**Australian Curriculum:  
Digital Technologies**

**Years 9–10**

**Sample assessment task**

**Measuring and tracking environmental factors in the school**

**Assessment focus:** Australian Curriculum:Digital Technologies

(digital systems and data)

# About this assessment task

This sample assessment task comprises a series of sample assessment activities. These activities and related sample assessment tasks have been prepared to assist teachers with the implementation of the Australian Curriculum: Digital Technologies, with a focus on digital systems and data.

These show how aspects of the Digital Technologies curriculum related to digital systems and data may be assessed using contexts from other learning areas and subjects. These contexts could be content that students have recently completed or are learning concurrently. This approach should enhance the manageability of the curriculum while still providing a targeted focus on Digital Technologies.

## Purpose

The sample activities aim to:

* demonstrate meaningful curriculum links to:
* Digital Technologies curriculum:
  + achievement standard
  + content descriptions
  + content strands
  + key concepts
  + key ideas (Technologies)
* general capabilities
* cross-curriculum priorities
* other learning areas. See Appendix 1 for specific links for this task.
* provide teacher support materials, suggested adjustments for students with diverse needs and resources. See Appendix 2.
* provide a template to create your own assessment task. See Appendix 3.
* provide background information. See Appendices 4–9.
* provide student activities with detailed instructions. See *Investigating environmental data with microcontrollers activity guide.*

## How to use these sample activities

These sample activities can be implemented as standalone tasks or could be used to inform planning of a:

* unit of work that might accompany the sample activities
* similar activities and/or units of work with a focus on digital systems and data.

Two types of assessment are indicated here: formative and summative. (For extended information see Appendix 2).

### Formative examples

These examples could be used to gain an insight into students’ understandings.

* Remove commenting from a provided program and ask for each line to be commented, inline within the program code.
* Give students a program that has statements that are misspelt, incorrectly capitalised or in the wrong order, and use the integrated development environment (IDE) to detect and correct.
* Use a [Parson’s problems](https://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-88.pdf) approach to test understanding of program flow and logic.
* Ask students to determine what correlation exists between temperature and humidity when measured using an ESP01 microcontroller (ie expand on Activity 3).
* Ask students to code an inexpensive (ESP01) microcontroller, take home or to a location where free, non-authenticated wi-fi exists and send environmental measurements to the cloud.

### Summative examples

* Ask students to write an explanatory text that may, for example explain the function of an IDE, and list the things that could confuse a user (especially first-time users) in using it.
* Create a report which documents students’ findings and proposed solutions to various environmental conditions in their school environments.
* Design a prototype system for a manufacturer who needs to maintain environmental factors such as temperature and humidity.

# Title: Measuring and tracking environmental factors in the school

**Assessment focus:** Australian Curriculum: Digital Technologies (Digital systems and data). This task is also linked to Science. Depending on modifications made to this task, opportunities exist to link this task to Mathematics and/or English.

**Band:** Years 9 and 10

**Context:** Digital Technologies/Science

**Duration:** Dependent on how the task is to be implemented – up to 1 term unit

**Prior learning:** Students will have:

* had some exposure to coding in a general-purpose programming language such as JavaScript, Python or C++ (Arduino)
* used some physical devices (for example, micro:bits or Arduinos) in combination with computer hardware such as laptops or desktop computers
* had exposure to data analysis tools such as a spreadsheet or other data visualisation applications.

## Task summary

**Key inquiry question:**

How can we determine if our school environment is at a level conducive to optimal learning?

**Focus questions:**

Depending on the depth that the teacher chooses, the focus questions could be:

* How can we measure and track environmental factors in our school?
* Could we give warnings when metrics such as low light or high temperatures are evident?
* How could we design a system to manage environmental variables for a manufacturer?

Discuss with students how environmental variation in conditions such as ultraviolet light, carbon dioxide (CO2) and particulates in the air can affect health. There is a body of evidence indicating that environmental conditions including noise level, lighting level, temperature level, CO2 level, particulate level and so on can affect our health (look at background information – teacher guidance and support below for links to relevant research).

Students will undertake activities that will allow them to measure an environment for some of the factors listed above. By creating and coding digital systems, students will collect data as evidence and determine ways to make their environment more optimal for learning or to control a manufacturing environment.

Students will use their knowledge of digital systems to create monitoring systems that alert the users when conditions fall outside acceptable levels. This digital system together with the students’ planned solutions may be presented in some format to key stakeholders.

**Students will:**

* research optimal levels of a variety of environmental factors
* create digital systems which monitor environmental factors and provide relevant data
* create mechanisms within the systems that provide feedback based on the environmental condition/s being monitored
* design and generate algorithms that can be used to implement a suitable digital system
* implement digital systems using a general-purpose programming language
* determine thresholds for the environmental data used by the digital systems and have the systems alert the users in some way when thresholds have been crossed
* identify solutions and put procedures in place to remedy undesirable environmental conditions
* document findings and solutions in a format of their choice
* use a variety of networking facilities such as I2C, serial, http, API and wi-fi
* design and create algorithms that explain the computational thinking and logic of their created digital systems
* determine thresholds for the digital systems and have the systems alert the users in some way when thresholds have been crossed.

### Task features

Students will be asked to complete the following activities. (Initial activities can be completed as a whole class. As students progress, subsequent activities can be completed in small groups or individually depending on the amount of equipment available.)

While these activities can be done using a variety of microcontrollers, this document will concentrate on using the ESP8266 microcontroller family only.

* Identify environmental factors that can be undesirable.
* Experiment with ESP32, ESP8266 or ESP01S microcontrollers to collect data using external sensors.
* Calibrate measurements by comparing to other, known measuring tools.
* Visualise collected time-series data.
* Incorporate digital procedures to provide feedback regarding environmental conditions.
* Create a report which documents the students’ findings and proposed solutions, possibly presenting the completed report to stakeholders such as a manufacturer, school staff, other classes, a parent body or a school executive.

# Background information

## Teacher guidance and support

* Collect the necessary resources – this task needs a microcontroller and some peripherals. In effect, you could just choose the activities that suit the equipment you have.
* Skim read through the activity guide.
* Determine the entry point for your students. The tasks are graded from beginner to advanced and your students’ current level of understanding should inform this decision. This is up to a term of lessons, depending on where you start. Students with more experience may want to start later in the sequence of tasks, and some students will finish a task earlier than others, again, depending on experience.
* Consider presenting each of the activities in the order that they are given here. However, as they are designed to build on knowledge gained in the previous activity, use your knowledge of your students’ abilities and experience to decide where you will start.

The suggested assessments are merely that – you may decide to assess your students in   
other ways.

### Suggested introductory activity

Use the ACARA computational thinking poster as a stimulus to identify the aspects of computational thinking involved in this activity. See:

<https://www.australiancurriculum.edu.au/media/5013/computational-thinking_poster_v3.pdf>

#### Decomposition, algorithms and data

Where relevant, each activity is decomposed and expressed as structured English (pseudocode).

Structured English is a very handy term as the word ‘structured’ refers to data as well as algorithms. We need structure in our language – the order and placement of words in a sentence influence meaning. Consider: ‘The man killed the snake’ compared with ‘The snake killed the man’ and further examples in the following tables.

|  |  |
| --- | --- |
| **Statement** | **Meaning** |
| Only women dance at weddings. | men don’t dance at weddings |
| Women only dance at weddings. | women don’t eat or talk, only dance |
| Women dance at weddings only. | women don’t dance anywhere else |

|  |  |
| --- | --- |
| **Statement** | **Meaning** |
| Only men cook at barbecues. | women don’t cook at barbecues |
| Men only cook at barbecues. | men don’t talk, only cook |
| Men cook at barbecues only. | men don’t cook unless at a barbecue |

Similarly, data has an order: Wise Street is different from Street Wise.

Whereas English uses spaces to separate words, structured data has delimiters such as TAB characters (the basis of a TABle) or commas to indicate the different elements of data.

Look at the following 2 examples.

##### Using a table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit/building** | **Nº** | **street\_name** | **street\_type** | **town or village** | **state or region** | **postcode** | **country** |
| Unit 2 | 1 | Wise | Street | Smithtown | NSW | 2440 | Australia |

##### Using commas

Unit/building, Nº, street\_name, street\_type, town or village, state or region, postcode, country

Unit 2, 1, Wise, Street, Smithtown, NSW, 2440, Australia

Data structures in these activities will start with the simple case of inserting TAB characters into data streams so that the data can be copied from the serial monitor and pasted into a spreadsheet.

Activity 5 will introduce the array as an example of a data structure.

The algorithmic thinking to build on the activities here to construct an environmental monitor will be very similar to what we have already used.

#### Environmental variables and their effects

Several variables have been identified as having a debilitating effect on learning. All of these environmental factors don’t need to be measured for this task. Activities are designed to use a variety of sensors, and a simple measurement of temperature and humidity may suffice for a variety of situations. Some examples used in this series of activities are listed below, and specific items, including costs, are outlined in Activity 0.

#### Light levels

Measured in lux, low light levels can cause eyestrain, headaches and loss of concentration. See:

<https://www.edutopia.org/article/science-of-effective-learning-spaces-melina-uncapher>

#### Temperature

Temperatures can quickly climb in classrooms containing 25–30 humans each producing around 85 watts. See: <https://phys.libretexts.org/Bookshelves/College_Physics/Book%3A_College_Physics_(OpenStax)/07%3A_Work_Energy_and_Energy_Resources/7.08%3A_Work_Energy_and_Power_in_Humans>

#### Humidity

High humidity is something experienced in most classrooms, but low humidity can also affect mental activity. See: <https://link.springer.com/article/10.1007/s10111-015-0342-2>

#### Carbon dioxide levels

CO2 levels can play a major part in students’ abilities to learn. With as little as 1,000 parts per million and likely lower still, CO2 induces sleepiness, poor concentration, abnormal heart rates and even nausea, as expressed in an article about a study from the Harvard School of Public Health [www.tinyurl.com/yclw6kzu.](https://thinkprogress.org/exclusive-elevated-co2-levels-directly-affect-human-cognition-new-harvard-study-shows-2748e7378941/) Similarly, it appears that air pollution has an enormous effect on learning. A study reported on in *The Guardian* [www.tinyurl.com/y92t7yz9](https://tinyurl.com/y92t7yz9) suggests that high levels of urban pollution have a major impact on attainment, with some students dropping a whole year of progress over their school lives.

Of course, high levels of CO2 may be beneficial in other situations such as greenhouses where maximum photosynthetic output is desirable. Many CO2 sensors are on the market: inexpensive ones that estimate CO2 levels and expensive ones that accurately measure CO2 levels and may need calibration.

#### Air pressure

Air pressure may play a role in affecting cognitive abilities. This is under research; however, the common complaint of sinus headaches when air pressure changes will obviously affect one’s ability to learn. Think about how your students behave on a windy day.

#### Prompting further investigations

There are many other things you can do with microcontrollers. Students will often come up with great ideas once they are aware of what a device can do. A way to start the conversation is to say something such as: “If we know how to measure temperature, what other sorts of things could we measure that may influence our learning, safety or comfort in the classroom or in a manufacturing environment? What sensors are available?” This practice may lead to better formative or summative assessment vehicles than those proposed in this document.

# Links to the Australian Curriculum

Table 1 shows the related Australian Curriculum links to this task. For a more in-depth exploration of the links to the curriculum, see **Appendix 1**.

Table 1: Links from the task to the Australian Curriculum

|  |  |  |  |
| --- | --- | --- | --- |
| **Digital Technologies**  ***Achievement standard***  Aspects addressed by this task are highlighted. | By the end of Year 10, students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users. They explain simple data compression, and why content data are separated from presentation.  Students plan and manage digital projects using an iterative approach. They define and decompose complex problems in terms of functional and non-functional requirements. Students design and evaluate user experiences and algorithms. They design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and data entities. They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions. They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise. They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects. | | |
| ***Strands*** | Digital Technologies knowledge and understanding   * Digital systems   Digital Technologies processes and production skills   * Collecting, managing and analysing data * Creating digital solutions by: * investigating and defining * generating and designing * producing and implementing * evaluating | | |
| ***Content descriptions*** | * Investigate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems ([ACTDIK034](http://www.scootle.edu.au/ec/search?accContentId=ACTDIK034)) * Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements ([ACTDIP036](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP036)) * Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data ([ACTDIP037](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP037)) * Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs ([ACTDIP038](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP038)) * Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics ([ACTDIP039](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP039)) * Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases ([ACTDIP040](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP040)) * Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language ([ACTDIP041](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP041)) * Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability and provide opportunities for innovation and enterprise ([ACTDIP042](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP042)) * Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability ([ACTDIP044](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP044)) | | |
| ***Key concepts*** | * abstraction * data collection * data interpretation * specification * algorithms * implementation * digital systems * impact | ***Key ideas*** | Thinking in Technologies   * + systems thinking   + computational thinking   + design thinking |
| ***Cross-curriculum priorities*** | * Sustainability | ***General capabilities*** | * Information and Communication Technology (ICT) Capability * Literacy * Numeracy |

### Assessment planner

|  |  |
| --- | --- |
| **Achievement standard** (relevant aspect of the achievement standard  to be assessed) | **Student evidence** (what student evidence will be considered to judge if the achievement standard aspect has been met) |
| **Digital Technologies** | |
| Students explain the control and management of networked digital systems | * Communication between the microcontroller, its sensors and a cloud-based platform via wi-fi is successful |
| explain the … security implications of the interaction between hardware, software and users | * Wi-fi communications security, particularly firmware update over the air |
| plan and manage digital projects using an iterative approach | * Show, using Gantt chart or similar, how the understanding of a solution has been gained |
| They define and decompose complex problems | * Build a complex system by starting with simple measurements then moving to incorporate more sophisticated approaches including data structuring |
| design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and  data entities | * Complete activities 1 to 9 |
| Students test and predict results and implement digital solutions | * Build and test a solution and compare its output to that predicted |
| They take account of privacy and security requirements when selecting and validating data ... They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise | * Investigate risks involved in deploying IoT solutions |

### Assessment rubric

This rubric shows only Digital Technologies. **Note:** There are opportunities to include Science, Literacy and Numeracy in the assessment.

| **Relevant sections of the achievement standard** | **Below standard**  **Students:** | **At standard**  **Students:** | **Above standard**  **Students:** |
| --- | --- | --- | --- |
| **Digital systems**  Security considerations | * demonstrate knowledge of security issues that need to be taken into account | * identify specific security issues and describe methods to mitigate risk | * identify complex security issues related to code, bit firmware and access/ control, and outline methods to mitigate risk |
| **Data collection and interpretation**  Methods of acquiring and analysing data | * use a scatter plot to analyse data (see [ACMSP251](http://www.scootle.edu.au/ec/search?accContentId=ACMSP251)) by seeing that there is a relationship between a light dependent resistor reading and lux | * use a scatter plot to analyse data (see [ACMSP251](http://www.scootle.edu.au/ec/search?accContentId=ACMSP251)) by exploring various relationships such as power, log and exponent between 2 numerical variables | * use a scatter plot to analyse data (see [ACMSP251](http://www.scootle.edu.au/ec/search?accContentId=ACMSP251)) by using the equation provided by a spreadsheet that describes the relationships between 2 numerical variables |
| **Specification**  Specify design solution | * design a solution for a given problem | * design a solution for a given problem, taking into account user stories and technical requirements | * design a solution for a given problem, taking into account user stories, technical and security requirements |
| **Implementation**  Implement modular programming, with data structures | * successfully download a direct copy of code to a microcontroller | * successfully download a modified copy of code to a microcontroller | * successfully download a heavily customised copy of code to a microcontroller |
| **Implementation**  Show test using serial, uses stubs | * direct output of data to serial port | * copy and store data collected over a serial port | * copy and store data that has been structured and then collected over a serial port |
| **Interactions**  Compare user experience (UX) of ThingSpeak with TagoIO | * recognise that there are differences in UX between differing websites but are unable to suggest variations to accommodate the user | * recognise that there are differences in UX between differing websites and suggest how these help or hinder the user | * compare differences in UX between differing web platforms and justify a decision to use one rather than another |
| **Interactions**  Project management | * produce a time/action or finance plan that has limited connection to the design project | * produce a time/action and finance plan that has some appropriate aspects to the design project | * produce a time/action and finance plan that is appropriate and completed within timeframe |
| **Impact**  Critically evaluate solutions in the context of innovation and enterprise | * evaluate a simple solution as to its efficacy | * evaluate a simple solution as to its efficacy, taking into account known errors and deciding whether those errors will adversely influence the solution | * evaluate a simple solution as to its efficacy by critically analysing factors that will affect accuracy and how the solution will scale, and discussing its sustainability |

**Disclaimer:** ACARA does not endorse any product or make any representations as to the quality of such products. This resource is indicative only. Any product that uses material published on the ACARA website should not be taken to be affiliated with ACARA or have the sponsorship or approval of ACARA. It is up to each person to make their own assessment of the product, taking into account matters including the degree to which the materials align with the content descriptions and achievement standards of the Australian Curriculum. The Creative Commons licence BY 4.0 does not apply to any trademark-protected material.

All images in this resource are the author’s own or used with permission.

# Appendices

## Appendix 1

## Australian Curriculum links (in detail)

## Appendix 2

## Support materials

## Things to think about

*Assessment planning*

*Rich questions and discussion starters*

Students with diverse needs

## Resources

## Activity guide

## Glossary

## Bibliography

## Appendix 3

## Digital systems task planning template

## Appendix 4

## Background information on electronics

## Appendix 5

## Background information on breadboards

## Appendix 6

## Drawing circuit layouts

## Appendix 7

## Arduino tips

## Appendix 8

## Structure of object-oriented programming languages

## Appendix 9

## Serial and I2C communications

## Appendix 1

## Australian Curriculum links (in detail)

## Links to the Australian Curriculum

Digital Technologies

Achievement standard

By the end of Year 10, students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users. They explain simple data compression, and why content data are separated from presentation.

Students plan and manage digital projects using an iterative approach. They define and decompose complex problems in terms of functional and non-functional requirements. Students design and evaluate user experiences and algorithms. They design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and data entities. They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions. They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise. They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects.

Content descriptions

|  |
| --- |
| * Investigate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems ([ACTDIK034](http://www.scootle.edu.au/ec/search?accContentId=ACTDIK034)) * Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements ([ACTDIP036](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP036)) * Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data ([ACTDIP037](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP037)) * Define and decompose real- world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs ([ACTDIP038](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP038)) * Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics ([ACTDIP039](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP039)) * Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases ([ACTDIP040](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP040)) * Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language ([ACTDIP041](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP041)) * Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability and provide opportunities for innovation and enterprise ([ACTDIP042](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP042)) * Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability ([ACTDIP044](http://www.scootle.edu.au/ec/search?accContentId=ACTDIP044)) |

Content strands

|  |  |  |  |
| --- | --- | --- | --- |
| **Digital Technologies knowledge and understanding** | | **Digital Technologies processes and production skills** | |
| * Digital systems * Representation of data | X | Collecting, managing and analysing data  Creating digital solutions by:   * investigating and defining * generating and designing * producing and implementing * evaluating * collaborating and managing | X  X  X  X  X |

Links to the key ideas

|  |  |  |
| --- | --- | --- |
| **Creating preferred futures** | Students develop solutions to meet needs considering impacts on liveability, economic prosperity and environmental sustainability. | X |
| **Project management** | Students will develop skills to manage projects to successful completion through planning, organising and monitoring timelines, activities and the use of resources. |  |
| **Thinking in Technologies**   * Systems thinking | Systems thinking is a holistic approach to the identification and solving of problems where the focal points are treated as components of a system, and their interactions and interrelationships are analysed individually to see how they influence the functioning of the entire system. | X |
| * Design thinking | Design thinking involves the use of strategies for understanding design needs and opportunities, visualising and generating creative and innovative ideas, planning, and analysing and evaluating those ideas that best meet the criteria for success. | X |
| * Computational thinking | Computational thinking is a problem-solving method that is applied to create solutions that can be implemented using digital technologies. It involves integrating strategies, such as organising data logically, breaking down problems into parts, interpreting patterns and models and designing and implementing algorithms. | X |

Read more about the [key ideas in the Australian Curriculum: Technologies](https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/key-ideas/).

Links to the key concepts

The [key concepts](https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/)that underpin the Digital Technologies curriculum establish a way of thinking about problems, opportunities and information systems and provide a framework for knowledge and practice. (Colour coding is based on the [Australian Computing Academy scheme](https://aca.edu.au/#what-is-the-digital-technologies-curriculum).)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **abstraction** | underpins all content, particularly the content descriptions relating to the concepts of data representation; and specification; algorithms; and implementation | |
|  | **data collection** | (properties, sources and collection of data)   * *Students collect and analyse environmental data.* | X |
|  | **data representation** | (symbolism and separation) |  |
|  | **data interpretation** | (patterns and contexts)   * *Students draw conclusions from collected data.* | X |
|  | **specification** | (descriptions and techniques)   * *Students specify what data they will collect and what problems they hope to solve such as making a space optimal for its purpose.* | X |
|  | **algorithms** | (following and describing)   * *Students design algorithms to collect data with a microcontroller.* | X |
|  | **implementation** | (translating and programming)   * *Students implement and test algorithms to collect data with a microcontroller.* | X |
|  | **digital systems** | (hardware, software, and networks and the internet)   * *Students design digital systems that will measure environmental data.* | X |
|  | **interactions** | (people and digital systems, data and processes)   * *Students develop an understanding of the way systems can be used to help people (i.e. themselves and classmates).* * *Students design digital systems to be used by people to monitor environmental conditions to optimise learning.* | X |
|  | **impact** | (sustainability and empowerment)   * *Students develop an understanding of how systems can help us to live and work more sustainably.* * *Students develop an understanding of how they can design digital systems that can be used to help people.* | X |

## Cross-curriculum priorities [Read more…](https://www.australiancurriculum.edu.au/f-10-curriculum/cross-curriculum-priorities/)

|  |  |  |
| --- | --- | --- |
| **Aboriginal and Torres Strait Islander Histories and Cultures** | **Asia and Australia’s engagement with Asia** | **Sustainability** |
|  |  | X |

## General capabilities [Read more…](https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Literacy** | **Numeracy** | **ICT Capability** | **Critical and Creative Thinking** | **Ethical Understanding** | **Personal and Social capability** | **Intercultural Understanding** |
| X | X | X | X |  |  |  |

## Links to ICT Capability continuum: Level 6

Depending on the year level this activity is being used with, adjust content to the appropriate level;  
for example, Level 4 or 5.

|  |  |
| --- | --- |
| **Apply social and ethical protocols and practices when using ICT** | |
| identify and describe ethical dilemmas and consciously apply practices that protect intellectual property |  |
| use a range of strategies for securing and protecting information, assess the risks associated with online environments and establish appropriate security strategies and codes of conduct | X |
| independently apply appropriate strategies to protect rights, identity, privacy and emotional safety of others when using ICT, and discriminate between protocols suitable for different communication tools when collaborating with local and global communities |  |
| assess the impact of ICT in the workplace and in society, and speculate on its role in the future and how they can influence its use |  |
| **Investigating with ICT** | |
| select and use a range of ICT independently and collaboratively, analyse information to frame questions and plan search strategies or data generation | X |
| use advanced search tools and techniques or simulations and digital models to locate or generate precise data and information that supports the development of new understandings | X |
| develop and use criteria systematically to evaluate the quality, suitability and credibility of located data or information and sources | X |
| **Creating with ICT** | |
| select and use ICT to articulate ideas and concepts, and plan the development of complex solutions | X |
| Design, modify and manage complex digital solutions, or multimodal creative outputs or data transformations for a range of audiences and purposes | X |
| **Communicating with ICT** | |
| select and use a range of ICT tools efficiently and safely to share and exchange information, and to collaboratively and purposefully construct knowledge | X |
| understand that computer mediated communications have advantages and disadvantages in supporting active participation in a community of practice and the management of collaboration on digital materials | X |
| **Managing and operating ICT** | |
| justify the selection of, and optimise the operation of, a selected range of devices and software functions to complete specific tasks, for different purposes and in different social contexts | X |
| apply an understanding of networked ICT system components to make changes to functions, processes, procedures and devices to fit the purpose of the solutions | X |
| manage and maintain data securely in a variety of storage mediums and formats | X |

## Links to Literacy

In this Years 9 and 10 task in Digital Technologies, students develop literacy as they learn how to communicate ideas, concepts and detailed proposals to a variety of audiences; read and interpret detailed written instructions for specific technologies, often including diagrams and procedural writings such as software user manuals, design briefs, patterns and recipes; prepare accurate, annotated engineering drawings, software instructions and coding; write project outlines, briefs, concept and project management proposals, evaluations, engineering, life cycle and project analysis reports; and prepare detailed specifications for production.

By learning the literacy of technologies, students understand that language varies according to context and they increase their ability to use language flexibly. Technologies vocabulary is often technical and includes specific terms for concepts, processes and production. Students learn to understand that much technological information is presented in the form of drawings, diagrams, flow charts, models, tables and graphs. They also learn the importance of listening, talking and discussing in technologies processes, especially in articulating, questioning and evaluating ideas.

Visit Literacy general capability <https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/literacy/>

Visit National Literacy Learning Progression <https://www.australiancurriculum.edu.au/resources/national-literacy-and-numeracy-learning-progressions/national-literacy-learning-progression/>

## Links to Numeracy

In this Years 9 and 10 Digital Technologies task, students are given opportunities to interpret and use mathematical knowledge and skills in a range of real-life situations. Students use number to calculate, measure and estimate; interpret and draw conclusions from statistics; measure and record throughout the process of generating ideas; develop, refine and test concepts; and cost and sequence when making products and managing projects. In using software, materials, tools and equipment, students work with the concepts of number, geometry, scale, proportion, measurement and volume. They use 3-dimensional models, create accurate technical drawings, work with digital models and use computational thinking in decision-making processes when designing and creating best-fit solutions.

Visit Numeracy general capability <https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/numeracy/>

Visit National Numeracy Learning Progression <https://www.australiancurriculum.edu.au/resources/national-literacy-and-numeracy-learning-progressions/national-numeracy-learning-progression/>

Links to learning areas

|  |
| --- |
| **Science** |
| **Year 9**  By the end of Year 9, students explain chemical processes and natural radioactivity in terms of atoms and energy transfers and describe examples of important chemical reactions. They describe models of energy transfer and apply these to explain phenomena. They explain global features and events in terms of geological processes and timescales. They analyse how biological systems function and respond to external changes with reference to interdependencies, energy transfers and flows of matter. They describe social and technological factors that have influenced scientific developments and predict how future applications of science and technology may affect people’s lives.  Students design questions that can be investigated using a range of inquiry skills. They design methods that include the control and accurate measurement of variables and systematic collection of data and describe how they considered ethics and safety. They analyse trends in data, identify relationships between variables and reveal inconsistencies in results. They analyse their methods and the quality of their data, and explain specific actions to improve the quality of their evidence. They evaluate others’ methods and explanations from a scientific perspective and use appropriate language and representations when communicating their findings and ideas to specific audiences.  **Year 10**  By the end of Year 10, students analyse how the periodic table organises elements and use it to make predictions about the properties of elements. They explain how chemical reactions are used to produce particular products and how different factors influence the rate of reactions. They explain the concept of energy conservation and represent energy transfer and transformation within systems. They apply relationships between force, mass and acceleration to predict changes in the motion of objects. Students describe and analyse interactions and cycles within and between Earth’s spheres. They evaluate the evidence for scientific theories that explain the origin of the universe and the diversity of life on Earth. They explain the processes that underpin heredity and evolution. Students analyse how the models and theories they use have developed over time and discuss the factors that prompted their review.  Students develop questions and hypotheses and independently design and improve appropriate methods of investigation, including field work and laboratory experimentation. They explain how they have considered reliability, safety, fairness and ethical actions in their methods and identify where digital technologies can be used to enhance the quality of data. When analysing data, selecting evidence and developing and justifying conclusions, they identify alternative explanations for findings and explain any sources of uncertainty. Students evaluate the validity and reliability of claims made in secondary sources with reference to currently held scientific views, the quality of the methodology and the evidence cited. They construct evidence-based arguments and select appropriate representations and text types to communicate science ideas for specific purposes. |
| **Content descriptions**  **Year 9 Science Inquiry Skills**   * Select and use appropriate equipment, including digital technologies, to collect and record data systematically and accurately ([ACSIS166](https://www.scootle.edu.au/ec/search?accContentId=ACSIS166)) * Analyse patterns and trends in data, including describing relationships between variables and identifying inconsistencies ([ACSIS169](https://www.scootle.edu.au/ec/search?accContentId=ACSIS169))   **Year 10 Science Inquiry Skills**   * Select and use appropriate equipment, including digital technologies, to collect and record data systematically and accurately ([ACSIS200](https://www.scootle.edu.au/ec/search?accContentId=ACSIS200)) * Analyse patterns and trends in data, including describing relationships between variables and identifying inconsistencies ([ACSIS203](https://www.scootle.edu.au/ec/search?accContentId=ACSIS203)) |

Appendix 2

## Support materials

Things to think about

*Assessment planning*

*Rich questions and discussion starters*

Students with diverse needs

## Resources

Activity guide: *Investigating environmental data with microcontrollers activity guide (Years 9–10)*

Glossary

Bibliography

# Support materials

# **Things to think about**

### *Assessment planning*

#### Formative assessment

Student understanding can be assessed in a formative fashion using a number of tools, including quizzes, assignments or a class discussion where the logic of a program is discussed.

Some other methods which are particularly suitable to assessing computational thinking are comment insertion, desk checking and Parson’s problems.

**Comment insertion:** Lines of code with comments stripped out are given to a student who is required to comment on the function of each line.

**Desk checking:** A process usually carried out by a single programmer who takes on the role of a computer and ‘acts out’ the instructions in his or her program code to validate algorithms and programs. Here, a student pair could take on the role of instruction giver and instruction actor to test the logic of a program. This is particularly effective with pseudocode (structured English).

**Parsons’s problems:** Lines of code are printed out then presented in random order. The student needs to re-organise the lines to correctly describe program flow.

For more background, see:

<https://medium.com/swlh/how-to-study-computer-programming-parsons-problems-2bfdefabfd86>

and

<https://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-88.pdf>

Parson’s problems can be organised on a webpage or pages for the student to rearrange. The line-based grader is a way to make Parson’s problems quickly and easily with a single right answer. This grader does not require you to use a certain programming language, which makes it a great way to abstract processes and workflows one level higher than actual code.

See: <https://codio.github.io/parsons-puzzle-ui/dist/>

Pre-prepared Parson’s problems for Activities 1, 2 and 4 are at: [https://martinlevins.github.io/ParsonsProblems](https://martinlevins.github.io/ParsonsProblems/)

For example, the code:

void setup ()

{

Serial.begin(9600);

}

void loop()

{

while (true)

{

Serial.println("test");

delay(1000);

}

}

is pasted into the problem engine. Students could self-assess or peer assess, or they could screenshot a completed problem and email it to the teacher or paste it into an online portfolio document.

Screen recordings can work well here where the student vocalises their approach to explain the process they are undertaking.

Complex code won’t necessarily work with this, but it is a good idea for beginners who are a bit confused by braces and indentation. Indentation can be solved by Arduino’s Auto Format feature (Tools > Auto Format), but it’s a good practice to recognise its significance, nonetheless.

#### Summative assessment

This assessment task is based on the humidity, temperature and power management concepts covered in the Activities. Use it as it stands or as a template and modify. For example, students could suggest different sensors or different scenarios based on this template.

###### Brief

Build a solution for a company called Brie Bay, based in the Napa Valley, north of San Francisco   
in California.

They need to monitor humidity and temperature in their storage area, used for the ageing of cheeses.

###### User story

General Manager, Paris Stilton, explains:

“For storage, our cheeses require a temperature range of 10–15°C during the ageing process, with humidity between 75% and 95%.

We have a natural cave with near-perfect temperature and humidity outside our cheesemaking facility. This cave is very similar to those used by cheesemakers in France and we like to produce cheese using their traditional methods.

The problem is that people keep forgetting to check the conditions in the cave, unless they are taking a new batch to the cave or collecting cheese for delivery.

Our cheeses sell at a very high price to several restaurants, and we can’t afford any spoilage.

Can you build a monitoring system that will solve our problem? There’s no power in the cave but surprisingly you can get wi-fi there!”

###### The design

As designer, your original consideration is to use ESP microcontroller technology with DHT22 or DHT11 sensors, powered by a 3.3V battery or an external power source. Your system could also send an alert if the temperature or the humidity exceeded limits.

What questions do you need to ask before finalising your design?

What are the functional and non-functional requirements for your solution? See:

<https://www.guru99.com/functional-vs-non-functional-requirements.html>

Which cloud-based service would you use, ThingSpeak or TagoIO? Support your decision with a table that compares the features of both.

### *Rich questions and discussion starters*

Promoting computational thinking with programming

<https://dl.acm.org/doi/10.1145/2481449.2481466>

How Learning Coding Can Help You With Critical Thinking

<https://www.youngwonks.com/blog/Improving-Critical-Thinking-through-Coding>

Enhancing Students’ Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming

<https://www.jstor.org/stable/26229205?seq=1#metadata_info_tab_contents>

## Students with diverse needs

Students may need **scaffolded support materials**. Adjustments to this task might include:

* placing students in groups with students who can support them with encouraging questions and ideas during the analysis and design phase
* grouping students with peer-mentors who can support their literacy or numeracy needs (including training students who find the task too easy to be effective peer-mentors)
* having students with literacy support needs answer questions using video or recorded voice rather than writing or typing
* using teacher assistants to support literacy demands of a task to enable student to show evidence of digital technologies learning
* encouraging students to communicate via online secure chat for those who rarely speak up during group work
* checking in at frequent intervals to determine student’s understanding of the task
* focusing on what students can do rather than what they cannot do when providing feedback.

Use professional judgement to provide rapid support when students are struggling with a task due to its literacy or numeracy demands.

Students might need opportunities for **extension**. Adjustments for such students might include:

* the use of advanced technologies to collect data on temperature fluctuations in different areas in the school
* the design and implementation of digital survey tools to survey members of the school community about the learning environment
* training in mentorship and opportunities to support other students with encouraging questions and ideas.

Change the approach to delivery of this task if a student is disengaged or is finding activities too easy or too hard; adopt a different approach to teaching the same aspect of literacy or numeracy.

See also: [evidenceforlearning.org.au/guidance-reports/improving-literacy-in-secondary-schools/](https://evidenceforlearning.org.au/guidance-reports/improving-literacy-in-secondary-schools/)

## Glossary

**analogue** or **analog**

Of or relating to a signal voltage which can take any value in a range. For example, a light dependent resistor may present a voltage between 0 and 1 V to an analog inout on a microcontroller. The microcontroller will divide this range up into a binary number between 0 and 255 (8-bit or 28 analog) or 0–1023 (10-bit or 210 analog).

**general-purpose programming languages**

Programming languages in common use designed to solve a wide range of problems. Examples include C#, C++, Java, JavaScript, Python, Ruby and Visual Basic. In this document, we concentrate on the Arduino variant of C++.

**libraries**

Pieces of code contained in a file. These pieces carry out specific tasks that need to be written once and re-used each time the environment of the library is used. For example, the library ‘ESP8266Wi‑fi.h’ contains objects that allow interaction with the wi-fi capabilities of all members of the ESP family.

**lux**

A unit of the illumination or amount of light falling on a surface. For example, we can use a light meter or lux meter to measure the light reflected from a person’s face to set the exposure and shutter speed of a camera. This is done automatically in most cameras. Lux is calculated from light levels, but also the colour make-up of the incident light.

**objects**

A programming structure that contains (encapsulates) properties (data) and methods (functions).

**pseudocode**

A way of showing [algorithms](http://encyclopedia.kids.net.au/page/al/Algorithm) without use of any specific programming language. This makes the algorithm easy to understand for everyone, whatever programming language they might use. Pseudocode is usually written in English text. Purists would insist that pseudocode be written with no recognisable computer language words, but some common operation words, for example, *if, then* and *else,* are commonly found in pseudocode.

An object is a **variable**

A variable is the name given to a reserved area of computer memory to hold data that can change. They need to be declared in program code so that the computer knows to reserve sufficient memory. For example, a variable may be declared to hold the reading from a sensor as an integer (counting numbers: 0,1,2,3,4,5 etc.) or as a floating-point number (fractional numbers such as 1.56789), or it may be intended to hold a piece of text. In each case, the variable will need to be declared as a type that matches the data it is to hold. Variables may also need to be initialised to a set value before use.

**wi-fi**

A communications standard which is short for wireless fidelity. This networking system can have clients or access points to which clients connect. The access point can relay the client data to, or retrieve data from, a network for them. Some devices can act as both clients and access points.

See also the glossary for the Australian Curriculum: Technologies: [www.australiancurriculum.edu.au/f-10-curriculum/technologies/glossary/](http://www.australiancurriculum.edu.au/f-10-curriculum/technologies/glossary/)

## Bibliography

Allen, J. G., MacNaughton, P., Satish, U., Santanam, S., Vallarino, J. & Spengler, J. D. (2016). Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers: a controlled exposure study of green and conventional office environments. *Environmental health perspectives*, 124(6), 805–812.

Barrett, P., Davies, F., Zhang, Y. & Barrett, L. (2015). The impact of classroom design on pupils' learning: Final results of a holistic, multi-level analysis. Building and Environment, 89, 118–133.

Carrington, D. & Kuo, L. (2018). Air pollution causes ‘huge’ reduction in intelligence, study reveals. The Guardian, Aug  27. Retrieved from https://[www.theguardian.com/environment/2018/aug/27/air-pollution-causes-huge-](http://www.theguardian.com/environment/2018/aug/27/air-pollution-causes-huge-)reduction-in-intelligence-study-reveals

Graff Zivin, J., Hsiang, S. M. & Neidell, M. (2018). Temperature and human capital in the short and long run. *Journal of the Association of Environmental and Resource Economists*, 5(1), 77‑105.

Heppell, S. (n.d.). Learnometer. Retrieved from <http://www.learnometer.net/>

Mealings, K. (2016). Classroom acoustic conditions: Understanding what is suitable through a review of national and international standards, recommendations, and live classroom measurements. Conference paper, Acoustics 2016 Brisbane. Available at https://[www.researchgate.net/publication/310651345\_Classroom\_acoustic\_conditions\_Un](http://www.researchgate.net/publication/310651345_Classroom_acoustic_conditions_Un) derstanding\_what\_is\_suitable\_through\_a\_review\_of\_national\_and\_international\_standards\_ recommendations\_and\_live\_classroom\_measurements

Mooney, C. (2015). Paper finds a surprising link between warmer temperatures and math test scores. The Washington Post. Retrieved from https://[www.washingtonpost.com/news/energy-environment/wp/2015/05/12/paper-](http://www.washingtonpost.com/news/energy-environment/wp/2015/05/12/paper-) finds-a-surprising-link-between-warm-temperatures-and-math-test- scores/?postshare=7651431446442153

Park, R. J. (2019). Heat wave: Air conditioned schools would narrow the racial achievement gap. USA Today. Retrieved from https://[www.usatoday.com/story/opinion/2019/08/15/heat-](http://www.usatoday.com/story/opinion/2019/08/15/heat-)wave-students-need-air-conditioning-close-achievement-gap-column/1996394001/

Picard, M. & Bradley, J. S. (2001). Revisiting speech interference in classrooms. *Audiology*, 40(5), 221–244. Retrieved from <https://pubmed.ncbi.nlm.nih.gov/11688542/>

Romm, J. (2015). Elevated CO2 Levels Directly Affect Human Cognition. Climate Progress, Oct, 26. Retrieved from [https://thinkprogress.org/exclusive-elevated-co2-levels-directly- affect-human-cognition-new-harvard-study-shows-2748e7378941/](https://thinkprogress.org/exclusive-elevated-co2-levels-directly-%20affect-human-cognition-new-harvard-study-shows-2748e7378941/)

Zhang, X., Chen, X. & Zhang, X. (2018). The impact of exposure to air pollution on cognitive performance. *Proceedings of the National Academy of Sciences*, 115(37), 9193– 9197. DOI:10.1073/pnas.1809474115

Appendix 3

## Digital systems task planning template

This template is a suggested step-by-step approach that teachers might use to consider whether *all* or *any* of these links apply to an assessment task they develop themselves to better reflect the learning needs of their students and the context of their classroom and school.

# Planning template suggested approach

Below is a broad outline of how to use the assessment task planning template on the following pages. It reflects the work of Wiggins and McTighe (2012) on Understanding by Design which features a backward design approach.

1. Begin with Digital Technologies:
   1. determine the aspects of the achievement standard that will be the focus of the task
   2. highlight the relevant aspects of the standard
   3. identify what knowledge and skills students will need in order to demonstrate the achievement standards (content descriptions)
   4. identify the strands and threads that will need to be addressed.
2. As Digital Technologies is the driving learning area, it is suggested that only the key ideas for this learning area be identified.
3. Indicate the key concepts of Digital Technologies that will be addressed and how.
4. Scan the Australian Curriculum to find meaningful connections between:
   1. learning areas (2 learning areas helps keep learning focused; avoid more than 3)
   2. general capabilities
   3. cross-curriculum priorities.

For example, connections could be established on the grounds of:

1. common concepts/key ideas, such as data/design/ways of thinking
2. common words, such as ‘create’, ‘communicate’ and ‘control’
3. contexts, from learning areas such as Science, HASS, HPE, The Arts.
4. Indicate what general capabilities and cross-curriculum priorities can be meaningfully addressed in the assessment task.
5. Construct a task that allows for discrimination in performance and includes:
   * title
   * band level
   * duration
   * task summary, including prior learning
   * achievement standards and content descriptions
   * task
   * assessment rubric.

Search for xxxx and replace with your own text.

**Title: xxxx**

**Assessment focus:** Australian Curriculum: Digital Technologies   
(Digital systems and data). This task is also linked to xxxx. Depending on modifications made, opportunities may exist to link this task to xxxx.

**Band:** Years 9 and 10 (intended cohort Year xxxx)

**Context:** xxxx

**Duration:** xxxx

**Prior learning:** Students will have:

* xxxx

## Task summary

**Key inquiry question:**

* xxxx

**Focus questions:**

* xxxx
* xxxx

**Students will:**

* xxxx
* xxxx

### Task features

Students will be asked to complete the following:

* xxxx
* xxxx

## Digital Technologies

Achievement standard

By the end of Year 10, students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users. They explain simple data compression, and why content data are separated from presentation.

Students plan and manage digital projects using an iterative approach. They define and decompose complex problems in terms of functional and non-functional requirements. Students design and evaluate user experiences and algorithms. They design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and data entities. They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions. They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise. They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects.

Content descriptions

|  |
| --- |
|  |

## Content strands [X any that apply]

|  |  |  |  |
| --- | --- | --- | --- |
| **Digital Technologies knowledge and understanding** | | **Digital Technologies processes and production skills** | |
| Digital systems  Representation of data |  | Collecting, managing and analysing data  Creating digital solutions by:  investigating and defining  generating and designing  producing and implementing  evaluating  collaborating and managing |  |

## Links to the key ideas [X any that apply]

## Read more about the [key ideas in the Australian Curriculum: Technologies](https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/key-ideas/).

|  |  |  |
| --- | --- | --- |
| **Creating preferred futures** | Students develop solutions to meet needs considering impacts on liveability, economic prosperity and environmental sustainability. |  |
| **Project management** | Students will develop skills to manage projects to successful completion through planning, organising and monitoring timelines, activities and the use of resources. |  |
| **Thinking in Technologies**   * Systems thinking | Systems thinking is a holistic approach to the identification and solving of problems where the focal points are treated as components of a system, and their interactions and interrelationships are analysed individually to see how they influence the functioning of the entire system. |  |
| * Design thinking | Design thinking involves the use of strategies for understanding design needs and opportunities, visualising and generating creative and innovative ideas, planning, and analysing and evaluating those ideas that best meet the criteria for success. |  |
| * Computational thinking | Computational thinking is a problem-solving method that is applied to create solutions that can be implemented using digital technologies. It involves integrating strategies, such as organising data logically, breaking down problems into parts, interpreting patterns and models and designing and implementing algorithms. |  |

## Links to the key concepts

The [key concepts](https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/)that underpin the Digital Technologies curriculum establish a way of thinking   
about problems, opportunities and information systems and provide a framework for knowledge   
and practice. (Colour coding is based on the [Australian Computing Academy scheme](https://aca.edu.au/#what-is-the-digital-technologies-curriculum).)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **abstraction** | underpins all content, particularly the content descriptions relating to the concepts of data representation; and specification; algorithms; and implementation | |
|  | **data collection** | (properties, sources and collection of data) |  |
|  | **data representation** | (symbolism and separation) |  |
|  | **data interpretation** | (patterns and contexts) |  |
|  | **specification** | (descriptions and techniques) |  |
|  | **algorithms** | (following and describing) |  |
|  | **implementation** | (translating and programming) |  |
|  | **digital systems** | (hardware, software, and networks and the internet) |  |
|  | **interactions** | (people and digital systems, data and processes) |  |
|  | **impacts** | (sustainability and empowerment) |  |

Cross-curriculum priorities[X any that apply] [Read more…](https://www.australiancurriculum.edu.au/f-10-curriculum/cross-curriculum-priorities/)

|  |  |  |
| --- | --- | --- |
| **Aboriginal and Torres Strait Islander histories and cultures** | **Asia and Australia’s engagement with Asia** | **Sustainability** |
|  |  |  |

## General capabilities [X any that apply] [Read more…](https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Literacy** | **Numeracy** | **ICT Capability** | **Critical and Creative Thinking** | **Ethical Understanding** | **Personal and Social capability** | **Intercultural Understanding** |
|  |  |  |  |  |  |  |

## Links to ICT Capability continuum: Level [ ] [X any that apply] [Read more…](https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/information-and-communication-technology-ict-capability/)

Depending on the year level this activity is being used with, adjust content to the appropriate level;  
for example, Level 4 or 5.

|  |  |
| --- | --- |
| **Apply social and ethical protocols and practices when using ICT** | |
| identify and describe ethical dilemmas and consciously apply practices that protect intellectual property |  |
| use a range of strategies for securing and protecting information, assess the risks associated with online environments and establish appropriate security strategies and codes of conduct |  |
| independently apply appropriate strategies to protect rights, identity, privacy and emotional safety of others when using ICT, and discriminate between protocols suitable for different communication tools when collaborating with local and global communities |  |
| assess the impact of ICT in the workplace and in society, and speculate on its role in the future and how they can influence its use |  |
| **Investigating with ICT** | |
| select and use a range of ICT independently and collaboratively, analyse information to frame questions and plan search strategies or data generation |  |
| use advanced search tools and techniques or simulations and digital models to locate or generate precise data and information that supports the development of new understandings |  |
| develop and use criteria systematically to evaluate the quality, suitability and credibility of located data or information and sources |  |
| **Creating with ICT** | |
| select and use ICT to articulate ideas and concepts, and plan the development of complex solutions |  |
| Design, modify and manage complex digital solutions, or multimodal creative outputs or data transformations for a range of audiences and purposes |  |
| **Communicating with ICT** | |
| select and use a range of ICT tools efficiently and safely to share and exchange information, and to collaboratively and purposefully construct knowledge |  |
| understand that computer mediated communications have advantages and disadvantages in supporting active participation in a community of practice and the management of collaboration on digital materials |  |
| **Managing and operating ICT** | |
| justify the selection of, and optimise the operation of, a selected range of devices and software functions to complete specific tasks, for different purposes and in different social contexts |  |
| apply an understanding of networked ICT system components to make changes to functions, processes, procedures and devices to fit the purpose of the solutions |  |
| manage and maintain data securely in a variety of storage mediums and formats |  |

### Links to Literacy and Numeracy

Depending on the year level this activity is being used with adjust content to appropriate level.

Links to Literacy

xxxx

Links to Numeracy

xxxx

Appendix 4

## Background information on electronics

An explanation of some information on electronics, which might be useful before starting students on the activities

# Background information on electronics

### Voltage and current – what’s the difference?

Voltage measures how much energy can be given to a charge; current measures (in amps) how much charge is flowing in a circuit. Clearly, if a small number of charges experiences a low voltage, there won’t be much expended energy in total.

### Resistance

The resistance of a material measures how much energy needs to be given to move charges in that material. Hence, a high voltage won’t move many charges through a high-resistance material. This is why we can use rubber or plastic as insulators. However, if we provide a high enough voltage, current will flow.

### Short circuits

Very low resistance means lots of current, even for fairly low voltages. Lots of current means lots of heat. Lots of heat causes the leakage of ‘magic smoke’ from components, which usually results in their destruction.

### Colour conventions

Red is for the positive, often referred to as Vcc,which can be either 5V or 3.3V in most digital circuits. This document uses all 3.3V equipment and care must be taken to apply the correct voltage otherwise all the ‘magic smoke’ leaks out.

Black is traditionally used for ground (GND) – what some may refer to as negative.

### What happens when all the ‘magic smoke’ leaks out?

Leakage of smoke is, of course, a joke. There is no magic in the smoke – the real magic is in the conjunction of different semiconductors arranged in special ways so that devices do your bidding.

Overvoltage, or short circuits due to sloppy connections, are usually the cause of too much current which overheats components to the point of combustion, releasing smoke.

Don’t let the smoke escape.

### Suggestions regarding handling and connecting devices

Care, care and more care. No kid gloves needed, but connecting positive to negative, flexing of leads and pins, pushing pins or components too hard into breadboards can destroy equipment.

When connecting devices, do so with power turned off or disconnected. Before turning on or reconnection, double-check that positive is connected to Vcc and ground to black. (Breadboards can often use blue for ground, just to make life interesting.)

### What is a pull-up resistor and how do I work out how big it needs to be?

Some circuits (none in this document) refer to a ‘pull-up’ or perhaps a ‘pull-down’ resistor. Should you need advice on this, see:

<https://electronics.stackexchange.com/questions/23645/how-do-i-calculate-the-required-value-for-a-pull-up-resistor>

Appendix 5

## Background information on breadboards

An explanation of some information on breadboards, which might be useful before starting students on the activities

# Breadboards

Some issues that your students may face if not familiar with breadboards are outlined below.

### Spatial issues

It takes a while for students to get to grips with the spatial arrangement of a breadboard.

The purchase of transparent boards can help here.

### They are designed to come apart

Most boards are composed of 2–3 sections, bound together with tape or adhesive. Care in handling prevents this structure from coming apart.

### Clips in the board may not make good connections

Insertion of components requires care, and sometimes moving the component in and out can remove any high-resistance material that is preventing good contact.

### How do I connect components on a breadboard?

Buy a lot of short linking Dupont wires.

Buy double quantities of red and black as every project will use them.

### Online resources

* How to Use a Breadboard:  
  <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all>
* 5 Tips for Successful Breadboarding:  
  <https://www.instructables.com/Basics-of-Troubleshooting-Small-Breadboard-Circuit/>
* Breadboard Tips & Tricks:  
  <https://learn.adafruit.com/breadboards-for-beginners/breadboard-tips-and-tricks?view=all>

### Power supply to breadboards

The series of activities in *Investigating environmental data with microcontrollers activity guide* can be powered by USB via the cable from the computer to the microcontroller, but separate breadboard power supplies that provide 5V and 3.3V are available from many vendors.

The big issue here is that students carefully monitor which voltage is being supplied. The ESP series of microcontrollers will only work on 3.3V. Any higher and all the magic smoke leaks out.

Appendix 6

## Drawing circuit layouts

An explanation of some information on circuit layouts, which might be useful before starting students on the activities

## Drawing circuit layouts

Online resources are available should you want to print a breadboard layout. There are 2 main sites: circuit.io and Fritzing.

The circuito.io resource uses a drag-and-drop interface to select a microcontroller, then components, connects them up to a wiring diagram, gives you a bill of materials and creates Arduino code to test the system. See: <http://www.circuito.io/>

Graphical user interface

Description automatically generated  
The Circuito environment

It looks like Circuito is easy to use, but you still need to develop an algorithm and know the coding environment to provide a solution. It will be easier for students to:

1. think through the problem for which you are developing a solution
2. sketch a circuit diagram (see below)
3. assemble the system
4. use the Arduino IDE to code and download to the microcontroller.

Using a layout app such as Fritzing to design the circuit

As explained on the fritzing.org website:

‘Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional printed circuit boards.’

See:

<http://en.wikipedia.org/wiki/Open-source_hardware>

<http://processing.org/>

<http://arduino.cc/>

<http://fritzing.org/>

Fritzing costs about AU$12, so buying one copy for your classroom would be reasonable. This app has been used to prepare diagrams for this document, so you can see for yourself what it produces and work out whether it is worthwhile buying.

A picture containing diagram

Description automatically generated

The fritzing interface, giving access to a parts bin, and the ability to show connections   
and annotate areas of interest

Image source: fritzing.org CC-By-SA 4.0

Appendix 7

## Arduino tips

Background information on Arduino technologies, which might be useful before starting students on the activities

# Arduino tips

## Background

Arduino is both a microcontroller and an integrated development environment (IDE).

See: <https://en.wikipedia.org/wiki/Integrated_development_environment>

It uses a compiled language as opposed to Python, which is interpreted.

An interpreted language converts the text you see to machine code on the fly, whereas a compiled language uses another piece of software to convert the text into machine code, and this converted code is loaded to the microcontroller in one go.

See: <https://www.arduino.cc/reference/en/>

Here, we are using the Arduino IDE to program a different board which understands the language that Arduino uses.

### Access

You can download the IDE for Mac, Windows and Linux from the Arduino website. See: <https://arduino.cc>

Depending on your system, you may require admin privileges to install the IDE and/or some of the support files such as board descriptions, serial drivers and libraries.

### Choosing a board, and adding libraries

As it comes, the IDE only ‘understands’ the Arduino board (there are several varieties of these and the installed board will allow you to choose your particular model).

#### Set up Arduino for an ESP board

Graphical user interface, application

Description automatically generated  
Arduino Preferences setting for the ESP board

Paste this URL into the Additional boards section above:

<http://arduino.esp8266.com/stable/package_esp8266com_index.json>

### Choosing libraries

Libraries add functionality that can be re-used in different circumstances. A library might contain information on how to print, how to communicate with a particular sensor, and so forth.

Libraries may also have dependencies. In other words, they refer to data or functions contained in another library, so it must be present in your libraries as well.

Appendix 8 gives more detail.

### Check your spelling!

Spelling and capitalisation are paramount and will surely cause you grief unless you pay close attention. Over time, looking at your own code will make you blind to simple mistakes, so engage a buddy to read it through with you.

If pain persists, engage a second one. Double-check entries, and don’t rely on the colour coding for correct spelling. Sometimes you can have a function that looks as if it is spelt correctly, but you have passed too many arguments to it because you thought it was a different function.

For example, the functions writeField and writeFields look very similar but take different numbers of arguments and you will (for certain) miss the ‘s’ occasionally.

### Make it readable!

#### Comments

It’s good practice to use // before any explanatory (comments) text. This turns the text into comments for human use only – those lines are ignored by the compiler.

To start with, things like void () doesn’t make it easy for first-timers, so add comments liberally to tell others (and remind yourself) what different bits of your code are doing.

Commenting is also useful in debugging. If you are unsure why a program is not behaving, comment out (add // in front of) complex code and enter simpler code such as simply outputting the value of, say, variables to the serial port, so that you can check their value and compare to what you were expecting.

This process allows you to set a ‘flag’ that will highlight values.

Commenting can also help you with building the code piece by piece, inserting a method stub. See: <https://en.wikipedia.org/wiki/Method_stub>

#### Indentation

Use CTRL-T (Windows) or Command-T (Mac) in the IDE to automatically format (Tools > auto format), which will add indentation as needed.

### Use of variables

Variables can be different types such as char, byte, int, unsigned int, long, unsigned long, float, double, String, most common – find out what they mean at: <https://www.arduino.cc/en/Reference/VariableDeclaration>.

Write easy-to-read (and modify!) code. See guidelines at: <https://www.arduino.cc/en/Reference/StyleGuide>

Different types of variables need differing amounts of space, so they need to be declared so the computer knows how much memory needs to be reserved for data. Variables may also need to be initialised (set to a known value before use) depending on your code.

Camel case is the norm for function and variable names. This refers to compound terms such as iPhone and eBay, in which the capital letters within the ‘word’ look like a camel hump. Where the name is made up of several words, start with either case, then use a capital for the first letter of each new ‘word’.

For example:

ledPin

motorPin

prevPotVal

If they're single ‘word’ names, all lower case:

pot

servo

motor

Class names usually start with a capital letter, then follow the camel case form:

Servo

Serial

liquidCrystal

newPing

### What is a define?

Defines are often (but not always!) in upper case, often with individual ‘words’ separated by underscores:

LCD\_I2C\_ADDRESS

MAX\_DISTANCE

TRIG\_PIN

#define ledPin 3

// The compiler will replace any mention of ledPin with the value 3 at compile time.

Constants should really follow the same rule as defines, but will often be seen in the same camel case form as variables.

In general, the [[const](https://www.arduino.cc/reference/en/language/variables/variable-scope-qualifiers/const)](https://www.arduino.cc/reference/en/language/variables/variable-scope-qualifiers/const) keyword is preferred for defining constants and should be used instead of #define.

How to find commands? Look at keywords.txt (assuming the author has put this in).

USB programmer may need a driver. See:   
<https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all>

Appendix 8

## Structure of object-oriented programming languages

An explanation of object-oriented programming, which might be useful before starting students on the activities

# Structure of object-oriented programming languages

## Object oriented – what’s that?

‘Object Oriented Programming (OOP) is a programming model where programs are organized around objects and data rather than action and logic’. See:

<https://www.c-sharpcorner.com/UploadFile/mkagrahari/introduction-to-object-oriented-programming-concepts-in-C-Sharp/>

Objects combine data and the functions that operate on that data into classes. This allows a much more efficient organisation of the program, largely through leveraging abstraction. So, we don’t really need to know how to communicate with a given sensor and how to interpret the values that it is returning – we can just use a library that contains all the classes to do it for us.

### How libraries are used in Arduino

A library is a class in that it has data and functionality. There are lots of good tutorials online. See:

<https://www.arduino.cc/en/Hacking/libraryTutorial>

### Why classes?

An example of abstraction, classes have names and constructors with the same name as the class. Classes ideally abstract the functionality so they cover a lot of examples.

Class ledBlink

{

ledBlink(int pin, int delay)

blink();

}

Libraries take the idea of abstraction to another level – abstracting functionalities cover all the examples that the author has thought of.For example, one of the variable assignations we make in the DHT example in Activity 4 is:

float temperature= dht.getTemperature();

Here, our floating-point variable ‘temperature’ is given the value from the dht library, where the class getHumidity is defined. (Remember we told our program that it could replace DHTesp with dht so to save space and clarify the purpose of any call.)So, the program looks into the library and finds out how to return the results of the function.

And here it is below. All the complexity below is replaced with one phrase – abstraction at its greatest.

float DHTesp::getTemperature()

{

readSensor();

if (error == ERROR\_TIMEOUT)

{ // Try a second time to read

readSensor();

}

return temperature;

}

void DHTesp::readSensor()

{

// Make sure we don't poll the sensor too often

// - Max sample rate DHT11 is 1 Hz (duty cicle 1000 ms)

// - Max sample rate DHT22 is 0.5 Hz (duty cicle 2000 ms)

unsigned long startTime = millis();

if ((unsigned long)(startTime - lastReadTime) < (model == DHT11 ? 999L : 1999L))

{

return;

}

lastReadTime = startTime;

temperature = NAN;

humidity = NAN;

uint16\_t rawHumidity = 0;

uint16\_t rawTemperature = 0;

uint16\_t data = 0;

// Request sample

digitalWrite(pin, LOW); // Send start signal

pinMode(pin, OUTPUT);

if (model == DHT11)

{

delay(18);

}

else

{

// This will fail for a DHT11 - that's how we can detect such a device

delay(2);

}

pinMode(pin, INPUT);

digitalWrite(pin, HIGH); // Switch bus to receive data

// We're going to read 83 edges:

// - First a FALLING, RISING, and FALLING edge for the start bit

// - Then 40 bits: RISING and then a FALLING edge per bit

// To keep our code simple, we accept any HIGH or LOW reading if it's max 85 usecs long

#ifdef ESP32

// ESP32 is a multi core / multi processing chip

// It is necessary to disable task switches during the readings

portMUX\_TYPE mux = portMUX\_INITIALIZER\_UNLOCKED;

portENTER\_CRITICAL(&mux);

#else

// cli();

noInterrupts();

#endif

for (int8\_t i = -3; i < 2 \* 40; i++)

{

byte age;

startTime = micros();

do

{

age = (unsigned long)(micros() - startTime);

if (age > 90)

{

error = ERROR\_TIMEOUT;

#ifdef ESP32

portEXIT\_CRITICAL(&mux);

#else

// sei();

interrupts();

#endif

return;

}

} while (digitalRead(pin) == (i & 1) ? HIGH : LOW);

if (i >= 0 && (i & 1))

{

// Now we are being fed our 40 bits

data <<= 1;

// A zero max 30 usecs, a one at least 68 usecs.

if (age > 30)

{

data |= 1; // we got a one

}

}

switch (i)

{

case 31:

rawHumidity = data;

break;

case 63:

rawTemperature = data;

data = 0;

break;

}

}

#ifdef ESP32

portEXIT\_CRITICAL(&mux);

#else

// sei();

interrupts();

#endif

// Verify checksum

if ((byte)(((byte)rawHumidity) + (rawHumidity >> 8) + ((byte)rawTemperature) + (rawTemperature >> 8)) != data)

{

error = ERROR\_CHECKSUM;

return;

}

// Store readings

if (model == DHT11)

{

humidity = (rawHumidity >> 8) + ((rawHumidity & 0x00FF) \* 0.1);

temperature = (rawTemperature >> 8) + ((rawTemperature & 0x00FF) \* 0.1);

}

else

{

humidity = rawHumidity \* 0.1;

if (rawTemperature & 0x8000)

{

rawTemperature = -(int16\_t)(rawTemperature & 0x7FFF);

}

temperature = ((int16\_t)rawTemperature) \* 0.1;

}

error = ERROR\_NONE;

}

Appendix 9

## Serial and I2C communications

An explanation of serial and I2C bus communications, which might be useful before starting students on the activities

# Serial and I2C communications

## What’s serial?

We look at using serial communications in Activity 1 of *Investigating environmental data with microcontrollers activity guide*, where we send data down one line from one device to another.

That’s pretty much it. Each data bit is represented by the presence or absence of a pulse of electricity giving us the binary zeroes and ones that we’re used to.

The serial protocol needs a couple of extra signals to control the flow of bits. Not surprisingly, these are called control characters and used to be inserted by pressing the CTRL key with another symbol. So CTRL-n meant new line, CTRL-l meant start a new page and so forth.

Special control characters referred to as **ACK** and **NAK** are used to **ACK**nowledge receipt or **N**egatively **A**c**K**nowledge (which means, nuh – didn’t get that).

Another bunch of characters exist such as STOP and START bits that tell the receiver when the data stream starts and stops, and checksums that calculate what data was sent so that the receiver can do the same calculation to see if it received the correct message. If so, it may send an ACK; if not, it will send a NAK so the sender knows that it has to retransmit.

You can watch series of 4 videos showing animations of computer communications from serial to the internet at:<https://www.youtube.com/playlist?list=PLOZ9tlz1QeommInICVRA-1yfRyhaQ1v2s>

The most relevant is the first, starting at: <https://youtu.be/7izv9XxFt7A?list=PLOZ9tlz1QeommInICVRA-1yfRyhaQ1v2s&t=23>

The problem with serial is that it sends from one device to another.

What if you have several devices? Do you need one wire for each? Read on.

### The I2C bus

Serial communications are one bit at a time, presented by one electrical pulse. So do we need one serial line for each device?

We would, unless we get cunning and use an I2C protocol on a bus. (There are many other communications protocols out there, but we’ll work with this one here). Let’s start with the meaning of the word ‘bus’.

In Paris in the 17th century, rather than individually hire a horse-drawn cab, people could get on a bigger horse-drawn vehicle at different stops and get off at different stops. This vehicle was called an omnibus. ‘Omnibus’ means ‘all’, although only officers of the army and royalty were allowed on. It’s no small wonder then that it went broke and Paris didn’t have a bus service until much later.

So, ‘omnibus’ became shortened to ‘bus’ and this word was adopted by computing and electronics to mean that lots of signals can be combined on one or 2 wires and ‘hop on’ or ‘hop off’ different devices in an electrical circuit.

Just like a bus ticket, which tells the conductor (old school bus) where you’re getting off, each digital signal is encapsulated in a packet which has an address component.

That means that components which are all connected to the same bus know which packet to pluck off the wire. Only one wire will carry the data because its electrical signals are always compared with one ‘ground’ wire. It saves heaps of money in extra wiring.

All the binary numbers in the data are represented by electrical signals that are at some agreed voltage above ground on the data line (SDA), and the system knows how to separate a binary 00 from a binary 0.

Both of these will be 0 volts, so the system uses a clock to generate ticks, or pulses, on a separate wire called a clock line (SCL). If the binary 1 or 0 (HIGH or LOW) corresponds with a pulse, then we can separate 2 binary zeroes (they’ll last for 2 pulses) from one zero (that’ll last for only one pulse).

We also need some extra data such as where the packet starts and stops: ‘yep – got that’ (ACKnowledge), or I want to send data (Write), or I want to read data (Read).

Here is a representation of a packet of data:

Diagram

Description automatically generated

*CCBY4 from https://learn.sparkfun.com/tutorials/i2c/all*