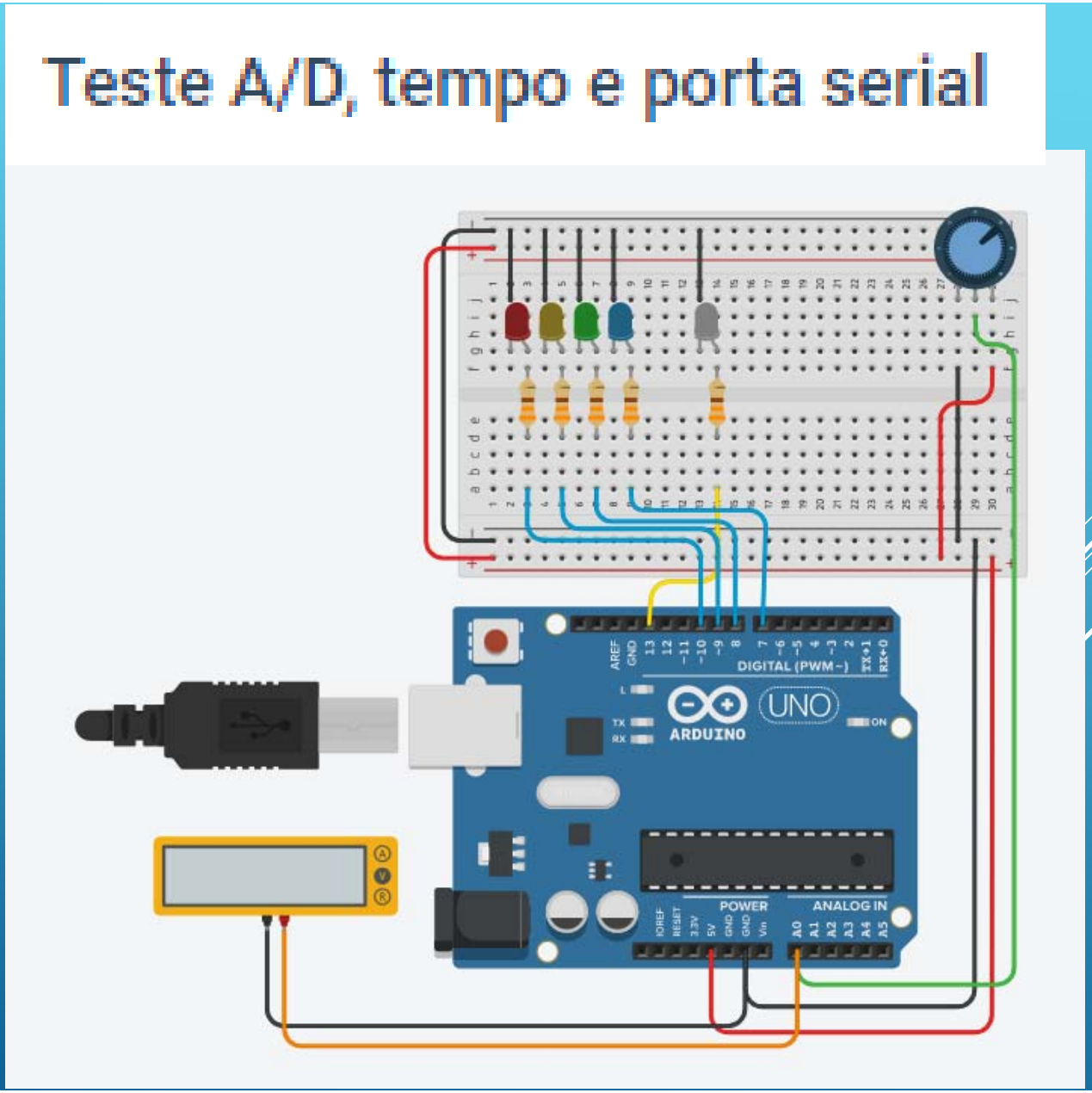
SISTEMAS EMBARCADOS APLICADOS I – SETUP BÁSICO E LEITURA ANALÓGICA

Prof. Dr. Dalton Vidor

ESTRUTURA BÁSICA - MICROCONTROLADORES:

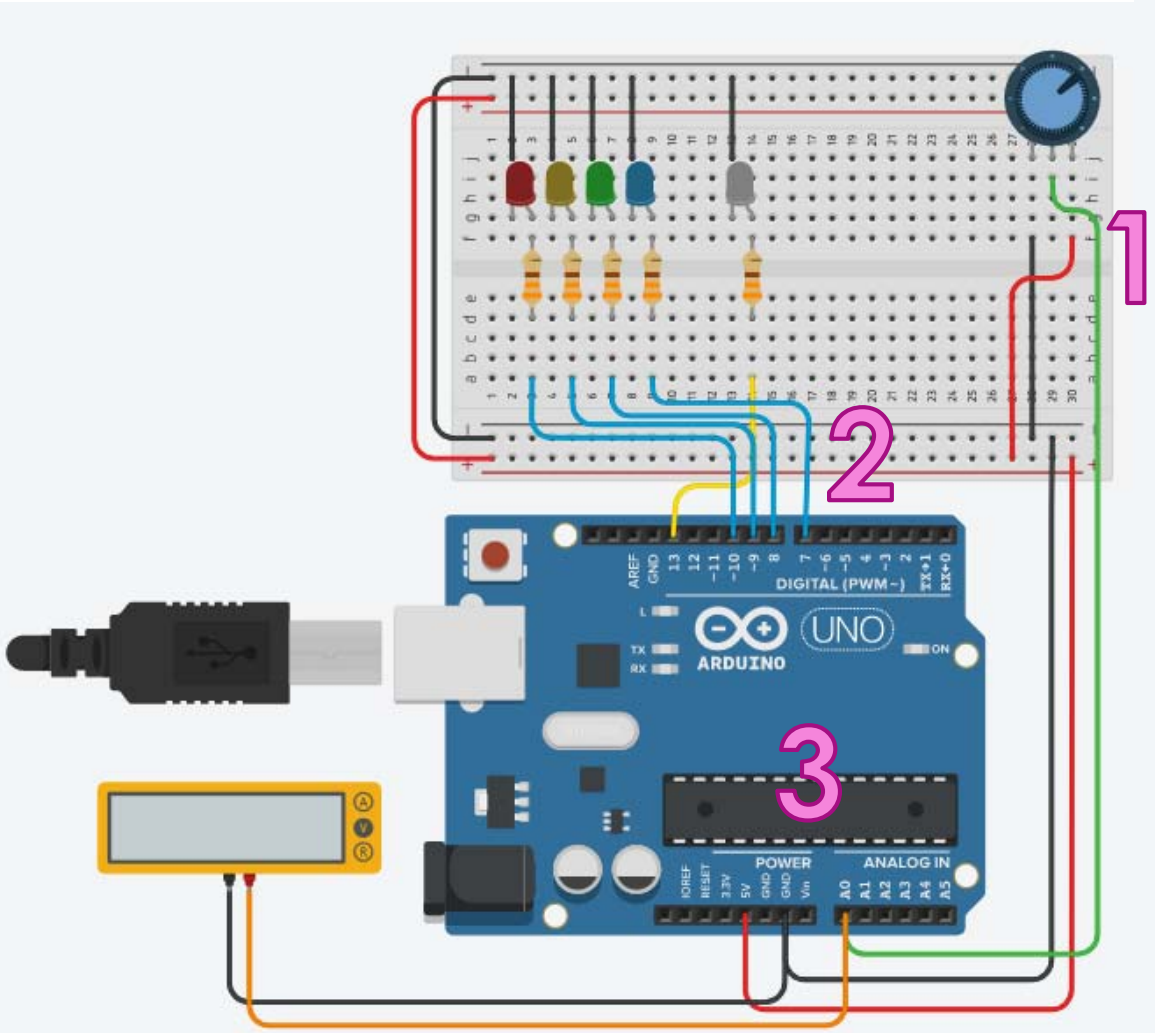
https://www.tinkercad.com/things/l67TBN3i7Vn-teste-ad-tempo-e-porta-serial



ESTRUTURA BÁSICA – ARDUINO UNO:

1. Leitura analógica entre 0 e 5V na porta analógica A0 conforme limite do microcontrolador.
2. Saídas digitais – nível lógico 0(false – falso) produz 0V e nível lógico 1 (true – verdadeiro) produz 5V.
3. Funções: LOOP (infinito), comunicação serial com o usuário, analogRead para leitura do sinal analógico, MAP para acertar escala, DELAY para consumir um tempo definido, teste/decisão IF e teste/decisão SWITCH-CASE,

Teste A/D, tempo e porta serial



ESTRUTURA BÁSICA – ARDUINO UNO:

- Software:
  - Definição dos pinos de saída e variável de contagem de eventos.
  - Esta opção ocupa posições de memória!!!
  - A utilização de

#define ledPin 13

#define Led1 7

... Só ocupa espaço no arquivo do código e dá trabalho ao compilador.

```
// INICIA COM AS DEFINIÇÕES DE VARIÁVEIS
// LED conectado ao pino digital 13
int ledPin = 13;
// contador para definir tarefas
int conta = 0;
const int Led1 = 7;
const int Led2 = 8;
const int Led3 = 9;
const int Led4 = 10;
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- Software:
  - Void SETUP () {
  - }

  - Configuração do sistema e necessária!
  - Serve também para a inicialização dos dispositivos ou hardware.
  - Roda antes do Loop infinito é a inicialização das configurações.

```
// SEMPRE TEM UM SETUP PARA DEFINIR HARDWARE E PINOS
void setup() {
// configura pinos de TX e RX como comunicação serial com taxa de 9600bps
Serial.begin(9600);
// configura pino digital como saída
pinMode(ledPin, OUTPUT);

pinMode(Led1, OUTPUT);
pinMode(Led2, OUTPUT);
pinMode(Led3, OUTPUT);
pinMode(Led4, OUTPUT);

}
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- Software:
  - Void LOOP () {
  - } o laço infinito que repete o código...
- AnalogRead(pino) é o comando para ler sinal analógico.
- A função MAP converte o valor analógico em outra faixa específica (somente valores inteiros).
- Outra opção é realizar a fórmula matemática para a conversão.

```
// LAÇO INFINITO PARA RODAR ETERNAMENTE O CÓDIGO PRINCIPAL
void loop() {
  // ********Código da leitura AD*********
  // faz a leitura analogical do pino 0:
  int sensorValue = analogRead(A0);
  // Converte a leitura analógica (que vai de 0 - 1023) para um valor de tensão (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // escreve o valor de tensão via comunicação serial:
  Serial.print("Valor real lido no pino   A0 = ");
  Serial.println(voltage);
  int voltage2 = map(sensorValue, 0, 1023, 0, 500);
  Serial.print("Valor real convertido c/ map = ");
  Serial.println(voltage2);
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- Serial. Print envia o texto ou variável via porta serial que já foi configurada no SETUP.
- Utilizando Println adiciona-se um “ENTER” ou “New Line” – nova linha (ou printLN de linha nova).
- Os caracteres são enviados em código ASCII.

```
// LAÇO INFINITO PARA RODAR ETERNAMENTE O CÓDIGO PRINCIPAL
void loop() {
  // ********Código da leitura AD*********
  // faz a leitura analogical do pino 0:
  int sensorValue = analogRead(A0);
  // Converte a leitura analógica (que vai de 0 - 1023) para um valor de tensão (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // escreve o valor de tensão via comunicação serial:
  Serial.print("Valor real lido no pino   A0 = ");
  Serial.println(voltage);
  int voltage2 = map(sensorValue, 0, 1023, 0, 500);
  Serial.print("Valor real convertido c/ map = ");
  Serial.println(voltage2);
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- A opção 1 para perder tempo é a função DELAY(valor em ms) com um tempo grande.
- Nesta situação o processador fica “preso” gastando tempo e não observando outras ocorrências (a não ser que houvesse interrupção).
- Inadequado para software de alta performance...

```
// ********Código do Pisca-Pisca********
 // OPÇÃO 1
 /*
 digitalWrite(ledPin, HIGH);   // liga o LED
 delay(500);                   // temporiza 1 segundo
 digitalWrite(ledPin, LOW);    // desliga o LED
 delay(500);                   // aguarda mais um segundo
 */
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- A opção 2 que utiliza uma contagem, permite ao dispositivo observar outras ocorrências.
- A função IF testa se o valor da contagem está abaixo, acima ou igual a determinados valores, ligando ou desligando o led e estabelecendo um valor máximo para a contagem.
- RTOS – real time operational system - em microcontroladores utilizam técnica similar – falar a respeito.

```
// OPÇÃO 2
 // liga o LED
 if (conta < 10)   digitalWrite(ledPin, HIGH);
 // desliga o LED
 if (conta >=10)  digitalWrite(ledPin, LOW);
 // aguarda mais 10 milisegundos
 delay(500);
 // incrementa a variável conta
 conta++;
 // limita o valor da variável conta em 200 com o
teste
 if (conta >= 20) conta =0;
```

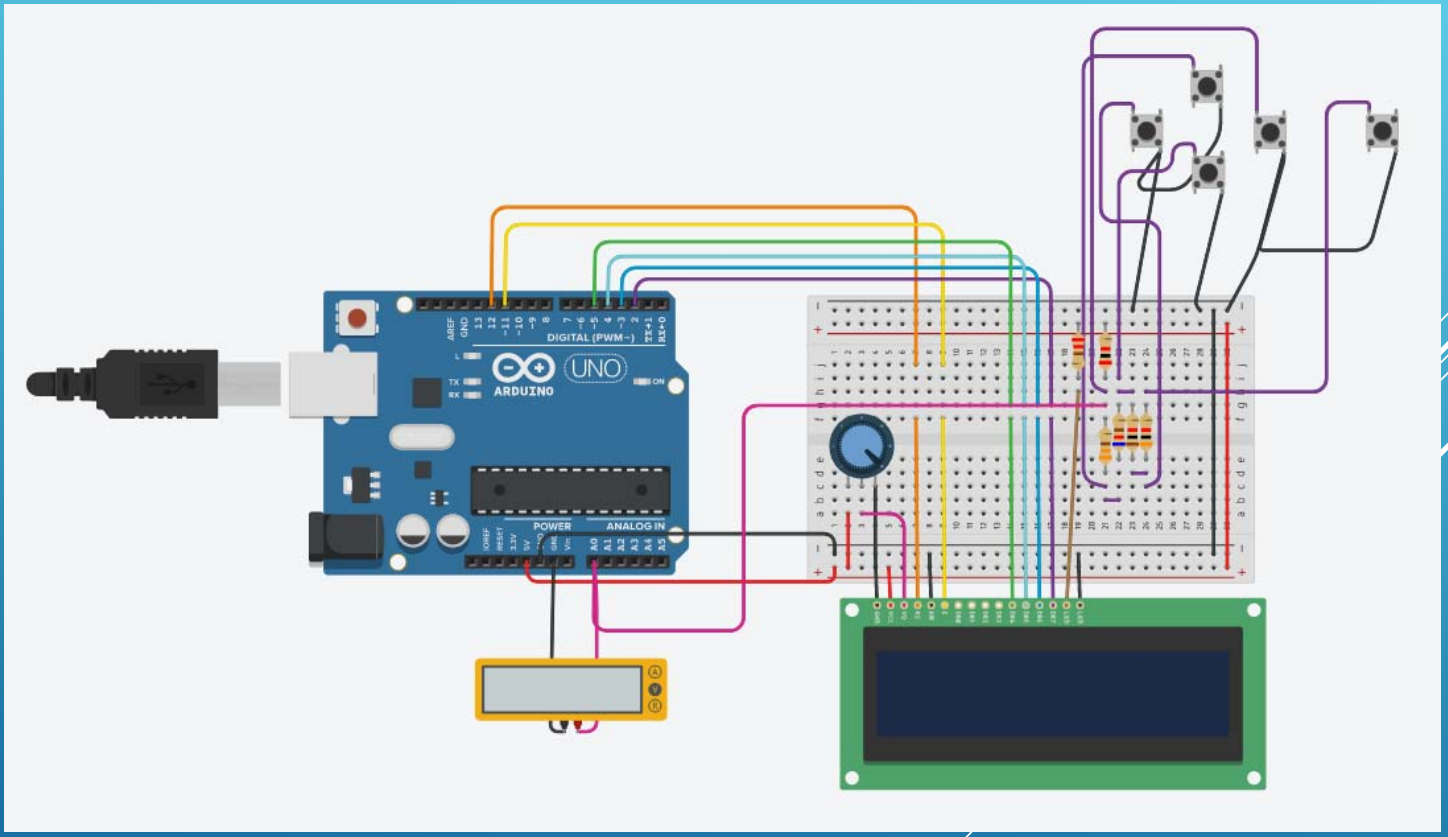
ESTRUTURA BÁSICA – ARDUINO UNO:

- Serial.Available() informa se tem algum caractere recebido na porta serial.
- SWITCH-CASE é uma estrutura de decisão para diferentes valores da variável observada (neste caso caracteres).
- DigitalWrite(pino, estado lógico) define o estado lógico de um pino de saída digital (LOW- 0V ou zero e HIGH – um ou 5V- alimentação positiva).

```
// ********Estrutura de decisão********
 if (Serial.available()) {
  char Caracter_Lido = Serial.read();
 switch (Caracter_Lido) {
   case '1':
    digitalWrite(Led1, HIGH);
    digitalWrite(Led2, LOW);
    digitalWrite(Led3, LOW);
    digitalWrite(Led4, LOW);
    break;
   case '2':
    digitalWrite(Led1, LOW);
    digitalWrite(Led2, HIGH);
    digitalWrite(Led3, LOW);
    digitalWrite(Led4, LOW);
    break;
   case '3':
    digitalWrite(Led1, LOW);
    digitalWrite(Led2, LOW);
    digitalWrite(Led3, HIGH);
    digitalWrite(Led4, LOW);
    break;
   case '4':
    digitalWrite(Led1, LOW);
    digitalWrite(Led2, LOW);
    digitalWrite(Led3, LOW);
    digitalWrite(Led4, HIGH);
    break;
   default:
    Serial.print("Valor ");
    Serial.print(Caracter_Lido);
    Serial.println(" incorreto.");
    digitalWrite(Led1, LOW);
    digitalWrite(Led2, LOW);
    digitalWrite(Led3, LOW);
    digitalWrite(Led4, LOW);
  }
 }
```

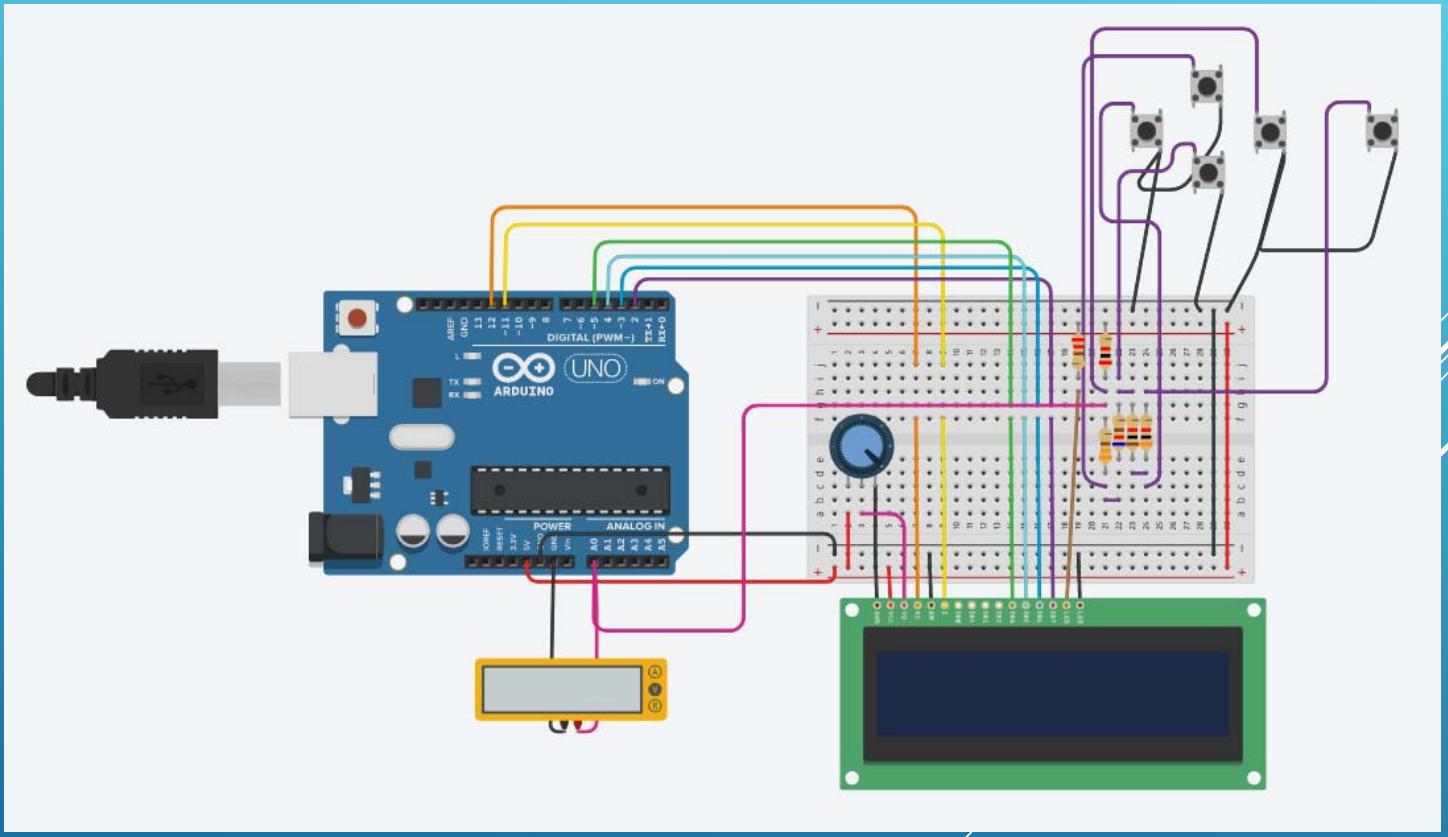
ESTRUTURA BÁSICA – ARDUINO UNO:

- DISPLAY – envio de caracteres por nibble (4 bits ao invés de 8)
- TECLADO – opção pela leitura analógica. Usa um pino e permite várias teclas, mas não se forem pressionadas simultâneamente...



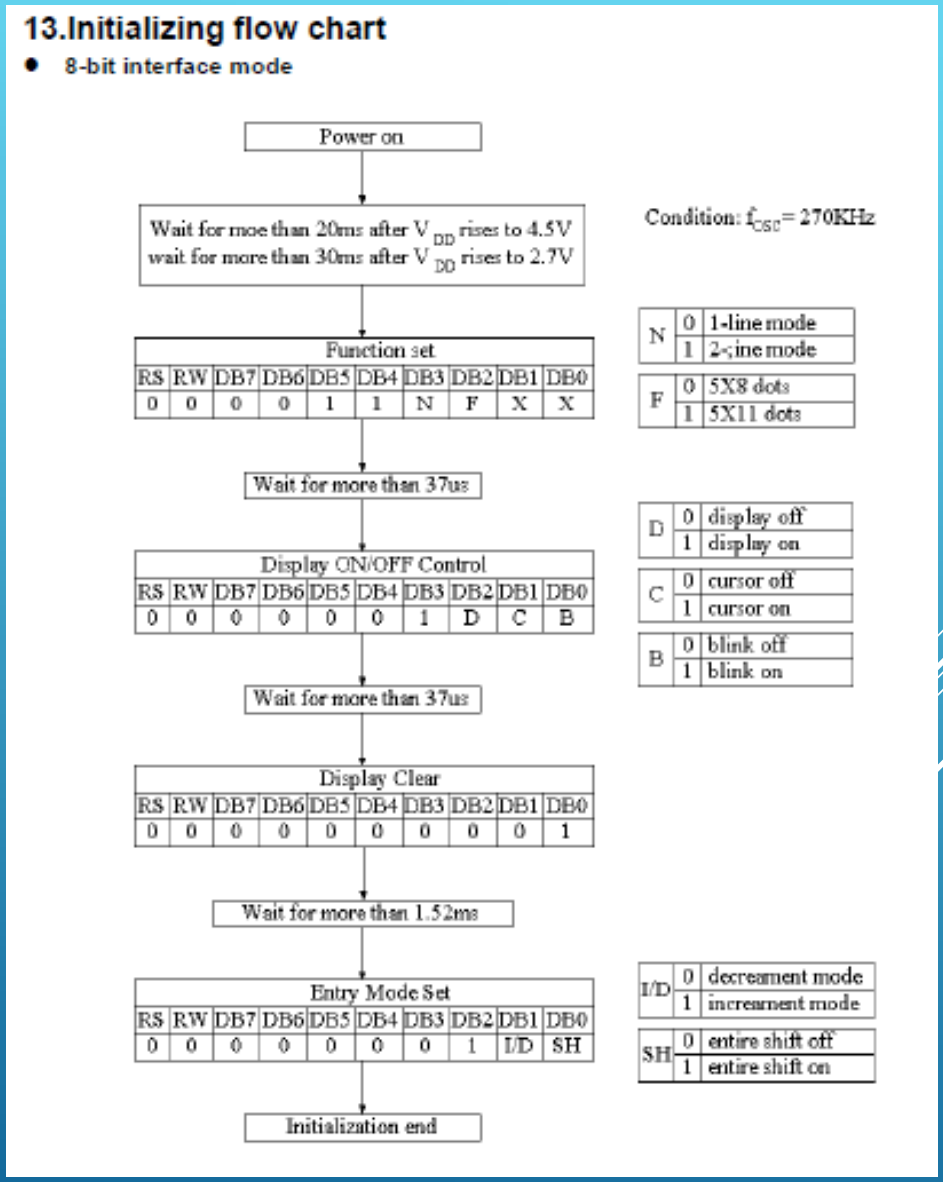
ESTRUTURA BÁSICA – ARDUINO UNO:

- DISPLAY – envio de caracteres por nibble (4 bits ao invés de 8)
- TECLADO – opção pela leitura analógica. Usa um pino e permite várias teclas, mas não se forem pressionadas simultâneamente...



ESTRUTURA BÁSICA – ARDUINO UNO:

- DISPLAY – inicialização
- Existem bibliotecas, mas observar diagrama de tempo e comandos definidos pelos fabricantes.



13.Initializing flow chart

- 8-bit interface mode

Power on

Wait for moe than 20ms after V_{DD} rises to 4.5V
wait for more than 30ms after V_{DD} rises to 2.7V

Condition: f_{OSC} = 270KHz

Function set

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	N	F	X	X

N	0	1-line mode
	1	2-;ine mode
F	0	5X8 dots
	1	5X11 dots

Wait for more than 37us

Display ON/OFF Control

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D	0	display off
	1	display on
C	0	cursor off
	1	cursor on
B	0	blink off
	1	blink on

Wait for more than 37us

Display Clear

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Wait for more than 1.52ms

Entry Mode Set

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

I/D	0	decreament mode
	1	increament mode
SH	0	entire shift off
	1	entire shift on

Initialization end

ESTRUTURA BÁSICA – ARDUINO UNO:

- DISPLAY – biblioteca
- LiquidCrystal lcd(8, 9, 4, 5, 6, 7); correspondem aos pinos (rs, enable, d0, d1, d2, d3) do display e agora ligados aos respectivos pinos do Arduino Uno.
- Observar biblioteca no GitHub… LiquidCrystal-master

28 src/LiquidCrystal.cpp

@@ -47,33 +47,26 @@

```
47  47  LiquidCrystal::LiquidCrystal(uint8_t rs,  uint8_t enable,
48  48                               uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3)
49  49  {
50  50    init(1, rs, 255, enable, d0, d1, d2, d3, 0, 0, 0, 0);
51  51  }
52  52
53  53  void LiquidCrystal::init(uint8_t fourbitmode, uint8_t rs, uint8_t rw, uint8_t enable,
54  54                           uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
55  55                           uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7)
56  56  {
57  57    _rs_pin = rs;
58  58    _rw_pin = rw;
59  59    _enable_pin = enable;
60  60
61  61    _data_pins[0] = d0;
62  62    _data_pins[1] = d1;
63  63    _data_pins[2] = d2;
64  64    _data_pins[3] = d3;
65  65    _data_pins[4] = d4;
66  66    _data_pins[5] = d5;
67  67    _data_pins[6] = d6;
68  68    _data_pins[7] = d7;
69  69
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- DISPLAY – criação de caracteres especiais

- Primeiro uma tabela de pontos (vetor) e
- Depois (no SETUP) apontar para o endereço de memória a ser utilizado.
- Por fim a utilização no código.

```
// Criando um caracter especial via tabela de pontos
byte CaracEspecial[8] = {
  B00000,
  B10001,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000,
};
```

Character Code (DDRAM Data)									CGRAM Address						Character Patterns (CGRAM Data)							
b8	b7	b6	b5	b4	b3	b2	b1	b0	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-	-	1	1	1	1	1
						0	0	0				0	0	1				0	0	1	0	0
						0	0	0				0	1	0				0	0	1	0	0
						0	0	0				0	1	1				0	0	1	0	0
						0	0	0				1	0	0				0	0	1	0	0
						0	0	0				1	0	1				0	0	1	0	0
						0	0	0				1	1	0				0	0	1	0	0
						0	0	0				1	1	1				0	0	0	0	0

```
lcd.createChar(0, CaracEspecial);
```

```
// apresenta o caracter especial por 5 segundos
 lcd.write(byte(0));
 delay(5000);
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- SETUP:
- Carrega caracteres especiais
- Posiciona cursor
- Escreve textos e caractere especial.

```
void setup() {
  // grava na memória do display o caracter especial na posição 0
  lcd.createChar(0, CaracEspecial);

  lcd.createChar(1, cima);
  lcd.createChar(2, baixo);
  lcd.createChar(3, esquerda);
  lcd.createChar(4, direita);

  // configurar número de colunas (16) e linhas (2)
  lcd.begin(16, 2);
  // Como a inicialização posiciona o cursor na primeira linha
  // e coluna, o comando abaixo apresenta o texto neste local.
  lcd.print("hello, world!");
  delay(1000);
  // define o cursor na primeira coluna (0) e segunda linha (1)
  lcd.setCursor(15,0);
  // apresenta o caracter especial por 5 segundos
  lcd.write(byte(0));
  delay(5000);

 // define o cursor na primeira coluna (0) e primeira linha (0)
 lcd.setCursor(0,0);
 // apresenta texto
 lcd.print("Arduino e Cia");
 // define o cursor na primeira coluna (0) e segunda linha (1)
 lcd.setCursor(0,1);
 // apresenta texto nesta posição
 lcd.print("Tecla :");
}
```

ESTRUTURA BÁSICA – ARDUINO UNO:

- LOOP infinito:
- Lê a tecla apertada através da tensão analógica gerada.
- A partir da tecla apertada apresenta o texto referente à tecla e apresenta o caractere especial.
- Observar que para apagar alguma indicação grande foram usados espaços após nomes pequenos. (ao invés de usar LCD.CLEAR())

```
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  //lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  //lcd.print(millis() / 1000);

 int botao;
 botao = analogRead (0);  //Leitura do valor da porta analógica A0
 lcd.setCursor(8,1);
 if (botao < 100) {
  lcd.print ("Direita ");
  lcd.setCursor(15,0);
  lcd.write(byte(4));
 }
 else if (botao < 200) {
  lcd.print ("Cima    ");
  lcd.setCursor(15,0);
  lcd.write(byte(1));
 }
 else if (botao < 400){
  lcd.print ("Baixo   ");
  lcd.setCursor(15,0);
  lcd.write(byte(2));
 }
 else if (botao < 600){
  lcd.print ("Esquerda");
  lcd.setCursor(15,0);
  lcd.write(byte(3));
 }
 else if (botao < 800){
  lcd.print ("Selecao ");
 }

}
```

ESTRUTURA BÁSICA – ARDUINO UNO:

https://www.tinkercad.com/things/j8KQZmjV1nP-display-e-teclado-com-ad