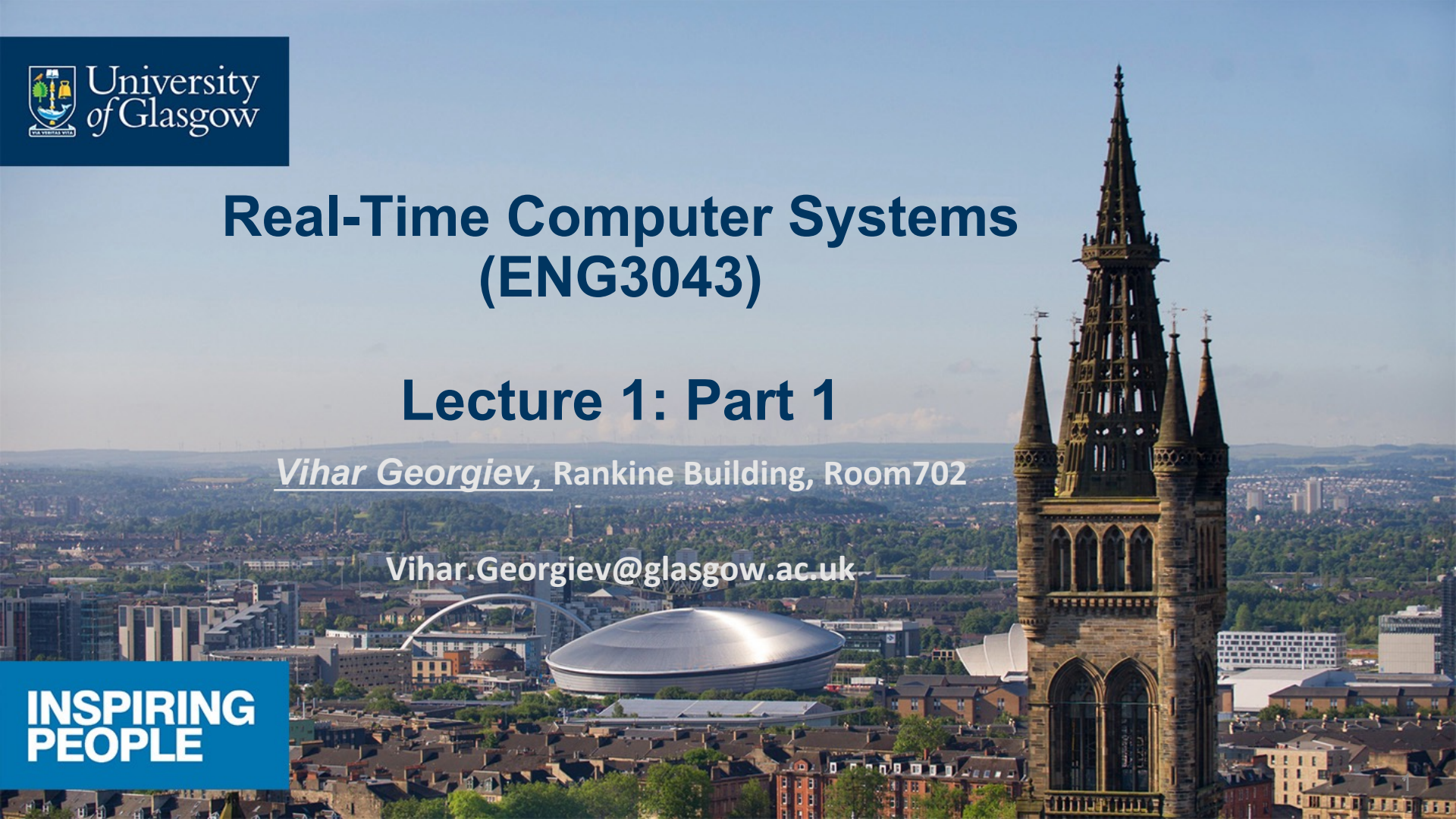

University of Glasgow
Real-Time Computer Systems (ENG3043)
Lecture 1: Part 1
Vihar Georgiev, Rankine Building, Room702
Vihar.Georgiev@glasgow.ac.uk
INSPIRING PEOPLE

Lecture 1: Part 1

Outline

- Introduction

University of Glasgow

Introduction

Aims

In this course you will study the following

- ARM Cortex M series *architectures* as examples of real-time systems.
- Programming in assembly language and high-level language, C / C++.
- Timers and interrupts. The implementation of interrupts.
- Operating systems - both real-time and non real-time. The interaction between software and hardware.

Intended learning outcomes (ILOs)

By the end of this course, you will be able to

- Describe the attributes of a real-time computer system;
- Understand how high-level codes written in C/C++ are broken into its component machine code instructions;
- Implement interrupts as a method of acquiring input from the physical world;
- Appraise the use of real-time operating systems and describe priority allocation policies such as rate monotonic scheduling.

Course arrangement

- 18 in-person lectures + 2 Tutorials (Tuesday 13:00 & Friday 10:00)
- Last week (28th Nov and 1st Dec) – Revision Week (no lectures)
- 3 Laboratories (Rankine Level 3 – Room 317):
  - Lab 1: End October
  - Lab 2: Middle November
  - Lab 3: End November
- Assessment
  - Summative
    - Written exam: 85%
    - Laboratory: 15%
- Minimum Requirement for Award of Credits

  Attend the final examination

  Attend and complete ALL laboratories

University of Glasgow

Lecture topics

Topic 1 – Course Introduction

Topic 2 – Microprocessor Basics

Topic 3 – Assembly language and C/C++

Topic 4 – Bus Architecture and GPIOs

Topic 5 – Timers and interrupts

Topic 6 – Computer Architecture and Operating Systems

Laboratories (15%)

- Lab 1. Students will exercise programming in C and assembly language in Keil IDE and understand how registers and memory cooperate when running a simple C programme.
- Lab 2. Students will work on a programme that is designed for better understanding interrupts
- Lab 3. Students will work on a programme that is designed for better understanding threads and how a RTOS deals with context switching

Assessments

- Report submission in Moodle

About the Labs

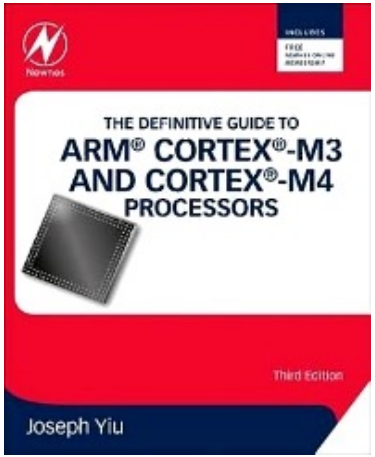
- STM32F4 Discovery Board or similar will be used throughout of these laboratories
- The labs are in person
- Lab reports will need to be submitted by the end of the semester

DEAD LINE: 4th December 2023

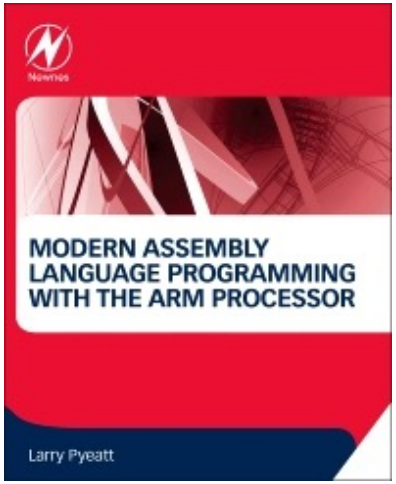
Reading list and books

Reding list is on MOODLE and link here

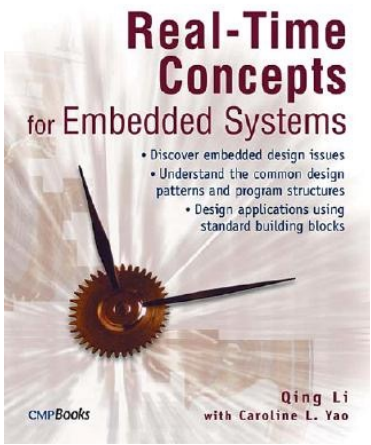
The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors 3rd ed
Joseph Yiu



Modern Assembly Language Programming with the ARM Processor 1st Edition
Larry Pyeatt



Real-Time Concepts for Embedded Systems
Li Qing , Yao Carolyn



Points to remember

- The difference between soft-deadline and hard-deadline
- Real-time systems are characterized by the fact that timing correctness is just as important as functional or logical correctness.
- The severity of the penalty incurred for not satisfying timing constraints differentiates hard real-time systems from soft real-time systems.
- Real-time systems have a significant amount of application awareness similar to embedded systems.
- Real-time embedded systems are those embedded system with real-time behaviours.
- An embedded system is built for a specific application. As such, the hardware and software components are highly integrated.
- Embedded systems are generally built using embedded processors.

End of the recording