



**Vyšší odborná škola  
a Střední průmyslová škola elektrotechnická  
Plzeň, Koterovská 85**

## **DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU**

**Téma: „Chytrý skleník“**

<b>Autor práce:</b>	<b>Jakub Ryba</b>
<b>Třída:</b>	<b>4. P</b>
<b>Vedoucí práce:</b>	<b>Jan Drexler</b>
<b>Dne:</b>	<b>29. 4. 2022</b>
<b>Hodnocení:</b>	



Vyšší odborná škola

a Střední průmyslová škola elektrotechnická  
Plzeň, Koterovská 85

## Zadání dlouhodobé maturitní práce

Žák: Jakub Ryba  
Třída: 4. P  
Studijní obor: 26-41-M/01  
Zaměření: Internet věci  
Školní rok: 2021 - 2022

Téma práce: ***Chytrý skleník***

***Pokyny k obsahu a rozsahu práce:***

1. Návrh konstrukce skleníku a uložení všech funkčních částí
2. Testování funkčnosti celého systému na nepájivém poli s zapojením ESP8266.(listopad 2021)
3. Programování systému skleníku a odesílání dat na internetovou stránku(prosinec 2021)
4. Návrh plošného spoje (leden 2021)
5. Kompletace skleníku a testování funkčnosti. (únor 2021)

***Požadavek na počet vyhotovení maturitní práce: 2 výtisky***

Termín odevzdání: **29.4.2022**  
Čas obhajoby: **15 minut**  
Vedoucí práce: **Jan Drexler**

Projednáno v **katedře ODP** a schváleno ředitelkou školy.

V Plzni dne: 30. září 2021

Ing. Naděžda Mauleová, MBA, v.r.  
*ředitelka školy*

## **Anotace**

Autonomní skleník

Cílem mojí ročníkové práce je seznámit se s problematikou pěstování rostlin ve skleníku, získat nové zkušenosti v realizaci projektů. Zároveň chci se lépe naučit v jazyce C a zlepšit schopnosti v programování a v konstruování skleníků. Skleník bude obsahovat senzor vlhkosti, teploty a světla. A také aktivní části jako je větrák, světla a vodní pumpa. S cílem co největší efektivity růstu květin a minimalizace plýtvání vody. Pro finální test se pokusím vypěstovat sazenici petržele. Výhled do budoucnost je poté model využít jako předlohu pro normální zahradní skleník.

## **Klíčová slova**

Skleník, ESP32, zavlažování

## **Prohlášení:**

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne: ..... Podpis: .....

## **Poděkování**

Tímto bych rád poděkoval panu Josefu Fořtovi za nenahraditelnou pomoc při tvorbě projektu. Dále bych rád poděkoval panu Englerovi za pomoc při konstrukčním řešení a svojí matce Markétě Rybové za finanční podporu během tvorby práce. V neposlední řadě bych chtěl poděkovat panu Liborovi Nebeskému za obrovskou pomoc při tvorbě návrhu a konstrukce skleníku.

# Obsah

1 Úvod.....	6
2 Elektrické a řídicí součásti .....	7
2.1 Řídicí blok.....	7
3 Senzory.....	8
3.1 Senzor vlhkosti půdy.....	8
3.2 Senzor intenzity světla .....	10
3.3 Senzor teploty a vlhkosti.....	11
4 Peristaltické čerpadlo .....	12
5 Ventilátor.....	13
6 Spínací prvek.....	14
7 Napájení .....	15
8 Použité technologie .....	17
8.1 Arduino IDE.....	17
8.2 Wiring .....	17
8.3 Ubidots .....	18
8.4 I <sup>2</sup> C.....	19
8.5 WIFI .....	20
9 Modelování a konstrukční řešení .....	22
9.1 Vývoj.....	22
10 Schéma zapojení.....	24
11 Program skleníku .....	26
12 Praktická realizace .....	30
13 Závěr .....	32
14 Zdroje a prostředky .....	34
15 Seznam obrázků .....	35
16 Seznam příloh.....	36

# 1 Úvod

V dnešní době při stále větším suchu a klimatických změnách ubývá úrodné půdy pro pěstování rostlin. Proto se stále více do popředí dostává chytré a efektivní zemědělství s využitím IoT řešení.

V chytrém zemědělství se stává stále důležitější a výnosnější způsob hospodaření. Senzory jsou dnes využívány k velkému množství věcí, které farmářům šetří čas i peníze. Dříve musel statkář denně chodit po poli a kontrolovat stav půdy. Dnes se dá zjistit stav, teplota, vlhkost nebo třeba i počet škůdců přímo z počítače.

Díky novým technologiím v mapování, GPS a IoT senzorů se nyní mohou sbírat nejrůznější data, které zvyšují efektivitu hnojení a zavlažování a snižují jejich spotřebu. Traktory vybavené chytrými zařízeními mohou automaticky upravovat množství hnojiva nebo vody podle stavu daného kusu půdy, který získá ze senzorů umístěných v zemi nebo ze satelitních snímků. Jedním z přínosů této technologie je zároveň i snížení dopadu na životní prostředí.

Všechny tyto principy se snažím uplatňovat a implementovat do mého „chytrého“ skleníku. Což je specifická stavba sloužící pro pěstování rostlin. Stavba je přímo designovaná pro pěstování rostlin, a proto se zde nechají vytvořit téměř dokonalé podmínky jako vlhkost, teplota, světelný svit.

V mé práci se zabývám problematikou pěstování rostlin ve skleníku a snažím se vytvořit funkční model s využitím IoT řešení, který bude co nejvíce autonomní. A bude po prvotním nastavení následně schopný téměř bez „cizí pomoci“ vypěstovat sám rostlinu.

Všechny naměřené parametry budou posílány do IoT platformy, kde se budou ukládat do databází a následně bude probíhat vizualizace dat do přehledných grafů. Přístup k těmto datům bude umožněn přes URL odkaz, aby se správce mohl kdykoliv podívat na stav rostliny.

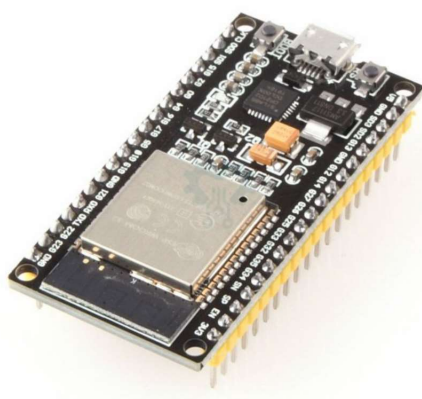
Výhled do budoucna je vytvořit z modelu funkční stavbu, kde bude možné pěstování rostlin ve větších objemech a použití navrhovaných řešení různých problémů v komerční sféře.

## 2 Elektrické a řídicí součásti

### 2.1 Řídicí blok

Pro ovládání skleníku jsem zvolil ESP32, které bude sloužit, pro vyhodnocování dat ze senzoru a následně řídit kdy se sepnou aktivní části skleníku (větrák, čerpadlo, osvětlení) .

Tento modul jsem zvolil především, kvůli přítomnosti wi-fi modulu, malým nákladům na pořízení cca 200 Kč a také možnosti vývojové desky programovat v softwaru Arduino IDE.



Obrázek 1: Vývojová deska ESP32

Specifikace:

- SoC: ESP32-D0WDQ6 (revision 1)
- Frekvence CPU 80 až 240 MHz
- Programová flash 4MB
- SRAM 520 kB
- WiFi 2,4-2,5GHz 802.11bgn (802.11n až do 150 Mbps)
  - režim Wifi - Station/softAP/SoftAP+station/P2P
  - zabezpečení WPA/WPA2/WPA2-Enterprise/WPS
- Bluetooth v4.2 BR/EDR a BLE, podpora audia CVSD a SBC
- Hardware akcelerované - Cryptography/RSA-4096
- Rozhraní paměťových modulů SD/SDIO/CE-ATA/MMC/eMMC
- Komunikační rozhraní 3xUART, 3xSPI, SDIO, 2xI2C, 2xI2S, IR
- Rozhraní vstupně-výstupní 34xGPIO, 18xADC, 2xDAC, 16xLED PWM, LNA, kapacitní dotykové senzory, hallové senzory
- RTC s odděleným oscilátorem 32kHz
- Interní 26MHz oscilátor pro CPU
- Různé režimy nízké spotřeby až 3mA v aktivním režimu, 2.5uA v hibernaci.

### 3 Senzory

#### 3.1 Senzor vlhkosti půdy

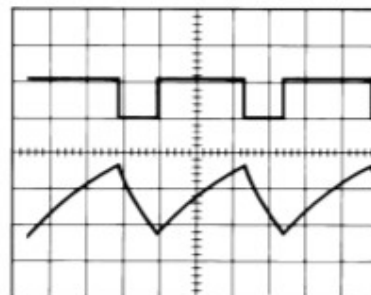
Pro měření vlhkosti půdy je využit kapacitní půdní vlhkoměr. Využívá nepřímé metody měření vlhkosti v půdě a funguje na principu – na senzoru je umístěn oscilátor 555 Timer IC, který generuje obdélníkový střídavý signál o určité frekvenci. (frekvence je dána vztahem

$$f = \frac{1}{T} = \frac{1,44}{(R_A + 2R_B)C}$$

do kondenzoru (dvě vodivé desky oddělené dielektrikem) a vlastnosti dielektrika se mění v závislosti na vlhkosti půdy. Tedy čím vyšší je vlhkost půdy, tím se zvyšuje i kapacita a snižuje tedy reaktance kondenzoru. Tyto změny jsou vyhodnocovány v obvodu a vráceny do vývojové desky jako analogový signál, který mi následně vyhodnocujeme.

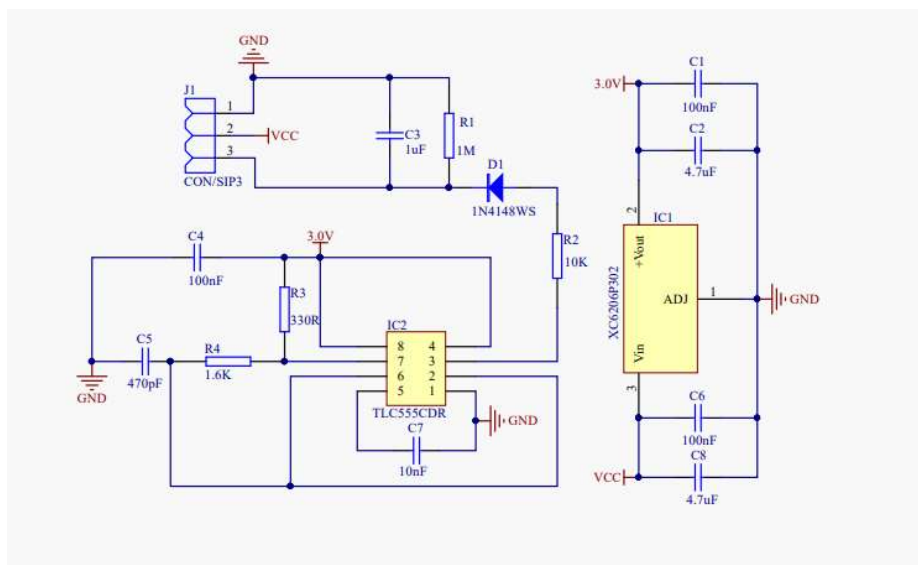
Výhodou je oproti běžnému měření vlhkosti půdy pomocí změny hodnoty odporu mezi dvěma elektrodami snímače, že při tomto typu měření se využívá střídavého proudu a kondenzátoru, který je ukryt pod ochranou vrstvou. Nedochozí tedy ke korozi odkrytých elektrod vlivem elektrolýzy jako při odporovém měření.

Mezi další kladné vlastnosti se řadí vysoká přesnost měření a nízká energetická náročnost. Napájecí napětí senzoru je 3,3V až 5V a výstupní napětí je 0V až 3V. A připojen s ESP32 rozhraním Rozhraní: PH2.0-3P.



Obrázek 2: Průběh napětí na senzoru





Obrázek 3: Vnitřní schéma zapojení vlhkoměru



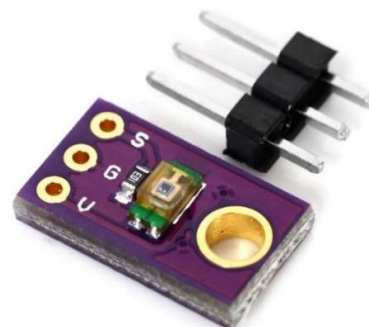
Obrázek 4: Půdní vlhkoměr

### 3.2 Senzor intenzity světla

Světelný senzor TEMT6000 je modul. Který obsahuje senzor osvětlení pracující podobně jako NPN tranzistor. Při nulovém osvětlení je přechod zavřený a na výstupu naměříme nula voltů. Se vzrůstajícím množstvím světla dopadajícím na senzor se zvyšuje napětí na tranzistoru, které mi měříme pomocí mikrokontroleru ESP32 a vyhodnocujeme ho.

Senzor je nejvíce citlivý na záření, které má vlnovou délku 570 nm a poloměr citlivosti senzoru je  $\pm 60$  stupňů. Pro napájení senzoru můžeme využít napájecí napětí 3,3 nebo 5 Voltů. A možnosti napájení jsou 3,3 až 5V.

Výhoda senzoru je jeho nízká cena a malé rozměry. Relativně jednoduché zapojení. A však nevýhodou je jeho možná nepřesnost.



Obrázek 5: Senzor TEMT6000

#### BASIC CHARACTERISTICS ( $T_{amb} = 25^{\circ}\text{C}$ , unless otherwise specified)

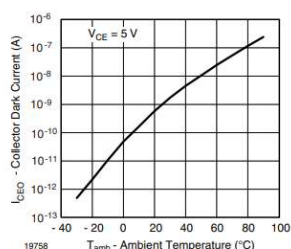


Fig. 1 - Collector Dark Current vs. Ambient Temperature

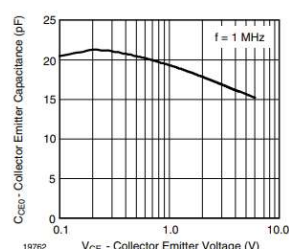


Fig. 4 - Collector Emitter Capacitance vs. Collector Emitter Voltage

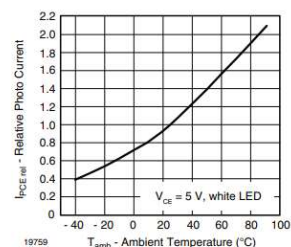


Fig. 2 - Relative Photo Current vs. Ambient Temperature

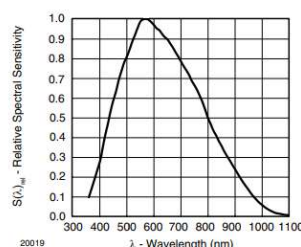


Fig. 5 - Relative Spectral Sensitivity vs. Wavelength

Obrázek 6: Základní charakteristiky pro tepelné a světelné podmínky

### 3.3 Senzor teploty a vlhkosti

Ve skleníku je použit konkrétně senzor HTU21D, což je vstupní modul komunikující přes sběrnici I2C, deska už je rovněž osazena 4,7K pull-up rezistory kvůli komunikaci I2C.

Nejčastější hodnoty pull-up rezistorů se pochybují v rozmezí od 2K do 10K ohmů, avšak dvě nejpoužívanější hodnoty jsou 4,7K a 10K. Použití odporů je důležité z důvodu udržení linky SDA a SCL během klidového stavu na hodnotě HIGH. Špatně zvolená hodnota pull-up odporu může způsobovat ztrátu informací.

Přesná hodnota pull-up se

vypočte ze vzorce:

$$R_{PUP}(\text{Max}) = \frac{T_R(\text{Max})}{0.8473 \times \text{XCL}}$$

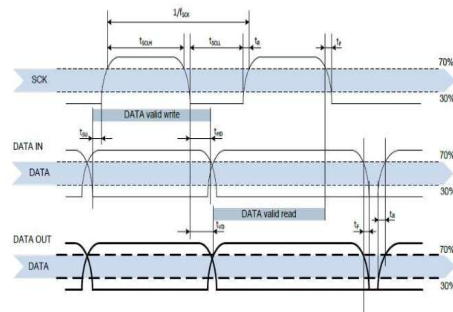
$$R_{PUP}(\text{Min}) = \frac{V_{CC} - V_{OL}(\text{Max})}{I_{OL}}$$

$T_R$  = maximální čas

náběhu;  $CL$  =

kapacita sběrnice;

$V_{OL}$  = low-level výstupní napětí;  $I_{OL}$  = výstupní proud



Obrázek 7: Časové průběhy na lince SCK

Senzor dokáže měřit teplotu v rozsahu -40 až +125 stupňů Celsia a vlhkost v rozsahu 0 až 100% relativní vlhkosti. Co se týká přesnosti měření, tak u teploty je přesnost typicky  $\pm 0,3$  °C a nejvíce přesné je měření v rozsahu 5 až 60 °C. Zatímco u vlhkosti je přesnost  $\pm 1\%$  RH a nejpřesněji dostaneme výsledek v rozsahu 10 až 90% RH.

Napájecí napětí pro tento modul je výrobcem doporučeno v rozsahu 3,3 až 5 Voltů. Odebíraný elektrický proud je velmi nízký, kdy v klidu dosahuje maximální hodnoty 140 nano Ampér a při měření maximálně 0,5 miliampér.

Senzor byl vybrán kvůli malým pořizovacím nákladům, malému odběru proudu, díky čemuž je vhodný i na bateriový provoz a také další benefity byla vysoká přesnost a malé rozměry



Obrázek 8: Senzor HTU21D

## 4 Peristaltické čerpadlo

Pracuje na principu paměti materiálu, ze které je vyrobena flexibilní trubice čerpadla, která je střídavě stlačována a uvolňována. Během pomalého otáčení rotoru dochází k postupné deformaci hadice, přičemž se uzavře malý objem kapaliny v trubici, který je poté před lopatkou (bývá 2 až 6 lopatek) vytlačován z hadice po směru rotace lopatky a za lopatkou tudíž vzniká podtlak, který má za následek opětovné nasátí čerpané kapaliny. Sání je řízen směrem, kterým se otáčí čerpadlo.

Peristaltická čerpadla jsou často využívána tam, kde je potřeba velmi přesné dávkování, jelikož velikost odebraného vzorku je dána počtem otáček čerpadla s nasátou kapalinou, kterou detekuje senzor v nasávacím potrubí. Další výhodou tohoto typu čerpadla spočívá v tom, že nedochází ke křížové kontaminaci.



Obrázek 9: Peristaltické čerpadlo

Čerpaná kapalina se nedostává do kontaktu s žádnými pohybujícími se díly čerpadla, protože je zcela uzavřená ve vyztužené hadici nebo hadičce. Navíc tím, že je kapalina zcela uzavřená v hadici nebo hadičce, se stává peristaltické čerpadlo dokonalým hygienickým řešením, kde nehrozí nebezpečí kontaminace. Tím se také snižují požadavky na údržbu, protože jedinou opotřebitelnou součástí čerpadla je hadice nebo hadička. Kvůli bezpečnosti a jistotě nemá čerpadlo žádné mechanické těsnění a v případě poškození hadice zůstane kapalina bezpečně uzavřená v čerpadle. Tento typ se používá např. ve zdravotnictví při dialýze.

Tento typ čerpadla byl vybrán, kvůli přednostem uvedeným výše, což je vysoká přesnost dávkování a bezpečnost. Vzhledem k tomu, že model skleníku je relativně malý. Je zde vyžadováno i malé a přesné dávkování vody, aby bylo zavlažování co nejefektivnější.

Zde je použito konkrétně čerpadlo typu Grothen G328 Peristaltické čerpadlo 12V. S parametry : DC Napětí: 12V, Proud: 200mA

## 5 Ventilátor

Ventilátor v naší konstrukci slouží ke snížení teploty a vlhkosti. Je použit model původně sloužící k chlazení CPU počítače. Byl vybrán pro jeho vhodné konstrukční řešení a malé náklady. Plus snadné a rychlé zapojení. Je napájen 12V a 0,15A. Při rozměrech 8x8 cm.

Ventilátor je lopatkový stroj s kolem nebo vrtulí pro dopravu a stlačování plynů a par do tlaku 10 kPa. Pohon zajišťuje elektromotor, což je v elektrotechnice elektrický stroj, který slouží k přeměně elektrické energie na mechanickou energii. Většina elektrických motorů využívá vzájemného silového působení magnetického pole a elektrického proudu procházejícího cívkou, výsledkem čehož je točivý moment přenášený na poháněné zařízení (např. vysavač, výtah, elektromobil). Motory jsou poháněny stejnosměrným nebo střídavým proudem, který je přiváděn z baterie, elektrického rozvodu nebo generátoru. Elektrický generátor je mechanicky stejný jako elektromotor, ale slouží k opačnému převodu mechanické energie na elektrickou. Existuje mnoho typů elektrických motorů podle napájení, konstrukce, provozních charakteristik a způsobu převodu elektrické energie na mechanickou. Nejrozšířenějším typem je asynchronní motor.

Elektromotory jsou ve srovnání se spalovacími motory lehčí, menší, jednodušší, efektivnější, lacinější, poskytují vysokou účinnost v celém rozsahu otáček. Přestože byly využívány v počátcích automobilismu, jsou elektromotory v roce 2020 méně rozšířené než spalovací, protože potřebují neprakticky velké a pomalu nabíjející se akumulátory, které mají nízkou životnost, malý rozsah provozních teplot a vysokou cenu. V této době dochází díky pokroku v konstrukci elektromotorů i akumulátorů k rozvoji elektromobilů.



Obrázek 10: Použitý ventilátor

Specifikace: Napětí: DC 12V; Výkon: 1,44W; Počet otáček za minutu: 3000

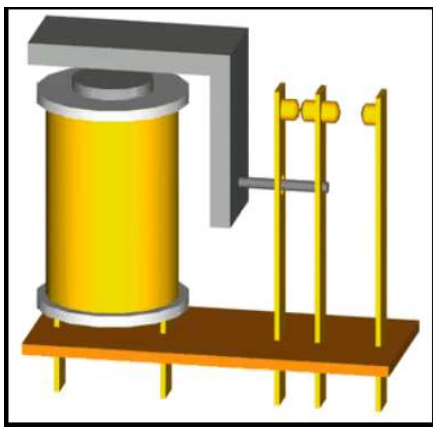
## 6 Spínací prvek

V projektu jsou funkční části pracující na vyšším napětí, než je schopné ESP32 vyvinout. Proto je použito 4-kanálové relé. Což nám umožňuje spínat ventilátor, pumpu a led pásek, protože všechny tyto části vyžadují 12V externí zdroj napětí.

Elektromagnetické relé je elektrotechnická součástka, která obsahuje elektromagneticky ovládané kontakty a funguje jako spínač. Je tvořeno elektromagnetem tedy cívkou navinutou na měkký feromagnetický materiál, kotvou, pružinou a elektrickými kontakty. Při klidovém režimu tlačí pružinka na kotvu a to spojuje v našem případě kontakt COM s kontaktem NO (normally-open) a po připojení cívky na elektrický zdroj. Vyvolá procházející elektrický proud cívkou magnetický tok a cívka tak přitáhne kotvu k sobě. Čímž pohne s kontaktem COM a propojí ho s kontaktem NC. A tím propojí elektrický obvod a ovládaný prvek se sepne. Relé se tak využívá k ovládní velkého výkonu malým výkonem.

V tomto projektu je konkrétně použito relé Songle SRD-12VDC-SL-C. Které má 4 kanály a každý kanál má svůj pin na který, když se přivede elektrický proud, kanál se sepne.

Specifikace jsou: pracovní napětí - 5V, vyžadovaný proud pro sepnutí 15-20mA a maximální parametry jsou 250VAC 10A. A využívá se zde low-level úroveň pro spínání. Musí se tedy na pin přivést zemní napětí.



Obrázek 12: Jednoduché schéma popisující vnitřní konstrukci relé



Obrázek 11: Použité 4 kanálové relé

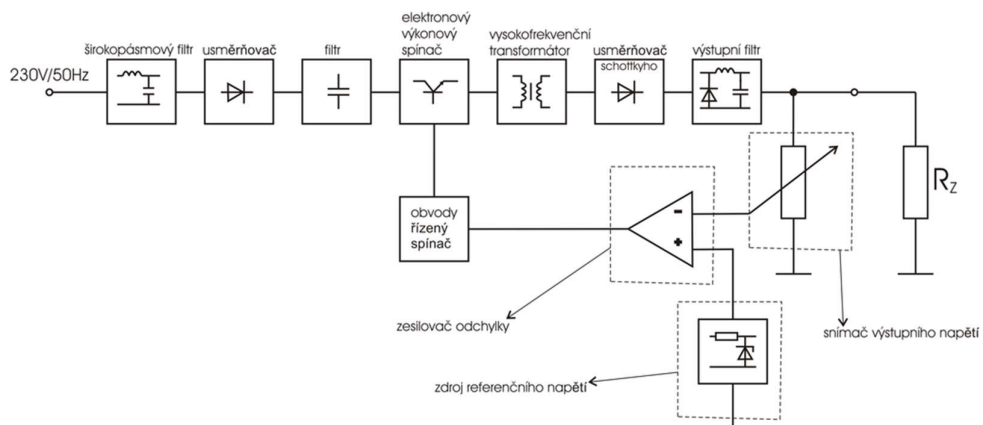
## 7 Napájení

Elektrický zdroj je v elektrotechnice označení pro zařízení měnící elektrickou energii na výstupní požadované napětí, proud a frekvenci. Dělíme je na dva základní typy a to zdroj napětí, který udržuje stabilní napětí a zdroj proudu který udržuje konstantní napětí proudu.

Poté se podle konstrukčního řešení dělí na další dva typy- spínaný zdroj a stabilizovaný zdroj.

**Lineární stabilizovaný zdroj** se skládá ze 4 částí, což jsou transformátor, usměrňovač, filtr a stabilizátor. Transformátor mění napětí z rozvodné sítě na menší – bezpečnější, ale musí být větší, než které požadujeme na výstupu zdroje. Na primární vynutí se připojuje síťové napětí a na sekundární vynutí je napojen usměrňovač, který usměrňuje střídavé napětí na stejnosměrné. Poté následuje filtr který vyrovňuje stejnosměrné napětí, aby mělo co nejmenší zvlnění. Nejjednodušší zapojení může být paralelně připojený kondenzátor . A poslední částí je stabilizátor, který zajišťuje, aby při změně odebíraného proudu zůstalo napětí stejné.

**Spínaný zdroj** je na rozdíl od transformátorového zdroje levnější, lehčí a účinnější, ale na druhou stranu je zase konstrukčně složitější. Hlavním rozdílem je užití výkonného spínacího tranzistoru například typu MOSFET. Toto konstrukční řešení se užívá v nabíječkách mobilních telefonů nebo ve zdrojích pro počítače a notebooky.



Obrázek 13: Blokové schéma spínaného zdroje



V projektu se používají dva síťové napájecí zdroje spínaného typu. Jeden má výkon 12W a je dimenzován pro napájení ventilátoru, LED pásku a pumpy. A druhý s výkonem 5W napájí mikrokontroler ESP32.

Konkrétně to je síťový napájecí adaptér 12V/1A 5,5/2,1mm

Specifikace:

Vstup: 110V-240V 50/60 Hz

Výstup: 1A 12V

Konektor: 5.5x2.1mm (vhodný pro Arduino Mega a UNO)

Orientace polarity: (+) uvnitř, (-) vnější obvod konektoru

Rozměry: 76x75x26mm

A druhý je síťový napájecí adaptér 5V/1A 5,5/2,1mm

Specifikace:

Vstup: 110V-240V 50/60 Hz

Výstup: 1A 5V

Konektor: 5.5x2.1mm

Orientace polarity: (+) uvnitř, (-) vnější obvod konektoru



Obrázek 14: Použitý síťový zdroj



## 8 Použité technologie

### 8.1 Arduino IDE

Arduino IDE je vývojové prostředí pro Arduino a řadu dalších vývojových desek. Které nám umožňuje jednoduše s deskou pracovat. Program byl psán v Javě a je kompatibilní s Windows, Linuxem a iOS. Krom textového editoru také software obsahuje konzoli, lištu pro práci s kódem a lištu pro nastavení IDE. Dále je součástí prostředí nástroj *serial monitor*, který umožňuje jednoduchou komunikaci s Arduino deskou na straně hardware k tomuto účelu pak slouží knihovna Serial.

### 8.2 Wiring

Wiring je programovací jazyk vytvořený pro programování mikrokontroléru bez specifických znalostí hardware. V současné době je nejznámější jako součást open-source platformy Arduino, kde má podobu frameworku v jazyce C++. Wiring vznikl pro vývojový kit podobný Arduino a vychází z dalšího open-source projektu Processing. Pro programování v jazyce Wiring se nejčastěji používá integrované vývojové prostředí Arduino IDE, k dispozici jsou ale i další vývojová prostředí jako Arduino Eclipse. Wiring vyžaduje mikrokontrolér se zaváděcím programem, typicky desku Arduino osazenou čipem ATmega. Prvotním autorem jazyka je Hernando Barragán, který ho definoval ve své diplomové práci na italském institutu IDII (Interaction Design Institute Ivrea) jako součást prototypovacích nástrojů pro elektroniku a programování. (Wikipedia, 2022)

```
int ledPin = WLED; // pojmenování LED

void setup () {
  pinMode(ledPin, OUTPUT); // nastavení pinu do digitální výstup
}

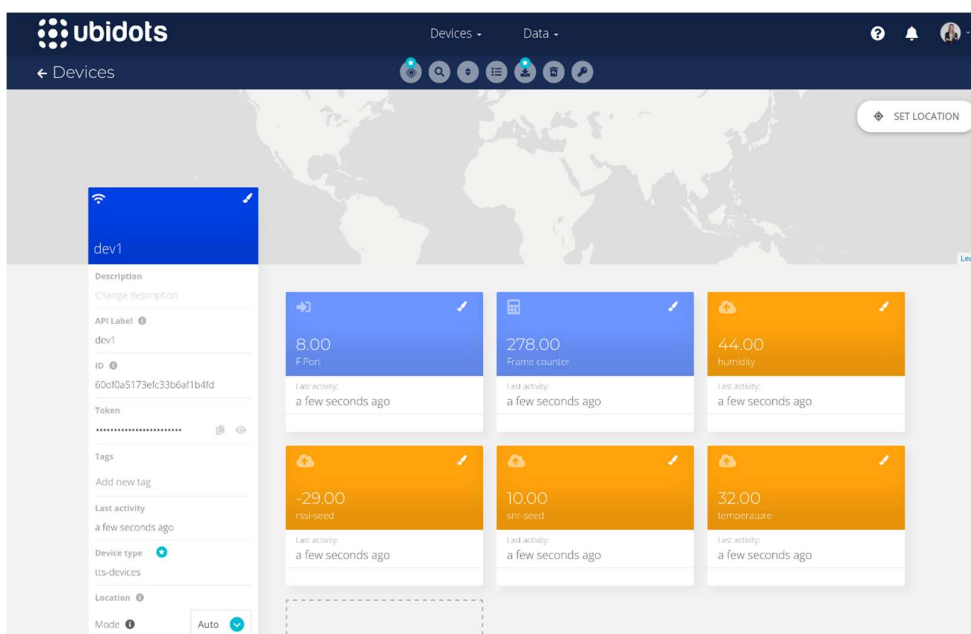
void loop () {
  digitalWrite(ledPin, HIGH); // zapnutí LED
  delay (1000); // počkáme jednu sekundu (1000 milisekund)
  digitalWrite(ledPin, LOW); // vypnutí LED
  delay (1000); // počkáme jednu sekundu
}
```

Obrázek 15: Ukázka jednoduchého kódu pro rozsvícení led diody

## 8.3 Ubidots

Ubidots je nástroj pro tvorbu IoT projektů s analýzou a vizualizací dat. Data ze senzorů mění na informace, které mají význam pro obchodní rozhodování, interakci mezi stroji, vzdělávací výzkum a zvýšenou ekonomizaci globálních zdrojů. Ubidots existuje jako snadný a cenově dostupný prostředek pro integraci síly a hodnoty internetu věcí do podniků a výzkumu. Společnost Ubidots, založená v roce 2013, je investory podporovaný startup poskytující platformu pro podporu aplikací internetu věcí pro kutily, pedagogy a profesionály.

Ubidots komunikuje přes rozhraní API (přístupným přes protokoly HTTP/MQTT/TCP/UDP), poskytuje jednoduché a bezpečné připojení pro odesílání a načítání dat do backendu (cloudu) optimalizovaného pro výkon dat IoT v čase. Platforma pro podporu aplikací Ubidots podporuje interaktivní vizualizaci dat v reálném čase (widgety) a nástroj IoT App Builder, který vývojářům umožňuje rozšířit platformu o vlastní kód HTML/JS pro případné soukromé přizpůsobení. Ubidots existuje pro podniky a výzkumné pracovníky, aby mohli efektivně propojovat zařízení, spravovat data a šetřit prostředí.

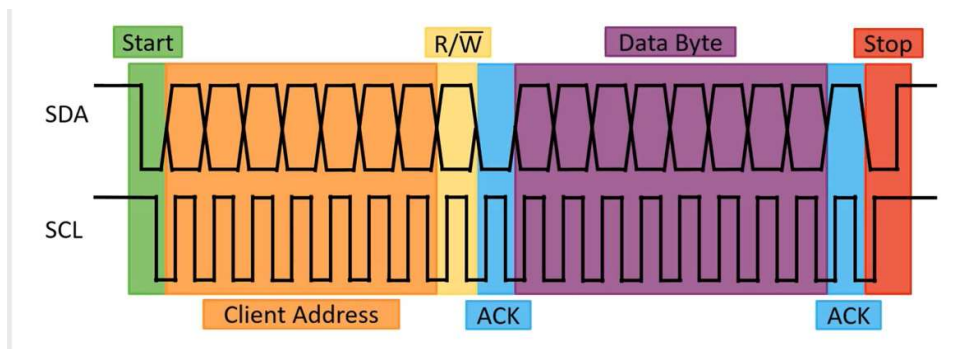


Obrázek 16: Příklad možného vzhledu dashboardu

## 8.4 I<sup>2</sup>C

V práci využívám teplotní čidlo HTU21D, které komunikuje s mikrokontrolerem přes I<sup>2</sup>C.

I<sup>2</sup>C bus je zkratka která vznikla z IIC bus, tedy Internal-Integrated-Circuit Bus. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Vyvinula ji firma Philips přibližně před 20 lety a od té doby prošla několika vylepšeními, ale o tom až později. V dnešní době tuto sběrnici podporuje řada integrovaných obvodů nejen firmy Philips. Jedná se především o mikrokontroléry, sériové paměti, inteligentní LCD, audio a video obvody, a/d a d/a převodníky a některé další digitálně řízené obvody. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - "data SDA (serial data)" a "hodiny SCL (serial clock)". To především u mikrokontrolérů výrazně optimalizuje nároky na počet vstupně-výstupních pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128 respektive 1024 čipů s různou adresou na jednu společnou sběrnici. V praxi jsou tato čísla však podstatně nižší, protože adresa čipu většinou nelze určit plnými 7 (10) bity ale třeba jen třemi. Někdy nelze určit vůbec a je dána na pevně pro daný typ čipů tedy na jedné sběrnici nemůže být více než jeden. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz. Ve vylepšených verzích to může být 400kHz nebo 1MHz, ale ne všechny integrované obvody tuto verzi podporují. Rychlost přenosu pak musí být přizpůsobena pochopitelně "nejpomalejšímu" čipu na sběrnici. Oba vodiče musí být implicitně v logické jedničce, a to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4k7.

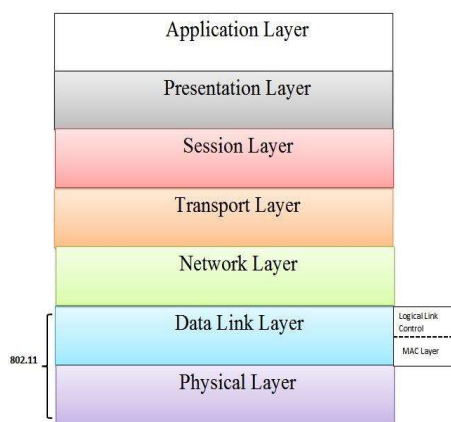


Obrázek 17: Datový rámec sběrnice I2C

## 8.5 WIFI

Pro posílání dat z ESP32 do ubidots databázi využívám WIFI standart. V informatice označení pro několik standardů IEEE 802.11 popisujících bezdrátovou komunikaci v počítačových sítích (též Wireless LAN, WLAN). Samotný název WiFi vytvořilo Wireless Ethernet Compatibility Alliance. Tato technologie využívá tak zvaného „bezlicenčního frekvenčního pásma“, proto je ideální pro budování levné, ale výkonné sítě bez nutnosti pokládky kabelů. Název původně neměl znamenat nic, ale časem se z něj stala slovní hříčka wireless fidelity (bezdrátová věrnost) analogicky k Hi-Fi (high fidelity – vysoká věrnost).

Původním cílem Wi-Fi sítí bylo zajišťovat vzájemné bezdrátové propojení přenosných zařízení a dále jejich připojování na lokální (např. firemní) síť LAN. S postupem času začala být využívána i k bezdrátovému připojení do sítě Internet v rámci rozsáhlejších lokalit a tzv. hotspotů. Wi-Fi zařízení jsou dnes prakticky ve všech přenosných počítačích, a i v některých mobilních telefonech. Úspěch Wi-Fi přineslo využívání bezlicenčního pásma, což má negativní důsledky ve formě silného zarušení příslušného frekvenčního spektra. Komerčním nástupcem Wi-Fi měla být bezdrátová technologie WiMAX zaměřená na zlepšení přenosu signálu na větší vzdálenosti, která se však u mobilních operátorů v České republice neujala. Ale v jiných zemích funguje bez problémů už několik let.



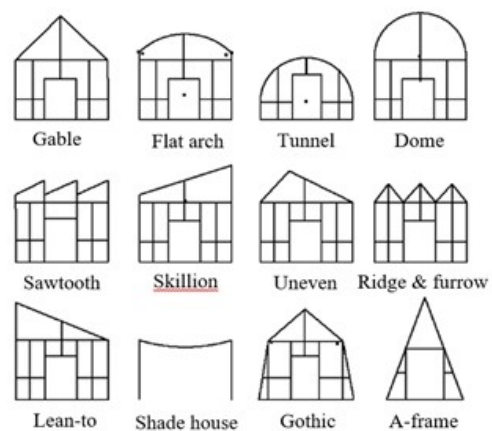
Obrázek 18: Referenční model ISO/OSI

Wi-Fi zajišťuje komunikaci na spojové vrstvě, zbytek je záležitost vyšších protokolů (na rozdíl od Bluetooth, který sám o sobě zajišťuje nejrůznější služby). Typicky se proto přenášejí zapouzdřené ethernetové rámce. Pro bezdrátovou komunikaci na sdíleném médiu (šíření elektromagnetického pole prostorem) je používán protokol CSMA/CA (Ethernet používá na vodičích CSMA/CD).

## 9 Modelování a konstrukční řešení

### 9.1 Vývoj

První kroky při vývoji konstrukce skleníku vedly k výzkumu, jaké nejčastější konstrukční řešení se používají, a které z nich bude pro náš projekt nejlépe vyhovovat, aby bylo možno ho vytvořit v našich podmínkách. Existuje nespočet typů skleníku.



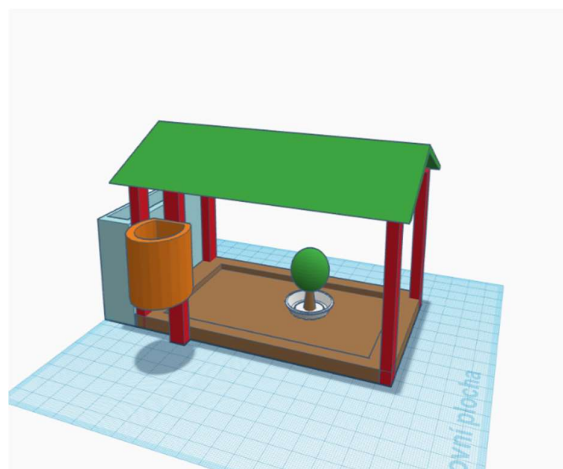
Obrázek 19 Typy skleníku

Navrhoval jsem skleník, tak aby ho bylo možno použít primárně uvnitř během zimních dnů a venku během léta. Skleník je navrhnut podle typu Gable s rozdílem, že nemá boční stěny. Z důvodu snadnější regulace teploty a přístupu do vně skleníku. Protože při montáži a testování at' už pumpy nebo ventilátoru by stěny překážely.

Pak následovalo první modelování v 3D programu Tinkercad od společnosti AutoCAD.

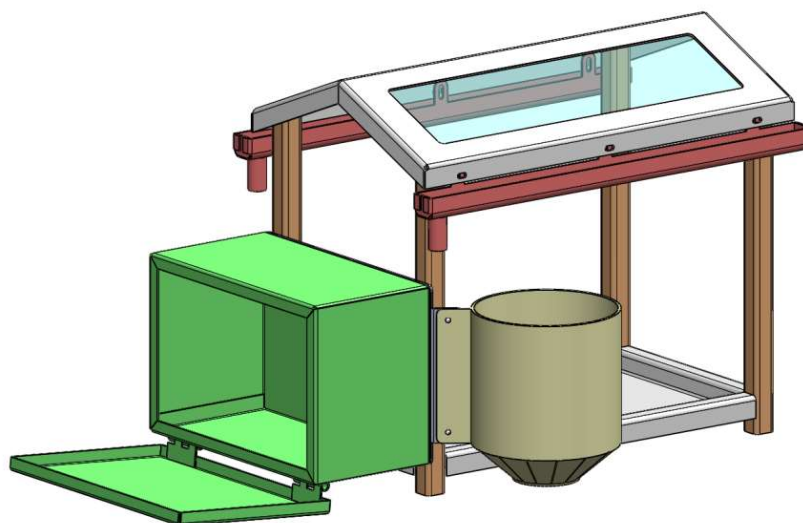
Z minulosti jsem neměl téměř žádné zkušenosti v 3D modelování. Proto jsem vybral program Tinkercad, který je ideální pro začátečníky, protože

tvoření v něm je intuitivní a jednoduché. V něm jsem vytvořil



Obrázek 20: První návrh modelu

i první model, což byl skleník, nádobka na vodu pro zavlažování, bedna na elektroniku kam se umístí mikrokontroler. Tento první model s dalšími připomínkami jsem odeslal inženýrovi z firmy Koramex, který podle mých instrukcí začal model upravovat tak, aby jej bylo možné strojově vyrobit. Po několik dnech intenzivní komunikace byl hotov první návrh konstrukčního řešení.



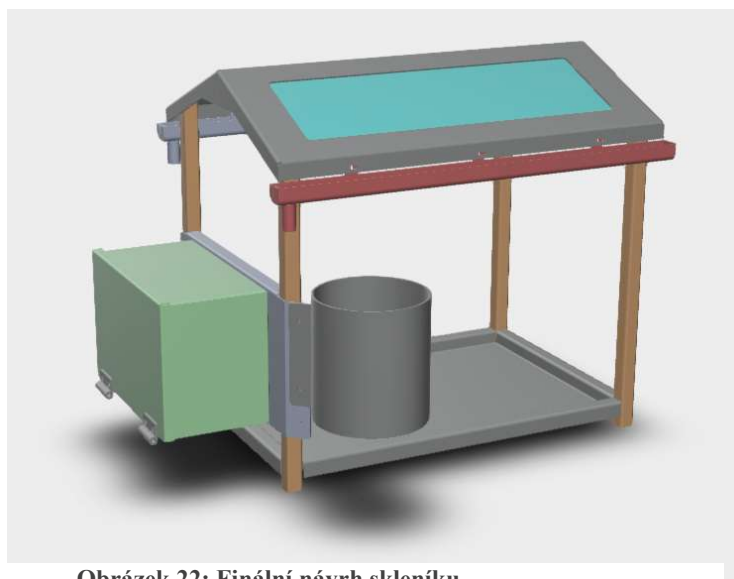
Obrázek 21: Druhý návrh skleníku

Tento návrh se zdál téměř ideální až na několik drobných nedostatků. A tím bylo umístění nádrže na vodu. Protože v této fázi projektu se ještě předpokládalo, že zavlažování bude realizováno pomocí bistabilního ventilu ovládaného impulzem. Voda zde měla téct samospádem, aby bylo co nejmenší využití energie. Bohužel po zakoupení ventilu jsem zjistil, že k plnému otevření je potřeba tlak alespoň 0,02MPa, a proto je nereálné jeho užití. A bylo místo toho využito peristaltické čerpadlo.

Další nedostatek byl až proporčně příliš velký rozvaděč oproti skleníku, který měl rozměry

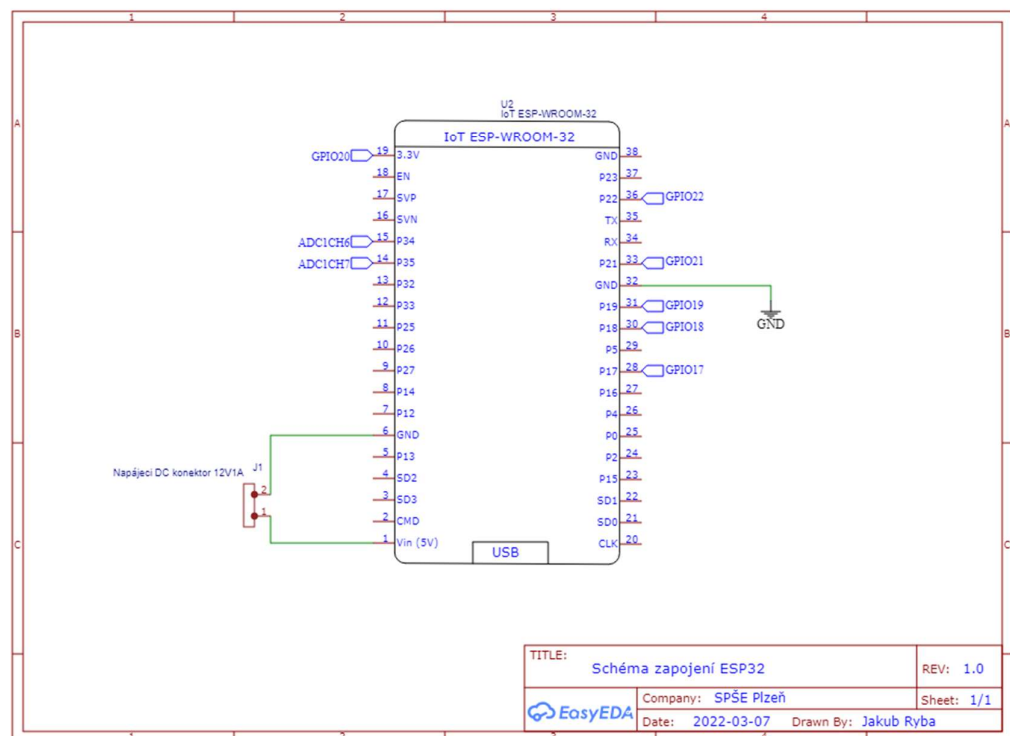
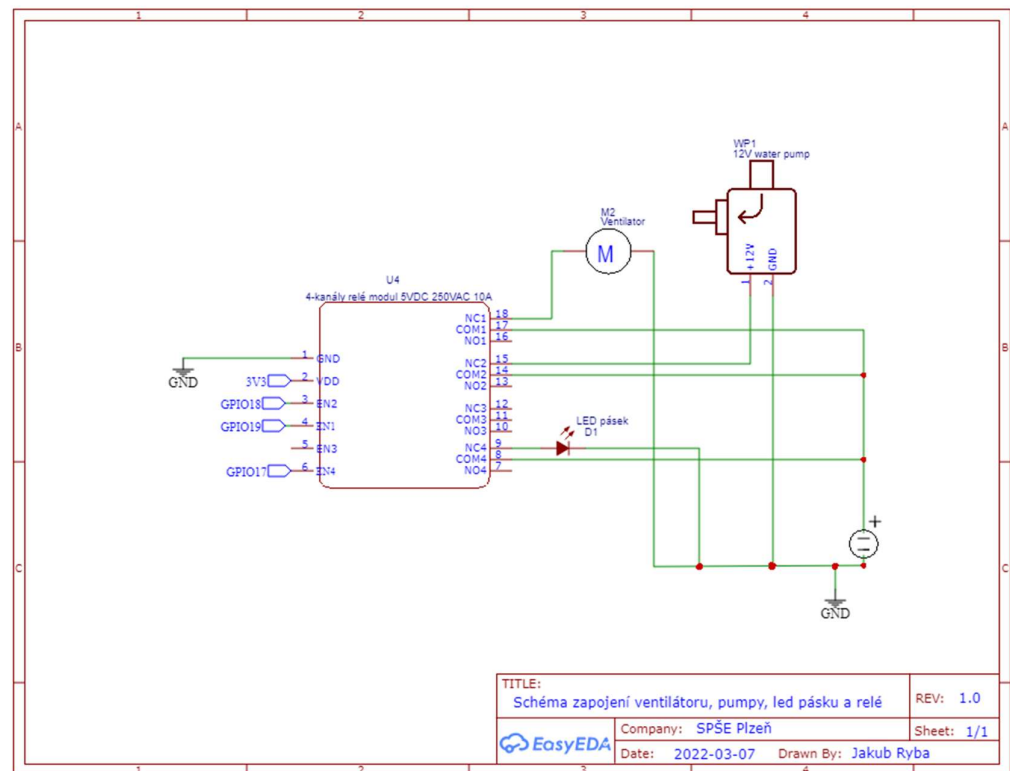
240x125x140mm a samotný skleník byl o rozměrech 226x350x400mm.

Z toho důvodu jsem navrhl zvětšení rámu skleníku o dalších 10cm. Po těchto úpravách byl vývoj hotov a model se mohl začít vyrábět.

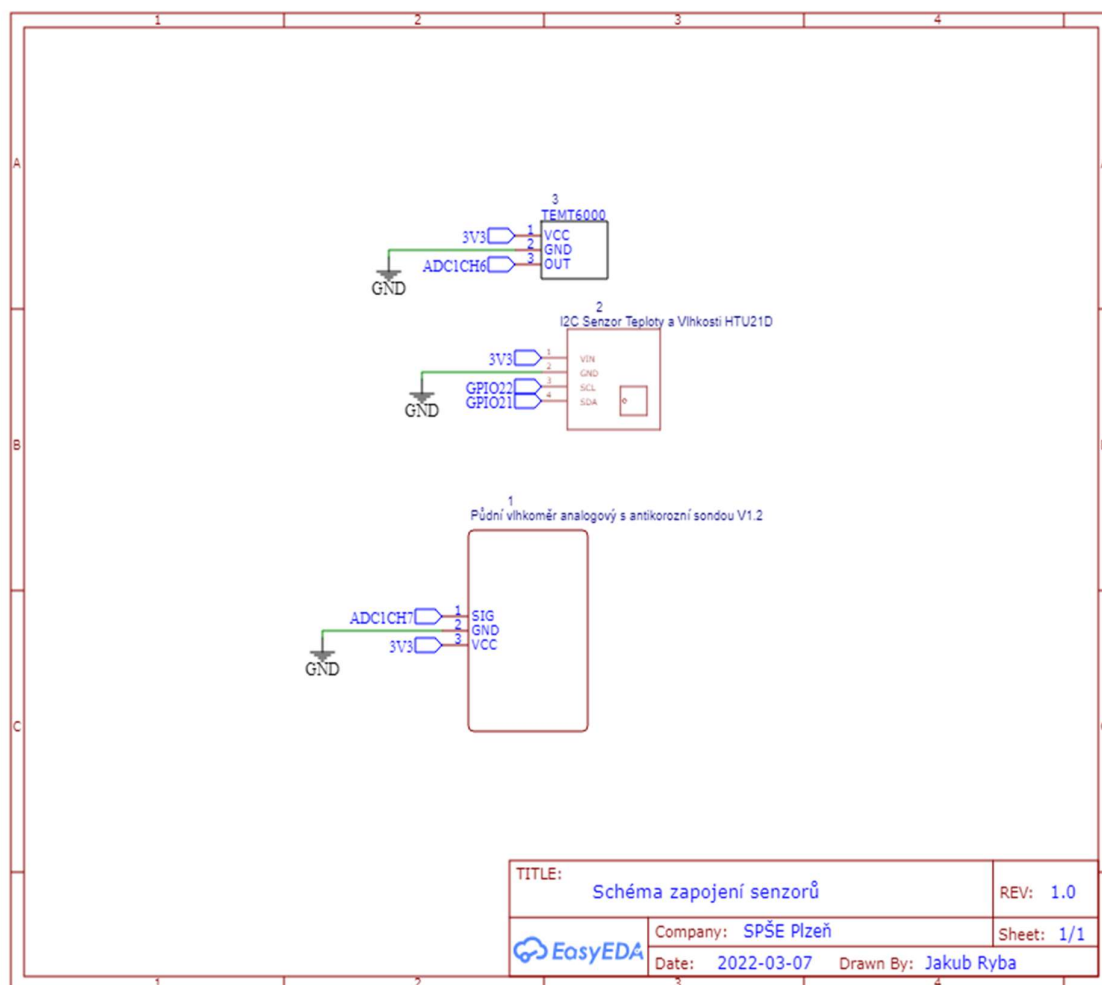


Obrázek 22: Finální návrh skleníku

## 10 Schéma zapojení







## 11 Program skleníku

V programu se nejdříve nastavují hodnoty proměnných, definují se vstupní a výstupní piny, wifi heslo a jméno ke kterému se bude vývojová deska připojovat.

Poté následuje připojení k wifi síti a k platformě Ubidots přes mqtt. Dále jsou zjišťovány hodnoty ze senzoru a ty se vypisují do sériového monitoru a poté následuje odesílání dat na platformu Ubidots. V poslední části se vyhodnocují data ze senzorů a podle jejich hodnot jsou nastaveny podmínky pro sepnutí.

```
/* **** */
#include <NTPClient.h>
#include "UbidotsEsp32Mqtt.h"
#include <Wire.h>
#include "SparkFunHTU21D.h"
HTU21D mujHTU;
WiFiUDP ntpUDP;
#define UTC_OFFSET_IN_SECONDS 3600
/* **** */
* Define promene
/* **** */
const char *UBIDOTS_TOKEN = "BBFF-
RNbcZtepq2pqsIRybuz7aGeZtcgzNT"; // Proměna do které se vloží
Ubidots token
const char *WIFI_SSID = "jakub"; // proměna s jmenm WIFI
const char *WIFI_PASS = "jakub1234"; // proměna s heslem k
WIFI
const char *DEVICE_LABEL = "esp33"; // Jméno zařízení které se
bude publikovat
const char *PROMENA_SVETLO = "svetlo"; // Proměna s názvem dat
pod kterým budou data zveřejňovány
const char *PROMENA_TEPLOTA = "teplota"; // Proměna s názvem dat
pod kterým budou data zveřejňovány
const char *PROMENA_VLHKOST = "vlhkost"; // Proměna s názvem dat
pod kterým budou data zveřejňovány
const char *PROMENA_VLHKOSTPUDY = "vlhkostPudy"; // Proměna s
názvem dat pod kterým budou data zveřejňovány
#define pinPumpa 18 //číslo pinů
#define pinDioda 17 //číslo pinů
#define pinVetrak 19 //číslo pinů
bool vodaStav = false; // proměná ve kterém stavu se zrovna
nachází pumpa
int test = 250;

float pudaVlhkost = 0;
int analogHodnota;
float prepocet;
//unsigned long timer;
#define analogPin 35
#define analogPin2 34
Ubidots ubidots(UBIDOTS_TOKEN);
NTPClient timeClient(ntpUDP, "de.pool.ntp.org",
UTC_OFFSET_IN_SECONDS); //nastavení NTP serveru ze kterého se
bude brát čas

/* **** */
```

```

*****/

void callback(char *topic, byte *payload, unsigned int length)
{
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++)
    {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

/*****
*hlavni cast
*****/

void setup()
{
    // část kodu která proběhne pouze jednou
    pinMode(pinDioda, OUTPUT); //nastavení pinu do výstupního
    režimu
    pinMode(pinVetrak, OUTPUT); //nastavení pinu do výstupního
    režimu
    pinMode(pinPumpa, OUTPUT); //nastavení pinu do výstupního
    režimu
    Serial.begin(115200);
    ubidots.connectToWifi(WIFI_SSID, WIFI_PASS); //připojení s WIFI
    sítí
    ubidots.setCallback(callback);
    ubidots.setup();
    ubidots.reconnect();
    timer = millis();
    mujHTU.begin();
}

void loop()
{
    // podmínka k připojení k ubidots
    if (!ubidots.connected())
    {
        ubidots.reconnect();
    }

    // načtení hodnoty z analogového pinu zjisti svietelnosti
    analogHodnota = analogRead(analogPin2);

    // výpis načtených dat po sériové lince
    Serial.println();
    Serial.print("Svietelnost: ");
    Serial.print(analogHodnota);
    Serial.println();

    delay(1000);
    // zjiteni teploty a vlhkosti vzduchu
    float teplota = mujHTU.readTemperature();
    float vlhkost = mujHTU.readHumidity();

```

```

if (teplota > 125 | vlhkost > 100) {
    Serial.println("Chyba komunikace se senzorem HTU21D!");
}
else {
    // vytištění naměřených údajů, 1 ve výpisu označuje
    // počet desetinných míst
    Serial.print("HTU21D | Teplota: ");
    Serial.print(teplota, 1);

    Serial.print(" st C | ");
    Serial.print("Vlhkost: ");
    Serial.print(vlhkost, 1);
    Serial.print("%");
    Serial.println();
}
// pauza 1 sekunda mezi výpisy
delay(1000);
//zjisteni teploty pudy
pudaVlhkost = analogRead(analogPin);
// vytištění výsledků po sériové lince
Serial.print("VLhkost Pudy: ");
Serial.print(pudaVlhkost);
Serial.println();
// pokud je překonána nastavená hranice,
// vytiskneme ještě upozornění

// pauza před novým měřením
delay(500);
// volání metody pro odesílání dat
posilaniDat(teplota, vlhkost, analogHodnota, pudaVlhkost);

//metody pro zapínání funkčních částí
svetlo(analogHodnota);
vetrak(teplota, vlhkost);
pumpa(pudaVlhkost);

}
//metoda pro odesílání dat
void posilaniDat(float teplota01, float vlkost01, float
svetlo1, float vlhkostZem) {

    ubidots.add(PROMENA_TEPLOTA, teplota01);
    ubidots.add(PROMENA_VLHKOST, vlkost01);
    ubidots.add(PROMENA_SVETLO, svetlo1);
    ubidots.add(PROMENA_VLHKOSTPUDY, vlhkostZem);
    ubidots.publish(DEVICE_LABEL);
}
/*****
/*****
//metoda-podmínka pro spínání vodní pumpy
void pumpa(float vlhkostPuda) {

    int refVlhkost = 300;
    if(vlhkostPuda < refVlhkost & vodaStav ==false ){
        digitalWrite(pinPumpa, HIGH);
        delay(5000);
        vodaStav = !vodaStav;
        digitalWrite(pinPumpa, LOW);
        Serial.println();
        Serial.println("Sviti");
    }
}

```

```

    }
    else {
        Serial.println("nesviti");
        vodaStav = !vodaStav;
    }
}

//metoda-podmínka pro spínání ventilátoru
void vetrak(float vlhkostVzduch, float teplotaVzduch) {

    int refTeplo = 25;
    int refVlhkostVz = 70;
    if (refTeplo > teplotaVzduch || refVlhkostVz > vlhkostVzduch ){
        digitalWrite(pinVetrak, HIGH);
        Serial.println("Vetrak zapnut");
    }
    else
    {
        digitalWrite(pinVetrak, LOW);
    }
}

//metoda-podmínka pro rozsvícení světel
void svetlo(float hladinaSvetlo){

    timeClient.update();
    int cas = timeClient.getHours()+1;
    int refSvetlo= 500;
    if(hladinaSvetlo<refSvetlo & cas > 7 & cas <20){
        digitalWrite(pinDioda, HIGH);
        Serial.println("Dioda zapnuta");
    }
    else{
        digitalWrite(pinDioda, LOW);
    }
}
}

```

## 12 Praktická realizace

Po vytvoření finální verze modelu započala výroba. První bod bylo vytvoření výkresu, podle kterých se bude celá výroba řídit. Poté se vytvořil program, kterým se řídil CNC laser. A vypálili se z plechu o různých rozměrech všechny části potřebné pro složení. Po vypálení všech potřebných částí se každá část



Obrázek 23: Svařený a ohnutý rám

musela ohnout do požadovaných rozměrů a poté svařit, natřít a naposledy smontovat. Obecně kvůli malým rozměrům a tenkým materiálům je vše navrhováno, tak aby se minimalizovalo svařování, tj. ohýbané spoje z jednoho kusu plechu nebo, pokud to je vhodné, šroubové či nýtované spoje místo svarů.

Rám je tvořen podlahou ohýbanou z jednoho kusu 2mm plechu a v rozích jsou přivařeny stojiny z jeklu. Střecha je také z jednoho kusu plechu a po stranách ohraněná (vyšší tuhost a mizí ostré hrany). Nádobu je tvořena z tenkostěnné trubky a k ní je přivařeno rovnostěnné dno. Krabice na elektroniku je ohnuta z jednoho kusu výpalku a je stažena nýty, aby držela pohromadě. A vytvořena z 1,2mm silného pozinkovaného plechu a k rámu je připevněna 4 šrouby. Víko a krabice jsou spojené přišroubovanými panty, aby vzájemnou polohou, aby bylo možné nastavit vůli ve spojení a víko na krabici přesně pasovalo. Na vrch víka je připevněn magnet, aby drželo zavřené.

Okapy nebylo možné jednoduše vyrobít do tvaru U bez předchozího výzkumu a vývoje. Proto bylo použito čtvercové řešení s větším poloměrem ohybu, což připomíná tvar U a jsou opatřeny uchy pro přichycení šrouby ke střeše. Čímž se okap přesadí směrem dovnitř skleníku.

Po zhotovení konstrukce skleníku bylo potřebné začít s osazováním elektroniky a všech funkčních částí. Jako první bylo přilepeno plexisklo do vnitřní strany střechy pomocí lepícího tmelu. A na vnitřní stranu střechy do štítu byl přidělán LED pásek pro osvětlování. Dále byla vyvrtán otvor do krabice na elektroniku, aby se umožnilo optimální vedení vodičů.

Následně na univerzálním plošném spoji byli realizovány všechny cesty pro napájení a vedení signálů ze senzorů a spínacího prvku. A k senzorům byly připájeny kabely, které vedly následně na univerzální plošný spoj umístěný v rozvodně. Připájením samotných kabelů k senzorům se zaručila větší pevnost celého projektu. Následně bylo vybráno to nejlepší umístění pro všechny aktivní elektronické části(ventilátor, pumpa, světla, senzory) viz obrázek. A následně jejich přichycení šrouby ke konstrukci skleníku.



Obrázek 24: Finální podoba skleníku

## 13 Závěr

Cílem práce bylo vytvoření „chytrého“ skleníku tj. stavby, která po prvotním nastavení bude schopna pěstované rostlině poskytnout, co nejlepší podmínky pro růst a zároveň zde bude kladen důraz na co nejmenší přičinění ze strany člověka. To vše s využitím moderních technologií a svěžího pohledu na problematiku pěstování rostlin v dnešní době.

Během cesty k vytvoření funkčního modelu jsem narazil na několik překážek. První problém byly mé nedostatečné znalosti v oblasti pěstování květin a technologiích pro to užívaných. Proto byl nutný nejdříve výzkum a sebevzdělání.

Prvotní fázi provázelo hledání a zkoumání jakým způsobem by se nechal celý projekt zhotovit. Po konzultacích s panem Fořtem, spolužáky a nastudování odborné literatury jsem došel k závěru, že nejlepší bude použít peristaltické čerpadlo pro zavlažování, které se bude spínat podle výstupní hodnoty z kapacitního měřiče vlhkosti. Původně jsem zamýšlel použití bistabilního ventilu namísto čerpadla, ale toto se následně ukázalo jako nereálné, protože pro plné otevření ventilu je nutný tlak alespoň 0,05MPa.

Jako externí zdroj světla jsem zvolil LED pásek jehož zapínání a vypínání závisí na množství světla dopadajícího na foto senzor TEMT6000 a také se řídí aktuálním časem, protože rostliny mají svůj vlastní cyklus a bylo by nežádoucí, aby např. v noci kdy hladina světla není dostatečná byla osvětlována celou noc.

Pro zjišťování vlhkosti a teploty vzduchu je využíván teplotní senzor HTU21D, který se ukázal jako ideální, ačkoliv jeho cena byla vyšší. Teplota a vlhkost vzduchu jsou následně vyhodnocovány v kódu a podle jejich hodnot se spíná ventilátor, protože však skleník nemá postranní stěny je regulace teploty a vlhkosti dosti omezená. Větrák slouží tedy spíše jako simulace možného ventilačního systému, který by užil v případě opravdové realizace.

Na závěr mohu zhodnotit, že snaha o vytvoření „chytrého“ skleníku byla úspěšná. Budova je schopna po prvotním nastavení sama pečovat o rostliny a poskytuje příznivé podmínky pro růst. Jsou zde však stále možnosti



k zefektivnění např. použití pouze jednoho zdroje pro napájení vývojové desky i spínaných částí. Dále použití dražších avšak kvalitnějších a přesnějších senzorů. Vylepšení konstrukce skleníku tak, aby bylo už přímo rozvrhnuta a designována na použití všech technologií a ujasnění přesně jejího užití, tedy jestli se bude neházet v interiéru nebo exteriéru.

Výhled do budoucna tkví v užití návrhu při opravdové realizaci „velkého“ skleníku pro komerční využití. Nebo alespoň implementování některé z technologií užitých v tomto projektu v zemědělství.

## 14 Zdroje a prostředky

Výpočet pull-up rezistorů [Online]. Dostupné z:

<https://atman-iot.com/blog/i2c-pull-up-calculator/>

Peristaltické čerpadlo [Online]. Dostupné z:

[https://cs.wikipedia.org/wiki/Peristaltick%C3%A9\\_%C4%8Derpadlo](https://cs.wikipedia.org/wiki/Peristaltick%C3%A9_%C4%8Derpadlo)

Využití pull-up rezistorů [Online]. Dostupné z:

[https://community.silabs.com/s/article/i2c-pull-up-resistor-calculation?language=en\\_US](https://community.silabs.com/s/article/i2c-pull-up-resistor-calculation?language=en_US)

Datasheet k HTU21D [Online]. Dostupné z:

<https://pdf1.alldatasheet.com/datasheet-pdf/view/880699/TEC/HTU21D.html>

Svetelný senzor [Online]. Dostupné z:

<https://cs.lambdageeks.com/light-sensors/>

1. Wikipedia. Wiring. [Online] 20. 4 2022

[https://cs.wikipedia.org/wiki/Wiring\\_\(programovac%C3%AD\\_jazyk\)](https://cs.wikipedia.org/wiki/Wiring_(programovac%C3%AD_jazyk)).

Informace od prodejece o senzoru vlhkosti [Online]. Dostupné z:

<https://dratek.cz/arduino/1487-iic-i2c-senzor-teploty-a-vlhkosti-htu21d.html>

Datasheet k senzoru TEMP6000 [Online]. Dostupné z:

<https://www.vishay.com/docs/81579/temt6000.pdf>

Popis sběrnice I<sup>2</sup>C – It network[Online]. Dostupné z:

<https://www.itnetwork.cz/hardware-pc/arduino/arduino-a-i2c-sbornice>

Popis fungování sběrnice – DH servis[Online]. Dostupné z:

<http://www.dhservis.cz/iic.htm>

Praktické využití I<sup>2</sup>C sběrnice – Vývoj HW[Online]. Dostupné z:

<https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbornice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-EEPROM-24LC256>

Popis principu fungování vlhkoměru – How to electronics [Online]. Dostupné z:

<https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/>

## 15 Seznam obrázků

Obrázek 1: Vývojová deska ESP32 .....	7
Obrázek 2: Průběh napětí na senzoru .....	8
Obrázek 3: Vnitřní schéma zapojení vlhkoměru .....	9
Obrázek 4: Půdní vlhkoměr .....	9
Obrázek 5: Senzor TEMT6000 .....	10
Obrázek 6: Základní charakteristiky pro tepelné a světelné podmínky .....	10
Obrázek 7: Časové průběhy na lince SCK .....	11
Obrázek 8: Senzor HTU21D .....	11
Obrázek 9: Peristaltické čerpadlo .....	12
Obrázek 10: Použitý ventilátor .....	13
Obrázek 11: Použité 4 kanálové relé .....	14
Obrázek 12: Jednoduché schéma popisující vnitřní konstrukci relé .....	14
Obrázek 13: Blokové schéma spínaného zdroje .....	15
Obrázek 14: Použitý síťový zdroj .....	16
Obrázek 15: Ukázka jednoduchého kódu pro rozsvícení led diody .....	17
Obrázek 16: Příklad možného vzhledu dashboardu .....	18
Obrázek 17: Datový rámec sběrnice I2C .....	20
Obrázek 18: Referenční model ISO/OSI .....	20
Obrázek 19 Typy skleníku .....	22
Obrázek 20: První návrh modelu .....	22
Obrázek 21: Druhý návrh skleníku .....	23
Obrázek 22: Finální návrh skleníku .....	23
Obrázek 23: Svařený a ohnutý rám .....	30
Obrázek 24: Finální podoba skleníku .....	31

## **16 Seznam příloh**

### **Příloha I.**

Soubor se schématy zapojení

### **Příloha II.**

Soubor s 3D modelem

### **Příloha III.**

Kód s programem