Light Pollution

Year level band:  7-8

Description:
This lesson introduces students to littleBits programming, while at the same time allowing them to think about environmental and sustainability challenges.

Type: General Purpose Programming Language

Resources:
- littleBits Arduino Coding Kit
- littleBits Arduino Coding Kit Invention Guide (included with the kit)
- Mac or PC with latest Arduino IDE installed:
  For Mac: http://arduino.cc/en/Guide/MacOSX
- Spare 9V batteries
- Phillips-head screwdrivers (possibly not required in introduction lesson)
- Attached sketch for turning light off after 5 secs
- Construction materials (cardboard, sticky tape, glue, etc) (optional)
- littleBits Deluxe Kit (optional)

Prior Student Learning:
Previous introduction to algorithms and programming would be an advantage.

Digital Technologies Summary

Students broaden their programming experiences to include general-purpose programming languages, and incorporate subprograms into their solutions. They predict and evaluate their developed and existing solutions, considering time, tasks, data and the safe and sustainable use of information systems, and anticipate any risks associated with the use or adoption of such systems.

<table>
<thead>
<tr>
<th>Band</th>
<th>Content Descriptors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 7 and 8</td>
<td>- Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)</td>
</tr>
<tr>
<td></td>
<td>o identify and clarify the main arduino functions that allow for reading input from the various sensors connectors, such determining if the light button is pressed</td>
</tr>
<tr>
<td>Element</td>
<td>Summary of tasks</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Learning hook| **Introduction to Light Pollution**  
Discuss with students the idea of Light Pollution.  
  - What is light pollution?  
  - Why is this a problem?  
Show this short video from BTN (3.56mins)  
[http://www.abc.net.au/btn/story/s4230574.htm](http://www.abc.net.au/btn/story/s4230574.htm)  
You may also like to use this website to further develop understanding.  
Discuss how light pollution is causing many problems around the world and ask students to list some of the things that communities are already |
doing to try and reduce the effects of light pollution, such as having lights direct their glow to where it is needed.

Ask students to brainstorm other ways we could reduce Light Pollution. Encourage all ideas and record these for students to consider.

Direct students thoughts to consider the types of things they can do in their own homes each night to reduce light pollution. Encourage thinking around turning off lights when no one is in the room.

Ask students to consider the idea of having senor lights within a house.

- How would this help light pollution?
- What types of things would we need to consider in order to design this type of lighting for a house?
- How could we design and build a prototype?

<table>
<thead>
<tr>
<th>Achievement Standards</th>
<th>Learning Map (Sequence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>By the end of Year 8, students plan and manage digital projects to create interactive information. They define and decompose problems in terms of functional requirements and constraints. Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability.</td>
<td></td>
</tr>
</tbody>
</table>
| - Students brainstorm ideas to alleviate Light Pollution in the world.  
  - Students work together to plan a solution to a problem  
  - Students construct a prototype model (optional)  
  - Student use littleBits to construct a circuit  
  - Students create/adjust a series of instructions (Algorithms) with alternative solutions should one plan fail to program the littleBits circuit. |

<table>
<thead>
<tr>
<th>Learning input</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to littleBits</td>
<td></td>
</tr>
</tbody>
</table>
| Show students the littleBits Kit and some of the bits which are inside.  
  - What do you think these might be?  
  - How do you think they work?  
  - How do you think these will help us with the Light Pollution issue?  |
| Show students the following TED Talk to help build excitement about the littleBits kits. This video should also help students to understand how the bits connect together and the significance of their colours.  
https://www.ted.com/talks/ayah_bdeir_building_blocks_that_blink_bEEP_and_teach#t-307212 |
Discuss the colour coding and how electricity flows through the littleBits. You may also like to expand on their understanding of electronics vocabulary such as voltage, resistance, current, amps, etc

Hand out the kits and give students a small amount of time to experiment and start connecting bits together.

- Ask students to see if they can make a light come on.
- Then ask if they can use a button to control the light.
- If time, you may like to give students more experimental time to see what else they can make.

Introduce or elaborate on programming languages and the importance of clear instructions (Algorithms) when setting tasks.

Introduce the concept of functions and discuss how arduino has pre-defined functions for all the LilyPad components. Some of these functions include:

- loop - for repetition
- delay - delays action
- pinMode - for configuring the specified pin to behave either as an input or an output
- digitalWrite - for writing a low or high value to a pin

A full documentation and description of these functions is available at:


Students will need to use the bargraph for the light if only the Arduino kit is available.

Introduction to Arduino and Programming

Once students have been introduced to the littleBits kits, they will then need to learn how to program these to achieve the desired effect (Lights to automatically
switch off after a designated amount of time, after a button is pushed to turn it on).

This link will help you to get started if you are new to the Arduino Coding kit
http://discuss.littlebits.cc/t/getting-started-with-arduino/109

Demonstrate to students:
- How to connect the Arduino to the computer using the USB cable
- How to ensure the correct board / port is selected
- How to upload
- How to upload a sketch?

Introduce or elaborate on programming languages and the importance of clear instructions (Algorithms) when setting tasks.

Before downloading the sketch, students write the algorithm of what needs to happen, the algorithm for turning light off after (5) seconds:

1. If button is pressed, light comes on
2. Wait five seconds
3. Light goes off

A useful way for guiding the students towards decomposition is by getting them to think about each littleBit in terms of its key functionality, its inputs and potential outputs, and the key Arduino functions that guide their functionality.

Students then identify the key functions that can be used to solve this problem:

1. If button is pressed, light comes on - digitalRead (to check the button), digitalWrite - to turn the light on
2. Wait five seconds - delay
3. Light goes off - digitalWrite - to turn the light off

Students load the code (see appendix) into the Arduino IDE and identify the key functions.

Students identify any other functions used (such as readAxis) and discuss why they are used.

Students run and debug the code as needed.

Encourage students to test and debug on a regular basis as they work through this task.

<table>
<thead>
<tr>
<th>Learning construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Students work in small groups to construct a house from cardboard or other construction materials, or possibly source an old dolls house for this task.</td>
</tr>
</tbody>
</table>

Students will then wire the lighting in the house using the littleBits kit.

Students will then program the Arduino to make the lights in the house turn off after a certain amount of time, after pushing the button.

As students are working remind them not to worry if it doesn’t work the first time or it isn’t perfect. It is an important element on the design process to test and debug as you go along.

<table>
<thead>
<tr>
<th>Learning demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask students to demonstrate their working models (or just the circuits) to the rest of the class. Discuss how the circuit is connected and how the</td>
</tr>
<tr>
<td>programming works to achieve the desired outcome.</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>- What have you changed in this code and why?</td>
</tr>
<tr>
<td>- What do you think are the most important functions?</td>
</tr>
<tr>
<td>- Have you had to do any debugging?</td>
</tr>
<tr>
<td>- Can you think of an alternative way to achieve the same outcome?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning reflection</th>
<th>During the reflection stage, ask students the following questions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- How does the littleBits circuit work?</td>
<td></td>
</tr>
<tr>
<td>- What were the biggest challenges?</td>
<td></td>
</tr>
<tr>
<td>- What were the fun moments?</td>
<td></td>
</tr>
<tr>
<td>- What did you learn from creating this?</td>
<td></td>
</tr>
<tr>
<td>- What would you do next time if you had more time to keep improving your design?</td>
<td></td>
</tr>
<tr>
<td>- What other ideas do you have for helping to reduce light pollution?</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>During the reflection stage, ask students the following questions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- How does the littleBits circuit work?</td>
</tr>
<tr>
<td>- What were the biggest challenges?</td>
</tr>
<tr>
<td>- What were the fun moments?</td>
</tr>
<tr>
<td>- What did you learn from creating this?</td>
</tr>
<tr>
<td>- What would you do next time if you had more time to keep improving your design?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reflect on students’ experiences modifying and writing code.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- What challenges did you have when programming the littleBits Arduino? How did you resolve those?</td>
</tr>
<tr>
<td>- Were you able to easily change existing code to produce a different outcome?</td>
</tr>
<tr>
<td>- Did you learn any new terminology?</td>
</tr>
<tr>
<td>- What were the rewarding parts of coding in pairs?</td>
</tr>
<tr>
<td>- What are the advantages/disadvantages of using functions in code?</td>
</tr>
<tr>
<td>- Did they discover an interesting/useful function they want to share?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remind students that littleBits is a prototyping platform but that the code and circuits we have been exploring exist in real-world products.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- What real-world situations would you expect similar code to be used?</td>
</tr>
<tr>
<td>- Perhaps give the example of traffic lights - how do you think these worked prior to digital technology?</td>
</tr>
<tr>
<td>- Can you think of any exciting products that could be created with this sort of technology?</td>
</tr>
<tr>
<td>- What other components could be used as inputs or outputs? Consider what digital and analogue inputs and outputs you see each day at school, at home, in transport, sports grounds, etc.</td>
</tr>
</tbody>
</table>
Assessment:

Formative Assessment

- Teachers observe students using the Arduino, creating their algorithms and debugging.
- Use questioning to elicit student understanding of the functions of littleBits and Arduino, the programming platform and their algorithmic thinking.
- You might take photos/videos of the students’ work to document their progress – or in the final presentations.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Quantity of knowledge</th>
<th>Quality of understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre-structural</td>
<td>Uni-structural</td>
</tr>
<tr>
<td>Algorithms Programming</td>
<td>No program written.</td>
<td>Algorithm only shows a limited number of instructions but do not allow use of button, or automatic turn off.</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>No specific / technical terms used.</td>
<td>The terms program or code may be used as a general description.</td>
</tr>
</tbody>
</table>
Teacher/Student Instructions:

When you turn on the power bit it can take a few seconds for the Arduino bit to start up - wait for the flashing light on the Arduino bit and then you are ready to go.

If using the mounting board:

- Snap your circuit together before pressing into the mounting board
- Press down Bitsnaps (the coloured edges) rather than the white circuit board

If you encounter any problems setting up the software, check the Arduino troubleshooting site: http://arduino.cc/en/Guide/Troubleshooting

CSER Professional Learning:

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

- F-6 Digital Technologies: Foundations
  - Unit 7: Algorithms and Programming
- F-6 Digital Technologies: Extended
  - Unit 2: Algorithms & Programming Extended
- 7-8 Next Steps
  - Unit 2 - Functions
  - Unit 3 – problem definition and design

See: http://csermoocs.adelaide.edu.au/moocs

Further Resources:

- http://littlebits.cc/
- http://littlebits.cc/6arduino-sketches
- http://www.abc.net.au/btn/story/s4230574.htm
- https://www.ted.com/talks/ayah_bdeir_building_blocks_that_blink_bEEP_and_teach#t-307212
- Digital Technologies Hub: www.digitaltechnologieshub.edu.au_CSER:
- https://csermoocs.adelaide.edu.au
Appendix 1 – LED on for 5 secs

/*
littleBits Arduino Module

_LED_on_for_5_secs_

What is supposed to happen:
This sketch turns on an output module for a number of seconds when a button is
pressed

This sketch includes the following commands:
pinMode( pin, mode ); pin = a digital pin, mode = OUTPUT or INPUT
digitalWrite( pin, value ); pin = a digital pin that has been set to OUTPUT, value = HIGH
or LOW
delay( time ); time = time of delay in milliseconds
digitalRead(pin); pin = a digital pin
*/

/*
setup - This is where we initialize the sketch
Setup includes commands that set the stage for the rest of the sketch.
The commands inside of setup are only executed once.
After all the commands in setup have been executed, the sketch moves to loop.
*/
void setup() { // this is the beginning of setup

  // Since we are going to use pin 11 to control an output module
  // we need to set the pinMode of pin 11 to OUTPUT,
  // this means voltage will flow out of the pin.
  pinMode(11, OUTPUT); // set pin 11 to OUTPUT mode
  pinMode(A5, INPUT); // set A5 as the input - button

} // this is the end of setup

/*
loop - This is the main part of every arduino sketch.
Loop is a series of commands that executed one after another in order of top to bottom.
When all the commands have been executed, loop begins again from the top.
This goes on for as long as there is power running the Arduino.
*/
void loop() { // This is the beginning of loop, each command below is executed in order of
top to bottom
    if (digitalRead(A5) == HIGH){
        digitalWrite( 11, HIGH ); // Turn on pin 11 and thus turn on the output module attached
to 11
        delay( 5000 ); // wait for five seconds
// the number inside the () is the amount of time in milliseconds to wait: 1000ms = 1
second
// change the number inside the () to change length of delay
    }
    else {
        digitalWrite( 11, LOW ); // Turn off pin 11 and thus turn off the output module attached to
11
    }
} // this is the end of loop, now return to the beginning of loop and execute the
commands again