Instructor: Professor T.W. Pearce, P.Eng., room 3215VS, e-mail pearce@sce.carleton.ca (Please send from your cmail account and include “SYSC 4906” in the subject line to ensure a response.)

Office hours: By appointment. See course web site for additional hours.

Instructional Hours per Week:
- 3 Lecture hours: Monday and Wednesday: 14:35 – 15:55
- 2 Lab hours: Thursday 14:35 – 16:25

Prerequisites: ELEC 2501 AND ELEC 2607 AND (SYSC 2003 OR SYSC 3006) AND SYSC 2006 (or equivalent) AND department approval.

Department Approval: The course enrolment will be limited to 16 students.

If you would like questions about the course, please send email from your cmail account to Prof. Pearce (pearce@sce.carleton.ca) and include “SYSC 4906” in the subject line. In the body of the email please ask your question.

If you wish to take the course, please fill in the “contact form” (the link is at the bottom of this page: http://sce.carleton.ca/ughelp) with subject “About the course SYSC 4906”. In the body of the email indicate that you would like permission to take the course. You may also include any extra information such as extra-curricular experience with embedded systems that should be considered during evaluation (e.g. projects and/or work experience). Permission to take the course will be based primarily on GPA (i.e. students with higher GPAs will be given preference over students with lower GPAs) and possibly extra-curricular experience, and then on a first-come-first-served basis. Permission will be issued by Friday June 20 (at the latest) for any requests received before Thursday June 19.

Calendar Description
Embedded systems: applications, requirements, software development, firmware, design and implementation using microcontrollers. Microcontroller systems concepts: system on chip, configuration, initialization, clock selection, internal bus architecture, memory architecture, peripherals, interrupts, exceptions, power saving modes, external interfacing buses and protocols, debug support. Lab intensive: programming in C, hardware interfacing, students work in teams of 2.
After completing the course, students will have experienced and demonstrated:

- The conversion of application requirements into working prototypes of embedded systems. Involves engineering methods applied to ARM-based microcontrollers, including: developing designs and test plans from requirements, using an appropriate toolchain to implement designs and verify requirements, and debugging as needed.
- Interfacing to ARM-based microcontrollers using standards as well as ad hoc techniques. Some lab exercises will involve building ad hoc interface circuits (such as interfacing external signals), while others will involve the use of standards (such as serial protocols).
- Working in small groups to complete engineering tasks. This includes planning and dividing work assignments among team members, completing work assignments individually, and then integrating assignments into a working system.
- The development of firmware for embedded systems.

**Textbook:**

“Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers, Volume 2”, 2nd Edition, Jonathan W. Valvano, 2012 (Note: The 3rd Edition may be available by September, in which case the 3rd edition will be used)

**Supporting Reference:**


**Web Page:** The course will use cuLearn (details to follow). Students are required to check this page often for course updates. Course materials will be posted there for student use.

**Grading Scheme:** (for students who pass the Final Exam and Lab Component)

- Lab Component: .............................................................. = 40 %
- In-Class Test: tba (in class time) ........................................ = 10 %
- Final Exam (scheduled, 3 hours, closed-book) ............ = 50 %

**Important Notes:**

1) **Students must pass both the Final Exam and the Lab component** in order to pass the course. (i.e. Failing to pass the Final Examination or the Lab Component will result in an F grade for the course).
2) **Students are expected to attend all lectures and lab sessions.** If a student is absent, it is up to the student to obtain missed material and information from colleagues in the course.

3) **Students who miss the midterm test** due to illness must provide a valid medical certificate to the instructor within 48 hours of returning to campus. A medical certificate must adhere to the format required by the Registrar. The format is available as a PDF form through the Registrar’s website [http://www.carleton.ca/registrar/forms](http://www.carleton.ca/registrar/forms). Once the certificate has been verified, the test weight will be added to the final examination weight.

4) **Plagiarism** (e.g. copying and/or handing in for credit someone else's work) is a serious instructional offence that will not be tolerated. Please refer to the section on instructional offences in the Undergraduate Calendar for additional information.

5) **Deferred Exams.** Students who miss the final exam may be granted permission to write a deferred examination (see the Undergraduate Calendar for regulations on deferred exams).

6) Accreditation of Engineering programs requires that classes and laboratories, tutorials, or problem analysis sessions continue to run through any review period of the fall term.

7) **Accommodation**: You may need special arrangements to meet your academic obligations during the term. For an accommodation request the processes are as follows:

   - **Pregnancy obligation**: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website [http://www.carleton.ca/equity/accommodation/](http://www.carleton.ca/equity/accommodation/)

   - **Religious obligation**: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website [http://www.carleton.ca/equity/accommodation/](http://www.carleton.ca/equity/accommodation/)

**Academic Accommodations for Students with Disabilities**: The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your Letter of Accommodation at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website for the deadline to request accommodations for the formally-scheduled exam (*if applicable*).
Health and Safety:

Every student should have a copy of our Health and Safety Manual. An electronic version of the manual can be found at:


**Additional Information:** The following lecture content will be spread throughout the course in (roughly) the indicated amount:

**Application Firmware Development (6 hrs):** The engineering methods used in developing embedded application software typically involve: cross-compilation (often including early simulation on the host), downloading into the microcontroller, testing and debugging. The application software is often referred to as *firmware* because it is integrated tightly with the target hardware. Downloading the firmware is mandatory during prototyping and development, but is not necessarily a functional requirement of the application being developed. Testing requires demonstrating that the system meets its requirements. Firmware design must include support for the specific tests that will be used in system verification; i.e. the software must be designed to facilitate testing. Modern debugging relies on the microcontroller’s JTAG interface which provides direct access to dedicated debug blocks in the microcontroller’s integrated circuits. The lab exercises will entail applying methodical engineering approaches when developing embedded systems.

**Device Configuration and System Initialization (3 hrs):** The configuration and initialization of embedded systems is a key aspect in creating application-specific systems. Microcontrollers are designed to provide flexibility in how the integrated resources are configured when the device is powered on. This flexibility allows the device to be customized for specific applications, as well as to awake quickly from power-saving modes without requiring re-initialization by software. The lab exercises will include configuring and initializing microcontroller-based embedded systems.

**Operating Modes (3 hrs):** A microcontroller offers a set of processor-specific operating modes that can be entered by various hardware and/or software events. Entering a mode causes the processor’s state to change automatically to accommodate particular conditions typical to applications. Power management is an important requirement that is usually implemented using operating modes. The lab exercises will involve designing applications around operating modes and leveraging the modes to manage power consumption.

**Memory (2 hrs):** Microcontrollers support a variety of memory types, such as ROM, RAM, flash and EEPROM, and memory protection units are becoming increasingly common. The lab exercises will involve using the memories in building embedded systems and providing memory protection.

**I/O Peripherals (9 hrs):** Embedded systems typically require interactions with external signals, and microcontrollers include a wide range of I/O peripherals to accomplish this. A sampling of peripherals includes general purpose digital I/O, ADC, DAC, analog comparators, PWM, and various serial devices.
(often including hardware support for common standard serial protocols such as \( \text{I}^2\text{C}, \text{SPI}, \text{USART}, \text{JTAG}, \text{CAN}, \text{USB}, \text{and} \text{Ethernet} \)). Peripherals are typically accessed as memory-mapped port-based devices. The lab exercises will give practical experience with common peripherals, interfacing external circuits with microcontrollers, and understand how to leverage internal components like DMA and the interrupt controller to improve the performance of embedded systems.

**Operating Systems and Middleware Support (4 hrs):** The application supported by an embedded system is determined by its firmware. The need for application-specific code is unavoidable; however, vendors (and open source communities) often make microcontroller-specific operating systems and libraries available to deal with lower-level hardware details. The middleware is sometimes called a board support package (BSP) and BSPs are available for virtually all microcontroller-based boards. Some lab exercises will include the use of appropriate 3rd-party software to construct embedded systems.

**Applications (7 hrs):** An embedded system implements a specific application. Some lecture time must be spent introducing the applications specific to the lab exercises. The exact content depends on the final exercises selected, which is still under development.