You Light Up My Life!

Grades 9-11

DRAFT

SCALE K-12
SCalable Asymmetric Lifecycle Engagement
Precollege Microelectronics Workforce Development

INSPIRE
Research Institute for Pre-College Engineering

PURDUE UNIVERSITY

Cover Information

Copyright SCALE K-12 © 2024 Purdue University Research Foundation

Unit Title: Logarithms and Light
Grade Level Range: Grades 9-11

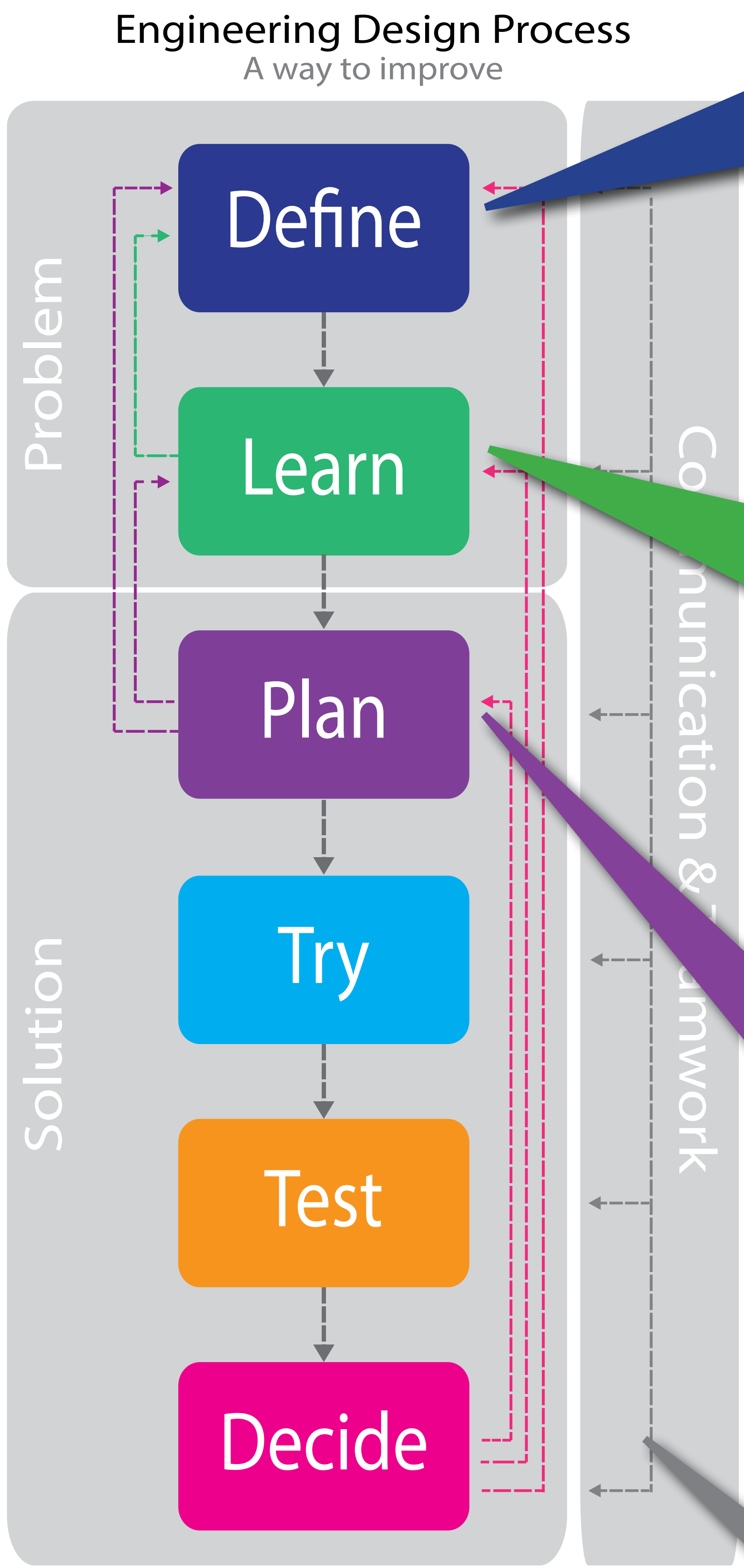
Acknowledgments

Teacher Authors
Jayden Barley
Laura Litwiler
Bill Riley
Andrea Sutton

Program Authors
Tamara Moore

Contributors
Contributor 1
Contributor 2

Overview: Engineering Design Process



Copyright © 2015 PictureSTEM- Purdue University Research Foundation

DEFINE THE PROBLEM

- Who is the **client**?
- What does the client need?
- Why does the client need it?
- Who is the **end user**?
- Why might the end user want it?
- What are the **criteria** (requirements)and **constraints** (limits) of the solution?

Problem Scoping:
WHO needs **WHAT** because **WHY**

LEARN ABOUT THE PROBLEM

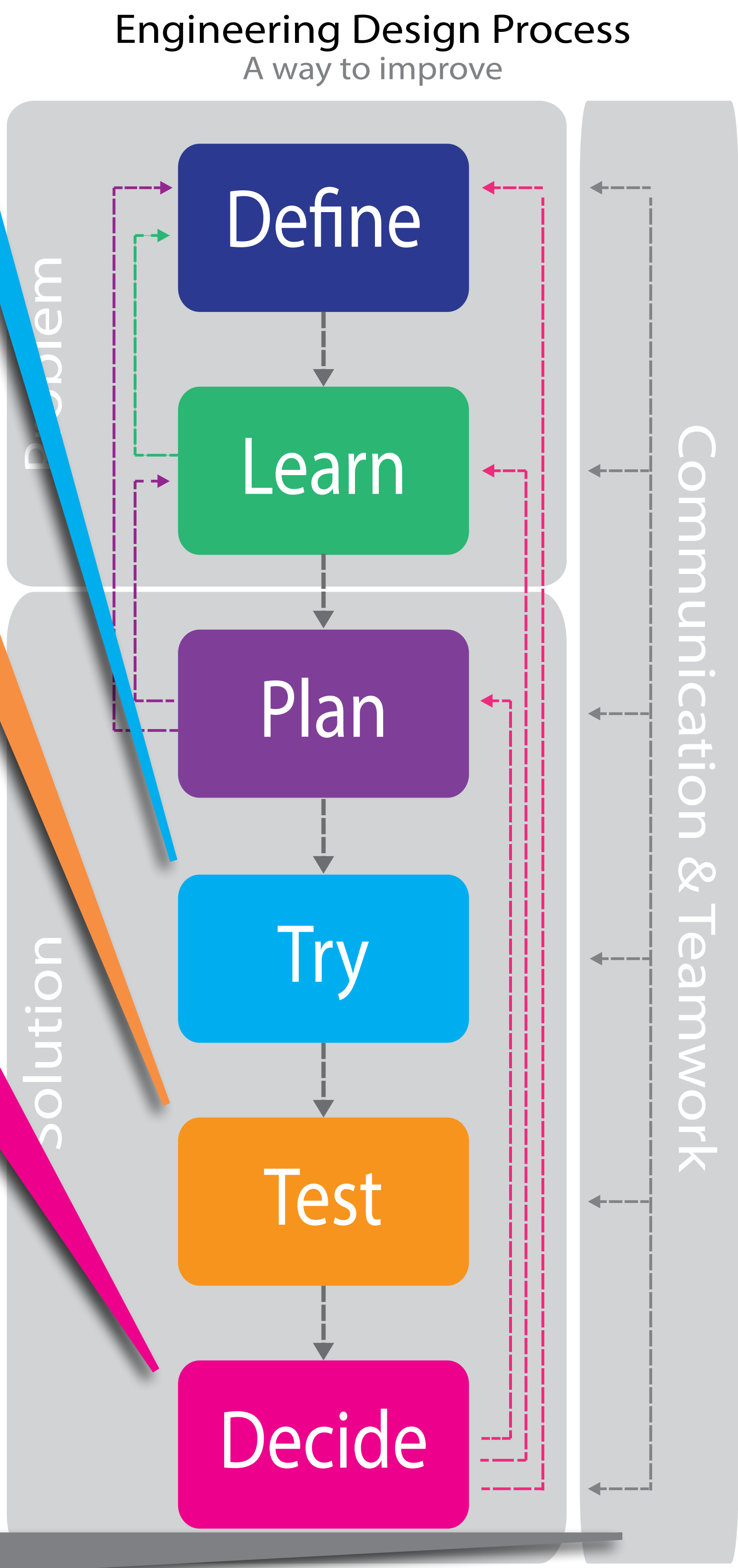
- What kind of background knowledge is needed?
  - What science/math knowledge will be needed?
  - What materials will be needed?
- What has already been done to solve the problem?
- What products fill a similar need?
- How should we measure success and improvement?

PLAN A SOLUTION

- Continue to specify the criteria/ constraints
- Generate ideas of possible solutions
- Develop multiple solution paths
- Consider **constraints, criteria**, and **trade-offs** (criteria that compete with one another)
- Choose a solution to try
- Develop plans (blueprints, schematics, cost sheets, storyboards, notebook pages, etc.)

COMMUNICATION

- Communicate the solution clearly and make sure it is easily understandable
- Use **evidence** to support why the client should use your solution



TRY A SOLUTION

- Put the plan into action
- Consider risks and how to optimize work
- Use criteria/constraints and consider trade-offs from the problem/plan to build a **prototype** (a testable representation of a solution), **model**, or **product**

TEST A SOLUTION

- Consider testable questions or hypotheses
- Develop experiments or rubrics to determine if the solution is meeting the stated criteria, constraints, and needs
- Collect and analyze data

DECIDE IF THE SOLUTION IS GOOD ENOUGH

- Are users able to use the design to help with the problem?
- Does the design meet the criteria and constraints?
- How could the design be improved based on test results and feedback from the client/user?

Iterative nature of design: Always consider which step should be next!

TEAMWORK

- Discuss in teams how the solution meets the criteria and needs of the client
- Consider different viewpoints from each teammate

Copyright © 2015 PictureSTEM- Purdue University Research Foundation

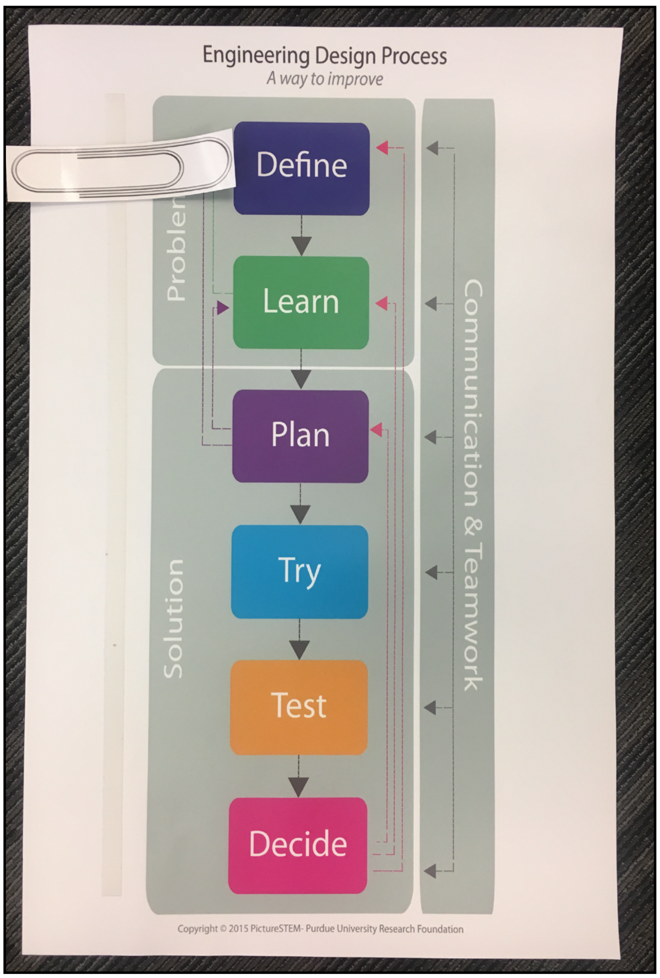
Overview: How to make EDP sliders

How to create the poster

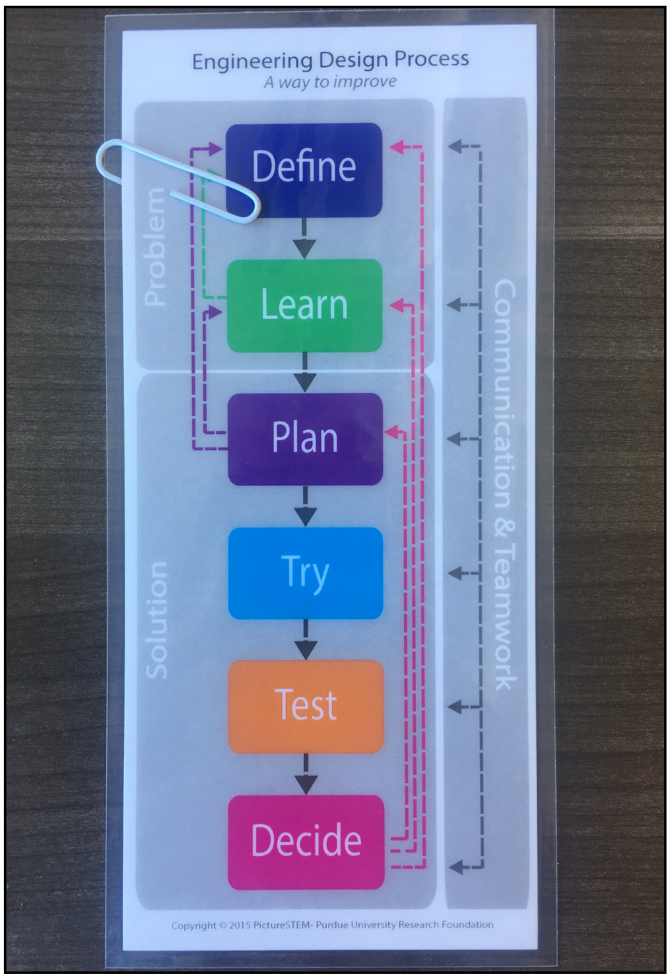
1. Download the high-quality PictureSTEM Slider Poster and the paper clip images from PictureSTEM.org.
2. Print the poster and the paper clip on poster-sized paper and cut to size. High-gloss or semi-gloss paper is the best choice.
3. Use self-sticking Velcro on the back of the paper clip and down the side of the poster so that the paper clip can be placed to point at all 6 sections of the slider.

How to create individual sliders

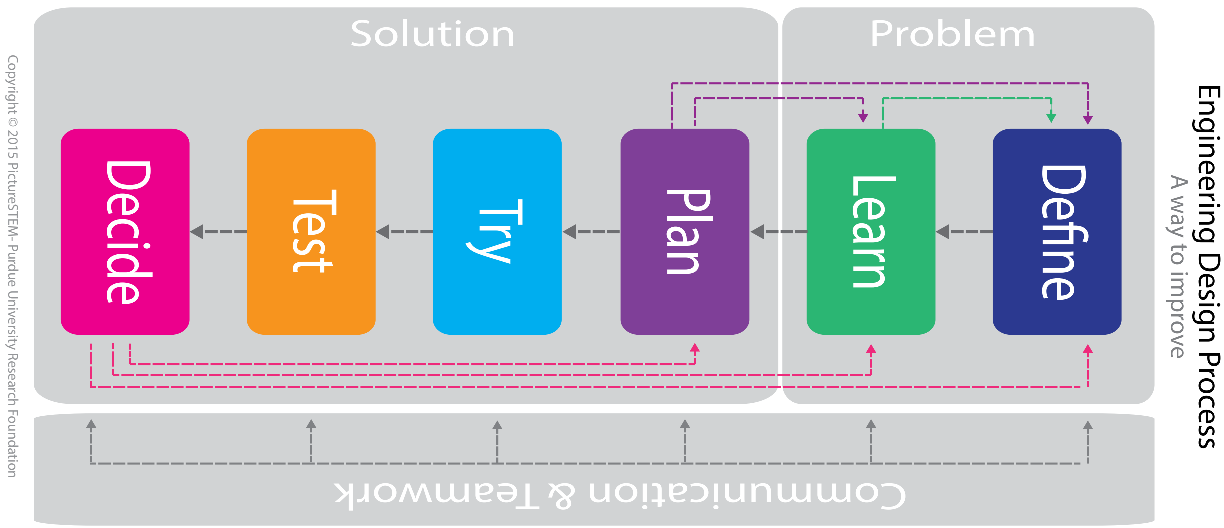
1. Print the sliders on the opposite page - enough for one slider per student in your class.
2. Cut the sliders apart.
3. Laminate the sliders individually.
4. Use a jumbo paper clip as the pointer for each slider.

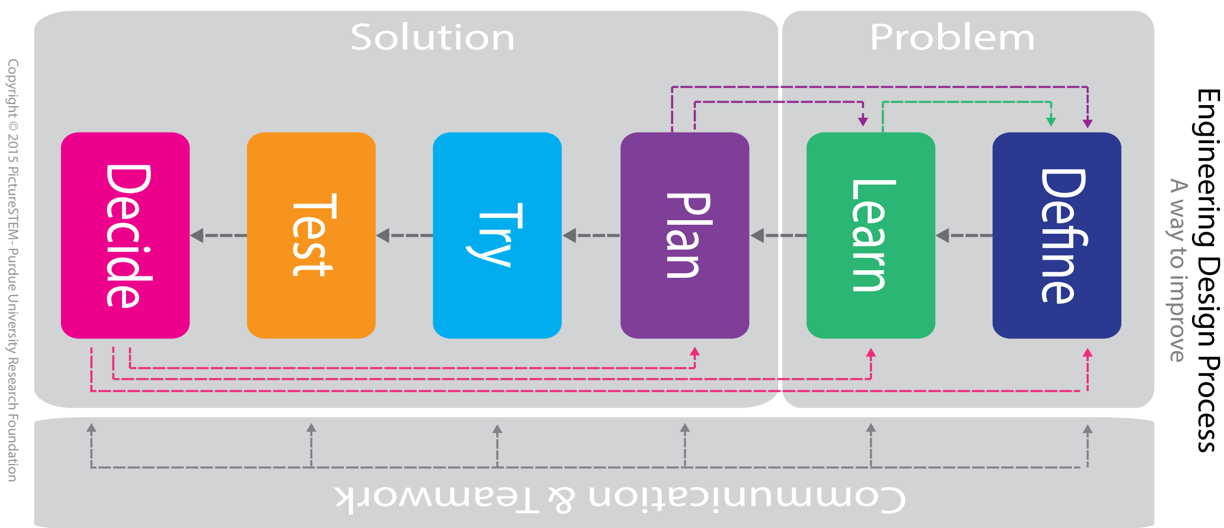


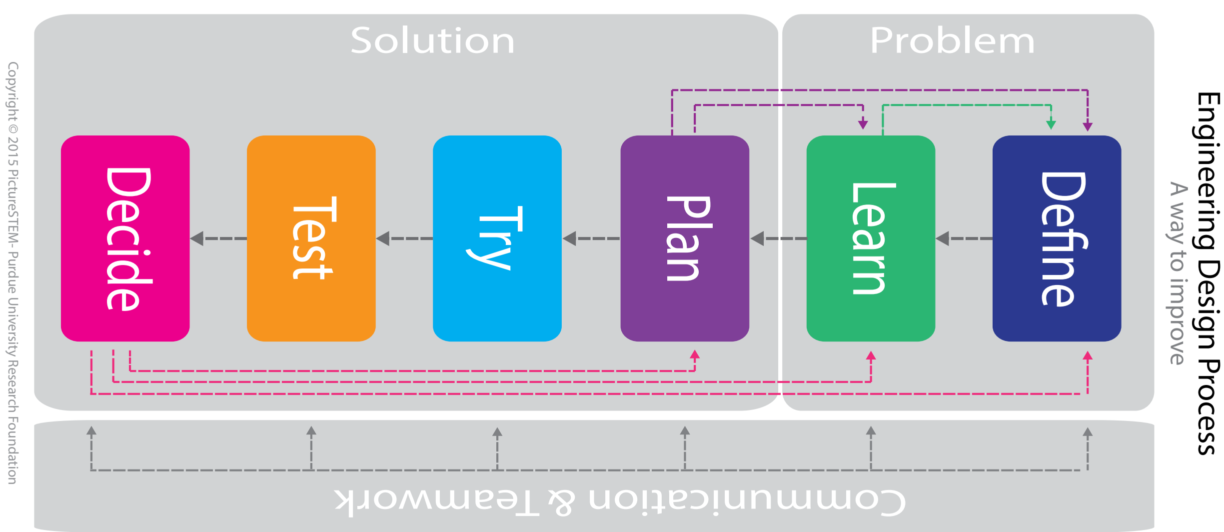
Poster



Individual slider


Engineering Design Process
A way to improve
Problem
Solution
Define
Learn
Plan
Try
Test
Decide
Communication & Teamwork
Copyright © 2015 PictureSTEM-Purdue University Research Foundation


Engineering Design Process
A way to improve
Problem
Solution
Define
Learn
Plan
Try
Test
Decide
Communication & Teamwork
Copyright © 2015 PictureSTEM-Purdue University Research Foundation


Engineering Design Process
A way to improve
Problem
Solution
Define
Learn
Plan
Try
Test
Decide
Communication & Teamwork
Copyright © 2015 PictureSTEM-Purdue University Research Foundation

Unit Overview

Grade Levels:	**Grades 9-11**
Approximate Time Needed to Complete Unit:	**TBD**

Unit Summary:

In this engineering design unit, students design custom party lights for a school group. They learn about logarithms, including graphing a logarithmic function, how they are the inverse of the exponential function. They learn that human perception is logarithmic, and use logarithms to control luminosity changes in their party lights so that the changes look smooth and even to the human eye.

Subject Connections:

Science Connections	Technology and Engineering Connections	Mathematics Connections
Human Perception of Light Carrying out Investigations Analyzing and Interpreting Data Engaging in Argument from Evidence Obtaining, Evaluating, and Communicating Information	Computer Engineering Computer Programming Microelectronics Engineering Design Process Problem Definition Seeking Stakeholder Input Designing Solutions Developing a Testing Protocol	With logarithms: • Solve equations • Create algebraic and graphical representations • Identify key features of graphs • Use technology to fit and transform equations • Solve a real-world problem • Define functions and their inverses • Apply rules and properties

Standards:

2023 Indiana Academic Standards: Mathematics

Mathematics Process Standards

PS.1: Make sense of problems and persevere in solving them.
PS.2: Reason abstractly and quantitatively.
PS.3: Construct viable arguments and critique the reasoning of others.
PS.4: Model with mathematics.
PS.5: Use appropriate tools strategically.

Algebra II

AII.ASE.4: Solve exponential and logarithmic equations in one variable.
AII.FF.1: Using technology, identify, create, and connect algebraic and graphical representations of each of the function families listed: … (e) exponential; (f) logarithmic … .
AII.FF.2: Graph each of the families of function with and without technology. Identify and describe key features, such as intercepts, domain and range, asymptotes, symmetry, and end behavior. Create inverse functions algebraically and/or graphically based on a given function. Model real-world situations with each function family.
AII.FF.3: Use graphical and algebraic structures and techniques to transform functions into equivalent forms to expose different information and identify key features. Connect the meaning of the key features to contextual situations.
AII.FF.4: Solve real-world problems with each function family, including situations in the context of science and economic phenomena.
AII.MFD.1: Define functions and their inverses and illustrate examples algebraically and graphically. Identify real-world situations that can be modeled using functions.
AII.MFD.3: Use technology to find a linear, quadratic, or exponential function that models a relationship for a bivariate dataset to make predictions; interpret the correlation coefficient for linear models. Compare and evaluate model fit using different function families.
AII.MFD.4: Explore the effects of function transformations using graphing technology. Explain the effects of transformations of functions such as $f(x)+k$, $kf(x)$, $f(kx)$, or $f(x+k)$ for different functions and values of k.

2023 Indiana Academic Standards: High School Integrated STEM

HS.IPS.1: Conduct or extend an investigation, analyze results, iterate, and revise to improve the design.
HS.IPS.2: Determine one or more viable solutions using data and information to resolve a scenario given criteria and constraints.
HS.IPS.3: Integrate processes and methodologies across disciplines to incorporate multiple sources of evidence, including data generated by the student, to support defining a solution.
HS.IPS.5: Design and conduct surveys or experiments minimizing bias and defining limitations of the data set used for analysis (e.g., measurement error, sample selection).
HS.DM.1: Use multiple systems of measurement (i.e., standard and metric) and data sets (e.g., plots, tables, graphs, charts) defined in course-level content standards to analyze real-world scenarios and the mathematical relationships represented by the data.
HS.DM.3: Use approximations and evaluate reasonableness of observations, results, and solutions throughout processes.
HS.CC.1: Communicate evidence, investigations, analyses, and the solution(s) of a problem in multiple media forms appropriate for the audience.

HS.CC.2: Implement roles and responsibilities to collaborate, contribute, and/or lead within and across various group settings (i.e., online, onsite and/or hybrid) and situations.
HS.CC.3: Evaluate competing solutions or arguments in a systematic way based on qualitative and/or quantitative evidence.

2023 Indiana Academic Standards: Science and Engineering Practices
SEP.1: Asking questions (for science) and defining problems (for engineering).
SEP.2: Developing and using models.
SEP.3: Planning and carrying out investigations.
SEP.4: Analyzing and interpreting data.
SEP.5: Using mathematics and computational thinking.
SEP.6: Constructing explanations (for science) and designing solutions (for engineering).
SEP.7: Engaging in argument from evidence.
SEP.8: Obtaining, evaluating, and communicating information.

Lesson Summaries

Lesson 1: What is the problem?
Students are challenged to produce party lights that can be adjusted in color and intensity to match the intensity of music: e.g. slowly dimming/brightening calm colors for slow dances; bright colors and quickly brightening/dimming light for high-energy dancing. They create a problem definition. They learn about how different kinds of light can change the energy of the room, and do some initial research on what colors are most calming / exciting.

Lesson 2: Let’s explore light: color, movement, and brightness
The objective of this lesson is for students to learn how to modify a program for LEDs using the micro:bit. Students use micro:bits to program LEDs to flash colors in patterns that make the light look like it’s dancing and moving, and to slowly increase and decrease the intensity of lights. They notice that the change in luminosity is more / less smooth depending on brightness.

Lesson 3: Plan a party lighting system
Students update the Problem Definition, including criteria and constraints, to incorporate new information. They imagine and plan how they will test their prototypes to ensure they address criteria and meet constraints. They plan their designs for party lights that meet the criteria and stay within constraints specified by the client, and they justify their design decisions using evidence-based reasoning.

Lesson 4: Try it out: create a prototype.
Students refer to their plans from Lesson 3 as they program their micro:bits to provide for multiple options of light patterns. They create programs that can match the intensity and mood of a variety of different songs and types of music.

Lesson 5: How do we make luminosity changes more even?
Students conduct an experiment to examine how LED changes in brightness are perceived for a variety of colors. They graph the data they collected on brightness perceived as a function of luminosity.
Then, they use the data they've collected to think about how they would program a "smooth" change in luminosity.

Lesson 6: An Introduction to Logarithms
Students learn about the history of logarithms. They come to understand that logarithms are a form of math that was invented to solve a problem. Then they talk about more modern ways that logarithms are used. Students learn about Fechner's Law: "subjective sensation is proportional to the logarithm of the stimulus intensity" S=k*log(R) where S=subjective sensation and R=intensity. They learn that perceived brightness increases according to a natural logarithmic scale.

Students discuss how logs are the inverse of an exponent. They graph logarithms and identify key features such as intercepts, domain and range, asymptotes, and end behavior. They use the properties of exponents to derive the properties of logarithms. They solve logarithmic equations in one variable.

Lesson 7: Using logarithms to control luminosity with the micro:bit
Students use Fechner's Law to describe their luminosity data by figuring out the mathematical formula for each color. Students learn about logarithmic rules and properties, and use them to revise their micro:bit programs, incorporating logarithmic brightening and dimming cycles.

Lesson 8: Test and Redesign
In this lesson, students revisit their ideas for testing from Lesson 3. They decide, as a class, how they will address testing of each criteria, and create a testing protocol. Teams then swap prototypes to test each other's work and give feedback. Finally, teams redesign and incorporate new feedback.

Lesson 9: Share with the Client
Each team of students presents their design with an example and either a poster or a presentation.

Possible extension lessons (possible to create later):
(Extension Lesson 1): Making our designs react to the volume of music
Students experiment with using the micro:bit's onboard sound sensor to adjust brightness/dimming according to the level of sound it detects. Students learn that people also perceive the volume of sound according to a natural logarithmic scale.

(Extension Lesson 2): Making our designs react to the pitch of music
Students add an extension program that identifies up to 3 pitches detected by the micro:bit's onboard sound sensor. Students use this data to match colors to pitches in their party lights.

Unit Overview

Unit Planner

Lesson	Time Needed	Objectives	Materials	Duplication Master
1. What is the Problem?	1 class	• Create a problem definition • Explain criteria and constraints • Identify characteristics of LED party lights from research • Explain the Engineering Design Process	• EDP slider and jumbo paperclip • Computing Device • Engineering notebook	1.A Content Pre-Assessment 1.B Client Letter 1.C: Problem Definition 1.D: Talking to Stakeholders
2. Let's Explore Light: Color, Movement, and Luminosity	1-2 classes	• Hook up the micro:bit and LED strip to the computer. • Modify a given program to adjust colors, movement, and luminosity in an LED strip. • Examine a given program to make predictions about what it will do when run.	• Computers • 1 micro:bit with Keyes breakout board • 1 LED strip with Dupont connector • USB-A cable	2.A Student Guide to Programming LED Strips with the Micro:bit 2.B Client Memo 2 2.C Generating Ideas
3. Plan a party lighting system	1 class	• Consider criteria and constraints in planning and designing a lighting system. • Justify design decisions in terms of criteria, constraints, results from end-user interviews, and exploration of the materials. • Argue for a position using evidence and clear justifications. • Collaborate effectively to develop a team plan.	• EDP slider and jumbo paperclip • Computer • Engineering notebook	3.A Evidence-Based Reasoning Part 1 3.B Evidence-Based Reasoning Part 2
4. Try it out: create prototype	1-2 classes	• Develop a prototype that addresses criteria and satisfies constraints. • Collaborate in teams, distributing roles and efforts as needed to achieve a goal.	4-5 Power strips Computers 1 micro:bit with breakout board 10 LED strips with Dupont connector USB-A cable 9V power adapter [Optional] Foamboard [Optional] Push-pins, paperclips, or masking tape	4.A How to Power Multiple LED Strips 4.B Coding Support Guide 4.C Troubleshooting and Debugging

Lesson	Time Needed	Objectives	Materials	Duplication Master
5. How do we make luminosity changes more even?	1 class	• Explain how human perception of light can be characterized as non-linear. • Conduct an experiment to determine how the human eye perceives changes in LED luminosity across colors and light levels • Collect data and use it to construct a graph of luminosity • Come up with ideas for how to even out the changes in luminosity for human perception	• Computers • 4-5 Power strips • Micro:bits with breakout boards • LED strips with Dupont connectors • USB-A cables • 9V power adapters	5.A Client Memo 3 5.B Experimenting with LED Strip Luminosity
6. Understanding Logarithms	2-3 classes	• Discuss the problem that logarithms were invented to solve. • Describe how mathematical processes can be designed to solve particular problems. • Describe Fechner's Law. • Explain how Fechner's Law is relevant to their findings about human perception of luminosity in the LED strips. • Identify the logarithmic function as the inverse of the exponential function	• Computers	6.A Evaluating Logarithms 6.B Graphing Logarithms 6.C Practice Working with Logarithms (homework)

Unit Overview

Unit Planner

Lesson	Time Needed	Objectives	Materials	Duplication Master
7. Using logarithms to control luminosity with the micro:bit	2 classes	• Justify design decisions with math and science evidence. • Apply understanding of the logarithmic power rule to create a power function for the micro:bit • Apply understanding of the logarithmic change of base rule to create a logarithmic function for the micro:bit • Apply understanding of logarithmic properties and inverse functions to control luminosity changes smoothly on the micro:bit. • Revise prototypes and test that they function by collecting and analyzing data.	• Computers • 4-5 Power strips • Micro:bits with breakout boards • LED strips with Dupont connectors • USB-A cables • 9V power adapters	7.A Coding Mathematical Functions in MakeCode
8. Test and Redesign	1-2 classes	• Develop a testing plan that will address criteria from the client and provide specific and measurable results. • Provide clear, specific, and actionable feedback and critique to peer teams on their prototypes. • Analyze data from testing. • Use data from testing to inform design decisions and further prototype revisions.	• Computers • 4-5 Power strips • Micro:bits with breakout boards • LED strips with Dupont connectors • USB-A cables • 9V power adapters	8.A Design a Testing Protocol
9. Share with the Client	2 classes	• Communicate science, technology, engineering, and mathematics ideas • Present their solution to the client's problem. • Use evidence-based reasoning to support their engineering decisions	• Computers • 4-5 Power strips • Micro:bits with breakout boards • LED strips with Dupont connectors • USB-A cables • 9V power adapters	9.A Communication Requirements 9.B Client Memo 4 9.C Content Post-Assessment 9.D Content Post-Assessment Key

Master Material List

	Unit Materials	Lessons Where Material is Used
Per Classroom	• 5 power strips • Black construction paper	
Per Group	• Computer • USB-A cable • Micro:bit • Keyes breakout board for micro:bit • 10 short LED strips by Elecfreaks • 9V power adapter • Poster-sized black foam board	
Per Student	• Engineering Notebook	

LESSON ONE:

Lesson Objectives
Students will be able to:
- Create a problem definition
- Explain criteria and constraints
- Identify characteristics of LED party lights from research
- Explain the Engineering Design Process

Time Required
One 45-minute lesson + homework

Standards Addressed
IAS Science and Engineering Practices
- HS.SEP.1

IAS Integrated STEM
- HS.IPS.3, HS.IPS.5

Key Terms
- Client
- Engineering design process
- Problem definition
- Criteria
- Constraints
- Requirements
- Microelectronics
- LED strips
- Luminosity

Lesson Summary

Students are challenged to produce party lights that can be adjusted in color and intensity to match the intensity of music: e.g. slowly dimming/brightening calm colors for slow dances; bright colors and quickly brightening/dimming light for high-energy dancing. They create a problem definition. They learn about how different kinds of light can change the energy of the room, and do some initial research on what colors are most calming / exciting.

Lesson Background

The first step in engineering design is to define the problem. Creating problem definition is a skill to be learned: the budding engineer must express the existing problem or challenge in terms of meaningful parameters and boundaries, criteria for success, constraints on the shape of the eventual design, and strategies to approach solving it. Just as creating a problem definition sets the stage for pursuing an engineering design solution, this lesson sets the stage for the learning and design work to come. In the next lesson, students will explore the technologies made available to them for creating an party lighting system with microcontrollers and LED strips.

Micro:bits are a type of microcontroller, so called because they have most of the controls of a computer, but in a much smaller "micro" package. These devices have chips that are responsible for all of the features, such as lights and sound outputs, and input reception from the on-board buttons or a computer via the micro usb cable. These devices are made even more powerful when used with sensors that give the micro:bit the ability to sense all types of things from temperature, moisture, or in this case heart rate. Students will learn about the constraints of the equipment as they program LED strips. In this lesson, students will work together in small groups to get the heart rate monitors, micro:bit, and provided code to function as designed.

Teacher Background

Teamwork

Students should be grouped strategically. When forming student groups, consider academic, language, and social needs. In place of strategic grouping, a random grouping can be substituted. Students will work in these groups, or "teams" throughout the unit.

What is the problem?

Effective teamwork is essential in this unit as well as in engineering in general; however, this unit does not provide specific support to develop those skills. If your students do not have experience with teamwork, it is highly recommended that you do some targeted team-building activities prior to beginning this unit.

Engineering Design Process

NOTE: If students are familiar with the engineering design process (EDP) before beginning the unit, the teacher can skip the (EDP) introduction.

The engineering design process (EDP) is an iterative, systematic process used to guide the development of solutions to engineering problems. There is no single engineering design process, just like there is no one scientific method. However, the various engineering design processes have similar components. The engineering design process (EDP) involves understanding the problem, learning background information necessary to solve the problem, planning, trying, testing the solution, making changes based on the tests, and communicating ideas. Students will use an engineering design process slider throughout the unit to help them understand where they are in the design process. For more information about the steps of the engineering design process presented in this unit, see the front matter section about it.

Criteria and Constraints

One of the key elements of the engineering design process, and often a confusing part for students, is the criteria and constraints. Criteria are the features of the solution that the client wants, while constraints are the limiting factors that the design must adhere by. Another way of thinking about the criteria are that they are anything the client and the engineers will use to judge the quality of a solution. Constraints are a specific type of criteria; they are those criteria that limit design possibilities, or the ways that the problem can be solved. If constraints are not met, the design solution is by default not a viable solution to the problem. Go over these definitions with your students before asking them to identify criteria and constraints from the client letter.

Engineering Notebook

Throughout the unit students will be recording information in an engineering notebook. You may want to have your students write in their notebooks in two different colors – one for thoughts and

Lesson Materials

Per classroom

- EDP Poster

Per student

- EDP slider and jumbo paperclip
- Laptop/Chromebook/ Tablet
- Engineering notebook

Duplication Masters

- 1.A Content Pre-Assessment
- 1.B Client Memo 1
- 1.C: Problem Definition
- 1.D: Talking to Stakeholders
- 1.E Content Pre-Assessment Answer Key
- 1.F Problem Definition Answer Key

Assessment

Pre-Activity Assessment

Use 1.A Content Pre-Assessment (found at the end of Lesson One in this guide) to assess what students know prior to beginning this unit. If students are having trouble with exponential functions, you may wish to review that content.

Activity Embedded Assessment

Circulate as students are working in teams on 1.C Problem Definition and 1.D Talking to End-Users. Provide support where students are struggling to complete the assignments in a way that is specific and responsive to the needs of the client.

Post-Activity Assessment

Examine the statements made in 1.C Problem Definition, and ensure that they are specific and accurate.

Examine each team's 1.D Talking to End-Users interview protocol. Look to see that the interview questions address the criteria of interest to the client.

prompts that are individual and one for thoughts and prompts that they discuss in their teams. Or, they may choose the left side of their notebooks for individual thoughts and the right side for teamwork. This will help you assess the students' ideas as well as help them recognize their own contributions and ideas. You also may want to have students complete a Notebook Cover and start a Table of Contents page. You may choose to have students tape/glue copies of the notebook prompts and/or the duplication masters into their notebooks or they can copy down a projected version of the prompts.

Vocabulary

Students will be introduced to new science and engineering vocabulary terms throughout the unit. It may be helpful to create a vocabulary section in their notebook with term definitions and memory clues. Additionally, the class could maintain a word wall or anchor chart.

Before the Activity

- Assemble the Engineering Design Process Sliders and post the EDP Poster in the classroom (see the front matter for how to assemble them). Make sure you and your students can refer to the EDP sliders and/or poster throughout the unit.

- Determine student teams of three or at most four. These teams should be their teams throughout the rest of the unit.

- Print and make copies of the following duplication masters in the labeled amounts:
  - 1.A Content Pre-Assessment (1 per student)
  - 1.B Client Memo 1 (1 per student or team)
  - 1.C: Problem Definition (1 per student or team)
  - 1.D: Talking to Stakeholders (1 per student)

- Give the pre-assessment for homework prior to starting the unit. Distribute 1.A Content Pre-Assessment and have students answer the questions individually. Make sure to tell students that this is just to assess prior knowledge, so it is okay to not know the answers.

- Based on student answers, decide whether you should, in preparation for this unit, (1) spend some time reviewing exponential functions; (2) spend some time discussing the engineering design process using materials in the Overview.

Classroom Instruction

Introduction

1. **Introduce the theme of the unit.** Explain that students will get to apply mathematics and science as they create a design for a client that involves light.
2. **Present the client brief.** You can pass out copies of 1.B Client Memo 1 and/or project it on the board. Read it aloud to students or have students take turns doing so.
3. **Ask for student reactions about LED lights.** You may ask, for example, *do any of you have experience with LED strip lights? What do you know about them? Have any of you tried to program strip lights before? What did that involve? Have you seen LED lights in use at a party?*
4. **Ask for student reactions about the engineering challenge.** *Does this challenge surprise you? Why? What kind of engineer do you think works on this sort of problem? What skills and knowledge do you think such an engineer needs?*
5. **Show a micro:bit and talk about microelectronics relevance.** Students will be using micro:bits as the microcontrollers for the LED Strips. Ask if students have heard anything about the new microelectronics industry work that is coming to Indiana. Talk about where microelectronics are most commonly found in the devices that we use every day.
6. **Ask:** ***what is a microcontroller and how is it used?*** Microcontrollers are tiny computers that use very little power and can be programmed to respond to inputs with outputs, among other things. Micro:bits are in all kinds of devices, from car key fobs and coffee makers to TV remotes and toys.
7. **Form student teams.** Explain that students will be working in these small teams to address the challenge brought to them by the client. They will follow the engineering design process, and they will need to use their math skills to complete the challenge.

Activity

8. **Pass out and introduce the engineering notebooks. Say:** *Engineers use notebooks to document their design process and keep notes. We will also be using engineering notebooks throughout our engineering challenge. Each day, you'll use the notebooks to take notes and record what you are learning. In addition, there are questions that you'll be asked to answer individually and in teams. Each day, turn in your engineering notebooks before you leave class.*

9. **Distribute 1.C Problem Definition to student teams.** Point to the engineering design process slider first step, 'Define.' Explain that engineers need to clarify the problem or challenge before they begin work on designing. Review the questions on 1.C with the students, and make sure they know what "criteria" and "constraints" are. Make sure students have access to 1.B Client Letter, as they work, and let them know that most (but not all) of what they need to define the problem is in the client letter.
10. **Have teams complete 1.C Problem Definition.** Circulate and answer student questions.
11. **Gather student attention to the EDP slider.** Point to the EDP slider second step, Learn.
12. **Explain about learning from end-users.** Tell students that since the challenge is to design a lighting system for high school students, they can learn a lot about what the final design should look like from high school students like themselves.
13. **Distribute 1.D Talking to End-Users. T**o learn from their end-users, they need to come up with a plan for who to talk to and what questions to ask. In their teams, they should come up with their best plan, and test it by interviewing each other. They will then revise their questions and testing plan, and each student will use the plan to interview another one or two high school students for their homework.
14. **Have teams complete 1.D Problem Definition.** Circulate and answer student questions. Make certain students test their questions by trying to answer the questions themselves.

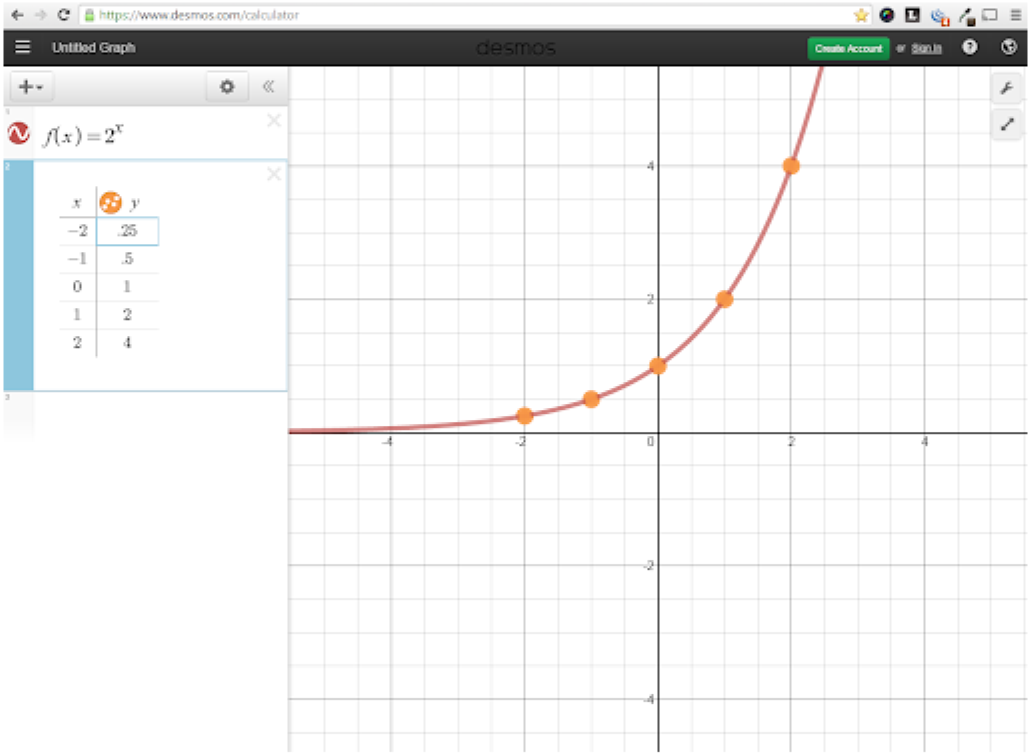
Closure

15. **Remind students of the homework assignment.** Each student should interview one or two other high school students using the questions and plan that they came up with in their team. They will need to write down the responses of their end-users so they can report out to the class.
16. **Ask: what else will you need to know in order to address this challenge?** Remind students that they addressed this question in their groups on 1.C Problem Definition. Encourage students to discuss the ideas as a class, as time allows.

Name______________________________________ Date______________ Period __________

1.A Content Pre-Assessment

1. What is an inverse function? Explain and demonstrate by drawing the inverse of the function below.



2. Simplify this function, showing your work: $f(x) = (x^{12})(x^6)$

   a. What is the domain of the equation in question 2? What is the range?

Name______________________________________ Date______________ Period __________

1.A Content Pre-Assessment

3. Simplify or find the value of each of the following:

   a. 5^4
   b. 5^{-4}
   c. $25^{1/2}$
   d. $(x^5)^4$
   e. $a^3 \times 3a^6$
   f. $a^2 \times b^2$
   g. $a^5 \div a^3$
   h. $z^{-3} \div z^4$

4. What is a logarithm?

5. What is the value of log(1000)?

6. What is the value of $\log_2(16)$?

7. Write the equation $12^4 = 20{,}736$ as a logarithm.

Name____ Date_______________ Period __________

1.A Content Pre-Assessment

8. How can you use logarithms and exponents to help you make a quick estimate (without a calculator) of the product of multiplying 10,343,074,143 by 1,322,124,456? Show your work.

9. What is an engineer? What kind of work does an engineer do?

10. Have you ever done some engineering? Explain what you did and why.

11. What does the term, “microelectronics” mean?

12. How are microelectronics used in the field of algebra or pre-calculus?

13. What jobs would you be interested in that use microelectronics? Provide one example of how microelectronics is used in that job.

1.B Client Memo 1

Dear Engineers,

I am writing on behalf of the dance committee at Anytown High School. We are contacting you to enlist your help to design the decorations for the end-of-year dance.

This year our theme is starlight! We want to decorate our dance hall with LEDs arranged into symbols of light at night. We also want the lights to change color and seem to dance, brighten, or sparkle with the music. And, we want to be able to change the pattern of color and brightness to match with high-energy dance tunes, slow romantic music, and other kinds of dance music.

We know a few things about light. For example, we know that bright red and yellow lights tend to be the most exciting. We also know that slower, softer light changes are more romantic than fast changes in color or brightness. However, we would like you to find out more about teens' preferences in dance lighting by interviewing some high school students.

Please create some prototypes of possible dance lighting systems. To create your prototypes, you will need to program a microcontroller to control LED light strips Your prototypes need to show off at least three different programs for color and brightness changes, controlled with the press of a button. You will display your prototypes on a poster-sized sheet of black foam board, so you can also exhibit the shapes you will arrange with your LED strips to fit the theme. Use the engineering design process, mathematics skills, science knowledge, and your creativity to come up with some designs that will wow us!

Yours sincerely,

Gina Dancer, Grade 12 Student
Head of the Dance Committee of Anytown High School

Name____ Date________________ Period ___________

1.C Problem Definition

Define the engineering problem or challenge that you are trying to solve.

1. What is the engineering problem or challenge?

2. Who is the client?

3. Who are the end users?

4. Why is this challenge important to the client and end users?

5. An effective solution for the client will meet the following criteria:

6. The constraints (or limits) on our solution are:

7. What do you already know that will be helpful to work on this challenge?

8. What are two specific things you still need to know to address this challenge?

1.D Talking to End-Users

To learn more about what the end-users (high school students who will attend a dance) would like to see in your lighting system design, you can talk to some people in that demographic group to get their ideas.

Your job on this sheet is to make a plan to elicit the ideas and information you need from the people you interview. Think about and decide on a plan for the following:

1. What do you need to know? You should refer to 1.B Client Letter and 1.C Problem Definition to ensure you have a plan to address everything that you need to know from end-users.

2. Who will you ask? How many people will you interview? When would be a good time to approach potential interviewees, and why? You should consider who you will interview in terms of fairness, so you get a good variety of opinions and don't rely just on one type of student. Also, if you have a lot of questions, you might wish to split them up among different people so as not to burden one interviewee too much.

3. How will you make the request? How will you approach someone you wish to interview in a way that is respectful of their time and opinion?

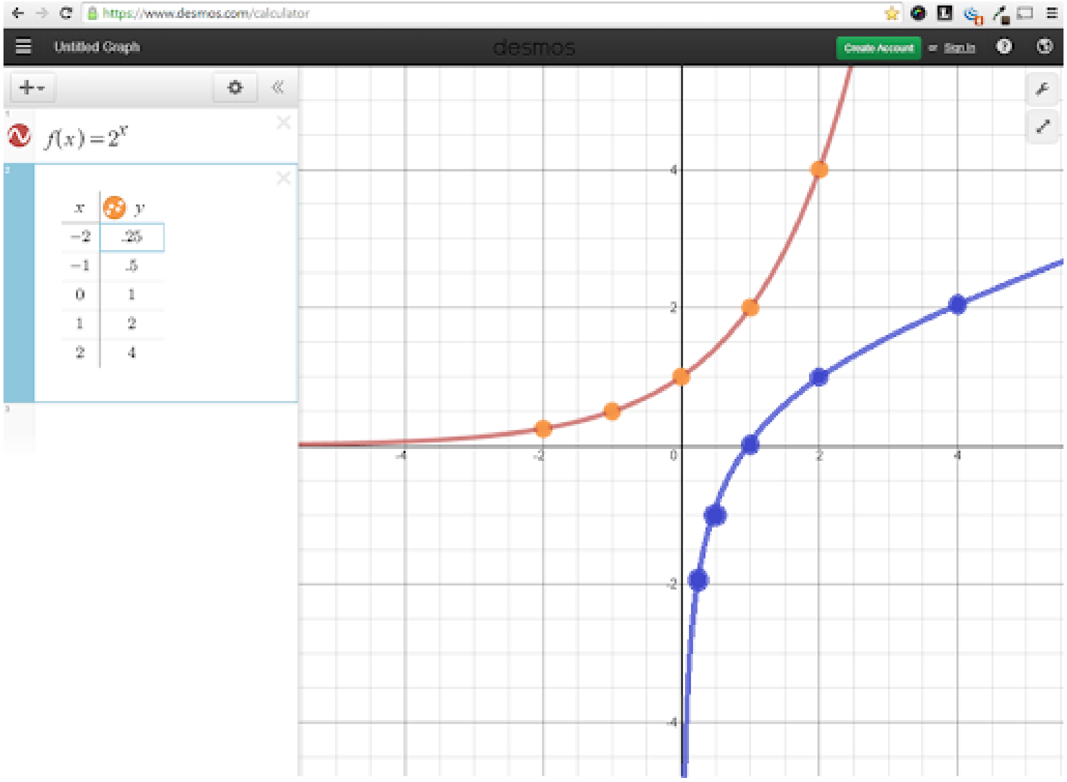
4. How will you set the context? How will you make sure that the people you interview understand what you are asking and why you are asking, so they can give you an informed opinion?

5. What will you ask? What questions will elicit the information that you need?

6. How will you close the interview? How will you ensure that your interviewee does not feel too burdened by the time spent asking them questions, and that they feel appreciated for their time?

1. What is an inverse function? Explain and demonstrate by drawing the inverse of the function below.

   An inverse function is a function that reverses the operations of another function.



2. Simplify this function, showing your work: $f(x) = (x^{12})(x^6)$

   $f(x) = x^{18}$

   a. What is the domain of the equation in question 2? What is the range?

      The domain is all real numbers. The range is all positive real numbers > 0.

1.E Content Pre-Assessment Answer Key

3. Simplify or find the value of each of the following:

   a. 5^4 = **625**

   b. 5^{-4} = **1/625**

   c. $25^{½}$ = **5**

   d. $(x^5)^4$ = **x^{20}**

   e. $a^3 \times 3a^6$ = **$3a^9$**

   f. $a^2 \times b^2$ = **$(ab)^2$**

   g. $a^5 \div a^3$ = **a^2**

   h. $z^{-3} \div z^4$ = **z**

4. What is a logarithm?

   A logarithm is the inverse function to an exponent. Given a base and a number, the logarithm is the exponent you would need to raise the base to in order to produce the number.

5. What is the value of log(1000)?

   log(1000) = 3

6. What is the value of log2(16)?

   $\log_2(16) = 4$

7. Write the equation 124 = 20,736 as a logarithm.

   $\log_{12}(20{,}736) = 4$

8. How can you use logarithms and exponents to help you make a quick estimate (without a calculator) of the product of multiplying 10,343,074,143 by 1,322,124,456? Show your work.

   Estimate the first number and the second number as powers of 10, so:
   $10^{10}10^9=10^{19}$ which gives an estimate of 10,000,000,000,000,000,000

1.E Content Pre-Assessment Answer Key

9. What is an engineer? What kind of work does an engineer do?

    An engineer is a person who applies mathematics, science, creativity, and technical skills to design solutions to problems.

10. Have you ever done some engineering? Explain what you did and why.

    Responses will vary.

11. What does the term, “microelectronics” mean?

    Student answers may vary, but the formal definition of microelectronics is the design, manufacture, and use of microchips.

12. How are microelectronics used in the field of algebra or pre-calculus?

    There are many answers to this question, but one example could be using microelectronics to collect a specific type of data for graphical analysis.

13. What jobs would you be interested in that use microelectronics? Provide one example of how microelectronics is used in that job.

    Students’ answers will vary based on interest. Credit may be given as long as at least one job example is provided with their logic behind how that job uses microelectronics.

1.F Problem Definition Answer Key

Define the engineering problem or challenge that you are trying to solve.

1. What is the engineering problem or challenge?

   The challenge is to design a lighting system for a school dance.

2. Who is the client?

   The client is the Dance Committee of Anytown High School, or the Head of the Dance Committee, Gina Dancer.

3. Who are the end users?

   The end users are students of Anytown High School who attend the school dance.

4. Why is this challenge important to the client and end users?

   The challenge is important to the client and end users because they want to have fun and exciting lighting decorations that wow them.

5. An effective solution for the client will meet the following criteria:

   1. **the LEDs are arranged into symbols of light at night**
   2. **the lights change color**
   3. **the lights seem to brighten, dance, or sparkle with the music**
   4. **the light patterns can be changed to match at least 3 different kinds of music**

5. The constraints (or limits) on our solution are:

   1. **the designs must use a microcontroller to control LED strips**
   2. **the designs must be changeable with the press of a button**
   3. **the prototype must be displayed on poster-sized black foam board**

4. What do you already know that will be helpful to work on this challenge?

   1. **We know that red and yellow lights are exciting**
   2. **We know that slower light changes and softer lights are more romantic**
   3. **(Further answers may vary)**

4. What are two specific things you still need to know to address this challenge?

   Responses will vary.

LESSON TWO:

Lesson Objectives
Students will be able to:
- Hook up the micro:bit and LED strip to the computer.
- Modify a given program to adjust colors, movement, and luminosity in an LED strip.
- Examine a given program to make predictions about what it will do when run.

Time Required
One or two 45-minute class periods

Standards Addressed
IAS Mathematics
- HS.SEP.3

Key Terms
- Micro:bit
- MakeCode
- Breakout board
- Luminosity

Lesson Summary
The objective of this lesson is for students to learn how to modify a program for LEDs using the micro:bit. Students use micro:bits to program LEDs to flash colors in patterns that make the light look like it's dancing and moving, and to slowly increase and decrease the intensity of lights. They notice that the change in luminosity is more / less smooth depending on brightness.

Lesson Background
This lesson is designed to give students a gentle introduction to block coding and use of microcontrollers using MakeCode and the micro:bit. Students will be given some pre-existing programs and a guide. The guide both points out relevant features of the programs, and prompts students to try to accomplish certain tasks. This will work best if students record the prompts in their engineering notebooks and carefully record the things they try and what happens. Then, teams should talk about what changes in the code resulted in changes in the LED strips. They can test if they are correct about how the code works, by seeing if they can make predictions about what will cause changes to the LED strips, before checking to see if their predictions are correct.

In the next lesson, students will use what they learned in this lesson, plus what they learned from interviewing peers, to plan out a design for their LED lighting system.

Teacher Background

Teamwork
Students work in their teams to discuss the provided programs, make predictions about function, hypothesize how the blocks of code work together to control the LED strip, and test their hypotheses by making changes to the code and running the adjusted programs.

Engineering Notebook
Students use their engineering notebooks to keep track of their predictions, experiments, and observations.

Vocabulary
micro:bit, LED strip, breakout board, MakeCode, luminosity

Let's explore light: color, movement, and brightness

Before the Activity

- Try the activity in this lesson yourself so you can know how to address possible difficulties that students might have with it, and to have a micro:bit and LED ready to use.

- Prepare a micro:bit and LED strip to show to students.

- Print and make copies of the following duplication masters in the labeled amounts:
    - 2.A Student Guide to Programming LED Strips with the Micro:bit (1 per team)
    - 2.B Client Memo 2 (1 per team)
    - 2.C Generating Ideas (1 per student)

- Prepare kits of materials to hand out to teams.

- Decide how to store student materials between lessons.

Classroom Instruction

Introduction (10 minutes)

1. **Pass out the engineering notebooks.**
2. **Review findings from student homework as a class.** Have students report out what they learned from interviewees, and create a chart of findings that emphasizes both congruent and contradictory findings. Have students record the class findings in their engineering notebooks.
3. **Show students the micro:bit and connected LED strip you prepared.** Give them a short overview of the activity described in the student guide, that you tested in preparation.
4. **Explain that today, teams will work to 'Learn' about how to program LED strips.** Indicate the second step of the engineering design process on the slider you have posted.

Activity

5. **Give teams the materials needed for the activity,** including: 2.A LED Strip Programming, 1 micro:bit, 1 USB-A cord, 1 Keyes breakout board, and 1 rainbow LED strip.
6. **Let students know that you are available to assist them** as they work through the directions.
7. **Have each team work through the student guide,** conduct the relevant tests, figure out how to make changes to the programs, and record what they are learning in their engineering notebooks. Circulate and provide assistance as needed, to enable students to figure out how to solve the mini-challenges in the guide.

Lesson Materials

Per classroom

- EDP Poster

Per Group (3 per group)

- USB-A cable
- Micro:bit
- Keyes breakout board for micro:bit
- 1 short LED strip

Per student

- EDP slider and jumbo paperclip
- Laptop/Chromebook/ Tablet
- Engineering notebook

Duplication Masters

- 2.A Student Guide to Programming LED Strips with the Micro:bit
- 2.B Client Memo 2
- 2.C Generating Ideas

Assessment

Pre-Activity Assessment

Listen to class discussion of the interviews students conducted with end-users to assess their understanding of the goal of the design challenge and criteria for success.

Activity Embedded Assessment

Pay attention to conversations within teams about what they are seeing in the code provided and what they understand or hypothesize about its function.

Post-Activity Assessment

Examine students' engineering notebooks for evidence that they examined the provided programs, made predictions about how they would function, and tested their ideas by making changes to the program and running those changes to observe and record what happened.

Closure

8. **Ask what students noticed about color and luminosity.** *What were the different methods to show colors in MakeCode? What seemed to be the advantages and disadvantages of each method? What was the "Strip show" mini-block for? When was it needed? Did anyone see the luminosity changes speed up or slow down when running the second program? Do they have any ideas for how to better control the luminosity changes?*
9. **Conduct a class discussion of students' experiences with the activity.** *What was difficult? What was surprising? What was interesting? Do students agree that this was part of the "Learn" step in the engineering design process? What would they like to learn more about? What, do they think, will they need to learn more about to complete this challenge?*
10. **Present 2.B Client Memo 2.** Explain that teams will have 10 LED strips, a power adapter, and one microcontroller setup like they used in this lesson, to complete their designs. They should add these constraints to their Engineering Journal, to supplement 1.C Problem Definition.
11. **Assign homework: individual brainstorming.** Assign students to brainstorm as many ideas (big and small) as they can to address the challenge in their Engineering Journals. Once they have done that, they should plan out at least two of the ideas, using 2.C Generating Ideas as a scaffold. They will need to refer back to 1.B Client Memo 1, 2.B Client Memo 2, 1.C Problem Definition, and the data they collected from 1.D Talking to End-Users, so they can remember and keep in mind the criteria and constraints, as well as end-user preferences. They should be prepared to present and discuss their ideas with their team in the next class.

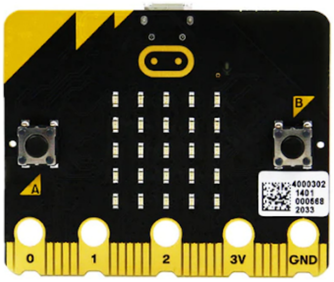
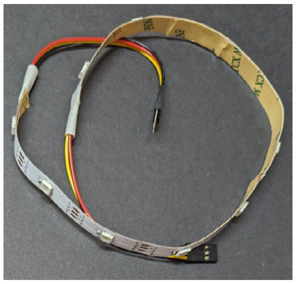
Let's explore light: color, movement, and brightness

2.A Student Guide to Programming LED Strips with the Micro:bit

Part 1: Set up

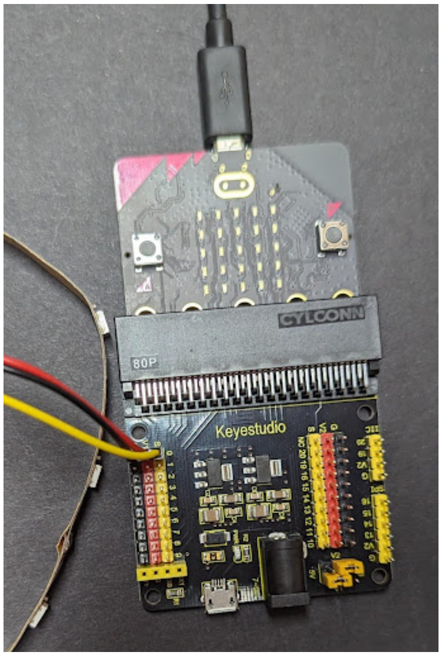
This guide is designed to demonstrate a variety of things you can do using MakeCode to program a micro:bit to control lights on one or more LED strips. In each section, you will find a link to a project example that you can open, make changes to, and download.

Before you begin, it's time to identify and set up the hardware. In addition to a computer, you should have the following items:

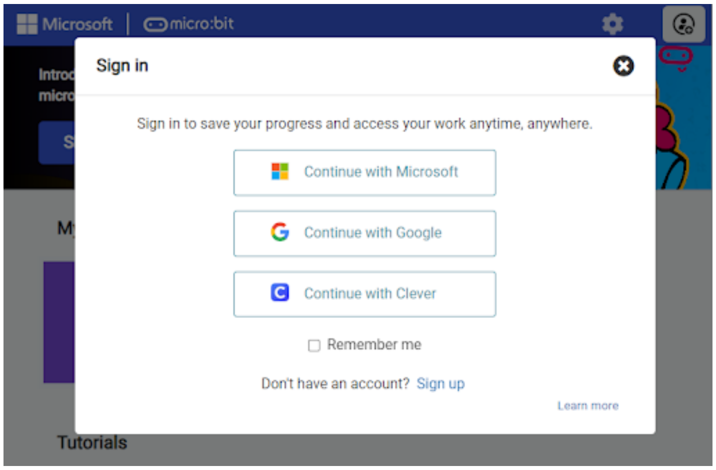
			
micro:bit (color may vary)	USB-A cable	Keyestudio breakout board	Programmable RGB 10-LED strip

The first thing you need to do to get started is to put these things together:

1. Tuck the micro:bit into the top of the breakout board.
2. Plug the end of the LED strip with holes into the top left set of pins on the breakout board, carefully matching the colors of the wires to the colors of the pins.
3. Plug the small end of the USB-A cable into the top of the micro:bit.



2.A Student Guide to Programming LED Strips with the Micro:bit



Next, open a browser window in the computer.

Surf to:
https://makecode.microbit.org/#

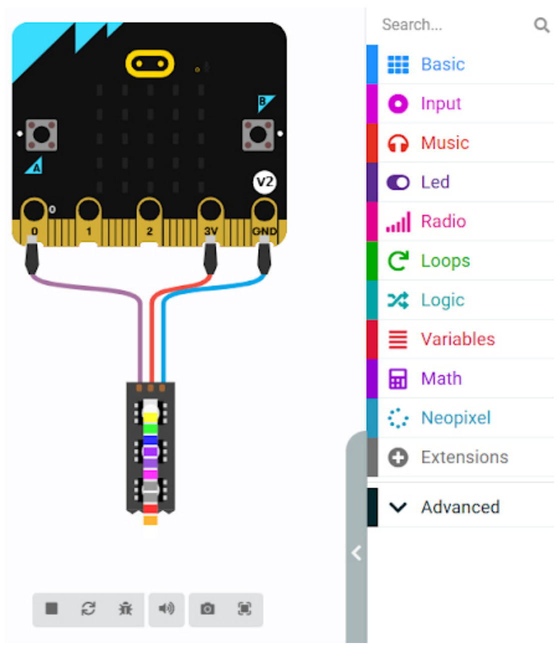
If you don't yet have an account, sign up to get one now.

Once you have a MakeCode account, add the project you can find at this link (or the QR Code).

In the top right corner, click the button that says "Edit Code" and the project will be added to your account.

The name of the project is "LEDstrip_moving_rainbow."

https://tinyurl.com/Guide-Step1



Once you've opened the project to edit it, you should see several important areas of the MakeCode window.

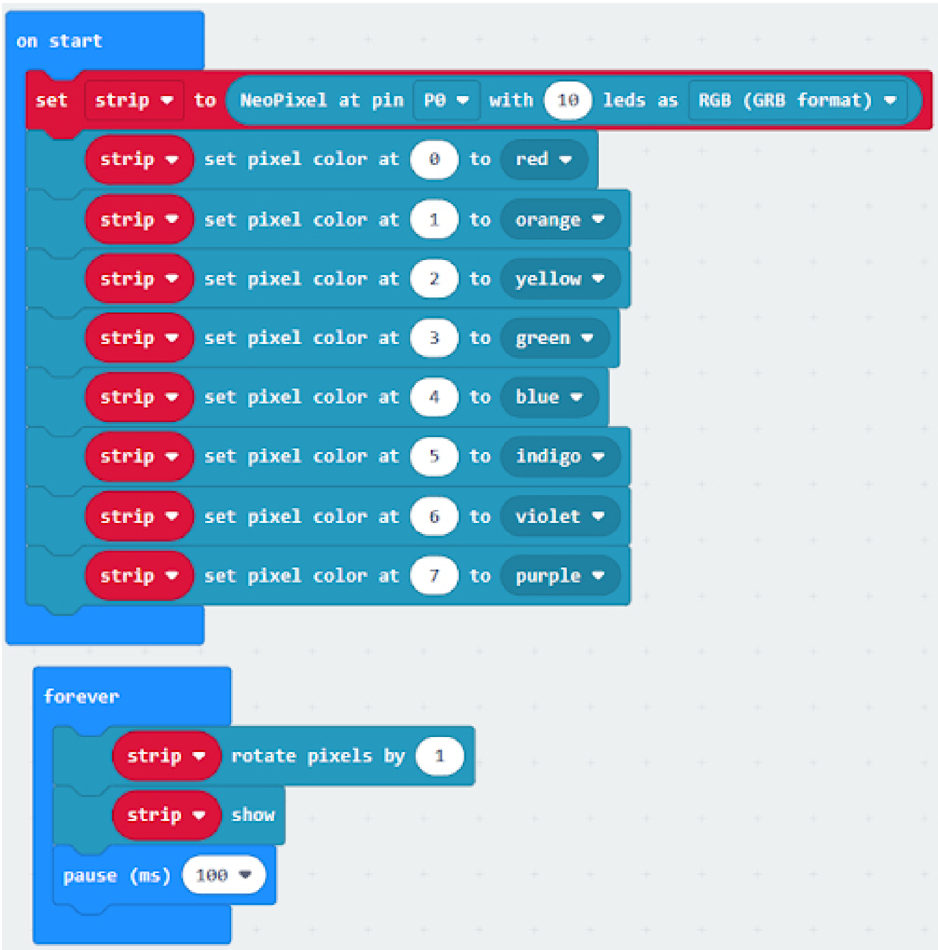
To the far left you should see a simulation of what downloading the code to your micro:bit will do.

Below the simulation, there is a gray bar with buttons to Stop, Restart, Debug, Mute, Take a Screenshot, and Launch in Fullscreen.

Just to the right of that is a search bar and a menu. The menu contains code "blocks" that you can add to your program. Try clicking on these to see what's there.

The Neopixel menu is not a standard menu; you have to add it by clicking on "Extensions." It contains the code blocks to manipulate LED strips. (The LED menu only manipulates the red LEDs on the micro:bit.)

Part 2: Experiment with LED color and movement



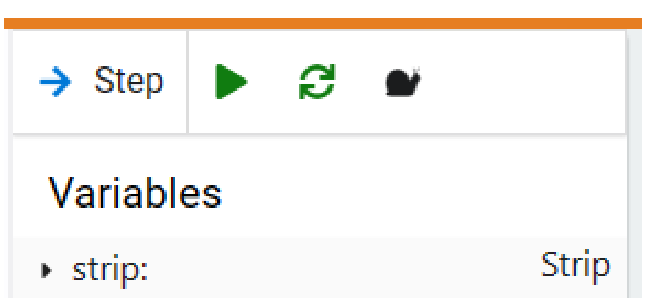
The section to the right of the screen is the program that will download to the micro:bit to control your LED strip.

Try downloading it now by pressing the purple "Download" button on the lower left.

The first time you do this you will see a pop-up window asking you to "Connect your micro:bit to your computer." If you press the "Next" button, it will then ask you to "Pair your micro:bit to your browser." Press the "Pair" button and follow the instructions.

Now you should be able to download the code. What happens when you do so?

Hint: if the answer is "nothing" check that you connected everything correctly!



Once you have the code working and your LED strip is flashing, it's time to play! Try the following:

1. Run the program in "debug" mode. Press the debug button in the gray strip under the simulation. The menu will disappear. Instead, you will see the "Step" section pictured here, and a yellow highlight around one of the code blocks. Press "Step" to see what happens. Press it again! (and again…)

More things your team can try:

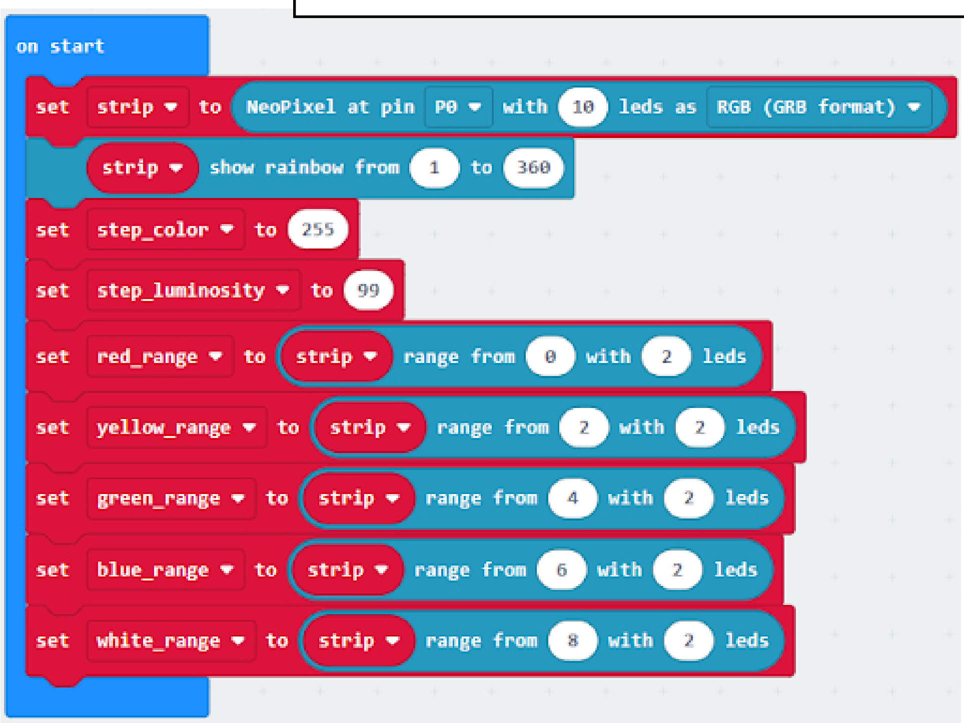
1. See if you can get the pixel colors to rotate more slowly.
2. Make the colors stop rotating.
3. Try changing the colors.
4. Try replacing the “set pixel color” blocks with the “Show Rainbow” block. What changes? What happens if you change the numbers in the “Show rainbow” block?
5. Try replacing the “rotate pixels block” with the “shift pixels” block. What happens? What if you add multiple copies of this block, one after the other?
6. What else can you think of to try?
7. If you ever want to go back to the original code, just use the link and “Edit Code” again.

Part 3: Experiment with LED luminosity

Now let’s try another program. This one will change the luminosity, or brightness, of the LEDs on your strip.

Using either the URL: https://tinyurl.com/Guide-Step2 or the QR Code to the right, open the project “LEDstrip_change_luminosity.” Then add it to your MakeCode account by pressing the “Edit Code” button.

Notice that, where the LEDstrip_moving_rainbow program had two sections of blocks, this program has three.



Take a look at the first block of code, "on start." This block of code sets the initial values of a number of variables, and it turns on the LED strip as a rainbow.

Write down the variables in your Engineering Notebook. Discuss with y our teammates your best guess as to what those variables represent, and write that best guess also.

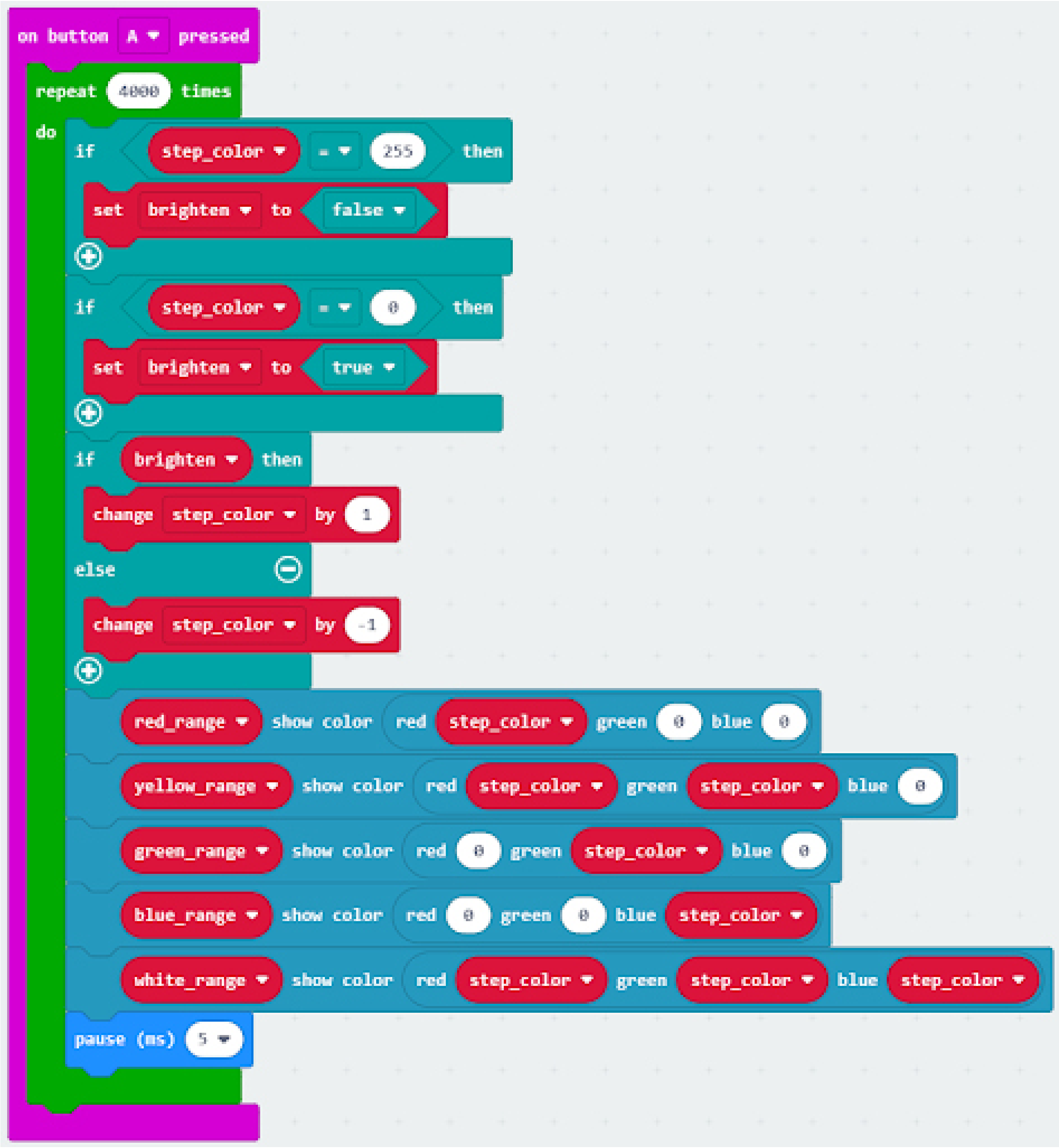
2.A Student Guide to Programming LED Strips with the Micro:bit

The second block of code says what should happen when button A on the micro:bit is pressed.

Notice that this block uses a different method to show colors than either the first block in this program or in the prior program.

Record all the different methods you've seen to light the LED strip in your Engineering Notebook.

What do you think will happen when you run this code? Discuss with your team. Then, make a prediction in your Engineering Notebook.



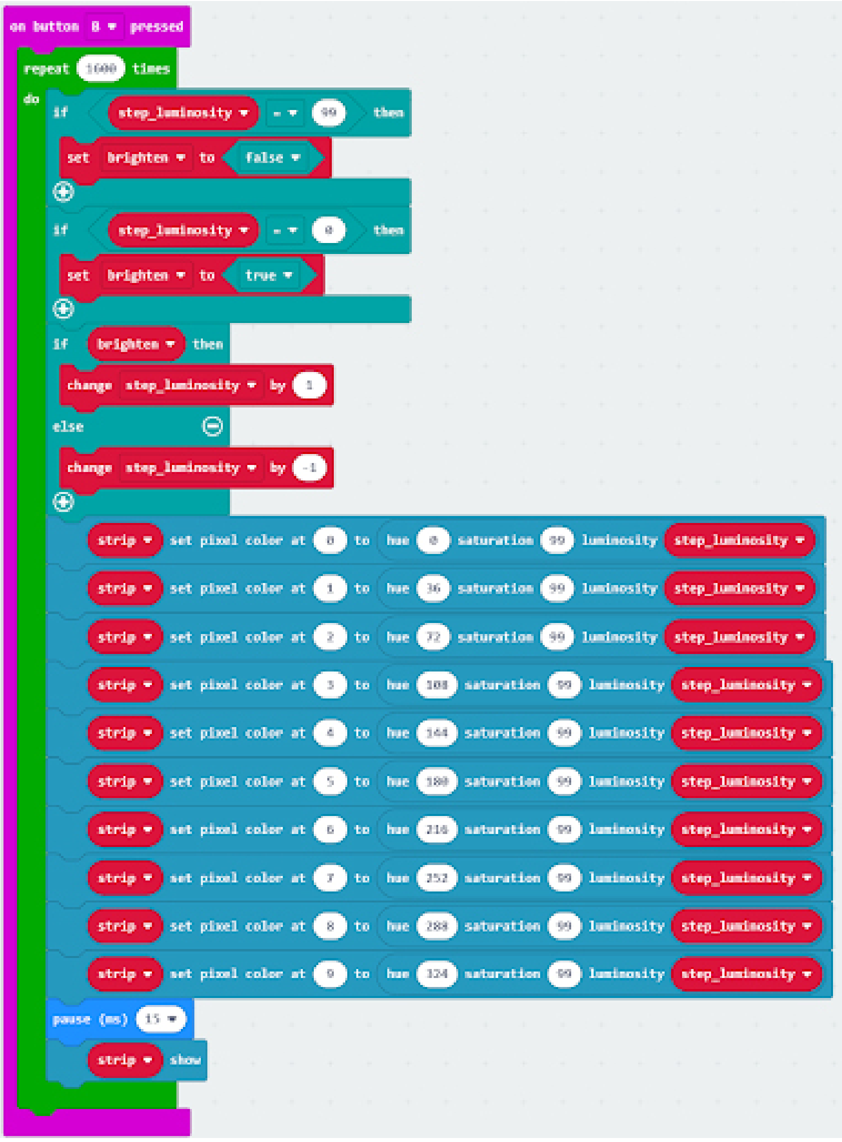
The third block of code says what will happen when button B is pressed.

Compare the button A and button B code blocks. What is similar? What is different? Write what you notice in your Engineering Notebook, and make a prediction about what will happen when you press button B.

Once you've done that, it's time to press the "Download" button.

Try pressing button A. Watch as the LEDs get brighter and dimmer. Is there a point in the cycle when the LEDs seem to brighten faster? Slower?

Now press button B. How are these colors different from when you pressed button A? Again, when do the LEDs seem to brighten faster? Slower?



How close were your predictions about what would happen? Do you understand how the code works? If not, try the debugging tool, and try experimenting with making small changes to the code. Keep track of your experiments in your Engineering Notebook.

2.B Client Memo 2

Dear Engineers,

I'm so glad to hear of your progress in learning to program LED strip lights! So you may continue to develop your design, the Anytown High School dance committee is providing each team with 10 short LED-strips and the micro:bit with breakout board, a power adapter to provide sufficient power for more LEDs, and a sheet of foam board on which to arrange the prototype visually.

Keep in mind as you create a visual arrangement for our theme of Starlight, that in the final design, the short LED-strips could be replaced with longer ones, or more short LED-strips could be added. Also, the breakout board for your micro:bit can power up to 20 LED strips.

Remember that we are looking for three or more different programs to light the LED strips to match with different types of music. We have talked as a committee and would like to clarify that all three programs should involve different (1) changes in color and (2) changes in luminosity, and at least one program should (3) show the appearance of movement.

Most importantly, remember the purpose for this design: to promote enjoyment of and create a festive atmosphere for the upcoming school dance.

Yours sincerely,

Gina Dancer, Grade 12 Student
Head of the Dance Committee of Anytown High School

Name____ Date_______________ Period __________

2.C Generating Ideas

Solution Idea #1 (write and/or sketch with labels and annotations):

Does this idea address all of the criteria? (yes/no)
For each of the criteria, explain how this idea does or does not address it.

Does this idea meet constraints? (yes/no)
For each of the constraints, explain how this idea does or does not meet it.

What are the best features of this idea?

What features of this idea need more work, and why?

Solution Idea #2 (write and/or sketch with labels and annotations):

Does this idea address all of the criteria? (yes/no)
For each of the criteria, explain how this idea does or does not address it.

Does this idea meet constraints? (yes/no)
For each of the constraints, explain how this idea does or does not meet it.

What are the best features of this idea?

What features of this idea need more work, and why?

LESSON THREE:

Lesson Objectives
Students will be able to:
- Consider criteria and constraints in planning and designing a lighting system.
- Justify design decisions in terms of criteria, constraints, results from end-user interviews, and exploration of the materials.
- Argue for a position using evidence and clear justifications.
- Collaborate effectively to develop a team plan.

Time Required
One 45-minute class period

Standards Addressed
IAS Integrated STEM
- HS.CC.3, HS.IPS.2

IAS Science and Engineering Practices
- HS.SEP.6, HS.SEP.7

Key Terms
- Evidence-based reasoning
- Criteria
- Constraints
- Prototype

Lesson Summary
Students update the Problem Definition, including criteria and constraints, to incorporate new information. They imagine and plan how they will test their prototypes to ensure they address criteria and meet constraints. They plan their designs for party lights that meet the criteria and stay within constraints specified by the client, and they justify their design decisions using evidence-based reasoning.

Lesson Background
In this lesson, teams will discuss and plan two ideas for how to address the design challenge. They should draw upon the ideas that team members came up with for homework, discuss pros and cons of the various ideas, and combine ideas to create the best idea they can come up with to make plans for a prototype. As they detail these plans, they will use evidence-based reasoning (EBR) to think deeply about their design decisions and to justify them with information about the engineering problem and their science and mathematics knowledge.

Evidence-based reasoning (EBR) refers to the engineering practice of providing rationales for design ideas and decisions. It is somewhat similar to scientific argumentation in the sense that it involves using evidence and explanations to support a statement, but it is ultimately different. In EBR, the statement being supported is an engineering design idea or decision, whereas in scientific argumentation it is a claim or conclusion about a natural phenomenon. EBR is used in the context of generating solutions for engineering problems; scientific argumentation is used to answer scientific questions about nature. Science and mathematical principles are important justification for scientific argumentation and EBR. However, EBR often also includes justifications related to context, criteria, and constraints of the engineering problem (e.g., cost, user needs, technical feasibility).

Teacher Background

Teamwork
In this lesson, students will need to work together and compromise to come up with a plan. Engineering requires effective, collaborative teamwork. This means that each member of a team should pay attention both to how they are contributing to the team and to how the team is working together. If your students are having trouble effectively managing their projects, you may wish to spend some

time on a class discussion of characteristics of effective teams. For example, you might give examples of how to respectfully discuss and critique the contributions of others.

Engineering Design Process
Students discuss the ideas they came up with for homework in their teams. Then, they discuss how to choose from and combine their ideas as they narrow down what they want to do next. Finally, they make a plan for how to create their prototypes, that takes into account evidence, criteria, constraints, and so on.

Criteria and Constraints
Criteria and constraints are important to take into account when deciding what ideas to choose and plan how to implement as prototypes.

Engineering Notebook
Students record their ideas and plans in their engineering notebooks. They should make detailed notes that will help them as they continue in the next lesson with constructing prototypes.

Vocabulary
evidence-based reasoning, prototype

Before the Activity
- Make copies of the duplication masters:
  - 3.A Evidence-Based Reasoning Part 1 (1 per student)
  - 3.B Evidence-Based Reasoning Part 2 (1 per student)

- Project or display a large version of 3.A Evidence-Based Reasoning Part 1 so students can see what needs to be included

Classroom Instruction

Introduction
1. **Revisit the engineering problem.** Remind students that in Lesson 1 they developed 1.C Problem Definition, and they also collected data from fellow students about their lighting preferences.

Lesson Materials
Per classroom
- EDP Poster

Per student
- EDP slider and jumbo paperclip
- Laptop/Chromebook/Tablet
- Engineering notebook

Duplication Masters
- 3.A Evidence-Based Reasoning Part 1 (1 per student)
- 3.B Evidence-Based Reasoning Part 2 (1 per student)

Assessment

Pre-Activity Assessment

Examine student homework from Lesson 2 to ensure that students have a good sense of how to come up with relevant and appropriate ideas that address the challenge including criteria and constraints.

Activity Embedded Assessment

Circulate to observe and support students as they work in teams. Ensure that all team members are participating and taking each other's ideas seriously. Watch for confusion about how to connect ideas and plans to evidence, as well as criteria, constraints, and how to test them. Also encourage students to make their plans specific enough to be useful in Lesson 4, when they create their first prototype.

Post-Activity Assessment

Review team plans. Make sure that sufficient detail is included to make the plans useful as students work on constructing their prototypes in the next step. Check that students refer to evidence, criteria, and constraints when they justify their design decisions.

2. **Remind students of where they are in the engineering design process.** For homework, students came up with at least two ideas each. Point to the EDP poster. Today, students will work with their teams to Plan their designs.
3. **Display 3.A Evidence-Based Reasoning for all students to refer to as you discuss it.**
4. **Introduce evidence-based reasoning. Say:** *Engineers use evidence to make decisions. To help you with evidence-based reasoning, we have these questions, which you should address in your engineering notebooks.*
5. **Ask students to think about how they would update the Problem Definition.** Point to the 3.A Evidence-Based Reasoning Part 1. *What documents and information do we already have that we can use to improve our problem definition?* Students will discuss this in teams, and then the class will revisit the problem definition later in the lesson.
6. **Discuss what it would mean to test a prototype.** For now, students should think generally about how they will test the prototypes, because that will help them to decide what to include in their prototypes. Make sure that students know that they will improve the testing procedure later, before they test their finished prototypes.
7. **Explain how to go about creating a plan.** Once students have completed 3.A Evidence-Based-Reasoning Part 1, they will need to decide what idea or ideas they are most interested in turning into a prototype. They should review all team members' ideas that they came up with for homework, and then decide how to combine those ideas into something they all agree will work best.
8. **Show 3.B Evidence-Based Reasoning Part 2,** where they will expand upon their team's chosen idea or ideas, to create a full plan to create and test their prototypes.
9. **Explain that the goal for today is to consider what they know, decide on ideas, and complete a plan to create a prototype and test it.**

Activity

10. **Work in teams to complete the 3.A Evidence-Based Reasoning Part 1.** Teams should rewrite their Problem Definition in their engineering notebooks and also write down their assumptions, considering the new information they have gathered from interviews and trying out programs, and that they have received from Client Memo 2. Teams should create a first draft of the testing plan.

11. **Work in teams to complete 2.C Generating Ideas to discuss and integrate ideas.** Teams will need to choose two ideas to plan out further, to prepare to try out with a prototype. Teams can use 2.C Generating Ideas as a scaffold to elaborate on their new ideas. This can be a rough sketch; they will plan in more depth using 3.A Evidence-Based Reasoning. All work should go into engineering notebooks.
12. **Work in teams to complete 3.B Evidence-Based Reasoning Part 2.** Now, teams should make their ideas into concrete plans, with as much detail as they can think of before getting to work on the code again. They may also wish to revisit the testing plan from part 1.
13. **Remind students of the importance of criteria and constraints.** Have students refer back to prior information from end-user interviews and client memos, as well as their notes from experimenting with MakeCode and the micro:bit setup as they work.

Closure

14. **Conduct a class discussion about the problem, criteria and constraints, and testing procedures.** Have students share their ideas, and work to come to some consensus, to ensure the class is on the same page. You may wish to project 3.A Evidence-Based Reasoning Part 1 and create a class-consensus document.
15. **Connect to the EDP. Ask:** *Do you feel that you did good work today in creating a Plan? What will we be doing next?* Explain that students will use their plans in the next lesson to create their prototypes. Remind them that they are trying to develop the best possible solution to the client's problem, and as is the case for professional engineers, the first idea or solution is rarely the best.

Name______________________________________ Date______________ Period __________

3.A Evidence-Based Reasoning Part 1

1. Restate the Problem Definition, including criteria and constraints, given what you learned in Lesson 2 from Client Memo 2 and from learning to program the LED strips using micro:bits.

2. How will you test your prototypes?
    a. Test for criteria__________________:
    b. (more tests for criteria)

Name______________________________________ Date______________ Period __________

3.B Evidence-Based Reasoning Part 2

In your engineering notebook, answer the prompts below for at least one idea to address the client's design challenge (a party lighting system).

Idea #____

1. Describe your design plan. Include drawings and dimensions. Label materials. Highlight interesting features. Explain your programming approach. Include any relevant observations, evidence, or data to support your design plan.

2. Describe why you think your design will be successful. Refer to the problem, criteria, constraints, idea, and data or evidence, including things you have already tried.

LESSON FOUR:

Lesson Objectives
Students will be able to:
- Develop a prototype that addresses criteria and satisfies constraints.
- Collaborate in teams, distributing roles and efforts as needed to achieve a goal.

Time Required
One or two 45-minute class periods

Standards Addressed
IAS Mathematics
- HS.PS.1, HS.PS.2

IAS Science and Engineering Practices
- HS.SEP.6

IAS Integrated STEM
- HS.CC.2, HS.DM.3

Key Terms
- Luminosity
- Microcontroller
- Power adapter

Lesson Summary

Students refer to their plans from Lesson 3 as they program their micro:bits to provide for multiple options of light patterns. They create programs that can match the intensity and mood of a variety of different songs and types of music.

Lesson Background

In this lesson, students will create their first prototype. The time they spent planning should be helpful to them as they decide how each of them will take on roles in this activity. However, there is still plenty they need to figure out about how to change the code to do what they want it to do.

This will be the first time that students are using more than one LED strip with their micro:bits. They will need to plug in their breakout boards using the 9V power adapter in order to supply sufficient current to power more than one LED strip. Make sure that students remember to plug the micro:bit (NOT the breakout board) into the computer. The new LED strips can be connected either to each other (end-to-end) or into pins on the breakout board: P0 (pin 0) was used in Lesson 2; now they can plug in to pins 1-9 in the same way, and control them using new (or copied) blocks of code.

There are a number of approaches that students can take if they are feeling stuck at any point. One approach is to have someone else look at their code while they explain it. Often that will surface small errors that are getting in the way of functionality. A second approach is to use the debugger in MakeCode. Students can place breakpoints in the code where it will stop, and/or use the “step through” option, so they can see what the code is doing step by step, check the values of variables, and ensure that things are happening in the expected order. A third approach is to conduct an internet search to see if others have had similar issues, and how they solved them.

Try it Out: Create a Prototype

Teacher Background

Teamwork

In this lesson, teams will need to collaborate to code their micro:bits according to the plan they made in Lesson 3. Engineering requires effective, collaborative teamwork. This means that each member of a team should pay attention both to how they are contributing to the team and to how the team is working together. If your students are having trouble effectively managing their projects, you may wish to spend some time on a class discussion of characteristics of effective teams. For example, you might give examples of how to respectfully discuss and critique the contributions of others.

Engineering Design Process

This is the Try step: students put their plan into action as they build their prototypes.

Engineering Notebook

Students should write down their ideas, frustrations, and observations in their engineering notebooks. They can draw what they observe too!

Vocabulary

luminosity, microcontroller, power adapter

Before the Activity

Decide whether you have the capacity to store student projects on the poster-sized foam board at this time, or if you should keep them in ziplock bags or small bins.

Figure out where to plug in the power strips so that each team has access to one to plug in their breakout board.

Prepare additional materials for each team.

Make copies of the duplication masters:

- 4.A How to Power Multiple LED Strips (1 per team)
- 4.B Coding Support Guide (1 per team or student)
- 4.C Troubleshooting and Debugging (1 per team or student)

Lesson Materials

Per classroom

- EDP Poster
- 4-5 Power Strips, distributed so each group has access to one.

Per Group (3 per group)

- 1 micro:bit with Keyes breakout board
- 10 LED strips with Dupont connector
- USB-A cable
- 9V power adapter
- [Optional] Poster-sized foam board
- [Optional] Push-pins, paper clips, or masking tape to hold the LED strips in place temporarily

Per Student

- EDP slider and jumbo paperclip
- PC/Laptop/Chromebook/Tablet
- Engineering notebook

Duplication Masters

- 4.A How to Power Multiple LED Strips
- 4.B Coding Support Guide
- 4.C Troubleshooting and Debugging

Assessment
Activity Embedded Assessment
Circulate to observe and support students as they work in teams. Ensure that all team members are participating and taking each other's ideas and contributions seriously. Watch for struggles with creating code that works. You may need to prod students to use the guide 4.C Troubleshooting and Debugging, or arrange that teams exchange code to try to find each others' bugs.

Post-Activity Assessment
Examine engineering notebooks to look for students' observations and notes about their experiences. Examine students' code and test it to see if it's doing what the teams wanted.

Classroom Instruction

Introduction

1. **Remind students of where they are in the engineering design process.** Point to the EDP poster. Today, students will Try their designs.
2. **Pass out materials to teams.** Show students how they will plug in their breakout boards before adding more LEDs to the lighting systems.
3. **Remind students that part of their design will be a visual arrangement of LED strips** on a poster-sized foam board. For now, students should work on getting the code to work and sketch their visual layouts in their engineering notebooks; later they will be given foam board to which they can attach their LED strips**.**
4. **Pass out and briefly review 4.A How to Power Multiple LED Strips.**
5. **Pass out and give an overview of 4.B Coding Support Guide.** Tell students they will want to save their code and make a copy before making major changes. 4.A Coding Support Guide has instructions for how to do this, as well as other hints and guidance to help with coding. They can also use the Guide they were given for Lesson 2: Explore and the sample code that came with it to start their projects.
6. **Pass out and briefly review 4.C Troubleshooting and Debugging.** This will be helpful if students get stuck.

Activity

7. **Have students work to implement their preferred plan** from 3.B Evidence-Based Reasoning Part 2.
8. **Circulate and assist as needed.** Be careful to avoid telling or showing students what to do. Instead, ask questions designed to get them thinking and explaining their reasoning.
9. **[Optional] If you decided to provide foam board during this lesson, pass it out to teams** after they have gotten their code partly functional and allow them to attach LED strips temporarily using push pins between the red, yellow, and black wires, or else paper-clips or masking tape. Students should take care not to pierce the insulation on the wires.

Closure

10. **Conduct a class discussion about the day's activities. Ask:** *What went well? What was challenging? How hard was it to program color changes? Brightness changes?*
11. **Connect to the EDP.** In this lesson, students work on the Try stage, making prototypes. However, going through the EDP isn't always (or even usually) something that engineers do in order! Frequently, there's a need to backtrack, particularly when new information surfaces.

4.A How to Power Multiple LED Strips

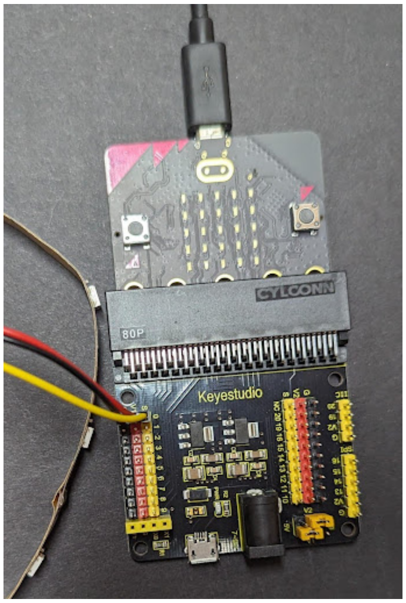
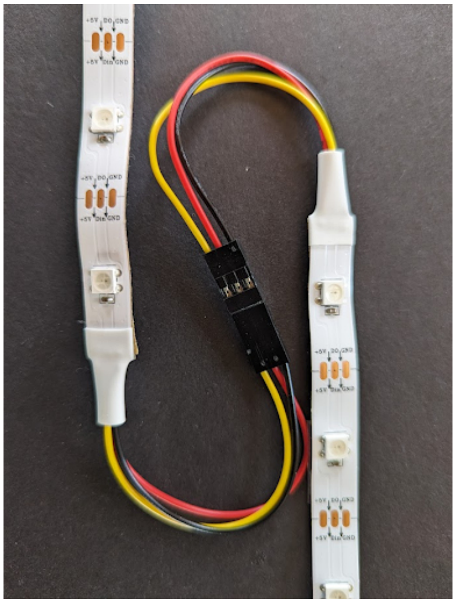
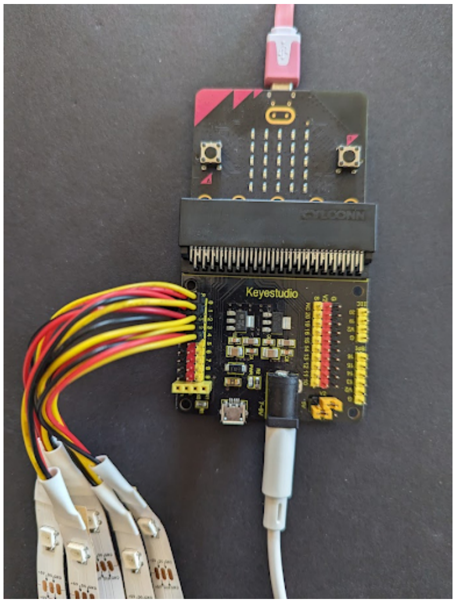
Congratulations! You are now ready to power and control multiple LED Strips. This guide will show you how to set up the hardware and add to your program.

Adding LED Strips. First, you'll need to add more LED's to your hardware setup. You should be starting with your micro:bit, breakout board, and first LED strip looking like Image 1.

You can connect new LED strips in two ways:

1. You can make an LED strip longer by connecting multiple strips end-to-end (Image 2)
2. You can add strips to the breakout board (Image 3).

Either way, make sure you check that you **line up the wire colors** with each other and with the breakout board colors!

Image 1	Image 2	Image 3
		

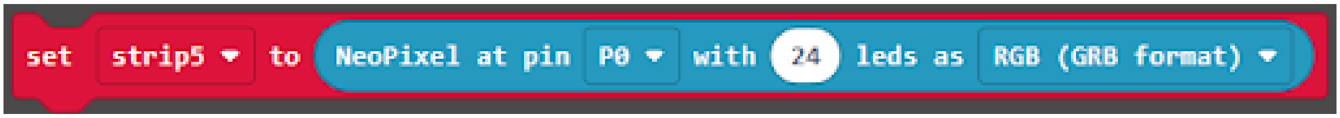
Adding power. As soon as you add more than one LED strip to the micro:bit, you're going to need to add power – the amount of power coming through the cable from your computer to your micro:bit won't be enough.

1. Plug in the 9v power adapter to your breakout board – you can see what this looks like in Image 3.
2. Continue to plug your USB-A cord into your micro:bit (not the breakout board) – you'll need it to download programs.
3. Once your programs are done, the micro:bit can run the programs without the connection to the computer – but it will still need power, so leave the 9v power adapter plugged into the wall or power strip.

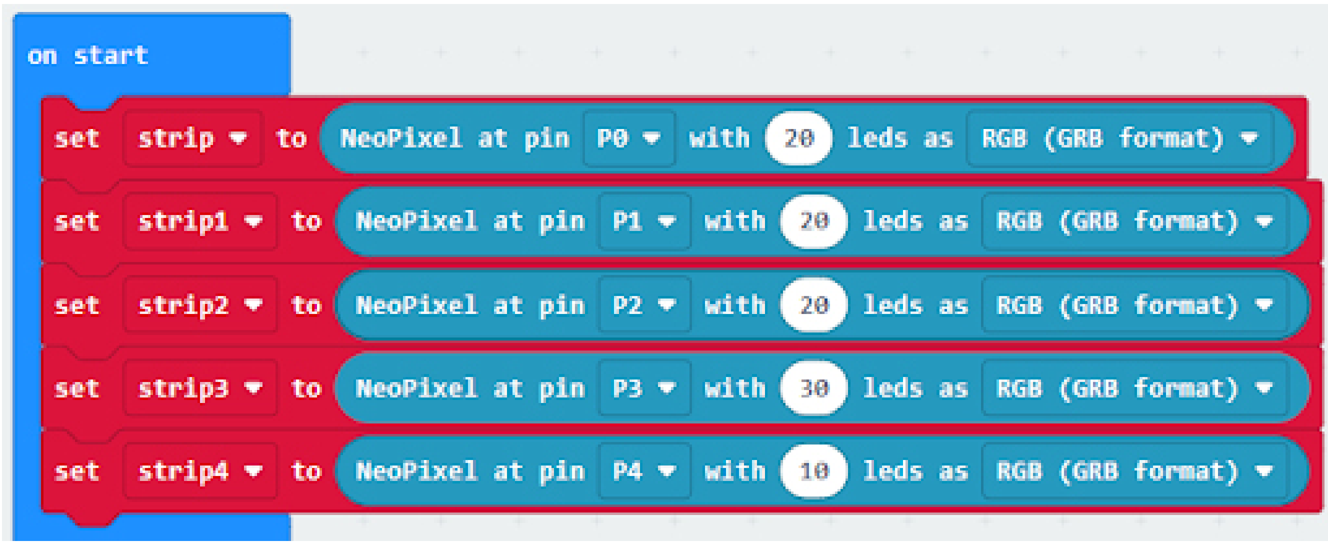
4.A How to Power Multiple LED Strips

Adding code for new or longer LED strips. Now you need to tell those new and/or longer strips what to do! First, you need to let the micro:bit know that there are multiple LED strips:

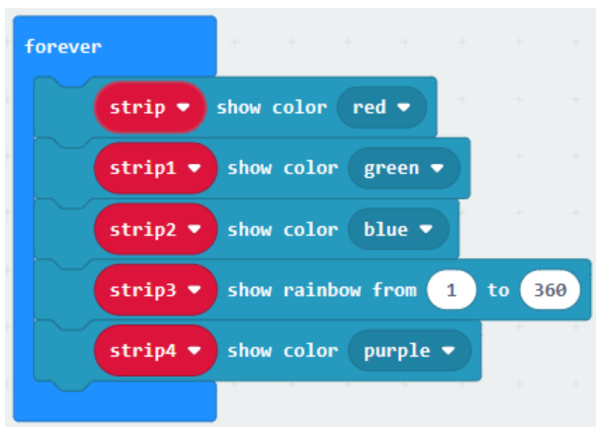
1. For each LED strip, add the following block to **on start** from the neopixel menu:



2. You will need to make each of these blocks different.
   a. Give each strip a different name: for example, strip1, strip2, strip3, strip4, strip5.
   b. Give each strip a different pin address. If you have the LED strips arranged as in Image 3, this will be P0, P1, P2, P3, P4.The pin numbers are printed in yellow on the breakout board if you need to check which ones you are using.
   c. Enter the correct number of LEDs for each strip. The strips you have are 10 LEDs long, so if you plug two of them together, the strip length will be 20.
   d. All of the strips should be set to **[RGB (GRB format)]**.
   e. Here's an example for a setup with 3 20-LED strips (each with 2 short strips plugged together), one 30-LED strip, and one 10-LED strip.



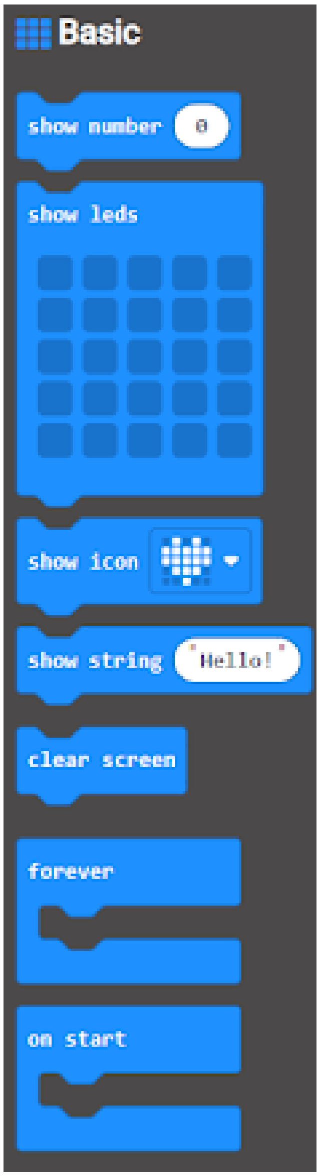
3. Now, every time you tell a strip what to do, you will have to specify which strip you mean in the drop-down. Here's a program that sets all of the strips to a different color or colors:



4.B Coding Support Guide

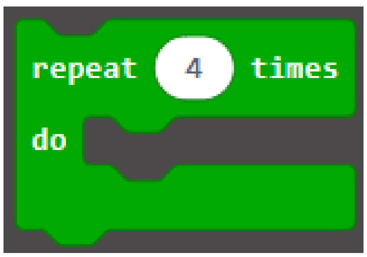
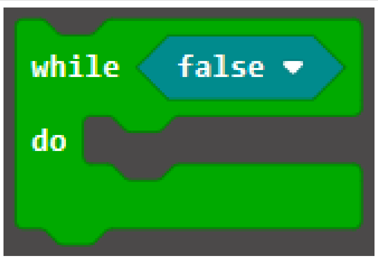
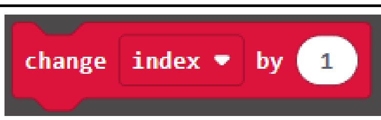
Coding is easiest when you start with a program that does something similar to what you want to do! We suggest you begin with one of the programs provided to you in Lesson 2. Copy it, and make changes to it.

Here's what's in the Basic menu:

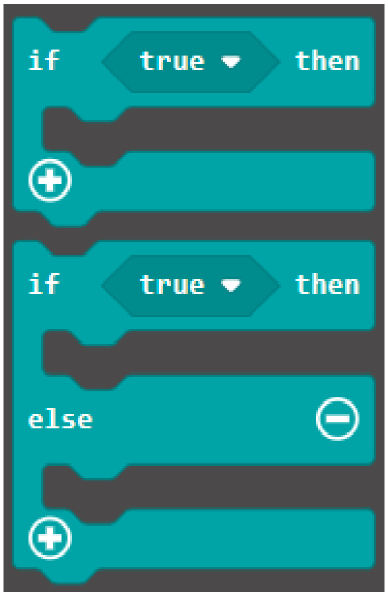
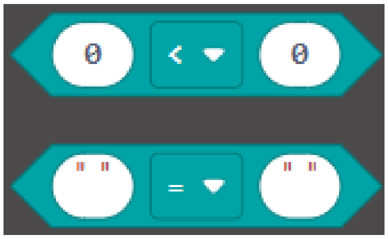
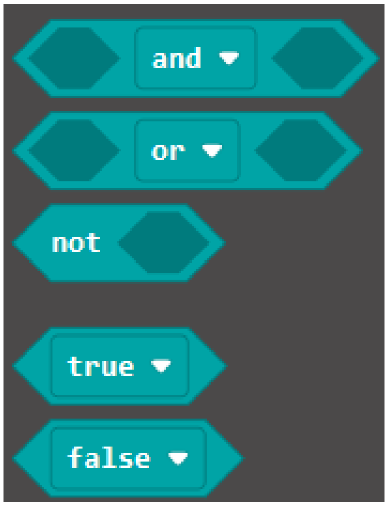
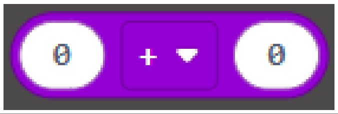


- **show number (#)** displays the number in the ( ) on the micro:bit. This could be a way to keep track of things you are counting, a way to give numeric information to the user about what the micro:bit is doing, or a way to give data to the user, just for a few examples.
- **show leds** allows you to show a picture on the micro:bit by selecting which LEDs to turn on in the grid.
- **show icon** has some preset images to choose from to display on the micro:bit (like **show leds** does).
- **show string (" ")** prints the characters you enter on the micro:bit. If there is more than one character, they will scroll by. The default string is 'Hello!'
- **clear screen** turns off all the micro:bit LEDs, which can be useful if you want to show 2 or more things in a row.
- **forever** is a block you use for stuff you want to keep happening over and over and over: for example if you want to use show leds to show a graph of how bright the environment is at any one time.
- **on start** is where you put things you want to happen just when the micro:bit turns on (and not again). Here is where you will (probably) want to set all your variables to an initial value.
- **pause** lets you select how long you want the program to wait. This can be useful so that people can see what is happening. The micro:bit can do things very quickly!

Here's some useful code blocks from the Input, Loops, and Variables menus:

	• **on button [A] pressed** tells the micro:bit what should happen when one or both buttons are pressed. You choose which button (or button) using the dropdown: A, B, or A+B. Those are the only buttons available, but if you need more there are creative ways to add buttons: see, for example https://youtu.be/GEpZrvbsO7o
	• **button [A] is pressed** is a **boolean** (true or false) block. It can be plugged into another block, for example an if or **while** block. It will fit into any block that has an opening with pointed ends.
	• **repeat (#) times… do** allows you to repeat some set of things for a specific number of times (which you enter into the bubble). This is particularly useful if you know exactly how many times to repeat, OR you can make a **variable** that can go in the bubble, and adjust it in the program.
	• **while <boolean>... do** repeats the code inside for as long as the condition in the pointed opening is true. You can put any block with pointed ends here – for example, the **button [A] is pressed** block. This is particularly useful if you want something to keep happening as long as some condition is true.
	• **every (#) ms** will do everything inside the block over and over, with a pause between each time that is <#> milliseconds long. As with all bubble-shaped openings, you can put a **variable** in here too.
	• **set [variable] to (value)** sets the value of a variable to whatever you want: either a number or a "string" (in quotes). You have the option to name a new variable – look in the drop-down menu.
	• **change [variable] by(value)** changes the value of a numeric variable by the amount of the number (or variable) in the bubble.

Here's some useful things from the Logic and Math menus:

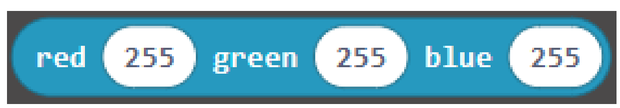
	• **if <boolean> then** will ONLY do what's inside the block if the boolean condition in the pointy opening is true. You can plug any pointy-ended **boolean** block in here. Often you will want to use some of the Logic **boolean** blocks here, so you can compare different variables, or compare a variable to a specific value. • **if <boolean> then… else** is a lot like the block above it, but instead of just skipping the **if <boolean> then** section of code if the **boolean** condition is false, it runs the **else** block of code. That way, you can have the block run one set of code if the condition is true, and a different set of code if it's false.
	All pointy-ended blocks (like those below) are called **booleans:** they will always either be true or false.
	• **<#> <compare> <#>** lets you compare 2 numbers to see if they are equal, not equal, or one is greater than the other. You can put variables in the bubbles. • **<" "> <compare> <" ">** is similar but instead of numbers you compare strings of characters.
	• **<boolean> [and] <boolean>** will be true ONLY if both of the boolean conditions are true. • **<boolean> [or] <boolean>** will be true if either OR both of the boolean conditions are true. • **not <boolean>** will be true if the **boolean** condition is false. • **<true>** is always true. • **<false>** is always false.
	There are lots of mathematical operations you can do with Math blocks. If you would like to learn more go here: https://makecode.microbit.org/reference/math

Here's some useful things from the Neopixel menu:

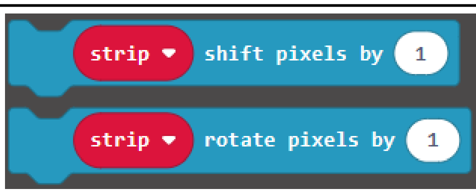
- **set [strip] to (Neopixel at pin [P#] with (value) leds as [Format])** will tell the micro:bit that you have plugged in an LED strip (or block or ring) with the number of leds you enter for the **(value)** at the **[pin number]** you select. You can plug in a bunch of LED strips, each at a different pin number, but you will have to give them all different (strip) names.
- **set [range] to ((strip) range from (start value) with (number value) leds)** gives a name to a contiguous subset of the leds on the strip you choose. That way you can have different ranges doing different things on the same strip.

- **(strip) show color (color)** will set all the leds in the strip to the color you name. You can use a named color in the drop-down, or you can use a **color definition** block (see below).

- **(strip) show** is a block you need to use with a **set color** block, to tell the micro:bit to change the leds to be the **set color** color.
- **(strip) clear** turns off all of the leds on the strip.



- **hue (value) saturation (value) luminosity (value)** is one kind of **color definition** block. Find the values for each color here: https://www.quackit.com/css/color/charts/hsl_color_chart.cfm
- **red (value) green (value) blue (value)** is another kind of **color definition** block. You can find this bubble in the **...more** menu just below the **Neopixel** menu. Find the values for each color here: https://www.quackit.com/css/color/charts/rgb_color_chart.cfm

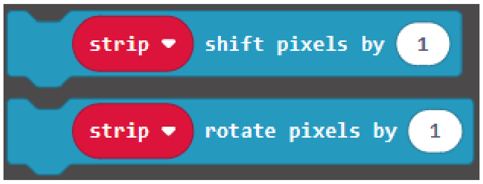


- **(strip) shift pixels by (value)** moves the color values for all the leds in a strip over by **(value)**.
- **(strip) rotate pixels by (value)** moves the color values for all the leds in a strip over by **(value)** over and over again (so it looks like they are moving).

4.B Coding Support Guide

- **set [strip] to (Neopixel at pin [P#] with (value) leds as [Format])** will tell the micro:bit that you have plugged in an LED strip (or block or ring) with the number of leds you enter for the **(value)** at the **[pin number]** you select. You can plug in a bunch of LED strips, each at a different pin number, but you will have to give them all different (strip) names.
- **set [range] to ((strip) range from (start value) with (number value) leds)** gives a name to a contiguous subset of the leds on the strip you choose. That way you can have different ranges doing different things on the same strip.

- **(strip) show color (color)** will set all the leds in the strip to the color you name. You can use a named color in the drop-down, or you can use a **color definition** block (see below).

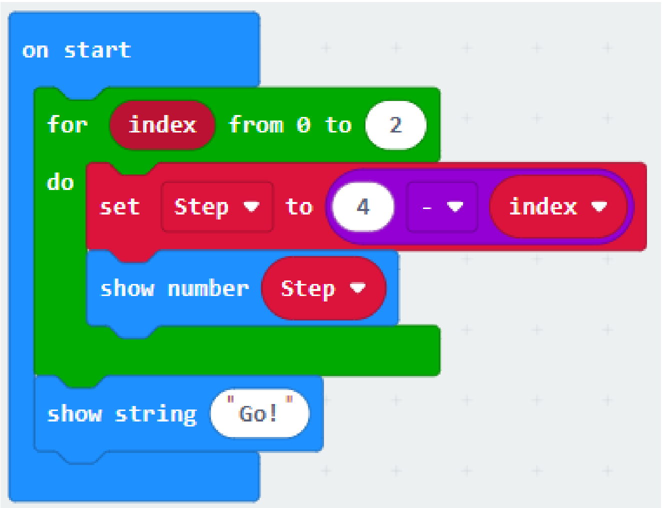


- **(strip) set brightness (value)** sets the maximum brightness of the strip to **(value).** A value of 255 is the brightest the LEDs can get. Note: this does not adjust the brightness of the LED strip over time! You can only set it lower once you set it, not higher. Yes, this is confusing.
  (find in the **...more** menu)

Well, it's happened. Your LED strips are not behaving as you expected, or you're getting an error message. Never fear, help is here!

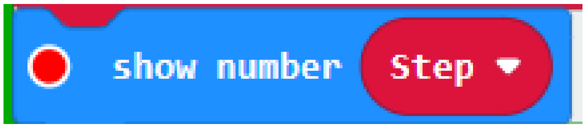
The first thing to try is to ask someone else to check your code. Sometimes all you need to figure out what's wrong is for fresh eyes to look it over.

The second step, if that didn't work, is to try the debugger. MakeCode has a friendly debugger that can help you see what's happening at each step in your line of code. Let's look at this little "countdown" program:

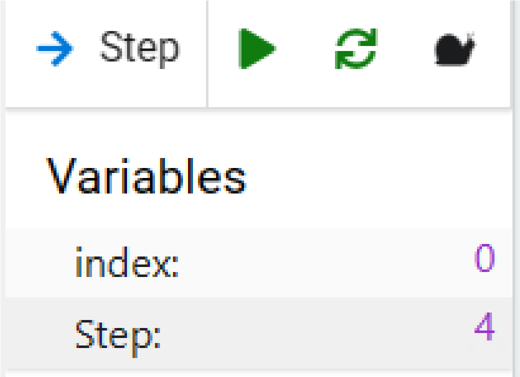


When I press the button in the simulator (the image of the micro:bit on the left), the display shows "4 3 2 Go!" How do I fix it to say 3?

1. Press debug:
2. The debugger appears next to the simulator.
3. I put a "break" on the **show number (step)** block by clicking the gray dot to turn it red:



4. Now I have a choice. If I press → Step the program will proceed one step at a time, and I can see what happens with every step in the Variables box.
5. If I press ▶ or the program will proceed at regular or slow speed until it hits the block with the red dot, so I can see what the variable values are at that point.



Doing this, I can see that the first time I run the program, **step** is set to "4" instead of "3" because the first **index** is "0" and my math block calculates step = 4 - 0. Oops! I change the block to say **set step to (3) - (index)** and it's fixed.

If all else fails, you can search or ask for help online at one of these sites:
https://forum.makecode.com/ – this is the MakeCode forum
https://github.com/microsoft/pxt-microbit – you can find talk about MakeCode here too!

LESSON FIVE:

Lesson Objectives
Students will be able to:
- Explain how human perception of light can be characterized as non-linear.
- Conduct an experiment to determine how the human eye perceives changes in LED luminosity across colors and light levels
- Collect data and use it to construct a graph of luminosity
- Come up with ideas for how to even out the changes in luminosity for human perception

Time Required
One 45-minute class period

Standards Addressed
IAS Mathematics
- AII.FF.4, AII.MFD.3

IAS Integrated STEM
- HS.CC.2, HS.DM.1, HS.DM.3, HS.IPS1

IAS Science and Engineering Practices
- HS.SEP.2, HS.SEP.3, HS.SEP.4

Key Terms
Weber's Law
Physical stimulus
Perception

Lesson Summary
Students conduct an experiment to examine how LED changes in brightness are perceived for a variety of colors. They graph the data they collected on brightness perceived as a function of luminosity. Then, they use the data they've collected to think about how they would program a "smooth" change in luminosity. In the next lesson they will learn about logarithms, and in Lesson 7 they will have the opportunity to implement their ideas to smooth out the luminosity.

Lesson Background
In this lesson, students use a given program to observe and record their perception of changes in luminosity at small intervals, starting with low-light luminosity values and ending with the brightest values. Full instructions for the experiment are given in 5.B Experimenting with LED Strip Luminosity. Students will need to take on roles during the experiment, and work together to collect the data in a reliable manner.

There are four programs because your class will be experimenting with four LED colors: white, red, green, and blue. You will need to decide how to divide up the teams so that all four experiments are completed by at least one team (preferably two or more). Each team will complete the designated experiment and record their data in a copy of the spreadsheet template provided, LED Luminosity Change Data. This is provided as a Google doc, though you can port the program to Excel or another spreadsheet program if you prefer.

Once all teams have collected their data and copied it to Desmos, conduct a class discussion about what they noticed, what they learned, and what ideas they have for how to make the luminosity changes more smooth and even.

Teacher Background
Teamwork
This activity includes 3 structured roles; students will need to negotiate who will take on which role.

Engineering Design Process
This is a return to the "Learn" step, so students can figure out more about luminosity changes and how they are perceived by the human eye.

How do we make luminosity changes more even?

Engineering Notebook
Students should take notes on their observations and thoughts about what they see in their engineering notebooks.

Vocabulary
Weber's Law, physical stimulus, perception

Before the Activity
- Try the activity in this lesson yourself so you can know how to address possible difficulties that students might have with it.
- Each team should work with one of the four available colors: white, red, green, and blue; decide how you will distribute color assignments so that at least 2 teams conduct the experiment for each of the colors.
- For students to do this experiment, it should be fairly dark in the classroom. Figure out how you will make sure this can happen.
- We call for students to use the 9V power adapters in this lesson, even though they conduct the experiment with only one LED strip, because the LEDs will be brighter and differences easier to see if the power adapter is used.
- Each team will need to copy and rename the Google Sheets template *LED Luminosity Change Data*.
- Because this and further lessons utilize Demos in some capacity, students should create an account to be used throughout this engineering design unit.
- If your students don't have much experience with Desmos, you may want to introduce them to it briefly before beginning this lesson. Show them how to paste in a data table from a spreadsheet, and how to write in an equation. There's a lot of great info and hints that they can access with the (?) button in the upper right. Keyboard shortcuts are also good to know – ctrl+^ for superscripts, and ctrl+_ for subscripts are good to start with. Later you can show students more about how to fit equations to data.
- Make sure your students know how to share their graphs with you, so you can display them for the class. You might have them post the link to their chart in your student learning system, or email it to you.
- Make copies of the duplication masters:
    - 5.A Client Memo 3 (1 per team)
    - 5.B Experimenting with LED Strip Luminosity (1 per team or student)

Lesson Materials

Per Classroom
- EDP Poster

Per Group (3 per group)
- USB-A cable
- Micro:bit
- Keyes breakout board for micro:bit
- Short LED strip
- 9v power adapter
- Sheet of black construction paper
- Google Sheets template: LED Luminosity Change Data

Per Student
- EDP slider and jumbo paperclip
- Computer
- Engineering notebook

Duplication Masters
- 5.A Client Memo 3
- 5.B Experimenting with LED Strip Luminosity

Assessment

Pre-Activity Assessment

Observe student contributions to the class discussion of 5.A Client Memo 3, to see if students can use mathematical vocabulary to talk about Weber's Law and make connections to what they noticed about LED strip luminosity changes in Lessons 2 and 4.

Activity Embedded Assessment

Circulate and observe how students conduct their experiments, collect data, and construct a graph of their data.

Post-Activity Assessment

Examine teams' work in the Luminosity Data spreadsheets and Desmos, as well as observations and hypotheses students made in their engineering notebooks.

Classroom Instruction

Introduction

1. **Pass out the engineering notebooks.**
2. **Display 5.A Client Memo 3 on the board for students.** Ensure that students understand the information provided.
3. **Ask:** *Would this new information explain anything we've noticed about LED strip luminosity changes thus far?* Help students to use the terms linear and non-linear in discussing the information in the memo.
4. **Explain that today, teams will conduct an experiment.** When new information surfaces, engineers cycle back to earlier stages of the engineering design process. Indicate the second step of the engineering design process on the EDP poster. For this experiment, teams will collect data on how they perceive differences in brightness as they step up the luminosity of the LED's bit by bit.

Activity

5. **Give teams the materials needed for the activity**, including: 5.B Experimenting with LED Strip Luminosity, 1 micro:bit, 1 USB-A cord, 1 Keyes breakout board, and 1 rainbow LED strip.
6. **Let students know that you are available to assist them** as they work through the directions.
7. **Have each team work through the student guide**, conduct the relevant tests, record data in the Google Sheet template, graph data in Desmos, and record what they are learning in their engineering notebooks. Circulate and provide assistance as needed, to enable students to figure out how to solve the mini-challenges in the guide.

How do we make luminosity changes more even?

Closure

8. **Ask what students learned from their experiments.** *How would you describe your perception of the changes in brightness from the lowest luminosity levels to the highest? Were the luminosity changes similar or different between white, red, green, and blue? Do your findings seem to bear out the information you received in 5.A Client Memo 3?*
9. **Ask teams about their ideas** to make the perception of changes in brightness more even.
10. **Conduct a class discussion of students' experiences with the activity.** *What was difficult? What was surprising? What was interesting? Do students agree that this was part of the "Learn" step in the engineering design process?*

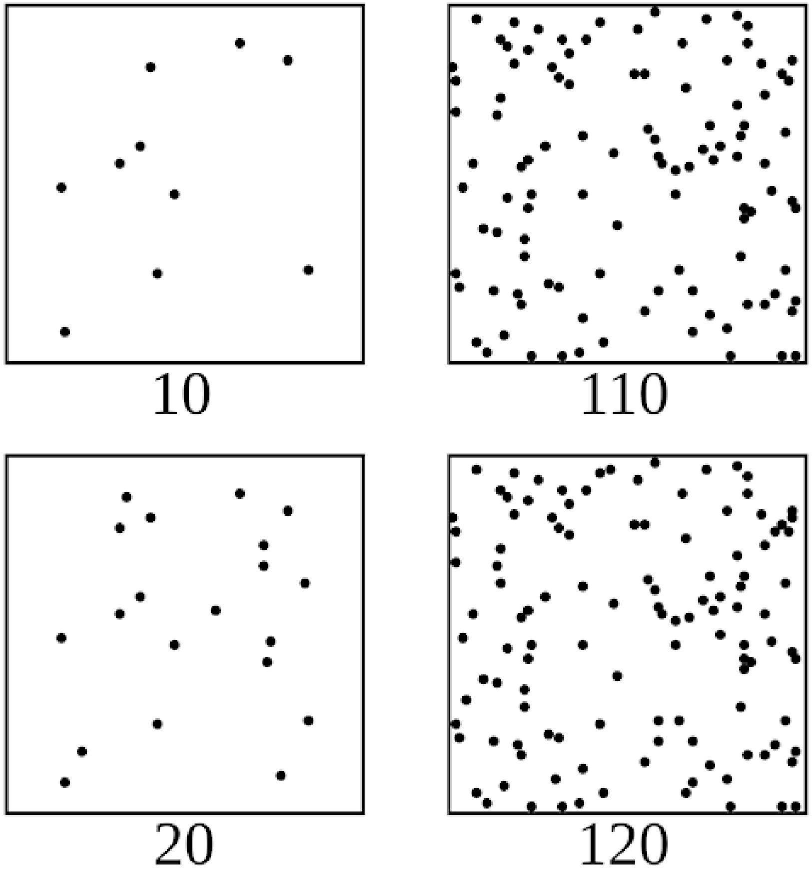
5.A Client Memo 3

Dear Engineers,

I heard that you are making good progress on prototypes for our upcoming dance party. I am excited to see them!

I learned something today in my psychology class that might help you figure out why the LED strips pulse so strangely when you make them brighter and dimmer. It has to do with something called Weber's Law, which says that the relationship between the actual change in a physical stimulus (like light) and the way a human being perceives that stimulus is not linear.

Instead, the same difference in the actual stimulus becomes harder and harder to perceive as the stimulus gets stronger. Whether we are holding weights, or listening to music, or looking at light, it's easier to perceive a change when there is just a little bit of stimulus than when there is already a lot. It works with all kinds of stimuli.



For example, it's easier to see the difference between 10 and 20 dots in the two boxes on the left, than it is to see the difference between 110 and 120 dots in the two boxes on the right.

This makes me think that, in order to make the changes in LED strip brightness look smooth and even, we first have to figure out how we perceive the changes. Would you please investigate?

Yours sincerely,

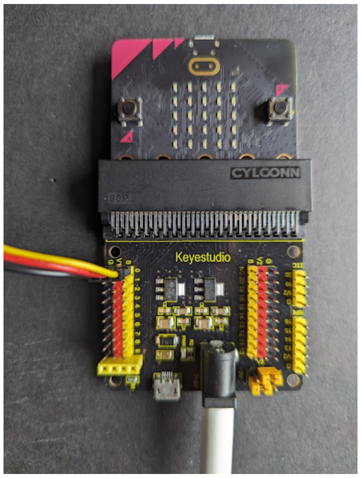
Gina Dancer, Grade 12 Student
Head of the Dance Committee of Anytown High School

5.B Experimenting with LED Strip Luminosity

Part 1: Set up

The purpose of this activity is to experiment with different luminosity levels of the LED's using your micro:bit. Your job is to answer these questions:

A. Why do the LEDs on the LED strip seem to brighten and dim sometimes slower and sometimes faster?
B. Can we adapt the way the micro:bit changes luminosity levels so it looks more even as the LEDs brighten and dim?

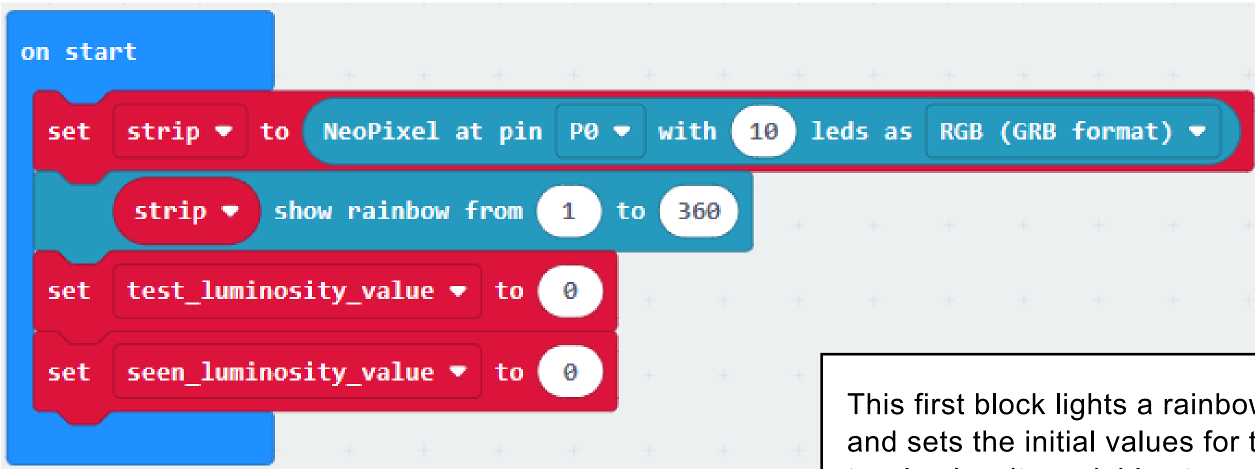
Setting up this time will be quite similar to last time. 1. Tuck the micro:bit into the top of the breakout board, if it isn't already there. 2. Plug the end of the LED strip with holes into the top left set of pins on the breakout board, carefully matching the colors of the wires to the colors of the pins. 3. Plug the small end of the USB-A cable into the top of the micro:bit. 4. Place the circuitry on a sheet of black construction paper. 5. Plug the 9V adapter into the breakout board.	

Depending on which color you are going to test, open the matching URL (or use the QR code) to open the code for experimenting with that color. The link will open a project named "LEDStrip_Compare_ Luminosity_<color>." Then add it to your MakeCode account by pressing the "Edit Code" button. Also open and make a copy, personalized for your team, of the following Google Sheets template: https://tinyurl.com/Luminosity-data	White	https://tinyurl.com/Guide-White	
	Red	https://tinyurl.com/Guide-Red	
	Green	https://tinyurl.com/Guide-Green	
	Blue	https://tinyurl.com/Guide-Blue	

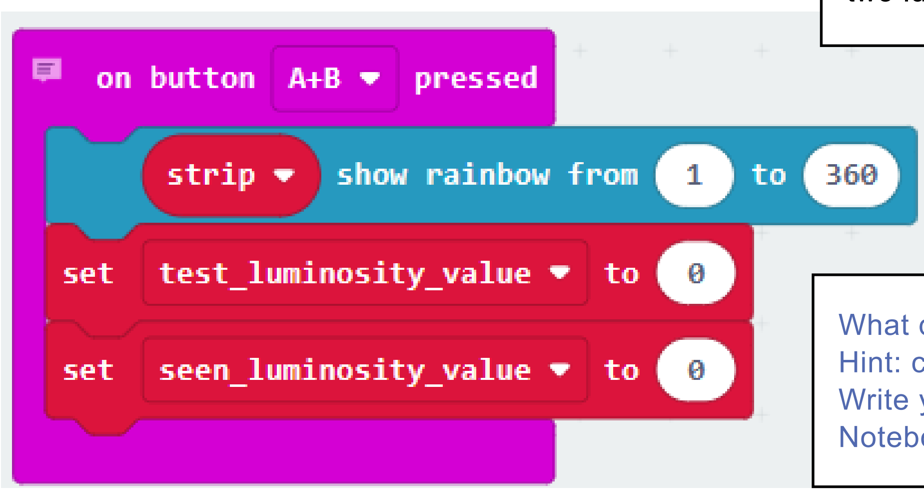
5.B Experimenting with LED Strip Luminosity

Part 2: Understand the program

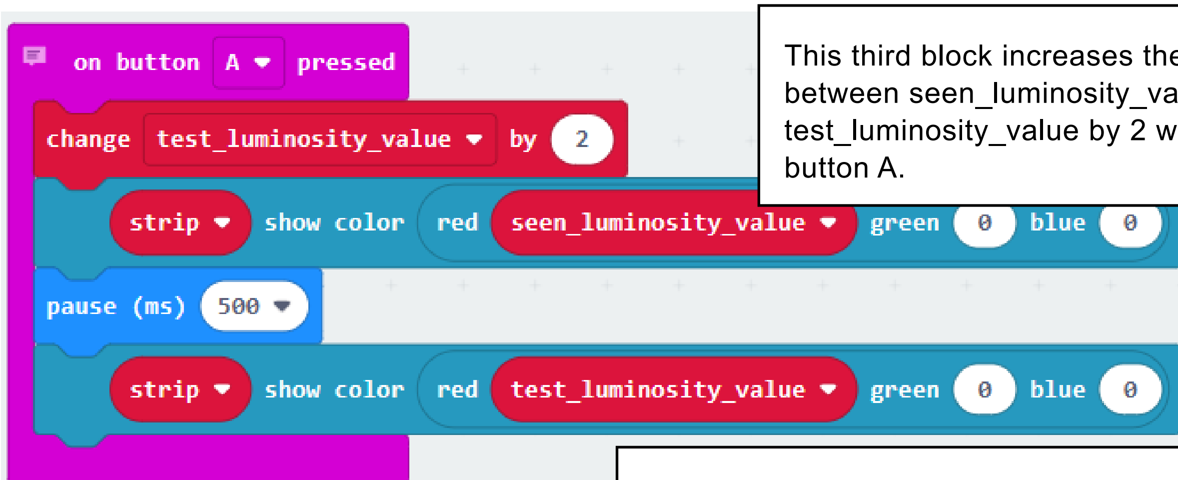
On opening the program, you should see four blocks (we are using Red for examples).



This first block lights a rainbow and sets the initial values for the two luminosity variables to zero.



What do you think the second block does? Hint: compare it to the "on start" block. Write your predictions in your Engineering Notebook.

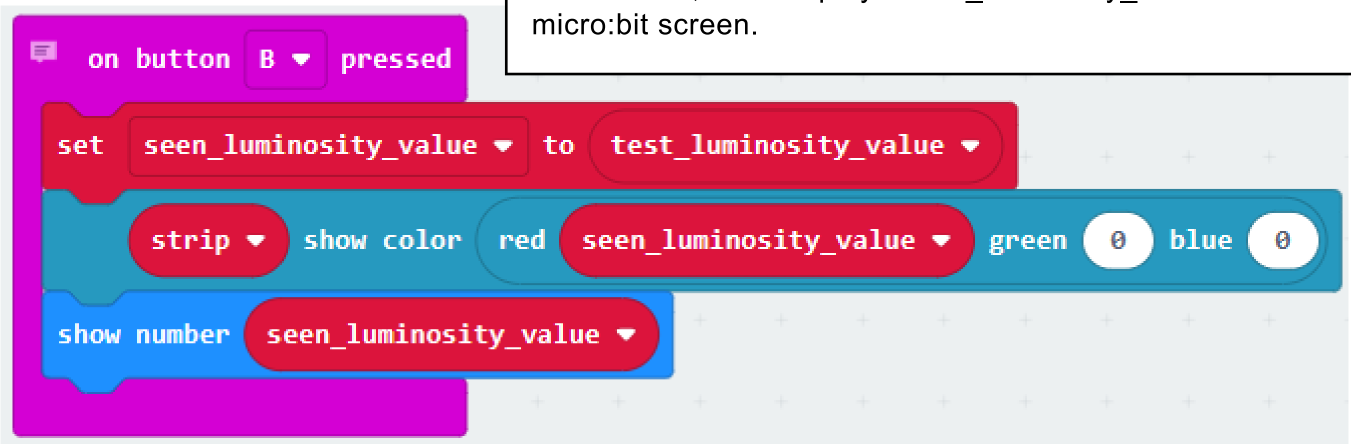


This third block increases the distance between seen_luminosity_value and test_luminosity_value by 2 with each press of button A.

It will also show LEDs at seen_luminosity_value, pause for half a second, and then at test_luminosity_value

5.B Experimenting with LED Strip Luminosity

When button B is pressed, the fourth block sets seen_luminosity_value to equal test value, shows the LEDs at the value, then displays seen_luminosity_value on the micro:bit screen.



Part 3: Conduct the Experiment

Your goal is to compare the actual luminosity changes to your perception of the luminosity changes. The steps are provided below. Run through them once, record your data, then feel free to make changes to program or process and try again. Write everything you observe and any changes you make (and what you see) in your engineering notebook.

<u>Get Started</u>

1. Download the program for your color to the micro:bit (press the purple button).
2. Assign roles to each team member.
    a. Tester: press the buttons and read the numbers.
    b. Recorder: record the changes in the Google Sheet.
    c. Observer: observe the LEDs and stop the team when you observe a change.
3. Tester:
    a. Press A on the micro:bit. The first time you do this, you should see the LED change in luminosity from off to on. You won't record anything for this change.
    b. Press B on the micro:bit. The micro:bit's LEDs should show the number "2."
    c. Press A again. The LEDs will get brighter.
    d. Press B again. The micro:bit should show the number "4". Tell the recorder "4."
4. Observer:
    a. Observe how much the LEDs change between the first 'B' and the second 'A' press.
        i. You should clearly notice the LED's shift from dimmer to brighter.
        ii. Consider this the maximum StepSize of "5". Tell the recorder "5."
        iii. If at any time you need to start over, have the tester press A+B together to reset the program.
        iv. If you're still having a hard time seeing any change in brightness, try folding the construction paper over to make it darker around the LED.
5. Recorder:
    a. Record "4" in the column labeled "x = luminosity" right under the 0 (in row 3).
    b. Record "5" in the column labeled "StepSize" under "OFF" (in row 3)

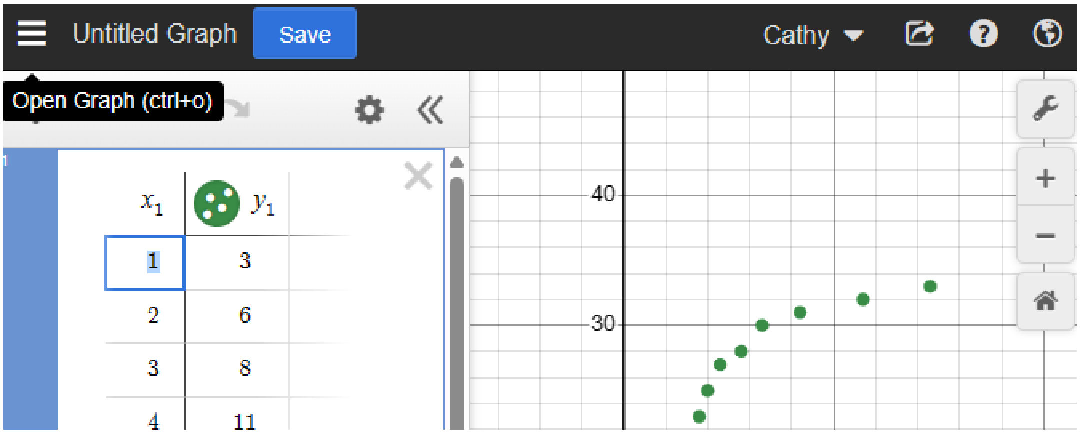
5.B Experimenting with LED Strip Luminosity

Continue to Test

6. Tester:
    a. Press A. Press A again about once per 2 seconds until the observer tells you to stop.
    b. When the observer says "stop" press B.
    c. Read the luminosity value from the micro:bit's screen for the Recorder. You can press B again if needed.
        d. Start again from A when the recorder and observer are ready.
7. Observer:
    a. Observe how much the LED's change.
    b. If you see a change in luminosity, tell the tester to stop.
    c. Tell the observer the size of the change on a scale of 1 to 5, with 5 being the size of change you saw in the first round of testing.
8. Recorder:
    a. Record the observer's number (the step size) in the next row of the StepSize column.
    b. Record the tester's corresponding number (the luminosity value) in the next row of the x = luminosity column.
9. Everyone:
    a. Stop when the tester's number is bigger than 245, or the LED's go off again.

Graph the Data in Desmos

11. Open Desmos (desmos.com).
12. In the LED Luminosity Change Data sheet, copy the two columns labeled X and Y.
13. Paste into a row in Desmos, to the left of the graph.



14. What does the graph look like?
15. Write your observations in your engineering notebook.

Consider: How would you smooth the curve?

16. As a team, discuss ideas for how you could make the perception of luminosity change more even for your LED strip.
17. Write your ideas in your engineering notebook.

LESSON SIX:

Lesson Objectives
Students will be able to:
- Discuss the problem that logarithms were invented to solve.
- Describe how mathematical processes can be designed to solve particular problems.
- Describe Fechner's Law.
- Explain how Fechner's Law is relevant to their findings about human perception of luminosity in the LED strips.
- Identify the logarithmic function as the inverse of the exponential function

Time Required
Two or three 45-minute class periods

Standards Addressed
IAS Mathematics
- AII.EL.1, AII.EL.4, AII.EL.5, AII.FF.2, AII.FF.3, PS.1, PS.2

IAS Science and Engineering Practices
- HS.SEP.5

Key Terms
- Logarithm
- Geometric sequence
- Arithmetic sequence
- Euler's number (e)
- Common log
- Natural log
- Domain
- Range
- Asymptote

Lesson Summary
Students learn about Fechner's Law, which built on Weber's Law (discussed in Lesson 5): "subjective sensation is proportional to the logarithm of the stimulus intensity" S=k*log(R) where S=subjective sensation and R=intensity." They learn that perceived brightness increases according to a natural logarithmic scale. Students then learn about the history of logarithms. They come to understand that logarithms are a form of math that was invented to solve a problem, the multiplication by hand of very large numbers, and the mathematical basis for why that works. Then they learn about more modern ways that logarithms are used. They learn that logarithms are the inverse of exponents, and they practice evaluating logarithms, finding inverses to exponential and logarithmic functions, and graphing logarithmic functions.

Lesson Background
The purpose of this lesson is to review exponential functions, learn about logarithms, and get comfortable using Desmos to graph these functions and find the intercepts, domain and range, asymptotes, and end behavior. This lesson assumes that students have some skill with solving exponential functions in one variable. They will be asked to work in teams to evaluate and graph logarithms, find inverses of a variety of functions, develop logarithmic equations to fit given scenarios, and come up with equations to make transformations from logarithmic to linear patterns of data. After a brief review of exponential functions, students will need to use a device like a Chromebook or iPad with access to Desmos to complete activities. Instructors can use the Desmos activities linked in this curriculum to monitor student work and check for understanding mid-activity.

Teacher Background
Teamwork
Students work in teams on mathematical exercises.

Vocabulary
logarithm, geometric progression, arithmetic progression, exponential function, domain, range, inverse function, asymptote, end behavior

Before the Activity

Prepare to project the slide show, L6 - Logarithms - an Introduction for the class.

Make copies of the duplication masters:

- 6.A Evaluating Logarithms (1 per student)
- 6.B Graphing Logarithms (1 per student)
- 6.C Practice Working with Logarithms (1 per student - homework)

Classroom Instruction

Introduction

1. **Explain to students that they will learn some math that will help them with their designs.** To learn the math we need to understand Fechner's Law, we need to understand Logarithms.

Activity

2. **Open the slide Deck *L6 Logarithms: An Introduction*** for presentation to students.
3. **Begin the lesson with a return to Weber's Law and introduction to the mathematics of Fechner's Law.**
4. **Ask students if they know or can guess what a logarithm is,** or if they recognize the notation.
5. **Introduce the origin story of logarithms** using the PowerPoint *L6 Logarithms – an Introduction*.
6. **Starting with Slide 12, remind students of the properties of exponential functions.** Emphasize that y gets very large, very quickly with an exponential function.
7. **Discuss how logarithmic functions are related to exponential functions.**
8. **Practice evaluating some logarithmic functions.** Have students complete 6.A Evaluating Logarithms with their teammates.
9. **Examine some properties of graphs of logarithms.** Continue working through the slide deck. Prompt students to think about the questions or discuss them with a partner before showing the answers on the slides.
10. **Practice working on graphs of logarithms.** Have students work in teams on 6.B Graphs of Logarithms. Have them use Desmos in their explorations. Make sure they know how to get help in Desmos using the "?" button on the upper right of the screen. Keyboard shortcuts are particularly useful!

Lesson Materials

Per Classroom

- Slide Deck L6 Logarithms: An Introduction
- Projection technology

Per Student

- EDP slider and jumbo paperclip
- Laptop/Chromebook/Tablet
- Engineering notebook

Duplication Masters

- 6.A Evaluating Logarithms
- 6.B Graphing Logarithms
- 6C Practice Working with Logarithms (homework)

LESSON SIX:

Assessment

Pre-Activity Assessment

Revisit 1.A Content Pre-Assessment.

Activity Embedded Assessment

Circulate to check on team conversations and group work on 6.A Evaluating Logarithms and 6.B Graphing Logarithms.

Post-Activity Assessment

Examine student work on the homework assignment, 6.C Practice Working with Logarithms.

11. **Circle back to discuss Fechner's Law.** In Desmos, show the constant k stretches the log graph to larger values of y without changing its shape.
12. **Pass out engineering notebooks** if you haven't already.
13. **Have teams work on the question of how they could apply Fechner's Law** to improve the smoothness of luminosity changes in their prototypes. Students can spend 5-10 minutes at this point – they will have more time after a class discussion in the next lesson.

Closure

14. **Remind students that their knowledge of logarithms will help them** to do further work on their prototypes. In the next lesson, they will revisit their data before working to fit equations to it.
15. **Assign Homework 6.C Practice Working with Logarithms.**

Name____ Date_______________ Period __________

6.A Evaluating Logarithms

1. Write the following equations as logarithmic functions instead of exponential functions.

   a. $6^2 = 36$
   b. $8^3 = 512$
   c. $2^{11} = 2048$
   d. $10^{-4} = 0.0001$
   e. $4^0 = 1$
   f. $12^1 = 12$
   g. $\sqrt{16} = 4$
   h. $5^{-2} = 1/25$

2. Find the value of each of the following logarithms.

   a. $\log_2 8$
   b. $\log_{10} 10{,}000$
   c. $\log_5 125$
   d. $\log_{64} 8$
   e. $\log_6 1/36$
   f. $log_2 64$
   g. $\log_a \sqrt{a}$
   h. $\log_k k$

3. One person creates a meme and shares it. Every hour, the number of people who have seen her meme increases ten-fold (times 10).

   a. How many hours will it be until more than 1 million people have seen the meme?

   b. Write an equation to express the exponential growth of the number of meme-viewing people.

   c. Write an equation to express the number of hours it takes to reach a given number of meme-viewing people.

Name____ Date_______________ Period ___________

6.B Graphs of Logarithms

1. Complete the table below for $f(x) = y = 3^x$

x	-3	-2	-1	0	1	2	3
$y = 3^x$							

   a. For every x, will there be a value of y?

   b. Will your answer to (a) also be true about the inverse of this function? How do you know?

   c. Complete the table below for the function's inverse, $f^{-1}(x) = y = \log_3 x$

x							
$y = \log_3 x$							

   d. In Desmos, graph the set of points from the first table in one line, and then graph $f(x) = 3^x$ in a second line. Do they match? If not, figure out why not, and fix the problem.

   e. In Desmos, graph the set of points from the second table in one line, and then graph $f^{-1}(x) = \log_3 x$ in a second line. Do they match? If not, figure out why not, and fix it.

2. If the function $y = 10 + \log_2 (x + 8)$ was graphed in the coordinate plane, what would its y-intercept be? Show your work here. Check in Desmos.

3. What is the vertical asymptote of $y = \log_2 (x + 4) + 10$? Explain how to figure it out mathematically.

Name____ Date_______________ Period __________

6.B Graphs of Logarithms

4. Find the inverse of each of the following equations.

   a. $y = log_3 x$

   b. $y = \sqrt[3]{x}$

   c. $y = 2^x$

   d. $y = x^5$

   e. $y = log_{10} (x+5)$

   f. $y = 15x^2$

5. Using Desmos, graph the set of points (1,0), (2,1), (4,2), (8,3), (16,4), (32,5), (64,6)

   a. What is the equation that describes this data? Think about it a bit and try making a prediction before going to Desmos to work on it.

   b. Come up with at least one way you could mathematically transform the series of values of x so that the data would be linear. Express the function that would transform x as an equation.

6. How could you use logarithms to make a quick estimate (without a calculator) of the product of multiplying 1,354,391 by 10,342,768? Show your work!

Name______________________________________ Date______________ Period __________

6.C Practice Working with Logarithms

1. Write the following equations as logarithmic functions instead of exponential functions.

   a. $4^3 = 64$

   b. $9^4 = 6561$

   c. $5^1 = 5$

   d. $2^{-4} = 1/16$

   e. $\sqrt[3]{125} = 5$

   f. $8^{1/3} = 2$

2. Find the value of each of the following logarithms.

   a. $log_4 1024$

   b. $log_7 49$

   c. $log_3 1/243$

   d. $log_{10} 10^8 - log_{10} 10^5$

   e. $log_a \sqrt[3]{a}$

   f. $log_2 1024 - log_2 512$

3. Find the inverse of each of the following equations.

   a. $y = log_5 x$

   b. $y = 12^x$

   c. $y = 5+2^x$

   d. $y = log_{10} 5x$

4. Using Desmos, graph the equation y = *a* + *log(bx + c)* and add sliders. For each of the constants *a, b,* and *c:*

   a. What happens if you set the constant equal to zero?

   b. What happens if you increase its value?

   c. What happens if you make it a negative number?

   d. Find values for *a, b,* and *c* such that *y = a + log(bx + c)* has a y-intercept at -2.

5. What is the vertical asymptote of $y = 12 + log_{10}(x + 2)$? Explain how to figure it out mathematically.

6.D Evaluating Logarithms - Answer Key

1. Write the following equations as logarithmic functions instead of exponential functions.

   a. $6^2 = 36$ **$log_6 36 = 2$**
   b. $8^3 = 512$ **$log_8 512 = 3$**
   c. $2^{11} = 2048$ **$log_2 2048 = 11$**
   d. $10^{-4} = 0.0001$ **$log_{10} 0.0001 = -4$**
   e. $4^0 = 1$ **$log_4 1 = 0$**
   f. $12^1 = 12$ **$log_{12} 12 = 1$**
   g. $\sqrt{16} = 4$ **$log_{16} 4 = 1/2$**
   h. $5^{-2} = 1/25$ **$log_5 1/25 = -2$**

2. Find the value of each of the following logarithms.

   a. $\log_2 8$ **= 256**
   b. $\log_{10} 10{,}000$ **= 4**
   c. $\log_5 125$ **= 3**
   d. $\log_{64} 8$ **= 1/2**
   e. $\log_6 1/36$ **= -2**
   f. $log_2 64$ **= 6**
   g. $\log_a \sqrt{a}$ **= 1/2**
   h. $\log_k k$ **= 1**

3. One person creates a meme and shares it. Every hour, the number of people who have seen her meme increases ten-fold (times 10).

   a. How many hours will it be until more than 1 million people have seen the meme?

      It will take about 6 hours for the number of people to exceed 1 million.

   b. Write an equation to express the exponential growth of the number of meme-viewing people.

      If P = Number of People; and h = number of hours, then $P = 10^h$

   c. Write an equation to express the number of hours it takes to reach a given number of meme-viewing people.

      $h = \log_{10}(P)$

6.E Graphs of Logarithms - Answer Key

1. Complete the table below for $f(x) = y = 3^x$

x	-3	-2	-1	0	1	2	3
$y = 3^x$	**1/27**	**1/9**	**1/3**	**1**	**3**	**9**	**27**

a. For every x, will there be a value of y?

Yes, because the domain of this function is all real numbers.

b. Will your answer to (a) also be true about the inverse of this function? How do you know?

No, because the range of this function is y>0, so the domain of the inverse will be x > 0.

c. Complete the table below for the function's inverse, $f^{-1}(x) = y = \log_3 x$

x	**1/27**	**1/9**	**1/3**	**1**	**3**	**9**	**27**
$y = \log_3 x$	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**

d. In Desmos, graph the set of points from the first table in one line, and then graph $f(x) = 3^x$ in a second line. Do they match? If not, figure out why not, and fix the problem.

e. In Desmos, graph the set of points from the second table in one line, and then graph $f^{-1}(x) = \log_3 x$ in a second line. Do they match? If not, figure out why not, and fix it.

2. If the function $y = 10 + \log_2 (x + 8)$ was graphed in the coordinate plane, what would its y-intercept be? Show your work here. Check in Desmos.

Set x = 0, so $y = 10 + \log_2 (0 + 8)$
$y = 10 + \log_2 (8)$
$y = 10 + 3 = 13$

3. What is the vertical asymptote of $y = \log_2 (x + 4) + 10$? Explain how to figure it out mathematically.

The vertical asymptote is -8. It has to be -8 because normally it would be at 0 if the argument of the log were just x, but the argument for the logarithm is x + 8, so the logarithmic curve is shifted to the left by 8. Mathematically, to find the asymptote of the logarithm you set the argument to be zero (x + 8 = 0) and solve for x (x = -8).

6.E Graphs of Logarithms - Answer Key

4. Find the inverse of each of the following equations.

   a. *$y = \log_3 x$* **$y = 3^x$**
   b. $y = \sqrt[3]{x}$ **$y = \log_x 1/3$**
   c. *$y = 2^x$* **$y = \log_2 x$**
   d. *$y = x^5$* **$y = \log_x 5$**
   e. *$y = \log_{10}(x+5)$* **$y = 10^{x+5}$**
   f. *$y = 15x^2$* **$y = \sqrt{x/15}$ when $x>0$**

   For the last 2 in particular, it helps to remember that you can find the inverse by swapping x and y in the equation and then solving for y.

5. Using Desmos, graph the set of points (1,0), (2,1), (4,2), (8,3), (16,4), (32,5), (64,6)

   a. What is the equation that describes this data? Think about it a bit and try making a prediction before going to Desmos to work on it.

   $y=\log_2 x$

   b. Come up with at least one way you could mathematically transform the series of values of x so that the data would be linear. Express the function that would transform x as an equation.

   $x'=2^x$ will transform x such that $y=\log_2(2^x)$ is linear.

6. How could you use logarithms to make a quick estimate (without a calculator) of the product of multiplying 1,354,391 by 10,342,768? Show your work!

   If you estimate the first number and the second number as powers of 10, so: $10^6 \times 10^7 = 10^{13}$ which gives an estimate of 10,000,000,000,000

6.F Practice Working with Logarithms - Answer Key

1. Write the following equations as logarithmic functions instead of exponential functions.

a. $4^3 = 64$ **$\log_4 64 = 3$**

b. $9^4 = 6561$ **$\log_9 6561 = 4$**

c. $5^1 = 5$ **$\log_5 5 = 1$**

d. $2^{-4} = 1/16$ **$\log_2 1/16 = -4$**

e. $\sqrt[3]{125} = 5$ **$\log_{125} 5 = 1/3$**

f. $8^{1/3} = 2$ **$\log_8 2 = 1/3$**

2. Find the value of each of the following logarithms.

a. $\log_4 1024$ **= 5**

b. $\log_7 49$ **= 2**

c. $\log_3 1/243$ **= -5**

d. $\log_{10} 10^8 - \log_{10} 10^5$ **= 3**

e. $\log_a \sqrt[3]{a}$ **= 1/3**

f. $\log_2 1024 - \log_2 512$ **= 19**

3. Find the inverse of each of the following equations.

a. $y = \log_5 x$ **$y = 5^x$**

b. $y = 12^x$ **$y = \log_{12} x$**

c. $y = 5+2^x$ **$y = \log_2(x - 5)$**

d. $y = \log_{10} 5x$ **$y = 1/5\ 10^x$**

4. Using Desmos, graph the equation y = *a* + *log(bx + c)* and add sliders. For each of the constants *a, b,* and *c:*

a. What happens if you set the constant equal to zero?
If a=0, nothing changes. If b=0, the function is undefined. If c=0, the vertical asymptote is 0.

b. What happens if you increase its value?
If you increase a, the curve shifts up without changing its shape. If you increase b, it does the same thing as increasing a, but not as fast. If you increase c, it shifts the vertical asymptote to a negative value of x.

c. What happens if you make it a negative number?
If a is negative, the curve shifts down without changing its shape. If you make b negative, the curve reverses around the y-axis. If you make c negative, it shifts the vertical asymptote to be greater than x=0.

d. Find values for *a, b,* and *c* such that *y* = *a* + *log(bx + c)* has a y-intercept at -2.
a=-3, b=1, c=2… answers may vary.

5. What is the vertical asymptote of $y = 12 + \log_{10}(x + 2)$? Explain how to figure it out mathematically.
The vertical asymptote is -2. To find the asymptote of the logarithm you set the argument to be zero (x + 2 = 0) and solve for x (x = -2).

LESSON SEVEN:

Lesson Objectives
Students will be able to:
- Justify design decisions with math and science evidence.
- Apply understanding of the logarithmic power rule to create a power function for the micro:bit
- Apply understanding of the logarithmic change of base rule to create a logarithmic function for the micro:bit
- Apply understanding of logarithmic properties and inverse functions to control luminosity changes smoothly on the micro:bit.
- Revise prototypes and test that they function by collecting and analyzing data.

Time Required
Two 45-minute class periods

Standards Addressed
IAS Mathematics
- AII.FF.1, AII.FF.3, AII. FF.4, AII.MFD.1PS.1, PS.2, PS.3, PS.4, PS.5

IAS Science and Engineering Practices
- HS.SEP.4, HS.SEP.5, HS.SEP.6

Lesson Summary

Students use Fechner's Law to describe their luminosity data by figuring out the mathematical formula for each color. Students learn about logarithmic rules and properties, and use them to revise their micro:bit programs, incorporating logarithmic brightening and dimming cycles.

Lesson Background

There are at least three broad approaches to solving this problem. One approach is to address what values of luminosity to turn on. Students can figure out what values of luminosity are spaced in such a way as to look linear to the human eye by choosing values of y that are evenly spaced and calculating what the corresponding value of x would be. With a bit more thought, they can come up with an equation that will help them achieve this, realizing that if they replace x with 10^x in Fechner's Law then they will have both x's and y's in an arithmetic progression. It can help to use the table function in Desmos for the logarithmic curve, and put a series of evenly spaced values of x expressed as 10^x (e.g. 10^1, $10^{1.1}$, $10^{1.2}$, $10^{1.3}$) and see that y is indeed an arithmetic progression in this case. If they are using the red-green-blue **show color** blocks in MakeCode, they will have a luminosity range of 0-254 to work with, which would equate to values of 10^0 to $10^{2.4}$. Students will need to figure out how many luminosity steps they want to put into their programs, and use small enough changes in the exponents to get that many steps (e.g. making a change of .05 to the exponent in each step).

A second approach is to address the shape of the curve. A serious challenge is overcoming the flashing that appears at low levels of light, as it is not possible (given the equipment they have at hand) to make luminosity changes that are small enough that humans can't discern them at low luminosity levels. If students examine the logarithmic curves they fitted, they can determine a lower-level "cutoff" point and use only higher-value luminosity steps in their programs. This approach can be combined with other approaches.

A third approach is to address the timing. During their experiment, the timing was the same for each luminosity change; students can use logarithms to lengthen the timing of pauses in the program at low light while shortening it at the brightest luminosity values to compensate for the "cliff" of perception of light changes at low light levels. They can do this by multiplying a base wait time (e.g. 10ms) by the log of the ratio between the current and prior luminosity levels and dividing by the log of the ratio between the brightest and second-brightest luminosity levels (a constant: log(254/252) = 0.003433).

Once students have figured out their mathematical approach, they need to code it in MakeCode, which does not have a function for logarithms – and also has a buggy power function. MakeCode does, however, have the ability to calculate e^x and ln(x) in JavaScript or Python. Therefore, students will learn to make their own MakeCode function blocks for the exponential and logarithmic functions, using logarithmic rules: the power rule to create an exponential function, and the change of base rule to create a logarithmic function.

It will be important for you to be available to support students as they work to apply mathematics to get their programs and hardware to do what they want. Students can make significant progress if they make use of Desmos and spreadsheets to work out their ideas mathematically before trying to implement them in MakeCode. Be aware that they will need to be patient, persistent, and focused to be successful. It can help to stop the class at strategic times so teams can share their progress and frustrations, and help each other.

Teacher Background

Teamwork

Students will need to collaborate and include each other's contributions in order to have a project that represents the whole team's investment. Watch out for students who are not contributing, or who are dominating the discussion and decision-making.

Engineering Design Process

In this lesson, students work on a new iteration of their prototype. They incorporate what they have learned and are learning about logarithms to rewrite their code to smooth out changes in luminosity.

Engineering Notebook

Encourage students to take notes in their engineering notebooks. They should especially write down their ideas about how to incorporate mathematics into their code, and their observations about how well their attempts succeed.

Lesson Materials

Per Classroom

- EDP Poster
- 4-5 Power Strips, distributed so each group has access to plug into one.

Per Group (3 per group)

- 1 micro:bit with Keyes breakout board
- 10 LED strips with Dupont connector
- USB-A cable
- 9V power adapter

Per Student

- EDP slider and jumbo paperclip
- Computer
- Engineering notebook

Duplication Masters

- 7.A Coding Mathematical Functions in MakeCode (1 per student)

Assessment
Activity Embedded Assessment
Observe students working on teams, and their contributions to the class discussions and teamwork on fitting functions, logarithm rules, and writing functions using the mathematical operations available for the micro:bit.

Post-Activity Assessment
Examine teams' code and students' engineering notebooks to see what they accomplished. Check their math!

Before the Activity

Collect all of the student graphs into your Desmos account. Use it to combine the data for each color from the teams who tested that color into one graph (for example, the white graph should have the data from 2 or 3 teams in it).

Be prepared to project Desmos so the class can view and discuss each of the combined graphs.

Prepare to project the slide show, *L7 - Logarithm Rules* for the class.

Print copies of 7.A Coding Math Functions in MakeCode

Classroom Instruction

Introduction

1. **Describe today's challenge:** to identify a mathematical function that they can use to even out the luminosity changes for each color in their micro:bit.
2. **Identify where they are in the engineering design process (Plan). Ask:** *What phase of the design process do you think we will be in today?* Students will Plan their designs and Try them in their prototypes. This is at least their second iteration.
3. **Explain that students will first work to identify what logarithmic function best describes their data** from the experiments teams conducted in Lesson 5. Then, they will work to figure out how to adjust their programs to correct for the logarithmic nature of human perception.

Activity

4. **Pass out engineering notebooks.**
5. **Lead a class discussion of findings for white from team experiments in Lesson 5.** Project Desmos on the board, and display the combined data for the color white.
6. **Have students reflect on reliability across team data. Ask:** *How well do the findings from each team match?*
7. **Open the data for each of the other colors (red, green, blue) in turn.** Ask students to point out how the data collected from different teams is similar / different. **Ask:** *What might account for the differences?*
8. **Lead a class discussion to find the best-fitting logarithmic curve for white.** First try fitting a curve in Desmos by inputting

the equation "y = k log (x)" which is Fechner's Law. Add the slider for k and adjust it to find a good match to the data. Then compare to the parameters Desmos provides when you enter the regression equation "y ~ k log (x)." Discuss the size of R^2 (how close is it to 1?) and the look of the fit to the data points.

9. **Share the combined data files with students, and have them fit logarithmic functions in teams.** Each team should come up with a logarithmic function for each of the three colors (red, green, blue). This will give them valuable practice with fitting curves to data.
10. **Lead a class discussion of the functions fitted by the teams. Ask:** *How well do the logarithmic functions each team wrote match the plotted data points? How well do they match each other?*
11. **Guide the class to come to consensus** on what logarithmic formula best matches the combined data for each color.
12. **Lead a class discussion, soliciting general ideas for how to even out the luminosity changes.** What functions or approaches could students use? Emphasize that there is more than one approach possible. If students are having trouble coming up with ideas, suggest those presented in the Lesson Background (change the series of luminosity values that are sent to the LED strips; don't show low-level luminosity values; change the timing between changes in luminosity values).
13. **Once all teams have settled on a mathematical approach, use the powerpoint L7 - Logarithm Rules** to introduce the logarithmic rules and properties.
14. **Have students work in pairs or teams on the questions** in green on each slide.
15. **Work out examples as needed.** One or two examples are given on each slide.
16. **When you reach slide 11, tell students** that they will use some or all of these rules and properties from the presentation in working with the micro:bit to program smoother luminosity changes.

This may be a good place to break between classes, if additional time is needed.
You could assign students to figure out functions for the last 2 slides for homework.

17. **Have students work out the math they need to create functions in Make:Code.** Beginning with slide 11 of L7 - Logarithm Rules, the powerpoint presents the math challenge.
18. **Have teams work on how to use the new functions to smooth the luminosity changes** for their LED strips. Helpful tools include Desmos and a spreadsheet program (e.g. a new worksheet in LED Luminosity Data). Give them at least 10 minutes, but more time will help them to think more deeply about the math they need to do.
19. **[Optional – if students need more help] Conduct a class discussion of team ideas.** Help students to think about the implications of their ideas, and how they might approach implementing them.
20. **[Optional – if students need more help] Use the MakeCode program for white light as an example,** projecting it on the board, so you can have students point out where and how they might add code.
21. **Pass out 7.A Coding Mathematical Functions in MakeCode.**
22. **Pass out team prototypes and other equipment for this project.**
23. **Let students know that they should work to incorporate mathematics** with the help of 7.A Coding Mathematical Functions in MakeCode.
24. **Remind students that they can take on roles in their teams to help the team be more productive. I**f there have been any issues with teamwork, you might hold a short discussion of what teamwork approaches work best for teams to do their best work.
25. **Circulate to support students as they work.**

Closure

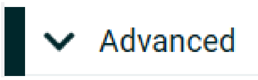
26. **Conduct a class discussion about the day's activities. Ask:** *What went well? What was challenging? How hard was it to program the luminosity with mathematics to even out how it will be perceived?*
27. **Connect to the EDP.** In this lesson, students worked again on the Plan and Try stages.

7.A Coding Mathematical Functions in MakeCode

Welcome to coding exponential and logarithmic functions in MakeCode!

There's just one problem: there aren't any blocks that will do the job. (Note: There is a power block, but it will only handle integers, and the simulation of it has a bug. Bugs happen.)

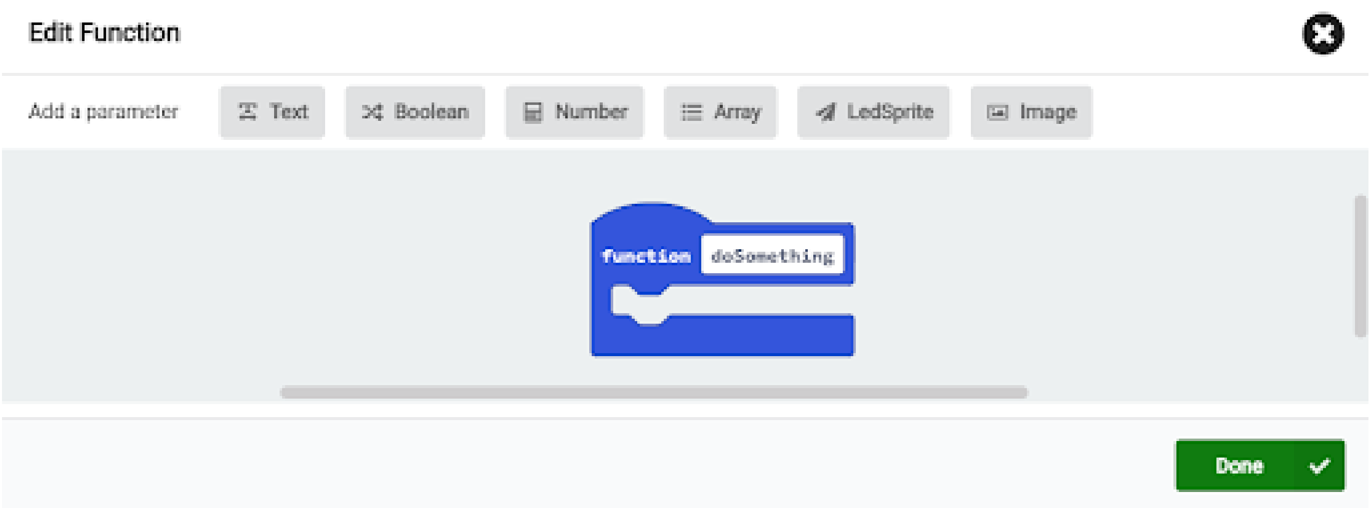
So what do we do? Enter… the function block!

1. Click on this menu: 

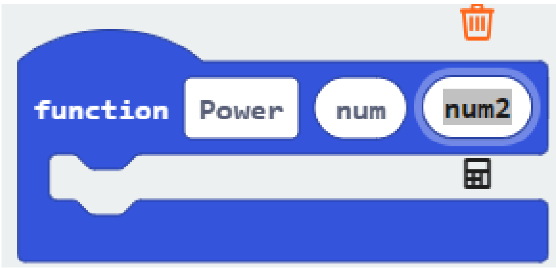
2. With that open, a bunch more menus appear. Choose this one: 

3. You should see a gray column appear. Click this button: 

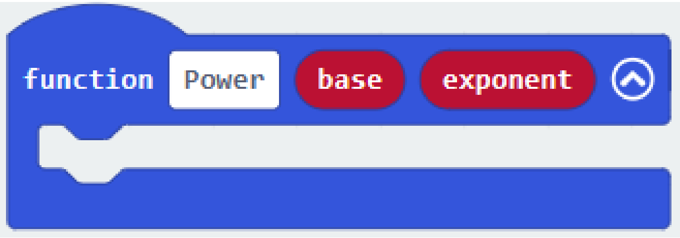
4. and a new pop-up window should appear that looks like this:



5. In place of "doSomething" type "Power"
6. Now click the **Number** button (from the top row of buttons) twice, so your function looks like this:
7. In place of "num" type "base"
8. In place of "num2" type "exponent"
9. Click "Done"

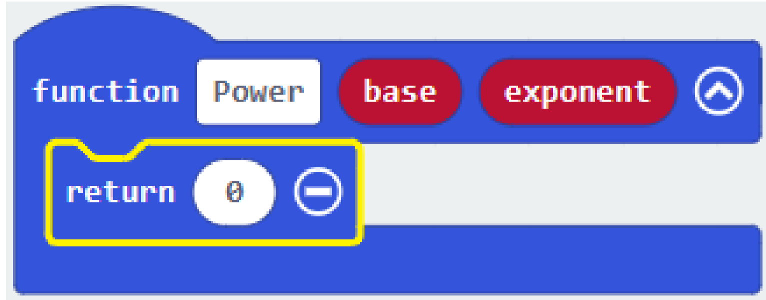


10. The **Power function** appears in your MakeCode blocks area looking like this:
11. Click on the **Functions** menu again, and this time pull out the new block **return**, and drop it into the slot inside your **Power function**.



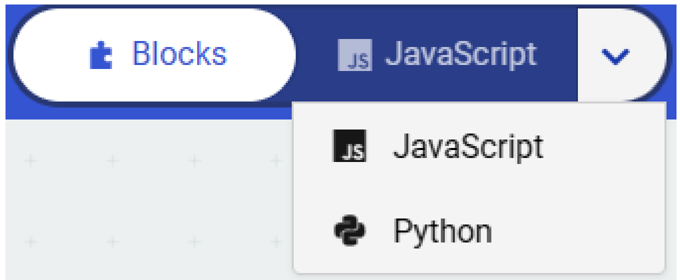
7.A Coding Mathematical Functions in MakeCode

12. Your **Power function** should now look like this:
13. Now your job is to tell the function what to return as output. You should already have worked out with your class how to use the functions Math.exp and Math.log, together with the base and exponent, to get the value of base M raised to the exponent k.



Mathematically, that looks like this: $e^{(M*\ln(k))}$ In Javascript, we will write that as: Math.exp(exponent * Math.log(base))
in the **return** bubble. But there's a trick to it!

14. At the top of the MakeCode screen, press "Javascript":
Javascript is a text-based programming language. In MakeCode, you can program with blocks, with Javascript, or with another text-based language called Python.



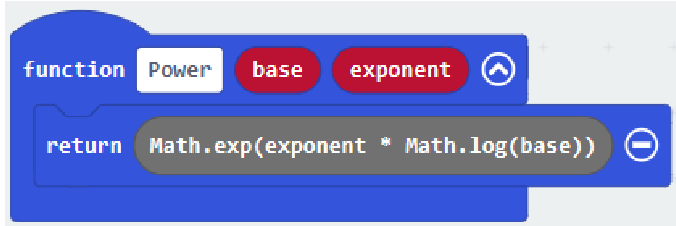
15. Now you should see some text-based code like this:
16. Do you see how the top 3 lines are the same as your **Power function** block?

```
function Power (base: number, exponent: number) {
    return 0
}
basic.forever(function () {

})
```

17. Right where it says return, replace the **0** with the text: **Math.exp(exponent * Math.log(base))**

18. When you press Blocks to go back to the block form, your function should look like this:
19. Now you can use exponential functions in your code!



20. All you need to do is call your new function. You do that in the **Function** menu again: wherever there's a bubble in a block, pop in **call Power** to get the power of the first number raised to the second number.

Now that you've seen how to make the exponent function, try it yourself!

21. Design a new function called **Logarithm**.
22. Try designing other functions! For example, you can make a function called **SetBrightness** that sets the brightness of all of your LED strips at once.
Functions can do more than just math!

LESSON EIGHT:

Lesson Objectives
Students will be able to:
- Develop a testing plan that will address criteria from the client and provide specific and measurable results.
- Provide clear, specific, and actionable feedback and critique to peer teams on their prototypes.
- Analyze data from testing.
- Use data from testing to inform design decisions and further prototype revisions.

Time Required
One or two 45-minute class periods

Standards Addressed
IAS Science and Engineering Practices
- HS.SEP.4, HS.SEP.5, HS.SEP.6

Key Terms
- Testing protocol
- Design decisions
- Iteration

Lesson Summary

In this lesson, students revisit their ideas for testing from Lesson 3. They decide, as a class, how they will address testing of each criteria, and create a testing protocol. Teams then swap prototypes to test each other's work and give feedback. Finally, teams redesign and incorporate new feedback.

Lesson Background

Engineers need to ensure that their prototypes address criteria and meet constraints through specific, measurable metrics. Creating those metrics, and a procedure for testing that can be reliably followed by testers, is an important part of the design process. Engineers need data to decide whether their prototypes are addressing the problem, and to compare different prototypes to decide which is best to continue to work on and finalize for the client.

Teacher Background

Teamwork

Teams will test each other's prototypes and give feedback. If you think your students would benefit, spend a bit of time talking about how to test fairly, and how to give feedback that is specific, clear, and non-judgmental. Teams will also work on redesigning their prototypes to address feedback and testing results, and may need support to negotiate the process of deciding what to change, taking into account everyone's ideas and opinions.

Engineering Design Process

Students **Test** their prototypes, analyze the test results and feedback, and then **Decide** how they will revise, based on the data and evidence, in order to better align with criteria.

Engineering Notebook

Students develop a testing protocol in their engineering notebooks. Next, they record test results and feedback, and then develop ideas to revise their prototypes.

Vocabulary

Testing protocol, design decisions, iteration

Before the Activity

Prepare to record a class consensus testing protocol, either by projecting it and recording on the projection, or by recording on the board / easel paper.

Decide how teams will evaluate each other's work. If you have more time, you might have each team present their design to the class for everyone to assess, and do the testing protocol publicly. If pressed for time, assign pairs of teams that will test each other's prototypes and give feedback.

Print copies of 8.A Design a Testing Protocol (1 per team or student)

Classroom Instruction

Introduction

1. **Revisit the EDP.** This lesson addresses the Test and Decide (also called Evaluate) steps of the Engineering Design Process.
2. **Introduce the activity. Say:** *Today we will be testing solutions generated from other teams. You will be taking notes of your experiences in your engineering notebooks, and using the evaluation criteria to give the team an idea of how their solution does or does not solve the client problem, and ways that their solutions can be improved.*
3. **Prepare for discussion of a common testing protocol.** Tell students to open their engineering notebooks to the point where they recorded 3.A Evidence-Based Reasoning Part 1. Project a blank document (or make space on the board / chart paper) to prepare to write down student suggestions.
4. **Open discussion of a testing protocol. Say:** *Now that it's time to test our prototypes, we need to come to a consensus on how we should test. Why do you think it's best for all of us to agree on one common way to test our prototypes?* Look for students to recognize that, if they all use the same protocol, they can compare their prototypes on a single set of standards.
5. **[Optional] Have teams and/or individual students brainstorm specific ideas for the testing protocol** prior to conducting the class discussion.
6. **Solicit suggestions for what criteria to test and how to test those criteria.** Allow students to offer revisions. Ensure that all criteria are addressed. Students may need assistance to think about how they will assess whether their designs promote enjoyment and a festive atmosphere – they will need to ask for each other's opinions.

Lesson Materials

Per Classroom

- EDP Poster
- 4-5 Power Strips, distributed so each group has access to plug into one.

Per Group (3 per group)

- 1 micro:bit with Keyes breakout board
- 10 LED strips with Dupont connector
- USB-A cable
- 9V power adapter
- Poster-sized foam board (black)

Per Student

- EDP slider and jumbo paperclip
- Laptop/Chromebook/ Tablet
- Engineering notebook

Duplication Masters

- 8.A Design a Testing Protocol

LESSON EIGHT:

Assessment

Pre-Activity Assessment

Pay attention to student contributions of ideas for how to test their prototypes and how to design the testing protocol. Student contributions should show attention to criteria, constraints, and data from users. Contributions should also attend to the need to collect data on performance that is specific and measurable.

Activity Embedded Assessment

Check that all members of the teams are participating productively

Post-Activity Assessment

Examine the notes that students took during testing, and the feedback they provided to other teams. Examine the changes students made to their protocols after testing. To the changes address issues that arose during testing?

Activity

7. **Pass out materials and testing assignments** once the class has agreed on a protocol. Remind teams to be fair and honest when evaluating another team's prototype, and to record all observations in their engineering notebooks.
8. **Observe and support students as they work to complete testing.**
9. **Have teams summarize their findings for each prototype** they tested, and give the summary to the relevant team.
10. **When every team has received feedback, allow teams time to iterate.** Encourage teams to address the feedback and results of testing that they received as they redesign their prototypes.

Closure

11. **Lead a class discussion of the testing process. Ask:** *Do you feel this was a fair test? What feedback was most useful to you? What feedback was less helpful?*
12. **Connect to the EDP. Ask:** *What phase of the engineering design process do you think we will be in for the next lesson? Do you feel like we are able to make a successful recommendation to the client yet? Why or why not?*

Test and Redesign

Name_______________________________________ Date_______________ Period __________

8.A Design a Testing Protocol

Complete a testing protocol for each of the criteria and constraints.

Test of __________________________________ (criterion / constraint):

- Explain the procedure for testing in detail.
- What data will be recorded?
- How/where will the data be recorded?
- What will count as a successful test? Not successful?
- Is there anything you want to ask of the testers? Observations or reactions?

To finish the test, it's generally a good idea to ask your testers to summarize findings, and to provide feedback or suggestions to the design team.

Name____ Date_______________ Period __________

LESSON NINE:

Lesson Objectives
Students will be able to:
- Communicate science, technology, engineering, and mathematics ideas.
- Present their solution to the client's problem
- Use evidence-based reasoning to support their engineering decisions

Time Required
Two 45-minute classes

Standards Addressed
IAS Science and Engineering Practices
- HS.SEP.7, HS.SEP.8

IAS Integrated STEM
- HS.CC1, HS.CC2, HS.CC3

Key Terms
- Criteria
- Constraint
- Communication

Lesson Summary
Each team of students prepares a poster or presentation of their design and prototype. Teams then present their designs, preferably to an invited audience, with an example and either a poster or a presentation.

Lesson Background
Communication is one of the most important skills for engineers or really any industry professional to have. We want students to be able to understand the client's problem and their own proposed solution so well that they are able to explain it to others who were not involved with the project. Students will be working together in their teams to produce a presentation or poster to communicate their solution. After the 'Decide' stage of the Engineering Design Process, engineers often have to communicate their redesigned solution to the client. Students must be able to articulate why their solution meets the client's criteria and explain how it fits within the constraints of the project. This is a great opportunity for students to practice using math vocabulary as they back up their decisions with evidence.

Before the Activity
- If allowing students to present using posters, gather and set up poster materials.
- If using PowerPoint or an equivalent software, it might be helpful to create a template before class that students can use to organize their thoughts. Approximately six slides should be sufficient for teams to meet the communication requirements.
  - Introduction of students
  - Problem summary
  - Approach to solving the problem
  - Summary of design
  - Test results
  - Final design recommendation to the client
- Decide whether to invite others, such as members of student government or school administrators, to attend the Design Showcase. Plan for this event to take place during the second class period for this unit.
- (Optional): If you are interested in learning more about the workforce needs in the microelectronics industry, reading over the CHIPS Act of 2022 is a good place to start. It will give the instructor more context when discussing microelectronics-

related careers in engineering. https://tinyurl.com/CHIPS-Act-2022

- Print and make copies of the following duplication masters in the labeled amounts:
    - (1 per group) 8.A Communication Requirements
    - (1 per student) 8.C Content Post-Assessment

Classroom Instruction

Introduction

1. **Remind students of the engineering challenge. Ask:** *Who is our client? What did they challenge us to design? What were the criteria and constraints?*
2. **Identify where they are in the engineering design process** (decide)
3. **Explain to students:** Throughout the entire engineering design process, we have been working in teams and communicating within your teams and with other teams in the class. Now you need to communicate your proposed solutions to the client, Gina Dancer.

Activity

4. **Review Client Memo.**
5. **Explain what is needed in the presentation (or posters).** Hand out 9.A Communication Requirements. Briefly describe what is required of the groups. If you have developed a presentation template for your students, share this now.
6. **Facilitate a class discussion about the purpose of the presentation/poster. Ask students:** *Why does the client need to hear from you? What do you need to communicate to her?*
7. **Teams prepare design review presentations (or posters).** Students may wish to take pictures of their engineering notebooks or any other artifacts from their work on the unit. Teams should have their final designs mounted on the black foam board, ready to demonstrate. Teams should practice with each other once presentations are drafted.
8. **Circulate and assist teams as needed.** Students may need assistance with teamwork, presentation/poster design, or what is most important to communicate.
9. **[Second class period] Prepare for the Design Showcase.** Ensure student presentations are lined up on the presentation computer, or hang posters on the wall. Ensure all prototypes are ready to show off. Arrange seating for any guests.

Lesson Materials

Per Classroom

- EDP Poster

Per Group (3 per group)

- Presentation software such as Google Slides or PowerPoint OR Poster

Per Student

- EDP slider and paperclip
- Laptop / Chromebook / Tablet
- Engineering Notebook
- Completed prototypes

Duplication Masters

- 9.A Communication Requirements
- 9.B Client Memo 4
- 9.C Content Post-Assessment

LESSON NINE:

Assessment
Pre-Activity Assessment
Class discussion about what is important in a presentation.

Activity Embedded Assessment
Check that all members of the teams are participating productively.

Post-Activity Assessment
Final presentation assessed using 8.A Communication Requirements. Assess student understanding of the unit using the 8.C Content Post- Assessment, reviewed with 8.D Content Post Assessment Key.

10. **Hold the Design Showcase.** Encourage teams to support each other by giving specific appreciations and asking questions. Each student should write down at least one thing they noticed or appreciated from each team's presentation or poster in their engineering notebooks.
11. **Discuss how microelectronics can be used to design and create things for people to enjoy.** *What other types of things could you design with the micro:bits or other microelectronics? What other uses do you see for programmable LED lighting? Do you think you would enjoy taking up microelectronics as a hobby? Would you be interested in a career involving microelectronics?*
12. **Make connections to careers relating to microelectronics.** Help students understand why the microelectronics industry is relevant to their lives and communities. In states like Indiana and Ohio, there are many jobs available in this field, some of which require only a high school degree, others requiring a college degree. Talk about the CHIPS Act of 2022: https://tinyurl.com/CHIPS-Act-2022

Closure

13. **Conduct a class discussion to debrief the unit.** You might use these questions as prompts: *What did you learn? What did you think of this problem? What do you think of how logarithms may be useful in science and engineering? What do you think about engineering as a field?*
14. **Provide feedback from the client.** You may need to wait to do this if you need time to review the team's presentations/posters and write comments. Read 9.B Client Memo 4 to the class. Reflect on the engineering design process and how it was used throughout the unit.
15. **Have students complete 9.C Content Post Assessment.**

Design Showcase - Share with the Client

9.A Communication Requirements

Your presentation or poster must include the following parts:

- ◻ Introduce all members of the team.
- ◻ Summarize the client's problem including criteria and constraints.
- ◻ Explain why the problem is important to the client.
- ◻ Demonstrate and describe the team's solution to the problem.
  - Demonstrate the three (or more) lighting modes and how they are intended to be used, with justifications from initial interviews with end-users.
  - Describe features of the lighting system including the math concepts and code.
  - Explain test results from lesson seven.
  - Describe the team's thoughts on how they would redesign and why, referencing test results.
- ◻ Explain the role of the engineering design process in crafting their designs

Be sure that:

- ◻ Data and evidence gathered and used in the design are presented.
- ◻ You explain and justify your design decisions using data and evidence.
- ◻ All team members have a role in developing the poster or presentation

9.B Client Memo 4

Dear Engineers,

On behalf of the dance committee at Anytown High School, I thank you for your work. The designs you submitted are creative and exciting, and they will enliven our end-of-year dance.

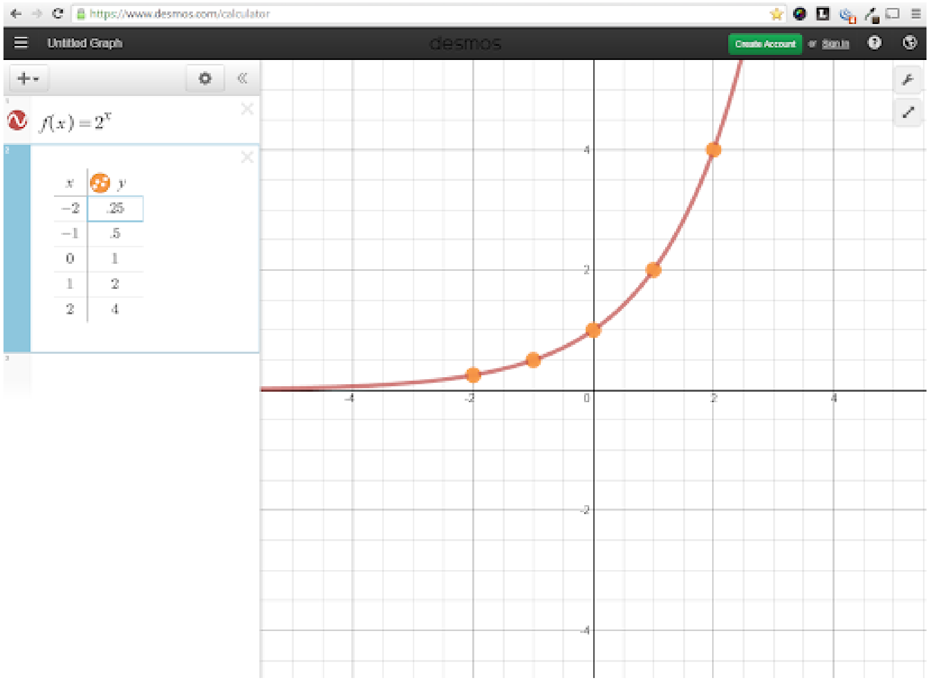
Yours sincerely,

Gina Dancer, Grade 12 Student
Head of the Dance Committee of Anytown High School

Name____ Date_______________ Period __________

9.C Content Post-Assessment

1. What is an inverse function? Explain and demonstrate by drawing the inverse of the function below.



2. Simplify this function, showing your work: $f(x) = (x^{12})(x^6)$

   a. What is the domain of the equation in question 2? What is the range?

3. Simplify or find the value of each of the following:

   a. 5^4
   b. 5^{-4}
   c. $25^{1/2}$
   d. $(x^5)^4$
   e. $a^3 \text{ x } 3a^6$
   f. $a^2 \text{ x } b^2$
   g. $a^5 \div a^3$
   h. $z^{-3} \text{ x } z^4$

Name____ Date_______________ Period ___________

9.C Content Post-Assessment

4. What is a logarithm?

5. What is the value of log(1000)?

6. What is the value of $\log_2(16)$?

7. Write the equation $12^4 = 20{,}736$ as a logarithm.

8. How can you use logarithms and exponents to help you make a quick estimate (without a calculator) of the product of multiplying 10,343,074,143 by 1,322,124,456? Show your work.

9. What is an engineer? What kind of work does an engineer do?

Name______________________________________ Date______________ Period __________

9.C Content Post-Assessment

10. Have you ever done some engineering? Explain what you did and why.

11. What does the term, “microelectronics” mean?

12. How are microelectronics used in the field of algebra or pre-calculus?

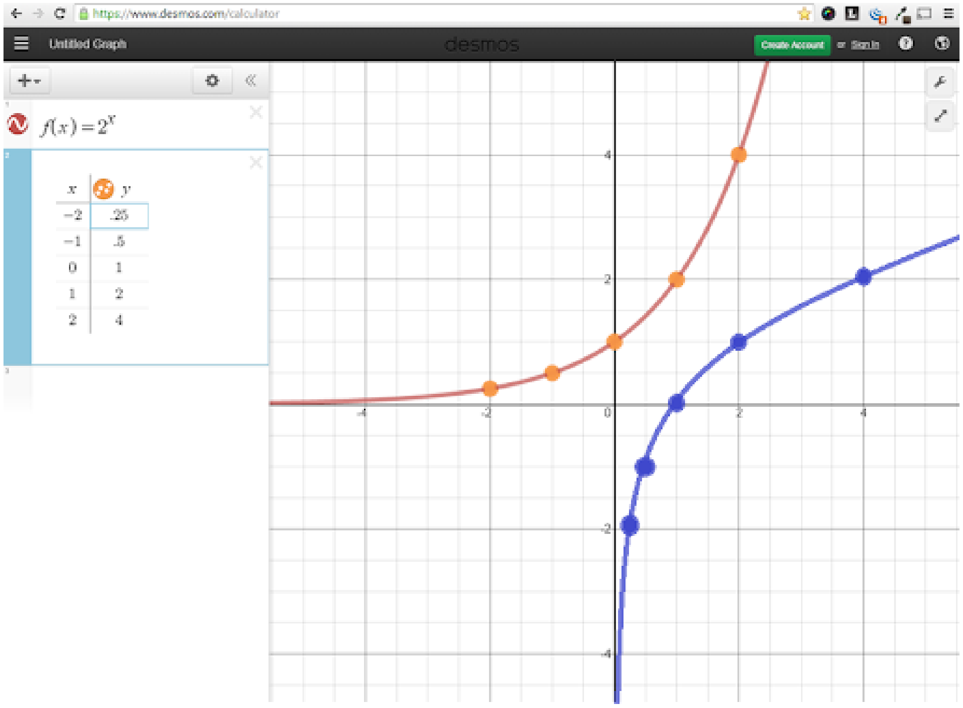
13. What jobs would you be interested in that use microelectronics? Provide one example of how microelectronics is used in that job.

Name____ Date_______________ Period ___________

9.D Content Post-Assessment Key

1. What is an inverse function? Explain and demonstrate by drawing the inverse of the function below.

An inverse function is a function that reverses the operations of another function.



2. Simplify this function, showing your work: $f(x) = (x^{12})(x^6)$

$f(x) = x^{18}$

a. What is the domain of the equation in question 2? What is the range?

The domain is all real numbers. The range is all positive real numbers > 0.

3. Simplify or find the value of each of the following:

a. 5^4	**= 625**	b. 5^{-4}	**= 1/625**
c. $25^{1/2}$	**= 5**	d. $(x^5)^4$	**= x^{20}**
e. $a^3 \times 3a^6$	**= $3a^9$**	f. $a^2 \times b^2$	**= $(ab)^2$**
g. $a^5 \div a^3$	**= a^2**	h. $z^{-3} \times z^4$	**= z**

9.D Content Post-Assessment Key

4. What is a logarithm?

   A logarithm is the inverse function to an exponent. Given a base and a number, the logarithm is the exponent you would need to raise the base to in order to produce the number.

5. What is the value of log(1000)?

   log(1000) = 3

6. What is the value of $\log_2(16)$?

   $\log_2(16) = 4$

7. Write the equation $12^4 = 20{,}736$ as a logarithm.

   $\log_{12}(20{,}736) = 4$

8. How can you use logarithms and exponents to help you make a quick estimate (without a calculator) of the product of multiplying 10,343,074,143 by 1,322,124,456? Show your work.

   Estimate the first number and the second number as powers of 10, so:
   $10^{10}10^{9}=10^{19}$ which gives an estimate of 10,000,000,000,000,000,000

9. What is an engineer? What kind of work does an engineer do?

   An engineer is a person who applies mathematics, science, creativity, and technical skills to design solutions to problems.

9.D Content Post-Assessment Key

10. Have you ever done some engineering? Explain what you did and why.

    Responses will vary.

11. What does the term, “microelectronics” mean?

    Student answers may vary, but the formal definition of microelectronics is the design, manufacture, and use of microchips.

12. How are microelectronics used in the field of algebra or pre-calculus?

    There are many answers to this question, but one example could be using microelectronics to collect a specific type of data for graphical analysis.

13. What jobs would you be interested in that use microelectronics? Provide one example of how microelectronics is used in that job.

    Students’ answers will vary based on interest. Credit may be given as long as at least one job example is provided with their logic behind how that job uses microelectronics.