E3252 Embedded Systems Design for Power Applications (3:1)
Kaushik Basu, Department of EE, IISc.

Though the course emphasizes platform-independent learning, it is impossible to learn without working with an actual MCU, so we used TMS320F28379D. The first part of the course is about learning the MCU. The second part is to apply all this learning for a successful embedded system design of a simple power converter that captures most of the essential aspects of the subject.

Part A

Introduction to embedded systems design of a power electronic converter. Introduction to the real-time processors-microcontrollers with DSP engine. Reading the datasheet of an MCU. Minimum hardware design to make an MCU operational: Introduction to the concept of a launchpad. Software initialization: clock, watchdog timer, Flash memory etc. Introduction to embedded C and modular programming (LAB1). Introduction to interrupts and peripherals-Timer and GPIO (LAB2). Memory organization-ROM, RAM (local, global, message etc.), linker files, structure and location of special function registers. Placing a table in ROM (LAB3). Inter-processor communication (IPC-LAB3), control law accelerator (CLA-LAB3). PWM peripheral (complimentary with deadtime, HRPWM, Trip zone) (LAB4), quadrature encoder peripheral (QEP-LAB4). UART communication with a GUI developed on PC over MODBUS protocol (SCI-LAB5). SPI communication with an external DAC IC (LAB6). I2C communication with an external EEPROM IC (LAB6). Analog signal conditioning, biasing and calibration, ADC and DAC peripheral (LAB7). Fixed point arithmetic with C, intrinsic and mixed assembly coding (LAB8).

Part B

Embedded systems design of a simple power converter-multiloop programming-protection, supervisory control, communication monitoring, debugging etc. Codesign with CPU1, CPU2 and CLA (LAB9). Dynamic modelling of the converter, nested loop feedback controller design, modelling the ADC and PWM modulator, digital implementation in fractional arithmetic, limit cycle oscillation (LAB10). Controller implementation using FPU and CLA (LAB10).

I want to add a few items:
1) DMA to manage bulk data transfer between CPU2 and GUI, 2) VCU for CRC computation and enhancing the speed of controller computation, 3) USB 4) CAN.