Vitor Amadeu Souza

IoT

para monitoramento de

Sensores

Com 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.

Abril 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.

"Aqueles que, como eu, foram humildes pioneiros da conquista do ar, pensavam mais em criar novos meios de expansão pacífica entre os povos do que lhes fornecer novas armas."

Alberto Santos Dumont

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-tec.com.br

Sumário

01. Metodologia de desenvolvimento **7**

02. Conhecendo o ThingSpeak **8**

03. Medindo a concentração de Amônia **13**

04. Amperímetro **30**

05. Barômetro **43**

06. Medindo a concentração de Benzeno **53**

07. Célula de Carga **69**

08. Medição de Monóxido de Carbono **83**

09. Medição de Dióxido de Carbono **101**

10. Decibelímetro **120**

11. Dinamômetro **138**

12. Medindo Etanol em ppm **148**

13. Medindo o fluxo de massa **162**

14. Gaussímetro **176**

15. Medindo a concentração de GLP **192**

16. Medindo a concentração de H_2 **208**

17. Higrômetro **224**

18. Medindo a concentração de Ozônio **234**

19. Medição de pH **250**

20. Termômetro **265**

21. Medindo a radiação ultravioleta **276**

22. Voltímetro **293**

Capítulo 1

Metodologia de desenvolvimento

1. Introdução

A proposta deste livro é explorar a Internet das Coisas ou *Internet of Things (IoT)* aplicado ao monitoramento dos sensores listados no sumário da obra através de um exemplo prático que permita conectar o ESP32 com base no módulo NODEMCU-32S a internet, atualizando periodicamente uma página de modo a visualizar em modo gráfico o seu comportamento ao longo do tempo. Desta forma, será possível conectar o ESP32 a rede para o monitoramento citado, implantando assim o conceito IoT.

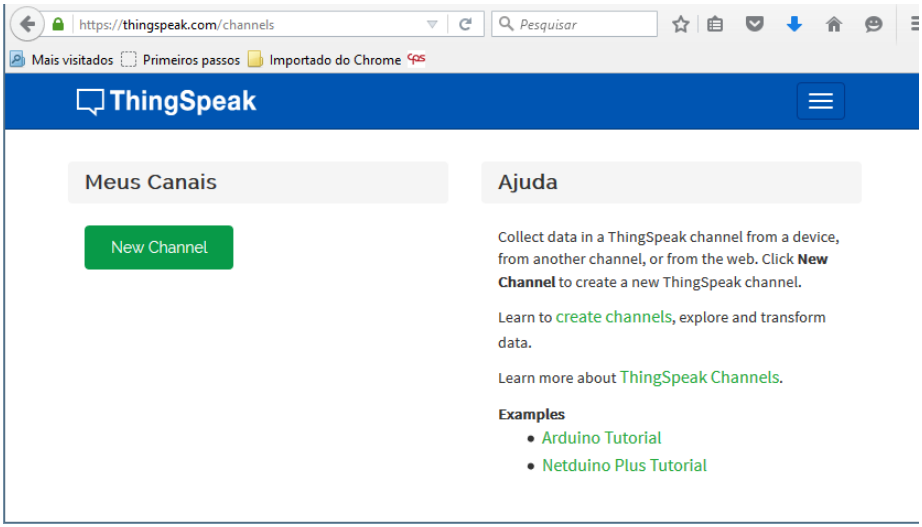
A parte Web foi desenvolvida usando ThingSpeak (www.thingspeak.com), ferramenta que possibilita o recebimento, armazenamento e plotagem de dados de maneira gratuita, tendo sido desenvolvida para aplicações IoT. A programação do ESP32 foi baseada no Arduino.

Capítulo 2

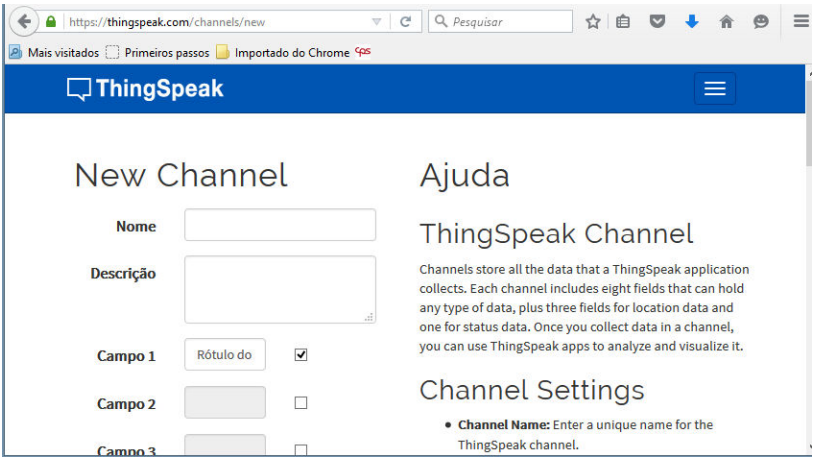
Conhecendo o ThingSpeak

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.

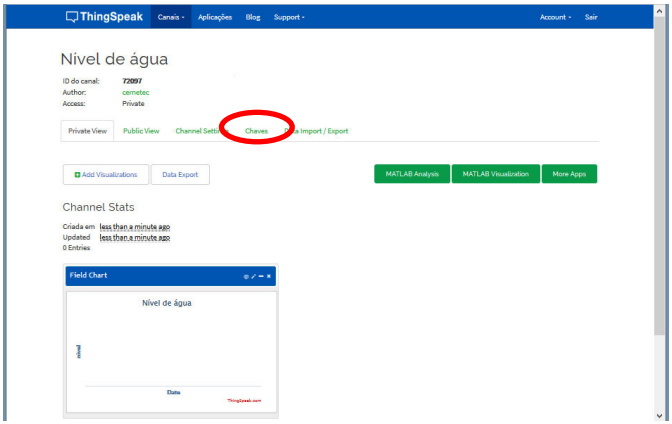


Clique no botão *New Channel* para criar um novo canal. A tela fica como apresentado abaixo.

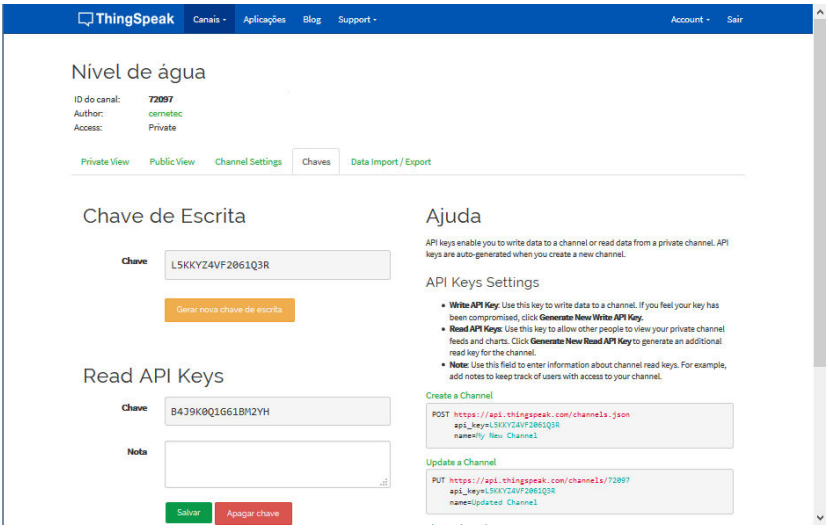


Preencha o campo *Nome* para “Amônia” e *Campo 1* para ppm. Em seguida, pressione o botão *Save Channel*. Observe que pode haver mais campos em uma tela, mas no momento estamos considerando apenas um para a experiência.

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]
```

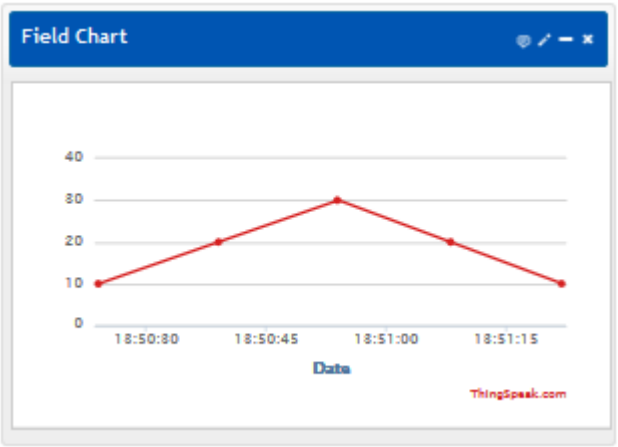
Em CHAVE informa-se o valor obtido através do último passo e no campo *field1* o VALOR passado como parâmetro para a página, de modo a ser armazenado e plotado em seguida.

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 NODEMCU-32S 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

Medindo a concentração de Amônia

1. Efeitos da Amônia ao ser humano

O gás amônia ou amônia, cuja fórmula molecular é NH_3 está disposto geometricamente da seguinte forma.

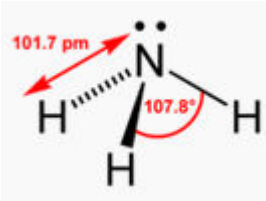


Figura 3.1: Representação da amônia
Fonte: http://pt.wikipedia.org/wiki/Am%C3%B4nia

É um gás tóxico em temperatura ambiente e pressão atmosférica e na presença da umidade torna-se corrosivo. É um gás inflamável, incolor, com odor irritante e possui alta solubilidade em meio aquoso, como a água. Possui diversas aplicações, como agente refrigerante industrial e como fertilizante.

Em altas concentrações, produz tosses fortes devido a sua ação nas vias respiratórias além de causar queimadura nos olhos. Além

disso, a vítima sofrerá irritação dos pulmões podendo evoluir a um edema pulmonar em pouco tempo, sendo necessária a rápida remoção para não agravar tal situação. Em caso de ingestão, provocará corrosão da boca, garganta e estômago.

Segundo o anexo 11 da NR 15 (Norma Regulamentadora 15), a concentração máxima para uma exposição semanal de até 48 horas é de 20 ppm. Caso este limite seja extrapolado, o ambiente é considerado medianamente insalubre.

Os principais riscos oferecidos pela Amônia são ligados a saúde, conforme mostra a classificação de rotulagem da figura a seguir.

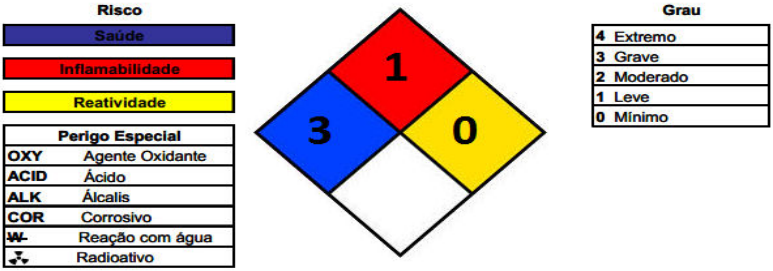


Figura 3.2: Riscos associados a Amônia
Fonte: http://pt.wikipedia.org/wiki/Am%C3%B4nia

Note pelo quadro que os riscos associados a reatividade são mínimos (0), enquanto que o de inflamabilidade é leve (1) e saúde grave (3).

Face ao exposto, torna-se importante dispor de equipamentos que meçam a concentração de Amônia de maneira contínua e possam alertar os seus usuários caso níveis críticos sejam atingidos.

Neste capítulo, será desenvolvido um dispositivo capaz de medir a concentração de Amônia no ambiente e que envia pela serial tal valor em ppm (parte por milhão).

2. Sensor MQ-135

O sensor MQ-135, desenvolvido pela Hanwei Eletronics (www.hwsensor.com), permite medir a concentração em ppm da Amônia (NH_3) com alta repetibilidade e seletividade. A próxima figura mostra o encapsulamento deste sensor.



Figura 3.4: Encapsulamento do MQ-135
Fonte: https://www.sparkfun.com/products/10916

O MQ-135 opera como um resistor variável, que acompanha a variação de Amônia no ambiente. A sua resposta é linear e quanto

maior a concentração de Amônia, menor a resistência de saída. A faixa de concentração aceita pelo sensor é de 10 a 300 ppm.

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

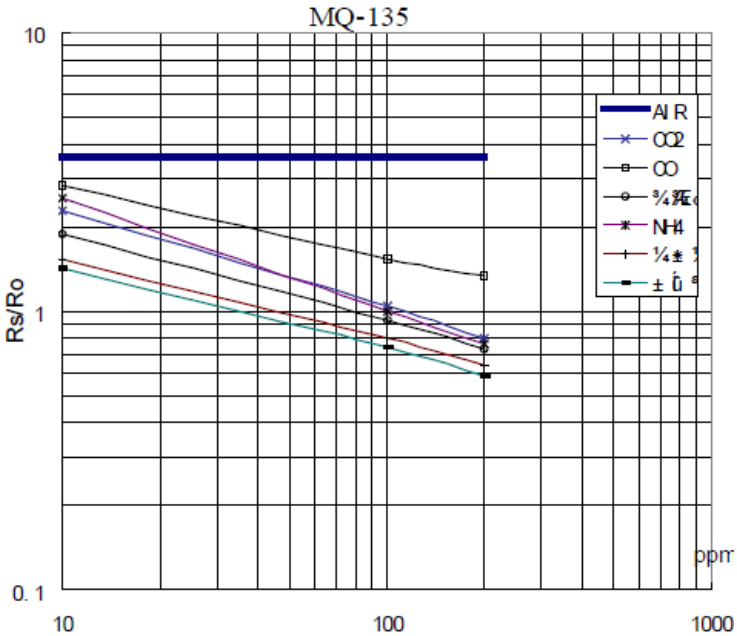


Figura 3.5: Resposta de acordo com a concentração do meio
Fonte: MQ-135 Datasheet

Observe que em 100 ppm a relação entre a resistência de saída e a de referência (Rs/Ro) é igual a 1. A curva é linear, facilitando assim a

detecção do Arduino. Além de Amônia, o sensor é sensível a outros gases como álcool e Benzeno.

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

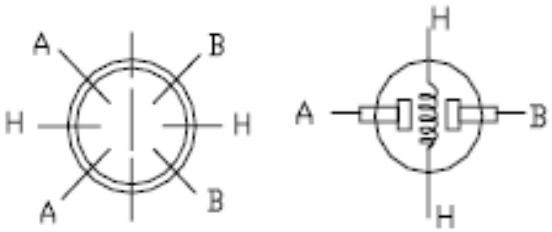


Figura 3.6: Pinagem do MQ-135
Fonte: MQ-135 Datasheet

A próxima figura mostra o circuito típico a ser montado para obter o valor da concentração de Amônia.

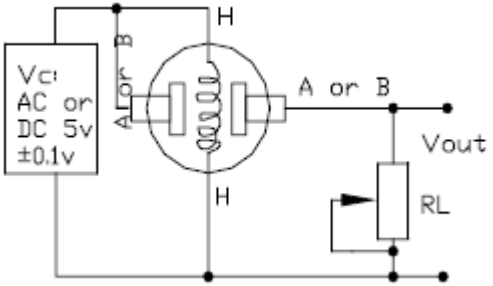


Figura 3.7: Circuito típico
Fonte: MQ-135 Datasheet

Nos pinos identificados como H o sensor será conectado a 5V e GND. No pino A em 5V e em B a saída proporcional a concentração de Amônia do ambiente. O resistor RL tipicamente é de 20 kΩ para a medição de Amônia, no entanto, pode ser recomendado utilizar um trimpot para calibração, já que este valor pode variar de acordo com a temperatura e umidade relativa do ar.

A próxima figura apresenta o sensor alimentado e com o resistor RL conectado na saída de medição.

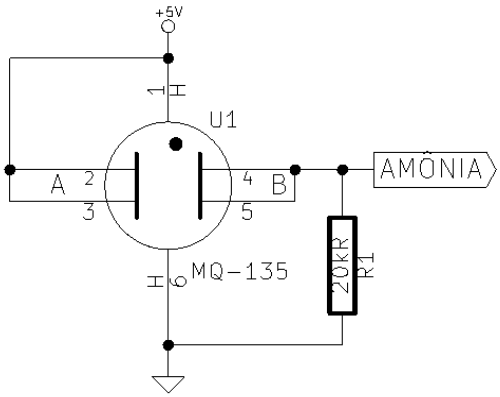


Figura 3.8: Circuito elétrico
Fonte: MQ-135 Datasheet

Com este circuito, na concentração máxima, ou seja, 300 ppm, haverá uma tensão de 5V na entrada analógica do Arduino enquanto que para a concentração mínima, 0 ppm, haverá 0V.

A curva que relaciona a tensão de saída do sensor de Amônia em ppm terá a característica apresentada a seguir.

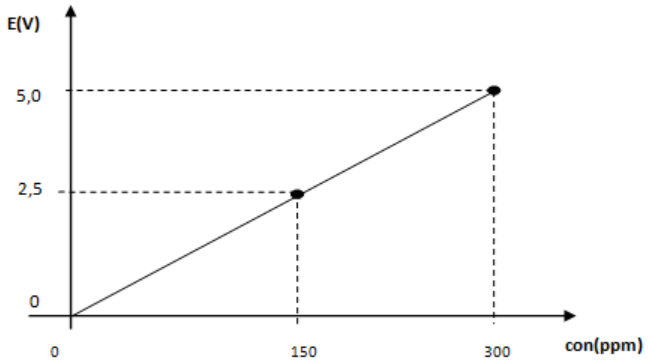


Figura 3.9: Resposta do sensor em Volts

Um divisor resistivo deve ser montado na saída do sensor de modo a fornecer de 0 V a 3,3V a tensão analógica na entrada do processador. O conversor ADC do ESP32 é de 12 bits, o que sugere que seja capaz de obter uma resolução de 4096 pontos, já que 2^{12} é igual a este valor. O sensor estará ligado na entrada analógica, no qual fornece como resultado um valor binário na faixa de 0 a 4095 de acordo com a tensão de 0 a 3,3 V. Sendo assim, pode-se modelar o seguinte gráfico, no qual converte a escala de concentração (ppm) para os bits referentes a conversão AD.

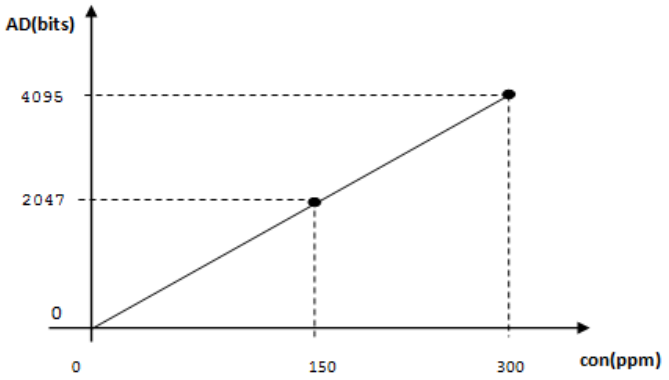
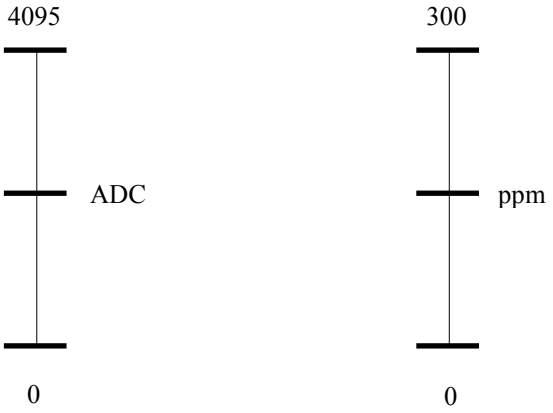


Figura 3.10: Equivalência com resposta do CAD

Desta forma, há a seguinte relação matemática entre o resultado da conversão ADC e medição da concentração de Amônia.



$$\frac{ADC - 0}{4095 - 0} = \frac{ppm - 0}{300 - 0}$$

$$\frac{ADC}{4095} = \frac{ppm}{300}$$

$$ppm = \frac{300.ADC}{4095}$$

Figura 3.11: Cálculo matemático para achar o ppm

Ou seja, após a leitura da entrada analógica, deve-se executar o cálculo matemático acima de forma a obter o valor da concentração em ppm de Amônia.

3. Esquema elétrico

O esquema elétrico para este exemplo está apresentado a seguir, onde observa-se o sensor que é a fonte de dados para o ESP32.

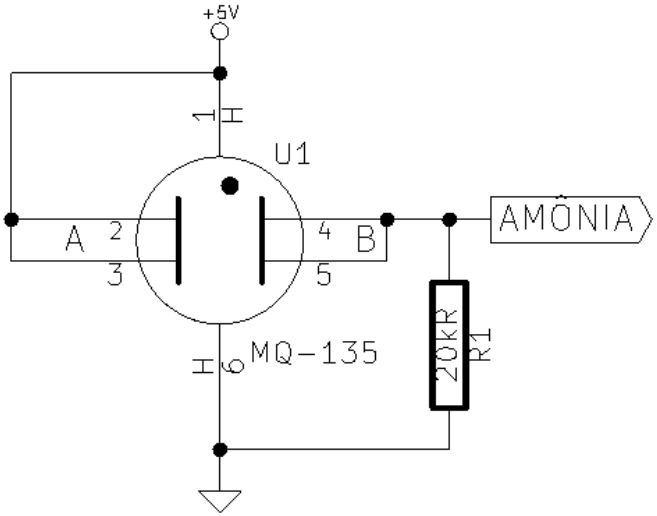


Figura 3.12: Esquema elétrico completo

4. Fluxograma

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

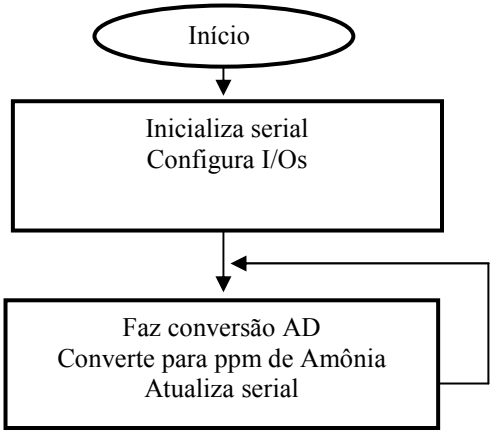


Figura 3.13: Fluxograma

5. Código fonte

A seguir o código fonte que mede a concentração em ppm de Amônia em determinado meio, através do sensor MQ-135 e atualiza a porta serial com o resultado da medição em ppm. Após a leitura do canal analógico aplica-se a relação matemática mostrada em 3.11, encontra-se a concentração de Amônia em ppm. O retorno da função é a concentração em ppm.

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

#define _ADC0      36
```

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

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *           Função de loop            *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void loop(void)
{
  Serial.println(String(le_ppm_NH3()) + " ppm");
  delay(1000);
}

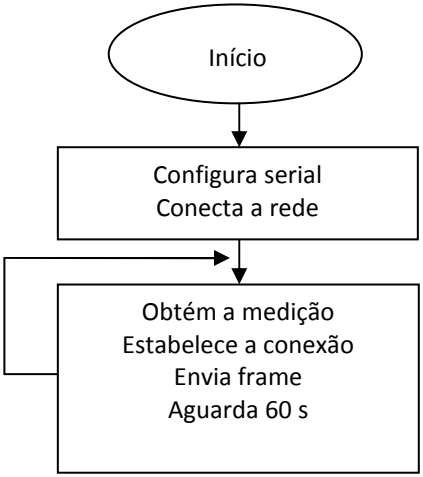
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *  Função para ler Amônia em ppm     *
 *  *  *  *  *  *  *  *  *  *  *  *  */
float le_ppm_NH3(void)
{
  float ppm,ad;
  ad = analogRead(_ADC0);
  ppm=(300.0*ad)/4095.0;
  return(ppm);
}
```

6. IoT

Este capítulo apresenta os passos para comunicar o módulo NODEMCU-32S que faz uso do CI ESP32 a internet, permitindo assim enviar periodicamente para a página construída anteriormente as informações referentes à variável mensurada.

7. Fluxograma

O fluxograma deste experimento pode ser observado a seguir.



8. Código fonte

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

```
#include <WiFi.h>

const char* ssid     = "Cerne";
const char* password = "Tecnologia";
const char* host     = "184.106.153.149";
const int httpPort   = 80;

#define _ADC0 36

void setup()
{
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
    delay(500);
}

void loop()
{
  WiFiClient client;
   if (client.connect(host, httpPort))
  {
    String url = "GET /update?api_key=L5KKYZ4VF2061Q3R
              &field1="+String(le_adc());

    client.print(url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");

    unsigned long timeout = millis();
    while (client.available() == 0)
    {
      if (millis() - timeout > 5000)
      {
        client.stop();
        return;
      }
    }

    while(client.available())
     String line= client.readStringUntil('\r');

    delay(60000);
  }
}

 float le_adc(void)
 {
   float ppm,ad;
   ad = analogRead(_ADC0);
   ppm=(300.0*ad)/4095.0;
   return(ppm);
```

```
}
```

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

```
#include <WiFi.h>

const char* ssid     = "Cerne";
const char* password = "Tecnologia";
const char* host     = "184.106.153.149";
const int httpPort   = 80;
```

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, usando o IP fixo do ThingSpeak (184.106.153.149).

```
if (client.connect(host, httpPort))
  {
    String url = "GET /update?api_key=L5KKYZ4VF2061Q3R
               &field1="+String(le_adc());

    client.print(url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");

    unsigned long timeout = millis();
    while (client.available() == 0)
    {
      if (millis() - timeout > 5000)
      {
        client.stop();
        return;
      }
    }

    while(client.available())
     String line= client.readStringUntil('\r');

    delay(60000);
  }
}
```

Observe a presença da chave obtida do capítulo II na string que fornece a conexão com o servidor IoT e o campo 1 (field1) carregado com o valor da variável mensurada. Após o envio aguarda-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.

Capítulo 4

Amperímetro

1. Introdução

Um amperímetro é um instrumento capaz de medir a corrente de um circuito e apresentá-la em uma interface homem-máquina, que pode ser do tipo display de 7 segmentos, LCD, display gráfico, etc. Estes instrumentos possuem uma baixa-impedância de entrada, para que assim não interfiram no funcionamento do circuito mensurado e podem medir ampla faixa de corrente a nível de corrente contínua (DC) ou alternada (AC). Neste capítulo, será desenvolvido um amperímetro DC utilizando uma entrada analógica do ESP32, onde o resultado mensurado é apresentado em um display LCD, conforme sugere o modelo apresentado a seguir.

