

```

{ ****
*
*           V O N O F F P . P A S
*
**-----**
*   Task           : Demonstrates video display enable and disable on
*                   EGA and VGA cards.
*
**-----**
*   Author          : Michael Tischer
*   Developed on    : 08/05/90
*   Last update     : 02/18/92
*
*****}

```

```
program VOnOffP;
```

```
uses DOS, CRT;                                { Add CRT and DOS units }
```

```
{-- Constants -----}
```

```

const EV_STATC      = $3DA;      { EGA/VGA color status register }
      EV_STATM      = $3BA;      { EGA/VGA mono status register }
      EV_ATTR       = $3C0;      { EGA/VGA attribute controller }

```

```

procedure CLI; inline( $FA );      { Disable interrupts }
procedure STI; inline( $FB );      { Enable interrupts }

```

```

{ ****
*   ScrOff : Disables the EGA/VGA screen.
*
**-----**
*   Input   : None
*
*****}

```

```
procedure ScrOff;
```

```

var dummy : BYTE;           { Dummy variable for register contents }

begin
    cli;                     { Disable interrupts }
    dummy := port[EV_STATC];  { Reset color status reg }
    dummy := port[EV_STATM];  { Reset mono status reg }
    port[EV_ATTR] := $00;     { Mask bit 5 from access }
                                { to CRT controller }
    sti;                     { Enable interrupts }
end;

{ *****
*   ScrOn : Enables the EGA/VGA screen.   *
*-----*
*   Input   : None                       *
*-----*
*-----*
}

procedure ScrOn;

var dummy : BYTE;           { Dummy variable for register contents }

begin
    cli;                     { Disable interrupts }
    dummy := port[EV_STATC];  { Reset color status reg }
    dummy := port[EV_STATM];  { Reset mono status reg }
    port[EV_ATTR] := $20;     { Set bit 5 for access }
                                { to CRT controller }
    sti;                     { Enable interrupts }
end;

{ *****
*   IsEgaVga : Determines whether an EGA or a VGA card is installed.   *
*-----*
}

```

```

**-----**
*   Input   : None                                     *
*   Output  : TRUE if EGA or VGA card, otherwise FALSE *
*****}

```

```
function IsEgaVga : boolean;
```

```
var Regs : Registers;           { Processor registers for interrupt call }
```

```
begin
```

```
  Regs.AX := $1a00;              { Function 1AH applies to VGA only }
```

```
  Intr( $10, Regs );
```

```
  if ( Regs.AL = $1a ) then      { Is the function available? }
```

```
    IsEgaVga := TRUE
```

```
  else
```

```
    begin
```

```
      Regs.ah := $12;            { Call function 12H, }
```

```
      Regs.bl := $10;            { sub-function 10H }
```

```
      intr($10, Regs);           { Call video BIOS }
```

```
      IsEgaVga := ( Regs.bl <> $10 );
```

```
    end;
```

```
end;
```

```

{*****}
{**               M A I N   P R O G R A M               **}
{*****}

```

```
var ch : char;                  { Get a key }
```

```
begin
```

```
  ClrScr;
```

```
  writeln( 'VONOFFP - (c) 1992 by Michael Tischer'#13#10 );
```

```

if IsEgaVga then
    { EGA or VGA card? }
    begin
        { Yes --> Do it }
        writeln( 'ATTENTION: Screen will go black in five seconds. ' );
        writeln( 'Press any key to enable screen again. ' );
        Delay( 5000 );
        { Wait five seconds }
        while KeyPressed do
            { Purge all keys from the keyboard buffer }
            ch := ReadKey;
            ScrOff;
            { Screen off }
            ch := ReadKey;
            { Wait for a key }
            ScrOn;
            { Screen on }
        writeln ( #13#10#10#10 + 'End program' );
    end
else
    { No --> No EGA or VGA }
    writeln( 'Warning: No EGA or VGA card found' );
end.

```