Project Documentation

Embedded Linux

Robert Wieland, Kevin Anderson, Michael Divitto, Talita Atahalpa, Bryan Darcy

5/1/2015

Contents

Group Members

Rob Wieland – Group Leader, Software/Python Code, Assembly
Kevin Anderson – Wiring and Power
Mike Divitto - Documentation
Talita Atahalpa – Software/Website, Documentation
Bryan Darcy – Wiring, Assembly

Project Description

Develop a system that can monitor temperature at different depths in a lake at regular intervals using a buoy model. The system should store the data locally until a network connection is available, then it will transmit that data to a central repository. End users will be able to access the data via a Web Interface that retrieves the data from the Central Repository. The system should be powered on its own, with enough power to run all the components.

Project Goals

- Construct a waterproof, floating Buoy
- Read temperatures at specified levels
- Manage data both locally and on a central server
- Serve the information to the end users via a Web Interface
- The Buoy should be extensible to allow other sensors to be attached
- Self-powered with solar and batteries

Components

Hardware

- Raspberry Pi
- 5 Temperature Sensors
- Cables (5-wire 18-awg)
- Solar Panels
- Batteries
- Bucket / Container (Weatherproof)
- Motor
- Breadboard, wiring, etc.
- Rope and Anchor
- Floating structure
    - Water Bottles, Plexiglass, Duct Tape

Software (Python/HTML Code)

- Read sensors and store in database (SQLite3)
- Sync data to the repository
- Display the data in a website

Implementation

There were two possible implementations that we planned. The major difference between the two is whether we have one sensor that moves up and down with a motor, or several sensors (5) that are on a fixed length of rope.

The common features will include:

- A bucket which contains the Pi and the motor.
- A floatation device (Soda Bottles and Plexiglass)
- A guide rope that the sensor(s) will be attached to or guided on
- Power will be supplied by a combination of batteries and a solar panel

Implementation 1

The first implementation utilizes a motor to be used as a winch. The idea is to have a wire hang down with an attached anchor, then the temperature sensor connected to another wire would go up and down the main wire. The main wire is only used to guide the moving wire. Figure 1 shows how this is intended to be set up.

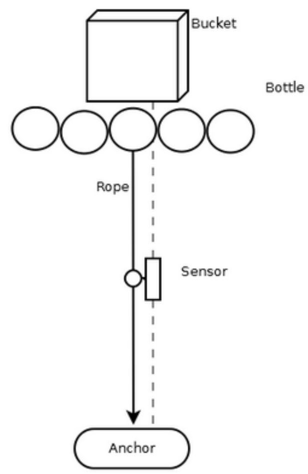


Figure 1. Diagram of the first implementation

Implementation 2

The second implementation does not use a motor, but instead relies on several sensors attached to a fixed length of wire. The anchor will still be attached to a rope to keep the length of wire underwater. Benefits of using this implementation would be less power consumption and it would be less likely that the wiring would get tangled with the motor system.

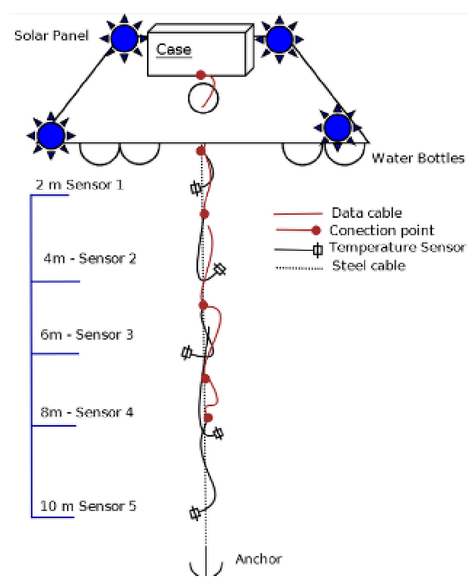


Figure 2. Diagram of the second implementation

Wiring and Power

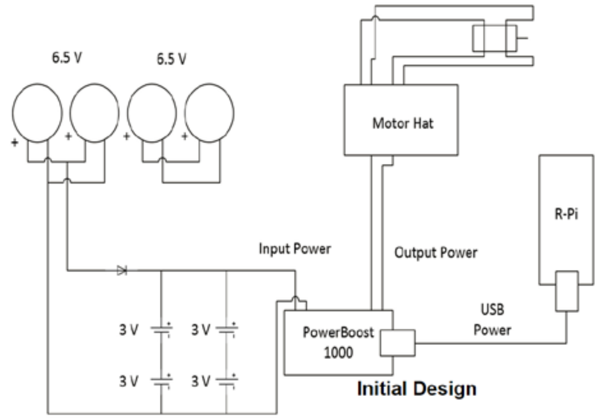


Figure 3. Common Power System

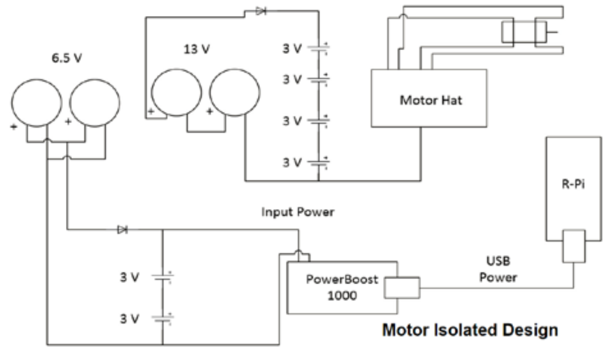
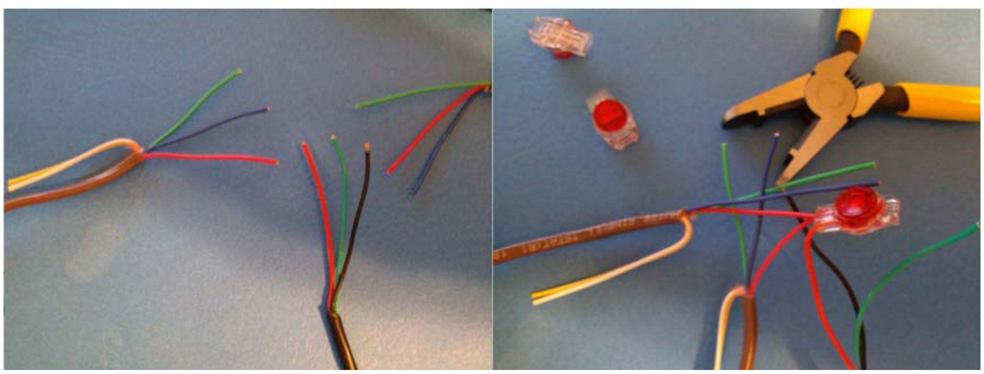


Figure 4. Separate Power System

The initial Power System had all the components on a common system. This was to make the wiring easier, and everything went through the PowerBoost 1000 as shown in Figure 3. We ran into some difficulties trying to power all the systems at the same time, so we decided to isolate the power systems so that the Pi was powered separately from the HAT/motor.



The main wire that connects to the sensors is a 5-wire 18-awg wire. The sensors are spliced into the cable using the clips in Figure 5. Since the 1-wire system allows multiple sensors to be on the same line, this worked out well for us.

Software (Python and HTML)

A Python script is used to read the temperatures and store them to a local database. The script is registered as a cron job, since we are utilizing a linux platform on the Raspberry Pi. By using a cron job, it lets us specify a time interval to run the script, and allows the script to be run again even if it crashed later on. We set the cron job to run every 15 minutes.

The Web Interface displays the graphed data at each level. A Google Map is provided on the site to show where the location of the buoy is. This does not yet utilize GPS functionality, but a location is manually entered just so the user has an idea of the location these readings are taking place. Finally, the site also shows team member information as a “credits” for the project.

Review

We had a mixed group of Computer Science and Computer Engineering students, which made the project easier when it came to things like assembling powering the Pi, then writing software to make it work. Most of the time was spent planning out the details on the build of the project, making it easier to implement. Collecting materials was made easy by utilizing items that are easily in reach such as Soda Bottles or Solar Panels and wiring from Resnick. The only things that took some time were those that we had to buy online or in store such as the bucket, wire, and spool.

We did run into several issues with the motor and getting it to work. We spent several hours trying to get it to work without success. We then decided to switch to Implementation 2 in order to compensate for lack of time left. We were later able to get the motor to spin by asking around, and found that it works when plugged into the other port. However, since we already implemented and assembled the Implementation 2, we stuck with that. The motor can be used as possible expansion for other sensors.

Future Goals

In the future, we want to be able to expand the sensor capabilities to add in pH sensors as well as other sensors.