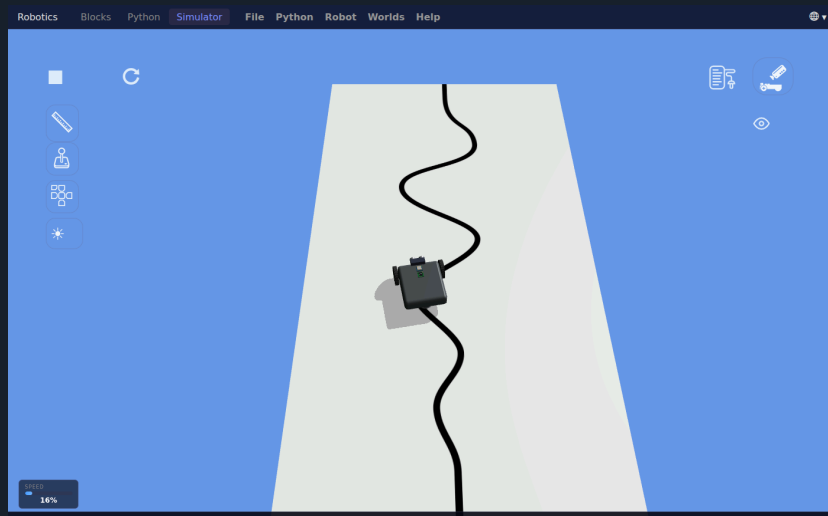


mytekDev

Robotics: Follow a Line with One Sensor

Grades 5-8 · 45 minutes

Computer science · Engineering



Slides, rubric and answer key

Print and go

No student accounts

No robot kit needed

What's included

- A one-page overview: objectives, essential question, vocabulary, standards and assessment
- A ten-minute walkthrough to rehearse before class, and background in plain words for the teacher
- A 45-minute lesson sequence: warm-up, direct instruction, guided practice, independent practice, closure
- Slides for the board: a PowerPoint file that also opens in Google Slides
- A student worksheet with a QR code
- A 3-question exit ticket, two to a page, with an answer key
- A rubric: 3 criteria, 4 levels, scored out of 12
- Differentiation for English learners, students with IEPs or 504 plans, and students who need more support or more challenge
- Cross-curricular extensions, and notes for teaching it in grades 3-4 and 9-12
- Standards: CSTA 2017 and NGSS engineering

Robotics: Follow a Line with One Sensor

Grades 5-8 · 45 minutes · Computer science, Engineering

Students run a simulated robot that follows a black line using one light sensor. They watch the sensor read about 16 over the line and about 100 over the white mat, and find the program's rule: below 58, turn one way; otherwise, turn the other way, checked over and over. They predict what a gentler turn will do, see the robot run off the mat at the first curve, and then test other turn values as a class. By the end, they can explain how one sensor and one cut-off number keep a robot on a line it cannot see ahead of.

At a glance

ESSENTIAL QUESTION

How can a robot with one sensor stay on a line it cannot see ahead of?

OBJECTIVES

Students will be able to:

- explain how a light sensor and a cut-off number let the robot tell the black line from the white mat
- read an if-else inside a loop and say what the robot does over dark and over light
- predict how a changed turn value will change the drive, then test it and record what happened
- use the class's test results to choose a turn value and explain why a turn that is too gentle cannot follow a curve

VOCABULARY

sensor, intensity, threshold, condition, loop, variable, test

STANDARDS

1B-AP-15, 2-AP-17, 3-5-ETS1-3

TIME AND MATERIALS

45 minutes, grades 5-8.

- One computer or Chromebook per student or pair
- A projector
- The worksheet and exit ticket, one copy each per student
- A class results table on the board (the worksheet has the same table)
- The rubric, one copy per student if you grade the project

PREP

Make copies of the worksheet and exit ticket. Draw the class results table on the board: turn values 20, 30, 50, 60, 70 and 80 down the side, and columns for Reached the end, Left the line and came back, and Ran off the mat. Before class, open mytekdev.com/try/robotics?lesson=follow-a-line-with-one-sensor&ref=tpt-robotics-1 once on a student computer and press Run, to check that your school's filter allows it and that the 3D view runs. No account is needed.

ASSESSMENT

- Formative: the questions under What to look for, asked while students work, and the exit ticket (answer key included).
- Summative: the worksheet's prediction table and the student's written choice of a turn value, scored with the rubric out of 12.

TECHNOLOGY AND PRIVACY

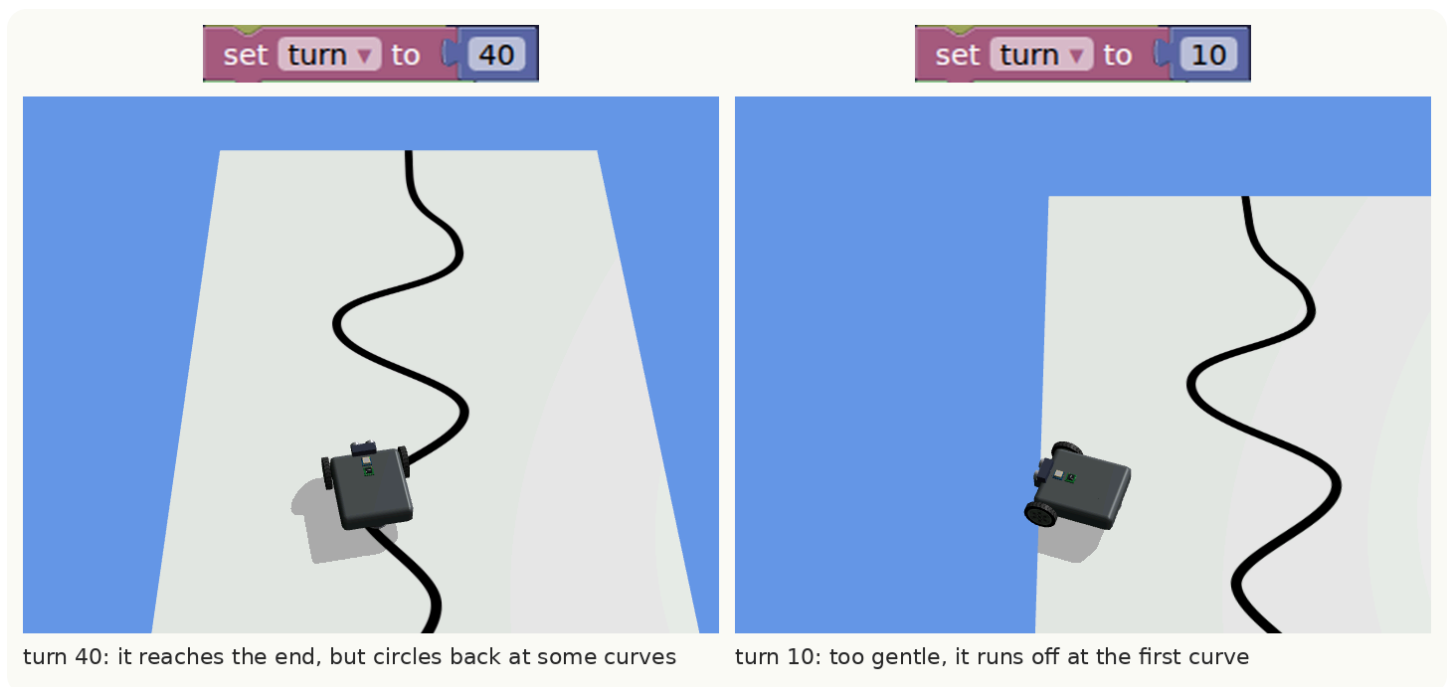
Runs in any browser, Chromebooks included, with nothing to install. The 3D view is drawn by the computer's graphics, and how often the robot checks its sensor depends on how fast the computer draws: a slower computer checks less often, so the same program can wobble more. That is part of the lesson, not a fault. Students need no account and no email: the link works signed out, and nothing they make is stored on our server unless they choose to save it to an account. If your school

blocks new sites, ask IT to allow
mytekdev.com.

Before you teach: do it once yourself

Run the robot and make the lesson's change yourself once before class, about ten minutes.

1. **Open the lesson.** Go to mytekdev.com/try/robotics?lesson=follow-a-line-with-one-sensor&ref=tpt-robotics-1. The app opens on the Simulator. Press Run, the play button at the top left. The robot follows the black line to the far end of the mat and stops.
2. **Watch the sensor.** Press reset, the circle arrow next to Run. Open the sensor readings with the sensors button at the top right (it looks like a page of lines, just left of the camera button), and press Run. Under `in1: Color` `Sensor`, `Intensity (%)` drops to the teens over the line and rises to about 100 over the mat.
3. **Read the program.** Open `Blocks`: set turn to 40, then repeat while gps y is less than 90, with an if-else inside it that compares color_sensor reflected light intensity to 58.
4. **Make it fail.** Change 40 to 10. Go back to `Simulator`, press reset, then Run. The robot runs off the mat at the first curve. Set it back to 40.
5. **Try one class value.** Try 80 the same way, so you have seen one of the values students will test.



Turn 40 reaches the end, circling back to the line at some curves. Turn 10 is too gentle and runs off the mat at the first one.

Run and reset are only on the Simulator tab, so after every change in Blocks, students have to switch back. Run starts from wherever the robot is now, so a robot that was not reset starts at the far end or off the mat. Typing in the Python tab brings up a warning that Python changes cannot be converted back into blocks; after that, Run uses the Python and ignores changes to the blocks.

Background in plain words

The robot has one light sensor under its front, pointing down. It shines on the mat and measures how much light comes back: black reflects little, so the reading is low (about 16), and white reflects a lot (about 100). The program picks a cut-off between them, 58. Below it counts as the line, and above it counts as the mat. That cut-off is called a threshold.

One sensor cannot tell which side of the line the robot is on, only whether it is over dark or light. So the program follows the edge of the line: over dark it steers one way, over light it steers the other way, and it checks again every few milliseconds. The robot never drives straight. It zigzags along the edge, correcting each time it crosses. The turn value is how hard each correction is. Too gentle, and the robot cannot keep up with a bend. Too hard, and the robot spends its time turning instead of driving: at 80 it takes about two minutes to reach the end, and at 100 the wheels turn opposite ways and it only spins in place.

How often the robot checks matters too. In this simulator, the robot reads its sensor once each time the screen is drawn, so a slower computer checks less often, and the robot drives further between corrections. On this line the difference is small: in our tests, each turn value in this plan gave the same result on fast and slow computers, and only the times changed. Real robots have the same problem, which is why engineers test a program on more than one machine before they trust a number.

STUDENTS ASK	A GOOD ANSWER
Why does it wiggle instead of driving straight?	It only knows dark or light. Each time it checks, it turns toward the other side, so it keeps crossing the edge of the line.
Why 58?	It sits between the line's reading, about 16, and the mat's, about 100. Readings in between happen when the sensor is half over the edge.
How does it know when to stop?	The loop repeats while the GPS y is less than 90. 90 is just short of the far end of the mat, so the loop ends there and the robot stops.
Why did my result not match my partner's?	First check that you both pressed reset before Run. Then check that neither of you typed in the Python tab, which makes Run ignore the blocks. In our tests, every value in the table gave the same result on fast and slow computers.

Lesson sequence

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
0 to 7	Warm-up Hook	Show the title, essential question and Today I can slides (about 1 minute). Then have students open the link or scan the QR code on the worksheet. The app opens on the Simulator with the robot at the bottom of a long white mat and a curvy black line. A short guide runs along the bottom: tell students to do its first step only, which is to press Run (the play button) and watch, then wait. The robot follows the line to the far end of the mat and stops. Ask: "This robot cannot see the line ahead of it. How does it stay on the line?"	Run the program, watch the robot follow the line, and answer the question.

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
7 to 17	Direct instruction (I do) Model it	On the projector, press reset (the circle arrow next to Run), open the sensor readings with the sensors button at the top right (it looks like a page of lines, just left of the camera button), and press Run. Point to Intensity (%) under in1: Color Sensor: it drops to the teens over the black line and rises to about 100 over the white mat. Show the What the sensor reads slide and the vocabulary. Then open Blocks and read the program from the top: set turn to 40; repeat while gps y is less than 90 (near the far end of the mat); inside it, if color_sensor reflected light intensity is less than 58, move steering with direction turn and speed 20, else the same block with minus turn; then sleep for 10 milliseconds. Say: the robot has one sensor, so it only ever knows dark or light. Over dark it turns one way and over light the other, so it zigzags along the edge of the line, checking many times a second. It's working when: The readings list shows every sensor on the robot. The one for this lesson is in1: Color Sensor, row Intensity (%).	Watch the reading change, fill in worksheet step 2, and say what the robot does over dark and over light.
17 to 25	Guided practice (We do) Predict, then test	Together: in Blocks, change set turn to 40 to 10. Before anyone presses Run, students write a prediction in the table. Then go back to the Simulator, press reset, then Run. The robot runs off the mat at the first curve: its swings are too gentle to follow the bend. Show the One number, two robots slide. Set turn back to 40. Before moving on, check that every table has a prediction and a result for this row. Watch for: Run and reset are on the Simulator tab only. Run starts from wherever the robot is now, so a robot that was not reset starts at the far end or off the mat, and the result makes no sense.	Write a prediction, reset and Run, and record what happened.

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
25 to 38	Independent practice (You do) Test it as a class	Give each pair two turn values from the class table (20, 30, 50, 60, 70, 80), so every value is tested by at least two pairs on different computers. For each value they predict, reset and Run, and record one of three results: reached the end, left the line and came back, or ran off the mat. Each run goes on until the robot stops at the far end or leaves the mat, which can take about two minutes. As pairs finish, they add tally marks to the board. Walk the room and ask the questions under What to look for. When the table is full, ask: which value would you choose, and what is your evidence? Watch for: In our tests, each value in the table gave the same result on fast and slow computers, though the times differed. Turn 40 reaches the end, but at some curves it circles off the line before finding it again, so it belongs under Left the line and came back. Turn 30 circles off the line and back for a minute or more before it runs off the mat, so it belongs under Ran off the mat; a pair that stops watching early records it as came back. Turn 80 stays on the line but crawls, and takes about two minutes to reach the end. If two pairs report different results for the same value, first check that both pressed reset before Run, then treat the split as data for the closing question, not as a mistake. Watch for: A student who edits the Python tab gets a warning, and from then on Run uses the Python and ignores the blocks. Keep changes in Blocks.	Test two turn values, writing each prediction before Run, and add the results to the class table. Then write which value they would choose and why.
38 to 45	Closure Read the data, and exit ticket	Show the last slide and ask: "Turn 10 ran off the mat and turn 40 reached the end. Why does a harder turn help on a curve?" (The robot cannot see the curve coming. It only finds out it has left the line when the sensor reads light, and each correction has to swing it back before it drives off. A gentle turn swings back too slowly to keep up with the bend. If any pairs reported different results for the same value, ask what they would check first: was the robot reset, and were the blocks changed?) Ask the essential question again, then hand out the exit ticket.	Answer out loud, then answer the exit ticket on their own.

What to look for

Ask these while students work. A student who has understood the lesson can answer them without help:

- The robot has one sensor. How does it know which way to turn?
- Why does it zigzag instead of driving straight along the line?
- Turn 10 missed the first curve. Why?
- Where does the number 58 come from?
- Two groups got different results for the same value. What would you check first?

The last one is the real test. A student who checks how a test was run (was the robot reset, were the blocks changed) before blaming the program is thinking like a robotics engineer.

Cross-curricular extensions

- Math: the line reads about 16 and the mat about 100. What number is exactly halfway? (58.) Why might a program use the halfway point as its cut-off? (A reading half over the edge lands near the middle.)
- Science: black things absorb more light than white things. Where else does a machine use reflected light? (Answers vary: barcode scanners, automatic doors, the sensor that turns on street lights at dusk.)
- Engineering: real line-following robots often use two sensors, one each side of the line. What could a robot learn from two sensors that it cannot learn from one? (Which side of the line it has drifted to.)
- Computer science: open the Python tab. The if block is one line that starts with if. Find it, and find the 58 in it.

Vocabulary

sensor	A part that measures something about the world. This robot's light sensor points down and measures how much light the mat reflects.
intensity	How much light comes back to the sensor, from 0 to 100. Black reflects little, white reflects a lot.
threshold	A cut-off number. Readings below it count as the line, and readings above it count as the mat.
condition	A question the program asks that is true or false, like: is the reading less than 58?
loop	Part of a program that repeats. This one repeats while the robot's GPS y is less than 90, which is near the far end of the mat.
variable	A named value the program can use in more than one place. Here, turn sets how hard the robot steers.
test	Running the program on purpose to find out what it does, and writing down the result.

Differentiation

ENGLISH LEARNERS

- Post the vocabulary with a picture for each word: the sensor under the robot, the black line, the white mat, the 58 in the program.
- Model each step on the projector before students try it, and point at the block as you name it.
- Let students record results with the three words from the table: end, came back, off.

STUDENTS WITH IEPs OR 504 PLANS

- Where the IEP allows a modified assignment, test one turn value instead of two.
- Pair students: one changes the number, the other presses reset and Run and reports the result.
- Raise the browser zoom so the numbers in the blocks are easier to read.
- Read the exit ticket aloud, or let students answer by pointing at the block on their screen.

STUDENTS WHO NEED MORE SUPPORT

- Give a sentence starter for the table: "I think turn ... will ... because ..."
- Write on the board: dark is below 58, light is above 58.
- Check in at minute 30: does the student press reset before every Run?

STUDENTS READY FOR MORE

- Change the cut-off from 58 to 90, predict what happens, and test it. Then try 30. Which readings count as the line now?
- Open the Worlds menu, choose Select World, pick Sharp Turns under Select Challenge, and press Ok. Before running, predict whether your chosen turn value will still reach the end, then test it.
- Open the Python tab and find the line that is the if block. Explain each part of it in your own words.

Adapting for other grades

Younger students: For grades 3 and 4, do the turn 10 test together on the projector, have each pair test one value, and score only the first two rubric rows (out of 8; 6 or more meets the objectives).

Older students: For grades 9 to 12, have students work in the Python tab, test the cut-off as well as the turn value, and write a short note on what would happen to the zigzag if the robot checked its sensor less often.

Student worksheet

Name: _____ Date: _____



Open the app: scan the QR code, or go to mytekdev.com/try/robotics?lesson=follow-a-line-with-one-sensor

Essential question: How can a robot with one sensor stay on a line it cannot see ahead of?

1. **Run it.** Press Run, the play button. Watch the robot follow the line.
2. **Read the sensor.** Open the sensor readings. Intensity over the black line: _____ over the white mat: _____
3. **Reset first.** Before every Run, press reset, the circle arrow, so the robot starts over.
4. **Predict, then test.** For each turn value, write your prediction first. Then reset, Run, and write what happened: reached the end, left the line and came back, or ran off the mat.

TURN VALUE	I PREDICT	WHAT HAPPENED
10		
My first value: ____		
My second value: ____		

1. Which turn value would you choose? What results from the class table support it?

2. Why does the robot zigzag instead of driving straight?

Exit ticket

Name: _____

1. The light sensor reads 20. Where is it most likely to be? (A) over the black line, (B) over the white mat, (C) you cannot tell.
2. The program says: if intensity is less than 58, steer one way, else steer the other way. Why does the robot zigzag instead of driving straight? (A) the line is crooked, (B) it only knows dark or light, so it keeps turning back across the edge, (C) the motors are uneven.
3. With turn set to 10, the robot ran off the mat at the first curve. What would you try next, and why?

Exit ticket

Name: _____

1. The light sensor reads 20. Where is it most likely to be? (A) over the black line, (B) over the white mat, (C) you cannot tell.
2. The program says: if intensity is less than 58, steer one way, else steer the other way. Why does the robot zigzag instead of driving straight? (A) the line is crooked, (B) it only knows dark or light, so it keeps turning back across the edge, (C) the motors are uneven.
3. With turn set to 10, the robot ran off the mat at the first curve. What would you try next, and why?

Answer key: exit ticket

Measured in the app: the color sensor's intensity reads about 16 over the black line and about 100 over the white mat. With turn 10 the robot runs off the mat at the first curve; with turn 40 it reaches the end of the line.

QUESTION	ANSWER	WHY
1	A	20 is close to the line's reading, about 16, and far below the cut-off of 58.
2	B	One sensor only tells dark from light. Each check turns the robot back across the edge, so it zigzags along it.
3	A bigger turn value, such as 40, because 10 turns too gently to follow the bend	Full credit for a larger value and a reason about how hard the robot turns.

Rubric

Score each row from 1 to 4.

Name: _____

CRITERIA	4 EXCEEDS	3 MEETS	2 APPROACHING	1 BEGINNING
Predicts, then tests	Every prediction was written before Run and has a result, and the student explains the one that surprised them most.	Every prediction was written before Run and has a result.	Results are written, but some predictions are missing or were written after Run.	Numbers were changed, but nothing is written down.
Explains the sensor and the rule	Explains that the reading is low over the line and high over the mat, that 58 splits them, and why the robot zigzags along the edge.	Explains that the robot turns one way below 58 and the other way above it.	Says the robot uses the sensor, but not how the reading decides the turn.	Cannot yet say what the sensor measures.
Chooses a value from evidence	Chooses a turn value using the class results, and explains why turns that are too gentle run off at a curve.	Chooses a turn value and points to test results that support it.	Chooses a turn value without pointing to any results.	Does not choose a value.

Total: ____ / 12. A score of 9 or more meets the objectives.

Standards

These CSTA 2017 standards describe testing a program and fixing it until it runs as intended. Students read a program with an if-else inside a loop, predict what a changed value will do, test it, and choose a value from the results. The NGSS standard describes fair tests that look for failure points.

FRAMEWORK	CODE	GRADES	STANDARD	HOW THE LESSON MEETS IT
CSTA 2017	1B-AP-15	3-5	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Students see turn 10 run the robot off the mat, explain why its turns are too gentle, and name a value to try instead.
CSTA 2017	2-AP-17	6-8	Systematically test and refine programs using a range of test cases.	Students test a range of turn values, predict each result before running it, and record what happened in a class table.
NGSS	3-5-ETS1-3	3-5	Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.	Students change only the turn value, run fair tests, and record where the robot fails (leaves the line or runs off the mat) to choose a better value.

Checked October 2026 against: CSTA 2017 progression chart, CSTA 2017 to 2026 crosswalk, NGSS 3-5 Engineering Design.

Optional: save student work in a free class

The whole lesson runs at the link on the worksheet, with no accounts and no sign-up. Work stays in the browser tab until it is closed.

To keep students' work, and see every student's program in one place, make a free class. It takes about ten minutes, once, and covers every lesson in this series: mytekdev.com/tools/class-setup

Students in a class open mytekdev.com/tools/robotics/lessons/follow-a-line-with-one-sensor/start instead of the worksheet link. Each one gets their own copy of the line follower, and the same link reopens it next time. A student who started at the worksheet link and then signs up keeps what they made.

The Robotics simulator is built on GEARS (GearsBot) by Cort Wee, open source under the GNU General Public License 3.0. Its blocks are Blockly, its 3D view is Babylon.js and its Python runs in Skulpt.

Terms of use

Thank you for downloading this lesson. The plan, the handouts and the art were made by mytekDev.

You may:

- print and copy it for your students
- change it for your own classes
- post it on your class website or learning management system
- share this file with other teachers, whole and unchanged, with this page included

You may not:

- sell it, or put it in a product you sell
- share a changed copy
- claim it as your own

The app this lesson uses is free and runs in the browser. Students need no account. It is at mytekdev.com.

Questions, or found a mistake? Ask on the Teachers Pay Teachers Q&A for this resource, or write to us at mytekdev.com/contact.

The Robotics simulator is built on GEARS (GearsBot) by Cort Wee, open source under the GNU General Public License 3.0. Its blocks are Blockly, its 3D view is Babylon.js and its Python runs in Skulpt.

© 2026 mytekDev. All rights reserved.