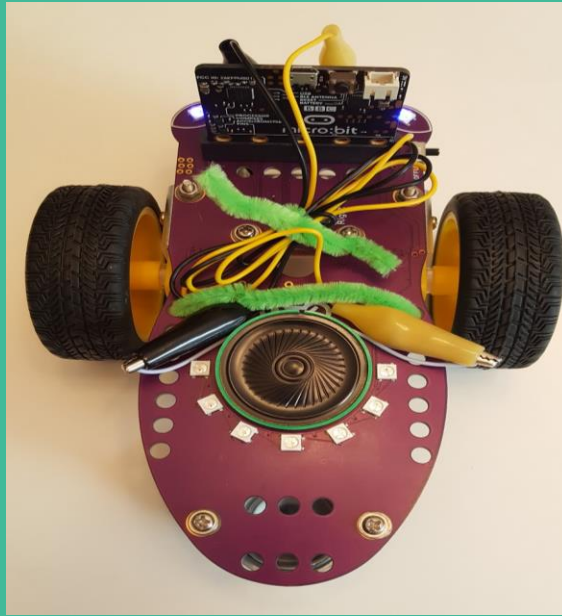


# GIGGLY GIGGLEBOT

## MISSION 7



# TABLE OF CONTENTS

• GIGGLY GIGGLEBOT .....	1
Your Role: Script Writer .....	1
Your Task: Turn the GiggleBot into a Robot Actor .....	1
Considerations.....	1
Materials.....	1
 • LEARN: SPEAKER .....	2
Connect the Speaker .....	2
Create Music .....	2
 • PLAN IT OUT.....	7
Build Your Character .....	8
Build Your Program.....	8
 • ARE YOU STUCK?? .....	9
Let's Start Coding.....	9
Sensor Use.....	10
 • TRY IT OUT .....	12
Extension .....	12



## > GIGGLY GIGGLEBOT

### YOUR ROLE: SCRIPT WRITER

 A script writer is a person who writes a the text for a play, movie, or broadcast.

In this case you are going to be writing the script in MakeCode.

### YOUR TASK: TURN THE GIGGLEBOT INTO A ROBOT ACTOR

You are going to create a character for a book or movie and bring it to life with sounds using the GiggleBot with a speaker.

### CONSIDERATIONS

- > Is your robot good or evil?
- > Does it laugh when it does something bad? Or, is it silly and just laughs at everything?
- > Is it mischievous or curious?
- > Does your GiggleBot drive around playing background music or sounds for a movie scene?

### MATERIALS

- > GiggleBot and good batteries
- > micro:bit and provided cable
- > Laptop / computer
- > Speaker
- > Alligator clips
- > Pipe cleaners or tape to attach the speaker
- > Optional: craft supplies to transform your GiggleBot into a character (Ex. construction paper, pipe cleaners, or cardboard).

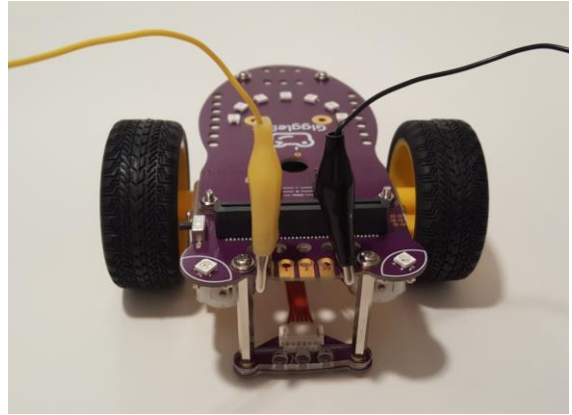
## > LEARN: SPEAKER

### CONNECT THE SPEAKER

The GiggleBot can make sounds if we connect a speaker.

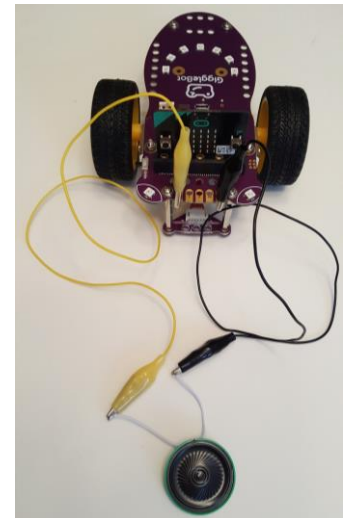
To do this, we will use the pins on the front of the robot, just like in mission 3.

- > Connect one alligator clip to the GND pin and another alligator clip to **P0**.



**i** For the music blocks, only **P0** will work.

- > Connect the other ends of the alligator clips to the wires on the speaker.
- > Make sure that the alligator clips are connected to the metal wire on the speaker, not the insulation covering the wire.
- > Make sure the alligator clips do not touch each other.



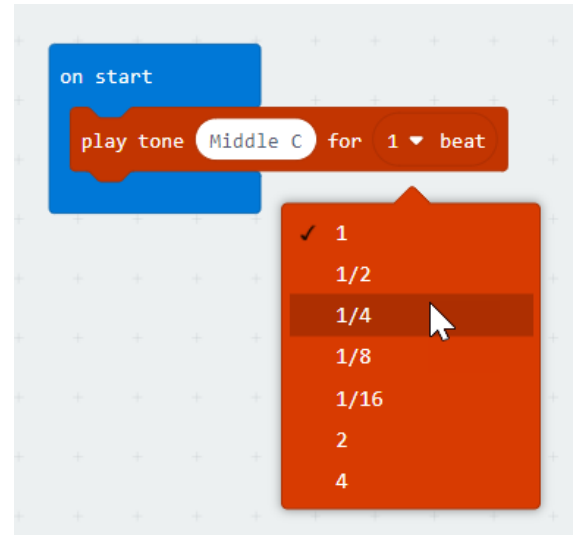
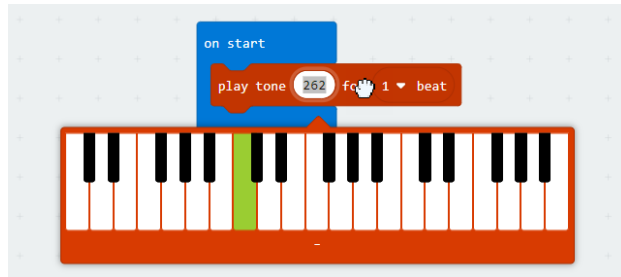
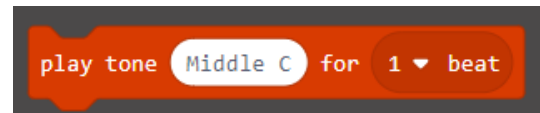
### CREATE MUSIC

Now, let's program the speaker to make some noise. Move an **on start** block to your workspace. Then, look under **Music** to see some of the block options.

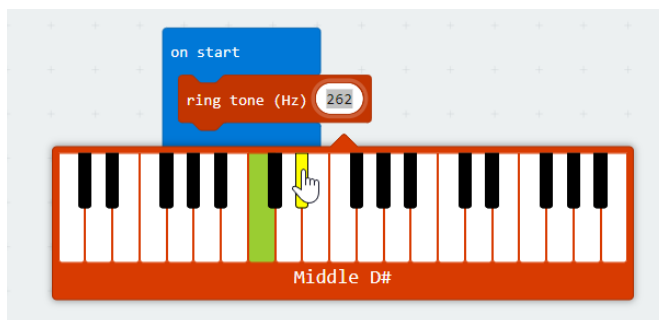
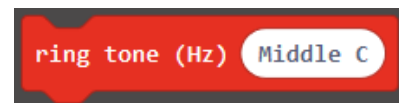
Turn your computer sound on. If you move these blocks into the **on start** block, the emulator will play the sounds for you to hear. This will help you preview what the speaker on the GiggleBot will sound like.

Think of the first three blocks like you play an instrument.

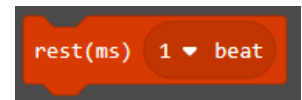
- 1 The **play tone \_\_\_ for \_\_\_ beat** block plays a specific note for a specified number of beats. You can change the note and how much of a beat using the drop-down menus.



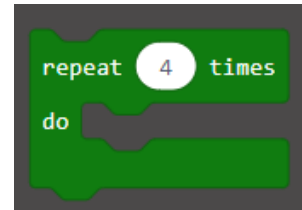
- 2 The **ring tone (Hz) \_\_\_** block plays and holds the selected note until you play another note. Again, you can change the note by clicking on the dropdown.



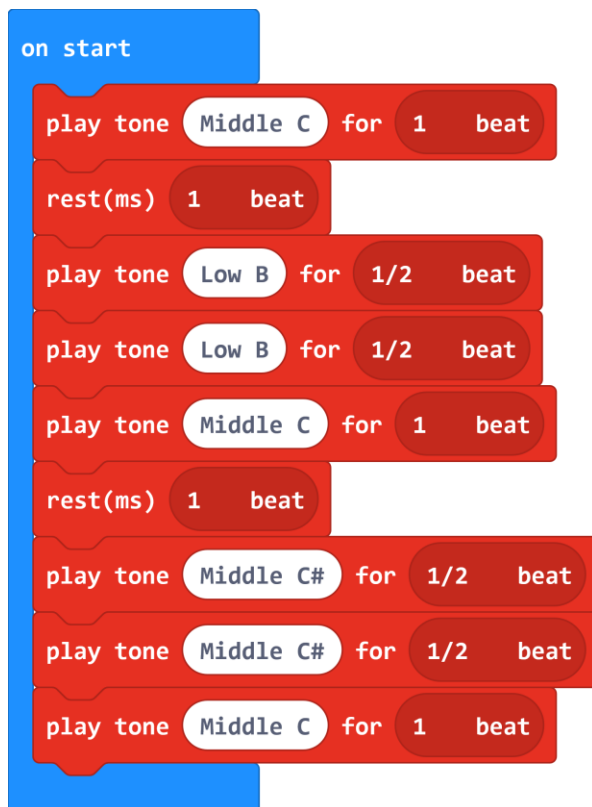
- 3 The **rest (ms) \_\_\_\_ beat** block is just like a rest in music. It tells the GiggleBot to wait for the given duration of time before moving on. You can change the length of the rest using the drop-down menu on the block.



Create a tune using a combination of these three music blocks. If you want the GiggleBot to repeat something you can use a **repeat** loop block.



Once you are finished with your program, download and transfer it to your GiggleBot . When your GiggleBot turns on, it will play the music you created. Two examples are shown below. Try them both, one after the other.





You can change the speed of your music using the tempo blocks. Add either the **change tempo** block or the **set tempo** block to your program and see how it changes your music.

Download and transfer the modified program to your GiggleBot and play your music. An example is shown below.

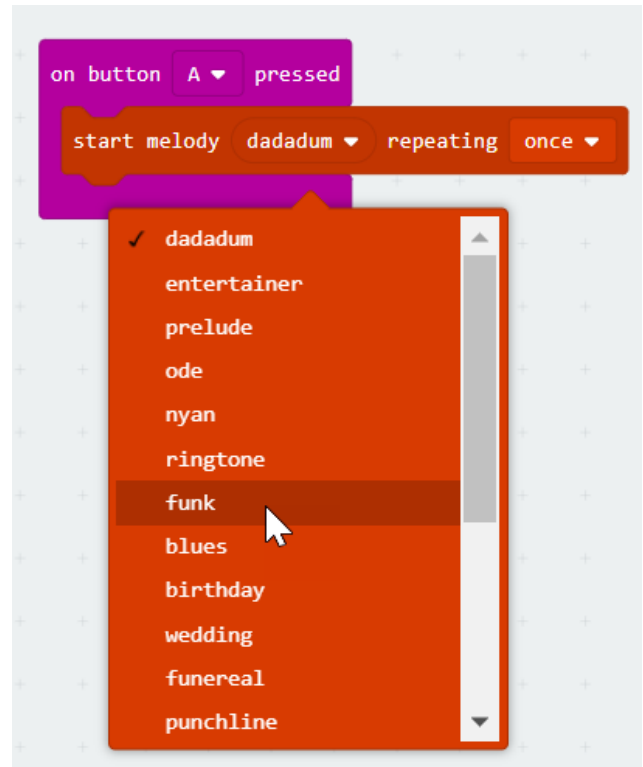




There are also blocks with premade music clips. Using the **start melody** \_\_\_\_ **repeating once**

block you can play a short melody from the list.

- > Attach this block to a **on button A pressed** block.
- > Click on the drop-down menu to see the options.
- > Choose one of the melodies.
- > Using the second drop-down menu on the block you can tell the GiggleBot to play the melody multiple times.
- > Download your new program and transfer it to your GiggleBot. Press the **A** button on the micro:bit and listen to the melody play.



Think about how you can use these blocks to further develop a character.



## > PLAN IT OUT

### PERSONALITY

- > Determine what kind of character your GiggleBot will be.
- > Is it a good character or an evil character?
- > Is it helpful? Curious? Mischievous?
- > Think about how you can give these characteristics to your robot.

### SOUNDS

- > Is your GiggleBot going to laugh while it drives around?
- > Will it play music?
- > Will it show emotions using the LEDs on the micro:bit and corresponding sounds?
- > Does it play a sound when it is turned upside down?
- > What else could you use sounds for?

### ADDITIONAL SENSORS

- > Do you want to use sensors from previous missions?
- > Does your character make sounds when it gets close to an obstacle (distance sensor)?
- > Does your character squeal when the lights go out and run around (light sensor)?

### APPEARANCE

- > How will you attach the speaker to the GiggleBot so that the robot can move around?
- > What is your character going to look like?
- > What materials will you use to create the character?
- > Jot down your ideas on a piece of paper before starting the build.

## BUILD YOUR CHARACTER

- > Using craft materials, transform your GiggleBot into a character.
- > Ensure that the speaker is securely attached and will not fall off when it is driving around.
- > Also make sure that the speaker, alligator wires, and decorations do not interfere with the movement of the wheels.

## BUILD YOUR PROGRAM

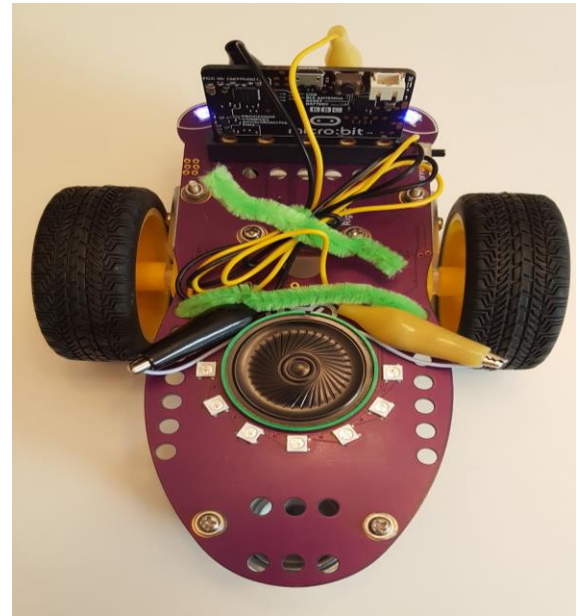
- > Look back at your plan notes and design a program to develop your character.
- > Think about all of the different ways to initiate your program or sounds.
- > Use the skills you learned in previous missions as well.

## > ARE YOU STUCK??

Let's build together! First, use a pipe cleaner to wrap up the alligator clip wires. Then, use another pipe cleaner to attach the speaker to the back of the GiggleBot. This will allow the GiggleBot to move around when the speaker is attached.

Here are my notes for a GiggleBot character:

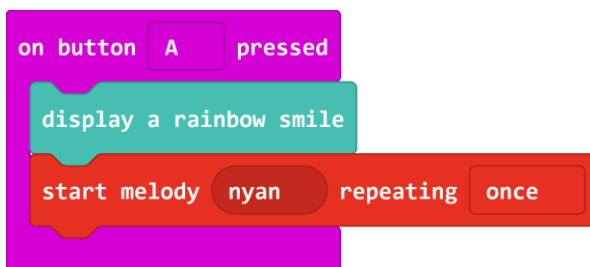
- > Silly creature that drives around crazily
- > Rainbow colors
- > Plays upbeat music while it drives
- > Laughs when it gets dark



## LET'S START CODING.

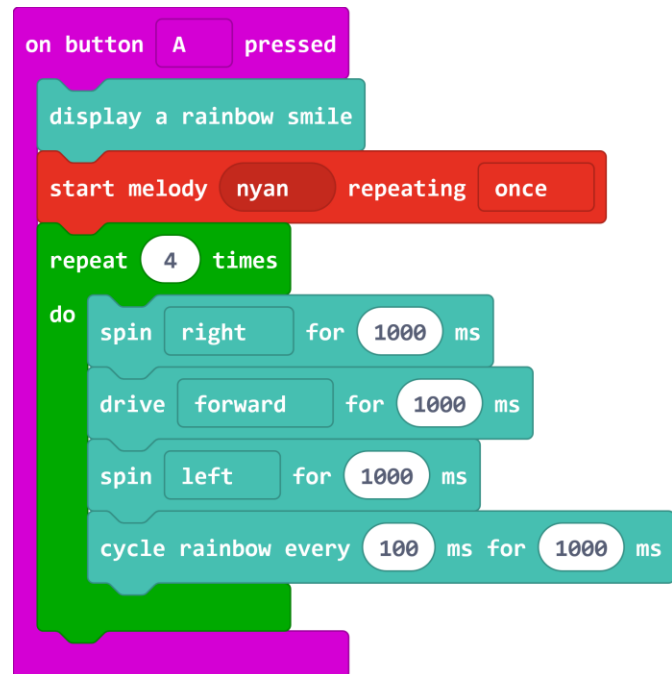


- > Move an **on button A pressed** block to the workspace. Next, add a **display rainbow smile** block to the **on button A pressed** block.



- > Then, pick a melody for the GiggleBot to play. I chose the *nyan* melody. It can repeat once or multiple times.

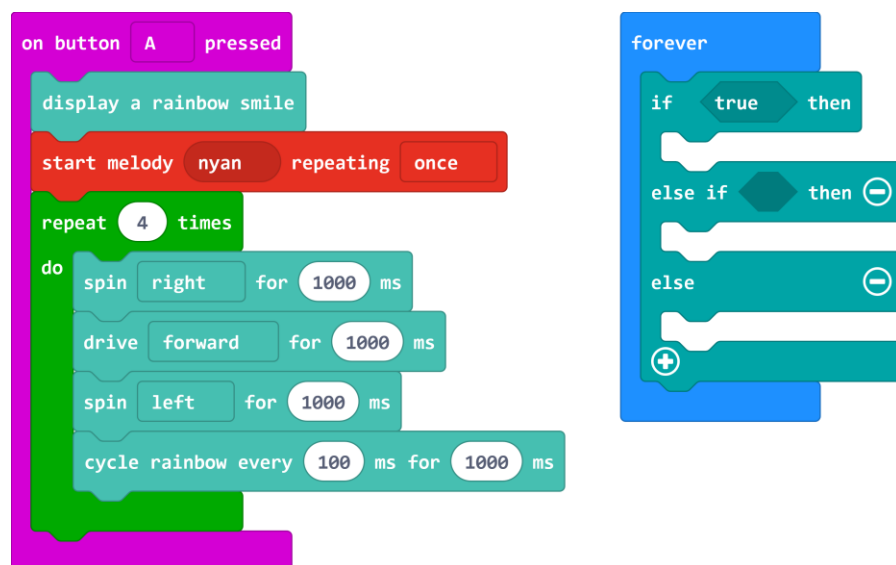
Last, let's add some movement. Choose blocks to tell the GiggleBot how to move. An example is shown here.



## SENSOR USE

Now that the GiggleBot is programmed to drive around to a melody, let's add in one of the built-in sensors. Since the character laughs when it gets dark we need to tell the GiggleBot to react to changes in light levels. We want the GiggleBot to do this any time the light changes, so move a **forever** block onto your workspace.

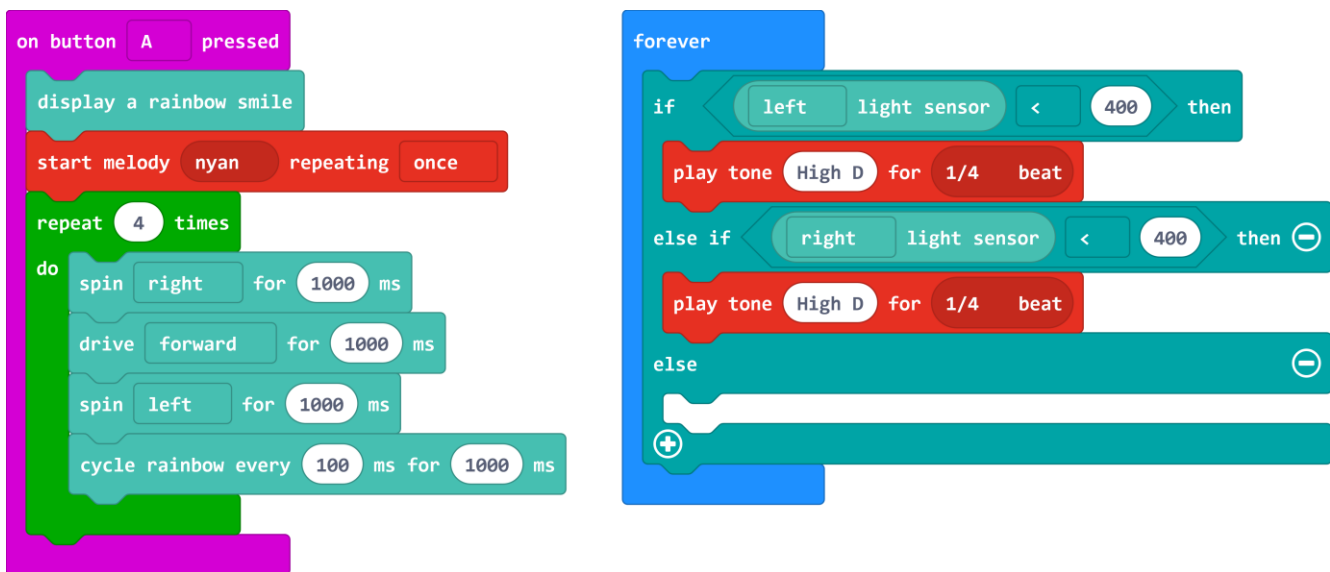
Add an **if \_\_ then \_\_ else \_\_** block. Click on the plus sign to change the block to include an **else if \_\_ then \_\_** statement.



Next, we need to determine what light level should be considered dark. For the example below a value of 400 is considered dark. The values for your room may be different. Remember that values for the light sensor range from 0 (dark) to 1024 (very bright). If you need to check the light levels in your room, look back for the directions in the Learn section of Mission 6.

Then, add blocks that will sound like laughter. In the example the tone High D is played for  $\frac{1}{4}$  beat. This will repeat whenever the light level is below 400.

Repeat this process for the other light sensor.



The **else** statement is left blank because we do not want the GiggleBot to laugh if the light level is greater than 400.

Download and transfer the program to your GiggleBot.

## > TRY IT OUT

- > Did your GiggleBot sound and behave like you planned?
- > Do you see any areas that you would like to modify or improve upon?

In engineering, we:

- > try a solution
- > think about what needs to change
- > and iterate

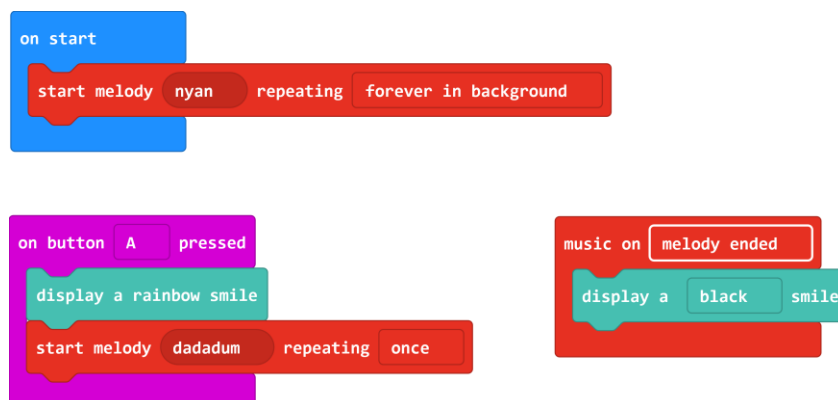
As you modify and revise your program, save each new program with a version number – you never know when you might want to take another look at an older version (for example: Speaker\_V1 where “V” stands for version).

## EXTENSION

- > How can you add more “personality” to your robot with servos, lights, and a speaker?
- > Build a “mouth” using the micro:bit leds and sync the sounds the robot makes with the mouth?

## PLAYING BACKGROUND MUSIC

The **start melody** block has the option to play music in the background. This music will be interrupted whenever music in the foreground is played and will be continued when the foreground music is done. There's also an event block to do some code when the foreground music ends. Experiment with this program and see if you want to change anything in your project.





Copyright Dexter Industries 2019. All rights reserved. Reproduction and distribution of the Mission without written permission of Dexter Industries is prohibited. GiggleBot is a registered Trademark of Dexter Industries.

Contact [dexterred@dexterindustries.com](mailto:dexterred@dexterindustries.com) for permissions and questions.