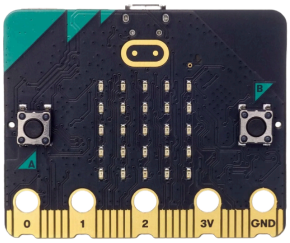
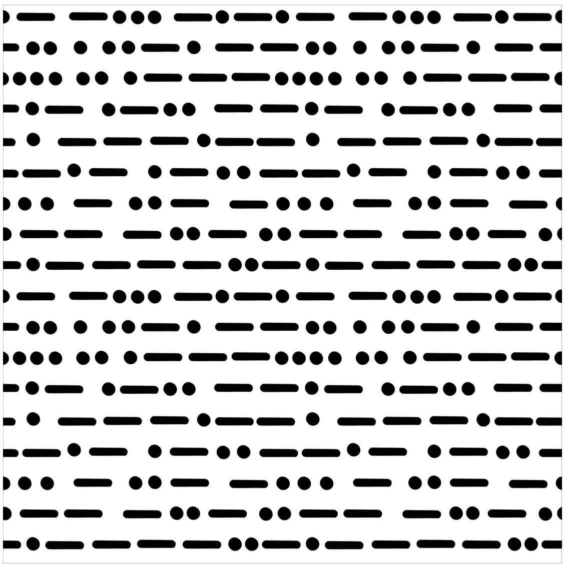


MORSE CODE MACHINE

Hello!

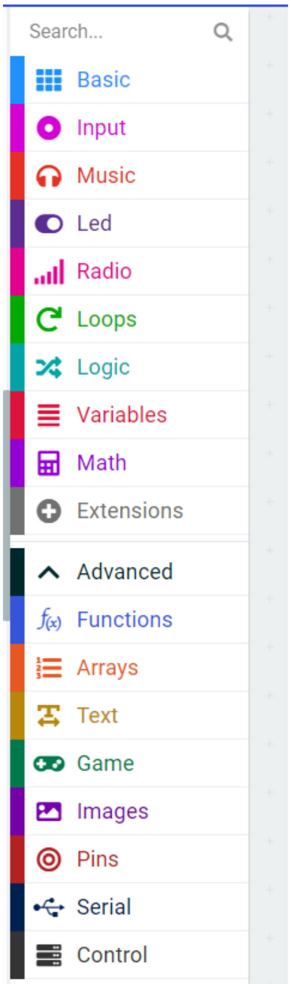
My name is Andy and I'm going to explain how to make your very own 'morse code' machine on a BBC Microbit.

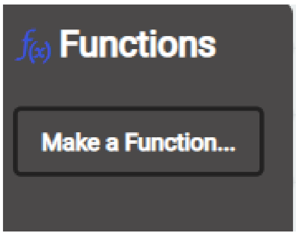


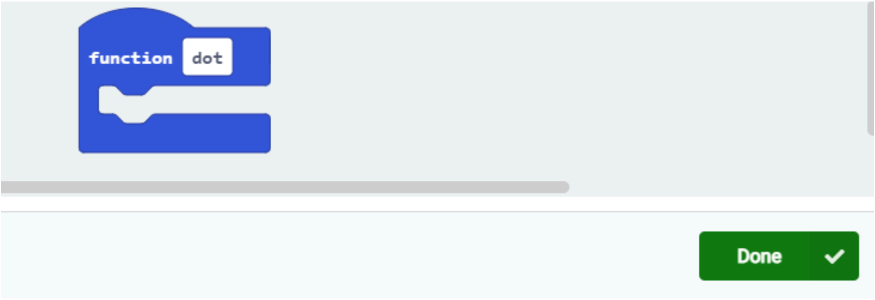
First, we need to **create two functions**.

A function is a bit like a storage box. We can make some code, put it in the storage box then tell the rest of our code to open the storage box whenever we need what's inside. A function basically means we don't need to keep writing lots of the same code over and over again.

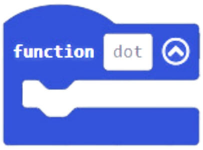
Click on **advanced and select the function** option. Then select **create a new function**. Click on "do something" and change the name to dot.



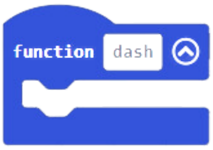




As you progress with coding you will quickly discover that most of coding is actually repeating yourself, and this next step is an example of that. You need to **create a second function** and call this one **'dash'.**



function dot



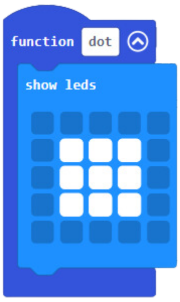
function dash

Now we have our storage boxes, we need to put some code inside them.

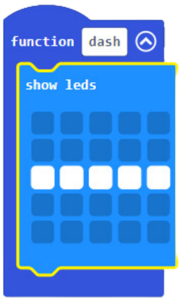
The first thing we need is to tell the lights on the front of the Microbit to light up. Go across to the basic menu and select **show LEDs**. This is a block with 25 squares on. We'll need two of these, so go ahead and grab them.

Then click, hold and drag one into each of the functions.

Following this, we're going to make these lights look like a dot and a dash.



function dot
show leds

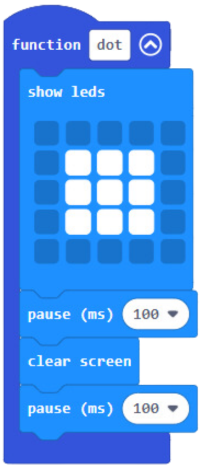


function dash
show leds

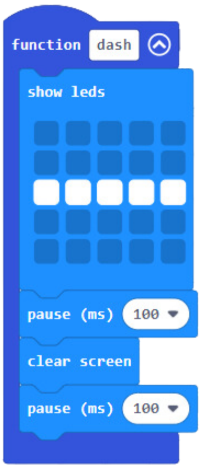
Go back to the **basic menu** and get yourself four of the **pause (ms) 100.** Then go back to the **basic menu** and get two **clear screen** instructions. Once you have all of them, you need to layer them so it goes **show LEDs > pause > clear screen > pause** in each of the function storage boxes.

The functions are telling our Microbit to show some LEDs, pause for 100 milliseconds then clear the screen before pausing again. If you like you can change the number 100 to change how long your Microbit will pause in between instructions.

These are now two of our two function storage boxes complete so we can tell all the rest of our code to call on them rather than us having to type it out all the time- because that would be repetitive.



function dot
show leds
pause (ms) 100
clear screen
pause (ms) 100



function dash
show leds
pause (ms) 100
clear screen
pause (ms) 100

But how does our Microbit know whether to make a dot or a dash? Well, at the moment it doesn't, which means we need to add some more code by adding what is called an input. An input could be lots of different things. But in this instance, our input is going to be pushing a button and it just so happens the Microbit has two of them.

Go to the menu and select the **input** option and select the top option **on button A pressed**.

Because we've already made our functions, this bit of code is really simple. Go to **function** and select **call dot** and click, hold and drag it into the on button A pressed storage box. Once that's there, go to the **radio** menu option and select **radio send string " ".**

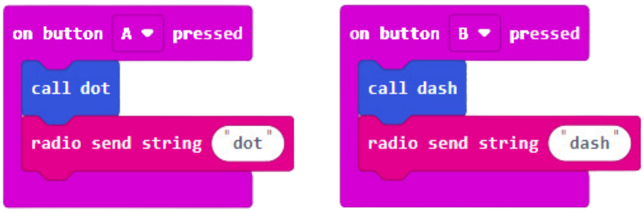
This also goes into the on button A pressed storage box. Click on the **" " and type in dot.**

Remember when I said coding means we repeat ourselves a lot? It's time to copy what we've just made. Select another **on button A pressed** from the **input** option on the menu. When this drops into the code it will probably turn a strange colour of grey.

Don't panic, because we can fix this! Click on the little **downward-facing arrow next to the A and change it to B**. This means this will now do something when button B is pressed.

Time for some repeating ourselves. Go to **function** and get **call dash** and drop it into when button B pressed, and get another **radio send string " "** but this time change the text to dash.

Then you should have something that looks like this.

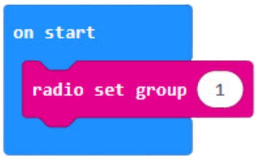


```
on button A ▾ pressed
    call dot
    radio send string "dot"
```

```
on button B ▾ pressed
    call dash
    radio send string "dash"
```

Okay, now we need a really short bit of code. We want to tell our Microbit to turn on a radio group when it turns on. For this, go into the **basic menu** and select **on start.** This doesn't need an input for our MicroBit to do something. We are saying to our MicroBit "When you turn on do this..."

The only thing we're going to put inside this storage box is in the radio menu which is called **radio set group 1** right at the top of the menu. This number can be changed to any number from 1 to 255. If you're sending your morse code to your friend then make sure they put their MicroBit on the same radio group as you are on.

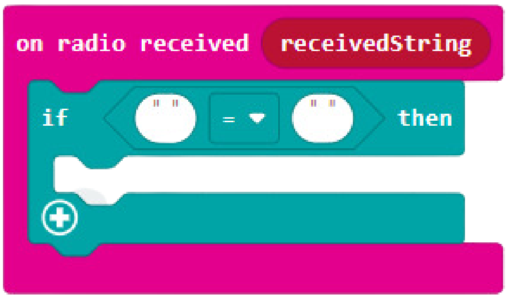


```
on start
    radio set group 1
```

Rightyho, one last bit of code to go. To finish off we need to tell our MicroBit what to do when it receives a message from the same radio group.

Go to the radio menu and select **on radio received receivedstring** and click it onto your code. Just for added information string means text. So it really says *when you receive some text over the radio* do something.

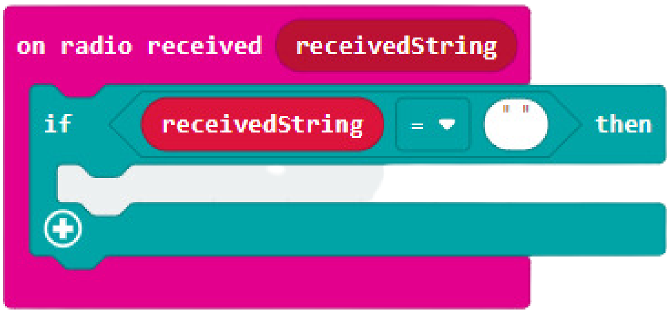
Next, go to the **logic** menu and select **if true then** or the top option if you want to keep it simple. But this into the on radio received recievedstring storage box. Head back to logic and get **" " = " "**. This is difficult to explain but you want to click, hold and drag the **" " = " "** over the word **true** so it replaces it.



```
on radio received receivedString
  if " " = " " then
```

It can be a little bit fiddly but I believe you can do it, so don't give up.

Next click and hold on the bit that says **recievedstring** and drag it into the first **" "**. Again, this can be a little bit fiddly but you'll get it.

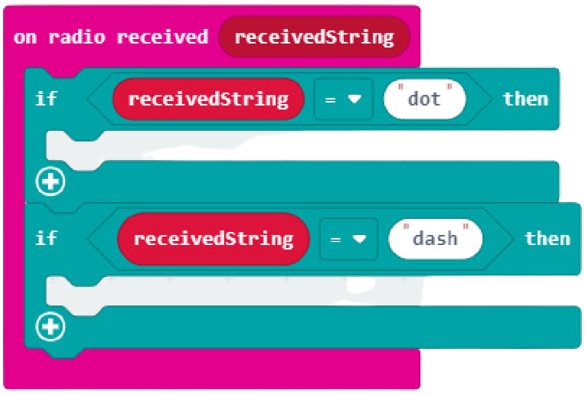


```
on radio received receivedString
  if receivedString = " " then
```

The final step for this little bit is to change that second **" "** to say **dot.**

Once you've done that you need to repeat the previous steps, but this time change the second **" "** to **dash.**

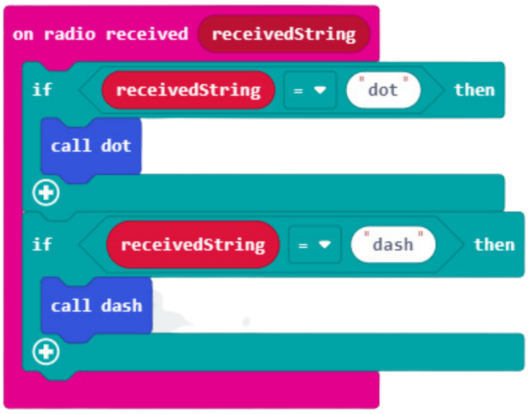
So you should have something that looks like this.



```
on radio received receivedString
  if receivedString = "dot" then
  if receivedString = "dash" then
```

Our final bits of code before we have ourselves a working Morse code radio machine.

Go to **function** and get **call dot** and **call function**. Once you have them add the **call dot** underneath the **receivedstring = dot** and do the same with **call dash** except put it under **recievedstring dash.**



```
on radio received receivedString
  if receivedString = ▾ "dot" then
    call dot
  if receivedString = ▾ "dash" then
    call dash
```

And there we have it. We should now have a fully working morse code radio machine! Time to load it onto a couple of Microbits and send some morse code messages!

Challenge time!

Here are a few extra challenges if you want to really test yourself.

- Add a command which allows you to send an "endword" symbol so your friend knows you've ended the word you were sending. Perhaps look at inputs such as "when A + B pressed"
- Create your own Morse code language
- Get your MicroBit to send a whole morse code word E.G. "SOS"