos gráficos simples tais como :

```
uint32_t ili9488_draw_pixel(uint32_t ul_x, uint32_t ul_y);

void ili9488_draw_line(uint32_t ul_x1, uint32_t ul_y1,
                      uint32_t ul_x2, uint32_t ul_y2);

void ili9488_draw_rectangle(uint32_t ul_x1, uint32_t ul_y1,
                           uint32_t ul_x2, uint32_t ul_y2);

void ili9488_draw_filled_rectangle(uint32_t ul_x1, uint32_t ul_y1,
                                   uint32_t ul_x2, uint32_t ul_y2);

uint32_t ili9488_draw_circle(uint32_t ul_x, uint32_t ul_y, uint32_t ul_r);

uint32_t ili9488_draw_filled_circle(uint32_t ul_x, uint32_t ul_y, uint32_t ul_r);
```

## RGB

O ili9488 está configurado nesse exemplo para operar em modo RGB, as cores são definidas do tipo **ili9488\_color\_t** e possuem formato RGB, sendo 8 bits para definir a intensidade de cada cor, como nos exemplos a seguir :

```
#define COLOR_BLUE          (0x0000FFu)
#define COLOR_GREEN         (0x00FF00u)
#define COLOR_RED           (0xFF0000u)
#define COLOR_NAVY          (0x000080u)
#define COLOR_DARKCYAN      (0x008B8Bu)
#define COLOR_TURQUOISE     (0x40E0D0u)
```

Porém esse formato deve ser convertido para a transmissão, via a macro : **COLOR\_CONVERT**. Exemplo :

```
ili9488_set_foreground_color(COLOR_CONVERT(COLOR_WHITE));
```

## Convertendo uma imagem para o LCD

Para conseguirmos atualizar o LCD com uma imagem pré definida será necessário convertermos essa imagem para o padrão de pixels definido na secção anterior

e depois alocarmos essa imagem em uma constante (ou no sdcard) para que o microcontrolador possa enviar ao ili9488.

Passos :

1. Definir a imagem na dimensão correta (em pixels)
2. Converter a imagem para o formato do LCD
3. Gerar um arquivo .h para ser incluído no projeto
4. Ler o arquivo .h e atualizar o LCD via a interface com ili9488

Para isso iremos utilizar o programa lcd-image-convert localizado em : Softwares Extras/lcd-image-convert/ para fazer a conversão da imagem para o formato correto.

## 1

Considere a imagem a seguir :



Figure 5: Diagrama de blocos

com dimensões de :  $398 \times 161$  px

## 2 e 3

Carrega a imagem no software lcd-image-convert e siga os passos a seguir :

## 4

Uma vez exportado o arquivo para o .c é necessário agora adicionarmos ao projeto, siga a implementação exemplo em `/src/logo.h` para criar uma constante que possui a imagem alocada.

No código temos algumas opções :

1. Atualizar o LCD pixel a pixel com os dados contidos no vetor
  - pouco eficiente, o LCD demorará para atualizar a imagem
2. Fazermos uma transferência direta de memória entre o uc e o ili9488 (burts)
  - mais eficiente e melhor método.

Para implementarmos a transferência direta de memória é necessário configurarmos a região do LCD que será atualizado, para isso utilizaremos a função *ili9488\_draw\_pixmap* que possui implementa uma atualização parcial de uma região de memória do LCD. A implementação da função está na biblioteca do ili9488 e copiado a baixo :

```

/**
 * \brief Draw a pixmap on LCD.
 *
 * \param ul_x X coordinate of upper-left corner on LCD.
 * \param ul_y Y coordinate of upper-left corner on LCD.
 * \param ul_width width of the picture.
 * \param ul_height height of the picture.
 * \param p_ul_pixmap pixmap of the image.
 */
void ili9488_draw_pixmap(uint32_t ul_x, uint32_t ul_y, uint32_t ul_width,
                        uint32_t ul_height, const ili9488_color_t *p_ul_pixmap)
{
    uint32_t size;
    uint32_t dwX1, dwY1, dwX2, dwY2;
    dwX1 = ul_x;
    dwY1 = ul_y;
    dwX2 = ul_x + ul_width - 1;
    dwY2 = ul_y + ul_height - 1;

    /* Swap coordinates if necessary */
    ili9488_check_box_coordinates(&dwX1, &dwY1, &dwX2, &dwY2);

    /* Determine the refresh window area */
    ili9488_set_window(dwX1, dwY1, (dwX2 - dwX1 + 1), (dwY2 - dwY1 + 1));

    size = (dwX2 - dwX1) * (dwY2 - dwY1);

    ili9488_write_register(ILI9488_CMD_MEMORY_WRITE, p_ul_pixmap, size * LCD_DATA_COLOR_UNIT);

    /* Reset the refresh window area */
    ili9488_set_window(0, 0, ILI9488_LCD_WIDTH, ILI9488_LCD_HEIGHT);
}

```