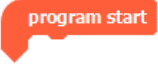



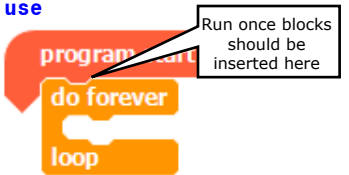
KS2 Cards


CODE

Crumble Code layout

Snippet:  

Usual use

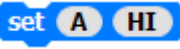



Location: 

Function:
This is the starting block, which is created when a new Crumble 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

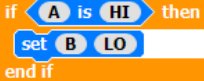
Operate a Digital Output

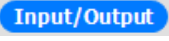
Snippet: **Block**  

Parameters

- Pin: the pin letter, A, B, C or D
- value: Lo or Hi (off or on)

Example





Location: 

Function:
This will set a pin to output Lo or Hi, on or off, click on the pin letter or value to change.

CODE

Reading a digital input

Snippet: **Block**  

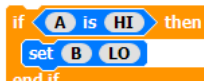
Parameters

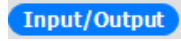
- pin: the pin you wish to read A, B, C or D

Return

- Lo or Hi (off or on)

Example

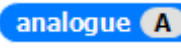



Location: 

Function:
Reads the value from a specified digital pin, either **Lo** or **Hi**. You can select any pin by clicking on the pin letter, clicking on the Lo/Hi will change the value to be tested.

CODE

Reading an analog input

Snippet: **Block**  

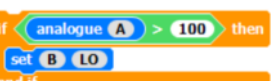
Parameters

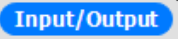
- Value: the value from an analog pin A, B, C or D

Return

- 0 to 255

Example






Location: 

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

CODE


How to delay for a set time


Snippet: **Block**  
 

Parameters

- Time: the number of milliseconds, the same applies to the seconds variant.

Example





Location: 

Function:
Pauses the program for the amount of time (in ms) specified as parameter. (There are 1000 milliseconds in a second.) Or in seconds using the second block.
The time value can be a variable or calculation.

CODE

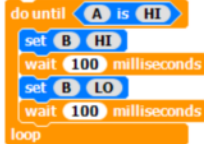
Do Looping until

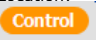
Snippet: **Syntax**  

Parameters

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

Example





Location: 

Function:
Do 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


Do a number of times loop


Snippet: **Block**  

Parameters

- Value: number of times to do the loop

Example

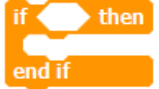



Location: 

Function:
The **Do** loop simply repeats the code within the loop the until the number of times has been done. The number can also be a variable if required.

CODE


If decision making


Snippet: **Block**  

Parameters

- Comparisons: =, <, >, ≠

Example




Location: 

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 statements inside the brackets are run. If not, the program skips over the code.

KS2 Cards


CODE

If ... Else decision making

Snippet:
Block 

Parameters

- Comparisons: =, <, >, ≠


Example 

Location:
Control





Function:
if/else allows greater control over the flow of code than the basic if statement, by allowing multiple tests to be grouped together. For example, an analog input could be tested and one action taken if the input was less than 128, and another action taken if the input was 128 or greater.

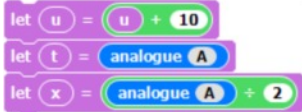
CODE

Basic maths

Snippet:
Arithmetic Operators 

The basic maths operations:

- Addition 
- Subtraction 
- Multiplication 
- Division 


Example 

Location:
Operators

Function:
The maths operations follow standard maths operations, they need to be placed into a let variable block. The calculations can include variables, numbers or analog value. You select the required maths operation from the operators list.

CODE

Create a variable


Snippet:
Block 

Parameters

- Name: the name of your variable

Return

- None

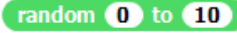
Example 

Location:
Variables

Function:
This is the method use to set up numerical variables, they may need to be initialised at the start of your code - see the Crumble code layout card. Click Add New Variable and enter the name of your new numerical variable, it will now be added to your variable list.

CODE

Choosing a Random number


Snippet:
Block 

Parameters

- Smallest number, largest number needed

Returns

- A random number between smallest & largest

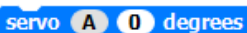
Example 

Location:
Operators

Function:
This block stores a random number into the given variable. Use the variables to select other variables if needed. To use a new new, you need to create a new variable - see the Create a variable card.


CODE

Set servo position

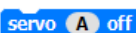
Snippet:
Block 

Parameters

- Pin: which pin to use A, B, C or D
- Angle: what angle to set motor to


Example 

Location:
Input/Output

Function:
This block controls the position of an attached servo motor. For a standard servo the angle is 0 to 180. A continuous rotation version, 0 = rotate in one direction, 90 = stop and 180 rotates in the other direction. This block operates all the time unless stopped using: 

CODE

Set a SPARKLES colour


Snippet:
Block 

Parameters

- Sparkle number, sparkle colour

Returns

- None


Example 

Location:
Sparkles

Function:
This block turns on a specific Sparkle (LED) with a specific colour. Sparkles are RGB (Red Green Blue) LEDs, which can be connected in series to make a chain of LEDs or even arranged in a matrix.

CODE

Turn a SPARKLE off


Snippet:
Block 

Parameters

- Sparkle number

Returns

- None

Example 

Location:
Sparkles

Function:
This block turns off a specific Sparkle (LED) with a specific colour. Sparkles are RGB (Red Green Blue) LEDs, which can be connected in series to make a chain of LEDs or even arranged in a matrix.

CODE

Set all SPARKLES colour

Snippet:
Block 

Parameters

- None

Returns

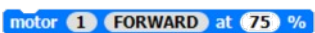

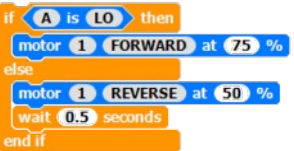

- None

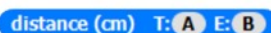

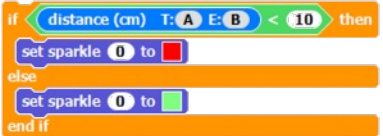

Example 

Location:
Sparkles

Function:
This block turns all Sparkles to a specific colour. Sparkles are RGB (Red Green Blue) LEDs, which can be connected in series to make a chain of LEDs or even arranged in a matrix.

KS2 Cards

CODE	
Motor control	
Snippet:	
Block	
Parameters	<ul style="list-style-type: none">Motor number, direction, speed%
Example	
Location:	
Function:	Controls motors which are connected to the motor pins. The motors can be made to turn forwards, reverse and stop, they can also be given a speed from 0% to 100%, 50% is half speed.

CODE	
Distance sensing	
Snippet:	
Block	
Parameters	<ul style="list-style-type: none">Trigger pin, echo pin
Example	
Location:	
Function:	Measures the distance from the Ultrasonic sensor to an object in cm's. The block is designed for the HC-SR04 ultrasonic sensor module, which is readily available online.