Code a guard dog

Teacher's notes

Doc 3.2.4.5.3.2.1 Rev 1.1

Contents

Go ahead – show off!

We love seeing how classrooms use Edison! If you and your students want to share your Hour of Code Edison EdVenture, be sure to tag us into the fun!

@meetedison twitter.com/meetedison

@meet_edison instagram.com/meet_edison

@meetedison facebook.com/meetedison

About this lesson and guide

This guide offers teachers and instructors overview information, facilitation recommendations and other supporting information for the *Code a guard dog* lesson available at https://meetedison.com/robotics-lesson-plans/guard-dog

Do you need to read this whole guide to run the lesson? **Absolutely not!**

As long as the robots and programming devices are set up[1], you can start learning along with your students! The student sheets for this lesson have been designed to allow students to work through the stages of the lesson independently, learning key computer science concepts and practicing skills along the way. This guide simply offers further information for teachers and instructors to help make using this lesson easy and fun.

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

How it works

Information about the technology and computer science concepts being highlighted in that section.

Delivery recommendations

Suggestions for how you can cover the lesson section if you want to run the lesson in a more facilitator-led capacity.

Tips and tricks

Helpful hints and ways to overcome common issues students may encounter.

Creative Commons licence attribution details

Developed and written by: Kat Kennewell
Illustrations by: Emma Dewar

[1] The *Getting started with Edison and EdBlocks* set available at https://meetedison.com/robotics-lesson-plans/guard-dog has step-by-step help for setting up your robots and programming devices. If you are new to Edison or EdBlocks, it is recommended you start with that guide.
[2] https://meetedison.com/robot-programming-software/edblocks/#edblocks-resources
[3] http://creativecommons.org/licenses/by-sa/4.0/

Lesson overview

Introduce the key computational concepts of loops and conditionals using Edison robots and the EdBlocks programming language[4]. This lesson utilises the Edison robot's infrared (IR) light sensors through a set of progressive programming tasks. Students apply sequential programming and decomposition to the challenges while exploring how robots can sense and react to the world.

Grade levels	Difficulty	Prerequisites	Duration
Year 2 – Year 6	Beginner	None[5]	55 minutes

Computer science and computational thinking topics	☐ Sequential programming ☐ Loops ☐ Conditionals (sensor-driven events) ☐ Decomposition and problem solving
Tie-ins to other subjects	☐ Science: light spectrum and IR light

Supplies you need

- Full set of Edison robots[6] and EdComm programming cables
- Full set of prepared programming devices (computers or tablets)
- 4x AAA batteries per robot
- Print-outs or digital copies of the student sheets
- Objects for the robots to detect

Some great advice from the Hour of Code team

It's okay not to know! Respond to student questions and struggles with phrases like:

- "I don't know. Let's figure this out together."
- "Technology doesn't always work out the way we want."
- "Learning to program is like learning a new language; you won't be fluent right away."

And don't forget to have fun! (^_^)

[4] https://meetedison.com/robot-programming-software/edblocks/

[5] You and your students should be familiar with Edison and EdBlocks already, however. The *Getting started with Edison and EdBlocks* set available at https://meetedison.com/robotics-lesson-plans/guard-dog has a step-by-step activity for introducing Edison and EdBlocks. If this is your first-time using Edison or EdBlocks, start with that guide and activity.

[6] You may want to run an obstacle detection calibration in the Edison robots before this activity. The special barcode you need and full instructions for performing an obstacle detection calibration can be found at https://meetedison.com/content/Edison-robot-barcodes.pdf

Section 1 – What is infrared light?

Start off the lesson with a quick exploration of invisible light.

How it works

The Edison robot's obstacle detection capabilities use the robot's infrared (IR) light LEDs and IR sensor to detect objects directly in front of the robot. Spending a few minutes exploring the basic idea of the electromagnetic spectrum, including visible and non-visible wave types, will help students understand the fundamental technology behind Edison's IR sensor system.

Delivery recommendations

- Recommended time: 5 minutes

Tips and tricks

- A great way to see the Edison robots in action using the IR sensor system straight away is to run the *Avoid obstacles* pre-set program using the special Edison robot barcode[7]. Using barcodes with Edison robots is one of the most unique and fun ways to kick-start your Edison adventure no matter your coding experience!

Section 2 – How does Edison use infrared light?

Look at the science and technology of Edison's infrared (IR) light sensors, then put them to work in a basic driving program. Understanding how this tech works (and how the output students see in their robots connects to the code they see on their programming screens) will help them to use the same technology in their own coding creations.

How it works

This section first explores how the infrared light sensors work in Edison. Students then recreate 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.

Delivery recommendations

- Recommended time: 10 minutes
- If students are new to programming, you may want to point out the sequential order of the code blocks, noting why it's important that the blocks are arranged in this order for the program to work.

[7] https://meetedison.com/barcodes/

- If students are new to EdBlocks, you may want to run this section as a group, discussing what each of the blocks in the program is telling the robot to do.

Tips and tricks

- 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. An option that works well is to have students make a 'wall' by holding their hand in the robot's path.
- If an Edison robot is not detecting obstacles correctly, you may need to run an obstacle calibration on that robot. The special barcode you need and full instructions for performing an obstacle detection calibration can be found at https://meetedison.com/content/Edison-robot-barcodes.pdf
- Have students make the program exactly as it appears in the example. Switching blocks (such as swapping the blue 'drive forward' block for a time-input 'drive forward for [time]' block) will alter the behaviour of the program. The 'drive forward' block used in this program 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:

Section 3 – What are loops?

Explore the coding concept of loops. Students expand on the program they made in the previous section, using a 'forever' loop (infinite loop) and additional drive blocks to get the robot to detect, then react to obstacles in its path.

How it works

This section explains what a loop is in coding and introduces the different types of loops in the 'control' block category 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 sequential order for as long as the loop conditions are met.

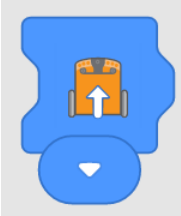
Delivery recommendations

- Recommended time: 10 minutes
- Pointing out the similarities of the program in this and the previous section can help students cement their understanding of what the code blocks are telling the robot to do. This is important for the next section, where students will need to expand on what they have learned to create their own programs from scratch.

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.
- There are several different types of drive blocks in EdBlocks. For this activity, students need to use both the ‘drive forward’ and the ‘drive [direction] for [time]’ blocks:



  Remind students to use the right type of ‘drive’ block in the right spot of the program.
- Tell 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.
- Remind students to use obstacles that are opaque but not too dark (e.g. not black) and at least as tall as Edison so that the robot can detect them.

Section 4 – How can Edison patrol, detect and alert?

Pull everything together and get creative by using the programming and robotics concepts from the lesson in your own coding creation!

How it works

This section builds on the Edison robot’s IR sensor’s ability to detect objects, allowing the programmer to set up a unique ‘reaction’ when an obstacle is detected. Students use decomposition to break the goal of getting Edison to ‘patrol, detect and alert’ into four parts, tackling each part one at a time.

This programming task encourages students to combine different types of blocks together into a logical sequence, using a conditional loop to move the robots from

‘patrolling’ to ‘alerting’. By using the IR sensors with an obstacle, students can see how their program in EdBlocks ties into the behaviour they see in their Edison robot in the form of inputs and outputs.

Delivery recommendations

- Recommended time: 30 minutes
- Resilience is key in programming! If students are getting stuck with this section, help them break down what they are trying to do into smaller chunks, then work out a solution for each part one at a time. Practicing decomposition in this way is a valuable computational thinking skill.
- If students are new to EdBlocks, be sure to share with them that any code blocks in the programming area that are NOT attached to the ‘start’ block will not download to the robot. This is really helpful in this section because students can test sections of their coding solution (for example, their alarm) by attaching just those bricks to the ‘start’ block in EdBlocks, then downloading and running the program in Edison. They don’t need to delete the other blocks. Instead, they can leave the rest of their program in the programming area unattached to the ‘start’ block. This way students don’t need to start over from scratch each time they want to check just one section of their code.

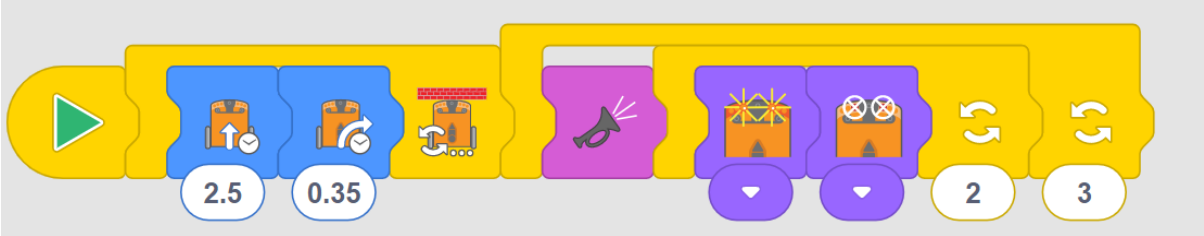
Tips and tricks

- You may want to encourage students to choose a simple ‘patrol path,’ such as moving in a square pattern around a desk or a chair.
- Students will be most successful if they select the time-controlled drive blocks for driving the patrol path:

- 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 several different ‘loop’ blocks in EdBlocks. For ‘*Part 2: Keep patrolling until an obstacle is detected*’, students will be most successful if they use the ‘loop until an obstacle is detected’ loop:

- Remind students that the loop blocks will stretch to fit over several blocks already linked together when you drag the loop over an existing string of blocks.

- The ‘alarm’ portion of the program is completely up to the students. You can let them be as creative with this as they like, or you might encourage them to create code that will get Edison to imitate real-life alarms.
- Remind students to use obstacles that are opaque but not too dark (e.g. not black) and at least as tall as Edison so that the robot can detect them.
- There is no ‘correct’ answer to this section, but here is what one sample solution looks like:



What should I do if a student finishes the lesson early?

- Encourage students to write another program for Edison in EdBlocks. What do the other blocks do? What can they get their Edison robot to do using code in EdBlocks?
- Or, ask students who finish early to help classmates who are having trouble with the activity.