microBlocks
Small, Fast, Human Friendly

Make a micro:bit Dance | Teacher Notes

Introduction

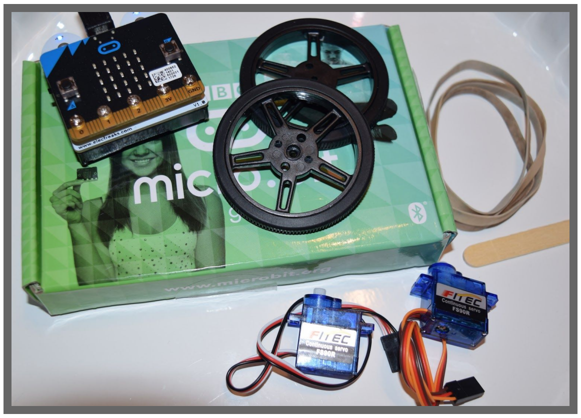
In this activity, students will be able to make a micro:bit creature that moves and dances out of a few simple parts: a micro:bit, the box it comes in (or something similar), rubber bands, servos, wheels, and a popsicle stick. Students will also explore the power of remote control to move their projects in creative ways.

Goal

Students will create a dancing micro:bit by affixing two continuous rotation servos with wheels to a box. Students will write scripts to control their project by adjusting the speed and direction of the servos and wheels.

Materials needed:

- micro:bit with battery pack and extension board (such as the ring:bit or basic:bit)
- 2 continuous rotation servos
- 2 wheels
- Small box (like the one many micro:bits come in)
- At least 4 rubber bands (#64, 3-1/2 x ¼ work well)
- Small popsicle stick
- Tape

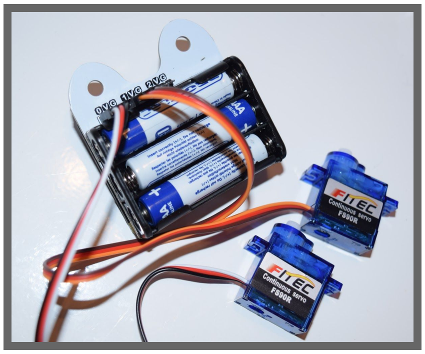


Procedure

Introduction

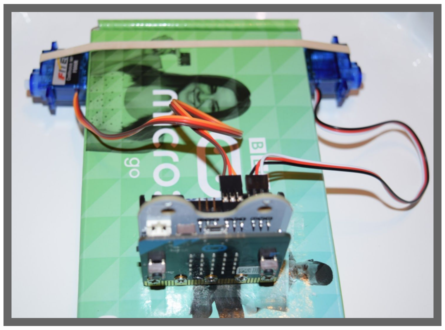
- This activity connects to many curriculum standards. Students may make connections to simple machines, discussing wheels and axles. There is also an opportunity to connect to physics concepts around motion.
- One way to introduce this activity and gather background knowledge is to show students the materials they will use (or project the image above) and ask them what they think they might make.
- Explain to students that these components will allow them to make their creation move in all sorts of directions - wiggle from side to side, turn in place, or move forward and backward.
- It might be fun to brainstorm simple dances that students imagine their micro:bits performing!

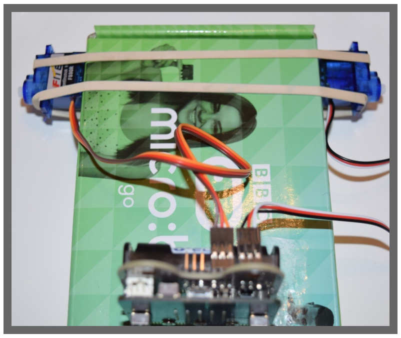
Getting Set Up



The first step in assembling this project is connecting the servos to your micro:bit. Here, we are using a ring:bit. Connect the servos so that the dark wire (black or brown) is connected to the "G" or ground on the ring:bit. *If you are not using an extension board with its own batteries, you will need to attach a battery to your micro:bit. The micro:bit cannot power the servos using only power from the USB cable.*

Next, using a rubber band, attach one servo to the box as you see in the picture to the right. Then, carefully slide the other servo under the rubber band on the opposite side. Attach an additional rubber band for support.

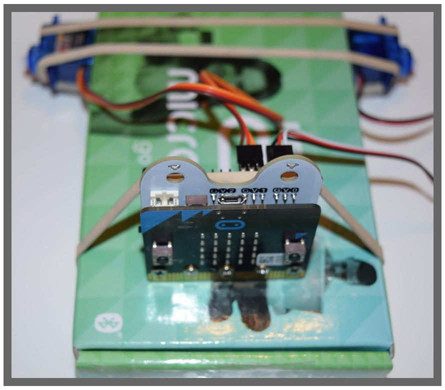


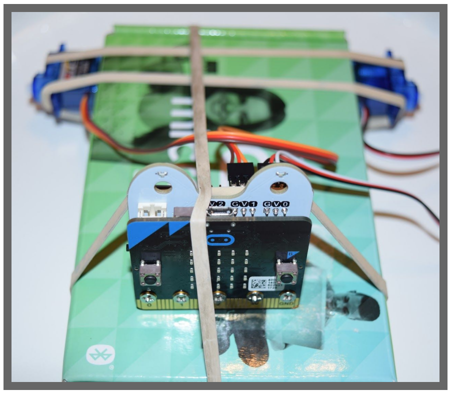


To mount your micro:bit on the box, attach a rubber band around the box widthwise.

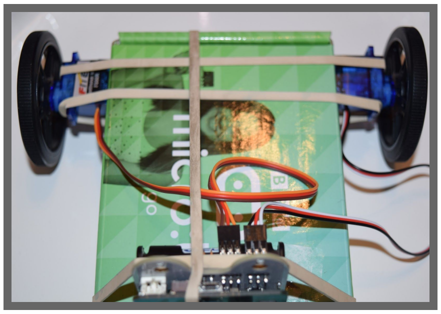
Note: you may want to remind students to be gentle with the servo connections.

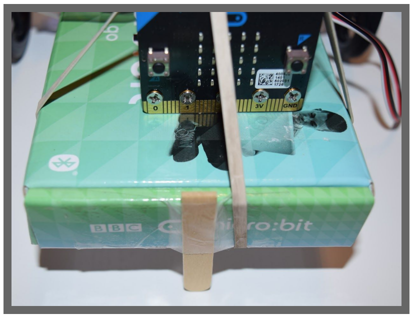
The final rubberband helps hold the micro:bit more securely on top of the box. Stretch the rubber band along the box lengthwise.





Next, affix the wheels to the servos. The screws that come with the servos can be used to secure the wheels if they fall off too easily. Finally, attach the popsicle stick to the end of the micro:bit so that it is opposite the wheels. This helps the box slide easily on hard surfaces, enabling the servos and wheels to pull the weight of the box as it dances!





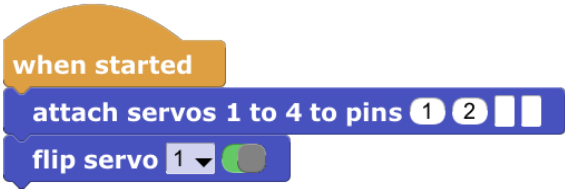
For more information, visit microblocks.fun
Questions? Email us at interest@ microblocks.fun

microBlocks
Small, Fast, Human Friendly

Make it Move

Now that the box is assembled, students will want to make it move! Connect to MicroBlocks by attaching the USB cable to the micro:bit. *Note: make sure that the extension board you are using is turned on.* It is helpful to use a long USB cable for this connection. Alternatively, students can also code the micro:bit before the project is assembled. Students could connect the servos and wheels to the micro:bit to test the movement then make adjustments once the project is put together.

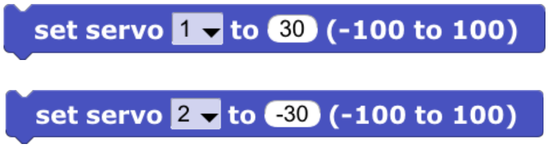
To begin, add an *attach servos* block and change the numbers to match the pins to which the servos are connected. In this example, the servos are attached to pins 1 and 2. The *flip servo* block is needed because one of the motors had to be physically "flipped around" in order to attach it to the opposite side of the box. The flip block makes both wheels push the box in the same direction when the servos are set to the same numbers.



```
when started
  attach servos 1 to 4 to pins 1 2
  flip servo 1
```

Safety Note: For the next few pieces of code, it is helpful to instruct students to hold their box off a surface so they can see the rotation of the wheels. This will prevent the USB cable from becoming taut and possibly damaging the project or computer.

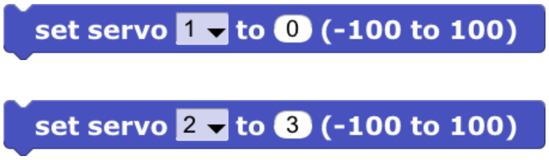
Next, drag two *set servo* blocks into the scripting area. Encourage students to experiment with different numbers. What happens when they set two positive numbers? What about two negative numbers? By setting two positive numbers, the box should move forward. Two negative numbers and the box should move backward. One positive and one negative will make the box turn! *Hint: remember to include the flip servo block in the setup script.*



```
set servo 1 to 30 (-100 to 100)

set servo 2 to -30 (-100 to 100)
```

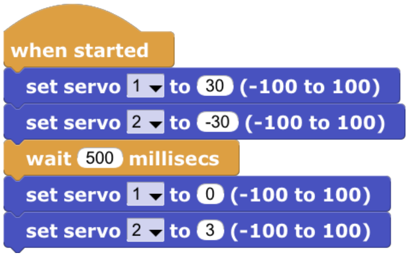
Once students have made the box move, they may want to make it stop and get ready for another dance move. Using the same *set servo* blocks, find the number that makes the wheels stop rotating. This is often 0, but may need to be adjusted based on the servo. For example, sometimes it requires a number slightly above or below 0 to stop rotating completely. Here, that number is 3.



```
set servo 1 to 0 (-100 to 100)

set servo 2 to 3 (-100 to 100)
```

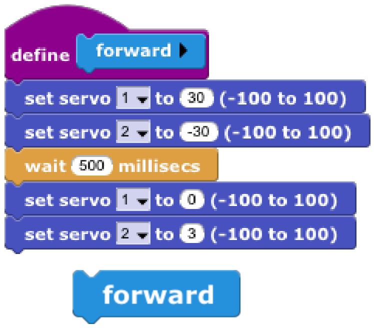
Put these scripts together with a *wait* block in between. When the program is started, the box will move forward and stop.



```
when started
  set servo 1 to 30 (-100 to 100)
  set servo 2 to -30 (-100 to 100)
  wait 500 millisecs
  set servo 1 to 0 (-100 to 100)
  set servo 2 to 3 (-100 to 100)
```

For more information, visit microblocks.fun
Questions? Email us at interest@ microblocks.fun

This is a great opportunity to show students how to make their own block in MicroBlocks! This code can be made into a block labeled "forward." Click on the **My Blocks** category, add a command block, and name it forward. Next, move all the scripts previously under *when started* to define the forward block. Drag the new forward block into the scripting area to use this new command. Encourage students to make different command blocks to make their projects move and dance.

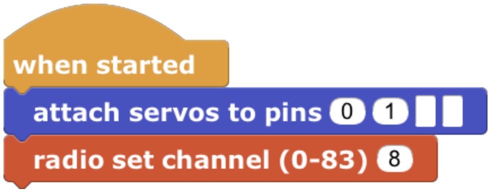


```
define forward
  set servo 1 to 30 (-100 to 100)
  set servo 2 to -30 (-100 to 100)
  wait 500 millisecs
  set servo 1 to 0 (-100 to 100)
  set servo 2 to 3 (-100 to 100)

forward
```

Setting Up a Remote Control

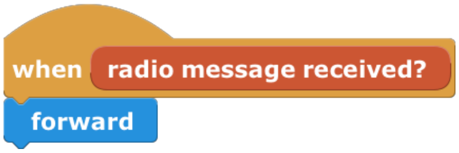
Once the project is moving the way students would like, they may want to control it from afar. Students can write a code to use another micro:bit to control the micro:bit on the box. This will make their project hands free! This process introduces the concepts of remote control using radio blocks. Follow the steps below to set up a remote control.



```
when started
  attach servos to pins 0 1
  radio set channel (0-83) 8
```

First, while the micro:bit on the box is still connected, add a *radio set channel* block to its code under the *when started* block. Choose a number between 0 and 83 and remember it!

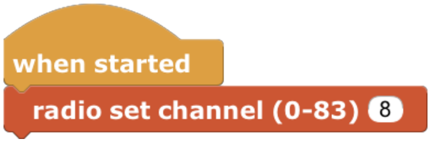
Next, add a *when radio message received* block and tell the box what to do when it receives the message.



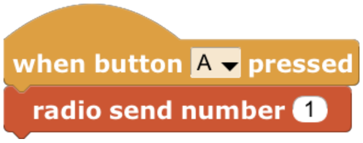
```
when radio message received?
  forward
```

Teaching Note: if you are working in a classroom with several micro:bits, you will want students to choose different channels so as not to interfere with each other's projects. Make a list on the board so everyone knows what numbers are in use.

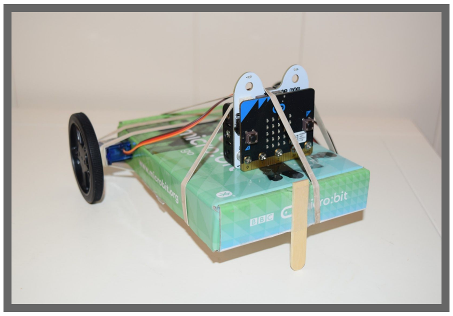
Save your current MicroBlocks project and open a new one. This is the one that will program your second micro:bit, which will act as a remote control. Disconnect the micro:bit on the box and connect your remote control micro:bit to your computer. Add these two instructions for the remote control.



```
when started
  radio set channel (0-83) 8
```



```
when button A pressed
  radio send number 1
```



First, set the channel to be the same as the micro:bit on your box. Next when button A is pressed, use the *radio send number* block to send the command to the micro:bit on the box. Unplug the remote control micro:bit from the computer (make sure it has a battery pack attached) and control your project!

For more information, visit microblocks.fun
Questions? Email us at interest@ microblocks.fun