Stage 4 Technology Mandatory – Virtual crack the code

## Summary

In this unit students will develop a fundamental understanding of the Arduino microcontroller and the basic range of inputs and outputs that can be used with it.

## Duration

4 lessons

## Outcomes

A student:

* **TE4-1DP** designs, communicates and evaluates innovative ideas and creative solutions to authentic problems or opportunities
* **TE4-2DP** plans and manages the production of designed solutions
* **TE4-4DP** designs algorithms for digital solutions and implements them in a general-purpose programming language
* **TE4-10TS** explains how people in technology related professions contribute to society now and into the future

[Technology Mandatory Years 7-8](https://www.educationstandards.nsw.edu.au/wps/portal/nesa/k-10/learning-areas/technologies/technology-mandatory-7-8-new-syllabus) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2017

## Unit Overview

Using Tinkercad and the Arduino IDE, students are required to complete a series of plug run play (PRP) activities to develop basic programming skills. Each practical activity is designed to extend upon the previous activity and enhance the student’s skills with a range of inputs and outputs that can be used with an Arduino UNO R3 microcontroller.

## Resources overview

The resources and links listed below are referenced within the teaching program but are not an exhaustive list of resources available. Teachers can add to these resources as needed.

### Physical resources

* Computer with internet access
* Student workbook
* Optional: computer mouse
* Optional: Arduino UNO R3 or equivalent and basic electronics components (light emitting diodes (LEDs), light dependant resistors (LDRs), potentiometers, buttons, buzzers and resistors (10kΩ, and 220Ω).

### Websites

* Tinkercad circuits – [www.tinkercad.com/circuits](https://www.tinkercad.com/circuits)
* Optional: Arduino IDE download – [www.arduino.cc/en/main/software](https://www.arduino.cc/en/main/software)
* Optional: Arduino IDE web application – [create.arduino.cc/editor](https://create.arduino.cc/editor) (note: to use the web application you need to create an Arduino account)
* Binary translator - [www.binarytranslator.com](https://www.binarytranslator.com/)
* YouTube clips:
  + Department of Education – Learning Systems – Crack the Code playlist [www.youtube.com/watch?v=Z6f4P9Blkic&list=PL4OaBCdO34bBcyOm-pS5KE8oYuap\_22T6](https://www.youtube.com/watch?v=Z6f4P9Blkic&list=PL4OaBCdO34bBcyOm-pS5KE8oYuap_22T6)

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Teaching and learning** | **Evidence of learning** | **Adjustments and registration** |
| **Lesson 1** | **Introduction to Arduino.**  Students:   * Watch [Introduction to the Arduino UNO and the MAAS ThinkerShield](https://youtu.be/Z6f4P9Blkic) (duration 3:24). * Label the parts of the Arduino UNO in the student workbook. * Watch [Connecting to the Arduino](https://youtu.be/CRRSUmkse9A) (duration 3:37). * Label the parts of the Arduino IDE in the student workbook. | * Students identify and articulate the function of the main parts of the Arduino UNO. |  |
| * design algorithms that use a range of data types, branching and iteration and represent them diagrammatically and in English * trace algorithms to predict output for a given input and to identify errors | **Basics of Arduino**  Students:   * Login to Tinkercad or create an account using department of education credentials (the student workbook provides detailed instructions). * Open blink circuit in Tinkercad, switch to text-based code and test the code. Make observations of what happens on page 7 in the student workbook. * Watch [Tracing the 01\_Blink sketch](https://youtu.be/FW3XJrnB35Y) (duration 8:30) and answer the questions in workbook. * Edit the blink code in Tinkercad and make the LED turn on for 2 seconds and then off for 2 seconds. * In the student workbook record how they changed the code, and make predictions about how they could edit the code if the LED was instead connected to pin 8. * Attach an additional LED according to student workbook and then edit their code to make both LEDs flash at the same time. * Complete the traffic light pseudocode in the student workbook. * Attach an additional LED and attempt the traffic light challenge in the student. | * Students demonstrate understanding by describing the function of each line of code in the blink sketch and completing activities in the workbook. * Students can edit the blink sketch to modify the function intentionally. * Students can use pseudocode as a method of representing computer algorithms and refer to them when designing and modifying codes. |  |
| **Lesson 2:**   * design algorithms that use a range of data types, branching and iteration and represent them diagrammatically and in English * trace algorithms to predict output for a given input and to identify errors * implement and modify programs involving branching, iteration and functions in a general-purpose programming language | **Digital inputs.**  Students:   * Open the button circuit in Tinkercad, switch text based code and test the code. Make observations of what happens in the student workbook. * Watch [The if statement](https://youtu.be/oAlExniCQzg) (duration 9:18) and answer the questions in the student workbook. * Open the button circuit in Tinkercad and adjust the code according to page 16 of the student workbook, and record observations. * Complete button challenge in the student workbook, to find an alternative way to reverse the action of the button. * Complete the button challenge using the flowchart of the button sketch to inform the solution. | * Students identify similarities between the blink sketch and the button sketch. * Students articulate the purpose of using ‘if’ statements in programming to enable microprocessors to make decisions based on data from sensors for example. * Students successfully edit the button sketch in multiple ways to achieve the same outcome of reversing the action of the button. * Students use flowcharts as a method of representing computer algorithms and refer to them when designing and modifying codes. |  |
| **Lesson 3:**   * investigate how digital systems represent text, image and audio with whole numbers * explain how and why whole numbers are represented in binary in digital systems * explore how data is transmitted and secured in wired, wireless and mobile networks | **Introduction to binary.**  Students:   * Watch [An Introduction to Binary](https://www.youtube.com/watch?v=b82kHMdEM4g) (duration 6:34) and answer the questions in the student workbook. * Use a [binary translator](http://www.binarytranslator.com) to copy the binary code from the front of the student workbook and decode it. * Complete the activity of creating a short message and converting it to binary in the student workbook. | * Students demonstrate how whole numbers can be represented in binary. * Students articulate why binary is used in digital systems. * Students show understanding of how binary can be used to represent data other than numbers, for example letters. |  |
| **Lesson 4:**   * design algorithms that use a range of data types, branching and iteration and represent them diagrammatically and in English * trace algorithms to predict output for a given input and to identify errors * implement and modify programs involving branching, iteration and functions in a general-purpose programming language | **Analogue inputs.**  Students:   * Watch [Analogue inputs](https://youtu.be/3mocouv9WQY) (duration 9:19) and complete the questions in the student workbook. * Open the analog input circuit in Tinkercad, switch to text based code and test the code. Make observations of what happens in the student workbook. * Edit the code according to the student workbook and use the serial monitor to observe the value of the sensor being read in real-time. * Complete extension activity to combine the analog input sketch and if statement used in the button sketch to create a light that switches when the potentiometer is read to have a value that is below 500. | * Students articulate the difference between digital and analogue sensors. * Students show understanding of applications for digital and analogue sensors. * Students demonstrate when to use digital and analogue functions in the analog input sketch. |  |

## Evaluation

Evaluation of learning activities should be an ongoing process that happens throughout the delivery of this unit. Teachers should document their evaluation of learning activities throughout the program. The space provided below is to evaluate the overall unit of work.

|  |
| --- |