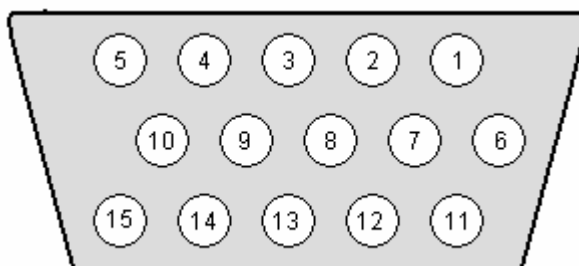


Monitor signals

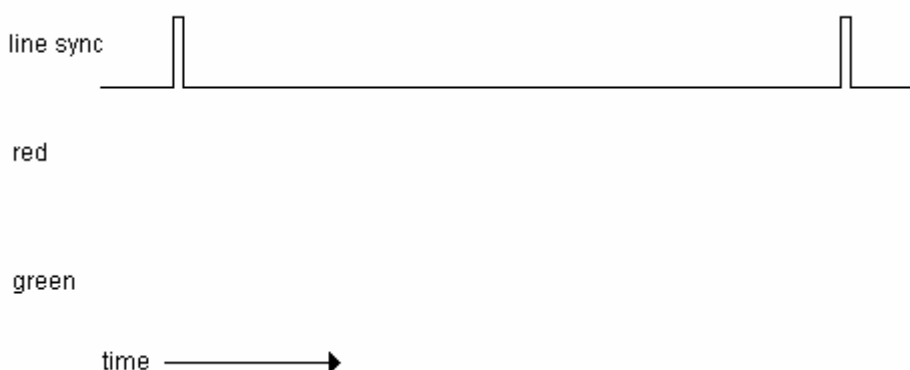
You are going to investigate the signals which transfer picture information between a laptop computer and an external monitor.

- 1 Locate the DE-15 socket at the back of the laptop which feeds signals to external monitors. Each signal goes out on one socket and has a ground (0 V) return on another, as shown in the table below. Each socket is a snug fit for standard breadboard wire. You will need to ensure that the laptop feeds signals to an external monitor as well as its own screen.

signal	out	return
red video	1	6
green video	2	7
blue video	3	8
line sync	13	5
frame sync	14	10



- 2 Use an oscilloscope to study the frame sync out signal. Earthing the frame sync return signal at the oscilloscope should make the trace clearer. Carefully measure the time period of the frame sync pulses. Calculate the frame refresh rate.
- 3 Measure the time period of the line sync pulses. Calculate the number of lines per frame.
- 4 Use a graphics package, such as PAINT, to cover the **whole** laptop screen with a few alternating vertical strips of red interleaved with green. Complete this timing diagram to show what you expect the red and green video signals to look like between line sync pulses.



- 5 Use the oscilloscope to look at both the red video signal and the line sync channel. Trigger on the line sync signal. Verify that the red video signal has the expected shape.
- 6 Repeat step 5 for the green video signal.

7 Now use the graphics package to cover the whole screen with vertical strips of red of different intensity. Start from black on the left increasing in steps to bright red on the right.



8 Use the oscilloscope to complete this timing diagram.



9 Use the graphics package to cover the whole screen with eight horizontal stripes, each with the different proportions of red, green and blue shown in the table.

stripe	red	green	blue	colour
one	000	000	000	black
two	000	000	255	blue
three	000	255	000	green
four	000	255	255	cyan
five	255	000	000	red
six	255	000	255	magenta
seven	255	255	000	yellow
eight	255	255	255	white

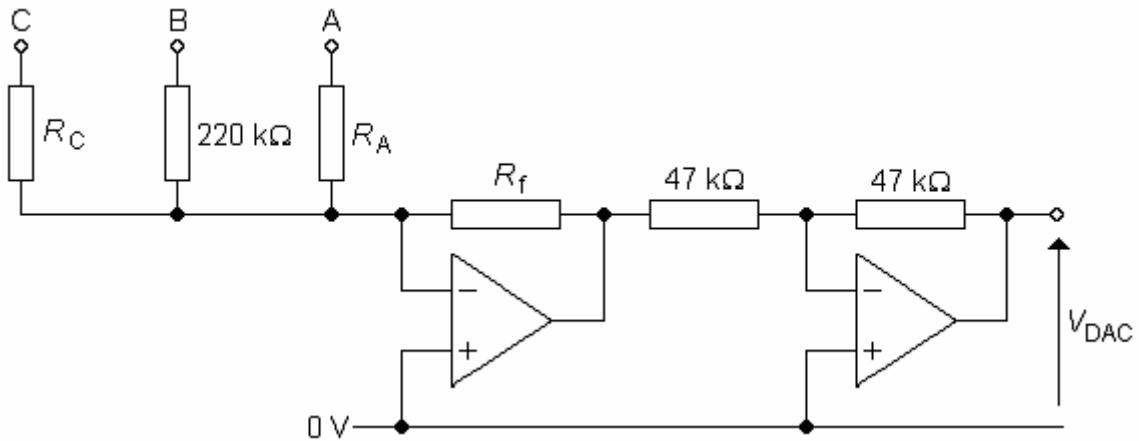
10 Complete this timing diagram to show what you expect the video signals to look like.



11 Trigger the oscilloscope on the frame sync signal. Study each of the three video signals and verify that they behave as you predicted.

DAC design

You are going to design and test digital-to-analogue converters based on summing amplifiers.



- 1 The DAC shown above is required to have eight different levels with a range from 0.0 V to +1.4 V. Calculate suitable values for R_A , R_C and R_f . Choose resistors from this list.

10 kΩ 22 kΩ 47 kΩ 100 kΩ 220 kΩ 470 kΩ 1 MΩ

- 2 Assemble the circuit. Use a DIL switch to connect C, B and A to the 5 V rail. Use a voltmeter to complete the table.

CBA	000	001	010	011	100	101	110	111
V_{DAC} / V	0.00 V							

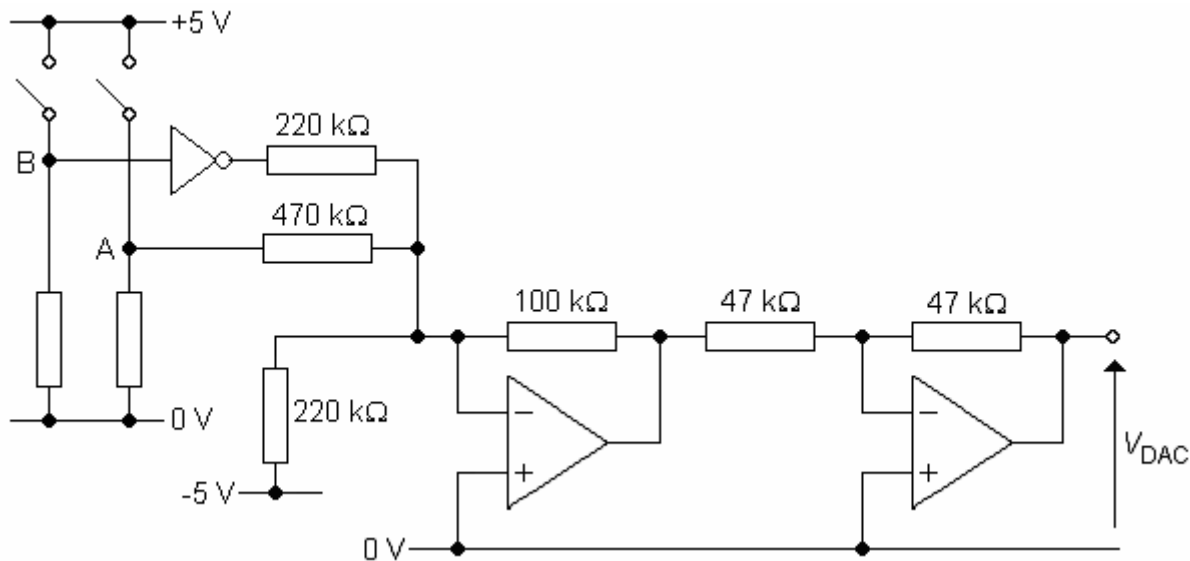
- 3 Add another resistor to improve the resolution to 0.1 V without changing the range. Choose resistor values from the list above.

- 4 Complete the table to test the four-bit DAC you have just assembled.

DCBA	0000	0001	0010	0011	0100	0101	0110	0111
V_{DAC} / V	0.00 V							

DCBA	1000	1001	1010	1011	1100	1101	1110	1111
V_{DAC} / V								

5 Do calculations with the formula $- \frac{V_{out}}{R_f} = \frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3}$ to complete the table for this DAC.



BA	V_{DAC}
01	
00	
11	
10	

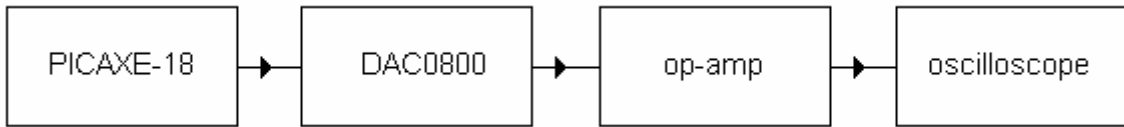
6 Assemble the DAC and test it with a voltmeter.

7 Adapt the DAC above so that it obeys this table. Test it with a voltmeter.

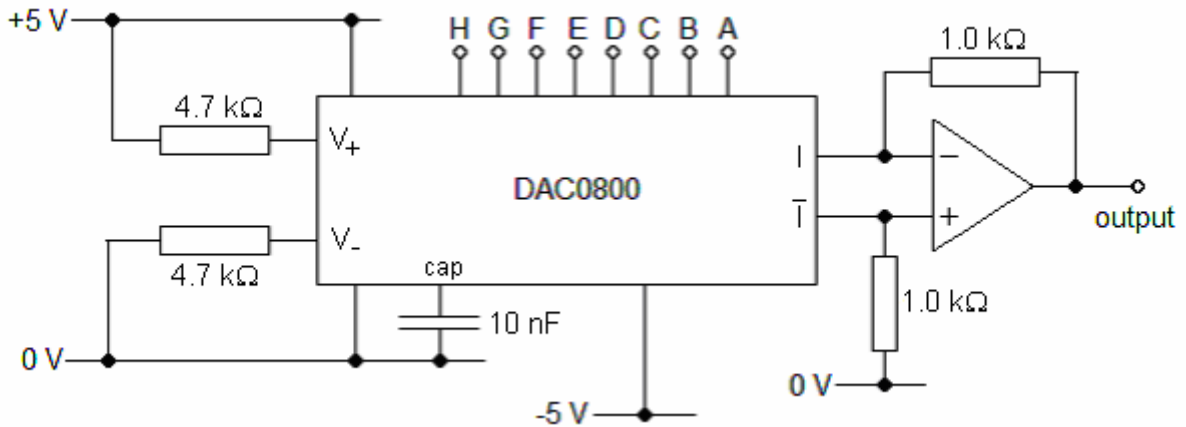
BA	V_{DAC}
011	+1.5 V
010	+1.0 V
001	+0.5 V
000	0.0 V
111	-0.5 V
110	-1.0 V
101	-1.5 V
100	-2.0 V

Digital waveforms

You are going to combine a DAC with a PIC to generate some waveforms. The block diagram is shown below.



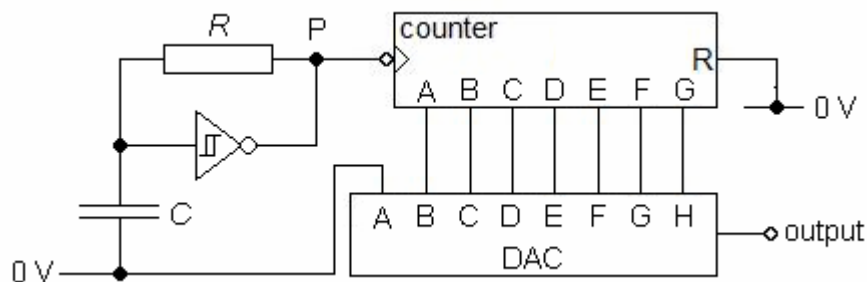
- 1 Start off by assembling the DAC and op-amp as shown below.



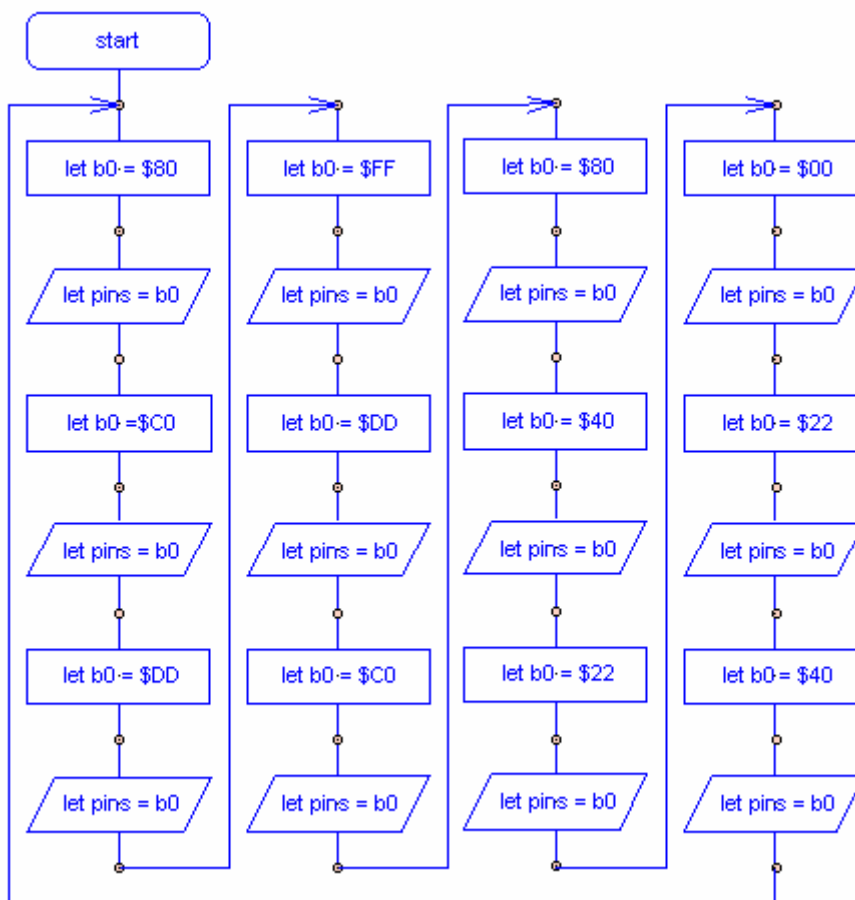
- 2 Connect a voltmeter to the output. Connect all of the digital inputs to 0 V. By connecting selected inputs to +5 V, verify that the output obeys this table.

HGFE DCBA	output / V
0000 0000	-1.0
0100 0000	-0.5
1000 0000	0.0
1100 0000	+0.5
1111 1111	+1.0

- 3 Add a relaxation oscillator and 7-bit counter as shown below. Select values of *R* and *C* which make *P* oscillate with a frequency of 5 kHz. Look at the output of the DAC with an oscilloscope. The voltage should rise steadily in small steps from -1 V to +1 V before suddenly dropping back down to -1 V and rising steadily once more.



- 4 Now write this program into a PICAXE-18 PIC. It continuously feeds out a series of 12 bytes, one after the other.



- 5 Use the PIC to generate the byte at the input of the DAC. Use an oscilloscope to verify that the DAC output is a digital approximation of a sine waveform.
- 6 Adapt the PIC program so that the DAC output produces the waveforms shown below.

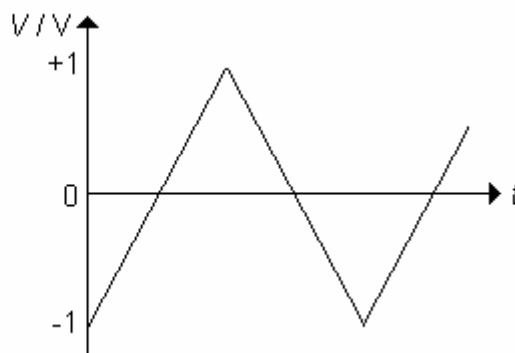
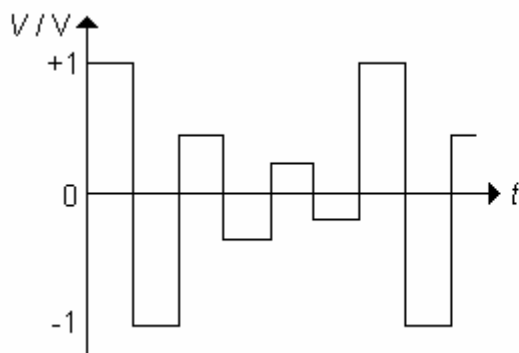


Image compression

You are going to investigate the effect of different compression techniques on an image.

- 1 Obtain a colour digital photograph. View the photograph in Paint.
- 2 Select part of the image about 12 cm high and 15 cm across. Choose part of the image with lots of detail and different colours. Copy it to a suitable location, calling it **image0**.
- 3 Open **image0** in Paint.
From **View/Zoom/Custom** select **800%** so that you can see the clusters of pixels.
- 4 From **Colors/Edit colors** select **Define Custom Colors**. This allows you to explore the effect of altering the level of intensity of the separate pixels in a cluster.
- 5 Click on each of the **basic colors** given in the table. If in doubt, select the brightest colour that you can find. Enter the levels of the pixels of the cluster.

basic color	red pixel	blue pixel	green pixel
white			
red			
green			
blue			
black			

- 6 From **View/Zoom/Custom** select **100%** so that you can see the whole image.
From **View** tick the **Status Bar**. Verify that this allows you to see the cursor coordinates in a window at the bottom right-hand of the screen.
- 7 Click on the **select tool**. Determine the number of clusters in the image, as follows:
 - place the cursor to verify that the cluster at the top left-hand corner has coordinates 0,0
 - find the coordinates of the cluster at the bottom right-hand corner
 - add one to the bottom right-hand coordinates
 - multiply them together

You should find that have about 300 000 clusters in your image.

8 Complete the calculations to determine the number of bytes required to store the image.

Each pixel has 256 possible levels of intensity, so needs _____ bits.

Each cluster has three pixels, so needs _____ \times _____ = _____ bits.

Each frame has _____ clusters, so needs _____ \times _____ = _____ bits.

So the saved file = _____ / 8 = _____ bytes or _____ / 1024 = _____ KB.

9 From **File/Save As** save image0 in the 24-bit Bitmap format. This is the uncompressed image. View the saved file details to find out its actual size in KB, where 1 KB = 1024 bytes. It should be in very close agreement to the result of the calculations in step 8.

10 Complete the first row and third column of the table.

file name	file format	clusters	bits per pixel	bits per frame	compression
image0	24-bit bitmap		256		1.00
image1	256 color bitmap				0.33
image2	16 color bitmap				
image3	monochrome				
image4	GIFF				
image5	JPEG				

11 Complete the rest of the table as follows:

- open up **image0**, the uncompressed image
- save it as **imageN** in the specified format, compressing the image
- view the effects of the compression
- use the file details to find the **bits per frame** (1KB = 1024 bits)
- calculate the effective **bits per pixel**
- calculate the **compression**, how much smaller each file is than the original

Of course, the number of clusters is the same for all file formats.

12 You might expect that the bits per pixel for the monochrome format should be 1.0.

In practice it is slightly larger. The file clearly holds more information than just the number of bits per pixel. Suggest a reason for this. Support your answer with calculations of the extra information required.