

# MSX Article

**MARMSX**

*A Memória do  
MSX na Prática*

## Resumo

O objetivo deste artigo é apresentar na prática alguns assuntos abordados no artigo “A Memória do MSX”.

## 1- Introdução

Serão propostas duas experiências práticas:

Experiência 1: copiar um bloco para a página 1 da RAM.

Experiência 2: criar ROM para cartucho.

## 2- Experiência 1: cópia do bloco para a página 1

O objetivo deste exercício é mostrar como copiar e executar um programa na página 1 da memória, que é ocupada pela ROM da BIOS no modo Basic.

O programa a seguir irá carregar o trecho assinalado em amarelo na página 1. Assim, a execução começa pelo trecho em verde, que é o responsável por fazer a cópia e executar o programa do trecho amarelo.

End	Assembly	Linha	Etiqueta	Instrução	Comentário
		10		ORG &H9000	; Início do programa na memória
9000	CD C3 00	20		CALL &HC3	; Rotina de CLS da BIOS
9003	C9	30		RET	; Retorna
9004	DB A8	40		IN A,(&hA8)	; Lê configuração atual de slots
9006	47	50		LD B,A	; Cópia em B
9007	CB 3F	60		SRL A	; Desloca 2 bits à direita
9009	CB 3F	70		SRL A	
900B	B0	80		OR B	; Faz slot pg 1 = slot pg 2
900C	D3 A8	90		OUT (&HA8),A	; Envia dado de A para porta A8
900E	AF	100		XOR A	; Limpa registrador F
900F	21 FF FF	110		LD HL,&HFFFF	;
9012	36 00	120		LD (HL),0	; Garante o sub-slot 0
9014	21 00 90	130		LD HL,&H9000	; Origem de cópia
9017	11 00 40	140		LD DE,&H4000	; Destino de cópia
901A	06 04	150		LD B,4	; Contador
901C	7E	160	LOOP:	LD A,(HL)	; Carrega valor da memória em A
901D	12	170		LD (DE),A	; Carrega valor de A para memória
901E	23	180		INC HL	; Desce ponteiro
901F	13	190		INC DE	; Desce ponteiro
9020	10 FA	200		DJNZ LOOP	; B=B-1: Se B>0, pule para LOOP
9022	CD 00 40	210		CALL &H4000	; Roda rotina na página 1
9025	DB A8	220		IN A,(&HA8)	; Lê configuração de slots
9027	E6 F0	230		AND &B1111000	; Máscara para fazer pag 1 = slot 0
9029	D3 A8	240		OUT (&HA8),A	; Envia dado de A para porta A8
902B	C9	250		RET	; Retorna

O programa deve ser compilado para o endereço de memória &H9000 (linha 10). Ao salvar em disco, deve-se fornecer o endereço de execução inicial, que é &H9004:

```
BSAVE"exp1.bin", &H9000, &H902B, &H9004
```

O programa em amarelo tem como função apenas apagar a tela. É um efeito que pode ser facilmente apreciado até pelos mais leigos.

A seguir, a explicação do programa em verde por trechos de código.

Linha	Etiqueta	Instrução	Descrição detalhada
40		LD A,&B10101000	Ajusta a página 1 para RAM.  Aqui um truque é aplicado para descobrir o slot da RAM: * A = s1s10000, onde s1 é o slot da RAM * B = A * A = 00s1s100 * B = s1s1s100 OU * A = s1s1s100
50		LD B,A	
60		SRL A	
70		SRL A	
80		OR B	
90		OUT (&HA8),A	
100		XOR A	Limpa o flag F. Necessário para as rots. da ROM.
110		LD HL,&FFFF	Força o sub-slot da página 1 a ser 0.
120		LD (HL),0	
130		LD HL,&H9000	Posiciona o ponteiro de origem dos dados
140		LD DE,&H4000	Posiciona o ponteiro de destino dos dados
150		LD B,4	Inicia contador, de acordo com o tamanho da cópia
160	LOOP:	LD A, (HL)	Copia os dados da página 2 para a página 1
170		LD (DE),A	
180		INC HL	
190		INC DE	
200		DJNZ LOOP	
210		CALL &H4000	Executa o programa em amarelo
220		IN A, (&HA8)	Ao retornar, devolve a página 1 para a ROM
230		AND &B1111000	
240		OUT (&HA8),A	
250		RET	Retona ao Basic

Encontra-se na página deste artigo o código fonte desse programa para o assembler RSCII, junto com o código binário. Os arquivos são:

- exp1.asm – código fonte para o assembler RSCII
- exp1.txt – código fonte em formato texto
- exp1.bin – código fonte em Assembly. Rode com bload mais a opção “,r”
- exp1.bas – programa em Basic com o código em linguagem de máquina.

Para rodar em Basic direto (sem compilar), deve-se digitar o seguinte programa:

```
10 FOR E=&H9000 TO &H902B
20 READ A$
30 A = VAL("&h"+A$)
40 POKE E,A
50 NEXT E
60 DEFUSR=&H9004 : X=USR(0)
```

```

70 DATA CD,C3,00,C9,DB,A8,47,CB,3F,CB,3F,B0,D3,A8,AF,21
80 DATA FF,FF,36,00,21,00,90,11,00,40,06,04,7E,12,23,13
90 DATA 10,FA,CD,00,40,DB,A8,E6,F0,D3,A8,C9

```

## 2.1- Testando no debugger do emulador blueMSX

O emulador blueMSX possui uma excelente ferramenta para ver o que está se passando no MSX enquanto um programa executa, que é o debugger. Ele pode ser acessado através do menu superior de ferramentas, opção debugger. A figura 1 apresenta o emulador e a ferramenta debugger.

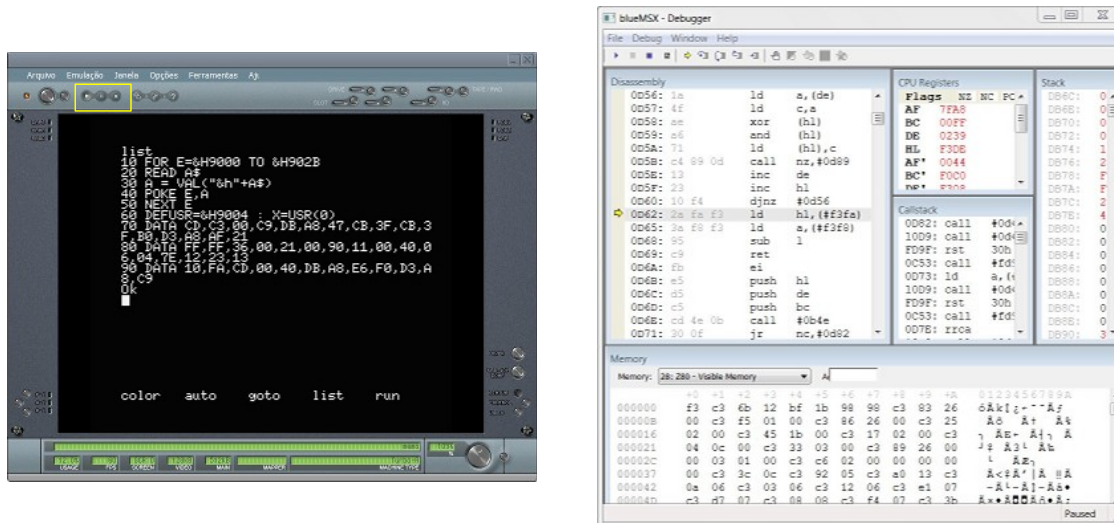


Figura 1. Emulador blueMSX e a ferramenta debugger.

Para que o debugger apresente os dados, é necessário pausar o emulador. O botão de pausa é o botão central dos três botões assinalados pelo retângulo amarelo na figura 1.

O primeiro passo é carregar o programa “exp1.bin” ou “exp1.bas” na memória do MSX, sem executá-lo. Para isso utilize o comando load sem a opção “,r” no primeiro caso, e o comando load no segundo.

O debugger permite a execução passo a passo das instruções em Assembly, bem como criar *breakpoints* ou pontos de parada, que param a execução do MSX no endereço marcado, antes de rodar a próxima instrução. Então, o próximo passo é criar um *breakpoint* no endereço &H9004, que é o início da execução do programa em verde. Mas antes será mostrado como exibir um determinado endereço no debugger.

Preparando o debugger: após dar pausa no emulador, clicar na janela do debugger e teclar “control + G”. Um diálogo é aberto, perguntando qual o endereço a ser exibido na janela do “Disassembly”. Digitar 9004. Para criar um *breakpoint*, é só clicar ao lado esquerdo do endereço desejado, onde uma bolinha vermelha surge identificando o ponto de parada. Para desmarcar, clique sobre a bolinha vermelha. Serão criados mais dois *breakpoints*, além do &H9004. São eles: &H9014 e &H9022.

Rodando o programa: clicar no botão “play” do emulador, o mais à esquerda dos três botões marcados com o retângulo amarelo na figura 1.

Dessa forma, deve-se comandar em Basic, de acordo com o arquivo escolhido:

- exp1.bin → DEFUSR=&H9004 : X=USR(0)
- exp1.bas → RUN

Ao ser executado o programa, o MSX será travado no endereço &H9004, conforme mostra a figura 2.

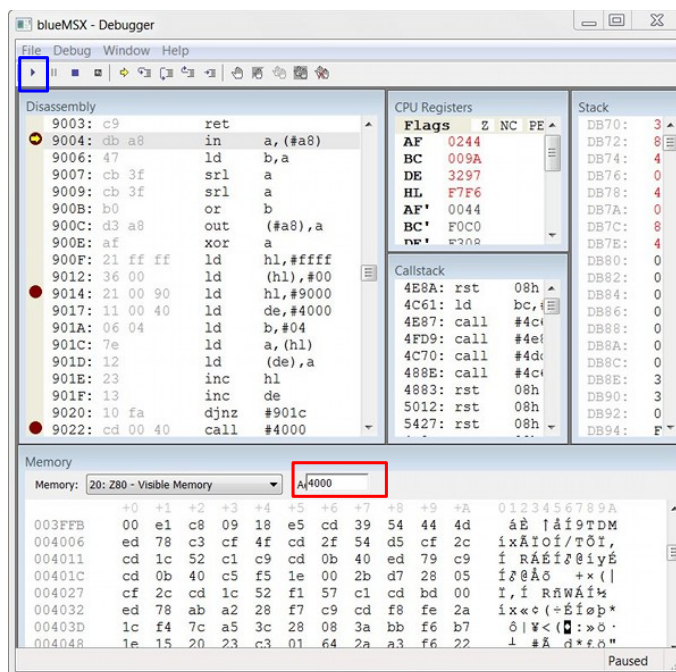


Figura 2. Início da execução do programa no debugger.

Observe o círculo vermelho ao lado do endereço &H9004. Dentro desse círculo, há uma seta amarela, indicando o endereço de execução atual. O programa também aparece na janela “Disassembly” em mnemônicos.

De forma a visualizar a área de memória em &H4000, digitar esse valor na caixa destacada com o retângulo vermelho na figura 2. Essa área de memória ainda está com a ROM.

É possível rodar o debugger passo a passo através da tecla F11 e acompanhar as alterações nos registradores do Z-80. Entretanto, nessa experiência, o debugger irá rodar direto até o próximo *breakpoint*.

O primeiro trecho que será executado mudará a página 1 da ROM para a RAM. Para executar de &H9004 a &H9014, clicar no botão “play” do debugger, destacado com o retângulo azul na figura 2. A página 1 é então trocada, conforme mostra a figura 3.

O trecho de programa entre &H9014 e &H9022 irá copiar o programa em amarelo para a região de memória entre &H4000 e &H4003, conforme mostra a figura 4.

Por último, rodar o programa até o final.

	+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A		+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A
003FFB	00 e1 c8 09 18 e5 cd 39 54 44 4d	003FFB	00 e1 c8 09 18 ff ff ff ff ff ff
004006	ed 78 c3 cf 4f cd 2f 54 d5 cf 2c	004006	ff ff ff ff ff ff ff ff ff ff ff
004011	cd 1c 52 c1 c9 cd 0b 40 ed 79 c9	004011	ff ff ff ff ff ff ff ff ff ff ff
00401C	cd 0b 40 cd f5 1e 00 2b d7 28 05	00401C	ff ff ff ff ff ff ff ff ff ff ff

a) página 1 com a ROM

b) página 1 com a RAM

Figura 3. Alteração da página 1.

	+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A
003FFB	00 e1 c8 09 18 cd c3 00 c9 ff ff
004006	ff ff ff ff ff ff ff ff ff ff ff
004011	ff ff ff ff ff ff ff ff ff ff ff
00401C	ff ff ff ff ff ff ff ff ff ff ff

Figura 4. Cópia do programa na página 1.

### 3- Experiência 2: criando ROM para cartucho

Conforme visto na primeira parte do artigo sobre a memória do MSX, a ROM inserida em um cartucho possui um *header* (cabeçalho) contendo instruções importantes para que o interpretador Basic possa acessar o conteúdo dela. Além disso, as ROMs são de dois tipos: Assembly, começando em &H4000 e Basic, em &H8000.

#### 3.1 - ROM em Basic Adaptado de [1]

O primeiro passo para criar uma ROM do tipo Basic é alterar o endereço inicial do programa em Basic na memória, de forma que se introduza antes dele o *header*. Além disso, o primeiro byte antes do programa em Basic deverá ser 0.

```
POKE &HF676,&H11 : POKE &HF677,&H80 : POKE &H8010,0 : NEW
```

Nota: executar o comando anterior em uma linha.

Assim, TEXT irá apontar para &H8010 e o programa em Basic começar em &H8011. O passo seguinte é criar o *header* da ROM.

```
10 AD = &H8000
20 FOR I = 0 TO 15
30 POKE AD + I, 0
40 NEXT I
50 POKE &H8000,ASC("A")
60 POKE &H8001,ASC("B")
70 POKE &H8008,&H10
80 POKE &H8009,&H80
```

Modifica o  
TEXT

Depois, carregar ou digitar o programa em Basic que irá residir na ROM. Pode utilizar o comando NEW ou carregar um programa, que ele sempre irá começar em &H8011.

Por fim, salvar o endereço de memória de &H8000 a &HBFFF em um arquivo, que será gravado em ROM.

Obs: como o arquivo irá ser salvo em Basic, ele irá gerar 7 bytes de *header*. Nesse caso, deve-se utilizar o programa noheader [2] ou um editor hexadecimal para remover o *header*.

### 3.2 - ROM em Assembly

O programa a seguir é do curso de Assembly da página MarMSX Development [3], e servirá como base para criar o cartucho em Assembly.

End	Assembly	Linha	Etiqueta	Instrução	Comentário
		10		ORG &H4010	; Início do programa na memória
4010	CD C6 00	20		CALL &H6C	; Chama a screen 0 (INITXT)
4013	11 00 00	30		LD DE,0	; Endereço da VRAM
4016	21 21 40	40		LD HL,NOME	; Endereço inicial da frase
4019	01 0A 00	50		LD BC,10	; Comprimento da String
401C	CD 5C 00	60		CALL &H5C	; Chama sub-rotina de escrita na tela
401F	18 FE	70	AQUI:	JR AQUI	; Trava
4021	4F 20 4D 53	80	NOME:	DEFM "O MSX vive"	; Frase
-	58 20 76 69				
402A	76 65				

A linha 20 é necessária para mudar o modo de tela no boot do MSX.

Para criar o arquivo ROM, pode ser usada qualquer ferramenta que crie o arquivo binário acima. Entretanto, deve-se reservar 16 bytes para o *header*, colocando a ID "AB" mais o endereço &H4010 nos dois bytes seguintes. Após o *header*, colocar o programa acima. Por fim, preencher com o valor &HFF o restante do arquivo. Ele deverá ter 16 KB.

O começo do arquivo "omsvive.rom" deverá ser então:

```
0000 41 42 10 40 00 00 00 00 00 00 00 00 00 00 00 | AB.@.....
0010 CD 6C 00 11 00 00 21 21 40 01 0A 00 CD 5C 00 18 | .1....!!@....\..
0020 FE 4F 20 4D 53 58 20 76 69 76 65 FF FF FF FF FF | .O MSX vive.....
```

## 4- Créditos e Referências

O artigo foi escrito por Marcelo Silveira, Engenheiro de Sistemas e Computação, formado pela UERJ.

Escrito em: maio de 2004.

Revisado em: julho de 2017.

e-mail: flamar98@hotmail.com

Homepage: <http://marmsx.msxall.com>

Referências:

[1] - MSX 2 Technical Handbook, ASCII Corporation, 1987.

[2] - Noheader, Tools, MarMSX Development em <http://marmsx.msxall.com>

[3] - Curso de Assembly, Tools, MarMSX Development em <http://marmsx.msxall.com>