**STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

**Obor č. číslo oboru: Název oboru**

(červeně psané věci přepište podle skutečnosti a změňte jejich barvu na černo. Toto doporučení vymažte)

**Plný název práce v češtině**

**Jméno autora/autorky/autorů**

**Oficiální název kraje Město a vročení**

**STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

**Obor č. číslo oboru: Název oboru**

**Plný název práce v češtině**

**Plný název práce v angličtině**

**Autoři: Petr Franc**

**Škola: SPŠE Ječná**

**Kraj: Praha**

**Konzultant: Jura**

PRAHA 2023

**Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne xx.xx.2023 Petr Franc

**Poděkování**

Na tomto místě můžete vložit poděkování těm, kteří vám s tvorbou práce SOČ pomohli. Poděkování je vaše autorské dílo, nemá předepsanou podobu a není povinnou součástí práce SOČ. Záleží jen na vás, zda, komu a jakým způsobem poděkujete

**Anotace**

Tato práce popisuje návrh a vývoj meteorologické stanice s mikrokontrolérem ESP32 a e-ink displejem. Stanice je navržena pro sledování aktuálních meteorologických podmínek, jako jsou teplota a vlhkost. Stanice také nabízí uživatelské rozhraní, které umožňuje snadnou vizualizaci dat. Hlavním cílem práce je vytvořit spolehlivou a efektivní meteorologickou stanici, která bude vhodná pro použití v různých prostředích. Práce popisuje detailní postup návrhu hardwaru a programování mikrokontroléru, včetně použitých komponentů. Tato práce může být užitečná pro vývojáře a nadšence, kteří chtějí vytvořit vlastní meteorologickou stanici s ESP32 mikrokontrolérem a e-ink displejem.

**Klíčová slova**

ESP32, C++, IOT, E-INK, Wi-Fi konektivita

**Annotation**

This work describes the design and development of a meteorological station with ESP32 microcontroller and e-ink display. The station is designed for monitoring current weather conditions, such as temperature, humidity, pressure, and wind speed. The sensor outputs are processed in the ESP32 microcontroller and transmitted to a server using Wi-Fi connectivity. The station also offers a user interface that allows easy visualization of data. The main goal of the work is to create a reliable and efficient meteorological station suitable for use in various environments. The work describes a detailed process of hardware design and microcontroller programming, including the sensors and other components used. This work may be useful for developers and enthusiasts who want to create their own meteorological station with ESP32 microcontroller and e-ink display.

**Keywords**

ESP32, C++, IOT, E-INK, Wi-Fi connectivity

Obsah

[1 Úvod 5](#_Toc128643785)

[2 Jazyk, vývojové prostředí 5](#_Toc128643786)

[2.1 Jazyk 5](#_Toc128643787)

[2.2 Vývojové prostředí 5](#_Toc128643788)

[2.2.1 PlatformIO 5](#_Toc128643789)

[3 Meteorologická stanice 5](#_Toc128643790)

[3.1 Co to je meteorologická stanice 5](#_Toc128643791)

[4 Architektura systému 6](#_Toc128643792)

[4.1 Komunikace se sensorem HTU21D 6](#_Toc128643793)

[5 Technologie a vybavení 6](#_Toc128643794)

[5.1 ESP32 6](#_Toc128643795)

[5.2 E-ink 6](#_Toc128643796)

[5.3 Sensor HTU21D 6](#_Toc128643797)

[5.4 Baterie 7](#_Toc128643798)

[5.5 Krabička 7](#_Toc128643799)

[6 Popis programu 7](#_Toc128643800)

[6.1 Flow Diagram 7](#_Toc128643801)

[6.2 Pokus o připojení 7](#_Toc128643802)

[6.3 AP Mode 7](#_Toc128643803)

[6.4 Stáhnutí dat z internetu 8](#_Toc128643804)

[6.5 Načtení dat ze sensoru 8](#_Toc128643805)

[6.6 Zobrazení dat na display 8](#_Toc128643806)

[6.7 Přechod do deepsleep 8](#_Toc128643807)

[7 Struktura programu 8](#_Toc128643808)

[7.1 Class Diagram 8](#_Toc128643809)

[7.2 Main 8](#_Toc128643810)

[7.3 Setup 9](#_Toc128643811)

[7.4 GxEPD display 9](#_Toc128643812)

[7.5 UIDocument 9](#_Toc128643813)

[7.5.1 Json document 9](#_Toc128643814)

[7.5.2 Element 10](#_Toc128643815)

[7.6 UIRenderer 10](#_Toc128643816)

[7.7 Wifi Manager 11](#_Toc128643817)

[7.8 File System 11](#_Toc128643818)

[7.9 HTU21D 12](#_Toc128643819)

[7.10 HTTPClient 12](#_Toc128643820)

[7.10.1 REST API Request 13](#_Toc128643821)

[7.11 fileSystemUtils.hpp 13](#_Toc128643822)

[7.12 jsonUtils.hpp 14](#_Toc128643823)

[8 Uživatelské rozhraní 14](#_Toc128643824)

[9 Popis vývojového procesu 14](#_Toc128643825)

[10 Výsledky a zhodnocení 14](#_Toc128643826)

[10.1 Shrnutí dosažených cílů 14](#_Toc128643827)

[10.2 Zhodnocení průběhu projektu 14](#_Toc128643828)

[10.3 Možná zlepšení 14](#_Toc128643829)

[11 Závěr 14](#_Toc128643830)

[12 Použitá literatura 15](#_Toc128643831)

[13 Seznam obrázků a tabulek 15](#_Toc128643832)

# Úvod

Tato dokumentace popisuje vývoj meteorologické stanice s mikrokontrolérem ESP32 a e-ink displejem. Tato stanice slouží k monitorování aktuálních meteorologických podmínek, jako jsou teplota, vlhkost. Stanice využívá Wi-Fi konektivitu k přenosu dat na server a nabízí uživatelské rozhraní pro snadnou vizualizaci dat. Hlavním cílem tohoto projektu je vytvořit efektivní a spolehlivou meteorologickou stanici, která bude vhodná pro použití v různých prostředích.

Tento projekt vyžaduje znalosti v oblasti hardwaru, software a programování mikrokontrolérů. V této dokumentaci najdete informace o návrhu hardwaru, programování mikrokontroléru a rozhraní s uživatelem. Dokumentace by měla být užitečná pro vývojáře a nadšence, kteří chtějí postavit vlastní meteorologickou stanici s ESP32 mikrokontrolérem a e-ink displejem.

# Jazyk, vývojové prostředí

## Jazyk

Zvolil jsem jazyk C++ pro programování ESP32, protože ho na dobré úrovni ovládám a mám s ním zkušenosti. C++ je objektově orientovaný programovací jazyk, který je výkonný, flexibilní a používá se pro vývoj softwaru. Podporuje mnoho funkcí, jako jsou šablony, přetěžování operátorů a vícevláknové programování. Je známý pro svou rychlost a efektivitu, ale může být náročný na učení se.

## Vývojové prostředí

Jako IDE používám Visual Studio Code s pluginem Platform.io, který umožňuje práci s mikrokontrolery. Visual Studio Code (VS Code) je open-source editor kódu, který se používá pro vývoj softwaru na různých platformách. Jeho flexibilita, možnosti rozšíření a snadná konfigurace ho dělají oblíbeným mezi programátory.

### PlatformIO

PlatformIO je plugin pro VS Code, který usnadňuje vývoj softwaru pro embedded systémy a IoT zařízení. Podporuje mnoho různých mikrokontrolérů a nabízí užitečné funkce, jako jsou inteligentní kódo-ové zvýraznění syntaxe, ladění kódu a správa knihoven. To usnadňuje programování pro embedded systémy a IoT aplikace, což je jinak náročné a časově náročné.

# Co to je meteorologická stanice

Meteorologická stanice, kterou vytvářím informuje o teplotě a vlhkosti v Praze, stručný popis počasí, a teplota a vlhkost v místnosti. Ještě k tomu stanice vypíše na displej datum a čas. Teplota a vlhkost v místnosti se získá pomocí HTU21D sensoru. Teplota, vlhkost, popis počasí, datum a čas v Praze se získá přes internet pomocí http requestu na server „api.openweathermap.org“. Jelikož stanice získává data přes internet, musí být připojena přes wifi. Po načtení a zobrazení dat jde stanice do deepsleepu po dobu 15 minut, poté se proces opakuje. Pokud se stanici nepodaří připojit na internet. Přepne se do AP modu, ve kterém muže uživatel zadat wifi na kterou se má stanice připojit. Pokud i to selže stanice se podívá na poslední data, které stáhla a uložila do paměti a zobrazí je. Toto neplatí pro data o místní teplotě a vlhkosti, jelikož ta se získávají přes sensor HTU21D, který se dá použít nezávisle na připojení internetu.

# Architektura systému

## Schéma zapojení

## Komunikace se sensorem HTU21D

Komunikace mezi senzorem HTU21D a mikrokontrolerem ESP32 probíhá pomocí protokolu I2C (Inter-Integrated Circuit). I2C je sériový komunikační protokol, který umožňuje komunikaci mezi mikrokontrolery a dalšími periferiemi, jako jsou senzory, displeje nebo paměťová média.

Dotaz na senzor HTU21D se skládá z několika kroků. Nejprve musí být poslána zahajovací sekvence, která oznamuje senzoru, že bude následovat dotaz. Poté se posílá samotný dotaz, který obsahuje instrukce pro získání dat o teplotě a vlhkosti. Senzor HTU21D odpoví na tento dotaz přenosem dat o teplotě a vlhkosti zpět k mikrokontroleru ESP32. Na závěr se komunikace ukončí vysláním ukončovací sekvence.

O tyto kroky používáme knihovnu sensoru. Více informací o sensoru zde, nebo o knihovně zde.

## Komunikace s e-ink displejem

Komunikace mezi ESP32 a e-ink displejem pomocí knihovny GxEPD (Grayscale e-paper display) probíhá pomocí komunikačního rozhraní SPI (Serial Peripheral Interface), které je běžné u mnoha mikrokontrolérů, včetně ESP32.

ESP32 musí být propojen s displejem pomocí čtyř vodičů: SCK, MOSI, MISO a CS. Když ESP32 potřebuje ovládat displej, nastaví CS na LOW a začne zasílat příkazy pomocí SPI. Poté, co jsou příkazy odeslány, nastaví se CS zpět na HIGH.

Komunikace s e-ink displejem pomocí knihovny GxEPD je tedy poměrně jednoduchá a snadno se dá integrovat do kódu programu na ESP32.

# Technologie a vybavení

Stručný popis komponentů

## ESP32

Text

ESP32 je výkonný a všestranný mikrokontrolér, který se často používá pro vývoj IoT aplikací a embedded systémů. Obsahuje dvě jádra procesoru, Wi-Fi a Bluetooth modul, a mnoho periférií pro komunikaci a ovládání různých zařízení. ESP32 je také nízkopříkonový a relativně levný, což z něj činí ideální volbu pro vývoj aplikací pro IoT.

## E-ink

E-ink display (též nazývaný jako elektronický papír) je typ displeje, který nevyžaduje podsvícení a spotřebovává energii pouze při změně zobrazovaného obsahu. To z něj činí ideální volbu pro aplikace, které vyžadují nízkou spotřebu energie, jako jsou například elektronické čtečky knih.

E-ink displeje mohou být snadno řízeny pomocí mikrokontroléru, jako je ESP32 nebo Arduino, a jsou k dispozici knihovny pro řízení těchto displejů. Tyto knihovny umožňují jednoduché zobrazování textu, obrázků a grafiky na displeji s nízkou spotřebou energie. E-ink displeje jsou tedy ideální volbou pro aplikace, které potřebují nízkou spotřebu energie a jednoduché zobrazování informací.

Tento displej už je integrovaný v desce.

Odkaz na desku zde.

## Sensor HTU21D

HTU21D je digitální teplotní a vlhkostní senzor vyrobený společností Measurement Specialties, který je obvykle používán v embedded aplikacích a IoT zařízeních. Tento senzor měří relativní vlhkost a teplotu okolního prostředí a komunikuje pomocí digitálního rozhraní I2C. Senzor HTU21D je také známý pro svou vysokou přesnost a malé rozměry, což z něj činí ideální volbu pro aplikace, které vyžadují přesné měření teploty a vlhkosti s malou spotřebou energie. Senzor měří teplotu s přesností ±0,3°C a vlhkost s přesností ±2% relativní vlhkosti.

## Baterie

GeB Li-Ion Baterie 1x18650 1S1P 3.7V 3200mAh je dobíjecí lithium-iontová baterie navržená pro širokou škálu aplikací, včetně elektronických zařízení, robotiky, přenosných zdrojů energie, osvětlení a mnoha dalších. Tato baterie má kapacitu 3200mAh a napětí 3,7V. Její výstupní proud může dosáhnout až 10 A.

Baterie GeB Li-Ion 1x18650 je balena v kovovém pouzdře s rozměry 18 x 65 mm a hmotností 50 gramů. Je vybavena ochranou proti přepólování a přetížení. Baterie používá lithiové elektrody pro ukládání energie a má velmi dobrou výdrž, nízkou sebevybíjecí rychlost a vysoký počet cyklů nabíjení.

Tato baterie je určena pro profesionální a průmyslové aplikace, ale může být také vhodná pro použití v hobby projektech. Je důležité si uvědomit, že použití jakékoli lithium-iontové baterie vyžaduje dodržování bezpečnostních postupů, jako je používání správného nabíječe a skladování na bezpečném místě, aby se minimalizovalo riziko požáru nebo výbuchu.

Odkaz na baterii [zde](https://www.laskakit.cz/geb-li-ion-baterie-1x18650-1s1p-3-7v-3200mah/?)

## Krabička

# Uživatelské rozhraní

# Popis programu

## Flow Diagram

Diagram a stručný popis.

## Pokus o připojení

Připojení k Wifi sítím probíhá pomocí knihovny Wifi Manager pro ESP32 mikrokontroléry a probíhá následovně:

1. Pokud ESP32 mikrokontrolér nenajde žádnou konfigurovanou Wifi síť, automaticky se spustí Wifi Manager a začne nabízet možnost nastavení nové Wifi sítě.
2. V případě, že je již Wifi síť nastavena, ESP32 se připojí k této Wifi síti pomocí uložených přihlašovacích údajů.
3. Pokud se ESP32 mikrokontrolér nedokáže připojit k Wifi síti pomocí uložených údajů, automaticky se spustí Wifi Manager a nabídne možnost zadání nových přihlašovacích údajů pro Wifi síť.

Připojení k Wifi síti pomocí knihovny Wifi Manager je tedy velmi jednoduché a uživatelsky přívětivé. Uživatel nemusí ručně zadávat konfigurační údaje pro Wifi síť, což šetří čas a minimalizuje chyby při nastavování Wifi připojení.

## AP Mode

AP (Access Point) mód v rámci Wifi Manageru pro ESP32 mikrokontroléry umožňuje vytvoření vlastní Wifi sítě, ke které se mohou další zařízení připojit a nakonfigurovat Wifi Manager. V AP módu se ESP32 mikrokontrolér chová jako Wifi router a vytváří svou vlastní Wifi síť s předem definovaným názvem a heslem.

Pro nastavení Wifi Manageru v AP módu je potřeba nejprve zavolat funkci wifiManager.autoConnect() a připojit se k Wifi síti vytvořené ESP32 mikrokontrolérem. Poté lze přistoupit ke konfiguraci Wifi Manageru přes webové rozhraní, které je dostupné na adrese 192.168.4.1.

V AP módu lze konfigurovat Wifi Manager pomocí jakéhokoli zařízení, které podporuje Wifi připojení, jako jsou například chytré telefony, počítače nebo tablety. Po dokončení konfigurace se ESP32 mikrokontrolér automaticky připojí k nové Wifi síti a přestane fungovat jako Wifi router v AP módu.

AP mód je užitečný v situacích, kdy není k dispozici žádná jiná Wifi síť pro připojení a konfiguraci Wifi Manageru.

## Stáhnutí dat z internetu

Stanice stahuje meteorologická data ze serveru "api.openweathermap.org" pomocí třídy HttpClientu, která je k dispozici v knihovně pro ESP32. HttpClient poskytuje prostředky pro komunikaci s webovými servery pomocí protokolu HTTP. Stanice se připojí k serveru a odešle požadavek na získání dat. Po obdržení odpovědi od serveru analyzuje stanice data a ukládá je do paměti.

### OpenWeatherMap API

api.openweathermap.org je veřejné API pro získání aktuálního a předpovědního meteorologického obsahu, jako jsou teplota, vlhkost, rychlost větru, atmosférický tlak a mnoho dalšího pro libovolné geografické místo na světě. Toto API je poskytováno společností OpenWeatherMap, která sbírá a zpracovává data od meteorologických stanic a dalších zdrojů. Tato data jsou pak poskytována v různých formátech, jako jsou JSON, XML nebo HTML. API je běžně používáno pro vývoj mobilních aplikací, webových stránek a IoT projektů, které potřebují aktuální meteorologické informace.

### Použití OpenWeatherMap API

Pro použití API openweather.org je potřeba nejprve získat svůj API klíč, který se používá pro autorizaci při každém dotazu na API.

Po získání API klíče je možné vytvořit dotaz na API pomocí HTTP protokolu. Dotaz je tvořen URL adresou, na kterou se odesílá dotaz s parametry.

Například pro získání aktuálního počasí pro město Praha v jednotkách Celsius je URL adresa:

http://api.openweathermap.org/data/2.5/weather?q=Praha&units=metric&appid={API\_KEY}

V této adrese jsou tři parametry:

* q - označuje město, pro které se má počasí získat
* units - označuje jednotky, ve kterých se má počasí zobrazit
* appid - API klíč pro autorizaci

Po odeslání dotazu na API se v odpovědi vrátí JSON objekt s požadovanými informacemi o počasí. Tento objekt je nutné zpracovat a případně zobrazit na webové stránce nebo v aplikaci.

## Načtení dat ze sensoru

Stanice získává data o teplotě a vlhkosti v místnosti pomocí senzoru HTU21D, který je připojen k ESP32 mikrokontroléru. Senzor HTU21D je digitální teploměr a hygrometr, který poskytuje přesné měření teploty a vlhkosti.

## Zobrazení dat na display

Data se na display zobrazují díky třídě UIRenderer.

## Přechod do deepsleep

Po zobrazení dat na displeji meteorologické stanice se stanice přepne do režimu hlubokého spánku, zvaného "deepsleep", po dobu 15 minut. Během této doby jsou vypnuty všechny nezbytné komponenty stanice, s výjimkou ESP32 mikrokontroléru, který zůstává v režimu hlubokého spánku, přičemž spotřeba energie se minimalizuje. Tento krok je důležitý pro prodloužení životnosti baterie stanice a zabraňuje zbytečné spotřebě energie, když stanice nevykonává žádnou funkci. Po uplynutí 15 minut se stanice probudí z režimu hlubokého spánku a opakuje cyklus.

# Struktura programu

## Class Diagram

TODO:Diagram a stručný popis.

## Main

Hlavní část programu, kde se nachází funkce „setup“ a „loop“. Ve funkci „setup“ se odehrává všechen kód a po ukončení funkce jde stanice do deepsleepu.

Funkce v souboru umožňují:

* Připojení k WiFi a automatické připojení k předchozí síti při spuštění.
* Dotazování se na OpenWeatherMap API a získávání aktuálního počasí. Pokud dotaz selže, použije se uložené lokální počasí.
* Načítání a zobrazování dat na displeji pomocí knihoven GxEPD, UIRenderer a UIDocument
* Ukládání a čtení dat pomocí SPIFFS (SPI Flash File System).
* Používání HTU21D senzoru pro měření teploty a vlhkosti uvnitř.

## Setup

Funkce "setup" se spouští na začátku programu a připravuje zařízení pro další běh. Spustí se funkce "init", která inicializuje zařízení, a nakonec se spustí funkce "run", která provede hlavní běh programu. Funkce "run" je hlavní funkcí programu, která řídí celý průběh aplikace. Nejprve inicializuje objekt "StationData", který slouží pro ukládání dat o počasí. Poté se inicializuje a vykreslí se stránka pro připojení k Wi-Fi sítím. Pokud se podaří připojení k internetu, tak se načtou data o počasí z webového API. Pokud k připojení nedojde, tak se načtou data o počasí z paměti. Nakonec se vykreslí hlavní menu aplikace s aktuálními daty o počasí. Po dokončení běhu funkce "run" se na konzoli vypíše zpráva "Done going to sleep" a zařízení přejde do hlubokého spánku pomocí funkce "esp\_deep\_sleep\_start".

## GxEPD display

GxEPD je knihovna pro Arduino a ESP32 platformy, která umožňuje snadné ovládání E-ink displejů (elektronických papírů) vytvořených společností Good Display. Tato knihovna podporuje různé velikosti a typy displejů, včetně barevných a černobílých, a umožňuje snadnou a rychlou integraci displeje do projektu s ESP32 mikrokontrolérem.

Knihovna GxEPD poskytuje funkce pro inicializaci displeje, vykreslování textu a grafiky, aktualizaci displeje a další. Podporuje také nízkou spotřebu energie a umožňuje zobrazovat informace na displeji po dlouhou dobu bez nutnosti častého obnovování.

Díky svým vlastnostem se GxEPD knihovna stala oblíbenou volbou pro vývojáře, kteří chtějí použít E-ink displeje v různých projektech, jako jsou například nízkopříkonové zařízení pro IoT aplikace, čtečky e-knih, nebo displeje pro náramkové hodinky a další podobné zařízení.

## UIDocument

Třída UIDocument je součástí knihovny pro tvorbu uživatelského rozhraní v prostředí Arduino a umožňuje vytvoření hierarchického stromu elementů, které tvoří uživatelské rozhraní. Tyto elementy jsou definovány v JSON formátu.

Konstruktor třídy přijímá JSON objekt, šířku a výšku displeje a pole JSON tříd, které mohou být použity k definování vzhledu elementů. Metoda getFileAsString() načte obsah souboru do řetězce a používá se pro načítání souborů z disku.

Metoda jsonToElement() převádí JSON objekty na objekty typu Element, které jsou použity pro reprezentaci uživatelského rozhraní. Metoda printElement() slouží k výpisu hierarchie elementů v konzoli pro účely debugování.

Další metody třídy UIDocument jsou použity pro výpočet rozmístění elementů vzhledem k ostatním elementům v hierarchii, zarovnání elementů a nalezení konkrétních elementů pomocí jejich názvu.

### Json document

JSON (JavaScript Object Notation) je jednoduchý formát pro výměnu dat. Jedná se o textový formát, který slouží k reprezentaci strukturovaných dat. JSON dokument se skládá z kolekcí párů klíč-hodnota, které mohou obsahovat různé datové typy, jako jsou řetězce, čísla, pole a další objekty. Dokument začíná a končí složenými závorkami a každý pár klíč-hodnota je oddělen čárkou. JSON se používá v mnoha aplikacích, protože je snadno čitelný a zapisovatelný pro člověka i stroj.

### Element

Třída Element obsahuje různé vlastnosti, které popisují element v uživatelském rozhraní. Má proměnnou border, která určuje, zda má element okraj, výčtový typ ElementType, který popisuje typ elementu, proměnnou name, která udává název elementu, proměnné posX a posY, které určují pozici elementu na ose X a Y, a proměnné sizeX a sizeY, které určují šířku a výšku elementu.

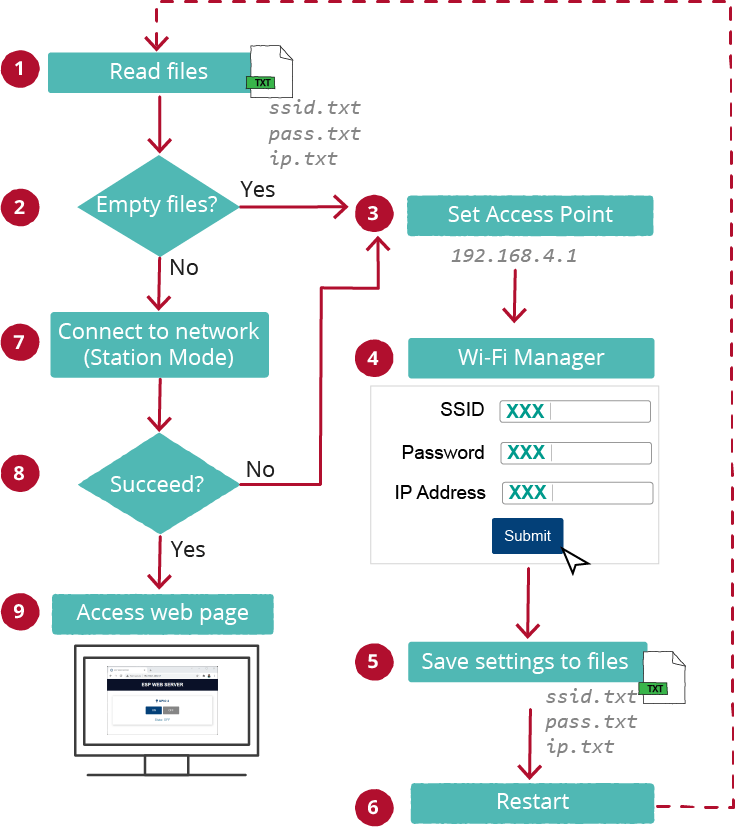
Dále má proměnnou value, která udává hodnotu elementu, proměnnou textSize, která určuje velikost textu, výčtový typ AlignType, který určuje typ zarovnání, počet potomků (nOfChildren) a ukazatel na pole potomků (children) typu Element.

## UIRenderer

Třída UIRenderer slouží k vykreslování prvků uživatelského rozhraní na displej. Obsahuje metody pro vykreslení rámce prvku, textu a pro rekurzivní vykreslení celého dokumentu.

Konstruktor třídy "UIRenderer" přijímá ukazatel na displejový objekt typu "GxEPD\_Class", který je používán k vykreslení prvků na displej. Tato třída obsahuje tři funkce pro vykreslení různých prvků UI: "drawBorder" pro vykreslení rámečku prvku, "drawText" pro vykreslení textového prvku a "drawElement" pro rekurzivní vykreslení celého stromu prvků. Funkce "drawElement" je hlavní funkcí, která vykresluje celý strom prvků. Tato funkce rekurzivně prochází všechny prvky ve stromu a pro každý prvek volá příslušnou vykreslovací funkci. Pokud má prvek nastavenou vlastnost "border" na hodnotu "true", funkce "drawBorder" vykreslí rámeček kolem prvku. Pokud se jedná o textový prvek, funkce "drawText" vykreslí text v souladu s pozicí a velikostí prvku. Celý dokument reprezentovaný instancí třídy "UIDocument" lze poté vykreslit na displej voláním funkce "render" s ukazatelem na instanci "UIDocument" jako parametrem. Tato funkce zavolá funkci "drawElement" s kořenovým prvkem stromu prvků jako parametrem, která nakonec vykreslí celý dokument na displej.

## Wifi Manager



Wifi Manager je knihovna pro ESP32, která usnadňuje konfiguraci Wi-Fi připojení z mikrokontroléru. Tato knihovna umožňuje uživateli konfigurovat Wi-Fi připojení přímo z aplikace, což eliminuje potřebu ruční konfigurace pomocí sériového monitoru.

Wifi Manager funguje tak, že vytvoří vlastní Wi-Fi síť, ke které se uživatel může připojit a zadat své přihlašovací údaje pro připojení k cílové Wi-Fi síti. Tento proces se odehrává v prostředí, které je uživatelsky přívětivé a jednoduché na použití.

Po úspěšném připojení k cílové Wi-Fi síti se Wifi Manager ukončí a ESP32 se připojí k internetu. Toto usnadňuje konfiguraci Wi-Fi připojení z mikrokontroléru a umožňuje rychlé a snadné vytvoření aplikací pro IoT, které vyžadují připojení k internetu.

Odkaz na knihovnu [zde](https://github.com/tonyp7/esp32-wifi-manager).

## File System

ESP32 obsahuje souborový systém, který umožňuje ukládání a správu dat na flash paměti mikrokontroléru. Tento souborový systém se nazývá SPIFFS (SPI Flash File System) a je obvykle používán pro ukládání konfiguračních souborů, webových stránek a jiných dat v aplikacích pro IoT.

SPIFFS je implementován jako virtuální souborový systém, který se nachází na vrcholu flash paměti a používá jednoduché rozhraní API pro práci se soubory. Tento souborový systém umožňuje snadné ukládání a načítání dat, a to i při běhu aplikace.

SPIFFS je velmi užitečný pro vývoj aplikací pro IoT, které vyžadují ukládání dat, a to zejména v případech, kdy nejsou k dispozici jiné úložné médium, jako například microSD karty.

Ukázka použití ve funkci getFileAsString:

String getFileAsString(String path){

if(!SPIFFS.exists(path)){

Serial.println("file not found! path:" + path);

return "";

}

File f = SPIFFS.open(path, "r");

if (!f) {

Serial.println("file open failed");

}

String s = f.readString();

f.close();

return s;

}

Odkaz na knihovnu [zde](https://github.com/espressif/arduino-esp32/tree/master/libraries/SPIFFS)

## HTU21D

Tato třída z knihovny „SparkFunHTU21D“ umožňuje jednoduché ovládání sensoru HTU21D, který používáme pro získání teploty a vlhkosti v místnosti.

Knihovna pro senzor HTU21D poskytuje programátorům základní funkce pro čtení a zpracování dat z tohoto senzoru. Senzor HTU21D je digitální senzor pro měření vlhkosti a teploty vzduchu a má rozhraní I2C pro komunikaci s mikrokontrolérem.

Typické funkce, které knihovna poskytuje, jsou inicializace senzoru, nastavení různých módů měření (včetně rychlosti a rozlišení), čtení aktuální vlhkosti a teploty a výpočet různých dalších parametrů, jako je teplotní rosný bod.

Knihovna pro senzor HTU21D zjednodušuje programování mikrokontroléru a umožňuje rychlý přístup k datům ze senzoru. Díky této knihovně může programátor snadno integrovat senzor HTU21D do svých projektů a využívat jeho měřicích schopností pro různé aplikace.

Použití v programu je jednoduché stáčí vytvořit instanci třídy HTU21D, zavolání funkce „begin“. Pro získání teploty použijeme funkci třídy „readTemperature“, která nemá žádný vstupní parametr a vrací hodnotu typu float, která už je převedena na stupně celsia. Pro získání vlhkosti použijeme funkci třídy „readHumidity“, která nemá žádný vstupní parametr a vrací hodnotu typu float, která už je převedena na procenta vlhkosti.

Ukázka použití v programu:

//reading data from sensor

float humd = indoorSensor.readHumidity();

float temp = indoorSensor.readTemperature();

Odkaz na knihovnu [zde](https://github.com/sparkfun/SparkFun_HTU21D_Breakout_Arduino_Library).

## HTTPClient

httpClient je knihovna pro ESP32, která umožňuje mikrokontroléru komunikovat s webovými servery pomocí protokolu HTTP. Tato knihovna umožňuje mikrokontroléru odesílat HTTP požadavky a zpracovávat HTTP odpovědi.

HttpClient podporuje standardní metody HTTP, jako jsou GET, POST, PUT a DELETE, a také umožňuje nastavit vlastní hlavičky HTTP. Tato knihovna také umožňuje použití HTTPS připojení pomocí certifikátů.

HttpClient je užitečný pro vývoj aplikací pro IoT, které vyžadují přístup k internetu a komunikaci s webovými servery. Tato knihovna umožňuje mikrokontroléru získat data z webových stránek, odesílat data na webové servery a využívat různé webové API.

### REST API Request

REST API request je zpráva, kterou klient odesílá serveru, a obsahuje několik klíčových prvků, jako je URL adresa požadovaného zdroje, HTTP metoda, a možná také tělo požadavku v případě, že klient odesílá data na server.

**Metoda** - HTTP metoda určuje akci, kterou klient požaduje na zdroji na serveru. Nejběžnějšími metodami jsou GET, POST, PUT a DELETE.

**Cílová adresa** - cílová adresa (URL) specifikuje, na jaký zdroj na serveru se požadavek vztahuje. Adresa se skládá z protokolu, domény, portu (nepovinné) a cesty.

**Hlavičky** - hlavičky HTTP umožňují klientovi a serveru vyměňovat metadata spojená s požadavkem. Hlavičky obsahují informace o typu dat, jazyku, autentizaci a dalších.

**Tělo požadavku** - tělo požadavku obsahuje data, která klient odesílá na server. Toto tělo je nepovinné a může obsahovat různé typy dat, jako jsou JSON, XML nebo textové řetězce.

Ukázka použití HTTPClientu v programu:

HTTPClient client;

client.begin(PROTOKOL + HOST\_NAME + URL + APPID);

int http\_code = client.GET();

if(http\_code > 0){

String payload = client.getString();

//process payload

//...

}

Odkaz na knihovnu [zde](https://github.com/espressif/arduino-esp32/blob/master/libraries/HTTPClient/src/HTTPClient.h).

## fileSystemUtils.hpp

Soubor "fileSystemUtils.hpp" obsahuje funkce pro práci souborovým systémem na platformě Arduino pomocí knihovny "FS" a "SPIFFS". Soubor slouží jako pomocník pro ukládání a načítání dat spojených s meteorologickou stanicí.

Funkce "getFileAsString" slouží k načtení obsahu souboru na dané cestě jako řetězec. Funkce "getSavedWeatherData" načítá uložená data o počasí a ukládá je do předané struktury "StationData". Pokud tato data nejsou k dispozici, funkce "tryGetSavedWeatherData" se pokusí o načtení dat a uložení do "dataBuffer", ale v případě neúspěchu nevyvolá chybu. Funkce "saveWeatherData" slouží k uložení dat o počasí do souboru na předané cestě.

Celkově tedy soubor poskytuje užitečné funkce pro ukládání a načítání dat o počasí, což může být pro meteorologickou stanici velmi užitečné.

## jsonUtils.hpp

Tento header file obsahuje deklarace funkcí pro práci s JSON formátem dat. Konkrétně jsou zde deklarovány funkce pro konverzi řetězce ve formátu JSON na dynamický objekt JSON a pro konverzi dynamického objektu JSON na instanci struktury "StationData", která je používána v projektu jako reprezentace dat ze stanice.

Tyto funkce jsou důležité pro práci s daty v projektu, protože formát JSON se používá pro ukládání a zpracování dat ze stanice a struktura "StationData" obsahuje informace o aktuálních meteorologických podmínkách na stanici.

# Popis vývojového procesu

# Výsledky a zhodnocení

## Shrnutí dosažených cílů

## Zhodnocení průběhu projektu

## Možná zlepšení

# Závěr

Kapitola nazvaná povinně Závěr plní v práci SOČ dva úkoly. Za prvé musí stručně shrnout dosažené výsledky, což ovšem neznamená zopakování již dříve uvedených informací. Není potřeba znovu vypočítávat všechny dílčí výsledky, kterých jste ve své práci dosáhli. Místo toho stručně zopakujte cíl práce formulovaný v Úvodu, a volně na něj navažte. Např. „V úvodu práce jsem si jako cíl vytkla ověření čistoty studánek v okrese Bruntál. Tohoto cíle jsem dosáhla; zjistila jsem, že čistota studánek se ve srovnání s předchozím měřením zlepšila.“

Za druhé pak musí provést diskuzi dosažených výsledků. Formulujte také možné návaznosti práce jako inspiraci pro někoho, kdo by ve vaší činnosti chtěl pokračovat. V některých oborech je zvykem oddělit diskuzi do zvláštní kapitoly.

# Použitá literatura

BIERNÁTOVÁ, Olga, Jan SKŮPA. Bibliografické odkazy a citace dokumentů dle ČSN ISO 690 (01 0197) platné od 1. dubna 2011. [online]. c2004-2011 [cit. 2016-10-07]. Dostupné z: <http://citace.com/soubory/csniso690-interpretace.pdf>.

[1] RYBIČKA, Jiří, Petra ČAČKOVÁ a Jan PŘICHYSTAL. Průvodce tvorbou dokumentů. 1. vyd. Bučovice: Martin Stříž, 2011. ISBN 978-80-87106-43-3.

[2] …

nebo

1. RYBIČKA, Jiří, Petra ČAČKOVÁ a Jan PŘICHYSTAL. Průvodce tvorbou dokumentů. 1. vyd. Bučovice: Martin Stříž, 2011. ISBN 978-80-87106-43-3.

2. …

# Seznam obrázků a tabulek

[Obr. 1: Logo soutěže Středoškolská odborná činnost: Times New Roman, 10 b 7](#_Toc465598341)

[Tab. 1: Přehled parametrů jednotlivých stylů textu: Times New Roman, 10 b 7](#_Toc465598348)