

Laboratório de Arquitetura de Computadores

Guião 4

MediaCenter: interface multimédia

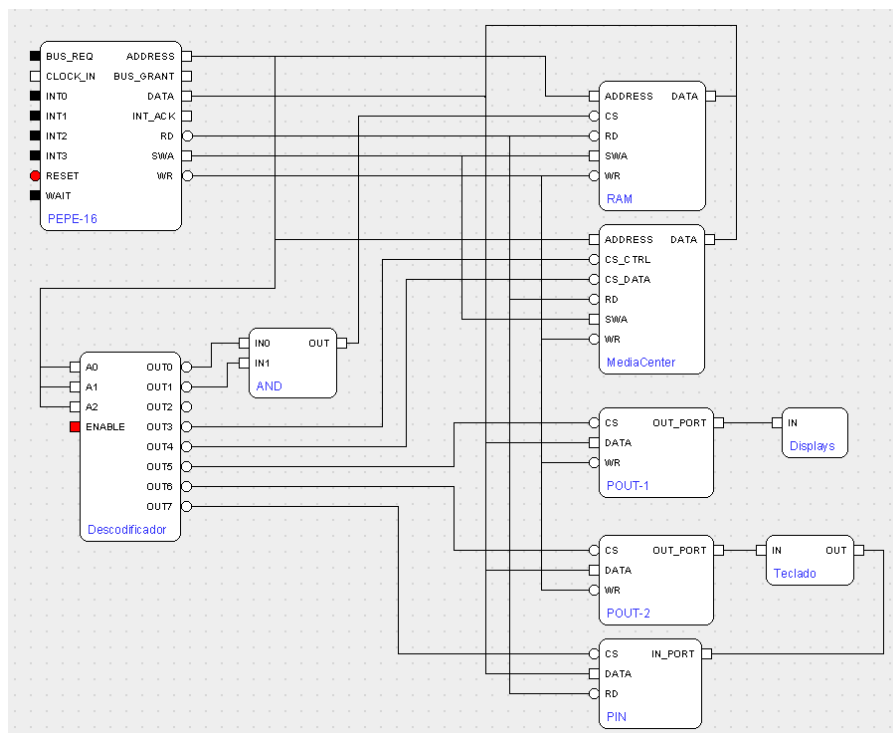
1 Objetivos

Com este guião pretende-se que o leitor continue a praticar programação em linguagem *assembly* do PEPE-16, usando o módulo MediaCenter como exemplo de interação com periféricos complexos (suportando vários comandos).

2 O módulo MediaCenter

2.1 Organização interna

Clique em **Load** (📁) e carregue o circuito contido no ficheiro **lab4.cir**. Este circuito é semelhante ao já usado no Guião 3, mas tem mais um periférico (MediaCenter, com um ecrã gráfico e suporte para áudio e vídeo), permitindo desenhar objetos gráficos, pixel a pixel e construir aplicações multimédia. O Help do simulador contém detalhes adicionais sobre o módulo MediaCenter.



Estes endereços são internos ao MediaCenter. No entanto, o PEPE-16 “vê” a memória interna do MediaCenter a partir do endereço 8000H (determinado pelo sistema de decodificação de endereços deste circuito), pelo que estes endereços, do ponto de vista do PEPE-16, correspondem ao intervalo 8000H a e 8FFFH (endereço do último pixel: 8FFEH).

Tal como a RAM do sistema, a memória do MediaCenter pode ser acedida em *byte* (com **MOVB**) ou em *word* (com **MOV**). No entanto, é mais simples lidar com uma palavra de cada vez, com **MOV**, pois assim lida-se logo com um pixel inteiro.

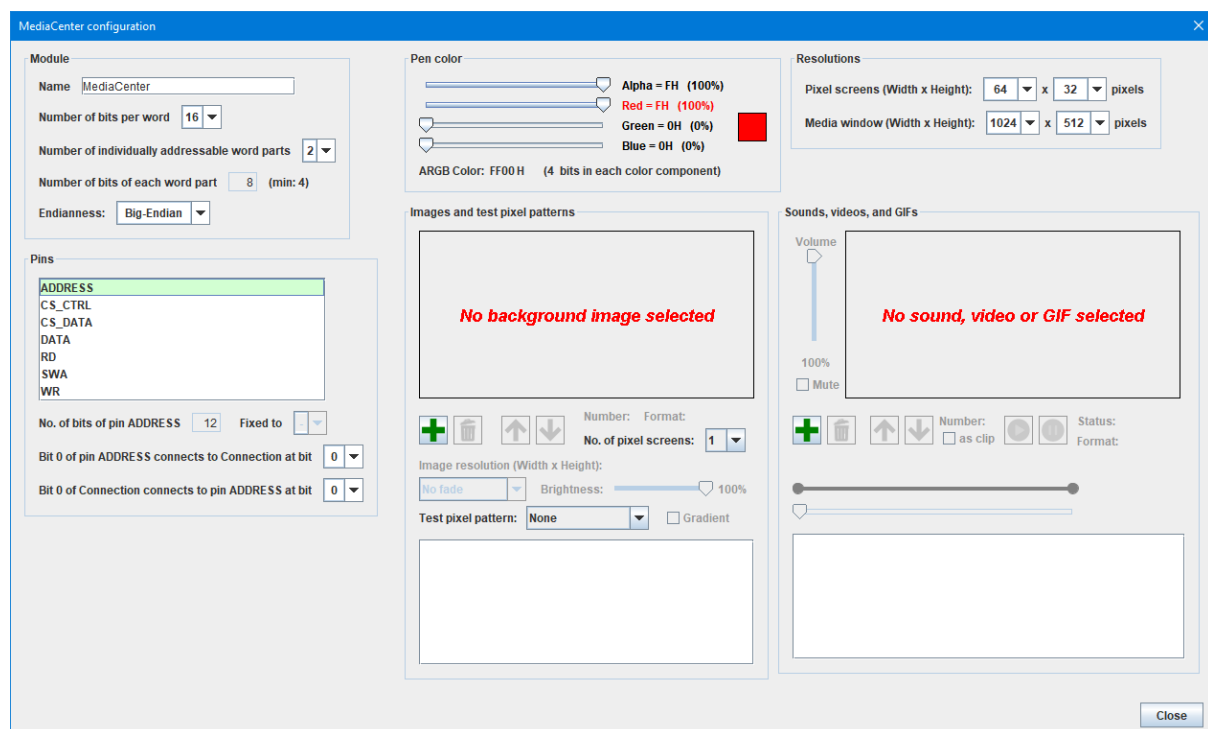
Em vez de escrever diretamente na memória dos pixels, o PEPE-16 pode dar comandos ao MediaCenter, escrevendo em endereços a partir de 6000H, da forma explicada mais adiante. O MediaCenter tem dois *chip selects*, um para a memória de pixels e outra para os comandos.

Desta forma, o mapa de endereços (em que os dispositivos podem ser acedidos pelo PEPE-16) disponível neste circuito é o seguinte:

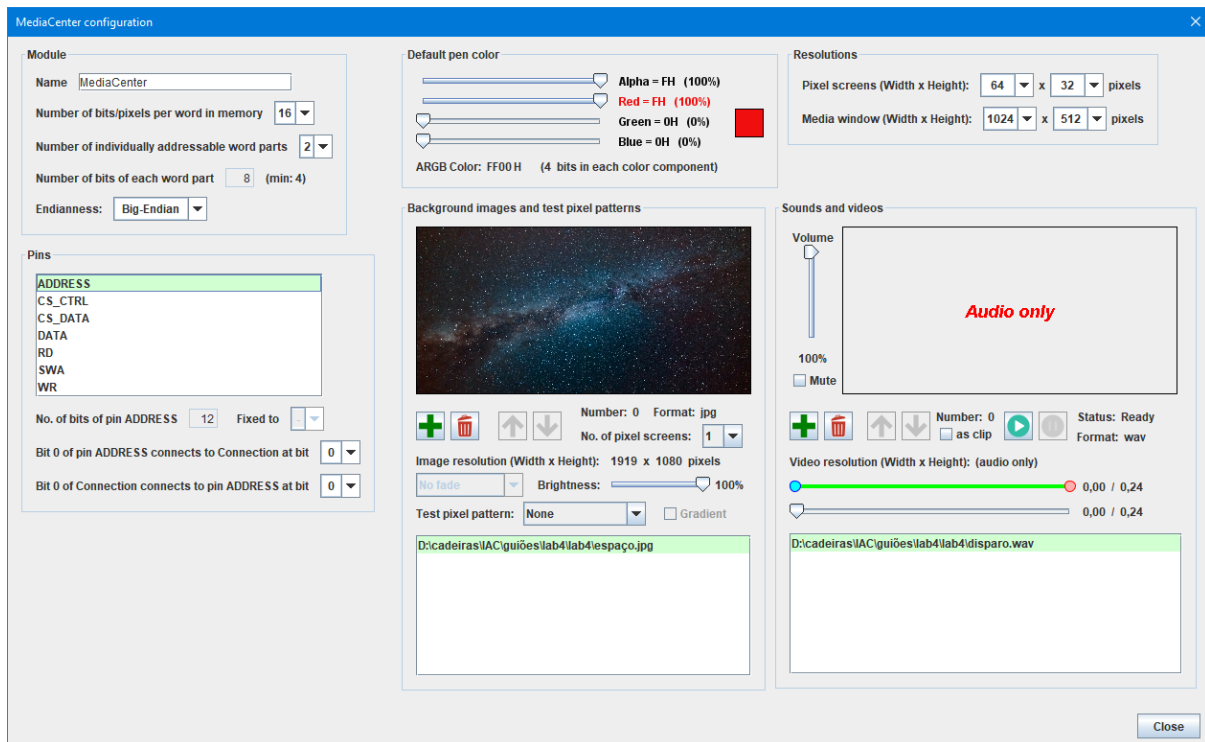
Dispositivo	Endereços
RAM	0000H a 3FFFH
MediaCenter (acesso aos comandos)	6000H a 6069H (ver secção 2.3)
MediaCenter (acesso à sua memória)	8000H a 8FFFH
POUT-1 (periférico de saída de 8 bits)	0A000H
POUT-2 (periférico de saída de 8 bits)	0C000H
PIN (periférico de entrada de 8 bits)	0E000H

2.2 Configuração

A figura seguinte ilustra a interface de configuração do MediaCenter (acedida com clique duplo no MediaCenter em modo “Design”).



Esta é a visão do MediaCenter sem nada definido. No entanto, ele já é fornecido neste guião com uma imagem (**espaço.jpg**) e um som (**disparo.wav**). São meros exemplos. Pode acrescentar outras imagens, sons, vídeos e até GIFs.



Para além da configuração dos pinos, já usada noutros módulos, é possível configurar:

- A cor por omissão da caneta, incluindo opacidade (*Alpha*), com que os pixels serão desenhados no ecrã, em formato ARGB (*Alpha*, *Red*, *Green* e *Blue*);
- A resolução do ecrã de simulação, aquele em que o PEPE-16 escreve os pixels. Na realidade, é possível definir até 16 ecrãs de pixels, sobrepostos, suportando sobreposição de objetos diferentes sem interferência uns nos outros;
- A resolução dos cenários de fundo, imagens/vídeos definidos em ficheiros que, se definidos, aparecem por trás dos pixels desligados (que são 100% transparentes). Esta resolução é geralmente bastante maior que a anterior, mas a imagem de menor dimensão é esticada para se sobrepor à outra;
- Os ficheiros com imagens de cenário (até 64), havendo um painel para a pré-visualização. As extensões de imagem suportadas são as seguintes: ".jpg", ".png", ".bmp" e ".gif" (sem animação, sendo apresentada apenas a primeira imagem do GIF). No entanto, outras poderão ser suportadas, dependendo da plataforma. As imagens podem ser adicionadas (carregando no "+" da categoria "Background images and test pixel patterns"), apagadas ou mudadas de ordem. Para serem guardadas de forma portável no ficheiro do circuito, de modo a funcionarem noutro computador, os ficheiros de imagem devem estar no mesmo diretório (ou subdiretório dele) que o ficheiro do circuito (caso contrário, é guardado o *path* absoluto de cada ficheiro);
- Padrões de teste, que se sobrepõem à imagem para se ter uma ideia de como ficam os objetos em cima das imagens), e até padrões de *fading* ao mudar de cenário ou vídeo;

- Os ficheiros com sons, vídeos e GIFs (até 64). As extensões de áudio e vídeo suportadas são as seguintes: "aif", "aiff", "mp3", "wav", "mp4", "m4a", "m4v" e "gif". Podem ser adicionados (carregando no "+" da categoria "Sounds and videos"), apagados ou mudados de ordem. Para serem guardados de forma portável no ficheiro do circuito, de modo a funcionarem noutro computador, os ficheiros de som/vídeo devem estar no mesmo diretório (ou subdiretório dele) que o ficheiro do circuito (caso contrário, é guardado o *path* absoluto de cada ficheiro). Quando houver pelo menos um ficheiro, definido, há um botão para se poder reproduzir e é possível ver o seu formato e duração, reduzir o seu volume de som e até definir os pontos de início e fim de reprodução.

2.3 Controlo do MediaCenter por comandos

Uma forma de escrever pixels no ecrã do MediaCenter é usando a sua interface de comandos. Para alterar algo no ecrã (a cor de um pixel, selecionar uma linha ou uma coluna, etc.), o PEPE-16 faz uma escrita num endereço a partir do endereço de base dos comandos (6000H, no circuito deste guião), com um valor que corresponde ao argumento do comando. Para ler informação do ecrã (cor de um pixel, que linha está atualmente selecionada, etc.), o PEPE-16 faz uma leitura de um endereço a partir do endereço de base dos comandos (6000H). O valor lido é a informação pretendida.

Portanto, os endereços usados identificam os comandos e os valores (escritos ou lidos) indicam a nova informação, para alterar no ecrã, ou a informação lida do ecrã. Um dado endereço pode ter significados diferentes, consoante seja usado num acesso de escrita ou leitura. Estes acessos pelo PEPE-16 não acedem realmente a uma memória, apenas executam comandos (o endereço é usado apenas como referência ao número do comando).

IMPORTANTE – Porque alguns comandos podem envolver valores que não cabem num byte, todos os comandos escrevem ou leem uma palavra completa (16 bits). Por isso, todos os endereços usados para especificar os comandos são pares. Para usar os comandos deve-se usar instruções de **MOV** (acesso em *word*), somando o endereço de base (6000H) ao comando. O endereço indicado nas tabelas em baixo é simplesmente o dobro do número do comando.

Os comandos de escrita atualmente disponíveis no MediaCenter são os seguintes:

Comando		Descrição	Argumento (valor a escrever)
número	endereço		
0	00H	Apaga todos os pixels do ecrã especificado	0 .. nº ecrãs-1
1	02H	Apaga todos os pixels de todos os ecrãs	---
2	04H	Seleciona o ecrã especificado	0 .. nº ecrãs-1
3	06H	Mostra o ecrã especificado	0 .. nº ecrãs-1
4	08H	Esconde o ecrã especificado	0 .. nº ecrãs-1
5	0AH	Especifica a linha a usar no próximo comando	0 .. n.º linhas-1
6	0CH	Especifica a coluna a usar no próximo comando	0 .. n.º colunas-1
7	0EH	Especifica o nº do pixel a usar no próximo comando	0 .. n.º pixels-1
8	10H	Especifica o auto-increment. Se ligado, os comandos marcados com (*) avançam automaticamente para o(s) pixel(s) seguinte(s)	0 - desliga; 1 - liga (desligado por omissão)
9	12H	(*) Altera a cor do pixel na posição corrente	Palavra com ARGB
10	14H	Especifica a cor da caneta	Palavra com ARGB

11	16H	Altera a cor do pixel indicado para a cor da caneta	0 .. n.º pixels-1
12	18H	Apaga o pixel indicado	0 .. n.º pixels-1
13	1AH	(*) Altera a cor do pixel na posição corrente	0 - apaga; 1 - caneta
14	1CH	(*) Altera a cor de 8 pixels, a partir da maior posição múltipla de 8 que não é maior que a posição corrente	Cada bit: 0 - apaga; 1 - caneta
15	1EH	(*) Altera a cor de 16 pixels, a partir da maior posição múltipla de 16 que não é maior que a posição corrente	Cada bit: 0 - apaga; 1 - caneta
16	20H	Desenha um padrão (um dos 12 que se podem indicar na janela de configuração do MediaCenter)	0 .. 11
17	22H	Igual ao anterior, mas com gradiente na cor	0 .. 11
---	Números de comando 18 .. 31 reservados para expansão futura		---
32	40H	Apaga o cenário de fundo (e elimina o aviso: "No background image selected")	---
33	42H	Seleciona o n.º do cenário de fundo a visualizar	0 .. n.º imagens-1
34	44H	Apaga o cenário frontal	---
35	46H	Seleciona o n.º do cenário frontal a visualizar	0 .. n.º imagens -1
36	48H	Seleciona um som/vídeo/GIF para comandos seguintes	0 .. n.º sons/videos/GIFs-1
37	4AH	Especifica o volume do som (percentagem) do som/vídeo selecionado	0 .. 100
38	4CH	Corta o volume (<i>mute</i>) do som/vídeo especificado	0 .. n.º sons/videos/GIFs-1
39	4EH	Retoma o volume do som/vídeo especificado	0 .. n.º sons/videos-1
40	50H	Corta o volume (<i>mute</i>) de todos os sons/vídeos a reproduzir	---
41	52H	Retoma o volume de todos os sons/vídeos a reproduzir	---
42	54H	Especifica o brilho dos vídeos e do cenário de fundo	0 .. 100
43	56H	Especifica um padrão de transição entre vídeos (um dos 5 que se podem indicar na janela de configuração do MediaCenter)	0 - no fading; 1 - very slow; 2 - slow; 3 - fast; 4 - very fast
44	58H	Especifica o número de vezes que um som/vídeo deve ser reproduzido	0 - reprodução em ciclo até comando para parar
45	5AH	Inicia a reprodução do som/vídeo especificado	0 .. n.º sons/videos-1
46	5CH	Reproduz o som/vídeo especificado em ciclo até ser parado	0 .. n.º sons/videos-1
47	5EH	Pausa a reprodução do som/vídeo especificado	0 .. n.º sons/videos-1
48	60H	Continua a reprodução do som/vídeo especificado	0 .. n.º sons/videos-1
49	62H	Pausa a reprodução de todos os sons/vídeos a reproduzir	---
50	64H	Continua a reprodução de todos os sons/vídeos em pausa	---
51	66H	Termina a reprodução do som/vídeo especificado	0 .. n.º sons/videos-1
52	68H	Termina a reprodução de todos os sons/vídeos a reproduzir	---

Os comandos de leitura atualmente disponíveis no MediaCenter são os seguintes:


Comando		Descrição	Argumento (valor a escrever)
número	endereço		
0	00H	Obtém o n.º de colunas do ecrã (potência de 2)	1 .. 2048
1	02H	Obtém o n.º de linhas do ecrã (potência de 2)	1 .. 1024
2	04H	Obtém o ecrã atualmente selecionado	0 .. n.º ecrãs-1
3	06H	Obtém a visibilidade do ecrã atualmente selecionado	0 - escondido; 1 - visível
4	08H	Obtém a linha da posição corrente	0 .. n.º linhas-1
5	0AH	Obtém a coluna da posição corrente	0 .. n.º colunas-1
6	0CH	Obtém o n.º do pixel da posição corrente	0 .. n.º pixels-1
7	0EH	Obtém o estado do auto-increment. Se ligado, os comandos que alteram a cor de pixels (com *) avançam automaticamente para o(s) pixel(s) seguinte(s)	0 - desligado; 1 - ligado
8	10H	(*) Obtém a cor do pixel na posição corrente	Palavra com ARGB
9	12H	Obtém a cor atual da caneta	Palavra com ARGB
10	14H	(*) Obtém o estado da cor do pixel corrente	0 - apagado; 1 – cor não zero
11	16H	(*) Obtém o estado da cor de 8 pixels, a começar na maior posição múltipla de 8 que não é maior que a posição corrente	Byte, com cada bit: 0 - apagado; 1 – cor não zero
12	18H	(*) Obtém o estado da cor de 16 pixels, a começar na maior posição múltipla de 16 que não é maior que a posição corrente	Palavra, com cada bit: 0 - apagado; 1 – cor não zero
---	Comandos números 13 .. 31 reservados para expansão futura		---
32	40H	Obtém n.º de imagens definidas	0 .. 64
33	42H	Obtém o n.º do cenário de fundo selecionado	-1 .. n.º imagens-1 (-1 se nenhum)
34	44H	Obtém o n.º do cenário frontal selecionado	-1 .. n.º imagens-1 (-1 se nenhum)
35	46H	Obtém n.º de sons/vídeos/GIFs definidos	0 .. 64
36	48H	Obtém o n.º do som/vídeo/GIF selecionado	-1 .. n.º sons/vídeos/GIFs-1 (-1 se nenhum)
37	4AH	Obtém o volume do som (percentagem) do som/vídeo selecionado	0 .. 100
38	4CH	Obtém o estado de <i>mute</i> do som/vídeo especificado	0 - normal; 1 - muted
39	4EH	Obtém o valor do brilho (percentagem)	0 .. 100
40	50H	Obtém o n.º do padrão de transição entre vídeos (um dos 5 que se podem indicar na janela de configuração do MediaCenter)	0- no fading; 1 - very slow; 2 - slow; 3 - fast; 4 - very fast
41	52H	Obtém o estado do som/vídeo/GIF selecionado	0 - Unknown; 1 - Ready; 2 - Paused; 3 - Playing; 4 - Stopped; 5 - Error



42	54H	Obtém o n.º dos sons/vídeos/GIFs a reproduzir	0 .. n.º sons/videos-1
43	56H	Obtém o n.º de vezes que o som/vídeo selecionado será reproduzido	0 - reprodução em ciclo até comando para terminar

2.4 Exemplos de desenho de pixels no MediaCenter (por comandos)

Para exemplificar o uso de comandos com o ecrã, edite e observe o programa contido no ficheiro **lab4-comandos.asm**, que desenha um pixel numa dada linha e coluna e com uma dada cor.

Para apagar um pixel já desenhado, basta desenhá-lo de novo, na mesma linha e coluna, mas com cor 0.

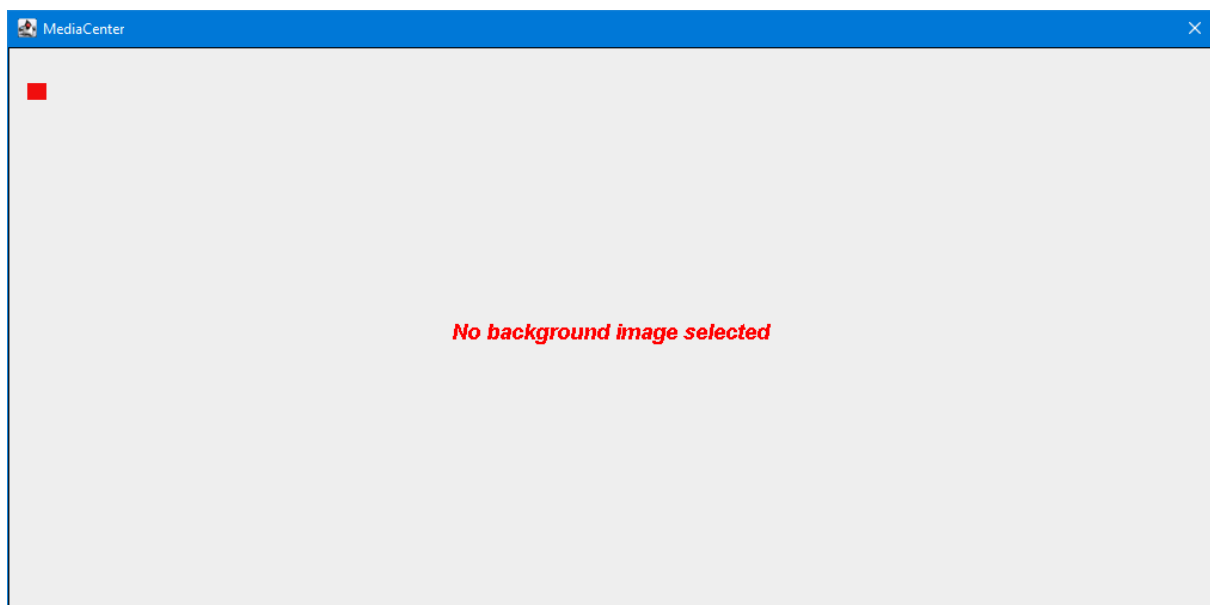
Passe o simulador para “Simulation” e carregue o ficheiro **lab4-comandos.asm** no PEPE-16 (com o botão **Load source**, , ou por *drag & drop*).

Execute o programa passo a passo, com o botão **Step** () , SEM carregar em **Start** () , e verifique que o pixel da linha 2, coluna 1, se liga e que no fim desaparece.

Repare nos três comandos para ligar o pixel (definir linha, coluna e cor) e que basta apenas um comando para o apagar (a linha e coluna já tinham sido especificadas).

Note que tanto as linhas como as colunas começam em 0 (canto superior esquerdo do ecrã).

Termine a execução do programa, carregando no botão **Stop** () do PEPE-16.



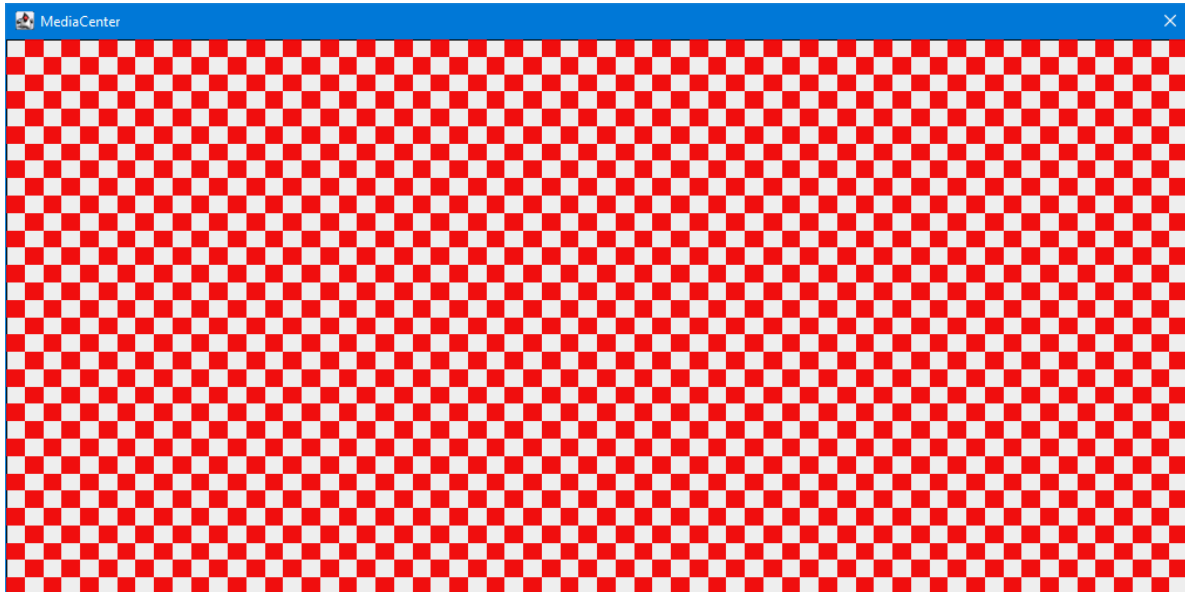
O aviso “**No background image selected**” existe para lembrar que os cenários de fundo têm de ser explicitamente selecionados, com o comando de escrita com o endereço 42H (número 33). No entanto, se quiser eliminar o aviso sem selecionar nenhum cenário, basta executar o comando de escrita com o endereço 40H (número 32). Isto é ilustrado nos dois exemplos seguintes.

Carregue agora o programa contido no ficheiro **lab4-comandos-xadrez.asm**, que desenha um padrão de xadrez no ecrã.

Este programa escreve um pixel numa dada linha e coluna, tal como no programa anterior, mas agora dentro de dois ciclos:

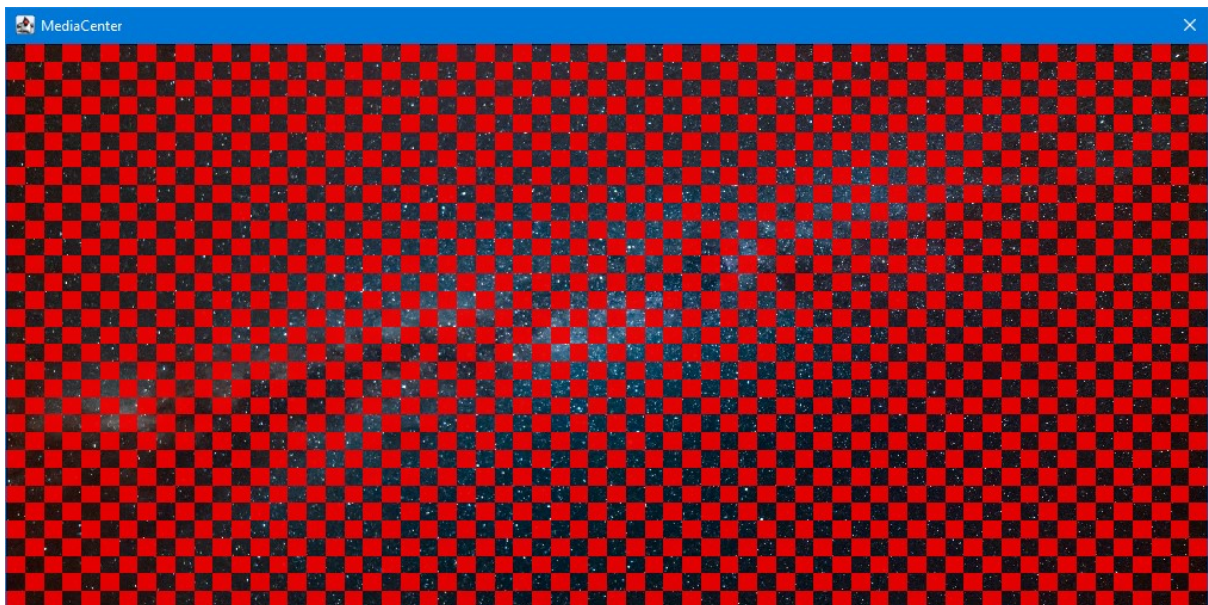
- Um para percorrer todos os pixels de uma dada coluna;
- Outro para percorrer todas as linhas do ecrã.

De cada vez que escreve um pixel ou uma linha troca o valor do pixel (entre desligado e vermelho), e o resultado é um padrão em xadrez, tal como se pode ver na figura seguinte.



Este programa já apaga o aviso “No background image selected”, mas podemos mesmo seleccionar um cenário de fundo, nomeadamente usando a imagem **espaço.jpg**.

Carregue o programa contido no ficheiro **lab4-xadrez-imagem.asm**. Este programa é igual ao anterior, exceto pelo facto de seleccionar um cenário de fundo sobre o qual o padrão de xadrez é desenhado. Execute o programa e verifique que os pixels desligados são na realidade totalmente transparentes (pois a sua componente de opacidade está a 0).



Termine a execução do programa, carregando no botão **Stop** (■) do PEPE-16.

2.5 Exemplos de desenho de pixels no MediaCenter (por acesso à memória)

Em vez de dar comandos ao MediaCenter, também é possível escrever diretamente na sua memória, que neste ecrã de 32 linhas por 64 colunas é constituída por 2048 palavras (uma palavra por cada um dos 32 x 64 pixels). A primeira palavra está localizada no endereço 8000H e a última em 8FFEh, de acordo com o mapa de endereços indicado atrás.

Os programas equivalentes ao **lab4-comandos.asm** e **lab4-comandos-xadrez.asm**, mas com escrita dos pixels na memória, estão contidos nos ficheiros **lab4-memoria.asm** e **lab4-memoria-xadrez.asm**, respetivamente, que também são fornecidos.

NOTA – Se quiser limpar o ecrã (para eliminar o xadrez do exemplo anterior), sem ser por comandos no programa, basta fazer **Stop** (■) e **Restart** (▶) na janela do simulador (não na do PEPE-16).

Pode comparar os dois casos mais simples (**lab4-comandos.asm** e **lab4-memoria.asm**), a partir do label “escreve_pixel”. Com comandos bastam 3 MOVs, mas com acesso à memória são precisas mais algumas instruções.

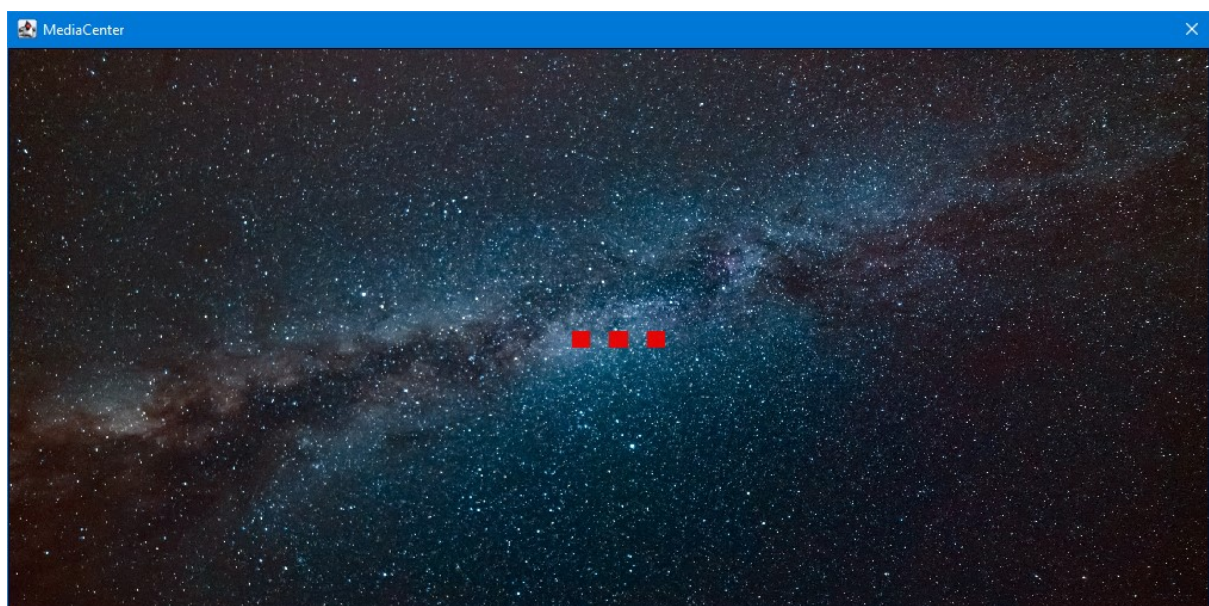
No entanto, mesmo com acessos à memória, os comandos são essenciais para diversas funcionalidades, como apagar o aviso de nenhum cenário de fundo selecionado ou até para limpar o ecrã dos pixels já escritos. O programa do ficheiro **lab4-memoria-xadrez.asm**, embora escreva os pixels na memória, usa comandos para estas funcionalidades. Por tudo isto, é mais prático usar apenas comandos para lidar com o MediaCenter.

2.6 Desenho de bonecos

Desenhar um boneco no MediaCenter é desenhar um conjunto de pixels, que devem estar definidos numa tabela que o programa percorre, em vez de explicitar no programa os comandos para desenhar cada um dos pixels. Assim, mudar o desenho é só mudar a tabela e não o programa.

Toma-se um dos pixels do objeto como referência da sua posição no ecrã (por exemplo, o pixel do canto superior esquerdo) e faz-se um programa que percorra essa tabela e vá desenhando pixel a pixel, em posições sucessivas, relativas à posição do pixel de referência.

Carregue o programa **lab4-boneco.asm** e execute-o. Deverá obter a figura seguinte.



Note a forma como o boneco é definido, numa tabela feita com várias WORDs consecutivas, em que a primeira indica a largura em pixels do boneco e as restantes as cores dos respetivos pixels. Depois há um ciclo no programa que vai desenhando cada pixel da tabela em colunas sucessivamente mais para a direita.

Este é um boneco só com largura (altura de 1 pixel). Para desenhar bonecos com altura A e largura L , basta indicar tanto a largura como altura (2 WORDs), seguidas de A linhas de L WORDs cada. O código para desenhar o boneco tem agora de ter dois ciclos, o interior com L iterações para desenhar cada linha, e o exterior com A iterações para desenhar as várias linhas. Em cada iteração tem de se ir aumentando a coluna (em cada linha) e a linha (sempre que se passa à linha seguinte).

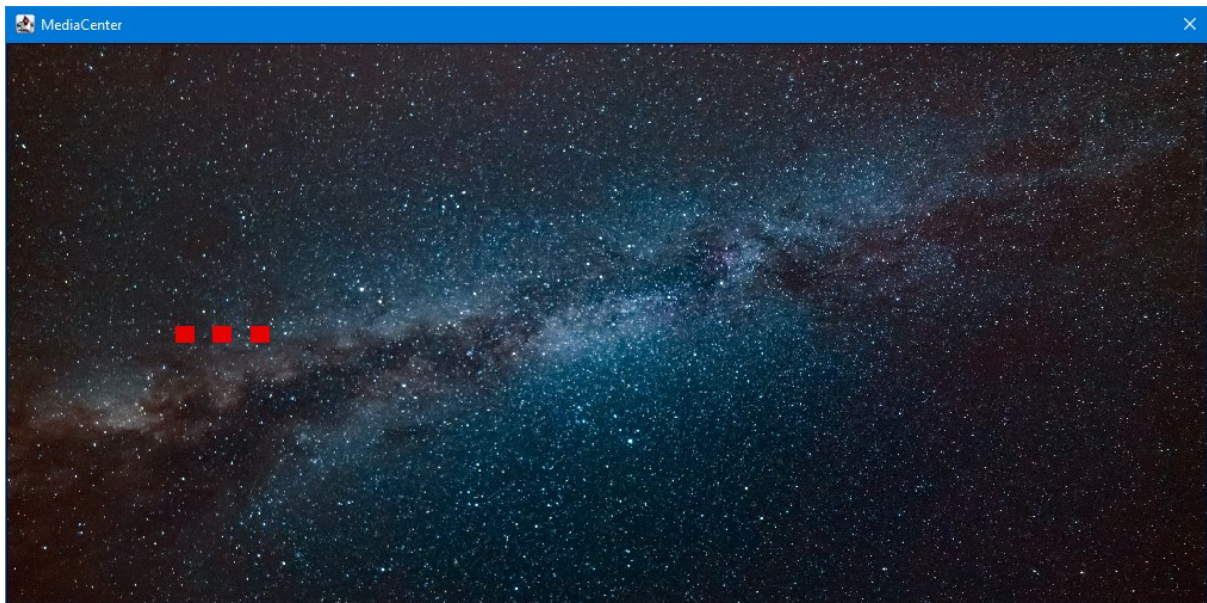
2.7 Movimento de bonecos

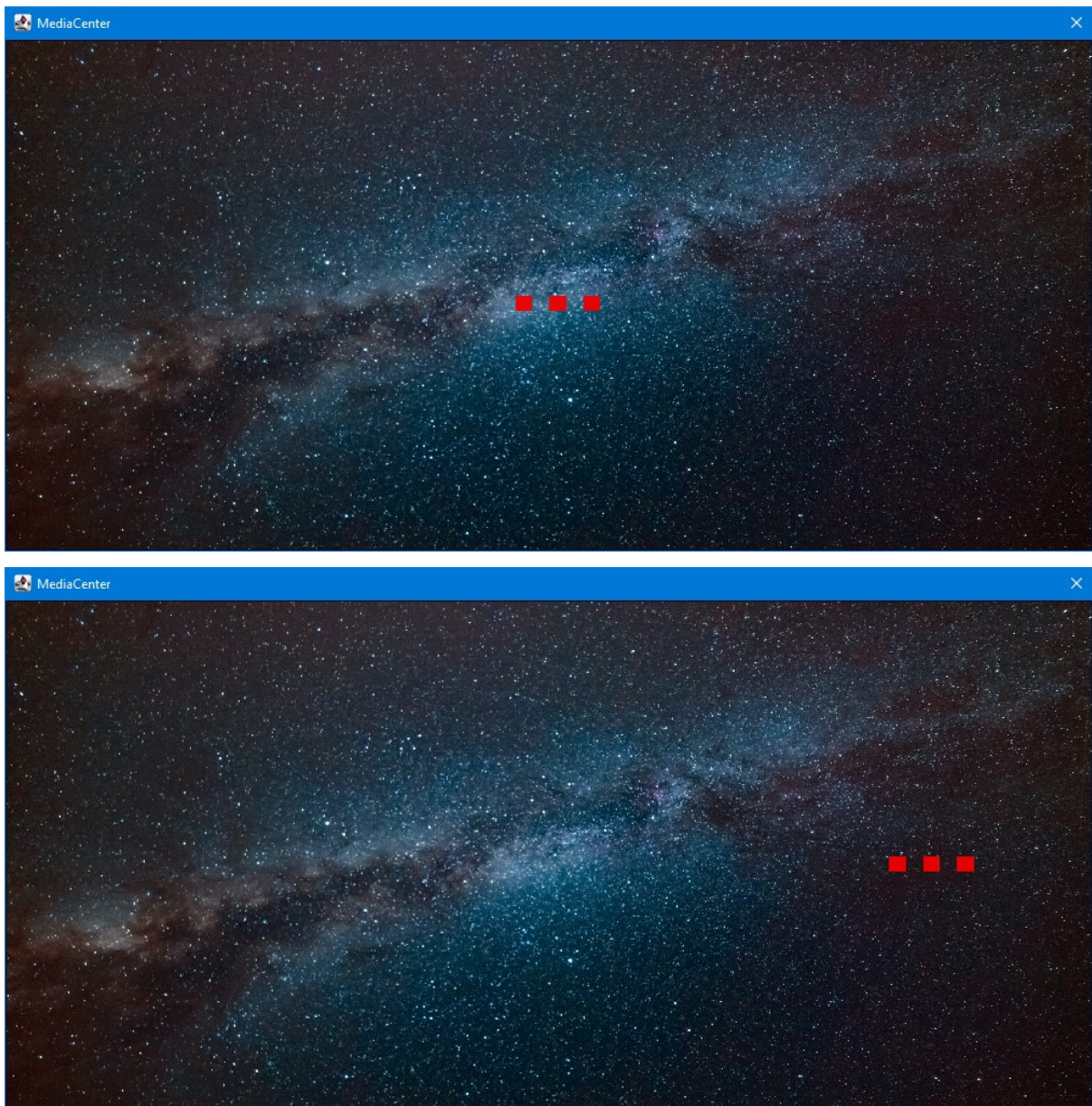
Movimentar um boneco no MediaCenter é apagar o objeto na sua posição corrente, mudar a posição corrente e voltar a desenhá-lo na nova posição. O efeito visual é ele mover-se.

Posição corrente é a posição (linha, coluna) do pixel de referência do objeto (por exemplo, o pixel do canto superior esquerdo). Todos os restantes pixels são desenhados em posições relativas à do pixel de referência.

Carregue o programa **lab4-move-boneco.asm** e execute-o. O boneco deverá movimentar-se de um lado para o outro, tal como ilustrado nas figuras seguintes.

A constante ATRASO é usada para limitar a velocidade do movimento. Poderá ter de a ajustar (maior ou menor), dependendo da velocidade do seu computador.





Este programa desenha o boneco numa dada coluna, faz o ciclo de atraso, apaga o objeto nessa coluna, testa os limites para saber se precisa de inverter o sentido, calcula a nova coluna (somando +1 ou -1 à coluna anterior) e volta atrás para desenharmos de novo o boneco e repetir o ciclo.

2.8 Múltiplos ecrãs de pixels

Na janela de configuração do módulo MediaCenter (accedida com clique duplo no MediaCenter em modo “Design” – ver secção 2.2), é possível especificar o número de ecrãs (pixel screens) que se pretendem (até 16).

Cada ecrã está num plano diferente e a vantagem é que os bonecos que se desenharm num ecrã podem “passar” por cima ou por baixo de bonecos noutros ecrãs, sem interferência uns com os outros.

Depois de definir o número de ecrãs, é possível o programa seleccionar o ecrã onde os próximos pixels irão ser desenhados, bem como esconder ou mostrar um ecrã inteiro. Tal faz-se por meio de comandos (ver secção 2.3).

2.9 Cenários de fundo e frontal

Por trás de todos os ecrãs de pixels está uma janela que pode ter uma imagem, que atua como cenário de fundo. Na frente de todos os ecrãs de pixels está outro, que tipicamente tem uma imagem com fundo transparente e assim permite colocar letreiros, por exemplo, à frente de tudo o resto (cenário frontal).

Na janela de configuração do módulo MediaCenter (acedida com clique duplo no MediaCenter em modo “Design” – ver secção 2.2), é possível definir as imagens que se pretendem (até 64), que são automaticamente esticadas para acompanhar a resolução da janela de simulação do MediaCenter. Depois no programa é possível seleccionar a que se quer como cenário, e ir mudando de acordo com os requisitos do programa, por meio de comandos (ver secção 2.3).

2.10 Sons, vídeos e GIFs

Na janela de configuração do módulo MediaCenter (acedida com clique duplo no MediaCenter em modo “Design” – ver secção 2.2), é também possível definir sons, vídeos e GIFs (até 64). Os vídeos são automaticamente esticados para acompanhar a resolução da janela de simulação do MediaCenter e estão em planos entre o cenário de fundo e os ecrãs de pixels.

Depois no programa é possível seleccionar os sons, vídeos e GIFs e proceder à respetiva reprodução. Existem vários comandos para este efeito (ver secção 2.3). Na janela de configuração é ainda possível obter informação sobre cada som, vídeo e GIF, e até definir os pontos de início e fim do tempo de reprodução.

A escolha “*as clip*” deve ser reservada para efeitos sonoros muito curtos (1 segundo, ou assim). Ocupam mais memória que os sons “não clip”, mas a grande vantagem é a latência (começam a reproduzir mais rapidamente) e podem ser recomeçados antes de acabarem. Em compensação, não se podem terminar a meio, nem fazer pausa, como os outros.

Os ficheiros de áudio e vídeo são carregados quando se passa o simulador para “Simulation”. Embora raro, acontece por vezes o carregamento de um destes ficheiros falhar, caso em que aparece uma mensagem de erro e esse ficheiro não poderá ser reproduzido. Basta passar para “Design” e de novo para “Simulation”.