

# Závěrečná studijní práce

## dokumentace

### Chytrý Květináč

#### David Balcar



**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování

**Třída:** IT4

**Školní rok:** 2024/2025

### ***Poděkování***

*Rád bych vyjádřil upřímné poděkování panu Mgr. Marcelu Godovskému za zapůjčení hardwaru a cenné rady, které výrazně přispěly k realizaci tohoto projektu. Zároveň děkuji panu Ing. Petru Grussmannovi za jeho pomoc a odborné konzultace v oblasti vybraných technologií.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl všecky použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední Průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě                    31.12.2024

---

*podpis autora práce*

## **Anotace:**

Tento projekt se zaměřuje na vývoj systému pro automatické zavlažování květiny s cílem optimalizovat péči o rostliny a usnadnit jejich správu. Hlavním cílem projektu je vytvoření systému, který monitoruje vlhkost půdy, hladinu vody v zásobníku a vlhkost vzduchu. Na základě těchto údajů je automaticky řízeno zavlažování květiny. Systém zobrazuje statistiky z různých senzorů, což umožňuje uživateli mít přehled o stavu rostliny a jejího prostředí.

Logika automatizace je postavena na několika klíčových podmínkách: pokud je vlhkost půdy nízká, ale zároveň je dostatečná hladina vody v zásobníku a vlhkost vzduchu není vysoká, systém aktivuje čerpadlo pro zavlažování. Tento mechanismus byl navržen s ohledem na prevenci dvou problémů – čerpadlo neběží na prázdro, pokud není dostatek vody v zásobníku, a zároveň se zamezí nadmerné vlhkosti vzduchu, která by mohla vést k tvorbě plísni. Tento systém tak zajišťuje efektivní a bezpečné zavlažování květiny.

**Klíčová slova:** Automatické zavlažování, čerpadlo, statistiky senzorů, prevence plísni.

# Obsah

<b>Úvod</b>	<b>4</b>
<b>1 Cíl projektu</b>	<b>5</b>
<b>2 Použité technologie</b>	<b>5</b>
2.1 <i>ESP8266 (Wemos D1 Mini)</i> . . . . .	5
2.2 <i>LittleFS</i> . . . . .	6
2.3 <i>ESPAAsyncWebServer</i> . . . . .	6
2.4 <i>MQTT</i> . . . . .	6
2.5 <i>Home Assistant</i> . . . . .	6
2.6 <i>ESPhome</i> . . . . .	6
<b>3 Technický návrh</b>	<b>7</b>
3.1 <i>Model květináče</i> . . . . .	7
3.2 <i>Schéma zapojení</i> . . . . .	8
3.3 <i>Popis funkcionality</i> . . . . .	8
<b>4 Programování</b>	<b>9</b>
4.1 <i>Struktura Kódu</i> . . . . .	9
4.2 <i>Použité knihovny</i> . . . . .	9
<b>5 Ukázka kódu</b>	<b>10</b>
<b>6 Problémy a řešení</b>	<b>12</b>
6.1 <i>Problém s kalibrací senzorů vlhkosti</i> . . . . .	12
6.2 <i>Problém s čerpadlem při nízké hladině vody</i> . . . . .	12
6.3 <i>Problémy s pomalým načítáním dat na webové stránce</i> . . . . .	12
<b>7 Integrace s Home Assistant</b>	<b>13</b>
7.1 <i>Mosquitto broker</i> . . . . .	13
7.2 <i>ESPhome</i> . . . . .	14
<b>8 Závěr</b>	<b>14</b>
<b>9 Splněné a nesplněné cíle</b>	<b>15</b>
9.1 <i>Splněné cíle</i> . . . . .	15
9.2 <i>Nesplněné cíle</i> . . . . .	15
<b>10 Seznam použitých informačních zdrojů</b>	<b>16</b>

# Úvod

*Rostliny jsou nezbytnou součástí našeho života, ať už jde o estetický přínos v domácnosti, nebo o jejich ekologický význam. Pro správný růst a zdraví rostlin je klíčová pravidelná péče, zejména zavlažování. Nicméně, hektický životní styl často brání lidem věnovat rostlinám dostatek času a pozornosti. V mnoha případech vede nesprávné zalévání bud' k přemokření, nebo naopak k vysychání půdy, což má negativní dopad na růst rostlin. Automatizace v této oblasti může výrazně zjednodušit péči o rostliny a zároveň zlepšit jejich podmínky.*

*Cílem mého projektu je vytvořit chytrý květináč, který dokáže automaticky zavlažovat rostliny na základě aktuální vlhkosti půdy a dalších environmentálních parametrů. Tento květináč bude vybaven senzory pro monitorování vlhkosti půdy, hladiny vody v zásobníku a vlhkosti vzduchu. Na základě těchto dat bude automaticky řízeno čerpadlo, které zajišťuje zavlažování. Logika systému je navržena tak, aby čerpadlo spustilo zavlažování pouze tehdy, pokud je vlhkost půdy nízká, voda je v zásobníku dostupná a vlhkost vzduchu není příliš vysoká. Tyto podmínky zajišťují efektivní zavlažování, minimalizují riziko vzniku plísní a zároveň chrání čerpadlo před chodem na sucho.*

*Projekt se nezaměřuje pouze na automatizaci zavlažování, ale i na sběr a analýzu dat ze senzorů. Statistiky, jako je průměrná denní vlhkost nebo teplotní výkyvy, budou ukládány a zobrazovány uživatelům. Tyto informace umožní uživatelům lépe pochopit podmínky, ve kterých jejich rostliny rostou, a případně přizpůsobit nastavení květináče.*

*V této dokumentaci detailně popisují návrh a implementaci systému chytrého květináče. Zaměřuji se na propojení jednotlivých komponent, jako jsou senzory, čerpadlo a řídicí jednotka, a jejich integraci do funkčního celku. V úvodní části se věnuji návrhu hardwarové a softwarové architektury. Dále popisují způsoby zpracování dat, automatizační logiku a možnosti manuálního ovládání. V závěru se věnuji vizuálnímu zpracování a způsobu prezentace statistických údajů.*

# 1 Cíl projektu

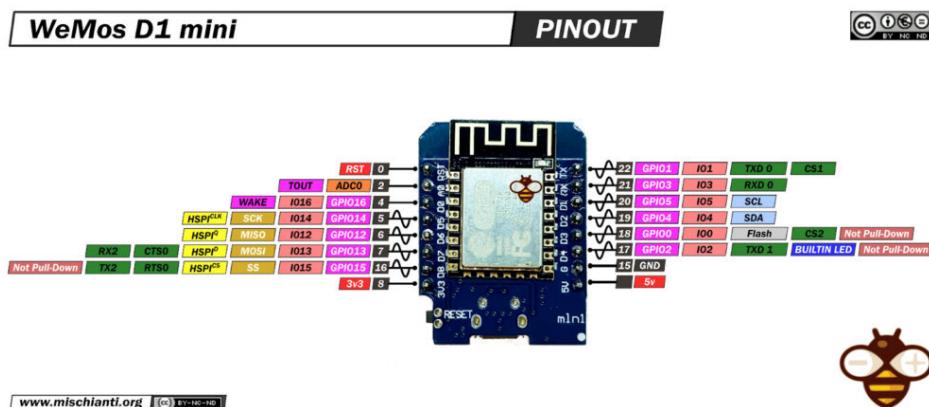
Cílem tohoto projektu bylo navrhnout a implementovat chytrý květináč, který bude automaticky zavlažovat rostliny na základě předem definovaných parametrů a aktuálních podmínek monitorovaných senzory. Systém měl za úkol sledovat různé environmentální faktory, jako je vlhkost půdy, hladina vody v zásobníku a vlhkost vzduchu. Na základě těchto dat bude automaticky řízeno čerpadlo, které aktivuje zavlažování pouze tehdy, když jsou splněny požadované podmínky, jako je nízká vlhkost půdy a dostatečná hladina vody v zásobníku. Tento mechanismus měl zaručit optimální a efektivní zavlažování, přičemž minimalizuje riziko přemokření rostliny nebo zbytečného čerpání vody, když to není potřeba.

Dalším cílem projektu bylo vytvořit webovou stránku, která bude poskytovat přehled o aktuálních hodnotách měřených senzorů. Na stránce budou zobrazeny statistiky, jako jsou průměrné denní hodnoty vlhkosti půdy, teploty a vlhkosti vzduchu, spolu s grafy pro vizualizaci těchto dat. Uživatelé budou mít přístup k historickým datům a statistikám, které jim pomohou lépe pochopit podmínky panující v okolí jejich rostlin a optimalizovat jejich péči. Cílem bylo také zajistit přehlednost a jednoduchost v ovládání systému, což by uživatelům usnadnilo manipulaci s chytrým květináčem.

## 2 Použité technologie

### 2.1 ESP8266 (Wemos D1 Mini)

ESP8266 je cenově dostupná a výkonná WiFi platforma, která umožňuje snadné připojení k síti a dálkové ovládání zařízení. V rámci projektu „Chytrý květináč“ byla tato platforma použita k připojení zařízení k internetu, sběru dat ze senzorů a ovládání zavlažování prostřednictvím webového rozhraní.



Obrázek 1: Piny Wemos D1 Mini

## 2.2 LittleFS

*LittleFS je souborový systém optimalizovaný pro mikrokontroléry, který umožňuje efektivní ukládání a čtení dat z flash paměti. Tento souborový systém byl v předchozích verzích projektu, než byl přechod na Home Assistant, použit k ukládání historických dat senzorů, což umožnilo sledování dlouhodobých změn podmínek v prostředí.*

## 2.3 ESPAsyncWebServer

*ESPAsyncWebServer je knihovna pro Arduino, která umožňuje vytváření asynchronních webových serverů. V předchozích verzích projektu byla použita k vytvoření webové stránky, která zobrazovala aktuální hodnoty senzorů, statistiky a stav zavlažování.*

## 2.4 MQTT

*MQTT je lehký a efektivní protokol pro zasílání zpráv, ideální pro IoT aplikace. V projektu byl použit k zajištění komunikace mezi zařízením a Home Assistantem před přechodem na ESPhome. Tento protokol umožnil spolehlivou výměnu dat mezi zařízeními a centrálním systémem pro monitoring a ovládání chytrého květináče.*

## 2.5 Home Assistant

*Home Assistant je open-source platforma pro automatizaci domácnosti, která umožňuje integraci a správu různých zařízení. Byla použita k centralizaci řízení a sledování systému „Chytrý květináč“ a poskytla uživatelům přehledné rozhraní pro ovládání a monitorování všech souvisejících senzorů a funkcí.*

## 2.6 ESPhome

*ESPhome je nástroj pro snadné vytvoření firmwaru pro zařízení založená na platformách ESP8266 a ESP32. V tomto projektu bylo ESPhome použito k implementaci řízení činností chytrého květináče, včetně integrace s Home Assistantem. ESPhome zjednodušilo konfiguraci a správu zařízení, umožňující snadné připojení k platformě pro automatizaci domácnosti.*

### 3 Technický návrh

#### 3.1 Model květináče

Tento model květináče jsem obdržel od pana Mgr. Marcella Godovského. Květináč je navržen jako modulární systém, kde jednotlivé části slouží specifickému účelu a spolupracují na zajištění optimální péče o rostlinu. Model vychází z návrhu dostupného na stránce <https://www.instructables.com/Build-a-Self-Watering-System-Soil-Moisture-Sensor-/>.

V horní části květináče se nachází Wemos D1 mini s připojeným senzorem DHT11, který měří teplotu a vlhkost vzduchu. Do této části je také integrována hadička sloužící k zavlažování. Voda je čerpána čerpadlem a vedena touto hadičkou přímo na rostlinu. Tato horní část je propojena s prostorem pro rostlinu prostřednictvím dvou nosných sloupků, kterými jsou vedeny hadička a kabeláž.

Pod touto horní částí se nachází prostor pro samotnou rostlinu. K rostlině je připojen senzor vlhkosti půdy, který monitoruje vlhkost substrátu a umožňuje tak přesnou kontrolu zavlažování.

V nejnižší části květináče je odnímatelný zásobník na vodu. Tento zásobník obsahuje čerpadlo, které čerpá vodu hadičkou nahoru k rostlině. V zásobníku se rovněž nachází senzor hladiny vody, který umožňuje sledovat množství dostupné vody a upozorňuje na potřebu jejího doplnění.

Tento model je navržen s ohledem na jednoduchou obsluhu, snadnou údržbu a maximální automatizaci, čímž uživateli usnadňuje péči o rostlinky.



Obrázek 2: Model květináče

## 3.2 Schéma zapojení

Pro realizaci projektu byl zvolen mikroprocesor *ESP8266*, konkrétně model *Wemos D1 mini*, který slouží k řízení celého systému. Systém obsahuje několik senzorů a dalších komponent, které jsou k němu připojeny. Mezi hlavní součásti patří:

- **DHT senzor** pro měření teploty a vlhkosti vzduchu, připojený na *GPIO0*.
- **Senzor hladiny vody** připojený na analogový vstup *A0*.
- **Senzor vlhkosti půdy HW-080** připojený na *GPIO13*.
- **Pumpa** připojená na *GPIO2* pro automatické zavlažování.

Všechny komponenty jsou připojeny k mikroprocesoru a komunikují s ním prostřednictvím digitálních a analogových vstupů a výstupů. Kromě toho je systém napojen na WiFi síť a komunikuje s webovým serverem pro sledování aktuálního stavu.

## 3.3 Popis funkcionality

Projekt je navržen tak, aby automaticky monitoroval vlhkost půdy, teplotu a hladinu vody v květináči. Na základě těchto informací je aktivována pumpa, která zavlažuje rostlinu. Systém umožňuje také manuální ovládání pumpy a sledování dat v reálném čase prostřednictvím webového rozhraní.

Hlavní funkcionality systému:

- **Sledování teploty a vlhkosti vzduchu** pomocí DHT senzoru.
- **Sledování vlhkosti půdy** pomocí digitálního senzoru.
- **Monitorování hladiny vody** v nádrži.
- **Automatické zavlažování** aktivováním pumpy, pokud je půda suchá.
- **Manuální ovládání pumpy** prostřednictvím webového rozhraní.
- **Ukládání dat** pro pozdější analýzu a vykreslování průměrných hodnot.

Všechny tyto funkce jsou realizovány pomocí *ESP8266*, který zajišťuje sběr dat a jejich následné odesílání na *Home Assistant*. Systém je navržen pro bezproblémový běh i v případě výpadků napájení, kdy se data odesílají přímo do *Home Assistant* pro jejich zpracování a analýzu.

## 4 Programování

### 4.1 Struktura Kódu

Kód projektu je strukturován do několika částí:

- **Hlavní část programu (setup a loop):** Obsahuje inicializaci hardwarových součástí, připojení k WiFi, spuštění webového serveru a nekonečnou smyčku pro udržování běhu zařízení.
- **Funkce pro čtení senzorů:** Zajišťují čtení dat z DHT senzoru, senzoru hladiny vody a senzoru vlhkosti půdy.
- **Obsluha webového serveru:** Endpointy webového serveru poskytují přístup k aktuálním datům ze senzorů a umožňují ovládání pumpy.
- **Práce se souborovým systémem (LittleFS):** Slouží k ukládání a načítání souborů, například HTML, CSS a JavaScript.

Tato struktura zajišťuje přehlednost kódu a snadnou rozšířitelnost projektu.

Tento původní kód byl vytvořen jako základní verze projektu, než došlo k migraci na **ESPHome**, což je modernější a efektivnější přístup pro integraci s Home Assistant.

### 4.2 Použité knihovny

V této starší verzi projektu byly využity následující knihovny:

- **ESPAsyncWebServer:** Pro implementaci asynchronního webového serveru umožňujícího efektivní obsluhu více klientů současně.
- **SimpleDHT:** Pro čtení dat z DHT11 senzoru, který měří teplotu a vlhkost vzduchu.
- **LittleFS:** Pro práci se souborovým systémem, kde jsou ukládány HTML, CSS a JavaScript soubory.
- **ESP8266WiFi:** Pro připojení zařízení k WiFi síti.

Tyto knihovny byly zvoleny s ohledem na omezené hardwarové prostředky ESP8266 a jejich kompatibilitu s danými funkcemi.

Aktuálně je však projekt implementován pomocí **ESPHome**, což výrazně zjednoduší konfiguraci a integraci s Home Assistant. ESPHome nahradil ruční implementaci těchto funkcí, čímž se projekt stal snadněji udržovatelným a přehlednějším.

## 5 Ukázka kódu

Níže uvedený kód představuje původní strukturu programu pro *ESP8266*, který sloužil k měření parametrů souvisejících s rostlinou, jako je teplota a vlhkost vzduchu, vlhkost půdy a úroveň vody. Tento kód byl použit před přechodem na *Home Assistant* a *ESPHome*. Obsahuje logiku pro připojení k WiFi síti a načítání a zpracování dat z různých senzorů.

```
void setup() {
    Serial.begin(115200);
    pinMode(WATER_SENSOR_PIN, INPUT);
    pinMode(PUMP_PIN, OUTPUT);
    pinMode(SOIL_SENSOR_PIN, INPUT);
    digitalWrite(PUMP_PIN, LOW);

    Serial.print("Pripojuju se k WiFi... ");
    WiFi.begin(ssid, password);
    int retries = 0;
    while (WiFi.status() != WL_CONNECTED && retries < 20) {
        delay(1000);
        Serial.print(".");
        retries++;
    }
    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("WiFi-pripojeno");
        Serial.print("IP-adresa:-");
        Serial.println(WiFi.localIP());
    } else {
        Serial.println("Nepodarilo-se-pripojit-k-WiFi");
        return;
    }

    if (!LittleFS.begin()) {
        Serial.println("Chyba-pri-inicializaci-LittleFS!");
        return;
    }
    createFilesIfNotExist();
    listFiles();

    server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
        File file = LittleFS.open("/index.html", "r");
        if (file) {
            String response = file.readString();
            file.close();
```

```
        unsigned long currentMillis = millis();
        if (currentMillis - lastDHTRead >= dhtDelay) {
            lastDHTRead = currentMillis;
            byte tempByte = 0;
            byte humByte = 0;
            int err = dht11.read(&tempByte, &humByte, NULL);
            if (err == SimpleDHTErrSuccess) {
                temperature = tempByte;
                humidity = humByte;
            } else {
                temperature = 0;
                humidity = 0;
                Serial.println("Chyba - pri - cteni - z - DHT - senzoru !");
            }
        }

        int waterLevel = analogRead(WATER_SENSOR_PIN);
        soilMoisture = digitalRead(SOIL_SENSOR_PIN);

        response.replace("{{ temperature }}", String(temperature));
        response.replace("{{ humidity }}", String(humidity));
        response.replace("{{ waterLevel }}", String(waterLevel));
        response.replace("{{ soilMoisture }}", String(soilMoisture ? "Suc

        request->send(200, "text/html", response);
    } else {
        request->send(404, "text/plain", "Soubor - nenalezen");
    }
});

server.begin();
}
```

- **Setup funkce:** V této části kódu se provádí inicializace sériové komunikace, nastavení pinů pro senzory a pumpu, připojení k WiFi a LittleFS, a vytvoření souborů pro ukládání dat.
- **Web server:** Tento kód obsahuje také jednoduchý webový server, který sloužil k zobrazení aktuálních hodnot teploty, vlhkosti vzduchu, úrovně vody a vlhkosti půdy na webové stránce. Data byla pravidelně načítána ze senzorů a zobrazena v HTML šabloně.

## 6 Problémy a řešení

Během vývoje systému chytrého květináče se vyskytlo několik problémů, které bylo potřeba vyřešit, aby systém fungoval spolehlivě. V této sekci popíšu některé z hlavních problémů, které nastaly, a postupy, jakými byly vyřešeny.

### 6.1 Problém s kalibrací senzorů vlhkosti

Jedním z prvních problémů bylo dosažení správné kalibrace senzoru vlhkosti půdy. Původní hodnoty, které senzor vrácel, nebyly přesné a neodpovídaly skutečnému stavu vlhkosti. Tento problém jsem vyřešil několika způsoby:

- Ověřil jsem kalibraci senzoru pomocí externího měření vlhkosti, abych měl referenční hodnoty pro úpravu.
- Nakonec jsem implementoval možnost ručního nastavení citlivosti senzoru, což umožnilo uživateli přizpůsobit systém různým typům půdy.

### 6.2 Problém s čerpadlem při nízké hladině vody

Dalším problémem, který nastal, byla situace, kdy čerpadlo mělo běžet, ale v zásobníku vody bylo velmi málo tekutiny. Tento problém mohl vést k poškození čerpadla. Řešení spočívalo v implementaci detekce nízké hladiny vody:

- Implementoval jsem senzor hladiny vody, který měří výšku vody v nádrži.
- Software byl upraven tak, aby čerpadlo spustil pouze tehdy, pokud hladina vody byla dostatečně vysoká. Pokud byla hladina vody příliš nízká, čerpadlo se neaktivovalo.

### 6.3 Problémy s pomalým načítáním dat na webové stránce

Na začátku testování jsem se setkal s problémem pomalého načítání dat na webové stránce, což zdržovalo interakci uživatele se systémem. Tento problém jsem vyřešil implementací asynchronní knihovny pro načítání dat, která zajistila plynulé načítání bez zbytečného zpoždění. Později jsem přešel na integraci s platformou Home Assistant, což umožnilo lepší správu a monitorování systému, čímž se výrazně zlepšil výkon a uživatelská zkušenosť.

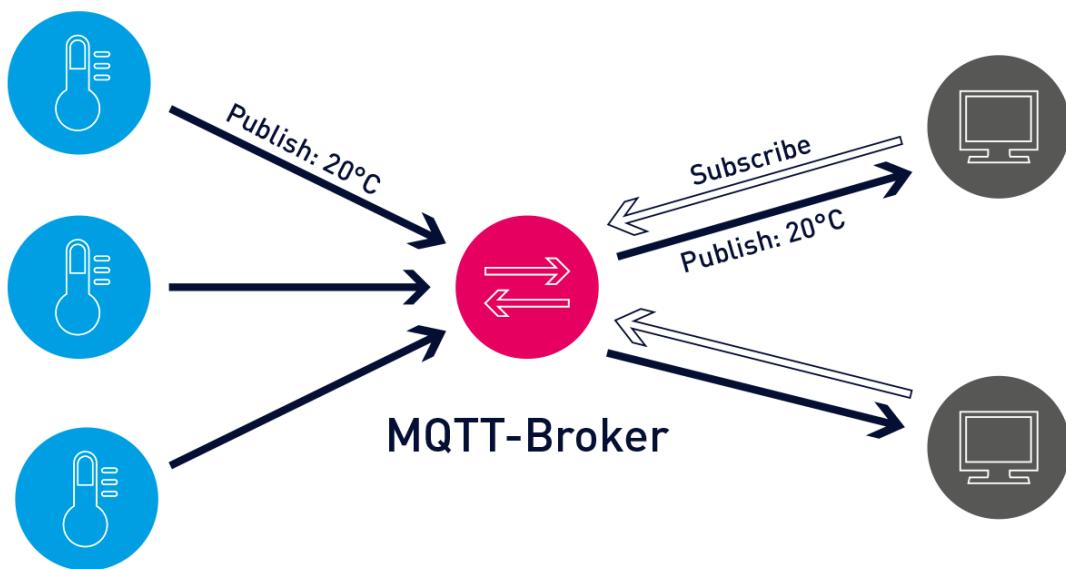
## 7 Integrace s Home Assistant

Systém byl úspěšně integrován s platformou Home Assistant, což umožňuje snadné monitorování a ovládání chytrého květináče. Tato integrace výrazně zjednodušila práci s daty ze senzorů a ovládání pump. Home Assistant poskytuje centralizovanou platformu pro sledování a řízení všech zařízení v domácnosti, což umožňuje uživateli pohodlně spravovat a automatizovat různé funkce, jako je zalévání rostliny, v závislosti na podmínkách měřených senzory. Díky podpoře MQTT a ESPHome je integrace s touto platformou rychlá a efektivní.

### 7.1 Mosquitto broker

Mosquitto broker byl původně nasazen jako zprostředkovatel pro komunikaci mezi zařízeními pomocí protokolu MQTT. Tento broker umožňuje efektivní a spolehlivý přenos dat mezi chytrým květináčem a Home Assistantem. V projektu byl Mosquitto broker klíčový pro implementaci asynchronního přenosu dat. Později byl však nahrazen ESPHome, který poskytl lepší integraci a jednodušší správu zařízení.

**Jak funguje Mosquitto broker:** Mosquitto broker je software, který zprostředkovává komunikaci mezi různými zařízeními v IoT (Internet of Things) sítích prostřednictvím protokolu MQTT (Message Queuing Telemetry Transport). Tento protokol umožňuje zařízení zasílat a přijímat zprávy bez nutnosti přímé komunikace mezi sebou. Mosquitto broker funguje jako prostředník, který distribuuje zprávy mezi zařízeními a zajišťuje, že všechny zprávy jsou doručeny těm, kteří je potřebují.



Obrázek 3: Schéma fungování Mosquitto brokeru v IoT síti.

## 7.2 ESPHome

*ESPHome byl použit pro konfiguraci a správu ESP8266 zařízení. Tento nástroj nahradil ruční implementaci mnoha funkcí, jako je čtení senzorů nebo ovládání pumpy, a zároveň poskytl přímou podporu pro Home Assistant. ESPHome umožňuje snadné nastavení a správu zařízení bez nutnosti psát komplexní kód. Nabízí širokou škálu komponent a automatizačních funkcí, které umožňují snadno přizpůsobit a rozšířit funkčnost systému.*

*Díky ESPHome byla integrace do Home Assistant velmi jednoduchá, protože ESPHome nativně podporuje MQTT a další protokoly, což umožnilo přímou komunikaci mezi zařízením a platformou. Nástroj také umožňuje automatickou aktualizaci zařízení a poskytuje webové rozhraní pro snadnou konfiguraci.*

## 8 Závěr

*Projekt Chytrý květináč úspěšně splnil svůj hlavní cíl – vytvořit automatizovaný systém pro monitorování a péči o rostliny. Mezi klíčové výsledky projektu patří:*

- Integrace více senzorů pro měření vlhkosti půdy, teploty, vlhkosti vzduchu a hladiny vody.
- Úspěšné zavedení manuálního i automatizovaného ovládání pumpy.
- Bezproblémová integrace s Home Assistant pomocí ESPHome.
- Ukládání dat a jejich přehledné zpracování pro vizualizaci.

*Projekt však také narazil na některé výzvy, například omezené hardwarové možnosti ESP8266 při práci s více knihovnami a nutnost optimalizace paměti. Tyto problémy byly většinou úspěšně vyřešeny migrací na ESPHome, která poskytla efektivnější způsob správy zařízení.*

*Budoucí možnosti zahrnují:*

- Rozšíření systému o nové senzory, například světelné senzory nebo senzory pH půdy.
- Přidání pokročilých automatizačních scénářů v Home Assistant.
- Optimalizaci energetické spotřeby zařízení pro dlouhodobý provoz.
- Přidání systému pro dosvítování rostliny pro lepší růst při nízkém osvětlení.

## 9 Splněné a nesplněné cíle

### 9.1 Splněné cíle

- *Implementace systému pro monitorování vlhkosti půdy, teploty, vlhkosti vzduchu a hladiny vody.*
- *Zavedení manuálního i automatizovaného ovládání pumpy.*
- *Zavedení ukládání a vizualizace dat.*
- *Integrace s Home Assistant a zajištění jednoduchého ovládání přes tuto platformu.*

### 9.2 Nesplněné cíle

- *Integrace pokročilejších algoritmů pro analýzu dat, například predikce zalévání na základě historických hodnot.*
- *Optimalizace energetické spotřeby pro napájení zařízení z baterie.*
- *Displej zobrazující hodnoty ze senzorů.*

## 10 Seznam použitých informačních zdrojů

1. *Mischianti. Wemos D1 mini high resolution pinout and specs.*  
*Dostupné z: <https://mischianti.org/wemos-d1-mini-high-resolution-pinout-and-specs/>*
2. *ESPHome. ESPHome documentation.*  
*Dostupné z: <https://esphome.io/>*
3. *Home Assistant. Home Assistant official website.*  
*Dostupné z: <https://home-assistant.io/>*
4. *ESPAAsyncWebServer. ESPAAsyncWebServer GitHub repository.*  
*Dostupné z: <https://github.com/me-no-dev/ESPAAsyncWebServer>*
5. *Random Nerd Tutorials. Random Nerd Tutorials blog.*  
*Dostupné z: <https://randomnerdtutorials.com/>*
6. *Home Assistant Community. Home Assistant Community Forum.*  
*Dostupné z: <https://community.home-assistant.io/>*
7. *Adafruit. Adafruit - ESP8266 and ESP32 tutorials.*  
*Dostupné z: <https://learn.adafruit.com/category/esp8266>*
8. *PubSubClient. PubSubClient GitHub repository.*  
*Dostupné z: <https://github.com/knolleary/pubsubclient>*
9. *Arduino. Arduino official website.*  
*Dostupné z: <https://www.arduino.cc/>*
10. *MQTT.org. MQTT - A Simple and Lightweight Messaging Protocol.*  
*Dostupné z: <https://mqtt.org/>*
11. *GitHub. MQTT integration in Home Assistant.*  
*Dostupné z: <https://github.com/home-assistant/core/blob/dev/homeassistant/components/mqtt/>*
12. *Instructables. Build a Self-Watering System: Soil Moisture Sensor.*  
*Dostupné z: <https://www.instructables.com/Build-a-Self-Watering-System-Soil-Moisture-Sensor-/>*