

**Česká zemědělská univerzita v Praze**

**Technická fakulta**

**Katedra elektrotechniky a automatizace**



## **Diplomová práce**

**Tvorba inteligentního detektoru kouře a požáru**

**Ilia Kazakov**

**© 2023 ČZU v Praze**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

Ilia Kazakov

Informační a řídicí technika v agropotravinářském komplexu

Název práce

**Tvorba inteligentního detektoru kouře a požáru**

Název anglicky

**Creation of an intelligent smoke and fire detector**

---

## Cíle práce

Cílem práce je vytvoření inteligentního detektoru kouře přes síť WiFi, který bude umožňovat monitoring prostředí a následně zasílání emailů či SMS zpráv o alarmech. Zařízení bude doplněno o detekci zplodin hoření a měření změny teploty indikující požár. Zařízení bude připojeno k asistentu domácí automatizace a bude možné uživatele informovat pomocí SMS či emailem.

## Metodika

Prostudování hardwarových a softwarových možností řešení. Navržení několika variant provedení úlohy. Výběr nejvhodnější varianty s kritickým hodnocením návrhu. Specifikace funkcí modelu podle cílů práce.

## Doporučený rozsah práce

60stran, bez příloh

## Klíčová slova

Arduino, detektor, WiFi, smart, automatizace

---

## Doporučené zdroje informací

BELL, C A. *Beginning sensor networks with Arduino and Raspberry Pi*. [New York, New York]: Apress, 2013. ISBN 1430258241.

DENNIS, A K. *Raspberry Pi home automation with Arduino : automate your home with a set of exciting projects for the Raspberry Pi!*. Birmingham: Packt Publishing, 2013. ISBN 978-1-78439-920-7.

GARDNER, J W. – VARADAN, V K. – AWADELKARIM, O O. *Microsensors, MEMS, and smart devices*. Chichester: Wiley, 2001. ISBN 0-471-86109-.

GOODWIN, S. *Smart home automation with Linux and raspberry Pi*. New York: Apress, 2013. ISBN 978-1-4302-5887-2.

MORRISS, S B. *Automated manufacturing systems : actuators, controls, sensors, and robotics*. New York: Glencoe, 1995. ISBN 0028023315.

---

## Předběžný termín obhajoby

2021/2022 LS – TF

## Vedoucí práce

doc. Ing. Miloslav Linda, Ph.D.

## Garantující pracoviště

Katedra elektrotechniky a automatizace

Elektronicky schváleno dne 2. 6. 2021

**doc. Ing. Miloslav Linda, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2022

**doc. Ing. Jiří Mašek, Ph.D.**

Děkan

V Praze dne 23. 02. 2023

### **Čestné prohlášení**

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

V Praze dne \_\_\_\_\_

### **Poděkování**

Rád bych touto cestou poděkoval doc. Ing. Miloslavu Lindovi, Ph.D. za pomoc s výběrem tématu a podnětné připomínky při psaní této práce.

# Tvorba inteligentního detektoru kouře a požáru

## Abstrakt

Tato diplomová práce se zaměřuje na vytvoření inteligentního detektoru kouře a požáru, který umožní rychlou a spolehlivou detekci nebezpečných situací v prostředí. Detektor je vybaven technologiemi pro měření teploty a detekci zplodin hoření, což umožňuje rychlou reakci na případné požáry.

V rámci práce jsou zkoumány různé metody detekce požáru a jsou implementovány jednotlivé komponenty detektoru, včetně rozhraní domácí automatizace. Výsledkem je inovativní řešení, které je spolehlivé, flexibilní a snadno ovladatelné, které umožňuje odesílat emaily a SMS zprávy ohledně alarmů.

Detektor kouře navržený v této práci je vhodný jak pro domácí, tak i pro komerční a průmyslové aplikace. S jeho pomocí lze minimalizovat riziko požáru a zvýšit bezpečnost v prostředí.

**Klíčová slova:** Arduino, detektor, WiFi, smart, automatizace.

# **Creation of an intelligent smoke and fire detector**

## **Abstract**

This thesis focuses on the creation of an intelligent smoke and fire detector, which enables quick and reliable detection of dangerous situations in the environment. The detector is equipped with technologies for temperature measurement and detection of combustion by-products, allowing for rapid response to potential fires.

Various fire detection methods are examined within the scope of this work, and individual components of the detector are implemented, including a home automation interface that enables the sending of emails and SMS messages regarding alarms. The result is an innovative solution that is reliable, flexible, and easily controllable.

The smoke detector designed in this work is suitable for both domestic and commercial/industrial applications. Its use can help minimize the risk of fires and increase safety in the environment.

**Keywords:** Arduino, detector, WiFi, smart, automation.

# Obsah

<b>1 Úvod</b> .....	<b>12</b>
<b>2 Cíl práce a metodika</b> .....	<b>13</b>
2.1 Metodika .....	13
<b>3 Teoretická východiska</b> .....	<b>14</b>
3.1 Chování detektoru ve fázích požáru.....	14
3.2 Senzory.....	15
3.2.1 Plynový senzor MQ-2.....	15
3.2.1.1 Vnitřní struktura .....	16
3.2.1.2 Princip činnosti.....	18
3.2.1.3 Kalibrace modulu .....	18
3.2.1.4 Pinování senzoru .....	19
3.2.2 Senzor plamene YG1006.....	20
3.2.2.1 Provoz modulu.....	20
3.2.2.2 LED indikace.....	21
3.2.3 Teploměr DS18B20 .....	21
3.3 Wifi modul .....	23
3.3.1 Standardy IEEE 802.11.....	24
3.3.2 Modul ESP8266.....	25
3.3.3 Modul ESP32.....	28
3.4 SIM 800L .....	30
3.4.1 Komponenty GSM modulu.....	31
3.4.2 LED indikace .....	32
3.5 Vývojová deska.....	33
3.5.1 Konektivita.....	34
3.6 Vývojové prostředí.....	36
3.7 Blynk.....	37
<b>4 Vlastní práce</b> .....	<b>39</b>
4.1 Programování ESP32 v Arduino IDE .....	39
4.2 Schéma zapojení a návrh plošného spoje.....	42
4.3 Vývojový diagram.....	45
4.4 Blynk.....	47
4.4.1 Implementace Blynk knihoven v sketchu .....	49
4.5 Web rozhraní.....	51
4.6 Implementace kódu detektoru kouře a ohně .....	52



4.6.1	Monitorování kouře a plamenů.....	52
4.6.2	Monitorování teploty .....	53
4.6.3	Odesílání alarmů .....	54
4.7	Pouzdro .....	55
4.8	Indikace stavů zařízení pomocí LED diody .....	57
<b>5</b>	<b>Hodnocení výsledků .....</b>	<b>59</b>
<b>6</b>	<b>Závěr.....</b>	<b>63</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>64</b>
<b>8</b>	<b>Přílohy .....</b>	<b>66</b>

## Seznam obrázků

Obrázek 1:	MQ-2 plynový senzor.[6].....	16
Obrázek 2:	Externí komponenty snímače plynu MQ-2.[5] .....	16
Obrázek 3:	Vnitřní struktura s citlivým prvkem a spojovacími piny senzoru MQ-2.[5].....	17
Obrázek 4:	Citlivým prvkem senzoru MQ-2.[5] .....	17
Obrázek 5:	Vnitřní struktura citlivého prvku snímače plynu MQ-2.[5] .....	18
Obrázek 6:	Potenciometr nastavení citlivosti modulu snímače plynu MQ-2.[6] .....	19
Obrázek 7:	Pinování modulu snímače plynu MQ-2.[6].....	19
Obrázek 8:	Senzor plamene YG1006.[8].....	20
Obrázek 9:	Indikace modulu YG1006.[8] .....	21
Obrázek 10:	Struktura kódu 1-Wire.[10].....	22
Obrázek 11:	Typy senzorů DS18B20.[10] .....	22
Obrázek 12:	Wi-fi modul ESP8266.[18] .....	26
Obrázek 13:	Sériové rozhraní UART, I <sup>2</sup> C , SPI.[10].....	27
Obrázek 14:	Modul ESP32.[20].....	29
Obrázek 15:	Vstupy a výstupy modulu SIM800L.[23] .....	31
Obrázek 16:	Komponenty modulu SIM800L.[23] .....	32
Obrázek 17:	ESP32 DEVKIT.[24] .....	33
Obrázek 18:	ESP32 DEVKIT rozvržení pinů.[24] .....	34
Obrázek 19:	Aplikace Blynk.[28].....	37
Obrázek 20:	Přidání desek ESP32 do seznamu nabídek v prostředí Arduino IDE. ....	40
Obrázek 21:	Manažér desek v prostředí Arduino IDE. ....	40
Obrázek 22:	Výběr vhodné desky- ESP32 Dev Module. ....	41

Obrázek 24: Plošný spoj. ....	43
Obrázek 25: 3D náhledu plošného spoje. ....	44
Obrázek 26: Vývojový diagram detektoru.....	45
Obrázek 27: Plátno projektu detektorů kouře a požáru. ....	47
Obrázek 28: Položka výběr zařízení v aplikaci Blink.....	48
Obrázek 29: Získání autentizační kód v aplikaci Blynk. ....	49
Obrázek 30: Web rozhraní pro získání uživatelských dat. ....	52
Obrázek 31: Pouzdro pro detektor. ....	56
Obrázek 32: Přední strana detektoru.....	57

## Seznam tabulek

Tabulka 1: Srovnání parametrů standardů IEEE 802.11.[15].....	25
Tabulka 2: Přesnost detekce kouře. ....	59
Tabulka 3: Přesnost detekce plamene.....	59
Tabulka 4: Přesnost detekce změny teploty.....	60
Tabulka 5: Rychlost doručení alarmů.....	60
Tabulka 6: Stabilita detektoru.....	61
Tabulka 7: Náklady na součástky detektoru.....	61

## Seznam použitých zkratk

WiFi - Bezdrátová síť s vysokým přenosem dat

MOS - Kovoxydový polovodič

DC - Stejnosměrný proud

V - Volty - jednotka elektrického napětí

mW - Milivaty - jednotka výkonu

LPG - Zkapalněný petrolej

PPM - Částic na milion

LED - Dioda emitující světlo

VCC - Společný kladný napěťový zdroj

GND - Zemní kontakt

nm - Nanometry - jednotka délky

NPN - Negativní-pozitivní-negativní - typ bipolárního tranzistoru

ROM - Paměť pouze pro čtení

C - Kapacita - elektrická vlastnost součástky

mA - Miliampéry - jednotka elektrického proudu

IoT - Internet věcí

IEEE - Institut pro elektrické a elektronické inženýrství

AP - Přístupový bod

GHz - Gigahertz - jednotka frekvence

SPI - Sériové periferní rozhraní

UART - Univerzální asynchronní přijímač/vysílač

IP - Internetový protokol

HTTP - Hypertextový přenosový protokol

IPv4 - Internetový protokol verze 4

GPIO - Vstupně-výstupní piny s obecným účelem

RAM - Paměť s náhodným přístupem

SRAM - Statická paměť s náhodným přístupem

OTA - Bezdrátová aktualizace firmwaru

GSM - Globální systém pro mobilní komunikaci

IPX - Internetwork Packet Exchange - síťový protokol

A - Ampéry - jednotka elektrického proudu

UFL - Ultra malý koaxiální konektor

F - Farady - jednotka elektrické kapacity

DevKit - Vývojová sada

USB - Univerzální sériová sběrnice

DAC - Digitálně-analogový převodník

ADC - Analogově-digitální převodník

API - Rozhraní pro programování aplikací

DNS - Systém doménových jmen

# 1 Úvod

V dnešní době, kdy se snažíme co nejvíce chránit naše domovy a pracovní prostory před různými nebezpečími, je zásadní mít k dispozici spolehlivý způsob detekce požáru. V této diplomové práci se zaměříme na vytvoření inteligentního detektoru, který bude schopen monitorovat prostředí a v případě detekce požáru, kouře, zplodin hoření nebo změny teploty, bude uživatele informovat prostřednictvím SMS zpráv nebo emailů.

Samotnou práci je možné rozdělit do dvou hlavních částí. Jak část teoretická, tak část praktická jsou členěny do podkapitol zabývajících se řešením jednotlivých problematik. Teoretická část popisuje především výběr použitých komponentů a hardwarové vlastnosti snímačů. Praktická část je pak zaměřena na prostudování softwarových možností řešení realizace projektu a možnosti integrace zařízení s asistentem domácí automatizace.

Výsledkem práce bude spolehlivé zařízení pro detekci kouře a požáru, které bude uživatelům poskytovat včasné a spolehlivé informace o potenciálních nebezpečích v jejich domovech a pracovních prostorech.

## **2 Cíl práce a metodika**

Cílem práce je vytvoření inteligentního detektoru kouře přes síť WiFi, který bude umožňovat monitoring prostředí a následně zasílání emailů či SMS zpráv o alarmech. Zařízení bude doplněno o detekci zplodin hoření a měření změny teploty indikující požár. Zařízení bude připojeno k asistentu domácí automatizace a bude možné uživatele informovat pomocí SMS či emailem.

### **2.1 Metodika**

Prostudování hardwarových a softwarových možností řešení. Navržení několika variant provedení úlohy. Výběr nejvhodnější varianty s kritickým hodnocením návrhu. Specifikace funkcí modelu podle cílů práce.

### 3 Teoretická východiska

V této kapitole jsou představována teoretická východiska týkající se popisu senzorů použitých v této práci. Kromě toho jsou také uvedeny informace o wifi modulech, GSM modulu SIM 800L, vývojové desce a vývojovém prostředí, které jsou klíčové pro správné fungování celého systému. Konkrétně jsou popsány vnitřní struktura a princip činnosti plynového senzoru MQ-2, senzoru plamene YG1006 a teploměru DS18B20, jejich kalibrace a pinování. Dále jsou prezentovány standardy IEEE 802.11b/g/n a moduly ESP8266 a ESP32. Tato kapitola je důležitá nejen pro pochopení celého systému a jeho funkčnosti, ale také pro lepší pochopení dalšího výkladu tématu.

#### 3.1 Chování detektoru ve fázích požáru

Pro správné porozumění problematice detekce kouře a vznikajícího požáru je nejdůležitější znát chování požáru v jednotlivých stádiích jeho vývoje. Intenzita požáru se v průběhu času mění a požár má obvykle čtyři fáze, pokud není hašen. Délka každé fáze se může lišit a závisí na množství hořlavých látek, jejich požárně technických charakteristikách a podmínkách, které ovlivňují šíření požáru.

První fáze se nazývá fáze počátečního hoření a nastává, kdy požár teprve začíná. V této fázi se plameny postupně šíří po celé ploše hořlavých materiálů, které jsou v tomto případě nazývány hořlavým zatížením. Požár lze v této fázi ještě relativně snadno uhasit pomocí hasicího přístroje. [1]

Druhá fáze, fáze rozvoje požáru, následuje poté, co se plameny rozšíří a plně obklopí hořlavé materiály. Tato fáze trvá od okamžiku plného zahalení hořícími materiály plameny až po standardní rychlost jejich vypálení. [1]

Třetí fáze, vespělá fáze nebo fáze plného hoření, je obdobím nejintenzivnějšího hoření. V této fázi se plameny šíří velmi rychle a hořlavý materiál se vypaluje rychleji, než dokážeme reagovat. [1]

Poslední fáze je fáze uhašení, kdy se rychlost vypalování hořlavých látek začne postupně snižovat a materiály se ochlazují na původní teplotu (v tomto případě se jedná o střední objemovou teplotu, tedy průměr různých míst). [1]

Pro minimalizaci rizika požáru je důležité mít k dispozici hasicí přístroje, detektory kouře a sprinklery, a v případě požáru ihned kontaktovat místní hasičskou jednotku. Detektor bude fungovat ve třech fázích, přičemž každá fáze odpovídá jiné úrovni nebezpečí. V první

fázi detekuje kouřový senzor částice nečistot v ovzduší, které mohou být známkou začínajícího požáru, zatímco v druhé a třetí fázi senzor teploty detekuje náhlé zvýšení teploty, což je známka požáru.

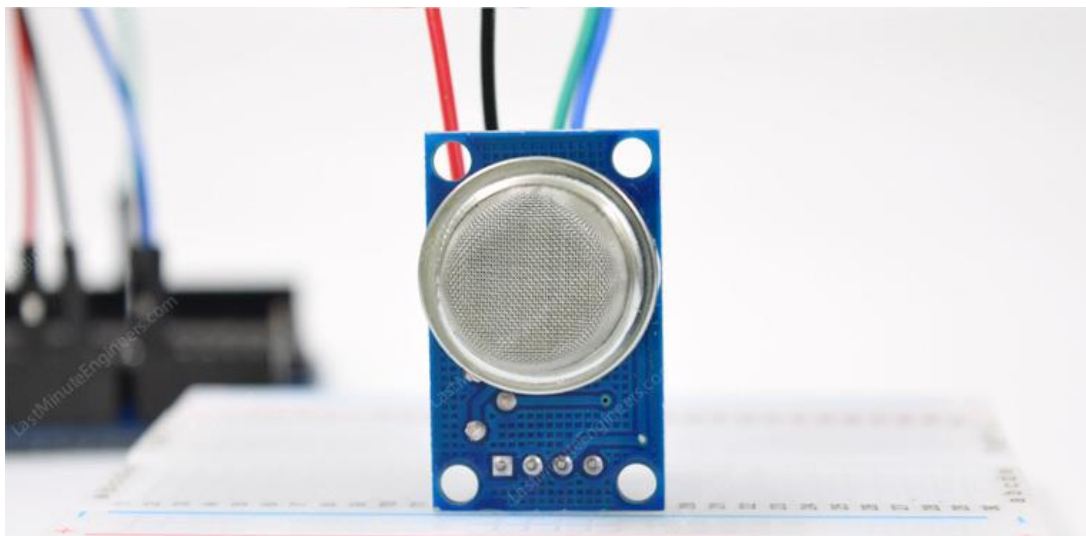
## **3.2 Senzory**

Pro efektivní fungování detektoru a získávání relevantních informací je nezbytné propojit ho s vhodným senzorem. Senzor je zařízení, které zachytává specifické údaje z prostředí a konvertuje je na srozumitelnou formu, jako je například binární hodnota High nebo Low. V současné době je na trhu široká škála senzorů, jejichž kvalita závisí na zamýšleném použití - od jednoduchých pro domácnosti až po sofistikované průmyslové nebo vojenské verze, které jsou přesnější a odolnější. Tyto vlastnosti ovšem mohou výrazně zvyšovat jejich nákupní cenu. Senzory se také liší v způsobu, jakým předávají zaznamenaná data. Mezi základní typy analogových, digitálních a sběrníkových senzorů byly vybrány následující senzory pro implementaci inteligentního detektoru kouře a požáru. [2,3]

### **3.2.1 Plynový senzor MQ-2**

Pro detekci plynu a kouře se používá senzor plynu MQ-2, který je spolehlivý a vhodný pro stanovení koncentrace různých plynů ve vzduchu. Tento senzor plynu MQ-2 je velmi oblíbený a je vhodný pro vytvoření systému kontroly kvality vzduchu v místnosti nebo pro včasnou detekci požáru. [6]

MQ-2 je jedním z nejvyužívanějších senzorů plynu z řady MQ senzorů. Je to senzor typu MOS (kov-oxid-polovodič), detekce se provádí na základě změny odporu citlivého materiálu, když se plyn dostane do kontaktu s tímto materiálem. [4]



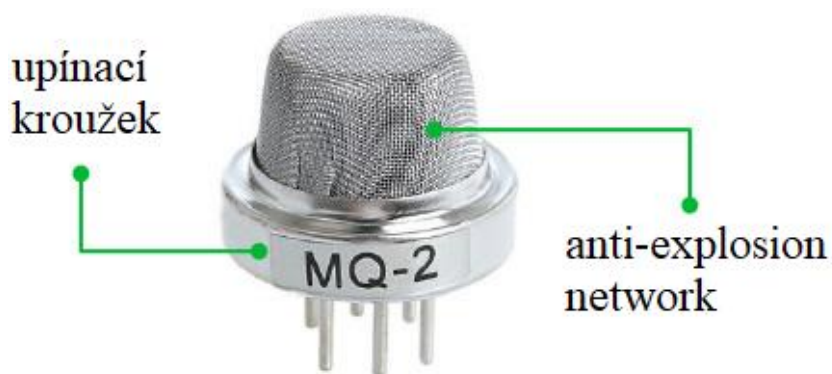
Obrázek 1: MQ-2 plynový senzor.[6]

MQ-2 plynový senzor funguje s napájecím napětím 5V DC a má spotřebu přibližně 800 mW. Tento senzor dokáže detekovat koncentrace různých plynů, včetně LPG, kouře, alkoholu, propanu, vodíku, metanu a oxidu uhelnatého, v rozmezí od 200 do 10 000 částic na milion (ppm). [5]

Při měření koncentrace plynů se nejčastěji používají dvě jednotky – procentuální koncentrace a částice na milion (ppm). Ppm se vyjadřuje jako poměr jednoho plynu k druhému, přičemž například 1000 ppm oxidu uhelnatého znamená, že z milionu molekul plynu je 1000 molekul oxidu uhelnatého a zbylých 999 000 molekul tvoří jiný plyn. [5]

### 3.2.1.1 Vnitřní struktura

Senzor je ve skutečnosti uzavřen ve dvou vrstvách tenké mřížky z nerezové oceli, která se nazývá "anti-explosion network". Zajišťuje, že topné těleso uvnitř senzoru nezpůsobí výbuch při hledání hořlavých plynů. [5]

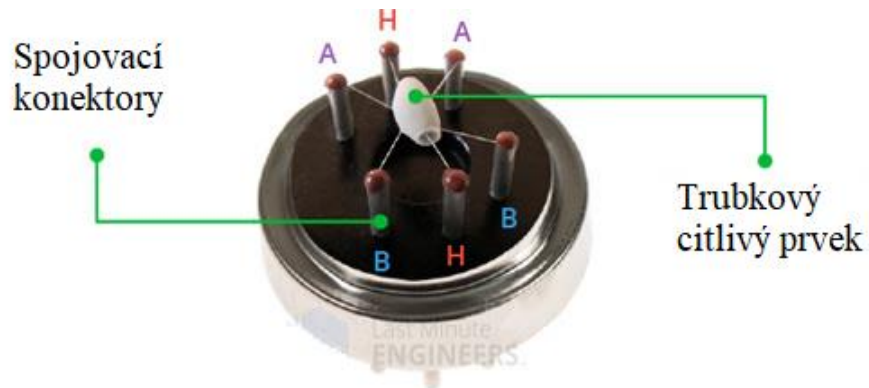


Obrázek 2: Externí komponenty snímače plynu MQ-2.[5]



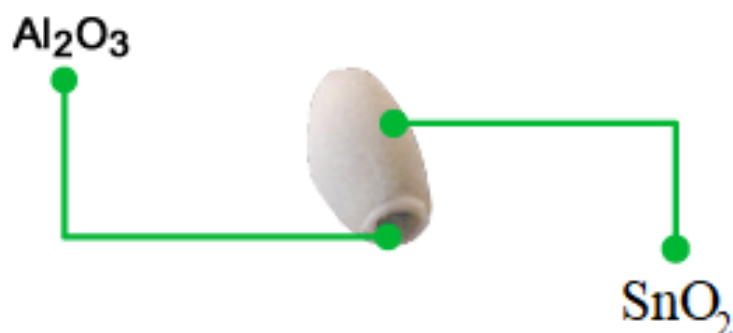
Senzor je uzavřen v anti-explozní síti z nerezové oceli, která zajišťuje bezpečnost a filtruje pevné částice, takže dovnitř mohou vstupovat pouze plyny. Anti-explozní síť je spojena s pouzdem pomocí měděného upínacího kroužku, aby senzor zůstal chráněn.[4,5]

Při odstraňování vnější mřížky se objeví hvězdicovitá struktura složená z citlivého prvku a šesti spojovacích konektorů. Dva vodiče (H) jsou zodpovědné za ohřev citlivého prvku a jsou spojeny s cívkou z nikl-chromového drátu, což je vodivá slitina. [5]



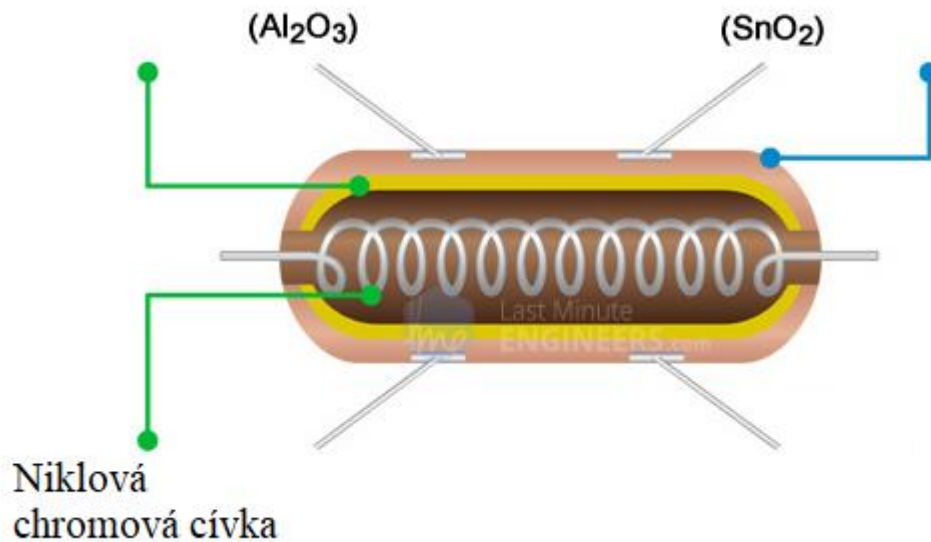
Obrázek 3: Vnitřní struktura s citlivým prvkem a spojovacími piny senzoru MQ-2.[5]

Čtyři zbývající piny (A a B) pro výstupní signály jsou propojeny platinovými vodiči s pouzdem citlivého prvku. Tyto vodiče přenášejí malé změny proudu, který prochází citlivým prvkem. Samotný citlivý prvek je trubkového tvaru a vyroben z keramiky na bázi oxidu hlinitého  $\text{Al}_2\text{O}_3$ . Jeho povrch je potažen oxidem cínu  $\text{SnO}_2$ , který je nejdůležitějším materiálem, protože je citlivý na hořlavé plyny. Keramický substrát slouží k efektivnímu ohřevu a udržování snímače na pracovní teplotě. [5]



Obrázek 4: Citlivým prvkem senzoru MQ-2.[5]

Topný systém senzoru zahrnuje niklovou chromovou cívku a keramiku na bázi oxidu hlinitého, které jsou zodpovědné za ohřev oblasti snímače na provozní teplotu. Senzorický systém je tvořen platinovými dráty a povlakem z oxidu cínu, které přenášejí malé změny proudu procházejícího citlivým prvkem a umožňují detekci hořlavých plynů. [4,5]



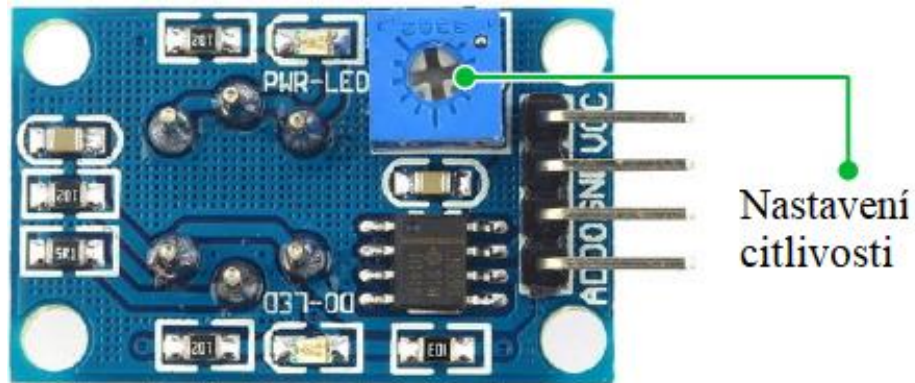
Obrázek 5: Vnitřní struktura citlivého prvku snímače plynu MQ-2.[5]

### 3.2.1.2 Princip činnosti

Když je oxid cínu zahříván na vysokou teplotu na vzduchu, adsorbuje kyslík na jeho povrchu. Když je však přítomen redukční plyn, reaguje s kyslíkem a snižuje jeho hustotu na povrchu oxidu cínu, čímž umožňuje průchod elektrického proudu senzorem. Koncentrace plynu ovlivňuje výstupní napětí, takže vyšší koncentrace plynu způsobí vyšší výstupní napětí a nižší koncentrace plynu nižší výstupní napětí. Analogový signál z senzoru plynu MQ-2 vstupuje do komparátoru LM393 pro digitalizaci. Vedle komparátoru se nachází malý potenciometr, který umožňuje nastavení citlivosti senzoru. [6]

### 3.2.1.3 Kalibrace modulu

Postup kalibrace zařízení zahrnuje umístění senzoru plynu vedle detekovaného kouře nebo plynu a následné otáčení potenciometrem až do doby, kdy se na modulu rozsvítí červená LED. Pokud je zapotřebí zvýšit citlivost senzoru, potenciometr je třeba otočit ve směru hodinových ručiček, zatímco pro snížení citlivosti se potenciometr otočí proti směru hodinových ručiček. [4,6]



Obrázek 6: Potenciometr nastavení citlivosti modulu snímače plynu MQ-2.[6]

Na modulu je umístěn komparátor, který neustále monitoruje analogový výstup (A0) a porovnává jej s prahovou hodnotou nastavenou potenciometrem. Pokud se dosáhne této hodnoty, digitální výstup (D0) se změní na vysokou logickou úroveň a kontrolka LED se rozsvítí. [4]

#### 3.2.1.4 Pinování senzoru



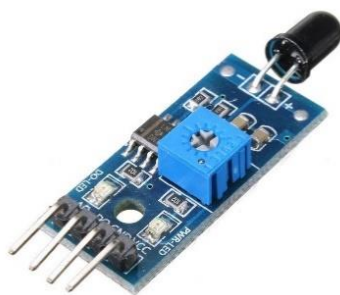
Obrázek 7: Pinování modulu snímače plynu MQ-2.[6]

- VCC poskytuje napájení modulu.
- GND země.
- D0 poskytuje digitální pohled na přítomnost hořlavých plynů.
- A0 poskytuje analogové výstupní napětí úměrné koncentraci kouře / plynu.

### 3.2.2 Senzor plamene YG1006

Existuje několik typů senzorů používaných pro detekci požárů, jako jsou ultrafialové senzory, infračervené senzory, infračervené termokamery, kombinované infračervené a ultrafialové senzory a další. [8]

V tomto projektu je využit infračervený senzor s infračervenou diodou YG1006, který slouží jako přijímač a umožňuje identifikaci zdrojů infračerveného záření, což umožňuje přímou detekci otevřeného ohně. Požární senzorový modul obsahuje fotodiodu (infračervený přijímač), rezistor, kondenzátor, potenciometr a komparátor LM393 v jednom integrovaném obvodu. Potenciometr umožňuje regulaci citlivosti senzoru, který má digitální a analogový výstup a pracuje s napětím mezi 3,3 V až 5 V DC. [7]



*Obrázek 8: Senzor plamene YG1006.[8]*

Infračervený požární senzor použitý v projektu je vybaven vysokorychlostním a vysoce citlivým NPN křemíkovým fototranzistorem, který může detekovat infračervené světlo o vlnové délce mezi 760 nm až 1100 nm a jeho detekční úhel je přibližně 60°. [7]

#### 3.2.2.1 Provoz modulu

Infračervená dioda v senzoru plamene YG1006 slouží k vysílání infračerveného záření směrem k oblasti, kterou má senzor sledovat. Fotodioda je připojena na vstup integrovaného obvodu LM393, který obsahuje komparátor. Ladicím rezistorem se nastavuje práh, který určuje citlivost požárního senzoru. Pokud senzor detekuje plamen s větším jasem než je



## 64-BIT LASERED ROM CODE Figure 6

8-BIT CRC		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (28h)	
MSB		LSB	MSB	LSB	MSB	LSB	

Obrázek 10: Struktura kódu 1-Wire.[10]

Senzor teploty DS18B20 je dostupný v dvou verzích. Jedna je dodávána v balení TO-92, což je podobné vzhledu tranzistoru. Druhá verze je vodotěsná sonda, což je užitečné, pokud je potřeba měřit teploty na velké vzdálenosti, pod vodou nebo pod zemí. [10]



Obrázek 11: Typy senzorů DS18B20.[10]

Uživatel si může nastavit rozlišení senzoru teploty na 9, 10, 11 nebo 12 bitů, ale výchozí nastavení je 12 bitů, což znamená přesnost měření 0,0625°C. Senzor je schopen měřit teploty v rozsahu od -55°C do +125°C s přesností ±0,5°C. Napájení senzoru může být zajištěno z napěťového zdroje s hodnotou mezi 3 V a 5,5 V a senzor během aktivního měření spotřebuje pouze 1 mA. [9]

### 3.3 Wifi modul

Pro sběr dat z různých senzorů v navrhovaném systému je nutné mít jednotku, která dokáže s těmito senzory komunikovat. Tato jednotka musí obsahovat rozhraní, které je kompatibilní s rozhraními na senzorech, musí mít dostatečný výkon pro zpracování dat v reálném čase a rozhraní pro komunikaci se zbytkem systému. Moduly ESP32 a ESP8266 od Espressif Systems jsou vhodným řešením, protože mají dostatečný výkon pro zpracování dat ze senzorů, nízkou spotřebu energie a vestavěný Wi-Fi modul, který umožňuje bezdrátové připojení senzorů k systému a připojení dostatečného počtu periferních rozhraní. K porozumění současnému stavu problematiky je důležité seznámit se s architekturou ESP8266 a ESP32, s vlastnostmi těchto modulů a s jejich vzájemnými rozdíly. Důležité je také seznámit se se standardy Wi-Fi a jejich vlastnostmi. [11]

Wi-Fi je technologie bezdrátového připojení k internetu a přenosu dat mezi různými zařízeními. Termín "Wi-Fi" je zkratkou pro "Wireless Fidelity" a byl vytvořen v roce 1999 konsorciem výrobců elektroniky. Wi-Fi se v současné době používá nejen v domácnostech, ale také v různých veřejných prostorách, jako jsou kavárny, hotely, letiště, nemocnice a další. Wi-Fi také umožňuje spojení různých zařízení v rámci Internetu věcí, což usnadňuje vytváření chytrých domácností a měst, inteligentních továren a dalších aplikací. Wi-Fi funguje na základě bezdrátového přenosu dat pomocí rádiových vln v oblasti ISM pásma. Tyto pásma jsou nechráněná a mohou být používána bez licencí, což umožňuje bezplatné používání Wi-Fi. Wi-Fi umožňuje rychlé bezdrátové připojení k internetu a propojení různých zařízení, jako jsou počítače, chytré telefony, tablety, tiskárny, televize a další. Zařízení s Wi-Fi mohou být připojena k routeru nebo přístupovému bodu, který umožňuje připojení k internetu. [12]

Wi-Fi standardy jsou definovány organizací IEEE Tyto standardy definují různé protokoly pro bezdrátový přenos dat, jako jsou 802.11a, 802.11b, 802.11g, 802.11n a 802.11ac. Každý z těchto standardů se liší v rychlosti přenosu dat, dosahu signálu a dalších parametrech. [13]

Access point (AP) a klientská zařízení jsou základními prvky bezdrátových sítí Wi-Fi sítí. AP je zařízení, které umožňuje bezdrátové připojení k síti a zajišťuje propojení mezi klientskými zařízeními a sítíovou infrastrukturou. AP obvykle vysílá Wi-Fi signál, který je detekován a přijat klientskými zařízeními. Připojením klientského zařízení k AP může uživatel přistupovat k síťovým zdrojům, jako jsou internetové služby, sdílené soubory,

tiskárny a další. Klientské zařízení, také označováno jako stanice, se připojuje k AP a využívá jeho bezdrátovou konektivitu k připojení k síti. Klientská zařízení mohou být notebooky, smartphony, tablety, chytré televize, herní konzole a další. Klientská zařízení přijímají signál od AP a používají ho k přenosu dat do a z internetu a dalších síťových zdrojů.[12,13]

Zabezpečení přenosu dat pomocí Wi-Fi je velice důležité, neboť nezabezpečené sítě jsou citlivé na útoky hackerů, kteří mohou přistupovat k datům uživatelů bez oprávnění. K zajištění bezpečnosti Wi-Fi se používají různé protokoly, jako například WPA2 nebo WPA3. Existují také různé metody připojení k Access pointu Wi-Fi, z nichž první způsob je otevřený a nešifrovaný přístup. Nicméně existují také autentizační protokoly, které zajišťují bezpečný přístup. Původně byl pro standard IEEE 802.11 navržen autentizační protokol WEP, který byl zodpovědný jak za autentizaci přístupu, tak za šifrování. WEP používal klíč o délce pouze 40 bitů, což se ukázalo jako velmi náchylné k útokům hrubou silou. Proto mnoho výrobců navrhlo rozšíření klíče na 104 bitů. Samotné šifrování bylo založeno na proudové šifře RC4.[14]

Druhým protokolem pro přístup na Access point je protokol WPA. Tento protokol byl vyvinut jako náhrada za svého předchůdce, protokol WEP. WPA se poprvé objevil jako součást standardu IEEE 802.11i, který řešil bezpečnostní otázky v bezdrátových sítích. Hlavním problémem předchozího protokolu WEP byly útoky hrubou silou, které byly vyřešeny díky novému algoritmu TKIP používanému u WPA. Tento algoritmus generuje klíč o délce 128 bitů pro každý přenášený paket zvlášť. Nicméně, protože WPA nezahrnoval celý standard 802.11i a vzhledem k dalším úpravám, byl vytvořen protokol WPA2, který již plně pokrývá tento standard a obsahuje několik vylepšení oproti WPA. [14]

### **3.3.1 Standardy IEEE 802.11**

Dále se zaměřím na několik standardů IEEE 802.11, přičemž pouze ty, které jsou relevantní pro moduly ESP32 a ESP8266 tedy IEEE 802.11b/g/n, následující informace budou obsahovat jenom základní parametry. Je důležité si uvědomit, že tyto standardy se liší v některých klíčových vlastnostech, které mohou ovlivnit jejich použití v konkrétních aplikacích. [15]

Standard IEEE 802.11b byl přijat v roce 1999 a umožňoval přenosovou rychlost až 11 Mb/s při využití frekvence 2,4 GHz. S nástupem technologie IEEE 802.11g v roce 2003 se



zvýšila rychlost přenosu až na 54 Mb/s a byla zajištěna kompatibilita s původním standardem 802.11b. [15]

V roce 2009 byl přijat standard IEEE 802.11n, který přinesl řadu vylepšení. Tento standard umožňuje přenos dat rychlostí až 600 Mb/s a zlepšuje pokrytí signálem díky použití technologie MIMO (multiple-input multiple-output). IEEE 802.11n je také kompatibilní s předchozími standardy 802.11b/g, což umožňuje snadnou integraci nových zařízení do stávající sítě. [15]

Standardy IEEE 802.11b/g/n jsou důležité pro bezdrátové sítě a jsou používány po celém světě. Tyto standardy umožňují snadné a pohodlné připojení k internetu a síťovým zařízením bez nutnosti kabelových propojení. Díky technologii MIMO jsou tyto standardy schopny poskytnout rychlé a spolehlivé připojení s dobrým pokrytím signálem. S rozvojem technologií se mohou očekávat další vylepšení standardů pro bezdrátové sítě Wi-Fi. [15]

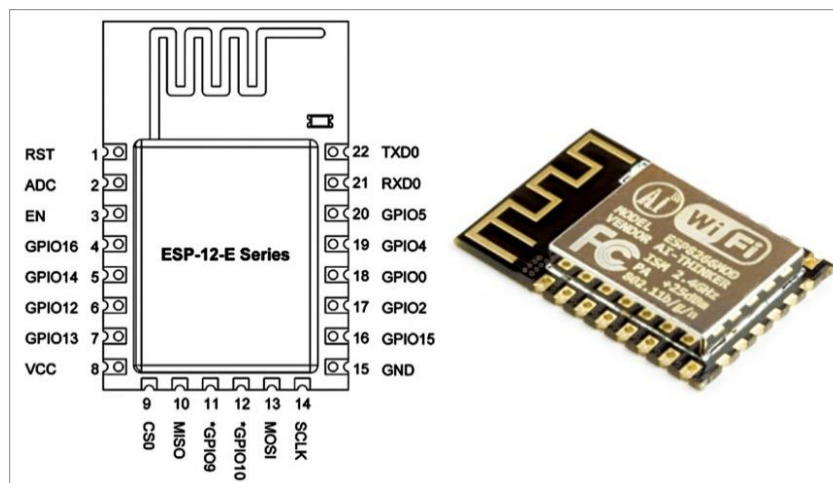
*Tabulka 1: Srovnání parametrů standardů IEEE 802.11.[15]*

<b>Standard</b>	<b>Frekvenční pásmo (GHz)</b>	<b>Maximální rychlost v laboratorním prostředí (Mbit/s)</b>	<b>Maximální rychlost v reálném prostředí bez okolních rušení (Mbit/s)</b>
802.11b	2,4	11	6
802.11g	2,4	54	25
802.11n	2,4/5	150/600	50/150

### **3.3.2 Modul ESP8266**

ESP8266 je malý, výkonný a cenově dostupný Wi-Fi modul, který umožňuje připojení k internetu prostřednictvím Wi-Fi sítě. Tento modul byl vyvinut firmou Espressif Systems a je vhodný pro využití v různých projektech Internetu věcí. [18]

ESP8266 obsahuje vestavěný 32bitový mikrokontroler s možností volby hodinové frekvence 80 nebo 160 MHz a Wi-Fi anténu, což z něj činí samostatné zařízení pro připojení k Wi-Fi sítím. Modul podporuje různé Wi-Fi protokoly, jako jsou 802.11b, 802.11g a 802.11n. ESP8266 lze použít jako samostatnou Wi-Fi síť nebo jako další zařízení v existující síti. [16]



Obrázek 12: Wi-fi modul ESP8266.[18]

Modul ESP8266 může sloužit také jako Wi-Fi adaptér, což umožňuje připojení k bezdrátovému přístupu do počítačové sítě nebo na internet prostřednictvím libovolného návrhu obsahujícího mikrokontroler. Propojení mezi těmito prvky je snadné a zprostředkováno pomocí sériových rozhraní, jako je například SPI, UART nebo I<sup>2</sup>C. [16]

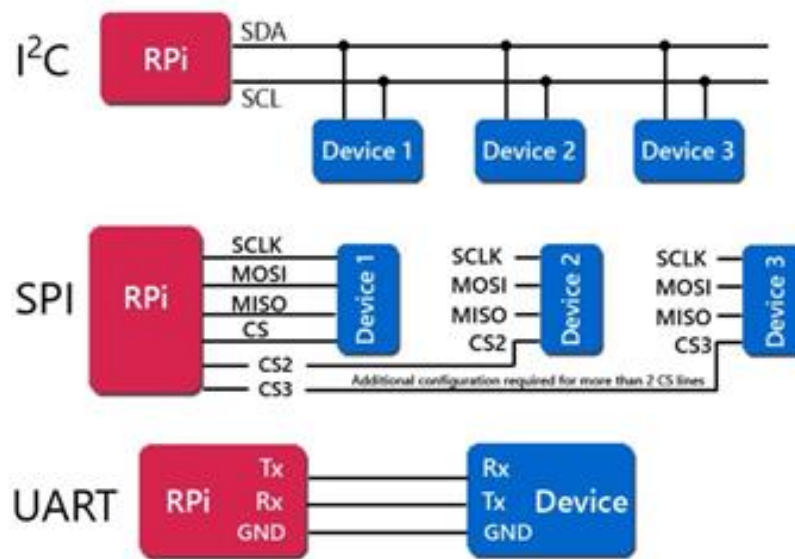
SPI, UART a I<sup>2</sup>C jsou tři různé sériové sběrnice, které se často používají pro přenos dat mezi mikrokontrolery a dalšími periferiemi. I když tyto sběrnice slouží ke stejnému účelu, existují mezi nimi některé rozdíly.[19]

SPI je plně duplexní sériová sběrnice, což znamená, že může současně přenášet i přijímat data. Sběrnice SPI vyžaduje minimálně čtyři vodiče (CLK, MOSI, MISO a CS), což z ní dělá relativně rychlou a výkonnou sběrnici, vhodnou pro přenos velkých objemů dat mezi různými periferiemi. [19]

UART je asynchronní sériová sběrnice, která se obvykle používá pro přenos malých objemů dat. UART využívá jen dva vodiče (RX a TX) pro přenos dat a nepoužívá žádné hodinové signály pro synchronizaci přenosu dat. Toto znamená, že přenos dat není tak rychlý jako u sběrnice SPI, ale je mnohem jednodušší na použití. [19]

I<sup>2</sup>C je synchronní sériová sběrnice, která používá pouze dva vodiče (SDA a SCL) pro přenos dat mezi periferiemi. I<sup>2</sup>C se často používá pro připojení menších periferií, jako jsou senzory, displeje a paměti. Je pomalejší, než SPI, ale méně složitý na použití a má větší rozsah adresování. [19]

Hlavní rozdíl mezi těmito protokoly spočívá v tom, jakým způsobem přenášejí data, jaké dráty používají a pro jaký účel jsou určeny.



Obrázek 13: Sériové rozhraní UART, I<sup>2</sup>C, SPI.[10]

Kromě výše zmíněných rozhraní má toto zařízení také integrovaný 10bitový A/D převodník, TCP/IP zásobník, fázový závěs, různé regulátory a jednotky na řízení spotřeby, pulzně šířkovou modulaci nebo i rozhraním I<sup>2</sup>S. [17]

Tento modul také nabízí řadu pinů pro různé účely, včetně GPIO pinů, vstupů analogového signálu (ADC), I/O pinů pro komunikaci s periferiemi. Modul lze napájet z napětí v rozmezí od 2,5 do 3,6 V a podporuje Wi-Fi s frekvencí 2,4 GHz. Wi-Fi má podporu pro šifrování WPA a WPA2 s AES nebo WEP, aby byla zajištěna bezpečnost dat. ESP8266 také podporuje funkce jako OTA (over the air) aktualizace firmwaru a snadné propojení s dalšími zařízeními v síti. [16]

ESP8266 má významnou nevýhodu v podobě omezené velikosti RAM, zhruba polovina z ní je využívána Wi-Fi modulem. Programování této Wi-Fi části také přináší problémy, protože pro udržení spojení je vyžadována neustálá programová obsluha. Tento požadavek může omezovat řízení programového kódu pro regulaci. [17]

### 3.3.3 Modul ESP32

ESP32 je vylepšená verze Wi-Fi modulu ESP8266, vyvinutá firmou Espressif Systems. Podobně jako ESP8266, je i ESP32 určen pro použití v aplikacích Internetu věcí, a nabízí mnoho vylepšení oproti svému předchůdci. [20]

ESP32 nabízí rychlejší Wi-Fi s přenosovou rychlostí až 150 Mbps a hlavní inovací je použití dvoujádrového procesoru Tensilica Xtensa LX6, s možností dosáhnout frekvence až 2x240 MHz. Jádra procesoru mohou pracovat nezávisle na sobě nebo společně pro větší výpočetní výkon. Jedno jádro slouží k programování, zatímco druhé se věnuje Wi-Fi komunikaci. Navzdory těmto vylepšením má ESP32 nižší spotřebu energie než jeho předchůdce ESP8266. Modul je navržen tak, aby byl co nejvíce robustní, přizpůsobivý, spolehlivý a výkonný pro radiotechniku. Především se hodí pro mobilní aplikace a aplikace internetu věcí, ale také pro vývoj nositelné elektroniky. [21]

ESP32 využívá operační systém freeRTOS a síťový protokol LwIP, což umožňuje bezproblémovou komunikaci s dalšími zařízeními v síti. Aby byla zajištěna vysoká úroveň bezpečnosti, je zařízení vybaveno bezpečným OTA nahráváním firmware, což umožňuje snadnou aktualizaci firmware i po nasazení zařízení v reálném provozu. Tento prvek je zejména užitečný pro zařízení, která jsou umístěna na těžko dostupných místech nebo v místech s omezeným přístupem. Aktualizace firmware jsou rychlé, snadné a uživatelé se nemusí obávat o bezpečnost svých dat. [21]

Modul má celkem 38 pinů, z nichž některé jsou určeny pro napájení a uzemnění, ale další jsou důležité pro specifické funkce, jako jsou přijímač a vysílač rozhraní UART. Některé piny jsou připojeny k integrované flash paměti na modulu, takže není doporučeno je používat k jiným účelům. [21]



Obrázek 14: Modul ESP32.[20]

ESP32 podporuje Wi-Fi protokoly 802.11b/g/n, stejně jako Bluetooth a Bluetooth Low Energy (BLE). Modul také podporuje různé režimy vysílání, včetně vysokého výkonu a nízkého výkonu, což umožňuje úsporu energie. [21]

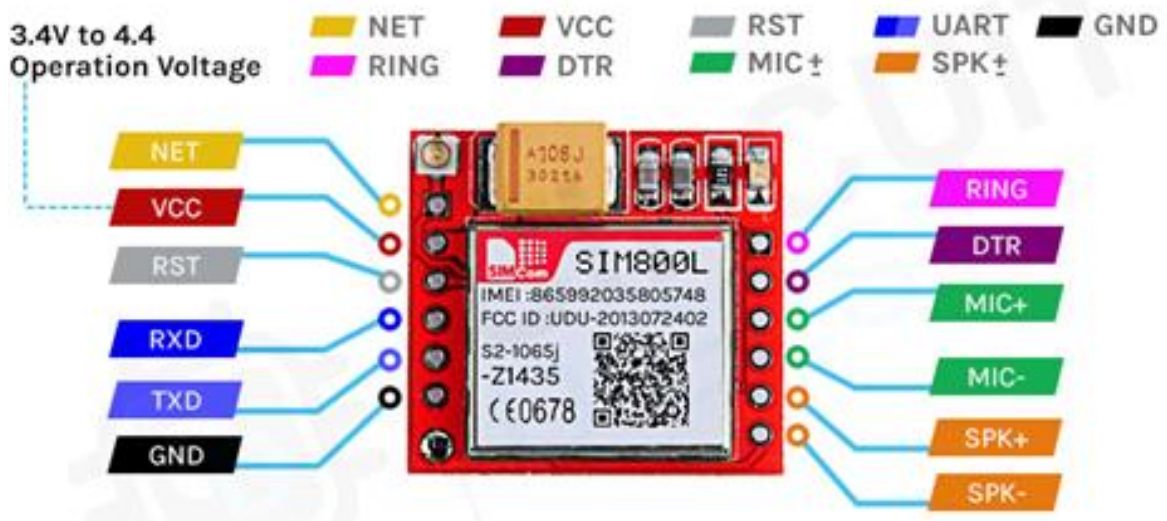
ESP32 má také větší paměťové možnosti než ESP8266. Modul má až 512 kB SRAM a až 4 MB flash paměti pro ukládání programů a dat. ESP32 také podporuje externí paměťová média, jako jsou microSD karty. [21]

### 3.4 SIM 800L

V cíli této práce je specifikováno, že detektor musí umět nejen posílat e-maily, ale také zasílat SMS zprávy na telefonní číslo. Možností posílání SMS pomocí internet služeb je mnoho, nicméně v případě detektoru bude k dispozici GSM modul pro zasílání SMS zpráv.

SIM800L je kompaktní modul, který byl vyvinut společností SIM Technology Group Limited z Číny. Tento modul komunikuje s mikrokontrolérem nebo PC pomocí AT příkazů po sériovém portu. Díky své podpoře všech frekvenčních rozsahů pásma GSM, včetně rozšířeného pásma EGSM, modul umožňuje přenos dat, SMS a MMS. Modul je také vybaven piny pro připojení reproduktoru a mikrofonu, což umožňuje odposlech objektu. Pro napájení modulu lze použít LiPo baterii s napětím mezi 3,6 V a 4,4 V DC nebo integrovaný regulovaný stabilizátor LM317 s odporovým děličem. Modul je také vybaven IPX konektorem pro připojení antény, a obsahuje drátovou anténu. LED dioda signalizuje stav konektivity, a rychle blikající LED dioda označuje silný signál sítě, zatímco pomalu blikající LED dioda značí špatně dostupný signál. Tento modul se obvykle používá v mobilních telefonech a bezpečnostních systémech díky své malé velikosti a příznivé ceně. Je třeba však dávat pozor na jeho velký proudový odběr, který může dosáhnout až 2 A. [22]

Pro snadnou konfiguraci podporuje modul automatickou detekci přenosové rychlosti sériového portu. Pro provoz je nutné připojit externí anténu a k dispozici jsou dvě varianty antén, z nichž je spirálová anténa součástí dodávky. Pokud chcete umístit anténu dále od modulu, je možné připojit externí anténu k UFL konektoru na modulu. Modul je kompatibilní s jakoukoli SIM kartou podporující 2G. [23]



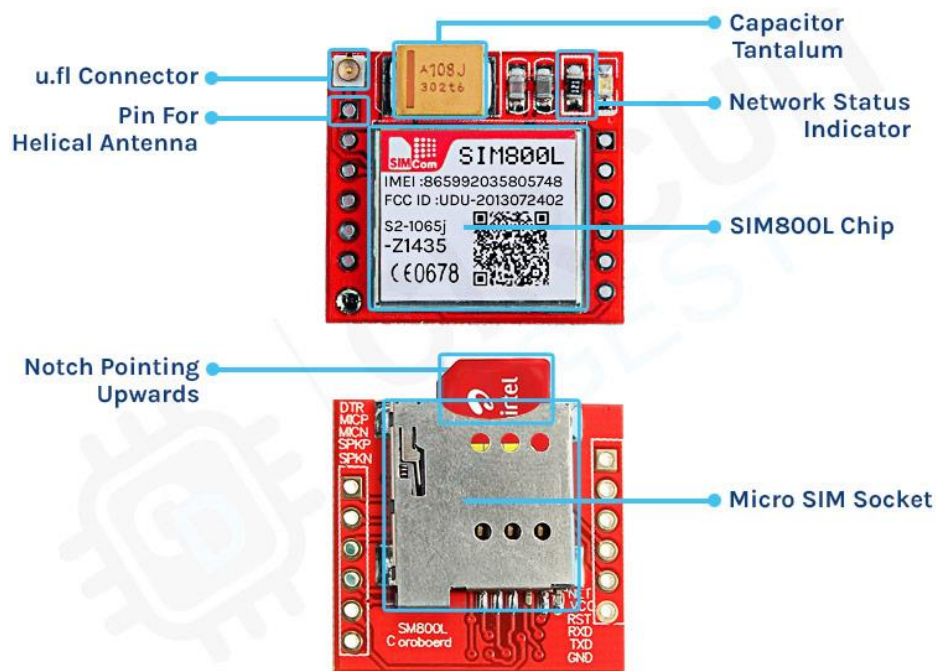
Obrázek 15: Vstupy a výstupy modulu SIM800L.[23]

Vstupy a výstupy modulu SIM800L. [23]

- RING (není označeno na PBC, první čtvercový pad) - LOW stav při příjmu hovoru
- DTR – režim spánku, výchozí stav v režimu HIGH (modul v režimu spánku, zakázaná sériová komunikace) - po nastavení v režimu LOW se modul probudí
- MICP, MICN – mikrofon (P + / N -)
- SPKP, SPKN – reproduktor (P + / N -)
- NET – drátková anténa
- VCC – napájecí napětí
- RESET – resetování
- RXD – sériová komunikace
- TXD – sériová komunikace
- GND – zem

### 3.4.1 Komponenty GSM modulu

SIM800L je modul pro připojení k mobilní síti GSM, který je malý a snadno použitelný. Modul se skládá z několika komponent, které jsou přehledně znázorněny na přiloženém obrázku. [22]



Obrázek 16: Komponenty modulu SIM800L.[23]

Na přední straně modulu se nachází UFL konektor pro připojení antény a samotný modul SIM800L. Dále je zde umístěn 1 k $\Omega$  rezistor pro omezování proudu LED, který slouží k signalizaci stavu konektivity. Na této straně modulu se nachází také velký tantalový kondenzátor o kapacitě 100uF a napětí 16V, který slouží k filtrování napájení. [22]

Na zadní straně modulu se nachází držák určený pro vložení SIM karty. Aby SIM karta mohla správně fungovat, stačí ji jednoduše zasunout do držáku na modulu. [22]

### 3.4.2 LED indikace

Modul SIM800L má na přední straně LED indikátor, který signalizuje různé stavy modulu. Pokud LED bliká každou sekundu, znamená to, že modul byl zapnut, ale ještě nenavázal spojení s mobilní sítí. Pokud LED bliká každé dvě sekundy, znamená to, že GPRS připojení je připraveno. Když LED bliká každé tři sekundy, signalizuje to, že modul je připojen k síti a je schopen odesílat a přijímat hlas a SMS zprávy. [23]



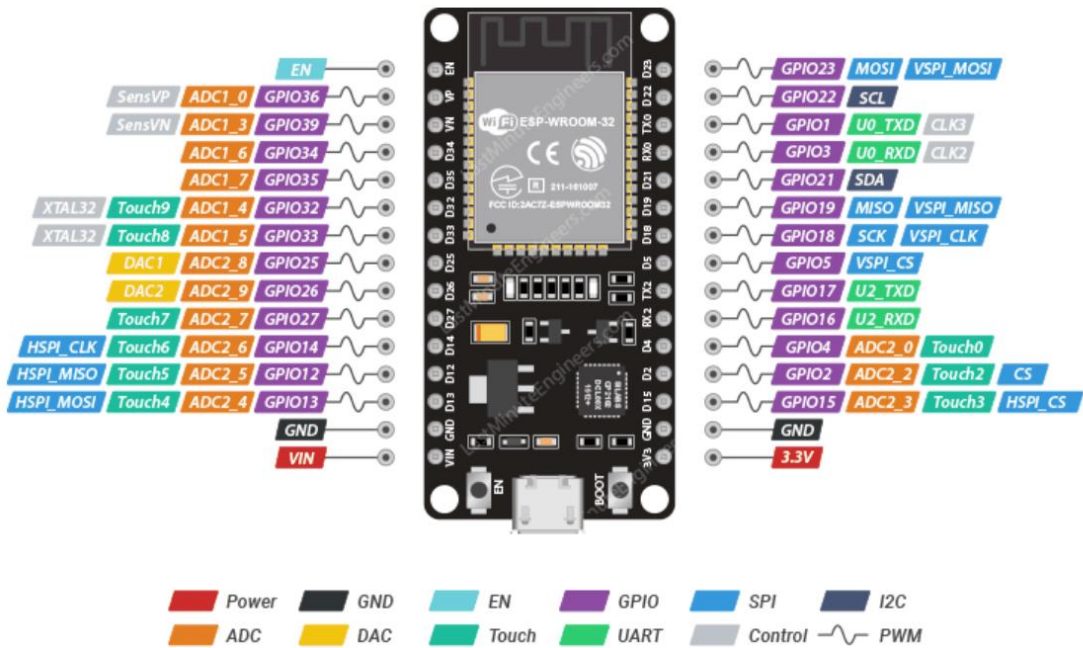
### 3.5 Vývojová deska



*Obrázek 17: ESP32 DEVKIT.[24]*

ESP32 DEVKIT V1 je vývojová deska založená na čipu ESP32, která byla vyvinuta společností Espressif pro vývoj internetu věcí. Tato deska nabízí mnoho funkcí a rozhraní, což ji činí velmi vhodnou pro vývoj IoT projektů, které vyžadují vysokou úroveň konektivity, výkonu a nízkou spotřebu energie. Díky architektuře ESP32 vývojová deska obsahuje dva jádra s frekvencí až 240 MHz, 520 KB SRAM a 4 MB flash paměti, což poskytuje dostatek výkonu a paměti pro různé projekty. Deska je vhodná pro různé Arduino projekty, které vyžadují bezdrátové připojení pomocí WiFi nebo Bluetooth bez potřeby přídatných modulů.[24]

### 3.5.1 Konektivita



Obrázek 18: ESP32 DEVKIT rozvržení pinů.[24]

Vývojová deska ESP32 DEVKIT V1 obsahuje následující prvky:

- ESP32-WROOM-32 modul s Wi-Fi a Bluetooth funkcionalitou
- Převodník USB-UART s čipem CP2102 umožňuje spojení modulu ESP 32-WROOM s USB portem počítače. Při připojení k počítači se platforma ESP32 DevKit rozpoznává jako virtuální COM port.
- Micro-USB konektor je určen pro programování a napájení platformy ESP 32 DevKit pomocí počítače.
- Indikátor napájení platformy
- Uživatelská LED na 2. pinu mikrokontroléru. Při nastavení hodnoty "vysoké úrovně" se LED zapne, při "nízké" se vypne.
- Tlačítko RESET je určeno pro ruční resetování programu - analogové tlačítko RESET běžného počítače.
- Tlačítko BOOT slouží pro ruční přepnutí modulu do režimu programování.
- Lineární snižující regulátor napětí AMS1117-3.3 zajišťuje napájení mikrokontroléru. Výstupní napětí 3,3 voltu s maximálním proudem 1 A.[25]

Napájecí piny:

- VIN: Pin pro připojení externího zdroje napětí v rozmezí 5 až 14 voltů.

- 3V3: Pin z napětového stabilizátoru s výstupem 3,3 voltu a maximálním proudem 1 A. Regulátor zajišťuje napájení modulu ESP32-WROOM.
- GND: Zemní piny.[25]

Vstupy/výstupy:

- Digitální vstupy/výstupy: 21 pinů 1–5, 12–19, 21–23, 25–27, 32 a 33. Piny s obecným použitím mohou být nastaveny jako vstup nebo výstup. Logická úroveň jedničky je 3,3 V, nuly je 0 V. Maximální výstupní proud je 12 mA.
- Digitální vstupy: 4 piny 34–36 a 39. Vstupy s obecným použitím. Může být nastaven pouze jako vstup.
- PWM: všechny vstupy/výstupy. Umožňuje výstup analogových hodnot jako PWM signál s rozlišením 16 bitů. Maximální počet kanálů je 16.
- ADC: 15 pinů 2, 4, 12–15, 25–27, 32–36 a 39. Umožňuje digitální reprezentaci analogového napětí s rozlišením 12 bitů.
- DAC: piny 25(DAC1) a 26(DAC2). Analogový výstup digitálně-analogového převodníku, který umožňuje vytvořit 8bitové úrovně napětí. Výstupy lze použít jako audio výstup. [25]

Rozhraní:

- I<sup>2</sup>C
- SPI
- UART/Seriál
- I<sup>2</sup>S. [25]

### 3.6 Vývojové prostředí

ESP32-DevKitC lze programovat pomocí různých vývojových prostředí, jako jsou Arduino IDE, PlatformIO a ESP-IDF. Arduino IDE a PlatformIO jsou ideální pro začátečníky, kteří se s vývojem embedded systémů teprve seznamují, a poskytují snadno použitelné grafické rozhraní pro psaní a nahrávání kódu na desku. ESP-IDF je vývojové prostředí vyvinuté společností Espressif Systems pro pokročilé uživatele, které poskytuje kompletní sadu nástrojů pro vývoj a ladění aplikací pro ESP32.[24]

Pro úspěšný vývoj aplikací s ESP32 DEVKIT V1 je také důležité mít znalosti programování v jazyce C nebo C++, alespoň základní znalosti elektroniky a schopnost číst datasheety. Pro vývoj je k dispozici také mnoho knihoven a příkladů kódu, které usnadňují vývoj aplikací pro různé účely, jako je ovládání LED diod, měření teploty a vlhkosti, čtení a zápis dat na microSD karty a další. [25]

Arduino IDE je vývojové prostředí určené pro programování desek Arduino a dalších mikrokontrolérů s podobnou architekturou. IDE zahrnuje textový editor, který umožňuje psát a upravovat kód programů, a také řadu nástrojů pro nahrávání kódu do desky a ladění programů. [26]

Jednou z největších výhod Arduino IDE je jeho jednoduchost a přívětivost pro začátečníky. IDE je intuitivní a snadno ovladatelné, což z něj činí skvělou volbu pro lidi bez předchozích zkušeností s programováním mikrokontrolérů. Základní funkce IDE zahrnují barevné zvýrazňování syntaxe, automatické doplňování kódu a schopnost vytvářet projekty s více soubory. [26]

Arduino IDE podporuje mnoho druhů desek a mikrokontrolérů, což umožňuje programovat různé projekty s různými hardwarovými nároky. IDE také obsahuje sadu standardních knihoven, které usnadňují vývoj konkrétních aplikací a projektů. [26]

Další výhodou Arduino IDE je jeho kompatibilita s mnoha operačními systémy, včetně Windows, macOS a Linux. To umožňuje používat IDE na široké škále zařízení a systémů. [26]

Vývojové prostředí Arduino IDE je zdarma ke stažení a použití, což z něj činí cenově dostupnou volbu pro všechny, kdo se chtějí naučit programovat mikrokontroléry nebo vyvíjet své vlastní projekty. [26]

### 3.7 Blynk

Blynk je výkonným nástrojem pro vytváření chytrých zařízení a může sloužit jako asistent domácí automatizace. Tato platforma umožňuje vývojářům a nadšencům v oblasti IoT vytvářet mobilní aplikace pro ovládání a monitorování zařízení. [27]

Jednou z hlavních výhod použití Blynk v inteligentní domácnosti je jeho pohodlné a intuitivní rozhraní, které umožňuje snadné ovládání zařízení. Blynk umožňuje uživatelům vytvářet mobilní aplikace pro Android a iOS pomocí intuitivního drag-and-drop rozhraní bez nutnosti programovat. Uživatelé mohou také využít Blynk knihovny a API pro vytváření vlastních vývojových nástrojů a integraci s jinými platformami pro IoT. Blynk také poskytuje cloudové řešení pro ukládání a sdílení dat ze senzorů a zařízení, a také umožňuje vzdálenou správu zařízení pomocí mobilní aplikace. Blynk nabízí mnoho různých widgetů, jako jsou tlačítka, grafy a displeje, které lze snadno přidávat do aplikace. Tyto widgety umožňují uživatelům ovládat a sledovat různé funkce svých zařízení IoT, jako jsou například světla, klimatizace, senzory teploty a bezpečnostní systémy. [27]



Obrázek 19: Aplikace Blynk.[28]

Kromě toho je Blynk bezplatným nástrojem, což ho dělá více dostupným ve srovnání s jinými placenými asistenty domácí automatizace, jako je Google Assistant nebo Xiaomi AI. Další výhodou Blynk je jeho flexibilita a možnost integrace s jinými platformami a službami. [28]

Je však třeba poznamenat, že Blynk je nejvhodnější pro malé a vzdělávací projekty na Arduino, nikoli pro složitější systémy pro správu místností. Plná verze Blynku je placená, ale platforma nabízí také bezplatnou verzi, která umožňuje uživatelům vytvořit a ovládat jeden projekt s omezeným počtem widgetů. Celkově lze však použití Blynk jako asistenta domácí automatizace považovat za užitečné a účinné řešení pro vytvoření inteligentní domácnosti pro tento projekt. [28]

## 4 Vlastní práce

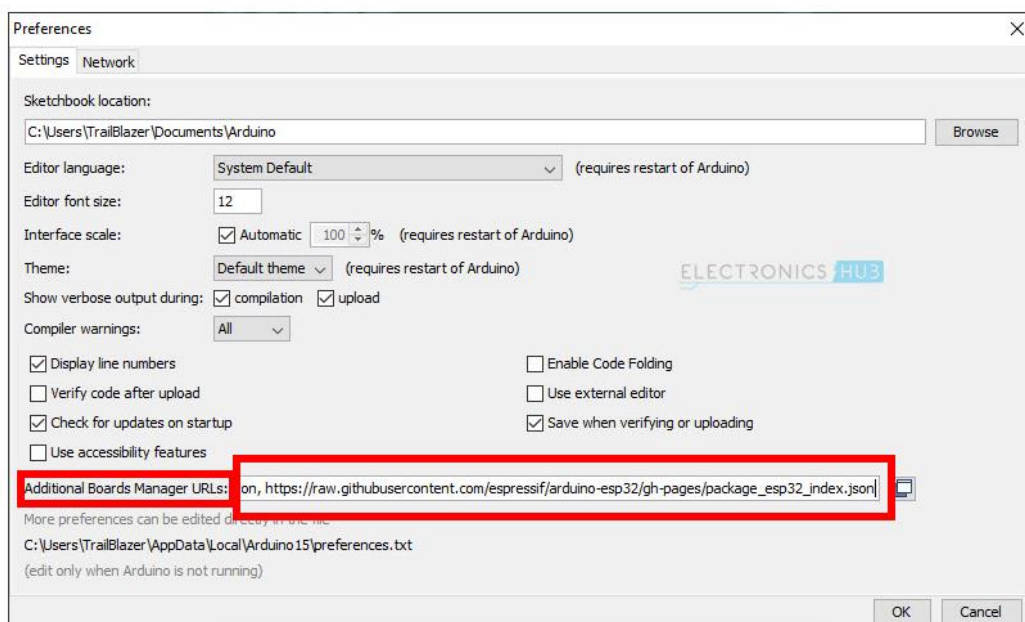
Tato kapitola se věnuje procesu realizace projektu detektoru kouře a ohně s použitím platformy ESP32 a vývojového prostředí Arduino IDE. V ní jsou prezentována schémata a diagramy popisující vývojový proces, spolu s podrobným vysvětlením kódu pro detektor kouře a ohně, jeho interakce s Blynkem a webovým rozhraním. Kromě toho je zde popsán proces výroby pouzdra pro zařízení.

Všechny tyto kroky jsou navzájem propojeny a sloučeny, aby byl vytvořen plně funkční detektor kouře a ohně, který může v reálném čase odesílat upozornění v případě vzniku požáru nebo kouře v prostoru.

### 4.1 Programování ESP32 v Arduino IDE

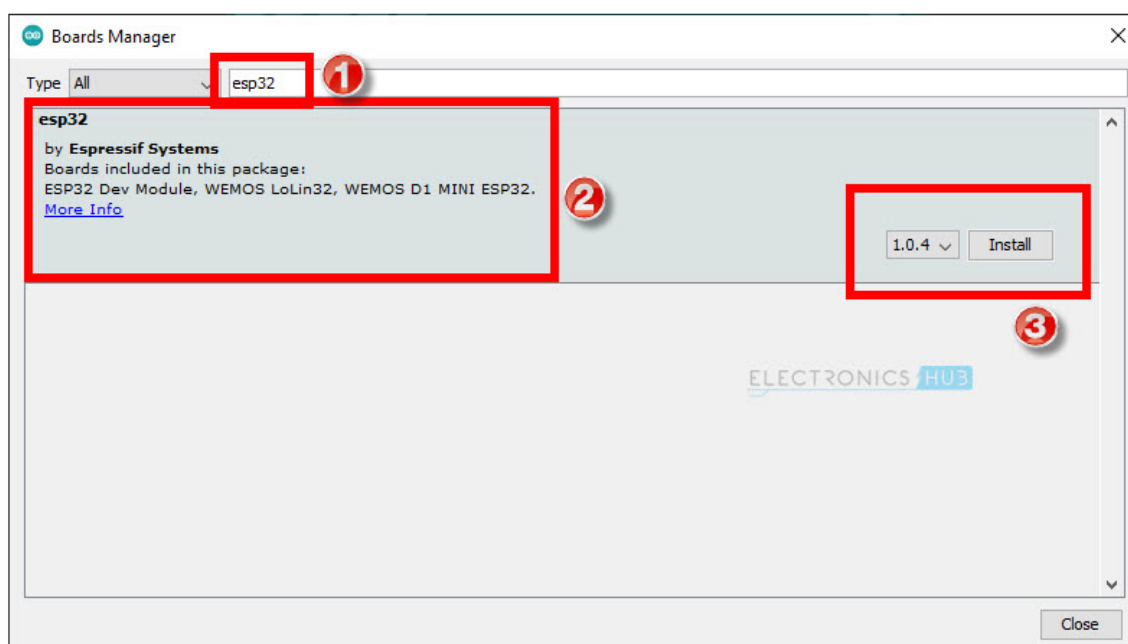
V prvním kroku při programování ESP32 v prostředí Arduino IDE je nutné stáhnout a nainstalovat tento software na svůj počítač. Tento software lze stáhnout z oficiální stránky Arduino IDE: <https://www.arduino.cc/en/software>. Poté je možné vytvářet sketchy pro ESP32 v prostředí Arduino IDE. Před nahráním sketche je však důležité vybrat typ používané desky ze seznamu v nabídce "Nástroje" > "Deska". Výchozí seznam desek v prostředí Arduino IDE však nezahrnuje desky ESP32. Proto je nutné provést konfiguraci tak, aby se tyto desky objevily v seznamu nabídek. V prostředí "Arduino IDE" existuje "Správce desek", který pomůže rychle přidat zajímavé desky do seznamu. Pro dosažení tohoto cíle byly podniknuty následující kroky.

Pro přidání desek ESP32 do seznamu nabídek v prostředí Arduino IDE je nutné otevřít prostředí a přejít do nabídky "Soubor" a následně vybrat "Nastavení". V otevřeném okně je pole s názvem "Další odkazy pro Správce desek". Do tohoto pole je nutné zadat následující adresu: [https://raw.githubusercontent.com/espressif/arduinoesp32/ghpages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduinoesp32/ghpages/package_esp32_index.json) a kliknout na tlačítko "OK". Tímto způsobem jsou desky ESP32 úspěšně přidány do seznamu nabídek v prostředí Arduino IDE.



Obrázek 20: Přidání desek ESP32 do seznamu nabídek v prostředí Arduino IDE.

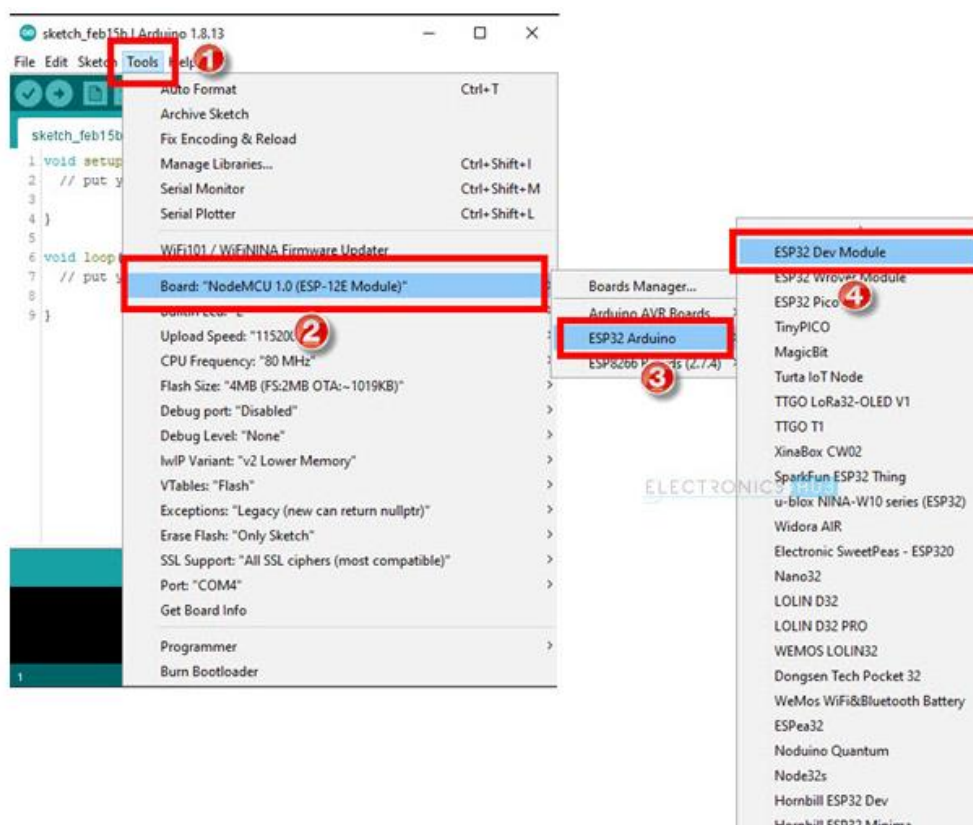
Pro pokračování v nastavení prostředí pro programování mikrokontroléru ESP32 je třeba spustit Správce desek. Uživatel může toto okno otevřít kliknutím na možnost "Nástroje" > "Deska" > "Manažér desek". Následně se zobrazí seznam dostupných desek, kde je nutné vybrat "esp32 od společnosti Espressif Systems". Po výběru desky je nutné kliknout na tlačítko "Instalovat". Uživatel může zvolit požadovanou verzi sestavy v závislosti na svých potřebách.



Obrázek 21: Manažér desek v prostředí Arduino IDE.



Pro výběr vhodné desky pro psaní sketchů je nutné opět přejít do nabídky "Nástroje" a zvolit "Deska". V této nabídce bude vidět nová volba s názvem "ESP32 Arduino", která byla přidána díky konfiguraci, kterou uživatel provedl v předchozím kroku. Po najetí kurzorem myši na tuto volbu se zobrazí seznam desek ESP32, které jsou podporovány v prostředí Arduino IDE. Vzhledem k tomu, že se v diplomové práci bude používat univerzální deska ESP32 DevKit, je nutné z tohoto seznamu vybrat "ESP32 Dev Module". Tato deska se používá pro většinu univerzálních 30-pinových desek ESP32, které jsou dnes k dispozici.

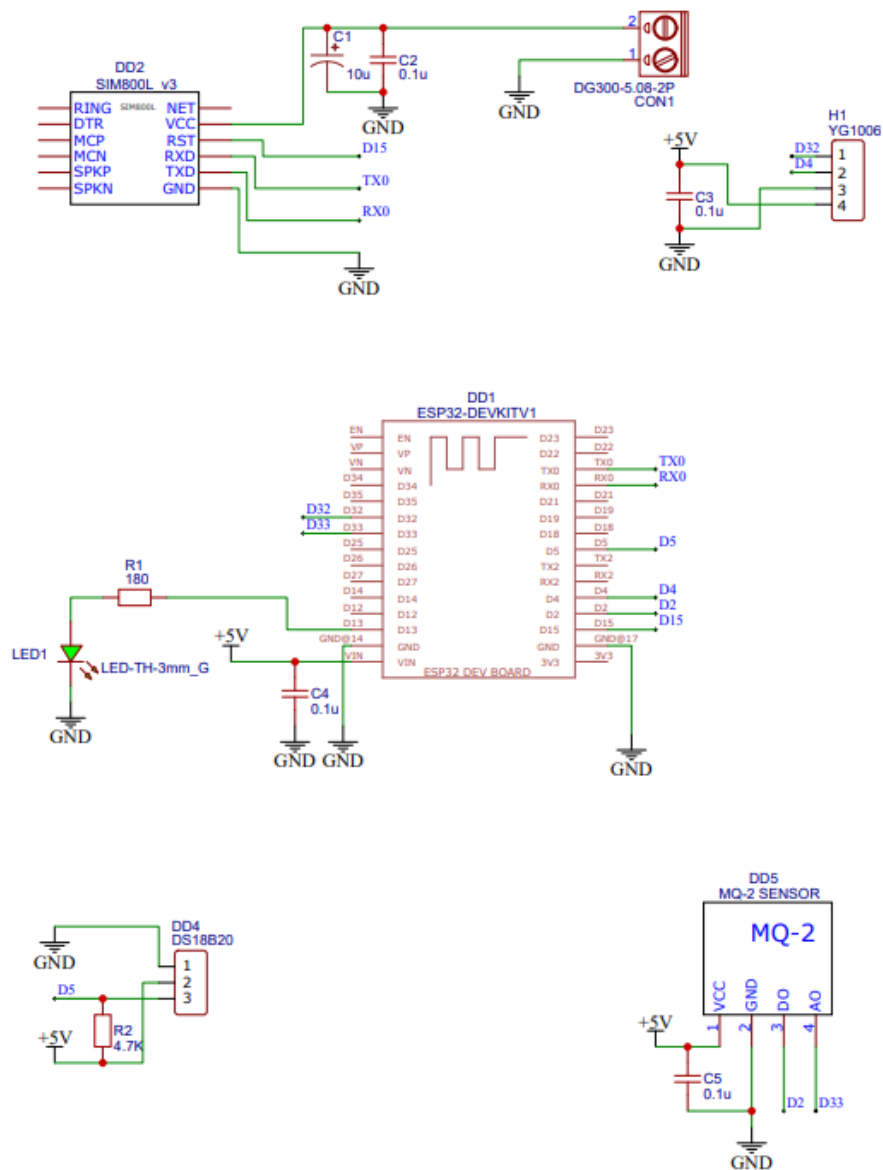


Obrázek 22: Výběr vhodné desky- ESP32 Dev Module.

## 4.2 Schéma zapojení a návrh plošného spoje

Tato kapitola se zabývá procesem návrhu detektoru v programu EasyEDA, což zahrnuje vytváření schémat zapojení, konstrukci a tvorbu 3D modelů plošných spojů. Na začátku procesu bylo vytvořeno schéma zapojení, kde bylo možné snadno přidávat a upravovat komponenty, propojení a další prvky schématu.

Na schématu je vidět, že LED dioda je připojena k pinu D13 a zemi. Tímto způsobem bude možné identifikovat, že zařízení je připojeno k Wi-Fi síti.

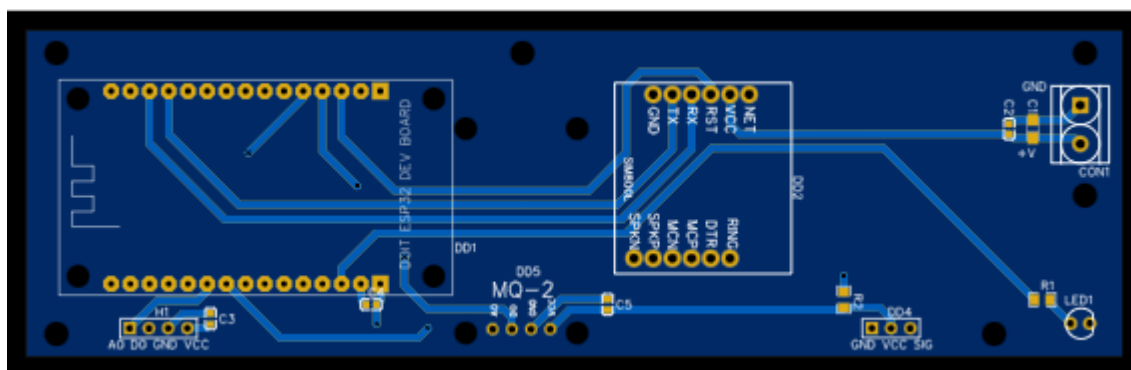


Obrázek 23: Schéma zapojení detektorů kouře a požáru.

Pro připojení senzorů plynu MQ-2 a plamene YG1006 k ESP32 je třeba použít 5V napájení a připojit jejich digitální výstupy na vstupy D2 a D4, zatímco analogové výstupy jsou připojeny na vstup D33 a D32 na desce. Při připojování senzoru teploty DS18B20 je nezbytné připojit červený vodič k napájecímu pinu desky, černý vodič k uzemnění (GND) desky a žlutý vodič k digitálnímu vstupu D5 na desce ESP32. Kromě toho, je také nutné připojit pull-up rezistor o hodnotě 4,7k $\Omega$  mezi napájecí pin a digitální vstup D5 na desce, což zajišťuje stabilitu signálu z senzoru a přesnost výstupu.

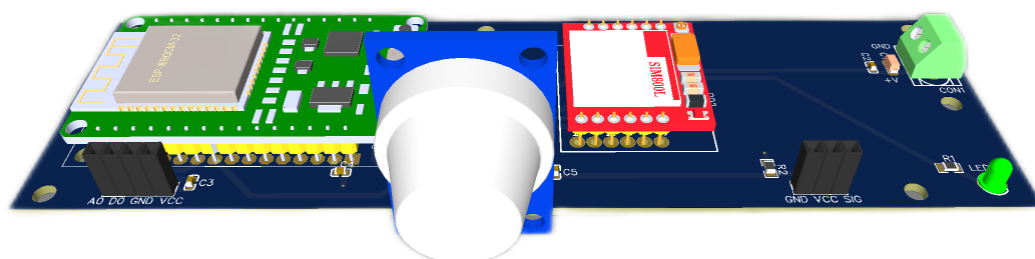
Při připojování modulu SIM800L k desce ESP32, je nutné nejprve připojit TX pin modulu SIM800L k RX pinu desky ESP32 a RX pin modulu SIM800L k TX pinu desky ESP32. Pro napájení modulu je doporučeno napětí v rozmezí od 3,3 V do 4,1 V na pinu VCC. Jelikož je důležité, aby modul byl spolehlivě napájen a nebyl poškozen příliš vysokým napětím, bude k modulu připojena externí baterie s napětím 4,1 V. Dodržení této hodnoty je důležité pro spolehlivé napájení modulu a aby nedošlo k jeho poškození vlivem příliš vysokého napětí.

Po vytvoření a ověření konečného schématu zapojení byl navržen plošný spoj. Při návrhu plošného spoje byly na desce umístěny komponenty dle schématu zapojení a byly navrženy a optimalizovány spoje mezi jednotlivými komponentami. Po kontrole a validaci celkového návrhu plošného spoje byl vytvořen 3D model plošného spoje.



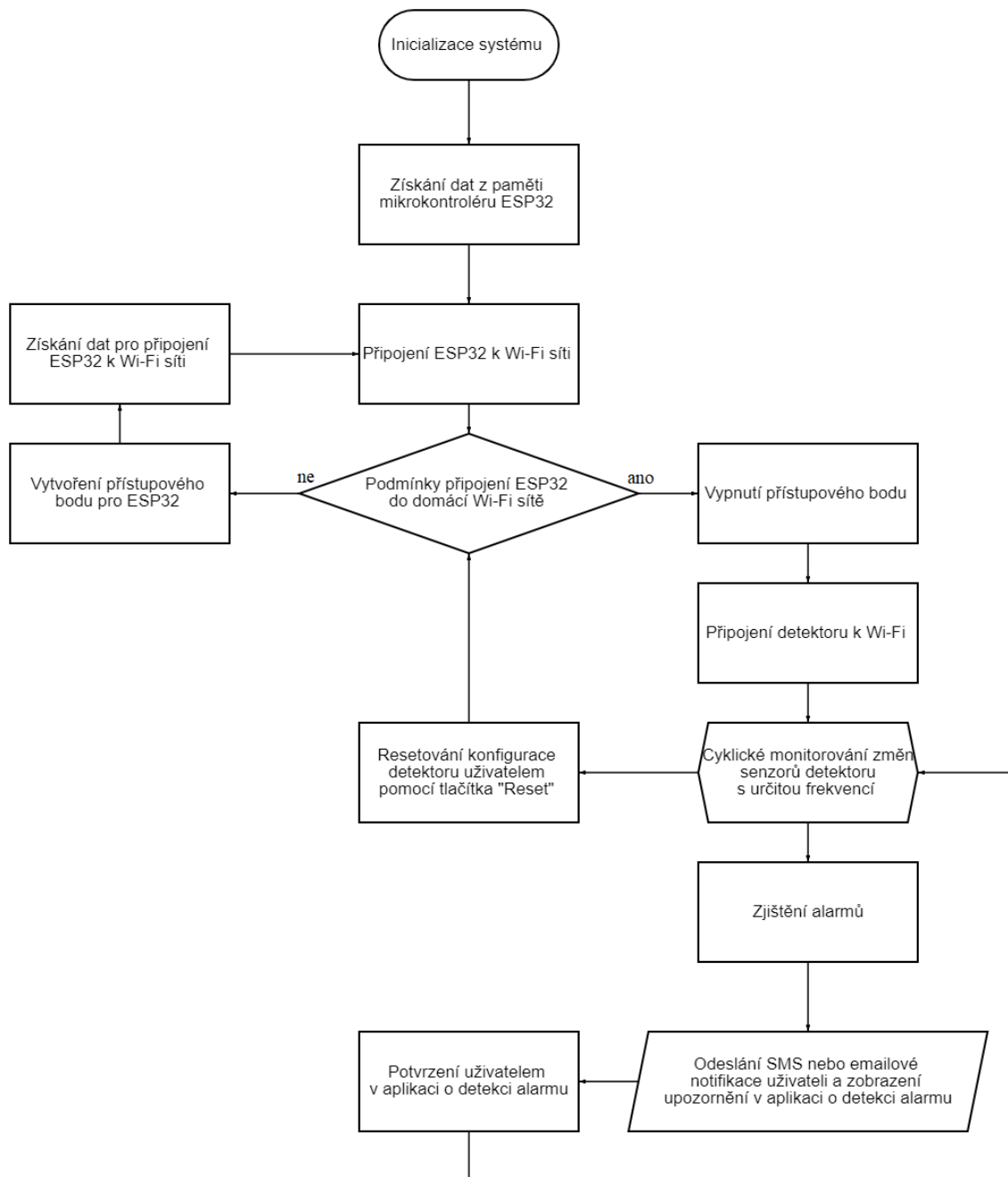
Obrázek 23: Plošný spoj.

Pro vývoj 3D modelu byla zahájena funkce 3D náhledu plošného spoje v EasyEDA a naimportovány 3D modely jednotlivých komponent. Poté byl vytvořen celkový 3D model plošného spoje, zahrnující všechny komponenty a spoje. Nakonec byl proveden export 3D modelu ve vhodném formátu pro další zpracování a prezentaci v rámci diplomové práce.



*Obrázek 24: 3D náhledu plošného spoje.*

### 4.3 Vývojový diagram



Obrázek 25: Vývojový diagram detektoru.

Tento vývojový diagram popisuje proces spojení mikrokontroléru ESP32 s Wi-Fi sítí, připojení senzoru a monitorování změn na senzoru s následným odesláním upozornění uživateli o spuštění poplachového signálu.

Proces se zahajuje inicializací systému, následně se získávají data z paměti mikrokontroléru ESP32 a informace pro připojení k Wi-Fi síti. Poté se kontrolují podmínky

pro připojení mikrokontroleru ESP32 k domácí Wi-Fi síti, a pokud jsou splněny, dochází k připojení mikrokontroleru ESP32 k Wi-Fi síti a odpojení přístupového bodu. Pokud podmínky nejsou splněny, vytvoří se přístupový bod pro ESP32.

Následuje připojení detektoru k Wi-Fi a pravidelné sledování změn na senzoru s danou frekvencí. Po zjištění změn na senzoru se uživateli zašle upozornění prostřednictvím SMS nebo e-mailu a zobrazí se v aplikaci.

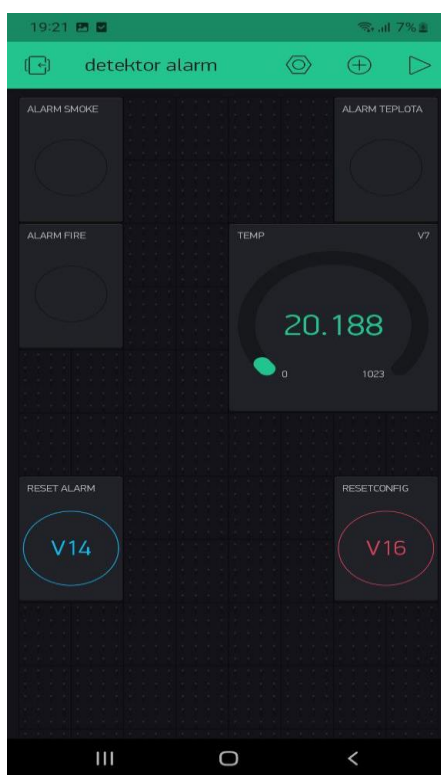
Vývojový diagram dále obsahuje části pro aktivaci poplachového signálu a možnost resetování konfigurace senzoru uživatelem prostřednictvím tlačítka "Reset" v aplikaci Blynk.

## 4.4 Blynk

Blynk je platforma, která umožňuje snadné vytváření aplikací a ovládání zařízení přes internet. V této kapitole se zaměříme na proces sestavování projektu pomocí aplikace Blynk.

Uživatel musí nejprve vytvořit nový projekt v Blynk a k tomu je nezbytná registrace a přihlášení do účtu. Po úspěšném přihlášení uživatel klikne na tlačítko "Vytvořit nový projekt" a zvolí zařízení ESP32.

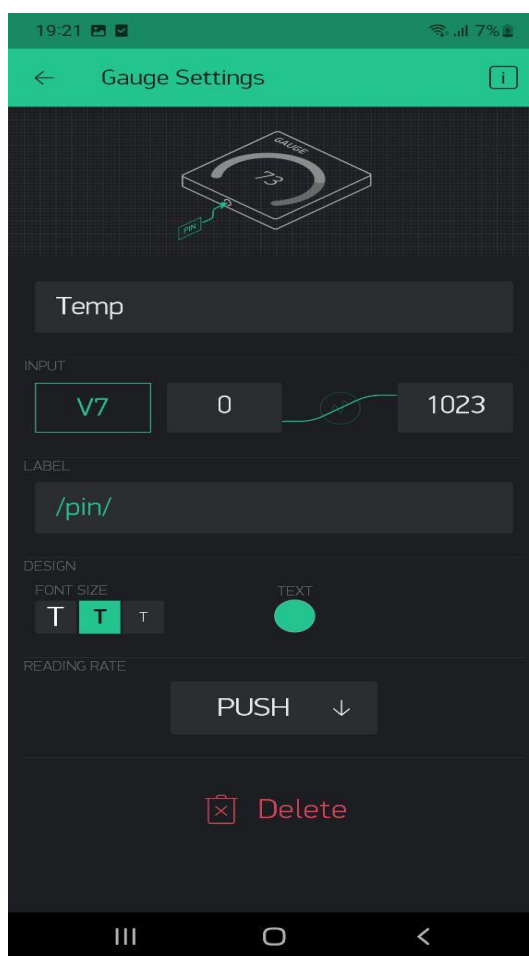
Dalším krokem je přidání různých widgetů, které jsou k dispozici v aplikaci Blynk. Widgety zahrnují tlačítka, posuvníky, grafy a mnoho dalšího. Pro přidání widgetu do projektu jednoduše uživatel přetáhne požadovaný widget na plátno.



Obrázek 26: Plátno projektu detektorů kouře a požáru.

Na plátno projektu jsou umístěny tři signalizační lampy, které slouží k signalizaci při detekci kouře, ohně nebo prudkého zvýšení teploty. Kromě toho je na plátně umístěn číselník, který zobrazuje aktuální teplotu vzduchu. Součástí projektu je také tlačítko pro vymazání uživatelských dat a tlačítko, kterým může uživatel informovat systém o tom, že viděl poplachy ze zařízení. Je důležité zajistit, aby tyto funkce byly bezpečné a spolehlivé a aby uživatelé byli dostatečně informováni o tom, jak je používat.

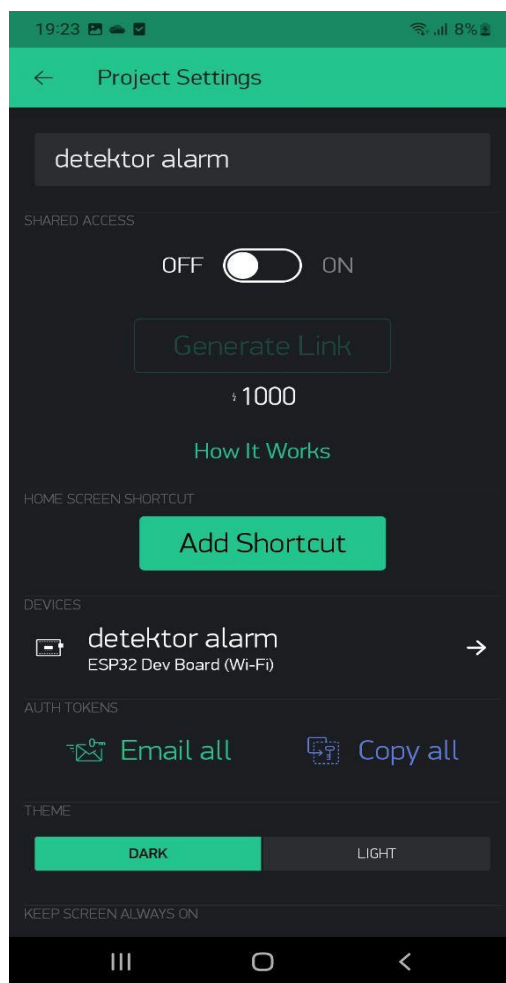
Po umístění widgetů na plátno projektu je důležité je dále upravit podle potřeb projektu. Konkrétně lze upravit vlastnosti jako velikost, barvu, text a také definovat akce, které budou vyvolány kliknutím na widget. Kromě toho je nezbytné zajistit propojení widgetů s senzory. K tomu je třeba použít nabídku vlevo, kde se nachází položka "Device Selector". Poté je potřeba vybrat zařízení, které bude sloužit k řízení widgetů. Je důležité zajistit, aby propojení mezi widgety a senzory bylo správně nastavené, aby projekt fungoval bez problémů.



Obrázek 27: Položka výběr zařízení v aplikaci Blink.



Pro dokončení procesu je nutné získat autentizační kód, který bude umožňovat použití aplikace i mimo interní síť. K tomu je třeba použít nastavení a nechat si zaslat unikátní kód na zadanou e-mailovou adresu. Díky tomuto kódu bude možné využívat aplikaci na různých zařízeních a z jakéhokoli místa s připojením k internetu. Je nutné dbát na to, aby autentizační kód byl bezpečně uložen a nepřístupný pro neoprávněné osoby.



Obrázek 28: Získání autentizační kódu v aplikaci Blynk.

#### 4.4.1 Implementace Blynk knihoven v sketchu

```
#define USE_CUSTOM_BOARD
#define APP_DEBUG
#define BLYNK_PRINT Serial
WidgetLED ledSmokeAlarm(V2);
WidgetLED ledFlameAlarm(V4);
WidgetLED ledTempAlarm(V6);
```

Tento kód je částí programu (sketch) pro detektor kouře a ohně na platformě ESP32, který využívá knihovnu Blynk pro projekt. Jednotlivé řádky kódu jsou vysvětleny následovně:

`USE_CUSTOM_BOARD` definuje použití vlastní desky, což znamená, že projekt používá hardwarové zařízení, které není standardní deska nebo vývojová deska podporovaná Blynk. `APP_DEBUG` nastavuje režim ladění aplikace. Pokud je toto makro definováno, umožňuje to vývojáři výstup informací o aplikaci, které pomáhají při ladění. Naopak, v opačném případě se tyto informace nezobrazí. `BLYNK_PRINT Serial` definuje, jaký port bude použit pro výstup z knihovny Blynk. V tomto případě se používá sériový port.

`WidgetLED ledSmokeAlarm(V2)`, `ledFlameAlarm(V4)`, `WidgetLED ledTempAlarm(V6)` definují nové objekty typu `WidgetLED`, které jsou přiřazeny ke virtuálním pinům V2, V4 a V6 v Blynk aplikaci. Tyto objekty umožňují ovládání LED diod v Blynk aplikaci. Tyto LED diody slouží jako indikátory pro alarmy různých senzorů projektu.

Celkově tento kód definuje použití vlastní desky, zapnutí režimu ladění aplikace, nastavení sériového portu pro knihovnu Blynk a vytvoření tří LED diod přiřazených k různým virtuálním pinům v Blynk aplikaci. Tyto LED diody slouží jako indikátory pro alarmy různých senzorů projektu, což umožňuje uživatelům sledovat stav senzorů pomocí Blynk aplikace. Díky využití knihovny Blynk je také možné ovládat a monitorovat detektor kouře a ohně z dálky, což zvyšuje jeho užitečnost a efektivitu.

## 4.5 Web rozhraní

K získání uživatelských dat, jako jsou přihlašovací údaje k WiFi, telefonní číslo, e-mailová adresa a ověřovací kód, se využívá webové rozhraní, které je vytvořeno s využitím tří knihoven.

```
#include <WiFiClient.h>
#include <WebServer.h>
#include <DNSServer.h>
```

Tyto knihovny jsou určeny pro programování mikrokontroléru ESP32 a umožňují jeho propojení s WiFi sítí a webovým serverem.

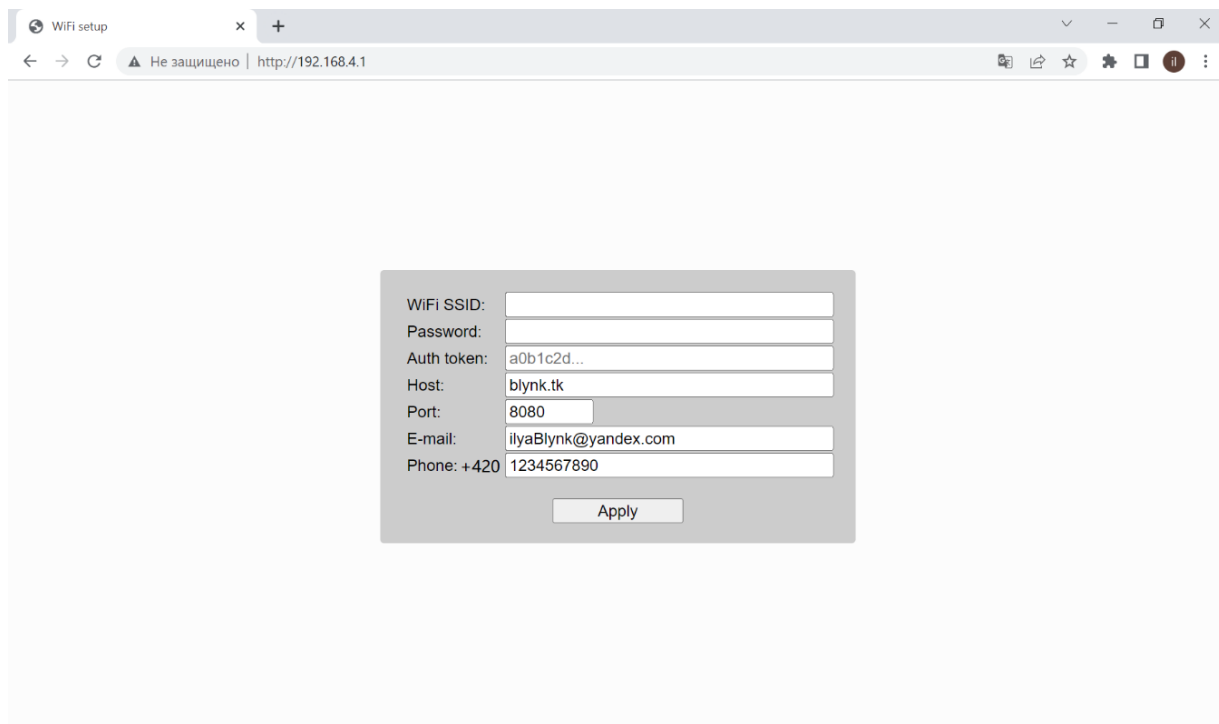
WiFiClient.h: Tato knihovna poskytuje třídu WiFiClient, která slouží k vytvoření TCP spojení s jiným zařízením v síti. Tuto třídu lze použít pro odesílání a přijímání dat z internetu nebo z jiných zařízení v místní síti.

WebServer.h: Tato knihovna umožňuje vytvoření webového serveru na ESP32, který lze použít k poskytování různých webových služeb. S touto knihovnou lze vytvořit webové rozhraní pro ovládání zařízení, zobrazení dat a mnoho dalšího.

DNSServer.h: Tato knihovna umožňuje ESP32 fungovat jako DNS server, což umožňuje vytvářet vlastní lokální jmenné servery. Toto může být užitečné pro testování nebo pro vývoj aplikací, které vyžadují vlastní jmenný server.

Všechny tyto knihovny jsou součástí vývojového prostředí pro ESP32 a jsou k dispozici zdarma. Tyto knihovny jsou často používány v IoT aplikacích, které využívají bezdrátovou komunikaci a webové rozhraní pro ovládání zařízení.

Po zapnutí detektoru se vytvoří Wi-Fi přístupový bod, ke kterému se uživatel může připojit pomocí svého telefonu nebo počítače. Poté bude moci odesílat data detektoru pomocí připojení k serveru na adrese 192.168.4.1. Po úspěšném připojení uživatel obdrží webovou stránku s údaji detektoru a možnostmi jeho ovládání. Webová stránka byla navržena s cílem poskytnout uživatelům co největší pohodlí a jednoduchost v ovládání detektoru. Je velmi intuitivní a srozumitelná pro všechny uživatele bez ohledu na jejich technické znalosti. Navigace je snadná a uživatelské rozhraní je přehledné, což umožňuje uživatelům rychle a efektivně ovládat detektor a získávat důležité informace o prostředí, ve kterém se nachází.



Obrázek 29: Web rozhraní pro získání uživatelských dat.

## 4.6 Implementace kódu detektoru kouře a ohně

V této části bude podrobněji popsán kód detektoru, včetně klíčových částí. Je důležité poznamenat, že část kódu je k dispozici v příloze této diplomové práce.

### 4.6.1 Monitorování kouře a plamenů

```
int CO2pin = 2;
int flamePin = 4;
int smokeAlarm = 0;
int flameAlarm = 0;
int flagOnGuard = 1;

void loop() {
  BlynkProvisioning.run();
  if (digitalRead(CO2pin)) {
    if (!smokeAlarm && flagOnGuard) {
      ledSmokeAlarm.on();
      Blynk.email(configStore.email, " Alarm ", " Smoke detected ");
      send_SMS(1);
    }
  }
}
```

```

    }
}
if (digitalRead(flamePin)) {
    if (!flameAlarm && flagOnGuard) {
        ledFlameAlarm.on();
        Blynk.email(configStore.email, " Alarm ", "Flame detected ");
        send_SMS(2);
    }
}
}

```

Tento kód definuje dvě vstupní piny, CO2pin a flamePin, které slouží k monitorování přítomnosti kouře a plamenů.

Funkce loop () je hlavní smyčkou programu a provádí se neustále. První věc, kterou dělá, je volání funkce BlynkProvisioning.run (), která zřejmě zajišťuje připojení k serveru Blynk.

Poté následuje podmínka pro monitorování CO2pin a flamePin. Pokud CO2pin detekuje kouř, a flagOnGuard je nastaven na true, bude spuštěn alarm pro kouř a bude odeslán e-mail a SMS zpráva. Podobně, pokud flamePin detekuje plameny, spustí se alarm pro plameny a odešle se e-mail a SMS zpráva.

#### 4.6.2 Monitorování teploty

```

#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 5;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(oneWire);
int tempAlarm = 0;
float prevTemp;
float newTemp;

void loop() {
    sensors.requestTemperatures();
    float temp = sensors.getTempCByIndex(0);
    if ((newTemp - prevTemp) > 5 ) {
        Serial.println("(newTemp - prevTemp) > 2");
        if (!tempAlarm && flagOnGuard) {
            ledTempAlarm.on();

```

```

        Blynk.email(configStore.email, "Alarm! Ripid temperature rise
detected!");
        send_SMS(3);
    }
    prevTemp = newTemp;
    prevMillis = millis();
}
}

```

Tento kód používá knihovny OneWire a DallasTemperature pro monitorování teploty pomocí čidla teploty. Knihovna OneWire je používána pro komunikaci s různými zařízeními připojenými k jedinému vodiči dat. Jednoduchý jednočinný protokol zahrnuje zaslání jedné příkazové zprávy na sběrnici, která obsahuje identifikátor zařízení a požadovanou akci. Knihovna DallasTemperature je specifickou implementací protokolu OneWire pro čtení teplot z čidla teploty DS18B20 od společnosti Dallas Semiconductor. Tato knihovna poskytuje snadný a přehledný způsob, jak komunikovat s těmito teplotními čidly a získat teplotní informace. Knihovna umožňuje načíst teplotu z více čidel připojených na stejnou sběrnici OneWire. Čidlo teploty je připojeno k pinu 5 (D5) a používá se k měření teploty v jednotkách Celsius.

Funkce požaduje aktualizaci hodnoty teploty z čidla a porovnává rozdíl mezi předchozí a novou hodnotou teploty. Pokud rozdíl překročí 5 stupňů, spustí se alarm pro teplotu a odešlou se e-mail a SMS zpráva, pokud flagOnGuard je nastaven na true. Předchozí teplota a čas jsou aktualizovány k poslední známé hodnotě.

### 4.6.3 Odesílání alarmů

```

void send_SMS(int typeOfAlarm)
{
    Serial2.println("AT+CMGF=1");
    String prefix = "AT+CMGS=\"+420";
    String suffix = "\"";
    String number = prefix + configStore.phone + suffix;
    Serial2.println(number);
    delay(150);

    switch (typeOfAlarm) {
        case 0: //
            Serial2.print("Fire Security System restarted");
    }
}

```

```

    break;
case 1: //
    Serial2.print("Alarm! Smoke detected!");
    break;
case 2: //
    Serial2.print("Alarm! Flame detected!");
    break;
case 3: //
    Serial2.print("Alarm! Rapid temperature rise detected!");
    break;
}
delay(150);
}

```

Tento kód slouží k odeslání SMS zprávy. Nejprve se nastaví režim textové zprávy pomocí příkazu "AT+CMGF=1". Poté se vytvoří text zprávy s prefixem "+420" (pro české čísla) a telefonním číslem, které je uloženo v konfiguračním souboru. V závislosti na typu alarmu se vybere odpovídající textová zpráva a odešle se pomocí příkazu "AT+CMGS". Poté se zpožděním 150 milisekund ukončí odeslání zprávy.

## 4.7 Pouzdro

Pouzdro pro detektor kouře a požáru je nezbytným prvkem pro ochranu zařízení před poškozením a vlivy okolního prostředí. Materiál pouzdra může být volen z různých možností, včetně kovů, plastů a karton. Při volbě vhodného materiálu je třeba zvážit podmínky provozu detektoru, jeho cenu, možnosti modifikace a vzhled. Je důležité najít rovnováhu mezi těmito faktory a zvolit materiál, který nejlépe vyhovuje konkrétnímu zařízení.

Kovové pouzdro poskytuje největší pevnost a spolehlivost, chrání před mechanickými poškozeními a odolává vysokým teplotám a požárům. Na druhou stranu je kovové pouzdro dražší. Nehořlavé kartonové pouzdro je levnější a snadněji modifikovatelné, ale má nižší pevnost a odolnost vysokým teplotám, což může být v některých podmínkách provozu důležitým faktorem.

Pro tento projekt bylo vybráno plastové pouzdro, vytvořené pomocí 3D tisku, které nabízí nejoptimálnější řešení. Plastové pouzdro má mnoho výhod, jako je nízká cena a snadná modifikace, díky tomu lze pouzdra vytvářet v různých barvách a designech.



*Obrázek 30: Pouzdro pro detektor.*

Je třeba mít na paměti, že i když plastové pouzdro představuje optimální řešení pro daný projekt, nejedná se o nejsilnější a nejodolnější možnost. Pouzdro z plastu může být poškozeno mechanickým účinkem nebo v případě požáru, což znamená, že není vhodné pro použití v extrémních podmínkách.

Kromě toho, je třeba si uvědomit, že pouzdro je pouze jednou částí detektoru kouře a ohně a jeho kvalita není jediným faktorem, který ovlivňuje účinnost a spolehlivost zařízení. Dalšími důležitými faktory jsou senzory, algoritmy zpracování dat a možnosti připojení k požární signalizaci.



## 4.8 Indikace stavů zařízení pomocí LED diody

Stav zařízení ukazuje LED dioda umístěná v pravém dolním rohu přední strany detektoru.



Obrázek 31: Přední strana detektoru.

Tento kód představuje zpracování stavů zařízení a každý stav má své vlastní indikace na LED diodě. Celkem existuje 6 stavů, jako jsou `MODE_RESET_CONFIG`, `MODE_WAIT_CONFIG`, `MODE_CONFIGURING`, `MODE_CONNECTING_NET` a `MODE_RUNNING`, a také stav výchozího nastavení. Každý stav má svou specifickou indikaci na LED diodě.

```
switch (currState) {  
    case MODE_RESET_CONFIG:    return beatLED((int[]){ 50, 50 });  
    case MODE_WAIT_CONFIG:    return beatLED((int[]){ 50, 500 });  
    case MODE_CONFIGURING:    return beatLED((int[]){ 200, 200 });  
    case MODE_CONNECTING_NET: return beatLED((int[]){ 50, 500 });  
    case MODE_RUNNING:        return waveLED(5000);  
    default:                   return beatLED((int[]){ 80, 100, 80,  
1000 } );  
}
```

Funkce `beatLED()` se používá k nastavení LED diody do režimu blikání s danými parametry. První argument funkce je pole celých čísel obsahující dvě hodnoty: dobu zapnutí LED diody a dobu vypnutí LED diody v milisekundách. Funkce `waveLED()` se používá k vytvoření vlnkového efektu na LED diodě.

Každý stav a odpovídající indikace jsou popsány následovně:

**MODE\_RESET\_CONFIG:** Tento stav označuje začátek procesu resetování konfigurace zařízení na tovární nastavení, aby mohlo být nastaveno znovu. Obvykle se to děje při prvním spuštění nebo pokud uživatel rozhodne změnit nastavení. LED dioda bliká rychlostí 20krát za sekundu po dobu 50 milisekund zapnutí a 50 milisekund vypnutí.

**MODE\_WAIT\_CONFIG:** Tento stav označuje, že zařízení je v režimu čekání na nastavení a nemůže pracovat, dokud nejsou zadána nová nastavení. LED dioda bliká rychlostí 2krát za sekundu po dobu 50 milisekund zapnutí a 500 milisekund vypnutí.

**MODE\_CONFIGURING:** Tento stav označuje proces nastavování zařízení, včetně připojení k síti, nastavení síťových parametrů atd. LED dioda bliká rychlostí 5krát za sekundu po dobu 200 milisekund zapnutí a 200 milisekund vypnutí.

**MODE\_CONNECTING\_NET:** Tento stav označuje proces připojení zařízení k místní síti. LED dioda bliká rychlostí 2 krát za sekundu, s dobou svícení 50 milisekund a dobou zhasnutí 500 milisekund.

**MODE\_RUNNING:** Tento stav ukazuje, že zařízení je v provozním režimu, kdy jsou již všechny nastavení provedena a zařízení je připraveno k použití. LED dioda vytváří vlnkový efekt po dobu 5 sekund.

**Výchozí stav:** Pokud je zařízení v neočekávaném stavu nebo došlo k nějaké chybě v programu, LED dioda bude blikat rychlostí 3 krát za sekundu s pauzou mezi zapnutími 80 milisekund a poté zhasne na 1 sekundu.

## 5 Hodnocení výsledků

V této části bude posuzována efektivita detektoru kouře a plamenů. Klíčové aspekty a parametry výkonu detektoru byly analyzovány a vyhodnoceny z hlediska jeho schopnosti úspěšně plnit svou funkci.

Přesnost detekce je zásadním kritériem při hodnocení účinnosti detektoru. Během testování bylo zjištěno, že detektor správně rozpoznal většinu skutečných požárních událostí, což svědčí o jeho schopnosti efektivně detekovat kouř a plameny s nízkým počtem falešných poplachů. Citlivost senzorů detektoru byly nastaveny na úroveň 75 %.

Pro ověření přesnosti byl detektor testován ve třech různých vzdálenostech od zdroje kouře a plamenů. Pro každou vzdálenost bylo provedeno 50 testů a zaznamenána úspěšnost detekce kouře.

*Tabulka 2: Přesnost detekce kouře.*

Vzdálenost (metry)	Počet testů	Průměrná doba reakce (sekundy)	Úspěšnost detekce (%)
0.3	50	2	98
0.5	50	4	96
0.7	50	9	92
Průměrná přesnost			95

*Tabulka 3: Přesnost detekce plamene.*

Vzdálenost (metry)	Počet testů	Průměrná doba reakce (sekundy)	Úspěšnost detekce (%)
0.5	50	1	98
1	50	1.3	96
1.5	50	1.4	92
Průměrná přesnost			95

Detekce plamenů probíhá rychleji než detekce kouře z několika důvodů. Jedním z hlavních faktorů, který ovlivňuje rychlost detekce plamenů, je jejich záření. Plameny vydávají infračervené a ultrafialové záření, které mohou být snadno zachyceny senzory detektorů. Toto záření se šíří rychleji než kouř, což umožňuje rychlejší detekci plamenů.

Na druhou stranu, detekce kouře je ovlivněna mnoha faktory, které ztěžují jeho rychlé rozpoznání. Faktory ovlivňující rozšíření kouře zahrnují teplotu, vlhkost, proudění vzduchu a další. Tyto faktory mohou způsobit, že kouř se šíří pomaleji nebo se rozptýluje v různých směrech, což ztěžuje jeho detekci. Byl také testován teploměr pro zjištění přesnosti měření teploty.

*Tabulka 4: Přesnost detekce změny teploty.*

<b>Skutečná teplota (°C)</b>	<b>Naměřená teplota (°C)</b>	<b>Přesnost detekce (%)</b>
10	10.12	99
15	14.97	99
20	20.04	99
25	25.11	99

Rychlost doručení alarmů byla testována měřením času mezi detekcí hrozby detektorem a doručením upozornění prostřednictvím SMS a e-mailu. Bylo provedeno 10 testů a zaznamenán průměrný čas doručení upozornění. SMS zprávy jsou obvykle doručovány rychleji než e-maily, protože jsou odesílány přes mobilní sítě, zatímco e-maily musí projít různými servery a směrovacími zařízeními. SMS zprávy také obvykle vyžadují méně dat než e-maily, což zkracuje čas potřebný k doručení zprávy.

*Tabulka 5: Rychlost doručení alarmů.*

<b>Typ upozornění</b>	<b>Počet testů</b>	<b>Průměrná doba doručení (sekundy)</b>
SMS	10	3
E-mail	10	5
Průměrná rychlost reakce		4

Spolehlivost a stabilita detektoru představují další klíčové aspekty, které je nutné vzít v úvahu při jeho posuzování. Během testovací fáze nebyly identifikovány žádné závažné problémy spojené s hardwarem či softwarem detektoru. Pro ověření stability detektoru bylo sledováno maximální odchýlení detekce za různých teplotních a vlhkostních podmínek. Bylo realizováno tři testy za různých okolností a zjištěna průměrná maximální odchylka. Během testů byla průměrná maximální odchylka zaznamenána jako rozdíl mezi naměřenými a

skutečnými hodnotami detekce. Tyto odchylky mohou pramenit z různých faktorů, jako je citlivost senzorů, zpracování signálu detektoru nebo vnějších podmínek, jako jsou teplota a vlhkost prostředí.

*Tabulka 6: Stabilita detektoru.*

<b>Test č.</b>	<b>Teplota (°C)</b>	<b>Vlhkost (%)</b>	<b>Max. odchylka (%)</b>
1	10	30	1.2
2	20	45	0.8
3	30	60	1

Je důležité zohlednit i cenovou dostupnost detektoru, která hraje klíčovou roli. Detektor kouř a plamenů využívá cenově dostupné součástky a materiály, což z něj činí cenově přijatelné zařízení.

*Tabulka 7: Náklady na součástky detektoru.*

<b>Komponent</b>	<b>Cena (Kč)</b>
MQ-2	45
YG1006	30
DS18B20	70
SIM 800L	80
ESP32 DEV KID	160
3D tisk	400
Výroba DPS	350
Celková cena	1135

Uživatelská přívětivost je dalším důležitým aspektem při hodnocení detektoru. Zařízení bylo navrženo tak, aby bylo snadno použitelné a nastavitelné. Indikace stavu zařízení pomocí LED diod zajišťuje jasnou a snadno rozpoznatelnou vizuální signalizaci, která umožňuje uživatelům snadno pochopit aktuální stav detektoru. Navíc, proces konfigurace a připojení k síti byl zjednodušen, což usnadňuje instalaci a údržbu zařízení.

Důležitým krokem při hodnocení energetické účinnosti detektoru je měření jeho skutečné spotřeby energie. K tomuto účelu byl použit multimetr, který umožňuje měření elektrického proudu a napětí a tím i výpočet spotřeby energie. Po provedení měření bylo zjištěno, že spotřeba energie detektoru činí 3W.

Celkem lze říci, že hodnocení ukázalo, že detektor kouře a plamenů je efektivní a spolehlivý s vysokou přesností detekce, rychlým reagováním, spolehlivostí, stabilitou, cenovou dostupností, přívětivostí pro uživatele a energetickou efektivitou. Vzhledem k těmto vlastnostem představuje tento detektor ideální řešení pro zajištění bezpečnosti a ochrany proti požárům v různých druzích prostředí.

## 6 Závěr

V této diplomové práci byl úspěšně navržen a implementován inteligentní detektor, který umožňuje monitorování prostředí a zasílání emailů nebo SMS zpráv o alarmech. Detektor je doplněn o detekci zplodin hoření a měření změny teploty indikující požár. Zařízení umožňuje uživatelům být informováni pomocí SMS nebo emailu.

V rámci práce byla provedena studie hardwarových a softwarových možností řešení úlohy. Byla vybrána nejvhodnější varianta s kritickým hodnocením návrhu a specifikovány funkce modelu podle cílů práce. Dále byla prozkoumána teorie chování detektoru ve fázích požáru a senzory používané k detekci kouře, plamene a teploty.

V práci byly popsány senzory MQ-2, YG1006 a DS18B20 a podrobně popsány jejich vnitřní struktury, principy činnosti a kalibrace. Dále byly popsány wifi moduly, vývojová deska a vývojové prostředí používané k implementaci detektoru kouře a požáru. Práce také obsahuje podrobný popis implementace kódu, včetně monitorování kouře a plamenů, měření teploty a odesílání alarmů.

Výsledkem práce je plně funkční inteligentní detektor kouře, které umožňuje uživatelům rychlé a včasné reagování na nebezpečné situace. Detektor může být v budoucnu dále vylepšen a rozšířen o další funkce, které zvýší jeho spolehlivost a flexibilitu.

Děkuji všem, kteří mi poskytli pomoc, rady a nápady při práci na této diplomové práci. Doufám, že práce přinese užitek a pomůže chránit domovy a pracovní prostory.

## 7 Seznam použitých zdrojů

1. Hasicido. “Požár – definice, rozdělení, pásma, fáze hoření”. [online] [cit. 10.01.2023]. Dostupné z: <https://www.hasicido.cz/faq/pozar-definice-rozdeleni-pasma-faze-horeni>.
2. MORRISS S.B. “Automated manufacturing systems : actuators, controls, sensors, and robocs” New York: Glencoe, 1995. ISBN 0028023315.
3. GARDNER J.W., VARADAN V.K., AWADELKARIM O.O. “Microsensors, MEMS, and smart devices” Chichester: Wiley, 2001. ISBN 0-471-86109.
4. Pololu. “Robotics and Electronics”. [online] [cit. 10.01.2023]. Dostupné z: <https://www.pololu.com/file/0J309/MQ2.pdf>.
5. Last Minute Engineers. “How MQ2 Gas/Smoke Sensor. Works Interface it with Arduino.” [online] [cit. 15.01.2023]. Dostupné z: <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial>.
6. Návodů Drátek. “Senzor hořlavých plynů MQ-2 ”. [online] [cit. 15.01.2023]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/senzor-horlavych-plynu-mq-2.html>.
7. rhydoLABZ INDIA. “Technical Data Sheet”. [online] [cit. 16.01.2023]. Dostupné z: [https://www.rhydolabz.com/documents/29/YG1006\\_Datasheet.pdf](https://www.rhydolabz.com/documents/29/YG1006_Datasheet.pdf).
8. Electronics Hub - Tech. “Arduino Flame Sensor Interface – Working, Circuit Diagram, Code”. [online] [cit. 25.01.2023]. Dostupné z: <https://www.electronicshub.org/arduino-flame-sensor-interface>.
9. Dallas Semiconductor. “DS18B20 Datasheet(PDF)”. [online] [cit. 25.01.2023]. Dostupné z: <https://www.alldatasheet.com/datasheet-pdf/pdf/58557/DALLAS/DS18B20.html>.
10. Electronics Hub. “ESP32 DS18B20 Tutorial | DS18B20 Temperature Sensor with ESP32 ”. [online] [cit. 25.01.2023]. Dostupné z: <https://www.electronicshub.org/esp32-ds18b20-tutorial>.
11. BELL C.A. “Beginning sensor networks with Arduino and Raspberry Pi”. [New York, New York]: Apress, 2013. ISBN 1430258241.
12. Milan Klement Kl. “Technologie bezdrátových sítí – základní principy a standardy”. 2017. ISBN 978-80-244-5156-5.
13. GAST M. “802.11ac: A Survival Guide”. 2013. ISBN 978-1449343149.
14. PERAHIA E. STACEY R. “Next Generation Wireless LANs: 802.11n and 802.11ac”. 2013. ISBN 978-0521763288.



15. O'HARA B. "The 802.11 Handbook: A Designer's Companion". 2017. ISBN 978-0750677297.
16. Espressif. "ESP8266 Technical Reference". [Citace: 11.02.2023] Dostupné z :  
[https://www.espressif.com/sites/default/files/documentation/esp8266-](https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf)
17. [technical\\_reference\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf).
18. Electronics Hub. "ESP8266 using Arduino IDE". [online] [cit. 11.02.2023]. Dostupné z: <https://www.electronicshub.org/programming-esp8266-using-arduino-ide>.
19. Electronics Hub. "ESP8266 Pinout | Pins and Their Functions of ESP8266 SoC". [online] [cit. 12.02.2023]. Dostupné z: <https://www.electronicshub.org/esp8266-pinout>.
20. Mbtechworks. "I2C, SPI, UART Data Communications on the Raspberry Pi". [online] [cit. 12.02.2023]. Dostupné z: <https://www.mbtechworks.com/hardware/raspberry-pi-UART-SPI-I2C.html>
21. Electronics Hub. "ESP32 Projects for Beginners". [online] [cit. 25.02.2023]. Dostupné z: <https://www.electronicshub.org/esp32-projects/>
22. Espressif. "Get Started - ESP32". [online]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>
23. 599.cz. "První pokusy s GSM modulem SIM800L". [online] [Citace: 25.02.2023] Dostupné z: <https://599.cz/view.php?cisloclanku=2020022301>
24. SimCom. "SIM800L Design Datasheet". [online] [Citace: 01.03.2023] Dostupné z: <https://datasheetspdf.com/pdf/989664/SIMCom/SIM800L/1>
25. Last Minute Engineers. "ESP32 Pinout Reference". [online] [cit. 01.03.2023]. Dostupné z: <https://lastminuteengineers.com/esp32-pinout-reference/>
26. Návodý Drátek. "Vývojová deska ESP32". [online] [cit. 04.02.2023]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/vyvojova-deska-esp32.html>.
27. Circuto team. " Everything you need to know about arduino code ". [online] [cit. 25.02.2023]. Dostupné z: <https://www.circuito.io/blog/arduino-code/>
28. Blynk. "Welcome to Blynk. Introduction " [online] [Citace: 04.03.2023] <https://docs.blynk.io/en/>
29. Luca Cavallin. "Build a good-looking IoT device." [online] [Citace: 05.03.2023] <https://itnext.io/build-a-neat-iot-product-with-the-particle-photon-blynk-and-a-3d-printed-enclosure-63d78d871778>

## 8 Přílohy

### Příloha 1: Konfigurační portál pro WiFi.

```
#include <WiFiClient.h>
#include <WebServer.h>
#include <DNSServer.h>
WebServer server(WIFI_AP_CONFIG_PORT);
DNSServer dnsServer;
const byte DNS_PORT = 53;
const char* config_form = R"html(
<!DOCTYPE HTML>
<html>
<head>
  <title>WiFi setup</title>
  <style>
  body {
    background-color: #fcfcfc;
    box-sizing: border-box;
  }
  body, input {
    font-family: Roboto, sans-serif;
    font-weight: 400;
    font-size: 16px;
  }
  .centered {
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    padding: 20px;
    background-color: #ccc;
    border-radius: 4px;
  }
  td { padding:0 0 0 5px; }
  label { white-space:nowrap; }
  input { width: 20em; }
  input[name="port"] { width: 5em; }
  input[type="submit"], img { margin: auto; display: block;
width: 30%; }
  </style>
</head>
<body>
<div class="centered">
  <form method="get" action="config">
    <table>
      <tr><td><label for="ssid">WiFi SSID:</label></td>
<td><input type="text" name="ssid" length=64
required="required"></td></tr>
```



```

else
    dnsServer.start(DNS_PORT, BOARD_CONFIG_AP_URL,
WiFi.softAPIP());
    DEBUG_PRINT(String("AP URL: ") + BOARD_CONFIG_AP_URL);
endif
server.on("/", []() {
    server.send(200, "text/html", config_form);
});
server.on("/config", []() {
    String ssid = server.arg("ssid");
    String ssidManual = server.arg("ssidManual");
    String pass = server.arg("pass");
    if (ssidManual != "") {
        ssid = ssidManual;
    }
    String token = server.arg("blynk");
    String host = server.arg("host");
    String port = server.arg("port");
    String Email = server.arg("email");///
    String phone = server.arg("phone");///
    DEBUG_PRINT(String("E-mail: ") + Email);
    DEBUG_PRINT(String("Phone: +420") + phone);
    String content;
    unsigned statusCode;
    DEBUG_PRINT(String("WiFi SSID: ") + ssid + " Pass: " +
pass);
    DEBUG_PRINT(String("Blynk cloud: ") + token + " @ " + host
+ ":" + port);
    if (token.length() == 32 && ssid.length() > 0) {
        configStore.flagConfig = false;
        CopyString(ssid, configStore.wifiSSID);
        CopyString(pass, configStore.wifiPass);
        CopyString(token, configStore.cloudToken);
        if (host.length()) {
            CopyString(host, configStore.cloudHost);
        }
        if (port.length()) {
            configStore.cloudPort = port.toInt();
        }
        CopyString(Email, configStore.email);
        CopyString(phone, configStore.phone);
        content = R"json({"status":"ok","msg":"Configuration
saved"})json";
        statusCode = 200;
        server.send(statusCode, "application/json", content);
        BlynkState::set(MODE_SWITCH_TO_STA);
    } else {
        DEBUG_PRINT("Configuration invalid");
        content = R"json({"status":"error","msg":"Configuration
invalid"})json";
        statusCode = 404;
    }
}

```

```

        server.send(statusCode, "application/json", content);
    }
});
server.on("/board_info.json", []() {
    char buff[256];
    snprintf(buff, sizeof(buff),
R"json({"board": "%s", "vendor": "%s", "tpl_id": "%s", "fw_ver": "%s", "hw_ver": "%s"})json",
        BOARD_NAME,
        BOARD_VENDOR,
        BOARD_TEMPLATE_ID,
        BOARD_FIRMWARE_VERSION,
        BOARD_HARDWARE_VERSION
    );
    server.send(200, "application/json", buff);
});
server.on("/reset", []() {
    BlynkState::set(MODE_RESET_CONFIG);
    server.send(200, "application/json",
R"json({"status": "ok", "msg": "Configuration reset"})json");
});
server.on("/reboot", []() {
    restartMCU();
});
server.begin();
while (BlynkState::is(MODE_WAIT_CONFIG) ||
BlynkState::is(MODE_CONFIGURING)) {
    dnsServer.processNextRequest();
    server.handleClient();
    if (BlynkState::is(MODE_WAIT_CONFIG) &&
WiFi.softAPgetStationNum() > 0) {
        BlynkState::set(MODE_CONFIGURING);
    } else if (BlynkState::is(MODE_CONFIGURING) &&
WiFi.softAPgetStationNum() == 0) {
        BlynkState::set(MODE_WAIT_CONFIG);
    }
}
server.stop();
}
void enterConnectNet() {
    BlynkState::set(MODE_CONNECTING_NET);
    DEBUG_PRINT(String("Connecting to WiFi: ") +
configStore.wifiSSID);
    WiFi.begin(configStore.wifiSSID, configStore.wifiPass);
    unsigned long timeoutMs = millis() +
WIFI_NET_CONNECT_TIMEOUT;
    while ((timeoutMs > millis()) && (WiFi.status() !=
WL_CONNECTED))
    {
        delay(100);
    }
}

```

```

    if (!BlynkState::is(MODE_CONNECTING_NET)) {
        WiFi.disconnect();
        return;
    }
}
if (WiFi.status() == WL_CONNECTED) {
    BlynkState::set(MODE_CONNECTING_CLOUD);
} else {
    BlynkState::set(MODE_CONFIGURING);
}
}
void enterConnectCloud() {
    BlynkState::set(MODE_CONNECTING_CLOUD);
    Serial.print("enterConnectCloud()");
    Blynk.disconnect();
    Blynk.config(configStore.cloudToken, configStore.cloudHost,
configStore.cloudPort);
    Blynk.connect(0);
    unsigned long timeoutMs = millis() +
WIFI_CLOUD_CONNECT_TIMEOUT;
    while ((timeoutMs > millis()) &&
        (Blynk.connected() == false))
    {
        Blynk.run();
        if (!BlynkState::is(MODE_CONNECTING_CLOUD)) {
            Blynk.disconnect();
            return;
        }
    }
    if (Blynk.connected()) {
        BlynkState::set(MODE_RUNNING);

        if (!configStore.flagConfig) {
            configStore.flagConfig = true;
            config_save();
            DEBUG_PRINT("Configuration stored to flash");
        }
    } else {
        BlynkState::set(MODE_CONFIGURING);
    }
}
}
void enterSwitchToSTA() {
    BlynkState::set(MODE_SWITCH_TO_STA);
    DEBUG_PRINT("Switching to STA...");
    WiFi.mode(WIFI_OFF);
    delay(1000);
    WiFi.mode(WIFI_STA);
    BlynkState::set(MODE_CONNECTING_NET);
}
}
void enterError() {
    BlynkState::set(MODE_ERROR);
}
}

```

```

unsigned long timeoutMs = millis() + 10000;
while (timeoutMs > millis() || g_buttonPressed)
{
    delay(10);
    if (!BlynkState::is(MODE_ERROR)) {
        return;
    }
}
DEBUG_PRINT("Restarting after error.");
delay(10);
restartMCU();
}

```

## Příloha 2: Konfigurační úložiště pro systém.

```

struct ConfigStore {
    uint32_t magic;
    char version[9];
    uint8_t flagConfig:1;
    uint8_t flagApFail:1;
    uint8_t flagSelfTest:1;
    char wifiSSID[34];
    char wifiPass[64];
    char cloudToken[34];
    char cloudHost[34];
    uint16_t cloudPort;
    char email[34];
    char phone[11];
    uint16_t checksum;
} __attribute__((packed));
ConfigStore configStore;
const ConfigStore configDefault = {
    0x326C6E6B, // 0x626C6E6B,
    BOARD_FIRMWARE_VERSION,
    0, //use this config - put 1
    0, 0,
    "crew deck", //wi-fi
    "Temp123!", //pass
    "", // Blynk auth
    "blynk.tk", 8080,
    "ship.tmp@gmail.com", // "",
    "1234567890", // telephone
    0
};
#include <Preferences.h>
Preferences preferences;
void config_load()
{
    memset(&configStore, 0, sizeof(configStore));
    preferences.getBytes("config", &configStore,
sizeof(configStore));
    if (configStore.magic != configDefault.magic) {

```

```

    DEBUG_PRINT("Using default config.");
    configStore = configDefault;
    return;
}
}
bool config_save()
{
    preferences.putBytes("config", &configStore,
sizeof(configStore));
    return true;
}
bool config_init()
{
    preferences.begin("blynk", false);
    config_load();
    return true;
}
void enterResetConfig()
{
    DEBUG_PRINT("Resetting configuration!");
    configStore = configDefault;
    config_save();
    BlynkState::set(MODE_WAIT_CONFIG);
}
template<typename T, int size>
void CopyString(const String& s, T(&arr)[size]) {
    s.toCharArray(arr, size);
}
}

```