Motivation

Software most complex component of critical systems

Overview of the Week

Monday
- Introduction
- Micro-architectures
  - CPUs, busses, memory, caches
- Assembler
  - x86 vs. MIPS, ARM

Tuesday
- Basics of C
  - language
  - bit-vector semantics
  - functions
  - pointers, structs, data structures
  - compilation units and modularisation
  - memory-mapped I/O, interrupts, threads
Wednesday

- C++
  - classes and objects
  - inheritance
  - exceptions and resources
  - templates
- IEEE floating point
  - representation of numbers and rounding
  - compiler support in C
  - common mistakes and problems
  - basic numerical recipes
- Tooling
  - git, subversion
  - unit testing, regression testing
  - coverage metrics and safety standards, MISRA-C

Thursday

- Labview
- Done by Alessandro Abate

Friday

- Unified Modeling Language (UML)
  - design spiral
  - behavioural diagrams
  - class diagrams

What?

Prerequisites
- Basic imperative programming
- Basics of computing hardware

Learning Outcomes
At the end of the course students will:
- Be able to undertake basic programming assignments
- Self-instruct further techniques and details, as needed

Reading

- The C++ Programming Language
  Bjarno Stroustrup
- Programming for Engineers: A Foundational Approach to Learning C and Matlab
  Aaron R. Bradley
Schedule

Lectures: MT week 6 Mo–Fr 10–12, LR7
Labs: MT week 6 Mo–Fr 14–16

Labs

- Run by Pascal Kesseli
- DPhil in CS
- He has emailed you with installation instructions

Assessment

- Assessment is by report
- Due Monday week 7
- Primarily cover what you have done, i.e., your labs, and the lab sheets
- You can skip the trivial stuff
- You can skip Thursday's stuff