UF ECE Design 2 Guidelines - January 2017

Pair up with a partner. 1 CE + 1 EE often makes for a good project. No 3 person projects.

Basic design requirements:

- Adequate analog design. TBD per design. (Not required for CE's)
- Microprocessor(s) with software intelligence.
- You will design and produce your own working PCB(s).
- No development boards will be allowed in the final project unless your project has adequate complexity. Examples of development boards are Arduino, Raspberry Pi, Beagle Bone, etc.
- You are not required to design RF circuitry for wireless functionality instead use "off the shelf" RF modules.

Ideas considered too basic or simple for UF ECE Design 2:

- parking garage counters or equivalent.
- home security or automation system... Unless you have some serious intelligence or complexity involved.

Formulating Your Project/Outlining your project idea:

- What is the purpose?
- How does your design accomplish its purpose?
- Describe the system in terms of:
  - Inputs: sensors (Analog or Digital?), or controls
  - Outputs: various
  - Flow charts (Please use appropriate flow chart design elements in reports)
  - User interface (design it for non-engineers)
  - Platform or physical system

Resources:

- INTERNET !!!!
  - How to solve specific problems and what kinds of sensors are used.
  - Check on the various types of hardware already available.
  - Circuit examples. Don't just copy and paste, you will need to understand the circuitry.
  - Data sheets. <== Many of your answers are found here!
- Senior Design Personnel such as the TA in lab, Dr. Cheney NEB265/NPB2232, Mike NEB239, Eric NEB236.
- ECE professors per field of study/research related to your project.

Time Management/One Semester (4 month) Hardware/Software Design Strategy:

1. Research
   - Methods of accomplishing design challenges. What is the state of the art?
   - Inputs: sensors, controls
   - Microprocessors

   Using the exactly same Microprocessor/Micro-Controller from Design 1 is not allowed.
   Scale your micro to your specific application. Choose only what your design calls for with some extra I/O.
   I/O - determined by platform, physical system and user interface.

2. Begin Testing with Proto-boards (i.e. 4744 uP board, DSP class board, commercial proto-boards, etc.)
   Interface sensors, microcontrollers, LCD screens, servos, any and all components of your system.
   Initial modeling is fine for determining scale and range of sensor families, but there is no better 'simulation' than plugging things in and learning how they operate.
   Make breakout boards for surface mount components in the first few weeks of the semester.
   This means "Start your Altium schematic immediately!"
   Multiple schematic pages for different sections of your design, avoid clutter, stay organized at the start.
   Always use lab power supplies with a current limiting function until the board is verified as "safe".

3. Initial Main PCB design - First draft
   Don't make it too small, leave yourself room to solder and test points on key nodes.
   Examples of test points are: Communication lines such as USART, SPI, I2C and I2S. I/O port breakouts.
   More test point examples: Input + Output stages of filters, op amps, etc. Jumpers in between circuits?
   Think ahead about the physical placement of your components and run traces on the bottom layer.
   Add pads as needed for extra test components.
   Use wide 30 mil ground node traces. Star ground, avoid ground loops!
   Most other components will have 20 mil traces with 50 mil clearance.

4. Final PCB(s)
   Can be smaller with surface mount parts, but don't go below 0805 SMD res + caps.
   Try to eliminate breakout boards and include all circuitry on a single board if possible.
   Consider having made a professional PCB with through-hole plating (usually at least 2 layers) and a solder mask.
   MOUNTING HOLES TO PUT THE DESIGN IN A BOX! Design the PCB to be mounted into a case of some sort.
   Project box "off the shelf" or a 3D modeled case. Laser cutter/engraving is also available.
   - ***Check with Eric about 3D printing and/or laser cutting BEFORE you start your 3D design.***
   - Design with your case with the following in mind: panel mount controls, LED indicators, LCD mounting, knobs, etc.

Electrical & Computer Engineering - Capstone Design Requirements - January 2017

1. Design, built and test your own PCB. If the PCB is substantially complex enough, then two people may participate on the design and testing as a team effort. In this event, each student should be proficient in using the Altium Designer PCB creation software.

2. The design should comprise a microprocessor, microcontroller and/or digital signal processor (DSP). FPGAs are also strongly recommended. The selected components should be scaled to the application, i.e., you don't need a super computer for a wrist watch. Use the size/cost processor that makes sense for the application/product.

3. The design should also contain an analog and digital portion. The digital portion should be comprised of both hardware and software. Below are examples of common analog and digital designs.

Analog Design

sensor signal conditioning, signal synthesis of analog signals, class AB Amps, power supply design (linear and switching), high speed (clock rates greater than 50 MHz) PCB design, multi-phase motor control, low noise preamplifiers, RF circuitry, battery charging circuits (Li, Li-Poly), ultra low power electronic design (choosing & implementing the proper components), energy harvesting (vibration, solar, RF), electromagnetic applications (antennas, toroid design, inductive charging, boost converters)

Digital Design

Implementing microcontrollers, microprocessors and Digital Signal Processors. Writing efficient assembly and/or "C" code for the previously mentioned devices. Interfacing Analog to Digital (A/D) and Digital to Analog (D/A) converters, SRAM, DRAM, Flash, Codecs, LCDs, Graphic Displays (parallel and serial) and other digital peripherals. Sampling digital signals and outputting digital pulse trains. i.e., serial communication (I2C, SPI, RS232, RS485, I2S, USB, the more IEEE standards the better). Implementing complex algorithms in your digital processor. i.e., filtering, FFTs, image recognition, speech recognition, proximity/range Detection, artificial intelligence, control related (PID Loop, Fuzzy, Neural Nets).

Special Note: Computer Engineers (CEs) are not required to create an analog design. However the software portion of their digital design should be much more extensive than a typical embedded program created by an electrical engineer (EE), i.e., the CE should design a user interface that is targeted for non-engineering "lay person" -level usage. EEs, on the other hand, should have more extensive hardware designs and software user interfaces that at an engineering usage level. CEs, therefore, should be designing very "user friendly" intuitive interfaces.

If you are not sure that your design meets the EE Capstone Design Criteria listed above, please contact ECE Design 2 personnel (Dr. David Cheney, Mike Stapleton, Eric Liebner) directly for additional information.