Name: ______________________________

Class: ______________________________

14 - PCM encoding

Experiment 14 – PCM encoding

Preliminary discussion
As you know, digital transmission systems are steadily replacing analog systems in commercial communications applications. This is especially true in telecommunications. That being the case, an understanding of digital transmission systems is crucial for technical people in the communications and telecommunications industries. The remaining experiments in this book use the Emona DATEx to introduce you to several of these systems starting with *pulse code modulation* (PCM).

PCM is a system for converting analog message signals to a serial stream of 0s and 1s. The conversion process is called *encoding*. At its simplest, encoding involves:

- Sampling the analog signal's voltage at regular intervals using a sample-and-hold scheme (demonstrated in Experiment 13).

- Comparing each sample to a set of reference voltages called *quantisation levels*.

- Deciding which quantisation level the sampled voltage is closest to.

- Generating the binary number for that quantisation level.

- Outputting the binary number one bit at a time (that is, in serial form).

- Taking the next sample and repeating the process.

An issue that is crucial to the performance of the PCM system is the encoder's clock frequency. The clock tells the PCM encoder when to sample and, as the previous experiment shows, this must be at least twice the message frequency to avoid aliasing (or, if the message contains more than one sinewave, at least twice its highest frequency).

Another important PCM performance issue relates to the difference between the sample voltage and the quantisation levels that it is compared to. To explain, most sampled voltages will not be the same as any of the quantisation levels. As mentioned above, the PCM Encoder assigns to the sample the quantisation level that is closest to it. However, in the process, the original sample's value is lost and the difference is known as *quantisation error*. Importantly, the error is reproduced when the PCM data is decoded by the receiver because there is no way for the receiver to know what the original sample voltage was. The size of the error is affected by the number of quantisation levels. The more quantisation levels there are (for a given range of sample voltages) the closer they are together. This means that the difference between the quantisation levels and the samples is smaller and so the error is lower.

A little information about the PCM Encoder module on the Emona DATEx
The PCM Encoder module uses a PCM encoding and decoding chip (called a *codec*) to convert analog voltages between -2V and +2V to an 8-bit binary number. With eight bits, it's possible to produce 256 different numbers between 00000000 and 11111111 inclusive. This in turn means that there are 256 quantisation levels (one for each number).

Each binary number is transmitted in serial form in *frames*. The number's most significant bit (called bit-7) is sent first, bit-6 is sent next and so on to the least significant bit (bit-0). The PCM Encoder module also outputs a separate *Frame Synchronisation* signal (*FS*) that goes high at the same time that bit-0 is outputted. The *FS* signal has been included to help with PCM decoding (discussed in the preliminary discussion of Experiment 15) but it can also be used to help "trigger" a scope when looking at the signals that the PCM Encoder module generates.

Figure 1 below shows an example of three frames of a PCM Encoder module's output data (each bit is shown as both a 0 and a 1 because it could be either) together with its clock input and its *FS* output.

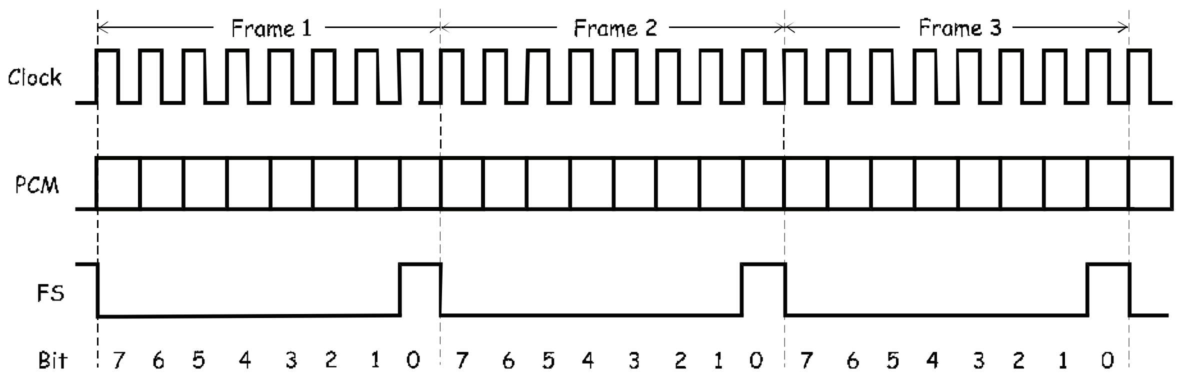

Figure 1

The experiment
For this experiment you'll use the PCM Encoder module on the Emona DATEx to convert the following to PCM: a fixed DC voltage, a variable DC voltage and a continuously changing signal. In the process, you'll verify the operation of PCM encoding and investigate quantisation error a little.

It should take you about 1 hour 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

Procedure

Part A – An introduction to PCM encoding using a static DC voltage

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

9. Adjust the function generator for a 10kHz 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.

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

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

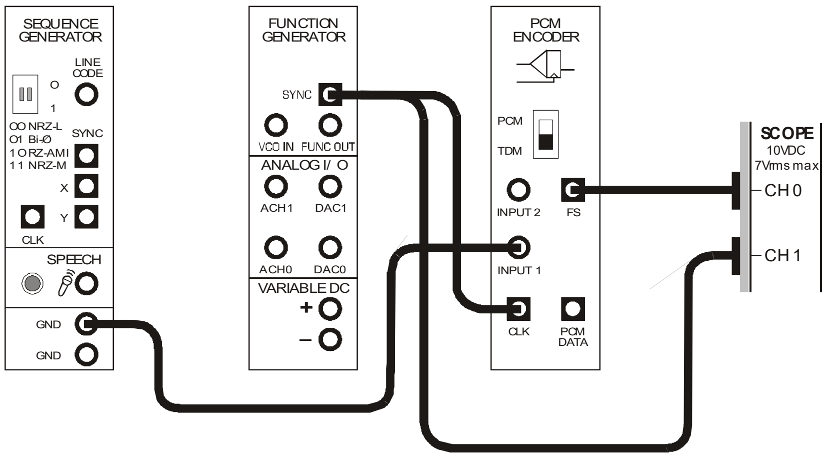

Figure 2

This set-up can be represented by the block diagram in Figure 3 below. The PCM Encoder module is clocked by the function generator output. Its analog input is connected to 0V DC.

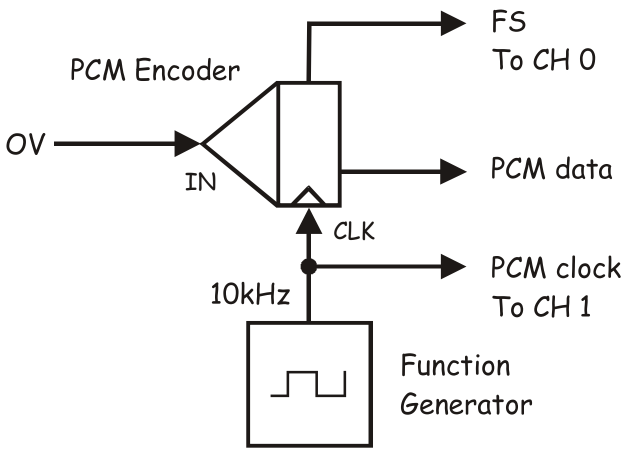

Figure 3

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

12. 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.

13. Locate the PCM Encoder module on the Emona DATEx SFP and set its soft *Mode* switch to the *PCM* position.

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

15. Set up the scope per the procedure in Experiment 1 (page 1-12) with the following changes:

    - *Scale* control for both channels to *2V/div* instead of *1V/div*
    - *Coupling* control for both channels to *DC* instead of *AC*
    - *Trigger Level* control to *2V* instead of *0V*
    - *Timebase* control to *200μs/div* instead of *500μs/div*

16. Set the scope's *Slope* control to the ⏷ position.

Setting the *Slope* control to the "-" position makes the scope start its sweep across the screen when the *FS* signal goes from high to low instead of low to high. You can really notice the difference between the two settings if you flip the scope's *Slope* control back and forth. If you do this, make sure that the *Slope* control finishes on the "-" position.

17. Set the scope's *Timebase* control to the *100μs/div* position.

    Note 1: The *FS* signal's pulse should be one division wide as shown in Figure 4. If it's not, adjust the function generator's output frequency until it is.

    Note 2: Setting the function generator this way makes each bit in the serial data stream one division wide on the graticule's horizontal axis.

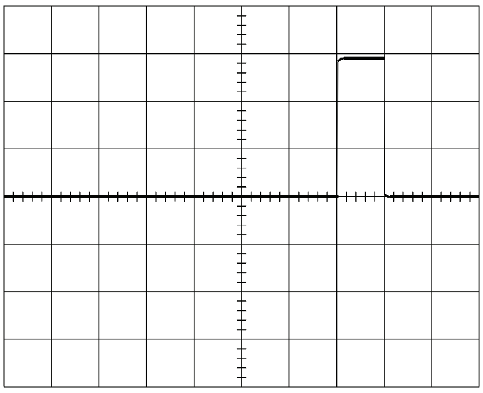
Figure 4

18. Activate the scope's Channel 1 input (by checking the Channel 1 *Enabled* box) to observe the PCM Encoder module's *CLK* input as well as its *FS* output.

    Tip: To see the two waveforms clearly, you may need to adjust the scope so that the two signals are not overlayed.

19. Draw the two waveforms to scale in the space provided below leaving enough room for a third digital signal.

    Tip: Draw the clock signal in the upper third of the graph paper and the *FS* signal in the middle third.

Ask the instructor to check your work before continuing.

20. Connect the scope's Channel 1 input to the PCM Encoder module's output as shown in Figure 5 below.

Remember: Dotted lines show leads already in place.

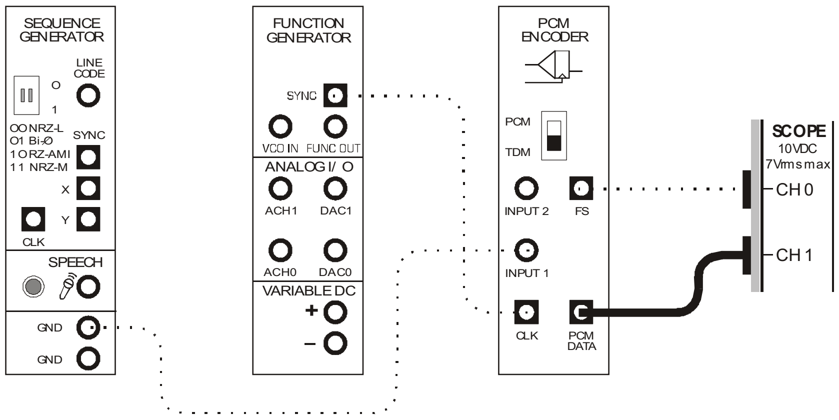

Figure 5

This set-up can be represented by the block diagram in Figure 6 below. Channel 1 should now display 10 bits of the PCM Encoder module's data output. Reading from the left of the display, the first 8 bits belong to one frame and the last two bits belong to the next frame.

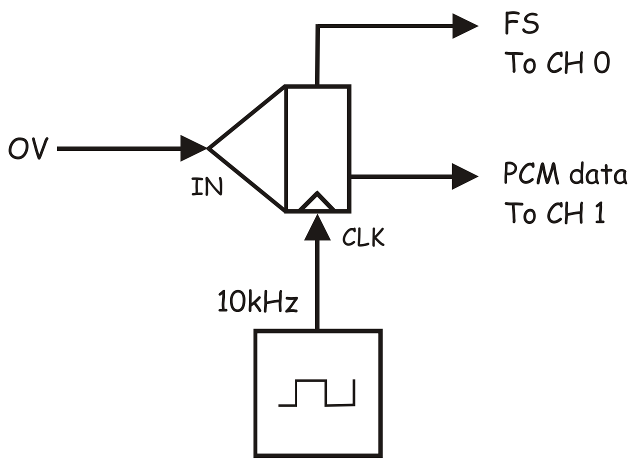

Figure 6

21. Draw this waveform to scale in the space that you left on the graph paper.

Question 1
Indicate on your drawing the start and end of the frame. **Tip:** If you're not sure where these points are, see the preliminary discussion.

Question 2
Indicate on your drawing the start and end of each bit.

Question 3
Indicate on your drawing which bit is bit-0 and which is bit-7.

Question 4
What is the binary number that the PCM Encoder module is outputting?

Question 5
Why does the PCM Encoder module output this code for 0V DC and not 0000000?

Ask the instructor to check your work before continuing.

Part B – PCM encoding of a variable DC voltage
So far, you have used the PCM Encoder module to convert a fixed DC voltage (0V) to PCM. The next part of the experiment lets you see what happens when you vary the DC voltage.

22. Launch and run the NI ELVIS II Variable Power Supplies VI.

23. Set the Variable Power Supplies two outputs to 0V.

24. Unplug the patch lead connected to the ground socket.

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

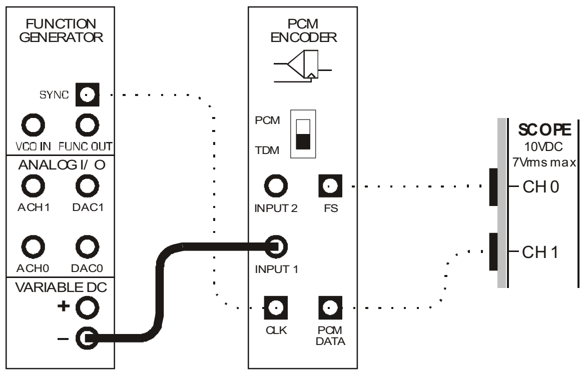

Figure 7

This set-up can be represented by the block diagram in Figure 8 on the next page. The NI ELVIS II Variable Power Supplies is used to let you vary the DC voltage on the PCM Encoder module's input. The scope's external trigger input is used to obtain a stable display.

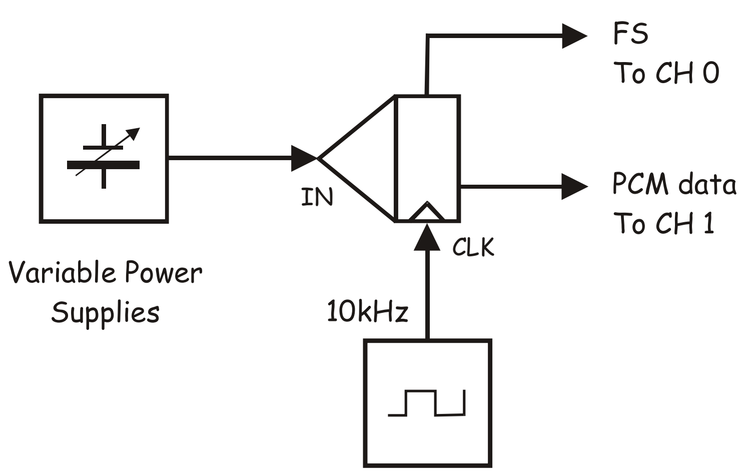

Figure 8

26. Determine the code on the PCM Encoder module's output.

    Tip: Remember, the first eight horizontal divisions of the scope's graticule correspond with one frame of the PCM Encoder module's output.

    Note: You should find that the PCM Encoder module's output is a binary number that is reasonably close to the code you determined earlier when the module's input was connected directly to ground.

Ask the instructor to check your work before continuing.

27. Increase the Variable Power Supplies' negative output voltage in -0.1V increments and note what happens to the binary number on the PCM Encoder module's output.

    Tip: This is easiest to do by simply typing the required voltage in the field under the negative output's *Voltage* control. When you do, don't forget to put a minus sign in front of the voltage you enter.

Question 6
What happens to the binary number as the input voltage increases in the negative direction?

---

28. Determine the first instance of the changing negative voltage that produces the number 00000000 on the PCM Encoder module's output.

29. Record this voltage in Table 1 below.

Table 1

PCM Encoder's output code	PCM Encoder's input voltage
00000000	

✓ Ask the instructor to check your work before continuing.

30. Modify the set-up as shown in Figure 9 below.

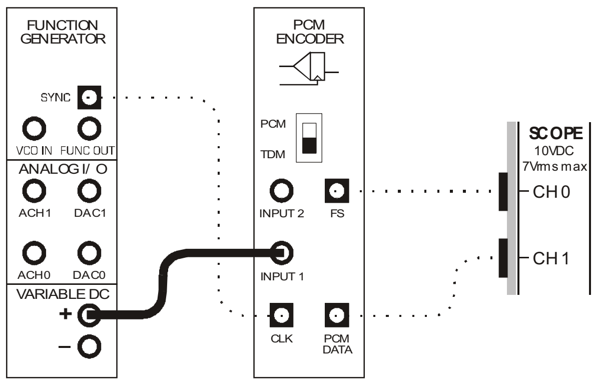

Figure 9

This set-up can be represented by the block diagram in Figure 10 below.

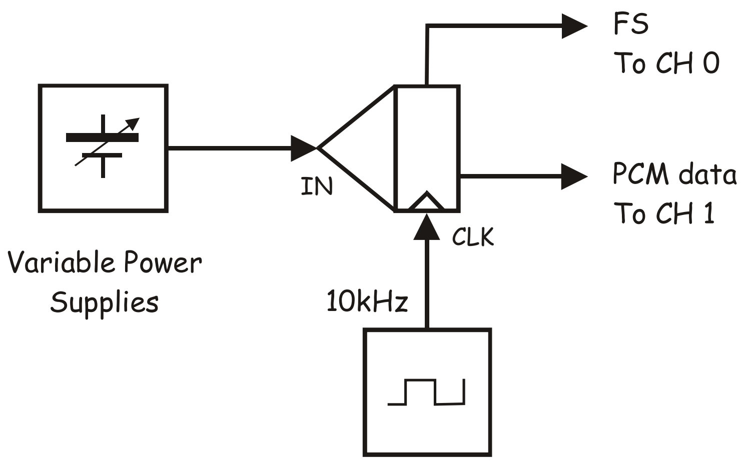

Figure 10

31. Increase the Variable Power Supplies' positive output voltage in +0.1V increments and note what happens to the binary number on the PCM Encoder module's output.

Question 7
What happens to the binary number as the input voltage increases in the positive direction?

___

32. Determine the lowest positive voltage that produces the number 11111111 on the PCM Encoder module's output.

33. Record this voltage in Table 2 below.

Table 2

PCM Encoder's output code	PCM Encoder's input voltage
11111111	

Question 8
Based on the information in Tables 1 & 2, what is the maximum allowable peak-to-peak voltage for an AC signal on the PCM Encoder module's *INPUT*?

___

Question 9
Calculate the difference between the PCM Encoder module's quantisation levels by subtracting the values in Tables 1 & 2 and dividing the number by 256 (the number of codes).

___

___

✓ Ask the instructor to check your work before continuing.

Part C – PCM encoding of continuously changing voltages
Now let's see what happens when the PCM encoder is used to convert continuously changing signals like a sinewave.

34. Close the Variable Power Supplies VI.

35. Disconnect the plugs to the Variable Power Supplies positive output.

36. Modify the set-up as shown in Figure 11 below.

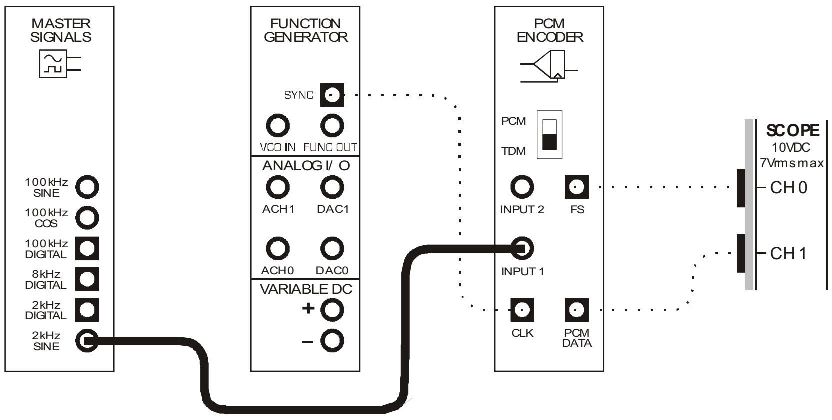

Figure 11

37. Set the function generator's output frequency to 50kHz.

38. Watch the PCM Encoder module's output on the scope's display.

Question 10
Why does the code on PCM Encoder module's output change continuously?

Ask the instructor to check your work before finishing.

Name: ______________________

Class: ______________________

15 - PCM decoding

Experiment 15 – PCM decoding

Preliminary discussion
The previous experiment introduced you to the basics of pulse code modulation (PCM) which you'll recall is a system for converting message signals to a continuous serial stream of binary numbers (*encoding*). Recovering the message from the serial stream of binary numbers is called *decoding*.

At its simplest, decoding involves:

- Identifying each new frame in the data stream.
- Extracting the binary numbers from each frame.
- Generating a voltage that is proportional to the binary number.
- Holding the voltage on the output until the next frame has been decoded (forming a pulse amplitude modulation (PAM) version of the original message signal).
- Reconstructing the message by passing the PAM signal through a low-pass filter.

The PCM decoder's clock frequency is crucial to the correct operation of simple decoding systems. If it's not the same frequency as the encoder's clock, some of the transmitted bits are read twice while others are completely missed. This results in some of the transmitted numbers being incorrectly interpreted, which in turn causes the PCM decoder to output an incorrect voltage. The error is audible if it occurs often enough. Some decoders manage this issue by being able to "self-clock".

There is another issue crucial to PCM decoding. The decoder must be able to detect the beginning of each frame. If this isn't done correctly, every number is incorrectly interpreted. The synchronising of the frames can be managed in one of two ways. The PCM encoder can generate a special *frame synchronisation* signal that can be used by the decoder though this has the disadvantage of needing an additional signal to be sent. Alternatively, a frame synchronisation code can be embedded in the serial data stream that is used by the decoder to work out when the frame starts.

A little information about the DATEx PCM Decoder module
Like the PCM Encoder module on the Emona DATEx, the PCM Decoder module works with 8-bit binary numbers. For 00000000 the PCM Decoder module outputs -2V and for 11111111 it outputs +2V. For numbers in between, the output is a proportional voltage between ±2V. For example, the number 10000000 is half way between 00000000 and 11111111 and so for this input the module outputs 0V (which is half way between +2V and -2V).

The PCM Decoder module is not self-clocking and so it needs a digital signal on the *CLK* input to operate. Importantly, for the PCM Decoder module to correctly decode the PCM data generated by the PCM Encoder module, it must have the same clock signal. In other words, the decoder's clock must be "stolen" from the encoder.

Similarly, the PCM Decoder module cannot self-detect the beginning of each new frame and so it must have a frame synchronisation signal on its *FS* input to do this.

The experiment
For this experiment you'll use the Emona DATEx to convert a sinewave and speech to a PCM data stream then convert it to a PAM signal using the PCM Decoder module. For this to work correctly, the decoder's clock and frame synchronisation signal are simply "stolen" the PCM Encoder module. You'll then recover the message using the Tuneable Low-pass filter module.

It should take you about 45 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
- One set of headphones (stereo)

Procedure

Part A – Setting up the PCM encoder
To experiment with PCM decoding you need PCM data. The first part of the experiment gets you to set up a PCM encoder.

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. Launch and run the NI ELVIS II Variable Power Supplies VI.

9. Set the Variable Power Supplies' positive output to 0V.

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

11. Set the PCM Encoder module's soft *Mode* switch to the *PCM* position.

12. Connect the set-up shown in Figure 1 below.

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

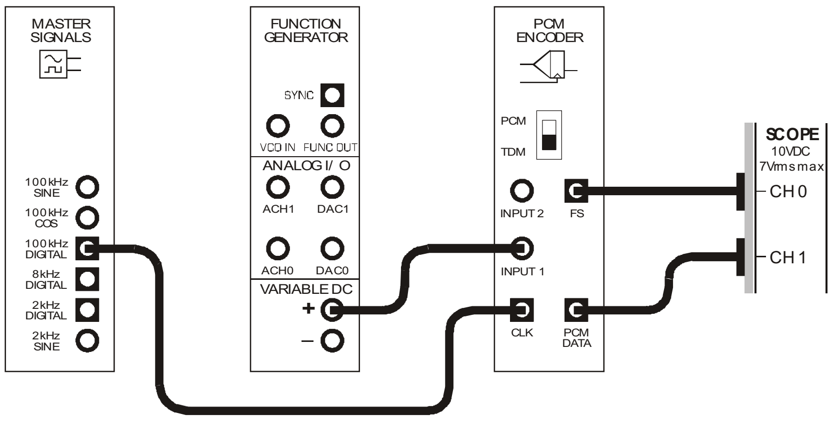

Figure 1

This set-up can be represented by the block diagram in Figure 2 below. The PCM Encoder module is clocked by the Master Signals module's *100kHz DIGITAL* output. Its analog input is the Variable Power Supplies' positive output.

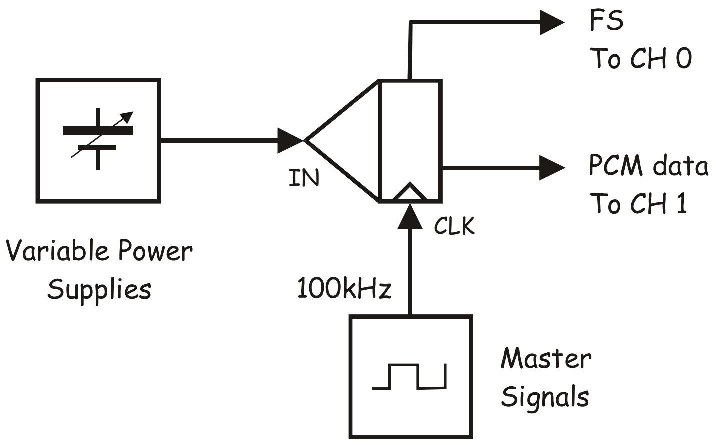

Figure 2

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

14. Set up the scope per the procedure in Experiment 1 (page 1-12) with the following changes:

    - *Scale* control for both channels to *2V/div* instead of *1V/div*
    - *Coupling* control for both channels to *DC* instead of *AC*
    - *Trigger Level* control to *2V* instead of *0V*
    - *Timebase* control to *10µs/div* instead of *500µs/div*

15. Set the scope's *Slope* control to the "-" position.

16. Activate the scope's Channel 1 input (by checking the Channel 1 *Enabled* box) to observe the PCM Encoder module's *PCM DATA* output as well as its *FS* output.

17. Vary the Variable Power Supplies positive output *Voltage* control left and right (but don't exceed 2.5V).

    Note: If your set-up is working correctly, this last step should cause the number on PCM Encoder module's *PCM DATA* output to go down and up. If it does, carry on to the next step. If not, check your wiring or ask the instructor for help.

18. Close the Variable Power Supplies VI.

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

20. Adjust the function generator using its soft controls for an output with the following specifications:

    - Waveshape: Sine
    - Frequency: 500Hz
    - Amplitude: 4Vpp
    - DC Offset: 0V

21. Disconnect the plug to the Variable Power Supplies' positive output.

22. Modify the set-up as shown in Figure 3 below.

Remember: Dotted lines show leads already in place.

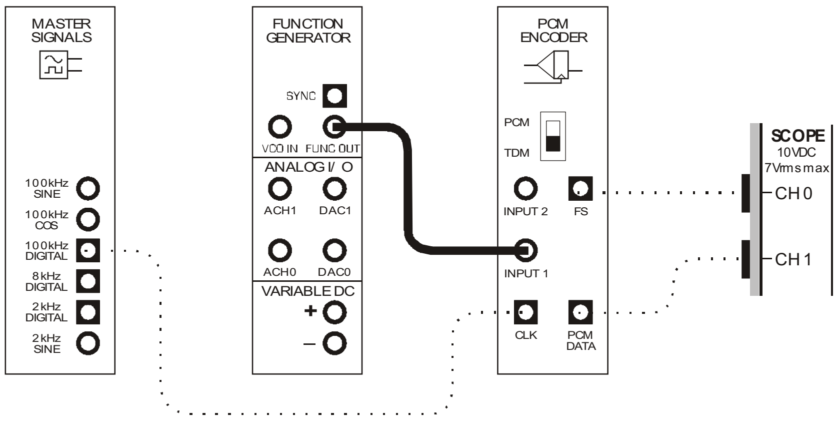

Figure 3

This set-up can be represented by the block diagram in Figure 4 below. Notice that the PCM Encoder module's input is now the function generator's output.

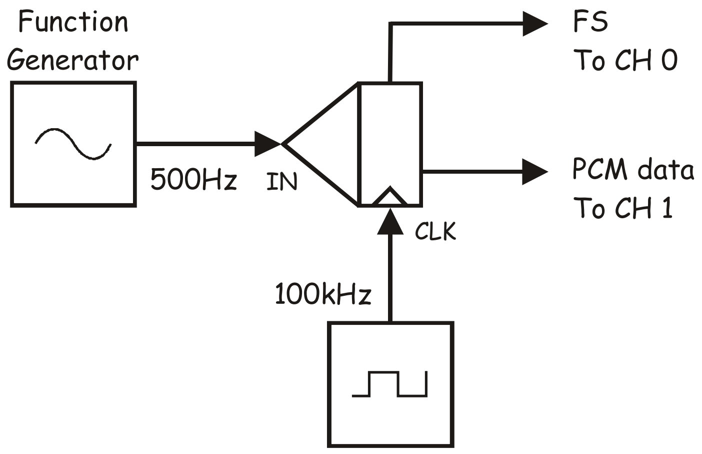

Figure 4

As the PCM Encoder module's input is a sinewave, the module's input voltage is continuously changing. This means that you should notice the *PCM DATA* output changing continuously also.

Ask the instructor to check your work before continuing.

Part B – Decoding the PCM data

23. Deactivate the scope's Channel 1 input.

24. Modify the set-up as shown in Figure 5 below.

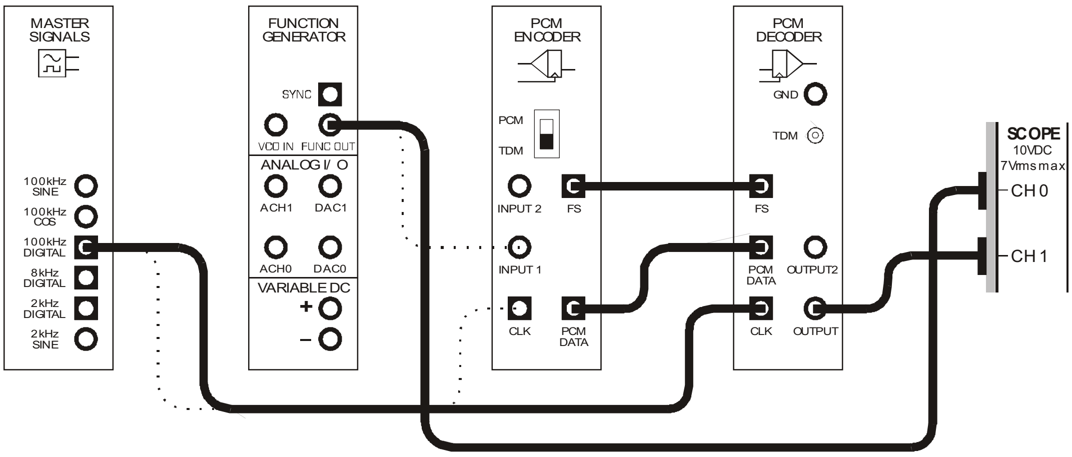

Figure 5

The entire set-up can be represented by the block diagram in Figure 6 on the next page. Notice that the decoder's clock and frame synchronisation information are "stolen" from the encoder.

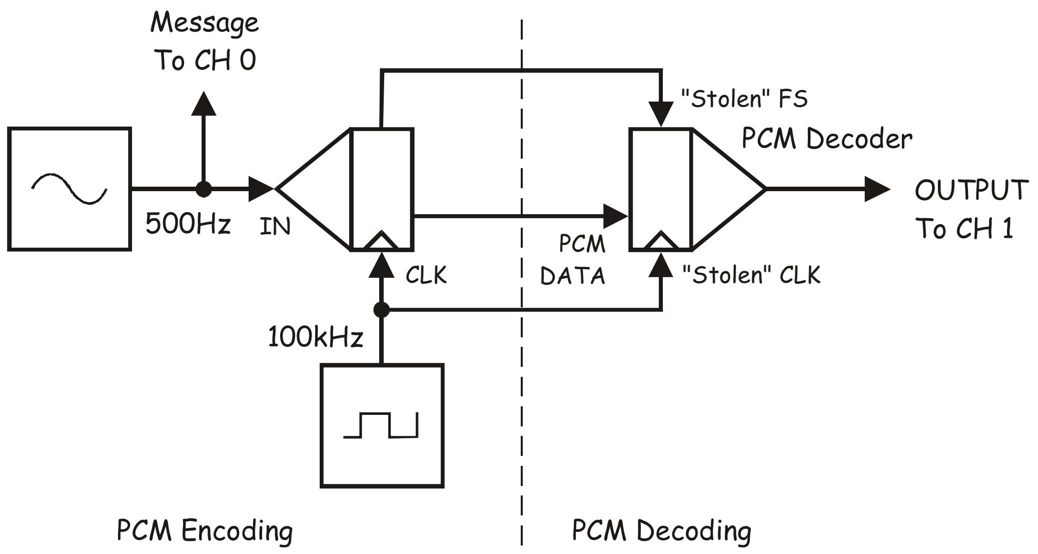

Figure 6

25. Adjust the scope as follows:

- *Scale* control for both channels to *1V/div*
- *Coupling* control for both channels to *AC*
- *Trigger Level* control to *0V*
- *Slope* control back to the "+" position
- *Timebase* control to *500µs/div*

26. Activate the scope's Channel 1 input to observe the PCM Decoder module's output as well as the message signal.

Question 1
What does the PCM Decoder's "stepped" output tell you about the type of signal that it is? **Tip:** If you're not sure, see the preliminary discussion for this experiment or for Experiment 13.

Ask the instructor to check
your work before continuing.

The PCM Decoder module's output signal looks very similar to the message. However, they're not the same. Remember that a "sampled" message contains many sinewaves in addition to the message. The next part of this experiment lets you verify this using the NI ELVIS II Dynamic Signal Analyzer.

27. Suspend the scope's VI.

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

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

Input Settings

- *Source Channel* to *SCOPE CH 1*
- 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 *Edge*

Frequency Display

- *Units* to *dB*
- *Mode* to *RMS*
- *Scale* to *Auto*
- *Cursors On* box unchecked (for now)

30. Activate the signal analyzer's cursors (by checking *Cursors On* box).

31. Use the signal analyzer's *C1* cursor to examine the frequency of the significant sinewaves that make up the sampled message.

32. Use the *C1* cursor to locate the sinewave in the sampled message that has the same the frequency as the original message.

Ask the instructor to check
your work before continuing.

You have probably just noticed that some of the extra sinewaves in the sampled message are at audible frequencies (that is, between about 20Hz and 20kHz, and in particular those about 12.5kHz). This means that, although the message and sampled messages are similar in shape, you may be able to hear a difference between them. Keep in mind that many of the components below the -40dB line are not significant. Signals that are visually below this level (but bigger numerically) are less than 1% of the level of any signals at 0dB.

Where do these "extra" components come from?
As the data clock rate is 100kHz, the "frame rate" is 1/8th of that because there are 8 bits per frame. Therefore, the PCM system is sampling the message at 12.5kHz or every 80μs.

Importantly, this means that the PCM module's output changes voltage in steps every 80μs. The spectral composition of these steps creates extra "images" of the message at 12.5kHz, 2 x 12.5kHz (25kHz), 3 x 12.5kHz (37.5kHz) and so on. It is these "images" that you can see in the spectrum.

33. Add the Amplifier module to the set-up as shown in Figure 7 below leaving the scope's connections as they are.

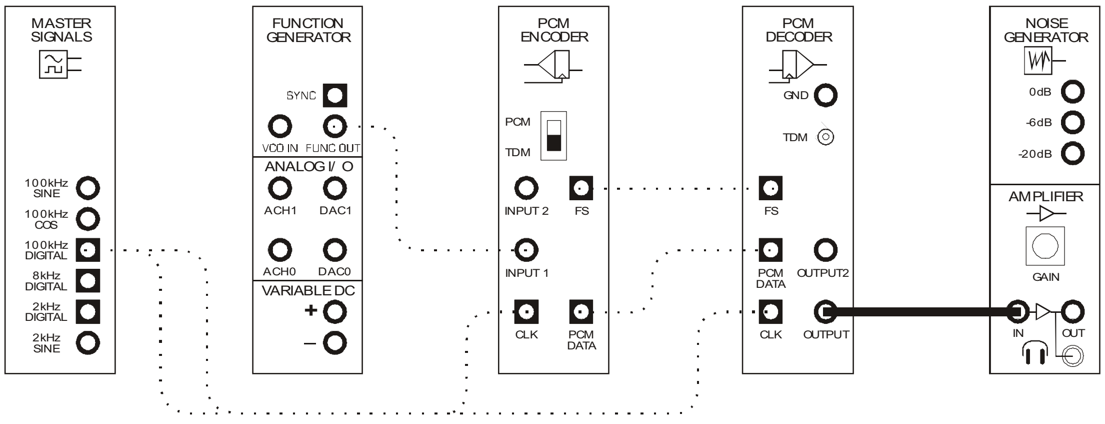

Figure 7

34. Locate the Amplifier module on the DATEx SFP and turn its soft *Gain* control fully anti-clockwise.

35. Without wearing the headphones, plug them into the Amplifier module's headphone socket.

36. Put the headphones on.

37. Turn the Amplifier module's soft *Gain* control clockwise until you can comfortably hear the PCM Decoder module's output.

38. Listen to how the sampled message sounds and commit it to memory.

39. Disconnect the Amplifier module's lead where it plugs to the PCM Decoder module's output.

40. Modify the set-up as shown in Figure 8 below, again leaving the scope's connections as they are.

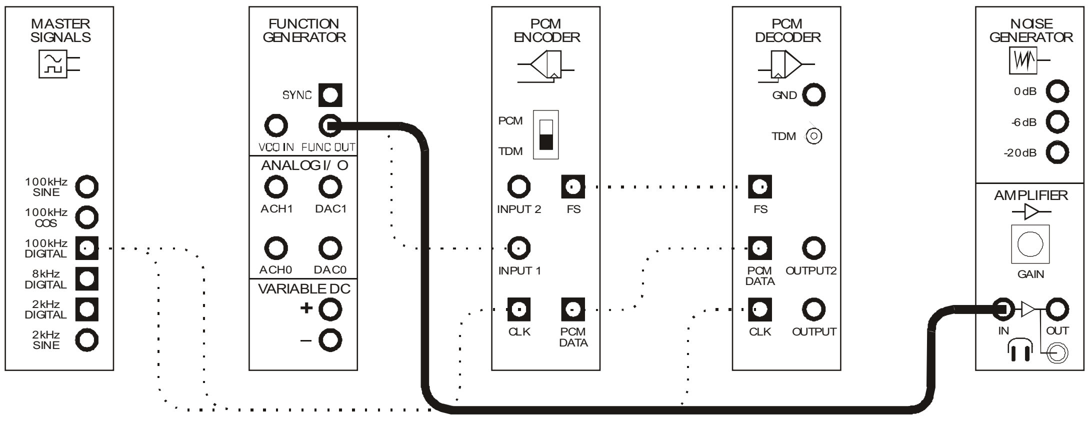

Figure 8

41. Compare the sound of the two signals. You should notice that they're similar but clearly different.

Question 2
What must be done to the PCM Decoder module's output to reconstruct the message properly?

Ask the instructor to check
your work before continuing.

Part C – Encoding and decoding speech

So far, this experiment has encoded and decoded a sinewave for the message. The next part of the experiment lets you do the same with speech.

42. Close the signal analyzer's VI and restart the scope's VI.

43. Adjust the scope so that you can observe two or so cycles of the original and sampled messages again.

    Tip: Don't forget to check that the scope's *Trigger Source* control is set to the *CH 0* position.

44. Completely remove the Amplifier module from the set-up while leaving the rest of the leads in place.

45. Disconnect the plugs to the function generator's output.

46. Modify the set-up as shown in Figure 9 below.

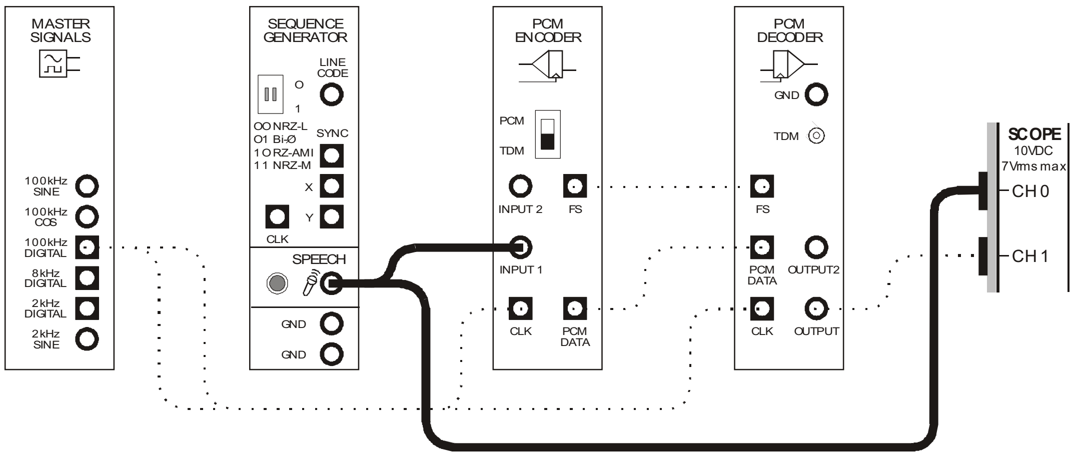

Figure 9

47. Hum and talk into the microphone while watching the scope's display.

Ask the instructor to check your work before continuing.

Part D – Recovering the message

As mentioned earlier, the message can be reconstructed from the PCM Decoder module's output signal using a low-pass filter. This part of the experiment lets you do this.

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

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

50. Disconnect the plugs to the Speech module's output.

51. Modify the set-up as shown in Figure 10 below.

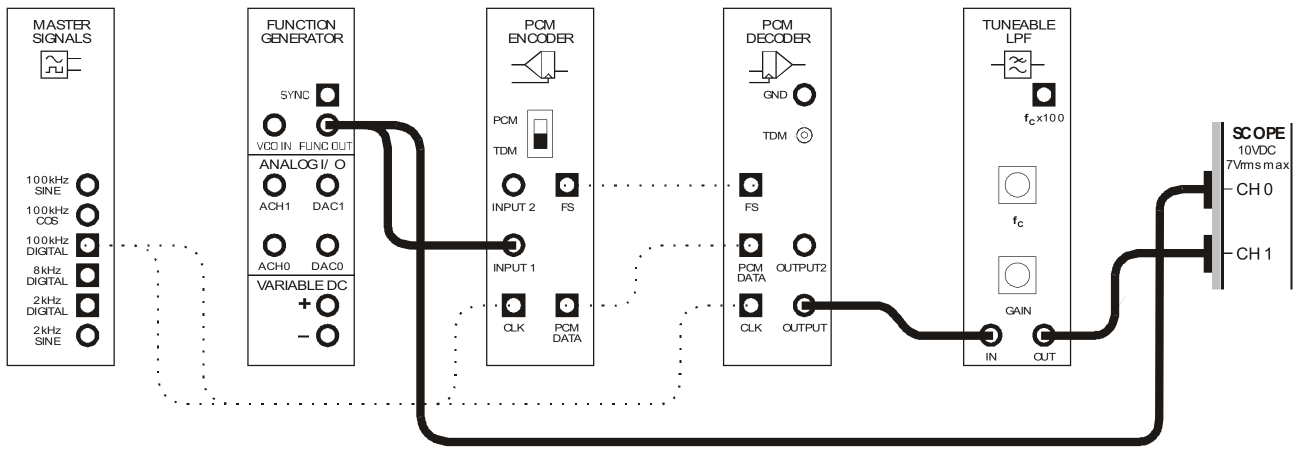

Figure 10

The entire set-up can be represented by the block diagram in Figure 11 on the next page. The Tuneable Low-pass Filter module is used to reconstruct the original message from the PCM Decoder module's PAM output.

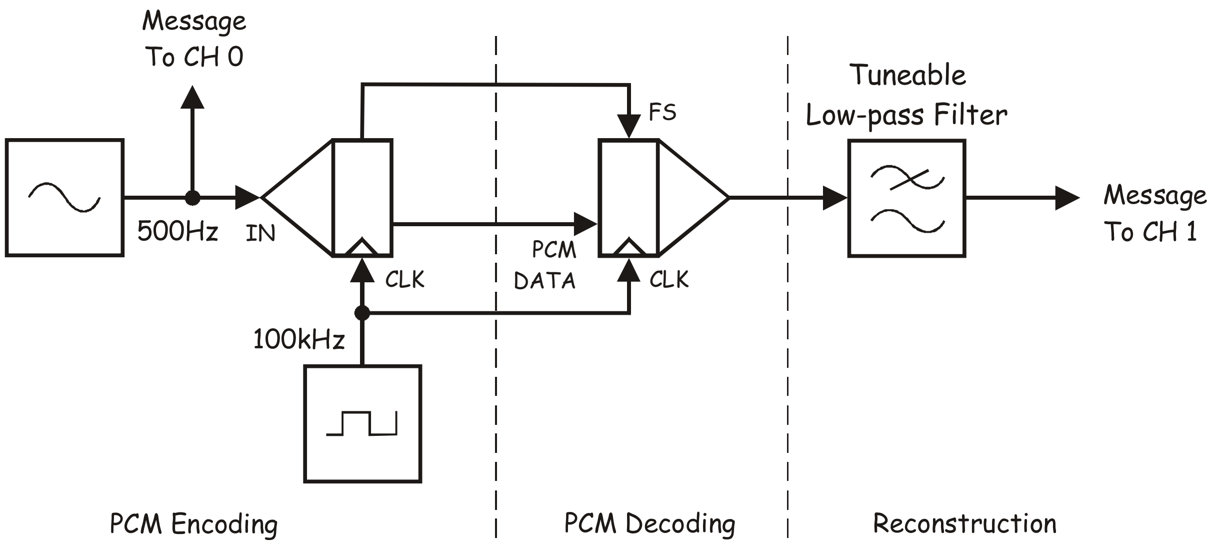

Figure 11

52. Slowly turn the Tuneable Low-pass Filter module's soft *Cut-off Frequency* control clockwise and stop the moment the message signal has been reconstructed (ignoring phase shift).

The two signals are clearly the same so let's see what your hearing tells you.

53. Add the Amplifier module to the set-up as shown in Figure 12 below leaving the scope's connections as they are.

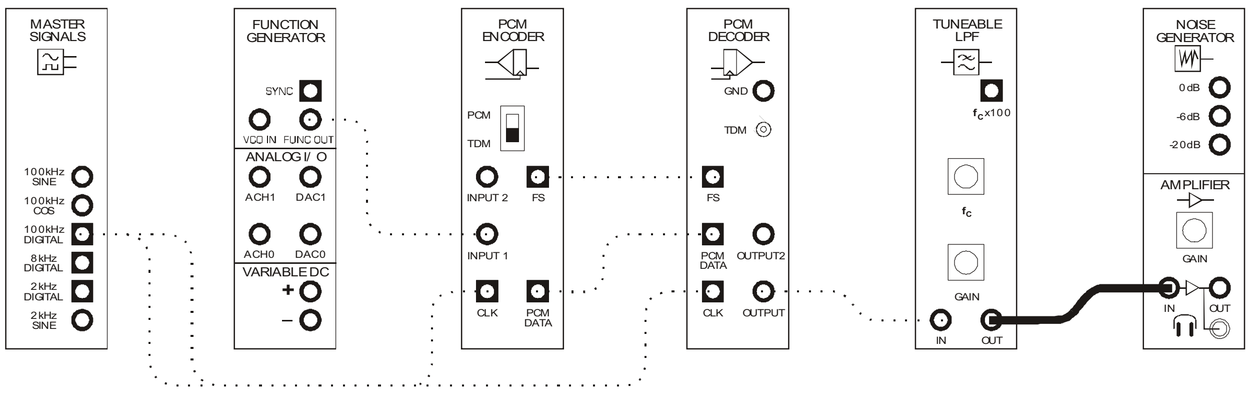

Figure 12

54. Turn the Amplifier module's soft *Gain* control fully anti-clockwise.

55. Put the headphones on.

56. Turn the Amplifier module's soft *Gain* control clockwise until you can comfortably hear the Tuneable Low-pass Filter module's output.

57. Commit the recovered message's sound to memory.

58. Disconnect the Amplifier module's lead where it plugs to the PCM Decoder module's output and connect it to the function generator's output (in the same way that you did when wiring the set-up in Figure 8).

59. Compare the sound of the two signals. You should find that they're very similar.

Question 3
Even though the two signals look and sound the same, why isn't the reconstructed message a perfect copy of the original message? **Tip:** If you're not sure, see the preliminary discussion for Experiment 14.

____

____

✓ Ask the instructor to check your work before finishing.