

Vitor Amadeu Souza

Enviando a medição de

Voltímetro

para uma conta do

Twitter

Com ESP8266 (NodeMCU) programado em Arduino

© 2017 by Cerne Tecnologia e Treinamento Ltda.

© 2017 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 2017

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.

“Ele lhe preserva todos os ossos; nem sequer um deles se quebra.”

SI 34:20

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM, Arduino, etc. Veja os detalhes de um de nossos kits.



Kit Cerne Arduino

- Microcontrolador ATMEGA8;
- Comunicação serial RS232;
- Alimentação de 12V;
- Pinos de I/O;
- Gravação ICSP.

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc. Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.



www.cerne-tec.com.br

Sumário

| | |
|--|-----------|
| I. Metodologia de desenvolvimento | 6 |
| 1. Introdução | 6 |
| II. Conhecendo o ThingSpeak | 7 |
| 1. Introdução | 7 |
| III. Voltímetro | 12 |
| 1. Esquema elétrico | 12 |
| 2. Código fonte | 17 |
| IV. Twitter | 19 |
| 1. Introdução | 19 |
| 2. Esquema elétrico | 19 |
| 3. Fluxograma | 19 |
| 4. Código fonte | 20 |
| Referências | 23 |

Capítulo I

Metodologia de desenvolvimento

1. Introdução

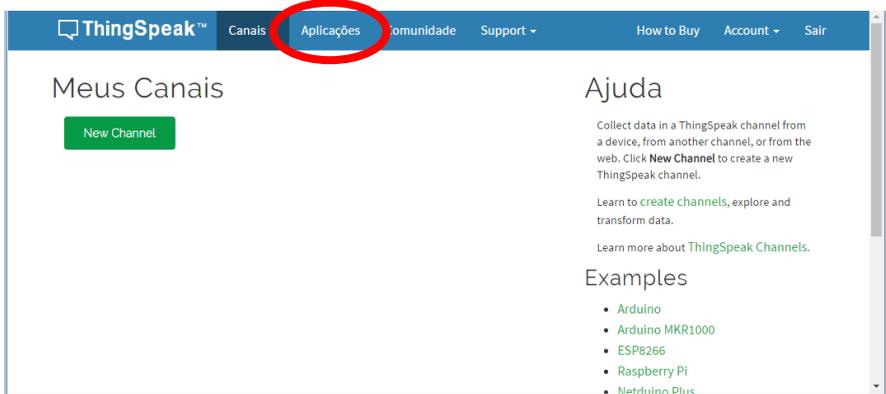
A proposta deste livro é enviar para uma conta no Twitter o resultado da medição de voltímetro, através de um exemplo prático que permita conectar o ESP8266 com base no módulo NodeMCU a internet, atualizando-a periodicamente sendo a programação do ESP8266 baseada no Arduino.

Capítulo II

Conhecendo o ThingSpeak

1. Introdução

O ThingSpeak é uma ferramenta gratuita com foco no IoT (Internet das Coisas) e outras aplicações. Acesse o site através do endereço www.thingspeak.com. Em seguida, faça o cadastro clicando no botão *Cadastro*. Após tal etapa faça o *Login* na plataforma, a tela fica como apresentado a seguir.



Clique no botão *Aplicações* marcado na figura acima. Na tela que for carregada, clique na opção ThingTweet.



Uma nova janela será carregada, no qual o botão Vincular a uma conta no Twitter estará disponível. Clique nele e vincule uma conta no Twitter ao ThingSpeak. O resultado final será algo como na figura abaixo.

Aplicações / ThingTweet

Vincular a uma conta do Twitter

| Login no Twitter | Chave | Ação |
|------------------|------------------|---|
| teste_espa66 | YwDOUWS52MXLKJSK | Gerar novamente a chave Desautorizar |

Ajuda

Use the ThingTweet app to link a Twitter® account to your ThingSpeak account. Your devices can then send alerts via Twitter by using the TweetControl API. For example, you can make a device tweet you when the temperature in your greenhouse drops or when the batteries in a device are running low.

Click [Link Twitter Account](#) to link a Twitter account to your ThingSpeak account. When asked to authorize ThingTweet to use your account:

- Enter your Twitter username and password.
- Check the **Remember me** box to cache your Twitter credentials on your computer.
- Click **Authorize app**.

Example API Endpoint

Update Twitter Status

Atenção especial ao campo Chave, pois é através dele poderemos carregar uma nova mensagem no Twitter usando o ThingSpeak como acessório. A atualização da página é feita pelo método POST, no qual é passado como parâmetro a chave observada.

Observe que neste exemplo a chave é YWDOUWS52MXLKJSK (atualize conforme o seu canal). Para enviar dados para o servidor, utilizamos a seguinte string.

```
https://api.thingspeak.com/apps/thingtweet/1/statuses/  
update?api_key=[CHAVE]&status=[VALOR]
```

Em CHAVE informa-se o valor obtido através do último passo e no campo *status* o VALOR passado como parâmetro para a página, de modo a ser enviado para a conta Twitter em seguida.

Para carregar alguns valores de teste no servidor, foram feitas as requisições apresentadas a seguir pelo browser, onde foi enviado para a chave gerada anteriormente (atualize conforme o seu canal) os valores 30, 29 e 28.

```
https://api.thingspeak.com/apps/thingtweet/1/statuses/  
update?api_key=YWDOUWS52MXLKJSK&status=28
```

```
https://api.thingspeak.com/apps/thingtweet/1/statuses/  
update?api_key=YWDOUWS52MXLKJSK&status=29
```

```
https://api.thingspeak.com/apps/thingtweet/1/statuses/  
update?api_key=YWDOUWS52MXLKJSK&status=30
```

Este mesmo procedimento deve ser feito no NodeMCU periodicamente, permitindo assim manter o servidor atualizado a respeito da variável medida.

Observando a conta no Twitter é possível visualizar os valores enviados via browser.



Capítulo III

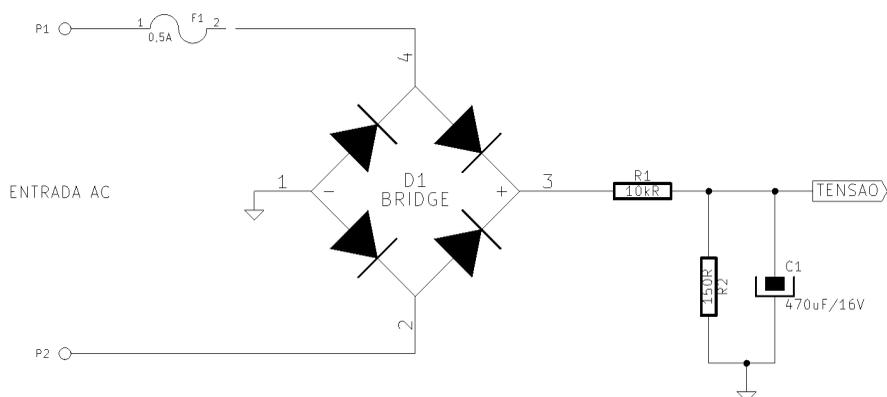
Voltímetro

1. Esquema elétrico

Para realizar a medição de fase há a necessidade de projetar um conversor para transformar a tensão alternada em contínua, de modo a conectá-la ao NodeMCU, já que a tensão fornecida a entrada analógica é filtrada, pois há um capacitor eletrolítico para isso.

Sendo assim, o seguinte circuito foi projetado para transformar AC em DC além de diminuir a tensão para a faixa de 0 a 3,3V. O circuito está projetado para a tensão nominal de 127 VAC.

Obs: *Muito cuidado ao montar este circuito, pois a tensão nominal da rede é de 127 VAC. Para garantir a segurança do leitor, utilize um transformador isolador, disjuntor e fusíveis além de uma máscara facial ou óculos de proteção.*



A saída do conversor AC-> DC fica conectada a entrada analógica do NodeMCU. Esta tensão é gerada após passar por uma ponte de diodos do tipo 1N4007 de modo a retificar o sinal senoidal e um divisor resistivo, que abaixa a tensão de forma a possibilitar a medição dentro da faixa do NodeMCU. Além disso, há um capacitor eletrolítico ligado em paralelo conectado a entrada analógica do NodeMCU. Na configuração apresentada, o circuito é capaz de medir a tensão máxima de até 150 V_{RMS} , ou seja, a saída de tensão após o divisor resistivo e devidamente filtrada será dada por.

$$V_{out} = V_{in} * \frac{150}{10150}$$

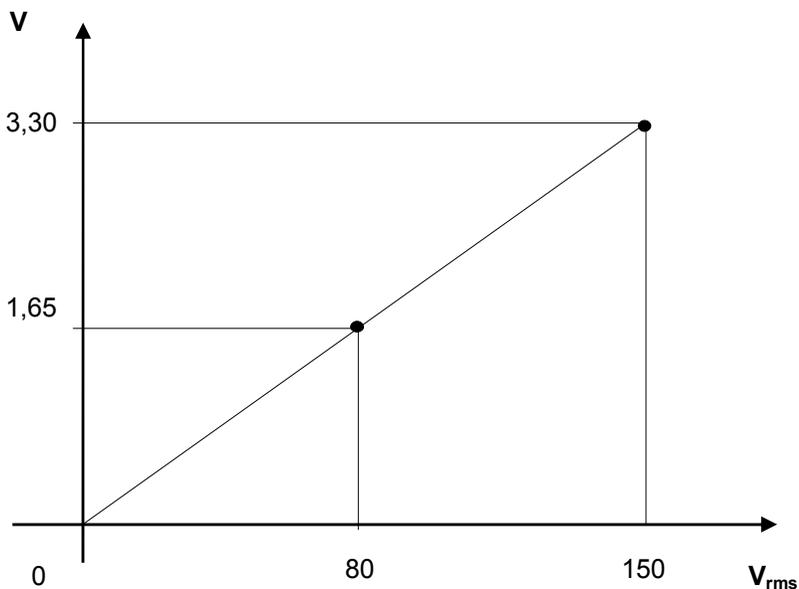
Como V_{in} máximo é de 150V que é para o que o circuito projetado a tensão de saída V_{out} conectado ao NodeMCU é encontrado da seguinte forma.

$$V_{out} = 226 * \frac{150}{10150}$$

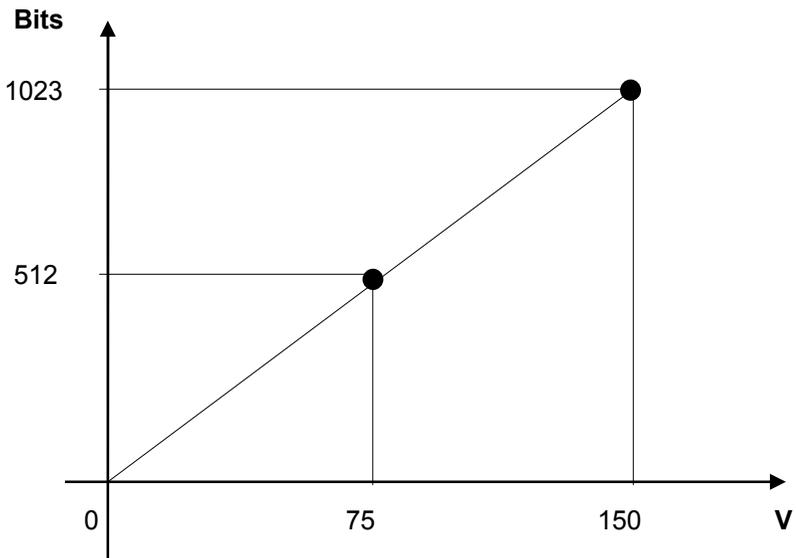
$$V_{out} = 3,3V$$

O que está dentro da faixa de operação da entrada analógica do NodeMCU que é de até 3,3 V.

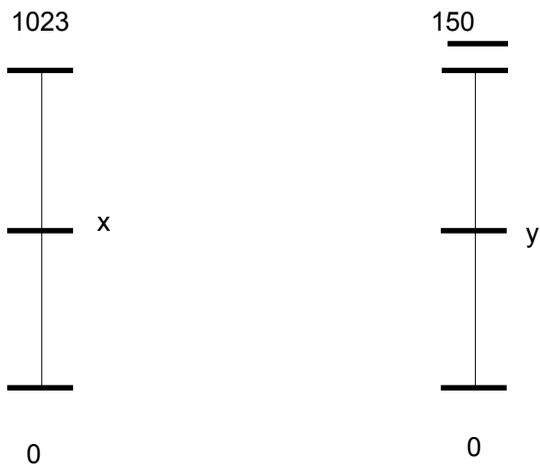
A tensão de saída V_{out} é proporcional a tensão RMS de entrada, onde caso haja a tensão mínima (0 V) observa-se na saída do conversor a tensão de 0 V. Caso haja a tensão máxima (150 V) observa-se na entrada analógica do NodeMCU aproximadamente 3,3 V e assim linearmente, conforme sugere o gráfico abaixo.



O conversor ADC do NodeMCU é de 10 bits, indicando ser capaz de fornecer uma resolução de 1024 pontos, já que 2^{10} é igual a este valor. O conversor está conectado a uma das entradas analógicas do NodeMCU, no qual fornece como resultado um valor binário na faixa de 0 a 1023. Sendo assim, pode-se modelar o seguinte gráfico para a conversão da escala de tensão RMS para os bits referentes a conversão ADC.



Desta forma, há a seguinte relação matemática entre o resultado da conversão ADC e medição dada em V_{rms} .



$$\frac{x - 0}{1023 - 0} = \frac{y - 0}{150 - 0}$$

$$\frac{x}{1023} = \frac{y}{150}$$

$$y = 150 * \frac{x}{1023}$$

Ou seja, após a leitura da entrada analógica deve-se executar o cálculo matemático acima de forma a obter o valor da tensão RMS.

2. Código fonte

A seguir o código fonte para medir a tensão RMS e enviar o resultado da medição pela serial.

```

/* * * * * * * * * * * * * *
 *           Cerne Tecnologia           *
 *           www.cerne-tec.com.br       *
 *           Projetos em Arduino - RMS  *
 * * * * * * * * * * * * */

/* * * * * * * * * * * * * *
 *           Função de inicialização     *
 * * * * * * * * * * * * */
void setup(void)
{
  Serial.begin(9600);
}

/* * * * * * * * * * * * * *
 *           Função de loop              *
 * * * * * * * * * * * * */

```

```
void loop(void)
{
  Serial.println(String(150*analogRead(0)/1023) + " V");
  delay(1000);
}
```

Capítulo IV

Twitter

1. Introdução

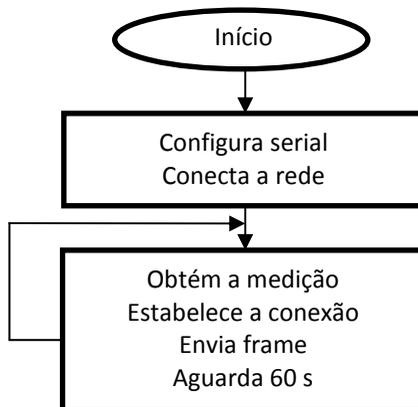
Este capítulo apresenta os passos para comunicar o módulo NodeMCU que faz uso do CI ESP8266 a internet, permitindo assim enviar periodicamente para o Twitter as informações referentes à variável mensurada.

2. Esquema elétrico

É o mesmo utilizado no capítulo passado.

3. Fluxograma

O fluxograma deste experimento pode ser observado a seguir.



4. Código fonte

A seguir está apresentado o código fonte que implementa o algoritmo anterior. Sua explicação está detalhada logo em seguida.

```
#include <ESP8266WiFi.h>

const char* ssid="Cerne";
const char* password="Tecnologia";
String tweetAPIKey = "YWDOUWS52MXLKJJSK";

void setup()
{
  WiFi.begin(ssid,password);

  while(WiFi.status() !=WL_CONNECTED)
    delay(500);
}

void loop()
{
  WiFiClient client;

  if(client.connect("api.thingspeak.com", 80))
  {
    String tweetData=String(le_valor());
    tweetData = "api_key="+tweetAPIKey+"&status="+tweetData;

    client.print("POST /apps/thingtweet/1/statuses/update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(tweetData.length());
    client.print("\n\n");
    client.println(tweetData);
  }
  delay(60000);
}

float le_valor(void)
{
  float valor,ad;
  ad = analogRead(0);
  valor=150*(ad/1023.0);
  return(valor);
}
```

O programa inicia com a definição da biblioteca (library) utilizada para comunicação WiFi assim como o nome da rede (SSID), senha (password) e chave de acesso a conta Twitter. Observe que os dois primeiros parâmetros devem ser atualizados de acordo com a sua rede local.

```
#include <ESP8266WiFi.h>

const char* ssid="Cerne";
const char* password="Tecnologia";
String tweetAPIKey = "YWDOUWS52MXLKJSK";
```

O próximo passo é iniciar a conexão com a rede interna de acordo com o SSID e senha definido.

```
void setup()
{
  WiFi.begin(ssid,password);

  while(WiFi.status() !=WL_CONNECTED)
    delay(500);
}
```

O bloco seguinte verifica se há conexão com o servidor, de modo a fazer a medição da variável do capítulo passado e formar a string a ser enviada para o servidor.

```
void loop()
{
  WiFiClient client;
```

```

if(client.connect("api.thingspeak.com", 80))
{
    String tweetData=String(le_valor());

    tweetData = "api_key="+tweetAPIKey+"&status="+tweetData;

    client.print("POST /apps/thingtweet/1/statuses/update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(tweetData.length());
    client.print("\n\n");
    client.println(tweetData);
}
delay(60000);
}

```

Observe a presença da chave obtida do capítulo II na string que fornece a conexão com o servidor ThingSpeak. Após o envio aguarde-se 60 s para que uma nova requisição seja feita. Após compilar e gravar o programa faça o teste de modo a comprovar o funcionamento da experiência.

Referências

SOUZA, Vitor Amadeu. **Implementando IoT**. São Paulo. Clube dos Autores, 2015.