Information and Communications Technology Industry Training Advisory Committee
Software Products and Software Services (SW) branch
Unit of Competencies

1. Title	Understand key characteristics of embedded software systems
2. Code	ITSWAR521A
3. Range	Demonstrate clear understanding of the key characteristics of embedded software systems through the application of suitable principles in the design of such software systems [Architecture – Embedded Software Architecture]
4. Level	5
5. Credit	3
6. Competency	Performance Requirement 6.1 Understand key characteristics of embedded software systems — Be able to explain key characteristics of embedded software systems and their effects on the design of embedded software systems See Remark 1 for some examples of key characteristics of embedded software systems 6.2 Understand the overall design requirements of embedded software systems — Be able to explain the overall design requirements of embedded software systems See Remark 2 for some examples of design requirements of embedded software systems 6.3 Understand the design principles applicable to embedded software systems — Be able to explain the various design principles and how they are applicable to embedded software systems 6.4 Apply the suitable principles in the design of embedded software systems — Be able to use the suitable design principles in the design of embedded software system 6.5 Design the embedded software system in a professional way — Be able to demonstrate high degree of professionalism and competence in the design of embedded software systems
7. Assessment Criteria	The integrated outcome requirement of this UoCs is the ability to apply suitable principles professionally in designing embedded software systems.
Remark	1. Some examples of key characteristics of embedded software systems are a) high dependence on hardware platform; b) small software footprints and code efficiency; c) many are real-time in nature; d) purpose-built for very specific problems; e) high reliability and robustness (self diagnostics and self correction are common requirements); and f) extremely high software quality (as distribution of software corrections after production release can be prohibitively costly). 2. The overall design requirements of embedded system might include: a) purpose of the system; b) input and output specifications; c) performance criteria; d) usage patterns and environment; e) usability requirements and constraints; f) hardware platform; and g) physical constraints and footprint.