

# Advise for the Class

## RC car parts

- Read [this article](#)
- Begin to acquire parts for the RC car (including sensors etc.)
- Reminder:
  - Your worth as new graduate should be about \$50/hour (as an engineer as of 2022)
  - Therefore, buy \$30 sensor rather than \$2 ebay sensor to avoid writing software to deal with unreliable sensors

## General Advice

Here is some raw feedback from former students:

- I would Highly recommend that future peers take this class.
- Reduce amount of concurrent classes / work hours while taking this class.
- Take this course if you really want to learn, not just to complete credits. If you are not contributing in group project/ group lab assignments, you might get the grades but not what this class is for.
- Focus on the class heavily during the first month especially given the tedious first couple of labs and the CAN bus lectures.
- Take CMPE244 first in order to fast pace your learning for this class.
- Start right from the moment you register your course. knowing required tools, skills and software would help a lot
- For your personal development: Take the lab assignments and unit tests seriously. This is not for the grade, but for personal learning, because this material is directly applicable to interview questions for Embedded/Firmware engineering positions. I had multiple interviews where I directly used concepts/problem solving skills I developed while completing the lab / project assignments. For the team project: Take the time to discuss among the team members and vote/decide on a weekly "team meeting" time-slot. This will serve as your weekly "planning meeting" where everyone comes together quickly to discuss past progress, plan the next individual assignments, and coordinate the next in-person integration/testing session. Take the time to coordinate weekly in-person integration/testing sessions. This turned out to be most convenient to do on-campus, on weekends. The student union is open on the weekends, and has working space/power outlets. This is a great time to take everyone's individual implementations, and integrate them with the physical hardware, as well as debug all the problems that will inevitably come up.
- Spend a good amount of time from day one of class in knowing every aspect of assignment.
- Earnestly engage with the material. Don't take shortcuts or do the bare minimum for labs and project. Get comfortable with the sjtwo workspace and build tools and use them frequently (several times a week). Be honest about your shortcoming and what you don't understand so you can do something about it, there's an

economy to honesty. You're here to learn and grow as an engineer.

- Don't be afraid to ask questions or for more examples.
- Don't leave the labs until the last minute. It's better to make small meaningful progress as you are technically doing your project while doing the labs. Rushing the labs will lead to a lesser project down the road.
- This is a 3 man project done by 5-6 people. It will seem as if there is not enough stuff for people to do, but there is always something to find to do.
- Understand CAN bus and how to read sensor datasheets
- Make teams early and get the hardware part- RC car (it takes time in assembly), sensors and other stuff. Focus on the lectures and labs they will be used in the RC Car software development. And be excited about learning new everyday.
- Should have good experience in coding and also should have worked on atleast one development board to get the maximum benefit of the class.
- A strong suggestion to take this course with special interest and prepare in advance a little to get the most out of it

## Advise from 2024 class:

- Don't miss class. Not able to catch up
- Be consistent, hardwork and you will see the results at any point of the class.
- Read the lab assignments and requirements multiple times to get the better understanding of the lab. This class is rigorous; planning and starting early on labs and projects will be the best help.
- Definitely take the class to learn about CAN and embedded
- This is definitely a fast paced course, it is recommended not to take this class when you plan to take 4 subjects.
- Plan ahead. This is possibly the best advice. Analyse what others have done in the past and make actionable plans well in advance. Go through all the team reports from past 2 semesters at least.
- Start your homework early.
- 1. A piece of advice I received from former students is to wholeheartedly enroll in this class if you aim to pursue a career in the embedded industry, and I support that. 2. I personally know individuals who transitioned from non-embedded backgrounds to thriving in reputable companies after taking Preet's class. 3. Encountering such committed professors, relevant courses, detailed projects, and thorough documentation is rare. Therefore, I urge you to seize this opportunity. The knowledge gained here is practical and invaluable, so push yourself to explore additional credit sections in assignments, experiment with code, and go the extra mile—you won't regret it! As Preet often emphasizes, there is no magic.. Hard Work will be paid off well.
- Invest enough time initially to get to know the development environment and sjtwo board. Dont rush unit tests, take time to understand.
- 1. Start looking for parts and begin ordering them as soon as you establish your team. 2. Don't be afraid and wait for the last minute to ask questions in Slack. People need time to view your message and respond. Also, you are likely experiencing a similar problem as someone else. 3. Consider your hardware design early in the course so that you can order misc components such as extra wires, perfboard, resistors, stand offs, etc.
- it is not easy
- Think about the end-goal from Day1. It is not about this/next week's assignment submission. Even if you struggle initially to complete assignments, do it the right way.
- Try to work earlier, contact with teammates earlier.
- Definitely starting early. As a student who experienced undergrad in a quarter system school to now being in a semester school, the first half of the class ran extremely similarly to that of the quarter system: fast-paced, but the work is manageable as long as you start early. The amount of work you have in other classes is irrelevant as long as you manage your time well and appropriately. Starting early doesn't

necessarily mean finishing early, even if it is a plus!

- Put in the time to understand how things work and start on the assignments/projects right away. Ask questions.
- Wander through the code there's a lot to learn!! Just be curious and the rest will come super easily :)

---

Revision #5

Created 2 years ago by [Preet Kang](#)

Updated 3 months ago by [Preet Kang](#)