

Figure 3kHz waveform

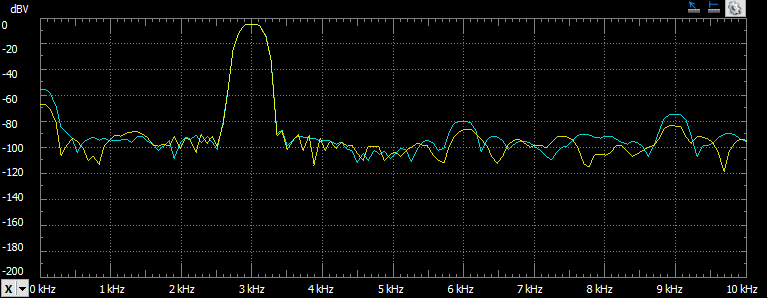


Figure 3kHz FFT

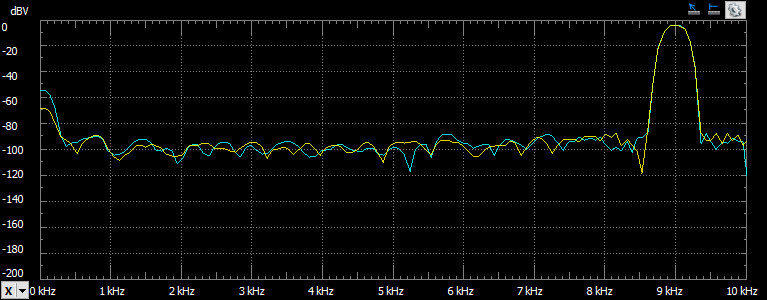


Figure 9kHz FFT

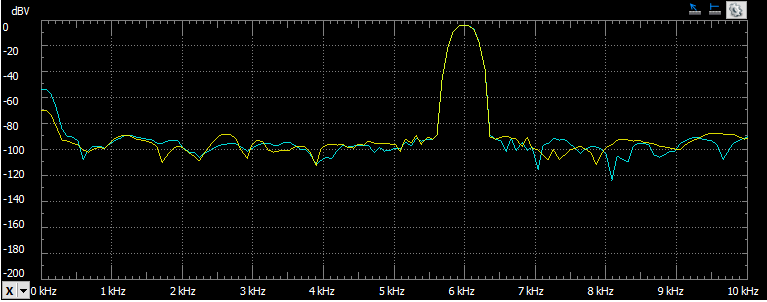


Figure 6kHz FFT

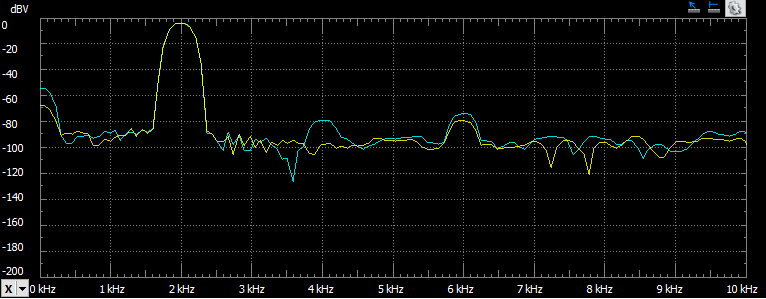


Figure 2kHz FFT

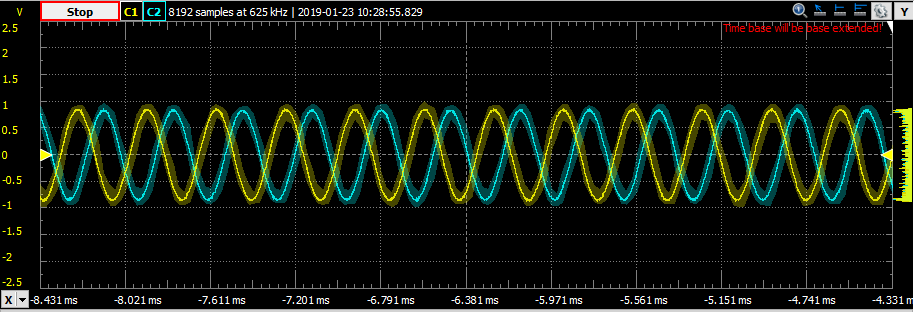


Figure 3kHz initial waveform

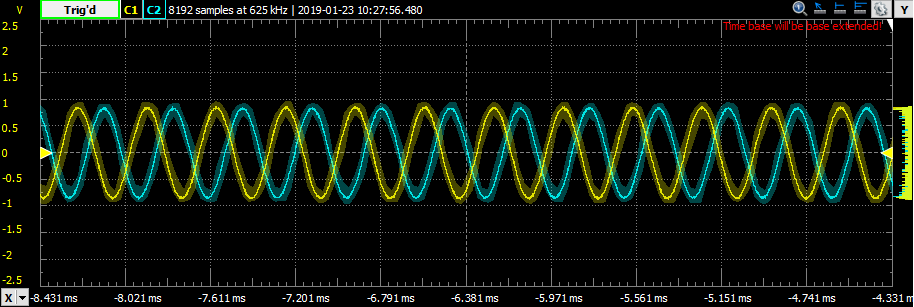


Figure 3kHz waveform after 10 minutes

Sinewave.c

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

{

**float** ycData = AMPLITUDE; // Sine Sample

**float** oldYcData; // Sine Sample delayed

**float** ysData = AMPLITUDE; // Sine Sample

**float** oldYsData; // Sine Sample delayed

**float** sinVal = **sin**(OMEGA\_0); // sin value at predetermined w0

**float** cosVal = **cos**(OMEGA\_0); // cos value at predetermined w0

Init();

// Infinite loop: Each loop reads/writes one sample to the left and right channels.

**while** (*true*){

//store y[n-1]

oldYcData = ycData;

oldYsData = ysData;

// compute output as shown in Proakis text, p. 349

ycData = (oldYcData\*cosVal - oldYsData\*sinVal);

ysData = (oldYcData\*sinVal + oldYsData\*cosVal);

//output samples

// wait for xmit ready and send a sample to the left channel.

**while** (!CHKBIT(MCASP->SRCTL11, XRDY)) {}

MCASP->XBUF11 = (int16\_t) ycData;

// wait for xmit ready and send a sample to the right channel.

**while** (!CHKBIT(MCASP->SRCTL11, XRDY)) {}

MCASP->XBUF11 = (int16\_t) ysData;

}

}

Sinewave.h

// Set the sampling frequency at which the DAC will be operating.

**#define** SAMP\_FREQ (24000) // Default

// The frequency of the sine wave that will be output to the line out port.

//#define SIN\_FREQ (2000) // in Hz

**#define** SIN\_FREQ (3000) // in Hz

//#define SIN\_FREQ (6000) // in Hz

//#define SIN\_FREQ (9000) // in Hz

**#define** PI (3.14159265358979)

// Amplitude of the sinewave

**#define** AMPLITUDE (20000)

// Frequency of the sinewave

**#define** OMEGA\_0 (2.0\*PI\*SIN\_FREQ/SAMP\_FREQ)