Due by 330 pm Tues. Feb. 11 on BCourses.

The purpose of the project proposal is to give you early feedback on your design ideas, to help to steer you away from some of the common design flaws. A good motto to keep in mind is "measure twice, drill once". The pace of the rest of the course is quite fast, and now is the time to plan what equipment you want, and where it should be mounted. The project proposal is a non-trivial amount of work; we would like to see 10 hours of effort per team member. Please read the NatCar rules from the UC Davis site. Here is an outline for you to follow:

1. Overall Strategy (1 page) (10 %)

1.1 (10%) Think about "winning" strategies. A "winning" strategy is one which results in a functional, thoroughly tested, and reliable vehicle with a minimum of effort. Separately list essential and "wouldn't it be great to have" features. A good rule of thumb is that you would like to be 80% complete with 20% of budget. Now speed is of course an important consideration, but speed without stability and robustness won't get you far. Will your vehicle rely on raw speed, expert navigation, expert braking, etc? How does your strategy affect the types of sensors and control strategies you will use? What type of camera will you use, fast line scan or slow web cam?

2. Hardware Design (60%)

2.1 Attachments to Vehicle (one page) (5%)

List every attachment you anticipate adding to your vehicle. (You are not committing to adding these devices, just leaving room for them if desired). Some possible things to attach: CPU board, IO board, battery pack, DC-DC converter, user interface, sensors, **crash bumper**, etc.

2.2 Detailed Mechanical Drawings of Vehicle (2 pages) (20%)

Draw a detailed layout of the vehicle with attachments from section 2.1. Include dimensions of circuit boards and vehicle. Expected detail is to level of screw holes for what you need to mount. Show location of flag mounting post, emergency stop switch, and battery pack. Do some rough sketches first: can you change the battery pack without unscrewing anything? Are switches for mode/power/reset readily accessible? Have you left room for expansion circuitry components if needed? (You can use ppt to sketch locations on top of provided drawings.)

Include a labelled photo which indicates good locations for mounting an encoder wheel for speed sensing. (5 pts)

2.3 List of Special Materials (0.5 page) (5%)

Do you need metal or plastic brackets? Metal or plastic plates? Special switches or connectors? If there is enough interest, we can arrange car pools to TAP plastics or Fry's. We could also put together periodic class mail orders to Digikey.

2.4 Motor Drive Circuitry (1 page) (25%)

Show a detailed schematic (example `BeagleBone_Blue_sch.pdf`) for your proposed motor drive circuitry, including values and pin #s. Be sure to include protection logic (if needed) and **e-stop** (15 pts). Draw expected parts layout (example layout: `BeagleBone-placement.pdf`). Is there room for heatsinks and mounting holes? Parts layout must be white background! Do not route or show copper (10 pts).

3. Software Strategy (1 page) (35%)

3.1 (10 pts) List anticipated IO needs and how they will be used. **Estimate number of lines of digital and analog IO you will use for each function.** Note BeagleBone Blue connectors being used (see `files\BeagleBone_Blue_sch.pdf`, and `files/BeagleBone_Blue_Pin_Table.xlsx` e.g. `GP0, GP1, ADC, etc.`)

3.2 (5 pts) List modules from `librobotcontrol` you anticipate using.

3.3 (10 pts) Make a first pass, high-level block diagram of the software system. (A block diagram should be like Fig. 5 in Thrun et al. [1] showing functional modules and their interconnection, **not a flow chart**.) Be sure to include UI/debugging block.

3.4 (10 pts) Make a table listing each module, key input/outputs, how often the module should execute, expected data rate for module input/output. For each module, specify whether the module runs as part of `main()` loop, a thread, or real-time in the PRU. List those things you need to understand more completely before you will be able to design your software.

References

[1] S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge." *Journal of Field Robotics,* 23(9), 661-692, 2006.