Sysblocks

Music with microcontrollers

MATRIX

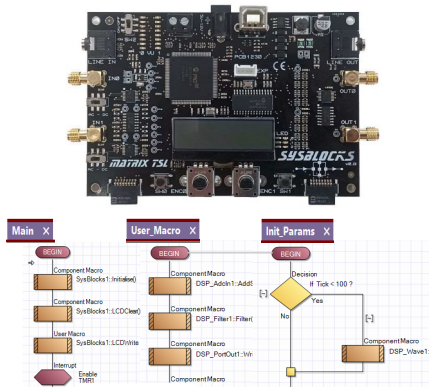
CP1210

Contents

Worksheet 1	The ‘main’ program	3
Worksheet 2	Program 1: In and Out	5
Worksheet 3	Program 2: The ‘click’ generator	8
Worksheet 4	Program 3: Control the Output	10
Worksheet 5	Program 4: Control the Tone	12
Worksheet 6	Program 5: Anatomy of an Echo	14
Worksheet 7	Program 6: Multiple Echoes	16
Worksheet 8	Program 7: Reverberation	18
Worksheet 9	Program 8: Mixing signals	20
Worksheet 10	Program 9: Equalisation	23
Worksheet 11	Program 10: Sampling	25
Version control		27

Worksheet 1
The ‘Main’ Program

Music with microcontrollers



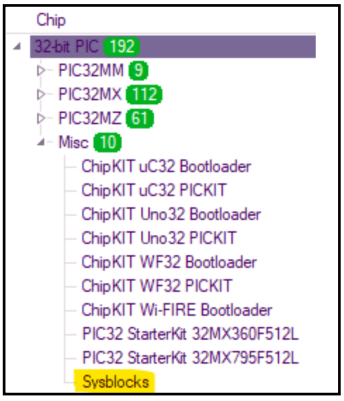
Sysblocks relies on a powerful PIC32 processor with A/D and D/A interfaces. Its signal processing power makes it an ideal tool for teaching programming and music technology.

Flowcode is a graphical programming environment that allows you to create programs by linking ‘drag_and_drop’ icons on the programming workspace.

Flowcode programs consist of a ‘Main’ program and a number of associated macros.

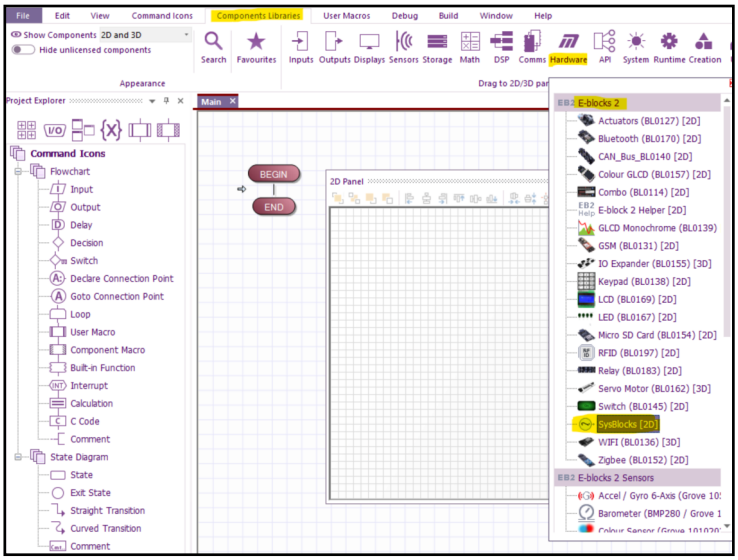
The ‘Main’ program:

This part of the flowchart, described here, stays the same for most programs in this module.

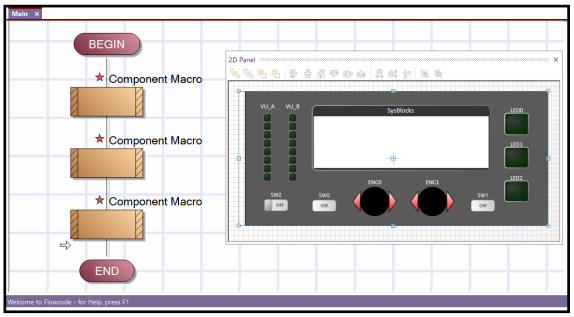


Over to you:

- Open FlowCode ***10*** and start a new embedded project.
- Choose SysBlocks as the target, (found under ‘32-bit PIC->Misc’ as shown opposite).
- In ‘Components Libraries’, locate the ‘Hardware’ menu and add a SysBlocks component to the 2D panel.



- In the ‘Main’ section of the flowchart, add three ’Component Macros’, found in the ‘Command Icons’ tab.

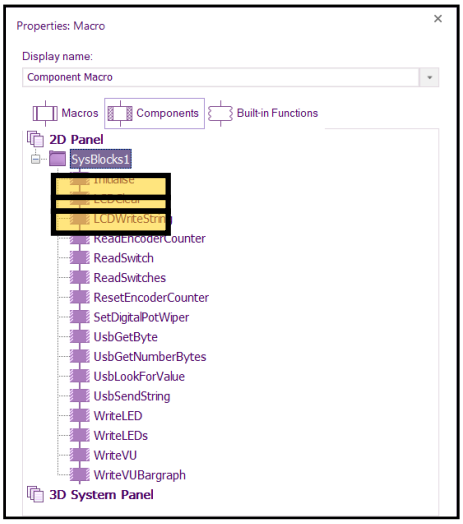


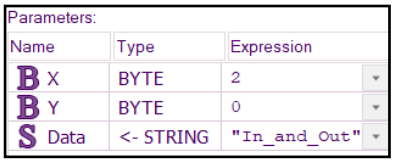
Worksheet 1
The ‘Main’ Program

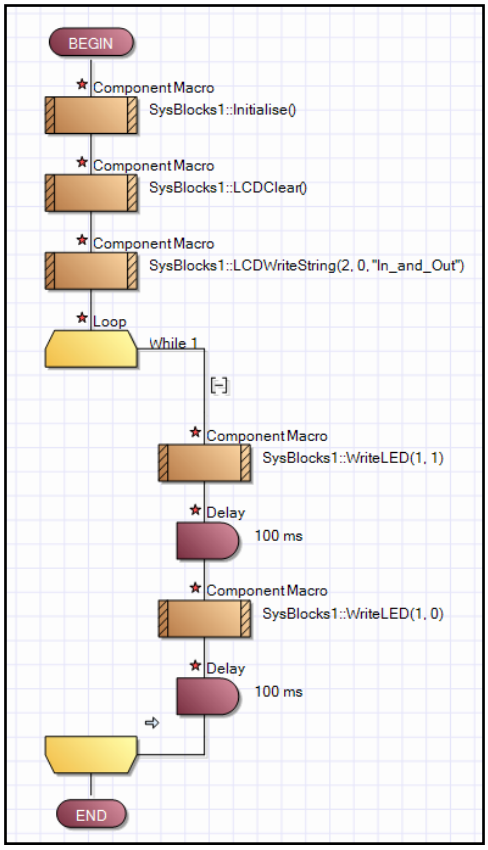
Music with microcontrollers

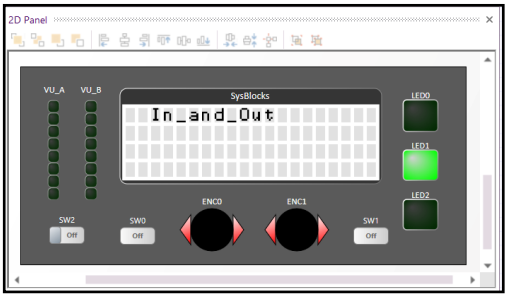
The ‘Main’ program .....

- Double-click on the first component macro to open its ‘Properties Macro’ panel.
- Click on the ‘+’ next to ‘Sysblocks1’.
- Select ‘Initialise’ from the available options.
- In the same way, choose the ‘LCDClear’ function for the second component macro and the ‘LCDWriteString’ function for the third.
- Double-click on the ‘LCDWriteString’ component macro to open its properties.
- Add ‘X’ and ‘Y’ co-ordinates to specify where data should be placed on the Sysblocks LCD screen and then add the data - In_and_Out”, (the name of the first program.)
- Next add a ‘While’ loop. (The default is fine - it loops forever.)
- Inside this, add two SysBlocks ‘Write LED()’ component macros, one to turn on LED 1, the other to turn it off.
- Add a ’Delay’ command, configured to cause 100 ms delays after each.
- At this stage, the flowchart looks like that opposite.
- It can be tested by running in ‘Debug’ mode.
- Click on the ‘Debug’ tab and then on ‘Run’.
- The SysBlocks component on the 2D panel should display the text “In_and_Out” and flash the middle LED (‘LED 1’) on and off.



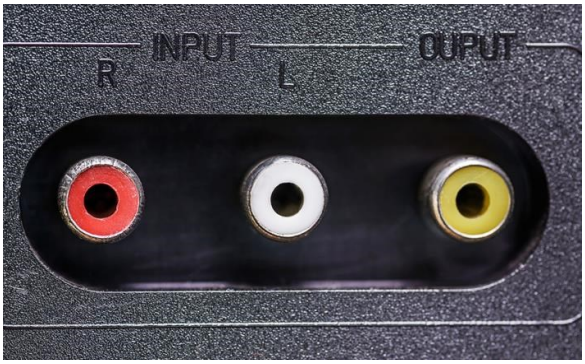






Worksheet 2
In and out

Music with microcontrollers



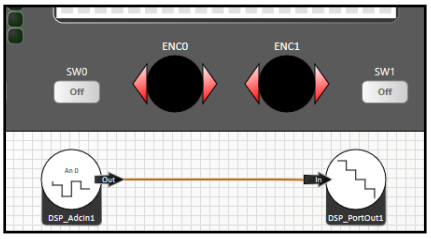
The first program simply transfers an audio signal from the input to the output of the system.

Although there is a lot of hidden processing, the output signal should be identical to the input.

This program forms a base from which others will be developed.

Over to you:

- Locate the DSP blocks, in the ‘Component Libraries’ tab.
- Add an ‘Input ADC’ and an ‘Output Port’ to the 2D panel.
- Both the inputs and outputs to the Sysblocks board have 3.5mm jack connections and SMA co-axial connectors.



There is an important difference between the input and output. The jack connection on the output, mirrors the signals sent to the SMA connectors. On the input side, however, the jack connection is independent and is connected to different input channels. These links are shown in the following table.

Direction	Connector	Port	Channel
Input	Line In	Left	An1
		Right	An2
	SMA	IN0	An0
		IN1	An3
Output	Line out	Left	PORTH
		Right	PORTJ
	SMA	OUT0	PORTJ
		OUT1	PORTH

The other difference between input and out is coupling options: for this worksheet we will be using **AC coupling** which is dictated by switches under the SMA input connectors.

It is important that the correct properties are chosen for the ‘Input ADC’ block to ensure that the input signal reaches the software.

On the input side, the SMA connectors and the jack socket are separate, providing four channels, ’An0’ to ’An3’, to deliver signals to the microprocessor. On the output side, there are just two, PORTH and PORTJ.

PORTJ is connected to a DAC, (digital-to-analogue converter) and then to parallel buffers connected to both the SMA ‘OUT0’ terminal and the right channel of ‘Line out’.
PORTH is connected to the ‘OUT1’ terminal and left channel of ‘Line out’.

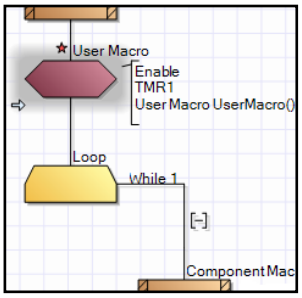
Worksheet 2
In and out

Music with microcontrollers

Over to you .....

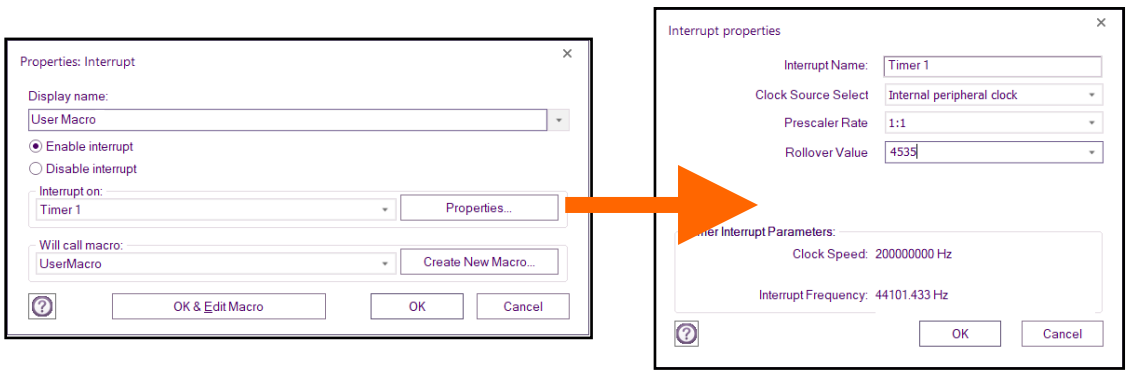
- Double-click on the ‘Input ADC’ icon on the 2D panel to open its properties:
  - Under the ’Connections’ properties, check that channel ‘AN0’ is selected.
- Double-click on the output component icon to open its properties:
  - Click on the ’Connect to’ property and select ’DSP_AdcIn1:Out’.
  - This connection is then shown on the 2D panel.
  - Set the ‘Autoscale’ property to ‘Yes’.
  - Set the ‘Port’ property to ‘Port H’, causing the blocks to write to one of the physical outputs, ‘OUT1’, on the SysBlocks board.

To service the DSP blocks, create a timer interrupt using the icon found in the ‘Command icons’ tab.
This is placed in the ‘Main’ section, of the program, just before the ‘While’ loop.

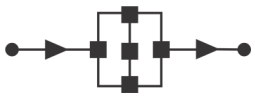


- Double-click on the interrupt icon to open its properties panel and set these as follows:
  - In the ‘Properties: Interrupt’ panel, click on the button alongside ‘Timer 1’ to open the ‘Interrupt Properties’ panel, shown below.
  - Change the rollover value to 4535, to give an interrupt frequency of around 44.1 kHz (a standard sampling rate for high quality audio).

The effect of all this is to allow one of the internal timers, ‘Timer 1’ to cause an interrupt periodically. Whenever this happens, the program runs the macro called ‘UserMacro’.



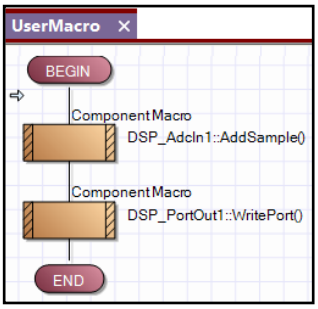
Worksheet 2
In and out



Music with microcontrollers

Over to you .....

- Back on the ‘Properties: Interrupt’ panel, click on ‘Create New Macro’ alongside the ‘UserMacro’ label.
- Add the name ‘UserMacro’ in the ‘Name of new macro’ box.
- Leave the other boxes unchanged and click on ‘OK’.
- Click on the ‘OK & Edit Macro’ button.
- A blank flowchart opens for you to create this macro.
- Add two component macros to this flowchart.
- Select the AddSample() and WritePort() options for these.
- The flowchart now resembles the one opposite:



This completes the ’In and Out’ program.

Testing the program :

- Save the program as ‘In and Out’.
- Use the ’Compile to Target’ option under the ’Build’ tab to transfer this program to the Sysblocks board.
- Connect a signal source, such as the ‘AWG’ signal generator, part of Picoscope, to input 0 (IN0) of the Sysblocks board, (since channel ‘AN0’ was selected earlier for the ‘Input ADC’ block.)
- Choose a sine wave signal with a frequency around 300Hz.
- Connect a loudspeaker or headphones to the Sysblocks ‘LINE OUT’ socket.
- Switch on the power supply.
- The Sysblocks LCD screen should display the name of the program, “In_and_Out”.
- LED1, the centre one on the Sysblocks board, should flash on and off several times each second.
- You should hear an audio signal.
- Connect an oscilloscope, such as a Picoscope, to output 1 (OUT1) of Sysblocks. You should see the waveform of the signal appearing at the output of the board. Compare it with that applied to the input.

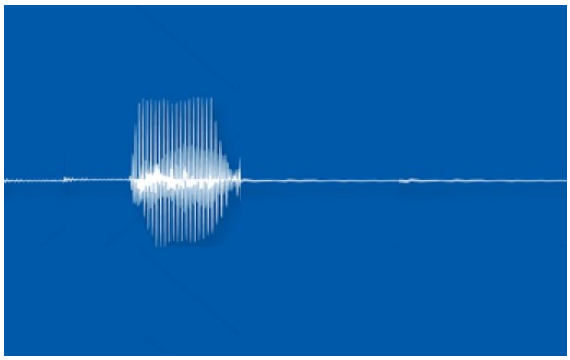
Challenge:

- Investigate the effect of changing the frequency, amplitude and shape of the signal on what you hear and see on the oscilloscope trace.
- Try playing music through the system from a source such as a smartphone or laptop. (Remember to change the ‘Input ADC Connection’ property to ‘An1’ or ‘An2’.)

Worksheet 3

The ‘click’ generator

Music with microcontrollers



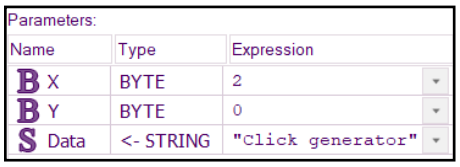
Sysblocks includes a built-in signal source component called ‘Waveform Generator’.
This outputs a signal, which will be used periodically in the course.

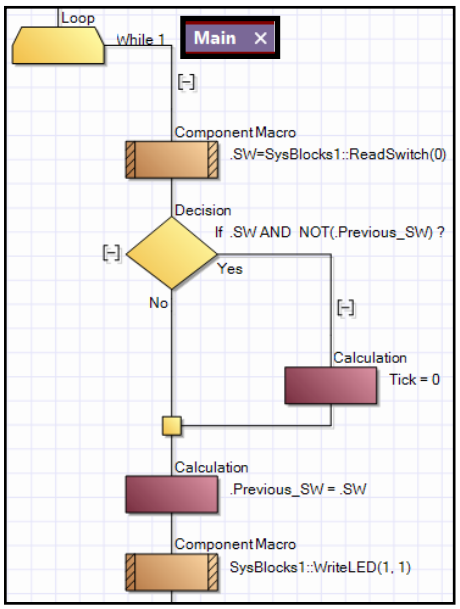
The following program shows how to use it to create a ‘click’ sound for use in testing.

Over to you:

In the ‘Main’ program:

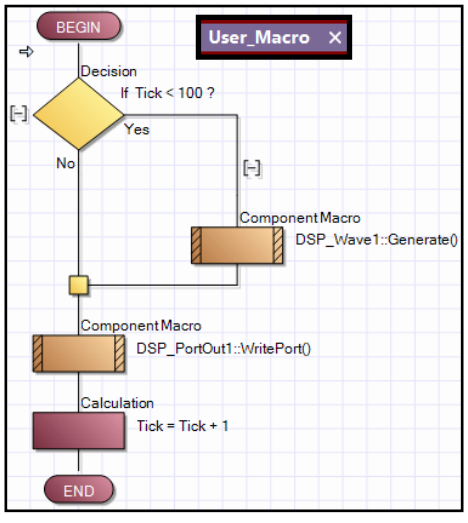
- Change the name displayed on the LCD screen by modifying the ‘LCDWriteString’ component macro properties as shown:



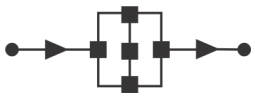


- Modify the ‘Loop’ section as shown, to check if switch ‘SW0’ on the Sysblocks board is pressed.
- The Interrupt and Timer 1 properties are the same as in the first program.
- (Note that the ‘.’ in front of a variable means that it is a local variable. You can use local or global variables here.)

- Modify the ‘User_Macro’ by adding a ‘tick’ counter, as shown opposite.
- The wave is then generated only if this ‘tick’ count is less than 100.
- Pressing switch ‘SW0’ resets the ‘tick’ count to 0.
- In this way, the duration of the signal is limited, producing a ‘click’ sound each time the switch is pressed.

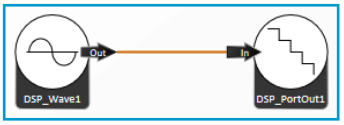


Worksheet 3
The ‘click’ generator

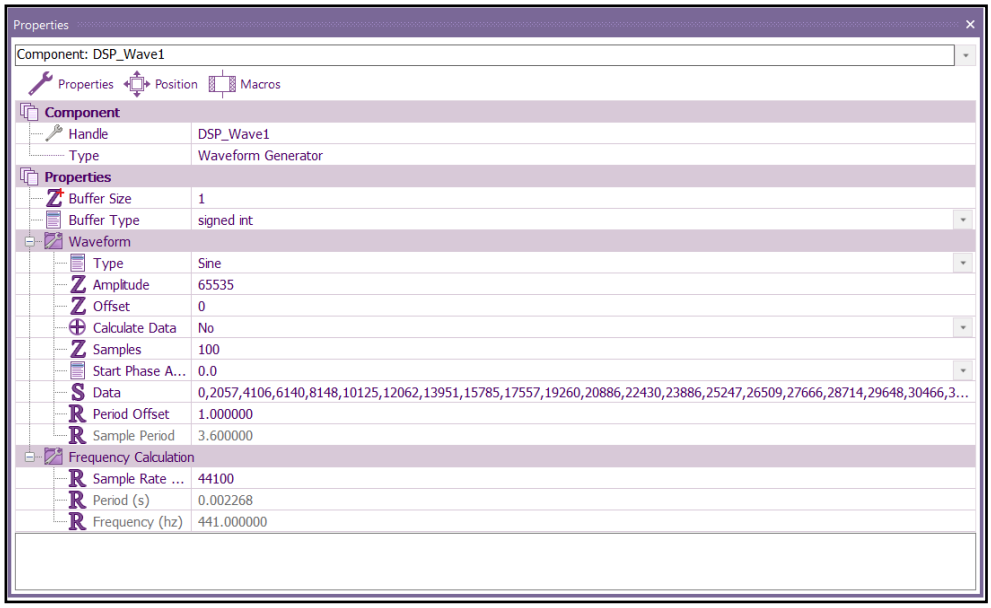


Music with microcontrollers

Over to you .....



- From the ‘DSP’ section of ‘Component Libraries’, add a ‘Waveform Generator’ and an ‘Output Port’ component to the 2D panel. Notice that, this time, there is no ‘InputADC’ block - the signal is generated internally.
- Set the properties of the ‘Waveform Generator’ as shown below:



(As soon as you select ‘Sine’ as the waveform type, the data is generated for you.)

- Set up the properties of the output port as before, but with the ‘Connect To’ property set to ‘DSP_Wave1:Out’.

This completes the program.

Testing the ‘click’ generator

- Save the program as the ‘Click Generator’ program.
- Use the ’Compile to Target’ option under the ’Build’ tab to transfer this program to the Sysblocks board.
- Connect headphones or a loudspeaker to the ‘LINE OUT’ socket of the Sysblocks board.
- Switch on the power supply.
- Press and release the Sysblocks ‘SW0’ switch.
- You should hear a brief click each time the switch is pressed.

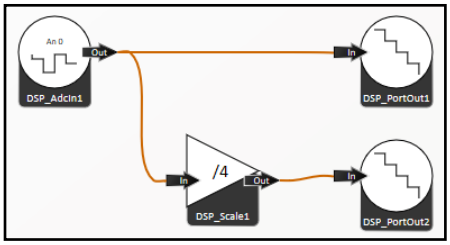
Worksheet 4
Control the output

Music with microcontrollers

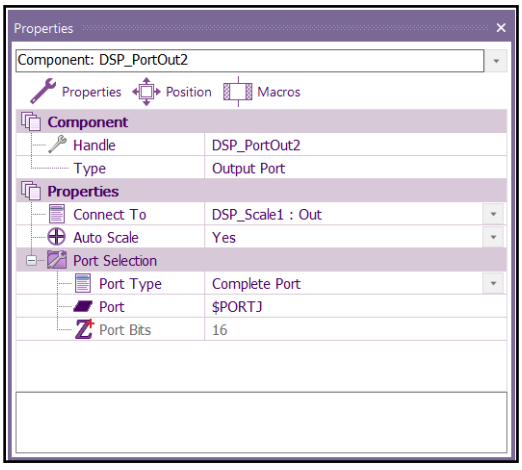
In addition to basic DSP input and output blocks, Sysblocks offers a range of operator blocks that perform a variety of mathematical transformations on the signal.
This program shows how to create a ‘volume’ control, to control the amplitude of the output signal using a DSP Scale block.

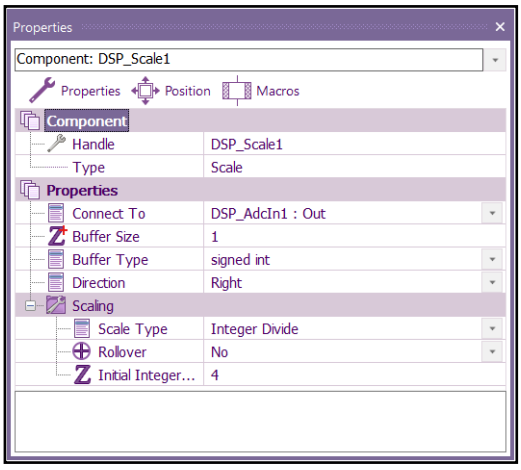
Over to you:

- Start with the set up used for the ‘In and Out’ program and add a second ‘Output Port’ and a ‘DSP Scale’ block to the 2D panel, connected as shown.



- Check that the ‘Input ADC’ block is connected to channel ’An0’.
- The first output port is configured as before.
- The images below show how to configure this second output port and the DSP Scale block.

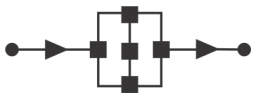




Notice that the output port is assigned to ‘$PORTJ’, i.e. ‘OUT0’ on the Sysblocks board.

The ‘Initial Integer Scaler’, set to ‘4’ initially, determines the effect on the output signal.

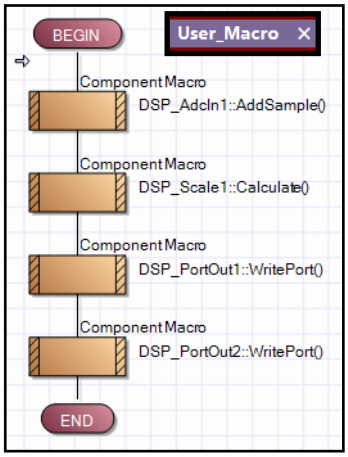
Worksheet 4
Control the output



Music with microcontrollers

Over to you .....

- The ‘Main’ program is the same as before, but change the name displayed on the LCD screen to “Control the Output.”
- Modify the ‘User_Macro’ by adding two more component macros, one for the second outputport and the other for the scale block.
- The resulting ‘User_Macro is shown opposite.



Testing the program :

- Save the program as the ‘Control the Output’ program.
- Use the ’Compile to Target’ option under the ’Build’ tab to transfer this program to the Sysblocks board.
- Connect a signal source, to input 0 (IN0) of the Sysblocks board.
- Set it to a frequency around 300Hz. Signal amplitude 600mV.
- Connect loudspeakers or headphones to the Sysblocks ‘LINE OUT’ socket.
- Switch on the power supply.
- You should hear an audio signal. Listen to each loudspeaker or headphone separately. Compare what you hear.
- Connect a dual-beam oscilloscope to display the signals from outputs ‘OUT0’ and ‘OUT1’ of the Sysblocks board. Compare the two traces.

So what?

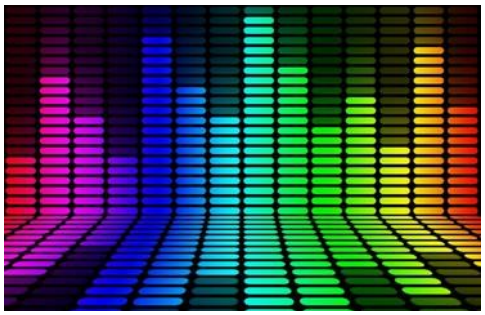
Note that with a scale factor of 1 the output is slightly larger than the input. This is because the Autoscale on the output is set to True: the input is a 12 bit ADC. The output is a 16 bit DAC. This means that the output is 16/12 bigger than the input.

Challenge:

Investigate the effect of changing the ’Initial Integer Scaler’ property on what you hear and see on the oscilloscope traces.

Worksheet 5
Control the tone

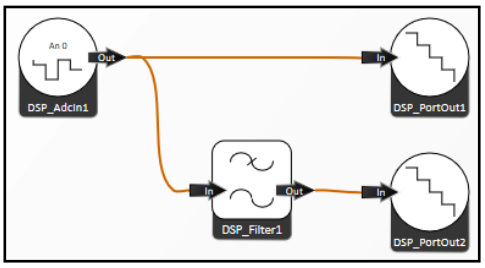
Music with microcontrollers



In music technology, ‘tone’ and ’frequency spectrum’ mean pretty much the same thing - the mixture of frequencies blending to produce the musical note.
‘Middle C’ on a piano sounds different to the same note on a violin because the instruments produce different blends of frequencies, different frequency spectrums.
This program shows how to control the tone of the output using a DSP Filter block.

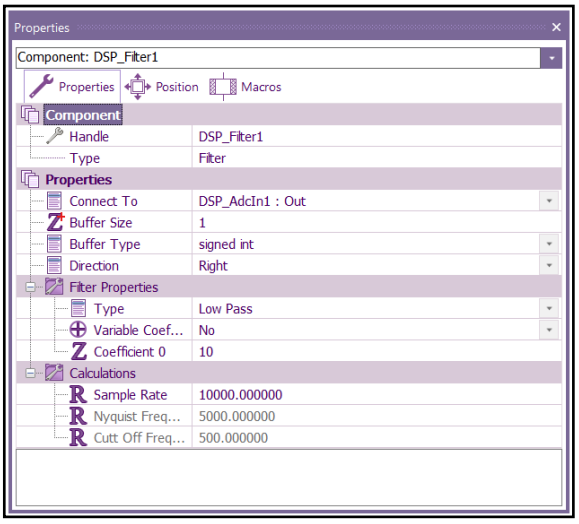
Over to you:

- Now, modify the ‘Control the Output’ program as follows:
  - Delete the ‘DSP_Scale’ block from the 2D panel and the ‘DSP_Scale1:Calculate()’ component macro from the 2D panel.
  - Instead, add a ‘DSP Filter’ block to the 2D panel, connected as shown opposite.

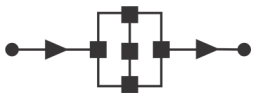


- The input port and output ports are configured in the same way as before.

- Set the filter properties as shown opposite.
- Notice that ‘Type’ is set to ‘Low Pass’ and the ‘Coefficient’ to ‘10’. These determine the effect of the filter on the signal.



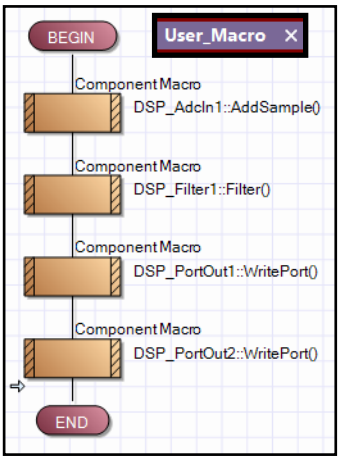
Worksheet 5
Control the tone



Music with microcontrollers

Over to you:

- The ‘Main’ program is the same as before, but change the name displayed on the LCD screen to “Control the Tone.”

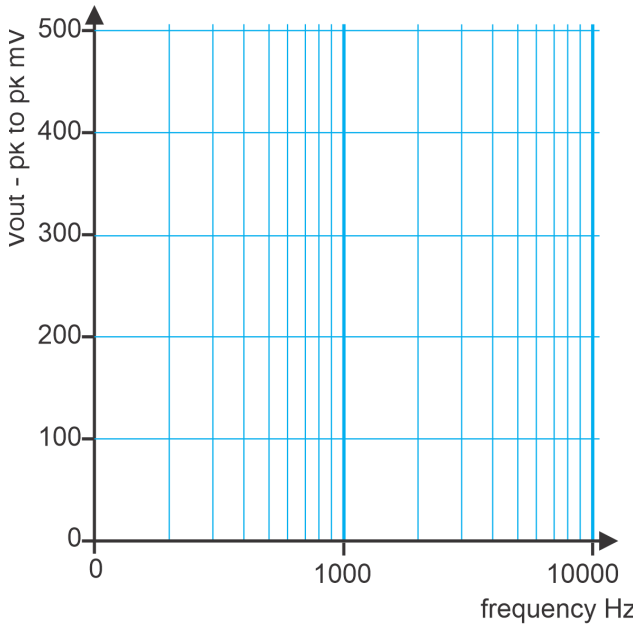


- Modify the ‘User_Macro’ by adding a component macro for the filter.
- The new ‘User_Macro is shown opposite.

Testing the program:

- Save the program as the ‘Control the Tone’ program.
- Use the ’Compile to Target’ option under the ’Build’ tab to transfer this program to the Sysblocks board.
- Test the program in the same way as for the previous program.
- Compare what you hear on the loudspeakers / headphones and see on the oscilloscope traces.
- Measure the output voltages for different frequencies and fill in the table and graph.

f Hz	Low pass RMS mV	High pass RMS mV
100		
300		
600		
1000		
2000		
3000		
5000		
8000		



Challenge:

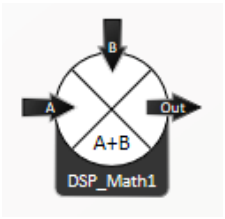
Investigate the effect of changing the DSP_Filter Type property to High pass what you hear and see on the oscilloscope traces. Record the results in the table in different columns. You can plot the output on the same graph.

Try playing music through the system from a source such as a smartphone or laptop. (Remember to change the ‘Input ADC Connection’ property to ‘An1’ or ‘An2’.) how does it sound?

Worksheet 6
Anatomy of an echo

Music with microcontrollers

At the heart of this program is the DSP_Math block.

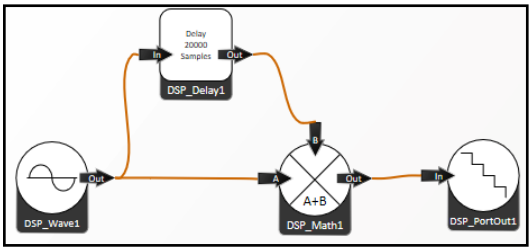


It is configured to add together two signals, signal A, received directly from the signal source, and signal B, the same signal, but delayed.

The output signal is the sum of these, A + B, which gives rise to an echo effect.

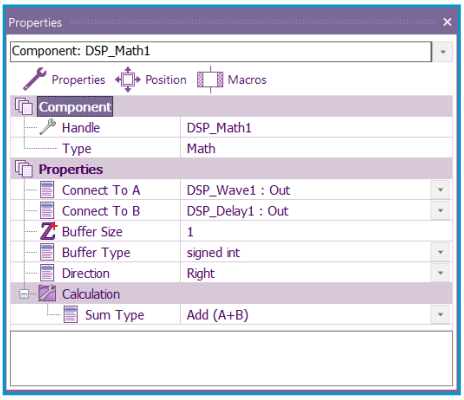
Over to you:

- Starting with the setup for the ‘Click Generator’ program, add a ‘DSPMath’ block and a ‘DSPDelay’ block to the 2D panel and connect them as shown opposite.

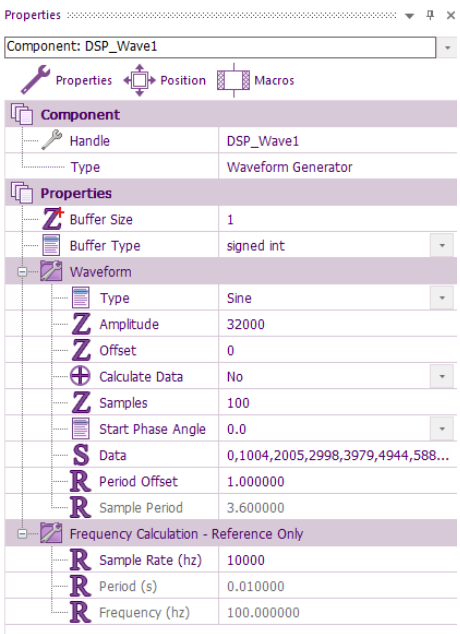


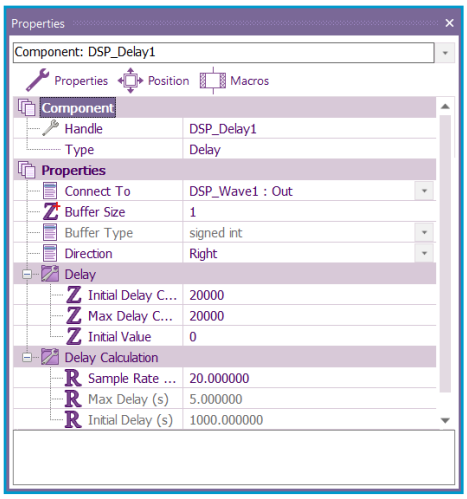
- The ‘Main’ program is the same as that in the ‘Click Generator’ program, but change the name displayed on the LCD screen to “Anatomy of an Echo.”

- The images opposite show the properties required for the new blocks.
- Configure the DSP blocks as shown.



- Reduce the amplitude of the DSP_Wave to 32000 as shown:



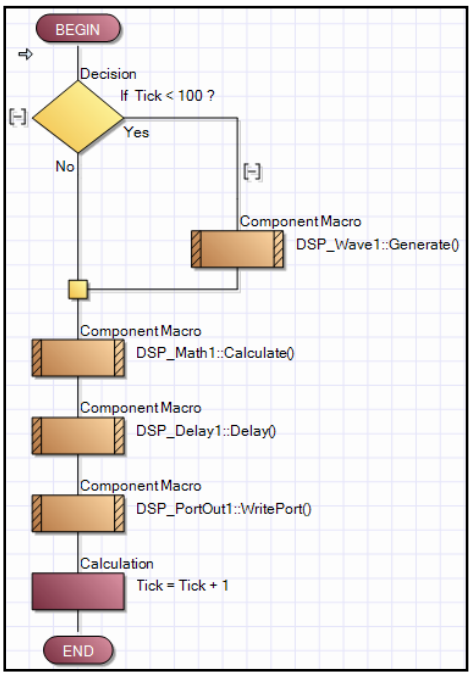


Worksheet 6
Anatomy of an echo

Music with microcontrollers

Over to you .....

- Modify the ‘User_Macro’ as shown opposite.



Testing the program :

- Save the program as ‘Anatomy of an Echo’.
- Use ‘Compile to Target’ to transfer it to Sysblocks.
- Connect a loudspeaker to the ‘LINE OUT’ socket of the Sysblocks board.
- Switch on the power supply.
- Press and release the ‘SW0’ switch on the Sysblocks board. You should hear a click, followed by an ’echo’ of the click each time the switch is pressed.
- Connect an oscilloscope to output 1 (OUT1) of the Sysblocks board.
- You can see both the click and its echo.

So what?

- The echo signal is delayed and added to the input signal. You can see this on an oscilloscope waveform.

Over to you:

- Explore the effects of changing the ‘delay’ properties of the DSP_Delay component.
- Replace the ‘WaveGenerator’ block with an ‘InputADC’ block, configured appropriately, and play music through the system from a source such as a smartphone or laptop.
- Increase the tick variable reset count to 10000
- Take the DSP_Addsample command out of the if statement and put it in the main interrupt routine so that the echo is directly linked to the input.

Worksheet 7
Multiple echoes

Music with microcontrollers

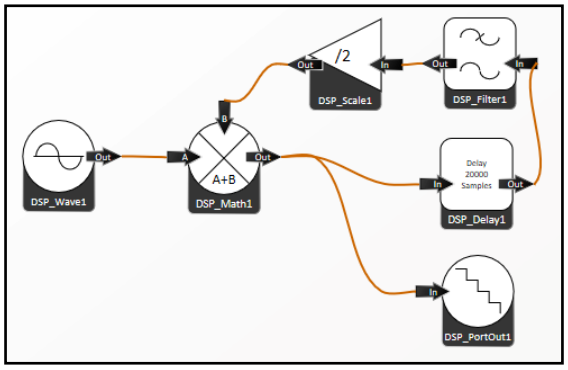
This time, the output signal is fed back through the system repeatedly via the DSP_Math block. This adds it to whatever has gone before.

Each time, however, its amplitude is halved by the DSP_Scale block.

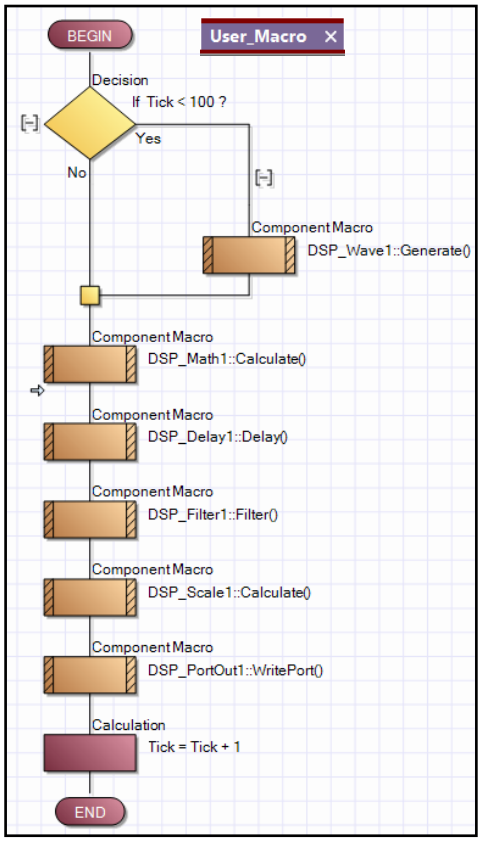
The result is that each echo is quieter than the one before - the echoes slowly die away.

Over to you

- Starting with the previous program, modify the 2D panel by adding a DSP_Scale block and a DSP_Filter block, connected as shown opposite. (The DSP_Filter and DSP_Scale blocks are shown 'flipped' right to left using the 'Direction' property to make the diagram easier to follow.)



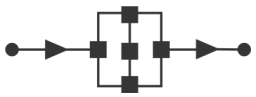
- The Filter properties are the same as in program 4, except that the 'coefficient 0' value is set to '4'.
- The Scale properties are the same as in program 3, except that the 'Initial Integer Scaler' value is set to '2'.
- The 'Main' program is the same as before but with the name displayed changed to "Multiple Echoes."
- Change the Delay property Initial delay count to 5000 samples.



- Modify the 'User_Macro' by adding component macros for each of the DSP blocks.
- It should resemble that shown opposite.

This completes the program.

Worksheet 7
Multiple echoes



Music with microcontrollers

Over to you .....

Testing the program :

- Save this as the ‘Multiple echoes’ program.
- Use the ‘Compile to Target’ option to transfer the program to Sysblocks.
- Connect a loudspeaker / headphones to the Sysblocks ‘LINE OUT’ socket.
- Switch on the power supply.
- Press and release the Sysblocks ‘SW0’ switch.
- You should hear a click, followed by echoes that die away.
- Connect an oscilloscope to output 1 (OUT1) of the Sysblocks board.
- You can see both the click and its echoes.

Challenge:

Explore the effects of changing the properties of:

- the DSP Scale component;
- the DSP Filtercomponent;
- the DSP Delay component.

Once again, replace the ‘WaveGenerator’ block with an ‘InputADC’ block and play music through the system from a source such as a smartphone or laptop.

Worksheet 8
Reverberation

Music with microcontrollers

Reverberation occurs naturally, caused by repeated echoes from nearby objects.

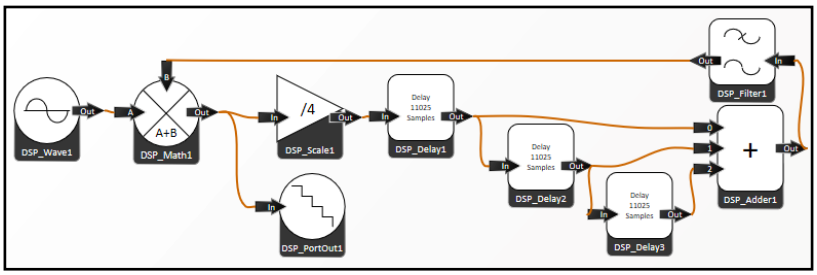
For example, it occurs when someone is talking in a large room or building with sound-reflecting surfaces.

Equally, it can be created artificially to add a sense of space and make a sound appear ‘more natural’.

To create artificial reverberation, the ‘Multiple Echoes’ system is extended by adding two more DSP_Delay blocks and a DSP_Adder block.

Over to you

- Add these blocks and connect the system as shown in the flowing diagram.



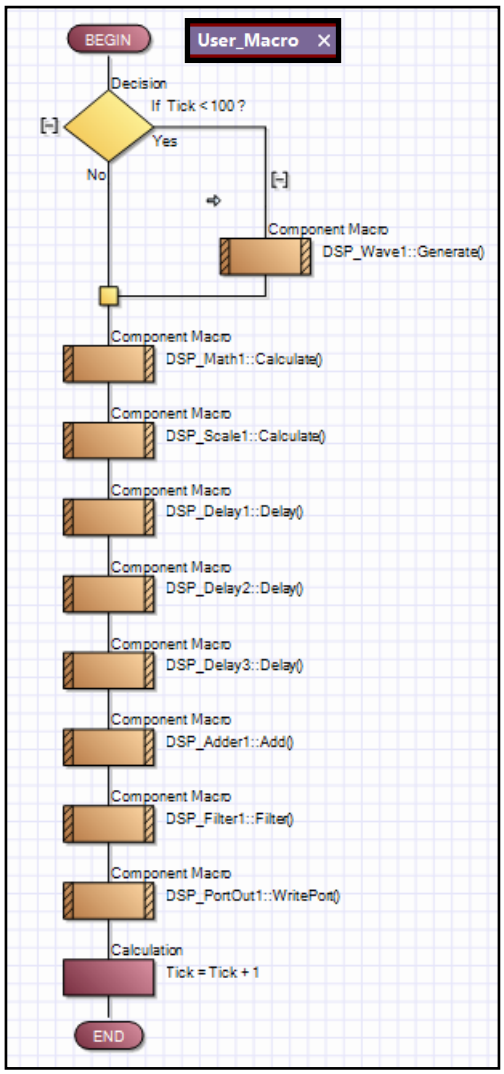
- The DSP Wave, Maths and PortOut blocks are configured as in previous programs. Configure the added DSP blocks as described in the table that follows:

Block	Property
Adder	set to handle three inputs.
Delays (x3)	set to counts of 11025 samples, (i.e. quarter second delay) each.
Scale	‘Initial Integer Scaler’ - set to divide by ‘4’.
Filter	‘Type’ - set to ‘Low Pass’
	‘Coefficient’ - set to ‘4’.

- The ‘Main’ program is the same as before but change the name displayed on the LCD screen to “Reverberation”.

Over to you

- Modify the ‘User_Macro’ by adding component macros for each of the DSP blocks.
- It should resemble the one in the image below.



Testing the program :

- Save the program as ‘Reverberation’.
- Use the ‘Compile to Target’ option to transfer the program to Sysblocks.
- Use the same test procedure as in the previous program.
- Once again, replace the ‘WaveGenerator’ block with an ‘InputADC’ block, configured appropriately and try playing music from a source such as a smartphone or laptop.

Worksheet 9
Mixing signals

Music with microcontrollers

In a recording studio, often, audio from several sources is combined into one stream.

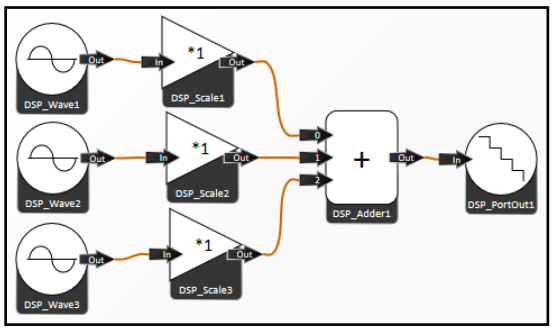
One way to do this is to use a mixing desk, like the one shown in the photo. It allows each source to be boosted or attenuated and faded in or out.

It may also offer tone controls for each channel as well.

Over to you

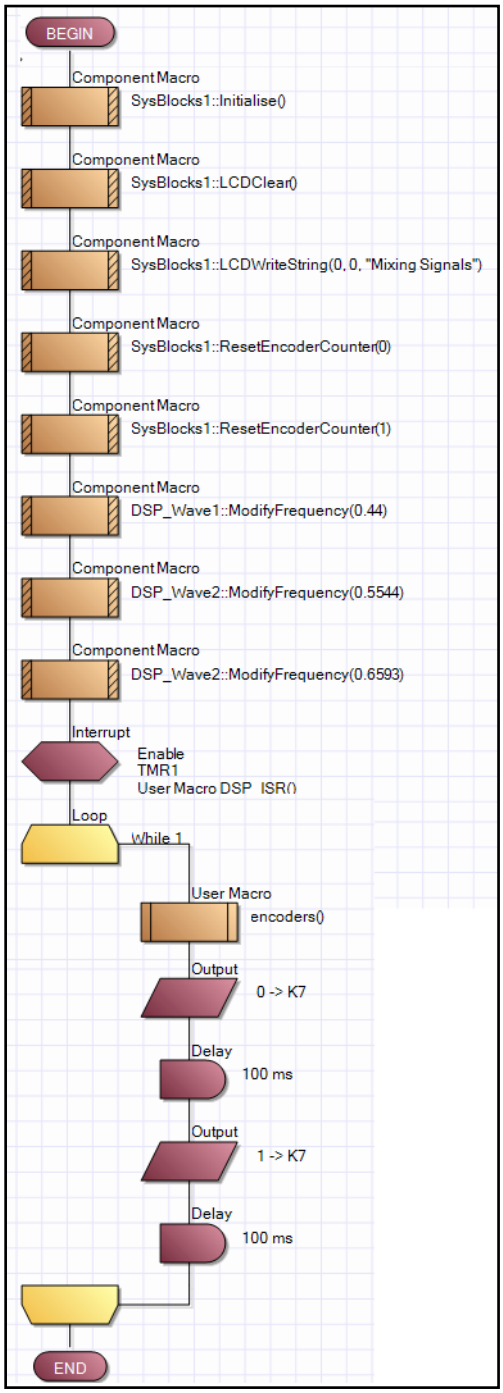
- The 'Main' program is shown opposite.
- Change the name displayed to "Mixing Signals."

- Set up the 2D panel as shown below.
- It shows three wave generators providing signals which are then mixed by the DSP_Adder block

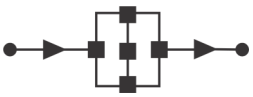


The program includes the following three macros:

- 'encoders' - called by the 'User Macro inside the loop.
- 'DSP_ISR' - called by the Interrupt.
- 'Set_Levels' - called by the 'Set scales' macro in the 'encoders' sub-routine.



Worksheet 9
Mixing signals

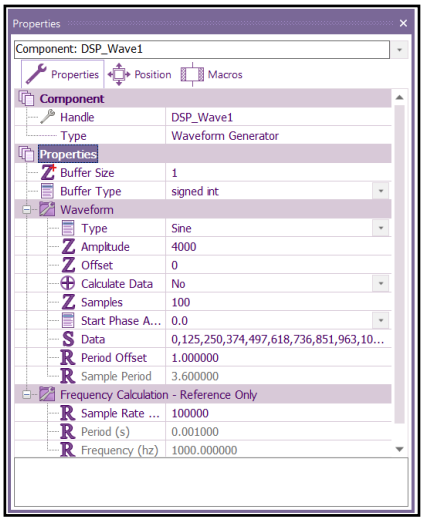


Music with microcontrollers

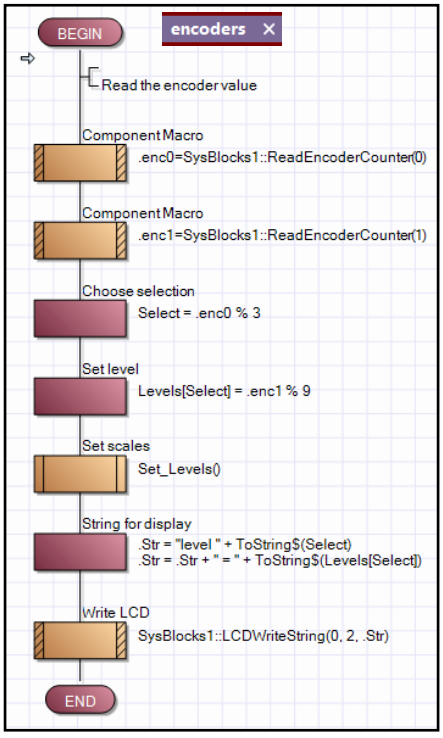
Over to you .....

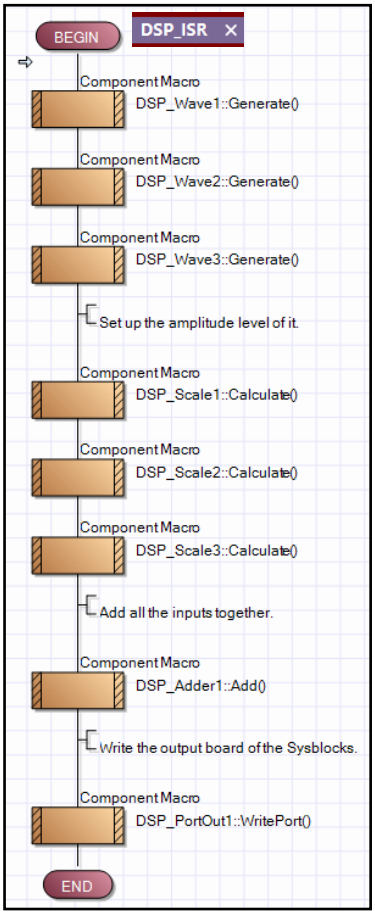
- Configure the DSP blocks using the information in the table below.

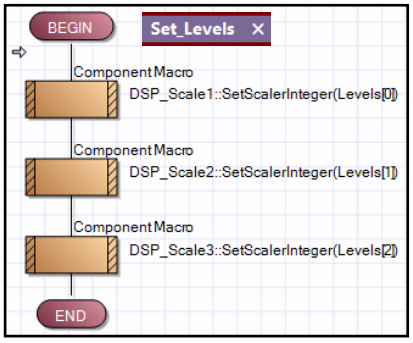
Block	Property
Adder	set to handle three inputs.
DSP_Scale (x3)	'Initial Integer Scaler' - set to multiply by '1'.
	runs the macro "SetLevels".

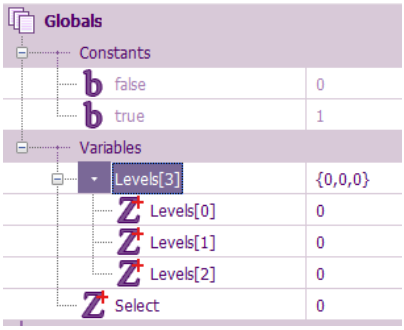


- All three wave generators have the same properties, shown opposite, and call the macro 'ModifyFrequency'.
- The images below show the structure of the three macros. Create them and complete the rest of the program.
- For the encoder routine you will need to add a global array 'Levels' of type UINT and a variable 'select' of type UINT. You will also need Local variables 'enc0' and enc1' of type UINT in the Encoders macro.
- The % operator is a MODulo operator which takes the remainder from a division

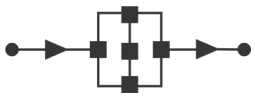








Worksheet 9
Mixing signals



Music with microcontrollers

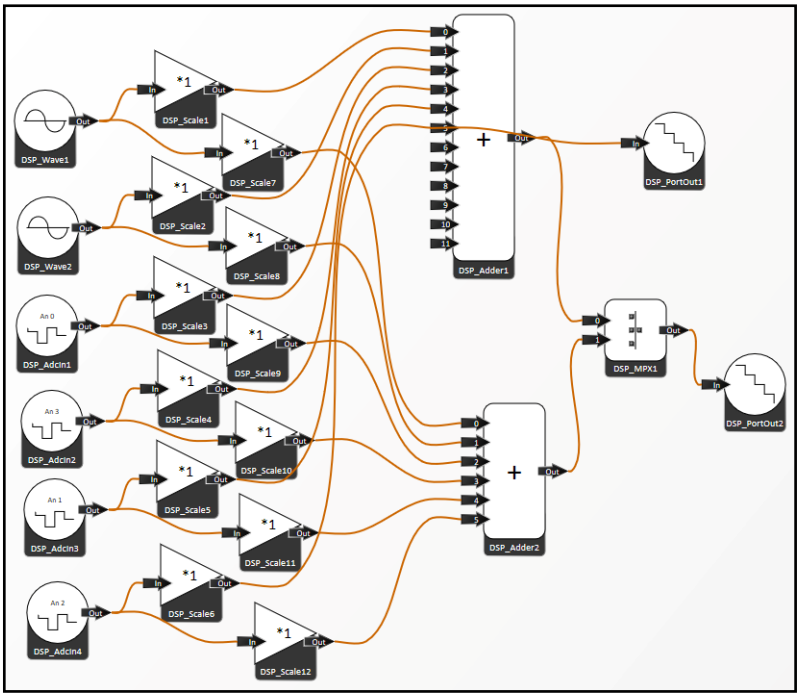
Over to you .....

Testing the program :

- Save the program as ‘Mixing signals’.
- Use the ‘Compile to Target’ option to transfer the program to Sysblocks.
- Connect a loudspeaker / headphones to the Sysblocks ‘LINE OUT’ socket.
- Connect an oscilloscope to output 0 (OUT0) of the Sysblocks board.
- Switch on the power supply.
- Encoder ‘ENC0’ is used to select the signal source, ‘0’, ‘1’ or ’2’. The other encoder ‘ENC1’ is used to set the signal level, ‘0’ to ‘8’. Test them, while you listen to the sound produced and observe the trace on the oscilloscope.

Challenge:

- Run and test the program ‘Mixing Desk Demo’. This has six possible signal sources, two using the built-in wave generators and four using input ports.

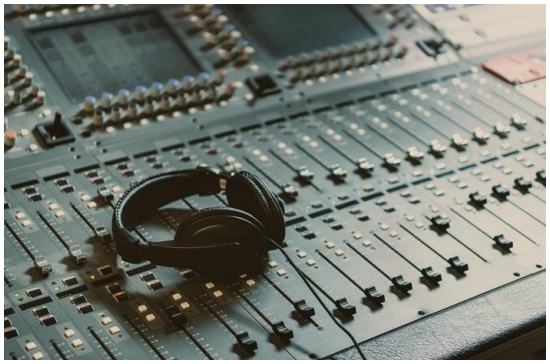


- Configure the DSP_Input ports so that you can try signals from sources such as smartphones or laptops.

- Experiment with building and testing a similar program for yourself.

Worksheet 10
Equalisation

Music with microcontrollers



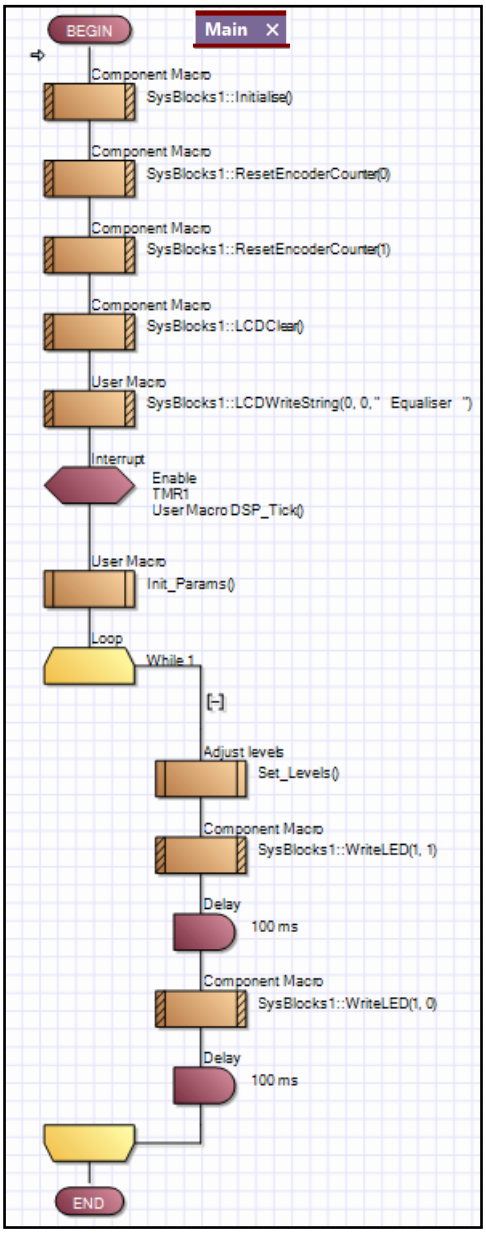
An audio (graphics) equaliser adjusts the way the system responds to different frequencies.

It does so in order to correct for any effects of the surroundings or the behaviour of other components in the system, such as speakers or microphones.

Over to you

- Use the information given on this and the next page to create the ‘Equaliser’ program.
- The ‘Main’ program is shown opposite.
  - The program includes three macros:
  - ‘DSP_Tick’, called by the Interrupt;
  - ‘Init_Params’, called by ‘User Macro’ macro following the Interrupt;
  - ‘Set_Level’ called by the ‘Set_Levels’ macro inside the loop.

The properties of these macros are shown below.

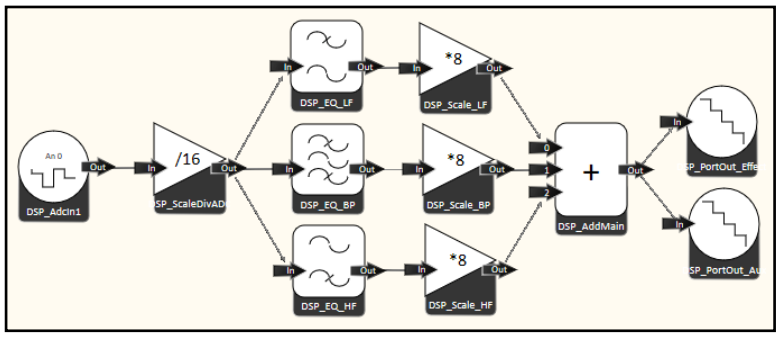


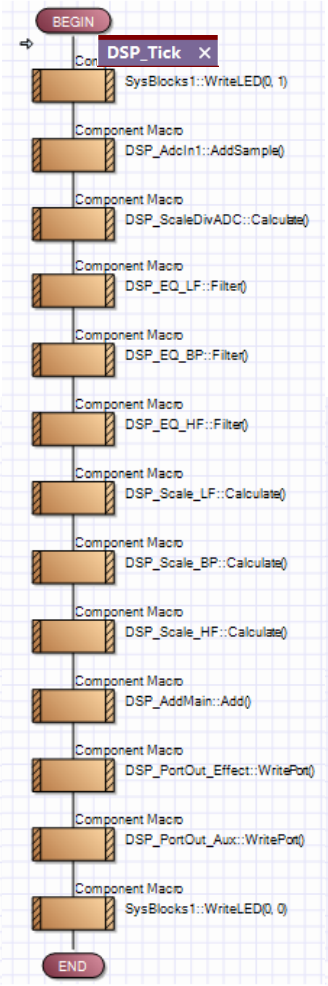
Worksheet 10
Equalisation

Music with microcontrollers

Over to you .....

The diagrams show the layout of the system and the structure of the three macros:



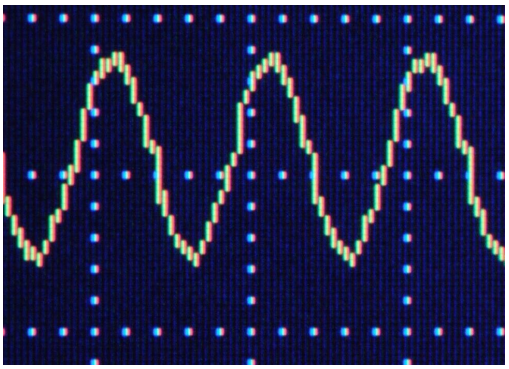


Testing the program :

- Save the program as ‘Equalisation’
- Use the same test procedure as in the previous program.
- Again, try playing music through the system from a smartphone or laptop, using an appropriate configuration.

Worksheet 11
Sampling

Music with microcontrollers



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 the resulting data file. There must be a compromise!

The image shows the effect of sampling rate on an oscilloscope trace of an audio signal.

Over to you

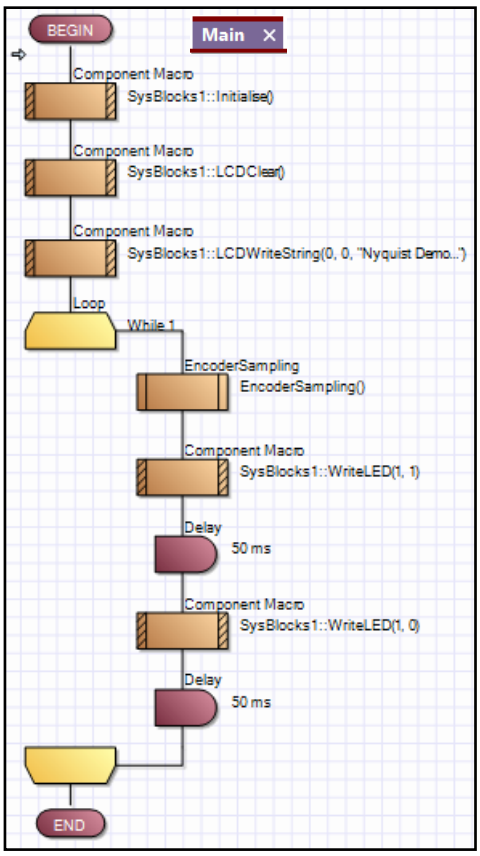
- Use the information given on this and the next page to create the sampling program.

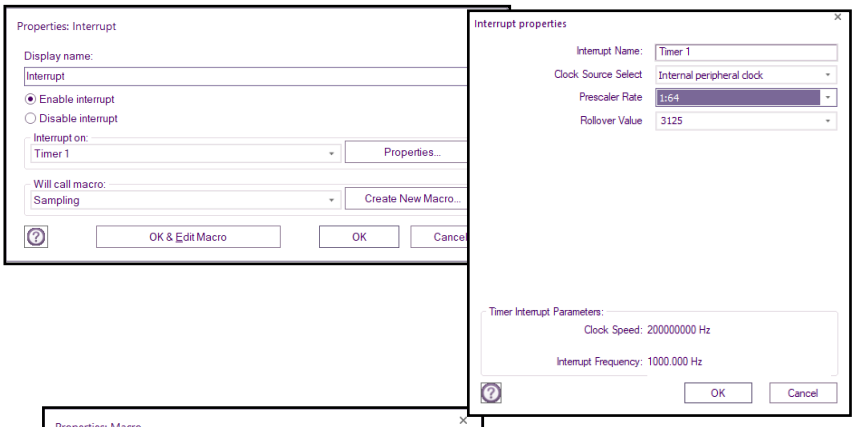
The ‘Main’ program is shown opposite.

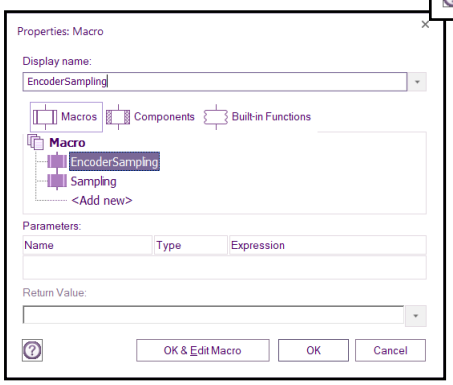
The program includes two macros:

- ‘EncoderSampling’, called by the first macro in the loop in the main program;
- ‘Sampling’, called by the Interrupts in the ‘EncoderSampling’ macro.

The properties of these macros are shown below.





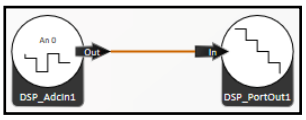


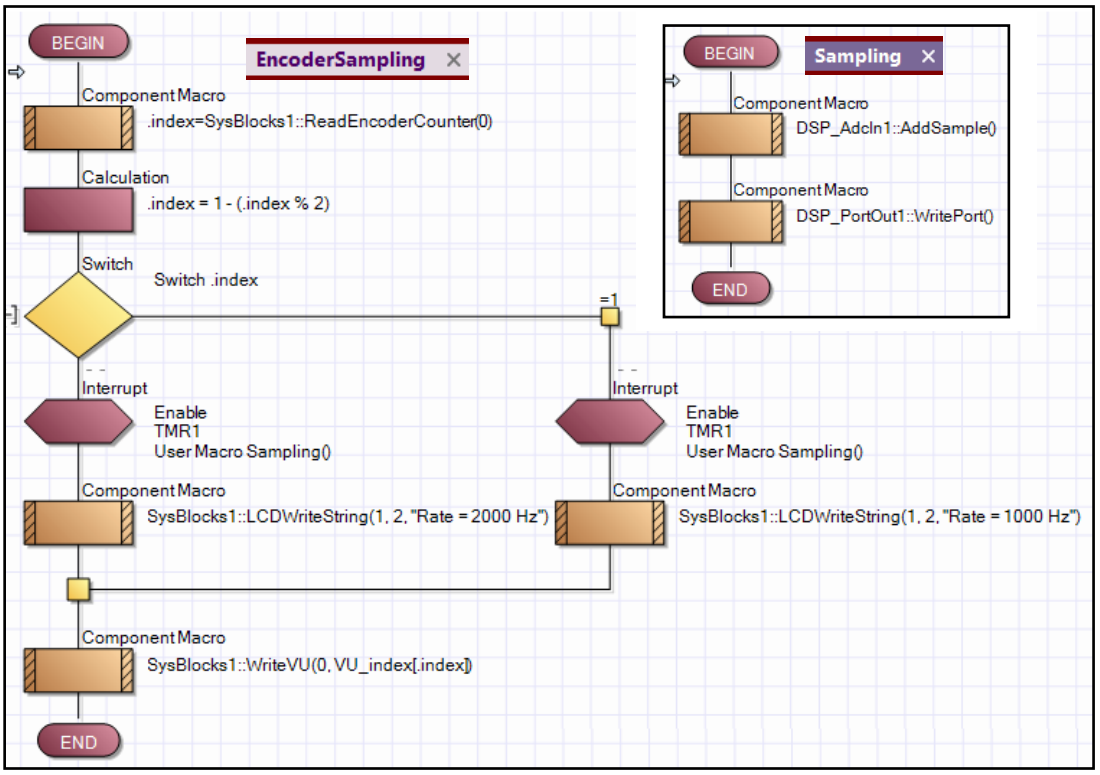
Worksheet 11
Sampling

Music with microcontrollers

Over to you .....

The diagram opposite shows the layout of the system.
It is identical to that used in the 'In and Out' program.
The next diagrams show the structure of the two macros:





Testing the program :

- Save the program as 'Sampling'.
- Use the 'Compile to Target' option to transfer the program to Sysblocks.
- Connect a signal source, such as the 'AWG' signal generator, to Sysblocks input 0 (IN0) (since channel 'AN0' was selected for the 'Input ADC' block.)
- Set it to generate a sine wave with a frequency of around 4000Hz.
- Connect a loudspeaker / headphones to the Sysblocks 'LINE OUT' socket.
- Connect an oscilloscope to output 0 (OUT0) of the Sysblocks board.
- Switch on the power supply.
- Encoder 'ENC0' is used to select the sampling rate, either 1000 or 2000 samples per second. Test them, while you listen to the sound produced and observe the trace on the oscilloscope.
- Examine the output signal trace on the oscilloscope for both settings of sample rate.
- Listen to the sounds produced by each.
- In each case, you might notice a slight difference.

Version control

30 11 23 first release