

MPhil in Advanced Computer Science

Low Power Embedded Systems Programming

Leader:	Ian Wassell
Timing:	Michaelmas Term
Prerequisites:	Familiarity with C and Linux
Structure:	8 × 2 hour classes

AIMS

This module teaches students how to write programs for low power embedded devices that communicate with other computers or devices to solve practical problems. The course will cover the relevant theoretical knowledge required for writing such applications, and provide practical experience writing C programs for Atmel microprocessors. The module will also give students an appreciation for some of the research issues in low power embedded systems.

SYLLABUS

Each of the first five classes will follow a two-part format where a practical session is used to reinforce understanding of the material introduced in the first part of each class. It is expected that the short lecture session will involve progressively greater student involvement as the course progresses. The intention is that each student will formulate a project proposal by the end of the 5th class that they will execute during the final 3 classes. During the practical sessions embedded devices will be programmed, guided by a module handbook in the early classes. Feedback from the other students as well as from the module demonstrators will be offered during these sessions. Demonstrators will be available throughout each practical session to assist students with their work and to lead discussion on general problems as they arise. Students will be expected to review a set of reading material before each class.

In the final 3 sessions, the students will concentrate on implementing their chosen project. In addition, further interaction and feedback among the students will be encouraged by the inclusion of Q and A sessions. In the final session, each student will give a brief summary of their project to the rest of the class. The coursework will be completed over the Christmas vacation.

The topics of each practical session are:

1. Setting-up the software – tool chain. Setting-up the hardware – microcontroller master clock, basic Input/Output (I/O) techniques.
2. Basic serial communication, serial communications using interrupts, analogue to digital conversion (ADC).
3. Creating precise timing using Timers. Interfacing to a Liquid Crystal Display (LCD).

4. Accurately measuring time interval between input events – Interrupts generated from input transitions and use of timers. Processor low power and sleep modes.
5. Interfacing to an external SPI memory device – use of libraries, use of SPI interface, use of flash memory device.

6-8 Project work.

OBJECTIVES

On completion of this module students should:

- be able to read and write C programs for Atmel microprocessors;
- understand the event-based programming paradigm presented by interrupts in C;
- know how to minimise power usage in embedded and mobile platforms;
- have an appreciation for the research issues presented by programming embedded and mobile devices.

COURSEWORK

Write a 4000 word report on an embedded C program application that the student has designed, written and tested. The report must be submitted by the first day of lectures in the Lent Term.

PRACTICAL WORK

The first five practical sessions will each involve an assessed exercise that the students will complete in the practical session. Prior to session 6 a project proposal will have been formulated and agreed with the Module Leader. In the final 3 sessions the students will work on their projects.

ASSESSMENT

For each component of the assessment please describe:

- Ticks at the end of sessions 1 to 5.
- Assessment of the written report describing the conduct and outcome of the project. The report will follow a format similar to that of the Part II undergraduate projects, i.e., the 'pink book'. The report will contribute 75% of the final module mark.
- The ticks will contribute 25%.
- The final module mark will be expressed as a percentage.

RECOMMENDED READING

- Kernighan, B.W. & Ritchie, D.M. (1988). *The C programming language*. Prentice Hall (2nd ed.).

- Useful websites:

Atmel 8 bit avr micro

controllers: <http://www.atmel.com/products/avr/default.asp>

Atmel application notes for 8 bit avr:

http://www.atmel.com/dyn/products/app_notes.asp?family_id=607

avr-libc library: <http://www.nongnu.org/avr-libc/>

Last updated: Aug, 2009