System backup ideas 3/5/2017

Overview

The purpose of this document is to describe backup ideas for our 549 embedded system, in the case of module failure.

We divide our system up into several subparts, which are all linked to a microcontroller, which eventually impact the user through the user application. This system breakdown is shown in Figure 1.

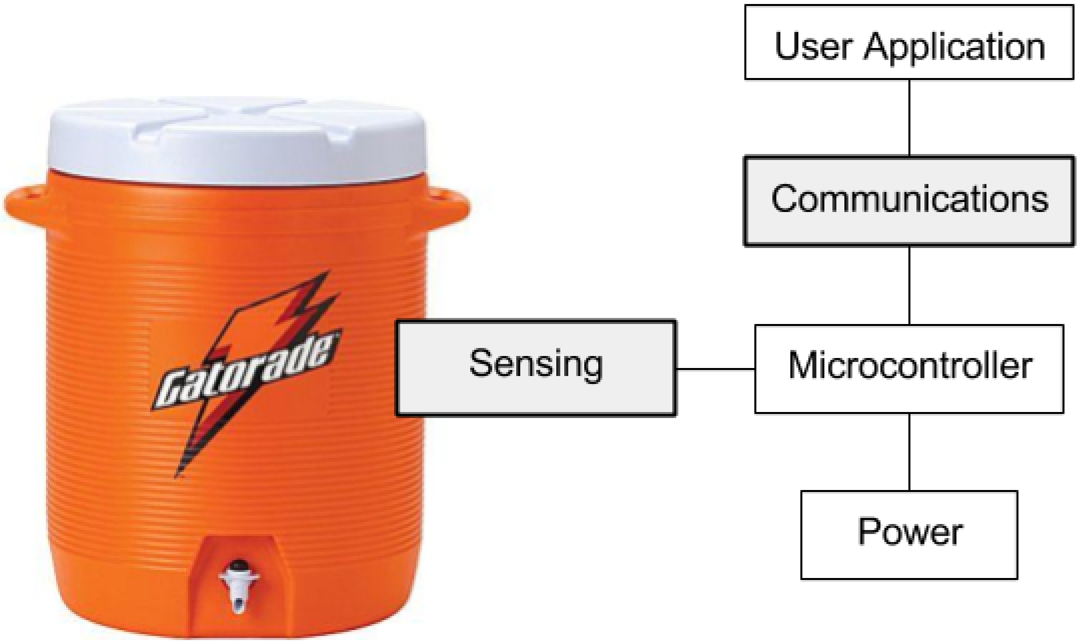

Figure 1: Overall System Design

Since failure necessitating redesign is most likely to be isolated in these subparts, particularly sensing and communications, we describe different backups for these subpart.

Sensing Backup

Our backup for sensing is to use a flow rate monitor to check how much liquid is coming out of the container. We would also include two buttons: one to indicate to the system that the container has just been refilled, and one to help the system calibrate the empty level. The user would start with a full container, and press the “full” button. When the container is emptied for the first time, the user would press the “empty” button; at this point, the system would know how much liquid flows through the exit valve to make the container empty. After this calibration sequence, the system would be able to notify the user when the container is getting close to empty, through the communications subpart and user application. This system design would require that the user press the “full” button every time they refill the container.

There is room to expand the design to create more of a spectrum of fill, so the user could indicate to the system that they had partially refilled the container. Modifying the buttons, the system could require the user to press and hold a button, with a longer hold indicating more full.

However, for the base level backup, the system would not handle partial refills. The backup system is shown in Figure 2.

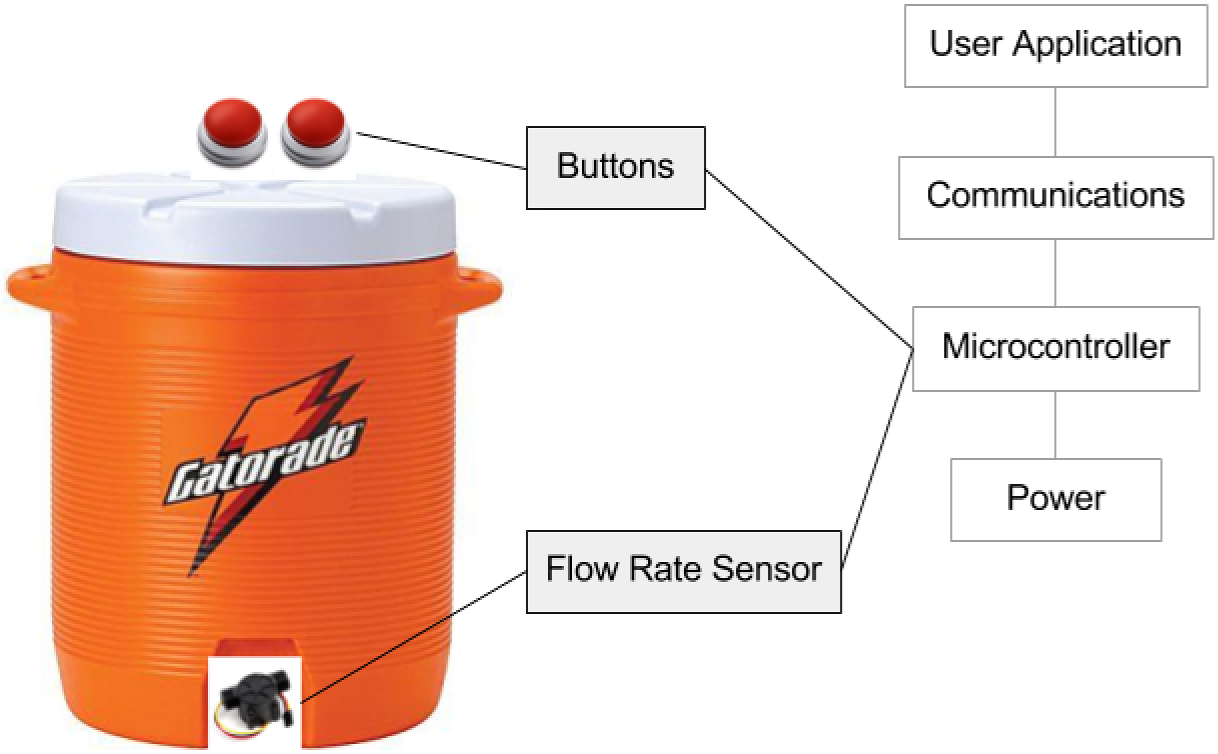

Figure 2: Sensing subsystem backup

Communications Backup

Our system's current communications scheme is to communicate to a backend web server using WiFi, and then communicate between the user application and the backend using traditional internet WiFi communication. If this fails, our plan is to cut out the backend web server, and communicate directly with the phone using WiFi.