Vitor Amadeu Souza

Desenvolvimento de projetos de monitoramento de

Sensores

através do

Thinger.io

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

"Ensina-me, Senhor, o teu caminho, e andarei na tua verdade; dispõe o meu coração para temer o teu nome."

Sl 86:10

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. Thinger.io **8**

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

04. Amperímetro **37**

05. Barômetro **58**

06. Medindo a concentração de Benzeno **76**

07. Célula de Carga **99**

08. Medição de Monóxido de Carbono **121**

09. Medição de Dióxido de Carbono **147**

10. Decibelímetro **174**

11. Dinamômetro **200**

12. Medindo Etanol em ppm **218**

13. Medindo o fluxo de massa **239**

14. Gaussímetro **261**

15. Medindo a concentração de GLP **285**

16. Medindo a concentração de H_2 **309**

17. Higrômetro **332**

18. Medindo a concentração de Ozônio **350**

19. Medição de pH **374**

20. Termômetro **397**

21. Medindo a radiação ultravioleta **416**

22. Voltímetro **440**

Capítulo 1

Metodologia de desenvolvimento

1. Introdução

A proposta deste livro é permitir o monitoramento dos sensores listados no sumário da obra remotamente através da plataforma thinger.io. Para isso, foi utilizado o ESP32 encapsulado no módulo NODEMCU-32S com a programação em Arduino.

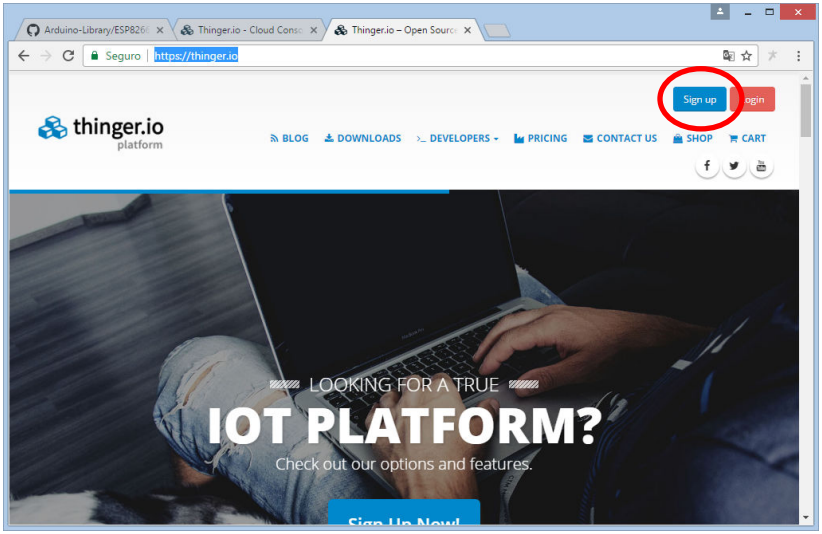
É fundamental que o leitor utilize como referência ou possua experiência dos assuntos abordados na literatura *Programando o ESP32 no Arduino (2018)* do mesmo autor e editora.

Capítulo 2

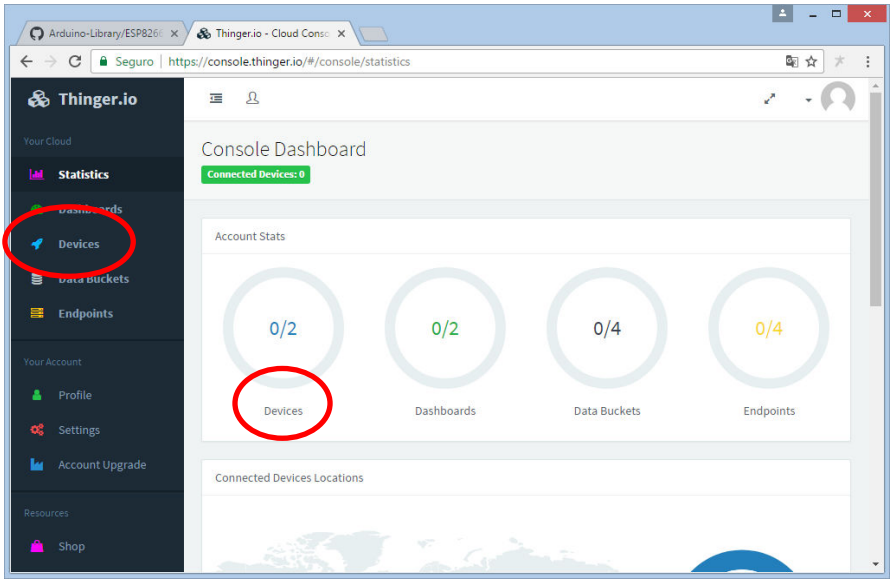
Thinger.io

1. Obtendo o id e credencial

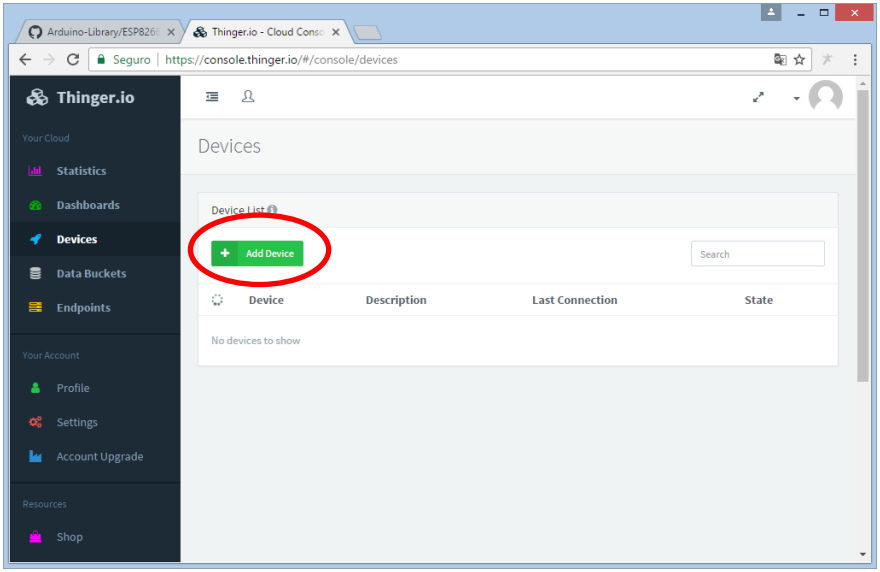
O Thinger.io é uma plataforma IoT (*Internet Of Things* ou Internet das Coisas) que permite o envio e recepção de dados através de um computador ou smartphone, já que ele opera através de um navegador de internet. Para permitir a conexão com o ESP32 é necessário criar um id e credencial e para obtê-lo, acesse o endereço https://thinger.io/.



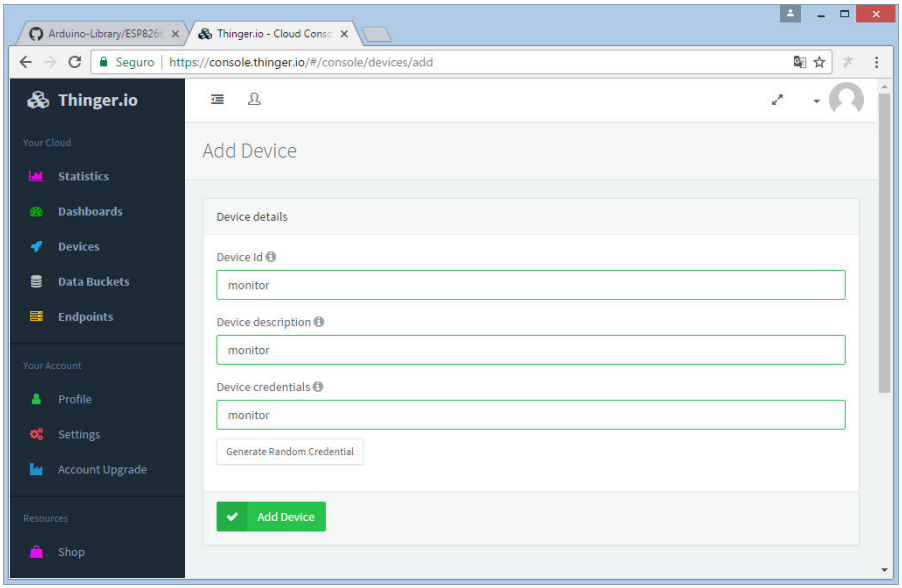
Clique no botão *Sign up* para fazer o cadastro na plataforma. Após este passo, acesse a mesma, onde a tela abaixo será carregada.



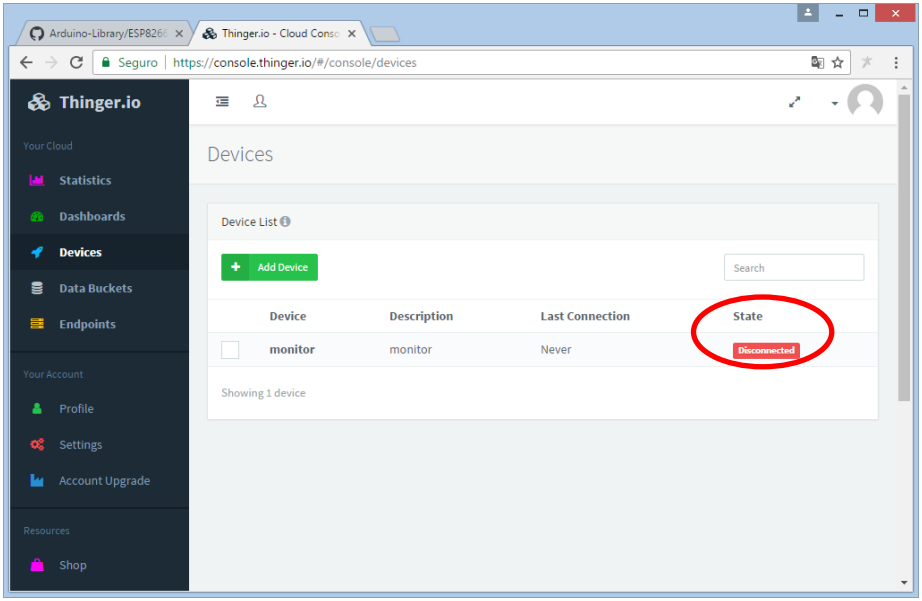
Clique no botão *Devices* para criar um id e credencial.



Neste momento a tela fica como apresentado a seguir, onde é possível definir o id, credencial e descrição do *Device* (dispositivo). Neste exemplo, todos os campos foram preenchidos como monitor.



Clique no botão *Add Device*. A partir deste momento, é possível visualizar que um dispositivo foi criado e seu estado inicial é desconectado, já que a programação para o ESP32 ainda não foi desenvolvida.



Clicando sobre o *Device* monitor é possível visualizar quantos bytes foram transmitidos ou recebidos assim como outras informações pertinentes, sendo que a visualização desta janela ficará mais clara assim que a programação do ESP32 estiver concluída.

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.

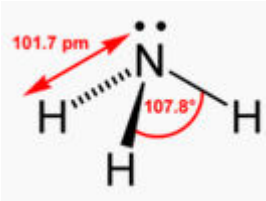


Figura 3.1: Representação da amônia
Fonte: http://pt.wikipedia.org/wiki/Am%C3%B4nia

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.

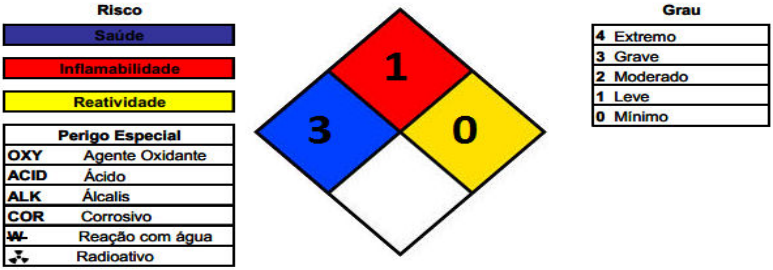


Figura 3.2: Riscos associados a Amônia
Fonte: http://pt.wikipedia.org/wiki/Am%C3%B4nia

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.

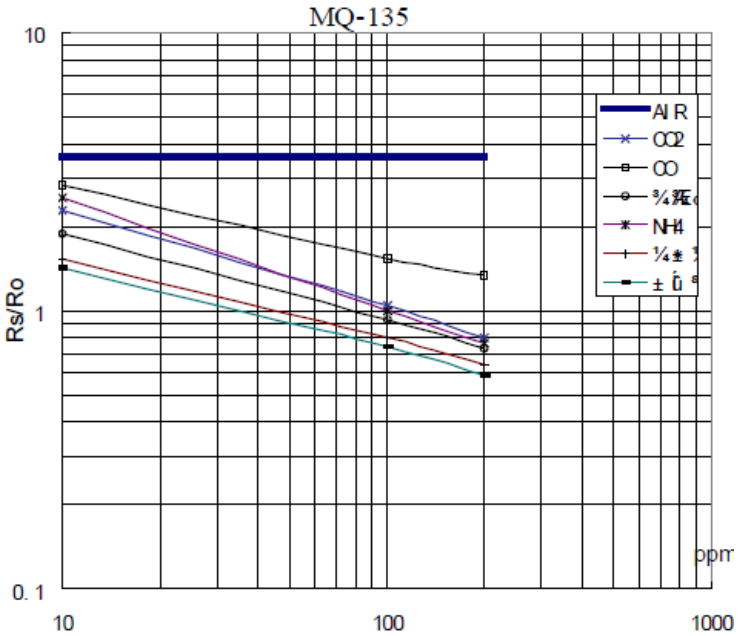


Figura 3.5: Resposta de acordo com a concentração do meio
Fonte: MQ-135 Datasheet

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.

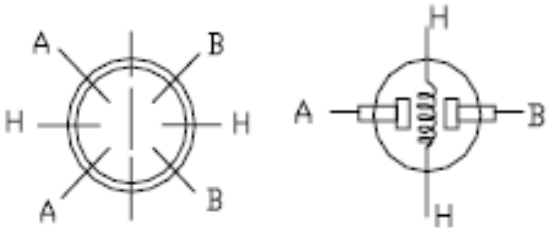


Figura 3.6: Pinagem do MQ-135
Fonte: MQ-135 Datasheet

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

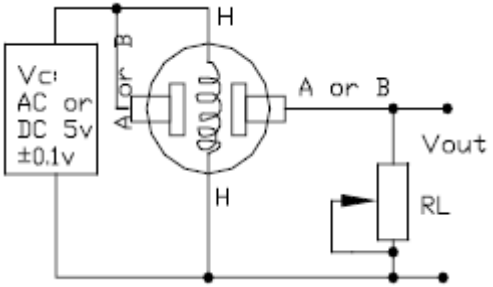


Figura 3.7: Circuito típico
Fonte: MQ-135 Datasheet

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.

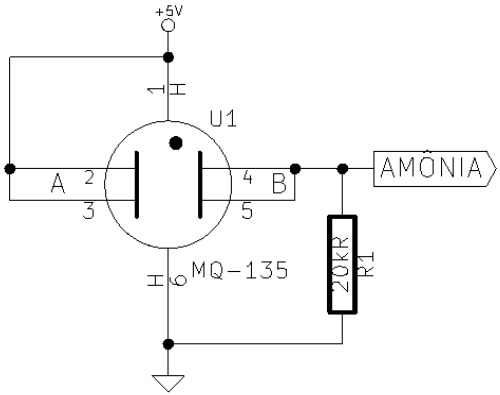


Figura 3.8: Circuito elétrico
Fonte: MQ-135 Datasheet

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.

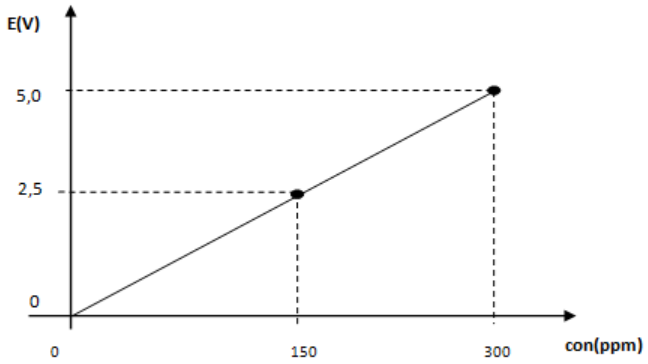


Figura 3.9: Resposta do sensor em Volts

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.

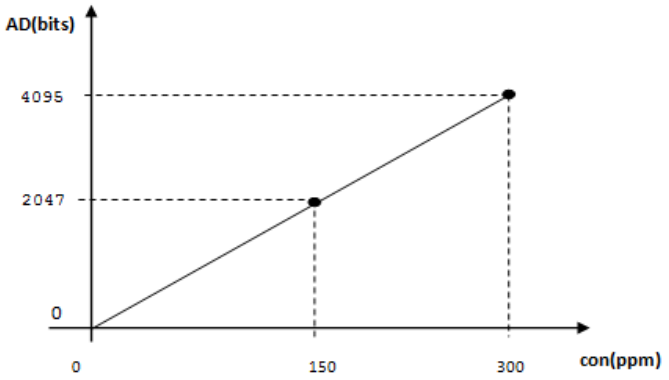
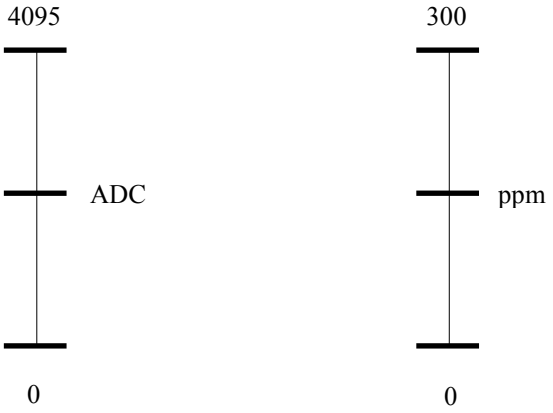


Figura 3.10: Equivalência com resposta do CAD

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.

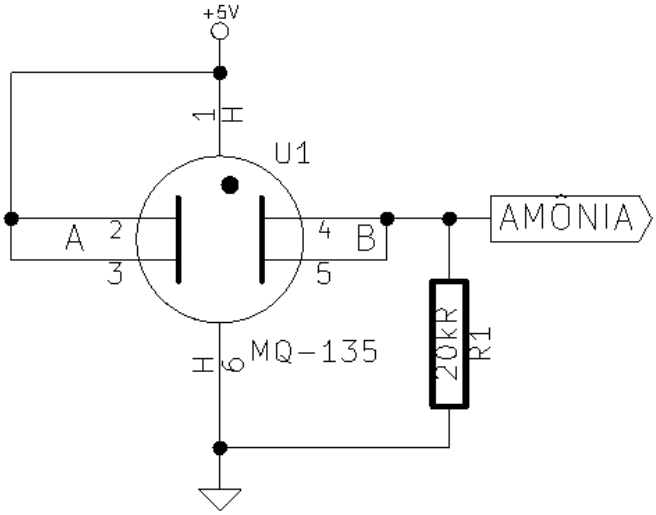


Figura 3.12: Esquema elétrico completo

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.

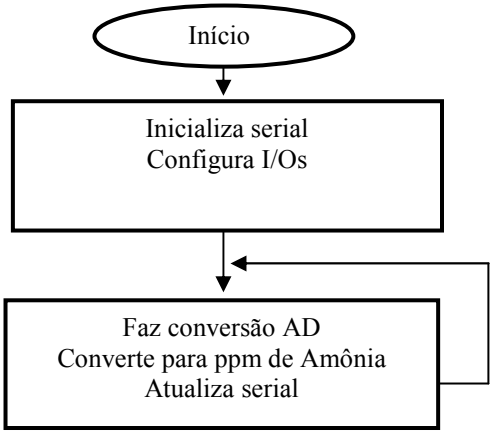


Figura 3.13: Fluxograma

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
```

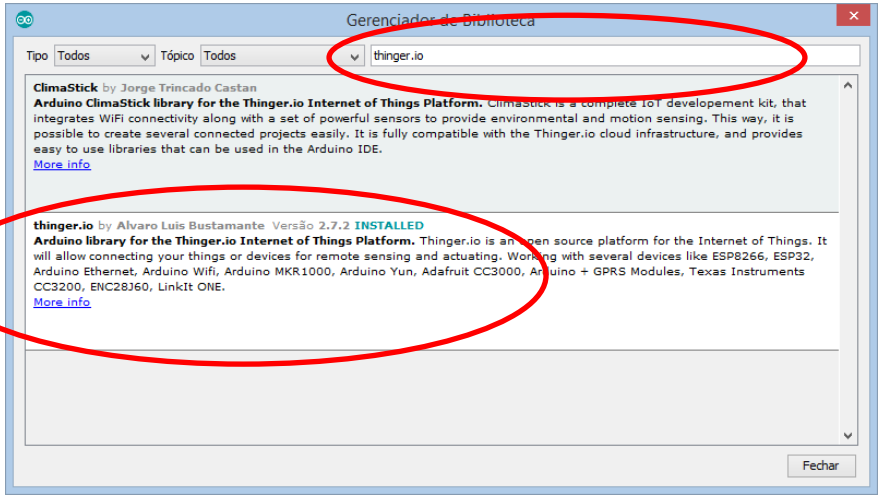
```
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *      Função de inicialização      *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void setup(void)
{
  Serial.begin(9600);
}

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *          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. Instalando a biblioteca

Por default, o Arduino não incorpora a biblioteca para comunicação com o Thinger.io. Para adicioná-la, vá ao menu Sketch-> Incluir Biblioteca -> Gerenciar Bibliotecas e digite no campo marcado abaixo thinger.io.



Instale a biblioteca thinger.io para concluir a operação.

Obs: A conexão a internet deve estar habilitada.

7. Código fonte

A seguir está apresentado o código fonte que implementa a funcionalidade esperada. Sua explicação está logo em seguida.

```
#include <WiFi.h>
#include <ThingerESP32.h>

#define USERNAME          "cernetec"
```

```
#define DEVICE_ID         "monitor"
#define DEVICE_CREDENTIAL "monitor"
#define SSID              "cerne"
#define SSID_PASSWORD     "tecnologia"

ThingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

void setup()
{
  thing.add_wifi(SSID, SSID_PASSWORD);

  thing["adc"] >> [] (pson & out)
  {
    out=le_ppm_NH3();
  };
}

void loop()
{
  thing.handle();
}

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *  Função para ler Amônia em ppm    *
 *  *  *  *  *  *  *  *  *  *  *  *  */
float le_ppm_NH3(void)
{
  float ppm,ad;
  ad = analogRead(0);
  ppm=(300.0*ad)/4095.0;
  return(ppm);
}
```

O programa inicia com a definição das bibliotecas utilizadas ao longo do projeto.

```
#include <WiFi.h>
```

```
#include <ThingerESP32.h>
```

Em seguida há a definição do nome de usuário, id e credencial criados no thinger.io. Observe que estes parâmetros devem ser atualizados segundo o nome de usuário que o leitor criou juntamente com o id e credencial definidos no capítulo II, que neste exemplo foi considerado monitor.

```
#define USERNAME          "cernetec"
#define DEVICE_ID         "monitor"
#define DEVICE_CREDENTIAL "monitor"
```

O próximo bloco define o nome e senha da rede que permite o acesso a internet através da rede local, sendo que estes parâmetros devem ser atualizados segundo a sua rede interna.

```
#define SSID              "cerne"
#define SSID_PASSWORD     "tecnologia"
```

O próximo trecho define um objeto chamado thing segundo o nome de usuário, id e credencial especificados anteriormente.

```
ThingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
```

O bloco setup faz o acesso a rede WiFi segundo o SSID e PASSWORD. Em seguida, cria um elemento para leitura da variável

analógica do capítulo passado. O nome “adc” será usado para identificá-lo em breve no thinger.io.

```
void setup()
{
  thing.add_wifi(SSID, SSID_PASSWORD);
  thing["adc"] >> [] (pson & out)
  {
    out=le_ppm_NH3();
  };
}
```

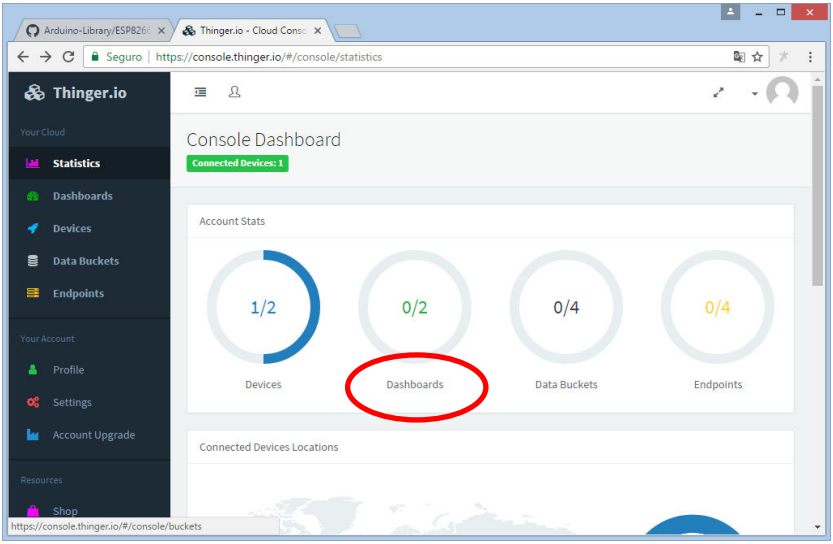
Obs: Mais elementos podem ser associados assim como outros nomes.

O bloco loop consiste em tratar as mensagens trocadas com a plataforma IoT através do método `handle()`.

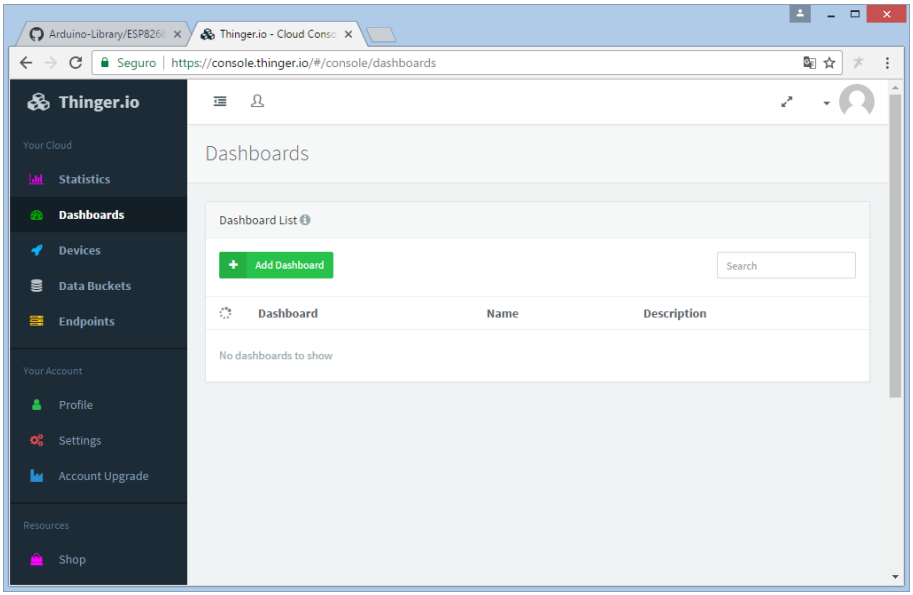
```
void loop()
{
  thing.handle();
}
```

8. Dashboard

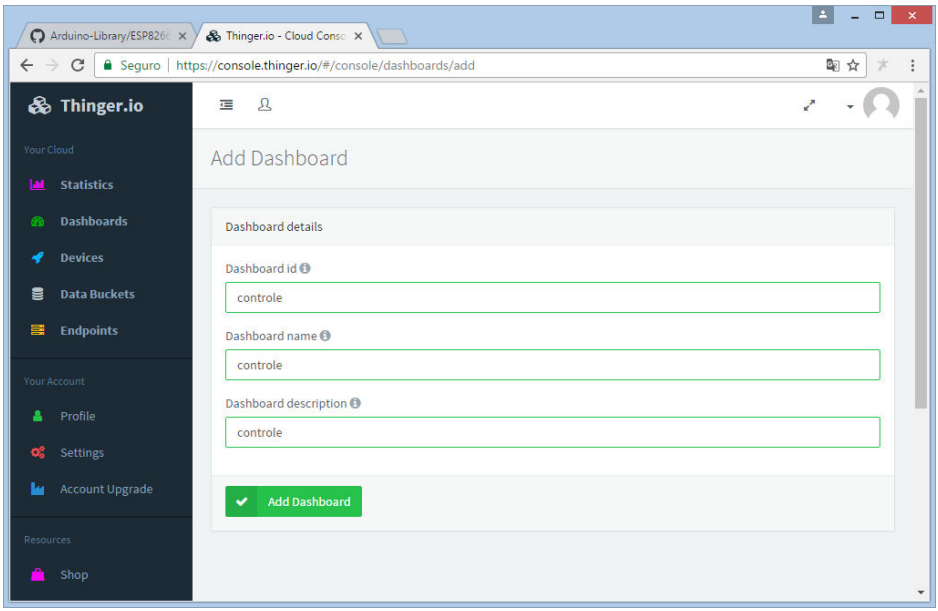
Através do *Dashboard* no thinger.io é possível acompanhar a leitura da variável analógica remotamente. Para acessá-lo, clique sobre a opção *Dashboard*.



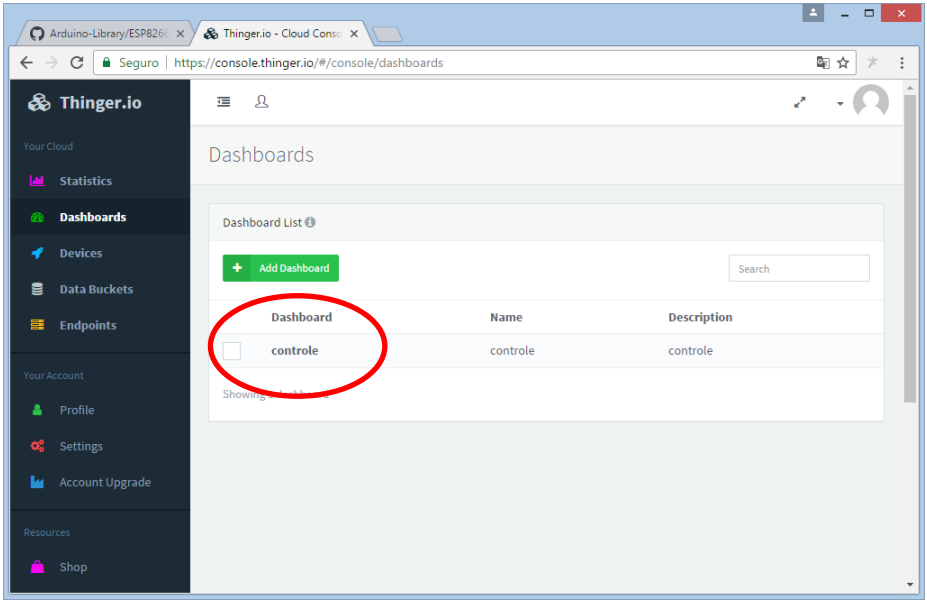
A próxima janela será apresentada.



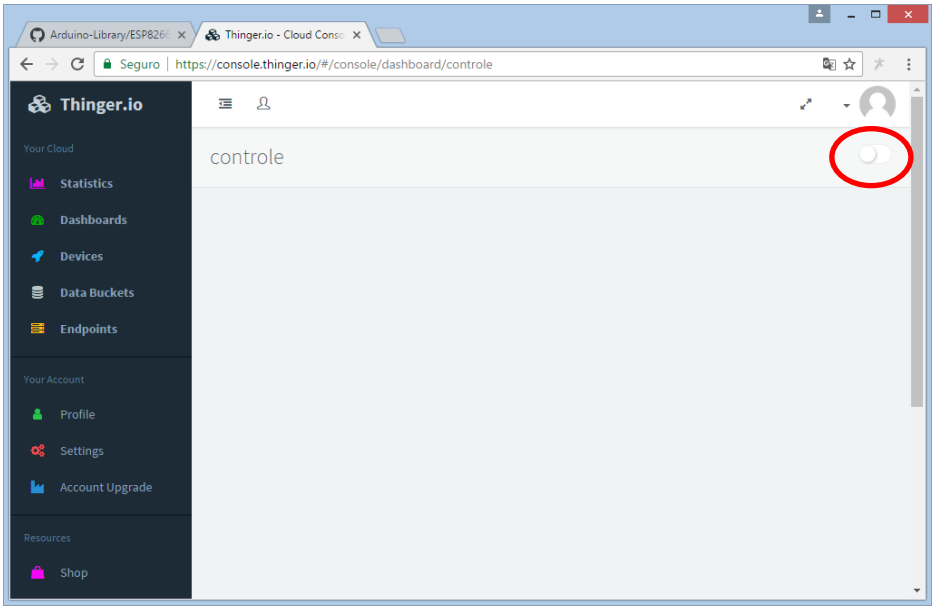
Clique sobre o botão *Add Dashboard* e no campo id, name e description digite controle e encerre clicando sobre o botão *Add Dashboard*.



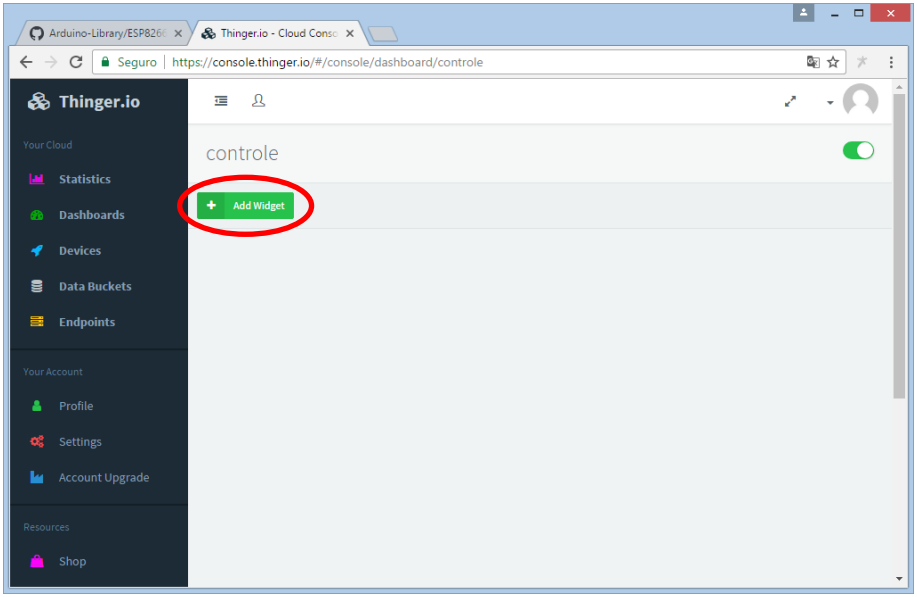
Note que há a presença de um novo *Dashboard* a partir deste momento.



Clique sobre controle e habilite a chave marcada na figura abaixo, movendo-a para direita.

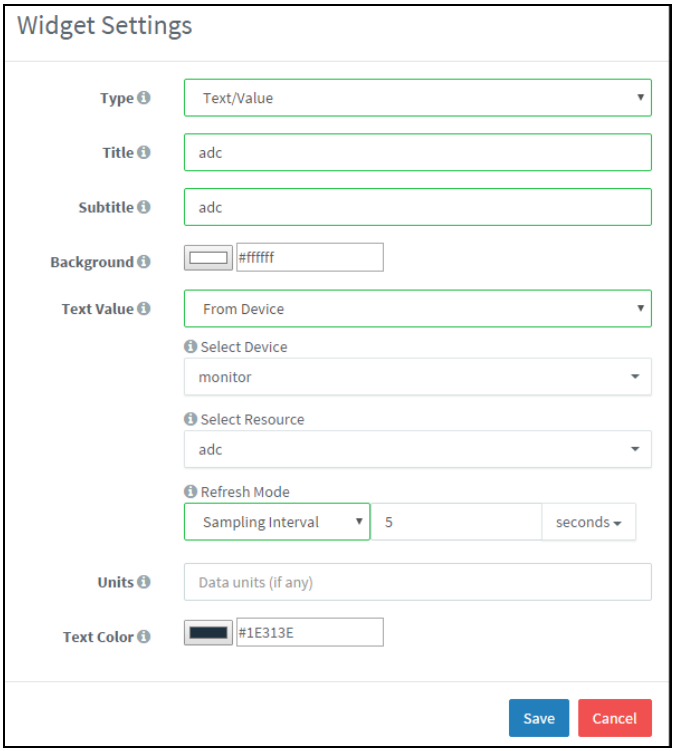


Clique sobre o botão *Add Widget*.



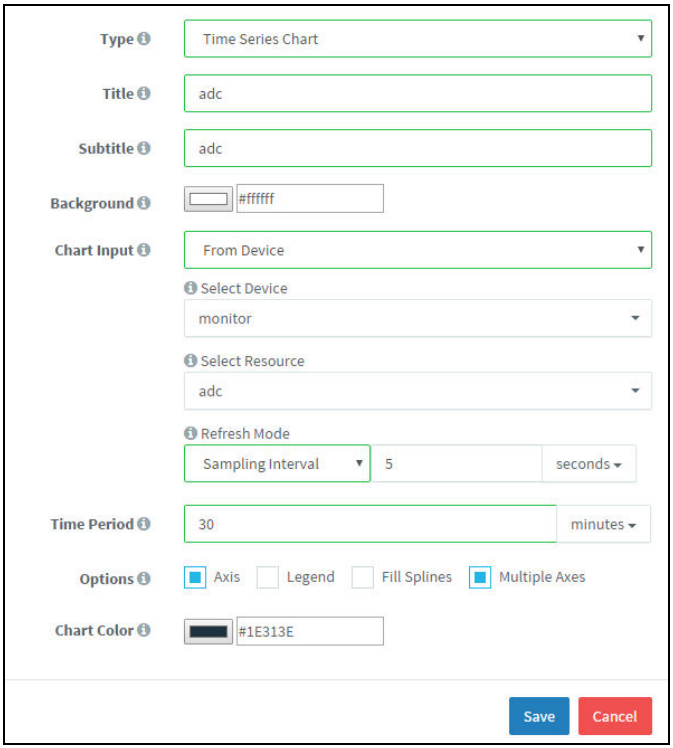
Neste momento uma nova janela é apresentada, solicitando o tipo de apresentação a ser utilizada. Escolha a opção *Display-> Text/Value* para apresentar o valor da variável analógica do capítulo passado conectado ao ESP32.

Em seguida, a janela é atualizada e novas informações são solicitadas como Title, Subtitle etc. Deixe a configuração como indicado abaixo.

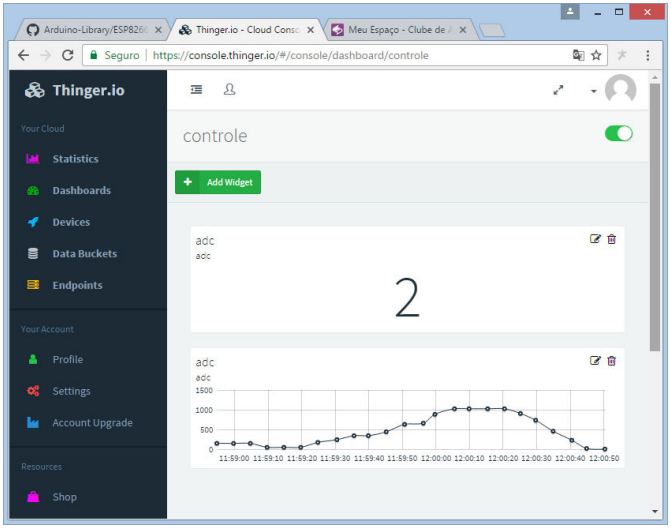


Foi definido o título e sub-título como adc e escolhido o elemento adc do dispositivo monitor para atualizar a informação. Observe que a cada 5 s o valor presente na placa será atualizado na tela. Após estas definições clique no botão Save.

Crie um novo Widget para monitorar o estado da variável de forma gráfica. Deixe a janela como apresentado abaixo.



O resultado final está apresentado abaixo, onde nota-se dois Widgets que permite mensurar remotamente a variável analógica do capítulo passado.

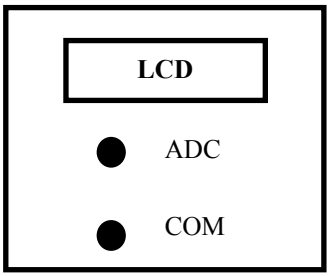

Arduino-Library/ESP826
Thinger.io - Cloud Cons
Meu Espaço - Clube de
Seguro | https://console.thinger.io/#/console/dashboard/controle
Thinger.io
Your Cloud
Statistics
Dashboards
Devices
Data Buckets
Endpoints
Your Account
Profile
Settings
Account Upgrade
Resources
Shop
controle
Add Widget
adc
adc
2
adc
adc
1500
1000
500
0
11:59:00 11:59:10 11:59:20 11:59:30 11:59:40 11:59:50 12:00:00 12:00:10 12:00:20 12:00:30 12:00:40 12:00:50

Capítulo 4

Amperímetro

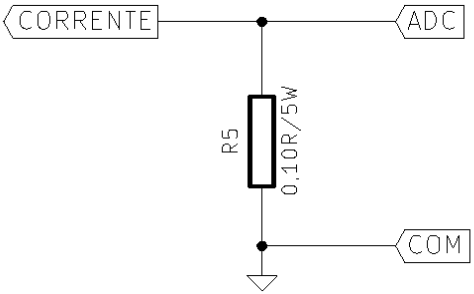
1. Introdução

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.



Como o amperímetro fica em série com o circuito a ser mensurado, a entrada COM representa a entrada enquanto que ADC representará a saída do medidor.

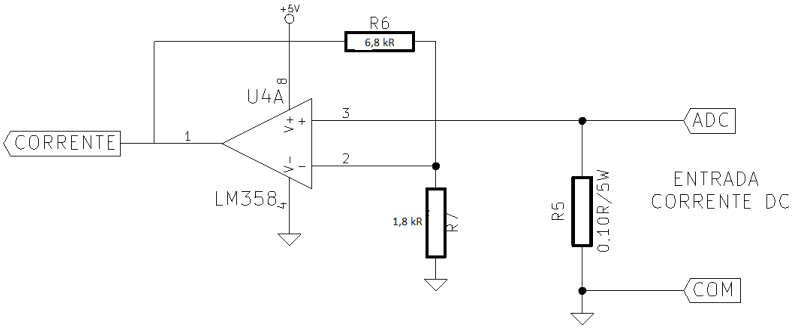
A faixa de corrente contínua projetada para medição será de 0,0 A a 7,0 A. Como o ESP32 mede na faixa de 0 V a 3,3V, é necessário um circuito de conversão corrente para tensão utilizando um resistor shunt de 0,10Ω/5W, que fornece uma tensão proporcional a corrente que passa pelo mesmo. Observe o circuito de conversão corrente-tensão a seguir que oferece o funcionamento comentado.



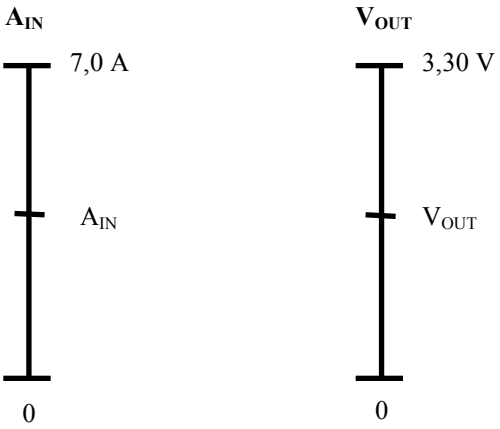
As entradas ADC e COM servem para realizar a medição no circuito enquanto que a saída CORRENTE fornece a tensão proporcional a corrente que passa pelo resistor shunt. Quando a corrente de 7 A passar pelo circuito, é observado a seguinte tensão no terminal CORRENTE, segundo a 1º Lei de Ohm.

$$V = R.I$$
$$V = 0{,}10.7$$
$$V = 0{,}70V$$

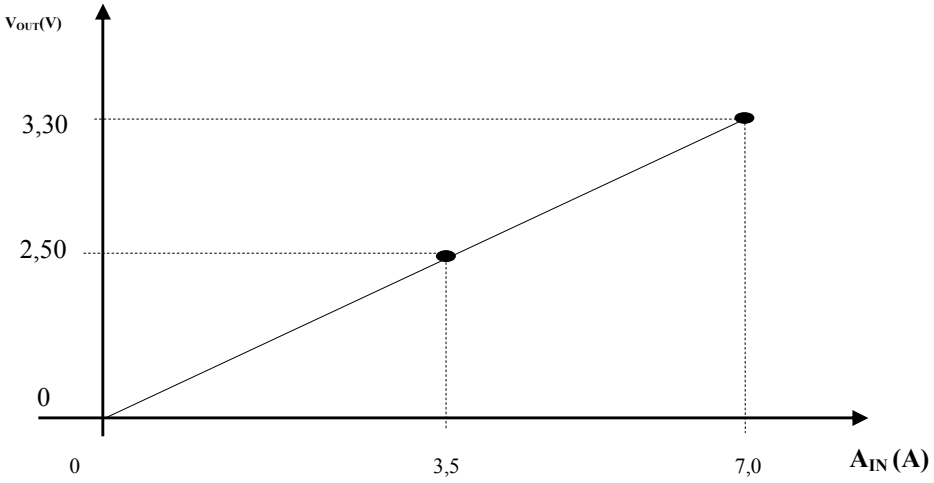
Esta tensão é muito pequena para ser tratada pelo conversor ADC do ESP32 que possui faixa de 0 a 3,3 V. Sendo assim é necessário amplificá-la através de um amplificador operacional (AOP) não inversor com ganho de 4,7. Isso será obtido através do circuito abaixo.



A tensão de 3,3 V será a máxima na entrada analógica do ESP32. Desta forma, a figura abaixo mostra a relação da corrente medida e da entrada analógica do ESP32.

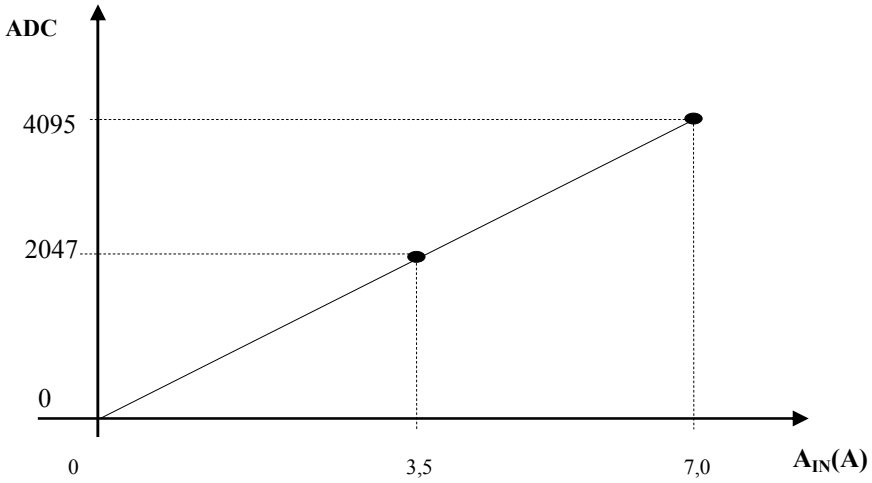


A curva que relaciona a tensão de saída (V_{OUT}) com a corrente de entrada (A_{IN}) terá a característica apresentada a seguir.

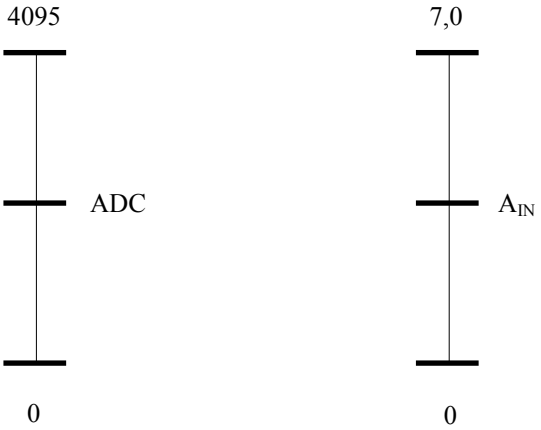


O conversor ADC do ESP32 é de 12 bits, o que indica que o mesmo é capaz de ter uma resolução de 4096 pontos, já que 2^{12} é igual a este

valor. A tensão a ser medida está ligada a uma das entradas analógicas do ESP32, 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 entrada para os bits referentes a conversão ADC.



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



$$\frac{ADC\ -0}{4095\ -0} = \frac{A_{IN} -0}{7{,}0 - 0}$$

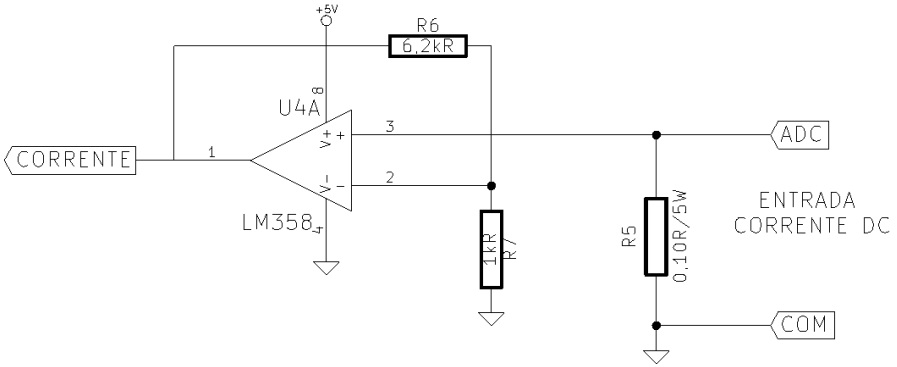
$$\frac{ADC}{4095} = \frac{A_{IN}}{7{,}0}$$

$$A_{IN} = 7{,}0\frac{ADC}{4095}$$

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

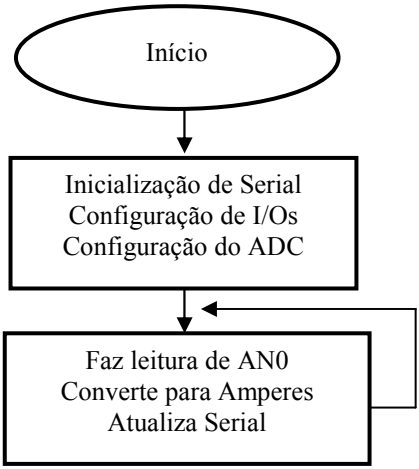
2. Esquema elétrico

O esquema elétrico para este projeto está apresentado a seguir, onde a entrada de tensão DC proporcional a corrente está conectada a entrada analógica 0 do ESP32.



3. Fluxograma

O algoritmo para o desenvolvimento deste projeto está apresentado a seguir.



4. Código fonte

O código fonte completo, seguindo o algoritmo visto anteriormente está apresentado a seguir. Observe que no loop principal, o programa fica constantemente lendo a entrada analógica AN0 para que em seguida aplique a fórmula de conversão de ADC para corrente de entrada, atualizando o LCD em seguida.

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

#define _ADC0      36
```

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

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *          Função de loop           *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void loop(void)
{
    Serial.println(String((analogRead(_ADC0)*7.0)/4095)+"A");
    delay(500);
}
```

5. Instalando a biblioteca

Por default, o Arduino não incorpora a biblioteca para comunicação com o Thinger.io. Para adicioná-la, vá ao menu Sketch-> Incluir Biblioteca -> Gerenciar Bibliotecas e digite no campo marcado abaixo thinger.io.