



EdCreate

Teaching guide

| | | | | | |
|--|--|--|---|--|---|
| <pre>1 #-----Setup----- 2 3 4 import Ed 5 6 Ed.EdisonVersion = Ed.V2 7 8 Ed.DistanceUnits = Ed.CM 9 Ed.Tempo = Ed.TEMPO_MEDIUM 10 11 #-----Your code below----- 12 13 drawDiagonal(15) 14 15 #draws a line forwards and left 16 def drawDiagonal(numCM): 17 #constrain input value 18 if numCM > 15: 19 numCM = 15 20</pre> | A photograph of an Edison robot built with orange and grey LEGO bricks. It has a vertical grey sensor or camera module mounted on top and a long, thin arm extending from the side. | | | | |
| A photograph of an Edison robot with a complex gear mechanism on top, featuring several black gears of different sizes connected by grey axles. | <table border="1"><tr><td data-bbox="813 1232 1109 1456">A photograph of an Edison robot modified with a grey tank-like track system.</td><td data-bbox="1109 1232 1422 1456">A photograph of an Edison robot with a black bucket or scoop attachment on the front.</td></tr><tr><td data-bbox="813 1456 1109 1695">A blue button with a white envelope icon and an upward-pointing arrow, with the number "0.5" in a white circle below it.</td><td data-bbox="1109 1456 1422 1695">A blue button with a white envelope icon and a downward-pointing arrow, with the number "0.5" in a white circle below it.</td></tr></table> | A photograph of an Edison robot modified with a grey tank-like track system. | A photograph of an Edison robot with a black bucket or scoop attachment on the front. | A blue button with a white envelope icon and an upward-pointing arrow, with the number "0.5" in a white circle below it. | A blue button with a white envelope icon and a downward-pointing arrow, with the number "0.5" in a white circle below it. |
| A photograph of an Edison robot modified with a grey tank-like track system. | A photograph of an Edison robot with a black bucket or scoop attachment on the front. | | | | |
| A blue button with a white envelope icon and an upward-pointing arrow, with the number "0.5" in a white circle below it. | A blue button with a white envelope icon and a downward-pointing arrow, with the number "0.5" in a white circle below it. | | | | |



The EdCreate Teaching guide by [Brenton O'Brien and Kat Kennewell](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Contents

| | |
|---|----|
| EdCreate and education..... | 4 |
| Using this guide | 4 |
| A note on the programming projects | 4 |
| EdBuild – EdTank | 5 |
| Tips and tricks for this build | 5 |
| Activity extensions | 6 |
| Programming project – firefighting water cannon | 6 |
| The scenario | 6 |
| Setting up the scenario | 6 |
| Programming task..... | 6 |
| Sample program solution (EdBlocks)..... | 6 |
| Things to look out for..... | 7 |
| EdBuild – EdDigger | 9 |
| Tips and tricks for this build | 9 |
| Activity extensions | 9 |
| Programming project – remote control events | 10 |
| Project objective..... | 10 |
| Setting up..... | 10 |
| Programming task..... | 13 |
| Sample program solution (EdBlocks)..... | 13 |
| Things to look out for..... | 14 |
| Programming project – semi-automated remote control | 15 |
| The scenario | 15 |
| Setting up the scenario | 15 |
| Programming task..... | 15 |
| Sample program solution (EdBlocks)..... | 15 |
| Things to look out for..... | 17 |
| EdBuild – EdRoboClaw | 18 |
| Tips and tricks for this build | 18 |
| Activity extensions | 19 |
| Programming project – hazardous material removal | 19 |
| The scenario | 19 |

| | |
|---|----|
| Setting up the scenario | 19 |
| Programming task | 20 |
| Sample program solution (EdBlocks) | 20 |
| Things to look out for..... | 21 |
| EdBuild – EdCrane..... | 22 |
| Tips and tricks for this build | 22 |
| Activity extensions | 23 |
| The EdCrane program in EdScratch | 23 |
| The EdCrane program in EdPy | 23 |
| Programming project – marine port crane | 24 |
| The scenario | 24 |
| Setting up the scenario | 24 |
| Programming task | 25 |
| Sample program solution (EdPy) | 25 |
| Things to look out for..... | 25 |
| EdBuild – EdPrinter | 26 |
| Tips and tricks for this build | 26 |
| Activity extensions | 27 |
| Programming project – make your own shape | 27 |
| The challenge..... | 27 |
| The open-ended programming task | 27 |
| Things to look out for..... | 28 |
| EdCreate EdChallenges..... | 29 |
| Example challenge | 29 |

EdCreate and education

The EdCreate building system is a construction system consisting of over 100 interlocking building blocks, pegs, gears and parts. Designed to work with the Edison robot, EdCreate enriches students' STEM education experience through interactive engineering and programming projects.

The EdCreate system can be used with Edison robots to complete five EdBuild projects:

- the EdTank,
- the EdDigger,
- the EdRoboClaw,
- the EdCrane, and
- the EdPrinter.

The EdBuild projects require progressive levels of coding capabilities, making them a perfect addition to the Edison programming and robotics curriculum. EdCreate can also be used in a wide range of open-ended engineering, design and programming challenges. Open-ended projects, including the supplementary EdCreate EdChallenges, promote creative problem solving and give students opportunities to find STEM applications to real-world scenarios.

Using this guide

This guide provides teachers and instructors with:

- activity extension ideas,
- programming projects, and
- supplementary supporting information per EdBuild, as required.

This guide is designed to complement the EdCreate EdBuilds and EdChallenges, available at <https://meetedison.com/edcreate> which comprise the bulk of the EdCreate project materials.

These builds and challenges can also be used as a starting point to develop your own customised lessons and activities.

A note on the programming projects

All of the additional programming projects provided have been written using the programming language requiring the least amount of experience with programming available for that activity. These programming tasks can also be recreated in any of the more advanced languages. For example, an activity written for use with EdBlocks can be replicated in EdPy. However, the reverse is not true.

Programming projects include a sample solution, but student solutions will vary.

EdBuild – EdTank

The EdTank is actually two builds in one: the basic EdTank and the complete EdTank with rubber band cannon.

The basic EdTank using one Edison robot which can be programmed to drive forwards, backwards and turn right or left using barcodes and a TV or DVD remote control. The complete EdTank includes a second Edison robot which controls the firing of the rubber band cannon.

| EdBuild | Difficulty rating | Recommended age | Programming skills needed |
|-------------------|-------------------|-----------------|-------------------------------|
| EdTank (basic) | 1 | 8+ | - Barcodes and remote control |
| EdTank (complete) | 2 | 8+ | - Barcodes and remote control |

To complete the programming project listed below, students will also need competency in programming using EdBlocks: <https://meetedison.com/robot-programming-software/edblocks/>

Please note: The extension and programming activities in this guide are best suited to the complete EdTank build with cannon.

Tips and tricks for this build

- Students may find it easiest to first lay out all of the EdCreate pieces onto their work surface and organise the parts into groups of the same piece type and colour. This can help students identify and use the correct pieces as they work through the build.
- You will need to reset the cannon and reload a rubber band each time you fire the cannon. This should be done manually to ensure the new band is loaded properly and that the firing 'pin' is pushed completely back into the starting location.
- For best results, use the orange bands that come with your EdCreate kit in the cannon.
- Just like many real tanks, the EdTank's design means it will be slow to turn left or right. When the EdCreate tracks are brand new, they may be extra grippy, which will make the EdTank turn even slower. You can reduce the grip by removing the tracks and lightly dusting them with talcum powder. Be sure to knock any excess powder off the tracks before putting them back onto the EdTank.

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. Turn the rubber band cannon into some hands-on maths. Set up a 'firing range' to measure how far the cannon fires. Time how long the rubber band takes to complete its flight path, then calculate the rate at which the band flies.
Note: the band flies quite quickly, so you may need to use a slow-motion video camera or app with a camera phone.

Programming project – firefighting water cannon

The scenario

Fighting fires is a dangerous but necessary task. When terrible fires (including major disasters such as large-scale forest fires or bush fires) break out, the conditions can become extremely dangerous for humans. One way to help fight these fires is by using firefighting robots.

For this scenario, the EdTank's cannon is a water cannon that can shoot a jet of water (represented by the rubber band) to help fight a fire.



1 - PHOTO: The TAF 20 firefighting robot can blast water up to 90 metres and spray water mist or foam for 60 metres. ([ABC News: Brigid Glanville](#))

Setting up the scenario

Create two parallel black lines on a light coloured (e.g. white) surface, such as a large poster, desk, table or floor. The black lines should be approximately 1.5cm (0.6 inches) wide. You can put them as far apart from each other as you like.

One black line represents the firefighting team's base, and the other represents the 'ash line' beyond which the fire is burning.

N.B. – The EdTanks will fire their rubber bands out from the 'ash line', so make sure that this line is facing somewhere where no one will get hit, such as a wall.

Programming task

Write a set of programs to control the EdTank so that the robot will drive until it encounters the 'ash line' (i.e. black line), stop, shoot out its water payload (i.e. fire the cannon) and return to the firefighting team's base (i.e. original black line).

Sample program solution (EdBlocks)

Both the top and bottom Edison robots will need to be programmed separately to complete this task.

The following two EdBlocks programs are a solution for this problem. These two programs allow the two Edison robots to communicate to each other to control how the EdTank moves and fires the cannon.

Bottom robot (driving Edison):



This program has the bottom (driving) robot first wait to receive an orange message. When the play (triangle) button on the bottom robot is pushed, this program will start, but will then 'wait for' the orange message before continuing with the program. This will allow students time to connect the top Edison to the bottom Edison.

Once the play (triangle) button on the top robot is pushed, and the bottom robot receives the orange message, this program will continue. Thus, the bottom Edison will drive while on a white surface, but stop when not on a white surface (i.e. when it encounters a black surface). It will then send out the purple message to the top robot. The bottom robot then waits until it receives the blue message before continuing. Once the bottom robot receives that blue message, it backs up, turns around and drives forward while on a white surface until it encounters a black surface, at which point the program ends.

Top robot (water cannon Edison):



When the play (triangle) button on the top robot is pushed, this program will first send out an orange message, triggering the bottom robot to continue its program. The top robot will then wait until it receives the purple message, which will trigger it to continue its program. The top robot will fire off the 'water cannon' and then send out the blue message to signal it has completed the task.

Things to look out for

- These programs use Edison's IR messages. Each message only fires out once. Occasionally, the receiving Edison can miss the message when it is sent, causing the whole program to stop as the receiving Edison is still waiting for the message. If students encounter this issue, simply have them reset and run the program again from the start. (They do not need to reprogram the

Edison robots – just push stop on each, then restart each robot’s program by pushing the play button on each in the correct order.)

- The top and bottom Edison robots will need to be separated to program each one, as well as to push the start or stop buttons on the bottom robot.
- Once both robots are programmed, press the play (triangle) button on the bottom robot, then re-attach the top robot. Set up the rubber band cannon and place the robot on a white surface in front of the firefighting team’s base (i.e. original black line) with the EdTank pointing towards the ‘ash line’. When this is set up, press the play (triangle) button on the top robot.
- If multiple students are running programs using the same coloured messages in close physical proximity to each other, they may experience ‘cross-talk’ where one set of robots receives and reacts to the message from a different set of robots.
- Students will need to experiment with the drive times to optimise how the robots perform in their physical set-up.

EdBuild – EdDigger

The EdDigger is a remote-controlled excavator, or digger, with a scoop that you can drive around (forwards, backwards and spin right or left). The digger scoop of the EdDigger can lift or lower and can carry small objects, such as parts from the EdCreate kit.

| EdBuild | Difficulty rating | Recommended age | Programming skills needed |
|----------|-------------------|-----------------|-------------------------------|
| EdDigger | 2 | 8+ | - Barcodes and remote control |

To complete the programming projects listed below, students will also need competency in programming using EdBlocks: <https://meetedison.com/robot-programming-software/edblocks/>

Tips and tricks for this build

- Students may find it easiest to first lay out all of the EdCreate pieces onto their work surface and organise the parts into groups of the same piece type and colour. This can help students identify and use the correct pieces as they work through the build.
- The top robot connects to the bottom robot on the third row of studs from the front of the bottom robot. Thus, the top robot overhangs off the back of the bottom robot by approximately 2 cm.

Activity extensions

1. It is sometimes easy to forget how much technology is all around us. The way the EdDigger looks and operates is modelled after real-world excavators – the EdDigger is a miniature working example of purpose-built technology. Find and discuss other examples of real-life technology used around your local area.
2. Learn more about how remote controls work by doing a ‘deep-dive’ investigation of the technology. Explore how remote controls and the devices they control are programmed. For example, how does a television set ‘know’ to turn up the volume when we press the ‘volume up’ button?

Programming project – remote control events

Project objective

All of the Edison robot programming languages allow you to program Edison to respond to a TV or DVD remote control within a program. In this project, students use a programming language to program the EdDigger to respond to remote control button pushes. They will need to write a set of programs to respond to remote control codes in order to drive the EdDigger forwards, backwards, turn or spin it left, turn or spin it right, lift the bucket and lower the bucket.

While you still need to use the barcodes to program the EdDigger to respond to remote control signals, using a programming language will allow students to create more control in their remote-controlled operation of the EdDigger. This project serves as a good bridging lesson to using remote control commands inside future student-created programs.

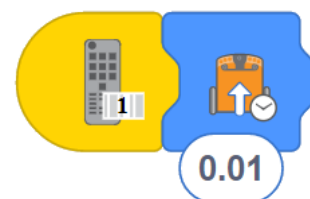


Setting up

Print out the remote control barcodes on the following pages. Students will need to program the robots with these barcodes, tying each to a button their remote control. They will then need to write and download a program into each of the robots and push the play (triangle) button to activate those programs.

N.B. – These remote control barcodes are the same ones used to drive Edison robots remotely. As such, when the students first link the robot to a button, it will react by demonstrating the default behaviour of that barcode. For example, ‘TV/DVD remote code #1’ defaults to ‘drive backwards’.

Once you download your program and press the play (triangle) button on the robot, the robot will run the programmed action instead of the barcode default. For example, if you program remote control one to be ‘drive forward for 0.01 seconds’, then while the downloaded program is running, that robot will respond to the button push affiliated with TV/DVD remote code #1 with ‘drive forward for 0.01 seconds’ not ‘drive backwards’.



Remind students that they must press the play (triangle) button on the Edison robot to run their downloaded program and override the default barcode behaviours.

Remote control barcodes

Reading the barcodes

1. Place Edison facing the barcode on the right side
2. Press the record (round) button 3 times
3. Edison will drive forward and scan the barcode
4. Press a button on your TV/DVD remote that you want to activate that function



Barcode – TV/DVD remote code #1



Barcode – TV/DVD remote code #2



Barcode – TV/DVD remote code #3



Barcode – TV/DVD remote code #4



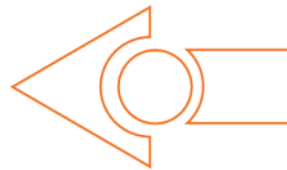
Remote control barcodes

Reading the barcodes

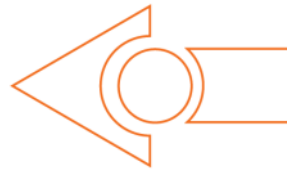
1. Place Edison facing the barcode on the right side
2. Press the record (round) button 3 times
3. Edison will drive forward and scan the barcode
4. Press a button on your TV/DVD remote that you want to activate that function



Barcode – TV/DVD remote code #5



Barcode – TV/DVD remote code #6



Programming task

Write a set of programs that control the EdDigger through a remote control by linking each of the following tasks to a remote control code:

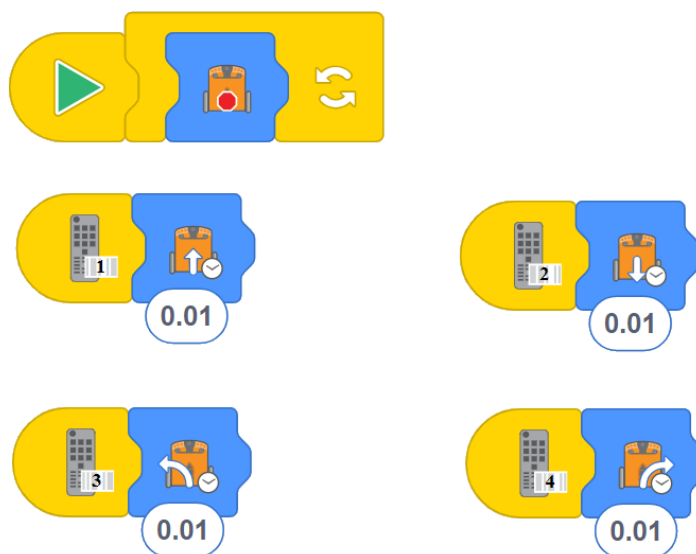
- Drive forwards a set distance or amount of time
- Drive backwards a set distance or amount of time
- Spin (turn) left for a set distance or amount of time
- Spin (turn) right for a set distance or amount of time
- Lift the bucket up
- Lower the bucket down

Sample program solution (EdBlocks)

Both the top and bottom Edison robots will need to be programmed separately to complete this task.

The following two EdBlocks programs are a solution for this problem. These two programs allow the two Edison robots to be controlled by remote controls to control how the EdDigger moves and operates the bucket.

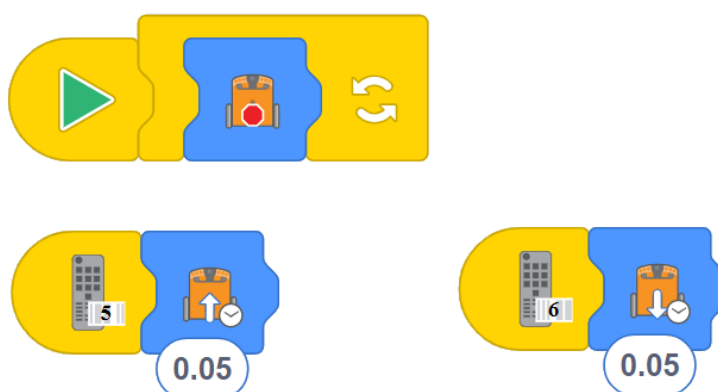
Bottom robot (driving Edison):



The main program says to stop the motors forever, meaning that when there are no remote codes detected, the program effectively tells the Edison robot to do nothing. If a remote code is detected, however, this main program is interrupted and the robot will run the action of that remote code event. That's why the program also specifies an action for a given duration for Edison to perform for each of four remote codes:

| Remote code # | Action | Duration |
|---------------|----------------|----------|
| 1 | Drive forward | 0.01 |
| 2 | Drive backward | 0.01 |
| 3 | Turn left | 0.01 |
| 4 | Turn right | 0.01 |

Top robot (scoop bucket Edison):



The main program says to stop the motors forever, meaning that when there are no remote codes detected, the program effectively tells the Edison robot to do nothing. If a remote code is detected, however, this main program is interrupted and the robot will run the action of that remote code event. That's why the program also specifies an action for a given duration for Edison to perform for each of two remote codes:

| Remote code # | Action | Duration | Results in |
|---------------|----------------|----------|--------------------|
| 5 | Drive forward | 0.05 | Bucket lifts up |
| 6 | Drive backward | 0.05 | Bucket lowers down |

Things to look out for

- The top and bottom Edison robots will need to be separated to program each one, as well as to push the start or stop buttons on the bottom robot.
- If multiple students are running programs using the remote control codes in close physical proximity to each other, they may experience 'cross-talk' where one set of robots receives and reacts to the remote control from a different set of robots.
- Remind students that they must press the play (triangle) button on the Edison robot to run their downloaded program and override the default barcode behaviours.
- Students should experiment with the drive times to optimise how the robots perform. This is a good opportunity for students to experiment with how far they want the robot to move per button press.
- Remind students that they should use different codes (4 for the bottom robot and 2 for the top robot) for the two robots and that each code needs to be linked to a unique button on their remote control.

Programming project – semi-automated remote control

The scenario

Remote-controlled construction machines allow people to handle potentially hazardous tasks or environments more safely. An example of this is construction projects involving demolition, where it is best to keep operators away from flying debris.

In some situations, much of a task can be pre-set, which limits the amount of active operation required by the controller. Pre-setting, or automating, this type of task can save time and limit the chances of human error.



2 - PHOTO: [Caterpillar's 'RemoteTask'](#) system allows operators to remotely control construction machines with precision using a 'belly box' control system.

Setting up the scenario

Create a 'worksite' with a physical objective for students to solve by creating a semi-automated EdDigger program. This can be as simple as putting some small objects, such as EdCreate pieces, on one side of the floor to represent rubble and creating a drop-off zone where the rubble needs to be delivered.

Students can also design their own physical worksites to create programs to solve or work in groups solving a different group's worksite setup.

N.B. – For any remote control commands students want to use in their semi-automated programs they will need to program the robots to respond to those commands. Refer to programming project 'remote control events' in this guide for details on programming using barcodes and the Edison robot programming languages.

Programming task

The programming task will depend on the physical worksite setup. For this example, the physical worksite set-up and task details are:

- Rubble is on the ground on one side of the worksite and needs to be cleared into the drop-off zone on the opposite side of the worksite.
- The program should only require one button press by the remote control operator.

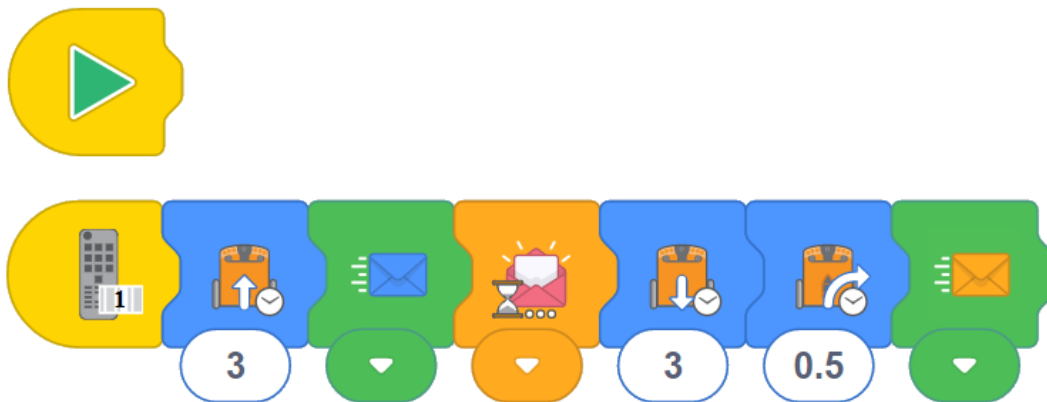
Sample program solution (EdBlocks)

Both the top and bottom Edison robots will need to be programmed separately to complete this task.

The following two EdBlocks programs are a solution for the example detailed above.

These two programs allow the two Edison robots to communicate to each other to control how the EdDigger moves and manipulates the scoop bucket.

Bottom robot (driving Edison):



The main program is empty, effectively meaning the robot will do nothing when the play (triangle) button is pressed. Once remote code #1 is detected, however, the robot starts a sub-routine. This triggers the robot to drive forward for 3 seconds, then send the blue message, signalling the top robot to act. This robot then waits for the pink message. Once the bottom robot receives the pink message, it drives backwards for 3 seconds, turns right for 0.5 seconds and sends the orange message, signalling the top robot to act again. This ends the sub-routine for this robot, and the main program resumes.

Top robot (scoop bucket Edison):



The main program is empty, effectively meaning the robot will do nothing when the play (triangle) button is pressed. Once this robot receives the blue message, it will move its motors forward for 0.05 seconds, lifting up the bucket scoop. It then waits for 0.5 seconds to allow the 'rubble' to settle in the bucket before sending out the pink message to signal the bottom robot it is ready to move. The top robot then waits for the orange message, signalling that the EdDigger is at the drop-off zone, at which point the top robot lowers the bucket scoop to drop off the rubble. This ends the sub-routine for this robot, and the main program resumes.

Things to look out for

- The top and bottom Edison robots will need to be separated to program each one, as well as to push the start or stop buttons on the bottom robot.
- If multiple students are running programs using the remote control codes or Edison messages in close physical proximity to each other, they may experience 'cross-talk' where one set of robots receives and reacts to the remote control or message from a different set of robots.
- Remind students that they must press the play (triangle) button on the Edison robot to run their downloaded program and override the default barcode behaviours.
- Students should experiment to find a solution that best suits the 'worksite' setup and problem. This is a good opportunity for students to work through the natural trial-and-error nature of programming by testing different inputs, such as the drive times, as well as actions and sequences of actions to optimise how the robots perform.

EdBuild – EdRoboClaw

The EdRoboClaw is a remote-controlled articulated robotic arm, which can be programmed using the barcodes in this guide and controlled with a standard TV or DVD remote control. You will be able to drive the EdRoboClaw forwards, backwards and spin it right or left. You will also be able to open and close the claw to pick up and carry an object, such as one of the EdCreate beams.

| EdBuild | Difficulty rating | Recommended age | Programming skills needed |
|------------|-------------------|-----------------|-------------------------------|
| EdRoboClaw | 3 | 8+ | - Barcodes and remote control |

To complete the programming project listed below, students will also need competency in programming using EdBlocks: <https://meetedison.com/robot-programming-software/edblocks/>

Tips and tricks for this build

- Students may find it easiest to first lay out all of the EdCreate pieces onto their work surface and organise the parts into groups of the same piece type and colour. This can help students identify and use the correct pieces as they work through the build.
- The top robot connects to the bottom robot on the second row of studs from the front of the bottom robot. Thus, the top robot overhangs off the back of the bottom robot by approximately 1 cm.
- The claw is composed of 3 'fingers' – two parallel fingers which are stationary (made from grey beams) and the frontmost finger which moves. The row of gears in the articulated arm controls this forward finger, including its positioning relative to the stationary fingers. The alignment of the forward most two gears can affect how the moving finger sits relative to the stationary fingers when the claw is fully open. When the claw is fully open, and the EdRoboClaw is sitting on a flat surface (such as a table or desk), the moving finger should be high enough that one of the EdCreate grey beams can slide under it between the finger and the table. If the moving finger is not this high, try gently separating the front of the arm and rotating the frontmost gear by one or two teeth clockwise independently of the next gear. You will be able to see this move the front finger. Reconnect the gears and the arm.
- The EdRoboClaw can pick up the 7-hole long grey beam from the EdCreate kit well, but will not be as able to pick up and carry objects which are round, such as a pen.

Activity extensions

1. Encourage students to see what they can and cannot pick up with the EdRoboClaw. Explore the physics behind why these items can or cannot be picked up and carried by the claw arm.
2. Investigate the types of things articulated arms are used for in the real-world, including the manufacturing sector.
3. Is it just the 'dull, dirty and dangerous' task or are we moving towards a future where 'humans need not apply' – what is the place of robots and humans in the workforce of today and tomorrow? Research the types of work robots currently do and may do in the future. This research can form the basis of reports, presentations, discussions or structured debates.

Programming project – hazardous material removal

The scenario

Many types of materials can be hazardous to people, such as biomedical waste, by-products from manufacturing or radioactive material. To keep people safe, these types of materials need to be safely moved and disposed of away from populated areas. Remotely operated vehicles with robotic arms are one way to handle this task.

For this scenario, you will need to use the EdRoboClaw and a 7-hole long grey beam from the EdCreate kit to represent the hazardous material.



3 – PHOTO: This MATS vehicle, equipped with a robotic arm, handles a highly radioactive material used for a trial at [DRDC](#) – Suffield Research Centre. The MATS system allows the operator to stay a safe distance away during manipulation of the source.

Setting up the scenario

Create a 'disposal zone' by marking out an area, such as a square, with black lines on a white background surface. The black lines should be approximately 1.5cm (0.6 inches) wide, and the white background surface should be a large area, such as a large poster, desk, table or floor. The inside of the 'disposal zone' is where the hazardous material should be put, but the robots should not enter the zone (the arm can reach into the disposal area, however). Once the material is dropped off, the robot should retreat away from the zone.

N.B. –For this program, it will be easiest to start the robot with the claw already hovering open over the 7-hole long grey beam from the EdCreate kit representing the hazardous material.

Programming task

Write a set of programs to control the EdRoboClaw so that the robot will pick up the hazardous material, drive until it finds the 'disposal zone' (i.e. one of the black lines), stop, drop off the hazardous material, then retreat from the disposal zone safely.

Sample program solution (EdBlocks)

Both the top and bottom Edison robots will need to be programmed separately to complete this task.

The following two EdBlocks programs are a solution for this problem. These two programs allow the two Edison robots to communicate to each other to control how the EdRoboClaw moves and manages the articulated claw arm.

Bottom robot (driving Edison):



This program has the bottom (driving) robot first wait to receive a blue message. When the play (triangle) button on the bottom robot is pushed, this program will start, but will then 'wait for' the blue message before continuing on with the program. This will allow students time to connect the top Edison to the bottom Edison.

Once the play (triangle) button on the top robot is pushed, this program will continue. Thus, the bottom Edison will drive while on a white surface, but stop when not on a white surface (i.e. when it encounters a black surface). It will then send out the pink message to the top robot. The bottom robot then waits 2 seconds before continuing, allowing the claw arm to drop off the hazardous material. The bottom robot then backs away from the disposal zone, and the program ends.

Top robot (claw arm Edison):



When the play (triangle) button on the top robot is pushed, this program will first pick up the hazardous material. It will then send out a blue message, triggering the bottom robot to continue its program. The top robot will then wait until it receives the pink message, which will trigger it to continue its program. The top robot drops off the load, finishing its program.

Things to look out for

- The top and bottom Edison robots will need to be separated to program each one, as well as to push the start or stop buttons on the bottom robot.
- Once both robots are programmed, press the play (triangle) button on the bottom robot, then reattach the top robot. Make sure the top robot is reattached at the correct row of studs and that the reattached robots have all parts of the arm in the correct spots. Have students refer to the final build picture for reference.
- Start the robot with the claw open. Place the 7-hole long grey beam from the EdCreate kit representing the hazardous material directly below the open claw.
- If multiple students are running programs using the same coloured messages in close physical proximity to each other, they may experience 'cross-talk' where one set of robots receives and reacts to the message from a different set of robots.
- Students will need to experiment with the drive times to optimise how the robots perform in their physical set-up.
- These programs use Edison's IR messages. Each message only fires out once. Occasionally, the receiving Edison can miss the message when it is sent, causing the whole program to stop as the receiving Edison is still waiting for the message. If students encounter this issue, simply have them reset and run the program again from the start. (They do not need to reprogram the Edison roots – just push stop on each, then restart each robot's program by pushing the play button on each in the correct order.)

EdBuild – EdCrane

The EdCrane is a remote-controlled crane with a magnetic hook which you can raise and lower. The magnetic hook of the EdCrane can lift or lower small ferrous-metal objects, such as metal paper clips or small screws, as well as the metal peg from the EdCreate kit. The metal peg can be attached to one of the parts from the EdCreate kit, such as a 3-hole beam, and used with the magnetic hook.

| EdBuild | Difficulty rating | Recommended age | Programming skills needed |
|---------|-------------------|-----------------|---|
| EdCrane | 4 | 10+ | <ul style="list-style-type: none">- Barcodes and remote control AND- EdScratch OR- EdPy |

To complete the programming project listed below, students will need competency in programming using EdPy: <https://meetedison.com/robot-programming-software/edpy/>

Tips and tricks for this build

- Students may find it easiest to first lay out all of the EdCreate pieces onto their work surface and organise the parts into groups of the same piece type and colour. This can help students identify and use the correct pieces as they work through the build.
- It is strongly recommended that the magnetic hooks be set up in advance of the build with adult supervision to ensure that these are set up correctly. Once the hook is set up the first time, it can be used as a single part over and over in the EdCrane build.
- The magnetic hook's spool can spin both clockwise and counter-clockwise. Whenever the string of the hook is fully extended out from the spool (i.e. the string is released all the way), whichever way the hook starts to wind will become the 'raise hook' direction. The opposite direction will be the 'lower hook' direction.
- Remind students that they must press the play (triangle) button on the Edison robot to run their downloaded program and override the default barcode behaviours.
- If multiple students are running programs using the remote control codes in close physical proximity to each other, they may experience 'cross-talk' where one set of robots receives and reacts to the remote control from a different set of robots.

Activity extensions

1. Turn the magnetic hook into some hands-on maths. Test the strength of the magnet by seeing how much weight it can hold. Alternatively, work out the 'load-limit' of the crane by working out the total force it can tolerate on the string without compromising the crane's stability.
2. Explore how the download programs work. Being able to understand what a program that someone else has written does by parsing through it is a valuable skill in computer science. See a basic explanation for each of the programming languages' programs below.

The EdCrane program in EdScratch

The share code <https://www.edscratchapp.com?share=5DMQ3XDw> opens the official EdCrane program in EdScratch.

The comments (pink 'Comment' blocks) will help students read through this program.

This program contains a main program (consisting of an empty infinite loop) and a subroutine triggered by any 'Remote code received' event. This program will tell Edison to do nothing forever (rather, the program will run until it is ended by pushing the stop button on Edison) but any time a remote code is received, Edison will execute one of several possible actions. The program reads the last seen remote code, and then works through a series of nested 'if' statements to determine the action that should be taken whenever a remote code is detected:

| If remote code # | Action | Speed | Results in |
|---|--|-------|---|
| 1 | Set right motor to forwards for .2 sec | 5 | Magnetic hook spool spins clockwise |
| 2 | Set right motor to backwards for .2 sec | 5 | Magnetic hook spool spins counter-clockwise |
| 3 | Set left motor to backwards for .2 sec | 5 | EdCrane spins clockwise |
| 4 | Set left motor to forwards for .2 sec | 5 | EdCrane spins counter-clockwise |
| Final action (only action for remote code other than 1, 2, 3, or 4) | Stop both motors, clear the remote sensor data | n/a | EdCrane does nothing |

The EdCrane program in EdPy

The share code www.edpyapp.com/share/99ihb opens the official EdCrane program in EdPy.

The comment code (the lines starting with #) will help students read through this program.

This program sets variable 'remoteValue' to 0, then enters into a 'while True' (infinite) loop. Inside the loop, the value of variable 'remoteValue' is set to be the value of the last seen remote code. An if/elif/else statement then runs to determine the action that should be taken:

| If remoteValue = | Action | Speed | Results in |
|----------------------------|-----------------------|-------|---|
| 1 | Drive forward left | 2 | Magnetic hook spool spins clockwise |
| 2 | Drive backwards left | 2 | Magnetic hook spool spins counter-clockwise |
| 3 | Drive forward right | 3 | EdCrane spins counter-clockwise |
| 4 | Drive backwards right | 3 | EdCrane spins clockwise |
| Else (all other instances) | Stop | 1 | EdCrane does nothing |

Programming project – marine port crane

The scenario

Commercial marine ports handle huge volumes of cargo ships which enter and leave every day. The cargo on these ships is often packed in container boxes. When many ships need to be unloaded quickly, ports need effective ways to remove containers from a ship, shift them into the yard and stack them up with other containers.



4 - PHOTO: Remote crane operation is an integral part of automation, an important component for marine terminal operation. This improves safety, working environment and overall operator productivity in container terminals. ([ABB](#))

Automatic stacking cranes, which are operated by a remote crane operator, are one of the most efficient ways to handle this situation. With most of the crane cycle handled automatically, the crane operator's main task is to supervise the process and take action only when required.

Setting up the scenario

Create a 'marine port' with a physical objective for students to solve by creating an automated EdCrane program. A good physical setup will require the students to lower and raise the magnetic hook to at least two different heights as well as spin the crane.

Programming task

The programming task will depend on the physical marine port setup. For this example, the physical marine port set-up and task details are:

- A ship has docked to the right of the crane. The next container to be removed is approximately 2 cm higher than 'ground level' of the dock. (**Note** – the level of the bottom of the crane base is considered to be 'ground level'.) The container needs to be stacked at a point approximately 12 cm higher than ground level in the main yard, to the left of the crane. The crane should then reset to its original starting position.
- The container is represented by a 3-hole EdCreate beam with the metal peg inserted into the centre hole.
- The program should be fully automated.
- The student can assist the container's attachment to the magnetic hook in terms of aligning the container and hook manually. The student may remove the container from the hook manually.

Sample program solution (EdPy)

A sample solution to this problem can be accessed through the EdPy share code:

<https://www.edpyapp.com/share/Udcih>

The program contains comment code (lines starting with #) to explain how the program works.

The program first winds the hook completely up, then turns the crane 90 degrees to the right. The hook is lowered to the height of the container, and the crane pauses to enable a manual check by a human to be performed, ensuring the container is attached. The program then continues, lifting the hook and turning the crane 180 degrees into the yard. The hook is lowered to the container stack level, and the crane pauses to enable the manual removal of the container by a human. Finally, the program lifts the hook away from the stack, returns to its starting position, and resets by fully releasing the hook down.

Things to look out for

- This program requires that the magnetic hook start all the way down, with no string wound onto the magnetic hook spool.
- Students should experiment to find a solution that best suits their 'marine port' physical setup and programming task. This is a good opportunity for students to work through the natural trial-and-error nature of programming by testing different inputs, such as the drive times, as well as actions and sequences of actions to optimise how the robot performs.
- The share code provided can be used as a starting point for students to modify the program to suit their own physical setup.

EdBuild – EdPrinter

The EdPrinter is a pen plotter which is run using two Edison robots programmed using EdPy. By attaching a writing utensil, such as a felt-tip pen, inserting paper into the base of the EdPrinter, and using a set of basic functions in EdPy, the EdPrinter can be programmed to create a range of different shapes.

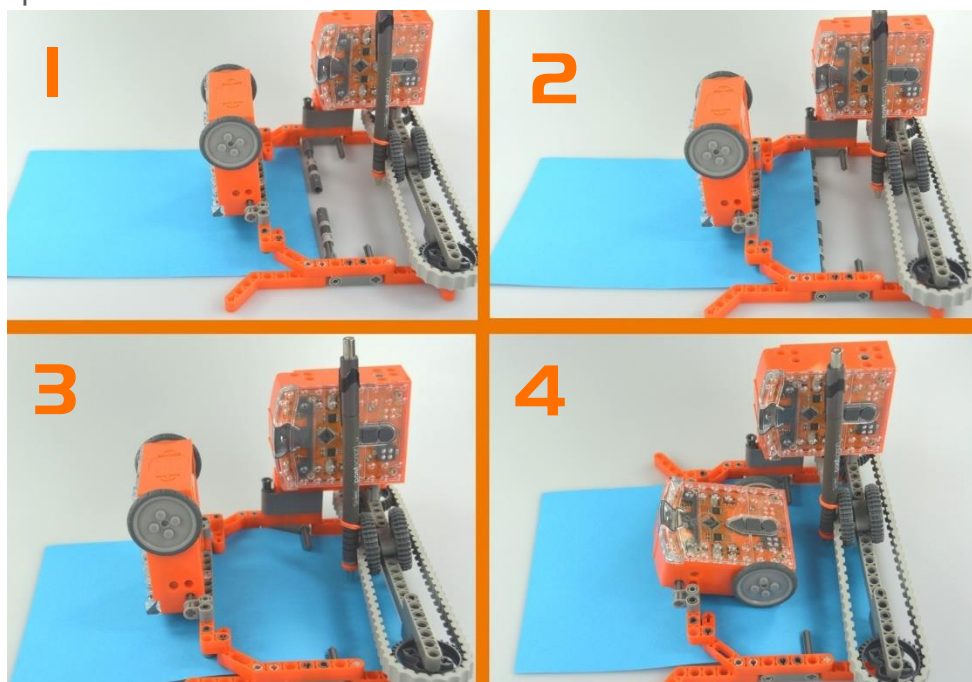
| EdBuild | Difficulty rating | Recommended age | Programming skills needed |
|-----------|-------------------|-----------------|---------------------------|
| EdPrinter | 5 | 12+ | - EdPy |

To complete the basic programming for this build and the activities listed below, students will need competency in programming using EdPy:

<https://meetedison.com/robot-programming-software/edpy/>

Tips and tricks for this build

- Students may find it easiest to first lay out all of the EdCreate pieces onto their work surface and organise the parts into groups of the same piece type and colour. This can help students identify and use the correct pieces as they work through the build.
- Paper is loaded into the bottom of the frame of the EdPrinter as follows:



- The interior frame base of the EdPrinter is approximately 15cm wide. Standard-weight (80-150 GSM) A5-size copy paper (14.8cm x 21cm) works best with the EdPrinter. Standard USA letter sized paper (8.5 inches x 11 inches) or A4 paper (21cm x 29.7cm) halved width-wise will also work.
- The EdPrinter works best with a writing utensil that writes with low pressure, such as a felt-tip pen or marker. You can also use a coloured pencil, so long as the pencil's lead is fairly soft.

- Be sure to push the paper back so that the pen sits on the paper. The EdPrinter works best if at least 3cm of paper extends back beyond the pen.
- The EdPrinter uses separate EdPy programs in the paper-controlling and pen-controlling robots. The paper-controlling robot is reactionary – responding to the commands of the pen-controlling robot, which acts as the primary driver of the EdPrinter.
- Be sure that the paper-controlling robot's play (triangle) button is pressed before the pen-controlling robot's play (triangle) button is pressed.
- Start with the pen closest to the pen-controlling robot.

Activity extensions

1. Explore the history of peripheral computing devices with an emphasis on printers and the historical use of plotters in computing.
2. Discover how plotters are used today to 'print' on materials that cannot be fed through a printer, such as tracing out the wiring path on a copper board in creating printed circuit boards.
3. Visually, seismometers and pen plotters share some characteristics. Are seismometers plotters? Compare and contrast the technology and functionality of seismometers, printers and plotters.

Programming project – make your own shape

The challenge

The base program for the EdPrinter, comprised of the paper-controlling robot program (available from share code

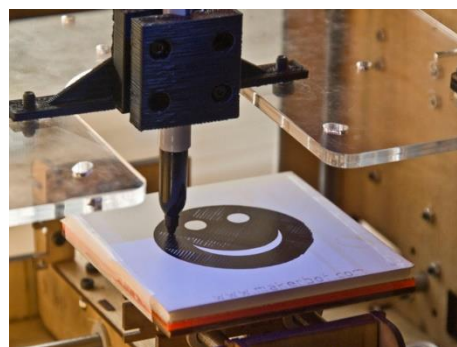
www.edpyapp.com/share/oLn1L) and the pen-controlling robot program (available from share code www.edpyapp.com/share/GzSRI), creates a rectangle.

What other shapes can you make?

The open-ended programming task

Challenge students to create their own program for the EdPrinter to create a shape other than a rectangle or another drawing.

Students should use the set of basic functions for controlling the EdPrinter used in the pen-control base program (available from share code www.edpyapp.com/share/GzSRI) to help them create their own program. These four functions enable the EdPrinter to draw lines in four directions: to the right, to the left, to the front and to the back. By rearranging the order in which these functions are called and adjusting the value of the input variable, students can create their own programs.



5 - PHOTO: Pen plotters can be made from many different materials, like this 3D printer option on Thingiverse, and used to print a wide range of images.

To help students, you may also want to provide them with share code www.edpyapp.com/share/bHgCF which has the function definition and function call program to create a diagonal line forwards to the right. Students can use this function in addition to the four base functions to create additional shapes. Students can also try to create other diagonal line functions using the one provided as a starting point.

N.B. – The official EdPrinter paper-controlling robot program (available from share code www.edpyapp.com/share/oLn1L) can be used as-is for any EdPrinter programs. To make your own EdPrinter programs, change the pen-controlling robot program only.

Things to look out for

- The 'sendValue' variable used in the EdPrinter programs messaging to communicate between the two robots does not need to be changed in order to change the program.
- Be sure that the paper-controlling robot's play (triangle) button is pressed before the pen-controlling robot's play (triangle) button is pressed.

EdCreate EdChallenges

An EdChallenge is an open-ended challenge with no set solution or a single way of approaching the problem. Open-ended challenges like the EdCreate EdChallenges necessitate students to think creatively, be resourceful and inventive. These challenges are designed to encourage creative problem solving, exploration, and the tangible application of learned skills.

The EdCreate EdChallenges are designed to require a combination of applied engineering and computer programming skills. Each challenge lays out a scenario or task and any success criteria for completing the challenge. Students may then invent whatever they like to solve the challenge.

All of the EdCreate EdChallenges require the use of an Edison robot. Any of the components included in the EdCreate creator's kit, as well as additional Edison robots and any other materials selected by students, may also be used.

You can create your own design and programming challenges for students using the EdChallenges as a base. Alternatively, you can modify the existing EdChallenges to give additional requirements, limitations or options.

Example challenge

In the EdCreate EdChallenge *Let's invent a burglar alarm*, students are presented with the following scenario:

You have been entrusted with a valuable treasure to protect. But there are sneaky thieves out to get it! Make a burglar alarm to keep the treasure safe and sound.



They are also given the build success requirements:

To complete this burglar alarm challenge, you need to build an alarm using your Edison robot. Your alarm must:

- use at least one of Edison's sensors to trigger the alarm, and
- do something to scare the intruder away.

Students can invent, build and program whatever solution they like to solve the problem.