

---

## Overview

This standard identifies the competences you need to develop the specialist software which directly interacts with the hardware. These software functions can be provided singly (such as a specific driver for a particular piece of hardware), as a group of related software functions (such as a library of common drivers), or as part of an operating system, which provides a low level functionality to software applications (such as synchronisation, scheduling, memory management and file handling).

You will be given a detailed brief on the hardware and will be required to assess these requirements and to extract all necessary information in order to develop software, which exploits the capabilities of the hardware.

You will need to select the appropriate implementation method (such as machine code, assembler, hardware specific implementation) to use, based on the size and complexity of the software functions. You will be expected to use current British, European, international and company standards to produce the software.

Your responsibilities will require you to comply with organisational policy and procedures for software development. You will be required to report any problems with computer hardware, software or procedures that you cannot personally resolve, or that are outside your permitted authority, to the relevant people. You will be expected to work to verbal/written instructions and draft specifications, with a minimum of supervision, taking personal responsibility for your own actions and for the quality and accuracy of the work that you carry out.

Your underpinning knowledge will provide a good understanding of your work and will provide an informed approach to applying software development procedures. You will understand the computer system and software used, and its application, and will know about the various tools and techniques used to program, test and analyse the software under development, and to ensure that they are sufficient for the intended role within a defined operational environment.

You will understand the safety precautions required when working in the software development team. You will be required to demonstrate safe working practices throughout and will understand the responsibility you owe to yourself and others in the

workplace.

---

## Performance criteria

*You must be able to:*

1. work safely at all times, complying with health and safety legislation, regulations, directives and other relevant guidelines
2. plan and prepare the programming activities before you start them
3. use appropriate sources to obtain the required information for the hardware being programmed
4. select an appropriate implementation method
5. ensure the correct utilisation of the development tools being used to produce the software
6. produce the software
7. produce evidence of correct operation
8. save and store the evidence as the appropriate file type and in the correct location
9. review and report your findings on the developed software (such as defects, limitations on use, dependencies on other software/hardware)
10. deal promptly and effectively with problems within your control, and seek help and guidance from the relevant people if you have problems that you cannot resolve

## Knowledge and understanding

### *You need to know and understand:*

1. the specific safety precautions to be taken when working with software development environment hardware (to include such items as safety guidance relating to the use of visual display unit (VDU) equipment and workstation environment; repetitive strain injury (RSI); the dangers of trailing leads and cables; how to spot faulty or dangerous electrical leads, plugs and connections)
2. how to return the work area to a safe and useable condition (such as cleaning down work surfaces; putting media, manuals and unwanted items of equipment into safe storage; leaving the work area in a safe and tidy condition)
3. the relevant sources and methods for obtaining any required technical information relevant to the low level programming (such as hardware data sheets, specification sheets, request for changes or modifications to code; technical information)
4. specialist software development methods and tools used for low level programming (such as national, international and relevant company software development procedures, methods and tools)
5. identification of the correct version of software tool, and the various techniques that are supported by the tool
6. developing low level software
7. testing low level software
8. how to use and configure the tools
9. why it is necessary to be able to recall previous issues of test results
10. how to access, recognise and use a wide range of standard components and symbol libraries from the tools
11. the need for configuration control on all artefacts (such as ensuring that completed results are approved, labelled and stored on a suitable storage device)
12. when to act on your own initiative, and when to seek help and advice from others

## Scope/range

1. Prepare for software development, by carrying out all of the following:
  - 1.1 check that all software development environment hardware is in a safe, tested and usable working condition (such as cables undamaged, correctly connected, safely routed)
  - 1.2 identify the required standards and all relevant sources (such as Defence Standards, other sector specific software standards, software requirements, software design and code standards)
  - 1.3 identify the evidence required to allow software to be released to the users
  - 1.4 identify all test and diagnostic tools being used as part of the development (such as in-circuit-emulation (ICE), data-bus analysers and temporal execution time measurement devices)
2. Use three of the following to obtain the necessary data to program the low level code:
  - 2.1 software design
  - 2.2 specifications (hardware and software)
  - 2.3 change order/modification request
  - 2.4 regulations
  - 2.5 software safety requirements
  - 2.6 software development process definition documentation
  - 2.7 quality requirements
3. Carry out both of the following before commencing the low level programming:
  - 3.1 ensure that the data and information you have is current, complete and under configuration control
  - 3.2 check that all equipment being used is calibrated and tested for electrical safety, as per company standards and procedures
4. Produce and execute the software, using two of the following:
  - 4.1 target processor
  - 4.2 target emulator
  - 4.3 representative development environment
5. Test the software on two of the following:
  - 5.1 target processor
  - 5.2 target emulator
  - 5.3 representative development environment

Using two of the following:

4. performance analysis tool
5. in circuit emulation
6. hardware probes
7. instrumented code

1. Check the test results for all of the following:

- 1.1 completeness
- 1.2 traceability
- 1.3 accuracy
2. Save and store the results in appropriate locations, to include carrying out all of the following:
  - 2.1 ensure that your results have been checked and that it complies with your organisation procedure
  - 2.2 check that the results are correctly titled, referenced and annotated
  - 2.3 save the results to an appropriate location (such as storage device, configuration database)
  - 2.4 ensure a separate backup copy is created and placed in safe storage
  - 2.5 register and store the results in the appropriate company information system

SEMETS371

Performing low level programming for engineering software



---

<b>Developed by</b>	Enginuity
<b>Version Number</b>	3
<b>Date Approved</b>	30 Mar 2021
<b>Indicative Review Date</b>	01 Mar 2024
<b>Validity</b>	Current
<b>Status</b>	Original
<b>Originating Organisation</b>	Enginuity
<b>Original URN</b>	SEMETS371
<b>Relevant Occupations</b>	Engineering, Engineering and Manufacturing Technologies, Engineering Technicians
<b>Suite</b>	Engineering Technical Support Suite 3, Advanced Manufacturing
<b>Keywords</b>	engineering; technical; support; software design; specifications; regulations; target processor

---