

```

////////////////////////////////////////////////////////////////////
// Florida Atlantic University, 2014
// Biomedical Signal Processing Laboratory
// KHRYSTSINA NAVUMENKA
// Power Line Noise eliminating filter C code
// For TI OMAP-L138 LCDK
// Filename: DigitalNotchFilter.c
//
// Synopsis: Interrupt service routine for codec data transmit/receive
//
//////////////////////////////////////////////////////////////////

#include"DSP_Config.h"

// Data is received as 2 16-bit words (left/right) packed into one
// 32-bit word. The union allows the data to be accessed as a single
// entity when transferring to and from the serial port, but still be
// able to manipulate the left and right channels independently.

#defineLEFT 0
#define RIGHT 1

volatileunion {
    Uint32UINT;
    Int16Channel[2];
} CodecDataIn, CodecDataOut;

/* add any global variables here */
#define N 2 // IIR filter order

float B[N+1] = {0.2810, -0.5567, 0.2810}; // numerator coefficients, q=0.1
float A[N+1] = {1.0000, -0.5567, -0.4379}; // denominator coefficients
float x[N+1] = {0.0000, 0.0000, 0.0000}; // input value (buffered)
float y[N+1] = {0.0000, 0.0000, 0.0000}; // output values (buffered)

interruptvoidCodec_ISR()
//////////////////////////////////////////////////////////////////
// Purpose: Codec interface interrupt service routine
//
// Input: None
//
// Returns: Nothing
//
// Calls: CheckForOverrun, ReadCodecData, WriteCodecData
//
// Notes: None
//////////////////////////////////////////////////////////////////
{
    /* add any local variables here */

    if(CheckForOverrun()) // overrun error occurred (i.e. halted DSP)
return; // so serial port is reset to recover

```

```

CodecDataIn.UINT = ReadCodecData();    // get input data samples

/* IIR filter routine */
x[0] = CodecDataIn.Channel[LEFT]; // current input value
CodecDataIn.Channel[RIGHT]=CodecDataIn.Channel[LEFT];
y[0] = -A[1]*y[1]-A[2]*y[2]+ B[0]*x[0] + B[1]*x[1]+B[2]*x[2];/calc. the output

CodecDataOut.Channel[LEFT] = 25*y[0]; // setup the LEFT value
CodecDataOut.Channel[RIGHT] = 25*y[0];// setup the Right value

/* end of my IIR filter routine */

WriteCodecData(CodecDataOut.UINT);    // send output data to port
x[2] = x[1];                          // setup for the next input
y[2] = y[1];
x[1] = x[0];
y[1] = y[0];
x[1] = x[0];
}

```