VEX IQ Field Engineer · 2nd generation

Build a rover. Read the world. Engineer a mission.

A hands-on course for your VEX IQ Education Kit (2nd generation, 228-8899). Work through short investigations with the real Brain, four motors and every sensor in your kit. Start with the first unit, keep an evidence notebook, and grow from guided builds into an invention of your own.

36 lessons · Education Kit 228-8899 · VEXcode IQ Blocks

Contents

Kit & setup

VEX IQ Education Kit (2nd generation) · 228-8899

What you already have

- 1 second-generation Brain with built-in Inertial Sensor
- 4 Smart Motors: two wheels, one arm, one claw
- 1 Controller and 1 Robot Battery
- 1 Touch LED and 1 Bumper Switch
- 1 second-generation Distance Sensor and 1 Optical Sensor
- Smart Cables and the included USB A–C data cable

All robot builds use this one kit, rebuilt between activities. Household field materials are additional. No competition upgrade, field tiles, Vision Sensor, pneumatics, extra motors, paid course, or AI account is required.

VEX · Your Education Kit (228-8899) and parts inventory

Gather these household materials

- Mac with Chrome and VEXcode IQ; a USB adapter if your Mac needs one
- Clear floor space, about 1.5 × 1.5 m, away from stairs and feet
- Removable tape, ruler, paper, pencil, cardboard and colored paper
- One light, empty paper supply parcel that fits the claw

1. Identify your kit

Use VEX IQ Education Kit 228-8899, 2nd generation. Check its inventory and set up a parts tray. Take turns as builder and checker. Keep the pin tool handy.

2. Adult setup session

Allow 30–45 minutes before the first download: charge the battery and Controller, install any needed Mac USB adapter, open VEXcode IQ in Chrome, and follow the official firmware/connection guides. Do not unplug during a firmware update.

3. Build in short sessions

Allow roughly 60–120 minutes for a first BaseBot and another 60–120 minutes for the Clawbot extension, split as needed. These are planning estimates, separate from the 25-minute lessons. Use the official 2nd generation guides, not the older Super Kit Clawbot guide.

4. Create a device map

With power off, trace each cable. Configure the project to those exact Smart Ports; no two devices share a port. Use Drivetrain for the two wheel motors, ArmMotor and ClawMotor for the other two, and TouchLED1, Bumper1, Distance1 and Optical1 for added sensors. The Brain Inertial Sensor uses no Smart Port.

5. Practice stopping before moving

First use a screen-only project to identify VEXcode's Stop control and the running-project stop control on your own Brain with an adult. Practice starting and ending a project locally. Do not begin an untethered driving lesson until both learner and adult can stop the Brain immediately.

6. Prepare the testing space

Drive only on a flat clear floor, away from stairs, edges, people and pets. Download while stationary, then disconnect the USB cable before motion. Recheck the whole swept area, not just a center line. Keep the robot still during startup calibration.

7. Check moving mechanisms

Power off before changing parts, cabling or loading a parcel. Adult supervision is required for first arm/claw calibration, powered mechanism changes and recovery from a jam. Keep fingers clear; use only a light empty paper parcel. Do not hold a stalled motor or increase torque to overcome binding.

8. Keep the work

Save a configured master .iqblocks project, then a separate copy for each lesson. Save before closing VEXcode. Use this academy's notebook for predictions and results; download it regularly. Browser notes and VEXcode project files are separate.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Build your first 2nd generation BaseBot
VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Connect a Brain to web VEXcode IQ on a Mac
VEX · Update 2nd generation Brain firmware
VEX · Open and save projects in Chrome
VEX · Download and run a Blocks project
VEX · Configure a 2-motor drivetrain (2nd gen)
VEX · Pair a 2nd generation Controller and Brain

Your actual device map

Record physical port numbers and update them whenever the build changes. The names below are course conventions, not factory port assignments.

Device worksheet

Device name	What to record
Drivetrain	Left/right physical ports; wheel travel; gear ratio; reversals; Brain Inertial
ArmMotor	Physical port; forward=up; verified low lift angle
ClawMotor	Physical port; forward=close; verified gentle closing angle
TouchLED1 / Bumper1	Separate actual ports; mount location; stationary press check
Distance1	Actual port; second generation; forward sight line; reference gap
Optical1	Actual port; mounting height/orientation; light power; raw dark/light readings

Build library

▼ Brain bench

Brain, charged battery and USB cable; motors unplugged

Keep the Brain flat. Start with no Controller or drivetrain configured. Add only the device named in the lesson.

VEX · Connect a Brain to web VEXcode IQ on a Mac
VEX · Open and save projects in Chrome

▼ BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot
VEX · Configure a 2-motor drivetrain (2nd gen)

▼ BaseBot + Touch LED

BaseBot and the kit's Touch LED

With power off, fix the Touch LED on top where it is easy to reach and route its cable clear of wheels. Add Touch LED in Devices, use its actual free Smart Port, and name it TouchLED1. Keep fingers away from moving parts.

VEX · Build your first 2nd generation BaseBot
VEX · Touch LED blocks

▼ BaseBot + Bumper Switch

BaseBot, Bumper Switch and Touch LED

Mount the Bumper Switch at the front so a broad cardboard wall can press and release it without trapping it. With power off, connect it to a free port. Configure Bumper1 and TouchLED1 at their actual ports. Verify pressing by hand while stationary before any drive test.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Bumper Switch blocks
VEX · Touch LED blocks

▼ BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Distance Sensor blocks (use the 2nd gen section)

▼ Optical investigation bench

Brain, Optical Sensor and Touch LED; wheel motors unplugged

Connect and configure Optical1 and TouchLED1 at different free ports. Mount the Optical Sensor rigidly facing the sample, about 10–20 mm away as a starting point; adjust for reliable readings. Keep distance, angle and lighting consistent. This is the kit's Optical Sensor, not the older Color Sensor.

VEX · Optical Sensor blocks
VEX · Touch LED blocks

▼ Clawbot

Official 2nd generation Clawbot; all four kit motors

Complete the official Clawbot build in a separate build session. Configure its drivetrain and two individual motors named ArmMotor and ClawMotor at the physical ports. Adult checks gear engagement, free travel and motor direction. Never assign these same motors a second time. Calibrate small, empty movements in lesson 3.3 before handling a parcel.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Motor block reference

▼ Clawbot expedition

Clawbot, forward Distance Sensor and accessible Touch LED

Use the verified Clawbot configuration from 3.3, adding Distance1 facing forward and TouchLED1 on top. Keep the arm low; the parcel must not block the sensor. No Controller is configured. Recheck distances from the sensor face, not from the front of the wheels.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Distance Sensor blocks (use the 2nd gen section)
VEX · Touch LED blocks

▼ BaseBot line explorer

BaseBot and Optical Sensor facing down

Rebuild as BaseBot or remove the claw attachment with power off. Mount Optical1 rigidly in front of the wheels, facing down 10–20 mm above a flat paper field, then adjust for clear dark/light readings. Secure a wide dark tape line to pale paper on the floor. Do not use floor edges as a line.

VEX · Build your first 2nd generation BaseBot
VEX · Optical Sensor blocks

▼ Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

▼ BaseBot + Controller

BaseBot, charged 2nd generation Controller and Brain

Pair the Controller and Brain using the linked 2nd generation guide. Save a separate driver project. Add Controller in Devices and choose the lesson’s joystick mode. Leave arm/claw buttons unassigned. Remove Controller from autonomous project copies so two command sources cannot compete.

VEX · Pair a 2nd generation Controller and Brain
VEX · Configure Controller drive modes

The kit detective

Identify the kit's inputs, outputs and structure, then make a stable frame.

A field team has sent you a box of parts. Before a rover can help anyone, its engineer needs to know what each part can do.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

2. Understand the idea

A robot combines a structure, a power source, inputs, a program and outputs. A sensor reports a clue; a motor produces movement. The Brain follows the program. It does not know what your mission means.

A pin joins parts; a shaft carries rotation. Two spaced joints usually resist twisting better than one pivot. Building well is part of programming well: a loose sensor can make good code look wrong.

input	Information entering a system.
output	An action or signal produced by the system.
structure	The parts that hold the robot together.

Predict before you try

Can a Touch LED belong in two groups?

▼ Compare your prediction

Yes. Its touch reading is an input; its colored light is an output.

3. Build, then investigate

1. Use the linked kit inventory to find the Brain, battery, four motors, Controller and four separate sensors. Put any missing-part question in your notebook.
2. Sort five example parts into structure, power, input and output. Put the Touch LED in both input and output and explain why.
3. Join two beams with one pin and gently feel the pivot. Add a second spaced pin where holes align. Compare resistance to twisting; do not bend the beams.
4. Draw a rover idea. Label one job, one sensor and one moving part. Photograph or sketch the sorted tray before packing up.

Your engineering plan

```
INPUT: Bumper Switch is pressed
PROGRAM: decide that contact happened
OUTPUT: show a stop message
STRUCTURE: hold the bumper in front of the chassis
```

▼ What to look for

You can explain each electronic part and show a frame joint that changes from a pivot to a more rigid connection.

▼ Stuck? Follow this clue

If two holes do not align, check beam length and pin type against the parts poster. Remove a pin with the pin tool; forcing a joint rarely fixes the geometry.

ENGINEERING INVESTIGATION

Does a second connection reduce twisting?

Change
One pin, then two spaced pins.

Measure
Describe the amount of twist with the same gentle hand force.

Keep the same
Use the same two beams and do not force the plastic.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Design a small freestanding sensor sign from beams and pins.

Success checks

- It stands for 10 seconds on a flat surface.
- You can point to the feature that prevents tipping.
- A sketch labels input, program and output.

▼ One worked approach · try yours first

Use a wide base and two spaced connections under the sign. Keep the top light. Explain that the sign is a model, not an operating sensor.

▼ Go further

Use fewer parts without making the sign less stable.

5. Explain your evidence

How could a weak frame affect a sensor reading?

▼ Compare your explanation

It could let the sensor move or change angle, so the sensor measures a different part of the scene.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Your Education Kit (228-8899) and parts inventory

Build a rover that rolls straight

Inspect a completed BaseBot and record its actual wiring and wheel configuration.

Your rover needs its first roadworthiness inspection. A robot that drags a wheel cannot deliver supplies reliably.

1. Get ready

BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot
VEX · Configure a 2-motor drivetrain (2nd gen)

Bring this learning: 1.1 · The kit detective

2. Understand the idea

Follow the official 2nd generation BaseBot guide, one numbered step at a time. Building may take several sessions; this lesson is the 25-minute inspection after that build. Count holes from the same end as the picture.

The two wheel motors are a drivetrain. Their mounting is mirrored, so matching software directions matters. A cable port is an address: the program must refer to the place where that device is really plugged in.

drivetrain	The motors and wheels that move the chassis.
port	A numbered socket connecting a device to the Brain.
friction	Resistance when surfaces move against one another.

Predict before you try

Will a different port number automatically tell your project where the motor moved?

▼ Compare your prediction

No. The physical connection and the project's Devices settings must match.

3. Build, then investigate

1. Build with the official guide and an adult. Pause at a step boundary whenever concentration fades. Keep the battery disconnected while adjusting parts.
2. Trace each wheel cable to the Brain. Record left wheel port, right wheel port, wheel type and gear ratio in the setup notebook. Left means the robot's left when facing forward.
3. Check that tires are seated, shafts are supported and wheels do not rub beams or collars. Gently roll the unpowered robot along a short tape line.
4. Open VEXcode IQ and configure the 2-motor drivetrain as described under Robot setup. Save the empty configured project as BaseBot-master.iqblocks. Keep a copy before adding lesson code.

Your engineering plan

```
BUILD → INSPECT → RECORD PORTS → CONFIGURE → SAVE
Left wheel port: record your actual number
Right wheel port: record your actual number
Wheel travel / gear ratio: match your build
Inertial sensor: Brain (built in)
```

▼ What to look for

The rover rolls without a snag, cables clear the wheels, and your saved master project matches the physical robot.

▼ Stuck? Follow this clue

If Drivetrain blocks are missing, add the drivetrain in Devices. If the robot spins during a later forward test, stop and check motor sides/reversal before changing the program distance.

ENGINEERING INVESTIGATION

What makes the unpowered rover veer?

Change

Check one rubbing wheel or mis-seated tire, then correct it.

Measure

Compare a gentle 300 mm roll before and after the correction.

Keep the same

Use the same flat surface and starting direction.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Give another engineer a five-point inspection they can follow.

Success checks

- It includes cables, wheels, fasteners, port mapping and clear space.
- The partner can find both wheel ports without guessing.
- You save the exact build configuration.

▼ One worked approach · try yours first

Number your checks: seated pins; freely moving wheels; clear cables; correct port map; flat, clear floor. Record the build guide name and any deliberate change.

▼ Go further

Explain why changing a wheel may require a configuration change.

5. Explain your evidence

Why inspect the mechanism before debugging a long program?

▼ Compare your explanation

A rubbing wheel or incorrect connection can cause errors even when the program is logically correct.

Notebook: project / prediction / change / raw result / explanation / next test.

[VEX · Build your first 2nd generation BaseBot]

[VEX · Configure a 2-motor drivetrain (2nd gen)]

Hello, field station!

Save, download and run a real VEXcode IQ Blocks project on the second-generation Brain.

The field station needs proof that your computer can send a message to the rover. Your first test uses the screen, so nothing needs to move.

---

1. Get ready

Brain bench

Brain, charged battery and USB cable; motors unplugged

Keep the Brain flat. Start with no Controller or drivetrain configured. Add only the device named in the lesson.

VEX · Connect a Brain to web VEXcode IQ on a Mac
VEX · Open and save projects in Chrome

Bring this learning: 1.2 · Build a rover that rolls straight

---

2. Understand the idea

The academy explains the lesson; VEXcode IQ is the separate application that programs your Brain. A saved .iqblocks file preserves your editable work on the Mac. Download sends a compiled project to a selected Brain slot.

Blocks below are a reading plan. Recreate each block in VEXcode IQ: rounded values are number or text inputs, square brackets indicate menu choices, and indentation places blocks inside a loop or condition. The word end marks the bottom of a C-shaped block; it is not another block to drag.

download	Send a program to the robot Brain.
project	Your blocks together with the device configuration.
sequence	Instructions executed in a particular order.

Predict before you try

Does editing the message on the Mac immediately change the program already stored on the Brain?

▼ **Compare your prediction**

No. Download the new version, then run the selected project again.

---

3. Build, then investigate

1. Have an adult charge the battery and check firmware using the setup guide. Open codeiq.vex.com in Chrome, create a New Blocks Project and choose 2nd gen in Devices. Leave motors disconnected.
2. Use the kit's USB data cable and any needed Mac adapter. Connect through the Brain menu; select the IQ Brain serial device in Chrome. Follow the Mac guide if multiple ports appear. Confirm the Brain icon turns green.
3. Drag the Screen blocks below under Events → when started. Type the message, name the project 1-3-field-station and save an .iqblocks copy on the Mac.
4. Choose a Brain slot, Download and wait for completion. Run, read the Brain, then use Stop in VEXcode. Change the message, download again, and check which version appears. Practice the Brain's own stop control with an adult before later untethered driving.

Build this stack in VEXcode IQ

```
when started
clear screen
set cursor to row (1) column (1) on screen
print [Field team ready] on screen
```

▼ **What to look for**

The actual Brain screen shows your short message. No wheels or mechanisms move.

▼ **Stuck? Follow this clue**

Gray connection icon: check power, a data-capable cable and Chrome permission. Green icon but old message: check the selected slot and download again. Do not place a USB-tethered moving robot on the edge of a desk.

ENGINEERING INVESTIGATION

Which version is running?

Change

Change ready to hello, first without a download and then with one.

Measure

Record the screen message after each run.

Keep the same

Use the same Brain slot.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Create a welcome screen containing your rover's invented name and team number on separate rows.

Success checks

- Both lines fit the screen.
- The saved file reopens in VEXcode.
- You demonstrate how to stop the project.

▼ One worked approach · try yours first

Print a short rover name at row 1, column 1. Set the cursor to row 2, column 1, then print a team number. Save, download, and rerun.

▼ Go further

Create a third row describing the rover's job in two words.

5. Explain your evidence

How are Save, Download and Run different?

▼ Compare your explanation

Save keeps editable work; Download transfers the program to the Brain; Run starts the transferred program.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Connect a Brain to web VEXcode IQ on a Mac

VEX · Open and save projects in Chrome

VEX · Download and run a Blocks project

VEX · Brain screen blocks

UNIT 2 / Program the field station · LESSON 2.1 · 25 MIN

Send a secret light message

Build and trace a timed sequence that gives a useful signal.

The rover is too far away to read a tiny screen. Design a light message that a teammate can recognize.

1. Get ready

BaseBot + Touch LED

BaseBot and the kit's Touch LED

With power off, fix the Touch LED on top where it is easy to reach and route its cable clear of wheels. Add Touch LED in Devices, use its actual free Smart Port, and name it TouchLED1. Keep fingers away from moving parts.

VEX · Build your first 2nd generation BaseBot
VEX · Touch LED blocks

Bring this learning: 1.3 · Hello, field station!

2. Understand the idea

A sequence has a beginning and an order. A wait gives a person time to see a signal. Without it, the program may change color much faster than your eyes can separate the changes.

A useful signal has an agreed meaning. Also print a short word so the message does not depend on color alone. Here all actions are screen or light outputs; the wheels stay still.

timing	When an action starts and how long it lasts.
trace	Follow a program step by step on paper.

Predict before you try

Which part takes roughly four seconds?

▼ Compare your prediction

The two waits together. The other instructions add a small amount of processing time.

3. Build, then investigate

1. Copy the configured BaseBot master and add TouchLED1. Leave wheel motors unplugged for this bench test. Use the Robot setup instructions to verify the sensor port.
2. Build the stack. Before running, draw a timeline from 0 to 4 seconds with the two colors and final off state.
3. Run and ask a teammate to say the message aloud without seeing the blocks.
4. Change just one wait value, download, and compare the timing with your prediction. Save as 2-1-signal.

Build this stack in VEXcode IQ

```
when started
clear screen
print [Ready] on screen
set [TouchLED1] color to [blue]
wait (2) seconds
clear screen
print [Checking] on screen
set [TouchLED1] color to [yellow]
wait (2) seconds
set [TouchLED1] color to [none]
```

▼ What to look for

Blue/Ready appears, then yellow/Checking, then the light turns off.

▼ Stuck? Follow this clue

If the last text is the only thing visible, check the waits between messages. If text overlaps, clear the screen before printing the next message.

ENGINEERING INVESTIGATION

What makes the message easiest to read?

Change
Try waits of 0.5, 1 and 2 seconds.

Measure
Have a teammate identify the two states correctly in three trials.

Keep the same
Keep colors, words and distance from the robot unchanged.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	____________	____________	____________

Trial	Condition / change	Prediction	Observed / units
2	_______________	_______________	_______________
3	_______________	_______________	_______________

4. Take the challenge

Create a three-state ready / working / finished signal.

Success checks

- Each state has a word and a light.
- A reader can identify the order.
- The signal ends in a known state.

▼ One worked approach · try yours first

Use three groups of clear screen, print, set color, and wait. End with Finished on screen and green light; do not add a drive command.

▼ Go further

Ask a partner to decode your signal without using the colors.

5. Explain your evidence

Why is the wait part of communication?

▼ Compare your explanation

It allows the receiver time to notice and interpret the output.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Brain screen blocks
VEX · Touch LED blocks
VEX · Control blocks

Give your rover a memory

Create a variable and distinguish its name from its changing value.

The station needs to count supply parcels. A label called parcels can hold the number that changes as work happens.

1. Get ready

Brain bench

Brain, charged battery and USB cable; motors unplugged

Keep the Brain flat. Start with no Controller or drivetrain configured. Add only the device named in the lesson.

VEX · Connect a Brain to web VEXcode IQ on a Mac

VEX · Open and save projects in Chrome

Bring this learning: 2.1 · Send a secret light message

2. Understand the idea

A variable is a named value the program can update. Start with a known value every time you run a test. Otherwise you may accidentally reason from a value that your program has not established.

Printing the word parcels displays letters. Dragging the oval parcels reporter into print displays the current number. A variable can describe a model; it does not prove a physical parcel was delivered.

variable	A named value a program can read and change.
initialize	Give a variable its starting value.

Predict before you try

What does a second run print?

▼ **Compare your prediction**

2 again, because the first block resets parcels to zero.

3. Build, then investigate

1. Start a Brain-only project. In Variables choose Make a Variable and name it parcels.
2. Recreate the stack. Drag the oval parcels reporter into the print input; do not type its name as plain text.
3. Write the value after each set/change block before downloading.
4. Run twice. Then change the second addition from 1 to 2 and predict again.

Build this stack in VEXcode IQ

```
when started
set [parcels] to (0)
change [parcels] by (1)
change [parcels] by (1)
clear screen
print (parcels) on screen
```

▼ What to look for

The Brain shows 2. It is a counter demonstration, not a sensor-based delivery record.

▼ Stuck? Follow this clue

If the screen says parcels, replace that text with the oval reporter. If values look joined, clear the screen and set the cursor before printing again.

ENGINEERING INVESTIGATION

How does changing an addition affect the final value?

Change
Replace the last +1 with +2, then +3.

Measure
Record a predicted and observed final number for each version.

Keep the same
Keep the initial value at zero.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Model a store with five parcels, then remove two and add one.

Success checks

- The final value is 4.
- The screen prints the variable, not its name.
- You explain why this is a model.

▼ One worked approach · try yours first

Set parcels to 5, change parcels by -2, change parcels by 1, then print the oval parcels reporter. No real sensor counted these parcels.

▼ Go further

Find two different sequences that end with the same value.

5. Explain your evidence

What is the difference between a variable name and its value?

▼ Compare your explanation

The name identifies the stored information; the value is the information currently stored there.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Brain screen blocks

Build a launch console

Combine a variable, ordered output and timing into a small project you can test.

Your station needs a launch countdown. For now, launch means a light message: a countdown is never permission to drive into an uncleared field.

1. Get ready

BaseBot + Touch LED

BaseBot and the kit's Touch LED

With power off, fix the Touch LED on top where it is easy to reach and route its cable clear of wheels. Add Touch LED in Devices, use its actual free Smart Port, and name it TouchLED1. Keep fingers away from moving parts.

VEX · Build your first 2nd generation BaseBot
VEX · Touch LED blocks

Bring this learning: 2.2 · Give your rover a memory

2. Understand the idea

Decomposition means splitting a job into parts: initialize, show a number, pause, update, and announce the end. Make each part work before joining it.

A requirement is something you can check. "Looks exciting" is hard to measure. "Shows 3, 2, 1 in order with a one-second pause" gives an engineer a clear test.

requirement	A condition a design must satisfy.
decomposition	Break a larger job into smaller jobs.

Predict before you try

Why do we subtract only twice for the numbers 3, 2, 1?

▼ **Compare your prediction**

The first number is the starting value. Two changes create the next two numbers.

3. Build, then investigate

1. Configure TouchLED1 and create a variable called count. Keep wheel motors unplugged.
2. Assemble the three countdown stages. Use set cursor after clear screen so every number begins in the same place.
3. Predict the values at 0, 1 and 2 seconds, then run and compare.
4. Ask a partner to test the requirements without being told which blocks you used. Save a working copy before your extension.

Build this stack in VEXcode IQ

```
when started
set [count] to (3)
clear screen
print (count) on screen
wait (1) seconds
change [count] by (-1)
clear screen
print (count) on screen
wait (1) seconds
change [count] by (-1)
clear screen
print (count) on screen
wait (1) seconds
clear screen
print [Ready to test] on screen
set [TouchLED1] color to [green]
```

▼ What to look for

3, 2, 1 appear one at a time, followed by Ready to test and a green light.

▼ Stuck? Follow this clue

If a number disappears too quickly, inspect the order of clear, print and wait. If the final value is wrong, write a trace table with one row per subtraction.

ENGINEERING INVESTIGATION

Can a reader distinguish every stage?

Change
Change only the pause duration.

Measure
Record which numbers a partner sees in order.

Keep the same
Keep the display position and start value fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make a five-second inspection countdown ending with Check field.

Success checks

- Numbers are in descending order.
- No motor command is included.
- A partner verifies the final message and saved project.

▼ One worked approach · try yours first

Start at 5. Repeat the show/wait/subtract pattern for 5, 4, 3, 2, 1, then clear and print Check field. In Unit 6 you will replace copied stages with a loop.

▼ Go further

Draw the repeated pattern you would like to turn into one reusable instruction.

5. Explain your evidence

Name one requirement and the evidence that would satisfy it.

▼ Compare your explanation

Example: numbers descend by one; a recorded trace and an observed 3 → 2 → 1 run agree.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Brain screen blocks
VEX · Touch LED blocks

The 200 mm delivery

Drive a measured distance and compare the command with three real outcomes.

A supply parcel needs to move to the next station. First prove your empty rover can reach a nearby mark.

1. Get ready

BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot
VEX · Configure a 2-motor drivetrain (2nd gen)

Bring this learning: 2.3 · Build a launch console

2. Understand the idea

A distance command requests a journey; the real result depends on the mechanism, surface and settings. Compare evidence in the same units: 200 mm is 20 cm.

The drive-for block normally waits for its movement to finish. Leave “and don’t wait” off. A drive timeout limits a stalled movement, but reaching a timeout is not reaching the target; stop the trial if movement is incomplete.

millimeter	A length unit; ten millimeters equal one centimeter.
error	The difference between the requested and observed result.
trial	One repeat of a controlled test.

Predict before you try

Is 20% velocity a distance of 20 mm?

▼ **Compare your prediction**

No. Velocity sets how fast it moves; the drive-for distance is 200 mm.

---

3. Build, then investigate

1. Copy BaseBot-master. Reconnect wheel motors with power off and check their ports. Mark a floor start line and a 200 mm finish line using the same chassis reference point.
2. Build the stack. Download while stationary, then remove the USB cable before driving. An adult verifies the local Brain stop control; clear the floor and keep the Brain still during calibration.
3. Run at 20% drive velocity. If it spins, rubs, stalls or heads outside the field, stop and fix setup before repeating.
4. Reset the same reference point and heading for three trials. Record actual travel and signed error: actual minus 200 mm.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
drive [forward] for (200) mm
stop driving
```

▼ **What to look for**

The rover advances about 200 mm, then stops. The three measurements may differ slightly.

▼ **Stuck? Follow this clue**

A large consistent distance error suggests wrong wheel travel/gear ratio. A changing error suggests wheel slip or an inconsistent start. Do not “fix” a wrong drivetrain configuration by blindly multiplying all distances.

ENGINEERING INVESTIGATION

How repeatable is your rover?

Change

Repeat the unchanged program three times.

Measure

Record actual distance and error in mm, then largest minus smallest distance.

Keep the same

Keep the start point, heading, surface, wheel configuration and battery condition similar.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Deliver to a 150 mm mark and return to the start.

Success checks

- Both commands use mm.
- The robot pauses at the delivery point.
- Three return errors are recorded, even if the robot misses.

▼ One worked approach · try yours first

Use the same speed and timeout setup, drive forward for 150 mm, wait 1 second, drive reverse for 150 mm, then stop driving.

▼ Go further

Compare the return error at 15% and 20% velocity without changing distance.

5. Explain your evidence

Does one successful run prove reliability?

▼ Compare your explanation

No. Repeated trials reveal variation and help you understand the conditions under which the design works.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Drivetrain block reference

VEX · Configure a 2-motor drivetrain (2nd gen)

UNIT 3 / Move with a plan • LESSON 3.2 • 25 MIN

Turn the corner

Distinguish a relative turn from a target heading and follow an L-shaped route.

The supply path bends around a marked work area. Your map must say which direction the rover faces after each segment.

---

1. Get ready

BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot

VEX · Configure a 2-motor drivetrain (2nd gen)

Bring this learning: 3.1 · The 200 mm delivery

---

2. Understand the idea

A relative turn changes direction from wherever you are facing. A heading is a target direction measured from a reference. Turning to heading 90 twice should face the same direction; turning right for 90 twice adds two turns.

Draw the chassis footprint as well as the route line. Wheels and corners need room during a turn. Keep the Brain still when a project calibrates its Inertial Sensor.

heading	Direction measured from a chosen reference.
relative	Measured from the current state rather than a fixed reference.

Predict before you try

Will the second turn to heading 90 add another quarter-turn?

▼ **Compare your prediction**

Normally no: the robot is already close to that heading. It may make a small correction.

---

3. Build, then investigate

1. Tape a 200 mm forward segment and a 150 mm right-hand segment on the floor. Leave generous turning space.
2. Point the rover along the first segment. Trace the blocks and draw arrows for the two final headings.
3. Run the stack and measure how far the final chassis reference point is from the goal.
4. Replace the second turn-to-heading with a relative right turn of 90 degrees. Predict and test the difference in place, on a clear floor.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
set drive heading to (0) degrees
drive [forward] for (200) mm
turn to heading (90) degrees
drive [forward] for (150) mm
turn to heading (90) degrees
stop driving
```

▼ **What to look for**

The rover follows an L shape. Its final turn-to-heading does not deliberately add another 90 degrees.

▼ **Stuck? Follow this clue**

Unexpected heading changes: check startup movement, Brain Inertial selection and the difference between turn for and turn to. A hand nudging the robot during calibration spoils the reference.

ENGINEERING INVESTIGATION

Do relative and absolute instructions mean the same thing?

Change

Use turn to heading 90 twice, then turn right for 90 twice.

Measure

Draw and record the final direction in each experiment.

Keep the same

Keep the same start heading and turn velocity.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Plan a three-segment route that ends facing its original direction.

Success checks

- The drawing labels lengths and headings.
- The footprint clears the work area.
- You compare the final direction with the start.

▼ One worked approach · try yours first

Drive 150 mm at heading 0; turn to 90 and drive 150 mm; turn to 0 and drive 100 mm; stop. Keep the shared low-speed setup and clear at least 600 mm around the route.

▼ Go further

Calculate total commanded path length and compare it with the straight-line displacement.

5. Explain your evidence

Why can the same heading command be useful more than once?

▼ Compare your explanation

It requests the same direction even when earlier motion has changed the robot’s orientation.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Drivetrain block reference

Meet the arm and claw

Verify motor directions and choose measured, gentle arm and claw travel limits.

A good delivery robot handles a light parcel without crushing it. Your job is to learn the mechanism before asking it to carry anything.

1. Get ready

Clawbot

Official 2nd generation Clawbot; all four kit motors

Complete the official Clawbot build in a separate build session. Configure its drivetrain and two individual motors named ArmMotor and ClawMotor at the physical ports. Adult checks gear engagement, free travel and motor direction. Never assign these same motors a second time. Calibrate small, empty movements in lesson 3.3 before handling a parcel.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Motor block reference

Bring this learning: 3.2 · Turn the corner

2. Understand the idea

Motor degrees describe rotation of the motor shaft, not automatically the angle of the arm. Gears can change both direction and the amount of movement. Do not copy another robot's full opening angle without testing your own build.

Setting a motor's position to zero only labels the current position; it does not move the mechanism to a safe home. Establish a known, gently open claw and low arm with the power off and an adult. Never drive against a hard stop to find zero.

calibration	Connect a measurement or setting to a known physical reference.
torque	A turning effect that can move a load.
limit	A boundary chosen to keep motion within a tested range.

Predict before you try

Does setting position to 0 open the claw?

▼ **Compare your prediction**

No. It relabels the current position. Mechanical position must be checked separately.

---

3. Build, then investigate

1. Build the official Clawbot in separate sessions. With an adult, check the arm and claw have free movement and configure ArmMotor / ClawMotor at the right ports.
2. Put the empty arm low, with room to lift slightly. Run the ArmMotor stack below using 10 motor degrees. Stop immediately if forward lowers into a stop; correct reversal in Devices so forward gently raises. Do not increase travel to fight a jam.
3. Duplicate the stack for ClawMotor, starting gently open with no parcel. Define forward as closing in Devices; test just 10 degrees then return. Keep the arm low and fingers clear.
4. Increase only in small, inspected increments if needed. Record an approved low lift angle and gentle close angle for your empty paper parcel. These measured values are required before the capstone; if not established, continue with an empty claw.

Build this stack in VEXcode IQ

```
when started
set [ArmMotor] velocity to (10) %
set [ArmMotor] max torque to (20) %
set [ArmMotor] timeout to (1) seconds
set [ArmMotor] stopping to [hold]
spin [ArmMotor] [forward] for (10) degrees
wait (1) seconds
spin [ArmMotor] [reverse] for (10) degrees
stop [ArmMotor]
```

▼ **What to look for**

A tiny empty movement in the intended direction and return, without binding. Results depend on your build and must be observed.

▼ **Stuck? Follow this clue**

Buzzing without movement means stop and inspect, not add more torque. A timed-out spin can leave the arm partway through travel; restore the known start safely before another trial.

ENGINEERING INVESTIGATION

What is the smallest useful claw movement?

Change

Increase the tested closing travel by a small amount within the adult-checked range.

Measure

Record travel, parcel contact and whether release succeeds.

Keep the same

Use the same empty paper parcel, starting opening and low speed; stop at first firm contact.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make a gripper passport for your particular robot.

Success checks

- Ports, reversed settings and forward directions are recorded.
- The lowest useful lift and gentle closing travel are measured.
- There is no stall, forced hard stop or hand near a moving claw.

▼ One worked approach · try yours first

Record: ArmMotor port and forward=up; ClawMotor port and forward=close; known start pose; approved motor degrees; speed/torque limits; release result. If a test was not run, write not run rather than inventing an angle.

▼ Go further

Explain how an extra gear between motor and arm could reverse the observed direction.

5. Explain your evidence

Why must the capstone use your measured angle?

▼ Compare your explanation

Mechanical travel depends on the actual build, starting position and parcel, so an arbitrary angle could jam or crush something.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · 2nd generation BaseBot and Clawbot build guides

VEX · Motor block reference

A button that means ready

Wait for a deliberate press and release before showing a ready message.

A field engineer should decide when a test begins. Your rover will acknowledge that decision with its light, without moving.

1. Get ready

BaseBot + Touch LED

BaseBot and the kit's Touch LED

With power off, fix the Touch LED on top where it is easy to reach and route its cable clear of wheels. Add Touch LED in Devices, use its actual free Smart Port, and name it TouchLED1. Keep fingers away from moving parts.

VEX · Build your first 2nd generation BaseBot
VEX · Touch LED blocks

Bring this learning: 3.3 · Meet the arm and claw

2. Understand the idea

A Boolean has two values, true and false. TouchLED1 pressed? is true while touched. It is not a record that somebody touched it earlier.

A held button can stay true through many checks. Waiting for release lets you distinguish a complete gesture from a long hold. Use not to reverse a Boolean.

Boolean	A value that is either true or false.
not	An operator that reverses true and false.

Predict before you try

When does green appear: at first touch or after release?

▼ **Compare your prediction**

After release, because both waits must finish.

3. Build, then investigate

1. Keep the rover stationary with wheel motors unplugged. Configure TouchLED1 and build the stack.
2. Find the hexagonal pressed? sensor block and fit it into wait until. Put it inside the Operators not block for the release check.
3. Start without touching, then hold a finger on the Touch LED. Observe that it has not reached the final signal until you release.
4. Run a second time while the Touch LED is already held. Explain exactly which wait is satisfied and which is still waiting.

Build this stack in VEXcode IQ

```
when started
set [TouchLED1] color to [yellow]
wait until <TouchLED1 pressed?>
wait until <not <TouchLED1 pressed?>>
set [TouchLED1] color to [green]
clear screen
print [Ready] on screen
```

▼ What to look for

The robot waits with a yellow light, then shows green/Ready after a press and release. It never drives.

▼ Stuck? Follow this clue

If the program waits forever, verify the port, device name and press reading while stationary. Stop the project manually; there is no moving motor to leave running.

ENGINEERING INVESTIGATION

Does a long hold create a different gesture?

Change
Try a quick tap and a three-second hold.

Measure
Record when the light changes relative to release.

Keep the same
Keep the same stack and only vary the touch duration.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Require two separate press-and-release gestures.

Success checks

- A single held touch is insufficient.
- The first gesture shows a waiting message.
- Only the second complete gesture shows Ready.

▼ One worked approach · try yours first

Repeat the pair of waits twice in sequence. Between them print One more. End with Ready. No motion blocks are needed.

▼ Go further

Draw the true/false trace of a long hold followed by a second tap.

5. Explain your evidence

Why is a press-and-release different from a single pressed check?

▼ Compare your explanation

The first waits for a complete change of state; a single check can remain true throughout a held touch.

Notebook: project / prediction / change / raw result / explanation / next test.

[VEX · Touch LED blocks]
[VEX · Control blocks]

Dock at the soft wall

Stop a short, slow approach on contact or after a fixed time limit.

The rover needs to recognize a docking wall. The first test uses a broad, light cardboard barrier and a very short route.

1. Get ready

BaseBot + Bumper Switch

BaseBot, Bumper Switch and Touch LED

Mount the Bumper Switch at the front so a broad cardboard wall can press and release it without trapping it. With power off, connect it to a free port. Configure Bumper1 and TouchLED1 at their actual ports. Verify pressing by hand while stationary before any drive test.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Bumper Switch blocks
VEX · Touch LED blocks

Bring this learning: 4.1 · A button that means ready

2. Understand the idea

The Bumper Switch reports physical contact; it does not measure distance. Verify it changes state before allowing the rover to move.

The plain drive block continues until another command changes it. Here wait until has two ways out: contact OR a timer limit. The drive timeout setting alone does not bound a plain drive command.

or	True when at least one condition is true.
timeout	A limit that ends an attempt after too much time.

Predict before you try

What ends the attempt if the bumper never presses?

▼ **Compare your prediction**

The timer condition becomes true after roughly one second, then stop driving runs.

3. Build, then investigate

1. Mount and test Bumper1 by hand with motors unplugged. In VEXcode's Monitor, add the Bumper1 pressed? value and confirm it changes. Build the supplied timer condition with the Operators or and greater-than blocks.
2. Place the cardboard wall about 30–50 mm ahead of the bumper. Clear the floor beyond it too. Set velocity to 10%; use the one-second bound exactly for the first trial.
3. Download, unplug USB, reset the start, clear hands and run. Observe the cause of stopping; contact and elapsed time are different outcomes.
4. Try one no-wall trial on a clear floor to verify the time limit stops the robot. Restore the wall and repeat three docking trials.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (10) %
set drive stopping to [brake]
reset timer
drive [forward]
wait until <<Bumper1 pressed?> or <(timer in seconds) > (1)>>
stop driving
clear screen
print [Stopped: inspect] on screen
```

▼ **What to look for**

The rover stops on bumper contact or the short timer bound. Stopped does not mean docked; inspect and record which condition ended the run.

▼ **Stuck? Follow this clue**

If the chassis touches first, reposition the bumper rather than increasing drive time. If the timer never reaches the limit, make sure reset timer occurs once before driving, not repeatedly.

ENGINEERING INVESTIGATION

How reliably does the mount register contact?

Change

Repeat three trials, then adjust only the mount if it misses.

Measure

Record contact/timeout and whether the cardboard presses the switch freely.

Keep the same

Keep speed, start gap and wall shape fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make a docking test card with both success and failure cases.

Success checks

- A hand-pressed stationary sensor is checked first.
- Contact and timeout results are recorded separately.
- All runs end with stopped wheels.

▼ One worked approach · try yours first

Test cases: bumper released; bumper pressed before start; wall within reach; no wall. An already pressed bumper should lead to an immediate stop after the first check. Add a pre-start wait for release as an improvement.

▼ Go further

Explain why a bumper is a poor way to protect a fragile object from first contact.

5. Explain your evidence

Why does “stopped” not prove “docked”?

▼ Compare your explanation

The timer may have ended the attempt without contact, so the reason for stopping is part of the evidence.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Bumper Switch blocks

VEX · Timer blocks

VEX · Drivetrain block reference

The two-clue launch lock

Build and test a Boolean AND rule using touch and contact states.

The station should accept a ready signal only when the engineer touches the LED and the front bumper is clear.

1. Get ready

BaseBot + Bumper Switch

BaseBot, Bumper Switch and Touch LED

Mount the Bumper Switch at the front so a broad cardboard wall can press and release it without trapping it. With power off, connect it to a free port. Configure Bumper1 and TouchLED1 at their actual ports. Verify pressing by hand while stationary before any drive test.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Bumper Switch blocks
VEX · Touch LED blocks

Bring this learning: 4.2 · Dock at the soft wall

2. Understand the idea

AND needs both conditions true. OR needs at least one true. Make a four-row truth table before building a rule with two Boolean inputs.

A rule can be tested without driving. In this lesson the wheels remain unplugged, so you can explore every input combination by hand. The final message reports the rule at that instant, not a permanent guarantee about the field.

and	True only when both conditions are true.
truth table	A table listing input combinations and the result of a rule.

Predict before you try

What happens if the LED is touched while the bumper is pressed?

▼ **Compare your prediction**

The AND rule stays false because not pressed is false. It waits.

3. Build, then investigate

1. Write four cases: no touch/clear, touch/clear, no touch/pressed, touch/pressed. Predict which should be accepted.
2. Build AND with TouchLED1 pressed? and NOT Bumper1 pressed?. Keep the wheel motors unplugged.
3. For each case, stop and rerun the project. Hold the intended inputs and see whether Accepted appears.
4. Change AND to OR only in a copy and test again. Explain why that changes the requirement.

Build this stack in VEXcode IQ

```
when started
clear screen
print [Waiting] on screen
wait until <<TouchLED1 pressed?> and <not <Bumper1 pressed?>>>
clear screen
print [Accepted] on screen
set [TouchLED1] color to [green]
```

▼ **What to look for**

Only touch/clear satisfies the intended rule. No drivetrain command is present.

▼ **Stuck? Follow this clue**

A hexagonal condition must fit inside another condition, not a text input. Check the nesting: NOT applies only to the bumper state.

ENGINEERING INVESTIGATION

How many cases does AND accept compared with OR?

Change

Swap only the logical operator in a saved copy.

Measure

Record all four input combinations and the observed result.

Keep the same

Keep the two input expressions unchanged.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Explain the launch lock to a teammate using a table and plain language.

Success checks

- All four cases are tested.
- Only the intended case is accepted.
- You can point to the NOT inside the AND.

▼ One worked approach · try yours first

Table: false/false bumper → false; true/false bumper → true; false/true bumper → false; true/true bumper → false. Say: accept when touched AND the bumper is NOT pressed.

▼ Go further

Invent another real rule that needs AND and explain a case OR would wrongly accept.

5. Explain your evidence

Why test the rejected cases too?

▼ Compare your explanation

A system that accepts the right case but also accepts the wrong cases does not satisfy its requirement.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Touch LED blocks
VEX · Bumper Switch blocks
VEX · Control blocks

Measure an invisible gap

Compare the second-generation Distance Sensor with a ruler and identify missing readings.

Your rover must leave a gap in front of a station. Before trusting a threshold, investigate what this sensor sees.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)

VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 4.3 · The two-clue launch lock

2. Understand the idea

This kit’s Distance Sensor produces numeric measurements. A number does not guarantee that the intended target was detected. Always inspect found an object? alongside distance.

Measure from the sensor face. A ruler starting at the bumper creates a different reference point. Broad, flat cardboard makes an easier first target than a narrow rod or angled shiny surface.

reference point	The point from which a measurement starts.
numeric	Expressed as a number.
detection	Whether the sensor currently reports an object.

Predict before you try

If the ruler starts 50 mm behind the sensor face, will it agree exactly with the sensor?

▼ **Compare your prediction**

No. The measurements use different reference points, introducing an offset.

---

3. Build, then investigate

1. Keep motors unplugged for this bench investigation. Configure the 2nd generation Distance1, not the older distance device.
2. Add found an object? and object distance in mm reporters to VEXcode's Monitor. Place the same cardboard target at ruler distances 100, 200 and 300 mm.
3. Run the screen sample below at each position. Read the first row as detection and the second as distance; ignore a distance when detection is false.
4. Move the target away or angle it. Record missing detections honestly rather than treating them as zero distance.

Build this stack in VEXcode IQ

```
when started
clear screen
set cursor to row (1) column (1) on screen
print <Distance1 found an object?> on screen
set cursor to row (2) column (1) on screen
print (Distance1 object distance in mm) on screen
```

▼ **What to look for**

With a suitable target, detection is true and the numeric reading changes with the gap. Individual readings need not equal the ruler exactly.

▼ **Stuck? Follow this clue**

If the only reporter says distance in mm rather than object distance in mm, check that you selected the correct sensor generation. A beam in front of the sensor can become the closest target.

ENGINEERING INVESTIGATION

Does target angle affect the reading?

Change

Compare a flat target facing the sensor with the same target tilted.

Measure

Take three readings at each angle; mark undetected separately.

Keep the same

Keep target distance, size and material constant.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Choose a docking gap and explain a tolerance around it.

Success checks

- The gap is measured from the sensor face.
- Three readings support your choice.
- An undetected object is never logged as a valid zero.

▼ One worked approach · try yours first

For a 200 mm target, record three valid readings, their smallest/largest values and any misses. Propose a sensible band, such as 180–220 mm, only if your data support it.

▼ Go further

Test a smaller target and explain which target you would use for a reliable station.

5. Explain your evidence

Why record detection and distance separately?

▼ Compare your explanation

Detection tells you whether there is a reading to interpret; the distance alone can be misleading when no suitable object is found.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Distance Sensor blocks (use the 2nd gen section)

Sort a message by color

Use the actual Optical Sensor to distinguish colored samples under controlled conditions.

A supply station uses colored cards. Your first job is to discover which cards the rover can recognize consistently.

1. Get ready

Optical investigation bench

Brain, Optical Sensor and Touch LED; wheel motors unplugged

Connect and configure Optical1 and TouchLED1 at different free ports. Mount the Optical Sensor rigidly facing the sample, about 10–20 mm away as a starting point; adjust for reliable readings. Keep distance, angle and lighting consistent. This is the kit's Optical Sensor, not the older Color Sensor.

VEX · Optical Sensor blocks

VEX · Touch LED blocks

Bring this learning: 5.1 · Measure an invisible gap

2. Understand the idea

Color name and hue describe a color; brightness describes how much light is measured. A dark blue card and a dark red card can have similar brightness but different hues.

The Optical Sensor's own light helps keep illumination more consistent, but distance, angle and surrounding light still matter. A label from one reading is evidence for that setup, not a promise that every red object will work.

hue	A numeric description of color around a color wheel.
brightness	How much light a sensor measures.
sample	One object or reading used in an investigation.

Predict before you try

Must a darker sample have a different hue?

▼ **Compare your prediction**

No. Brightness and hue describe different properties of the detected light.

---

3. Build, then investigate

1. Prepare red, blue and pale paper samples. Mount Optical1 and configure TouchLED1. Leave all wheel motors unplugged.
2. Set the Optical light to 50%. Use the Monitor to inspect Optical1 found an object?, detects red? and color name as you position the paper.
3. Build the screen sample below. For each card, run again and record both color name and brightness.
4. Repeat under a different room light without changing the mount. Find a position at which the intended colors are reliably detected.

Build this stack in VEXcode IQ

```
when started
set [Optical1] light power to (50) %
wait (0.2) seconds
clear screen
print (Optical1 color name) on screen
set cursor to row (2) column (1) on screen
print (Optical1 brightness in %) on screen
```

▼ **What to look for**

Different sample cards give recorded labels and brightness values. The lesson may reveal an unreliable card that should be replaced.

▼ **Stuck? Follow this clue**

If labels flicker, fix the sample position and lighting first. This course uses Optical1; Color Sensor blocks for the older hardware are different.

ENGINEERING INVESTIGATION

Which sample is easiest to recognize?

Change

Present three card colors at the same position.

Measure

Record three color labels per card and the number that match the intended label.

Keep the same

Keep sensor distance, angle and light power fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make a two-card station code that another engineer can reproduce.

Success checks

- Each card is detected consistently in three trials.
- The mounting distance and lighting are documented.
- You include a case with no card.

▼ One worked approach · try yours first

Choose two reliably recognized cards. Record their observed color names, the mounting gap and three repeated results. For the no-card case, use found an object? to avoid claiming a meaningful card color.

▼ Go further

Try a patterned card and explain why the portion under the sensor matters.

5. Explain your evidence

Why might a correct color rule fail after moving the sensor?

▼ Compare your explanation

The sensor may now see a different area or receive different light, changing its measurements.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Optical Sensor blocks

Find the edge of the trail

Collect dark/light readings and calculate a justified threshold for a tape trail.

Your next rover will follow a trail. Before adding wheels, find a number that separates dark tape from pale paper.

1. Get ready

Optical investigation bench

Brain, Optical Sensor and Touch LED; wheel motors unplugged

Connect and configure Optical1 and TouchLED1 at different free ports. Mount the Optical Sensor rigidly facing the sample, about 10–20 mm away as a starting point; adjust for reliable readings. Keep distance, angle and lighting consistent. This is the kit's Optical Sensor, not the older Color Sensor.

VEX · Optical Sensor blocks
VEX · Touch LED blocks

Bring this learning: 5.2 · Sort a message by color

2. Understand the idea

A threshold turns a numeric measurement into a yes/no comparison. It should come from your own data. A supplied example such as 50% is a starting guess, not a universal definition of black.

Collect several readings. If the dark and light ranges overlap, no single threshold cleanly separates them. Fix the mounting or lighting and measure again. When the ranges separate, the midpoint between the darkest light reading and the brightest dark reading is a useful candidate.

threshold	A comparison boundary used to make a decision.
range	The smallest through largest values in a set.

Predict before you try

With dark readings 18, 22, 20 and light readings 72, 68, 70, what candidate separates their ranges?

▼ Compare your prediction

(22 + 68) ÷ 2 = 45. Fresh readings must still test that candidate.

---

3. Build, then investigate

1. Put a strip of dark tape on pale paper. Fix Optical1 pointing down, with the same height and 50% light power you will use later.
2. Run the sample five times above dark tape and five above pale paper. Use the notebook calibration helper to record the values.
3. Identify the highest dark value and lowest light value. If dark is reliably lower, calculate (highest dark + lowest light) ÷ 2.
4. Test the candidate on fresh samples, including the edge between the materials. Save the values, height and lighting for Unit 9.

Build this stack in VEXcode IQ

```
when started
set [Optical1] light power to (50) %
wait (0.2) seconds
clear screen
print (Optical1 brightness in %) on screen
```

▼ What to look for

A table of real dark/light readings and either a justified candidate or a clear reason to improve the setup.

▼ Stuck? Follow this clue

Overlapping ranges are a measurement problem, not a reason to hide outliers. Check that the sensor sees the tape rather than its edge and that paper is flat.

ENGINEERING INVESTIGATION

Will your threshold survive a new reading?

Change
Take three new readings on each surface after choosing the candidate.

Measure
Count correct dark/light classifications and record any ambiguous readings.

Keep the same
Keep height, sample material and light power fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Write a calibration card for your line explorer.

Success checks

- It includes raw readings, not just an average.
- It rejects overlapping ranges.
- A new trial tests the chosen threshold.

▼ One worked approach · try yours first

Example only: dark range 18–22, light 68–72 gives a candidate 45. Rule: brightness <45 means dark; at exactly 45 treat it as light for this two-way rule. Replace this example with your actual readings.

▼ Go further

Change sensor height, recalibrate, and explain why the previous threshold may no longer fit.

5. Explain your evidence

Why is a threshold more convincing when you show the readings?

▼ Compare your explanation

Other engineers can judge the separation, variation and conditions that support the decision.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Optical Sensor blocks

A patrol in four steps

Use a repeat loop and explain which instructions run once versus many times.

Your rover patrols a tiny square around a paper campsite. Can four repeated pairs replace eight separate movement blocks?

---

1. Get ready

BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot
VEX · Configure a 2-motor drivetrain (2nd gen)

Bring this learning: 5.3 · Find the edge of the trail

---

2. Understand the idea

A repeat block encloses the actions to repeat. Indentation in our plan shows what belongs inside its C shape. Setup before the loop happens once; actions after it wait until all repeats finish.

The turn at the end of each side also belongs inside the loop. Four right turns of 90 degrees add to 360 degrees. The rover may not return perfectly because errors accumulate.

loop	A group of instructions that repeats.
iteration	One trip through a loop's body.

Predict before you try

How many drive-for commands execute?

▼ **Compare your prediction**

Four, even though only one drive-for block is inside the repeat body.

3. Build, then investigate

1. Tape a 150 mm square on a clear floor with room for the full robot around it. Begin at one corner facing along an edge.
2. Drag repeat from Control and put drive for and turn for inside. Keep speed/timeout setup outside the repeat.
3. Trace the route after iterations 1, 2, 3 and 4. Then run and mark where the chassis reference point ends.
4. Repeat three complete patrols from the same start. Compare final error instead of adjusting the robot during a run.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
repeat (4)
  drive [forward] for (150) mm
  turn [right] for (90) degrees
end
stop driving
```

▼ **What to look for**

Four short sides and four turns form an approximate square, then the wheels stop.

▼ **Stuck? Follow this clue**

If it drives four times then turns once, the turn is outside the C block. Stop, inspect nesting, and paper-trace the corrected stack before running.

ENGINEERING INVESTIGATION

Does the route close consistently?

Change

Repeat the same patrol three times.

Measure

Record final distance from the start and final heading error.

Keep the same

Keep the starting pose, floor and low velocities fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Create a two-lap patrol without duplicating all eight actions.

Success checks

- You can explain inner and outer counts.
- A full lap includes four turns.
- You keep the bounded route inside the clear field.

▼ One worked approach · try yours first

Place the four-repeat square inside an outer repeat 2. Alternatively repeat the side-and-turn pair 8 times. Both give eight short sides; the nested version names the idea of a lap.

▼ Go further

Try a small triangle using three sides and 120-degree relative turns, then explain exterior angles.

5. Explain your evidence

What changes when a block moves outside the loop?

▼ Compare your explanation

It no longer executes once per iteration; it executes when the program reaches its new position.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Control blocks

VEX · Drivetrain block reference

Teach the rover a new word

Define and call a My Block with a numeric distance parameter.

Your delivery map uses the same short journey at different lengths. Give that journey a name and a number it can use.

1. Get ready

BaseBot

Official 2nd generation BaseBot; two wheel motors

Use the official BaseBot build linked below. In Devices, select 2nd gen, add a 2-motor Drivetrain, match the physical left/right ports, select the Brain Inertial Sensor, and match wheel travel and gear ratio to the build. Keep the robot still during startup calibration. Do not assign Controller joysticks for autonomous lessons.

VEX · Build your first 2nd generation BaseBot
VEX · Configure a 2-motor drivetrain (2nd gen)

Bring this learning: 6.1 · A patrol in four steps

2. Understand the idea

A My Block groups instructions under one name. A definition explains what that name does; a call asks the robot to do it now. Definitions are separate stacks, not extra commands placed under when started.

A parameter is an input to a reusable block. Use the parameter's oval reporter in the drive distance. Typing the parameter's name as text will not substitute its value.

function	A named, reusable group of instructions; called a My Block here.
parameter	An input supplied when a reusable block is called.

Predict before you try

What are the farthest intended distance and total path length?

▼ Compare your prediction

Farthest: 200 mm from start. Total path: 100 + 100 + 200 + 200 = 600 mm.

3. Build, then investigate

1. In My Blocks choose Make a Block. Name it delivery and add a number input called distance. Place the definition separately in the workspace.
2. Under define delivery, insert the parameter reporter in both forward and reverse distances. Keep default waiting behavior on both movements.
3. Under when started, configure low speeds/timeouts and call delivery with 100, then 200. Put both calls below setup.
4. Predict the longest distance from start and total commanded travel. Test on the same short clear lane.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
delivery (100)
delivery (200)
stop driving

define delivery (distance)
  drive [forward] for (distance) mm
  wait (1) seconds
  drive [reverse] for (distance) mm
```

▼ What to look for

The rover makes a 100 mm out-and-back trip, then a 200 mm out-and-back trip.

▼ Stuck? Follow this clue

If nothing happens, check that the custom block is called under when started. A definition alone does not run. If both trips have the same length, check you used the parameter reporter rather than a fixed number.

ENGINEERING INVESTIGATION

Does the same block work with a new input?

Change
Call delivery with 100, 150 and 200 in separate trials.

Measure
Record outward and return errors.

Keep the same
Keep the definition and start pose fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Add a second number input for the delivery pause.

Success checks

- One definition handles at least two distances and pauses.
- Each parameter reporter appears where its value is needed.
- The robot returns after each bounded trip.

▼ One worked approach · try yours first

Define delivery(distance, pause). Put distance into both movement blocks and pause into wait. Call delivery(100, 1) and delivery(150, 2). The notation is a guide for creating the number inputs in VEXcode.

▼ Go further

Explain why changing the definition once updates every later call.

5. Explain your evidence

How does a parameter avoid copying a whole program?

▼ Compare your explanation

The same tested instructions can receive different inputs, so only the call's values need to change.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · My Blocks and parameters
VEX · Drivetrain block reference

Search, then stop searching

Build a bounded repeat-until search that checks fresh sensor readings.

A cardboard station might be just outside the rover's view. Search through a small arc, then report whether the station was found.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 6.2 · Teach the rover a new word

2. Understand the idea

Repeat until tests its condition before each iteration. A true condition at the start can mean zero movements. Include a count limit so a missing target cannot create endless turning.

A loop should update the information that controls it. This search rereads Distance1 and changes attempts after each turn. Later decisions can distinguish success from an exhausted search.

bounded	Limited by a count, time or other stopping condition.
fresh reading	A measurement taken again rather than reusing an old value.

Predict before you try

How many turns happen if detection starts true?

▼ **Compare your prediction**

Zero. The exit condition is checked before the body runs.

---

3. Build, then investigate

1. Place a broad cardboard station slightly to the right of the starting view; keep all objects outside the robot's turning footprint. Create attempts.
2. Build repeat until with found an object OR attempts >3. Put a 10-degree right turn, count update and short wait inside.
3. Predict all four attempts on paper when no target is detected. The maximum intended turn is 40 degrees.
4. Test target visible at start, target within the arc and target absent. Read detection and attempts in the Monitor and record all outcomes.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
set [attempts] to (0)
repeat until <<Distance1 found an object?> or <(attempts) > (3)>>
  turn [right] for (10) degrees
  change [attempts] by (1)
  wait (0.1) seconds
end
stop driving
clear screen
print (attempts) on screen
```

▼ **What to look for**

The rover turns at most four small increments and stops. Attempts is 0–4; detection tells you whether it found something.

▼ **Stuck? Follow this clue**

If attempts stays zero while turning, check the change block is inside the loop. If a background wall ends every search, change the field so the intended target can be distinguished.

ENGINEERING INVESTIGATION

Can the search distinguish three situations?

Change

Start with the station visible, just to the right, or absent.

Measure

Record attempts, final detection and whether the detected object really was the station.

Keep the same

Keep the same arc, step size and sensor mount.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Add a readable result instead of showing only the attempt count.

Success checks

- Found and not-found outcomes are distinct.
- The search stays limited to four turns.
- A visible-at-start case makes no turn.

▼ One worked approach · try yours first

After the loop, inspect found an object? in an if/else block: true prints Target seen; false prints No target. Unit 7 explains these branches in detail. Stop driving before reporting.

▼ Go further

Explain a background object that could cause a false identification even when detection works correctly.

5. Explain your evidence

Why are both detection and an attempt limit useful?

▼ Compare your explanation

Detection ends useful searching; the limit ends an unsuccessful attempt. Either should leave the robot stopped.

Notebook: project / prediction / change / raw result / explanation / next test.

[VEX · Distance Sensor blocks (use the 2nd gen section)]
[VEX · Control blocks]

One question, two messages

Use if/else to select exactly one outcome from a sensor condition.

The station accepts red request cards. It should tell a learner whether the card is accepted or needs checking.

1. Get ready

Optical investigation bench

Brain, Optical Sensor and Touch LED; wheel motors unplugged

Connect and configure Optical1 and TouchLED1 at different free ports. Mount the Optical Sensor rigidly facing the sample, about 10–20 mm away as a starting point; adjust for reliable readings. Keep distance, angle and lighting consistent. This is the kit's Optical Sensor, not the older Color Sensor.

VEX · Optical Sensor blocks
VEX · Touch LED blocks

Bring this learning: 6.3 · Search, then stop searching

2. Understand the idea

An if/else chooses one branch when execution reaches it. It does not keep watching by itself. A red card shown after the branch has already run will not change that old decision.

Use words as well as lights. An else branch should have a useful meaning: not accepted includes other colors and no card, so it must not pretend to identify an alternative color.

branch	One possible path through a program.
condition	A true/false expression used to choose an action.

Predict before you try

Will a card inserted after the decision change the message?

▼ Compare your prediction

No. This stack makes one decision and then finishes.

---

3. Build, then investigate

1. Use the reliable red card and mounting from Unit 5. Keep motors unplugged. Put the card in place before running.
2. Build the if/else stack and place the two message groups in their correct branches.
3. Run once with red, once with blue and once with no card. Write the expected branch before each run.
4. Run with no card and only then insert red. Observe that the message does not automatically update.

Build this stack in VEXcode IQ

```
when started
set [Optical1] light power to (50) %
wait (0.2) seconds
clear screen
if <Optical1 detects [red]?> then
  print [Accepted] on screen
  set [TouchLED1] color to [green]
else
  print [Check card] on screen
  set [TouchLED1] color to [yellow]
end
```

▼ What to look for

A red card is accepted when detected; other or missing cards produce Check card.

▼ Stuck? Follow this clue

If both messages appear, check that one is inside then and the other inside else. A print after the entire if/else always runs.

ENGINEERING INVESTIGATION

When is the decision made?

Change

Present the same card before the run, then after it.

Measure

Record the message and whether the branch was reached before the card arrived.

Keep the same

Use the same card position and program.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Build a blue-card station with useful text for rejected cases.

Success checks

- Blue is accepted in repeated tests.
- Red and no-card cases are rejected.
- The else message does not falsely label every rejected sample red.

▼ One worked approach · try yours first

Change the detects choice to blue. Keep an else message such as Check card rather than Red card. Test blue, red and absent samples.

▼ Go further

Explain where a wait-for-touch block could let an engineer choose the decision time.

5. Explain your evidence

What does the else branch really tell you?

▼ Compare your explanation

The tested condition was false. It does not identify the reason unless you test further.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Optical Sensor blocks
VEX · Control blocks

Near, ready, or too far?

Classify a distance using nested decisions, including exact boundary values and missing data.

A docking station needs more than a red/green guess. Report Too near, Ready, Far, or No target.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)

VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 7.1 · One question, two messages

2. Understand the idea

Test whether an object exists before using its distance. Nested decisions let you ask a second question only after the first one gives enough information.

Write boundary cases explicitly. Here less than 180 is too near; greater than 220 is far; 180 through 220 inclusive is the ready band. Test 179, 180, 220 and 221 on paper, because placing a ruler to a perfect sensor millimeter is not realistic.

nested	Placed inside another block or decision.
boundary	A value where the outcome of a rule changes.

Predict before you try

What message belongs to exactly 220 mm?

▼ **Compare your prediction**

Ready, because 220 is not greater than 220 and is not less than 180.

---

3. Build, then investigate

1. Keep wheel motors unplugged and mount the target as in 5.1. Create a variable called gap.
2. Recreate the nested branches, with the no-target message in the outer else.
3. Paper-trace missing, 179, 180, 200, 220 and 221 mm. Then position the real target in clearly near, middle and far regions.
4. Record real readings alongside messages. If they fluctuate around a boundary, describe that variation instead of forcing a preferred answer.

Build this stack in VEXcode IQ

```
when started
clear screen
if <Distance1 found an object?> then
  set [gap] to (Distance1 object distance in mm)
  if <(gap) < (180)> then
    print [Too near] on screen
  else
    if <(gap) > (220)> then
      print [Far] on screen
    else
      print [Ready] on screen
    end
  end
else
  print [No target] on screen
end
```

▼ **What to look for**

One message explains the chosen region or the missing detection.

▼ **Stuck? Follow this clue**

A missing target must not fall into Ready. Inspect the outer found-an-object condition before tweaking the thresholds.

ENGINEERING INVESTIGATION

Does the decision match the actual reported distance?

Change

Use three clearly separated target positions.

Measure

Record detection, reported gap, predicted message and observed message.

Keep the same

Keep target material, mount and threshold values fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Choose a new ready band from your measured data.

Success checks

- Your lower threshold is below your upper threshold.
- Both endpoints have defined behavior.
- The no-target case still has its own branch.

▼ One worked approach · try yours first

Example: change the limits to 150 and 190; ready includes both endpoints. Test missing, 149, 150, 190 and 191 on paper, then test real targets clearly inside each region.

▼ Go further

Compare a narrow band with a wide band and explain the precision versus tolerance tradeoff.

5. Explain your evidence

Why test the values exactly on a boundary?

▼ Compare your explanation

A strict < or > treats equality differently, and a hidden gap in the logic often appears at the endpoint.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Distance Sensor blocks (use the 2nd gen section)

VEX · Control blocks

A rover that can say “I cannot tell”

Combine a sensor decision with one small, bounded movement and a conservative fallback.

A field engineer trusts a rover more when it admits missing information. Your rover may approach a visible far station, but should stop when it cannot detect one.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)

VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 7.2 · Near, ready, or too far?

2. Understand the idea

A decision policy maps each condition to an action. Missing information is its own case; do not treat an absent distance as an open route.

A short step limits how far the robot travels between observations. This program takes at most one 20 mm step. It is not obstacle avoidance for people or a guarantee about objects outside the sensor’s view.

policy	A set of rules mapping conditions to actions.
fallback	A chosen response when the usual information or action is unavailable.

Predict before you try

What does missing detection cause?

▼ Compare your prediction

No movement. The fallback reports No target: hold.

---

3. Build, then investigate

1. Keep the target at least 300 mm away for the first physical approach and confirm it is detected. Clear the rest of the floor.
2. Build the nested decision. Each movement uses drive for 20 mm; leave default waiting behavior on.
3. Paper-test found/far, found/near and missing. Then run those cases on the floor with no body parts in the path.
4. Record whether the rover stepped or held. Ask a teammate to move the station between separate runs to test the same rule.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
clear screen
if <Distance1 found an object?> then
  if <(Distance1 object distance in mm) > (220)> then
    print [One step] on screen
    drive [forward] for (20) mm
  else
    print [Hold position] on screen
  end
else
  print [No target: hold] on screen
end
stop driving
```

▼ What to look for

Exactly one short forward step when a suitable detected target is farther than 220 mm; otherwise no intentional travel.

▼ Stuck? Follow this clue

If the rover moves several steps, look for an unintended loop or Controller assignment. This lesson contains one decision and no repeating movement.

ENGINEERING INVESTIGATION

Can a partner predict the rover's behavior?

Change

Change only the target condition between trials.

Measure

Record detection, gap and whether one step happened.

Keep the same

Use the same policy and initial robot pose.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Write a test table that would expose a mistaken > / < comparison.

Success checks

- Near and far cases give different actions.
- The missing-data case holds.
- You include the exact 220 boundary as a paper test.

▼ One worked approach · try yours first

Tests: detected 300 → step; detected 180 → hold; detected 220 → hold; absent → hold. Swapping > for < would reverse the first two outcomes, so these cases expose it.

▼ Go further

Explain a thin object the sensor may miss and why a clear supervised test field still matters.

5. Explain your evidence

What makes a fallback useful?

▼ Compare your explanation

It specifies a predictable response for a case that otherwise might produce an unsafe or unexplained guess.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Distance Sensor blocks (use the 2nd gen section)

VEX · Drivetrain block reference

VEX · Control blocks

Draw a mission worth testing

Turn a supply-delivery brief into a field map, stages and measurable acceptance checks.

A remote research camp needs a light parcel delivered to its receiving zone. You are the engineer responsible for a repeatable route and an honest report.

1. Get ready

Clawbot expedition

Clawbot, forward Distance Sensor and accessible Touch LED

Use the verified Clawbot configuration from 3.3, adding Distance1 facing forward and TouchLED1 on top. Keep the arm low; the parcel must not block the sensor. No Controller is configured. Recheck distances from the sensor face, not from the front of the wheels.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Distance Sensor blocks (use the 2nd gen section)
VEX · Touch LED blocks

Bring this learning: 7.3 · A rover that can say "I cannot tell"

Your supply-station field · top view, schematic

Start	200 mm →	Receiving zone	Backboard
Claw open · arm low	Clear travel lane	Inspect actual parcel	Visible beyond route

Measure gaps from the Distance Sensor's face. Leave more than 150 mm of sensor clearance through the entire trip. Draw the robot's full footprint on your own dimensioned map.

2. Understand the idea

A capstone brings familiar pieces together. Keep the first field simple: a marked start, a clear 200 mm route and a receiving box. Add a turn only after the straight mission works.

A stage has an entry condition, an action and evidence of completion. A timeout or missed destination is a failed trial, not an excuse to call the next stage successful. Plan where an observer will stop the attempt if a stage goes wrong.

acceptance check	An observable result required for a successful trial.
constraint	A boundary your design must work within.
stage	One clearly defined part of a larger mission.

Predict before you try

Is “the program finished” enough evidence of delivery?

▼ Compare your prediction

No. The observer must verify where the parcel ended and whether the trial met every acceptance check.

3. Build, then investigate

1. On the floor, mark a start line and a receiving zone centered about 200 mm along the route. Put a broad cardboard backboard beyond the zone; choose its position so the sensor still reads more than 220 mm at the start and more than 150 mm throughout the planned trip.
2. Make a light, empty paper parcel that fits the verified claw opening. Use your gripper passport from 3.3. Keep the arm low throughout travel; no lifting is required for this mission.
3. Draw the robot footprint at start and finish. Name stages: ready gesture, inspect target, close gently, travel, release, report. Write a stopping rule for each failed stage.
4. Paper-walk the route with your teammate. Choose acceptance checks: parcel inside zone, no contact with backboard, no uncontrolled motion, and all stages explained. Record exact start pose and sensor gaps.

Your engineering plan

```
READY: engineer checks field and gives press/release
CHECK: if target is absent or too close, stop the project
GRIP: close by the measured gentle angle
TRAVEL: move in small measured steps along the clear route
RELEASE: reopen by the same tested amount
REPORT: observer records delivered / missed / stopped early
```

▼ What to look for

A dimensioned field sketch, known gripper settings and a test plan before the integrated robot run.

▼ Stuck? Follow this clue

If the sensor sees the parcel or claw instead of the backboard, change the sensor mount with power off and repeat its measurement tests. Do not hide that reading in the program.

> **ENGINEERING INVESTIGATION**
>
> **Can another engineer reconstruct your start?**
>
> **Change**
> Have a partner reset the field from your drawing.
>
> **Measure**
> Check start line, heading, parcel position and target distance against the written values.
>
> **Keep the same**
> Do not coach them until they finish the first setup attempt.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Produce a mission contract that a teammate can score.

Success checks

- It labels distances, sensor reference and clear space.
- It specifies a known gentle close angle from 3.3.
- It defines success, failure and an early stop.

▼ One worked approach · try yours first

A useful contract specifies a 200 mm route, a 150 × 150 mm receiving zone centered at the nominal release point, a broad target safely beyond it, a verified gripper angle and three unchanged trials. Score delivery by where the actual parcel lands.

▼ Go further

Offer two designs and explain which removes the most uncertain mechanical step.

5. Explain your evidence

Why does a sketch include the whole robot footprint?

▼ **Compare your explanation**

The chassis and claw occupy space even where the route's center line is clear.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · 2nd generation BaseBot and Clawbot build guides

VEX · Distance Sensor blocks (use the 2nd gen section)

Integrate the delivery rover

Assemble a complete bounded delivery stack and test its stages before adding a parcel.

Today the parts become a mission. First run empty, then add the light parcel only after the sequence behaves as predicted.

1. Get ready

Clawbot expedition

Clawbot, forward Distance Sensor and accessible Touch LED

Use the verified Clawbot configuration from 3.3, adding Distance1 facing forward and TouchLED1 on top. Keep the arm low; the parcel must not block the sensor. No Controller is configured. Recheck distances from the sensor face, not from the front of the wheels.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Distance Sensor blocks (use the 2nd gen section)
VEX · Touch LED blocks

Bring this learning: 8.1 · Draw a mission worth testing

Your supply-station field • top view, schematic

Start	200 mm →	Receiving zone	Backboard
Claw open · arm low	Clear travel lane	Inspect actual parcel	Visible beyond route

Measure gaps from the Distance Sensor's face. Leave more than 150 mm of sensor clearance through the entire trip. Draw the robot's full footprint on your own dimensioned map.

2. Understand the idea

Integration reveals interactions that separate tests can miss. A claw can block a sensor; a shifted start can change every later measurement. Keep your first combined mission small.

The plan uses a closeAngle variable initialized to 0 so it cannot assume an unverified mechanism range. Replace 0 only with the approved value in your gripper passport. At 0, you can test the whole route with an empty claw. An early stop leaves the gripper where it is; an adult helps recover with power off.

integration	Join tested parts into one working system.
recovery	Return to a known condition after an interrupted attempt.

Predict before you try

Does the green light prove the parcel is in the zone?

▼ **Compare your prediction**

No. It reports reaching the end of the stack. A person must inspect the actual delivery.

3. Build, then investigate

1. Create closeAngle in Variables and copy your Clawbot configuration with Distance1 and TouchLED1. Leave closeAngle at 0 for the empty test. Use a known open claw, arm low and no Controller.
2. Build the complete stack. The early exits use Control → stop project. Each occurs after stopping the drivetrain.
3. Run the empty mission on the field from 8.1. Check each of ten 20 mm steps and the final zone. If a movement times out, stalls or deviates, the observer stops the whole project; do not continue to release.
4. After a clean empty run, set closeAngle to your measured gentle travel. Test gripping/release stationary first. Reset the exact pose, load the empty paper parcel with power off, then run one complete trial.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
set [closeAngle] to (0)
set [ClawMotor] velocity to (10) %
set [ClawMotor] max torque to (20) %
set [ClawMotor] timeout to (1) seconds
set [ClawMotor] stopping to [hold]
set [TouchLED1] color to [yellow]
wait until <TouchLED1 pressed?>
wait until <not <TouchLED1 pressed?>>
if <not <Distance1 found an object?>> then
  stop driving
  stop project
end
if <(Distance1 object distance in mm) < (220)> then
  stop driving
  stop project
end
spin [ClawMotor] [forward] for (closeAngle) degrees
repeat (10)
  if <not <Distance1 found an object?>> then
    stop driving
    stop project
  end
  if <(Distance1 object distance in mm) < (150)> then
    stop driving
    stop project
  end
  drive [forward] for (20) mm
end
stop driving
spin [ClawMotor] [reverse] for (closeAngle) degrees
stop [ClawMotor]
set [TouchLED1] color to [green]
clear screen
print [Inspect delivery] on screen
```

▼ What to look for

A bounded 200 mm nominal trip followed by release, or an early stop when target information fails. The green light is an inspection prompt.

▼ Stuck? Follow this clue

If the parcel is not gripped, zero closeAngle may still be set intentionally. Do not guess a larger number: revisit 3.3. If a step times out, the code can continue; the observer must stop and mark that run failed.

ENGINEERING INVESTIGATION

Does integration change a previously working behavior?

Change

Compare empty-claw and light-parcel trials after verifying both configurations.

Measure

Record gripping, detection, route and release separately.

Keep the same

Keep start pose, route, sensor mount and low-speed settings fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make each early exit explain why the mission stopped.

Success checks

- Missing target and too-close cases have different messages.
- Each failure leaves the drive stopped and ends the project.
- The observer still verifies physical delivery.

▼ One worked approach · try yours first

Before each stop project, clear the screen and print either No target or Too close. Set TouchLED1 to yellow. Keep the motor stop before the message and do not add an automatic recovery drive.

▼ Go further

Factor the repeated target checks into a My Block called checkStation.

5. Explain your evidence

Which evidence comes from code, and which from observation?

▼ Compare your explanation

The light/message shows which program path was reached. An observed parcel position and unobstructed run show whether the physical mission succeeded.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Motor block reference

VEX · Drivetrain block reference

VEX · Distance Sensor blocks (use the 2nd gen section)

VEX · Control blocks

Prove it, improve it, present it

Run a repeatable acceptance test, make one evidence-based improvement and explain its limits.

The research team is ready to use your rover. They need a trustworthy report, including the trials that failed.

1. Get ready

Clawbot expedition

Clawbot, forward Distance Sensor and accessible Touch LED

Use the verified Clawbot configuration from 3.3, adding Distance1 facing forward and TouchLED1 on top. Keep the arm low; the parcel must not block the sensor. No Controller is configured. Recheck distances from the sensor face, not from the front of the wheels.

VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Distance Sensor blocks (use the 2nd gen section)
VEX · Touch LED blocks

Bring this learning: 8.2 · Integrate the delivery rover

Your supply-station field • top view, schematic

Start	200 mm →	Receiving zone	Backboard
Claw open · arm low	Clear travel lane	Inspect actual parcel	Visible beyond route

Measure gaps from the Distance Sensor's face. Leave more than 150 mm of sensor clearance through the entire trip. Draw the robot's full footprint on your own dimensioned map.

2. Understand the idea

Reliability is about repeated behavior under stated conditions. Three trials are a small classroom sample, not a safety certification. Keep all outcomes; removing a failed run makes your report less useful.

Separate a design change from a test change. Moving the finish zone after every miss improves the score without improving the rover. Freeze the field and acceptance criteria while comparing versions.

reliability	How consistently a system performs its intended task under stated conditions.
iteration	A cycle of making, testing and improving a design.

Predict before you try

If v2 succeeds three times, what can you claim?

▼ Compare your prediction

It met these checks in these three trials under these conditions. You cannot claim it always works on every field.

3. Build, then investigate

1. Save the working .iqblocks file as expedition-v1. Restore the field and use the same parcel for three trials. Do not change code between them.
2. Score each stage and record final parcel position, near misses and any early stop. Label not-run tests explicitly.
3. Choose one cause supported by evidence: loose sensor mount, inconsistent start, unsuitable parcel or code parameter. Make one change and save expedition-v2.
4. Run the same three trials for v2. Present the map, two versions, six results and a short explanation of what you still do not know.

Your engineering plan

```
TRIAL | VERSION | GRIP | ROUTE | RELEASE | RESULT | EVIDENCE
1     | v1      |      |       |         |        |
2     | v1      |      |       |         |        |
3     | v1      |      |       |         |        |
Repeat the same three cases for v2 after one justified change.
```

▼ What to look for

A small engineering report with all six outcomes, a specific improvement and a clear scope for the result.

▼ Stuck? Follow this clue

If results vary widely, improve the reset procedure before changing many parameters. Check battery, wheels, starting pose and whether the claw begins at the same opening.

ENGINEERING INVESTIGATION

Did your change improve the mission?

Change

Change one identified design feature between v1 and v2.

Measure

Compare success counts and the same stage measurements.

Keep the same

Keep the field, parcel, start procedure and scoring fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Give a two-minute design review using your evidence.

Success checks

- Show the original brief and field.
- Explain one failure and its supported cause.
- Distinguish tested conditions from untested extensions.

▼ One worked approach · try yours first

Present: our requirement; our design; one failed trial; the single change; before/after results; what we would test next. Credit help and label paper reasoning versus physical results.

▼ Go further

Test a slightly different parcel in a new, clearly labeled experiment after the original comparison is complete.

5. Explain your evidence

What makes an engineering report believable?

▼ Compare your explanation

It includes a reproducible setup, clear criteria, all results and a claim no broader than the evidence.

Notebook: project / prediction / change / raw result / explanation / next test.

CMU · Coding and Computational Thinking with VEX IQ (2nd gen)

Make a live station display

Place a decision inside a loop and explain why new observations can change the output.

The station moves a little as people set it up. The rover's display should respond to the current gap rather than yesterday's measurement.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)

VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 8.3 · Prove it, improve it, present it

2. Understand the idea

A continuous decision repeatedly senses, decides and acts. In a classroom test, we still bound the run. A one-time snapshot and a live display can use the same condition but behave differently over time.

A short wait between checks makes the screen readable. Put the sensor reading inside the repeated decision; a value measured once before the loop stays stale.

feedback	Use an observed result to influence what happens next.
stale	No longer current because the world may have changed.

Predict before you try

About how long do the waits total?

▼ **Compare your prediction**

20 × 0.5 = 10 seconds, with a little additional time for the other blocks.

---

3. Build, then investigate

1. Keep wheel motors unplugged. Place a broad target in front of Distance1 and build the bounded screen loop.
2. Predict the display for near, far and absent targets.
3. Run and move only the target during the ten-second observation window. Keep hands clear of electronics and cables.
4. Compare this with the single-decision lesson 7.2. Explain why a new target position now changes the message.

Build this stack in VEXcode IQ

```
when started
repeat (20)
  clear screen
  if <Distance1 found an object?> then
    if <(Distance1 object distance in mm) < (200)> then
      print [Near] on screen
    else
      print [Far] on screen
    end
  else
    print [No target] on screen
  end
  wait (0.5) seconds
end
```

▼ **What to look for**

The message changes as the target changes, then the bounded observation finishes.

▼ **Stuck? Follow this clue**

If the message never changes, check whether gap was sampled only once before the loop. If the screen flashes unreadably, increase the pause and avoid clearing more often than necessary.

ENGINEERING INVESTIGATION

How does update rate affect the display?

Change

Change the wait from 0.5 to 0.2 seconds.

Measure

Observe how quickly a changed target produces a new message.

Keep the same

Keep the branch rules and target movement similar; note that total run time also changes.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Keep the display active until a touch gesture or five seconds have elapsed.

Success checks

- There are two exit conditions.
- Readings happen inside the loop.
- The wheels stay unplugged.

▼ One worked approach · try yours first

Configure TouchLED1. Reset timer, then repeat until TouchLED1 pressed? OR timer >5. Keep the display decision and a 0.2-second wait inside. Stop manually if needed.

▼ Go further

Explain what a forever loop would do and why this test uses a bounded version.

5. Explain your evidence

What turns a one-time decision into feedback?

▼ Compare your explanation

Repeatedly taking new observations and using them in the next decision.

Notebook: project / prediction / change / raw result / explanation / next test.

[VEX · Distance Sensor blocks (use the 2nd gen section)]

[VEX · Control blocks]

[VEX · Timer blocks]

Approach and hold your place

Use repeated small corrections with a hold band and an explicit no-target response.

Your rover should approach a visible station, then leave a generous gap instead of bumping it.

1. Get ready

BaseBot + Distance Sensor

BaseBot, 2nd generation Distance Sensor, Touch LED

Mount Distance1 facing forward, clear of beams, cables and the claw. Select the 2nd generation Distance Sensor in Devices, match the port, and name it Distance1. Start with a large matte cardboard target 300 mm from the sensor face, with space behind it. Configure TouchLED1 too.

VEX · Build instructions (choose Education Kit, 2nd generation)

VEX · Distance Sensor blocks (use the 2nd gen section)

Bring this learning: 9.1 · Make a live station display

2. Understand the idea

Small steps reduce the distance traveled before another measurement. This is a simple feedback controller: sense the gap, choose a step or hold, then sense again.

A deadband is a region in which no correction is requested. This course never automatically reverses toward an unseen object behind the robot. If it is already too close, it holds and asks for a reset.

deadband	A range in which the controller makes no correction.
controller	A rule that uses observations to choose corrective actions.

Predict before you try

Will starting at 170 mm make this rover reverse?

▼ Compare your prediction

No. It holds. Reverse movement would require separate knowledge about the space behind it.

---

3. Build, then investigate

1. Use a broad target about 350 mm ahead. Clear the floor and verify detection before reconnecting wheel motors.
2. Build the loop with ten iterations and 20 mm maximum per step. The whole attempt requests at most 200 mm of forward travel.
3. Run from the marked pose. Measure the final sensor gap, not the wheel-to-wall distance.
4. Test with a near starting target and with no target. Those cases must hold rather than begin a blind approach.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
repeat (10)
  clear screen
  if <Distance1 found an object?> then
    if <(Distance1 object distance in mm) > (220)> then
      drive [forward] for (20) mm
    else
      stop driving
      print [Hold: inspect gap] on screen
    end
  else
    stop driving
    print [No target: hold] on screen
  end
  wait (0.1) seconds
end
stop driving
```

▼ What to look for

The rover takes short approach steps when the detected gap exceeds 220 mm, otherwise holds. It ends after ten checks.

▼ **Stuck? Follow this clue**

Overshooting can come from a too-large step, wrong reference point or stale measurement. Inspect the configuration and reduce step size; do not increase speed to compensate.

ENGINEERING INVESTIGATION

How does step size affect the final gap?

Change

Compare 10 mm and 20 mm steps over ten checks.

Measure

Record final gap and number of steps. Note that shorter steps also reduce total possible travel.

Keep the same

Use the same initial gap, speed, target and thresholds.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Display Too near below 180 mm and Ready in the 180–220 band.

Success checks

- Missing, too-near, ready and far cases are distinct.
- Only far commands a small forward step.
- Every trial has a finite limit and ends stopped.

▼ **One worked approach · try yours first**

Inside the detected branch, first check gap >220 for a step. Otherwise check gap <180 for Too near; else print Ready. Both lower regions stop driving. Keep all ten-iteration and velocity limits.

▼ **Go further**

Explain why detecting a station is not the same as detecting every possible obstacle.

5. Explain your evidence

Why can holding be a better response than correcting in both directions?

▼ Compare your explanation

The robot may know the space in front but not behind, so a reverse correction can introduce a new unobserved risk.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Drivetrain block reference

VEX · Distance Sensor blocks (use the 2nd gen section)

VEX · Control blocks

Follow the edge of a trail

Use a calibrated Optical threshold to make a bounded, two-way line-following experiment.

A tape trail leads between field stations. Your rover will follow one edge by making tiny corrections, then stop for inspection.

1. Get ready

BaseBot line explorer

BaseBot and Optical Sensor facing down

Rebuild as BaseBot or remove the claw attachment with power off. Mount Optical1 rigidly in front of the wheels, facing down 10–20 mm above a flat paper field, then adjust for clear dark/light readings. Secure a wide dark tape line to pale paper on the floor. Do not use floor edges as a line.

VEX · Build your first 2nd generation BaseBot

VEX · Optical Sensor blocks

Bring this learning: 9.2 · Approach and hold your place

2. Understand the idea

One downward sensor cannot see the whole line. This simple strategy follows an edge: on dark it turns one way, on light it turns the other. Small alternating corrections can keep the sensor near the boundary.

Choose the edge and turn directions together. With dark tape to the sensor's left and pale paper to its right, turn slightly right when dark and left when light. This is a starting strategy to test on your mount, not a universal line follower.

edge tracking	Use measurements on either side of a boundary to stay near it.
calibration transfer	Recheck that earlier calibration still fits the new setup.

Predict before you try

Why would using the opposite tape edge require different turn choices?

▼ **Compare your prediction**

Dark and light would be on the opposite sides, so the same correction would push the sensor away from the intended boundary.

3. Build, then investigate

1. Rebuild the BaseBot line explorer. Secure a straight 400 mm dark tape trail on pale paper; no tight curves. Recollect dark/light readings at the new mounted height. Create threshold and replace the example 45 with your justified value.
2. Place the sensor above the right-hand edge of the dark tape, facing along it. Paper-trace which direction each branch requests.
3. Build 12 iterations of a tiny turn and 10 mm forward step. Run once on the clear floor, watching the sensor path. Stop if it leaves the paper field.
4. Compare three trials with the same start. Record number of edge crossings and where the sensor ends. Photograph or sketch the path rather than declaring the whole trail solved.

Build this stack in VEXcode IQ

```
when started
set drive velocity to (20) %
set turn velocity to (15) %
set drive stopping to [brake]
set drive timeout to (3) seconds
set drive velocity to (10) %
set turn velocity to (10) %
set [Optical1] light power to (50) %
wait (0.2) seconds
set [threshold] to (45)
repeat (12)
  if <(Optical1 brightness in %) < (threshold)> then
    turn [right] for (3) degrees
  else
    turn [left] for (3) degrees
  end
  drive [forward] for (10) mm
end
stop driving
```

▼ **What to look for**

A short zigzag investigation near the chosen tape edge, limited to 120 mm commanded forward travel. Calibration and tuning may be needed.

▼ **Stuck? Follow this clue**

If it immediately turns away, verify which edge you chose. If it always chooses the same branch, inspect dark/light readings and sensor height before modifying the turn angle.

> **ENGINEERING INVESTIGATION**
>
> **Which correction stays closest to the edge?**
>
> **Change**
> Compare turn increments of 2 and 3 degrees in separate saved versions.
>
> **Measure**
> Record departures from the edge and final sensor location over three trials each.
>
> **Keep the same**
> Keep threshold, start position, step length and sensor mount unchanged.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Make a trail-following test report that another engineer can repeat.

Success checks

- It includes raw calibration values and the chosen edge.
- It identifies the correct turn direction on each surface.
- It includes failed trials and the finite travel limit.

▼ **One worked approach · try yours first**

Document dark-on-left/pale-on-right, measured threshold, 10 mm steps, chosen small turn and 12 iterations. Show the observed path and explain whether it stayed near the edge. Recalibrate before increasing the route length.

▼ **Go further**

Try one gentle curve after the straight-line tests; keep the same short travel limit and label the new conditions.

5. Explain your evidence

Why is this feedback rather than a memorized route?

▼ **Compare your explanation**

Each correction depends on a newly measured surface brightness rather than a predetermined sequence of left and right turns.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Optical Sensor blocks

VEX · Drivetrain block reference

VEX · Control blocks

Make it strong without making it heavy

Compare two frame designs with the same gentle loading test.

A wobbly rover wastes accurate code. Your engineering team needs a frame that holds its shape while using parts wisely.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 9.3 · Follow the edge of a trail

2. Understand the idea

An unbraced rectangle with pivoting corners can change shape. A diagonal can restrict that movement. The exact effect depends on where the connections are and whether the joints are rigid.

A fair comparison uses the same span, load and measurement. Adding many parts can improve stiffness, but also uses material and increases mass. The goal is a design that meets its requirement efficiently.

stiffness	How much a structure resists changing shape under a load.
brace	A member added to resist unwanted movement.
tradeoff	An improvement in one property that costs something in another.

Predict before you try

Why can a diagonal change how a rectangle behaves?

▼ **Compare your prediction**

It can prevent opposite corners from moving closer or farther apart, limiting the shape change.

---

3. Build, then investigate

1. Build a small four-beam rectangle with joints that can pivot gently. Sketch its dimensions and count the parts.
2. Add a diagonal using an angled beam or a braced arrangement that aligns with the kit's hole pattern. Never force unmatched holes.
3. Stand each version on the same flat surface and apply the same very gentle sideways hand force. Compare the change in shape qualitatively.
4. Explain where a sensor would be mounted and why changing its angle could hurt measurement quality.

Your engineering plan

```
DESIGN A: unbraced frame
DESIGN B: same frame plus a diagonal brace
TEST: same orientation and gentle sideways force
RECORD: part count, visible change in shape, any loosened joint
```

▼ **What to look for**

A documented structural comparison with a drawing and a reasoned choice, without bending parts to failure.

▼ **Stuck? Follow this clue**

If a joint keeps loosening, inspect pin length and engagement rather than adding random beams. Do not use enough force to damage the parts.

ENGINEERING INVESTIGATION

Does a brace make a useful difference?

Change
Add one brace or a pair of rigid corner connections.

Measure
Compare visible shape change and count extra parts.

Keep the same
Keep the frame size, orientation and gentle force similar; note that hand force is only approximate.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Build a rigid sensor mount using the fewest parts that satisfy your test.

Success checks

- It holds its angle during three gentle checks.
- The cable route stays clear of moving parts.
- You justify the selected design with evidence.

▼ One worked approach · try yours first

Use two spaced attachment points and a short supported beam. Compare it with a long single-ended mount. Select the version that keeps the sensor aimed consistently, and document any extra parts used.

▼ Go further

Explain why “strongest” and “best for this job” are not always the same requirement.

5. Explain your evidence

What makes this comparison fair?

▼ Compare your explanation

Both designs face a similar test; the intended design feature changes while the other conditions stay as consistent as possible.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Your Education Kit (228-8899) and parts inventory

Trade speed for turning force

Measure a gear ratio by hand and explain the speed/torque tradeoff.

A fast claw is not automatically a useful claw. Use the kit's 12- and 36-tooth gears to investigate a slower, stronger output.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 10.1 · Make it strong without making it heavy

2. Understand the idea

When a 12-tooth driver meshes with a 36-tooth driven gear, the driven gear makes one turn for three driver turns. Neighboring external gears turn in opposite directions.

Ideally the slower output can provide more torque, but friction reduces what a real mechanism delivers. Gears do not create extra energy. This lesson uses hand rotation, so no motor configuration or loaded stall test is needed.

driver	The input gear receiving the turn.
driven	The output gear moved by the driver.
ratio	A comparison between two quantities.

Predict before you try

How far does the 36-tooth output turn for one 12-tooth input turn?

▼ **Compare your prediction**

One third of a revolution, in the opposite direction.

3. Build, then investigate

1. With battery disconnected, support two shafts on beams so a 12-tooth and 36-tooth gear mesh freely. Use suitable spacing from the kit's hole system; ask an adult if the teeth bind.
2. Mark one tooth on each gear with a small removable paper flag. Turn the small driver by hand exactly three revolutions.
3. Record the large gear's turns and direction. Swap which gear you turn and repeat.
4. Draw a table for driver turns, output turns and direction. Explain which arrangement you would investigate for a slow lifting arm.

Your engineering plan

```
12-tooth DRIVER → 36-tooth DRIVEN
3 driver turns → about 1 output turn, opposite direction
36-tooth DRIVER → 12-tooth DRIVEN
1 driver turn → about 3 output turns, opposite direction
```

▼ What to look for

Hand measurements agree approximately with the tooth-count ratio when gears mesh without slipping.

▼ Stuck? Follow this clue

Binding indicates alignment or spacing problems. Support shafts and adjust the build; do not force teeth together or use a motor to push through resistance.

ENGINEERING INVESTIGATION

What changes when input and output swap?

Change
Drive the small gear, then drive the large gear.

Measure
Count output revolutions and direction for a known number of input turns.

Keep the same
Use the same gear pair, shafts and free-moving support.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Explain a gearing choice for a slower arm without claiming free power.

Success checks

- The ratio is calculated from tooth counts.
- Input/output and direction are labeled.
- The explanation includes friction and an energy tradeoff.

▼ One worked approach · try yours first

Use a small driver and larger driven gear to reduce output speed. For 12:36, output speed is one third and ideal torque is three times input torque; real friction reduces the available torque. The motor still supplies the energy.

▼ Go further

Add an idler gear and investigate direction while keeping the first and last gear sizes fixed.

5. Explain your evidence

Why is a slower output sometimes useful?

▼ Compare your explanation

It can provide a more suitable turning force and gentler motion for the task, at the cost of output speed.

Notebook: project / prediction / change / raw result / explanation / next test.

[VEX · Your Education Kit (228-8899) and parts inventory]

Keep the load low

Investigate stability and explain why a delivery rover carries a light load low.

A rover can have perfect directions and still tip if its load is badly placed. Design the cargo location before increasing the mission's ambition.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 10.2 · Trade speed for turning force

2. Understand the idea

A structure is more likely to tip when its combined center of mass moves beyond its support area. Keeping mass low and within the base helps stability.

A longer arm also increases the turning effect of a load about its pivot. A robot that can hold a load still may behave differently when accelerating or turning. Use a small unpowered model for this investigation rather than tipping a powered robot.

center of mass	A balance point representing how mass is distributed.
support area	The area enclosed by the points supporting a structure.

Predict before you try

Does the same parcel become harmless just because the motor can lift it?

▼ Compare your prediction

No. Stability, leverage and moving behavior still need separate consideration.

3. Build, then investigate

1. Build a small low platform from beams and attach an identical small beam bundle as a model load. Test on a flat surface with power disconnected.
2. Place the same load low and centered, then higher or farther to one side. Keep a hand nearby to catch it before it falls.
3. Use a gentle sideways nudge and compare which arrangement resists tipping better. Record observations without pretending hand forces are precisely equal.
4. Apply the finding to the Clawbot: use a light parcel, a low arm and slow turns. Do not test the robot's maximum lifting capacity.

Your engineering plan

```
A: same load, low and centered
B: same load, higher or off-center
OBSERVE: base contact, tilt tendency, joint movement
APPLY: low cargo, clear turns, modest speed
```

▼ What to look for

A qualitative comparison and a cargo-placement sketch connected to the rover design.

▼ Stuck? Follow this clue

If the model twists instead of tipping, the frame may be too flexible for this comparison. Improve the support before interpreting the result as a balance experiment.

ENGINEERING INVESTIGATION

Which cargo position is more stable?

Change
Change only the load location on the same unpowered model.

Measure
Describe its tipping tendency under a similar gentle nudge.

Keep the same
Keep the base and load unchanged.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	________________	________________	________________
2	________________	________________	________________
3	________________	________________	________________

4. Take the challenge

Redesign the delivery plan to reduce tipping tendency.

Success checks

- The parcel stays low and inside a reasonable support footprint.
- Turns have clear space and low speed.
- Your reason refers to the mechanical experiment.

▼ One worked approach · try yours first

Keep the arm at its verified low travel pose, choose an empty light parcel, avoid sharp high-speed turns and use a broad clear path. Document the tradeoff: less reach, but a more stable delivery.

▼ Go further

Sketch a wider base and explain which field clearances it would change.

5. Explain your evidence

Why should design tests separate lifting and stability?

▼ Compare your explanation

They measure different properties: a motor's ability to move a load does not establish that the whole robot remains balanced.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Your Education Kit (228-8899) and parts inventory

Earn your rover driving license

Pair and configure the included Controller, then demonstrate deliberate starts and stops.

Some field jobs need a person's judgment. Your first driver test rewards precise, gentle control rather than speed.

1. Get ready

BaseBot + Controller

BaseBot, charged 2nd generation Controller and Brain

Pair the Controller and Brain using the linked 2nd generation guide. Save a separate driver project. Add Controller in Devices and choose the lesson's joystick mode. Leave arm/claw buttons unassigned. Remove Controller from autonomous project copies so two command sources cannot compete.

VEX · Pair a 2nd generation Controller and Brain

VEX · Configure Controller drive modes

Bring this learning: 10.3 · Keep the load low

2. Understand the idea

The Controller adds a human decision-maker. Pairing connects Controller and Brain; configuring a drive mode tells the project how joystick input should control the drivetrain.

A driver project and an autonomous project should be separate saved files. If automatic joystick mapping remains active during an autonomous stack, a second command source may interfere. Joystick deflection affects speed; the drive-velocity setting in an autonomous stack is not a guarantee that mapped joystick speed is capped.

teleoperation	A person controls a robot from a distance.
interface	The controls and feedback through which someone uses a system.

Predict before you try

Does pairing alone define how your joysticks drive?

▼ **Compare your prediction**

No. The drive mode is part of the project configuration.

---

3. Build, then investigate

1. Charge Controller and Brain. Follow the linked 2nd generation pairing guide: Brain Settings → Link, then perform the L-button/Power-button gesture shown on its screen. Confirm the connected indicator.
2. Save a separate BaseBot-driver project. In Devices add Controller and choose Left Arcade for the drivetrain; leave other motors unassigned. Download the configured project.
3. On a clear floor, center the joystick before starting. Use very small deflections for short forward, reverse and turning movements, with an adult ready at the Brain stop control.
4. Practice five starts and releases to center. Then drive between two marks 200 mm apart. Stop the project between trials; do not leave a driver session running unattended.

Your engineering plan

```
DEVICES: Drivetrain configured to actual BaseBot
CONTROLLER: Left Arcade → Drivetrain
OTHER MOTOR BUTTONS: unassigned
TEST: small forward input → center → small turn → center
FINISH: stop the project
```

▼ **What to look for**

Small deliberate joystick movements and centered releases produce controlled starts and stops in the supervised test area.

▼ **Stuck? Follow this clue**

If the project will not run, check the paired Controller and its configuration. If autonomous behavior changes after this lesson, return to the autonomous file and remove Controller assignment.

ENGINEERING INVESTIGATION

How precisely can you stop at a mark?

Change

Practice the same short route three times.

Measure

Record stopping error and any unintended movement.

Keep the same

Use the same mode, route, surface and gentle input range.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Demonstrate a five-point driver check.

Success checks

- You identify the paired robot.
- You start with centered controls and clear space.
- You stop on request and end the project.

▼ One worked approach · try yours first

Check robot identity; clear field; centered joystick; short low-deflection test; stop and end session. Record whether every step was demonstrated, not just whether the route was completed.

▼ Go further

Explain why a faster successful run may be less useful than a slower repeatable one.

5. Explain your evidence

Why keep separate driver and autonomous files?

▼ Compare your explanation

Their command sources and configuration differ, and separate files make that ownership clear.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Pair a 2nd generation Controller and Brain

VEX · Configure Controller drive modes

Design controls for a person

Compare arcade and tank drive modes with the same measured task.

An interface that feels obvious to one person may confuse another. Test two control layouts and support your choice with evidence.

1. Get ready

BaseBot + Controller

BaseBot, charged 2nd generation Controller and Brain

Pair the Controller and Brain using the linked 2nd generation guide. Save a separate driver project. Add Controller in Devices and choose the lesson's joystick mode. Leave arm/claw buttons unassigned. Remove Controller from autonomous project copies so two command sources cannot compete.

VEX · Pair a 2nd generation Controller and Brain
VEX · Configure Controller drive modes

Bring this learning: 11.1 · Earn your rover driving license

2. Understand the idea

Left Arcade combines driving and steering on one joystick. Tank assigns the left and right drivetrain sides to separate joysticks. Neither is best for every learner or job.

Learning affects a comparison. If you always test one mode second, practice may make it look better. Alternate the order and allow equal practice before scored trials.

usability	How effectively a person can use a system for a task.
confounding factor	Another changing influence that can obscure what caused a result.

Predict before you try

Why might the second mode seem better even if the interface is not the cause?

▼ Compare your prediction

The driver has already practiced the route and may simply be improving.

3. Build, then investigate

1. Make two copies of the verified driver project. Configure one Left Arcade and the other Tank; keep arm/claw controls unassigned.
2. Tape the same short L-shaped route with a generous turn area. Let the driver practice each mode for the same short duration.
3. Run arcade, tank, then tank, arcade at small joystick deflections. Record stopping accuracy, unintended turns and driver confidence.
4. Ask the driver which felt easier and compare that opinion with the measured results. Explain any mismatch.

Your engineering plan

```
MODE A: Left Arcade — one joystick for drive and steer
MODE B: Tank — one joystick for each drivetrain side
ORDER: practice both; A, B, B, A
MEASURE: route error, accidental turns, confidence
STOP: center controls and end each trial
```

▼ What to look for

A small comparison with equal practice and a justified interface choice, not a claim that one mode is universally superior.

▼ Stuck? Follow this clue

If tank behaves unexpectedly, verify both motor sides and the selected mode in Devices. Avoid editing motor reversal simply because one interface feels unfamiliar.

ENGINEERING INVESTIGATION

Which mode helps this driver complete this route?

Change

Change only the Controller drive mode.

Measure

Record accuracy, unintended movements and a 1–5 confidence rating.

Keep the same

Keep robot, route, practice time and gentle driving limits similar.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Recommend a control layout to a new driver.

Success checks

- You explain how each tested mode maps input to wheels.
- You show both measured results and the driver's opinion.
- Your conclusion is limited to the tested person and task.

▼ One worked approach · try yours first

Choose the mode with fewer unintended actions and acceptable stopping accuracy. Show the four trial results, include confidence ratings, and explain that a different task or driver could prefer another mode.

▼ Go further

Test a second driver with the order reversed and compare preferences.

5. Explain your evidence

What is one confounding factor in a driving comparison?

▼ Compare your explanation

Practice, tiredness, different speed, changed route or a different starting pose could affect the result.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Configure Controller drive modes

Make the handover unmistakable

Design and test a clear transition between human driving and an autonomous mission.

One engineer drives to the start area; the rover then performs a saved delivery. Everyone must know who controls the wheels at each moment.

---

1. Get ready

BaseBot + Controller

BaseBot, charged 2nd generation Controller and Brain

Pair the Controller and Brain using the linked 2nd generation guide. Save a separate driver project. Add Controller in Devices and choose the lesson’s joystick mode. Leave arm/claw buttons unassigned. Remove Controller from autonomous project copies so two command sources cannot compete.

VEX · Pair a 2nd generation Controller and Brain

VEX · Configure Controller drive modes

Bring this learning: 11.2 · Design controls for a person

---

2. Understand the idea

Two competing sources of motion commands make behavior difficult to predict. This course uses an explicit stop-and-switch handover between separate projects, with the field inspected again.

Human procedures are part of a system. A protocol is a clear sequence people follow. Test it with a person who did not write it to discover ambiguous words.

protocol	An agreed sequence for a task or exchange.
ownership	Which person or program is currently responsible for an action.

Predict before you try

Is releasing a joystick the same as ending the driver project?

▼ Compare your prediction

No. The mapped driver project can still respond to later joystick input until it is stopped.

---

3. Build, then investigate

1. Write a four-stage handover: driver stops; driver project ends; engineer resets and inspects the field; autonomous project starts after a ready gesture.
2. Confirm the autonomous project has no Controller device mapping. Name both files and Brain slots clearly.
3. Rehearse with motors unplugged. A partner calls out the current owner at every stage and identifies the next required check.
4. Only after the rehearsal, demonstrate the handover with a short clear BaseBot route. Stop between modes and never switch while moving.

Your engineering plan

```
1. DRIVER owns motion → center sticks and stop project
2. NO ACTIVE MOTION → reset start and check clear field
3. ENGINEER selects verified autonomous project, no Controller mapping
4. PROGRAM owns motion → ready gesture, bounded run, stop
Any ambiguity → stop and clarify before starting
```

▼ What to look for

A partner can identify who controls motion and demonstrate a stopped state between modes.

▼ Stuck? Follow this clue

If the rover responds to sticks during an autonomous test, stop and check the selected project and Controller configuration. Do not add counteracting drive commands.

ENGINEERING INVESTIGATION

Can a new reader follow your handover?

Change
Give the written procedure to another learner.

Measure
Record skipped checks and ambiguous steps during an unpowered rehearsal.

Keep the same
Do not explain missing details until after their first attempt.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Create a handover card that prevents an accidental mixed mode.

Success checks

- It explicitly ends the current project.
- It names the autonomous project and required configuration.
- It requires a fresh field check and start confirmation.

▼ One worked approach · try yours first

Use short numbered commands and an ownership label at each stage. Include a stop-and-ask branch if the file, slot or Controller configuration is uncertain. Rehearse with the Brain screen before motion.

▼ Go further

Explain how a status light could help communicate ownership while still using a physical stop between projects.

5. Explain your evidence

Why is the stopped state useful during handover?

▼ Compare your explanation

It creates an observable boundary where the old command source ends and the next setup can be verified.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Configure Controller drive modes
VEX · Download and run a Blocks project

Find a problem you care about

Write an original problem brief that can be investigated with one Education Kit.

You now have a toolbox of ideas. Choose a small job at a model research camp that you would like a robot to do.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 11.3 · Make the handover unmistakable

2. Understand the idea

Begin with a need, not a complicated mechanism. A model library courier, a tabletop color request station or a trail-inspection rover can each teach different engineering ideas.

A good brief states who needs help, what counts as success and what constraints matter. Choose a classroom model: the project must not handle medication, hot objects, pets, people or real emergency decisions.

brief	A short description of the need, success criteria and constraints.
prototype	A simplified first version made to learn something.

Predict before you try

Why might a smaller prototype teach you more quickly?

▼ **Compare your prediction**

It isolates the uncertain part so you can tell what caused a result before other parts interact.

---

3. Build, then investigate

1. Choose one model job: transport an empty parcel, signal a colored request, inspect a tape trail, or another similarly small task.
2. Write three measurable requirements and three constraints. Include one kit, short bounded motion, light materials and a clear supervised field.
3. Sketch two possible designs using at most four motors. Reuse one robot build and sensor you have already tested.
4. Select the smallest useful prototype and identify its riskiest unanswered question. Use the teaching-guide rubric to plan your evidence.

Your engineering plan

```
FOR: who uses this model?
NEED: what small job needs doing?
SUCCESS: three observable checks
CONSTRAINTS: one kit, clear field, bounded motion
FIRST QUESTION: what must we learn before combining everything?
```

▼ **What to look for**

A clear brief, two sketches and a testable first question using hardware already in your kit.

▼ **Stuck? Follow this clue**

If the idea needs more than four motors or a sensor absent from the kit, simplify the job or use a paper model for the missing part. Do not present a paper model as a functioning device.

ENGINEERING INVESTIGATION

Which design best fits the brief?

Change
Compare two sketches against the same requirements.

Measure
List expected benefits, uncertain points and required kit parts.

Keep the same
Keep the user need and success criteria unchanged.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Defend your design choice in one minute.

Success checks

- Name the user and job.
- Identify a tradeoff and a testable uncertainty.
- Show that no extra robot hardware is required.

▼ One worked approach · try yours first

Example: a request lamp uses Optical1 and TouchLED1 with motors unplugged. It recognizes two tested cards and displays a readable status. It is simpler than moving a parcel, but it only communicates a request.

▼ Go further

Ask a teammate to challenge one requirement and revise it if their evidence is convincing.

5. Explain your evidence

What makes a requirement testable?

▼ Compare your explanation

It describes an observable outcome or measurement, not just a wish such as make it amazing.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Your Education Kit (228-8899) and parts inventory

Build the smallest useful prototype

Test one uncertain subsystem, then integrate only after its evidence is understood.

Inventors learn by making a small idea tangible. Your first prototype should answer a question, even when it fails.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 12.1 · Find a problem you care about

2. Understand the idea

Choose a subsystem: gripping, sensing, structure, navigation or communication. Use the relevant earlier lesson as the starting point and preserve its setup and stopping limits.

Change one thing per comparison. Save named versions and label whether a result came from a physical run, a paper trace or an unpowered build. These kinds of evidence answer different questions.

subsystem	One part of a larger system with its own job.
evidence	An observation or record supporting a claim.

Predict before you try

If the prototype fails, has the lesson failed?

▼ **Compare your prediction**

No. A controlled, recorded failure can identify the next useful design change.

3. Build, then investigate

1. Choose the earlier lesson that tests your first question: 3.3 for grip, 5.2/5.3 for optical sensing, 7.3/9.2 for approach, 10.1 for structure, or 2.1 for messages.
2. Use that lesson's exact robot setup and conservative first program. The build label here is an unpowered planning bench; switch to the linked lesson's build before running anything.
3. Make a named prototype-v1 project or sketch. Predict three cases, including one case that should fail or be rejected, then run only within the earlier lesson's field limits.
4. Record results, make one justified revision and test the same cases again. Combine subsystems only when each can be explained separately.

Your engineering plan

```
QUESTION → REUSE A TESTED LESSON → PREDICT THREE CASES
BUILD A SMALL VERSION → RECORD ALL RESULTS
CHANGE ONE THING → REPEAT THE SAME TESTS
INTEGRATE ONLY AFTER THE SUBSYSTEM IS UNDERSTOOD
```

▼ **What to look for**

A working or informative prototype with documented limits and one evidence-based revision.

▼ **Stuck? Follow this clue**

If many parts changed at once, you may not know which caused the improvement. Return to the last saved version and compare one change at a time.

ENGINEERING INVESTIGATION

Did the revision address the first uncertainty?

Change
Change one design feature between versions.

Measure
Compare the same three cases before and after.

Keep the same
Keep the success criteria, build setup and test method fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Prepare a prototype handoff for another engineer.

Success checks

- The setup and program version are identifiable.
- All cases have observed or not-run labels.
- The next change follows from the results.

▼ One worked approach · try yours first

Include a build sketch, device map, saved .iqblocks filename, raw results, one failure explanation and the next proposed test. If you only paper-traced the logic, state that hardware remains untested.

▼ Go further

Ask the partner to predict a fourth case before running it.

5. Explain your evidence

Why preserve a failed version?

▼ Compare your explanation

It records what was tested and allows a fair comparison with the revised design.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Open and save projects in Chrome

Welcome to the engineering showcase

Present a reproducible project, explain the evidence and choose a thoughtful next investigation.

Your final achievement is not only a robot that does something. It is a design that another young engineer can understand, test and improve.

1. Get ready

Unpowered engineering bench

Kit beams, pins, shafts, collars, 12/36-tooth gears and wheels

Disconnect the battery and work on a flat surface. Use spare parts or rebuild after photographing your robot. Turn mechanisms gently by hand. Keep loose parts in a tray; use the pin tool rather than teeth or force.

VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)

Bring this learning: 12.2 · Build the smallest useful prototype

2. Understand the idea

A showcase explains need, design, program, tests and limits. Demonstrate one representative success and discuss one difficult case. A clear explanation matters as much as an exciting run.

Self-assessment is a conversation with evidence. A reviewed lesson checkmark records learning activity; it is not an automatic engineering qualification. Use the rubric to choose which skill you want to practice next.

reproducible	Described clearly enough that another person can repeat the setup and test.
reflection	Use evidence to understand what you learned and what to investigate next.

Predict before you try

What should you do if the live demonstration fails?

▼ **Compare your prediction**

Stop the robot, describe what happened, show the earlier evidence and propose a focused test. Do not hide the failure or improvise uncontrolled motion.

3. Build, then investigate

1. Prepare the final build drawing, actual device map, saved project and a brief test report. Use the robot configuration and field limits from your chosen prototype lesson.
2. Have a partner attempt a stopped, unpowered setup from your instructions. Fix unclear steps before any demonstration.
3. Present the user need, one major design choice, how a sensor affects a decision, and what ends the program. Show results from three unchanged trials or label the unrun tests.
4. Use the guide rubric to discuss build, code, testing and explanation. Save your notebook and .iqblocks files. Choose a new investigation, such as a second parcel shape or a different trail, with its own fresh test plan.

Your engineering plan

```
MY PROJECT: need and constraints
MY DESIGN: build, device map and program stages
MY EVIDENCE: three trials plus a difficult case
MY LIMITS: conditions and tests not yet tried
MY NEXT QUESTION: one measurable investigation
```

▼ **What to look for**

A complete engineering portfolio and a next-step question grounded in what you actually observed.

▼ **Stuck? Follow this clue**

A last-minute failure is a reason to use your documented evidence and a stopped explanation. Reset only after checking the cause, not repeatedly hoping for a lucky run.

ENGINEERING INVESTIGATION

Can someone reproduce your project?

Change

Give a partner the final setup and test instructions.

Measure

Record where they need clarification during an unpowered walkthrough.

Keep the same

Keep the project version and acceptance checks fixed.

Your three-trial record

Trial	Condition / change	Prediction	Observed / units
1	______________	______________	______________
2	______________	______________	______________
3	______________	______________	______________

4. Take the challenge

Give a three-minute showcase with an honest claim.

Success checks

- The claim matches the recorded trials.
- You can trace input → decision → output.
- Another engineer can locate the build, project and test records.

▼ One worked approach · try yours first

Say: our model met these checks in these trials under these conditions. Show a sensor reading, trace its chosen branch, and explain the stopping condition. End with a specific new question rather than an unsupported claim that it always works.

▼ Go further

Teach a younger learner one small idea using your unpowered model, then ask them to explain it back.

5. Explain your evidence

What habit will you carry into the next project?

▼ **Compare your explanation**

Predict, make one clear change, test fairly, record the result, and explain what the evidence does and does not establish.

Notebook: project / prediction / change / raw result / explanation / next test.

VEX · Your Education Kit (228-8899) and parts inventory

FOR LEARNERS & GROWN-UPS

Teaching guide & engineering rubric

A course for your kit

This is a hands-on VEX IQ (2nd generation) course for one learner or a pair aged about 10–11. No prior code is required. All four Smart Motors, the Controller and each of the four separate sensors in Education Kit 228-8899 appear in the course. The built-in Inertial Sensor supports drivetrain heading.

The academy is a lesson and notebook app. Recreate block plans in VEXcode IQ and download from there to your real robot. The academy itself does not connect to, simulate or execute VEX hardware programs. Printed block plans and text downloads are readable guides, not importable .iqblocks files.

Choose a pace that leaves time to think

There are 36 lessons in 12 units, about 15 hours at 25 minutes each, plus building, setup and longer project trials. At three lessons a week, plan around 12 weeks. Units 1–9 form the coding core; Units 10–12 extend mechanics, human control and independent invention. A slower pace is welcome.

Treat construction as its own session. The BaseBot inspection lesson assumes the official build is complete; the claw calibration assumes a completed Clawbot. End at a clean step boundary, save a photo and record the next step. If a capstone needs another session, preserve the start conditions and resume.

A repeatable 25-minute routine

Start with a prediction before touching the robot. Read the goal and robot setup, then assemble the small program. Test the original version before changing anything. Reserve time for a challenge, an explanation and saving the evidence.

Do not race the clock when a build is unclear. The learner can work independently on reading, block assembly, paper traces, notebook work and unpowered comparisons after adult setup. Adults supervise first moving tests and all unfamiliar mechanism setup. If hardware is unavailable, paper-trace the program and label hardware tests not run.

Build progression and reuse

Unit 1 builds BaseBot; Unit 2 adds the Touch LED. Unit 3 introduces measured driving and then the Clawbot arm/claw. You can keep Clawbot for later BaseBot-style lessons if its arm remains low, the same drivetrain is retained and attachments do not obscure sensors; otherwise rebuild BaseBot. Update the device map whenever anything changes.

Units 4–5 investigate Bumper, Distance and Optical sensing. The optical bench uses no moving motors. Unit 8 combines the Clawbot, forward Distance Sensor and Touch LED. Unit 9 uses a downward Optical mount on a BaseBot. Unit 10 is unpowered mechanics. Unit 11 uses the Controller in separate driver projects. Unit 12 reuses a chosen tested build.

How to recreate a block plan

Open a New Blocks Project for 2nd gen. Start from a saved configuration for the right robot. Use Events for when started, Screen for print, Drivetrain for whole-robot movement, Motion for ArmMotor/ClawMotor, Sensing for device readings, Control for waits/loops/branches, Operators for comparisons and Boolean logic, Variables for stored values and My Blocks for named procedures.

Create named variables before using their set/change/oval reporter blocks. [Square brackets] in a reading plan represent a menu choice or entered text; (parentheses) represent a value or oval reporter; <angles> represent Boolean conditions. Nest the actual hexagonal operators and sensors in VEXcode. Indentation places actions inside C-shaped blocks. The word end in a plan labels the C block's bottom; it is not a separate block.

Keep the normal waiting form of drive for, turn for and spin for. A separate define stack belongs in the same workspace and runs only when its My Block is called. If a label differs after a VEXcode update, use the linked official block help to identify the operation and confirm its device generation.

First connection and each later download

Use a charged Brain, USB data cable and Chrome on the Mac. Select 2nd gen in VEXcode's Devices window, then connect through the Brain menu and choose the Brain serial device. The linked Mac guide covers selecting among multiple ports. A green Brain icon indicates a connection. An adult handles any firmware prompt using the official guide.

Save your editable .iqblocks file on the Mac. Select a Brain slot, Download, wait for completion, then Run the correct project. Editing without downloading leaves the older Brain program in place. Check the project name and slot when results look stale. Stop first, disconnect USB before driving, and run from the Brain in the cleared floor area.

Device configuration without guessing

Choose Drivetrain 2-Motor, identify robot-left and robot-right ports, select Brain Inertial and set wheel travel/gear ratio to match the actual build. Never configure those wheel motors again as independent motors in the same project. Follow a short forward test with immediate stopping if the sides or reversals are wrong.

Add only needed sensors and individual mechanism motors. Name devices to match the lesson plans, then use those names in drop-downs. The kit has a second-generation Distance Sensor and an Optical Sensor; older distance and Color Sensor blocks are not substitutes. Keep driver and autonomous project configurations separate. A configured Controller must be paired, and joystick mapping can interfere with autonomous movement.

Limits, timeouts and honest outcomes

Use the supplied low velocities and short distances for initial tests. A drive timeout bounds a waiting drive-for or turn-for attempt. It does not automatically bound a plain drive command, an endless loop or a wait-for-sensor. The contact lesson explicitly includes an elapsed-time condition and a stop command.

A timeout can let a stack continue even though the intended motion was incomplete. The course requires an observer to stop and mark that trial failed when a movement stalls, times out or leaves the route. A reached screen message is not proof of physical success. Do not automatically recover by driving in an unobserved direction.

A stopped gripper with hold may still exert force. End the project and have an adult help restore the known start with power off. Motor angles and calibration values must come from the learner’s actual robot. This course has been reviewed against public documentation; its block plans have not been physically validated on your kit.

Help without taking over

Use three prompts: What did you predict? What did you observe? What is the smallest change that could test your explanation? Let the learner point to a block, sensor reading or joint before offering the answer.

For support, reduce the route, keep a partially assembled stack visible, read the instructions aloud, or compare two choices. For extension, use the lesson's stretch question or a fresh test case. Pair learners as builder/programmer and checker/test recorder; swap every lesson. Keep short words on screen as well as colors.

An engineering notebook worth keeping

Record the date, project filename, build revision, actual ports, start pose, hypothesis, one changed factor, unchanged conditions, raw trial results and the next question. Save drawings and photos alongside the .iqblocks files on the Mac; this text notebook does not store images.

Use the lesson's three-row trial table for quick measurements. For longer investigations, add raw values in the notes. Include failures, no detections and not-run tests. A paper trace supports logical reasoning; a physical trial supports behavior under its particular conditions. Browser review checkmarks are organizational aids.

Capstone review and portfolio

Review build, programming, testing and explanation separately using the rubric below. Ask the learner to show evidence for each claim. A learner may be independent at code tracing while still needing help with mechanical assembly.

For the supply expedition, require a field map, known gentle gripper settings, a named project, three unchanged trials, one justified revision and a comparison. For the final invention, require a need, constraints, two candidate designs, a subsystem test, a reproducible setup and a clear statement of limits. These are classroom learning outcomes, not competition scores or a certification.

Troubleshooting in the right order

Connection problem: inspect charged power, cable and USB adapter, Chrome permission, closed competing VEX applications and firmware. Program not changing: verify file, slot, download and selected Brain project. Missing blocks: inspect Devices, generation and the configured device type.

Movement problem: stop and inspect physical build, ports, wheel size, reversal, rubbing parts and calibration before changing logic. Sensor problem: inspect its mount, sight line and raw readings before changing thresholds. Logic problem: trace a minimal example and test both accepted and rejected cases. Save a known working version before experimenting.

Sources and the rest of the academy

CMU's public VEX IQ Gen 2 sequence informed the first nine unit topics. These are original lessons and challenges; the supply-station expedition is not a reproduction of CMU's subterranean lesson. Official VEX pages supply hardware/build references and block semantics. External build guides and VEXcode need an internet connection when opened.

The existing Reachy missions, Python coding course and Spark Lab remain separate learning paths. Their progress is not migrated into VEX. The academy's current knowledge-atlas evidence audit does not include this new VEX course or the newer Python coding path; atlas coverage must not be treated as a learner's mastery.

Discuss your evidence together

Engineering portfolio rubric • discuss, do not auto-score

Area	With support	Developing independence	Independent & explains why
Build & setup	I need help matching parts and ports.	I build and configure with a checklist.	I independently inspect the build and explain a mechanical choice.
Program & explain	I can trace a small supplied sequence with help.	I can modify a sequence, loop or branch and predict the result.	I decompose a task, handle missing input and explain why every path ends.
Test & improve	I record an observation with support.	I run fair repeated tests and preserve failures.	I justify one revision using raw data and test meaningful boundary cases.
Communicate & reflect	I can describe what my robot did.	I link a design claim to recorded evidence.	Another engineer can reproduce my setup; I state limits and a useful next question.

Source notes & course mapping

Original academy explanations, block plans, experiments and challenges. Units 1–9 follow the public topic progression of CMU Robotics Academy's VEX IQ (2nd gen) overview; Units 10–12 extend mechanical design, driver control and invention. Only the public CMU/CS2N outline was reviewed, not restricted lesson media or assessments. VEX Robotics is credited for the linked official hardware, build and programming documentation. This independent course is not endorsed or certified by CMU or VEX.

Reference progression → academy units

Public topic / extension	This course

Public topic / extension	This course
Getting Started	1. Meet your engineering team
Programming the Brain	2. Program the field station
Robot Movement	3. Move with a plan
Digital Sensors	4. Listen for a yes or no
Analog Sensors	5. Measure the world
Loops	6. Repeat with a purpose
Discrete Decisions	7. Choose the right action
Capstone: Subterranean Challenge	8. The supply-station expedition
Continuous Decisions	9. Keep checking the world
Academy extension · mechanics	10. Think like a mechanical engineer
Academy extension · controller	11. Put a human in the loop
Academy extension · invention	12. Your own engineering studio

CMU · Coding and Computational Thinking with VEX IQ (2nd gen)
CS2N · Public course outline
VEX · Your Education Kit (228-8899) and parts inventory
VEX · Build instructions (choose Education Kit, 2nd generation)
VEX · Build your first 2nd generation BaseBot
VEX · 2nd generation BaseBot and Clawbot build guides
VEX · Connect a Brain to web VEXcode IQ on a Mac
VEX · Update 2nd generation Brain firmware
VEX · Open and save projects in Chrome
VEX · Download and run a Blocks project
VEX · Configure a 2-motor drivetrain (2nd gen)
VEX · Drivetrain block reference
VEX · Motor block reference
VEX · Brain screen blocks
VEX · Touch LED blocks
VEX · Bumper Switch blocks
VEX · Distance Sensor blocks (use the 2nd gen section)
VEX · Optical Sensor blocks
VEX · Control blocks
VEX · Timer blocks
VEX · My Blocks and parameters
VEX · Pair a 2nd generation Controller and Brain
VEX · Configure Controller drive modes

Your engineering vocabulary

acceptance check

An observable result required for a successful trial.

Learn it in 8.1 · Draw a mission worth testing

and

True only when both conditions are true.

Learn it in 4.3 · The two-clue launch lock

Boolean

A value that is either true or false.

Learn it in 4.1 · A button that means ready

boundary

A value where the outcome of a rule changes.

Learn it in 7.2 · Near, ready, or too far?

bounded

Limited by a count, time or other stopping condition.

Learn it in 6.3 · Search, then stop searching

brace

A member added to resist unwanted movement.

Learn it in 10.1 · Make it strong without making it heavy

branch

One possible path through a program.

Learn it in 7.1 · One question, two messages

brief

A short description of the need, success criteria and constraints.

Learn it in 12.1 · Find a problem you care about

brightness

How much light a sensor measures.

Learn it in 5.2 · Sort a message by color

calibration

Connect a measurement or setting to a known physical reference.

Learn it in 3.3 · Meet the arm and claw

calibration transfer

Recheck that earlier calibration still fits the new setup.

Learn it in 9.3 · Follow the edge of a trail

center of mass

A balance point representing how mass is distributed.

Learn it in 10.3 · Keep the load low

condition

A true/false expression used to choose an action.

Learn it in 7.1 · One question, two messages

confounding factor

Another changing influence that can obscure what caused a result.

Learn it in 11.2 · Design controls for a person

constraint

A boundary your design must work within.

Learn it in 8.1 · Draw a mission worth testing

controller

A rule that uses observations to choose corrective actions.

Learn it in 9.2 · Approach and hold your place

deadband

A range in which the controller makes no correction.

Learn it in 9.2 · Approach and hold your place

decomposition

Break a larger job into smaller jobs.

Learn it in 2.3 · Build a launch console

detection

Whether the sensor currently reports an object.

Learn it in 5.1 · Measure an invisible gap

download

Send a program to the robot Brain.

Learn it in 1.3 · Hello, field station!

driven

The output gear moved by the driver.

Learn it in 10.2 · Trade speed for turning force

driver

The input gear receiving the turn.

Learn it in 10.2 · Trade speed for turning force

drivetrain

The motors and wheels that move the chassis.

Learn it in 1.2 · Build a rover that rolls straight

edge tracking

Use measurements on either side of a boundary to stay near it.

Learn it in 9.3 · Follow the edge of a trail

error

The difference between the requested and observed result.

Learn it in 3.1 · The 200 mm delivery

evidence

An observation or record supporting a claim.

Learn it in 12.2 · Build the smallest useful prototype

fallback

A chosen response when the usual information or action is unavailable.

Learn it in 7.3 · A rover that can say “I cannot tell”

feedback

Use an observed result to influence what happens next.

Learn it in 9.1 · Make a live station display

fresh reading

A measurement taken again rather than reusing an old value.

Learn it in 6.3 · Search, then stop searching

friction

Resistance when surfaces move against one another.

Learn it in 1.2 · Build a rover that rolls straight

function

A named, reusable group of instructions; called a My Block here.

Learn it in 6.2 · Teach the rover a new word

heading

Direction measured from a chosen reference.

Learn it in 3.2 · Turn the corner

hue

A numeric description of color around a color wheel.

Learn it in 5.2 · Sort a message by color

initialize

Give a variable its starting value.

Learn it in 2.2 · Give your rover a memory

input

Information entering a system.

Learn it in 1.1 · The kit detective

integration

Join tested parts into one working system.

Learn it in 8.2 · Integrate the delivery rover

interface

The controls and feedback through which someone uses a system.

Learn it in 11.1 · Earn your rover driving license

iteration

One trip through a loop’s body.

Learn it in 6.1 · A patrol in four steps

limit

A boundary chosen to keep motion within a tested range.

Learn it in 3.3 · Meet the arm and claw

loop

A group of instructions that repeats.

Learn it in 6.1 · A patrol in four steps

millimeter

A length unit; ten millimeters equal one centimeter.

Learn it in 3.1 · The 200 mm delivery

nested

Placed inside another block or decision.

Learn it in 7.2 · Near, ready, or too far?

not

An operator that reverses true and false.

Learn it in 4.1 · A button that means ready

numeric

Expressed as a number.

Learn it in 5.1 · Measure an invisible gap

or

True when at least one condition is true.

Learn it in 4.2 · Dock at the soft wall

output

An action or signal produced by the system.

Learn it in 1.1 · The kit detective

ownership

Which person or program is currently responsible for an action.

Learn it in 11.3 · Make the handover unmistakable

parameter

An input supplied when a reusable block is called.

Learn it in 6.2 · Teach the rover a new word

policy

A set of rules mapping conditions to actions.

Learn it in 7.3 · A rover that can say "I cannot tell"

port

A numbered socket connecting a device to the Brain.

Learn it in 1.2 · Build a rover that rolls straight

project

Your blocks together with the device configuration.

Learn it in 1.3 · Hello, field station!

protocol

An agreed sequence for a task or exchange.

Learn it in 11.3 · Make the handover unmistakable

prototype

A simplified first version made to learn something.

Learn it in 12.1 · Find a problem you care about

range

The smallest through largest values in a set.

Learn it in 5.3 · Find the edge of the trail

ratio

A comparison between two quantities.

Learn it in 10.2 · Trade speed for turning force

recovery

Return to a known condition after an interrupted attempt.

Learn it in 8.2 · Integrate the delivery rover

reference point

The point from which a measurement starts.

Learn it in 5.1 · Measure an invisible gap

reflection

Use evidence to understand what you learned and what to investigate next.

Learn it in 12.3 · Welcome to the engineering showcase

relative

Measured from the current state rather than a fixed reference.

Learn it in 3.2 · Turn the corner

reliability

How consistently a system performs its intended task under stated conditions.

Learn it in 8.3 · Prove it, improve it, present it

reproducible

Described clearly enough that another person can repeat the setup and test.

Learn it in 12.3 · Welcome to the engineering showcase

requirement

A condition a design must satisfy.

Learn it in 2.3 · Build a launch console

sample

One object or reading used in an investigation.

Learn it in 5.2 · Sort a message by color

sequence

Instructions executed in a particular order.

Learn it in 1.3 · Hello, field station!

stage

One clearly defined part of a larger mission.

Learn it in 8.1 · Draw a mission worth testing

stale

No longer current because the world may have changed.

Learn it in 9.1 · Make a live station display

stiffness

How much a structure resists changing shape under a load.

Learn it in 10.1 · Make it strong without making it heavy

structure

The parts that hold the robot together.

Learn it in 1.1 · The kit detective

subsystem

One part of a larger system with its own job.

Learn it in 12.2 · Build the smallest useful prototype

support area

The area enclosed by the points supporting a structure.

Learn it in 10.3 · Keep the load low

teleoperation

A person controls a robot from a distance.

Learn it in 11.1 · Earn your rover driving license

threshold

A comparison boundary used to make a decision.

Learn it in 5.3 · Find the edge of the trail

timeout

A limit that ends an attempt after too much time.

Learn it in 4.2 · Dock at the soft wall

timing

When an action starts and how long it lasts.

Learn it in 2.1 · Send a secret light message

torque

A turning effect that can move a load.

Learn it in 3.3 · Meet the arm and claw

trace

Follow a program step by step on paper.

Learn it in 2.1 · Send a secret light message

tradeoff

An improvement in one property that costs something in another.

Learn it in 10.1 · Make it strong without making it heavy

trial

One repeat of a controlled test.

Learn it in 3.1 · The 200 mm delivery

truth table

A table listing input combinations and the result of a rule.

Learn it in 4.3 · The two-clue launch lock

usability

How effectively a person can use a system for a task.

Learn it in 11.2 · Design controls for a person

variable

A named value a program can read and change.

Learn it in 2.2 · Give your rover a memory