Vitor Amadeu Souza

Acessando a memória

EEPROM

no

STM32F103C8

Programado em Arduino

© 2020 by Cerne Tecnologia e Treinamento Ltda.

© 2020 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.

Setembro de 2020

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

"Venha também sobre mim a tua benignidade, ó Senhor, e a tua salvação, segundo a tua palavra."

Sl 119:41

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 – Ligando um led 7

1. Montando o hardware 7
2. Programando o Arduino 8

Capítulo III – Piscando um led 12

1. Introdução 12
2. Montando o hardware 12
3. Programando o Arduino 12

Capítulo IV – Display LCD em modo paralelo 14

1. Introdução 14
2. Montando o Hardware 14
3. Programando o Arduino 16

Capítulo V – Aplicação 18

1. Introdução 18
2. Esquema elétrico 19
3. Fluxograma 20
4. Biblioteca 20
5. Código fonte 21

Capítulo I
Metodologia de desenvolvimento

1. Introdução

Esta literatura é uma continuação da obra *ARM no Arduino – Com base no modelo STM32F103C8 (2019)* do mesmo autor e editora, sendo importante que o leitor tenha ciência dos assuntos abordados na obra citada, para que haja um melhor aproveitamento do conteúdo a ser apresentado.

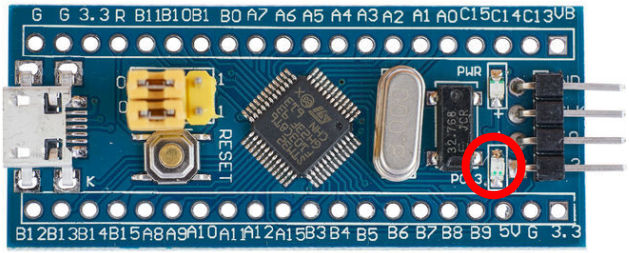
O módulo utilizado foi a *Blue Pill* STM32F103C8, onde os recursos para testar os circuitos propostos serão conectados através das conexões disponíveis na lateral do kit. Tal kit está à venda no site www.cerne-tec.com.br.

Capítulo II

Ligando um Led

1. Montando o Hardware

Neste primeiro exemplo são apresentados os passos para ligar um led no STM32F103C8 que está conectada ao pino PC13 no qual já há um led conectado ao processador na própria placa.



Na figura abaixo há a figura de um led.



Existem vários tipos de leds com diferentes cores, sendo que isto não interfere na execução do exemplo. Observe que o led é composto de dois terminais, sendo o maior o terminal positivo enquanto o menor o negativo. Isso deve ser observado com atenção, pois se invertido o led não funcionará. Além disso, é necessário ligar um resistor em série com o led, de forma a não queimá-lo. Um resistor típico pode ter sua resistência de 220 Ω a 470 Ω.

2. Programando o Arduino

A estrutura de um programa feito no Arduino IDE é a seguinte.

```
void setup()
{
}

void loop()
{
}
```

Dentro das chaves do `void setup()` são feitas as configurações básicas do Arduino. Entenda este bloco como a BIOS do computador, em que realizam-se todas as configurações iniciais para que este venha a funcionar em seguida. Dentro das chaves do bloco `void loop()`, será feita a lógica de programação que deseja-se que a placa execute de forma contínua.

Observe o pequeno trecho de código inicial no Arduino IDE e note que esta refere-se à estrutura básica do programa, onde ainda não foi feita a programação propriamente dito para ligar o led.

Um pino digital pode ser configurado tanto entrada quanto saída, ou seja, se conectar um led ao pino ele será uma saída para o MCU e se ligar um botão será uma entrada. Como um led está conectado ao pino digital deve-se configurar esta porta como saída e para isso, usa-se a função `pinMode()` que permite configurar um pino como entrada ou saída. A seguir, é apresentada a configuração desta função.

```
pinMode(pino, modo);
```

Em que na opção pino, informa-se qual pino deseja-se configurar e em modo se ele será configurado como entrada ou saída, onde coloca-se `INPUT_PULLUP` ou `OUTPUT`, respectivamente. Sendo assim, o programa fica da forma abaixo, em que o pino PC13 está configurado como saída.

```
void setup()
{
     pinMode(PC13,OUTPUT);
}

void loop()
{
```

```
}
```

O pino está configurado como saída, mais ainda não foi ligado. Para fazer isso, usa-se a função descrita a seguir.

```
digitalWrite(pino, modo);
```

Em que na opção `pino` define-se qual pino estará fazendo-se acesso e na opção modo o estado que vai assumir após o comando, neste caso em nível alto (3,3V) ou nível baixo (0V). Para nível alto, deve-se escrever `HIGH` ou `1` e para nível baixo `LOW` ou `0`. Abaixo o código fonte completo que permite acender o led conectado ao PC13.

Obs: Este LED é ativo em 0.

```
void setup()
{
     pinMode(PC13,OUTPUT);
}

void loop()
{
     digitalWrite(PC13,0);
}
```

Após digitar o código na área de edição, salve o arquivo indo na opção Arquivo -> Salvar. Em seguida, deve-se verificar se o código escrito está correto e para isso, utiliza-se a ferramenta de compilação