



Welcome to the GENIE microcontroller system!

The GENIE Audio Kit is an intelligent, programmable electronic circuit that can play real (16 channel) MIDI music and life-like sound effects through its loudspeaker.



A GENIE Audio Kit (shown on its own, without a product design).



2 in 1

To showcase the GENIE Audio Kit, we have used it to produce two signature projects: **GENIE Cuddly Creatures** (left) and **GENIE Rock Star** (below).

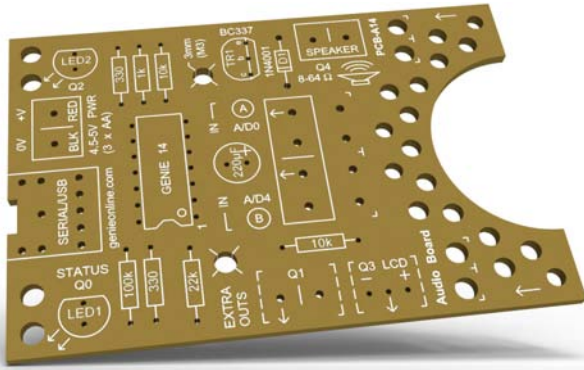
These projects use CAD/CAM to combine the GENIE Audio Kit with a range of inventive ideas incorporating sound and music.

You can find out about these projects in their respective PDFs.





Making the GENIE 2



◀ The GENIE Audio Kit consists of a **printed circuit board (PCB)**, shown on the left) onto which are placed a number of electronic components.

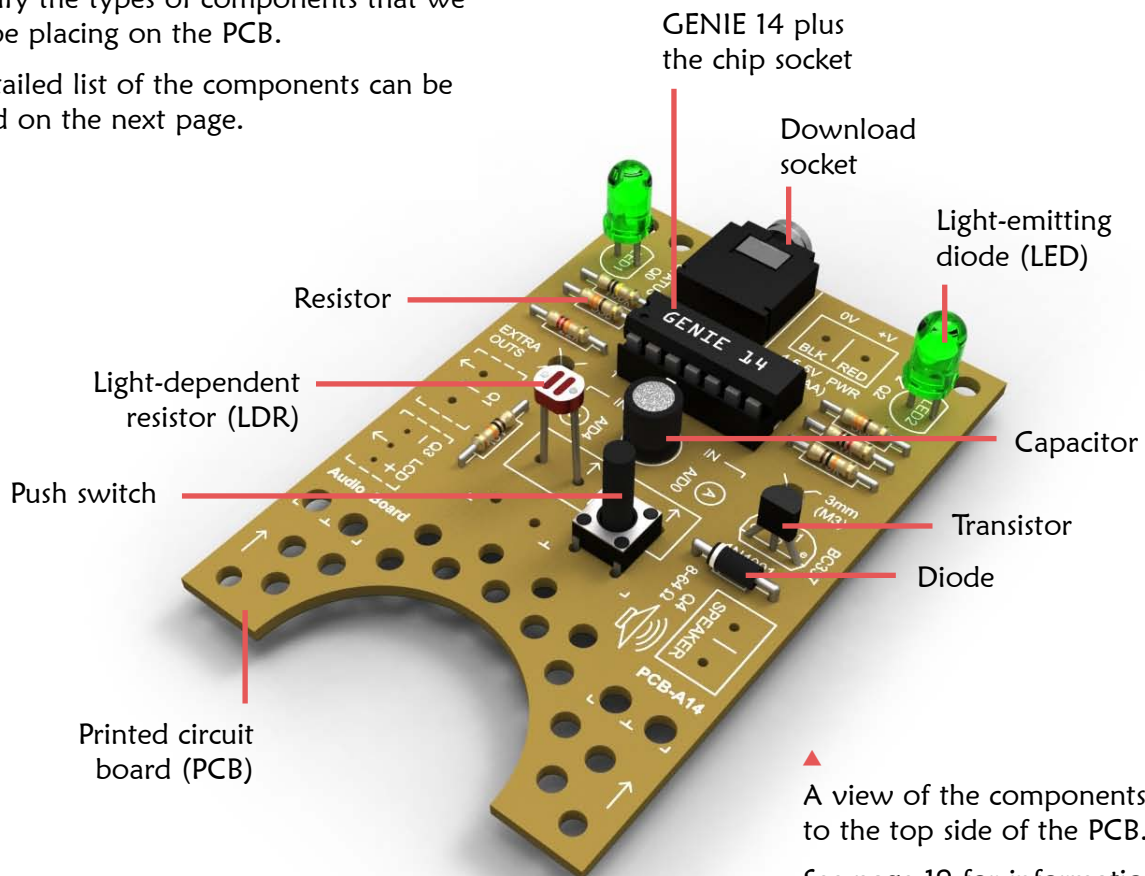
It is these components that provide the kit with the ability to play music.

The components must be placed in specific positions on the PCB. These positions are indicated by a series of markings and labels on the upper-side of the PCB. These markings are known as the **silk screen**.

Once placed, components must then be soldered to the under-side of the PCB so that they make an electrical contact.

The picture below will allow you to identify the types of components that we will be placing on the PCB.

A detailed list of the components can be found on the next page.



▲ A view of the components mounted to the top side of the PCB.

See page 19 for information on the Q1 and Q3 outputs.

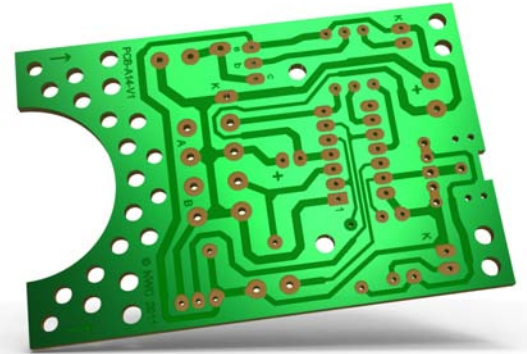




Making the GENIE 3

A GENIE Audio Kit has the following component parts:

Qty.	Component description
1	GENIE Audio printed circuit board (PCB)
1	GENIE 14 (14-pin) integrated circuit (IC)
1	3xAA battery pack
1	Battery clip
1	50mm loudspeaker
1	14-pin chip socket
1	Download socket
2	330 ohm (330 Ω) resistor
1	1,000 Ohm (1k Ω) resistor
2	10,000 Ohm (10k Ω) resistor
1	22,000 Ohm (22k Ω) resistor
1	100,000 Ohm (100k Ω) resistor
1	220 micro-Farad (220 μ F) electrolytic capacitor
1	1N4001 diode
1	BC337 transistor
1	Push switch
1	Light-dependent resistor (LDR)
2	Green light-emitting diodes (LEDs)



A view of the underside of the printed circuit board. The component legs will be soldered to this side of the board.

Most of the components are easy to identify (if you need guidance, see the picture on the previous page). The seven resistors however are more difficult. These need to be identified by their different colour bands.



The resistor values and colour bands are:

- 300 Ω Orange, orange, brown and gold
- 1k Ω Brown, black, red and gold
- 10k Ω Brown, black, orange and gold
- 22k Ω Red, red, orange and gold
- 100k Ω Brown, black, yellow and gold

Ω is the symbol for **Ohm**, which is a measure of how much it resists the flow of electricity. A higher value will result in a smaller electrical flow.





Making the GENIE 4

The following instructions describe how to solder the different components to the printed circuit board.

First, switch on the soldering iron. It will only take a few minutes for the iron to reach operating temperature. Once the soldering iron is hot, carefully clean the soldering iron tip with a moist (but not too wet) sponge.

Melt some solder at the chamfered end of the soldering iron tip. This is called 'tinning' and it will aid the flow of solder from the soldering iron to the copper track on the printed circuit board and component pins.



The LDR and LEDs are shown mounted above the base of the printed circuit board. This is optional for single-panel designs, but is recommended for double-panel designs (so that the components are not obscured by the upper panel).

Next, solder each component onto the board. We recommend that you fit them in this order:

1. Resistors
2. Diode
3. Download socket
4. 14-pin chip socket (not the GENIE IC)
5. Transistor
6. Capacitor
7. Light-emitting diodes (LEDs)
8. Light-dependent resistor (LDR)
9. Push switch

Use the picture on page 2 along with the PCB's silk screen as guides when placing the different components. See on the right for further help.

When fitting components such as resistors, we recommend that you use long-nosed pliers to bend the legs through 90 degrees. This will make the components easier to place.

Some of the components need to be fitted the correct way around:

- ◆ The diode should be placed so that the stripe on the diode matches the stripe on the board.
- ◆ The 14-pin chip socket should be positioned so that the notch matches the notch shown on the printed circuit board's silk screen.
- ◆ The flat side of the transistor must match the flat side shown on the board.
- ◆ When fitting the electrolytic capacitor, you need to ensure that the positive side of the capacitor (the side without the stripe) is nearest to the '+' sign on the board.
- ◆ The LEDs should be fitted so that the flat edges on the base of the LEDs line up with the flat edges shown on the board.
- ◆ So that the LDR and LEDs are visible when used with a double-panel design, we suggest sitting the bottom of those components above the printed circuit board (shown left).

To solder a pin, hold the soldering iron onto the board for a few seconds, then quickly touch the tip with a small amount of solder.

You should always remember to replace the soldering iron back into the stand after soldering and repeat cleaning the tip of the iron with the moist sponge before the start of each soldering operation.

Finally, cut off any excess wire or component legs for a tidy finish.





Making the GENIE 5

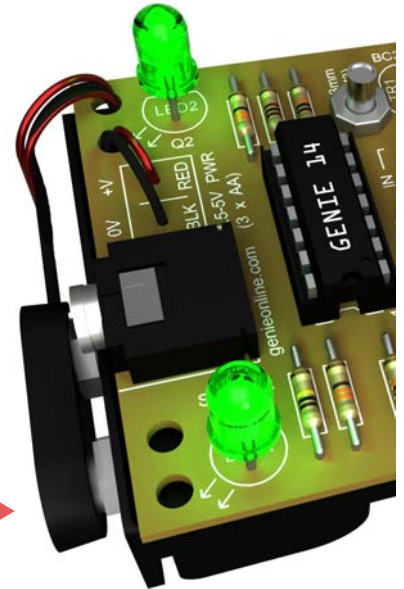
With the components mounted on the printed circuit board, the power and loudspeaker connections can then be soldered.

To wire up the power connections, you need to solder the red (positive) and black (negative) wires coming from the battery clip to the '+V' and '0V' holes on the PCB, pictured on the right.

Before soldering, we recommend looping the wires from the battery clip through the two nearby holes as shown. This acts as a strain relief, helping to protect the wires from damage.



◀ Loudspeaker wired to the board.



▶ Power connections wired to the board.

For the loudspeaker, two wires will need to be soldered to the 'SPEAKER' holes on the PCB, as shown above. The other end of these wires will be soldered to the two solder tabs on the loudspeaker itself.

For neatness, you can run the wires through one or more of the sound holes. A wire length of about 4 to 5 cm is normally sufficient. Remember to strip the wires at both ends before soldering.

The final step required to complete the building of the GENIE Audio Kit is to fit the GENIE microcontroller into the chip socket.

As with the chip socket itself, the GENIE microcontroller must be fitted the correct way around. You will notice that the chip has both a notch and a 'dot' at one end of the chip. These identify which of the chip's pins is number 1. Insert the chip so that these match the notch and '1' shown on the printed circuit board.

Care should be taken not to damage the chip's legs, but you may need to bend them gently as you insert the chip. The chip should be seated flat against the top of the socket.

▶ The GENIE chip inserted into the 14-pin chip socket. The pin 1 'dot' is highlighted.





Making the GENIE 6

A power cell.

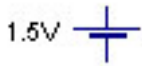


The positive end of an AA cell is usually marked by a '+' sign.

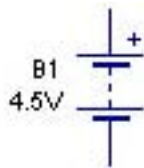
These AA power cells typically produce an electrical potential of about 1.5 volts each.

For the GENIE Audio Kit we will need three of these cells, giving a total of 4.5 volts.

On a circuit diagram this would be represented by the following circuit symbols:



Circuit symbol for an individual power cell.



Circuit symbol for a battery of cells.

With the electronic circuit built, the three cells should then be inserted into the battery box.

As with all power cells, it matters which way around the AA cell is inserted: there is a **positive** end and a **negative** end. The positive end is usually marked by a '+' sign.

The cells are placed in **series**, meaning that the negative end of the first cell connects to the positive end of the second, and so on.

The two pictures on the right show how to insert the AA cells into the battery box.

You should now be ready to switch on your circuit. However, for any electrical or electronic circuit to work there first needs to be a source of **power**.

Like most electronic toys, the GENIE Audio Kit gets its power from standard AA power cells, shown on the left. These cells are measured by their **voltage** (voltage is a measure of the cell's electrical potential).



Spring

An empty battery box before the cells have been inserted. The negative end of the power cell should be against the spring. Notice that the middle spring is at the other end, which means that the cell must also go in the other way around:



Battery box showing all three cells inserted and the battery clip connected.



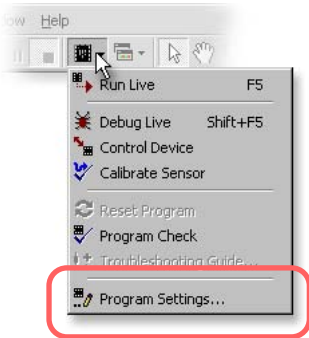


Telling the GENIE your wishes

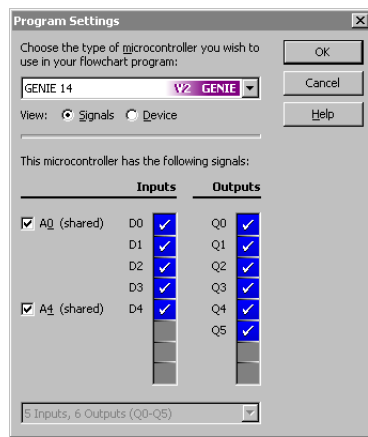


For your project to work, you need to tell the GENIE microcontroller what it should do.

This involves writing a sequence of commands in a **flowchart**. Your flowchart is then sent down the cable and stored on the GENIE chip. By changing the flowchart, you can vary how the GENIE behaves.



First of all, you need to tell GENIE which type of chip you are using. To do this, click on the **Microcontroller** button on the toolbar and choose **Program Settings**.



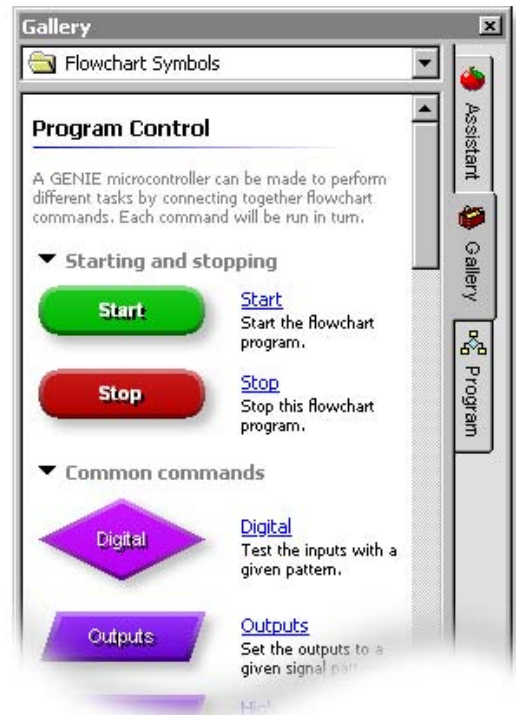
Select the v2 **GENIE 14** chip.

The inputs and output signals for this type of microcontroller are fixed, so click on **OK** when you are ready to continue.

Available Signals

These are the **input and output signals** available in your flowchart:

Input	Description
A/D0	Analogue or digital
A/D4	Analogue or digital
Output	Description
Q0	Green LED
Q1	Extra output
Q2	Green LED
Q3	LCD output
Q4	Loudspeaker output



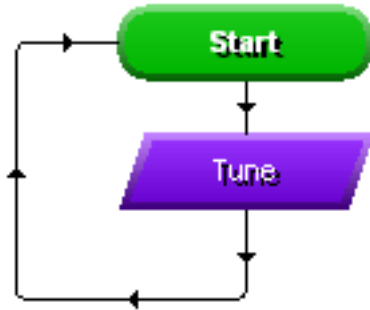
You can now decide which commands you want your GENIE to perform. To do this, drag commands from the **Gallery**.

See the next worksheet for flowchart ideas.





Programming the GENIE to play music 8



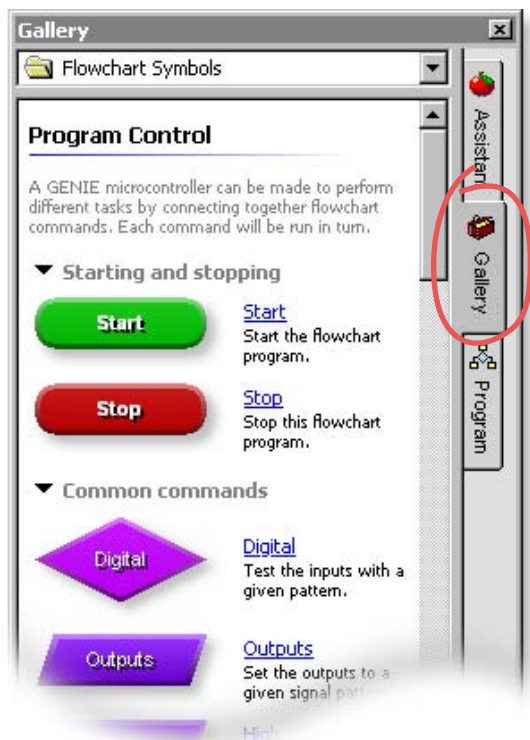
- ◀ The first flowchart we will look at is shown on the left. It plays a tune and then repeats. The LEDs will flash in time with the music. All GENIE flowcharts begin with a **Start** command. When running the flowchart, GENIE simply follows the arrows. Each command that GENIE encounters while following the arrows will be run. With most commands, including the two shown here, arrows go both into and out of the command.

When you create a new flowchart, a **Start** command will be placed automatically on the page. To add other commands, you should use the **Gallery**.

If the Gallery is not visible, click on the **Gallery** tab that appears to the right of application's window. From here you can drag commands onto the page.

For the flowchart above, we will want to locate the **Tune** command and drag it below the **Start** command. To locate it, you will need to scroll down through the list of commands.

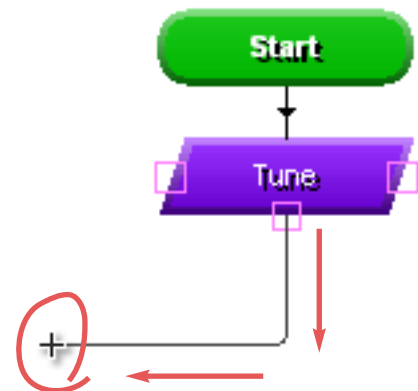
When you drop a command below (and close by) another one, a line and arrow will be placed automatically. To add the second line (the one looping back to the **Start** command), you need draw it using your mouse.



- ◀ Lines can be drawn from any one of the three squares shown on the **Tune** command. Move the mouse over the bottom square then click and release the left mouse button.

Now move the mouse down and to the left. A line will follow your mouse.

Click with the left mouse button to add a bend when you are not over anything. Keep moving the mouse upwards so that it is level with the **Start** command. Click again to add a second bend and then move the mouse right until the square appears. When you are over the square, press the left mouse button a final time to connect line to the **Start** command.





Choosing which music is to be played

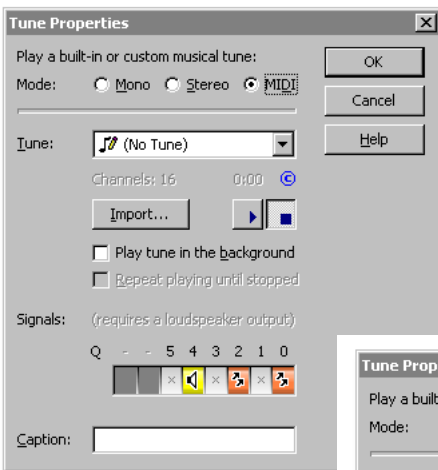
Once you have added a **Tune** command to your flowchart, you must next select the MIDI music that is to be played when the command is run (you can learn more about MIDI on the next page).

For those using Circuit Wizard 3, you can select MIDI tunes through the **Music Centre**.



Click on the **Tune** command that you wish to change (this is not needed if there is only one command).

Once done, open the **Music Centre** and move the mouse to the right-hand edge to reveal a list of available tunes.

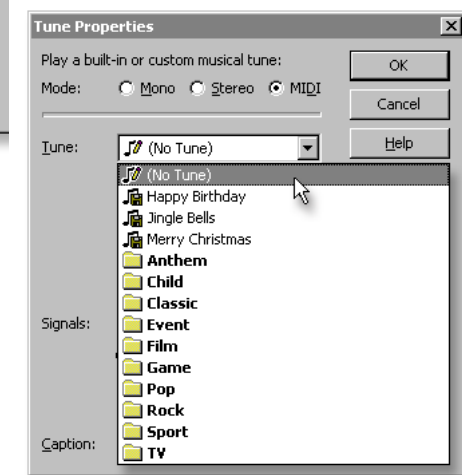


Click on a tune to select it (selecting one with a folder icon will bring up the list of tunes in that folder).



With the GENIE Programming Editor, you should instead double-click with the left mouse button on the **Tune** command.

Finally, click on the **Tune** drop-down box to reveal a list of available tunes. As before, selecting one with a folder icon will bring up the list of tunes in that folder.



First you must select the **MIDI** option.

Next, click on the **Signals** so that they match the ones shown on the far left. If the signals are currently empty, you should click on signal number **4** once, followed by signal **2** and then signal **0**.

These steps are not needed if you are modelling the GENIE Audio Kit within Circuit Wizard 3 because it can select the settings automatically, since it knows which printed circuit board you are using.

Because GENIE uses MIDI, you are not limited to the tunes that come with the software. With both Circuit Wizard 3 and GENIE Programming Editor, you can import MIDI tunes from other musical products, such as **Sibelius** or **Cubase**. To import, simply drag a MIDI file onto the **Music Centre** or **Tune** windows.





Understanding MIDI music and audio

10

The GENIE microcontroller at the heart of the GENIE Audio Kit stores tunes in a musical format known as **MIDI**.

MIDI is short for **Musical Instrument Digital Interface** and is a digital standard developed in the early 1980s, primarily to control keyboard synthesizers (though it later progressed into electronic drum sets and other musical instruments).

Unlike musical formats such as MP3 which store music as a series of waveforms or samples, MIDI instead works by storing numbers representing the actual musical notes being played (with 60 being middle C).

Because MIDI stores just the pitches and durations of each note, it takes up far less space, however it does mean that MIDI cannot record singing voices.

MIDI was used to play music in video games in the 1980s and 1990s and for ringtones on second-generation mobile telephones, from around 2002.

You can learn more about MIDI via the following Wikipedia article: <http://en.wikipedia.org/wiki/MIDI>

▼ A low-cost home MIDI keyboard.



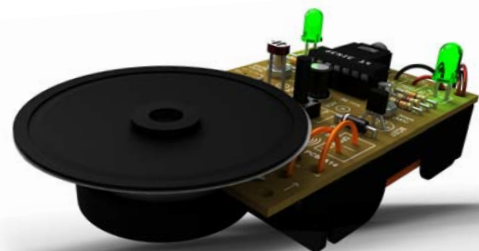
GENIE supports **16-channel** MIDI music, which means that up to 16 notes can be played back at the same time. This allows GENIE to play complex tunes involving chords.

The GENIE Audio Kit works by taking the notes within the MIDI file and then vibrating the **loudspeaker** cone at the correct frequency. These sound frequencies are measured in **Hertz (Hz)**, which is the number of waves per second. Humans can typically hear from about 10Hz to 10,000Hz.

The frequency determines the pitch of the sound. Higher frequencies result in higher (treble) sounds, whereas lower frequencies result in lower (bass) sounds.



▲ A MIDI synthesizer in a recording studio. Image courtesy of Wikipedia.

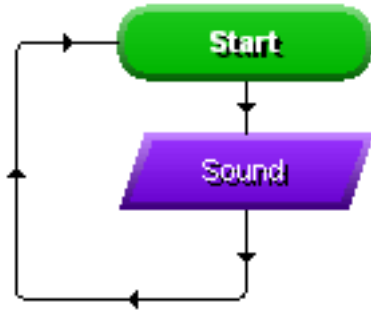


▲ The loudspeaker cone is controlled by a magnet (which is itself controlled by the GENIE microcontroller) that forces the cone to move. This movement generates compression waves in the air, which are eventually picked up by your ear, forming the sound you hear.





Programming the GENIE to play sound effects 11



◀ The first flowchart we will look at is shown on the left. It plays a sound and then repeats. The LEDs will flash along with the sound.

All GENIE flowcharts begin with a **Start** command. When running the flowchart, GENIE simply follows the arrows. Each command that GENIE encounters while following the arrows will be run.

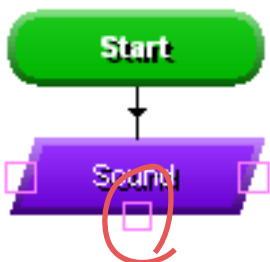
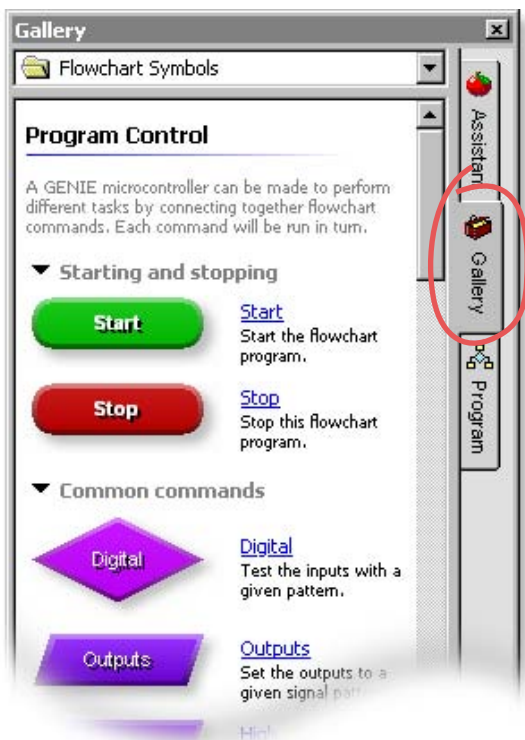
With most commands, including the two shown here, arrows go both into and out of the command.

When you create a new flowchart, a **Start** command will be placed automatically on the page. To add other commands, you should use the **Gallery**.

If the Gallery is not visible, click on the **Gallery** tab that appears to the right of application's window. From here you can drag commands onto the page.

For the flowchart above, we will want to locate the **Sound** command and drag it below the **Start** command. To locate it, you will need to scroll down through the list of commands.

When you drop a command below (and close by) another one, a line and arrow will be placed automatically. To add the second line (the one looping back to the **Start** command), you need draw it using your mouse.

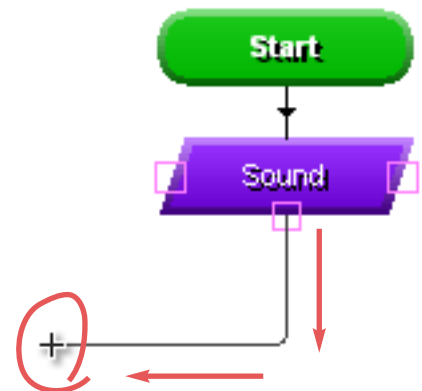


◀ Lines can be drawn from any one of the three squares shown on the **Sound** command.

Move the mouse over the bottom square then click and release the left mouse button.

Now move the mouse down and to the left. A line will follow your mouse.

Click with the left mouse button to add a bend when you are not over anything. Keep moving the mouse upwards so that it is level with the **Start** command. Click again to add a second bend and then move the mouse right until the square appears. When you are over the square, press the left mouse button a final time to connect line to the **Start** command.





Choosing which sound is to be played 12



Once you have added a **Sound** command to your flowchart, you must next select the sound effect that is to be played when the command is run (see the next page for more on sound effects).

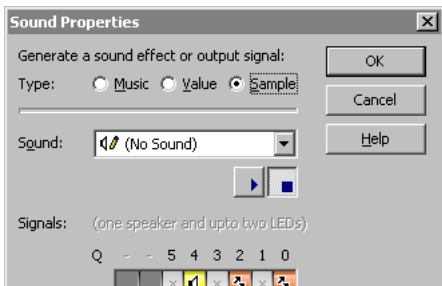
- ◀ For those using Circuit Wizard 3, you can select sound effects through the **Sound FX Centre**.

Click on the **Sound** command that you wish to change (this is not needed if there is only one command).



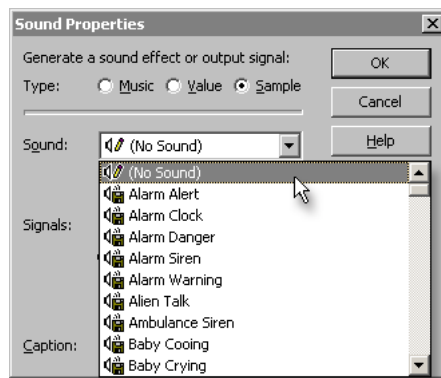
Once done, open the **Sound FX Centre** and move the mouse to the right-hand edge to reveal a list of available sounds, as shown on the left.

- ▶ Click on a sound to select it



- ◀ With the GENIE Programming Editor, you should instead double-click with the left mouse button on the **Sound** command.

First you must select the **Sample** option.



Next, click on the **Signals** so that they match the ones shown on the far left. If the signals are currently empty, you should click on signal number **4** once, followed by signal **2** and then signal **0**.

These steps are not needed if you are modelling the GENIE Audio Kit within Circuit Wizard 3 because it can select the settings automatically, since it knows which printed circuit board you are using.

Finally, click on the **Sound** drop-down box to reveal a list of available sounds.



You are not limited to the sounds that come with the software. With both Circuit Wizard 3 and GENIE Programming Editor, you can import sound effects from audio software such as **Audacity**. To import, simply drag a '.wav' file (8- or 16-bit, uncompressed) onto the **Sound FX Centre** or **Sound** windows.





Understanding sound samples and audio

13

The GENIE microcontroller at the heart of the GENIE Audio Kit plays sound effects by replicating sound waves.

Sound waves

Sound waves are compressions in the air molecules that are picked up by your ear or a sensor, such as a microphone.

These waves are then **digitised** by GENIE so that they can be stored in the microcontroller's memory (see the middle picture on the right). This process, known as **sampling**, involves determining a numerical value for the amplitude (its height or intensity) at each point along the wave.

The quality of a digitised sound is determined by two main factors, (1) the number of separate measurements that are taken, and (2) the range of numbers that are used to store each of those individual amplitude measurements.

For the GENIE microcontroller, it takes 16,000 measurements of the wave per second, and records each of those measurements as a value ranging between 0 and 255.

When the GENIE microcontroller is playing back the sound, it takes each of these stored numerical values and then adjusts the position of the loudspeaker cone to replicate the compression wave of the original sound.

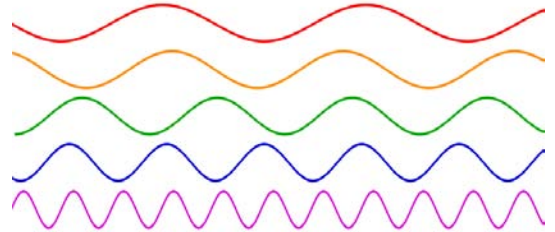
You can learn more about sound via the following Wikipedia article: <http://en.wikipedia.org/wiki/Sound>

Compression

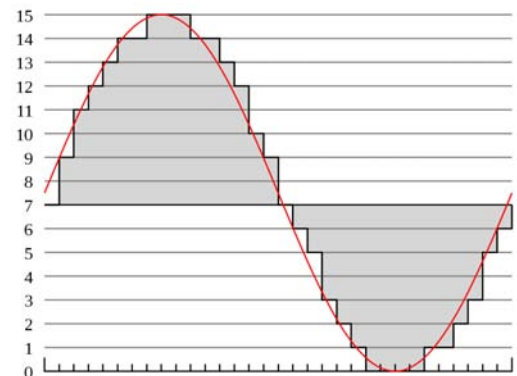
Digitised sound waves take up a lot of memory. For example, to store just one second of a sound recording at 16,000Hz with 8-bits per value (giving a range of 0 to 255) requires exactly 16,000 bytes of memory.

On a desktop or tablet computer this would be no problem, but a small microcontroller such as the GENIE 14 used on the GENIE Audio Kit only has about 10,000 bytes to store your entire flowchart or BASIC program.

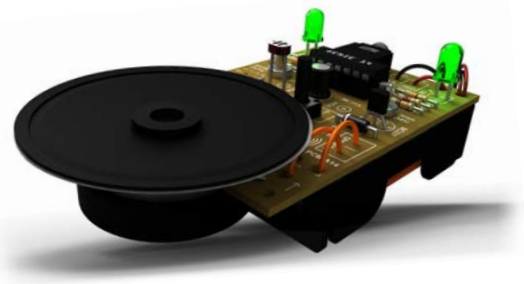
To resolve this, GENIE **compresses** the digitised sound so that it occupies a much smaller amount of memory, which is why you can have several different sound effects in your programs (the number depending on the sounds used).



▲ Sound waves with different frequencies. The waves shown are pure sine waves.



▲ A digitised section of a sound wave.



▲ The loudspeaker cone is controlled by a magnet (which is itself controlled by the GENIE microcontroller) that forces the cone to move. This movement generates compression waves in the air, which are eventually picked up by your ear, forming the sound you hear.





Responding to inputs from the push switch

14

Some types of input signal, such as push switches, can only be either on or off. These are known as **digital** signals.

On the GENIE Audio Kit there is a push switch. It is connected to the input marked by the text 'A/D0'.

This means that it is the input number **0**. The 'A/D' prefix indicates that the GENIE pin to which the input is wired allows both analogue and digital components to be connected. When used for digital within the software, it will be called **D0**.

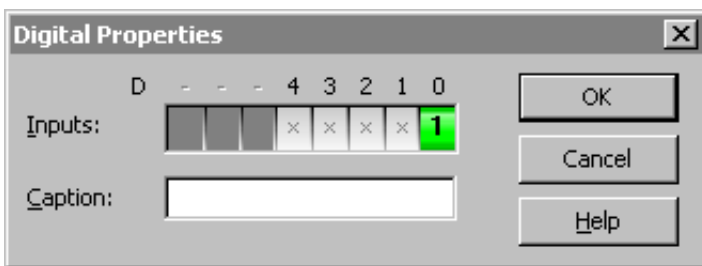


Use the **Digital** command to respond to a digital signals.

The **Digital** command allows you to make a decision based on whether the push switch (or any digital signal) is either pressed (on) or not pressed (off).

When a digital signal is on, it has the value '1' whereas when it is off, it has the value '0'.

Double-click on the command to select which digital inputs you wish to check. GENIE will follow the 'Y' (yes) path when the digital signal matches the chosen pattern, otherwise it will follow the 'N' (no) path. For the GENIE Audio Kit, we suggest clicking on the right-most input until it shows a '1'.



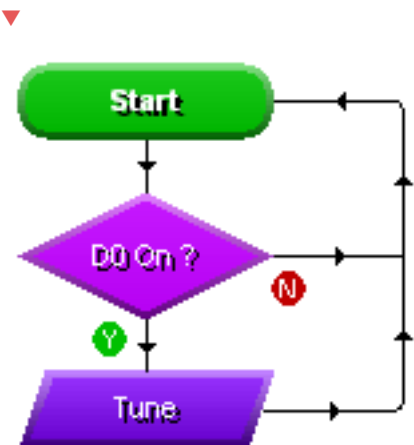
The **Digital** command should be incorporated into a larger flowchart. The flowchart tells the GENIE microcontroller what action to take when it reaches the command.

This can be seen on the right. In the flowchart shown, a tune will be played whenever the switch is pressed.



▲ A view of the GENIE Audio Kit, showing the push switch.

A simple flowchart that plays a tune whenever the push switch is pressed. The **Digital** command has two lines coming out of it, unlike the others.





Responding to inputs from the light sensor 15

Not all input signals are digital. Some types of signal, such as those for temperature or light, can be at a number of different levels. These are known as **analogue** signals.

On the GENIE Audio Kit there is a light sensor. It is connected to the input marked by the text 'A/D4'.

This means that it is the input number **4**. As with the push switch, the 'A/D' prefix indicates that the GENIE pin to which the input is wired allows both analogue and digital components to be connected. When used for analogue, it will be called **A4**.



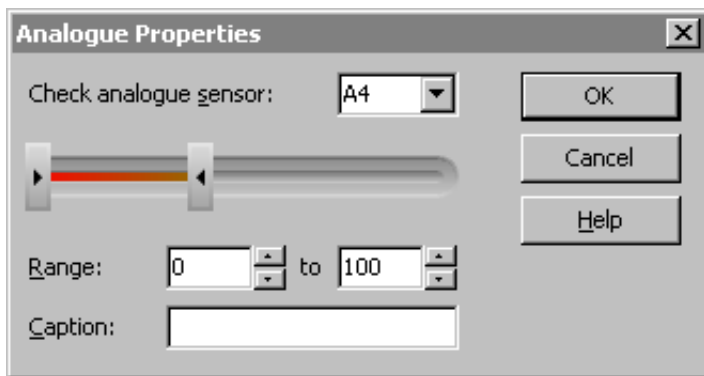
Use the **Analogue** command to respond to analogue signals.

The **Analogue** command allows you to check an analogue signal to see if its value lies within a given range.

With GENIE, analogue levels can vary between **0** (the lowest level) and **255** (the highest). For the light sensor, low values are when less light is detected, and high values when it is brighter.

Double-click on the command to select a sensor to check and a range. GENIE will follow the 'Y' (yes) path when the signal is in range, otherwise it will follow the 'N' (no) path.

For example, to test if the light sensor on analogue signal A4 is between 0 and 100, you should enter the following:



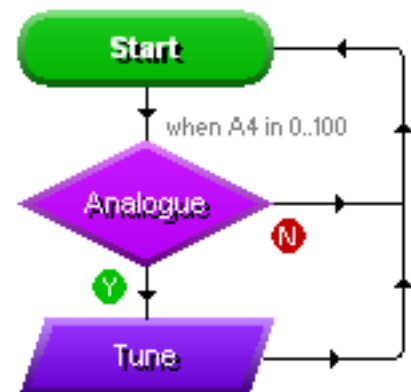
To allow you to fine-tune the sensor values, you can make use of the **Calibrate Sensor** command with a real GENIE Audio Kit.

A view of the GENIE Audio Kit, showing the light sensor.



Light sensor

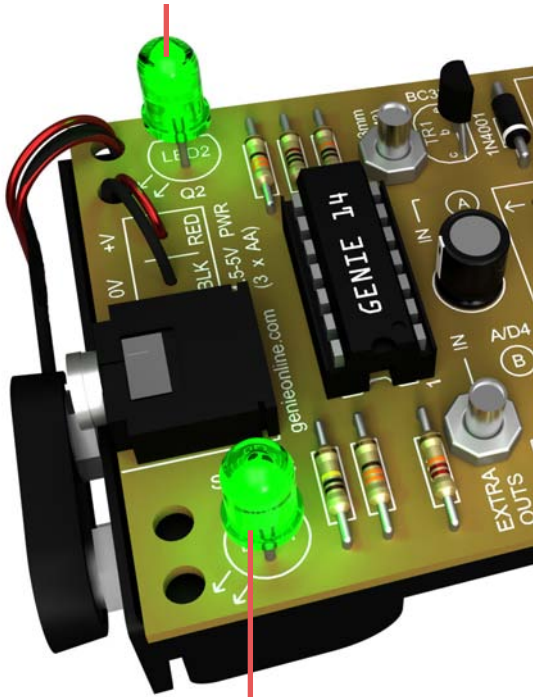
A simple flowchart that plays a tune whenever the light level goes low, such as when the sensor is covered. The **Analogue** command, like the **Digital** command, has two out lines.





Controlling the two light-emitting diodes (LEDs) 16

LED connected to output Q2



LED connected to output Q0

▲ A view of the GENIE Audio Kit, showing the two light-emitting diodes (LEDs).



▲ The first LED (connected to output Q0) has an extra significance when programming GENIE. This output is also referred to as the **STATUS** output because it will flash whenever a program is being downloaded to the microchip (it helps you check that everything is working okay).

The GENIE Audio Kit has two **light-emitting diode**, or **LED**, outputs. These LEDs will light up whenever the appropriate output signal goes high.

Normally the LEDs will flash in time with the music, however when the music is not playing, you can turn the LEDs on or off yourself via flowchart commands.



Use the **High** command to turn a single LED on.



Use the **Low** command to turn a single LED off.

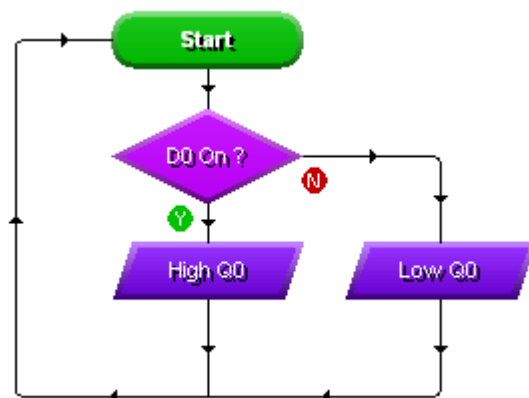


Use the **Outputs** command to control more than one LED.

The two LEDs are connected to the following outputs on the GENIE 14 microcontroller:

- ◆ The first LED is connected to output **Q0**.
- ◆ The second LED is connected to output **Q2**.

An example can be seen below. In the flowchart, the first LED is turned on whenever the push switch is pressed (and turned off again when it is released).



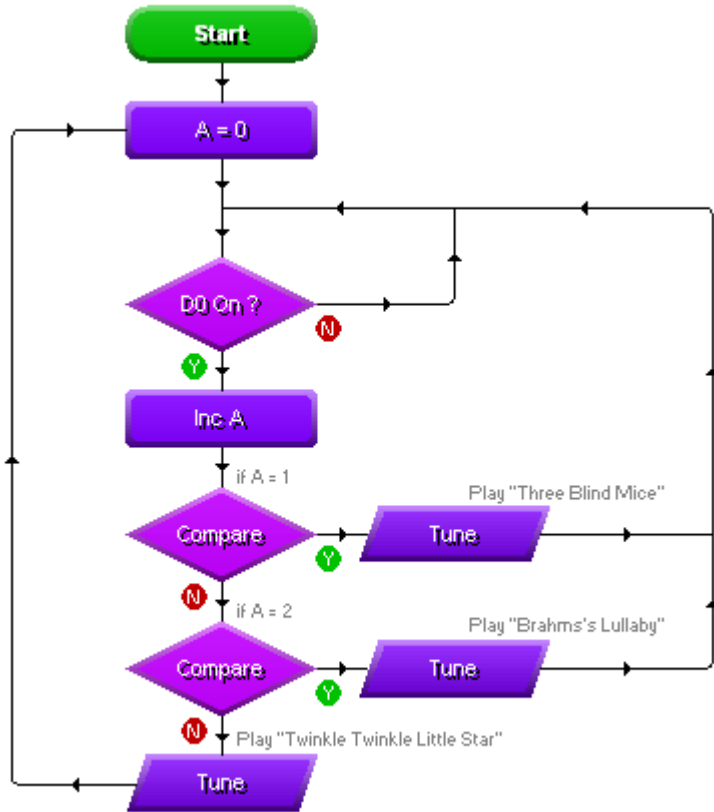
▲ Controlling the LED connected to output Q0.





Playing more than one tune

17



◀ The GENIE microcontroller at the heart of the GENIE Audio Kit allows even more complex flowcharts to be programmed.

One example can be seen on the left. It extends the push switch flowchart so that a different tune will be played (from a list of three) each time the switch is pressed.

The flowchart works by storing the number of the next tune to be played in a **variable**.

Variables are special parts of the GENIE microcontroller's memory that allow you to perform and remember calculations.

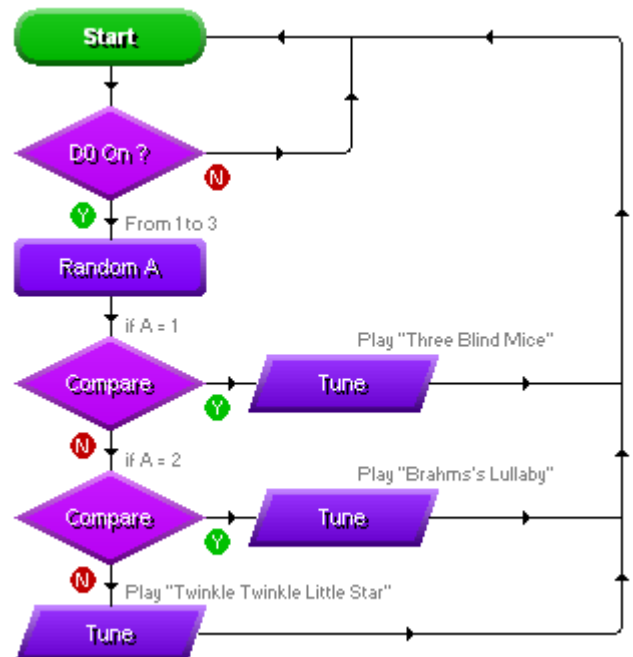
These variables can hold any whole number between 0 and 255 and are referred to by different letters of the alphabet. Here we are using the first variable, named **A**.

The flowchart increases the value stored in variable **A** each time the switch is pressed (this is the **Inc A** command). It then **Compares** the value of **A** to select a tune.

A similar approach is taken in the flowchart on the right. Here, instead of playing the next tune in the sequence, a tune is selected based on the value of a random number between 1 and 3 (inclusive) using the **Random** command.

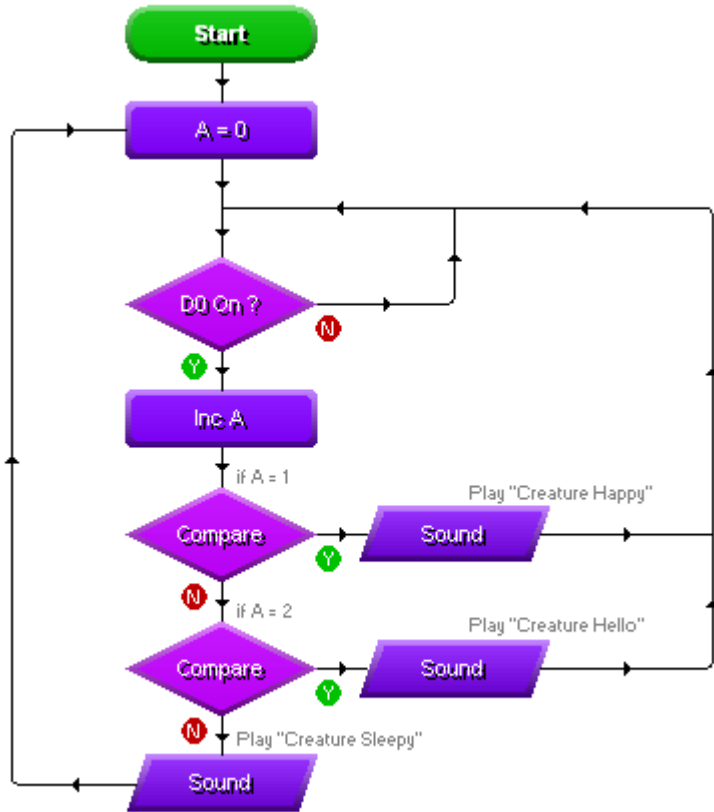
Because the flowchart does not check to see if the value of the random number generated differs from the one chosen last time, the flowchart will sometimes play the same tune twice in succession.

You should also note that because MIDI tunes take up a lot of GENIE's memory, you may need to select shorter tunes when playing more than one.





Playing more than one sound effect 18



◀ On the left is another example of a more complex flowchart. It is similar to the ones before, but instead plays sound effects rather than music whenever the switch is pressed.

The flowchart works by storing the number of the next sound to be played in a **variable**.

Variables are special parts of the GENIE microcontroller's memory that allow you to perform and remember calculations.

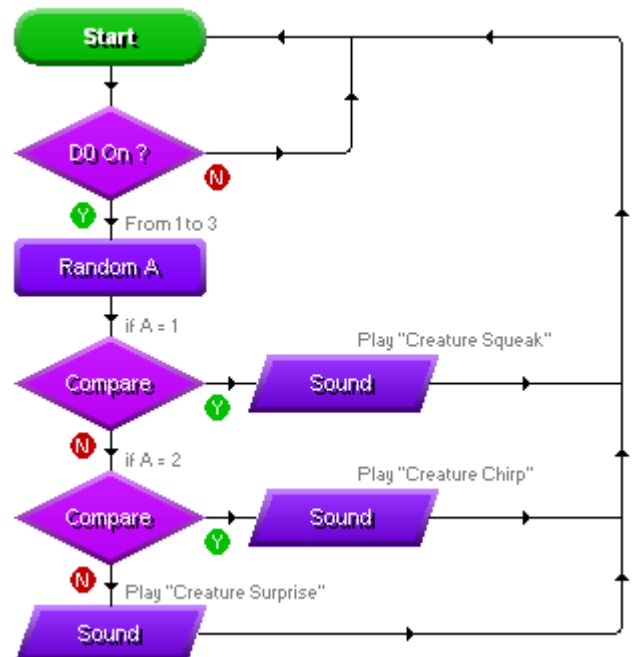
These variables can hold any whole number between 0 and 255 and are referred to by different letters of the alphabet. Here we are using the first variable, named **A**.

The flowchart increases the value stored in variable **A** each time the switch is pressed (this is the **Inc A** command). It then **Compares** the value of **A** to select a tune.

A similar approach is taken in the flowchart on the right. Here, instead of playing the next sound in the sequence, a sound effect is selected based on the value of a random number between 1 and 3 (inclusive) using the **Random** command.

Because the flowchart does not check to see if the value of the random number generated differs from the one chosen last time, the flowchart will sometimes play the same sound twice in succession.

You should also note that because sound effects take up a lot of GENIE's memory, you may need to select shorter ones when playing several.



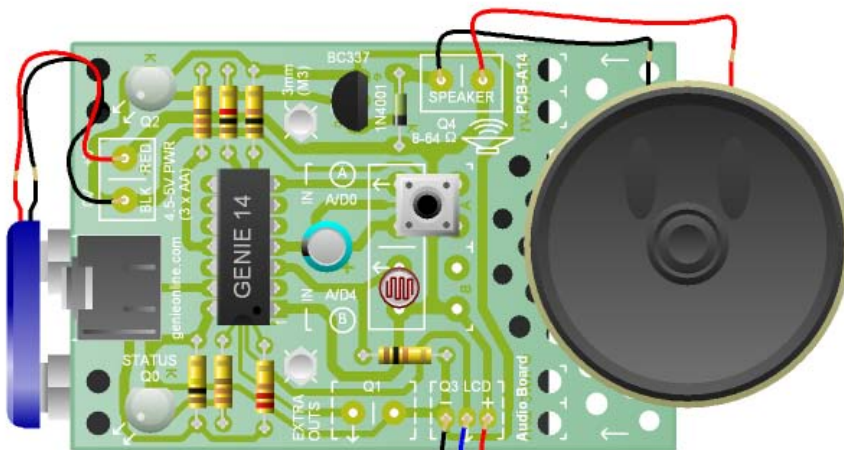
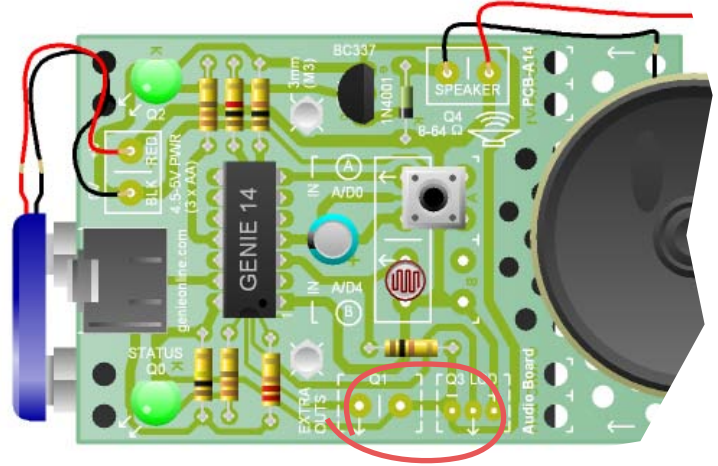


Advanced output options

There are two additional outputs available on the GENIE Audio Kit printed circuit board:

Q1 is an unused low-power output (which means there are no transistors or other drivers on the board itself).

For this connection there are two pins: one with an arrow that connects directly to the Q1 output pin on the microcontroller, and a second pin that connects to 0 volts.



Q3 provides connections for a separate LCD module. For this output there are three pins for 0 volts, signal and power supply wires respectively. The picture on the left shows how an LCD can be connected.

For more information on using the LCD module, please refer to the separate **GENIE Serial LCD module pdf**.

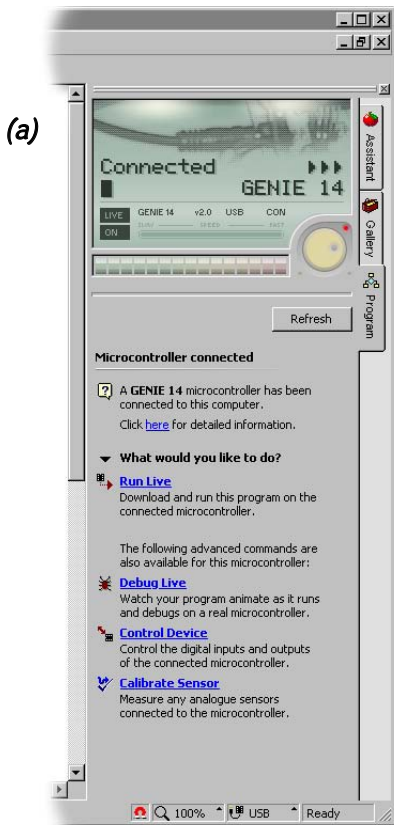




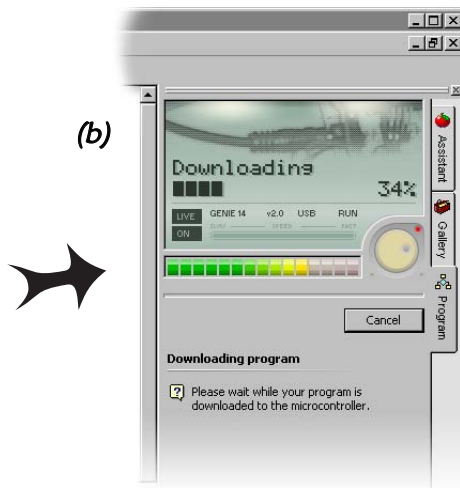
Downloading a program to the GENIE microcontroller 20

Once you have written your flowchart program, you need to store it on the GENIE chip. Here's how you do it:

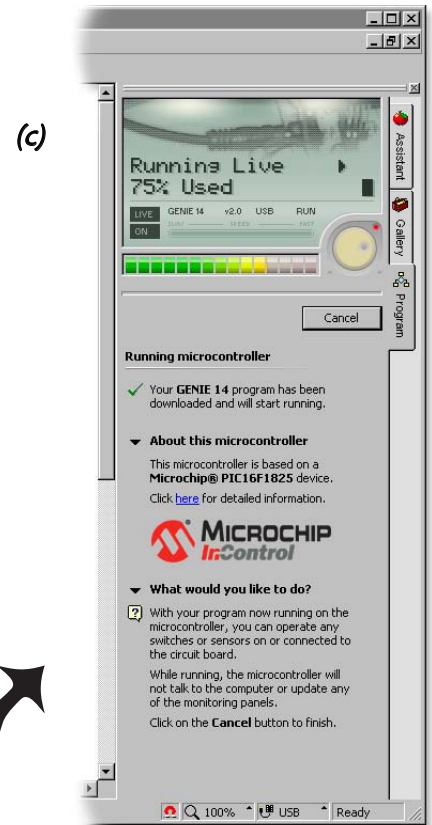
- 1 Wire-up the built GENIE circuit board and connect up a suitable battery power supply.
- 2 Plug the GENIE cable into the download socket on the GENIE circuit board.
- 3 Once done, the **Program** panel in the software will then show a 'Connected' message (see picture a).
- 4 Click on the **Run Live** option. Your flowchart will be transferred onto the GENIE chip—this is known as **downloading** (see picture b).



The green STATUS LED will flash as the download takes place. It tells you everything is OK!



If you have problems downloading your flowchart, see the fault-finding and troubleshooting hints and tips on the next page.



As soon as the program has been downloaded you will see the above screen (c) and GENIE will start running your flowchart.

Your GENIE project is now ready to go! You can disconnect the cable and use your GENIE Rock Star away from the computer.

Finished!





Fault-finding and troubleshooting 21

If you are unable to connect to a GENIE microcontroller or download a program, you should go through the following troubleshooting hints and tips.

A Run the GENIE troubleshooting tool

The GENIE troubleshooter will automatically check your cable and software to ensure that the computer can access the GENIE cable.

To run the GENIE troubleshooter, choose **Troubleshoot GENIE...** from the **Help** menu of the Circuit Wizard or GENIE software.

If that option is not shown in your version of the software, you can download it separately from www.genieonline.com/cable.

Step through the on-screen instructions.



B Step through the following checklist of common problems

Cable

- Circuit Wizard, GENIE Design Studio and the GENIE Programming Editor software all check and report problems involving the cable. If given, follow through on the on-screen advice.
- Unplug the cable, wait a few seconds and then plug it back in. Windows can occasionally fail to detect that a cable has been inserted.

Power

- Check that the voltage of the battery is sufficient. For this project, the battery voltage should be in the range of 4.5 volts. You should never connect batteries totalling 6V or more.
- Check the voltage level across the power connections (+V and 0V) on the board. This can identify if there is a problem with the battery clip or battery holder. Ensure that the wiring has not become loose and the batteries are properly seated in the holder.

Circuit

- Try plugging the cable into another GENIE board if you have one available. When powering up this circuit, the green STATUS LED should flash once (when properly connected it will flash repeatedly).
- Try with another GENIE microcontroller if possible.
- Visually inspect the board for bad solder joints or cases where soldering has incorrectly bridged pins together. Note that for the download socket, the two left-most pins should be connected together, as should the two right-most pins.

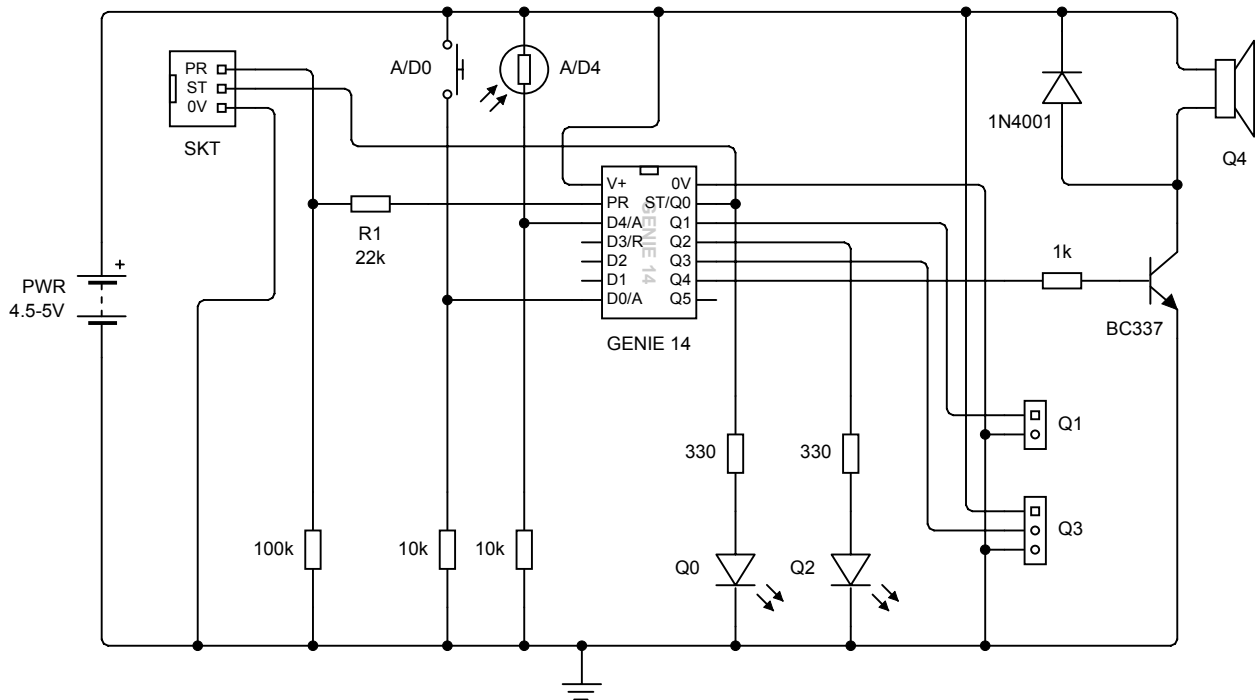
For more troubleshooting hints and tips, please read the separate **GENIE Troubleshooting Guide**.





The technical bit... it's only needed if you want to learn more!

This is the **circuit diagram**. It shows how all of the components in the electronic circuit are connected.



If you have the Circuit Wizard 3 software, you find additional GENIE resources within the **Resource Centre**.

