

KS2 Cards

CODE

Micro:Bit Code layout

Snippet:

Location: Basic

Function:
The two basic starting blocks. On start the code in this block only runs once when the code starts, used to setup variables and constant values.
The Forever block provides the continual loop that will be the core of your code, any code placed in it will run all the time.

CODE

How to delay for a set time

Snippet:

Parameters

- ms: the number of milliseconds to pause, choose from list or type in value required

Return

- None

Location: Basic

Function:
Pauses the program for the amount of time (in ms) specified as parameter. (There are 1000 milliseconds in a second.)

CODE

Reading a digital input

Snippet:

Parameters

- pin: the number of the digital pin you want to read

Return

- 0 or 1

Location: Advanced Pins

Function:
Reads the value from a specified digital pin, either **0** or **1**. You can select any pin from the drop-down menu, P0, P1 and P2 are the large ring connections for clips, the others will require a breakout board with a full connector.

CODE

Reading an analog input

Snippet:

Parameters

- Value: the value from the analog pin

Return

- 0 to 1023

Example

Location: Advanced Pins

Function:
Reads the analog voltage on a pin and returns a value 0 to 1023.
Only applies to pins: P0, P1, P2, P3, P4 and P10.

CODE

Create a variable

Snippet:

Parameters

- Name: the name of your variable

Return

- None

Location: Variables

Function:
This is the method used to set up numerical variables, they may need to be initialised at the start of your code - see the Micro:Bit code layout card. Click on the make a variable button and enter the name of your new numerical variable.

CODE

Operate a Digital Output

Snippet:

Parameters

- pin: the pin number
- value: 0 or 1

Return

- None

Location: Advanced Pins

Function:
This will set a pin to output 0 or 1, on or off.

CODE

Operate an Analog Output

Snippet:

Parameters

- pin: the pin to write to
- value: the duty cycle: between 0 (always off, 0%) and 1023 (always on, 100%)

Return

- None

Location: Advanced Pins


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 makes the pin will generate a steady square wave of the specified duty cycle until the next command on the same pin.

CODE

On shake event

Snippet:

Parameters



Location: Input



Function:
This is an **EVENT** block which is used separately to On start and Forever blocks, and is triggered when the Micro:Bit is shaken. Use the drop down menu to select other shake options.



KS2 Cards

CODE

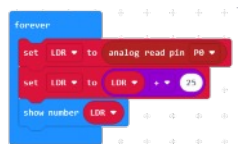
Display a number


Snippet: **Block**  

Parameters

- number: the variable or number to display

Example





Location: 

Function:
This block will display a number on the LED matrix, if the number is more than 1 digit it will scroll the value.

CODE


Display an LED pattern


Snippet: **Block**  

Parameters

- Click the blocks to turn on/off individual LEDs

Example





Location: 

Function:
This will display a pattern on the LEDs as set in the LED matrix.


CODE

Show an icon


Snippet: **Block**  


Parameters

- Select from:



Example





Location: 

Function:
This block display a one of the in-built icons, select the drop down menu to choose other icons.

CODE

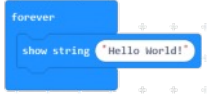
Show a string


Snippet: **Block**  

Parameters

- String: the text you wish to display

Example





Location: 

Function:
This block displays a string (a piece of text), it can contain up to 255 characters.

CODE

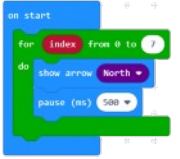
Show a direction arrow


Snippet: **Block**  

Parameters

- Direction: 8 compass directions

Example





Location: 

Function:
This block will display one of 8 compass directions, use a value 0 to 7 to select each one.

- N, NE, E, SE, S, SW, W, NW - 0 to 7


CODE


Clear screen

Snippet: **Block**  

Parameters

Example





Location: 

Function:
This block clears the current contents of the LED matrix - all LEDs are off.

CODE

Create a string Array




Snippet: **Block**  

Parameters

- Name: of your array
- (-): delete a value
- (+): adds a value

Return

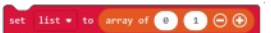

- None

Location:   

Function:
A string is used to store text that we might print, display or send to another device - computer, printer or file

CODE

Create a numerical Array




Snippet: **Block**  

Parameters

- Name: of your array
- (-): delete a value
- (+): adds a value

Return

- None

Location:   



Function:
An array is a collection of numerical variables that are accessed with an index number (pointer). A pointer is a variable used to select which value you require from the array. Arrays are **zero indexed**, that is, the first element of the array is at index 0

KS2 Cards

CODE

Looping for a **While**


Snippet:
Syntax

Parameters

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

Example





Location:
Loops

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 of times

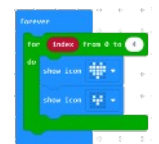
Snippet:
Block

Parameters

- Index: counter variable
- test: max count value

Example





Location:
Loops

Function:
The **for** statement is used to repeat a block of statements inside the loop. Each time through the loop, the index is increased by 1 until the maximum value is reached, then loop then ends. The number of times the loop is done is 1 more than the max value, since the loop starts at 0.

CODE

Repeat a number of times


Snippet:
Block

Parameters

- Expression or number: number of repeats

Example





Location:
Loops

Function:
The **repeat** loop simply repeats the code within the loop the specified number of times.

CODE

If decision making

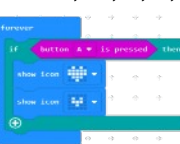
Snippet:
Block

Parameters

- Comparisons: ==, !=, >, <, <=, >=

Example





Location:
Logic

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.

CODE

If ... Else decision making


Snippet:
Block

Parameters

- Comparisons: ==, !=, >, <, <=, >=

Example





Location:
Logic

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 500, and another action taken if the input was 500 or greater.

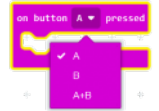
CODE

On Button event

Snippet:
Block

Parameters





Location:
Input

Function:
This is an **EVENT** block which is used separately to On start and Forever blocks, and is triggered when the Micro:Bit is button(s) is pressed. Use the drop down menu to select other on button options.

CODE

Mapping to a range

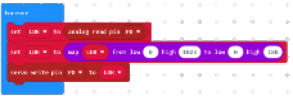
Snippet:
Block

Parameters

- value: variable to remap
- From low: old lowest value, high: old highest value
- To low: new low value, high: new highest value

Example



Location:
Math


Function:
Re-maps a number from one range to another range.

In the example an analogue value ranging from 0 to 1023 is remapped to 0 to 180 to set the position of a servo motor.





CODE

Basic maths

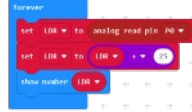
Snippet:
Arithmetic Operators



The basic maths operations:

- Addition 
- Subtraction 
- Multiplication 
- Division 

Example



Location:
Math

Function:
The maths operations follow standard maths operations, they need to be placed into a set variable block.

CODE

On pin event

Snippet: **Block**

Parameters

- on pin P0 pressed
- dropdown menu with P0, P1, P2

Location: Input

Function:
This is an **EVENT** block which is used separately to On start and Forever blocks, and is triggered when a Micro:Bit is pressed. Use the drop down menu to select other pin options.

CODE

Playing a ring Tone

Snippet: **Block**

Parameters

- Frequency

Location: Music

Function:
This block generates an audio tone of set frequency on pin P0. The frequency can be changed by clicking into the note box and pick a new note from the keyboard. The tone plays continuously, use this block to stop all sounds

CODE

Choosing a **Random** number

Snippet: **Block**

Parameters

- Min value and max value limits

Returns

- a random number from min to max values

Example

Location: Math

Function:
This block produces a random number from the min value to the max value

CODE

Play a musical note

Snippet: **Block**

Parameters

- Frequency and duration

Location: Music

Function:
This block plays a musical note for a set length of time. Click in the note box to select a note form the keyboard and a note length form the beat drop down menu.

CODE

Play a melody

Snippet: **Block**

Parameters

- 8 notes of different frequencies and a tempo

Location: Music

Function:
This block plays a set of 8 notes at a set tempo. The notes can be selected by clicking on the melody box and the tempo can be edited by entering a value into the tempo box. Each note (column) can have 1 of 8 values (row), low note at the bottom and high note at the top.

CODE

Play a sound effect

Snippet: **Block**

Parameters

- Type of sound effect and playing options

Location: Music

Function:
This block plays a sound effect from a small set of game related sounds. Use the drop down menu to select the effect and the one on the repeat option to select playing options.

CODE

Stopping **notes** playing

Snippet: **Block**

Parameters

- none

Location: Music

Function:
This block stops the playing of all sounds.

CODE

Reading **Data** from an Array

Snippet: **Block**

Parameters

- Name: name of array
- Position: which value

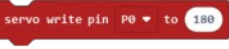

Example

Location: Arrays

Function:
This block gets a value from an array at position given by the variable or number in the at box and places into the first variable.

CODE


Set **servo** position


Snippet:
Block  

Parameters

- Pin: which pin to use
- Angle: what angle to set motor to

Example





Location:


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.

CODE

Constrain a value

Snippet:
Block  


Parameters

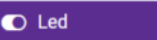
- Low limit and high limit

Return

- A value between low and high only

Example

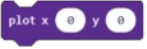



Location:


Function:
Constrains a number to be within a range, if the value has a value less than the lower limit, it is set to that value, similarly if it is larger than the upper limit it is set to that value, other values remain unchanged.

CODE

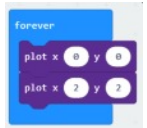
Plot a matrix LED

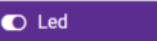
Snippet:
Block  

Parameters

- X: x position 0-4
- Y: y position 0-4

Example



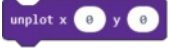

Location:


Function:
This block will plot (turn on) an LED on the LED matrix.

The origin 0,0 is the top left and 4,4 is the bottom right LED.

CODE

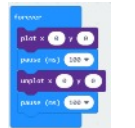
Unplot a matrix LED

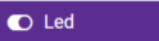
Snippet:
Block  

Parameters

- X: x position 0-4
- Y: y position 0-4

Example





Location:


Function:
This block will unplot (turn off) an LED on the LED matrix.

The origin 0,0 is the top left and 4,4 is the bottom right LED.

CODE

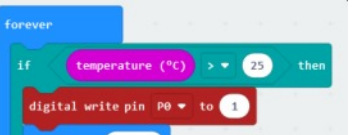
Reading **Temperature** sensor


Snippet:
Block  

Return

- Temperature value in °C

Example





Location:


Function:
This block reads the internal temperature sensor as a value in °C, which then can be stored in a variable or used directly in an if block.

Note: the sensor isn't that accurate, but is ok for simple use, otherwise use an external sensor.

CODE

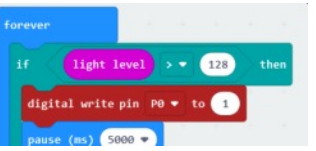
Reading **Light** sensor

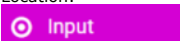
Snippet:
Block  

Return

- Light level value 0 to 255

Example





Location:


Function:
This block reads the light sensor as a value. 0 = dark and 255 = max light, which then can be stored in a variable or used directly in an if block.

Note: the sensor isn't that accurate, but is ok for simple use, otherwise use an external sensor.

CODE

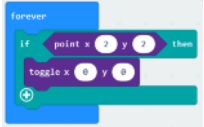
Is a matrix LED on or off?

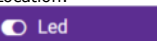
Snippet:
Block  

Parameters

- X: x position 0-4
- Y: y position 0-4

Example

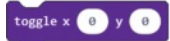



Location:


Function:
This block is used to see if a matrix LED is on or off. It is used in if...then blocks with a comparison to check and then do something as a result.

CODE


Toggle a matrix LED

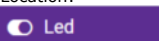
Snippet:
Block  

Parameters

- X: x position 0-4
- Y: y position 0-4

Example



Location:


Function:
This block will toggle an LED on the LED matrix, if it is on it will be turned off, if it is off it will be turned on.



The origin 0,0 is the top left and 4,4 is the bottom right LED.

KS2 Cards

CODE

Reading **Compass** sensor

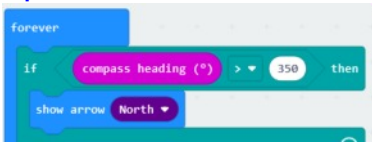
Snippet:

Block   micro:bit

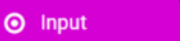
Return

- Light level value 0 to 359 in degrees

Example



Location:

 Input

Function:
This block reads the compass sensor as a value. 0 to 359 degrees, which then can be stored in a variable or used directly in an if block.