

INSTITUTO FEDERAL

Santa Catarina
Câmpus Tubarão

Introdução a Linguagens de Programação

Curso Superior de Tecnologia em Sistemas Embarcados

Professor: Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

Março de 2023

Cronograma

| Encontro | Data | Nº Aulas | Conteúdo |
|----------|---------|----------|---|
| 1 | 29-fev. | 04 | Recepção e Apresentação do Unidade / Apresentação do Plano de Ensino / Avaliação Diagnóstica / Introdução a sistemas embarcados |
| 2 | 02-mar. | 04 | Conceitos e Características e Aplicações de Sistemas Embarcados / Histórico de Sistemas Embarcados / Práticas com Arduino |
| 3 | 07-mar. | 04 | Microcontroladores, Microprocessadores / Periféricos / Introdução ao Arduino / Introdução ao C |
| 4 | 14-mar. | 04 | Introdução à Linguagens de Programação |
| 5 | 21-mar. | 04 | Linguagem de Programação C |
| 6 | 28-mar. | 04 | Linguagens de Programação C para arduino |
| 7 | 04-abr. | 04 | Variáveis e Operadores |
| 8 | 11-abr. | 04 | Estruturas Condicionais |
| 9 | 18-abr. | 04 | Estruturas de Repetição |
| 10 | 25-abr. | 04 | Avaliação 01 |

Cronograma

| Encontro | Data | Nº Aulas | Conteúdo |
|----------|---------|----------|---|
| 11 | 02-mai. | 04 | Microcontroladores |
| 12 | 09-mai. | 04 | Entradas e Saídas Digitais |
| 13 | 16-mai. | 04 | Conversor Analógico-Digital |
| 14 | 18-mai. | 04 | Sensores |
| 15 | 23-mai. | 04 | Comunicação Serial |
| 16 | 06-jun. | 04 | PWM |
| 17 | 13-jun. | 04 | Temporizadores |
| 18 | 20-jun. | 04 | Interrupções |
| 19 | 27-jun. | 04 | Avaliação 02 |
| 20 | 04-jul. | 04 | Conselho de Classe / Atividades de Encerramento da UC |
| | | 80 | |

Pauta

- Microcontroladores e Microprocessadores;
- Periféricos;
- Arduino;
- Introdução ao C;
- Variáveis;
- Operadores;

Microcontroladores e Microprocessadores

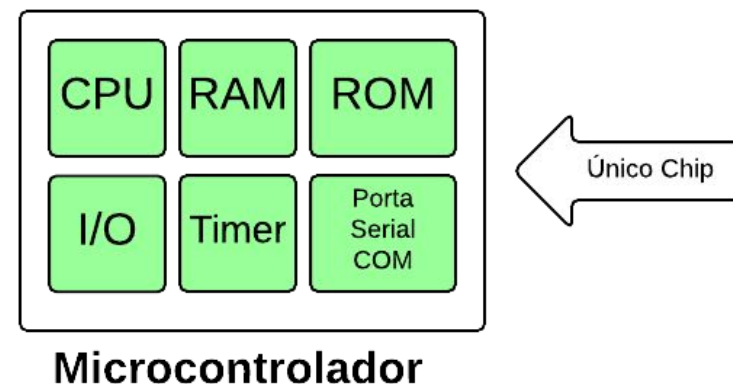
Microcontroladores e Microprocessadores

O que são microcontroladores e microprocessadores?

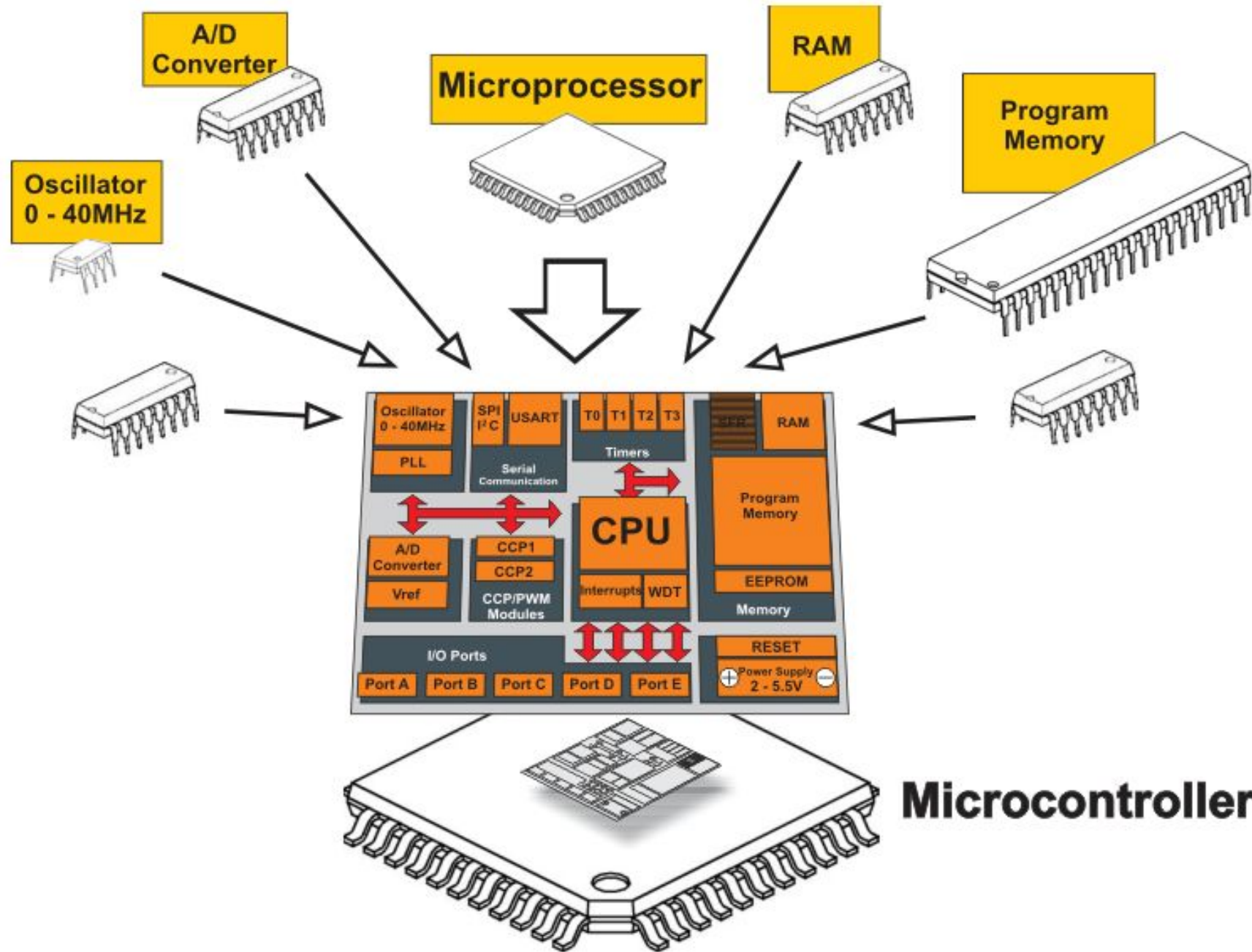


Microcontroladores e Microprocessadores

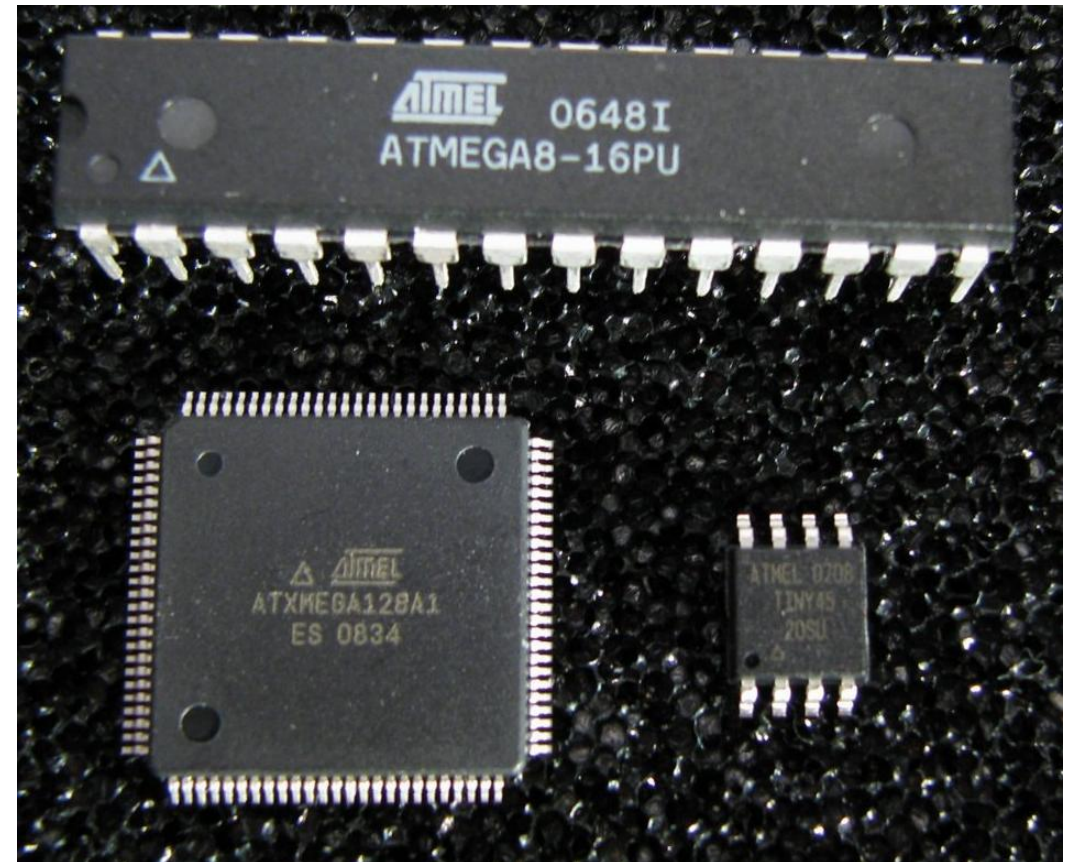
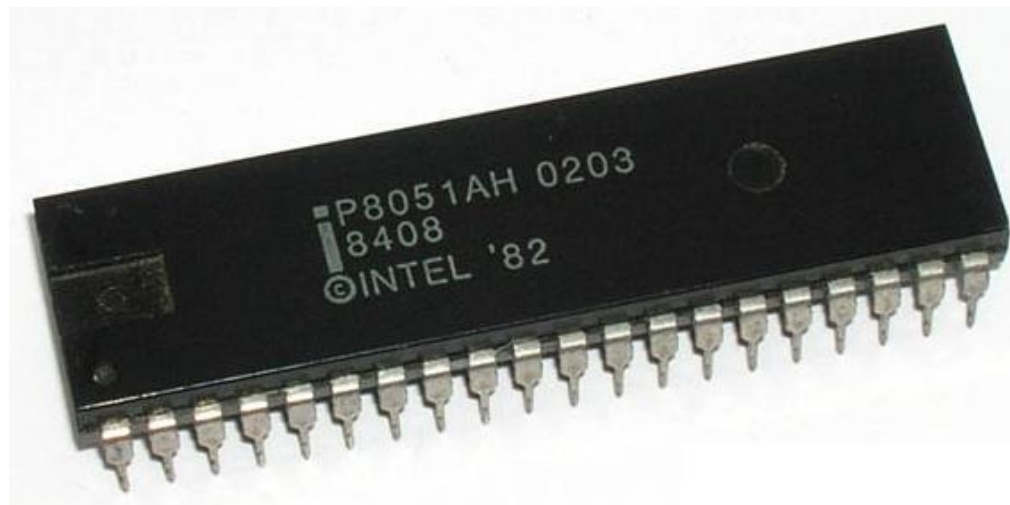
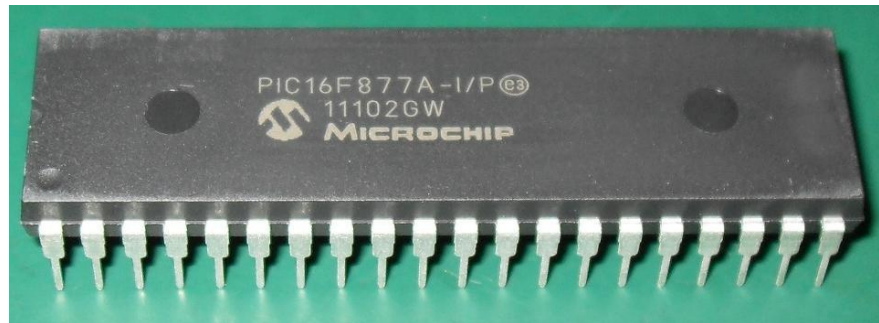
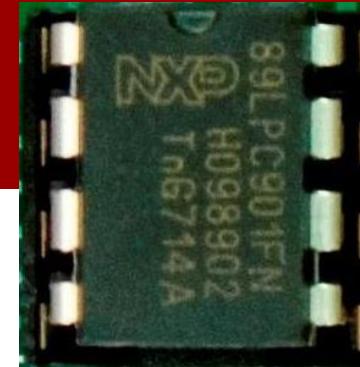
- ❑ Computador em um único chip;
- ❑ Custo reduzido;
- ❑ Recursos integrados:
 - ❑ RAM;
 - ❑ ROM;
 - ❑ Conversor A/D e D/A;
 - ❑ Portas I/O;
- ❑ Timers e Interrupções;
- ❑ Podem ser programados com código Assembly ou em linguagens de alto nível;



Microc



Microcontroladores



Microprocessadores

- ❑ Circuito integrado que realiza as funções de cálculo e tomada de decisão de um computador;
- ❑ Controlado por um programa armazenado na memória, executando diferentes operações;
- ❑ Memórias RAM e ROM portas de I/O e Timers externos;
- ❑ Computador de propósito geral;
- ❑ Comunicação por barramentos.

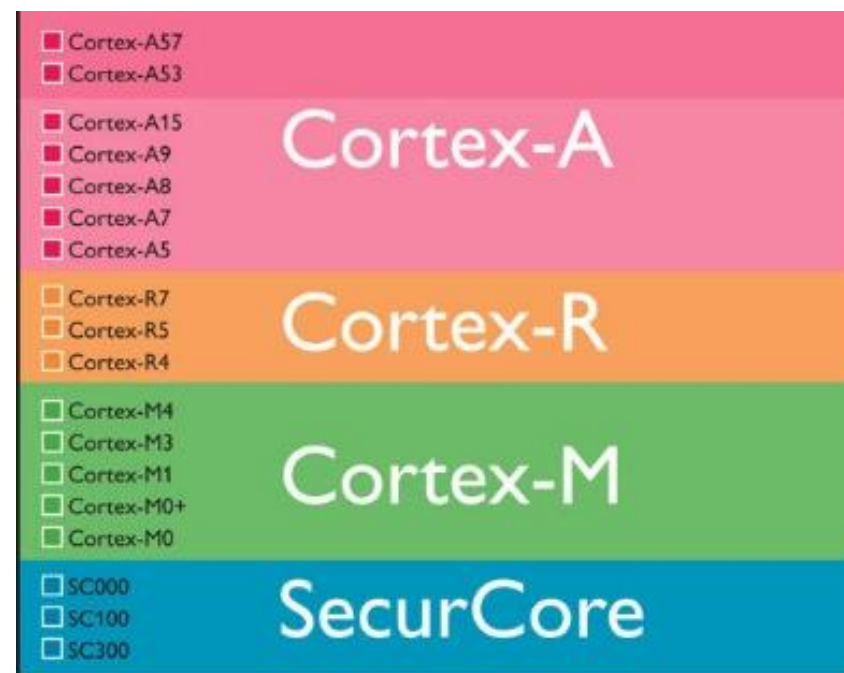
Microprocessadores - Características

- ❑ **Memória de programa:** Armazena as instruções que um microprocessador deve executar (ROM);
- ❑ **Memória de dados:** Área de leitura e escrita de dados temporários sempre que necessário (RAM);
- ❑ **ULA:** Unidade Lógica Aritmética, responsável pelos cálculos e a lógica matemática para tomada de decisão;

Microprocessadores - Características






- ❑ **CPU:** Unidade Central de Processamento, responsável pelo processamento de dados da unidade, ela quem interpreta os comandos e ativa os dispositivos de entrada e saída;
- ❑ **Periféricos:** Circuitos que realizam funções específicas auxiliando a CPU no controle e interface com dispositivos externos.

Microprocessadores








Microcontroladores Vs Microprocessadores

Microcontrolador:

-  CPU, RAM, ROM, I/O e Timer encapsulados em um único chip;
-  Tamanho fixo no chip de RAM, ROM e portas I/O;
-  Aplicada a soluções onde custo, consumo de energia e espaço são críticas;
-  Propósito específico;
-  Ex: 8051, PIC, Atmel, Motorola, etc.

Microprocessador:

-  CPU independente, RAM, ROM, I/O e Timer separados;
-  Desenvolvedor pode definir a quantidade de RAM, ROM e portas I/O;
-  Custo mais elevado;
-  Propósito geral;
-  Ex: 8085, 8086, Cortex-M4 Cortex-A5, etc.

Periféricos

Periféricos - Características

- ❑ Responsáveis pelo sensoriamento e atuação das aplicações;
- ❑ Diferentes características devem ser consideradas na sua especificação (consumo de energia, protocolo de comunicação, taxa de amostragem);
- ❑ Grande variedade existente hoje no mercado;
- ❑ Especificação diretamente dependente do escopo e orçamento do projeto;
- ❑ Necessário um profundo conhecimento do escopo da aplicação para sua definição.

Periféricos - Exemplos de Uso

- ❑ Avaliar se há pessoas em um ambiente;
- ❑ Monitorar nível de luminosidade;
- ❑ Monitorar nível de umidade do solo;
- ❑ Controlar quantidade de água dispersa em determinado local;
- ❑ Medir distância de um objeto;
- ❑ Seguir determinada trajetória;

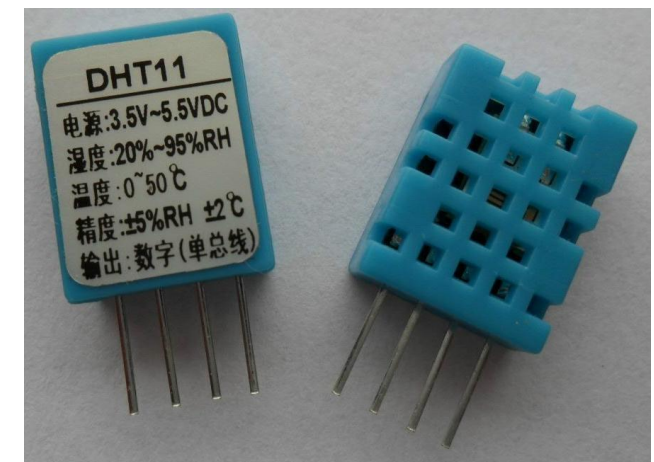
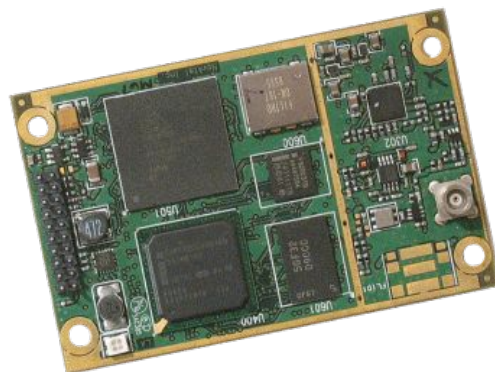
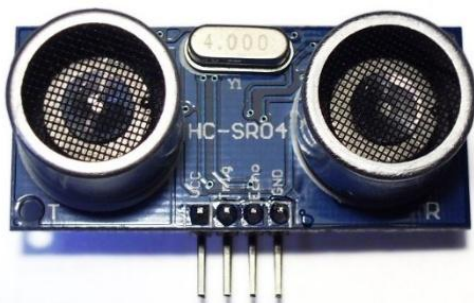
Periféricos - Cuidados ao Especificar

- ☐ Qual a taxa de amostragem que eu preciso?
- ☐ Qual o protocolo de comunicação? Ele é compatível com a placa que estou utilizando?
- ☐ O custo deste sensor está dentro do orçamento do projeto?
- ☐ Qual o seu consumo de energia?

Periféricos - Cuidados ao Manusear

- ❑ Verificar se as ligações estão corretas antes de ligar;
- ❑ Verificar se os polos de alimentação estão corretamente ligados;
- ❑ Verificar se a tensão de alimentação e de comunicação estão corretas;
- ❑ Verificar compatibilidade com a plataforma embarcada;
- ❑ Verificar necessidade de algum componente adicional no circuito de ligação (resistor, capacitor).

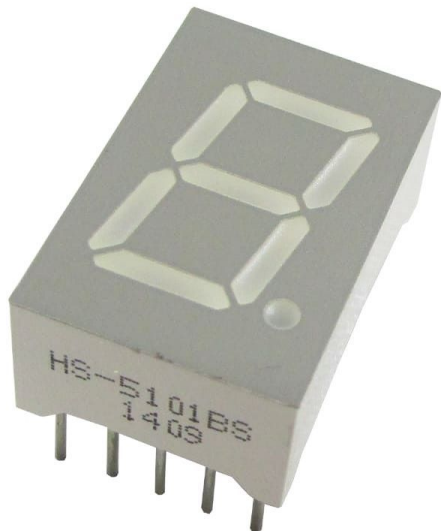
Sensores



Sensores



Displays

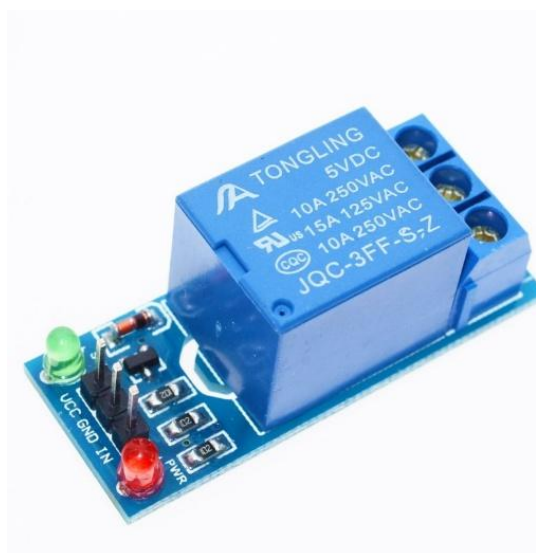
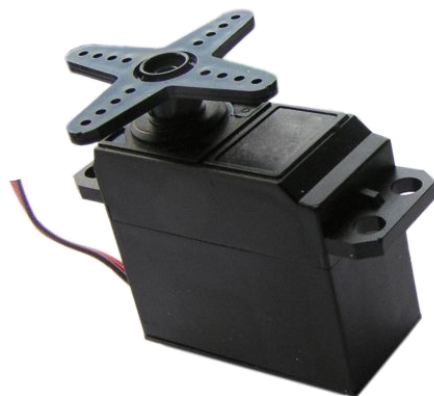


7 inch Display with SSD1963 Controller Board

Arduino Shield

Arduino MEGA2560

Atuadores



Onde comprar material?

☐ No Brasil

- ☐ Filipe Flop (www.filipeflop.com);
- ☐ Proesi (www.proesi.com.br);
- ☐ Robocore (www.robocore.net);

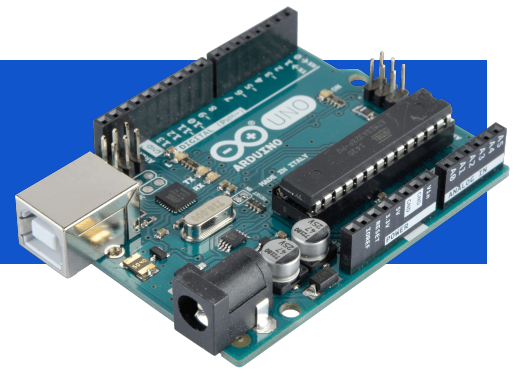
☐ No exterior

- ☐ Ali Express (pt.aliexpress.com/);
- ☐ Deal Extreme (www.dx.com/pt/);
- ☐ Ebay (www.ebay.com);

Arduino

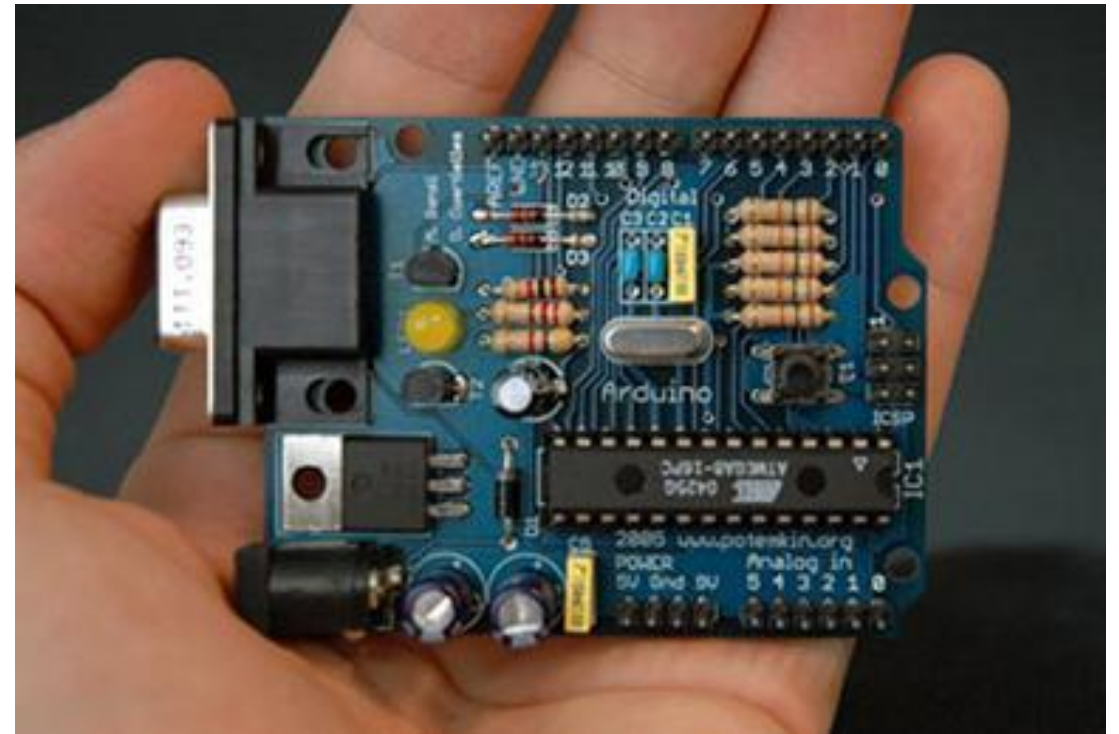
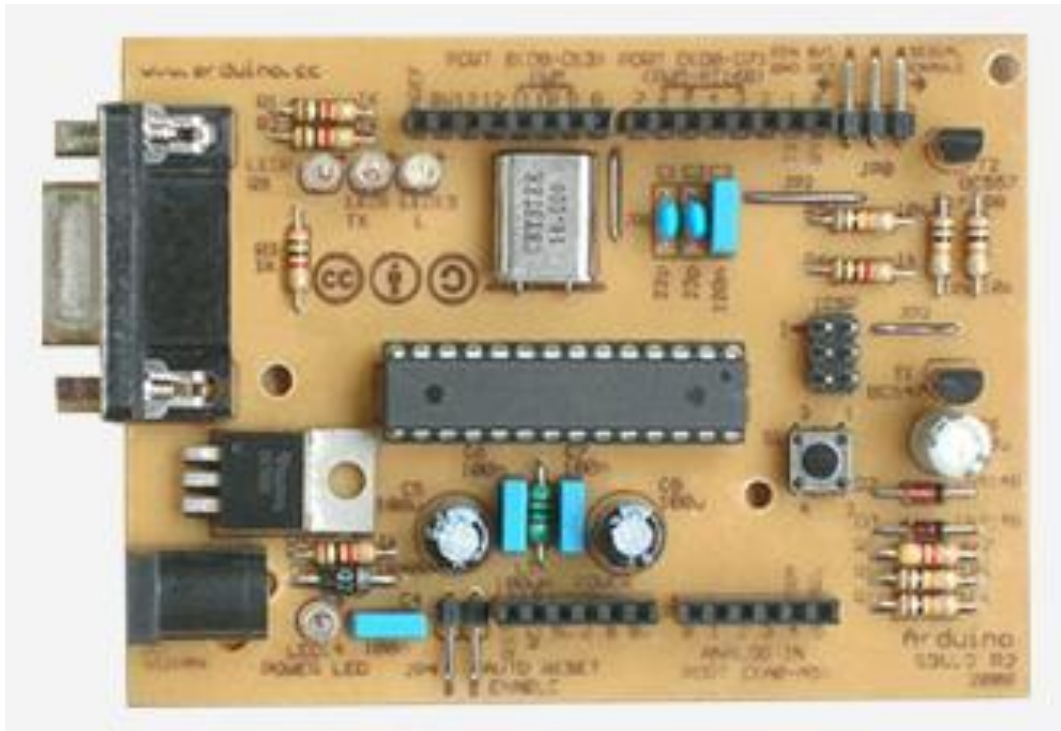
Arduino

Arduino

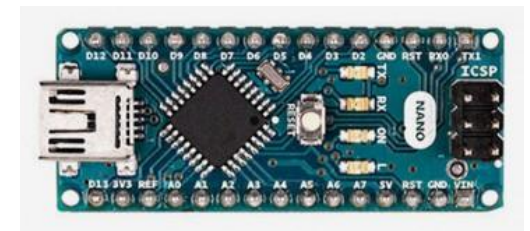
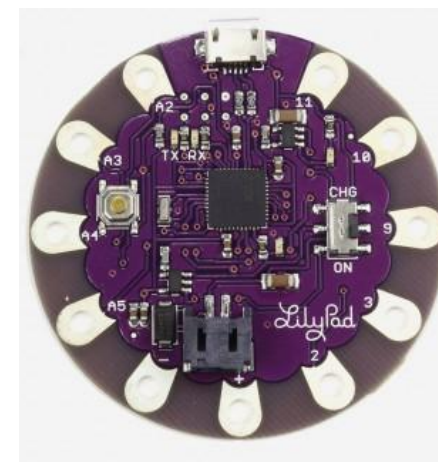
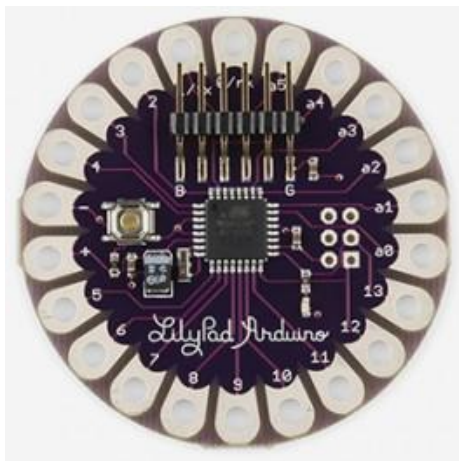
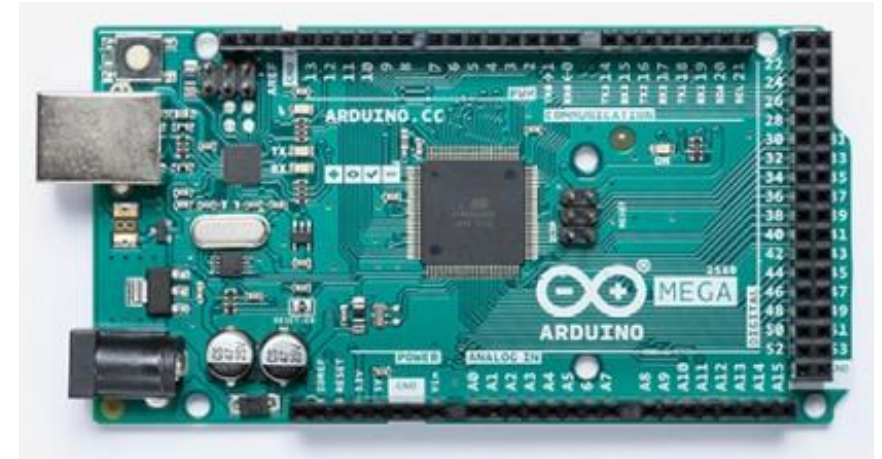


- ❑ Plataforma Open-Source de protótipos eletrônicos baseados em hardware e software flexível e fácil de usar;
- ❑ Destinado a qualquer pessoa interessada em criar objetos ou ambientes interativos;
- ❑ Projeto teve início em 2005 na cidade de Ivrea, Itália;
- ❑ Placa baseada em microcontrolador Atmel AVR e um ambiente de desenvolvimento baseado em C++;

Arduino - Primeiras versões



Arduino - Modelos existentes



Características do Arduino UNO

| | |
|-------------------------------|--------------------------------|
| Microcontrolador: | ATmega328 |
| Tensão de Operação: | 5V |
| Tensão de Entrada: | 7 – 12v |
| Pinos Digitais: | 14 (com 6 pinos com saída PWM) |
| Pinos Analógicos: | 6 |
| Corrente por pino I/O: | 40mA |
| Memória Flash: | 32KB |
| SRAM: | 2KB |
| Clock Speed: | 16MHz |

Portas

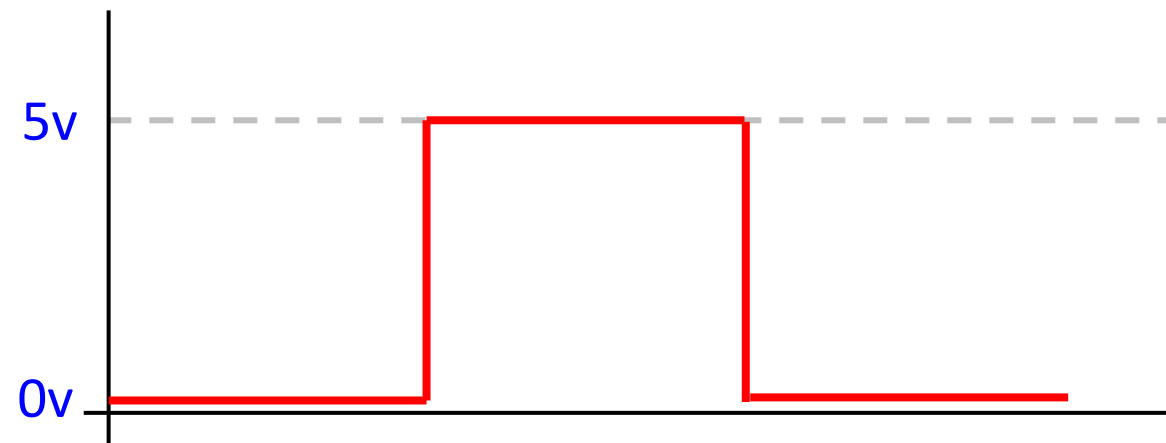
Portas

Tipos de Portas

- ❑ Portas Digitais;
- ❑ Portas Analógicas;
- ❑ Portas PWM;

Portas Digitais

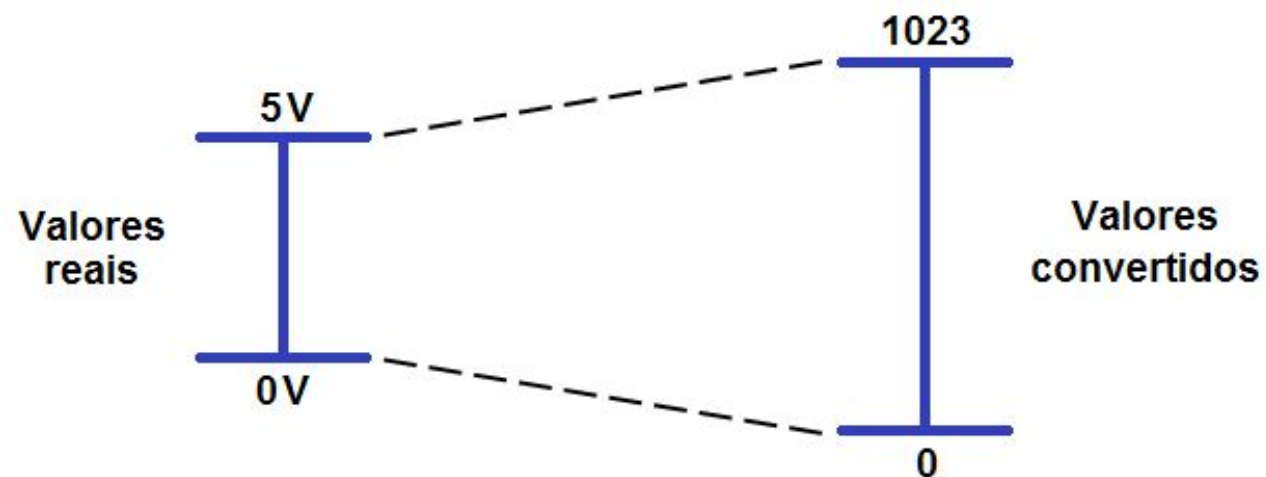
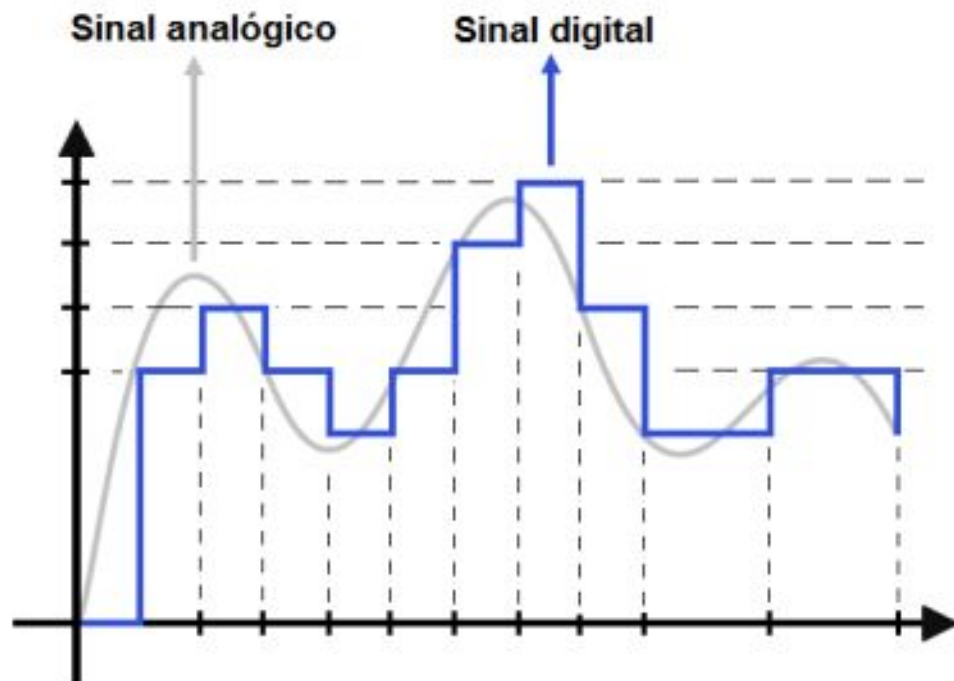
- ❑ 14 Portas Digitais;
- ❑ 0 / 5v
 - ❑ LOW e HIGH;



Portas Analógicas

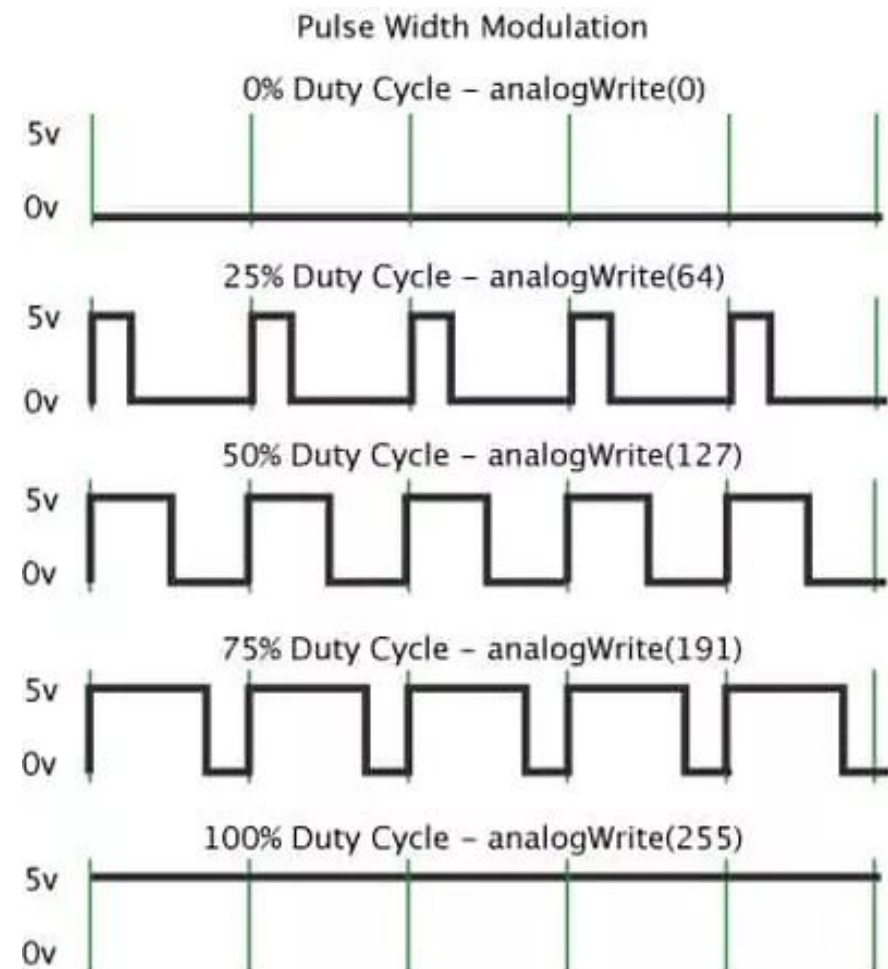
6 Portas Analógicas

0 ~ 5V.



Portas PWM (~)

- ❑ Onda quadrada onde se controla a porcentagem do tempo em que ela permanece em nível lógico alto (Duty Cycle).
- ❑ Alteração no Duty Cycle provoca mudança no valor médio da onda, variando entre 0v e 5v.
- ❑ Valor do Duty Cycle armazenado em um registrador de 8 bits variando entre 0 e 255.



Shields

Shields

Shields

- ❑ Shields são placas que podem ser conectados sobre o Arduino estendendo as suas funcionalidades;
- ❑ Shields seguem a mesma filosofia do Arduino:
 - ❑ Open;
 - ❑ Fácil de Montar;
 - ❑ “Barato”

Sh



Software de Desenvolvimento

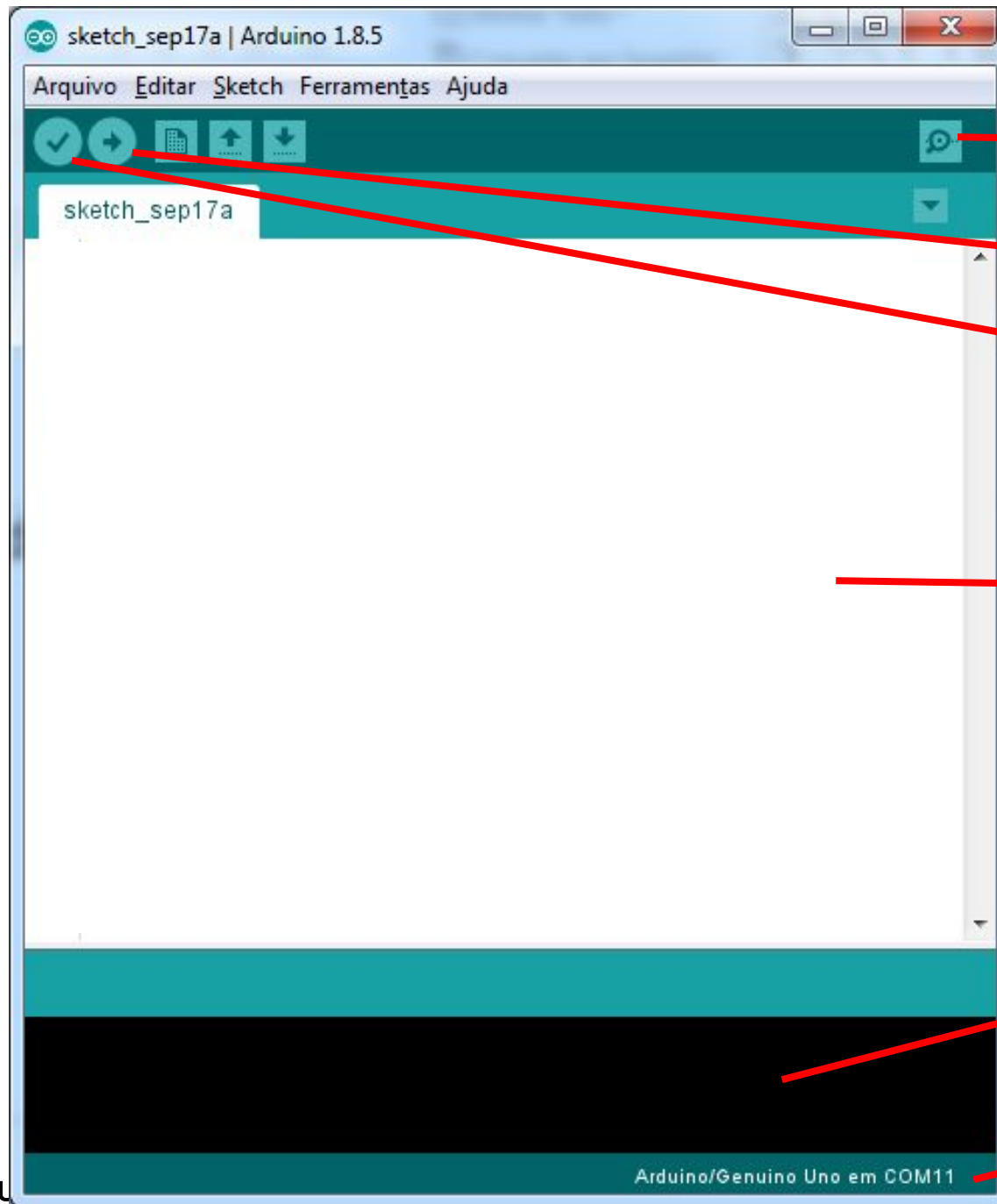
Software de Desenvolvimento

- ❑ Open-Source;
- ❑ Windows, Mac e Linux;
- ❑ Disponível em:
<http://arduino.cc>



Características da IDE

- ❑ Facilita o desenvolvimento de software;
- ❑ Muitos exemplos para as bibliotecas padrão;
- ❑ Permite a gravação do código no microcontrolador;
- ❑ Monitor serial para troca de mensagens;



Monitor Serial

Gravação do Código

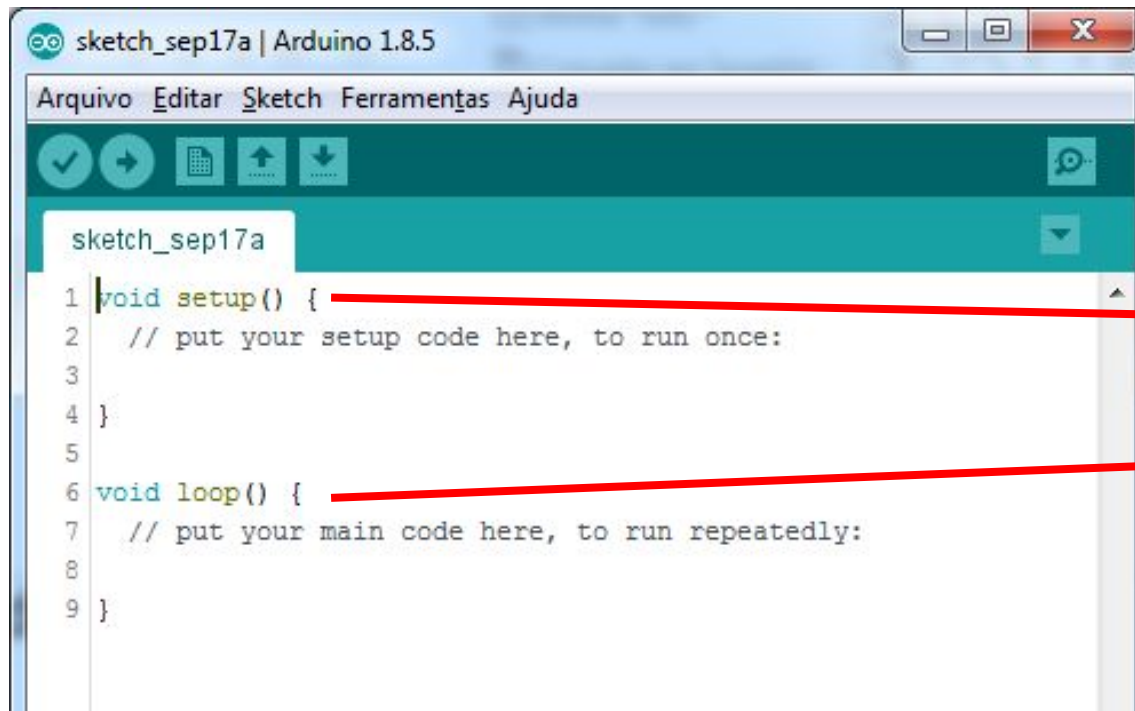
Compilar Código

Área de Programação

Informação

Comunicação

Desenvolvimento para Arduino



```
sketch_sep17a | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda

sketch_sep17a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

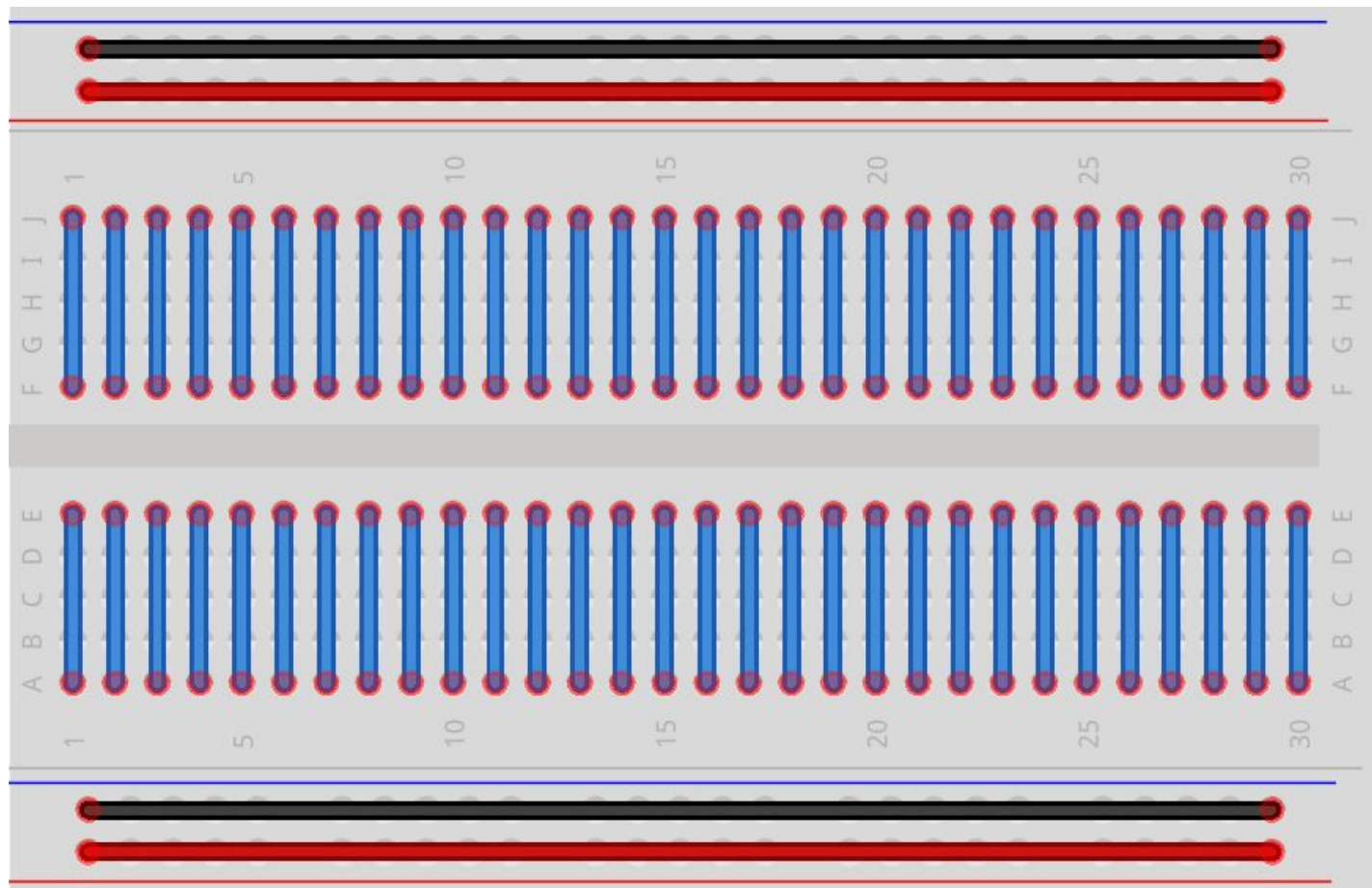
Configuração inicial

Loop de execução

Primeiros Passos

Primeiros Passos

Protoboard e Jumpers



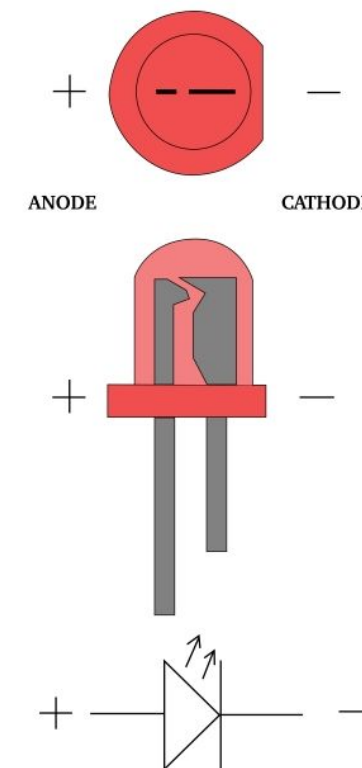
Led

❑ Led Difuso

- ❑ Tensão 2V.
- ❑ Corrente 20 mA ~ 0,02 A

$$R = \frac{(V_{fonte} - V_{led})}{I_{led}}$$

- ❑ **R** = Resistência;
- ❑ **V_{fonte}** = Tensão da fonte;
- ❑ **V_{led}** = Tensão do Led;
- ❑ **I_{led}** = Corrente do Led.
- ❑ <http://blog.novaeletronica.com.br/calculadora-online-resistor-limitador-led/>



Resistor

4- Faixas

1.0 K Ω \pm 5%

1st 2nd 3rd 4th

| Cor | 1ª Faixa | 2ª Faixa | 3ª Faixa | Multiplicador Decimal | | Tolerância |
|----------|----------|----------|----------|-----------------------|---------------|------------|
| Preto | 0 | 0 | 0 | 1 | 1 | |
| Marrom | 1 | 1 | 1 | 10 | 10 | \pm 1% |
| Vermelho | 2 | 2 | 2 | 100 | 100 | \pm 2% |
| Laranja | 3 | 3 | 3 | 1K | 1.000 | |
| Amarelo | 4 | 4 | 4 | 10K | 10.000 | |
| Verde | 5 | 5 | 5 | 100K | 100.000 | |
| Azul | 6 | 6 | 6 | 1M | 1.000.000 | |
| Violeta | 7 | 7 | 7 | 10M | 10.000.000 | |
| Cinza | 8 | 8 | 8 | | 100.000.000 | |
| Branco | 9 | 9 | 9 | | 1.000.000.000 | |
| Ouro | | | | | 0.1 | \pm 5% |
| Prata | | | | | 0.01 | \pm 10% |
| Branco | | | | | | \pm 20% |

1st 2nd 3rd 4th 5th

254 Ω \pm 1 %

5- Faixas

Resistor de 4 faixas

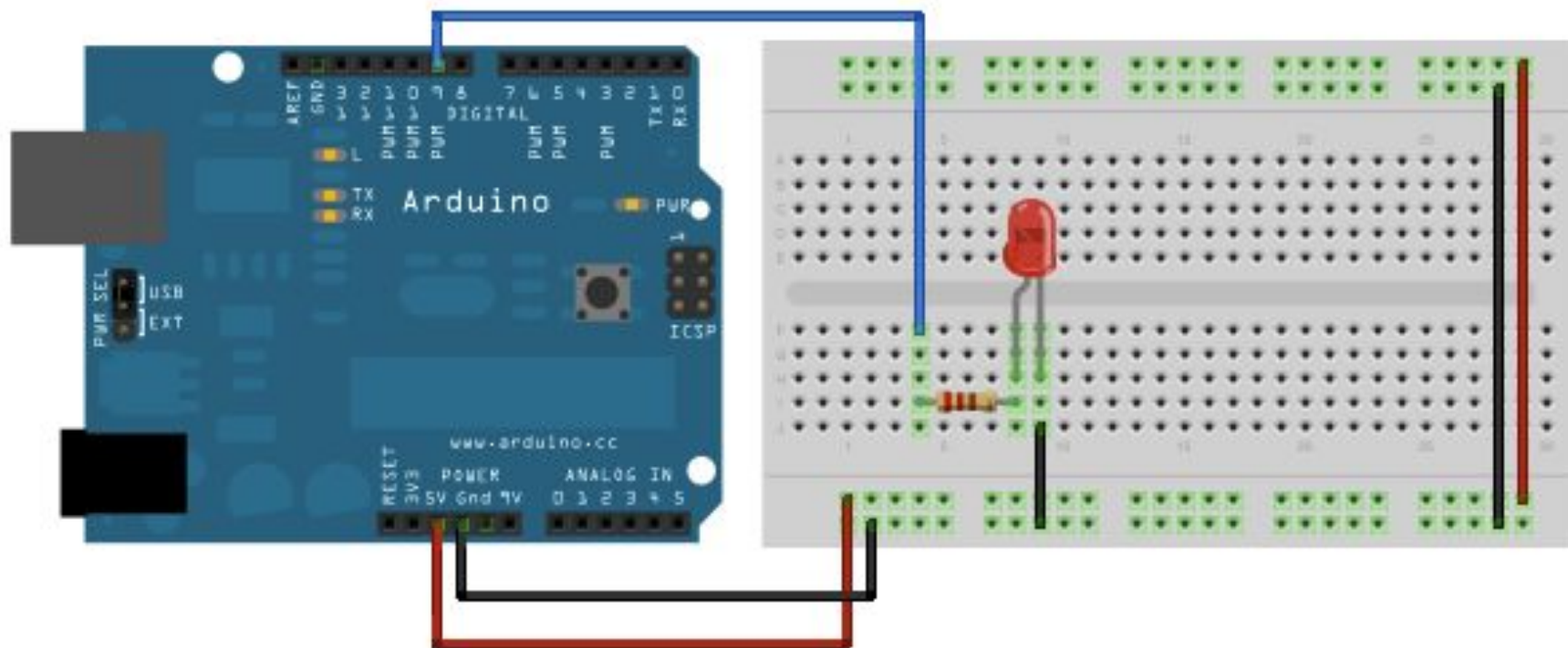
- Faixa 1: Valor
- Faixa 2: Valor
- Faixa 3: Fator Multiplicador
- Faixa 4: Tolerância

Resistor de 5 faixas

- Faixa 1: Valor
- Faixa 2: Valor
- Faixa 3: Valor
- Faixa 4: Fator Multiplicador
- Faixa 5: Tolerância

http://www.novaeletronica.com.br/ferramentas_online/cores-de-resistor-online.php

Pratica 1: Acionando um Led



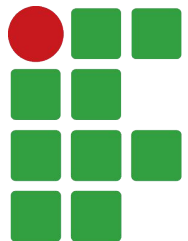
Prática 1: Acionando um Led

```
#define led1 9
Void setup(){
    pinMode(led1, OUTPUT);
}
Void loop(){
    digitalWrite(led1, HIGH);
    delay(1000);
    digitalWrite(led1, LOW);
    delay(1000);
}
```

Prática 2: Semáforo

- ❑ Com base no exemplo anterior, monte um circuito que demonstre o funcionamento de um semáforo de trânsito. Esse deve obedecer os seguintes critérios:
 - ❑ A luz verde deve ficar acesa por 1,5 segundos;
 - ❑ A luz amarela deve ficar acesa por 1 segundo;
 - ❑ A luz vermelha deve ficar acesa por 3 segundos.
- ❑ Os leds devem ser acesos na sequência correta de funcionamento do semáforo de trânsito;
- ❑ Não deve haver mais de um led aceso ao mesmo tempo.





INSTITUTO FEDERAL

Santa Catarina
Câmpus Tubarão

Obrigado!

Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

se.cst.tub@ifsc.edu.br