Capstone Projects

A requirement for CS 437 IoT

Overview:

With any subject, it is important to not just sit and watch. It is important to do things for real. Doing things for real helps you learn things deeper and gives you confidence in the material. With this class, we have such a cool opportunity here. I mean, think of all the amazing IoT devices you could build. A smart door to your apartment that plays music when you walk in? A face monitor to alert you if your spouse is upset? A drone that waters your garden?

So as part of our class, you will be doing an independent project, where you build a real IoT device. Choose something you are interested in and would be excited about. Your topic should be IoT related and should build your skill sets in some way (e.g., if you do microcontroller development for a living, get out of your comfort zone and work with IoT clouds). Don't be afraid to dream big - open-source platforms and libraries such as OpenCV, Pixhawk, make many problems that were once extremely hard totally tractable to do. You will earn a higher grade for proposing and executing projects that are novel or demonstrate creativity (try to choose projects that could become a startup, or solve a new problem for people where you don't see existing products, etc). Make sure you choose a project that has clear milestones you can realistically achieve within the semester timeline, and do have some concrete outcomes at the end, but please note grading is more based on what you learned along the way, so it is better to take risks and shoot big.

Project Topics:

The topic should be something you are interested in. If you're interested in it, you'll have more fun building it, and you'll learn more. Some ideas are provided below. I am providing these as examples to get you thinking, try not to just choose something on this list. We want you to be creative and think of your own novel ideas so you can earn the highest grade. .

1. A pet monitor system. What is your cat doing during the day? Use OpenCV or some equivalent platform to perform object and action recognition using computer vision. Collect results from some Raspberry Pis with cameras distributed around your apartment.
2. Github heads up display. Use Github APIs to monitor the status of your builds, how many open bugs you have, blockers, etc. Mount a cheap Raspberry Pi screen on top of your PC and display statistics related to your job.
3. Thing tracker. Never lose your keys again. Create miniature bluetooth beacons, attach to clips, clip them to various important items in your house. Deploy a few access points that maintain and report a list of item locations.
4. Smart barbell. Add gyroscope/accelerometers/etc to barbell. Design system to monitor lift speed/balance/etc. Give feedback to user on their technique. Can you detect injuries

before they happen?

To further inspire you, you might also look on youtube. For example, Michael Reeves, Simione Giertz, and Devon Crawford produce videos of projects along with specifications on how to build them. The instructor also maintains a (partial) list of IoT application areas which might be interesting to think about. Finally, the instructor (Matthew Caesar) also performs research on Internet of Things. You are welcome to reach out to him (caesar@illinois.edu) and he can mention some research projects he is doing that you could get involved in.

Project Scope:

One question is how "big" the projects should be to meet the requirements of the class. Try to choose a project that you can implement in approximately 50-100 hours of time, multiplied by the number of people in the group. It is ok, in fact encouraged, to use open-source code, but please be clear in your report about which parts of the project are your original contribution.

Working in Groups:

You may work by yourself or in groups of 2-4 people. If you work in a group, please keep in mind that each person needs to do a fair share of the work, and that having more people doesn't reduce the requirements per person. So if you have two people in your group, you should expect to spend 100-200 person hours (across the whole group) on your project. So larger groups should work on larger projects. Each person must contribute to the project implementation itself, not just planning or reporting. That is, each person must contribute substantially to the proposal, report, video, and code; each person must appear or speak for some non-trivial amount in the video and write some non-trivial amount of the paper, but everyone in the group can (and should) submit identical materials for recordkeeping purposes, along with a list of teammates.[1]

Project Proposal:

Your proposal should be approximately 500-750 words and must include the following information:

- Motivation: explain the purpose of your project, the problem it is trying to solve, and why someone should care about that problem..
- Timeline: list the specific deliverables you plan to produce in the project. Don't just say the end result, break the result into pieces - it is good to have intermediate checkpoints when you develop something. Specify when you plan to complete each item.
- Group: if you're working in a group, list the other people in their group (name and also netid).
- Contribution: briefly describe the contribution that each group member will make to the project.

[1] Points may be deducted if teammates do not turn in identical submissions, since the group should agree on the contents of the project before submitting it. (We won't try to guess which teammate's submission is the "real" one.)

Rubric (100 points total):

- Motivation is clear and convincing (20 points)
- Project topic is important and interesting (20 points)
- Proposed project is technically sound and will extend skill sets of participants (20 points)
- Milestones are clearly described and realistic (40 points)
- Topic/idea is novel/creative -- problem not solved by existing products, hasn't been done in prior instances of class (up to 30 extra credit points)

Project Submission:

After completing the project, at the end of the term, you will submit a final report on your project. Your submission will consist of two parts: a paper, and a video. To send us your video, please upload it on some site (e.g., Illinois Media Space, Illinois Box, Google Drive) and send us a link. Your link doesn't have to be public, but please make sure your link has sufficient rights to be viewed by course staff.

The submitted paper must include:

- Motivation: this should be more detailed than your initial writeup. Talk about the importance of this problem. Pretend you are trying to convince someone to give you funding, or purchase what you developed. You may also want to give references/citations here.
- Technical approach: describe the technical approach you used to complete your work. You can give your overall architecture, describe how data flows through your system, describe algorithms you developed, provide circuit diagrams, etc. Provide anything important in understanding *how* you did what you did.
- Implementation details: this is where you give the details in your implementation. Talk about specific software packages you used, hardware modules, any algorithms or research papers you referred to, data structure and protocol choices, etc. You should provide at least an informal list of citations of all these external materials that went into your project.
- Results: so, how did things turn out? You can provide performance results, experiences you had interacting with it, etc. Also talk about what the takeaway is - why should we care about your results? And, it is ok for things to go wrong - what did not go right in your project, what was hard and what lessons did you learn?
- Contribution: If you worked in a group, each member's contribution should be clearly stated at the end of the paper. The report will not be graded if this section is missing.

The video should not go over 10 minutes and must include:

- A demo of your project. Walk the viewer through how to use it. Please speak while you are doing this, explain what is going on on the screen.

Rubric:

- Paper (100 points)
    - Motivation (20 points)
    - Technical approach (25 points)
    - Implementation details (25 points)
    - Results (20 points)
- Video (60 points)
- Overall project (40 points)
    - Skill-building: did the project cause the group members to extend their skill sets? Did you learn the specific new things you had set out to learn, or were you able to do everything using only your prior knowledge (not as good)?

- Innovation: was the project “new” in some way? Was the idea itself new, or was some new technique used to approach an existing idea, or did the students substantially build upon existing research? Or did it bring something new into the student’s life?

Note: You are encouraged to not use AI generated text to write your project proposal and final paper (in our experience it makes your writing worse as these generators seem to create a lot of fluff/useless text, which will lower your grade). However, if you do end up using AI to generate some text, you must mark it in red.

Good Luck!