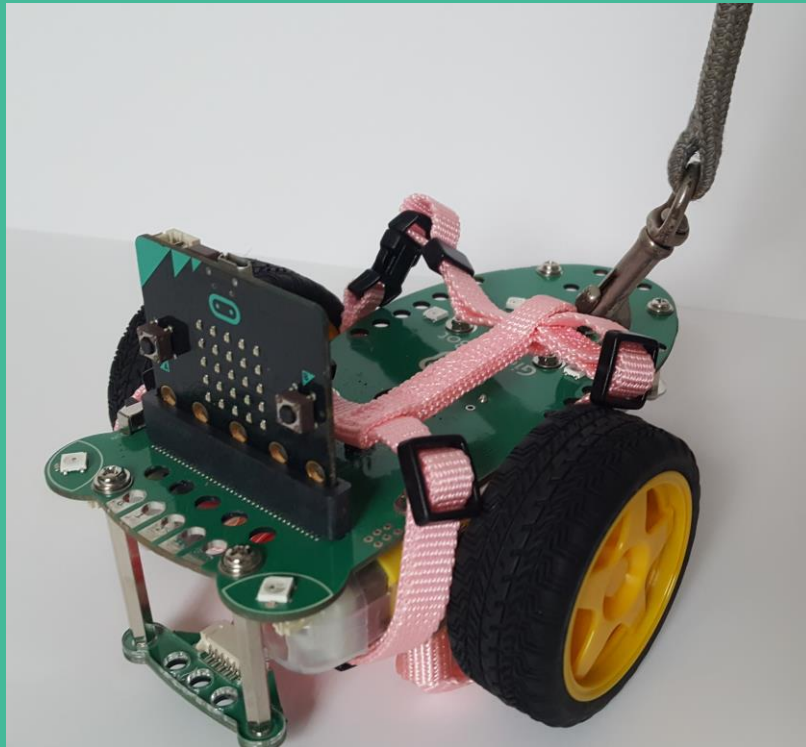


ROBOT PET

MISSION 3



Giggle
Bot

TABLE OF CONTENTS

• ROBOT PET	1
Your Role: Animal Trainer	1
Your Task: Build a Leash to Control the GiggleBot	1
Considerations	1
Materials	1
• LEARN: PINS	2
First Pin Use	3
Two Pins use	4
• LEARN: MAKING A CONTROLLER	5
• PLAN IT OUT	6
Build Your Controller	6
Build Your Program	6
• ARE YOU STUCK??	7
Write the Program	7
Build the Controller	8
Connect It All Together	9
• TRY IT OUT	10
Iterate	10
Extension	10



> ROBOT PET

YOUR ROLE: ANIMAL TRAINER

i **Animal training** is the act of teaching animals specific responses to specific stimuli. Training may be for purposes such as companionship, detection, protection, and entertainment. The type of training an animal receives will vary depending on the training method used, and the purpose for training the animal. For example, a seeing eye dog will be trained to achieve a different goal than an animal in a home.

YOUR TASK: BUILD A LEASH TO CONTROL THE GIGGLEBOT

You will transform your GiggleBot into an animal that can be controlled using a leash controller that you will build and program.

CONSIDERATIONS

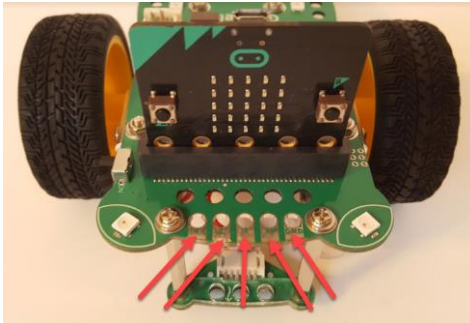
- > What movements do you want the GiggleBot to do?
- > What will the controller be made out of?
- > What will the controller look like?

MATERIALS

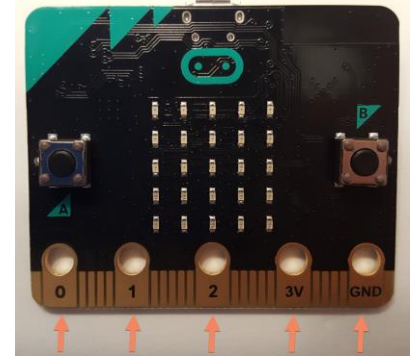
- > GiggleBot and good batteries
- > micro:bit and provided cable
- > Laptop / computer
- > Alligator clips
- > Craft supplies to build a controller for the GiggleBot to include conductive materials (ex. cardboard, aluminum foil, salt dough, construction paper)

> LEARN: PINS

Take the micro:bit out from the GiggleBot and look at the bottom edge. There are external connectors on this edge labeled **0**, **1**, **2**, **3V**, and **GND** called **pins**.



Put your micro:bit back into the GiggleBot and look at the front of the robot between the two LED eyes.



The same 5 pins on the micro:bit are accessible on the front of the robot. These pins are labelled **0**, **1**, **2**, **3V**, and **GND** on the front of the GiggleBot as well.

Pins **0**, **1**, and **2** can be used for a variety of things and are called *general purpose input and output* (**GPIO**) pins. You can use your alligator clips to connect these to different things.

The **3V** (3 volt) pin can be used to power additions to your micro:bit such as an LED, buzzer, or speaker.

ⓘ Caution: The 3V and GND pins should NOT be connected together. Connecting them can cause a “short” and damage your micro:bit board.

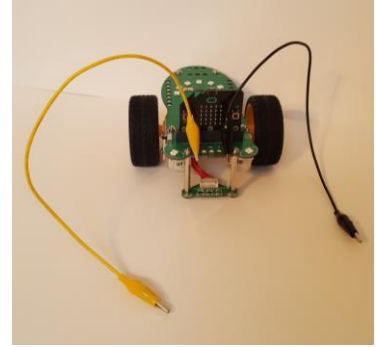
The GND (ground) pin is needed when completing a circuit and connects to the electrical ground (not quite the same thing as the actual ground you stand on). We are going to focus on pins 0, 1, 2, and GND today.

ⓘ Pay attention when handling the alligator clips! They can bite. Thankfully these alligators will not lead to permanent injuries but they can inflict a bit of pain if you are not careful.

FIRST PIN USE

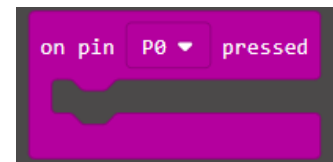
Let's try it out! Connect one of your alligator clips to the **0** pin and another alligator clip to the **GND** pin.

Instead of using an “**on button A pressed**”, “**on start**”, or “**forever**” block, we are going to try new input blocks.

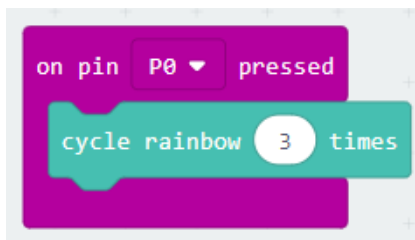


> Under **Input**, find the **on pin P0 pressed** block and move it to the workspace. Now we need to decide what we want the GiggleBot to do when **pin 0** is pressed.

- Do you want it to display a rainbow?
- Spin around?
- Show an icon on the micro:bit?



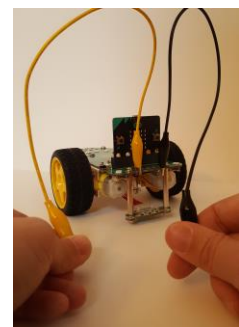
Put blocks inside of the “**on pin P0 pressed**” block to tell the GiggleBot what to do. You can add one block or multiple blocks.



In this example, the GiggleBot will cycle a rainbow three times on the smile LEDs when **P0** is pressed.

Download and transfer your program to the GiggleBot.

Turn on your GiggleBot. With one hand, touch the metal part of the loose end of the alligator clip connected to **P0**. With your other hand, touch the loose end of the alligator clip connected to the **GND** pin. This will complete the circuit and start your program. You should see the the rainbow display three times if you used the above example.



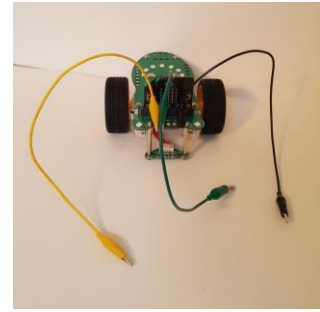
IT DIDN'T WORK?

Check that your alligator clips are making a good and solid connection. Make sure your skin touches the metal part of the clip.

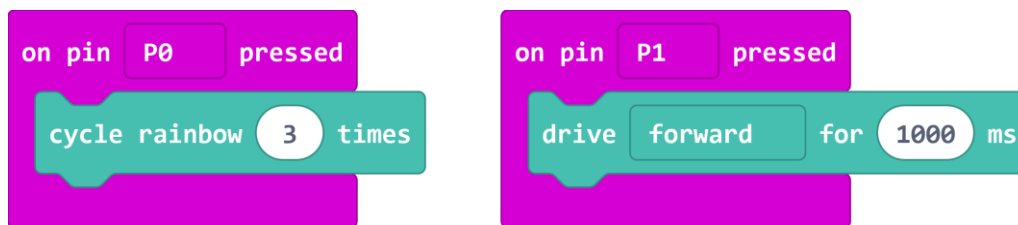
TWO PINS USE

Let's add another input to our program!

- > Attach a new alligator clip to **P1**. What do you want the GiggleBot to do if **P1** is pressed?
- > Go back to your program and add a new block that controls **P1** (use the dropdown in a **on pin P0 pressed** block).



In this example we have added the following behavior: the GiggleBot will move forward for one second when **P1** is pressed.



Download and transfer your program to the GiggleBot. Turn on your GiggleBot and try it out. Remember you need to touch the alligator clip connected to the pin AND the alligator clip connected to **GND**.

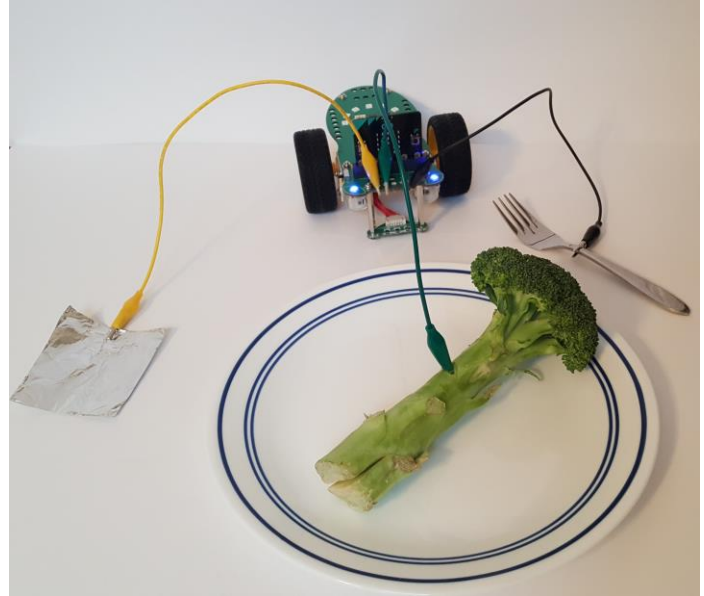
WHY DOES THIS WORK?

By touching the two alligator clips you are completing the circuit because YOU conduct electricity. You are a conductor! Think about other materials that conduct electricity: instead of only using yourself as the conductor, you could use a fork, a glass of water, or some aluminum foil (try them!). We can use other materials to create buttons instead of just using the ends of the alligator clips.

> LEARN: MAKING A CONTROLLER

What conductive materials do you have? You can use things like:

- > aluminum foil
- > salt dough
- > bananas
- > apples
- > broccoli
- > a metal fork
- > a key
- > coins
- > pencil graphite



or anything that conducts electricity to create buttons for each of the pins. Experiment with what you have.

Keep the alligator clips attached to the GiggleBot. On the other end of each alligator clip attach one of the conductive materials.

Turn on your GiggleBot and try out your new controller. Remember that you need to touch the **GND** pin and one of the other pins to complete the circuit. You can also move the objects so they touch directly, removing you from the circuit. Can you think of ways you could build a foot pedal?

> PLAN IT OUT

Decide what actions or movements you want your GiggleBot pet to do. You can use P0, P1, and P2. This means that you can have three different movements or sequences of movements and actions.

Think about what materials you want to use to make a controller so that you can walk around with your pet robot and you are not stuck at the table or on the floor. You can use a piece of cardboard to make a base for your controller and then add on conductive materials.



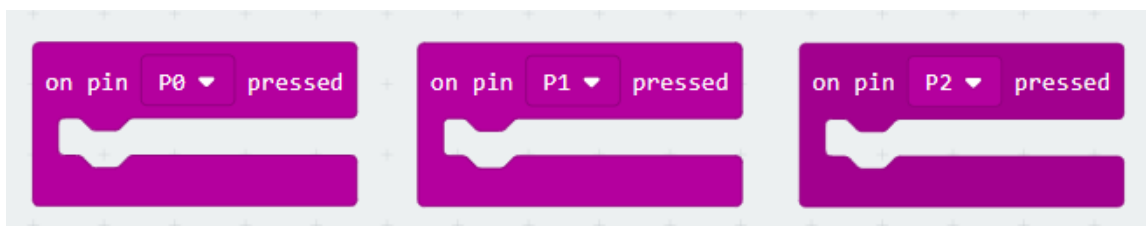
Pro tip: Create a wristband for the GND pin so that you only need to touch the P0, P1, and P2 pins. The GND pin will already be connected since it is touching your skin! One way to do this is to put a piece of foil or dough underneath a bracelet made from a pipe cleaner.

BUILD YOUR CONTROLLER

Create your controller. Be sure to have buttons for **P0**, **P1**, **P2**, and **GND**. Check to make sure that the connections are secure so that you can walk around with your new robot pet.

BUILD YOUR PROGRAM

You can add on to the program you created in the Learn section or create a completely new one. Put blocks inside each of the **on pin ____ pressed** blocks to program your controller.



Once you are finished with your program, download it and transfer it to your GiggleBot.

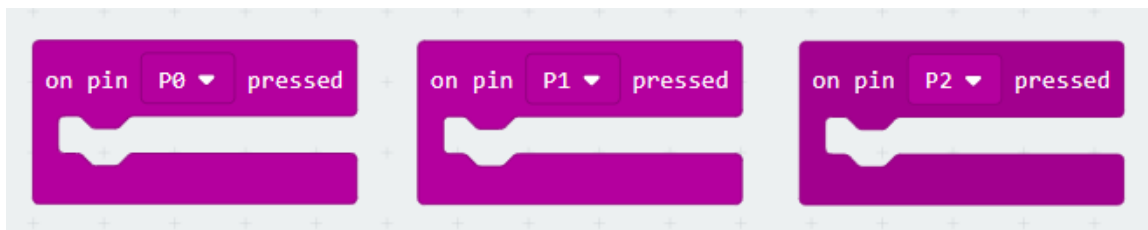
> ARE YOU STUCK??

Let's build together!

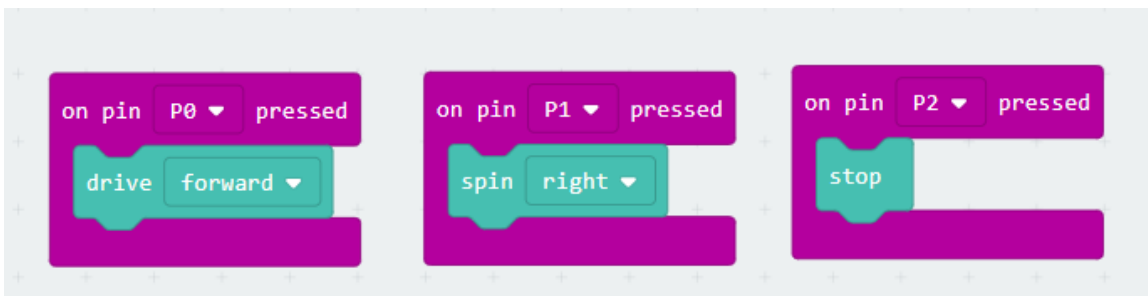
We are going to create a controller that allows us to control when the GiggleBot moves forward, spins, and stops. This will allow us to take the GiggleBot for a walk around the neighborhood just like a pet dog.

WRITE THE PROGRAM

First, move three "on pin __ pressed" blocks to the workspace. Using the dropdown menu on each block, change the blocks to read **P0**, **P1**, and **P2**.



Next, we will tell the GiggleBot what to do when each pin is pressed. When **P0** is pressed the GiggleBot will drive forward. When **P1** is pressed, it will spin right and when **P2** is pressed it will stop.



You can modify this basic framework to include lights, speed, or other movements.

BUILD THE CONTROLLER

Next, we are going to create the controller. You will need the following materials:

- > cardboard
- > conductive material (such as aluminum foil or salt dough)
- > scissors
- > tape or glue
- > marker or pen
- > 4 alligator clips (to connect it to the GiggleBot)

Cut out the cardboard in your desired controller shape. The example below is taped together around the edges.

Create four buttons for your controller using aluminum foil, dough, or another conductive material. Glue the pieces onto the cardboard. Remember: You need to be touching the GND pin at all times. Put it in a place that is convenient for holding the controller. This is where having built a wristband can be of help.



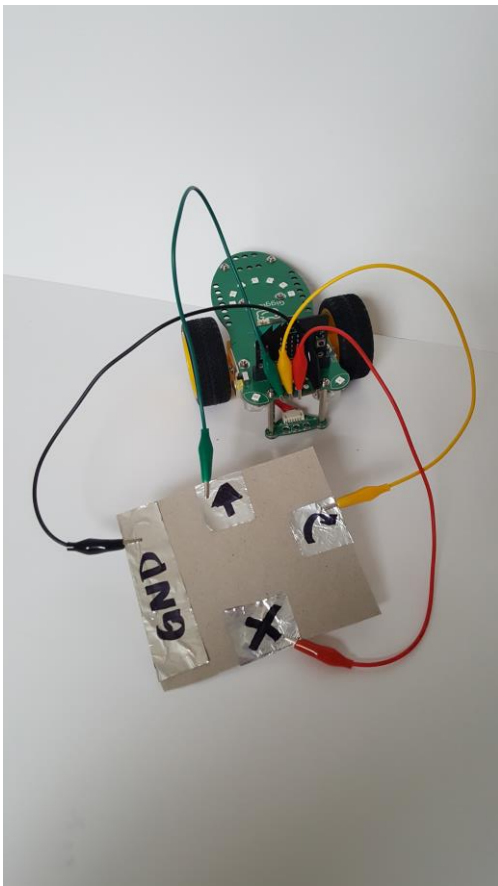
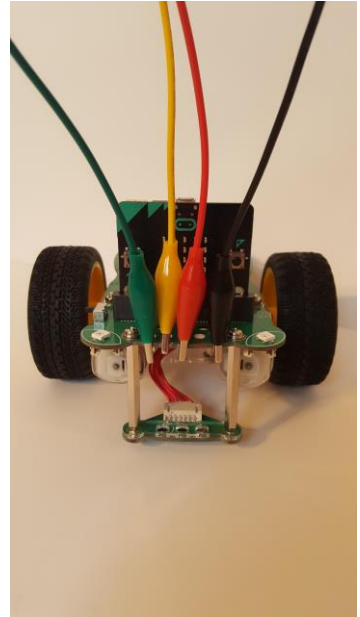
Using a marker or pen label each of the buttons so that you know what it does.

CONNECT IT ALL TOGETHER

Now, we will connect the controller to the GiggleBot. Attach alligator clips to P0, P1, P2, and GND).

Connect the other end of each alligator clip to the correct button on the controller based on your program. If you used the program above:

- > P0 connects to forward
- > P1 connects to spin right
- > P2 connects to stop
- > GND connects to GND



> TRY IT OUT

Ensure that all of your alligator clips are securely attached to your controller and GiggleBot. Then, turn on your robot and test out your program and controller. Since your controller is tethered to the GiggleBot, remember that you will need to move with the GiggleBot as it drives around, just like walking a pet. Maybe you can even take the GiggleBot for a stroll around your neighborhood or school.

ITERATE

Did your GiggleBot follow your commands via the controller? Did your controller work well? Do you see any areas that you would like to modify or improve upon?

In engineering, we:

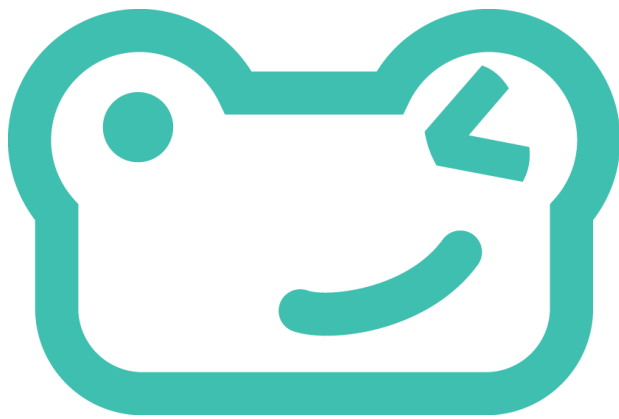
- > try a solution,
- > think about what needs to change,
- > and iterate.

We almost never get the design right on the first try!

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: Pet_V1 where “V” stands for version).

EXTENSION

- > What other uses for the controller can you think of?
- > How can you incorporate ideas from missions 1 and 2 into this program?
- > What other behaviors can the robot have?



Giggle Bot

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 dextered@dexterindustries.com for permissions and questions.