```
/*
 *  Copyright 2010 by Spectrum Digital Incorporated.
 *  All rights reserved. Property of Spectrum Digital Incorporated.
 */

/*
 *  AIC3204 Test
 *
 */

#define AIC3204_I2C_ADDR 0x18
#include "usbstk5515.h"
#include "usbstk5515_gpio.h"
#include "usbstk5515_i2c.h"
#include "stdio.h"

extern Int16 aic3204_loop_stereo_in1( );

/* ------------------------------------------------------------------------ *
 *                                                                          *
 *  _AIC3204_rget( regnum, regval )                                         *
 *                                                                          *
 *      Return value of codec register regnum                               *
 *                                                                          *
 * ------------------------------------------------------------------------ */
Int16 AIC3204_rget(  Uint16 regnum, Uint16* regval )
{
    Int16 retcode = 0;
    Uint8 cmd[2];

    cmd[0] = regnum & 0x007F;       // 7-bit Register Address
    cmd[1] = 0;

    retcode |= USBSTK5515_I2C_write( AIC3204_I2C_ADDR, cmd, 1 );
    retcode |= USBSTK5515_I2C_read( AIC3204_I2C_ADDR, cmd, 1 );

    *regval = cmd[0];
    USBSTK5515_wait( 10 );
    return retcode;
}

/* ------------------------------------------------------------------------ *
 *                                                                          *
 *  _AIC3204_rset( regnum, regval )                                         *
 *                                                                          *
 *      Set codec register regnum to value regval                           *
 *                                                                          *
 * ------------------------------------------------------------------------ */
Int16 AIC3204_rset( Uint16 regnum, Uint16 regval )
{
    Uint8 cmd[2];
    cmd[0] = regnum & 0x007F;       // 7-bit Register Address
    cmd[1] = regval;                // 8-bit Register Data

    return USBSTK5515_I2C_write( AIC3204_I2C_ADDR, cmd, 2 );
}

/* ------------------------------------------------------------------------ *
 *                                                                          *
 *  aic3204_test( )                                                           *
 *                                                                          *
 * ------------------------------------------------------------------------ */
Int16 aic3204_test( )
{
    // Configure AIC3204
    AIC3204_rset( 0, 0);         // Select page 0
    AIC3204_rset( 1, 1);         // Reset codec
    AIC3204_rset( 0, 1);         // Point to page 1
```

```
    AIC3204_rset( 1, 8);         // Disable crude AVDD generation from DVDD
    AIC3204_rset( 2, 1);         // Enable Analog Blocks, use LDO power
    AIC3204_rset( 0, 0);         // Select page 0

    // PLL and Clocks config and Power Up
    AIC3204_rset( 27, 13);       // I2S, 16bits, BCLK e WCLK output, DOUT alta
impedancia
    AIC3204_rset( 28, 0);       // Data ofset = 0
    AIC3204_rset( 4, 3);         // PLL setting: PLLCLK <- MCLK, CODEC_CLKIN <-PLLCLK
    AIC3204_rset( 6, 8);         // PLL setting: J=8
    AIC3204_rset( 7, 18);        // PLL setting: MSB D=4672
    AIC3204_rset( 8, 64);        // PLL setting: LSB D=4672
    //AIC3204_rset( 29, 0);         // BDIV_CLKIN = DAC_CLK
    AIC3204_rset( 30, 0x98);    // BCLK / 24 = 705600 Hz
    AIC3204_rset( 5, 145);       // PLL setting: Power up PLL, P=1 e R=1
    AIC3204_rset( 13, 0 );       // MSB for DOSR = 128
    AIC3204_rset( 14, 128);      // LSB for DOSR = 128
    AIC3204_rset( 20, 128);      // AOSR for AOSR = 128
    AIC3204_rset( 11, 134);    // Power up NDAC and set NDAC value to 6
    AIC3204_rset( 12, 134);    // Power up MDAC and set MDAC value to 6
    AIC3204_rset( 18, 134);    // Power up NADC and set NADC value to 6
    AIC3204_rset( 19, 134);    // Power up MADC and set MADC value to 6

    // DAC ROUTING and Power Up
    AIC3204_rset(  0, 1);      // Select page 1
    AIC3204_rset( 12, 8);      // LDAC AFIR routed to HPL
    AIC3204_rset( 13, 8);      // RDAC AFIR routed to HPR
    AIC3204_rset(  0, 0);      // Select page 0
    AIC3204_rset( 64, 2);      // Left vol=right vol
    AIC3204_rset( 65, 16);     // Left DAC gain to xdB VOL; Right tracks Left
    AIC3204_rset( 63, 0xd4);   // Power up left, right data paths and set channel
    AIC3204_rset(  0, 1);      // Select page 1
    AIC3204_rset( 16, 0x06);   // Unmute HPL , 6dB gain
    AIC3204_rset( 17, 0x06);   // Unmute HPR , 6dB gain
    AIC3204_rset(  9, 0x30);   // Power up HPL, HPR
    AIC3204_rset(  0, 0);      // Select page 0
    USBSTK5515_wait( 500 );    // Wait

    // ADC ROUTING and Power Up
    AIC3204_rset(  0, 1 );       // Select page 1
    AIC3204_rset( 52, 0x30 );    // IN2_L to LADC_P through 40 kohm
    AIC3204_rset( 55, 0x30 );    // IN2_R to RADC_P through 40 kohmm
    AIC3204_rset( 54, 0x03 );    // CM2L (common mode) to LADC_M through 40 kohm
    AIC3204_rset( 57, 0x03 );    // CM2R (common mode) to RADC_M through 40 kohm
    AIC3204_rset( 59, 0x00 );    // MIC_PGA_L unmute
    AIC3204_rset( 60, 0x00 );    // MIC_PGA_R unmute
    AIC3204_rset(  0, 0 );       // Select page 0
    AIC3204_rset( 81, 0xc0 );    // Power up Left and Right ADC
    AIC3204_rset( 82, 0x00 );    // Fine gain
//  AIC3204_rset( 83, 0x20 );    // Unmute Left ADC
//  AIC3204_rset( 84, 0x20 );    // Unmute Right ADC
    AIC3204_rset( 0, 0 );
    USBSTK5515_wait( 200 );  // Wait

    // I2S settings
    I2S0_CR = 0x8010;   // 16-bit word, slave, habilita I2S
    //I2S0_SRGR = 0x0000;   // quando slave não tem efeito
    //I2S0_ICMR = 0x003f; // habilita todas as interrupções

    /*AIC3204_rset( 0, 0 );          // Select page 0
    AIC3204_rset( 1, 1 );          // Reset codec
    AIC3204_rset( 0, 1 );          // Point to page 1
    AIC3204_rset( 1, 8 );          // Disable crude AVDD generation from DVDD
    AIC3204_rset( 2, 1 );          // Enable Analog Blocks, use LDO power
    AIC3204_rset( 0, 0 );          // Select page 0
    // PLL and Clocks config and Power Up
    AIC3204_rset( 27, 0x0d );      // BCLK and WCLK is set as o/p to AIC3204(Master)
    AIC3204_rset( 28, 0x00 );      // Data ofset = 0
```

```
    AIC3204_rset( 4, 3 );          // PLL setting: PLLCLK <- MCLK, CODEC_CLKIN <-PLL
CLK
    AIC3204_rset( 6, 8 );          // PLL setting: J=8
    AIC3204_rset( 7, 15 );         // PLL setting: HI_BYTE(D)
    AIC3204_rset( 8, 0xdc );       // PLL setting: LO_BYTE(D)
    AIC3204_rset( 30, 0x88 );      // For 32 bit clocks per frame in Master mode ONLY
                                   // BCLK=DAC_CLK/N =(12288000/8) = 1.536MHz = 32*fs
    AIC3204_rset( 5, 0x91 );       // PLL setting: Power up PLL, P=1 and R=1
    AIC3204_rset( 13, 0 );         // Hi_Byte(DOSR) for DOSR = 128 decimal or 0x0080
DAC oversamppling
    AIC3204_rset( 14, 0x80 );      // Lo_Byte(DOSR) for DOSR = 128 decimal or 0x0080
    AIC3204_rset( 20, 0x80 );      // AOSR for AOSR = 128 decimal or 0x0080 for
decimation filters 1 to 6
    AIC3204_rset( 11, 0x88 );      // Power up NDAC and set NDAC value to 8
    AIC3204_rset( 12, 0x82 );      // Power up MDAC and set MDAC value to 2
    AIC3204_rset( 18, 0x88 );      // Power up NADC and set NADC value to 8
    AIC3204_rset( 19, 0x82 );      // Power up MADC and set MADC value to 2
    // DAC ROUTING and Power Up
    AIC3204_rset(  0, 0x01 );      // Select page 1
    AIC3204_rset( 12, 0x08 );      // LDAC AFIR routed to HPL
    AIC3204_rset( 13, 0x08 );      // RDAC AFIR routed to HPR
    AIC3204_rset(  0, 0x00 );      // Select page 0
    AIC3204_rset( 64, 0x02 );      // Left vol=right vol
    AIC3204_rset( 65, 0x00 );      // Left DAC gain to 0dB VOL; Right tracks Left
    AIC3204_rset( 63, 0xd4 );      // Power up left,right data paths and set channel
    AIC3204_rset(  0, 0x01 );      // Select page 1
    AIC3204_rset( 16, 0x06 );      // Unmute HPL , 6dB gain
    AIC3204_rset( 17, 0x06 );      // Unmute HPR , 6dB gain
    AIC3204_rset(  9, 0x30 );      // Power up HPL,HPR
    AIC3204_rset(  0, 0x00 );      // Select page 0
    USBSTK5515_wait( 500 );        // Wait

    // ADC ROUTING and Power Up
    AIC3204_rset( 0, 1 );          // Select page 1
    AIC3204_rset( 0x34, 0x30 );    // STEREO 1 Jack
                                   // IN2_L to LADC_P through 40 kohm
    AIC3204_rset( 0x37, 0x30 );    // IN2_R to RADC_P through 40 kohmm
    AIC3204_rset( 0x36, 3 );       // CM_1 (common mode) to LADC_M through 40 kohm
    AIC3204_rset( 0x39, 0xc0 );    // CM_1 (common mode) to RADC_M through 40 kohm
    AIC3204_rset( 0x3b, 0 );       // MIC_PGA_L unmute
    AIC3204_rset( 0x3c, 0 );       // MIC_PGA_R unmute
    AIC3204_rset( 0, 0 );          // Select page 0
    AIC3204_rset( 0x51, 0xc0 );    // Powerup Left and Right ADC
    AIC3204_rset( 0x52, 0 );       // Unmute Left and Right ADC

    AIC3204_rset( 0, 0 );
    USBSTK5515_wait( 200 );  // Wait
    // I2S settings
    I2S0_SRGR = 0x0;
    I2S0_CR = 0x8010;     // 16-bit word, slave, enable I2C
    I2S0_ICMR = 0x3f;     // Enable interrupts*/

    printf( "Audio Configurado\n" );

    return 1;

}
```