VEXcode EXP Python Resource for the Team Freeze Tag STEM Lab Unit

A Note about Implementing VEX EXP STEM Lab Units with Python:

STEM Labs are designed to be an interactive Unit of instruction that you can use with your students to implement VEX EXP in your setting. The student-facing content of the STEM Labs is written using VEXcode EXP Blocks, but the flexibility of the STEM Lab Units enables you to teach them using VEXcode EXP Python with ease.

This Python Resource is the teacher-facing Python companion to the Facilitation Guide for this Unit (like a teacher's manual appendix) providing the materials needed to be able to complete this Unit using VEXcode EXP Python successfully. To have your students dive deeper into coding with Python, see the Computer Science Level 1 - Python course for VEXcode VR. If you want to personally learn more about teaching Python, see the Computer Science Level 1 - Python Educator Certification.

For more detailed information about implementing STEM Labs in your classroom, visit VEX PD+ for videos, expert tips, and more.

Please Note: Python can introduce additional complexity to this Unit. We suggest beginning programmers start with Blocks-based projects, as this STEM Lab Unit is not focused on teaching coding.

VEXcode EXP Python Resources – Unit Overview

In the Team Freeze Tag Unit, students will build the BaseBot and learn to drive it with the Controller, in order to play a game of Team Team Freeze Tag in the Unit Competition.

- In **Lesson 1: Introduction** students will build the BaseBot and be introduced to the challenge of Team Freeze Tag.
  - To teach this Lesson with Python, no additional materials are needed.
- In **Lesson 2: Driving with the EXP Controller**, students will test different driver configurations on the Controller, as they learn to drive the BaseBot via remote control.
  - To teach this Lesson with Python, no additional materials are needed.
- In **Lesson 3: Changing the Wheels**, students will test various wheel combinations to learn about the different types of wheels that can be used on the BaseBot, and how the wheels affect the movement of their robot.
  - To teach this Lesson with Python, no additional materials are needed.
- In **Lesson 4: Adding the Bumper Switch and Printing to the Brain Screen**, students will continue to adapt their BaseBot by adding a Bumper Switch and Touch LED so that their robot is ready to play Team Freeze Tag!
  - Python versions of the Using the Bumper Switch, Printing to the Brain Screen, and Assigning a Drivetrain to Controller Joysticks Lesson Summaries are included below.
    - **Note**: The Python project linked in the Practice and Compete documents has the Controller already configured. Students will not need to go through the process of Assigning a Drivetrain to Controller Joysticks. They can adjust the configuration if needed.
  - Python versions of the Bumper Press Practice Activity and One-on-One Freeze Tag Challenge Activity are included below.

        - **Note:** In these activities, a specific VEXcode EXP Python project has been linked that should be downloaded to students' devices, and opened in VEXcode EXP in order to download it to the robot. For information about opening projects in VEXcode EXP Python, see these articles.

- In **Lesson 5: Team Freeze Tag Competition**, student groups will join together for a classroom competition of Team Freeze Tag, applying what they have learned in the previous Lessons to try to win the game.
    - A Python version of the Team Freeze Tag Competition is included below.

        - **Note:** In this activity, a specific VEXcode EXP Python project has been linked that should be downloaded to their device, and opened in VEXcode EXP in order to download it to their robot. For information about opening projects in VEXcode EXP Python, see these articles.

- **Lesson 6: Conclusion** wraps up the Unit by introducing students to STEM careers related to the learning they did in this Unit, like Race Engineer; then engages students in a debrief conversation to share their learning, and reflect on their experiences.
    - To teach this Lesson with Python, no additional materials are needed.

Python-specific troubleshooting and support articles for this Unit can be found in the VEXcode EXP section of the STEM Library.

See the Facilitation Guide for the Team Freeze Tag Unit for Lesson setup information, teacher notes, and additional facilitation support.

Name: _____ Date: _____

VEX EXP Using the Bumper Switch

In this Lesson: You learn about what the Bumper Switch is, how it can be configured in VEXcode EXP, and how it can be configured in VEXcode EXP, and how it can be used in a project to make your robot accomplish a task.

Definition(s):

- **Bumper Switch:** a simple, mechanical sensor that reports if it is pressed or not pressed (released).

Notes:

- A Bumper Switch is a kind of sensor, which allows the robot to interact with its environment, and report if it is pressed or not pressed.
- A Bumper Switch can be triggered by an element of the environment, like a wall or another robot, or by a user, like pressing the Bumper Switch by hand.
- To configure the Bumper Switch in VEXcode EXP:
  - In VEXcode EXP, open the Devices Window
  - Select ‘Add a Device’
  - Select ‘3-Wire’
  - Select ‘Bumper’
  - Select the port on the Brain that your Bumper Switch is plugged into
  - Select ‘Done’
- Commands that can be used with the Bumper Switch will be added to the Toolbox in VEXcode EXP Python when the Bumper Switch is added, including:

```
bumper.pressing()   bumper.pressed(callback)
```

Note: *To learn more about each of these commands, you can access the Help in VEXcode EXP.*

Name: _____ Date: _____

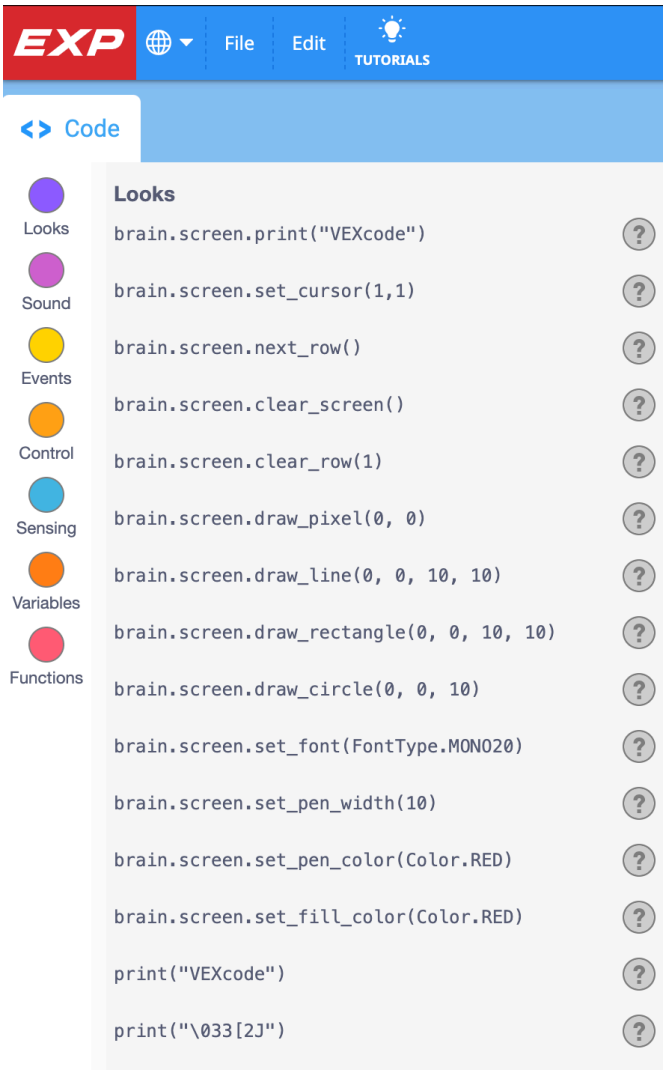
VEX EXP Printing to the EXP Brain Screen

In this Lesson: You learn about what can be printed on the VEX EXP Brain Screen, and how to use VEXcode EXP in order to print to the Brain screen.

Notes:

- With VEXcode EXP, you have the ability to print text, colors, values, and shapes directly to the Brain screen on your robot

- Printing to the Brain screen is a useful tool for many things, including:
    - customizing your robot
    - indicating what team your robot is on
    - indicating that something has occurred as your robot is running (i.e. turning a certain color when the Bumper Switch is triggered, or increasing a counter each time a Buckyball is in the intake)



Using VEXcode EXP to Print to the Brain screen

- Print blocks are automatically populated in the Looks category of the Toolbox when VEXcode EXP is opened (like those shown in the image to the right from the video)

- Print blocks can accept text or reporter values to print values from the robot

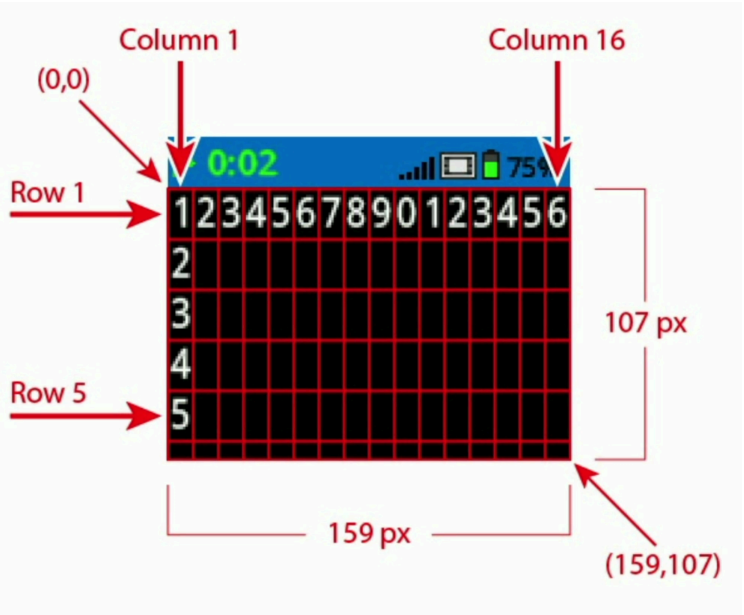
- You can change the color of what is printed to the screen from the default value of white using the ***set_pen_color*** command.

- The Looks section of blocks enables you to draw lines and shapes on the screen, using location coordinates on the Brain screen and pixel values, as shown in these examples from the video.

- ***brain.screen.draw_line(START_X, START_Y, END_X, END_Y)***
- ***brain.screen.draw_rectangle(X, Y, WIDTH, HEIGHT)***

Locations on the Brain Screen

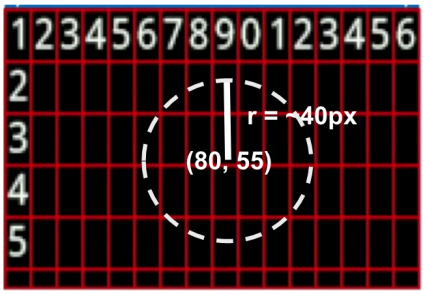
- The Brain Screen is divided into rows and columns, as well as an X, Y coordinate grid using pixel (px) values.
  - The top left corner of the screen has a coordinate value of (0,0)
  - The bottom right corner of the screen has a coordinate value of (159, 107)
- These values can be used to set the location on the screen in which your text, color, shape, or value is printed.

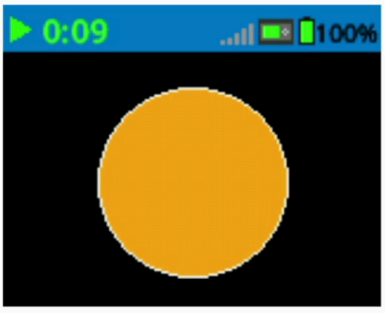


- In the example below from the video, the values in the ***draw_circle*** command are estimated on the Brain screen, and the end result of what is printed on the screen is shown on the right.

```
# Begin project code

brain.screen.set_fill_color(Color.ORANGE)
brain.screen.draw_circle(80, 55, 40)
```





Printing Changing Values on the Brain Screen

- You can create a project to print changing values on the Brain screen, as shown in this example project from the video.
  - In this project the font size and color have all been adjusted

- The ***while True*** command sets the cursor to a specific location on the screen, and repeatedly prints values every 0.1 seconds, essentially creating a timer on the screen.

```
brain.screen.set_font(FontType.MONO60)
brain.screen.set_pen_color(Color.PURPLE)

while True:
    brain.screen.set_cursor(1, 1)
    brain.screen.print(brain.timer.time(SECONDS))
    wait(0.1, SECONDS)
    brain.screen.clear_screen()
    wait(5, MSEC)
```

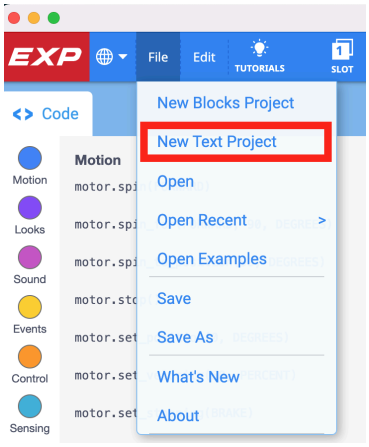
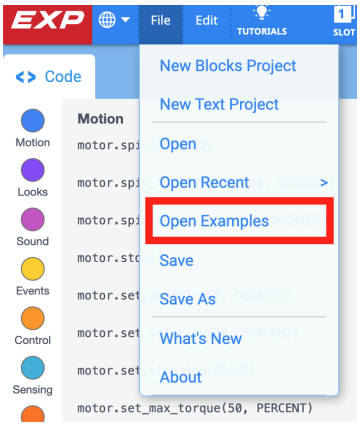
Note: *To learn more about each of these blocks, you can access the Help in VEXcode EXP.*

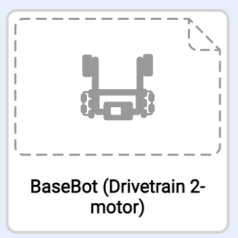
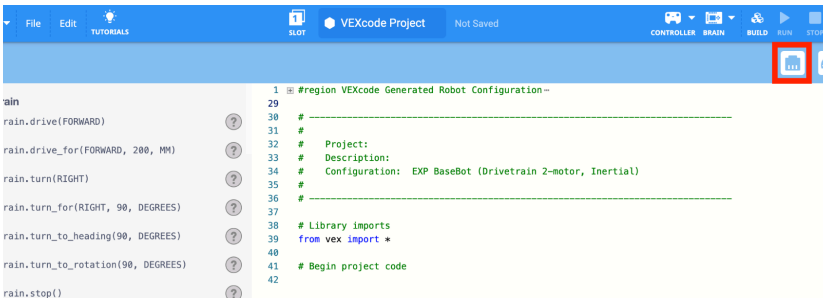
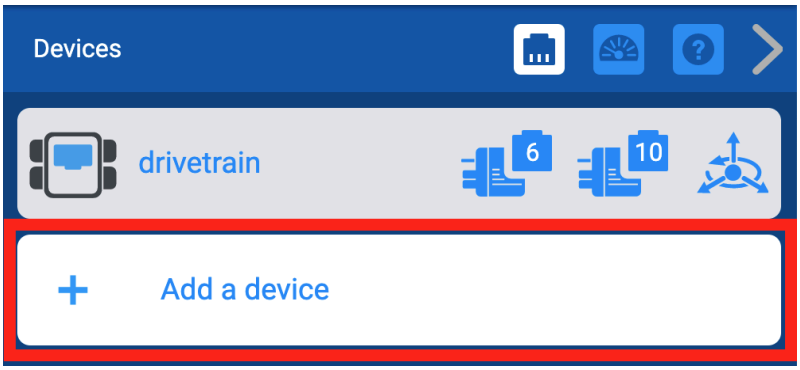
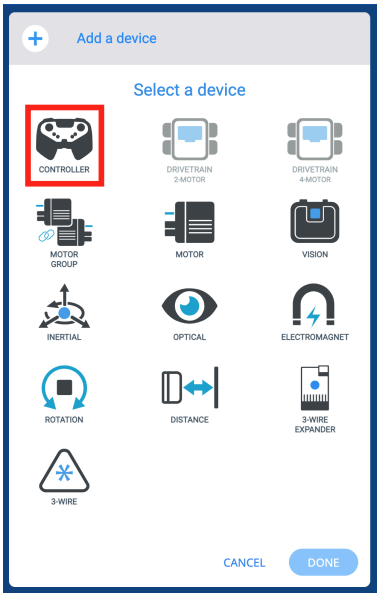
Name: _____ Date: _____

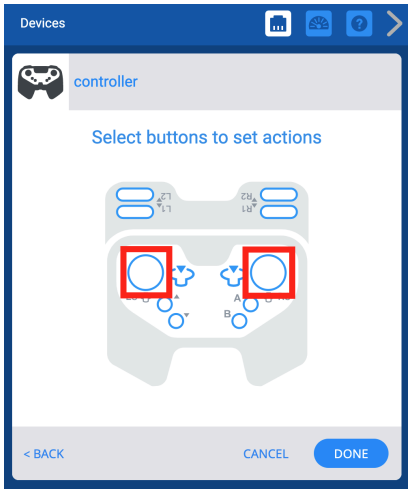
VEX EXP Assigning the Drivetrain to the Controller's Joysticks

In this Lesson: Learn how to assign the Drivetrain to the Controller's Joysticks in order to drive the BaseBot.

How to Assign the Drivetrain to the Controller's Joysticks:

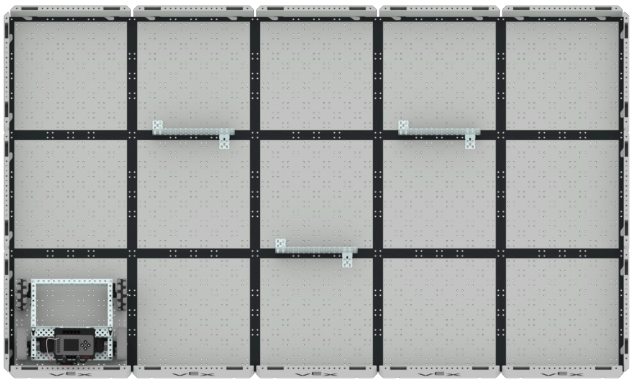
1. In VEXcode EXP, select "File" then "New Text Project"	
2. Select "Python"	
3. Select "File" then "Open Examples"	

4. Select “Templates”	
5. Select “Basebot (Drivetrain 2-motor)”	
6. Open the Devices Window by selecting the Port Icon in the upper right corner.	
7. Select “Add a Device”	
8. Select “Controller”	

9. Choose a drive mode by clicking on the joystick icon. You will see the choices of: a. Left arcade b. Right arcade c. Split arcade d. Tank drive	
10. After selecting a drive mode, select “Done”	

11. Name your program, and save it!
12. Download your project to the robot’s Brain.

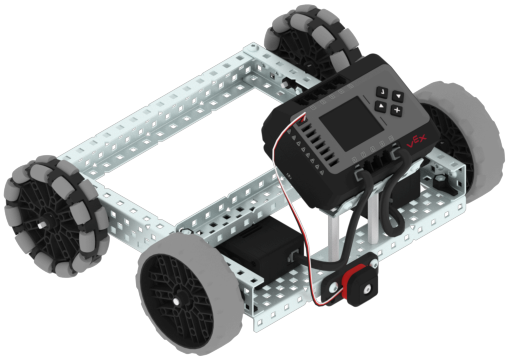
VEX EXP Practice Activity



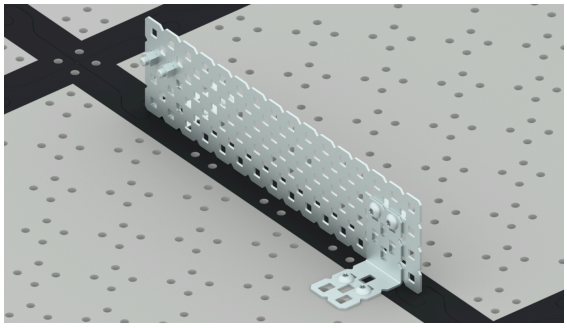
Bumper Press

Practice driving through plates!

How to Practice



1. Ensure your robot has a Bumper Switch mounted on the back and a 1x2x1x16 C-Channel attached to the front. See the image below.
2. Download this project to the EXP Brain. In this project, when the Bumper Switch is pressed, the robot stops driving, and the Brain screen turns red for three seconds. Then the robot can continue driving.
3. Your 3’ x 5’ Field should have three 4x16 plates with brackets on either side screwed to the Field, as in the image below. You will need to use two 2x16 plates screwed together for the third obstacle if you are using a single EXP kit.



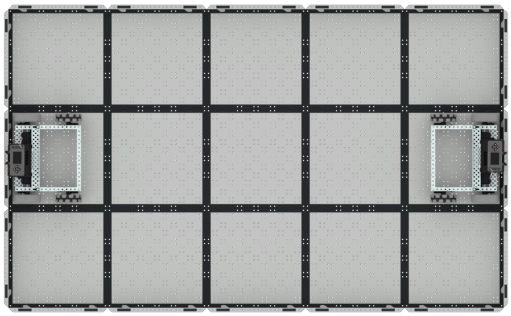
4. Your robot should start in the bottom left corner as shown in the top left image.
5. Have each member on your team drive the robot through the plates on the Field and try not to trigger the Bumper Switch.
6. After each of you drive through the plates, record how many times your Bumper was pressed and how quickly you navigated the course. How easy was it to maneuver your robot? Write down the results in your engineering notebook and use this information to decide who will be the driver in future challenges.

Pro Tip

Change your Controller Configuration

- Change your controller configuration in VEXcode EXP so that you can drive with your preferred configuration (left arcade, right arcade, split arcade, tank drive).
- Ensure that the Bumper Switch is installed so that it sits as high as possible on the back of the robot.

VEX EXP Challenge Activity



One-on-One Freeze Tag

Play a game of one-on-one freeze tag!

How to Play

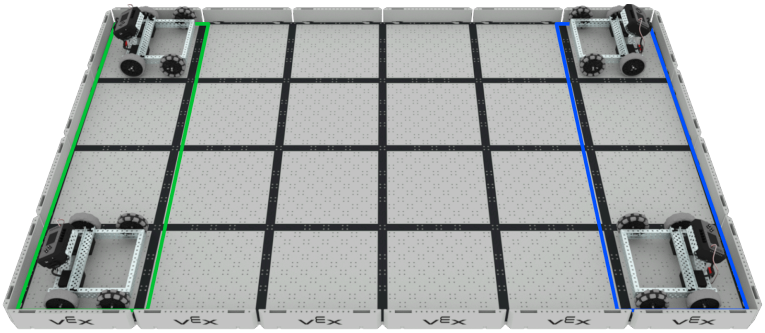
- **Check your Field and Robot Setup:**
  - Your Field for the Challenge Activity should match the image shown above.
  - Each robot must start with the Bumper Switch against the back wall. Ensure that each robot has a 1x2x1x16 C-Channel attached to the front and the Bumper Switch is on the back.
    - **Note:** Ensure that the Bumper Switch is installed so that it sits high, so that it can make contact with the C-Channel from another BaseBot.
  - Ensure your robot has this project downloaded. One robot should have the color ‘green’ on their Brain screen and the other should have the color ‘blue.’
  - Choose the controller configuration most comfortable for you in the Devices window and download the project to the Brain.
- **One-on-One Freeze Tag Challenge rules:**
  - Each match is one minute.
  - Every time your Bumper Switch is pressed, the other robot gets a point.
    - Your controller will be disabled for 3 seconds after being tagged.
  - The robot with the most points at the end of 60 seconds, wins!

Pro Tip

Record information from each match. What patterns can you find when you look at the data? Can you use this in your game strategy?

- Which controller configuration did they use?
- Which wheel combination did they use?
- How many points did they earn?

VEX EXP Competition Activity

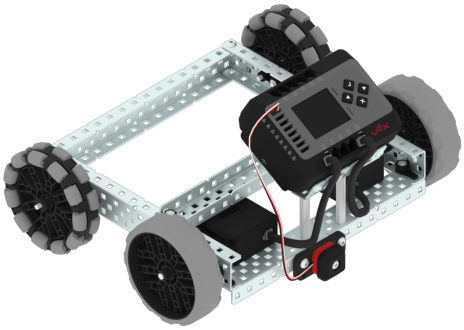


Team Freeze Tag

Play 2-on-2 Freeze Tag!

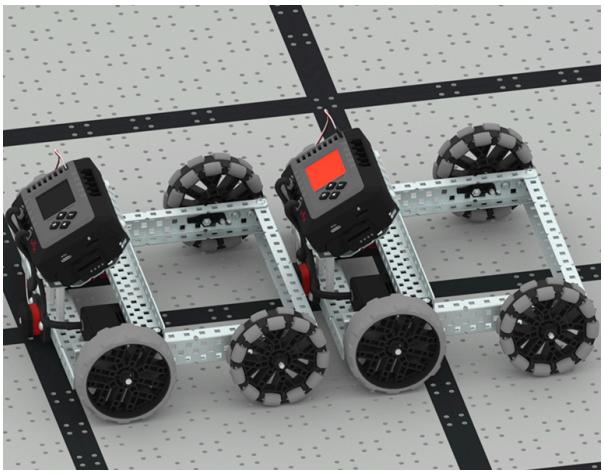
How to Play Team Freeze Tag:

1. **Check your Field and Robot setup:**
    - Your field for the challenge should match the image shown above.
    - Ensure your robot has this project downloaded. One team of two robots should have the color “green” on their Brain screen and the other team should have the color “blue.”
    - Choose the controller configuration most comfortable for you in the Devices window and download the project to the Brain.
2. **Team Freeze Tag Challenge rules:**
    - Must have a Bumper Switch on the back of the Robot, as shown in the image below.



    - Your robot must fit inside of a square on the Tiles (approximately 30cm by 30cm, or 12" by 12").
    - Each match is 1 minute long and is played by two teams (with two robots each) on a 4' by 6' field.
    - During the match, your robot must remain in motion unless you are tagged.
    - No trapping other robots into a corner. Any tags made under this condition do not count as points.
    - You may only tag another robot twice in a row. After that, the tagging robot must retreat at least one field tile (12 inches) away before tagging the same robot again.

- Each time your robot is tagged, the other team earns 1 point. The team with the most points at the end of the match, wins.
    - A tag is whenever the Bumper Switch on a robot is pressed.
- When tagged, the Brain screen will display red and your controller will be disabled for 3 seconds.



- If your Bumper Switch is pressed and triggered by another obstacle (i.e. Walls, the other robot on your team), the other team will earn 1 point.