DT Challenge Year 7/8 Arduino – Sound

Module 1: Hello, Arduino!
https://groklearning.com/learn/aca-dt-78-ar-sound/1/0/

Previous: None

Next: Using Sensors

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as integer and string
Specification, algorithms, implementation	Simple Algorithms, user input
Digital Systems	
Interactions	Interaction (input/output)
Impact	

Objectives (Content Descriptions)

ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints.
ACTDIP030	Implement and modify programs in a general-purpose programming language.

What are we learning? (Abstract)

Embedded computing, involving microcontrollers (really small, cheap computers that perform a single function), are what run most pieces of technology today. Including your mouse, drill, remote control and car. This module introduces embedded programming through the Arduino Esplora. We learn how programs run, control flow, syntax or the Arduino language, and how we control different sensors and LEDs.

After the overview, this lesson goes through the control-flow of an Arduino, how the program gets converted to text on the screen to something actually running on the board.

We also learn how to write to the outputs on the board, in particular the RGB LED.

Module outline

Students are introduced to the concept of an embedded system, and how almost every interaction students have with technology is through embedded systems.

In the first activity, students learn how to write a simple program for the Arduino, through setting the LED. They learn how we set multiple colours, how the primary colours add together to form other colours, and how our eyes perceive colour.

The middle section focuses on syntax and how the Arduino system runs. We learn what a block of code is, and how to import code from a *library*. Students learn how to write notes in code through comments, and how the syntax of a programming language is important, just like natural languages.

In the final section, students are introduced to controlling the timing of embedded systems, and further explore how colours are made. They are led to think about how eyes see colour (additive), contrasted with how printers create colour (subtractive).

Guiding Question

How do we make a computer do exactly what we want?
How can we input and output information with a computer?
How are colours created?
How do we control the timing of devices?

Elements

Representations
Sequencing
User / Environmental Input
Output
Timing
Colour

Funded by the Australian Government Department of Education and Training.

Purpose/Hook - Embedded devices are everywhere!

Video:
Make: Introduction to Arduino: https://www.youtube.com/watch?v=CqrQmQqpHXc

This video is quite well-produced about the Arduino environment and gives people an idea of what kind of projects are possible with the Arduino. Since electronic devices are permeating a large portion of our lives, there are infinite possibilities of things that we can make to interact with specific devices, or create something to improve our daily lives.

Students should be guided through a class discussion on where embedded devices are used every day, as well as what kinds of needs they have that could be solved using an embedded device.

Discussion:
Where are these microcontrollers used in our everyday lives? What objects in the classroom would use one? What kind of things would students like to make with the Arduino?
Examples include:

- Temperature/humidity logging: https://create.arduino.cc/projecthub/Guitarman1/temperature-and-humidity-data-logger-5e587e
- Detect when an object is close to you: http://www.instructables.com/id/Arduino-Distance-Detector-with-a-Buzzer-and-LEDs/
- Robot ESKY: https://www.hackster.io/hackerhouse/make-an-autonomous-follow-me-cooler-7ca8bc
- Self-balancing robot: https://diyhacking.com/build-arduino-self-balancing-robot/

Structure of the lesson:

Types of component:

Discussion	**Worksheet**	**Plugged Activity**	
Group Activity	**Unplugged Activity**	**Video**	
Animation	**Reflection**	**Game**	**App**

Funded by the Australian Government Department of Education and Training.

Lesson Components:

Why do we need to learn programming?

Discussion: As described above

Step 1: Getting Started

Plugged Activity
In the first activity, students write set the colour of the LED, and are introduced to how the Arduino system runs

https://groklearning.com/learn/aca-dt-78-ar-sound/1/0/

Step 2: Arduino Syntax

Plugged Activity
In this activity, students learn how syntax is important and how common errors occur

Start at Module 1, slide 4
https://groklearning.com/learn/aca-dt-78-ar-sound/1/7/

Step 3: Arduino Functions

Plugged Activity
In this activity, students are introduced to controlling the timing of the Arduino system, and performing multiple operations in the one program

Start at module 1, slide 8
https://groklearning.com/learn/aca-dt-78-ar-sound/1/11/

Review:

Variables are encountered everywhere

Group Activity
What kind of things can we control with the Arduino? Is it just blinking lights?

Discuss how the Arduino Esplora is different to a regular Arduino. The Esplora is purpose-built with lots of the sensors wired in already, but has limited scope to attach external inputs and output (the 4 IO tabs on the top can achieve this).

Why we're using the Arduino Esplora instead of a regular Arduino? The Esplora doesn't require any additional components, so we can access sensors and get a genuine idea of what it's like to interface with additional components without the need to do any electronics. We're learning the *programming* of electronics, without the need for wiring.

Funded by the Australian Government Department of Education and Training.

Resources and Links

Printables

Digital Resources

Make: Introduction to Arduino: https://www.youtube.com/watch?v=CqrQmQqpHXc

Feedback Link: ⭐⭐⭐⭐⭐

Funded by the Australian Government Department of Education and Training.

DT Challenge Year 7/8 Arduino – Sound

Module 2: Using Sensors
https://groklearning.com/learn/aca-dt-78-ar-sound/2/0/

Previous: Hello, Arduino

Next: Making Decisions

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as an integer and a string
Specification, algorithms, implementation	Simple algorithms, user input
Digital Systems	
Interactions	Interaction (user input, command line output
Impact	

Objectives (Content Descriptions)

ACTDIK024	Investigate how digital systems represent text data
ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints
ACTDIP030	Implement and modify programs with user interfaces involving iteration in a general-purpose programming language

Review of:

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems

What are we learning? (Abstract)

In the previous lesson we learned about writing to the Esplora's outputs. In this lesson we are reading from sensors, or gathering inputs. Students learn that sensors just give us a *number*, and we can save that number into a *variable* and use it for a range of things.

Funded by the Australian Government Department of Education and Training.

Even though sensors detect different things, the result is still a number, and students have to interpret that number in order to make sense of the data.

Students also learn how different types of data, text and numbers, are handled differently in a program.

Module outline

Students are introduced to the concept of a variable, which can save different types of information. There are different types of variables, local and global which depend on where they are defined.

Students are asked to take information from a variable and are shown they can output (print) it to the Serial port, and also can write that number to an output on the board (like the speaker or LED).

Guiding Question

How do we observe inputs and control the outputs?
How do we store information and use that information later on?
How can we control the timing of devices?

Elements

Input
Output
Timing
Variables
Colour

Purpose/Hook - Storing Data!

Worksheet: Data types
(see at the end of this document)

Discussion:
How do we store information? Are variables visible everywhere in a program, or can they be hidden from other parts? How do we know what type of information is stored?

How do computer collect information? How do they present information back to the users?

By understanding that computers reads information from sensors as numbers, and the output are also numbers, students will understand the power of simple mathematics operations to perform tasks.

Funded by the Australian Government Department of Education and Training.

Structure of the lesson:

Types of component:

Discussion **Worksheet** **Plugged Activity**

Group Activity **Unplugged Activity** **Video**

Animation **Reflection** **Game** **App**

Lesson Components:

Step 1: Variables

Plugged Activity

The first activity describes storing variables as different *types* of data, but focuses specifically on *integers.* It looks at variable *scope*, which means where the variables are visible in the program, and how global variables can be changed in each iteration to keep track of the program operation.

https://groklearning.com/learn/aca-dt-78-ar-sound/2/0/

Step 2: Serial

Plugged Activity

This activity focuses on printing *different types* of information to the user, and how we print *multiple types*, also how we present information to have greater meaning.

https://groklearning.com/learn/aca-dt-78-ar-sound/2/7/

Step 3: Sensors

Plugged Activity

The final activity focuses on *gathering information* from sensors, which is a number, and *outputting* that information via the user, and the terminal. But they are both the different ways of representing the *same information.*

https://groklearning.com/learn/aca-dt-78-ar-sound/2/14/

Funded by the Australian Government Department of Education and Training.

Review:

Variables are encountered everywhere

Group Activity

Students should discuss how we keep track of information globally within a program, and how the *same information* can be printed to the screen or printed to the LED.

Resources and Links

Feedback Link: ★★★★★

Funded by the Australian Government Department of Education and Training.

Data Types Worksheet

Background on data types:

A data type is a kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

Just like there are different kinds of animals in biology, there do exist different types of data in computing. Each of them has a particular purpose and can store different things.

Almost all programming languages include the notion of data type, though different languages may use different terminology. Some of the common data types include:

String: Can store anything, such as '18 Sunshine Road, 1234 Moon Colony
Integer: Can be a whole number, such as 12345
Boolean: Can be a Yes/No decision, often expressed as 'true' or 'false'
Float: Can be any number, such as 1.23478

Select which data types you would use to store the following information

1) An entry in a phone book
2) Your friend's mobile phone number
3) A shopping list
4) The result of rolling a dice
5) The result of 3 divided by 2
6) The decision of going to the cinema tonight, or not.
7) Your favourite colour
8) The number of people in the world
9) The number of relatives a person has
10) The height (in metres) of the Eiffel Tower in Paris

Funded by the Australian Government Department of Education and Training.

Solutions

1) String, because a phone book entry consists of a name and a number
2) String, because of the international prefix (+61) or because of leading zero, such as 02. An Integer would not be able to represent the '+' or the leading zero.
3) String
4) Integer, because a dice will only produce whole numbers
5) Float
6) Boolean, because it is sufficient to store yes or no
7) String, because a colour has a name
8) Integer
9) Integer
10) Integer

Funded by the Australian Government Department of Education and Training.

DT Challenge Year 7/8 Arduino – Sound

Module 3: Making Decisions
https://groklearning.com/learn/aca-dt-78-ar-sound/3/0/

Previous: Using Sensors
Next: Using Numbers

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as an integer and a string
Specification, algorithms, implementation	Simple Algorithms, user input, decisions, branching, comparison operators
Digital Systems	
Interactions	Interaction (user input, command line output)
Impact	

Objectives (Content Descriptions)

ACTDIP024	Investigate how digital systems represent text, image and audio data in binary
ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints
ACTDIP030	Implement and modify programs with user interfaces involving branching and iteration in a general-purpose programming language

What are we learning? (Abstract)

In the previous lesson we learned about how to gather information and present it in different ways. This lesson we're understanding how programs can change and make decisions based on information gathered from external sources, either from the use or by measuring the environment.

We look at how we *compare* two sides, whether they are equal (==), not equal (!=), greater than (>) or less than (<), and how we use those different types of operations. But realising we're still just dealing with *numbers*. Students then learn the *else* statement, which captures when the condition is not satisfied.

We learn how the Arduino sets a *state* and that there are multiple different states possible for one type of decisions.

Funded by the Australian Government Department of Education and Training.

Module outline

Students are introduced to the concept of a variable, which can save different types of information. There are different types of variables, local and global which depend on where they are defined.

Students are asked to take information from a variable and are shown they can output (print) it to the Serial port, and also can write that number to an output on the board (like the speaker or LED).

Guiding Question

How do we get computers to make simple decision?
What types of decisions can we make?
How can more complex decisions be made?

Elements

Input
Output
Branching
Representations
Types of data
Sequencing
User / Environmental Input
Designing Algorithms
Variables
Inequalities

Purpose/Hook - Storing Data!

Discussion:
What type of decisions did you make today? Did you put on sunglasses because it was sunny? Did you put on a jacket because it was cold? Did you wear a helmet because you rode a bike? We make decisions all the time!

Funded by the Australian Government Department of Education and Training.

Structure of the lesson:

Types of component:

Discussion **Worksheet** **Plugged Activity**

Group Activity **Unplugged Activity** **Video**

Animation **Reflection** **Game** **App**

Lesson Components:

Step 1: The If Statement

Plugged Activity

This activity introduces if statements, in the form of performing an action when a button is pushed. We also learn that we can add multiple if statements after each other to make more decisions

https://groklearning.com/learn/aca-dt-78-ar-sound/3/0/

Step 2: Else Statements

Plugged Activity

With the else statement, students learn how to branching works when a condition is *not* true.

https://groklearning.com/learn/aca-dt-78-ar-sound/3/6/

Step 3: Conditions

Plugged Activity

In conditions, we learn the types of conditions that can be evaluation. Equals (==), not equal, (!=), less than (<), greater than (>), and how we compare numbers to make decisions, regardless of where the number originates from.

https://groklearning.com/learn/aca-dt-78-ar-sound/3/9/

Step 4: Multiple Statements

Plugged Activity

Funded by the Australian Government Department of Education and Training.

The final activity focuses on extending the *if* statements to add multiple branches. Students learn the else-if block to check another condition only if the original condition was not satisfied. This is when multiple possible checks lead to a different outcome.

Students should realise the practical outcomes with the temperature monitor, and be encouraged to think of other devices that could be made using if statements.

https://groklearning.com/learn/aca-dt-78-ar-sound/3/15/

Review:

Variables are encountered everywhere

 **Group Activity**

Students should discuss the applications from their everyday lives with technology as to how decisions are made. Are the keys their pressing on a keyboard being checked by an *if-statement*?
What kind of devices that we use in the real world could we make with just that one thing?
Keyboard?
Remote control?
Motion sensor?

Resources and Links

Digital Resources

Hour of Code - Bill Gates explains If statements:
https://www.youtube.com/watch?v=m2Ux2PnJe6E

Feedback Link: ★★★★★

Funded by the Australian Government Department of Education and Training.

DT Challenge Year 7/8 Arduino – Sound

Module 4: Using Numbers
https://groklearning.com/learn/aca-dt-78-ar-sound/4/0/

Previous: Making Decisions
Next: Looping

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as an integer and a string
Specification, algorithms, implementation	Simple Algorithms, user input
Digital Systems	
Interactions	Interaction (user input, command line output
Impact	

Objectives (Content Descriptions)

ACTDIP024	Investigate how digital systems represent text, image and audio data in binary
ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints
ACTDIP030	Implement and modify programs with user interfaces involving branching and iteration in a general-purpose programming language

What are we learning? (Abstract)

In this lesson, we are learning about how to use some of the arithmetic operations (+, -, *, /), and how the order of operations affects calculations, just like in maths.

We extend decision making to incorporate logical statements and (&&) and or (||) to make more complex decisions, and how we can add more than one condition to an action.

The last section looks at how calculations are performed on different types of data.

Module outline

In this module, students are introduced to arithmetic operations and performing calculations. They also learn how numbers can be used not just to create light, but also produce sound.

Students learn how to make more complex decisions with the AND (&&) and OR (||) statement, where they combine an ambient light and surrounding sound to create a security alarm and using the OR statement to determine when an object has moved in either direction.

Finally, students do unit conversions and learn how to calculate percentages.

Guiding Question

How do we perform calculations on a computer?
Does it make a difference when we do calculations on *different types* of data?
How can we make more complex decisions?

Elements

Input
Output
Sound
Logic
Calculations

Purpose/Hook - Storing Data!

Discussion:

What happens if we ....
... want different conditions to perform the same action?
... convert a number to something more meaningful?
... multiply a value to produce the desired output?
We can learn all of these things by manipulating numbers.

Funded by the Australian Government Department of Education and Training.

Structure of the lesson:

Types of component:

Discussion **Worksheet** **Plugged Activity**

Group Activity **Unplugged Activity** **Video**

Animation **Reflection** **Game** **App**

Lesson Components:

Step 1: Making Music

Plugged Activity

This lesson we learn how to play tones on the Arduino Esplora, and how performing a calculation on an input affects the output, which we can hear from the speaker.

https://groklearning.com/learn/aca-dt-78-ar-sound/4/0/

Step 2: Logic

Plugged Activity

This lesson introduces logical operators AND (&&) and OR (||), and shows how we can use them to make more complex decisions on numbers.
https://groklearning.com/learn/aca-dt-78-ar-sound/4/6/

Step 3: Floats

Plugged Activity

In floats, we use the **float** data type, and how computers store information on different types. We investigate how calculations are performed on a computer, in terms of converting units and calculating percentage. We also see how *information is lost* when the calculations are not done correctly.

https://groklearning.com/learn/aca-dt-78-ar-sound/4/13/

Funded by the Australian Government Department of Education and Training.

Review:

Variables are encountered everywhere

Group Activity

Students should think of examples of where calculations are performed, and what goes into making those calculations behave correctly. Doing temperature conversion, what other conversions can be made?

Students should brainstorm some examples where multiple decisions need to be made?

- When you use a game controller
- When an air conditioner is on cooling/heating mode
- When a microwave is already on
- When an elevator is moving up or down

Resources and Links

Feedback Link: ★★★★★

Funded by the Australian Government Department of Education and Training.

DT Challenge Year 7/8 Arduino – Sound

Module 5: Looping
https://groklearning.com/learn/aca-dt-78-ar-sound/5/0/

Previous: Using Numbers Next: Arrays

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as an integer and a string
Specification, algorithms, implementation	Simple Algorithms, user input, loops
Digital Systems	
Interactions	Interaction (user input, command line output)
Impact	

Objectives (Content Descriptions)

ACTDIP024	Investigate how digital systems represent text, image and audio data in binary
ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints
ACTDIP030	Implement and modify programs with user interfaces involving branching and iteration in a general-purpose programming language

What are we learning? (Abstract)

Looping means repeating the same, or very similar, thing multiple times. This course focuses on the *for* loop. The while loop is taught in the Year 7 Python - Chatbot course.
We've been doing looping already, as it is inbuilt into how the Arduino runs. So a lot of the solution to these problems could actually be done without an additional loop.

Module outline

This module introduces *for loops*, which is the ability to repeat a section of code for a fixed number of times. We take the example of fading up the LED, we know there are 256 brightness levels, so

Funded by the Australian Government Department of Education and Training.

to go through each one we can loop a fixed number of times. We use the variable, *i*, to keep track of the iteration and to set the brightness level using that variable.

We demonstrate that it's possible to loop down from a fixed value, down to zero. We demonstrate this by fading *down* the LED.

We also introduce compound operators, to modify the *i* variable by an amount greater than 0, and start the loop at any number.

Guiding Question

Why do we need to repeat things?
How can we repeat things a given number of times?

Elements

Input
Output
Sound
Logic
Calculations
Iteration
Absolute values
Indexing

Purpose/Hook – Repeating Things!

Discussion:
Computers are really good at doing boring, repetitive tasks. To make things simpler, we can loop a given number of times.
What are some examples where we want to repeat a task?

- Sending "Happy Birthday!" message to everyone in your class?
- Chiming a clock every hour

Structure of the lesson:

Types of component:

Discussion **Worksheet** **Plugged Activity**

Group Activity **Unplugged Activity** **Video**

Animation **Reflection** **Game** **App**

Funded by the Australian Government Department of Education and Training.

Lesson Components:

Why do we need to repeat things?

Discussion: As described above

Step 1: Loops

Plugged Activity
This section introduces the for loop, by demonstrating how to fade up the LED just with a couple of lines of code.

https://groklearning.com/learn/aca-dt-78-ar-sound/5/0/

Step 2: Absolutely!

Plugged Activity
This section introduces the *abs* function, which removes the negative sign of a number.

https://groklearning.com/learn/aca-dt-78-ar-sound/5/7/

Step 3: Looping the other way

Plugged Activity
In this section we learn how to make loops go *down*, and make a siren by looping both ways!

https://groklearning.com/learn/aca-dt-78-ar-sound/5/9/

Review:

Loops can be used to repeat things easily!

Group Activity
Students should come up with other ideas of where they might use loops, and recognise that we've been using loops this whole time within the Esplora loop()-function! Whenever we want to check if a button has been pushed, we need to check continually if that has happened!

Feedback Link: ★★★★★

Funded by the Australian Government Department of Education and Training.

DT Challenge Year 7/8 Arduino – Sound

Module 6: Arrays
https://groklearning.com/learn/aca-dt-78-ar-sound/6/0/

Previous: Looping Next: Project

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as an integer and a string
Specification, algorithms, implementation	Simple Algorithms, user input
Digital Systems	
Interactions	Interaction (user input, command line output
Impact	

Objectives (Content Descriptions)

ACTDIP024	Investigate how digital systems represent text, image and audio data in binary
ACTDIP027	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints
ACTDIP030	Implement and modify programs with user interfaces involving branching and iteration in a general-purpose programming language

What are we learning? (Abstract)

This is about using arrays, which are more complicated data structures that are used to any given number of integers (or any data type) in a single variable. We learn how to write to, and read from, an array by accessing the *index* of the element that we would like to change.

We learn about how arrays can be accessed using *for* loops.

Module outline

The module starts by introducing the purpose of arrays, to store collections of information in a much easier format. The example given is storing notes, it doesn't make sense to store a large

number of different variables, so we create an array to store a number of notes. Arrays can store any number of values (within memory constraints) of the same type. We can access each element in the array by using the index. The index starts at 0 and stops at the number of elements – 1. This is the same as strings from the Python – Chatbot course.

We then use loops, which were introduced in the previous section, to access arrays. We use the iteration number from the for loop to access each element. There are many ways to use loops with arrays, finally we look at playing a sequence of tones stored in an array.

Guiding Questions

How can we store groups of information more easily?
How can we access that information?

Elements

User Input
Output
Indexing
Iteration
Data structures
Data types
Frequencies
Linear conversion
Calculations

Purpose/Hook – Storing Lots of Data!

Discussion:
What are examples of collections of information we might want to store?

It could be a sequence of dates, storing the last positions of the joystick, the number of items in a set of drawers.

Structure of the lesson:

Types of component:

Discussion **Worksheet** **Plugged Activity**

Group Activity **Unplugged Activity** **Video**

Animation **Reflection** **Game** **App**

Funded by the Australian Government Department of Education and Training.

Lesson Components:

Step 1: Arrays

Plugged Activity
This activity introduces arrays and how we access each element.

https://groklearning.com/learn/aca-dt-78-ar-sound/6/0/

Step 2: Choosing Colours

Plugged Activity
This lesson shows how we can select elements from the array by using the buttons.

https://groklearning.com/learn/aca-dt-78-ar-sound/6/6/

Step 3: The Map Function

Plugged Activity
In this lesson we convert values from any range of values to access elements in an array using the map function.

https://groklearning.com/learn/aca-dt-78-ar-sound/6/10/

Review:

Group Activity

Resources and Links

Arduino Arrays - https://www.arduino.cc/en/Reference/Array

`Feedback Link: ★★★★★`

Funded by the Australian Government Department of Education and Training.