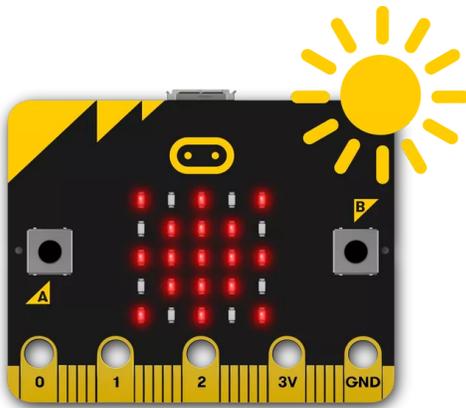


Modeling Solar Powered Tiny Homes with Micro:bits

Middle School STEAM

Summary

Modeling tiny homes has become an increasingly popular 5th–8th grade STEAM activity, due to its incorporation of engineering constraints, applied math skills, and artistic expression.¹ The project outlined below will vary slightly in typical tiny home lesson plans, as student teams will integrate a Micro:bit microcontroller (5cm x 4 cm) into the design to be used as a proxy for solar panel power production. The LED lights on Micro:bits will provide quantitative data on light input, which can be used to estimate potential solar power production.² Student teams will learn how to optimize solar energy through their home’s solar panel position and angle using the coding platform available on [Microsoft MakeCode](https://makecode.com/).



Micro:bit from the “Sunlight Sensor Project”³

Tiny House Design from Northwestern’s Tiny House Project⁴

¹ Please watch PBLwork’s video on tiny house PBL: <https://www.pblworks.org/video-tiny-house-project>

² How LEDs can measure light input:

<https://support.microbit.org/support/solutions/articles/19000024023-how-does-the-light-sensing-feature-on-the-micro-bit-work->

³<https://microbit.org/projects/make-it-code-it/sunlight-sensor/>

⁴ From Northwestern University: <https://design.northwestern.edu/projects/profiles/tiny-house.html>

Engineering Connection

As efforts to reduce carbon emissions continue to dominate both local and geopolitical dialogue, it is imperative that K12 students also participate in the revisioning of our electric grid. Traditional solar panels continue to decrease in price, and new panel technology (such as organic polymer and perovskite photovoltaics) promise faster investment returns on domestic installation.

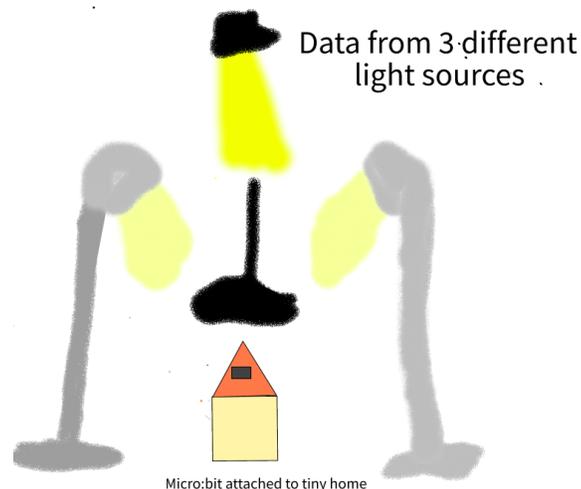
Additionally, concepts of maximizing area and volume, material and weight considerations for 3D design, input stimuli and output data from microcontrollers, and block code programming are all incorporated in this PBL curriculum. These are foundational material, engineering, and computer science concepts appropriate for 7th and 8th graders.

Learning Objectives

The goal of this project will be to model a tiny home that meets all designated design constraints in addition to maximizing solar potential of their roof/panel setup. As part of the tiny home constraints, the student team's floor plan must be less than 500 sq. cm. and include separate spaces for at least one bathroom, kitchen, and bedroom.

This activity will not focus on building furniture inside the home as many tiny lesson plans do, but will rather emphasize roof structure. Student teams will be given several possible sheet materials to make the frame of their tiny home. Student teams will identify the constraints of solar panel installation (weight, rigidity), and how they can best address these issues in their roof design.

The total solar potential of each home will be determined from three trials of one lamp shining on the house from three equidistant locations. Students will be informed of these locations in advance. Students will learn how to code input and output variables on Microsoft MakeCode to gather and analyze this light data from their attached Micro:bits.



Materials List

- graph paper
- cardboard
- balsa wood
- any other sheet building material for scaffolding (aluminum, paper, styrofoam)
- tape (masking, scotch, duct)
- glue
- rulers & protractors
- Micro:bits & their batteries (~\$30)
- computer device with bluetooth or USB ports to connect to Micro:bits
- *if available in a Makerspace*- materials for 3D printing or laser cutters

Worksheets and Attachments

The student report (worksheet) is located at the bottom of this PDF.

Prerequisite Knowledge

While this lesson is made for all student levels, it would benefit students to have some understanding of how our current electric grid is designed, and how solar panels play an important role in decreasing carbon footprints and increasing energy independence. Students should also be familiar with finding and calculating areas using the metric system (cm).

Introduction/Motivation

Ask students if they know where any solar panels in town are located. Show students various examples of tiny homes on Youtube and of unique solar panel technology (solar farm, solar buoys, solar windows). Discuss the pros and cons of both tiny homes and solar panel installation. What are some reasons people might be hesitant or unable to install solar panels on their homes? Inform students that they will be designing their own tiny homes with a solar panel (Micro:bit) on top.

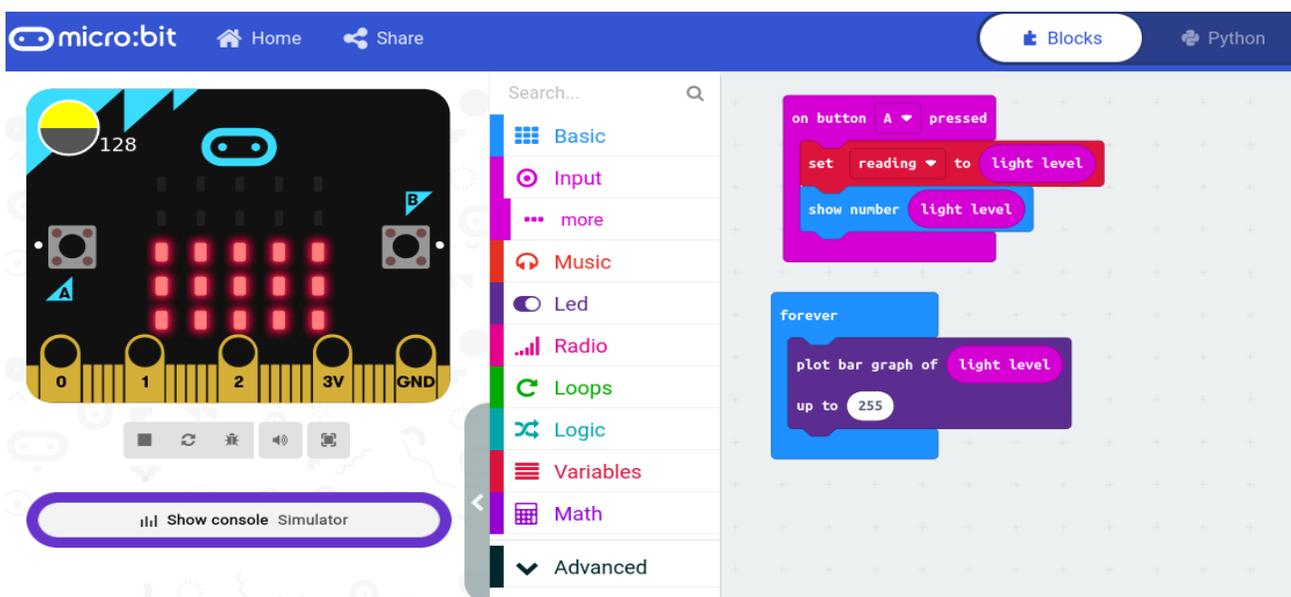
Phase Two: Build the Design (2-3 days) Begin building the tiny house with the provided materials. Make sure students build walls separating the rooms, and measure the angle of the roof where the Micro:bit is attached.

Phase Three: Code the Micro:bit (2-3 days) Walk students through the Microsoft MakeCode website listed for measuring light input. Bluetooth devices can connect directly to the Micro:bit. The “show console” link provides a graph of varying light input.

Teachers can use their discretion in this coding practice. Some teachers may wish to warm up with simpler coding projects (available on the MakeCode website) before jumping into the light input code. Some teachers may want to provide greater background in basic computer science concepts (input/output, bluetooth, loops, etc.).

Another area of possible discussion is the ability of LED and solar panel materials to both conduct electricity **and** provide a source of light (even though these technologies are optimized to do one or the other). That is to say, if current is applied to solar panel material, the panel *can* produce light. Likewise, if light is absorbed by LED material, then the LED can generate current (like our Micro:bit LEDs will).

Once students have running code, they will measure the light input of their homes from three different light locations by pushing the “A” button on their homes.



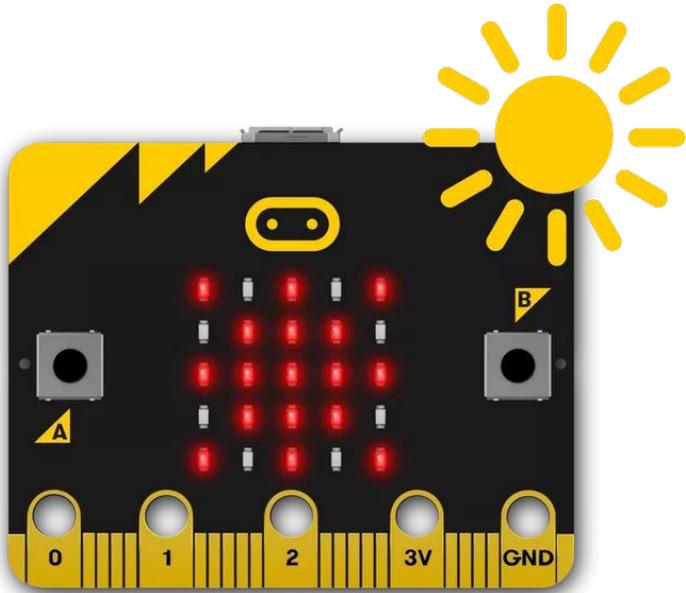
Phase Four: Finish Student Report

Complete the student report, including pictures of your draft, actual home design, explanation of code, and the analysis questions.

Extensions & Scaling

- Spreadsheet data analysis can easily be incorporated into this project. After hitting the “A” button several times during the trial, students would be able to download the .csv file of their light input data.
- Advanced students or groups that finish early can learn other Micro:bit coding to add their home. A myriad of simple projects are available on the MakeCode website.

Solar Tiny House Student Report



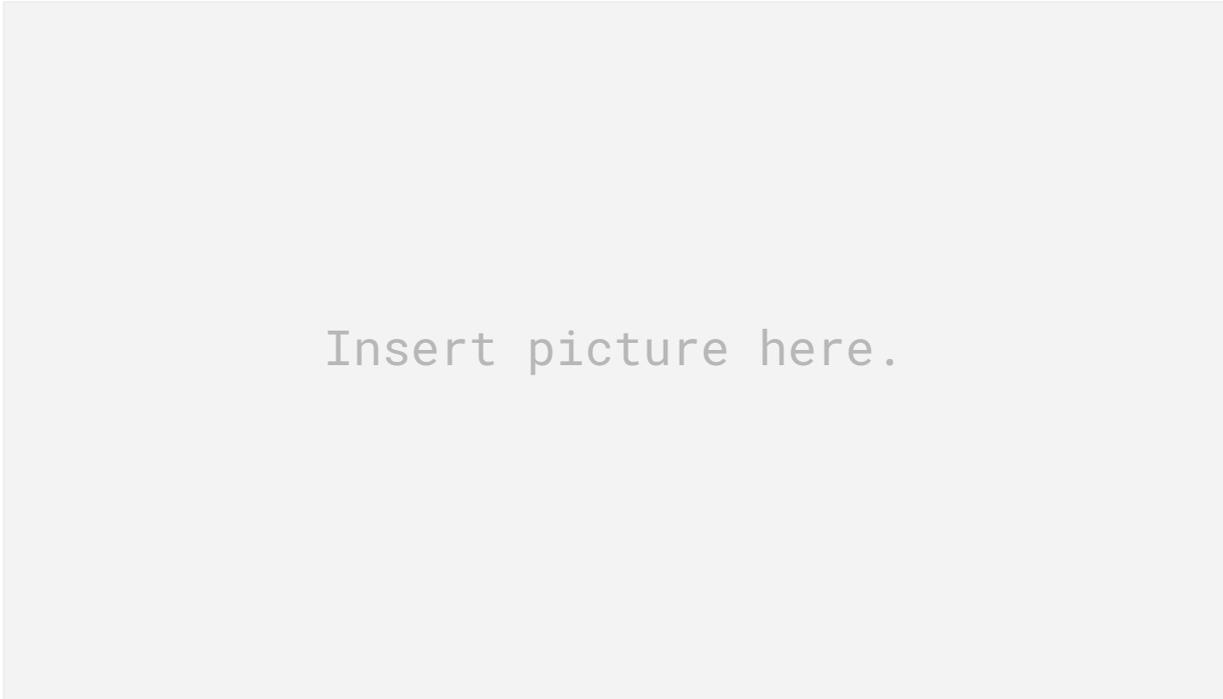
Micro:bit from the “Sunlight Sensor Project”

From: <https://microbit.org/projects/make-it-code-it/sunlight-sensor/>

Bellwork / Warm-up Question:

Homes almost *never* have completely flat roofs.

Why do you think houses have slanted roofs? 

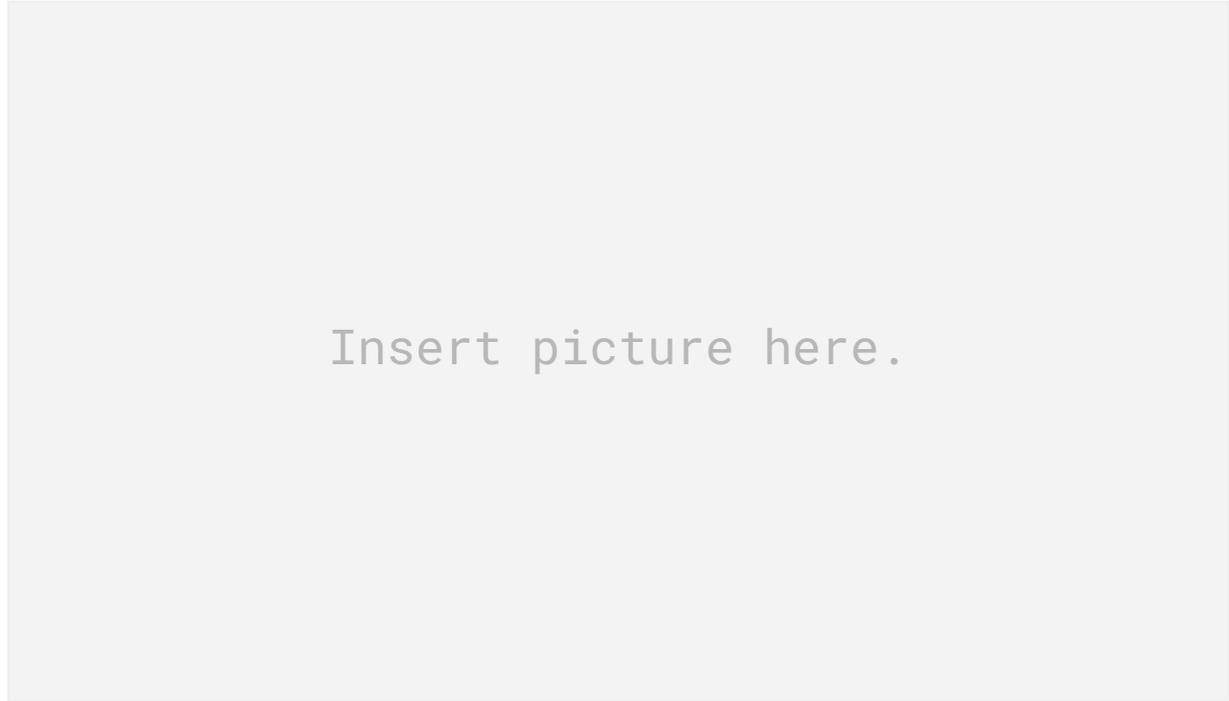


Constraints:

- 1. Floor plan must be less than 500 cm²
- 2. Must include a bathroom, kitchen, and bedroom
- 3. Roof angle cannot be 0 degrees (flat)

- 1. What materials will you use for the frame and roof? Remember your roof will have the additional weight of the battery.

- 2. Why? -----
- 3. Which angle will you place your solar panel at? -----
- 4. Why did you choose that angle?-----



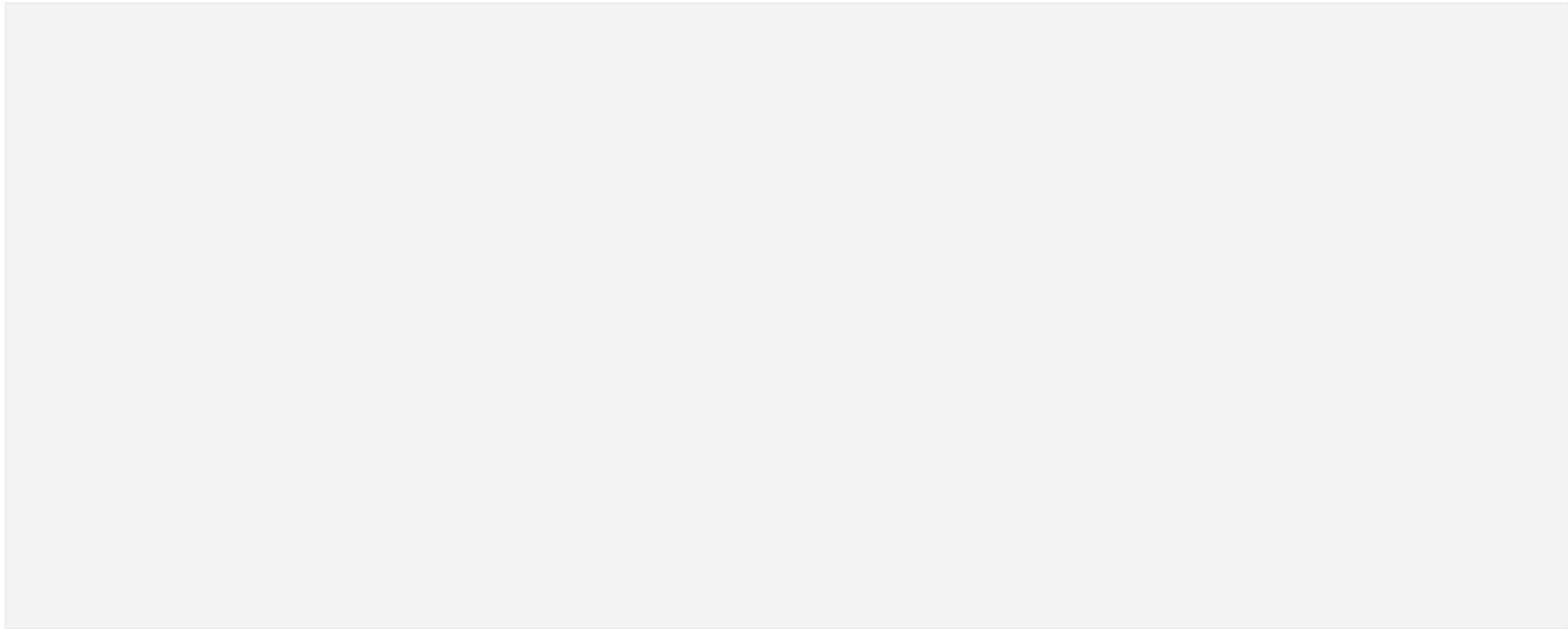
Insert picture here.

Constraints:

1. Floor plan must be less than 500 cm²
2. Must include a bathroom, kitchen, and bedroom
3. Roof angle cannot be 0 degrees (flat)

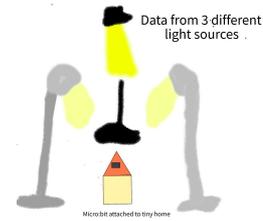
1. Which changes did you make from your original draft? _____

2. Why? _____



1. Why do we have a “forever” block surrounding our “plot bar graph” block?

2. What would happen if we set the plot light level to a maximum of 50 for all three of the light trials?



1. What was the the light input value from the three different locations?

Trial 1: _____

Trial 2: _____

Trial 3: _____

Average across all 3 trials: _____

2. Which trial provided the great light input and why do you think that was the case?

3. If given the opportunity, which other angle or panel location would you chose for your design and why?

4. Out of our entire class, which design provided the greatest average light output? What angle and position were their panels at?

5. Out of our entire class, which design do you think was the most unique or appealing and why?

6. Would the code below still provide the light input data we need? **First**, identify what was changed from our original code. **Second**, explain why or why not the code will still work.



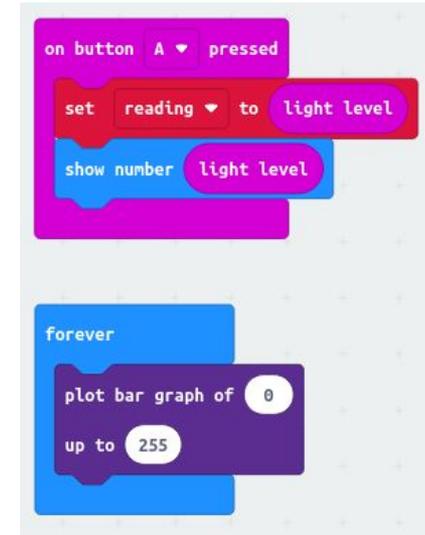
```
on button A pressed
  set foofoo to light level
  show number light level

forever
  plot bar graph of light level
  up to 255
```



```
on button A pressed
  set reading to 0
  show number light level

forever
  plot bar graph of light level
  up to 255
```



```
on button A pressed
  set reading to light level
  show number light level

forever
  plot bar graph of 0
  up to 255
```

Answer: _____

Answer: _____

Answer: _____
