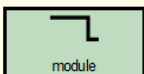
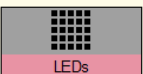


## Mr Bit TASK GUIDES

The task guides provide an alternative approach to the instructions for completing an Exercise or Experiment activity. The format was initially designed to help *slow readers* by reducing the amount of reading compared with the balloon instruction pages on the Mr Bit screens.

Task	What you need	Program script
A single sentence describing the aim of the task.	Visual cues for the blocks needed in the System view.	The target program for the module block.
When there are several stages or extensions to the task, further separate rows are shown.	Additional blocks and prompts appear when the task has more stages.	When more than one sentence appears, this implies a <i>sequence</i> of instructions. When there is more than one module, the script for each is cued with the markers: <b>m1</b> , <b>m2</b> , <b>m3</b> etc.

### Example from Exercise 1 ‘Beautiful Image’

Task	What you need	Program script
Make the micro:bit display an image of your own design.	 	Show the LED image (Star) for 2 seconds. Repeat.

Experience has shown that *confident readers* also like this format, because it involves less reading! If such pupils get stuck, they can always fall back to reading the step-by-step instructions in the balloon pages which are present by default.

The format has also found favour with pupils who have already acquired and are *confident with foundation skills* and have reduced need for the step-by-step instructions in the balloon pages.

The Task Guides for the Mr Bit Exercises on the following pages may be freely printed for classroom use.

Teachers may download versions of the activities files which contain the Task Guides on the Mr Bit screens here:



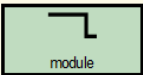
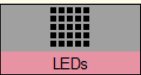
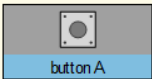
[Task Guides](#)

# 1

## BEAUTIFUL IMAGE



**What's new:**  
 Make an image  
 UNTIL TIME condition  
 WHEN TIME gives a flash


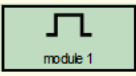
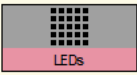

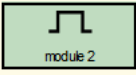


Task	What you need	Program script
Make the micro:bit display an image of your own design.	 	Show the LED image (Star) for 2 seconds. Repeat.
Make the image flash on and off.	Link WHEN to TIME	After 2 seconds show the LED image (Star) for 2 seconds
Add button A so that the image only shows while the button is pressed.		When button A is pressed, show the LED image (Star) until button A is free.

# 2

## SMILEY & FROWNY



**What's new:**  
 Use 'gets pressed' condition with button  
 Use a 'gesture' input for condition

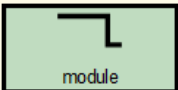
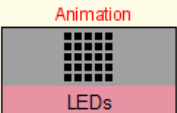

Task	What you need	Program script
Make the micro:bit display a smiley face when you press button A and changes to a frown when you press button B.	    	<b>m1</b> When button A gets pressed, show the LED image (smile) until button B gets pressed.  <b>m2</b> When button B gets pressed, show the LED image (frown) until button A gets pressed.
Use gesture sensors instead of the buttons so that the face smile when the micro:bit is upright and frowns when it lies flat.	Link UNTIL to WHEN (gives WHILE)   + 	<b>m1</b> While the micro:bit is upright, show the LED image (smile).  <b>m2</b> While the micro:bit is face up, show the LED image (frown).

# 3

## SNOWFALL



**What's new:**  
 Make an animation  
 Link UNTIL to empty space to make 'exit' (forever)  
 Use a 'gesture' input with WHEN and UNTIL

Task	What you need	Program script
Make an animation which shows snowflakes falling.	 	Show the LED animation (snowflakes) until exit.
Link the gesture sensor so that the animation only shows when the micro:bit is upright.	+ 	While the micro:bit is upright, show the LED animation (snowflakes).

# 4

## ANSWERING MACHINE



### What's new:

Use buttons and gesture conditions

Task	What you need	Program script
Make the LEDs show the message "ASK ME A QUESTION" when you stand the micro:bit upright on its edge.		<p><b>m1</b> When the micro:bit becomes upright, show the LED message "ASK ME A QUESTION" for 10 seconds.</p>
Make the LEDs show the answer "YES" when button A is pressed, and "NO" when button B is pressed.	<p>Signal Control modules 2 and 3 LEDs - message button A and button B</p>	<p><b>m2</b> When button A gets pressed, show the LED message "YES" for 5 seconds.</p> <p><b>m3</b> When button B gets pressed, show the LED message "NO" for 5 seconds.</p>
Add a "DON'T know" instruction which shows if the micro:bit is shaken.	<p>+</p>	<p><b>m4</b> When the micro:bit is shaken, show the LED message "DON'T KNOW" for 5 seconds.</p>

# 5



## FLASHING HEART

### What's new:

Add a second instruction  
Use a Pause module

Task	What you need	Program script
Make the micro:bit show a heart image which flashes as your heart beats. Adjust the pause time to match your own pulse.		<p>Show the LED image (heart) for 0.5 seconds.</p> <p>Wait for 0.4 seconds.</p>

# 6



## GLOWING SWORD

### What's new:

Use a Pulse module  
Make an image fade

Task	What you need	Program script
Make the LEDs show an image of a sword which varies in brightness to give a glowing effect.	<p>Link UNTIL to WHEN (gives WHILE) (Image: Fade IN and OUT)</p>	While button A is pressed, flash the LED image (sword).
Add a similar system for button B, but showing a sword pointing the opposite way.		While button B is pressed, flash the LED image (opposite sword).

# 7

## LUCKY 7



### What's new:

Use a Counter module  
Display Counter value  
Use 'is not equal to.....' condition

Task	What you need	Program script
Make the micro:bit count the number of times a button is pressed up to 7 times.		<b>m1</b> Count how many times button A gets pressed until the counter is greater than 7.
Show the number on the LEDs. When the count reaches 7, show an animated image of the number '7'.	<p><b>Simple Control</b> module 2</p> <p><b>Simple Control</b></p>	<b>m2</b> Show the LED number (counter) until the counter is equal to 7.  Show the LED animation until the counter is not equal to 7.
Use button B to stop the counting.	<p>Link button B to the counter module.</p>	<b>m1</b> Count how many times button A gets pressed until the counter is greater than 7 or button B gets pressed..

# 8

## MAGIC ARROW



### What's new:

Drag a line from UNTIL to WHEN to make WHILE  
Use 'tilt' gesture

Task	What you need	Program script
<p>Make the micro:bit display an arrow which always points upward, no matter which way you hold it in a vertical plane.</p>	<p>Link UNTIL to WHEN (gives WHILE)</p>	<p><b>m1</b> While the micro:bit is upright, show the LED image (up arrow).</p> <p><b>m2</b> While the micro:bit is inverted, show the LED image (down arrow).</p> <p><b>m3</b> While the micro:bit is tilted left, show the LED image (right arrow).</p> <p><b>m4</b> While the micro:bit is tilted right, show the LED image (left arrow).</p>

# 9



## SECRET MESSAGE

### What's new:

Use UNTIL 'shake' condition  
Use 'face down' gesture condition

Task	What you need	Program script
<p>Make the LEDs show an animation which changes to your secret message when the micro:bit is shaken.</p>	<p><b>+ Simple Control</b></p>	<p>Show the LED animation (screen wipe) until the micro:bit is shaken.</p> <p>Show the LED message "my secret message" for 11 seconds.</p>
<p>For extra secrecy, modify the second instruction so that the message only appears when the micro:bit is facing upside down.</p>	<p><b>+</b></p>	<p>When the micro:bit is face down, show the LED message "my secret message" for 11 seconds.</p>

# 10

## SPIRIT LEVEL



### What's new:

- Use 'Graph' display type (point) on LEDs
- Use a 'tilt' sensor (pitch & roll)

Task	What you need	Program script
Make the LEDs display a single point which behaves like a bull's eye spirit level for showing when the micro:bit is perfectly horizontal.	<p>X = roll, Y = pitch</p>	Plot the LED point graph (spirit level) until exit.

# 11

## NIGHT LIGHT



### What's new:

- Alter the brightness of a message

Task	What you need	Program script
Make the LEDs show a "Goodnight" message which gradually reduces in brightness after pressing a button.	<ul style="list-style-type: none"> <li>+ 3 Simple controls</li> <li>+ Pause instruction</li> </ul>	<p>Show the LED message " TO SLEEP, PRESS A" until button A gets pressed.</p> <p>Show the LED message "GOODNIGHT" for 6 seconds.</p> <p>Show the LED message "GOODNIGHT" at 5/10 brightness for 6 seconds.</p> <p>Show the LED message "GOODNIGHT" at 1/10 brightness for 6 seconds.</p> <p>Wait until button A gets pressed.</p>

# 12

## GUESS THE NUMBER



### What's new:

- Use a random number variable (Assign module)

Task	What you need	Program script
Make the micro:bit choose a random number and create a 'Guess the number' game. Press button A to have your guess. Press button B to see the answer.	<ul style="list-style-type: none"> <li>+ Simple Control</li> <li>+ Assign module (answer) [Set to Random number]</li> <li>+ Simple Control</li> </ul>	<p>Show the LED image (arrow) until button A gets pressed.</p> <p>Show the LED message "GUESS A NUMBER FROM 1 to 10" until button B gets pressed.</p> <p>Set answer to a random number from 1 to 10.</p> <p>Show the LED message (answer) for 5 seconds.</p>

# 13

ROCK  
PAPER  
SCISSORS



**What's new:**  
Use WHEN shaken condition

Task	What you need	Program script
<p>Create a 'Rock-Paper-Scissors' game by making the micro:bit choose a random number and display an image:</p> <p>rock      paper      scissors</p>	<p><b>Assign module:</b> Set to Random number.</p>	<p><b>m1</b> When the micro:bit is shaken, set variable to a random number from 1 to 3.</p> <p><b>m2</b> While the variable is equal to 1, show the LED image (rock).</p> <p><b>m3</b> While the variable is equal to 2, show the LED image (paper).</p> <p><b>m4</b> While the variable is equal to 3, show the LED image (scissors).</p>

# 14



COUNTING  
UP & DOWN

**What's new:**  
Use 2 Counter modules (up & down)

Task	What you need	Program script
<p>Make the micro:bit count how many times button A is pressed, but reduce the count each time button B is pressed. Set the count value to zero when the micro:bit is shaken.</p>	<p><b>Counter modules:</b> Link UNTIL to shaken sensor Link COUNTING to buttons and counter</p>	<p><b>m1</b> Count how many times button A gets pressed until the micro:bit is shaken.</p> <p><b>m2</b> Count (down) how many times button B gets pressed until the micro:bit is shaken.</p> <p><b>m3</b> Show the LED number (counter) until exit.</p>

# 15



LOVE METER

**What's new:**  
Use a 'touch' input  
Use 'Graph' display type (1 bar) on LEDs

Task	What you need	Program script
<p>Make the micro:bit show a bar graph when two people each touch an input pin PO and GND. The result is a random number, but the bar pretends to show how well-matched the people are when they hold hands.</p>	<p><b>Assign module:</b> Set to Random number. <b>LEDs:</b> Set Bar 1 to index and Maximum to 8. <b>+ Simple Control</b></p>	<p><b>m1</b> When the input gets connected, set index to a random number from 1 to 10</p> <p><b>m2</b> When the input gets connected, plot the LED bar graph (index) until the index is greater than 8.</p> <p>Show the LED message "MATCHED" for 5 seconds.</p>

# 16

ZOOMER



### What's new:

Use an 'acceleration' sensor

Use a Calculation module

Task	What you need	Program script
<p>Make the micro:bit measure acceleration while you hold it in your hand and swing your arm. Keep button A pressed until the moment you want to record the acceleration. (Use Transfer mode)</p> <p>Show the result when you press button B.</p>	<p>Calculation module: Link CALCULATION to data.</p>	<p><b>m1</b> When button A is pressed set data to acceleration.</p> <p><b>m2</b> Calculate result = data - 1000 until exit.</p> <p><b>m3</b> When button B gets pressed, show the LED number (result) until button B gets pressed again.</p>

# 17



TRUTH  
OR  
DARE

### What's new:

Use the 'Message array' display type on LEDs

Task	What you need	Program script
<p>Create a 'Truth or Dare' game by making the micro:bit choose a number from random and display the word 'TRUTH' or 'DARE'. The game starts when you press button A.</p>	<p><b>+</b> Assign module: Set to Random number. <b>+</b> Simple Control: Index = variable</p>	<p>Show the LED image (arrow) until button A gets pressed.</p> <p>Set variable to a random number from 1 to 2.</p> <p>Show the LED array message (Truth or Dare) until button A gets pressed.</p>

# 18



SPINNER

### What's new:

Use the 'Image array' display type on LEDs

Task	What you need	Program script
<p>Make the micro:bit show an animation when button A is pressed.</p> <p>Then an arrow points at random to one of four players sat around the micro:bit.</p>	<p><b>+</b> Assign module: Set to Random number. <b>+</b> Simple Control: Index = variable</p>	<p>When button A is pressed, show the LED animation for 2 seconds.</p> <p>Set variable to a random number from 1 to 4.</p> <p>Show the LED array image (variable) until button A gets pressed.</p>



# 19



## Revision:

Use a Gesture input (shaken)  
Use the 'Image array' display type on LEDs

Task	What you need	Program script
<p>Make the micro:bit behave like a die when rolled on to a table: At random it shows one of six possible images:</p>	<p><b>+</b> Simple Control: Index = variable</p>	<p>When the micro:bit is shaken, set variable to a random number from 1 to 6. Show the LED array image (variable) for 5 seconds.</p>

# 20



## KNOW YOUR TABLES

## Revision:

Use a Counter module  
Use a Calculation module

Task	What you need	Program script
<p>Make the micro:bit calculate multiplication tables by pressing the buttons: Button A chooses the times table; Button B steps through each line of the table.</p>	<p><b>+</b> Simple Control</p> <p><b>Calculation module:</b> Link CALCULATION to Counter 1 and Counter 2</p>	<p><b>m1</b> Count how many times button A gets pressed until counter 1 is greater than 10. <b>m2</b> Count how many times button B gets pressed until counter 2 is greater than 10. <b>m3</b> Calculate result = counter 1 x counter 2 until exit. <b>m4</b> Show the LED number (counter 1) until button B gets pressed. Show the LED number (result) until button A gets pressed.</p>

# 21



## DIGITAL PET

## Revision:

Make an animation  
Link UNTIL to WHEN to make WHILE

Task	What you need	Program script
<p>Make a micro:bit 'digital pet' which needs to be fed and allowed to sleep. When it gets hungry, feed it by pressing button A. If you stroke it by pressing button B, it will go to sleep for a while.</p>	<p><b>+</b> Signal Control: Link UNTIL to WHEN</p> <p>module 1: <b>+</b> Signal Control</p> <p>module 2: <b>+</b> Signal Control</p>	<p><b>m1</b> Show the LED image (awake) until button A gets pressed or button B gets pressed. While button A is pressed, show the LED animation (eating). <b>m2</b> When button B is pressed, show the LED image (asleep) for 10 seconds. Show the LED image (awake) until button A gets pressed or button B gets pressed.</p>

# 22

## SQUARES



### Revision:

- Use a Counter module
- Use a Calculation module

Task	What you need	Program script
<p>Make the micro:bit calculate the square of a number entered by pressing button A. The result is shown by pressing button B.</p>	<p><b>+</b> Simple Control</p>	<p><b>m1</b> Count how many times button A gets pressed until the counter is greater than 12.</p> <p><b>m2</b> Calculate result = counter x counter until exit.</p> <p><b>m3</b> Show the LED number (counter) until button B gets pressed.</p> <p>Show the LED number (result) until button A gets pressed.</p>

# 23

## SPEED BUTTON



### Revision:

- Use a Counter module
- Use an Assign module to set a variable

Task	What you need	Program script
<p>Make the micro:bit count how many times you can press button A in 5 seconds. You win a point if you can press it more than 15 times in the allotted time.</p>	<p>module 1: <b>+</b> Counter module</p> <p>module 2: <b>+</b> Assign module: Set variable to 1</p> <p>module 3: <b>+</b> Assign module: Set variable to 1</p>	<p><b>m1</b> After 5 seconds, show the LED image (arrow) until button A gets pressed.</p> <p>Count how many times button A gets pressed for 5 seconds.</p> <p>Set variable to 1.</p> <p><b>m2</b> When the variable is equal to 1 and the counter is less than 15, show the LED message "TOO SLOW" for 5.4 seconds.</p> <p>Set variable to 0.</p> <p><b>m3</b> When the counter is greater than 14, show the LED message "WIN" for 5 seconds.</p> <p>Set variable to 0.</p>

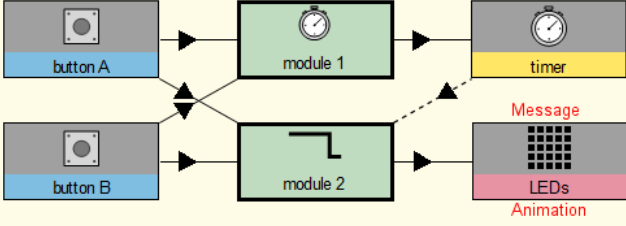


# 24



## STOPWATCH

### What's New?

Use a Timer module.

Task	What you need	Program script
<p>Make the micro:bit measure and display the time interval between two presses of button A. Button B resets the micro:bit ready for another measurement.</p> <p>====</p> <p>Press A to START or STOP timing.</p>	 <p>module 1:  <b>Pause</b> instruction</p> <p>module 2:  <b>2 Simple controls</b></p>	<p><b>m1</b> When button A gets pressed, measure the time until button A gets pressed again. Wait until button B gets pressed.</p> <p><b>m2</b> Show the LED message "Press A" until button A gets pressed. Show the LED animation until button A gets pressed. Show the LED message (timer) until button B gets pressed.</p>