

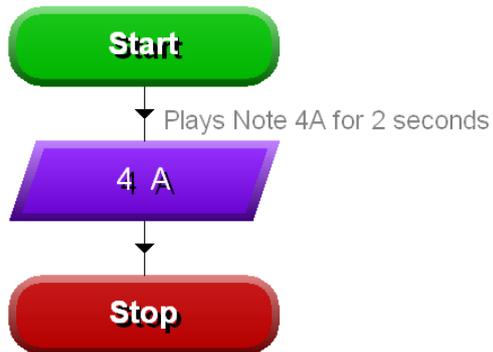
Using a **PIC Microchip** in Project Design



Presenters: *Martin Cassidy*
Pat Lynch

Before using this package ensure that you have selected the correct software. The following courseware assumes the use of GENIE DESIGN STUDIO. Go to Program, Program Settings and choose the GENIE C08 project board.

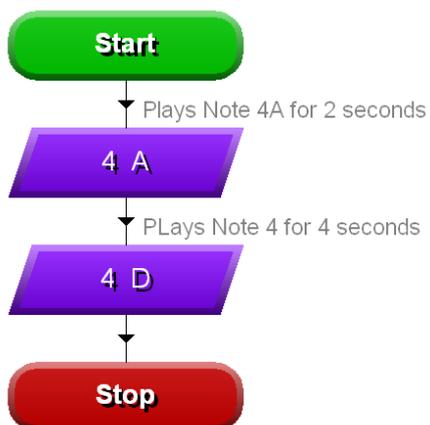
PROG.1 – ‘SOUND’ command



Drag each command from the right hand side of your screen. Click and hold down the left mouse button to drag in each command. The second programming block makes use of the sound command. Double click the left button on the sound command. This will open a window and allow you to select a sound value. Scroll or type to enter the selected value, set G4 as output and set the note to 4A. You can set the duration of the note by placing a value in the time box i.e. 2s.

Add a STOP command and select ‘Run Live’ to download your programme to the PIC.

PROG.2

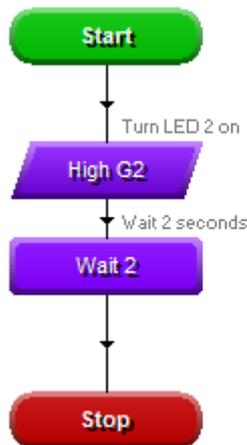


Add a second sound command. Select a suitable note and time value.

Make sure the programming blocks are connected.

Download and test run your programme.

PROG.3 – 'OUTPUT' & 'WAIT' command



Outputs are used to turn LEDs, light bulbs or other outputs on and off. Note that the PIC cannot switch bulbs directly but must use a driver circuit (in most cases – a transistor)

WAIT is used to set the length of time the lamp / LED is turned on. WAIT is set using seconds or fractions of a second e.g. 2 seconds in this case.

'LED ON' is called a label. Note, however, it is setting the output to a 1 which causes the LED to turn on – NOT the label. Why does the LED not turn off when you run the above flowchart? How would you correct this?

PROG.4

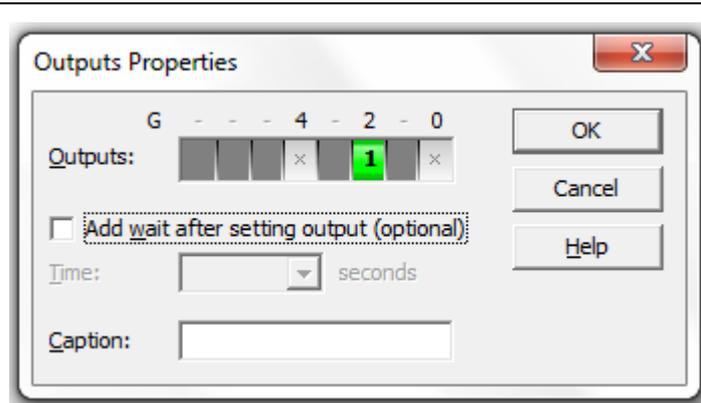


It is possible to turn on a number of outputs at the same time by setting the correct 'bits'. Each bit controls one output. In this case two LEDs are driven from output G0 and output G2.

Text boxes may be added to help understand what should be happening in the programme.

Note that in the OUTPUT command box both LEDs will light (High). Wait for 3 seconds is set this time using the wait option available in the OUTPUT command. The program then moves to the SOUND block. The LEDs do not turn off at this point but remain on. If we wish the LEDs to be turned off then we must add a second OUTPUT command and set both outputs to 0. The STOP command is added to end the programme.

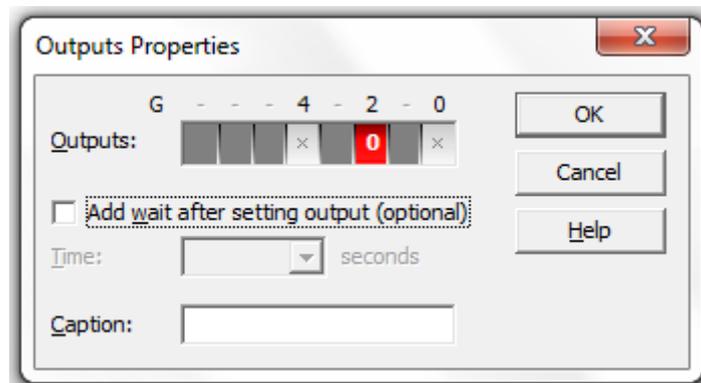
The 'OUTPUT' command – Bit Configuration



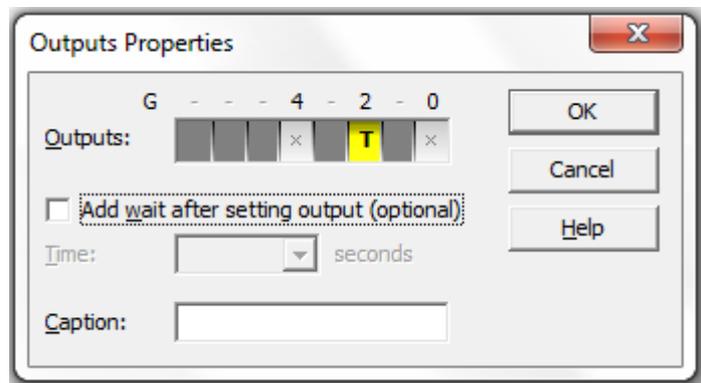
There are four settings that can be used when setting outputs.

(a) Let us look at output 2.

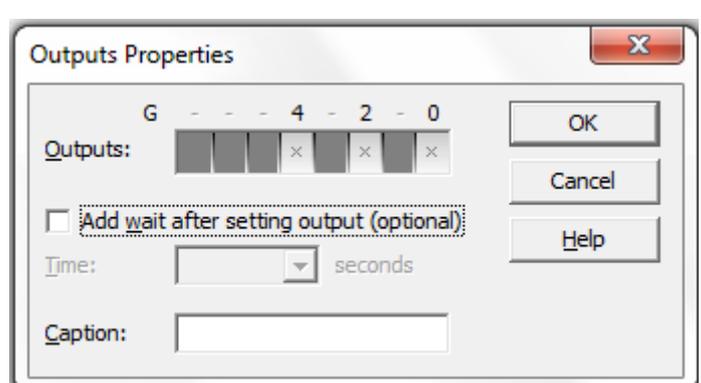
In this case it has been set to 1 or ON.



(b) Output 2 has now been set to 0 which means that output 2 is switched OFF.

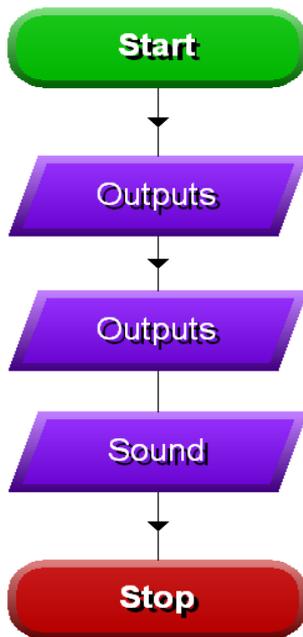


(c) Setting output 2 to T will toggle the previous setting i.e. 1 is set to 0, 0 is set to 1.



(d) Leaving outputs set to X will cause the programme to ignore all outputs.

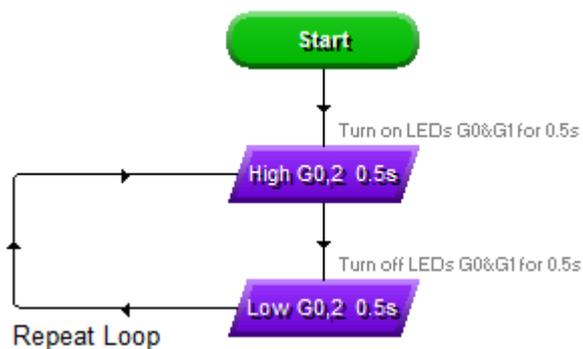
PROG.5 - Exercise



Write a program to turn on outputs G0 & G2 to which LEDs are connected for 3 seconds. Then turn off both outputs and wait for 1 second before playing the note 6G for 0.5s. Use the flowchart shown as a guide. Add text boxes to explain your program.

Remember we must tell the program which pin is connected to the speaker i.e. G4.

PROG. 6 – ‘LOOP’ command



The program shown will cause two LEDs to flash on and off forever. This is achieved by using a repeat loop. This is really useful as many procedures in real life run continually. These include automated traffic light sequences which run forever.

The advantages of using the PIC should start becoming obvious. If we need to change the programme or part of the programme sequence (e.g. change the wait to 1 second) we simply rewrite the software, not remake the circuit.

PROG.10 – decision command using a DIGITAL programme block

In real life as well as processes which run continually there are times when user input is necessary. Nowadays however there is a tendency to automate the process in response to changes in light, temperature, liquid level or other stimuli. So far we have used the PIC to drive outputs such as LEDs and a speaker. The real value of control technology is when outputs are driven or controlled by the input conditions.

The GENIE C08 project board has two inputs, G1 & G3. G1 (A/D) can be an analogue or digital input, while G3 (DIG) is digital. Let us begin by designing a program to include a switch (Input G3) as a digital input.

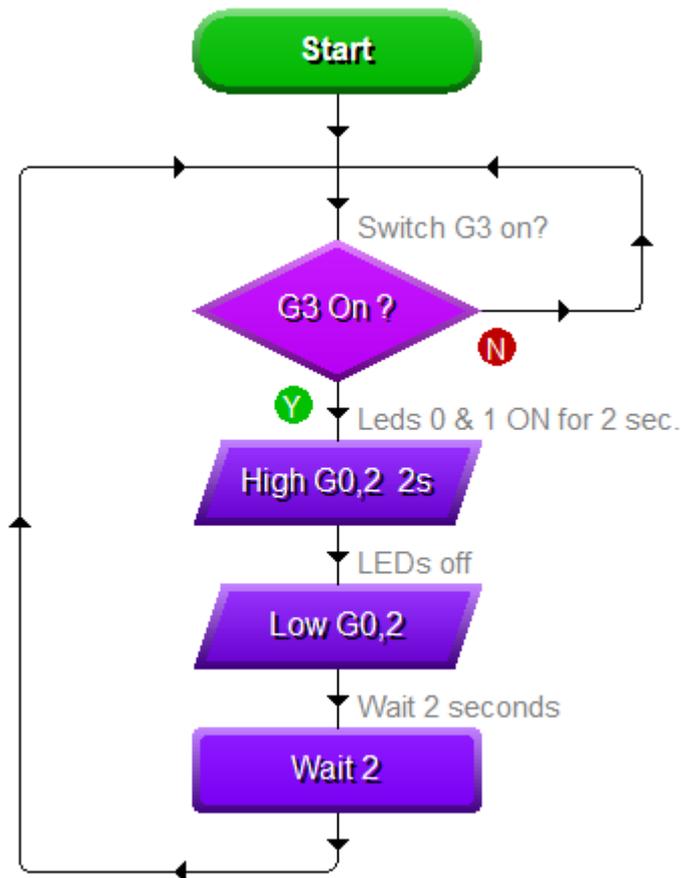
'DIGITAL' block



How do I set an input? Well this is quite easy really, once you begin to understand decision boxes. By dragging in a Digital Box and clicking on it, the window shown opens. You can

see there are only two inputs available and that the numbers correspond with the inputs identified above. One thing you must understand is that when switch G3 is off, the input at G3 is represented by a 0. When the switch G3 is on, the input at G3 is represented by a 1.

By using this syntax we are really asking a question and as a result making a decision. Has the switch which is connected to input G3 been pressed? If the answer is yes carry out an instruction, if not, carry out a different instruction. Please note that it is *not the label* which determines the decision but rather which inputs are set to 1 or 0.



Build the flowchart show. Be careful to use a digital box and not a compare box which performs a different function. Remember to set input G3 to a 1 as shown previous. When you start the program nothing happens until the switch (Input G3) is pressed. When the switch is off, the programme will continually check the status of the switch to determine if it is on (1).

If the switch is on, the input at G3 will be set to 1, and the YES route will be followed. This means that the LEDs will be turned on for two seconds and the programme continues.

Please note.

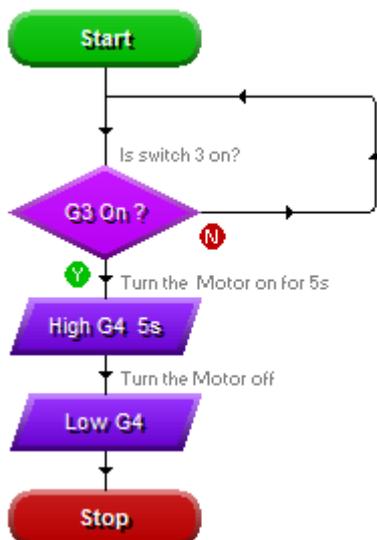
Understanding the operation of a digital box is vital to maximizing the use of control technology. Do not progress until you have a full understanding of the above program.

PROG.11 – Motor control using output G4

To control a motor on the GENIE C08 board, we connect it to output G4. A transistor allows output G4 to control more powerful output devices.

PICTURE OF PIC BOARD

The program below shows how we use a push to make switch, connected to input G3, to turn on a motor which is connected to output G4, for 5 seconds. Remember, when the switch is off the input is 0 but 1 when on. Input G3 therefore, is set to look for when the switch is pressed i.e. a 1.



Two further points to note:

The label used this time is ‘Switch G3 on?’ Remember the label is only a description; it is the inputs which are set to check for 0 or 1.

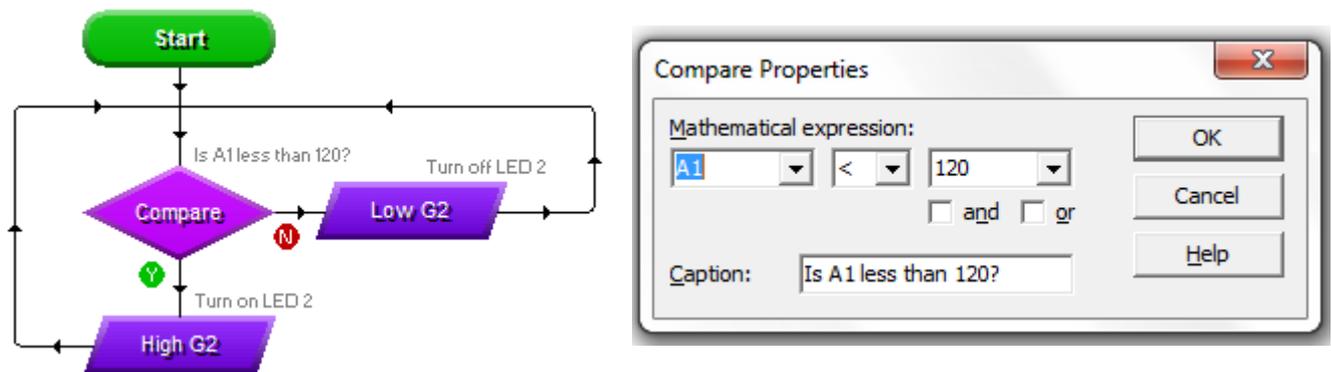
What would be the result if the second output command was not included in the flowchart?

What changes are required in the programme to allow the system run continuously?

PROG.24 – Analogue Sensors

Temperature and light are examples of analogue conditions. As mentioned earlier, the GENIE C08 board can accommodate one analogue sensor at input A/G1. (This input can actually accept an analogue or digital signal.)

Let us begin by considering the LDR connected to input A/G1. Assume we wish to turn on a security light if the light intensity drops below a particular level. For this exercise we will use the 'Compare' command to detect light levels. The 'Analogue' command could also be used.



In the program above a compare command is used to determine if the light level is less than 120, this is an arbitrary value. If the light level is greater, then the LED is off. If the reading is less than 120 then the LED is switched on. You will need to determine a value which will provide the correct response for you. Use the 'Celebrate Sensor' option to achieve this value.

Now edit your flowchart to turn on a motor if it gets dark.

Debug Live Command:

Use of the 'Debug Live' command allows the changing values to be viewed live on screen. This is extremely useful when projects require the calibration of their analogue sensors.

Hand drier challenge:

Design a flowchart to turn on a motor if a light beam which is shining on an LDR is broken.