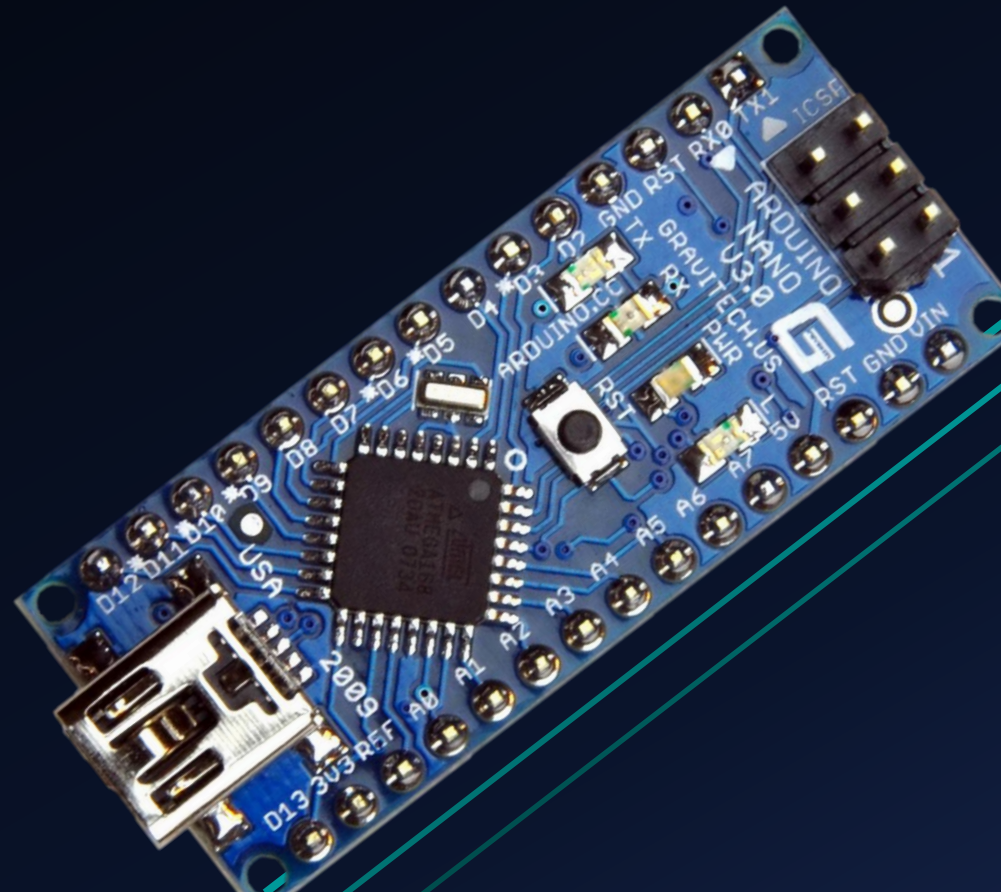


Gabriel de Melo Araújo  
Lucas Nunes  
Max Rodrigues  
Vitor Paiva

PET Engenharia Elétrica – UFRN  
ufrnpetee@gmail.com  
facebook.com/ufrnpetee

# MINICURSO DE ARDUÍNO

## AULA 04



# SUMÁRIO

- LCD
- Projeto
  - Ligações
  - Explicando o que cada pino faz
  - Código
  - Caracteres personalizados

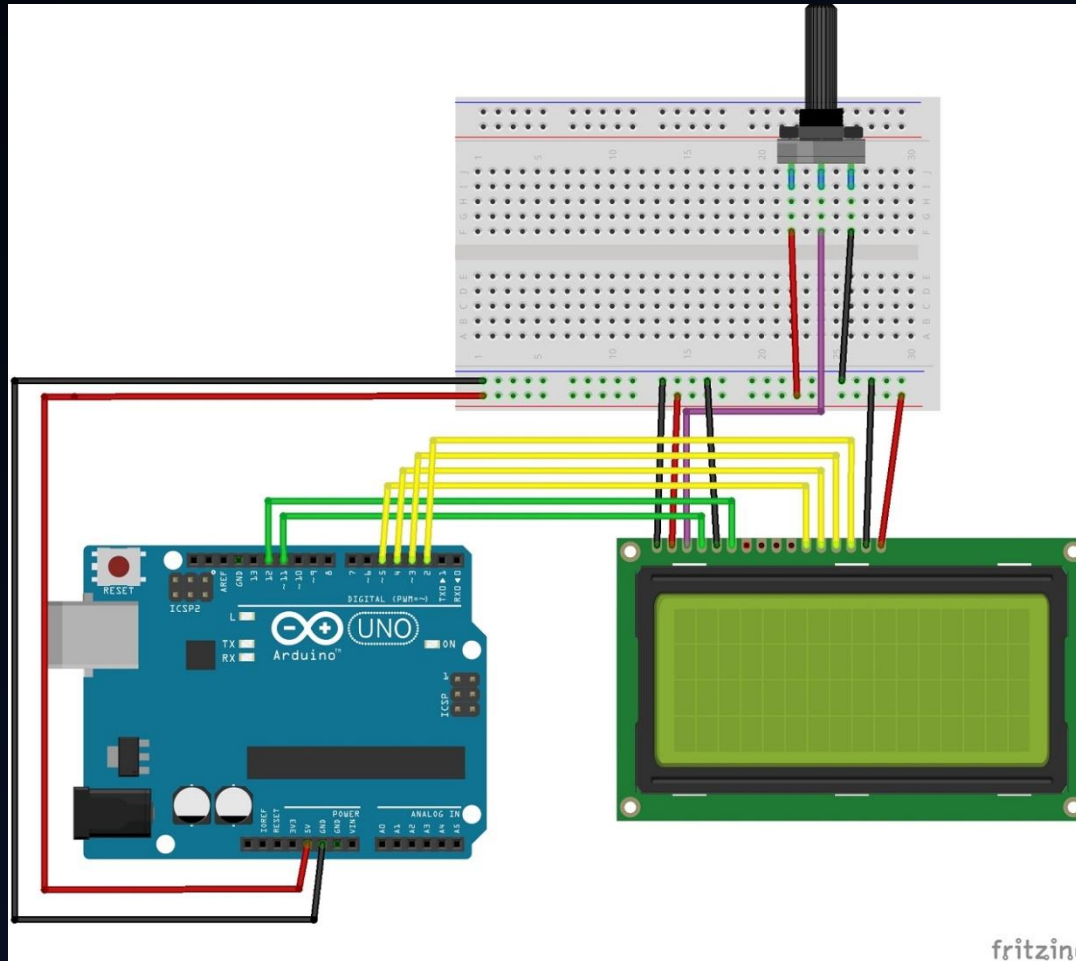
# LCD

Método muito comum de exibição de textos e símbolos, os LCDs (Liquid Crystal Display, ou display de cristal líquido). LCDs são displays tipicamente utilizados em calculadoras e despertadores. Displays de LCD requerem chips controladores para controlá-los; esses chips são integrados ao display.

A criação de projetos com base em displays LCD é muito simples, graças a um conjunto prontamente disponível de bibliotecas de código para LCDs. O IDE do Arduíno vem com uma biblioteca desse tipo, a *LiquidCrystal.h*, que tem uma grande lista de recursos. Essa é a biblioteca que nós utilizaremos em nossos projetos.



# PROJETO



# PROJETO

Pino	Símbolo	Função
1	VSS	GND(Alimentação)
2	VDD	5V(Alimentação)
3	V0	Ajuste de Contraste
4	RS	Habilida/Desabilita Seletor de Registrador
5	R/W	Leitura/Escrita
6	E	Habilita Escrita no LCD
7	DB0	Dado
8	DB1	Dado
9	DB2	Dado
10	DB3	Dado
11	DB4	Dado
12	DB5	Dado
13	DB6	Dado
14	DB7	Dado
15	A	5V(Backlight)
16	K	GND(BackLight)

# Explicando o que cada pino faz

- Os Pinos 1 e 2 servem para alimentar o dispositivo.
- O pino 3 serve para controlar o contraste da tela. Ele recebe um sinal de 0 a +5Vcc controlado por um potenciômetro de 10 KΩ ou 20KΩ.
- Os pinos 4 e 6, possuem a função de definir se o LCD vai receber um comando (setar cursor/ limpar) quando está em nível lógico baixo (0) ou se vai receber um caractere(a,b...1,2...) podendo estar ativo ou inativo, isso quando está em nível lógico alto (1). Mas isso não é um problema que devemos levar em consideração, pois a biblioteca que vamos utilizar na programação do Arduino fica responsável de fazer toda a parte de desenvolvimento de baixo nível.
- O pino 5, é responsável por controlar se a operação vai ler o dispositivo ou escrever no mesmo.

## Explicando o que cada pino faz

- Os pinos de 7 a 14, são responsáveis pelo barramento de dados. Existe duas formas de se utilizar esses 8 pinos: A primeira e menos usada, é quando se utiliza todos eles em paralelo para fazer a comunicação em 8 bits (ou 1 byte). A mais utilizada e que vai estar presente em nossa programação é a com 4 pinos, que transmite os dados em dois pacotes, sendo cada um com 4 bits (ou 1 nibble). O motivo de usarmos esse, é que reduz o numero de pinos de entrada de saída do Arduino, porem gasta mais memória.
- Os pinos 15 e 16, são responsáveis pelo acendimento do Back-Light.





## CARACTERES PERSONALIZADOS

A maioria dos LCDs permite que você programe seus próprios caracteres personalizados. Um LCD normal de 16 x 2 tem espaço para armazenar oito caracteres personalizados na memória. Os caracteres têm 5 pixels de largura e 8 pixels de altura.

```
byte sad[8] = {  
    B00000,  
    B00000,  
    B10001,  
    B00000,  
    B01110,  
    B10001,  
    B00000,  
    B00000  
};
```

```
byte happy[8] = {  
    B00000,  
    B00000,  
    B10001,  
    B00000,  
    B10001,  
    B01110,  
    B00000,  
    B00000  
};
```

# CARACTERES PERSONALIZADOS

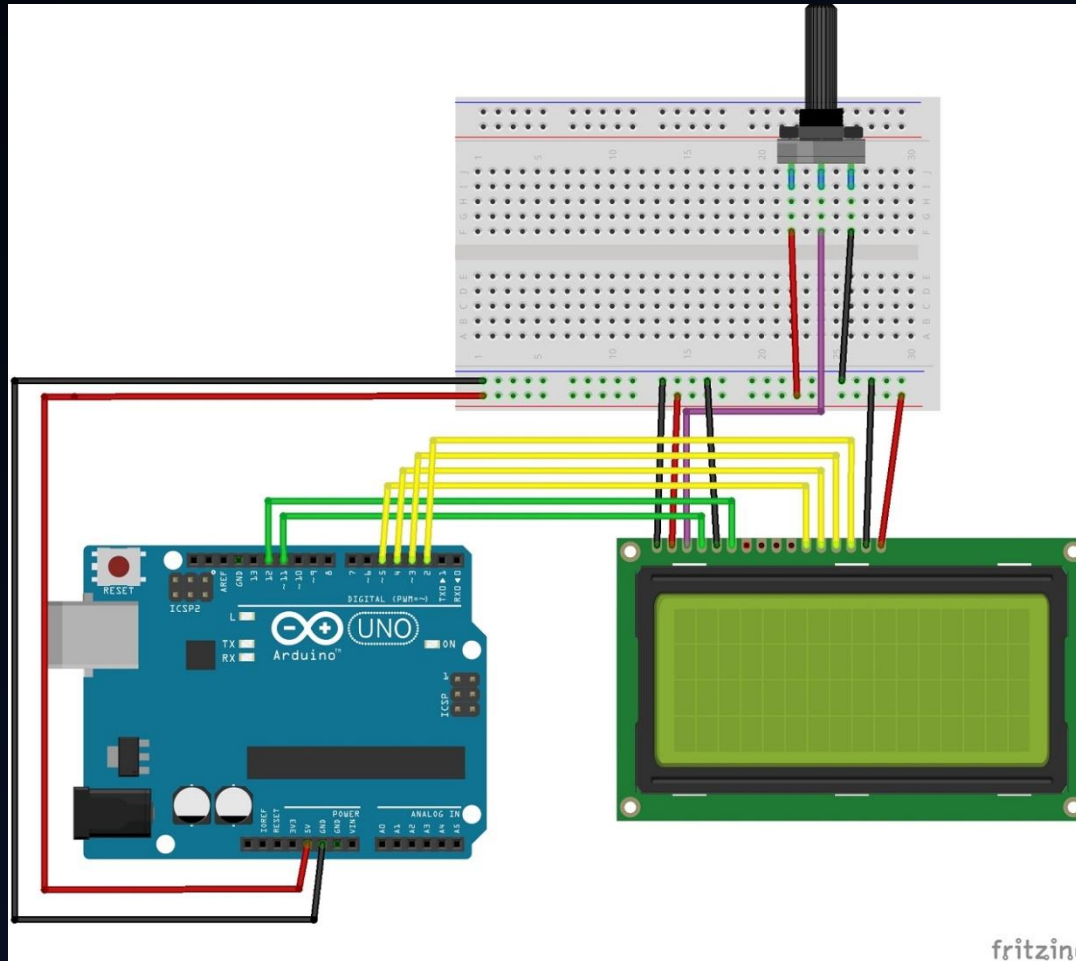
- `lcd.createChar(1, sad);`
- `lcd.createChar(2, happy);`

```
for(int x=0; x<5; x++) {  
    lcd.setCursor(8,0);  
    lcd.write(byte(1));  
    delay(1000);  
    lcd.setCursor(8,0);  
    lcd.write(byte(2));  
    delay(1000);  
}
```

## Exercício

- Escrever “Hello World!” no lcd;
- Colocar 😊;
- Colocar outro símbolo à sua escolha;
- Mostrar o cursor;
- Rolar o texto para a esquerda/direita;

# PROJETO



## Recebendo dados da porta Serial;

- O Arduino pode comunicar-se com o computador não apenas para enviar dados, como também recebe-los. Podemos usar a porta serial para enviar dados para o arduino e fazer com que este imprima-os no display de lcd.
- `Serial.read();`
- `Serial.available();`
- `Serial.parseInt();`
- `Serial.parseFloat();`
- `Serial.peek();`
- `Serial.readString();`
- `Serial.readByte();`

## Esquematisação e Código

- O esquema é o mesmo do exemplo anterior.
- Código:
- Inicie o serial;
- Inicie o display de lcd;
- Se o serial estiver disponível, leia caracter a caracter;
- Imprima os caracteres no display de lcd;
- Bônus: faça o arduino pular as linhas automaticamente