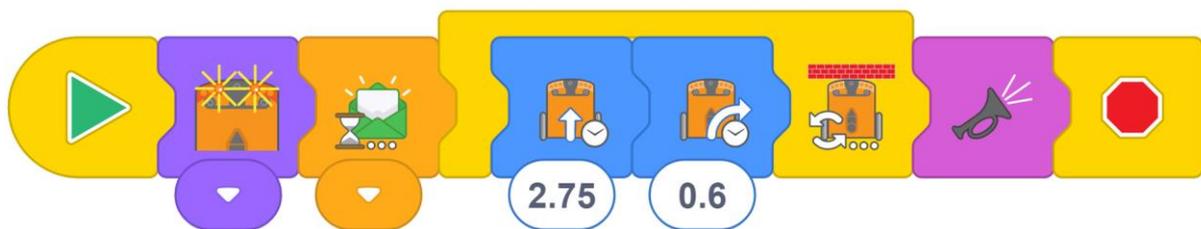


Teacher's guide to EdBlocks

Teaching compendium and answer key for the EdBlocks lesson activities



The EdBlocks Lesson Activities Collection by [Emma Dewar and Kat Kennewell](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Contents

About this guide	3
Before you start.....	4
Activity 1 – Meet Edison	10
Activity 2 – Clap controlled driving	11
Activity 3 – Avoid obstacles.....	12
Activity 4 – Follow a torch.....	13
Activity 5 – Line tracking.....	14
Activity 6 – Bounce in borders.....	15
Activity 7 – Sumo wrestle	16
Activity 8 – Welcome to EdBlocks	17
Activity 9 – Let’s download a program.....	18
Activity 10 – Let’s drive.....	19
Activity 11 – Let’s turn	21
Activity 12 – Let’s try a maze.....	22
Activity 13 – Let’s follow a line.....	24
Activity 14 – Let’s stop at a black line.....	26
Activity 15 – Let’s make noise	27
Activity 16 – Let’s use the lights	29
Activity 17 – Let’s detect obstacles	31
Activity 18 – Let’s detect and avoid	33
Activity 19 – Let’s stay in the borders.....	34
Activity 20 – Let’s make music	36
Activity 21 – Let’s perform	38
Activity 22 – Let’s avoid the light	39
Activity 23 – Let’s have a dance party.....	40

About this guide

This guide offers teachers and instructors overviews, activity extensions and other supporting information for the EdBlocks lesson activities available at <https://meetedison.com/robot-programming-software/edblocks/>.

The EdBlocks activity worksheets are designed to allow students to work through activities independently, gradually learning about both the Edison robot and EdBlocks. This guide offers further information for teachers and instructors to help make teaching EdBlocks easy and fun.

Each EdBlocks activity is included in this guide along with any relevant supporting information for that activity. Supporting information is divided into the following sections:

How it works

Information about how the activity, or the science behind the activity, work.

Tips and tricks

Helpful hints and ways to overcome common issues.

Activity extensions

Optional extension activities and lesson ideas to complement the activity worksheets.

Answer key

Solutions and/or example answers to the 'find the answer' sections of the activity.

Creative Commons licence attribution details

The EdBlocks Lesson Activities Collection is comprised of the EdBlocks lesson activities and this guide. The collection is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Activity development: Emma Dewar and Kat Kennewell

Instructional design: Kat Kennewell

Illustrations: Emma Dewar

Teacher's guide: Kat Kennewell

Before you start

Before using Edison with your students, you will need to set up the programming devices, (i.e. the computers, laptops, Chromebooks or tablets) you will be using with the EdBlocks app and get the Edison robots ready.

This guide shows the basics and troubleshooting help for getting set up to use Edison with EdBlocks. Additional details, including information about Edison's other programming languages, can be found in the free *Getting started with Edison guide* available at <https://meet Edison.com/content/Get-started-with-Edison-guide-English.pdf>

Get Edison ready

To get Edison ready for use, you need to:

1. Open the battery compartment at the back of Edison and remove the EdComm programming cable.
2. Insert 4 'AAA' batteries. Please refer to the picture to ensure that the batteries are inserted correctly. Be sure to reclose the battery case by clipping the battery cover back on.



Ensure the batteries are in the right way.

Please note: Low or flat batteries can cause a range of issues with Edison. For this reason, always use fresh, fully charged batteries in your robots.

Choosing batteries: If using disposable batteries with Edison, only ever use alkaline batteries. (These are the most common standard AAA batteries you will find in just about any shop.) If you are using rechargeable batteries with Edison, only use nickel metal hydride (NiMH) rechargeable batteries. Never use lithium rechargeable, heavy-duty disposable, super heavy-duty disposable or carbon zinc batteries.

3. To turn Edison on, flip the robot over. Slide the power switch to the 'on' position, as shown in the picture. This will turn Edison on, and the red LED lights will start flashing.



Push the switch towards the 'on' symbol.

Please note: While Edison will turn off automatically if not used after five minutes, we recommend you turn the robots off manually when not in use.

Set up your programming devices for EdBlocks

The best way to set up your programming devices is to run a test program in EdBlocks. Follow these six steps to test EdBlocks on your device:

1. Load the EdBlocks app by opening www.edblocksapp.com in a browser (we strongly recommend Google Chrome¹). Launch the programming app by pushing the orange 'Launch EdBlocks' button. Make sure you allow pop-ups for www.edblocksapp.com.
2. Once the app opens, you will see the programming environment. Open 'Menu' from the menu bar and select 'Load Demos'. A list of demo programs will open in a pop-up window. Select the program called 'Drive example' which will load in the programming environment.
3. Adjust your device's volume to maximum or 100%. Plug the EdComm programming cable into the audio jack of your device.

NOTE: many devices have built-in safety settings that reduce the volume when an audio device is connected to the headphone jack. Always double-check the volume settings after plugging in the EdComm cable to your device.

4. Turn your Edison robot on. Connect the EdComm cable to the bottom of the robot, near the power switch. Press the round (record) button one time.
5. In the EdBlocks app, press the 'Program Edison' button. Follow the instructions on the pop-up and then press the 'Program Edison' button on the pop-up to download the program into Edison.

NOTE: if the 'There seems to be a network issue accessing the compiler' warning message pops up at this point, see 'Troubleshooting 1: Check the connectivity status' section below.

6. While the program is downloading, you will hear a whirring sound, a bit like a dial-up modem. When the download is done, you will hear one of two sounds: the 'success' sound (the same chirping beep Edison makes when you first turn the robot on) or the 'fail' sound (a descending beeping sound)².

SUCCESS: If the robot makes the 'success' sound, unplug it from the EdComm cable, then press the triangle (play) button on Edison one time to run the program. If the program runs successfully in the Edison robot, your programming device is ready to use! There's nothing further you need to do to set-up your device.

FAIL: If the robot fails to download the program, or the program does not play in the robot, work through the 'Troubleshooting' section that follows.

¹ EdBlocks is compatible with Chrome, Safari, Microsoft Edge and Firefox. To ensure optimal performance, however, it is strongly recommended that you use EdBlocks with Chrome.

² You can hear recordings of both the success and fail sounds at <https://meet Edison.com/edison-robot-support/trouble-shooting/#success-fail-sounds>

Troubleshooting

Depending on the type of programming devices you are using and your network, there are a few things you will need to do to troubleshoot your devices and get them working with the EdBlocks app.

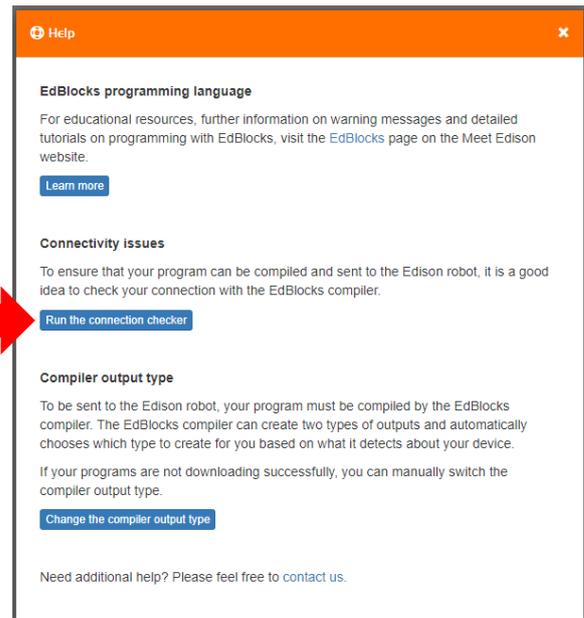
Troubleshooting 1: Check the connectivity status

If you see the *'There seems to be a network issue accessing the compiler'* warning message after pressing the 'Program Edison' button in the EdBlocks app or if the program failed to download successfully, you will need to check the app's connectivity status.

To work, the EdBlocks app needs to access the compiler (which is what converts the EdBlocks programs into a format that can be sent to the Edison robot). Inside the EdBlocks app at www.edblocksapp.com, open 'Menu' in the upper left-hand corner and select 'Help'. This will open a pop-up which includes the option to 'Run the connection checker'. Click this button to check your connection.

If the connection test result shows 'NO SERVER FOUND' then you may be behind a firewall, common at schools, which is blocking access to the compiler. You will need the network administrator to unblock ports 80, 8080, 443 and 8443 and white list these addresses:

- <https://local.edblocks.com>
- <https://api.edisonrobotics.net>
- 52.8.213.196
- 13.210.175.93
- 52.79.71.19



SUCCESS: Once the connection checker shows you are connected, try downloading and running a test program again. If the program downloads and runs successfully in the Edison robot, your programming device is ready to use! There's nothing further you need to do to set-up your device.

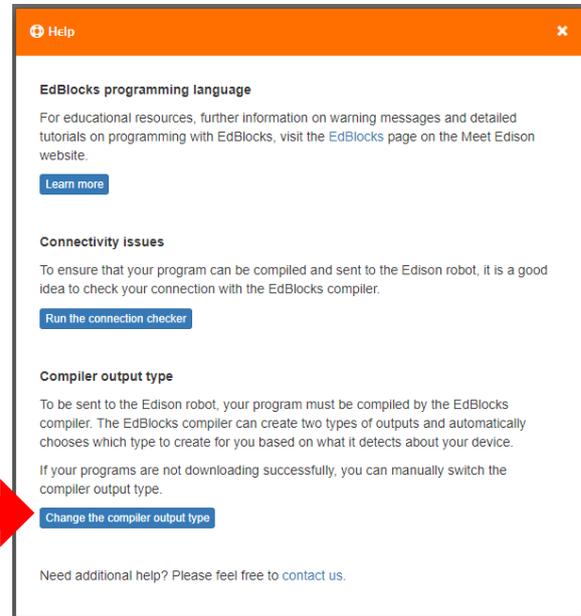
FAIL: If the connection checker shows you are connected, but you are still not able to program Edison, move on to 'Troubleshooting 2: Switch the compiler output type'.

Troubleshooting 2: Switch the compiler output type

To be sent to the Edison robot, your program must be compiled by the EdBlocks compiler. The EdBlocks compiler can create two types of outputs and automatically chooses which type to create for you based on the type of device it detects you are using (such as an Apple tablet or a Windows laptop).

If your programs are not downloading successfully, you can manually switch the compiler output type. Inside the EdBlocks app at www.edblocksapp.com, open 'Menu' in the upper left-hand corner and select 'Help'. This will open a pop-up which includes the option to 'Change the compiler output type'. Click this button to check what device and settings are being detected.

If the device being detected is not accurate, or if your programs are not downloading successfully, you can manually switch the compiler output type. Use the following information to select the output best suited to your device:



Long pulse compiler output

This output type works well on devices with low output volume, including some tablets. If you are using a Mac computer, an iPad tablet or a Windows or Android tablet, the long pulse compiler output should work best for your device.

Short pulse compiler output

This output type works well on devices with sound enhancement software, including most Windows desktop and laptop computers. If you are using a Windows desktop or laptop computer, the short pulse compiler output should work best for your device.

SUCCESS: Once you have changed the compiler output, try downloading and running a test program again. If the program downloads and runs successfully in the Edison robot, your programming device is ready to use! There's nothing further you need to do to set-up your device.

FAIL: If you are still not able to program Edison after changing the compiler output type, check the device-specific troubleshooting advice that follows.

Windows computers – troubleshooting

If you are running a laptop or desktop with a Windows operating system and are still unable to program Edison after running the set-up steps above, try these additional troubleshooting steps.

Disable sound enhancements

If you are using desktops or laptops running Windows operating systems and both the short pulse (recommended) and long pulse compiler output types are failing, you will need to disable sound enhancements.

Please go to <https://meetedison.com/edison-robot-support/trouble-shooting/#soundenhancements> to find step-by-step video guides showing you how to disable sound enhancements for standard Windows sound enhancements software as well as the most common third-party software programs.

Once you have sound enhancements disabled, use the long pulse compiler output option.

Check for a volume ‘hard lock’

Some devices, especially in Europe, have a hard lock on volume whenever an audio device is detected. This means that the device is ‘locked’ to only deliver a maximum volume of approximately 75% of the device max volume when an audio device is detected. To correct this, go into the device’s settings and disable the hard lock to enable the device to emit true full volume, even with an audio device plugged in.

Mac computers – troubleshooting

If you are running a Mac laptop or desktop and are still unable to program Edison after running the set-up steps above, try this additional troubleshooting step.

Check the volume settings

Some Macs experience audio clipping errors when attempting to program Edison. If you experience these issues, please try dropping your volume from 100% to between 50% and 90% instead.

Tablets – troubleshooting

If you are running an Apple, Android or Windows tablet and are still unable to program Edison after running the set-up steps above, try this additional troubleshooting step.

Check the volume settings

Many devices have built-in safety settings that reduce the volume when an audio device is connected using the headphone jack. Please check that your volume is

turned all the way up to 100% after plugging in the EdComm programming cable to your device.

Some devices, especially in Europe, have a hard lock on volume whenever an audio device is detected. This means that the device is 'locked' to only deliver a maximum volume of approximately 75% of the device max volume when an audio device is detected. To correct this, go into the device's settings and disable the hard lock, to enable the device to emit true full volume, even with an audio device plugged in.

Please note: most mobile phones do not have the audio output to program Edison using EdBlocks. We do not recommend using mobile phones as programming devices with Edison.

Chromebooks – troubleshooting

If you are running a Chromebook and still unable to program Edison after running the set-up steps above, try this additional troubleshooting step.

Disable sound enhancements

Some Chromebooks, including some Dell Chromebooks, have low audio output but also have sound enhancements. If you are using a Chromebook and both the short pulse (recommended) and long pulse compiler output types are failing, you will need to disable sound enhancements.

Sound enhancements are common on Windows machines and we have step-by-step video guides at <https://meet Edison.com/edison-robot-support/trouble-shooting/#soundenhancements> showing you how to disable sound enhancements for standard Windows sound enhancements as well as the most common third-party software programs. Depending on the manufacturer, your Chromebook may have similar sound enhancement software.

Once you have sound enhancements disabled, use the long pulse compiler output option.

Still not working?

You can find additional troubleshooting guidance on our website at <https://meet Edison.com/edison-robot-support/trouble-shooting/> or you can contact us for support at <https://meet Edison.com/edison-robot-support/contact-us/>

Our team of friendly Technical Support Officers will do their best to help you out!

Activity 1 – Meet Edison

This initial activity is designed to allow students to become familiar with the Edison robot, including the various sensors and buttons on Edison.

How it works

The Edison robot is programmable, LEGO brick-compatible and has built-in sensors, lights and sounds. Edison is capable of several autonomous behaviours, which students will explore in future lessons. Developing an understanding of Edison's parts will help students use Edison in future, more challenging lessons.

Tips and tricks

- It may be helpful to have students keep this activity sheet for their review of Edison's components in future lessons.
- All of the EdBlocks lesson activities contain information the students will need for the activity. Developing a good habit of reading the activity sheet and following its directions will help students work through future lessons independently. Reading this lesson and finding Edison's parts together may be helpful to begin to create this habit.

Activity extensions

1. Explore the many types of robots that exist in the world, such as robotic arms in manufacturing or Roomba-style robotic vacuum cleaners. Talk about what robots do to help people.
2. Discuss robots in pop culture, such as Disney's WALL-E and Star Wars. Explore the differences and similarities between real-world robots and fictional robots.

Activity 2 – Clap controlled driving

In this activity, students program Edison using a barcode which activates a pre-set program enabling them to drive Edison by clapping their hands.

How it works

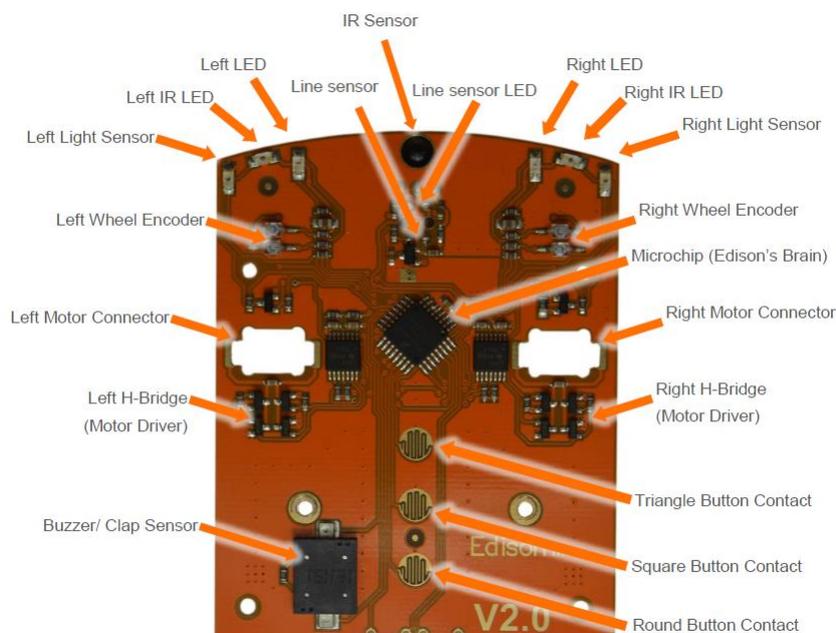
The clap controlled driving program uses the Edison robot's sound sensor to detect loud sounds like hands clapping. The pre-set program will turn Edison to the right when one clap is detected and drive Edison forward for a few seconds when two claps are detected.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's sound sensor before students begin the activity. You can refer to the images in activity one. For a more technically accurate guide, please download the [Edison motherboard layout guide](#).
- The robots may struggle to detect sounds when there is a high level of background noise. Having students tap a finger near the sound sensor on their Edison will create the same effect as clapping.

Activity extensions

1. Build a physical race course with barriers and turns to navigate. This can create an additional challenge for students, requiring them to use the variable inputs of single and double claps to control Edison.
2. Make a race track with a left-hand turn, encouraging students to use creative problem-solving skills.
3. Experiment to learn what sounds other than clapping and tapping Edison can detect. Discuss if any of these sounds are practical to use to control Edison.



Download the full-size motherboard layout guide at <https://meetedison.com/content/Edison-Motherboard-layout-V2.pdf>

Activity 3 – Avoid obstacles

In this activity, students program Edison using a barcode which activates a pre-set program enabling Edison to change directions while driving to avoid obstacles.

How it works

The avoid obstacles program uses the Edison robot's infrared (IR) light LEDs and IR sensor to detect objects directly in front of the robot. Once the pre-set program is activated, the Edison robot will drive forward, turning as needed to avoid obstacles it encounters.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's infrared light LEDs and sensor before students begin the activity.
- The obstacles need to be opaque but not too dark (e.g. not black) and at least as tall as Edison for the robot to detect them.

Activity extensions

1. Explore the concept of the electromagnetic spectrum, including visible and non-visible wave types. Learn more about infrared light, including its many common uses (e.g. in TV remote controls).
2. Discuss real-world applications of robotic obstacle avoidance, such as driverless cars' use of LIDAR laser remote sensing systems and camera sensor systems.

Activity 4 – Follow a torch

In this activity, students program Edison using a barcode which activates a pre-set program enabling Edison to drive by following the light of a torch.

How it works

The follow a torch program uses the Edison robot's light sensors to detect differences in the levels of light between the robot's two sensors. Once the pre-set program is activated, the Edison robot will drive towards the higher light level.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's two light sensors.
- Edison's light detecting programs rely on reading the difference between the light input to the two sensors. If students are struggling with a light detecting program, remind them to point the light at only one of the two sensors, not straight at the front of Edison.

Activity extensions

1. Experiment with the levels of light difference needed between Edison's two sensors to make the program work.
2. Explore the concept of animal behaviours in robotics, including the topic of 'phototropism' (the orientation of a plant or other organism in response to light). The 'follow torch' activity is an example of the Edison robot mimicking positive phototropism (moving towards the light), similar to moths flying towards a bright light.
3. Introduce the concept of intelligence, including artificial intelligence. Discuss what this program demonstrates about the intelligence of both insects and robots.
4. Pose the question 'why is an insect which is attracted to light alive, but a robot which is attracted to light is not alive?'

Activity 5 – Line tracking

In this activity, students program Edison using a barcode which activates a pre-set program enabling Edison to drive by following a dark coloured line.

How it works

The line tracking program uses the Edison robot's reflected light sensor to detect differences between dark and light surfaces beneath the robot. Once the pre-set program is activated, the Edison robot will drive until it finds a dark coloured line, then follow that line.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's reflected light sensor, located on the bottom of the robot.
- Make sure students start by placing Edison next to the black line, not on top of it.
- Edison's line tracking programs rely on reading the difference between the amounts of light reflected off of the surface the robot is above. Dark coloured surfaces reflect less light because they absorb more light than light coloured surfaces. If students are struggling with a line tracking program, check that the line they are using is a very dark colour, such as black, and approximately 1.5cm (0.6 inches) wide. Also, make sure the background is white or another very light (i.e. highly reflective) colour.

Activity extensions

1. Download the free EdMat from the www.meetedison.com website for an additional track for Edison to follow.
2. Explore the concept of how different colours have different light absorption and light reflection capabilities. Discuss real-world applications of using different coloured materials to either reflect light or absorb it, such as using dark cloth in a UV-absorbing shade sail.
3. Create different tracks with different coloured lines and different thicknesses of lines. Experiment to see which tracks the Edison robots can and cannot follow.

Activity 6 – Bounce in borders

In this activity, students program Edison using a barcode which activates a pre-set program preventing Edison from crossing a dark coloured line.

How it works

The bounce in borders program uses the Edison robot's reflected light sensor to detect differences between dark and light surfaces beneath the robot. Once the pre-set program is activated, the Edison robot will drive until it encounters a dark coloured line. It will then turn around and drive a different direction without crossing that line.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's reflected light sensor, located on the bottom of the robot.
- Edison's line tracking programs rely on reading the difference between the amounts of light reflected off of the surface the robot is above. Dark coloured surfaces reflect less light because they absorb more light than light coloured surfaces. If students are struggling with a line tracking program, check that the line they are using is a very dark colour, such as black, and approximately 1.5cm (0.6 inches) wide. Also, make sure the background is white or another very light (i.e. highly reflective) colour.

Activity extensions

1. Download the free EdMat from the www.meetedison.com website for an additional track for Edison to bounce inside.
2. Create different size tracks and test how many Edisons can run the program inside each at the same time without pushing any other robot out. Work as a class to make hypotheses about how many Edisons each track will be able to take, test and then present the data.

Activity 7 – Sumo wrestle

In this activity, students program Edison using a barcode which activates a pre-set program enabling multiple Edison robots to engage in a sumo-style battle.

How it works

The sumo wrestle program uses two of Edison's other programs – bounce in borders and obstacle detection. For this program to work, you need to place at least two Edison robots need to on a white-coloured surface with a black-coloured outline. The obstacle detection part of the program helps an Edison robot find the other robots while the line detection program helps Edison detect the dark-coloured borderline.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's reflected light sensor and Edison's infrared light LEDs and sensor.
- You will need at least two Edison robots for this activity. Both need to scan the sumo wrestle program.
- Make sure the sumo ring is large enough for both robots to drive around inside, however, if the ring is too large it will take longer for the robots to find each other
- Edison's line tracking programs rely on reading the difference between the amounts of light reflected off of the surface the robot is above. Dark coloured surfaces reflect less light because they absorb more light than light coloured surfaces. If students are struggling with a line tracking program, check that the line they are using is a very dark colour, such as black, and approximately 1.5cm (0.6 inches) wide. Also, make sure the background is white or another very light (i.e. highly reflective) colour.

Activity extensions

1. You can use the EdMat as the sumo ring. You can download the free EdMat from www.meetedison.com.
2. Explore the sport of sumo wrestling, including its cultural significance in Japan.

Activity 8 – Welcome to EdBlocks

This activity introduces students to EdBlocks, learn how Edison reads EdBlocks, and practice using EdBlocks offline.

How it works

EdBlocks is a robot programming language that has been created using Scratch 3.0 as a base. The horizontal icon-based language is comprised of different types of blocks which each contain different programming instruction for the Edison robot. EdBlocks are read left to right, one block at a time.

Tips and tricks

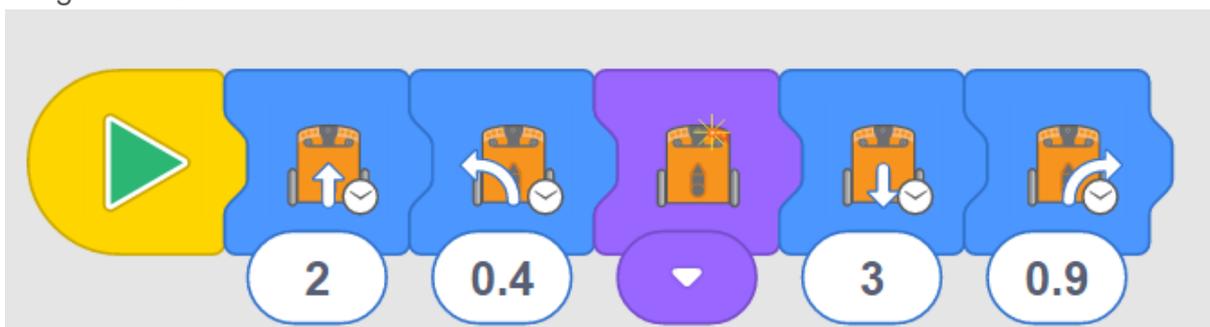
- There are different types of EdBlocks which will be introduced over time. Having students first interact with EdBlocks offline is a good way to teach the basic concepts of EdBlocks and limit confusion around more advanced block types.
- Most of the blocks in the activity have a clock symbol. Blocks with this symbol have a number field below the block which allows users to specify an amount of time for that action to occur. Understanding that this is a variable field inside EdBlocks is an important concept which students are exposed to during this activity.
- It may be helpful to introduce how EdBlocks works as a full class activity. A slide with the critical teaching information from this activity is included as an appendix at the end of this document for your use.

Activity extensions

1. Explore the idea that time is a variable in performing an action. Discuss how giving an amount of time for the robot to do an action is important.
 - *What would happen if we just told the robot to drive forward? How would it know how far to drive?*
2. Talk with students about the importance of giving exact directions in the right order. Practice the concept through activities such as having the class direct you step by step on how to make a sandwich.
 - *Example: <https://www.youtube.com/watch?v=RjHzD2sfWcQ>*

Answer key

Program order:



Activity 9 – Let's download a program

In this activity, students are introduced to EdBlocks inside the EdBlocks app (available online at www.edblocksapp.com) and learn how to download a program from EdBlocks to Edison.

How it works

The primary objective of this activity is to learn how to download a program. The program that this activity uses is a pre-set 'follow a torch' program which is functionally the same as the one students encountered previously using barcodes in Activity 4 'Follow a torch.'

Tips and tricks

- Practice steps 1,2,6,7, and 8 with students as these are the steps they will always need to download a program to Edison.
- Some devices, especially tablets, automatically lower the volume when they detect that an audio device, such as headphones, has been connected. Make sure the volume on the computer or tablet is still turned all the way up after the EdComm cable is plugged in.
- While a program is downloading to Edison, Edison makes a whirring sound. Once the program downloads successfully, Edison will make a chirping beep. Make sure students do not unplug the EdComm cable until after they hear the success sound beep.
- If the program fails while downloading, Edison will make a 'fail beep'. You can hear what the success and fail noises sound like at <https://meet Edison.com/edison-robot-support/trouble-shooting>
- When many students are downloading programs at the same time, you may experience slower internet speeds, causing the program to take longer to create the 'program Edison' pop-up box and for the program to download to Edison. Remind students to listen for the success sound before unplugging the EdComm cable to ensure they wait until the program fully downloads.
- Blocks must be attached to the start block to be downloaded to Edison. Floating blocks not attached to a start block will not download to Edison.
- You can remove blocks you don't want by dragging them into the trash bin in the lower right corner.

Activity extensions

1. Discuss how students were able to get Edison to run the 'follow a torch' program first using barcodes and now using EdBlocks. Explore the idea that this means that the same data can be represented in different ways.

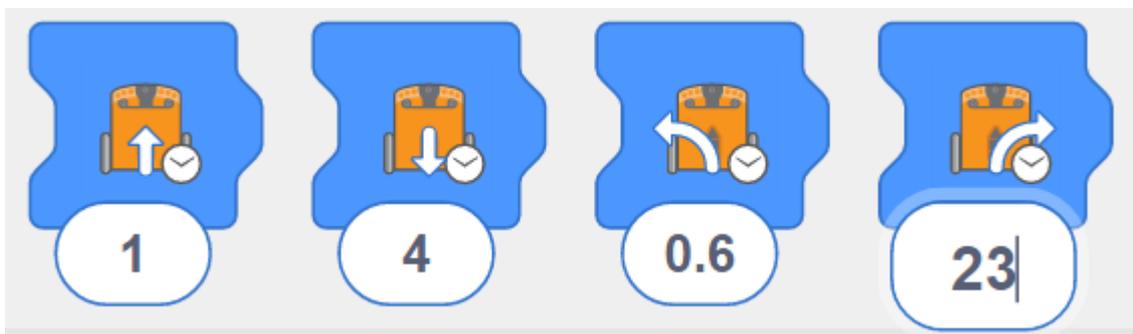
Activity 10 – Let’s drive

In this activity, students are asked to select and use EdBlocks inside the EdBlocks app (available online at www.edblocksapp.com) to complete the task of programming Edison to drive forward a set distance.

How it works

This activity introduces the important concept that everything you program Edison to do requires both an action and a duration.

In EdBlocks, some individual blocks contain both an action and a duration. For many of these blocks, the duration is time (in seconds) that the user can input. These blocks have a little clock symbol and a user input time box:



Other blocks only contain an action. The block ‘turn right LED on’ is an example of an action-only block. The duration of this block’s action is controlled by what happens in the program after the block, for example, if another block tells the program ‘turn right LED off’:



In this activity, to get Edison to drive forward a set distance, students need to select the action of ‘drive forward’ with the time controlled duration, then experiment with the amount of time they input to get Edison to drive the correct distance.

Tips and tricks

- Practice how to download a program with the students. Remind students to listen for the success sound before unplugging the EdComm cable to ensure the program fully downloads.
- There are several different ‘drive forward’ blocks in EdBlocks. For this activity, make sure students are selecting the time-controlled drive icon:



- Remind students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.

Activity extensions

1. Practice decimals, especially the tens and hundreds places.
2. Create math word problems based on this activity:
 - *If Edison travels the track in 2.3 seconds and the track is 23 centimetres long, how fast is Edison moving?*

Answer key

Please note: different Edison robots will travel at slightly different speeds, which may cause students to get slightly different results. Answers assume the activity sheet is printed on A4 paper.

Approximately 1 second.

Activity 11 – Let's turn

In this activity, students explore new blocks inside the EdBlocks app and are given the task of programming Edison to make two different time controlled turns.

How it works

This activity reinforces the concept that everything you program Edison to do requires both an action and a duration. It further encourages students to experiment with the user-input time blocks as a means of programming Edison to carry out tasks.

Tips and tricks

- Remind students how to download a program and to remember to listen for the success sound before unplugging the EdComm cable.
- There are several different 'turn' blocks in EdBlocks. For this activity, make sure students are selecting the turn blocks which are time-controlled:



- Remind students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.

Activity extensions

1. Review decimals, especially the tens and hundreds places.
2. Explore how geometry is displayed in this activity, especially how angles act as measures of turn.

Answer key

Please note: different Edison robots will travel at slightly different speeds, which may cause students to get slightly different results. Answers assume the activity sheet is printed on A4 paper.

1. (90-degree turn) Approximately 0.35 seconds.
2. (180-degree turn) Approximately 0.7 seconds.

Activity 12 – Let's try a maze

In this activity, students write a multi-block program using different time controlled drive blocks.

How it works

This activity helps students connect what they are doing in EdBlocks as true programming. Students will need to select the correct icons and durations, test the program, then adjust the program to correct for issues.

Tips and tricks

- Remind students how to download a program and to remember to listen for the success sound before unplugging the EdComm cable.
- There are several different 'drive' blocks in EdBlocks. For this activity, make sure students are selecting the time-controlled blocks.
- Remind students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.
- If your students are struggling to attach the EdComm cable to Edison, you may want to make some EdCoasters. You can get the free 3D printer file on our website at <https://meet Edison.com/content/EdCoaster.zip>

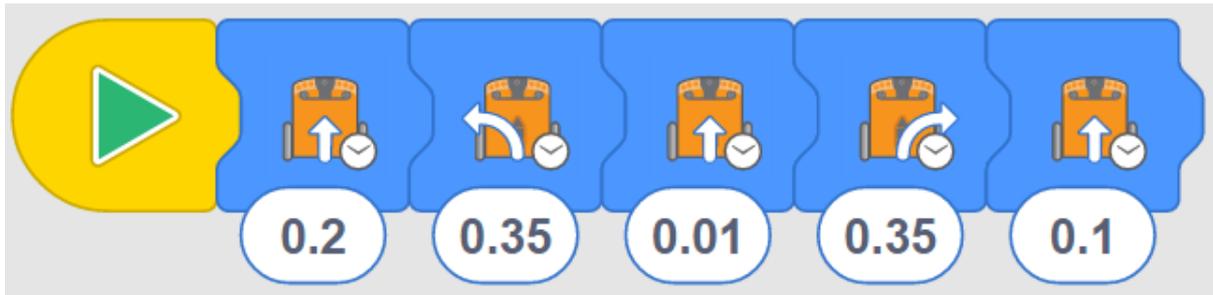


Activity extensions

1. Have students create and then exchange mazes with each other, then write and test the programs to solve the mazes.
2. Explore programming as a job, learning what programmers do and in what industries programmers work.

Answer key

Please note: different Edison robots will travel at slightly different speeds, which may cause students to get slightly different results. Answers assume the activity sheet is printed on A4 paper.



This program finishes with the front of Edison just touching the finish line.

Activity 13 – Let’s follow a line

In this activity, students continue to explore new blocks inside the EdBlocks app, including ‘follow a line’ and speed control.

How it works

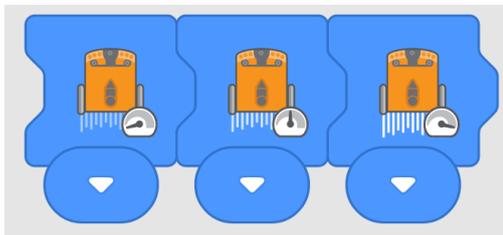
This activity reintroduces Edison’s ability to follow a line, using EdBlocks to create the program. Edison’s drive motors can be set to run at different speeds. This allows the motors to turn faster or slower than normal, enabling different behaviours in the robot.

Tips and tricks

- There are three different ‘follow a line’ blocks in EdBlocks. For this activity, make sure students are selecting the time-controlled ‘follow a line’ block:



- Remind students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.
- There are three speed blocks in EdBlocks, slow, normal and fast. The ‘normal’ block sets the drive motors to Edison’s approximate default drive speed.



- Because Edison reads EdBlocks left to right, a speed block must be placed to the left of the drive blocks. A speed block will affect all drive blocks to the right of that speed block in the program until the speed is reset by another speed block or the program ends.
- Running the drive motors at a faster speed for prolonged periods can run down Edison’s batteries more rapidly.

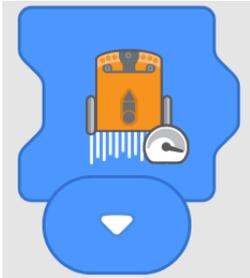
Activity extensions

1. Find the other two ‘follow a line’ blocks in EdBlocks. Discuss what each block does and why each would be useful.
2. The block ‘follow a line forever’ has a different shape than the other two ‘follow a line’ blocks. Ask students why they think this is the case.
3. The block ‘follow a line forever’ uses the infinity symbol in its image. Explore the idea of infinity.

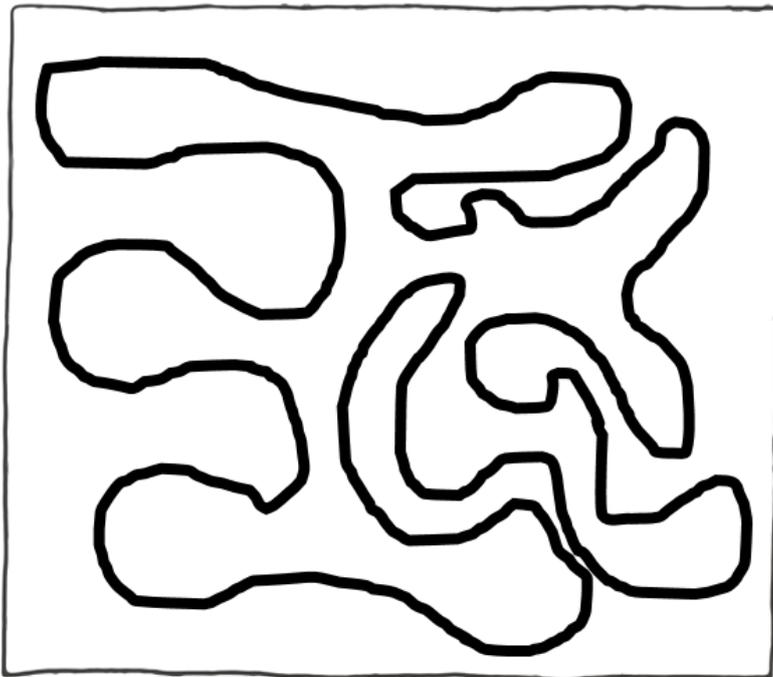
4. Have students make new programs with the other 'follow a line' blocks, then try their tracks again. Compare the different results.

Answer key

Please note: answers 2, 3 and 4 are example answers. Student results will depend on their experiences.



- 1.
2. Example time: 6.5
3. Example track:



4. Example write up: *The track lines were too close in some places. This made Edison go onto the wrong part of the track in these places. The track could be made better by having more space in between the track lines. Edison didn't finish the whole track. The program could have been improved by having a longer time.*

Activity 14 – Let's stop at a black line

This activity introduces students to the 'wait until' group of blocks and experiment with Edison's tracking sensor.

How it works

This activity uses Edison's built-in tracking sensor to allow students to program Edison with conditions. It also introduces the 'wait until' category of blocks, which can be used to make more complex programs where external inputs can influence Edison's program.

Tips and tricks

- This activity requires students to use a conditional drive block: the 'drive forward while on a white surface' block. Remind students to use this block, not a time-controlled drive block.



- Be sure to print the activity sheets are in full colour. Having low toner in your printer can make the coloured bars print in lighter shades, which may affect how Edison reads the bars.

Activity extensions

1. Explore the science behind Edison's line tracking sensor. Discuss how the blue and green coloured lines reflect only blue or green light, respectively. The red line only reflects red light. Because Edison's LED is red, all the red light shone onto this red line is reflected back to Edison. The blue and green lines reflect back no red light, so Edison detects these as non-reflective, same as the black line.
2. Hypothesise what other coloured surfaces will and will not reflect back the red LED light. Test and record the results. Present the result data as different types of charts and graphs.

Answer key

Please note: answer 4 is an example answer. Student results will depend on their experiences.

1. No
2. Yes
3. Yes
4. *I think black is the best colour to use to get Edison to stop because it absorbs all colour light, so Edison will always stop.*

Activity 15 – Let's make noise

In this activity, students explore more of the 'wait until' group of blocks and create programs which rely on user input.

How it works

This activity uses 'wait until' clap blocks to allow students to create a program using Edison's built-in sound sensor. Students will create a program that requires user input, in the form of hand claps, to progress.

Tips and tricks

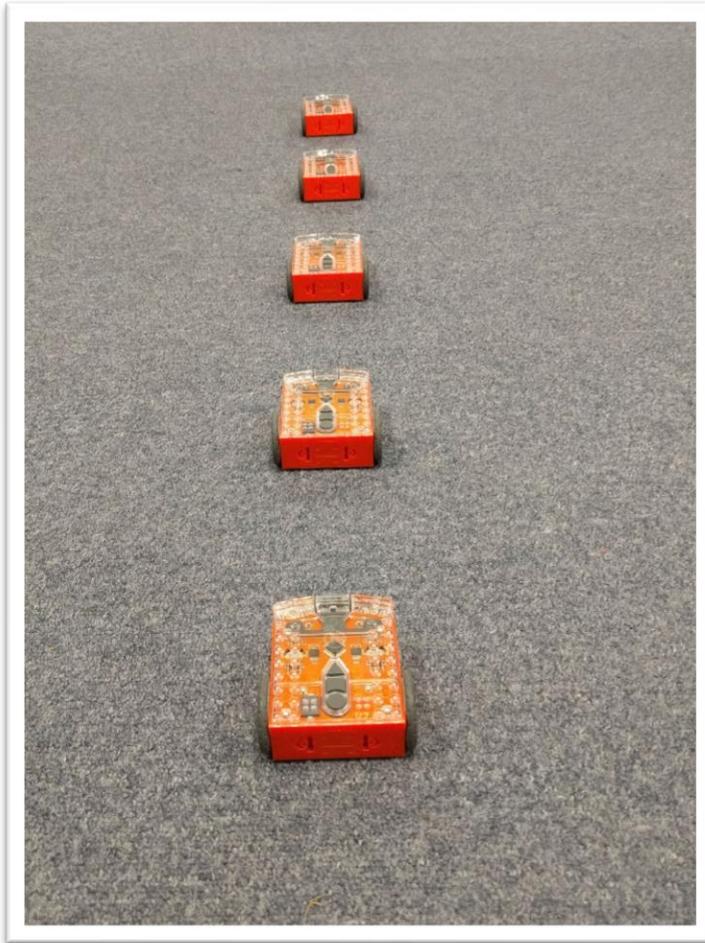
- It may help to review Edison's components, emphasising the location of Edison's sound sensor before students begin the activity.
- The robots may struggle to detect sounds when there is a high level of background noise. Having students tap a finger near the sound sensor on their Edison will create the same effect as clapping.
- Remind students that Edison reads EdBlocks one block at a time. Therefore, when a block tells Edison to 'wait until' a condition, Edison will wait until that condition is met before moving to the next block in the program.

Activity extensions

1. Introduce the idea of 'IF' statements. Pose the question 'if no clap is detected, what will Edison do?' Discuss how 'IF' statements could help in designing programs.
2. Play dominos with Edison. Program multiple Edison robots with a program that has them drive forward once a clap is detected.



1- Dominos example program



2- Dominos set-up example

Queue up the Edison robots one behind the next with no more space between them than the Edison will cover during its 'drive forward' block. Tap on the first Edison and watch as it causes a domino effect when it bumps into the Edison in front of it. See example program and set-up images for examples.

Answer key

Please note: this program is an example answer. Student results will depend on their experiences.



Activity 16 – Let's use the lights

In this activity, students work with two new sets of blocks inside the EdBlocks app – lights and loops – and explore the time-controlled 'wait until' block.

How it works

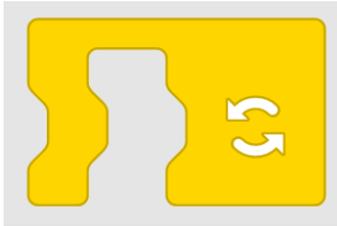
This activity introduces students to the loops group of blocks in EdBlocks. Loops allow users to create iteration (or repetition) in their programs. In EdBlocks, any blocks inside a loop will repeat in left-to-right order for as long as the loop conditions are met.

This activity also introduces the time-controlled 'wait-until' block. This block allows the user to program Edison to wait until the specified amount of time has elapsed before continuing to the next block in the program.

In addition, this activity introduces Edison's LED lights. Students are introduced to the lights and learn how to control turning the lights on and off.

Tips and tricks

- There are several different 'loop' blocks in EdBlocks. For this activity, make sure students are selecting the infinite loop:



- You can put many EdBlocks into a single loop. The loop blocks will stretch to fit over several blocks already linked together when you drag the loop over an existing string of blocks. Alternatively, you can put down the loop block first and then drop blocks inside the loop block.
- The 'wait until' time block allows for user input to control the amount of time for which Edison will wait. Remind students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.
- Unlike other blocks in EdBlocks, the LED lights operate in the background of the program. Once Edison's LED lights are turned on, they will stay on until they are turned off by the program, or the program ends.
- Edison's LEDs can be turned on and off individually or both together.

Activity extensions

1. Explore the other loop blocks in EdBlocks. Experiment making programs using the other loop blocks, testing how each allows external factors to influence the program.

- EdBlocks allows loops to be stacked inside of each other. Try using stacked loops and the LED blocks to have Edison send out messages using Morse code. For example, if having both LEDs turn on represents a dash, and only one LED turn on represents a dot, then you can make a program that sends out an 'SOS' message:

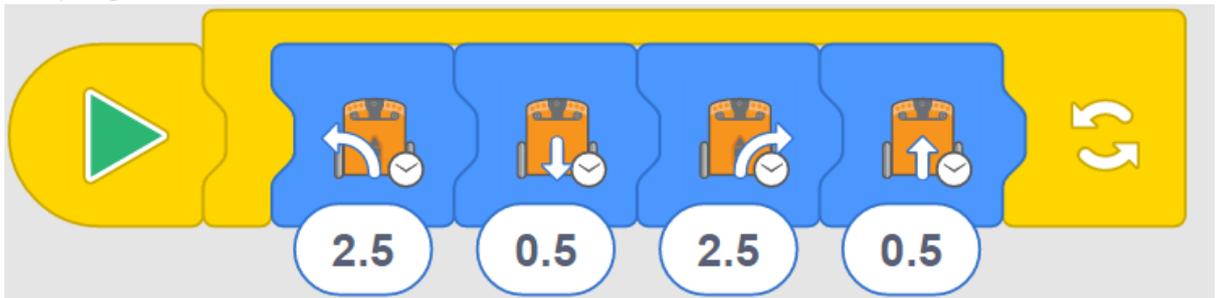


Learn more about Morse code as a class. Allow students to try sending out their own Morse code messages using lights and loops.

Answer key

Please note: answers 3 and 4 are example answers. Student results will depend on their experiences.

- The loop block allows a program to have blocks which repeat.
- If there was no loop block in the lights program, the program would turn the lights on, wait for 1 second, then turn the lights off, wait for one second, then the program would end.



-
-
-
- My program makes Edison spin left for 2.5 seconds, then drive backwards for .5 seconds, spin right for 2.5 seconds, then drive forwards for .5 seconds. When it loops for a while, it looks like Edison is doing a waddling dance.*

Activity 17 – Let's detect obstacles

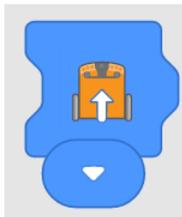
In this activity, students learn more about how Edison's infrared light sensors work and program Edison to detect obstacles.

How it works

In this activity, students create a program which has Edison avoid driving into obstacles by using the robot's infrared (IR) light LEDs and IR sensor to detect objects directly in front of the robot. This activity further explores how the infrared light sensors work in Edison and gets students to test what obstacles can and cannot be detected.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's infrared light LEDs and sensor before students begin the activity.
- The obstacles need to be opaque but not too dark (e.g. not black) and at least as tall as Edison for the robot to detect them.
- This program will work best using the 'drive forward' block in EdBlocks. This block sets Edison's motors to 'forward' until a condition or another drive block tell the motors to do something else. Ensure students are using the correct block:



Activity extensions

1. Have students present their findings from the activity. Practice good presentation skills including planning out their presentation, having a visual and projecting their voice.
2. Create a master list of class results, having each student add their results to the main list. Share the class's results with other classes or schools using online programs such as a Google Classroom or a forum. Look at the similarities and differences between what different obstacles people tested. What does this tell us about what other schools have in their classrooms? What questions can we think of looking at other results? Practice engaging in good online etiquette and safe online sharing behaviours.

Answer key

Please note: this table is an example answer. Student results will depend on their experiences.

Object	Colour and Shape	Did Edison stop? Why or why not?
<i>Water bottle</i>	<i>Clear, tall cylinder</i>	<i>No. The bottle is tall but does not reflect the infrared light.</i>
<i>Edison robot</i>	<i>Orange, short box</i>	<i>Yes, but only after the first Edison bumped into it. I think it is not quite tall enough.</i>
<i>Stapler</i>	<i>Dark yellow, long, medium-tall, sort of oval</i>	<i>Yes. The stapler was taller than Edison and must reflect enough light.</i>
<i>Textbook</i>	<i>Black with white stripes, large box-shaped</i>	<i>No. The book was taller than Edison, but the black colour absorbs the light.</i>

Activity 18 – Let’s detect and avoid

In this activity, students explore more ways to create programs in EdBlocks which use Edison’s infrared light sensors.

How it works

This activity reinforces students’ understanding of Edison’s obstacle detection abilities. In this activity, students create a program to have Edison avoid an obstacle once the robot detects the obstacle. The activity also challenges students to begin to use programming for creative problem-solving.

Tips and tricks

- It may help to review Edison’s components, emphasising the location of Edison’s infrared light LEDs and sensor before students begin the activity.
- The obstacles need to be opaque but not too dark (e.g. not black) and at least as tall as Edison for the robot to detect them.
- Encourage students to build on their prior learning by looking up which obstacles Edison was able to detect in activity 17.

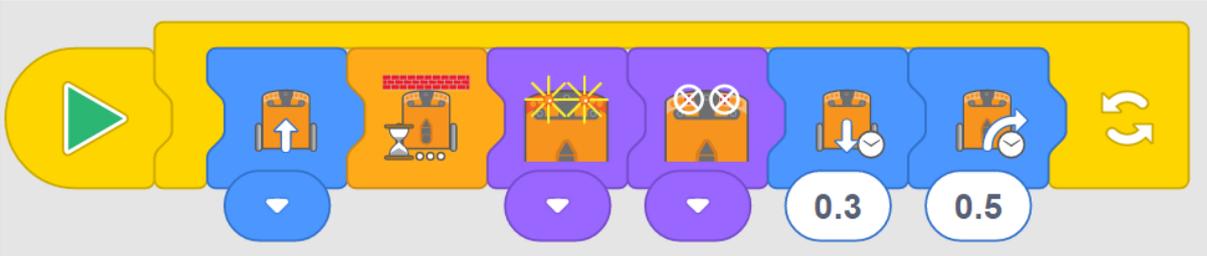
Activity extensions

1. Discover more ways infrared light is used in the real world. Research different devices that use infrared light, how each uses IR and what benefit the devices have for people or the community.
2. Imagine different ways students might use Edison’s IR sensors to create solutions for problems at school. For example, to act as hall monitor alarm systems if an obstacle (e.g. a person) is detected when there shouldn’t be anyone.

Answer key

Please note: different Edison robots will travel at slightly different speeds, which may cause students to get slightly different results for answers 1 and 2. The explanations in answers 1 and 2 and all of answer 3 are example answers. Student results will depend on their experiences.

1. 0.3 seconds. This time was enough for Edison to back up away from the object and have enough space to turn.
2. 0.5 seconds. This was enough time for Edison to turn far enough away from the object to not run into it again.

3. 

The image shows a sequence of EdBlocks programming blocks. From left to right: a yellow 'Start' block with a green play button; a blue 'Move Forward' block with an upward arrow; an orange 'Detect Obstacle' block with a brick wall icon; a purple 'Turn Right' block with a right-turn arrow; a purple 'Turn Left' block with a left-turn arrow; a blue 'Wait' block with a downward arrow and a clock icon, labeled '0.3'; another blue 'Wait' block with a downward arrow and a clock icon, labeled '0.5'; and finally a yellow 'End' block with a circular arrow icon.

Activity 19 – Let's stay in the borders

This activity has students use EdBlocks to create a new program utilising the Edison robot's reflected light sensor and challenges students to begin to consider how the concepts they are learning apply to real-world technology.

How it works

This activity uses the reflected light sensor to detect differences between dark and light surfaces beneath the robot. Students program Edison to stay within the borders of a dark line – same as the program they ran using a barcode in activity 6.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's reflected light sensor, located on the bottom of the robot.
- Remind students that Edison's line tracking programs rely on reading the difference between the amounts of light reflected off of the surface the robot is above. Check that the line they are using is a very dark colour, such as black, and approximately 1.5cm (0.6 inches) wide. Also, make sure the background is white or another very light (i.e. highly reflective) colour.
- Remind students that there are three speed blocks in EdBlocks, slow, normal and fast. The 'normal' block sets the drive motors to Edison's approximate default drive speed.
- Because Edison reads EdBlocks left to right, a speed block must be placed to the left of the drive blocks. A speed block will affect all drive blocks to the right of that speed block in the program until the speed is reset by another speed block or the program ends.
- Running the drive motors at a faster speed for prolonged periods can run down Edison's batteries more rapidly.

Activity extensions

1. Discuss how we can think of programming in terms of problems and solutions. Phrase the activity in this terminology. For example, 'the problem is we need to keep Edison from crossing a black line. How can we solve this problem?'
2. Examine how the 'problem' of keeping Edison from crossing a black line can be 'solved' in different ways using EdBlocks. Try creating different programs that all solve the same problem. See how many different solutions students can create. Discuss as a class the idea that all of the different solutions are valid so long as they all solve the problem.

Answer key

Please note: different Edison robots will travel at slightly different speeds, which may cause students to get slightly different results for answers 1 and 2. Answer 3 and 4 are example answers. Student results will depend on their experiences.

1. 0.3 seconds
2. 0.4 seconds
3. *The normal speed block worked best. The slow speed block was boring to watch. The fast speed block would sometimes run over the edge of the black line before backing up.*
4. *I think it would be good for cars to have a program like this in real life. I think this because then you could have intersections change to be a colour the car would always see when the light is red. That way, cars would have to stop and couldn't go through intersections at a red light so that you would have fewer accidents.*

Activity 20 – Let’s make music

In this activity, students explore a new set of blocks inside the EdBlocks app, the music blocks, and create programs using Edison’s sound capabilities.

How it works

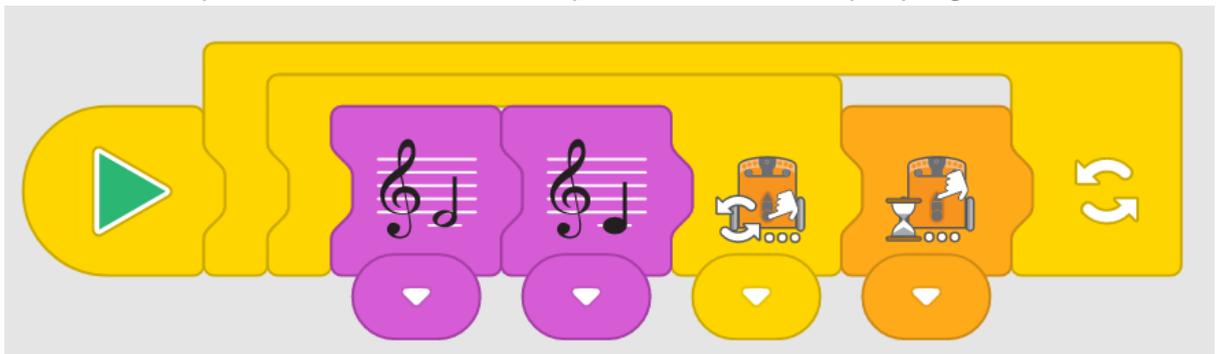
This activity uses Edison’s buzzer. This buzzer enables the robot to play a range of different musical notes and can also be used as a sound sensor to detect loud sounds, such as claps.

Tips and tricks

- V1 Edison robots require a [firmware update](#) available on the [meetedison.com](#) website to be able to use the musical blocks in EdBlocks. V2.0 Edison robots do not require this update. You can tell if your Edison is a V2.0 by looking for the little ‘V2.0’ in the lower right-hand corner on the top of your Edison.
- Because Edison reads EdBlocks left to right, a metronome (tempo) block must be placed to the left of the music blocks. A metronome (tempo) block will affect all music blocks to the right of that tempo block in the program until the tempo is reset by another tempo block or the program ends.
- It may help to have some simple songs ready to use for students who aren’t as familiar with music.

Activity extensions

1. Play musical chairs with Edison. Have students create programs that will play music in a loop until the round button is pressed. See example program:



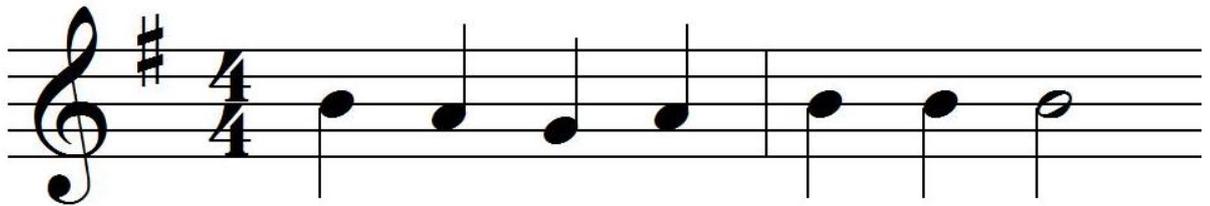
Once a student presses the round button, the program will finish playing the notes, then wait until the triangle button is pressed again. Have students take turns being in charge of running their musical chairs program for a group.

2. Have students work together to have several Edison robots play a tune in a round. Use time-controlled ‘wait until’ blocks to program different Edisons to begin playing a tune (like Row, Row Your Boat) at different times. Experiment with time amounts in the ‘wait until’ blocks to get the Edison robots to play in time with one another.

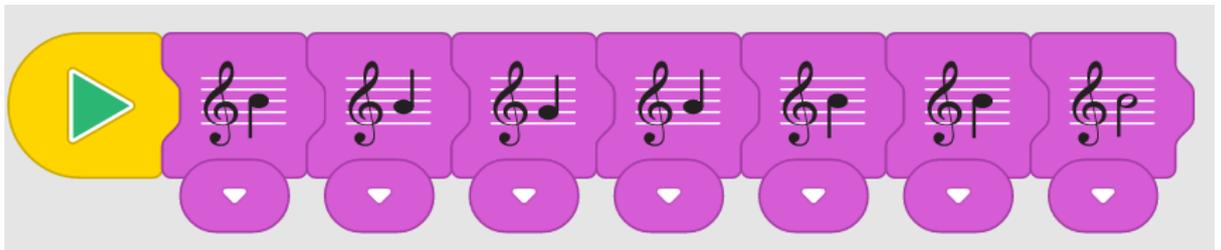
Answer key

Please note: answers 1 and 2 are example answers. Student results will depend on their experiences. Guidance for what to look for with question 1 has been provided.

1. While the program can be whatever the student likes, you may want to see their program as well as their answer to confirm they have translated the notes from one format (the online program) to another (sheet music) correctly. Example:



Ma - ry had a lit - tle lamb



2. I used the 'fast' metronome block. It made Edison play the song faster, which sounded funny.
3. Edison drove backwards and played the notes at the same time. Edison played through the notes two times while driving backwards.

Activity 21 – Let's perform

In this activity, students begin to combine multiple types of blocks from EdBlocks to produce a creative program that employs Edison's light sensor.

How it works

This activity builds on the ability of the Edison robot's light sensors being able to detect differences in the levels of light between the robot's two sensors. By blocking out one sensor completely, students can trigger Edison to react when exposed to any light.

This programming task encourages the students to begin to combine different types of blocks together as well as tying their programs in EdBlocks back to the Edison robot's various sensors.

Tips and tricks

- It may help to review Edison's components, emphasising the location of Edison's two light sensors.
- Edison's light detecting programs rely on reading the difference between the light input to the two sensors. In this activity, one of the two light sensors must be covered completely.
- Ensure students are selecting the light detection block in their program, compared to their Edison. If the right light detector is covered, their program needs to have the 'detect left light' block and vice versa.
- With only one light sensor exposed, any light will trigger the program. Including the time-controlled 'wait until' block enables Edison to be placed somewhere dark. If a student is struggling with having the music start too soon, increase the time in the 'wait until' block.

Activity extensions

1. Use this activity as a starting point to explore performance and theatre. Research different types of theatrical performances that include music, such as musical theatre and opera. Compare and contrast the similarities of this activity to the different types of theatre. Present the results in formats such as Venn diagrams.
2. Review the different costumes different students made. Explore the different materials students used and how they attached costumes to Edison. Discuss which costumes worked best in the experiment and what made them the most suitably designed solutions.

Answer key

Please note: Student results will depend on their experiences. The activity includes instructions to leave the buttons exposed and only cover one side – these criteria can be used to ensure students followed directions for marking if desired.

Activity 22 – Let’s avoid the light

In this activity, students continue to explore how to use EdBlocks to create programs which harness the power of Edison’s various sensors, connecting programming and robotics together.

How it works

This activity uses the Edison robot’s light sensors ability to detect differences in the levels of light between the robot’s two sensors to affect how Edison behaves. In this activity, the program tells Edison to avoid light from both sensors, which enables user input to control the robot’s behaviour.

Tips and tricks

- It may help to review Edison’s components, emphasising the location of Edison’s two light sensors.
- Remind students that Edison’s light detecting programs rely on reading the difference between the light input to the two sensors. If students are struggling with this program, remind them to point the light at only one of the two sensors, not straight at the front of Edison.

Activity extensions

1. Revisit the concept of animal behaviours in robotics, including the topic of ‘phototropism’ (the orientation of a plant or other organism in response to light). This activity is an example of the Edison robot mimicking negative phototropism (moving away from the light). Explore biological examples of negative phototropism, such as plant roots growing away from the light.
2. Encourage students to begin to think creatively about programming. Challenge them to write a program with EdBlocks which would help them win the cockroach game. For example, a student might write a program that combines the ‘stay in borders’ program from activity 19 and the ‘avoid light’ program from this activity to help their robot stay in the circle.

Answer key

Please note: answers are example answers. Student results will depend on their experiences. Guidance for what to look for with question 3 has been provided.

1. *Charlotte*
2. *Charlotte’s program used the fast speed block.*
3. *Some insects, like cockroaches, run away from the light. In the game, all the Edison robots avoided the light. They behaved like cockroaches. That’s why I think it is called ‘the cockroach game.’*

Activity 23 – Let’s have a dance party

In this capstone activity, students learn to use start events and messaging, then create their own program using a range of block types.

How it works

This activity introduces two of the most complicated block types in EdBlocks: messaging and start events. Because these types of blocks can be a challenge, this activity uses you as the teacher to participate, helping to simplify the student’s initial exposure.

You will need to program one Edison to be the leader, sending out the pink message. The students will need their programs to react upon receiving that message, triggering their Edison robot’s to ‘dance.’

Additional information: how messaging and start events work

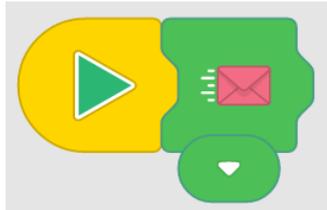
Messages: Edison robots can send and receive infrared messages to one another. In EdBlocks, the coloured envelopes are the blocks used for this messaging function. To use the messages, you need at least two Edison robots, and you will need to write a separate program for each robot. One robot will need to send one of the coloured messages out as a part of its program. The second robot will need to have a program that contains either a ‘wait until’ or ‘start event’ message block of the same colour. When the first robot’s program progress to the ‘send message’ block, it will send out an infrared message. This will trigger the second robot to begin or continue its program from the message block in the second robot’s program.

Start events: The ‘start events’ blocks allow a user to make a sub-program begin from a specific event while the main program is running. These blocks emulate the computer programming concept of ‘interrupts.’ Start events allow users to vertically layer programs, with secondary programs interrupting the main program when the start event is triggered.

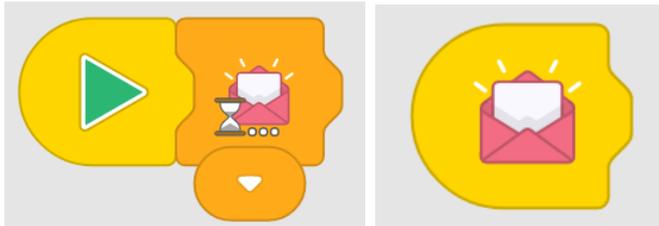
To use a start event, simply add it to the working space in the EdBlocks app and drag and drop program blocks after it, as you would the normal start block. When you download your program to Edison, both the main program and the subprogram beginning with the new start event will download. When the play (triangle) button on Edison is pushed, Edison will run the main program. If the condition of the start event occurs, Edison will run the start event sub-program to completion, then return to the main program and continue the main program. This will happen whenever the conditions of the start event are met. If no blocks are attached to the main program, Edison will wait until the condition of the start event is met, then run the start event sub-program.

Tips and tricks

- This activity requires that one Edison robot be the ‘leader’ and send out the pink message:



- Make sure that students' 'dance routine' programs start only once the pink message is received. To do this, students need to start their dance either after a 'wait until' pink message block or after the pink message start event:



- Edison's messaging uses infrared, which has a limited range, similar to a TV remote control. If any robots are not detecting the pink message, they may be too far away from the lead robot.
- In EdBlocks, messaging is restricted to the six pre-set channels depicted by the six coloured envelopes. Robots must use the same coloured envelope to send and receive. For example, if robot A sends the blue message but robot B has 'wait until green message' in its program, the message sent by robot A will not trigger robot B to continue its program.
- Explain to students what the message block is doing. In this activity, when the leader robot sends the pink message, that is the equivalent of that robot saying 'start!' to all the other robots. The other robots then know to move to the next block in their programs. Explain that the lead robot is not telling the other robots *how* to dance, only that they should move from their 'wait until' block to their dance blocks.
- Remind students that because Edison reads EdBlocks left to right, Edison will wait until a message block is received before continuing on to any blocks placed to the right of the 'wait until' message block.

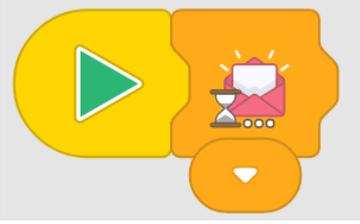
Activity extensions

1. Explore the other start blocks. Try creating programs which use the different start blocks.
2. Try creating a program using the main start block and a subprogram using a start block. Experiment with how Edison switches from the main program to the subprogram when the start condition is triggered.

Answer key

Please note: answer 2 is an example answer. Student results will depend on their experiences. A note on question 2: having students capture their programs with a

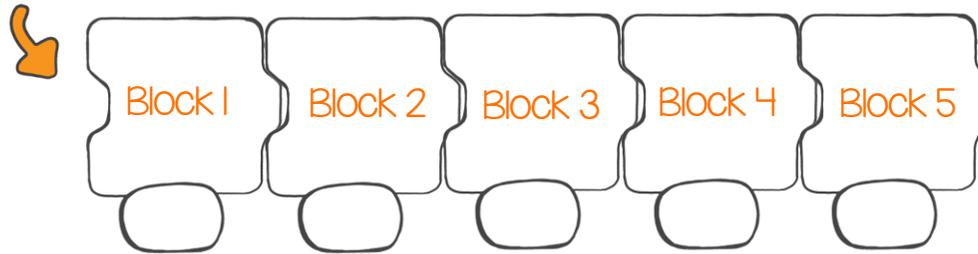
screenshot tool is a good chance to practice additional computer programs and peripheral devices, including the printer.

1.  and 

2.  

How do EdBlocks work?

Now it is time to use EdBlocks.



What is EdBlocks?

EdBlocks is a robot programming language we can use to program Edison.

You can program Edison to do different things using EdBlocks.



How do EdBlocks work?

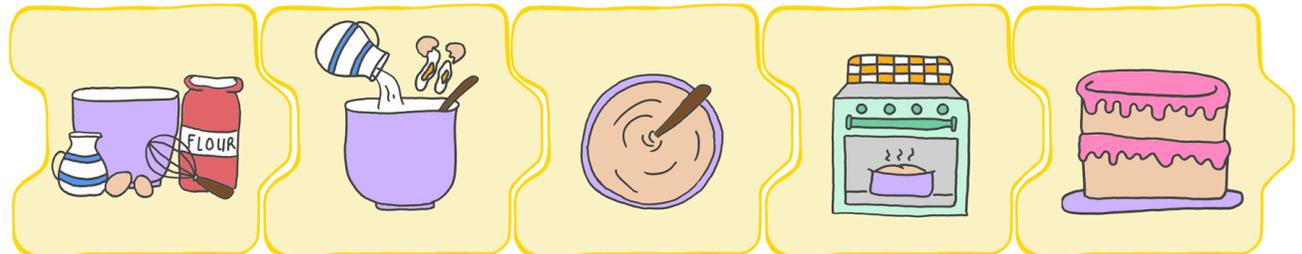
Imagine you are baking a cake. How would you know what to do?

When you read a cookbook, you follow each step one by one.

Edison also follows steps one by one, using EdBlocks.

Edison reads EdBlocks from left to right, starting at block one. Edison then moves along the EdBlocks one block at a time.

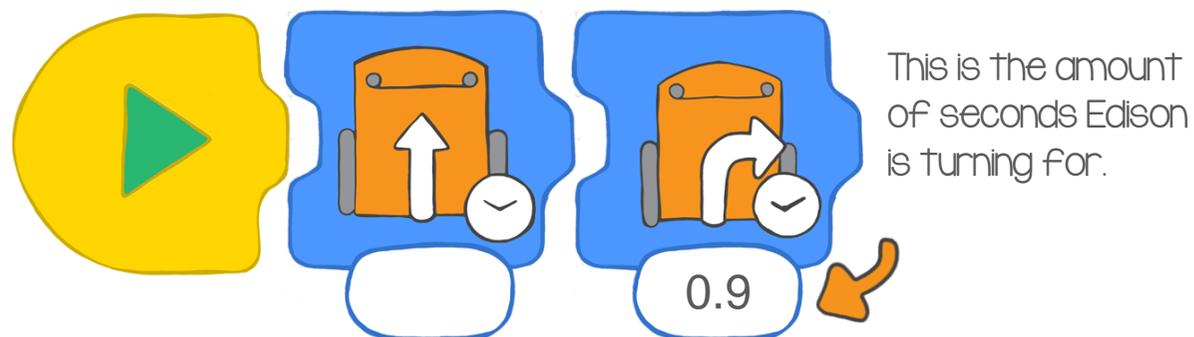
So, Edison can follow the same instructions, just in a different way.



Let's practice using EdBlocks.

We need to get Edison to do several things, one by one. Get Edison to do the following things in this order:

1. Drive forward for 2 seconds
2. Turn left for 0.4 seconds
3. Turn on the right LED light
4. Drive backwards for 3 seconds
5. Turn right for 0.9 seconds



These are the blocks we need to write the program, but they are in the wrong order.



Let's put the blocks in the correct order.

