# COMP8291

## IoT Embedded Operating Systems and Application Programming

Session 2, In person-scheduled-weekday, North Ryde 2026

*School of Computing*

## Contents

# General Information

Unit convenor and teaching staff
Convenor, Lecturer
Yimeng Feng
yimeng.feng@mq.edu.au
Contact via by email yimeng.feng@mq.edu.au
217, 4RPD
By appointment

Credit points
10

Prerequisites
COMP6291

Corequisites

Co-badged status

Unit description
This unit is intended to provide a practical understanding of embedded/real-time operating systems (OS). The unit investigates the Embedded OS internals by examining key concepts such as kernel architecture and threads, device models, interfaces and drivers, system calls, kernel, filesystem, and system utilities. It also provides students with the knowledge and skills to begin developing and implementing IoT applications.

## Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

# Learning Outcomes

On successful completion of this unit, you will be able to:

**ULO1:** Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.

**ULO2:** Program, performance test, and debug embedded software and IoT application programs.

**ULO3:** Demonstrate an ability to process data fed from sensors and be able to program

using that data.

**ULO4:** Incorporate security protections into the embedded OS and/or Application.

**ULO5:** Apply methodologies and tools to design and develop software modules using a wide range of IoT application platforms.

**ULO6:** Analyse and assess different IoT application platforms.

# General Assessment Information

**General Notes**

Students will engage in a blended learning approach that combines theoretical study, software development, and collaborative discussions during lectures. Workshop sessions will provide opportunities for applying and validating skills in theory, software implementation, and problem-solving. In this unit, students are expected to:

1. Actively participate in lectures by attending, taking comprehensive notes, and engaging in discussions.
2. Attend your workshop or practical sessions, and actively seek feedback from your lecturer regarding your assignments.
3. Engage with the assigned readings, supplementing your notes and formulating questions for further clarification during discussions with your lecturer or SGTA.
4. Thoughtfully craft responses to workshop questions, demonstrating a deep understanding of the material.
5. Dedicate time to complete any assigned tasks or projects within the given timeframe.

**Attendance and Participation Guidelines**

It is strongly advised to attend the majority of classes, actively participate in discussions, ask and answer questions, and share perspectives from your background and workplace. Supplementary resources, including lecture materials and digital recordings of lectures, are available for review purposes and in the event of missed classes. These resources can be accessed through Echo360 via an iLearn login. However, it is essential to note that these recordings should not be solely relied upon, and copyrighted material may be excluded. For effective out-of-class communication and engaging in discussions on various topics, iLearn is the primary platform. Accessible through iLearn, the platform offers forums for active participation and serves as a valuable resource for disseminating information. Regularly reviewing iLearn and conducting background reading before each class is highly encouraged to enhance your overall learning experience.

# Assignment Submission

Your assignment is to be submitted online using iLearn.

## Release Dates

- Assignment 1 (Data Processing in Real-Time Operating Systems): To be released no later than 10th August.
- Assignment 2 (Applications of Real-Time Operating Systems): To be released no later than 14th September.

## Late Assessment Submission Penalty

Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark of the task) will be applied for each day a written report or presentation assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is **11:55 pm**. A 1-hour grace period will be provided to students who experience a technical concern. For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration.

## Assessments where Late Submissions will be accepted

- Assignment 1 (Applications of Real-Time Operating Systems) - YES, Standard Late Penalty applies
- Assignment 2 (Data Processing in Real-Time Operating Systems) - YES, Standard Late Penalty applies
- Lab Report - YES, Standard Late Penalty applies

## Supplementary Exam

If you receive Special Consideration for the final exam, a supplementary exam will be scheduled after the normal exam period, following the release of marks. By making a special consideration application for the final exam, you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

## Requirements to Pass this Unit

To pass this unit, you must achieve a total mark equal or greater than 50%. This unit does not have hurdle assessments.

## Special Consideration

The Special Consideration Policy aims to support students who have been impacted by short-term circumstances or events that are serious, unavoidable, and significantly disruptive, and which may affect their performance in assessment. If you experience circumstances or events

that affect your ability to complete the assessments in this unit on time, please inform the convenor and submit a Special Consideration request through connect.mq.edu.au

## Data Processing in Real-Time Operating Systems

Assessment Type [1]: Programming Task Indicative Time on Task [2]: 37 hours Due: **Due Monday 14th September 23:55** Weighting: **40%** Groupwork/Individual: Short extension [3]: No AI Approach:

Students will develop a program that reads, processes, and transmits sensor data using an RTOS, focusing on real-time multitasking, scheduling, and memory management in a resource-constrained IoT environment.  A technical report documenting implementation, testing, and performance evaluation is required.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Apply methodologies and tools to design and develop software modules using a wide range of IoT application platforms.
- Analyse and assess different IoT application platforms.

## Applications of Real-Time Operating Systems

Assessment Type [1]: Project Indicative Time on Task [2]: 37 hours Due: **Due Monday 26th October 23:55** Weighting: **40%** Groupwork/Individual: Short extension [3]: No AI Approach:

Students will design and develop an IoT prototype using an RTOS, demonstrating task scheduling, inter-process communication, real-time data processing, and resource management. A technical report documenting the design, implementation, and performance evaluation is required.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application

programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Incorporate security protections into the embedded OS and/or Application.

## Lab Report

Assessment Type [1]: Practice-based task Indicative Time on Task [2]: 24 hours Due: **Due Monday 2nd November 23:55** Weighting: **20%** Groupwork/Individual: Short extension [3]: No AI Approach:

This lab offers hands-on experience with embedded systems and RTOS. Students will submit a report based on the lab work completed at the end of the session.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Apply methodologies and tools to design and develop software modules using a wide range of IoT application platforms.
- Analyse and assess different IoT application platforms.

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- Academic Success for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation.

[3] An automatic short extension is available for some assessments. Apply through the Service Connect Portal.

# Assessment Tasks

| Name | Weighting | Hurdle | Due | Groupwork/Individual | Short Extension | AI Approach |
|---|---|---|---|---|---|---|
| Data Processing in Real-Time Operating Systems | 40% | No | 14/09/2026 | Individual | Yes | Open |
| Applications of Real-Time Operating Systems | 40% | No | 26/10/2026 | Individual | Yes | Open |
| Lab Report | 20% | No | 02/11/2026 | Individual | Yes | Open |

## Data Processing in Real-Time Operating Systems

Assessment Type [1]: Experiential task
Indicative Time on Task [2]: 37 hours
Due: **14/09/2026**
Weighting: **40%**
Groupwork/Individual: Individual
Short extension [3]: Yes
AI Approach: Open

You will develop a program that reads, processes, and transmits sensor data using an RTOS, focusing on real-time multitasking, scheduling, and memory management in a resource-constrained IoT environment.  A technical report documenting implementation, testing, and performance evaluation is required.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Apply methodologies and tools to design and develop software modules using a wide range of IoT application platforms.
- Analyse and assess different IoT application platforms.

## Applications of Real-Time Operating Systems

Assessment Type [1]: Portfolio

Indicative Time on Task [2]: 37 hours
Due: **26/10/2026**
Weighting: **40%**
Groupwork/Individual: Individual
Short extension [3]: Yes
AI Approach: Open

You will design and develop an IoT prototype using an RTOS, demonstrating task scheduling, inter-process communication, real-time data processing, and resource management. A technical report documenting the design, implementation, and performance evaluation is required.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Incorporate security protections into the embedded OS and/or Application.

## Lab Report

Assessment Type [1]: Practice-based task
Indicative Time on Task [2]: 24 hours
Due: **02/11/2026**
Weighting: **20%**
Groupwork/Individual: Individual
Short extension [3]: Yes
AI Approach: Open

This lab offers hands-on experience with embedded systems and RTOS. You will submit a report based on the lab work completed at the end of the session.

On successful completion you will be able to:

- Demonstrate knowledge of Real Time Embedded OS architecture and build, maintain, and Install and troubleshoot simple software for real time IoT based embedded systems.
- Program, performance test, and debug embedded software and IoT application programs.
- Demonstrate an ability to process data fed from sensors and be able to program using that data.
- Apply methodologies and tools to design and develop software modules using a wide range of IoT application platforms.

- Analyse and assess different IoT application platforms.

---

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- Academic Success for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation.

[3] An automatic short extension is available for some assessments. Apply through the Service Connect Portal.

# Delivery and Resources

**Delivery Time**

COMP8291 is taught via lectures and workshop sessions. Lectures are scheduled on Thursday from 9:00 AM to 11:00 AM, followed by workshops (information will be announced in class and in iLearn). This attendance policy underscores the interactive and real-time nature of the learning experience, encouraging students to actively participate in lectures and workshops for an optimal educational outcome.

**Week 1 Classes**

In Week 1, only the lecture session will take place, covering Introduction to Real-Time Operating Systems. Workshops will commence from Week 2.

**Lectures**

The lecture sessions for COMP8291 are dedicated to exploring real-time operating systems (RTOS) through a blend of theoretical concepts and hands-on applications, particularly utilizing the STM32 microcontroller. This approach aims to provide students with a comprehensive understanding of the internal workings and design principles of RTOS, with a focus on their practical implementation in building embedded systems for the Internet of Things (IoT).

Various RTOS simulation software tools, including STM32 Cube and FreeRTOS, will be employed throughout the unit to support the learning process. Lecture sessions will lay the foundation for grasping RTOS intricacies, and this knowledge will be put into practice during workshops. The workshops will involve running and porting the FreeRTOS Real-Time Operating System on STM32-based Microcontrollers.

While lecture notes will be provided weekly, viewing them as a guide rather than a substitute for personal notes or the recommended reading list is essential. Students are encouraged to engage with the material actively, supplementing their understanding through the provided lecture notes, personal notes, and the suggested reading list for each week. This proactive approach will contribute to a robust comprehension of the syllabus topics and enhance the overall learning experience in real-time operating systems.

### Assignments

Students are allotted a four-week timeframe to complete each assignment, emphasizing a structured and thorough approach to their work. Assignments 1 and 2 contribute 40% of each assignment to the overall course grade. Evaluation criteria for assignments encompass three key components: First, the complexity of the designed programming application will be assessed to determine the level of innovation and technical challenge demonstrated by the student. Second, the student's understanding of the assignment will be evaluated through their presentation, focusing on how clearly and effectively they communicate their approach and insights. Finally, comprehension of the Real-Time Operating System (RTOS) concept will be assessed through the project report, which should follow the provided template to ensure consistency and depth in the explanation. The completed report is then to be submitted through the designated system. Additionally, one week after the submission deadline, you will have the opportunity to present your work. This presentation is a platform to showcase your understanding of the material and the practical applications of RTOS programming concepts.

Assignment 1 is due by 11:55 PM on Monday, 14th September (the beginning of Week 8), while Assignment 2 is due by 11:55 PM on Monday, 26th October (the beginning of Week 12). Both assignments involve tackling real-world problems through the lens of Real-Time Operating System (RTOS) programming. Throughout these assignments, you will delve into problem-solving cases around RTOS programming. The objective is to enable you to apply the theoretical concepts you've learned using one of the widely recognized commercial open-source RTOS platforms.

### Workshop

The workshop sessions will be conducted onsite over 12 meetings, running from Week 2 to Week 13. From Week 2 to Week 11, students will focus on completing practical modules designed to build their skills in embedded systems and Real-Time Operating Systems (RTOS). Weeks 12 and 13 are reserved for make-up sessions, which are available only to students who were unable to attend earlier workshops. Each student may attend a maximum of two make-up classes. These sessions offer valuable hands-on experience, and students are required to submit a report summarizing the lab work completed at the end of the session. The workshop component accounts for 20% of the overall assessment.

### Course Reading Material Information

Please be aware that there isn't a single textbook covering all the content for this unit. However, your lecturer will provide comprehensive reading materials and detailed notes corresponding to each week's lecture topics. These resources will be provided every week for your convenience. It's essential to note that purchasing these books is not obligatory; their inclusion is based on individual preferences and needs. Students are encouraged to evaluate their requirements and determine whether acquiring these supplementary resources aligns with their learning preferences and objectives.

1. Jim Cooling, "Real-time Operating Systems Book 1," ISBN: 1795340657
2. Jim Cooling, "Real-time Operating Systems Book 2 - The Practice: Using STM Cube, FreeRTOS and the STM32 Discovery Board: 1," ISBN: 1973409933

**Methods of Communication:** Our primary means of communication will be through your university email and announcements on iLearn. It is crucial to consistently check your university email for important updates and information related to the course. Additionally, significant announcements will be posted on iLearn, a centralized platform for accessing vital details about the course. Should you have any queries or require assistance from the teaching staff, including the unit convenor, you have two communication channels. Firstly, you can post your questions on the iLearn discussion board, providing an interactive space for instructors and peers to engage in discussions. Alternatively, you may send emails to the corresponding addresses of the teaching staff using your university email address for official communication. Through these communication methods, we aim to ensure effective and timely dissemination of information and provide the necessary support throughout the course.

# Unit Schedule

Unit Schedule

| Week | Topic | Learning Outcome |
|---|---|---|
| 1 | Introduction to IoT Embedded Operating Systems | UL01, UL02 |
| 2 | Scheduling - Free RTOS Scheduler (Task Priorities, queues, and Thread management) | UL01, UL02 |
| 3 | FreeRTOS Semaphores (Mutex and Counting Semaphores) | UL02, UL03 |
| 4 | Concurrency (Priority, task blocking, and protocol) | UL02, UL03 |
| 5 | Intertask communication | UL03, UL04 |
| 6 | Memory Usage and Management | UL02, UL03, UL04 |
| 7 | Applications of Real-Time Operating Systems | UL02, UL03, UL04 |
| 8 | Multiprocessor systems (Partitioning and allocation) part 1 | UL01-UL04 |
| 9 | Multiprocessor systems (Partitioning and allocation) part 2 | UL01, UL02 |
| 10 | Distributed Systems | UL03, UL04 |
| 11 | Analysis and review of scheduling policies | UL03, UL04 |
| 12 | Performance, and benchmarking of RTOSs | UL03, UL04, UL05 |
| 13 | IoT applications and Unit review | UL05, UL06 |

# Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://policies.mq.edu.au). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy
- Fitness to Practice Procedure
- Assessment Procedure
- Complaints Resolution Procedure for Students and Members of the Public
- Special Consideration Policy

Students seeking more policy resources can visit Student Policies (https://students.mq.edu.au/support/study/policies). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

To find other policies relating to Teaching and Learning, visit Policy Central (https://policies.mq.edu.au) and use the search tool.

## Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/admin/other-resources/student-conduct

## Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit connect.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

# Academic Integrity

At Macquarie, we believe academic integrity – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a range of resources and services to help you reach your potential, including free online writing and maths support, academic skills development and wellbeing consultations.

# Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

## Academic Success

Academic Success provides resources to develop your English language proficiency, academic writing, and communication skills.

- Ask a Study Coach
- Access StudyWISE

- Upload an assignment to Studiosity
- Complete the Academic Integrity Module

The Library provides online and face to face support to help you find and use relevant information resources.

- Subject and Research Guides
- Ask a Librarian

## Student Services and Support

Macquarie University offers a range of Student Support Services including:

- IT Support
- Accessibility and disability support with study
- Mental health support
- Safety support to respond to bullying, harassment, sexual harassment and sexual assault
- Social support including information about finances, tenancy and legal issues
- Student Advocacy provides independent advice on MQ policies, procedures, and processes

## Student Enquiries

Got a question? Ask us via the Service Connect Portal, or contact Service Connect.

## IT Help

For help with University computer systems and technology, visit https://students.mq.edu.au/support/technology/service-desk.

When using the University's IT, you must adhere to the Acceptable Use of IT Resources Policy. The policy applies to all who connect to the MQ network including students.

# Changes from Previous Offering

We value student feedback to be able to continually improve the way we offer our units. As such we encourage students to provide constructive feedback via student surveys, to the teaching staff directly, or via the FSE Student Experience & Feedback link in the iLearn page.

Student feedback from the previous offering of this unit was very positive overall, with students pleased with the clarity around assessment requirements and the level of support from teaching staff. As such, no change to the delivery of the unit is planned, however we will continue to strive to improve the level of support and the level of student engagement.

---

Unit information based on version 2026.03 of the Handbook