

# LilyPad Personalised Alert Buzzer

**Year level band:** 7-8

**Description:** In this lesson students will create a personalised musical buzzer by programming the LilyPad Arduino to play a sound using the Main board, Buzzer and Button, coding the note frequency in Arduino IDE. The lesson can be extended to include LED's that light up or flash according to the note played to enable the buzzer to be used for the hearing impaired.

During the testing phase, the LilyPad components will be connected using alligator clips before progressing to hand stitching them together using conductive thread for a final product that can be held in a person's hand or attached to their clothing.

The buzzer is to be personalised with a tune or tone of the student's choosing and will be used as an alert system for a person who is incapacitated.

Type: General Purpose Programming Language

## Resources:

- LilyPad ProtoSnap kit with rechargeable battery
- LilyPad LED's
- Conductive thread
- Alligator clips
- Arduino IDE installed onto computers
- Mini USB cable
- Chalk or pen for marking fabric
- Felt/Material
- Scissors
- Paper
- Needle
- Pencils for design sketches
- Embroidery hoop (suggested)
- Mac or PC with latest Arduino IDE installed:

For Windows: <http://arduino.cc/en/Guide/Windows>

For Mac: <http://arduino.cc/en/Guide/MacOSX>

## Prior Student Learning:

An understanding of basic circuitry is essential for this activity as it is important that the threads do not cross over or touch as you may short circuit the board.

Simple hand stitching skills would be beneficial

<b>Digital Technologies Summary</b>
-------------------------------------

Students will use technology to create a personalised alert audio buzzer that can also incorporate a visual aid. They will focus on further developing understanding and skills in computational thinking such as decomposing problems and prototyping.

This task provides an opportunity for students to create a digital solution that is interactive, programmable and related to a real world situation. They progress from designing the user interface to considering user experience factors such as user expertise, accessibility and usability requirements.

They 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.

Band	Content Descriptors
7-8	<p>Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)</p> <ul style="list-style-type: none"> <li>determining the factors that influence proposed solution ideas, for example user dexterity affects the size of buttons, hearing or vision loss concerns when designing the alert buzzer</li> </ul> <p>Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)</p> <ul style="list-style-type: none"> <li>presenting and comparing alternative designs to a solution for a problem, for example presenting alternative design mock-ups to the class</li> </ul> <p>Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)</p> <ul style="list-style-type: none"> <li>comparing student solutions with existing solutions that solve similar problems eg comparing current alter buzzers with their own</li> </ul> <p>The particular elements of Critical and Creative Thinking addressed by this content description</p> <p>Generating ideas, possibilities and actions</p> <ul style="list-style-type: none"> <li>Imagine possibilities and connect ideas</li> <li>Consider alternatives</li> </ul> <p>Analysing, Synthesising and evaluating reasoning and procedures</p> <ul style="list-style-type: none"> <li>Draw conclusions and design a course of action</li> <li>Evaluate procedures and outcomes</li> </ul>

Element	Summary of tasks

<p>Learning hook</p>	<p>Discussion prompts:</p> <ul style="list-style-type: none"> <li>● What would the advantages be of a wearable personal buzzer?</li> <li>● Why would it be useful for a personalised alert tone'? (Carers would know who was in need of attention quickly)</li> <li>● How could the buzzer be adapted for use in a noisy environment? (Include LEDs)</li> <li>● What other applications would a personalise buzzer be useful for? (eg. Quiz game buzzer)</li> </ul> <p>Pose a scenario to the students that they have been approached by a Retirement Village to produce personalised buzzers for a range of clients who have varying needs. These include but are not limited to the elderly who are hearing impaired, visually impaired, have mobility issues and/or vocal impairment.</p> <p>These clients require a means of getting the attention of their carers when they are in areas of the Retirement village other than their room. These devices must be wearable or able to attach to them in some way and be aesthetically pleasing.</p> <p>Introduce the concept of how these devices are programmable and thus code can be written to achieve the planned outcomes. Introduce the concept of functions and discuss how arduino has pre-defined functions for all the LilyPad components. Some of these functions include:</p> <ul style="list-style-type: none"> <li>● loop - for repetition</li> <li>● delay - delays action</li> <li>● pinMode - for configuring the specified pin to behave either as an input or an output</li> <li>● digitalWrite - for writing a low or high value to a pin</li> </ul> <p>A full documentation and description of these functions is available at:  <a href="https://www.arduino.cc/en/Reference/">https://www.arduino.cc/en/Reference/</a></p>
<p><b>Achievement Standards</b></p>	<p>Students <b>explain</b> a range of needs, opportunities or problems and define them in terms of functional requirements and constraints.</p> <p>Students generate and document in digital and non-digital form, <b>design</b> ideas for different audiences using appropriate technical terms, and graphical representation techniques including algorithms.</p> <p>They independently and safely plan, <b>design</b>, test, modify and create a range of digital solutions that meet intended purposes including user interfaces and the use of a programming language.</p> <p>They <b>develop</b> criteria for success, including innovation and sustainability considerations, and use these to judge the suitability of their ideas, solutions and processes.</p>
<p>Learning Map (Sequence)</p>	<ul style="list-style-type: none"> <li>● Identify LilyPad main board, Buzzer and Button</li> <li>● Review basic electronics &amp; circuitry.</li> <li>● Explore Arduino IDE software that has been installed onto computers &amp; configure port plus select LilyPad in Board Manager</li> <li>● Connect LilyPad components together using alligator clips for testing</li> <li>● Type Alert Buzzer Basics program into Arduino IDE &amp; upload to LilyPad</li> </ul>

	<ul style="list-style-type: none"> <li>● Establish the needs of the client who the personalised buzzer will be designed for &amp; research the constraints that will limit the design.</li> <li>● Discuss possible designs and present to peers for feedback</li> <li>● Create diagram flow chart to show connections needed for LilyPad components and the chosen design of the Alert Buzzer</li> <li>● Create algorithm to create personalised musical sequence which includes more than three notes.</li> <li>● Students identify the ways in which the program can be decomposed into functions</li> <li>● Students identify the key arduino functions and learn how to change parameters and use these these functions to achieve planned outcomes</li> <li>● OPTIONAL - include visual component using LED's and alter algorithm in Arduino IDE. Include changes to the diagram chart to indicated additional changes.</li> <li>● Hand stitch LilyPad components onto fabric following the flow chart</li> <li>● Present final project to the class.</li> </ul>
Learning input	<p>The teacher should have an understanding of :</p> <ul style="list-style-type: none"> <li>● circuitry,</li> <li>● basic hand sewing techniques</li> <li>● the elements of C language used in the Arduino IDE  <a href="http://sewelectric.org/diy-projects/3-programming-your-lilypad/basic-code-elements/">http://sewelectric.org/diy-projects/3-programming-your-lilypad/basic-code-elements/</a>  <a href="https://ileadusaininois.files.wordpress.com/2015/03/lilypadarduino.pdf">https://ileadusaininois.files.wordpress.com/2015/03/lilypadarduino.pdf</a>  This activity can be completed in small groups or individually depending on the experience of the students with the technology &amp; Arduino IDE</li> </ul>
Learning construction	<ul style="list-style-type: none"> <li>● Students review knowledge of basic electronics and the issue of short circuits. <a href="http://www.instructables.com/lesson/Sew-a-Circuit/">http://www.instructables.com/lesson/Sew-a-Circuit/</a></li> <li>● Introduce conductive thread discussing issues with insulation &amp; resistance. The kit contains 2 ply thread that will compromise the brightness of the LED if sewn over a long distance. (This should not be a problem with this task)</li> <li>● Open Arduino IDE and connect board to computer using mini USB cable</li> <li>● Select the board type and port - Tools_Board Manager_ LilyPad Arduino. Then Tools_port (select the serial port that is corresponds to the USB port eg:COM3)</li> <li>● Set up LilyPad - Using alligator clips, connect the positive (+) on the Buzzer to Port 5 on the Main Board. Attach the negative (-) on the Buzzer to the negative port on the Main Board. Attach the (+) on the Button to Port A3 on the main board and (-) on the Button to (-) on the Main Board to complete the circuit.</li> <li>● Type the following code into the Arduino IDE and save as Alert Buzzer Basic</li> </ul> <pre>// Pin the buzzer is attached to int buzzerPin = 5;</pre>

```
// Pin the button is attached to
int buttonPin = A3;

// Variable to store the button's state
int buttonState = 0;

// Set a time in milliseconds for all delays
int delayTime = 100;

// Notes
const int C = 1046;
const int D = 1175;
const int E = 1319;
const int F = 1397;
const int G = 1568;
const int A = 1760;
const int B = 1976;
const int C1 = 2093;
const int D1 = 2349;

void setup()
{
  // Set the buzzer pin as an OUTPUT
  pinMode(buzzerPin, OUTPUT);
  //Set the button as INPUT
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop()
{
  buttonState = digitalRead(buttonPin);
```

```

if (buttonState == LOW)
{
  // Call a function named playSong()
  playSong();
} else
{
  noTone(buzzerPin);
}
delay(delayTime);
}
void playSong()
{
  tone(buzzerPin, C);
  delay(delayTime);
  // Use noTone() to shut off the buzzer and delay to create a 'rest'
  noTone(buzzerPin);
  delay(delayTime);
}

```

- Upload Sketch to LilyPad and test by pressing the Button. There should be a single pulse tone played when the button is pressed.
- Once the test program is successful, students move on to designing the user interface for the Alert Buzzer based on the needs of their client.
- Identify the key blocks of the code, in particular the setup() and loop() functions, and discuss their purpose. What about the delay() function? What other functions of interest can students find?
- Share designs with class members for feedback and make any changes are required.
- Create a labelled diagram flow chart to show connections needed for LilyPad components and the chosen design of the Alert Buzzer
- Currently the Alert Buzzer only plays a single tone using the frequency of C=1046. Make alterations to the program to incorporate a minimum of three varying tones.
- Students decompose this problem into key elements:
  - setting the first tone - using the tone function
  - delay - using the delay function
  - setting the second tone - using the tone function
  - delay - using the delay function

	<ul style="list-style-type: none"> <li>This can be done by adding additional lines of code under the new function called void playSong ()</li> </ul> <p>Eg:</p> <pre>tone(buzzerPin, D); delay(delayTime); tone(buzzerPin, A); delay(delayTime);</pre> <ul style="list-style-type: none"> <li>Students can explore the varying tones of each note added. The program will play sequentially through each command repeatedly when the button is pressed. Remind the students to upload the sketch each time they make a change.</li> <li>For a more advanced activity, LED's can be incorporated into the design at this stage but is not an essential component for this task.</li> <li>Hand stitch all components to fabric following the diagram flow chart.</li> <li>Present final product to class and evaluate designs based on the original design brief.</li> </ul>
Learning demo	Students present their final product to the class, together with their diagram flow chart and evaluate the Alert Buzzer against the original design brief.
Learning reflection	<p>How has the final product met the requirements of the Retirement Village brief?</p> <p>How user friendly is the design for the client? Can it be attached or worn with ease whilst still being accessible for the patient to push the Button?</p> <p>Is the alert tone personalised?</p> <p>How could they market this device? Could it be reproduced easily?</p> <p>What is the role of technology in this task? Could the Alert buzzer be produced without it?</p> <p>What are the advantages/disadvantages of using functions in code?</p> <p>Did they discover an interesting/useful function they want to share?</p>

**Assessment:**

**Formative Assessment**

- Observation of students creating circuit including Buzzer and Button
- Review diagram sketch with students prior to connecting the components with conductive thread
- Successful upload & execution of alert buzzer basic program

- Documentation of evidence of iteration of project, particularly changes made between the initial diagram sketch and the final product and the consideration of the needs of the client.
- Completed product with personalised alert tone and a user friendly design that meets all requirements of the task.

Criteria	Quantity of knowledge			Quality of understanding	
	Pre-structural	Uni-structural	Multi-structural	Relational	Extended abstract
Algorithms Programming	Some attempt to copy the provided code into the program has been made.	Provided code has been copied into the program successfully.  Limited customisation is evident. Eg. additional tones	Algorithm has been altered to include a personalised tone containing min 3 notes  Some instructions have been commented out in the program.	program has been extended to include 4-6 notes to create an interesting melody  Algorithm has instructions commented out through the program.	Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes branching, variables and loops. Eg: complex melody  Full use of Programming interface is evident
Diagram	Diagram contains all components of the project.  Some evidence of labels.	Complete diagram includes labels using correct vocabulary. Eg input/output, Button, Buzzer  Some notation of changes included.	Diagram is complete, clearly labeled includes symbols and is easy to follow.  Changes or iterations evident.	Specific vocabulary is used throughout the diagram.  Proper symbols and explanation key is included.  Iterations made during the project are documented.	Correctly labelled diagram complete and presented to an excellent standard.  Complete detail of all connections and hardware including description of components included.  Iteration and changes documented,

					including a final reflection.
Vocabulary	No specific / technical terms used.	The terms program or code may be used as a general description.	The terms program or code are used as a general description. The terms analogue and digital are known and used correctly.	Specific terms such as program, loop, debug are used confidently with specific reference to learner's work. Code is commented in specific places.	Understanding of specific terms such as constant, function and variable. Understanding of terms in the code such as <code>digitalWrite</code> . Code is well commented throughout.

## Teacher/Student Instructions:

This task requires a more advanced level of programming using the Arduino IDE and incorporates two or more components of the LilyPad Kit

## CSER Professional Learning:

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

- CSER 7 & 8 Digital Technologies: Next Steps
- Unit 2.2 - General Purpose Programming Languages
- Unit 2.3 - Functions

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

## Further Resources:

<https://www.instructables.com/howto/lilypad/> Website with additional ideas and instructions.

<http://www.instructables.com/class/Wearable-Electronics-Class/> Free online course teaching techniques to build wearable electronics and program interactions using Arduino software. There are variances in the LilyPad kit supplied by CSER and those referred to in the lessons.



Author: Toni Falusi

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). Computer Science Education Research (CSER) Group, The University of Adelaide.