Emona DATEx
Lab Manual

Volume 3
Programming and Controlling DATEx with NI LabVIEW

Carlo Manfredini

DATEx

Emona DATEx
Lab Manual

Volume 3
Programming and Controlling DATEx
with NI LabVIEW

Carlo Manfredini

Emona DATEx Lab Manual for NI™ ELVIS I and II/+
Volume 3 -
Programming and Controlling DATEx with NI LabVIEW.

Author: Carlo Manfredini
Editor: Alfred Breznik

Issue Number: Vol3 issue 1.1

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

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.

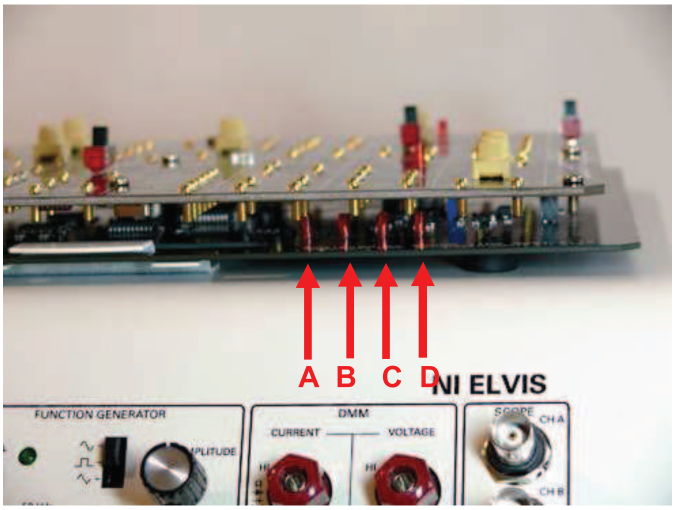


Figure 1: COMMS LEDs on the board

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.

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".

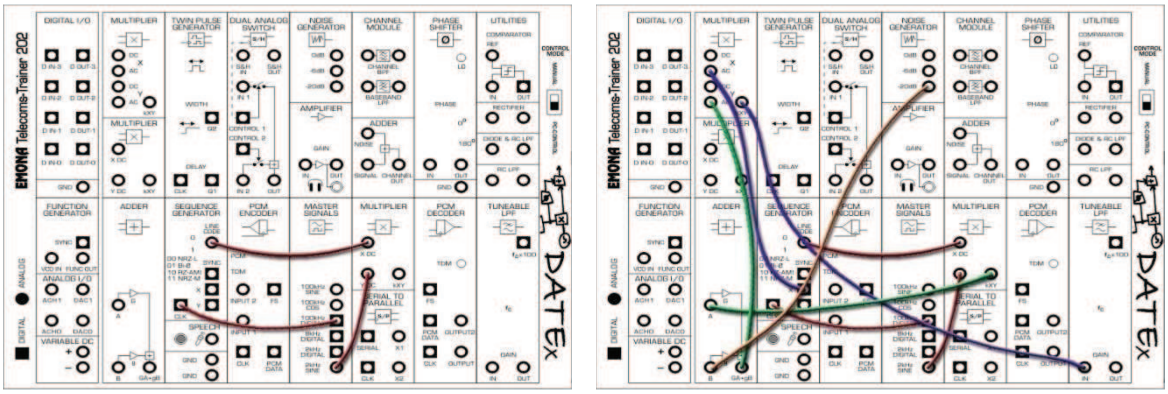


Figure 2: Two examples of "prewired background" images

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:

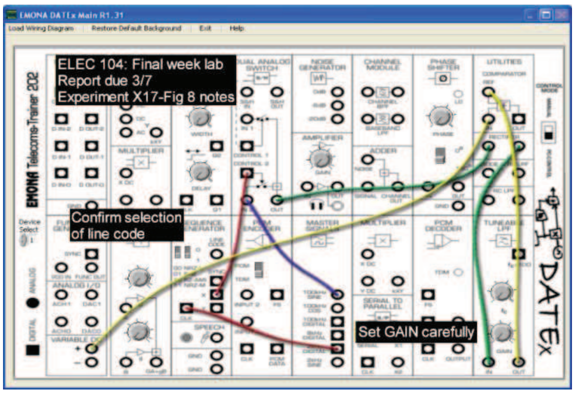


Figure 3: Example of an annotated and prewired background image

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.

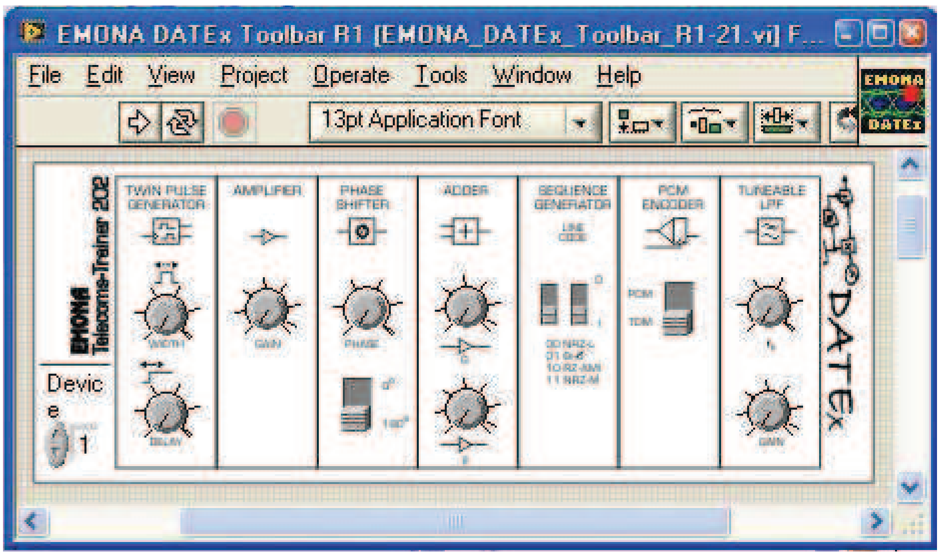


Figure 5: DATEx "Toolbar" SFP

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.

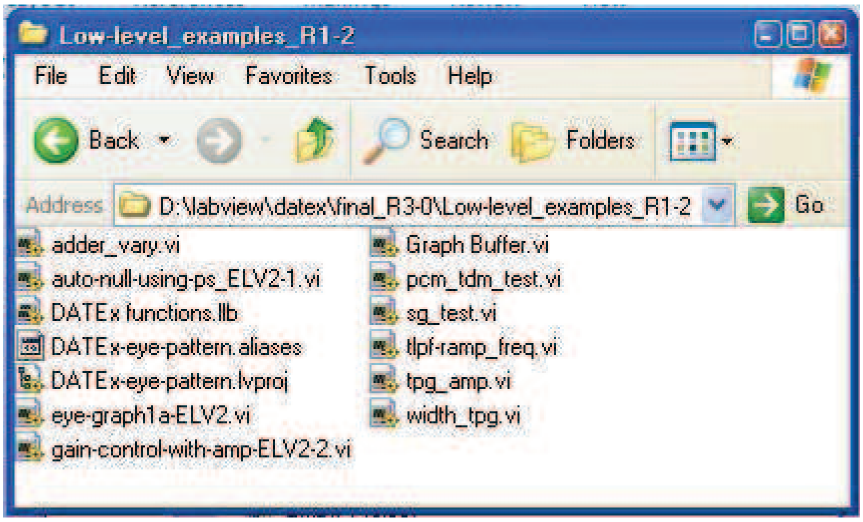


Figure 6: Directory listing of the "low level examples" folder on the CD

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.

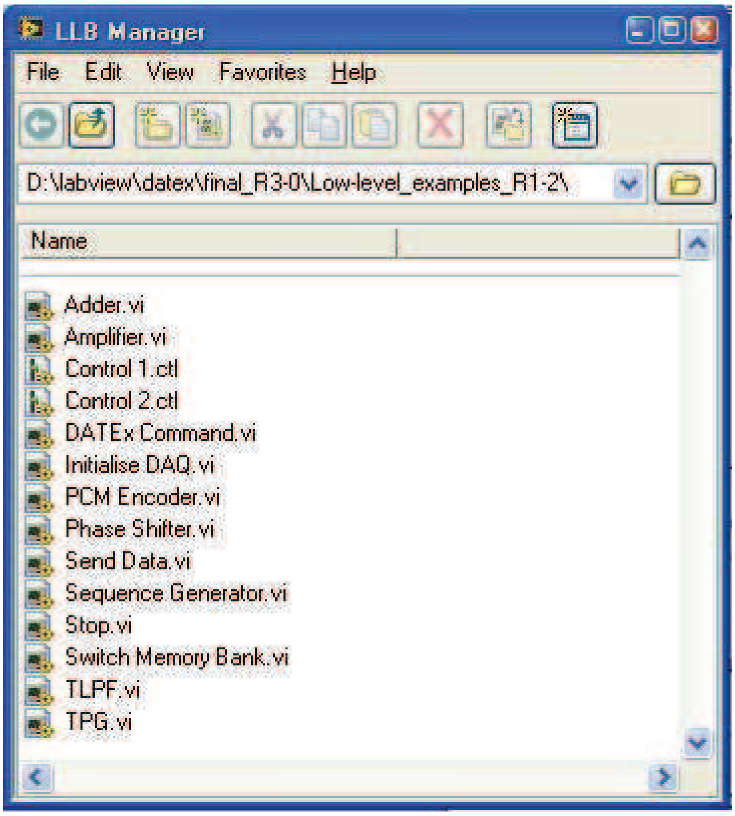


Figure 7: DATEx Library "DATEx functions.llb"

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.

2 - Programming Amplitude Control Blocks

In the transmission model of telecommunications, when dealing with the transmission of a message across a medium, be it radio frequency radiation or any other waveform based medium, there are four parameters of the transmission medium which can be varied: Amplitude, Phase, Frequency and Synchronization.

In this experiment we will investigate the use of the Amplitude control blocks available in the DATEx board. The blocks which control amplitude via the control of the gain stage of an amplifier are the ADDER block and the AMPLIFIER block.

ADDER block:

The ADDER block on DATEx has two inputs and one output. Each input has variable GAIN, and these varied inputs are then summed prior to the output. Hence the parameters available for programmatic control are the two input GAINS. These are known as k1 and k2. Figure 8 shows the ADDER block as a hardware block and its corresponding low-level VI block.

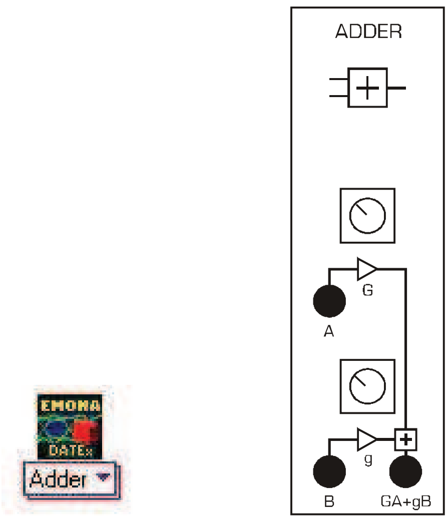


Figure 8: ADDER LabVIEW block, ADDER hardware block

The full range of GAIN from minimum to maximum for each input is set with an input value in the range of 0 – 127. The data type required is U8: Unsigned Integer.

An example for controlling an ADDER block from LabVIEW is shown in Figure 9:

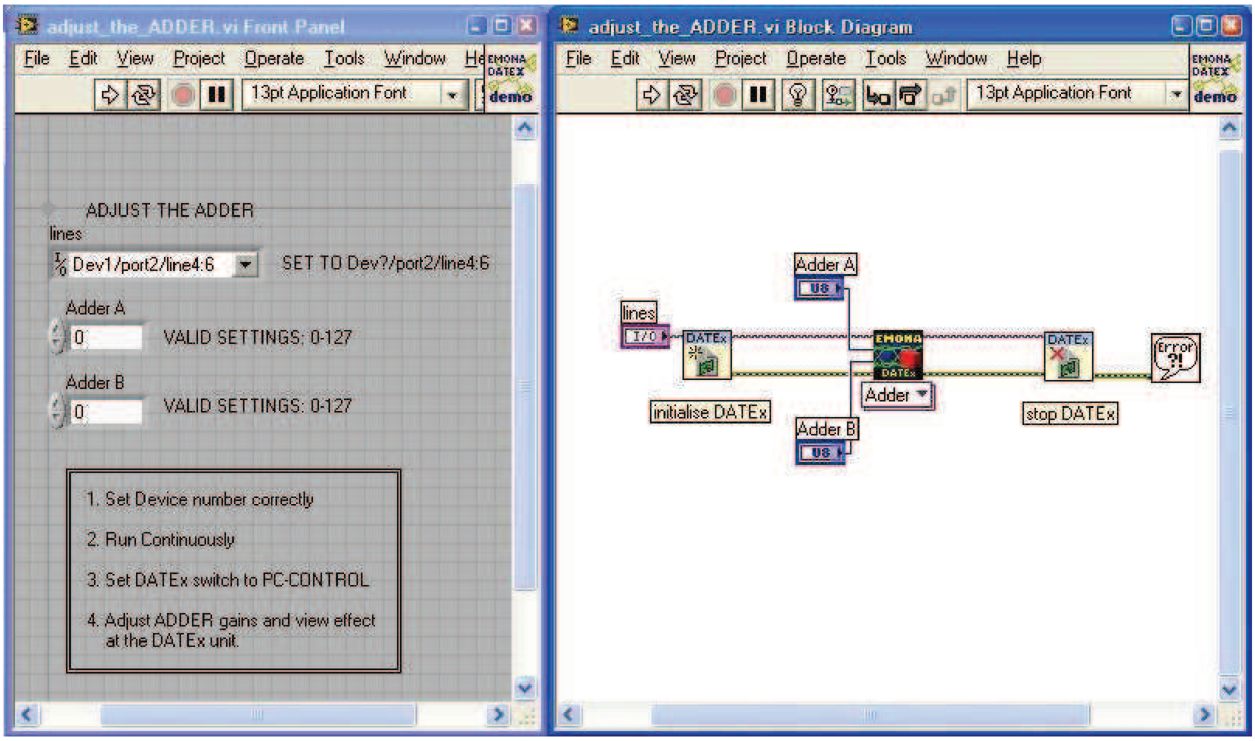


Figure 9: "adjust_the_ADDER.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of Port2.
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select ADDER from its drop down selector)
4) Insert a 'Stop the DAQmx interface' VI upon exit.
5) Add an "error" dialog if required.

In this example the ADDER GAIN is controlled directly from numeric controls on the VI front panel. As for all LabVIEW programs these variables can be supplied from constants, controls or any other program value.

This program will initialize the DAQmx, send one command to the DATEx ADDER, then stop the DAQ mx, and then end. Clearly, it is not necessary to initialize and stop the DAQmx for every command. An improved VI is shown in the next part of this section.

In the figure below you can see the previous program, but with the "one-off" commands outside of the main operating loop.

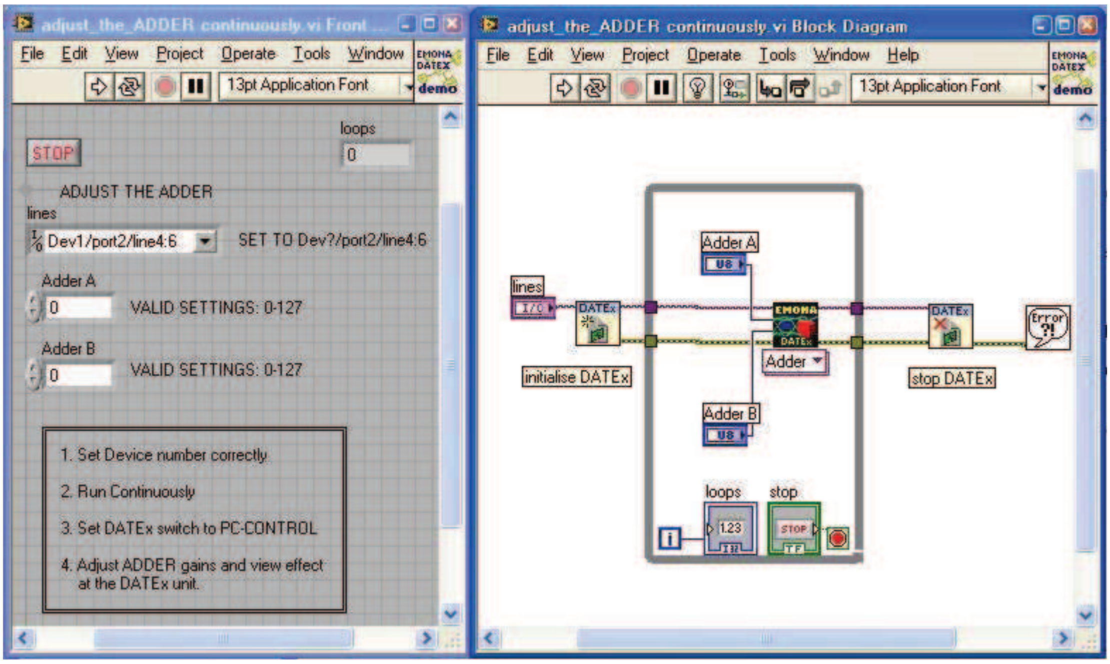


Figure 10: "adjust_the_ADDER_continuously.vi"

We have placed the "start" and "stop" VI outside the loop as they only need to be run once upon initialization and completion.

Testing with the hardware:

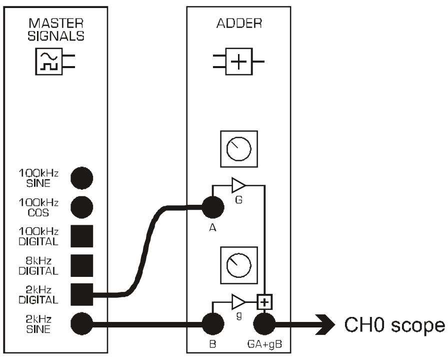


Figure 11: "adjust_the_ADDER" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 11.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the ADDER gains onscreen.

Take note of the VALID SETTINGS ranges.

Programming Tasks:

TASK 1: write a LabVIEW program to vary the A gain linearly, from minimum to maximum in 10 seconds. Keep the B gain at unity.

TASK 2: Add to the LabVIEW program in TASK 1 by writing a LabVIEW program to vary the B gain sinusoidally from minimum to maximum.

AMPLIFIER block:

The AMPLIFIER block on DATEx has one input and one output. This input has variable GAIN. Hence the parameter available for programmatic control is the input GAIN. This is known as "k". Figure 11 shows the AMPLIFIER block as a hardware block and its corresponding low-level VI block.

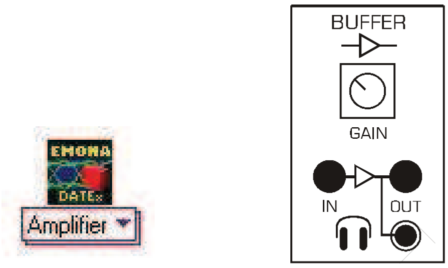


Figure 12: amplifier block, amplifier VI

The full range of GAIN from minimum to maximum for each input is set with an input value in the range of 0 – 127. The data type required is U8: Unsigned Integer.

An example for controlling an AMPLIFIER block from LabVIEW is shown in Figure 12:

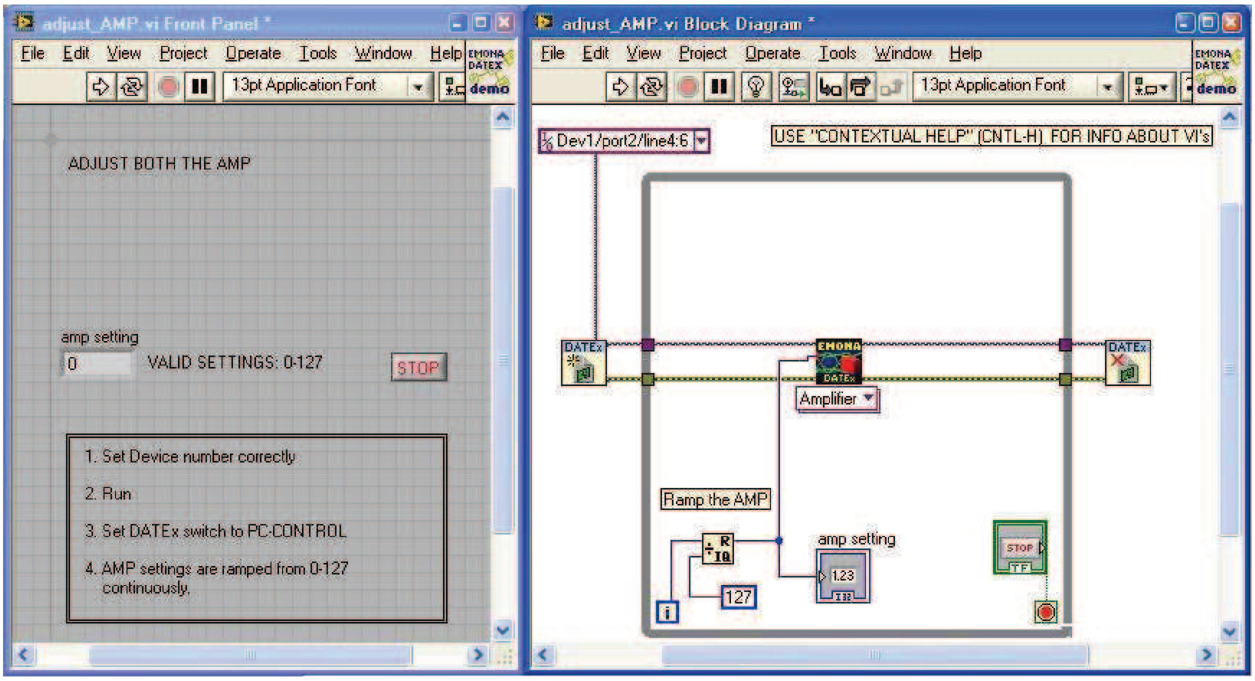


Figure 13: "adjust_AMP.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select AMPLIFIER from its drop down selector)
4) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the AMPLIFIER GAIN is controlled from the while loop iteration counter and is continuously ramped from minimum to maximum. As for all LabVIEW programs this variable can be supplied from constants, controls or any other program value.

We have placed the "start" and "stop" VI outside the loop as they only need to be run once upon initialization and completion.

Testing with the hardware:

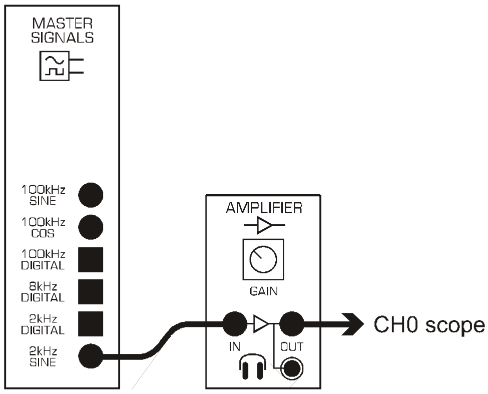


Figure 14: "adjust_the_AMP" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 14.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while observing the AMP gain input value in the indicator control.

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program to implement an automatic gain control feedback loop. Maintain the output in the range of 0.5 to 1.0 V RMS.

TASK 2: Write a LabVIEW program to implement a non-linear compander e.g.: A-law or u-law companding function.

3 - Programming Frequency Control Blocks

In this experiment we will investigate the use of the frequency control blocks available in the DATEx board. The TLPF VI controls the DATEx TUNEABLE LPF functional block's amplitude (GAIN) and cut-off frequency (fc) settings.

TLPF block:

The TUNEABLE LPF functional block on DATEx as one input and output. This input has variable GAIN and a variable corner FREQUENCY adjustment control. The figure shows the TLPF filter module as a hardware block and its corresponding low-level VI Block

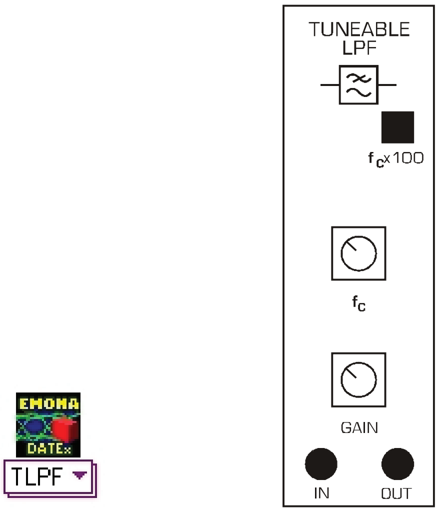


Figure 15 :TLPF VI, TLPF block

The full range of GAIN from minimum to maximum for this input is set with an input value in the range of 0 – 127. The data type required is U8:Unsigned Integer.

The full range of FREQ from minimum to maximum for this input is set with an input value in the range of 0 – 255. The data type required is U8:Unsigned Integer.

An example for controlling a TLPF block from LabVIEW is shown in Figure 14:

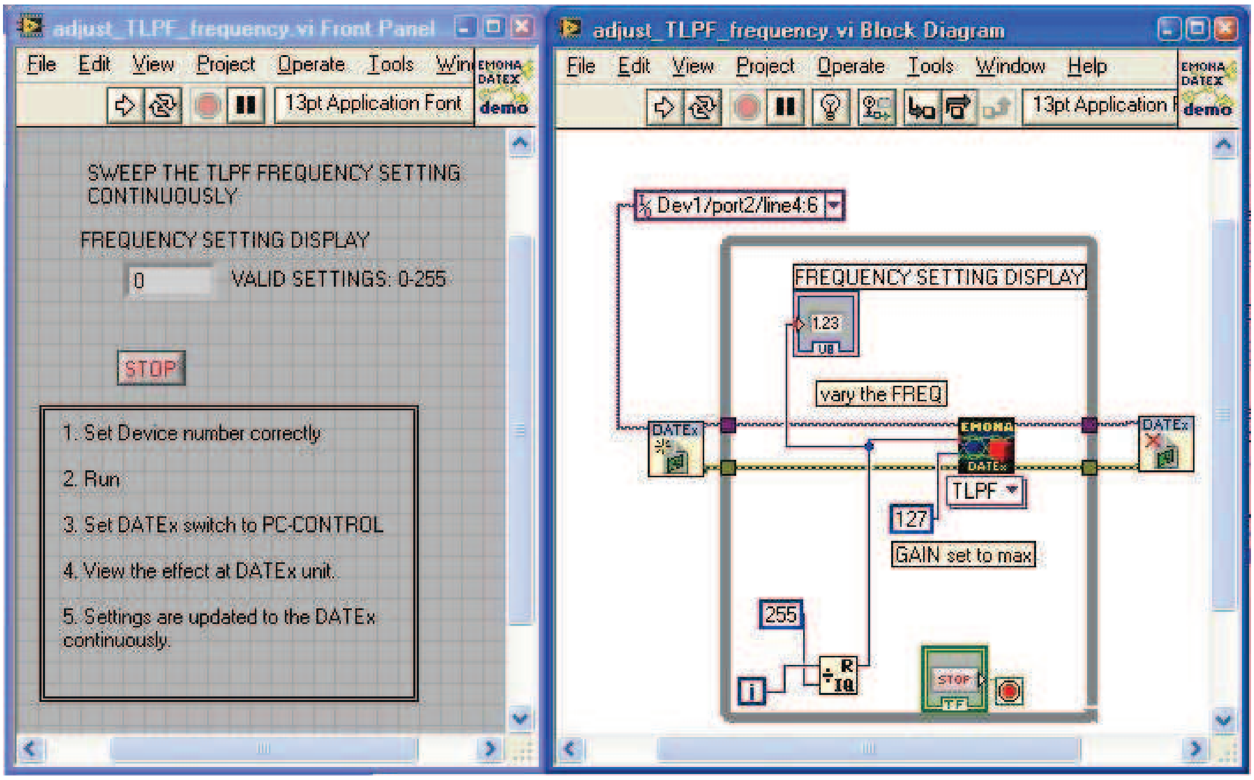


Figure 16: "adjust_TLPF_frequency.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port 2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select TLPF from its drop down selector)
4) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the corner frequency fc is controlled from the while loop iteration counter and is continuously ramped from minimum to maximum. The GAIN is set to a constant maximum value. As for all LabVIEW programs this variable can be supplied from constants, controls or any other program value.

Testing with the hardware:

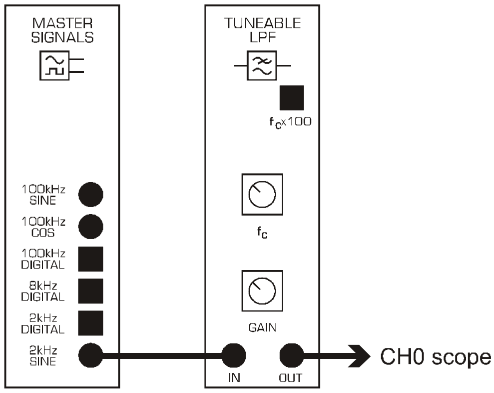


Figure 17: "adjust_the_TLPF" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 17.

2. Select the DATEx board's PC CONTROL mode and RUN this VI.

3. View the output with the scope, whilst noting the FREQ input value in the indicator.

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program to extract the fundamental frequency of any square wave signal.

4 - Programming Phase Control Blocks

In this experiment we will investigate the use of the phase control blocks available in the DATEx board. The block which controls the phase shift of a signal via the control of the PHASE knob and 180 degree switch is the PHASE SHIFTER block.

PHASE SHIFTER block:

The PHASE SHIFTER block on DATEx has one input and output. This input has a continuously variable phase shift control and a constant 180° phase shift switch. Figure 18 shows the PHASE SHIFTER block as a hardware block and its corresponding low-level VI block

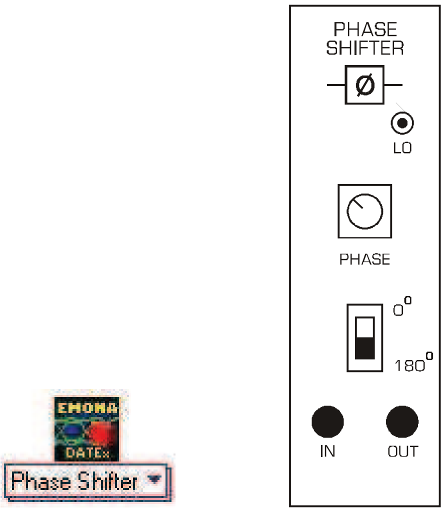


Figure 18: phase shifter VI, phase shifter block

The full range of PHASE SHIFT from minimum to maximum for this input is set with an input value in the range of 0 – 254. The data type required is U8:Unsigned Integer. Note that the maximum is 254, not 255 (for internal circuit realization reasons)

The 180 DEGREE shift input is Boolean. The data type required is a single Boolean.

An example for controlling a PHASE SHIFTER block from LabVIEW is shown in Figure 19.

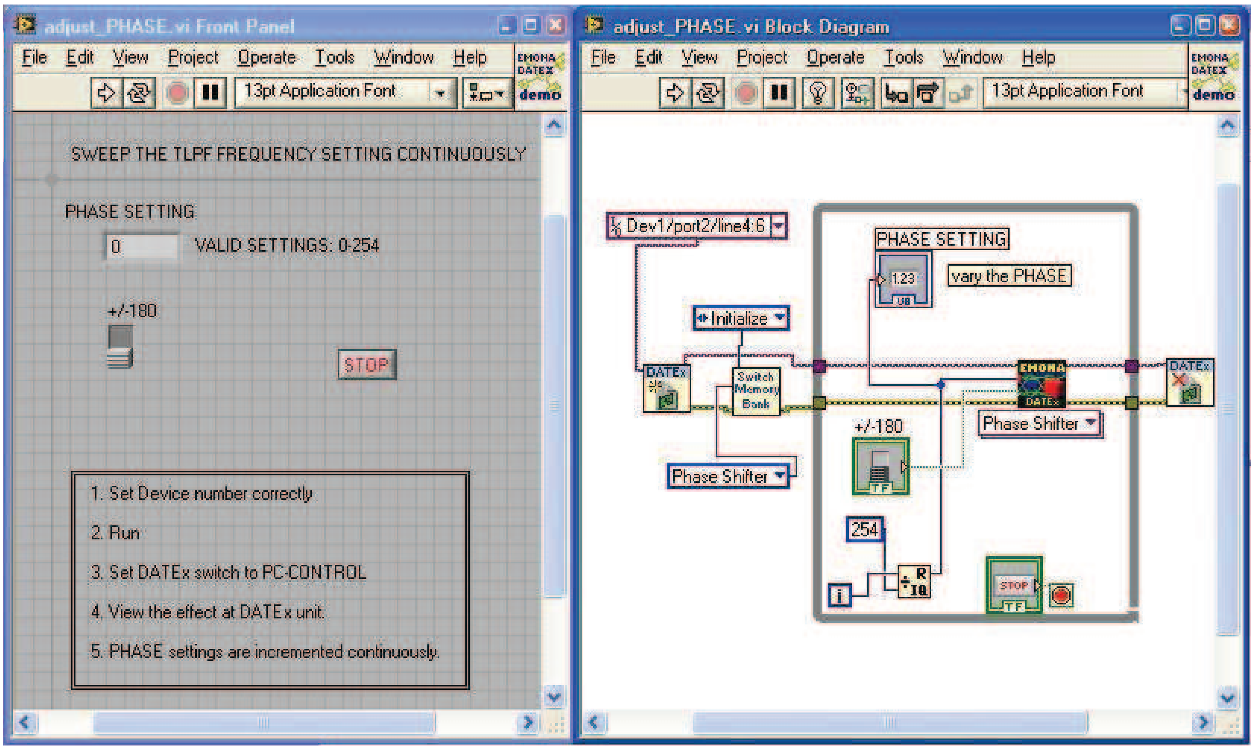


Figure 19: "adjust_PHASE.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port 2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select PHASE SHIFTER from its drop down selector)
4) As well, you must also insert the "Switch Bank" block , from the DATEx functions library (.llb), in order to maintain a record of the switch states. Simply place it as shown in this example. It can be placed outside the loop.
5) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the phase shift is controlled from the iteration counter of the while loop, and so it continuously sweeps the PHASE setting from minimum to maximum. The 180 degree switch is controlled from an onscreen front panel switch. As for all LabVIEW programs these variables can be supplied from constants, controls or any other program value.

Testing with the hardware:

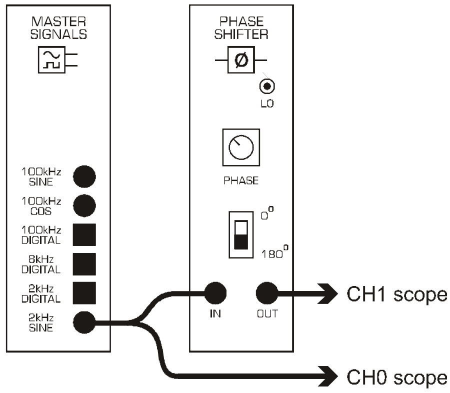


Figure 20: "adjust_PHASE" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 20.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the 180 DEGREE switch occasionally to invert the signal output .

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program which will always provide the quadrature of the input signal.

TASK 2: Write a LabVIEW program which will attempt to equalize the phase shift caused by the TLPF block at any one sinusoidal frequency.

5 - Programming Timing Control Blocks

In this experiment we will investigate the use of the timing control blocks available in the DATEx board. The block which controls position and width of pulses via the WIDTH knob and DELAY knob is the TWIN PULSE GENERATOR (TPG) block.

TWIN PULSE GENERATOR block:

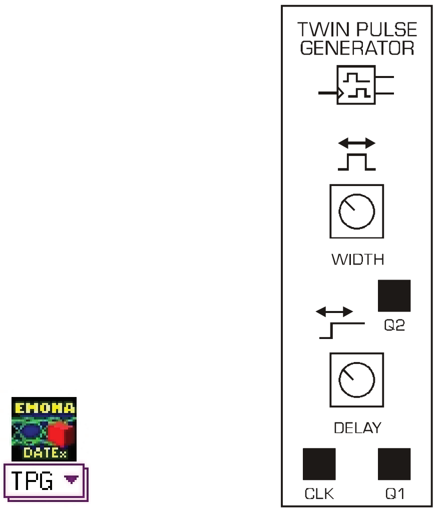


Figure 21: TPG VI, TPG block

The TPG on DATEx has one input and two outputs. The width of both output pulses is controlled by the WIDTH knob. The delay between the two output pulses is controlled by the DELAY knob. Figure 21 shows the TPG block as a hardware block and its corresponding low-level VI Block

The full range of WIDTH and DELAY from minimum to maximum for each input is set with an input value in the range of 0 – 255. The data type required is U8:Unsigned Integer.

An example for controlling a TPG block from LabVIEW is shown in Figure 22.

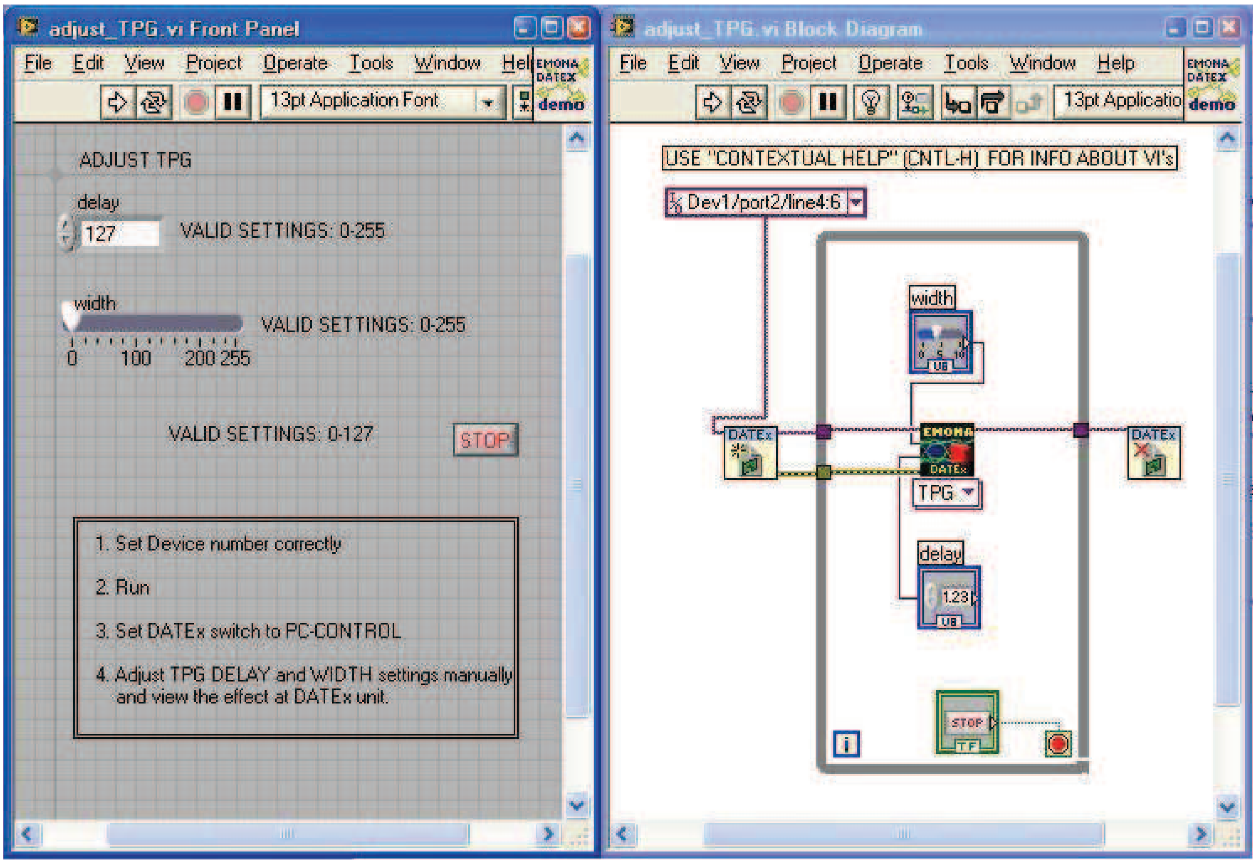


Figure 22: "adjust_TPG.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port 2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select TPG from its drop down selector)
4) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the WIDTH is controlled from the front panel slider. The DELAY is also controlled from a front panel numeric input control. As for all LabVIEW programs these variables can be supplied from constants, controls or any other program value.

Testing with the hardware:

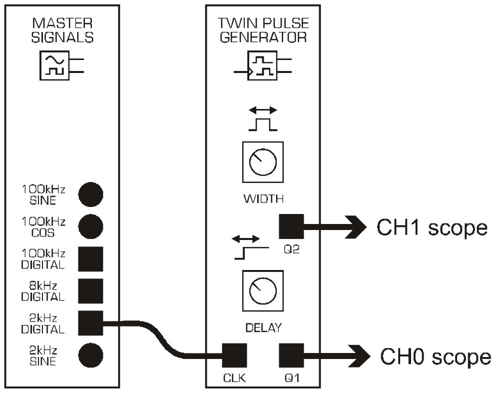


Figure 23: "adjust_TPG" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 23.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the WIDTH and DELAY settings onscreen.

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program which will output a square wave with a delay of 45 degrees and maintain an even duty cycle, whilst the input clock's frequency varies slightly.

6 - Programming Mode Control Blocks

In this experiment we will investigate the use of the mode control blocks available in the DATEx board. The blocks which control the mode switches of particular circuit functions are the PCM/TDM block and the SEQUENCE GENERATOR block.

PCM/TDM block:

The PCM/TDM on DATEx has several inputs and several outputs. Whether the PCM ENCODER block is implementing a single input PCM, or dual input TDM function, is controlled by the TDM switch. Figure 24 shows the PCM/TDM block as a hardware block and its corresponding low-level VI block.

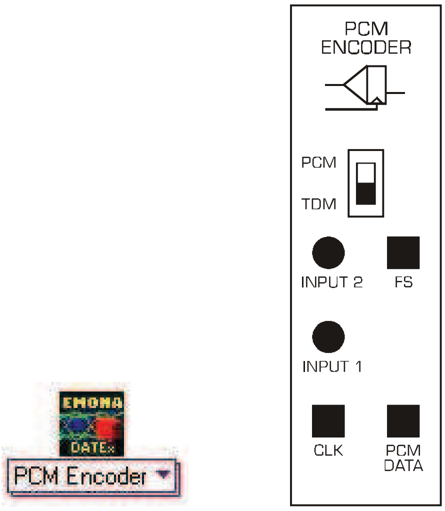


Figure 24: PCM/TDM VI, PCM/TDM block

The TDM switch is a Boolean function. The data type required is single Boolean.

An example for controlling PCM/TDM block from LabVIEW is shown in Figure 25:

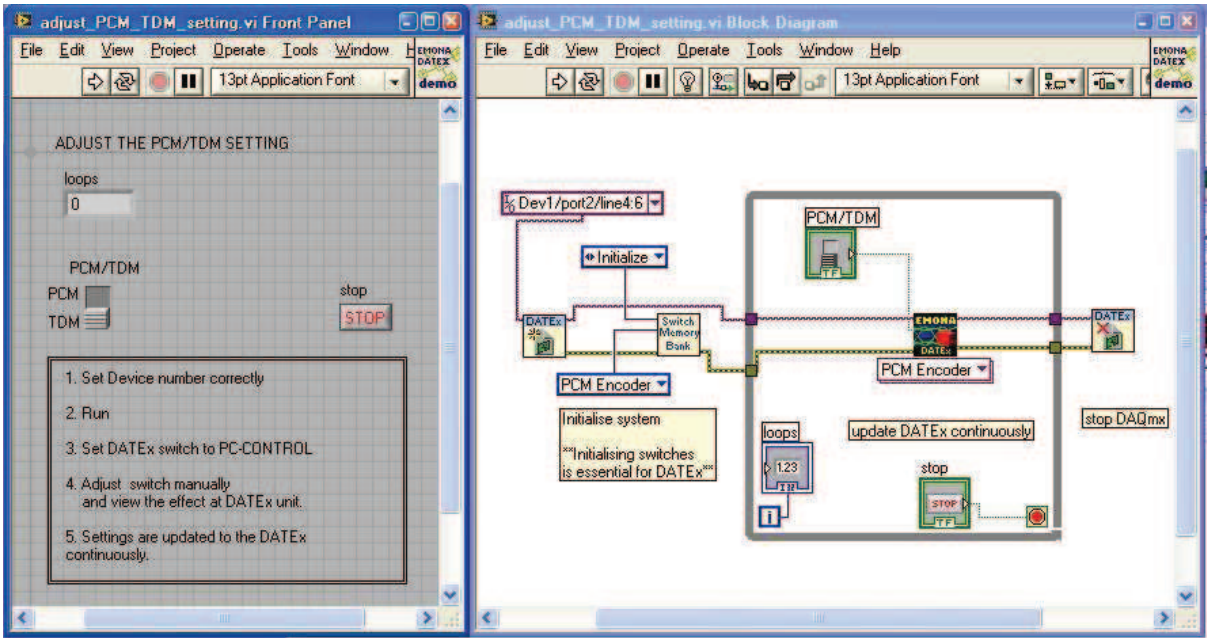


Figure 25: "adjust_PCM_TDM_setting.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select PCM/TDM from its drop down selector)
4) As well, you must also insert the "Switch Bank" block , from the DATEx functions library (.llb), in order to maintain a record of the switch states.
5) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the PCM/TDM mode is controlled from the on-screen switch. As for all LabVIEW programs these variables can be supplied from constants, controls or any other program value.

Changing the mode of this block programmatically can be used to change the sampling rate of a single channel, or to introduce the sampling of a second channel at the output for multiplexing purposes.

Testing with the hardware:

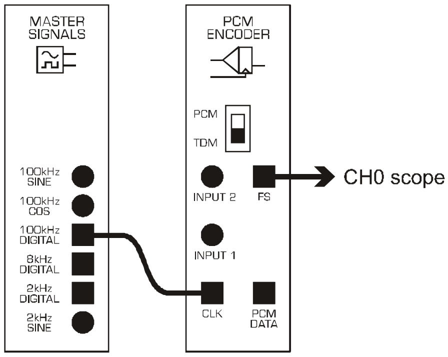


Figure 26: "adjust_the_PCM_TDM" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 26.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the TDM select switch onscreen. You should see the TDM mode led change.

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program that switches from PCM to TDM mode only when a signal is presented at INPUT 2.

TASK 2: Write a LabVIEW program to mimic a PCM ENCODER and output the CLK, PCM DATA & FS signals via the NI ELVIS DIGITAL OUTPUT lines and recover the analog signal with the PCM DECODER block. Your bit rate will most likely be very low, and your analog voltage can be slowly varying DC.

SEQUENCE GENERATOR/LINE CODE block:

The SEQUENCE GENERATOR block on DATEx has one input and several outputs. The output which is controlled by the DIP SWITCH on the board is the LINE CODE output. Four different line codes can be selected from the dip switch. The figure shows the SEQUENCE GENERATOR block as a hardware block and its corresponding low-level VI block.

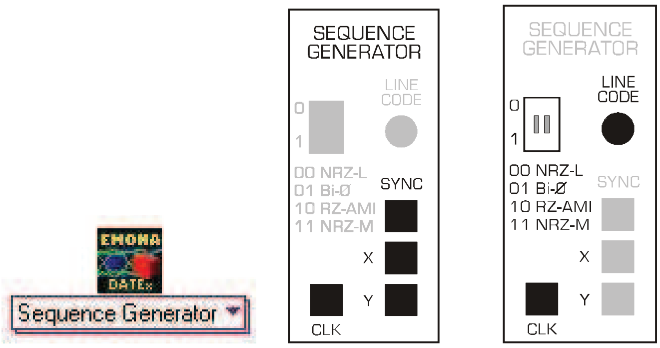


Figure 27: SEQUENCE GENERATOR VI, SEQUENCE GENERATOR block, LINE CODE block

The LINE CODE DIP switch is a dual Boolean function. The data type required is two independent Boolean variables.

An example for controlling the SEQUENCE GENERATOR block from LabVIEW is shown in Figure 28:

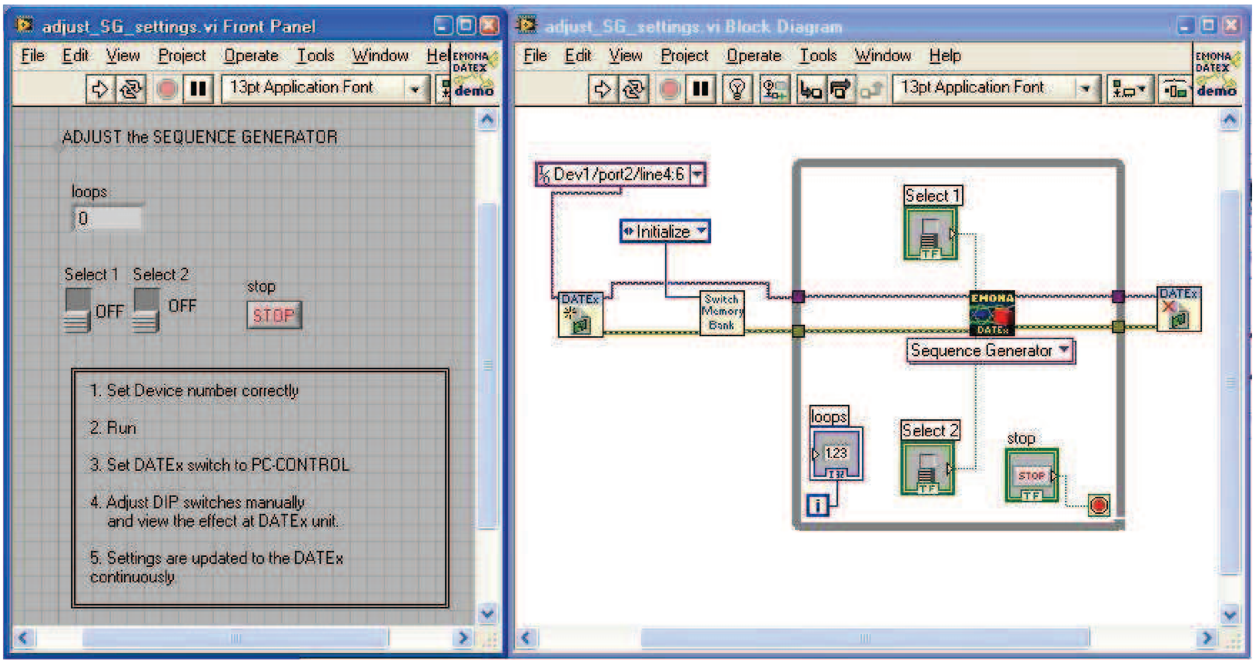


Figure 28: "adjust_SG_settings.vi"

Programming procedure:

1) The DAQmx task must be inserted and set to suit the Device Number of the NI ELVIS. As well the lines selected MUST BE lines 4:6 of port2
2) Insert and wire a DAQmx Start task
3) The DATEx VIs can then be inserted and controlled by the program. Insert the DATEx polymorphic VI (browse to its location and select), then select Sequence Generator from its drop down selector)
4) As well, you must also insert the "Switch Bank" block , from the DATEx functions library (.llb), in order to maintain a record of the switch states. Simply place it as shown in this example.
5) Insert a 'Stop the DAQmx interface' VI upon exit.

In this example the LINE CODE selected is controlled from the dual on-screen switches, mimicking a dual DIP switch. As for all LabVIEW programs these variables can be supplied from constants, controls or any other program value. Changing the mode of this block programmatically can be used to change the line coding of the signal output for various purposes during an experiment.

Testing with the hardware:

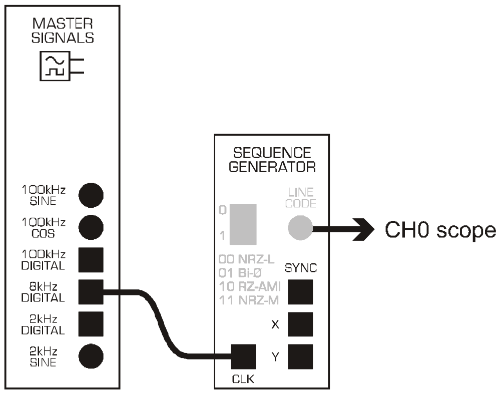


Figure 29: "adjust_SG_settings" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 29.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the LINE CODE selection onscreen.

Take note of the VALID SETTINGS ranges.

Programming tasks

TASK 1: Write a LabVIEW program to vary the TLPF corner frequency so as to attenuate all the harmonics of the line coded output signal except for the main lobe of the spectrum. Vary the LINE CODE selection switch and have the TLPF adapt accordingly.

7 - Sequencing and Combining the DATEx Blocks

Several DATEx blocks can be used in the same program. In this way several DATEx hardware functions can be controlled programmatically from a program within the same experiment. As all DATEx blocks share the same communications bus, the commands to each DATEx block must be sent sequentially. Communications to the DATEx blocks is rapid. Typically, several commands per second may be sent and implemented by the DATEx blocks.

"Sequence" type structures may be useful for maintaining this requirement. In this example we see the commanding of several DATEx blocks one after the other, in a continuous loop. There is no particular order for commanding each DATEx block. The DATEx hardware will simply implement each command when issued.

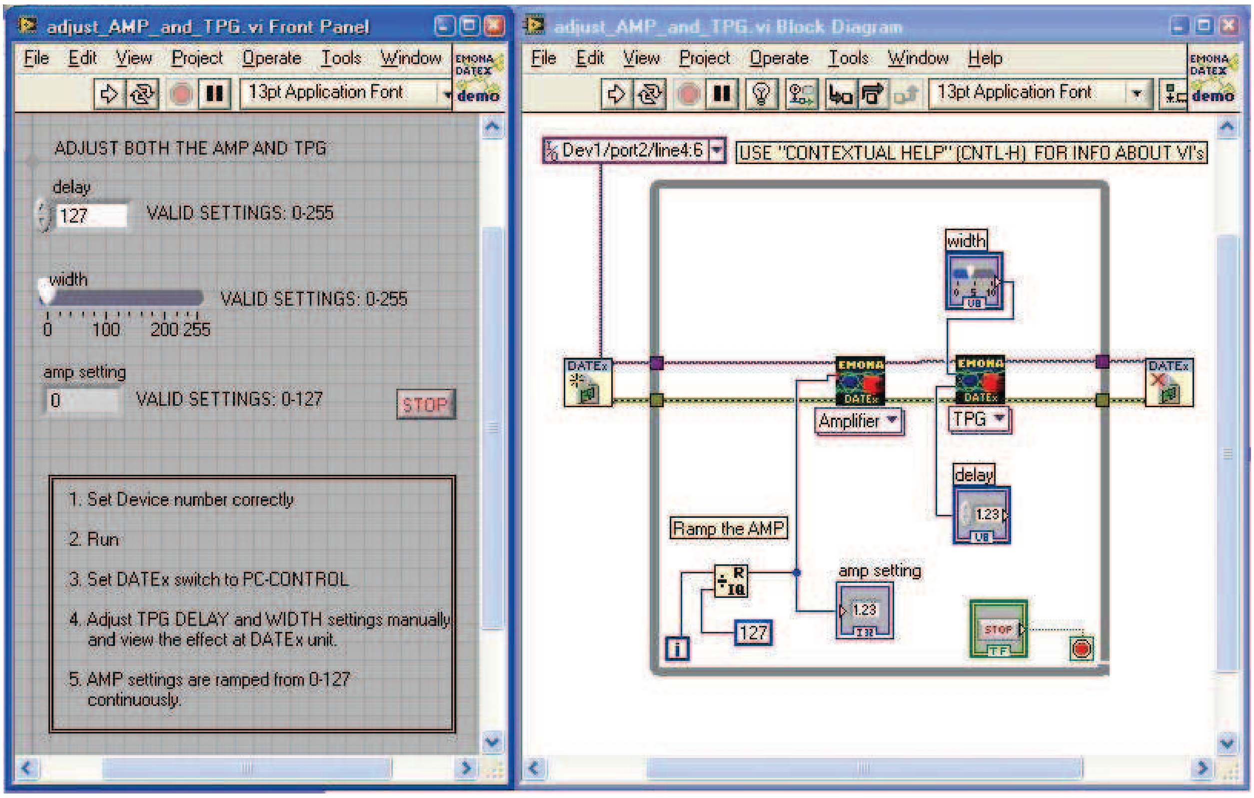


Figure 30: "adjust_AMP_and_TPG.vi"

In this simple example, for each iteration of the loop, one command is sent to the AMP, then one command to the WIDTH block, and one command to the DELAY block. So three commands are sent per loop iteration. You can view the rate of command communications by noting the rate of change of the "amp setting", which changes once per loop.

Testing with the hardware:

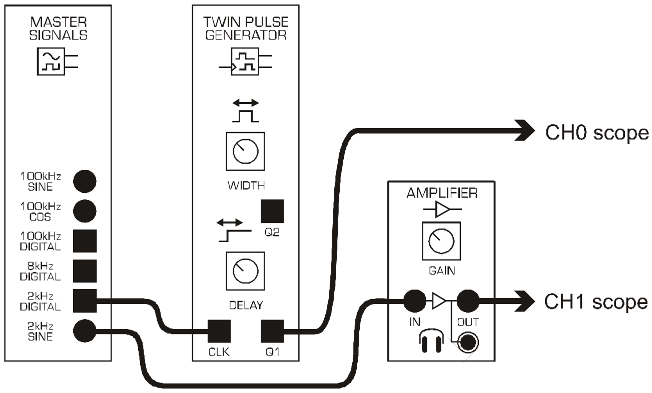


Figure 31: "adjust_AMP_and_TPG.vi" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 31.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the ADDER gains onscreen.

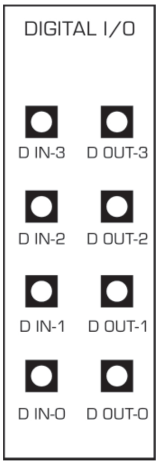
Take note of the VALID SETTINGS ranges.

8 - Using NI ELVIS Instruments on the DATEx

There are a number of NI ELVIS instruments available directly from the DATEx board. These instruments are located on the left-hand side of the board. How to access these instruments is discussed as follows:

DIGITAL I/O:

Four digital inputs and four digital output lines are available on the DATEx board. These are controlled from the DIGITAL READER and DIGITAL WRITER Express VI as follows:



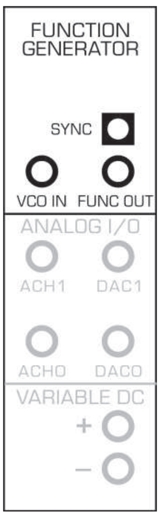
D-IN-3 : DIO 3
D-IN-2 : DIO 2
D-IN-1 : DIO 1
D-IN-0 : DIO 0

D-OUT-3: DIO 11
D-OUT-2: DIO 10
D-OUT-1: DIO 9
D-OUT-0: DIO 8

The digital output lines can be used to provide control signals or digital data sequences to the experiments. The digital input lines can be used to read the digital signals from the experiment.

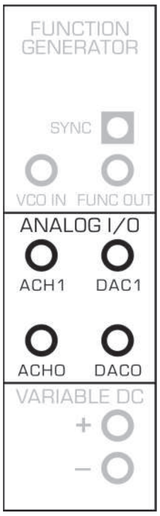
FUNCTION GENERATOR:

The function generator built into NI ELVIS is available at these terminals. This block can also be programmatically controlled from LabVIEW via its Express VI. It is a wide-ranging oscillator which can be used for a number of purposes.



ANALOG I/O:

Two analogue inputs and two analog outputs are available at the DATEx board. These are connected to ELVIS as follows:



ACH1 : Dev?/ai1+ (ai1- connected to GND)
ACH0 : Dev?/ai0+ (ai0- connected to GND)

DAC1 : Dev?/ao1
DAC0: Dev?/ao0

[Dev? represents the Device Number of your NI ELVIS unit.]

These can be programmatically controlled via the ELVISmx blocks on the LabVIEW MEASUREMENT I/O palette. These analog outputs are normally driven by the arbitrary wave form generator instrument supplied with the NI ELVIS. You can also use these outputs to create your own custom instrument such as the VAR_DC.vi supplied the DATEx CD and shown in Figure 32 below. This instrument was created using an analog output to provide a continuously variable DC voltage within a specific range, -2.5 to +2.5 V controlled by an onscreen slider. This custom instrument is useful in AM experiments where a bipolar DC voltage is needed.

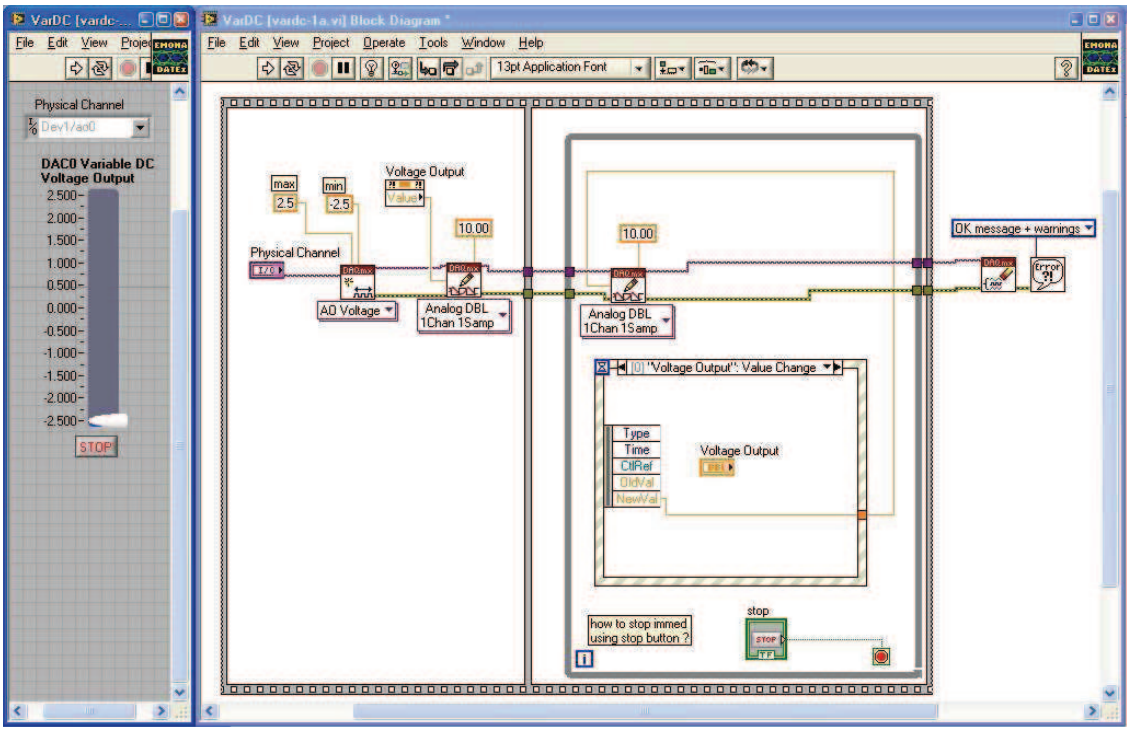
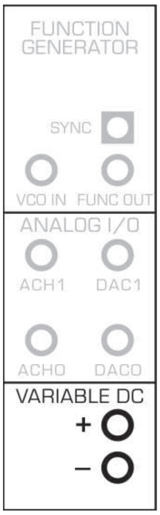


Figure 32: "VarDC.vi"

VARIABLE DC:

The variable power supplies are available on the DATEx board. These can also be programmatically controlled via their Express VI. These are generally used by DATEx as variable DC signals, not as power supplies.

Note that the "VarDC.vi" instrument described in the previous section can often replace the use of these in DATEx experiments.



9 - Building LabVIEW Controlled DATEx Experiments

Example: Automatic nulling using the PHASE SHIFTER

The correct and precise alignment of the phase of carriers is an essential and important task in telecommunications. In order to guarantee the maximum recovered signal, the phase of the received signal and the local carrier must be as closely aligned as possible. In this example we create a "scanning loop" in order to determine the maximum recovery of the received signal.

The final part of this program sets the phasing of the carriers to the determined minima point.

There is a certain input value which when input to the phase shifter VI, will create a null, or minima, in the output. This program aims to determine what that input value is. A faster more real world implementation of this system would aim to keep the phases aligned continuously.

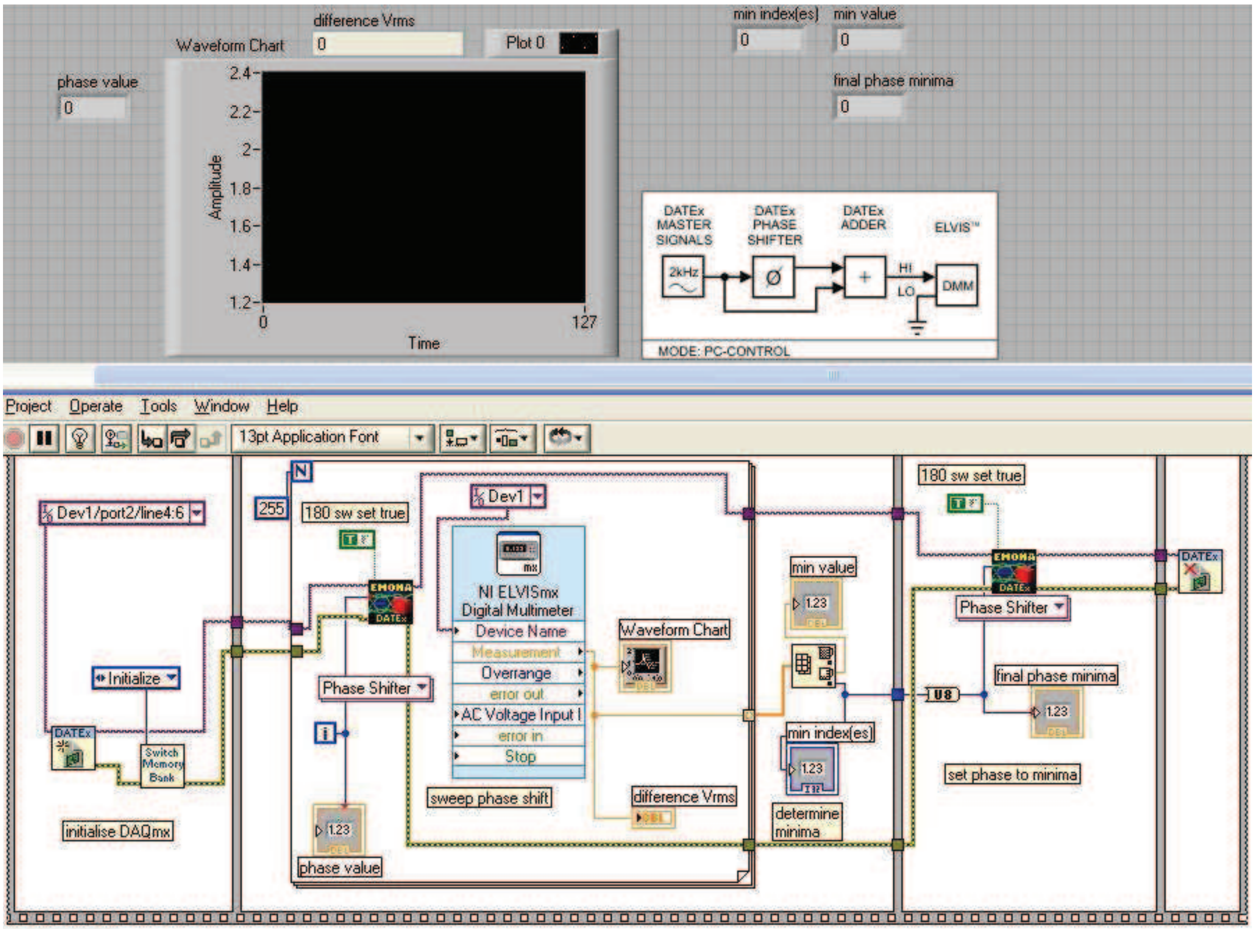


Figure 33: Screenshot of "auto null using PS" VI supplied on DATEx CD in "low level examples" directory.

Frame 1: Initialize the DAQmx interface to DATEx, create the task, setup to the correct device number, and "initialize" the DATEx "switch bank" VI.

Frame 2: Ramp the input values to the PHASE SHIFTER from minimum to maximum while taking readings using the NI ELVIS multimeter, to determine the output voltage. Store those voltage readings in an array and search for the minimum value in that array.

Frame 3: Pass this minimum value to the PHASE SHIFTER to set the circuit to null voltage.

Frame 4: Halt the program and stop the DAQmx interface. The PHASE SHIFTER remains at the last set value.

Patch together the experiment as shown in the wiring diagram on the front panel.

The 180° switch input to the phase shifter VI is set to a constant Boolean value. It is not used in this example.

The maximum rate at which this loop will execute is determined by the time taken for the ELVIS DMM to return a reading. This program is designed as a teaching example rather than an "optimized" real world implementation.

Testing with the hardware:

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 33.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output with the scope while varying the ADDER gains onscreen.

Take note of the VALID SETTINGS ranges.

Programming tasks

Coherent telecommunications schemes require correct phase alignment of the carrier. For example, coherent product demodulation of AM and DSB and demodulation of BPSK are covered in Volumes 1 and 2 of the DATEx Lab Manual.

TASK 1: Modify this program to measure and optimize the output amplitude for some of these coherent schemes.

TASK 2: Vary this program to create a characterization table for the PHASE SHIFTER block. By measuring and storing the relationship between the phase shift index and the actual phase shift of the PHASE SHIFTER circuit block you can then apply this data in your coherent demodulation experiments.

Example: Viewing filter responses using FFTs

In this experiment we will utilize the DATEx DUAL ANALOG SWITCH in a novel way as an 2-input analog multiplexer. In this way we can programmatically switch between two input signals while viewing a single output. We will use the NOISE GENERATOR module as a broadband signal source for the spectrum display to visualize the filter response of the TUNEABLE LOW PASS module.

The circuit has 2 inputs:
1) D OUT-0 controls the switching of the "signal" to be filtered.
2)D OUT-1 controls the switching of the "noise" used to outline filter response.

As the noise has many more components than can be individually displayed, the spectrum of the "noise" signal displayed becomes the envelope of the frequency response of the filter. This serves to display the entire frequency response of the filter at any particular point in time.

We use a front panel toggle to switch between the 2 possible input signals to the filter via the DAS module. The control signals are output from the DATEx DIGITAL I/O terminals.

The TUNEABLE LOW PASS FILTER can be tuned from the on-screen slider control. The "corner frequency" parameter is transmitted to the DATEx board once per while loop, and the filter is tuned accordingly.

The output from the TLPF module is sampled as an Analog Voltage by the DAQmx, which connects to the signal via the ACH0 terminal on DATEx.

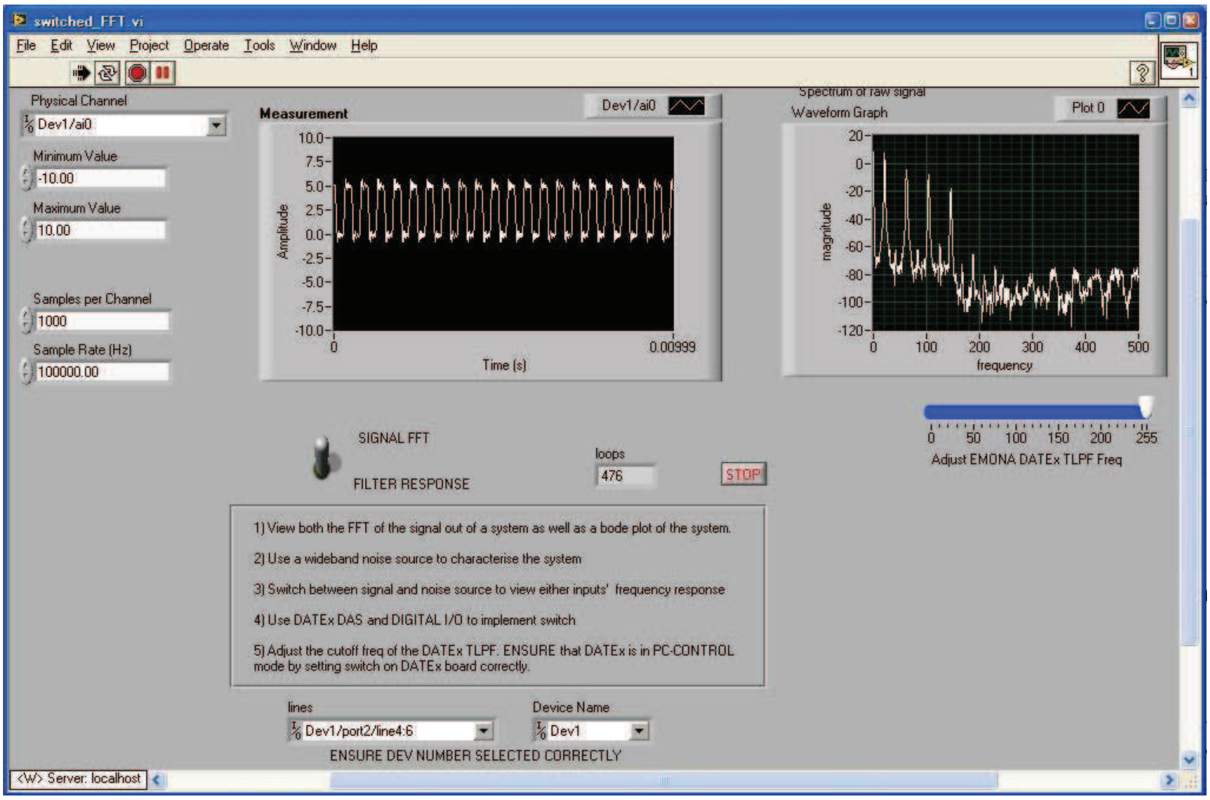


Figure 34: "switched FFT.vi" front panel

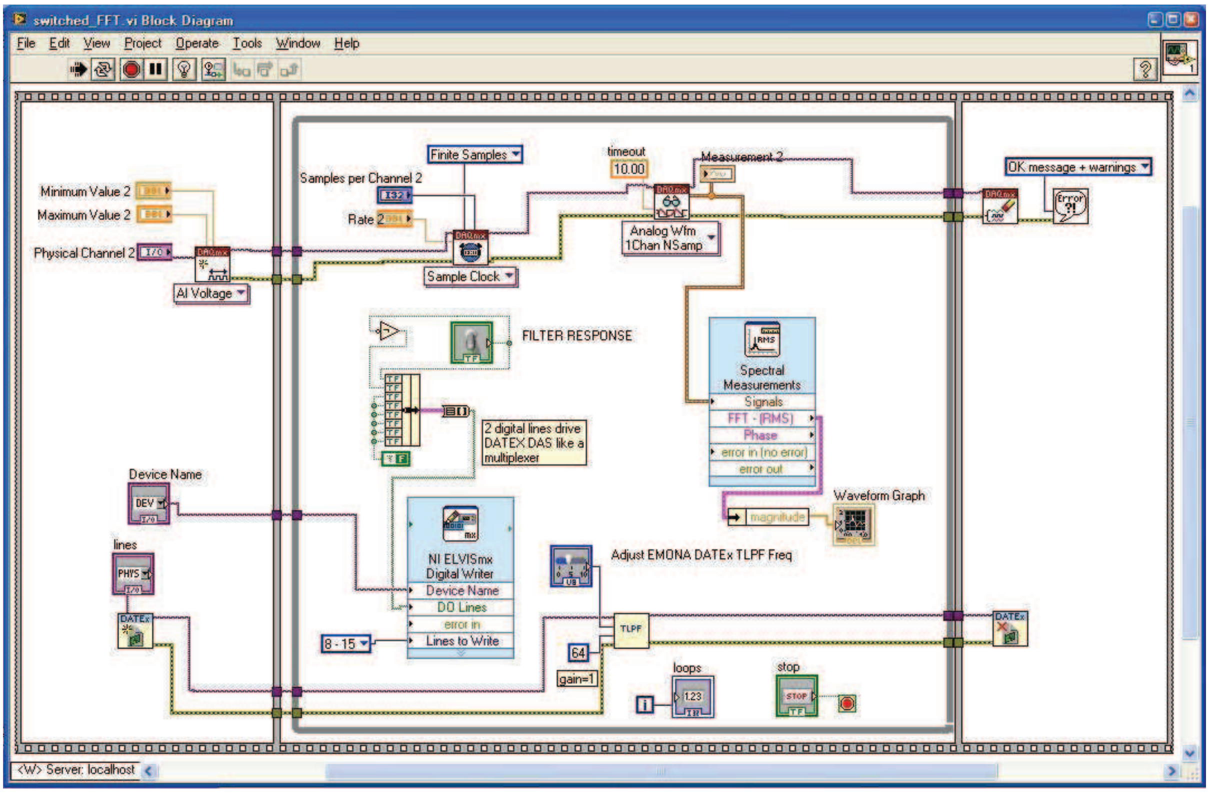


Figure 35: "switched FFT.vi" code

Frame 1: Initialize the DAQmx interface to DATEx, create the task, setup to the correct device number, and "initialize" the DATEx "switch bank" VI, as well as setup the Analog input for sampling.

Frame 2: Vary the corner frequency parameter to the TLPF, implement the multiplexer switching and sample the signal for the FFT display.

Frame 3: Halt the program and stop both DAQmx interfaces.

Wire up the experiment as shown on the VI front panel.

Testing with the hardware:

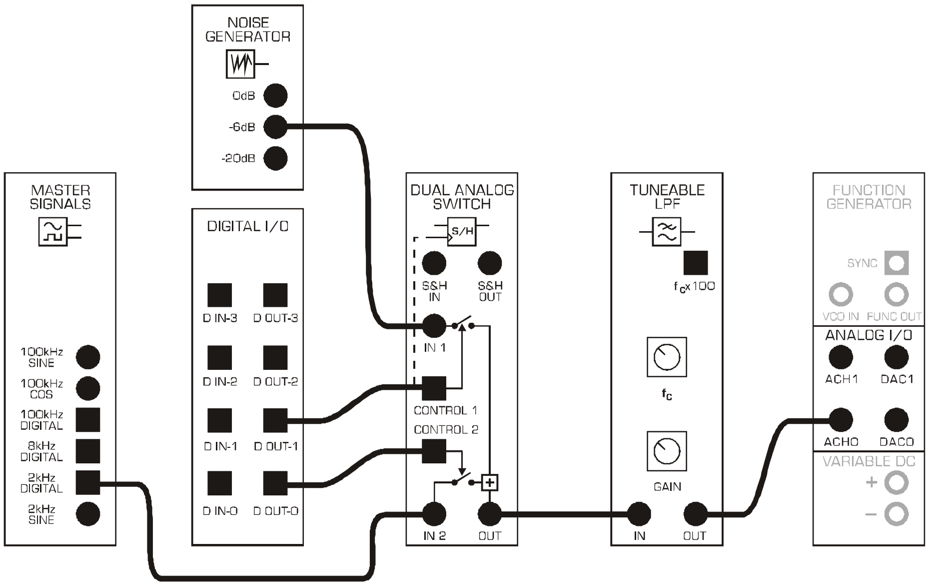


Figure 36: "switched FFT.vi" wiring diagram

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 36.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output while varying the TLPF corner frequency.
4. Switch between the signal and the filter response while varying the corner frequency.

This practical experimenting with the filter aims to reinforce understanding of the filter response in relation to the resulting signal time and frequency domain signals.

As well this experiment shows how DATEx blocks can be used to implement various circuit functions.

Programming tasks

TASK 1: Explore the frequency domain with this circuit by inputting different signals. Some suggested signals which will be interesting are:

2kHz sine (from MASTER SIGNALS)
Speech (from SPEECH module)
2kHz NRZ-L (from SEQUENCE GENERATOR: LINE CODE output)

TASK 2:Experiment with changing the sampling rate of the ANALOG INPUT to see its effect on the captured signal.

Example: Analyzing noise circuit performance

In this experiment we will utilize the analysis capabilities of LabVIEW to compare the performance of the real circuit implementation of the DATEx NOISE GENERATOR to a simulated Gaussian noise source.

The DATEx noise signal is sampled using Ch0 of the NI ELVIS SCOPE.

The simulated noise is created using an Express VI. As well, the simulated noise is filtered at 240 kHz to model the real circuit bandlimiting of the DATEx noise module.

The time and frequency domains of both signals are displayed, as is the distribution histogram.

The aim of this experiment is to gain familiarity with the analysis of real world electrical signals using LabVIEW.

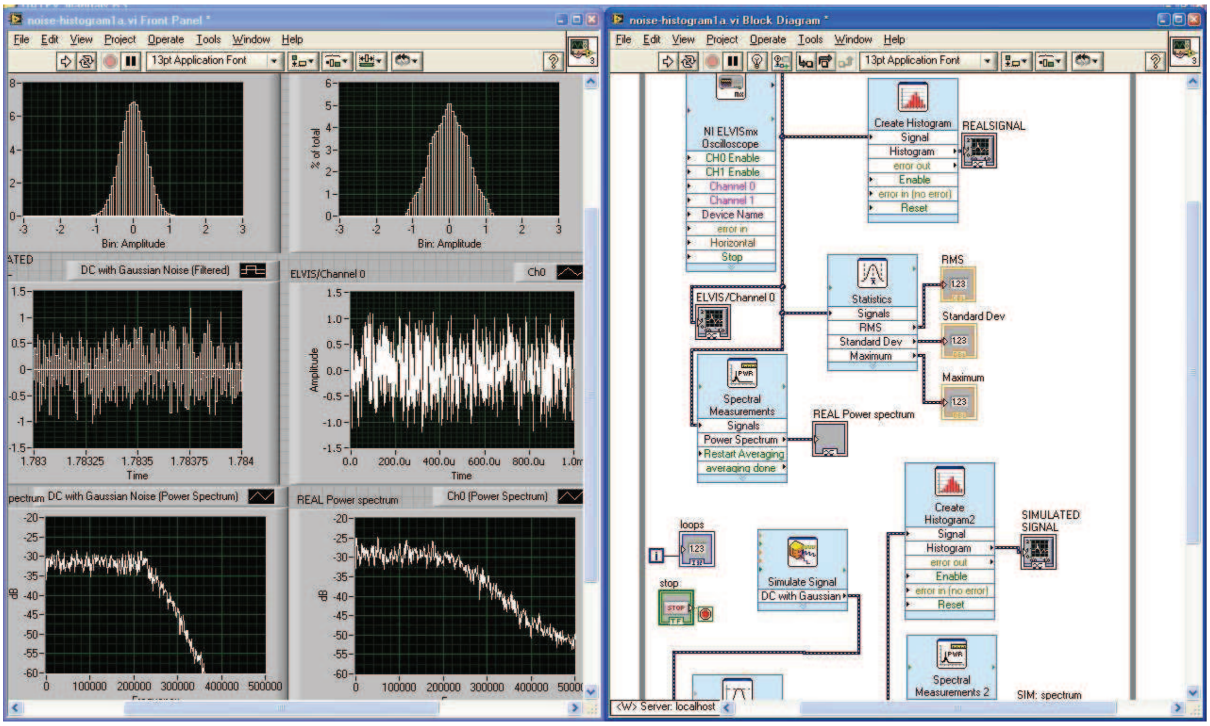


Figure 37: "noise-histogram.vi" front panel and code

The top part of the code implements the capturing and analysis of the real noise signal.

The bottom part implements the creation and analysis of the simulated noise signal.

Testing with the hardware:

1. Connect Ch0 of the scope to the -20dB output of the NOISE GENERATOR. Connect the black scope lead to a GND terminal.

Programming tasks

TASK 1: Can you explain any differences between the real signals and the simulated ones.? Can you vary the program to highlight these differences.?

TASK 2: Can you improve the program to give more insights into the signals being investigated ?

Example: Automatic gain control

In this experiment we will introduce the use of DATEx for gain control in circuits. We will start by programmatically controlling an AMPLIFIER block. This AMPLIFIER block is used to create a varying input signal which is then measured by the LabVIEW program and adjusted to have a constant output amplitude. In this example we control the input gain using software controlled hardware, and the output gain purely with software. A second stage of this experiment introduces the use of the Analog output ports ACH0 and a hardware MULTIPLIER block to implement output gain control.

The input signal, in this case a 2kHz SINE, is passed through an AMPLIFIER block, whose gain is controlled from the LabVIEW code. The varying signal is measured by LabVIEW and a scale factor is calculated and applied to the output display signal. The range of the AMPLIFIER gain is limited to 40 so as to not saturate the amplifier's output signal.

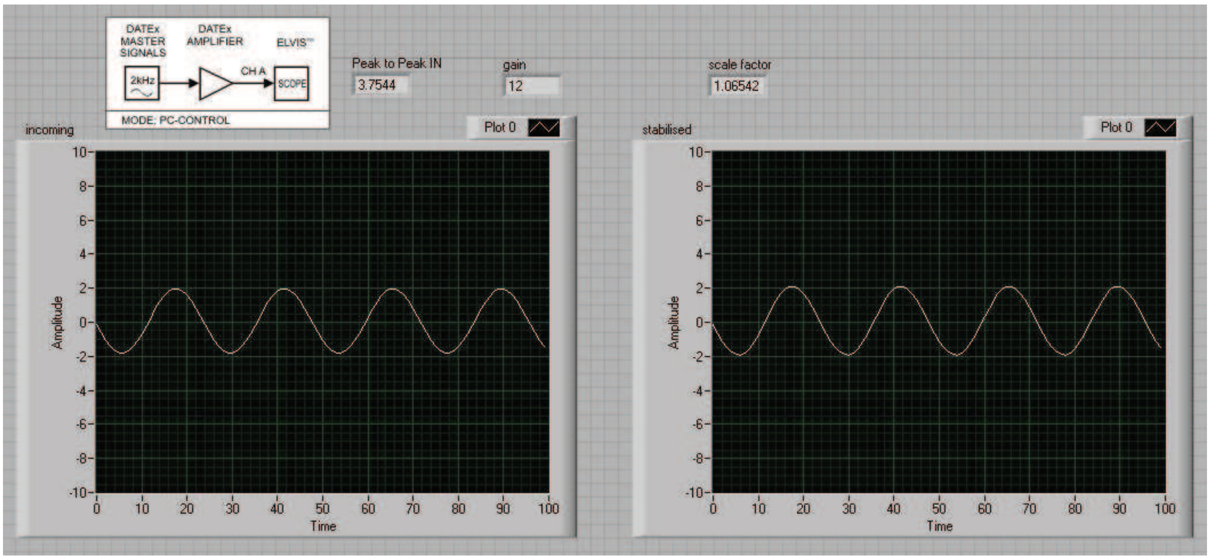


Figure 38: "gain control with amp.vi" front panel

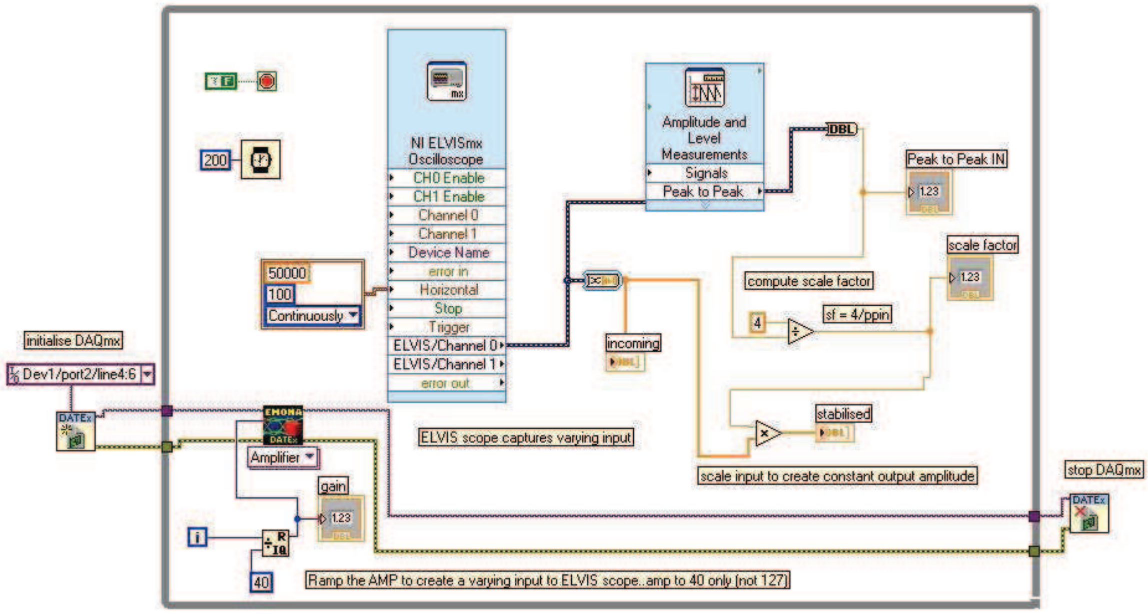


Figure 39: "gain control with amp.vi" code

As a first step we initialize the DATEx block.

Inside the loop we vary the amplifier gain and take measurements with the ELVIS Scope. The scale factor is computed and applied to the output display to create a constant output amplitude.

Wire up the experiment as shown on the vi front panel.

Testing with the hardware:

1. Use patching cords to wire up DATEx blocks according to the wiring diagram in Figure 38.
2. Select the DATEx board's PC CONTROL mode and RUN this VI.
3. View the output and notice the scaling factor value.

Programming tasks

TASK 1: Modify a copy of this program in order to further investigate automatic gain control. Change the scale factor control to a graph indicator so as to see the relationship of the scale factor to the input amplitude.

TASK 2: Write a LabVIEW program which introduces a hardware MULTIPLIER gain block into the circuit. The LabVIEW multiplier block used to implement the scale factor can be replaced with a hardware multiplier circuit. The hardware MULTIPLIER has two inputs. If one input is connected to the varying "scale factor" as a voltage, the output of the MULTIPLIER will be scaled accordingly. The scale factor voltage can be output from the ELVIS ANALOG I/O on the DATEx, DAC0 or DAC1.

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.

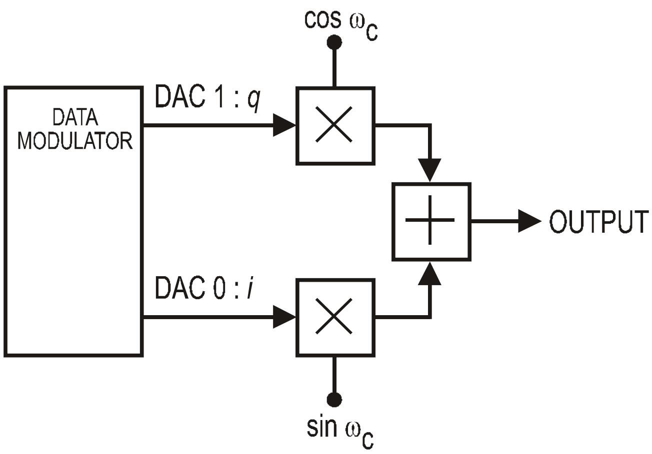


Figure 40: I/Q quadrature modulation

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.

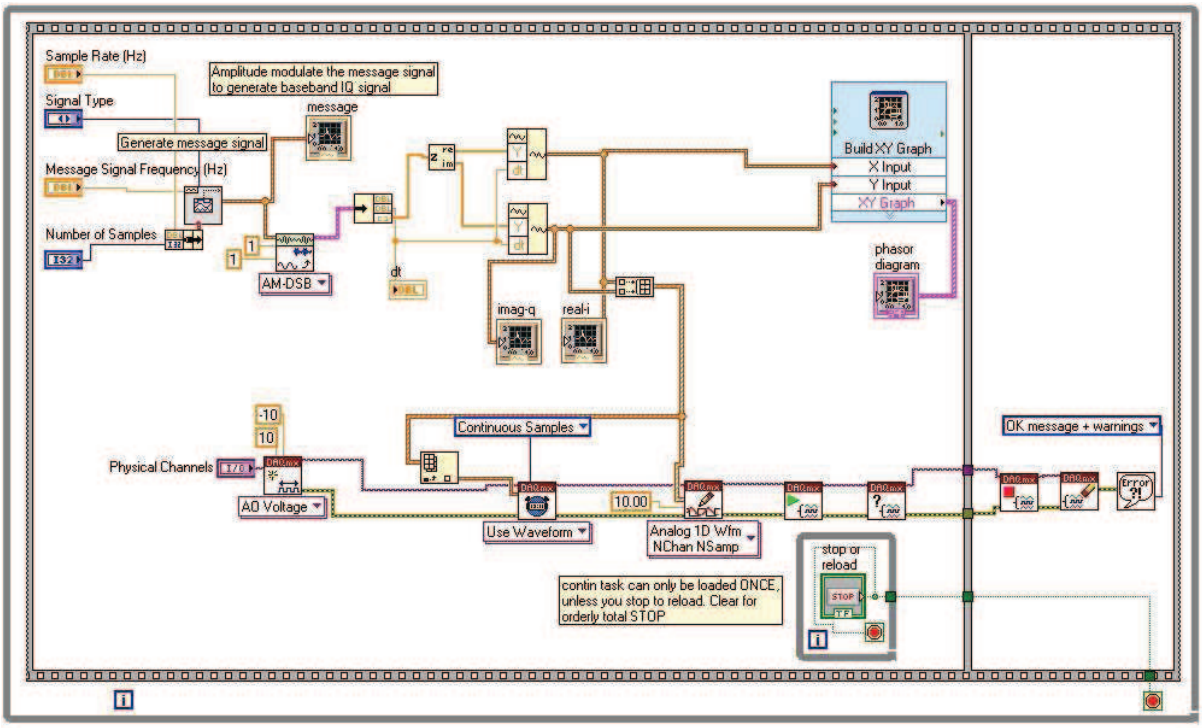


Figure 41: LabVIEW block diagram for AM modulation "am-iq-to-dac.vi"

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.

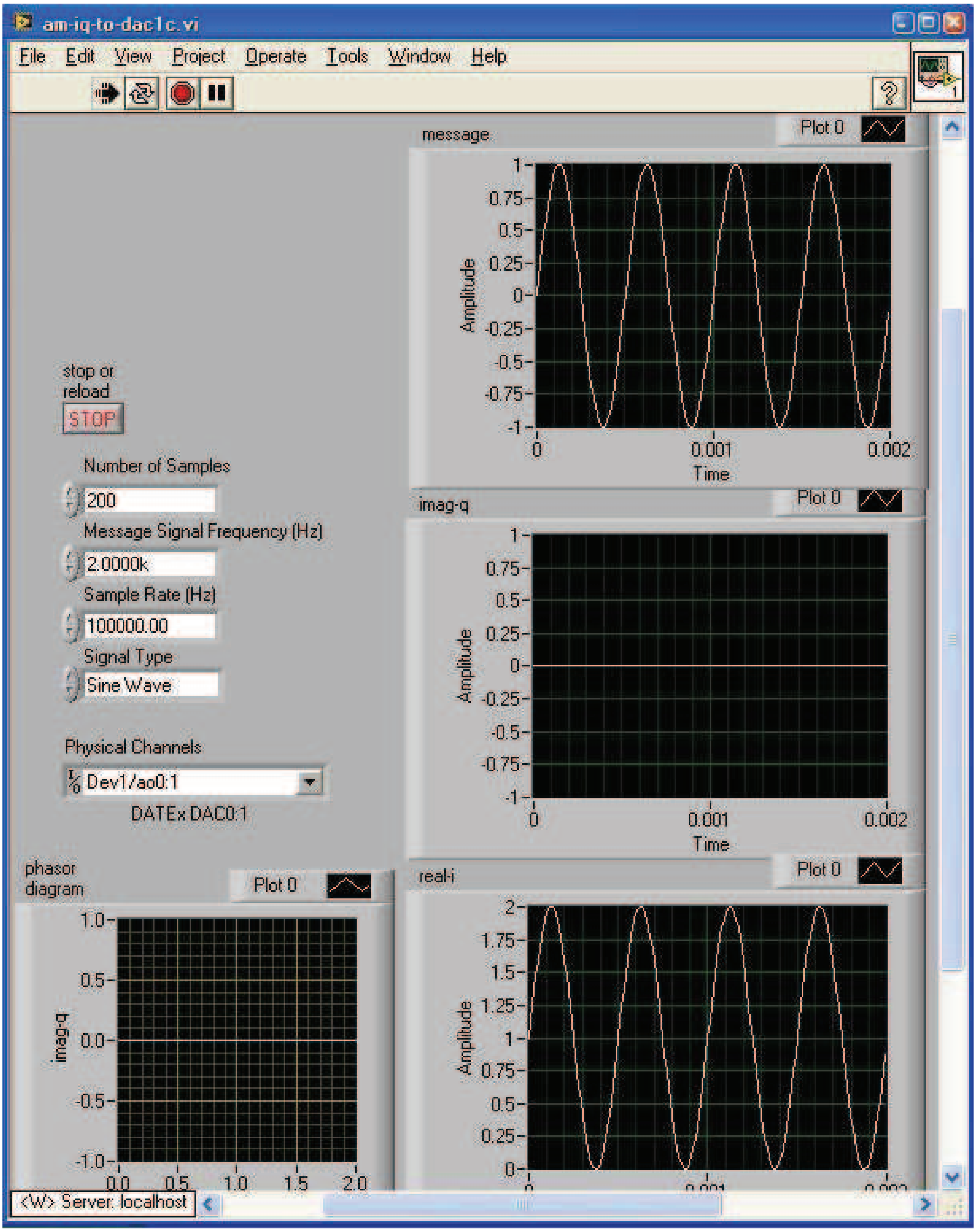


Figure 42: Front panel of the AM experiment

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.

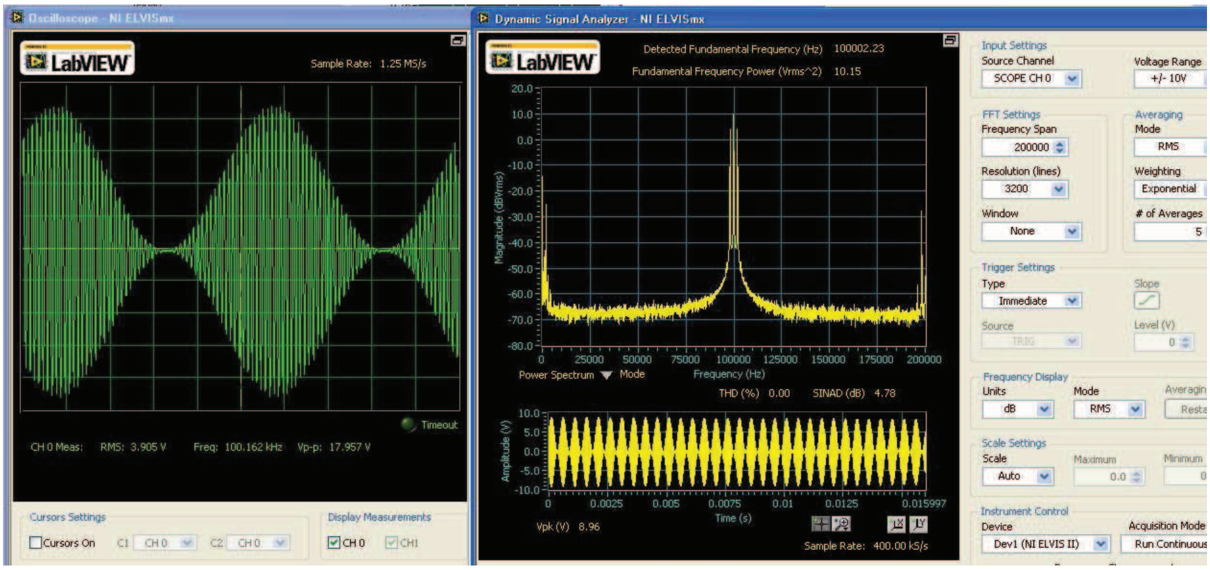


Figure 43: Screenshots from the ELVIS scope and spectrum analyzer

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.

Example: Armstrong's Phase Modulator using the LV Modulation Toolkit

In this experiment we will generate Phase Modulation (PM) using Armstrong's method and a baseband quadrature modulation structure. As discussed in previous experiments in Volume 2 of the Lab manual , Armstrong's method creates narrowband phase modulation. We will use the same modulation structure as for the previous experiments, shown below in Figure 1.

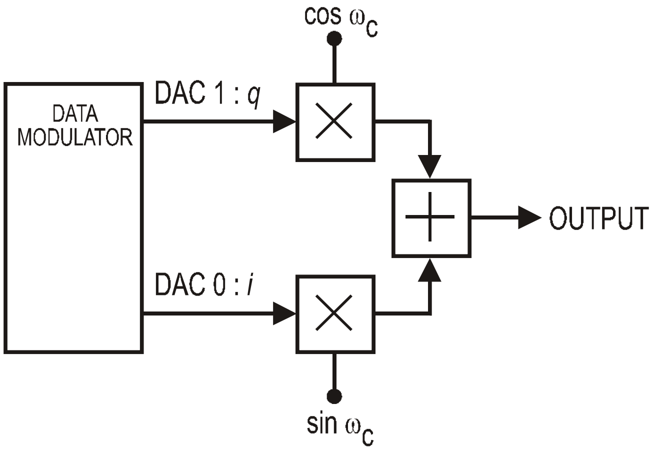


Figure 44: I/Q quadrature modulation

Patch together the experiment as follows, referring to Figure 44:

1. DAC 0 and 1 connect to DAC0 and DAC1 outputs on the DATEx ANALOG I/O
2. $sinw_c$ and $cosw_c$ connect, respectively, to the 100kHz SINE and 100kHz COS from the MASTER SIGNALS block on DATEx.
3. Select any two of the available MULTIPLIER modules and use the DC coupled inputs only.
4. Use the ADDER module which has the dual variable gain knobs. Set both gains to maximum, as actual signal levels will be set to the output signals by the program.

In this experiment both I and Q branches will be utilized. The I branch creates a carrier which has very little modulation and has a very "shallow" envelope. The Q branch creates a smaller DSBSC signal in quadrature with the I branch. Refer to the experiments in Volume 2 for detailed phasor diagrams of this method. Look closely at the amplitude and the offset of the I and Q signals as displayed on the front panel below in Figure 45. You will notice that the real I component is close to 1 V, whereas imaginary component is a small bipolar sinusoid. This implementation varies both the in phase and quadrature components to maintain a constant envelope for the resultant PM signal.

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.

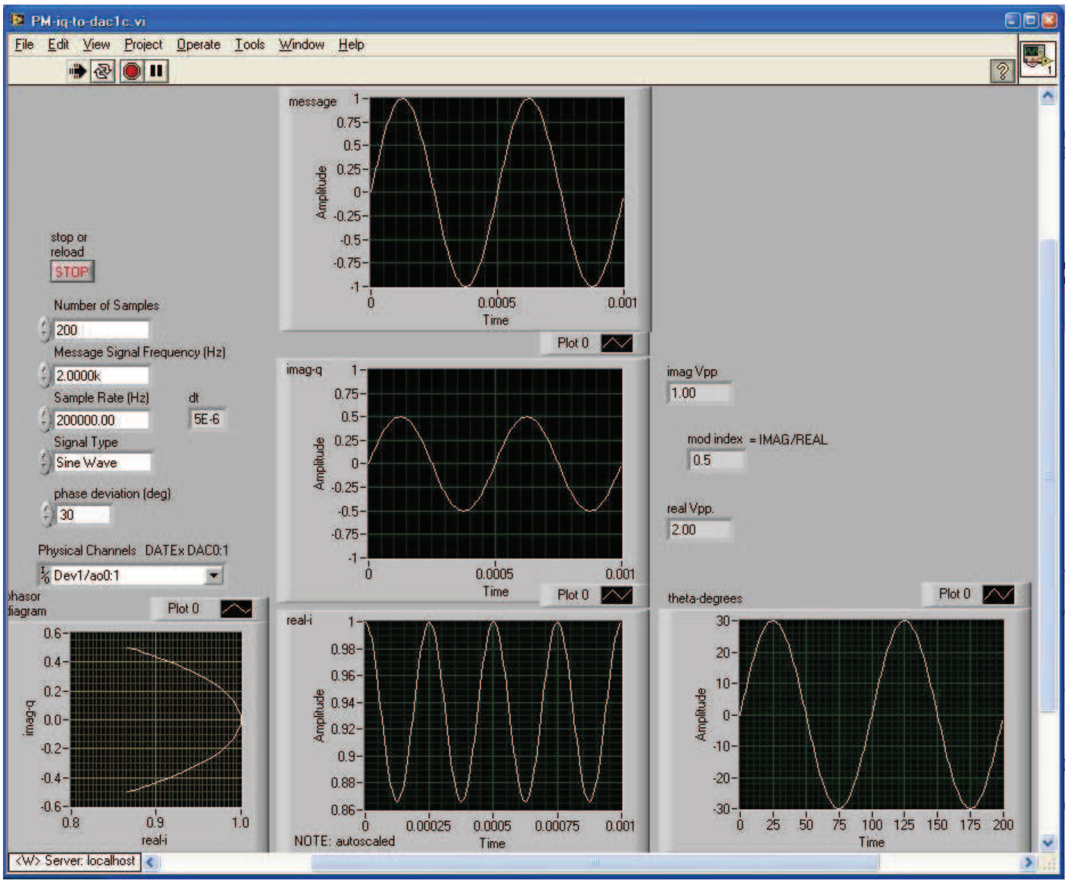


Figure 45: Front panel for LabVIEW based Armstrong's Phase Modulator

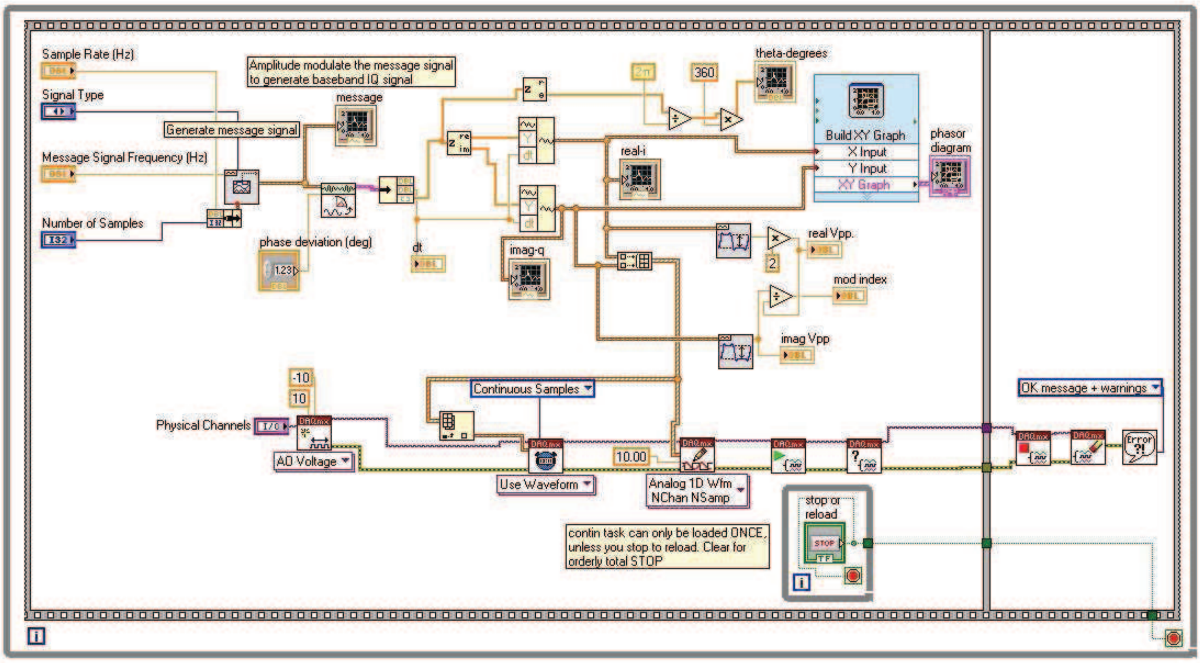


Figure 46: Program for Armstrong's method "PM-iq-to-dac.vi"

Programming tasks:

TASK 1: modify the phase or diagram display on the front panel to remove the auto scaling and display the phasor diagram in correct proportions.

Example: MSK modulation using the LV Modulation Toolkit

In this experiment we will generate MSK modulation using the baseband quadrature modulation structure. We will use the same modulation structure as for the previous experiments, shown below in Figure 47.

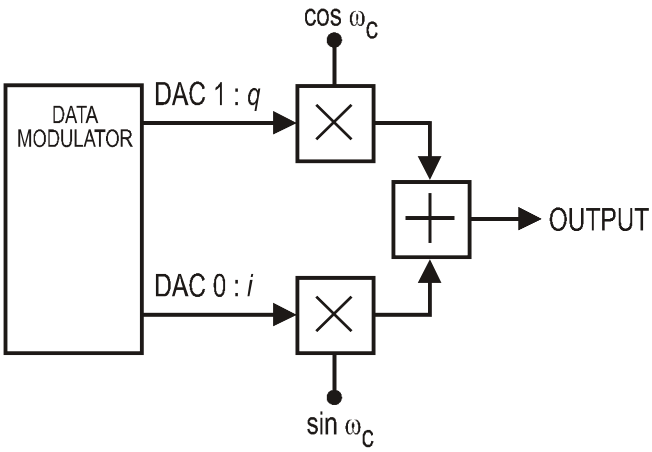


Figure 47: I/Q quadrature modulation

Patch together the experiment as follows, referring to Figure 47:

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 100kHz COS from the MASTER SIGNALS block on DATEx.

3. Select any two of the available MULTIPLIER modules 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 to the output signals by the program.

In this experiment both I and Q branches will be utilized equally. The baseband MSK signals are similar to the QPSK signals encountered in previous experiments, except that the square pulse shapes are half-sinusoidally shaped. These are readily seen by viewing the front panel screenshot of Figure 48.

The sinusoidal shaping of the baseband signals causes the constellation to appear circular.

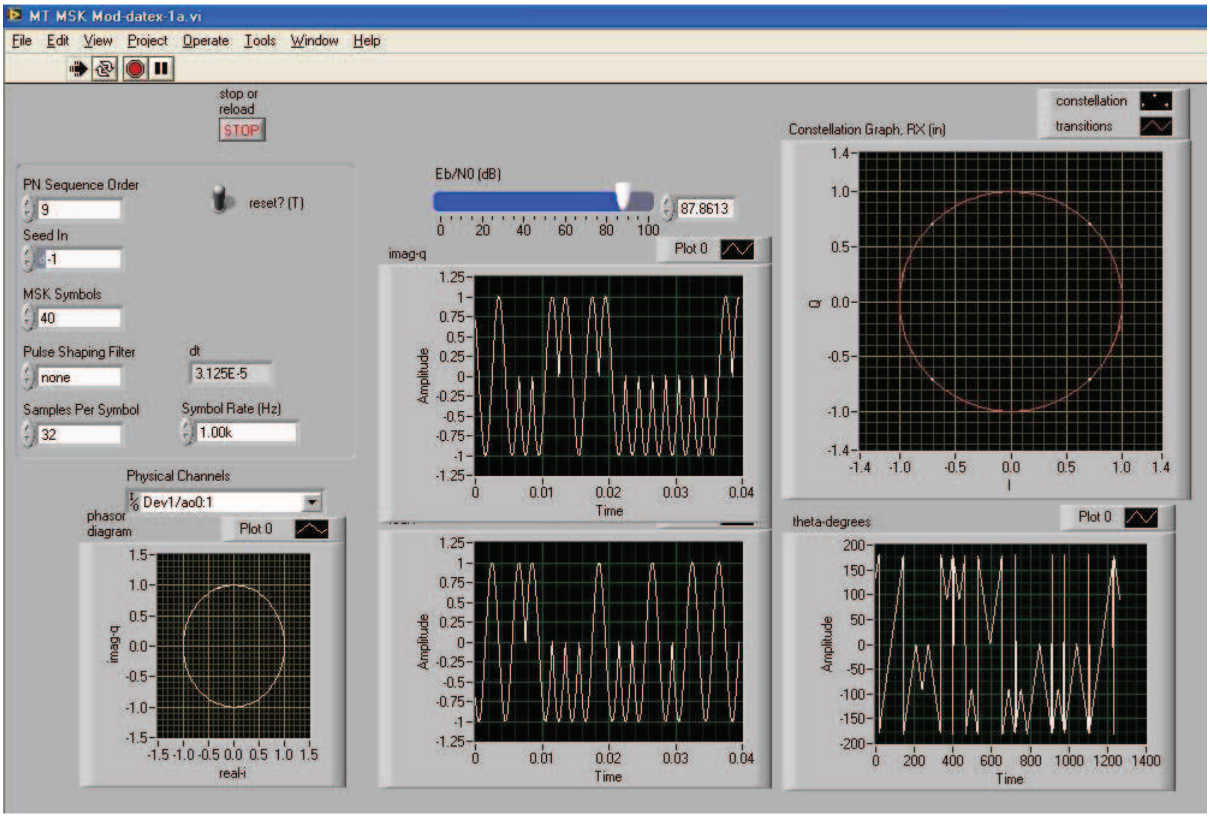


Figure 48: MSK modulator front panel

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 ELVIS Scope and Dynamic Spectrum Analyzer (DSA) instruments running in their own separate windows to view signals on the board itself.

The top half of the program in figure 49 creates the base signals with options for adding noise and filtering. It is derived from an example supplied with the LabVIEW modulation Toolkit for MSK modulation and demodulation. The bottom half of the program displays this signals for evaluation and passes the date to the analogue output terminal, DAC0 and DAC1, on the DATEx board. These signals are then multiplied by quadrature carriers and added to form the MSK signal at the passband as per previous experiments in this chapter.

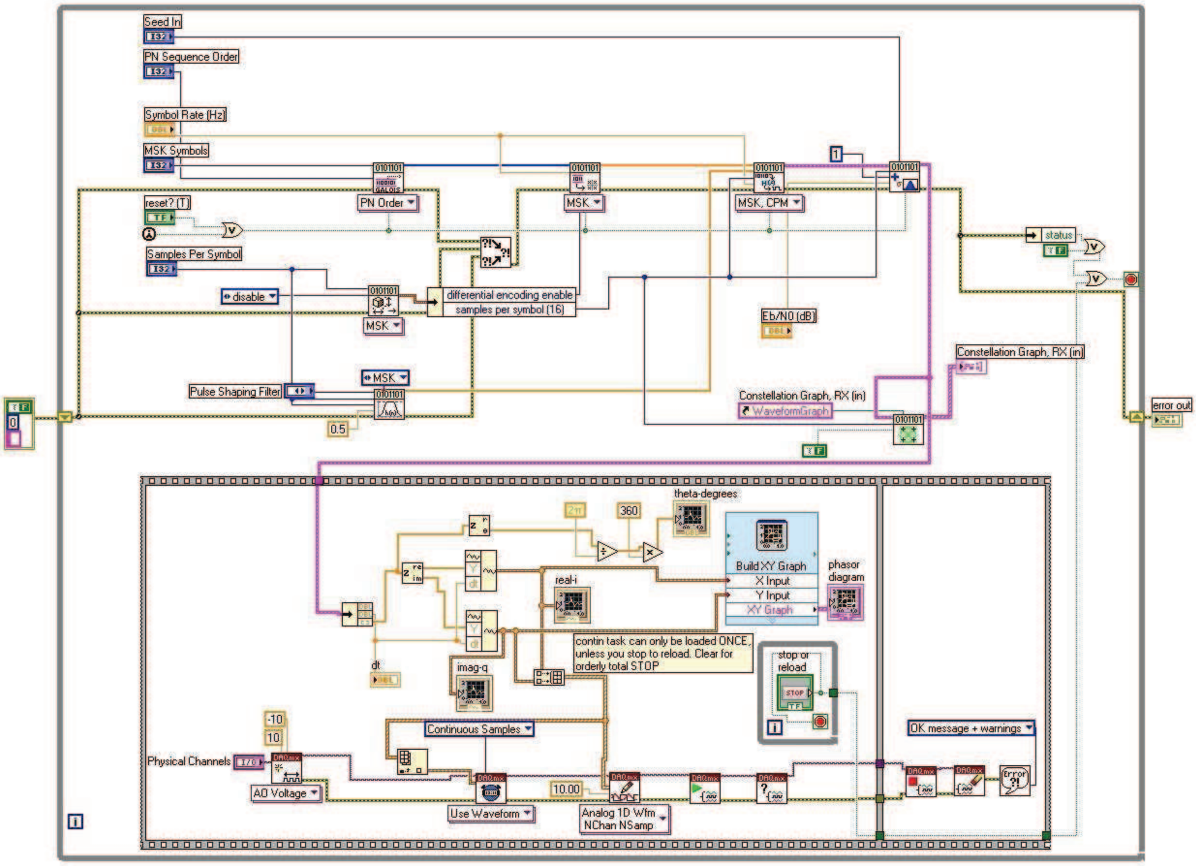


Figure 49: Block diagram for MSK modulation

Programming tasks:

TASK 1: modify the program so as to be able to update the noise levels and other settings in real time.

Example: FM generation using the LV Modulation Toolkit

In this experiment we will generate FM using the baseband quadrature modulation structure. We will use the same modulation structure as for the previous experiments, shown below in Figure 50.

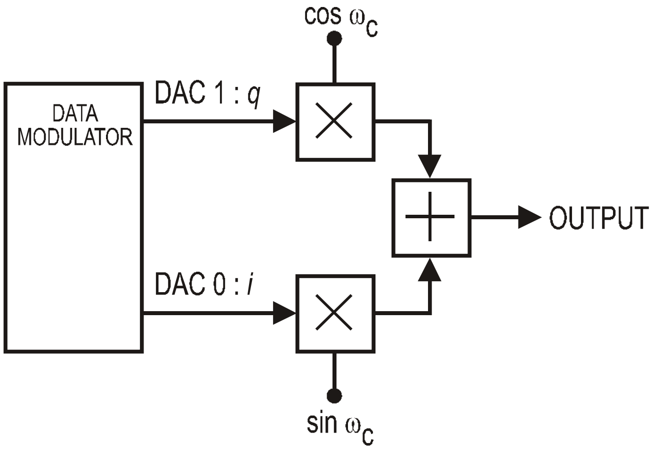


Figure 50: I/Q quadrature modulation

Patch together the experiment as follows, referring to Figure 50:

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 100kHz COS from the MASTER SIGNALS block on DATEx.
3. Select any two of the available MULTIPLIER modules 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 to the output signals by the program.

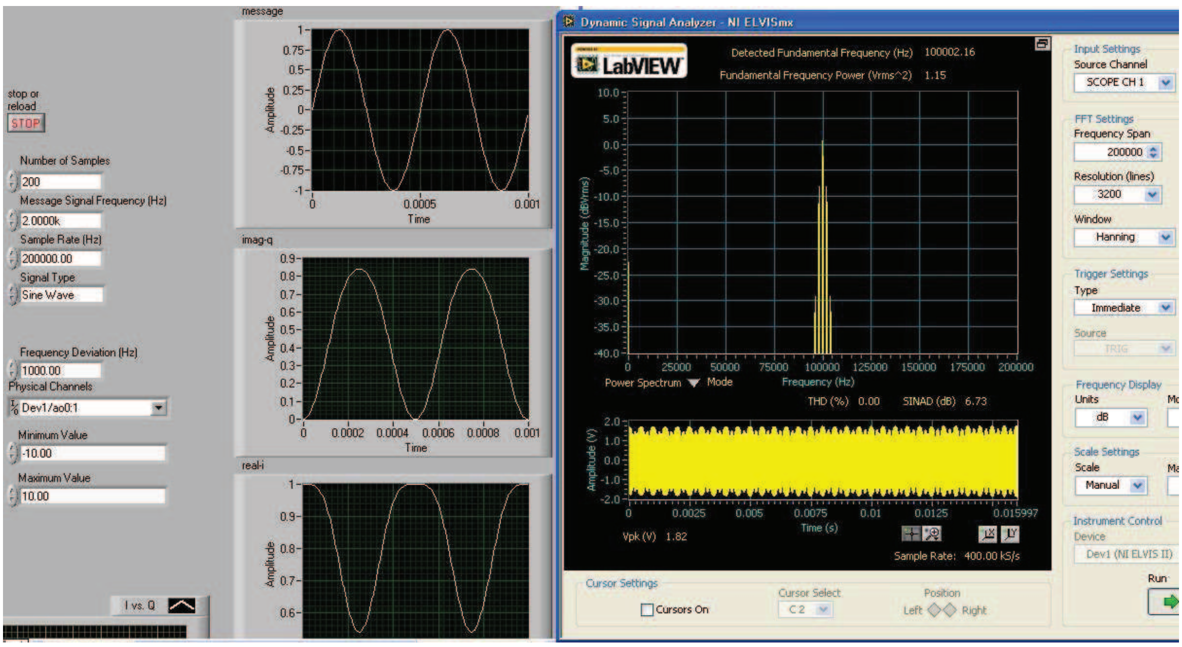


Figure 51: Front panel for FM generation experiment

The I & Q signals generated from the message are able to modulate the phase of the resultant carrier at 100 kHz and vary its frequency proportionally to the message. You can view the spectrum of the generated FM signal using the Dynamic Signal Analyzer and experiment with varying the input message signal to the various other options: Triangle, sawtooth and square wave. It is hard to tell the effect of these different message signals by simply viewing the output spectrum.

To confirm that these various message signals are indeed being FM modulated you will need to demodulate the FM signal. This can be simply achieved by using a zero crossing detector (such as the COMPARATOR module), which clocks a constant width pulse generator (such as the TPG module) and then a low pass filter. (Unfortunately the TPG module on DATEx does not provide a narrow enough pulse for this purpose of demodulation at 100kHz. External equipment such as the EMONA TIMS telecommunications modeling system can be used to implement this method of demodulation.) You may also be able to demodulate this signal using LabVIEW after sampling the signal at a high enough rate using the analogue inputs on the DATEx. The demodulation of FM is not covered in this experiment and may form the basis for an interesting "further programming" task.

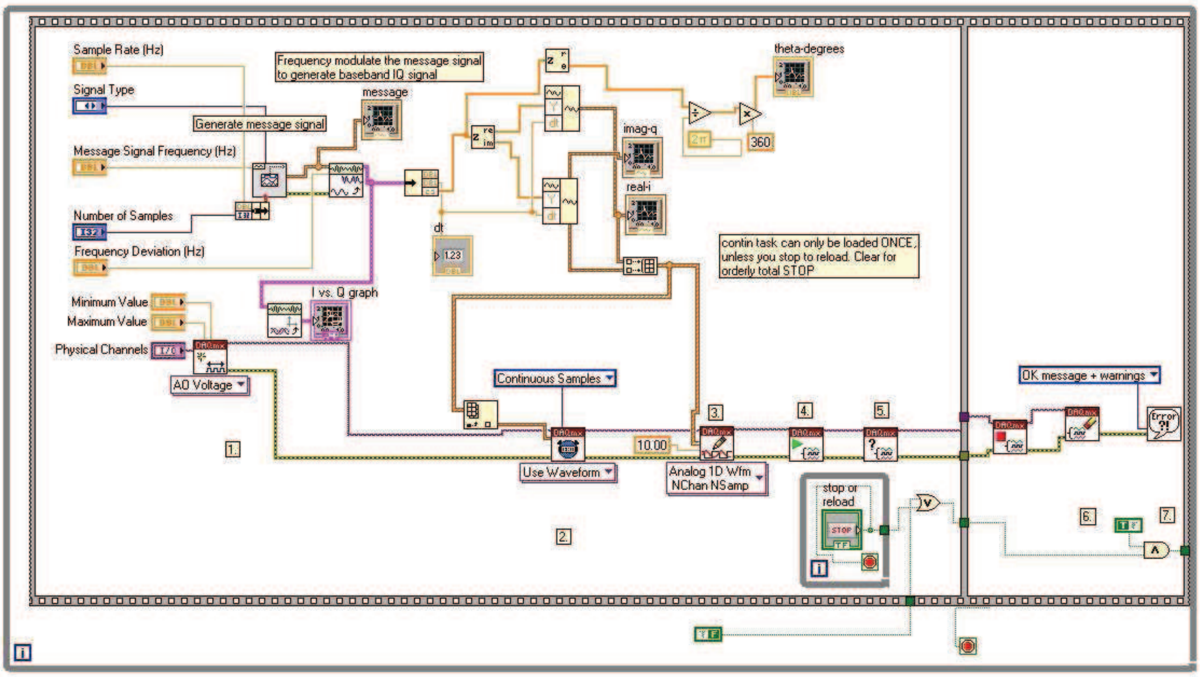


Figure 52: Program block diagram for FM generation experiment

It is interesting to consider how an FM signal is generated using complex I & Q signals. By "drilling down" into the FM modulation block provided by the LabVIEW Modulation Toolkit we can see how this method is implemented. At the lowest level we have a discrete implementation of a fundamental equation for PM generation. This forms the basis for a simple phase modulator. This phase modulator is preceded by an integration block, which forms a simple narrow band FM modulator. Parameters such as frequency deviation are supplied to this FM modulator which is being used in the highest level of the program where it accepts an input message signal. Figure

53 shows this hierarchy of implementation and the overlaying of the blocks mimics the hierarchy of the programming.

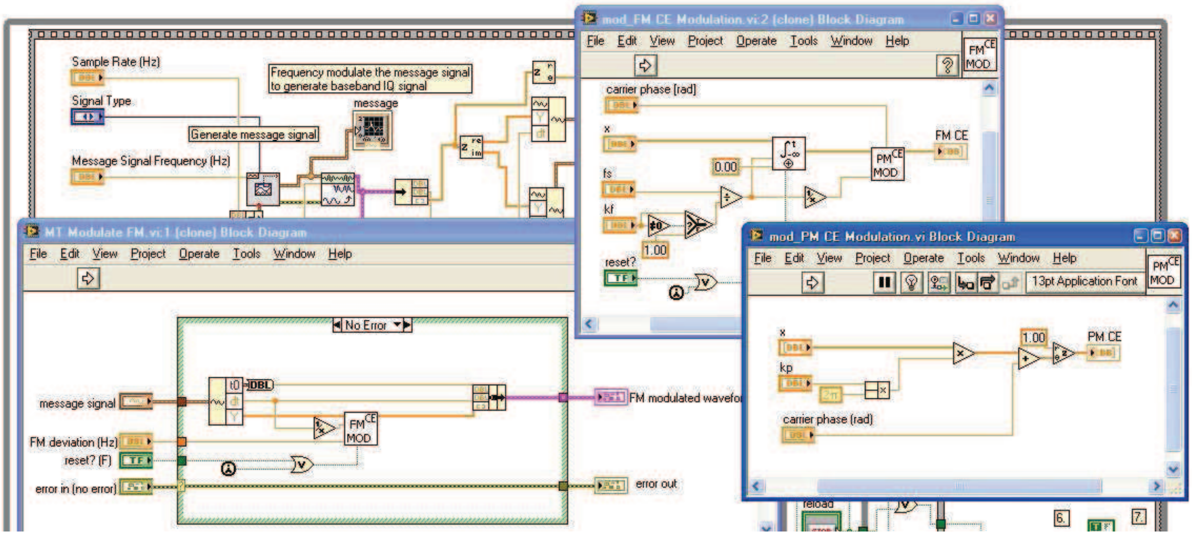


Figure 53: Hierarchy of subVIs used in FM MODULATION block

Take a close look at the PM block which forms the lowest level of subVI in Figure 53. The final block is a "polar to complex" conversion. From previous experiments we have learnt how the modulated signal at the carrier may be expressed as a phasor. This phasor is the resultant of real and imaginary components, which can be expressed as a complex value. It is at this point that the complex values, I & Q, are created from the mathematical theoretical definition of Phase Modulation. It is fascinating to see how the mathematics can be implemented firstly in software and then ultimately brought into real-world hardware. DATEx and LabVIEW together make this interaction easier to comprehend and explore by the student.

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.

10 - Further LabVIEW Programming Tasks

This manual aims to teach you how to use LabVIEW to program and control the individual DATEx blocks. There many different experiments which can be constructed using these blocks and the many functions are available in LabVIEW. You may wish to also use the many telecoms functions in the LabVIEW Modulation toolkit. You can use LabVIEW to generate signals and further process them using the DATEx hardware modules. Or you can process signals from the DATEx in software using LabVIEW. Or a combination of both. DATEx blocks with variable parameters can be controlled from your LabVIEW program.

Following are some further suggestions for advanced programming experiments using LabVIEW and the DATEx /NI ELVIS bundle. These can form the basis for student projects, group work or student thesis work.

- Characterization of hardware circuit blocks in DATEx

- Phase modulation using the PHASE SHIFTER module

- Generating M Level constellations using the ELVIS ANALOG outputs

- M-Level Constellations in a noisy channel

- Adaptive FSK demodulation using the TUNEABLE LPF module

- Automatic nulling of quadrature to signals in a product demodulator

- Viewing eye diagrams using LabVIEW to compute parameters and optimum decision point

- Creating a spectrum analyzer using swept frequencies from the FUNCTION GENERATOR and product demodulation using MULTIPLIERS, as well as ANALOG I/O

- Characterization of DATEx filter blocks using the ELVIS Bode Analyzer

- Generating Line codes in LabVIEW and analyzing their spectrum

- Creating filters in LabVIEW and reviewing their performance on DATEx

- Create a custom X-Y scope display using both DATEx ANALOG I/O as inputs

- Create a custom Spectrum Analyzer using both DATEx ANALOG I/O as inputs

- Demodulate signals in LabVIEW using as input signals to DATEx ANALOG I/O

11 – Controlling DATEx remotely across the Internet

As with many LabVIEW VIs, it is possible to control the DATEx SFP remotely across the Internet. Whilst LabVIEW 'Help' covers this topic in detail, this chapter will outline some of the key steps involved in enabling remote control of the DATEx across a network, be it a LAN or the Internet.

i) Remote control across a network involves establishing a server and client. The DATEx / NI ELVIS units are located at the server. The client windows will enable remote users to vary controls of the DATEx and ELVIS units.

ii) Ensure that the client PC is running an appropriate installation of LabVIEW or at least the LabVIEW runtime engine. The minimum runtime engine required is for "Web browser only".

iii) Place all the VI's that you wish to be accessible remotely to a particular folder of the server PC.

iv) Enable the Web Server feature of the Server PC and create an HTML page with the VI to be controlled embedded in that page. E.g. my_vi.html

v) Run the VIs to make them available

vi) The remote client can address and control the VI at the server PCs IP address e.g.: http://IP_address/my_vi.html

An example of the server PC screen during remote access by clients is shown below in Figure 1. The speed of update will be dependent on the bandwidth of the network connection. Faster ADSL connections are the minimum for satisfactory real time interaction with the NI ELVIS unit.

Most VIs can be optimized to improve the remote access performance if necessary.

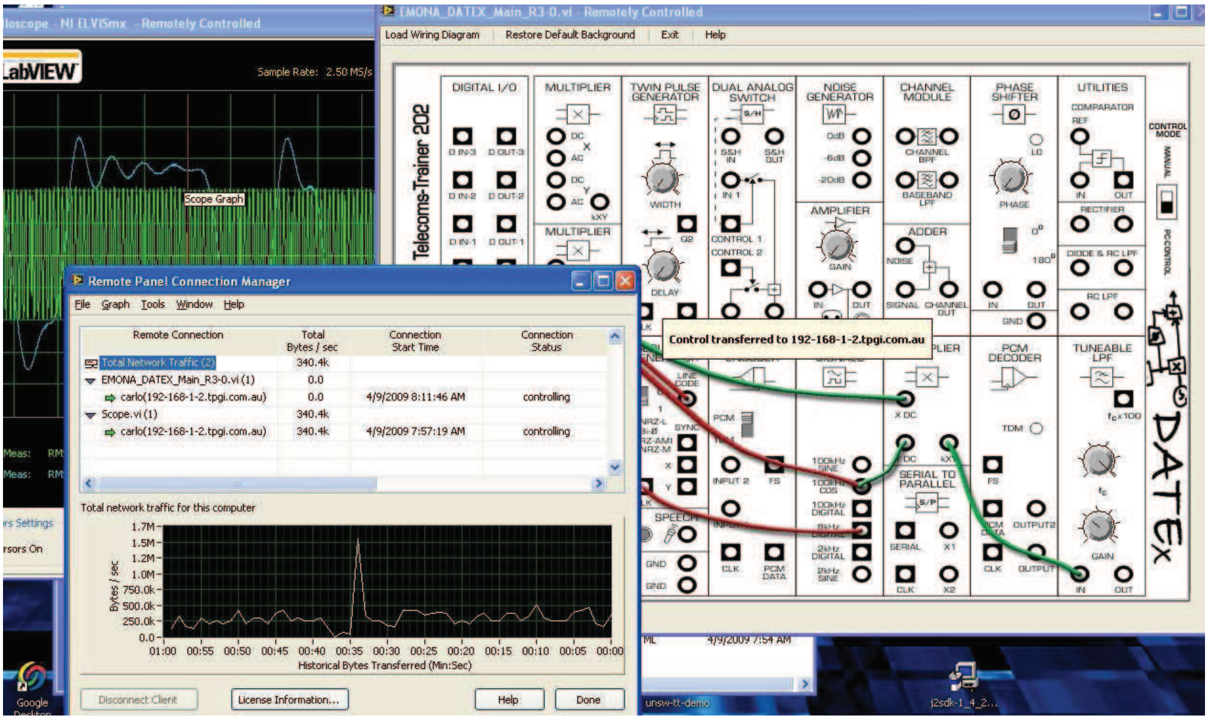


Figure 54: Example Server PC screen during remote access of DATEx / NI ELVIS units

Emona DATEx™ Telecommunications Trainer Lab Manual Volume 3 - Programming and Controlling DATEx with NI LabVIEW.

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