Emona DATEx SAMPLE Lab Manual

Volumes 1, 2 & 3
Experiments in Modern Analog & Digital Telecommunications
For NI™ ELVIS I & II+

Emona DATEx SAMPLE Lab Manual for NI™ ELVIS I & II/+
Volumes 1, 2 & 3 – Extracts
Experiments in Modern Analog and Digital Telecommunications.

Issue Number: 2.0

Published by:

Emona Instruments Pty Ltd,
78 Parramatta Road
Camperdown NSW 2050
AUSTRALIA.

web: www.emona-tims.com
telephone: +61-2-9519-3933
fax: +61-2-9550-1378

Printed in Australia

Emona DATEx SAMPLE Lab Manual

Contents

Emona DATEx VOLUME 1

Contents

Name: ______________________________

Class: ______________________________

4 - Using the Emona DATEx to model equations

Experiment 4 – Using the Emona DATEx to model equations

Preliminary discussion
This may surprise you, but mathematics is an important part of electronics and this is especially true for communications and telecommunications. As you'll learn, the output of all communications systems can be described mathematically with an equation.

Although the math that you'll need for this manual is relatively light, there is some. Helpfully, the Emona DATEx can model communications equations to bring them to life.

The experiment
This experiment will introduce you to modelling equations by using the Emona DATEx to implement two relatively simple equations.

It should take you about 40 minutes to complete this experiment.

Equipment

- Personal computer with appropriate software installed
- NI ELVIS II plus USB cable and power pack
- Emona DATEx experimental add-in module
- Two BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads

Something you need to know for the experiment

This box contains the definition for an electrical term used in this experiment. Although you've probably seen it before, it's worth taking a minute to read it to check your understanding.

When two signals are **180° out of phase**, they're out of step by half a cycle. This is shown in Figure 1 below. As you can see, the two signals are always travelling in opposite directions. That is, as one goes up, the other goes down (and vice versa).

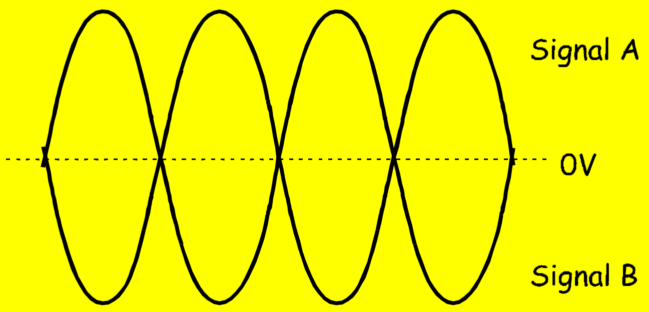


Figure 1

Procedure

In this part of the experiment, you're going to use the Adder module to add two electrical signals together. Mathematically, you'll be implementing the equation:

Adder module output = Signal A + Signal B

1. Ensure that the NI ELVIS II power switch at the back of the unit is off.

2. Carefully plug the Emona DATEx experimental add-in module into the NI ELVIS II.

3. Set the *Control Mode* switch on the DATEx module (top right corner) to *PC Control.*

4. Connect the NI ELVIS II to the PC using the USB cable.

   Note: This may already have been done for you.

5. Turn on the NI ELVIS II power switch at the rear of the unit then turn on its *Prototyping Board Power* switch at the top right corner near the power indicator.

6. Turn on the PC and let it boot-up.

7. Launch the NI ELVISmx software.

Ask the instructor to check
your work before continuing.

8. Launch and run the NI ELVIS II Oscilloscope virtual instrument (VI).

9. Set up the scope per the procedure in Experiment 1 (page 1-12) ensuring that the *Trigger Source* control is set to *CH 0.*

10. Launch the DATEx soft front-panel (SFP).

11. Check you now have soft control over the DATEx by activating the PCM Encoder module's **soft** *PDM/TDM* control on the DATEx SFP.

    Note: If you're set-up is working correctly, the PCM Decoder module's LED on the DATEx board should turn on and off.

12. Locate the Adder module on the DATEx SFP and set its soft *G* and *g* controls to about the middle of their travel.

13. Connect the set-up shown in Figure 2 below.

    Note: Although not shown, insert the black plugs of the oscilloscope leads into a ground (*GND*) socket.

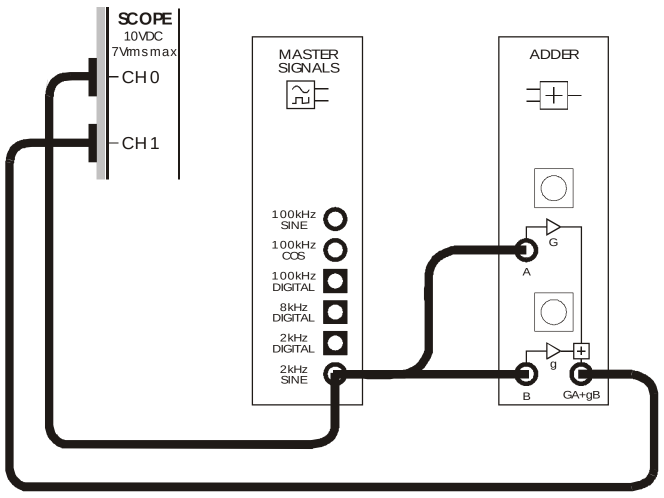


Figure 2

This set-up can be represented by the block diagram in Figure 3 below.

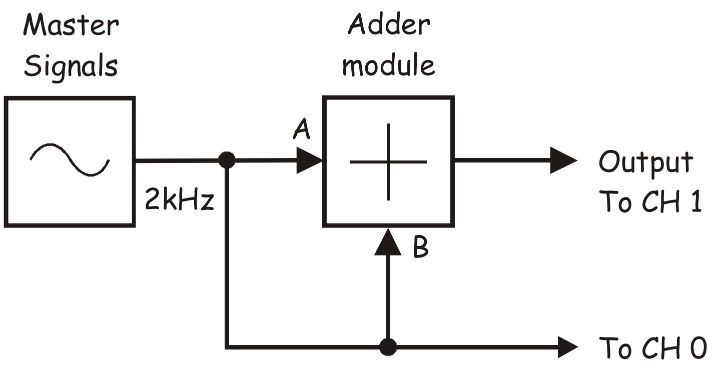


Figure 3

14. Adjust the scope's *Timebase* control to view two or so cycles of the Master Signals module's *2kHz SINE* output.

15. Measure the amplitude (peak-to-peak) of the Master Signals module's *2kHz SINE* output. Record your measurement in Table 1 on the next page.

16. Disconnect the lead to the Adder module's *B* input.

17. Activate the scope's Channel 1 input by checking the Channel 1 *Enabled* box to observe the Adder module's output as well as its input.

18. Adjust the Adder module's soft *G* control until its output voltage is the same size as its input voltage (measured in Step 15).

    Note 1: This makes the gain for the Adder module's *A* input -1.

    Note 2: Remember that you can use the keyboard's *TAB* and arrow keys for fine adjustment of the DATEx SFP's controls.

19. Reconnect the lead to the Adder module's *B* input.

20. Disconnect the lead to the Adder module's *A* input.

21. Adjust the Adder module's soft *g* control until its output voltage is the same size as its input voltage (measured in Step 15).

    Note: This makes the gain for the Adder module's *B* input -1 and means that the Adder module's two inputs should have the same gain.

22. Reconnect the lead to the Adder module's *A* input.

The set-up shown in Figures 3 and 4 is now ready to implement the equation:

Adder module output = Signal A + Signal B

Notice though that the Adder module's two inputs are the same signal: a 4Vp-p 2kHz sinewave. So, for these inputs the equation becomes:

Adder module output = 4Vp-p (2kHz sine) + 4Vp-p (2kHz sine)

When the equation is solved, we get:

Adder module output = 8Vp-p (2kHz sine)

Let's see if this is what happens in practice.

23. Measure and record the amplitude of the Adder module's output.

Table 1

Input voltage	**Output voltage**

Question 1
Is the Adder module's measured output voltage **exactly** 8Vp-p as theoretically predicted?

No.

Question 2
What are two reasons for this?

1) Loading (that is, the Adder's input is not exactly 4Vp-p)

2) The gains aren't exactly -1.

✓ Ask the instructor to check your work before continuing.

In the next part of the experiment, you're going to add two electrical signals together but one of them will be phase shifted. Mathematically, you'll be implementing the equation:

Adder module output = Signal A + Signal B (with phase shift)

24. Locate the Phase Shifter module on the DATEx SFP and set its soft *Phase Change* control to the *0°* position.

25. Set the Phase Shifter module's soft *Phase Adjust* control about the middle of its travel.

26. Connect the set-up shown in Figure 4 below.

Note: Insert the black plugs of the oscilloscope leads into a ground (*GND*) socket.

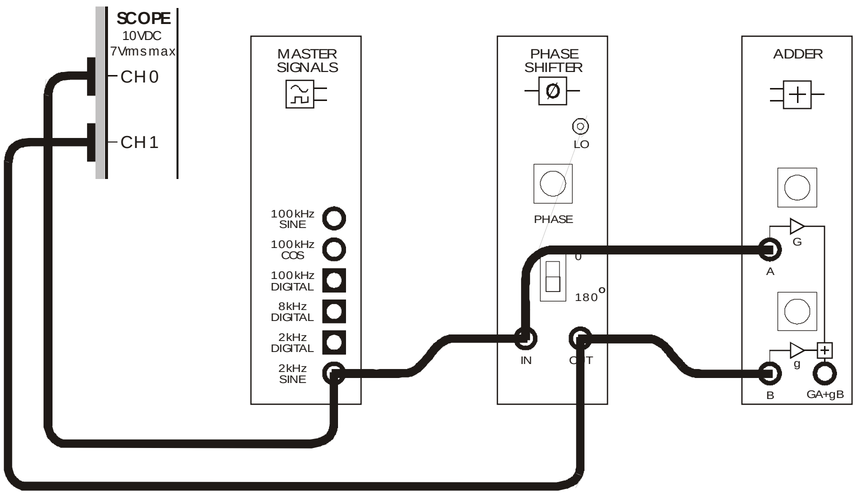


Figure 4

This set-up can be represented by the block diagram in Figure 5 on the next page.

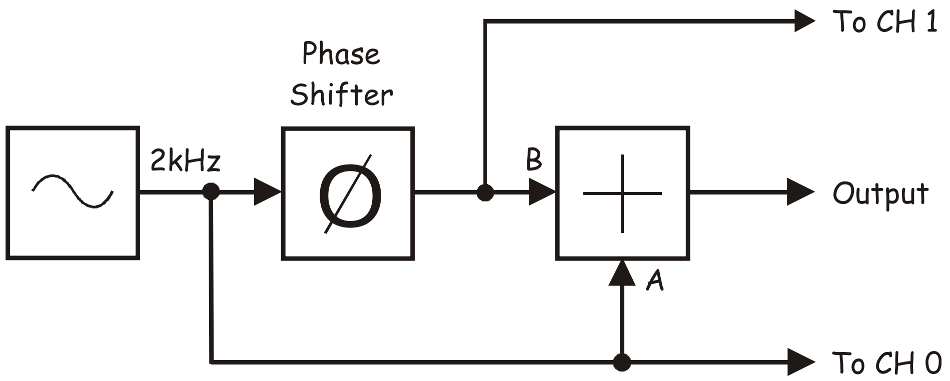


Figure 5

The set-up shown in Figures 4 and 5 is now ready to implement the equation:

Adder module output = Signal A + Signal B (with phase shift)

The Adder module's two inputs are still the same signal: a 4Vp-p 2kHz sinewave. So, with values the equation is:

Adder module output = 4Vp-p (2kHz sine) + 4Vp-p (2kHz sine with phase shift)

As the two signals have the same amplitude and frequency, if the phase shift is exactly 180° then their voltages at any point in the waveform is always exactly opposite. That is, when one sinewave is +1V, the other is -1V. When one is +3.75V, the other is -3.75V and so on. This means that, when the equation above is solved, we get:

Adder module output = 0Vp-p

Let's see if this is what happens in practice.

27. Adjust the Phase Shifter module's soft *Phase Adjust* control until its input and output signals look like they're about 180° out of phase with each other.

28. Disconnect the scope's Channel 1 lead from the Phase Shifter module's output and connect it to the Adder module's output.

29. Adjust Channel 1's *Scale* control to resize the signal on the display.

30. Measure the amplitude of the Adder module's output. Record your measurement in Table 2 below.

Table 2

Output voltage

Question 3
What are two reasons for the output not being 0V as theoretically predicted?

1) The phase difference between the Adder's two inputs is not **exactly** 180°; and

2) The gains aren't **exactly** the same.

✓ Ask the instructor to check your work before continuing.

The following procedure can be used to adjust the Adder and Phase Shifter modules so that the set-up has a *null* output. That is, an output that is close to zero volts.

31. Use the keyboard's *TAB* and arrow keys to vary the Phase Shifter module's soft *Phase Adjust* control left and right a little and observe the effect on the Adder module's output.

32. Use the keyboard to make the necessary fine adjustments to the Phase Shifter module's soft *Phase Adjust* control to obtain the smallest output voltage from the Adder module.

Question 5
What can be said about the phase shift between the signals on the Adder module's two inputs now?

The phase shift is much closer to 180° (but it's probably still not exactly 180°)

33. Use the keyboard to vary the Adder module's soft *g* control left and right a little and observe the effect on the Adder module's output.

34. Use the keyboard to make the necessary fine adjustments to the Adder module's soft *g* control to obtain the smallest output voltage.

Question 6
What can be said about the gain of the Adder module's two inputs now?

They're much closer to each other (but they're still probably not exactly the same)

You'll probably find that you'll not be able to null the Adder module's output completely. Unfortunately, real systems are never perfect and so they don't behave exactly according to theory. As such, it's important for you to learn to recognise these limitations, understand their origins and quantify them where necessary.

Ask the instructor to check your work before finishing.

Emona DATEx VOLUME 2

Contents

Name: ______________________

Class: ______________________

14 - PN sequence spectra and noise generation

Experiment 14 – PN sequence spectra and noise generation

Preliminary discussion
Pseudo-noise sequences (or just PN sequences) are very useful signals in communications and telecommunications, especially for implementing modulation schemes such as DSSS and CDMA (among others). They can also be used to generate noise for experimental purposes when modelling real world communications systems. But what exactly is a PN sequence?

To understand the answer to this question, you must return to the spectral composition of pulse trains. Recall that a pulse train is made up of a theoretically infinite number of sinewaves – the fundamental and its harmonics. Recall also that the frequency and amplitude of a pulse train's sinusoidal components affects its frequency and mark-space ratio (or duty cycle). Despite this, the spectral composition of all pulse trains follows the pattern of the (truncated) *Sinc Function* shown in Figure 1 below.

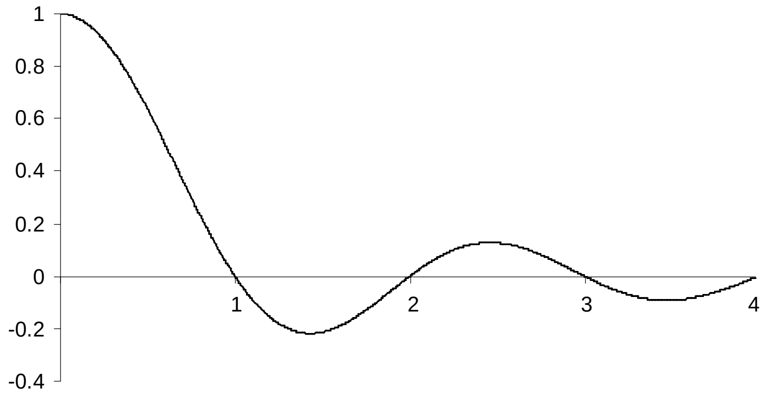


Figure 1

Figure 2 below illustrates this with an example of a 1kHz squarewave (a pulse train with a mark-space ratio of 1:1 or a duty cycle 50%). This is a spectrum that would be familiar to you.

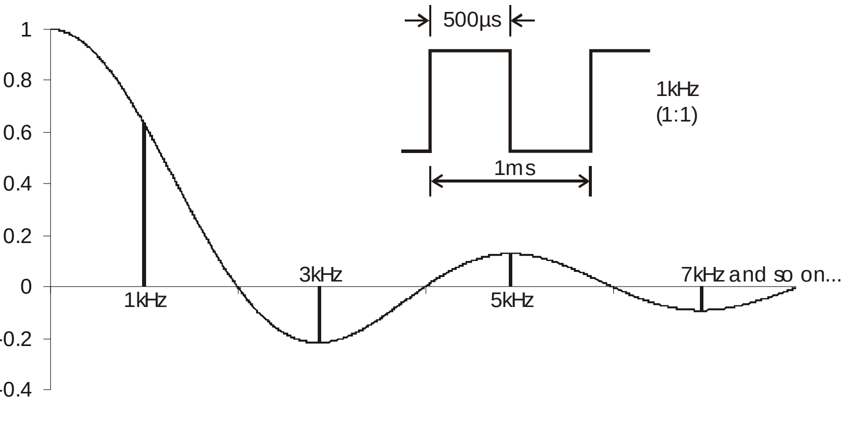


Figure 2

Figure 3 below shows the spectral composition of a 1kHz pulse train pulse having a mark-space ratio of 1:3 (or a duty cycle 25%). Notice that it too follows the pattern of the Sinc Function.

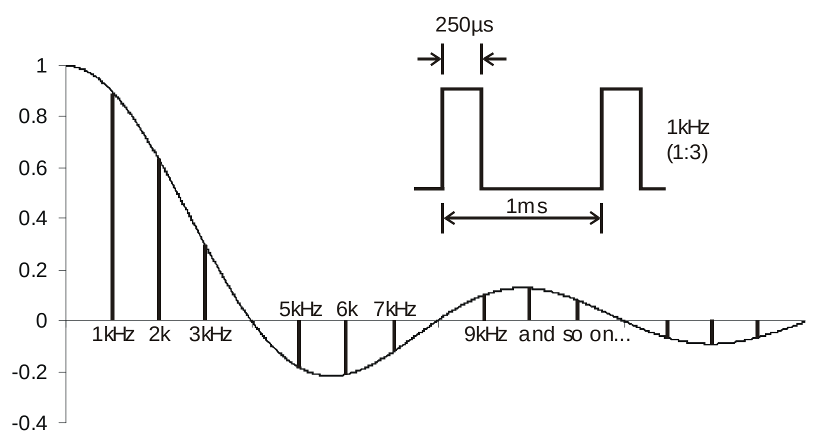


Figure 3

The examples in Figures 2 and 3 are instructive. Together, they show us that some harmonics of the pulse trains have an amplitude of zero (or are "nulled") and this is true of all pulse trains. Second, a comparison of Figures 2 and 3 shows us that, as the pulse train's mark-space ratio decreases, the number of significant harmonics that make it up increases. Or, put another way, as the mark-space ratio decreases, the number of harmonics that are present in each of the Sinc Function's lobes increases.

Now, suppose a sequence generator continuously outputs the sequential 4-bit binary number 1000 with each bit being 250μs wide (requiring a bit-clock of 4kHz). In the time domain, the resulting digital data signal is identical to the pulse train in Figure 3. This means that the sequence's spectral composition must be identical to the spectrum in Figure 3 also.

This fact has a couple of important implications. First, we can establish a general rule for determining the nulled harmonics in repeated sequential binary number sequences. They correspond with whole number multiples of the digital signal's bit-clock (that is, f_{bit}, $2f_{bit}$, $3f_{bit}$ and so on). In the case of our repeated sequential 4-bit binary number 1000 generated using a 4kHz bit-clock, the nulls occur at 4kHz, 8kHz, 12kHz and so on to infinity (theoretically).

Second, if the sequence generator's continuously repeated output is changed to the 5-bit binary number sequence 10000, the mark-space ratio of the resulting digital data signal decreases and so more harmonics are present between the nulls. Importantly though, if a 4kHz bit-clock is used to generate the 5-bit sequence, the nulls occur at the same frequencies as our example in Figure 3. So, with the nulls occurring at the same frequencies but with more harmonics between them, the spectral composition of the 5-bit sequence must be richer than that of its 4-bit counterpart. This gives us a second general rule. The greater the number of bits in a repeated sequence for a given bit-clock, the greater the sequence's spectral composition (though this doesn't apply to PN sequences with internally repeated sequences like 101010… and 11001100…).

Using the Sinc Function to analyse the spectral composition of several binary number sequences like 1000, 10000, 100000 and so on would quickly show that the number of harmonics in each lobe is the same number as the sequence's length (though the last one is nulled).

Finally, we can now return to the question of what is a pseudo-noise sequence. If the length of certain binary number sequences is long enough, their spectral composition becomes so dense that it can be used to model bandwidth limited white noise. That said, there would still be a repetitive element to the "noise signal" and so they're called pseudo (or "apparent") noise sequences.

The experiment
For this experiment you'll use the Emona DATEx to consider a 31-bit and 255-bit binary number sequence in the time domain. You'll then look at the data signals' spectra in the frequency domain to confirm their spectral composition. Finally, you'll use the sequences to generate electrical noise and compare their effectiveness.

It should take you about 50 minutes to complete this experiment.

Pre-requisites:
Experiments 1, 2 & 3 (Vol. 1): Intros to the NI ELVIS II, the Emona DATEx and SFP control

Equipment

- Personal computer with appropriate software installed
- NI ELVIS II plus USB cable and power pack
- Emona DATEx experimental add-in module
- Three BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads

Procedure

Part A – Observations of PN sequences in the time domain
The next part of this experiment gets you to set up a 31-bit and a 255-bit binary number sequence and consider them in the time domain as preparation for looking at their spectra.

1. Ensure that the NI ELVIS II power switch at the back of the unit is off.

2. Carefully plug the Emona DATEx experimental add-in module into the NI ELVIS II.

3. Set the *Control Mode* switch on the DATEx module (top right corner) to *PC Control*.

4. Connect the NI ELVIS II to the PC using the USB cable.

   Note: This may already have been done for you.

5. Turn on the NI ELVIS II power switch at the rear of the unit then turn on its *Prototyping Board Power* switch at the top right corner near the power indicator.

6. Turn on the PC and let it boot-up.

7. Launch the NI ELVISmx software.

8. Connect the set-up shown in Figure 4 below.

   Note: Insert the black plugs of the oscilloscope leads into a ground (*GND*) socket.

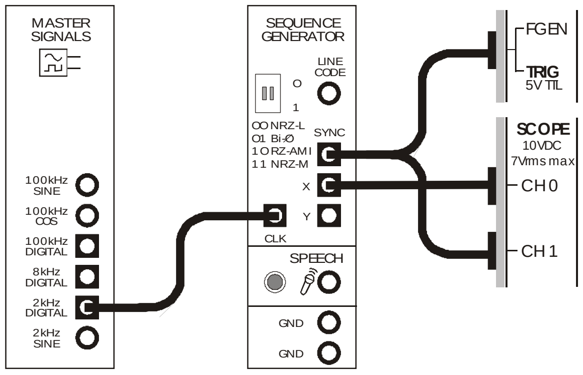


Figure 4

The set-up in Figure 4 can be represented by the block diagram in Figure 5 below. The Master Signals module's *2kHz DIGITAL* output is used to provide the Sequence Generator module's bit-clock. The Sequence Generator module's *X* output is a continuous 31-bit sequential binary number. The module's *SYNC* output is a pulse that corresponds with the sequence's first output bit on every repetition.

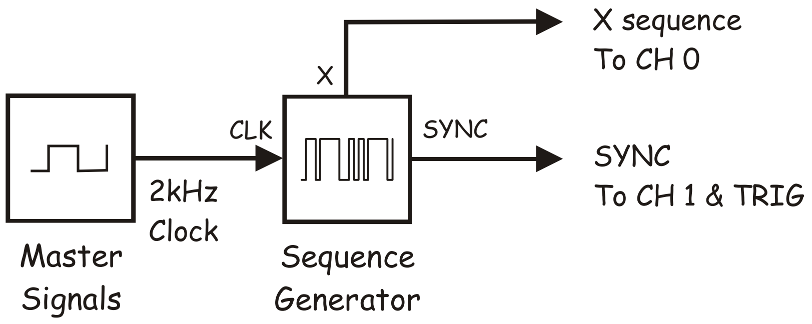


Figure 5

9. Launch and run the NI ELVIS II Oscilloscope VI.

10. Adjust the scope to view the Sequence Generator module's *X* output as a stable display. Essential scope settings include:

    - *Timebase* to *2ms/div*
    - *Trigger Type* to *Digital*
    - CH 1 *Vertical Position* to *-5V*

11. Activate the scope's Channel 1 input (by checking the Channel 1 *Enabled* box) to view both the Sequence Generator module's *X* and *SYNC* outputs.

✓ Ask the instructor to check your work before continuing.

Question 1
Calculate the width of each bit in the Sequence Generator module's *X* sequence for the bit-clock used. **Note:** For accuracy here, you need to be aware that the Master Signals module's 2kHz outputs are actually 2.083kHz.

Bit-width = $\frac{1}{f_{bit}} = \frac{1}{2.083kHz}$ = 480μs.

Question 2
Calculate the duration of the entire 31-bit sequence.

Duration = bit-width × number of bits = 480μs × 31 = 14.88ms

The next part of the experiment gets you to verify your answer to Question 2 using the scope's cursors.

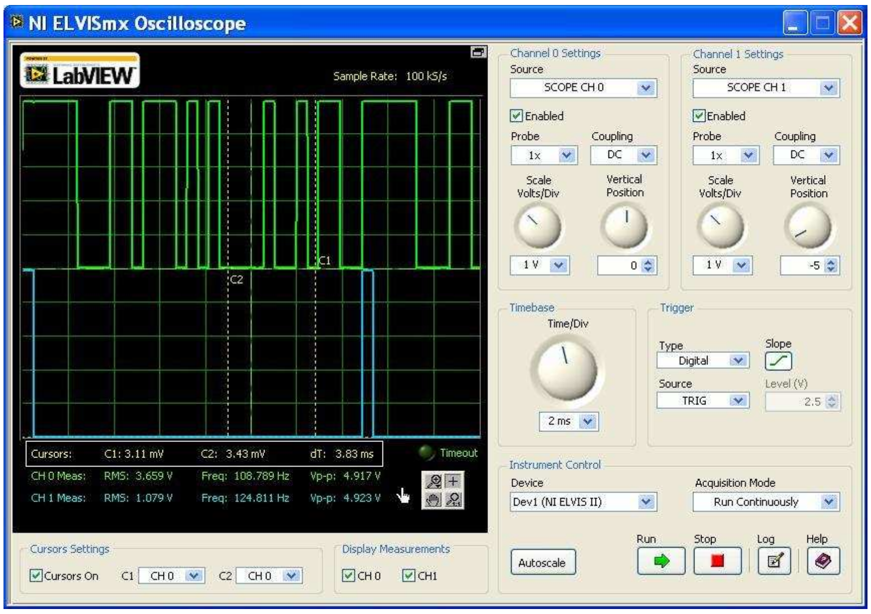
12. Activate the scope's cursors by checking (that is, ticking) the scope's *Cursors On* box.

The NI ELVIS II Oscilloscope has two cursors (*C1* and *C2*) that default to the left most side of the display when the scope's VI is first launched. They're repositioned by "grabbing" their vertical lines with the mouse and moving the mouse left or right.

13. Use the mouse to grab and move the vertical line of cursor *C1*.

14. Repeat Step 13 for cursor *C2*.

The NI ELVIS II Oscilloscope cursors are actually measurement points. They measure the absolute instantaneous voltage of the signal on either Channel 0 or Channel 1 (the default is set to Channel 0 for both cursors). And, they measure the time difference between them. This information is displayed just below the signal display using brown text on the line labelled "Cursors:" and is highlighted in Figure 6 below.



```
NI ELVISmx Oscilloscope
Sample Rate: 100 kS/s

Cursors:      C1: 3.11 mV    C2: 3.43 mV       dT: 3.83 ms
CH 0 Meas:    RMS: 3.659 V   Freq: 108.789 Hz  Vp-p: 4.917 V
CH 1 Meas:    RMS: 1.079 V   Freq: 124.811 Hz  Vp-p: 4.923 V

Channel 0 Settings: Source SCOPE CH 0, Enabled, Probe 1x, Coupling DC, Scale Volts/Div 1 V, Vertical Position 0
Channel 1 Settings: Source SCOPE CH 1, Enabled, Probe 1x, Coupling DC, Scale Volts/Div 1 V, Vertical Position -5
Timebase: Time/Div 2 ms
Trigger: Type Digital, Source TRIG, Level (V) 2.5
Instrument Control: Device Dev1 (NI ELVIS II), Acquisition Mode Run Continuously
Cursors Settings: Cursors On, C1 CH 0, C2 CH 0
Display Measurements: CH 0, CH1
```

Figure 6

Notice that the absolute voltage of the CH 0 signal at *C1* in Figure 6 is 3.11mv and the absolute voltage of the CH 0 signal at *C2* is 3.43mV. Also notice that the time difference between the cursors is 3.83ms.

Now, to verify your answer to Question 2…

15. Move *C1* to the extreme left of the scope's display.

    Note: This aligns *C1* with the beginning of the sequence on the Sequence Generator module's *X* output.

16. Align *C2* with the next positive edge of the Sequence Generator module's *SYNC* signals.

    Note: This aligns *C2* with the beginning of the next sequence on the Sequence Generator module's *X* output.

17. Note the time difference between the cursors.

    Note: It should be the same as your answer to Question 2. If not, work out which one is wrong.

Ask the instructor to check your work before continuing.

18. Deactivate the scope's cursors.

19. Modify the set-up as shown in Figure 7 below.

Note: Remember that the dotted lines show leads already in place.

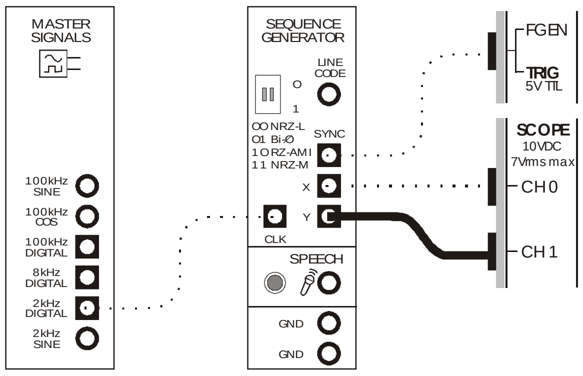


Figure 7

This set-up can be represented by the block diagram in Figure 8 below.

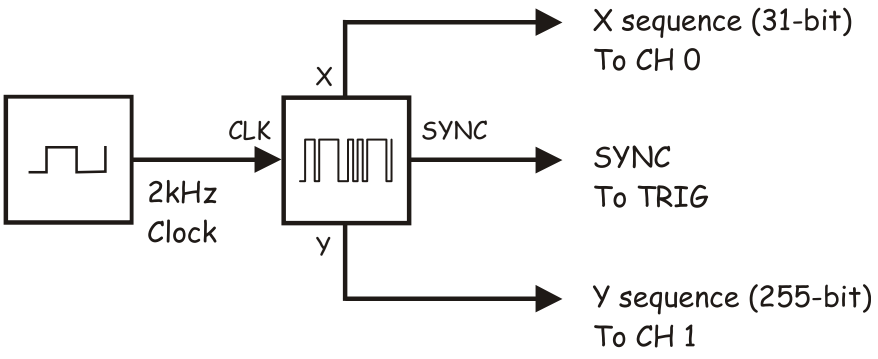


Figure 8

The scope will now be showing the Sequence Generator module's two sequences. However, you'll notice that the scope cannot trigger on the *Y* sequence. This is because the scope triggers on the first bit of the *X* sequence on every repetition using the Sequence Generator module's *SYNC* signal. As the *Y* sequence is 255 bits long, and as 255 ÷ 31 is not a whole number, every sweep of the scope's display starts at a different point in the *Y* sequence resulting in a different pattern for it on every sweep.

Ask the instructor to check your work before continuing.

Part B – Observations of PN sequences in the frequency domain
The next part of the experiment gets you to examine the spectral composition of the 31-bit and a 255-bit Sequence Generator module's *X* and *Y* outputs. But first, a little preparation is necessary.

Question 3
Calculate the frequency of the first four nulled harmonics in the Sequence Generator module's *X* sequence? **Tip:** If you're not sure how to calculate this, re-read the preliminary discussion.

As the bit-clock is 2.083kHz, the first four nulls occur at 2.083kHz, 4.166kHz,

6.249kHz and 8.332kHz.

Let's verify your answer using the NI ELVIS II Dynamic Signal Analyzer.

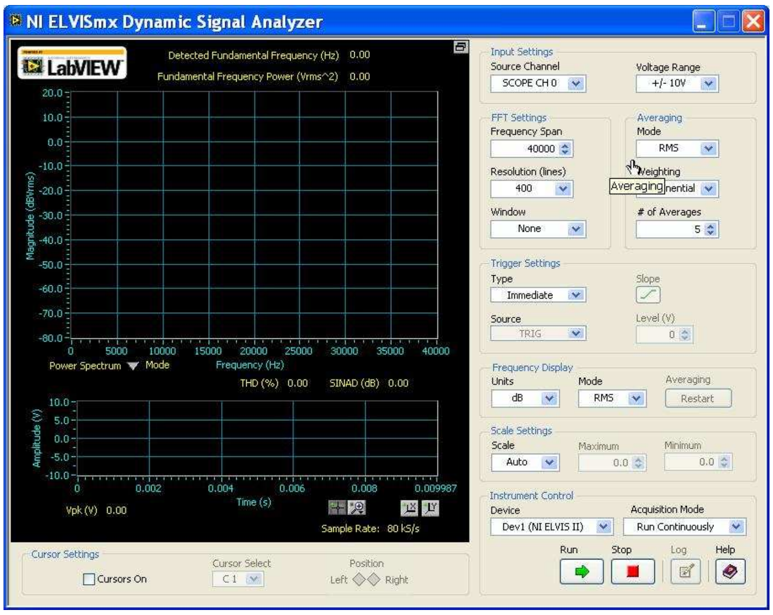
20. Suspend the scope's operation by clicking on its *Stop* control once.

    Note: The scope's display should freeze and its hardware has been deactivated. This is a necessary step as the scope and signal analyzer share hardware resources and so they cannot be operated simultaneously.

21. Minimise the scope's VI.

22. Launch and run the NI ELVIS II Dynamic Signal Analyzer VI.

Note: If the Dynamic Signal Analyzer VI has launched successfully, the instrument's window will be visible (see Figure 9).


NI ELVISmx Dynamic Signal Analyzer

Detected Fundamental Frequency (Hz) 0.00
Fundamental Frequency Power (Vrms^2) 0.00

Input Settings
Source Channel: SCOPE CH 0
Voltage Range: +/- 10V

FFT Settings
Frequency Span: 40000
Resolution (lines): 400
Window: None

Averaging
Mode: RMS
Weighting: Averaging nential
# of Averages: 5

Trigger Settings
Type: Immediate
Slope
Source: TRIG
Level (V): 0

Frequency Display
Units: dB
Mode: RMS
Averaging: Restart

Scale Settings
Scale: Auto
Maximum: 0.0
Minimum: 0.0

Instrument Control
Device: Dev1 (NI ELVIS II)
Acquisition Mode: Run Continuously

THD (%) 0.00 SINAD (dB) 0.00
Vpk (V) 0.00
Sample Rate: 80 kS/s

Cursor Settings
Cursors On
Cursor Select: C 1
Position: Left Right

Figure 9

23. Adjust the signal analyzer's controls as follows:

Input Settings

- *Source Channel* to *SCOPE CH 0*
- Voltage Range to ±10V

FFT Settings

- *Frequency Span* to *40,000*
- *Resolution* to *400*
- *Window* to *7 Term B-Harris*

Averaging

- *Mode* to *RMS*
- *Weighting* to *Exponential*
- *# of Averages* to *3*

Trigger Settings

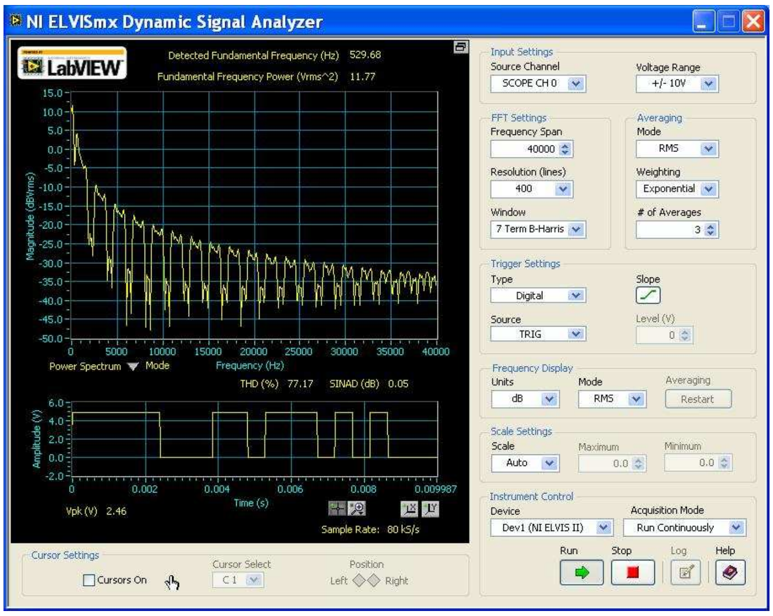
- *Type* to *Digital*

Frequency Display

- *Units* to *dB*
- *Mode* to *RMS*
- *Scale* to *Auto*

- *Cursors On* box unchecked (for now)

Once adjusted correctly, the signal analyzer's display should look like Figure 10 below.



NI ELVISmx Dynamic Signal Analyzer

Detected Fundamental Frequency (Hz) 529.68
Fundamental Frequency Power (Vrms^2) 11.77
THD (%) 77.17 SINAD (dB) 0.05
Vpk (V) 2.46
Sample Rate: 80 kS/s

Input Settings: Source Channel SCOPE CH 0; Voltage Range +/- 10V
FFT Settings: Frequency Span 40000; Resolution (lines) 400; Window 7 Term B-Harris
Averaging: Mode RMS; Weighting Exponential; # of Averages 3
Trigger Settings: Type Digital; Source TRIG; Slope; Level (V) 0
Frequency Display: Units dB; Mode RMS; Averaging Restart
Scale Settings: Scale Auto; Maximum 0.0; Minimum 0.0
Instrument Control: Device Dev1 (NI ELVIS II); Acquisition Mode Run Continuously
Cursor Settings: Cursors On; Cursor Select C 1; Position Left Right

Figure 10

If you've not used the signal analyzer before, its display may need a little explaining here. There are actually two displays, a large one on top and a much smaller one underneath. The smaller one is a time domain representation of the input (in other words, the display is a scope).

The larger of the two displays is the frequency domain representation of the 31-bit sequence on the Sequence Generator module's *X* output. The humps or lobes represent groups of the sinewaves and, as you can see, they follow the general pattern of the Sinc Function.

Question 4
Why are there many more lobes in the *X* sequence's spectrum than suggested in Figures 1, 2 and 3 of the preliminary discussion?

Because Figures 1, 2 and 3 are truncated for discussion purposes. Theoretically, the graphs' x-axes and the SINC functions continue to infinity.

If you have used the signal analzyer before and have also used its cursors, go directly to Step 31 on the next page.

24. Activate the signal analyzer's cursors by checking (that is, ticking) *Cursors On* box.

    Note: When you do, green horizontal and vertical lines should appear on the signal analyzer's frequency domain display.

The NI ELVIS II Dynamic Signal Analyzer has two cursors *C1* and *C2* that default to the left most side of the display when the signal analyzer's VI is launched. Like the scope's cursors, they're repositioned by "grabbing" their vertical lines with the mouse and moving the mouse left or right.

25. Use the mouse to grab and move the vertical line of cursor *C1*.

    Note: As you do, notice that cursor *C1* moves along the signal analyzer's trace and that the vertical and horizontal lines move so that they always intersect at *C1*.

26. Repeat Step 25 for cursor *C2*.

    Note: Fine control over the cursors' position is obtained by using the cursor's *Position* control in the *Cursor Settings area (below the display*).

The NI ELVIS II Dynamic Signal Analyzer includes a tool that measures the difference in magnitude and frequency between the two cursors. This information is displayed in green between the frequency and time domain displays.

27. Move the cursors while watching the measurement readout to observe the effect.

28. Position the cursors so that they're on top of each other and note the measurement.

    Note: When you do, the measurement of difference in magnitude and frequency should both be zero.

Usefully, when one of the cursors is moved to the extreme left of the display, its position on the X-axis is zero. This means that the cursor is sitting on 0Hz. It also means that the measurement readout gives an absolute value of frequency for the other cursor. This makes sense when you think about it because the readout gives the difference in frequency between the two cursors but one of them is zero.

29. Move *C2* to the extreme left of the display.

30. Move *C1* to any point on any of the spectrum's lobes.

    Note: The readout will now be showing you the frequency of a sinewave at that point in the lobe.

31. Use the signal analyzer's *C1* cursor to check that the frequencies you listed in your answer to Question 3 are indeed nulled as you predicted.

✓ Ask the instructor to check your work before continuing.

Question 5
Theoretically, how many harmonics make up each lobe in the spectrum of the Sequence Generator module's *X* output?

Sequence length = 31 bits so 31 harmonics. That said, the last one is nulled.

Question 6
Why can't you see each of the harmonics individually?

With a span of 40kHz, the signal analyzer's resolution isn't sufficient to separate them.

32. To verify the answer to your questions, set the signal analyzer's *Frequency Span* to 2,500Hz.

    Note: Once the signal analyzer's display has updated, you should see a number of discrete sinewaves representing all of the harmonics in the signal's first lobe.

33. Use the signal analyzer's *C1* cursor to locate the null at 2.083kHz.

34. Count the number of significant harmonics in the signal's first lobe.

    Note: Your count should match your answer to Question 5. If not, investigate which one is wrong.

Ask the instructor to check your work before continuing.

Now let's consider a longer sequence. But first, a little more preparation.

Question 7
Calculate the frequency of the first four nulled harmonics in the Sequence Generator module's *Y* sequence?

The bit-clock is still 2.083kHz, so the first four nulls occur at the same frequencies

as before. That is, 2.083kHz, 4.166kHz, 6.249kHz and 8.332kHz.

35. Return the signal analyzer's *Frequency Span* to 40,00Hz.

36. Set the signal analyzer's *Source Channel* to *SCOPE CH 1*.

37. Use the signal analyzer's *C1* cursor to verify your answer to Question 7.

Question 8
Theoretically, how many harmonics make up each lobe in the spectrum of the Sequence Generator module's *Y* output?

Sequence length = 255 bits so 255 harmonics with the last one nulled.

38. See if you can verify your answer to Question 8 by setting the signal analyzer's *Frequency Span* to 2,500Hz to examine the spectral composition of the signal's first lobe.

Question 9
Why can't you accurately count the harmonics in the signal's first lobe?

There are too many and so they're too close together. (Note: The resolution can be improved by reducing the span but you would lose the upper portion of the lobe.)

Question 10
Which of the Sequence Generator module's two sequences has the greater harmonic content?

The *Y* sequence.

39. As there are 255 harmonics in the each of the signal's lobes, they should be about 8.16Hz apart (2.083kHz ÷ 255). Reduce the signal analyzer's *Frequency Span* to see if you can measure this separation between them using the cursors.

    Note: You may need the instructor's help with adjusting some of the signal analyzer's other controls.

Ask the instructor to check your work before continuing.

Part C – Using PN sequences to generate noise
Generating the theoretical proposition of white noise is impossible. To explain, an infinite number of sinewaves (with or without equal power density) would require an infinite amount of power! However, as you have just seen, long PN sequences are rich in harmonics. Moreover, although the spectrum of PN sequences have lobes of changing amplitude, small portions of its spectrum are relatively flat (a point you may have noticed at Step 38). That being the case, it's possible to isolate a small portion of a PN sequence's spectrum using a filter to model band-limited white noise. The next part of this experiment demonstrates this.

40. Suspend the signal analyzer's VI.

41. Completely dismantle the current set-up.

42. Launch and run the NI ELVIS II Function Generator VI.

43. Adjust the function generator for a 150kHz output.

    Note: It's not necessary to adjust any other controls as the function generator's *SYNC* output will be used and this is a digital signal.

44. Connect the set-up shown in Figure 11 below.

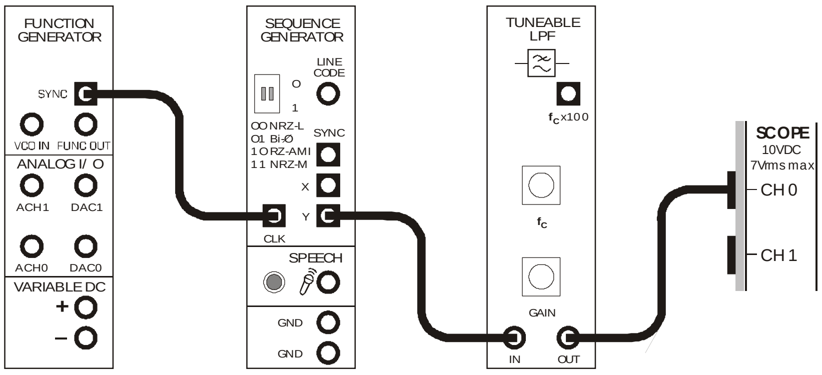


Figure 11

The set-up in Figure 11 can be represented by the block diagram in Figure 12 below. A quick mathematical analysis tells us that, with a 150kHz bit-clock, the spectral composition of the Sequence Generator module's *Y* output includes 255 sinewaves per lobe and so they are separated by 588Hz.

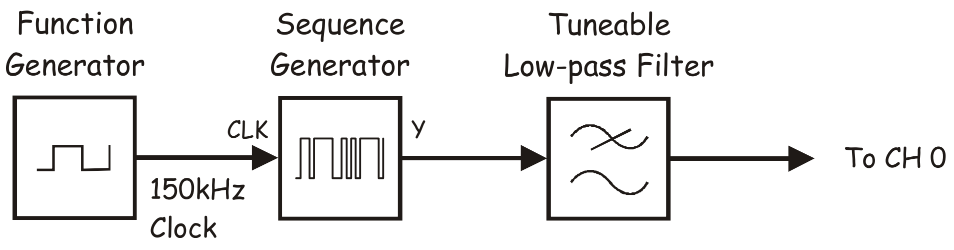


Figure 12

45. Launch the DATEx soft front-panel (SFP) and check that you have soft control over the DATEx board.

46. Locate the Tuneable Low-pass Filter module on the DATEx SFP and set its soft *Gain* to about the middle of its travel.

47. Turn the Tuneable Low-pass Filter module's soft *Cut-off Frequency Adjust* control fully clockwise.

    Note: This sets the Tuneable Low-pass Filter module's cut-off frequency to 15kHz.

48. Restart the scope's VI.

49. Adjust the scope to view the Tuneable Low-pass Filter module's output. Essential scope settings include:

    - *Timebase* to *1ms/div*
    - *Trigger Type* to *Immediate*
    - CH 1 deactivated

50. Observe the signal.

Question 11
What does the signal on the Tuneable Low-pass Filter module's output look like?

Electrical noise.

Ask the instructor to check your work before continuing.

The signal on the Tuneable Low-pass Filter module's output isn't "white" noise because it is bandwidth limited. Nor is the signal truly "noise". This can be demonstrated using the scope. True noise is non-repetitive. However, the signal on the Tuneable Low-pass Filter module's output repeats itself every 1.7ms. [This figure is calculated using the bit-clock's period (1÷150,000Hz) and multiplying it by the PN sequence's number of bits (255).] The repetitive nature of the "noise" you have modelled can be observed using the scope.

51. Suspend the scope's VI to stop the signal from jumping around on the screen.

52. Look closely at the signal - You should see it repeat itself about 5 times.

53. Activate the scope's cursors and use them to measure the signal's period.

    Note: You should find it is close to the figure quoted above.

Question 12
Given the signal on the Tuneable Low-pass Filter module's output is repetitive, what's a better name for it than "noise"?

Pseudo-noise.

Ask the instructor to check your work before continuing.

Question 13
How many sinewaves fall inside the Tuneable Low-pass filter's 15kHz pass-band?

25.5

---

54. Restart the signal analyzer's VI and make the following adjustments:

    - *Source Channel* to *SCOPE CH 0*
    - *Frequency Span* to *20,000Hz*
    - *Trigger Type* to *Immediate*

55. Use the signal analyzer's *C1* cursor to indicate on the screen the Tuneable Low-pass Filter's cut-off frequency.

56. Count the number of significant harmonics between 0Hz and the filter's cut-off frequency.

    Note: Your count should match your answer to Question 13. If not, work out which one is wrong.

57. Suspend the signal analyzer's VI.

58. Restart the scope's VI.

59. Modify the set-up as shown in Figure 13 below.

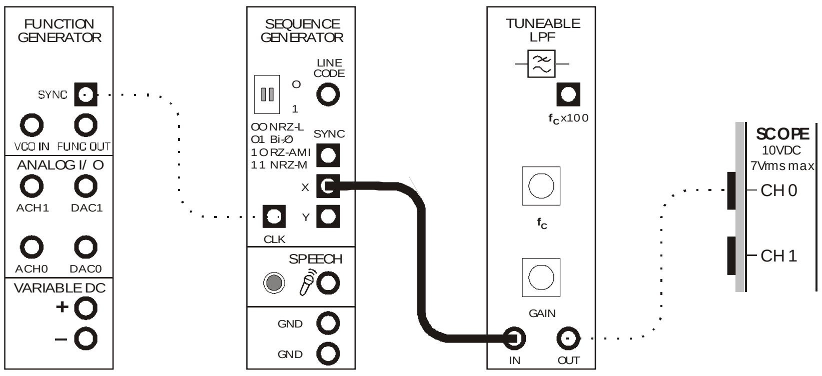


Figure 13

60. Observe the Tuneable Low-pass filter module's new output signal.

Question 14
Why doesn't the signal look like electrical noise any more?

Because the Sequence Generator module's *X* output is only a 31-bit sequence which

means that there are only 3 or 4 significant sinewaves inside the filter's pass-band.

61. To verify your answer to Question 14, suspend the scope's VI.

62. Restart the signal analyzer's VI.

63. Count the number of significant harmonics between 0Hz and the filter's cut-off frequency.

Ask the instructor to check
your work before finishing.

Why are the DATEx Sequence Generator module's outputs 31 and 255 bits long?
You may be wondering why the length of the sequences for the DATEx Sequence Generator module's outputs are 31 and 255 bits long. To explain, shift register circuits known as *linear feedback shift registers* have been used to generate them. These circuits are similar to ring counters in that they recycle data through the shift register. However, they also include feedback via exclusive OR gates at key points to change the data word as it cycles through. That said, this doesn't generate purely random numbers as the data pattern must repeat. However, certain feedback connections create particularly long sequences and these are known as *maximal length sequences* (MLS).

Maximal length sequences have several interesting properties:

i) They have almost equal 1s and 0s…(actually 1 less 0 than 1s)

ii) They have equal number of "runs" of 1s and 0s (you can readily see this using the Sequence Generator module's 31-bit sequence

iii) They create a spectrum with no missing harmonics

iv) They never repeat within themselves (unlike the sequences mentioned at bottom of page 14-3)

These properties make maximal length sequences ideal for PN sequences used for applications like encryption, encoding etc. Naturally then, linear feedback shift registers generating maximal length sequences are used for the Emona DATEx Sequence Generator module.

For the *X* output, a 5-bit shift register is used making it 31 bits long (that is, 2^5-1) and for the *Y* output an 8-bit bit shifter register is used giving 255 bits (that is, 2^8-1).

Emona DATEx VOLUME 3

Contents

1 – Introduction

The EMONA DATEx board uses a block diagram approach to building telecommunications experiments. The individual blocks are wired together in accordance with the block diagram to build simple and complex modulation systems. LabVIEW is a graphical programming language in which graphical programming blocks are wired together on screen to build simple and complex programs. As well there are blocks which represent real hardware instruments, which a program can directly interact with and control.

The EMONA DATEx board has a PC control mode whereby a LabVIEW program can directly interact with and control various hardware circuits on the DATEx board.

This manual is designed as a guide to using LabVIEW to interact with and control the various DATEx blocks to build telecommunications experiments. As well, users will see examples of how to program with ELVISmx blocks to create their own custom instruments relating to DATEx based telecommunications experiments. This manual assumes that the user has a basic understanding of using LabVIEW. Information and tutorials for learning LabVIEW programming are available at:

http://www.ni.com/academic/labview_training/ and there are a number of resources available at:

http://cnx.org/content/col10629

Whether you are an introductory or advanced user of LabVIEW, using LabVIEW with the DATEx board is an interesting and highly interactive hands-on opportunity to experiment in telecommunications. The analog and digital, input and output functions from the ELVIS unit, as well as the many independent circuit blocks of the DATEx board enable a very wide range of experimental set-ups to be created.

LabVIEW Control of DATEx Hardware Overview

In Volume 1 & 2 of the DATEx Lab Manuals, the student has controlled the DATEx hardware functional blocks via the DATEx SFP in a manual and non-programmatic manner. In Volume 3 the student will learn to access and control the "low-level" LabVIEW blocks for the various DATEx hardware functions. These "low-level" LabVIEW blocks are described in detail in this manual.

The first sections of this manual provide simple introductions to programming the "low-level" blocks. Later sections give more complex examples and exercises of hardware/software systems.

DATEx is an ideal LabVIEW programming target for students to learn about controlling real hardware. The DATEx functional blocks provide functionality with which the student is already familiar, so control programs can be tried out manually and then developed and debugged progressively.

As the DATEx board is an add-on board for the NI ELVIS unit, the interface to the DATEx blocks is via the DAQmx blocks of the NI ELVIS unit. Your LabVIEW control program will communicate with the DATEx board via several specific lines of the NI ELVIS. The instant in which commands are sent to the board can be visually confirmed using the onboard COMMS LEDs on the lower DATEx circuit board. This simplifies debugging by confirming that the command did or did not make it to the DATEx board as expected. These LEDs are shown in Figure 1.

Individual commands are sent to the DATEx board from the LabVIEW program via DAQmx, and the circuitry on the DATEx board responds accordingly. These commands can only be sent sequentially with a maximum rate of about 7 commands per second. This will be the upper limit on the rate at which you can change the parameters of a particular control on the DATEx board. The lines used by DATEx for communication are Port 2, lines 4, 5 & 6 from the ELVIS unit. These are reserved and should not be used by your LabVIEW program.

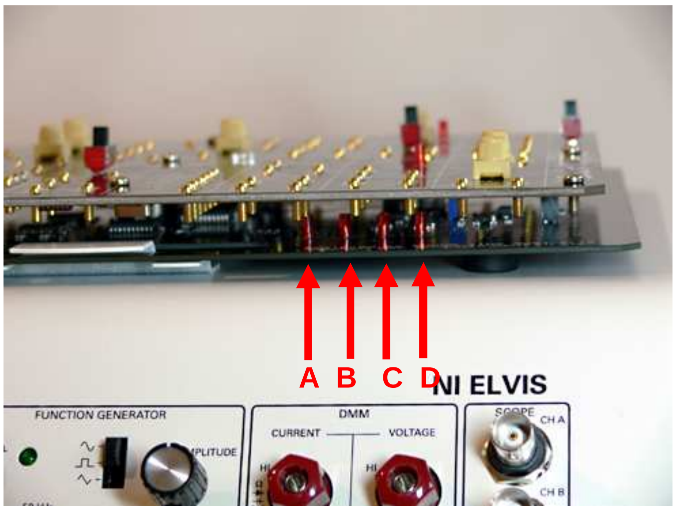


Figure 1: COMMS LEDs on the board

Confirming communications to the DATEx using MAIN SFP VI:

1. Slide the DATEx mode switch to the PC CONTROL position and RUN the MAIN SFP VI.

2. Click on the on-screen TDM switch and confirm the TDM LED changes and the COMMS LEDs flash.

The MAIN SFP is designed to mimic the look and feel of the actual DATEX hardware board. It is a large scale front panel which can be used alongside the various instruments from the NI ELVIS launcher.

Using Prewired Backgrounds on the DATEx MAIN SFP

In a typical experiment the student will progressively and systematically wire up a number of modules to build a particular experiment according to the block diagram. Each of these stages is described step by step in figures throughout the Volume 1 & 2 of the manuals. It is possible to load different backgrounds into the MAIN SFP which correspond to each of these wiring stages. In this way the SFP will mimic the DATEx board along with the wiring required for that stage. Figure 2 shows two of these "pre-wired backgrounds".

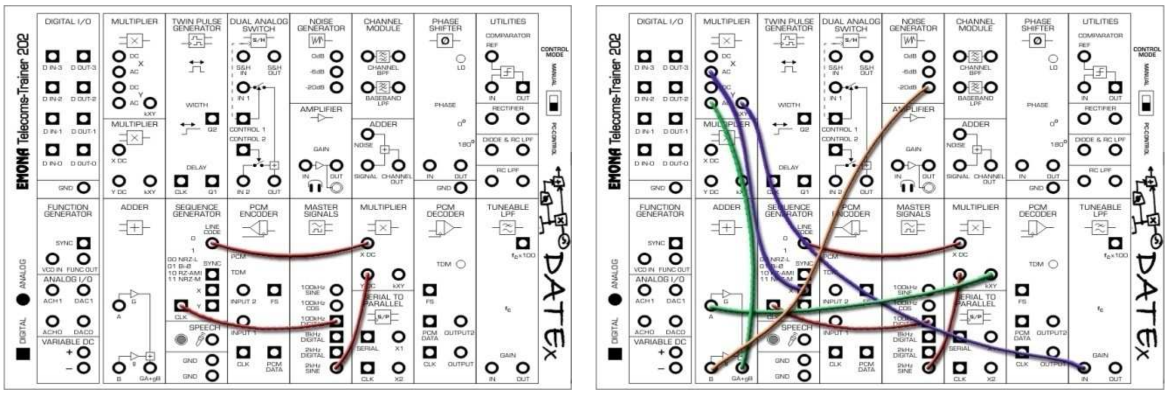


Figure 2: Two examples of "prewired background" images

These backgrounds are supplied for every figure in Volume 1 on the supplied CD in directory "Experiment Wiring". They are arranged in folders labeled after the individual chapter number of the Lab Manual Volume 1.

Since the background is simply a graphic file, the default background is available on the CD as "Emona-DATEx-MAIN.bmp". The user can edit this file to add their own images or text comments in order to further assist students in completing the experiment. These backgrounds can be saved as either .bmp or .jpg files. An example of an annotated background is shown in Figure 3:

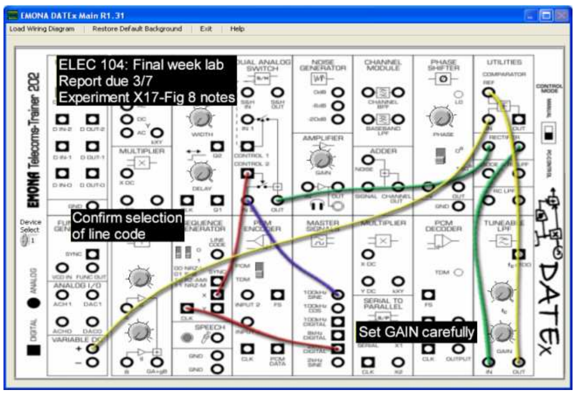


Figure 3: Example of an annotated and prewired background image

To create your own custom backgrounds, open a copy of the default background file "Emona-DATEx-MAIN.bmp" in Paint, or other graphics package, add your edits, then save as a .bmp or .jpg file . Ensure that it is the same size image, i.e. 836 x 566 pixels, for correct dimensioning in the DATEx MAIN SFP.

Loading the images is via the top menu buttons: Load Wiring Diagram

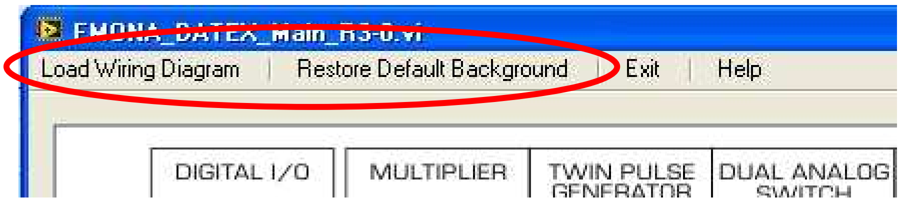


Figure 4: Load and restore image buttons

Saving Screen Space with the DATEx 'Toolbar' SFP

As the DATEx MAIN SFP is designed to mimic the actual DATEx board, it is a large SFP (836 x 566 pixels) graphic. Once users become familiar with the use of the MAIN SFP, they can choose to use a smaller SFP which contains only the control knobs and switches from the DATEx. This smaller SFP (421 x 161 pixels) graphic fits easily onto the screen alongside other NI ELVIS instrument panels. Its functionality is exactly the same as for the MAIN SFP.

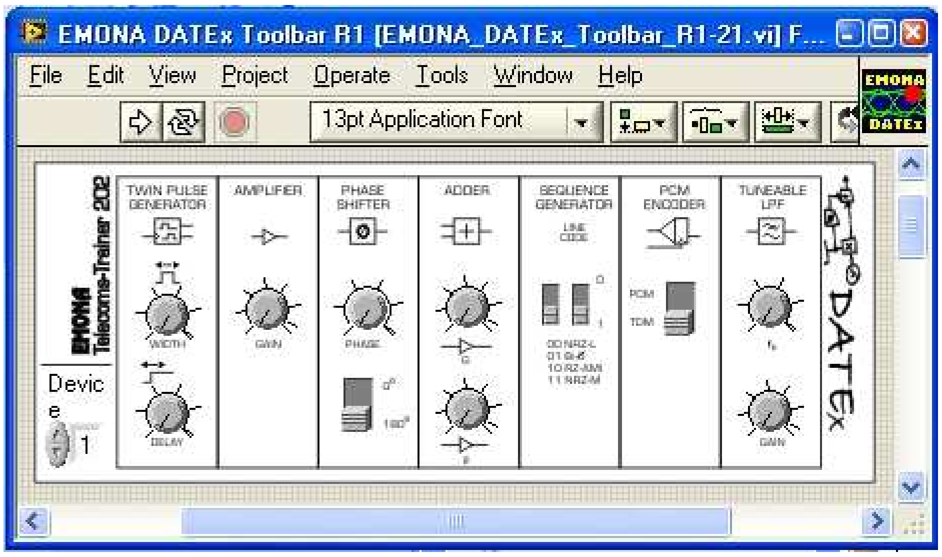


Figure 5: DATEx "Toolbar" SFP

To confirm correct operation of the Toolbar SFP:

1. Run the SFP toolbar and select PC CONTROL from the mode switch on the DATEx board.

2. Click the onscreen PCM/TDM switch and confirm the TDM led changes and COMMS LEDs flash each time.

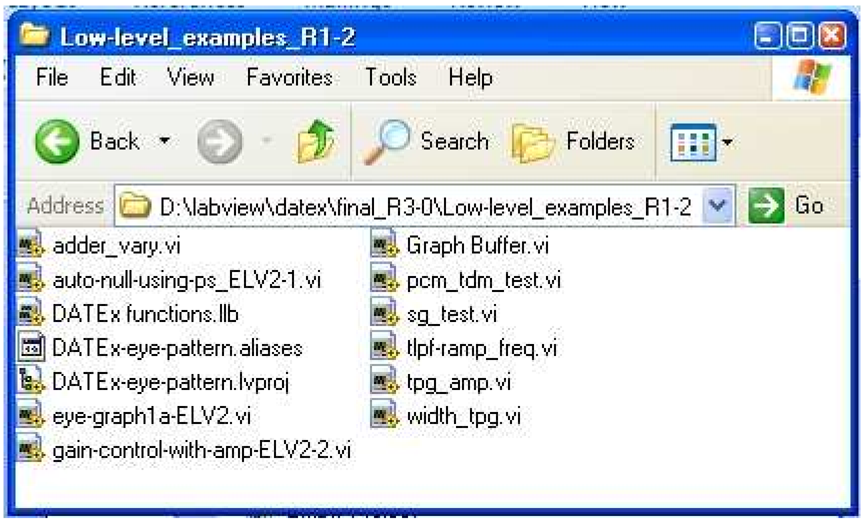
Equipment required for running the "low-level" DATEx VIs

- PC with LabVIEW 8.5 (or later) and DATEx software installed
- NI ELVIS I or II/+ unit connected to PC with software installed.
- Emona DATEx experimental add-in board
- Three BNC to 2mm scope leads
- Assorted 2mm banana-plug patch leads

Low Level DATEx VIs

There are 7 LabVIEW controllable hardware blocks on the DATEx board. The use of each one of them will be described individually, and later in this manual the combined use of these will be discussed.

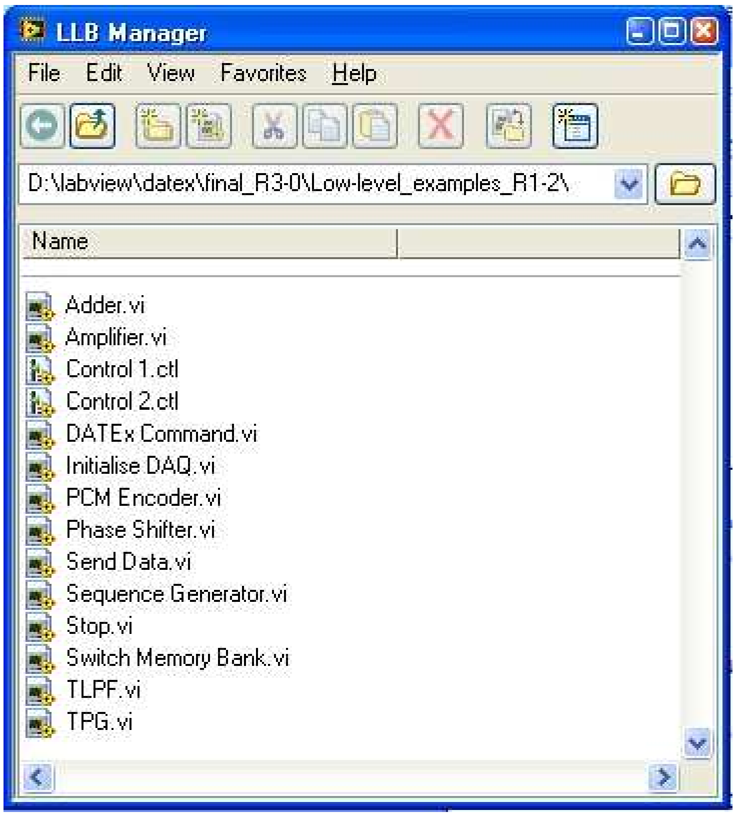
On the CD supplied with the DATEx board are located the "low level" VIs for programmatic control of the DATEx blocks. These are found in the "low level examples…" directory of the CD. Figure 6 shows the directory listing for these VIs and their examples.


Low-level_examples_R1-2
File Edit View Favorites Tools Help
Back Search Folders
Address D:\labview\datex\final_R3-0\Low-level_examples_R1-2 Go

adder_vary.vi
auto-null-using-ps_ELV2-1.vi
DATEx functions.llb
DATEx-eye-pattern.aliases
DATEx-eye-pattern.lvproj
eye-graph1a-ELV2.vi
gain-control-with-amp-ELV2-2.vi
Graph Buffer.vi
pcm_tdm_test.vi
sg_test.vi
tlpf-ramp_freq.vi
tpg_amp.vi
width_tpg.vi

Figure 6: Directory listing of the "low level examples" folder on the CD

The DATEx VIs are supplied as a library "DATEx functions.llb" containing the main DATEx COMMAND VI as a polymorphic VI, along with individual VIs and supporting sub VIs. Also shown in this directory are some examples for using low level DATEx VIs which will be discussed later in this manual.



D:\labview\datex\final_R3-0\Low-level_examples_R1-2\

Adder.vi
Amplifier.vi
Control 1.ctl
Control 2.ctl
DATEx Command.vi
Initialise DAQ.vi
PCM Encoder.vi
Phase Shifter.vi
Send Data.vi
Sequence Generator.vi
Stop.vi
Switch Memory Bank.vi
TLPF.vi
TPG.vi

Figure 7: DATEx Library "DATEx functions.llb"

These are described as follows:

DATEx Command.vi: polymorphic VI for all DATEx blocks.

Adder, Amplifier, PCM Encoder, Phase Shifter, Sequence Generator, TLPF and TPG VIs are individual DATEx block VIs: these are individual versions of those contained in the polymorphic VI.

Initialize DAQ.vi: set up DAQ mx

Stop.vi: stop DAQmx

Control1, Control2, Send Data: internal function sub VIs

Switch memory bank.vi: used with Sequence Generator and PCM VIs.

To access the VIs in the library simply double click on the .llb file and the LLB MANAGER will display as shown above in Figure 7.

Example: Introducing complex I/Q modulation using LV Modulation Toolkit

In this experiment we will introduce the use of complex IQ modulation methods which can be used for a variety of modulation schemes. In previous experiments documented in Volume 1 and 2, Amplitude Modulation is implemented with two different models. A similar but more generic block diagram is used in this experiment as shown in Figure 1.

If the signal output from DAC 0 is the message signal plus DC, and the signal output from DAC 1 is 0 V, then the output of the block diagram in figure 1 will be AM. Refer back to previous experiments to confirm this for yourself.

You can see that the arrangement in Figure 1 is the standard block diagram for a quadrature modulated scheme. There are two baseband signals, I and Q, which are respectively multiplied by quadrature carriers sin(wt) and cos(wt). These two products are then added to form the final output in the passband. The multiplication and addition functions are implemented using hardware DATEx blocks. The data modulator, is implemented using LabVIEW and output from the Analog I/O block, as DAC 0 and DAC 1, on the DATEx board.

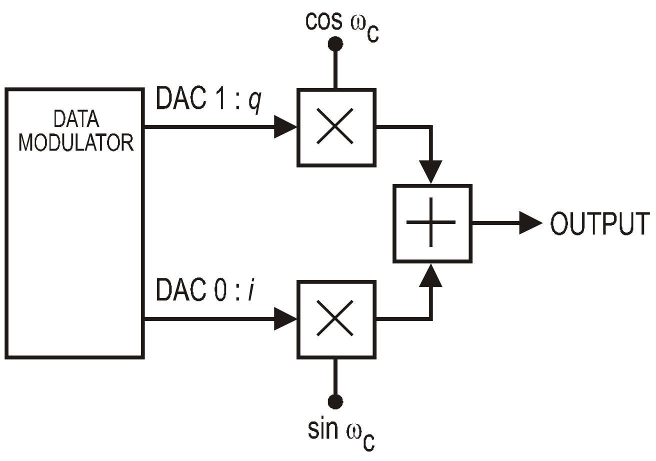


Figure 40: I/Q quadrature modulation

In this way, producing the appropriate baseband signals from the data modulator enables this block diagram structure to create many different modulation schemes. LabVIEW treats the in-phase (i) component and the quadrature-phase (q) data for the signal as complex data which is processed and output by various blocks in the LabVIEW Modulation Toolkit.

You can think about the complex data pair as simply being the data for two signals simultaneously held. The real part of the data relates to the signal to be multiplied by the in-phase carrier, sin(wt), and the complex part of the data relates to the signal to be multiplied by the quadrature carrier, cos(wt). Thinking about this pair of signals as phasors also helps to understand how this method is used. You may wish to refer back to previous experiments in which phasor diagrams were used and discussed.

Patch together the experiment as follows, referring to Figure 40:

1. DAC 0 and 1 connect to DAC0 and DAC1 outputs on the DATEx ANALOG I/O

2. $sinw_c$ and $cosw_c$ connect to the 100kHz SINE and 1ookHz COS from the MASTER SIGNALS block on DATEx.
3. Select any two of the available MULTIPLIERS and use the DC coupled inputs only.
4. Use the ADDER block which has the dual variable gain knobs. Set both gains to maximum, as actual signal levels will be set by the program.

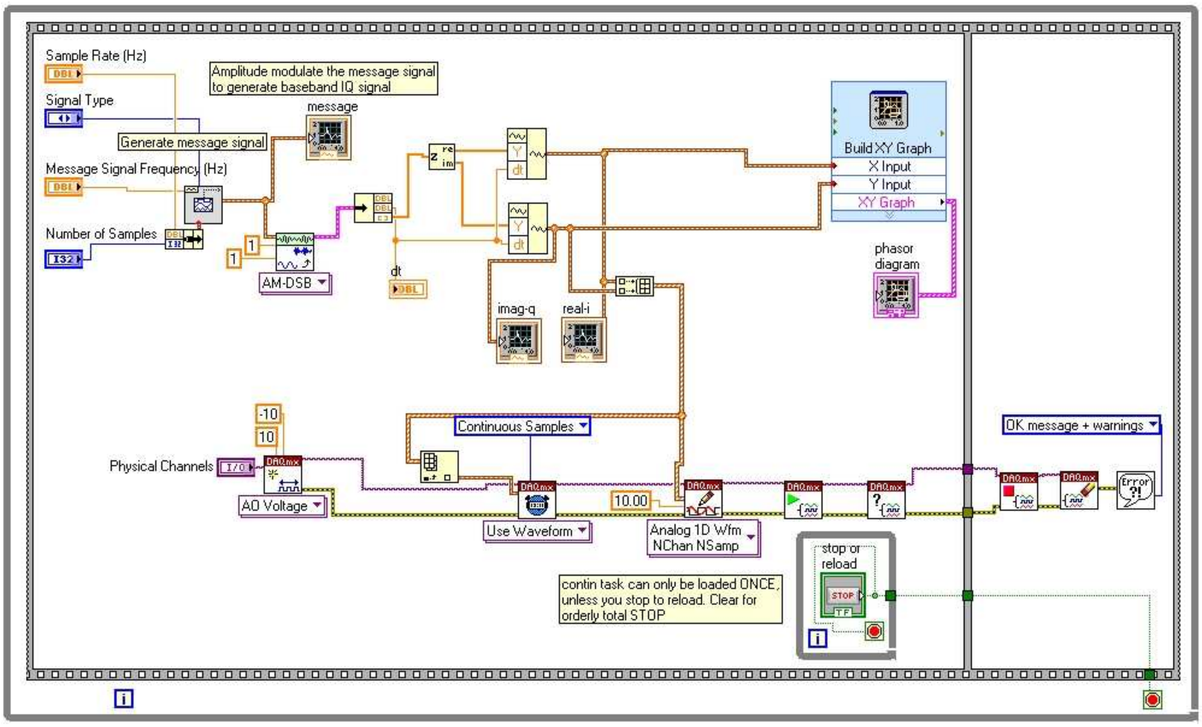
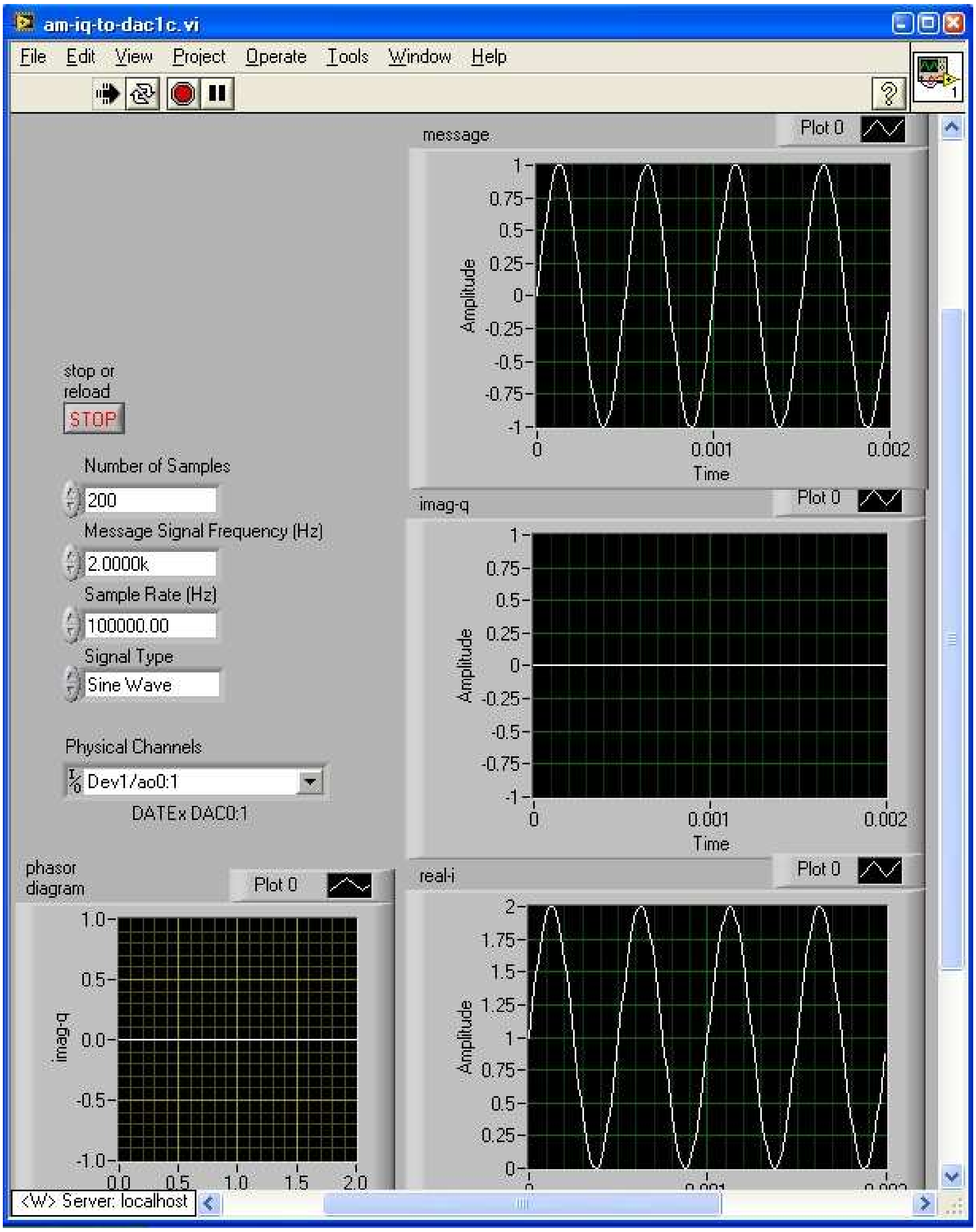


Figure 41: LabVIEW block diagram for AM modulation "am-iq-to-dac.vi"

In the block diagram shown in Figure 41 you can see that a message signal is generated, then passed to an AM-DSB modulator block (from the Modulation Toolkit) which outputs complex data relating to the I and Q baseband signals. As well as being displayed on the soft front panel of this LabVIEW program, this complex signal is separated into its real and imaginary parts and output from the Analog I/O terminals on the DATEx board. The bottom half of the block diagram is an example of how data signals can be output to real hardware. Similar examples of how to output data to the hardware via the ELVISmx functions are available in the "Find examples" section of the LabVIEW Help menu.

Figure 42 is the soft front panel of this program which displays the message signal and the respective I and Q baseband signals, as well as an XY representation of the I and Q signals, known as a phasor diagram.

When using this program remember to stop the program using the front panel "stop and reload" button which will ensure that the task is properly closed. You can then vary the parameters on the front panel and restart the program again. Use the NI ELVIS Scope and Dynamic Spectrum Analyzer (DSA) instruments running in their own separate windows to view signals on the board itself.



stop or reload

Number of Samples: 200

Message Signal Frequency (Hz): 2.0000k

Sample Rate (Hz): 100000.00

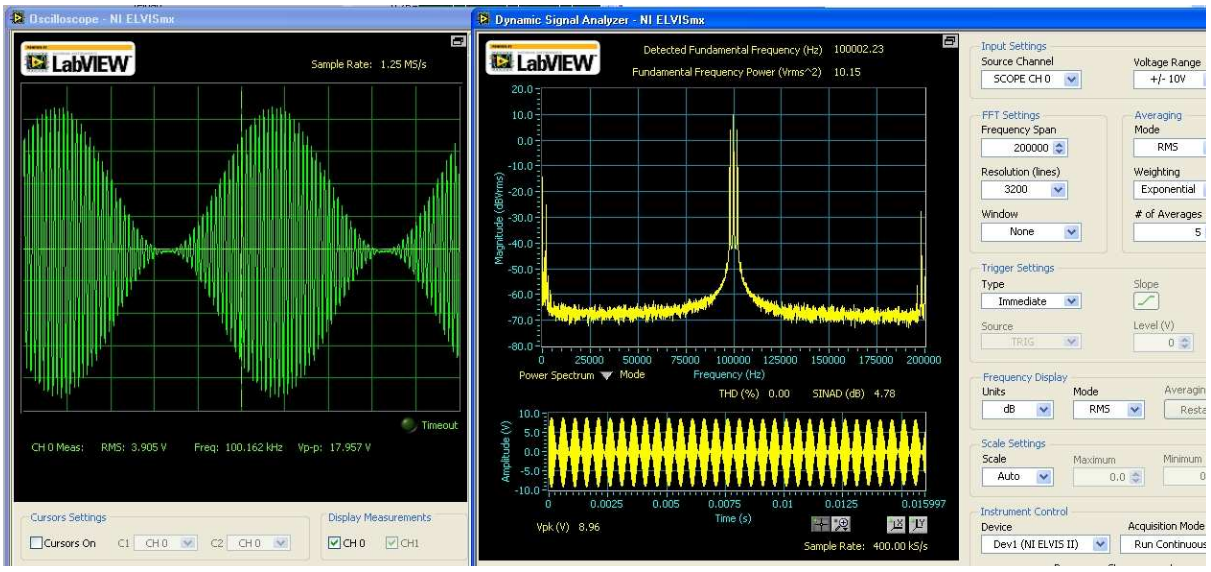
Signal Type: Sine Wave

Physical Channels: Dev1/ao0:1

DATEx DAC0:1

Figure 42: Front panel of the AM experiment

Figure 43 shows screenshots from the NI ELVIS scope and Dynamic Signal Analyzer from the output of the ADDER block. You can see that this signal as a modulation index of one, which corresponds with the use of a real baseband I signal with an amplitude of 1 V and a DC offset of 1 V.



Sample Rate: 1.25 MS/s

CH 0 Meas: RMS: 3.905 V Freq: 100.162 kHz Vp-p: 17.957 V

Cursors Settings: Cursors On ☐ C1 CH 0 C2 CH 0

Display Measurements: ✓ CH 0 ✓ CH1

Detected Fundamental Frequency (Hz) 100002.23

Fundamental Frequency Power (Vrms^2) 10.15

THD (%) 0.00 SINAD (dB) 4.78

Vpk (V) 8.96

Sample Rate: 400.00 kS/s

Input Settings: Source Channel SCOPE CH 0; Voltage Range +/- 10V

FFT Settings: Frequency Span 200000; Resolution (lines) 3200; Window None

Averaging: Mode RMS; Weighting Exponential; # of Averages 5

Trigger Settings: Type Immediate; Slope; Source TRIG; Level (V) 0

Frequency Display: Units dB; Mode RMS; Averagin; Resta

Scale Settings: Scale Auto; Maximum 0.0; Minimum 0

Instrument Control: Device Dev1 (NI ELVIS II); Acquisition Mode Run Continuous

Figure 43: Screenshots from the ELVIS scope and spectrum analyzer

Although in this experiment we are not using the quadrature, Q, branch of the modulator, as it is set to 0 V, it is a simple and worthwhile introduction to the use of these generic quadrature modulation arrangement. Once you are familiar with outputting LabVIEW generated signals to the external hardware there is very little limit to the types of signals to create an experiments that you can implement.

Programming tasks:

TASK 1: Select the other options in the AM block from the Modulation Toolkit used above and investigate the I and Q signals using both the front panel displays and the NI ELVIS scope .

TASK 2: Modify the program to display parameters such as the modulation index.

Emona DATEx™ Telecommunications Trainer Lab Manual Volumes 1, 2 & 3 - SAMPLE MANUAL

Emona Instruments Pty Ltd
78 Parramatta Road
Camperdown NSW 2050
AUSTRALIA

web: **www.emona-tims.com**
telephone: +61-2-9519-3933
fax: +61-2-9550-1378