The T.S.A.R. project will build an open-source data acquisition and control system for the Portland State Aerospace society's liquid fuel engine test stand with the end goal of using the system on full-scale launches for LV4 and beyond.

1. Beagle-Bone Black/Raspi 3, Marionette DAQ Board, Python scripted PyQT GUI, safety features.
2. Project is innovative because it is OpenSource and robust. A quick internet search of 'open source rocket engine controls' shows that a robust, scalable solution with extensive safety features does not exist for the amateur domain. T.S.A.R. will be the first publicly available system and will greatly advance the state of the art for amateur groups seeking to use liquid fuel engines. (Discuss benefit of liquid over solid)
3. T.S.A.R. will focus on thorough documentation including videographic and photographic records. Each system will be designed with the express desire that other amateur groups can implement the system in under $1000 dollars and with low technical expertise (give resource list, etc.)
4.
5.

Counsel Questions

- Not super-sure there's much innovation here as opposed to just standard measurement. Better define or explain your innovation.

Nothing for our domain.

- Description of work insufficiently detailed to assess feasibility of project. What is your innovative contribution?

See above

- I appreciate the link to the cost spreadsheet, but it would be nice to just include the overall cost estimate as a hard number in your proposal (for those reading this offline)

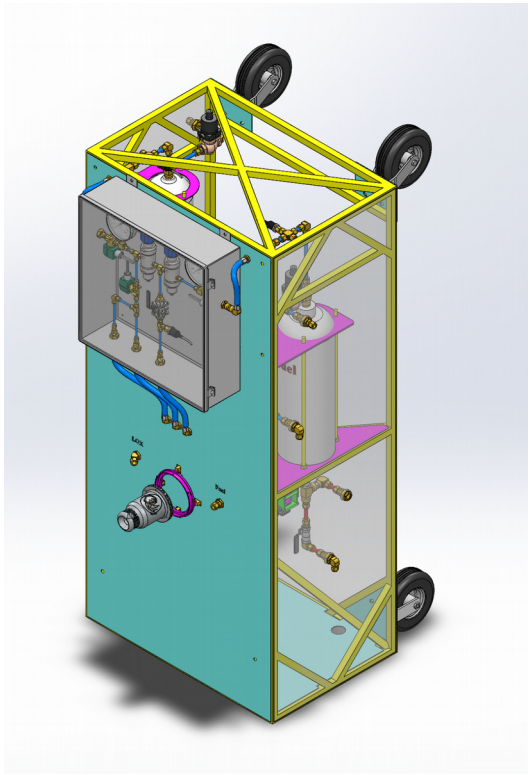
Atleast $1000, with a working

- What is an "in-house force sensor"? There are many, many commercially available pressure, force, temperature and flow rate sensors available. This seems like a routine specification of a DAQ system. What is unique about this application that requires an "innovative" solution? Or, in other words, what is innovative about this project physical data is being recorded? Flow rate? Mass in the tank as a function of time? Thrust? Temperature? Fluid pressure? The proposal mentions force and associates this with "cryogenic flow monitoring", but no concrete details are given. The team needs to find other members who have knowledge of the basic engineering data necessary to monitor and analyze combustion, flow, and thrust measurements. There

After discussing the ‘In-house force sensor’ with the mechanical team, that will not be our main focus as we ought to use off-the-shelf components AMAP.
NOTHING.IN.OUR.DOMAIN.SPACE.STOP.ASKING.

* Get details about the measurements. WE HAVE A SIT LOAD OF ME KIDDOS TWIDLING THEIR THUMBS OMG.

TSAR Test Stand Automation and Regulation.



The Finished Test Stand

What We Are An open-source solution to the data acquisition and control requirements of the Portland State Aerospace Society's liquid fuel engine test stand with the future goal of being a part of the avionics system for future launches.

Our Innovative Contribution

A quick Google search of 'Open source rocket engine control' yields either space agency level solutions[1], unsafe home-brew solutions[2], or poorly documented systems[3]. We will focus on documentation, ease-of-use at the amateur level, and scalability of implementation. In addition, a unique thrust plate (where the engine attaches to the test stand) is being developed by an LFETS masters team member and unique load cells must be designed and integrated with in order to gather force data from the engine.

Cost

1. Sensors ($700 Pressure transducers, thermocouples)
2. Misc. networking paraphernalia ($200, wireless cards, etc.)
3. Power ($100),, Connectors (heat sensitive)

Short term Goals

1. Front-end design tool-set well defined.
2. State machine description, sensor descriptions, failsafe systems.
4. Parallel development in support of LFETS, testing the igniter, injector, etc.

Hot fire by spring term (June 2018)

---

1 NASA Data Acquisition System Software Development for Rocket Propulsion Test Facilities - https://ntrs.nasa.gov/search.jsp?R=20150010686

2 Open Source 3D Printed Rocket Engine controlled by Arduino - https://tinyurl.com/ybj8tbje

3 https://github.com/cuspaceflight/m3-avionics (Note the complete lack of a readme file indicating poor documentation)