

Listing: CTTS.C

```

/*****
/*          C T T S . C                      */
/**-----*/
/* Task      : Interface functions for accessing Creative Labs */
/*          driver "CTTS.DRV" (Text to Speech)                */
/**-----*/
/* Author       : Michael Tischer / Bruno Jennrich           */
/* Developed on  : 03/20/1994                                  */
/* Last update   : 03/25/1994                                  */
/**-----*/
/* COMPILER      : Borland C++ 3.1, Microsoft Visual C++ 1.5 */
/*****
#ifndef __CTTS_C                      /* CTTS.C can be #Included */
#define __CTTS_C

/*- Add include files -----*/

#include <dos.h>
#include <stdlib.h>

#include "ctts.h"

/*****
/* ctts_GetDrvVer : Get driver version number                */
/**-----*/
/* Input : lpEntry - Address of driver entry point           */
/* Output : Version number - HiByte / LoByte                  */
/*          0xFFFF - no driver entry point                   */
/*****
WORD ctts_GetDrvVer( LPVOID lpEntry )
{ WORD cret;
```

```

if( lpEntry )
{
    _asm mov BX, CTTS_DRIVERVERSION
    _asm call lpEntry
    _asm mov cret, AX
    return cret;
}
return 0xFFFF;
}

/*****
/* 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 */
/*          0xFFFF - no driver entry point */
*****/
WORD ctts_GetEnvSettings( LPVOID lpEntry )
{
    WORD cax, ces, cdi;
    LPCHAR lpBlaster;
    if( lpEntry )
    {
        lpBlaster = getenv( "BLASTER" );
        ces = FP_SEG( lpBlaster ); /* Pass */
        cdi = FP_OFF( lpBlaster ); /* Address of "Blaster" string */
        _asm mov ES, ces
        _asm mov DI, cdi
        _asm mov BX, CTTS_SETSETTINGS
        _asm call lpEntry
        _asm mov cax, AX
        return cax;
    }
}

```

```

    }
    return 0xFFFF;
}

/*****
/* ctts_Init : Initialize driver */
/**-----**/
/* Input : lpEntry - Address of driver entry point */
/* Output : 0 - everything is OK */
/*          <> 0 - faulty initialization */
/*          0xFFFF - no driver entry point */
/**-----**/
/* Info : - Call 'cttsInit' after */
/*          'cttsSetEnvSettings'. cttsInit tries to initialize */
/*          the SBTALKER-Driver - which must be started in advance. */
/*****
WORD ctts_Init( LPVOID lpEntry )
{ WORD cret;
  if( lpEntry )
  {
    _asm mov BX, CTTS_INIT
    _asm call lpEntry
    _asm mov cret, AX
    return cret;
  }
  return 0xFFFF;
}

/*****
/* 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 )                                  */
/*****/
VOID ctts_SetSpeechParam( LPVOID lpEntry,
                          BYTE bGender,
                          BYTE bTone,
                          BYTE bVolume,
                          BYTE bPitch,
                          BYTE bSpeed )
{
    if( lpEntry )
    {
        _asm mov BX, CTTS_SETSPEECHPARAM
        _asm mov AL, bGender
        _asm mov AH, bTone
        _asm mov DL, bVolume
        _asm mov DH, bPitch
        _asm mov CL, bSpeed
        _asm call lpEntry
    }
}

/*****/
/* ctts_Terminate : Uninstall driver                                  */
/**-----**/
/* Input : lpEntry - Address of driver entry point                    */
/**-----**/
/* Info : - This function must be called prior to                    */
/*          'sb_UnloadDriver'                                          */

```

```

/*****
VOID ctts_Terminate( LPVOID lpEntry )
{
    if( lpEntry )
    {
        _asm mov BX, CTTS_TERMINATE
        _asm call lpEntry
    }
}

/*****
/* ctts_Say : Say text */
/**-----**/
/* Input : lpEntry - Address of driver entry point */
/*          lpString - Text to be output (cannot contain any */
/*                  umlauts ) */
/* Output: 0 - everything is OK */
/*          1 - Output string blank or NULL */
/*          2 - Output string too long */
/*          0xFFFF - no driver entry point */
/*****
WORD ctts_Say( LPVOID lpEntry, LPCHAR lpString)
{ WORD cret, ces, cdi;
  if( lpEntry )
  {
      ces = FP_SEG( lpString );
      cdi = FP_OFF( lpString );
      _asm mov ES, ces
      _asm mov DI, cdi
      _asm mov BX, CTTS_SAY;
      _asm call lpEntry
      _asm mov cret, AX

```

```
        return cret;  
    }  
    return 0xFFFF;  
}  
#endif
```