

NOME:

## AULA 5 – NÚMEROS BINÁRIOS

Quando falamos que o computador é de 32 bits ou 64 bits, o que isso quer dizer? E por que não podemos colocar memórias maiores que 4 GB em computador com 32 bits.

## MARCAS DE CONTÁGEM

## O SISTEMA DECIMAL

## O SISTEMA BINÁRIO

## O SISTEMA HEXADECIMAL

**VAMOS CONTAR ATÉ TRINTA**

[illegible]

## COMO CONVERTER DE BINÁRIO PARA DECIMAL

Computadores usam o sistema binário. Há outros sistemas, como visto na tabela anterior, mas vamos focar neste sistema. Seja um número binário qualquer. Por exemplo:

1 0 1 1

Vamos numerar os caracteres da direita para a esquerda, começando pelo zero:

<b>Número binário</b>	1	0	1	1
<b>Posição</b>	3	2	1	0

Para determinar o valor do número binário, basta elevar a posição como uma potência de 2 e multiplicar pelo valor do caractere correspondente e depois somar todos os números. Vejamos:

<b>Número binário</b>	1	0	1	1
<b>Posição</b>	3	2	1	0
<b>Operação 2<sup>posição</sup></b>	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
<b>Multiplicando pelo caractere</b>	$1 \times 8 = 8$	$0 \times 4 = 0$	$1 \times 2 = 2$	$1 \times 1 = 1$

Agora é só somar o resultado da última linha:

$$8 + 0 + 2 + 1 = 11$$

Mas como diferenciamos se um número é binário ou decimal?  
Resolvemos isso com um subscrito, ou seja

$$1011_2 = 11_{10}$$

que pode ser lido como: “um zero um um na base dois é igual à um um na base dez” ou “um zero um um na base dois é igual à onze na base dez”.

Vamos treinar um pouco.

## EXERCÍCIOS

Converta os números abaixo, em binário, para decimal.

$1, 1_1 =$

**2.  $100_1 =$**

**3.  $11_1 =$**

4.  $1\,000_1 =$

5.  $1111_1 =$

**6.  $1\ 0101_2 =$**

**7.**  $1111\ 1111_1 = 2$

## MEMÓRIA DO COMPUTADOR E OS NÚMEROS BINÁRIO

Toda informação deve estar guardada em um espaço de memória de um computador. Assim, se você quiser guardar o número de carteiras na sala de aula, por exemplo, deve reservar um espaço de memória de um computador.

Digamos que você reservou um byte apenas, isto é, 8 bits. Isso quer dizer que você pode guardar desde o número

0000 0000<sub>1</sub>

até o número

1111 1111<sub>1</sub>

## PROFESSOR DANILO

que, convertendo para a base dez, vai de 0 até 255 (veja exercício 7). Ou seja, apenas um byte é suficiente para armazenar o número de carteiras de cada sala de aula, pois provavelmente não teremos mais que 255 carteiras em uma sala. Mas isso é suficiente para armazenar a quantidade de todas as carteiras na escola?

No Arduino, é comum falarmos que uma porta tem resolução de 10 bits, assim, quer dizer que ele aceita números que vão de 0 até  $2^{10} - 1 = 1023$ .

Ou seja, se tivermos  $n$  bits de espaço em memória, podemos armazenar números que vão de 0 até  $2^n - 1$ .

**O SISTEMA DECIMAL**

Agora podemos finalmente voltar à pergunta no início da aula. Primeiro, vamos determinar qual o maior número armazenado, no sistema decimal, em uma memória de 32 bits:

$$2^{32} - 1 = 4\,294\,967\,295$$

Isto é, computadores de 32 bits podem armazenar números que vão de 0 até 4 294 967 295.

Agora, vamos converter 1 GB

$$4\text{ GB} = 4 \times 1\,073\,741\,824\text{ Bytes}$$
$$= 4\,294\,967\,295\text{ Bytes}$$

Mas olha que curioso: os 32 bits de um computador é um espaço especial da memória que serve para armazenar informações de endereço de cada componente, incluindo de cada byte de memória, assim, não há espaço para armazenar o endereço de cada espaço de memória, e por isso não é possível usar mais que 4 GB (na verdade, o limite é até menor). Mas e o computador de 64 bits, quanto ele pode armazenar? Fazendo as contas com uma calculadora especial, chegamos em

$$18\,446\,744\,073\,709\,551\,616$$

ou dezoito quintilhões quatrocentos e quarenta e seis quatrilhões e setecentos e quarenta e quatro trilhões e setenta e três bilhões e setecentos e nove milhões e quinhentos e cinquenta e um mil e seiscentos e dezesseis.

Espaço para colocar todos os endereços é o que não falta, por isso, praticamente não há limite de memória RAM em computador de 64 bits, uma vez que é possível endereçar a quantidade de memórias que há no mercado, mesmo colocando várias delas ao mesmo tempo.

**ATIVIDADE PRÁTICA**

Vamos considerar que nosso kit irá armazenar dados: como temos 8 botões, podemos dizer que temos um minicomputador manual de 8 bit.

Assim, vamos colocar oito LEDs e representar alguns números em binário.

Como atividade prática, além de montar o circuito descrito abaixo, **represente os números usados nos exercícios feitos em sala de aula**. Na ausência de números à esquerda, considere como sendo o número 0.

**O CIRCUITO**

Vamos usar

- 8 LEDs da cor que você quiser;
- 8 resistores de 200 ohm;
- 9 fios.

Você deverá realizar as seguintes conexões:

- O pino menor de cada LED no GND;
- Cada pino maior do LED deverá ir ao terminal de um resistor (cada LED com um resistor);
- O outro terminal de cada resistor deve ir à cada um dos bornes (de B1 a B8).

9º ANO – ROBÓTICA – 05/04/2024

Note que você pode consultar a figura a seguir, que mostra a numeração dos bornes de nosso kit.



Figura 1: Nomes dos bornes do nosso kit

**Não usaremos as saídas de 5V nem de 9V.**

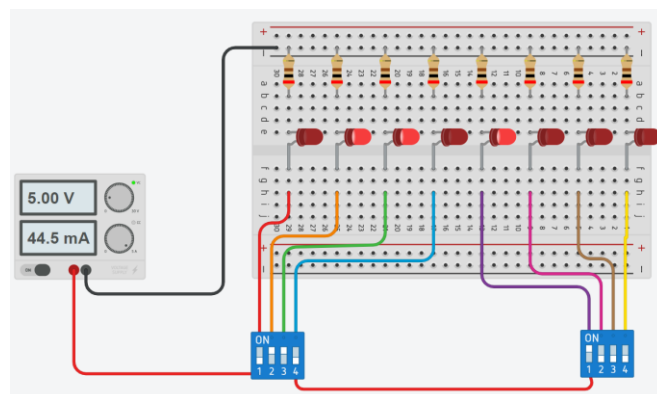


Figura 2: Esquema de ligação usando o Tinkercad para simular o nosso kit

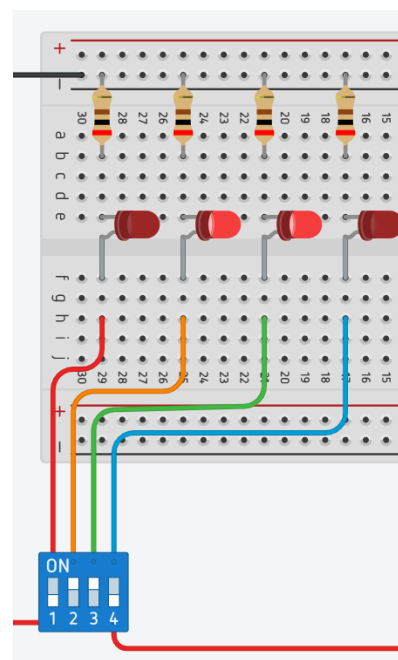


Figura 3: Detalhe do esquema de ligação de alguns bornes