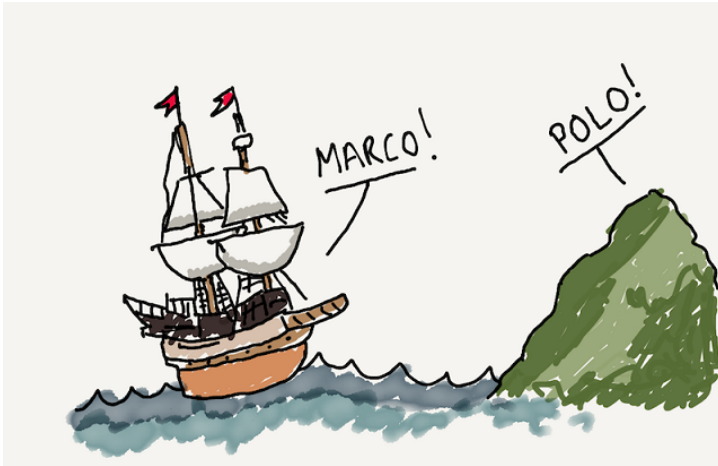


## Activity: Marco Polo and Morse code



Guide the students in creating programs that use the radio communication blocks to send and receive data between two micro:bits.

Notes:

- When using the radio blocks, the micro:bit simulator will show two micro:bits
- In the simulator, a radio transmission icon will appear in the top right corner of the micro:bit. The icon will light up as the micro:bit is transmitting data.
- In the simulator, all the code in the coding workspace runs on both virtual micro:bits. You should include for how to send data as well as what to do when it receives data.

---

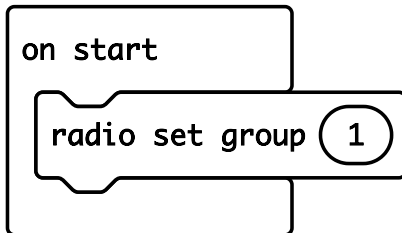
### Marco Polo

Send and receive strings between micro:bits. On button A pressed, we will send the string Marco and on button B pressed we will send the string Polo.

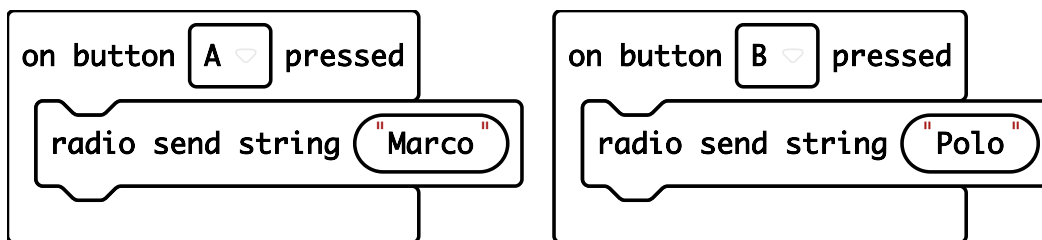
- When communicating between micro:bits, it is important that the micro:bits involved are all using the same group ID. So, the first thing we will do is set the

group ID number.

- From the Radio menu, drag a 'radio set group' block to the coding workspace and place the block into the on start block.
- In the 'radio set group block', leave the default value of 1 for the group ID



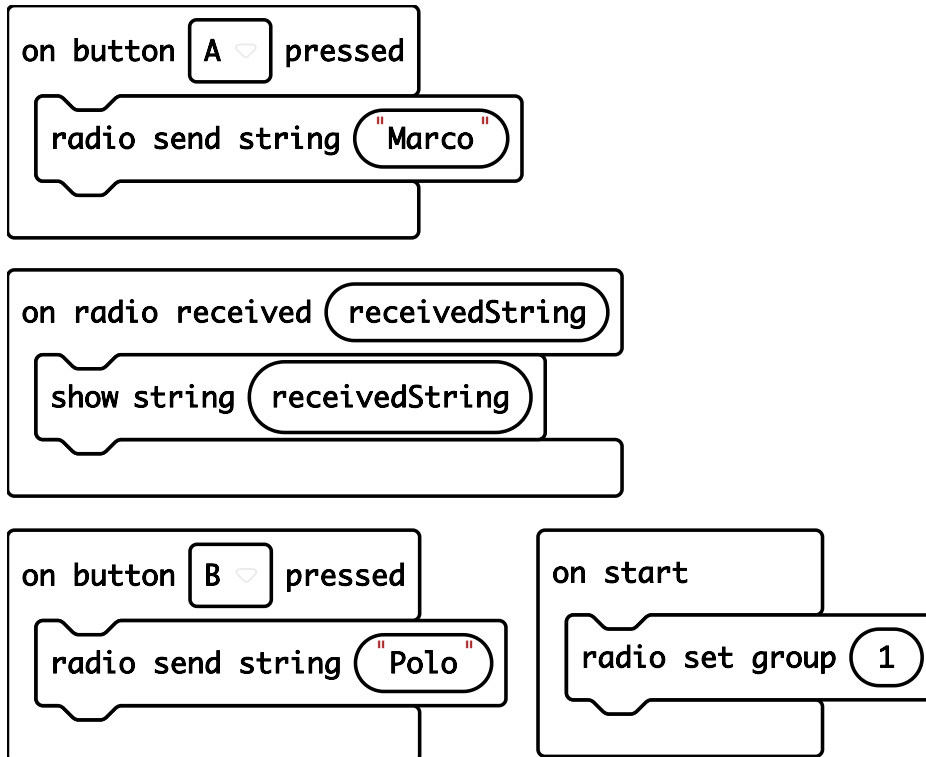
- Drag 2 'on button pressed' blocks to the coding workspace
- Leave one with the default value A and change the other button to B
- From the Radio Toolbox drawer, drag 2 'radio send string' blocks to the coding workspace
- Place one 'radio send string' block into the 'on button A pressed' block, and the other 'radio send string' block into the 'on button B pressed' block
- In the 'on button A pressed' block, change the default empty string value of the 'radio send string' block to the string "Marco"
- In the 'on button B pressed' block, change the default empty string value of the 'radio send string' block to the string "Polo"



- To display the data sent between the micro:bits, drag an 'on radio received receivedString' block to the coding workspace
- From the Basic Toolbox drawer, drag a 'show string' block into the 'on radio received receivedString' block

- From the 'on radio received receivedString' block, drag the 'receivedString' variable block into the default string value of "Hello" in the 'show string' block

Here is the complete Marco Polo program:



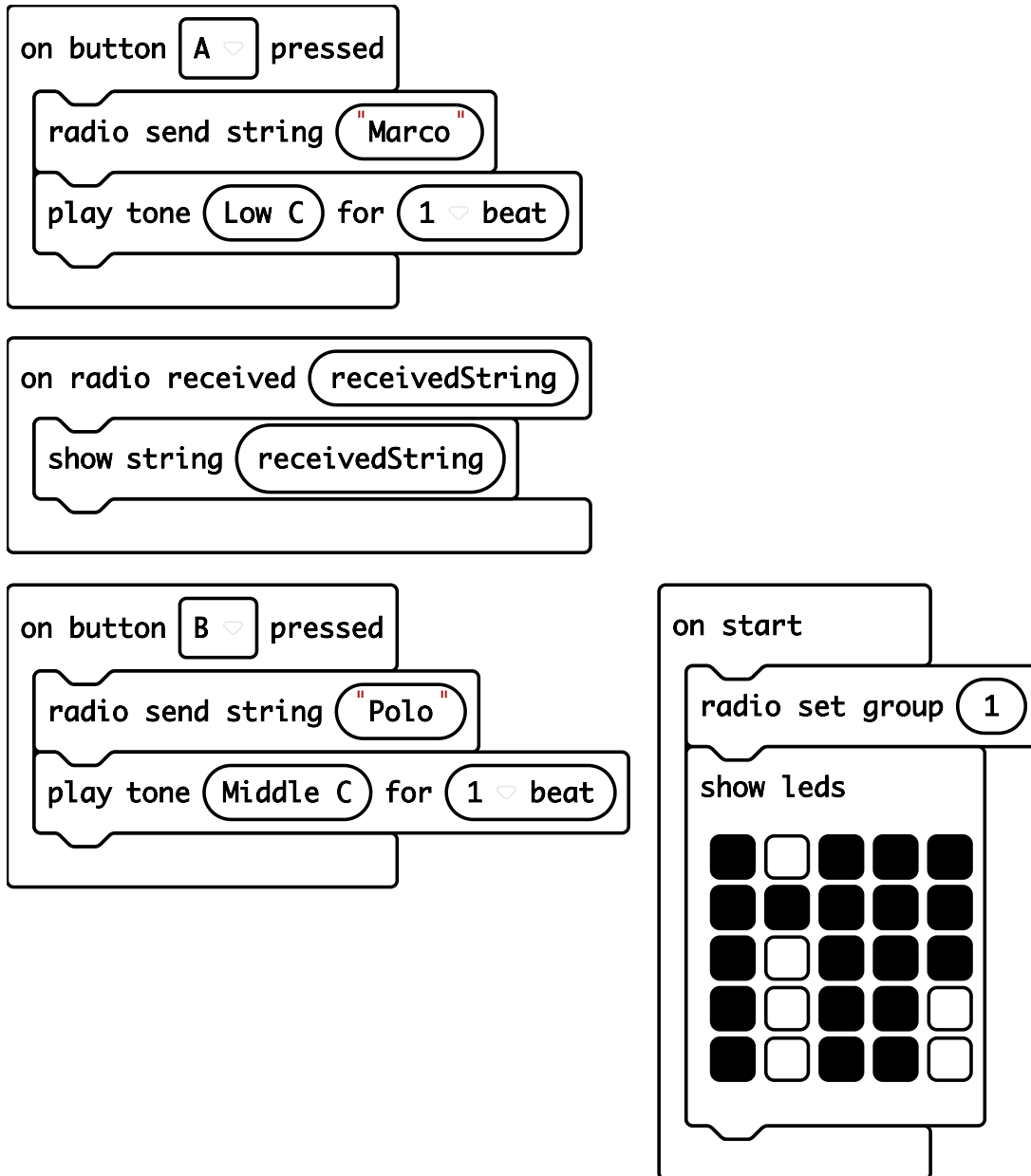
---

## Mods

- Add a 'show leds' block to the 'on start' block. We created an image of the initials MP.
- From the Music Toolbox drawer, drag 2 'play tone' blocks to the coding workspace. See **hack your headphones** (</projects/hack-your-headphones>) for how to connect a speaker or headphones to the micro:bit.
- Drag one of the 'play tone' blocks to the 'on button A pressed' block, and the other one to the 'on button B pressed' block.

- Change the default value in the 'play tone' block that is inside the 'on button A pressed' block to the value Low C.

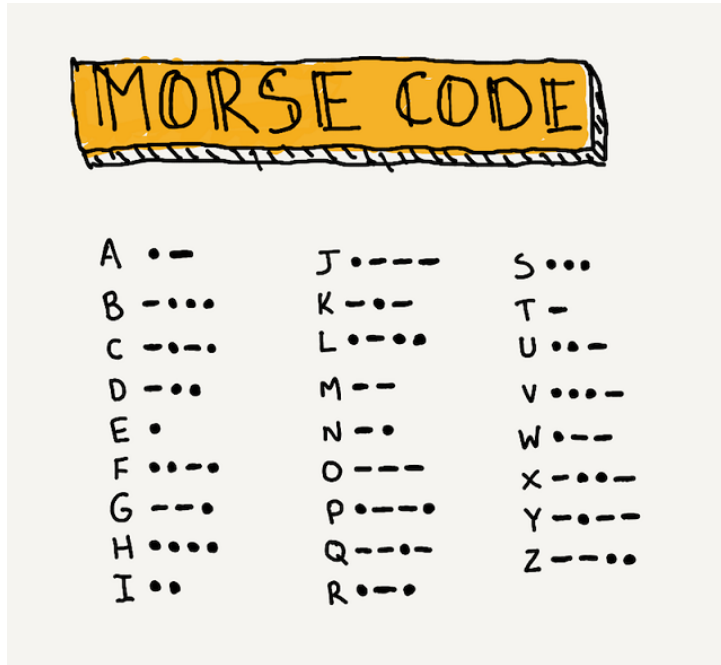
Complete Marco Polo program with mods:



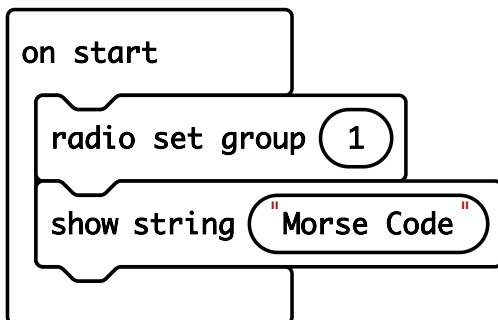
---

## Morse Code

Send and receive numbers between micro:bits. Depending on the button pressed, send a different number value between micro:bits. On receiving a number, display a different image unique to the number sent. One number will represent a dot, another a dash and another a space or stop.

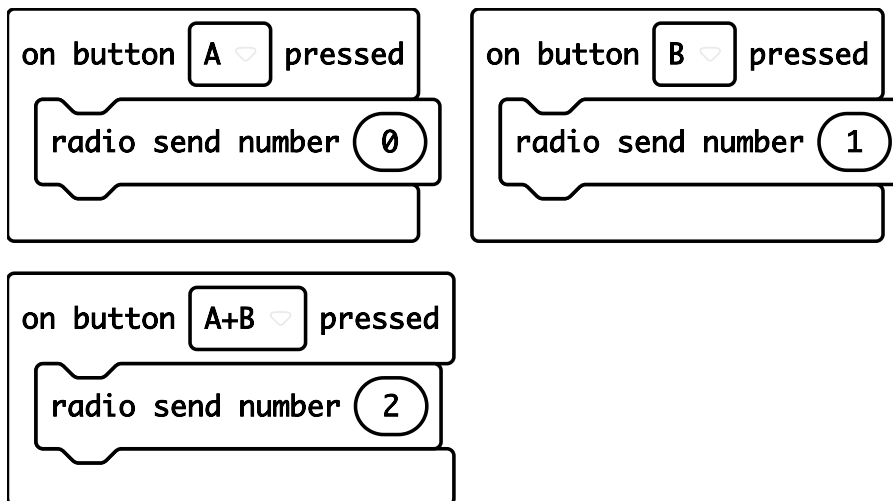


- Set the group ID number.
- Add a 'show string' block to the 'on start' block, to identify the program.
- We choose to change the default string value of "Hello" to the value "Morse Code"



- Drag 3 'on button pressed' blocks to the coding workspace.
- Leave one with the default value A, change the value in the second block to B, and change the value in the third block to A+B.

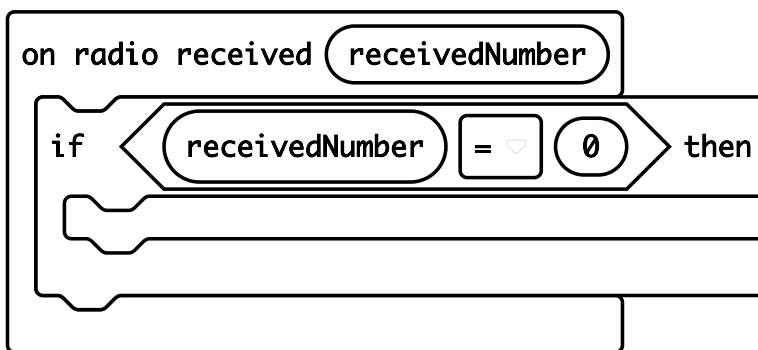
- From the Radio Toolbox drawer, drag 3 'radio send number' blocks to the coding workspace.
- Place one radio send number block into each of the 'on button pressed' blocks.
- In the 'on button A pressed' block, leave the default number value of the 'radio send number' block as 0.
- In the 'on button B pressed' block, change the default number value of the 'radio send number' block to the value 1.
- In the 'on button A+B pressed' block, change the default number value of the 'radio send number' block to the value 2.



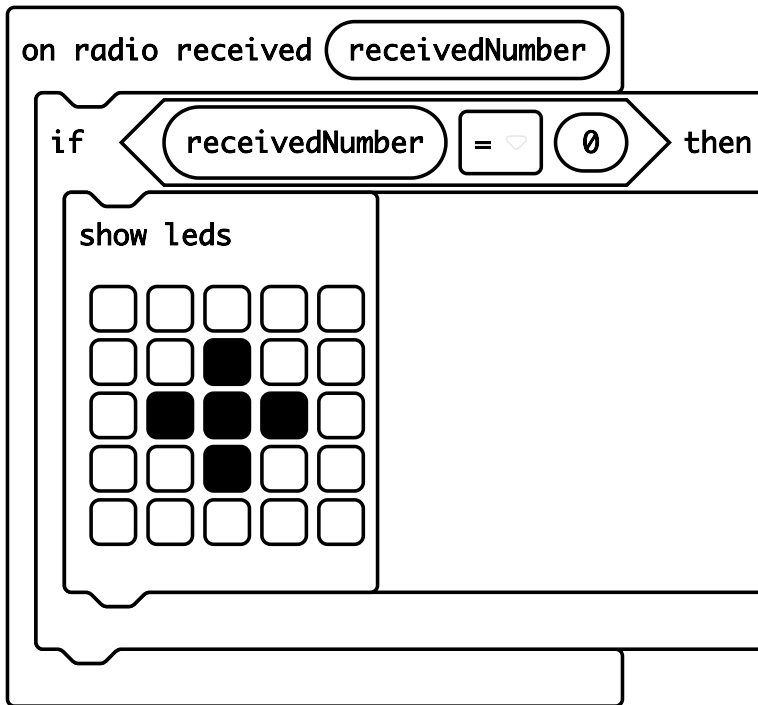
- From the Radio Toolbox drawer, drag an 'on radio received receivedNumber' event handler to the coding workspace.
- Since we will display a different image depending on the number value received, we need a logic block.
- From the Logic Toolbox drawer, drag an 'if...then' block to the coding workspace and place it in the 'on radio received receivedNumber' event handler.

In order to know whether to display a dot, a dash, or a space/stop image, we need to compare the number received to the values 0, 1, and 2.

- From the Logic Toolbox drawer, drag a 0=0 comparison block into the coding workspace.
- Replace the default value 'true' of the 'if...then' block with the comparison block.
- From the 'on radio received receivedNumber' block, pull down the 'receivedNumber' variable block and drop it into the first slot of the comparison block
- Leave the righthand side default value of zero in the 0=0 block.



- Place a 'show leds' block in the space after the then of the 'if...then' block.
- Create an image to represent a dot.

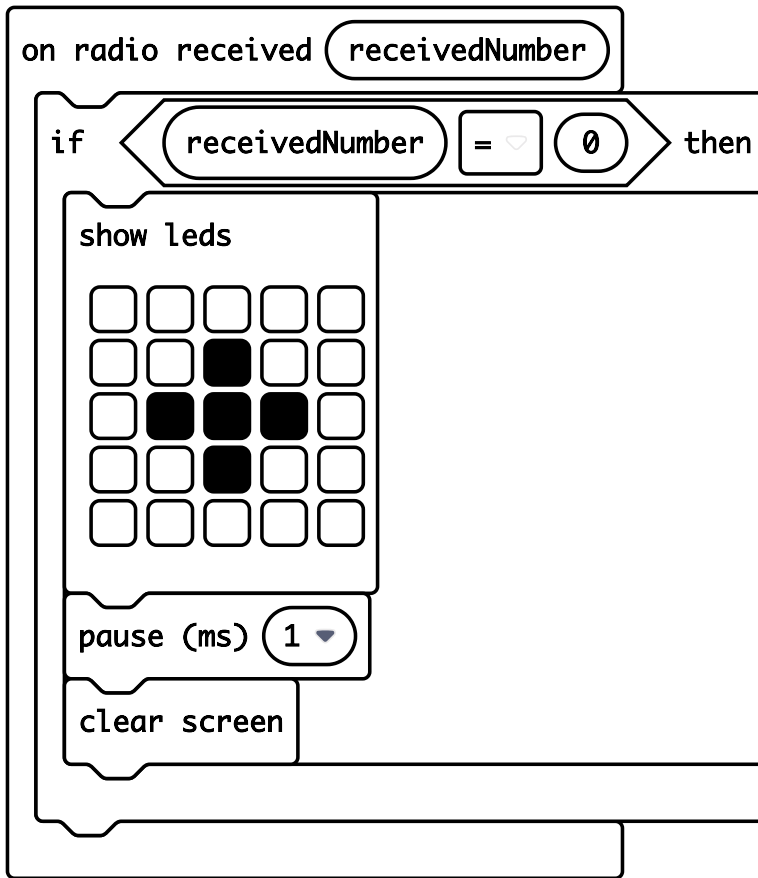


## Try it!

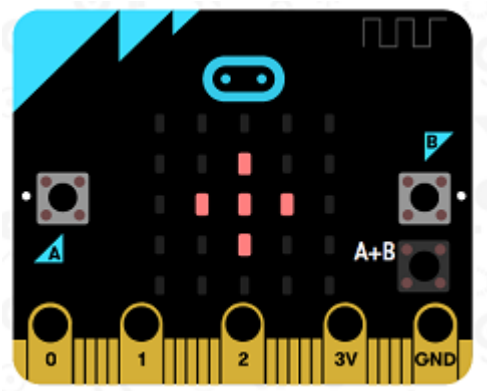
- Download your program to the micro:bit
- Press button A on the sending micro:bit
- Does this cause a dot to be displayed on the receiving micro:bit?
- However, pressing button A again does not appear to send another dot as the image on the receiving micro:bit does not appear to change.

Challenge question: How can we fix this?

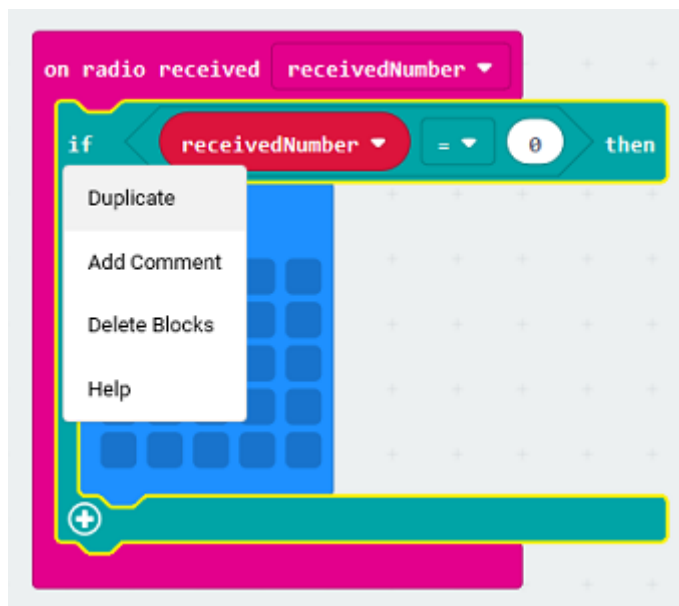
- Add a 'pause' block and a 'clear screen' block after the 'show leds' block



Try running the program again. Now each time the sender presses button A, you see a dot appear.

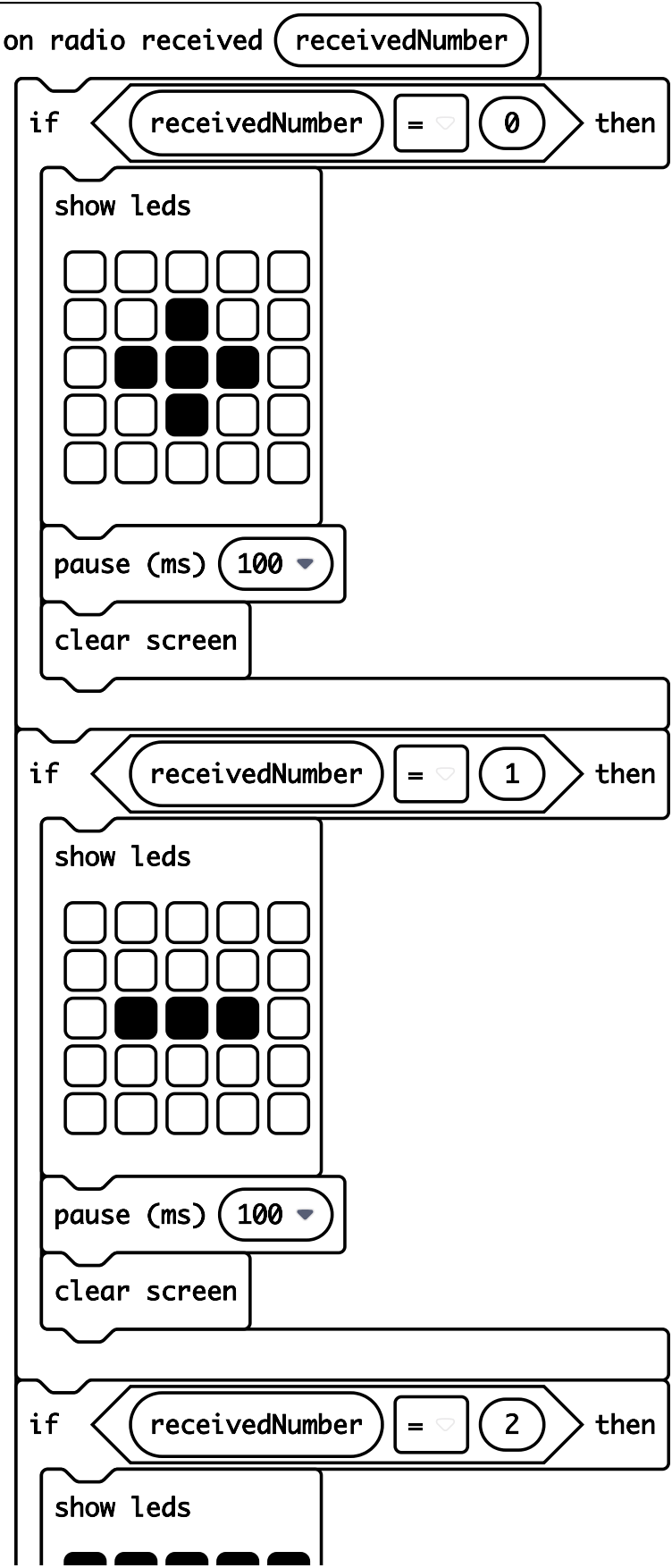


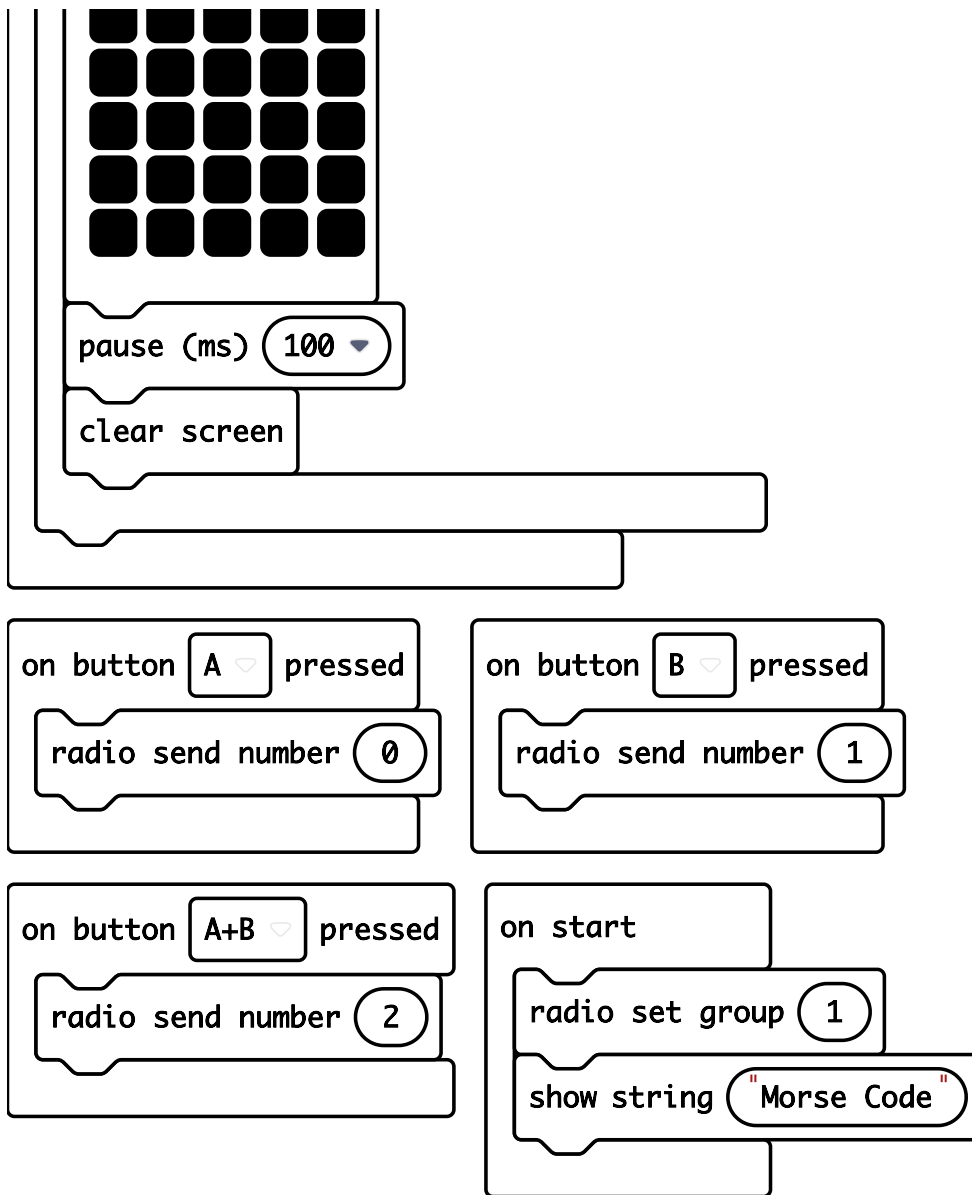
- You can now right-click on the 'if...then' block and select Duplicate to copy that piece of code twice for the other 2 values that a sender may send.



- Change the values on the righthand side of the comparison block to 1, and 2.
- Modify the images displayed to show a dash, and a full screen of lights

## Morse code program





## Try it!

- Download your program to the micro:bit
- Press buttons A, B, and A+B together on the micro:bit

Challenge question: Can our code be made more efficient?

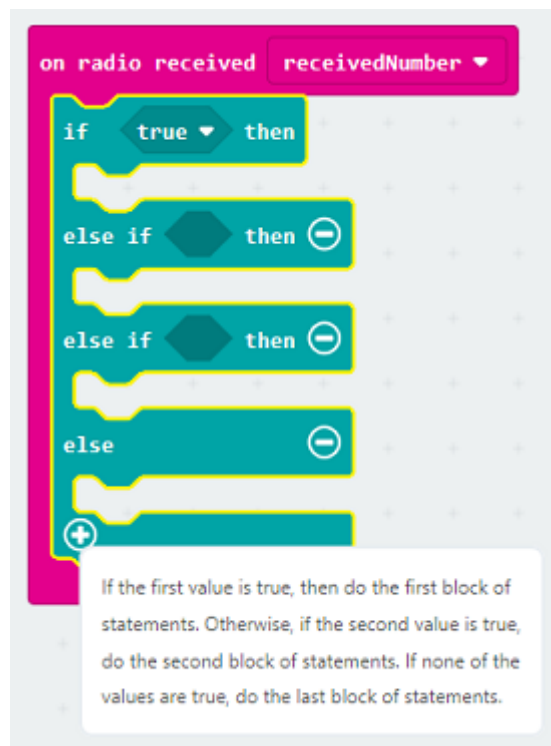
- Whenever you look over a program and see the same lines of code repeated, there is usually a chance to improve the code making it more efficient by reducing the number of lines of code
- What lines are repeated in our program? If...then, pause, clear screen

- Can we edit the code to use only one 'if...then' block, one 'pause' block, and one 'clear screen' block? Yes!

---

## Making our code more efficient

Remind students that they can edit the 'if...then' block, adding as many 'else if' conditions as needed. They can do this by clicking on the (+) or (-) symbols on the 'if...then' block.



A final else In a conditional that might receive a number of different values, it is good coding practice to have a catch-all 'else' clause. In our example, if any number value other than the ones we coded for (0, 1, and 2) is received, we can signal the user that an error has occurred by using a 'show icon' block to display an X.

The pause and clear screen Rather than repeat these lines of code 3 times, we can move the 'pause' block and the 'clear screen' block outside of the edited 'if...

then...else' block.

Now our program runs as we designed it to run and is more efficient, too!

Final Morse Code Program:

on button **A** pressed

radio send number **0**

on button **B** pressed

radio send number **1**

on button **A+B** pressed

radio send number **2**

on radio received **receivedNumber**

if **receivedNumber** = **0** then

show leds

|                          |                                     |                                     |                                     |                          |
|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |

else if **receivedNumber** = **1** then

show leds

|                          |                                     |                                     |                                     |                          |
|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |

else if **receivedNumber** = **2** then

