



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

## NÁVRH A REALIZACE AUTOMATICKÉHO SYSTÉMU PRO ZAVLAŽOVÁNÍ POKOJOVÝCH ROSTLIN

DESIGN AND IMPLEMENTATION OF AN AUTOMATIC SYSTEM FOR WATERING INDOOR PLANTS

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Ondřej Szymik

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Stanislav Věchet, Ph.D.

BRNO 2024

# Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky  
Student: **Ondřej Szymik**  
Studijní program: Mechatronika  
Studijní obor: bez specializace  
Vedoucí práce: **doc. Ing. Stanislav Věchet, Ph.D.**  
Akademický rok: 2023/24

Ředitel ústavu Vám v souladu se zákonem č.1111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## **Návrh a realizace automatického systému pro zavlažování pokojových rostlin**

### **Stručná charakteristika problematiky úkolu:**

Práce se zabývá realizací automatického systému pro zavlažování pokojových rostlin. Systém má být navržen tak, aby bylo možné každou pokojovou rostlinu pěstovat samostatně a tedy byl systém modulární a umožnil měřit nutné environmentální veličiny a za vlašovat každou rostlinu nezávisle na ostatních.

### **Cíle bakalářské práce:**

Navrhněte vhodnou architekturu celého systému.

Vyberte vhodnou sensoriku a akční členy.

Systém sestavte a ověřte jeho funkcionalitu.

### **Seznam doporučené literatury:**

LIGHT, R. A., Mosquitto: server and client implementation of the MQTT protocol, The Journal of Open Source Software, vol. 2, no. 13, May 2017, DOI: 10.21105/joss.00265.

BALÁTĚ, J.: Technické prostředky automatického řízení. Praha, SNTL 1986.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty

## Abstrakt

Tato práce je zaměřena na návrh a realizaci automatického zavlažovacího systému pro pokojové rostliny. Důraz je kladen na modularitu systému, tedy aby bylo možné zavlažovat rostliny a měřit jejich environmentální veličiny nezávisle na sobě. Součástí práce je návrh vhodné architektury systému, výběr vhodné sensoriky a akčních členů, sestavení systému a ověření jeho funkčnosti.

## Summary

This work is focused on the design and realization of an automatic irrigation system for houseplants. The emphasis is on the modularity of the system, i.e. to be able to irrigate plants and to measure their environmental variables independently. The work includes the design of a suitable system architecture, the selection of suitable sensors and actuators, the assembly of the system and the verification of its functionality.

## Klíčová slova

MQTT komunikační protokol, Raspberry Pi Zero, vývojová deska ESP32, Internet věcí

## Keywords

MQTT messaging protocol, Raspberry Pi Zero, ESP32 development board, Internet of Things

## Bibliografická citace

SZYMIK, Ondřej. *Návrh a realizace automatického systému pro zavlažování pokojových rostlin* [online]. Brno, 2024 [cit. rrrr-mm-dd]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/158208>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce doc. Ing. Stanislav Věchet, Ph.D..

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Stanislava Věcheta, Ph.D. a že jsem uvedl všechny zdroje a literaturu, ze kterých jsem během vypracování čerpal.

**Ondřej Szymik**

V Brně 24. května 2024

.....

Děkuji všem, kteří při mně během práce na tomto bakalářském projektu stáli či mě jakkoliv podporovali, především panu doc. Ing. Stanislavu Věchetovi, Ph.D. za jeho cenné rady a vedení.

**Ondřej Szymik**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
<b>2</b>	<b>Rešeršní studie</b>	<b>9</b>
2.1	Automatické zavlažovací systémy . . . . .	9
2.2	Využitelný hardware . . . . .	9
2.2.1	Raspberry Pi Zero . . . . .	9
2.2.2	Vývojová deska ESP32 DevKit V1 . . . . .	10
2.2.3	Senzory a akční členy . . . . .	12
2.3	MQTT komunikační protokol . . . . .	14
<b>3</b>	<b>Analýza problému</b>	<b>17</b>
<b>4</b>	<b>Realizace procesu řešení</b>	<b>19</b>
4.1	Realizace modulární jednotky . . . . .	19
4.2	Komunikace mezi jednotkami . . . . .	27
4.3	Realizace čerpání vody . . . . .	27
4.4	Použité knihovny . . . . .	30
<b>5</b>	<b>Ověření funkcionality systému</b>	<b>31</b>
5.1	Sběr a zápis dat . . . . .	31
5.2	Testování senzorů . . . . .	31
<b>6</b>	<b>Závěr</b>	<b>35</b>
	<b>Seznam literatury</b>	<b>36</b>

# 1 Úvod

V dnešní době se stále více rozvíjí trend chytrých domácností. Chytrou domácností se myslí implementace moderních technologií pro zjednodušení běžných činností, jako například automatizované vytápění a větrání místností, rozsvěcování světel, virtuální hlasoví asistenti, a v neposlední řadě právě třeba automatická péče o rostliny.

Pro udržení zdraví a vitality rostlin je důležité mimo jiné jejich správné zavlažování. Aby tohoto cíle bylo dosaženo, je výhodné do takovéto chytré domácnosti zařadit automatický zavlažovací systém pokojových rostlin.

Tato práce se věnuje návrhu a realizaci právě takového systému, a to s požadavkem na modularitu systému, tedy aby bylo možné dle potřeb uživatele zavlažovat vícero rostlin a měřit jejich environmentální veličiny zcela nezávisle na sobě.

Práce je členěna do čtyř hlavních částí: rešeršní studie, analýza problému, realizace procesu řešení a ověření funkcionality systému. Rešeršní studie rozebírá již dostupné automatické zavlažovací systémy a následně senzory a akční členy využitelné v takovém systému (jako třeba senzory pro monitorování teploty a tlaku vzduchu, vlhkosti půdy, koncentrace  $\text{CO}_2$  ve vzduchu, senzor pro indikaci nízké hladiny vody v nádržce a čerpadlo), dále řídicí jednotky (jednodeskový počítač či mikrokontrolér) vhodné pro řízení celého systému a následně pak komunikační protokol vhodný pro využití v Internetu věcí. Další část se věnuje analýze problému, tedy zhodnocení poznatků z rešeršní studie a popis jejich využití v projektu. Kapitola Realizace procesu řešení se pak věnuje návrhu modulární jednotky a její komunikace se řídicí jednotkou a následnou samotnou realizací. Závěrečná kapitola pojednává o testování a ověření funkčnosti systému.



## 2 Rešeršní studie

Tato kapitola se věnuje rešeršní studii zavlažovacích systémů, komponent, které se v takovém systému dají použít, a dále komunikačního protokolu důležitého pro automatizaci systému a splnění požadavku modularity.

### 2.1 Automatické zavlažovací systémy

Existují různé běžně dostupné varianty automatických zavlažovacích systémů. Za zmínku stojí například celá řada různých variant zavlažovacího systému od firmy Blumat<sup>1</sup>. Ten funguje na principu jednoduchého vodou naplněného keramického „senzoru“ ve tvaru špičatého kuželu zapíchnutého do půdy. Při vysychání půdy tahají kořeny rostliny z kužele vodu, vytvoří se tak podtlak a ten buď jednoduše tahá vodu ze zásobníku, do kterého je zavedena z kužele hadička, nebo otevře ventil k přívodu vody, a nebo který je měřen senzorem tlaku, na základě kterého se spouští zavlažování.

Dalšími systémy, které stojí za zmínku, jsou různé automatické jednotky s čerpadly a hadičkovými rozvody k jednotlivým rostlinám, vhodné jak pro pokojové rostliny, tak i pro venkovní použití na terasy, balkóny apod. Dá se na nich nastavit frekvence a délka zavlažování<sup>2</sup>.

K pokročilejším systémům patří například systém EcoDuino<sup>3</sup> vytvořený společností DFRobot. Ten je založen na snímání hodnot ze senzorů vlhkosti, teploty, intenzity osvětlení apod. a jejich následném zpracování v mikrokontroléru, který (mimo jiné) dává pokyn čerpadlu k zavlažování.

### 2.2 Využitelný hardware

V následující části je přiblížený hardware využitelný pro použití v automatickém zavlažovacím systému.

#### 2.2.1 Raspberry Pi Zero

Jak se můžeme dočíst v [7] nebo [10], Raspberry Pi Zero je malý jednodeskový počítač vyvinutý britskou společností Raspberry Pi Foundation ve spolupráci s americkou společností Broadcom Inc. Disponuje následujícím:

- procesorem Broadcom BCM2835 s 1GHz jednojádrovým CPU<sup>4</sup>,
- vestavěnou pamětí RAM 512 MB,

<sup>1</sup><https://www.tropf-blumat.cz/>

<sup>2</sup>například <https://www.biano.cz/produkt/v/185564699-vidaxl-automaticky-zavlazovaci-system-s-postrikovacem-a-casovacem-41597> nebo [https://www.eprumysl.cz/solarni-zavlazovaci-system-number\\_of\\_sections-kapac-5-m-hadice/](https://www.eprumysl.cz/solarni-zavlazovaci-system-number_of_sections-kapac-5-m-hadice/)

<sup>3</sup><https://www.dfrobot.com/product-641.html>

<sup>4</sup>Central Processing Unit - centrální procesorová jednotka

- externí paměť na microSD kartě,
- připojením k WiFi a Bluetooth,
- miniHDMI portem,
- microUSB OTG portem,
- microUSB napájecí portem,
- 40 GPIO<sup>5</sup>,
- CSI<sup>6</sup> konektorem,
- operačním systémem například: Raspberry Pi OS a
- možností tvorby projektů například v jazyce Python verze 3.



Obrázek 2.1: Raspberry Pi Zero

### 2.2.2 Vývojová deska ESP32 DevKit V1

ESP32 DevKit V1 je vývojová deska od výrobce DoIT založena na SoC<sup>7</sup> od společnosti Espressif. Tato vývojová deska je pravděpodobně nejoblíbenější a nejpoužívanější z řady mikrokontrolérů postavených na ESP32. Schéma vývojové desky a zapojení vývodů na ní je na obrázku 2.2. Dále se z [5] a [8] můžeme dozvědět následující dispozice desky:

- 34 dostupných GPIO z 40 celkových,
- vestavěná modrá LED dioda připojená na pin GPIO2,
- 3 asynchronní UART<sup>8</sup> periferie,

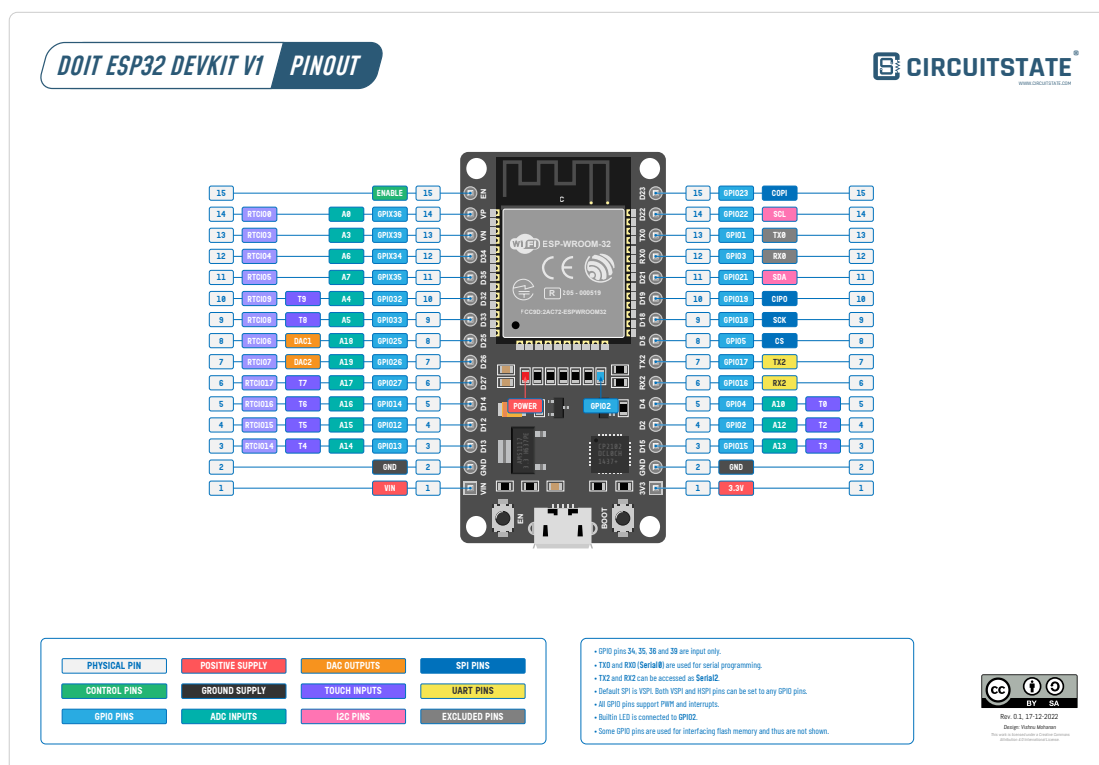
<sup>5</sup>General-Purpose Input/Output - univerzální vstupní/výstupní pin

<sup>6</sup>Camera Serial Interface - sériové rozhraní pro připojení kamery

<sup>7</sup>System on Chip - systém na čipu

<sup>8</sup>Universal Asynchronous Receiver-Transmitter - univerzální asynchronní přijímač-vysílač

- 4 SPI<sup>9</sup>,
- 16 dostupných kanálů ADC<sup>10</sup>,
- 2 DAC<sup>11</sup>,
- 10 dotykových kapacitních kanálů,
- 2 periferie I<sup>2</sup>C<sup>12</sup>,
- 16 na sobě nezávislých PWM<sup>13</sup> kanálů,
- připojení k WiFi a Bluetooth a
- programovatelnost v jazyce založeném na programovacích jazycích C/C++.



Obrázek 2.2: Schéma zapojení vývodů vývojové desky ESP32 DevKit V1. Převzato z [5]

<sup>9</sup>Serial Peripheral Interface - sériové periferní rozhraní

<sup>10</sup>Analog to Digital Converter - převodník analogového signálu na digitální

<sup>11</sup>Digital to Analog Converter - převodník digitálního signálu na analogový

<sup>12</sup>Inter-Integrated Circuit - vnitřní integrovaný obvod, též známý pod TWI - Two Wire Interface - dvoudrátové rozhraní

<sup>13</sup>Pulse Width Modulation - pulzně šířková modulace

### 2.2.3 Sensory a akční členy

#### Senzor teploty vzduchu a atmosférického tlaku BMP280

Senzor teploty vzduchu a atmosférického tlaku BMP280 od firmy Bosch je usazený na nízkonákladovém modulu s nízkou spotřebou a dále dvěma kondenzátory filtrujícími napětí ze zdroje, třemi pull-up rezistory a jedním pull-down rezistorem [2]. Tento senzor komunikuje s mikrokontrolérem přes rozhraní buď I<sup>2</sup>C, nebo SPI. Pro implementaci v kódu je možné využít již předchystanou volně dostupnou knihovnu `Adafruit_BMP280.h` od společnosti Adafruit (dostupnost viz sekce [4.4]). Modul je na obrázku 2.3.



Obrázek 2.3: Modul se senzorem teploty a atmosférického tlaku BMP280

#### Sensory vlhkosti půdy

Senzor vlhkosti půdy YL-38/69 se skládá ze dvou částí: procesní modul YL-38 a sonda YL-69 (viz obrázek 2.4). Funguje na principu měření odporu mezi nožičkami sondy, nemůžeme přesně určit měřitelnou jednotku vlhkosti, číselný výstup senzoru můžeme pouze namapovat na nějakou stupnici a pomocí ní určit, zda je půda suchá, středně vlhká, nebo vlhká [6]. Pro projekt automatického zavlažovacího systému je to zcela postačující. Procesní modul dále disponuje trimrem pro nastavení detekčního prahu vlhkosti užitečným pro zpracování signálu bez využití procesní jednotky. Při zpracování signálu digitálně v mikrokontroléru, kdy tuto hranici můžeme jednoduše nastavit v kódu, tato funkce pozbývá smyslu.

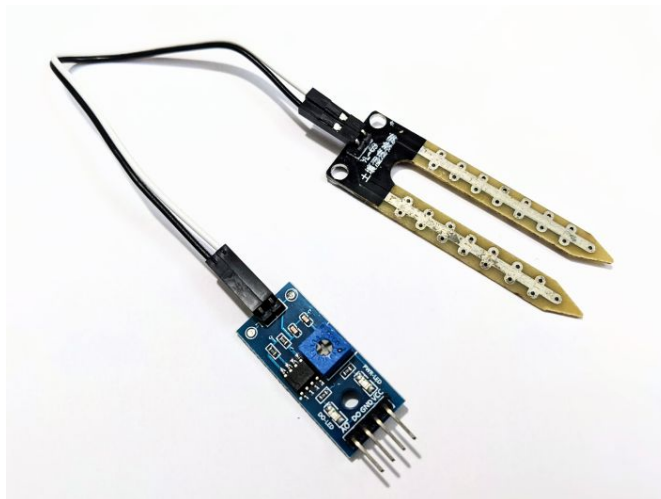
Existuje celá řada jiných senzorů vlhkosti půdy. Za zmínku stojí například senzor od společnosti DFRobot<sup>14</sup>, který funguje na kapacitním principu a který je potažený antikoročním nátěrem zvyšujícím odolnost sondy.

#### Senzor koncentrace CO<sub>2</sub> MHZ19-B

Senzor koncentrace CO<sub>2</sub> MHZ19-B je od firmy Winsen Electronics. Funguje na principu NDIR<sup>15</sup>, o kterém se můžeme více dočíst v [9]. Komunikace probíhá přes sériovou

<sup>14</sup><https://www.dfrobot.com/product-2054.html>

<sup>15</sup>Nondispersive Infrared - nedisperzní infračervený



Obrázek 2.4: Senzor vlhkosti půdy

linku (UART) nebo pomocí PWM. Rozsah měření je 0 až 5000 ppm<sup>16</sup>. Je tedy zcela dostačující pro použití ve vnitřních místnostech, kde se CO<sub>2</sub> vyskytuje v koncentraci do 1500 ppm [1]. Opět je možné využít již existující knihovnu, konkrétně MHZ19.h (dostupnost viz sekce [4.4]).



Obrázek 2.5: Senzor koncentrace CO<sub>2</sub>. Převzato z [<https://dratek.cz/arduino/7731-detektor-oxidu-uhliciteho-co2-mh-z19.html>]

### Plovákový senzor hladiny vody

Plovákový senzor hladiny vody je založený na jednoduchém sepnutí magnetického

<sup>16</sup>parts per million - počet částic ku milionu

snímače při dotyku plovoucí části senzoru spodní respektive horní hranice jeho pohybu. Senzor dokáže reagovat na sepnutí při instalaci ve dvou směrech, pro tuto změnu je však nutné otočit i plovoucí část senzoru. Na obrázku 2.6 můžeme vidět, jak tento senzor vypadá. Pro použití s mikrokontrolérem je třeba připojit mezi pin, který snímá sepnutí, a zem pull-down rezistor o velikosti 1 k $\Omega$  až 10 k $\Omega$  [4]. Senzor lze ale využít i samostatně jako spínač do celkového zatížení 50 W.



Obrázek 2.6: Plovákový senzor hladiny vody

### Mini ponorné čerpadlo

Ponorné čerpadlo od firmy Eses pohání stejnosměrný motorek, který je napájen napětím 2,5 až 6 V, dosahuje výkonu 0,4 až 1,5 W a pracuje s průtokem 80 až 120 l/hod. Dokáže přečerpat vodu až do výšky 110 cm. Čerpadlo má sací otvor zespodu a výtokový otvor je pak vyveden tak, že na něj lze napojit hadičku o vnitřním průměru 6 mm. Tyto informace jsou dostupné z eshopu, kde bylo čerpadlo zakoupeno<sup>17</sup>.

Jelikož čerpadlo má na mikrokontroléry velký odběr proudu, je třeba jej v takovémto případě zapojit na vlastní zdroj napětí, toho může být dosaženo využitím například spínacího modulu IRF520 MOSFET, ten lze ovládat z příslušných pinů mikrokontroléru už bez přílišného odběru proudu. Výhodou dále je, že na modul je možné přivádět signál PWM a ovládat tím velikost napětí přiváděného na čerpadlo. Tento modul si můžeme prohlédnout na obrázku 2.8.

## 2.3 MQTT komunikační protokol

Je potřeba zajistit, aby mezi sebou dokázaly jednotlivé jednotky systému komunikovat. Obě tyto desky disponují připojením k WiFi síti, nabízí se proto připojení právě přes ni prostřednictvím komunikačního protokolu MQTT<sup>18</sup>. Tento protokol je pro své vlastnosti hojně využívaný v IoT<sup>19</sup> aplikacích, mezi něž automatický zavlažovací systém vytvořený v rámci tohoto projektu patří. V následujícím seznamu jsou tyto vlastnosti a další před-

<sup>17</sup><https://dratek.cz/arduino/1271-eses-mini-čerpadlo.html>

<sup>18</sup>název pro komunikační protokol vycházející z akronymu Message Queues Telemetry Transport

<sup>19</sup>Internet of Things - internet věcí



Obrázek 2.7: Ponorné čerpadlo



Obrázek 2.8: MOSFET spínací modul

nosti tohoto protokolu uvedeny (převzato z [3]):

- Protokol je určen pro komunikaci mezi zařízeními (klienty) a je založený na principu publikování zpráv do určitého tématu a přijímání zpráv na základě přihlášení k odběru zpráv daného tématu. Jednotliví účastníci komunikace tedy o sobě navzájem nemusí vědět, zprávu dokážou odeslat, aniž by ji někdo jiný přijal.
- O přeposílání zpráv odběratelům se stará MQTT prostředník (broker). Tento prostředník může být realizovaný fyzicky v lokální WiFi síti, kdy i všichni klienti musí být připojeni na tutéž síť, nebo využitím různých cloudových služeb, kdy tento prostředník je na „cloudu“ a klienti mohou komunikovat mezi sebou, i když jsou připojeni na různé WiFi sítě. Mezi tyto služby patří například i Hivemq<sup>20</sup> využitý v tomto projektu.
- Protokol disponuje třemi úrovněmi QoS<sup>21</sup>, kdy jednotlivé úrovně garantují, že publikovaná zpráva přijde odběrateli nejvýše jednou (QoS = 0), nejméně jednou (QoS = 1), nebo právě jednou (QoS = 2). Pro tento projekt je zcela dostačující nejnižší stupeň QoS = 0.
- MQTT umožňuje perzistentní spojení mezi klientem a prostředníkem. Pokud klient není dostupný, zpráva setrvává u prostředníka, dokud nedojde k opětovnému připojení. Protokol tedy zvyšuje spolehlivost spojení i při komunikaci skrze méně spolehlivá připojení.
- Prostředník dokáže uchovávat poslední zprávu z daného tématu, takže když se klient k odběru tohoto tématu přihlásí, dostane vždy minimálně tuto poslední zprávu.
- Další z vlastností MQTT komunikace je Poslední vůle a závět (LWT<sup>22</sup>), tedy zpráva, kterou rozešle prostředník při ztrátě spojení s klientem. Toto je opět výhodné právě pro zařízení s nespolehlivým připojením.
- V tomto protokolu se dá snadno využít hierarchie témat pro snadnější organizaci a přístup k datům.

---

<sup>20</sup><https://www.hivemq.com/>

<sup>21</sup>Quality of Service - kvalita služby

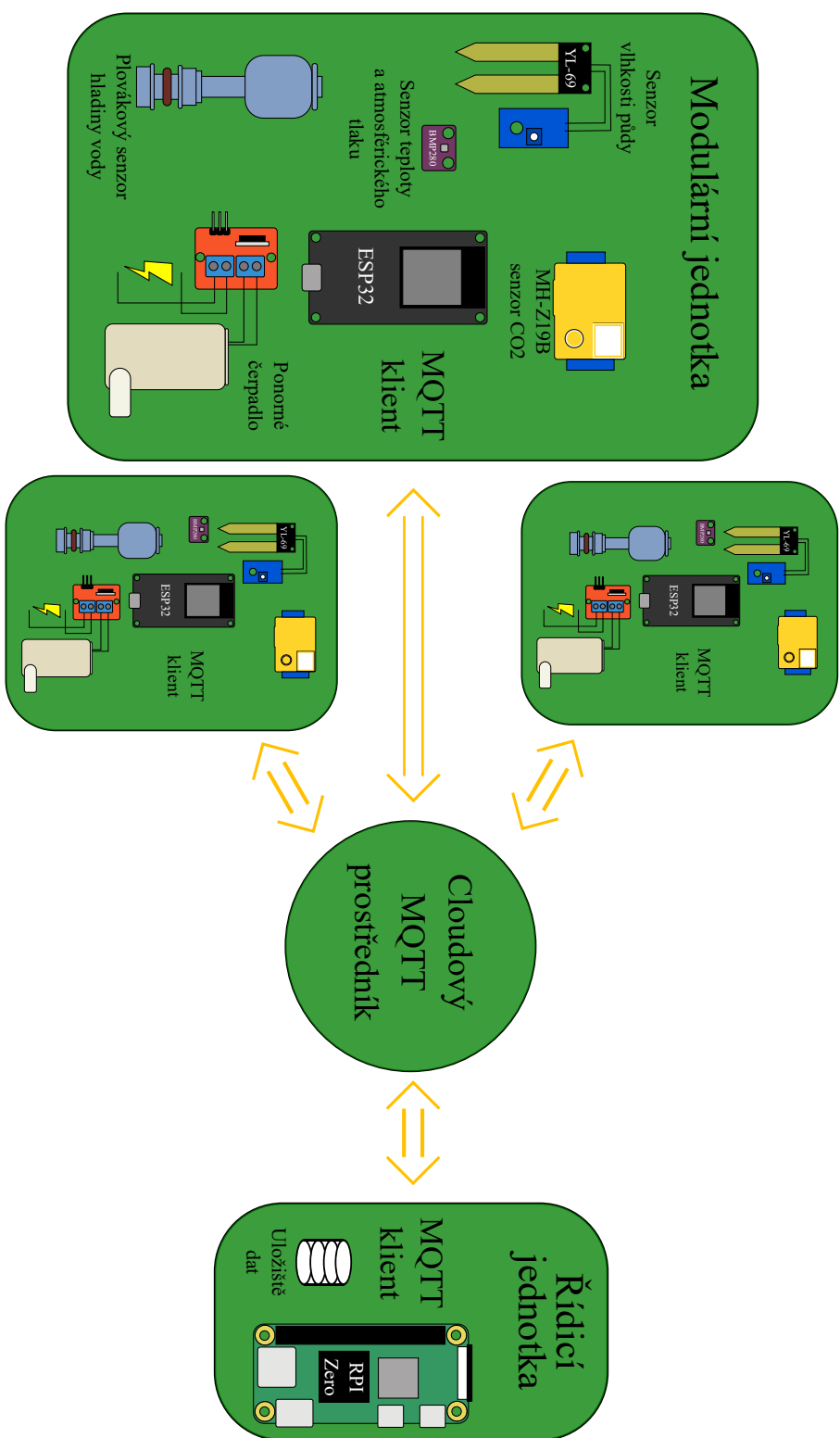
<sup>22</sup>Last Will and Testament - poslední vůle a závět



# 3 Analýza problému

Pro splnění cílů tohoto bakalářského projektu byla navržena architektura, která je patrná z obrázku 3.1. Na základě rešeršní studie byly zvoleny následující prvky:

- **HiveMQ**  
jako cloudový MQTT prostředník,
- **Raspberry Pi Zero**  
jako řídicí jednotka přijímající data ze senzorů pomocí MQTT komunikace, ukládající tato data do své paměti a dále vysílající příkazy jednotlivým modulárním jednotkám opět pomocí MQTT komunikace a
- **vývojová deska ESP32 DevKit V1**  
jako modulární jednotka komunikující skrze MQTT. Na tuto vývojovou desku jsou připojeny následující senzory a akční členy:
  - **senzor vlhkosti půdy YL-38/69,**
  - **senzor teploty vzduchu a atmosférického tlaku BMP280**  
(obě tyto veličiny jsou měřeny pro informaci pro uživatele, v samotné péči o rostliny využity nejsou, mohou sloužit pro případné rozšíření systému, kde by se i tyto sledované veličiny mohly implementovat a dále využít například propojením s termostatem),
  - **senzor koncentrace CO<sub>2</sub> MHZ19-B**  
(stejně jako teplota a atmosférický tlak, tak i hodnota koncentrace oxidu uhličitého ve vzduchu je pouze informativní, dala by se využít v případném nadřazeném systému chytré domácnosti například pro větrání v místnostech, tedy pro zkvalitnění dýchaného vzduchu),
  - **plovákový senzor hladiny vody,**
  - **mini ponorné čerpadlo.**



Obrázek 3.1: Architektura zavlažovacího systému

## 4 Realizace procesu řešení

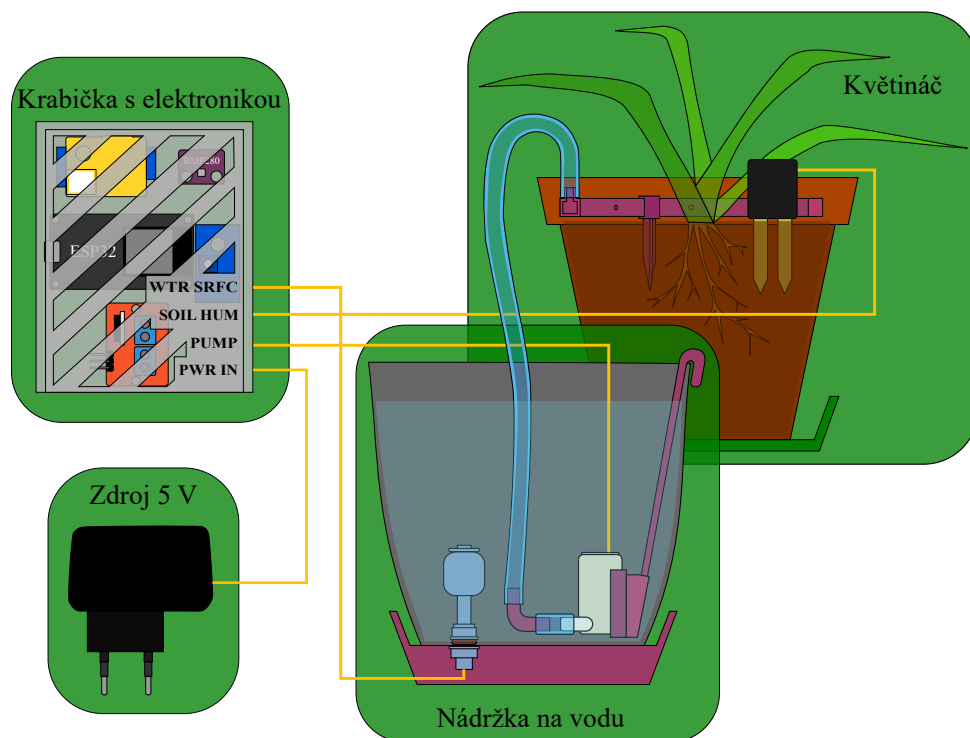
Tato kapitola popisuje návrh a vlastní realizaci zavlažovacího systému, konkrétně realizaci modulární jednotky a její komunikace se řídicí jednotkou, a dále se zabývá praktickým určením množství zavlažované vody na již sestavené modulární jednotce.

### 4.1 Realizace modulární jednotky

Schéma modulární jednotky je patrné z obrázku 4.1. Za hlavní část této jednotky můžeme označit **krabičku s elektronikou** připojenou ke **zdroji napětí 5 V**. Tento zdroj napájí vývojovou desku ESP32. Ta dále napájí a ovládá veškeré senzory a akční členy s výjimkou čerpadla, které je napájeno odděleně přes spínací modul s tranzistorem. Dalším prvkem modulární jednotky je **nádržka na vodu** s plovákovým senzorem hladiny a čerpadlem. Z čerpadla pak vede hadička do zavlažovače v **květináči** s rostlinou. Dále je v květináči v půdě ještě sonda senzoru vlhkosti půdy propojená s elektronikou v krabičce.

Zbývající senzory teploty a atmosférického tlaku a koncentrace CO<sub>2</sub> ve vzduchu jsou spolu se spínacím modulem s tranzistorem, vývojovou deskou a modulem senzoru vlhkosti půdy uloženy v krabičce na elektroniku v blízkosti květináče.

Dále následuje detailní popis realizace jednotlivých částí modulární jednotky mimo již výše [3] zmíněné senzory a čerpadlo.



Obrázek 4.1: Schéma modulární jednotky

### Krabička na elektroniku

Krabička na elektroniku je vymodelována a následně vytisknuta na 3D tiskárně. Model krabičky byl vytvořen s ohledem na

- otvory pro teplotní senzor a senzor koncentrace CO<sub>2</sub> (viz obrázek 4.2), aby tyto senzory mohly správně měřit dané veličiny,
- otvory pro 2,5mm jack zdířky (detailní pohled na stranu krabičky se zdířkami je na obrázku 4.3) pro konektory
  - senzoru hladiny vody („**WTR SRFC**“ - water surface - hladina vody),
  - sondy senzoru vlhkosti půdy („**SOIL HUM**“ - soil humidity - vlhkost půdy),
  - čerpadla („**PUMP**“) a
  - zdroje napětí vývojové desky ESP32 a samostatně čerpadla („**PWR IN**“ - power in)
- a celkově na větrání z důvodu eliminace přehřívání elektroniky (obrázek 4.4).



Obrázek 4.2: Detailní pohled na větrací otvory pro senzory teploty a koncentrace CO<sub>2</sub>

Uspořádání elektroniky (senzorů a dalších modulů) lze vidět na obrázku 4.5.

### Nádržka na vodu

Problémem při realizaci nádržky na vodu bylo, že pro instalaci senzoru hladiny vody (obrázek 4.6) bylo potřeba do dna nádoby vyvrtat pro něj díru. Senzor tak vyčnívá zespodu nádržky, proto bylo potřeba vytvořit pro nádobu podstavec s ohledem na prostor pro vyčnívající senzor. Tento podstavec byl opět na míru vymodelován a vytisknut na 3D tiskárně (viz obrázek 4.7).



Obrázek 4.3: Pohled na krabičku na elektroniku ze strany s jack zdíčkami



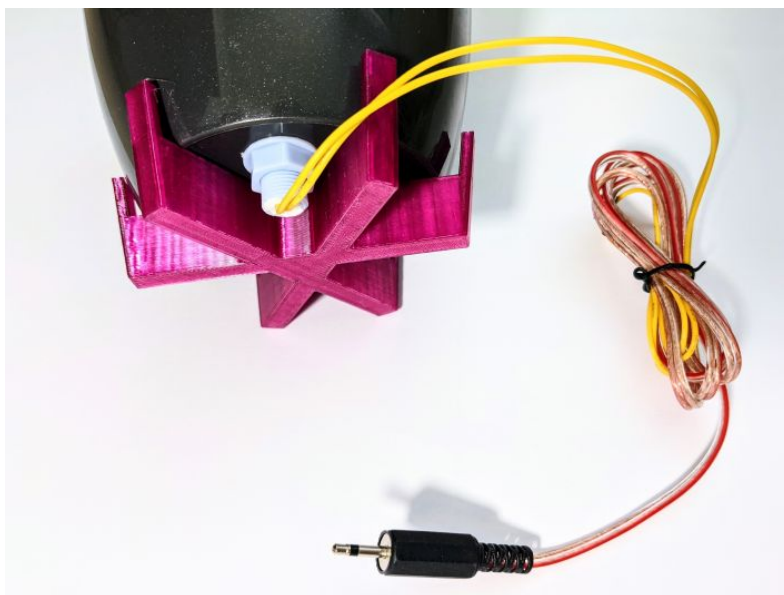
Obrázek 4.4: Krabička na elektroniku



Obrázek 4.5: Pohled na uspořádání elektroniky v krabičce

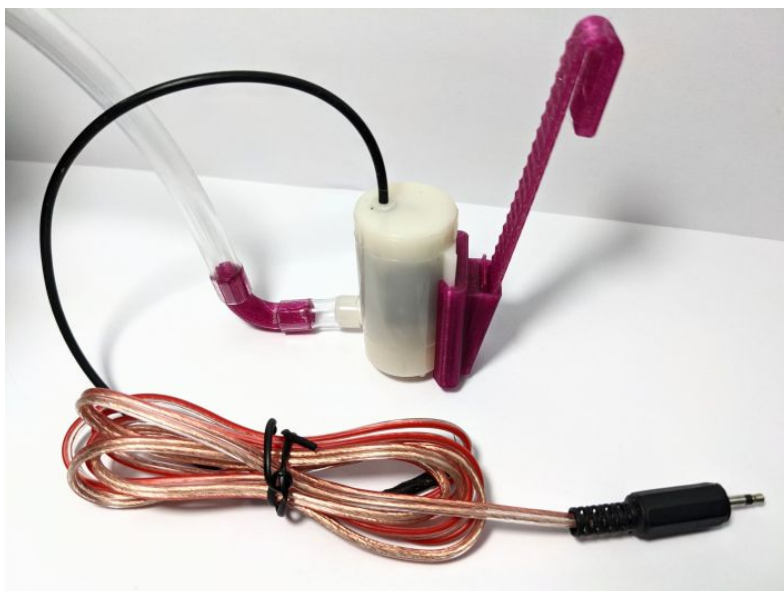


Obrázek 4.6: Nádržka na vodu se senzorem hladiny vody



Obrázek 4.7: Podstavec na nádržku na vodu

Pro instalaci čerpadla na dno nádoby již bylo zvoleno méně invazivní řešení, byl vytvořen (opět pomocí 3D tisku) držák, který se zahákne o okraj nádoby. Pro vyvedení hadičky z nádoby byla ještě vytisknuta pravoúhlá spojka. Ta zajišťuje, aby nedošlo k přiskřípnutí hadičky při ohnutí, tím by došlo k jejímu zaškrcení a zamezení průtoku v ní (viz obrázek 4.8).



Obrázek 4.8: Čerpadlo s držákem a pravoúhlou spojkou

### Zavlažovač

Pro uchycení hadičky v květináči a pro rozptřeni zavlažovacího proudu vody byl vy-

modelovaný a vytisknutý na 3D tiskárně na míru pro daný průměr květináče zavlažovač<sup>1</sup> s bodáky pro jeho uchycení do půdy (obrázek 4.9).



Obrázek 4.9: Zavlažovač s bodáky k upevnění do půdy

### Ochranná krabička elektroniky sondy senzoru vlhkosti půdy

Pro zvýšení voděodolnosti a snížení rezivění, tedy pro zvýšení životnosti sondy senzoru vlhkosti půdy byla vytisknutá pro její vrchní část ochranná krabička<sup>2</sup> (viz obrázek 4.10).

### Indikátor nízké hladiny vody

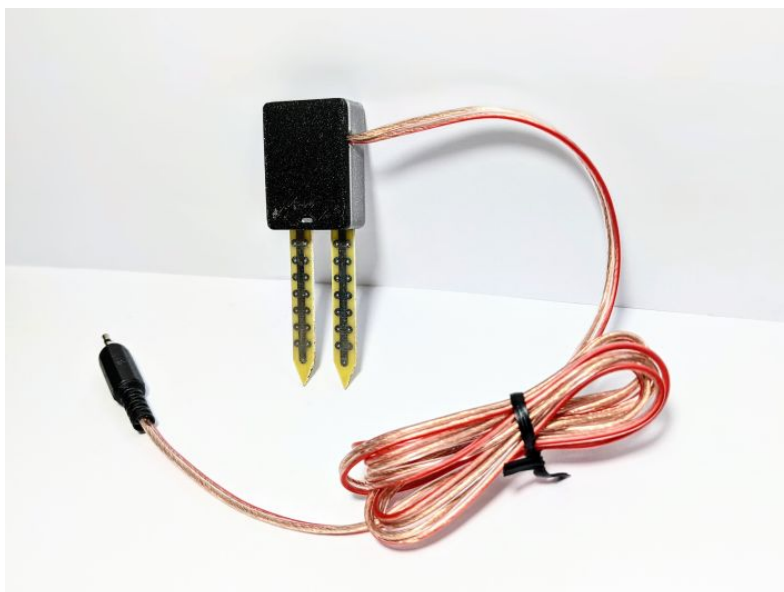
Vestavěná modrá LED dioda na vývojové desce je využita jako indikátor nízké hladiny vody (připojená na pin GPIO2, viz schéma vývodů 2.2). Implementace je taková, aby se se změnou stavu plovákového snímače hladiny vody spustil interrupt (přerušovač), který vyhodnotí, v jakém stavu se snímač nachází, a podle toho zapne nebo vypne diodu (viz výpis 4.1). Tento způsob je namísto pouhé změny stavu LED diody volen proto, že při připojení modulární jednotky ke zdroji napětí by vývojová deska neměla jak zjistit, v jakém stavu se snímač nachází. Detailní pohled na vývojovou desku s rozsvícenou diodou je na obrázku 4.11.

Realizovaná modulární jednotka instalovaná ke květináči je pak na obrázku 4.12.

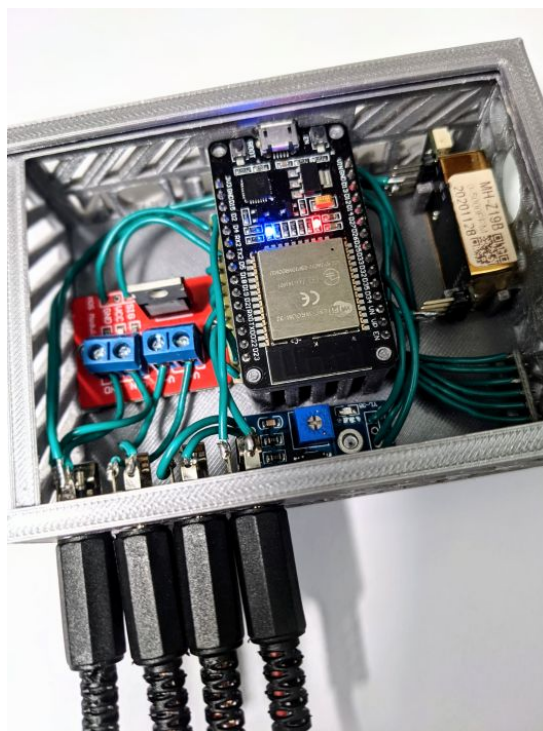
<sup>1</sup>inspirace na <https://www.printables.com/model/128674-sprinkler-for-plants-with-nozzle-for-hose/files>

<sup>2</sup>převzato z <https://www.printables.com/model/38961-moisture-sensor-cover-non-capacitance/files>





Obrázek 4.10: Ochranná krabička elektroniky sondy snímače vlhkosti půdy



Obrázek 4.11: Detailní pohled na modrou indikační LED diodu

```
1 void setup() {  
2   attachInterrupt(digitalPinToInterrupt(digitalPinWtrSrfc),  
3   wtrSrfcControl, CHANGE);  
4 }  
5  
6 void wtrSrfcControl() {  
7   if (digitalRead(digitalPinWtrSrfc)==1){  
8     digitalWrite(digitalPinBlueLED, 1);  
9   } else {  
10    digitalWrite(digitalPinBlueLED, 0);  
11  }  
12 }
```

Výpis kódu 4.1: Implementace indikační diody pro nízkou hladinu vody



Obrázek 4.12: Modulární jednotka

```

1 @scheduler.scheduled_job("cron", hour="*")
2 def every_sharp_hour_job():
3     client.publish("myRoom/flower1/getData", payload="temp", qos=0)
4     client.publish("myRoom/flower1/getData", payload="soilHum", qos=0)
5     client.publish("myRoom/flower1/getData", payload="press", qos=0)
6     client.publish("myRoom/flower1/getData", payload="co2", qos=0)
7
8 @scheduler.scheduled_job("cron", hour="18", minute="0", second="5")
9 def once_a_day_job():
10    client.publish("myRoom/flower1/run", payload="pump", qos=0)

```

Výpis kódu 4.2: Příklad příkazů vydaných řídicí jednotkou

## 4.2 Komunikace mezi jednotkami

Komunikace a řízení systému jsou realizovány pomocí komunikačního protokolu MQTT popsaném v kapitole s rešeršní studií [2.3].

Příklad příkazů publikovaných z řídicí jednotky (Raspberry Pi Zero) je ve výpise 4.2. Každou hodinu jsou publikovány příkazy pro získání dat ze senzorů modulární jednotky (řádky 1-6) a dále pak jednou za den příkaz pro vyhodnocení, zda je třeba zavlažovat, respektive tedy příkaz pro případné spuštění čerpadla (řádky 8-10). Tento příkaz je naplánován na 18:00:05, tj. s pětisekundovým zpožděním po 18:00, kdy probíhá čtení hodnot ze senzorů, tedy aby se eliminovaly případné kolize a byla zaručena funkčnost systému. Analogicky by příkazy vypadaly při zapojení více modulárních jednotek (nahrazením „flower1“ za „flower2“, „flower3“, atd.). Pro funkce spouštěné v daný čas (`every_sharp_hour_job` a `once_a_day_job`) byla využita knihovna `APScheduler`, konkrétně z ní `scheduled_job` dekorátor<sup>3</sup>.

Příklad přijetí příkazu modulární jednotkou ke změření teploty a odeslání hodnoty zpět na řídicí jednotku a spuštění čerpadla je ve výpise 4.3.

## 4.3 Realizace čerpání vody

Tato sekce se dostává již k samotnému zavlažování. Pro nastavení množství vody potřebné pro zalití rostliny bylo provedeno měření závislosti množství přečerpané vody na čase čerpání. Pro každý čas čerpání bylo čerpadlo spuštěno třikrát a získané hodnoty byly zprůměrovány na jedno přečerpání pro zvýšení přesnosti měření. Experiment byl proveden pomocí sestavy na obrázku 4.13. Naměřená data byla zapsána do tabulky 4.1.

Data z tabulky 4.1 byla dále vynesena do grafu 4.14 a následně proložena přímkou. Rovnice této přímky byla upravena tak, aby znázornila závislost času čerpání  $t$  na požadovaném objemu vody  $V$  (rovnice (4.1)). V této rovnici se dosazuje objem  $V$  v jednotkách  $\text{cm}^3$  a získává se čas  $t$  v sekundách.

$$t = 0,1082V + 0,3268 \quad (4.1)$$

Lze si všimnout, že přímka v grafu 4.14 protíná osu  $y$  v záporném bodě -3,0202, což

<sup>3</sup>viz oficiální dokumentaci: <https://apscheduler.readthedocs.io/en/3.x/modules/triggers/cron.html>

```
1 void callback(char* topic, byte* payload, unsigned int length) {
2   String incomingMsg = "";
3   for (int i = 0; i < length; i++) incomingMsg += (char)payload[i];
4   if (strcmp(topic, "myRoom/flower1/getData") == 0) {
5     if (incomingMsg.equals("temp")) {
6       sprintf(msg, 50, "%.2f", bmp.readTemperature());
7       client.publish("myRoom/flower3/temperature", msg);
8     }
9   } else if (strcmp(topic, "myRoom/flower1/run") == 0) {
10    if (incomingMsg.equals("pump")) {
11      waterSurface = digitalRead(digitalPinWtrSrffc);
12      readSoilHumidity(&mySoilHumidity);
13      if (waterSurface == 0 & myMoisture.analog <=30){
14        client.publish("myRoom/flower1/pump", "start pump");
15        analogWrite(PWMPinPump, 200);
16        delay(8000);
17        analogWrite(PWMPinPump, 0);
18        client.publish("myRoom/flower1/pump", "stop pump");
19      } else {
20        client.publish("myRoom/flower1/pump",
21          "not necessary to pump or not enough water");
22      }
23    }
24  }
25 }
```

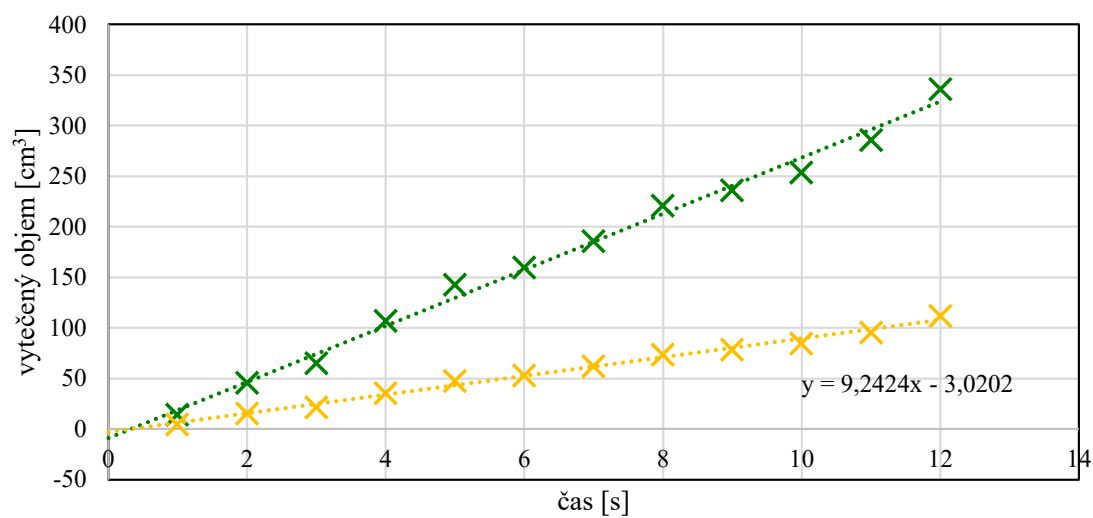
Výpis kódu 4.3: Příklad příkazů přijatých modulární jednotkou



Obrázek 4.13: Sestava pro určení objemu vytečené vody v závislosti na čase čerpání

Tabulka 4.1: Naměřené hodnoty přečerpané vody v závislosti na čase čerpání

čas [s]	objem za tři čerpání [cm <sup>3</sup> ]	objem za jedno čerpání [cm <sup>3</sup> ]
1	14	4,67
2	46	15,33
3	65	21,67
4	107	35,67
5	143	47,67
6	160	53,33
7	186	62,00
8	221	73,67
9	236	78,67



× vytečený objem vody za tři čerpání    × vytečený objem vody zprůměrovaný na jedno čerpání

Obrázek 4.14: Graf závislosti přečerpané vody na čase čerpání

odpovídá faktu, že při určování vytečeného objemu vody je třeba počítat s objemem hadičky a zavlažovače, kterými musí voda protéct dříve, než se dostane k rostlině.

Čerpadlo je pomocí spínacího modulu s tranzistorem IRF520 připojeno samostatně ke zdroji napětí a je řízeno pomocí signálu PWM, díky němuž může být snížena výtoková rychlost tak, aby se voda stačila vsakovat do půdy a nepřetékala přes okraj květináče.

Implementace na vývojové desce ESP32 je pomocí kódu ve výpise 4.3 na řádcích 15-17.

Na základě získané závislosti vytečeného objemu na čase čerpání (rovnice (4.1)) může být dle požadavků specifických pro každou pěstovanou rostlinu (například druh či velikost) určeno, jakým množstvím vody bude rostlina zavlažena. Pro případ testování byla zvolena rostlina rýmovník citronový, pro kterou byl zvolen čas čerpání 8 sekund (respektive 8000 milisekund - viz 16. řádek ve výpise 4.3), tedy objem  $V = 9,2424 \cdot 8 - 3,0202 = 70,919 \text{ cm}^3$ .

## 4.4 Použité knihovny

V tomto projektu jsou využity různé knihovny, a to jak Python knihovny, tak C/C++ knihovny. V této sekci jsou všechny jednotlivé knihovny vypsány:

- pro implementaci komunikačního protokolu MQTT: `paho-mqtt`<sup>4</sup>, `PubSubClient.h`<sup>5</sup>,
- pro vydání příkazu v určitý čas: `APScheduler`<sup>6</sup>,
- pro sběr dat: `logging`<sup>7</sup>,
- pro senzor teploty a atmosférického tlaku BMP280: `Adafruit_BMP280.h`<sup>8</sup>,
- pro implementaci senzorů komunikujících přes I<sup>2</sup>C rozhraní: `Wire.h`<sup>9</sup> a
- pro implementaci CO<sub>2</sub> senzoru: `MHZ19.h`<sup>10</sup>.

---

<sup>4</sup><https://github.com/eclipse/paho.mqtt.python>

<sup>5</sup><https://github.com/knolleary/pubsubclient>

<sup>6</sup><https://github.com/agronholm/apscheduler/tree/3.x>

<sup>7</sup>standardní knihovna jazyka Python

<sup>8</sup>[https://github.com/adafruit/Adafruit\\_BMP280\\_Library](https://github.com/adafruit/Adafruit_BMP280_Library)

<sup>9</sup><https://github.com/arduino/ArduinoCore-avr/blob/master/libraries/Wire/src/Wire.h>

<sup>10</sup><https://github.com/WifWaf/MH-Z19>

# 5 Ověření funkcionality systému

Závěrečná kapitola se věnuje ověření funkčnosti automatického zavlažovacího systému a popisuje, co bylo třeba pro toto ověření udělat.

## 5.1 Sběr a zápis dat

Pro ověření funkčnosti senzorů a celkově celého systému a také pro možné budoucí vylepšení systému bylo třeba implementovat sběr a zápis dat z jednotlivých senzorů. Data byla měřena automaticky každou celou hodinu, pro specifické děje (např. při vyvětrání místnosti, respektive tedy při snížení koncentrace CO<sub>2</sub>, nebo při zavlažení květiny, tedy při skokové změně vlhkosti půdy) byla data měřena s vyšší frekvencí právě pro posouzení správného fungování daných senzorů.

Způsob získávání dat byl nastíněn již v sekci [4.2] a pro zápis dat na řídicí jednotce byla využita standardní knihovna jazyka Python `logging`.

## 5.2 Testování senzorů

Po dobu necelých dvou měsíců byla měřena a zapisována data z jednotlivých senzorů automatického zavlažovacího systému. Následně byla vynesena do grafů v závislosti na čase (obrázek 5.1). Z těchto grafů lze posoudit funkčnost a vhodnost vybraných senzorů.

### Senzor atmosférického tlaku

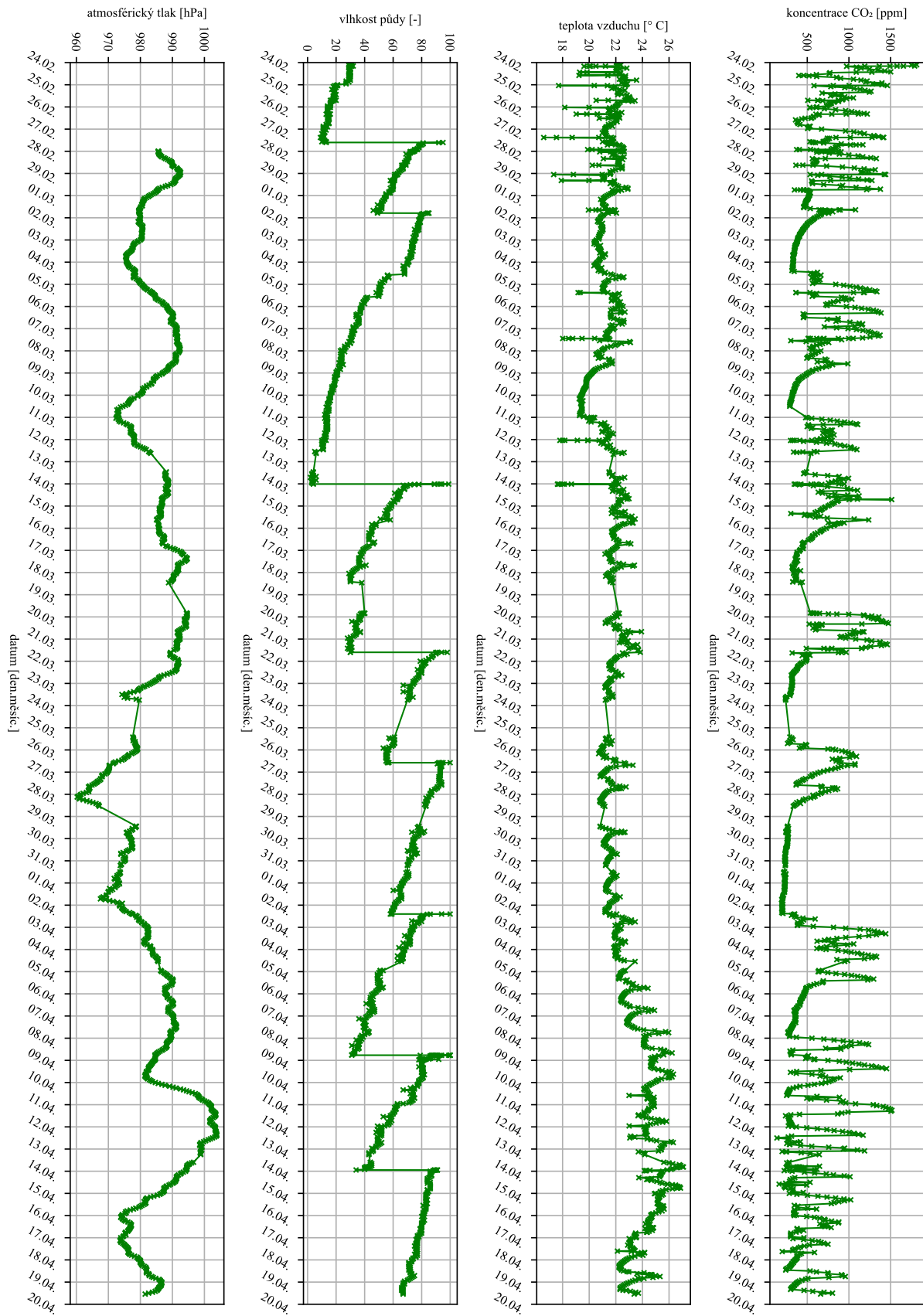
Hodnoty atmosférického tlaku byly vyneseny do grafu pro kratší období (obrázek 5.2) a porovnány s hodnotami atmosférického tlaku měřenými na nejbližší meteorologické stanici Brno-Tuřany (dostupné z portálu ČHMÚ) (obrázek 5.3). Lze vidět, že křivky z obou grafů jsou téměř totožné a senzor tedy je možné označit za funkční.

### Senzory koncentrace CO<sub>2</sub> a teploty

Pro posouzení funkčnosti senzorů koncentrace CO<sub>2</sub> a teploty byly vyneseny do grafu (obrázek 5.4) přes sebe hodnoty ze senzorů pro kratší časový úsek.

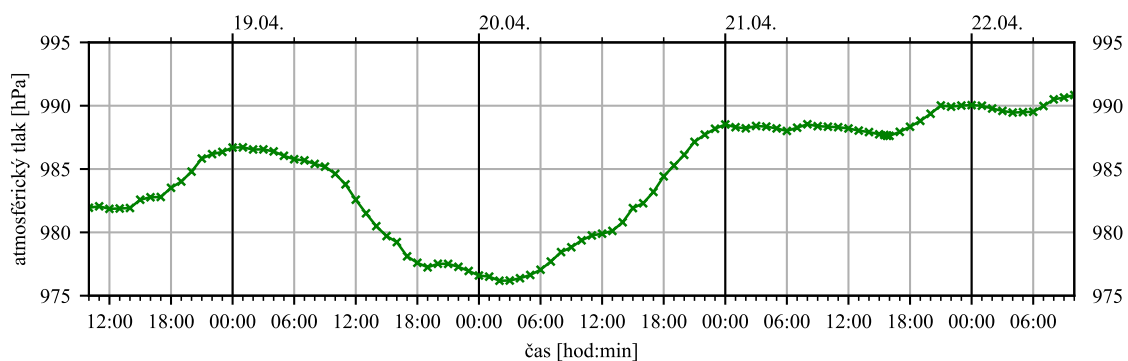
Lze si všimnout, že když bylo v místnosti s modulární jednotkou snímající a zapisující data ze senzorů vyvětráno, přičemž pro tento moment byla krátkodobě snížena perioda snímání hodnot přibližně na jednu minutu, současně s poklesem koncentrace CO<sub>2</sub> ve vzduchu klesla i teplota vzduchu téměř skokově až o 4 °C. Pro porovnání bylo vybráno období od 25.2. po 3.3., kdy se venkovní teploty pohybovaly o několik jednotek níže než ve vytápěné místnosti, proto je rozdíl tak patrný.

Dále lze pozorovat rozdíl mezi časovými úseky, kdy byla místnost obývaná (koncentrace CO<sub>2</sub> má rostoucí tendenci, teplota vzduchu kolísá dle nastavení vytápění) a kdy ne (např. v období 2.3. až 4.3. koncentrace CO<sub>2</sub> kvůli netěsnící izolaci v oknech postupně klesala a teplota vzduchu byla pouze mírně ovlivněna venkovní teplotou, kolísala přibližně o 0,5 °C

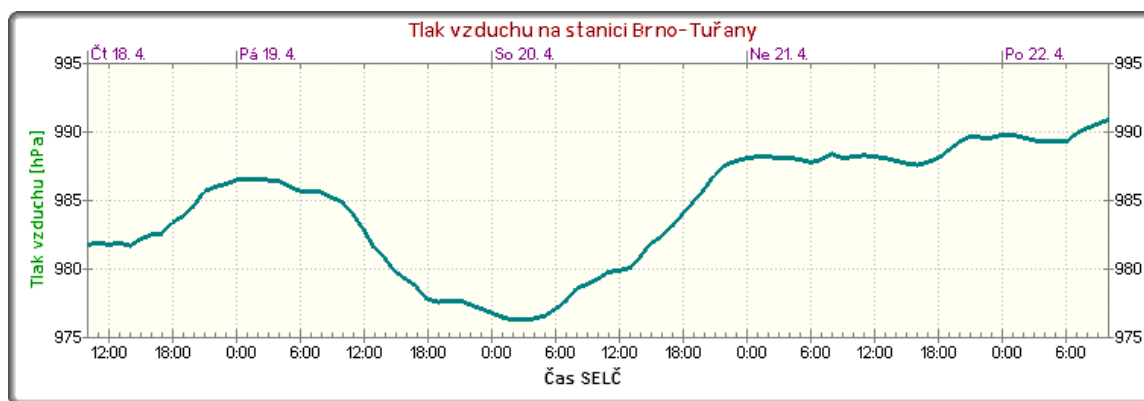


Obrázek 5.1: Data ze senzorů zavlažovacího systému v období od 24.02. do 19.04.

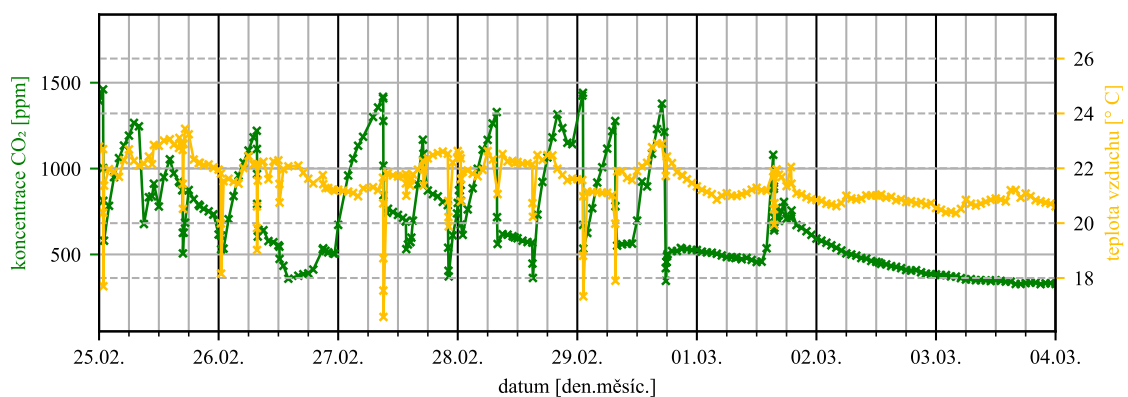




Obrázek 5.2: Data ze senzoru atmosférického tlaku v zavlažovacím systému v období od 18. do 22.4.



Obrázek 5.3: Data ze senzoru atmosférického tlaku na meteorologické stanici Brno-Tuřany v období od 18. do 22.4. Převzato 22.4.2024 z [<https://www.chmi.cz/aktualni-situace/aktualni-stav-pocasi/ceska-republika/stanice/profesionalni-stanice/prehled-stanic/brno-turany>]



Obrázek 5.4: Hodnoty ze senzorů koncentrace CO<sub>2</sub> a teploty vzduchu od 25.2. po 3.3.2024

v noci a ve dne).

### **Senzor vlhkosti půdy**

Na základě sledování hodnot ze senzoru vlhkosti půdy byla z grafu stanovena pro rostlinu, na které byla modulární jednotka testována, stanovena hranice vlhkosti půdy, při které má dojít ke spuštění čerpadla. Jedná se o hodnotu 30 (na škále 0 až 100), kdy tato veličina reprezentuje, jak již bylo v [2.2.3] popsáno, pouze orientační, ale pro tento projekt dostačující posouzení, že půda je suchá a potřebuje zavlažit.

## 6 Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat zavlažovací systém pro pokojové rostliny s požadavkem modularity systému, tedy aby bylo možné pěstovat rostliny každou zvlášť a nezávisle na ostatních.

Součástí práce bylo navrhnout vhodnou architekturu tohoto systému. Bylo navrženo takové řešení, které umožňuje, aby modulární jednotka přijímala jednoduché příkazy od řídicí jednotky jak pro sběr a zápis dat, tak pro vyhodnocení čerpání vody rostlině a jeho případné následné spuštění. Pro komunikaci mezi řídicí jednotkou a jednotlivými modulárními jednotkami byl zvolen protokol MQTT, který je se svými vlastnostmi pro daný účel více než dostačující. Mezi tyto vlastnosti patří například, že je vhodný pro systémy, které obsahují spoustu jednotek, které o sobě navzájem nutně nemusí vědět, nebo že je vhodný pro systémy s méně spolehlivým připojením k internetu.

Dále bylo třeba vybrat vhodné senzory a akční členy. Jednotlivé prvky byly voleny s přihlédnutím na dostupnost, jednoduchost při splnění požadavků pro projekt zavlažovacího systému a kromě senzoru koncentrace  $\text{CO}_2$  také na nízkou cenu. Vysoká cena senzoru koncentrace  $\text{CO}_2$  (v době, kdy byl projekt vypracováván, stál 999 Kč) by se dala vyřešit například vytvořením různých druhů modulárních jednotek, které by tento prvek nemusely obsahovat, přičemž jednotka s tímto senzorem by byla umístěna v každé místnosti jen jednou.

V rámci sestavení celého systému bylo třeba kromě zapojení jednotlivých komponent vymyslet i jejich vhodné umístění. Za tímto účelem byla vytvořena krabička s ohledem jak na funkčnost (s přihlédnutím na otvory pro větrání, aby nedocházelo k přehřátí elektroniky, otvory pro zdířky pro připojení konektorů senzoru hladiny, sondy vlhkosti půdy, čerpadla a zdroje a dále také na celkovou kompaktnost), tak i na vzhled. Tato krabička byla vymodelována na míru a následně vytisknuta na 3D tiskárně, stejně tak i ostatní komponenty, jako jsou podstavec na nádržku na vodu, držák na čerpadlo, pravoúhlá spojka pro vyvedení hadičky od čerpadla zespodu nádržky a pak i zavlažovač v květináči.

Funkcionalita systému byla ověřena na základě necelých dvou měsíců měření, zápisu a následného vyhodnocení veškerých hodnot ze senzorů systému.

Během práce na tomto projektu jsem se seznámil s mnoha novými informacemi z oblastí elektroniky, programování, modelování a zpracování dat. Jejich vhodné propojení umožňuje vytvářet komplexní systémy, které by pouze v rámci jednotlivých samostatných oborů vytvořit nebylo možné. Toto považuji za značný přínos pro mé další studium a pro profesní růst.

# Seznam literatury

- [1] CEZ.CZ. *Koncentrace oxidu uhličitého může být doma nebezpečná. Hodí se měřič CO<sub>2</sub>* online. Leden 2024. Dostupné z: <https://www.cez.cz/cs/clanky/plyn/koncentrace-oxidu-uhliciteho-muze-byt-doma-nebezpecna.-hodi-se-meric-co2-188858>. [cit. 2024-05-11].
- [2] DAS, D. *How Does the BMP280 Digital Pressure Sensor Work and how to Interface it with Arduino?* online. Duben 2022. Dostupné z: <https://circuitdigest.com/microcontroller-projects/interfacing-bmp280-sensor-with-arduino>. [cit. 2024-04-27].
- [3] HIVEMQ TEAM. *Introducing the MQTT Protocol – MQTT Essentials: Part 1* online. Leden 2015. Aktualizováno Únor 2024. Dostupné z: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/#heading-introducing-the-mqtt-protocol-mqtt-essentials-part-1>. [cit. 2024-05-11].
- [4] M., LUBOŠ. *Plovákový senzor vodní hladiny* online. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/plovakovy-senzor-vodni-hladiny.html>. [cit. 2024-05-11].
- [5] MOHANAN, V. *DOIT ESP32 DevKit V1 Wi-Fi Development Board – Pinout Diagram & Arduino Reference*. online. Prosinec 2022. Dostupné z: <https://circuitstate.com/esp32dkpinref>. [cit. 2024-04-25].
- [6] PARMAR, C. *Know Your Sensor — YL38 Soil Hygrometer* online. Únor 2019. Dostupné z: <https://medium.com/@chirag.parmar/know-your-sensor-yl38-soil-hygrometer-fceca860faac>. [cit. 2024-05-22].
- [7] RASPBERRYPI.COM. *Raspberry Pi Zero* online. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-zero/>. [cit. 2024-05-10].
- [8] WIKIPEDIA CONTRIBUTORS. *Nondispersive infrared sensor — Wikipedia, The Free Encyclopedia* online. 2024. Dostupné z: <https://en.wikipedia.org/w/index.php?title=ESP32&oldid=1220549564>. [cit. 2024-04-26].
- [9] WIKIPEDIA CONTRIBUTORS. *Nondispersive infrared sensor — Wikipedia, The Free Encyclopedia* online. 2024. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Nondispersive\\_infrared\\_sensor&oldid=1220696133](https://en.wikipedia.org/w/index.php?title=Nondispersive_infrared_sensor&oldid=1220696133). [cit. 2024-05-11].
- [10] WIKIPEDIA CONTRIBUTORS. *Raspberry Pi — Wikipedia, The Free Encyclopedia* online. 2024. Dostupné z:

[https://en.wikipedia.org/w/index.php?title=Raspberry\\_Pi&oldid=1218999355](https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=1218999355).  
[cit. 2024-04-26].