

```

DSPUTIL.H
/*****
/*          D S P U T I L   . H          */
/**-----**/
/* Task          : Header file for DSPUTIL.C          */
/**-----**/
/* Author         : Michael Tischer / Bruno Jennrich          */
/* Developed on   : 03/20/1994          */
/* Last update    : 04/05/1995          */
/*****
#ifndef __INC_DSP_UTIL_H
#define __INC_DSP_UTIL_H

#include "types.h"
#include "sbutil.h"

typedef struct tagDSPRECPPLAY
{
    INT iStereo;
    INT iBits;
    WORD uFrequency;
} DSPRECPPLAY;
typedef DSPRECPPLAY *PDSPRECPPLAY;

#define DSP_ERRRESET          1          /* Error codes */
#define DSP_ERRVERSION        2
#define DSP_ERRFRQ            3
#define DSP_ERR4DACFRQ        4
#define DSP_ERR4ADCFRQ        5
#define DSP_ERRSPEAKER        6
#define DSP_ERRTRANSSIZE      7
#define DSP_ERRILLSIZE        8

```

```

#define DSP_ERR8DAC      9
#define DSP_ERR8ADC     10
#define DSP_ERRPLAY     11
#define DSP_ERRRECORD   12

#define FRQ1_MONO_ADC    13000U    /* Highest AD sampling frequencies */
#define FRQ2p_MONO_ADC  13000U    /* DSP Version 2.01+ */
#define FRQ2p_HIMONO_ADC 15000U
#define FRQ3_MONO_ADC    23000U
#define FRQ3_HIMONO_ADC  44100U
#define FRQ3_STEREO_ADC  22050U
#define FRQ4_ADC         44100U

#define FRQ1_MONO_DAC    23000U    /* Highest DA sampling frequencies */
#define FRQ2p_MONO_DAC  23000U    /* DSP Version 2.01+ */
#define FRQ2p_HIMONO_DAC 44100U
#define FRQ3_MONO_DAC    23000U
#define FRQ3_HIMONO_DAC  44100U
#define FRQ3_STEREO_DAC  22050U
#define FRQ4_DAC         44100U

#define DSP_1XX 0x100
#define DSP_200 0x200
#define DSP_201 0x210
#define DSP_3XX 0x300
#define DSP_4XX 0x400

/* Version numbers */

/* DSP Register Offsets */
#define DSP_WRITESTATUS 0x0C /* Bit 7 indicates whether write operations are
allowed */
#define DSP_WRITECMDATA 0x0C /* Write data */

```

```

#define DSP_READSTATUS      0x0E  /* Bit 7 indicates whether read operations
are allowed */
#define DSP_IRQACK          0x0E      /* Interrupt acknowledge */
#define DSP_READDATA        0x0A      /* Read data */

#define DSP_RESET           0x06      /* Reset Port */

#define DSP_GETVERSION      0xE1      /* DSP commands */
#define DSP_8DAC            0x10
#define DSP_8DMADAC         0x14
#define DSP_8DMAAUTODAC     0x1C

#define DSP_8ADC            0x20
#define DSP_8DMAADC         0x24
#define DSP_8DMAAUTOADC     0x2C

#define DSP_SETTIMECONSTANT 0x40
#define DSP_SETTRANSFER SIZE 0x48

#define DSP_DMAPAUSE        0xD0
#define DSP_DMACONTINUE     0xD4

#define DSP2p_8DMAHIAUTODAC 0x90
#define DSP2p_8DMAHIDAC     0x91

#define DSP2p_8DMAHIAUTOADC 0x98
#define DSP2p_8DMAHIADC     0x99

#define DSP_SPEAKERON       0xD1
#define DSP_SPEAKEROFF      0xD3
#define DSP2p_SPEAKERSTATUS 0xD8
#define DSP4_EXIT16DMA      0xD9

```

```

#define DSP4_EXIT8DMA          0xDA
#define DSP2p_EXITAUTOINIT    0xDA

#define DSP3_MONOADC           0xA0          /* DSP3 */
#define DSP3_STEREOADC         0xA8

#define DSP4_DACFRQ            0x41
#define DSP4_ADCFRQ            0x42

#define DSP4_CMDADC            0x08          /* Command bit for recording */
#define DSP4_CMDAUTOINIT       0x04          /* Bit for Autoinit mode */
#define DSP4_CMDFIFO           0x02
#define DSP4_CMDDAC            0x00
#define DSP4_CMD8DMA           0xC0          /* Hi-Nibble for 8 and 16 bit */
#define DSP4_CMD16DMA          0xB0          /* Commands */

#define DSP4_MODESTEREO        0x20
#define DSP4_MODEMONO          0x00
#define DSP4_MODESIGNED        0x10
#define DSP4_MODEUNSIGNED      0x00

/*- Prototypes -----*/
WORD dsp_SetBase( PSBBASE pSBBASE, WORD iReset );
WORD dsp_Write( WORD iVal );
WORD dsp_Read( PBYTE pVal );
WORD dsp_Reset( VOID );
WORD dsp_GetVersion( PSBBASE pSBBASE);
#ifdef DSP_VERSIONONLY
WORD dsp_AdjustFrq( PWORD pFrq, INT iADC, PINT pStereo );
WORD dsp4_DACFrq( WORD uFrq );
WORD dsp4_ADCFrq( WORD uFrq );
WORD dsp_SetFrq( PWORD pFrq );

```

```

WORD dsp_CanStereo( void );
WORD dsp_IsHIMONOADCFrq( WORD uFrq );
WORD dsp_IsHIMONODACFrq( WORD uFrq );
int dsp_MaxBits( VOID );
WORD dsp_SetSpeaker( WORD iState );
WORD dsp_SetTransferSize( WORD uSize );
VOID _interrupt _FP dsp_IrqHandler( );
VOID dsp_RestoreIrqHandler( VOID );
VOID dsp_InitIrqHandler( VOID );
VOID dsp_SetUserIRQ( VOID ( _FP *lpFunc)( LONG ) );
VOID dsp4_SetUserIRQ( VOID ( _FP *lpFunc)( BYTE ) );
WORD dsp_WaitForNextIRQ( VOID ( _FP *lpDoSomething)( LONG ),
                        LONG lPar );
VOID dsp_InitWaitForIRQ( VOID );
WORD dsp_8DAC( BYTE bVal );
WORD dsp_8ADC( PBYTE pVal );
WORD dsp_PLAY( PBYTE pBuffer, WORD uSize, WORD uDelay );
WORD dsp_RECORD( PBYTE pBuffer, WORD uSize, WORD uDelay );
VOID dsp_InitBuffers( VOID );
INT dsp_FileOpen( PCHAR pFile, INT iADC, PINT pHandle );
INT dsp_FileClose( INT iHandle );
INT dsp_ReadHeader( INT iHandle, PDSPRECPLAY pDRP );
INT dsp_WriteHeader( INT iHandle, PDSPRECPLAY pDRP );
INT dsp_ReadBuffer( INT iHandle, LPBYTE lpBuffer, INT iHalfSize );
INT dsp_WriteBuffer( INT iHandle, LPBYTE lpBuffer, INT iHalfSize );
VOID dsp_DoRECPPLAY( INT iHandle,
                    INT iADC,
                    INT iSource,
                    PDSPRECPLAY pDRP,
                    INT iSecs,
                    LPBYTE lpBuffer,
                    UINT uMemSize );

```

```
#endif  
#endif
```

```
/* #ifndef DSP_VERSIONONLY */
```