

Projekt **Akwarium**

Dokumentacja

Programowanie Współbieżne i Rozproszone

Arkadiusz Pasek

Technologia: **erlang**

16.01.2021

Opis projektu

Celem projektu jest utworzenie aplikacji do zarządzania inteligentnym akwariem. Aplikacja składa się z głównego modułu i kolejnych współbieżnych modułów komunikujących się z innymi połączonymi modułami.

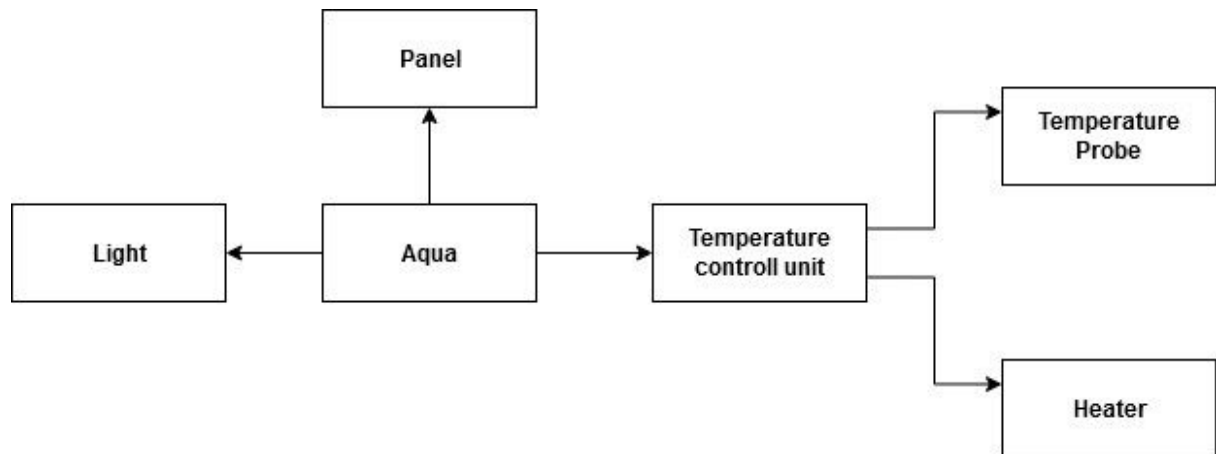


Diagram przedstawiający zaimplementowane moduły

Moduły

Aplikacja składa się z modułów:

- **aqua** - moduł główny tworzący proces główny agregujący komunikację pomiędzy wszystkimi procesami zależnymi oraz same procesy zarządzające innymi modułami
- **panel** - moduł odpowiedzialny za interakcję z użytkownikiem w założeniu docelowym otrzymywałby informację z panelu zamieszczonego w wybranym miejscu na akwariu i po wciśnięciu danego przycisku lub wprowadzenia informacji wysyła informacje do modułu aqua, który następnie decyduje co z nią zrobić. w aktualnej wersji symuluje taką interakcję poprzez poczekanie zadanego czasu i wysłanie informacji o zaświeceniu podświetlenia
- **temperature control unit** - moduł odpowiedzialny za utrzymanie odpowiedniej temperatury w akwariu, jest modułem stanowym zawierającym informacje o aktualnym stanie grzałki, sam aktualnie zawiera dwa podprocesy które komunikują się z nim: miernik temperatury oraz grzałka, implementuje logikę utrzymywania temperatury akwariu na poziomie 20 stopni, odpowiednio otrzymując powiadomienia o temperaturze i sterując grzałką
- **temperature probe** - moduł odpowiadający za zbieranie informacji z czujnika temperatury zamieszczonego w odpowiednim miejscu akwariu, aktualnie nie przeprowadza interakcji ze sprzętem fizycznym, symuluje odczyty w

postaci generowania pomiarów bliskich docelowej temperaturze z pewnym odchyleniem, na które reaguje jednostka kontrolna

- **heater** - moduł odpowiadający za generowanie ciepła w akwarium, realne wdrożenie może zawierać wiele takich procesów odpowiadających za kolejne grzałki, zarządzaniem i synchronizacją ich zajmuje się jednostka kontrolna temperatury
- **protocol** - procesy komunikują się wysyłając do siebie asynchronicznie informacje znając swoje pid, ten moduł zawiera funkcje pozwalające utworzyć pakiet danych do wysłania, dzięki temu lepiej widać w jaki sposób tworzone są wysyłane dane

Komunikacja

Procesy komunikują się odpowiednio zaimplementowane zapamiętywanie i przekazywanie sobie pid modułów, realizowane poprzez wysyłanie erlangowych wiadomości

Przykłady przesyłanych informacji

Informacja przesyłana przez grzałkę do jednostki kontrolnej informująca o swoim stanie:

{data, heater, State} - gdzie State jest stanem grzałki, aktualnie mówi o tym czy jest włączona oraz na jaką moc jest skonfigurowana

{action, light, on} - informacja informująca o akcji przesyłana z panelu do modułu aqua, który następnie przetwarza ją wysyłając informację o zapaleniu światła do modułu light

Oprócz tego procesy przekazują sobie informację o {error, ..} przy pewnym błędzie w funkcjonowaniu systemu

Rozwój

Program przedstawia działanie inteligentnego akwarium w dość minimalistycznej wersji, przykładowe wdrożenie może zawierać więcej komponentów, na przykład jednostka temperatury mogłaby składać się z wielu grzałek osobno zarządzanych, wielu mierników pozwalających na asymetryczne ogrzewanie, więcej akcji dostępnych na panelu kontrolnym użytkownika.

Aktualnie aplikacja wypisuje co dzieje się na poszczególnych modułach, pewnym udoskonaleniem byłoby co najmniej dodanie jak poważne komunikaty powinny się

wyświetlać (np. debug, info, warn, error: czy użytkownik chce wiedzieć o wszystkim co się dzieje czy tylko o najważniejszych zdarzeniach lub błędach) jeśli system byłby zarządzany np przez aplikację smartfonową, jeśli nie należałoby dodać moduł wyświetlacza, który byłby odpowiednio aktualizowany z aktualnym stanem.

Oprócz tego należałoby dodać większą obsługę błędów, ciężko to aktualnie określić, gdyż błędy prawdopodobnie wynikałyby z problemów ze sprzętem fizycznym i odpowiednio na niego reagować, powiadamiając użytkownika (na wyświetlaczu albo w aplikacji)