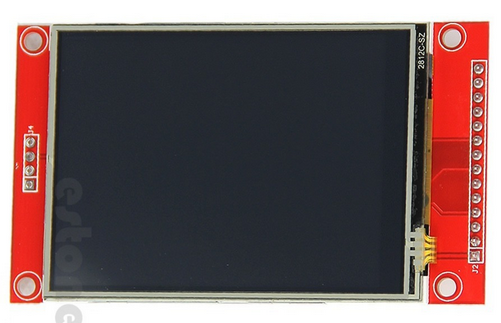
**Usando Display Gráfico TFT**

Neste exemplo utilizamos um display gráfico TFT com o controlador ili9431 plotando imagens em uma resolução de 320x240, chegando até 256 mil cores. Alguns exemplos do uso do gerenciador gráfico:

|  |  |
| --- | --- |
| Exemplo de preenchimento da tela | Exemplo de Quadrados |
| Exemplo de círculos | Exemplo de botões |
| Exemplo de triângulos | Exemplo de botões |
| Exemplo de retângulos com cantos arredondados | Exemplo de printf e textos |
| Exemplo de retângulos | C:\Users\alex\AppData\Local\Microsoft\Windows\INetCache\Content.Word\20170606_095621.jpg  Exemplo de linhas |

**Conexão com o Kit ARM**

Geralmente estes displays são vendidos em Shields incluindo o controlador, TFT e outros periféricos como slot para cartão MMCD e CI touch screen:



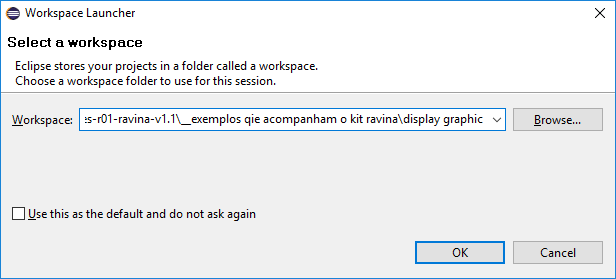
Independe de usar Shields ou não, a conexão é semelhante. O controlador ili9341 requer uma porta de comunicação SPI e alguns pinos de controle. Usaremos a porta SPI1 do ARM e pinos IO extras, isto porque este kit ARM não contém uma porta específica para o TFT. A tabela abaixo descreve a conexão:

|  |  |  |
| --- | --- | --- |
| **Módulo Display** | **Conector** | **Pino ARM** |
| Reset (RST) | P7 pino 1 | 60 (P2\_0) |
| Data/Command (D/C) | P7 pino 2 | 59 (P2\_1) |
| LED (BackLight) | P7 pino 3 | 35 (P1\_28) |
| Chip select (CS#) | P5 pino 5 | 45 (P0\_18) |
| SPI CLK | P5 pino 2 | 63 (P0\_7) |
| SPI MISO (SDO) | P5 pino 3 | 62 (P0\_8) |
| SPI MOSI (SDI) | P5 pino 4 | 61 (P0\_9) |
| VCC (3V a 5V) | P5 pino 1 |  |
| GND | P5 pino 6 |  |



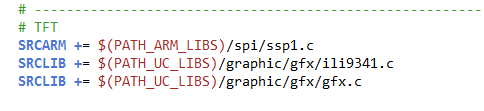
**Programa Exemplo**

Abra o exemplo da pasta “display graphic” com o Eclipse:

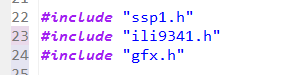


No makefile, na secção de inserção de fontes, foi incluindo as seguintes fontes:

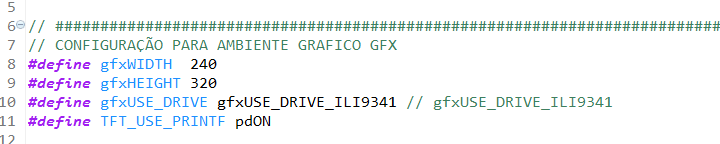
* ssp1.c: Código de acesso a porta SPI1 do ARM;
* ili9341.c: Controlador do display tft;
* gfx.c: Gerenciador gráfico.



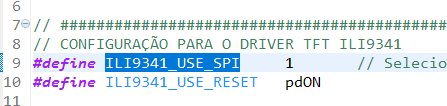
Em app.h foi adicionado os includes das respectivas fontes de códigos inseridos anteriormente:



Em \_config\_lib\_graphics.h é feito as configurações da resolução do display e de outros recursos do gerenciador gráfico:



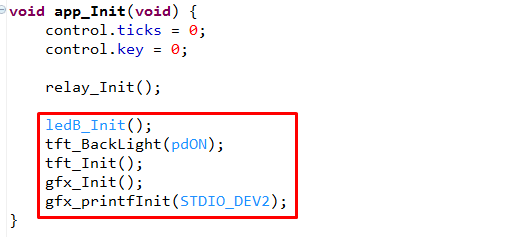
Em \_config\_cpu\_spi.h é feito a configuração de qual porta SPI de acesso para o controlador ili9341, e se vai ou não usar o pino de reset do display:



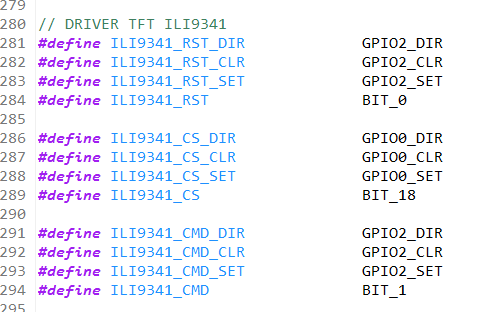
Em main.c dentro do int main(void), é inicializado a porta SPI1 com velocidade de 10MHz. Entretanto, nos testes esse display chegou a trabalhar até a 50Mhz, contudo, segundo as literaturas o controlador deve trabalhar não mais que 10Mhz:



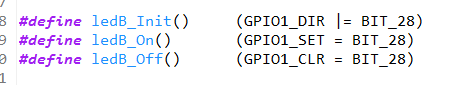
Em app.c na função app\_Init são inicializados o controlador ili9341 e o gerenciador gráfico:



Caso queira mudar de pinos dos controles é preciso que altere as suas respectivas definições no arquivo arch\_hardware\_resources.h:

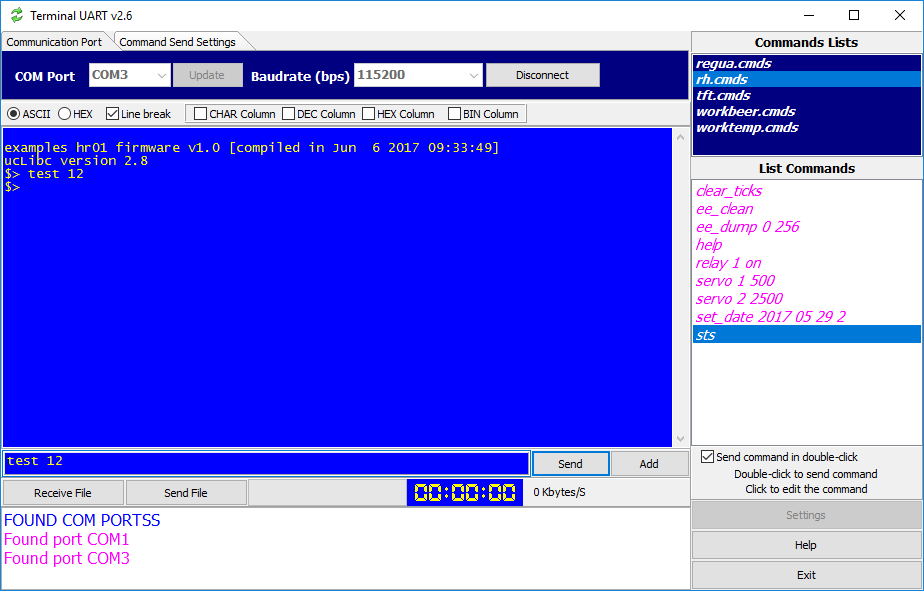


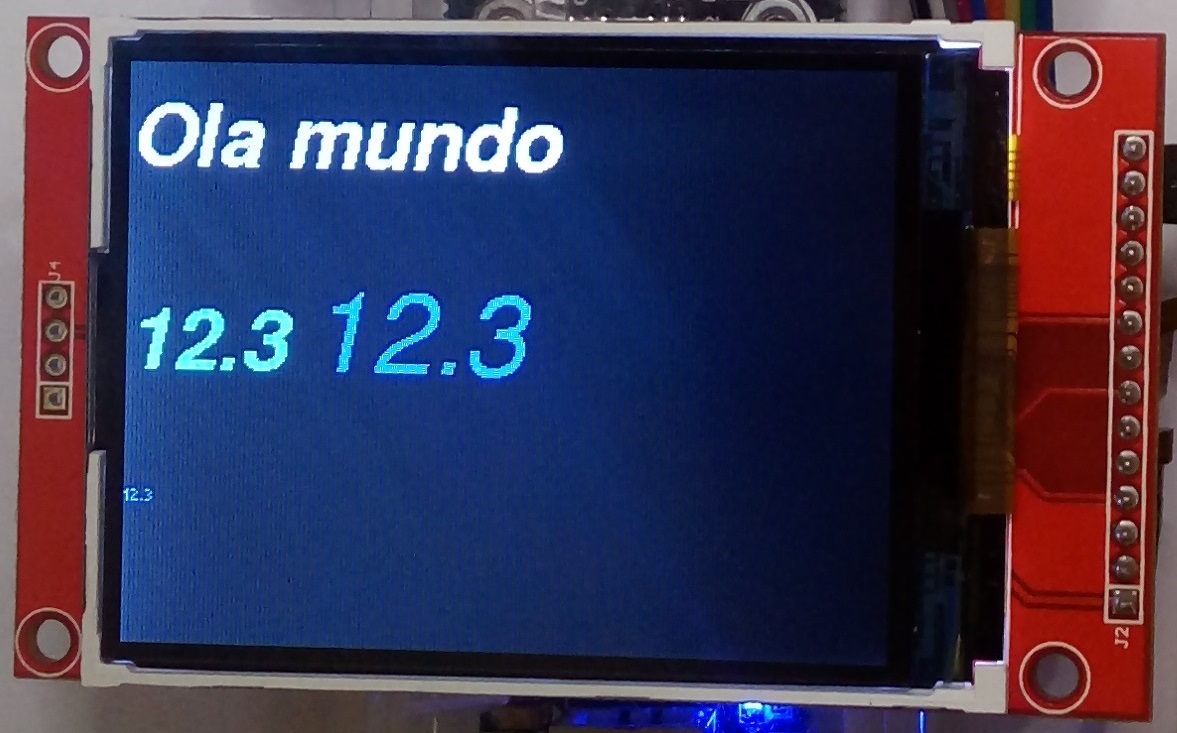
Já o pino do backlight está especificado em app.h, pois é um controle da aplicação é não dos drivers.



**Compile e Execute**

Com tudo instalado e configurado, compile a aplicação, grave e execute no ARM. Use o CLI junto ao um terminal para enviar comandos ao display:





Para mais comandos digite help:

