



AULA 03

- Monitorar os dados recebidos
 - Mandar para um LCD
- Resistor = controlar corrente
- Zener = regular tensão
- LCD
 - GND = NEGATIVO
 - VO = contraste
 - RS, RW, ENABLE, DATA
 - DB0 ATÉ DB07
 - Por eles envia um byte
 - código ascii
 - imprimir letra 'a', mandar código ascii da letra a
 - também podem identificar um comando
 - comando 1 para apagar a tela, por exemplo
 - Através do RS

- Seletor da função do LCD
- Dependendo do estado, instrui como ele deve interpretar o byte
- Enable
 - Display: executa instrução dependendo do RS
- RW
 - Informar se quer escrever algo no display ou fazer uma leitura
 - Fixar em ground para informar que só vamos escrever nele
- Sem o contraste, não da pra ver nada no display
- Não usar pinos 0 e 1 para RS, usados para comunicação serial
- Multiplexar o dado que vem pelo barramento
- Em vez de mandar para o databus do display um dado de 8 bits, manda 4 bits e depois mais 4 bits
- Fazer o display funcionar
 - Precisa de uma biblioteca
 - <https://docs.arduino.cc/learn/electronics/lcd-displays/>
 - `#include <LiquidCrystal.h>`
 - Informar quais são as IOS para controlar o display (construtor)
 - <https://maxpromer.github.io/LCD-Character-Creator/> para criar custom emoji por exemplo
- Forma mais conhecida de mandar dados para fora: abrir serial
- Mandar do terminal para o arduino
 - Exemplo: mandar mensagem "LED ACENDA"

- Arduino acende led
- Toda vez que o arduino recebe um caracter ele gera uma interrupcao
 - Mesmo que esteja ocupado, ele vai receber e guardar em uma memória temporada
 - Serial.available() vai retornar quantos bytes estão enfileirados
 - Se for maior que 0, tem dado enfileirado para pegar
 - Como pega?
 - Chama Serial.read()
 - ATMEGA 328 tem 2k de memória ram
 - Não pode abusar na quantidade de caracteres que ele pode memorizar por interrupção