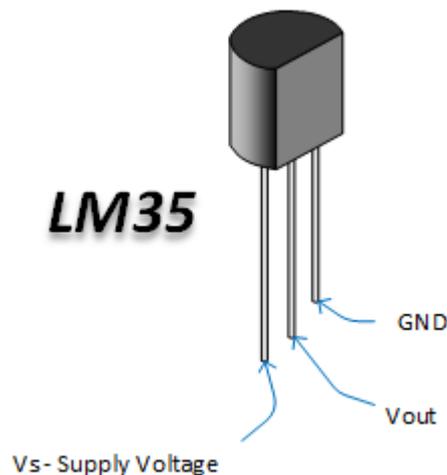


TERMÔMETRO COM O LM35 – Parte I

ARDUINO UNO R3

Neste projeto vamos desenvolver um termômetro com o sensor de temperatura LM35, que é um sensor de baixo custo e de boa qualidade.

Vamos dar uma incrementada no projeto para que o mesmo além da temperatura ambiente registre as temperaturas máxima e mínima, durante o tempo em que estiver ligado.



Vs = alimentação, que pode variar de 4 até 20 volts. Neste projeto adotaremos a alimentação de 5 volts.

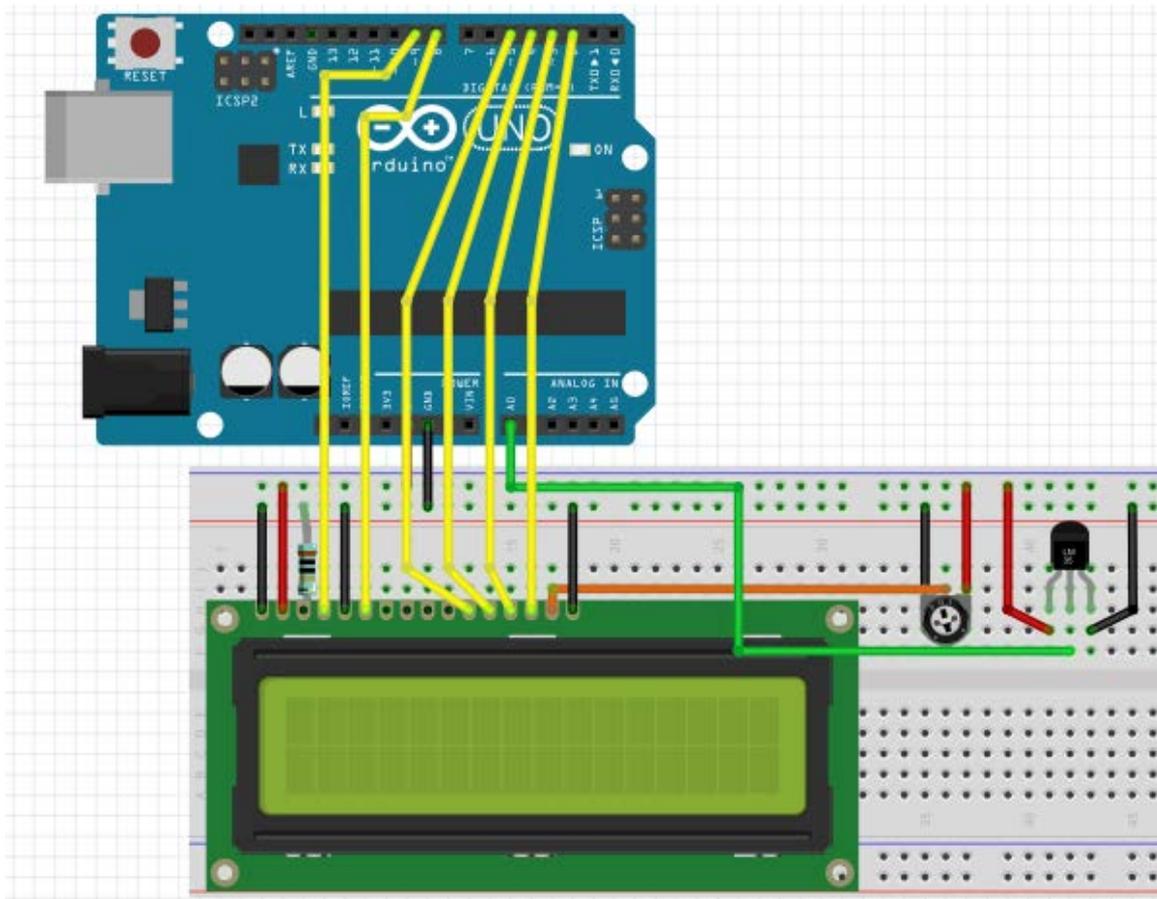
O pino Vout deve ser ligado em uma das entradas analógicas do Arduino (A0~A5).

A saída do sensor será monitorada por um display LCD 16 x 2, indicando:

- Temperatura ambiente
- Temperatura máxima
- Temperatura mínima



A figura abaixo mostra o layout do projeto, desenvolvido no software *Fritzing*.



O trimpot tem por finalidade controlar o brilho do display (backlight).

O contraste pode ser controlado por um trimpot, mas neste projeto foi inserido um resistor de 10 ohms do pino 3 do display ao terra.

Em alguns tipos de displays, o pino 3 pode ser conectado diretamente ao terra.

A entrada analógica usada para o LM35 é a A0.

COMO FUNCIONA:

Ao ligar o circuito os 3 valores que correspondem às temperaturas Ambiente, Mínima e Máxima deverão coincidir, até que se estabilizem.

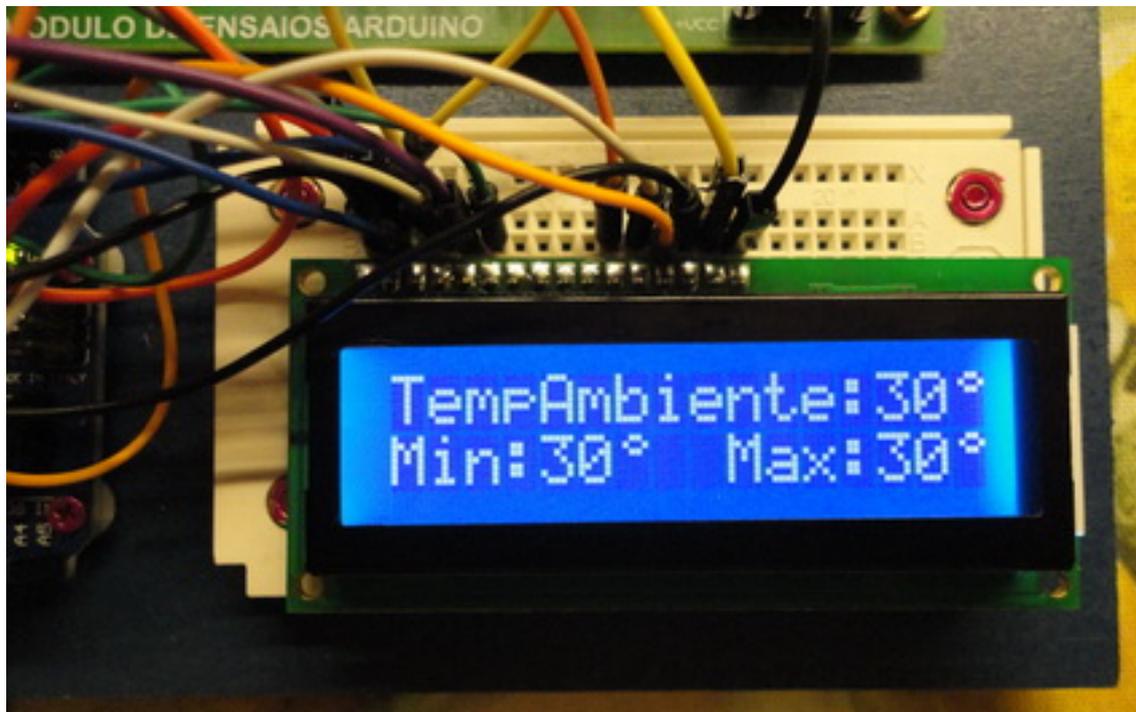
A variação da temperatura ambiente registrará os valores mínimos e máximos nos campos correspondentes.

→ Ao aproximar a ponta do soldador no LM35 as temperaturas ambiente e máxima subirão simultaneamente, ficando assim registrado o valor máximo.

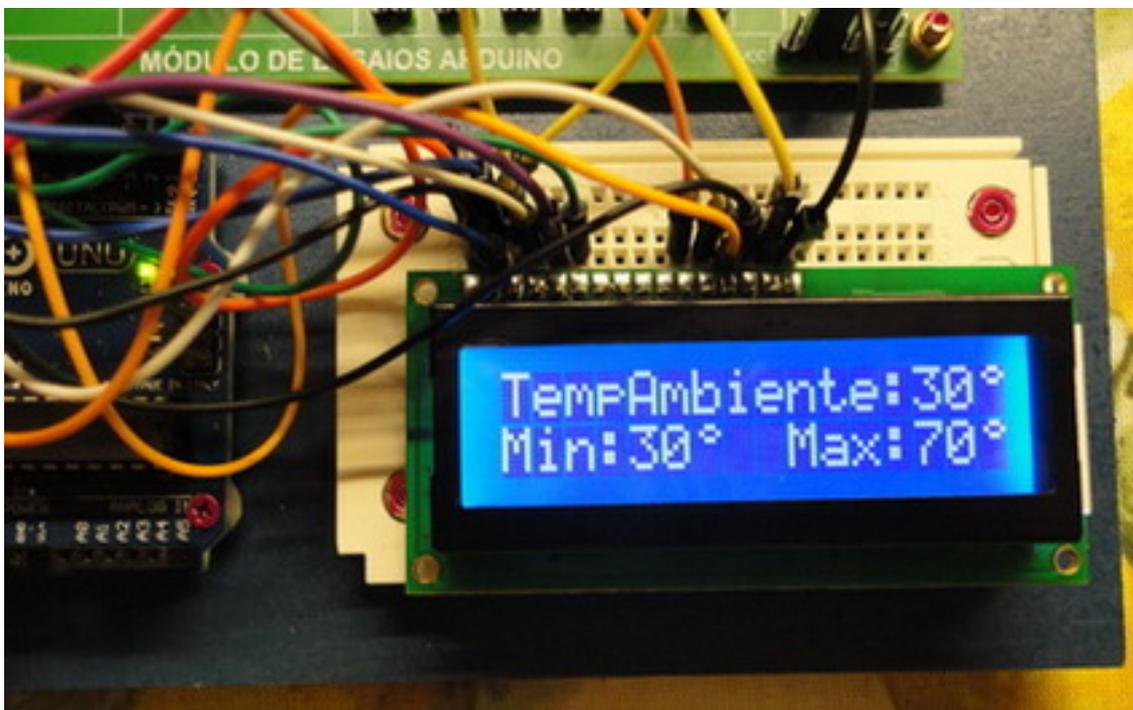
→ Ao aproximar, por exemplo, um soprador (tipo usado em estação de retrabalho) as temperaturas ambiente a mínima abaixarão simultaneamente, registrando assim o valor mínimo.

Veja a sequência abaixo:

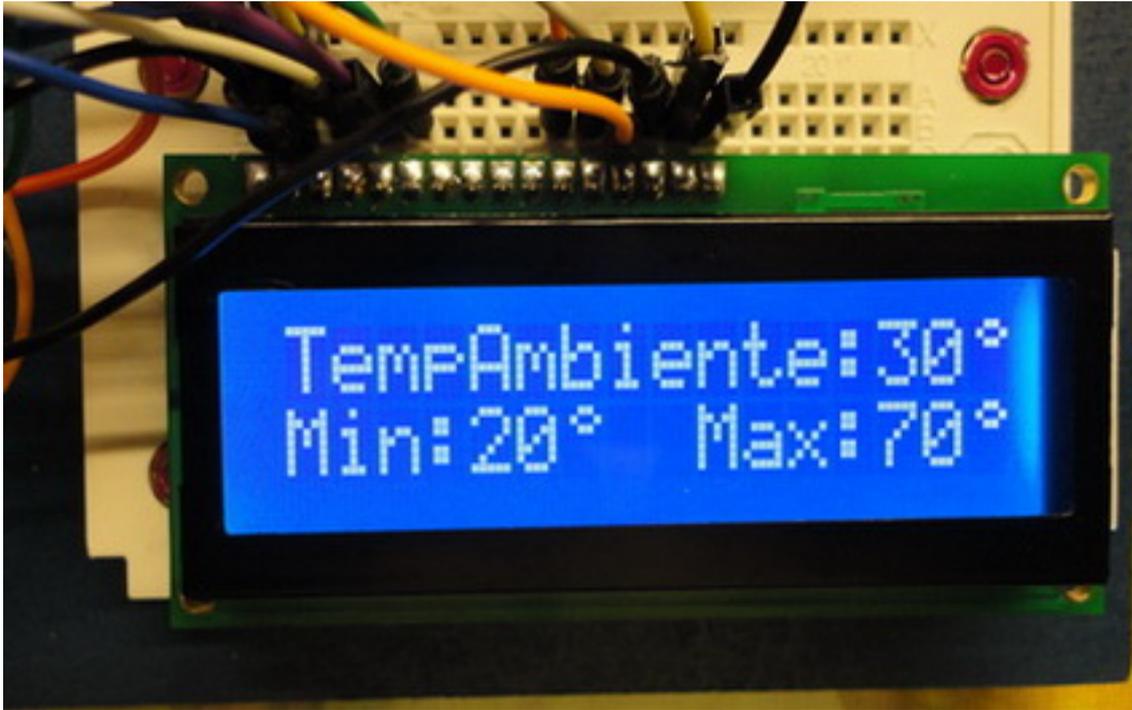
Ao ligar o circuito as temperaturas são coincidentes



Registro da temperatura máxima

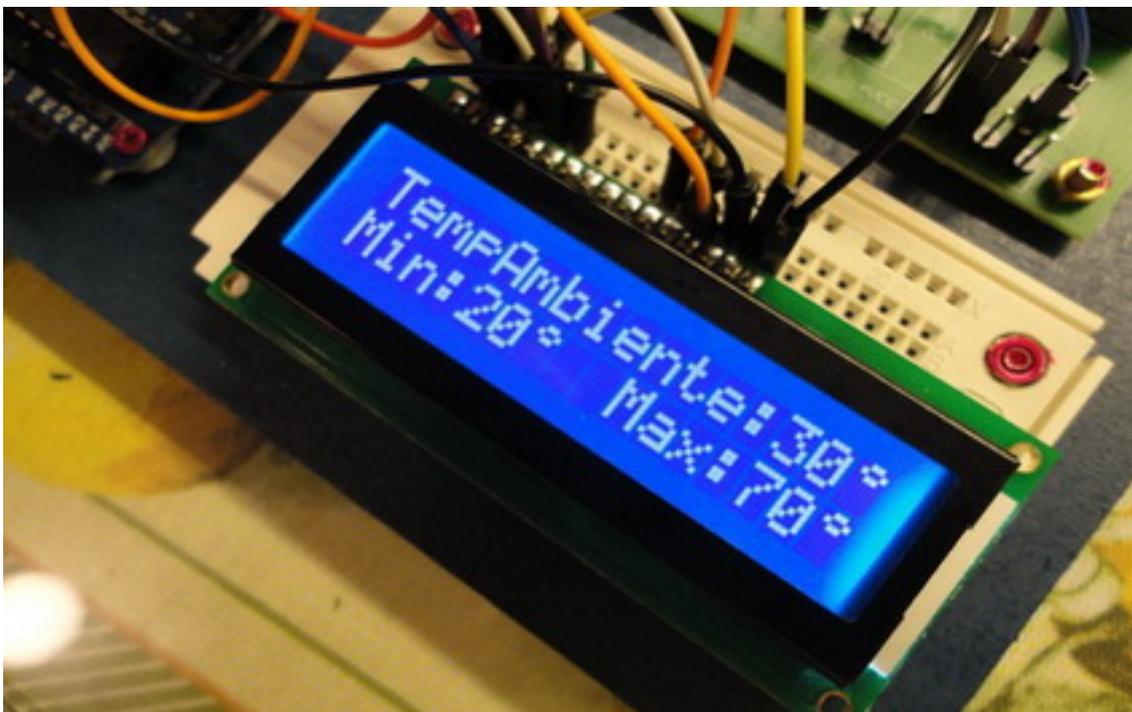


Registro da temperatura mínima



Funcionamento normal

Qualquer variação, tanto da temperatura mínima como da temperatura máxima, ficará registrada no display, até que o circuito seja desligado.



A figura a seguir mostra o LM35 na placa do MÓDULO DE ENSAIOS ARDUINO.

PROGRAMAÇÃO:

```
#include <LiquidCrystal.h> //(Biblioteca do display)

int lm35=0; // ligação do sensor (out) à entrada analógica A0 do Arduino
int tempc=0, tempf=0; // variáveis para armazenar o valor da temperatura em
Celsius ou Fahrenheit
int samples[8]; // para melhorar a precisão da leitura essa variável (array)
coleta 8 amostras ou samples (0 a 7)
int maxtemp= -100, mintemp= 100; // armazena a temperatura máxima e a
temperatura mínima
int i;

LiquidCrystal lcd(9,8,5,4,3,2); // pinos de ligação do display ao Arduino

byte a[8]={B00110,B01001,B00110,B00000,B00000,B00000,B00000,}; //esse
array ou arranjo desenha o símbolo do grau

void setup(){
  Serial.begin(9600); // inicia a comunicação serial
  lcd.begin(16,2); // identifica o tipo de display – 16 colunas, 2 linhas
  lcd.print("TempAmbiente: ");
  lcd.createChar(1,a); // ao array "a" é atribuído o valor 1. O array "a" é o que
desenha o símbolo do grau
  lcd.setCursor(6,1); //posicionamento do cursor, coluna 6 linha 1
  lcd.write(1); // escreve o símbolo do grau
  lcd.setCursor(15,0);
  lcd.write(1);
  lcd.setCursor(15,1);
  lcd.write(1);
}

void loop(){
  for(i=0;i<=7;i++) //loop que faz a leitura 8 vezes (0 a 7); se for efetuada a
leitura 7 vezes (0 a 6), por exemplo, a linha ficará assim: for(i=0;i<=6;i++)

  {
    samples[i] = (5.0 * analogRead(lm35) * 100.0) / 1024.0;
    tempc = tempc + samples[i]; // incrementa o valor da variável tempc à cada
leitura
    delay(100);
  }
  tempc=tempc/8.0; // divide a variável tempc por 8, para melhorar a precisão
na leitura
  tempf=(tempc * 9)/5 + 32; // converte a temperatura para Fahrenheit
armazenando em tempf
essa linha poderá ser também escrita assim: tempf=(tempc*1.8) + 32;
  if (tempc>maxtemp){ maxtemp=tempc; } // armazena a temperatura máxima em
tempc
  if (tempc<mintemp){ mintemp=tempc; } // armazena a temperatura mínima em
tempc
```

**/* as linhas abaixo escrevem na saída serial o valor da temperatura.
Lembrar que DEC é o valor da temperatura*/**

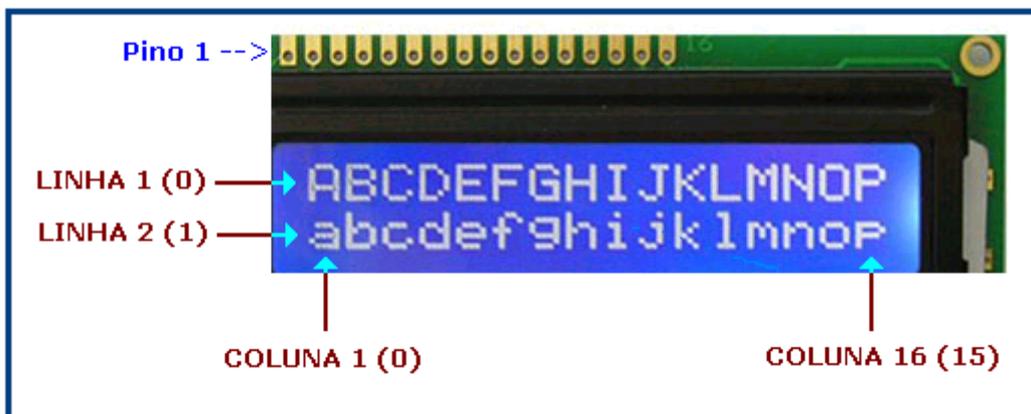
```
Serial.print(tempc,DEC);  
Serial.print(" Celsius, ");  
Serial.print(tempf,DEC);  
Serial.print(" Farenheit, ");  
Serial.print(maxtemp,DEC);  
Serial.print(" Max. ");  
Serial.print(mintemp,DEC);  
Serial.println(" Min");  
delay(100);
```

/* as linhas abaixo escrevem no display os valores das temperaturas*/

```
lcd.setCursor(13,0);  
lcd.print(tempc,DEC);  
lcd.setCursor(0,1);  
lcd.print("Min:");  
lcd.setCursor(4,1);  
lcd.print(mintemp,DEC);  
lcd.setCursor(9,1);  
lcd.print("Max:");  
lcd.setCursor(13,1);  
lcd.print(maxtemp,DEC);
```

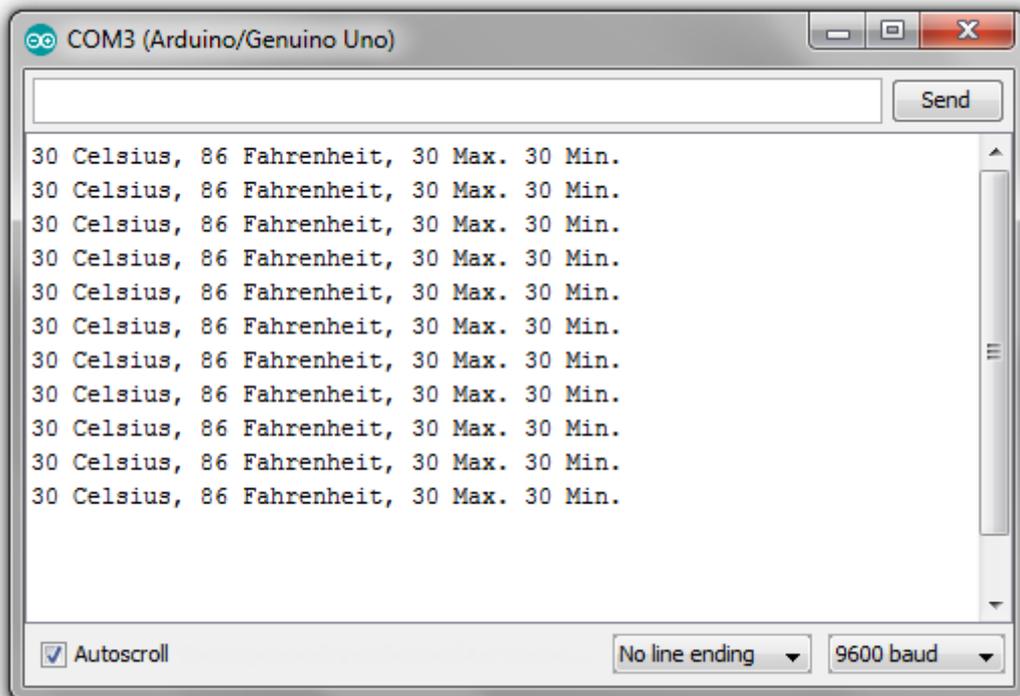
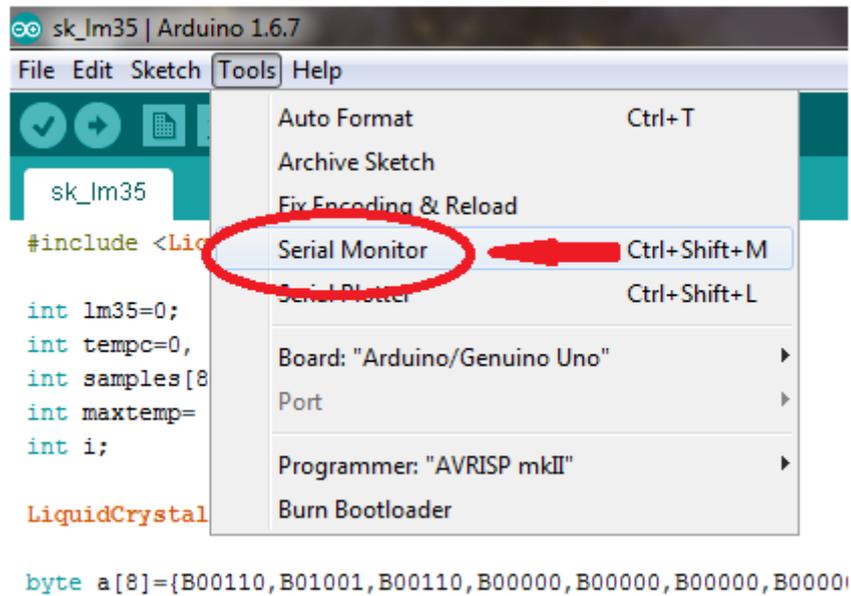
tempc=0; **//valor armazenado (graus Celsius) → reinicia o loop**

A figura a seguir orienta a numeração das linhas e colunas do display.



Muita atenção para a identificação das linhas 1 e 2, que devem ser identificadas como 0 e 1 e também com relação às colunas 1 a 16, que devem ser identificadas como 0 a 15. Exceção deve ser feita quanto a identificação dos pinos, que deve obedecer a ordem de 1 a 16.

É possível monitorar também as atividades do circuito através do Serial Monitor:



A figura acima mostra o circuito no momento em que é ligado, onde as temperaturas Ambiente, Máxima e Mínima são iguais.

Observe que a temperatura em Celsius e Fahrenheit são equivalentes, conforme já visto em uma das linhas de código, onde:

$$\text{Temp em Fahrenheit} = (\text{Temp em Celsius} \times 9) / 5 + 32$$

$$\text{Temp em Fahrenheit} = (30 \times 9) / 5 + 32 = \mathbf{86}$$