

## Introduction to Computer Science with micro:bit have-a-go activities

Welcome from the Arm School Program! This document contains tasks that relate to our video playlist here:

[https://www.youtube.com/playlist?list=PLXwAdcOI0ICrPbDxaRtnUFC0hkFGKdG\\_G](https://www.youtube.com/playlist?list=PLXwAdcOI0ICrPbDxaRtnUFC0hkFGKdG_G)

### Module 1: Getting Started

- Plug in the micro:bit
- Drag the hex file to the micro:bit folder
- Click download
- Drag the blocks onto the code area
- Give the file a name
- Create a new project
- Test the program on the virtual micro:bit

### Module 2: First Programs

- Write a program to display a heart when the micro:bit loads up
- Change this program so it displays first a heart then a happy face when the micro:bit loads up
- Write a program that doesn't display an image when it starts, but then repeatedly (forever) displays a happy face, then a square, then a t-shirt
- Change the order in the program so it displays a t-shirt, then a square, then a happy face
- Write a program to display an icon of your choice when the program starts. Then edit the program so it repeatedly display four different images, one after the other
- Write a program that repeatedly (forever) turns on all the LEDs then turns off the LEDs
- Edit the program so it turns on the LEDs one at a time, and then turns them off one at a time
- Write a program that repeatedly displays each letter of your name, one at a time
- Write a program that makes a dot move across each row of the LEDs, one LED at a time

### Module 3: LEDs

- Write a program that displays the first letter of your name. Edit the program so it displays one LED at a time.
- Write a program that makes a sprite move around the LEDs as though it is running
- Edit this program so that it appears as though another LED is chasing it around the micro:bit LED display
- Write a program that makes the first LED turn on, then off, then on etc. repeatedly, changing every 1 second
- Write a program that toggles all of the LEDs on and off, waiting half a second before changing them all each time
- Write a program that makes it appear as though an LED moves across the first row, waiting for 2 seconds between movement

## Module 4: Inputs

- Write a program that outputs a happy face when button **a** is pressed, and an unhappy face when button **b** is pressed
- Edit the program so a sequence of images is displayed when each button is pressed
- Write a program that outputs the letter of the button that has been pressed i.e. A or B
- Edit the program so that when neither button is pressed (on start), all the LEDs are turned on

## Module 5: Data Types

- Write a program that outputs your name
- Write a program that counts from 10 down to 0 and then says “Blast Off”

## Module 6: Variables and Mathematical Operators

- Write a program that displays numbers starting at 0 and increasing by 10 each time
- Write a program that displays the 5 times table, i.e. 5 then 10, 15, 20 etc.
- Write a program that starts with 1, then doubles each time and displays the number i.e. 1, 2, 4, 8, 16, 32 etc.
- Write a program that outputs the result of:
  - 136 plus 25
  - 136 minus 25
  - 136 multiplied by 25
  - 136 to the power of 25
- Write a program that outputs  $2 ** 2$  when button a is pressed
- Write a program that outputs  $10 ** 2$  when button b is pressed
- Write a program that starts with 2 in the variable. Multiply the value in the variable by itself forever, outputting the value each time
- Write a program that starts with 2 in the variable. Calculate the variable to the power of 2, and store this in the variable. Do this forever, outputting the value each time.
- Write a program that starts with the number 1 in a variable. When button a is pressed, multiply this value by 2, store it in the variable and output it. When button **a** is pressed, calculate the variable to the power of 2, store it in the variable and output it

## Module 7: Selection & Boolean

- Write a program that outputs the number 1 if a variable stores a number less than 100, and the number 0 otherwise
- Write a program that adds 1 to a variable each time button **a** is clicked. Make sure it:
  - It continually (forever) checks the value of this variable
  - If the number of presses is less than 10 it outputs a cross
  - If the number of presses is 10 or more, it outputs a tick
- Write a program that records the number of times button **b** has been pressed. If it has not been pressed (0 times), output a confused face. If it is less than 10 times, output an unhappy face. If it is 10 or more times, output a happy face

- Edit the program so it also records the number of times button **a** is pressed. Output the letter of the button that has been pressed more, i.e. A if **a** is pressed more. If both are equal, turn all of the LEDs on

## Module 8: Boolean Operators

- Write a program that counts the number of times button **a** and **b** are pressed
- Output an image when neither have been pressed
- Output an image when button **a** and button **b** have both been pressed at least once, but less than 10 times
- Output an image when one of the buttons has been pressed more than 10 times
- Output an image when both of the buttons have been pressed more than 15 times

## Module 9: Random Values

- Write a program that generates six lottery numbers between 0 and 49 and outputs each one
- Write a program that acts like a magic 8 ball. The user clicks button **a** and it outputs either:
  - Yes
  - No
  - Maybe
  - Try again later

## Module 10: Sensors – Temperature

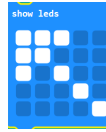
- Write a program that continually (forever) outputs the current temperature
- Write a program that converts the current temperature to Fahrenheit (you will need to research how to do this) and output it
- Write a program so that:
  - When button **a** is pressed the temperature is output in Celsius
  - When button **b** is pressed the temperature is output in Fahrenheit
- Write a program that outputs a different message (or icon) depending on the current temperature. Include at least four different options

## Module 11: Light Sensors

- Write a program that continually checks the light level and temperature
- It should output:
  - Very bright and hot, when light level is above 200 and temperature is above 25
  - Bright and hot, when light level is between 140 and 200 and temperature is above 25
  - Dark and hot, when light level is 140 or below and temperature is above 25
  - Very bright and warm, when light level is above 200 and temperature is between 17 and 25
  - Bright and warm, when light level is between 140 and 200 and temperature is between 17 and 25
  - Dark and warm, when light level is 140 or below and temperature is between 17 and 25
  - Very bright and cold, when light level is above 200 and temperature is below 17
  - Bright and cold, when light level is between 140 and 200 and temperature is below 17
  - Dark and cold, when light level is 140 or below and temperature is below 17

## Module 12: Accelerometer

- Write a program that outputs a left arrow when the micro:bit is tilted left, and a right arrow when the micro:bit is tilted right
- Write a program that outputs an up arrow when the micro:bit is tilted away from you, and a down arrow when the micro:bit is tilted towards you
- Edit the program, so that when the micro:bit is tilted to the left and away from you, this arrow is displayed. Repeat for the other 3 corners



- Edit the program so if the micro:bit is moving upwards, it outputs “Flying” and if it is moving down it outputs “Falling”

## Module 13: Compass

- Write a program that outputs N, S, E or W when facing approximately in those directions. You will need to split the 360 degrees into equal segments
- Edit the program so it also displays NE, NW, SE and SW. You will need to split the 360 into smaller segments
- Write a program that directs people which way they need to face. For example, first they need to face north – when they face approximately north, give them another instruction such as face south etc.

## Module 14: On Movement

- Write a program that outputs the light level each time the micro:bit is shaken
- Write a program that outputs a description of the temperature each time the micro:bit is tilted (left or right) e.g. “Gosh it’s hot today”
- Write a program that outputs a sleeping face when the micro:bit is face down, and a happy face when it is face up
- Write a program that begins with the centre column of LEDs lit up. When the micro:bit is tilted left the LEDs move to the left, when it is tilted right they move to the right

## Module 15: Input Pins

- Write a program that outputs the temperature when pin **0** is pressed, and the light level when pin **1** is pressed.
- Output a maths question where the answer is 0, 1 or 2. When the user clicks on that pin, output if they were correct or not
- Create a game of choices, where users are given part of a story and have to choose which action to take by clicking on one of the pins

## Module 16: Output Pins

- Create a siren that continually plays for 1 second then stops for 1 second, then plays again
- Create a temperature alert, when the temperature gets above a certain temperature, or below a certain temperature output the buzzer

- Create a counting system: each time the user presses a button it adds 1 to a total. Every time the total is a multiple of 10 output a buzzer sound

## Module 17: Loops

- Write a program that counts from 1 to 10
- Write a program that outputs the 2 times table (a variable will need to count and then be multiplied before being output)
- Write a program that outputs a buzzer sound for 2 seconds, 15 times
- Write a program where when the user clicks on button **a** an animation starts and repeats forever
- Edit the program so that it stops when the user clicks button **b** and a different animation starts looping continually. This should stop when the user clicks button **b** again
- Write a program that challenges the user to press button **a** when the letter A appears on screen. The program should output random letters continually. When the user clicks button **a** the program should stop if they pressed it when A was on screen, otherwise it continues

**Hint:** You will need a variable to keep track of which letter is currently output, then check what this is when the button is pressed

## Module 18: Functions

- Create an animation using the LEDs. Make each part of the animation a separate function
- Write a program that replicates a Bop-it game ([https://en.wikipedia.org/wiki/Bop\\_It](https://en.wikipedia.org/wiki/Bop_It)). The user has to either shake it, tilt left, tilt right, drop it, flip it (turn it upside down)

**Hint:** You will need a function that generates the next action using a random number. You will also need a variable that records the number of the action e.g. *shake it* is number 0. This variable can be checked in each of the actions to make sure it was correct

- Extend the Bob-it game so that it counts how many actions you got correct before you went wrong
- Extend the example quiz program, so that there are 10 questions and each one plays the buzzer if incorrect. After question 1, each subsequent question should be its own function that is called from the button presses

## Project 1 – Button presses game

Write a program where the micro:bit displays a different message depending on which button(s) the user has clicked (i.e. button **a**, or **b**, or **a** and **b** together)

Before you start the project you need to plan it.

### Planning

- Write down the inputs your program will need
- Write down the outputs your program will need
- Write down which output happens when each input takes place
- When you have planned your project you need to write it, then test it

## Testing

- You need to make sure your program fully works
- Check that every input you wrote down produces the correct output by performing each input

## Project 2 – Maths Quiz

Write a program where:

- Each time the micro:bit is shaken, it generates two random numbers between 1 and 10 e.g. 2 and 4
- The micro:bit outputs this as an addition e.g. 2 + 4
- The user then has to click button **a** the same number of times as the answer, in this case 6 times
- The user should then click button **b** to say that they have finished
- The micro:bit should then output if they are correct or not

## Planning

- Write down the inputs your program will need
- Write down the outputs your program will need
- Write down the processes (the calculations e.g. generating a random number, counting the number of times a button is pressed)
- When you have planned your project you need to write it, then test it

## Testing

- You need to make sure your program fully works
- You need to check if you get the answer correct, it outputs this
- ...and if you get it incorrect it outputs this also

## Challenge

Extend the program so it also randomly chooses between addition, subtraction and multiplication.

### Hint:

- with multiplication you will need to be careful the number generated is not too large
- with subtraction you will need to be careful the number generated is not negative

## Project 3 - Maze

Write a program, where the user has to travel through a maze e.g.

- The user presses button **a** to load a maze
- The user controls an LED that starts at the beginning of the maze
- The user tilts the micro:bit to move the lit LED
- The maze should be different depending on:
  - the temperature
  - the light level
  - the direction the micro:bit is pointing in
- The user should be able to restart a maze by shaking the micro:bit

## Planning

- Write down the inputs your program will need
- Write down the outputs your program will need
- Write down the processes (e.g. calculating which maze to display, moving the LED)
- When you have planned your project you need to write it, then test it

## Testing

- You need to make sure your program fully works
- You need to check all the different mazes that you programmed