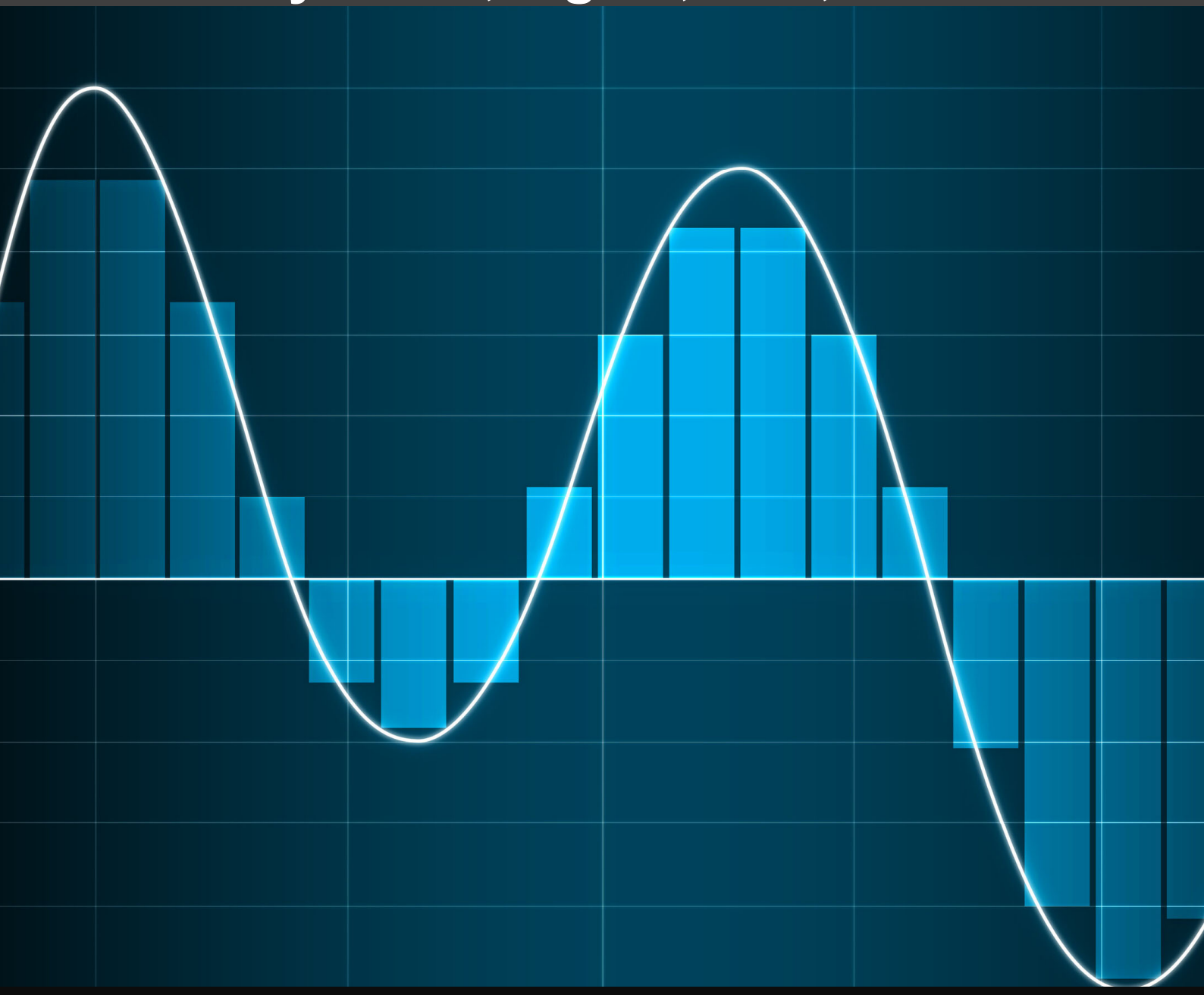
MATRIX | SYSBLOCKS

Systems, signal, DSP, FFT



MATRIX

CP2398

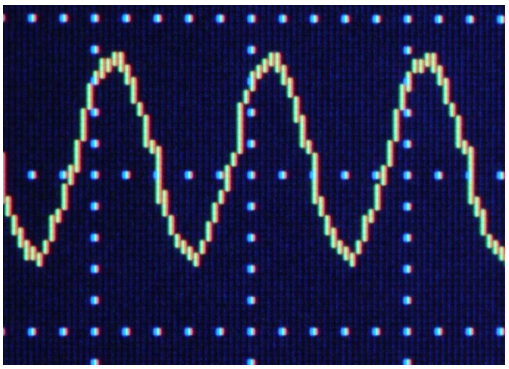
www.matrixtsl.com

Contents

Worksheet 1
Sampling

MATRIX | SYSBLOCKS

Systems and Signals



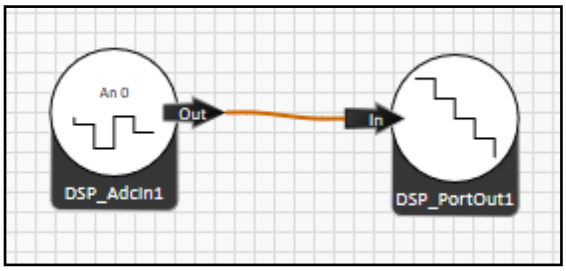
Digital audio recording takes samples of the audio signal at regular intervals. The number of samples taken each second is known as the ‘sample rate’. The higher the sample rate, - the more closely the result resembles the original, but the bigger is the resulting data file. There must be a compromise!

About the program

This program samples an analogue signal at input ‘IN0’ and reconstructs it at output ‘OUT0’. Since the signal is sampled at discrete times, rather than being monitored continuously, there will be steps in the output signal. The analogue input signal varies continuously but the output signal changes only at discrete times, when sampled.

The program uses the encoder ‘ENC0’ setting to set the interrupt rate. The interrupt routine samples the input signal periodically and uses it to generate the output.

The sample rate of 44.1 kHz is included because it is typical of digital audio applications. It is greater than twice the upper frequency limit of human hearing. Mathematically, it also has a large number of factors, a useful property!



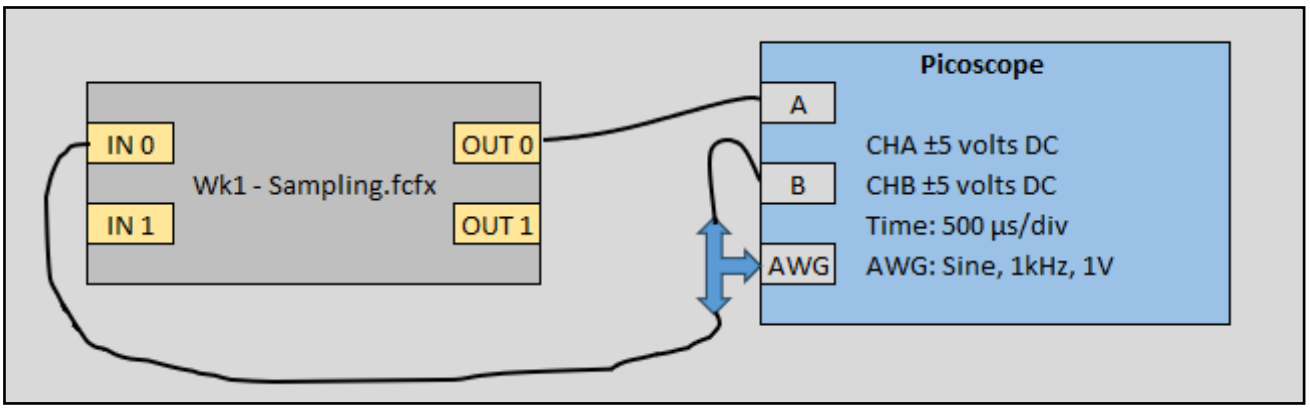
Project settings:

Firmware	Wk1 - Sampling.fcfx
Inputs	‘IN0’ - signal to be sampled (AC coupled)
	AWG - set to generate sinusoidal signal with amplitude 1V and frequency 1kHz
Outputs	‘OUT0’ = reconstructed signal
DSP Interrupt	8 kHz, 16 kHz or 44.1 kHz
Controls	‘ENC0’ selects the interrupt rate and thus sample rate.
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running. Display indicates the selected sample rate.

Worksheet 1
Sampling

Systems and Signals

Hardware and set up:



Over to you:

- Set the PicoScope AWG to generate a sine wave with frequency 1 kHz and amplitude 1 V.
- Use a T-piece connector to connect this signal to both the Sysblocks input socket ‘IN0’ and Picoscope channel B.
- Connect the Sysblocks output socket ‘OUT0’ to Picoscope channel A, with a timebase setting of 500 µs/div.
- Compare the output signals for sample rates of 8, 16 and 44.1 kHz.
- Use cursors to measure the duration of each step, (corresponding to the sample time.)
- Compare the input and output waveforms. Notice that the reconstructed signal is effectively delayed by half the sample period.

Challenge

- Using the ‘Math’ tool of the oscilloscope, subtract the reconstructed signal from the original. The result is called ‘quantisation noise’.
- See how this varies with the sample rate.
- Change the shape and frequency of the input signal and look at the effects.
- Plug either speakers or headphones into the ‘LINE OUT’ socket and compare the sounds that are produced.

Worksheet 2
Nyquist's theorem

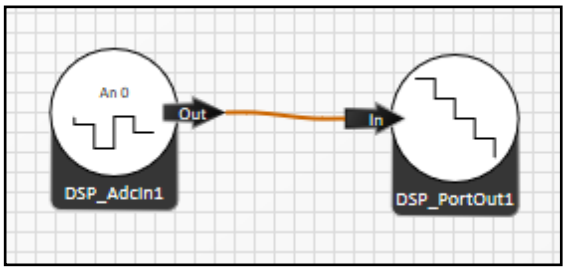
MATRIX | SYSBLOCKS

Systems and Signals

When the sample rate is much lower than the signal frequency, the resulting reconstruction is not a good representation of the original. It may even generate false output frequencies, known as aliases. Nyquist's sampling theorem specifies the minimum sample rate for a given signal.

About the program

Once again, the program samples an analogue signal at input 'IN0' and reconstructs it at output 'OUT0'. The encoder 'ENC0' again sets the sample rate.
As the signal frequency is increased, eventually, there comes a point where the reconstructed wave appears to be of a lower frequency than the input wave form. This is called aliasing.



Project settings:

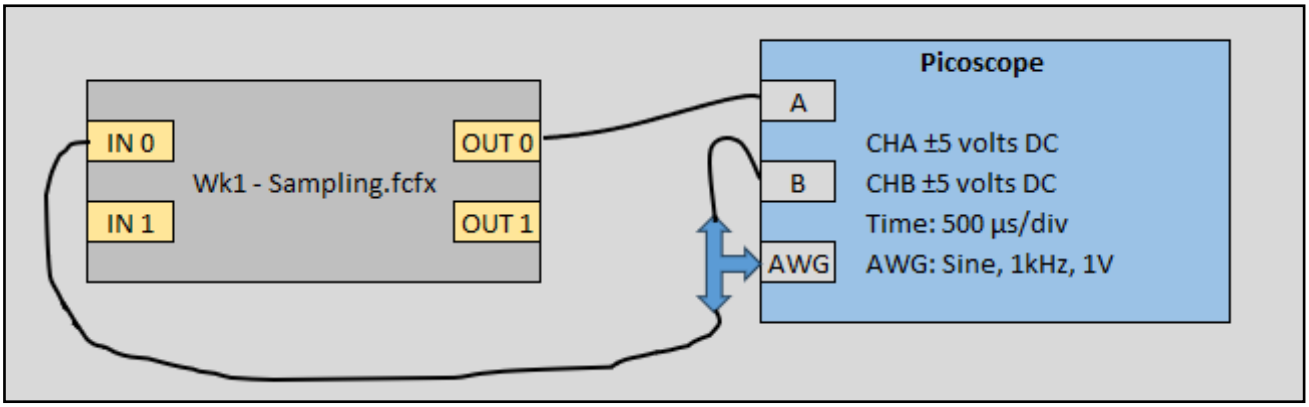
Firmware	Wk2 - Nyquist.fcfx
Inputs	'IN0' - signal to be sampled (AC coupled)
	AWG - set to generate sinusoidal signal with amplitude 1V and frequency 1kHz
Outputs	'OUT0' - reconstructed signal
DSP Interrupt	10 kHz or 20 kHz
Controls	'ENC0' selects the interrupt rate and thus sample rate.
Indicators	'LED1' is a 'heartbeat' showing that the main loop is running. Display indicates the selected sample rate.

Worksheet 2
Nyquist's theorem

MATRIX | SYSBLOCKS

Systems and Signals

Hardware and set up:



The hardware is set up as in worksheet 1.

Over to you:

- Set the PicoScope AWG to generate a sine wave with frequency 1 kHz and amplitude 1 V.
- Set the sample rate to 10 kHz.
- Compare the original and output signals as before.
- Increase the frequency of the input signal in steps of 1 kHz and observe the effect.
- Find the point at which the reconstructed wave appears to be of a lower frequency than the input signal. Measure the apparent frequency of this reconstructed wave.
- Repeat the same procedure for a sample rate of 20 kHz and compare the effect.
- Use the spectrum analyser view on the oscilloscope and compare the two sample rates. Try to interpret what the spectrum analyser is showing.

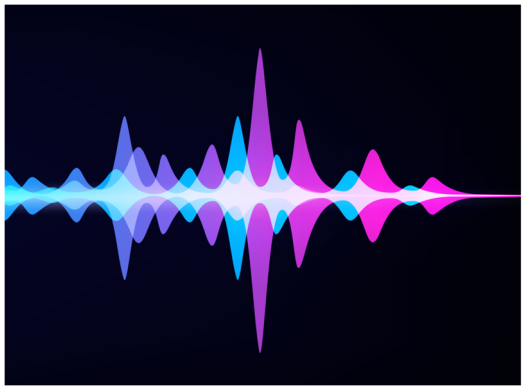
Challenge

- Change the PicoScope AWG to generate square and then triangular waves. How well are they reproduced?
- What is the effect of a square wave on the spectrum analyser and why?
- Look at the spectrum analyser view. These waves have components at frequencies higher than their fundamental. While the fundamental frequency might fulfil the Nyquist criterion, the higher frequency components will be lost. The frequency of the output may match that of the input but its shape is not the same.

Worksheet 3
Analogue-to-digital conversion

MATRIX | SYSBLOCKS

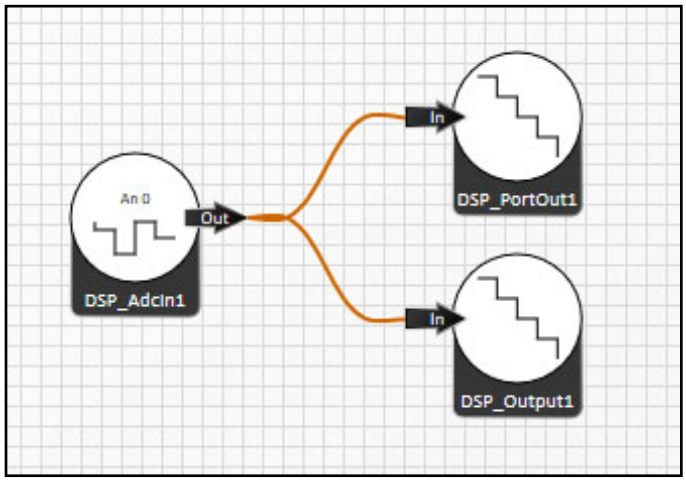
Systems and Signals



The outside world is a largely analogue environment - louder / quieter sounds, brighter / dimmer lights etc. Microprocessors, and much of modern electronics, are digital with quantities usually expressed using binary numbers, '0's and '1's.
When information is passed from the outside world to a microprocessor system, an ADC converts between these data formats.

About the program

The first program uses signals from the Picoscope 'AWG' and converts them into digital signals. These are displayed on the 'VU' matrix on the Sysblocks board.



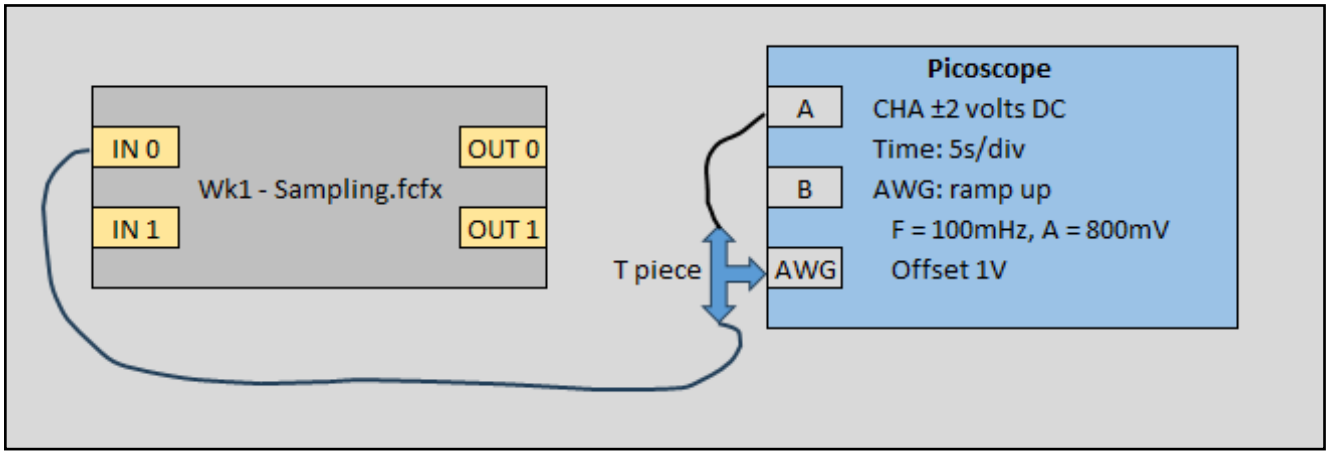
Project settings:

Firmware	Wk3 - ADC.fcfx
Inputs	'IN0' - signal to be sampled (DC coupled)
	AWG - set to generate rising ramp signal with amplitude 800mV, a frequency 100mHz and DC offset 1V
Outputs	LEDs on 'VU' matrix on Sysblocks board
DSP Interrupt	100 kHz
Controls	None
Indicators	'LED1' is a 'heartbeat' showing that the main loop is running.

Worksheet 3
Analogue-to-digital conversion

Systems and Signals

Hardware and set up:



Over to you:

- To begin with, set the Picoscope AWG to deliver a rising ramp signal with a frequency of 100mHz, an amplitude of 800mV and a DC offset of 1V.
- Observe the VU matrix LEDs. They show the digital output corresponding to the ramp input signal.
- Experiment with the AWG settings, observing the VU matrix LEDs as you do so.

Challenge

- Next, change the PicoScope AWG to generate a DC voltage. In this mode, the “Amplitude” field is greyed out and the “Offset” field determines the size of the output voltage.
- For a range of DC voltages, plot a graph of the resulting binary number displayed on the VU matrix. Use the results to determine the resolution of the ADC.

Worksheet 4
Digital-to-analogue conversion

MATRIX | SYSBLOCKS

Systems and Signals

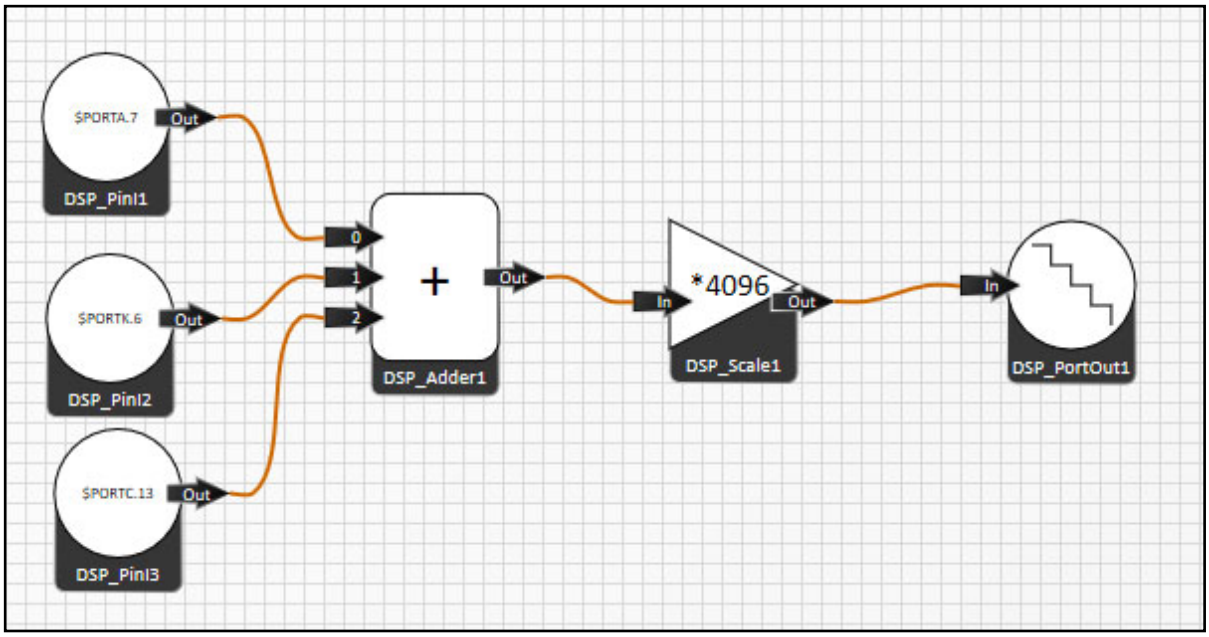
The outside world is a largely analogue environment - louder / quieter sounds, brighter / dimmer lights etc.

Microprocessors, and much of modern electronics, are digital with quantities usually expressed using binary numbers, '0's and '1's.

Conversion between these is done using a DAC.

First program

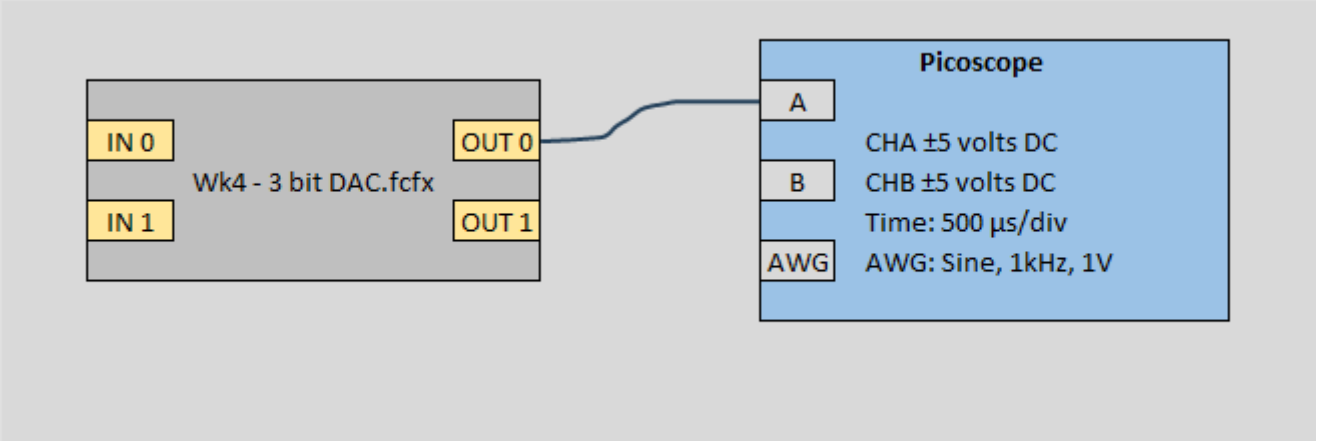
The first program uses the three switches on the Sysblocks board to input a 3-bit digital number. The analogue equivalent appears at output 'OUT0'.



Project settings:

Firmware	Wk4 - 3-Bit DAC.fcfx
Inputs	'SW2' - most-significant bit (2)
	'SW0' - bit (1)
	'SW1' - least-significant bit (0)
Outputs	'OUT0' - result of digital-to-analogue conversion
DSP Interrupt	100 kHz
Controls	None
Indicators	'LED1' is a 'heartbeat' showing that the main loop is running.

Hardware and set up:



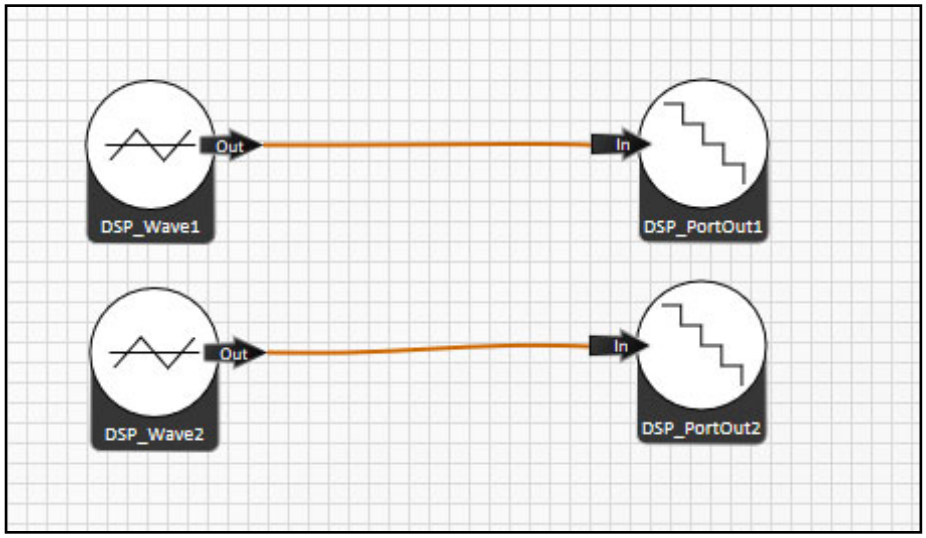
Over to you:

- The program allows you to input a 3-bit digital number, via the switches on the Sysblocks board, converts this into an analogue voltage and outputs it to output ports ‘OUT0’. View the result on the oscilloscope, running a slow-timebase setting and watch the effect of operating the switches.
- Measure the quantisation voltage (voltage step) generated by the program.
- Complete a table showing the analogue output voltage for each value of digital input.

Second program

In the next program, two DSP ’Waveform Generators’ create triangular waveforms at different frequencies. These are sent via two DAC channels and can then be viewed on an oscilloscope.

These signals are quantised in both voltage and in time. The quantisation in time reflects the sampling rate for the signal. The quantisation in voltage is the result of two different processes - the sample steps in the DSP generator component and the resolution of the digital oscilloscope.

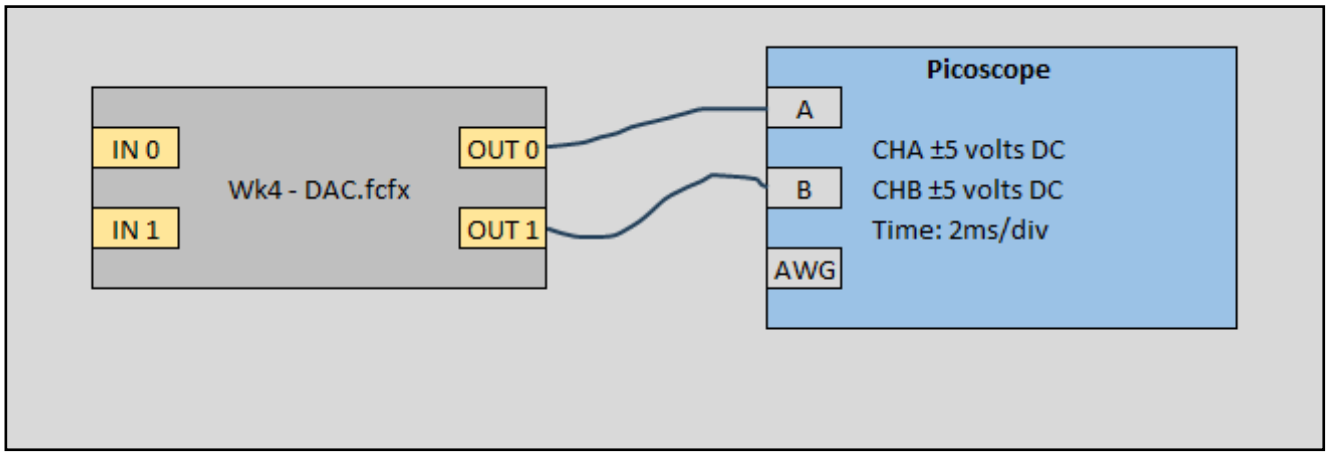


Worksheet 4
Digital-to-analogue conversion

MATRIX | SYSBLOCKS

Systems and Signals

Hardware and set up:



Project settings:

Firmware	Wk 4 - DAC.fcfx
Inputs	None
Outputs	OUT0 = Generated Wave 1
	OUT1 = Generated Wave 2
DSP Interrupt:	100kHz
Controls	None
Indicators	LED1 is a ‘heartbeat’ showing that the main loop is running.

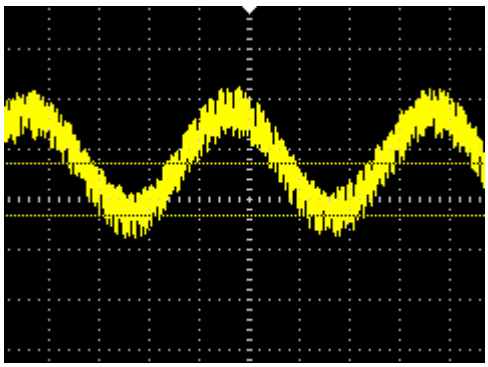
Over to you:

- In this program, two DSP ’Waveform Generators’ output triangular waveform signals having different frequencies at the output ports. View them on the oscilloscope.
- Both show quantisation steps, both in voltage and in time. Time quantisation reflects the sampling rate used. The quantisation in voltage is the result of two effects, the sample steps in the DSP generator component and the resolution of the digital oscilloscope.
  Use the oscilloscope cursors to measure these quantisation steps, in voltage and in time. (Hint - a better method? - measure ten steps and then divide by ten.)
- What would be the effect of using an 8-bit DAC rather than 16-bit device?

Worksheet 5
Noise

MATRIX | SYSBLOCKS

Systems and Signals



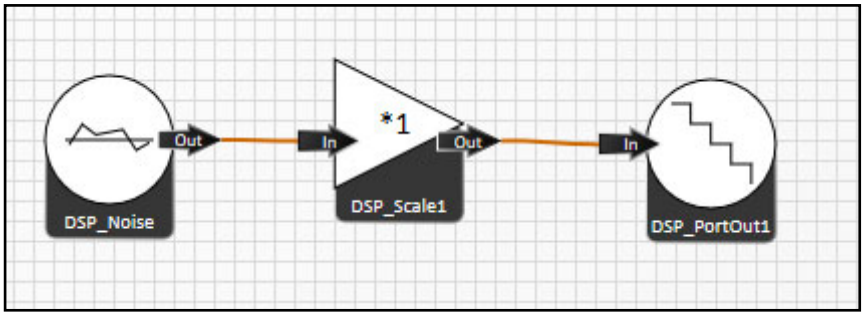
Electrical noise is everywhere:

- noise from the electricity mains supply;
- noise from electrical motors;
- noise from the Sun;
- noise from fluorescent lights;
- noise from lightning strikes.

Sometimes, there is so much electrical noise that the signal itself is indistinguishable from it.

About the program

In this program, a DSP Wave Generator uses random numbers to create a ‘white’ noise signal. ‘White noise’ describes a noise signal that has equal intensity across all frequencies produced by the generator. It is used in applications such as the production of electronic music and in measuring impulse response, the response of a system, such as an amplifier to a sudden ‘step’ input signal.



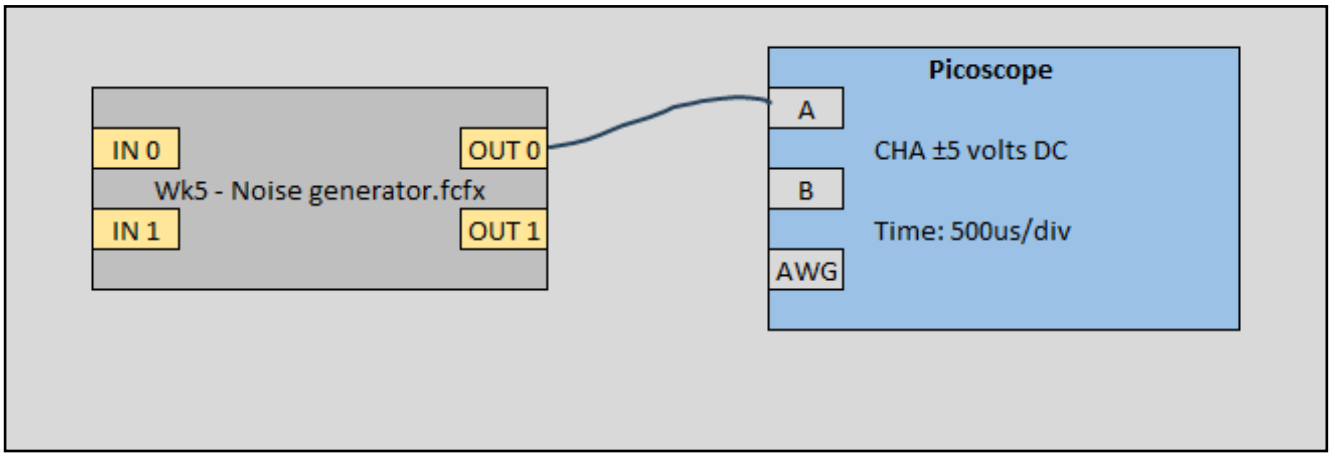
Project settings:

Firmware	Wk 5 - Noise generator.fcfx
Inputs	None
Outputs	‘OUT0’ - generated noise signal
DSP Interrupt	44.1 kHz
Controls	‘ENC0’ selects the noise level.
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Worksheet 5
Noise

Systems and Signals

Hardware and set up:



Over to you:

- View the oscilloscope trace of the signal from ‘OUT0’.
- Turn the encoder ‘ENC0’ to increase the noise signal intensity.
- As you do so, look at the effect on the oscilloscope trace.
- Add a spectrum analyser and look at the way the intensity of the noise signal is independent of frequency.
- Use the encoder to increase the noise signal and watch the effect on the spectrum.

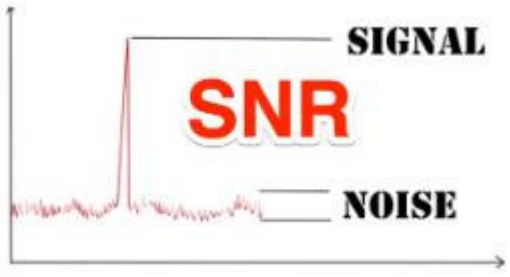
Challenge

- Plug in headphones or a speaker to the ‘LINE OUT’ socket. Listen to the effect of increased noise.

Worksheet 6
Signal-to-noise ratio

MATRIX | SYSBLOCKS

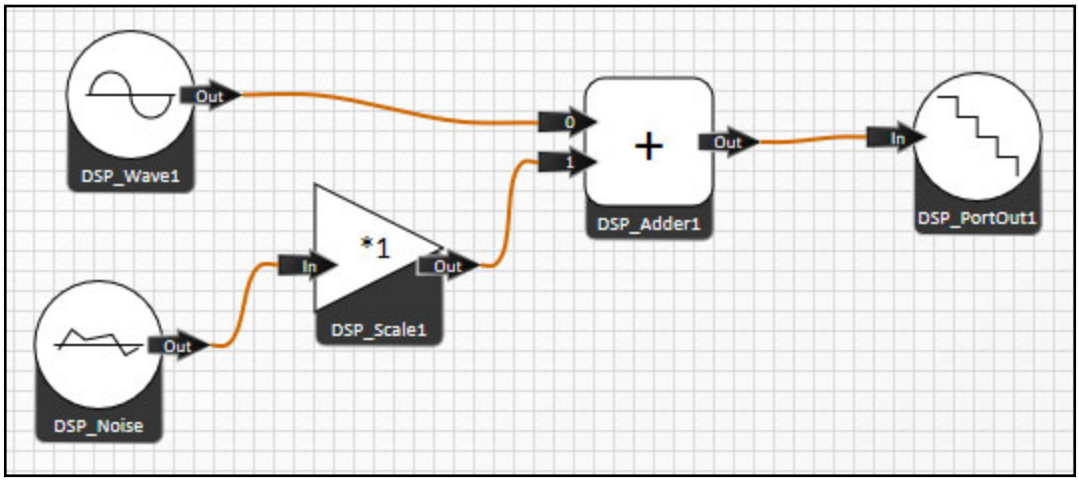
Systems and Signals



Holding a conversation in a noisy environment, like a bar, can be difficult. The signal - your conversation - can be drowned in the cacophony of surrounding sounds. Similarly in electronic communications, the signal, the information we are trying to communicate, is sometimes indistinguishable from electrical noise.
One advantage of digital signals is that they are more immune to noise than analogue signals.

About the program

In this program, one DSP 'Waveform Generator' creates a sine wave signal and the other 'white' noise. The noise is then scaled and added to the sine wave. The result is then sent to the output port. The goal is to see what a signal looks like with different levels of added noise.



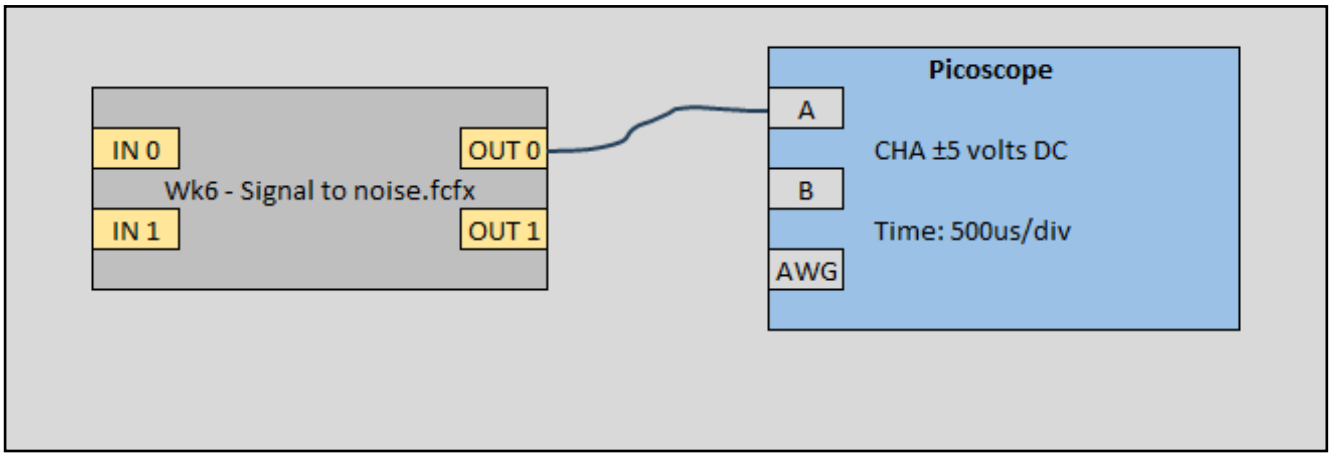
Project settings:

Firmware	Wk 6 - Signal_to_noise.fcfx
Inputs	None
Outputs	'OUT0' - generated signal
DSP Interrupt	44.1 kHz
Controls	'ENC0' selects the noise level.
Indicators	'LED1' is a 'heartbeat' showing that the main loop is running.

Worksheet 6
Signal-to-noise ratio

Systems and Signals

Hardware and set up:



Over to you:

- View the oscilloscope trace of the signal from ‘OUT0’.
- Turn ‘ENC0’ to increase the noise As the noise level is increased, look at what happens to the signal. As the noise increases, the trigger will jitter and the wave will be less stable on the display.
- Add a spectrum analyser and watch what happens as the noise level is increased. The sharp peak represents the sine wave signal. The remainder represents the noise.
  As the noise level is increased, the peak remains unaltered while the noise ‘floor’ rises across the whole spectrum.

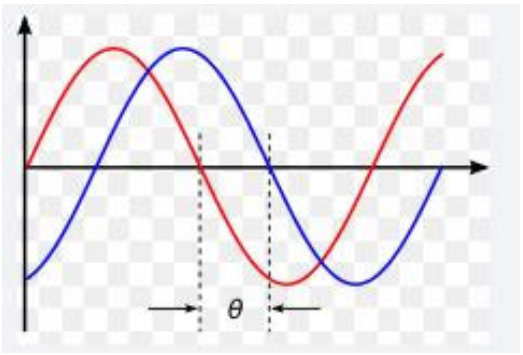
Challenge

- Plug in headphones or a speaker to the ‘LINE OUT’ socket. Listen to the effect of increased noise.

Worksheet 7
Phase

MATRIX | SYSBLOCKS

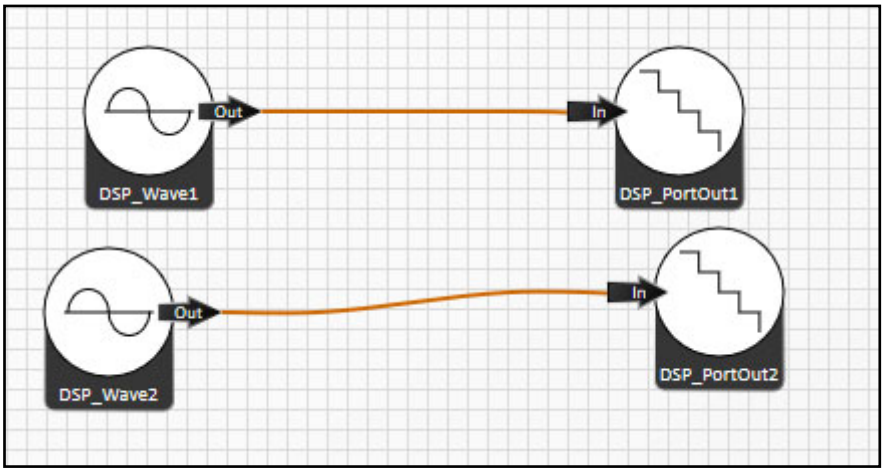
Systems and Signals



A sinusoidal signal increases in strength positively, then decreases, increases negatively and rises again before repeating the whole process.
A second sinusoidal signal may be ‘in step’ with the first, in which case we say that they are **in phase**. They may be ’out of step’ in which case we say that there is a **phase difference** between them.
Phase difference is usually expressed as an angle.

About the program

The DSP ’Waveform Generators’ create sine wave signals, having identical frequencies and amplitudes but with different phase angles. The results are sent to the two output ports.
The aim is to use an ’X-Y’ plot to identify the phase angle between the signals.



Project settings:

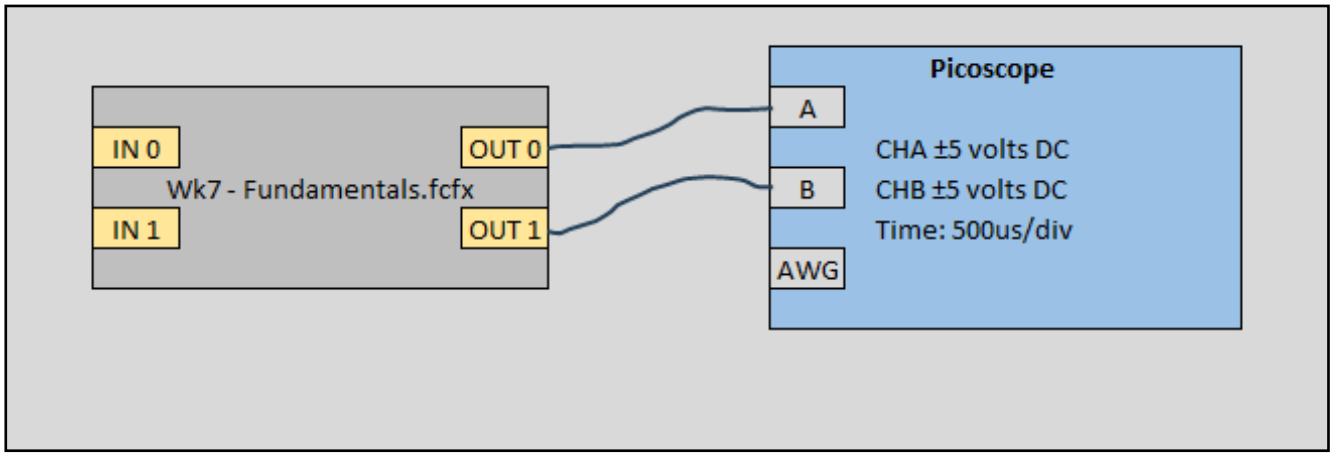
Firmware	Wk 7 - Fundamentals.fcfx
Inputs	None
Outputs	‘OUT0’ - generated signal A
	‘OUT1’ - generated signal B
DSP Interrupt	44.1 kHz
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Worksheet 7
Phase

MATRIX | SYSBLOCKS

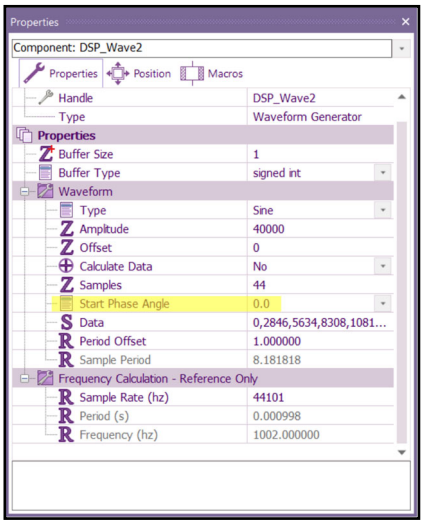
Systems and Signals

Hardware and set up:



Over to you:

- View the oscilloscope trace of the signal A (from 'OUT0').
- Add a spectrum analyser and look at the spectrum of the signal. What does this show?
- Do the same for the signal B, from 'OUT1'. Initially, this has a phase (relative to the signal A) of 90^0.
- Now add a 'X-Y' plot view. (sometimes called a Lissajous figure.) With a phase shift of 90^0 this plot is a circle (if the scales on the axes are identical).
- Double-click or right-click on the 'Wave Generator2' icon in the 2D panel and open its properties.
- Change the 'Start Phase Angle' setting to 0^0.
- Save the program and compile it to the target (under 'Build').
- Look at the effect on the X-Y plot.
- Try other phase angles and test the effect in the same way.



Challenge

- Investigate the effect of changing the frequency of signal B.
  To do so, open the 'Properties' panel for 'Waveform Generator 2' again but this time make a small adjustment to the 'Samples' setting. Notice that the calculated frequency, at the bottom of the panel, changes.
- As before, save the program and compile it to the target.
- Look at the effect on the X-Y plot. Changing the relative phase distorts the circle. Making small frequency changes makes the circle spin. Larger frequency differences create other shapes.

Worksheet 8
Manipulating signals

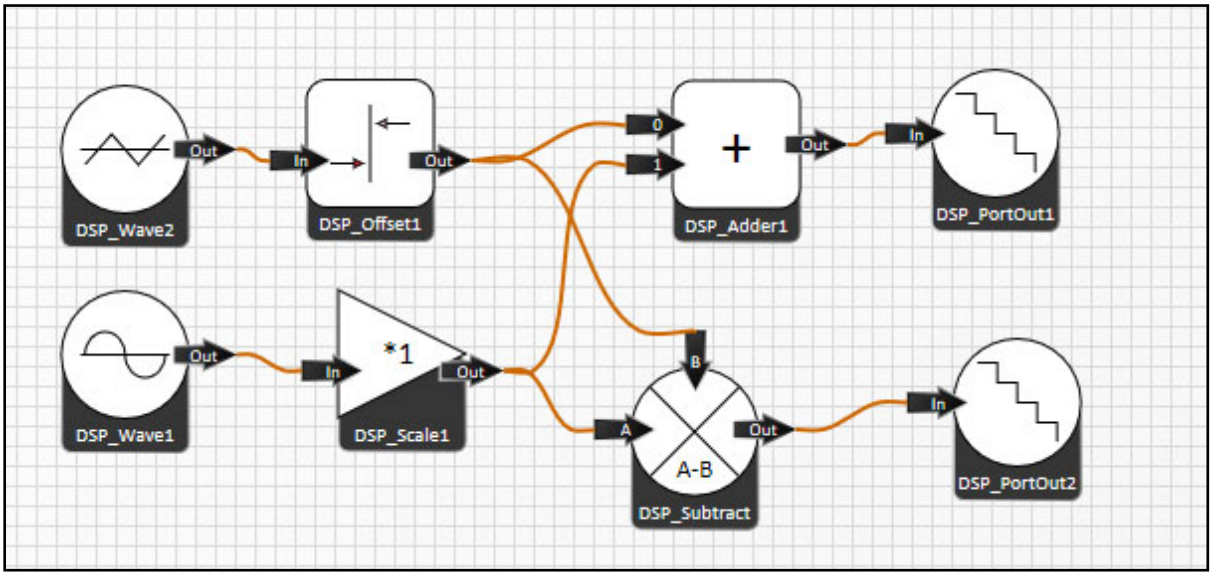
MATRIX | SYSBLOCKS

Systems and Signals

In electronic systems, signals are often combined by some mathematical operation - addition, subtraction, multiplication, integration .... - to give a new signal with different properties.
For example, integration is central to Fourier transforms and convolution. Closed-loop control systems generate an error signal by subtracting the feedback signal from the setpoint signal.

About the program

Signal processing techniques are common in a wide range of electronic applications, including telecommunications, audio and video processing, image processing, speech recognition, and control systems. This program generates two signals - one triangular and the other sinusoidal, having different frequencies. Using the encoders, the amplitude of the sinusoidal signal and the DC offset of the triangular signal can be adjusted. Sum and difference signals, simple examples of manipulation, appear at outputs ‘OUT0’ and ‘OUT1’ respectively.

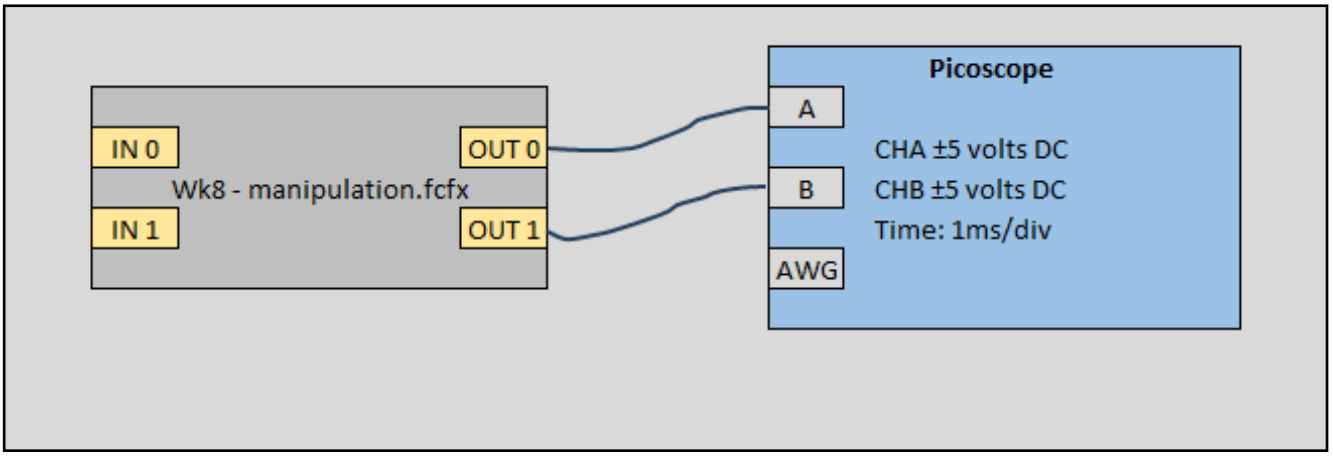


Project settings:

Firmware	Wk 8 - Manipulation.fcfx
Inputs	None
Outputs	‘OUT0’ - (sinusoidal + triangular) signal
	‘OUT1’ - (sinusoidal - triangular) signal
DSP Interrupt	44.1 kHz
Controls	‘ENC0’ changes the amplitude of the sinusoidal signal.
	‘ENC1’ changes the DC offset of the triangular signal.
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Worksheet 8
Manipulating signals

Hardware and set up:



Over to you:

- View the individual oscilloscope traces of the signals and the 'X-Y' plot view.
- Observe the changes occurring when you adjust the amplitude of the sinusoidal signal and the DC offset of the triangular wave.
- Eventually, increasing these quantities leads to 'wrap-around'. Notice the effect this has on the traces.
  Both waveform generators use the 'signed int' data type. In due course, the buffer storing the data reaches its maximum value (32767) and 'wraps around' (restarts).
  In other words, 32767 + 1 ➔ -32768! Note that reducing either the amplitude or DC offset will remove it.

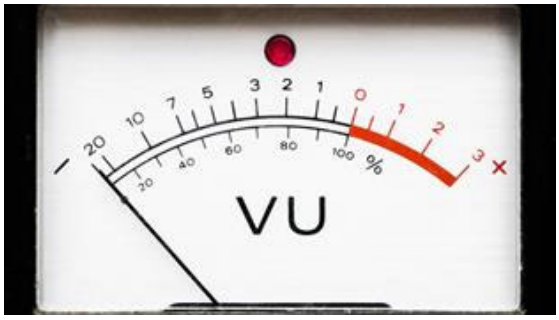
Challenge

- Investigate the effect of changing types of wave generated.
- Change the arithmetic operation and observe the effect.
- Replace one of the waveform generators with an ADC input block and hence manipulate a signal from an external signal generator. (Experiment with AC and DC coupling.)

Worksheet 9
Level detection

MATRIX | SYSBLOCKS

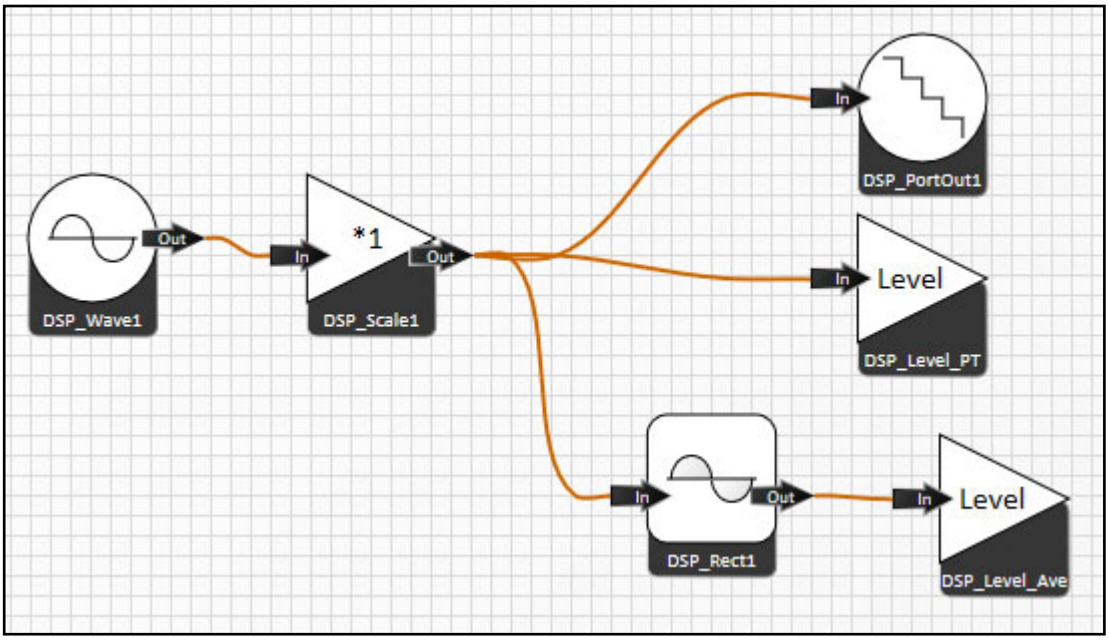
Systems and Signals



Measuring electrical signals is a vital part of a variety of applications, from audio processing, wireless communication, spectrum analysis, power generation and even measuring seismic activity.

About the program

This program generates a sinusoidal signal and then calculates its average, peak and trough voltage values.



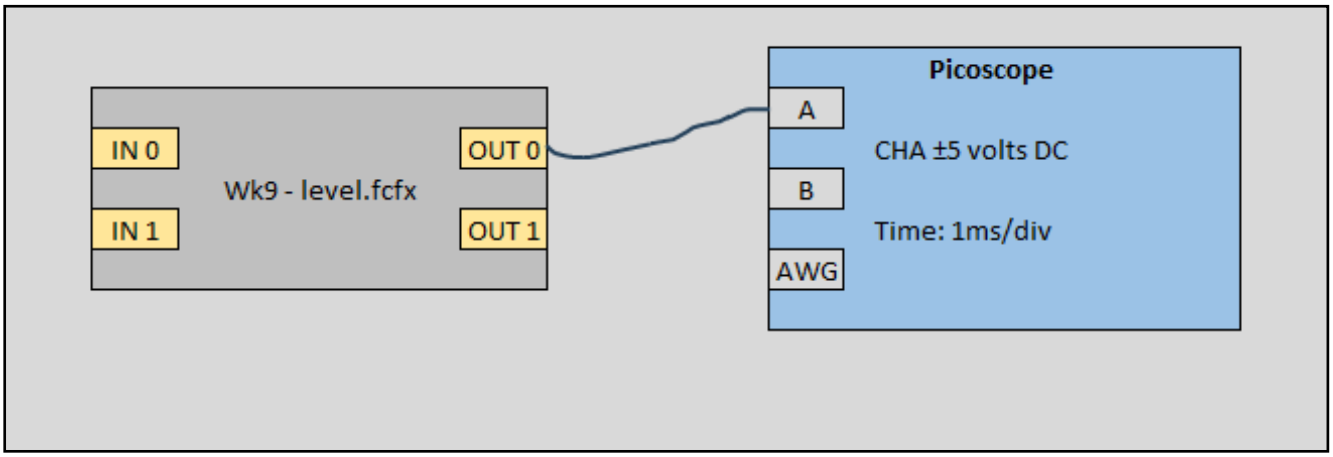
Project settings:

Firmware	Wk 9 - Level.fcfx
Inputs	None
Outputs	‘OUT0’ - generated signal
DSP Interrupt	44.1 kHz
Controls	‘ENC0’ sets the scale (amplitude) of the signal
	‘ENC1’ selects which measurement is displayed
	‘SW0’ resets peak and trough readings when scale is reduced
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Worksheet 9
Level detection

Systems and Signals

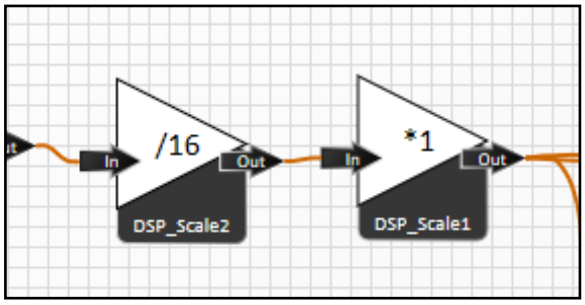
Hardware and set up:



Over to you:

- Look at the signal generated by the program on the oscilloscope trace (from 'OUT0').
- Notice the effect of increasing the 'scale' setting on encoder 'ENC0'.
- Turn encoder 'ENC1' and view the peak, trough and average voltage values for the signal.
- The 'DSP_Level_Ave' component generates the average voltage value of the signal. Why is there a DSP rectifier block before it?
- Notice that reducing the scaling does not automatically reduce the peak and trough readings. Press switch 'SW0' to correct them.
- Use the oscilloscope to measure maximum, minimum, peak-to-peak, DC average, AC rms and true rms values for the output signal. Compare them to the measurements presented on the SysBlocks display.
- Double-click or right-click on the wave generator icon in the 2D panel and open its properties. Change the properties to experiment with other types of wave.
- Observe the effect of altering its offset on the peak and trough voltage values.

Challenge

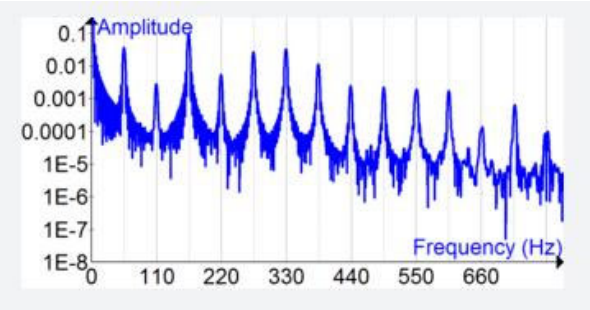


- Load the program 'Wk 9 - LevelExt'.
  In this, a fractional scaler has been created, dividing the signal by 16 and then multiplying by a value from 0 to 16 determined by the encoder. This effectively gives scaling in steps of 16ths.
- Use an external signal source such as the Picoscope AWG to input a sinusoidal signal with a frequency of 1kHz and amplitude 1V.
- Repeat the steps shown above using this signal source.
- Relate the parameters of the applied signal to the SysBlocks level measurements and then the Picoscope measurements of the output signal.

Worksheet 10
Fourier analysis

MATRIX | SYSBLOCKS

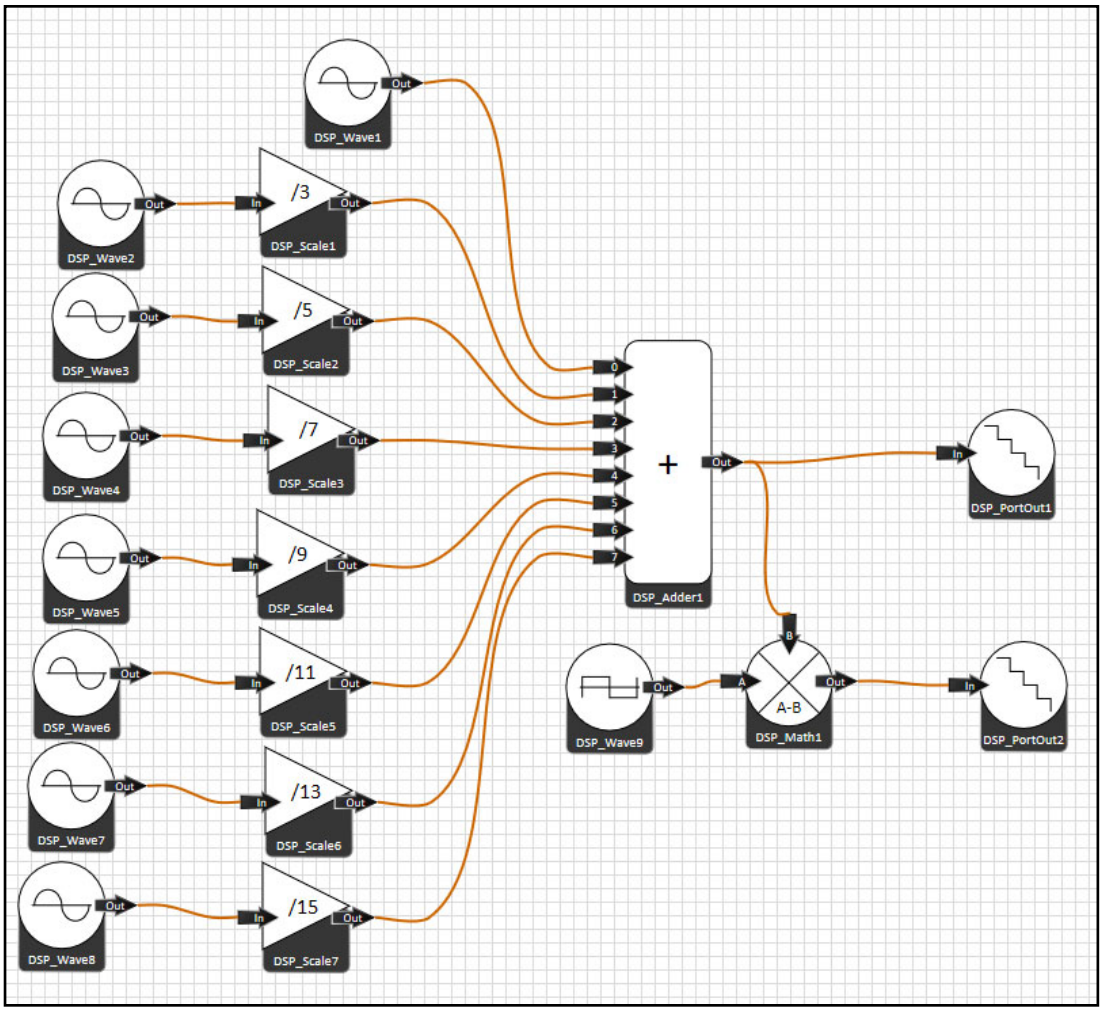
Systems and Signals



According to Fourier, any repetitive signal can be thought of as a series of sinusoidal signals, having appropriate amplitudes, frequencies and phases. For example, a 100 Hz square wave is the sum of a 100 Hz sinusoid (the 'fundamental') and its odd 'harmonics' - a 300 Hz sinusoid with 1/3rd of its amplitude, a 500 Hz sinusoid with 1/5th amplitude, a 700Hz sinusoid with 1/7th amplitude and so on.

About the program

This firmware adds the first seven odd harmonics to sinusoidal signal (the fundamental) to build up a square wave. It also outputs a signal showing the difference between the constructed wave and an ideal square wave.



Worksheet 10
Fourier analysis

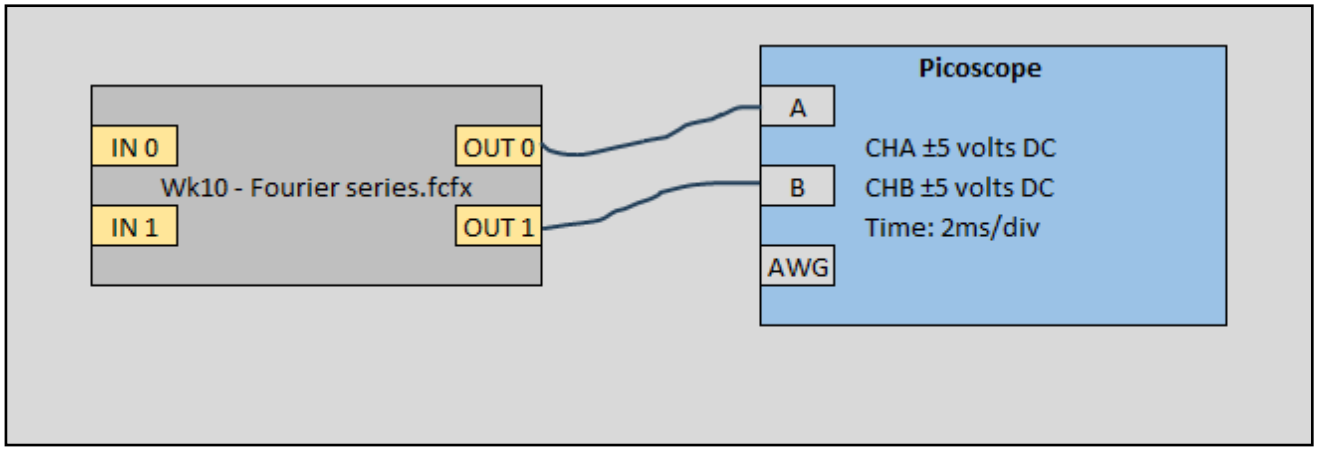
MATRIX | SYSBLOCKS

Systems and Signals

Project settings:

Firmware	Wk 10 - Fourier_Basic.fcfx
Inputs	None
Outputs	'OUT0' - sum of fundamental plus harmonics
	'OUT1' - difference between sum and generated square wave
DSP Interrupt	150 kHz
Controls	None
Indicators	'LED1' is a 'heartbeat' showing that the main loop is running.

Hardware and set up:



Over to you:

- Look at the two output traces, from 'OUT0' and 'OUT1', on the oscilloscope.
- Notice that the 'sum' output (from 'OUT0') is roughly a square wave but is far from precise.
- Even adding 7 harmonics to the fundamental is not enough to create a good representation of a square wave.

Challenge

- Load the program 'Wk10 - Fourier_Series.fcfx'.
  This performs the same function as the previous one, but adds the ability to control how many harmonics are added (up to a maximum of 7)
- When the software starts, it is showing a single sine wave. Rotate the encoder to add more harmonics to the output wave. Note that each harmonic brings the output a little closer to a square wave.
- Use the Picoscope's 'AC RMS' measurement on channel B to see that the error is reduced at each step. (Allow a few moments after each adjustment for the reading to settle.)

Advanced

- Using the program 'Wks1_Fourier_Basic.fcfx', construct other signal waveform shapes. To do so, change the 'Initial Integer Scaler' property for each of the scale components.

Worksheet 11
Fourier basics

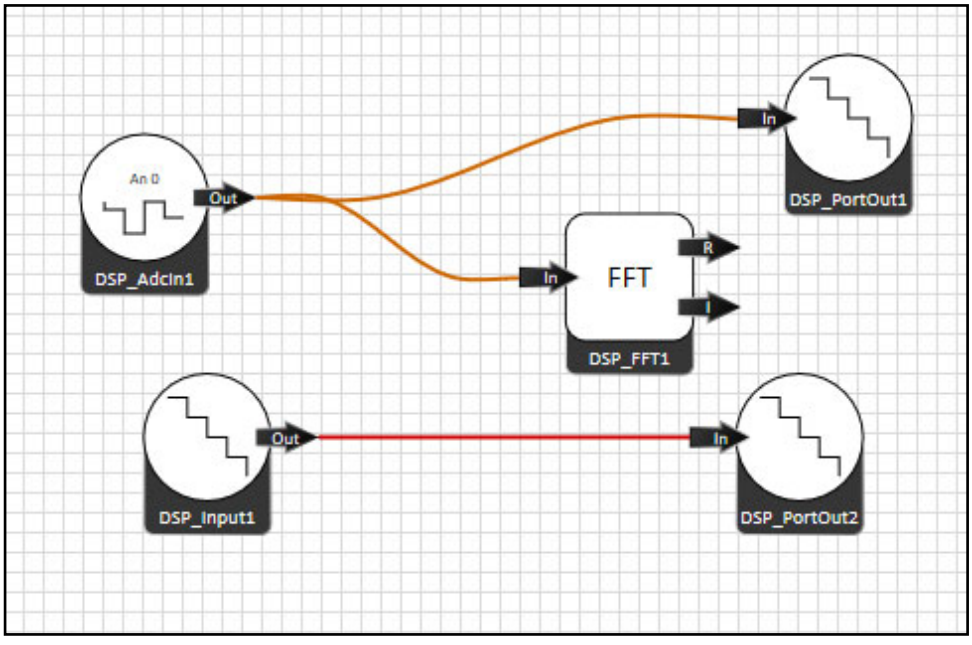
MATRIX | SYSBLOCKS

Systems and Signals

Using Fourier transformation to convert vibrations into a frequency spectrum means that;
noise ‘crackle’ can be removed from audio recordings;
buildings can be designed to withstand earthquakes;
computer data can be filtered, discarding less-important data from the original to reduce file size. (This is why JPEG and MP3 files are much smaller than .raw, .bmp or .wav files).

About the program

The program reads the signal presented at input ‘IN0’ and reproduces it at output ‘OUT0’.
At the same time, it performs a Fast Fourier Transform, (FFT) on that signal. That gives us an array of 256 data points, each representing the intensity of the signal in a particular frequency bin (interval), 100Hz ‘wide’. In all then, the process covers a frequency range 0 to 25.6kHz.
Data points are outputted one at a time. The oscilloscope assembles them as a frequency spectrum - a histogram of intensity (vertical axis) against frequency (horizontal axis).
(The horizontal axis may appear to show time, having a scale in ms. In reality, this is because the first data point is outputted and plotted and then, 20μs later, the next is plotted, and so on.)
The signal sample rate of 51.2kHz gives a Nyquist frequency limit of 25.6 kHz.
The first frequency bin has been overwritten with a large peak to make it easier to trigger the oscilloscope.



Worksheet 11
Fourier basics

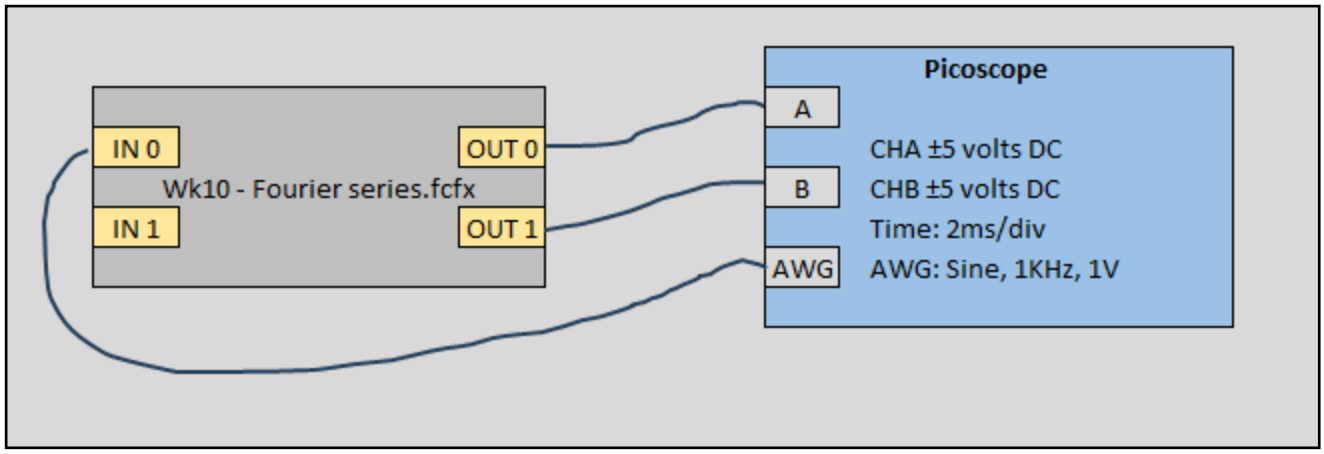
MATRIX | SYSBLOCKS

Systems and Signals

Project settings:

Firmware	Wk 11 - FFT.fcfx
Inputs	‘IN0’ - input signal to be sampled (AC coupled)
Outputs	‘OUT0’ - sampled signal
	‘OUT1’ - Fourier transform of waveform
DSP Interrupt	51.2 kHz
Controls	None
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Hardware and set up:



Over to you:

- Use the Picoscope AWG, or equivalent, to input a 1kHz, 1V sinusoidal signal to input ‘IN0’.
- Set the oscilloscope to trigger on channel B at a level of 2.5V.
- Set the ‘Pre-trigger’ level to 0% to get rid of negative time values.
- Observe what happens to the spectrum displayed when the AWG frequency is changed in the range 100Hz up to 12kHz.
- Change the AWG signal to a square wave and notice the effect on the spectrum.
- Experiment with other waveforms.

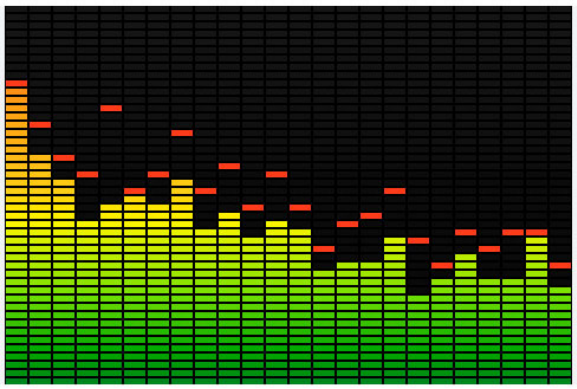
Challenge

- If one is available, use a SysBlocks mixer board to combine two signals.
- Examine the resulting spectrum.

Worksheet 12
Digital filter

MATRIX | SYSBLOCKS

Systems and Signals

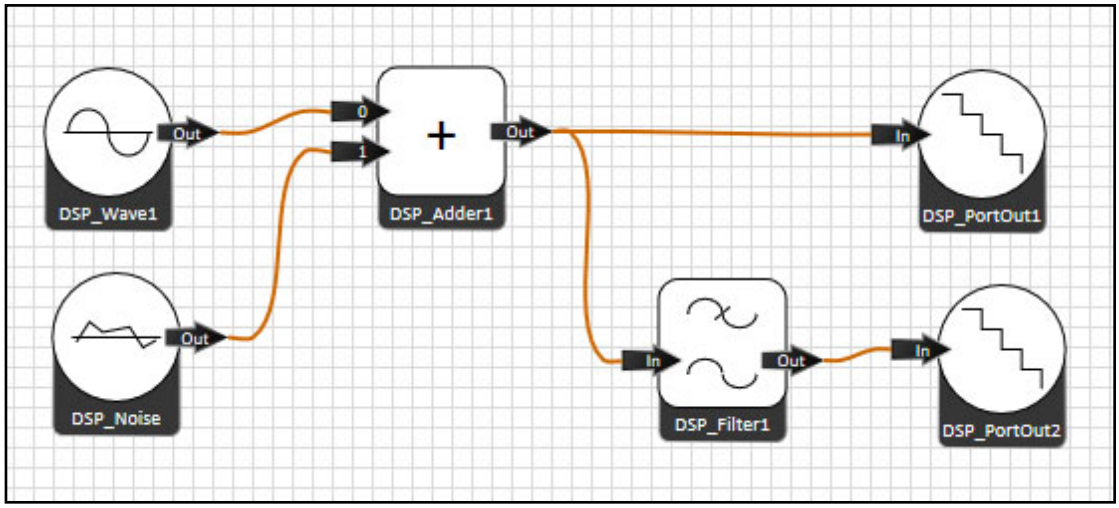


Filtering is one of the biggest benefits of DSP. Digital filters can deliver a performance beyond that achievable using analogue components.
For the digital filter:

- data stored in memory replace the physical components of the analogue filter;
- behaviour can be changed easily by modifying the software.

About the program

The firmware adds noise to a sinusoidal signal. This resulting signal appears at output ‘OUT0’. A DSP filter block, configured as a low-pass filter with a cut-off of 2 kHz, filters the result and it appears at output ‘OUT1’.



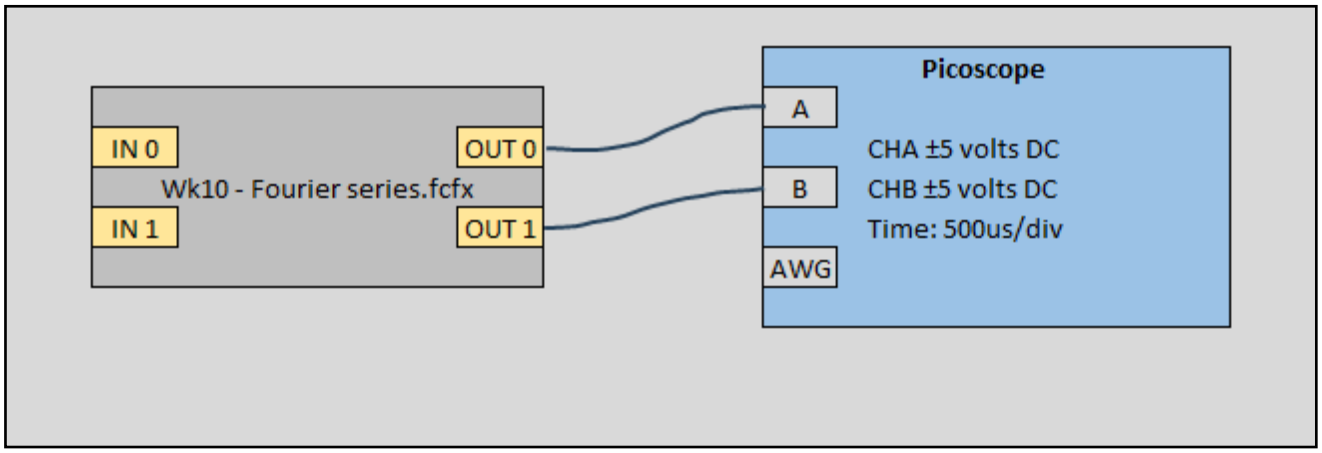
Project settings:

Firmware	Wk 12 - Digital_Filters.fcfx
Inputs	None
Outputs	‘OUT0’ - generated signal plus noise
	‘OUT1’ - filtered signal
DSP Interrupt	100 kHz
Controls	None
Indicators	‘LED1’ is a ‘heartbeat’ showing that the main loop is running.

Worksheet 12
Digital filter

Systems and Signals

Hardware and set up:



Over to you:

- Examine the oscilloscope traces of the signals generated by the program. The ‘OUT0’ signal shows the problem - signal plus noise. The ‘OUT1’ signal from shows the result of filtering.
- Open a ‘spectrum view’. The spectrum of the ‘raw’ signal includes significant noise, whereas the filtered signal is much ‘cleaner’.
- Modify the type of filter and notice the effect.
- Change the parameters of the filter and notice the effect.

Challenge

- Replace the ‘Waveform Generator’ component with an ‘Input ADC’ component.
- Try filtering external signals and view the results.

Version control

14 12 23 First release