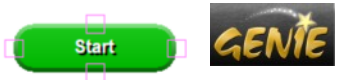


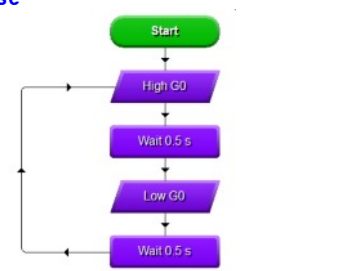
CODE

Genie Code layout

Snippet:



Usual use




Function:
This is the starting block, which is created when a new Genie program is opened. The start block is the program start, any variables or actions that need to be done first should be placed after this block and before the main program loop.

CODE

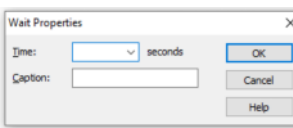
How to delay for a set time

Snippet:



Parameters

s: the number of seconds to pause, type in value required or a variable select from list.



Return

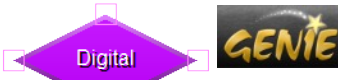
- None

Function:
Pauses the program for the amount of time (in s) specified as parameter. Right clicking on the block brings up the properties box, which allows to set the parameters you require.

CODE

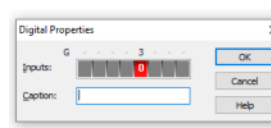
Reading a digital input

Snippet:



Parameters

- pin: the number of the digital pin you want to read, select from the properties box.



Return


- 0 or 1

Function:
Reads the value from a specified digital pin, either **0** or **1**. You can select any pin from the drop-down menu, only those pins that can be used as inputs will be shown.

CODE

Reading an analog input

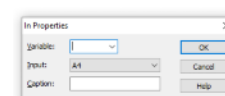
Snippet:



Parameters

- Variable: the variable to use
- Input: which input to use - analog

Use the properties box to select the required values.



Return


- 0 to 255

Function:
Reads the analog voltage on a pin and returns a value 0 to 255 into a variable.
Only applies to pins that have analog functions.

CODE

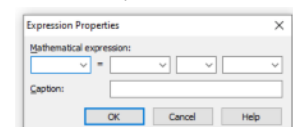
Create a variable

Snippet:



Parameters

- Name: the variable to use
- Value: the value required



Return

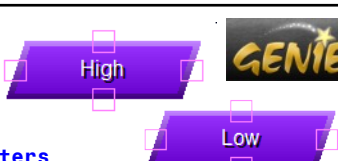
- None unless a calculation is performed

Function:
This is used to select numerical variables. Genie has a set number of variables labelled A to Z plus Time., they may need to be initialised at the start of your code - see the Genie code layout card. It is also used to perform calculations.

CODE

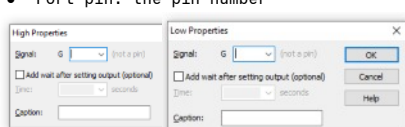
Operate a Digital Output

Snippet:



Parameters

- Port pin: the pin number



Return


- None

Function:
High will set the selected pin to 1, Low will set the pin to 0.

CODE

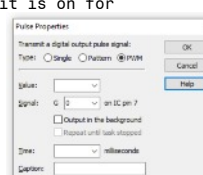
Operate an Analog Output

Snippet:



Parameters

- Value: 0 to 255, fast to slow
- Signal: which pin to use
- Time: how long it is on for



Return


- None

Function:
Writes an analog value (**PWM wave**) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. The block make the pin will generate a steady square wave of the specified value for the set time.

CODE

Pause in milliseconds


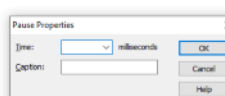
Snippet:



Parameters

- Via properties box



Example

Function:
This block will pause for the required number of milliseconds, you can specify a value or use a variable.

CODE


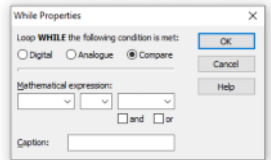
Looping for a **While**

Snippet: **Syntax**  

Parameters

- Type of condition
- expression: a statement that evaluates to **true** or **false**



Example

Function:
while loops will loop continuously, and infinitely, until the expression becomes false. Something must change the tested variable, or the while loop will never exit. This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.

CODE


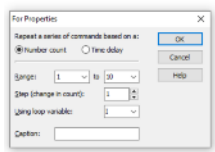
Looping **FOR** a number times

Snippet: **Block**  

Parameters

- Range: count from and to
- Step: increment value + or -
- Loop variable: which variable to use



Example

Function:
 The **for** statement is used to repeat a block of statements inside the loop. Each time through the loop, the variable is increased by 1 from the starting value until the maximum value is reached, then loop then ends.

CODE

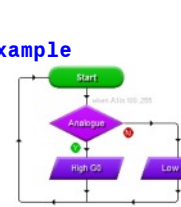
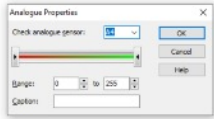
If decision making - analog

Snippet: **Block**  

Parameters

- Analog pin: which pin to use
- Range: is the analog value in that range?



Example

Function:
 The **analogue** block allows decisions to be made based upon the value or range of values seen on the analog pin, which is usually connected to a sensor.

CODE

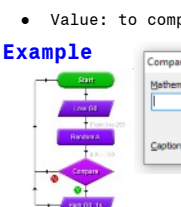
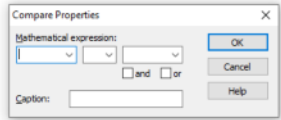
If decision making - variables

Snippet: **Block**  

Parameters

- Variable: which variable to use
- Comparisons: ==, !=, >, <, <=, >=
- Value: to compare against


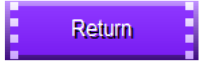

Example

Function:
if, is used in conjunction with a comparison to tests whether a certain condition has been reached, such as an input being above a certain number. if the comparison is true, the **Yes** branch is executed. If not, the **No** branch is executed.

CODE

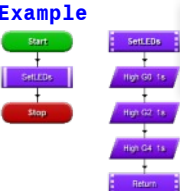
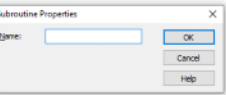
Creating a **Sub routine**

Snippet: **Block**   

Parameters

- Name: name of the routine



Example

Function:
 A **sub routine** is a piece of code that can be used many times over, but only has to be written once. It is used in your main code by the **Call** block.

CODE



If **input** decision making

Snippet: **Block**  

Parameters

- Inputs: which pin(s) to use



Example

Function:
 This **if** block is specifically for use with inputs, the input pin(s) needs to be selected, and the off or on needs to be selected as well.
Note: the inputs you can select are those available on the Genie type you are using.

CODE

Choosing a **Random** number

Snippet: **Block**  

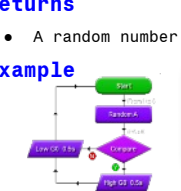
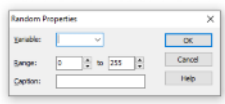
Parameters

- Variable to use
- Range: min to max value

Returns

- A random number between 0 and 255


Example

Function:
 This block stores a random number into the given variable. Use the down arrow to select other variables if needed.

CODE

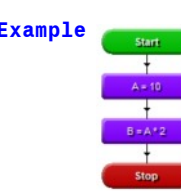

Basic maths

Snippet: **Arithmetic Operators** 

The basic maths operations:

- Addition
- Subtraction
- Multiplication
- Division



Example

Function:
 The maths operations follow standard maths operations, you select the required maths operation from the drop down menu.
 There are some additional advanced operations in the list.

CODE


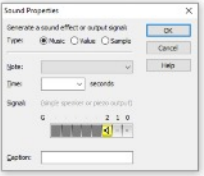
Play a musical note

Snippet: **Block**  

Parameters

- Note: which musical note
- Time: how long to play for in s
- Pin: which pin to use



Example

Function:
This block generates an audio tone of set frequency, for a specified duration in s, the pin to use is selected using the down arrow.

CODE


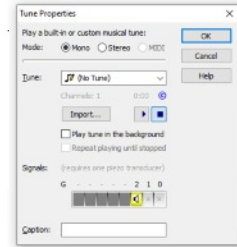
Play a tune (built-in)

Snippet: **Block**  

Parameters

- Via properties



Example

Function:
This block plays one of built-in tunes, with this block only certain pins can be used depending upon the Genie chip type.

CODE

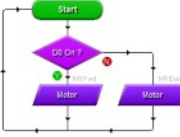
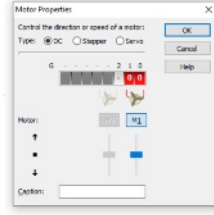
Operate a DC Motor

Snippet: **Block**  

Parameters

- Via properties box - DC



Example

Function:
This block operates a DC motor using two output pins to give forward, stop and reverse control. You will need external electronics to drive the motor such as an L293D or SN754410NE.

CODE

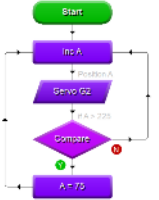
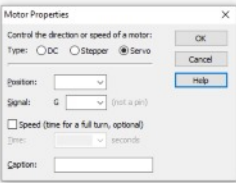
Set servo position

Snippet: **Block**  

Parameters

- Via properties box - Servo



Example

Function:
This block controls the position of an attached servo motor. For a standard servo the angle is 0 to 180, using values 75 to 225, 0 turns the servo off. A continuous rotation version, 75 = rotate in one direction, 150 = stop and 225 rotates in the other direction. The set servo block initialises the servo.

CODE

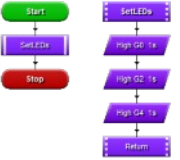
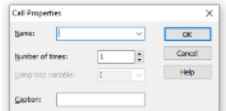
Calling a Sub routine

Snippet: **Block**  

Parameters

- Name: name of the routine to call
- Times: number of times to call - optional



Example

Function:
The **Call** block is used to run the named sub routine once, but you can also adjust that to more than 1 if needed.

CODE

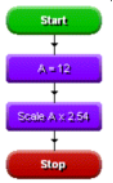
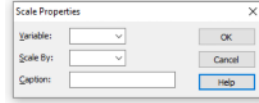
Scale a variable

Snippet: **Block**  

Parameters

- Via properties box



Example

Function:
This block allows a variable to be scaled by the following amounts: 0.1, 2.54, 3.14. The variable is multiplied by the scale factor, useful when dealing with converting sensor readings into real-world measurements - distance etc.

CODE

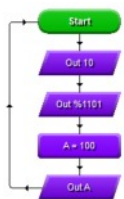
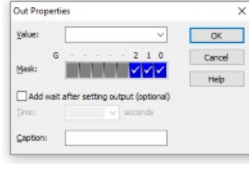
Output a number to pins

Snippet: **Block**  

Parameters

- Variable: which variable to use
- Mask: which bits to use



Example

Function:
This block outputs a number in binary to the output pins as G4, G2, G2, G1, G0. You can mask out bits that you don't want the block to operate on via the properties box.

CODE

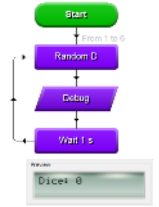
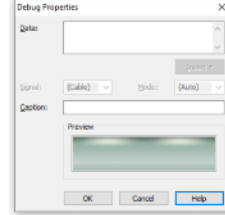
Debug your code

Snippet: **Block**  

Parameters

- Via properties box

Example

Function:
This block sends a message or data back to the computer via the programming cable when using the **Debug Live** feature. The message or data you wish to send is set in the properties box.

CODE

Input a value to a variable

Snippet: **Block** GENIE

Parameters

- Variable: to store input value in
- Input: which input to read and store

Example

Function:
This block reads the inputs and stores the value into the specified variable. Digital reads the input pins being used, it also can read analog pins and a 1-wire temperature sensor.

CODE

Toggle an output pin

Snippet: **Block** GENIE

Parameters

- Signal: which pin to use

Example

Function:
This block toggles the state of an output pin, low to high or high to low (0 > 1 or 1 > 0).

CODE

Repeat Until loop

Snippet: **Block** GENIE

Parameters

- Via properties box

Example

Function:
This block repeats a set of commands until a comparison is met, this is set as an expression in the properties box, which can be digital pins, analog input value or a variable comparison.

CODE

Pulse a pin a number of time

Snippet: **Block** GENIE

Parameters

- Via properties box

Example

Function:
This block pulses an output pin either with a single pulse of specified duration, or a specific pattern of on's & off's or as a PWM signal.

CODE

Increment a variable

Snippet: **Block** GENIE

Parameters

- Variable: to increment

Example

Function:
This block increments the specified variable by 1. If the value goes above 255 then it rolls over to 0.

CODE

Decrement a variable

Snippet: **Block** GENIE

Parameters

- Variable: to decrement

Example

Function:
This block decrements the specified variable by 1. If the value goes below 0 then it rolls over to 255.

CODE

Set a bit in a variable

Snippet: **Block** GENIE

Parameters

- Bit: to set to 1 (High)

Example

Function:
This block sets a bit in a variable to 1 (High), order of bits is 76543210. If no bit is specified **all** bits set to 1.

CODE

Clear a bit in a variable

Snippet: **Block** GENIE

Parameters

- Bit: to set to 1 (High)

Example

Function:
This block sets a bit in a variable to 0 (Low), order of bits is 76543210. If no bit is specified **all** bits cleared to 0.