

## Entender código de Gráficos en Turbo Pascal

Escrito por Proyectos\_ - 23/05/2023 09:59

---

Tengo este código:

<http://190.53.102.175/api.7z>

En el directorio apiusr\_CACHE\_mario para Pascal y  
apiusr\_CACHE\_mario\_DJGPP para DJGPP/C/NASM.

Necesito pasar código de Turbo Pascal a DJGPP para activar gráficos.

¿Pone esto la tarjeta en Mode X o en cuál modo?:

```
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
{INIT: Configure VGA to mode ???}
```

```
asm
    mov     ax,13h
    int     10h
end;
{ if not InGraphicsMode then}
    InitVGA;

{   SetWidth (1610 shr 1);
    mov     ax, BYTES_PER_LINE shr 1
    push    ax

    pop     ax

}
```

```
asm
                                {3d4}
{   mov     dx, CRTC_INDEX
    mov     ax, 13h
    out     dx, al
    inc     dx
    in      al,dx
    mov     bl,al
}

    mov     dx, CRTC_INDEX {3d4 0x13 offset or logical width}
    mov     ax, 13h
    out     dx, al
    inc     dx
    mov     al,bl
```

```
    add al,5 {0x28 to 0x2D???}  
mov al,02Dh {0B4h} {028h}  
mov al,02Dh  
{05Ah}  
    out    dx, al  
end;
```

```
asm  
{underline}  
    mov    dx, CRTC_INDEX  
    mov    al, UNDERLINE  
    out    dx, al  
    inc    dx  
    in     al, dx  
    and    al, not 40h  
mov al,0  
    out    dx, al
```

```
{mode control}  
    dec    dx  
    mov    al, MODE_CONTROL  
    out    dx, al  
    inc    dx  
    in     al, dx  
    or     al, 40h  
mov al,0E3h  
    out    dx, al  
end;
```

```
asm  
{memory mode}  
    mov    dx, SC_INDEX  
    mov    al, MEMORY_MODE  
    out    dx, al  
    inc    dx  
    in     al, dx  
    and    al, not 8  
    or     al, 4  
mov al,6  
    out    dx, al  
  
{graphics mode}  
    mov    dx, GC_INDEX {0x3CE}  
    mov    al, GRAPHICS_MODE  
    out    dx, al
```

```
    inc    dx
    in     al, dx
    and    al, not 10h
mov al,40h
    out    dx, al
    dec    dx

{miscellaneous}
    mov    al, MISCELLANEOUS
    out    dx, al
    inc    dx
    in     al, dx
    and    al, not 2
MOV AL,0E1h
    out    dx, al
```

```
{vertical retrace end, turn off write protect}
    mov dx,03D4h
    mov ax,02C11h
    mov ax,00E11h
    out dx,ax
```

```
{vertical total}
    mov ax,00D06h
    mov ax,0BF06h
    out dx,ax
```

```
{overflow register}
    mov ax,03E07h
    mov ax,01F07h
    out dx,ax
```

```
{vertical retrace start}
    mov ax,0EA10h
    mov ax,09C10h
    out dx,ax

{vertical display enable end}
    mov ax,0DF12h
    mov ax,08E12h
    out dx,ax

{start vertical blanking}
    mov ax,0E715h
    mov ax,09615h
    out dx,ax

{end vertical blanking}
    mov ax,00616h
    mov ax,0B916h
    out dx,ax

{vertical retrace end AND wr.prot}
    mov ax,0AC11h
    mov ax,08E11h
    out dx,ax

end;
```

```
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
{END: Configure VGA to mode ???}
```

=====