Vitor Amadeu Souza

Entendendo a

Interrupção Externa

no

ESP32

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.

Março 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

“A liberação da energia atômica mudou tudo, menos nossa maneira de pensar.”

Albert Einstein

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

I. Metodologia de desenvolvimento **6**

1. Introdução 6

II. Interrupção externa **7**

1. Introdução 7
2. Interrupção externa por borda de descida 8
3. Interrupção externa por borda de subida 11
4. Interrupção externa por mudança de estado 13

Referências **16**

Capítulo I
Metodologia de desenvolvimento

1. Introdução

Esta literatura é uma continuação da obra *Programando o ESP32 no Arduino(2018)* 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.

A proposta deste livro é aprender a usar a interrupção externa do ESP32, aprendendo a configurá-la de modo a tratar um processo quando um pino digital for estimulado externamente.

O módulo utilizado foi o NodeMCU-32S, 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

Interrupção Externa

1. Introdução

O ESP32 disponibiliza em todos os pinos digitais uma fonte de interrupção externa. Desta forma, o processo corrente no bloco `loop()` pode ser interrompido quando um destes pinos for estimulado externamente.

Tal recurso é importante em aplicações que exigem resposta imediata do sistema quando esta solicitação por feita, como em aplicações ligadas a segurança do trabalho, medição de rotação, detecção de fase e podem ser acionadas tanto na borda de subida, descida, mudança de estado ou por nível.

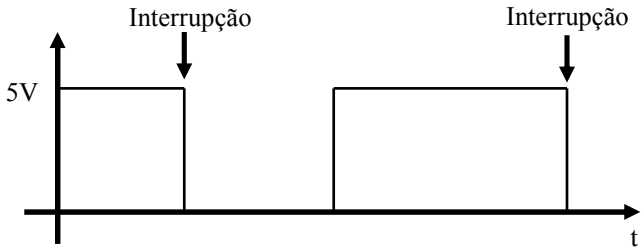
Para habilitar uma interrupção externa utiliza-se a função `attachInterrupt()` e informa-se alguns parâmetros, como o pino utilizado, nome da função responsável pelo seu tratamento e nível que faz a interrupção ser acionada, segundo o modelo apresentado abaixo.

```
attachInterrupt(pino,função,NÍVEL);
```

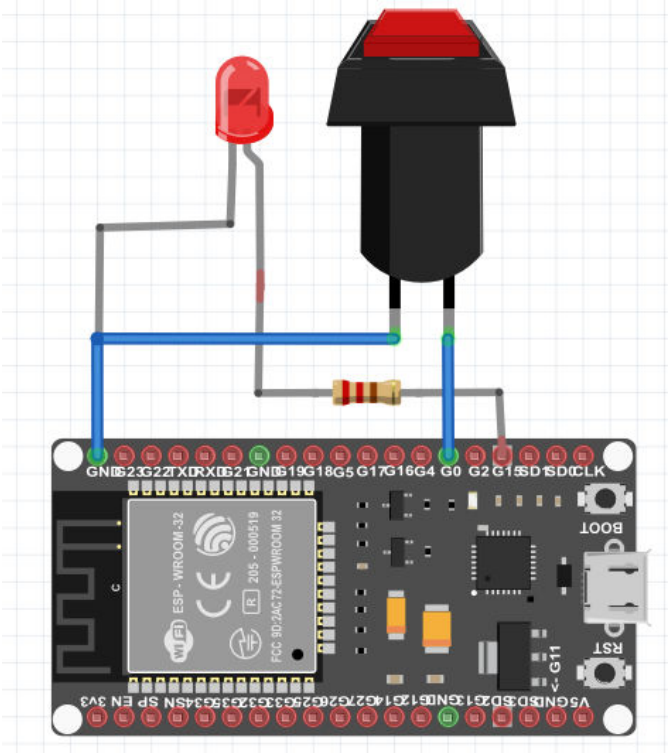
Em `pino` informa-se o pino habilitado como interrupção, como os já citados. Em `função` configura-se a ISR (Rotina de tratamento de interrupção) responsável por atender tal pedido e em `NÍVEL` o estado digital no qual ocorrerá a interrupção, podendo ser em borda de descida (`FALLING`), subida(`RISING`), mudança de estado(`CHANGE`), nível baixo(`LOW`) ou nível alto (`HIGH`).

2. Interrupção externa por borda de descida

Esta interrupção ocorre na borda de descida, como apresentado no próximo gráfico.



A seguir é apresentado um exemplo, onde o led conectado ao pino digital D1 muda de estado quando o pino digital D0 for pressionado, ou seja, ocorrer uma borda de descida neste pino. Observe inicialmente o esquema elétrico para o exercício:



O código fonte está apresentado a seguir.

```
#define LED1 1
#define LED2 2
#define BOT1 0

void setup()
{
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(BOT1,INPUT_PULLUP);
  attachInterrupt(BOT1,isr_ext0,FALLING);
}
void loop()
```

```
{
  digitalWrite(LED1,1);
  delay(500);
  digitalWrite(LED1,0);
  delay(500);
}


void IRAM_ATTR isr_ext0()
{
  digitalWrite(LED2,!digitalRead(LED2));
}
```

No bloco `setup()` é feita a configuração dos pinos através das funções `pinMode()`. Em seguida, a interrupção externa do pino GPIO1 é habilitada e configurada para chamar a função `isr_ext0()` assim que a mesma ocorrer por borda de descida (`FALLING`). Finalmente o programa entra no loop, no qual oscila continuamente um led sendo esta a rotina ou processo do experimento.

Quando o botão for pressionado e a interrupção for gerada, o programa é desviado para a função `isr_ext0()`, sendo esta responsável pelo tratamento da interrupção externa 0. Neste exemplo, um led troca de estado de modo a indicar que a interrupção ocorreu e foi tratada.

Note que neste intervalo, o led da rotina permanece no estado em que a interrupção ocorreu, voltando a piscar assim que a interrupção acabar.