

**Jihočeská univerzita v Českých Budějovicích  
Přírodovědecká fakulta**

**Navrhnout a realizovat automatizovaný systém pro chov  
chrostíků s možností vzdáleného monitorování**

Bakalářská práce

**Miloslav Dušek**

Školitel: Ing. Šerý Michal, Ph.D.

České Budějovice 2022

**Bibliografické údaje:**

Dušek M., 2022: Navrhnout a realizovat automatizovaný systém pro chov chrostíků s možností vzdáleného monitorování. [Design and implement an automated system for caddisfly larvae breeding with the possibility of remote observation. Bc. Thesis, in Czech] – 55 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

**Annotation:**

This bachelor thesis deals with the design of automation of caddisfly breeding and measurement of selected parameters to achieve their successful breeding. Included work is the design of control electronics and peristaltic pumps needed for controlling water flows in the device. There are also designed Software and hardware needed to implement a web server that presents the measured data sent from the device and allows remote control of the application from the web server.

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích,

dne 12.4.2022

Podpis studenta.....

## **Poděkování**

Zde bych rád poděkoval panu Ing. Michalu Šerýmu, Ph.D. za jeho odborné rady, při zpracovávání této Bakalářské práce a za jeho ochotný a trpělivý přístup.

# Obsah

|       |                                           |    |
|-------|-------------------------------------------|----|
| 1     | Úvod.....                                 | 7  |
| 2     | Chrostíci.....                            | 8  |
| 2.1   | Chov larev chrostíků.....                 | 9  |
| 2.1.1 | Larvy chrostíků v stojatých vodách .....  | 9  |
| 2.1.2 | Larvy chrostíků v proudících vodách ..... | 10 |
| 2.2   | Sledované parametry .....                 | 10 |
| 2.2.1 | Teplota.....                              | 10 |
| 2.2.2 | Rychlost průtoku .....                    | 10 |
| 2.2.3 | Výška hladiny.....                        | 10 |
| 2.2.4 | Intenzita osvětlení .....                 | 11 |
| 3     | Arduino .....                             | 12 |
| 3.1   | Arduino – vývojové prostředí.....         | 12 |
| 3.2   | Arduino UNO .....                         | 13 |
| 3.2.1 | Arduino UNO pinout.....                   | 14 |
| 3.3   | Výhody .....                              | 14 |
| 4     | 3D tisk.....                              | 15 |
| 4.1   | Materiály pro 3D tisk.....                | 15 |
| 4.2   | Modely pro 3D tisk.....                   | 16 |
| 5     | CAD systémy .....                         | 17 |
| 5.1   | FreeCAD.....                              | 17 |
| 5.2   | OpenSCAD .....                            | 18 |
| 5.3   | SolidWorks .....                          | 18 |
| 6     | Peristaltická pumpa .....                 | 20 |
| 6.1   | Chemická odolnost .....                   | 20 |
| 6.2   | Okluze.....                               | 20 |
| 6.3   | Průtok.....                               | 21 |

|       |                                       |    |
|-------|---------------------------------------|----|
| 6.4   | Použití .....                         | 21 |
| 7     | Internet věcí (IOT) .....             | 22 |
| 7.1   | Rizika využívání IOT .....            | 22 |
| 8     | Krokový motor.....                    | 23 |
| 8.1   | Fungování krokového motoru .....      | 23 |
| 8.2   | Režimy krokového motoru .....         | 23 |
| 8.2.1 | Režim plného kroku .....              | 23 |
| 8.2.2 | Režim půlkroku .....                  | 24 |
| 8.2.3 | Režim mikrokroku .....                | 24 |
| 9     | Komunikační protokoly/sběrnice .....  | 25 |
| 9.1   | I2C komunikace.....                   | 25 |
| 9.2   | UART .....                            | 25 |
| 9.3   | USB komunikace .....                  | 26 |
| 10    | Hardware .....                        | 27 |
| 10.1  | Převodník logické úrovně .....        | 27 |
| 10.2  | Senzor DS18B20 .....                  | 27 |
| 10.3  | DHT 22 .....                          | 28 |
| 10.4  | Fotorezistor .....                    | 29 |
| 10.5  | HC-SRO4.....                          | 29 |
| 10.6  | Magnetický spínač .....               | 30 |
| 10.7  | ESP8266 .....                         | 31 |
| 10.8  | Pir senzor AM312.....                 | 32 |
| 10.9  | Krokové motory a jejich drivery ..... | 32 |
| 11    | Výroba součástek .....                | 35 |
| 11.1  | Peristaltická pumpa.....              | 35 |
| 11.2  | Chovné komůrky .....                  | 40 |
| 12    | Návrh chovného zařízení.....          | 42 |

|      |                                       |    |
|------|---------------------------------------|----|
| 12.1 | Samotný návrh.....                    | 42 |
| 13   | Zařízení dálkového monitorování ..... | 43 |
| 13.1 | Raspberry Pi 4 .....                  | 43 |
| 13.2 | ESP8266 .....                         | 44 |
| 13.3 | Přenos dat do ESP8266.....            | 46 |
| 14   | Výsledky.....                         | 47 |
| 15   | ZÁVĚR.....                            | 48 |
|      | Literatura .....                      | 49 |
|      | Seznam obrázků .....                  | 54 |
|      | Seznam příloh.....                    | 56 |

# 1 Úvod

Chov larev chrostíků, je stále do dnešní doby problematické zařídit. Jelikož biologové z Jihočeské univerzity chtěli chovat larvy chrostíků z důvodu genetického výzkumu a hledání genů, zodpovědných za vlastnosti vláken, do kterých se larvy chrostíků kuklí. Rozhodl jsem se tedy v této práci tedy sestavit zařízení, v kterém se budou moct chovat larvy chrostíků a další druhy larev. Navrhnout a realizovat jejich chovné komůrky, v kterých budou uloženy. Měřit parametry, které jsou potřebné pro chov larev chrostíků a dále je publikovat na web. Jelikož celý tento systém musí mít průtočnou vodu, kterou zařídím s pomocí peristaltické pumpy, jejíž části navrhnu a vytisknu pomocí 3D tisku a budu ji pohánět za pomoci krokového motoru řízeným přes Arduino UNO.

Toto téma jsem si zvolil za účelem lepšího seznámení zejména se systémy využívajícími Arduino UNO a jeho možnostmi pro automatizování a poté za lepším seznámením s různými CAD programy a jejich možnostmi, jelikož jsem se doteď seznámil pouze se SolidWorks.

## 2 Chrostíci

Chrostíci náleží do nadřádu Panorpoiidea, jsou blízcí příbuzní motýlů. Chrostíci mají v dospělosti zakrnělé ústní ústrojí, proto se v dospělosti živí zejména sladkými šťávami, dospělí chrostíci se krmí například sladkou vodou, vodou ředěnou medem či šťávou z ovoce. [1] [2]

V našich krajinách žije okolo 240 druhů chrostíků, ty se rozdělují do několik skupin. Nejvíce druhů chrostíka lze najít v proudících potocích s čistou vodou, některé druhy však nejsou tak závislé na čisté vodě a proudící vodě, takže je lze najít i v rybnících či řekách. [1] [2]



*Obrázek 1) dospělý chrostík [převzato upraveno 3]*

Larvy chrostíků se nacházejí na písčitých dnech těchto potoků či rybníků, kde spoluvytvářejí místní faunu. [1] [2]

Některé tyto larvy mají velice neortodoxní způsob obrany. Tyto larvy jsou samy od sebe vcelku velice málo chráněné vlastním krunýřem, ani nemají na obranu šupiny. Zato si tyto larvy vyvinuly obranu lepením si předmětů na své tělo, čímž si vytvářejí dokonalou obranu, tedy lepením na sebe malých kamínek či jiných předmětů které se dají nalézt na dně potoků. Existují však i druhy, které tento způsob nevyužívají a používají více ortodoxní způsoby, například jsou druhy, které si na dnech potoků staví svá hnízda, kde

jsou chráněné a zároveň si díky trychtýřovitému tvaru těchto hnízd do nich chytají potravu. [1] [2]

Určité druhy larev chrostíků jsou výhradně masožravci, kteří se živí lovem drobných říčních živočichů. Býložravé druhy pojidají místní řasy či si staví sítě, do kterých chytají říční plankton a živí se ním. Nejrozšířenější jsou však larvy, které jsou všežravé a pro potravu kombinují výše zmíněné možnosti. [1] [2]



*Obrázek 2) larva chrostíka [převzato upraveno 4]*

## **2.1 Chov larev chrostíků**

Chov larev chrostíků zejména záleží na druhu chrostíků, s kterým se potýkáme. Tyto druhy se dělí na 2 skupiny.

### **2.1.1 Larvy chrostíků v stojatých vodách**

Chov larev chrostíků v stojatých vodách je výrazně jednodušší než chov druhů, které potřebují proudící vodu. U těchto druhů je konstrukce chovných zařízení vcelku

jednoduchá, je možné je chovat v pouhých normálních akváriích s písčitým dnem. Stále je však nutné sledovat čistost vody a také její teplotu. [1] [2]

### **2.1.2 Larvy chrostíků v proudících vodách**

Chov larev, které potřebují proudící vodu je značně složitější. Pro tyto larvy je potřeba navrhnout a sestavit systém komůrek, které budou ideálně každá pro pouze jednu larvu. Tyto komůrky musí mít děravé stěny, aby skrz ně mohla proudit voda, ideálně by velikost těchto děr neměla překračovat 1 mm, aby zabránily úniku larev. Nároky na čistost a teplotu vody mají stejnou jako larvy, které žijí v stojatých vodách. [1] [2]

## **2.2 Sledované parametry**

Pro chov larev chrostíků je potřeba sledovat a udržet několik parametrů, aby bylo zajištěna jejich úspěšné odchování.

### **2.2.1 Teplota**

Teplota udává kinetickou energii kmitajících částí látek a těles. V tomto případě je teplota velice důležitá, jelikož larvy chrostíků potřebují udržovat takovou teplotu, jaká byla v potoku, z kterého byly odebrané. Jelikož chrostíci se vyskytují od vysokohorských potoků až po potoky v nížinách, rozmezí této teploty může být až 5-12 °C. [5] [6]

### **2.2.2 Rychlost průtoku**

Existuje mnoho druhů chrostíků a jejich larev, jsou druhy, které jsou adaptované tak, aby ve své larvální fázi žily a byly schopné lovit ve stojaté vodě, ale jsou i druhy které jsou závislé na proudící vodě. Proto je v tomto zařízení potřeba sledovat, pokud voda proudí a jakou rychlostí.

### **2.2.3 Výška hladiny**

Pro chrostíky, nebo případně jiné živočichy chované pomocí tohoto zařízení může být výška hladiny důležitým faktorem, je tedy potřeba ji sledovat. Také je potřeba znát výšku hladiny, aby bylo možné spočítat rychlost průtoku vody u samotných klícek, kde bude probíhat chov. Sloupec vody nad larvami chrostíků by měl být v rozmezí mezi 10 až 30 mm.

#### **2.2.4 Intenzita osvětlení**

Jelikož larvy chrostíků nejsou v přírodě normálně vystavovány intenzivnímu osvětlení, je potřeba sledovat, jestli larvy nadměrná intenzita osvětlení neovlivní. Dále je potřeba osvětlení sledovat, jelikož v přírodě je denní a noční cyklus, ale v uzavřené místnosti může být umělé osvětlení nezávislé na dni či noci.

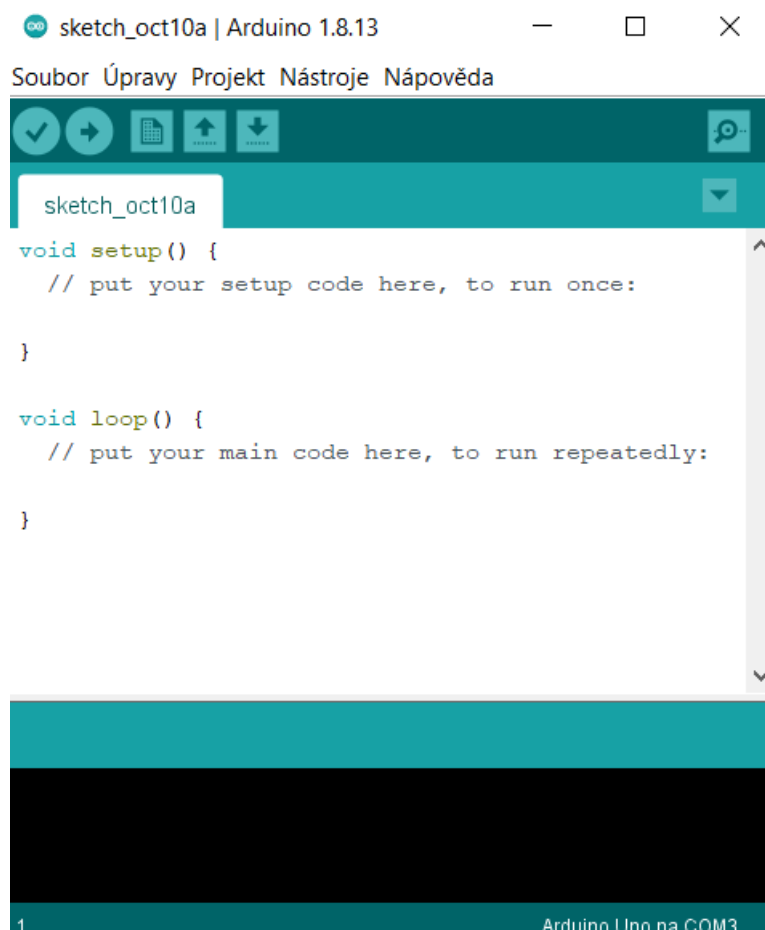
## 3 Arduino

Arduino bylo vytvořeno v Ivrea Interaction Design institute v roce 2005. bylo zamýšlené jako jednoduchý přístroj pro širokou veřejnost, která by chtěla s nástroji tohoto typu pracovat. Hlavním využitím mělo být pomoci studentům s projekty a s tvorbou prototypů. Hlavní využití je dnes ve školství, kde se Arduino využívá pro výuku na středních a vysokých školách. [7] [8]

Arduino je jednodeskový počítač, a i když se svým výkonem nemůže rovnat s jinými mikropočítači, například Raspberry Pi, dokázalo si Arduino díky své ceně a jednoduchosti na zapojení a ovládání vytvořit značnou, stále rostoucí komunitu, díky které se uplatnění Arduina navíc zvyšovalo. Kvůli této popularitě začaly mikroprocesory typu Arduino vyrábět i další výrobci, čímž opět zvýšily jeho možnosti. [7] [8]

### 3.1 Arduino – vývojové prostředí

Arduino se programuje přes Arduino IDE, což je open-source software, který vyvinulo Arduino.cc a využíváný převážně pro Arduino. Je designovaný tak, aby programování na něm bylo jednoduché i pro úplného začátečníka, který nemá s programováním žádné zkušenosti. V Arduino IDE se programuje v jazycích C a C++. Arduino IDE je použitelné na platformách Windows, Linux či MAC. Každé Arduino má v sobě ATmega mikročip, který přijímá informace a je částí, která je programovaná. [9] [10]



*Obrázek 3) Arduino vývojové prostředí [autor]*

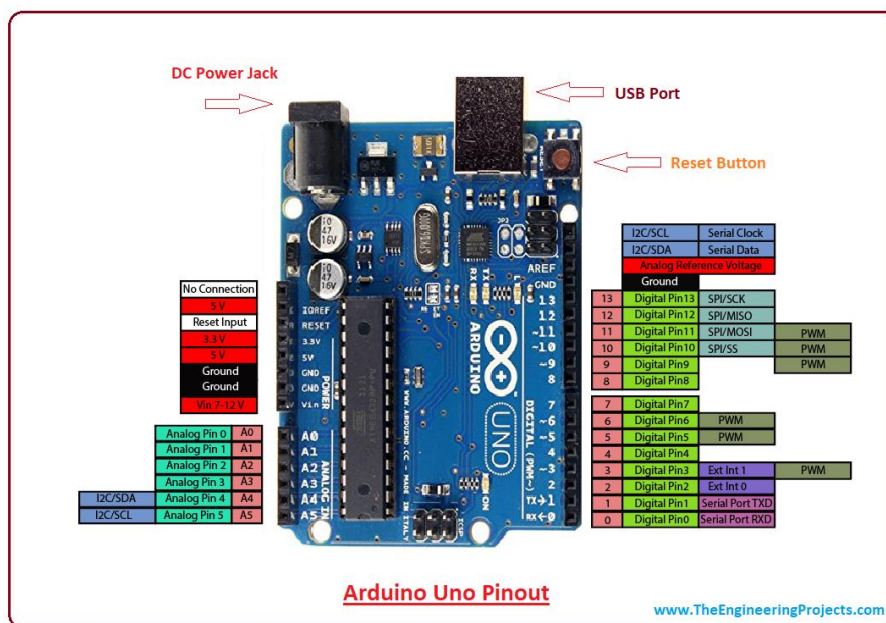
Arduino IDE funguje jednoduše, je rozdělené na 2 části, void setup() a void loop(). Při zapnutí se jednou aktivuje void setup() a tím se tedy připraví část, kterou chceme mít vždy na začátku programu nastavenou. Zato část void loop() se zapíná po celou dobu, kdy je Arduino aktivní a stále se opakuje. [9] [10]

## 3.2 Arduino UNO

Deska Arduino UNO obsahuje procesor ATmega328P, který má paměť 32KB a je jeho hlavní částí. Deska také obsahuje 2KB SRAM a 1KB EEPROM. Dále Arduino UNO obsahuje 14 digitálních I/O pinů, z těchto pinů výchozí maximální signál 5 V a dají se maximálně zatížit proudem 40 mA. 6 analogových pinů a také má v sobě zabudovaný USB-UART převodník, jelikož ATmega328P komunikuje prostřednictvím UART a Arduino UNO posílá a přijímá informace za pomoci USB komunikace. Arduino UNO také obsahuje stabilizátor napětí. Pro napájení Arduino UNO potřebuje buďto stabilizované napětí 5 V,

nebo externí napájení 7-12 V, které je redukováno vestavěným stabilizátorem napětí. Minimální napájecí proud je 300 mA. [10]

### 3.2.1 Arduino UNO pinout



Obrázek 4) Arduino UNO Pinout [převzato upraveno 11]

### 3.3 Výhody

Arduino má jako jeho největší výhodu jeho jednoduchost použití, které má díky jeho již zabudovaným funkcím, jako například oscilometr, mikrokontroler, 5 V regulátor či LED. Další jeho výhoda je jednoduchost načítání dat, kde k načítání nového kódu stačí pouze USB připojení, u samotného programování je také velice nápomocný počet knihoven, kterými Arduino disponuje. Rozsáhlá komunita, kterou má Arduino je také velkou výhodou, jelikož díky ní má velice rozsáhlá diskusní fóra.[12]

## 4 3D tisk

První primitivní použití 3D tisku se datuje do roku 1945, avšak tato technologie se začala skutečně vyvíjet až od roku 1971, díky nimž se roku 1981 podařilo Hull's company vytvořit první 3D tiskárnu. Od roku 1999 zažívá 3D tisk rapidní pokrok a od roku 2010 se z něj stal celosvětově využívaný princip pro mnohá odvětví, a i pro hobby využití, jejich počet se každým rokem rychle zvyšuje. [13]

3D tisk je forma aditivní výroby, koncept je stejný jako u normální 2D tiskárny, pouze rozšířený o další dimenzi. Princip je různý od tradičních způsobů výroby, například kde se z kusu materiálu odebírají části a tím se tvaruje. U 3D tisku se materiál nanáší na plochu a tím ji tvaruje. Existuje mnoho druhů 3D tisku, ale všechny mají společný koncept, že se na plochu nanese materiál, ať už předem roztavený či roztavený až na ploše pomocí laseru a tím tiskne vyžadovaný objekt. [14]

### 4.1 Materiály pro 3D tisk

Nejčastějším dnes používaným materiálem na 3D tisk jsou plasty. Jejich přednostmi jsou zejména jejich relativně malá cena, malá hmotnost, předměty z nich jsou velice hladké a bez mnoha poruch a může se při tisku použít libovolná barva plastu. [15]

Nejpoužívanější plasty používané pro 3D tisk:

Polyactic acid (PLA), tento materiál vyniká zejména tím, že je velice ekologický a velice pevný. Dnes PLA není nejoblíbenějším materiálem, ale předpovídá se, že jeho popularita rychle poroste díky jeho výborným vlastnostem.[15]

Acrylonitrile butadiene styrene (ABS). ABS je materiál velice oblíbený zejména mezi domácími 3D tisky, kvůli jeho výborné pevnosti a bezpečnosti. [15]

Polyvinyl Alcohol Plastic (PVA). PVA patří mezi materiály s horšími vlastnostmi, není tedy ideální pro namáhané díly. Ale tento materiál je také levnější než jiné, kvalitnější materiály, využívá se tedy zejména v domácích 3D tiskárnách na tisk méně namáhaných předmětů. [15]

Polycarbonate(PC). PC se používá méně používané materiály, jedním z důvodů je, že k tisku PC je potřeba, aby 3D tiskárna měla trysku a aby operovala za vysokých teplot. Mezi jeho největší výhodu patří zejména jeho nízké náklady. [15]

Dalším populárním druhem materiálů je prášek. V 3D tiskárně je tento prášek roztaven a poté po vrstvách nanášen na plochu, dokud těchto vrstev není dostatek na požadované rozměry předmětu. Tyto prášky mohou být z mnoha různých materiálů, mezi těmi nejpoužívanějšími jsou například: [15]

Polyamide (Nylon). Nylon vyniká především svou pevností a flexibilitou. Další jeho předností je možnost modelovat s velkým detailem. [15]

Alumide, tento materiál se skládá z kombinace Polyamidu a šedého hliníku, díky této kombinaci jsou modely z alumidu jedny z nejpevnějších 3D modelů. Alumide se velice využívá i v průmyslu díky jeho výborným vlastnostem. [15]

## **4.2 Modely pro 3D tisk**

U tvarování modelů pro 3D tisk je potřeba vždy myslet na způsob tisku, není možné efektivně tisknout předměty které nemají pro jejich části svislou spojitou podporu. Také je důležité při navrhování modelu myslet na to, kde a kterým způsobem model namáhán a model tomu přizpůsobit. Jako CAD program pro 3D modely je vhodné vybrat si takový, který má jednoduchý převod do programu pro tisknutí.

## 5 CAD systémy

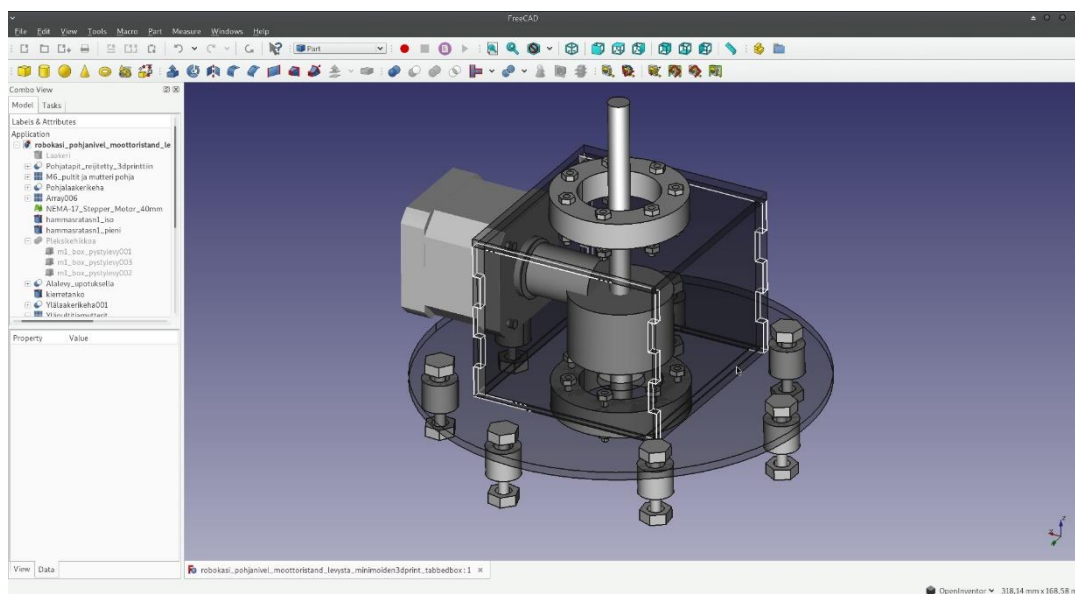
CAD, neboli computer-aided design, tyto programy jsou využívány pro designování 2D a 3D modelů a jejich dokumentaci. Díky těmto programům můžeme dnes lehce navrhnout díly určené pro 3D tisk a lehce je následovně vytisknout. [16] [17]

Programů CAD existuje mnoho, jejich nejjednodušší rozdělení je na placené programy a programy zadarmo. Placené programy jsou samozřejmě lepší, jelikož mají všeobecně více funkcí a mají méně chyb. Neplacené programy mají sice méně funkcí, ale pro jednodušší projekty jsou dostačující. A některé neplacené programy mají i více přehledné ovládání nežli placené programy. [16] [17]

Mezi nejlepší placené CAD programy jsou například AutoCAD, SolidWorks či Fusion 360. Nejlepší možné programy z neplacených programů jsou například FreeCAD, IronCAD či OpenSCAD. [16] [17]

### 5.1 FreeCAD

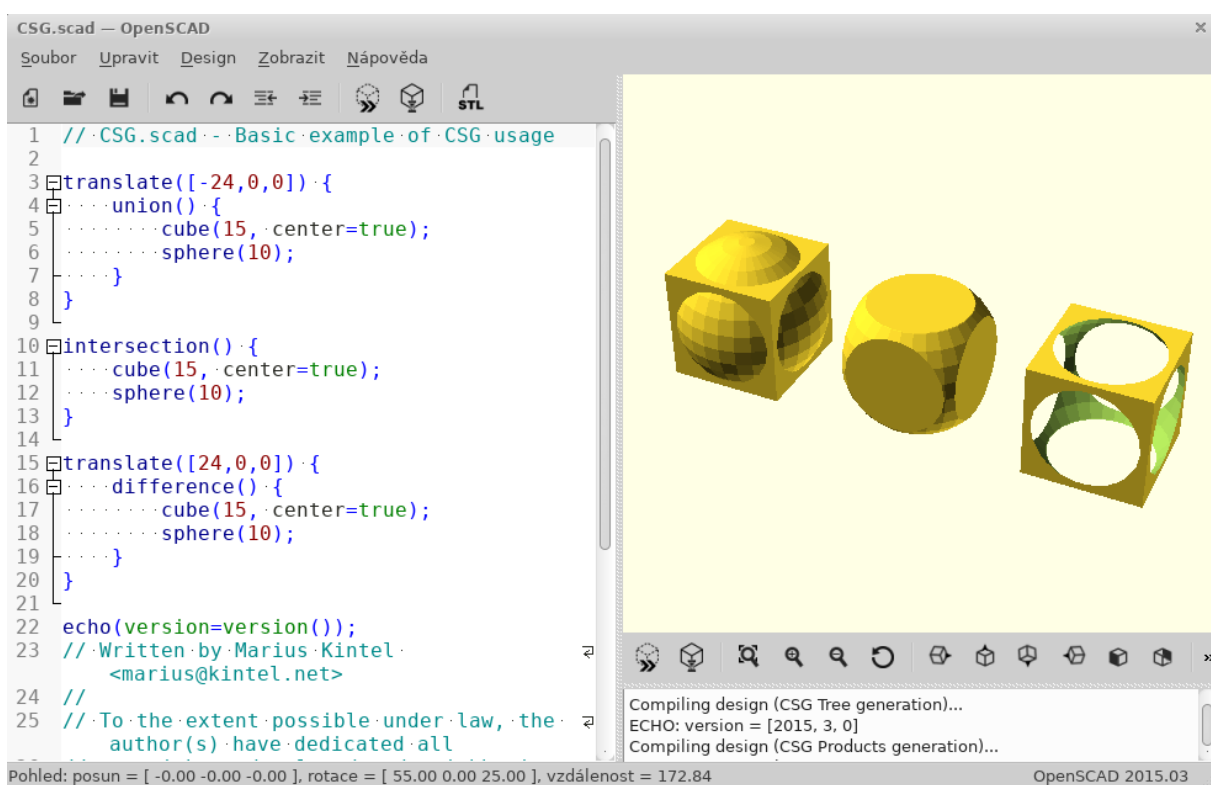
FreeCAD je 3D CAD software, který může operovat na Linux, Mac či na Windows. Na rozdíl od SolidWorks je FreeCAD zadarmo, s čímž bohužel přichází zvýšený počet chyb v softwaru. Oproti SolidWorks má v mnoha ohledech přehlednější ovládání, ale zároveň vůči SolidWorks postrádá řadu funkcí. [17]



Obrázek 5) FreeCAD [převzato upraveno 21]

## 5.2 OpenSCAD

OpenSCAD je 3D CAD software na vytváření 3D modelů, jednou z jeho výhod je že je dostupný zadarmo. Je dostupný pro Windows, Linux a další. OpenSCAD je unikátní vůči normálním 3D CAD softwarům jako například SolidWorks, jelikož v OpenSCAD se místo fyzického modelování modeluje za pomoci programování a systém pouze čte, co má vytvořit. Tento způsob je zejména zaměřen na uživatele, kteří umí programovat, oproti tomu normální 3d modelování jako v SolidWorks je od samého začátku velice intuitivní a lehce naučitelný. Hlavní výhodou modelování v OpenSCAD je, že jelikož uživatel píše celý proces modelace, má nad vši naprostou kontrolu, díky čemuž může předejít zbytečným chybám.[18]

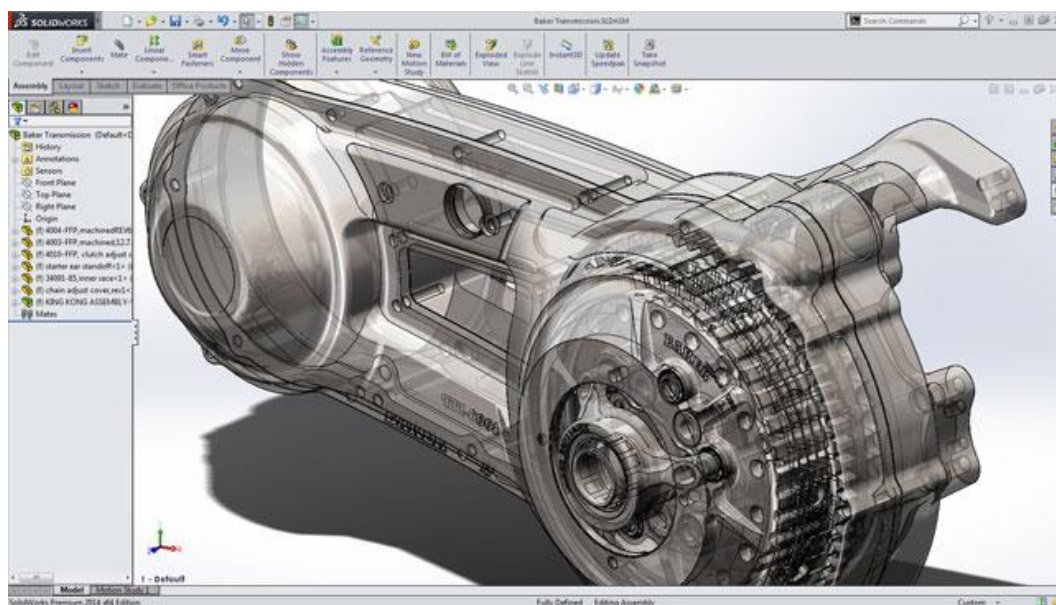


Obrázek 6) OpenSCAD [převzato upraveno 19]

## 5.3 SolidWorks

Solidworks je velice úspěšný 3D CAD software, nabízí velice přehledné ovládání a je jednoduché se v něm orientovat. SolidWorks může operovat pouze na Windows, čímž je velice omezen. Navíc jelikož je SolidWorks placeným programem, tak je o něj dobře staráno a je v něm jenom naprosté minimum chyb.[20]

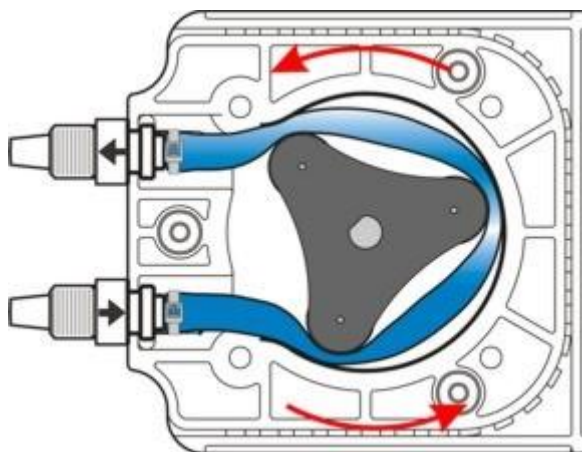
SolidWorks je velice oblíbený jako software pro studenty, na kterém se učí 3D CAD. Jak již bylo zmíněno, SolidWorks je placeným programem, avšak pro školy, které disponují licencí a jejich studenty je SolidWorks zadarmo.[20]



Obrázek 7) SolidWorks [převzato upraveno 22]

## 6 Peristaltická pumpa

Peristaltická pumpa funguje za pomoci podtlaku, kde vnější stěna tohoto zařízení je tvořena půlkruhem, po jehož obvodu vede trubička z elastického materiálu, která je stlačována válečkami z pumpy, které provádějí kruhový pohyb a vytvářejí tedy tímto podtlak, díky kterému se pumpuje voda. [23]



*Obrázek 8) princip peristaltické pumpy [převzato upraveno 23]*

### 6.1 Chemická odolnost

Tento parametr se vztahuje pouze na trubičku, kterou prochází tekutina. Tato trubička musí být dostatečně elastická a schopna vydržet opakovaná stlačení a zachovat si při tom své původní parametry. Pro tyto účely se dnes využívá několik materiálů, například PVC či silikon. [7]

### 6.2 Okluze

Okluze je v tomto případě myšlena jako minimální mezera mezi válečkem a obalem peristaltické pumpy. Za příliš velké okluze dochází k rychlejší degradaci trubičky, a tedy je nutné častější výměna. Za příliš malé okluze se trubička nemusí úplně stlačit, čímž vznikne v trubičce mezera, níž může tekutina proudit zpět a peristaltická pumpa ztrácí efektivitu do bodu, kdy může přestat fungovat. Okluze se uvádí v % a je velice významným parametrem pro konstrukci. [7]

$$O_s = \frac{2t - c}{2t} * 100 [\%]$$

(1)

$O_s$  = okluze,  $t$  = tloušťka stěny,  $c$  = mezera mezi válečkem a stěnou [7]

## 6.3 Průtok

Průtok je jedním z nejzásadnějších parametrů pumpy. Vypočítá se vzorečkem:

$$Q = V * N * n [\text{mm}^3/\text{s}]$$

(2)

$Q$  = průtok,  $V$  = objem mezi rameny,  $N$  = počet ramen,  $n$  = rychlost otáčení [7]

Z tohoto vzorečku je jasně poznat, že hlavními parametry pro průtok je vnitřní průměr hadičky, kterou bude kapalina proudit a vzdálenost hadičky od středu rotace. Dalším zásadním parametrem pro průtok je rychlost otáčení. [7]

## 6.4 Použití

Peristaltická pumpa má mnoho výhod vůči normálním pumpám. Její největší výhodou je, že transportovaná látka nemá žádný kontakt s vnějším prostředím, pouze s trubicí, kterou prochází. Dále je její provoz velice tichý vůči ostatním pumpám a nevyvolává nežádoucí otřesy. Díky těmto výhodám je pumpa velmi žádoucí v lékařském či chemickém průmyslu, kde by znečištění vnějším prostředím mohlo být katastrofické. [7]

## 7 Internet věcí (IOT)

Internet věcí byl poprvé definován britským inženýrem Kevinem Ashtonem v roce 1999. Avšak myšlenka vzájemného propojení je mnohem starší, datující se až do 19. století s vývojem telegrafu. Jedním z prvních novodobých příkladů je automatický automat od Coca-Coly, který samostatně monitoroval teplotu nápojů, které v něm byly uloženy. IOT se dále rozvíjelo v začátcích 21. století, kdy se také udála první konference o IOT v roce 2008 ve Švýcarsku. V roce 2011 byl zaveden IPv6, který zajistil přístup k dostatku IP adres a s ním zažil IOT velký boom, kdy se na něj začaly zaměřovat velké společnosti. [24]

IOT znamená propojení strojů a zařízení, jejichž vzájemná komunikace může probíhat nezávisle na asistenci člověka. IOT s připojením k internetu získává mnohé možnosti využití. Nejdůležitější součástí IOT jsou senzory, existuje mnoho různě zaměřených senzorů, například senzory měřící vzdálenost, teplotu, vlhkost či pohyb. Všechny senzory se dají propojit přes internet a tím vytvořit s IOT jeden celek.[24]

### 7.1 Rizika využívání IOT

Největší nevýhodou IOT je bezpečnost. Jelikož je velké množství zařízení připojené a často i kontrolované přes internet, je riziko hackerského útoku velice velké. Zabezpečení IOT je dnes stále nedořešenou problematikou a její řešení se zatím nezdá být v dohlednu. Hackerové například využívají DDoS útoku pro odcizení dat, ale i špatné nastavení IOT zařízení může vést k ztrátě či poškození dat. IOT. [24]

## 8 Krokový motor

Na pohon peristaltické pumpy si můžeme vybrat mezi standartními DC motory, nebo krokovým motorem. Výhoda DC motoru je, že má lepší výkon, další výhodou krokového motoru je lepší možnost řízení, během jeho provozu také dobře víme, kolik tekutiny projde, není tedy nutné zařizovat druhý přístroj na měření průtoku. Pro tuto práci jsem si vybral krokový motor, jelikož zde není potřeba výkon, který by krokový motor nezvládl.

### 8.1 Fungování krokového motoru

Pohyb krokového motoru se provádí za pomoci 2 jeho hlavních částí, rotoru a statoru. Stator je vnější částí krokového motoru a je nepohyblivý. Rotor se nachází v prostředku motoru, kde je osazený na hřídeli a je pohyblivou částí motoru.[25] [26]

Stator je složený z cívek, které pracují jako elektromagnety a jsou uloženy kruhově kolem rotoru. Pokud se do cívek pustí proud, tak v závislosti na jeho směru vzniká kolem cívek magnetické pole.[25] [26]

Rotor má na sobě instalovaný magnetický zub, který je přitahován magnetickým polem vytvořeným cívkami, čímž v závislosti na parametrech motoru vzniká rotace o určitý úhel. Po přemístění zubu k jedné cívce se zapne další elektromagnet, čímž se rotor opět posune a opakováním této činnosti motor rotuje. Díky tomuto jednoduchého principu přepínání cívek se může motor libovolně přesouvat vpřed i vzad.[25] [26]

### 8.2 Režimy krokového motoru

#### 8.2.1 Režim plného kroku

Motor se v tomto režimu otáčí podle počtu kroků motorů, nejběžnější jsou krokové motory s 200 kroky, v tomto případě každý krok pohne motorem o  $1,8^\circ$ . [28]

Režim plného kroku se může realizovat pomocí dvou způsobů, v prvním případě se napájí pouze 1 cívka, druhým způsobem je napájet 2 cívky naráz. Pokud se napájí pouze jedna cívka, tak je potřeba pouze malý výkon driveru a potřebuje pouze poloviční energii, ale vůči napájení dvou cívek naproti sobě má výrazně menší rychlost a krouticí moment. [27] [28]

### **8.2.2 Režim půlkroku**

Kombinace napájení jedné a dvou cívek, tak aby motor vždy postoupil pouze o půl kroku, tedy u 200 krokového motoru je v tomto režimu potřeba 400 kroků a každý krok znamená pouze  $0,9^\circ$ . [27] [28]

### **8.2.3 Režim mikrokoku**

Režimu mikrokoku lze dosáhnout za pomoci regulace proudu protékajícím cívkami a za určitých poměrů dosáhnout zmenšení kroků krokového motoru. [28]

Tímto způsobem lze například docílit například  $1/4$  kroků, díky kterým u 200 krokového motoru bude potřeba 800 mikrokroků pro jeho úplnou otočku a každý z těchto kroků posune rotor o  $0,45^\circ$ . Díky tomuto procesu je celý průběh motoru mnohem plynulejší a kroky jsou méně znatelné, až téměř nepozorovatelné. Nejmenší možné kroky tímto způsobem jsou  $1/256$  z původního kroku. [27] [28]

## 9 Komunikační protokoly/sběrnice

Kvůli různým komunikačním protokolům jednotlivých součástek je s nimi nutné použít různé způsoby komunikace. V této části práce se pokusím rozebrat jednotlivé způsoby komunikace použité v této práci a popsat je.

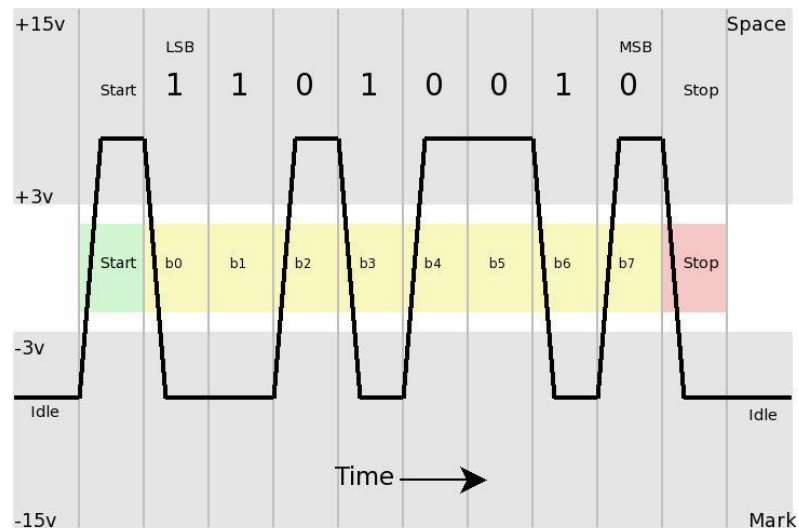
### 9.1 I2C komunikace

Sběrnice I<sup>2</sup>C využívá dvou vodičů na vytvoření datového propojení na principu Master/Slave, v tomto zapojení je vždy pouze jedno zařízení připojeno jako Master a ostatní jako Slave. [29]

Přenos dat začíná u Mastera, který zformuje podmínku startu, v této podmínce na bitech 7 až 1 vyšle adresu zařízení, pro které jsou informace určeny a dále pošle v bitu 0 směr přenosu. Poté master vyšle datový byte, kterým potvrdí komunikaci, po kterém se až do přerušení mohou posílat byty. [29]

### 9.2 UART

Asynchronní sériová komunikace na principu přenosu dat přes dva kabely, každý jedním směrem, zpráva začíná vždy start bitem a je zakončena end bitem. Na každém zařízení se jmenují tyto piny stejně, tedy RX (receiver) a TX (transmitter). Rychlost přenosu je udávána v baud rychlostech, kde 1 baud znamená přenos maximálně 1 bitu za sekundu. Typickými baud rychlostmi pro UART přenos jsou například 1200, 2400, 4800, 9600, 115200 baudu. V mém případě jsem využíval rychlosti 9600 a 115200 baudu za sekundu. Všeobecně nejpoužívanější rychlostí je 9600 baudu za sekundu. [7] [30]



Obrázek 9) UART přenos informací [převzato upraveno 31]

### 9.3 USB komunikace

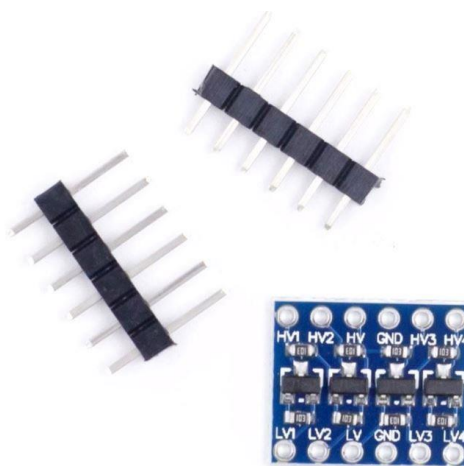
USB je zařízení typu Master, kde si PC buď žádá data nebo je posílá. Data se posílají v bajtech ve velikosti od 8 do 256 znaků. Přenos dat se uskutečňuje pouze v rámcích, které trvají přesně 1 milisekundu. Během 1 rámce je možné zpracovat informace i z několika různých zařízení. Bity jsou kódovány metodou Non Return to Zero Inverted. Jednotlivé USB kabely mohou být až 5 metrů dlouhé a s použitím rozbočovačů mohou dosáhnout délky až 30 metrů. Maximální rychlost přenosu dat přes USB komunikaci je 60 MB/s. [32] [33]

## 10 Hardware

### 10.1 Převodník logické úrovně

Jelikož ESP8266 má maximální napájení 3,6 V, nemůže být tedy přímo napájeno ze 5 V výstupu, který se nachází na Arduino UNO. Může být napájeno výstupem 3,3 V, ale ESP8266 má může mít větší spotřebu, než kolik tento zdroj dává proudu. Ideální zapojení je tedy 5 V výstup regulovaný pomocí převodníku logické úrovně na 3,3 V.

Pro tuto práci jsem použil obousměrný převodník logické úrovně, který má 4 piny pro vysoké napětí, 4 pro nízké, 2 pro referenční vysoké a nízké napětí a 2 pro GND. [34]



*Obrázek 10) převodník logické úrovně [převzato upraveno 35]*

### 10.2 Senzor DS18B20

Jako senzor teploty, který má za úkol měřit teplotu vody jsem zvolil DS18B20, který je vodotěsný, tedy ideální pro tento účel. Tento senzor je použitelný v teplotách v rozmezí - 55 °C do 125 °C, s přesností mezi teplotami - 10 °C do 85 °C na  $\pm 0,5$  °C. Potřebuje pouze napájení 3–5 V, takže může být napájen přes Arduino UNO. Komunikuje pomocí digitálních dat, takže i s dlouhými kabely neztrácí přesnost. [36]

V této práci bude senzor velice potřebný, jelikož bude kontrolovat teplotu vody v systému, tedy jeden z nejdůležitějších parametrů. Tato teplota bude měřena ve skladové nádobě na vodu. [36]



*Obrázek 11) senzor teploty DS18B20 [převzato upraveno 37]*

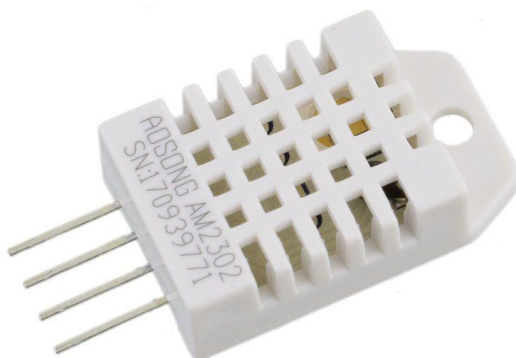
### **10.3 DHT 22**

Na měření teploty mimo vodu a také na měření vlhkosti vzduchu byl použit senzor DHT 22. Tento senzor je schopen zároveň měřit teplotu vzduchu i jeho vlhkost, tyto informace navíc přenáší jako digitální signál, takže není potřeba analogového zapojení. Další jeho výhodou je, že je dopředu kalibrován a Arduino IDE pro něj již má knihovnu, takže čtení informací ze senzoru prostřednictvím Arduino UNO je velice jednoduché. [38]

Hlavní nevýhodou senzoru je, že senzor je schopen vysílat data 1x za 2 sekundy, což by při některých aplikacích nemuselo stačit, a proto je třeba při výběru senzoru zohlednit všechny parametry. [38]

Senzor může být napájen napětím 3,6 - 6 V, takže může být lehce napájen přes samotné Arduino UNO. Na měření teploty používá termistor a na měření vlhkosti používá kapacitní senzor pro měření vlhkosti. Senzor je schopen měřit vlhkost ovzduší v rozmezí mezi 0–100 % s přesností  $\pm 2$  % vlhkosti. A je schopen měřit teplotu v rozmezí od  $-40$  °C až do  $+80$  °C s přesností  $\pm 0,5$  °C. [38]

S pomocí tohoto senzoru můžeme po celou dobu sledovat, jaká je teplota a vlhkost v blízkosti chovných komůrek. A s těmito daty dále sledovat jaké mají okolní podmínky vliv na larvy chrostíků.



*Obrázek 12) senzor teploty a vlhkosti DHT22 [převzato upraveno 39]*

## **10.4 Fotorezistor**

Základem senzoru je polovodič, u kterého se elektrický odpor snižuje s množstvím dopadajících fotonů, díky tomuto jevu můžeme pomocí fotorezistoru měřit intenzitu osvětlení. [40]

Jelikož larvy v přírodě mají normálně pevně daný den a noc, mohlo by je příliš intenzivní světlo či jeho vystavení v hodiny, v kterých je normálně v přírodě noc rozhodit a případně ovlivnit jejich vývoj, díky fotorezistoru můžeme tento jev sledovat a dokumentovat. [40]

## **10.5 HC-SRO4**

Senzor HC-SRO4 se skládá z dvou ultrasonických měničů. První sloužící jako vysílač, který vysílá 40KHz ultrasonické vlny, které produkuje převodem z elektrického proudu. A druhý sloužící jako přijímač, převádí příchozí vlny na napětí, z kterého se později

dá určit v jaké vzdálenosti je předmět. Senzor detekuje v rozmezí od 2 do 400 cm s přesností na 3 mm.[41] [42]

Tento senzor sleduje výšku hladiny u chovných komůrek, za pomoci uchyceného tělesa, které se vznáší na hladině.



*Obrázek 13) Senzor vzdálenosti [autor]*

## 10.6 Magnetický spínač

Magnetický spínač se skládá z dvou feromagnetických magnetů, uzavřených ve skleněné kapsli, aby byly magnety izolované od okolních jevů. Pokud je těmto feromagnetickým magnetům přiložen další magnet, tak se magnety díky elektromagnetickému poli sepnou. [43]

Magnetický snímač je v této práci využit s člunkovým srážkoměrem, snímač zaznamenává, kdy a kolikrát se srážkoměr naplní a překllopí a s těmito informacemi se dále vypočítá průtok.



*Obrázek 14) magnetický spínač [autor]*



*Obrázek 15) člunkový srážkoměr [převzato upraveno 44]*

## 10.7 ESP8266

ESP8266 je levný WI-FI mikročip se zabudovaným TCP/IP. Začal se velice využívat, když byl uveden na trh jeho modul ESP-01 v roce 2014. Čip je vyráběn firmou Espressif z Šanghaje. Vyžaduje napájení 3,3 V a v závislosti na jeho módu spotřebuje až 300 mA. Má k dispozici 2 GPIO piny.

ESP8266 má dnes již nástupce v podobě ESP32. V tomto projektu jsem se rozhodl využít starší verzi, jelikož nová verze nemá výhodu ovlivňující tento projekt, její cena je výrazně větší a jelikož je mladší má méně diskusních fór s menším obsahem než ESP8266. [45] [46]

ESP8266 v tomto projektu zařizuje připojení zařízení k internetu přes Wi-Fi a také posílání informace na web a z webu.



*Obrázek 16) ESP8266-01 [autor]*

## 10.8 Pir senzor AM312

Pir senzor AM312 je určený pro detekci pohybu, který dokáže spolehlivě detekovat na 3 metry. K detekci pohybu používá pyroelektrický senzor. [47]

S pomocí tohoto senzoru budu sledovat pohyb, který se vyskytuje v místnosti s larvami. A dlouhodobě pomocí tohoto sledování zjistil, pokud má na ně pohyb v místnosti a možné následné otřesy po pohybu vliv.



*Obrázek 17) Pir senzor AM312 [autor]*

## 10.9 Krokové motory a jejich drivery

Jako první krokový motor byl zvolen 17PS-C103-P1, zkráceně NEMA 17, tento motor má parametry 42x42x39 mm, jeden krok je 3,6 °, takže na plnou rotaci je zapotřebí  $360/3,6 = 100$  kroků a jeho krouticí moment = 0,22 Nm. [48]



*Obrázek 18) NEMA 17 [autor]*

Jako druhý krokový motor byl zvolen 23HY1406, zkráceně NEMA 23, která má parametry 56x56x50 mm. Jeden krok je  $1,8^\circ$ , takže na celou rotaci potřebuje  $360/1,8 = 200$  kroků. [49]



*Obrázek 19) NEMA 23 [autor]*

Jako jeho driver pro NEMA 17 byl zvolen A4988, který má parametry: minimální provozní napětí  $U_{min} = 8\text{ V}$ , maximální provozní napětí  $U_{max} = 35\text{ V}$ , maximální trvalý proud na fázi  $I_t = 1\text{ A}$  a maximální špičkový proud na fázi  $I_\xi = 2\text{ A}$ . [50]



*Obrázek 20) driver krokového motoru A4988 [autor]*

Jako driver pro NEMA 23 byl zvolen TB6600. Které má minimální provozní napětí  $U_{min} = 9 \text{ V}$ , maximální provozní napětí  $U_{max} = 40 \text{ V}$ , minimální trvalý proud na fázi  $I_t = 0,7 \text{ A}$  a maximální špičkový proud na fázi  $I_s = 4 \text{ A}$ . [51]

## 11 Výroba součástek

Součástky byly vytisknuty na 3D tiskárně Prusa Mini. Použitý materiál na 3D tisk je Polylatic Acid (PLA).

### 11.1 Peristaltická pumpa

Byly navrženy a vytisknuty 3 verze peristaltické pumpy, 2 pro NEMA 23 a 1 pro NEMA 17. Dvě verze pro NEMA 23 jsou designově stejné a hlavní rozdíl mezi nimi je jejich rozměr.

První verze je designovaná pro trubičku s vnitřním průměrem  $d = 1$  mm a s tím plánovaným a relativně malým průtokem pouhých  $Q = 0,11$  ml/s při rychlosti otáčka za 1 sekundu. Tento průtok není dostatečný na zásobování vody pro většinu druhů chrostíků a tato verze je spíše navržena pro menší živočichy, kteří potřebují menší průtok, či druhy chrostíků, které nejsou závislé na průtoku.

Rozměry:

Poloměr plošinky:  $d_{p1} = 23$  mm

Vzdálenost středu ramen od středu rotace:  $l_{r1} = 19,5$  mm

Poloměr ramen:  $r_{r1} = 5$  mm

Vnitřní průměr obalu  $D_{o1} = 51,2$  mm

Počet ramen:  $N_{r1} = 3$

Rozměry trubičky:

Vnitřní průměr:  $d_{trub1} = 1$  mm

Vnější průměr:  $D_{t1} = 3$  mm

Tloušťka stěny:  $t_1 = 1$  mm

Okluze:  $O_{c1} = 0,2$  (20%)

Výpočty:

Mezera mezi krajem ramen a obalem [7]

$$S = 2t * (1 - O_c) = 1,6 \text{ mm}$$

(3)

Plocha průřezu vnitřní strany trubičky [7]

$$S_{trub} = \frac{\pi d_{trub}^2}{4} = 0,79 \text{ mm}^2$$

(4)

Vzdálenost mezi rameny [7]

$$l_{120} = 2\pi \left( \frac{\frac{D}{2} + \left( \frac{D}{2} - D_{trub} \right)}{6} \right) - \frac{D_{trub}}{2} = 48,97 \text{ mm}$$

(5)

Obsah mezi rameny [7]

$$V_g = l_{120} * S_{trub} = 38,6 \text{ mm}^3$$

(6)

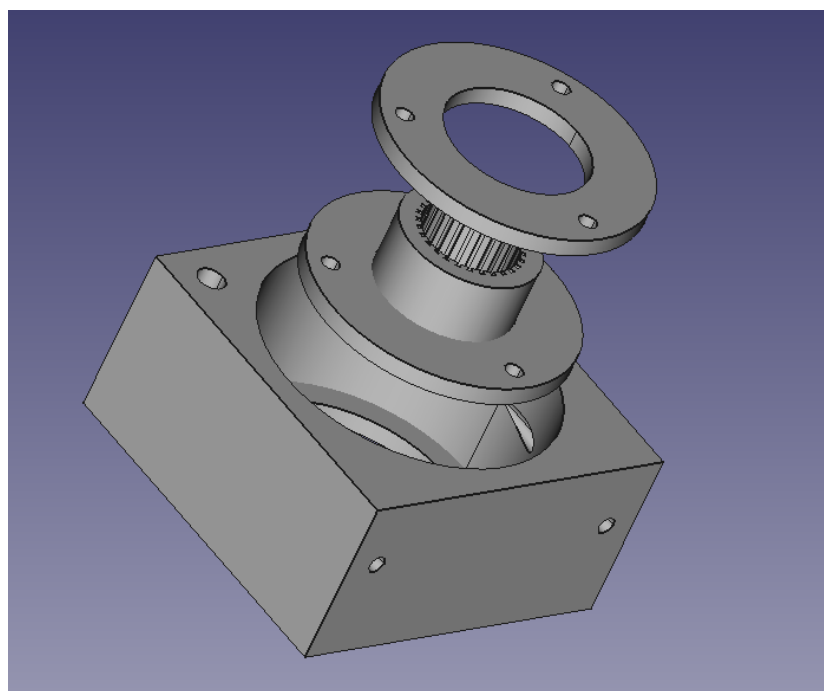
Průtok za 1 otáčku/s, použita rovnice (2) ze stránky 15 [7]

$$Q_{th} = V_g * N_r * n = 116,07 \text{ mm}^3/\text{s}$$

Pomocí těchto výpočtů známe rozměry obalu, obou rotačních disků, potřebných ložisek a potřebné trubičky. Jako ložiska pro tuto verzi jsem zvolil 10x4x3mm. [7]

Obal, na který peristaltická pumpa vyvíjí tlak má v jeho nejslabším místě tloušťku 4,4mm. Materiál v tlustších částech obalu jsem mohl oříznout na ušetření použitého materiálu, ale zanechal jsem ho, abych zvýšil pevnost celé části.

Rotační disky jsou 4 mm tlusté a jsou v nich 3 otvory pro ramena, které jsou symetricky rozmístěné. Jeden disk funguje jako nosný a nachází se v něm i otvor pro hřídel, druhý disk byl přidán, aby byla zvýšena stabilita ramen a sníženo namáhání na první disk. Druhý disk je pouze nosný a jeho jediný kontakt s prvním diskem je přes ramena, tento disk nemá kontakt s hřídelí.



*Obrázek 21) 1. verze peristaltické pumpy [autor]*



*Obrázek 22) zhotovená peristaltická pumpa [autor]*

Druhá verze je designovaná pro NEMA 17, díky tomu má menší spotřebu, ale s tím i menší výkon. Také tato verze musí mít odlišné uchycení ke krokovému motoru od předchozích dvou verzí. Tato verze má průtok při jedné otáčce z vteřinu  $Q = 1,07 \text{ ml/s}$ . S tímto průtokem by i tento slabší motor měl být schopný zásobovat dostatečný průtok pro larvy chrostíků a jim podobné živočichy.

Poloměr plošiny:  $d_{p2} = 25 \text{ mm}$

Vzdálenost středu ramen od středu rotace:  $l_{r2} = 19,4 \text{ mm}$

Poloměr ramen:  $r_{r2} = 7 \text{ mm}$

Vnitřní průměr obalu  $D_{o2} = 56 \text{ mm}$

Počet ramen:  $N_{r2} = 3$

Rozměry trubičky:

Vnitřní průměr:  $d_{t2} = 3 \text{ mm}$

Vnější průměr:  $D_{t2} = 5 \text{ mm}$

Tloušťka stěny:  $t_2 = 1 \text{ mm}$

Okluze:  $O_{c2} = 0,2 \text{ (20\%)}$

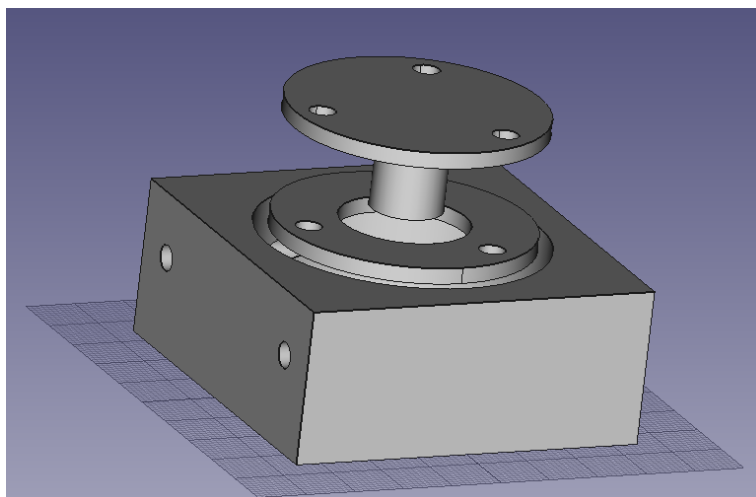
Mezera mezi krajem ramen a obalem  $S = 1,6 \text{ mm}$

Plocha průřezu vnitřní strany trubičky  $S_{trub} = 7,08 \text{ mm}^2$

Vzdálenost mezi rameny  $l_{120} = 50,4 \text{ mm}$

Obsah mezi rameny  $V_g = 356,88 \text{ mm}^3$

Průtok za 1 otáčku/s  $Q_{th} = 1070,64 \text{ mm}^3$



Obrázek 23) 2. verze peristaltické pumpy [autor]

Třetí verze je designovaná také pro NEMA 23, ale tato verze má mnohem větší průtok za otáčku a je optimální pro chov larev chrostíků i jiných živočichů s podobnými potřebami. Tato verze má průtok při jedné otáčce za vteřinu  $Q = 2,5 \text{ ml/s}$ .

Poloměr plošiny:  $d_{p3} = 30 \text{ mm}$

Vzdálenost středu ramen od středu rotace:  $l_{r3} = 24,8 \text{ mm}$

Poloměr ramen:  $r_{r3} = 10 \text{ mm}$

Vnitřní průměr obalu  $D_{o3} = 76 \text{ mm}$

Počet ramen:  $N_{r3} = 3$

Rozměry trubičky:

Vnitřní průměr:  $d_{t3} = 4 \text{ mm}$

Vnější průměr:  $D_{t3} = 8 \text{ mm}$

Tloušťka stěny:  $t_3 = 2 \text{ mm}$

Okluze:  $O_{c3} = 0,2 \text{ (20\%)}$

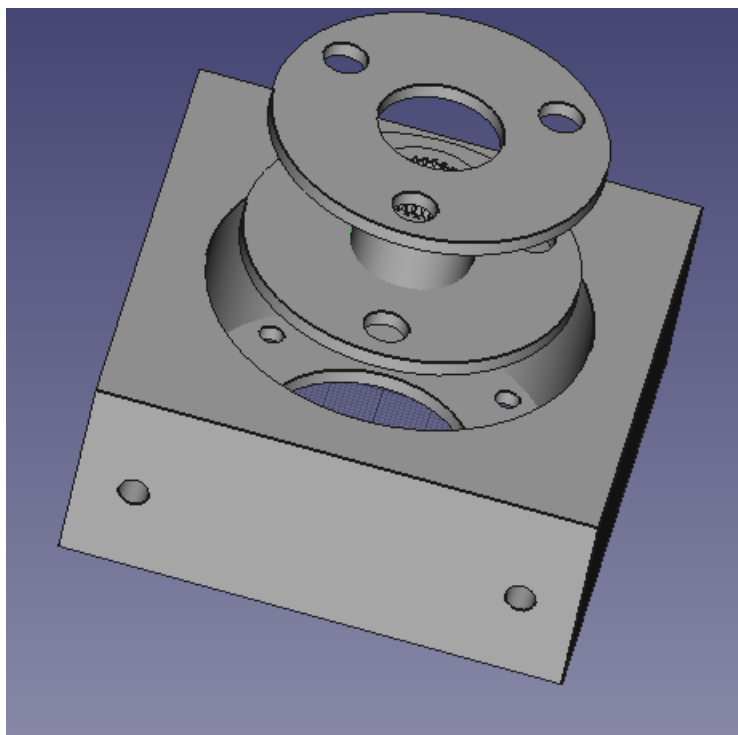
Mezera mezi krajem ramen a obalem  $S_3 = 3,2 \text{ mm}$

Plocha průřezu vnitřní strany trubičky  $S_{trub3} = 12,56 \text{ mm}^2$

Vzdálenost mezi rameny  $l_{120} = 67,2 \text{ mm}$

Obsah mezi rameny  $V_{g3} = 844,03 \text{ mm}^3$

Průtok za 1 otáčku/s  $Q_{th3} = 2532,09 \text{ mm}^3$

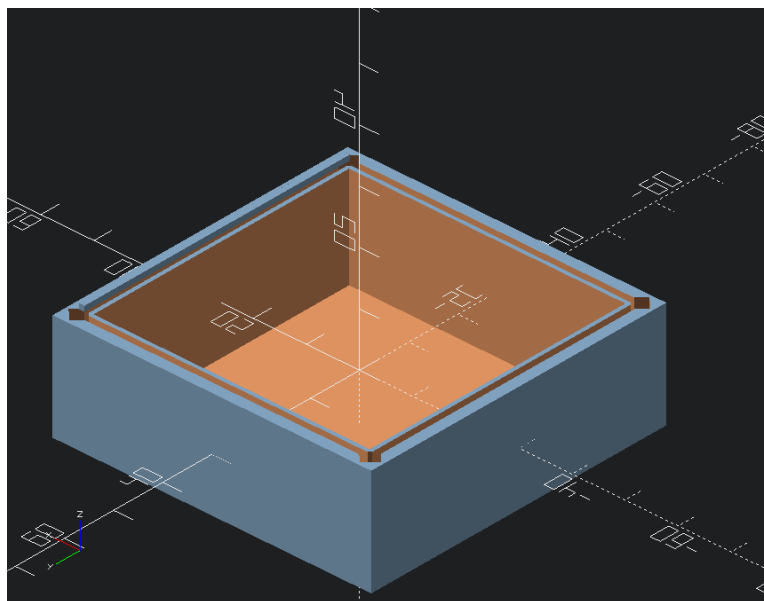


*Obrázek 24) 3. verze peristaltické pumpy [autor]*

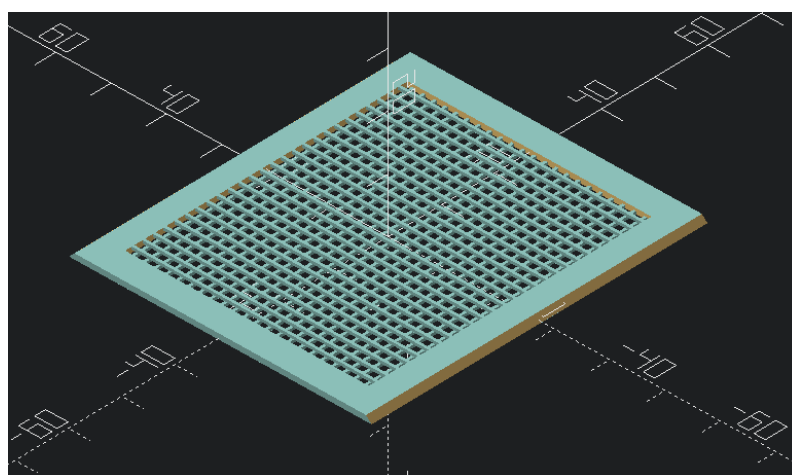
## 11.2 Chovné komůrky

V programu OpenSCAD jsem vymodeloval komůrky pro larvy chrostíků, tyto komůrky jsou vymodelovány tak, aby v každé komůrce byla pouze jedna larva a tak, aby horní částí komůrek mohla plynout voda.

Sestavené komůrky mají rozměry 60x60x90 mm. Jejich boční části jsou tvořeny stěnami s otvory, kterými proudí voda.



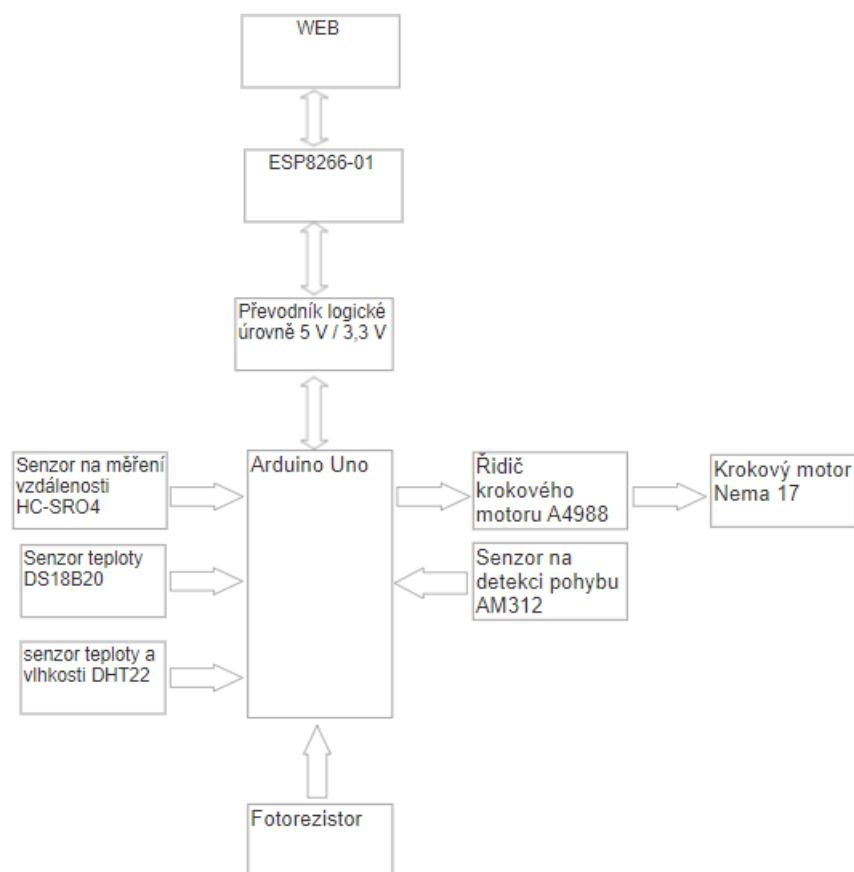
*Obrázek 25) krabíčka pro chov chrostíků [autor]*



*Obrázek 26) stěna krabíčky [autor]*

## 12 Návrh chovného zařízení

Blokový diagram



Obrázek 27) blokový diagram [autor]

### 12.1 Samotný návrh

Samotné místo chovu se skládá z chovných komůrek sériově seřazených tak, aby byly jednotlivé larvy chrostíků od sebe oddělené, jelikož pokud by byli ve společné komůrce, chrostíci by se vzájemně napadali a případně pozřeli. Takto v sérii jsou za sebou umístěny 3 komůrky, které jsou uloženy v komoře s vnitřními rozměry 200x65x100 mm. Těchto komůrek může být více seřazených pod sebou způsobem, že voda vytékající z první komory vtéká na začátek další komory a tím vznikne všude průtok. Do těchto komůrek je

vháněna voda za pomoci peristaltické pumpy. Tato voda pochází ze skladovací nádoby, kde je také měřena její teplota. Do této nádoby také bude voda svedena poté, co projde chovnými komůrkami. Pokud by se z důvodu většího průtoku muselo použít jiné čerpadlo než peristaltické, použil by se v tomto zapojení člunkový srážkoměr na měření průtoku, jelikož v zapojení s peristaltickou pumpou ho můžeme lehce spočítat pouze s jejím výkonem. V komoře, kde se nachází chovné komůrky je měřena výška hladiny díky HC-SRO4 a dále je v této komoře měřena teplota a vlhkost vzduchu za pomoci DHT 22 senzoru. Dále je v komoře měřena intenzita osvětlení pomocí fotorezistoru. Na vnější části chovného zařízení je umístěn senzor na měření pohybu, díky němuž můžeme sledovat pohyb v místnosti

Všechny tyto senzory jsou ovládány přes Arduino UNO, které z nich také sbírá informace. Všechny tyto informace jsou poté posílány dále přes Arduino UNO na ESP8266 wifi modul, přes který je hostován web server, kde jsou tyto informace dostupné, zároveň s možností ovládat krokový motor, který je součástí peristaltické pumpy.

## **13 Zařízení dálkového monitorování**

### **13.1 Raspberry Pi 4**

Jako první možnost pro dálkové ovládání přes internet byla zvolena kombinace Raspberry Pi 4 a Arduino UNO, které byly spojené prostřednictvím USB komunikace.

V této kombinaci bylo Raspberry Pi 4 napojeno na místní Wi-Fi, případně připojeno za pomoci ethernetu a hostovalo na IP adrese této wifi web server. Dále v tomto zapojení bylo Arduino UNO, na které byly přímo připojeny veškeré senzory a také krokový motor sloužící k pohonu peristaltické pumpy. Informace z těchto senzorů byly zaslány na Arduino UNO a poté se přes USB komunikaci posílaly dále na Raspberry Pi 4, z kterého se dále přenášely na webovskou stránku. Zároveň by fungovala i komunikace z web serveru na Raspberry Pi 4 a poté dále na Arduino UNO, pomocí níž by se ovládal krokový motor. Tento server byl hostován za pomoci NGINX, které jsem poté programoval jazyky PHP a HTML.

Bohužel po značné době experimentů s touto sestavou se ukázalo, že i když komunikace funguje oběma směry, zdá se, že v určitém kroku komunikace se náhodně ztrácejí data a posouvají se bity, čímž se může přeskočit část z původní zprávy. A někdy určité bity přišly přeházené, čímž formovaly pouze nesrozumitelnou zprávu.

## 13.2 ESP8266

Poté bylo rozhodnuto vyzkoušet Wi-Fi moduly s jinými způsoby komunikace. Hned mezi prvními kandidáty byly wifi moduly ESP8266 a ESP32, z nichž je ESP32 pokročilejší verze vůči staršímu ESP8266. ESP32 a ESP8266 jsou si velice podobné ohledně používání na Wi-Fi, ESP32 má výhodu, že má menší spotřebu, ale vůči ESP8266 je výrazně dražší. Navíc ESP8266 má díky delší době existence větší komunitu a díky tomu značně rozsáhlejší diskusní fóra, takže případné problémy mohou být lehčí vyřešit. Z tohoto důvodu bylo rozhodnuto jako první vyzkoušet sestavu s ESP8266, které bude stejně jako Raspberry Pi 4 hostovat web server přes Wi-Fi, ale v tomto případě budou ESP8266 a Arduino UNO propojené přes UART komunikaci, která je mnohem starší, pomalejší a jednodušší než USB komunikace, ale pro tento projekt je adekvátní.

Poté co se začalo pracovat s ESP8266 byly zpočátku pouze problémy s nahráváním dat a jeho napájením, jelikož nemůže být napájené přímo z 5 V napájení, které poskytuje Arduino UNO, protože ESP8266 má nejvyšší dovolené napětí 3,6 V, jinak dojde k poškození čipu. A 3,3 V napájení, které poskytuje Arduino UNO také není ideální, jelikož tento pin poskytuje pouze 50 mA a ESP8266 maximálně potřebuje 150 mA, takže v tomto zapojení se riskuje, že se poškodí Arduino UNO. Správné napájení, kde Arduino UNO funguje jako zdroj, je redukovat 5 V napájení za pomoci převodníku logické úrovně na 3,3

V, jelikož 5 V pin na Arduino UNO poskytuje až 200 mA, tak s tímto připojením bude mít ESP8266 všechny potřebné parametry.

Poté, co byly vyřešeny tyto základní problémy, začalo ESP8266 fungovat bez problémů a brzy s ním bylo dosaženo bodu, kam se předtím došlo s Raspberry Pi 4 a velice brzy poté s ESP8266 fungovala komunikace oběma směry, tedy s informacemi, které poslalo Arduino UNO na web a opačně. A to bez ztrát dat či jejich přetvoření. Proto bylo pro hostování webu využíváno v této práci ESP8266 jako spolehlivý a lehce opravitelný nástroj. Dále se muselo rozhodnout, jestli chci tyto informace publikovat a ovládat přes různé cloudové služby, nebo jestli hostovat web server. Nakonec bylo vybráno hostovat server na IP adrese místní Wi-Fi, jelikož s ním zůstává více možností a není potřeba se zabírat limity cloudových služeb.

```
// zjistí jestli je seriová linka dostupná
while (Serial.available() > 0) {
    inputChar = Serial.read();

    // pokud jsme uprostřed přenosu dat, pokračujeme k parsování
    if (recvInProgress == true) {
        // pokud se znak nerovná ukončovacímu znaku ">" pokračujeme dál
        if (inputChar != endMarker) {
            // uložíme aktuální znak co přišel ze sériové linky do pole znaků na konkrétní pozici
            receivedChars[poziceZnaku] = inputChar;
            poziceZnaku++;
            //
            if (poziceZnaku >= numChars) {
                poziceZnaku = numChars - 1;
            }
        }
        else {
            receivedChars[poziceZnaku] = '\0'; // terminate the string
            recvInProgress = false;
            poziceZnaku = 0;
        }
    }
    // pokud zrovna parsovaný znak je "<", tak zahájíme process parsování
    else if (inputChar == startMarker) {
        recvInProgress = true;
        // jakmile přijdou nová data je potřeba je znovu zpracovat až je načteme
        dataRefreshRequested = true;
    }
}

if (dataRefreshRequested == true) {
    parseData(receivedChars);
    dataRefreshRequested = false;
}
```

*Obrázek 28) příklad kódu z ESP8266 [autor]*

Bylo tedy použité ESP8266 na vytvoření webové stránky, tímto způsobem je zařízení schopné přijímat i vysílat informace. Tento způsob funguje i na normálně nastavené Wi-Fi, ale pouze v uzavřeném oběhu, takže na web se dostane jenom zařízení připojené na tu stejnou Wi-Fi. Aby byl web přístupný z internetu, musí se na Wi-Fi nastavit správný port forwarding.

### 13.3 Přenos dat do ESP8266

Bylo rozhodnuto programovat ESP8266 přes Arduino UNO, tento přenos potřebuje má několik kroků, bez kterých přenos nebude fungovat.

Celé programování a později přenos dat se provádí v programu Arduino IDE. V tomto programu se musí stáhnout knihovna pro ESP8266. Při přenosu přes Arduino UNO je nutné, aby Arduino UNO mělo v sobě nahraný pouze prázdný kód, jinak přenos nebude fungovat.

Dále je nutné zapnout v zapojení, kde je s arduinem normálně propojeno, s tím že piny na přenos informací má propojené jako RX-RX a TX-TX. A dále musí být ESP8266 v režimu pro programování, toho dosáhne, pokud jeho pin GPI-0 uzemněn a poté se jeho reset pin musí zapojit aspoň na 1 sekundu do GND, odkud se poté musí vyjmout.

Poté se mohou informace nahrát přes Arduino UNO. Po nahrání informací se musí ESP8266 uvést zpátky do režimu práce, toho dosáhne pokud se jeho GPI-0 pin vypojí z GND a poté se opět jeho RESET pin zapojí na 1 sekundu do GND, načež se znovu vyjme.

Pro následovnou práci s arduinem unem se také musí změnit zapojení komunikačních pinů, tedy se musí vrátit do zapojení RX-TX TX-RX. Při tomto zapojení spolu Arduino UNO a ESP8266 komunikují za pomoci UART komunikace. Tedy je velice důležité, aby obě zařízení měli stejný Baud rychlosti. Jelikož nemůžeme informace posílat jednotlivě, jelikož škodí prohození informací mezi sebou. K tomuto účelu slouží kód znázorněný na obrázku 28, který ukládá data od symbolu < do symbolu > do pole v jasném pořadí a díky tomu je již jednoduché je rozdělit podle pořadí.

## 14 Výsledky

Vytvořené zařízení má několik možných průtoků, jelikož pro tento účel byly navrhnuty a vytvořeny 3 různě verze peristaltické pumpy. Dále toto zařízení zaznamenává parametry, které jsou nutné pro jejich úspěšný chov. Měří se výška hladiny za, pomocí dálkového senzoru. Dále se měří teplota vody a ovzduší kolem chovných komůrek, kde se také měří vlhkost. Intenzita osvětlení se měří za pomoci fotoresistoru, aby bylo možné sledovat případné změny ve vývoji larev při příliš intenzivním světle, či přílišného osvětlení, které se v přírodě nevyskytuje. Do chovných komůrek je vháněna voda peristaltickou pumpou, která je kdykoliv ovladatelná z internetu. Rychlost této vody je také měřena výkonem peristaltické pumpy a pokud by bylo nezbytné využít jinou pumpu může se rychlost měřit člunkovým srážkoměrem. Kolem chovného zařízení je zaznamenáván případný pohyb, aby se mohly později zjistit, pokud jsou larvy chrostíků ovlivněny pohybem v místnosti či vibracemi vyvolanými tímto pohybem.

Dálkové monitorování a ovládání je zařízení přes ESP8266, které hostuje web na místní Wi-Fi. Na tento web jsou přenášeny veškeré informace získané z senzorů a také je z něj možné ovládat peristaltickou pumpu.

## 15 ZÁVĚR

Výsledkem práce je realizace zařízení, v kterém je možné chovat larvy chrostíků a další živočichy s podobnými potřebami. Toto zařízení sleduje veškeré potřebné parametry, které tyto larvy ovlivňují. Tedy teplotu vody, vnější teplotu, intenzitu osvětlení, vlhkost, výšku hladiny, rychlost průtoku vody a pohyb v okolí zařízení. Dále toto zařízení posílá veškerá data přes ESP8266 na web, odkud se dá pomocí tohoto zapojení řídit průtok vody.

Během zařizování dálkového připojení jsem narazil na problém, jelikož jsem se nejdříve rozhodl používat Raspberry Pi 4 v sériovém zapojení s Arduino UNO, bohužel se toto zapojení ukázalo jako velice problémové, jelikož data se v něm velice nestabilní, jelikož při experimentování s přenosem dat se text náhodně posouval a jeho části se přehazovaly.

Poté se začalo používat ESP8266 byly nejdříve problémy s elektronickým zapojením a přenosem dat, ale po nastudování dostatečné literatury se tento problém vyřešil. A brzy poté se ukázalo, že toto zapojení funguje účinněji než zapojení s Raspberry Pi, jelikož s ESP8266 velice rychle fungoval přenos dat oběma směry a nikdy nebyl nijak posunut a poslané informace zůstaly vždy v celku.

Dále bylo nutné naučit se model ve FreeCAD a OpenSCAD a vytvořit za pomoci těchto programů peristaltickou pumpu a chovné komůrky pro larvy chrostíků. Nakonec bylo rozhodnuto vytvořit 3 verze pro peristaltickou pumpu, aby byla možná lepší flexibilita při výběru průtoku pro různé druhy larev, nebo případně jiných živočichů chovaných v tomto zařízení.

## Literatura

- [1] NOVÁK, Karel. Chrostíci, jejich chov a pozorování. Živa, 1962, 3, 104-110 s.
- [2] NOVÁK, Karel. Trichoptera – Chrostíci. V. Skuhrový et al: Metody chovu hmyzu (Rearings of insects), 202-207s, Academia, Praha
- [3] Chrostíci (Trichoptera) | Hmyz | jablib.cz. Fotografie, rybaření, houbaření, ptáci, ryby, houby. [online]. [cit. 06.04.2022]. Dostupné z: <https://www.jablib.cz/muskareni/chrostici-trichoptera/>
- [4] Co z vodního hmyzu žije na slatiništích | Víte...? | věda.muni.cz. Magazín M: Zprávy z MUNI | em.muni.cz [online]. [cit. 06.04.2022]. Dostupné z: <https://www.em.muni.cz/vite/4550-co-z-vodniho-hmyzu-zije-na-slatinistich>
- [5] HRABOVSKÝ, Michal. Bezdrátová meteorologická stanice [online]. Brno, 2019 [cit. 2022-04-07]. Dostupné z: <https://theses.cz/id/yfpl4f/>. Bakalářská práce. Vysoké učení technické v Brně.
- [6] TEPLoměRY a měření teploty, digitální teploměry, vpichovací teploměry, registrační teploměry, infra teploměry, dotykové teploměry, HACCP teploměry pro potravinářství, kontrolní teploměry, bimetalové teploměry. [cit. 2022-04-07]. Dostupné z: <https://theses.cz/id/ob5icj/>.
- [7] CHARVÁT, Jakub. Přípravek pro řízení peristaltické pumpy [online]. Brno, 2021 [cit. 2022-04-07]. Dostupné z: <https://theses.cz/id/ob5icj/>. Bakalářská práce. Vysoké učení technické v Brně.
- [8] Arduino - Home [online]. [cit. 2022-04-07]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [9] Introduction to Arduino IDE - The Engineering Projects. Home - The Engineering Projects [online]. [cit. 2022-04-07]. Dostupné z: <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>
- [10] HIPÍK, Martin. Připojení vývojové desky Arduino k bezdrátové síti WiFi pomocí modulu ESP8266 [online]. Zlín, 2018 [cit. 2022-04-07]. Dostupné z: <https://theses.cz/id/onqt4l/>. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Ing. Jan Dolinay, Ph.D.

- [11] Introduction to Arduino Uno - The Engineering Projects. Home - The Engineering Projects [online]. [cit. 2022-04-07]. Dostupné z: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>
- [12] Advantages and disadvantages of using arduino [online] [cit. 2022-04-07]. Dostupné z: <https://engineerexperiences.com/advantages-and-disadvantages.html>
- [13] History of 3D Printing: It's Older Than You Think [Updated]. Redshift | Exploring the Future of Making [online]. Copyright © 2022 Autodesk, Inc All Rights Reserved [cit. 07.04.2022]. Dostupné z: <https://redshift.autodesk.com/history-of-3d-printing/>
- [14] SVOBODA, Tomáš. 3D tisk a 3D skenování v animaci [online]. Praha, 2020 [cit. 2022-04-11]. Dostupné z: <https://theses.cz/id/6q50ty/>. Bakalářská práce. Vysoká škola kreativní komunikace, s.r.o. Vedoucí práce MgA. Martin Hovorka.
- [15] What Materials Are Used for 3D Printing? | Sharretts Plating Company. Industrial Plating Company | Metal Plating Services [online]. Copyright © 2017. All Rights Reserved [cit. 07.04.2022]. Dostupné z: <https://www.sharrettsplating.com/blog/materials-used-3d-printing/>
- [16] CAD Software | 2D And 3D Computer-Aided Design | Autodesk. Autodesk | 3D Design, Engineering & Construction Software [online]. Copyright © 2022 Autodesk Inc. All rights reserved [cit. 07.04.2022]. Dostupné z: <https://www.autodesk.com/solutions/cad-software>
- [17] 12 best CAD software 2022 (free & paid). WiseStamp - Generate & manage professional email signatures [online]. Copyright © WiseStamp2022 [cit. 07.04.2022]. Dostupné z: <https://www.wisestamp.com/blog/best-cad-software/>
- [18] OpenSCAD - About. OpenSCAD - The Programmers Solid 3D CAD Modeller [online]. [cit. 07.04.2022]- Dostupné z: <https://openscad.org/about.html>
- [19] OpenSCAD • BI-3DT • FIT ČVUT Course Pages. FIT CTU Courses [online]. Copyright © [cit. 11.04.2022]. Dostupné z: <https://courses.fit.cvut.cz/BI-3DT/tutorials/openscad.html>
- [20] SOLIDWORKS řešení | SolidVision. SolidVision [online]. [cit. 07.04.2022]. Dostupné z: <https://www.solidvision.cz/solidworks-reseni>

- [21] Robotic arm - FreeCAD Forum. FreeCAD Forum - FreeCAD forum [online]. Copyright © phpBB Limited [cit. 08.04.2022]. Dostupné z: <https://forum.freecadweb.org/viewtopic.php?style=10&t=39553>
- [22] 3D graphics for SolidWorks - DEVELOP3D. DEVELOP3D - Frontpage [online]. Copyright © 2008 [cit. 08.04.2022]. Dostupné z: <https://develop3d.com/features/3d-graphics-for-solidworks/>
- [23] Principy průmyslových čerpadel – 7.díl - hadicová čerpadla | Automatizace.HW.cz. Automatizace.HW.cz | Elektronika v automatizaci [online]. [cit. 07.04.2022]. Dostupné z: <https://automatizace.hw.cz/principy-prumyslovych-cerpadel-7dil-hadicova-cerpadla>
- [24] internet věcí (IoT): definice, příklady využití, produkty | Rascasone. WEB & MOBILE DEVELOPMENT AGENCY | Rascasone [online]. Copyright © [cit. 06.04.2022]. Dostupné z: <https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie>
- [25] Elektronická učebnice - ELUC. Elektronická učebnice - ELUC [online]. [cit. 07.04.2022]. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/809>
- [26] JOCHEC, Jan. Řízení krokových motorů pomocí programovatelného logického automatu [online]. Brno, 2021 [cit. 2022-04-13]. Dostupné z: <https://theses.cz/id/w20z1m/>. Bakalářská práce. Vysoké učení technické v Brně.
- [27] Stepper motors and drives, what is full step, half step and microstepping?. RS Components International [online]. [cit. 07.04.2022]. Dostupné z: <https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping>
- [28] Krokový motor – druhy a příklady aplikací krokových motorů | Elektronické díly. Distributor a obchod online - Transfer Multisort Elektronik. Redirecting to /en/ [online]. [cit. 07.04.2022]. Dostupné z: <https://www.tme.eu/cz/news/library-articles/page/41861/krokovy-motor-druhy-a-priklady-aplikaci-krokovych-motoru/>
- [29] \*DH servis\* Vývoj a výroba elektroniky na zakázku [online]. [cit. 07.04.2022]. Dostupné z: <http://www.dhservis.cz/iic.htm>

- [30] What is Baud Rate & Why is it important?. Premium Sensing Solutions | Setra Systems [online]. Copyright © 2021 Setra Systems. All Rights Reserved. [cit. 11.04.2022]. Dostupné z: <https://www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1>
- [31] Elektronická učebnice - ELUC. Elektronická učebnice - ELUC [online]. [cit. 07.04.2022]. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/874>
- [32] USB – Wikipedie. [online]. [cit. 07.04.2022]. Dostupné z: <https://cs.wikipedia.org/wiki/USB>
- [33] Jak fungují porty USB | PeKro spol. s r.o.. Prodej, servis a opravy počítače tiskárny nejen v Brně | PeKro spol. s r.o. [online]. [cit. 07.04.2022]. Dostupné z: <https://www.pekro.cz/jak-funguji-porty-usb-A/>
- [34] IIC I2C 5V na 3.3V Obousměrný Převodník Logické Úrovně | dratek.cz. dratek.cz: VELKOOBCHOD, MALOOBCHOD S ARDUINEM [online]. Copyright © Copyright ECLIPSE s.r.o. [cit. 06.04.2022]. Dostupné z: <https://dratek.cz/arduino/1481-iic-i2c-5v-na-3.3v-obousmerny-prevodnik-logicke-urovne.html>
- [35] Převodník 5V na 3,3V logické úrovně IIC/I2C | rase.cz. rase.cz – specialista na konektory [online]. [cit. 07.04.2022] Dostupné z: <https://www.rase.cz/prevodnik-5v-na-3-3v-p24218/>
- [36] High Temp Waterproof DS18B20 Digital temperature sensor. Home [online]. [cit. 07.04.2022]. Dostupné z: <https://shop.evilmadscientist.com/productsmenu/717>
- [37] Teplotní čidlo Dallas originál DS18B20 vodotěsné - HWKITCHEN. Ochutnejte s námi bastlení! | HWKitchen.cz [online]. Copyright © HWKITCHEN, všechna práva vyhrazena [cit. 08.04.2022]. Dostupné z: <https://www.hwkitchen.cz/teplotni-cidlo-dalls-original-ds18b20-vodotesne/>
- [38] 5. DHT22 - Digital Temperature and Humidity Sensor. [online]. [cit. 07.04.2022]. Dostupné z: <https://cityos-air.readme.io/docs/4-dht22-digital-temperature-humidity-sensor>
- [39] Digitální teploměr a vlhkoměr, THT DHT22 | GM electronic, spol. s r.o.. GM electronic | elektronické součástky, komponenty . | GM electronic, spol. s r.o. [online]. [cit. 07.04.2022]. Dostupné z: <https://www.gme.cz/digitalni-teplomer-a-vlhkomer-s-dht22>

- [40] Fotorezistor: zařízení, princip činnosti, vlastnosti. ElecroExpert [online]. [cit. 07.04.2022]. Dostupné z: <https://electroexp.com/cs/chto-takoe-fotorezistory.html>
- [41] Arduino měřič vzdálenosti ultrazvukový | dratek.cz. dratek.cz: VELKOOBCHOD, MALOOBCHOD S ARDUINEM [online]. Copyright © Copyright ECLIPSE s.r.o. [cit. 06.04.2022]. Dostupné z: <https://dratek.cz/arduino/846-eses-ultrazvukovy-meric-vzdalenosti-hc-04-pro-jednodeskove-pocitace.html>
- [42] How HC-SR04 Ultrasonic Sensor Works & How to Interface It With Arduino. Last Minute Engineers - [online]. Copyright © 2021 LastMinuteEngineers.com. All rights reserved. [cit. 06.04.2022]. Dostupné z: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>
- [43] Reed Switch - Reed Switch Info. Wayback Machine [online]. [cit. 06.04.2022]. [cit. 07.04.2022]. Dostupné z: <https://web.archive.org/web/20140101000815/http://reed-switch-info.com/>
- [44] Srážkoměr SR03 500cm2 | FIEDLER. Homepage | FIEDLER | Electronics for ecology [online]. Dostupné z: <https://www.fiedler.company/cs/produkty/snimace-meteorolog-velicin/destove-srazky/srazkomer-sr03>
- [45] AI-Thinker ESP8266 ESP-01 Wifi microcontroller - Solarbotics Ltd.. Electronic Products Calgary | Parts & Tools | Solarbotics Ltd. [online]. [cit. 07.04.2022]. Dostupné z: <https://www.solarbotics.com/product/29246>
- [46] ESP8266 - Wikipedia. [online]. [cit. 07.04.2022]. Dostupné z: <https://en.wikipedia.org/wiki/ESP8266>
- [47] AM312 Low Power Ultra-Miniature PIR Sensor. Hobby Components [online]. Copyright © 2022 [cit. 06.04.2022]. Dostupné z: <https://hobbycomponents.com/sensors/1065-am312-low-power-ultra-miniature-pir-sensor>
- [48] Krokové motory Minebea - Ukončeno - CNC Fórum. CNC Fórum - Obsah [online]. [cit. 07.04.2022]. Dostupné z: <https://forum.strojirenstvi.cz/viewtopic.php?t=3830>
- [49] STEPPER MOTOR 23HY1406 | Vector Four Engineering. Vector Four Engineering [online]. [cit. 07.04.2022]. Dostupné z: <https://vector4engineering.com/product/nema-23/>

[50] Driver, řadič pro krokový motor, modul s A4988 : Půhy.cz. Půhy.cz - obchod s technikou a elektronikou [online]. Copyright © [cit. 13.04.2022]. Dostupné z: <https://www.puhy.cz/driver-radic-pro-krokovy-motor-modul-s-a4988-308557.html>

[51] MCI Electronics - Electrónica para makers y profesionales DIY [online]. Copyright © [cit. 11.04.2022]. Dostupné z: [https://www.mcielectronics.cl/website\\_MCI/static/documents/TB6600\\_data\\_sheet.pdf](https://www.mcielectronics.cl/website_MCI/static/documents/TB6600_data_sheet.pdf)

[52] Build an ESP8266 Web Server - Code and Schematics (NodeMCU) | Random Nerd Tutorials. Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi [online]. Copyright © 2013 [cit. 13.04.2022]. Dostupné z: <https://randomnerdtutorials.com/esp8266-web-server/>

[53] Is there a easier way of parsing a string? - Project Guidance - Arduino Forum. Arduino Forum [online]. Dostupné z: <https://forum.arduino.cc/t/is-there-a-easier-way-of-parsing-a-string/578360>

## Seznam obrázků

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Obrázek 1) dospělý chrostík [převzato upraveno 3].....                  | 8  |
| Obrázek 2) larva chrostíka [převzato upraveno 4] .....                  | 9  |
| Obrázek 3) Arduino vývojové prostředí [autor] .....                     | 13 |
| Obrázek 4) Arduino UNO Pinout [převzato upraveno 11] .....              | 14 |
| Obrázek 5) FreeCAD [převzato upraveno 21] .....                         | 17 |
| Obrázek 6) OpenSCAD [převzato upraveno 19].....                         | 18 |
| Obrázek 7) SolidWorks [převzato upraveno 22] .....                      | 19 |
| Obrázek 8) princip peristaltické pumpy [převzato upraveno 23] .....     | 20 |
| Obrázek 9) UART přenos informací [převzato upraveno 31] .....           | 26 |
| Obrázek 10) převodník logické úrovně [převzato upraveno 35] .....       | 27 |
| Obrázek 11) senzor teploty DS18B20 [převzato upraveno 37].....          | 28 |
| Obrázek 12) senzor teploty a vlhkosti DHT22 [převzato upraveno 39]..... | 29 |
| Obrázek 13) Senzor vzdálenosti [autor] .....                            | 30 |
| Obrázek 14) magnetický spínač [autor] .....                             | 30 |
| Obrázek 15) člunkový srážkoměr [převzato upraveno 44] .....             | 31 |
| Obrázek 16) ESP8266-01 [autor] .....                                    | 31 |
| Obrázek 17) Pir senzor AM312 [autor] .....                              | 32 |

|                                                         |    |
|---------------------------------------------------------|----|
| Obrázek 18) NEMA 17 [autor].....                        | 33 |
| Obrázek 19) NEMA 23 [autor].....                        | 33 |
| Obrázek 20) driver krokového motoru A4988 [autor] ..... | 34 |
| Obrázek 21) 1. verze peristaltické pumpy [autor] .....  | 37 |
| Obrázek 22) zhotovená peristaltická pumpa [autor].....  | 37 |
| Obrázek 23) 2. verze peristaltické pumpy [autor] .....  | 39 |
| Obrázek 24) 3. verze peristaltické pumpy [autor] .....  | 40 |
| Obrázek 25) krabice pro chov chrostíků [autor].....     | 41 |
| Obrázek 26) stěna krabice [autor] .....                 | 41 |
| Obrázek 27) blokový diagram [autor] .....               | 42 |
| Obrázek 28) příklad kódu z ESP8266 [autor] .....        | 45 |

## **Seznam příloh**

Příloha č.1: Kód pro ESP8266

Příloha č.2: Kód pro Arduino UNO

# Přílohy

## Příloha č.1: Kód pro ESP8266

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
const byte numChars = 128;
String prijataData[128] = {};

// wifi jmeno heslo
const char* ssid      = "ssid";
const char* password = "heslo";

// wifi port nastavena na 80
WiFiServer server(80);

// nejdulezitejsi promenna
String header;

unsigned long currentTime = millis();

unsigned long previousTime = 0;

const long timeoutTime = 2000;

void setup() {
    Serial.begin(115200);

    // připojení k wifi
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    // vypsání IP adresy
    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    server.begin();
}

void loop() {
    WiFiClient client = server.available();    // poslouchat pro příchozí
    informace
```

```

if (client) { // pokud je dostupny client
    Serial.println("New Client.");
    String currentLine = ""; // string pro nova data
    currentTime = millis();
    previousTime = currentTime;
    while (client.connected() && currentTime - previousTime <=
timeoutTime) { // loop behem spojeni
        currentTime = millis();
        if (client.available()) {

            char c = client.read(); // čtení a pridavani bytu

            header += c;
            if (c == '\n') {

                if (currentLine.length() == 0) {

                    // tato cast by tu vubec nemela byt, ale bez ni nic nefunguje
                    if (header.indexOf("GET /5/on") >= 0) {

                        } else if (header.indexOf("GET /5/off") >= 0) {

                        } else if (header.indexOf("GET /4/on") >= 0) {

                        } else if (header.indexOf("GET /4/off") >= 0) {
                        }
                        else if (header.indexOf("GET /6/input") >= 0) {
                            Serial.println(header);

                        }

                        // začátek html stranky
                        client.println("<!DOCTYPE html><html>");
                        client.println("<head><meta name=\"viewport\"
content=\"width=device-width, initial-scale=1\">");
                        client.println("<link rel=\"icon\" href=\"data:,\>");
                        client.println("<style>html { font-family: Helvetica; display:
inline-block; margin: 0px auto; text-align: center;}");
                        client.println(".button { background-color: #195B6A; border:
none; color: white; padding: 16px 40px;");
                        client.println("text-decoration: none; font-size: 30px;
margin: 2px; cursor: pointer;}");
                        client.println(".button2 {background-color:
#77878A;}</style></head>");

                        // samotná stránka
                        client.println("<body><h1>Data</h1>");

```

```

        client.println("<p>Teplota vzduchu:" + prijataData[0] +
"&deg;C<p>");
        client.println("<p>Vlhkost:" + prijataData[1] + "%<p>");
        client.println("<p>Vyska hladiny:" + prijataData[2] +
"mm<p>");
        client.println("<p>Intenzita osvetleni:" + prijataData[3] +
"<p>");
        client.println("<p>Prutok:" + prijataData[4] + "ml/s<p>");
        client.println("<p>Teplota vody:" + prijataData[5] +
"&deg;C<p>");
        client.println("<p>pohyb:" + prijataData[6] + "jednicka
znamena pohyb, nula znamena klid<p>");

        //přidáme tlačítko s polem pro vstup hodnoty a
javascriptem pro odeslání
        client.println("<p>zadejte rychlost v rpm, zadejte pouze v
tvaru <rychlost><p>");
        client.println("<input type=\"text\" id=\"hodnota1\"> <button
id=tlacitko1>odeslat</button>");
        client.println("<script type=\"text/javascript\">");
        client.println("document.getElementById(\"tlacitko1\").onclick
= function sendData() {");

        client.println("const inputValue =
document.getElementById(\"hodnota1\").value");
        client.println("const Http = new XMLHttpRequest();");
        client.println("const url=\"'/6/input?data=\"'+inputValue;");
        client.println("Http.open(\"GET\", url);");
        client.println("Http.send();");
        client.println("}");
        client.println("</script>");

        client.println();

        break;
    } else { // pokud nova informace - vycistit
        currentLine = "";
    }
    } else if (c != '\r') { // pokud neco dalsiho, pridat na konec
        currentLine += c;
    }
}
// vycistit header
header = "";
// zavrit spojeni
client.stop();
Serial.println("Client disconnected.");
Serial.println("");
}
}
char receivedChars[numChars];

//=====
void initCommunicationProcess() {
    processRawData();
}

```

```

// Funkce která zajišťuje komunikaci přes sériovou linku a data v syrovém
formát eg.<1,2,55,78> uloží do proměnné receivedChars (pole znaků) jako
"1,2,55,78"
void processRawData() {
    static boolean recvInProgress = false;
    char receivedChars[32] = {};
    static byte poziceZnaku = 0;
    char startMarker = '<';
    char endMarker = '>';
    char inputChar = (char) 0;
    char tempChars[numChars];
    boolean dataRefreshRequested = true;

    // zjistí jestli je seriová linka dostupná
    while (Serial.available() > 0) {
        inputChar = Serial.read();

        // pokud jsme uprostřed přenosu dat, pokračujeme k parsování
        if (recvInProgress == true) {
            // pokud se znak nerovná ukončovacímu znaku ">" pokračujeme dál
            if (inputChar != endMarker) {
                // uložíme aktuální znak co přišel ze sériové linky do pole znaků
                // na konkrétní pozici
                receivedChars[poziceZnaku] = inputChar;
                poziceZnaku++;
                //
                if (poziceZnaku >= numChars) {
                    poziceZnaku = numChars - 1;
                }
            }
            else {
                receivedChars[poziceZnaku] = '\0';
                recvInProgress = false;
                poziceZnaku = 0;
            }
        }
        // pokud zrovna parsovaný znak je "<", tak zahájíme process parsování
        else if (inputChar == startMarker) {
            recvInProgress = true;
            // jakmile přijdou nová data je potřeba je znovu zpracovat až je
            // načteme
            dataRefreshRequested = true;
        }
    }

    if (dataRefreshRequested == true) {
        parseData(receivedChars);
        dataRefreshRequested = false;
    }
}

// převede pole charu na jednotlivé hodnoty, tak docílíme toho, že nám
například zůstanou vcelku desetinná čísla, čísla od sebe oddělujeme čárkou
", "
void parseData(char receivedChars[]) { // rozdeleni dat na casti

```

```
char *ptr;

Serial.print("Test string: ");
Serial.println(receivedChars);
ptr = strtok (receivedChars, ",");
int i = 0;
while (ptr != NULL)
{
    prijataData[i] = ptr;
    i++;
    ptr = strtok (NULL, ",");
}

Serial.println(receivedChars);
Serial.println(prijataData[3]);
}
```

[52] [53]

## Příloha č.2: Kód pro Arduino UNO

```
#include <DHT.h>;
#include <OneWire.h>
#include <DallasTemperature.h>

#define DPIN 2
#define TYP DHT22
DHT dht(DPIN, TYP);
#define vzd_trig 4
#define vzd_echo 3
int pohyb = 6;
int voda_t = 9;
int kroky = 7;
int smer = 8;
OneWire oneWire(voda_t);
DallasTemperature sensors(&oneWire);

const byte numChars = 32;
long doba;
int vzdalenost;
float vlhkost;
float teplota;
float vyska;
float svetlo;
float rychlost;
float voda;
float poh;
int podminka;

String prijataData[128] = {};

void setup() {
    Serial.begin(115200);
    dht.begin();

    pinMode(vzd_trig, OUTPUT); // vzdalenost - trig
    pinMode(vzd_echo, INPUT); // vzdalenost - echo
    pinMode(5, INPUT_PULLUP);
    pinMode(6, INPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    sensors.begin();
}

void loop() {

    teplota_vlhkost();
    rozdil();
    detekce_pohybu();
    fotoresistor();
```

```

    voda_teploata();
    Serial.write("<");
    Serial.print(teplota);
    Serial.write(",");
    Serial.print(vlhkost);
    Serial.write(",");
    Serial.print(vyska);
    Serial.write(",");
    Serial.print(svetlo);
    Serial.write(",");
    Serial.print(rychlost);
    Serial.write(",");
    Serial.print(voda);
    Serial.write(",");
    Serial.print(poh);
    Serial.write(">");

    motor();

}
void teploata_vlhkost() {

    vlhkost = dht.readHumidity();
    teploata= dht.readTemperature();

}

void detekce_pohybu() {
    int status;
    status = digitalRead(pohyb);
    if (status == HIGH) {
        poh = 1;
    }
    else{
        poh = 0;
    }
}

void fotoresistor() {
    svetlo = analogRead(A0);

}

void voda_teploata() {
    sensors.requestTemperatures();
    voda = sensors.getTempCByIndex(0);
}
void rozdil() {
//predpokladana vyska ulozeni senzoru 10cm

digitalWrite(vzd_trig, LOW);

```

```

delayMicroseconds(2);
digitalWrite(vzd_trig, HIGH);
delayMicroseconds(10);
digitalWrite(vzd_trig, LOW);
doba = pulseIn(vzd_echo, HIGH);
vzdalenost = doba * 10 * 0.034 / 2;
vyska = 100 - vzdalenost;
}

char receivedChars[numChars];

//=====
void initCommunicationProcess() {
    processRawData();
}

// Funkce která zajišťuje komunikaci přes sériovou linku a data v syrovém
formát eg.<1,2,55,78> uloží do proměnné receivedChars (pole znaků) jako
"1,2,55,78"
void processRawData() {
    static boolean recvInProgress = false;
    char receivedChars[32] = {};
    static byte poziceZnaku = 0;
    char startMarker = '(';
    char endMarker = ')';
    char inputChar = (char) 0;
    char tempChars[numChars];
    boolean dataRefreshRequested = true;

    // zjistí jestli je seriová linka dostupná
    while (Serial.available() > 0) {
        inputChar = Serial.read();

        // pokud jsme uprostřed přenosu dat, pokračujeme k parsování
        if (recvInProgress == true) {
            // pokud se znak nerovná ukončovacímu znaku ">" pokračujeme dál
            if (inputChar != endMarker) {
                // uložíme aktuální znak co přišel ze sériové linky do pole znaků
                na konkrétní pozici
                receivedChars[poziceZnaku] = inputChar;
                poziceZnaku++;
                //
                if (poziceZnaku >= numChars) {
                    poziceZnaku = numChars - 1;
                }
            }
            else {
                receivedChars[poziceZnaku] = '\0'; // terminate the string
                recvInProgress = false;
                poziceZnaku = 0;
            }
        }
        // pokud zrovna parsovaný znak je "<", tak zahájíme process parsování
        else if (inputChar == startMarker) {
            recvInProgress = true;
            // jakmile přijdou nová data je potřeba je znovu zpracovat až je
            načteme
            dataRefreshRequested = true;
        }
    }
}

```

```

    if (dataRefreshRequested == true) {
        parseData(receivedChars);
        dataRefreshRequested = false;
    }

    //return receivedChars;
}

//=====

// převede pole charu na jednotlivé hodnoty, tak docílíme toho, že nám
// například zůstanou vcelku desetinná čísla, čísla od sebe oddělujeme čárkou
//,"
void parseData(char receivedChars[]) { // split the data into its
    parts
    char *ptr;

    ptr = strtok (receivedChars, ","); // Look for commas...
    int i = 0;
    while (ptr != NULL)
    {
        prijataData[i] = ptr;
        i++;
        ptr = strtok (NULL, ","); // Continue to look...
    }

}

void motor() {

    int y;
    char c;
    int smer;
    int kroky;
    float r;
    initCommunicationProcess();

    y = prijataData[0].toInt();
    podminka = y;
    r = y * 83,3;
    rychlost = 1,7 / (200 * y);

    // 200y je cyklus, 1 cyklus je 1,7ml, y je 1,7/200
    if (podminka > 0){
        digitalWrite(smer, HIGH);
        for (int x=0; x < 1000; x++){
            digitalWrite(kroky, HIGH);
            delayMicroseconds(r);
            digitalWrite(kroky, LOW);
            delayMicroseconds(r);
        }
    }
}

```

}

[53]