Vitor Amadeu Souza

Desenvolvimento de Projetos

IoT

para monitoramento de

Sensores

Programado em Arduino no STM32F103C8

© 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

Dedicatória

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

"A maioria de nós prefere olhar para fora e não para dentro de si mesmo."

Albert Einstein

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os MCUs 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.

Cerne
Conhecimento para o Desenvolvimento

www.cerne-tec.com.br

Sumário

01. Metodologia de desenvolvimento **6**

02. Conhecendo o ThingSpeak **7**

03. Medição de corrente de 4 a 20 mA **12**

04. Medindo Álcool em ppm **25**

05. Medindo a concentração de Amônia **41**

06. Amperímetro **60**

07. Barômetro **77**

08. Medindo a concentração de Benzeno **90**

09. Medição de Monóxido de Carbono **108**

10. Medição de Dióxido de Carbono **130**

11. Decibelímetro **152**

12. Dinamômetro **173**

13. Medindo o fluxo de massa **187**

14. Gaussímetro **203**

15. Medindo a concentração de GLP **221**

16. Medindo a concentração de H_2 **240**

17. Higrômetro **258**

18. Medindo a concentração de Ozônio **271**

19. Medição de pH **289**

20. Medição de tensão RMS **307**

21. Medição de temperatura com NTC **321**

22. Medindo a radiação ultravioleta **335**

Capítulo 1
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, onde o objetivo é conectar os sensores listados no sumário, apresentando assim um exemplo de internet das coisas (IoT). É importante que o leitor tenha ciência dos assuntos abordados na obra citada, para que haja um melhor aproveitamento do conteúdo a ser apresentado.

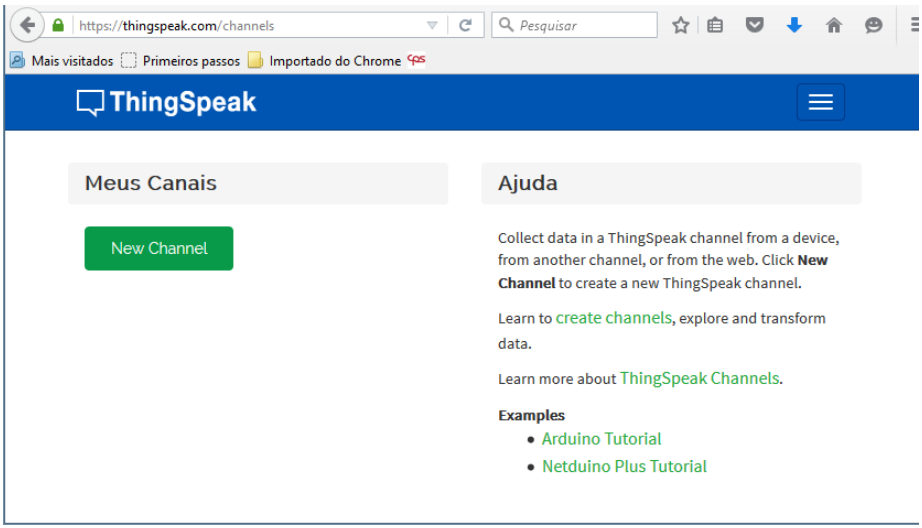
A placa didática utilizada foi a *Blue Pill* STM32F103C8, onde os recursos para testar os circuitos propostos serão conectados através das conexões disponíveis na lateral do kit. Tal kit está à venda no site www.cerne-tec.com.br.

Capítulo 2

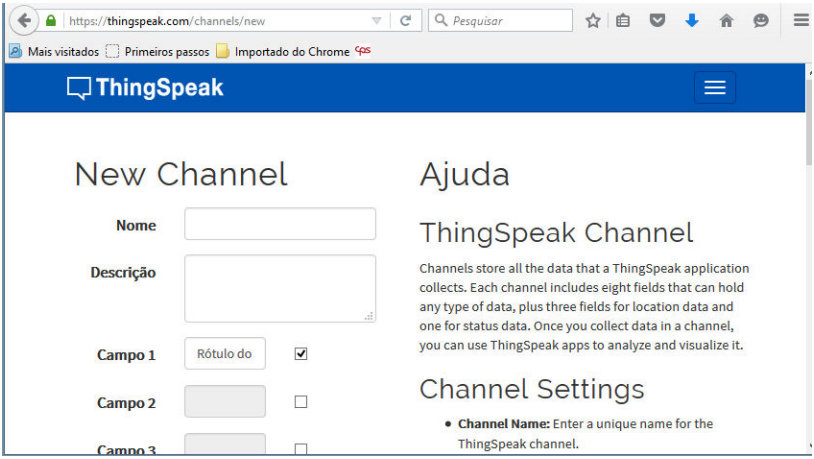
Conhecendo o ThingSpeak

1. Introdução

O ThingSpeak é uma ferramenta gratuita com foco no IoT. 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.

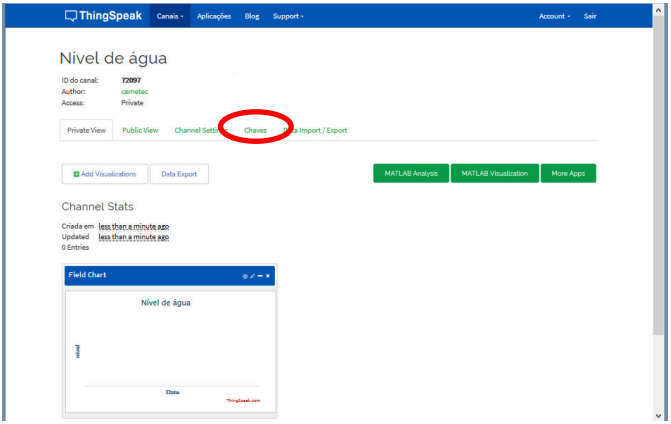


Clique no botão *New Channel* para criar um novo canal. A tela fica como apresentado abaixo.

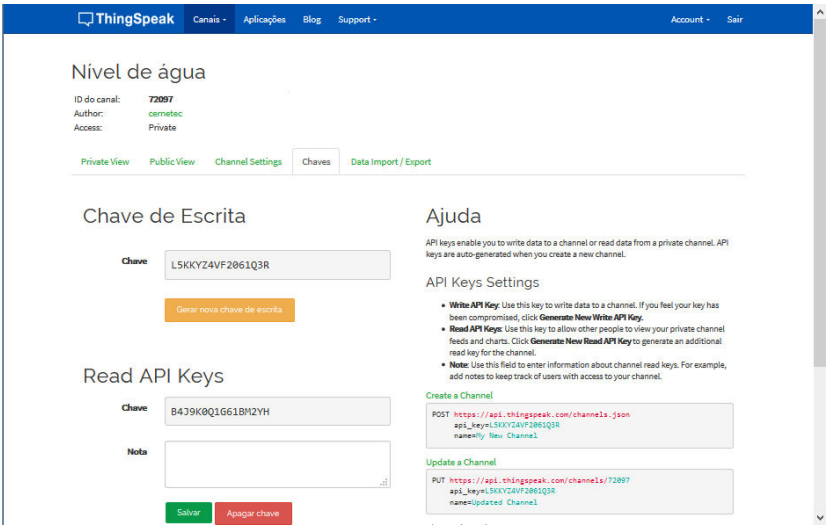


Preencha o campo *Nome* para “Amperímetro” e *Campo 1* para A. Em seguida, pressione o botão *Save Channel*. Observe que pode haver mais campos em uma tela, mas no momento estamos considerando apenas um para a experiência.

Neste momento é criado o canal, no qual a próxima janela já mostra o nome definido assim como o nome do eixo y definido. O eixo x representa o tempo, ou seja, a hora do recebimento de cada atualização que o canal sofreu.



A atualização da página é feita pelo método GET, no qual é passado como parâmetro uma chave criada no mesmo momento do canal. Para visualizá-la, clique na área hachurada na figura anterior.



Observe que neste exemplo a chave é L5KKYZ4VF2061Q3R (atualize conforme o seu canal).

Para enviar dados para o servidor, utilizamos a seguinte string.

```
http://api.thingspeak.com/update?key=[CHAVE]&field1=[VALOR]
```

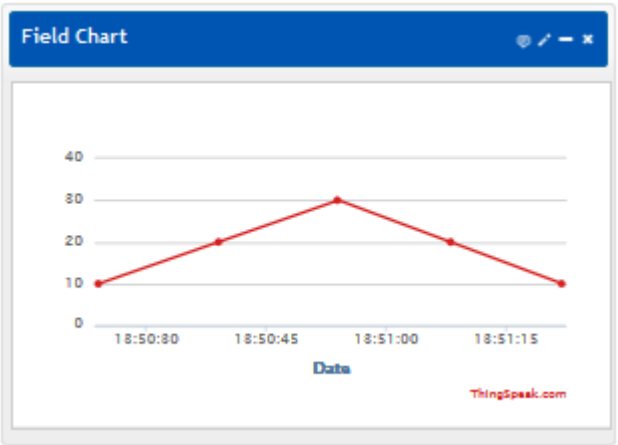
Em CHAVE informa-se o valor obtido através do último passo e no campo *field1* o VALOR passado como parâmetro para a página, de modo a ser armazenado e plotado em seguida.

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 10, 20, 30, 20 e 10.

```
http://api.thingspeak.com/update?key= L5KKYZ4VF2061Q3R&field1=10
http://api.thingspeak.com/update?key=L5KKYZ4VF2061Q3R&field1=20
http://api.thingspeak.com/update?key=L5KKYZ4VF2061Q3R&field1=30
http://api.thingspeak.com/update?key=L5KKYZ4VF2061Q3R&field1=20
http://api.thingspeak.com/update?key=L5KKYZ4VF2061Q3R&field1=10
```

Este mesmo procedimento deve ser feito no STM32 periodicamente, permitindo assim manter o servidor atualizado a respeito da variável medida.

Voltando a tela gráfica (Private View) observe os valores recebidos plotados no gráfico.



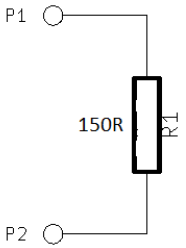
Note que o eixo das ordenadas refere-se ao valor enviado como parâmetro enquanto que o das abscissas o horário que o dado foi recebido. Estes dados podem ser exportados clicando-se nesta mesma página no botão *Data Export*, no qual pode exportar em diversos formatos os dados armazenados.

Capítulo 3

Medição de corrente de 4 a 20 mA

1. Conversor corrente-tensão

Para realizar a medição da corrente faz-se necessário um conversor de corrente em tensão, já que o ARM é capaz de medir apenas tensão pelo conversor ADC. Tal efeito é alcançado com o emprego de um resistor de 150Ω acoplado a entrada de corrente, como poder ser visto na figura a seguir.



Pela 1º Lei de Ohm, a tensão é dada da seguinte forma:

$$V = RI$$

Onde V é a tensão em Volts, R é a resistência em Ω e I á a corrente em Amperes. Sendo assim, com a resistência de 150 Ω e com

a entrada máxima de corrente que é de 20 mA, observa-se a seguinte tensão na entrada analógica do ARM.

$$V = 150 * 0{,}02$$

$$V = 3{,}0V$$

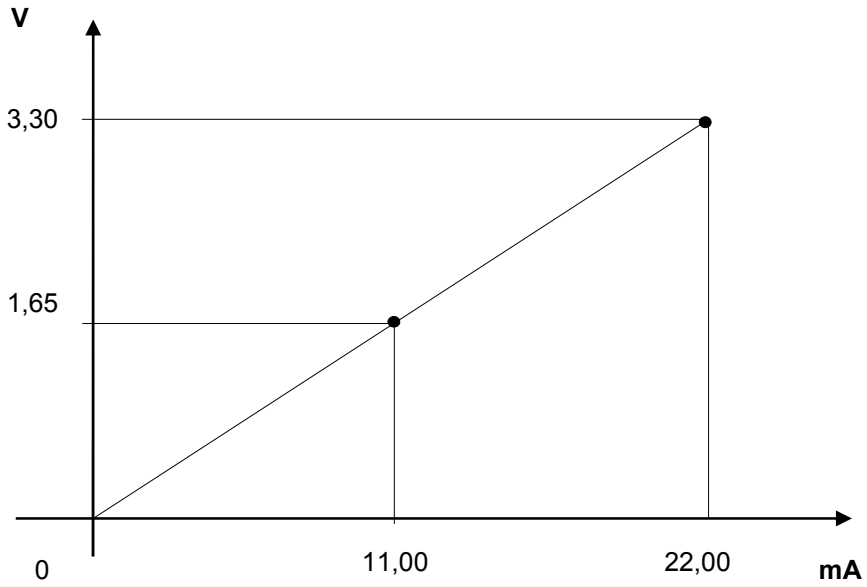
Quando a corrente mínima for medida (4 mA) a seguinte tensão é observada.

$$V = 150 * 0{,}004$$

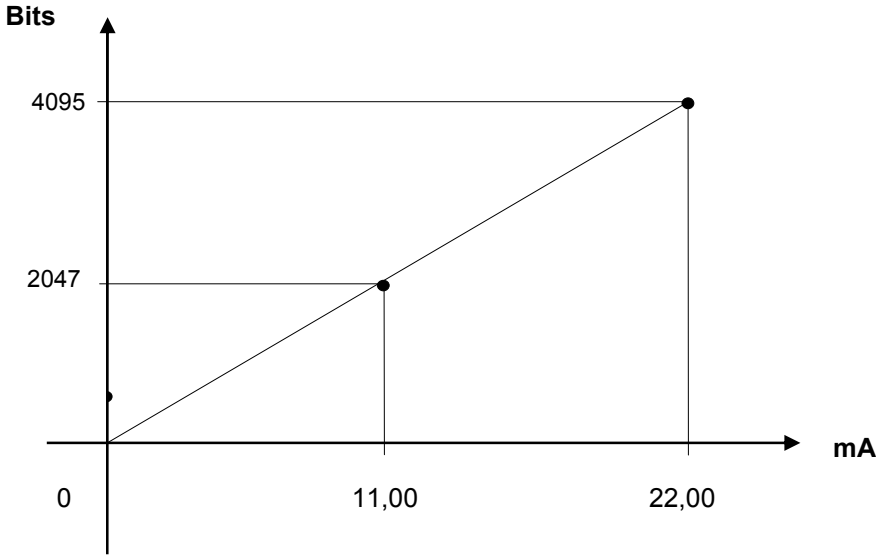
$$V = 0{,}6V$$

É assim que sabe-se a corrente na entrada do ARM, passando primeiro pelo conversor corrente-> tensão e em seguida medindo o mesmo de forma linear.

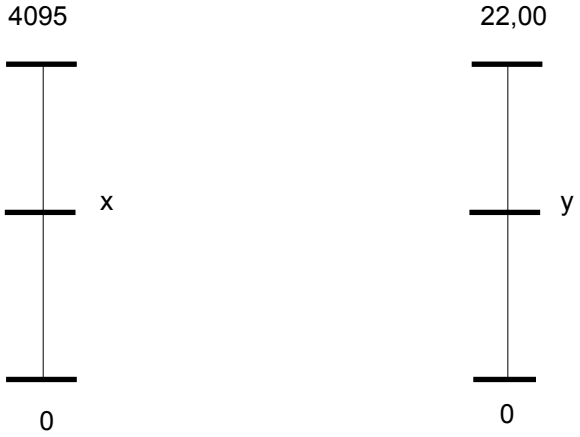
A medição é feita na faixa de 0 a 22,00 mA. A tensão de saída Vout é proporcional a corrente de entrada, onde caso haja a corrente mínima de 0 mA na saída do conversor observa-se a tensão de 0 V. Caso haja a corrente máxima de 22,00 mA a tensão no pino Vout é de 3,3 V e assim linearmente, conforme sugere o gráfico abaixo.



O conversor ADC do ARM é de 12 bits, fornecendo uma resolução de 4096 pontos, pois 2^{12} é igual a este valor. O sensor de corrente está ligado a uma das entradas analógicas do ARM, no qual fornece o resultado binário na faixa de 0 a 4095. Sendo assim é possível modelar o seguinte gráfico, no qual converte a escala de corrente para os bits referentes à conversão ADC.



Desta forma a seguinte relação matemática é observada entre o resultado da conversão ADC e a medição dada em URA.



$$\frac{x-0}{4095-0}=\frac{y-0}{22{,}00-0}$$

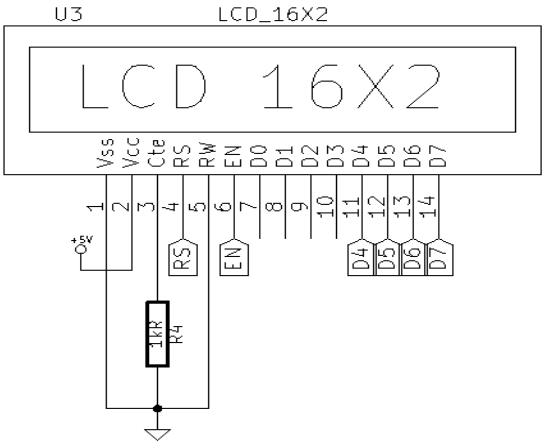
$$\frac{x}{4095}=\frac{y}{22{,}00}$$

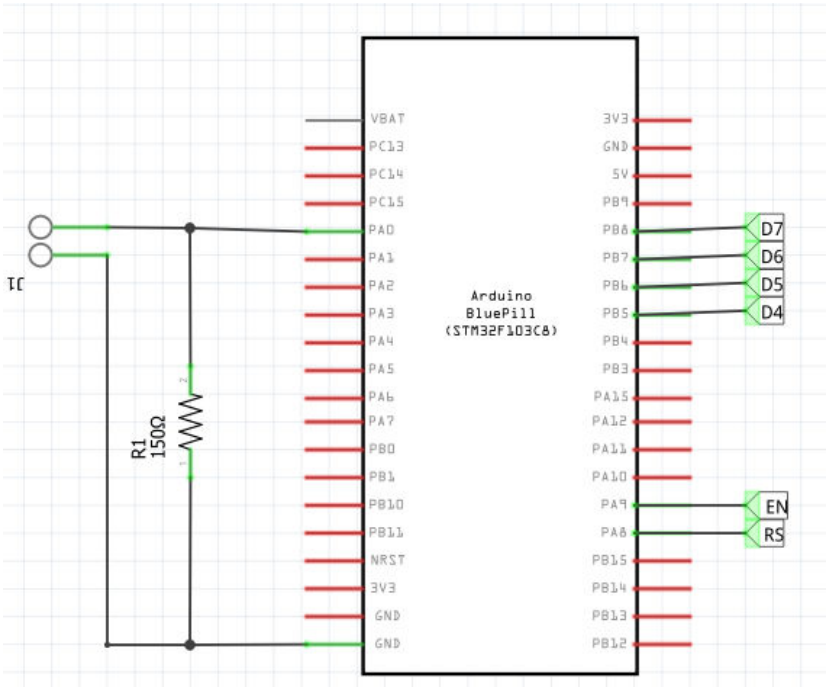
$$y=22{,}00*\frac{x}{4095}$$

Ou seja, após a leitura da entrada analógica, deve-se executar o cálculo matemático acima de forma a obter o valor da corrente e apresentá-la no display.

2. Esquema elétrico

A próxima figura apresenta o esquema elétrico, no qual observa-se o conversor corrente-tensão conectado a entrada analógica do ARM e este conectado a um display LCD.





3. Código fonte

A seguir o código fonte para medir a fonte de corrente de 4 a 20 mA e apresentar no LCD o resultado da medição. Após a leitura do canal analógico é feita a conta matemática apresentada anteriormente para se obter o resultado em mA e atualizar no LCD tal resultado.

```
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *          Cerne Tecnologia           *
 *        www.cerne-tec.com.br         *
 *    Projetos em Arduino -Corrente    *
 *  *  *  *  *  *  *  *  *  *  *  *  */
#include <LiquidCrystal.h>
LiquidCrystal lcd(PA8, PA9, PB5, PB6, PB7, PB8);

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *       Função de inicialização       *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void setup(void)
{
  lcd.begin(16,2);
}
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *           Função de loop            *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void loop(void)
{
  int corrente;
  corrente=(22.00*analogRead(0))/4095;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(corrente,DEC);
  lcd.print(" mA");
  delay(1000);
}
```

4. IoT

Este capítulo apresenta os passos para comunicar o módulo ESP01 que faz uso do CI ESP8266 ao STM32, permitindo assim conectá-lo a Internet e enviar periodicamente para a página construída no capítulo II as informações referentes à variável mensurada.

5. Módulo WiFi ESP8266

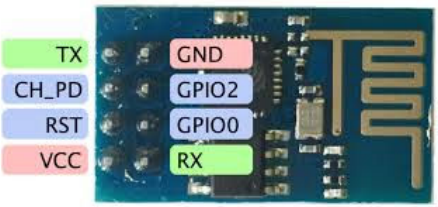
Encontramos no mercado diversos módulos que fazem uso do CI ESP8266, porém nesta literatura o modelo ESP01 foi considerado, cuja aparência é como o da foto abaixo.



Fonte: aliexpress.com

Este módulo consome em média 200 mA quando em utilização e foi projetado para aplicações IoT, possuindo baixo custo e facilidade de aquisição no mercado.

A sua conexão é feita com o STM32 através do canal de UART, pelos pinos TX e RX. A próxima figura mostra a descrição dos 8 pinos disponíveis em seu conector de acesso.



Fonte: fabacademy.org

A conexão com o STM32 é feita através dos pinos TX e RX. O pino TX do ESP01 fica ligado diretamente ao de RX do MCU enquanto o TX do MCU conectado ao RX do módulo.

Este módulo por padrão vem configurado para comunicar a 115200 bps. Para a troca de dados entre o processador e o ESP01, há uma lista de comandos AT, no qual as utilizadas neste exemplo são explanadas a seguir.

6. Comandos AT

O comando AT+CWMODE=1 configura o ESP8266 para acesso padrão a um roteador, que permita o acesso a internet. Outros parâmetros, como 2 ou 3 permitem que ele também possa ser um access point (AP).

O comando AT+CWJAP=”SSID”,”SENHA” permite conectar a uma rede WiFi. Considerando o SSID *cernetec* e a senha *iot* este comando fica da seguinte forma: AT+CWJAP=”cernetec”,”iot” (atualize tais parâmetros conforme a rede a ser conectada).

Através do comando AT+CIPSTART=”TCP”,”IP”,PORTA é estabelecido um soquete com algum IP e PORTA da rede. Como o IP do ThingSpeak é fixo e a porta utilizada é a 80, este parâmetro fica como: AT+CIPSTART=”TCP”,”184.106.153.149”,80.

Para enviar dados para o servidor, faz-se uso do comando AT+CIPSEND=TAMANHO, onde em TAMANHO informa-se a quantidade de caracteres a serem enviados. A string a ser enviada obedece ao formato a seguir.

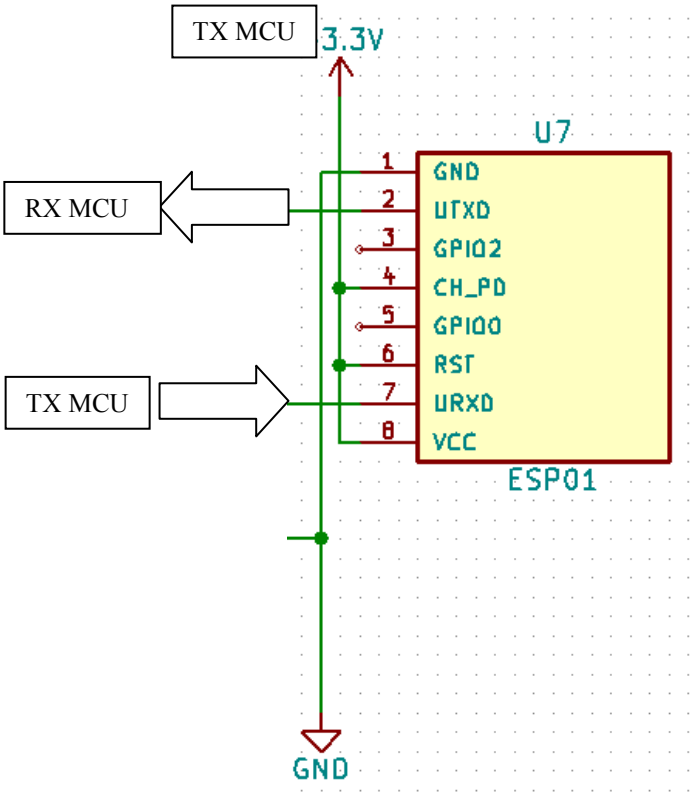
```
http://api.thingspeak.com/update?key=[CHAVE]&field1=[VALOR]
```

Onde o campo CHAVE é obtido conforme apresentado no capítulo II e o campo VALOR é referente à variável mensurada.

Após o envio do pacote, a porta pode ser fechada pelo comando AT+CIPCLOSE.

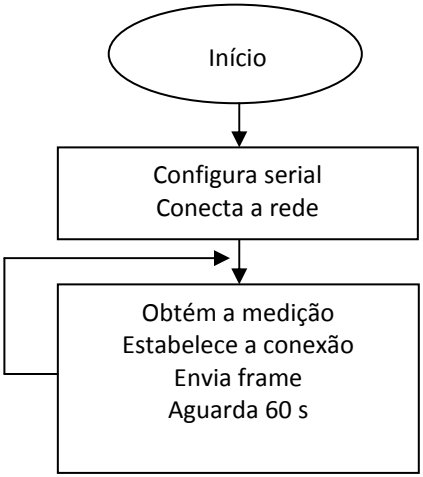
7. Esquema elétrico

Mantenha o circuito do sensor estudado anteriormente e adicione o módulo ESP01 ao processador.



8. Fluxograma

O fluxograma deste experimento pode ser observado a seguir.



9. Código fonte

O código fonte deste experimento pode ser verificado a seguir, seguindo todos os comentários feitos até então.

```
void setup()
{
  Serial.begin(115200);
  Serial.println("AT+CWMODE=1");
  delay(5000);
  Serial.println("AT+CWJAP=\"cernetec\",\"iot\"");
  delay(5000);
}


void loop()
{
  int adc=(22.00*analogRead(0))/4095;

  Serial.println("AT+CIPSTART=\"TCP\",\"184.106.153.149\",80");
  delay(5000);
  String frame="GET /update?api_key=L5KKYZ4VF2061Q3R&field1="
                              +String(adc)+"\r\n\r\n";
  Serial.print("AT+CIPSEND=");
  Serial.println(frame.length());
```

```
  delay(5000);
  Serial.print(frame);
  delay(60000);
}
```

No bloco setup() é feita a habilitação da porta para comunicar a 115200 bps e a conexão com a rede WiFi, onde informa-se o SSID e a senha.

No bloco loop(), a cada minuto uma nova aquisição de dados é feita e desta forma estabelecida a conexão com o servidor, enviando a medição realizada usando-se o método GET com a CHAVE obtida pela plataforma IoT e o campo field1 tendo como parâmetro a variável mensurada. Em seguida, aguarda-se 1 minuto para nova aquisição.

Capítulo 4

Medindo Álcool em ppm

1. Sensor TGS 822

O sensor TGS 822, desenvolvido pela Figaro Sensor (www.figarosensor.com), permite medir uma série de vapores orgânicos como Metano, Monóxido de Carbono, Isobutano, Hexano, Benzeno, Álcool e Acetona. Cada um destas soluções, apresenta uma curva diferente de concentração em ppm (Parte Por Milhão), na faixa de 50 a 5000 ppm.

Este sensor apresenta alta sensibilidade, estabilidade, baixo custo e sua interface elétrica é relativamente simples. Suas principais aplicações são na detecção de álcool (podendo ser usado como bafômetro), detecção de vazamento de gás, detectores de solvente em fábrica etc. O elemento sensor é composto de Dióxido de Estanho (SnO_2), que varia sua condutividade de acordo com a concentração do gás presente em seu ambiente de instalação. Com o aumento da concentração de gás no ar, sua condutividade aumenta, em que desta forma um ARM pode mensurar a concentração de determinado gás neste ambiente.

A próxima figura mostra a resposta de sensibilidade do sensor, de acordo com a concentração em ppm do solvente orgânico que estiver sendo medido pelo mesmo.

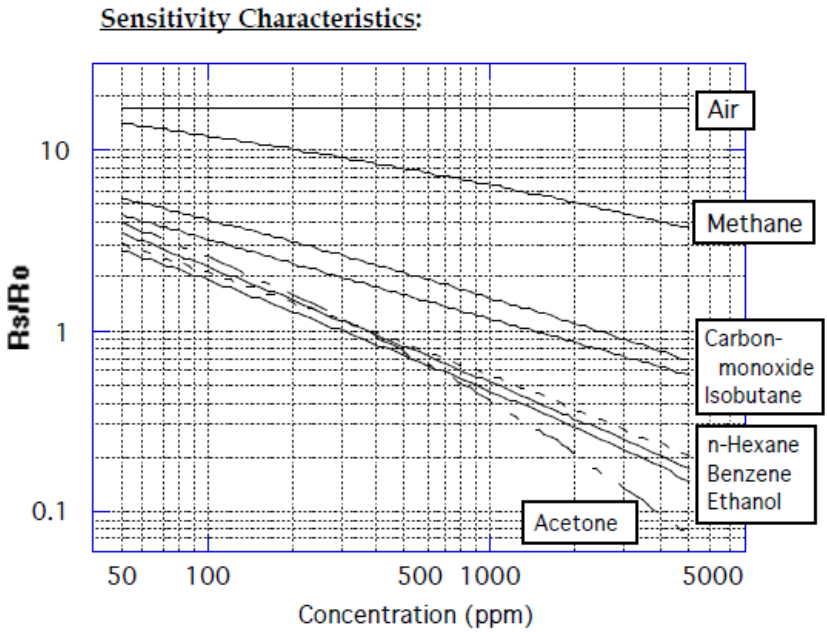


Figura 2.1: Resposta de acordo com a concentração de solvente
Fonte: TGS 822 Datasheet

A resposta do sensor varia de acordo com a temperatura e umidade relativa (R.H) do ambiente em que o mesmo estiver instalado, de acordo com a figura a seguir fornecida pelo próprio datasheet do componente.

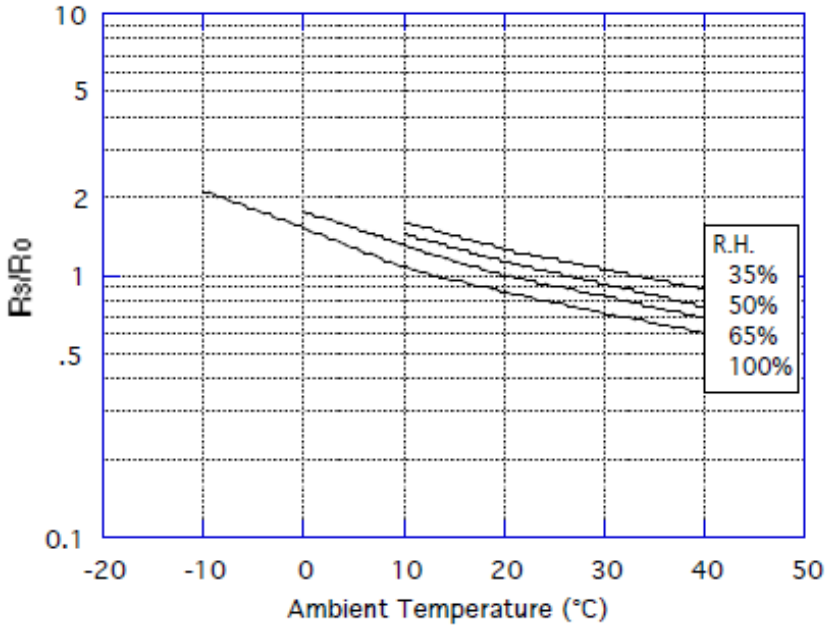


Figura 2.2: Influência da variação de umidade e temperatura
Fonte: TGS 822 Datasheet

Em determinados projetos que necessitem de maior precisão na resposta da medição de gases, pode ser necessário haver um sensor de umidade e temperatura, de modo a ajustar a resposta do TGS 822 de acordo com estas variações.

Este sensor é composto de 6 pinos, conforme pode ser visto na figura a seguir.

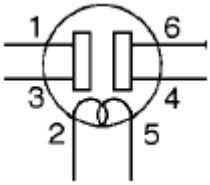


Figura 2.3: Pinagem do TGS 822
Fonte: TGS 822 Datasheet

A função da cada um dos pinos está descrita a seguir.

Pino	Função	Descrição
1 e 3	V_C	Alimentação positiva. Máximo de 24 Vdc.
2	V_H	Tensão de aquecimento. Tensão de 5V DC ou AC
5	GND	Terra da fonte
4 e 6	V_{RL}	Tensão variável de acordo com a concentração de gases

Fonte: TGS 822 Datasheet

Normalmente, os pinos 1, 2 e 3 ficam ligados juntos na tensão de 5 V. A saída V_{RL} fica acoplada a um resistor de carga (R_L), pois a resistência interna do sensor, ou seja, sua condutividade, varia segundo esta concentração, permitindo ao ARM mensurá-la.

A próxima figura demonstra um circuito típico a ser montado para uso deste sensor.

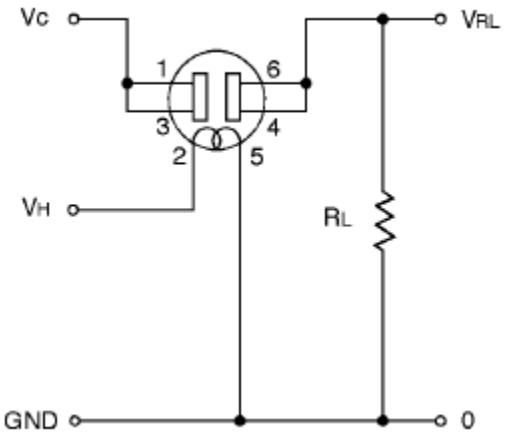


Figura 2.4: Circuito típico com TGS 822
Fonte: TGS 822 Datasheet

Um valor usual para R_L é de 10 kΩ. Observe pela curva do gráfico 2.1 que a concentração em ppm é encontrada de acordo com a relação R_S/R_0. O valor de R_S é definido como a resistência do sensor de acordo com a concentração de gás ambiente enquanto que R_0 para uma concentração de 50 ppm de Álcool, tipicamente 5.000Ω. O valor de R_S será encontrado segundo a fórmula a seguir.

$$R_S = \left(\frac{V_C}{V_{RL}} - 1 \right).R_L$$

Considerando V_C=5V e R_L=10.000Ω, o cálculo fica da seguinte forma.

$$R_S = \left(\frac{5}{V_{RL}} - 1 \right).10000$$

O valor de V_{RL} é encontrado através da leitura do conversor analógico do ARM (CAD), estando na faixa de 0 a 3,3 V. Após encontrar R_S, basta encontrar a relação deste valor com R_0 (que vale 5.000Ω) para em seguida encontrar a concentração em ppm.

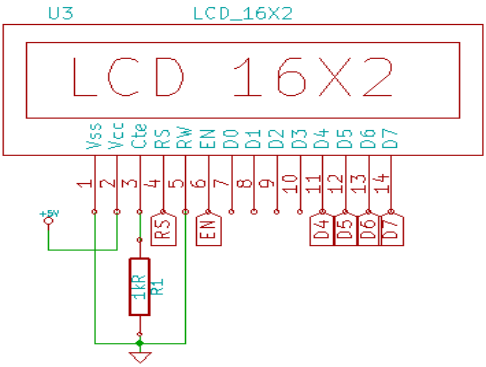
Chamando de *x* a relação entre R_S/R_0 e *ppm* a concentração, a função a seguir descreve como é encontrada a concentração.

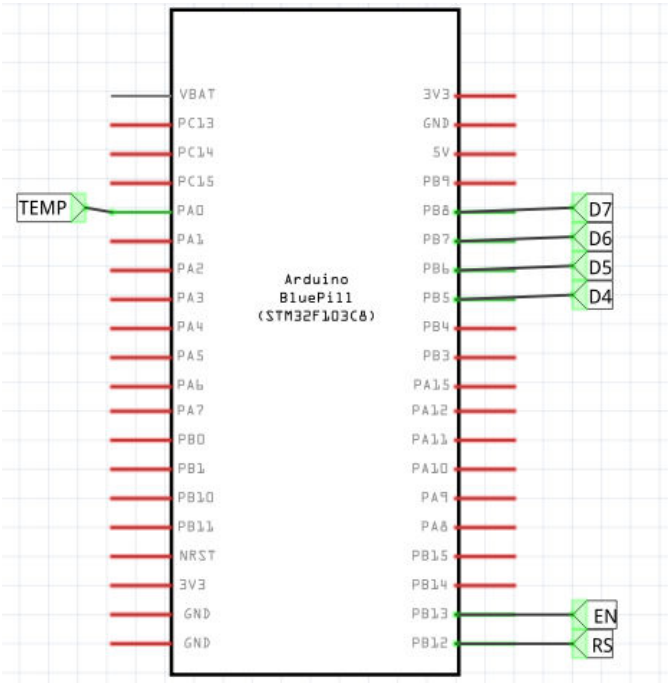
$$ppm = 169.(\log(x))^4 - 147.(\log(x))^3 + 207.(\log(x))^2 - 496.\log(x) + 332$$

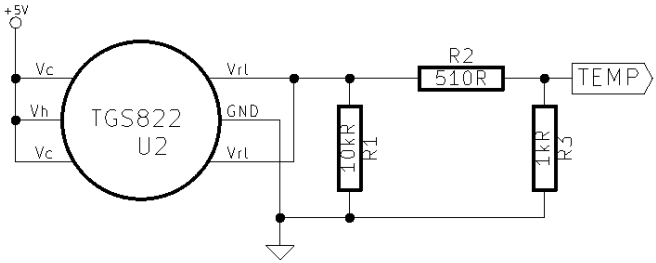
A descrição de como esta função foi encontrada está disponível no artigo de Jiana Shin, onde o link está disponível na seção Referências desta literatura.

2. Esquema elétrico

O esquema elétrico para este exemplo está apresentado a seguir, onde observa-se o display LCD que mostra o valor da concentração em ppm de Álcool e o sensor que é a fonte de dados para o ARM.

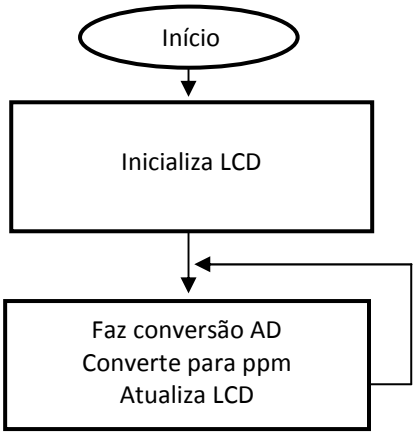

U3
LCD_16X2
LCD 16X2
Vss
Vcc
Cte
RS
RW
EN
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
+5V
RS
EN
D4
D5
D6
D7
1KR
R1


TEMP
VBAT
PC13
PC14
PC15
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB10
PB11
NRST
3V3
GND
GND
Arduino
BluePill
(STM32F103C8)
3V3
GND
5V
PB9
PB8
PB7
PB6
PB5
PB4
PB3
PA15
PA12
PA11
PA10
PA9
PA8
PB15
PB14
PB13
PB12
D7
D6
D5
D4
EN
RS



3. Fluxograma

O fluxograma que rege o funcionamento do programa está apresentado a seguir. Observe que inicialmente é feita a inicialização do LCD e no loop principal a leitura contínua da entrada analógica e em seguida a conversão para ppm, para que finalmente o LCD seja atualizado.



4. Código fonte

A seguir o código fonte em ARM que mede a concentração em ppm do Álcool em determinado meio através do sensor TGS 822 e apresenta no LCD o resultado da medição em ppm. Após a leitura analógica do canal 0 encontra-se a relação R_S/R_0 para se aplicar a função apresentada, passando como parâmetro esta relação. O retorno da função é a concentração em ppm para que em seguida, este resultado seja apresentado no LCD.

```
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *          Cerne Tecnologia          *
 *        www.cerne-tec.com.br        *
 *    Projetos em Arduino - Álcool    *
 *  *  *  *  *  *  *  *  *  *  *  *  */
#include <LiquidCrystal.h>
LiquidCrystal lcd(PB12,PB13,PB5,PB6,PB7,PB8);

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *      Função de inicialização       *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void setup(void)
{
  lcd.begin(16,2);
}

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *          Função de loop            *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void loop(void)
{
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print(le_ppm_alcool());
```

```
  lcd.print(" ppm");
  delay(1000);
}

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *  Função para ler ppm do ambiente  *
 *  *  *  *  *  *  *  *  *  *  *  *  */
float le_ppm_alcool(void)
{
  float ppm,ad,vrl,rs,rs_ro,x;

  ad = analogRead(0);      //Lê AD de 0 a 1023
  vrl=(ad*3.3)/4095;       //Encontra Vrl
  rs=((3.3/vrl)-1)*10000; //Encontra Rs
  rs_ro=rs/5000;           //Encontra relação Rs/Ro
  x=log(rs_ro);            //Acha o log
  ppm=169*(x*x*x*x)-147*(x*x*x)+207*(x*x)-496*x+332;
                           //Equação para achar ppm
  return(ppm);             //Concentração em ppm
}
```

5. IoT

Este capítulo apresenta os passos para comunicar o módulo ESP01 que faz uso do CI ESP8266 ao STM32, permitindo assim conectá-lo a Internet e enviar periodicamente para a página construída no capítulo II as informações referentes à variável mensurada.

6. Módulo WiFi ESP8266

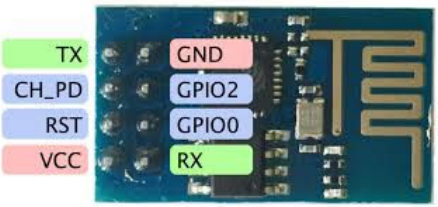
Encontramos no mercado diversos módulos que fazem uso do CI ESP8266, porém nesta literatura o modelo ESP01 foi considerado, cuja aparência é como o da foto abaixo.



Fonte: aliexpress.com

Este módulo consome em média 200 mA quando em utilização e foi projetado para aplicações IoT, possuindo baixo custo e facilidade de aquisição no mercado.

A sua conexão é feita com o STM32 através do canal de UART, pelos pinos TX e RX. A próxima figura mostra a descrição dos 8 pinos disponíveis em seu conector de acesso.



Fonte: fabacademy.org