# Digital Technologies – 7 and 8_ Creating digital solutions

| Strand | Knowledge and understanding | | Processes and production skills | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | *Creating digital solutions by:* | | | | | | | |
| | Digital systems | Representation of data | Collecting, managing and analysing data | Investigating and defining | | Generating and designing | | Producing and implementing | Evaluating | Collaborating and managing |
| **Content Description** | Investigate how data is transmitted and secured in wired, wireless and mobile networks, and how the specifications affect performance (ACTDIK023 ) | Investigate how digital systems represent text, image and audio data in binary (ACTDIK024 ) | Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025) | Analyse and visualise data using a range of software to create information, and use structured data to model objects or events (ACTDIP026 ) | Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027) | Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028) | Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029) | Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030) | Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031) | Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032) |

| **Sequence of Lessons / Unit** | **Approx. time rq'd** | **Year A or B** | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # | CD | Achievement standard # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Create an app or a game | 16 | 7 | ☐ | | ☐ | | ☐ | | ☐ | | ☑ | 4 | ☑ | 5 | ☑ | 5 | ☑ | 5 | ☑ | 6 | ☐ | |
| Robotics and embedded systems | 20 | 8 | ☑ | 1 | ☐ | | ☐ | | ☐ | | ☑ | 4 | ☑ | 5 | ☑ | 5 | ☑ | 5 | ☑ | 6 | ☐ | |

| **Years 5 and 6 Achievement Standard** | **Years 7 and 8 Achievement Standard** | **Years 9 and 10 Achievement Standard** |
|---|---|---|
| By the end of Year 6:<br>• Students explain the fundamentals of digital system components (hardware, software and networks) and how digital systems are connected to form networks.<br>• They explain how digital systems use whole numbers as a basis for representing a variety of data types.<br>• Students define problems in terms of data and functional requirements and design solutions by developing algorithms to address the problems.<br>• They incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program.<br>• They explain how information systems and their solutions meet needs and consider sustainability.<br>• Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols. | By the end of Year 8<br>• Students distinguish between different types of networks and defined purposes. (1)<br>• They explain how text, image and audio data can be represented, secured and presented in digital systems. (2)<br>• Students plan and manage digital projects to create interactive information. (3)<br>• They define and decompose problems in terms of functional requirements and constraints. (4)<br>• Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. (5)<br>• They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability. (6)<br>• They analyse and evaluate data from a range of sources to model and create solutions. (7)<br>• They use appropriate protocols when communicating and collaborating online. (8) | By the end of Year 10<br>• Students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users.<br>• They explain simple data compression, and why content data are separated from presentation.<br>• Students plan and manage digital projects using an iterative approach.<br>• They define and decompose complex problems in terms of functional and non-functional requirements.<br>• Students design and evaluate user experiences and algorithms.<br>• 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.<br>• They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions.<br>• They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise.<br>• They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects. |

**Robotics and embedded systems**

Student should develop an understanding of computer programming as a collection of smaller programs – functions that collectively work to solve complex problems. Students could use programmable robots or microcontrollers to solve problems of increasing complexity, progressively adding additional functions such as the control of motors, lights, sounds and sensors. Students should apply the problem-solving processes of defining the problem, generating design solutions and prototypes and following their algorithm when implementing their program. The final solution should be evaluated against stated criteria.

| Flow of activities | | | | |
|---|---|---|---|---|
| Short text | *The Internet of Things*<br>*Incorporate an electronic programming board when creating a digital solutions for a range of problems.* | *Designing for the user*<br>*Generate design ideas to map out the user experience.* | *Programming a solution*<br>*Test and make modifications to their program for their digital solution.* | *Evaluate*<br>*Organise a sharing of completed projects as part of the evaluation stage.* |
| Questions to guide exploration | *What problems can be solved using a programming board?* | *How do I design my digital solution?* | *How do I program my solution?* | *How well did my solution work?* |
| AC Alignment | Investigating and defining (ACTDIP027) | Generating and designing (ACTDIP028) (ACTDIP029) | Producing and implementing (ACTDIP030) | Evaluating (ACTDIP031) |
| What's this about? | Before a solution can be designed and created it is necessary to find out what is the cause of any existing problem and what will solve it or for a new situation, what is required of a solution. This means students must initially define the problem and decompose into a set of functional requirements that consider the social, technical and usability constraints to their solution.<br><br>Electronic programming boards can be used by students to create digital solutions for a range of problems. The programming boards typically use a microcontroller which is a small chip (a tiny computer) that sends and receives signals to turn things on and off. The microcontroller is connected to inputs such as buttons or sensors and outputs such as lights or a speaker. These components combined together are referred to as an embedded system. An embedded system is designed to run one program.<br><br>Examples of programing boards include Arduino (many different types including Lily Pad, Nano or Esplora), BBC MicroBit, Raspberry Pi and BlueBerry4.<br><br>LEGO® MINDSTORMS® products such as EV3 incorporate an on-board microcontroller referred to as an intelligent brick. | Programmable robots or microcontrollers can be incorporated into digital solutions to solve problems of increasing complexity, progressively adding additional functions such as the control of motors, lights, sounds and sensors.<br><br>At this level, students should be generating design ideas using techniques such as brainstorming, forced analogies, prototyping and SCAMPER (substitute, combine, add something, magnify or minify, put to other use, eliminate), A paper prototype can also be used in the design process to map out plans what's on screen, the logic behind transitioning between screens and how various elements may work together as a system. The paper prototype can inform algorithm development.<br><br>Algorithms are generally written as a flowchart or in pseudocode. At this level, students are expected to write their algorithms in structured English. | Student should develop an understanding of computer programming as a collection of smaller programs – functions, that collectively work to solve complex problems.<br><br>Link to Digital Systems: Many educational robot kits and microcontrollers can be connected together to form a networked environment, with opportunities to explore how data is transmitted to and from devices using wired connection, infrared, wireless connections, and in some cases data transmission methods such as sound, light or touch.<br><br>At this level, students are required to test and make modifications to their solutions as they are developing it. Testing involves selecting specific functions/features of the solution to check that they operate as planned, for example, did a light go on when a specific button was pushed? | Evaluation differs from testing as it requires a judgement about how well the entire solution meets the functional requirements. This process happens once the solution has been created, whereas testing takes placing during the development of the solution.<br>When evaluating, students may assess their solutions on:<br>• how well they meet user needs<br>• how innovative their solution is compared to existing solutions<br>• how sustainable their solution will be for different users, purposes, and technology improvements<br>• how well they collaborated and managed the project. |

| | | | | |
|---|---|---|---|---|
| The focus of the learning (in simple terms) | Provide students with the opportunity to brainstorm a list of problems that can be solved digitally by creating a solution using resources available in the classroom that incorporate a microcontroller.<br>For the selected problem, students should state two or three features (requirements) that the solution must be able to perform. They also need to consider if there are any special user needs or technical requirements regarding the solution.<br><br>Discuss the 'Internet of Things' and the way in which devices around the home can be controlled via networked devices. Brainstorm solutions to problems that they can design a prototype to meet the need. For example, while away a plant needs to be watered, turning lights on and off to mimic being at home while being on holiday or an alarm system.<br><br>Students can negotiate their own projects to use embedded systems for example, projects that use Raspberry Pi / Lilypad Arduino, Arduino Nano / BBC:microbit or BlueBerry4 .<br><br>Provide an overview and walk through of the resources available for example if students have access to the Arduino Lily Pad discuss the input and outputs so that students are aware of its capabilities when designing their solutions. At a later stage they will need guidance as to the syntax used to program the board. | It is important that in their design, students consider the 'user experience' as well as the writing of instructions to operate the solution. For example, can output be shown in multiple ways such as sound and action or are the controllers of a suitable size to allow accessibility for people with special needs?<br><br>Provide opportunities for pairs of students to verbally or physically follow the algorithmic instructions of their partner. For example, for a robotic solution, a partner walks the route as stated in the algorithmic diagram or structured English – this allows any design errors to be located early in the problem-solving process.<br><br>Support and guide students with the design process, planning and project management. | Working from their designs, students build their robotic device from a commercially available robotic kit or from purpose-selected electronic equipment.<br><br>Support students to learn the syntax of the particular programming language required to code the programming board.<br><br>Introduce libraries of code and creation of functions / procedures. For example, Arduino has a list of Example sketches students can use to get started.<br><br>Students should develop a small testing table at the beginning of this process and carry out these tests. For example, the table could identify three functions that are going to be tested, state what result they expect to see and then the results when the functions were actually tested. If the results did not match, students should make modifications to their solution. They might need assistance in carrying out these modifications. | Organising or attending a culminating event where students showcase their projects is a great way to evaluate the digital solution and celebrate the learning.<br><br>Some innovative designs that are created by students to solve a problem can be taken to the 'pitch' stage of the problem-solving process. Have students create a 60 second video to pitch their solution similar to those featured on Kickstarter projects. They should tell how their solution meets specific requirements. |
| Supporting resources and tools and purpose/ context for use. | BBC Micro:Bit projects<br>Use these projects to inspire students to explore the functions available in the Micro:Bit. Once familiar with the functions run student-negotiated projects.<br><br>micro:bit Crash Course<br>Learn how to program a BBC micro:bit. No experience | LilyPad Personalised Alert Buzzer<br>This task provides an opportunity for students to create a digital solution that is interactive, programmable and related to a real world situation.<br><br>*BBC turns micro:bit computers into IoT devices*<br>*A prototype method for safely and securely turning* | Introduction to LilyPad & Arduino | LilyPad Light Up Soft Toy<br>This lesson compares student solutions with existing solutions as part of the evaluation.<br><br>Robocup Junior<br>Competition activities where robots must solve various problems. |

| | | | | |
|---|---|---|---|---|
| | required. Learn the basics of programming in Python with the BBC micro:bit simulator.<br><br>MBot / Hummingbird Electronics<br>Uses Scratch (block-based coding) but these boards can then be programmed using Arduino IDE (text-based).<br><br>DT Challenge 7/8 Arduino – Sound<br>Students learn to write code to create their own musical instrument using Arduino.<br><br>Picaxe<br>PICAXE chips can be programmed in a very simple to learn BASIC language or via graphical blocks or flowcharts. The programming language is designed to give you all the powerful features of the microcontroller without any complicated programming language to learn. | *BBC micro:bit into an internet of things (IoT) device.*<br><br>*Cheap Arduino Robot with everything you need*<br>*View Travis Burroughs' blog on how to create a cheap robot using electronic components.* | | Vex Robotics tournaments<br>Register for robotic tournaments. |
| Assessment | **Suggested approaches may include:**<br>List of two problems that could be solved using a programmable board with an explanation as to why.<br><br>List of specific technical devices needed for the solution (technical constraints).<br><br>Explanation as to why a programmable board solution would meet a social need.<br>**Achievement standard**<br>**Define** and **decompose** problems in terms of functional requirements and constraints. | **Suggested approaches may include:**<br>Recording (visual/sound) of a partner following the written/diagrammatic instructions for the solution.<br><br>Prototype of solution.<br>**Achievement standard**<br><br>**Design user experiences and algorithms incorporating branching and iterations**, and test, modify and implement digital solutions. | **Suggested approaches may include:**<br>Testing table.<br><br>Demonstration of solution.<br>**Achievement standard**<br><br>Design user experiences and algorithms incorporating branching and iterations, and **test**, **modify** and **implement digital solutions**. | Assessing students' work in robotics<br><br>**Suggested approaches may include:**<br>Recorded pitch of solution.<br><br>Three key evaluation questions that could be asked to determine the value of the solution.<br><br>**Achievement standard**<br><br>**Evaluate** information systems and their solutions in terms of meeting needs, innovation and sustainability. |