

Aula 01 - Display com SPI

Review do display

Tudo no arquivo [Manual do display](#).

- Ligações elétricas
- Modos de operação

16/09/21- testes pra ligar o display

Depois de umas 3 semanas tentando, pedi ajuda na aula e o sor e o pessoal me ajudaram; O teste 03 descreve o que funcionou.

teste 01

- Criei um novo projeto no STMCube;
- Configurei os pinos no IOC conforme o sor fez, todos GPIO_Output;

- ```
#define LCD_CS 12 //CE
#define LCD_RST 10 //RST
#define LCD_MO 15 //DIN
#define LCD_SCK 13 //CLK
#define LCD_DC 14 //DO
#define PORT GPIOB //GPIO onde esta o display
```

- Copiei as pastas de baixo, e colei no projeto que eu criei, sobreescrevendo

- | Nome                   | Data de modificação | Tipo               | Tamanho |
|------------------------|---------------------|--------------------|---------|
| .settings              | 23/08/2021 01:12    | Pasta de arquivos  |         |
| Core                   | 23/08/2021 01:12    | Pasta de arquivos  |         |
| Debug                  | 25/08/2021 12:43    | Pasta de arquivos  |         |
| Drivers                | 23/08/2021 01:12    | Pasta de arquivos  |         |
| IDE .cproject          | 11/08/2021 19:55    | Arquivo CPROJECT   | 35 KB   |
| .mxproject             | 11/08/2021 19:55    | Arquivo MXPROJE... | 7 KB    |
| IDE .project           | 08/08/2021 21:43    | Arquivo PROJECT    | 2 KB    |
| DispNK-sw Debug.launch | 23/08/2021 20:18    | Arquivo LAUNCH     | 8 KB    |
| DispNK-sw.ioc          | 11/08/2021 19:55    | Arquivo IOC        | 7 KB    |
| metodologia            | 09/08/2021 11:16    | Documento de Te... | 1 KB    |
| STM32F446RETX_FLASH.Id | 11/08/2021 19:55    | Arquivo LD         | 5 KB    |
| STM32F446RETX_RAM.Id   | 08/08/2021 21:43    | Arquivo LD         | 5 KB    |

Compilei sem mexer em mais nada

**não funcionou**

## teste 02

- O comando original era 0xC6
- Troquei pra B6 pois estava comentado do lado

```
LCD5110_LCD_write_byte(0xB6,0); //B6 ou C6
```

não funcionou

## Teste 03 - com ajuda do sor

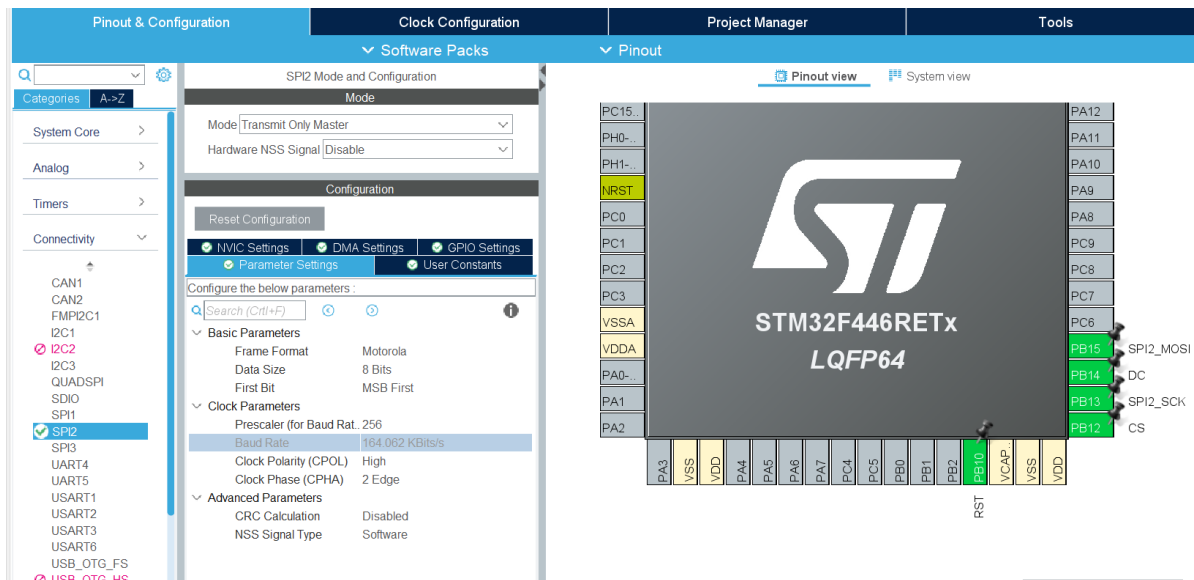
O reset estava sendo feito por hardware, então as funções do reset deviam estar descomentadas!

- descomentar
  - define do pino RST
  - parte do reset no LCD\_init
  - Função do pino de reset

!funcionou!

## Configuração no IOC

- SPI 2
- 84MHz
- ir no Project manager e configurar para criar pares .c e .h



## Configuração da SPI

- MSB
  - Bit mais significativo vai primeiro
- prescaler = 256

- Dividir o clock do periférico pelo maior possível, para trabalhar na menor velocidade primeiro (menos chances de erro)
- Transmit only master
  - Transmit only master, pois nós só vamos enviar dados ao display e vamos mandar o clock
- Modo: O display espera que o dado esteja estável para leitura na borda de subida do clock
  - borda de subida --> CPOL = 1 (high)
  - inicia na segunda borda do clock (2 edge)

## SPI no código

---

- comenta a parte do MO e do SCK feita pelo software, pois agora a SPI 2 vai fazer a rotina
- Invéz do for no write byte, usa `HAL_SPI_Transmit(&hspi2, &dat, 1, 200);`
- include spi.h

## O phsi

É usado para apontar para a estrutura de dados da hspi usada.

Assim tu só diz no init (na main) qual SPI tu ta usando (&hspi2, por exemplo) e a biblioteca irá usar os dados dessa estrutura para executar as tarefas.