

IoT Akvárium

Martin Beránek

4. května 2016

Obsah

1	Úvod	1
2	Popis součástek	2
3	Schéma	3
4	Aplikace	4
5	Komunikace	4
6	Načasování	5
7	Čtení senzorů	5
7.1	Teplota	5
7.2	Spínací plovák	6
7.3	Relé	7
8	Komunikace s CACTI	7
9	Závěr	8

Seznam obrázků

1	Architektura	2
2	Diagram bubínku s krmením	3
3	Schéma zapojení součástek	4
4	Graf teploty v akváriu	7

Seznam tabulek

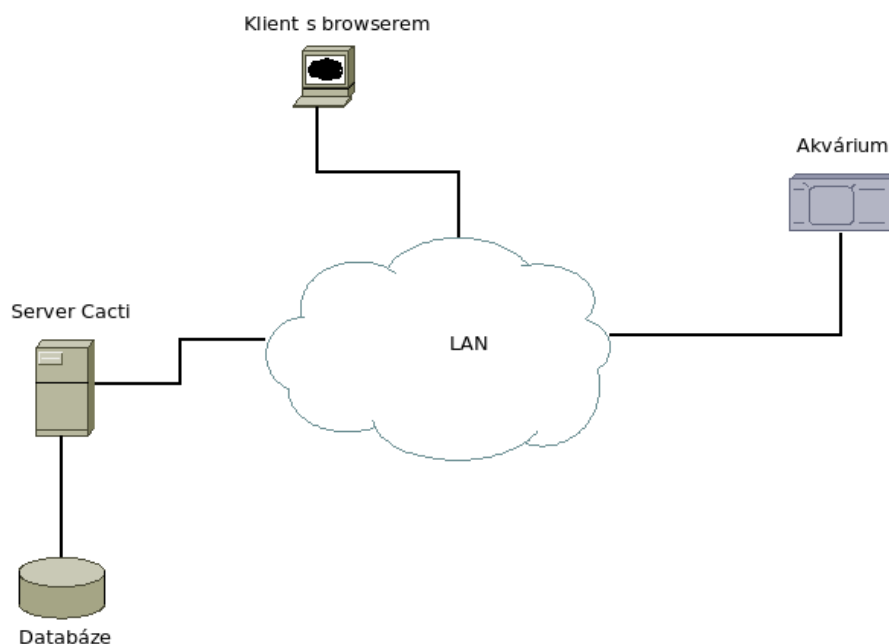
1 Úvod

Semestrální úloha má za cíl vytvořit ovladatelné akvárium, které bude schopné postarat se o rybičky. Mezi základní vlastnosti, které je nutné spravovat, je:

- Krmení,

- Rozsvědčení světla a zhasínání,
- Regulace vyhřívání,
- Kontrola výšky hladiny.

Údaje o akváriu je nutné uchovávat pro statistiky. Pro potřeby upozornění je nutné neustále kontrolovat stav hladiny. Na základě toho je stanovena jednoduchá architektura obsahující server, který sbírá informace a logický obvod akvária tvořený platformou **Arduino**.



Obrázek 1: Architektura

Pro vytváření grafů přímo na serveru slouží aplikace **Cacti**. Ta pomocí polleru spouští skript, který opakově otvírá TCP spojení na **Arduino** a zjišťuje současný stav. Na základě toho vytváří pomocí **RRDTool** graf.

2 Popis součástí

Na základě definovaných vlastností (1) byly vybrány následující součástky na platformu **Arduino**:

Krokový motor – pro otáčení bubínku s krmením pro rybičky.

Relé – pro spínání již vestavěného vytápění, osvětlení a filtru.

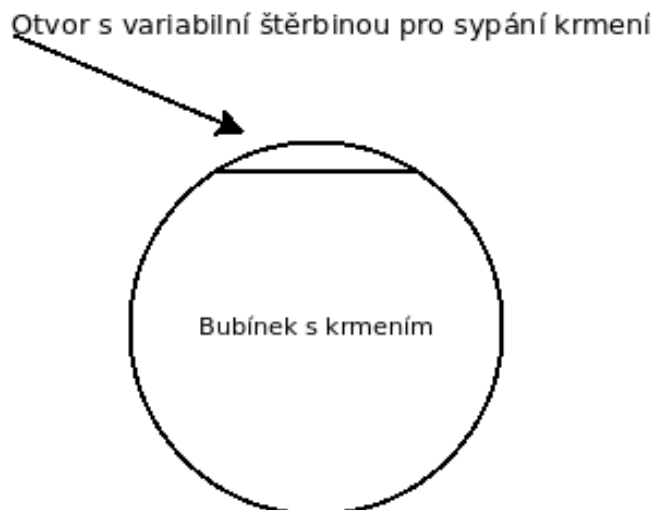
Teploměr – pro měření teploty vody v akváriu.

RTC – („Real-time clock“) pro reálný čas.

Bójka jako spínač hladiny – v případě, že v akváriu klesá hladina vody, sepne se bójka a **Arduino** uloží hodnotu o problému.

Ethernet shield – pro komunikaci se serverem pomocí TCP komunikace.

Vzhledem k tomu, že otáčení bubínku s krmením pro rybičky nevyžaduje nějakou větší sílu, je použit jednoduchý krokový motor 28BYJ-48. Ten je připevněn na bubínek a je řízen driverem ULN2003.



Obrázek 2: Diagram bubínku s krmením

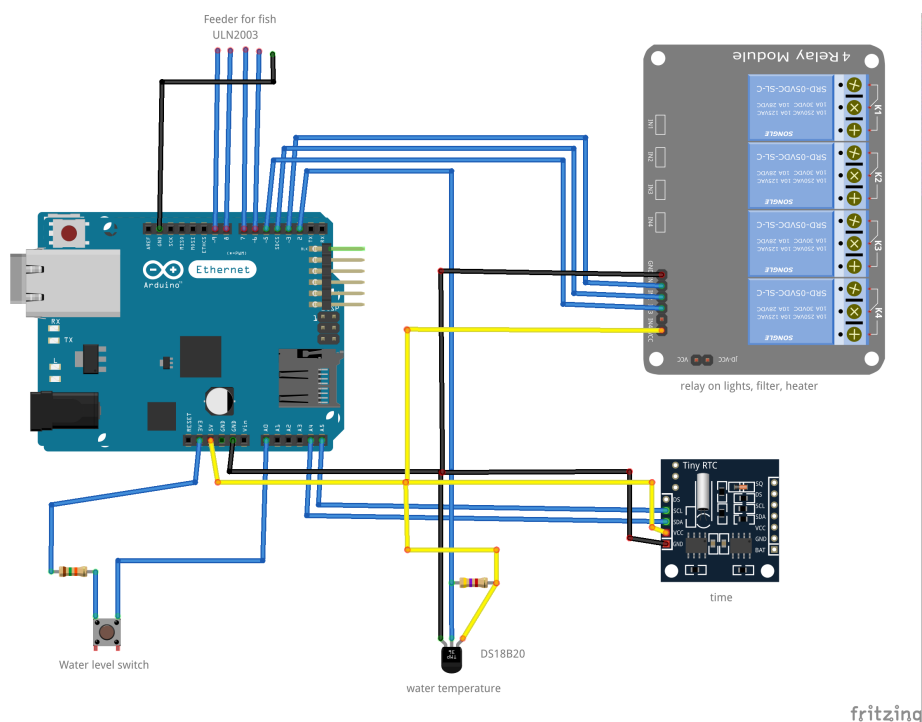
Pro spínání aktivních součástí akvária, které v něm už jsou zabudované, slouží relé. Relé jsou připojena na napájení ze sítě. Spínané součásti jsou: osvětlení, vyhřívání a filtr.

Teploměr musí být vodotěsný. V této práci je použita teplotní sonda s čidlem DS18B20. Ta je ponořena v akváriu na místě, kde je velký proud vody, aby rychle reagovala na změny teploty.

Arduino neobsahuje žádný modul reálného času. Kvůli nestabilnímu krystalu není ani možné měřit přesné časové intervaly. Na základě toho je připojen k **Arduino** obvod RTC s obvodem DS3231.

3 Schéma

Na základě výběru součástek je vytvořeno schéma celé zapojení obvodu:



Obrázek 3: Schéma zapojení součástek

4 Aplikace

5 Komunikace

Komunikace probíhá pomocí TCP/IP komunikace. Arduino naslouchá na pevně zadané IP adrese. Na základě příkazů dlouhých několik bytů aplikace na Arduino vytvoří zprávu a odešle jí v rámci komunikace. Příkazy, které Arduino zná jsou v následujícím seznamu:

- GZ - vrátí stav spínače hladiny
- GM - vrátí teplotu, která má být v akváriu
- GT - vrátí reálnou teplotu v akváriu
- GL - vrátí stav relé obsluhující světlo
- GF - vrátí stav relé obsluhující filtr
- GH - vrátí stav relé obsluhující vytápění
- GN - vrátí reálný čas Arduino
- GP - otočí bubínkem s krmením
- GW - vrátí čas, kdy má Arduino zapnout světla v akváriu
- GE - vrátí čas, kdy má Arduino vypnout světla v akváriu
- GK - vrátí čas, kdy má Arduino nakrmit rybičky

- SE hh mm - nastaví čas, kdy má Arduino vypnout světla v akváriu
- SW hh mm - nastaví čas, kdy má Arduino zapnout světla v akváriu
- SF hh mm - nastaví čas, kdy má Arduino nakrmit rybičky

ST nn - nastaví teplotu v akváriu, které má Arduino udržovat
SN -- nastaví čas pomocí NTP serveru

V případě, že Arduino nemá nastavený správný čas, vrátí příkaz GN samé nuly. Pomocí příkazu SN Arduino zavolá NTP server a nastaví si čas podle času na NTP. Výstup nastaveného času může vypadat takto:

```
Trying 10.0.0.173...
Connected to 10.0.0.173.
Escape character is '^'.
SN
8:36:50 4/5/16 Day of week: Wednesday
Connection closed by foreign host.
```

6 Načasování

Všechny časovače pracují se strukturou obsahující binární proměnnou, které hlídá, aby v intervalu nedošlo k vypnutí nebo zapnutí vícekrát.

```
struct Tmr{
    byte h;
    byte m;
    byte done;
};
```

Časovač pracuje pouze s hodinou a minutou. V případě přepnutí je nastavena i proměnná `done`. K jejímu navrácení do původní podoby musí uběhnout časový interval zvolené minuty.

```
void feedFish(){
    if (minute == feedt.m && hour == feedt.h && feedt.done == true){
        turnAround();
        feedt.done = false;
    }else if (minute != feedt.m && feedt.done == false){
        feedt.done = true;
    }
}
```

7 Čtení senzorů

7.1 Teplota

Teploměr pro měření je DS18S20. Komunikace s Arduino probíhá pomocí protokolu `OneWire`. Načítání probíhá pomocí zavolání senzoru na adrese [1]. Potom jsou načteny byty, které vystaví senzor. Ty jsou upraveny do podoby jednoho float.

```
float getTemp(){
    byte data[12]; // data
    byte addr[8]; // adresa
```

```

if ( !ds.search(addr)) {
    ds.reset_search(); // hledání senzoru
    return -1000;
}

if ( OneWire::crc8( addr, 7) != addr[7]) {
    Serial.println("CRC is not valid!"); // kontrola CRC
    return -1000;
}

if ( addr[0] != 0x10 && addr[0] != 0x28) {
    Serial.print("Device is not recognized");
    return -1000;
}

ds.reset();
ds.select(addr);
ds.write(0x44,1); // Začátek komunikace

byte present = ds.reset();
ds.select(addr);
ds.write(0xBE);

for (int i = 0; i < 9; i++) { // Načítání dat
    data[i] = ds.read();
}

ds.reset_search();

byte MSB = data[1];
byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB); // převod načtené teploty
float TemperatureSum = tempRead / 16;

return TemperatureSum;
}

```

7.2 Spínací plovák

Jelikož už na Arduinu nezbyl žádný digitální GPIO. Byl pro spínač použit analogový vstup. Měření musí proběhnout tedy několikrát, protože i při sepnutém plováku může AD převodník vrátit nulu. Server tedy při měření zasílá hned několik dotazů.

Arduino tedy jen čte stav analogového vstupu:

```

void initSurface(){
pinMode(SWITCH_SUR, INPUT);

```

```

}

int getSurface(){
    int buttonState = analogRead(SWITCH_SUR);
    if (buttonState > 0)
        return true;
    else
        return false;
}

```

7.3 Relé

8 Komunikace s CACTI

CACTI je služba, která je primárně určená ke sběru dat o zařízeních na síti. Na základě načtených dat potom vytváří pomocí RRDTool grafy. V tomto případě se vytváří graf teploty v akváriu. V případě, že vytápění a teploměr pracují v pořádku, graf by měl být konstantní. Odběr dat z Arduino probíhá pomocí skriptu v jazyce Python:

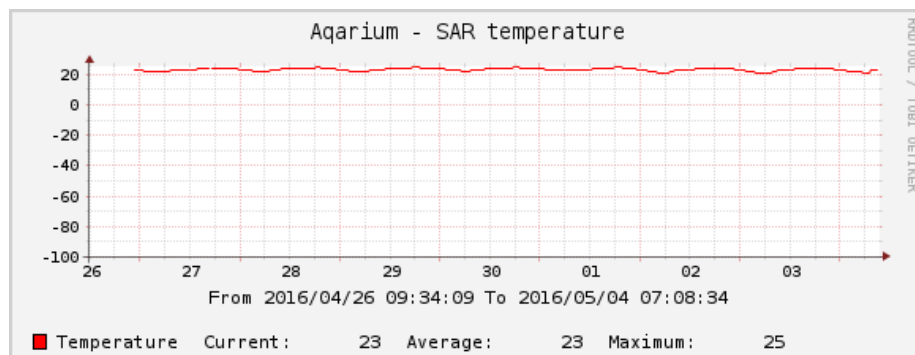
```

#!/usr/bin/python
from __future__ import print_function
import socket

HOST = '10.0.0.173'      # The remote host
PORT = 9011              # The same port as used by the server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.sendall('GT\r\n')
data = s.recv(10)
s.close()
print('Temperature:' + str(data),end='')

```

Výsledný graf jasně ukazuje, že teplota kolísá mezi dvěma stupni. Kolísání způsobuje otevřené okno na noc, které vodu ochladí a vytápění vždy teplotu postupně dorovná. K datu psaní této práce vypadá graf takto:



Obrázek 4: Graf teploty v akváriu

9 Závěr

Reference

- [1] Electrodragon.com: DS18B20. [cit. 2016-04-15]. Dostupné z: <http://www.electrodragon.com/w/DS18B20>