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.

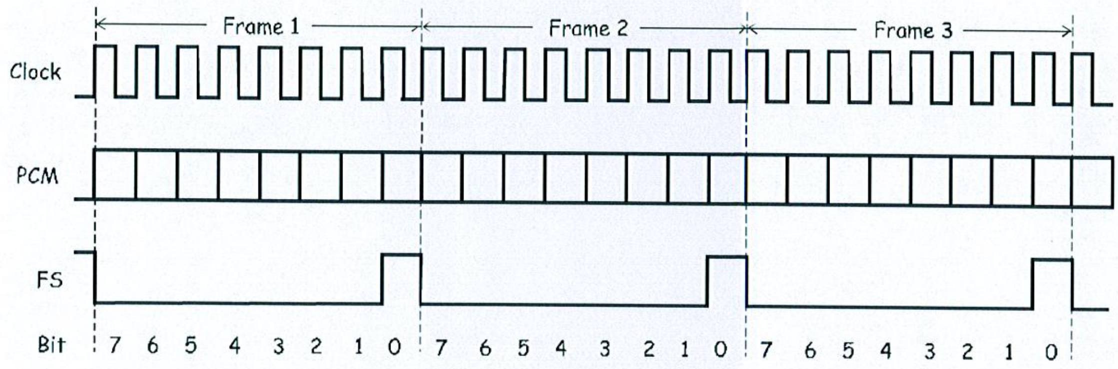

Figure 1

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.

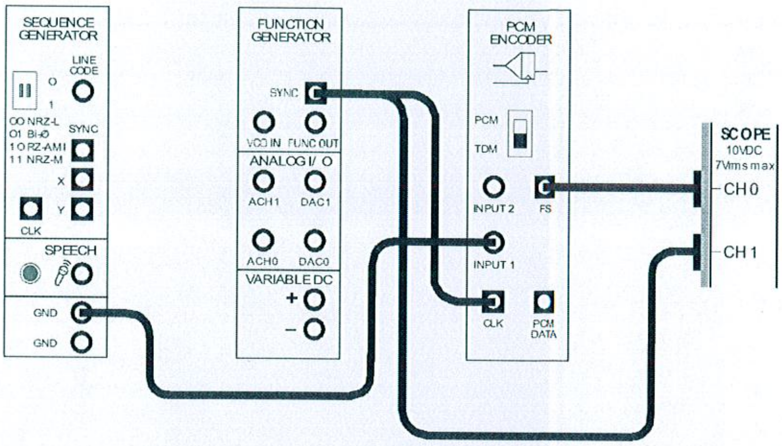

Figure 2

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.

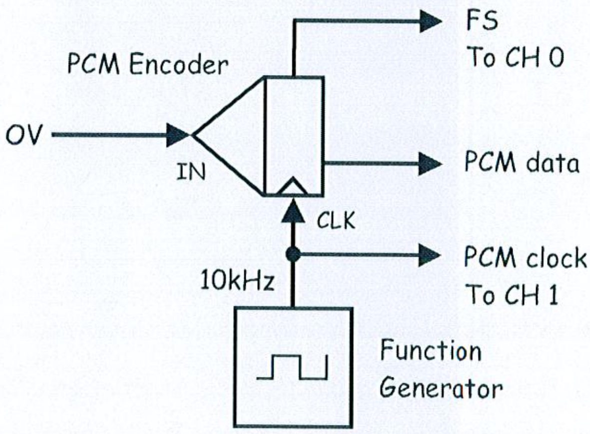

Figure 3

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.

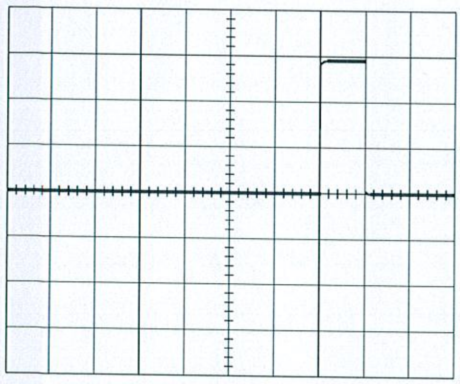

Figure 4

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.

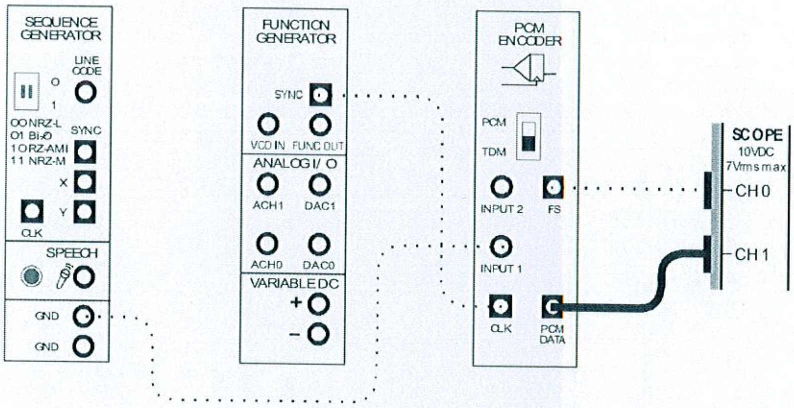

Figure 5

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.

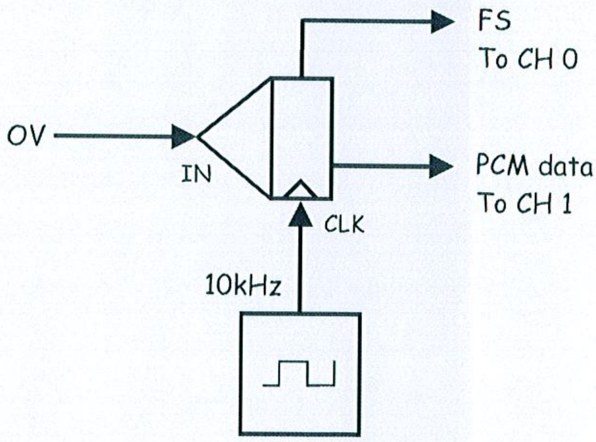

Figure 6

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.

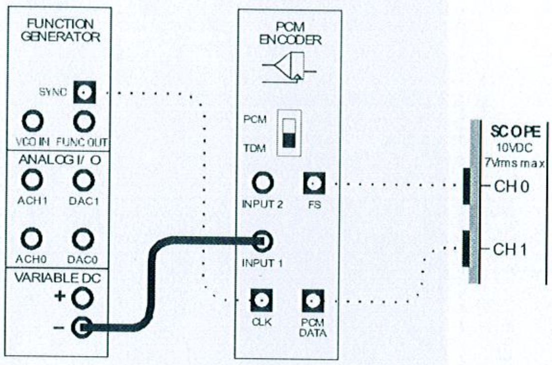

Figure 7

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.

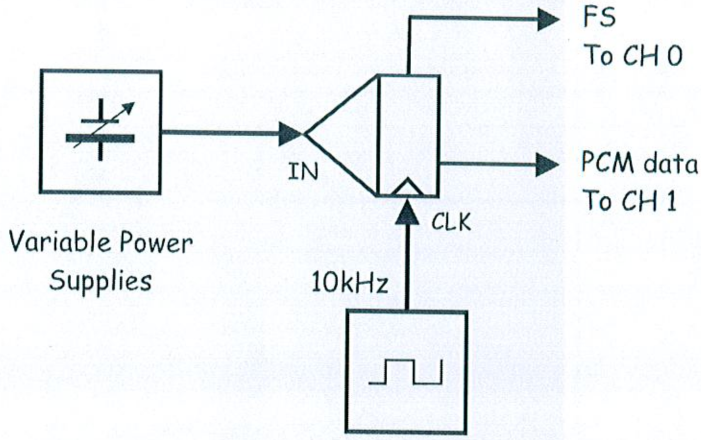

Figure 8

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.

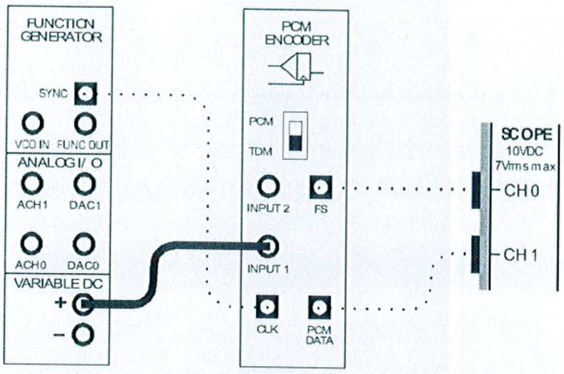

Figure 9

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

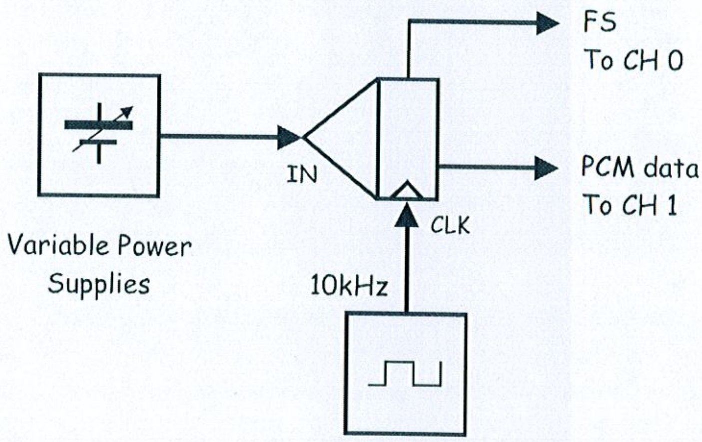

Figure 10

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.

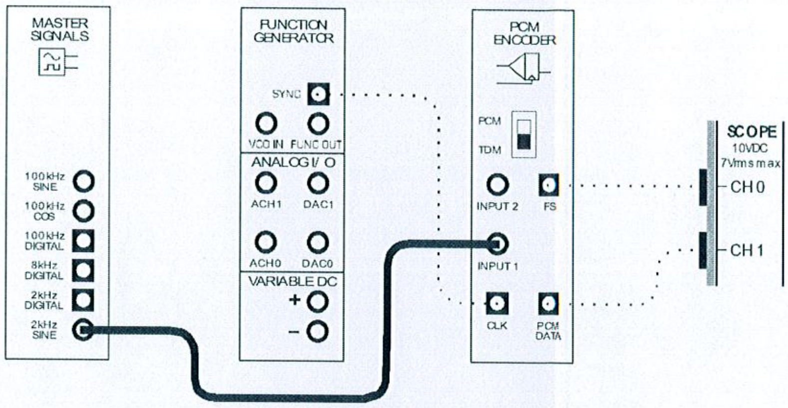

Figure 11

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.

