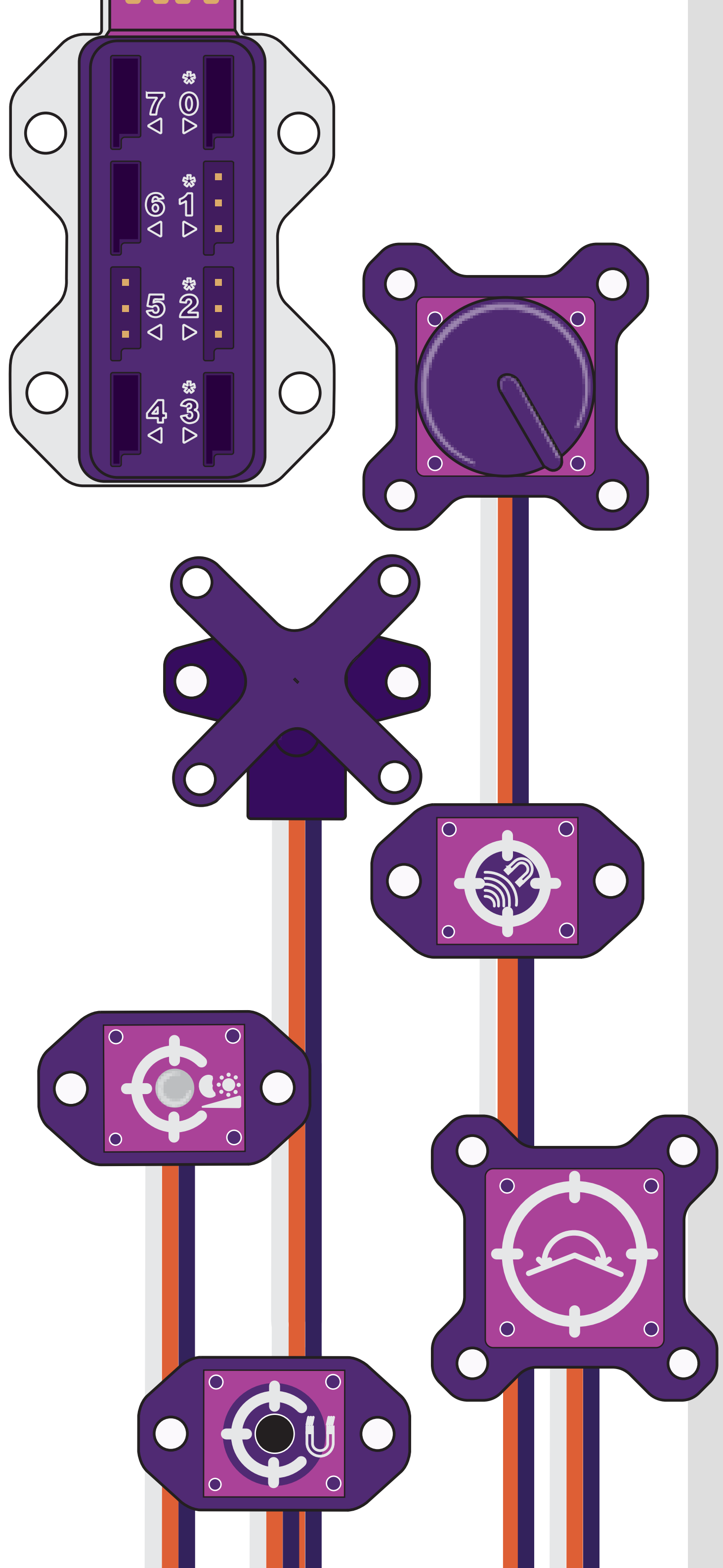


Invention Engine Lessons
Unit 2

Engineer to invent – Teacher guide

Invention Engine

The fun of inventing is the journey

Release number: 3.2.4.8.2.1

Contents

Introduction

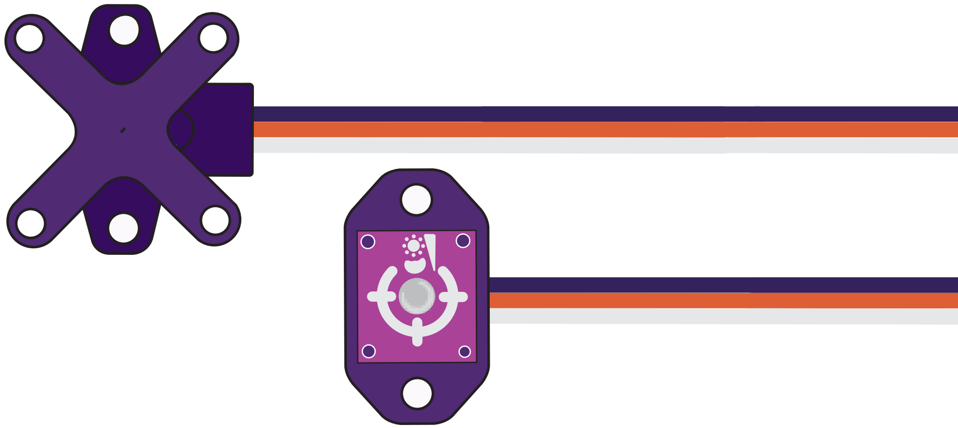
Teaching using Invention Engine 1
Class culture 2
Activities timings 2
Feedback 2
Troubleshooting 3

Lessons

The invention cycle 4
Lesson 1 – A solution for drowsy driving 11
Lesson 2 – Mapping and divergent thinking 12
Let's invent Activity 2 14
Lesson 3 – Snippets and movement 15
Let's Invent Activity 3 18
Lesson 4 – Boolean expressions and light sensitivity 19
Let's Invent Activity 4 21
Lesson 5 –Random numbers and the interactive book 22
Let's Invent Activity 5 23

Lesson scope and sequence

The invention cycle 24
Lesson 1 – A solution for drowsy driving 25
Lesson 2 – Mapping and divergent thinking 26
Lesson 3 – Snippets and movement 27
Lesson 4 – Boolean expressions and light sensitivity 28
Lesson 5 –Random numbers and the interactive book 29
What to look for in students work 30



Invention Engine Lessons – Unit 2 by

microbric
motivate • create • educate

Illustrated by
Emma Dewar

Contributions from
Whymaker

Teaching using Invention Engine

Welcome! Thank you for choosing Invention Engine!

In this unit, your students will develop a robust process for engineering, making, tinkering, and inventing, whilst developing advanced coding skills.

Each activity focuses on developing a key computer science concept and introducing a new bit.

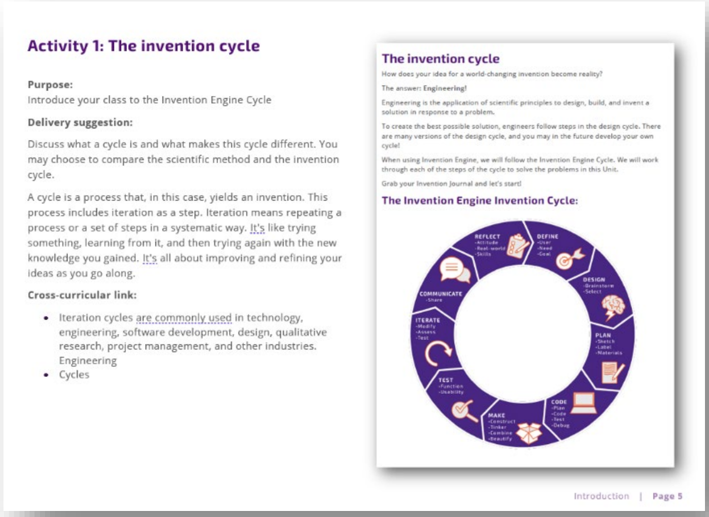
Each of the five activities include a ‘Let’s invent’ project, designed to support students in consolidating their skills.

When inventing, students should follow the steps in the Invention cycle and record their work in the Invention Journal.

Journaling is an important part of the process because it helps students document the process of inventing. The journal and the invention are both; proof of the learning journey and the outcome achieved.

This guide is structured to work in parallel with the student guide. This way, the student guide and the corresponding teaching notes can be viewed at the same time.

Teacher guide **Student guide**



Activity 1: The invention cycle

Purpose:
Introduce your class to the Invention Engine Cycle

Delivery suggestion:

Discuss what a cycle is and what makes this cycle different. You may choose to compare the scientific method and the invention cycle.

A cycle is a process that, in this case, yields an invention. This process includes iteration as a step. Iteration means repeating a process or a set of steps in a systematic way. It's like trying something, learning from it, and then trying again with the new knowledge you gained. It's all about improving and refining your ideas as you go along.

Cross-curricular link:

- Iteration cycles are commonly used in technology, engineering, software development, design, qualitative research, project management, and other industries. Engineering
- Cycles

The invention cycle

How does your idea for a world-changing invention become reality?

The answer: **Engineering!**

Engineering is the application of scientific principles to design, build, and invent a solution in response to a problem.

To create the best possible solution, engineers follow steps in the design cycle. There are many versions of the design cycle, and you may in the future develop your own cycle!

When using Invention Engine, we will follow the Invention Engine Cycle. We will work through each of the steps of the cycle to solve the problems in this Unit.

Grab your Invention Journal and let's start!

The Invention Engine Invention Cycle:

Introduction | Page 5

Tip: In computer science, engineering, and design thinking, there are many ways to solve a problem. Encourage students to develop their own solution and approach.

Class culture

Foster a shared culture that encourages creativity and celebrates generating ideas.

Declare that in your class, we (as a class) embrace challenges, praise effort, display persistence, focus on the process not the result, seek feedback and see failure as an opportunity to grow.

Remind students that learning is supposed to feel uncomfortable. Provide your own examples.

It is brave to try new things, and to learn new skills.

Activities timing

The timing for each activity depends on your students. The best learning happens when students are invested in their projects and have time to fail, and problem solve.

Agreeing on the milestones for each section is an effective way to support students' progress through the project.

Tip: Agree on a 'clean up time' with your students and share your expectations with your class before starting any project.

Feedback (assessment)

Think of assessment as providing information to students to improve.

While working on the invention, ask students to form a small group of 'critical friends' to bounce ideas off. Critical friends are a good way to encourage collaboration.

Consider providing students the opportunity to hand over an early version of their journal and invention and to resubmit a final version of their journal and invention once they have addressed your feedback.

A guide with suggestions on what to notice in students work and journals is provided in the final pages of this guide.

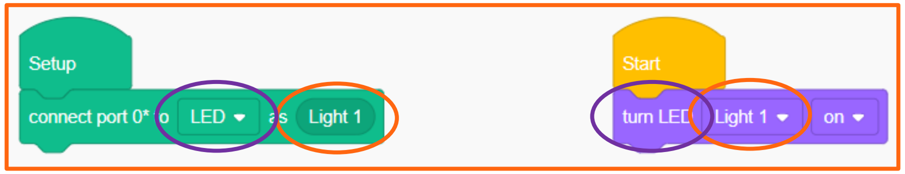
Troubleshooting

Troubleshooting 1: *I can't get the Hub to program.*
Check:

- that the Hub is pushed into the USB A socket of your device all the way.
- that there is an internet connection available.
- your code.

Troubleshooting 2: *I can't get the program to work. The bits are not doing what I coded them to do.*
Check:

- that the bit connected to the Hub is firmly connected, and that it matches the bit selected in the Setup blocks.
  For example:
  - the bit connected to the **Hub port** labelled '0' is the LED bit.
  - the 'LED' parameter is selected under the **Setup block** 'connect port 0'.
- that the name of the bit AND type of bit under the Setup block matches the information of the block under the Start block.

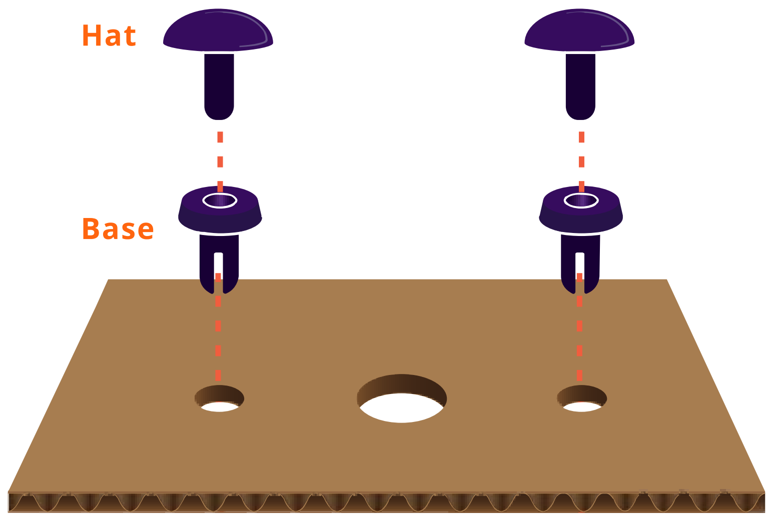


- that the power source (the Battery box bit or the computer) is appropriately powering the Hub and bits.
- the bug box for errors in the code.

Troubleshooting 3: *I can't attach the bits to the cardboard, or the bits keep falling off.*

Check:

- that the cardboard is the appropriate thickness (use the gauge in the stencil).
- that the rivets are fastened. The base goes into the cardboard first, then the hat is pushed all the way in through the base and the cardboard.
- that the cardboard is sturdy.



The invention cycle

Purpose:

To introduce your class to the Invention Engine Cycle, the different steps and the associated tools and models applicable within each of the steps.

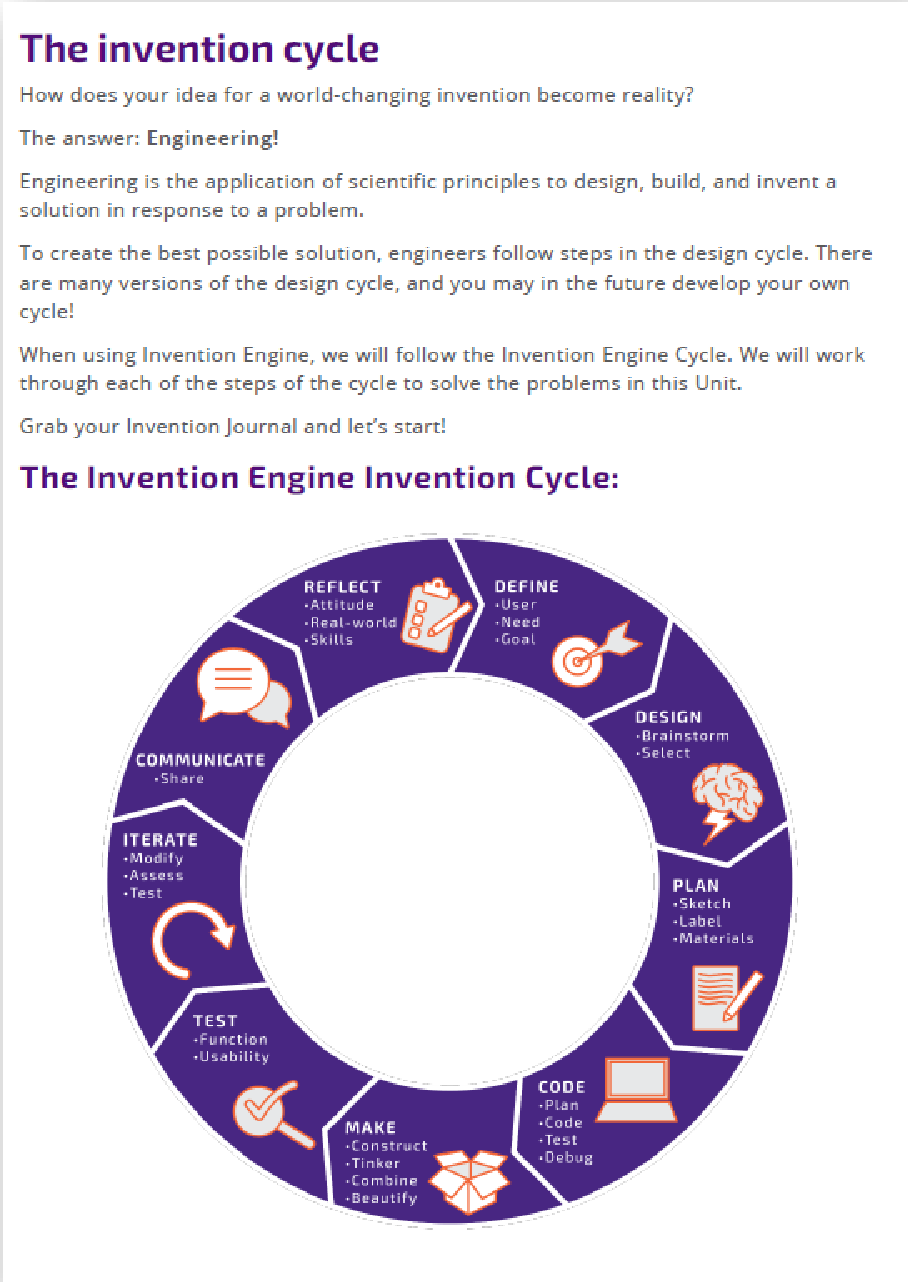
Delivery suggestion:

Discuss what a cycle is and what makes this cycle different. You may choose to compare the scientific method and the invention cycle.

A cycle is a process that, in this case, yields an invention. This process includes iteration as a step. Iteration means repeating a process or a set of steps in a systematic way. It translates into trying something, learning from it, and then trying again with the new knowledge you gained. It is all about improving and refining your ideas as you go along.

Cross-curricular links:

- Life Sciences: Investigate how engineering can be applied to improve ecosystems, such as designing structures to prevent erosion, or creating habitats for local wildlife.
- History of Technology: Explore the historical development of various technologies and how they have shaped societies. Discuss the societal impacts of inventions and innovations.
- Art Sketching and Visualization: Teach students how to create detailed sketches and diagrams to communicate their design ideas visually.


The invention cycle

How does your idea for a world-changing invention become reality?

The answer: **Engineering!**

Engineering is the application of scientific principles to design, build, and invent a solution in response to a problem.

To create the best possible solution, engineers follow steps in the design cycle. There are many versions of the design cycle, and you may in the future develop your own cycle!

When using Invention Engine, we will follow the Invention Engine Cycle. We will work through each of the steps of the cycle to solve the problems in this Unit.

Grab your Invention Journal and let's start!

The Invention Engine Invention Cycle:

1. Define

Talk about empathy as the means to really understand what a problem is like for the user.

Explore the importance of defining a problem before solving it. Some considerations are:

- clarity and goal alignment: effective and efficient solutions need a clear definition to ensure we are solving the right problem.
- efficient resource allocation: understanding a problem makes planning for the adequate resourcing of a project easier. For example, money, time, or the relevant team members to work in the invention.
- enables evaluation: defining a problem correctly helps in defining the criteria to evaluate the solution.
- avoids solution bias: defining the problem upfront, helps to explore the issue from various angles, consider different perspectives, to generate a range of potential solutions, rather than rushing and settling on a particular solution without thoroughly evaluating the problem.

Tip: Encourage students to do background research to help them in solving the problem to gain insight on the problem, and gain inspiration from ideas that have been tried before.

1. Define

When we approach a problem, we want to solve with an invention, it is a good idea to consider the problem in detail. To do this, we can use **empathy** to help us understand the person's (user's) difficulty or problem.

We can ask ourselves:

- Why is this a problem for them? Is it a need or want?
- What would a great outcome for the user look like? What is the goal?

Fuel your vocabulary

Empathy is the ability to understand and share the feelings of another.

Engineers use **empathy** to understand the problem they are trying to solve and to anticipate how people might use a product. This perspective often produces a great solution to a problem.

You can learn more about your users by talking to and observing them. The information you learn will help you put yourself in your user's situation to design a great solution.

Some tools to help you cultivate your empathy are:

- Be curious about the user
- 'Walk' in their shoes
- Keep questions open ended
- Seek feedback
- Keep your personal biases in check

2. Design

The design step in the Invention Engine cycle is also known as ideation.

Tip: As a class, think about what is essential for the invention to do or have for it to solve the problem. Work to define the projects constraints and specifications.

Consider adding a 'project management' component to judge project success. Remind students that they have a responsibility for planning, organising, controlling resources, monitoring timelines and activities, and completing their invention.

The design, or ideation step, refers to the process of generating, developing, and refining ideas.

You may choose to lead your students in a brainstorming exercise, and work to select one idea together. The goal is to foster a creative and collaborative environment where students can express their thoughts openly, build upon each other's ideas, and explore different perspectives to find innovative solutions.

Alternatively, encourage students to work independently and collaborate where possible.

In the Invention cycle, the design step has been combined with the selection step to choose one idea to develop. Continue to work with your students to choose one idea by working through the provided list.

Optional activity: In addition to brainstorming, explore other ideation methods such as mind mapping.

2. Design

A design is a plan or drawing to show the look, and workings of your invention before it is made. In the design phase we brainstorm, select and plan.

Brainstorming:
Brainstorming is a technique that engineers and inventors use to generate, express, and organise ideas. It simply means thinking freely and suggesting as many ideas as possible, no matter how different the ideas may seem.

When brainstorming it may be helpful to:

- Stay focused! Brainstorming is a creative activity but keep your eye on the problem you are trying to solve.
- Produce ideas. Capture all ideas; at this stage, there are no 'bad' ideas.
- Respect ideas. If you are working as a team, remember that everyone's approach to solving a problem may be very different. This is what makes a team great!
- Combine and build on ideas. You can add, build, mix and match ideas to make an idea whole.

Select:
How do you choose just ONE idea?

- Cross out any idea that is unethical, illegal, or impossible.
- Does the idea solve the problem?
- Do you want to do it? Are you committed to it?
- Can you do it? Cross out any idea outside the project's **constraints**.
- Cross out ideas that don't meet the projects **specifications**.
- If your solution already exists, can your design improve the existing solution?
- Choose the most promising solution!

Fuel your vocabulary
Constraints are the 'things' that limit your design, like money, time, materials, etc.

For example, the constraints of using Invention Engine include: the bits can only be used with cardboard and the size of the bits means that you can't make micro-devices.

Specifications are the requirements that a solution, product or invention must meet. For example, be sturdy, be waterproof, fit in an adult's hand, etc.

3. Plan

The plan for the hardware should include a plan for the physical invention including the list of materials, a sketch, and annotations.

Sketching is an important part of the process because it represents student's ideas or concepts and the progression of their learning.

Concept sketches are simple and focus on capturing the basic form or structure of the idea. Sketches must be labelled and drawn to scale.

Stretch your students to include design notes or annotations to illustrate how the invention will work.

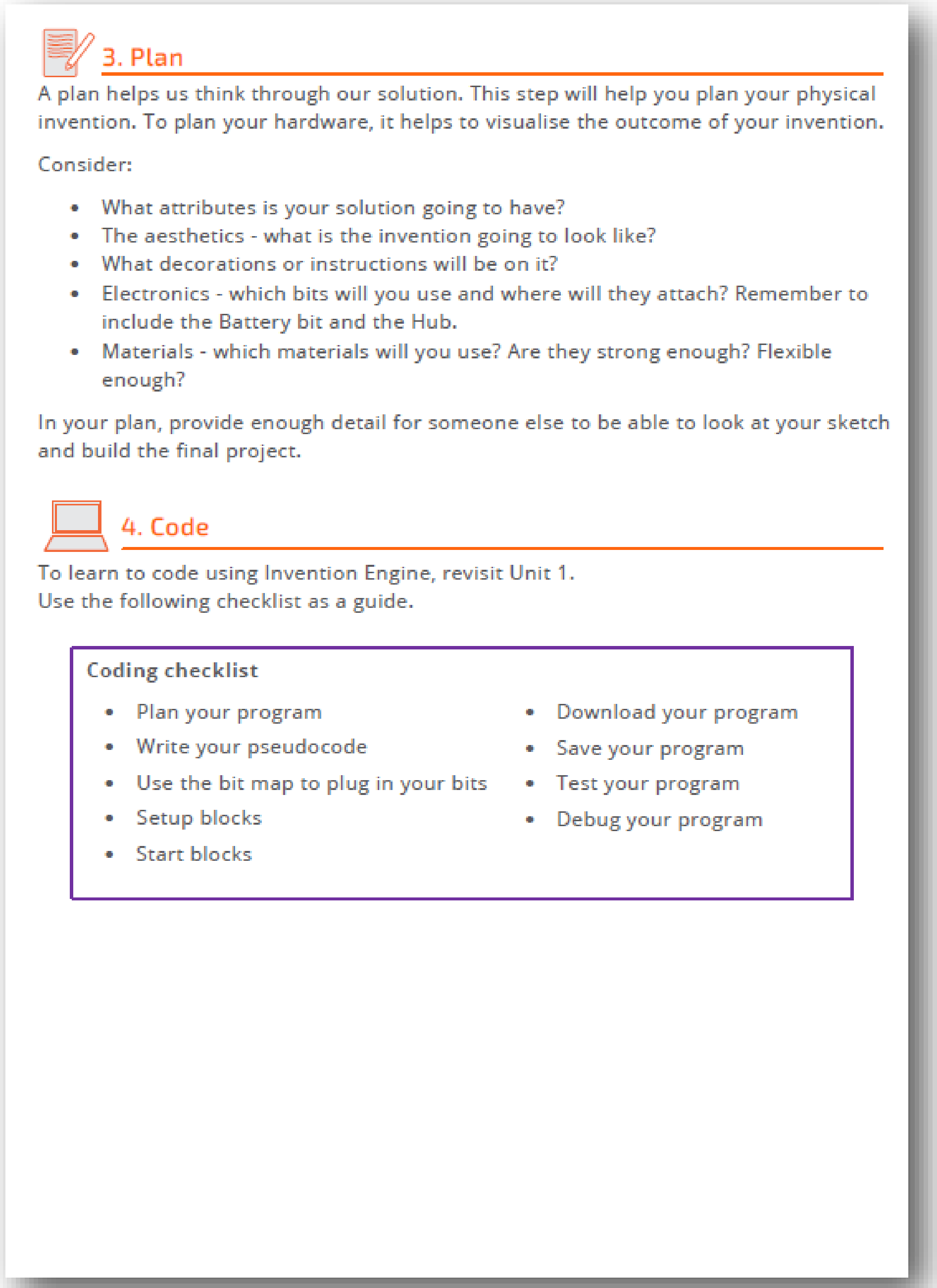
Optional activity: adapt the sketch requirements to suit your class and/or cross-curricular activity. Include prototyping sketches from different angles, or a story board set of sketches to illustrate the way the user would interact with the invention.

4. Code

Ask students to follow the prompts in the checklist and the Invention journal.

Remind your class to test, debug and finalise the code before attaching the bits to cardboard.

Tip: Lean on the skills developed in Unit 1 to code and debug. It is helpful to keep the Getting started with Invention Engine document at hand.



3. Plan

A plan helps us think through our solution. This step will help you plan your physical invention. To plan your hardware, it helps to visualise the outcome of your invention.

Consider:

- What attributes is your solution going to have?
- The aesthetics - what is the invention going to look like?
- What decorations or instructions will be on it?
- Electronics - which bits will you use and where will they attach? Remember to include the Battery bit and the Hub.
- Materials - which materials will you use? Are they strong enough? Flexible enough?

In your plan, provide enough detail for someone else to be able to look at your sketch and build the final project.

4. Code

To learn to code using Invention Engine, revisit Unit 1.
Use the following checklist as a guide.

Coding checklist

- Plan your program
- Write your pseudocode
- Use the bit map to plug in your bits
- Setup blocks
- Start blocks
- Download your program
- Save your program
- Test your program
- Debug your program

5. Make

Start building! For a refresher on how to attach the bits to cardboard, revisit the Getting started with Invention Engine document.

Construct, glue, deconstruct, tinker with your cardboard until satisfied with your cardboard construction. Once you are ready add the Invention Engine bits to the build.

Remind students about:

- safety: remember the inner punch tool is sharp!
- the constraints and specifications for the invention.
- the process: the iteration process may begin early if students discover that their design "on paper" does not work when they try to build it in real life.
- keep notes on iterations and the reasons why an iteration was made, and how it worked (or didn't.).

Tips:

- Offer abundant cardboard and include different sizes and shapes. Keeping enough cardboard in stock and reminding students that the bits are reusable will reinforce that they can make different iterations, and not worry about making mistakes.
- Agree on a timeline for your class with defined 'project goals' or milestones.
- Set a 'work in progress' area to store the projects. Individual project trays are helpful in keeping all the bits and developing inventions.

5. Make

When we make inventions using Invention Engine, we follow the checklist below.

Make checklist

Construct

- Choose your cardboard
- Draw on cardboard
- Measure twice, cut once
- Perforate the external bit shape
- Perforate the rivet holes
- Secure the bits
- Fix and tinker

Combine

- Create a grid
- Use the bit guide

Beautify

- Make it awesome!

Construct with cardboard:

- Choose a cardboard box that is right for your invention. Study the size, robustness, and thickness of the box.
- Draw the outline of the project on cardboard.
- Cut out the outline of the project if needed (measure twice, cut once).
- Remember to consider the specifications and constraints for the invention.

Combine: See the Getting Started with Invention Engine guide for detailed instructions on how to add the bits to cardboard.

Fix and tinker: You may find yourself in the test – fix loop a few times, and that is ok! This loop is the same when you are fixing the code, the hardware or when you when the electronics and the carboard are combined.

Focus on the learning and enjoy the process.

- Take a deep breath! Remember that tinkering and fixing are both essential!
- Find the problem, then fix it. Start by narrowing down the source of the fault by decomposing the problem. Is the problem in the hardware or the code? If it is the code, where in the code is it? Etc.
- Track changes. Update the design and sketch to include the modifications made to your plan.
- Ask fellow inventors or teachers for their feedback and/or help.

Keep tinkering and adjust it until you feel satisfied.

Beautify: and make it awesome! Decorate and add instructions to your invention. Use creativity to make the invention intuitive to use and add your personal touch to make it yours.

6. Test

Testing is a great opportunity to add a cross curricular link to the project. Tailor the Invention journal to include:

- Dependent and independent variables
- Making graphs to illustrate data
- Interpreting data
- Data verbal, graphic, numerical, and symbolic representations
- Testing and predicting results

7. Iterate

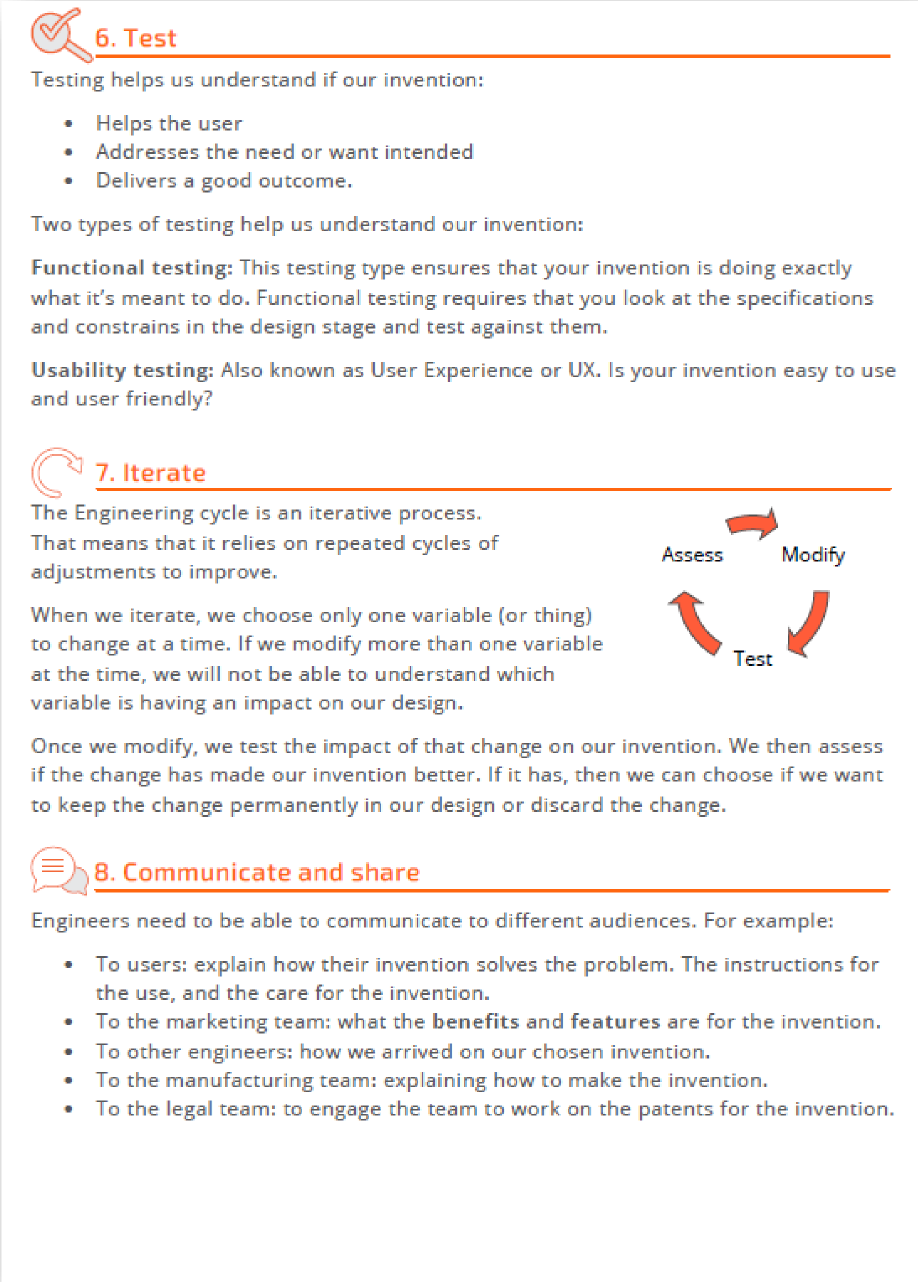
Students will engage in a few different testing loops, including the iterative loop. Once students are comfortable with their invention ask students to collaborate to test each other's inventions and provide and receive feedback.

If possible, direct students to test with the intended user for whom the invention was created for. Observe the invention in the 'wild' and ask questions to the user about the invention.

8. Communicate and share

Ideally, students have an opportunity to present their work to an audience. Sharing their work engages and motivates students to do their best work, develop empathy, receive, and give feedback, and is an opportunity to experience feeling proud of their work.

If needed, sharing can also be in writing. Choose an audience for students to write to and communicate aspects of their invention to.



6. Test

Testing helps us understand if our invention:

- Helps the user
- Addresses the need or want intended
- Delivers a good outcome.

Two types of testing help us understand our invention:

Functional testing: This testing type ensures that your invention is doing exactly what it's meant to do. Functional testing requires that you look at the specifications and constrains in the design stage and test against them.

Usability testing: Also known as User Experience or UX. Is your invention easy to use and user friendly?

7. Iterate

The Engineering cycle is an iterative process. That means that it relies on repeated cycles of adjustments to improve.

When we iterate, we choose only one variable (or thing) to change at a time. If we modify more than one variable at the time, we will not be able to understand which variable is having an impact on our design.

Once we modify, we test the impact of that change on our invention. We then assess if the change has made our invention better. If it has, then we can choose if we want to keep the change permanently in our design or discard the change.

8. Communicate and share

Engineers need to be able to communicate to different audiences. For example:

- To users: explain how their invention solves the problem. The instructions for the use, and the care for the invention.
- To the marketing team: what the **benefits** and **features** are for the invention.
- To other engineers: how we arrived on our chosen invention.
- To the manufacturing team: explaining how to make the invention.
- To the legal team: to engage the team to work on the patents for the invention.

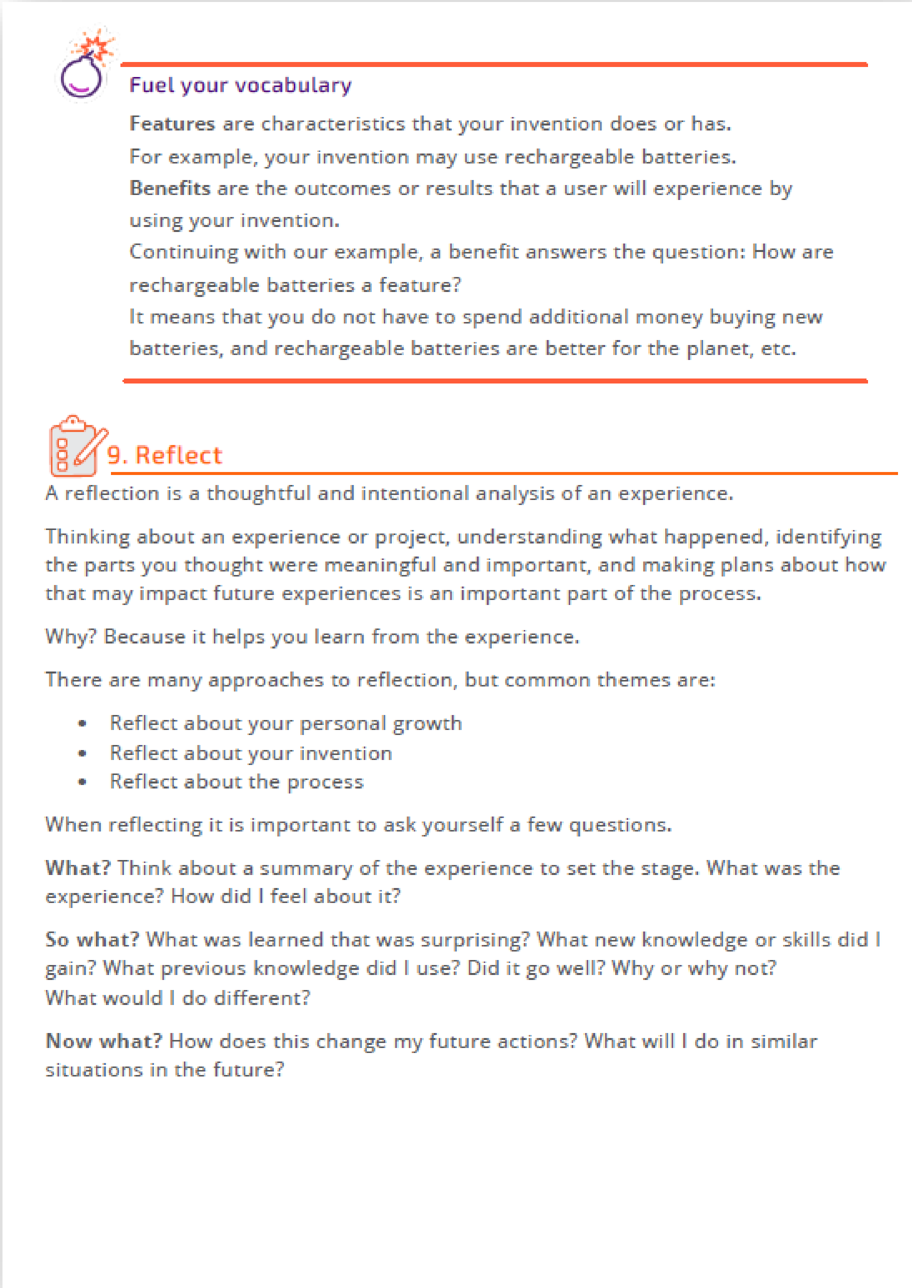
9. Reflect

Model reflection by sharing your own reflection about the process. Offer what you learnt, discuss the mistakes you made, the lessons learnt from the mistakes and what you will do differently next time.

Consider giving students the option to draw their reflections, or mind-map them, or to create a video blog about their project.

Reflection is an important part of the process because it helps to:

- Consolidate the learning, by consolidating the learning and moving the information from short-term to long-term memory.
- Identify gaps where the student may not have a strong grasp of the material.
- Problem solving: coding, engineering and design thinking often involves overcoming challenges and solving problems. Reflecting on the approach to solving past problems can lead to improved problem- solving skills in future projects.
- Learning from mistakes, analysing and reflecting on mistakes are opportunities to grow. Thinking about what went wrong and why can avoid repeating whose mistakes and develop a deeper understanding of the code.
- Tracking progress, it helps students see how much they have learned, how their skills have improved and thus, develop confidence in their ability to learn.



Fuel your vocabulary

Features are characteristics that your invention does or has.
For example, your invention may use rechargeable batteries.
Benefits are the outcomes or results that a user will experience by using your invention.
Continuing with our example, a benefit answers the question: How are rechargeable batteries a feature?
It means that you do not have to spend additional money buying new batteries, and rechargeable batteries are better for the planet, etc.

9. Reflect

A reflection is a thoughtful and intentional analysis of an experience.

Thinking about an experience or project, understanding what happened, identifying the parts you thought were meaningful and important, and making plans about how that may impact future experiences is an important part of the process.

Why? Because it helps you learn from the experience.

There are many approaches to reflection, but common themes are:

- Reflect about your personal growth
- Reflect about your invention
- Reflect about the process

When reflecting it is important to ask yourself a few questions.

What? Think about a summary of the experience to set the stage. What was the experience? How did I feel about it?

So what? What was learned that was surprising? What new knowledge or skills did I gain? What previous knowledge did I use? Did it go well? Why or why not? What would I do different?

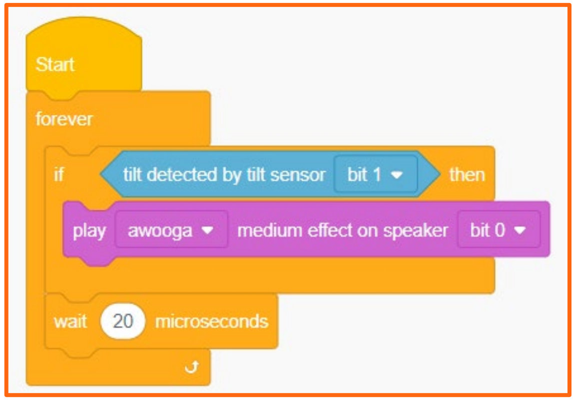
Now what? How does this change my future actions? What will I do in similar situations in the future?

Activity 1: A solution for drowsy driving

Purpose: To practice the steps in the invention cycle by creating an invention. To introduce students to the Tilt sensor bit.

Delivery suggestion:

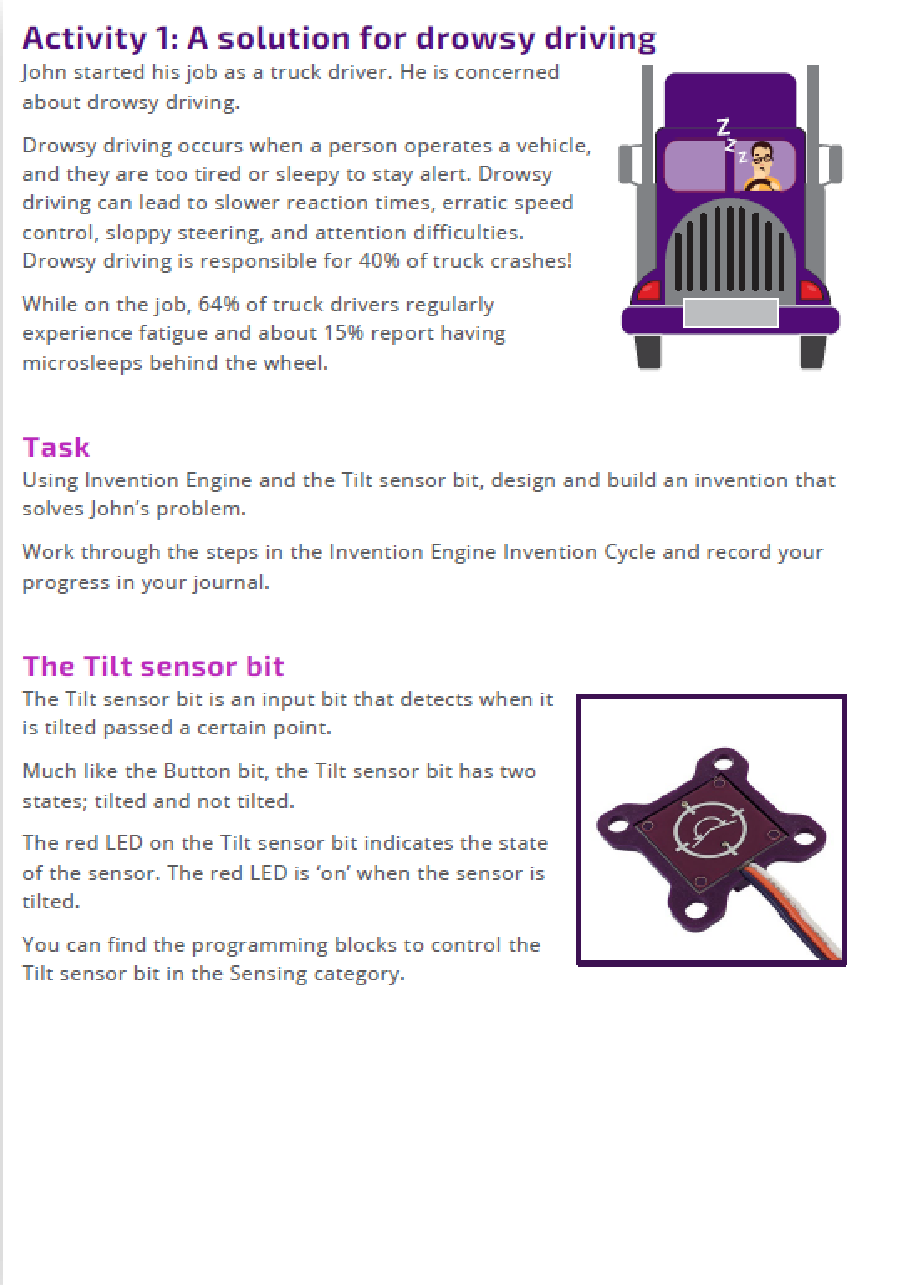
Code: Work with the class to code a sample program for the Tilt sensor bit. In the example below, when the tilt sensor is tilted it will cause the speaker to play an awooga sound. The example uses the 'if-then' block, however, it could also use the 'wait until' control block.



Encourage students to be creative with the code and experiment using additional bits.

Explore the concept of drowsy driving with the class. Ensure that the class understands the task at hand.

Modify the task as needed. Establish the expectations and the milestones for the project. Distribute the Invention Journal and direct students to work through all the steps of the cycle and record their work in the journal.



Activity 1: A solution for drowsy driving

John started his job as a truck driver. He is concerned about drowsy driving.

Drowsy driving occurs when a person operates a vehicle, and they are too tired or sleepy to stay alert. Drowsy driving can lead to slower reaction times, erratic speed control, sloppy steering, and attention difficulties. Drowsy driving is responsible for 40% of truck crashes!

While on the job, 64% of truck drivers regularly experience fatigue and about 15% report having microsleeps behind the wheel.

Task

Using Invention Engine and the Tilt sensor bit, design and build an invention that solves John's problem.

Work through the steps in the Invention Engine Invention Cycle and record your progress in your journal.

The Tilt sensor bit

The Tilt sensor bit is an input bit that detects when it is tilted passed a certain point.

Much like the Button bit, the Tilt sensor bit has two states; tilted and not tilted.

The red LED on the Tilt sensor bit indicates the state of the sensor. The red LED is 'on' when the sensor is tilted.

You can find the programming blocks to control the Tilt sensor bit in the Sensing category.

Activity 2: Mapping and divergent thinking

Purpose: to learn the computer science concept of mapping. To gain insight into divergent thinking and the role of serendipity in engineering and creating inventions. To learn about the Dial bit.

Cross-curricular links:

Maths: explore unit conversions. From the United States customary units, to the metric units. A good example is looking at a ruler because it is mapping from cm to inches.

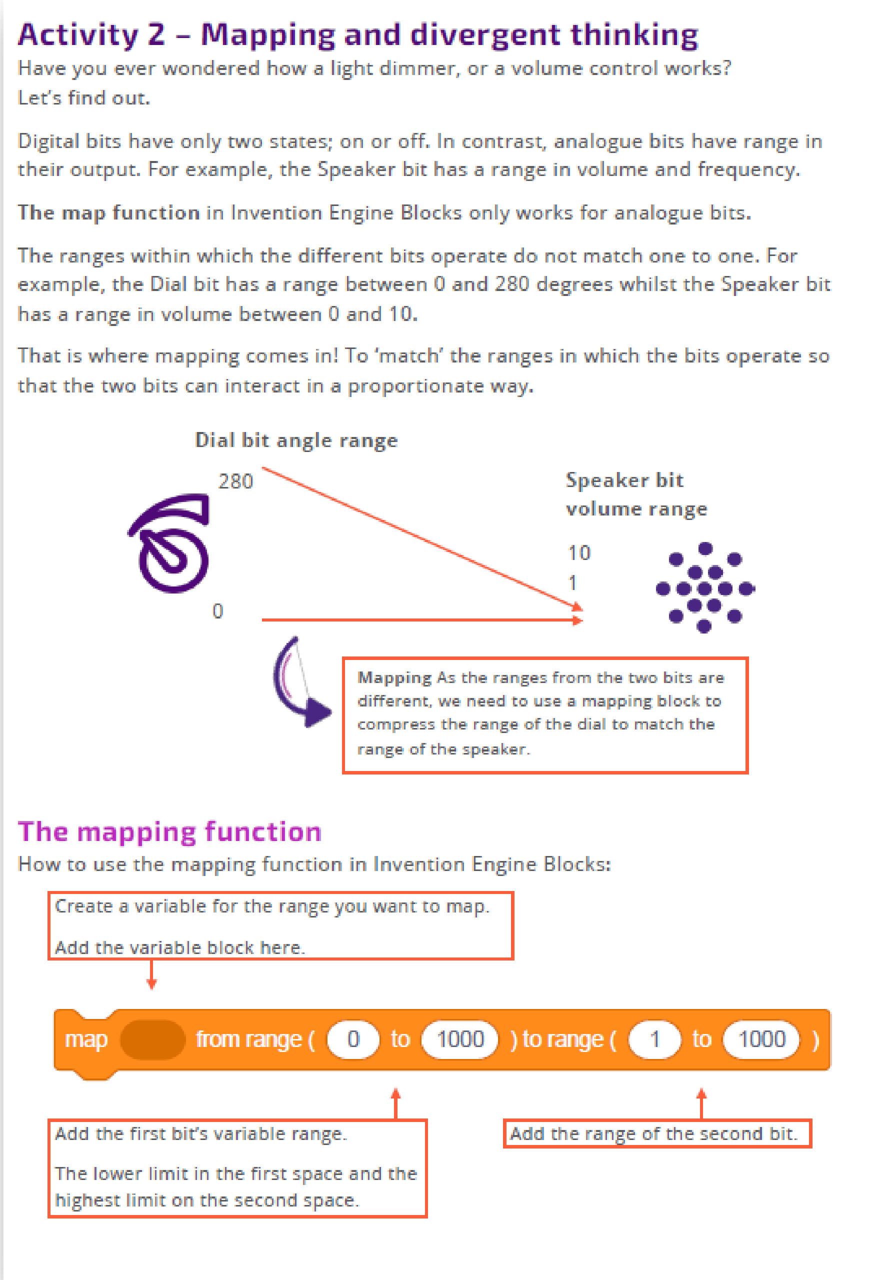
Maths: angles using the potentiometer

Science/biology: explore the concept of ranges of tolerance in living organisms. Explore salinity, temperature ranges, etc.

Delivery suggestion:

Tip: The concept of mapping hinges on students understanding these concepts:

- analogue vs digital: digital signals only have specific values; for example; a switch is either on or off. Analogue signals are continuous and can have many different values, for example; volume. Therefore, only analogue signals can be mapped.
- range: the range of an electronic bit is the boundaries within which that it can function properly. For example, the range of a speaker is about how softly and loudly it can play different sounds.
- variables: Unit 1 lesson 6, page 46.



Activity 2 – Mapping and divergent thinking

Have you ever wondered how a light dimmer, or a volume control works? Let's find out.

Digital bits have only two states; on or off. In contrast, analogue bits have range in their output. For example, the Speaker bit has a range in volume and frequency.

The map function in Invention Engine Blocks only works for analogue bits.

The ranges within which the different bits operate do not match one to one. For example, the Dial bit has a range between 0 and 280 degrees whilst the Speaker bit has a range in volume between 0 and 10.

That is where mapping comes in! To 'match' the ranges in which the bits operate so that the two bits can interact in a proportionate way.

Dial bit angle range

280

0

Speaker bit volume range

10

1

Mapping As the ranges from the two bits are different, we need to use a mapping block to compress the range of the dial to match the range of the speaker.

The mapping function

How to use the mapping function in Invention Engine Blocks:

Create a variable for the range you want to map.

Add the variable block here.

map [ ] from range (0 to 1000) to range (1 to 1000)

Add the first bit's variable range.

The lower limit in the first space and the highest limit on the second space.

Add the range of the second bit.

Delivery suggestion:

Ask students to program Invention Engine to replicate the code in the student guide.

Checking for understanding:

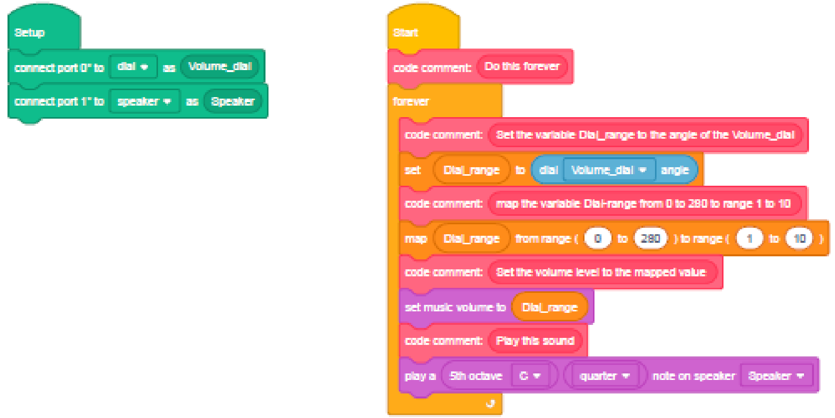
The code in the example will allow the Dial bit to control the volume played by the Speaker bit in Invention Engine.

Once successful, ask students to use the information on the table in the student guide to play and tinker with different bits, code combinations and blocks.

Optional activity: challenge your students to code a light dimmer (Dial bit + LED bit).

Tip: Tinkering and playing with the code will increase your students' confidence to challenge themselves in future programs.

What do you think Invention Engine will do if you run this program?



To map other bits, you can find the ranges for additional bits in the table below.

Name	Input range
Move servo (degrees)	0 -> 180
Rotate motor (SPEED)	0 -> 10
Set LED to level	0 -> 100
Display number on digital display	-9999 -> 30000
Display temperature on digital display	-99 -> 999
Display level on digital	0 -> 5
Send IR data	0 -> 255
Play a tone (Hz)	150 -> 5000
Set music tempo	60->250
Set music volume	1 -> 10

Let's invent Activity 2: A serendipitous invention using the dial bit

Delivery suggestion:

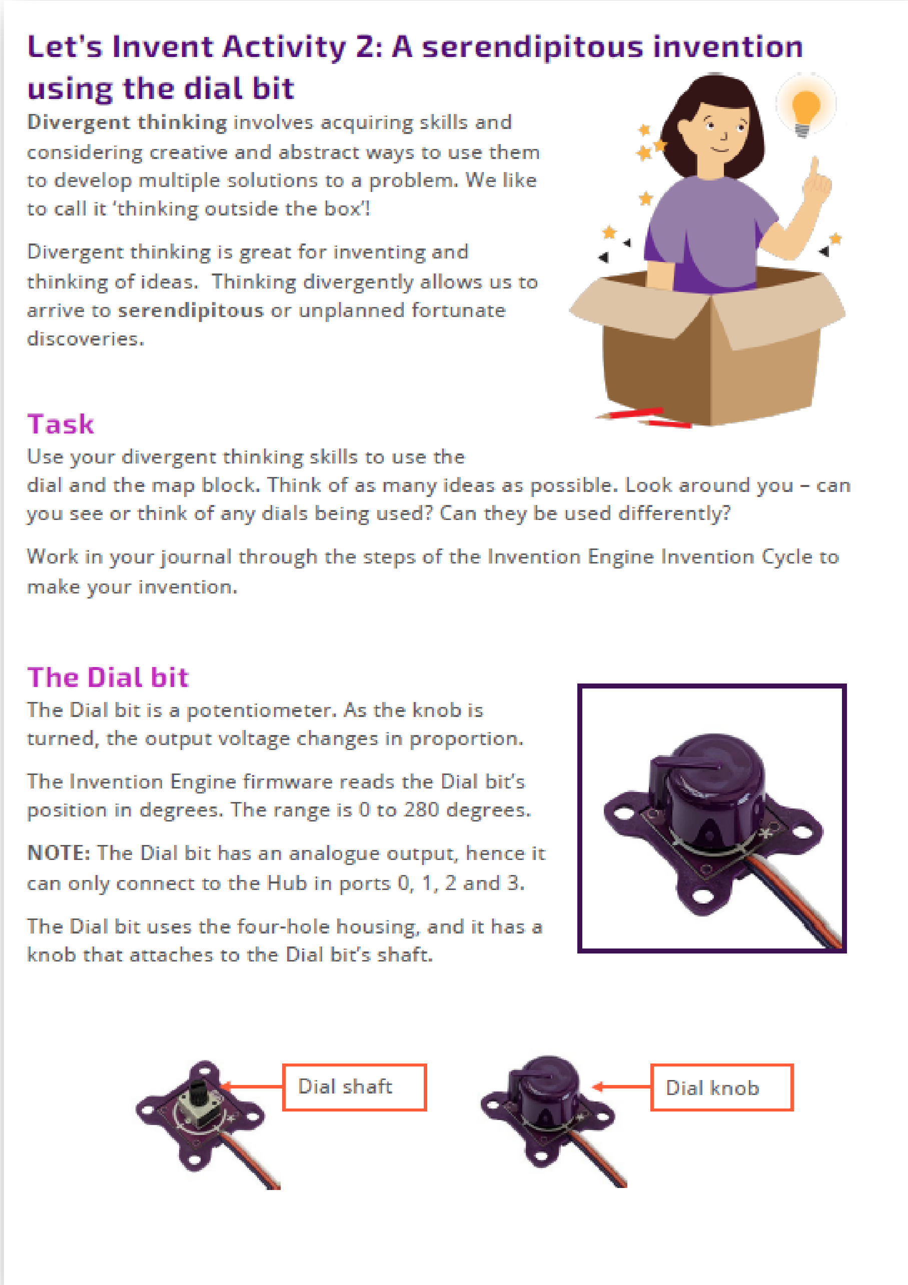
Consider divergent thinking vs. convergent thinking which focuses on finding one well-defined solution to a problem. Discuss tasks that would benefit from one strategy of thinking or the other.

Delve into the concept of serendipitous inventions. Find some interesting serendipitous inventions, one example is Play-Doh.

The doh in Play-Doh was used to remove dirt from wallpaper. When charcoal and wood for heating declined in use, Play-Doh became a new way to re-purpose the product.

Task:

- Ensure that the class understands the task at hand.
- Modify the task as needed. Establish the expectations and the milestones for the project.
- Distribute the Invention Journal and direct students to work through all the steps of the cycle and record their work.
- Learn about the Dial bit by consulting the datasheet.
- Some examples of serendipitous inventions to consider (if stuck for ideas) for this activity are:
  - A colour-intensity changing mood light that changes colour based on the position of the dial.
  - A theremin-like instrument, a simple musical instrument that generates different tones based on the position of the dial. As the user turns the dial, the frequency of the sound changes producing a theremin-like effect.



Let's Invent Activity 2: A serendipitous invention using the dial bit

Divergent thinking involves acquiring skills and considering creative and abstract ways to use them to develop multiple solutions to a problem. We like to call it 'thinking outside the box'!

Divergent thinking is great for inventing and thinking of ideas. Thinking divergently allows us to arrive to **serendipitous** or unplanned fortunate discoveries.

Task

Use your divergent thinking skills to use the dial and the map block. Think of as many ideas as possible. Look around you – can you see or think of any dials being used? Can they be used differently?

Work in your journal through the steps of the Invention Engine Invention Cycle to make your invention.

The Dial bit

The Dial bit is a potentiometer. As the knob is turned, the output voltage changes in proportion.

The Invention Engine firmware reads the Dial bit's position in degrees. The range is 0 to 280 degrees.

NOTE: The Dial bit has an analogue output, hence it can only connect to the Hub in ports 0, 1, 2 and 3.

The Dial bit uses the four-hole housing, and it has a knob that attaches to the Dial bit's shaft.

Activity 3 – Snippets and movement

Cross-curricular links:

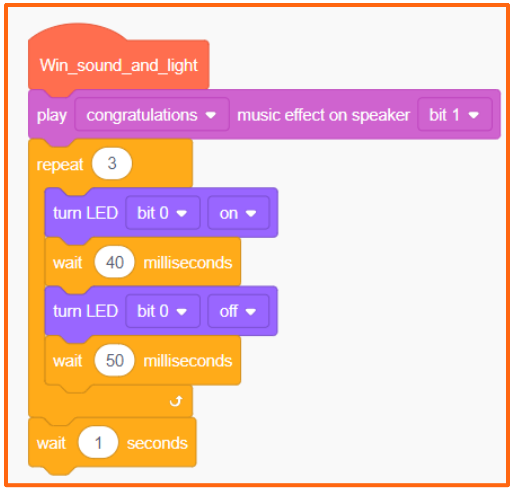
- Social Studies: history, changes through time, cultural traditions.
- Maths: angles, degrees, trigonometry, patterns – fractals
- Music: creating your own melody, tempo, notes.

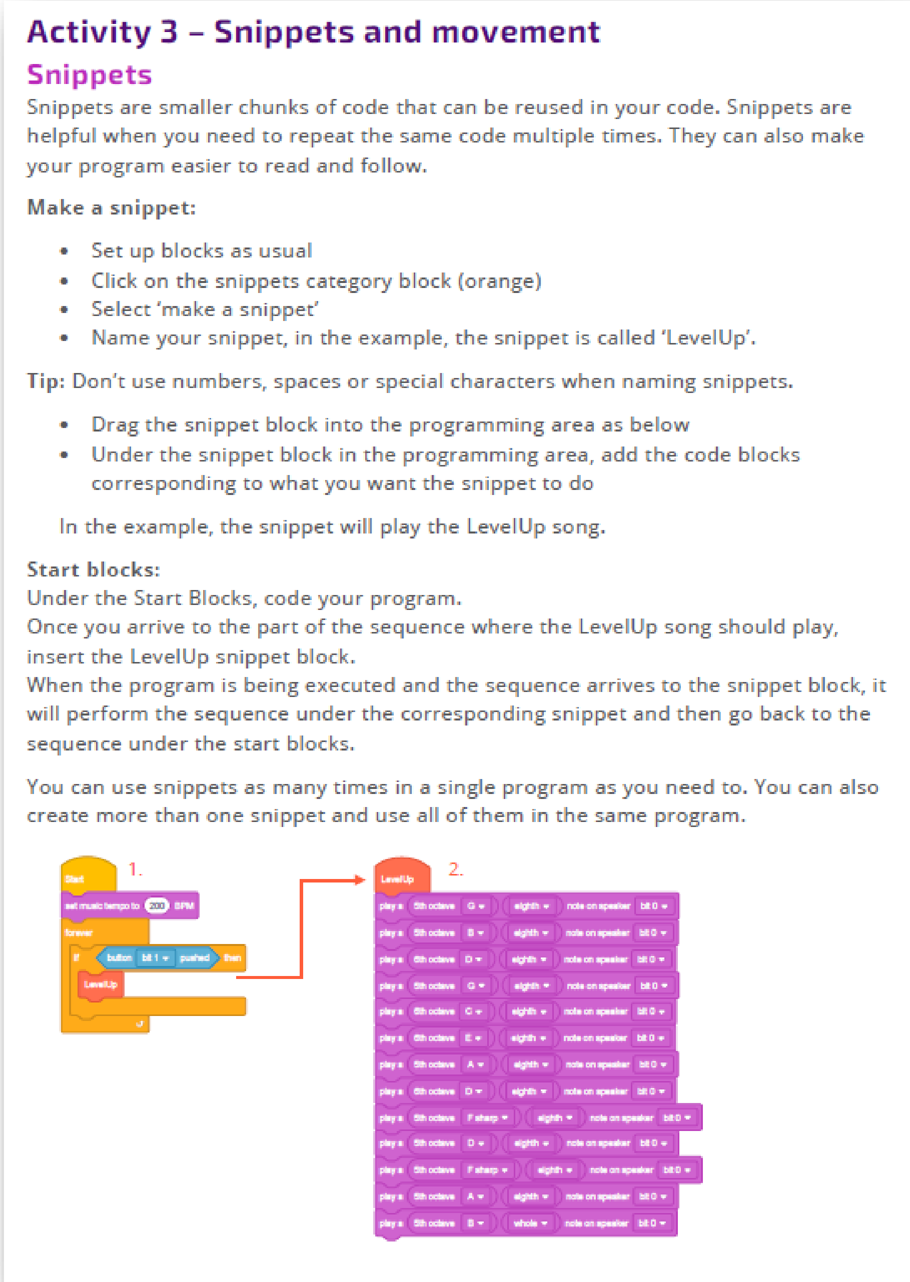
Purpose: to explore efficiency in coding through the concept of snippets. To introduce the servo to add movement to inventions, and the Magnet sensor and Magnet bits.

Delivery suggestion:

Ask students to imagine building a LEGO castle. However, instead of having to make each tower from individual blocks, using pre-built blocks groups that snap together to create the towers quickly. Snippets are like pre-built blocks for coding; a short set of instructions that someone else has written to do something specific.

The example in the student's guide has instructions to play a song. This short example below, is for a snippet to play a 'win' sound-light combo in a game.





Activity 3 – Snippets and movement

Snippets

Snippets are smaller chunks of code that can be reused in your code. Snippets are helpful when you need to repeat the same code multiple times. They can also make your program easier to read and follow.

Make a snippet:

- Set up blocks as usual
- Click on the snippets category block (orange)
- Select 'make a snippet'
- Name your snippet, in the example, the snippet is called 'LevelUp'.

Tip: Don't use numbers, spaces or special characters when naming snippets.

- Drag the snippet block into the programming area as below
- Under the snippet block in the programming area, add the code blocks corresponding to what you want the snippet to do

In the example, the snippet will play the LevelUp song.

Start blocks:
Under the Start Blocks, code your program.
Once you arrive to the part of the sequence where the LevelUp song should play, insert the LevelUp snippet block.
When the program is being executed and the sequence arrives to the snippet block, it will perform the sequence under the corresponding snippet and then go back to the sequence under the start blocks.

You can use snippets as many times in a single program as you need to. You can also create more than one snippet and use all of them in the same program.

Delivery suggestion:

This is a very exciting lesson! It introduces two bits and movement for the first time.

Explore the Magnet bits and the Servo bit in the accompanying datasheets for the bits.

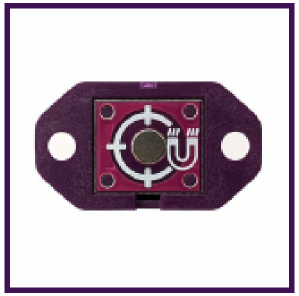
Ask your students to practice putting the servo horn on the servo shaft and removing it.

It can be hard to imagine how to use the servo in an invention. In the picture, the arm is being powered by the servo but only the four rivets attaching the servo horn are visible, because the servo is inside.

Tip: The Magnet bit is small and does not have a cable, ask your students to be careful when storing it and to keep it with the Magnet sensor to avoid losing it. If it is lost, it can be replaced by a different magnet.

The Magnet bits

There are two Magnet bits, one is a Magnet sensor bit and the other is a Magnet bit.



The Magnet sensor bit has a red LED which turns on when the Magnet bit is in close proximity (approx. 10mm). Larger and more powerful magnets are detected at greater distances.

The Magnet bit does not require a connection to the Invention Engine Hub.

The Magnet sensor bit and Magnet bit use the two-hole housing. There are no protruding parts, so there is no need to cut out the bit shape in the carboard.

The Magnet bit has only two states. Either a magnet is detected, or it is not detected. Coding using the Magnet sensor bit is like coding with the Button bit.

A use of the Magnet sensor bit is to detect the status of a door, window, drawer, or lid (opened or closed).

The Servo bit

The Servo bit has two parts. The servo and the horn.

The horn attaches to the servo shaft by press fitting the shaft into the servo horn fitting (gently push it in).

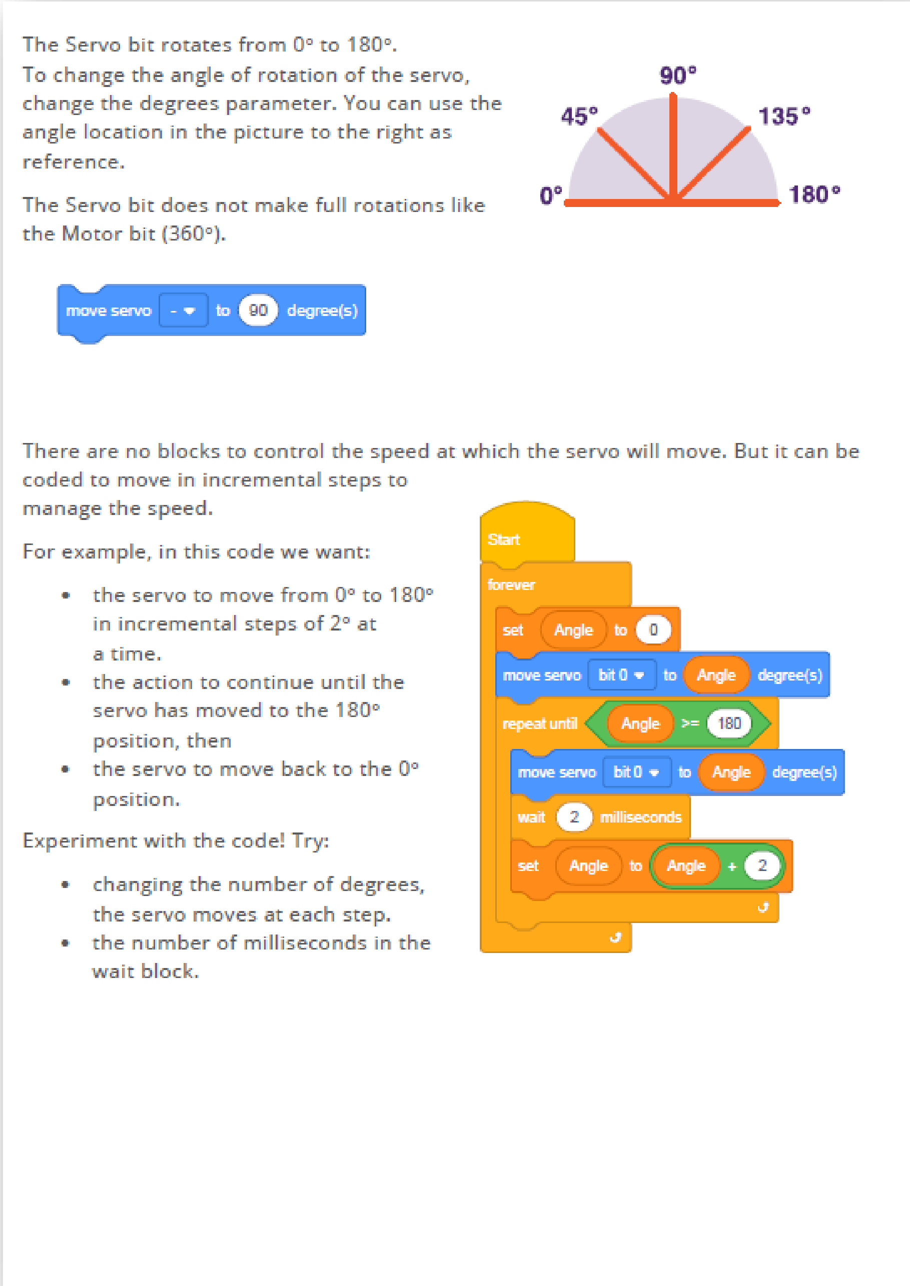
Use the stencil to punch the outline of the Servo bit's housing. The Servo bit uses the two-hole housing, and the servo horn uses four.

Attach the servo horn to the part of the cardboard you want to move and the servo to the part of the cardboard that will not move.

Delivery suggestion:

- Work with the class to code the Servo bit by following along step by step with the student guide.
- Review the concept of angles and degrees if needed.
- Allow for plenty of time to 'play' with the servo. Challenge the class to complete the 'experiment with the code' part at the bottom of the page.

Tip: To make the movement of the servo easier to see, attach a piece of paper to the servo shaft or the servo horn.



The Servo bit rotates from 0° to 180°.
To change the angle of rotation of the servo, change the degrees parameter. You can use the angle location in the picture to the right as reference.

The Servo bit does not make full rotations like the Motor bit (360°).

There are no blocks to control the speed at which the servo will move. But it can be coded to move in incremental steps to manage the speed.

For example, in this code we want:

- the servo to move from 0° to 180° in incremental steps of 2° at a time.
- the action to continue until the servo has moved to the 180° position, then
- the servo to move back to the 0° position.

Experiment with the code! Try:

- changing the number of degrees, the servo moves at each step.
- the number of milliseconds in the wait block.

Let's Invent Activity 3: A music box using the Servo and the Magnet bit

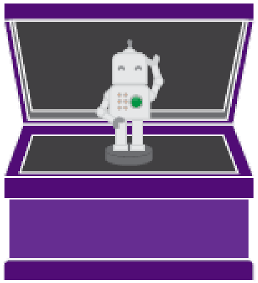
Delivery suggestion:

- Ensure that the class understands the task at hand.
- Modify the task as needed. Establish the expectations and milestones for the project.
- Distribute the Invention Journal and direct students to work through all the steps of the cycle and record their work in the journal.

Tips:

- Use your search engine to look for "cardboard working techniques" to discover many options and ideas to build, attach and use cardboard.
- Build: this build may be more challenging if a lid or top is included in the design and the 'character' needs to spring into action when the box is opened, and collapse when the lid is closed.
- An alternative is to use the magnet as a 'key' to manually awake the character without the use of a lid.

Let's Invent Activity 3: A music box using the Servo and the Magnet bit



In the 18th century, there were no radios, or inventions to play music. Sensors, electronics, and electricity were all yet to be invented.

Music boxes were often carried by gentlemen in their waist coat to play music and delight their friends.

Then, music boxes were powered by clockwork. Therefore, restricted in size, quiet and very basic. Modern music boxes continue to delight us in different ways. Think of music jewellery boxes, baby mobiles and alarms.

Task

Using Invention Engine, let's invent an electronic music box to delight our friends with.

You must use: Snippets, the Magnet bits, and the servo in your invention.

Work in your journal through the steps of the Invention Engine Invention Cycle to make your invention.

Activity 4 -Boolean expressions and exploring light sensitivity

Purpose: To learn about 'Boolean expressions' and how to apply the expression to use sensors in computer science.

To learn about the Light sensor bit.

Cross-curricular links:

English, Politics, Debate, Philosophy: learn about syllogisms to combine two true statements to produce a new conclusion.

Maths: Use Boolean expressions to create and analyse Venn diagrams, helping students to visualize the relationships between sets.

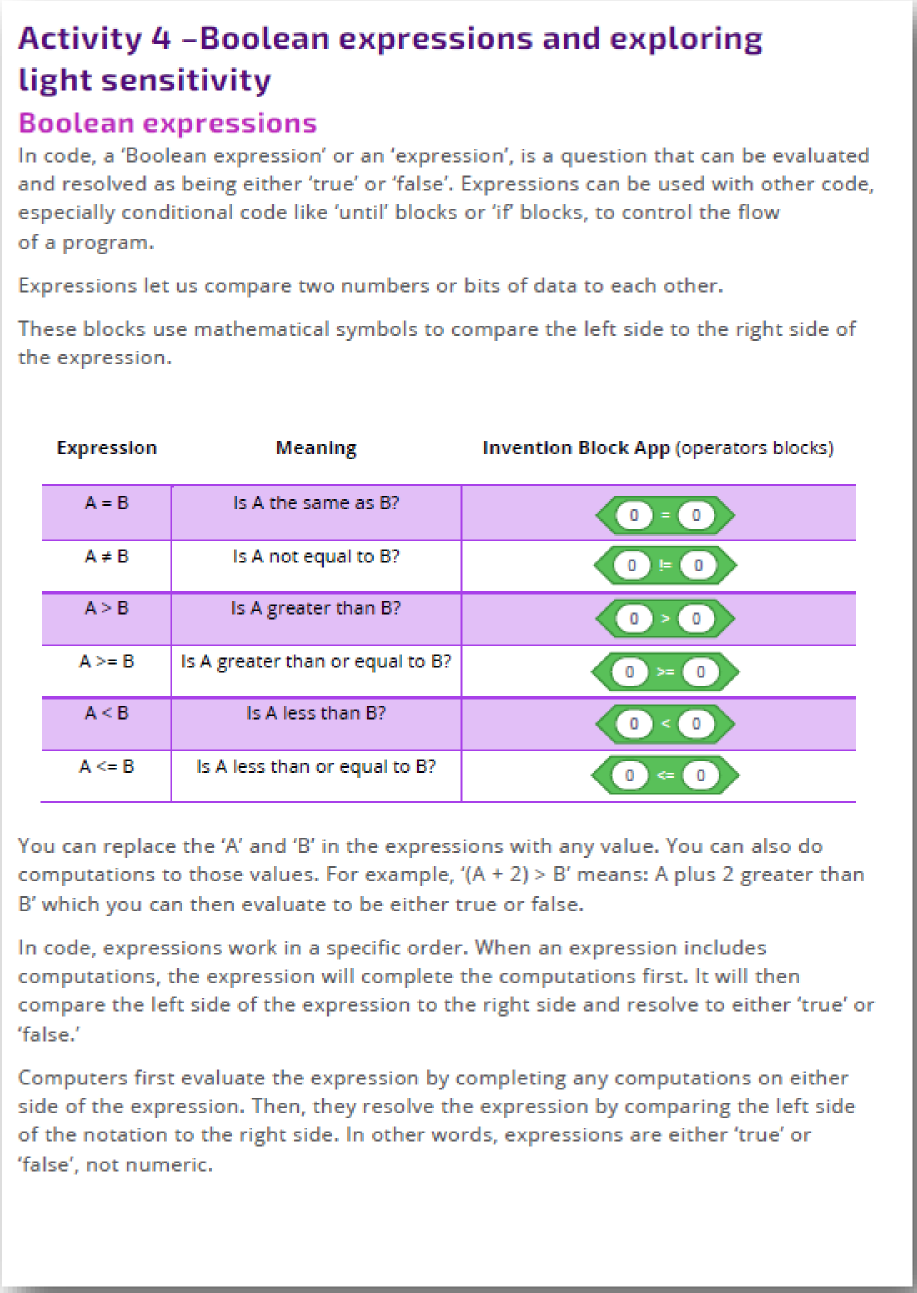
Biology/science: explore animal's features to maximise their environment, including light sensitivity. Explore genetics, evolution, and the role of diurnal and nocturnal animals in the ecosystem.

Delivery suggestion:

Use examples to review the meaning of true and false. For example: The grass is green, the moon is made of cheese.

Introduce Boolean operators: use words like "and," "or," "not," "same," "not equal to," "greater," "larger."

Combine operators: Once the students understand the individual operators, show how they can be combined. For example, "I have a cat OR a dog AND it is raining" (true) or "a cat is NOT EQUAL to a dog OR a horse.



Activity 4 –Boolean expressions and exploring light sensitivity

Boolean expressions

In code, a 'Boolean expression' or an 'expression', is a question that can be evaluated and resolved as being either 'true' or 'false'. Expressions can be used with other code, especially conditional code like 'until' blocks or 'if' blocks, to control the flow of a program.

Expressions let us compare two numbers or bits of data to each other.

These blocks use mathematical symbols to compare the left side to the right side of the expression.

Expression	Meaning	**Invention Block App** (operators blocks)
A = B	Is A the same as B?	0 = 0
A ≠ B	Is A not equal to B?	0 != 0
A > B	Is A greater than B?	0 > 0
A >= B	Is A greater than or equal to B?	0 >= 0
A < B	Is A less than B?	0 < 0
A <= B	Is A less than or equal to B?	0 <= 0

You can replace the 'A' and 'B' in the expressions with any value. You can also do computations to those values. For example, '(A + 2) > B' means: A plus 2 greater than B' which you can then evaluate to be either true or false.

In code, expressions work in a specific order. When an expression includes computations, the expression will complete the computations first. It will then compare the left side of the expression to the right side and resolve to either 'true' or 'false.'

Computers first evaluate the expression by completing any computations on either side of the expression. Then, they resolve the expression by comparing the left side of the notation to the right side. In other words, expressions are either 'true' or 'false', not numeric.

Delivery suggestion:

Provide the students with simple statements and ask them to determine if they are true or false. You can gradually introduce more complex Boolean expressions as the class becomes comfortable with the concept.

Checking for understanding:

A>=B
Meaning: A is equal or greater than B. Or, A>= B or 2 >= 4
Resolves to: False, because 2 is not equal or greater than 4

(A-1) < (B-3)
Meaning: (A-1) is less than (B-3). Or, (2-1) < (4-1)
Resolves to: False, because 1 is not less than 3

Explore the Light sensor bit, look at the accompanying datasheet. Think of some inventions that may use a sensor bit. For example: Automatic night lights, solar powered garden lights, automatic blinds, smart street lights, etc.

Tip: When coding the Light sensor bit, it is important to remember that it is an analogue bit, and therefore it will only work on ports 0, 2 and 3.

Try this - For these expressions, if A = 2 and B = 4, what does each of the following expressions mean (in other words, what is it trying to state) and what does each expressional statement resolve to (true or false)?

(A*2) = B
Meaning: 2*2 =4 (because A=2 and B=4)
Resolves to: True (because the two values are equal)

A >= B

Meaning: ______________________

Resolves to: ______________________

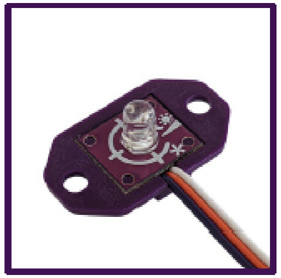
(A-1) < (B-3)
Meaning: ______________________

Resolves to: ______________________

Understanding what expressions mean and what they resolve to will help you follow a program just by 'reading' it. This important skill in programming is known as **tracing**.

The Light sensor bit

The Light sensor reads light levels between 0 to 100.



The Light sensor bit has an analogue output it can only be used at ports 0, 1, 2 and 3, which are the Invention Engine's analogue inputs.

The Light sensor is used in inventions such as an alarm that detects that the fridge door has been opened, or an invention that reacts when the sun has gone down.

The Light sensor bit almost looks identical to the LED bits, however, the sensor called a 'phototransistor' has a water clear lens.

The Light sensor bit works by comparing the light levels. For example, if it gets darker than *this* number, then do *this* action. This comparison is possible by using Boolean expressions.

Let’s Invent Activity 4: A nocturnal or diurnal creature

Delivery suggestion:

- Ensure that the class understands the task at hand.
- Modify the task as needed. Establish the expectations and the milestones for the project.
- Distribute the Invention Journal and direct students to work through all the steps of the cycle and record their work.

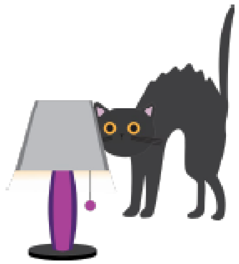
Review the concept of nocturnal (night) and diurnal (day) creatures. Encourage students to think of features that relate to the animal’s lifestyles and how they are connected to the ecosystem.

Some features to consider are eyesight, coloration, camouflage, body temperature regulation, activity patterns, enhanced smell and hearing and the use of new senses like echolocation.

Some examples for nocturnal creatures are of serendipitous inventions to consider (if stuck for ideas) for this activity are:

- A bat: use one LED and the light sensor in the eyes, the servo for the wing movement.
- Invent a mythological creature or a monster.
- Create a carnivorous plant that turns using the servo towards or away from the light.

Let's Invent Activity 4: A nocturnal or diurnal creature



Living organisms thrive at different light levels.
Some are nocturnal (active during night-time) and some diurnal (active during daytime), and they have evolved features to thrive in either low or high light environments.

For example, some animals are photophobic.
Do you know the movie Gremlins? The creatures in it are photophobic, or light fearing. On the other hand, most plants are photophilic, or light loving.

Can you think of one animal that is either? Think of their behaviour and their reaction to light. How have they adapted to their environment?

Task

Using Invention Engine, let’s invent an electronic creature that reacts to light.
It can be a companion animal or pet, or an animal that you like.

Work in your journal through the steps of the Invention Engine Invention Cycle to make your invention.

Activity 5: Random numbers and the interactive book

Purpose: To learn how to use random numbers in computer science. This activity includes the final 'Let's invent' project. Students have an opportunity to integrate previous computer science concepts, and use all the bits learnt in this Unit.

Cross-curricular links:

Maths: Introduce concepts of probability using random numbers. Discuss how random numbers are used to model uncertain events and calculate probabilities.

The 'Let's invent' activity can be modified to ask students to make an interactive book about a particular learning objective including History, science, art, health, etc.

Delivery suggestion:

Think of ways computers use random numbers. For example:

- In games, random numbers can determine where enemies appear, what loot you find, or even the outcome of a virtual dice roll.
- In simulations, random numbers can mimic uncertain events, like predicting the weather or the movement of a crowd.
- When creating secure passwords, random numbers make it harder for someone to guess your password.

Ask students to replicate the program in the student guide, and then to change the code to include different output bits.

Activity 5: Random numbers and the Interactive book

Random numbers

Random numbers are chosen by chance, they are randomly selected from a set of numbers. All the numbers in the set of numbers have the same probability of being chosen.

Find the random block in the Operators category in Invention Engine Blocks, and you can use it to ask Invention Engine to choose a random number.

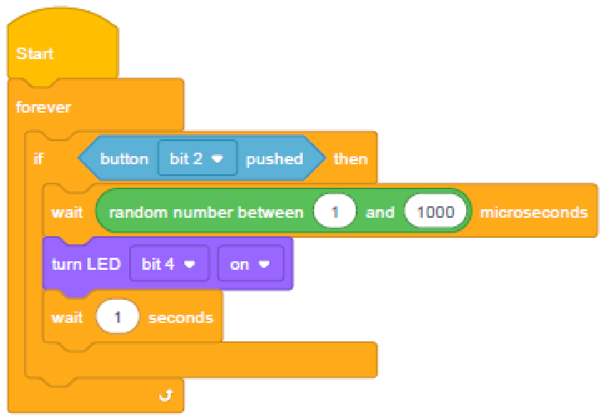
`random number between (1) and (1000)`

You can change the parameters of the random number by editing the lower and upper parameter in the block.

Random numbers are used to select lottery winners, sometimes in games or simulations, in random sampling, in medical trials, etc.

Random numbers can also be fun. For example, a jack-in-the-box is fun because we can't predict when the action will happen.

The code below uses random numbers in the number of milliseconds it will wait after the button push to turn on the light. So, we do not know how long it will be after we push the button for the light to come on.



Let's invent Activity 5: Interactive book

Delivery suggestion:

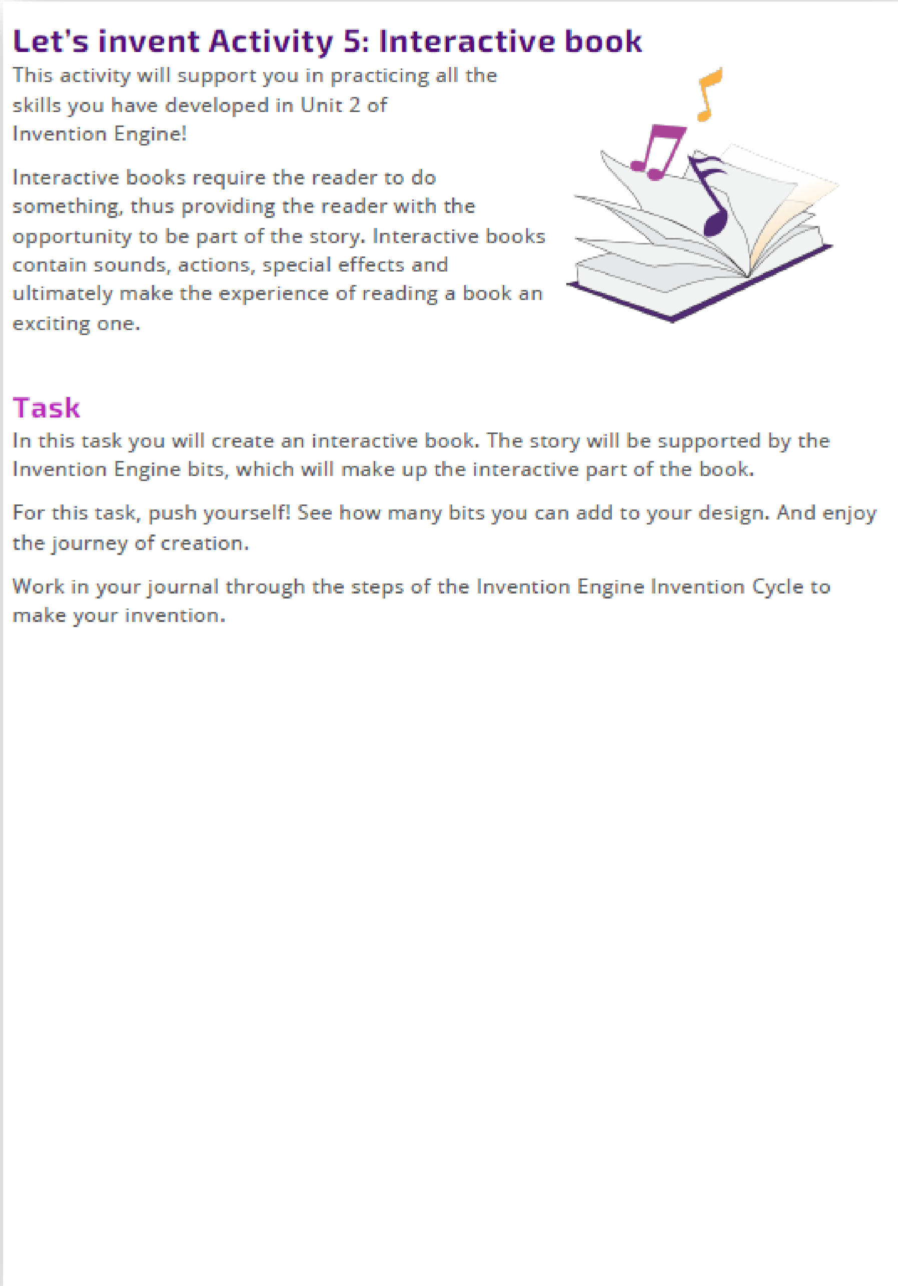
- Ensure that the class understands the task at hand.
- Modify the task as needed. Establish the expectations and the milestones for the project.
- Distribute the Invention Journal and direct students to work through all the steps of the cycle and record their work in the journal.

Use the opportunity to encourage students to research the topic for their book.

As a capstone project, this is an opportunity to introduce project management skills to your students. Consider transferring the agency to the class to set goals, decompose the project into smaller tasks, develop timelines, manage their resources, and monitor and adjust. This option makes for an interesting reflection question about self-improvement.

Tips: modify the project to suit the class:

- Focus on the content and the code and use mixed media, such as cut outs from magazines instead of drawing.
- Reduce the project time by creating a greeting card instead of a book.
- Shift the book design to create an interactive work of art, sculpture, or painting.



Let's invent Activity 5: Interactive book

This activity will support you in practicing all the skills you have developed in Unit 2 of Invention Engine!

Interactive books require the reader to do something, thus providing the reader with the opportunity to be part of the story. Interactive books contain sounds, actions, special effects and ultimately make the experience of reading a book an exciting one.

Task

In this task you will create an interactive book. The story will be supported by the Invention Engine bits, which will make up the interactive part of the book.

For this task, push yourself! See how many bits you can add to your design. And enjoy the journey of creation.

Work in your journal through the steps of the Invention Engine Invention Cycle to make your invention.

Lesson	**The invention Cycle**
Purpose	To introduce your class to the Invention Engine Cycle, the different steps and the associated tools and models applicable within each of the steps.
Resources required	• Student guide and teacher guide
Bits	• None
Computational thinking	• Code • Program • Plan • Test • Debug
Engineering, design thinking, making, tinkering, and inventing skills.	• Define • User; needs and wants • Empathy • Bias • Design • Brainstorm • Constraints and Specifications • Plan • Construct • Tinker • Test • Function • Usability • Feedback • Iterate • Modify • Assess • Communicate • Benefits and Features • Reflect
Cross-curricular links	• Life Sciences: Investigate how engineering can be applied to improve ecosystems, such as designing structures to prevent erosion, or creating habitats for local wildlife. • History of Technology: Explore the historical development of various technologies and how they have shaped societies. Discuss the societal impacts of inventions and innovations. • Art Sketching and Visualization: Teach students how to create detailed sketches and diagrams to communicate their design ideas visually. • Math Dependent and independent variables, making graphs to illustrate data, interpreting data. Data types; verbal, graphic, numerical, and symbolic representations. Testing and predicting results.

Let’s invent activity 1	A solution for drowsy driving
Purpose	To practice the steps in the invention cycle by creating an invention. To introduce students to the Tilt sensor bit.
Resources required	• A programming device with a USB A connection • Invention Engine Hub and LED • Internet connection • Invention Engine building tools; stencil, punch tools and rivets • Cardboard cutters • Scissors • Glue • Batteries • USB extension cable • Markers
Bits	• Tilt sensor bit • Digital display bit • Hub • Button bit • Battery bit • LED bit • Speaker bit • Proximity sensor bit
Computational thinking	• Data • Information • Loops • Conditionals • Variables • Value • User • Increment • Collaboration • Documenting
Engineering, design thinking, making, tinkering, and inventing skills.	• Invention Journal • Invention cycle: define, design, plan, code, make, test, iterate, communicate, and reflect. • Documenting, journaling • Empathy • Research
Cross-curricular links	• Ethics and critical thinking: Responsibilities of operating a vehicle, consequences. • English: persuasive writing advocating for new technology or new regulations. • Math: statistics and data analysis on drowsy driving and fatalities. Correlation variables. • Science: circadian rhythms, sleep, brain function and reaction time. • Health education: sleep hygiene, good sleep habits, strategies for improving sleep quality.

Activity 2	Mapping and divergent thinking
Purpose	To learn the computer science concept of mapping. To gain insight into divergent thinking and the role of serendipity in engineering and creating inventions. To learn about the Dial bit.
Let's invent activity 2	A serendipitous invention using the Dial bit
Resources required	• A programming device with a USB A connection • Invention Engine Hub and LED bit • Internet connection • Invention Engine building tools; stencil, punch tools and rivets • Cardboard cutters • Scissors • Glue • Batteries • USB extension cable • Markers
Bits	• Dial bit • Tilt sensor bit • Digital display bit • Hub • Button bit • Battery bit • LED bit • Speaker bit • Proximity sensor bit
Computational thinking	• Mapping function • Analogue vs digital • Range • Variables • Loops • Conditionals • Debugging • Value • Feature
Engineering, design thinking, making, tinkering, and inventing skills.	• Serendipity • Convergent thinking • Divergent thinking • Mapping blocks • Reading data sheets • Bit ranges • Units of measure
Cross-curricular links	• Maths: explore unit conversions. From the United States customary units to the metric units. A good example is looking at a ruler because it is mapping from cm to inches. • Maths: angles using the potentiometer. • Science/biology: explore the concept of ranges of tolerance in living organisms. Explore salinity, temperature ranges, etc.

Activity 3	**Snippets and movement**
Purpose	To explore efficiency in coding through the concept of snippets. To introduce the servo bit to add movement to inventions, and the Magnet sensor and Magnet bits.
Let's invent activity 3	A music box using the Servo and the Magnet bits
Resources required	• A programming device with a USB A connection • Invention Engine Hub and LED • Internet connection • Invention Engine building tools; stencil, punch tools and rivets • Cardboard and cutters • Scissors • Glue • Markers • Batteries • USB extension cable
Bits	• Servo bit • Magnet bit • Dial bit • Tilt sensor bit • Digital display bit • Proximity sensor bit • Button bit • Battery bit • LED bit • Speaker bit • Hub
Computational thinking	• Snippets and movement • Conditional statements • Outputs and inputs • Algorithms • Flowcharts • True / False • Condition • Abstraction • Creating • Design notes • Sketch • Label • Evaluation • Iteration
Engineering, design thinking, making, tinkering, and inventing skills.	• Snippets category • Movement • Servo bit • Abstract thinking • Angles • Engineering to create moving parts
Cross-curricular links	• Social Studies: history, changes through time, cultural traditions. • Maths: angles, degrees, trigonometry, patterns – fractals. • Music: creating your own melody, tempo, notes. • History of musical inventions, musical reproductions through time.

Activity 4	Boolean expressions and exploring light sensitivity
Purpose	To learn about 'Boolean expressions' and how to apply the expressions to use sensors in computer science.
Let's invent activity 4	A nocturnal or diurnal creature
Resources required	• A programming device with a USB A connection • Invention Engine Hub and LED bit • Invention Engine building tools; stencil, punch tools and rivets • Internet connection • Cardboard and cutters • Scissors / Glue / Markers • USB extension cable • Batteries
Bits	• Light sensor bit • Servo bit • Magnet bit • Dial bit • Battery bit • Tilt sensor • Digital display bit • Hub • Button bit • Proximity sensor bit • LED bit • Speaker bit
Computational thinking	• Boolean expressions • Sensing • Operators • Operators blocks • Conditionals • Resolving expressions • Tracing code • Inputs and outputs
Engineering, design thinking, making, tinkering, and inventing skills.	• Creativity • Using research facts to invent • Light sensitivity
Cross-curricular links	• English, Politics, Debate, Philosophy: learn about syllogisms to combine two true statements to produce a new conclusion. • Maths: Use Boolean expressions to create and analyse Venn diagrams, helping students to visualize the relationships between sets. • Biology / Science: explore animals features to maximise their environment, including light sensitivity. Explore genetics, evolution, diurnal and nocturnal animals in the ecosystem.

Activity 5	Random numbers and the interactive book	
Let's invent activity	Interactive book	
Resources required	• A programming device with a USB A connection • Invention Engine Hub and LED bit • Internet connection • Cardboard • Invention Engine building tools; stencil, punch tools and rivets	• Cardboard cutters • Scissors • Glue • Markers • Cookies • Batteries • USB extension cable
Bits	• Light sensor bit • Servo bit • Magnet bit • Dial bit • Tilt sensor bit • Digital display bit	• Hub • Button bit • Battery bit • LED bit • Speaker bit • Proximity sensor bit
Computational thinking	• Loops • Conditionals • Variables • Mapping functions	• Snippets • Boolean expressions • Random numbers
Engineering, design thinking, making, tinkering, and inventing skills.	• Managing own time. • Project management; goals, tasks, timelines, resources, modifying tasks.	
Cross-curricular links	• Maths: Introduce concepts of probability using random numbers. Discuss how random numbers are used to model uncertain events and calculate probabilities.	• The 'Let's invent' activity can be modified to ask students to make an interactive book about a particular learning objective including History, science, art, health, etc.

What to look for in student's work

Define: Define the problem you are trying to solve (user, need and goal)

- Student has designed the solution with a clear understanding of who the user is, 'why' the problem needs to be solved, what a successful solution to the problem is or does.
- There is evidence that empathy played an important role when designing the solution.

Design: Brainstorm and select one idea

- Design is well considered, researched, and understood.
- Sketch is clear and labelled.
- Design notes are present.
- Materials list is well understood, and materials choices are well considered.

Plan: Sketch, label, and list your materials

- The project design is within the project's constraints.
- The design is within the project's specifications.
- Brainstorming or ideation process is present.

Code: Plan, code, test, and debug

- The process to select one idea to develop is present and compatible with checklist.
- The code does what is designed to do.
- The newly introduced coding concept is used and understood.

Make

- The new bit(s) are used in the invention.
- The code is used to make the invention better.
- The construction is sturdy and well thought out.
- The placement of the bits is functional.
- The invention is visually pleasing.
- The invention is intuitive and easy to use.

Test

- Functional testing is understood. The test design is sound, the test has been conducted and the interpretation of the results is reasonable.
- Usability testing is understood. The test design is sound, the test has been conducted and the interpretation of the results is reasonable.

Iterate

- Iterations are well documented. Sketches, calculations, notes, etc are present as evidence of iterations.
- Modifications to the design are considered. Reasons to keep or kick the changes are documented.
- Evidence of learning, growth, and improvement in each iteration is present.
- Student built upon feedback and incorporated it into their next iteration.
- The audience has been considered.
- Student has communicated the required information in an effective way.

Communicate

- Presentation (verbal or oral) is thoughtful and prepared
- The student answers questions about the invention confidently.

Reflect

- Evidence of deep thinking, connections to prior knowledge or experiences, and insights gained from the learning process are present.
- Logarithmic thinking: Did the student work through the steps? Do they understand the steps? Would they be comfortable using the same steps to solve a different problem?

Engineering skills

Decomposition: Can the student break a complicated problem into smaller problems to solve?

- Mindset: The student displayed tenacity and persisted through the challenges the project presented.
- Communication skills: The student demonstrated good communication skills and collaborated with other students. The student was able to give and receive feedback in a respectful and helpful manner.