Engineering Tripos Part IIB, 4B25: Embedded Systems for the Internet of Things, 2018-19

Module Leader

Dr P Stanley-Marbell [1]

Lecturer

Dr P Stanley-Marbell [1]

Timing and Structure

Michaelmas term. 100% coursework

Prerequisites

3B2 useful

Aims

The aims of the course are to:

- Introduce students to the principles and practice of computation and sensing systems that interact with the physical world.

Objectives

As specific objectives, by the end of the course students should be able to:

- Define the role of uncertainty in measurements of physical signals and quantify measurement uncertainty for a given sensing system.
- Evaluate energy use in an embedded system using in-system current monitors.
- Define the role of noise in both measurements and displays and identify appropriate metrics to use in quantifying noise for a given design.
- Design communication subsystems and the required electrical circuit support between a collection of I2C- or SPI-interfaced sensor integrated circuits and an ARM Cortex-M0 microcontroller.
- Numerically quantify measurement uncertainty and noise in outputs given a system design.
- Recall and explain the interaction between displays and the human visual system.
- Design modifications to sensing, communication, and display systems to improve their energy efficiency.
- Design the logical organization and required firmware for new systems built around an ARM Cortex-M0 microcontroller, and sensors or displays connected via I2C and SPI communication interfaces.

Content

measurement uncertainty and noise, common sensor communication interfaces and how they interact with modern embedded microcontrollers such as the ARM Cortex-M family. The module will link these concepts in the signal acquisition and processing chain to a study of output interfaces in embedded systems. This exploration of output systems will be built on a study of the principles of operation of OLED displays and how the flexibility of the human visual system enables interesting circuit- and algorithm-level techniques to reduce display power dissipation.

Preliminary Syllabus

Lecture 1: System overview of sensing, computation, I/O, and displays in embedded systems; interpreting device and system datasheets. At the end of this lecture, students should be able to: enumerate the important components in an embedded system design; read and interpret the datasheet for a component in a system or for an entire system; propose and design changes to a system to extend its uses.

Lecture 2: Precision, accuracy, reliability, and measurement uncertainty. Noise sources in analog and digital systems; role of signal gain and restoring logic. At the end of this lecture, students should be able to: define precision, accuracy, reliability, and measurement uncertainty; analyze a system design and quantify these properties for a design's components; enumerate the sources of noise and measurement uncertainty in analog and digital systems; propose design changes to improve the robustness of systems to noise.

Lecture 3: C and assembly programming for embedded systems. At the end of this lecture, students should be able to: implement firmware that runs in the abscence of an operating system and which contains a mixture of C and ARM assembly code.

Lecture 4: Sensors, embedded I/O interfaces, and noise: Commercial sensor integrated circuits; I2C, SPI (and I2S, I3C, MIPI DSI, and MIPI CSI); noise in integrated circuits (Johnson-Nyquist noise, shot noise, 1/f noise, random telegraph noise). At the end of this lecture, students should be able to: enumerate the differences between the common embedded wired communication interfaces; select and substantiate a choice for an interface for a given design problem; enumerate the different potential sources of noise in integrated systems.

Lecture 5: Case study.

Lecture 6: Field-programmable gate arrays in low-power embedded systems; Verilog overview. At the end of this lecture, students should be able to: describe and explain the basic architecture of FPGAs; use their understanding of the Verilog hardware description language and FPGA synthesis tools to modify an existing Verilog design.

Lecture 7: Human color vision perception and its interaction withOLED displays: Their structure, interfaces, and techniques for energy-efficiency. At the end of this lecture, students should be able to: enumerate the properties of OLED displays; propose changes to existing system designs that use OLED displays in order to improve their energy efficiency; enumerate the basic properties of human color vision that have a bearing on the design of displays for embedded systems.

Lecture 8: Physical invariants in embedded systems. At the end of this lecture, students should be able to: define physical invariants in the context of a sensor-driven system; apply concepts from Lagrangians, Hamiltonians, the Euler-Lagrange Equations, Noether's theorem, and recent research on inferring Lagrangians and Hamiltonians from sensor data to embedded systems designs.

Lecture 9: Wireless communications using Bluetooth, 802.15.4/Zigbee, and LoRa; Bluetooth HCI interface. At the end of this lecture, students should be able to: enumerate the differences between the major low-power radio interfaces available for embedded or Internet-of-Things systems; propose energy-efficient choices for a wireless sensing system design given the application's design constraints.

Lecture 10: Schematic capture and basic printed circuit board layout using Eagle. At the end of this lecture, students should be able to: create a design ready to be submitted for manufacturing (Gerber files) using the Eagle schematic capture and printed-circuit-board layout tools.

Lecture 11: Designing new embedded systems to solve a specified application need. At the end of this lecture, students should be able to: propose an architectural design comprising sensing, computation, communication, and display to address a given application need, with the design implementable within the limitations of schematic capture and printed-circuit-board layout tools such as Eagle.

Further notes

Pictures of individual final project from previous years

The pictures below are for a sample of the individual student projects, for students who gave consent for pictures of their final demo system to be used in future instances of 4B25.

Pedometer (using 3-axis MEMS accelerometer, ARM Cortex-M0+, and 96x64 color OLED display from the course kit in custom laser-cut enclosure):

Bike theft alarm (using 3-axis MEMS accelerometer and ARM Cortex-M0+ from the course kit, in custom 3D-printed enclosure):

Pedometer (using 3-axis MEMS accelerometer, ARM Cortex-M0+, and color 96x64 OLED display from the course kit):

Plant health monitor using external conductivity, light, temperature, and humidity sensors together with the ARM Cortex-M0+ and color 96x64 OLED display from the course kit:

Sensor fusion (GPS + velocity) using FRDM KL25Z and GPS module together with 3-axis MEMS accelerometer and 96x64 OLED display from the course kit:

Razor cut prediction (HDC1000 humidity sensor and force-sensing resistor together with 3-axis MEMS accelerometer and ARM Cortex-M0+ from the course kit):

RFID-based class attendance monitor (using INA219 current measurement board, ARM Cortex-M0+, and color 96x64 OLED display from the course kit):

Custom multi-layer PCB for sports injury detection, using the same KL03 Cortex-M0+ as in the course kit:

Electric bike controller (using INA219 current monitor, ARM Cortex-M0+, and color 96x64 OLED display from the course kit):

Pedometer (using a counter module attached to the ARM Cortex-M0+ and 3-axis MEMS accelerometer from the course kit):

OpenGL subset (rotating pyramid demo) and OLED display driver implemented in only 2k RAM on the microcontroller, using the ARM Cortex-M0+ and color 96x64 OLED display from the course kit:

Coursework

Coursework	Format	Due date & marks
Coursework activity #1: Embedded processor emulator exercise Use the GCC and Binutils tools to compile, link, and disassemble binaries and use an open-source embedded system emulator to run a few different programs written in a combination of C and assembly language. <u>Learning objective:</u> After successfully completing this exercise, students should be able to: • Write simple programs using combination of C and assembler • Compile and run programs directly on an embedded processor with no OS • Create, use, and modify Makefiles and Linker Command Files • Use Linker Map Files and differentiate them from linker command files	Individual Source files, binaries, and text file with answers to questions. non-anonymously marked	Friday, 19th ([10%]
Coursework activity #1: OLED display control over SPI exercise Obtain hands-on experience writing a device driver in C for an SPI peripheral, using the FRDMKL03 ARM board and the OLED display from the course hardware kit. <u>Learning objective:</u> After successfully completing this exercise, students should be able to: • Read a datasheet for an unfamiliar embedded hardware component such as an SPI peripheral and write a device driver in C to interface with the peripheral.	Source files, binaries, picture of working system, wiring diagram, and text file with answers to questions. non-anonymously marked	Friday, 26th 4) [15%]
Coursework activity #3: Project proposal one-page report Identify an interesting engineering problem that can be addressed using an embedded system developed using the concepts, theory, techniques, and tools covered in this course. <u>Learning objectives:</u> After successfully completing this exercise, students should be able to:	Individual Report non-anonymously marked	Friday, 2nd 5) [5%]

Coursework	Format	Due date & marks
- Identify an interesting and important engineering challenge that can be solved using a combination of embedded sensing, embedded computation, and possibly displays and communication. - Present a clear list of design objectives for solving the engineering challenge using an embedded system.		
Coursework activity #4: Power measurement using TI INA219 I2C device exercise Obtain hands-on experience writing a device driver in C for an I2C peripheral, using the FRDMKL05 ARM board and the TI INA219 daughterboard from the course hardware kit. <u>Learning objective:</u> After successfully completing this exercise, students should be able to: Read a datasheet for an unfamiliar embedded hardware component such as an I2C peripheral and write a device driver in C to interface with the peripheral.	Source files, binaries, picture of working system, wiring diagram, and text file with answers to questions. non-anonymously marked	Friday, 9th N 6) [15%]
Coursework activity #5: Project interim report Present progress made towards final project goals, evaluate lessons learned so far, and obtain feedback and guidance on necessary plan adaptation. <u>Learning objectives:</u> After successfully completing the interim project report, students should be able to: - Identify and present progress made towards final project. - Identify and present potential challenges and propose necessary changes to project plan.	Individual Report non-anonymously marked	Friday, 23rd (week 8) [10%]
Coursework activity #6: Project concept, design, implementation, and final report Present the problem addressed, approach employed, system implemented, and system evaluation. <u>Learning objectives:</u> After successfully completing the final project, students should be able to: - Identify an interesting and important engineering challenge that can be solved using a combination of embedded sensing, embedded computation, and possibly displays and communication. - Design an embedded computing system that address the engineering challenge.	Individual report, source files, binaries, in-person demonstration at final feedback session. non-anonymously marked	Friday, 18th Term) [45%]

Coursework	Format	Due date & marks
• Prototype an embedded system design using a combination of sensors, microcontrollers, communication, displays, or FPGAs using the tools provided in the course kit, and potentially design a custom PCB implementing the design. • Quantitatively evaluate an embedded sensing and computation system in terms of its time efficiency (performance), energy efficiency (battery life), and measurement and data processing accuracy.		

Booklists

The following books are relevant to the material in the course and will all be available from the Engineering Library.

1. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach,* ISBN: 978-0262533812

2. *An Introduction to Uncertainty in Measurement*, ISBN: 978-0521605793
3. *Linkers and Loaders*, ISBN: 978-1558604964
4. *The Circuit Designer's Companion*, 3rd Edition, ISBN: 978-0080971384
5. *The Practice of Programming*, ISBN: 978-0201615869
6. *Expert C Programming*, ISBN: 978-0131774292
7. *C: A Reference Manual* (5th Edition), ISBN: 978-0130895929
8. *Bluetooth Low Energy: The Developer's Handboo*k, ISBN: 978-0132888363
9. *Programming Embedded Systems: With C and GNU Development Tools*, 2nd Edition, ISBN: 978-0596009830
10. *Embedded Systems Dictionary*, ISBN: 978-1578201204
11. *The Art of Designing Embedded Systems*, Second Edition, ISBN: 978-0750686440
12. *The Art of Electronics*, ISBN: 978-0521809269
13. *Color Science: Concepts and Methods, Quantitative Data and Formulae*, ISBN: 978-0471399186

Examination Guidelines

Please refer to Form & conduct of the examinations [2].

Last modified: 07/10/2018 18:56

Source URL (modified on 07-10-18): https://teaching21-22.eng.cam.ac.uk/content/engineering-tripos-part-iib-4b25-embedded-systems-internet-things-2018-19

Links
[1] mailto:ps751@cam.ac.uk
[2] https://teaching21-22.eng.cam.ac.uk/content/form-conduct-examinations