

```

/*****
/*
/*----- M I X U T I L . H -----*/
/* Task      : Header for MIXUTIL.H */
/*-----*/
/* Author     : Michael Tischer / Bruno Jennrich */
/* Developed on : 03/20/1994 */
/* Last update  : 04/06/1995 */
/*-----*/
/* COMPILER    : Borland C++ 3.1, Microsoft Visual C++ 1.5 */
/*****
#ifndef __INC_MIX_UTIL_H
#define __INC_MIX_UTIL_H

#include "types.h"

/* Mixer port offsets */
#define MIX_REGISTERPORT 0x04 /* Setting registers... */
#define MIX_DATAPORT 0x05 /* Read/write data */

/* CT1335 Mixer (in DSP 2.xx) */
#define MIX_RESET 0x00 /* Reset mixer */
#define MIX2_MASTERVOL 0x02 /* Total Volume (BITS 3,2,1) */
#define MIX2_MIDIVOL 0x06 /* MIDI(FM) Volume (BITS 3,2,1) */
#define MIX2_CDVOL 0x08 /* CD Volume (BITS 3,2,1) */
#define MIX2_VOICEVOL 0x0A /* VOICE(DSP) Volume (BITS 2,1) */

/* CT1345 Mixer (in DSP 3.xx) */
#define MIX3_VOICEVOL 0x04 /* VOICE Vol. (Bits 7,6,5 3,2,1) */
#define MIX3_MICVOL 0x0A /* Microphone Volume (BITS 2,1) */

#define MIX3_ADCSTATE 0x0C /* DSP input and filter */
/* Input filter */
#define MIX3_ADCFILTEROFF 0x20 /* Bit 5: Low pass filter on */
#define MIX3_LOWPASS88 0x08 /* Bit 3: 8.8 kHz Low pass filter */
/* Input source */
#define MIX3_MICSRC 0x00 /* BIT 2,1 = 00: Microphone */
#define MIX3_MICSRC_ 0x04 /* BIT 2,1 = 10: also microphone */
#define MIX3_CDSRC 0x02 /* BIT 2,1 = 01: CD */
#define MIX3_LINESRC 0x06 /* BIT 2,1 = 11: LINE IN */
#define MIX3_SRCMSK 0x06 /* Bit mask of source bits */

#define MIX3_DACSTATE 0x0E /* Output filter and stereo switch */
#define MIX3_DACFILTEROFF 0x20 /* Bit 5: Low pass filter on */
/* (see above) */
#define MIX3_STEREOON 0x02 /* Bit 1: Stereo output */

#define MIX3_MASTERVOL 0x22 /* Total vol. (BITS 7,6,5 and 3,2,1) */
#define MIX3_MIDIVOL 0x26 /* MIDI-Vol. (BITS 7,6,5 and 3,2,1) */
#define MIX3_CDVOL 0x28 /* CD-Vol (BITS 7,6,5 and 3,2,1) */
#define MIX3_LINEVOL 0x2E /* LINE-Vol. (BITS 7,6,5 and 3,2,1) */

/* CT1745 Mixer (in DSP 4.xx) as well as ASP */
/* Volumes: */
#define MIX4_MASTERVOL_L 0x30 /* Total left (BITS 7,6,5,4,3) */
#define MIX4_MASTERVOL_R 0x31 /* Total right (BITS 7,6,5,4,3) */
#define MIX4_VOICEVOL_L 0x32 /* VOICE(DSP) left (BITS 7,6,5,4,3) */
#define MIX4_VOICEVOL_R 0x33 /* VOICE(DSP) right (BITS 7,6,5,4,3) */
#define MIX4_MIDIVOL_L 0x34 /* MIDI left (BITS 7,6,5,4,3) */
#define MIX4_MIDIVOL_R 0x35 /* MIDI right (BITS 7,6,5,4,3) */
#define MIX4_CDVOL_L 0x36 /* CD links (BITS 7,6,5,4,3) */
#define MIX4_CDVOL_R 0x37 /* CD right (BITS 7,6,5,4,3) */
#define MIX4_LINEVOL_L 0x38 /* LINE left (BITS 7,6,5,4,3) */
#define MIX4_LINEVOL_R 0x39 /* LINE right (BITS 7,6,5,4,3) */
#define MIX4_MICVOL 0x3A /* Microphone (BITS 7,6,5,4,3) */
#define MIX4_PCSPEAKERVOL 0x3B /* PC speaker (BITS 7,6 ) */

#define MIX4_OUTSOURCE 0x3C /* Output sources */
#define MIX4_ADCSOURCE_L 0x3D /* Sample sources L/R */
#define MIX4_ADCSOURCE_R 0x3E

#define MIX4_MIDI_L 0x40 /* Active recording sources */
#define MIX4_MIDI_R 0x20

```

```

#define MIX4_LINE_L      0x10
#define MIX4_LINE_R      0x08
#define MIX4_CD_L        0x04
#define MIX4_CD_R        0x02
#define MIX4_MIC          0x01

/*--- Preamplifier for output(OUT) and samples(ADC) (Bits 7 and 6 )-- */
#define MIX4_ADCGAIN_L    0x3F
#define MIX4_ADCGAIN_R    0x40
#define MIX4_OUTGAIN_L    0x41
#define MIX4_OUTGAIN_R    0x42

#define MIX4_AGC      0x43          /* Microphone Preamplifier (20dB) */
#define MIX4_AGC_CON 0x01          /* Microphone Preamplifier on */

/*--- Treble and bass of preamplifier (BITS 7,6,5,4 )----- */
#define MIX4_TREBLE_L    0x44
#define MIX4_TREBLE_R    0x45
#define MIX4_BASS_L      0x46
#define MIX4_BASS_R      0x47
/*--- Which interrupts and DMA lines are being used ?-----*/
#define MIX4_IRQ      0x80
#define MIX4_IRQ2     0x01          /* 4 possible interrupt lines */
#define MIX4_IRQ5     0x02
#define MIX4_IRQ7     0x04
#define MIX4_IRQ10    0x08

#define MIX4_DMA      0x81          /*- Which DMA line is being used ? */
#define MIX4_DMA0     0x01
#define MIX4_DMA1     0x02
/* #define MIX4_DMA2 0x04          disk drive */
#define MIX4_DMA3     0x08
/* #define MIX4_DMA4 0x10          Cascading */
#define MIX4_DMA5     0x20
#define MIX4_DMA6     0x40
#define MIX4_DMA7     0x80

#define MIX4_IRQSOURCE 0x82
#define MIX4_IRQ8DMA   0x01          /* Interrupt of 8 bit DMA and Midi */
#define MIX4_IRQ16DMA  0x02          /* Interrupt of 16 bit DMA */
#define MIX4_IRQMPU    0x04          /* Interrupt of MPU */

/* MIXUTIL specific constants */
/* To access volume arrays, constants are used */
/* that specify whether a port address refers to the right channel, */
/* the left channel or both channels */
#define PORT      0      /* For Array access: 1st Element = Port */
#define CHANNEL   1      /* 2nd Element = Access code */

#define L 0          /* left */
#define R 1          /* right */

#define CH_LEFT 1          /* ID for left channel */
#define CH_RIGHT 2        /* right channel */
#define CH_BOTH 3         /* both channels */
#define CH_NONE 0         /* no channel */
#define CH_MAX 6          /* Use maximum of right and left */

/*--- Consecutive numbering of available sources ----- */
#define CD      0
#define LINE    1
#define VOICE    2
#define MASTER  3
#define MIDI    4
#define MIC      5
#define PCSPEAKER 6
#define NUM_SOURCES 7

#define CD_L      7
#define LINE_L    8
#define VOICE_L   9
#define MASTER_L 10
#define MIDI_L    11
#define CD_R      12
#define LINE_R    13
#define VOICE_R   14

```

```

#define MASTER_R      15
#define MIDI_R        16

#define MAX_SRC        17

#define DAC TRUE          /* for 'mix3_PrepareForStereo' */
#define ADC FALSE

/* Prototypes */

WORD mix_SetBase( PSBBASE pSBBASE, WORD iReset );
VOID mix_Write( WORD iReg, WORD iData );
WORD mix_Read( WORD iReg );
VOID mix_Reset( VOID );
VOID mix3_SetADCFilter( WORD iState );
WORD mix3_GetADCFilter( VOID );
VOID mix3_SetDACFilter( WORD iState );
WORD mix3_GetDACFilter( VOID );
VOID mix3_SetDACStereo( WORD iState );
WORD mix3_GetDACStereo( VOID );
VOID mix3_SetADDACLowPass( WORD iState );
WORD mix3_GetADDACLowPass( VOID );
VOID mix3_PrepareForStereo( WORD iMode );
VOID mix3_RestoreFromStereo( VOID );
VOID mix3_SetVolume( WORD iSource, WORD iVolL, WORD iVolR );
WORD mix3_GetVolume( WORD iSource );
VOID mix3_SetADCSource( WORD iSource );
WORD mix3_GetADCSource( VOID );

VOID mix4_PrepareForMonoADC( VOID );
VOID mix4_RestoreFromMonoADC( VOID );
VOID mix4_SetVolume( WORD iSource, WORD iVolL, WORD iVolR );
WORD mix4_GetVolume( WORD iSource );
VOID mix4_SetADCSourceL( WORD iSource, WORD iState );
VOID mix4_SetADCSourceR( WORD iSource, WORD iState );
WORD mix4_GetADCSourceL( WORD iSource );
WORD mix4_GetADCSourceR( WORD iSource );
VOID mix4_SetOUTSource( WORD iSource, WORD iState );
WORD mix4_GetOUTSource( WORD iSource );
VOID mix4_SetADCGain( WORD iGainL, WORD iGainR );
WORD mix4_GetADCGain( WORD iChannel );
VOID mix4_SetOUTGain( WORD iGainL, WORD iGainR );
WORD mix4_GetOUTGain( WORD iChannel );
VOID mix4_SetAGC( WORD iState );
WORD mix4_GetAGC( VOID );
VOID mix4_SetTreble( WORD iTrebleL, WORD iTrebleR );
WORD mix4_GetTreble( WORD iChannel );
VOID mix4_SetBass( WORD iBassL, WORD iBassR );
WORD mix4_GetBass( WORD iChannel );
#endif

```