

TUTORIAL

PORTA SERIAL E DISPLAY LCD2004A (20 colunas X 04 linhas)

Este tutorial deve ser lido com calma! **Tente não pular as etapas e siga os passos sugeridos um a um**, para que você compreenda melhor o conteúdo desta aula e não precise reler o texto novamente.

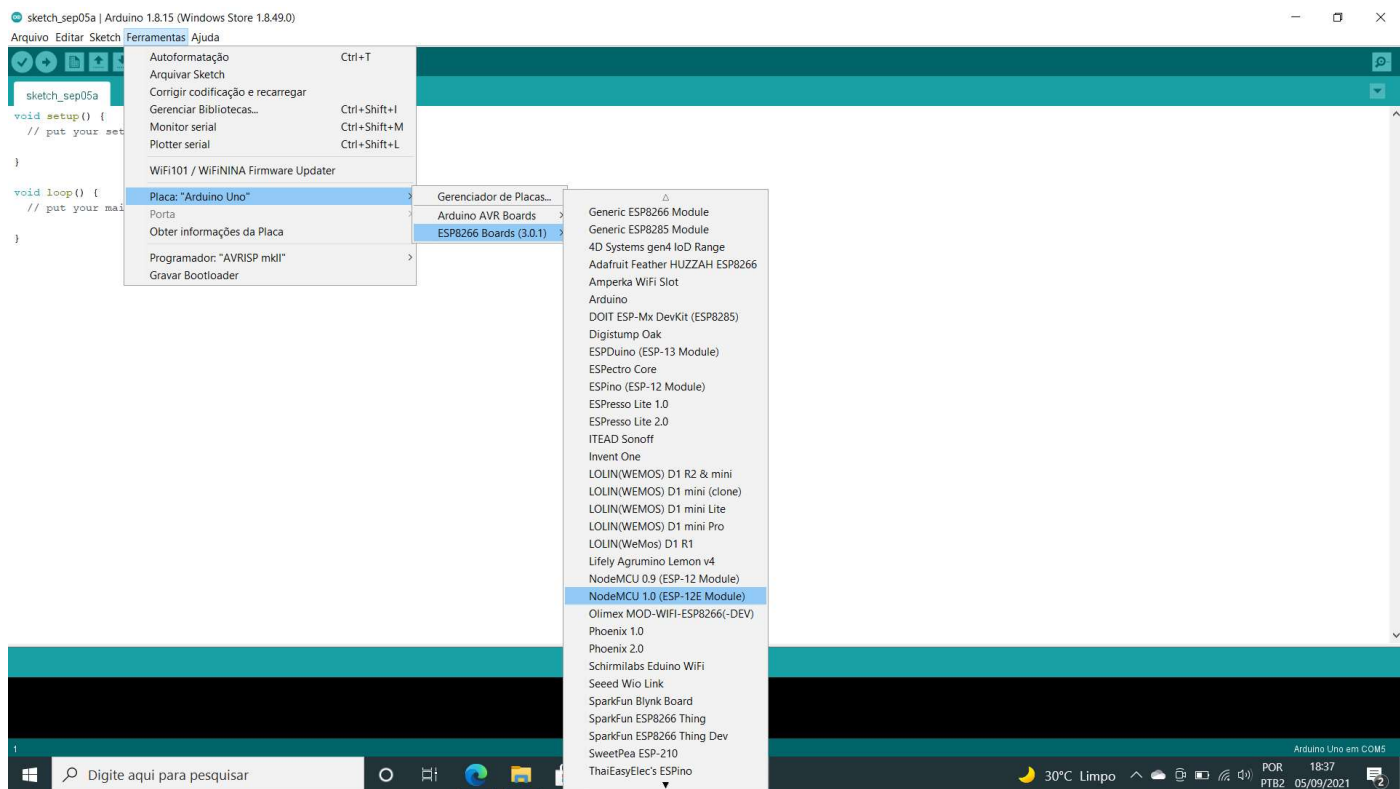
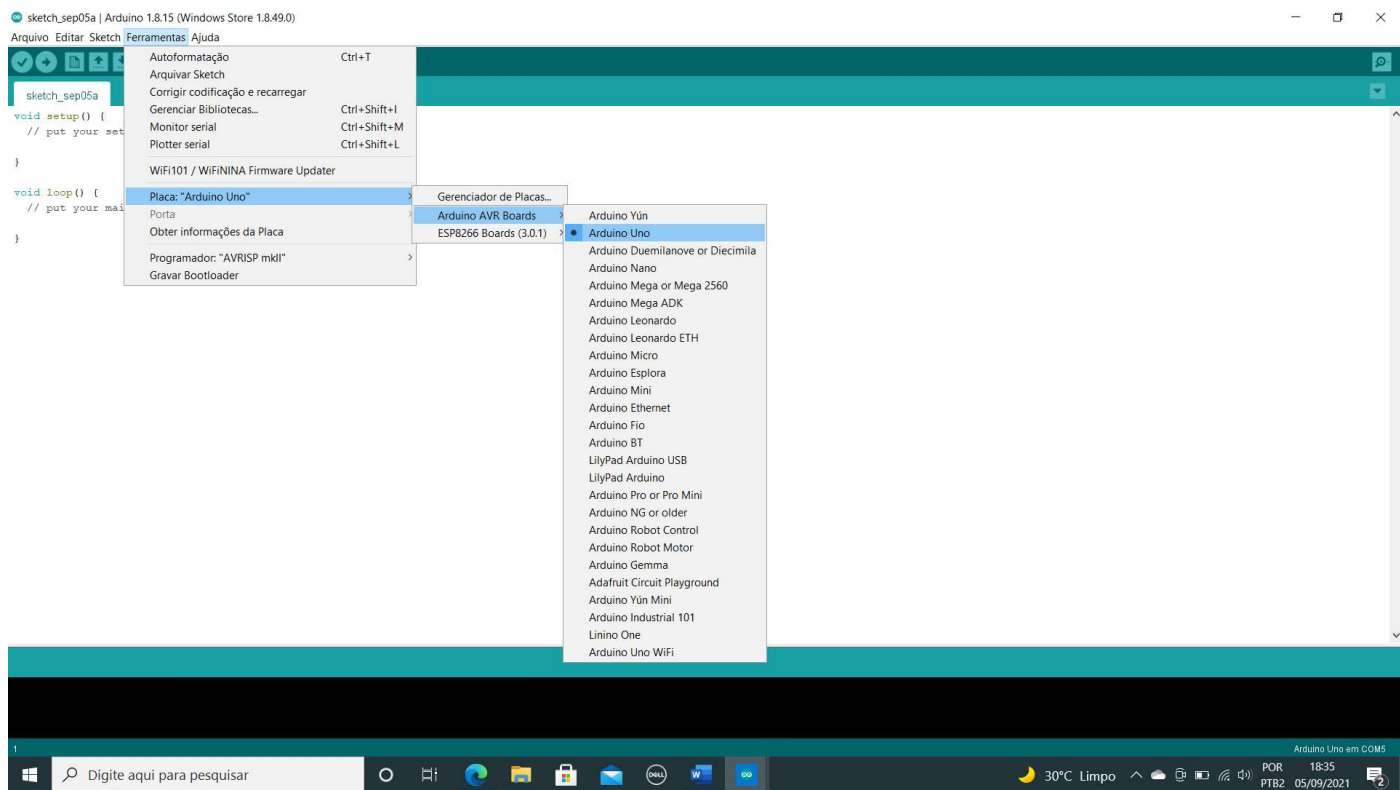
Nesta aula você aprenderá a utilizar duas formas de visualizar o que está acontecendo dentro do processador (“cérebro”) do ARDUINO ou do ESP8266. Essencialmente você aprenderá a transformar alguma informação que está virtualmente na memória do processador em informação visual (figura e texto, principalmente) que pode ser compreendida ou lida. Afinal, durante seu aprendizado, você poderá desenvolver algum tipo de projeto eletrônico em que você precise saber e visualizar quais valores de temperatura, humidade, velocidade, posição ou outra grandeza qualquer seu ARDUINO ou ESP8266 estão medindo. Outras vezes você precisará informar ou mostrar a alguém um dado ou imagem importantes, ou ainda precisará identificar em que parte da sua programação o processador se encontra, para identificar se sua programação está correta ou não (chamamos este último processo de “debug”).

Você estudará então: (1) a **Porta Serial** do ARDUINO ou do ESP8266 que se conectam ao computador pelo cabo USB ou pelos pinos/portas RX e TX; (2) o dispositivo conhecido como **LCD – Liquid Crystal Display**, ou um DISPLAY de cristal líquido.

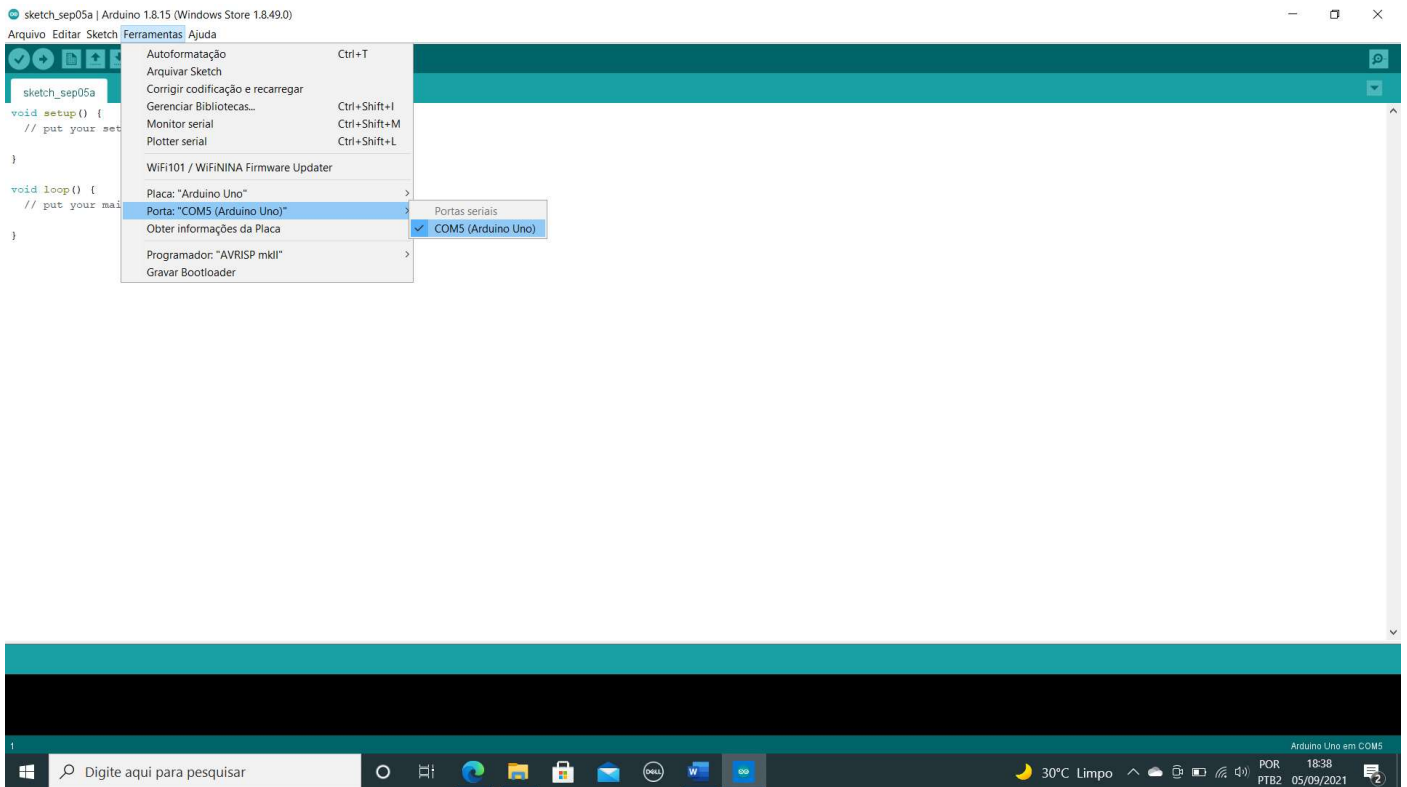
1. PORTA SERIAL

A **Interface Serial** ou **Porta Serial**, também conhecida como **RS-232** é uma porta de comunicação utilizada para conectar pendrives, modems, mouses, algumas impressoras, scanners e outros equipamentos de hardware. Na **Interface Serial**, os bits são transferidos em fila, ou seja, um bit de dados de cada vez. A **Porta Serial** do ARDUINO ou do ESP8266 já vem configurada de fábrica e pode ser utilizada diretamente a partir destes dispositivos. Basta conectar o ARDUINO ou o ESP8266 ao seu computador pelo cabo USB, abrir o aplicativo ARDUINO IDE no Windows, selecionar o tipo de dispositivo que está conectado ao

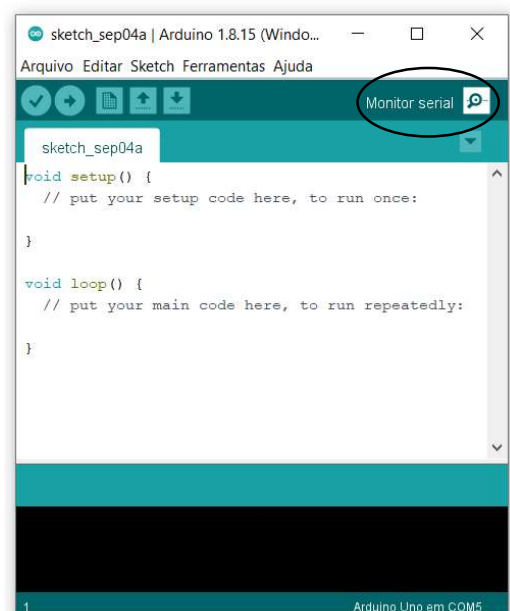
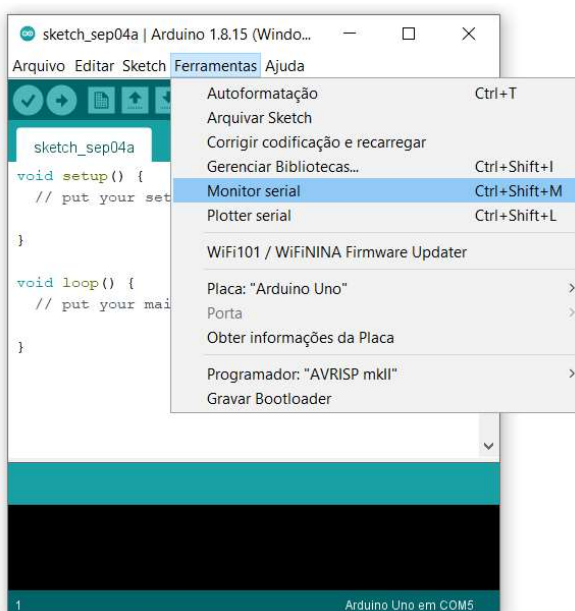
computador (ARDUINO UNO ou ESP8266), selecionar a **Porta Serial** de comunicação (**Porta COM**), e iniciar o Monitoramento Serial. **Realize estas etapas com as instruções das próximas figuras.** Nas duas figuras abaixo, demonstramos como selecionar o seu dispositivo.



Na figura abaixo, demonstramos como selecionar a porta COM. Neste caso, ao conectar o ARDUINO UNO ao computador, a Porta Serial utilizada foi a COM5, mas isto vai variar de computador para computador. Esta é a porta utilizada para a comunicação entre o processador do ARDUINO (ou do ESP8266) e o monitor do seu computador (ou mais especificamente você!)

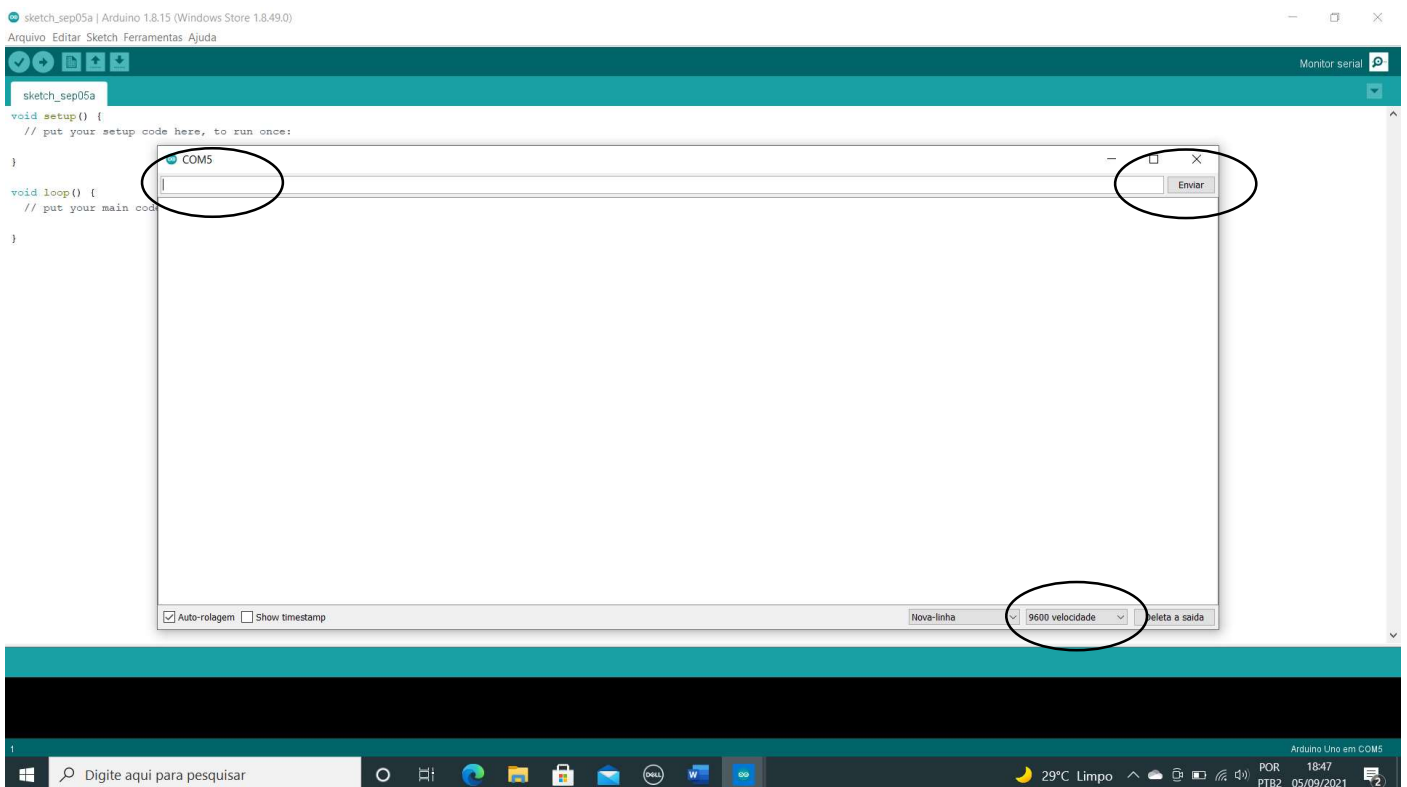


Veja, nas duas figuras a seguir, como iniciar o Monitoramento Serial.



Na primeira opção você deve clicar em Ferramentas > Monitor serial. Na segunda opção, clique no ícone em forma de lupa que se localiza à direita do ARDUINO IDE (destaque na figura).

Depois de iniciado o Monitoramento Serial, uma janela extra se abrirá na tela do computador. Será por meio desta janela que você receberá as informações do processador do ARDUINO ou do ESP8266. Será também por meio desta janela que você poderá enviar alguma informação para os processadores dos dispositivos.



Veja com cuidado algumas das opções que a tela de Monitoramento Serial possui! Estas opções estão circuladas na figura acima. **Tente identificar o que estas opções podem fazer e anote suas ideias abaixo.**

Para que você possa verificar o funcionamento da **Porta Serial**, você vai utilizar três programas simples. No primeiro, você extrairá uma informação virtual do processador do dispositivo (ARDUINO ou ESP8266) e ela será enviada pela **Porta Serial** para o monitor do computador (mais especificamente para a tela do Monitor Serial). Nos outros dois exemplos, você utilizará o Monitor Serial para enviar uma informação para o processador e este processador pegará a informação recebida e a reenviará de volta (modificada ou não) para a tela do Monitor Serial. **Digite os comandos do programa a seguir par um arquivo do ARDUINO IDE ou os copie de <https://labs.icb.ufmg.br/bioest/teaching/arduino/> e salve o arquivo (opção Salvar Como) para seu registro.**

```
//Start Program 01

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("Tempo decorrido de processamento (em milissegundos): ");
  Serial.println( millis() );
  delay(2000);
}

//End Program 01
```

Tente interpretar o código do Programa 1 e anote no campo abaixo o que você acredita ser cada um dos códigos escritos. Ao final, escreva o que você espera que aconteça quando você iniciar a tela do Monitor Serial! Se tiver dúvida, que tal fazer um “GOOGLE” e ver o que podem significar os comandos?

Agora, por fim, faça o UPLOAD do programa no ARDUINO/ESP8266 clicando no ícone da seta e logo em seguida abra o Monitor Serial. Veja se suas expectativas estavam corretas!

ATENÇÃO: Como você utilizou na sua programação o comando `Serial.begin(9600)` que definiu a comunicação entre o ARDUINO/ESP8266 e o computador para uma velocidade de 9600 bits/segundo (bits por segundo), o mesmo valor deve ser selecionado no Monitor Serial. Caso contrário ele lerá a informação recebida com outra velocidade diferente de 9600 bits/segundo, fazendo uma interpretação errada da informação recebida. Veja onde alterar este parâmetro e faça o teste com valores diferentes.

Agora copie o próximo programa para o ARDUINO IDE e salve (opção Salvar Como) o arquivo para seu registro.

```
//Start Program 02

char inputChar;

void setup() {
  Serial.begin(9600);
}

void loop() {
  if ( Serial.available() ) {
    inputChar = Serial.read();
    Serial.print("O caractere digitado foi: ");
    Serial.println(inputChar);
  }
}

//End Program 02
```

Tente interpretar o código do Programa 2 e anote no campo abaixo suas ideias. Ao final, escreva o que você espera que aconteça quando você iniciar o Monitor Serial e enviar o caractere “B” para o processador do ARDUINO/ESP8266! E se você enviar o seu nome, um número ou alguma frase? Dúvidas? Google?

Veja abaixo uma variação do programa anterior, que chamaremos de Programa 3.

```
//Start Program 03

int inputInteger;

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    inputInteger = Serial.parseInt();
    inputInteger = inputInteger + 10;
    Serial.print("O número digitado somado com 10 vale: ");
    Serial.println(inputInteger);
  }
}

//End Program 03
```

Tente interpretar o código do Programa 3 e tome nota das suas ideias. Ao final, escreva o que você espera que aconteça quando você iniciar o Monitor Serial e enviar o número "112" para o processador do ARDUINO/ESP8266!

2. DISPOSITIVO LIQUID CRYSTAL DISPLAY (LCD)

Para esta segunda parte do tutorial, você encontrará uma outra forma de visualizar as informações que estão no processador do ARDUINO ou do ESP8266. Desta vez você não verá as informações na tela do computador, mas em um display , que é um tipo de monitor mais simples. A vantagem deste dispositivo é que você pode realizar um projeto eletrônico e não necessitar do computador para ver o que está acontecendo no seu projeto! Neste tutorial utilizaremos um DISPLAY DE CRISTAL LÍQUIDO (LCD) modelo **LCD2004**. Além deste dispositivo, existem vários outros que também podem ser utilizados como monitores, telas e displays de informação. Veja abaixo duas fotos muito parecidas com o modelo de LCD que utilizaremos na aula.



FACE SUPERIOR

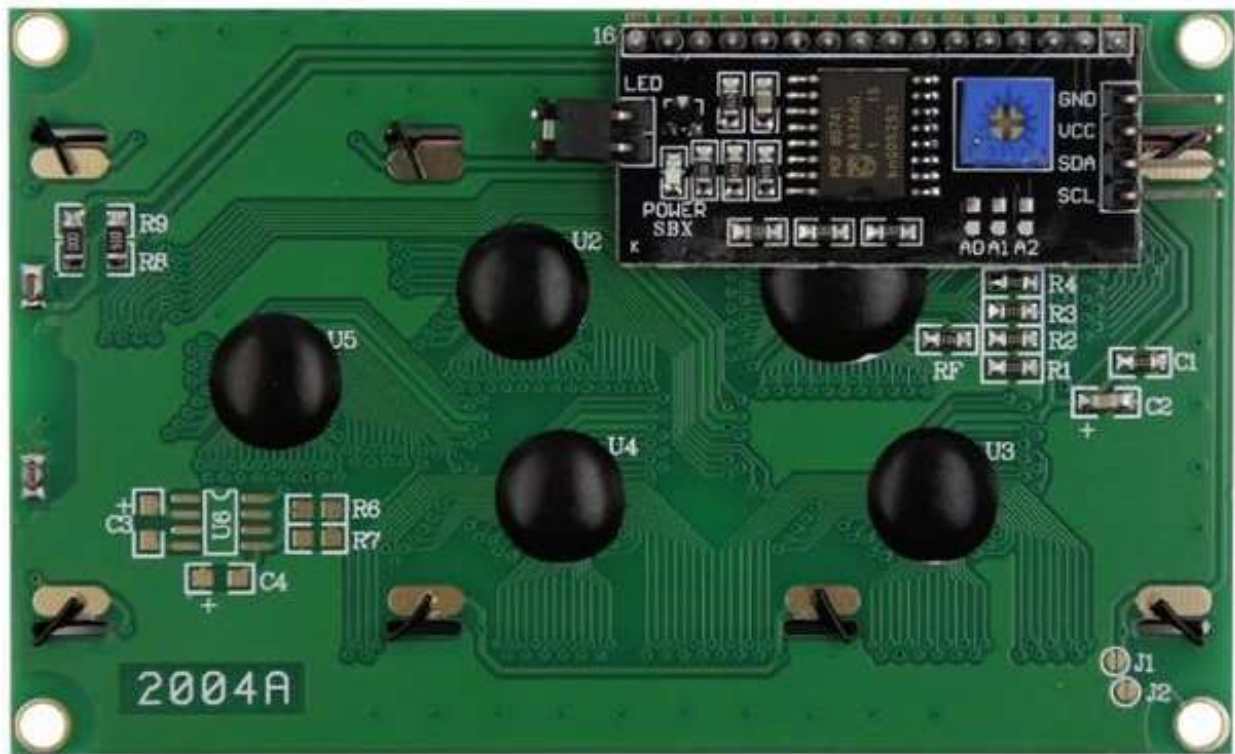


FACE INFERIOR

Atente-se para o fato de que este LCD permite ao usuário escrever até 20 caracteres em cada uma das 4 linhas que ele possui! As letras, números e caracteres permitidos são limitados e pode acontecer que algum caractere específico ou mesmo algum símbolo desejado só poderá ser desenhado no LCD por meio de bibliotecas específicas.

Outro ponto importante é que este dispositivo, a princípio, utiliza até 16 pinos/conexões para funcionar com todas suas funções. **Veja a localização dos pinos na parte superior do dispositivo.** Existe na internet uma gama variada de páginas e exemplos que explicam como utilizar apenas algumas destas portas para funcionamento do dispositivo, diminuindo em um pouco menos da metade o número de portas necessárias para sua utilização. Contudo, o uso de um item acessório com Comunicação I2C (veremos o que isto significa mais a frente)

permite reduzir para 4 o número de pinos necessários para o uso deste LCD. Por este motivo, nesta aula, aprenderemos como conectar e utilizar o **LCD2004** já com seu acessório I2C.



FACE INFERIOR DO **LCD2004** COM ACESSÓRIO I2C JÁ SOLDADO AO LCD

Gaste um tempo na internet procurando por outros modelos de DISPLAYS para plataformas ARDUINO/ESP8266. Procure no GOOGLE, por exemplo, por “LCD DEVICES FOR ARDUINO”. Anote no campo abaixo alguns modelos encontrados e algumas características visuais que parecem ser comuns a todos estes modelos. Pense ainda em algum projeto em que você possa querer utilizar um outro tipo de DISPLAY. Podemos estudá-lo depois!

Note que este dispositivo I2C acessório possui pelo menos 1 pino VCC e 1 pino GND. Se eles possuírem mais de um pino do mesmo tipo (com o mesmo nome), então qualquer um destes pinos pode ser utilizado; afinal, estão conectados internamente um ao outro.

O pino VCC é o pino para fornecer energia ao dispositivo, enquanto o pino GND é o pino de retorno (GROUND). Normalmente, estes dispositivos funcionam com 3.3V ou 5.0V de corrente contínua (DC – Direct Current). Se você colocar uma tensão acima deste valor pode queimar o dispositivo. **Localize estes pinos no seu dispositivo acessório ao DISPLAY.**

Um detalhe importante neste dispositivo é a presença de um quadrado azul. Este item é na verdade uma resistência variável no circuito, e sua função é possibilitar o aumento ou a diminuição do contraste do LCD. Para isto, ao variar a resistência deste dispositivo, a tela do **LCD2004** se ilumina mais ou menos, à critério do usuário. Para variar então a resistência deste componente, o usuário deve utilizar uma chave de fenda pequena ou chave Philips e girar o parafuso localizado no centro deste quadrado, para a direita ou para a esquerda.

Além disto, estes dispositivos possuem também dois outros pinos principais, chamados de pinos SDA e SCL! Estes dois pinos (SDA e SCL) são os pinos/portas/canais/conexões utilizados para que o ARDUINO ou o ESP8266 possam se comunicar com o **LCD2004**. Esta comunicação é feita por meio de uma linguagem de comunicação eletrônica chamada de I2C (i-dois-c) ou I²C (i ao quadrado c). Depois podemos ver alguns detalhes deste tipo de comunicação e quais os princípios em que elas se baseiam. Por hora, devemos entender que o I2C (Inter-Integrated Circuit) é um tipo de comunicação eletrônica desenvolvido pela Philips que é usado para conectar vários dispositivos de baixa velocidade a processadores, sistemas embarcados (ARDUINOS, ESP8266), celulares e outros aparelhos. Esta comunicação se faz, então, por 2 fios que foram denominados de SDA (Serial Data) e SCL (Serial Clock). Portanto, para que vários dispositivos se conectem ao ARDUINO ou ao ESP8266, é necessário que cada um deles apresente também pinos SDA e SCL, possibilitando a conexão ao mesmo circuito/barramento I2C. **Localize estes pinos no dispositivo acessório ao LCD2004.**

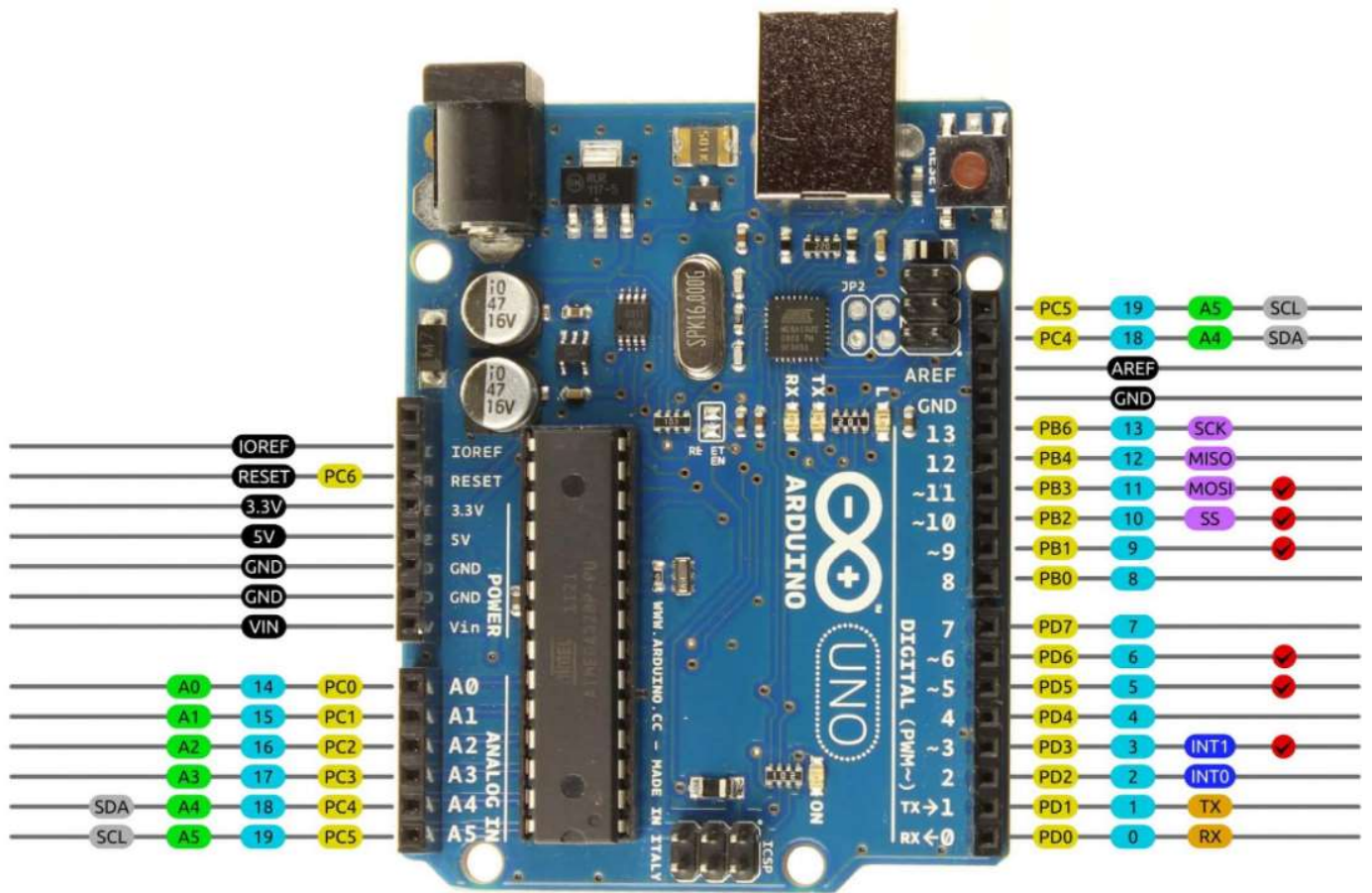
Além disto, cada dispositivo conectado ao circuito I2C possui um endereço eletrônico próprio (uma palavra-chave específica) que permite com que este dispositivo possa ser reconhecido dentre todos os demais, fazendo com que as mensagens sejam entregues especificamente para cada dispositivo sem confusão!

Imaginem por exemplo dois dispositivos (um display LCD e um relógio RTC) que se comunicam com o ARDUINO ou com o ESP8266 por meio do protocolo de comunicação I2C. Eles compartilharão os mesmos pinos/portas/canais SDA e SCL e, portanto, quando o ARDUINO/ESP8266 enviar uma mensagem para o relógio RTC, apenas este dispositivo poderá ler a mensagem, pois somente ele entenderá o conteúdo desta mensagem. Assim, o display LCD não fará nada com esta mensagem. Mas como o display LCD saberá que esta mensagem não é para ele? Ah... para isto, a mensagem enviada pelo ARDUINO/ESP8266 vai com a palavra-chave específica, que identifica cada um dos dispositivos I2C que estão conectados no mesmo circuito I2C.

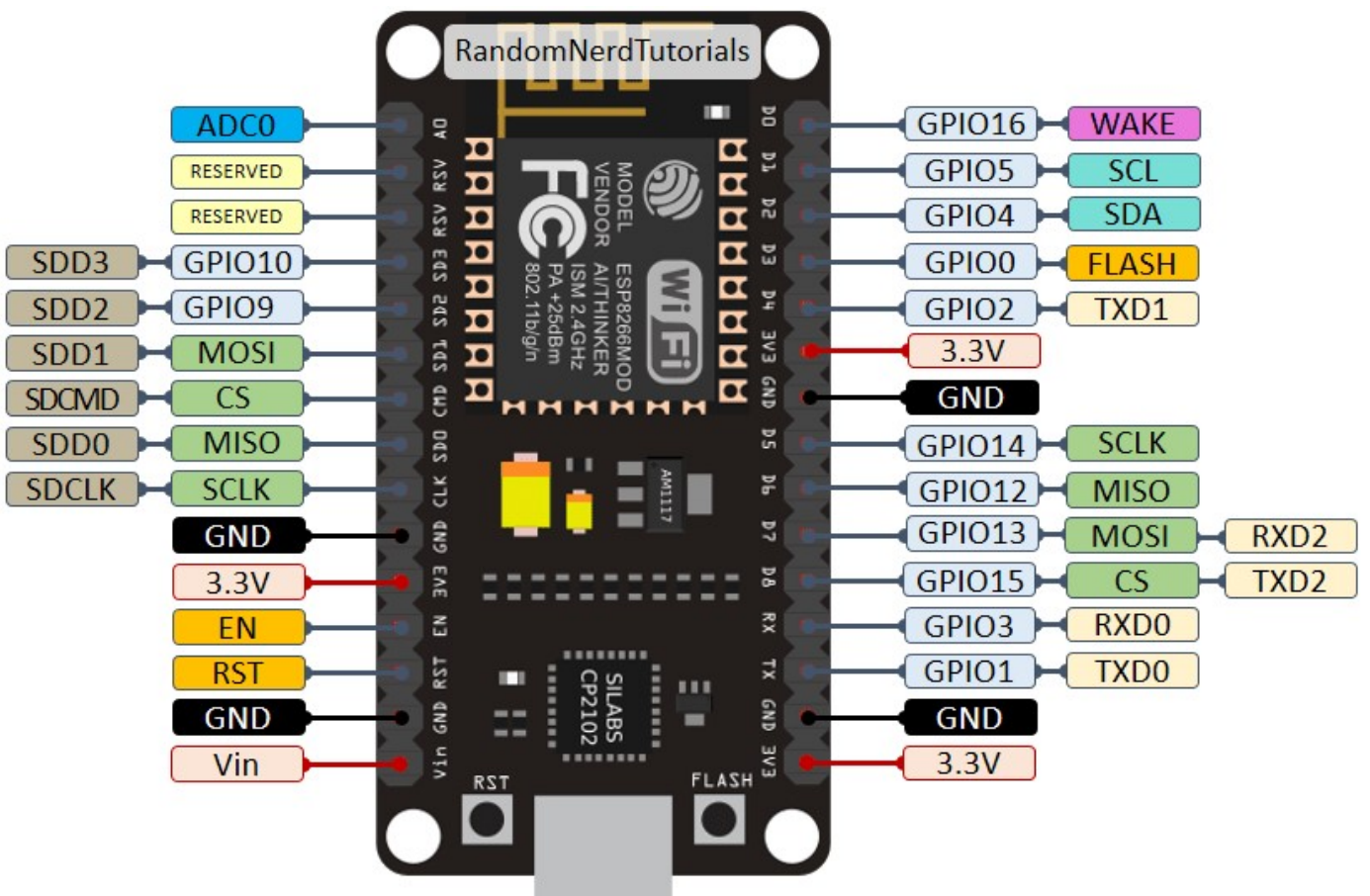
No **LCD2004**, o usuário pode definir até 8 endereços eletrônicos (palavras-chaves) diferentes através das conexões A0, A1 e A2 (localizadas abaixo do “quadrado azul” no acessório). **Localize estas conexões A0, A1 e A2. Podemos discutir um pouco mais sobre elas agora! Pergunte ao professor como elas funcionam e como utilizá-las para definir um endereço específico para este LCD.** A grande vantagem desta configuração independente é que você consegue conectar, por exemplo, até 8 **LCD2004** ao mesmo ARDUINO ou ESP8266 pela comunicação I2C; cada um deles com um endereço (palavra-chave) específico.

Muito bom se você já chegou até aqui! A princípio, com as explicações dadas, você já teria condições de conectar o ARDUINO ou o ESP8266 no seu **LCD2004** (por meio do acessório I2C). **Para isto, localize no ARDUINO ou no ESP8266 quais são os pinos específicos SDA e SCL utilizados por estas plataformas, e os anote no quadro abaixo. Você deve ainda procurar na internet outras plataformas (ESP32, ARDUINO MINI, ARDUINO NANO) e encontrar suas portas SDA e SCL, ok?** Na próxima página eu mostrarei o layout tanto da plataforma ARDUINO UNO quanto da plataforma ESP8266.

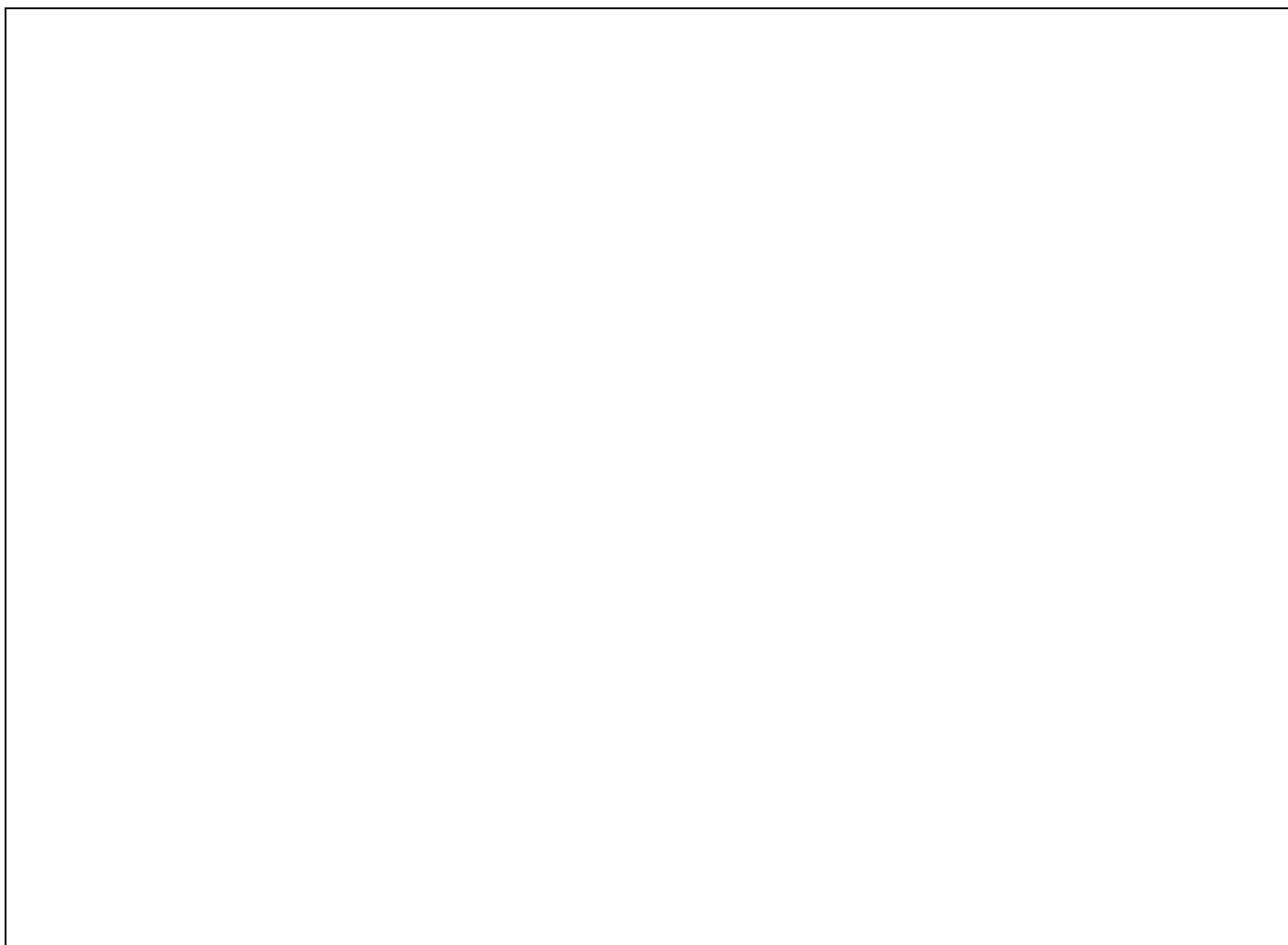
Plataforma Arduino UNO	Plataforma ESP8266
PINO SDA:	PINO SDA:
PINO SCL:	PINO SCL:



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT



Agora, antes de você prosseguir, **tente desenhar abaixo, o circuito eletrônico que você pretende montar com o seu ARDUINO ou com o seu ESP8266 e o dispositivo LCD2004.** Você deve desenhar na figura cada um dos dispositivos com seus pinos mais importantes e fazer linhas (que representarão os fios de contato) unindo os pinos. Você ainda pode representar o computador e o cabo USB que permitirá não apenas energizar seu ARDUINO ou seu ESP8266 como também fazer o UPLOAD do seu programa ao ARDUINO/ESP8266. Só depois de desenhar o circuito que você imaginou, você deve montá-lo fisicamente. **MAS NÃO CONECTE O CABO USB AINDA AO SEU COMPUTADOR!**



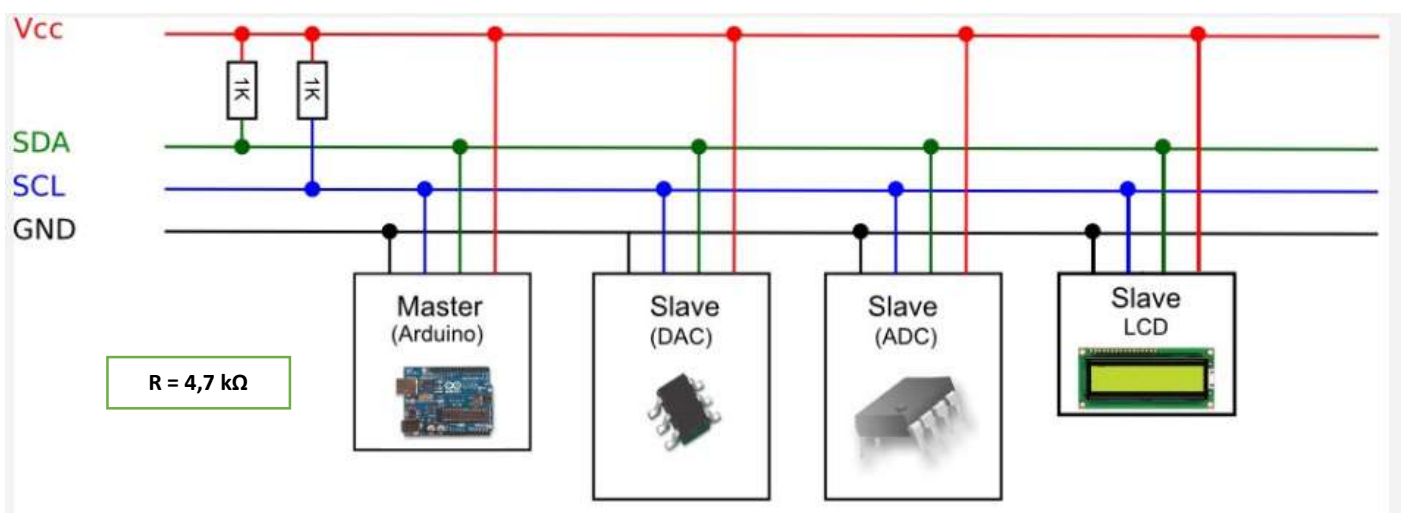
Verifique abaixo se você montou seu circuito corretamente.

1. O cabo USB deve ter uma ponta dele conectada ao ARDUINO ou ao ESP8266 e a outra ponta deveria estar conectada ao seu computador (**mas NÃO ligue ainda**).
2. Além disto, veja as demais conexões abaixo, que devem ser feitas com fios (representados pelo símbolo “-----”).

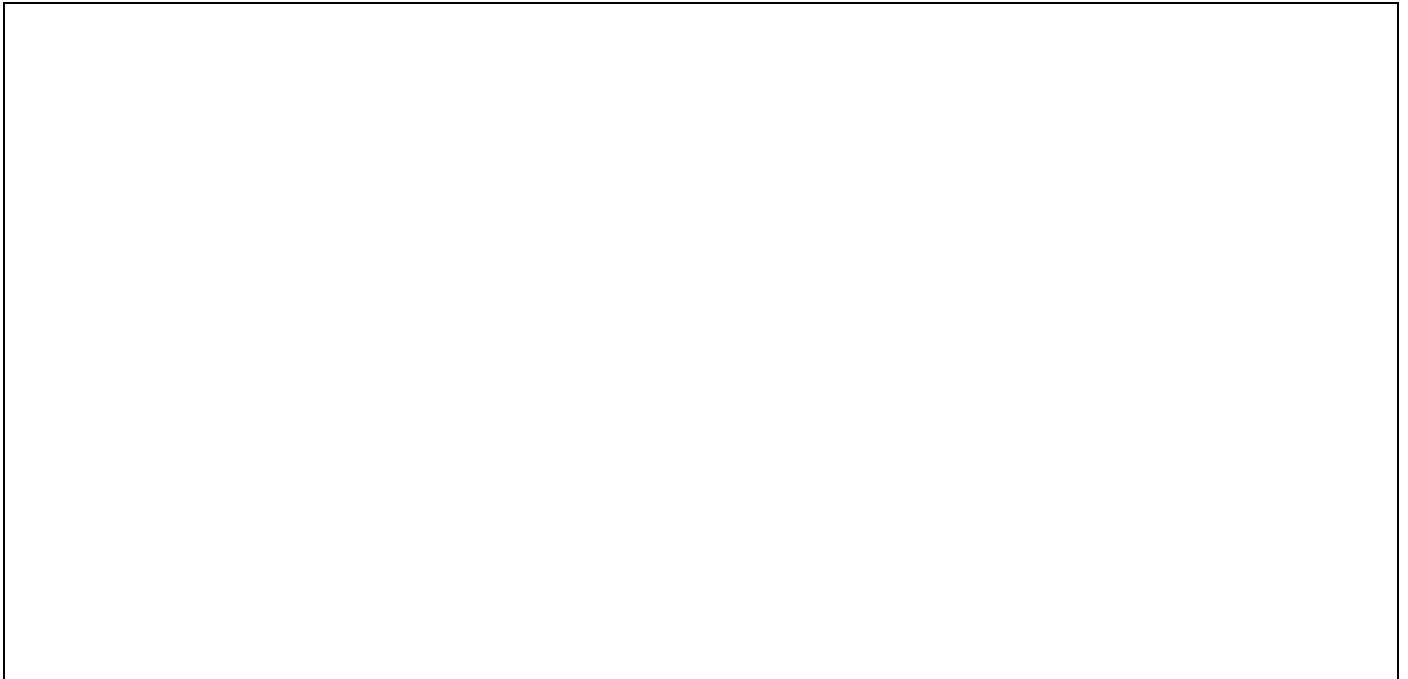
Plataforma Arduino UNO		Plataforma ESP8266	
LCD2004	ARDUINO	LCD2004	ESP8266
Pino SDA	----- Pino SDA	Pino SDA	----- Pino SDA
Pino SCL	----- Pino SCL	Pino SCL	----- Pino SCL
Pino GND	----- Pino GND	Pino GND	----- Pino GND
Pino VCC	----- Pino 5V ou Pino 3.3V	Pino VCC	----- Pino 3.3V

- Note que, desta forma, quem fornece energia ao ARDUINO ou ao ESP8266 é o computador, através do cabo USB. E por sua vez, é o ARDUINO ou o ESP8266 que fornece energia via 5V ou 3.3V para o dispositivo **LCD2004**.
- O ARDUINO ou o ESP8266 se comunicam com o dispositivo **LCD2004** pelos pinos SDA e SCL, via protocolo I2C.

Deve-se atentar ainda para o fato que, quando se utiliza as linhas SDA e SCL para comunicação com alguns dispositivos, incluindo o **LCD2004**, deve-se colocar resistências chamadas de “*pull up resistors*” nestas linhas, conforme a figura abaixo. Na figura, os autores utilizaram resistências cujos valores são de 1 kΩ, mas valores de 4,7 kΩ também são vistos em vários exemplos na internet.



Refaça, no espaço abaixo, o desenho do seu circuito colocando os resistores de 4,7 kΩ em cada uma das linhas SDA e SCL. Só depois de refazer a figura, conecte os resistores ao seu circuito físico. **MAS NÃO CONECTE O CABO USB AINDA AO SEU COMPUTADOR!**

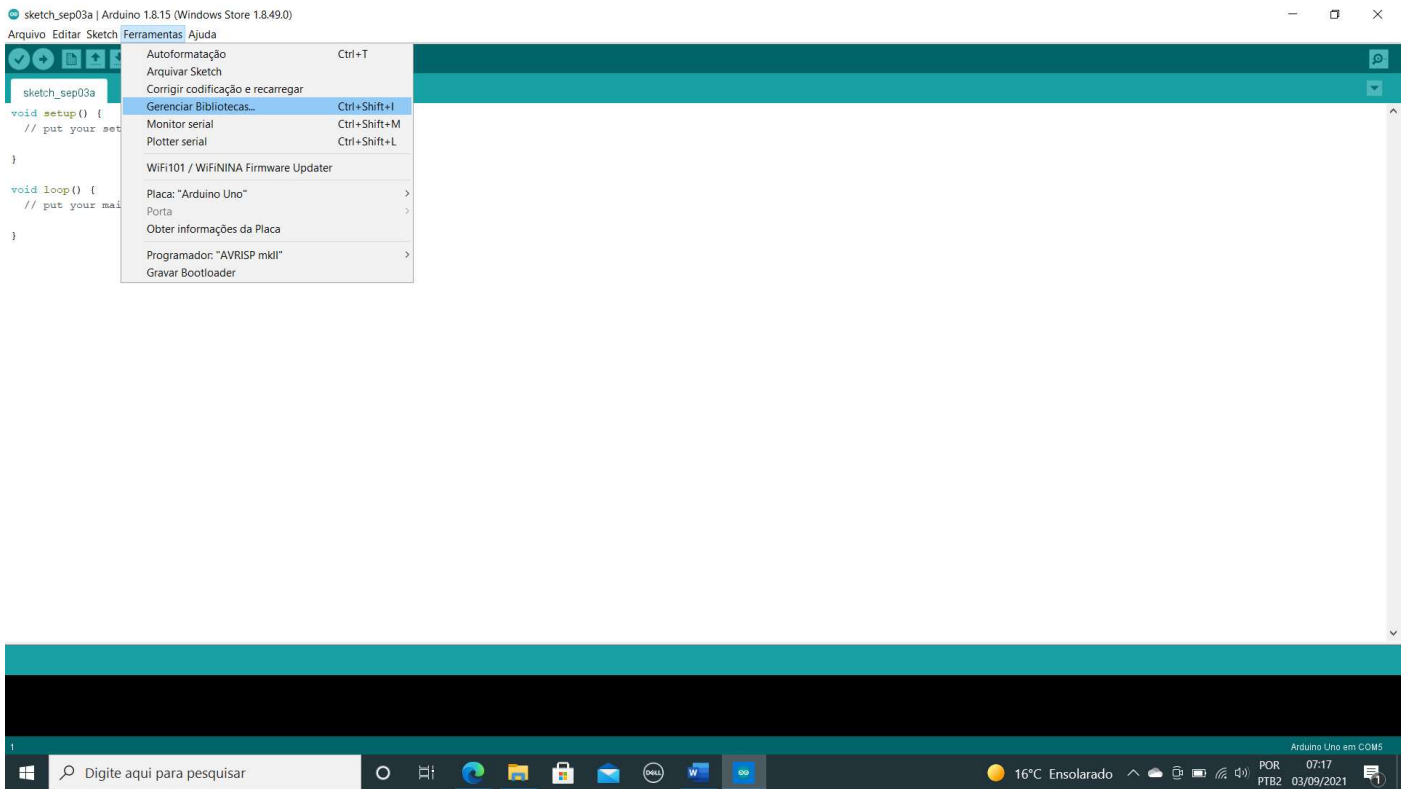


Por fim, é importante que você saiba que é possível energizar o **LCD2004** com uma tensão de entrada de 5V (máximo 5.5V) e esta alimentação pode vir, por exemplo, de uma fonte externa, não necessariamente do ARDUINO ou do ESP8266. Contudo, alguns cuidados devem ser tomados e isto será abordado na hora certa.

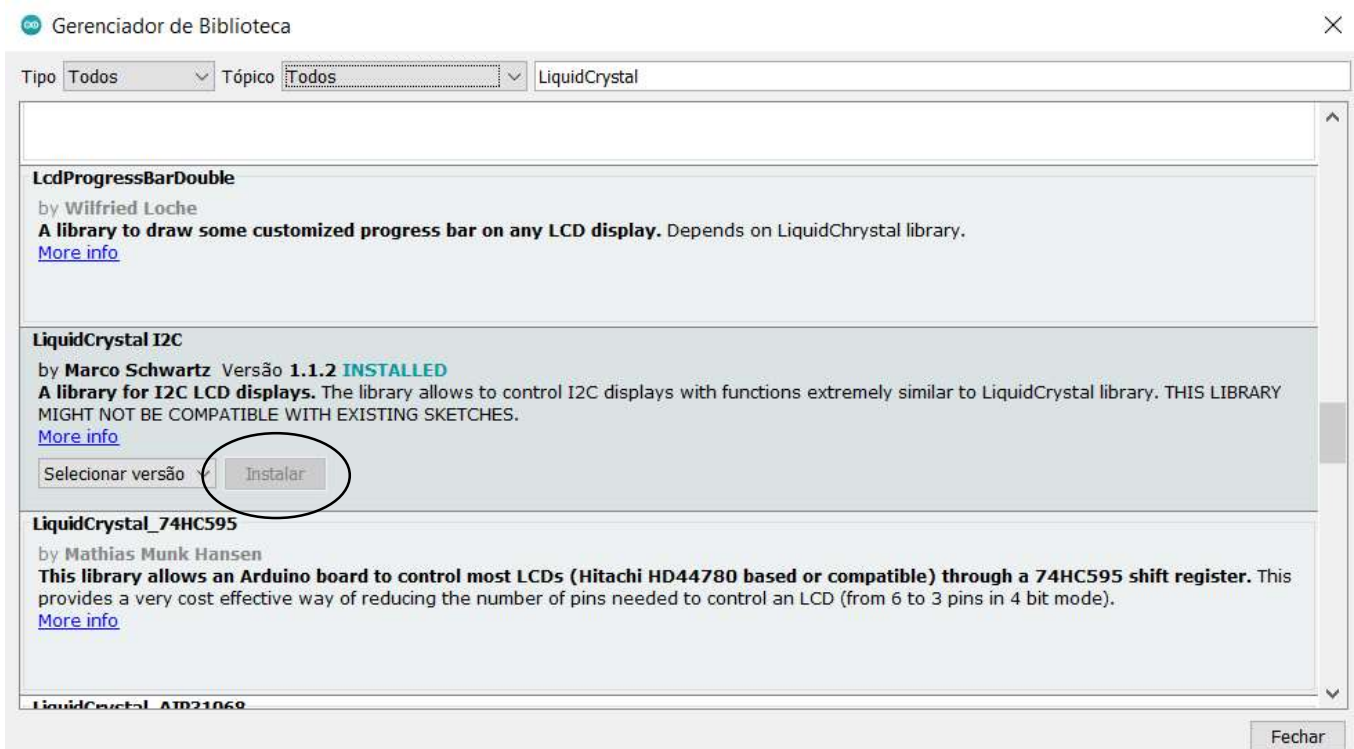
Bom, imaginamos que você esteja pronto para utilizar o dispositivo e assim usar as funcionalidades do **LCD2004**. Mas para fazer isto, você deve ter acesso a uma biblioteca de comandos do ARDUINO/ESP8266 que é capaz de permitir a comunicação com o dispositivo através de comandos de programação simples que podem ser utilizados em qualquer programa que você fizer.

Existem várias bibliotecas que foram construídas por usuários em todo o mundo, mas a que utilizaremos aqui se chama “**LiquidCrystal_I2C**” e pode ser baixada diretamente no SOFTWARE IDE DO ARDUINO. Veremos abaixo como instalar esta biblioteca e como utilizar as funcionalidades do **LCD2004**.

Para instalar as bibliotecas no ARDUINO IDE abra o aplicativo ARDUINO IDE no Windows e logo na primeira tela clique na opção “FERRAMENTAS > GERENCIAR BIBLIOTECAS...”. **Veja a figura a seguir.**



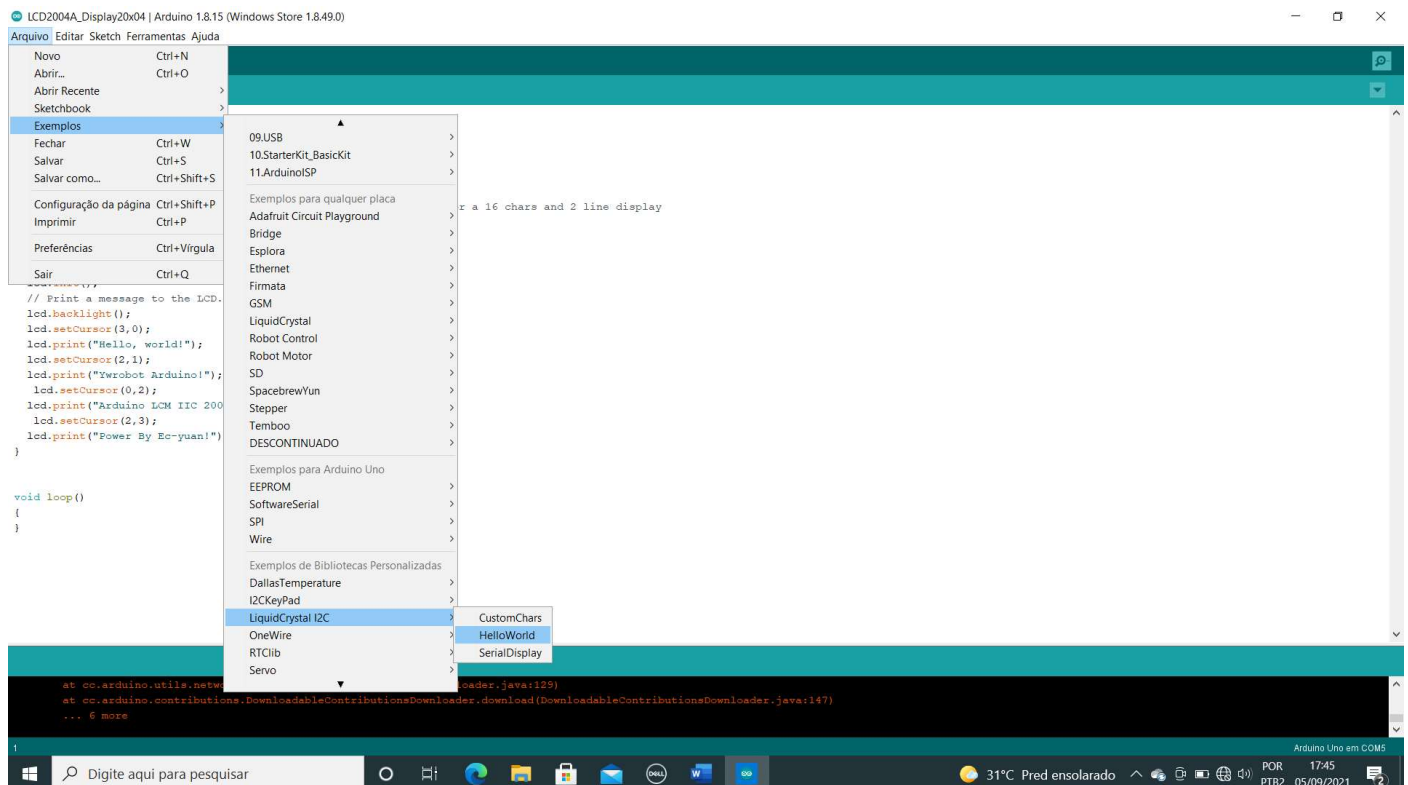
Isto abrirá uma tela que permite a você verificar, instalar e atualizar todas as bibliotecas que você já tem ou deseja instalar em seu ARDUINO IDE. Nós procuraremos pela biblioteca “**LiquidCrystal_I2C**”. Note que o termo LiquidCrystal identifica o tipo do display. Já o termo I2C informa que a biblioteca utiliza o protocolo I2C de comunicação entre os dispositivos.



Se você já possui esta biblioteca instalada em seu computador, ela aparecerá em sua tela com o detalhe **“INSTALLED”** (que significa “instalado”) escrito à frente da biblioteca. Caso contrário, você deve então selecionar a biblioteca que deseja instalar, escolher a versão e clicar em **INSTALAR!** **Veja se você consegue instalar a biblioteca LiquidCrystal_I2C, que foi desenvolvida por Marco Schwartz (um usuário do ARDUINO IDE). Atualmente a versão dela é a 1.1.2.**

Este procedimento para instalar bibliotecas é muito útil pois várias bibliotecas já foram catalogadas pelo ARDUINO IDE e fazem parte de seu pacote de bibliotecas. Contudo, caso você descubra alguma biblioteca nova, mas que ainda não foi incorporada ao ARDUINO IDE, você pode instalar esta biblioteca também; copiando toda a pasta de arquivos da biblioteca para um local específico das pastas do SOFTWARE ARDUINO IDE.

Depois que a biblioteca foi instalada, você poderá verificar, clicando em **ARQUIVO > EXEMPLOS**, que uma nova opção chamada **LiquidCrystal I2C** foi criada. Clicando ainda em **LiquidCrystal I2c** você poderá ver vários exemplos de programas que foram feitos para utilizar a biblioteca LiquidCrystal_I2C em conjunto com displays do tipo **LCD2004**. Apenas a título de curiosidade esta biblioteca também pode ser utilizada em dispositivos tipo **LCD1602**.



Você pode escolher qualquer um dos exemplos da última coluna (na figura acima) e testá-los para ver como eles funcionam. Você, certamente, precisará antes disto conectar o dispositivo **LCD2004** corretamente, ligar o ARDUINO/ESP8266 ao computador pelo cabo USB e fazer o UPLOAD deste arquivo para o ARDUINO/ESP8266 (**comando da “seta”**). Em alguns casos você precisará acessar o Monitor Serial para ver o funcionamento do programa (**comando da “lupa”**).

Vamos usar inicialmente o exemplo **“HelloWorld”**. **Abra o exemplo no ARDUINO IDE e tente identificar o significado de algumas linhas de programação que lhe parecem importantes. Anote suas ideias no quadro abaixo. Somente depois disto faça o upload do programa e veja os resultados.**

```
//Displays text on LCD

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 20 chars and 4 line display

void setup()
{
  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Hello, world!");
  lcd.setCursor(2,1);
  lcd.print("Ywrobot Arduino!");
  lcd.setCursor(0,2);
  lcd.print("Arduino LCM IIC 2004");
  lcd.setCursor(2,3);
  lcd.print("Power By Ec-yuan!");
}

void loop()
{
}

//Program end
```

Repita a análise e upload, mas agora usando o exemplo “SerialDisplay”. Você precisará utilizar o Monitor Serial e digitar algum caractere para ser enviado ao seu dispositivo (ARDUINO ou ESP8266).

```
//Displays text sent over the serial port (e.g. from the Serial Monitor) on an attached LCD.

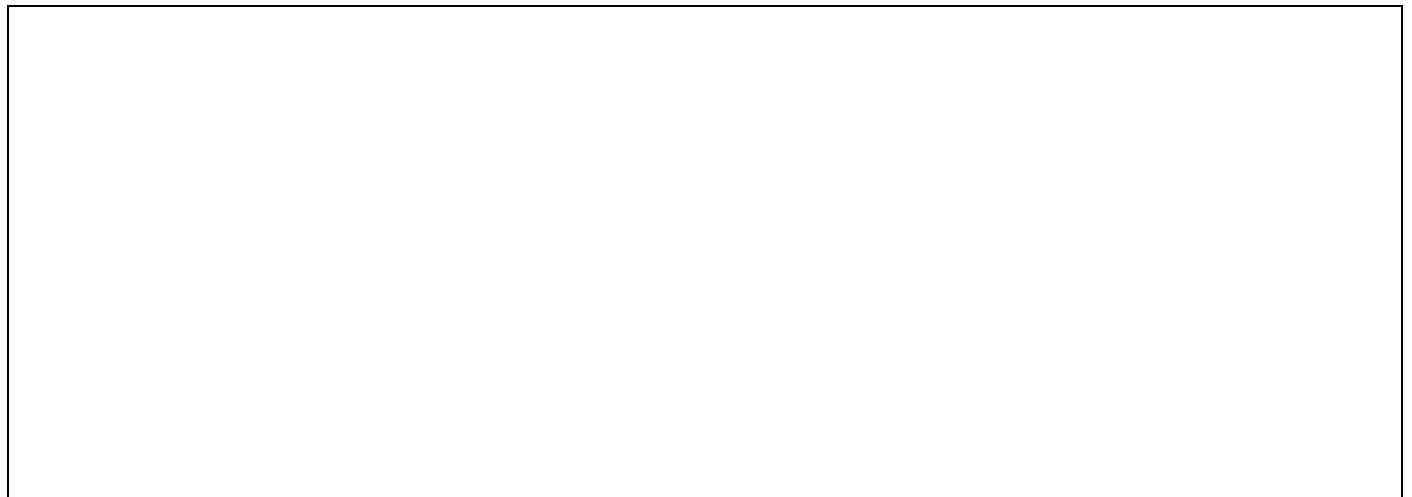
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()
{
  lcd.init(); // initialize the lcd
  lcd.backlight();
  Serial.begin(9600);
}

void loop()
{
  // when characters arrive over the serial port...
  if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}

//Program end
```



Observações pertinentes:

- 1) Todas as linhas começadas com “//” são linhas de comentário e podem ser excluídas do programa sem prejuízo ao seu funcionamento. Contudo, estes comentários são importantes para você programador, já que estes comentários são importantes para você saber o que está sendo programado e para lembrá-lo do que foi feito na programação!

- 2) Note que antes do início do programa; ou seja, antes do comando “void setup {}” existem três linhas de comando fundamentais, que devem constar em todos os programas que vocês utilizarem este dispositivo:

```
#include <Wire.h>
```

Esta linha de comando informa ao seu programa que ele precisa incluir a biblioteca <Wire.h>, necessária para que a próxima biblioteca possa ser utilizada.

```
#include <LiquidCrystal_I2C.h>
```

Esta linha de comando informa ao seu programa que ele precisa incluir a biblioteca <LiquidCrystal_I2C.h>, necessária para que você possa acessar as funcionalidades do LCD que você está utilizando.

```
LiquidCrystal_I2C lcd(0x27,20,4);
```

Esta linha de comando informa ao seu programa para ele criar um objeto virtual no programa chamado “lcd”. Você poderia chamá-lo, se quisesse, de “mylcd” mas o comando seria: `LiquidCrystal_I2C mylcd(0x27,20,4);`

Este objeto (“lcd”) representa o display físico que está instalado no seu circuito cujo endereço eletrônico (ou palavra-chave) é igual a “0x27”, possui “20” colunas e “4” linhas de texto! **E se fosse um LCD de 16 colunas e 2 linhas; como seria o comando acima?**

- 3) Dentro da programação propriamente dita, encontramos outros códigos importantes:

```
lcd.init(); [ mylcd.init(); ]
```

Utilizado normalmente dentro do “void setup {}” este comando inicia o objeto virtual.

```
lcd.backlight(); [ mylcd.backlight(); ]
```

Liga a luz de fundo do LCD.

```
lcd.clear(); [ mylcd.clear(); ]
```

Apaga todo o conteúdo que está sendo mostrado no LCD.

```
lcd.setCursor(x,y); [ mylcd.setCursor(x,y); ]
```

Posiciona o cursor na coluna “x” e na linha “y” do LCD, para imprimir algum caractere ou texto a partir desta posição.

```
lcd.print("text here"); [ mylcd.print("text here"); ]
```

Imprime no LCD a mensagem “text here”.

Vamos escrever abaixo outras observações importantes.