

# TUTORIAL MEMÓRIA FLASH

Autor: Tiago Lone

Nível: Básico

Criação: 11/12/2005

Última versão: 18/12/2006



**Maxwell Bohr**  
Instrumentação Eletrônica

<http://www.maxwellbohr.com.br>  
[contato@maxwellbohr.com.br](mailto:contato@maxwellbohr.com.br)

**PdP**

Pesquisa e Desenvolvimento de Produtos

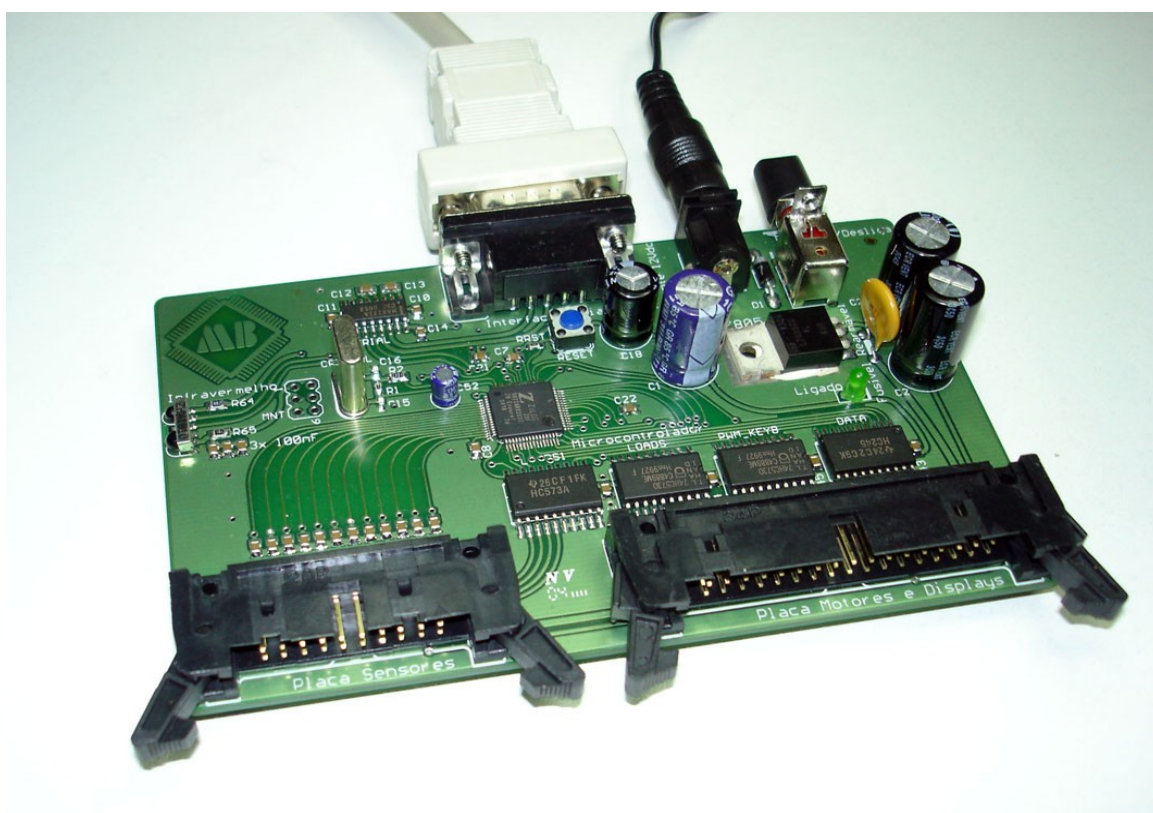
<http://www.automato.com.br>  
[atendimento@automato.com.br](mailto:atendimento@automato.com.br)

## 1 – Introdução

Nesse tutorial vamos aprender como armazenar informações na memória não volátil do Kit Didático de Robótica. Isso permite que possamos guardar informações que permanecem armazenadas no Kit mesmo após esse ser desligado e religado. Vamos criar um projeto que nos permita gravar, apagar e ler uma mensagem de texto nessa região de memória.

## 2 – Material

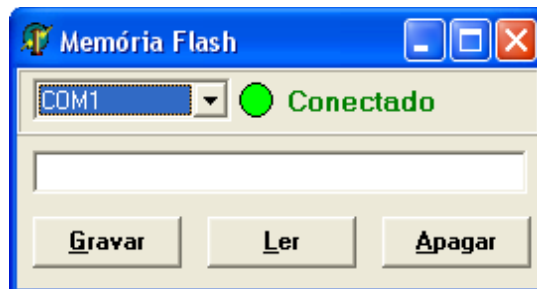
Para esse tutorial é necessário apenas o *Módulo Principal*, que é onde está o microcontrolador e internamente a esse, a memória não volátil do tipo flash do KDR5000. Para a criação do programa será necessário o Borland Delphi 6.



*Figura 1: Módulo Principal do Kit Didático de Robótica.*

## 3 – Projeto

Para auxiliar no entendimento do que deve ser feito para utilizar a memória Flash do Kit, será criado um programa que permite ler, gravar e apagar dados da memória flash. Esse programa será bem simples e irá se parecer com o seguinte.



*Figura 2: Programa que será criado nesse tutorial.*

Nosso primeiro passo é criar a interface gráfica. Vamos utilizar o projeto criado no tutorial Base que já nos fornece algumas funcionalidades interessantes. Para isso copiamos o projeto daquele tutorial e em cima dele vamos adicionar alguns componentes gráficos extras.

A interface do programa é composta por um componente Edit, utilizado para entrada do texto que será gravado na memória, e três componentes do tipo Button, que são botões, um para gravar o texto, um para ler o texto gravado na memória e por fim um botão que limpa a memória. Esses dois tipos de componentes podem ser encontrados na aba de componentes “Standard”.



*Figura 3: Aba "Standard" da Barra de componente.*

O componente Edit possui o seguinte ícone.



*Figura 4: Ícone do componente Edit.*

O componente Button possui o seguinte ícone.



*Figura 5: Ícone do componente Button.*

Primeiro adicionamos um componente Edit. Temos que modificar as seguintes propriedades desse componente.

**Name** = EditTexto

**Text** = (deixar em branco)  
**MaxLength** = 99

Nosso Form ficará da seguinte forma.



*Figura 6: Form após adição de um Edit para entrada de texto.*

Vamos adicionar agora os três botões. Eles vão possuir propriedades semelhantes e a seguir essas propriedades foram listadas.

**Name** = ButtonGravar  
**Caption** = Gravar  
**Font/Style/fsBold** = true

**Name** = ButtonLer  
**Caption** = Ler  
**Font/Style/fsBold** = true

**Name** = ButtonApagar  
**Caption** = Apagar  
**Font/Style/fsBold** = true

Após a adição desses três botões a interface gráfica está completa. Vamos apenas modificar um detalhe, a propriedade Caption do Form principal, que ainda possui a mensagem atribuída a ela no projeto Base. Mudaremos essa propriedade de “Projeto Base” para “Memória Flash”.



*Figura 7: Interface gráfica completa do programa.*

Com a interface gráfica completa vamos para a implementação das funcionalidades. O primeiro método da biblioteca de controle que trabalha com a memória flash que veremos é o método de gravação de informações. Esse método possui a declaração a seguir.

```
Procedure FlashWrite(data : DynByteArray);
```

Esse método possui um parâmetro “data”. É esse parâmetro que irá indicar ao método que dados deverão ser gravados na memória flash. Ele é do tipo DynByteArray, que é um array de bytes de tamanho variável. Podemos definir o tamanho de uma variável do tipo DynByteArray utilizando o método SetLength. Podemos também descobrir o tamanho de um array desse tipo utilizando o método Length.

Serão gravados na memória todos os bytes do array passado como parâmetro. É importante lembrar que essa região de memória pode armazenar no máximo 100 bytes.

Utilizaremos esse método no manipulador de evento do clique do botão de gravação de dados, o evento OnClick. Precisamos criar um manipulador para o evento OnClick desse botão. Para fazer isso podemos selecionar o componente Button correspondente, ir no **Object Inspector**, selecionar a aba Events e dar um duplo clique sobre a linha que está escrito OnClick. Uma forma mais fácil de fazer isso é apenas dar um duplo clique sobre o botão no Form e com isso o Delphi irá criar automaticamente um manipulador para o evento OnClick. O seguinte código será criado.

```
Procedure TFormMain.ButtonGravarClick(Sender:
                                                    TObject);

begin
end;
```

Iremos implementar a seguir o código de gravação da mensagem de texto na memória flash dentro desse manipulador de eventos.

```
Procedure TFormMain.ButtonGravarClick(Sender:
                                                    TObject);
```

```

var
    // Contador
    i : Integer;

    // Array de bytes com os dados
    dados : DynByteArray;
begin
    // Verifica se o texto do Edit não é vazio
    if Length(EditTexto.Text) > 0 then
    begin
        // Seto o tamanho do array de bytes para o
        // número de caracteres do texto mais um.
        SetLength(dados, Length(EditTexto.Text)+1);

        ...
    end;
end;

```

Esse código que adicionamos declara duas variáveis que vamos utilizar nesse método, uma variável do tipo Integer que vamos utilizar como contador e uma variável do tipo DynByteArray que irá conter os bytes que desejamos que sejam gravado.

Além da declaração dessas variáveis, verificamos se o texto no componente Edit não é nulo, isto é, verificamos se ele possui um tamanho maior que zero. Para saber o tamanho do texto utilizamos o método Length. Se o texto for maior que zero então definimos que o tamanho do array “dados” será igual ao número de caracteres do texto mais um.

Setamos o tamanho do array para o numero de caracteres mais um porque vamos utilizar a primeira posição desse array para armazenar o tamanho da mensagem de forma que quando ela for lida da memória possamos saber o seu tamanho. Para setar esse tamanho utilizamos o método SetLength e para saber o tamanho do texto no Edit utilizamos novamente o método Length.

Com isso temos nossas variáveis preparadas para serem utilizadas. Precisamos agora copiar os bytes da mensagem para dentro do array de bytes e gravar esses dados na memória flash do Kit. A seguir o código responsável por isso.

```

Procedure TFormMain.ButtonGravarClick(Sender:
                                                                    TObject);

var
    // Contador

```

```

i : Integer;

// Array de bytes com os dados
dados : DynByteArray;
begin
// Verifica se o texto do Edit não é vazio
if Length(EditTexto.Text) > 0 then
begin
// Seta o tamanho do array de bytes para o
// número de caracteres do texto mais um.
SetLength(dados, Length(EditTexto.Text)+1);

// A primeira posição do array irá conter o
// tamanho do texto
dados[0] := Length(EditTexto.Text);

// Copia para o array os caracteres do texto
for i:=1 to Length(EditTexto.Text) do
    dados[i] := BYTE(EditTexto.Text[i]);

// Grava na memória flash do Kit
Kit.FlashWrite(dados);
end;
end;

```

Primeiro descobrimos o tamanho do texto utilizando o método Length e atribuímos esse valor à primeira posição do array de bytes. Em seguida utilizamos um laço “for” para copiar o valor dos caracteres do texto para o array de bytes. Como esses caracteres são do tipo “CHAR” e o array de bytes armazena valores do tipo “BYTE”, então temos que fazer um “cast” de forma a converter os valores do tipo “CHAR” para valores do tipo “BYTE”. Para saber mais sobre “cast” de tipos procure em alguma documentação sobre programação Delphi.

Após ter copiado todos os caracteres para o array de bytes temos que gravá-los na memória flash. Fazemos isso chamando o método FlashWrite do objeto “Kit”, passando como parâmetro o array de bytes “dados”. Assim, finalizamos a implementação do manipulador do evento OnClick do botão de gravação.

O próximo passo é implementar o código de leitura dos dados. Assim como criamos um manipulador de eventos para o botão de gravação, vamos criar um para o botão de leitura. Para fazer isso podemos selecionar o componente Button de leitura, ir no **Object Inspector**, selecionar a

aba Events e dar um duplo clique sobre a linha que está escrito OnClick ou apenas dar um duplo clique sobre o botão no Form e com isso o Delphi irá criar automaticamente um manipulador para o evento OnClick daquele botão. O seguinte código será criado.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                         TObject);
begin
end;
```

Vamos adicionar o código de leitura dentro desse manipulador.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                         TObject);

var
    // Contador
    i : Integer;

    // Array de bytes com os dados
    dados : DynByteArray;

    // String que irá armazenar o texto
    texto : String;

begin
    // Lê todos os bytes da memória
    Kit.FlashRead(dados, 100);
    ...
end;
```

Esse início do código declara as variáveis que vamos utilizar nesse método, um contador, um array de bytes e uma string. Após a declaração das variáveis, utilizamos o método FlashRead para ler todos os 100 bytes da memória flash do Kit e armazenar no array “dados”. O método FlashRead possui a seguinte declaração.

```
Procedure FlashRead(var data : DynByteArray; num :
                                         Integer);
```

Esse método possui dois parâmetros. O primeiro um array de bytes que irá armazenar os dados lidos e o segundo um valor inteiro que indica quantos bytes devem ser lidos. Esse valor não deve ser maior do que 100. A seguir o final do código do manipulador do botão de leitura.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

var
    // Contador
    i : Integer;

    // Array de bytes com os dados
    dados : DynByteArray;

    // String que irá armazenar o texto
    texto : String;

begin
    // Lê todos os bytes da memória
    Kit.FlashRead(dados, 100);

    // Verifica validade do tamanho do texto
    if dados[0] < 100 then
    begin
        // Copia os caracteres do array para a String
        for i:=1 to dados[0] do
            texto := texto + CHAR(dados[i]);

        // Apresenta mensagem lida
        ShowMessage(texto);
    end;
end;
```

Após ler os dados, verificamos se a primeira posição do array de bytes, que contém o tamanho da mensagem, é menor do que 100, que é o número máximo de bytes que a memória pode armazenar. Se esse valor for válido utilizamos um laço do tipo “for” para criar, a partir dos caracteres do array de bytes, uma string temporária com a mensagem que deverá ser apresentada.

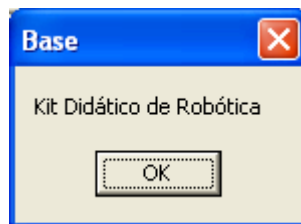
Utilizamos o operador de soma que vai concatenando, ao final da nossa string temporária, caracter por caracter da mensagem. Por fim, utilizando o método ShowMessage, apresentamos a mensagem ao usuário.

Com isso finalizamos a implementação do código de leitura e já podemos testar o programa. Para isso, vamos no menu **Run – Run** ou pressionamos F9. Se não houver nenhum erro o programa será compilado e executado. Com um Kit conectado em alguma porta serial podemos testar se o programa está funcionando. Para isso, escrevemos uma mensagem no Edit e pressionamos o botão gravar.



*Figura 8: O texto do Edit é gravado na memória pressionando-se o botão gravar.*

Em seguida pressionamos o botão de leitura e a mensagem deverá ser apresentada na tela. Teste esse programa com outras mensagens e verifique o resultado.



*Figura 9: Apresentação da mensagem gravada na memória.*

Para finalizar esse programa, vamos implementar o código do botão “Apagar”. Temos que criar um manipulador para o evento OnClick desse botão, como fizemos para os anteriores. A forma mais fácil de fazer isso é dar um duplo clique sobre o botão “Apagar”. O seguinte código será gerado.

```
Procedure TFormMain.ButtonApagarClick(Sender:
                                                    TObject);
begin
end;
```

Vamos utilizar o método FlashErase da biblioteca de controle do Kit. Esse método

apaga toda a memória flash do Kit. Na realidade ele apenas escreve em todas as posições da memória o valor 255. Com isso qualquer dado que estivesse gravado na memória será perdido. A declaração desse método é a seguinte.

```
Procedure FlashErase();
```

O manipulador do event OnClick do botão “Apagar” irá conter apenas uma linha. Essa linha apenas chama o método FlashErase. O código ficará da seguinte forma.

```
Procedure TFormMain.ButtonApagarClick(Sender:
                                                    TObject);
begin
    // Apaga dados da memória
    Kit.FlashErase;
end;
```

Quando esse botão for pressionado toda a memória flash será apagada. Qualquer bytes de dados que tenhamos gravado será sobrescrito por bytes com o valor 255. Se após clicar no botão “Apagar” tentarmos ler a mensagem pressionando o botão “Ler”, nada será apresentado. Isso ocorre porque, como o primeiro byte da memória flash, após ser apagada, será 255, a nossa checagem no início do manipulador do botão de leitura para verificar se o valor do primeiro byte do array, que indica o tamanho da mensagem, é menor que 100 vai ser falsa. Com isso não adianta tentar ler a memória flash, pois ela foi preenchida com uma mensagem com tamanho inválido. Com isso terminamos nosso tutorial sobre o uso de memória flash do Kit.

## 4 – Conclusão

Nesse tutorial vimos como gravar, ler e apagar dados da memória flash do Kit. Para isso criamos um programa que utiliza os três métodos de manipulação da memória não volátil do Kit fornecidos pela biblioteca de controle. São esses métodos FlashWrite, FlashRead e FlashErase. Com esses métodos é possível tirar todo o proveito da memória flash do Kit Didático de Robótica.