
Subject: cursor_image

Posted by [slashell](#) on Thu, 26 Jul 2001 14:45:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

IDL help for Cursor_image keyword says:

Specifies the cursor pattern. The value of this keyword must be a 16-line by 16-column bitmap, contained in a 16-element short integer vector. The offset from the upper left pixel to the point that is considered the hot spot can be provided via the CURSOR_XY keyword.

I assumed that each of the elements of the vector could be determined by writing the decimal equivalent of a binary number where the binary number specified which of the pixels in that row should be set and which should not. I further assumed that the least significant bit would be the rightmost bit:

32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4,2,1

Thus, to set only the third pixel from the left on a given row, one would specify the number 8192 for that row.

What I seem to have discovered is that the order of the bits is:

512,1024,2048,4096,8192,16384,32768,1,2,4,8,16,32,64,128,256

Thus, to set only the third pixel from the left on a given row, one would specify the number 2048 for that row.

Is this correct? If so, why????????? This arrangement makes no sense to me.

Looking for sense in a crazy software environment,
Sean

Subject: Re: cursor_image

Posted by [Alex Schuster](#) on Fri, 27 Jul 2001 12:18:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sean La Shell wrote:

- > Specifies the cursor pattern. The value of this keyword must be a
- > 16-line by 16-column bitmap, contained in a 16-element short integer
- > vector. The offset from the upper left pixel to the point that is
- > considered the hot spot can be provided via the CURSOR_XY keyword.
- >
- > I assumed that each of the elements of the vector could be determined
- > by writing the decimal equivalent of a binary number where the binary
- > number specified which of the pixels in that row should be set and

> which should not. I further assumed that the least significant bit
> would be the rightmost bit:
> 32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4,2,1
>
> Thus, to set only the third pixel from the left on a given row, one
> would specify the number 8192 for that row.
>
> What I seem to have discovered is that the order of the bits is:
> 512,1024,2048,4096,8192,16384,32768,1,2,4,8,16,32,64,128,256
>
> Thus, to set only the third pixel from the left on a given row, one
> would specify the number 2048 for that row.
>
> Is this correct? If so, why????????? This arrangement makes no
> sense to me.

The two bytes the 16-bit integer consists of are swapped. This usually comes from data being processed on machines with different endianness, sometimes the hi-value bytes come first, sometimes it's the other way around, depending on the CPU. You can use the `BYTEORDER` procedure or the `SWAP_ENDIAN()` function to change that.

Alex

--

Alex Schuster Wonko@planet-interkom.de
alex@pet.mpin-koeln.mpg.de
