IoT Water Level Monitoring System for High-Hazard Dams

Corinne Smith[1], John McCain[3], Austin R.J. Downey[1,2], Jasim Imran[2]
[1]Department of Mechanical Engineering, University of South Carolina, Columbia SC
[2]Department of Civil and Environmental Engineering, University of South Carolina, Columbia SC
[3]Dam Safety Program, South Carolina Department of Health and Environmental Control, Columbia SC

Background

- Traditional gages are large and permanent
- Tend to be more expensive
- There is a need for a low-cost and easy to deploy sensors for state regulated dams
- South Carolina Department of Health and Environmental Control Dam Safety Program (DSP) has been working on an open-sourced IoT water level sensor
- DSP design goals:
  - Less than $1,000 per unit
  - Flexible deployment locations
  - Quickly deployable
  - Real-time monitoring capabilities
  - Email alerts
  - Multiple sensing methods

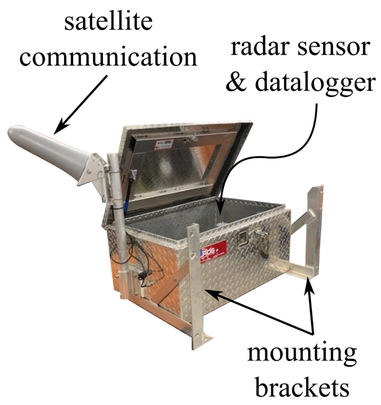

Figure 1. USGS Rapidly-Deployable Gage (RDG).

Hardware

- Based on Arduino open-source hardware and software
- Uses cellular connectivity to Internet of Things
- Measures water level with ultrasonic sensor and an in-water pressure transducer
  - Allows contacted and contactless readings
- Custom printed circuit boards (PCBs) allow compact design
- Powered by 12V Li+ battery charged by solar panel
- Small, portable, waterproof housing

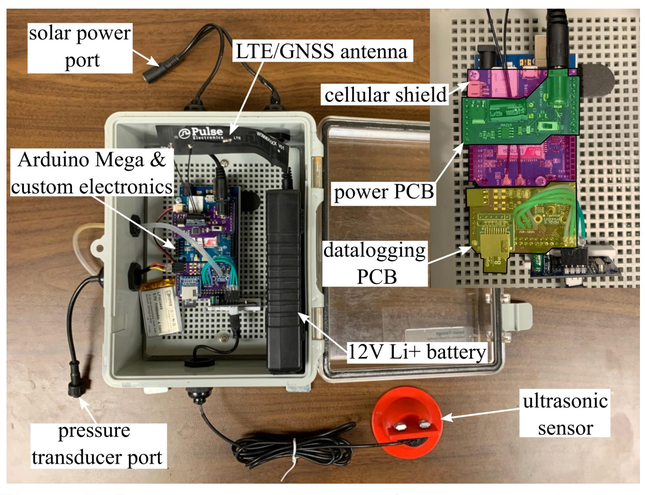

Figure 2. Internal components of remote monitoring system.

Data Collection and Transfer

- Remote monitoring system is connected to the Internet of Things (IoT) via cellular shield
- Message Queueing Transport Telemetry (MQTT) protocol is ideal for lightweight data transfer
- Adafruit IO is a cloud service that acts as an MQTT broker with a Graphical User Interface (GUI)
- MQTT uses publish/subscribe procedure
  - Publish = data from sensor to GUI
  - Subscribe = data from GUI to sensor
- User can control and view data from remote monitoring system from GUI
- Real-time monitoring is achieved
- Basic logic of the system:
  1) User initializes sampling rate and current water elevation in GUI
  2) System receives parameters from MQTT broker and waits until the user toggles the deploy switch on
  3) System collects data and publishes to MQTT broker, which displays data on GUI

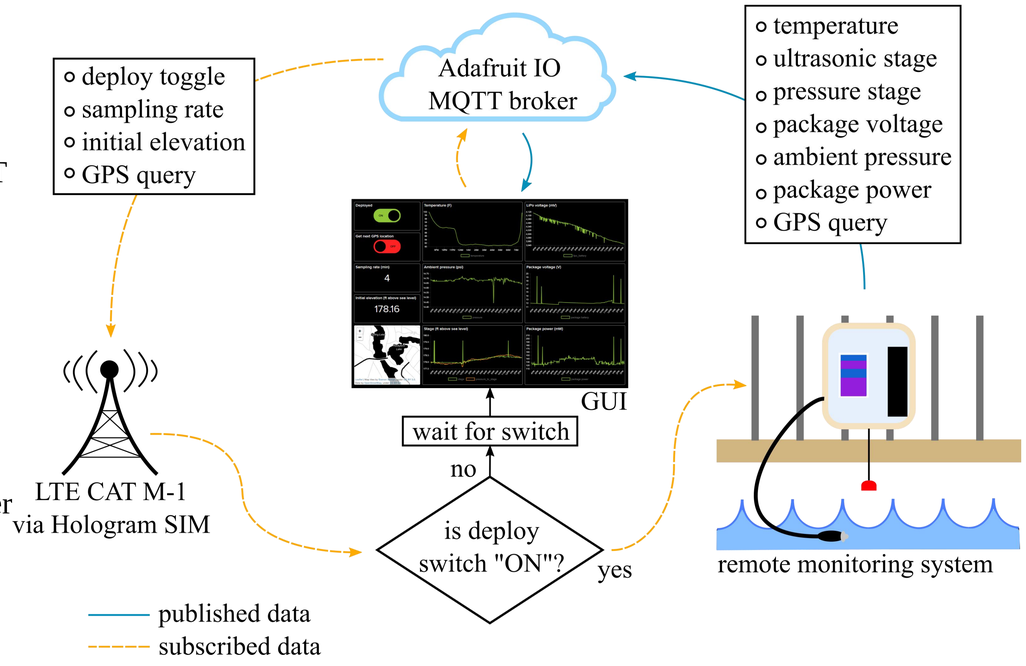

Figure 3. Data collection process with MQTT broker.

Testing and Results

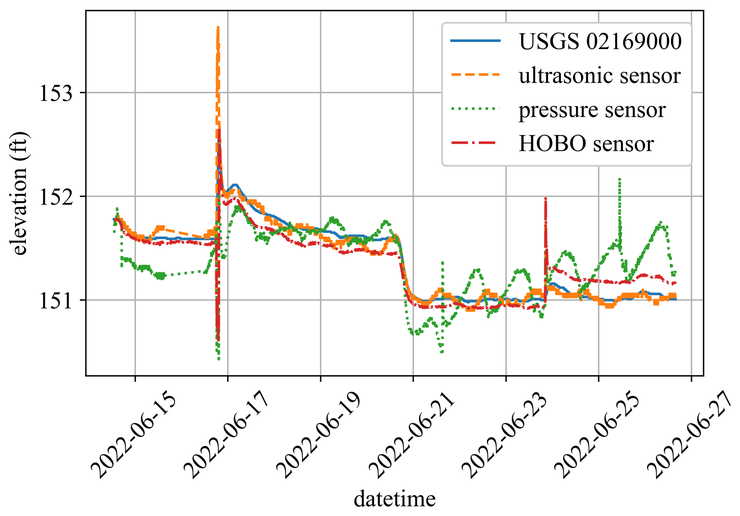

Figure 4. Field test at Saluda Riverwalk, Columbia, SC.

- Hydrodynamic test area to verify ability to capture changing water levels
- Validated by USGS gage and a commercial water level logger (HOBO)
- Both sensors remain relatively consistent
- Pressure sensor appears to experience high drift
- Data sent consistently to MQTT broker

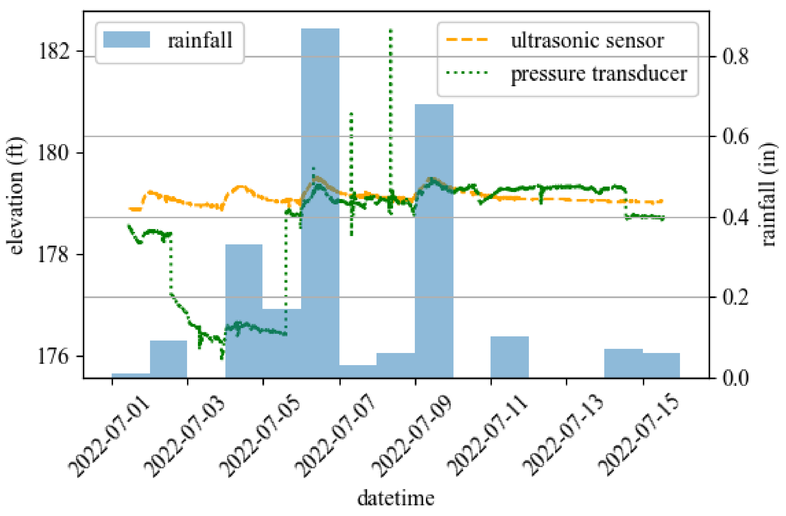

Figure 5. Field test on Spring Lake Dam, Richland County, SC.

- Dam test area to verify application to dams, which experience much lower water level fluctuation
- No USGS gages to validate readings, so sensor data is compared to rain gauge data
- Readings are generally consistent with one another
- Pressure transducer was tampered with during deployment, causing sudden drop until recalibration

Conclusion & Future Work

- The DSP remote monitoring system proves promising when tested against traditional gaging technology like the USGS gages and HOBO water level loggers
- Application for both hydrodynamic and dam environments are feasible
- Sensor meets cost, portability, open-source, real-time monitoring, and serviceability design goals
- Future work will focus on better pressure transducer readings, longer lasting deployments, and networking multiple sensors

References

1) CoCoRHAS. "Daily Precipitation Report for SC-RC-139". Colorado Climate Center, Retrieved 18 July 2022 from https://www.cocorahs.org/ViewData/ViewDailyPrecipReport.aspx?DailyPrecipReportID=f780e58e-d180-40dc-b3b5-c8a8659ad7f4
2) United States Geological Survey. "USGS 02169000 SALUDA RIVER NEAR COLUMBIA, SC." National Water Information System, Dominion Energy, 14 July 2022, https://waterdata.usgs.gov/usa/nwis/uv?02169000.
3) GitHub: Smith, C., Lovett, P., McCain, J., & Downey, A. (2022). IOT-Cellular-Dam-Water-Level-Sensor [GitHub Repository]. https://github.com/smitty444/IoT-Cellular-Dam-Water-Level-Sensor