Szegedi Tudományegyetem

Informatikai Intézet

A képen szöveg, embléma, címerpajzs, jelvény látható

Automatikusan generált leírás

Akvárium automatizálás és távfelügyelet megvalósítása

Implementation of aquarium automation and remote supervision

Szakdolgozat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Készítette: |  | Témavezető: |  |
|  | Takács Ábel |  | Dr. Mingesz Róbert Zoltán |  |
|  | mérnökinformatika szakos hallgató |  | egyetemi docens |  |

Szeged

2024

**A feladat kiírása**

Ahhoz, hogy egy akvárium mikroklímája megfelelő legyen, számos berendezés kell összehangoltan működjön. Ilyenek pl. a vízforgató és szűrő berendezések, fényforrások, fűtőszálak, ventilátorok és CO2 adagoló rendszerek. Ezek egy jó részét egyszerű időzítőkkel szokták vezérelni, működésüknek a monitorozása pedig általában nincs megoldva. Számos műveletet (pl. halak etetése, vegyszerek adagolása) pedig csak manuálisan lehet elvégezni. A hallgató feladata egy olyan rendszer megvalósítása, mely egyrészt vezérelni tudja a különböző eszközöket, másrészről pedig monitorozza az akvárium állapotát, a mért paramétereket pedig távolról is le lehet kérdezni. A megvalósítás során figyelni kell az üzembiztonságra, vagyis a rendszer semmilyen körülmény között nem fagyhat le. A következő funkciók jöhetnek szóba: • Hőmérséklet és fénymennyiség monitorozása • Világítás ki/be kapcsolása, több világítótest • Ventillátorok ki/be kapcsolása • Folyadékszínt monitorozása • Folyadékszínt utántöltése • Vegyszerek adagolása • Haleledel adagolása • Zajszintmérés, hibajelzés • Szivárgásdetektálás • Kémiai tulajdonságok monitorozása.

**Tartalmi összefoglaló**

A szakdolgozat témája akvárium automatizálás és távfelügyelet megvalósítása.

Az akváriumok szépen összehangolt és viszonylag kényes rendszerként működnek, mely rendszer üzemeltetése és karbantartása általában csak manuálisan megoldható, sok esetben viszont (Pl.: időhiány, utazás) ezek nem végezhetők el.

Talán ezek közül is az egyik legfontosabb a halak etetése, hiszen enélkül elpusztulhatnak. Emellett azonban a világítás, a szűrő- és levegőztetőpumpák kapcsolgatása a vízszint mérése, a megfelelő vízhőmérséklet szinten tartása, a takarítás rendszerességének betartása (szűrőbetét takarítás és vízcsere, ha nincs talajszűrő akkor porszívózás) is elég leterhelő és időigényes feladatok, arról nem is beszélve, ha nem vagyunk otthon akkor ezeket nem tudjuk mi magunk elvégezni. Emellett pedig a fénymennyiség is fontos tényező, ugyanis ennek figyelésével akár a kiégett lámpatestről is információt kaphatunk. Bizonyos esetekben pedig figyelni kell a víz Ph értékét is, kifejezetten ha a halaink érzékenyek erre.

Ezekenek a különféle berendezéseknek folyamatosan jól kell működniük, a különböző jellemzőket célszerű mindig egy adott értéken tartani, valamint, ha az akvárium esetleg szivárog arról minél előbb tudomást kell szereznünk, hiszen ez nem csak a lakói vesztét okozhatja, de a berendezések is tönkre mehetnek, akár még tűz is kialakulhat. Ebben a dolgozatban ezekre a problémákra nyújtok megoldást.

A felsoroltak közül a legtöbb folyamat jól automatizálható egy mikrovezérlő segítségével, ezzel a koncepcióval az akvárium jellemzőit folyamatosan tudjuk szabályozni, monitorozni, az értékeket pedig egy adatbázisban tároljuk, hogy bármikor megtekinthessük, statisztikát készíthessünk. Emellett pedig egy telefonos applikáción keresztül bárhonnan elérhetőek. Ez az applikáció szolgál a rendszer konfigurálására, az adatok elérésére, valamint távoli beavatkozásra is. Természetesen a szűrők takarítását és a bonyolultabb mechanikai behatásokat igénylő folyamatokat nehéz egy kompakt megvalósításban automatizálni, ezek elvégzésének a szükségességéről a felhasználó értesítésen keresztül tájékozódhat.

*Akvárium; Automata; Távfelügyelet; Mobil applikáció; Mikrovezérlő*

**Tartalomjegyzék**

1. Egy automata akvárium 5
2. Hardveres megoldások 7
   1. Szenzorok és aktuátorok illesztése 7
   2. Hálózati feszültség kezelése 9

**1 Egy automata akvárium**

Egy automata akvárium megvalósítása pontos és precíz tervezést ígényel, de ez nem elég, a kivitelezésnek (legalábbis a vízhez közeli részeinek) érdemes vízállónak lennie, emellett pedig rengeteg más dologra is oda kell figyelni. Na de lássuk mik is ezek.

Maga a rendszer egy ESP8266 NodeMCU mikrovezérlő- és WiFi modul segítségével fogja ellátni a meghatározott feladatokat. Ez az egység 3.3V-on üzemel, így kisebb lesz a fogyasztása mintha egy 5V-on üzemelő mikrovezérlőt használnánk. A boardnak szüksége lesz egy valós idejű órára, erre viszonylag egyszerű megoldást nyújt, ha egy NTP (Network Time Protocol) szervert használunk a pontos idő eléréséhez. Ezzel biztosíthatók a megfelelően pontos időzítések, és az adatok rögzítésének pontos időpontját is tudni fogjuk. A rendszerünk idejét elegendő napi egyszer szinkronizálni a szerver idejével, ezzel csökkentve a fogyasztást, ugyanakkor kellő pontosságot jelent (a belső RTC hibája max. ~ 1 perc / nap).

Tárhelyhatékony megoldást nyújt, ha a monitorozott adatokat egyből a rögzítésük után beírjuk az adatbázisba, ugyanis így nem kell a modulunk memóriáját használni ezek tárolására, ugyanis ez akár napi 24 \* 4 (15 perc a leggyorsabb beállítható mintavételi időköz) vagyis majdnem 100 (96) írás műveletet venne igénybe. Ez jelentősen rövidíthetné a rendszer élettartamát, hiszen a memória maximális írásszáma korlátozott.

Az etetéshez egy szervómotor van a segítségünkre, ezzel tudjuk forgatni a henger alakú eledeltárolót (a tároló egyik oldala nyitott), a forgások számával az eledel mennyisége is szabályozható.

A berendezések kapcsolgatása, időzítése, mivel ezek hálózati feszültséget használnak, relék segítségével valósul meg. (A megvalósítás rendelkezik egy erre a célra megfelelő tápegységgel és biztosítékkal!) Ezen reléket egy-egy kimeneti konnektor állásra csatlakoztatjuk, melyek szám szerint vannak nyilvántartva, így a felhasználó választhat, mely berendezéseket akarja időzíteni.

A vízhőmérséklet és Ph monitorozásához egy vízálló digitális hőmérőt és egy analóg Ph szenzort használunk. A fénymennyiség érzékelésére egy fotorezisztor szolgál. A vízszint méréséhez és a szivárgásdetektáláshoz (adott időintervallumon belüli feltűnően nagy vízveszteség észlelése) egy vízszintmérő szenzort használunk. (Utóbbi 2 szenzor első használatkor kalibrálásra szorul.) A vízpótlás és vízcsere problémáira annyiban nyújtunk megoldást, hogy értesítjük a felhasználót az applikáción keresztül ezeknek szükségességéről. (A koncepció blokkvázlatát az 1. számú melléklet tartalmazza.)

Az értesítéseket egy telefonos applikáción keresztül kapja meg a felhasználó, mely mind IOS mind Android telefonokra elérhető. Az applikáció mögött egy MySQL adatbázis áll, PhP alapú API hozzáféréssel.

A felhasználónak a rendszer sikeres működtetéséhez fel kell telepítenie telefonjára a vezérléshez készített applikációt, melyen regisztrálnia kell és megadni a rendszer azonosítóját, ezt a megvalósítás kijelzőjén láthatja majd. A rendszer teljes mértékben ezen az applikáción keresztül konfigurálható itt lehet megadni a kívánt időzítéseket, az ideális értékeket és a takarítási értesítések gyakoriságát is, melyekről az értesítéseket szintén itt kapjuk majd meg.

Természetesen a felhasználó nem csak az applikáción keresztül kaphat információt, erre szolgál a berendezésen található LCD kijelző is, melyen az idő, Ph, és hőmérséklet lesz kijelezve // TODO?, valamint a kezdeti konfiguráció elvégzéséhez szükséges információk is itt találhatóak majd meg. Ezen felül lesz egy vészjelző LED, melynek szerepe, hogy kigyullad ha valamely érték elhagyja a kívánt tartományt.

A rendszer fejlesztésére és tesztelésére egy saját üzemeltetésű 54 literes (60cm \* 30cm \* 30cm) akvárium szolgál. Fontos megjegyezni, hogy a rendszer csak édesvizű akváriumok esetén alkalmazható, mivel a sós vizű akváriumok más típusú (Pl.: sóálló) szenzorokat igényelnének.

**2 Hardveres megoldások**

A teljes rendszer szíve az ESP8266 12-E NodeMCU mikrovezérlő modul, ebben a fejezetben az ehhez használt kiegészítők és perifériák hardveres illesztéséről és ehhez tartozó megoldásokról lesz szó. Kezdetnek említsük meg, hogy ez a modul csak egy analóg bemenettel rendelkezik, így egy 2 \* 4 csatornás analóg dekóder (CD4052 – multiplexer/demultiplexer) segítségével tudjuk az analóg szenzorok által kiadott feszültésgszinteket erre az egy bementre kapcsolni, így az összes ilyen típusó szenzor (Fény/Vízszint/Ph) adatát ezen a bemeneten tudjuk majd fogadni, valamint mivel az ESP-nek csak korlátozott számú digitális bemenete van (9), így az egyszerűség kedvéért a relé modul vezérlését is ezen keresztül oldjuk meg, mivel analóg dekóderről van szó, nem jelent gondot a digitális jelek átvitele sem.

Ugyanakkor az ESP 3.3V-os feszültségszinten működik, így a továbbiakban a Vcc = 3.3V és GND = föld jelöléseket használjuk. Ezen felül viszont azt ezt igénylő perifériák számára biztosítjuk az 5V-os feszültséget, mivel az ESP Vin bemenetére ez a feszültség kerül, a 3.3V-ot ebből a modul belső áramkörei állítják majd elő.

**2.1 Szenzorok és aktuátorok illesztése**

Az első és talán legfontosabb perifériánk a hőmérő, ehhez egy vízálló digitális hőmérő (DS18B20) lesz a segítségünkre, melynek 3 kivezetése van, VCC és GND valamint az adat átvitelére szolgáló DAT. Utóbbi vezetéket egy 4,7kΩ-os felhúzó ellenállással együtt kell bekötni, ezt a kivitelező egy a szenzorhoz mellékelt adapterrel meg is oldotta, így nekünk csak a vezetékeket kell a megfelelő portokra kötni, a DAT vezetéket egy tetszőleges (digitális) bemenetre, a VCC és GND pedig értelem szerűen a Vcc és a GND portokon kerül bekötésre.

A második szenzorunk legyen a vízszint mérésére szolgáló vízszintmérő szenzor, mely egy analóg szenzor 3 kivezetéssel. Ebben az esetben is van egy VCC és egy GND kivezetésünk ezen felül pedig egy S névre hallgató, ami az analóg feszültséget fogja adni mely a szenzor vízbe merítésével változik. Fontos megjegyezni, hogy a VCC bemenetet egy digitális portra érdemes kötni, hogy ezzel szabályozzuk a mintavételezést és ne legyen folyamatosan feszültség alatt a szenzor, valamint azt is, hogy a szenzor méretei miatt maximum ~4cm vízszintkülönbséget tudunk mérni, de ez tökéletesen megfelelő, hiszen egy akvárium esetén már 2cm hiányzó víz is baj lehet. Vegyük például a célakváriumot (60cm \* 30cm alapterület), ebből könnyedén kiszámítható, hogy 60 \* 30 \* 2 = 3600 cm3 = 3,6l. Belátható, hogy ez már egy ilyen kis akvárium esetében is jelentős vízveszteség. Ez a szivárgásdetektáláshoz is megfelelő mérési alapot biztosít. Meg kell említeni, hogy a modul csatlakozásainak vízállósítása szükséges, melyet ragasztóval érünk el // TODO???.

Folytassuk a fénymennyiség monitorozásával. Ehhez egy fotorezisztor lesz a segítségünkre, melyet úgyszint vízállóvá kell tennünk, ezt átlátszó ragasztóval tehetjük meg, a szenzor működsésnek befolyásolása nélkül. A fotorezisztor bekötése a következőképpen alakul, egyik lábát kössük Vcc-re, a másik lábát kössük a multiplexer választott bemenére. A mért jelet pedig egy 10kΩ-os referencia ellenálláson keresztül földeljük le.

Haladjunk is tovább a Ph szenzorunkra, ez még nincs meg de meg lesz valamikor remélhetőleg// TODO

Következzenek az áramkörben használt beavatkozó szervek, tehát a szervó motor, amely a haleledel tartó forgatásáért és ezzel az etetésért felelős mechanikai behatásokért felelős, valamint a relék melyek az akvárium hálózati feszültségen működő berendezéseit vezérlik.

A szervómotor bekötése viszonylag egyszerű, egy 3 vezetékes szervóról beszélünk, általában fekete-piros-narancs színű vezetékekkel. A motorunk is ilyen. A fekete vezeték a GND, a piros a Vcc a narancssárga pedig a motor vezérvonala, ezt egy tetszőleges digitális portra kötjük. Ezek a szervók PWM elven működnek, az ESP digitális portjai pedig képesek ellátni ezt a feladatot (tetszőleges feszültségértéket vagy akár frekvenciát jeleníthetünk meg rajtuk, természetesen a 3.3V-os kötöttség mellett), valamint szoftveresen sem ütközünk nehézségekbe, hiszen ezekhez a motorokhoz kész nyílt forráskódú függvénykönyvtárak állnak a rendelkezésünkre, melyek implementálnak minden a szervómotor vezérléséhez szükséges függvényt.

Térjünk is rá a relék koncepciójára és bekötésére. A megvalósításban egy 4 csatornás relé modult használunk, mivel 3 csatornás nem kapható (költséghatékonyabb megoldás mintha egy 2 és egy 1 csatornás relénk lenne), és a kesőbbi bővítés lehetősége mindig jobb mint ha nem lenne rá lehetőség. Ennek a modulnak 6 bemenete van, ebből 2 van fenntartva a VCC és GND vezetékeknek, a maradék 4 pedig a 4 csatorna vezérlését hivatott elvégezni, ezeket digitális jelekkel irányíthatjuk, melyeket a dekóderen keresztül kapnak meg a kiválasztott relék, így a 4 vezérvonal csak 1 portot foglal a mikrovezérlőnkön. Fontos még megemlíteni, hogy a relék által kezelt feszültséget fizikailag jól elkülönítjük a megvalósításban törpefeszültséget használó áramkörtől ezért a relé modul nem kerül integrálásra a nyomtatott áramkörön, hanem csavar-terminálokon keresztül vezetékkel köthető be.

**2.2 Felhasználói felület**

A felhasználó tájékoztatására és adatok megjelenítésére egy 4 \* 20 karakteres LCD kijelző lesz a segítségünkre. Ezt már szerencsénkre vásárlás előtt felszerelik egy I2C interfésszel, melynek köszönhetően egyszerű SPI periféria válik belőle, így mindössze 4 portja van, egy VCC egy GND melyeknek bekötése egyértelmű, fontos megemlíteni, hogy ezt a VCC portot 5V-ra kell kötni (3.3V-al is működik, de a kontrasztos megjelenítés érdekében érdemes magasabb feszültségen üzemeltetni), melynek előállítása a következő fejezetben van kifejtve. Ezeken felül pedig egy SDA (adat) és egy SCL (órajel), ezeket az ESP-n kifejezetten SPI-re kialakított portokra kötjük, melyek rendre a D1 és D2 digitális GPIO portok.

A kijelzőn felül a felhasználó kap egy // TODO? gombot mely szenzorok kalibrálására szolgál, a felhasználó felállítja a rendszert, a vízbe helyezi a szenzorokat, majd a gombot megnyomva megtörténik ezek kalibrálása (Pl.: a vízszintmérő szenzor 100%-ot jelentő vízszint beállítása, a Ph szenzor kalibrálásához viszont desztillált vízre van szükség).

**2.3 Hálózati feszültség és konvertálása (230V-ból 5V)**

Mivel a rendszer hálózati feszültséget is használ (berendezések működtetése), sőt maga a megvalósítás csak ezt a feszültséget kapja bemenetként, így ennek a helyes elkülönítése valamint transzformálása is külön feladat, ebben a fejezetben erről ejtünk szót.

Maga a megvalósítás egy 250V (10A AC) fogadóképességgel bíró, biztosítékkal és kapcsolóval ellátott tápegységet használ bemenetként, melyet egy hálózati dugaj és kábel segítségével csatlakoztathatunk bármely európai foglalattal rendelkező konnektorálláshoz. Ezt az áramot szétosztva, reléken keresztül juttatjuk el az akvárium berendezéseihez, melyek egy-egy erre szolgáló, a megvalósításon lévő konnektoron keresztül vehetik fel ezeket, így ezen konnektorok áramellátásának időzítése (és ezzel az akváriumi berendezéseké) már csak szoftveres megoldásokat igényel. Ezen aljzatok időzítését szám szerint látjuk el, így a felhasználó dönthet melyik aljzat melyik berendezésért lesz felelős. Ezekből az aljazatokból 3-at biztosítunk (Világítás/Szűrő/Levegőztető), melyek egymás mellet foglalnak helyet a megvalósítás hátlapján.

Ebből a hálózati feszültségből kell 5V-ot előállítanunk, hogy az ESP modul Vin portján keresztül meg tudjuk azt hajtani (a Vin port max 15V feszültséggel hajtható meg, de ezt érdemes minél alacsonyabban tartani, hiszen a felesleges felvett áram hőként fog távozni, így 5V teljesen megfelelő számunkra). Erre a célra egy 5V-os tápegység adaptert használunk (RS-25-5), mely célja, hogy ezt a feszültségtranszformációt elvégezze nekünk. Ez az adapter a 250V váltóáramot átalakítja 5V (max. 5A) egyenárammá. Ezt pedig használhatjuk az 5V-ot igénylő áramkörök tápfeszültségeként is.

**3 Szoftveres megoldások**

// GITHUB TOKEN:

**ghp\_tbGUSjXY1OcpXwEFzpmG2drFl0kMJl43RJhv**

4 ESP8266 (Szoftveres megoldások)

**5 Adatbázis és API**

Az adatbázis egy fontos eleme a projektnek, hiszen itt kerülnek eltárolásra az adatok és a beállítások is, valamint innen tudjuk a telefonos applikációban megjeleníteni a szenzorok adatit is. Az itt eltárolásra kerülő adatok a felhasználó adatai (email cím, név, jelszó), az akvárium (egy felhasználónak lehet több is) adatai (azonosító, név, méretek), a konfigurációs adatok az adott akváriumhoz valamint az adott akvárium szenzorai által mért adatok. Utóbbiakat a mikrovezérlő adott időközzel mintavételezi (ez külön konfigurálható, max 15 perc), és feltölti azokat az adatbázisba. Ezen felül az értesítések a mikrovezérlőről érkeznek a PhP alapú API felé, ahonnan tovább lesznek küldve a telefonos applikációba úgynevezett ’push notification’ formájában.

**5.1 Az adatbázis**

Az adatbázist belső kapcsolatait leíró egyed-kapcsolat diagramot a // TODO melléklet tartalmazza. Ennek a digramnak a relációsémákká való leképzése itt látható (az Outlet helyett az OL rövidítést használjuk):

User(email, firstName, lastName, deviceToken, authToken);

Aquarium(Id, name, length, depth, height, fishCount);

HaveAquarium(*email*, *Id*);

Config(*Id*, minTemp, maxTemp, minPh, maxPh, OL1On, OL1Off, OL2On, OL2Off, OL3On, OL3Off, waterLvlAlert, prefLight, feedingTime, foodPortions, filterClean, waterChange, sampleTime, lastModifiedDate);

SensorSample(*Id*, When, temperature, Ph, waterLvl, lightAmount);

Ezek alapján már egyértelműen felvehetjük az adatbázisunk tábláit, melyeknek SQL leírása megtalálható a // TODO számú mellékletben. Értelemszerűn összekapcsoljuk a táblákat, hogy esetleges felhasználó törlés esetén az ehez tartozó adatok is törlődjenek, valamint ha mondjuk a felhasználó kitörli egyik akváriumát akkor az ahoz tartozó konfigurációk és minták is törlődjenek.

Az akvárium azonosítója automatikusan generálódik majd, amikor a rendszer először csatlakozik az adatbázishoz, ez a WiFi hálózat megadása után meg is történik (a rendszer létrehoz egy akváriumot az adatbázisban alapértelmezett paraméterekkel, melyek később módosíthatók), ekkor a felhasználó a rendszer kijelzőjén láthatja majd a kiosztott azonsítót, ezt a telefonos applikációba való regisztrációkor meg kell adni, vagy ha már regisztrált (ez implikálja, hogy már van egy rendszere), ilyenkor az új akvárium felvételéhez kell megadni az azonosítót. Ennek megadása nélkül nem köthető össze a felhasználó és az akvárium így ez kötelező. Ezután a felhasználó már tetszőlegesen állíthatja az akvárium további adatait.

Minden konfigurációnak van egy azonosítója, amely az akvárium azonosítójával egyezik meg, külső kulcsot alkotva, így minden akváriumhoz csak egy konfiguráció tartozhat és fordítva, melyet tetszőlegesen lehet állítani az applikációban bejelentkezés és az adott akvárium kiválasztása után.

**5.2 Php API**

A screen shot of a computer

Description automatically generatedAz API (Access Point Interface) minden alkalmazás esetén létfontosságú, hiszen ezen keresztül történnek az adatok kezelése és a kérések kiszolgálása is, ez adja a kapcsolatot az alkalmazás és az adatbázis közt. Esetünkben ez az API Php alapú, és kezeli mind a mikrovezérlő, mind a telefonos applikáció által küldött és fogadott adatokat. Implementálásra kerülnek benne a használt adatszerkezetek modelljei, az ezek alapján az adatbázissal interaktáló DAO (Data Access Object) valamint a különféle kérések kiszolgálására létrehozott vezérlések (minden kérés egy-egy különböző fájlba fut be, mely kezeli azt). Az API mappaszerkezetének felépítése a 5.2.1-es ábrán látható. Természetesen a CONTROLS mappa tartalma jelentősen több, hiszen ide kerül minden funkció megvalósítását ellátó fájl is, de szemléletesség céljából itt ennyi is elegendő. 5.2.1 ábra

**5.3 Funkcionális követelmények**

* Az API-nak minden lehetséges kérést ki kell tudnia szolgálni, hibás kérés esetén pedig jelezni a hibát egy hibaüzenettel.
* A jelszavak titkosítva kerüljenek átadásra, valamint az adatbázisban is titkosítva legyen tárolva.
* Lehetőség szerint minimalizáljuk a lekérdezéseket, a már lekérdezett és használni kívánt adatokat tároljuk el a szerveroldalon.
* Lehetőség szerint minimalizáljuk a mikrovezérlő felől érkező kéréseket, beüzemelés után csak a minták feltöltése, értesítések és esetleges hibák okozzanak kéréseket.
* A frissített beállítások, melyek befolyásolhatják a mikrovezérlő működését lehetőleg azonnal kerüljenek frissítésre az eszközön is.

**Mellékletek**

A diagram of a computer program

Description automatically generated

1. Melléklet - A koncepció blokkvázlata