

Tendo o rato a funcionar corretamente, i.e., o cursor a deslocar-se no ecrã refletindo o deslocamento físico do sensor, pede-se que se represente no ecrã do computador a imagem captada pelo sensor. Do *datasheet* lê-se que o registo `0x0b` do sensor se denomina *pixel-grabber* e que captura um *pixel* por *frame*. Quando este registo é lido e contém um valor válido da intensidade luminosa de um *pixel*¹ o MSB é 1 (bit de validade) e o contador interno é incrementado, fazendo com que na leitura seguinte seja lido o *pixel* com o índice seguinte. Note-se que apenas é capturado um *pixel* por *frame*, portanto não é possível obter um *frame* completo e representá-lo no ecrã. Não obstante como os *frames* consecutivos deverão ser semelhantes a imagem obtida parecerá o resultado de uma só captura. O *datasheet* menciona ainda que qualquer operação de escrita neste registo repõe o contador a zero.

A resolução do sensor do rato é 15 *pixels* por 15 *pixels* totalizando 225 *pixels* que formam a imagem captada pelo sensor. Tendo em conta que só é possível ler a informação relativa a um *pixel* de cada vez, são necessárias 225 leituras para obter a imagem completa.

Para o protocolo de comunicação com o PIC definiu-se que os primeiros 3 *bytes* do pacote são *bytes* de controlo sendo que, do *byte* 0 ao *byte* 2 os seus significados são: a comunicação consiste numa leitura ou numa escrita; o tamanho dos dados comunicados do PIC de volta ao computador; o registo de destino da operação. No caso da operação ser uma escrita, o quarto *byte* corresponde ao conteúdo a escrever no registo de destino. O segundo *byte* de controlo pode especificar ainda uma terceira operação que corresponde à leitura dos *pixels* remanescentes concatenados com as leituras realizadas dos registos do sensor que são os deslocamentos em *xx* e em *yy* do rato desde a última leitura.

Para se obterem todos os valores correspondentes aos *pixels* captados pelo sensor são necessárias pelo menos quatro comunicações com o PIC:

$$\left\lceil \frac{225}{64 - 3} \right\rceil = 4.$$

Para efeitos de maximização do desempenho decidiu-se realizar quatro e apenas quatro comunicações com o PIC para obter todos os dados necessários para mover o cursor e representar a imagem captada pelo sensor durante um ciclo do programa. Na primeira comunicação obtêm-se os primeiros 61 *pixels*, na segunda comunicação obtêm-se do 62º ao 122º *pixels*, na terceira do 123º ao 183º *pixels* e na quarta comunicação obtêm-se os restantes. A figura 1 ilustra a correspondência entre os índices dos *pixels* e as suas posições na grelha cartesiana.

Para desenhar a imagem captada pelo sensor no ecrã com base nos valores da intensidade luminosa associados a cada pixel utilizaram-se funções da biblioteca `gdi32` do Windows. Adaptou-se o código fornecido que utiliza esta API para desenhar um retângulo no ecrã para desenhar uma matriz de quadrados de cores diferentes em que cada quadrado corresponde a um *pixel* captado pelo sensor. O código resultante lista-se abaixo.

```
#define SCALE 15
```

```
HDC hdcScreen = GetDC(NULL);
HDC MemDCExercising = CreateCompatibleDC(hdcScreen);
HBITMAP bm = CreateCompatibleBitmap(hdcScreen, 15 * SCALE, 15 * SCALE);
SelectObject(MemDCExercising, bm);
```

¹O sensor apenas capta intensidade luminosa e não uma cor RGB

14	29	44	59	74	89	104	119	134	149	164	179	194	209	224
13	28	43	58	73	88	103	118	133	148	163	178	193	208	223
12	27	42	57	72	87	102	117	132	147	162	177	192	207	222
11	26	41	56	71	86	101	116	131	146	161	176	191	206	221
10	25	40	55	70	85	100	115	130	145	160	175	190	205	220
9	24	39	54	69	84	99	114	129	144	159	174	189	204	219
8	23	38	53	68	83	98	113	128	143	158	173	188	203	218
7	22	37	52	67	82	97	112	127	142	157	172	187	202	217
6	21	36	51	66	81	96	111	126	141	156	171	186	201	216
5	20	35	50	65	80	95	110	125	140	155	170	185	200	215
4	19	34	49	64	79	94	109	124	139	154	169	184	199	214
3	18	33	48	63	78	93	108	123	138	153	168	183	198	213
2	17	32	47	62	77	92	107	122	137	152	167	182	197	212
1	16	31	46	61	76	91	106	121	136	151	166	181	196	211
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210

Figura 1: Correspondência entre os índices dos *pixels* e as suas posições na grelha cartesiana.

```

HBRUSH hBrush;
HPEN hPen;

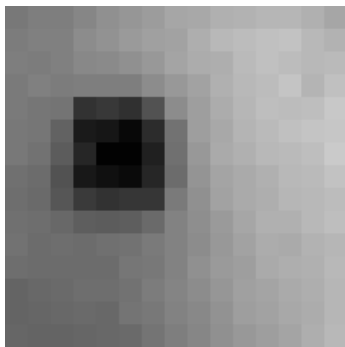
/* image is a vector that contains the values that refer to the luminosity
of the pixels */
for(int i = 0; i < 225; i++)
{
    int x = i / 15;
    int y = 14 - (i % 15);
    int color = image[i];

    hBrush = CreateSolidBrush(RGB(color, color, color));
    SelectObject(MemDCEexercising, hBrush);
    hPen = CreatePen(PS_SOLID, 1, RGB(color, color, color));
    SelectObject(MemDCEexercising, hPen);
    Rectangle(MemDCEexercising, x*SCALE, y*SCALE, (x+1)*SCALE, (y+1)*SCALE);
    DeleteObject(hBrush);
    DeleteObject(hPen);
}

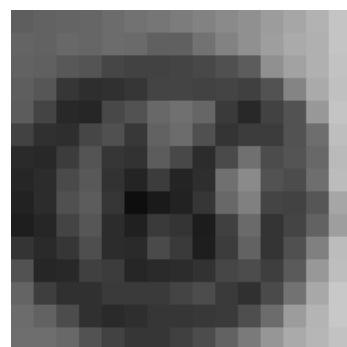
BitBlt(hdcScreen, 0, 0, 15*SCALE, 15*SCALE, MemDCEexercising, 0, 0, SRCCOPY);
DeleteObject(bm);
DeleteDC(MemDCEexercising);

```

A título de exemplo capturaram-se dois fotogramas da imagem capturada pelo sensor do rato e representada no ecrã. Estes resultados ilustram-se na imagem 2.



(a) Amostra 1



(b) Amostra 2

Figura 2: Imagens capturadas pelo sensor do rato e representadas no ecrã.