# Control systems and Computer Networks

## Module Overview

Dr Alun Moon

Lecture 0.1

# What will I learn?

On this module you will learn about computer networks and networked embedded control systems. Many systems now use a network of small computers and devices, from modern cars to the Internet of Things. This module looks at the interaction between devices (sensors and actuators) and the controlling computer. The various networking and communications technologies that connect these systems together, their infrastructure and protocols, will be discussed while considering cybersecurity architecture and operations as appropriate.

*from the Module Specification*

# What will I learn?

You will learn how to control hardware attached to computers. This involves understanding the basics of the electronic circuits used in attaching the hardware, the features of the processors that interact with hardware and the writing of device drivers (the code that controls the devices). You will learn about microprocessor organisation with respect to Input / Output (I/O) and software development for networked control systems using an appropriate language.

*from the Module Specification*

# Teaching Team

| | |
|---|---|
| Module Tutor | Dr Alun Moon `alun.moon@northumbria.ac.uk` |
| | Dr Michael Brockway |
| | Dr David Kendall |
| | Dr. Nanlin Jin |
| | R. Dinakaran |
| | Kezhi Wang |

# Teaching Structure

**Lectures** $2 \times 1$ h lectures

- roughly split into 2 interweaving streams
- will try to keep step with labs (no promises)

**Labs/Seminars** $1 \times 2$ h sessions

- Main practical work
- Largely self-contained

**Directed Study** Where appropriate I'll point you in the direction of resources to read

**Independent Learning** A further 9 h

# Teaching Plan I

| | | |
|---|---|---|
| Week 1 | Seminar | Digital I/O |
| | Lecture 1 | Introduction and Overview |
| | Lecture 2 | Digital Signals |
| Week 2 | Seminar | Events and interrupts |
| | Lecture 1 | Memory mapped I/O |
| | Lecture 2 | Discrete time & Interrupts |
| Week 3 | Seminar | Analogue I/O |
| | Lecture 1 | I2C, SPI, CAN, busses |
| | Lecture 2 | Analogue Signals |
| Week 4 | Seminar | LCD & serial comms |
| | Lecture 1 | Display code |
| | Lecture 2 | Sensors |
| Week 5 | Seminar | Control & feedback |
| | Lecture 1 | Control of systems |

# Teaching Plan II

| | | |
|---|---|---|
| | Lecture 2 | Finite state machines |
| Week 6 | Seminar | Assembler and ABI |
| | Lecture 1 | Sampling & Nyquist limit |
| | Lecture 2 | ARM architecture and assembler, ABI, |
| Week 7 | Seminar | Data logging |
| | Lecture 1 | Formats, XML, Json, Yaml |
| | Lecture 2 | TCP stack, RFC1122, RFC1123, RFC768 |
| Week 8 | Seminar | ? |
| | Lecture 1 | Server logging and analysis |
| | Lecture 2 | Application layer, RFC2616, RFC7540, RFC1123 |
| Week 9 | Seminar | ? |
| | Lecture 1 | Authentication, validity, md5 hashes, sha/NIST FIPS 180-4 |
| | Lecture 2 | TLS RFC5246, SSH / HTTPS RFC2818 |

# Teaching Plan III

| | | |
|---|---|---|
| Week 10 | Seminar | ? |
| | Lecture 1 | Encryption |
| | Lecture 2 | PGP RFC4880 |
| Week 11 | Seminar | Assignment support |
| | Lecture 1 | Assignment support |
| | Lecture 2 | Assignment support |

# Learning Outcomes

**Knowledge & Understanding:**

- Demonstrate knowledge and critical understanding of the interaction between physical systems, computer hardware and software, including control theories, network protocols, and cybersecurity architecture and operations
- Apply principles of design and implementation of stack models, network protocols, control systems, and security

# Learning Outcomes

**Intellectual / Professional skills & abilities:**

- Design, implement, test, document and evaluate a networked embedded control system
- Apply software development tools and best practice to produce, test and debug software for small networked control systems using specifications for embedded devices and associated hardware

# Learning Outcomes

**Personal Values Attributes (Global / Cultural awareness, Ethics, Curiosity) (PVA):**

- Demonstrate independent, critical and reflective thinking and practice in the development of a networked control system, and engagement with appropriate professional and technical literature to support and communicate such development

# Software tools and resources

## Hardware

FRDM k64F an ARM Cortex M3 based Micro-controller in an Arduino like form factor

MBED Application Shield an Arduino compatible shield providing more IO Devices

### Loans

- that they can come and sign out an MBED board from the FEE equipment loans facility in PAN 113/114
- The loans room will be open from 10-1 every weekday
- for the remainder of the semester.

### You have been warned

- Overdue returns will be subject to a fine
- Any lost or damaged equipment must be paid for or replaced.

# Software tools and resources

## Software

| | |
|---|---|
| Git | Github is used for disseminating exercises and examples]] **And** for submission of the assignment |
| Atom | a good text-editor and development environment |
| PlatformIO | a framework for Atom that supports embedded systems and IoT development |
| LLVM- CLang | used by atom and platformio for managing source code, providing syntax highlighting, syntax checking, and auto-completion. |
| Mercurial | another version control system, many of the Mbed libraries are available via a Mercuial repository. With mercurial platformio can automatically download these. |

# Availability

- All the tools are Open-Source and available for free for Linux, Apple OSX, and Windows
- I have build code for the K64F on all three OSs using this framework
- I've build code for
  - This platform K64F
  - the previous platform (LPC4088)
  - Arduino Uno (ATmega328P) using
    - the Arduino language framework (C++)
    - bare metal C
    - assembly language