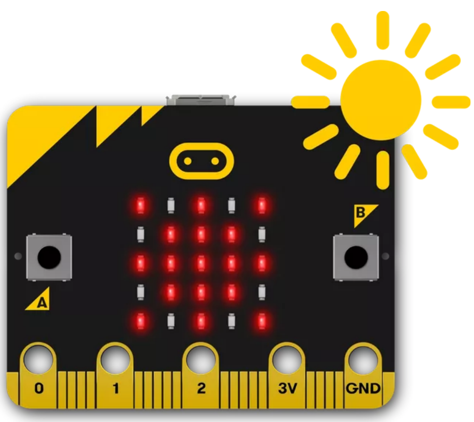
Modeling Solar Powered Tiny Homes with Micro:bits

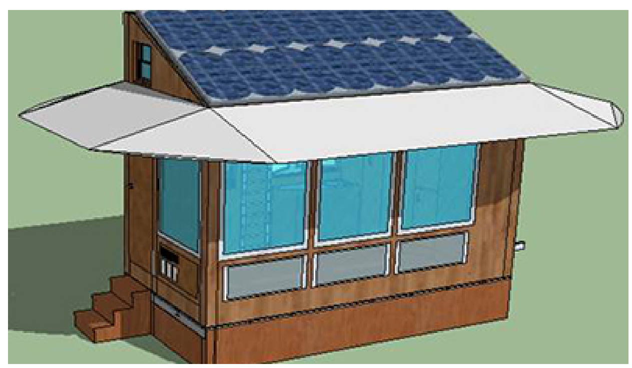
Middle School STEAM

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.](#)



Micro:bit from the "Sunlight Sensor Project"[3]



Tiny House Design from Northwestern's Tiny House Project[4]

[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: ttps://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.

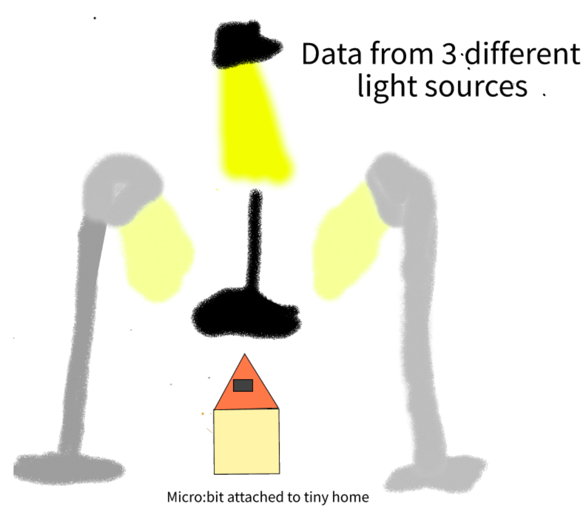
Additionally, concepts of maximizing area and volume, material and weight considerations for 3D design, input stimuli and output data from microcontrollers, and block code programming are all incorporated in this PBL curriculum. These are foundational material , engineering, and computer science concepts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

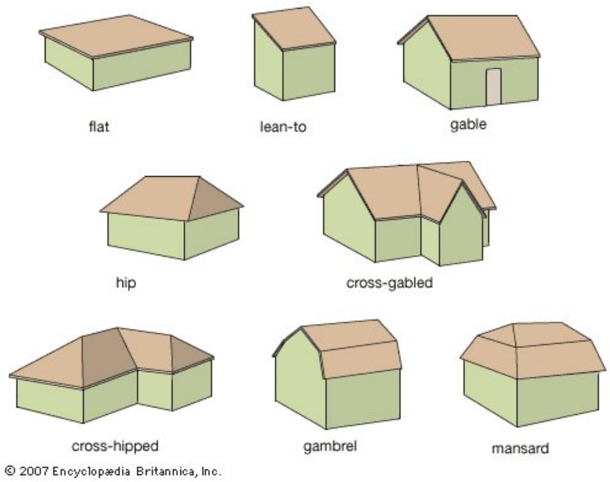
While this lesson is made for all student levels, it would benefit students to have some understanding of how our current electric grid is designed, and how solar panels play an important role in decreasing carbon footprints and increasing energy independence. Students should also be familiar with finding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

Introduction/Motivation

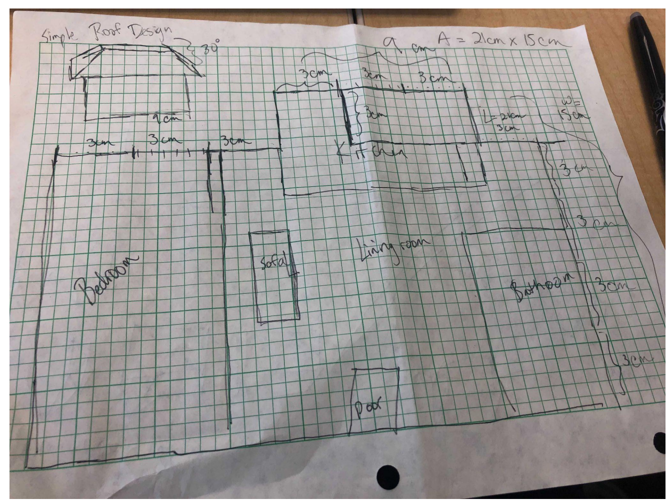
Ask students if they know where any solar panels in town are located. Show students various examples of tiny homes on Youtube and of unique solar panel technology (solar farm, solar buoys, solar windows). Discuss the pros and cons of both tiny homes and solar panel installation. What are some reasons 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.

Procedure

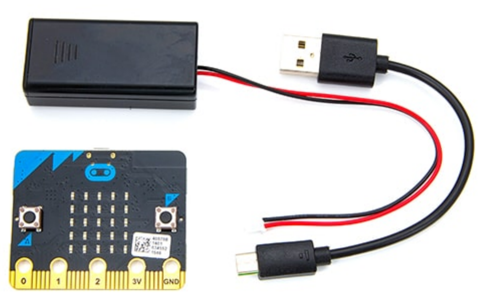
<u>Phase One: Design Draft (1-2 days)</u> On cm. by cm. graph paper, students will make a rough sketch of their tiny home floor plan and roof design/angle. In order to increase solar power, what angle should students place their solar panel at? Students will draw their floor plan to include a bathroom, bedroom, and kitchen totaling less than 500 cm. sq. They must indicate 1) what materials they plan to use for the frame and why, and 2) how the specified angle of roof design will optimize solar power. Teachers will probably have to show students how to use a protractor to estimate angle.



Roof Examples from Britannica[5]



Remind students that the Micro:bit battery (image below[6]) must attach or hang from the side of the roof, which will considerably add to the weight of the roof. This would be a great place to talk about the "battery challenge" of solar power (and DC/AC, inverters, etc.)
The final design must be approved by the teacher.



[5] "Roof." *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., www.britannica.com/technology/roof.
[6] Image from Microsoft MakeCode-- https://makecode.microbit.org/courses/csintro/making/activity

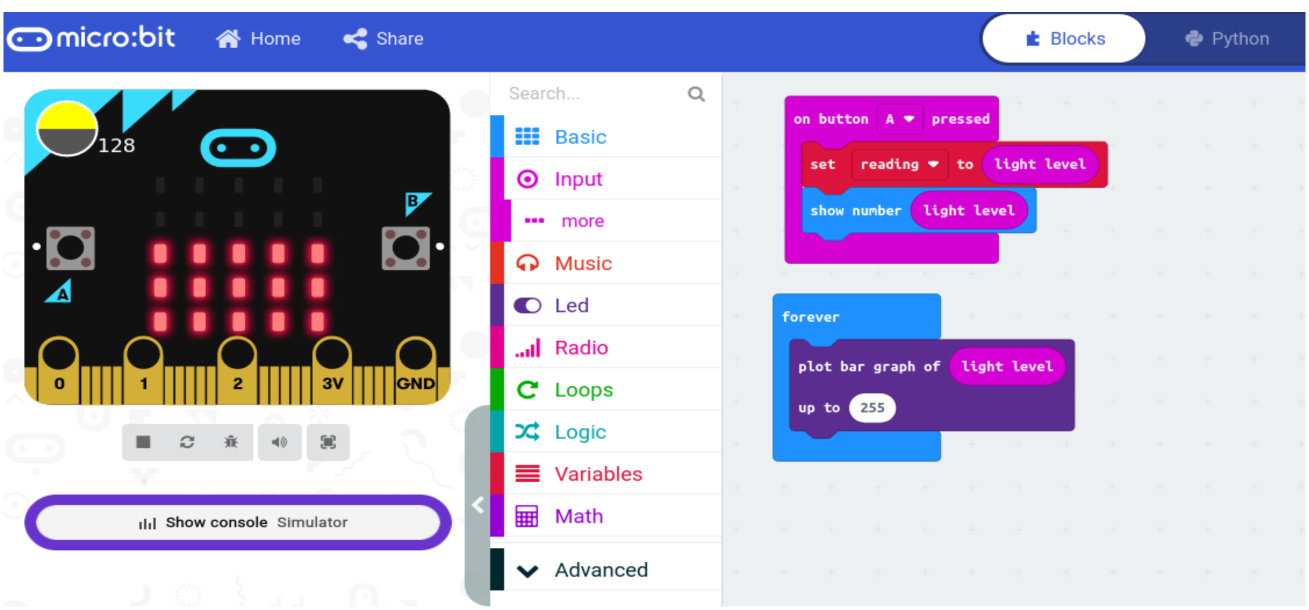
Phase Two: Build the Design (2-3 days) Begin building the tiny house with the provided materials. Make sure students build walls separating the rooms, and measure the angle of the roof where the Micro:bit is attached.

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.



<u>Phase Four: Finish Student Report</u>

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.

Relation to the Jankowski Lab at BSU

Computational modeling is used in Dr. Jankowski's lab to accelerate the identification of suitable organic photovoltaics (OPV), an emerging technology in solar panel development. Structural and electronic properties can be predicted using computer models of more candidate OPV materials than can be tested with wet lab experiments. From these predictions researchers can identify the best OPV candidates more inexpensively than without modeling. Learning how to assemble and organize these computer simulations was the joint objective of Dr. Jankowski's RET and REU students in 2021.

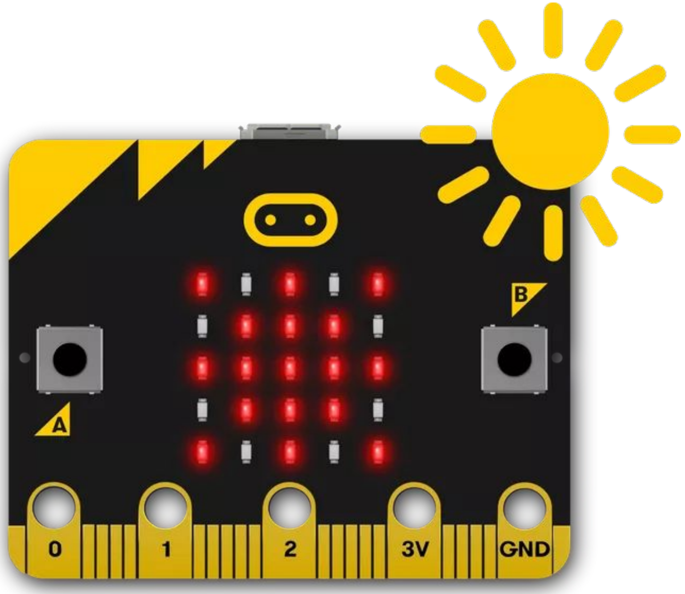
There are four main alignments between this lesson and Dr. Jankowski's lab in BSU's Material Science Department:

1) Building physical models to address scientific inquiries has always been a core cognitive skill to develop in middle school STEAM education. However, 21st century K-12 education has shifted to greater emphasis on *computer* modeling and simulations. In this lesson plan, I am incorporating both physical and computer modeling to utilize the logistical and financial benefits of applying both to emerging technologies.
2) The plastic polymers researched in the Jankowski lab must have properties appropriate for large scale for roll-to-roll manufacturing. At Tri-North Middle School, we will be encouraging students to take full advantage of our new MakerSpace (3D

printers, laser printers, vinyl cutters) for their tiny house structural design. This will allow students to discover the advantages and limitations of different plastic materials available in digital fabrication.

3) Dr. Jankowski's lab utilizes Python code, we will be using age-appropriate block coding to introduce concepts of functions and loops.
4) Students will be using the Micro:bit micro-computer as a proxy for solar panels. While a similar tiny home project could be executed with the use of actual mini solar panels, the Micro:bit provides light data that is automatically compiled in a .csv. The alternative of using mini solar panels (soldering, multimeter use, P = IV, etc.) requires an entire separate introduction into electricity. Therefore, just as computer simulations are used in the Jankowski lab for frontloading questions of efficient configuration, Micro:bit and MakeCode programming can allow students to model designs that maximize solar potential without needing to introduce other variables of electricity.

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?

Picture of Draft

Insert picture here.

Constraints:

1. Floor plan must be less than 500 cm^2
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?____

Picture of Final Construction

Insert picture here.

Constraints:

1. Floor plan must be less than 500 cm^2
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? ____

____

Screenshot of Micro:bit MakeCode

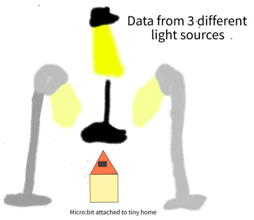
1. Why do we have a “forever” block surrounding our “plot bar graph” block?

   ____

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

   ____

Analysis of Light Input Data



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

    Trial 1: _________

    Trial 2: _________ Average across all 3 trials: ________

    Trial 3: _________

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?

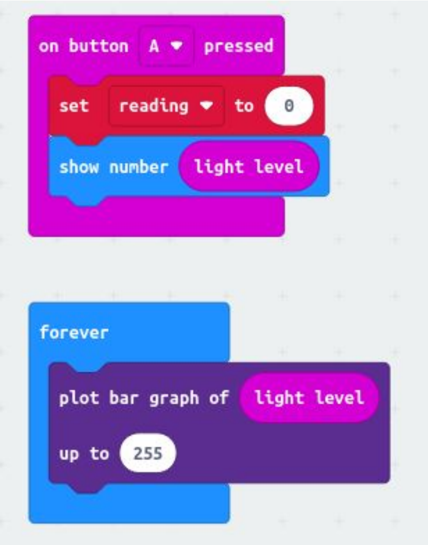
    ____

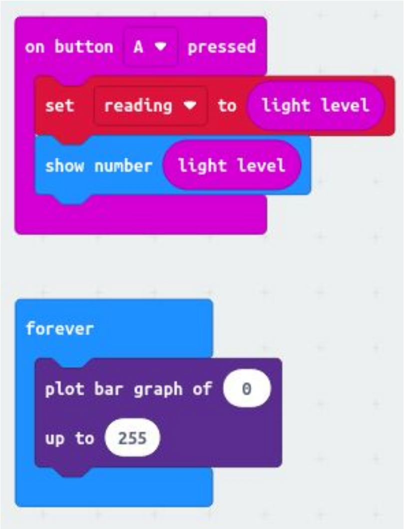
5. Out of our entire class, which design do you think was the most unique or appealing and why?

    ____

    ____

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.





Answer: ____________________

____

____

____

____

Answer: ____________________

____

____

____

____

Answer: ____________________

____

____

____

____