Vitor Amadeu Souza

Apresentando as

Armas Nacionais

Com display TFT programado no Arduino

© 2019 by Cerne Tecnologia e Treinamento Ltda.

© 2019 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Dezembro de 2019

Direitos reservados por:
Cerne Tecnologia e Treinamento Ltda
Produção: Cerne Tecnologia e Treinamento
E-mail da Empresa: cerne@cerne-tec.com.br
Home Page: www.cerne-tec.com.br.com.br
Atendimento ao Consumidor: sac@cerne-tec.com.br
Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

Dedicatória

Como nos meus outros livros, dedico este livro a minha querida esposa Renata Leal.

"Para quem tudo perde, ainda resta Deus."

Alfred de Musset

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android, iOS, Blackberry, Windows Phone e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.

Cerne
Conhecimento para o Desenvolvimento

cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento 6

1. Introdução 6

Capítulo II – Tela TFT 7

1. TFT 7
2. Pinagem 8
3. Alterando a cor de fundo..... 9
4. Desenhando linhas 13
5. Desenhando retângulos..... 15
6. Desenhando círculos 16
7. Imprimindo textos 17
8. Carregando imagens..... 19

Capítulo III – Aplicação 21

1. Introdução 21
2. Programação..... 21

Referências 25

Capítulo I
Metodologia de desenvolvimento

1. Introdução

Esta literatura é uma continuação da obra *Arduino – Prático e Objetivo* (2011) e *Programação para Arduino – Avançado* (2014) do mesmo autor e editora, onde o objetivo é controlar uma tela de TFT usando o Arduino para em seguida apresentar imagens relacionadas ao tema título da obra.

A placa didática utilizada foi a Arduino UNO em conjunto com um display TFT que faz uso do controlador ST7735, onde tais kits e acessórios estão à venda no site www.cerne-tec.com.br.

Capítulo II
Tela TFT

1. TFT

O TFT (Thin-Film-Transistor liquid-crystal display) utilizado nesta experiência é o modelo cuja imagem está apresentada a seguir.



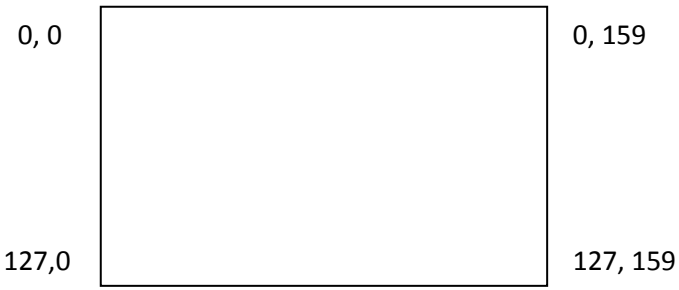
Fonte: https://www.arduino.cc/en/Guide/TFT#toc3

Este display possui uma tela de 1,8º e possui resolução 160x128 pixels (HxV). Utiliza um controlador, cujo código é ST7735 e a alimentação é feita em 5V.

Outros modelos podem ser utilizados, desde que o controlador seja compatível. O software Arduino a partir da versão 1.0.5 já dispõe de bibliotecas para acessá-lo.

A tela de TFT citada possui encaixe para cartão SD CARD, onde imagens podem ser gravadas no formato bitmap (bmp) para serem carregadas na tela. Para isso, faz-se necessário habilitar a conexão através do barramento SPI com o cartão. Este display pode ser encaixado como um shield a placa Arduino.

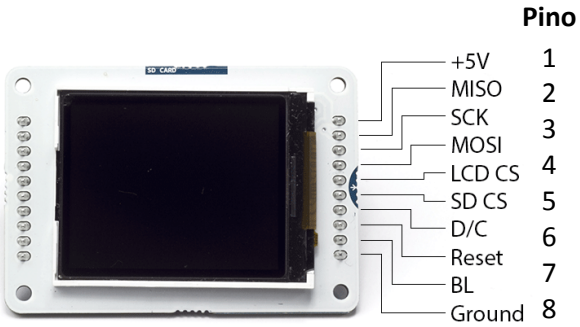
Por ter uma resolução de 160x128 pixels, podemos ver esta tela como um plano cartesiano, cujos pontos estão apresentados a seguir.



A resolução de cada pixel do display é de 16 bits, sendo 5 bits para R (Red-Vermelho) e B (Blue-Azul) e 8 para G (Green-Verde).

2. Pinagem

A pinagem deste display está apresentada abaixo.



Fonte: https://www.arduino.cc/en/Guide/TFT#toc3

A tela é alimentada em 5V através dos pinos 1 e 8. Os pinos MISO (2), SCK(3), MOSI(4) fazem a comunicação SPI entre o Arduino e o display e cartão SD. A seleção de comunicação se será feita com o TFT ou SD depende dos pinos digitais LCD CS (5) e SD CS (6). Os pinos DC e RESET também devem ser utilizados para conectar o TFT ao Arduino.

Quaisquer pinos podem ser utilizados para CS, DC e RESET, no entanto, para a comunicação SPI recomenda-se utilizar os pinos default para esta comunicação, disponível no conector ICSP da placa Arduino em razão da sua maior velocidade de transação.

3. Alterando a cor de fundo

Este experimento troca a cor da tela (background) de acordo com o parâmetro passado através do método `background` após a

inicialização do objeto `tela`. Acompanhe a seguir a listagem do código.

```
#include <SPI.h>
#include <TFT.h>

#define LCD_CS   10
#define DC       9
#define RESET    8

TFT tela = TFT(LCD_CS,DC,RESET);

void setup()
{
  tela.begin();
  tela.background(0,0,0);
}

void loop()
{

}
```

Inicialmente são feitas as inclusões referentes às bibliotecas de comunicação SPI e TFT. Em seguida, os pinos referentes à conexão com a tela gráfica como LCD_RS, DC e RESET são definidos no programa e o objeto `tela` é criado em função dos pinos definidos anteriormente.

No bloco `setup()` o TFT é inicializado pelo método begin e a cor de fundo do display é alterado pelo método `background()` já que o parâmetro passado é 0,0,0. Tal parâmetro refere-se às cores