

KidzCanCode

IGNITE, INSPIRE, INNOVATE

www.kidzcancode.com

Build a Stop Watch Timer

KIDZCANCODE

TREVOR WARREN

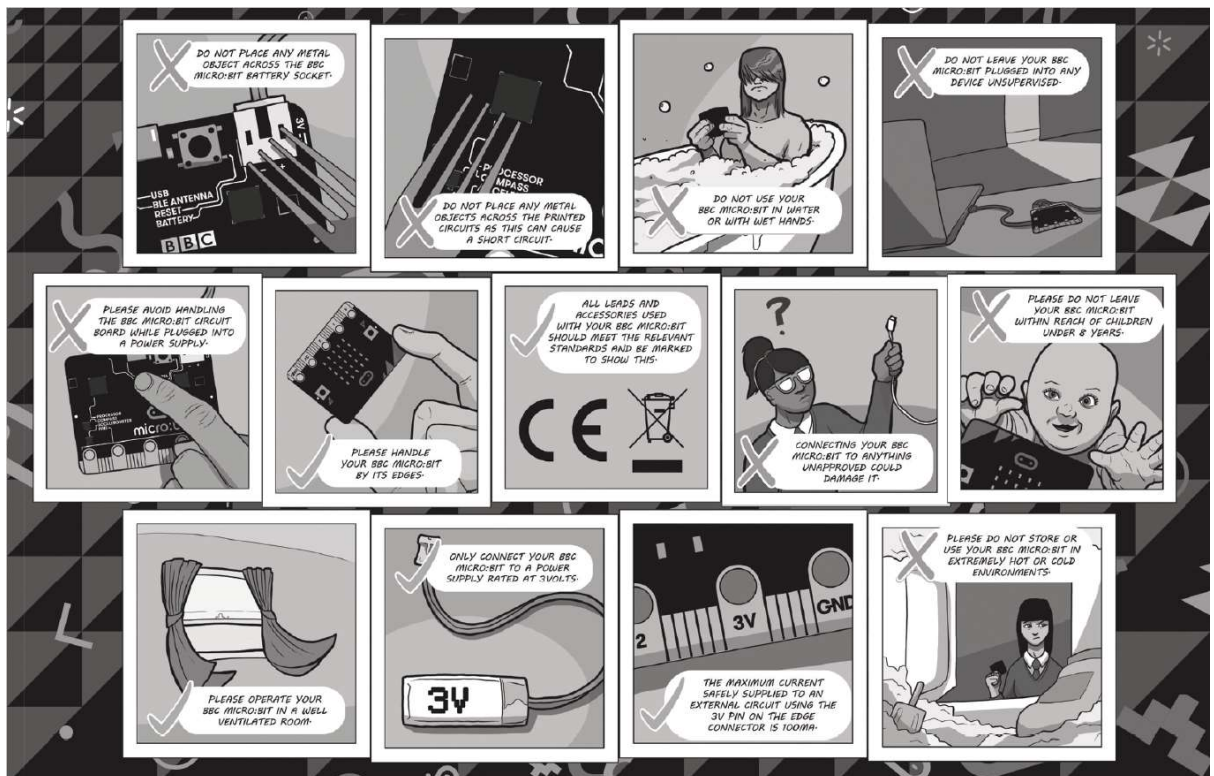
Tutorial 2.11 – Build A Stop Watch Timer

1. Safety Warnings

THE MICRO:BIT IS AN EXPOSED BOARD, TO BE USED WITH CARE PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE. You can read the detailed document at - <http://microbit.org/guide/safety-advice/>

a. General Safety Warnings

Using the BBC micro:bit is easy to use but is designed to have all the electrical parts on display. This does mean there's a small risk that the parts can be damaged and even overheat with a risk of injury but a little bit of care and caution will ensure you and your micro:bit will stay fit and healthy.

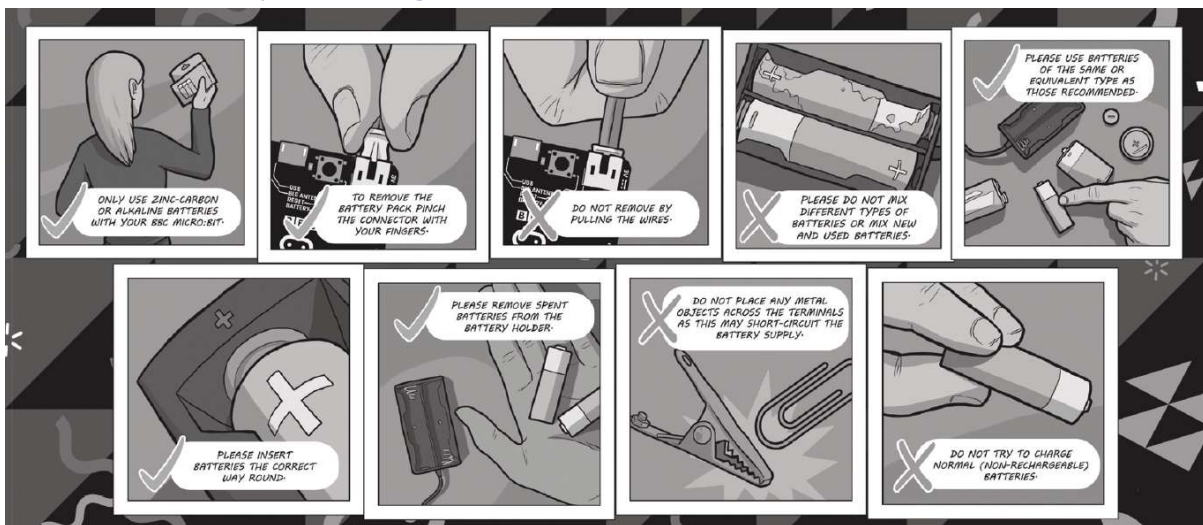


1. Always keep your BBC micro:bit in the anti-static bag when not in use. It's good practice for students to earth themselves before handling it.
2. Please handle your BBC micro:bit by its edges. This minimises the risk of damage through an electrostatic discharge.
3. Please use the battery pack and the USB lead provided to power your micro:bit. Do not use portable battery chargers or USB charging ports (often marked with a lightning bolt or 'SS'), to power your micro:bit. Using these may damage your micro:bit and stop it working properly.
4. Please avoid handling the BBC micro:bit circuit board while plugged into a power supply.
5. All peripherals (for example: USB cable, battery holder, sensors) used with your BBC micro:bit should comply with the relevant standards and should be marked accordingly.
6. Connecting your BBC micro:bit to any unapproved peripherals could damage your BBC micro:bit
7. Please do not attempt to keep using faulty micro:bits. If a school-issued micro:bit develops a fault, contact the vendor immediately.

Tutorial 2.1I – Build A Stop Watch Timer

8. The maximum current safely supplied to an external circuit using the 3V pin on the edge connector is 100mA. Please make sure this limit is not exceeded.
9. Please do not store or use your BBC micro:bit in extremely hot or cold environments.
10. Do not place any metal objects across the printed circuits on the board as this can cause a short circuit damaging your BBC micro:bit. This can cause risk of burn or fire.
11. Do not use your BBC micro:bit in water or with wet hands.
12. Do not leave your BBC micro:bit plugged into a computer or any other device unsupervised.
13. Please do not leave your BBC micro:bit within reach of children under 8 years of age.
14. Please operate your BBC micro:bit in a well ventilated room To remove the battery pack, pinch the connector with your fingers. Do not remove by pulling the wires.

b. Battery Warnings



1. Do not try to charge normal (non-rechargeable) batteries
2. Please do not mix different types of batteries or mix new and used batteries.
3. Please use batteries of the same or equivalent type as those recommended.
4. Please insert batteries the correct way round (with the correct polarity).
5. Please remove spent batteries from the battery holder.
6. Do not short-circuit the battery supply terminals, for example by placing a metal object across the terminals.
7. Only use Zinc or Alkaline batteries with your BBC micro:bit.
8. Please do not use rechargeable batteries

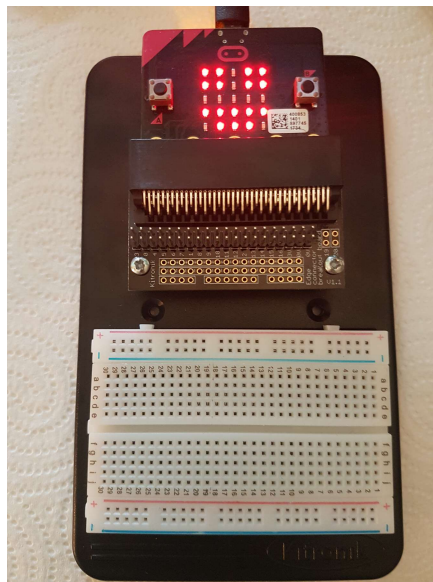
Tutorial 2.1I – Build A Stop Watch Timer

2. Pre-requisites

If you have questions with the assembly of the micro:bit, edge connector breakout board, mounting board and the breadboard please drop us a note at help@kidzcancode.com. The edge connector board, mounting board and the breadboard are part of the Kitronix Inventors kit which needs to be purchased separately.

To be able to perform this tutorial you will need the following components –

1. Parts required –
 - a. 1 x BBC Micro:bit
 - b. 1 x Mounting Plate
 - c. 1 x Edge connector breakout board
 - d. 1 x Bread board
2. Assembly required –
 - a. Bread board mounted on top of the mounting plate
 - b. BBC Micro:bit inserted into the Edge Connector breakout board



Before proceeding please check your setup and confirm that all the required parts are configured as demonstrated in the above picture.

Tutorial 2.1I – Build A Stop Watch Timer

3. Learning Objectives

The objectives of this tutorial are to introduce the student to the following concepts –

- Declaring variables and assigning values to variables at different times of execution in the program
- Understanding the use of Boolean variables and implementing them in a program
- Coding for use of the buttons on the micro:bit, perform a given action when a button is pressed.
- Calling code in a recursive manner using the forever code block

The BBC micro:bit is a powerful little computer. Through programming these games kids explore more advanced computer science concepts. Along the way kids are encouraged to share, create and extend the games using their own imagination and creativity.

This tutorial builds upon concepts introduced in previous tutorials so please make sure you have covered the previous tutorials before you dive into this one. So overall this tutorial intends to build upon concepts learnt in previous tutorials while exploring new concepts.

In future tutorials we will continue to build upon the concepts learned here and will build more complex interactive games using the functionality provided by the micro:bit.



Tutorial 2.1I – Build A Stop Watch Timer

4. Activity

a. Activity

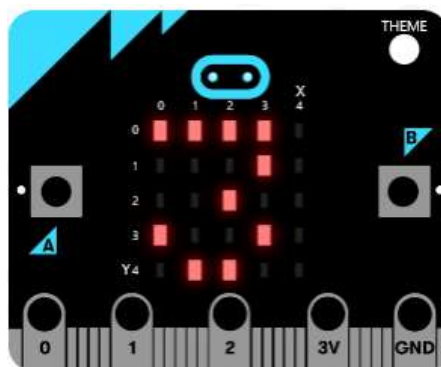
This activity involves create a Stopwatch (Countdown timer) using the BBC micro:bit. You do not need a micro:bit to complete this challenge as you can use the online block editor and emulator (www.microbit.co.uk) to code and test your program. You might have come across a countdown timer application on your digital watch or your cell phone. Very simply speaking a countdown timer allows for counting down from a pre-set number i.e. 30seconds, 60seconds, etc. Countdown timers are used during competitions, events, etc. to measure the amount of time that participants have to perform a given task.

As part of our activity designing a countdown time we will use both input buttons A and B and the LEDs for as an output.



The activity is designed to have additional challenges which allow the developer to keep pushing the boundaries. The main challenges in this activity involve –

- Button A is programmed to set the maximum value from where we will begin our countdown. By pressing button A, the user should be able to increment this value by 1 second at a time.
- Button B will be used to start the count down. Once this button is pressed the stopwatch should automatically decrement by 1 every second and it should stop when it reaches the value 0 (zero).
- The LEDs should be used to display the countdown value.



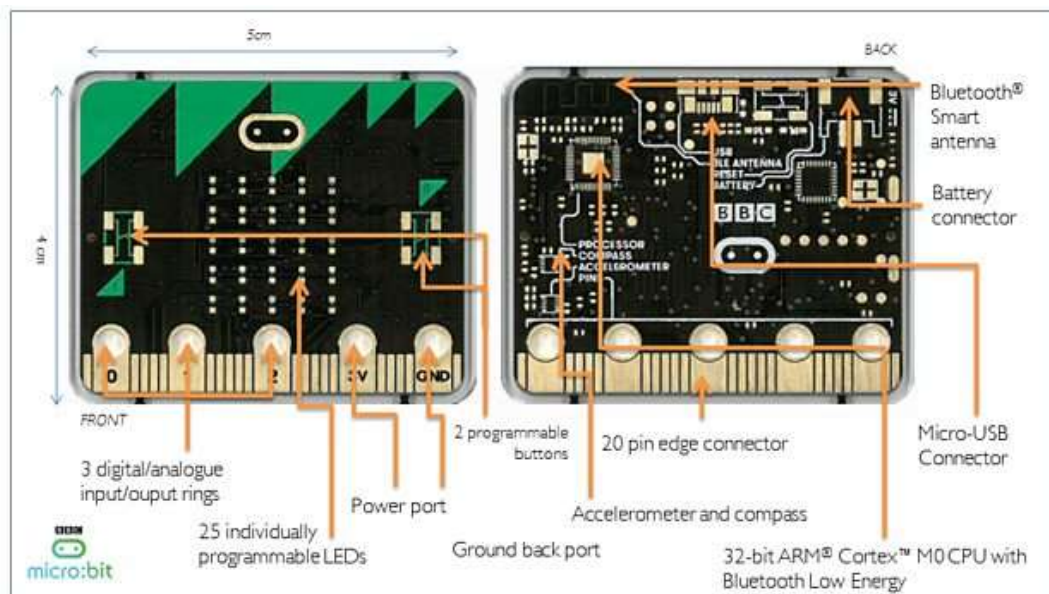
Tutorial 2.1I – Build A Stop Watch Timer

b. How Does It Work

This section talks about the BBC micro:bit and what it's made up of. If you have already read through this section then feel free to skip directly to the next section. The BBC micro:bit is a powerful little board and has various types of sensors on board. Here's what makes up the BBC micro:bit.

1. Size: approx. 5cm x 4cm.
2. Weight: 8g.
3. Processor: 32-bit ARM Cortex M0 CPU.
4. Bluetooth Low Energy.
5. Digital Compass.
6. Accelerometer.
7. Micro-USB controller.
8. 5x5 LED matrix with 25 red LEDs.
9. Pins for connecting external sensors, LED's, etc.

Here's what the micro:bit looks like –



Front of the board (left hand side)

1. Button A (left button with edge connector at the bottom) – labelled A on the board
2. Button B (right button with edge connector at the bottom) – labelled B on the board
3. P0 (left large pin (crocodile clip port) with edge connector at the bottom) - labelled 0 on the board
4. P1 (middle large pin (crocodile clip port) with edge connector at the bottom) - labelled 1 on the board
5. P2 (right large pin (crocodile clip port) with edge connector at the bottom) - labelled 2 on the board
6. +3V - labelled 3V on the board. This is 3V PWR OUT
7. GND
8. P3 – P22 pins from left to right with edge connector at the bottom. Referred to as Pins when referencing that part of the board. Text will talk about 'pins' when referring to

Tutorial 2.1I – Build A Stop Watch Timer

individual connections or the general way of connecting to the board – not labelled on the front of the board

9. LED matrix referred as the 'screen' - not labelled on the board
10. LED coordinates starting at 0,0 top left corner and ending at 4,4 at the bottom corner - not labelled on the board

The order of the large pins as follows: P0 P1 P2 3V GND labelled 0, 1, 2, 3V GND on the board

Rear of the board (Right hand side)

1. USB Plug (Micro-USB plug) – labelled USB on the board
2. Button R (reset button) – labelled Reset on the board
3. Status LED – not labelled on the board
4. Battery socket – labelled Battery on the board

Other components on the board include

1. Accelerometer
2. Compass
3. Bluetooth Smart Technology Antenna
4. AAA Battery Holder - not labelled on the board
5. Processor (Cortex M0)

The BBC micro:bit is programmable in a few different languages. You can write code for the micro:bit using the Makecode block coding interface, Javascript, Python or even in C. Most of our tutorials will cover the use of the Makecode block coding interface built by Microsoft for the micro:bit.



Tutorial 2.1I – Build A Stop Watch Timer

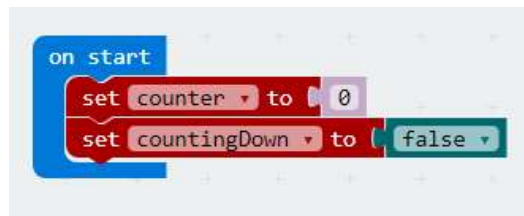
5. Writing Our Program

It's time to write some code and get going with coding our game. So let's head over to the micro:bit block code editor page (<https://makecode.microbit.org/>) and get coding!!!

a. On Start

In this challenge we are going to create a Stopwatch (Countdown timer) using the BBC micro:bit. We will use the online block editor and emulator (www.microbit.co.uk) to code and test your program. Our program will use both input buttons A and B and the LEDs for as an output:

We'll start by creating two variables. We'll create our first variable called counter which will be used to store the counter (maximum value from which we will start counting down) and the second variable called countingDown which we'll use to store a Boolean (true/false) value.



b. Implementing Button Functionality

Let's now write code so that Button A can be used to set the maximum value of the countdown timer i.e. the value from which we want to start counting down. By pressing A the user can increment the maximum value of the counter starting from 1 increasing by 1 for every click of the button.

We'll also write code for code for button B which starts the counting down. Our "on start" block sets variable "countingDown" to false which prevents the countdown from starting. However by clicking button B we want the user to be able to start the count down. We do this by changing the value of countingDown from false to true when the user clicks button B. Our forever block keeps checking the value of the variable "countingDown" and as soon as it changes from false to true it starts counting down.

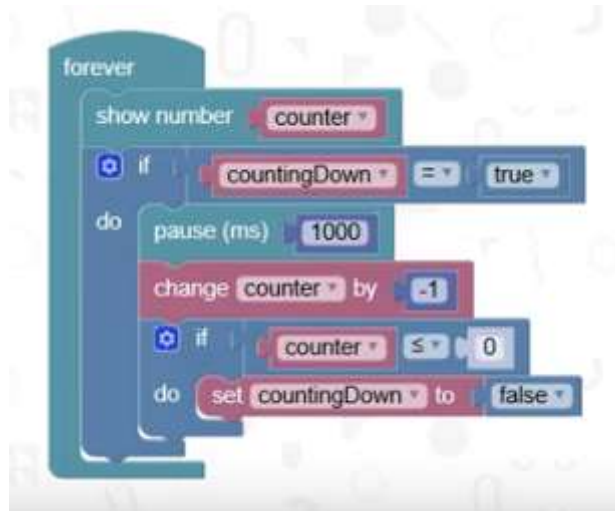


Tutorial 2.1I – Build A Stop Watch Timer

c. Main Code Block

The main code block is designed to show the current value of the counter on the LED display. It also then keeps checking to see if the “countingDown” variable has changed from false to true. As soon as the “countingDown” variable changes from false to true i.e. when the user clicks button B, the countdown timer is triggered. The countdown timer starts counting down from the current maximum value of the counter.

This counter is decreased by 1 every 1 second. This is similar to what any other countdown timer would do.

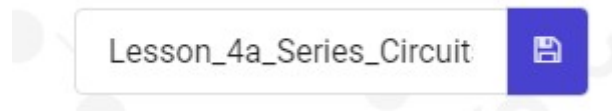


You are encouraged to make changes, improvise and customize the game using your own ideas.

Tutorial 2.1I – Build A Stop Watch Timer

6. Downloading Your Code To The micro:bit

Once you have completed the program, enter a name for your program using the option provided below i.e. in the text box adjacent to the small save button.



Now that you have given your program a name and saved it you can download it your micro:bit. But before we do that let's confirm what drive your micro:bit shows up as. On most machines the micro:bit will show up as an additional USB drive. So head over into windows explorer and confirm what drive name (D:, E:, F:, G:, etc.) the micro:bit shows up as. You need to absolutely be sure what drive the micro:bit shows up as. Once you've confirmed what drive the micro:bit shows up as on your machine you can select the right drive when downloading the code to the micro:bit. If in doubt please ask the volunteer/mentor/session facilitator helping out.



To download the newly written code to the micro:bit, hit the download button shown above. You should now see a dialog box open up and you will be asked to save the file somewhere on your machine. Please choose the drive your micro:bit shows up as i.e. D: or E: or F: or whatever it shows up as on your machine. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

If hitting the download button shown above does not open up a dialog box asking you to save to the micro:bit please save the file (you will have a <filename.hex> file) to your desktop. Then open up windows explorer and drag that file onto the drive which is your micro:bit. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

Please feel free to customize the code blocks, have a play. Add your own custom code and re-download the code to the micro:bit. Give yourself a tap on the back, you've just completed your first circuit!!!!

Tutorial 2.1I – Build A Stop Watch Timer

7. Challenges

Well done for completing the tutorial. There's a lot of ground we have covered in this tutorial so please feel free to make notes, come back to the tutorial at some point down the line and ask your learning facilitator any questions or doubts you might have on the concepts covered this far.

Now that we've built a simple countdown timer let's see how we might challenge ourselves. Here's a couple of ways you might consider challenging yourself and extending what you have already built.

- Can you update this code further so that when both buttons A and B are pressed at the same time, the timer is paused.
- Can you update this code so that when you press button A increase the counter by 5 instead of 1.
- Can you update this code so that when the timer reaches zero (0), it displays the some text, e.g. "!!STOP!!".

