

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Automatizace skleníku pomocí Raspberry Pi

Bc. Martin Dayef

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Martin Dayef

Systémové inženýrství a informatika
Informatika

Název práce

Automatizace skleníku pomocí Raspberry Pi

Název anglicky

Automated greenhouse with Raspberry Pi

Cíle práce

Cílem práce je vytvoření automatizovaného skleníku s využitím mikropočítače Raspberry Pi, hardwarových komponent a vhodného softwaru. Pro ovládání skleníku a interpretaci naměřených hodnot ze senzorů bude vytvořeno webové uživatelské rozhraní.

Metodika

V první části práce se zaměřím na teoretická východiska. Pro jejich popsání bude základem nastudování odborné literatury týkající se daného tématu. V druhé části vyberu jednotlivé senzory a součástky potřebné pro automatizaci. Automatizovaný skleník bude řízený pomocí mikropočítače Raspberry Pi. Na mikropočítači bude nainstalován a naprogramován software potřebný pro komunikaci se senzory a ovládacími prvky. Díky mikropočítači bude možné interpretovat data ve vytvořeném uživatelském rozhraní. Z uživatelského rozhraní bude možné skleník ovládat. Na závěr posoudím výhody a nevýhody navrženého řešení.

Doporučený rozsah práce

50 – 60 stran

Klíčová slova

Raspberry Pi, IoT, skleník, automatizace, mikropočítač

Doporučené zdroje informací

COX, Tim a Dr. Steven Lawrence FERNANDES. Raspberry Pi 3 Cookbook for Python Programmers: Unleash the potential of Raspberry Pi 3 with over 100 recipes. 3rd Edition. Birmingham: Packt Publishing, 2018. ISBN 9781788629874.

DOW, Colin. Internet of Things Programming Projects: Build modern IoT solutions with the Raspberry Pi 3 and Python. Birmingham: Packt Publishing, 2018. ISBN 1789134803.

KARVINEN, Tero, Kimmo KARVINEN a Ville VALTOKARI. Make: sensors. Sebastopol, CA: Maker Media, 2014. ISBN 9351106373.

UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 978-80-251-4819-8.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Marek Pícka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 9. 3. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 3. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 30. 11. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Automatizace skleníku pomocí Raspberry Pi" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.11.2020

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Marku Píckovi, Ph.D. za rady a vedení při tvorbě diplomové práce. Dále bych chtěl poděkovat mým nejbližším za jejich podporu a trpělivost.

Automatizace skleníku pomocí Raspberry Pi

Abstrakt

Diplomová práce se zabývá vytvořením funkčního prototypu automatizace skleníku pomocí mikropočítače Raspberry Pi.

V literární rešerši jsou popsána hardwarová zařízení, open source software, komunikační protokoly a přenosové technologie, pomocí kterých je možné automatizace dosáhnout. Je zde podrobněji představen mikropočítač Raspberry Pi včetně jeho vývoje a specifikací. Z hardwarových zařízení jsou dále popsány mikrokontrolery ESP32, ESP8266 a senzor Xiaomi Mi Flora. Přiblížena je i problematika internetu věcí a jeho architektury. Dále jsou zpracovány informace o MQTT, bluetooth a dalších bezdrátových technologiích. Z open source softwaru je nastíněn nástroj Node-RED.

Poznátky uvedené v teoretických východiscích jsou použity ve vlastní práci. Nejprve je zpracovaná analýza, podle které je zhotoven návrh systému automatizace. Návrh je poté s využitím vhodných technologií aplikován. Dále je naprogramovaná senzorová stanice ESP32, vytvořena databáze v MySQL a uživatelské rozhraní s centrální aplikací v nástroji Node-RED. Centrální jednotkou umožňující automatizaci je mikropočítač Raspberry Pi. Následně byl vytvořen prototyp automatizovaného skleníku. Funkčnost tohoto prototypu je ověřena testováním. Závěrem jsou diskutovány výhody a případné nevýhody použitého řešení prototypu automatizovaného skleníku.

Klíčová slova: Raspberry Pi, IoT, skleník, ESP32, Node-RED, MQTT, MySQL, Bluetooth, automatizace, mikropočítač

Automated greenhouse with Raspberry Pi

Abstract

The diploma thesis focuses on the creation of a functional prototype of an automated greenhouse using Raspberry Pi.

The literature search describes hardware devices, open-source software and communication protocols and transmission technologies, which can be used to achieve automation. First are presented development and specifications of microcomputer Raspberry Pi. From the hardware devices are described ESP32 and ESP8266 microcontrollers or the Xiaomi Mi Flora sensor. The Internet of Things and its architecture are also approached. Followed by information about MQTT, bluetooth and other wireless technologies. The theoretical research also includes description of open-source software Node-RED.

The knowledge presented in the literature research is applied in own work. First is made analysis. According to analysis is designed automation system. The design is then applied using appropriate technologies, the ESP32 sensor station is programmed, a database is created in MySQL and a user interface with a central application in the Node-RED tool. The central unit for automation is the Raspberry Pi. Subsequently, a prototype of an automated greenhouse is created and tested. Finally, the system is applied to a prototype greenhouse, which is tested. At the end is evaluation of economic aspect, benefits and possible extensions.

Keywords: Raspberry Pi, IoT, greenhouse, ESP32, Node-RED, MQTT, MySQL, Bluetooth, automatization, microcomputer

Obsah

1 Úvod	12
2 Cíl práce a metodika	13
2.1 Cíl práce.....	13
2.2 Metodika.....	13
3 Teoretická východiska	14
3.1 Raspberry Pi	14
3.1.1 Generace Raspberry Pi	15
3.1.2 GPIO.....	15
3.2 HW zařízení.....	16
3.2.1 ESP 8266.....	16
3.2.2 ESP32	17
3.2.3 Arduino Uno.....	19
3.2.4 Elektromagnetické relé.....	20
3.2.5 Xiaomi Mi Flora.....	21
3.2.6 Nepájivé pole.....	21
3.3 Internet věcí – IoT	22
3.3.1 Internet (sít')	23
3.3.2 Věc (thing).....	23
3.3.3 Architektura.....	23
3.3.3.1 Cloud architektura.....	23
3.3.3.2 Fog architektura	24
3.3.3.3 Třivrstvá architektura.....	25
3.3.3.4 Pětivrstvá architektura	25
3.4 Komunikační protokoly a přenosové technologie	26
3.4.1 MQTT.....	26
3.4.2 Bluetooth	27
3.4.3 ESP – NOW	28
3.4.4 SSH.....	31
3.4.5 VNC	32
3.4.6 Wi-Fi	32
3.4.7 Ethernet	33
3.5 Python.....	33
3.6 Node-RED	34

4	Analýza požadavků vlastní práce.....	35
5	Návrh vlastního řešení.....	36
5.1	Architektura systému	36
5.2	Měřicí a ovládací stanice ESP32	37
5.3	Senzory měřicí stanice.....	38
5.3.1	Senzor teploty a vlhkosti vzduchu	38
5.3.2	Senzor rostlin.....	38
5.4	Ovládací prvky	39
5.4.1	Osvětlení.....	39
5.4.2	Zavlažování	39
5.4.3	Odvětrání.....	39
5.4.4	Spínací prvek.....	40
5.5	Databáze	40
5.6	Uživatelské rozhraní	41
6	Implementace	42
6.1	Softwarové řešení Raspberry Pi	42
6.1.1	Instalace.....	42
6.1.2	Konfigurace	42
6.1.3	Vzdálený přístup – Putty a VNC.....	43
6.1.4	MQTT broker (Eclipse Mosquitto)	43
6.2	MySQL databáze	43
6.3	Node-RED	44
6.3.1	Flow uložení MQTT zpráv do databáze.....	44
6.3.2	Flow pro nastavení proměnných systému	45
6.3.3	Flow zpracování MQTT zpráv	46
6.4	Webové rozhraní.....	46
6.5	ESP32	49
6.6	DHT22 – senzor.....	50
6.7	Xiaomi Mi Flora – senzor.....	51
6.8	Realizace HW	53
6.8.1	Relé modul	54
6.8.2	Větrák	55
6.8.3	Servo motor	56
6.8.4	Pěstební osvětlení.....	56
6.8.5	Zavlažování – vodní čerpadlo	57
6.8.6	Zdroj.....	57
6.9	Stavba skleníku a instalace systému automatizace.....	58

7 Testování.....	61
7.1 Test periferií	61
7.1.1 DHT22.....	61
7.1.2 Xiaomi Mi Flora.....	61
7.1.3 Relé modul a připojené ovládací prvky.....	62
7.1.4 Servo motor	62
7.2 Test vytvořených programů.....	62
7.3 Test automatizovaného systému	62
8 Zhodnocení nákladů	63
9 Závěr	65
10 Bibliografie	67

Seznam obrázků

Obrázek 1 - Raspberry Pi [1].....	14
Obrázek 2 - GPIO [5]	16
Obrázek 3 - Přehled čipů ESP8266 [12]	17
Obrázek 4 - Mikrokontroler ESP32-DevKitC s modulem ESP32-WROOM-32 [15]	18
Obrázek 5 - Arduino Uno R3[18]	19
Obrázek 6 - Elektromagnetické relé [22]	20
Obrázek 7 - popis Xiaomi Mi Flora [25].....	21
Obrázek 8- Fog architektura [31]	24
Obrázek 9 - Třívrstvá architektura [31].....	25
Obrázek 10 - Komunikace přes MQTT [35]	27
Obrázek 11 - Komunikace přes bluetooth [37]	28
Obrázek 12 - Komunikace jedním směrem – více ESP odesílá data do jedné desky [42].....	30
Obrázek 13 - Kroky připojení SSH [43]	31
Obrázek 14 - Architektura systému automatizace.....	36
Obrázek 15 - Způsob komunikace mezi prvky systému	37
Obrázek 16 - Senzor DHT22 [60]	38
Obrázek 17 - Databáze systému	41
Obrázek 18 - Wireframe domovské obrazovky uživatelského rozhraní	41
Obrázek 19 - Flow uložení komunikace MQTT	44

Obrázek 20 - Flow nastavení proměnných systému.....	45
Obrázek 21 - Zdrojový kód – ukázka funkce nastavení systémových proměnných	45
Obrázek 22 - Flow pro zpracování MQTT zpráv na téma teploty ovzduší.....	46
Obrázek 23 - Uživatelské rozhraní – úvodní stránka	47
Obrázek 24 - Funkce pro zformátování dat.....	48
Obrázek 25 - Uživatelské rozhraní – výpis proběhlé komunikace přes MQTT.....	48
Obrázek 26 - Uživatelské rozhraní - nastavení mezních hodnot systému.....	49
Obrázek 27 - Zdrojový kód - připojení ESP32 k WiFi	49
Obrázek 28 - Zdrojový kód – ukázka callback funkce.....	50
Obrázek 29 - Zdrojový kód – získání dat ze senzoru DHT22.....	51
Obrázek 30 - přehled UUID charakteristik Xiaomi Mi Flora [61].....	52
Obrázek 31 - Zdrojový kód – získání stavu baterie Xiaomi Mi Flora	53
Obrázek 32 - Návrh zapojení hardware přes nepájivé pole	54
Obrázek 33 - Relé modul v testovacím zapojení.....	55
Obrázek 34 - Větrák pro regulaci teploty	56
Obrázek 35 - Konstrukce LED osvětlení	57
Obrázek 36 - Napájecí zdroj s vyhotovenou kabeláží.....	58
Obrázek 37 - Prototyp skleníku - pohled zevnitř	59
Obrázek 38 - Prototyp skleníku – čelní pohled	60

Seznam tabulek

Tabulka 1 - Generace Raspberry Pi [3].....	15
Tabulka 2 - popis komponent ESP32 [15]	18
Tabulka 3 - Akční rámec ESP-NOW[41]	29
Tabulka 4 - Vendor specific content ESP-NOW [41].....	29
Tabulka 5 - Náklady na vyhotovení práce.....	63

1 Úvod

S rostoucí popularitou internetu věcí a chytrých domácností dochází ke stále větší poptávce po automatizaci běžných úkonů, popř. jejich ulehčení. Automatizace přináší lidem řadu benefitů. Jedním z nich je úspora času – ten se v dnešní době více než kdy dříve stal drahocennou komoditou. Chytrá zařízení a automatizace však nejsou jediným fenoménem posledních let. Stále více se mluví o ekologii, udržitelnosti a úspoře přírodních zdrojů. K úspoře přírodních zdrojů může přispívat právě automatizace procesů pomocí vyspělých technologií.

S technologickým pokrokem se elektronika, elektronické součástky, mikrokontrolery, mikropočítače stávají levnější a dostupnější. Díky tomu se na trhu vyskytují od různých výrobců mnohé výrobky zajišťující automatizaci. Ovšem cena těchto hotových výrobků je celkem vysoká. Vzhledem k již zmiňované snazší dostupnosti jednotlivých součástí, senzorů a prvků je však možné vyvinout vlastní řešení. Využít lze i mnoha open source nástrojů, okolo kterých se mnohdy utvořily velké komunity. Dostupnost informací k těmto prostředkům je dobrá. Na internetu a v odborné literatuře lze často najít velké množství praktických rad, návodů nebo sérii projektů využívající tyto technologie. Výhodou vlastního řešení je zejména nižší cena a fakt, že zpracovatel ví, co chce, aby systém dělal. Zároveň je schopný i nadále rozšiřovat stávající systém o další prvky a udržet ho tak stále aktuální.

Ve své diplomové práci bych chtěl využít dostupných technologií a open source nástrojů k vytvoření vlastního systému pro automatizaci pěstování v domácím skleníku.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je vytvoření prototypu automatizovaného skleníku. Prototyp bude možné použít k aplikaci na automatizaci reálného skleníku. Automatizace bude vytvořena s využitím mikropočítače Raspberry Pi. Ovládání a měření hodnot skleníku zajistí vhodné hardwarové komponenty a software. Pro ovládání skleníku a interpretaci naměřených hodnot ze senzorů bude vytvořeno webové uživatelské rozhraní. Funkčnost automatizovaného prototypu skleníku bude ověřena testováním. Závěrem budou diskutovány výhody a případné nevýhody použitého řešení prototypu automatizovaného skleníku.

2.2 Metodika

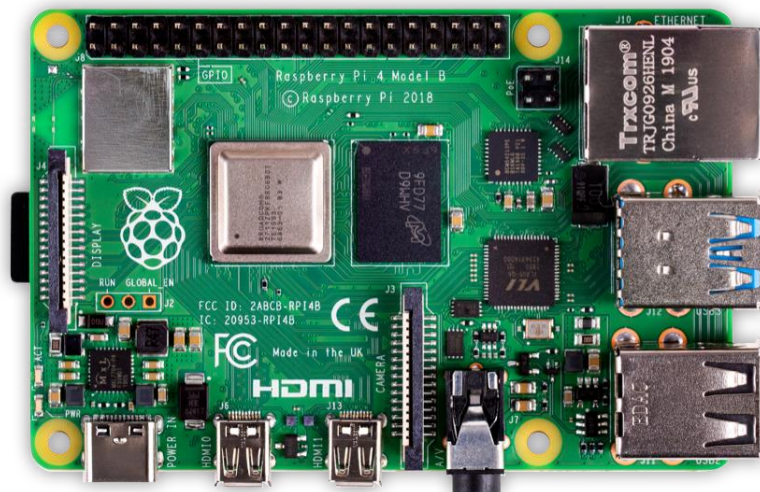
V první části své diplomové práce se zaměřím na teoretická východiska. Pro jejich popsání bude základem nastudování odborné literatury týkající se daného tématu. V druhé části nejdříve zanalyzuji požadavky na mou práci a poté navrhnu způsob jejího řešení. Vyberu jednotlivá zařízení, senzory a součástky potřebné pro automatizaci. Prototyp automatizovaného skleníku bude řízený pomocí mikropočítače Raspberry Pi. Na mikropočítači bude nejprve nainstalován operační systém a poté potřebný software. K dlouhodobému ukládání dat bude využita databáze MySQL. Následovat bude zapojení hardwarových prvků podle navrženého řešení. Mikrokontroler sensorové stanice bude ESP32. Ten bude ovládat akční prvky skleníku a získávat naměřená data ze senzorů. Po propojení prvků bude naprogramován a implementován software potřebný pro jejich komunikaci. Zařízení bude bezdrátově komunikovat s řídicí jednotkou Raspberry Pi přes protokol MQTT. Centrální aplikace a řídicí procesy systému budou vytvořeny v nástroji Node-RED. Pro umožnění interakce uživatele se systémem bude navrženo a vytvořeno uživatelské rozhraní, ze kterého bude možné skleník ovládat a interpretovat v něm naměřená data. K vytvoření uživatelského rozhraní bude využito nástroje Node-RED. Jako poslední bude vytvořen model skleníku, do kterého bude nainstalován systém automatizace. K ověření funkčnosti automatizovaného skleníku a jeho systému proběhnou jednotlivá testování. Na závěr posoudím výhody a případné nevýhody navrženého řešení.

3 Teoretická východiska

3.1 Raspberry Pi

Raspberry Pi je malý počítač o velikosti kreditní karty. Jako operační systém je možné využít Linux distribuce nebo Windows IoT Core. Výkonově stačí na základní stolní počítač slabšího výkonu. V případě, že je využíván jako stolní počítač, zvládne vše od prohlížení internetu přes zpracování dokumentů až po streamování hudby a sledování videa ve vysokém rozlišení. Díky dostupnosti a jednoduchosti je ideálním startovacím zařízením pro pochopení fungování počítačů a pro výuku programování, např. v podporovaném jazyce Python. Zároveň je široce využíván jako multimediální centrum a domácí asistent pro automatizační projekty. Propojení s periferiemi je možné skrze konektory USB, HDMI, zvukový 3,5 mm konektor nebo RJ-45 Ethernet.

Mikropočítače Raspberry Pi vyvíjí nadace Raspberry Pi Foundation, která vznikla v roce 2009 ve Velké Británii. Cílem nadace je rozvíjet vzdělávání dospělých a dětí, zejména v oblasti počítačů, informatiky a souvisejících předmětů. Nadace se snaží poskytovat levné, a přesto vysoce výkonné počítače, které lidé mohou používat k učení, řešení problémů a k zábavě. Nadace také poskytuje bezplatné zdroje, které lidem pomáhají učit se o práci na počítači. Zajišťuje i různá školení pro pedagogy, kteří poté mohou své studenty o technologiích vzdělávat. Platforma se setkala s úspěchem nejen ve vzdělávání, ale také v domácím prostředí a průmyslovém využití. [1] [2] [3] [4]



Obrázek 1 - Raspberry Pi [1]

3.1.1 Generace Raspberry Pi

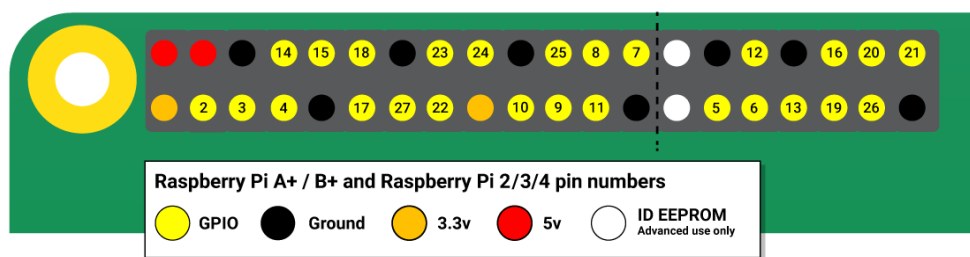
Generace	Datum vydání	Architektura	SoC	Procesor	Operační paměť	Síťové připojení	GPIO	Energetická náročnost	Rozměry		
Model A 1	01.02.2013			700 MHz ARM1176JZF-S	256 MB	X	8×	300 mA (1,5 W)	85.6 mm × 56.5 mm		
Model A 1+	01.11.2014				17×		200 mA (1 W)	65 mm × 56.5 mm			
Model B 1	4 - 6 2012				ARMv6 (32-bit)	Broadcom BCM2835	512 MB	Ethernet	8×	700 mA (3,5 W)	85.60 mm × 56.5 mm
Model B 1+	01.07.2014								17×	600 mA (3,0 W)	
Model B 2	01.02.2015	ARMv7 (32-bit)	Broadcom BCM2836	900 MHz 32-bit ARM Cortex-A7					17×	800 mA (4,0 W)	
Model B 3	01.02.2016	ARMv8 (64/32-bit)	Broadcom BCM2837	1,2 GHz 64-bit ARM Cortex-A53	1 GB	Ethernet, Wi-Fi 802.11n, Bluetooth 4.1	200 mA (1 W)				
Model B 3+	01.03.2018	ARMv8 (64/32-bit)	Broadcom BCM2837B0	1.4 GHz 64-bit Cortex-A53		Ethernet, Wi-Fi 802.11ac, Bluetooth 4.2 LS BLE	17×	459 mA (2.295 W)			
Model B 4	01.06.2019	ARMv8 (64-bit)	Broadcom BCM2711	1,5 GHz 64-bit Cortex-A72	1 - 8 GB	Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0, BLE		3A (15W)			
Zero 1.2	01.11.2015	ARMv6 (32-bit)	Broadcom BCM2835	1 GHz ARM1176JZF-S	512 MB	Wi-Fi 802.11n, Bluetooth 4.1	26×	~ 160 mA (0,8 W)	65 mm × 30 mm		
Zero 1.3	01.05.2016										
Zero W	01.02.2017										

Tabulka 1 - Generace Raspberry Pi [3]

3.1.2 GPIO

Název je zkratkou anglického označení General – Purpose Input/Output, tedy univerzálně použitelný vstup/výstup. Jedná se o vstupně/výstupní piny, přes které je možné připojit k Raspberry Pi další elektronické periferie. Samozřejmostí je možnost tyto periferie ovládat skrze knihovny. Knihovny podporují programovací jazyky Python, Java, C a jiné. V současnosti je na novějších modelech Raspberry Pi již 40 pinů, přičemž zůstává zpětná podpora starších modelů, které měly pinů 26.

Piny fungují ve stavu logické 0 (low) nebo 1 (high), které se nastavují pomocí napětí 0 V nebo 3,3 V. Vzhledem k tomu, že nejsou chráněny proti vyššímu napětí, může při překročení 3,3 V dojít k poškození součástek na desce a k nefunkčnosti Raspberry Pi.



Obrázek 2 - GPIO [5]

Rozložení pinů je zobrazeno na obrázku výše. Červeně označené piny jsou určeny jako výstupní s napětím 5 V. Oranžové piny jsou určeny k výstupnímu napájení 3,3 V. Černě označené piny slouží jako zem. Bílé piny značené jako ID EEPROM jsou určeny k manipulaci s EEPROM pamětí. Ostatní žlutě vyznačené vstupně/výstupní piny GPIO je možné nastavit šesti funkcemi. [5] [6]

3.2 HW zařízení

3.2.1 ESP 8266

Jako ESP8266 je označována série bezdrátových Wi-Fi mikrokontrolerů. Nejčastěji jsou desky založené na čipu ESP8266EX, popř. i na ESP8285. Oba čipy jsou od výrobce Espressif Systems.

ESP8266EX (ve většině případů označován pouze jako ESP8266) je tzv. system-on-chip (SoC). SoC je integrovaný obvod zahrnující CPU, paměť, vstupně/výstupní porty a častokrát dodatečné úložiště. ESP8266EX integruje 32-bit mikrokontroler Tensilica, rozhraní pro periferie, vysokofrekvenční stabilizátor, výkonový zesilovač a LNA. Dále je zahrnuta 2,4 GHz Wi-Fi (802.11 b/g/n/, podporující WPA/WPA2), 16 GPIO (univerzální vstup/výstup), I²C, převodník analog-digital (10-bit ADC), SPI (sériově periferní rozhraní), I²S rozhraní s DMA (sdílí piny s GPIO), UART (univerzální asynchronní přijímač/vysílač – pouze na určených pinech) a modulátor šířky pulzu PWM. 32bitové jádro procesoru Tensilica L106 funguje na frekvenci 80 MHz s možností přetaktování až na 160 MHz. Paměti jsou zde ROM a SRAM. Velikost RAM je ve station módu maximálně 50 kB. Jelikož se jedná o SoC jednotku, není zde programovatelná ROM. Uživatelské programy tak musí být uloženy na externí až 16 MB velké SPI flash paměti. Vše, co čip integruje zvládl výrobce vměstnat na desku o velikosti 5 mm x 5

mm. Dříve zmiňovaný čip ESP8285 má oproti ESP8266EX výhodu integrované flash paměti o velikosti až 2 MB.

Na obrázku níže jsou zachyceny vybrané verze mikrokontrolerů s ESP8266EX. ESP-01 je možné využít jako rozšíření k Arduino Uno. To je poté schopno připojit se bezdrátově k internetu. Wemos D1 Mini je již oproti tomu vybaveno 4 MB flash pamětí, jedenácti piny GPIO, microUSB konektorem a širokou škálou různých štítů tzv. shieldů. Shield je elektronický obvod, který rozšiřuje funkcionalitu desky. Je tak možné například přidat displej, externí úložiště, relé, baterii a další. [7] [8] [9] [10] [11]



Obrázek 3 - Přehled čipů ESP8266 [12]

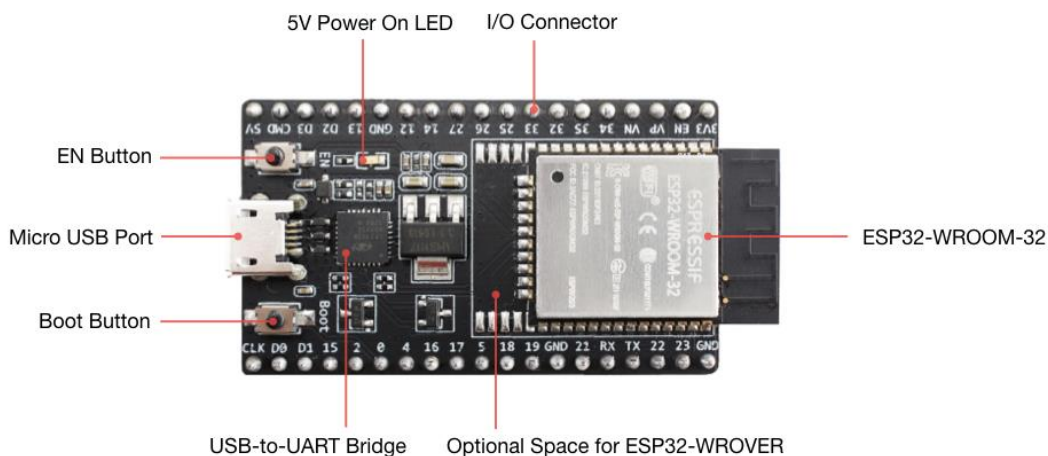
3.2.2 ESP32

ESP32 je čip od čínského výrobce Espressif Systems. Je nástupcem široce využívaného ESP8266, který na rozdíl od ESP32 nepodporoval Bluetooth. Mikrokontroler používá dvoujádrové mikroprocesory 32-bit Xtensa LX6 a SRAM paměť o velikosti 512 kB. Jednou z důležitých vlastností je více napájecích módů, ve kterých může čip fungovat. Je možné ho dle potřeby pravidelně probouzet ze spánku, aby čip vykonával určenou funkci. Díky této vlastnosti je zařízení málo energeticky náročné. Mikrokontroler podporuje protokol Wi-Fi 802.11 b/g/n s rychlostí až 150 Mb/s. Důležitou vlastností čipu je možnost využití Bluetooth 4.2 s podporou

BLE pro komunikaci se zařízeními. Na obrázku níže je popsán mikrokontroler ESP32 – DevKitC od Espressif s modulem ESP32–WROOM-32. Popis klíčových komponent je v tabulce níže. [13] [14] [15] [16]

Komponenta	Popis
ESP32-WROOM-32	Modul s čipem ESP32.
EN	Tlačítko pro reset.
Boot	Tlačítko pro nahrání programu. Podržení Boot a následně stisknutí EN uvede zařízení do módu stažení firmware skrze sériový port.
USB-to-UART Bridge	Čip zajišťuje přenos v rychlosti až 3 Mb/s.
Micro USB Port	USB prostředí přes které je možné se zařízením komunikovat z počítače nebo ho napájet.
5V Power On LED	Rozsvítí se, když je k mikrokontroleru připojené USB nebo zdroj 5V.
I/O	Vstupně/výstupní piny. Většina pinů je na desce otevřená a lze je tak napřímo připojit k dalším zařízením. Je přes ně možné ESP32 naprogramovat k využití mnoha funkcí jako jsou PWM, ADC, DAC, I2C, I2S, SPI a další.

Tabulka 2 - popis komponent ESP32 [15]

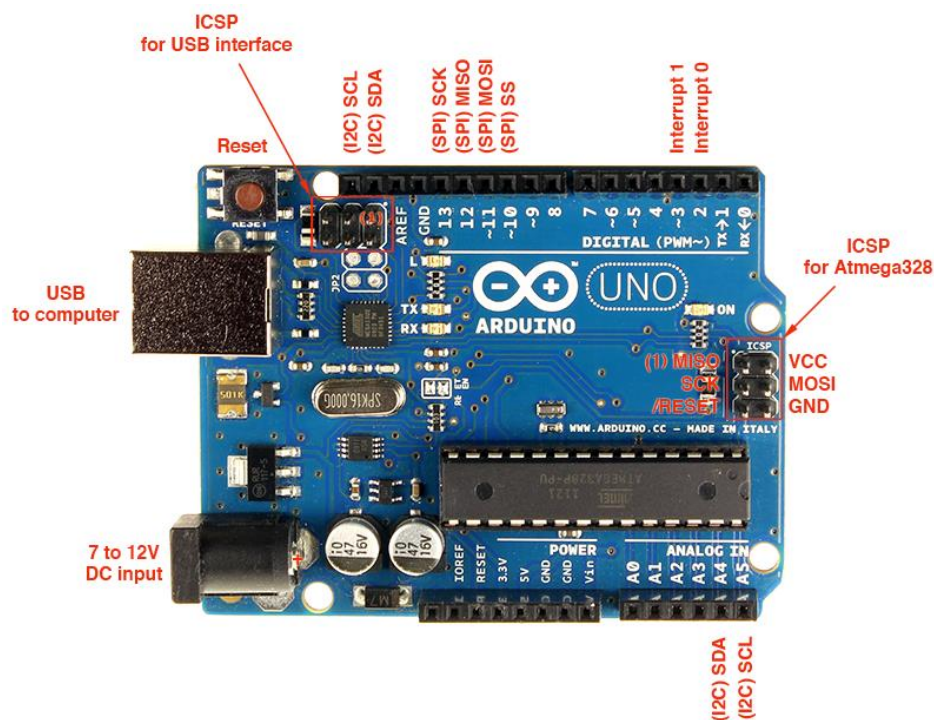


Obrázek 4 - Mikrokontroler ESP32-DevKitC s modulem ESP32–WROOM-32 [15]

3.2.3 Arduino Uno

Arduino je open-source platforma založená na jednoduše použitelném hardwaru a softwaru. Vývojové prostředí je dostupné napříč operačními systémy od Windows, přes macOS až k Linux. Právě dostupnost, jednoduchost i sdílení návodů a schémat stojí za velkým rozšířením platformy. Arduino vytvořil italský Interaction Design Institute, jež platformu začal vyvíjet už v roce 2005. Cílem vývoje byl jednoduchý a levný vývojový set pro studenty, pro které byly jiné desky (například Parallax Basic Stamp) drahé a nedostupné.

Typů desek existuje nepřehledné množství, nejvíce používaným typem je však Arduino Uno. O chod této desky se stará mikroprocesor ATmega328 od výrobce Atmel. Programy se nahrávají do malé paměti o velikosti 32 KB, ze které je ještě 0,5 KB využíváno bootloaderem. Nahrávání programů do mikrokontroleru probíhá nejčastěji přes USB port. Přes USB je možné Arduino Uno také napájet. Vhodnějším způsobem je však napájení přes souosý konektor, který vyžaduje zapojení adaptéru s doporučeným napětím 7 až 12 V (DC) a proudem do 1000 mA. Deska je vybavena čtrnácti digitálními vstupně výstupními piny a 6 analogovými piny. Digitální piny slouží k připojení modulů, senzorů nebo případně shieldů, které mohou rozšířit funkcionalitu nebo konektivitu desky. [17] [18] [19] [20]



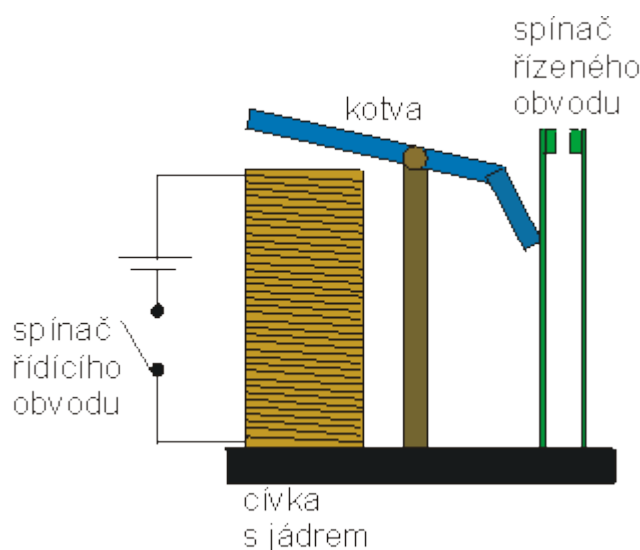
Obrázek 5 - Arduino Uno R3[18]

3.2.4 Elektromagnetické relé

Elektromagnetické relé je spínač, který je ovládaný slaboproudým signálem. Slouží ke spínání v silnoproudých obvodech. Zvládne tedy zapnout nebo vypnout přívod elektřiny k zařízení, které silnoproud využívá. Jak vyplývá z obrázku níže, relé se skládá z cívky s jádrem z feromagnetického materiálu, kotvy, spínače řízeného obvodu a spínače řídicího obvodu. Ve chvíli, kdy dojde k sepnutí řídicího obvodu, je na cívku přivedeno napětí. Cívka pod napětím vytvoří magnetické pole, které k cívce přitáhne kotvu. Kotva v tomto stavu sepne, rozepne nebo přepne řízený obvod. V případě, že dojde k rozeznutí řídicího obvodu a na cívku přestane být přiváděno napětí, cívka přestane vytvářet magnetické pole. Kotva se tak díky pružině vrátí do původní polohy a sepne, rozepne nebo přepne řízený obvod.

Relé může fungovat ve více stavech, které jsou určeny způsobem navržení. Prvním je normálně otevřený obvod. Tento typ, při rozeznutém řídicím obvodu, nepřivádí do cívky napětí a nespíná obvod řízený. Druhým typem je normálně zavřený obvod, který funguje opačně, tzn., že ve chvíli, kdy sepne řídicí obvod, dojde k rozeznutí řízeného obvodu. Řízený obvod je tedy v tomto případě sepnutý, když je řídicí obvod rozeznutý.

Elektromagnetické relé není jediným typem relé. Alternativou může být polovodičové, které spíná rychleji než elektromagnetické relé a vydrží více spínání. Často je však u polovodičového relé nutné dodatečné chlazení. Dalším negativem je cena, která mnohokrát převyšuje cenu elektromagnetického relé. [21] [22]



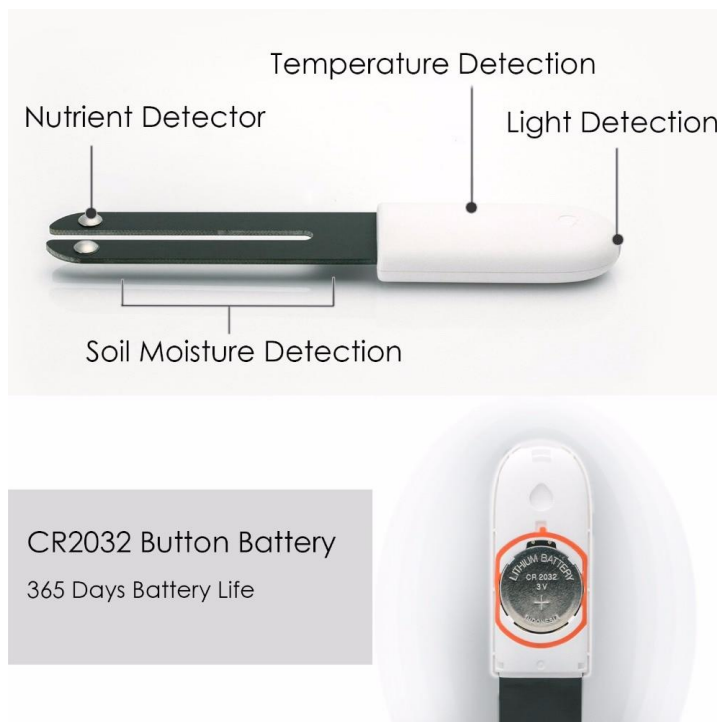
Obrázek 6 - Elektromagnetické relé [22]

3.2.5 Xiaomi Mi Flora

Xiaomi Mi Flora je zařízení pro měření půdních podmínek u rostlin a keřů. Jeho funkční design (viz obrázek níže) je velmi nadčasový a praktický. Díky němu se tento nástroj ztratí v každém květináči, popř. skleníku. Má vysokou přesnost naměřených hodnot a snadné ovládání. Je možné ho zapojit do domácí sítě chytrých zařízení a ovládat ho skrze mobilní aplikaci. Veškerá komunikace se zařízením probíhá přes Bluetooth (podpora 4.1). Napájení obstarává 1 x DC 3V CR2032 baterie na kterou je zařízení schopné vydržet funkční až dvanáct měsíců. [23] [24]

Zařízení obsahuje:

- Senzor vlhkosti a pH půdy
- Senzor světla s rozsahem 0 - 100000 lm a přesností měření 100 lm
- Teplotní senzor



Obrázek 7 - popis Xiaomi Mi Flora [25]

3.2.6 Nepájivé pole

Nepájivé pole je při propojování elektronických obvodů vyhledávaná pomůcka. Pomocí tohoto pole je možné vytvářet obvody bez nutnosti pájení. Je tak vhodné pro vývoj a testování navržených obvodů. Nepájivé pole lze propojovat mezi sebou. Také je možné je dostat

v různých velikostech i variantách. Pole funguje na principu propojování jednotlivých kovových svorek. Žlutě vyznačené zdírky jsou propojeny horizontálně (Connected Horizontally) a používají se pro připojení součástek k napájení (většinou jsou umístěny po stranách pole). Uprostřed pole jsou umístěny modře označené zdírky, které jsou vertikálně propojeny (Connected Vertically) po menším počtu. Propojování součástek v poli s jednotlivými svorkami je realizováno pomocí flexibilních propojovacích kabelů. Ty mohou mít na obou stranách koncovky typu samec – samec, samec – samice nebo samice – samice. Pro odlišení účelu mohou být propojovací kabely barevně odlišeny. [26] [27]

3.3 Internet věcí – IoT

Označení Internet věcí vznikl z překladu anglického Internet of things (dále jen v anglické zkratce IoT). Tento pojem se v posledních letech zmiňuje stále častěji.

“Internet věcí znamená: síť propojených objektů (věcí), které jsou jednoznačně adresovatelné s tím, že tato síť je založena na standardizovaných komunikačních protokolech umožňující výměnu a sdílení dat a informací, jejichž analýzou bude možné docílit vyšší přidané hodnoty“. [28]

Tedy, že koncept IoT je propojením věcí (fyzických nebo virtuálních objektů) vyměňujících si data skrze internetovou síť. Tyto objekty mohou být v rámci sítě IoT pospojovány tak, aby bylo dosaženo nové funkcionality, řešení složitějších procesů či vyššího cíle. IoT spojuje mnoho technologických odvětví. Jsou zde integrována pokročilá elektronická zařízení, telekomunikační technologie, internetová bezpečnost, cloudové služby a další technologie.

Již v roce 1999 se jako první zmiňoval Kevin Ashton o konceptu IoT. Upozornil na to, že pro lepší pochopení světa lze využít různých senzorů i jiných zařízení a to za předpokladu, že tyto nástroje mezi sebou i s jinými systémy dokáží sdílet data. Milníkem pro IoT byl přelom roku 2008 a 2009, kdy podle technologické společnosti Cisco došlo k překonání počtu světové populace počtem objektů propojených přes internet. Podle posledních odhadů by v roce 2020 mělo dojít k dosažení další hranice, kdy polovina zařízení připojených k internetu bude jiná než počítače, tablety a chytré mobilní telefony. [29] [30]

3.3.1 Internet (sít')

Sítí (Internet) se v IoT nerozumí pouze připojení ke globální internetové síti. Může se jednat o jakoukoliv počítačovou síť. Může jí být i lokální síť LAN, kdy zařízení komunikují pouze po této uzavřené místní síti a nesdílí data dále po internetu. Díky internetu je však možné využívat velké množství služeb, např. vzdálená komunikace se zařízeními na LAN síti nebo je možné se dostat k aktuálním datům ze zařízení. [31] [29]

3.3.2 Věc (thing)

Věcí se rozumí fyzický nebo virtuální objekt neživé povahy, vybavený senzory, elektronickými součástkami a programy, které slouží ke sběru dat. Data jsou dále v rámci sítě sdílána s ostatními věcmi a systémy, interpretována a využívána k naplnění cíle věci. Základem internetu věcí však nejsou zařízení, ale jimi poskytovaná data. [29]

3.3.3 Architektura

Popisem a návrhy architektur IoT se zabývala spousta specialistů v tomto oboru. Není však stanovena žádná přesná norma architektury IoT, tzn., že není pevně dané, jak má vypadat a z čeho se skládat. Níže jsou popsány čtyři základní typy architektury využívané v IoT:

- Cloud
- Fog
- Třívrstvá
- Pětivrstvá [32] [31]

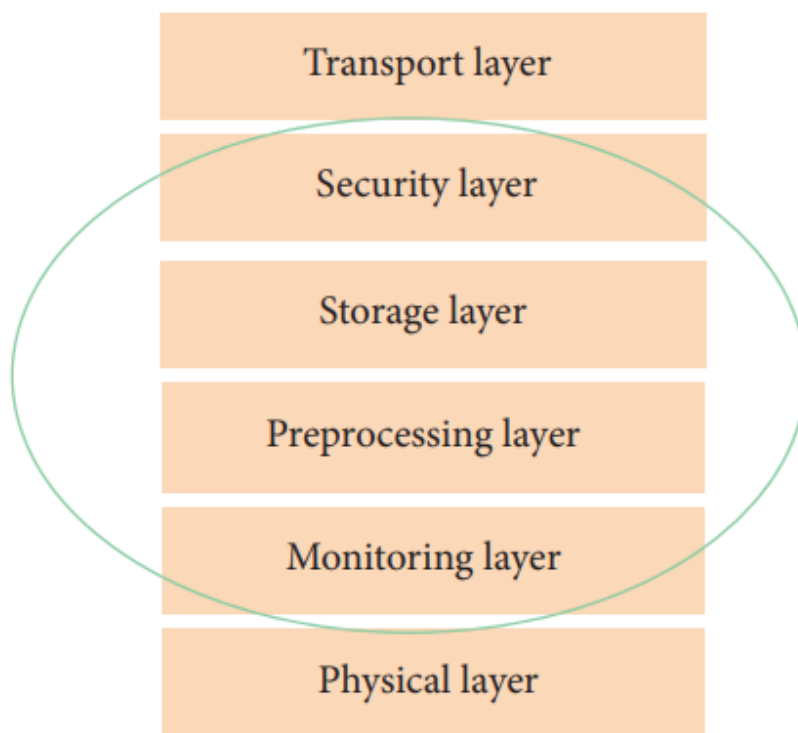
3.3.3.1 Cloud architektura

Cloud architektura je založena na udržování cloudu v centru architektury, kdy aplikační vrstva je nad ním a transportní vrstva pod ním. Tato architektura nabízí, stejně jako využití cloudových služeb v jiných oblastech IT, lepší škálovatelnost a flexibilitu oproti jiným řešením. Poskytuje úložiště, základní infrastrukturu, platformu a software. Díky tomu mají vývojáři velké množství možností a mohou využít softwarové nástroje pro rychlý vývoj, machine learning, data mining a vizualizaci dat. [31] [32]

3.3.3.2 Fog architektura

Fog architektura začíná být v poslední době stále více využívána. Ve fog computingu spolu zařízení komunikují napřímo (označováno jako peer-to-peer). V této architektuře jsou síťové prvky a senzory součástí zpracování a analýzy dat. Mezi fyzickou a transportní vrstvou jsou vsunuty monitorovací, předprocesová, úložná a bezpečnostní vrstvy. Monitorování a předzpracování je provedeno na okraji sítě ještě před odesláním dat do cloudu. Tyto vrstvy jsou znázorněny na obrázku níže a popsány následovně:

- Monitorovací vrstva (Monitoring layer) sleduje výkon, odpovědi, zdroje a služby.
- Předprocesová vrstva (preprocessing layer), neboli vrstva předběžného zpracování zajišťuje filtraci, analýzu a zpracování dat ze senzorů.
- Úložná vrstva (storage layer) distribuuje, ukládá a replikuje data.
- Bezpečnostní vrstva (security layer) šifruje nebo dešifruje data při zachování integrity a soukromí dat. [31] [32] [33]

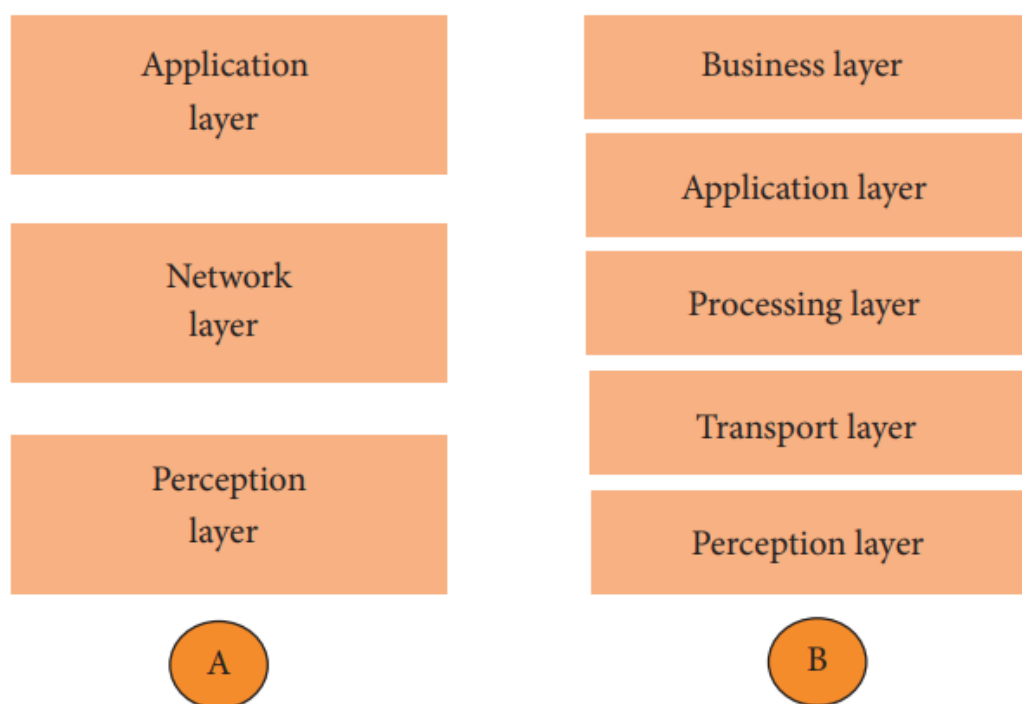


Obrázek 8- Fog architektura [31]

3.3.3.3 Třivrstvá architektura

Tato architektura je využívána již od počátků IoT. Jak napovídá název, je složena ze tří vrstev - fyzické, síťové a aplikační. Na obrázku níže v levé části A jsou vrstvy této architektury jednotlivě vyobrazeny v porovnání s pětivrstvou architekturou v pravé části B. Popsány jsou jako:

- Fyzická vrstva (perception layer) měří a sbírá data z okolí.
- Síťová vrstva (network layer) je nejvíce využívána ke zpracování a přenosu dat. Zajišťuje také konektivitu chytrých zařízení, senzorů a serverů.
- Aplikační vrstva (application layer) je zodpovědná za prezentaci dat uživateli a interakci s ním. [31] [32]



Obrázek 9 - Třivrstvá architektura [31]

3.3.3.4 Pětivrstvá architektura

Tato architektura je založena na principech třivrstvé architektury a jejího rozložení. Oproti třivrstvé architektuře tu jsou navíc obchodní, procesní a transportní vrstvy, které doplnily původní zachované vrstvy fyzickou a aplikační. Rozdíly ve vrstvách architektur jsou zachyceny

na předchozím obrázku, kde A je třívrstvá architektura a B pětivrstvá. Níže jsou popsány jednotlivé vrstvy a jejich funkce, ze kterých je architektura složena:

- Obchodní vrstva (business layer) řídí celý systém internetu věcí včetně aplikací, soukromí uživatelů a obchodních modelů.
- Aplikační vrstva (application layer) je stejně jako u třívrstvé architektury zodpovědná za prezentaci dat uživateli a interakci s ním.
- Procesní vrstva (processing layer) je schopna spravovat a poskytovat služby ze spodních vrstev. Transportní vrstva poskytuje velké množství dat, které procesní vrstva zpracovává, analyzuje a ukládá. Pro vykonávání své funkcionality používá technologie cloud computingu, databáze a zpracování big data.
- Transportní vrstva (transport layer) je transportním kanálem dat mezi fyzickou a procesní vrstvou. K přenosu dat používá přenos prostřednictvím sítí, kterými mohou být LAN, NFC, Bluetooth, 3G a další bezdrátové technologie.
- Fyzická vrstva (perception layer) stejně jako u třívrstvé architektury měří a sbírá data z okolí. [31] [32]

3.4 Komunikační protokoly a přenosové technologie

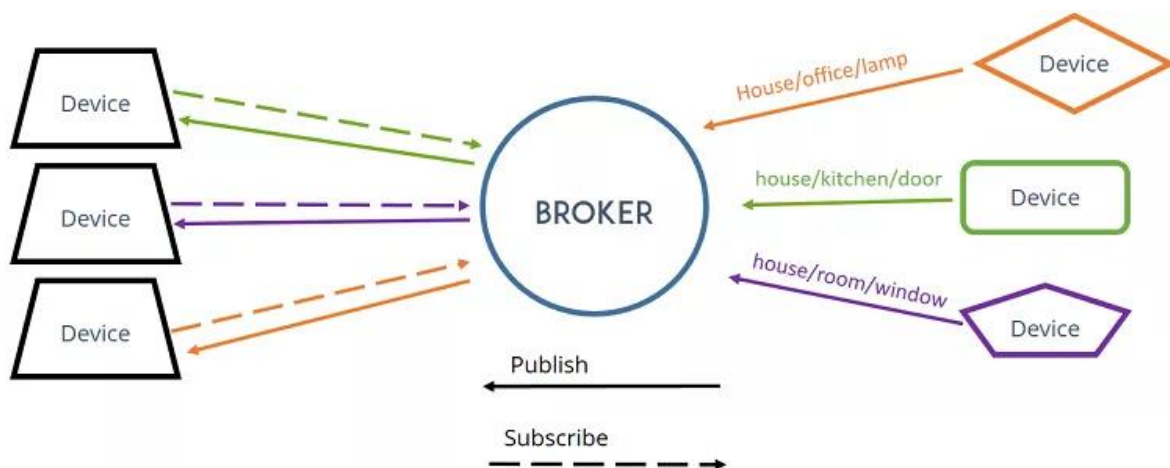
V této části jsou popsány a specifikovány technologie, které je v IoT možné využít k přenosu dat nebo ke komunikaci mezi elektronickými zařízeními.

3.4.1 MQTT

Název protokolu je zkratka z anglického názvu Message Queuing Telemetry Transport. Jedná se o jednoduchý protokol komunikující pomocí zpráv. Je vhodný pro zařízení s malou šířkou pásma. Umožňuje zasílání příkazů na zařízení, čtení a sdílení dat ze senzorů. Díky těmto vlastnostem je široce využívaným pro komunikaci zařízení v internetu věcí. MQTT funguje ve čtyřech základních stavech:

- Publish/subscribe (publikovat/odebírat). V systému publikování a odběru může zařízení publikovat zprávu na téma (topic) nebo ji může přihlásit k odběru určitého tématu pro příjem zpráv.

- Messages (zprávy). Zprávy jsou informace, které se vyměňují mezi zařízeními. Zprávou může být jakýkoliv příkaz k provedení na periférii v internetu věcí nebo data k přenesení.
- Topics (témata). Další důležitou funkcí jsou témata. Témata jsou způsob, jakým je registrován zájem o příchozí zprávy nebo jak určujeme, kde chceme zprávu zveřejnit. Témata jsou reprezentována textovými řetězci, které jsou odděleny lomítkem. Lomítka označují jednotlivé úrovně obsahu. Řetězce témat jsou citlivé na velká a malá písmena.
- Broker (zprostředkovatel). Broker zprostředkovává komunikaci mezi zařízeními přes protokol. Zajišťuje přenos zpráv, které přijímá a distribuuje dále na zařízení přihlášená k odběru. Brokerů existuje mnoho. Je možné využít cloudová řešení (CloudMQTT) nebo jiné alternativy jako je Mosquitto. [34] [35]



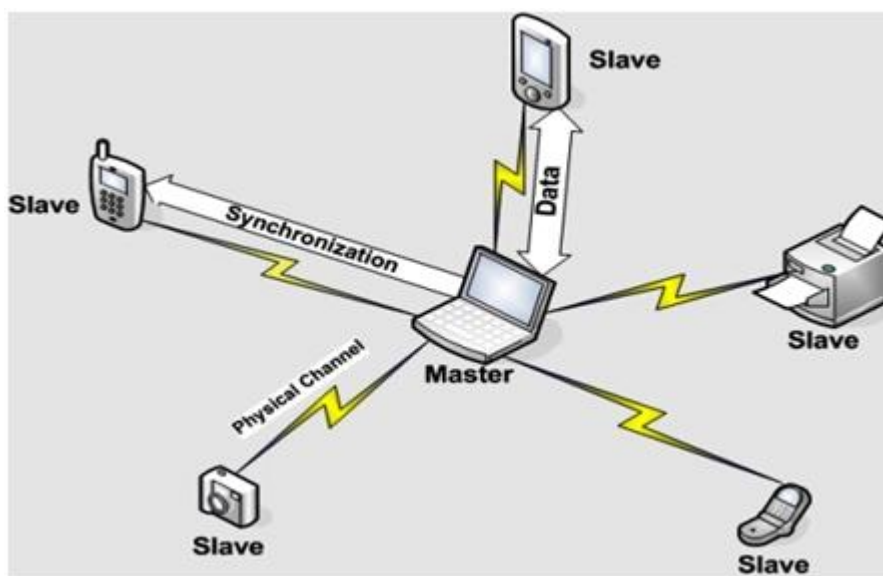
Obrázek 10 - Komunikace přes MQTT [35]

3.4.2 Bluetooth

Počátky tohoto protokolu sahají do roku 1994, kdy se hledala náhrada za RS-232. Označení nese po Vikingovi Haraldu Blätandovi (v překladu Harald Modrý zub), který dokázal sjednotit tehdejší rozdělená království. To lze připodobnit i k protokolu. Ten funguje na bázi radiových vln, kterými je schopný propojit dvě a více zařízení. V dnešní době je protokol široce využíván v elektronických zařízeních. Najdeme ho v mobilních telefonech, televizích, autech, audiozařízeních, počítačích, domácích spotřebičích nebo zařízeních v internetu věcí.

Frekvenční pásmo Bluetooth je 2,4 GHz při přenosové rychlosti dosahující 24 Mb/s. Obvyklý dosah protokolu je omezen na menší vzdálenosti okolo 10 m. Dosah je ovlivněn verzí

protokolu a překážkami mezi komunikujícími zařízeními. Zařízení využívající Bluetooth obsahují přijímač i vysílač signálu. Zařízení, které vyvolává a řídí síť je označováno jako master (jedno hlavní zařízení). Přes mastera probíhá komunikace ostatních zařízení označovaných jako slave (může jich být až sedm). Ta jsou hlavnímu master zařízení podřízená a řídí se jeho příkazy. Výhodou protokolu je jednoduchost, široké využití napříč zařízeními a systémy nebo oproti Wi-Fi nižší energetická náročnost. [36]



Obrázek 11 - Komunikace přes bluetooth [37]

3.4.3 ESP – NOW

Jedná se o rychlý bezdrátový komunikační protokol vyvinutý společností Espressif. Protokol, vycházející z podobných nízkoenergetických 2,4GHz protokolů, umožňuje rychlou bezdrátovou komunikaci mezi více ESP deskami bez nutnosti použití Wi-Fi. Při propojení zařízení protokolem jako první dochází k jejich párování. Jakmile je párování dokončeno, komunikace je považována za bezpečnou. Zařízení dále využívají peer-to-peer komunikaci bez nutnosti tzv. handshake. Spojení je tak stálé a pokud by došlo například k restartu zařízení, desky se rovnou automaticky spojí a budou pokračovat v přenosu informací. Přenos je zabezpečen kryptovacím CCMP mechanismem, který je založen na standartu AES. [38] [39]

ESP-NOW podporuje:

- Šifrovanou a nešifrovanou komunikaci.
- Přenos zpráv o velikosti až 250 B.
- Callback funkci (funkce zpětného zavolání) nastavitelnou na informování aplikační vrstvy o úspěchu či neúspěchu přenosu.

Limitace ESP-NOW:

- Limitovaný počet šifrovaných klientů. Ve Station módu je podporováno nejvýše deset šifrovaných klientů. V SoftAP nebo SoftAP Station módu je podporováno nejvýše šest šifrovaných klientů.
- Počet nešifrovaných klientů je v celkovém součtu s počtem šifrovaných klientů maximálně dvacet. [38]

ESP-NOW využívá pro dodavatele (vendor) specifický akční rámec. Přenosová rychlost protokolu je 1 Mbps. Akční rámec má následující architekturu:

MAC Header	Category Code	Organization Identifier	Random Values	Vendor Specific Content	FCS
24 bytes	1 byte	3 bytes	4 bytes	7-255 bytes	4 bytes

Tabulka 3 - Akční rámec ESP-NOW[41]

- Category Code (Kód kategorie): pole je v základu nastaveno na hodnotu 127 znamenající kategorii specifickou pro dodavatele (vendor)
- Organization Identifier (Identifikátor organizace): unikátní identifikátor (0x18fe34), tedy první tři jednotky (byte) MAC adresy určené výrobcem
- Random Values (Náhodné hodnoty): je určeno k zabezpečení komunikace a k ochraně před tzv. relay attacks (útok směřovaný mezi komunikaci dvou zařízení) [40]
- Vendor Specific Content (Obsah specifický pro dodavatele): obsah komunikace, který je definován níže: [41]

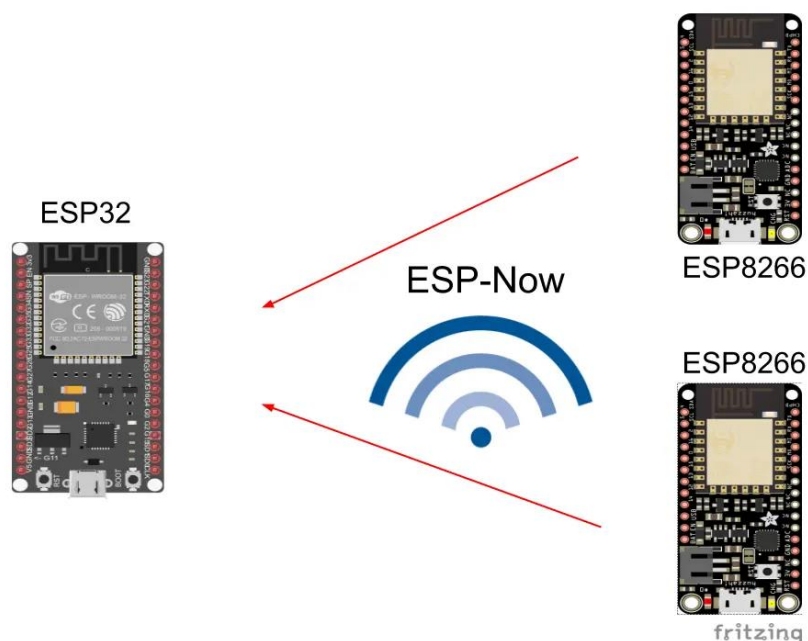
Element ID	Lenght	Organization Identifier	Type	Version	Body
1 byte	1 byte	3 bytes	1 byte	1 byte	0-250 bytes

Tabulka 4 - Vendor specific content ESP-NOW [41]

- Element ID (ID prvku): pole je nastaveno na hodnotu 221, označující prvek specifický pro dodavatele
- Length (Délka): obsahuje celkovou délku identifikátoru organizace, typu, verze a těla
- Organization Identifier (Identifikátor organizace): unikátní identifikátor (0x18fe34), tedy první tři jednotky (byte) MAC adresy určené výrobcem
- Type (Typ): pole je nastaveno na hodnotu 4 znamenající ESP-NOW
- Version (Verze): obsahuje verzi ESP-NOW
- Body (Tělo): pole reprezentující data zprávy ESP-NOW [41]

Možnosti komunikace mezi deskami ESP:

- Komunikace jedním směrem (
 - o jedna deska odesílá data do druhé (jeden přijímá data a druhá odesílá)
 - o jedna deska odesílá data do více zařízení (více zařízení je řízeno jedním hlavním)
 - o více ESP odesílá data do jedné desky (více zařízení je nařízenými jedné podřadné desky)
- Vzájemná komunikace oběma směry:
 - o Možnost vzájemné komunikace dvou a více ESP, kdy každé zařízení zároveň odesílá a přijímá data [42]



Obrázek 12 - Komunikace jedním směrem – více ESP odesílá data do jedné desky [42]

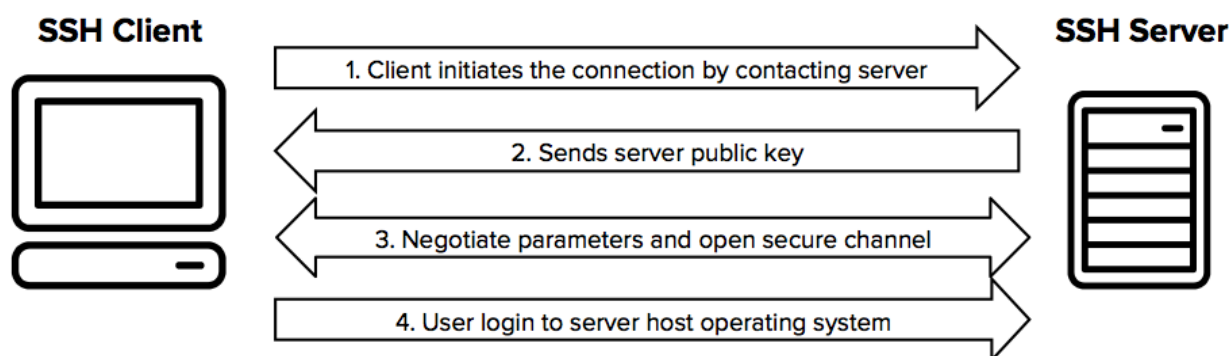
3.4.4 SSH

Secure Shell protokol je využíváný pro zabezpečený vzdálený přístup k počítači. Pomocí něj je možné vzdáleně ovládat zařízení s operačními systémy Linux, jako je na mikropočítačích Raspberry Pi. Protokol nabízí možnost silné autentifikace, zabezpečené komunikace a integrity pomocí šifrování. Protokol byl vytvořen jako náhrada starších nezabezpečených protokolů, např. Telnet, rLogin a protokolů pro přenos dat (např. FTP).

V praxi se SSH nejčastěji využívá:

- Pro správu síťové infrastruktury.
- K zajištění bezpečného přístupu uživatelů.
- K přenosům souborů (SFTP).
- K tunelování.

Protokol funguje jako model klient-server, tzn., že se SSH klient připojuje k SSH serveru. Klient při vytváření připojení používá k ověření identity SSH serveru jeho veřejný klíč. Veřejný klíč server zasílá na základě inicializace připojení ze strany klienta. Dalším krokem je domlouvání parametrů spojení. Na jeho základě se stanoví šifrovací algoritmus a vygeneruje se klíč použitý ke komunikaci mezi klientem a serverem. Nakonec dochází ke spojení a uživatel se tak může přihlásit k hostujícímu operačnímu systému na serveru. Mezi algoritmy využívané protokolem spadají např. šifrovací AES (Advanced Encryption Standard) poskytující bezpečnou komunikaci nebo hashovací SHA-2 (Standard Hashing Algorithm) zajišťující integritu dat. Jednotlivé kroky připojení jsou zachyceny na příkladu na obrázku č.13 [43] [44]



Obrázek 13 - Kroky připojení SSH [43]

3.4.5 VNC

Virtual network computing (virtuální síťové spojení) je jeden ze způsobů vzdáleného ovládání počítačů přes síťové připojení. VNC bylo vytvořeno v AT&T Laboratories, které zveřejnilo jeho zdrojový kód jako open source. Je to nezávislá platforma kompatibilní napříč operačními systémy. Umožňuje připojení více uživatelů. Stačí, aby zařízení, na kterých chceme VNC používat, využívala TCP/IP a měla otevřené porty pro možnost připojení jiných zařízení. [45]

Jedná se tedy o konkrétní softwarové řešení umožňující technickou podporu počítačů, serverů a dalších síťových zařízení. VNC využívající protokol RFB (Remote Frame Buffer) funguje na modelu klient/server. VNC klient je nainstalován na místním počítači a připojuje se na serverovou komponentu nainstalovanou na vzdáleném zařízení. Serverová komponenta následně přenáší v reálném čase obrazovku ovládaného zařízení na klienta. Klient, který je schopen vzdálené zařízení ovládat pomocí příkazů, např. klávesnicí nebo pohybem myši po obrazovce. [46] [47]

Jedním z nejvyužívanějších programů je VNC Viewer od RealVNC . Je kompatibilní s Windows, systémy na bázi Unixu, Mac OS nebo operačními systémy pro mobilní zařízení. [48]

3.4.6 Wi-Fi

WiFi je označení protokolu pro přenos dat založených na standardu IEEE 802.11. Vznik standardu se pohybuje okolo roku 1997. Standard je vydáván institucí IEEE, tj. zkratkou anglického Institute of Electrical and Electronics Engineers (v překladu Institut elektronického a elektrotechnického inženýrství). Tato instituce je neziskovou organizací starající se o technologický posun. WiFi standard je zaměřený na bezdrátové sítě a jejich využití pro komunikaci zařízení. Protokol se nejvíce využívá pro připojení zařízení do lokálních sítí nebo k připojení na internet, kde často dochází k přenosu objemných dat nebo streamování mediálního obsahu. Protokol je oproti Bluetooth více náročný na energii. Zároveň má však větší přenosovou rychlost a dosah signálu. Kvalita a dosah signálu jsou ovlivněny prostředím, ale i použitým hardwarem. Stejně jako Bluetooth je dnes WiFi zastoupena ve většině elektronických zařízení od mobilních telefonů, přes televize až po domácí spotřebiče. [36]

3.4.7 Ethernet

Ethernet je součástí síťových technologiích využívaných v lokálních sítích (LAN), metropolitních sítích (MAN) a rozlehlých sítích (WAN). Standardizován byl v roce 1983 institutem IEEE jako standart IEEE 802.3. Ethernet byl průběžně vylepšován. Postupem času získal podporu vyšších přenosových rychlostí a tím i dosah na větší vzdálenosti. Díky kompatibilitě nahradil ostatní technologie jako ARCNET, FDDI a Token Ring. [49]

V současné době Ethernet k propojení využívá kroucený dvoulinkový kabel nebo kabel z optických vláken. Díky neustálému technologickému progresu se přenosová rychlost dostala z počátečních 2,94 Mbit/s až k 400 Gbit/s (nazýváno Gigabitový ethernet). Po roce 2020 je dokonce očekávané dosažení rychlosti v jednotkách terabitů za sekundu, tj. označováno jako Terabitový ethernet. [50] [51]

Ethernet využívá referenční model OSI, konkrétně jeho fyzickou a linkovou vrstvu. Zařízení spojená přes ethernet rozdělují tok dat do rámců (z anglického označení frames). Jednotlivé rámce obsahují zdrojové a cílové adresy a kontrolní data pro detekci chyb. Tato kontrolní data umožňují detekovat a zahodit poškozené rámce. Vyšší vrstva se následně znovu pokouší o přenos chybných zahozených rámců. Pro internetové připojení je dnes ethernet široce využíván a jeho zastoupení tak lze najít napříč elektronickými zařízeními, které se k síti potřebují připojit. [52] [53]

3.5 Python

Python byl vytvořen v roce 1991 nizozemským programátorem Guido van Rossumem. Současným vývojářem je Python Software Foundation. Python je interpretovaný objektově orientovaný programovací jazyk s dynamickou sémantikou. Díky dynamické kontrole datových typů je využíván pro rychlý vývoj aplikací nebo pro propojování komponent. Jazyk je založen na jednoduché, snadno naučitelné syntaxi, čitelnosti kódu a na snížení nákladů na udržování programu. [54]

Jelikož neobsahuje nutnost kompilace, je rychlý při editaci, testování a opravě chyb. Pokud při ladění programu interpret nalezne chybu, je vyvolána výjimka. V případě, že není výjimka zachycena, interpret zobrazí trasování zásobníku. Debugger neboli ladicí program je obsažen přímo v Pythonu. Často využívaným a rychlým způsobem ladění je i vložení průběžných tisků na výstup programu. Tím lze interaktivněji zobrazovat průběh programu. [55]

Poslední vydanou verzí je k 18.12.2019 Python 3.8.1. V testování je od 25.1.2020 verze 3.9.0a3. Nejčastěji je jazyk využíván k tvorbě grafického rozhraní, aplikací týkajících se webového prostředí nebo umělé inteligence. Jazyk je open source a je součástí Linux systémů. Může tak být snadno přístupný a použitelný na Raspberry Pi. [56]

3.6 Node-RED

Node-RED je vizuální programovací nástroj založený na flow-based principu. Byl vyvinut společností IBM, která ho v roce 2016 zprostředkovala jako open source. Poskytuje grafické uživatelské rozhraní, ve kterém uživatel může intuitivně přemísťovat bloky. Bloky představují součásti většího systému, obvykle zařízení, softwarové platformy nebo webové služby, které mohou být propojeny. Mezi jednotlivé komponenty mohou být umístěny další bloky představující softwarové funkce, které zajišťují přenos nebo transformaci dat. Funkce mohou být psány v programovacím jazyku JavaScript. V nástroji je možné vytvářet i dashboardy (v překladu nástěnky). Dashboardy mohou zobrazovat grafy, přehledy nebo ovládací prvky systému, který je ovládán. [57] [58] [59]

4 Analýza požadavků vlastní práce

Cílem této práce je vytvořit funkční prototyp automatizovaného skleníku. Prototyp systému by měl být následně aplikovatelný na automatizaci reálného skleníku. Ke stanovení stěžejních bodů analýzy bylo přistoupeno k diskusi s potenciálními uživateli systému automatizace. Účastníky diskuze byli rodinní příbuzní. Tito lidé jsou dlouholetí pěstitelé ovoce a zeleniny. Vlastní pole o velikosti 1 Ha, jeho součástí je i zahradní skleník. Skleník a jeho okolí jsem osobně navštívil a zmapoval jeho možnosti. Po těchto proběhlých návštěvách a na základě diskuze byly účastníky stanoveny požadavky na systém automatizace skleníku následovně:

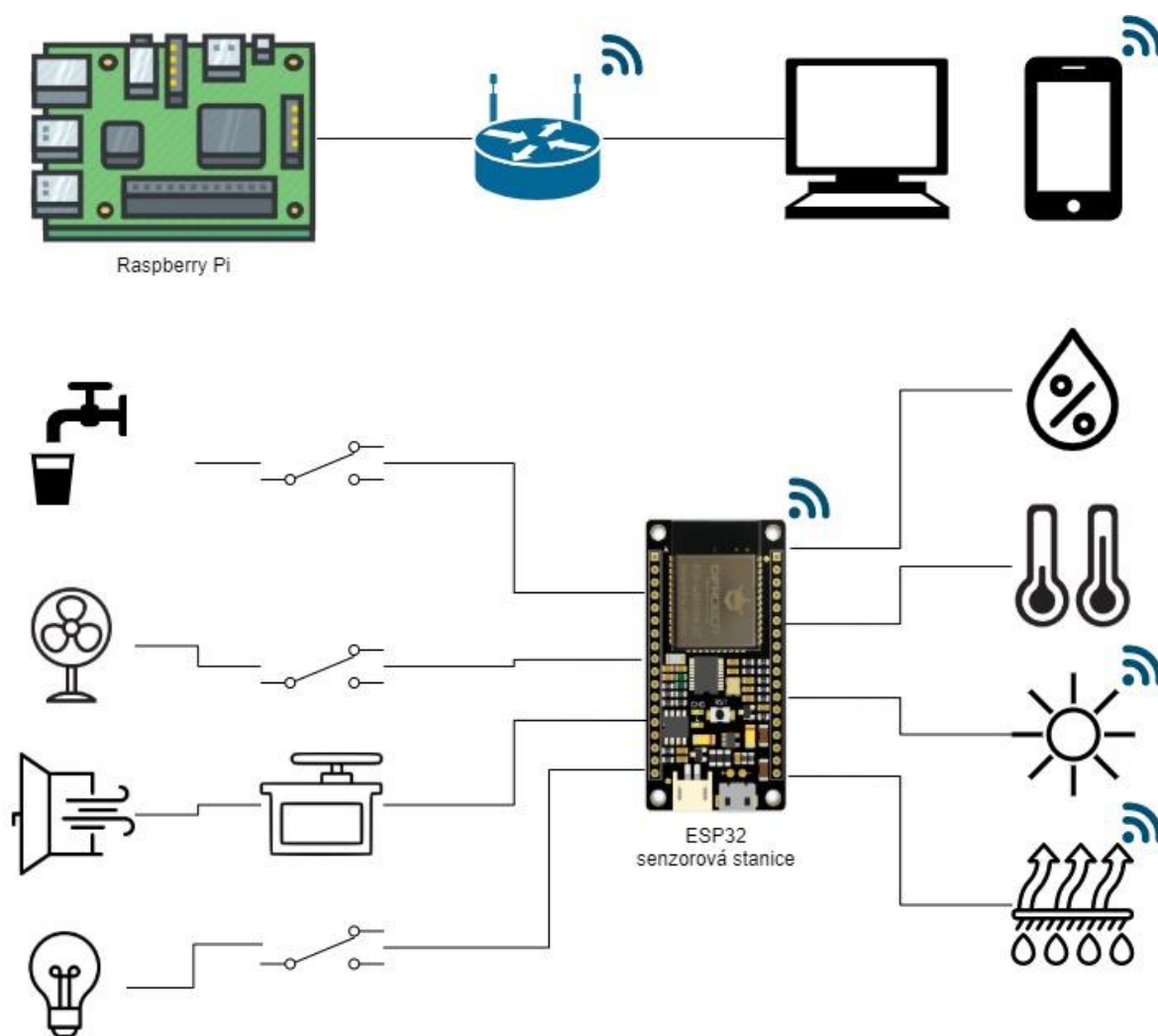
- Automatické zavlažování sadby
- Automatické osvětlení skleníku při nedostatku světla
- Automatické odvětrání skleníku pro snížení teploty
- Měření teploty ve skleníku
- Měření vlhkosti půdy
- Měření intenzity světla
- Měření půdní vodivosti/úrodnosti
- Ovládání systému z webového prostředí
- Grafická interpretace naměřených hodnot ve webovém rozhraní
- Základní zabezpečení přístupu k webovému prostředí

Jak vyplývá z výše uvedeného, požadavky se týkají spíše toho, jak by se měl systém chovat. Důležitým faktorem by dále měla být nižší finanční náročnost na realizaci. Hranice byla stanovena na 7 000 Kč. V diskusi bylo řečeno, že rodina pěstuje ve skleníku pouze v rozmezí od jara do podzimu. Skleník tak navyšuje svoji vnitřní teplotu pouze vlivem okolního počasí a není tedy požadavek na zařazení vytápění do systému. Pro regulaci teploty bylo stanoveno jako dostačující zautomatizování procesu odvětrávání. To doteď probíhalo otevřením oken skleníku. Jelikož je skleník umístěn od fyzické přípojky k internetu ve vzdálenosti přesahující deset metrů, měl by být kladen důraz na využití bezdrátových technologií a protokolů.

5 Návrh vlastního řešení

5.1 Architektura systému

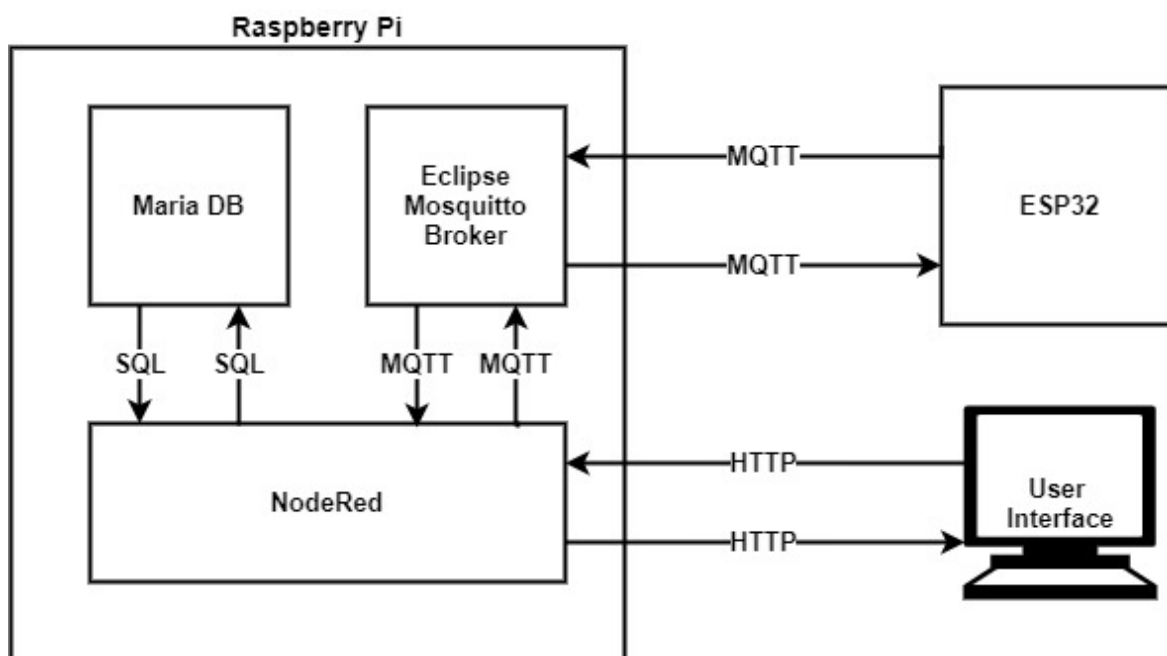
Centrální jednotkou systému automatizovaného skleníku byl zvolen mikropočítač Raspberry Pi 2. Tato řídicí jednotka zajišťuje komunikaci v systému. Součástí systému je také mikrokontroler s čipem ESP32. Mikrokontroler je určen ke sběru naměřených hodnot ve skleníku a jejich distribuci do systému. Zároveň na základě naměřených hodnot a zvolených preferencí obsluhuje k němu připojené periferie. Periferie byly zvoleny tak, aby dokázaly zajistit základní automatickou funkčnost skleníku. Ke komunikaci mezi ESP32 a Raspberry Pi je využíván protokol MQTT. Na následujícím obrázku je vyznačena základní komunikace v navrhovaném systému automatizace.



Obrázek 14 - Architektura systému automatizace

Stěžejními funkcemi navrhovaného systému je měření vlhkosti půdy, intenzity světla, teploty a vlhkosti vzduchu. Na základě naměřených dat bude systém schopen obstarat zavlažování, dodatečné osvětlení a regulaci vnitřní teploty skleníku.

Způsob komunikace mezi jednotlivými prvky v systému je zachycen na obrázku níže. Jak ze schématu vyplývá, senzorová stanice ESP32 bude s centrální jednotkou systému komunikovat výhradně přes protokol MQTT. Zprostředkovatelem komunikace skrze MQTT bude Eclipse Mosquitto Broker. Uživatelské rozhraní bude využívat protokol http. Pro ukládání dat bude použit databázový systém MariaDB. Mozkem zprostředkujícím zpracování dat a chování systému bude program Node-RED.



Obrázek 15 - Způsob komunikace mezi prvky systému

5.2 Měřicí a ovládací stanice ESP32

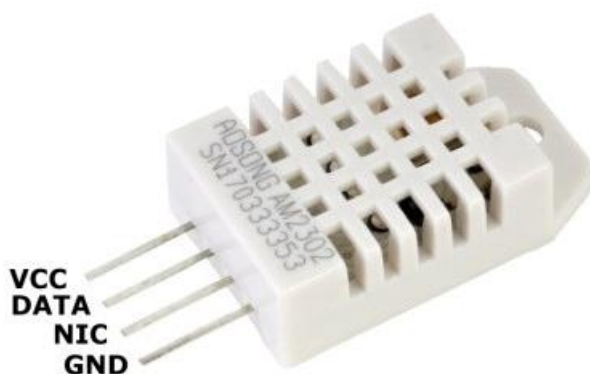
Jako senzorová stanice bude zvolen mikrokontroler ESP32. Mikrokontroler podporuje bezdrátovou komunikaci přes protokoly WiFi a Bluetooth, tudíž bude vhodný pro samostatné umístění do skleníku. ESP32 je méně energeticky náročné, přičemž je schopno ovládat periferie potřebné k automatizaci. Programy pro tuto platformu budou realizovány ve vývojovém prostředí Arduino IDE a budou využívat k tomu určené knihovny. Periferie připojené k mikrokontroleru budou rozebrány v následující části.

5.3 Senzory měřící stanice

Podle výstupu z analýzy požadavků na automatizaci skleníku byly určeny senzory potřebné k měření požadovaných veličin. Senzory měřící stanice byly rozděleny na dvě části.

5.3.1 Senzor teploty a vlhkosti vzduchu

Pro naměření vnitřní teploty a vlhkosti vzduchu ve skleníku byl zvolen senzor DHT22. Jeho předností je nízká cena a oproti verzi DHT11 také velká přesnost v měření hodnot. Další z výhod je programová podpora v podobě snadno dostupné knihovny. Operační napětí je 3 V až 5 V a proud 2,5 mA. Senzor je schopen získat data každé dvě sekundy. Na tuto skutečnost se musí brát zřetel při programování. Získání údajů o teplotě je prováděno pomocí termistoru. Teplotu je možné naměřit v rozmezí od $-40\text{ }^{\circ}\text{C}$ do $+125\text{ }^{\circ}\text{C}$ s přesností $\pm 0,5\text{ }^{\circ}\text{C}$. Získávání dat o vlhkosti obstarává součástka citlivá na vlhkost vzduchu. Vlhkost je měřena v rozmezí od 0% do 100% s přesností 2% až 5%. DHT22 má rozměry 15,1mm x 25mm x 7,7mm. Výstupem senzoru jsou čtyři piny zachycené na obrázku níže. Pin VCC pro připojení napájení. Druhý pin DATA je určený k přenosu dat. Třetí se nepoužívá a poslední pin GND je určen pro připojení k zemi.



Obrázek 16 - Senzor DHT22 [60]

5.3.2 Senzor rostlin

K měření základních hodnot u rostlin ve skleníku bylo zvoleno zařízení Xiaomi Mi Flora (lze nalézt i pod názvem Xiaomi Mi Plant). Prvek od Xiaomi zvládne naměřit teplotu vzduchu,

vlhkost půdy, intenzitu světla a půdní vodivost nebo úrodnost (z anglického conductivity/fertility). Díky energetické nenáročnosti zvládne zařízení vydržet až rok na jednu 3 V lithiovou baterii CR2032. Přenos naměřených dat ze zařízení je prováděn bezdrátově přes Bluetooth. Těchto zařízení je zároveň možné použít více. Je tak možné rozdělit skleník na sekce a měřit každou samostatně, tj. vhodné pro pěstování pro pěstování různých typů rostlin.

5.4 Ovládací prvky

Komunikace periférií pro závlahu, osvětlení a regulaci teploty s mikrokontrolerem ESP32 bude zajištěna fyzickým propojením pomocí kabeláže.

5.4.1 Osvětlení

Pro podpůrné pěstební osvětlení byl zvolen LED pásek s výkonem 12 W/m. Poměr světelného spektra je R4:B1, tedy 4 červené zdroje světla na 1 modrý. Poměr světelného spektra byl zvolen na základě požadavků majitelů skleníku na osvětlení. LED pásek vyžaduje napětí 12 V. Energetická náročnost na napájecí zdroj je ovlivněna délkou LED pásku. Osvětlení bude ovládáno sepnutím obvodu v požadovanou dobu s návazností na naměřené hodnoty. Sepnutí obvodu bude mikrokontrolerem ovládáno pomocí relé modulu.

5.4.2 Zavlažování

Zavlažování půdy ve skleníku bude na základě požadavků prováděno čerpadlem. K čerpadlu bude napojena hadice rozvádějící závlahu po skleníku k jednotlivým rostlinám. Pro účely funkčního prototypu a testování systému postačí menší malé čerpadlo s nízkým výkonem. Na prototyp tak bude aplikováno čerpadlo pracujícím na 12 V stejnosměrného napětí o výkonu 5 W. Aby byl systém závlahy použitelný na reálném skleníku, čerpadlo bude ovládáno spínáním obvodu pomocí relé modulu. Jediný rozdíl v použití jiného čerpadla by tak byla zdrojová náročnost.

5.4.3 Odvětrání

Regulace – snižování teploty uvnitř skleníku bude prováděna dvěma způsoby. Jelikož byla teplota dosud snižována otevřením oken, mechanismus zůstane zachován a zautomatizován použitím servomotoru. Servomotor na prototypu bude menší s nižším

výkonem. Pro model prototypu byl zvolen servomotor pracující na napětí 5 V, s utahovacím momentem 13 kg/cm. Servomotor bude připojen k potřebnému napájení. Ovládání pohybu jednotky bude zajišťovat mikrokontroler ESP32 s využitím knihovny Servo.h, která je k dispozici v Arduino IDE.

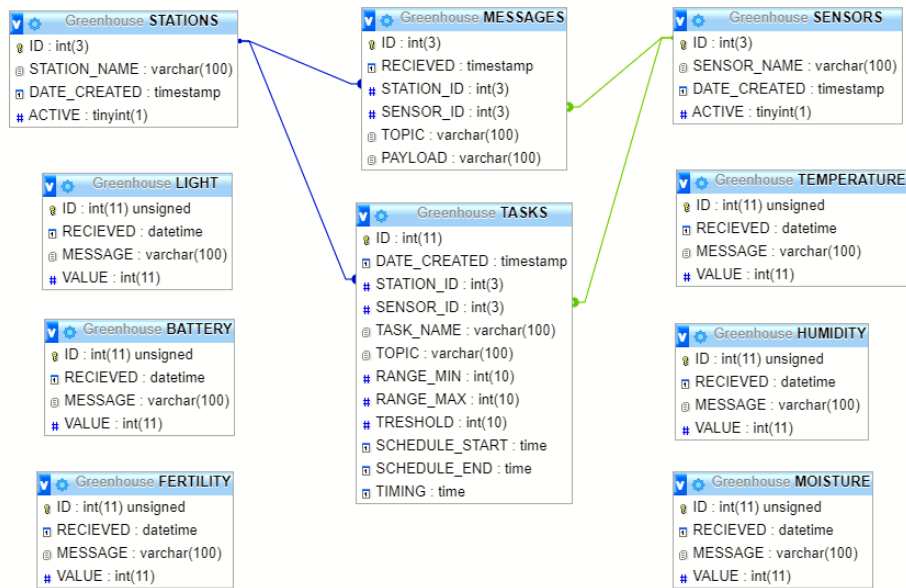
Jako druhý způsob regulace teploty uvnitř skleníku byl navrhnout systém využívající elektrického větráku pro rozpořbování vzduchu. Větrák využívá napětí 230 V a je možný ho napájet z běžné zásuvky. Pro jeho zapnutí/vypnutí bude využito relé modulu ovládaného mikrokontrolerem ESP32.

5.4.4 Spínací prvek

Pro účely spínání obvodů zavlažování, osvětlení a odvětrání bude použit relé modul. Relé modul byl vybrán se čtyřmi kanály, tzn., že obsahuje čtyři samostatná relé. Osazená relé Tongling JQC-3FF-S-Z jsou schopna spínat obvody střídavého proudu až 250V/10A a stejnosměrného proudu až 30V/10A. Modul je opticky oddělen a při případném přepětí nedojde k poškození připojeného mikrokontroleru. Každé relé je spínáno jednotlivě v závislosti na signálu, který vyšle řídicí deska ESP32. Výhodou modulu je možnost výměny prvků, které jsou potřeba sepnout bez nutnosti úpravy řídicího programu v ESP32.

5.5 Databáze

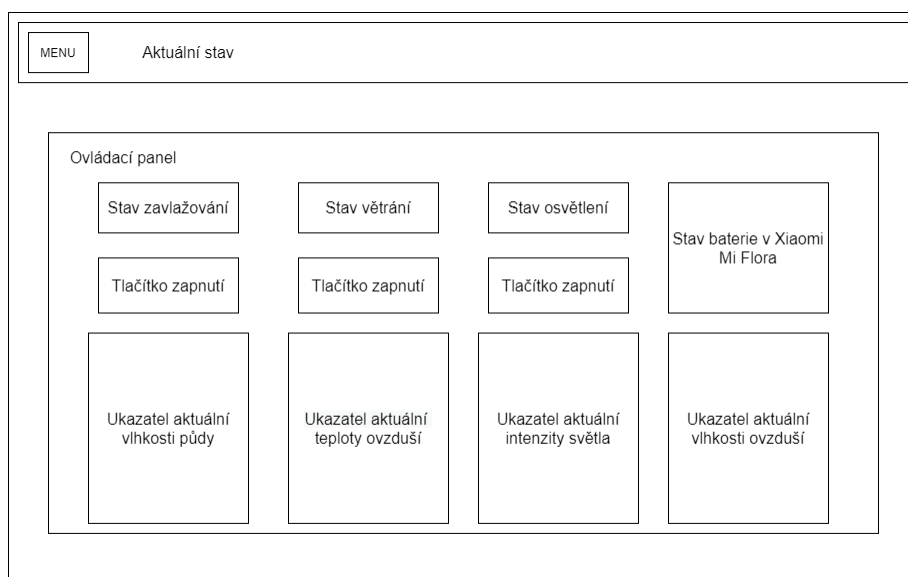
K dlouhodobému uložení dat byla zvolena relační databáze. S databází bude komunikovat pouze centrální systém Node-RED, který bude zodpovědný za manipulaci s daty. Aby nedošlo k nekonzistenci dat, uživatelské rozhraní prezentující data z databáze bude pouze přejímat data skrze systém Node-RED. Centrální systém bude data z databáze vyhodnocovat a na jejich základě se rozhodovat.



Obrázek 17 - Databáze systému

5.6 Uživatelské rozhraní

Uživatelské rozhraní je důležitou součástí celého systému automatizovaného skleníku. K rozhraní bude uživatel přistupovat pomocí webového prohlížeče po zadání uživatelského jména a hesla. Rozhraní bude poskytovat grafické znázornění posledních naměřených hodnot společně s přehledem jejich vývoje v delším časovém horizontu. Zároveň bude možné zobrazit přehled přijatých MQTT zpráv podle zadaného filtru. Dále bude součástí rozhraní možnost ovládat senzory a stanice a nastavovat příslušné parametry pro spouštění jednotlivých úloh.



Obrázek 18 - Wireframe domovské obrazovky uživatelského rozhraní

6 Implementace

6.1 Softwarové řešení Raspberry Pi

V následující kapitole byla popsána instalace programového vybavení Raspberry Pi a všech nutných kroků pro provozu schopnost a automatizaci skleníku.

6.1.1 Instalace

Operační systém (OS) je uložen na paměťové microSD kartě. Paměťová karta musela být zformátována a nastavena jako souborový systém typu FAT32. Dále došlo ke stažení nejaktuálnějšího obrazu operačního systému Raspbian Buster z oficiálních stránek vývojáře. Obraz operačního systému byl následně pomocí instalačního programu Raspberry Pi imager nainstalován na paměťovou kartu. Nakonec byla karta vložena do samotného mikropočítače. Po jeho spuštění došlo k načtení operačního systému a byla tak možná další práce se zařízením.

6.1.2 Konfigurace

Mikropočítač pro své první spuštění a konfiguraci potřeboval být připojený k napájení, monitoru, klávesnici s myší a internetu. Napájení zařízení bylo připojeno přes microUSB konektor a napájecí adaptér do elektrické sítě. Do HDMI konektoru byl připojen monitor. Klávesnice s myší využila standardní USB port. Připojení k internetu bylo realizováno pomocí ethernetového kabelu s koncovkou RJ45.

Po prvním spuštění a načtení OS bylo nutné nastavit lokalizaci, aktuální čas a časové pásmo. Pro zajištění větší bezpečnosti zařízení bylo změněno přístupové heslo u administrátorského účtu. Výrobce totiž dává v základním nastavení OS administrátorská práva účtu pi společně s všeobecným heslem raspberry. Dalším důležitým krokem bylo povolení vzdáleného přístupu přes protokoly SSH a VNC. Aby se zamezilo změně IP adresy mikropočítače v síti, na routeru byla zařízení přiřazena statická IP adresa. Po úspěšné konfiguraci bylo k zařízení přistupováno výhradně vzdáleně přes SSH nebo VNC.

6.1.3 Vzdálený přístup – Putty a VNC

Oba programy Putty i VNC jsou určeny pro vzdálený přístup k mikropočítači. Putty je příkazovým terminálem komunikujícím se vzdáleným zařízením přes SSH.

Pro vzdálené ovládání včetně grafického rozhraní je určen VNC. Konkrétně bylo využito řešení od společnosti RealVNC. K využití RealVNC bylo přistoupeno z důvodu instalace softwaru Npm a Node.js na Raspberry Pi. Software nebylo možné nainstalovat skrze Putty, respektive SSH. Úspěch zaznamenalo až využití RealVNC.

6.1.4 MQTT broker (Eclipse Mosquitto)

Pro komunikaci periférií a sdílení dat mezi zařízeními je nutné mít poskytovatele MQTT. Těch je na trhu dostupných velmi mnoho. V tomto případě byl na Raspberry Pi zvolen a nainstalován broker Eclipse Mosquitto ve verzi 1.5.7.

Instalace byla provedena přes příkazový řádek, pomocí příkazu `sudo apt install -y mosquitto mosquitto-clients`. Příkaz nainstaloval broker včetně klientů. Pro automatické spuštění brokeru po zapnutí mikropočítače byl následně proveden příkaz `sudo systemctl enable mosquitto.service`. Automatické spuštění brokeru je nutné zajistit pro chod systému skleníku. Pokud by došlo k restartu Raspberry Pi a nebyl by spuštěný MQTT broker, nemohlo by dojít ke komunikaci s ESP32, systém by tak neměl aktuální naměřené hodnoty a nešel by ovládat.

Otestování funkčnosti bylo provedeno zjištěním verze brokeru, které provedl příkaz `mosquitto -v`. Error na posledním řádku je brán jako potvrzení, že služba již běží a je správně nainstalována. Důležitou informací je dále port, na kterém je služba přístupná. Číslo portu odhalil předposlední řádek informací o brokeru. MQTT služba je tak dostupná na adrese 192.168.88.243:1883, kdy IP adresa je dána adresou Raspberry Pi a již zmíněným portem MQTT.

6.2 MySQL databáze

K ukládání dat pro dlouhodobé účely byla použita relační databáze MySQL ve verzi 10.3.22- MariaDB. K databázi lze přistupovat přes doinstalované webové prostředí phpMyAdmin. Pro přístup k datům byl vedle standartního admin účtu vytvořen uživatelský účet `nodered`, který má práva pouze na výběr, vkládání a aktualizaci příslušných tabulek. Účet

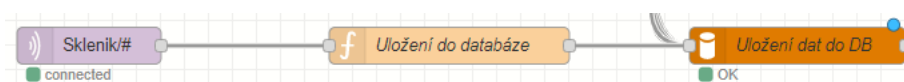
nodered byl následně použit v centrální aplikaci (pracujícím v systému Node-RED), která vyhodnocuje data ze skleníku. K zaznamenání časových údajů při vkládání nebo manipulaci s daty jsou určeny automatické spouště pracující nad jednotlivými tabulkami.

6.3 Node-RED

Důležitou součástí systému automatizace je vývojový nástroj Node-RED. Nástroj umožňuje efektivní komunikaci se zařízeními přes MQTT a je tak vhodný k využití na řešené automatizaci. V nástroji byla vytvořena jednotlivá flow, starající se o fungování celého systému skleníku. Každá flow je složená z uzlů vykonávajících příkazy dle aktuální situace a naměřených dat. V rámci vytváření systému byly naprogramovány procesní funkce potřebné pro chod systému jako takového.

6.3.1 Flow uložení MQTT zpráv do databáze

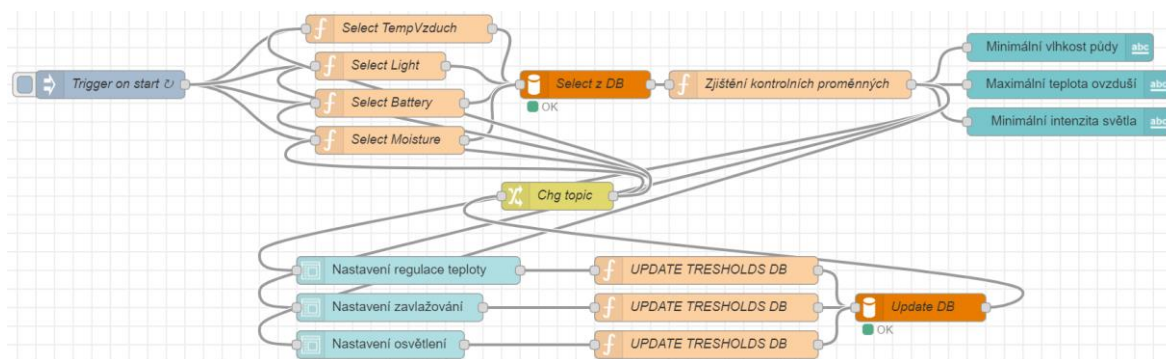
Uložení přijatých zpráv je rozděleno na ukládání přijatých zpráv jednotlivě po tématech a na zpracování veškeré MQTT komunikace. Přijaté naměřené hodnoty ze senzorů jsou ukládány do příslušných tabulek v databázi. Jako ukázka bude popsán proces uložení proběhlé MQTT komunikace. K zachycení všech MQTT zpráv je určena tabulka MESSAGES. Do tabulky jsou vkládány informace o datu a času, tématu a obsahu proběhlé zprávy a nakonec identifikátor stanice a senzoru, se kterými komunikace proběhla. Systém je připraven zaznamenávat komunikaci se senzory a stanicemi, které by mohly do budoucna případně systém automatizace rozšířit. Jelikož zatím systém obsahuje pouze jednu stanici, je do pole zaznamenána jeho výchozí hodnota ID. Ke komunikaci s databází slouží uzel nakonfigurovaný na připojení k MySQL databázi. Nejprve je však potřebné přijmout na začátku procesu MQTT zprávu. Ze zprávy jsou následně v naprogramované funkci extrahovány potřebné informace a je sestaven SQL příkaz pro vložení dat do tabulky. Příkaz je poté předán ke zpracování do zmiňovaného uzlu pro komunikaci s databází.



Obrázek 19 - Flow uložení komunikace MQTT

6.3.2 Flow pro nastavení proměnných systému

V systému Node-RED je možné pracovat s context, flow a global proměnnými. Context je možné využít pouze v daném uzlu. Proměnnou flow je možné využívat pro celou flow. Global lze využít napříč uzly i procesy v Node-RED. Pro nastavení mezních proměnných pro procesy zavlažování, větrání a osvětlení bylo nakonfigurováno následující flow.



Obrázek 20 - Flow nastavení proměnných systému

Vše začíná výběrem jednotlivých omezujících proměnných z databáze. Následně jsou pomocí funkce definovány pro celou flow mezní proměnné.

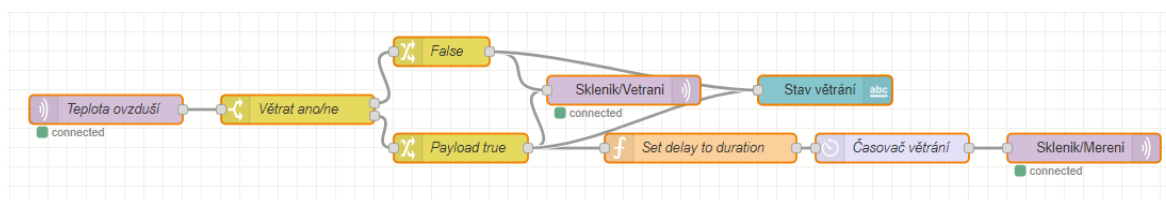
```
if (msg.payload[0].TOPIC=="Sklenik/Svetlo"){
  if (typeof ligmin=="undefined"){
    ligmin=500;}
  else if(ligmax=="undefined"){
    ligmax=40000;}
  else if(ligtres=="undefined"){
    ligtres=1000;}
  else if(ligst=="undefined"){
    ligst='17:00';}
  else if(ligend=="undefined"){
    ligend='20:00';}
  if(ligmin!=msg.payload[0].RANGE_MIN){
    ligmin=msg.payload[0].RANGE_MIN;}
  else if(ligmax != msg.payload[0].RANGE_MAX){
    ligmax = msg.payload[0].RANGE_MAX;}
  else if(ligtres != msg.payload[0].TRESHOLD){
    ligtres = msg.payload[0].TRESHOLD;}
  else if(ligst != msg.payload[0].SCHEDULE_START){
    ligst = msg.payload[0].SCHEDULE_START;}
  else if(ligend != msg.payload[0].SCHEDULE_END){
    ligend = msg.payload[0].SCHEDULE_END;}
  else if(ligschst != msg.payload[0].SCH_ST){
    ligschst = msg.payload[0].SCH_ST;}
  else if(ligschend != msg.payload[0].SCH_END){
    ligschend = msg.payload[0].SCH_END;}
  flow.set('ligmin',ligmin)
  flow.set('ligmax',ligmax)
  flow.set('ligtres',ligtres)
  flow.set('ligst',ligst*1000)
```

Obrázek 21 - Zdrojový kód – ukázka funkce nastavení systémových proměnných

Proměnné jsou po nastavení dále posílány do uživatelského rozhraní, konkrétně do menu Nastavení. Pokud uživatel v rozhraní změní a potvrdí mezní hodnotu některé z proměnných, data jsou předána dále do funkce, upravena do potřebné podoby a aktualizována v databázi. Následně ve flow proběhne přenastavení proměnných, které je dále promítnuto do uživatelského rozhraní.

6.3.3 Flow zpracování MQTT zpráv

Pro každé téma bylo v systému vytvořeno samostatné flow pro zpracování přijaté zprávy z ovládací stanice ESP32. Po přijetí zprávy na dané téma je nejprve ověřeno, zda naměřená hodnota není pod nebo nad mezí, případně jestli odpovídá časovému rozmezí. Mez je stanovena nastavením uživatele v uživatelském rozhraní. Po rozhodnutí, zda je nutné regulovat související veličinu, je zpět na stanici ESP32 s periferiemi odeslána zpráva určující, jak se má zachovat a co případně vykonat. Součástí procesu je i aktualizace zpráv na úvodní straně webového rozhraní, která uživateli poskytuje informaci, v jakém stavu se nachází periferie regulující související měřenou hodnotu. Na obrázku níže je znázorněna flow pro zpracování naměřené teploty vzduchu ve skleníku.



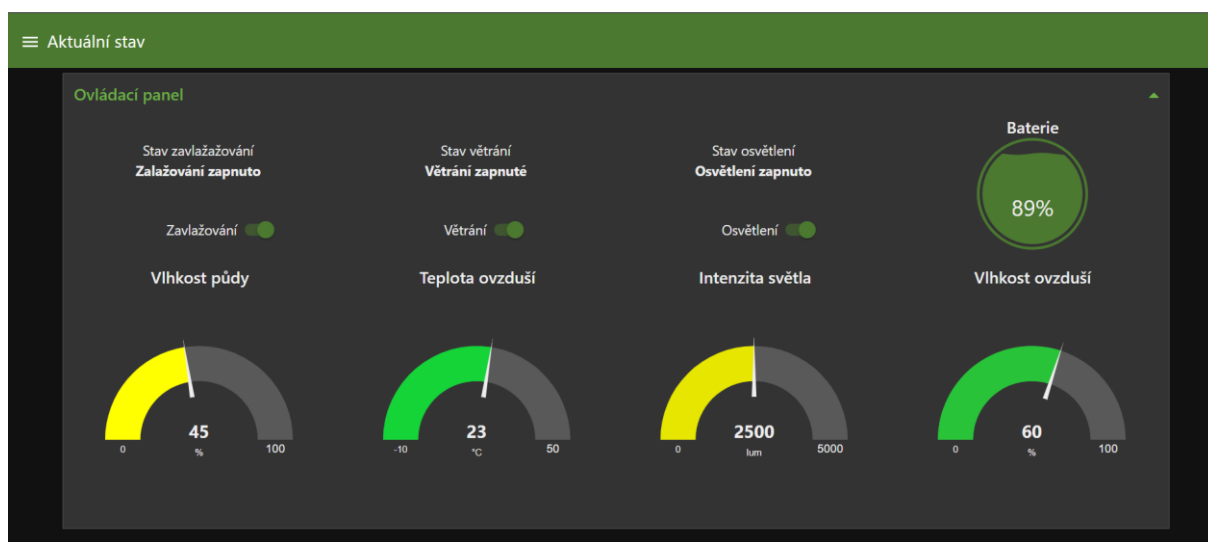
Obrázek 22 - Flow pro zpracování MQTT zpráv na téma teploty ovzduší

6.4 Webové rozhraní

Jednou z velkých výhod nástroje Node-RED je možnost tvorby webového rozhraní přímo z jeho vývojového prostředí. Softwarové řešení webového rozhraní pomocí nástroje Node-RED tak poskytuje přehlednou konfiguraci a manipulaci s daty v jednom nástroji a prostředí, společně s aplikací ke zpracování a vyhodnocení dat. Zároveň není centrální jednotka Raspberry Pi dále zatěžována dodatečným webovým serverem a systémem pro standardní webové rozhraní. Pro tvorbu a konfiguraci uživatelského rozhraní bylo zapotřebí doinstalovat balík node-red-dashboard. Ten obsahuje jednotlivé uzly (nodes), které po nastavení a konfiguraci vytváří grafické rozhraní. Data zobrazovaná v uživatelském rozhraní jsou dodávána z dat uložených v databázovém systému. Uživatel tak má vždy přístup k aktuálním i

historickým datům. Celé prostředí je přístupné z jakéhokoliv zařízení připojenému k lokální síti na 192.168.88.243:1880/UI. Přístup z ostatních sítí lze řešit buď přístupem přes VPN nebo přesměrováním portu.

Rozhraní je rozděleno do více sekcí. Zároveň bylo rozhraní zabezpečeno a povoluje přístup pouze oprávněným uživatelům. První je domovská obrazovka, na níž je vidět přehled aktuálně naměřených veličin ze skleníku a aktuálního stavu periferií starajících se o chod skleníku.



Obrázek 23 - Uživatelské rozhraní – úvodní stránka

Jednotlivé měřené veličiny mají každá vlastní stránku, na které je možné ručně zapnout/vypnout související periferii, zobrazit přehled naměřených dat a nahlédnout na historický vývoj těchto veličin. K zobrazení historického vývoje naměřených hodnot byly využity spojnicové grafy. Uzel generující tyto grafy však potřeboval data vybíraná z databáze připravit do pole. Bylo tedy nutné vytvořit funkci, která data do formátu připraví. Ukázka funkce je na obrázku níže.

```

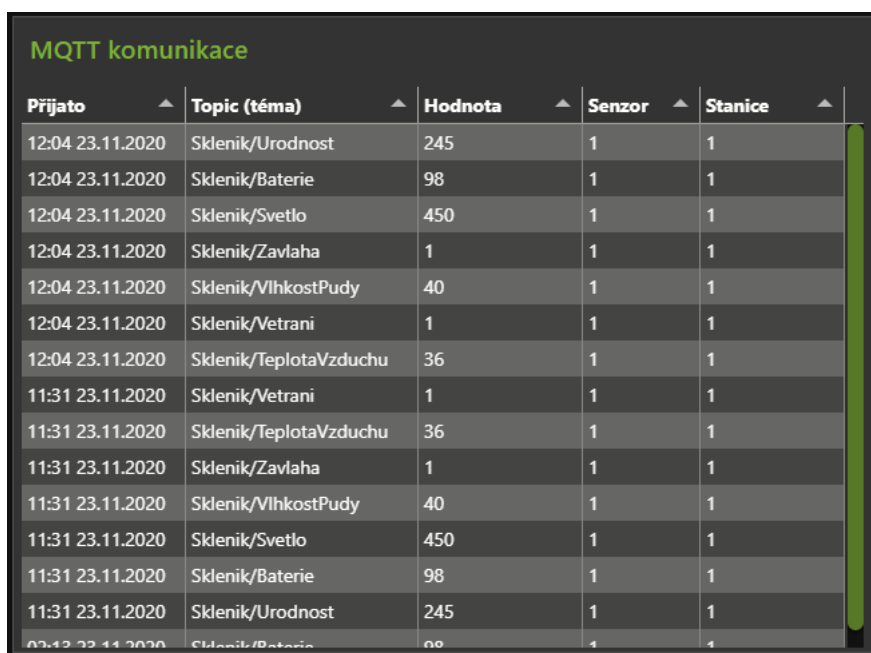
var data = [];
var time = [];
msg.payload.forEach(function(value) {
  data.push(value['Temperature']);
  time.push(value['TimeStamp']);
});

msg.payload = [{
  series: "Teplota",
  data: [data],
  labels: time
}];
return msg;

```

Obrázek 24 - Funkce pro zformátování dat

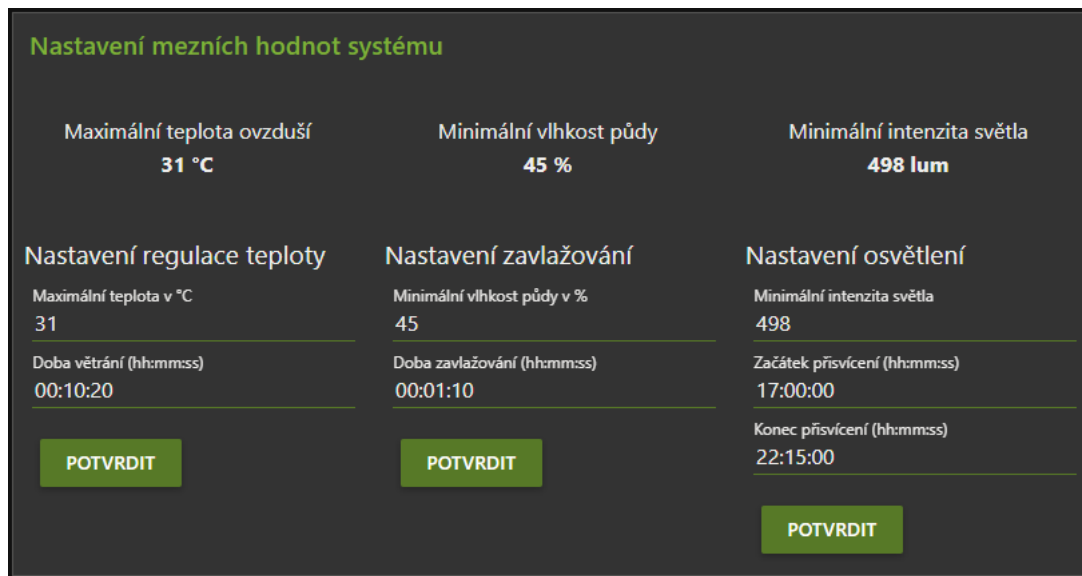
Další sekci je zobrazení zaznamenaných MQTT zpráv, které byly komunikovány mezi centrální aplikací pro zpracování dat a ovládací stanicí ESP32. Zobrazovány jsou od nejnovějších po nejstarší.



Přijato	Topic (téma)	Hodnota	Senzor	Stanice
12:04 23.11.2020	Sklenik/Urodnost	245	1	1
12:04 23.11.2020	Sklenik/Baterie	98	1	1
12:04 23.11.2020	Sklenik/Svetlo	450	1	1
12:04 23.11.2020	Sklenik/Zavlaha	1	1	1
12:04 23.11.2020	Sklenik/VlhkostPudy	40	1	1
12:04 23.11.2020	Sklenik/Vetrani	1	1	1
12:04 23.11.2020	Sklenik/TeplotaVzduchu	36	1	1
11:31 23.11.2020	Sklenik/Vetrani	1	1	1
11:31 23.11.2020	Sklenik/TeplotaVzduchu	36	1	1
11:31 23.11.2020	Sklenik/Zavlaha	1	1	1
11:31 23.11.2020	Sklenik/VlhkostPudy	40	1	1
11:31 23.11.2020	Sklenik/Svetlo	450	1	1
11:31 23.11.2020	Sklenik/Baterie	98	1	1
11:31 23.11.2020	Sklenik/Urodnost	245	1	1

Obrázek 25 - Uživatelské rozhraní – výpis proběhlé komunikace přes MQTT

Významnou součástí uživatelského rozhraní je možnost nastavení mezních hodnot systému, podle kterých je následně vyhodnocováno zda provádět obhospodařující akce skleníku či nikoli. Uživatel má možnost je nastavit v sekci nastavení.



Obrázek 26 - Uživatelské rozhraní - nastavení mezních hodnot systému

6.5 ESP32

Ovládací prvky systému jsou řízeny programem nahráním na mikrokontroler ESP32. Mikrokontroler se v systému zodpovídá centrální aplikaci a reaguje na její příkazy. Veškerá komunikace mezi centrální aplikací a mikrokontrolerem je zajištěna pomocí MQTT zpráv. Program ESP32 běží v nekonečné smyčce, přičemž jednotlivé akce vykonává v reakci na zaslané příkazy. Po nahrání programu a spuštění zařízení nejprve probíhá nastavení. V nastavení probíhá konfigurace připojení k internetu pomocí WiFi.

```
void Wifi_connect() {
    delay(10);
    Serial.print("Připojuji se k WIFI síti ");
    Serial.println(WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("ESP připojeno k WiFi \n");
    Serial.println("IP adresa ESP: ");
    Serial.println(WiFi.localIP());
}

void Wifi_disconnect() {
    WiFi.disconnect(true);
    Serial.println("WiFi odpojeno");
}
```

Obrázek 27 - Zdrojový kód - připojení ESP32 k WiFi

Po úspěšném připojení dochází ke spojení s MQTT brokerem a následně k nastavení callback funkce. Na závěr dojde k přihlášení odběru zpráv na definovaná témata. Callback funkce je stěžejní pro správnou funkčnost systému. Jedná se o funkci, kterou můžeme zavolat zasláním MQTT zprávy na určité téma. Ona následně vykoná určené úkony. Zde je přihlášena k tématům spouštějících nebo ukončujících závlahu, osvětlení a větrání a poslouchající tak pokyny z řídicí aplikace v Node-RED. Na závěr nastavovacího procesu mikrokontroleru dochází k přiřazení pinů k jednotlivým periferiím. Po skončení nastavení přejde program do smyčky, ve které jsou opakovaně prováděny určené funkce.

```
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Přijata zpráva na téma: ");
  Serial.print(topic);
  Serial.print(". Zpráva: ");
  String mTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    mTemp += (char)payload[i];
  }
  Serial.println();

  //nyní přijde na řadu provedení jednotlivých funkcí na základě přijatého tématu
  if (topic == "Sklenik/Vetrani"){
    Serial.print("Změna stavu okna: ");
    if (mTemp == "true") {
      servo.write(windowOpen);
      Serial.print("Otevřeno");
      digitalWrite(releVetrak,HIGH);
      Serial.print("Větrák zapnutý");
      delay(10000);
      digitalWrite(releVetrak,LOW);
      Serial.print("Větrák vypnutý");
    }
    else if (mTemp == "false") {
      servo.write(windowClose);
      Serial.print("Zavřeno");
      digitalWrite(releVetrak,LOW);
      Serial.print("Větrák vypnutý");
    }
  }
}
```

Obrázek 28 - Zdrojový kód – ukázka callback funkce

6.6 DHT22 – senzor

K získání dat ze senzoru DHT22 je využita dostupná knihovna DHT.h, která zajišťuje komunikaci s ESP32. Vzhledem k více typům senzorů DHT je v počátku potřeba definovat

použitý typ. Dále je určen pin, přes který probíhá komunikace. Tyto dvě proměnné jsou pak předány dále do proměnné dht, která volá příkazy z knihovny. Ke čtení teploty je použitý příkaz dht.readTemperature() a pro vlhkost ovzduší je to dht.readHumidity(). Knihovna je schopna vypočítat teplotní index, který je vypočten kombinací naměřené teploty a vlhkosti. V systému ovšem nebyl index využit. DHT22 je schopno naměřit data za zhruba 250 milisekund. Je však potřeba brát v potaz, že senzor je schopen tato měření opakovat nejrychleji každé 2 sekundy. Nakonec jsou data vypsaná do sériového portu a odeslána přes MQTT do centrálního systému Node-RED k vyhodnocení.

```
void ziskejDhtData() {
  //Přečte vlhkost
  humair = dht.readHumidity();
  //Přečte teplotu
  tempair = dht.readTemperature();

  if (isnan(humair) || isnan(tempair)) {
    Serial.println(F("Nepodařilo se přečíst data z DHT22"));
    return;
  }

  Serial.print(F("Vlhkost vzduchu v %: "));
  Serial.print(humair);
  Serial.println("Odesílám vlhkost");
  client.publish("Sklenik/VlhkostVzduchu", humair);
  Serial.print(F("Teplota vzduchu ve °C: "));
  Serial.print(tempair);
  Serial.println("Odesílám teplotu");
  client.publish("Sklenik/TeplotaVzduchu", tempair);
}
```

Obrázek 29 - Zdrojový kód – získání dat ze senzoru DHT22

6.7 Xiaomi Mi Flora – senzor

Komunikace mezi Xiaomi Mi Flora a ESP32 je prováděna výhradně přes bluetooth. Aby mohlo dojít ke spojení těchto dvou zařízení, bylo nutné do Arduino IDE přidat knihovnu podporující komunikaci skrze Bluetooth. K těmto účelům byla využita knihovna ESP32 BLE Arduino.

Po přidání knihovny bylo potřeba zjistit fyzickou adresu Xiaomi Mi Flora. Z tohoto důvodu byl do ESP32 nahrán program pro hledání Bluetooth zařízení v dosahu. Tento program

nebylo nutné vyvíjet. V Arduino IDE jsou totiž po přidání desky ESP32 a BLE knihovny dostupné příklady použitelných programů, které tyto knihovny využívají. Program BLE_scan pro skenování dostupných Bluetooth zařízení v okolí našel senzor Mi Flora s fyzickou adresou C4:7C:8D:6A:D3:D5 a service UUID 0000fe95-0000-1000-8000-00805f9b34fb. Obě tyto informace budou následně využity při získávání dat ze senzoru.

Podprogram nahraný na ESP32, který získává data z Xiaomi Mi Flora nejprve iniciuje bluetooth spojení snahou připojit se ke zjištěné fyzické adrese. Po úspěšném připojení k zařízení je volána charakteristika (záleží na informaci, kterou požadujeme). Na ukázce je použita charakteristika 00001a02-0000-1000-8000-00805f9b34fb pro informaci o stavu baterie a verzi firmware. V tabulce níže je přehled UUID pro komunikaci se senzorem.

Charakteristika (UUID)	Čtení/Zápis	Popis
00001a00-0000-1000-8000-00805f9b34fb	zápis	odeslání příkazu
00001a01-0000-1000-8000-00805f9b34fb	čtení	aktuálně naměřená data
00001a02-0000-1000-8000-00805f9b34fb	čtení	stav baterie a verze firmware
00001a11-0000-1000-8000-00805f9b34fb	čtení	historicky naměřená data
00001a10-0000-1000-8000-00805f9b34fb	zápis	neznámé
00001a12-0000-1000-8000-00805f9b34fb	čtení	čas zařízení

Obrázek 30 - přehled UUID charakteristik Xiaomi Mi Flora [61]

Následně jsou přečtena data v poli hexadecimálního tvaru. Údaje o stavu baterie, nacházející se na nulté pozici, jsou následně uložena do proměnné a odeslána přes MQTT. V ukázce jsou z důvodu optimalizace a názornosti připsány průběžné výpisy stavu zpracování do konzole. Řešení komunikace přes bluetooth vychází z programu vytvořeného Svenem Henkelem. [62]

```

//využití funkce získání dat dané charakteristiky pro informace o zařízení
pRemoteCharacteristic = pRemoteService->getCharacteristic(uuid_version_battery);
Serial.println(pRemoteService->toString().c_str());
if (pRemoteCharacteristic == nullptr) {
    Serial.print("Nepodařilo se připojit k UUID: ");
    Serial.println(uuid_version_battery.toString().c_str());
} else {
    Serial.println(" - Charakteristika nalezena");
}

//Přečte informace o zařízení, včetně stavu baterie
std::string value_battery = pRemoteCharacteristic->readValue();
Serial.println("Hodnoty požadované charakteristiky jsou v: ");
const char *valbat = value_battery.c_str();

Serial.println("Hex: ");
for (int i = 0; i < 16; i++) {
    Serial.print((int)valbat[i], HEX);
    Serial.print(" ");
}
Serial.println(" ");

battery = valbat[0];

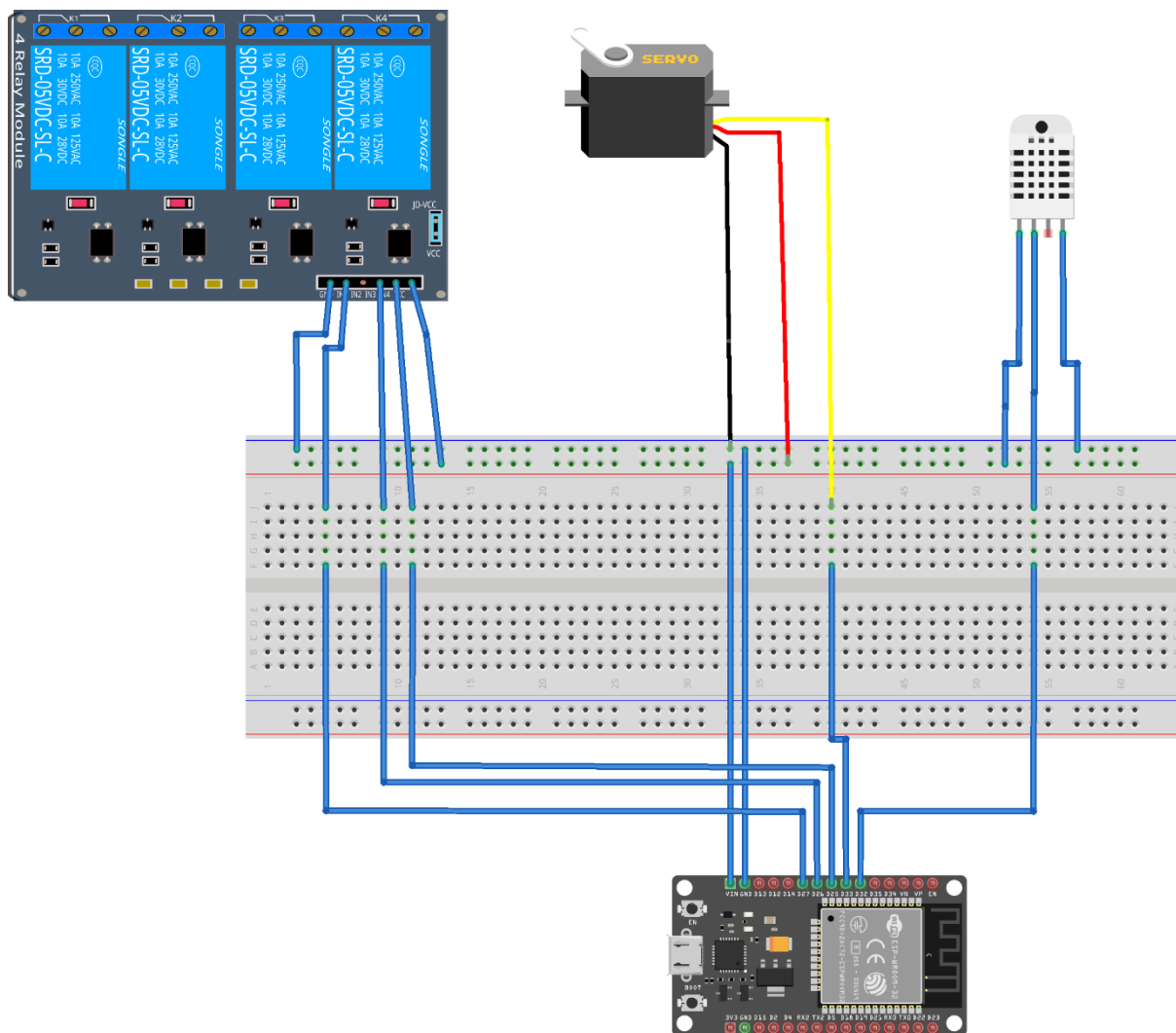
Serial.print("Baterie nabita na: ");
Serial.print(battery);
Serial.println("%");

```

Obrázek 31 - Zdrojový kód – získání stavu baterie Xiaomi Mi Flora

6.8 Realizace HW

Samotný mikrokontroler ESP32 byl pro účely prototypu automatizovaného skleníku instalován na nepájivé pole. Z nepájivého pole lze poté zapojení s ostatními prvky přenést na desku tištěných spojů. Mikrokontroler je možné napájet přes microUSB konektor připojený k 5V adaptéru nebo přes napájecí piny připojením na zdroj stejnosměrného signálu. Pro propojení s periferiemi bylo na nepájivém poli využito dupont kabelů. Tyto typy kabelů byly využity pro rychlost při vytváření a přepojování kabelových spojů. Na obrázku níže je zachycena jedna z možností spojení akčních prvků přes nepájivé pole. Podle potřeby bylo možné změnit využití části pole. Muselo být dodrženo pouze konečné spojení mezi piny kontroleru a periferií, tak jak je definováno v programu.



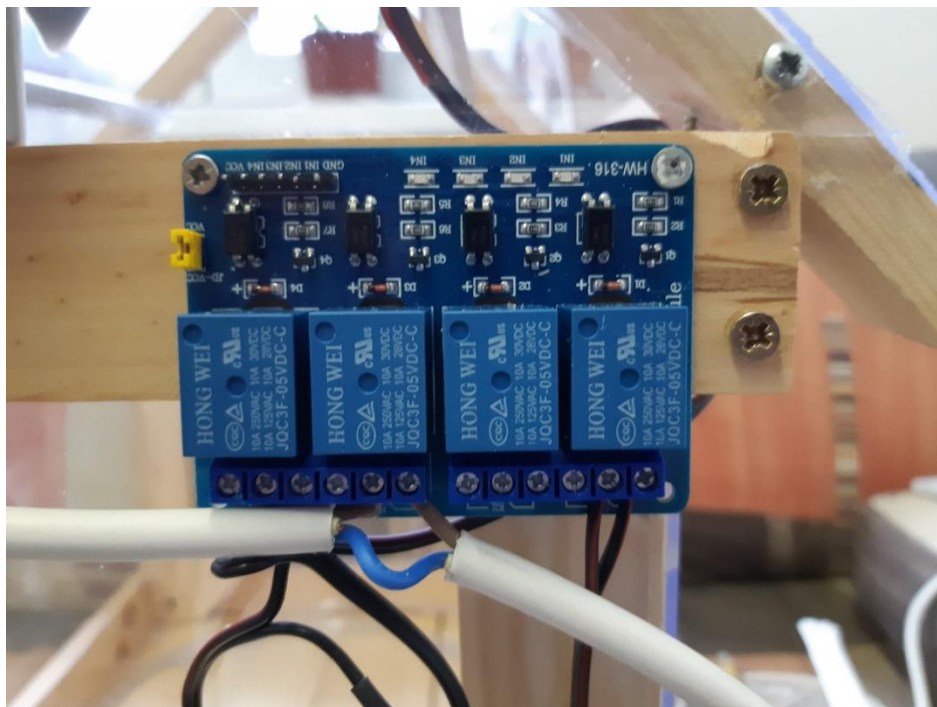
Obrázek 32 - Návrh zapojení hardware přes nepájivé pole

S mikrokontrolerem byl fyzicky propojen senzor DHT22, pro zjišťování teploty a vlhkosti ovzduší, servomotor pro větrání a relé modul ke spínání zavlažování, osvětlení a větráku. Tyto prvky budou dále rozebrány v podkapitolách.

6.8.1 Relé modul

Čtyř-kanálový relé modul pracuje na spínacím napětí 5V. Napájení tohoto modulu bylo přivedeno na vstupní piny GND a VCC. Ostatní piny IN1 až IN4 byly podle využití připojeny k GPIO ESP32, které spínání relé kanálů ovládá signálem. Typ relé je dimenzován na spínání střídavého i stejnosměrného signálu. To bylo důležité vzhledem k různým požadavkům na napájení použitých zařízení. Na obrázku je již nainstalovaný relé modul s připojenými akčními prvky. V této situaci byly prvky z důvodu testování zapojeny v běžně uzavřeném režimu. Byla

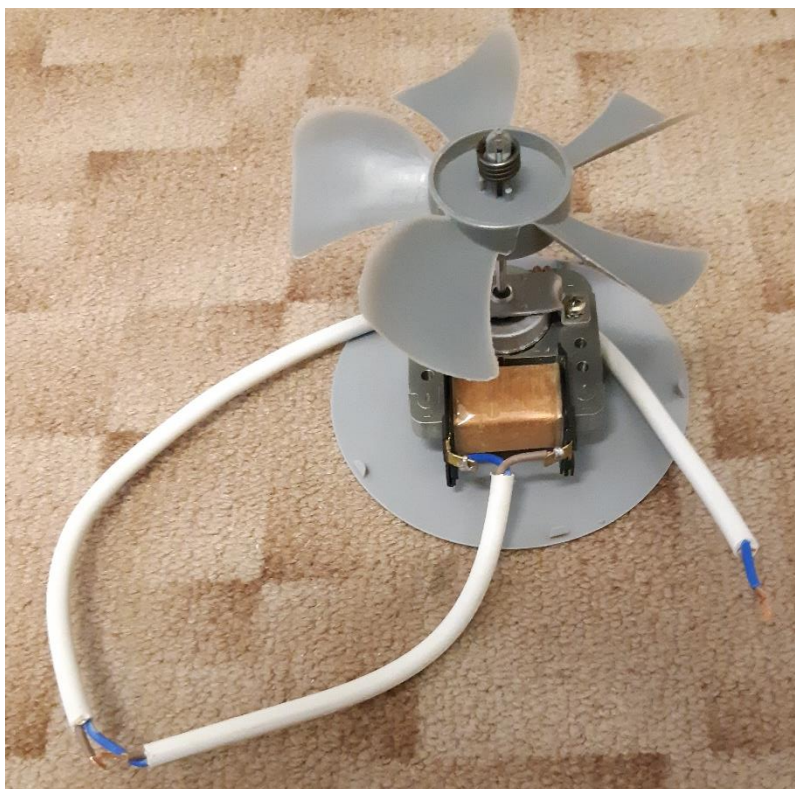
tak ověřována funkčnost relé, kabeláže a periférií. V běžném provozu byly obvody zapojeny v režimu normálně otevřeno (NO), aby mohlo dojít k sepnutí pouze v požadovaném okamžiku.



Obrázek 33 - Relé modul v testovacím zapojení

6.8.2 Větrák

Větrák, který je určen k vyfouknutí teplého vzduchu ven ze skleníku, je spínán pomocí relé. Pracuje na střídavém proudu 230V, který je přivedený přes modul ze zdroje. Jako kabeláž posloužil běžný typ CYKY 3x1,5. Aby šlo připojit periférii do relé, musel být jeden vodič přerušen. Pro ochranu větráku byla na jeho záda připevněna plastová krytka. Pro vyfukování vzduchu ven ze skleníku byl větrák umístěn do jeho štítu. Na obrázku je větrák připraven k instalaci do štítu a k připojení ke spínacímu obvodu.



Obrázek 34 - Větrák pro regulaci teploty

6.8.3 Servo motor

Pro otevírání okna byl v návrhu zvolen a popsán servo motor. Použitý servomotor nepotřeboval žádné speciální úpravy. Bylo ho pouze potřeba připojit pomocí kabeláže k napájecímu napětí 5V a určeným GPIO pinům ESP32. Upravena byla páka připevněná k motoru, pro větší rozsah otevření okna.

6.8.4 Pěstební osvětlení

LED světlo, pracující na napětí 12V DC bylo, z hlediska podmínek panujících uvnitř skleníku, potřeba ochránit před nežádoucími vlivy. Světelný zdroj byl umístěn do ochranné lišty. LED pásek byl vlepen do hliníkového profilu, který působí zároveň jako pasivní chlazení pro osvětlení. Profil byl nakonec zaklopen čirým difuzorem. Pro připojení napájecích kabelů došlo k připájení kabeláže na styčné plochy osvětlení. Na obrázku je zachyceno jedno ze dvou zkompletovaných pěstebních osvětlení.



Obrázek 35 - Konstrukce LED osvětlení

6.8.5 Zavlažování – vodní čerpadlo

Pro potřeby zavlažování byly ke zvolenému vodnímu čerpadlu napojeny dvě hadice zajišťující průchod vody. První je na vstupu čerpadla pro přívod závlahy z vodní nádrže. Druhá je připojena na výstup z čerpadla a je vedena dovnitř skleníku. Pro zavlažování byly vyhotoveny dva typy vodovodu. První typ má ústí pouze na konci hadice, a proto slouží k zavlažování pouze určité části půdy. Druhý typ umožňuje závlahu ve větší části skleníku díky hadici mající dírky po celé její délce. Pro udržení tlaku vody musí být hadice na konci zaslepena.

6.8.6 Zdroj

Zdroj prototypu skleníku je vybrán podle náročnosti jednotlivých komponent systému. Osvětlení bude, při odhadované délce LED pásku nejvýš 2 metry, vyžadovat zdroj stejnosměrného proudu o napájecím napětí 12 V a výkonem minimálně 28,8 W (2x12x1,2). Servomotor pro větrací systém bude potřebovat napájecí napětí 5 V stejnosměrného proudu. Další prvek větracího systému, větrák, pracuje na standardním střídavém proudu o napětí 230 V a není tak potřeba speciálního zdroje. Zavlažovací ponorné čerpadlo pracuje na 12 V stejnosměrného napětí s příkonem 4,2 W. Pro účely napájení použitého hardwaru byl vybrán zdroj pracující na vstupním napětí 230V. Na jeho výstupu je možnost k oddělenému připojení

na stejnosměrné napájení 5V, 12V a 24V. Při napájení 5V DC, je schopen zdroj zvládnout zátěž do 2A, při 12V DC až 7A a prozatím nepoužité 24V k 1.5A.



Obrázek 36 - Napájecí zdroj s vyhotovenou kabeláží

Napájecí zdroj bude připojován do zásuvkové sítě. Poté bude dále distribuovat elektrickou energii dále do systému. Z tohoto důvodu byl na vstupní konektory připevněn kabel CYKY 3x1,5 se zásuvkovou koncovkou.

6.9 Stavba skleníku a instalace systému automatizace

Jako první došlo ke zkonstruování kostry modelu skleníku. Délka je 50 cm, šířka a výška 35 cm (po hřeben 55 cm). Tvar modelu odpovídá reálně používanému rodinnému skleníku. Na kostru byly použity dřevěné laťky o velikosti 4x2 cm. Opláštění skleníku je z čírého plexiskla o tloušťce 2,5 mm. Pro simulaci otevírání střešního okna je jeden střešní panel upevněn pouze panty na začátku a konci hřebenu. Podlážka je v čelní části o 7 cm delší, pro připevnění komponent systému.

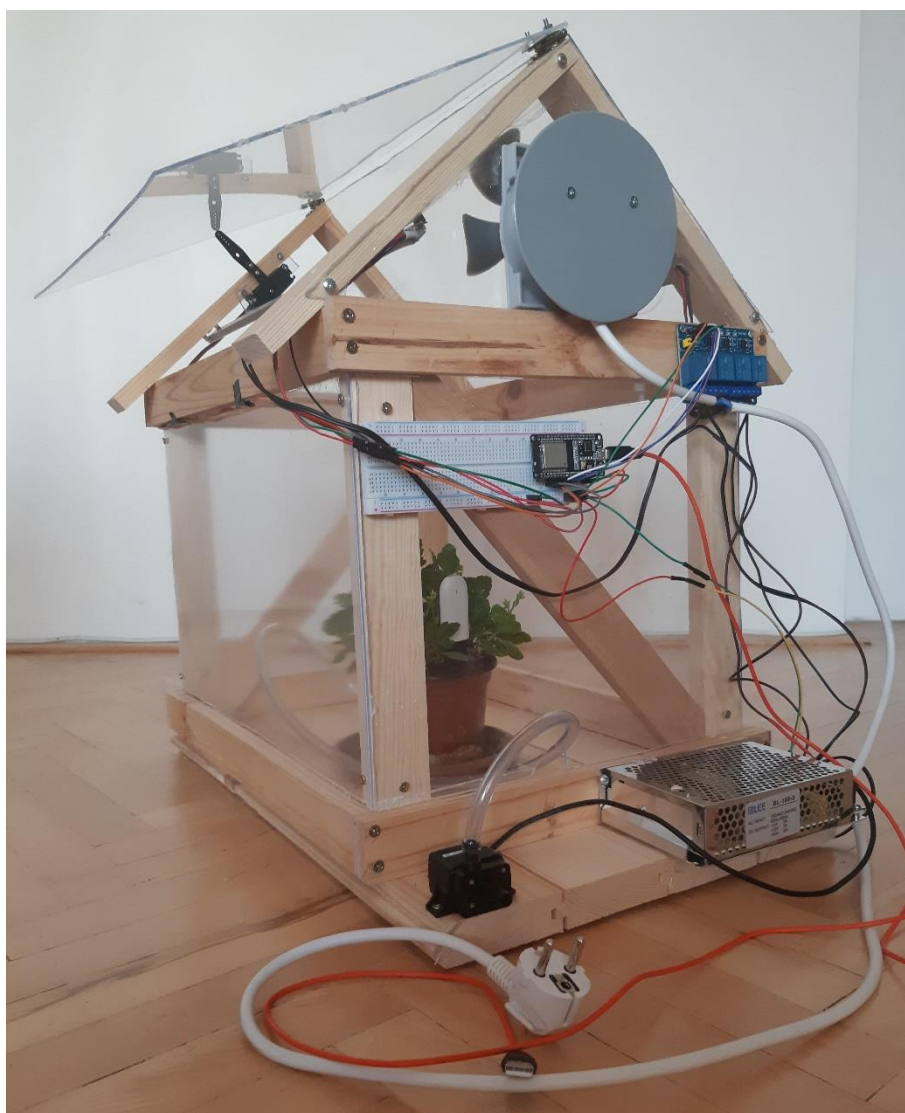
K podlážce byl vpředu připevněn zdroj, distribuující energii akčním prvkům a mikrokontroleru. Napájecí kabeláž je vedena v dvojlinkovém kabelu CYH 2x0,5mm. Vedle

zdroje bylo vpředu připevněno zavlažovací čerpadlo, tak aby mohla být jednoduše nasazena hadice pro čerpání vody. Hadice rozvádějící závlahu v modelu je provedena otvorem v dolní části čelní strany. Relé modul je umístěný nad zdrojem kvůli lepšímu kabelovému rozvodu. Ve štítu je umístěný větrák, který je zasazen do krytu. Motor je umístěn zvenčí, přičemž osa je protažena skrz plexisklo dovnitř. Servomotor pro otevírání střešního okna je umístěn na opačné straně skleníku pod krovem. Dvě lišty pěstebního osvětlení jsou uchyceny v kovových svorkách po obou stranách střechy. Je tak zajištěn rovnoměrný rozptyl světla. Senzor DHT22 byl umístěn dovnitř skleníku a kabely protaženy k ESP32. Xiaomi Mi Flora může být zastrčen kamkoliv do půdy uvnitř prototypu skleníku.



Obrázek 37 - Prototyp skleníku - pohled zevnitř

Mikrokontroler ESP32, který všechny uvedené akční prvky ovládá je v nepájivém poli umístěn na čelní stranu skleníku. Je možné ho napájet samostatně přes USB nebo přivést 5V napětí na jeho piny (například ze zdroje ovládacích prvků skleníku). Nakonec centrální jednotka systému automatizace Raspberry Pi může být umístěna kdekoliv, stačí když bude připojena ke stejné lokální síti jako ESP32. V tomto případě bylo Raspberry Pi připojeno do napájení a k internetu na druhé straně budovy, než byl model skleníku.



Obrázek 38 - Prototyp skleníku – čelní pohled

7 Testování

Při testování byl kladen důraz na ověření správnosti zapojení periférií, jejich funkčnosti a určení správnosti jimi naměřených hodnot. Poté bylo přistoupeno k ověření z hlediska vytvořených programů a navržených algoritmů. Na závěr došlo k otestování funkčnosti systému jako celku.

7.1 Test periférií

Jako první došlo k otestování prvků komunikujících s ESP32. Důraz byl kladen na správnou interpretaci a zpracování naměřených dat, které by mohlo způsobit neočekávané chování systému. Potřebné bylo také otestovat funkčnost akčních prvků.

7.1.1 DHT22

Pro otestování senzoru teploty a vlhkosti vzduchu byl vytvořen jednoduchý program nahraný na ESP32. K mikrokontroleru byly připojeny dva senzory DHT22, které po dobu 24 hodin měřily každých 15 minut teplotu a vlhkost ovzduší. Následně byly naměřené hodnoty mezi oběma senzory porovnány. Mezi senzory rozdíl v naměřených hodnotách nedosahoval ani odchylky uvedené v technickém dokumentu. Měření použitého senzoru tak bylo označeno za správné a bylo možno jej používat v automatizaci.

7.1.2 Xiaomi Mi Flora

Hodnocení funkčnosti senzoru bylo prováděno po jeden den. Nejprve bylo testováno měření teploty každých 15 minut, přičemž výsledné hodnoty byly porovnány s výstupem ze senzoru DHT22. Hodnoty se lišily ve stejných časových úsecích minimálně (nejvíce 1%). Měření intenzity světla senzorem bylo ověřeno posouzením schopnosti senzoru reagovat na změny světla v průběhu dne včetně reakcí na náhlé přisvícení externím zdrojem světla. Na stejném principu byl testován i senzor pro měření půdní vlhkosti, kdy byla ověřována reakce na zavlažování půdy. Senzory v obou případech reagovaly na stimuly podle očekávání.

7.1.3 Relé modul a připojené ovládací prvky

Nejprve byly do modulu připojeny tyto komponenty – větrák a osvětlení v režimu normálně uzavřeného obvodu. Bylo tak otestováno, zda je napájecí kabeláž správně zapojena. Při zapnutí zdroje vše fungovalo podle předpokladů a bylo tak možné testovat dále. Následně byly periferie přepojeny do režimu běžně otevřeného obvodu a testováno, jak relé spínají obvody. Všechny prvky připojené do modulu spínaly během dne, jak bylo požadováno a test byl tak úspěšný.

7.1.4 Servo motor

U serva byla testována schopnost manipulace se střešním oknem. Nejdříve byla otestována, zda byla správně nastavena výchozí pozice serva. Na základě testování tak byla proměnná pro nastavení této pozice zkalibrována pro potřeby prototypu skleníku. Dále byla testována síla motoru, zda zvládne okno otevřít, což se povedlo.

7.2 Test vytvořených programů

Za testování vytvořených programů bylo považována za dostatečnou jejich průběžná optimalizace během vývoje a jejich používání. Byly tak odhaleny chyby při tvorbě SQL příkazů v řídicím systému Node-RED nebo chybná komunikace skrze MQTT. Odhalené nedostatky byly opraveny v systému a aktualizované implementovány.

7.3 Test automatizovaného systému

Součástí systému je i uživatelské rozhraní, které bylo vytvořeno pro interakci s uživateli skleníku. Navržené a zrealizované prostředí jim bylo představeno a ponecháno k vyzkoušení jeho funkčnosti. Připomínky, které během používání systému zaznamenali byly zpracovány a obsaženy do finální podoby. Celkový systém automatizace byl průběžně sledován v testovacím provozu po dobu několika dní, přičemž nebyla objevena žádná anomálie. Po odsouhlasení bude systém moci být implementován v ostrém provozu.

8 Zhodnocení nákladů

Vzhledem ke specifičnosti požadavků na automatizaci skleníku a její aplikaci na prototyp je obtížné srovnávat náklady s existujícími řešeními. Na trhu jsou dostupné převážně profesionální systémy automatizace aplikované na pěstování ve velkém množství za účelem zisku. Náklady na tyto systémy jsou mnohonásobně vyšší, než by byl domácí pěstitel (pěstující pro vlastní potřeby) ochoten i schopen zaplatit. Pro domácí pěstitele jsou na trhu dostupná jednotlivá zařízení, která mohou pomoci k automatizaci pěstování. Tato zařízení by tak mohlo být možné samostatně aplikovat k automatizaci procesů ve skleníku. Příkladem může být zavlažovací počítač Select od výrobce Gardena, který nabízí možnost nastavení intervalů a časovačů zavlažování. Samotné zařízení není čerpadlem. Pouze vypíná nebo zapíná zavlažování připojené na vodovodní kohoutek. Zařízení stojí na portálu Alza.cz 1 290 Kč, tedy bezmála třetinu ceny celého vytvořeného systému v této práci. Počítač zároveň nereflektuje aktuální situaci ve skleníku a není tedy tolik efektivní. Výrobků podobného druhu (osvětlení, regulace teploty) je více, přičemž výsledek jejich srovnání s vytvořeným systémem je stejný jako u zmiňovaného zavlažovacího počítače.

Položka	Cena
Raspberry Pi	1 199 Kč
Xiaomi Mi Flora	599 Kč
ESP32	140 Kč
Čerpadlo	341 Kč
Servo motor	126 Kč
Větrák	20 Kč
LED světla	344 Kč
DHT22	155 Kč
Zdroj	550 Kč
Relé modul	58 Kč
Nepájivé pole + dupont kabely	78 Kč
Kabeláž	60 Kč
Konstrukce modelu skleníku	963 Kč
Celkem	4 633 Kč

Tabulka 5 - Náklady na vyhotovení práce

Ceny jednotlivých komponent jsou uvedeny včetně DPH. Pro snížení nákladů byly některé součásti zakoupeny na čínských e-shopech. Například cena ESP32 na českých portálech se pohybovala v rozmezí 250 Kč až 600 Kč, zatímco na Banggod.com vycházela na 140 Kč. V některých případech byly využity komponenty recyklované z nefunkčních elektronických zařízení. Příkladem může být větrák využitý z nefunkční mikrovlnné trouby. Náklady vynaložené na realizaci práce vyšly na 4 633 Kč včetně konstrukce modelu skleníku. Celkově se tedy vešly do hranice 7 000 Kč, která byla stanovena v analýze požadavků.

9 Závěr

Cílem mé diplomové práce bylo vytvoření prototypu automatizovaného skleníku s použitím Raspberry Pi. Výsledkem je automatizovaný systém využívající centrální jednotku systému, vytvořenou na platformě Raspberry Pi, a sensorovou stanicí ESP32.

Pro úspěšné splnění cíle práce bylo potřeba objasnit teoretická východiska, která jsou uvedena v první části této práce. Je zde vysvětlena problematika internetu věcí a jeho architektury. Dále byly zpracovány informace o komunikačních protokolech a přenosových technologiích, které je možné využít. Součástí jsou i materiály o využitelném softwaru a hardwaru. Představena byla i centrální jednotka systému automatizace, kterou je Raspberry Pi.

Po uvedení do problematiky byla vypracována analýza požadavků na vlastní práci. V ní došlo k otevření diskuze s účastníky, kteří obhospodařují vlastní skleník. Na základě jejich požadavků byly sestaveny preference na automatizovaný systém aplikovatelný na prototyp skleníku. Následně byl vypracován návrh celého systému automatizace. Nejprve byla popsána a vytvořena architektura systému. Centrální jednotkou systému byl určen mikropočítač Raspberry Pi. Softwarová architektura se skládá z uživatelského rozhraní, databáze MySQL, MQTT broker Eclipse Mosquitto, programu na sensorové stanici ESP32 a systému Node-RED, který má za úkol zpracování naměřených dat, jejich interpretaci a rozhodování o chování systému. Ke komunikaci mezi sensorovou stanicí a rozhodovacím systémem v Node-RED byl určen protokol MQTT.

Podle návrhu byl následně implementován systém automatizace. Na Raspberry Pi byl nakonfigurován MQTT broker, databáze a systém Node-RED. V Node-RED bylo vytvořeno uživatelské rozhraní pomocí kterého bylo možné interpretovat naměřená a uložená data. Vedle uživatelského rozhraní byly v Node-RED vytvořeny flow pro zpracování a uložení dat, naprogramovány funkce pro chování systému a definování jeho reakcí na zpracovávaná data. Systém komunikuje s databází a přes MQTT se sensorovou stanicí ESP32. Mikrokontroler ESP32 je připojen fyzicky k senzoru, který měří teplotu i vlhkost ovzduší, a také bezdrátově k senzoru Xiaomi Mi Flora. Sensorová stanice zároveň ovládá akční prvky, které se starají o chod skleníku. Pro chod sensorové stanice byl vytvořen program v prostředí Arduino IDE využívající dostupné knihovny. Celý systém byl poté aplikován na zkonstruovaném prototypu skleníku.

Mezi hlavní výhody navrženého a vytvořeného automatizovaného skleníku patří nízká cena vzhledem k funkcionalitě a dostupným komerčním řešením, přehledné uživatelské rozhraní poskytující možnost vzdálené správy skleníku a konfiguraci systému na základě mezních proměnných. Další výhodou je použití bezdrátových technologií a senzoru Xiaomi Mi Flora. Těchto zařízení se dá ve skleníku využít více. Dojde tak ke sběru většího množství dat o pěstovaných rostlinách. Systém je do budoucna možné rozšířit o další senzorové stanice, ovládací prvky nebo senzory. Rozšíření by jen bylo potřeba zakomponovat do stávajícího systému.

Žijeme v době, kdy se informační technologie neustále vyvíjí a stávají se dostupnější pro většinu populace. Pojmy udržitelnost, ekologie a efektivita jsou celospolečenskými tématy. Mnohdy přerůstají i v životní styl. Vzniká mnoho organizací na podporu stále ubývající přírody. Naším hlavním společným cílem by měla být snaha o využití těchto technologií směrem k udržitelnosti, efektivitě a ekologii. Právě řešení navržená v této práci mají potenciál zlepšit hospodaření s obnovitelnými zdroji, zvýšit efektivitu v jejich využívání a posunout propojení technologií s přírodou blíže ke všem lidem.

10 Bibliografie

- [1] About us. *Raspberry Pi* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.raspberrypi.org/about/>
- [2] What is a Raspberry Pi?. *Raspberry Pi* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [3] What is a Raspberry Pi?. *Opensource.com* [online]. [cit. 2020-10-30]. Dostupné z: <https://opensource.com/resources/raspberry-pi>
- [4] JOHNSON, BERNADETTE. How the Raspberry Pi Works. *How Stuff Works* [online]. [cit. 2020-10-30]. Dostupné z: <https://computer.howstuffworks.com/raspberry-pi2.htm>
- [5] AKBARI, Mohammadreza. Tutorial: Raspberry Pi GPIO Programming Using Python [Full Guide]. *ElectroPeak* [online]. [cit. 2020-10-30]. Dostupné z: <https://electropeak.com/learn/tutorial-raspberry-pi-gpio-programming-using-python-full-guide/>
- [6] GPIO. *Raspberry Pi* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>
- [7] SANTOS, Sara. ESP32 vs ESP8266 – Pros and Cons. *Maker Advisor* [online]. 2020 [cit. 2020-10-30]. Dostupné z: <https://makeradvisor.com/esp32-vs-esp8266/>
- [8] BASILE, Fiore. ESP8266 Introduction. *Networking ESP8266* [online]. 2015 [cit. 2020-10-30]. Dostupné z: <http://fabacademy.org/archives/2015/doc/networking-esp8266.html>
- [9] ESP8266EX Datasheet. *Espressif* [online]. 2020 [cit. 2020-10-30]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [10] ESP8266 Technical Reference. *Espressif* [online]. 2020 [cit. 2020-10-30]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
- [11] What is ESP8266?. *The internet of things with ESP8266* [online]. [cit. 2020-10-30]. Dostupné z: <http://esp8266.net/>

- [12] *Getting Started with ESP8266 WiFi Transceiver (Review)* [online]. [cit. 2020-11-10]. Dostupné z: <https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/>
- [13] ESP32-DevKitC. *Espressif* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.espressif.com/en/products/hardware/esp32-devkitc/overview>
- [14] *ESP32 Technical Reference Manual* [online]. 4.3. Šanghaj, 2020 [cit. 2020-10-30]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [15] ESP32-DevKitC V4 Getting Started Guide. *Espressif* [online]. [cit. 2020-10-30]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkitc.html>
- [16] Vývojová deska ESP32. *Arduino-shop* [online]. 2017 [cit. 2020-10-30]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/vyvojova-deska-esp32.html>
- [17] Arduino Introduction. *Arduino.cc* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.arduino.cc/en/guide/introduction>
- [18] MARIAN, Popescu. Arduino Uno Pinout. *ElectroSchematics* [online]. [cit. 2020-10-30]. Dostupné z: <https://www.electroschematics.com/arduino-uno-pinout/>
- [19] ARDUINO UNO REV3. *Arduino.cc* [online]. [cit. 2020-10-30]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [20] VODA, Zbyšek. *Průvodce světem Arduina* [online]. Bučovice: Martin Stříž, 2017 [cit. 2020-10-30]. ISBN 978-80-87106-93-8.
- [21] LIGO, George. Electromagnetic Relay. *ElectroSome* [online]. 2012 [cit. 2020-10-30].
- [22] Elektromagnetické relé. *Mendelova univerzita v Brně: eknihovna* [online]. [cit. 2020-10-30]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=629
- [23] Xiaomi Flora Plant Monitor teardown and analysis. *Hackerspace* [online]. 2020 [cit. 2020-10-30].
- [24] Xiaomi Mi Flora Monitor - chytrý senzor pro rostliny. *IStage* [online]. [cit. 2020-10-30].

- [25] Xiaomi Mi Flora. *Aliexpress.com* [online]. [cit. 2020-11-10]. Dostupné z: <https://ae01.alicdn.com/kf/HTB1WUrbcDnI8KJjSszbq6z4KFXaB/Original-Xiaomi-Mijia-Mi-Flower-Monitor-Plants-Grass-Soil-Water-Light-Smart-Tester-For-Xiaomi-Flora.jpg>
- [26] BUCKLEY, Ian. What Is a Breadboard and How Does It Work?. *Make Use Of* [online]. 2018 [cit. 2020-10-30].
- [27] Nepájivé kontaktní pole. *Ondrův web* [online]. 2011 [cit. 2020-10-30].
- [28] POHANKA, Pavel. Internet věcí. *Pavelpohanka.cz* [online]. 2020 [cit. 2020-10-31].
- [29] ROUSE, Margaret. Internet of things. *Techtarget: IoT Agenda* [online]. 2020 [cit. 2020-10-31]. Dostupné z: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- [30] EVANS, Dave. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. *Cisco* [online]. [cit. 2020-10-31].
- [31] Internet of Things: Architectures, Protocols, and Applications. SETHI, Pallavi a Smruti R. SARANGI. *Journal of Electrical and Computer Engineering* [online]. 2017, **2017**(2017), 25 [cit. 2020-11-07]. Dostupné z: doi:10.1155/2017/9324035
- [32] ITU-T. Overview of the Internet of things. *GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS* [online]. International Telecommunication Union, 2012, **2012**(2060), 22 [cit. 2020-11-07]. Dostupné z: <http://handle.itu.int/11.1002/1000/11559>
- [33] ZANDL, Patrick. Edge Computing a Fog Computing: o výhodách mlžných sítí. *Lupa.cz* [online]. 2016 [cit. 2020-10-31].
- [34] CloudMQTT Documentation. *CloudMQTT* [online]. [cit. 2020-10-31].
- [35] SANTOS, Rui. What is MQTT and How It Works. *Random Nerd Tutorials* [online]. 2017 [cit. 2020-10-31].
- [36] LABIOD, Houda, Hossam AFIFI a Constantino DE SANTIS. *Wi-Fi™, Bluetooth™, Zigbee™ and WiMax™*. Dordrecht: Springer, 2007. ISBN 978-1-4020-5396-2.
- [37] How does Bluetooth work?. *ElProCus* [online]. [cit. 2020-11-10]. Dostupné z: <https://www.elprocus.com/how-does-bluetooth-work/>

- [38] ESP-NOW: User Guide. *Espressif* [online]. 2016 [cit. 2020-10-31].
- [39] SANTOS, Sara. Getting Started with ESP-NOW (ESP32 with Arduino IDE). *Random Nerd Tutorials* [online]. 2020 [cit. 2020-10-31].
- [40] HOELSCHER, Penny. What is a relay attack (with examples) and how can you prevent them?. *Comparitech* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.comparitech.com/blog/information-security/what-is-relay-attack/>
- [41] ESP-NOW. *Espressif: ESP-IDF Programming Guide* [online]. [cit. 2020-10-31].
- [42] AZZOLA, Francesco. How to connect ESP32 and ESP8266 using ESP-Now protocol. *Surviving With Android* [online]. 2020 [cit. 2020-10-31].
- [43] SSH Protocol. *Ssh.com* [online]. [cit. 2020-10-31].
- [44] YLONEN, Tatu, Chris LONVICK, ed. *The Secure Shell (SSH) Protocol Architecture* [online]. [cit. 2020-10-31]. Dostupné z: doi:10.17487/RFC4251
- [45] RICHARDSON, T., Q. STAFFORD-FRASER, K. R. WOOD a A. HOPPER. *Virtual network computing* [online]. IEEE Internet Computing. 2. , 33-38 s. [cit. 2020-10-31]. Dostupné z: doi:10.1109/4236.656066
- [46] ROUSE, Margaret. Virtual network computing (VNC). *Techtarget* [online]. 2006 [cit. 2020-10-31].
- [47] MITCHELL, Bradley. What Is Virtual Network Computing (VNC)?. *Lifewire* [online]. 2020 [cit. 2020-10-31].
- [48] Real VNC. *Real VNC* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.realvnc.com/en/company-profile/>
- [49] Recent trends in next generation terabit Ethernet and gigabit wireless local area network. *2016 International Conference on Signal Processing and Communication (ICSC)* [online]. IEEE, 2016, s. 106-110 [cit. 2020-10-31]. ISBN 978-1-5090-2684-5. Dostupné z: doi:10.1109/ICSPCom.2016.7980557
- [50] SHUANG, Yu. *IEEE 802.3™ 'STANDARD FOR ETHERNET' MARKS 30 YEARS OF INNOVATION AND GLOBAL MARKET GROWTH* [online]. In: . [cit. 2020-10-31]. Dostupné z: http://standards.ieee.org/news/2013/802.3_30anniv.html

- [51] XEROX. *Alto: A Personal Computer System Hardware Manual* [online]. 1976 [cit. 2020-10-31]. Dostupné z: http://www.bitsavers.org/pdf/xerox/alto/Alto_Hardware_Manual_Aug76.pdf
- [52] SANTITORO, Ralph. *Metro Ethernet Services: A Technical Overview* [online]. 2003 [cit. 2020-10-31]. Dostupné z: https://www.mef.net/Assets/White_Papers/Metro-Ethernet-Services.pdf
- [53] XEROX, 3COM a John F. SHOCH. *Evolution of the Ethernet Local Computer Network* [online]. [cit. 2020-10-31]. Dostupné z: <https://ethernethistory.typepad.com/papers/EthernetEvolution.pdf>
- [54] COX, Tim a Steven FERNANDES. *Raspberry Pi 3 Cookbook for Python Programmers: Unleash the potential of Raspberry Pi 3 with over 100 recipes*. 3. Birmingham: Packt Publishing, 2018. ISBN 9781788629874.
- [55] LANGA, Lukasz. *Python 3.9.0a3 now available for testing* [online]. In: . [cit. 2020-10-31]. Dostupné z: https://pythoninsider.blogspot.com/2020/01/python-390a3-now-available-for-testing.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+PythonInsider+%28Python+Insider%29
- [56] PYTHON SOFTWARE FOUNDATION. What is Python? Executive Summary. *Python* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.python.org/doc/essays/blurb/>
- [57] HEATH, Nick. How IBM's Node-RED is hacking together the internet of things. *Tech Republic* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.techrepublic.com/article/node-red/>
- [58] Co je to Node-RED a ukázka práce. *Hardwario* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.hardwario.com/cs/academy/co-je-node-red/>
- [59] LEWIS, Karen. *Fantastic news for Open Source and IoT fans: Node-RED visual programming for the Internet of Things (IoT) is now a JS Foundation Project* [online]. [cit. 2020-10-31]. Dostupné z: <https://www.ibm.com/blogs/internet-of-things/open-source-iot/>
- [60] Senzor DHT22. *Hwkitchen.cz* [online]. [cit. 2020-11-10]. Dostupné z: <https://www.hwkitchen.cz/snimac-teploty-a-vlhkosti-dht22/>

- [61] Xiaomi Flowercare API. *Awesome Open Source* [online]. [cit. 2020-11-10]. Dostupné z:
<https://awesomeopensource.com/project/vrachieru/xiaomi-flower-care-api>
- [62] HENKEL, Sven. Xiaomi Mi Flora. *Github.com* [online]. [cit. 2020-04-30]. Dostupné z:
<https://github.com/sidddy/flora/blob/master/flora/flora.ino>