Solids Treatment and Remineralization (STREM) System version 2

Engineering sketch

Victor Lobanov

Department of Marine Sciences

University of Gothenburg, Sweden

Mobile +46 76 618 30 69

[victor.lobanov@gu.se](mailto:victor.lobanov@gu.se)

UW visiting address:

University of Washington

3900 7th Ave NE

Seattle WA 98195

Attn: Gundula Proksch, box 355720

Contents

[Objective 2](#_Toc71388499)

[Design requirements 2](#_Toc71388500)

[Defining the features 3](#_Toc71388501)

[Timeline 5](#_Toc71388502)

[Renumeration 5](#_Toc71388503)

# Objective

Control 7 devices (5 peristaltic pumps, 2 solenoid valves) and receive input-only from two devices.

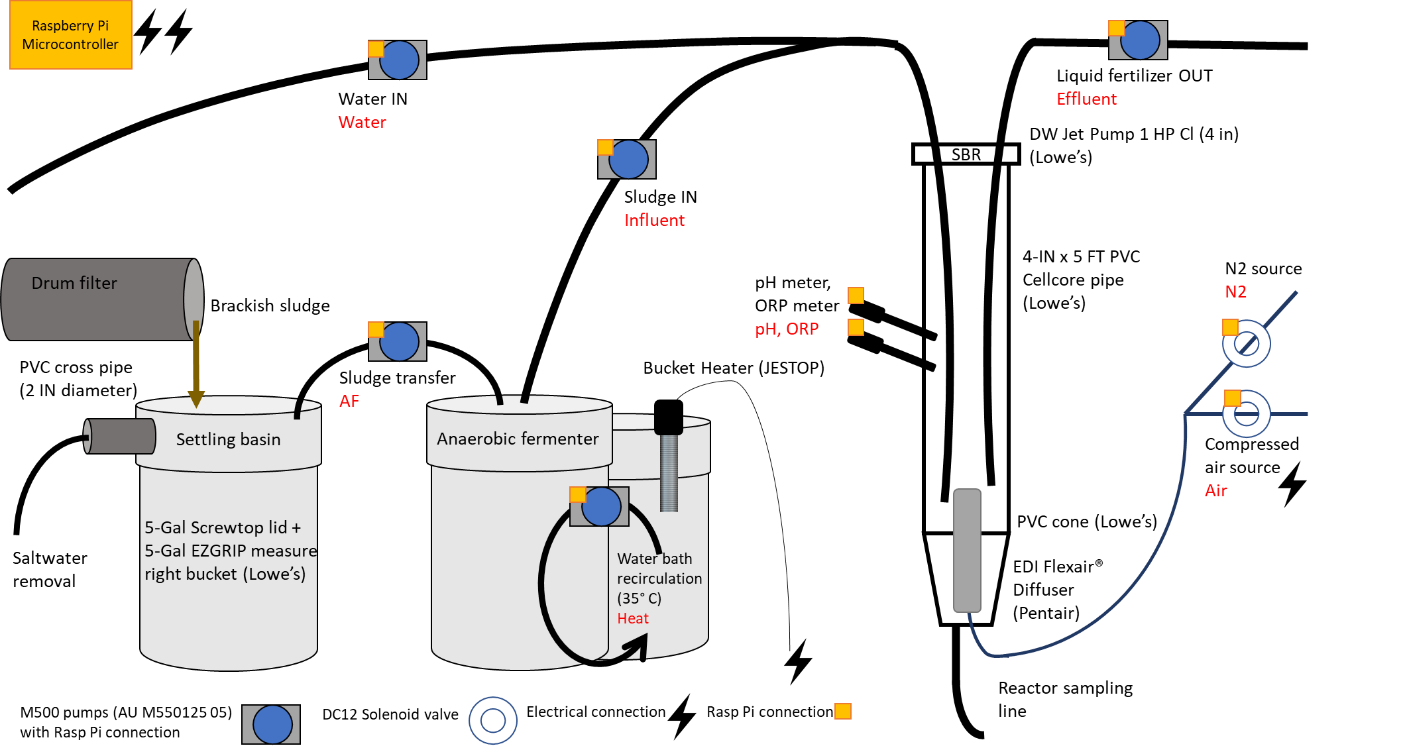


Figure 1. Sketch of STREM with Raspberry Pi connection indicated.

# Design requirements

1. The ability to add/ remove “features” comprised of “stages”, change the “duration”, and “stage mode”. Stage refers to the order the action occurs in, duration is the length of time (seconds), and stage mode refers to the type of action. Features keep cycling until the program is manually stopped.
2. Send out signals to 7 [GPIOs](https://en.wikipedia.org/wiki/General-purpose_input/output).
3. Receive input from 2 GPIOs (pH, ORP). Save the data in an output file in the format of: time stamp, value.

Allow time stamp frequency to be set manually, default every 5 minutes beginning at time zero.

1. Integrating a GUI:

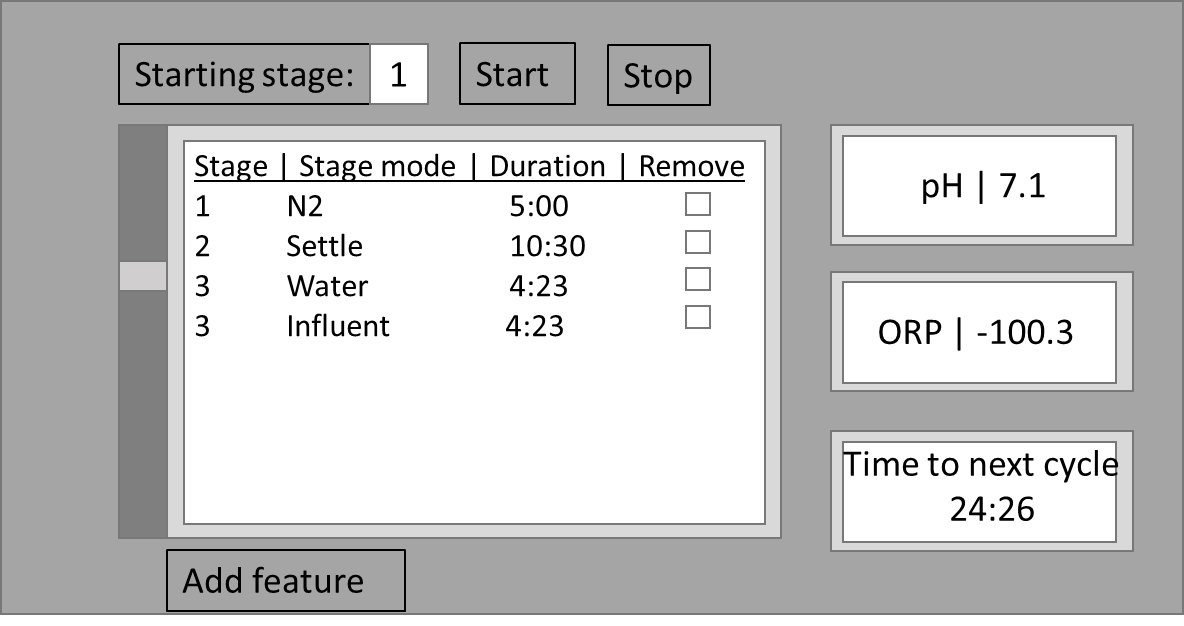


Figure 2. STREM GUI.

The GUI has the following features.

1. Starting stage refers to the stage the cycle should begin on. The default should be 1, however any existing stage should be an option.
2. Start button to begin at the starting stage.
3. Stop to halt the program.
4. All features are listed in the large box. A scroll wheel allows the user to scroll down if there are more features than fit on the page. The three properties of each features are listed in each row with the option to delete the feature.
5. Below is an option to add a new feature, at which point the user should be prompted to fill in the stage, stage mode, and duration.
6. Next are the two inputs, pH and ORP. The last data point collected is displayed.
7. The time to next cycle is defined as: (the sum of all durations for each feature) – (total duration completed). It should be represented as a minutes: seconds counter moving towards zero then restarting automatically. The timer should always refer to the time required until the stage resets to 1, regardless of the starting stage.

# Defining the features

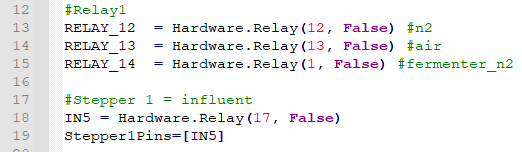
* The stage modes are currently stored in the file “normal.json” and are listed as: influent, effluent, n2, still, and air. Each stage mode (green highlight) is associated with a stage (yellow highlight), and a duration (blue highlight), see file.

"stage": 1,

"effluent":300

For version 2, however, stage modes are expanded to include: Water, Influent, Effluent, AF, Heat, N2, and Air (red text, figure 1).

* Each stage is coupled to a single pump. In the current program that description can be found in “main.py”. The script has the following components:



Here the GPIOs are associated with the relay. Physically this appears like in figure 3. Activating the GPIO associated with a relay pin sends a 3V current to the relay which open an electrical gate, allowing the power source to connect to the feature (pump or solenoid valve).

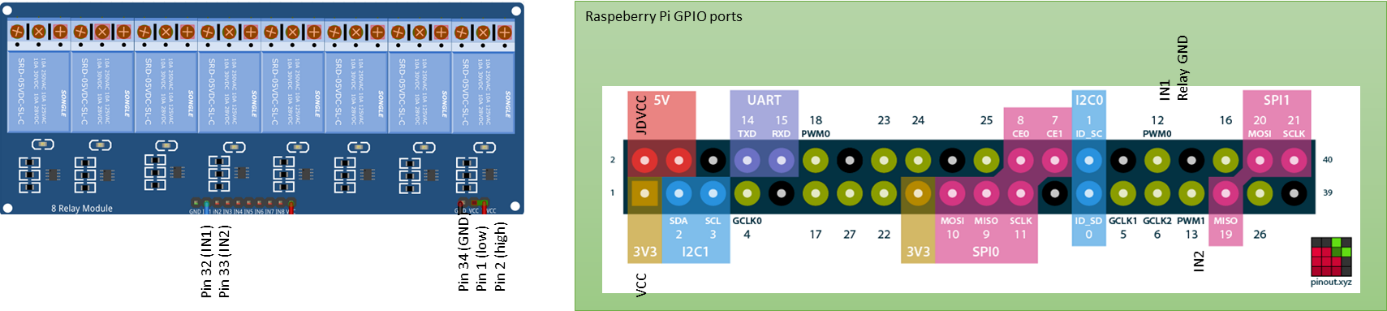
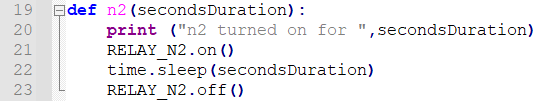
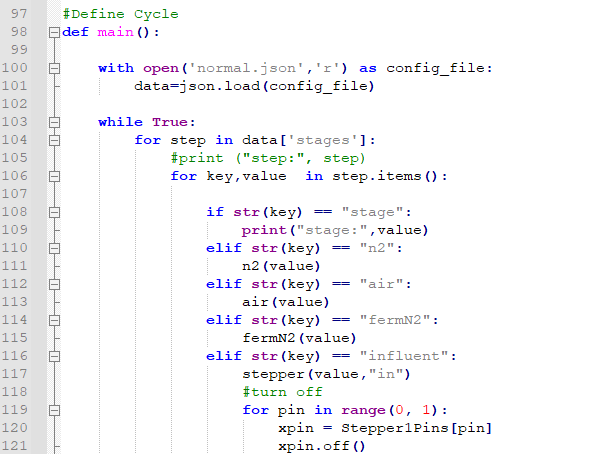


Figure 3. Relay and RPi pinout association.

Next, each of the features are defined:



This is as simple as pinging the GPIO, waiting for the duration, and then pinging the GPIO again to turn it off.



The function “main” imports the json config file, then cycles through the stage modes (here referred to as steps). **Note:** If it is complicated to integrate adding a stage mode into the GUI, we can keep it simple as done here where each stage mode is written into the function “main”.

* Version 1 did not have a pH or ORP probe. I bought a pH and ORP probe from this retailer:

<https://www.anyleaf.org/blog/measuring-ph-on-raspberry-pi>

It looks like it will be fairly simple to set up.

# Timeline

The project can be broken into two phases: implementing the processes and implementing the GUI. As the processes are more critical and likely less work to do as much of the structure is already complete, we should complete the testing by June 7th. This way, we can directly test how the functions work when the Raspberry Pi is connected to the relay (intermediary for the pumps/solenoid valves) and probes.

This likewise makes phase 2 less time sensitive as I could theoretically begin working with the program before it has a friendly GUI.

# Renumeration

Through Future Earth stipend (UW budget number: 63-5042) we will pay $20 per hour work. Work here is defined by demonstrable progress towards the completions of the project objective.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ben Nachmanson

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Victor Lobanov