Vitor Amadeu Souza

Desenvolvimento de Projetos para

Monitoramento

no Visual C# 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.

Outubro 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

“Se não se espera, não se encontra o inesperado.”

Heráclito

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 e etc. Veja os detalhes de cada um nas figuras abaixo:



Kit Arduino

- Gravação On-Board
- Comunicação Serial RS232
- Entrada de 12 V

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

Sumário

Capítulo 1 – Metodologia de desenvolvimento **6**
Capítulo 2 – Display LCD **7**
Capítulo 3 – Medição de corrente de 4 a 20 mA **11**
Capítulo 4 – Medindo Álcool em ppm **24**
Capítulo 5 – Medindo a concentração de Amônia **40**
Capítulo 6 – Amperímetro **60**
Capítulo 7 – Barômetro **76**
Capítulo 8 – Medindo a concentração de Benzeno **90**
Capítulo 9 – Célula de Carga **109**
Capítulo 10 – Medição de Monóxido de Carbono **127**
Capítulo 11 – Medição de Dióxido de Carbono **149**
Capítulo 12 – Decibelímetro **170**
Capítulo 13 – Dinamômetro **194**
Capítulo 14 – Medindo o fluxo de massa **207**
Capítulo 15 – Gaussímetro **224**
Capítulo 16 – Medindo a concentração de GLP **244**
Capítulo 17 – Medindo a concentração de H_2 ..... **263**
Capítulo 18 – Higrômetro **282**
Capítulo 19 – Medindo a concentração de Ozônio **296**
Capítulo 20 – Medição de pH **315**
Capítulo 21 – Medição de tensão RMS **335**
Capítulo 22 – Medição de temperatura com NTC **350**
Capítulo 23 – Medindo a radiação ultravioleta **363**
Capítulo 24 – Monitoramento **385**

Capítulo 1
Metodologia de desenvolvimento

1. Introdução

Esta literatura é uma continuação da obra *Programando o ARM no Arduino – Com base no modelo STM32F103C8* (2019) do mesmo autor e editora, onde o objetivo é monitorar os sensores listados no sumário da obra programado no Arduino e uma interface gráfica desenvolvida no Visual C# para monitorá-la.

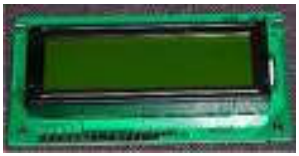
A placa didática utilizada foi a *Blue Pill* STM32F103C8, onde tal kit está à venda no site www.cerne-tec.com.br.

Capítulo 2

Display LCD

1. Introdução

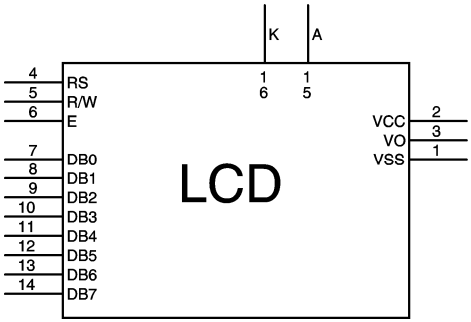
Os displays do tipo LCD são muito usados para mostrar informações do estado de uma máquina. Na figura a seguir, é apresentado aspecto típico desta interface.



Existem vários tipos de display, como os de 2x20, que significa 2 linhas por 20 colunas, 4x20, 4x16 e etc. Neste exemplo, será usado um display do tipo 2x16, onde haverá duas linhas e 16 colunas. Será apresentado um texto no LCD, de forma a entender como montar um programa para comandá-lo.

2. Montando o Hardware

Na figura a seguir, é apresentado a pinagem típica de um display LCD.



Siga a tabela apresentada a seguir para ligar o display LCD à placa STM32F103C8 Breakout Board.

Display LCD	Placa STM8S103F3 Breakout Board
VCC	5V
VSS	GND
VO	Em série com resistor de 1kΩ ao GND
RS	PA8
EN	PA9
D4	PB5
D5	PB6
D6	PB7
D7	PB8

Os pinos não citados não são usados.

3. Programando o Arduino

Foi utilizada as funções disponíveis em uma biblioteca do Arduino para que o exemplo fique funcional. A biblioteca é a LiquidCrystal.h que contém as funções de inicialização e escrita no LCD. Além disso, será necessário utilizar a função lcd.begin para configurar o tipo de display conectado ao processador. A seguir é apresentada tal função.

```
lcd.begin(colunas,linhas);
```

Em colunas, define-se o número de colunas no qual o display possui e em linhas o número de linhas. Após este passo, utiliza-se a função lcd.setCursor para posicionar a impressão de determinada informação. A seguir, tal função é apresentada.

```
lcd.setCursor(coluna,linha);
```

Onde caso coluna esteja em 0, a coluna 1 estará selecionada e caso esteja em 1, a coluna 2 e assim sucessivamente. Mesma ideia é válida para a opção linha. Finalmente, para imprimir um valor no LCD, utiliza-se a função lcd.print como apresentado a seguir.

```
lcd.print(texto);
```

A seguir os passos para mostrar uma mensagem na 1º e 2º linha do display lcd.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(PA8, PA9, PB5, PB6, PB7,
PB8);

void setup()
{
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print("Placa Arduino");
  lcd.setCursor(0,1);
  lcd.print("Exemplo LCD");
}


void loop()
{

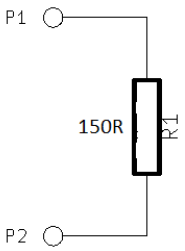
}
```

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.

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

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

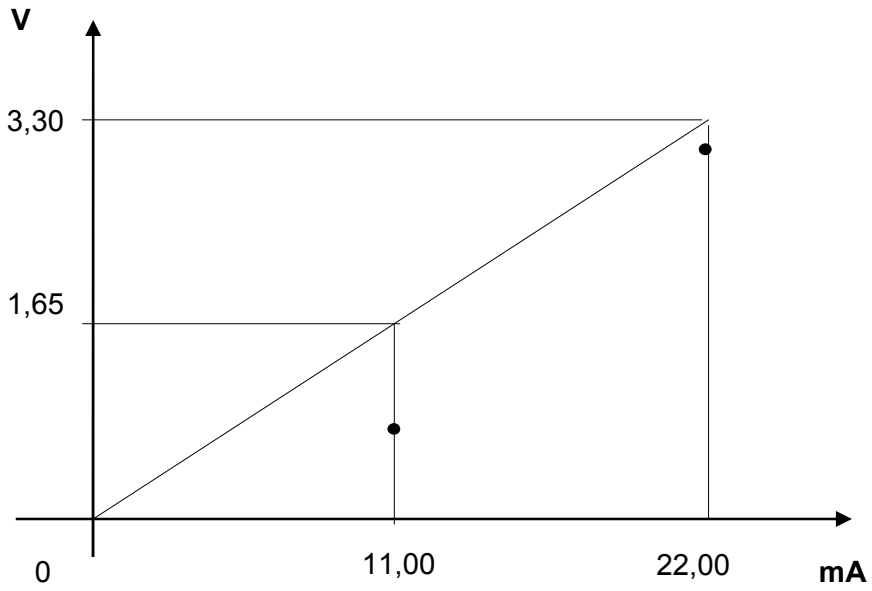
Quando a corrente mínima for medida (4 mA) a seguinte tensão é observada.

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

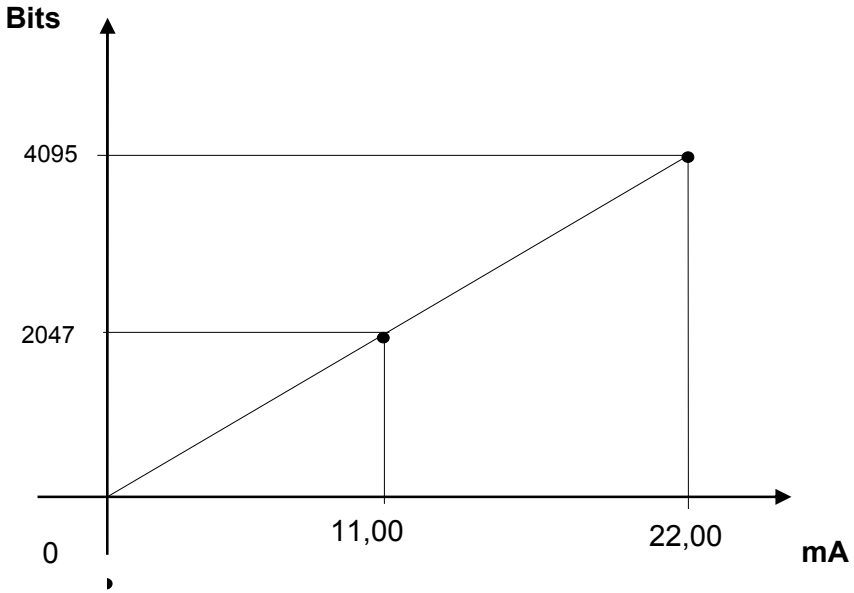
$$V = 0{,}6V$$

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.

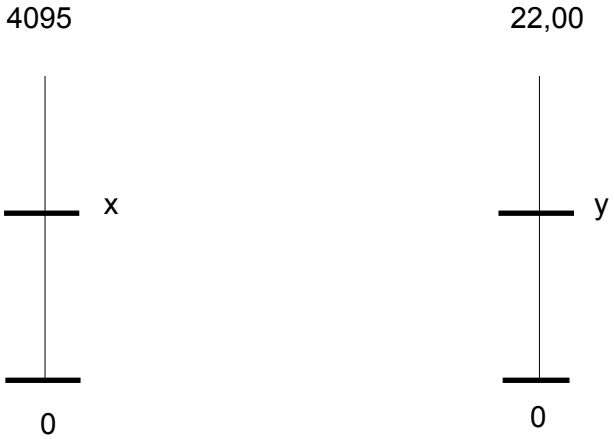
A medição é feita na faixa de 0 a 22,00 mA. A tensão de saída Vout é proporcional a corrente de entrada, onde caso haja a corrente mínima de 0 mA na saída do conversor observa-se a tensão de 0 V. Caso haja a corrente máxima de 22,00 mA a tensão no pino Vout é de 3,3 V e assim linearmente, conforme sugere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



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

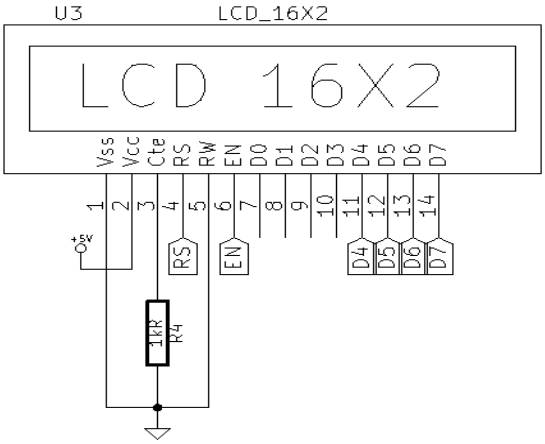
$$\frac{x}{4095}=\frac{y}{22{,}00}$$

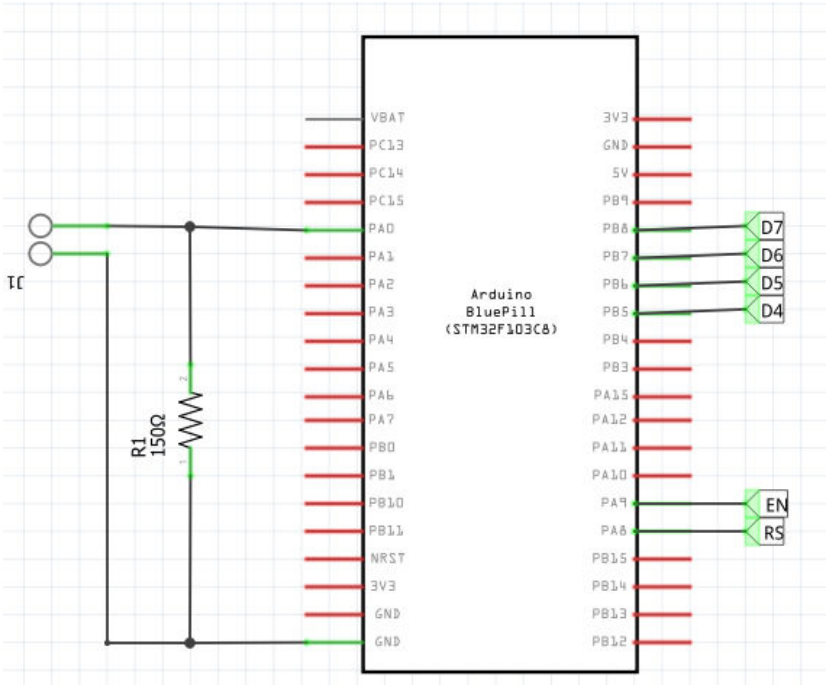
$$y=22{,}00*\frac{x}{4095}$$

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.


U3
LCD_16X2
LCD 16X2
Vss
Vcc
Cte
RS
RW
EN
D0
D1
D2
D3
D4
D5
D6
D7
+5V
RS
EN
D4
D5
D6
D7
1kR
R4


J1
R1
150Ω
Arduino
BluePill
(STM32F103C8)
VBAT
PC13
PC14
PC15
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB10
PB11
NRST
3V3
GND
GND
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. 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. Transmissão serial

A comunicação serial ainda é uma das mais usadas e neste capítulo será apresentado os passos para transmitir dados usando este tipo de interface. Será transmitida uma string, onde a placa *Blue Pill* dispõe dos recursos para que possa enviar tal informação.

5. Montando o Hardware

Este exemplo usa uma linha de transmissão e recepção que possibilitam a comunicação serial. O pino TX é identificado como PA9 e de recepção PA10. Conecte estes pinos assim como o GND a um conversor TTL<->USB, como o CH340 da figura abaixo.



6. Programando o Arduino

Deve-se configurar a velocidade no qual é feita a comunicação, podendo esta ser de várias velocidades, como 1200 bps (bits por

segundo), 2400 bps, 4800 bps, 9600 bps etc. Neste exemplo a velocidade usada é de 9600 bps. Abaixo está apresentado a função que configura a velocidade de comunicação:

```
Serial.begin(velocidade);
```

A função para enviar a string está apresentada a seguir:

```
Serial.println(texto);
```

O exemplo completo está apresentado logo abaixo enviando a string para o PC via RS232 a 9600 bps.

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Aprendendo Arduino");
}

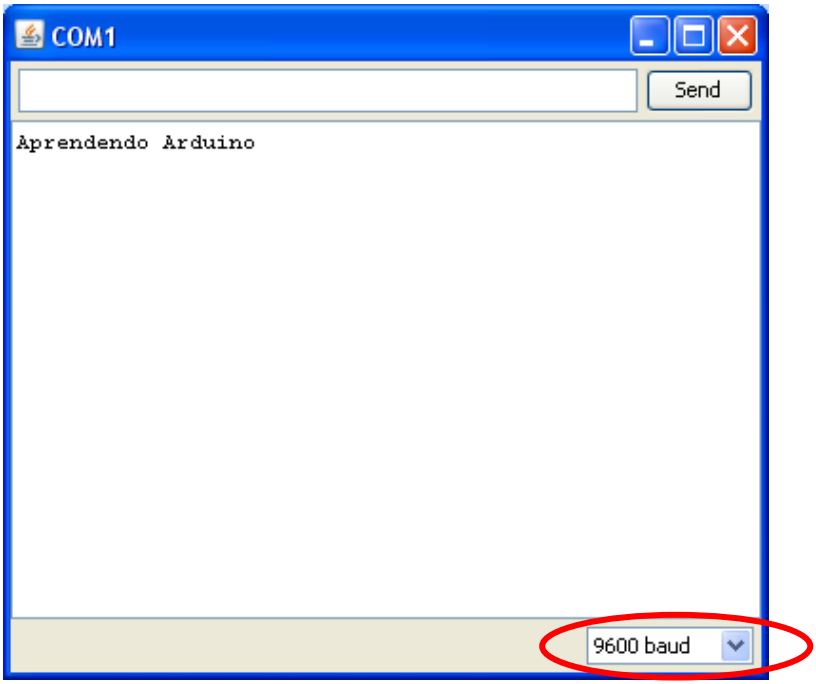
void loop()
{

}
```

Para visualizar os dados será utilizado o Serial Monitor, que já vem com o próprio software Arduino. Clique na opção marcada abaixo para inicializar este software.

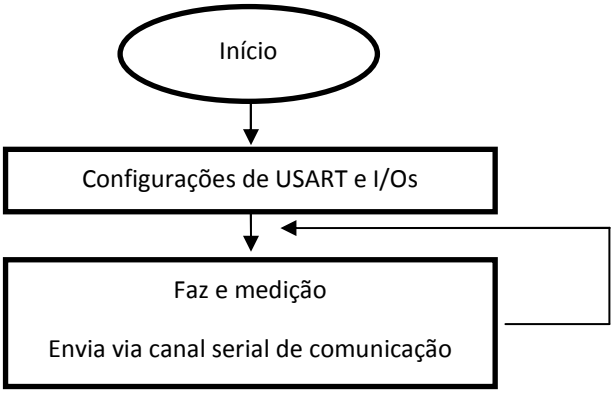


Neste momento, a seguinte tela surgirá, onde após a gravação da placa e remoção do jumper de gravação será possível visualizar o resultado apresentado abaixo a 9600 bps.



7. Medindo a variável e enviando pela serial

Neste exemplo é feita a medição da variável mensurada e tal valor é enviado periodicamente para o canal serial, de modo que através da porta serial o computador possa receber tal informação. Acompanhe a seguir o fluxograma para este experimento.



A seguir o código fonte que implementa tal algoritmo.

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

```
/*  *  *  *  *  *  *  *  *  *  *  *  *
 *       Função de inicialização       *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void setup(void)
{
  lcd.begin(16,2);
  Serial.begin(9600);
}

/*  *  *  *  *  *  *  *  *  *  *  *  *
 *           Função de loop            *
 *  *  *  *  *  *  *  *  *  *  *  *  */
void loop(void)
{
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print(le_adc (),DEC);
  lcd.print(" A");
  Serial.print(le_adc (),DEC);
  Serial.println(" A");
  delay(1000);
}

float le_adc(void)
{
  return((22.00*analogRead(0))/4095);
}
```

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.

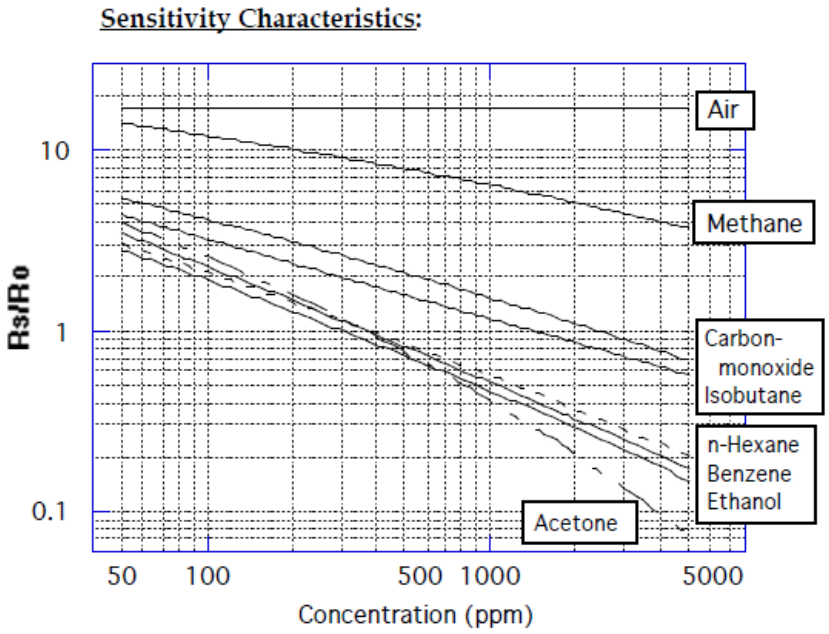


Figura 2.1: Resposta de acordo com a concentração de solvente
Fonte: TGS 822 Datasheet

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.

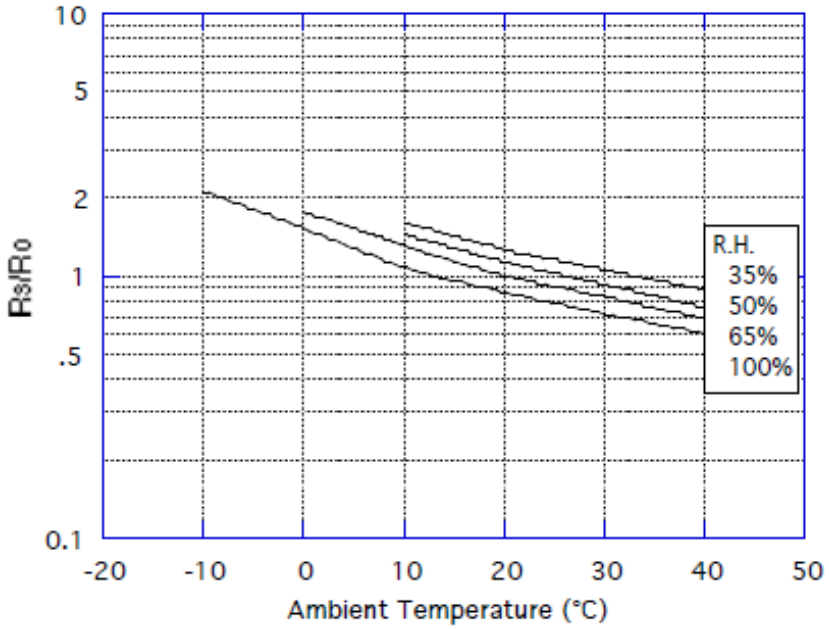


Figura 2.2: Influência da variação de umidade e temperatura
Fonte: TGS 822 Datasheet

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.

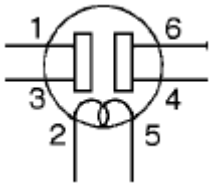


Figura 2.3: Pinagem do TGS 822
Fonte: TGS 822 Datasheet

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.

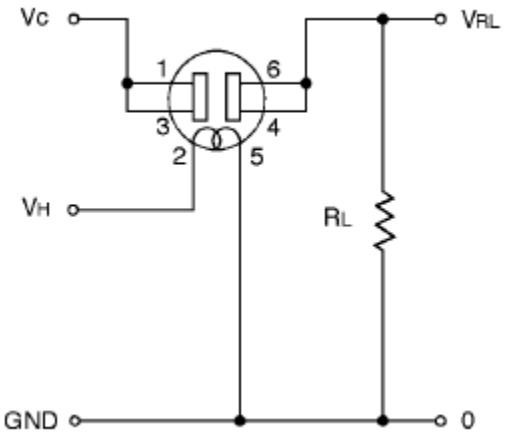


Figura 2.4: Circuito típico com TGS 822
Fonte: TGS 822 Datasheet

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.

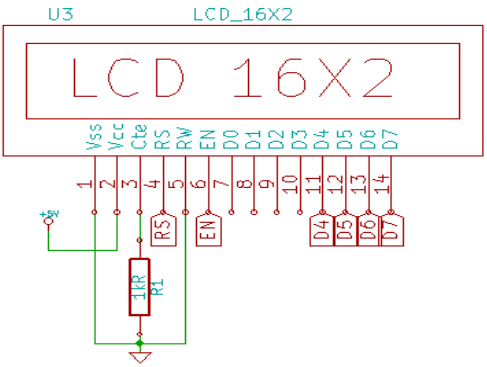
Chamando de *x* a relação entre R_S/R_0 e *ppm* a concentração, a função a seguir descreve como é encontrada a concentração.

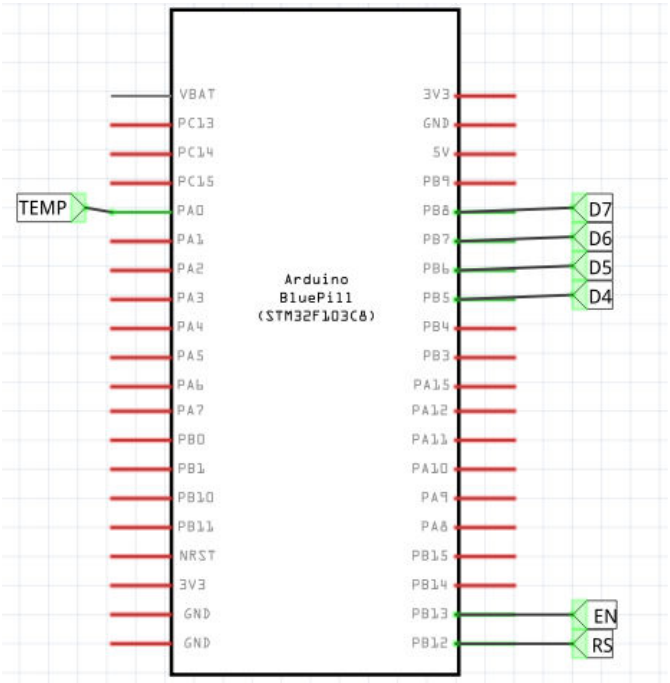
$$ppm = 169.(\log(x))^4 - 147.(\log(x))^3 + 207.(\log(x))^2 - 496.\log(x) + 332$$

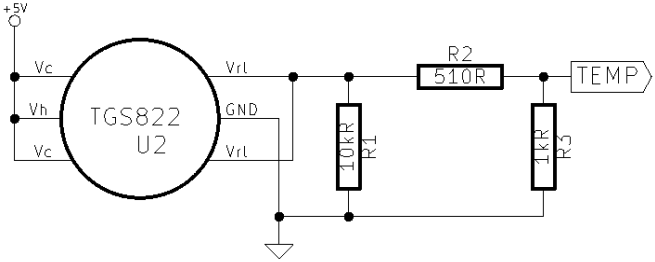
A descrição de como esta função foi encontrada está disponível no artigo de Jiana Shin, onde o link está disponível na seção Referências desta literatura.

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.

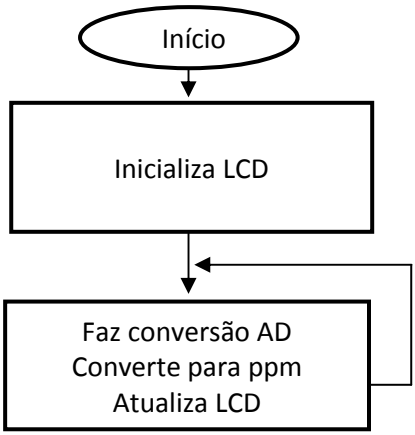

U3
LCD_16X2
LCD 16X2
Vss
Vcc
Cte
RS
RW
EN
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
+5V
RS
EN
D4
D5
D6
D7
1KR
R1


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



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. Transmissão serial

A comunicação serial ainda é uma das mais usadas e neste capítulo será apresentado os passos para transmitir dados usando este tipo de interface. Será transmitida uma string, onde a placa *Blue Pill* dispõe dos recursos para que possa enviar tal informação.

6. Montando o Hardware

Este exemplo usa uma linha de transmissão e recepção que possibilitam a comunicação serial. O pino TX é identificado como PA9 e de recepção PA10. Conecte estes pinos assim como o GND a um conversor TTL<->USB, como o CH340 da figura abaixo.



7. Programando o Arduino

Deve-se configurar a velocidade no qual é feita a comunicação, podendo esta ser de várias velocidades, como 1200 bps (bits por segundo), 2400 bps, 4800 bps, 9600 bps etc. Neste exemplo a velocidade usada é de 9600 bps. Abaixo está apresentado a função que configura a velocidade de comunicação:

```
Serial.begin(velocidade);
```

A função para enviar a string está apresentada a seguir:

```
Serial.println(texto);
```

O exemplo completo está apresentado logo abaixo enviando a string para o PC via RS232 a 9600 bps.

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Aprendendo Arduino");
}

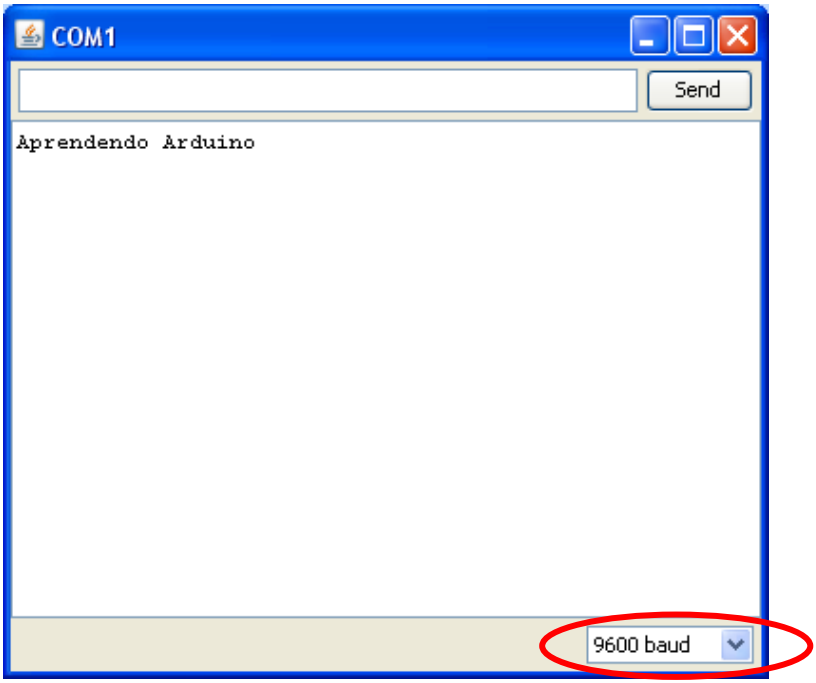
void loop()
{

}
```

Para visualizar os dados será utilizado o Serial Monitor, que já vem com o próprio software Arduino. Clique na opção marcada abaixo para inicializar este software.

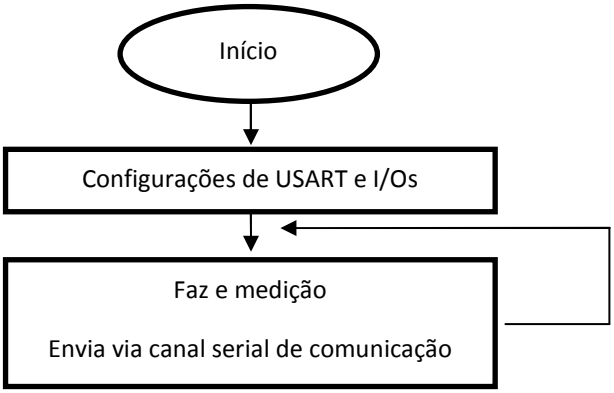


Neste momento, a seguinte tela surgirá, onde após a gravação da placa e remoção do jumper de gravação será possível visualizar o resultado apresentado abaixo a 9600 bps.


COM1
Send
Aprendendo Arduino
9600 baud

8. Medindo a variável e enviando pela serial

Neste exemplo é feita a medição da variável mensurada e tal valor é enviado periodicamente para o canal serial, de modo que através da porta serial o computador possa receber tal informação. Acompanhe a seguir o fluxograma para este experimento.



A seguir o código fonte que implementa tal algoritmo.

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