An IoT solution for the electronics for a N2O Cooler

Author: Fadi Saad Bunni (fsb@danstar.dk) — Network technology & IT (BSc)

Advisor: Ying Yan

20/2/2018

Introduction

At the Danish Student Association for Rocketry (DanSTAR), it was decided that three main projects; a Test stand, a Demo Engine and an N2O Cooler should be initiated. The main goal was to design, develop and test a N2O cooler for liqifying nitrous oxide used as an oxidizer for a DanSTAR developed bi-propellant rocket engine. This poster takes holds in the N2O Cooler, specifically the electronics and the control of the cooler.

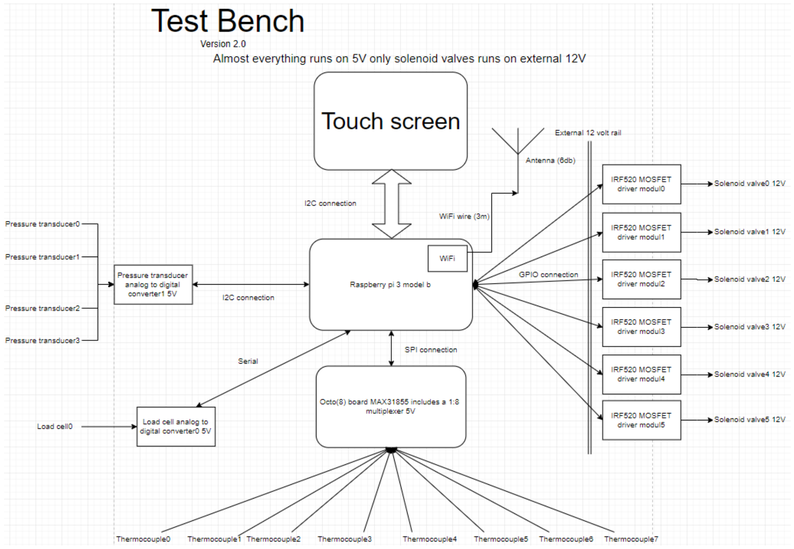
The requirements for the electronics for the N2O cooler:

- The N2O cooler needs to be controlled both automatically and manually
- All the data read by the sensors has to be presented with a neat GUI
- The N2O cooler has to be able to use its sensors to monitor and control itself, by reveicing the data from the sensors so that it can analyse them and act on its given state
- To control itself; a state machine is required
- To control it manually; it needs to have a working GUI that shows the necessary information
- The console and GUI has to be visualized on an on-board touch display or on a website set up by the on-board controller
- All the sensors have to be controlled by just one board (microprocessor)
- The electronics needs to be able to log all the data on a flash drive

For automation of the N2O cooler we need to look into using a state machine, which job is basically to figure out what the cooler should do about itself on a given state and what it should do if a sensor triggers a state change to a new one.

The requirements for the state machine (software wise):

- If the weight reaches a specific amount activate a given solenoid valve
- If a given thermocouples reaches to a specific temperature activate a given solenoid valve
- If a given pressure transducer reaches a specific pressure activate a given solenoid valve

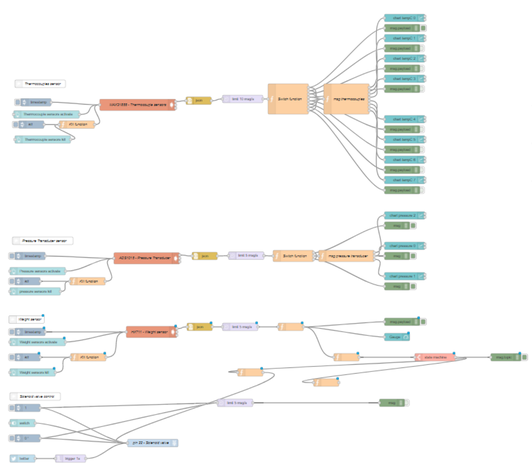


Above it is seen the overview of the electronics; the sensors, boards and solenoid valves used.

Implementation

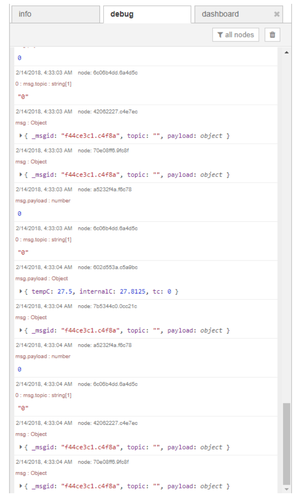
For implementation all the sensors were added to a Raspberry Pi 3 Model B,

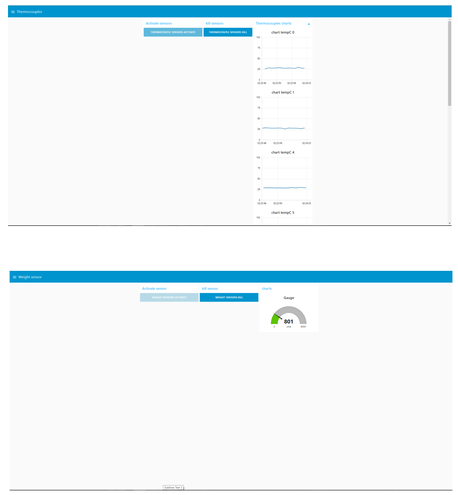
Embedded programming languages were used for reading the output from sensors. High level programming languages were used for running the state machine and controlling the solenoid valve furthermore to showcase the GUI interface.



Node-Red, a flow based programming language, was used to connect alle the sensors together with the state machine. To the left is an overall flow diagram for the alle the sensors, valves, and including the state machine.

Results





Above is the result that was aimed for in the introduction section. We can see the printouts for all the data from all the sensors and valves. Altso the GUI presents the data for the viewer in a better way.

Conclusion

It was concluded that:

- The cooler could run automatically and manually.
- A GUI was used and worked well for presenting the data, and control the cooler manually.
- A simple state machine was implemented and worked as inteted but only using one sensor to control the valves.
- Node-Red used for flowbased porgramming is good program for fast implementation and prelimiray designs.

Contact information

If you wish to help out DanSTAR with building rockets . Send us a mail—se below.

Website: www.danstar.dk

Email: contact@danstar.dk