Edison

Edison Card Coding
Teacher Guide

Teach screen free programming with the Edison V3 robot

microbric
motivate • create • educate

Document number: 3.2.4.11.1

Contents

Introduction to Edison Card Coding 3
Teacher introduction 4
Lesson one – Sequence coding 6
Lesson two – Loops and repeats 11
Lesson three – Drive forever and wait until 16
Lesson four – Make a function 19
Lesson five – Project 21

Introduction to Edison Card Coding

Welcome to teaching screen free robotics with the Edison robot!

This teacher guide has been developed in conjunction with the Edison Card Coding lessons and will allow you to get the most out of teaching your students the basics of programming.

Why programming?

While not every student will become a computer programmer, there are some huge advantages for students that can think computationally. Computational thinking is a problem-solving process that can be applied far beyond computer programming. It allows problems to be broken down and for logical step by step solutions to be found and implemented. These skills can be applied anywhere from everyday problem solving all the way through to an engineering career.

Why screenless?

The most efficient and effective way to develop a program is using a computer, however this does create some problems for some users and learning.

Age

Some students are young and not yet proficient at using a computer. This results in students struggling with technical problems of using the computer, rather than learning about programming.

Computer availability

Not all classrooms and students have individual access to a computer.

Truly collaborative

Collaboration when problem solving is very important and a valuable skill. The process of searching for the coding cards, laying them out, discussing which card goes where and then scanning each card is far more collaborative than one student holding a computer mouse or tablet. For students that work in small groups of two to four, expect to hear a lot of discussion (collaboration) about which card goes where.

Teacher introduction

Getting prepared

Go through at least some of the lessons outside of the classroom before teaching them to the class. You will then go into the classroom with greater confidence and a better ability to help students.

The thinking behind these lesson

The lessons in this document allow students to start learning the basic concepts of programming, then with each activity increase their abilities and understanding. This gradual learning approach provides them with the skills and confidence to complete the final project.

Lesson structure

Each of the first four lessons is comprised of:

1. Introductory information about the lesson
2. One to four programming activities
3. Two challenges
4. The required additional coding cards

The fifth lesson is an open-ended project and includes prompts for creating and recording the project.

Printing

If you choose to print the lessons and coding cards lesson by lesson, use the buttons at the top of the title page. Each button will send only the selected lesson to the printer.

Coding card reusability

There are two options to increase the robustness of the coding cards for reusability.

1. Laminating

Use a matte laminate (with no gloss) and ensure that you test several cards with a few different Edison V3 robots before laminating the entire set.

2. Thicker stock

Ensure that the thicker stock paper is not gloss and test a few cards with Edison before printing the entire set.

Firmware updates

The Edison V3 features are always being updated and increased. Be sure to check that before starting a lesson that the Edison robots have the latest firmware updates.

Updating firmware is a fast and easy process. Go to any of the online programming apps at:

edblocksapp.com

edscratchapp.com

edpyapp.com

Select Menu > Update firmware and follow the on-screen prompts.

Lesson one – Sequence coding

Coding a sequence is the first step in programming. It forms an important foundation that all other aspects of programming are built upon.

Students should already be familiar with the concept of sequence, as this is how a cooking recipe works. Each step in a recipe is completed one by one in a set order of sequence.

Getting started

In the first part of lesson one the students become familiar with the Edison robot and coding cards.

Meet Edison

The first part of lesson one involves looking at the Edison robot's sensors, lights and wheels.

To encourage students to look closely at Edison ask them to count how many little robot images they can find on the Edison robot. The total number is four.

Edison buttons

The students also need to understand what each of the three buttons do.

Coding cards

The students should also become familiar with the coding cards.

The coding cards are colour coded to help with recognition:

Blue – Perform driving actions

Teal – Perform output actions, such as beeps and lights

Green – Perform program related actions, such as program start and end

The main elements of the coding cards are:

Icon – The icon provides a quick visual reference to what the card does in a program

Action – The action is a short description of what the card does in a program

Description – The description provides more details about what a card does in a program

Barcode – The barcode is the part of the coding card that the Edison robot reads. Each barcode is like a letter of the alphabet, but for computers.

Barcodes can be found on just about every product in a shop. The students have probably used them before in the self-checkout aisle at a supermarket.

A program

Here the students see how a program is laid out using the coding cards.

Points to emphasise:

- ✓ Every program starts with a 'Start program' card
- ✓ Programs go left to right, just like writing
- ✓ The program 'runs' when the robot is doing the things in the program

Program one – A simple drive program

In this activity students create their first program using the coding cards. They learn to layout the cards in the correct order, then scan the cards into the Edison robot one by one. They then run the program to see what it does.

Points to emphasise:

- ✓ The Start card requires three presses of the round button
- ✓ Always scan each card one by one
- ✓ The Edison robot does exactly what the program tells it to do and, in that order

Program two to five – Drive through a maze

In these activities students use the printed mazes. These are good initial activities to get students thinking about how the robot moves and the importance of the sequence of the coding cards.

Teacher note

Turning accuracy can be improved by performing a turn calibration. More information here: https://meetedison.com/turn-calibrate-the-edison-v3-robot/

Edison turns and drives best on a flat smooth surface. Carpet can affect the accuracy of the 90 degree turns.

Points to emphasise:

- Review programs before scanning. Students should think about each move the robot will make and whether it is correct, then scan the cards.
- Congratulate students for writing their first robot program
- Failure is great! Just try again!!

Challenge one – Dance sequence

In this activity students create a sequence of moves, flashing lights and beeping to have the robot dance.

Points to emphasise:

- Dancing is made up of repetitive moves. For example, a person might: step forward, jiggle, step backwards, giggle…
- The robot should end the dance in a similar position to where it started
- Students can dress up their robot for extra flare

Challenge two – Make and drive a maze

In this activity students create their own maze to solve.

Teacher note

Turning accuracy can be improved by performing a turn calibration. More information here: https://meetedison.com/turn-calibrate-the-edison-v3-robot/

Edison turns and drives best on a flat smooth surface. Carpet can affect the accuracy of the 90 degree turns.

- ✓ Each length of the maze must be in multiples of 10cm
- ✓ Each turn must be 90 degrees
- ✓ Students can work together and combine their coding cards to make larger programs

Lesson two – Loops and repeats

Loops and repeats in programming are very important programming concepts. They allow a computer or robot to repeat the same task or action many times.

Loops and repeats are so important in programming that every piece of software uses them in some form.

What is a loop?

In programming a loop is when a program jumps back to an earlier part of the program and repeats all of the program inside of the loop.

The Edison 'Loop to start' card goes at the end of a program (instead of the 'End program' card). The 'Loop to start' card jumps the program back to the start and will keep repeating all of the program forever.

Points to emphasise:

- ✓ Loops repeat the program forever
- ✓ The 'Loop to start' card replaces the 'End program' card
- ✓ A lot of the programs in these lessons will use the 'Loop to start' card

What is a repeat?

Unlike a loop that goes forever, a repeat works for a set number of times.

The repeat cards allow programs to be shorter (use less cards). For example, a repeat 4 times card and a drive 10 cm card can make the Edison drive 40 cm. Without using a repeat card, the program would require four drive 10 cm cards, that's twice as many cards.

Points to emphasise:

- ✓ Repeats work for a set number of times
- ✓ Using repeats in your program can reduce the number of cards that you need
- ✓ Always look for opportunities to use a repeat in your programs

Program one – Loop it

In this activity students create their first looping program. This program illustrates how just two coding cards with a loop can create an interesting result. Students then have the opportunity to mix it up, by adding flashing lights to their looping program.

Students should also attempt creating their own program that does something else forever (hint: use light, sound, movement).

Points to emphasise:

- ✓ People don't like doing the same thing over and over, but robots don't mind at all
- ✓ Keep trying different ideas and arrangements of the cards
- ✓ Experimentation is good!

Program two – Repeating

In this activity students create their first program using a repeat card.

Students should try different repeat cards in their program to see what happens.

Points to emphasise:

- ✓ The repeat cards only cause the very next card to repeat
- ✓ The available repeat cards are only for 2, 3, 4, 5 and 10 times
- ✓ Experimentation is good!

Program three – Driving further with repeat blocks

In this activity students gain deeper understanding by putting the repeat cards to good use.

Students will measure or estimate the length of their desk, then select the appropriate repeat card to drive their robot from one side to the other side of their desk without driving over the edge.

Students can also get creative and add lights or sound to their program.

In this activity students might find issues with the restricted number of repeat cards. This will be solved in the next activity.

Points to emphasise:

- Be ready to catch the robot if it drives too far!
- Have the robot drive as close to the edge as possible without driving over
- Once again, experimentation is good!

Program four – Repeat the repeat

In this activity students learn how to use two repeat cards together. They learn that the repeat cards can multiply together to provide more repeat options.

They repeat the previous activity using two repeat cards. They should now be able to get their robot to drive more accurately to the edge of their desk.

This is obviously a bonus maths lesson, however to take the maths lesson further ask the students to drive their robots 70 cm (7 repeats) or 130cm (13 repeats) using just one drive card. Welcome to prime numbers!

Points to emphasise:

- Just in case anyone was in doubt, does everyone now agree that maths is important!?
- Always look for opportunities to use repeating repeat cards in your programs
- Count the number of drive movements to check that your program's math is correct

Challenge one – Robot security guard

In this activity students program the Edison robot to patrol a 'valuable' classroom item. The robot does this by driving around the item continuously (forever) using lights and sound to scare away any would-be thieves.

Points to emphasise:

- The robot must completely drive around the item

- As a hint, the robot should end each loop of the program in the same position and facing the same direction that it started.
- Use light and sound for maximum affect. Hint: add this after the robot can correctly drive around the item

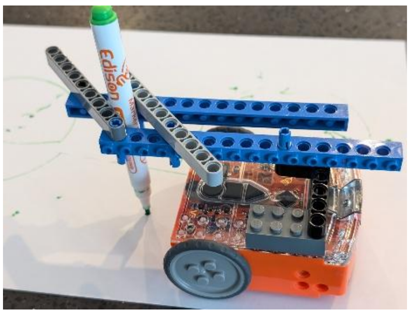
Challenge two – Draw a pattern

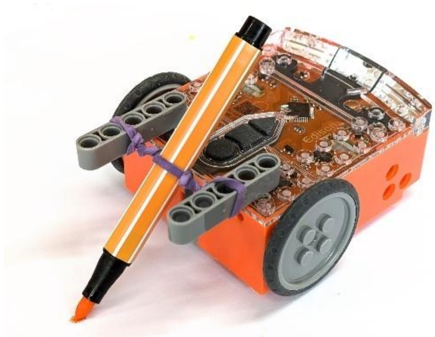
In this activity students build a pen holder onto the Edison robot using LEGO bricks or other building materials, such as cardboard. Then write a program that draws a repeating pattern.

Building the pen holder is an engineering challenge that requires experimentation and failure. Encourage students to keep trying by rapidly iterating their designs.

Points to emphasise:

- Use a rubber band to hold the pen
- Use a felt tip pen or marker
- Getting the pen pressure right can be the trickiest part. If the pressure of the pen is too weak, then the line is faint. If the pressure of the pen is too great, then the robot has trouble driving due to the friction
- Experiment, experiment, experiment!





Note that the engineering elements of the activity can be skipped by using the Edison robot's EdSketch pen holder and marker pen.

Lesson three – Drive forever and wait until

In this lesson students will learn about the single most important area of robotics, when they make their robot **autonomous**!

A robot is considered to be autonomous when it can act without human input and can make decisions on its own based on inputs from its environment. A good example of this in the real world is a robot vacuum cleaner.

A robot vacuum cleaner navigates its way around walls and furniture without falling down the stairs. This is possible because the robot vacuum has sensors that tell it about its environment. When it senses a wall, it makes a decision to change its direction by turning around. This is an autonomous behaviour and Edison can do this too.

For students to master this aspect of robot programming they need to understand two new card types.

1. Set drive cards

The set drive cards are different to the other drive cards that have the robot drive for a set distance or angle. Students can think of the set drive cards as just turning on the motors. Turning off the motors or changing the robot's direction requires another drive card.

2. Wait until cards

A wait until card pauses the program from running further until the card's 'condition' is met. When the condition is met, then the program continues to run. This is how the Edison robot makes a decision based on its sensors.

Program one – Drive for infinity

In this activity students create a program that causes the Edison to drive (for infinity) until the robot encounters a black line.

In this program the Edison is making a programmed decision about how to respond to the input from a sensor. This is a basic form of autonomous behaviour.

The students then repeat the activity in lesson two of driving the length of their desk, but this time a black line (black tape) is used to indicate the edge. Now the desk can be any length and the robot's autonomous behaviour will prevent it from driving over the edge. The human students no longer need to measure the desk and program the exact distance for the robot to travel.

To mix it up, students can use the robot arena and see if they can program the Edison robot to stay within the robot arena. This program requires that the robot drive, wait until black, turn 180 degrees and loop back to the start of the program.

Points to emphasise:

- ✓ When a robot can make a decision based on its environment it is an autonomous robot
- ✓ Autonomous robots are the most useful robots
- ✓ Autonomous robots save time for their human programmers

Program two – Follow the edge of a black line

In this activity students create a program that makes the robot follow a black line. In robotics, programming a robot to follow a line is like a rite of passage.

Students use the robot arena to test their program. They can also create their own line using black tape on a white desk or draw a thick black line (1.5 cm thick) on a large sheet of paper.

Points to emphasise:

- ✓ Press the triangle button to start the program with the robot next to the line, **not on the line**
- ✓ Use the 'Set drive right' and 'Set drive left' cards, not the turn 90 degrees cards
- ✓ Experiment - Try swapping the 'Set drive left' card with the 'Set drive right' card. What happens?

Challenge one – Avoid obstacles

In this activity students create a program to make the robot avoid obstacles. To do this they are provided with hints about what the program should tell the robot to do.

Teacher note

The sensitivity of the obstacle detector can be adjusted. Find more information here: https://meetedison.com/obstacle-detection-calibration/

Points to emphasise:

- ✓ Use the hints to create your program
- ✓ Once the basic program is working, then make it special by adding lights and sound
- ✓ Is this similar to how a robot vacuum cleaner moves around a home?

Challenge two – Clap controlled lighting

In this activity students create a program that turns the Edison robot's lights on when a clap is detected and turn off after 15 seconds. To do this they are provided with hints about what the program should tell the robot to do.

Points to emphasise:

- ✓ Use the hints to create your program
- ✓ Use more than one repeat card to get to 15 seconds
- ✓ Experiment with additional helpful features like:
    - A longer on time
    - A warning before turning off
    - Clap to turn off

Lesson four – Make a function

In lesson four students learn about the programming concept of functions. A function is a section of computer code that can be reused multiple times. This allows a program to use less space in the computer's memory and saves the programmer from re-writing the same code over and over again.

For Edison card coding the function card is called 'My function'. The 'My function' card is like a customisable coding card that can be used in a program to do, well, whatever you'd like it to do!

The things that the 'My function' card does are programmed between the 'Start my function' card and the 'End my function card'.

Program one – My function

In this activity students create a program that uses a function to drive the robot in the shape of a square. This is similar to Program one in Lesson two, but this time the robot only drives for the four sides of the square, not continuously around and around.

Points to emphasise:

- ✓ The 'My function' card is like your own customisable programming card. It can do what you want it to do
- ✓ The 'Start my function' card requires three presses of the round button to scan
- ✓ Experiment – Add additional coding cards and see how they affect the program

Challenge one – Clap and drive crazy

In this activity students create a program that responds to a clap and then drives in a crazy way with beeping and lights flashing. The craziness should last 10 seconds.

This activity is completed with only the brief description. No hints are provided.

To correctly complete this activity students will need to use the following card types:

- Wait until clap
- Repeat
- My function

Points to emphasise:

- Before laying out the cards students should think about and plan their program
- Writing down the steps the program must go through is a good way to plan a program
- Once the program is working, experiment with adding more craziness

Challenge two – Cookie alarm

In this activity students create a program to guard cookies. If there is an attempt at stealing the cookies the obstacle detector will detect the cookie thief's hand and raise the alarm. The alarm must sound for at least 30 seconds.

Points to emphasise:

- Before laying out the cards students should think about and plan their program
- Writing down the steps the program must go through is a good way to plan a program
- Experiment with the three different sound cards to create a distinctive alarm sound

Lesson five – Project

In lesson five students solidify what they have learned by developing their own robotics project that solves a real-world problem.

To complete their project students are encouraged to use additional building materials, such as LEGO bricks or cardboard. This helps to transform the robot into a machine that can perform more helpful tasks.

To complete their project, students work through the following stages:

Brainstorming

Students use a technique called brainstorming to come up with ideas for their project.

Brainstorming is a process that can be done alone or in a group. The process of brainstorming is to come up with as many ideas as possible. Each idea is written down and no idea is criticised at this point. Once a lot of ideas have been formed the ideas can be assessed and a winning idea can be selected.

Students are provided with some brainstorming starting points; however, they may have ideas outside of these points and that is okay.

Documenting the problem

Students document the problem that they want to solve. They should be encouraged to write about who has this problem and why this is a problem for this person.

Documenting the problem at this stage of the process helps students think through the problem more clearly. It also gives them a record of the problem that they can return to as they develop their solution.

Document the solution

Students document the solution that they plan to develop. They should be encouraged to write about how their solution will solve the problem.

Documenting the solution at this stage of the process helps students think through the solution more clearly.

It is very likely that the students will encounter problems with the development of their solution. Returning to their recorded original solution will show them how their learning through experimentation changed the way the problem was solved.

Draw the solution

Students draw their solution that they plan to develop. Drawing the solution helps students think through the solution more clearly.

Documenting what went wrong

Students document what went wrong when they were experimenting and developing their solution.

Documenting what went wrong helps students to recognise the difference between what they previously thought was possible and what they now know as possible is what they have learned.

This form of failure is great as it results in learning and should be encouraged. If every experiment worked the way you expected, then they wouldn't be experiments!

Class presentation

Students present their robotic invention to the class including a working demonstration.

The presentation should include:

1. Who the invention helps

This is the type of person and the situation they find themselves in and what they are trying to get done, but can't.

Example: Elderly people that can't easily get out of bed at night and want to turn the light on.

2. Why the features are helpful

These are the parts of the student's solution that helps the person do what they are trying to get done.

Example: The elderly person can wave their hand above their bedhead and the robot detects their hand and turns on the light for two minutes.

3. What went wrong

These are the results from the students experiments that didn't turn out as they expected. Each time this occurred the students will have learned that they needed to change their design.

Example: When we were testing in the dark, we kept on knocking the robot off the bedhead and the robot fell down behind the bed and it couldn't light up the room. To fix this we took off the wheels and made a solid stand out of LEGO bricks that wouldn't fall down if it was knocked.

Student project assessment rubric

	Exceeding	Meeting	Approaching	Beginning
Problem: Relevance and real-world nature of the problem being solved	The discovery and understanding of the problem came about through effective brainstorming and user interviews. The problem and the user were described in detail.	Effectively brainstormed a problem and described the problem and user.	Brainstormed a potential problem.	Encountered problems with brainstorming and did not discover or describe a problem. Teacher assisted to discover a problem.
Solution: How well the solution met the needs of the user's problem with its features	The features of the solution meet the needs of the user and their problem. The solution is described in detail and the illustration shows how the solution works including annotations.	The features of the solution meet the needs of the user and their problem, are well described and illustrated.	The features of the solution somewhat meet the needs of the user. The description is very general and the illustration is missing some of the solution details.	The features of the solution did not address the user's problem well, the description is short and illustration has very little detail.
Program: How well the program worked and made use of the coding cards to meet the needs of the problem	The program was successful at solving the problem and utilised repeat, wait until and my function cards to optimise the program.	The program was successful at solving the problem and used at least one of the following card types: repeat, wait until, my function.	The program was successful at solving the problem, but did not use any of the following card types: repeat, wait until, my function.	The program did not work as intended and was not successful at solving the user's problem.
Engineering: How well the solution was built to meet the needs of the problem	The engineered solution was successful at solving the user's problem and effectively used an appropriate building material. The structure was strong and met the needs of the user.	The engineered solution was successful at solving the user's problem.	The engineered solution was successful at solving the user's problem; however, the structure is not strong and has a tendency to break during use.	The engineered solution did not work as intended and was not successful at solving the user's problem.
Presentation: How well the student covered the presentation topics	The presentation: 1. Defined the intended user and the problem they have and why this is an important problem to solve 2. Detailed the features that solve the user's problem and why these features are important 3. Reflected on what went wrong during programming and building and what was done to overcome the issues 4. Suggested what could be done to further improve the robot invention	The presentation: 1. Defined the intended user and the problem they have 2. Detailed the features that solve the user's problem 3. Reflected on what went wrong during programming and building	The presentation: 1. Defined the problem 2. Detailed a feature of the solution 3. Reflected on something that went wrong	The presentation: 1. Defined the problem 2. Detailed a feature of the solution