**EECS 363: Digital Filtering**

**Lab 3 - 1/25/2017**

**Karan Shah**

**Code:**

**#include** "stdio.h"

**#include** "csl\_i2s.h"

**#include** "csl\_intc.h"

**#include** "usbstk5505.h"

**#include** "usbstk5505\_led.h" // added for led control

**#include** "aic3204.h"

//#include "usbstk5505\_gpio.h"//added, compiles without

//#include "usbstk5505\_i2c.h"//added, compiles without

**#include** "PLL.h"

**#include** "stereo.h"

**#include** "stereo.c"

Int16 left\_input; //do these interfere with declarations in aic3204.c?

Int16 right\_input;

Int16 left\_output;

Int16 right\_output;

Int16 mono\_input;

**#define** SAMPLES\_PER\_SECOND 1500000

**unsigned** **long** j = 0; //added for led control

**short** toggle = 0; //added for led control

**interrupt** **void** **codec\_read\_isr**(**void**);

**void** **main**( **void** )

{

/\* Initialize BSL \*/

USBSTK5505\_init( );

**USBSTK5505\_LED\_init**(); //added for LED control

/\* Initialize PLL \*/

**pll\_frequency\_setup**(100);

/\* Initialise hardware interface and I2C for code \*/

aic3204\_hardware\_init();

/\* Initialise the AIC3204 codec \*/

aic3204\_init();

**printf**("\n\nRunning Getting Started Project\n");

**printf**( "<-> Audio Loopback from Stereo IN --> to HP/Lineout\n" );

/\* Setup sampling frequency and 30dB gain for microphone \*/

set\_sampling\_frequency\_and\_gain(SAMPLES\_PER\_SECOND, 0);// was 30 dB; I changed.

**asm**(" bclr XF");

**IRQ\_plug**(RCV2\_EVENT, &codec\_read\_isr);

**IRQ\_enable**(RCV2\_EVENT);

**IRQ\_globalEnable**();

**while**(1)

{

// begin segment for led control

**if** (j++ == SAMPLES\_PER\_SECOND)

{

toggle = 1-toggle;

**asm**(" SSBX INTM");

**if** (toggle)

**USBSTK5505\_LED\_on**(0);

**else**

**USBSTK5505\_LED\_off**(0);

j = 0;

}

// end segment for led control

**asm**(" RSBX INTM");

}

/\* Disable I2S and put codec into reset \*/

aic3204\_disable();

**printf**( "\n\*\*\*Program has Terminated\*\*\*\n" );

SW\_BREAKPOINT;

}

**interrupt** **void** **codec\_read\_isr**(**void**)

{

left\_input = I2S2\_W0\_MSW\_R; // Read Most Significant Word of channel 1

right\_input = I2S2\_W1\_MSW\_R; // Read Most Significant Word of channel 2

left\_output = left\_input; // Replace with your own code!

right\_output = right\_input; // Directly connect inputs to outputs.

**if** (I2S2\_IR & RcvR)

{

I2S2\_W0\_MSW\_W = left\_output; // Left output

I2S2\_W1\_MSW\_W = right\_output; // Right output

}

**return**;

}