

## **Embedded Controller Programming II**

**INSTRUCTOR:** Ken Arnold send class related e-mail to: [ecp@hte.com](mailto:ecp@hte.com)

**CLASS WEB PAGE:** <http://www.hte.com/uonline/ecp>

**CREDIT:** 2 Units / 6 class meetings

**TEXTBOOK:** Schultz, C and the 8051 Vol. 1, and supplemental material from Ayala, The 8051 Microcontroller (text from ECP-I)

**PREREQUISITE:** ECP I or permission of instructor and some C

**OTHER STUFF:** email and web access, an IBM pc for programming

### **Course Description :**

This course continues the material presented in the ECP I class, which provides a hands on learning experience programming the most popular 8 bit microcontroller family, the 8051. ECP II will continue to explore the 8051 family's programming model as well as covering extended 8051 features and introducing the C programming language for programming the 8051. This class will also look at some basic interfacing in order to support extending programming outside of the basic 8051 architecture to standard external devices. Course topics will include the following; coverage of the popular Keil development environment, C programming for the 8051, the 8051's interrupt system and basic interrupt driven programming, additional/extended 8051 features, applications including displays and other output devices, input devices, programming and controlling external components such as temperature sensors and motors, as well as an introduction to an 8051 RTOS. In addition, some general topics relating to C programming for embedded applications will be discussed.

### **Methods of Instruction:**

The classes will consist of lecture and examples that students are likely to encounter in implementing basic embedded controller applications. Supplemental materials will be provided and students will work with an 8051 development kit (SDK) for developing and testing programs and for the course project.

### **Student Evaluation:**

Course grading for ECP II will be based on two components;

- 1) Each student will do a project using the 8051
- 2) A final exam will be given at the end of class

### **Objectives:**

The objective of ECP II is to continue learning the 8051 and embedded controller programming by introducing C programming as well as additional 8051 features and application examples. By the end of this course the student should be familiar with C language programming for embedded applications as well as modular software development and basic I/O interfacing.

## **ECP II – Course Outline**

*Please feel free to suggest other topics of interest that you would like to cover and we can try to include them as time permits;*

**SECTION 1** - Review of C language

**SECTION 2** – Introduction to the Keil Development Environment and the basic C language development and debugging tools. Basic C language program development and testing.

**SECTION 3** – Basic displays and output devices. Examples including simple LEDs, buzzers, 7 segment displays, LCDs, and the serial ports. Software time delays. Intro to modular code development.

**SECTION 4** – Basic input devices, including serial port, switches and sensors, and keypad. Additional 8051 features including timer the 8052/32's timer 2 and Dallas semiconductor's enhanced features.

**SECTION 5** – Control applications including motor control, driving stepper motors, A/D and D/A converters.

**SECTION 6** – Basic RTOS applications  
Embedded Programming Proverbs

In addition, we will dedicate some time during the last class session to present class projects. This will give the students an opportunity to see the designs/ideas of others and discuss project development. The sections above are broken down by topic rather than by class week. This will allow us to cover one or more sections per class session if necessary. This class will not have weekly homework assignments however, I will suggest outside reading material relevant to class discussions. You should be starting on your projects as soon as possible as we only have 5 weeks in which to complete them. All projects will be due at the last class session.