Unit 46: Embedded Systems

Unit code	**A/615/1514**
Unit level	**5**
Credit value	**15**

Introduction

An embedded system is a device or product which contains one or more tiny computers hidden inside it. This 'hidden computer', usually a microcontroller, is used to control the device and give it added 'intelligence'. Embedded systems are a key aspect of modern engineering and are applied in areas as diverse as automotive, medical, and industrial, and in the home and office. In many cases, embedded systems are linked together in networks. Embedded systems are the basis of a new wave of engineering design and practice, notably in machine-to-machine communication and in the Internet of Things.

This unit builds on introductory knowledge students have already gained in electronic circuits. It develops their knowledge of computer hardware, focussing on the small, low-cost type of computer (i.e. a *microcontroller*), usually used in embedded systems. It then develops skill in devising circuits which operate external to the microcontroller and interface with it; generally, these relate to sensors, actuators, human interface or data transfer. In parallel with this, students will be developing programming skills, writing programmes which download straight to the microcontroller and cause it to interact with its external circuit. Students will also explore the wider context of embedded systems, learning how they are applied in 'hi-tech' applications, in many cases revolutionising our ability to undertake certain activities.

Unit assessment will require the design, development, construction and commissioning of an embedded system, meeting a given design brief; this will develop skills which are in much demand in industry. A written assignment, exploring one or more of the many fast-moving embedded system applications in use today, will also be completed.

Learning Outcomes

By the end of this unit students will be able to:

1. Explore the principle features of a microcontroller and explain the purpose of its constituent parts.
2. Design and implement simple external circuitry, interfacing with a given microcontroller.
3. Write well-structured code in an appropriate programming language, to simulate, test and debug it.
4. Evaluate the applications of embedded systems in the wider environment, including in networked systems.

Essential Content

LO1 Explore the principle features of a microcontroller and explain the purpose of its constituent parts

Microcontroller architecture:

CPU (Central Processing Unit), the instruction set, programme memory, data memory, input/output (I/O), data and address buses, van Neumann and Harvard structures.

Peripherals, to include digital I/O, counter/timers, analogue to digital converter (ADC), pulse width modulation (PWM), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver/Transmitter (UART).

Memory types (overview only): Flash, Static RAM (Random Access Memory), EEPROM (Electrically Erasable Read Only Memory) and their applications.

Simple interrupt concepts.

LO2 Design and implement simple external circuitry, interfacing with a given microcontroller

Simple digital interfacing:

Switches, light emitting diodes (LEDs), keypads, and 7-segment displays.

DC and ADC applications:

DC load switching (e.g. of small motor or solenoid), use of PWM to provide variable DC motor speed control.

ADC application, including range and resolution.

Signal conditioning for analogue inputs, including simple op amp circuits to provide gain or level shifting.

Interfacing to external devices with serial capability, applying SPI and UART.

Power supply and clock oscillator.

LO3 Write well-structured code in an appropriate programming language, to simulate, test and debug it

The development cycle:

Integrated Development Environment, Assembler and High Level Languages, compilers, simulators, completing an in-circuit debug.

Devising a code structure e.g. using flow diagrams and pseudo code.

Programming languages and codes:

Review of an appropriate high level programming language (which is likely to be C). Language structure, data types, programme flow, looping, branching, and conditional.

Developing application code: initialisation, data input, conditional branching and looping, data output.

Code simulation, download, test and debug.

LO4 **Evaluate the applications of embedded systems in the wider environment, including in networked systems**

Review of application of embedded systems:

Using example sectors e.g. motor vehicle, smart buildings, medical, office, wearable. Review possible limiting factors in an embedded design e.g. power supply, reliability, security.

Review of current trends in embedded systems, including the Internet of Things and machine-to-machine.

Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
LO1 Explore the principle features of a microcontroller and explain the purpose of its constituent parts		**D1** Critically evaluate microcontroller architectures and subsystems, exploring characteristics such as electrical, timing and size (e.g. of memory or ALU).
P1 Examine the hardware interfaces and the software architecture of a selected microcontroller. **P2** Explain the function of the main microcontroller elements.	**M1** Evaluate microcontroller architectures and subsystems, exploring characteristics such as electrical, timing and size (e.g. of memory or ALU).	
LO2 Design and implement simple external circuitry, interfacing with a given microcontroller		**D2** Critically evaluate the functionality of external circuitry under a range of operating conditions.
P3 Design simple external circuits, sensors and actuators, from available designs. **P4** Apply simple external circuits, demonstrating effective interfacing and adequate functionality.	**M2** Adapt and improve simple external circuits, sensors and actuators, from available designs. **M3** Assess simple external circuits and evaluate functionality.	
LO3 Write well-structured code in an appropriate programming language, to simulate, test and debug it		**D3** Critically evaluate the code developed through simulation and in the hardware, demonstrating excellent functionality.
P5 Write well-structured working code, to meet an identified need. **P6** Test and de-bug code through simulation in the hardware, demonstrating functionality.	**M4** Adapt and improve given examples to produce well-structured and reliable code with meaningful programme identifiers, to meet an identified need.	
LO4 Evaluate the applications of embedded systems in the wider environment, including in networked systems		
P7 Evaluate current and emerging applications of embedded systems, e.g. in motor vehicles, health, or the Internet of Things.	**M5** Critically evaluate emerging applications of embedded systems, clearly identifying trends and recognising technical and economic factors.	

Recommended Resources

Textbooks

BLUM, J. (2013) *Exploring Arduino*. Wiley.

TOULSON, R. and WILMSHURST, T. (2012) *Fast and Effective Embedded System Design: Applying the ARM*. Newnes.

WILMSHURST, T. (2009) *Designing Embedded Systems with PIC Microcontrollers: Principles and Applications*. 2nd Ed. Newnes.

Links

This unit links to the following related units:

Unit 52: Further Electrical, Electronic and Digital Principles

Unit 54: Fundamentals of Control Systems