

# Projeto de Sistemas RF

## Aula 01 - Revisão de Programação

# Apresentação

---

Esta aula vai servir para relembrarmos alguns conceitos sobre programação. Mas o que é mesmo programação? Se puxarmos pela memória, lembraremos que programar significa dizer o que um computador ou dispositivo tem que fazer. Mas como dizemos o que um computador deve fazer?. Primeiro, temos que saber o que o computador sabe fazer, não é mesmo?

Pois bem, lembrando dos conceitos que o professor ensinou na disciplina de Programação Estruturada, nos vêm à mente alguns “comandos”, tais como “escreva”, “repita”, “se isso, faça aquilo”, “enquanto” etc. Programar significa passar esses comandos para o computador para que ele possa ir executando e cumprindo a tarefa para a qual você o programou.

Nas próximas seções, vamos relembrar esses comandos e como nós os inserimos no computador. Além do mais, vamos ver alguns aspectos de como o computador trata os números e exemplos disso tudo.

## Objetivos

Ao final desta aula, você será capaz de:

- Relembrar os conceitos mais importantes de programação e algoritmos.
- Implementar programas simples que contenham laços, condicionais etc.
- Descrever como o fluxo de um algoritmo funciona.

# Como dizer ao computador o que fazer?

---

Programar um computador significa dizer o que ele tem que fazer. Mas como dizemos isso a ele? Na verdade existem várias formas de fazer isso. Essas formas são chamadas de linguagens de programação de computadores.

Outra coisa importante para lembrar é como nós inserimos os comandos no computador. Para isso, digitamos esses comandos (e seus parâmetros) em um arquivo de texto (só texto, não vale usar o Word) na sequência em que queremos que o computador os execute. Quando salvamos esses arquivos, damos a eles nomes que lembrem o que o programa faz e uma extensão padrão que depende da linguagem utilizada (por exemplo, `media.c` ou `ordena_nomes.c`).

Depois de digitados os comandos, usamos um programa especial chamado compilador, o qual vai transformar o arquivo de texto que digitamos com os comandos que queremos que o computador execute em um código especial que só o computador entende, chamado código de máquina.

## Linguagens que o computador entende

Só para lembrar, existe uma variedade de linguagens de computador. A seguir, citamos algumas muito importantes.

- C
- C++
- Java
- Basic
- Perl
- Python
- PHP
- Ruby

- Lisp

Poxa, quanta linguagem, hein? E olha que isso é só a ponta do iceberg. Existem, hoje, por volta de 2000 linguagens diferentes!

## Se Liga!

Neste curso, a linguagem que iremos usar é a linguagem C. Lembre-se de que a linguagem C é parecida com algoritmo, só que é em inglês. Que bom que no curso temos a disciplina de Inglês, hein?

## Código de máquina? O que é isso mesmo?

---

Bem, o que o computador usa na verdade é o código de máquina, mas o que vem a ser isso? Código de máquina é uma sequência de números (isso mesmo, são só números) que o computador entende como comandos, dependendo da ordem em que aparecem no arquivo. Exemplos de código de máquina são arquivos com extensões .exe, .dll, .src etc. É bom lembrar que nem toda linguagem vira código de máquina de uma vez só. Linguagens como PHP, Rubi e outras possuem (ao invés de compiladores) interpretadores que vão transformando os comandos que nós demos em código de máquina na medida em que vão sendo executados.

## Comandos básicos e variáveis

É bom lembrar também que em um programa podemos definir variáveis, que são símbolos (pequenas palavras) determinados no meio do programa para armazenar um valor do qual precisemos. As variáveis podem ter vários tipos, como:

- real
- inteiro
- cadeia de caracteres
- sequência ou arranjo de valores

Existe uma “linguagem”... Bem, não é bem uma linguagem de programação de computador, mas serve para aprender, chamada portugol. Essa linguagem é usada para ensinar programação (você provavelmente já até usou). E cada comando nessa linguagem possui um equivalente em outra linguagem (C, Pascal, Python, etc.). Vamos lembrar alguns deles:

1	se (condição) então
2	comando 1
3	comando 2
4	...
5	fim do se

1	enquanto (condição) faça
2	comando 1
3	comando 2
4	...
5	fim do enquanto

1	Faça variável de valor 1 até valor 2 passo valor
2	comando 1
3	comando 2
4	...
5	Fim faça

1	Mais comandos:
2	escreva(texto)
3	variável = leia
4	pausa
5	além de muitas funções matemáticas
6	sin, cos, abs, rand etc.

## Se Liga!

Lembre-se: existem muitos (mas muitos mesmo) comandos que não dá tempo de listar aqui. Dependendo do que e de onde iremos programar, existem comandos para fazer uma infinidade de coisas e calcular outras também.

# O que é um microcontrolador?

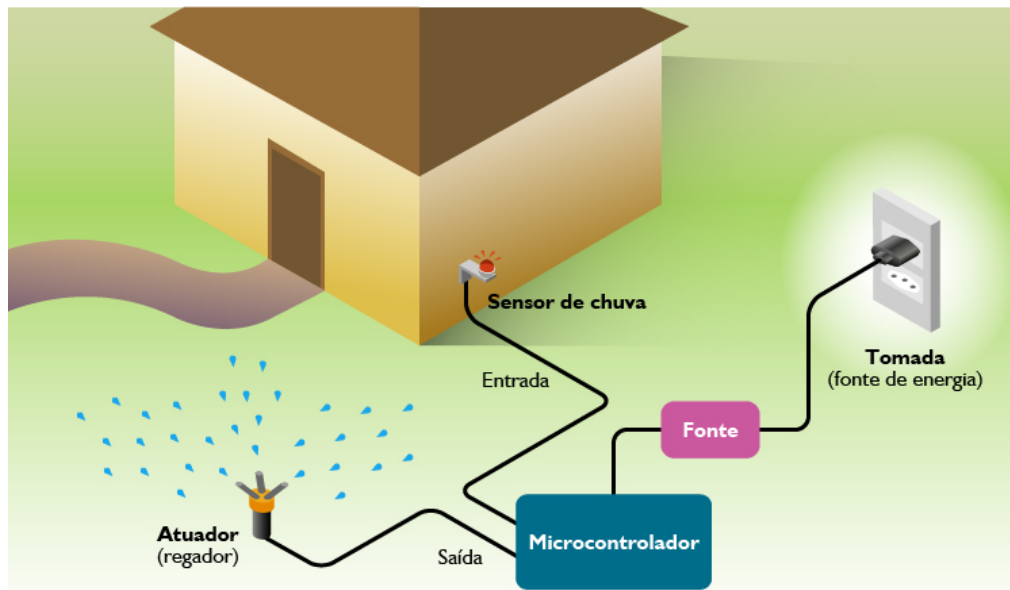
---

Nem sempre nós programamos um computador. Hoje em dia está cada vez mais comum programarmos dispositivos bem pequeninos. Um destes pequenos dispositivos é o microcontrolador. Ele é um circuito integrado que podemos ligar a um computador e escrever um programa que será executado nele (ao invés de ser executado no computador).

E por que precisamos programar microcontroladores? Bem, para isso precisamos lembrar o que um microcontrolador faz. Na verdade, ele é como se fosse um pequeno computador em um único chip (com capacidade de processamento bem menor), só que no lugar de monitor e teclado, ele tem pinos aos quais podemos ligar sensores e atuadores. Por exemplo, podemos ligar um sensor de chuva em um dos pinos e ligar o regador elétrico no outro (como mostra a Figura 1). Assim, podemos colocar um programinha que testa se o sensor detectou chuva, fazendo com que o regador desligue (para não gastar água em vão e ajudar a salvar o planeta).

Agora entendemos para que serve um microcontrolador. Ele serve para fazer tarefas bem simples que seriam um desperdício usar um computador para fazer. Já imaginou um computador no jardim para desligar a água, um computador no portão para abrir e fechar automático, até mesmo um em cada lâmpada de poste para desligar automaticamente?! É para esse tipo de coisa que usamos um microcontrolador.

**Figura 01** - Dispositivo que ajuda a salvar o planeta



## Programando o microcontrolador

Bem, como programamos então um microcontrolador se ele não tem teclado nem monitor? Na verdade, nós não programamos o microcontrolador diretamente, o que se faz é programar em um computador e aí sim enviar o programa para o microcontrolador (que deve estar ligado ao computador). A forma de programar é a mesma de um computador: escrevemos os comandos na sequência e em um arquivo de texto, então compilamos e geramos um arquivo com o código de máquina.

Agora, ao invés de rodar o arquivo de máquina no computador, usamos um programa para enviar e “gravar” esse código no microcontrolador. A partir daí podemos desligar o microcontrolador e utilizá-lo sozinho (ele vai ficar rodando sempre o mesmo programa).

## O que se “escreve” e “lê” de um microcontrolador?

Como o microcontrolador não tem teclado nem monitor, os comandos de “escrever” e de “ler” não existem como somos acostumados. No lugar deles existem comandos para “ler” informações de alguns pinos (que podem ter sensores ligados a eles), que são chamados de pinos de entrada, ou para “escrever” em pinos (que podem ter atuadores ligados a eles), que são chamados de saídas. Podemos ligar

pequenas chaves nos pinos de entrada e no meio do programa perguntar se as chaves estão ligadas ou desligadas. Podemos ligar LED nas saídas e mandar o LED acender ou apagar.

Neste curso, vamos ver como programar um microcontrolador para transmitir coisas através de uma conexão sem fio para outros micros. Imagine as possibilidades!

## Números binários

---

Neste curso, iremos precisar entender um pouquinho mais da linguagem que é exclusiva do computador, a linguagem binária. Por dentro de um computador, tudo são números. Os números significam as mais diversas coisas, desde comandos até a cor dos pontinhos da tela, passando pela tecla que você apertou no teclado.

Mais ainda, esses números são armazenados como números binários. Números binários, são números que só possuem 0 ou 1, por exemplo 1010011010. Nós estamos acostumados a usar números decimais, que são formados por 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9, como por exemplo, 24601, mas dentro do computador, os números são sempre binários.

Um número binário dentro do computador é armazenado em grupos (normalmente de 8 bits), e é muito importante que saibamos converter um número binário para decimal (que é o que estamos acostumados) ou vice-versa.

Só para lembrar, para converter um número binário para decimal, nós damos valores ao número 1 de acordo com a sua posição e somamos tudo. As posições são sempre potências de 2 ( $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$ ,...). Assim, o número 11111011100 em decimal é:

1	1	1	1	1	0	1	1	1	0	0
$2^{10} =$ 1024	$2^9 =$ 512	$2^8 =$ 256	$2^7 =$ 128	$2^6 =$ 64	$2^5 =$ 32	$2^4 =$ 16	$2^3 =$ 8	$2^2 =$ 4	$2^1 =$ 2	$2^0 =$ 1

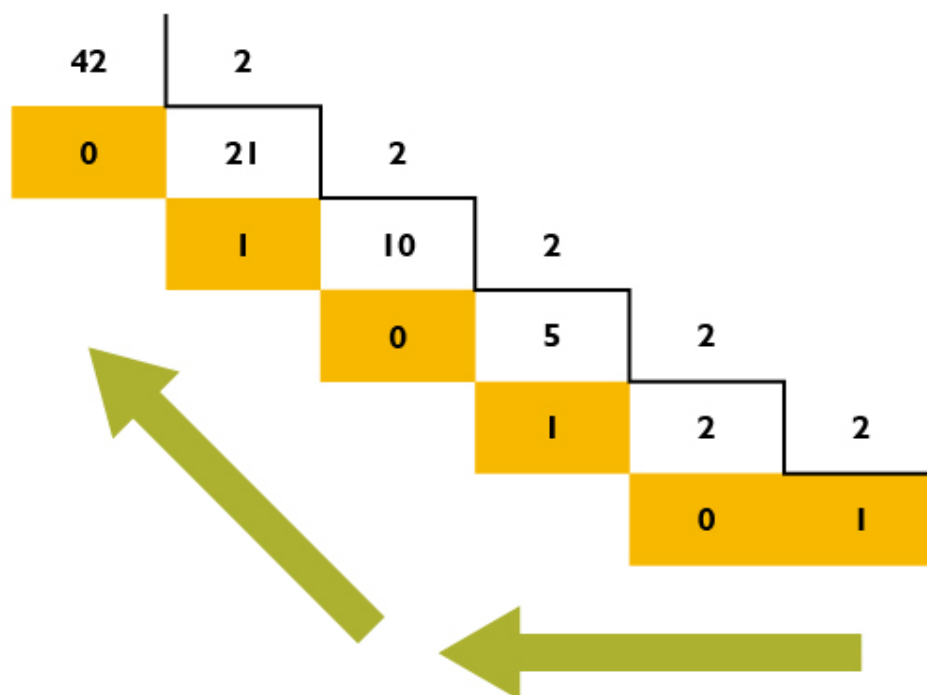
$$1024+512+256+128+64+16+8+4 = 2012$$



Repare que não somamos os números 1, 2 nem 32, pois eles tinham 0 em suas respectivas posições no número binário.

E para converter de decimal para binário? Bem, aí fazemos o processo de ir dividindo o número por 2 e pegando os restos. Por exemplo, o número 42 é convertido assim:

**Figura 02** - Ilustração da conversão do número 42 em binário. As setas indicam o sentido que se lê o resultado



Então, 42 em binário é 101010. Dessa maneira convertemos qualquer número inteiro em binário e vice-versa.

## Desafio

Será que você consegue fazer um programa (em C, por exemplo) que converte de decimal para binário e de binário para decimal?

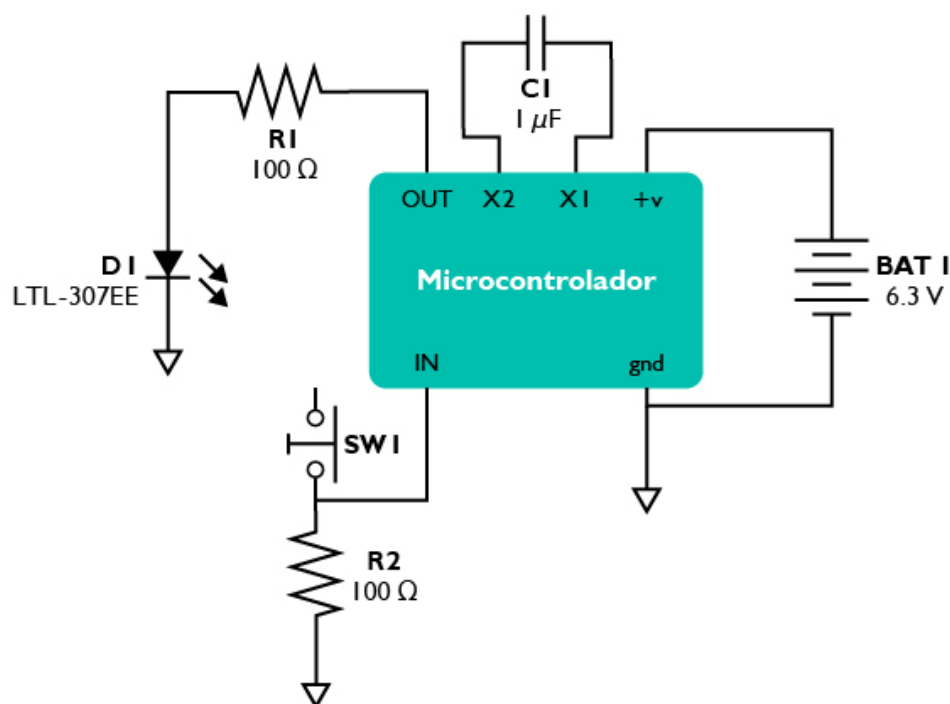
## Se Liga!

Em um microcontrolador, também só existem números binários. Inclusive, os pinos de entrada e saída são lidos todos de uma só vez como um grande número binário. Para saber qual dos pinos está ativo ou não, precisamos saber como ler só um bit de um número binário.

## Exemplo de um programa de pisca-pisca em um microcontrolador

A seguir, mostramos um exemplo de um programa que implementa um pisca-pisca em um microcontrolador. O microcontrolador deste exemplo é chamado de Arduino e está ficando bem popular nesses últimos tempos. A ligação é mostrada na Figura 3.

**Figura 03** - Ligação do microcontrolador para o exemplo do pisca-pisca



O programa fica assim:

```
1 void setup() {  
2   // Ajusta o pino 1 do micro-controlador para funcionar como saída  
3   pinMode(1, OUTPUT);  
4   // Ajusta o pino 2 do micro-controlador para funcionar como entrada  
5   pinMode(2, INPUT);  
6 }  
7 // aqui vem o loop principal do programa (esse loop vai ficar rodando repetidamente  
8 // enquanto o micro-controlador estiver ligado)  
9 void loop() {  
10  if (digitalRead(2)) { // verifica se o pino 2 está ligado  
11    digitalWrite(1, HIGH); // liga o led  
12    delay(1000); // espera 1 segundo  
13    digitalWrite(1, LOW); // desliga o led  
14    delay(1000); // espera 1 segundo  
15  }  
16 }
```

Como funciona esse programa? Bem, o Arduino sempre executa o que vem em `setup()` primeiro. Depois ele fica repetindo o que vem dentro de `loop()`. Dessa maneira, ele vai executar primeiro os comandos que configuram os pinos do microcontrolador, como entradas e saídas (pois em uma vai ficar o LED e na outra vai ficar a chave que aciona o pisca-pisca).

Depois de configurar, o Arduino vai ficar repetindo os comandos que testam se a chave está ligada (linha 10) e, caso esteja, pisque o LED uma vez (linhas 11 a 14).

## Se Liga!

Linhas ou partes que têm `//` são apenas comentários. O compilador não vai fazer nada com essas linhas (elas não são comandos). Elas servem para organizar o nosso programa e nos lembrar depois o que fizemos.

## Desafio

Será que dá para fazer um pisca-pisca “remoto”? Ou seja, que ele só pisque quando você apertar o botão estando do outro lado da sala? Parece mágica, não é? Quem sabe você aprende isso neste curso?

# Leitura Complementar

---

<[http://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](http://en.wikipedia.org/wiki/List_of_programming_languages)>

Nesse link, você encontrará uma lista super legal de quase todas as linguagens que existem. Vale a pena conferir!

<<http://pt.wikipedia.org/wiki/Microcontrolador>>

Aqui você pode conferir nesse site definições e informações sobre o que é um microcontrolador.

<<http://pt.wikipedia.org/wiki/Algoritmo>>

Essa página explica o que é um algoritmo e possui algumas definições importantes.

<<http://www.apoioinformatica.inf.br/produtos/visualg>>

Essa é a página de um software que serve para programar direto na linguagem portugol. O programa chama-se Visualg.

## Resumo

---

Nesta aula, fizemos uma revisão sobre programação e enfatizamos a programação para microcontroladores (que é o que vamos usar neste curso). Lembramos dos comandos, conceitos (como variáveis e números binários), utilizações e vimos até um pequeno exemplo de um programa de pisca-pisca. Agora, é tentar praticar bastante programação para ficar craque e sair programando tudo que é dispositivo que encontrar pela frente.

# Autoavaliação

---

1. Cite pelo menos 3 linguagens de programação de computadores.
2. Qual a diferença entre um microcontrolador e um computador?
3. Cite pelo menos 3 tipos de variáveis.
4. Quanto é 1942 em binário?
5. Quanto é 11000101010010010 em decimal?

Para verificar a resposta, clique [aqui](#).

## Resposta

1. Cite pelo menos 3 linguagens de programação de computadores.

C++, Java, Python.

2. Qual a diferença entre um microcontrolador e um computador?

O microcontrolador é como se fosse um pequeno computador em um único chip (com capacidade de processamento bem menor), só que no lugar de monitor e teclado, ele tem pinos aos quais podemos ligar sensores e atuadores. Eles são normalmente utilizados para realizar tarefas simples, diferentes dos computadores que realizam tarefas extremamente complexas.

3. Cite pelo menos 3 tipos de variáveis.

Real, Inteiro e cadeia de caracteres.

4. Quanto é 1942 em binário?

$$\frac{1942}{2} = 971 \text{ e resto } 0$$

$$\frac{971}{2} = 485 \text{ e resto } 1$$

$$\frac{485}{2} = 242 \text{ e resto } 1$$

$$\frac{242}{2} = 121 \text{ e resto } 0$$

$$\frac{121}{2} = 60 \text{ e resto } 1$$

$$\frac{60}{2} = 30 \text{ e resto } 0$$

$$\frac{30}{2} = 15 \text{ e resto } 0$$

$$\frac{15}{2} = 7 \text{ e resto } 1$$

$$\frac{7}{2} = 3 \text{ e resto } 1$$

$$\frac{3}{2} = 1 \text{ e resto } 1$$

$$\frac{1}{2} = 0 \text{ e resto } 1$$

Lendo de baixo para cima os restos temos: 11110010110

5. Quanto é 11000101010010010 em decimal?

Utilizando cada bit do número binário como multiplicador da exponenciação de base 2 e a posição do bit (que começa em zero, sendo zero o bit mais a direita e aumentando-se esse valor até o bit mais a esquerda) como expoente do 2, seguindo a fórmula:

$$\text{Valor do bit da posição } x \times 2^x$$

Soma-se o valor de cada bit convertido e temos o seguinte:

$$1 \times 2^{16} + 1 \times 2^{15} + 0 \times 2^{14} + 0 \times 2^{13} + 0 \times 2^{12} +$$

$$\begin{aligned} &1 \times 2^{11} + 0 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + \\ &0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + \\ &1 \times 2^1 + 0 \times 2^0 = 101010 \end{aligned}$$

## Referências

---

BIANCHI, Francisco et al. **Algoritmos e programação de computadores**. São Paulo: Editora CAMPUS, 2012.

PEREIRA, Fabio. **Microcontroladores PIC: Programação em C**. 2. ed. São Paulo: Editora Erica, 2003.

PEREIRA, Pedro; RODRIGUES, Pimenta; SOUSA, Manuela. **Programação em C++**. 10. ed. Lisboa: Editora FCA (BRASIL), 2010.

SENNE, Edson Luiz França. **Primeiro curso de programação em C**. 3. ed. Florianópolis: Editora Visual Books, 2009.

SOUZA, Vitor Amadeu. **Programação em C para o AVR: fundamentos**. São Paulo: Editora Ensino Profissional, 2011.