

**UNIVERZITA PALACKÉHO
V OLOMOUCI**

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Tomáš Vítek

Bakalářská práce

Návrh a konstrukce vzdáleného experimentu – Inteligentní skleník

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a použil jen uvedenou literaturu a zdroje.

V Hanušovicích dne 12. 4. 2015

.....

Tomáš Vitek

Poděkování:

Děkuji PhDr. PaedDr. Jiřímu Dostálovi, Ph.D., za odborné vedení bakalářské práce, poskytování rad a materiálových podkladů k práci.

Obsah

1 Úvod	7
Teoretická část	9
2 Experiment	10
2.1 Vymezení pojmu experiment	10
2.2 Rozdělení experimentů podle způsobu provádění	10
3 Vzdáleně ovládané experimenty	13
3.1 Výhody a nevýhody vzdáleně ovládaných experimentů.....	13
4 Vybrané příklady vzdáleně ovládaných laboratoří v České republice	15
4.1 Řízení výšky vodní hladiny.....	15
4.2 Určení tíhového zrychlení z doby kmitu matematického kyvadla.....	16
4.3 Meteorologická stanice	16
4.4 Monitorování radioaktivního pozadí.....	17
4.5 Regulace otáček cívky.....	18
5 Konstrukční požadavky na vzdálený experiment – Inteligentní skleník	19
5.1 Větrání.....	19
5.2 Topení	19
5.3 Závlaha	20
5.4 Osvětlení	21
5.5 Stínění	21
6 Automatizační systém	22
6.1 Arduino	22
6.1.1 Charakterizace systému Arduino	22
6.1.2 Arduino Mega2560	23
7 Konstrukce – volba vhodných čidel ovládacích mechanismů	25
7.1 Senzor světla	25
7.2 Průtokoměr.....	25
7.3 Senzor vlhkosti půdy.....	27
7.4 Senzor teploty a vlhkosti vzduchu	28

7.5	LCD displej	29
7.6	LCD seriálový převodník	30
7.7	Snímač vodní hladiny	30
7.8	Relé	31
7.9	Vibrační senzor	32
7.10	Krokový motorek	33
7.10.1	Výhody a nevýhody krokového motorku	35
7.10.2	Ovladač krokového motorku	38
7.11	Čerpadlo SEAFLO 350	38
7.12	Ethernet shield	39
7.13	Modul tlačítek	40
Praktická část		41
8	Volba zapojení	42
8.1	Senzor světla TEMT 6000	42
8.2	Průtokoměr	43
8.3	Vlhkost půdy	46
8.4	Senzor teploty a vlhkosti vzduchu	47
8.5	Relé	49
8.6	Relé a tlačítko	50
8.7	Krokový motorek	51
8.8	LCD displej	53
8.9	Ethernet	54
9	Propojení vstupů a výstupů	57
9.1	Ovládání větrání	57
9.2	Ovládání topení	59
9.3	Ovládání čerpadla	60
9.4	Ovládání světla	62
9.5	Ovládání rolet	63
9.6	Ovládání oken	66
10	Schéma vzdáleného experimentu	69
11	Vizualizace vzdáleného experimentu	70
12	Závěr	72

13	Zdroje	73
14	Seznam obrázků.....	75
15	Seznam příloh	77
16	Seznam zkratek.....	78

1 Úvod

Vývoj informačních technologií pociťuje každý z nás. Někdo více, někdo méně. Největší posun využívání informačních technologií v běžném životě je u dětí ve školách a dospívajících studentů. Nejlepší způsob, jak zaujmout žáky ve vyučování, je podřídit se jejich trendům. Tuto podmínku splňuje vzdálený experiment, ve kterém jsou využívány informační technologie. Žáci sledují online probíhající pokus vzdáleného experimentu a mohou do něho zasahovat. Zároveň se žáci seznamují s novým prostředím. I přes pozitiva vzdáleného experimentu není tento způsob výuky na školách příliš rozšířený.

Už v dobách knížete Aloise II. byl postaven první moderní skleník, ve své době z materiálů jako byla ocel, litina a sklo. Důvod pěstování ve skleníku byl ten, že každá rostlina má svůj geografický původ na jiném místě planety a má svoje vlastní maximální a minimální teploty pro svůj růst, v závislosti na denním a nočním světle. Protože lidé chtěli a chtějí pěstovat rostliny i jinde, než na původním geografickém místě původu, museli vytvořit tyto podmínky potřebné pro správný růst uměle. Za tímto účelem pěstování byl zkonstruován výrobek s názvem skleník.

Skleník je stavba, kterou tvoří nosná konstrukce osazená průhlednými deskami, upevněnými v okenních rámech z důvodu možností otevírání oken. Ve skleníku se ohřívá vzduch působením slunečního záření, které prochází pláštěm skleníku dovnitř, ale zároveň minimalizuje tepelné ztráty. Tento způsob získávání tepla se nazývá skleníkový efekt. Skleník nikdy nevytvoří původní klimatické podmínky pro rostlinu, ale pro možnost pěstování cizokrajných rostlin v ČR tyto podmínky podstatně zlepši kvalitu prostředí, kterou novodobé inteligentní skleníky umí dokonale vytvořit.

V dnešní době se automatické systémy již stávají nedílnou součástí každé moderní zahrady, okrasných parků, veřejných ploch a sportovišť. Není také divu, vždyť inteligentní systém ve skleníku pracuje zcela samostatně, čímž nám poskytne značnou úsporu času. Výhody pořízení automatické závlahy spočívají nejenom v úspoře našeho času, ale také v menším množství vody, potřebné ke kvalitní závlaze a spotřebě elektrické energie.

Cílem této bakalářské práce bude analýza teoretických vstupů vzdáleně ovládaných experimentů. Dále charakterizovat základní princip vzdáleně ovládaného

experimentu, vysvětlit jeho funkčnost, popsat jednotlivé prvky experimentu a postup při jeho tvorbě.

Dalším cílem bakalářské práce je navrhnout a zkonstruovat vzdáleně ovládaný experiment, který může sloužit jako pomůcka při výuce technické výchovy, hlavně na středních školách. Jedná se o inteligentní skleník, který lze ovládat pomocí internetu. Pro realizaci experimentu byl použit programovatelný čip Arduino. Pro svou nízkou pořizovací cenu a jednoduchost je Arduino ideální pomůckou při výuce technických předmětů na středních školách. Modul se vyznačuje svou praktičností a univerzálností. Arduino využívá programovací jazyk C a ke svému modulu dokáže připojit komunikační moduly Bluetooth, Ethernet či Wi-fi. Díky možnosti komunikace s běžným počítačem pomocí USB konektoru je Arduino vhodný k realizaci počítačem vzdáleně ovládaných experimentů.

Teoretická část

2 Experiment

2.1 Vymezení pojmu experiment

Podle Vybírala a Mechlové „se jedná o vědeckou metodu, při níž se uměle (tj. za účasti výzkumníka) navodí děj s předem stanovenými podmínkami tak, aby jej bylo možné za stejných podmínek opakovat. Při probíhajícím experimentu objektivně sledujeme měřením pomocí přístrojů vzájemnou závislost fyzikálních veličin za působení co nejmenšího počtu rušivých vlivů. Experiment je prostředkem poznání a současně specifickou formou praxe. Získané výsledky zobecňujeme do formy fyzikálního zákona (obvykle popsaného matematickým modelem), přičemž jeho správnost ověřujeme dalšími pokusy.“ (Vybíral, 2006; Mechlová, 1999)

Vzdělávání prochází kontinuálními změnami, proto je experimentování stále populárnější metoda ve školském prostředí. „Velmi významná je také potřeba zvýšení zájmu žáků o přírodovědné a technické obory, což vyplývá nejen z výzkumné sondy zadané Ministerstvem školství, mládeže a tělovýchovy ČR, ale i z prací řady autorů.“ (Dostál, 20013, s. 9).

Experiment slouží nejenom jako pomůcka učitele pro pozornost žáků, ale také jako prostředník mezi žáky a učivem. „Zařazení experimentu do výuky umožňuje žákům seznámit se se základními praktickými postupy a metodami práce v příslušné oblasti lidského konání a slouží jako prostředek k získávání nebo ověřování teoretických znalostí žáka. Díky tomu, že je zkušenost získávána přímo, umožňuje trvalé a důkladné osvojení objevených poznatků.“ (Dostál, 2013, s. 11)

2.2 Rozdělení experimentů podle způsobu provádění

Podle způsobu provádění se vzdálený experiment rozděluje do tří základních skupin:

1. Klasické:

Jedná se o experimenty prováděné přímo ve školních laboratořích. Žáci mají přímý kontakt s vybavením laboratoře. Při nesprávném postupu měření může učitel ihned žáky opravovat, přičemž tento způsob nabízí zpětnou vazbu mezi učitelem a žáky.

V zahraničí se tato metoda nazývá „Hände in experiment“, což můžeme volně přeložit jako laboratorní experiment.

2. Virtuální experimenty:

Ve virtuálních experimentech nepracují studenti či uživatelé s reálnými pomůckami, ale ovládají jen virtuální objekty.

„Virtuální experimenty lze využívat zejména při interpretaci reálných experimentů, ve školních podmínkách neproveditelných. Získávání a prohlubování manuálních dovedností, které jsou jednou z podstatných složek přírodovědného vzdělávání nelze nahradit prací s monitorem a klávesnicí. Ovšem vyhýbat se zprostředkovanému pozorování a práci s modely možné také není. V této oblasti nás čeká ještě řada výzkumů, jak potvrzujících tak nově orientovaných, které budou s to zodpovědět na otázky spojené s rozdílem a přínosem reálné a simulované experimentální činnosti.“ (BÍLEK, M., SKALICKÁ, P., RYCHTERA, J., MYŠKA, 2009, str. 3)

3. Vzdálené experimenty:

V našem kontextu se bude jednat o vzdálené experimenty, využitelné ke vzdělávacím účelům. Vzdálené experimenty se nachází v reálné laboratoři, které jsou dálkově ovládány z jakéhokoliv místa na světě pomocí internetu. Proto tento typ experimentu nazýváme „vzdálený experiment“.

Hellberg a Bílek uvádějí: „Vzdálená laboratoř představuje v edukativním pojetí otevřenou, vzdáleně přístupnou databázi objektů využitelných pro experimentální činnost studentů a učitelů“. Dále uvádí: „ve většině případů se jedná o zpřístupnění průběžně snímaných dat (např. meteorologické družice, seismografy, hmotnostní spektrografy, výkonné spektrální přístroje aj.), zřídka může vzdálený uživatel i ovlivňovat uspořádání měřicího systému a snímání dat podle vlastních potřeb.“
(HELLBERG, BÍLEK, 2000, 143 s.)

V dnešní době není pochopitelně možné experimentovat stejně jako před 15 lety. Experimenty musí vycházet z každodenní praxe současných studentů. Studenti si místo poznámek do sešitů dělají poznámky do notebooků, tabletů či chytrých telefonů. Mladí studenti využívají informační technologie do posledního detailu. První knížky čtou

pomocí elektronických čteček. Volný čas tráví na internetu a sociálních sítích, kde mohou ihned nalézt potřebné informace k danému problému.

3 Vzdáleně ovládané experimenty

Vzdálený experiment je experiment, který probíhá v reálném čase a je používán ve skutečné laboratoři. Může ho ovládat žák ve škole, učitel nebo jakýkoliv uživatel z počítače, s přístupem k internetu kdekoliv na světě. Uživatel nepracuje s experimentem na místě, kde byl zrealizován, ale ovládá ho pomocí webového prohlížeče. Experiment většinou obsahuje i webovou kameru, která umožňuje sledování experimentu v reálném čase. Uživatel musí mít připojení k internetu a prohlížeč s podporou Java od firmy Oracle.

Schéma vzdáleného experimentu:



Obrázek 1: Schéma vzdáleného experimentu¹

Použití vzdáleného experimentu je nejvíce užitečné tam, kde není dostupné potřebné vybavení k uskutečnění experimentu, nebo kde nelze experiment provést na živo.

3.1 Výhody a nevýhody vzdáleně ovládaných experimentů

Hlavní výhody vzdáleně řízených experimentů jsou:

- Uživatel může provádět experiment z libovolného místa na světě v kterémkoliv čase
- Vzdálený experiment je sestaven a ihned připraven k použití
- Oproti virtuálním simulacím pracuje uživatel se skutečnými pomůckami a nástroji
- Uživatel může při měření postupovat svým vlastním tempem. Pokud se mu měření nepovede, má možnost opakování

¹ Obrázek notebooku převzat z <http://blogbringit.com.br/>, obrázek serveru převzat z <http://blogbringit.com.br/>, obrázek experimentu převzat z <http://www.sklenik-expert.cz/>, obrázek kamery převzat z www.com-shop.me, propojení obrázků: vlastní návrh.

Podle Gröbra jsou vzdálené experimenty využívány i ve vědeckých výzkumech:

„Vzdáleně ovládané laboratoře se používají již řadu let v oblasti vědeckého výzkumu (např. v urychlovači částic), při studiu vesmírného prostoru prostřednictvím kosmických sond a teleskopů, ve vývoji nových technologií a v oblasti robotiky.“ (Gröber, 2008)

Výhody vzdáleného experimentu podle Auera:

Využití vzdáleně ovládaných experimentů ve vědecké a průmyslové sféře má několik základních výhod (Auer, 2001):

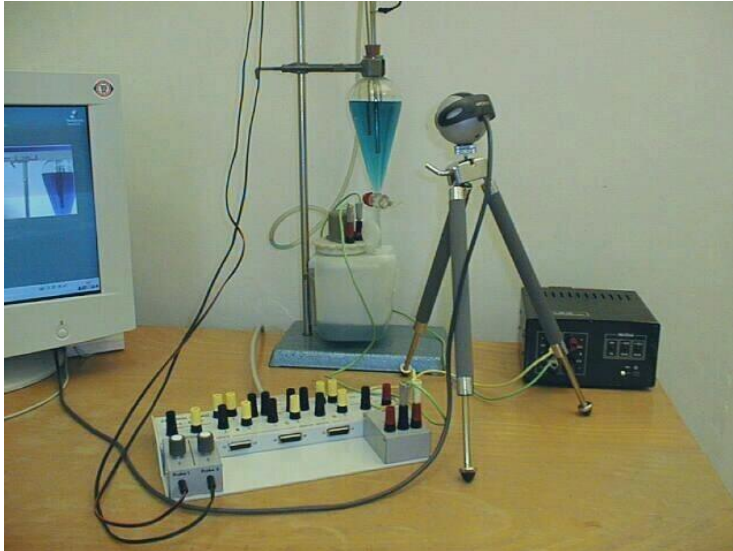
- *nákladné a složité přístroje mohou být používány z různých míst dané společnosti*
- *členové týmu, pracující v různých místech světa, mohou provádět stejná měření bez nutnosti cestování*
- *dlouhodobé měření může být jednoduše hlídáno (obsluhováno) z domova např. o víkendu*

Hlavní nevýhody vzdáleně řízených experimentů jsou:

- Nepřímý kontakt s experimentem
- Při velkém zájmu o experiment musí další zájemci čekat, než se experiment uvolní
- Pokud vznikne porucha při měření, vzdálený uživatel není schopen tento problém opravit a je vázán pouze na tvůrce
- Vždy je potřeba mít jeden počítač s připojením na internet
- Vzdálený experiment musí mít opatření proti poškození či zničení experimentu vzdáleným uživatelem
- Experiment je nutné zkonstruovat tak, aby se vždy po ukončení práce na modelu vrátil model do původního stavu a byl připraven pro další měření
- Tvůrce experimentu musí svůj vzdálený experiment kontrolovat a případné problémy ihned opravit

4 Vybrané příklady vzdáleně ovládaných laboratoří v České republice

4.1 Řízení výšky vodní hladiny

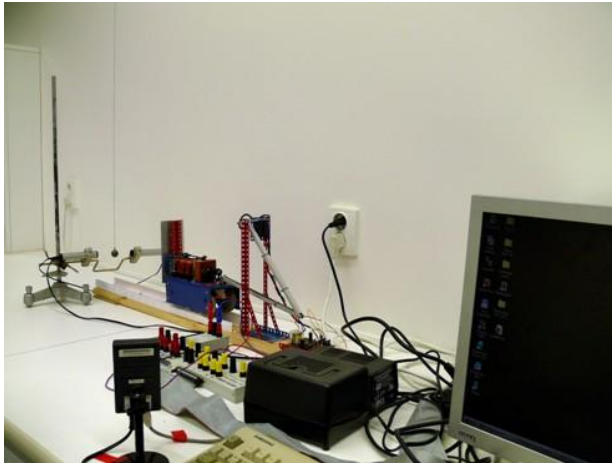


Obrázek 2: Vzdálený experiment - Řízení výšky vodní hladiny

(Zdroj: <http://kdt-34.karlov.mff.cuni.cz/cz/intro.html>)

Řízení výšky vodní hladiny je vzdálený experiment Univerzity Karlovy v Praze. V experimentu jsou použity dvě sondy pro snímání výšky vodní hladiny a čerpadlo. Výšku vodní hladiny snímá kovová sonda, která mění svou hodnotu vodivosti při dotyku s hladinou. Z důvodu lepší viditelnosti je voda obarvena modrým inkoustem. Řízení probíhá pomocí dvou tlačítek: zapnutí/vypnutí čerpadla a zapnutí/vypnutí kontroly přetečení vody.

4.2 Určení tíhového zrychlení z doby kmitu matematického kyvadla



Obrázek 3: Vzdálený experiment - Tíhové zrychlení kmitu matematického kyvadla

(Zdroj:<http://ictphysics.upol.cz/remotelab/rlab2.html>)

Tato vzdálená laboratoř se nachází v Univerzitě Palackého v Olomouci. Experiment obsahuje matematické kyvadlo, které je možné nastavit do 8. různých stupňů vychýlení. Po spuštění matematického kyvadla snímá optická závora průchod kuličky optickou závorou a zaznamenává jej do tabulky. Naměřená data lze vygenerovat jako tabulku pro Excel a dále s nimi pracovat.

4.3 Meteorologická stanice



Obrázek 4: Vzdálený experiment - Meteorologická stanice

(Zdroj: <http://ictphysics.upol.cz/remotelab/rlab4.html>)

Meteorologická stanice je vzdálený experiment, který snímá teplotu vzduchu, vlhkost vzduchu, atmosférický tlak a sluneční svit. Naměřená data se zobrazují on-line ve webovém prohlížeči. Po zobrazení se ukládají na disk pro pozdější možnost zpětného

zjištění. Naměřená data lze zobrazit jak číselně, tak i graficky. K přístupu do laboratoře nejsou potřebné žádné přihlašovací údaje.

4.4 Monitorování radioaktivního pozadí



Obrázek 5: Vzdálený experiment - Monitorování radioaktivního pozadí

(Zdroj: http://ictphysics.upol.cz/remotelab/rlab5_foto.html)

„Radioaktivitou rozumíme schopnost některých atomových jader vysílat záření. Při tom se takové jádro může přeměnit v jiné nebo alespoň ztratí část své energie. Při jaderné přeměně se mění struktura jádra, jeden nuklid se mění v jiný.“ (Látal, 2011)

Existují 3 základní typy záření:

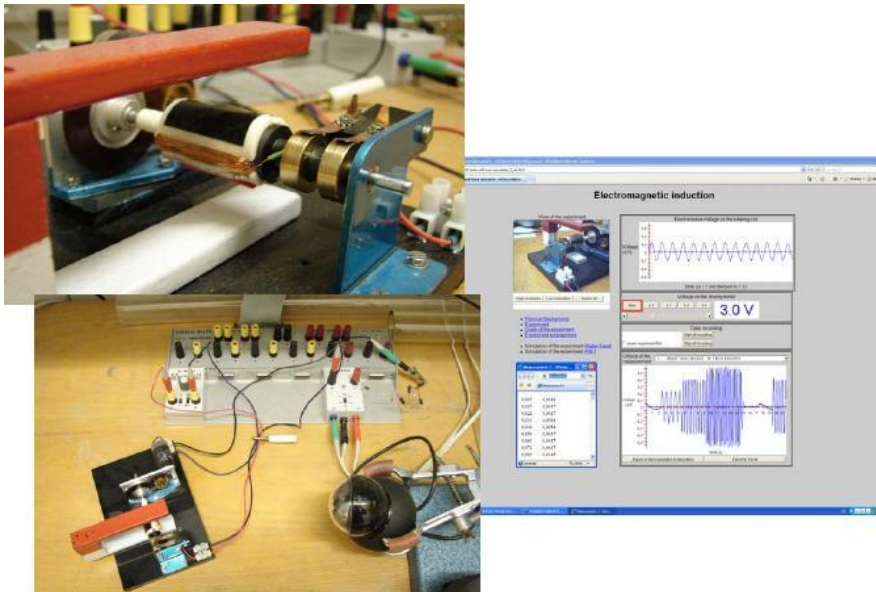
Záření alfa: Pohlcuje jej vzduch po uběhnutí několika centimetrů nebo jej pohltí list papíru.

Záření beta: Pohlcuje jej tenký hliníkový plech.

Záření gama: Je nejpronikavější, pohlcuje jej silná vrstva olova.

Hlavním základním kamenem této vzdálené laboratoře je Geiger-Müllerův přístroj, který měří hodnotu gama záření v ovzduší. V experimentu se zaznamenává počet zachycených částic, které dopadly na senzor v intervalech deseti sekund, třech minut, tří hodin a tří dnů. V grafu se každou minutu zaznamenávají hodnoty počtu částic, které dopadly na senzor. Naměřená data se ukládají na disk. Naměřené číselné hodnoty je možné vygenerovat pro další použití.

4.5 Regulace otáček cívky



Obrázek 6: Vydálený experiment - Regulace otáček cívky

(Zdroj: <http://eedu.eu/exp/0/>)

Elektromagnetická indukce

„Elektromagnetická indukce je jev vzájemné vazby magnetického a elektrického pole, při kterém vzniká elektrické pole vždy, když dochází k časové změně magnetického pole. Například v pokusu na obr. 1 se pohybuje tyčový magnet ve směru závitu, čímž se v místě závitu vytváří časově proměnné magnetické pole. Na připojeném měřicím přístroji pozorujeme výchylku, která je způsobena časově proměnným elektrickým polem v závitu. Tento jev se využívá např. v dynamech nebo alternátorech, což jsou zařízení na přeměnu mechanické energie na energii elektrickou.“ (Lustig)

Experiment se skládá ze dvou cívek různých vinutí, elektromotorku a voltmetru. Tyto přístroje jsou připojeny k řídicí jednotce (Pc). Pro přenos dat je potřebné připojení k internetu. Ve vzdálené laboratoři můžeme pomocí nastavování napětí v rozsahu 0-10V regulovat rychlost otáček cívky. Můžeme také měnit počet závitů cívky a to buď 18, nebo 36 závitů. To nám způsobí nižší nebo vyšší napětí na výstupu. Po spuštění experimentu se zobrazí výstupní napětí U v milivoltech. Graf indukovaného napětí je možné zaznamenávat, ukládat a exportovat jeho hodnoty pro další zpracování.

5 Konstrukční požadavky na vzdálený experiment – Inteligentní skleník

5.1 Větrání

V současné době, kdy skleníky mají velký koeficient propustnosti a nízké tepelné ztráty je nutné zajistit intenzivní větrání a to v první řadě větrání střešními okny. U menších domácích skleníků jsou použity doplňující ventilátory, umístěné v čelech skleníku, které slouží pro lepší proudění a usměrnění vzduchu. U velkých skleníků se používají hlavně střešní okna.

Nejlepší způsob větrání je přirozený, kde chladnější vzduch se drží u země a díky proudění odvádí přebytečnou vlhkost od rostlin, zároveň ale nevytváří průvan, který by mohl rostliny poškodit. Tento přirozený způsob pracuje na principu hustoty teplého a studeného vzduchu. Teplý vzduch stoupá od země vzhůru a je odveden ventilací. Na jeho místo se tlačí studený vzduch, který stoupá od kořenů. Výkon větrání je určen rozdílem vnějšího a vnitřního tlaku vzduchu.

5.2 Topení

Mezi základní prvky automatického skleníku patří regulace teploty. Pro časté přehřívání skleníku z důvodu skleníkového efektu v letním období a také pro zajištění přísunu oxidu uhličitého, který je důležitý pro růst rostlin.

Pokud chceme pěstovat rostliny ve skleníku celoročně, musíme mít skleník vybaven topným systémem. V menších sklenících jsou vhodná lokální topidla. Ve větších sklenících se používá propan-butanové topení, které má vestavěný ventilátor pro lepší tepelnou výměnu. V komerčních sklenících se používá více těchto jednotek pro zajištění rovnoměrného vytápění. Současné skleníky mají malé tepelné ztráty, což je pro vytápění výhodné. Malá tepelná ztráta ale zabraňuje přísunu CO₂ do skleníku, který je velmi potřebný pro rostliny. Instalováním plynového topení nejenom získáme teplo ale spalováním i oxid uhličitý.

5.3 Závlaha

Společnost Grundfos s.r.o. uvádí:

„Dodávání vody zavlažováním se provádí již po tisíce let. Závlaha zvýší jak kvalitu, tak i množství úrody. A to dokonce i v oblastech, kde je zemědělství závislé na srážkách. Při využití pro účely rekreace může závlaha zajistit svěží a atraktivní vzhled sportovních zatravněných ploch.“

Pro zvolení níže vybraných typů závlahy je nejdříve nutné zjistit základní parametry. Mezi tyto základní parametry podle společnosti Grundfos s.r.o. patří:

- *Druh plodin, které mají být pěstovány*
- *Klimatické podmínky*
- *Množství potřebné vody*
- *Přístupnost zdroje vody*
- *Členitost zavlažované plochy (rovinatá nebo hornatá)*
- *Druh půdy (jílovitá nebo písčítá)*
- *Počet měsíců v roce, kdy je nutná závlaha*

V modelu je použita kapková závlaha. Kapková závlaha je velmi účinný typ závlahy, který se využívá zpravidla u keřovitých kultur, hustých přizemních porostů, u kultur, kde je nebezpečí vzniku plísní při závlaze na list a podobně. Pro tuto závlahu se používají kapkovací hadice, které mají uvnitř potrubí rozmístěné kapkovače ve stanovených vzdálenostech od sebe (tzv. spon). Podle potřeby volíme kapkovací hadice s daným sponem tak, aby kapkovače byly co nejbližší u rostlin. Výkon jednoho kapkovače pro účely závlah na zahradách se pohybuje od 2 do 3 l/hod. Kapková závlaha zavlažuje pod nižším pracovním tlakem a vyžaduje delší dobu zavlažování (v řádech desítek minut), proto pro ni používáme samostatné čerpadlo.

5.4 Osvětlení

Existují různé důvody, proč pěstovat rostliny pod umělým osvětlením. Zpravidla je používají komerční velkopěstírny, které pěstují rostliny na malém prostoru v co nejkratší době. Rostlina musí mít zajištěny všechny podmínky pro svůj dokonalý růst. Tyto podmínky představují světlo a živiny. Mezi nejvhodnější svítidla pro pěstování rostlin pod umělým světlem, patří výbojky nebo úsporné zářivky. Bílé zářivky se používají při denním světle. Výbojky jsou nejdokonalejší, mají velkou výkonnost a účinnost. Pro zvýšení účinnosti umělého osvětlení se používají parabolické reflektory, které soustřeďují světlo pouze na záhon. Ideální intenzita osvětlení, soustředěná přes reflektory, je 60W na metr čtvereční. Nejvhodnější doba osvětlení rostlin je 12 hodin světlo a 12 hodin tma. Tento cyklus je pouze doporučený a je možné ho podle vlastních zkušeností upravit. Existují také baktericidní výbojky, které vytvářejí ultrafialové světlo. Toto světlo ničí bakterie a vytváří sterilní prostředí. Ultrafialové záření rostlinám při růstu škodí, proto se používá jen v krátkých intervalech.

5.5 Stínění

Jedním ze základních parametrů pro růst rostlin je teplo. To zajišťuje světelné záření, které proniká přes sklo do skleníku. Pokud chceme pěstovat rostliny i v zimě, musíme použít vytápění. Naopak v létě je sluneční záření tak silné, že může rostliny poškodit. Vhodným řešením, jak poškození rostlin předejít, jsou stínící rolety.

Automatické stínění skleníků dělíme na vnitřní a vnější. Vnitřní stínění se realizují pomocí clon, které umožňují regulaci stínění podle typu materiálu od 15 do 90 %. Vnější clony mají větší zatemňovací účinek než ty vnitřní. Tyto clony neslouží jen ke stínění, ale také jako ochrana skleníku před kroupami.

6 Automatizační systém

6.1 Arduino

Arduino založili Massimo Banzi a David Cuartielles v Itálii v roce 2005. Jejich cílem bylo vytvořit elektronickou platformu na uživatelsky jednoduchém hardware a software pro studenty či domácí kutily. K této platformě můžeme připojit vstupy, to jsou čidla různých typů, které mohou vnímat okolí. Dále můžeme připojit výstupy, které mohou toto okolí i řídit pomocí LED světla, krokových motorků, čerpadel či jiných fyzických výstupů.

Arduino má svůj vlastní programovací jazyk. Je založený na jazyku Wiring, který je velmi podobný C. Projekt Arduino je volně dostupný všem uživatelům, kteří jej chtějí využívat.

6.1.1 Charakterizace systému Arduino

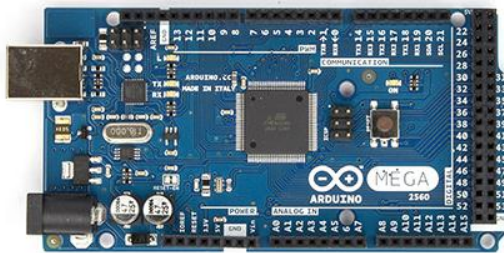
Pro práci s počítačem existuje spousta mikroovladačů a platforem. Parallax Basic Stamp, BX-24 od Netmedia, Phidgets, Handyboard od MIT a mnoho dalších, které nabízejí obdobnou funkčnost jako Arduino. Všechny tyto platformy se prezentují jako přehledné, ale ve skutečnosti to jsou jen balíčky, ve kterých jsou nepřehledné drobnosti. Oproti ostatním ovladačům Arduino také zjednodušuje práci s mikroovladači, ale oproti ostatním platformám poskytuje různé výhody pro uživatele.

Výhody Arduina:

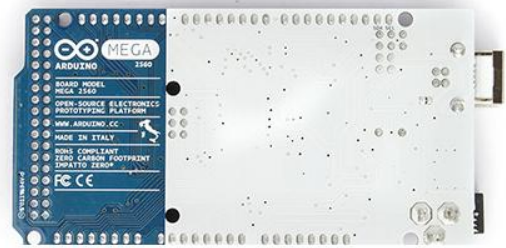
1. **Nízké náklady** – Desky Arduino jsou oproti ostatním prodávány levněji. Nejlevnější verze modulu Arduino je možné sestavit ručně.
2. **Multiplatformnost** – Software Arduino je přizpůsoben pro operační systémy Windows, Linux a Macintosh. Ostatní ovladače jsou omezeny pouze na Windows.
3. **Jednoduché programovací prostředí** – Programovací prostředí Arduino je vhodné jak pro začátečníky, tak pro pokročilé uživatele. Arduino je založeno na programovacím prostředí Processing, takže začínající programátoři nemají problém se s programovacím prostředím Arduina seznámit.

4. **Open source a rozšiřitelný software** – Software Arduino je vydáván jako open source nástroj. To znamená, že software Arduina má otevřený zdrojový kód, který jej může zkušený uživatel libovolně upravovat dle vlastní potřeby. Návrhy jsou licencovány společností Creative Commons. Takže zkušení programátoři si mohou vytvořit vlastní verzi. Jazyk může být rozšířen přes knihovny C++.

6.1.2 Arduino Mega2560



Pohled zepředu



Pohled zezadu

Obrázek 7: Arduino Mega2560

(Zdroj: <http://arduino.cc/en/Main/arduinoBoardMega2560>)

Arudino mega 2560 je deska založená na mikroprocesoru Atmega2560. Má 54 digitálních vstupních/výstupních pinů. Z toho může být 15 pinu použito jako PWM. PWM znamená nastavení pinu jako analogový výstup. Dále obsahuje 16 analogových vstupů, 4 UARTs (sériové porty), 16 MHz oscilátor, připojení na USB, který slouží jako napájení a zároveň jako komunikace s Pc. Dále obsahuje tlačítko reset. Arduino mega je kompatibilní s přídatnými štíty (shields). Jedná se o desky, které nasuneme přímo na piny arduina, přičemž nám piny zůstanou stále dostupné.

Napájecí piny:

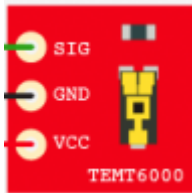
- Vin – pin používaný při použití externího napájení. Nejčastěji se jedná o napájení 9V (liší se v závislosti na typu desky – tento údaj je možný zjistit z příslušné dokumentace k desce)
- GND – zem obvodu
- 5V – regulovaný napájecí zdroj sloužící k napájení mikrokontroléru a dalších

komponent na desce.

- 3V3 – regulovaný výstup, který je možno použít pro napájení externího obvodu
- Reset – přivedením signálu resetujeme desku

7 Konstrukce – volba vhodných čidel ovládacích mechanismů

7.1 Senzor světla



Obrázek 8: Senzor světla TEMT 6000

(Zdroj: <http://www.hwkitchen.com/products/temt6000-breakout-board/>)

Senzor světla TEMT 6000 obsahuje křemíkový NPN fototranzistor v miniaturní formě pro povrchové upevnění na desce plošného spoje. Zařízení je citlivé ve viditelném spektru.

Vlastnosti:

- Senzor je přizpůsoben citlivosti lidského oka
- Široký uhel snímání $\varphi = \pm 60^\circ$

7.2 Průtokoměr

Průtokoměr je zařízení, které měří průtok kapaliny. Dokáže také změřit množství kapaliny určenou pro zálivku, která je důležitá pro růst rostlin.



Obrázek 9: Průtokoměr YF - S201

(Zdroj: <http://www.hwkitchen.com/products/g1-2-water-flow-sensor/>)

Průtokoměry můžeme rozdělit na:

(Ďad'o, Bejček, s. 30 – 50)

- 1) *Průřezové průtokoměry*
- 2) *Rychlostní průtokoměry*
- 3) *Průtokoměry s proměnným průřezem*
- 4) *Turbínové a lopatkové senzory průtoku*
- 5) *Objemové průtokoměry*
- 6) *Ultrazvukové senzory průtoku*
- 7) *Indukční senzory průtoku*
- 8) *Hmotnostní průtokoměry*
- 9) *Fluidikové průtokoměry*
- 10) *Tepelné průtokoměry*
- 11) *Průtokoměry se značením tekutiny*

Model YF – S201 patří do skupiny turbínových a lopatkových senzorů průtoku. Průtokoměr nabízí jednoduché řešení pro snímání průtoku kapaliny, v našem případě vody. Skládá se z plastového těla, rotoru a Hallovým snímačem. Když voda proudí skrz rotor, Hallův senzor rychlost proudění snímá a vydává odpovídající signál. Napojení průtokoměru provedeme na ½ coulovou hadici.

Měronosnou veličinou je frekvence impulzů f , která je úměrná objemovému průtoku q_v . V ideálním případě platí vztah: (Ďad'o, Bejček, s. 98 – 99)

$$f \text{ [hz]} = K_t * q_v \text{ [l / min]}$$

kde K_t je součinitel turbínového senzoru. V našem případě výrobce udává $K_t = 7,5$.

Ze vztahu tedy plyne:

$$q_v \text{ [l / min]} = \frac{f}{7,5}$$

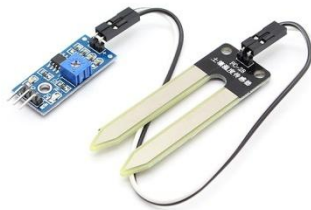
$$q_v \text{ [l / hod]} = \frac{f * 60}{7,5}$$

Výše uvedených vztahů využijeme v modelu inteligentního skleníku s průtokoměrem.

Vlastnosti:

- Pracovní napětí: 5 V
- Max. proud: 15 mA
- Hmotnost: 43 gramů
- Rozsah průtoku: 1 – 30 l / min
- Provozní teplota: 0 – 30 °C
- Provozní tlak: do 1,2 Mpa

7.3 Senzor vlhkosti půdy



Obrázek 10: Senzor vlhkosti půdy

(Zdroj: <http://www.santy.cz/moduly-c22/sensor-pudni-vlhkost-i132/>)

Senzory vlhkosti dělíme na 2 skupiny:

- Přímé – určuje se oddělené množství vody ve vzorku půdy
- Nepřímé – založená na elektrickém měření vody v půdě

Nepřímé metody dělíme na:

- Potenciometrová
- Odporová
- Kapacitní
- Dielektrická
- Tenziometrická
- Vodivostní
- Neutronová
- Gamaskopická

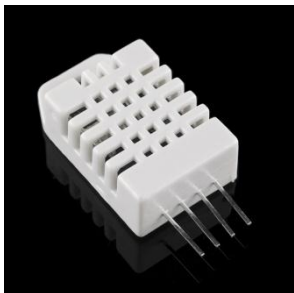
- Absorpční mikrovlnná

Modul Arduino LM393 patří do skupiny odporových snímačů vlhkosti půdy. Snímá množství vlhkosti obsažené v půdě, která jej obklopuje. Senzor je ideální na sledování zahrady nebo skleníku. Senzor používá dvě sondy k propouštění proudu zeminou, přičemž odečte odpor a tím zjistí úroveň vlhkosti. Větší vlhkost způsobí lepší vodivost elektřiny v půdě (nižší odpor), zatímco suchá půda vede elektřinu hůře (vyšší odpor).

Vlastnosti:

- Napájecí napětí: 5V
- Max. proud: 35mA
- Rozsah hodnot:
 - 0 ~ 300: ve vodě
 - 300 ~ 750: vlhká půda
 - 750 ~ 1023: suchá půda

7.4 Senzor teploty a vlhkosti vzduchu



Obrázek 11: Senzor teploty a vlhkosti vzduchu

(Zdroj: <http://www.hwkitchen.com/products/humidity-and-temperature-sensor-dht22-/>)

Senzor teploty a vlhkosti vzduchu RHT 03, známý také jako DHT-22, je nízkonákladový senzor s jedním digitálním rozhraním. Senzor je kalibrován a nevyžaduje další komponenty.

Vlastnosti:

- Nízké náklady
- Napájení: 3-5V
- Max proud: 2,5mA
- Rozsah vlhkosti: 0-100% s odchylkou 2-5%
- Rozsah teploty: -40 – 125°C s odchylkou +/- 0,5°C
- Rozměry: 15.1mm x 25mm x 7.7mm

7.5 LCD displej



Obrázek 12: LCD displej

(Zdroj: <http://www.hwkitchen.com/products/lcd-display-4x20-characters/>)

„Displej z kapalných krystalů LCD (liquid crystal display) využívá změnu propustnosti světla kapalným krystalem působením elektrického pole.“ (Reichl a Všetická, 2015)

LCD displej obsahuje 4 řádky. Jednotlivý řádek zobrazí maximálně 20 znaků. LCD displej má zelenožluté LED podsvícení obrazovky. Displej má vysoký kontrast a široký pozorovací úhel. Provozní teplota modulu je od – 20 do 70°C. Modul je řízen primárně přes paralelní rozhraní, ale je i možnost sériového převodníku.

Vlastnosti:

Hmotnost: 75 gramů

7.6 LCD seriálový převodník



Obrázek 13: Seriál LCD převodník

(Zdroj: <http://www.hwkitchen.com/products/serial-enabled-lcd-backpack/>)

Paralelně-sériový převodník umožňuje ovládání paralelního displeje prostřednictvím jednoho sériového rozhraní, které připojíme k arduinu. LCD přijímá všechny příkazy ovladače HD44780, umožňující bezproblémovou integraci s jakýmkoliv procesorem, který má možnost komunikace přes širokou škálu přenosových rychlostí. Seriál LCD v současnosti podporuje 16. a 20. znakovou šíři obrazovek se dvěma nebo čtyřmi řádky zobrazení.

7.7 Snímač vodní hladiny



Obrázek 14: Snímač vodní hladiny

(Zdroj: <http://www.hwkitchen.com/products/water-level-switch/>)

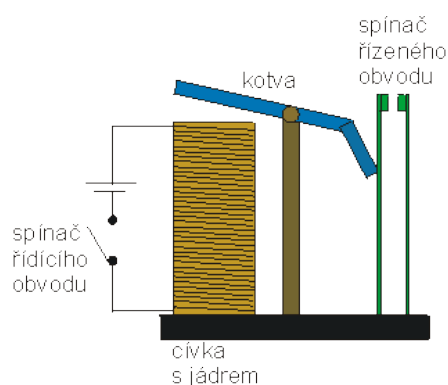
Spínač vodní hladiny je užitečná součást, používaná ke snímání hladiny kapaliny. To může ovládat indikátor, budík nebo čerpadlo. Čerpadlo nečerpá „naprázdno“ a tím nedojde k poškození zařízení.

Vlastnosti:

- Max. zatížení: 50W
- Maximální spínací napětí: 100 VDC
- Maximální spínací proud: 0.5A
- Max. konstantní odpor: 0,4Ω

7.8 Relé

Elektromagnetické relé je elektrotechnická součástka, která je potřebná ke spínání elektrického proudu. Skládá se z cívky, jádra a pohyblivé kotvy. Jakmile přivedeme ovládací napětí na elektromagnetické relé, kotva se přitáhne k cívce a sepne obvod. Ovládací proud pro spínání je velmi malý, o dost menší než proud, který prochází obvodem. Po odpojení ovládajícího napětí se kotva oddálí od cívky a kontakt se vrátí do klidového stavu.



Obrázek 15: Elektromagnetické relé

(Zdroj: http://elektross.gjn.cz/elmotor_magnet/rele.html)



Obrázek 16: Arduino relé modul

(Zdroj: <http://aukro.cz/8kanalove-rele-dc30v-10a-ac250v-arm-mcu-arduino-5v-i5188734176.html>)

8 kanálové relé Songle:

Vlastnosti:

- Obsahuje LED indikaci pro každé relé
- Maximální spínací napětí: AC250V / DC30V
- Maximální proud: 10A
- Pracovní napětí: 5V

7.9 Vibrační senzor



Obrázek 17: Vibrační senzor

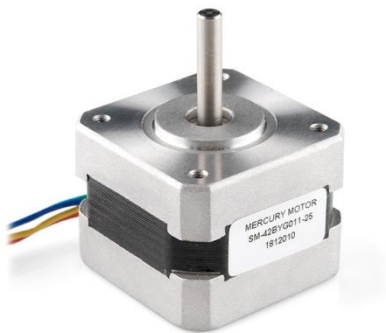
(Zdroj: <http://aukro.cz/ard3-42-arduino-vibrační-senzor-s-kabelem-i5201148520.html>)

Vibrační senzor obsahuje piezoelektrické vidličky, které pracují na daném kmitočtu v klidovém režimu. Rezonance kmitočtu se utlumí, když vidličky přijdou do kontaktu s médiem (deštěm, kroupami). Tuto změnu vibrační senzor vyhodnotí a vygeneruje spínací signál.

Vlastnosti:

- Pracovní napětí: 3.3V ~ 5V
- Typ výstupu: Digitální spínací výstup (0 a 1)
- Může detekovat vibrace okolního prostředí
- Detekce vibrací, alarmující funkce atd.

7.10 Krokový motorek

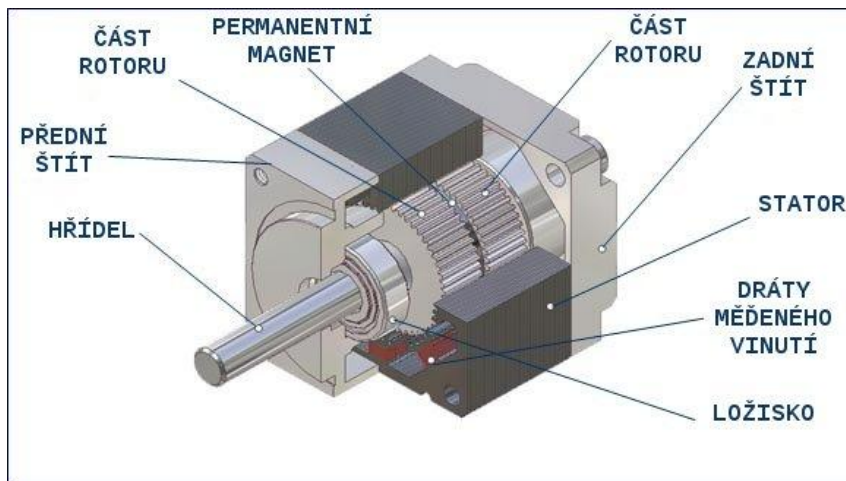


Obrázek 18: Krokový motorek

(Zdroj: <http://www.hwkitchen.com/products/stepper-motor-with-cable/>)

Mercury krokový motor SM-42BYG011-25

Krokový motor je synchronní motor, který se skládá z rotoru a statoru. U synchronních motorů se rotor točí stejnou rychlostí jako točivé pole ve statoru. Točivé pole je vytvářeno přiváděním proudu na jednotlivé cívky statoru. Stator se skládá z několika dvojic cívek. Hlavní rozdíl oproti elektromotorům je ten, že pohyb motoru je po jednotlivých krocích. Důležitým parametrem krokového motorku je úhel otočení o jeden krok. V experimentu byl použit motorek Mercury, který má úhel otočení $1,8^\circ$ což znamená, že celkový počet kroků = $360 / 1,8 = 200$ možných kroků. Počet zubů motorku vypočítáme počtem možných kroků / 4 – počet vinutí = 50 zubů.



Obrázek 19: Popis krokového motorku

(Zdroj: http://www.servo-drive.com/specialni_krokové_motory_a_krokové_motory_na_miru.php)

Vlastnosti:

- 200 kroku v jedné rotaci, jeden krok má úhel 1,8°
- Hmotnost 200g
- Dvě cívky
- Jmenovité napětí 12V
- Jmenovitý proud 0.33A
- Točivý moment 2,3 kg na cm²
- 5mm průměr hnací hřídele
- Velikost 42x42x35

7.10.1 Výhody a nevýhody krokového motorku

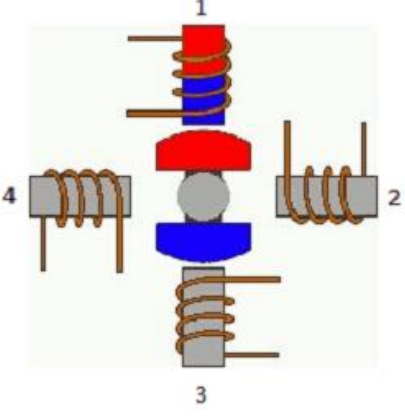
Výhody krokového motoru:

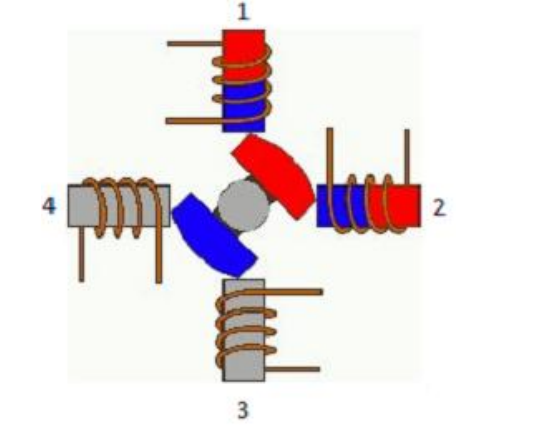
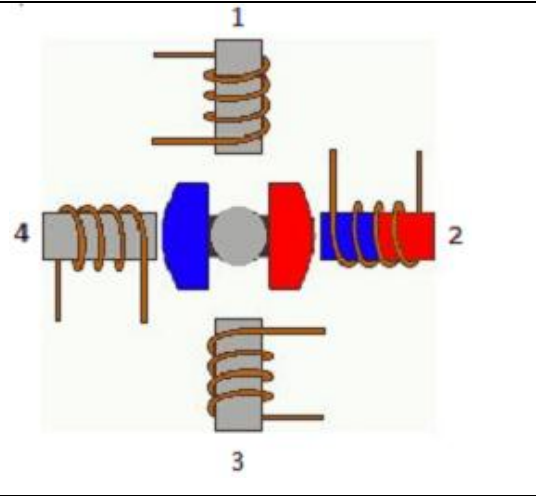
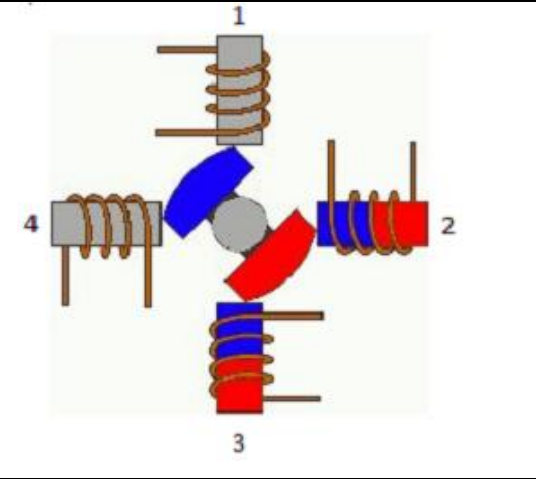
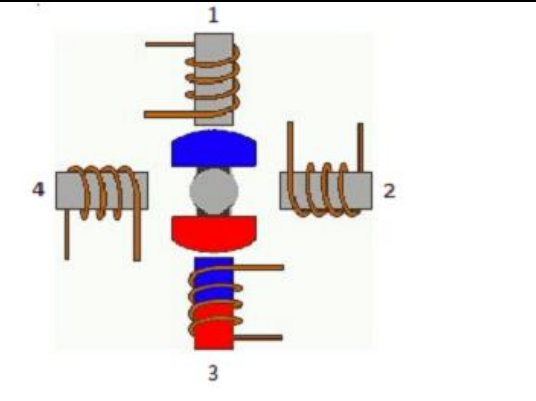
- Jediný zcela digitální motor
- Velmi jednoduše říditelný
- Jednoduchý a levný
- Žádná úhlová chyba
- Bezkartáčový
- Žádná údržba
- Velmi vysoký točivý moment
- Žádná zpětná vazba
- Otáčí se oběma směry
- Kompatibilní od různých výrobců

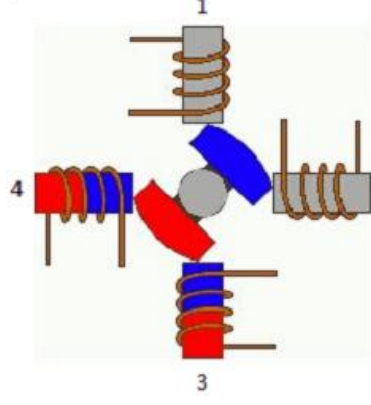
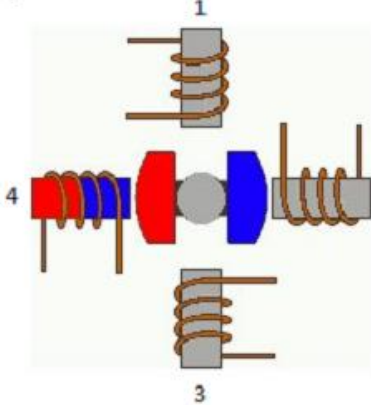
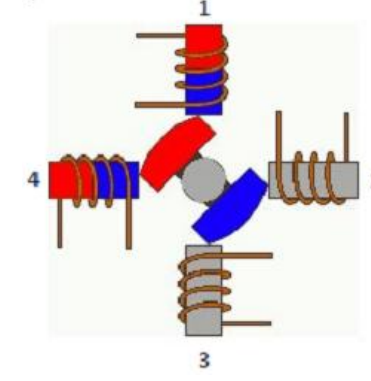
Nevýhody krokového motoru:

- Žádná zpětná informace o skutečné poloze
- Necítí rozdíl při různých zatíženích
- Při přetížení ztratí bez varování pozici
- Max. otáčky: zřídka více než 1 500 ot/min
- Ztrácí moment při rostoucích otáčkách
- Je poměrně hlučný

Otáčení motorku se realizuje postupným přiváděním proudu na cívky. To zajišťuje ovladač jednotlivých cívek krokových motorků, který generuje impulsy v určitém sledu a délce.

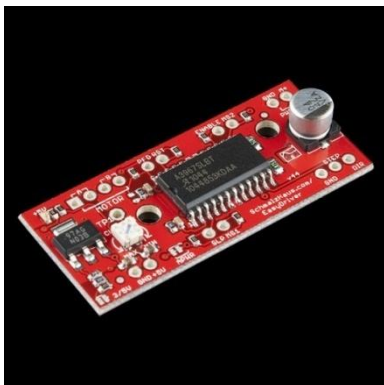
Stav cívek: 1=sepnuto 0=rozepnuto	Grafické znázornění:
Cívka 1 : 1 Cívka 2 : 0 Cívka 3 : 0 Cívka 4 : 1	

<p>Cívka 1 : 1</p> <p>Cívka 2 : 1</p> <p>Cívka 3 : 0</p> <p>Cívka 4 : 0</p>	
<p>Cívka 1 : 0</p> <p>Cívka 2 : 1</p> <p>Cívka 3 : 0</p> <p>Cívka 4 : 0</p>	
<p>Cívka 1 : 0</p> <p>Cívka 2 : 1</p> <p>Cívka 3 : 1</p> <p>Cívka 4 : 0</p>	
<p>Cívka 1 : 0</p> <p>Cívka 2 : 0</p> <p>Cívka 3 : 1</p> <p>Cívka 4 : 0</p>	

<p>Cívka 1 : 0</p> <p>Cívka 2 : 0</p> <p>Cívka 3 : 1</p> <p>Cívka 4 : 1</p>	
<p>Cívka 1 : 0</p> <p>Cívka 2 : 0</p> <p>Cívka 3 : 0</p> <p>Cívka 4 : 1</p>	
<p>Cívka 1 : 1</p> <p>Cívka 2 : 0</p> <p>Cívka 3 : 0</p> <p>Cívka 4 : 1</p>	

Obrázek 20: Popis cívek krokového motorku

7.10.2 Ovladač krokového motorku



Obrázek 21: Ovladač krokového motorku

(Zdroj: <https://www.sparkfun.com/products/13226>)

Řadič krokového motorku je jednoduchý integrovaný obvod na řízení motoru. Je kompatibilní s Arduinem, který ho řídí digitálním impulsem v rozmezí 0-5V. Řadič je schopen napájet krokový motor v rozmezí 7 – 20V DC. Napájení je samostatné. Řadič obsahuje regulátor napětí, který je možné nastavit na 5 nebo 3,3V.

7.11 Čerpadlo SEAFLO 350



Obrázek 22: Ponorné čerpadlo

(Zdroj: <http://www.hwkitchen.com/products/liquid-pump-350gph-12v-/>)

Klasické, ponorné a vodou chlazené čerpadlo, které je napájeno 12V a výkonem čerpání kapaliny 1325 litrů za hodinu. Čerpadlo má vyrobené tělo z tvrdého termoplastu. Čerpadlo lze připojit na $\frac{3}{4}$ coulovou hadici. Čerpadlo obsahuje krytku, která slouží jako ochrana vrtulky a zároveň zabraňuje nasátí velkých předmětů, které by mohly toto zařízení poškodit.

7.12 Ethernet shield



Obrázek 23: Ethernet shield

(Zdroj: <http://arduino.cc/en/Main/ArduinoEthernetShield>)

Velkou výhodou Arduina systému jsou přídavné desky (tzv. shields). Shieldy neumí pracovat samostatně. Pro chod se musí shield připojit na Arduino základní desku. Ethernet shield lze připojit k síti pomocí kabelu s RJ-45 koncovkou (síťový kabel) a dodržovat několik základních pokynů pro řízení svého projektu odkudkoliv z celého světa přes internetovou síť. Připojení k internetu zprostředkovává ethernet čip Wiznet W5100. Ten poskytuje IP adresu a podporu TCP a UDP. Pomocí knihovny Ethernet napíšeme program pro připojení k Internetu pomocí ethernetu. Ethernet shield obsahuje slot na micro-SD kartu, která může být použita k ukládání souborů pro další využití v síti.

Vlastnosti:

Provozní napětí: 5V

Rychlost připojení: 10/100 Mb

7.13 Modul tlačítek



Obrázek 24: Modul tlačítek

(Zdroj: <http://aukro.cz/ard4-04-arduino-kapacitni-klavesnice-8-pozic-i5128496800.html>)

TTP222 je integrovaný obvod detektoru dotykového panelu. Integrovaný obvod dotykové detekce je navržen jako náhrada tradičního tlačítkového panelu s pevně danou velikostí. Nízký energetický odběr a široké operační napětí jsou hlavními výhodami tohoto modulu.

Vlastnosti:

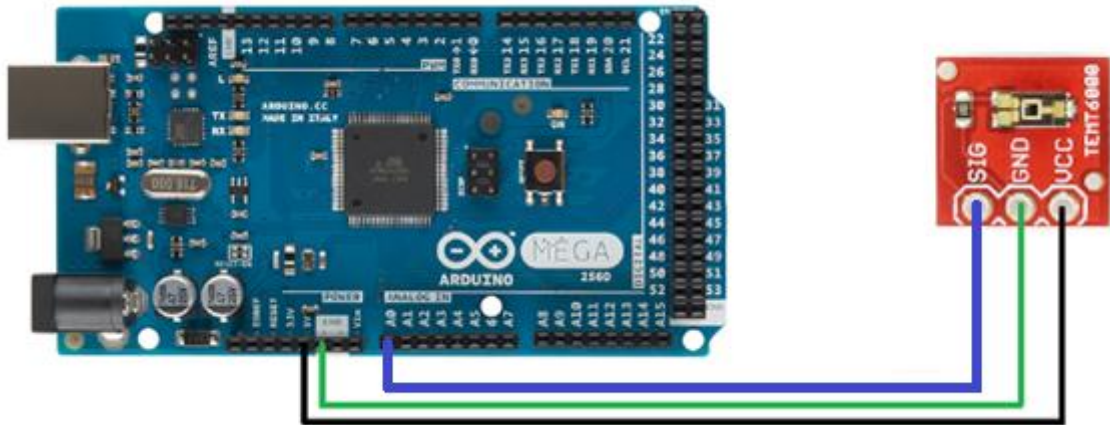
- Pracovní napětí: 3,3V
- Velikost desky: 47.5 (mm) x 46 (mm)

Praktická část

8 Volba zapojení

8.1 Senzor světla TEMT 6000

Schéma zapojení:²



Obrázek 25: Schéma zapojení senzoru světla

Popis 1:

Senzor vlhkosti obsahuje 3 konektory. Pro funkčnost musíme zapojit napájení a zem. Napájení z arduina připojíme na konektor VCC. Senzor světla pracuje při provozním napětí 5V, proto tedy zapojíme konektor VCC k napájecímu napětí Arduino 5V. Konektor GND na senzoru světla propojíme s konektorem na Arduino se zkratkou GND. Konektor SIG znamená Signál. Protože senzor světla vysílá analogový signál, tak připojíme konektor k arduinu, který obsahuje analogové vstupy A0-A15. V našem případě jsem si vybral pin A0.

Program:

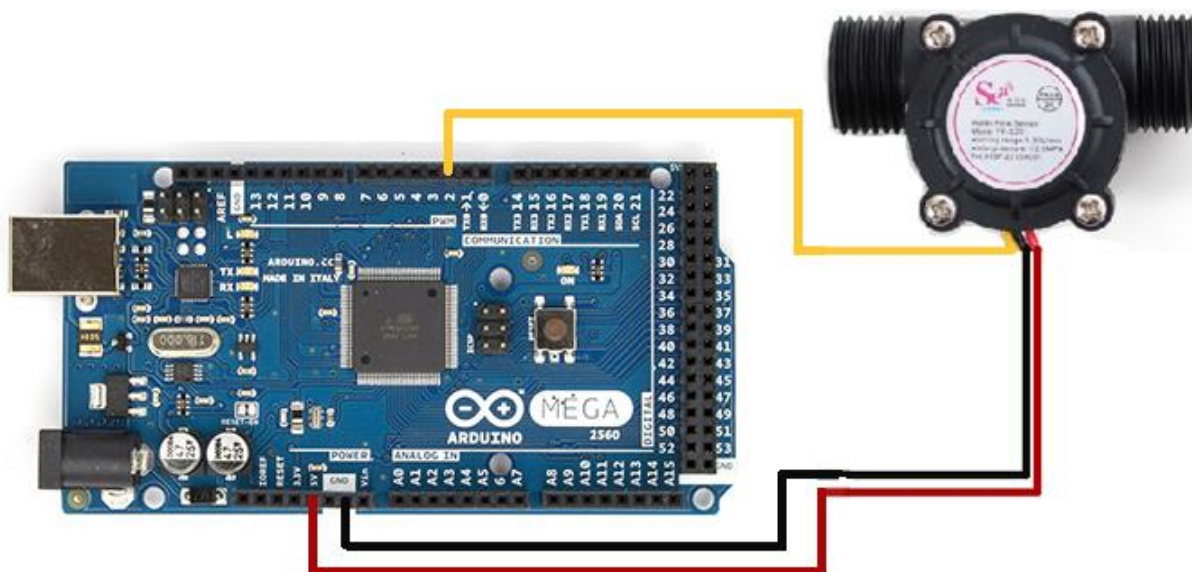
```
int temt6000Pin = A0; // data čidla jsou připojeny na pin A0
void setup() {
  Serial.begin(9600); } // nastavení sériové komunikace mezi arduinem a ...
void loop() { // ... počítačem
  int value = analogRead(temt6000Pin); //změř hodnotu
```

² Obrázek základní desky převzat z : www.arduino.cc , obrázek čidla převzat z: www.sparkfun.com , zapojení vodičů: vlastní návrh. Uvedené platí i pro následující obrázky.

```
Serial.println(value); // vypiš hodnotu světla na displej
delay(1000); } // každou vteřinu
```

8.2 Průtokoměr

Schéma zapojení:



Obrázek 26: Schéma zapojení průtokoměru

Popis 2:

Z průtokoměru jsou vyvedeny tři vodiče. Červený vodič znamená napájení, proto tedy připojíme červený vodič průtokoměru k napájecímu pinu Arduino 5V. Černý vodič charakterizuje zem, tedy GND. Černý vodič průtokoměru připojíme k pinu Arduino GND. Žlutý vodič průtokoměru znamená data. V programu je použita speciální funkce AttachInterrupt, která je kompatibilní pouze s piny uvedené výrobcem.

Board	int.0	int.1	int.2	int.3	int.4
Uno, Ethernet	2	3			
Mega2560	2	3	21	20	19
Leonardo	3	2	0	1	7

Obrázek 27: Funkce attachInterrupt

Tabulka nám zobrazuje, na které piny se můžou připojit data. V našem případě je použita deska Arduino Mega2560. Data připojíme k Arduino na pin 2.

Program:

Program byl převzat z webové stránky zdroj: (<http://diyhacking.com/projects/FlowMeterDIY.ino>) kde autor snímá aktuální průtok, který rozsvěcuje světlo LED. Tento program byl mnou upraven pro použití ve vzdáleném experimentu – Inteligentní skleník.

```
byte sensorInterrupt = 0;
byte sensorPin      = 2;
float calibrationFactor = 4.5; // Kalibrační faktor pro pozdější výpočet
byte pulseCount;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
unsigned long oldTime;
void setup() {
  Serial.begin(9600);
  pinMode(sensorPin, INPUT); // Nastavení pinu jako vstup
  digitalWrite(sensorPin, HIGH);
  pulseCount      = 0;
  flowRate        = 0.0;
  flowMilliLitres = 0;
  totalMilliLitres = 0;
  oldTime         = 0;
```

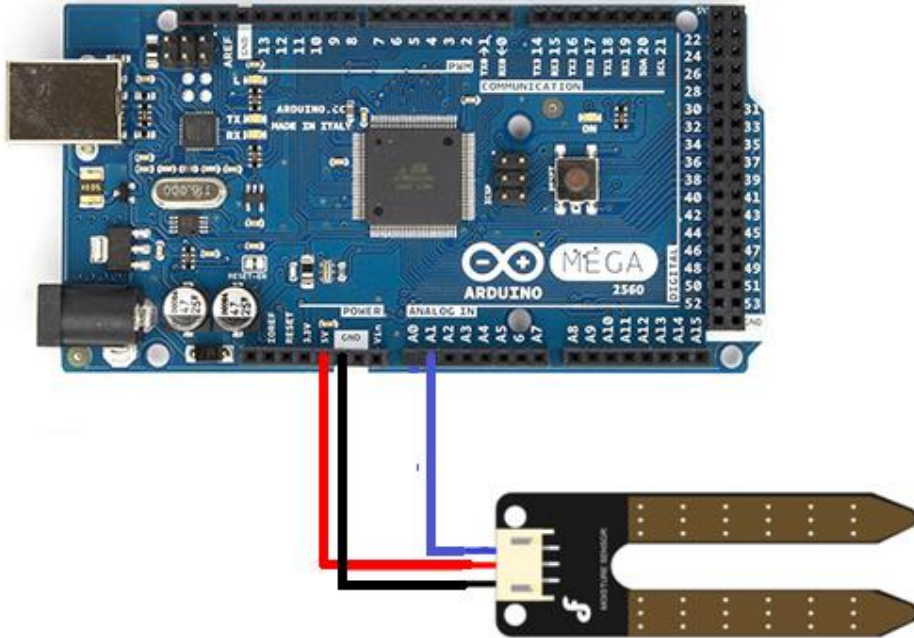
// Funkce Attachinterrupt je speciální funkce, která se používá proti zmeškání změny stavu na daném pinu. Výrobce nastavil pin sensorInterrupt 0 je digitální pin číslo 2. Dále sledujeme funkci interrupt s číslem sensorinterrupt a při zjištění změny stavu na pinu 2 zavolá funkci pulseCounter. Funkce FALLING určuje samostatnost snímání v jakémkoliv čase při změně signálu na pinu 2.

```
attachInterrupt(sensorInterrupt, pulseCounter, FALLING); }
void loop() {
```

```
if((millis() - oldTime) > 1000) { // průtok program přepočítává každou jednu
vteřinu
detachInterrupt(sensorInterrupt);
flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
oldTime = millis();
flowMilliLitres = (flowRate / 60) * 1000;
totalMilliLitres += flowMilliLitres;
unsigned int frac;
pulseCount = 0;
attachInterrupt(sensorInterrupt, pulseCounter, FALLING); } }
void pulseCounter() {
pulseCount++; }
```

8.3 Vlhkost půdy

Schéma zapojení:



Obrázek 28: Schéma zapojení senzoru vlhkosti půdy

Popis 3:

Červený vodič znamená napájení, proto tedy připojíme červený vodič průtokoměru k napájecímu pinu Arduina 5V. Černý vodič charakterizuje zem, tedy GND. Černý vodič průtokoměru připojíme k pinu Arduina GND. Senzor vlhkosti půdy vydává analogový signál, proto tedy připojíme modrý vodič k Arduinu na pin A1, protože pin A0 je již obsazen senzorem světla.

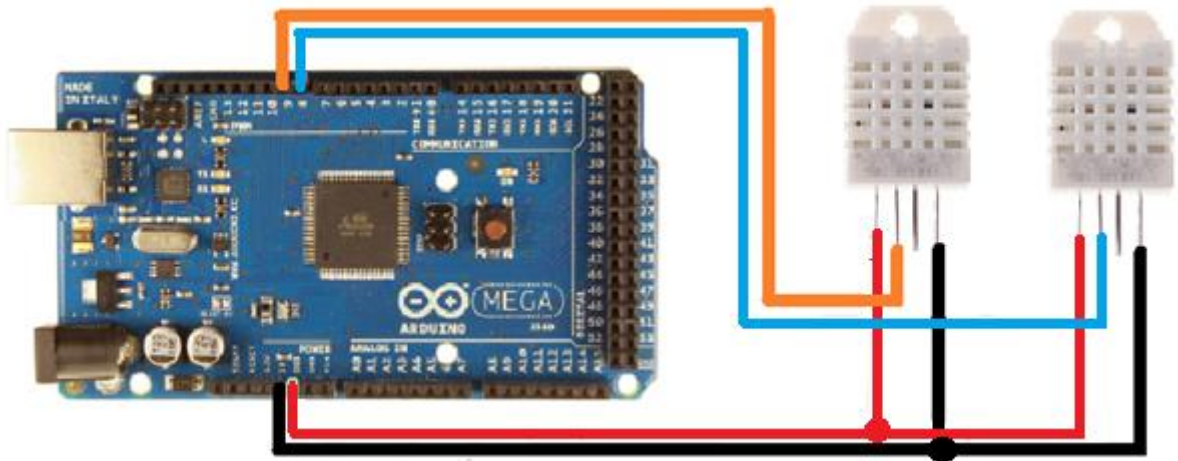
Program:

```
int sensorPin=A1; // senzor vlhkosti je připojen na analogový pin A1
void setup(){
  Serial.begin(9600);} // nastavení sériové komunikace mezi arduinem a pc
void loop(){
  int sensorValue = analogRead(1); // zjistí hodnotu vlhkosti půdy
  Serial.print(analogRead); // vypiš hodnotu na displej
  delay(2000); } // každé dvě sekundy
```

8.4 Senzor teploty a vlhkosti vzduchu

Program je navrhnut tak, aby mohla pracovat dvě čidla teploty a vlhkosti vzduchu současně. Tento způsob získávání hodnot z více čidel je uplatnitelný ve velkých prostorech, kde je teplota v různých částech rozdílná.

Schéma zapojení:



Obrázek 29: Schéma zapojení senzoru teploty a vlhkosti vzduchu

Popis 4:

Čidlo DHT22 obsahuje 4 konektory: zleva: VCC, Data, žádné připojení a GND. Červený vodič připojíme na první konektor DHT22 a druhý konec připojíme k Arduino na pin 5V. Čidlo DHT22 vysílá digitální data, proto připojíme oranžový vodič k Arduino na jakýkoliv volný pin digitálních vstupů. V našem případě jsem zvolil pin 8. Dále připojíme modrý vodič, tedy data druhého čidla k Arduino na volný digitální pin 9. Černý vodič znamená GND, proto zapojíme černý vodič k Arduino na pin GND. 3. konektor zleva nemá žádné připojení.

Program:

```
#include <DHT.h> // knihovna pro DHT
#define DHT1PIN 8 // Připojení teploty 1 na pin 8
#define DHT2PIN 9 // Připojení teploty 2 na pin 9
#define DHT1TYPE DHT22
#define DHT2TYPE DHT22
```

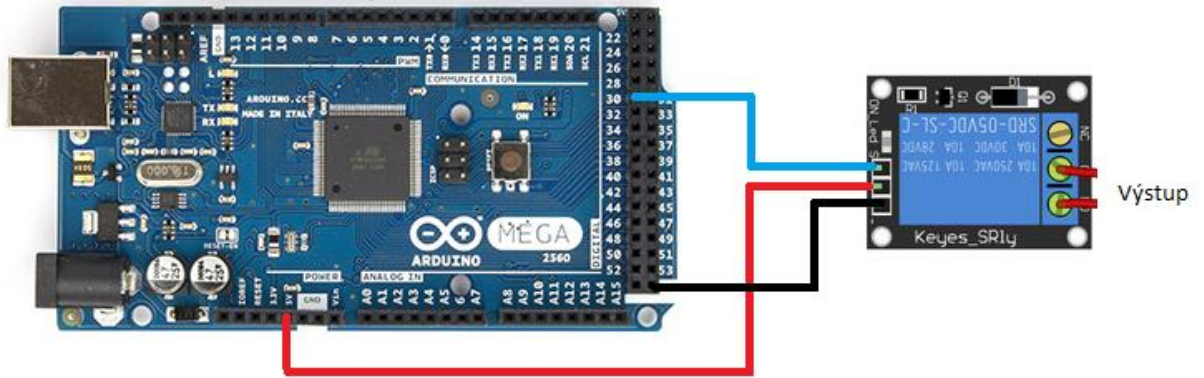
```

DHT dht1(DHT1PIN, DHT1TYPE);
DHT dht2(DHT2PIN, DHT2TYPE);
void setup() {
Serial.begin(9600);
dht1.begin(); // začni měřit teplotu a vlhkost 1
dht2.begin(); } // začni měřit teplotu a vlhkost 2
void loop() {
delay(4000); // čekej 4 sekundy
float h1 = dht1.readHumidity();
float t1 = dht1.readTemperature();
float h2 = dht2.readHumidity();
float t2 = dht2.readTemperature();
if (isnan(t1) || isnan(h1)) { // pokud nenajde program teplotu 1
Serial.println(„Chyba čtení z DHT“); // vypíše chybu na displej
} else { // pokud najde
Serial.print("Vlhkost 1: "); //vypíše hodnotu vlhkosti 1 v %
Serial.print(h1);
Serial.print(" %");
Serial.print("Teplota 1: "); // vypíše hodnotu teploty 1 v °C
Serial.print(t1);
Serial.println(" °C"); }
if (isnan(t2) || isnan(h2)) { //pokud nenajde program data z čidla 2
Serial.println("Chyba čtení z DHT2"); // vypíše chybu
} else { // pokud najde
Serial.print("Vlhkost 2: "); //vypíše Vlhkost 2 v %
Serial.print(h2);
Serial.print(" %");
Serial.print("Teplota 2: "); // a teplotu 2 v °C
Serial.print(t2);
Serial.println(" °C"); }
Serial.println(); }

```


8.5 Relé

Schéma zapojení:



Obrázek 30: Schéma zapojení relé

Popis 5:

Spínací relé obsahuje 3 konektory. VCC, GND a Data. Konektor VCC připojíme na napájecí pin Arduina na 5V. Zem, tedy GND, připojíme k arduinu na volný pin GND. Spínací relé je řízeno pomocí digitálního vstupu, proto tedy připojíme modrý vodič k Arduinu na jakýkoliv volný digitální pin. V našem případě na pin 30.

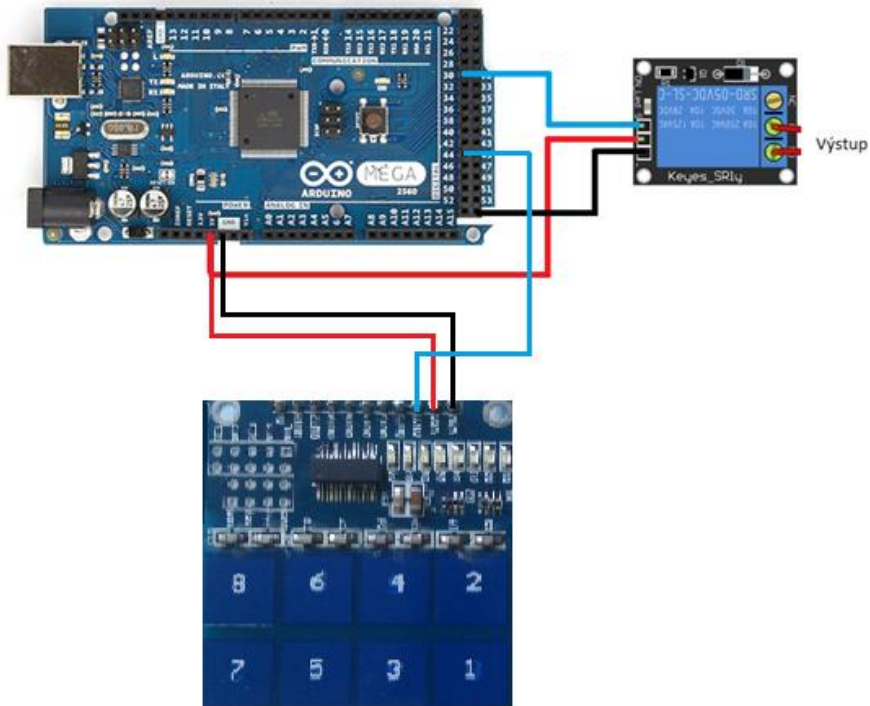
Program:

Pro spínání relé nám stačí příkaz HIGH a LOW. Příkaz HIGH relé spíná a příkaz LOW relé rozeptíná.

```
void setup() {  
  pinMode(30, OUTPUT);} // nastavení pinu 30 jako výstup  
void loop() {  
  digitalWrite(30, HIGH); // sepní relé  
  delay(1000); // čekej 1 sekundu  
  digitalWrite(30, LOW); // rozeptíná relé  
  delay(1000); } // čekej 1 sekundu
```

8.6 Relé a tlačítko

Schéma zapojení:



Obrázek 31: Schéma zapojení tlačítek a relé

Popis 6:

Zapojení relé je stejné jako v popisu 5. V předchozím případě nám relé řídil program. V tomto případě nám relé bude řídit tlačítko. Modul tlačítek obsahuje konektory VCC pro napájení, GND pro zem a jeden konektor pro každé tlačítko, tedy dohromady 8 použitelných tlačítek. Napájení modulu tlačítek je připojeno k Arduino na pin 5V. Konektor modulu GND připojíme k Arduino na volný pin GND. Po stisknutí tlačítka vyšle modul digitální signál do Arduina, proto připojíme konektor data na libovolný nezabraný digitální pin. V našem případě pin 44.

Program:

```
const int buttonPin = 44;    // tlačítko je připojeno na pin 44
const int relePin = 30;     // relé je připojeno na pin 30
int buttonState = 0;        // proměnná ukládá stav tlačítka
void setup() {
```

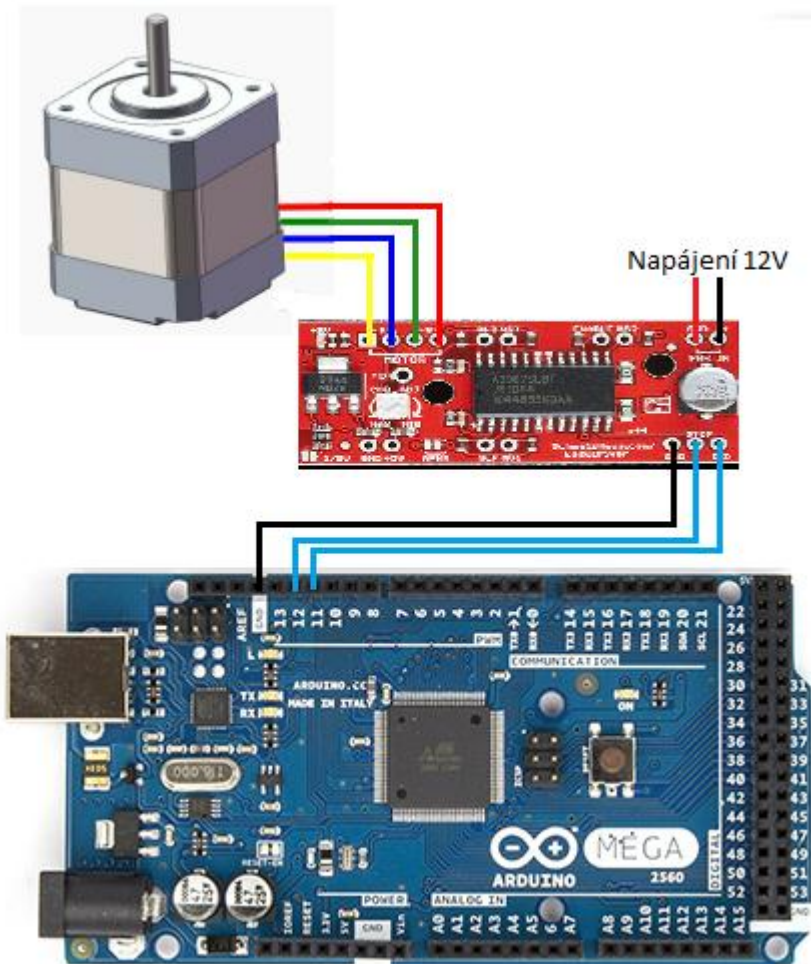
```

pinMode(relePin, OUTPUT); // rele pin nastaven jako vystup
pinMode(buttonPin, INPUT); } // pin tlačítka nastaven jako vstup
void loop(){
buttonState = digitalRead(buttonPin); // načteme si stav tlačítka
if (buttonState == HIGH) {           // je-li tlačítko stisknuto
digitalWrite(relePin, LOW); } // zapneme relé:
else {                                 // když není
digitalWrite(relePin, HIGH); } } // vypneme relé:

```

8.7 Krokový motorek

Schéma zapojení:



Obrázek 32: Schéma zapojení krokového motorku

Popis 7:

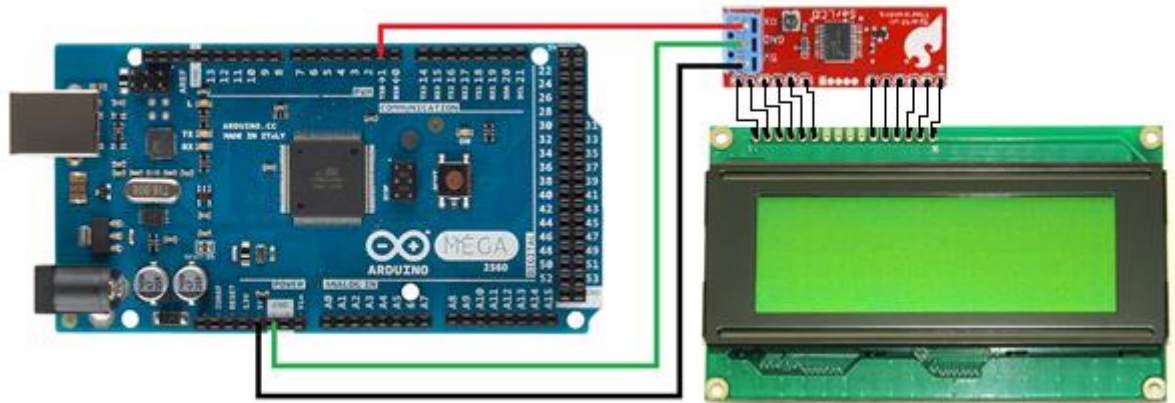
Motorek ovládá řadič, který je zapojen mezi Arduinem a motorkem. Řadič obsahuje celkem 9 zapojených konektorů. Čtyři konektory pro připojení motorku. Dva pro cívku A, dva pro cívku B. Modrý a žlutý vodič připojíme na piny cívky A. Zelený a červený vodič připojíme na piny cívky B. Napájení motorku je vedeno přes řadič. Na konektor řadiče M+ připojíme ze zdroje napájecí napětí 12V DC a na GND připojíme zem ze zdroje. Pro získání stejnosměrného napájecího napětí 12V byl použit laboratorní zdroj. Pro komunikaci obsahuje řadič celkem 3 konektory STEP, DIR a GND. Konektor STEP slouží jako prostředek k řízení cívek v motorku. Konektor DIR slouží pro nastavení směru otáčení motorku a GND je zem obvodu. Konektor DIR přijímá digitální signál, proto připojíme vodič k Arduinu na digitální pin 11. Konektor STEP také přijímá digitální signál, proto připojíme vodič k Arduinu na digitální pin 12. Třetí konektor GND připojíme k Arduinu na volný pin GND.

Program:

```
int dirpin = 11; // připojení konektoru dir na pin 11
int steppin = 12; // připojení konektoru step na pin 12
void setup() {
  pinMode(dirpin, OUTPUT); // nastavení dir jako výstup
  pinMode(steppin, OUTPUT); } // nastavení step jako výstup
void loop() {
  int i;
  digitalWrite(dirpin, LOW); // nastavení směru otáčení
  delay(100); // čekej 100 ms
  for (i = 0; i<4000; i++) { // otočení rotoru o 4000 kroků
    digitalWrite(steppin, LOW); // program přivede napětí na cívky v motorku
    delay(10);
    digitalWrite(steppin, HIGH); // program ukončí přívod napětí na cívky
    delayMicroseconds(500); } // hodnota rychlosti otáčení hřídele
```

8.8 LCD displej

Schéma zapojení:



Obrázek 33: Schéma zapojení LCD displeje s ovladačem

Popis 8:

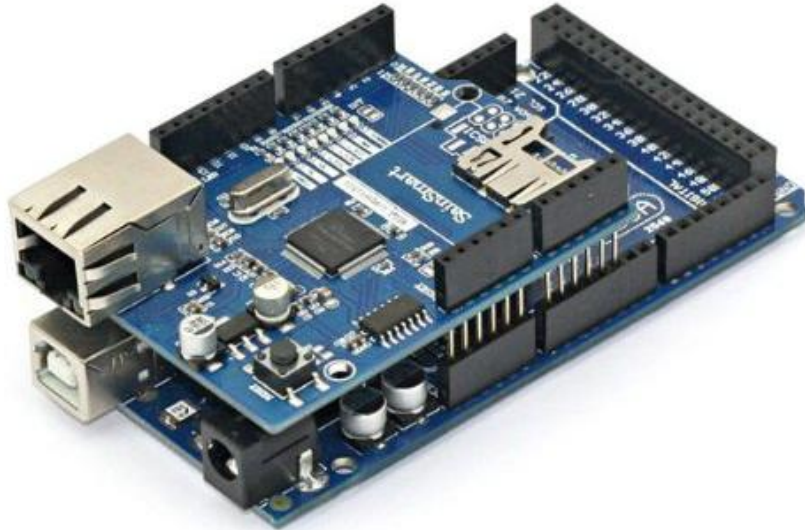
Mezi Arduinem a LCD displejem je zařazen sériový řadič pro LCD displeje. 12 konektorů, vedoucích ze sériového řadiče, připojíme k LCD displeji. Je jen jedna možnost, jak tyto konektory zapojit. Na výstupu řadiče jsou tři konektory. VCC, GND a data. VCC připojíme červeným vodičem k Arduinu na pin 5V. GND připojíme černým vodičem k Arduinu na pin GND a data připojíme modrým vodičem na pin TX1. Tento pin slouží pro připojení LCD displejů k Arduinu. Pokud bychom připojili konektor data např. k pinu 5 LCD, displej by nám nefungoval. Pro správnou funkci displeje je nutné upravit program tak, aby Arduino rozpoznalo, že LCD displej má velikost 4x20 bodů.

Program:

```
void setupDisplay() { // Natavení displeje na 4x20 bodů
  Serial.write(0x7C);
  Serial.write(3);}
  Serial.print("Dobrý den"); // na první řádek LCD vypíše program: Dobrý den
```

8.9 Ethernet

Schéma zapojení:



Obrázek 34: Schéma zapojení ethernet shieldu

Popis 9:

Zapojení je jednoduché. Ethernet shield obsahuje stejné piny jako Arduino Mega2560, tudíž tyto dvě desky do sebe pouze zasuneme.

Program:

```
#include <Ethernet.h>
byte mac[] = { 0xD4, 0xAD, 0xBE, 0xEF, 0xFE, 0x7D }; // MAC adresa daná
výrobcem
IPAddress ip(192,168,0, 150); // nastavení IP adresy
IPAddress dns(192.168.0.1); // nastavení serveru DNS
IPAddress gateway(192.168.0.1); // Nastavení výchozí brány
EthernetServer server(80); // komunikace mezi serverem a počítačem
void setup() {
  Ethernet.begin(mac, ip, dns, gateway); }
void loop() {
  EthernetClient client = server.available(); // Jestli je připojen klient
  while (client.connected()) { // čti data
```

```

if (client.available()) {
char c = client.read();
if (c == '\n') {
client.println("HTTP/1.1 200 OK"); // odešle klasickou hlavičku HTML stránky
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>Inteligentní skleník</title>"); // název stránky
client.println("</head>");
client.println("<body>");
client.print(" Inteligentní skleník – Tomáš Vítek"); }
client.print("</br>");
client.print("Teplota vzduchu 1:"); // zobrazení teploty vzduchu 1 ve °C
client.print(dht1.readTemperature(),1);
client.print((char)223); // vypíše znak °
client.print("C ");
client.print("</br>");
client.print("Teplota vzduchu 2:"); // zobrazení teploty vzduchu 2 ve °C
client.print(dht2.readTemperature(),1);
client.print((char)223); // vypíše znak °
client.print("C");
client.print("</br>");
client.print("Vlhkost vzduchu 1: "); // zobrazení vlhkosti vzduchu 1 v %
client.print(dht1.readHumidity(),1);
client.print("% ");
client.print("</br>");
client.print("Vlhkost vzduchu 2: "); // zobrazení vlhkosti vzduchu 2 v %
client.print(dht2.readHumidity(),1);
client.print("%");
client.print("</br>");
client.print("Vlhkost pudy: "); // zobrazení vlhkosti půdy v %

```

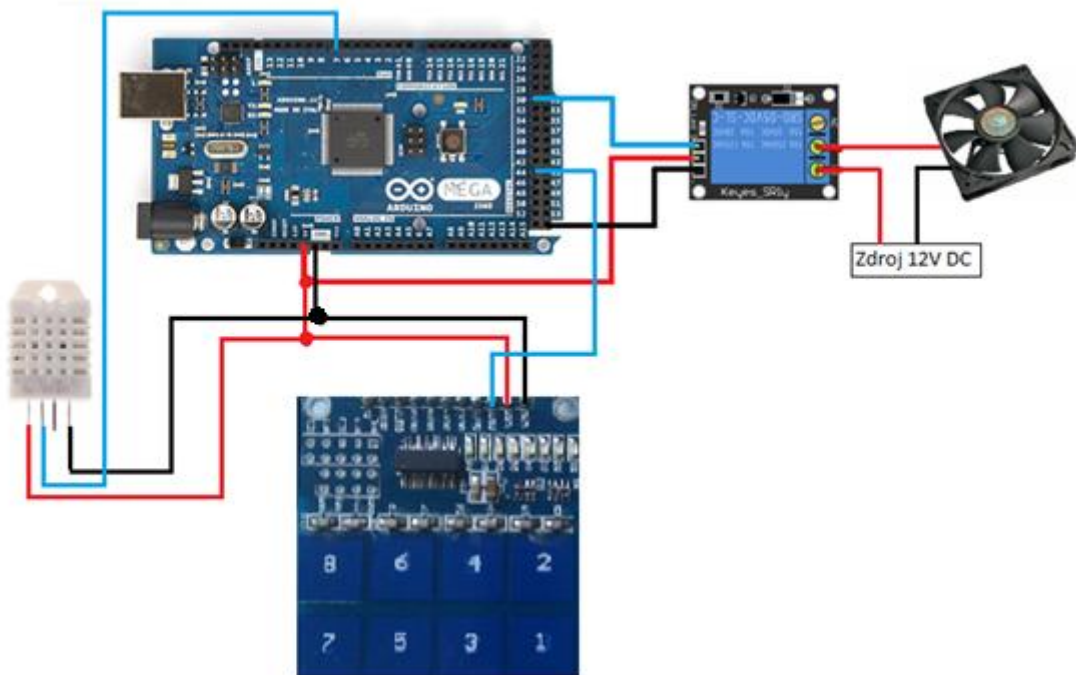
```
client.print(analogRead(A1)/11,1); // převedení hodnoty na %
client.print("% ");
client.print("</br>");
client.print("Hodnota svetla: "); // zobrazení hodnoty světla v %
client.print(analogRead(temt6000Pin)/11,1); // převedení hodnoty na %
client.print("% ");
client.print("</br>");
client.print("Aktualni prutok: "); // zobrazení aktuálního průtoku v litrech za
minutu
client.print(int(flowRate));
client.print(".");
frac = (flowRate - int(flowRate)) * 10;
client.print(frac, DEC) ;
client.print("L/m ");
client.print("</br>");
client.print("Celkovy prutok: "); // zobrazení celkového průtoku vody v
mililitrech
client.print(totalMilliLitres);
client.println("mL ");
```


9 Propojení vstupů a výstupů

9.1 Ovládání větrání

Ovládání ventilátoru pomocí teploty a vlhkosti vzduchu nebo tlačítka.

Schéma zapojení:



Obrázek 35: Schéma zapojení větrání

Popis 9:

Zapojení čidla teploty a vlhkosti vzduchu je stejné jako v popisu 4. Zapojení relé a modulu tlačítek je stejné jako v popisu 6. Ovládání ventilátoru připojíme červeným vodičem přes relé k napájecímu konektoru zdroje a černý vodič připojíme přímo na GND zdroje. Jako zdroj stejnosměrného napětí je v našem případě realizováno pomocí laboratorního zdroje.

Program:

```
#include <DHT.h> // vložení knihovny pro čidlo DHT
#define DHT1PIN 7 // teplota 1 je připojena na pin 7
#define DHT1TYPE DHT22 //definuje typ senzoru
DHT dht1(DHT1PIN, DHT1TYPE);
```

```

const int buttonPin1 = 50; // kam je připojeno tlačítko
const int relePin1 = 30; // kam je připojeno relé
int buttonState1 = 0; // nastavení tlačítka jako 0
void setup() {
Serial.begin(9600);
pinMode(relePin1, OUTPUT); // nastavení relé pinu jako výstup
pinMode(buttonPin1, INPUT); // Nastavení tlačítka pinu jako vstup
dht1.begin(); // začni měřit teplotu a vlhkost 1
void loop() {
buttonState1 = digitalRead(buttonPin1);
if(buttonState1 == HIGH) { // Pokud je tlačítko stisknuto

digitalWrite(relePin1, LOW); } // Sepni relé

else{

if(dht1.readTemperature() >= 30) { // Pokud je teplota větší nebo

digitalWrite(relePin1, LOW); } // rovna 30°C sepni relé

if(dht1.readTemperature() <= 23){ // Pokud je teplota nižší nebo

digitalWrite(relePin1, HIGH);} } // rovna 23°C rozepni relé

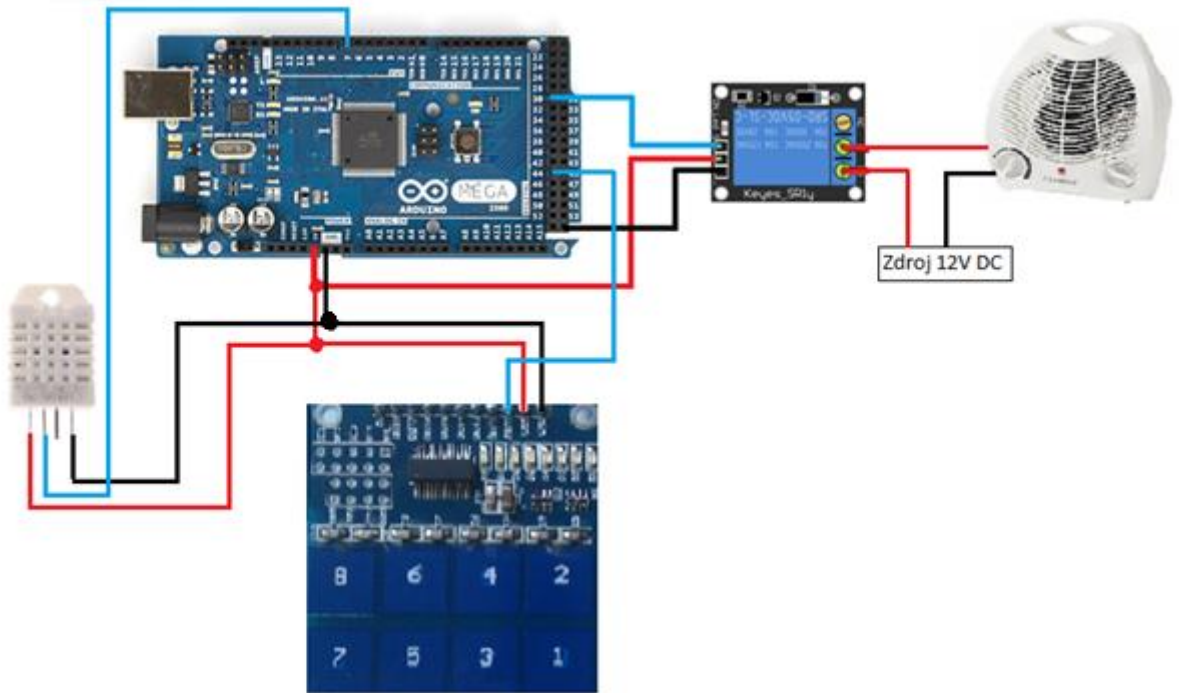
void teplotavlhkost();
float h1 = dht1.readHumidity();
float t1 = dht1.readTemperature();

```

9.2 Ovládání topení

Ovládání topení pomocí teploty vzduchu a tlačítka.

Schéma zapojení:



Obrázek 36: Schéma zapojení topení

Popis 10:

Zapojení je stejné jako v popisu 9. Topení obsahuje dva vodiče. Červený vodič, tedy napájení + přivedeme na zdroj napětí přes ovládací relé. Černý vodič zapojíme přímo na GND zdroje. Zdroj stejnosměrného napětí je v našem případě realizován pomocí laboratorního zdroje.

Program:

```
#include <DHT.h> // vložení knihovny pro čidlo DHT
#define DHT1PIN 7 // teplota 1 je připojena na pin 7
#define DHT2TYPE DHT22 //definuje typ senzoru
DHT dht1(DHT1PIN, DHT1TYPE);
const int buttonPin1 = 44; // kam je připojeno tlačítko
const int relePin1 = 30; // kam je připojeno relé
int buttonState1 = 0; // nastavení tlačítka jako 0
```

```

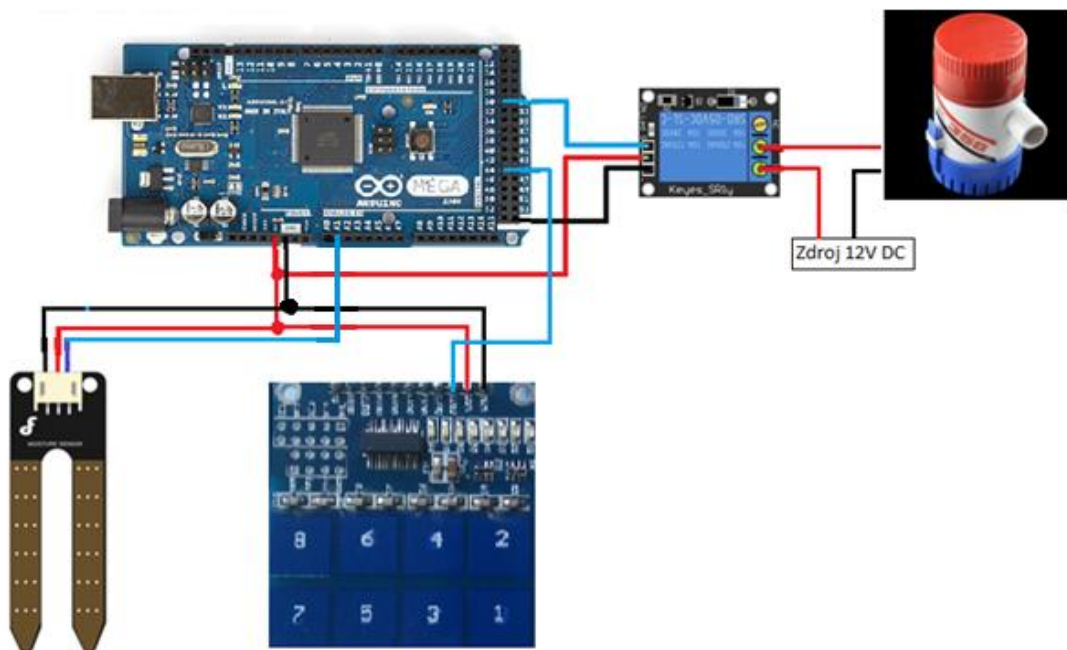
void setup() {
  Serial.begin(9600);
  pinMode(reléPin1, OUTPUT); // nastavení relé pinu jako výstup
  pinMode(buttonPin1, INPUT); // Nastavení tlačítka pinu jako vstup
  dht1.begin(); // začni měřit teplotu a vlhkost 1
  void loop() {
    buttonState2 = digitalRead(buttonPin2);
    if(buttonState2 == HIGH) { // Jestli je stisknuto tlačítko 2
      digitalWrite(reléPin3, LOW); // sepni relé 2
    } else{
      if(dht1.readTemperature() <= 18) { // Pokud je teplota nižší nebo ...
        digitalWrite(reléPin3, LOW);} //... rovna 18°C tak sepni relé(topení)
      if(dht1.readTemperature() >= 25){ // Pokud je teplota větší nebo ...
        digitalWrite(reléPin3, HIGH); } // ... rovna 25°C tak rozepni relé(topení)
    }
  }
}

```

9.3 Ovládání čerpadla

Ovládání čerpadla pomocí vlhkosti půdy a tlačítka.

Schéma zapojení:



Obrázek 37: Schéma zapojení čerpadla

Popis 11:

Zapojení vlhkosti půdy je stejné jako v popisu 3. Zapojení relé s modulem tlačítka je stejné jako v popisu 6. Čerpadlo se připojuje dvěma vodiči. Červený vodič, tedy napájení přivedeme na zdroj napětí přes ovládací relé. Černý vodič zapojíme přímo na GND zdroje. Zdroj stejnosměrného napětí je v našem případě realizován pomocí laboratorního zdroje.

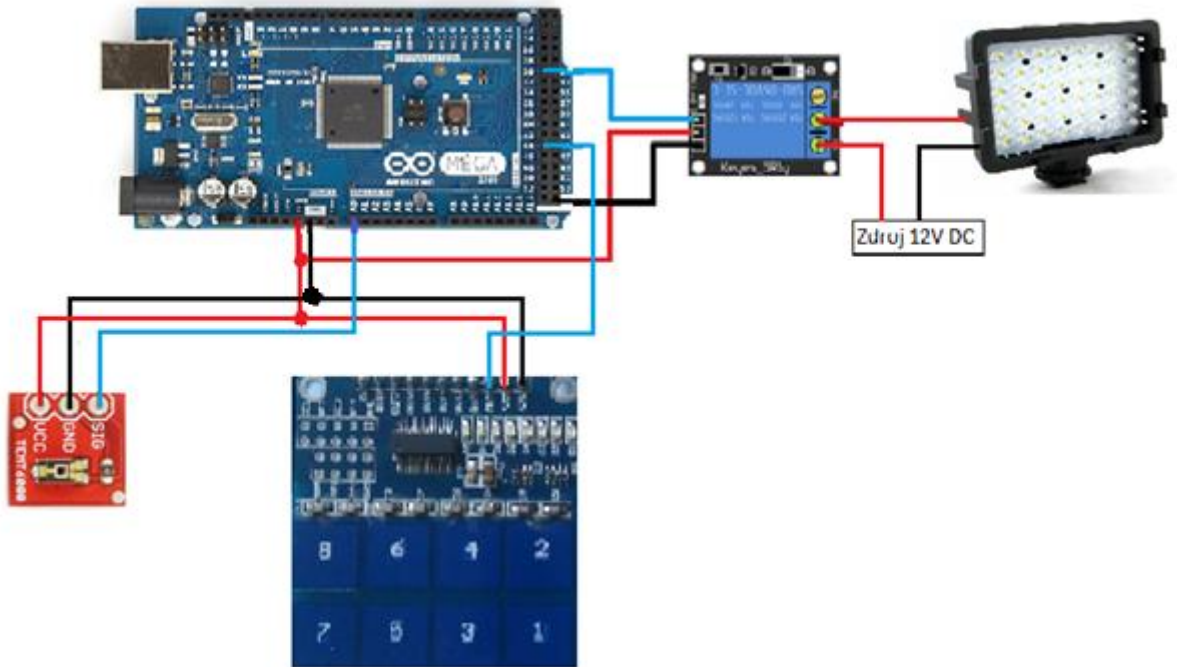
Program:

```
void (vlhkostpudy){
int sensorValue = analogRead(A1); // senzor vlhkosti půdy připojen na
analogový vstup A1
const int buttonPin1 = 44; // kam je připojeno tlačítko – pin 44
const int relePin1 = 30; // kam je připojeno relé – pin 30
int buttonState1 = 0; // nastavení tlačítka jako 0
void setup() {
Serial.begin(9600);
pinMode(relePin1, OUTPUT); // nastavení relé pinu jako výstup
pinMode(buttonPin1,INPUT); } // Nastavení tlačítka pinu jako vstup
void loop() {
buttonState3 = digitalRead(buttonPin3);
if(buttonState3 == HIGH) { //Jestli je tlačítko 3 sepnuto
digitalWrite(relePin4, LOW); } // sepni relé 3 (čerpadlo)
else{
if(analogRead(A1) >= 900) { // Pokud je hodnota vlhkosti půdy ...
digitalWrite(relePin4, LOW); } //... >= 900 tak sepni relé 3 (čerpadlo)
if(analogRead(A1) <= 400){// Pokud je hodnota vlhkosti půdy ...
digitalWrite(relePin4, HIGH); } } } // ... <= 400 tak rozepni relé 3 (čerpadlo)
```

9.4 Ovládání světla

Ovládání osvětlení pomocí senzoru světla a tlačítka.

Schéma zapojení:



Obrázek 38: Schéma zapojení osvětlení

Popis 12:

Zapojení senzoru světla je stejné jako v popisu 1. Zapojení relé s modulem tlačítka je stejné jako v popisu 6. Světlo obsahuje dva vodiče. Červený vodič, tedy napájení + přivedeme na zdroj napětí přes ovládací relé. Černý vodič zapojíme přímo na GND zdroje. Zdroj stejnosměrného napětí je v našem případě realizován pomocí laboratorního zdroje.

Program:

```
void svetlo(){
int value = analogReadA0(temt6000Pin); } // senzor světla připojíme na
analogový vstup A0
const int buttonPin1 = 44; // kam je připojeno tlačítko – pin 44
const int relePin1 = 30; // kam je připojeno relé – pin 30
int buttonState1 = 0; // nastavení tlačítka jako 0
```

```

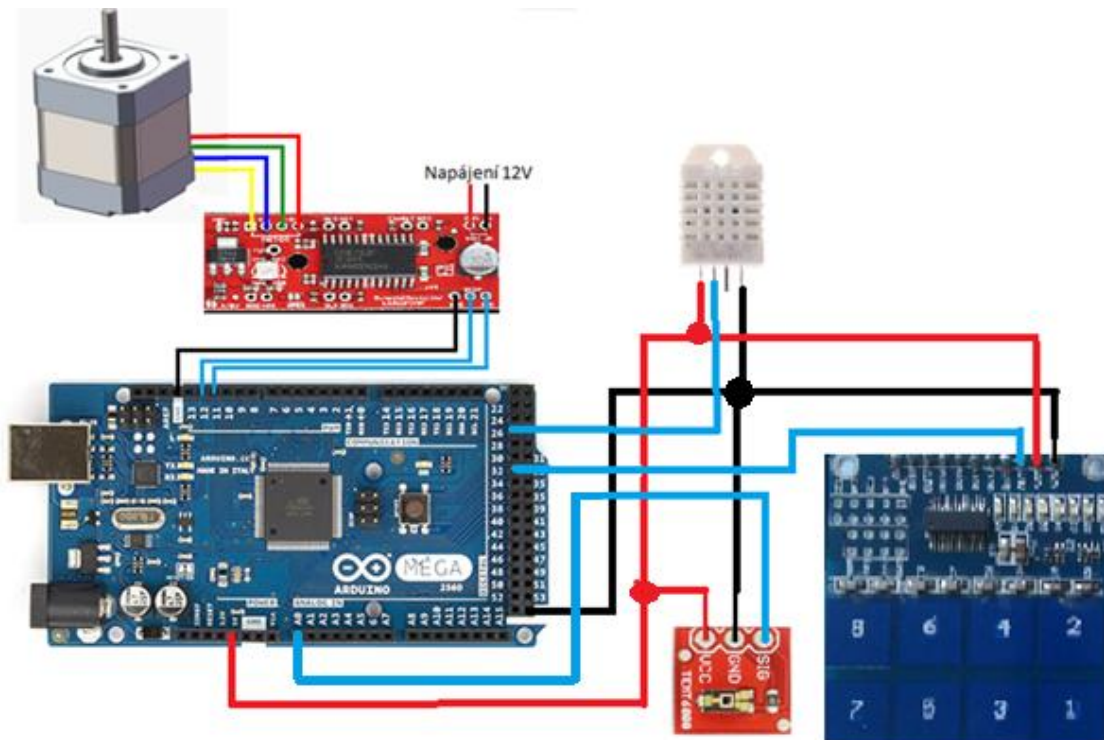
void setup() {
  Serial.begin(9600);
  pinMode(reléPin1, OUTPUT); // nastavení relé pinu jako výstup
  pinMode(buttonPin1, INPUT); } // Nastavení tlačítka pinu jako vstup
void loop() {
  buttonState4 = digitalRead(buttonPin4);
  if(buttonState4 == HIGH) { // Jestli je tlačítko 4 stisknuto
    digitalWrite(reléPin5, LOW); } // sepní relé 5 (LED světlo)
  else{
    if(analogRead(temt6000Pin) >= 700) { // Pokud hodnota světla je větší ...
      digitalWrite(reléPin5, HIGH); } // ... nebo rovna 700 tak sepní světlo
    if(analogRead(temt6000Pin) <= 400) { // Pokud je hodnota světla nižší ...
      digitalWrite(reléPin5, LOW); } } } // ... nebo rovna 400 tak rozepni světlo

```

9.5 Ovládání rolet

Ovládání rolet pomocí senzoru světla, senzoru teploty vzduchu a tlačítka.

Schéma zapojení:



Obrázek 39: Schéma zapojení ovládání rolet

Popis 13:

Zapojení motorku je stejné jako v popisu 7. Napájení senzoru teploty a vlhkosti vzduchu, modulu tlačítek a senzoru světla je připojeno červeným vodičem k Arduinu na napájecí pin 5V. GND senzoru teploty a vlhkosti vzduchu, modulu tlačítek a senzoru světla je připojeno černým vodičem k Arduinu na pin GND. Konektor data senzoru světla je připojen na analogový vstup A0. Konektor data senzoru teploty a vlhkosti vzduchu je připojen na digitální pin 26. Tlačítko 1 je připojeno na digitální pin 32.

Program:

```
#include <DHT.h> // Knihovna senzoru teploty DHT 22
#define DHT1PIN 26 // teplota 1
#define DHT2TYPE DHT22 // definuje typ senzoru
int temt6000Pin = A0; // senzor světla
int dirpin1 = 11; // motorek dir
int steppin1 = 12; // motorek step
const int buttonPin7 = 32; // kam je připojeno tlačítko
int buttonState1 = 0; // nastavení tlačítka jako 0
void setup() {
  Serial.begin(9600);
  pinMode(buttonPin7,INPUT); } // Nastavení tlačítka pinu jako vstup
void loop() {
  int i;
  buttonState7 = digitalRead(buttonPin7);
  if(roleta == false){ // Pokud je roleta zatažená
  if(buttonState7 == HIGH) { // a tlačítko stisknuto 7
  digitalWrite(dirpin1, LOW); // nastavení směru otáčení
  for (i = 0; i<12000; i++){ // otočíme rotor o 12 000 kroků – vytáhne rolety
  digitalWrite(steppin1, LOW);
  digitalWrite(steppin1, HIGH);
  delayMicroseconds(500); // nastavení rychlosti otáček motorku
  roleta = true; } } // rolety jsou vytažené
  else {
```



```

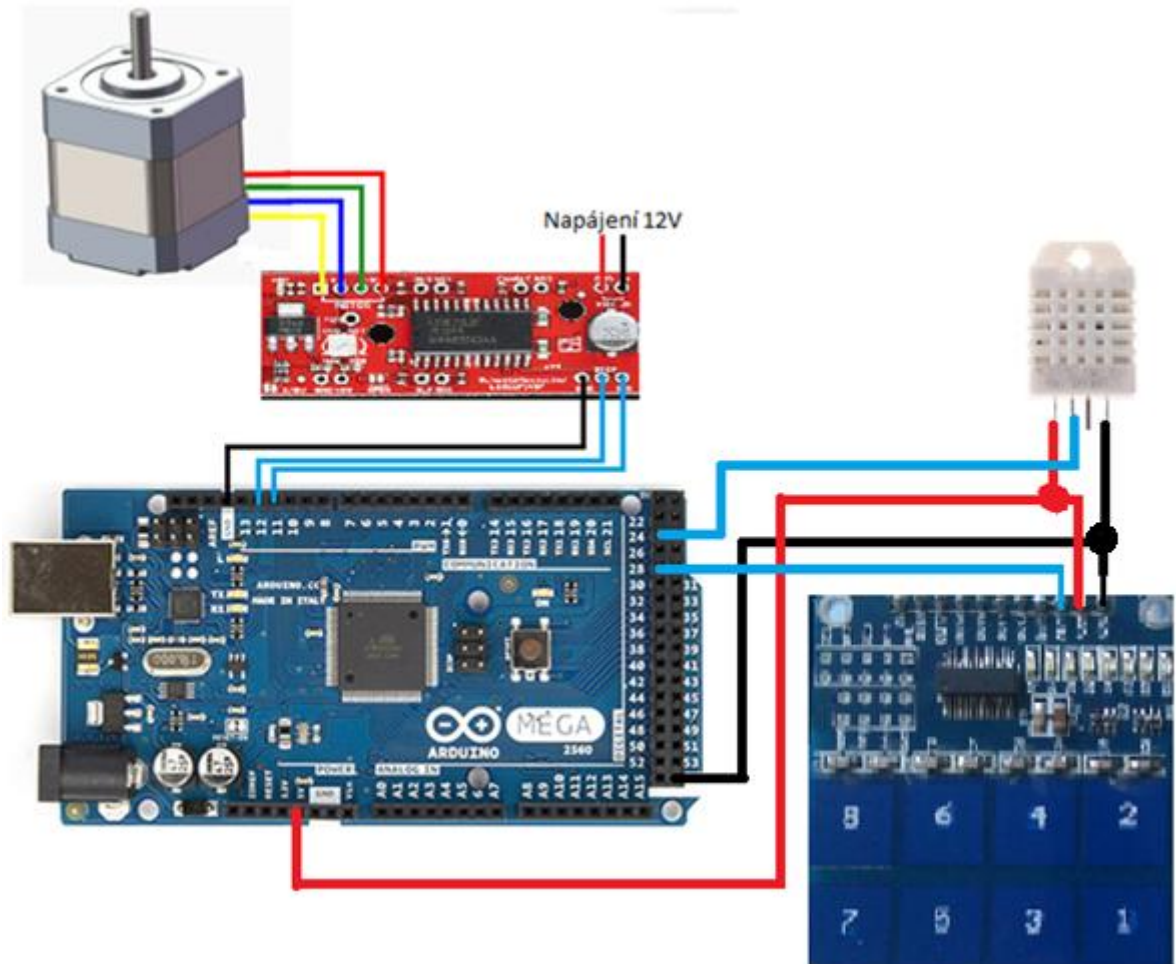
// Jestli je teplota větší nebo rovna 35°C a zároveň hodnota světla je větší nebo
rovna hodnotě 750
if(dht1.readTemperature() >= 35 && analogRead(temt6000Pin >= 750)) {
digitalWrite(dirpin1, LOW); // nastavení směru otáčení
for (i = 0; i<12000; I++){ // otočíme motorek o 12 000 kroků – vytáhne rolety
digitalWrite(stepin1, HIGH);
digitalWrite(stepin1, LOW);
delayMicroseconds(500); // rychlost otáček
roleta = true; } } } // rolety jsou vytažené
buttonState8 = digitalRead(buttonPin8);
if(roleta == true){ // Je-li roleta vytažená
if(buttonState8 == HIGH) { // a tlačítko 8 stisknuto
digitalWrite(dirpin1, HIGH); // nastavení směru otáčení
for (i = 0; i<12000; i++){ // otočení o 12 000 kroků – zatáhneme rolety
digitalWrite(stepin1, HIGH);
digitalWrite(stepin1, LOW);
delayMicroseconds(500); // rychlost otáček motorku
roleta = false; } } // roleta je zatažená
else {
// Jestli je teplota nižší nebo rovna 24°C a zároveň hodnota světla je nižší nebo
rovna 600
if(dht1.readTemperature() <= 24 && analogRead(temt6000Pin <= 600) ) {
digitalWrite(dirpin1, HIGH); // nastavení směru
for (i = 0; i<12000; i++){ // otočení o 12 000 kroků – zatáhne rolety
digitalWrite(stepin1, HIGH);
digitalWrite(stepin1, LOW);
delayMicroseconds(500); // rychlost otáček motorku
roleta = false; } } } } // roleta je zatažená

```

9.6 Ovládání oken

Ovládání oken pomocí teploty vzduchu nebo tlačítka.

Schéma zapojení:



Obrázek 40: Schéma zapojení ovládání oken

Popis 14:

Zapojení motorku je stejné jako v popisu 7. Konektor modulu tlačítek VCC je připojen k Arduino na pin 5V. Konektor modulu GND připojíme k Arduino na volný pin GND. Konektor data připojíme na libovolný nezabraný digitální pin. V našem případě pin 28. Senzor teploty a vlhkosti vzduchu zapojíme jako v popisu 4 s tím rozdílem, že data připojíme na digitální pin 24.

Program:

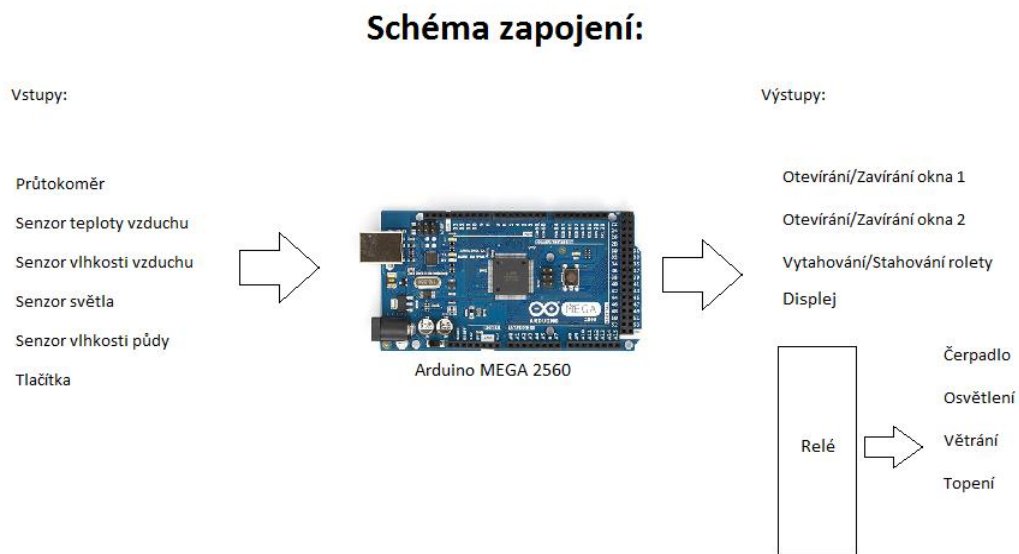
```
#include <DHT.h> // knihovna pro DHT
#define DHT1PIN 24 // Připojení teploty 1 na pin 24
#define DHT1TYPE DHT22 // typ senzoru
DHT dht1(DHT1PIN, DHT1TYPE);
int dirpin1 = 11; // motorek dir
int steppin1 = 12; // motorek step
const int buttonPin5 = 28; // kam je připojeno tlačítko
int buttonState1 = 0; // nastavení tlačítka jako 0
void setup() {
  Serial.begin(9600);
  dht1.begin(); // začni měřit teplotu a vlhkost
  pinMode(buttonPin5, INPUT); } // Nastavení tlačítka pinu jako vstup
void loop() {
  buttonState5 = digitalRead(buttonPin5);
  if(otevreno == false){ // Pokud je okno zavřené
    if(buttonState5 == HIGH) { // a tlačítko 5 stisknuté
      digitalWrite(dirpin2, LOW); // nastavení směru
      for (i = 0; i<4000; i++) { // otočení motorku o 4000 kroků – otevřeme okna
        digitalWrite(steppin2, LOW);
        digitalWrite(steppin2, HIGH);
        delayMicroseconds(500); // rychlost otáček
        otevreno = true; } // okna jsou otevřené
      } else {
        if(dht1.readTemperature() >= 30) } // Pokud je teplota vzduchu větší nebo
        rovna 30°C
        digitalWrite(dirpin2, LOW);
        for (i = 0; i<4000; i++); { // otočíme motorek o 4000 kroků - otevřeme okna
          digitalWrite(steppin2, HIGH);
          digitalWrite(steppin2, LOW);
          delayMicroseconds(500); // rychlost otáček
          otevreno = true; } } } }
  buttonState6 = digitalRead(buttonPin6);
```

```

if(otevreno == true){ // Pokud jsou okna otevřeny
if(buttonState6 == HIGH) { // a tlačítko 6 stisknuté
digitalWrite(dirpin2, HIGH); // nastavení směru otáčení
for (i = 0; i<4000; i++) { // otočíme motorek o 4000 kroků – zavřeme okna
digitalWrite(stepin2, HIGH);
digitalWrite(stepin2, LOW);
delayMicroseconds(500); // rychlost otáček
otevreno = false; } // okna jsou zavřena
} else {
if(dht1.readTemperature() <= 25) { // je-li teplota menší nebo rovna 25°C
digitalWrite(dirpin2, HIGH); // nastavení směru otáčení
for (i = 0; i<4000; i++){ // otočíme motorek o 4000 kroků – teplota zavře okna
digitalWrite(stepin2, HIGH
digitalWrite(stepin2, LOW);
delayMicroseconds(500); // rychlost otáček
otevreno = false; } } } // okna jsou zavřena

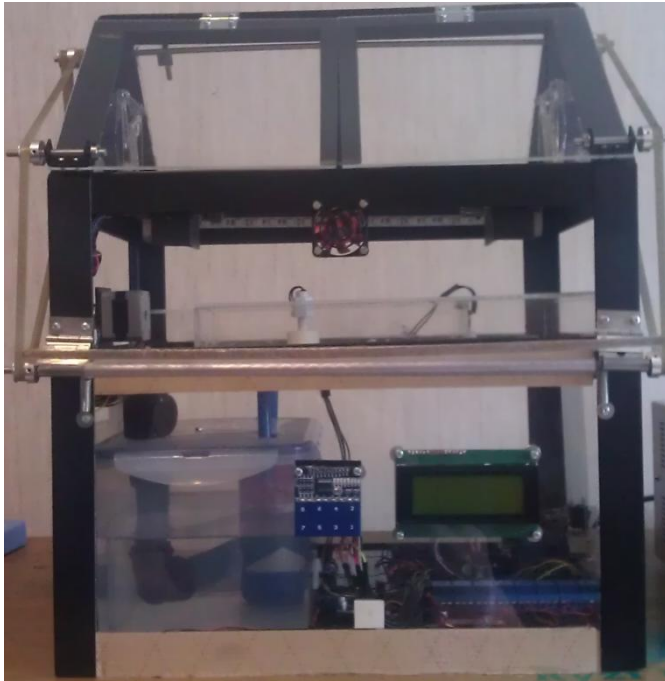
```

10 Schéma vzdáleného experimentu



Obrázek 41: Schéma zapojení: Návrh a konstrukce vzdáleného experimentu Inteligentní skleník

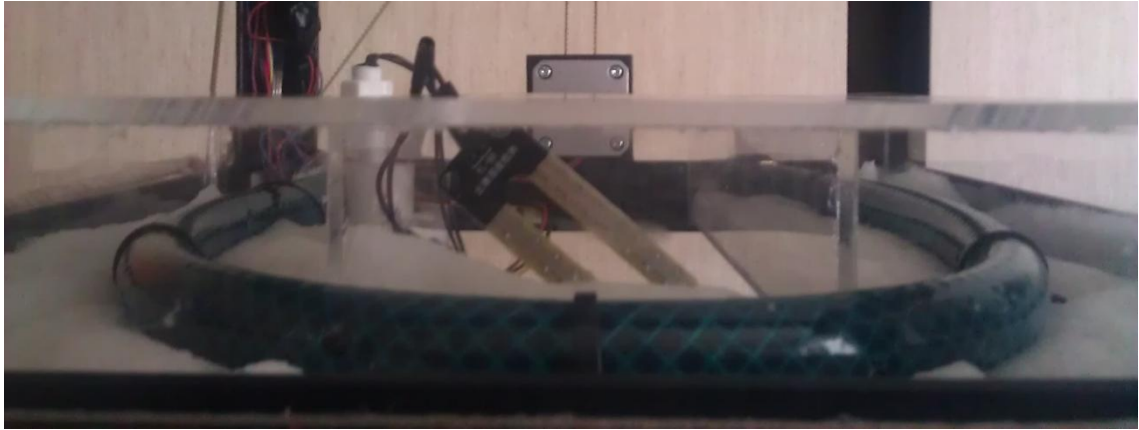
11 Vizualizace vzdáleného experimentu



Obrázek 42: Vizualizace skleníku - přední pohled



Obrázek 43: Vizualizace skleníku - boční pohled



Obrázek 44: Vizualizace skleníku – záhon



Obrázek 45: Vizualizace skleníku - roleta

12 Závěr

Cílem bakalářské práce bylo navrhnout a zkonstruovat vzdálený experiment – Inteligentní skleník. Experiment je zaměřený na použití vzdálených experimentů ve školních laboratořích.

Bakalářskou práci jsem rozdělil do dvou částí, teoretické a praktické. V první polovině teoretické části jsem se zaměřil na objasnění pojmu vzdálený experiment, vysvětluji jeho princip a uvádím jejich výhody a nevýhody. Dále popisuji technickou realizaci konkrétních vzdálených experimentů v České republice a charakterizuji jejich princip. V druhé polovině teoretické části jsem navrhl a zkonstruoval model inteligentního skleníku, který je připojený k internetu a vymezil jsem jeho konstrukční požadavky. Pro realizaci experimentu jsem použil stavebnici Arduino, která umožňuje snímání teploty a vlhkosti vzduchu, vlhkosti půdy, světla, průtoku kapaliny a snímání vodní hladiny. Tyto hodnoty jsou uvedeny na LCD displeji a na internetu. V první polovině praktické části popisuji zapojení jednotlivých součástek a jejich funkčnost. U každého příkladu zapojení je uveden kód s popisem. V druhé polovině praktické části jsou propojeny vstupy a výstupy experimentu, které slouží pro ovládání základních prvků ve skleníku. U každého ovládacího prvku je uveden kód s řádkovým popisem. Inteligentní skleník je ideální pomůckou při výuce technických předmětů, jak na základních, tak i středních školách. Žáci mohou zaznamenávat naměřená data přímo osobně z displeje modelu nebo vzdáleně pomocí internetu.

Vzdálený experiment – inteligentní skleník je možné díky Arduino systému dále rozšiřovat. Inteligentní skleník lze doplnit o zabezpečovací systém, dálkové řízení experimentu pomocí mobilní sítě GSM, snímání teploty zálivky nebo nainstalování nové nádoby s hnojivem, které by ponorné čerpadlo dodávalo rostlinám. Stejný princip lze uplatnit také, jako postřik rostlin proti vzniku plísní pomocí sprch, umístěných nad rostlinami.

13 Zdroje

1. AUER, M. E. (2001) *Virtual Lab versus Remote Lab. In: Proceedings of the 20th World Conference on Open Learning and Distance Education*. Düsseldorf, Germany.
2. DOSTÁL, J. *Elektrotechnické stavebnice (teorie a výsledky výzkumu)*. Olomouc: Votobia, 2008. 74 s. ISBN 978-80-7220-308-6.
3. ĎAĎO, Stanislav, BEJČEK a Antonín PLATIL. *Měření průtoku a výšky hladiny*. 1. vyd. Praha: BEN - technická literatura, 2005, 447 s. ISBN 80-730-0156-X.
4. GRÖBER, S.et al. (2008) Remotely controlled laboratories: *Aims, examples, and experience. American Journal of Physics*, 76 (4&5), pp. 374-378.
5. GROZMAN, Pavel. *Zavlažujeme zahradu*. 1. vyd. Praha : Grada, 2006, s. 111. ISBN 80-247-1663-1.
6. HAŠ, Stanislav. *Skleníky, jejich vlastnosti a vybavení*. 2. vyd. Praha : ÚZPI, 2004. 56 s. ISBN 80-7271-148-2.
7. HELLBERG, J., BÍLEK, M.: *K současnému stavu a vývojovým tendencím výuky chemii*. Hradec Králové : Gaudeamus, 2000, 143 s.
8. HRUŠKA, F. (2005). *Technické prostředky automatizace IV*. 3.vydání UTB Zlín ISBN: 80-7318-274-2.
9. LÁTAL, F.: *Vzdáleně ovládané experimenty na internetu*. MATEMATIKA – FYZIKA – INFORMATIKA, r. 20 (2010/2011), č. 6, s. 356-360. ISSN 1210-1761.
10. MAŇÁK, Josef a Martina HORÁKOVÁ. *Experiment v pedagogice*. Brno: Pedagogická knihovna Brno, 1994. ISBN 80-705-1076-5.
11. MASSIMO, Banzi. *Getting Started with Arduino*. Bolzano: Maker Media, Inc,2011, s.130. ISBN 978-1-4493-0987-9.
12. MECHLOVÁ, E. a kol. (1999) *Výkladový slovník fyziky pro základní vysokoškolský kurz*. Prometheus, Praha. ISBN 80-7196-151-5.
13. RIPKA, Pavel. *Senzory a převodníky*. 1. vyd. Praha : Vydavatelství ČVUT, 2005, s. 136. ISBN 80-01-03123-3.
14. SERAFÍN, Čestmír a Martin HAVELKA. *Elektrotechnické stavebnice*. 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2011, 78 s. ISBN 978-80-244-2834-5.
15. ŠRÁMEK, František. *Technická opatření k úspoře energie při vytápění skleníků*. Praha : Ústav zemědělských a potravinářských informací, 1995, s.27.
16. VIRIUS, Miroslav. *Jazyky C a C++ : kompletní průvodce*. 2. vyd. Praha : Grada, 2011, s. 367. ISBN 978-80-247-3917-5.

Internetové zdroje

17. DOSTÁL, Jiří. MULTIMEDIA, HYPERTEXT AND HYPERMEDIA TEACHING AIDS – A CURRENT TREND IN EDUCATION. *Journal of Technology and Information Education* [online]. 3/2009, Volume 1, Issue 2, [cit. 2015-03-12]. Dostupný z www: http://www.jtie.upol.cz/clanky_2_2009/multimedialni_hypertextove_a_hypermedialni_ucebni_pomucky.pdf ISSN 1803-537X.
18. DOSTÁL, J. 2013. *Experiment jako součást badatelsky orientované výuky*. Trends in Education. ISSN 1805-8949. [online] [cit. 2015-3-16] Dostupné z: http://www.kteiv.upol.cz/tvv_web/tvv13/tvv_2013_proceedings.pdf
19. DOSTÁL, J. *Zařazení elektrotechnických stavebnic do systému elektrostavebnic*. e-Pedagogium (on-line), 2004, roč. 4, č. 1. Dostupné na www: <http://epedagog.upol.cz/eped1.2004/index.htm>. ISSN 1213-7499
20. GRUNDFOS S.R.O. *Zavlažování I – typy závlah: Metody závlahy* [online]. 2013 [cit. 2015-02-11]. Dostupné z: <http://voda.tzb-info.cz/10194-zavlazovani-i-typy-zavlah>
21. KUSHNER, David. 2011. *The Making of Arduino*. [online]. [cit. 2014-04-15]. Dostupné z: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
22. LÁTAL, František. *Fyzikální princip*. [online]. 2011 [cit. 2015-04-13]. Dostupné z: http://ictphysics.upol.cz/remotelab/rlab5_princip.html
23. LÁTAL, František. *Vzdáleně ovládané experimenty ve výuce fyziky*. Olomouc, 2012. Dostupné z: http://theses.cz/id/wpnu70/Latal_disertace_vse.pdf. Disertační. Univerzita Palackého.
24. LUSTIG, František. *Veletrh 9* [online]. 2004 [cit. 2011-05-31]. *Jak si jednoduše postavit vzdálenou laboratoř na internetu*. Dostupné z WWW: http://kdf.mff.cuni.cz/veletrh/sbornik/Veletrh_09/09_19_Lustig.html.
25. LUSTIG, František. *Fyzikální základ: Elektromagnetická indukce* [online]. [cit. 2015-03-13]. Dostupné z: <http://kdt-20.karlov.mff.cuni.cz/ulohy/elmag.html#fyzikalnizaklad>
26. REICHL, Ladislav a Martin VŠETIČKA. *Displej z kapalných krystalů (LCD)* [online]. 2015 [cit. 2015-03-8]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/523-displej-z-kapalnych-krystalu-lcd>
27. ŠPRINGL, Vít. *Měření teploty - polovodičové odporové senzory teploty* [online]. [cit. 2006-5-1] Dostupný z WWW: <http://www.hw.cz/Teorie-a-praxe/Dokumentace/ART1141-Mereni-teploty---polovodicove-odporove-senzory-teploty.html>

14 Seznam obrázků

Obrázek 1: Schéma vzdáleného experimentu	13
Obrázek 2: Vzdálený experiment - Řízení výšky vodní hladiny	15
Obrázek 3: Vzdálený experiment - Tíhové zrychlení kmitu matematického kyvadla ...	16
Obrázek 4: Vzdálený experiment - Meteorologická stanice.....	16
Obrázek 5: Vzdálený experiment - Monitorování radioaktivního pozadí	17
Obrázek 6: Vzdálený experiment - Regulace otáček cívky.....	18
Obrázek 7: Arduino Mega2560	23
Obrázek 8: Senzor světla TEMT 6000	25
Obrázek 9: Průtokoměr YF - S201	25
Obrázek 10: Senzor vlhkosti půdy.....	27
Obrázek 11: Senzor teploty a vlhkosti vzduchu	28
Obrázek 12: LCD displej	29
Obrázek 13: Seriál LCD převodník	30
Obrázek 14: Snímač vodní hladiny.....	30
Obrázek 15: Elektromagnetické relé.....	31
Obrázek 16: Arduino relé modul	32
Obrázek 17: Vibrační senzor	32
Obrázek 18: Krokový motorek	33
Obrázek 19: Popis krokového motorku	34
Obrázek 20: Popis cívek krokového motorku.....	37
Obrázek 21: Ovladač krokového motorku.....	38
Obrázek 22: Ponorné čerpadlo.....	38
Obrázek 23: Ethernet shield.....	39
Obrázek 24: Modul tlačítek	40
Obrázek 25: Schéma zapojení senzoru světla.....	42
Obrázek 26: Schéma zapojení průtokoměru	43
Obrázek 27: Funkce attachInterrupt	43
Obrázek 28: Schéma zapojení senzoru vlhkosti půdy	46
Obrázek 29: Schéma zapojení senzoru teploty a vlhkosti vzduchu.....	47
Obrázek 30: Schéma zapojení relé.....	49
Obrázek 31: Schéma zapojení tlačítek a relé	50
Obrázek 32: Schéma zapojení krokového motorku	51
Obrázek 33: Schéma zapojení LCD displeje s ovladačem	53
Obrázek 34: Schéma zapojení ethernet shieldu	54
Obrázek 35: Schéma zapojení větrání	57
Obrázek 36: Schéma zapojení topení.....	59
Obrázek 37: Schéma zapojení čerpadla	60
Obrázek 38: Schéma zapojení osvětlení	62
Obrázek 39: Schéma zapojení ovládání rolet	63
Obrázek 40: Schéma zapojení ovládání oken	66
Obrázek 41: Schéma zapojení: Návrh a konstrukce vzdáleného experimentu Inteligentní skleníků.....	69

Obrázek 42: Vizualizace skleníku - přední pohled	70
Obrázek 43: Vizualizace skleníku - boční pohled	70
Obrázek 44: Vizualizace skleníku – záhon.....	71
Obrázek 45: Vizualizace skleníku - roleta.....	71

15 Seznam příloh

Příloha 1 - Výstup na obrazovku počítače

Příloha 2 - Celkový program

16 Seznam zkratek

3V3 – regulovaný výstup pro napájení externího obvodu

5V – regulovaný napájecí zdroj

AC – střídavý proud

DC – stejnosměrný proud

GND – „Ground“ zem obvodu

LCD – „liquid crystal display“, displej s kapalnými krystaly

LED – svítivá dioda

PWM – „Pulse Width Modulation“ simulace řízení výkonu u digitálních výstupů

SerLCD - Serial LCD, sériový převodník pro LCD display

STN - Super-twisted nematic, druh jednobarevné pasivní matrice displeje

TCP - „Transmission Control Protocol“ - primární přenosový protokol

UARTs - Sériové porty

UDP – „User Datagram Protocol“ - jeden ze sady protokolů internetu

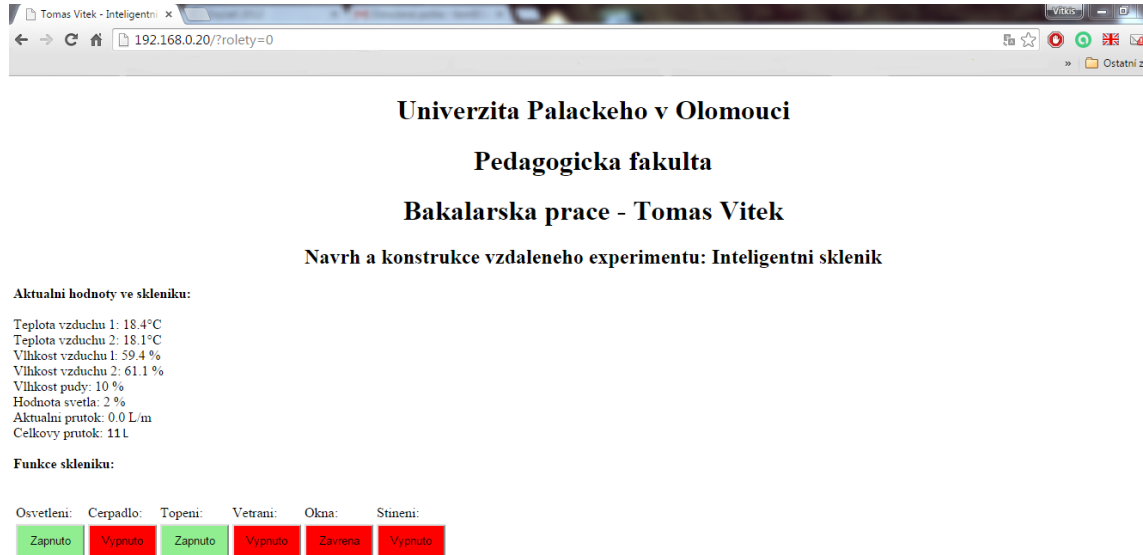
VCC – „Positive supply voltage“ - napájecí napětí

Vin – pin používaný při použití externího napájení

WiFi - Wireless Fidelity, komunikační standard pro bezdrátový přenos dat

Přílohy

Příloha 1: Výstup na obrazovku počítače



Příloha 2: Celkový program

```
1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include <DHT.h> // Knihovna senzoru teploty DHT 22
4 #define DHT1PIN 13 // teplota 1
5 #define DHT2PIN 12 // teplota 2
6 #define DHT1TYPE DHT22 // definuje typ senzoru
7 #define DHT2TYPE DHT22 // definuje typ senzoru
8 DHT dht1(DHT1PIN, DHT1TYPE); // definuje typ senzoru
9 DHT dht2(DHT2PIN, DHT2TYPE); // definuje typ senzoru
10 int temt6000Pin = A0; // senzor svetla
11 int dirpin1 = 5; // motorek 1 = okna dir
12 int steppin1 = 4; // motorek 3 = okna step
13 int dirpin2 = 8; // motorek 2 = okna dir
14 int steppin2 = 9; // motorek 2 = okna step
15 int dirpin3 = 6; // motorek 3 = rolety dir
16 int steppin3 = 7; // motorek 3 = rolety step
17 int i; // motorek otáčení
18 byte sensorInterrupt = 0; // začátek průtokoměr
19 byte sensorPin = 2;
20 float calibrationFactor = 4.5;
21 volatile byte pulseCount;
```

```

22 float flowRate;
23 unsigned int flowMilliLitres;
24 unsigned long totalMilliLitres;
25 unsigned long oldTime; // konec průtokoměr
26 const int buttonPin1 = 35; // tlačítko 1 - vetrani
27 const int buttonPin2 = 37; // tlačítko 2 - topeni
28 const int buttonPin3 = 39; // tlačítko 3 - čerpadlo
29 const int buttonPin4 = 41; // tlačítko 4 - led svetlo
30 const int buttonPin5 = 43; // tlačítko 5 - motorek 2 a 3 = okna - dolů
31 const int buttonPin6 = 45; // tlačítko 6 - motorek 2 a 3 = okna – nahoru
32 const int buttonPin7 = 47; // tlačítko 7 - motorek 1 = rolety - dolů
33 const int buttonPin8 = 49; // tlačítko 8 - motorek 1 = rolety - nahoru
34 const int relePin1 = 22; // rele 1 - vetrak 1
35 const int relePin2 = 24; // rele 2 - vetrak 2
36 const int relePin3 = 26; // rele 3 - topeni
37 const int relePin4 = 28; // rele 4 - čerpadlo
38 const int relePin5 = 30; // rele 5 - LED svetlo
39 int buttonState1 = 0; // tlacitko 1 = 0
40 int buttonState2 = 0; // tlacitko 2 = 0
41 int buttonState3 = 0; // tlacitko 3 = 0
42 int buttonState4 = 0; // tlacitko 4 = 0
43 int buttonState5 = 0; // tlacitko 5 = 0
44 int buttonState6 = 0; // tlacitko 6 = 0
45 int buttonState7 = 0; // tlacitko 7 = 0
46 int buttonState8 = 0; // tlacitko 8 = 0
47 boolean otevreno = false; // nastavení oken jako zavreno
48 boolean roleta = false; // nastavení rolety jako zavreno
49 unsigned long previousMillis = 0;
50 unsigned long interval = 4000; // interval vypisování LCD
51 byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xD3, 0xB9 }; // mac adresa
Ethernet shieldu
52 IPAddress ip(192, 168, 0, 20); // IP adresa
53 IPAddress dnserver(192, 168, 0, 1); // DNS server
54 IPAddress gateway(192, 168, 0, 1); // Výchozí brána
55 EthernetServer server(80); // port 80
56 //-----
57 void setup() {
58 Serial.begin(9600);
59 Ethernet.begin(mac, ip, dnserver, gateway); // inicializace serveru
60 setupDisplay();
61 pinMode(dirpin1, OUTPUT); // motorek 1 rolety vystup
62 pinMode(stepin1, OUTPUT); // motorek 1 rolety vystup
63 pinMode(dirpin2, OUTPUT); // motorek 2 = okna vystup
64 pinMode(stepin2, OUTPUT); // motorek 2 = okna vystup

```



```

65     pinMode(dirpin3, OUTPUT);           // motorek 3 = okna vystup
66     pinMode(steppin3, OUTPUT);        // motorek 3 = okna vystup
67     pinMode(relePin1, OUTPUT);        // rele 1 - vystup vetrak 1
68     pinMode(relePin2, OUTPUT);        // rele 2 - vystup vetrak 2
69     pinMode(relePin3, OUTPUT);        // rele 3 - vystup topeni
70     pinMode(relePin4, OUTPUT);        // rele 4 - vystup čerpadlo
71     pinMode(relePin5, OUTPUT);        // rele 5 - vystup LED svetlo
72     pinMode(buttonPin1,INPUT);        // tlacitko 1 vstup
73     pinMode(buttonPin2,INPUT);        // tlacitko 2 vstup
74     pinMode(buttonPin3,INPUT);        // tlacitko 3 vstup
75     pinMode(buttonPin4,INPUT);        // tlacitko 4 vstup
76     pinMode(buttonPin5,INPUT);        // tlacitko 5 vstup
77     pinMode(buttonPin6,INPUT);        // tlacitko 6 vstup
78     pinMode(buttonPin7,INPUT);        // tlacitko 7 vstup
79     pinMode(buttonPin8,INPUT);        // tlacitko 8 vstup
80     pinMode(sensorPin, INPUT);        //zacatek prutokomer
81     digitalWrite(sensorPin, HIGH);
82     pulseCount      = 0;
83     flowRate        = 0.0;
84     flowMilliLitres = 0;
85     totalMilliLitres = 0;
86     oldTime         = 0;
87     attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
88     // konec prutokomer
89     dht1.begin();           // zacni merit teplotu 1
90     dht2.begin();           // zacni merit teplotu 2
91     }
92     void loop() {
93     // když je připojen klient, zjistíme, zda jsou odeslána data od klienta směrem k
nám
94     EthernetClient client = server.available();
95     String buffer = "";           // vyrovnávací paměť pro tlačítka
96     unsigned int frac;           // průtokoměr
97     while (client.connected()) { // dokud je klient připojen
98         if (client.available()) { // čti data od něj dokud nenarazíš na znak
nového řádku
99             char c = client.read();
100            buffer = buffer + c;
101            if (c == '\n') {
102            // Propojení tlačítek na webu s relé
103                if(buffer.indexOf("osvetleni=1")>=0)
104                    digitalWrite(relePin5,LOW);
105
106                if(buffer.indexOf("osvetleni=0")>=0)

```

```

107     digitalWrite(relePin5,HIGH);
108
109     if(buffer.indexOf("cerpadlo=1")>=0)
110         digitalWrite(relePin4,LOW);
111
112     if(buffer.indexOf("cerpadlo=0")>=0)
113         digitalWrite(relePin4,HIGH);
114
115     if(buffer.indexOf("topeni=1")>=0)
116         digitalWrite(relePin3,LOW);
117
118     if(buffer.indexOf("topeni=0")>=0)
119         digitalWrite(relePin3,HIGH);
120
121     if(buffer.indexOf("vetrani=1")>=0)
122         digitalWrite(relePin1,LOW);
123         digitalWrite(relePin2,LOW);
124
125     if(buffer.indexOf("vetrani=0")>=0)
126         digitalWrite(relePin1,HIGH);
127         digitalWrite(relePin2,HIGH);
128
129     // -----Propojení tlačítek webu s motorky---okna-----
130         if(buffer.indexOf("okna=1")>=0) {
131             if(otevreno == false){
132                 digitalWrite(dirpin1, HIGH);           // nastaveni smeru
133                 digitalWrite(dirpin2, LOW);           // nastaveni smeru
134                 for (i = 0; i<1000; i++) {           // otočení o 4000 kroků
135                     digitalWrite(stepin1, LOW);
136                     digitalWrite(stepin2, LOW);
137                     delay(1);
138                     digitalWrite(stepin1, HIGH);
139                     digitalWrite(stepin2, HIGH);
140                     delayMicroseconds(500);         // rychlost otáček
141                 }
142                 otevreno = true;
143             }
144         }
145         if(buffer.indexOf("okna=0")>=0) {
146             if(otevreno == true){
147                 digitalWrite(dirpin1, LOW);           // nastaveni smeru
148                 digitalWrite(dirpin2, HIGH);          // nastaveni smeru
149                 for (i = 0; i<1000; i++)             // otočení o 4000 kroků
150                 {

```

```

151         digitalWrite(steppin1, HIGH);
152         digitalWrite(steppin2, HIGH);
153         delay(1);
154         digitalWrite(steppin1, LOW);
155         digitalWrite(steppin2, LOW);
156         delayMicroseconds(500);        // rychlost otáček
157     }
158     otevreno = false;
159     }
160 }
161 // -----Propojení tlačítek webu s motorkem--rolety-----
162     int i;
163     if(buffer.indexOf("rolety=1")>=0) {
164         if(roleta == false){
165             digitalWrite(dirpin3, LOW);    // nastaveni smeru
166             for (i = 0; i<12000; i++){    // otočení o 12 000 kroků
167                 digitalWrite(steppin3, LOW);
168                 delay(1);
169                 digitalWrite(steppin3, HIGH);
170                 delayMicroseconds(500);    // rychlost otáček
171                 roleta = true;
172             }
173         }
174     }
175     if(buffer.indexOf("rolety=0")>=0) {
176         if(roleta == true){
177             digitalWrite(dirpin3, HIGH);    // nastaveni smeru
178             for (i = 0; i<12000; i++){    // otočení o 12 000 kroků
179                 digitalWrite(steppin3, HIGH);
180                 delay(1);
181                 digitalWrite(steppin3, LOW);
182                 delayMicroseconds(500);    // rychlost otáček
183                 roleta = false;
184             }
185         }
186     }
187 // odešle klasickou hlavičku HTML stránky
188     client.println("HTTP/1.1 200 OK");
189     client.println("Content-Type: text/html");
190     client.println("Connection: close");
191     client.println("Refresh: 60");
192     client.println();
193     client.println("<!DOCTYPE HTML>");
194

```

```

195 // zde již klasická html data
196     client.println("<html>");
197     client.println("<head>");
198     // Název stránky
199     client.println("<title>Tomas Vitek - Inteligentni sklenik</title>");
200     client.println("</head>");
201     client.println("<body>");
202 // Text
203     client.print("<div style='text-align:center'>");
204     client.print("<H1>Univerzita Palackeho v Olomouci</H1>");
205     client.print("<H1>Pedagogicka fakulta</H1>");
206     client.print("<H1>Bakalarska prace - Tomas Vitek</H1>");
207     client.print("<H2>Navrh a konstrukce vzdaleneho experimentu:
Inteligentni sklenik</H2>");
208     client.print("</div>");
209     client.print("<b>Aktualni hodnoty ve skleniku:</b>");
210     client.print("<br><br>");
211     client.print("Teplota vzduchu 1: ");
212     client.print(dht1.readTemperature(),1);
213     client.print((char)176); // vypíše znak °
214     client.print("C");
215     client.print("<br>");
216     client.print("Teplota vzduchu 2: ");
217     client.print(dht2.readTemperature(),1);
218     client.print((char)176); // vypíše znak °
219     client.print("C");
220     client.print("<br>");
221     client.print("Vlhkost vzduchu 1: ");
222     client.print(dht1.readHumidity(),1);
223     client.print(" %");
224     client.print("<br>");
225     client.print("Vlhkost vzduchu 2: ");
226     client.print(dht2.readHumidity(),1);
227     client.print(" %");
228     client.print("<br>");
229     client.print("Vlhkost pudy: ");
230     client.print(analogRead(A1)/11,1); // převedení hodnoty na %
231     client.print(" %");
232     client.print("<br>");
233     client.print("Hodnota svetla: ");
234     client.print(analogRead(temt6000Pin)/11,1); // převedení hodnoty na %
235     client.print(" %");
236     client.print("<br>");
237     client.print("Aktualni prtok: ");

```

```

238     client.print(int(flowRate));
239     client.print(".");
240     frac = (flowRate - int(flowRate)) * 10;
241     client.print(frac, DEC) ;
242     client.print(" L/m");
243     client.print("</br>");
244     client.print("Celkovy prtok: ");
245     client.print(totalMilliLitres/1000);
246     client.println(" L");
247     client.print("</br>");
248     client.print("</br>");
249     client.print("<b>Funkce skleniku: </b>");
250     client.print("</br>");
251     client.print("<table>");
252     client.print("<tr><td>Osvetleni:</td><td>Cerpadlo:</td><td>Topeni:</td><td>
Vetrani:</td><td>Okna:</td><td>Stineni:</td></tr>");
253     client.print("</br>");
254     client.print("<tr>");
255     //Tlačítka webu
256     //-----osvětlení-----
257     if (digitalRead(relePin5)){
258     client.print("<td width='80'><a
href='http://192.168.0.20?osvetleni=0'><button
style='background:red;width:100%;height:40px'>Vypnuto</button></a></td>");
259     }else{
260     client.print("<td width='80'><a
href='http://192.168.0.20?osvetleni=1'><button
style='background:lightgreen;width:100%;height:40px'>Zapnuto</button></a><
/td>");
261     }
262     //-----čerpadlo-----
263     if (digitalRead(relePin4)){
264     client.print("<td width='80'><a
href='http://192.168.0.20?cerpadlo=0'><button
style='background:red;width:100%;height:40px'>Vypnuto</button></a></td>");
265     }else{
266     client.print("<td width='80'><a
href='http://192.168.0.20?cerpadlo=1'><button
style='background:lightgreen;width:100%;height:40px'>Zapnuto</button></a><
/td>");
267     }
268     //-----topeni-----
269     if (digitalRead(relePin3)){

```

```

270         client.print("<td width='80'><a
href='http://192.168.0.20?topeni=0'><button
style='background:red;width:100%;height:40px'>Vypnuto</button></a></td>");
271     }else{
272         client.print("<td width='80'><a
href='http://192.168.0.20?topeni=1'><button
style='background:lightgreen;width:100%;height:40px'>Zapnuto</button></a><
/td>");
273     }
274 //-----větraní-----
275     if (digitalRead(relePin1) & (relePin2)){
276         client.print("<td width='80'><a
href='http://192.168.0.20?vetrani=0'><button
style='background:red;width:100%;height:40px'>Vypnuto</button></a></td>");
277     }else{
278         client.print("<td width='80'><a
href='http://192.168.0.20?vetrani=1'><button
style='background:lightgreen;width:100%;height:40px'>Zapnuto</button></a><
/td>");
279     }
280 //-----okna-----
281     if (digitalRead(relePin2)){
282         client.print("<td width='80'><a
href='http://192.168.0.20?okna=0'><button
style='background:red;width:100%;height:40px'>Zavrena</button></a></td>");
283     }else{
284         client.print("<td width='80'><a
href='http://192.168.0.20?okna=1'><button
style='background:lightgreen;width:100%;height:40px'>Otevrena</button></a>
</td>");
285     }
286 //-----rolety-----
287     if (digitalRead(relePin2)){
288         client.print("<td width='80'><a
href='http://192.168.0.20?rolety=0'><button
style='background:red;width:100%;height:40px'>Vypnuto</button></a></td>");
289     }else{
290         client.print("<td width='80'><a
href='http://192.168.0.20?rolety=1'><button
style='background:lightgreen;width:100%;height:40px'>Zapnuto</button></a><
/td>");
291     }
292     client.print("</tr>");
293     client.println("</br>");

```

```

294     client.println("</body>");
295     client.println("</html>");
296 //ukončí přenos
297     client.stop();
298     Serial.println(buffer);
299     }
300 }
301 }
302 //-----průtokoměr-----
303 if((millis() - oldTime) > 1000){
304     detachInterrupt(sensorInterrupt);
305     flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
306     oldTime = millis();
307     flowMilliLitres = (flowRate / 60) * 1000;
308     totalMilliLitres += flowMilliLitres;
309     unsigned int frac;
310     pulseCount = 0;
311     attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
312 }
313 //-----vetrak1-----
314     buttonState1 = digitalRead(buttonPin1);
315     if(buttonState1 == HIGH) {
316         digitalWrite(relePin1, LOW);
317     }else{
318         if(dht1.readTemperature() >= 26) {
319             digitalWrite(relePin1, LOW);
320         }
321         if(dht1.readTemperature() <= 23){
322             digitalWrite(relePin1, HIGH);
323         }
324     }
325 //-----vetrak2-----
326     buttonState1 = digitalRead(buttonPin1);
327         if(buttonState1 == HIGH) {
328             digitalWrite(relePin2, LOW);
329         }else{
330             if(dht1.readTemperature() >= 26) {
331                 digitalWrite(relePin2, LOW);
332             }
333             if(dht1.readTemperature() <= 23){
334                 digitalWrite(relePin2, HIGH);
335             }
336         }
337 //-----topeni-----

```

```

338     buttonState2 = digitalRead(buttonPin2);
339         if(buttonState2 == HIGH) {
340             digitalWrite(relePin3, LOW);
341         }else{
342             if(dht1.readTemperature() >= 22) {
343                 digitalWrite(relePin3, HIGH);
344             }
345             if(dht1.readTemperature() <= 16){
346                 digitalWrite(relePin3, LOW);
347             }
348         }
349 //-----cerpadlo -----
350     buttonState3 = digitalRead(buttonPin3);
351     if(buttonState3 == HIGH) {
352         digitalWrite(relePin4, LOW);
353     }
354 //-----LED svetlo -----
355     buttonState4 = digitalRead(buttonPin4);
356     if(buttonState4 == HIGH) {
357         digitalWrite(relePin5, LOW);
358     }else{
359         if(analogRead(temt6000Pin) >= 700) {
360             digitalWrite(relePin5, HIGH);
361         }
362         if(analogRead(temt6000Pin) <= 100) {
363             digitalWrite(relePin5, LOW);
364         }
365     }
366 //motorek 3---rolety-----
367 //tlačítko 8 směr nahoru
368     int i;
369     buttonState8 = digitalRead(buttonPin8);
370     if(roleta == false){           //pokud je roleta zatažená můžeme jí vytahnout
371         if(buttonState8 == HIGH) {           // pokud je stisknuto tlačítko
372             digitalWrite(dirpin3, LOW);           // nastavení smeru
373             for (i = 0; i<12000; i++){           // otočení o 12 000 kroků
374                 digitalWrite(stepPin3, LOW);
375                 delay(1);
376                 digitalWrite(stepPin3, HIGH);
377                 delayMicroseconds(500);           // rychlost otáček
378                 roleta = true;           // roleta - vytažena
379             }
380         } else {
381 //automatické vytažení pomocí teploty a vlhkosti vzduchu

```



```

382     if(dht1.readTemperature() >= 35 && analogRead(temt6000Pin >= 950)) {
383         digitalWrite(dirpin3, LOW);           // nastaveni smeru
384         for (i = 0; i<12000; i++){           // otočení o 12 000 kroků
385             digitalWrite(stepin3, HIGH);
386             delay(1);
387             digitalWrite(stepin3, LOW);
388             delayMicroseconds(500);         // rychlost otáček
389             roleta = true;                  // roleta - vytažena
390         }
391     }
392 }
393 }
394 //tlačítko 7 směr dolů
395     buttonState7 = digitalRead(buttonPin7);
396     if(roleta == true){                     // pokud je roleta vytažená můžeme ji stáhnout
397         if(buttonState7 == HIGH) {         // pokud je stisknuto tlačítko
398             digitalWrite(dirpin3, HIGH);   // nastaveni smeru
399             for (i = 0; i<12000; i++){     // otočení o 12 000 kroků
400                 digitalWrite(stepin3, HIGH);
401                 delay(1);
402                 digitalWrite(stepin3, LOW);
403                 delayMicroseconds(500);   // rychlost otáček
404                 roleta = false;
405             }
406         } else {
407 //automatické zatažení pomocí teploty a vlhkosti vzduchu
408             if(dht1.readTemperature() <= 20 && analogRead(temt6000Pin <= 500) ) {
409                 digitalWrite(dirpin3, HIGH);           // nastaveni smeru
410                 for (i = 0; i<12000; i++){           // otočení o 12 000 kroků
411                     digitalWrite(stepin3, HIGH);
412                     delay(1);
413                     digitalWrite(stepin3, LOW);
414                     delayMicroseconds(500);         // rychlost otáček
415                     roleta = false;
416                 }
417             }
418         }
419     }
420 //-----otevirani a zavirani oken-----
421     buttonState5 = digitalRead(buttonPin5);
422     if(otevreno == false){
423         if(buttonState5 == HIGH) {
424             digitalWrite(dirpin1, HIGH);           // nastaveni smeru
425             digitalWrite(dirpin2, LOW);

```

```

426         for (i = 0; i<1000; i++) {           // otočení o 1000 kroků
427             digitalWrite(stepPin1, LOW);
428             digitalWrite(stepPin2, LOW);
429             delay(1);
430             digitalWrite(stepPin1, HIGH);
431             digitalWrite(stepPin2, HIGH);
432             delayMicroseconds(500);        // rychlost otáček
433         }
434         otevreno = true;    // okna otevřena
435     }
436 } else {
437 // automatické otevření oken
438     if(dht1.readTemperature() >= 30) {
439         digitalWrite(dirPin1, LOW);
440         digitalWrite(dirPin2, HIGH);
441         for (i = 0; i<1000; i++) {         // otočení o 1000 kroků
442             digitalWrite(stepPin1, HIGH);
443             digitalWrite(stepPin2, HIGH);
444             delay(1);
445             digitalWrite(stepPin1, LOW);
446             digitalWrite(stepPin2, LOW);
447             delayMicroseconds(500);        // rychlost otáček
448         }
449         otevreno = true;    // okna otevřena
450     }
451 }
452 buttonState6 = digitalRead(buttonPin6);
453 if(otevreno == true){
454     if(buttonState6 == HIGH) {
455         digitalWrite(dirPin1, LOW);
456         digitalWrite(dirPin2, HIGH);
457         for (i = 0; i<1000; i++)          // otočení o 4000 kroků
458         {
459             digitalWrite(stepPin1, HIGH);
460             digitalWrite(stepPin2, HIGH);
461             delay(1);
462             digitalWrite(stepPin1, LOW);
463             digitalWrite(stepPin2, LOW);
464             delayMicroseconds(500);        // rychlost otáček
465         }
466     }
467     otevreno = false; // okna zavřena
468 }
469 } else {

```

```

470 // automatické zavření oken
471     if(dht1.readTemperature() <= 20) {
472         digitalWrite(dirpin1, LOW);
473         digitalWrite(dirpin1, HIGH);
474         for (i = 0; i<1000; i++){
475             digitalWrite(stepin1, HIGH);
476             digitalWrite(stepin2, HIGH);
477             delay(1);
478             digitalWrite(stepin1, LOW);
479             digitalWrite(stepin2, LOW);
480             delayMicroseconds(500); // rychlost otáček
481         }
482     }
483     otevreno = false;
484 }
485 //-----lcd-----
486 if(millis() - previousMillis > interval) {
487     previousMillis = millis();
488     clearLCD();
489     backlightOn();
490     selectLineOne();
491     Serial.print("T1:");
492     Serial.print(dht1.readTemperature(),1);
493     Serial.print((char)223); // vypíše znak °
494     Serial.print("C ");
495     Serial.print("T2:");
496     Serial.print(dht2.readTemperature(),1);
497     Serial.print((char)223); // vypíše znak °
498     Serial.print("C");
499     //-----
500     selectLineTwo();
501     Serial.print("V1:");
502     Serial.print(dht1.readHumidity(),1);
503     Serial.print("% ");
504     Serial.print("V2:");
505     Serial.print(dht2.readHumidity(),1);
506     Serial.print("%");
507     //-----
508     selectLineThree();
509     Serial.print("VP:");
510     Serial.print(analogRead(A1)/11,1); // převedení hodnoty na %
511     Serial.print("% ");
512     Serial.print("S:");
513     Serial.print(analogRead(temt6000Pin)/11,1); // převedení hodnoty na %

```

```

514     Serial.print("%");
515     //-----
516     selectLineFour();
517     Serial.print("P:");
518     Serial.print(int(flowRate));
519     Serial.print(".");
520     frac = (flowRate - int(flowRate)) * 10;
521     Serial.print(frac, DEC) ;
522     Serial.print("L/m ");
523     Serial.print("Pc:");
524     Serial.print(totalMilliLitres)/1000;
525     Serial.println(" L ");
526 }
527 }
528 //SerialLCD Funkce
529 void setupDisplay(){ // nastaveni displeje na 4x20
530     Serial.write(0x7C);
531     Serial.write(3);
532 }
533 void selectLineOne(){
534     Serial.write(0xFE);
535     Serial.write(128);
536 }
537 void selectLineTwo(){
538     Serial.write(0xFE);
539     Serial.write(192);
540 }
541 void selectLineThree(){
542     Serial.write(0xFE);
543     Serial.write(148);
544 }
545 void selectLineFour(){
546     Serial.write(0xFE);
547     Serial.write(212);
548 }
549 void goTo(int position) {
550 //pozice = řádek 1: 0-19, řádek 2: 20-39, atd, 79+ zpět na 0
551 if (position<20){ Serial.write(0xFE);
552     Serial.write((position+128)); // pozice
553 }else if (position<40){ Serial.write(0xFE);
554     Serial.write((position+128+64-20)); // pozice
555 }else if (position<60){ Serial.write(0xFE);
556     Serial.write((position+128+20-40)); // pozice
557 }else if (position<80){ Serial.write(0xFE);

```

```

558         Serial.write((position+128+84-60)); //pozice
559     } else { goTo(0); }
560 }
561 void clearLCD(){
562     Serial.write(0xFE)
563     Serial.write(0x01);
564 }
565 void backlightOn(){
566     Serial.write(0x7C);
567     Serial.write(157);
568 }
569 void backlightOff(){
570     Serial.write(0x7C);
571     Serial.write(128);
572 }
573 void backlight50(){
574     Serial.write(0x7C);
575     Serial.write(143);
576 }
577 void serCommand(){
578     Serial.write(0xFE);
579 }
580 // -----senzor-světla-----
581 void svetlo(){
582     int value = analogRead(temt6000Pin);
583 }
584 //-----vlhkost-půdy-----
585 void vlhkostpudy(){
586     int sensorValue = analogRead(A2);
587 }
588 //-----teplota 1,2 --- vlhkost 1,2 -----
589 void teplotavlhkost(){
590     float h1 = dht1.readHumidity();
591     float t1 = dht1.readTemperature();
592     float h2 = dht2.readHumidity();
593     float t2 = dht2.readTemperature();
594     if (isnan(t1) || isnan(h1)) {
595         Serial.println("Chyba čtení z DHT #1");
596     }
597     if (isnan(t2) || isnan(h2)) {
598         Serial.println("Chyba čtení z DHT #2");
599     }
600 }
601 //-----prutokomer-----

```

```
602 void pulseCounter()
603 {
604     pulseCount++;
605 }
```

ANOTACE

Jméno a příjmení:	Tomáš Vítek
Katedra:	Katedra technické a informační výchovy
Vedoucí práce:	PhDr. PaedDr. Jiří Dostál, Ph.D.
Rok obhajoby:	2015

Název práce:	Návrh a konstrukce vzdáleného experimentu – Inteligentní skleník
Název v angličtině:	Design and construction of remote experiment - The intelligent greenhouse
Anotace práce:	Bakalářská práce je zaměřena na návrh a konstrukci vzdálených experimentů ve školních laboratořích. V teoretické části je vymezen pojem vzdálený experiment, jeho výhody a nevýhody. Dále jsou popsány konstrukční požadavky na inteligentní skleník. Samostatnost skleníku je realizována pomocí automatizačního systému Arduino. V praktické části je realizován vzdálený experiment – inteligentní skleník.
Klíčová slova:	vzdálený experiment, skleník, návrh, konstrukce, automatizace, Arduino
Anotace v angličtině:	This bachelor thesis is focused on the design and construction of remote experiments in school laboratories. The theoretical part defines the notion of remote experiment, its advantages and disadvantages. Furthermore, the construction requirements of an intelligent greenhouse are described. The greenhouse's independence is realized by using the arduino automation system. In the practical part, the remote experiment - intelligent greenhouse - is implemented.
Klíčová slova v angličtině:	remote experiment, greenhouse, design, construction, automation, arduino
Přílohy vázané v práci:	CD-ROM
Rozsah práce:	78
Jazyk práce:	Český jazyk