



# Smart Skleník

## MATURITNÍ PRÁCE

Studijní program: 18-20-M/01 Informační technologie

Studijní obor: Internet věcí

Autor: Stanislav Pixa

Třída: 4.I

Vedoucí práce: Bc. Jakub Škrabánek

Štětí, Duben 2025



## Prohlášení

Prohlašuji, že jsem maturitní práci *Smart Skleník* vypracoval samostatně za použití v práci uvedených pramenů a literatury.

Ve Štětí dne 11. dubna 2025

.....

Podpis studenta



## **Poděkování**

Tímto bych chtěl poděkovat mému vedoucímu práce za mnoho rad a podporu při vypracovávání této maturitní práce.



# Obsah

Úvod	9
<b>1 Využité technologie</b>	<b>11</b>
1.1 Wifi . . . . .	11
1.2 Micropython . . . . .	11
1.3 HTTP . . . . .	11
1.4 Apache2 . . . . .	11
1.5 HTML a CSS . . . . .	12
1.6 JavaScript . . . . .	12
<b>2 Popis využití techniky</b>	<b>13</b>
2.1 ESP32-PICO / M5stickC plus . . . . .	13
2.2 ENV III HAT . . . . .	13
2.3 Raspberry Pi . . . . .	14
<b>3 Popis práce</b>	<b>15</b>
3.1 Řešení napájení . . . . .	15
3.2 Popis konstrukce . . . . .	15
3.3 Popis kódu . . . . .	15
3.4 Popis postupu . . . . .	18
<b>4 Závěr práce</b>	<b>19</b>
<b>Seznam použité literatury</b>	<b>21</b>
<b>Seznam obrázků</b>	<b>23</b>
<b>Seznam zkratk</b>	<b>25</b>
<b>A Formulář v plném znění</b>	<b>29</b>
<b>B Zdrojové kódy výpočetních procedur</b>	<b>31</b>





# Úvod

Cílem tohoto projektu je navrhnout a realizovat chytrý skleník, který využívá znalostí IoT (Internet of Things) k usnadnění práce ve skleníku. Motivací pro výběr tohoto maturitního tématu je skutečnost, že doma vlastním skleník, pro který tento projekt využiji i v budoucnu. Chytrý skleník je pro mě výzvou, na kterou jsem se velice těšil, jelikož mohu uplatnit své zkušenosti z hodin a svou vášeň pro projekty tohoto typu.

Hlavním řešeným problémem je zajištění včasného zalévání skleníku tak, aby vlhkost nebyla příliš nízká a rostliny v něm mohly optimálně růst. Projekt se zaměřuje na sběr dat ze skleníku, jejich zpracování a následné zobrazení na webových stránkách. Tím bude možné sledovat stav skleníku v reálném čase a reagovat na případné změny.

Projekt kombinuje hardware (senzory a mikropočítače) s softwarem, který umožňuje sběr dat, jejich odesílání na servery a následné vizualizaci na webovém rozhraní. Tím bude skleník nejen chytrý, ale také snadno ovladatelný a dostupný odkudkoli.



# 1. Využité technologie

## 1.1 Wifi

WiFi technologie je klíčová pro komunikaci mezi zařízeními v rámci chytrého skleníku a externími systémy, jako jsou servery nebo mobilní aplikace. WiFi umožňuje bezdrátový přenos dat ze senzorů (teploty, vlhkosti, tlaku) přes cloud, až na webovou stránku (1)

## 1.2 Micropython

MicroPython je štíhlá a efektivní implementace programovacího jazyka Python 3 , který zahrnuje malou podmnožinu standardní knihovny Python a je optimalizován pro běh na mikrokontrolérech a v omezených prostředích. (2)

## 1.3 HTTP

HTTP je protokol pro komunikaci na webu, který se v projektu chytrého skleníku používá k přenosu dat mezi zařízeními (senzory, mikrokontroléry) a serverem. Klíčovou metodou je HTTP GET, která slouží k vyžádání dat ze serveru, například aktuálních hodnot teploty nebo vlhkosti. Tato metoda je jednoduchá a efektivní, protože data jsou odesílána přímo v URL požadavku.(3)

## 1.4 Apache2

Apache2 je open-source webový server, který se v projektu chytrého skleníku používá k hostování webového rozhraní a pro vizualizaci dat. Díky své spolehlivosti, flexibilitě a široké podpoře různých modulů je Apache2 ideální volbou pro hosting webové aplikace. (4)

## 1.5 HTML a CSS

HTML slouží k vytvoření struktury webového rozhraní pro zobrazení dat ze skleníku (teplota, tlak, vlhkost). CSS pak styluje stránku, aby byla přehledná a responzivní. Tyto technologie zajišťují, že webové rozhraní funguje na všech zařízeních. (5)

## 1.6 JavaScript

JavaScript je programovací jazyk pro vytváření interaktivních webových stránek. Byl použit pro zobrazení dat ze senzorů a pro interakci předpovědi počasí. Díky JavaScriptu je možné aktualizovat data bez nutnosti obnovení stránky. (6)

## 2. Popis využité techniky

### 2.1 ESP32-PICO / M5stickC plus

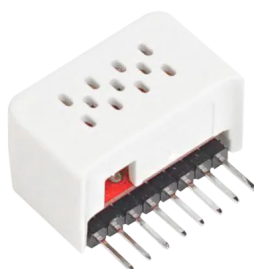
ESP32-PICO je kompaktní modul s mikrokontrolérem ESP32, který kombinuje výkon, nízkou spotřebu a integrované Wi-Fi a Bluetooth. V projektu chytrého skleníku slouží jako řídicí jednotka, která sbírá data ze senzorů. Díky vestavěnému Wi-Fi modulu umožňuje odesílání dat na server a komunikaci s webovým rozhraním. Výhodou tohoto mikrokontroleru je jeho menší velikost, kvůli které je ale její výkon o něco menší než u normální řady ESP32(7)



Obrázek 2.1: M5stickC plus(8)

### 2.2 ENV III HAT

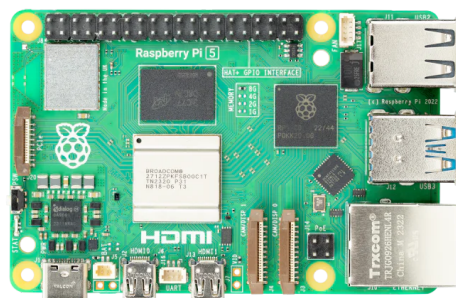
ENV III HAT je přídatný modul od společnosti M5Stack, který obsahuje senzory pro měření teploty, vlhkosti, atmosférického tlaku. V projektu chytrého skleníku slouží k monitorování klíčových parametrů prostředí, které jsou nezbytné pro optimální růst rostlin.(9)



Obrázek 2.2: ENV III HAT(10)

## 2.3 Raspberry Pi

Raspberry Pi je v projektu chytrého skleníku využíváno jako webový server, který hostuje webové rozhraní pro vizualizaci dat ze senzorů. Díky svému výkonu a flexibilitě dokáže Raspberry Pi hostovat webové stránky vytvořené v HTML, CSS a JavaScriptu.(11)



Obrázek 2.3: Raspberry Pi 5(12)

## 3. Popis práce

### 3.1 Řešení napájení

Napájení webového serveru, který hostuje Raspberry Pi, je zajištěno pomocí USB-C kabelu připojeného do elektrické sítě přes adaptér. Toto řešení poskytuje stabilní a spolehlivý zdroj energie, což je klíčové pro nepřetržitý provoz serveru.

Pro ESP32-PICO je zvoleno napájení přes USB-C z kompaktní powerbanky. Toto řešení zajišťuje mobilitu zařízení a umožňuje jeho provoz i v místech bez přímého přístupu k elektrické síti.

### 3.2 Popis konstrukce

Konstrukce využívá senzor ENV HAT III, který je přímo kompatibilní s mikro-počítačem M5StickC. Toto uspořádání umožňuje snadné zapojení bez zbytečných kabelů.

### 3.3 Popis kódu

S ohledem na omezenou paměť VRAM mikrokontroleru je jeho kód velmi strohý a obsahuje jen to nezbytné. Byl kladen důraz na spolehlivost zařízení, aby bylo zajištěno, že se každých 15minut odešlou údaje ze senzoru. Dále, webová stránka obsahuje čtyři soubory s kódem: HTML, CSS a dva JavaScript soubory. Stránka je v minimalistickém stylu a tím pádem je velmi skromná. JavaScript aktualizuje data o nejnovějších udajích a ukazuje předpověď počasí na další dny. A CSS stránku zabalí do příjemně-tajemného kabátu



Obrázek 3.1: Webová stránka

```

Matthias > @ matulap > ...
1 import os, sys, io
2 import RPi
3 from RPi import *
4 from hardware import I2C
5 from hardware import Pin
6 from hat import ENVMut
7 import time
8 import requests
9
10
11 label2, label3, label4, label5, X0, I2C0, hat_env3_0, text, posta, teplota, vlhkost, tlak = [None] * 12
12 rtc = RTC()
13 reset = 1
14
15 ##### Nastavení displaye #####
16 W5.begin()
17 label2 = widgets.Label("Poslední data:", 0, 0, 1.0, 0xffff, 0x222222, widgets.FONTS.DejaVu18)
18 label3 = widgets.Label("Teplota: None", 0, 29, 1.0, 0xffff, 0x222222, widgets.FONTS.DejaVu12)
19 label4 = widgets.Label("Vlhkost: None", 0, 45, 1.0, 0xffff, 0x222222, widgets.FONTS.DejaVu12)
20 label5 = widgets.Label("Tlak: None", 1, 61, 1.0, 0xffff, 0x222222, widgets.FONTS.DejaVu9)
21 I2C0 = I2C(0, scl=Pin(26), sda=Pin(0), freq=100000)
22 hat_env3_0 = ENVMut(I2C0, type=3)
23
24 ##### HTTP REQUEST #####
25 def http_request(server_address, teplota, tlak, vlhkost):
26     try:
27         response = requests.get("{}(server_address){field1}={teplota}&field2={tlak}&field3={vlhkost}")
28         if response.status_code == 200:
29             print(response.status_code)
30             gc.collect()
31             response.close()
32         else:
33             print("Chyba při odesílání teploty: {}".format(response.status_code))
34     except Exception as e:
35         print("Chyba spojení: {}".format(e))
36         return None
37
38 server_address = "https://api.thingspeak.com/update?api_key=QXWU8JACM9807P96"
39
40 ##### Loop #####
41
42
43 while True:
44     teplota = hat_env3_0.read_temperature()
45     vlhkost = hat_env3_0.read_humidity()
46     tlak = hat_env3_0.read_pressure()
47     label3.setText(str(teplota))
48     label4.setText(str(vlhkost))
49     label5.setText(str(tlak))
50     http_request(server_address, teplota, tlak, vlhkost)
51
52

```

Obrázek 3.2: MicroPython



```

Maturita > testik > o index.html > @html
1 <!DOCTYPE html>
2 <html lang="cs">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Sklenik</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10   <div class="header">
11     <div class="nav-bar">
12       <div class="logo">
13         <a href="#">Sklenik</a>
14       </div>
15       <div class="nav-links">
16         <div href="#">Stanislav Pixa</a></div>
17       </div>
18     </div>
19   </div>
20   <div class="background-container">
21     <div class="list">
22       
23     </div>
24     <div class="image-item">
25       <div class="img" width="450" height="200" style="border: 1px solid #000000;" src="https://thingspeak.com/channels/2784893/charts/1?bgcolor=323232&dynamic=true&results=0&title=teplota&type=line&xaxis=Datum&yaxis=K2300C"></div>
26       <div class="mappls">Poslední teplota:</div>
27       <div class="popis">Id=lastteplota&načítání...</div>
28     </div>
29     <div class="image-item">
30       <div class="img" width="450" height="200" style="border: 1px solid #000000;" src="https://thingspeak.com/channels/2784893/charts/2?bgcolor=323232&dynamic=true&results=0&title=tlak&type=line&xaxis=Datum"></div>
31       <div class="mappls">Poslední tlak:</div>
32       <div class="popis">Id=lasttlak&načítání...</div>
33     </div>
34     <div class="image-item">
35       <div class="img" width="450" height="200" style="border: 1px solid #000000;" src="https://thingspeak.com/channels/2784893/charts/3?bgcolor=323232&dynamic=true&results=0&title=vlhkost&type=line&xaxis=Datum&yaxis=K215Vlhkosti"></div>
36       <div class="mappls">Poslední vlhkost:</div>
37       <div class="popis">Id=lastvlhkost&načítání...</div>
38     </div>
39   </div>
40   <div class="pocasi">
41     <div class="weather-widget">
42       <a href="https://forecast7.com/cs/S00314026/rouhnice-nad-labem/" data-label="1">ROUHNEE NAD LABEM</a> data-icons="Climacore Animated" data-theme="sky" data-shadow="rgba(0, 0, 0, 0.45)" data-highcolor="#ff7474" data-lowcolor="#ff0000">
43     </div>
44   </div>
45   <script src="script.js"></script>
46   <script src="pocasi.js"></script>
47 </body>
48 </html>

```

Obrázek 3.3: HTML

```

Maturita > testik > # script > ...
1 const channelId = "2784893";
2 const url = "https://api.thingspeak.com/channels/${channelId}/feeds.json?results=1";
3 const apiKey = "EZ8J7K3XQ0Q1P8M";
4 const apiUrl = "https://api.thingspeak.com/channels/${channelId}/feeds.json?api_key=${apiKey}&results=1";
5
6 let lastEntryId = null;
7
8 async function teplota() {
9   const fieldNumber = 1;
10   try {
11     const response = await fetch(url);
12     const data = await response.json();
13     const lastEntry = data.feeds[data.feeds.length - 1]; // Poslední záznam
14     const lastTeplota = lastEntry[fieldNumber];
15     document.getElementById("lastTeplota").innerText = `${Math.round(lastTeplota)}°C || "Hejzou data";
16   } catch (error) {
17     document.getElementById("lastTeplota").innerText = "Chyba při načítání";
18     console.error("Chyba při načítání dat", error);
19   }
20 }
21
22 async function tlak() {
23   const fieldNumber = 2;
24   try {
25     const response = await fetch(url);
26     const data = await response.json();
27     const lastEntry = data.feeds[data.feeds.length - 1]; // Poslední záznam
28     const lastTlak = lastEntry[fieldNumber];
29     document.getElementById("lastTlak").innerText = `${Math.round(lastTlak)} hPa || "Hejzou data";
30   } catch (error) {
31     document.getElementById("lastTlak").innerText = "Chyba při načítání";
32     console.error("Chyba při načítání dat", error);
33   }
34 }
35
36 async function vlhkost() {
37   const fieldNumber = 3;
38   try {
39     const response = await fetch(url);
40     const data = await response.json();
41     const lastEntry = data.feeds[data.feeds.length - 1]; // Poslední záznam
42     const lastVlhkost = lastEntry[fieldNumber];
43     document.getElementById("lastVlhkost").innerText = `${Math.round(lastVlhkost)} % || "Hejzou data";
44   } catch (error) {
45     document.getElementById("lastVlhkost").innerText = "Chyba při načítání";
46     console.error("Chyba při načítání dat", error);
47   }
48 }
49
50 teplota();
51 vlhkost();
52 tlak();
53 function checkResponseData() {
54   fetch(apiUrl)
55     .then(response => response.json())
56     .then(data => {
57       const lastEntry = data.feeds[0]; // Poslední záznam
58       if (lastEntry && lastEntry.entry_id !== lastEntryId) {
59         lastEntryId = lastEntry.entry_id;
60       }
61     });
62 }

```

Obrázek 3.4: JavaScript

## 3.4 Popis postupu

Svůj maturitní projekt jsem začal sběrem dat pomocí ESP32-PICO, ke kterému jsem připojil senzor ENV III HAT a propojil jej se svým stolním počítačem. Jako první krok jsem stáhl oficiální nástroj M5Burner, který slouží k nahrání firmwaru do mikrokontroleru. Tento firmware umožnil programování ESP32-PICO pomocí webové aplikace, což výrazně zjednodušilo vývoj.

Po připojení všech komponentů a nahrání potřebného softwaru jsem začal psát kód pro sběr dat ze senzorů a jejich odesílání na server pomocí HTTP požadavků. Po dokončení kódu jsem provedl testování, které trvalo přibližně dvě hodiny.

Webová stránka pro mě obsahovala neznámý JavaScript, který jsem se učil několik hodin od základu. S využitím moderních technologií jsem se dopracoval k kostře stránky, na kterou jsem přidal grafy z stránek Thinkspeaku a přidal k nim aktuální hodnoty pomocí nově naučeného JavaScriptu. Poslední věcí byl widget na počasí, pro který jsem použil weatherwidget.io.

## 4. Závěr práce

Během práce jsem si zopakoval práci s mikrokontrolery, minipočítači, pythonem, html a dalšími. Naučil jsem se, jak pracovat s JavaScriptem a hlavně jsem si rozšířil obzory v práci s Webovými stránkami. I když se tato maturitní práce neobešla bez komplikací a problémů, jsem velmi spokojený s výběrem tématu a v budoucnu se jistě vrhnu do více podobných projektů. Budoucnost tohoto projektu bude na zahradě ve skleníku, kam patří.



# Seznam použité literatury

1. *Discover Wi-Fi* [online]. 2023. [cit. 2025-02-17]. Dostupné z: <https://www.wi-fi.org/discover-wi-fi>.
2. *MicroPython - Python for microcontrollers* [online]. 2014. [cit. 2025-02-17]. Dostupné z: <https://micropython.org/>.
3. *RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1* [online]. 1999. [cit. 2025-02-17]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc2616>.
4. *About the Apache HTTP Server Project* [online]. 1997. [cit. 2025-02-17]. Dostupné z: [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html).
5. *HTML: HyperText Markup Language* [online]. 1998. [cit. 2025-02-17]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
6. *JavaScript* [online]. 1998. [cit. 2025-02-17]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
7. *ESP32 Series* [online]. 2024. [cit. 2025-02-17]. Dostupné z: <https://www.espressif.com/en/products/socs/esp32>.
8. *M5stickC plus*. Dostupné také z: <https://www.ubuy.cz/en/product/ZDKKQC6-m5stack-m5stickc-plus-esp32-pico-d4-mini-iot-development-kit-deauther-watch-support-ble-4-2-and-wifi?ref=hm-google-redirect#gallery-10>.
9. *HAT ENV III* [online]. 2021. [cit. 2025-02-17]. Dostupné z: [https://docs.m5stack.com/en/hat/hat\\_envIII](https://docs.m5stack.com/en/hat/hat_envIII).
10. *ENV III HAT*. Dostupné také z: [http://cz.mouser.com/new/m5stack/m5stack-env-iii-hat/?\\_gl=1\\*p7tod1\\*\\_ga\\*MTU5NjAzMTM0NS4xNzM5ODQwMDAx\\*\\_ga\\_15W4STQT4T\\*MTczOTg0MDAwMS4xLjAuMTczOTg0MDAwMi41OS4wLjA](http://cz.mouser.com/new/m5stack/m5stack-env-iii-hat/?_gl=1*p7tod1*_ga*MTU5NjAzMTM0NS4xNzM5ODQwMDAx*_ga_15W4STQT4T*MTczOTg0MDAwMS4xLjAuMTczOTg0MDAwMi41OS4wLjA).
11. *Raspberry Pi 5* [online]. 2012. [cit. 2025-02-17]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-5/>.
12. *Raspberry Pi*. Dostupné také z: <https://vilros.com/products/raspberry-pi-5>.



# Seznam obrázků

2.1	M5stickC plus(8)	13
2.2	ENV III HAT(10)	13
2.3	Rapsberry Pi 5(12)	14
3.1	Webová stránka	16
3.2	MicroPython	16
3.3	HTML	17
3.4	JavaScript	17





# Seznam použitých zkratek

**Wi-Fi** Wireless Fidelity

**HTTP** HyperText Transfer Protocol

**USB** Universal Serial Bus

**CSS** Cascading Styleheets

**VRAM** Video random Access Memory

**HTML** Hypertext Markup Language

**URL** Uniform Resource Locator



# Přílohy



## **A. Formulář v plném znění**



## **B. Zdrojové kódy výpočetních procedur**