# 

**Curso:** Ciência de Computação  **Turno:** Diurno

**Disciplina:** Microcontroladores **Professor:** Átila Carvalho Junior

**Atividade:** Teste Teórico 1  **Data:** 15/09/2021

**Aluno:** João Vítor Fernandes Dias **Matrícula:** 00119110377

## **Descreva as principais diferenças entre um microcontrolador e um microprocessador. (2,0)**

O microprocessador geralmente tem maior capacidade de processamento, assim sendo capaz de realizar cálculos mais complexos. Ele é o cérebro que opera a maioria dos dispositivos eletrônicos atuais, porém depende de outros componentes para que funcione.

Já o microcontrolador apresenta diversos componentes internos. Além de um microprocessador, tem também memórias RAM e ROM, assim como outros. A junção de todos eles componentes torna o microcontrolador um componente bem completo e capaz de gerenciar diversos dispositivos que estiverem agregados.

## **Identifique e explique brevemente cada parte do Arduino Uno abaixo: (2,0)**

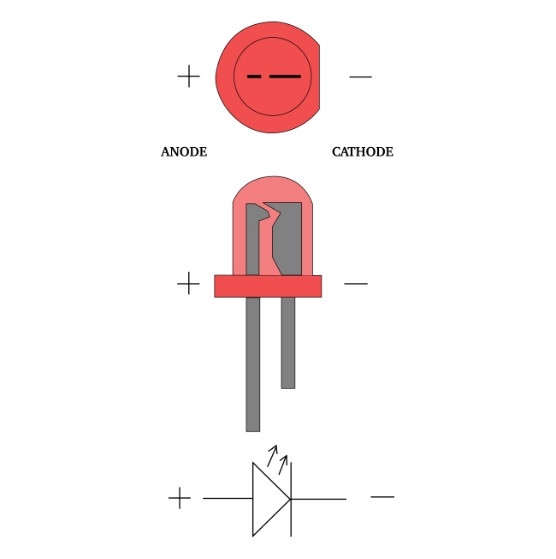


1. **Microprocessador**: É a parte que de fato processa e controla os valores de entrada e saída do microcontrolador.
2. **Porta de entrada USB**: É por onde é plugado o cabo USB para poder fazer o carregamento do código do computador para o armazenamento do Arduino. Também serve como entrada de energia via USB.
3. **Entradas analógicas**: Por onde são recebidos valores de entrada analógicos. Geralmente podendo variar entre 0 e 1023.
4. **Saídas de energia**: Contém diversos locais que provém energia e também alguns grounds.
5. **Botão de reset**: Pode ser apertado para reiniciar a programação configurada.
6. **Chip de controle USB**: É o chip que gerencia as informações recebidas via USB.
7. **Fonte de alimentação externa**: Caso seja necessário utilizar uma bateria extra para o projeto, sua conexão é feita por esta parte. Geralmente utilizado quando a entrada USB não disponibiliza energia o bastante ou quando há necessidade de desconectar o Arduino da conexão USB e torná-lo mais independente.
8. **LED indicador de atividade**: LED que se ilumina quando o Arduino está ativo.
9. **LED conectado ao pino 13**: LED conectado ao pino 13; nos programas exemplo disponibilizados pelo Arduino IDE, ele é referido no código pelo nome “LED\_BUILTIN”.

## **Cite e explique as principais funções existentes em um programa desenvolvido para Arduino. (2,0)**

1. Void setup () {}: É utilizada no início do código. O que estiver entre as chaves será executado apenas uma única vez. Geralmente utilizado para inicializar algumas configurações que serão utilizadas no decorrer do programa.
2. Void loop () {}: Ocorre após a função setup (). Ela tende a se repetir infinitamente, tal qual uma estrutura “while (true) {}”. Geralmente é nela que está escrita a maior parte do código.

## **Explique como é feita a correta polarização de um diodo (LED) para que haja condução de corrente elétrica num circuito. Desenhe um circuito para exemplificar. (1,5)**



O LED, sendo um diodo, apenas permite a passagem de corrente de um lado para o outro. Ele recebe energia por seu polo positivo.

Polo positivo: Por onde “entra” a corrente; a “perna” metálica é mais alongada; A parte plástica do LED é mais arredondada no lado do polo positivo; geralmente é representada com o símbolo de mais “+”; geralmente é representada pela cor vermelha; também chamada de ânodo; deve ser conectado a fonte de maior tensão.

Polo Negativo: Por onde “sai” a corrente; a “perna” metálica é mais curta; A parte plástica do LED é mais achatada (reta) no lado do polo negativo; geralmente é representada com o símbolo de menos “-”; geralmente é representada pela cor preta; também chamada de cátodo; deve ser conectado a fonte de menor tensão.

## **Explique de forma geral o código abaixo, identificando o objetivo dele e comentando todas as linhas do mesmo. (2,5)**

### **Código**

void setup () // é chamada a função setup

{ // inicia o bloco de códigos da função setup

pinMode (11, OUTPUT); // é definido como saída o pino 11

pinMode (4, OUTPUT); // é definido como saída o pino 4

pinMode (6, OUTPUT); // é definido como saída o pino 6

} // termina o bloco de códigos da função setup

void loop () // é chamada a função loop

{ // inicia o bloco de códigos da função loop

digitalWrite (11, HIGH); // aumenta a tensão do pino 11

delay (4000); // pausa o código por 4000 milissegundos

digitalWrite (11, LOW); // reduz a tensão do pino 11

digitalWrite (4, HIGH); // aumenta a tensão do pino 4

delay (4000); // pausa o código por 4000 milissegundos

digitalWrite (4, LOW); // reduz a tensão do pino 4

digitalWrite (6, HIGH); // aumenta a tensão do pino 6

delay (2000); // pausa o código por 2000 milissegundos

digitalWrite (6, LOW); // reduz a tensão do pino 6

} // termina o bloco de códigos da função loop

### **Explicação**

De uma forma geral: O código alterna intercaladamente os valores de tensão de 3 pinos diferentes, mantendo-os com tensão elevada por um certo período de tempo.

O objetivo dele pode se assemelhar a um código que represente um semáforo, visto que em um semáforo 3 luzes diferentes variam seus níveis de tensão alternadamente e mantém, um por vez, os seus níveis de tensão elevados por um determinado período de tempo.