

```

***** C T T S . P A S *****
{
  *-----*
  Task      : Interface functions for accessing Creative Labs
              "CTTS.DRV" driver (Text to Speech)
  *-----*
  Author     : Michael Tischer / Bruno Jennrich
  Developed on : 03/20/1994
  Last update : 10/8/1994
  *****
}

```

```

Unit CTTS;
Interface

```

```
Const

```

```

CTTS_DRIVERVERSION  = 0;
CTTS_SETSETTINGS    = 1;
CTTS_INI            = 2;
CTTS_TERMINAT       = 3;
CTTS_SETSPEECHPARA  = 4;
CTTS_SA             = 5;

```

```
Function ctts_GetDrvVer( lpEntry : Longint ) : Word;
```

```
Function ctts_GetEnvSettings( lpEntry : Longint ) : Word;
```

```
Function ctts_Init( lpEntry : Longint ) : Word;
```

```

Procedure ctts_SetSpeechParam( lpEntry : Longint;
                               bGender,
                               bTone,
                               bVolume,
                               bPitch,
                               bSpeed : Byte );

```

```
Procedure ctts_Terminate( lpEntry : Longint );
```

```
Function ctts_Say( lpEntry : Longint; SpeechStr : String) : Word;
```

```
Implementation
```

```
Uses DOS;
```

```

*****
{ ctts_GetDrvVer : Get driver version number
  *-----*
  { Input : lpEntry - Address of driver entry point
  { Output : Version number - HiByte / LoByte
              $FFFF - No driver entry point
  *****
}

```

```
Function ctts_GetDrvVer( lpEntry : Longint ) : Word;
```

```
var cret : Word;
```

```
Begin
```

```

  if lpEntry <> 0 then Begin
    asm
      mov BX, CTTS_DRIVERVERSION
      call lpEntry
      mov cret, AX
    End;
    ctts_GetDrvVer := cret;
  End else ctts_GetDrvVer := $ffff;
End;

```

```

*****
{ ctts_GetEnvSettings : Supply driver with environment variable
  *-----*
  { Input : lpEntry - Address of driver entry point
  { Output : 0 - Everything is OK
              1 - BLASTER string = NULL
              2 - BLASTER string defective
              $FFFF - No driver entry point
  *****
}

```

```
Function ctts_GetEnvSettings( lpEntry : Longint ) : Word;
```

```

var cax      : Word;
    lpBlaster : String;

```

```
Begin
```

```

  if lpEntry <> 0 then
    Begin
      lpBlaster := GetEnv( 'BLASTER' );
      asm
        les di, dword ptr lpblaster { Address of Blaster string }
        mov BX, CTTS_SETSETTINGS
        call lpEntry

```

```

        mov cax, AX
    end;
    ctts_GetEnvSettings := cax;
End
else ctts_GetEnvSettings := $FFFF;
End;

{*****}
{ ctts_Init : Initialize driver }
{*****}
{-----*}
{ Input : lpEntry - Address of driver entry point }
{ Output : 0      - Everything is OK }
{          <> 0    - defective initialization }
{          $FFFF  - No driver entry point }
{-----*}
{ Info : - cttsInit cannot be called until after cttsSetEnvSettings }
{         has been called. cttsInit attempts to initialize the }
{         SBTALKER driver - which must be started beforehand. }
{*****}
Function ctts_Init( lpEntry : Longint ) : Word;

var cret : Word;

Begin
    if lpEntry <> 0 then
    Begin
        asm
            mov BX, CTTS_INI
            call lpEntry
            mov cret, AX
        end;
        Ctts_init := cret;
    End
    else ctts_Init := $FFFF;
End;

{*****}
{ ctts_SetSpeechParam : Set speech parameters }
{*****}
{-----*}
{ Input : lpEntry - Address of driver entry point }
{         bGender - Gender ( 0 = masculine, 1 = feminine ) }
{         bTone   - 0 = Bass, 1 = Treble }
{         bVolume - Volume ( 0 - 9 ) }
{         bPitch  - Pitch ( 0 - 9 ) }
{         bSpeed  - Speed ( 0 - 9 ) }
{-----*}
Procedure ctts_SetSpeechParam( lpEntry : Longint;
                               bGender,
                               bTone,
                               bVolume,
                               bPitch,
                               bSpeed : Byte );

Begin
    if lpEntry <> 0 then
    asm
        mov BX, CTTS_SETSPEECHPARA
        mov AL, bGender
        mov AH, bTone
        mov DL, bVolume
        mov DH, bPitch
        mov CL, bSpeed
        call lpEntry
    end;
End;

{*****}
{ ctts_Terminate : Uninstall driver }
{*****}
{-----*}
{ Input : lpEntry - Address of driver entry point }
{-----*}
{ Info : - Call this function before calling }
{         'sb_UnloadDriver' }
{-----*}
Procedure ctts_Terminate( lpEntry : Longint );

Begin
    if lpEntry <> 0 then
    asm
        mov BX, CTTS_TERMINAT
        call lpEntry
    end;
End;

{*****}
{ ctts_Say : Recite text }
{-----*}

```

```

{ Input : lpEntry - Address of driver entry point }
{ SpeechStr - Text to be output (cannot contain any }
{ umlauts ) }
{ Output: 0 - Everything is OK }
{ 1 - Output string blank or NULL }
{ 2 - Output string too long }
{ $FFFF - No driver entry point }
{ ***** }
Function ctts_Say( lpEntry : Longint; SpeechStr : String) : Word;

var cret : Word;
    SpeechPtr : longint;

Begin
    SpeechStr := SpeechStr + #0;
    SpeechPtr := longint(@SpeechStr)+1;
    if lpEntry <> 0 then
    Begin
        asm
            les di, SpeechPtr
            mov BX, CTTS_SA
            call lpEntry
            mov cret, AX
        end;
        ctts_say := cret;
    End;
    ctts_say := $FFFF;
End;

End.

```