Extension: Co-op Bop!

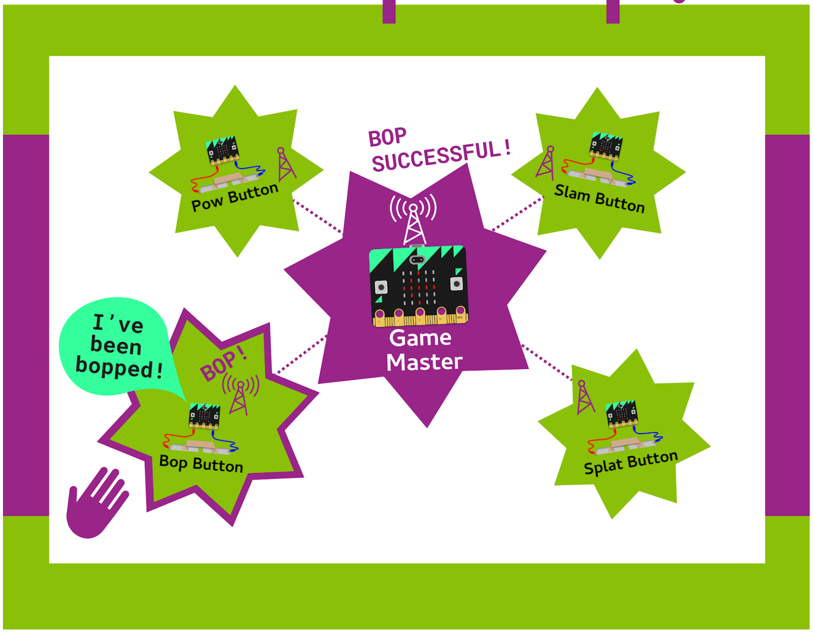
Make a Bigger, Better Bop It, and play as a team!

Spread out your game by making buttons that talk to your game with radio!

This extension has 2 parts.

- Making radio buttons.
- Changing your game to listen for the radio buttons.

You can work in groups!



Learning about Radio

We'll need to know how to use the radio for this extension. Here's some commands

Action	Code
Set the channel, we set it to 6.	`radio.config(channel=6)`
Turn the radio on	`radio.on()`
Send a message, we sent "bop"	`radio.send("bop")`
Receive a message, check if it matches 'bop'	`if radio.receive() == "bop":`

Part 1

Making Radio Buttons

A new file that sends a radio message every time you press a button.

Use the Micro:Bit buttons or craft your own!

Part 2

Listening for Radio Buttons

Adapt your game to use radio messages to complete actions.

You can connect lots of buttons, so work as a team to make more.

Part 1: Making Radio Buttons

We'll make a new program for our radio buttons.

It will:

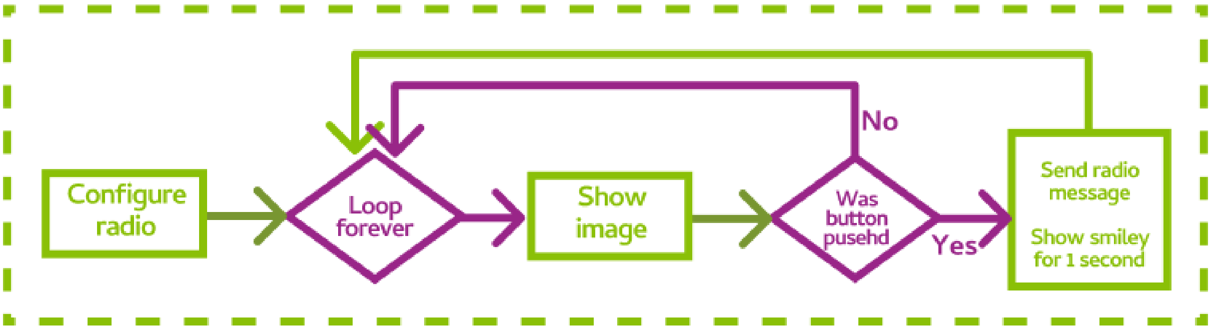
- Always show a picture so we can identify which button it is.
- Send a radio message when we press the button.

Task 1.1: Configure the Radio

We'll need to start a new file for our game master!

1. Click the **Project** button to the left, and then click **Create file**.
2. Name your new "race_friends" file.
3. At the top of your file, `import` the `microbit` and `radio` modules.
4. Turn the radio on with `radio.on()`.
5. Then configure the radio's channel with `radio.config(channel=100)`. Your room coordinator will tell you what number to use.

Next we'll make a loop that sends a message every time the button is pressed!



Task 1.2: Loopy for radio

1. **Create an infinite loop** after you have set up the radio

2. Inside the loop make the Micro:Bit display an image that will be it's identifier.

Task 1.3: Checking presses & sending messages

1. **Inside the loop create an if statement that checks for an action.**

   Actions: *We have the Micro:Bit buttons, shake and other gestures, and buttons you craft yourself! Remember to use the correct code, e.g.* `is_pressed()` or `read_digital()`.

2. **In the if statement, send your action to the game master**: *Remember to use your own action name:* `radio.send("bop")`

Task 1.4: Smiley faces for pressing!

1. After you send a message **show a smiley face** for 1 second

2. **Run your code** in the Grok simulator to see it communicate to the other MicroBit!

Joining a game

Now you have a working button you can join a game. You'll need to make sure your button's message is included in the Game Master code.

You can write or radio game code in part 2, or join a friend's or tutor's game!

In the game master code you'll need to:
- Include your button name in the list of actions
- Add an if statement that displays your button's image when it is chosen.
  Make sure your button name and image are unique for the game!

☑ CHECKPOINT ☑

If you can tick all of these off you have finished this Extension:

□ You have configured your radio with the channel your tutor gave you

□ When you press your button it sends a radio message to Game Master.

□ You show a smiley face every time you press your button

□ You added your button to an existing Co-op game!