DIY Computer Mouse

**Year level band:** 7-8

**Description:**
In this project, students build their own computer mouse using the littleBits kit and the general purpose programming language in the Arduino IDE.

**Type:** General purpose programming

**Resources:**
- littleBits Arduino Coding Kit
- littleBits Arduino Coding Kit Invention Guide (included with the kit)
- Mac or PC with latest Arduino IDE installed:
- Spare 9V batteries
- Phillips-head screwdrivers
- Construction materials (cardboard, sticky tape, glue, etc) (optional)
- littleBits Deluxe Kit (optional)
- This lesson is based on the littleBits Challenge published 15/5/2014 [http://littlebits.cc/projects/diy-computer-mouse](http://littlebits.cc/projects/diy-computer-mouse)
- littleBits TED Talk [https://www.ted.com/talks/ayah_bdeir_building_blocks_that_blink_bEEP_and_teach#t-307212](https://www.ted.com/talks/ayah_bdeir_building_blocks_that_blink_bEEP_and_teach#t-307212)

**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</td>
</tr>
<tr>
<td>Element</td>
<td>Summary of tasks</td>
</tr>
<tr>
<td>---------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| **Learning hook** | Discuss with students the purpose of a computer mouse.  
- What do we use a mouse for?  
- What are the alternatives to using a computer mouse (e.g., touch screen, touch pad, etc)?  
- How does a computer mouse work?  
- What are the main operations that a mouse can do?  
If you have access to an old/unused mouse, you might like to pull it apart to show students what it looks like on the inside. This You Tube video is also a good way to explore this  
[https://www.youtube.com/watch?v=.knzJ4wPh04](https://www.youtube.com/watch?v=.knzJ4wPh04) |
<p>| <strong>Achievement Standards</strong> | 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 |</p>
<table>
<thead>
<tr>
<th>Learning Map (Sequence)</th>
<th>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.</th>
</tr>
</thead>
</table>
|                        | • Students discuss the purpose and functions of a computer mouse.  
|                        | • Students learn how to create their own computer mouse using the littleBits.  
|                        | • Students create and program their own mouse and test it. |

<table>
<thead>
<tr>
<th>Learning input</th>
<th>If this is the first time students have used littleBits it is a good idea to spend some time introducing these to the class.</th>
</tr>
</thead>
</table>
|               | **Introduction to littleBits**  
|               | 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](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.  
|               | **Introduction to Arduino and Programming**  
|               | Once students have been introduced to the littleBits kits, they will then need to learn how to program these.  
|               | 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](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.

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:
Refer to the below link to download the sketch for this project. Depending on the students’ ability, you may like to take a section of the sketch out so that students will need to program a section themselves.

http://littlebits.cc/projects/diy-computer-mouse

### Learning construction

Students will work individually or collaboratively to create their mouse using the littleBits Arduino Coding Kit.

Build the circuitry as in the image.

![Image of the littleBits Arduino Coding Kit](image)

Before downloading the sketch, students write the algorithm of what needs to happen, following these guidelines:

- The two slide dimmers control the x and y mouse position
- The button controls mouse press
A sample algorithm decomposed into key steps could be:

1. Connect to and initialize circuit
2. Read x mouse position from dimmer1
3. Read y mouse position from dimmer2
4. Move mouse to x and y
5. if the button is pressed, press the mouse

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. Connect to and initialize circuit
2. Read x mouse position from dimmer1 - analogRead
3. Read y mouse position from dimmer2 - analogRead
4. Move mouse to x and y - Mouse.move
5. if the button is pressed, press the mouse - digitalRead, Mouse.press

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.

| Learning demo | Ask students to demonstrate their working computer mouses to the rest of the class. Discuss how the circuit is connected and how the program works to achieve the desired outcome.
|               | ● What have you changed in this code and why?
|               | ● What do you think are the most important functions?
|               | ● Have you had to do any debugging?
|               | ● Can you think of an alternative way to achieve the same outcome?

| Learning reflection | During the reflection stage, ask students the following questions:
|                    | ● How does the littleBits circuit work?
|                    | ● What were the biggest challenges?
|                    | ● What were the fun moments?
|                    | ● What did you learn from creating this?
|                    | ● What would you do next time if you had more time to keep improving your
Formative Assessment:

Reflect on students’ experiences modifying and writing code.

- What challenges did you have when programming the littleBits Arduino? How did you resolve those?
- Were you able to easily change existing code to produce a different outcome?
- Did you learn any new terminology?
- What were the rewarding parts of coding in pairs?
- What are the advantages/disadvantages of using functions in code?
- Did they discover an interesting/useful function they want to share?

Remind students that littleBits is a prototyping platform but that the code and circuits we have been exploring exist in real-world products.

- What real-world situations would you expect similar code to be used?
- Perhaps give the example of traffic lights - how do you think these worked prior to digital technology?
- Can you think of any exciting products that could be created with this sort of technology?
- 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.

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.
- Teachers 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></td>
<td></td>
</tr>
<tr>
<td></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></td>
<td></td>
</tr>
<tr>
<td></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 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

Author: Lauren Stanhope
Appendix - Code listing

/*
littleBits Arduino Module
code originally from Arduino Reference
modified by Matt Richard for littleBits

_Mouse Move And Click_

What is supposed to happen:
* Control your mouse(computer, not pet) with your littleBits Arduino Module!!
* Two slide dimmers control the x and y mouse position,
while a button attached to digital pin d0 controls mouse press

Circuit:
* littleBits slide dimmer on analog pin A0
* littleBits slide dimmer on analog pin A1
* littleBits button on digital pin d0

*/

const int xAxis = A0; // analog sensor for X axis
const int yAxis = A1; // analog sensor for Y axis

int range = 12; // output range of X or Y movement
int responseDelay = 2; // response delay of the mouse, in ms
int threshold = range/6; // resting threshold
int center = range/2; // resting position value
int minima[] = {
0, 0);    // actual analogRead minima for {x, y}
int maxima[] = {
1023, 1023};    // actual analogRead maxima for {x, y}
int axis[] = {
  xAxis, yAxis};  // pin numbers for {x, y}
int mouseReading[2];    // final mouse readings for {x, y}

void setup() {
  //Serial.begin(9600);
  pinMode(0, INPUT);
  Mouse.begin();
}

void loop() {

  // read and scale the two axes:
  int xReading = readAxis(0);
  int yReading = readAxis(1);

  //Serial.println(xReading);
  //Serial.println(yReading);
  //Serial.println();
  // move the mouse:
  Mouse.move(xReading, yReading, 0);
  delay(responseDelay);

  // read button and click mouse:
  //if the switch attached to pin d0 is pressed
  if(digitalRead(0) == HIGH){
    //press and hold the right mouse button
    //Serial.println("mousePress");
    Mouse.press();
// if the switch attached to pin d0 is not pressed
if(digitalRead(0) == LOW){
    // release the right mouse button
    // Serial.println("mouseRelease");
    Mouse.release();
}

/*
 reads an axis (0 or 1 for x or y) and scales the
 analog input range to a range from 0 to <range>
 */

int readAxis(int axisNumber) {
    int distance = 0;  // distance from center of the output range

    // read the analog input:
    int reading = analogRead(axis[axisNumber]);

    // map the reading from the analog input range to the output range:
    reading = map(reading, minima[axisNumber], maxima[axisNumber], 0, range);

    // if the output reading is outside from the
    // rest position threshold, use it:
    if (abs(reading - center) > threshold) {
        distance = (reading - center);
    }

    // the Y axis needs to be inverted in order to
    // map the movement correctly:
    if (axisNumber == 1) {
        distance = -distance;
    }
// return the distance for this axis:
return distance;