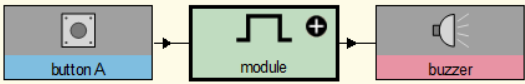# Mr Bit TASK GUIDES

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 *sequence* of instructions. When there is more than one module, the script for each is cued with the markers: m1, m2, m3 etc.

Example from Experiment 1 'Alarm Test'

Task	What you need	Program script
With a **buzzer** connected to the micro:bit pins, create a program to sound a series of beeps while you press a button.	⊕ **Pause** instruction	When button A is pressed, switch on the buzzer for 0.5 seconds. Wait for 0.5 seconds.
Check the **Connections** view.	Test in **Control** mode.	

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

EXPERIMENTS in CONTROL
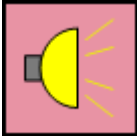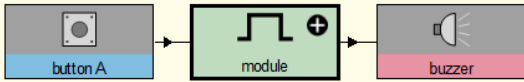
X1 — ALARM TEST

Revise:
UNTIL TIME condition
Wait with a Pause instruction

Task	What you need	Program script
With a **buzzer** connected to the micro:bit pins, create a program to sound a series of beeps while you press a button.	button A → module → buzzer ➕ **Pause** instruction	When button A is pressed, switch on the buzzer for 0.5 seconds. Wait for 0.5 seconds.

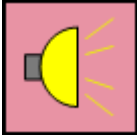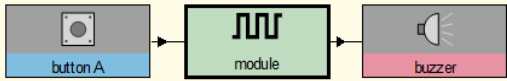Check the **Connections** view.   Test in **Control** mode.

X2 — TIME SIGNAL

Revise:
Use 'gets pressed' condition with a button
Use a 'Pulse Control' module
Adjust ON and OFF times for the pulse.

Task	What you need	Program script
With a **buzzer** connected to the micro:bit pins, create a program to sound six short beeps (like the time signal 'pips' on the radio) when a button gets pressed.	button A → module → buzzer Pulses: ON time = 0.2 seconds OFF time = 0.8 seconds	When button A gets pressed, pulse the buzzer 6 times.

Check the **Connections** view.   Test in **Control** mode.
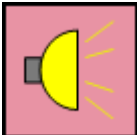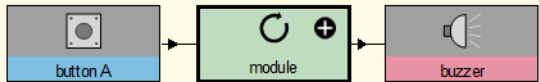
X3 — TELEPHONE CALL

Revise:
Use Repeat, Pulse Control and Pause modules.
Link UNTIL to WHEN to create WHILE.

Task	What you need	Program script
With a **buzzer** connected to the micro:bit pins, create a program to sound pairs of short beeps while a button is pressed.	button A → module → buzzer ➕ **Pulse Control** module ➕ **Pause** instruction	While button A is pressed, repeat: Pulse the buzzer twice. Wait for 1 second.

Check the **Connections** view.   Test in **Control** mode.

X4 — HAZARD WARNING

Revise:
UNTIL TIME condition
Wait with a Pause instruction

Task	What you need	Program script
With a **yellow LED** connected to the micro:bit, create a program to make it flash on and off continuously.	module → Yellow LED ➕ **Pause** instruction	Switch on the Yellow LED for 0.5 seconds. Wait for 0.5 seconds.

Check the **Connections** view.   Test in **Control** mode.

More things to do: Alter the times to make the LED flash faster or slower. How fast can you make it flash?
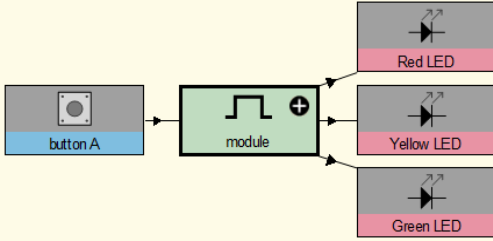
X5 — COLOUR CODE

Revise:
Make a sequence of instructions
.

Task	What you need	Program script
With a **three LEDs** (red, yellow, green) connected to the micro:bit pins, create a program to light up each LED in a rapid sequence when button A is pressed.	button A → module → Red LED, Yellow LED, Green LED ➕ 3 **Simple Control** modules	When button A is pressed, switch on the Red LED for 0.2 seconds. Switch on the Yellow LED for 0.2 seconds. Switch on the Green LED for 0.2 seconds. Switch on the Yellow LED for 0.2 seconds.

Check the **Connections** view.   Test in **Control** mode.

More things to do: Add a duplicate system but with button B as the input. Make up your own crazy sequence.
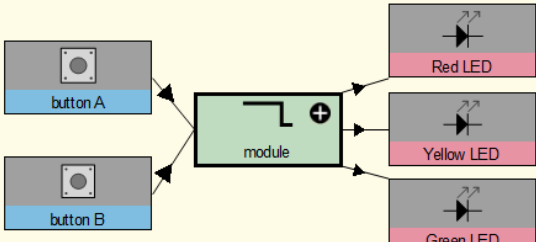
X6 — TRAFFIC LIGHTS

Revise:
Make a sequence of instructions.
Use UNTIL with a button condition.

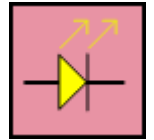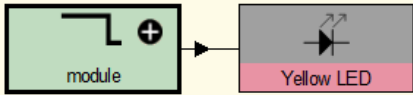| Task	What you need	Program script
With a **three LEDs** (red, yellow, green) connected to the micro:bit, create a program to light up each LED in the correct sequence for traffic lights. Make button A change the lights to red and button B to change them to green.	button A, button B → module → Red LED, Yellow LED, Green LED ➕ 3 **Simple Control** modules	Switch on the Red LED until button A gets pressed. Switch on the Red LED and the Yellow LED for 2 seconds. Switch on the Green LED until button B gets pressed. Switch on the Yellow LED for 2 seconds.

Check the **Connections** view.   Test in **Control** mode.

More things to do: Change the second instruction to a Pulse Control to make the yellow flash as it does in a Pelican Crossing.

X7 — LIGHT METER

What's new:
Show a number by linking OUTPUTS first to the LEDs then to the sensor.

Task	What you need	Program script
With a **Light sensor** connected to the micro:bit pins, create a program to show the light level value on the LEDs.	light level → module → Number LEDs	Show the LED number (light level) until exit.

Check the **Connections** view.　　Test in **Control** mode as you illuminate or shade the sensor.

More things to do: Add a button to the program so that the number only shows while the button is pressed.

X8 — LIGHT REPORT

Revise:
Link UNTIL to WHEN to create WHILE. Detect light and dark by comparing the light level with a threshold value.

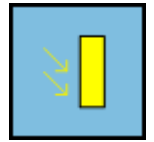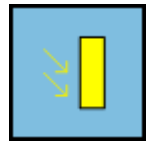| Task	What you need	Program script
With a **Light sensor** connected to the micro:bit pins, create a program to show a message on the LEDs: Show "DARK" when the light level is below a certain value, and "LIGHT" when the light is above the same value.	light level → module 1 / module 2 → Message LEDs	**m1** While it is light, show the LED message "LIGHT". **m2** While it is dark, show the LED message "DARK".

Check the **Connections** view.　　Test in **Control** mode as you illuminate or shade the sensor.

More things to do: The threshold value for light and dark is set at 50. You can alter this in **Design** mode when you double click the sensor box.

X9 — LIGHT GRAPH
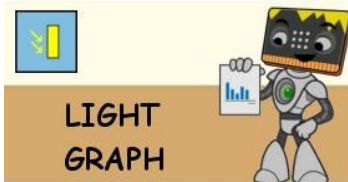
What's new:
Link OUTPUTS to LEDs with
Display type Graph, Graph type 1 bar
Select 1Bar: Light level, Maximum: 50
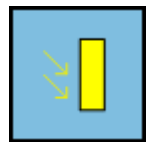
Task	What you need	Program script
With a **Light sensor** connected to the micro:bit pins, create a program to show the light level as a bar graph on the LEDs.	light level → module → Graph LEDs	Plot the LED bar graph (light level) until exit.

Check the **Connections** view.　　Test in **Control** mode as you illuminate or shade the sensor.

More things to do: You can adjust the graph maximum to change the sensitivity.

X10 SHADOW COUNTER

Revise:
Use a Counter module to detect changes in light level.

Task	What you need	Program script
With a **Light sensor** connected to the micro:bit pins, create a program to count the number of times light falling on the sensor becomes blocked by a shadow. The number is shown on the LEDs. Button A resets the counter.	button A → module 1 → counter light level → module 2 → LEDs (Number)	**m1** Count how many times it gets darker until button A gets pressed. **m2** Show the LED number (counter) until exit.

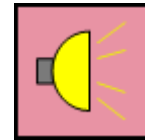Check the **Connections** view.   Test in **Control** mode as you pass your hand in front of the sensor.

X11 WHO IS THERE?

What's new:
Use a light sensor to control a buzzer.

Task	What you need	Program script
With a **Light sensor** and a **buzzer** connected to the micro:bit pins, create a program to sound the buzzer when the light falling on the sensor becomes blocked by a shadow.	light level → module → buzzer ➕ **Pause** instruction	When it is darker than 50, switch on the buzzer for 0.5 seconds. Wait for 1 second.

Check the **Connections** view.   Test in **Control** mode as you pass your hand in front of the sensor.

More things to do: Add a module to show the light level on the LEDs.

X12 PERFECT LIGHT

What's new:
Use a light sensor to control coloured LEDs.

Task	What you need	Program script
With a **Light sensor** and **yellow** and **red LEDs** connected to the micro:bit pins, create a program that lights up the red LED when the light level is below 50% OR above 60% brightness, but lights up the yellow LED when the light level is between 50% and 60%.	light level → module 1 → Red LED light level → module 2 → Yellow LED	**m1** While it is darker than 50 or lighter than 60, switch on the Red LED. **m2** While it is lighter than 50 and darker than 60, switch on the Yellow LED.

Check the **Connections** view.   Test in **Control** mode as you point the sensor in different directions.

X13 DIGITAL THERMOMETER

What's new:
Use a temperature sensor.

Task	What you need	Program script
With a **Temperature sensor** connected to the micro:bit pins, create a program to show the temperature value on the LEDs.	temperature → module → LEDs (Number)	Show the LED number (temperature) until exit.

Check the **Connections** view.   Test in **Control** mode as you touch the sensor.
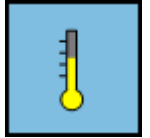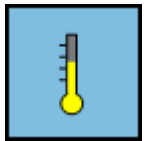
More things to do: To cool the sensor down, touch it with a wet cloth or paper tissue.


X14 CLIMATE MESSAGE

What's new:
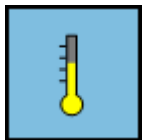Detect hot and cold by comparing temperature with a threshold value.

Task	What you need	Program script
With a **Temperature sensor** connected to the micro:bit pins, create a program to show a message on the LEDs: Show "COLD" when the temperature level is below a certain value, and "HOT" when the temperature is above the same value.	temperature → module 1 / module 2 → LEDs (Message)	**m1** While it is hot, show the LED message "HOT". **m2** While it is cold, show the LED message "COLD".

Check the **Connections** view.   Test in **Control** mode as you touch the sensor.

More things to do: The threshold value for hot and cold is set at 20. You can alter this in **Design** mode when you double click the sensor box.

X15 TEMPERATURE GRAPH
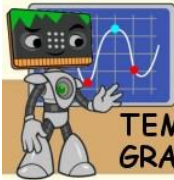
What's new:
Show a graph line on the LEDs to indicate temperature.

Task	What you need	Program script
With a **Temperature sensor** connected to the micro:bit pins, create a program to show the temperature as a line graph on the LEDs.	temperature → module → LEDs (Graph)	Plot the LED trace graph (temperature) until exit.

Check the **Connections** view.   Test in **Control** mode as you touch the sensor.
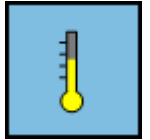
X16

WEATHER REPORT

What's new:
Combine two temperature values with AND logic to specify a condition.

Task	What you need	Program script
With a **Temperature sensor** connected to the micro:bit pins, create a program to show a message on the LEDs: Show "COLD", "COOL", "NICE", "WARM" or "HOT" according to the temperature measurement between 10 and 30 celsius.	module 1, module 2, temperature, module 3, LEDs (Message), module 4, module 5	**m1** While it is warmer than 30, show the LED message "HOT". **m2** While it is cooler than 30 and warmer than 25, show the LED message "WARM". **m3** While it is cooler than 25 and warmer than 20, show the LED message "NICE". **m4** While it is cooler than 20 and warmer than 10, show the LED message "COOL". **m5** While it is cooler than 10, show the LED message "COLD".

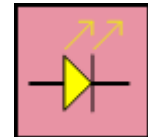Check the **Connections** view. Test in **Control** mode as you touch the sensor.

X17

COMFORT ZONE

What's new:
Use a temperature sensor to control coloured LEDs.

Task	What you need	Program script
With a **Temperature sensor, red** and **green LEDs** connected to the micro:bit pins, create a program to light up the red LED when it is too hot or cold, and to light up the green LED when the temperature is comfortable.	temperature, module 1, Green LED, module 2, Red LED	**m1** While it is warmer than 20 and cooler than 25, switch on the Green LED. **m2** While it is warmer than 25 or cooler than 20, switch on the Red LED.

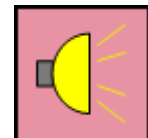Check the **Connections** view. Test in **Control** mode as you touch the sensor.

X18

EXTREME ALERT

What's new:
Use a temperature sensor to control a buzzer.

Task	What you need	Program script
With a **Temperature sensor** and a **buzzer** connected to the micro:bit pins, create a program to sound the buzzer when the temperature gets too cold or too hot.	temperature, module 1, module 2, buzzer. **Pause** instruction (module 1)	**m1** When it is cooler than 20, switch on the buzzer for 0.5 seconds. Wait for 1 second. **m2** When it is warmer than 25, switch on the buzzer for 2 seconds. Repeat.

Check the **Connections** view. Test in **Control** mode as you point the sensor in different directions.