**EcolibriumReef - Arduino**

This is an aquarium controller based on Arduino. Its functions are described below.

1. Timer Switches

Uses a DS3231RTC to keep time and activate/deactivate relays to turn lights on/off. The system displays PDT accordingly but only keeps PST. Timers are also on PST (We don't want to confuse the fish).

1. Cooling System

Uses a DS18B20 sensor to detect water temperature and control two fans with relays. Fan1 is turned on when temperature reaches the first threshold (81.0°F). Fan 2 is turned on when temperature reaches the second threshold (81.5°F). Both fans will continue to run until temperature drops below desired temperature (80.5°F).

1. Auto Top-off

Uses a float switch to detect water level. A LOW reading will activate relays to open solenoid valves connected to a water source and fill up the aquarium. When water reaches desired level (float switch reads HIGH), relays are deactivated and solenoid valves closed.

When float switch reads LOW, relays are activated immediately without debouncing and will run for at least half the maximum duration set by the user. This is to prevent bouncing conditions caused by the circuit or when float switch is in between HIGH and LOW.

As a safety feature, Auto Top-off also runs on a timer where a maximum valve opening duration (in minutes) is set by the user. In case of float switch malfunction, Auto Top-off will be disabled when the timer reaches maximum duration. A manual reset is required. In case of a power outage, everything will be reset. Since I am connecting my system to an endless water source, I have two solenoid valves connected in series to the inlet of the RO/DI system to open/close the water source. Two more solenoid valves connected in parallel to the outlet of the filtration system to direct water either to the aquarium or to the drinking bottle (a three-way solenoid is a lot more expensive). I figured a float switch, a timer and three solenoid valves, each activated by a separate relay coil, should be enough to prevent a flood.

1. Auto Top-off Bypass

Divert water from RO system to drinking bottle. This is on a timer.

1. Remote Control

Uses Bluetooth serial HC-05 to communicate with an Android app (only tested on Samsung Galaxy S9+ and S10+).

The Android app can:

* + - Enable (reset) Auto Top-off when it's disabled by the system.
    - Disable Auto Top-off for RO/DI maintenance.
    - Turn on/off Auto Top-off Bypass for drinking water.
    - Set Auto Top-off maximum duration.
    - Set Auto Top-off Bypass duration.
    - Test all relays, which control lights, fans, solenoid valves, individually.

An alternative to Bluetooth is infrared remote control. By using an infrared receiver, you can read the code emitted when pressing a button and use that code to control certain function of the controller. A TV remote control will work.

1. Display

Uses a LCD 2004 (20 columns, 4 rows) to display date, time, water and air temperatures, (ATO) auto top-off maximum durations, (RO) auto top-off bypass duration, relays states (lights, fans, solenoid valves).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 0 | W | E | D |  | 0 | 3 | - | 1 | 1 | - | 2 | 0 | 2 | 0 |  | L | B |  | L | W |
| 1 |  |  | 0 | 9 | : | 0 | 3 | : | 1 | 6 |  | P | M |  |  | F | 1 |  | F | 2 |
| 2 | W | : | 8 | 0 | . | 1 | ° |  | R | O | : | 6 | 7 | ' |  | V | 1 |  | V | 2 |
| 3 | A | : | 7 | 0 | . | 3 | ° | A | T | O | : | 2 | 0 | ' |  | V | 3 |  | V | 4 |