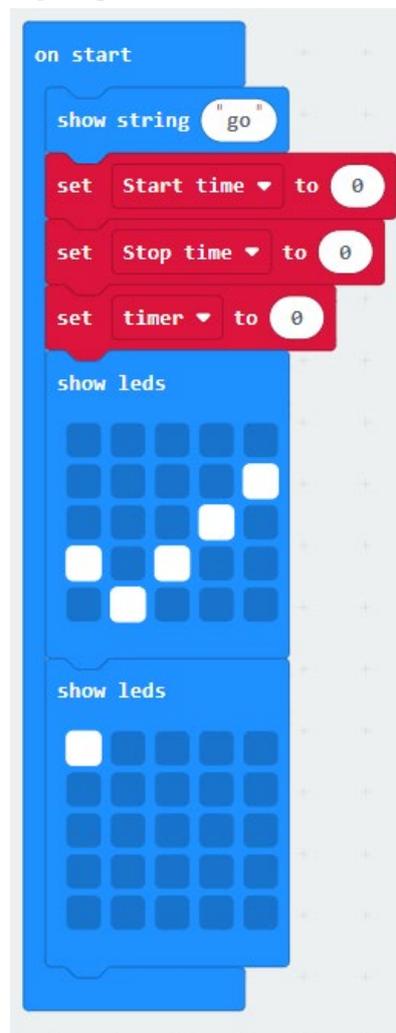


Detailed instructions for creating the timer program

Introducing block coding: using the BBC micro:bit in the science classroom

Code block 1: Starting the program and initializing variables



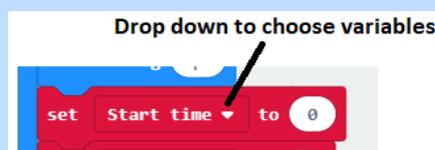
Starting the program; initializing the variables
Image courtesy of G. Michael Bowen

To create this code block follow these steps:

1. When you start the MakeCode tool, the 'on start' block should already be in the code area (but it is also under 'Basic' in the menu), and it has no other code blocks (which are

- instructions) within its 'jaws', so nothing will be happening on the emulator screen. You need to drag over other code and drop it into its jaw.
- The first block command to drag over and drop into the 'on start' jaw is 'show string', which is an optional command (found under 'Basic') that scrolls instructions across the matrix (our example shows the word 'go'). If you do use it, any text you place in the oval on the right of the block will scroll across the 5 × 5 matrix when the micro:bit starts.
 - Because the timer uses three variables, you will need to create them (a variable is essentially a container that holds information that is used by other code blocks). First click on 'Variables' in the menu, then on 'make a variable', enter the words 'start time', and click 'OK'. You have now created the variable 'start time', which will hold the running time (in ms) since the micro:bit was started. You need to do this two more times, entering 'stop time' and 'timer' to create those variables. Now that you've created variables, new menu items will show up under 'Variables'.
 - When a program starts, you need to set (or 'initialize' in coding speak) the variables, so the program knows what the variable starting values are. To do this for your three variables, under 'Variables', click on and drag the code block that starts with 'set' and drop it in the 'on start' jaws. Use the drop-down triangle to choose one of the variables, and then repeat this for the other two variables. Leave the variable value (showing in the oval) at a value of zero. Now the variables 'start time', 'stop time', and 'timer' are initialized.

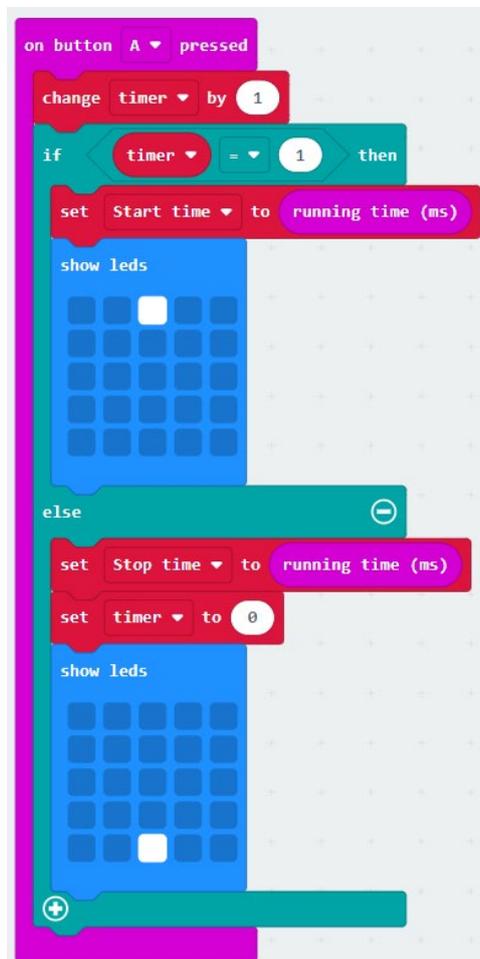
<CODING TIP> When working with more than one variable, you choose which variable you are designating by clicking the down triangle/arrow beside the variable name.



Showing the drop-down menu
Image courtesy of G. Michael Bowen

- Finally, you want to let the user know that the program is ready to accept input using the 5 × 5 LED matrix, so two 'show led' blocks are dragged over from 'Basic' and dropped in the jaws. Click on the dark-blue boxes in each 'show led' block to draw a check mark on the first and just one top-left dot in the second. In use, a check mark is first shown, so that the user knows that everything is functional, and then a dot on the top left indicates that the micro:bit is waiting for input.

Code block 2: Starting and stopping the timer function



Recording the start and stop times in variables
Image courtesy of G. Michael Bowen

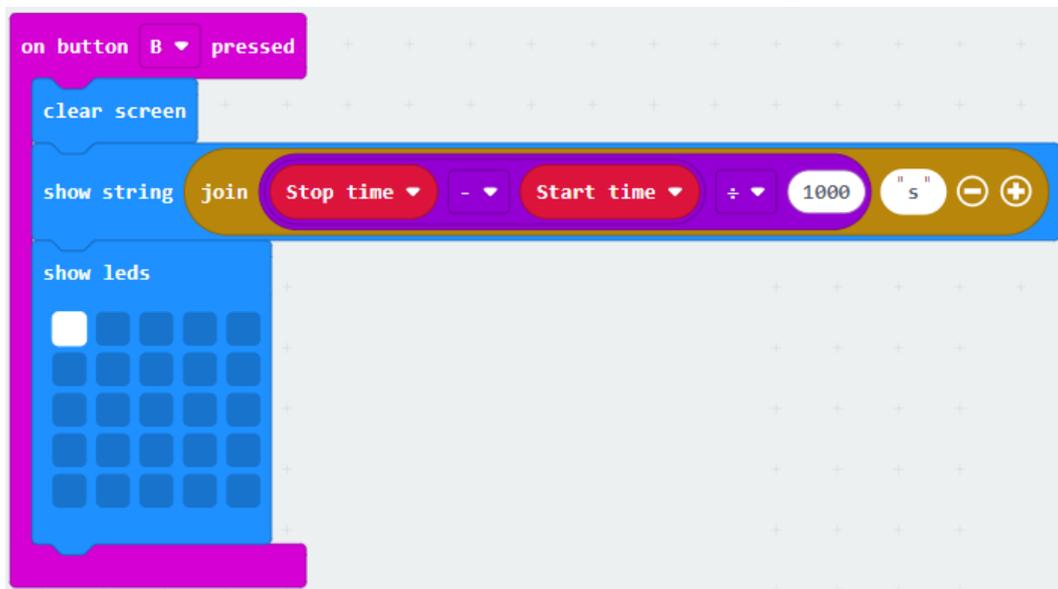
To create this code block follow these steps:

When button A is pressed, the value of the count variable is increased by 1, a dot briefly shows in the middle of the matrix, and then the top-left dot then appears to indicate that the program is waiting for input.

1. Drag over the 'on button' code block found under 'Input' and drop it in the code area. Make sure button A is indicated (the little white triangle shows the drop-down menu to make that choice).
2. Click on 'Variables' and drag over the 'change' code block and drop it in the 'on button' jaws. Make sure the variable 'timer' is shown (again, a drop-down menu is shown by clicking on the white triangle) and that the value is set at 1. Note that this code block option only shows in the Variables menu when a variable has been created.

3. Click on Logic and drag over the second 'if' block (the one that shows 'if' and 'else') and drop it in the button A jaws. Then under 'comparison', drag the first elongated hexagon (showing $0 = 0$) over the 'if' block and drop it into the blank area that it fits. Under Variables, drag the red oval with a variable listed in it over to the first oval in the elongated hexagon (where the first 0 is) and drop it into the area that it fits. Use the drop-down triangle to choose the variable 'timer'. Change the 0 in the second oval to the number 1.
4. Under 'Variables', drag a 'set' block into the first jaw in the if block. Use the drop-down menu to change the variable to 'start time'. Under 'Input-more', drag the oval 'running time (ms)' into the blank oval in the set block.
5. Under 'Basic', drag 'show leds' under the set block in the first jaw and then click on the middle-top block to make it light up.
6. Under 'Variables', drag a 'set' block into the second jaw in the 'if' block. Use the drop-down menu to change the variable to 'stop time'. Under 'Input-more', drag the oval 'running time (ms)' into the blank oval in the 'set' block.
7. Under 'Variables', drag a 'set' block into the second jaw in the 'if' block. Use the drop-down menu to change the variable to 'timer'. Make sure the value 0 is in the oval. [Hint: leaving this line of code out allows the timer to show lap times.]
8. Under 'Basic', drag 'show leds' under the 'set' block in the second jaw and then click on the middle-bottom block to make it light up.

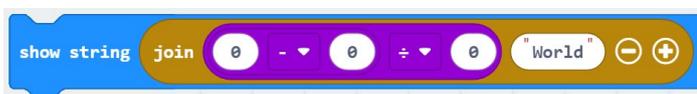
Code block 3: The output: calculate and show the elapsed time



Calculating and showing the elapsed time
Image courtesy of G. Michael Bowen

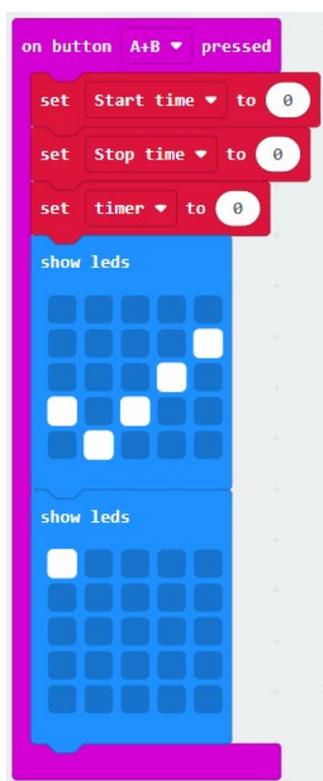
This column is the code for button B input, which shows the elapsed time. When button B is pressed, the matrix is blanked; the calculated elapsed time scrolls across the screen; and then the top-left dot shows to indicate that the micro:bit is, again, waiting for input.

1. Drag an 'on button' code block (under 'Input') and drop it in the code area. Click on the white triangle in the 'on button' code block and choose 'B' for button B.
2. Drag a 'show string' code block (under 'Basic') and drop it in the 'on button' jaws. Under 'Advanced – Text', click on the 'join' oval and drag it into the 'show string' block.
3. Under the 'Math' option, choose the fourth purple oval (with the divide symbol) and drag it into the first empty oval of the 'join' oval. Then drag over the second purple oval (with the minus sign) and drop it into the first oval before the divide symbol. You now have a purple oval embedded into another purple oval. It looks like the formula $0 - 0 \div 0$. [Hint: sometimes it helps to embed the second purple oval into the first purple oval before that one is placed in the first join oval.]



4. Under 'Variables', drag the 'stop time' oval into the first '0' oval (in the purple oval area), 'start time' into the second '0' oval, and enter 1000 after the divide sign (this converts milliseconds to seconds). Enter the letter s where the word 'world' appears (without quotation marks, the makecode tool will add them when you press 'enter').
5. Under 'Basic', drag 'show leds' under the 'show string' block and then click on the top-left block to make it light up.

Code block 4: Reset variables to zero



Reinitializing the variables

Image courtesy of G. Michael Bowen



The fourth code block column reinitializes the timer variables to zero when buttons A and B are pressed simultaneously and then lights the top-left LED to indicate that the micro:bit is awaiting input.

Use the drag-and-drop coding skills you used to code blocks 2 and 1 to create this block of code. Some hints:

- You need, in order, the 'Input', 'Variables', and 'Basic' code areas.
- You need to use the drop-down triangle to choose the 'A+B' button input option in the 'on button' block.
- Look at the instructions to code block 1 if you get stuck on finding the code blocks you need.