

Liverpool John Moores University

Title: DISTRIBUTED AND EMBEDDED SYSTEMS
Status: Definitive
Code: **6131COMP** (121308)
Version Start Date: 01-08-2021

Owning School/Faculty: Computer Science and Mathematics
Teaching School/Faculty: Computer Science and Mathematics

Team	Leader
David Lamb	Y

Academic Level: FHEQ6 **Credit Value:** 20 **Total Delivered Hours:** 57
Total Learning Hours: 200 **Private Study:** 143

Delivery Options

Course typically offered: Semester 2

Component	Contact Hours
Lecture	11
Workshop	44

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	A distributed software system	50	
Exam	AS2	Examination	50	2

Aims

To provide an overview of designing and engineering both distributed software systems and embedded systems with references to architectures, communication and synchronisation. The practical focus of this module is on developing software with concurrent and distributed components and developing for embedded systems.

Learning Outcomes

After completing the module the student should be able to:

- 1 Survey technologies and methods used in embedded systems design and development
- 2 Solve execution issues intrinsic to embedded architectures and develop software for embedded systems alongside other software systems
- 3 Appraise communications standards and techniques used in embedded systems
- 4 Critically evaluate operational issues in embedded, concurrent and distributed systems

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Software System and Report	1	2
Examination	3	4

Outline Syllabus

Monolithic vs Distributed Systems

Distribution vs Parallelisation

-Message Passing and Shared Memory

Concurrency and Scheduling in Software Systems

-Liveness, Races and Deadlocks

-Ensuring deterministic behaviour

-Shared resource access; synchronisation, mutual exclusion, atomicity

System architecture for embedded systems

-Microprocessor (modified) Harvard vs. Von Neumann architectures

-Programmable Controllers and Microcontroller (MCU) architectures

-Watchdogs

-Memory and memory units (EPROM, EEPROM, RAM, FLASH)

Software architectures for embedded systems

-Superloop and RTOS

-Managing I/O and interrupts

Communications standards used in embedded systems

-Shared medium/bus systems

-Dedicated line, clock synchronisation, A/D conversion

-Deterministic arbitration and access protocols

-Dedicated hardware and emulated ("bit-banged") software-defined implementations

Learning Activities

Students will participate mostly in workshop sessions with a lecture/taught element.

Notes

A largely practical software engineering course that will equip students with skills to work in distributed and concurrent software systems. It will also provide experience of development for embedded systems with consideration of relevant basic interfacing electronics concerns.