

## DIGITAL TECHNOLOGIES: DEVELOP AN ELECTRONIC ENVIRONMENT

Development of electronic environments focuses on the analysis of how electronic environments (functional combinations of hardware and embedded software in the real world i.e. circuits, prototypes or products) work in terms of their components, subsystems and software and how these components may be selected, subsystems put together and the hardware and software tested and debugged so that the electronic environment is functional with respect to agreed specifications. The model produced through these skills is a necessary precursor to developing a functional electronic and embedded system.

Initially, students learn basic functional modelling, circuit analysis, subsystem assembly and adjustment, testing and debugging skills. Students progress from this to more advanced skills to deal with more advanced and eventually complex environments. This progression will involve the introduction of more complex calculation and competency in the use and interpretation of data from devices such as multimeters (extended function), oscilloscopes and other test instruments. At the highest level, students will be able to analyse and develop complex electronic environments in terms of their subsystems and programming structures and employ mathematical calculations as part of this process.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	<i>Demonstrate ability to develop a basic electronic environment</i>	<i>Demonstrate ability to develop an advanced electronic environment</i>	<i>Demonstrate ability to develop a complex electronic environment</i>
TEACHER GUIDANCE	<p>To support students to demonstrate ability to develop a basic electronic environment at level 6, teachers could:</p> <ul style="list-style-type: none"> <li>Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some basic interfacing procedures.</li> <li>Provide experiences that involve functional modelling eg, breadboard simple circuits, use of simple simulation, kit-sets.</li> <li>Support students to analyse basic circuits (including those with a microcontroller) in terms of their subsystems (eg, voltage divider and transistor switch).</li> <li>Guide students to use functional modelling to perform measurements in, and to test, debug and make adjustments to voltage divider and transistor switch subsystems.</li> <li>Guide students to use functional modelling to develop software for a simple embedded system, given provided programme structures.</li> <li>Guide students to use functional modelling to test, debug and make adjustments to simple embedded software.</li> <li>Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller.</li> <li>Provide and explain a datasheet for a common component (eg, a resistor or LED).</li> <li>Provide opportunity for students to practice simple calculations based on selecting components for real circuits (eg, the value of a resistor to match an LED).</li> <li>Guide students to test and debug an integrated electronic environment to ensure functionality.</li> </ul>	<p>To support students to demonstrate ability to develop an advanced electronic environment at level 7, teachers could:</p> <ul style="list-style-type: none"> <li>Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some advanced interfacing procedures.</li> <li>Support students to analyse advanced circuits (those involving microcontrollers, timer/counter, amplifier stages etc) in terms of their subsystems.</li> <li>Provide functional modelling tools to enable students to perform measurements in, and to test, debug and make adjustments to advanced circuit subsystems.</li> <li>Guide students to use functional modelling to develop clearly annotated software (including variables, subroutines and an extended range of commands) for an advanced embedded system.</li> <li>Provide functional modelling tools to enable students to test, debug and make adjustments to advanced embedded software.</li> <li>Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller or other development environment such as Arduino, Raspberry Pi or tablet interface.</li> <li>Provide opportunity for students to practise interpreting datasheets and undertaking calculations based on real circuits, including voltage, current and power (eg, selecting components for a motor driver subsystem based on average and maximum power requirements).</li> <li>Support students to test and debug an integrated electronic environment to ensure functionality.</li> </ul>	<p>To support students to demonstrate ability to develop a complex electronic environment at level 8, teachers could:</p> <ul style="list-style-type: none"> <li>Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some complex interfacing procedures.</li> <li>Support students to analyse complex circuits (those involving FETs, npn and pnp transistors, SCRs, op-amps, LCD displays, servo and stepper motors etc.) in terms of their subsystems.</li> <li>Provide functional modelling tools to enable students to perform measurements in, and to test, debug, and make adjustments to complex circuit subsystems.</li> <li>Guide students to use functional modelling to develop clearly annotated, well-structured software (including communication protocols, macros, flags, interrupts, counters, bitwise AND/OR, pwm) for a complex embedded system.</li> <li>Provide functional modelling tools to enable students to test, debug and make adjustments to complex embedded software.</li> <li>Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller or other development environment.</li> <li>Guide students to perform complex calculations, including gain, RMS values and power, based on parameters important in the behaviour of real circuits.</li> <li>Support students to interpret datasheets and undertake calculations based on real circuits (eg, selecting component values and subsystem design so that circuits and data are matched to each other).</li> <li>Ensure students are able to test and debug the integrated electronic environment to ensure functionality.</li> </ul>
INDICATORS	<p>Students can:</p> <ul style="list-style-type: none"> <li>use datasheets or calculations to assist in choosing appropriate component types and values for the sensor and actuator subsystems</li> <li>write well-structured, clearly annotated, readily understandable interface software given simple programme structures</li> <li>interface subsystems to each other and to the embedded software in a microcontroller</li> <li>test and debug a functional model of the interface.</li> </ul>	<p>Students can:</p> <ul style="list-style-type: none"> <li>use provided functional sensor subsystems to interact with the environment</li> <li>use provided functional actuator subsystems to interact with the environment</li> <li>modify sensor subsystems to substantially improve the quality of the data delivered by the interface</li> <li>modify actuator subsystems to substantially improve the way they work</li> <li>write well-structured, clearly annotated, readily understandable software that interfaces effectively with the data provided by the sensors and with the actuators it controls</li> <li>interface subsystems to each other and to the embedded software in a microcontroller</li> <li>test and debug a functional model of the interface to achieve and demonstrate substantially improved operation.</li> </ul>	<p>Students can:</p> <ul style="list-style-type: none"> <li>devise and apply functional sensor subsystems that interact with the environment</li> <li>devise and apply functional actuator subsystems to interact with the environment</li> <li>analyse and modify sensor subsystems to substantially improve the quality of the data delivered by the interface</li> <li>analyse and modify actuator subsystems to substantially improve the way they work</li> <li>write well-structured, clearly annotated, readily understandable embedded software that interfaces effectively with the data provided by the sensors and with the actuators it controls</li> <li>interface subsystems to each other and to the embedded software in a microcontroller or other development environment.</li> <li>analyse, modify, test and debug a functional model of the interface to achieve and demonstrate substantially improved operation.</li> </ul>
AS	<p><b>AS91078 Digital Technologies 1.48</b> Implement basic interfacing procedures in a specified electronic environment</p>	<p><b>AS91375 Digital Technologies 2.48</b> Implement advanced interfacing procedures in a specified electronic environment</p>	<p><b>AS91639 Digital Technologies 3.48</b> Implement complex interfacing procedures in a specified electronic environment</p>
	<a href="#">Level 1 Digital Technologies standards &amp; assessment resources</a>	<a href="#">Level 2 Digital Technologies standards &amp; assessment resources</a>	<a href="#">Level 3 Digital Technologies standards &amp; assessment resources</a>