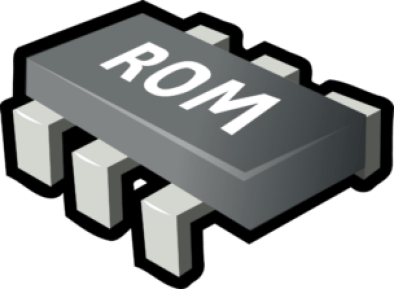
**Memória não Volátil EEPROM do ESP8266**

[**EEPROM**](https://portal.vidadesilicio.com.br/memoria-nao-volatil-arduino-eeprom/) é um tipo de **memória não volátil** presente na maioria dos sistemas e micro controladores, já que quase sempre é preciso guardar dados para que sejam lidos mesmo após a uma queda de energia por exemplo. Nesse tutorial você aprenderá a usar essa memória não volátil do [**ESP8266**](http://www.vidadesilicio.com.br/esp.html) usando o [**NodeMCU**](http://www.vidadesilicio.com.br/esp/wifi-esp8266-nodemcu-esp-12e.html)

O principal propósito da EEPROM, é guardar as informações (em Bytes) para que possamos ler a qualquer momento, e caso o sistema venha a reiniciar ou algo do tipo, as informações se manterão intactas



**Índice**[[Exibir](https://portal.vidadesilicio.com.br/eeprom-esp8266-memoria-nao-volatil/)]

**Sobre a EEPROM**

**EEPROM**, é uma memória não volátil, mas tem um limite relativamente pequeno de gravações (no ESP8266). Isto faz com que devemos tomar cuidado para por exemplo, **não colocar as funções de gravação em loop’s**. Normalmente é utilizada para armazenar dados sobre a inicialização do sistema ou algo relacionado que não seja necessário ser alterado com muita frequência.

**O ESP8266 não tem EEPROM dedicada**, porém, há uma emulação em um segmento da FLASH de 4B até 4096KB. **Cada Byte** da EEPROM no ESP8266, tem um limite de 10mil gravações e “infinitas” leituras. Sabendo disto, tome cuidado para não extrapolar nas gravações; a leitura é permitida quantas vezes forem necessárias.

Alguns exemplos práticos para a EEPROM:

* Manter a maior temperatura alcançada de um sensor.
* Guardar variáveis que o usuário inserir após o inicio padrão do sistema também é comumente usado, por exemplo, alocar as novas credenciais do WiFi após o cliente comprar o produto novo.
* Salvar o último dia em que um motor parou de funcionar para gerar relatórios.

As aplicações da EEPROM são as mais diversas e variadas. Pode estar presente na maioria dos projetos já que é facilmente manipulável.

**Mãos à obra – Utilizando a EEPROM do NodeMCU**

**Componentes utilizados**

* [NodeMcu ESP8266 ESP-12E](http://www.vidadesilicio.com.br/esp/wifi-esp8266-nodemcu-esp-12e.html)

**Software**

**#include <EEPROM.h>**

**void** setup()

{

Serial.begin(9600);//Inicia a comunicação Serial.

Serial.println();//-

Serial.println();//-Apenas para "limpar" a tela.

EEPROM.begin(4);//Inicia a EEPROM com tamanho de 4 Bytes (minimo).

Serial.println(EEPROM.read(0));//Mostra no Monitor oque há antes de efetuar a gravação

**if** (EEPROM.read(0) != 18)//Se não existir o numero escolhido (18), irá gravar. Isto é necessário para evitar regravações desnecessárias.

{

EEPROM.write(0, 18);//Escreve no endereço "0", o valor "18".

EEPROM.commit();//Salva o dado na EEPROM.

Serial.println(EEPROM.read(0));//Mostra o valor no endereço 0 novamente.

}

**else**//Se já existir o valor, irá avisar.

{

Serial.println("Dado ja cadastrado");

}

EEPROM.end();//Fecha a EEPROM.

}

**void** loop()

{

}

**Entendendo a fundo**

**Software**

**-Função EEPROM::begin()**

EEPROM.begin(4);

Necessária sempre que for ler ou escrever na EEPROM. O parâmetro é o tamanho de inicialização, que precisa estar entre **4 até 4096**. Neste caso, foi iniciado o mínimo possível, 4 Bytes.

**-Função EEPROM::read()**

EEPROM.read(0);

Faz a leitura do endereço desejado, retornando um valor **UINT8** (0 até 255). No caso do exemplo do tutorial, usamos o primeiro endereço da EEPROM, que é “0”. Você pode escolher qualquer endereço desde que esteja entre o parâmetro usado no **begin()**.

**-Função EEPROM::write()**

EEPROM.write(0, 18);

Escreve no endereço X, o valor Y. No caso do exemplo, escrevemos no endereço “**0**“, um número qualquer, “**18**“. Esta escrita deve ser feita em UINT8 (0 até 255), caso seja necessário valores maiores, precisara usar mais Bytes da EEPROM.

**-Função EEPROM:commit()**

EEPROM.commit();

Esta função salva o que escrevemos anteriormente, **sem isto, não irá salvar!** Também pode ser feito pelo comando **EEPROM.end()**, que também efetua o **commit()**.

**-Função EEPROM::end()**

EEPROM.end();

Fecha a EEPROM. Lembre-se que após usar esta função, não é possível escrever ou ler valores da EEPROM!

**Curiosidades**

Ao efetuar um teste que mostra o valor de cada Byte da EEPROM, os 4 primeiros Bytes do meu ESP8266 já estavam com valores modificados. Penso que tenha sido alguma biblioteca externa ou ?…. Teste você mesmo e diga nos comentários se também havia algo alterado!

**Código para efetuar a leitura completa da EEPROM:**

**#include <EEPROM.h>**

**void** setup()

{

Serial.begin(9600);

Serial.println();

Serial.println();

EEPROM.begin(4096);//Inicia toda a EEPROM.

**for** (**int** i = 0; i < 4097; i++)//Loop que irá mostrar no Serial monitor cada valor da EEPROM.

{

Serial.print(EEPROM.read(i));

Serial.print(" ");

}

EEPROM.end();//Fecha a EEPROM.

}

**void** loop()

{

}