

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Wi-Fi sensor pro dohledový systém

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Milan HOTOVEC**
Osobní číslo: **A17B0222P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Wi-fi sensor pro dohledový systém**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prozkoumejte typy sensorů vhodných pro domácí dohledový systém a jejich interface.
2. Prozkoumejte možnosti a navrhnete vhodný komunikační protokol mezi sensorovým modulem a hlavní jednotkou.
3. Pro kontroler ESP8266 pak implementujte software realizující funkci wifi-sensoru s důrazem na možnost rozšiřitelnosti a možnost připojení různých čidel změnou uživatelského nastavení.
4. Ověřte funkci wifi-sensoru a dokumentujte uživatelská nastavení pro různá čidla.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

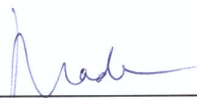
Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Tomáš Mainzer, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **7. října 2019**
Termín odevzdání bakalářské práce: **7. května 2020**



Doc. Dr. Ing. Vlasta Radová
děkanka



Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 15. října 2019

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2020

Milan Hotovec

Abstract

The aim of this thesis is to familiarize with the microchip ESP8266 and the types of sensors that can be connected to it. Furthermore, with communication options and existing protocols that can be used in implementation. Then design and create a suitable implementation of Wi-Fi sensor with emphasis on the most general use and easy extensibility.

Abstrakt

Cílem bakalářské práce je seznámit se s mikročipem ESP8266 a typy senzorů, které k němu je možné připojit. Dále s možnostmi komunikace a již existujícími protokoly, které je možné využít při realizaci. Následně navrhnout a vytvořit vhodnou realizaci Wi-Fi senzoru s důrazem na co nejobecnější použití a snadnou rozšiřitelnost.

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Tomášovi Mainzerovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

Obsah

1	Úvod	10
2	Internet věcí	11
3	Senzory	12
3.1	Digitální	12
3.1.1	Senzor pohybu	12
3.1.2	Požární senzor	13
3.2	Analogové	14
3.2.1	Půdní vlhkoměr	14
3.2.2	Měřič vzdálenosti	14
3.2.3	Senzor plynů	15
3.3	Sběrníkové	16
3.3.1	OneWire	16
3.3.2	I2C	17
4	Mikročip ESP8266	18
4.1	ESP-01	19
4.2	ESP-12E	20
4.3	NodeMCU v3.0	20
5	Možnosti programování Firmware	22
5.1	Arduino	22
5.2	Lua	23
5.3	MicroPython	23
6	Komunikační protokoly	24
6.1	MQTT	24
6.2	HTTP	25
6.3	Vlastní protokol	26
7	Možné způsoby rozšíření	27
7.1	SPIFFS	27
7.2	OTA nahrání Firmware	27
7.3	Konfigurace přes webové rozhraní	28
8	Návrh realizace	29

9	Jak inicializovat vývojové prostředí	30
9.1	Arduino IDE	30
9.2	SPIFFS	31
9.3	Arduino Json	31
9.4	ESP Async Web Server	31
9.5	PubSubClient	31
9.6	Uptime	31
9.7	Doporučené testovací elementy	32
10	Chování navrženého firmwaru	33
10.1	Výchozí hodnoty	34
10.2	Inicializace zařízení	34
10.3	Kontinuální chod	37
10.4	Formát výstupu dat do MQTT brokeru	38
11	Možnosti aktualizace	39
11.1	Vývojové prostředí Arduino IDE	39
11.2	Webové rozhraní	39
12	Aktuální možnosti senzorů	40
12.1	Lokální akce	40
12.2	Ukázky nastavení senzorů	40
13	Konfigurace	44
13.1	Konfigurace přes webové rozhraní	44
13.1.1	Úvodní stránka	45
13.1.2	Wifi	47
13.1.3	MQTT	49
13.1.4	Sensors	51
13.1.5	Tools	57
13.2	Přímá editace konfiguračního souboru	61
13.2.1	Objekt WifiSTA	61
13.2.2	Objekt WifiAP	61
13.2.3	Objekt MQTT	62
13.2.4	Pole Sensors	62
14	Jednotlivé soubory projektu	64
14.1	Samotný Firmware	64
14.2	Webové rozhraní	65

15 Jak přidat podporu pro další senzory	66
15.1 Seznam senzorů	66
15.2 Inicializace	66
15.3 Odebrání senzoru	67
15.4 Měření hodnoty	67
15.5 Hlášení volných senzorů do webového rozhraní	67
15.6 Validátor	67
16 Bezpečnost	68
17 Možné způsoby budoucího rozšíření	69
17.1 Přesnější čas	69
17.2 Logický a analogový filtr	69
17.3 Záznam dat	69
17.4 Záložní baterie	70
18 Závěr	71
Literatura	72

1 Úvod

V moderní době je trendem vše připojovat pomocí bezdrátové komunikace. Pomineme-li klasická zařízení, jako počítač, či mobilní telefon, mohou být na Wi-Fi připojena i zařízení jako lednička, žaluzie, či zrcadlo. Klasická síť senzorů, například pro zabezpečení domu, je vedena pomocí drátového rozhraní, ale vzhledem k vysoké dostupnosti a stále klesajícím cenám Wi-Fi zařízení se přímo nabízí najít způsob, jak vylepšit existující model.

Cílem této práce je navrhnout obecný Wi-Fi senzor, založený na mikročipu ESP8266 od společnosti Expressif Systems, který by dokázal svou univerzálností nahradit některé zastaralé systémy. Díky použití bezdrátové technologie by se jednalo o snížení ceny realizace zabezpečení domu, odpadá nutnost ke každému senzoru vést datový vodič, je třeba řešit pouze napájení. Wi-Fi senzor cílí na běžné uživatele, proto disponuje příjemným webovým konfiguračním rozhráním pro snazší ovládání.

2 Internet věcí

Jedná se o síť fyzických zařízení od vozidel, přes domácí spotřebiče, až po chytré žárovky, která jsou vybavena síťovou funkcionalitou, díky které dokáží přijímat, nebo odesílat data, získaná například ze senzorů. Smyslem je ulehčit člověku práci, například můžeme mít chytrý venkovní teploměr, který získává aktuální informace o teplotě a odesílá je do našeho chytrého zrcadla, které nám teplotu vizualizuje a doporučí nám, jak se obléknout, vzhledem k aktuální a výhledové teplotě, dostupné z předpovědí na internetu.

Příbytkem budoucnosti je tzv. Chytrý dům. V jádru je to běžný dům, ale obsahuje nespočet chytrých zařízení, díky kterým je jeho údržba a obývání příjemnější. Mezi tato zařízení bezpochyb mohl patřit i Wi-Fi senzor, popsany v této práci. Pomocí něj je možné získávat data pro úpravu vnitřní teploty, automatické zalévání květin, detekce úniku nebezpečných plynů, či požáru. Asi nejdůležitější je možnost použít jej i jako pohybové čidlo, při dostatečném rozmístění po objektu se jedná o plnohodnotný zabezpečovací systém.

3 Senzory

Aby Wi-Fi senzor byl vůbec k něčemu užitečný a posílal nějaká smysluplná data, bylo by vhodné k němu nějaký senzor připojit. Senzor je zařízení, které získává určité informace z okolního prostředí a tyto informace převádí na něco srozumitelného, v nejjednodušším případě to může být pouze binární informace Ano, či Ne (reprezentovaná výstupní hodnotou High, či Low). V dnešní době je k dispozici velké množství senzorů, jejich kvalita se také odvíjí od cílové skupiny zákazníků, od velmi jednoduchých pro domácí použití, až po průmyslové, či armádní, které jsou mnohem přesnější a trvanlivější. Tyto skutečnosti ovšem velice navyšují jejich cenu. Pro nejobecnější použití budeme předpokládat použití v domácnosti.

Senzory mají také rozdílné připojení, jedná se o způsob, jakým nám předají svá naměřená data. Mezi nejzákladnější rozdělení patří senzory analogové, digitální a sběrníkové, níže je popsáno několik významných zástupců, které by bylo možné prakticky využít v chytrém domě.

3.1 Digitální

Digitální senzory jsou senzory, jejichž výstupem je informace v digitální podobě, vedena například přes jeden vodič jako sériová komunikace. Ale i přes takovouto komunikaci může být výsledkem senzoru desetinné číslo, pouze přenesené digitálně, více u sběrníkových senzorů. V nejjednodušším případě se však může jednat pouze o binární výsledek, kdy se na výstupním vodiči vyskutekuje například hodnota Low a při sepnutí senzoru se hodnota změní na High. Tohoto je dosaženo většinou pomocí komparace referenční hodnoty, vůči aktuálně naměřené. Pokud je aktuální hodnota větší, než referenční, na výstupu bude hodnota High.

3.1.1 Senzor pohybu

Také známý jako PIR čidlo (PIR je zkratkou pro Passive Infrared, nebo-li pasivní infračervené čidlo), slouží k detekci pohybu živých objektů. Pracuje na principu snímání infračerveného záření, každý objekt vyzařuje teplo ve formě záření, které je schopný tento senzor zaznamenat. Zjednodušeně řečeno, umí zjistit, zda-li se před ním hnul nějaký živý objekt, ale již neumí říci kde. Existuje modul, který takovéto čidlo obsahuje, jeho označení je HC-SR501 (viz brázek 3.1). Je napájen 5V (podporuje až 12V) a v klidovém

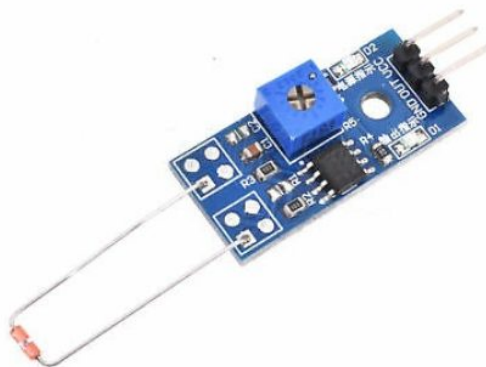
stavu potřebuje méně než $50\mu\text{A}$. Jeho dosah je 7m a reakční doba 0,2 vteřiny. Obsahuje trimry na nastavení intenzity a délky sepnutí výstupu. Výstupní informace je předána pomocí digitálního signálu v logice 3,3V (pouze jako binární hodnota). [14]



Obrázek 3.1: Modul pohybového senzoru, HC-SR501

3.1.2 Požární senzor

Tento senzor snímá teplotu v místnosti, proto je klíčové jeho umístění někam, kde předpokládáme vznik požáru. Například nad sporák. Výstupní informací senzoru je logická hodnota Ano/Ne. Senzor obsahuje trimr, pomocí kterého můžeme nastavit práh citlivosti. Napájení je možné na 3,3V, ale i na 5V. Ukázka senzoru, viz obrázek 3.2.[8]



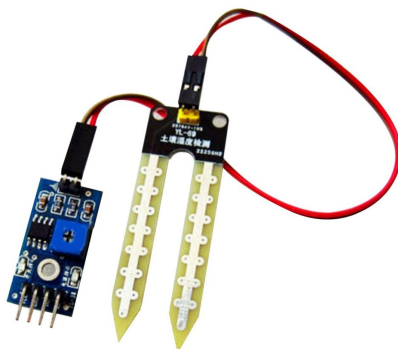
Obrázek 3.2: Modul požárního senzoru

3.2 Analogové

Analogové senzory jsou senzory, které svou naměřenou informaci předávají v analogové formě, zjednodušeně řečeno nejsou schopny rozhodnout, jak s naměřenou hodnotou naložit a toto rozhodnutí nechají na mikrokontroléru. Analogové senzory mají malou nevýhodu v podobě přesnosti. Jsou schopny naměřenou hodnotu reprezentovat pouze v omezeném rozsahu, pokud je rozsah příliš malý, může i velká změna naměřené hodnoty způsobit téměř nulovou změnu na výstupu senzoru.

3.2.1 Půdní vlhkoměr

Jedná se o modul, který měří intenzitu vlhkosti v půdě, hodnotu předává jako analogový výstup. Obsahuje i pin pro digitální výstup, kde je možnost nastavit pomocí trimru práh vlhkosti a pracovat pomocí binárního výstupu. Napájení je možné na 3,3V, ale i na 5V. Samotný modul je složen ze dvou částí, ze sondy a z desky (viz obrázek 3.3), která přeposílá naměřená data.[12]

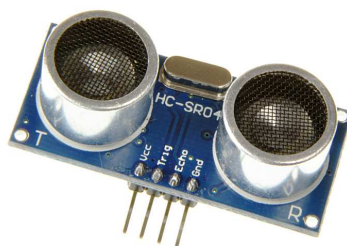


Obrázek 3.3: Modul půdního vlhkoměru

3.2.2 Měřič vzdálenosti

Modul s ultrazvukovým měřičem vzdálenosti nese označení HC-SR04 (viz obrázek 3.4), jeho použití je hlavně v robotech, ale jedná se o univerzální modul pro snímání vzdálenosti, takže se dá použít například i na detekci naplněnosti. Je napájen 5V a jeho typická spotřeba je 15mA, umí pracovat od vzdálenosti 2cm až po 4m, ale ideální vzdálenost je zhruba 2m.

Pracuje na principu echolokace, vyšle ultrazvukový signál a podle doby, za jakou se vrátil do přijímače (odražen od překážky), je vypočítána vzdálenost. Tento výsledek je závislý na konstantě rychlosti šíření zvuku ve vzduchu. [19]



Obrázek 3.4: Modul ultrazvukového měřiče vzdálenosti, HC-SR04

3.2.3 Senzor plynů

Tento senzor je obsažen v modulu MQ-2 (viz obrázek 3.5). Je vhodný pro detekci úniku nebezpečných plynů, jako jsou například zemní plyn, metan, butan, propan, či vodík. Vhodným umístěním (třeba v blízkosti plynového potrubí) je možné zajistit rychlejší detekci úniku.

Výstup je možný jak digitální, tak analogový. Analogová hodnota je v rozsahu 0 - 1023. Při začátku měření je nutno počkat 3 minuty, než se senzor ustálí, po tuto dobu jsou výsledky velmi nepřesné.

Napájení je na 5V. [9]



Obrázek 3.5: Modul senzoru plunů, MQ-2

3.3 Sběrnice

Technicky se vlastně již nejedná o typ senzorů jako takových, spíše o způsob jakým předávají data. Nespornou výhodou je možnost připojit více senzorů na stejnou sběrnici. V případě sítě senzorů je sběrnice řízena jedním zařízením (master) a je na ní připojeno více dalších zařízení, senzorů (slave), které neřídí komunikaci po sběrnici, ty pouze přijímají požadavky od mastera, případně mu na ně odpovídají.

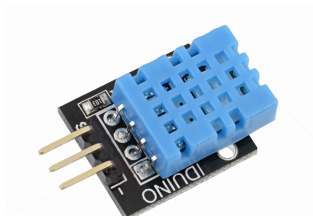
Sběrnice založený senzor je vlastně pouze obyčejný senzor, obohacen funkcionalitou připojit se na sběrnici, který ale svou naměřenou hodnotu neodesílá přímo, ale někde si jí uloží (většinou do určitého registru), dokud si o tuto hodnotu master nezažádá.

Níže jsou popsány dvě nejzajímavější sběrnice pro senzory, spolu se zástupci z řad senzorů.

3.3.1 OneWire

Sběrnice byla navržena firmou Dallas Semiconductor, umožňuje připojit několik zařízení k řídicí jednotce prostřednictvím pouhých dvou vodičů. Sběrnice má jeden řídicí obvod (master) a jedno či více ovládaných zařízení (slave). Všechny obvody jsou zapojeny na společnou zem a také paralelně na společný datový vodič. Tento datový vodič je připojen přes odpor cca 5k Ohm na napájecí napětí a „zdvihá“ tak sběrnici do log. 1.[16]

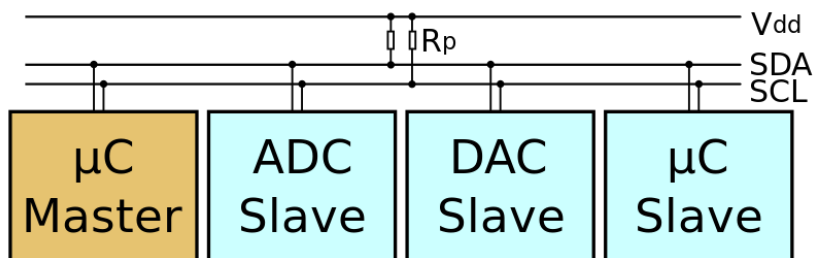
A jako zástupce, senzor teploty a vlhkosti (viz obrázek 3.6). Tento senzor nese označení DHT11 (existuje ještě jeho lepší verze DHT22), senzor pracuje na napětí 5V a jeho spotřeba je 0,3mA při měření a 100 μ A ve stand-by módu. Rozsah je 0 - 80 stupňů Celsia pro teplotu a 20 - 90 % pro vlhkost. Nebezpečí při použití tohoto senzoru může číhat v podobě knihovny DHT, která je k němu doporučena. Je hezké, že nemusíme příliš řešit komunikaci se senzorem a jen se zeptáme knihovní funkce na teplotu. Problém je, že tato funkce je blokovácí a může trvat až 250 mS, což je nepříjemně dlouhá doba. [17]



Obrázek 3.6: Modul senzoru teploty a vlhkosti, DHT11

3.3.2 I2C

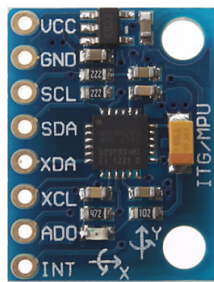
Každému zařízení na sběrnici je přiřazena jednoznačná adresa. Pomocí této adresy jsou na sběrnici identifikovány. Teoretické maximum připojených zařízení je 128. Jedno z připojených zařízení pracuje v režimu „master“, ostatní zařízení pak v režimu „slave“. Do režimu „master“ se většinou konfiguruje mikrořadič či mikroprocesor, který sbírá data z ostatních zařízení (senzory), popřípadě tato zařízení řídí (LED či LCD panely, regulované větráčky, hodiny reálného času). Kromě datového vodiče SDA (Signal Data) a vodiče SCL (Signal Clock) přenášejícího hodinový signál je nutné všechna zařízení propojit společnou signálovou zemí – GND (Ground). Vizualizace sběrnice, viz obrázek 3.7. [15]



Obrázek 3.7: Vizualizace zařízení na sběrnici I2C

A jako zástupce senzorů připojitelných na I2C sběrnici, tří osý gyroskop a akcelerometr (viz obrázek 3.8).

Použitý čip je MPU-6050. Jeho napájení je 3-5 V, obsahuje 16 bitový AD (Analog-Digital) převodník. Takže výstupní hodnota jednoho naměřeného čísla je maximálně v rozsahu 16 bitů.[18]



Obrázek 3.8: Tří osý gyroskop, MPU-6050

4 Mikročip ESP8266

Jedná se o jeden z nejběžnějších mikročipů, které slouží k připojení k bezdrátové Wi-Fi komunikaci. Společnost, která čip vyrábí se jmenuje Expressif Systems. Obrovskou výhodou tohoto čipu je nízká pořizovací cena a také fakt, že přímo v sobě obsahuje mikrokontrolér, kterým se dá řídit, a díky tomu není nutné připojovat další hardware. Podporuje několik různých způsobů použití, umí se chovat jako přístupový bod (AP mód), kdy řídí komunikaci. Vytvoří si vlastní síť, kterou je možné zabezpečit, má integrovaný DHCP server (zkratka DHCP znamená Dynamic Host Configuration Protocol), pomocí kterého dokáže přidělit připojeným zařízením jejich IP adresu (zkratka IP znamená Internet Protocol). Nebo se chová jako stanice, připojí se k existující Wi-Fi síti a tím se stává plnohodnotným členem. Popřípadě je možné spustit oba módy současně. Napájení je 3.3V, stejné napětí je použito i pro logiku. [5]

Samotný čip je složen z 32 bitového processoru, který je RISCové architektury (RISC je zkratkou pro Reduced Instruction Set Computer, nebo-li redukovaná instrukční sada), běžícího na taktu 80 MHz. Paměť je rozdělena na několik celků, je zde 32 KB pro instrukce, 32 KB instrukční cache, 80 KB pro uživatelská data a 16 KB pro systémová data. [5]

Programovatelná flash paměť může být o velikosti až 16 MB, běžně je k dispozici něco v rozmezí 512 Kb, až 4 MB. Toto se odvíjí od použitého typu modulu. [5]

Bezdrátový přenos pomocí Wi-Fi je dostupný ve standardech normy IEEE 802.11b/802.11g/802.11n. Pro zabezpečení je možné použít WEP, WPA a WPA2 (WEP, WPA a WPA2 jsou zabezpečovací protokoly pro komunikaci po síti Wi-Fi), případně je možné použít i nezabezpečené spojení. [5]

Pro připojení je z pouzdra vyvedeno 16 digitálních GPIO pinů (GPIO je zkratkou pro General Purpose Input Output, nebo-li Univerzální Vstupně Výstupní pin). [5]

K dispozici je také SPI (Serial Peripheral Interface, nebo-li Sériové periferní rozhraní), které je vyvedeno na příslušných pinech, bohužel se liší modul od modulu, je lepší dohledat v dokumentaci pro konkrétní modul. [5]

K dispozici je také I2C sběrnice, realizovaná pomocí softwarové implementace. [5]

Jedno z nejdůležitějších je UART (zkratka pro Universal asynchronous receiver-transmitter, nebo-li univerzální asynchronní přijímač-vysílač), opět vyvedený na příslušných pinech, k nalezení v dokumentaci. Slouží pro programování

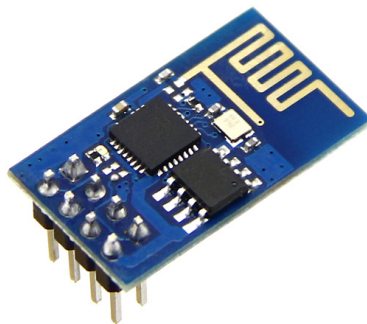
a komunikace se zařízením, jedná se vlastně o sériovou komunikaci bez synchronizace, na některých modulech je k dispozici převodník z UART na USB (zkratka pro Universal Serial Bus, nebo-li univerzální seriovou sběrnici), která se nachází na všech moderních počítačích.

Poslední důležitou součástí je vývod na 10 bitový analogový pin značený ADC (zkratka pro Analog to Digital Convertor), zde je možné připojit analogové senzory, jejich hodnoty mohou být v rozsahu 0 až 1023 hodnot, lepší citlivost nám zabudovaný ADC nenabízí, ale je možné v případě potřeby připojit externí.

Mikročip se nachází v několika běžně dostupných modulech, v modulu je obsažena kromě samotného čipu hlavně anténa (většinou leptaná do desky plošných spojů, v lepším případě keramická) a piny, které jsou připojeny na důležité vývody z mikročipu (jejich počet se liší modul od modulu). Nejzajímavější z modulů jsou popsány níže.

4.1 ESP-01

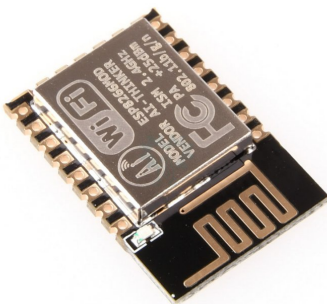
Tento modul patří mezi nejběžnější a nejlépe dostupné. Obsahuje pouze základní hardware pro svou funkci, k dispozici je pouze 8 pinů (minimum pro využitelnost). Mimo zemi, napájení, vývodů pro sériovou komunikaci přes UART (což jsou Rx a Tx), resetu a pinu pro zapnutí čipu je zde vyveden pouze 1 digitální pin. Tím pinem je GPIO2, je zde vyveden i GPIO0, ale ten se používá hlavně při nahrávání firmware. Vzhledem k jeho omezenosti, z hlediska výstupů není pro tuto práci ideální, ale pro menší projekty, například jen s jedním senzorem by se dal využít. Ještě stojí za zmínku, že anténa je zde leptaná do desky plošných spojů (viz obrázek 4.1). Spotřeba se zde pohybuje okolo 80 mA, při maximálním zatížení by neměla přesáhnout 170mA. V klidovém stavu ovšem může být i menší než $10\mu\text{A}$. Další omezení se objevuje v podobě velikosti flash paměti, která je zde pouze 512 kB. [3]



Obrázek 4.1: Modul ESP-01

4.2 ESP-12E

Tento modul je mnohem univerzálnější, kromě základních pinů, které má vyvedeny i ESP-01 (bez nich by byl vlastně k ničemu), je k dispozici mnoho dalších. Například vývody na SPI, či 9 digitálních pinů. Nejzajímavější je vývod pro ADC, který je možné využít pro připojení analogového čidla. Velká změna od modelu ESP-01 je ve velikosti flash paměti, typicky se zde nachází paměť o velikosti 4 MB. Tato velikost je plně dostačující pro běžné projekty a i pro potřeby této práce. Spotřeba je stejná jako u ESP-01. [4] Ukázka modulu na obrázku 4.2.

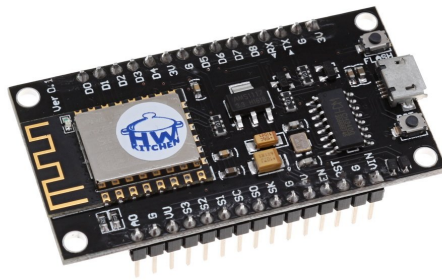


Obrázek 4.2: Modul ESP-12E

4.3 NodeMCU v3.0

Nejedná se přímo o modul, ale rovnou o vývojovou desku (viz obrázek 4.3). Vnitřně obsahuje modul z řady ESP-12x. Takže přirozeně disponuje flash

paměť o velikosti 4 MB. Bohužel, maximální proud se zvedl na 350 mA, ale pracovní zůstává stejný na 80 mA. Výhodou je možnost připojení rovnou na USB počítače. Obsahuje USB port sloužící k napájení desky a k sériové komunikaci. Deska již obsahuje převodník na UART (konkrétně mikročip CH340). K dispozici jsou i tlačítka pro reset ESP8266 a povolení přepsání firmware (je to bohužel řešeno trochu nešikovně, je třeba tlačítko držet po celou dobu nahrávání firmware). Samotná deska obsahuje vývody na 10 GPIO, rozhraní I2C, SPI a analogový pin. Bohužel máme sice k dispozici hojný počet digitálních pinů, ale stejně nejde všechny bezpečně použít, je třeba v dokumentaci dohledat, které piny nesmí být při startu uzemněny, či připojeny na logickou hodnotu 1. Ve výsledku máme zhruba 5 bezpečných pinů, a i z těch musíme 2 vyřadit pro realizaci I2C sběrnice. Ale i tak je deska pro vývoj naprosto ideální (pomineme-li tlačítko pro flash, ale dá se nahradit přepínačem a tím problém vyřešit). [1]



Obrázek 4.3: Deska NodeMCU v3

5 Možnosti programování Firmware

K naprogramování ESP8266 se nabízí několik možností, níže budou vypsány ty nejzajímavější. Ale je vhodné nejdříve vysvětlit co je vlastně firmware a proč jej programovat.

Klasický počítač má svůj operační systém a ten se stará o správnou funkčnost počítače jako celku, dělá co je mu nařízeno. Menší zařízení také potřebují něco, co řídí jejich chod a tuto pozici zastává právě jejich firmware. Je to jednoduchý program, obsahující sled instrukcí, který definuje veškeré možné stavy, které mohou nastat, a jak na daný stav reagovat. Většinou obsahují hlavní řídicí smyčku. Jedná se o nekonečný cyklus, který obsahuje řídicí konstrukce pro chod zařízení. Například, pokud je na vstupním pinu hodnota LOW (pin byl připojen na zem), či HIGH (pin byl připojen na kladný potenciál napájení), jak se má zachovat. Může tímto způsobem nastavit hodnotu jiného pinu, který například rozsvítí diodu.

Pomocí dostatečně sofistikovaného firmware jsme schopni zařízení připravit na všechny eventuality, které pro něj mohou nastat a říci mu, jak se k nim má zachovat. Proto je i cílem této práce navrhnout dostatečně vybavený firmware, který bude plnit co se po něm chce a bude dostatečně robustní, aby dokázal zvládnout nepřetržitý chod zařízení.

Při zakoupení zařízení je možné, že v něm už firmware bude nahrán, většinou se jedná o firmware, který pracuje pomocí tzv. AT příkazů. Takovýto firmware je užitečný, pokud je zařízení připojeno k jinému kontroléru a slouží pouze jako vysílač či přijímač zpráv přes Wi-Fi. V našem případě se snažíme docílit řešení, které nepotřebuje další kontrolér a proto je takovýto firmware naprosto nevhodný.

5.1 Arduino

Pojem Arduino, není vlastně název programovacího jazyka jako takového, ale mezi lidmi, kteří jej používají se velmi rozšířil. Pod správným názvem se jedná o framework Wiring, pro jazyk C++. Vzhledem k neuvěřitelné komplexnosti, se o frameworku wiring často hovoří jako o samostatném programovacím jazyku, i když to není správné pojmenování, ale velmi zažité a lze se s tím setkat v mnoha výukových materiálech nejen internetových, ale i

tištěných. [20]

Tento způsob programování, je velmi dobrý v tom, že je přímo podporován v Arduino IDE (Integrated Development Environment, nebo-li integrované vývojové prostředí), pro které existuje několik internetových učebnic, návodů a příkladů. Samotná složka s projekty by se měla nacházet v Dokumentech aktuálního uživatele, v podsložce Arduino a v ní složky s projekty.

Pro vývoj s ESP8266 je potřeba přidat v manažeru desek v Arduino IDE desky s ESP8266, kde jsou zahrnuty všechny běžně dostupné moduly s předem nastavenou konfigurací pro nahrávání firmware, jako je velikost paměti, takt procesoru atp.

Samotný firmware je do zařízení nahrán pomocí skriptu v jazyce Python.

Další nespornou výhodou je vcelku rozsáhlé množství knihoven dostupných jak od vývojáře Arduina, tak od nadšenců, kteří se o své knihovny podělili. Nevýhodou může být, že knihovny nejsou z oficiální distribuce a proto mohou obsahovat i závažné chyby, ale většina vývojářů přijímá zpětnou vazbu od uživatelů a chyby ve svých knihovnách poměrně rychle opravuje. Například přes vývojářský verzovací systém Gitlab, kde je možné chybu nahlásit pomocí tzv. issues.

5.2 Lua

Jedná se o skriptovací jazyk, který podporuje procedurální a objektové programování. Některé moduly (například NodeMCU od WeMosu) jsou prodávány s předinstalovanou podporou jazyka Lua. Programování je sice příjemné, intuitivnější než C++, ale zase je při použití v ESP často nestabilní a nepředvídatelné.

Například jednoduchý program zasílající každou hodinu data. Takovýto program bude pracovat jeden den a v průběhu druhého dne se ESP prostě zasekne a přestane fungovat. V lepším případě se samo restartuje, v horším je nutné program znovu nahrát.

Programování ESP v jazyce Lua není nejlepší volbou, byť je jeho vývoj poměrně snadný a rychlý.

5.3 MicroPython

Jedná se o upravenou verzi Pythonu 3, která je přizpůsobena chodu na mikrokontrolérech. Obsahuje méně knihoven, než Python 3. Ale zase obsahuje knihovnu machine, která slouží pro komunikaci s hardwarem.

Micropython je na práci poměrně jednoduchý, ale programy pomalé. [13]

6 Komunikační protokoly

Pro spolehlivou komunikaci mezi zařízením a řídicí jednotkou je potřeba vybrat vhodný protokol. Vzhledem k velké rozšířenosti Wi-Fi modulů už je několik takovýchto protokolů k dispozici, většinou ve formě knihoven.

Pokud by nám již existující protokol nevyhovoval, například proto, že je moc pomalý, datově náročný atp. Je možné si navrhnout vlastní. Ale navrhnout dobrý, stabilní, protokol není úplně lehké a může to zabrat mnoho času s ne příliš dobrým výsledkem. Bezpečnější je zvolit cestu již existujícího protokolu, který je odladěn a ideálně běžně používán.

6.1 MQTT

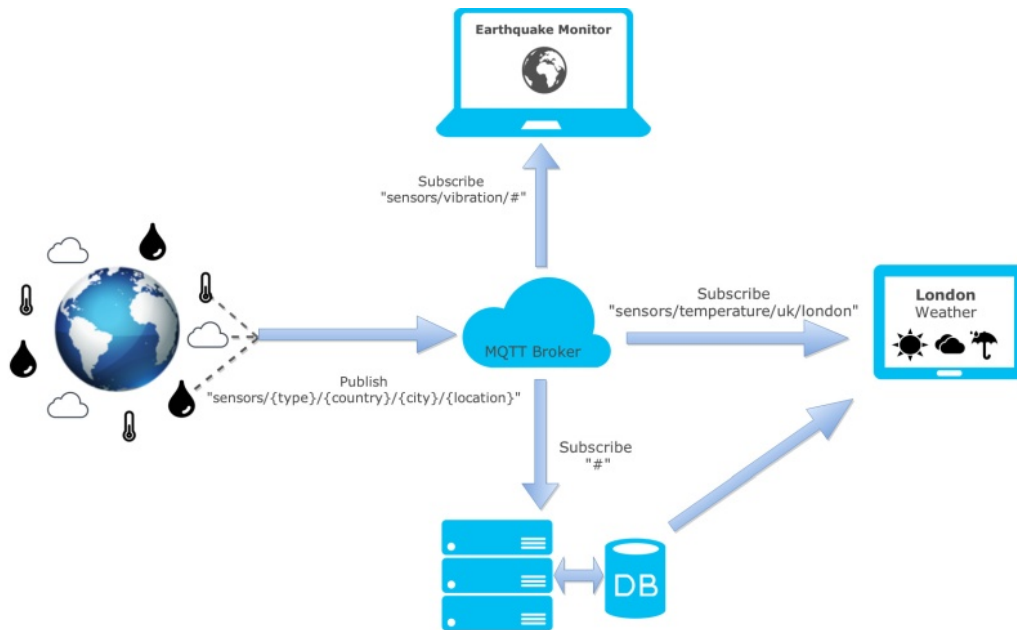
MQTT je zkratka pro MQ Telemetry Transport. Jedná se o mimořádně jednoduchý a nenáročný protokol postavený nad TCP/IP a navržený pro jednoduchá zařízení, úzkou šířku pásma, nespolehlivé sítě nebo sítě s vysokou latencí. Byl vyvinut v roce 1999 doktorem Andy Stanford-Clarkem z IBM a Arlenem Nipperem z firmy Arcom (nyní Eurotech). Od roku 2013 je pod správou sdružení OASIS (The Organization for the Advancement of Structured Information Standards).[11]

Základem MQTT komunikace je systém typu zveřejnit/odebírat novinky (publish/subscribe). Zprávy mohou produkovat například snímače, které je publikují do jednoho centra (MQTT Broker). Každá zpráva je publikována s názvem tématu (topic). Témata jsou hierarchická a oddělena lomítky, jedná se o řetězce v UTF-8, takže diakritika není problém. Například "dům/ložnice/světlo". Zprávy broker rozešle všem klientům, jež si je objednali k odběru (subscribe). Klientem může být webová služba, aplikace nebo konkrétní zařízení (např. mikro-kontrolér, displej. . .), které zprávu dále zpracuje. [11]

Výhodou je, že MQTT si samo hlídá, zda-li je zařízení stále živé. Pokud mu neodešle plánovanou zprávu, je okamžitě považováno za násilně odpojené a posílá se takzvaná závěť.

Navázání spojení je řešeno pomocí zprávy CONNECT, broker odpovídá zprávou CONNACK, tím potvrdí připojení. Pomocí zprávy SUBSCRIBE, následované názvem tématu, se zařízení přihlásí k odběru novinek. Broker potvrdí odesláním SUBACK. Ukončení odběru novinek je zajištěno pomocí zprávy UNSUBSCRIBE, která je brokerem potvrzena odesláním UNSUBACK. Odeslání novinky probíhá zprávou PUBLISH, tu může poslat jak

broker, tak i zařízení. Vizualizace je na obrázku 6.1 [10]



Obrázek 6.1: Vizualizace MQTT [10]

6.2 HTTP

HTTP (Hypertext Transfer Protocol) je internetový protokol určený pro komunikaci se servery. Obvykle používá port 80. Nejrozšířenější je pro přenos webových stránek. [7]

Samotný protokol nepodporuje šifrování, to je třeba řešit jinak, například použitím HTTPS protokolu (Hypertext Transfer Protocol Secure). To je obecně doporučeno k použití na všech webových stránkách. Zabezpečení je dosaženo pomocí protokolu TLS (Transport Layer Security), nebo SSL (Secure Sockets Layer). [6]

Protokol funguje způsobem dotaz-odpověď. Žadatel pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech zařízení apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu. [7]

Tento způsob komunikace je poměrně jednoduché aplikovat i na zasílání výsledků získaných ze senzoru do řídicí jednotky. Ovšem je zde odesíláno velké množství dat navíc.

6.3 Vlastní protokol

Návrh vlastního protokolu může být poměrně složitá práce. Je třeba jej navrhnout tak, aby byl co nejstabilnější. To přirozeně vyžaduje poměrně dlouhodobé testování, aby se předešlo vzniku nechtěných stavů, případně bylo zajištěno jejich korektní ošetření. V případě velmi primitivního protokolu můžeme vycházet z HTTP a pokusit se implementovat textový protokol na bázi dotaz-odpověď, vlastně osekáme HTTP na něco na způsob výčtu základních příkazů. Těmi mohou být GET, pro získání informace o něčem, nebo třeba SEND, pro odeslání nových dat.

Ovšem takovýto vývoj zabere mnoho času a nepřinese nic nového, funkcionality bude velice omezená a budoucí rozšíření by mohlo být vcelku složité. Navíc se jedná pouze o osekání HTTP protokol.

Cestu vlastního protokolu nepovažuji za nejlepší, ale v některých vyjimečných případech, kdy víme, že nikdy nebude třeba protokol upravovat, se může hodit.

7 Možné způsoby rozšíření

Vzhledem k usnadnění práce a hlavně co nejlepší následné údržby již nainstalovaných zařízení je při návrhu toto kritérium zohledněno a práce byla rozšířena o několik maličkostí, které zlepší používání obecného Wi-Fi senzoru.

7.1 SPIFFS

Jedná se vlastně o souborový systém ve flash paměti ESP modulu, tato část paměti není ovlivněna při nahrávání nového programu (nebo by alespoň neměla být při správném nastavení). Z hlediska programátora je to ideální místo kam ukládat trvalá data, jako například kód webové stránky pro komunikaci se zařízením, či jiné grafické doplňky jako např. rastrová či vektorová grafika. Je zde možné uložit i naměřená data ze senzorů. Proto je vhodné zvolit modul ESP s větší flash pamětí, aby bylo dost místa pro SPIFFS.

Asi nejlepší je ukládat si zde konfigurační soubory, aby po výpadku napájení zařízení nebyl nutné znovu konfigurovat a mohlo rovnou pokračovat v činnosti.

Pro zprovoznění je potřeba nahrát do Arduino IDE plugin, který přidá tuto funkcionalitu.

7.2 OTA nahrání Firmware

Zkratka OTA znamená Over The Air, nebo-li přes vzduch, jedná se o druh aktualizace firmwaru zařízení, namísto přes sériovou komunikaci, přes Wi-Fi. Tento způsob aktualizace je mnohem rychlejší než přímé připojení přes programátor. Nespornou výhodou je absence přímého připojení k počítači pomocí kabelu a tím možnost zařízení aktualizovat na dálku, například již nainstalované a umístěné na zahradě, kam sahá signál Wi-Fi. Samozřejmě pro správnou funkčnost je nutné být připojen s počítačem ve stejné Wi-Fi síti.

Tento způsob je ale také vhodný pro vývoj, díky rychlejšímu nahrávání (cca 5 vteřin) oproti sériové komunikaci (cca 30 vteřin) se mnohem rychleji odlaďují chyby. Tímto způsobem je možné nahrávat i data do SPIFFSu, za pomoci rozšíření v Arduino IDE.

7.3 Konfigurace přes webové rozhraní

Aby bylo zařízení co nejuniverzálnější je vybaveno i webovým konfiguračním rozhraním. Výsledná konfigurace je uložena ve SPIFFSu, jako konfigurační soubor. Ve webové konfiguraci by bylo vhodné nastavit IP adresu zařízení a SSID (jméno Wi-Fi) lokální Wi-Fi sítě ke které se má zařízení připojit, případně i její heslo. Dále můžeme nastavit jaký senzor je k zařízení připojen a na jakém pinu. Je zde zobrazen list podporovaných senzorů a pak také volby obecného digitálního senzoru (pro binární informaci) a obecného analogového senzoru.

8 Návrh realizace

Vzhledem k výše zmíněným možnostem je vcelku z čeho vybírat. Jako programovací jazyk, ve kterém se bude firmware vyvíjet, se jeví jako nejvhodnější zvolit C++ s frameworkem Wiring a vývoj provádět v prostředí Arduino IDE. Hlavně díky vysoké dostupnosti knihoven, které mohou velmi ulehčit práci při vývoji a stabilitě napsaného firmwaru. Takové stability například u jazyku Lua dosáhnout nelze (viz Lua).

Pro vývoj se jeví jako nejlepší použít desku NodeMCU v3.0 (viz NodeMCU v3.0), kterou je možné okamžitě připojit k počítači, bez nutnosti dalšího programátoru a s již vyvedenými GPIO piny z ESP8266. U všech ostatních zmíněných je potřeba použít externí programátor, případně moduly pájet (nemají-li piny již připraveny).

Jako komunikační protokol se jeví nejvhodnější MQTT, jedná se o velmi rozšířený protokol, který dokáže pracovat i v nespolehlivých sítích. Díky velmi nízkým nárokům na přenesená data (není třeba posílat více dat, než je nezbytně nutné), jednoduchosti a dostupnosti knihovny v jazyce C pro Arduino systémy se jedná o perfektního kandidáta (viz MQTT). Jako testovací MQTT broker je možné použít open-source program Mosquitto, který je vhodné použít jak na malých, tak velkých zařízeních. [2]

Díky SPIFFSu je vcelku snadné uložit kód webové stránky pro konfiguraci zařízení a hlavně konfiguračního souboru, který si uchová důležité informace, aby zařízení po výpadku napájení dokázalo bez nutnosti zásahu uživatele pokračovat v činnosti tam, kde skončilo.

Při prvním zapnutí by bylo vhodné uvést zařízení do módu, kdy vysílá svou vlastní Wi-Fi, ke které se můžeme připojit a pomocí webového rozhraní prvotně nakonfigurovat zařízení.

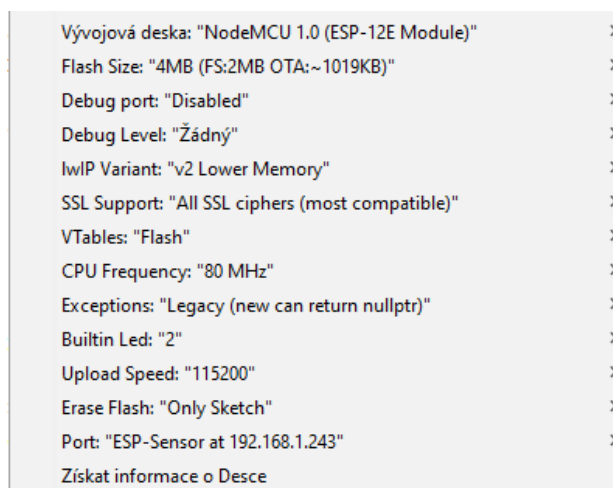
9 Jak inicializovat vývojové prostředí

Vzhledem k možným problémům v kompatibilitě mezi verzemi prostředí a knihoven jsou zde uvedeny včetně verzí a je silně doporučeno je pro úpravu tohoto firmwaru v doporučených verzích používat. V případě SPIFFSU byla nalezena v novější verzi vada, která způsobovala neočekávané chování při nahrávání souborů větších jak zhruba 12 kB. Nakonec došlo k vyřešení problému, použitá verze SPIFFSu nebyla kompatibilní se starší verzí nainstalovaných desek.

9.1 Arduino IDE

Pro vývoj bylo použito IDE ve verzi 1.8.5, aktuálně se jedná o jednu z posledních verzí. Aby bylo možné v IDE provádět vývoj, je potřeba přes manažer desek doinstalovat podporu pro ESP8266, použitá verze je 2.6.3. Pokud probíhá vývoj na NodeMCU v3.0, v deskách stačí vybrat NodeMCU 1.0 (ESP-12E Module), které odpovídá potřebám desky NodeMCU v3.0. Pro jiné desky je třeba dohledat požadovanou konfiguraci.

Na obrázku 9.1 je vidět doporučené nastavení pro desku NodeMCU v3.0.



Obrázek 9.1: Nastavení desky v Arduino IDE

9.2 SPIFFS

Jeho použitá verze je 0.5.0, bohužel se nedá nainstalovat přímo přes správce knihoven a je nutné to udělat ručně. Na adrese githubu vývojáře je k dispozici potřebná verze pluginu, odkaz je k dispozici [zde](#) . Jeho instalace probíhá následovně, ve složce s Arduino projekty je potřeba vytvořit podsložku „tools“, do ní se nahraje celý obsah archivu a po restartu prostředí by se měla v sekci Nástroje objevit položka ESP8266 Sketch Data Upload. V nastavení rozložení flash paměti je pro souborový systém doporučeno použít 2MB.

9.3 Arduino Json

Použitá verze je 6.15.1, od vývojáře Benoit Blanchon. Instalace probíhá přes správce knihoven.

9.4 ESP Async Web Server

Použitá verze je 1.2.3, od vývojáře Me-No-Dev. Instalace probíhá přes správce knihoven.

9.5 PubSubClient

Použitá verze je 2.7.0, od vývojáře Nick O’Leary. Instalace probíhá přes správce knihoven.

9.6 Uptime

Použitá verze je 1.0.0, od vývojáře Yiannis Bourkelis. Instalace probíhá přes správce knihoven.

9.7 Doporučené testovací elementy

V případě, že se zařízení nechová dle očekávání, například pokud senzor neposílá korektní data, je na místě nejdříve vyloučit chybu v senzoru. Například vyzkoušet senzory v naprosto separátním projektu a ideálně i zařízení, nebo nahradit je co nejjednoduššími prvky.

V případě digitálního vstupu je možné použít tlačítko s pull-down rezistorem, signál HIGH by neměl přesáhnout 3.3V, jinak hrozí poškození zařízení. V případě analogového vstupu potenciometer s vhodným rozsahem (doporučeno 5 KOhm). Zapojit jeho středový kontakt do analogového vstupu, levý kontakt na +3.3V (pin s označením 3V) a pravý kontakt uzemnit (pin s označením G).

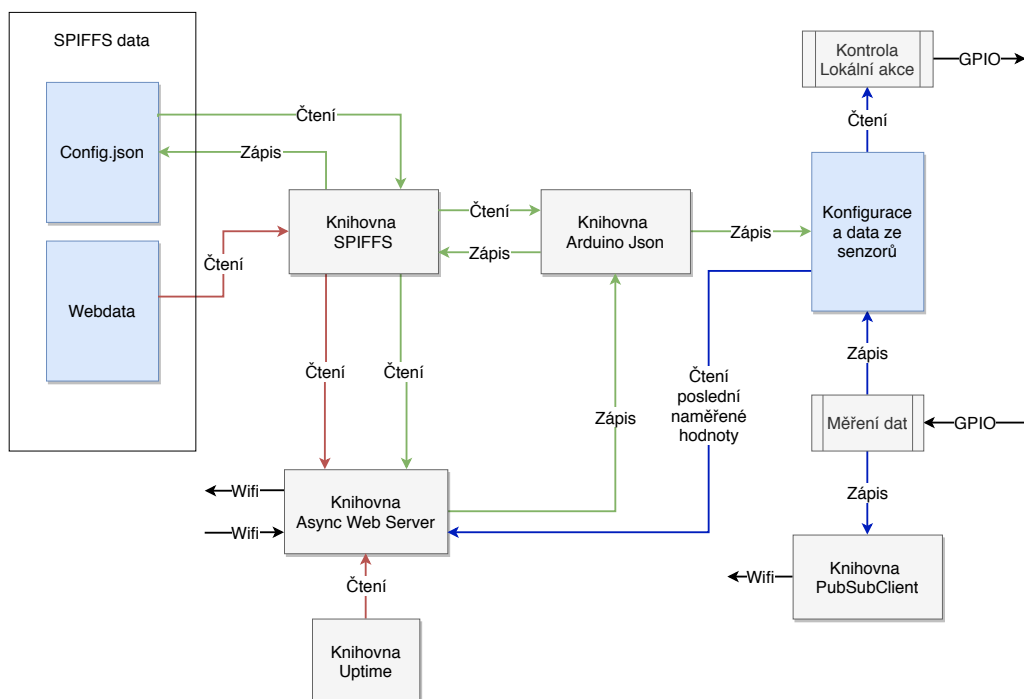
10 Chování navrženého firmwaru

Chování navrženého firmwaru je popsáno v následujících podkapitolách. Podkapitola výchozí hodnoty popisuje, jaké hodnoty lze předpokládat jako výchozí, při prvním spuštění zařízení, či závislé na verzi firmwaru.

Podkapitola inicializace zařízení popisuje úkony, které zařízení samo provede před úplným uvedením do provozu.

Podkapitola kontinuální chod popisuje úkony, které zařízení cyklicky provádí, pokud prošlo fází inicializace.

Na obrázku 10.1 je vyobrazen tok dat uvnitř zařízení. Červené šipky vyjadřují data pro webové rozhraní. Zelené šipky vyjadřují konfigurační data. Modré šipky vyjadřují naměřená data ze senzorů.



Obrázek 10.1: Diagram toku dat v zařízení

10.1 Výchozí hodnoty

Všechny klíčové informace jsou vypisovány na sériovou linku ve výchozí rychlosti 115200 baud/s. Výpis ladících informací je vypnut. Výchozí jméno zařízení a tím i jeho DNS jméno a název wifi sítě je: ESP_Universal_Sensor. Wifi síť, vysílaná zařízením je nezabezpečená. Pin je možné použít na zařízení pouze jednou, aby se zabránilo kritickému selhání v případě chyby v podobě nastavení pinu jako vstupní a výstupního zároveň. Podporované digitální piny jsou GPIO-12, GPIO-13 a GPIO-14. Důvod proč jen tyto 3, viz sekce o NodeMCU v3.0. Podporované analogové piny jsou A0 (vnitřně označen jako pin 17). I2C sběrnice je umístěna na pinech GPIO-4 (SDA pin, datový) a GPIO-5 (SCL pin, hodinový signál).

10.2 Inicializace zařízení

V případě, že nenastane žádná neočekávaná chyba, se firmware chová následujícím způsobem. Po připojení napájení proběhne inicializace, která je složena z následujících kroků:

- Kontrola SPIFFSu, pokud je dostupný a podařilo se jej připojit, může inicializace pokračovat, pokud ne, zařízení přejde do chybového stavu a je nutné jej restartovat, případně opětovně nahrát obsah SPIFFSu. Pravděpodobně totiž došlo k jeho poškození. Tento stav je signalizován blikáním integrované diody v intervalech 0,5 sekund svítí, 0,25 sekund nesvítí.
- Následně se pokusí načíst konfigurační soubor z paměti SPIFFSu. V případě, že neexistuje, vychází se z výchozích hodnot. V případě úspěchu proběhne validace tohoto načteného souboru. V případě, že validace neskončí úspěchem, dojde k odstranění konfiguračního souboru, aby se zabránilo možnému poškození zařízení zadáním nevalidních dat a pokračuje se s výchozími hodnotami.
- Díky získané konfiguraci je teprve možné začít se připojovat k Wifi síti, je-li konfigurace k dispozici. V případě volby DHCP je zažádáno o přidělení IP adresy od DHCP serveru, v případě volby Static je použita adresa, maska a výchozí brána z konfiguračního souboru. Počet pokusů o připojení je 10 a jsou po 1 vteřině. V případě úspěchu je signalizováno trvalým rozsvícením integrované diody. V případě neúspěchu dojde k vytvoření vlastní Wifi sítě, pojmenované podle názvu zařízení, a pokud

je k dispozici heslo sítě z konfiguračního souboru, bude i zaheslována tímto heslem. Je tomu tak proto, aby bylo možné znovu nastavit jméno a heslo do wifi sítě, kam je požadováno zařízení připojit.

- Po inicializaci wifi se provede inicializace webového serveru na portu 80.
- Dalším úkonem je inicializace spojení s MQTT brokerem, pokud je jeho konfigurace dostupná v konfiguračním souboru. Provede se pokus o připojení na zadanou IP adresu a port, v případě dostupnosti jména a hesla v konfiguračním souboru bude připojení i s autentizací. Počet pokusů o připojení je 5 a jsou po 1 vteřině. V případě neúspěchu je postup stejný jako kdyby MQTT vůbec nebylo nastaveno, zařízení se nebude pokoušet nikam posílat naměřená data.
- V poslední fázi se provede nahrání nastavení senzorů do vnitřních struktur zařízení a jejich inicializace, aby na nich mohlo být následně prováděno měření. V případě, že nejsou k dispozici senzory, se neděje nic.
- Inicializace je dokončena informací o dokončení a vykreslením loga vývojáře. Ukázka úspěšné inicializace je vidět na obrázku 10.2.

10.3 Kontinuální chod

Zařízení, po úspěšné inicializaci, je připraveno k provozu. V případě uživatelských požadavků z webového rozhraní je schopno je asynchroně vykonávat, téměř nezávisle na zbytku firmwaru. Důvod proč téměř nezávisle je ten, že využije časových mezer v podobě příkazů pozastavení běhu firmwaru v hlavní smyčce a místo čekání se vykonává kód pro obsluhu webu. Popřípadě se vykoná vždy po skončení vykonávání hlavní smyčky (a tím vlastně i před jejím opětovným započítím).

V základu se provádí ckylicky následující úkony:

- Kontrola na přijímání aktualizace firmwaru přes OTA. V případě požadavku na přijetí je chod zařízení pozastaven a aktualizace je přijímána.
- Změření času od zapnutí systému.
- Kontrola, zda-li lokální akce senzoru nemá být již ukončena.
- Kontrola připojení k wifi (bylo-li připojeno), v případě výpadku se pokouší asynchroně o opětovné připojení. Interval mezi pokusy je 5 sekund.
- Kontrola připojení k MQTT brokeru (bylo-li připojeno), v případě výpadku se pokusí o téměř asynchroní znovupřipojení. Důvod proč téměř asynchroní není vyložene chybou firmware, ale spíše knihovny obstarávající komunikaci s MQTT. Její funkce pro opětovné připojení je napsána jako částečně blokující. Interval mezi pokusy je 10 sekund.
- Samotné měření senzorů. Sensory jsou ve výchozím stavu měřeny každých 50 milisekund a pokud je definován interval k odesílání dat do MQTT serveru, tak i v tomto intervalu. Pokud zařízení má hlásit pouze změnu stavu, je po každém provedeném měření porovnáváno s předchozí naměřenou hodnotou, zda-li nedošlo ke změně. Pokud ano, tato informace je odeslána do MQTT brokeru. V případě intervalového měření se v intervalu mezi odeslanými zprávami ukládá minimální a maximální naměřená hodnota. Tyto 2 hodnoty, spolu s aktuálně naměřenou, jsou po těchto intervalech odesílány periodicky do MQTT brokeru.

10.4 Formát výstupu dat do MQTT brokeru

Zařízení umí odesílat do MQTT brokeru 2 podobné formáty zpráv. Důvod proč dva je ten, že při intervalovém měření jsou navíc, mimo aktuální hodnotu, odesílány i maximální a minimální hodnoty naměřené v intervalu. Zpráva je odeslána do topicu, který má senzor nakonfigurován. V principu se jedná o jednoduchý json objekt, více informací o formátu json je k dispozici v kapitole Konfigurace, v sekci Přímá editace konfiguračního souboru.

V případě změny stavu je formát následující: {"name": "Jméno senzoru", "publishEvent": "STATE_CHANGE", "curVal": "Aktuálně naměřená hodnota"}.

V případě intervalového měření je formát následující: {"name": "Jméno senzoru", "publishEvent": "INTERVAL", "minVal": "Nejmenší naměřená hodnota", "maxVal": "Největší naměřená hodnota", "curVal": "Aktuálně naměřená hodnota"}.

11 Možnosti aktualizace

Firmware popsany v této bakalářské práci nabízí možnost aktualizace, jak pro programátory, tak i pro uživatele bez vývojového prostředí. Ovšem bez vývojového prostředí není možné aktualizovat obsah vnitřního souborového systému.

Zařízení nemá žádné omezení pro příjem verze nového firmwaru, takže je možné pomocí obou níže popsanych postupů provést jak aktualizaci, tak návrat k předchozí verzi.

11.1 Vývojové prostředí Arduino IDE

Pro aktualizaci přes vývojové prostředí stačí otevřít projekt obsahující zdrojový kód pro příslušnou verzi firmwaru a spustit nahrávání kliknutím na tlačítko nahrát. V případě potřeby aktualizovat i obsah souborového systému je nutné mít v projektu složku data, jejíž obsah je to, co se nahraje do souborového systému zařízení. Aktualizace souborového systému se provede pomocí tlačítka „ESP8266 Sketch Data Upload“ v sekci Nástroje.

11.2 Webové rozhraní

V případě, že není k dispozici vývojové prostředí, je možné použít alternativu v podobě webového rozhraní. V Sekci Tools je dole, v podsekcí Firmware, nutné vybrat z lokálního uložistě soubor s kompilovaným firmwarem (soubor končící .bin) a ten nahrát do zařízení pomocí tlačítka Upload. Po nahrání se provede automaticky aktualizace a zařízení bude restartováno.

12 Aktuální možnosti senzorů

Aktuálně je možné k zařízení připojit níže popsané typy senzorů. Pro všechny senzory je k dispozici jak měření v intervalech, tak pro změnu stavu. Nastavení je plně v rukou uživatele.

Obecný digitální senzor, který hlásí pouze hodnotu HIGH, či LOW. Podporován je na všech volných digitálních pinech. Tato možnost pokryje všechny běžné digitální senzory pracující v logice 3V

Obecný analogový senzor, podporován je pouze na analogovém pinu zařízení, značeném jako A0. Jeho rozsah je pouze 0-1024. Jedná se o maximální napětí cca. 3V. Tato možnost pokryje všechny běžné analogové senzory, pracující v logice 3V.

Obecný I2C senzor, podporován je na pinech sběrnice I2C. Počet senzorů je limitován pouze maximálním počtem zařízení na sběrnici I2C. Na jednom senzoru je možné najednou měřit pouze 1 hodnotu o maximální velikosti 4 bajty. Je dovoleno nastavit, který bajt je významnější tzv. endianita. Podporováno je také nastavení pozice registru pro výběr dat a kolik bytů se má vybrat. A v případě potřeby je zde možnost inicializace senzoru zápisem hodnot do zadaných registrů. Tato možnost by teoreticky měla stačit pro všechny senzory pracující na sběrnici I2C.

12.1 Lokální akce

Při splnění zadané podmínky je možné vykonat určitou akci. Detailně popsáno v sekci Sensors, u Webového rozhraní na stránce 54.

12.2 Ukázky nastavení senzorů

Na obrázku 12.1 je ukázka nastavení obecného digitálního senzoru. Na obrázku 12.2 je ukázka nastavení obecného analogového senzoru. Na obrázku 12.3 je ukázka nastavení obecného I2C senzoru.

Main Wifi MQTT Sensors Tools

Sensor: Button

Name * Button

Type DIGITAL ▼

Pin * GPIO-12 ▼

Publish action STATE_CHANGE ▼

Publish * doma/pokoj/dummyButton

Local Action

Condition

Value is = Than 1

Obrázek 12.1: Ukázka nastavení obecného digitálního senzoru

Main Wifi MQTT **Sensors** Tools

Sensor: Potentiometer

Name * Potentiometer

Type ANALOG ▾

Pin * A0 ▾

Publish action INTERVAL ▾

Publish * doma/pokoj/dummyAnalog

Local Action

Condition

Value is Than

> 128

Obrázek 12.2: Ukázka nastavení obecného analogového senzoru

Main Wifi MQTT Sensors Tools

Sensor: Gyro

Name * Gyro

Type I2C

Init. reg. & data 3e 80;15 00;16 18

Data pos. & size 21 2

Little endian

Big endian

Address * 104

Publish action INTERVAL

Publish * doma/pokoj/gyro

Obrázek 12.3: Ukázka nastavení obecného I2C senzoru

13 Konfigurace

Díky dobře rozmyšlenému návržení firmwaru jsou všechny potřebné informace pro chod nastaveného zařízení uloženy v konfiguračním souboru v paměti SPIFFSu. Tento soubor přímo ovlivňuje výsledné chování obecného senzoru.

Konfiguraci je možné provádět přes velmi přívětivé webové uživatelské rozhraní, nebo přímo v konfiguračním souboru, který lze ve webovém rozhraní stáhnout a po editaci opět nahrát. Ovšem v případě přímé editace konfiguračního souboru je nutné zařízení restartovat, aby se aplikovali změny. Tento úkon je možné provést pomocí webového rozhraní.

V případě editace senzorů ve webovém rozhraní se změny aplikují okamžitě. V případě úpravy chování v sekci Wifi, nebo MQTT je nutné zařízení manuálně restartovat, aby se aplikovali změny.

13.1 Konfigurace přes webové rozhraní

Do webového rozhraní se dá dostat pomocí libovolného webového prohlížeče. Podmínkou je být ve stejné lokální síti, jako je zařízení. Stačí zadat do vyhledávací IP adresu zařízení, nebo jeho název ve formátu „http://{název}/“ a dojde k přesměrování na úvodní stránku.

V případě módu AP je IP adresa „192.168.4.1“, ve staničním módu s DHCP je možné stále použít přístup přes jméno a na hlavní stránce zjistit přiřazenou IP adresu zařízení.

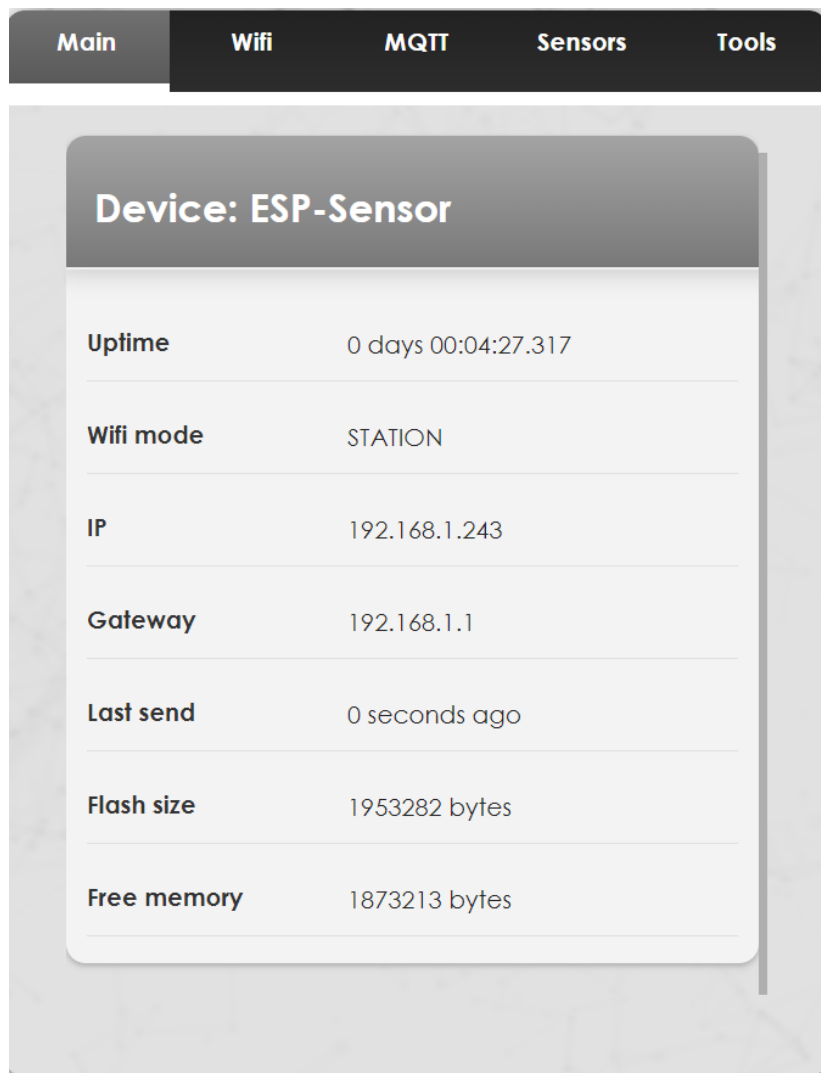
Při konfiguraci přes webové rozhraní si zobrazená webová stránka hlídá, zda-li jsou zadávaná data validní, takže nedovolí uložit nastavení, které je chybné. Například název wifi pro připojení je delší než 32 znaků, což je limit pro všechny názvy wifi zařízení.

Samotné webové rozhraní je přehledně členěno do několika samostatných webových stránek. Samotné stránky a jejich funkce jsou popsány níže.

13.1.1 Úvodní stránka

Úvodní stránka (viz obrázek 13.1) nedovoluje přímo žádné nastavení, slouží spíše jako informační. Zobrazuje jméno zařízení, na obrázku níže je to ESP-Sensor, a pak po řádcích oddělené informace, kterými jsou:

- Uptime, což je doba, která uběhla od zapojení zařízení do napájení. Je zobrazena ve formátu, počet dnů, počet hodin, počet minut, počet sekund a za tečkou počet milisekund. Na obrázku 13.1 se jedná o 4 minuty, 27 sekund a 317 milisekund.
- Wifi mode udává, jestli je zařízení připojeno k wifi, mód STATION, nebo pokud samo vysílá wifi jako přístupový bod, mód AP.
- IP vypisuje aktuální IP adresu zařízení.
- Gateway vypisuje adresu výchozí brány, pokud je připojen k wifi, pokud ne, tak je v módu přístupového bodu a nepotřebuje znát výchozí bránu.
- Last send udává, kolik sekund uběhlo od posledního úspěšného odeslání dat na MQTT server, tento interval by v případě nastaveného sensoru, který má odesílat po intervalu neměl být větší než je nastavený interval, pokud je větší, může to signalizovat chybu v komunikaci, nebo v samotném měření (trvá nepřírozeně dlouho).
- Flash size udává celkovou velikost souborového systému ve flash paměti v bytech.
- Free memory udává počet volných bytů v paměti souborového systému.



Obrázek 13.1: Vzhled úvodní stránky

13.1.2 Wifi

Stránka wifi (viz obrázek 13.2) slouží hlavně k nastavení jména a hesla k wifi, ke které je požadováno zařízení připojit. Hvězdička označuje povinný parametr, který musí být vyplněn, jinak není možné konfiguraci uložit. Po řádcích jsou zde oddělené informace, kterými jsou:

- Name (*), udává jméno zařízení, jeho DNS název a zároveň jméno wifi sítě, kterou umí vysílat. Maximální délka názvu zařízení je 32 znaků.
- SSID, jedná se o jméno wifi sítě, ke které je požadováno zařízení připojit. Maximální délka 32 znaků.
- Password, jedná se o heslo do wifi sítě, ke které je požadováno zařízení připojit. Maximální délka 64 znaků.
- IP mode, udává zda-li chceme nastavit vlastní IP adresu zařízení, static, nebo necháme přiřadit IP adresu DHCP server.
- AP password, jedná se o heslo do wifi sítě, kterou umí zařízení vysílat. Maximální délka 64 znaků, v případě nevyplnění hesla je síť nezabezpečená!

Konfiguraci uložíme stisknutím tlačítka Save. Pokud je v pořádku, je odeslána do zařízení. Pokud je chybná, dojde k upozornění uživatele a dokud nebude opravena, není odeslána.

Pro uplatnění nové konfigurace je nutné zařízení restartovat, buď fyzicky, nebo pomocí tlačítka Reboot v sekci Tools.

Main **Wifi** **MQTT** **Sensors** **Tools**

Device: ESP-Sensor

Name *	<input type="text" value="ESP-Sensor"/>
SSID	<input type="text" value="Atlantis"/>
Password	<input type="password" value="••••••••"/>
IP mode	<input type="text" value="DHCP"/>
AP password	<input type="password"/>

SAVE

Obrázek 13.2: Vzhled Wifi stránky

13.1.3 MQTT

Stránka MQTT (viz obrázek 13.3) slouží k nastavení adresy MQTT brokeru. Hvězdička označuje povinný parametr, který musí být vyplněn jinak není možné konfiguraci uložit. V případě, že nechceme používat MQTT, není třeba vyplnit pro funkčnost senzoru lokálně. Pokud již byla adresa vyplněna, po zadání adresy 0.0.0.0 dojde k vyřazení funkce MQTT v zařízení. Po řádcích jsou zde oddělené informace, kterými jsou:

- IP Address (*), udává IP adresu zařízení kde se nachází MQTT broker. Alternativně lze použít DNS název (zkratka DNS znamená Domain Name System, neboli systém doménových jmen), který má maximální povolenou délku 64 znaků.
- Port (*), udává port, na kterém MQTT broker naslouchá. Musí být v rozsahu 1-65535.
- User, udává uživatele, pod kterým se chceme přihlásit do MQTT brokeru. Maximální délka uživatelského jména je 64 znaků.
- Password, udává uživatelské heslo, pod kterým se chceme přihlásit do MQTT brokeru. Maximální délka uživatelského jména je 64 znaků.
- Interval, udává počet vteřin, po kterých se mají periodicky hlásit výsledky naměřených hodnot v tomto intervalu do MQTT brokeru. Toto nastavení je společné pro všechny senzory.

Konfiguraci uložíme stisknutím tlačítka Save. Pokud je v pořádku, je odeslána do zařízení. Pokud je chybná, dojde k upozornění uživatele a dokud nebude opravena, není odeslána.

Pro uplatnění nové konfigurace je nutné zařízení restartovat, buď fyzicky, nebo pomocí tlačítka Reboot v sekci Tools.

The image shows a web interface for configuring MQTT settings on an ESP-Sensor device. The interface has a dark navigation bar at the top with five tabs: 'Main', 'Wifi', 'MQTT', 'Sensors', and 'Tools'. The 'MQTT' tab is currently selected. Below the navigation bar, there is a header section with the text 'Device: ESP-Sensor'. The main content area contains a form with the following fields:

- IP Address ***: A text input field containing the value '192.168.1.25'.
- Port ***: A text input field containing the value '1883'.
- User**: An empty text input field.
- Password**: An empty text input field.
- Interval (s) ***: A text input field containing the value '3'.

At the bottom of the form, there is a large, light-colored button labeled 'SAVE'.

Obrázek 13.3: Vzhled MQTT stránky

13.1.4 Sensors

Stránka sensors je prakticky nejdůležitější ze všech, zde se přidávají senzory, které budou připojeny k zařízení. Stačí kliknout na tlačítko "ADD NEW DEVICE".

Vzhled stránky bez přidáných senzorů je možné vidět na obrázku 13.4.



Obrázek 13.4: Stránka Sensors, bez senzorů

Po kliknutí se otevře stránka s možnou konfigurací senzorů. Zde je možné nastavit si chování senzoru, jak má odesílat data a jestli je vyžadována lokální akce.

Položky pro základní nastavení senzoru jsou vidět na obrázku 13.5 a jsou následující:

- Name (*), udává jméno senzoru. Maximální délka 32 znaků.
- Type, udává typ senzoru. Typy podporovaných senzorů jsou v dostupné v kapitole Aktuální možnosti senzorů.
- Pin (*), udává pin, kam bude senzor připojen. Podle zvoleného typu senzorů se může lišit seznam povolených pinů.
- Publish action, udává, jak má senzor hlásit svá měření. INTERVAL, kdy posíláme data jen jednou za interval, nebo STATE_CHANGE, kdy posíláme data v případě změny stavu.
- Publish (*), jedná se o řetězec znaků, který udává téma (topic), do kterého má senzor hlásit svá měření v MQTT brokeru. Maximální délka 64 znaků.
- Local action, udává, zda-li se má vykonat určitá akce, pokud naměřená hodnota splní zadanou podmínku. Více na stránce 54.

Konfiguraci uložíme stisknutím tlačítka Save. Pokud je v pořádku, je odeslána do zařízení. Pokud je chybná, dojde k upozornění uživatele a dokud nebude opravena, není odeslána.

Main Wifi MQTT Sensors Tools

Sensor: Sensor0

Name * PIRCidlo

Type DIGITAL ▼

Pin * GPIO-12 ▼

Publish action STATE_CHANGE ▼

Publish * doma/obyvak/pohyb

Local Action

SAVE CANCEL

Obrázek 13.5: Stránka pro konfiguraci jednoho senzoru

Pokud chceme, aby se například při naměření hodnoty HIGH na digitálním senzoru vykonala určitá lokální akce, lze toho docílit pomocí zaškrtnutí funkce local action.

Lokální akce může být velmi užitečná, protože zařízení dokáže samo reagovat na naměřená data a to i bez nutnosti spojení s MQTT brokerem. To může být využitelné například v případě, že máme pohybové čidlo, které by mělo v případě pohybu po určitý interval rozsvítit světlo a zároveň nahlásit do brokeru, že zaznamenalo pohyb. Případně při naměření kritické hodnoty zapnout sirénu a upozornit uživatele, že podle naměřené hodnoty nastal problém.

Je zde třeba nastavit následující parametry:

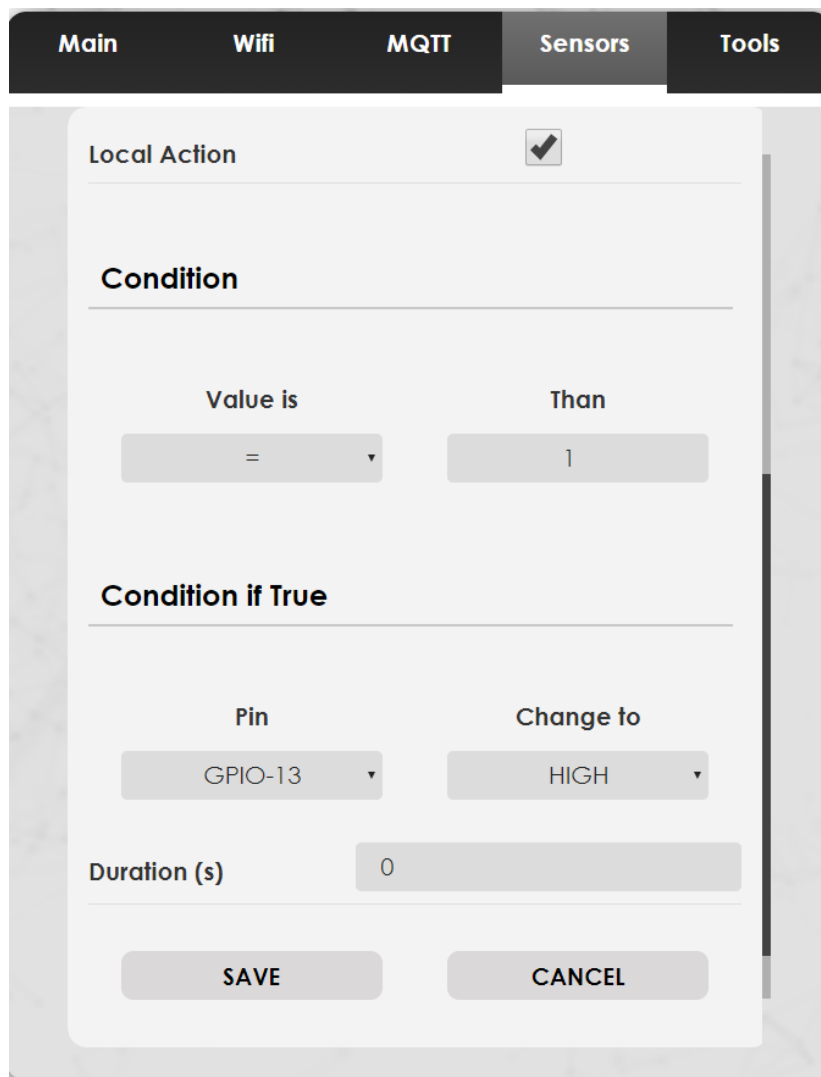
- Podmínku (condition), kde je možné zvolit: Větší, menší, nebo rovno. A než (than) jaká hodnota. V případě digitálních senzorů pouze hodnota 0 (LOW) nebo 1 (HIGH).
- Co se má při splnění podmínky vykonat. Na obrázku 13.6 je nastaveno, že se na pinu GPIO-13 má změnit hodnota z LOW na HIGH.
- Duration udává jak dlouho má být po skončení podmínky vykonávána akce. V případě kladné hodnoty se jedná o počet vteřin od posledního splnění podmínky. V případě 0, bude akce vykonávána pouze, dokud bude podmínka platit. V případě -1 se akce neukončí, dokud nedojde k restartu zařízení, nebo odebrání (či editaci) senzoru.

Konfiguraci uložíme stisknutím tlačítka Save. Pokud je v pořádku, je odeslána do zařízení. Pokud je chybná, dojde k upozornění uživatele a dokud nebude opravena, není odeslána.

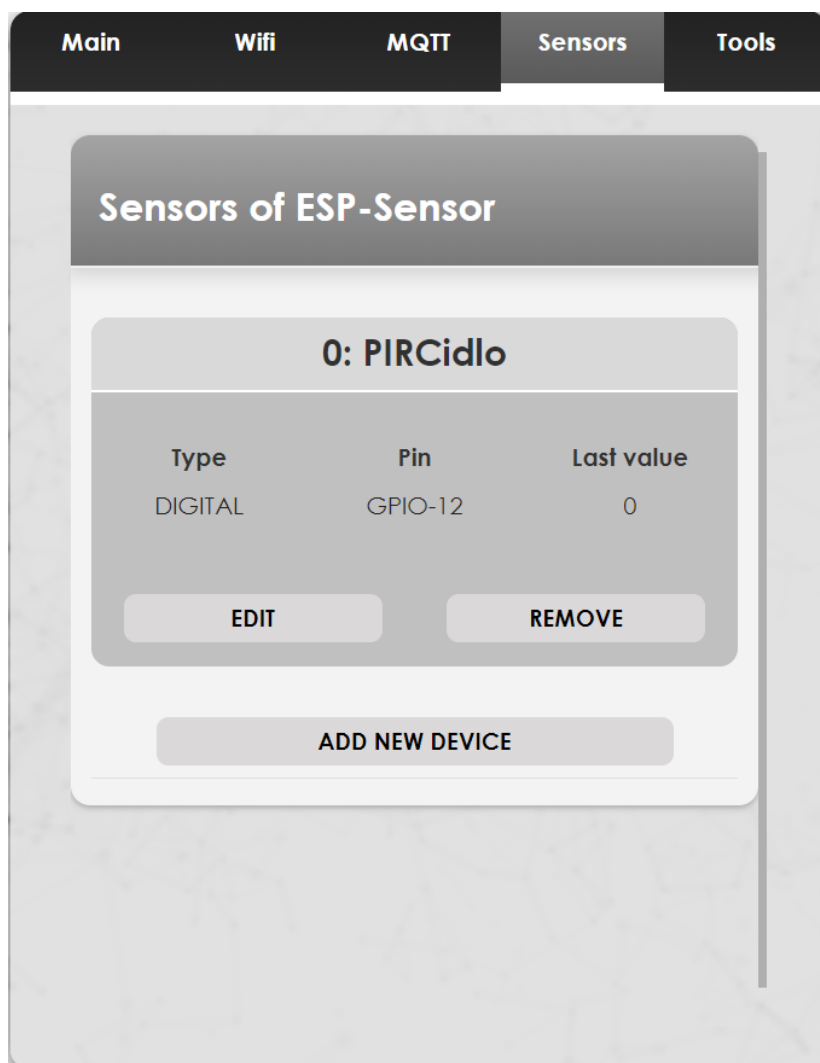
V případě již nastavených senzorů je v sekci Sensors vidět box se senzorem (viz obrázek 13.7) a rychlý přehled jeho základního nastavení. Tím je jeho typ, kam je připojen a poslední naměřená hodnota.

Dále je zde možnost editovat senzor, případně senzor odstranit.

Vzhledem k tomu, že není počet senzorů přímo omezen, je zde stále možnost přidat další senzor.



Obrázek 13.6: Stránka pro konfiguraci lokální akce jednoho senzoru



Obrázek 13.7: Stránka Sensors, s jedním senzorem

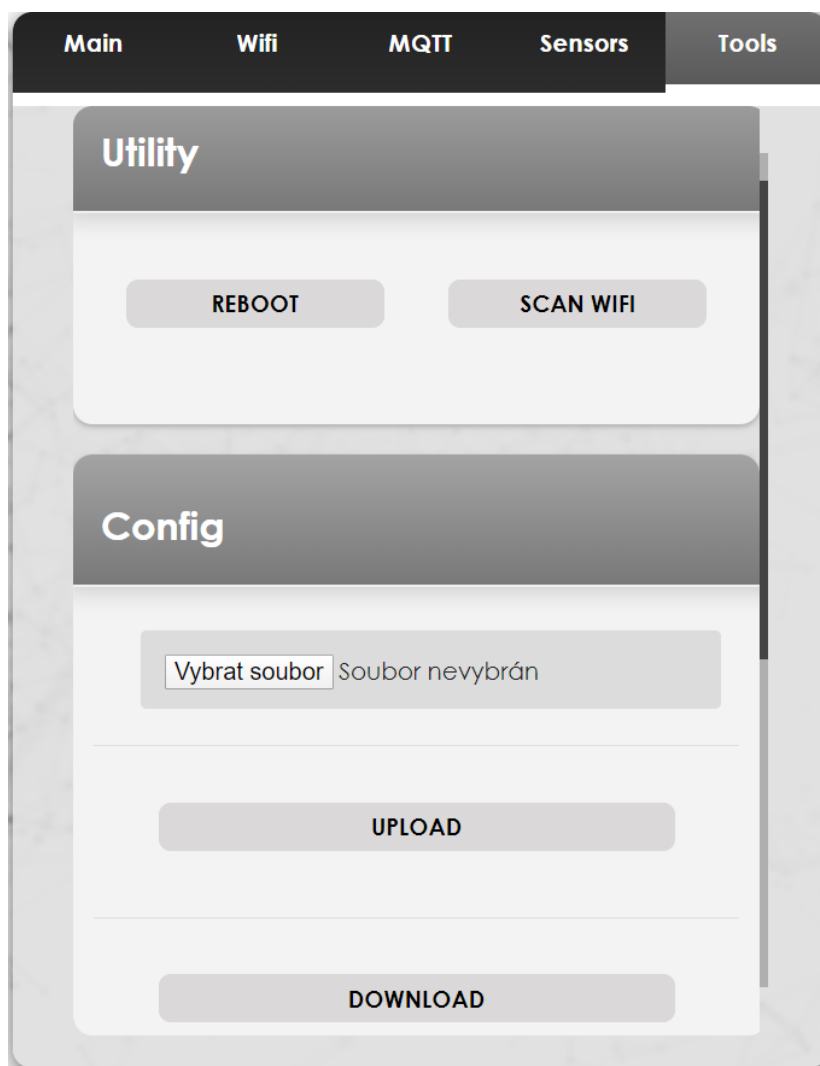
13.1.5 Tools

Na této stránce jsou umístěné dodatečné prvky (viz obrázek 13.8). Význam jednotlivých prvků je popsán níže.

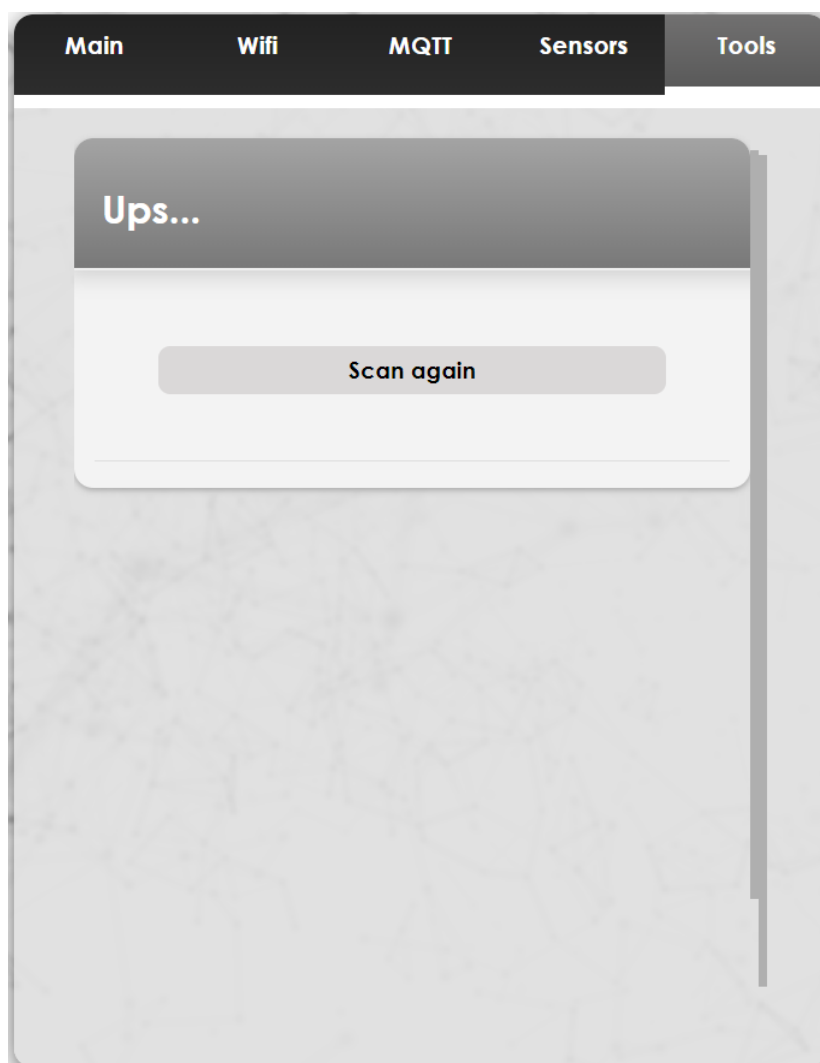
V sekci Utility je tlačítko Reboot, které slouží k restartu zařízení. Například pro vynucení změny konfigurace. Další prvek v této sekci je tlačítko Scan Wifi, které slouží pro zobrazení všech Wifi sítí v dosahu zařízení. ESP8266 občas neprovede skenování správně, nebo v dosahu nejsou žádné Wifi sítě a zobrazí se stránka s chybou, kde je možné spustit skenování znovu (viz obrázek 13.9).

Na stránce s výsledky skenování (viz obrázek 13.10) je zobrazen název Wifi sítě, kanál na kterém vysílá a barevně zvýrazněna síla signálu, zelená je nejsilnější, žlutá středně silný a červená je velmi slabý signál.

V sekci Config je možnost nahrát nový konfigurační soubor a po kliknutí na tlačítko Upload dojde k jeho nahrání do zařízení a k validaci. V případě, že je úspěšná, dojde k restartu zařízení a aplikování konfigurace. V opačném případě je nová konfigurace zamítnuta a uživatel je o tomto úkonu informován pomocí notifikací na stránce v levém dolním rohu (viz obrázek 13.11).

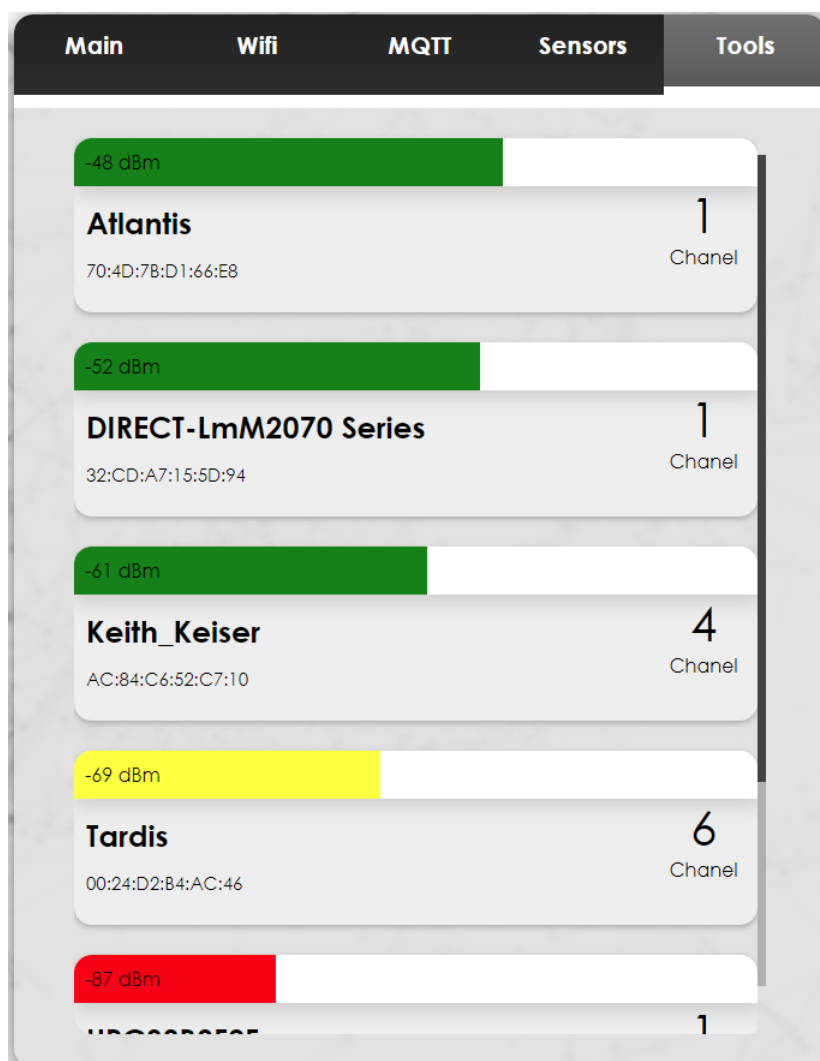


Obrázek 13.8: Vrchní část stránky Tools

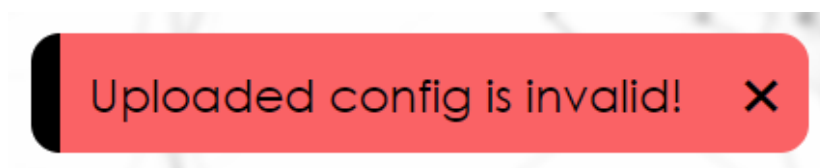


Obrázek 13.9: Zobrazení výsledku skenování Wifi sítí v případě selhání, či nedostupnosti

,



Obrázek 13.10: Zobrazení výsledku úspěšného skenování Wifi sítí



Obrázek 13.11: Notifikace o zamítnutí nahraného konfiguračního souboru

13.2 Přímá editace konfiguračního souboru

Před přímou editací konfiguračního souboru je nutné mít na paměti, že pokud v něm bude faktická chyba (například chci dát sensor na pin GPI-12 a jeho local action také na pin GPIO-12), tak se konfigurace nahraje a bude celá zamítnuta jako nevalidní.

Konfigurační soubor je ve formátu json (JavaScript Object Notation), jedná se o poměrně snadný formát díky stylu "Klíč a hodnota", kdy je určitému prvku, identifikovanému podle klíče, přiřazena určitá hodnota, ať už číslo, řetězec znaků, či další objekt.

Pro vysvětlení budeme uvažovat téměř maximálně využitý konfigurační soubor a v jednotlivých podkapitolách níže budou vysvětleny i podmínky, kdy je třeba co vyplnit. Samotný kořenový objekt může obsahovat klíč s názvem "Name", který udává jméno zařízení. Jeho maximální délka je 32 znaků.

13.2.1 Objekt WifiSTA

V tomto objektu je uložena staniční konfigurace zařízení, jako je jméno wifi pro připojení. Pokud existuje, musí v něm být klíč "SSID", který udává jméno wifi sítě pro připojení, maximální délka 32 znaků. Dále v něm musí v něm být klíč "PASS", který udává heslo do wifi sítě pro připojení, maximální délka 64 znaků. A ze základních, jako poslední klíč "mode", který udává jak bude získána IP adresa zařízení, povolené hodnoty jsou DHCP (získání od DHCP serveru), nebo Static.

Pokud je zvoleno Static, je potřeba dovyplnit ještě klíče "IP", "mask" a "DGW", ip udává IP adresu zařízení v síti, mask masku a dgw výchozí bránu. Všechny 3 klíče jsou pro IP adresy, a proto je jejich maximální délka 15 znaků.

13.2.2 Objekt WifiAP

V tomto objektu je uložena pouze konfigurace hesla do wifi sítě, vysílané zařízením. Pokud existuje, musí v ní být klíč "PASS", který má maximální povolenou délku 64 znaků.

13.2.3 Objekt MQTT

V tomto objektu je uložena konfigurace pro komunikaci s MQTT brokerem. Pokud existuje, musí v něm být klíč "IP", který udává adresu brokeru pro připojení, jak IP adresu, tak doménovou adresu. V případě hodnoty "0.0.0.0" dojde k vyřazení MQTT z provozu, maximální délka 64 znaků. Dále v něm musí být klíč "port", který udává síťový port, na kterém MQTT broker naslouchá, musí se jednat o číslo v rozahu 1-65535. A z povinných, jako poslední klíč "interval", jedná se o číselnou hodnotu ve vteřinách udávající, jak často má zařízení odesílat data, v případě, že senzor je nastaven na intervalové měření. Povolená jsou celá kladná čísla, větší než 0.

Pokud je požadováno dodatečné zabezpečení MQTT brokeru, je možné pomocí klíčů "user" a "pass" uložit uživatelské jméno a heslo pro přihlášení do brokeru. U obou je maximální délka 64 znaků.

13.2.4 Pole Sensors

V tomto poli je uložena konfigurace jednotlivých senzorů. Každý prvek pole představuje jeden senzor. Níže je popsáno, co musí senzor obsahovat.

Klíč "ID", který udává unikátní identifikátor senzoru. Musí to být celé číslo větší, nebo rovno nule. Klíč "Name", který udává jméno senzoru, maximální délka 32 znaků. Klíč "publishAction", který udává kdy se mají odesílat naměřená data. Povolené hodnoty jsou STATE_CHANGE, pro odesílání při změně stavu, nebo INTERVAL, pro odesílání v intervalu, který je definován v objektu MQTT. Klíč "publish", který udává v jakém tématu má být naměřená hodnota odesílána, maximální délka je 64 znaků. Klíč "type", který udává typ senzoru, povolené jsou DIGITAL, ANALOG, nebo I2C. Podle tohoto klíče se liší chování pro klíč "inPin", v případě digitálního se musí jednat o validní digitální pin (například: GPIO-12) a nesmí být již před tím někde využit. Pro analogové senzory platí to samé, pouze místo digitálních pinů se jedná o analogové. V případě I2C je to trochu složitější, v tu chvíli udává adresu zařízení na sběrnici a to musí být číselná hodnota v rozsahu 1-127. V případě volby I2C je ještě potřeba dodefinovat povinné klíče "initData", "dataPosition" a "useLe", initData říká, jak je třeba zařízení inicializovat před započítím měření, jedná se o sekvenci hexadecimálních čísel oddělených mezerou, ve formátu: 1. číslo = číslo registru, 2.-n. číslo = hodnota k zapsání. V případě potřeby zapsat do více registrů, které na sebe nenavazují, je možné za poslední hodnotu zapsat znak středník a opět provést identický zápis hexadecimálních čísel oddělených mezerou. Tento postup lze opakovat podle potřeby. V klíči dataPosition se nachází definice jak získat ze senzoru data, první číslo udává hexadecimálně číslo registru pro výběr dat a druhé

číslo, oddělené mezerou, udává počet bajtů ke čtení (povolené je čísla 1-4 bajtů). A poslední povinný klíč pro I2C, "useLe", který udává jestli se má použít čtení hodnoty stylem Little Endian, nastaveno na true, nebo Big Endian, nastaveno na false. Poslední povinný klíč "localAction", který udává zda li je vyžadována lokální akce, povolené hodnoty jsou textově "true", nebo "false". V případě, že je požadována lokální akce, je nutné dodefinovat následující klíče. Klíč "compare", který udává jak se bude naměřená hodnota porovnávat, povolené hodnoty jsou GREATER, větší, LOWER, menší, a EQUAL, pro rovno. Klíč "value" udává číselnou hodnotu, se kterou chceme naměřenou hodnotu porovnat. Klíč "outPin", který udává číslo digitálního pinu, kde se vykoná akce, při splnění podmínky. Opět se musí jednat o ještě nepoužitý pin. Klíč "newState", který udává do jakého stavu má pin přejít, povolené hodnoty jsou HIGH (log 1) a LOW (log 0). Posledním je klíč "duration", který udává jak dlouho má pin ve stavu setrvat od skončení posledního splnění podmínky. Hodnota -1 říká, setrvej stále, hodnota 0 říká, pouze po dobu kdy je podmínka splněna, poslední možností je kladné celé číslo, které bude udávat dobu trvání ve vteřinách.

14 Jednotlivé soubory projektu

Projekt je vlastně složen ze 2 téměř separátních projektů. Hlavním projektem je samotný firmware, obsahuje kód, který řídí zařízení. A pak projekt webového rozhraní, který je od hlavního projektu oddělen.

14.1 Samotný Firmware

Pro lepší přehlednost je zdrojový kód rozdělen do několika souborů. Soubory jsou vhodně pojmenovány, aby již názvem říkaly, co se v nich nachází. těmito soubory jsou:

- Soubor `UniversalWifiSensor.ino`, obsahuje všechny důležité importy knihoven, či jiných `.h` souborů a hlavně funkci `setup()`, pro inicializaci celého zařízení a funkci `loop()`, která se cyklicky vykonává.
- Soubor `MQTT.ino`, práce s `mqtt` a komunikace s brokerem.
- Soubor `OTA.ino`, handlers pro OTA updaty.
- Soubor `SPIFFS.ino`, práce se souborovým systémem.
- Soubor `Sensors.ino`, práce se senzory, měření, inicializace, přidání a odebrání.
- Soubor `Webserver.ino`, nastavení chování webového serveru.
- Soubor `WiFi.ino`, práce s wifi.
- Soubor `Utils.ino`, obsahuje pomocné funkce pro snazší práci a hlavně funkci pro validaci konfiguračního souboru.
- Soubor `structs.h`, obsahuje struktury použité v projekty.
- Soubor `consts.h`, obsahuje všechny důležité konstanty pro projekt, jsou pečlivě okomentovány, takže jejich význam by měl být jednoznačný.

14.2 Webové rozhraní

Web je složen z několika samostatných stránek (souborů s kódem stránek). Většinou pouze volají skript pro vykreslení, nacházející se v souboru `script.js`. Těmito soubory jsou:

- Soubor `wifi.html`, obsahuje volání pro vykreslení Wifi stránky.
- Soubor `mqtt.html`, obsahuje volání pro vykreslení MQTT stránky.
- Soubor `sensors.html`, obsahuje volání pro vykreslení Sensors stránky.
- Soubor `tools.html`, obsahuje volání pro vykreslení Tools stránky.
- Soubor `UploadDone.html`, obsahuje volání pro uložení výsledku nahrání a validace konfiguračního souboru jako úspěšné.
- Soubor `UploadInvalid.html`, obsahuje volání pro uložení výsledku nahrání a validace konfiguračního souboru jako neúspěšné.
- Soubor `style.css`, jedná se o kaskádové styly (upravují vzhled) webového rozhraní.
- Soubor `script.js`, obsahuje všechnu logiku webového rozhraní, stará se o vykreslení prakticky všech stránek a je v něm i kontrola uživatelských vstupů.

15 Jak přidat podporu pro další senzory

Firmware byl vyvíjen s důrazem na možnou budoucí rozšiřitelnost, a proto je z programátorského hlediska vhodně členěn a připraven přijímat i jiné typy senzorů s nutností minimálních zásahů. To platí hlavně pro webové rozhraní, pokud je možné použít pouze vstupní pin pro definici, není třeba do něj nijak zasahovat. Pro ulehčení hledání sekcí v kódu je připraveno 6 bloků, kterů je nutné přidat, a jsou v nich komentáře ve formátu `”//NEW SENSOR EDIT POINT”`, aby se daly snadno najít v tomto rozsáhlém projektu. Jediná podmínka, kterou by měl nový senzor splnit, je rychlé (jednotky ms) měření, jinak může dojít k nepříjemným zpomalením interakce a ovlivnit i ostatní senzory. Zpomalení by fatálně ovlivnilo všechny senzory pracující na principu změny stavu. Již by se neměřilo kontinuálně, ale došlo by k výpadku, který by trval tak dlouho, dokud by nový senzor neukončil měření. Níže je ve stručnosti popsáno, které části kódu je třeba upravit, či doplnit pro přidání dalšího senzoru.

15.1 Seznam senzorů

Nachází se v souboru `consts.h`, je potřeba přidat textové označení nového senzoru do pole `SENSOR_TYPES` a podle počtu senzorů v poli upravit konstantu `SENSOR_TYPES_SIZE`.

15.2 Inicializace

Nachází se v souboru `Sensors.ino` ve funkci `setupSensorPins()`. Zde je třeba do kódu doplnit, jaké piny senzor používá a zabrat si je, či jak se připojí na sběrnici. Případně podle potřeby provést inicializaci spojení. Inspirace může být v sekcích pro obecné senzory výše.

15.3 Odebrání senzoru

Nachází se v souboru `Sensors.ino` ve funkci `unsetupSensorPins()`, zde je třeba do kódu doplnit jak probíhá úklid při odpojení senzoru, jestli je třeba ukončit komunikaci se sběrnici, či pouze uvolnit zabrané piny. Inspirace může být v sekcích pro obecné senzory výše.

15.4 Měření hodnoty

Nachází se v souboru `Sensors.ino` ve funkci `measureSensors()`, zde je třeba do kódu doplnit jak probíhá změření dat senzorem a naměřená data uložit do proměnné `newValue`. Inspirace může být v sekcích pro obecné senzory výše.

15.5 Hlášení volných senzorů do webového rozhraní

Nachází se v souboru `WebServer.ino` ve funkci `setupWebServer()` v handleru pro `server.on("/FreePins", HTTP_GET, [] (AsyncWebServerRequest * request)`, zde je třeba do kódu doplnit jaké hodnoty se mají nabídnout uživateli ke zvolení ve webovém rozhraní, jméno senzoru je automaticky vloženo stačí přidat podmínku pro přidávaný senzor a co vložit do pole hodnot. Inspirace může být v sekcích pro obecné senzory výše.

15.6 Validátor

Nachází se v souboru `Utils.ino` ve funkci `validateJsonConfig()`, zde je třeba do kódu doplnit jaké hodnoty se mohou objevit v klíči pro vstupní pin nově přidaného senzore a případně za jakých jiných podmínek je konfigurace tohoto senzoru validní. Inspirace může být v sekcích pro obecné senzory výše.

16 Bezpečnost

V poslední době velmi oblíbené téma při vývoji jak softwarů, tak firmwarů. Firmware může být sebelepší, ale když se k němu dostane nepovolaná osoba, a firmware není zabezpečen, vzniká obrovské bezpečnostní riziko ztráty soukromých dat. Vzhledem k tomu, že se sice jedná o firmware pro obecný wifi senzor, asi je otázka bezpečnosti mírně nadnesená, ale při jeho použití jako náhrady profesionálního alarmu formou pohybových senzorů pro zabezpečení objektu, už ne.

Firmware popsany v této bakalářské práci neobsahuje příliš mnoho zabezpečovacích prvků. Proto by jeho provoz měl být striktně na vnitřní, bezpečné, síti. V případě připojení senzoru do internetu může snadno dojít k jeho napadení případným hackerem. ESP8266 sám o sobě není příliš vykonný, aby dokázal plnit na 100% všechny bezpečnostní šifry a protokoly, které dnes existují. V základu podporuje připojení k zabezpečené síti a připojení k zabezpečenému MQTT brokeru, pomocí ověření uživatelského jména a hesla. Ale tyto informace jsou stejně po síti odesílány jako běžný text. MQTT v základu také není navrženo příliš na to, aby bylo někde v nebezpečné síti. Heslo pro síť k připojení je také nešifrovaně uloženo ve vnitřní paměti zařízení. To také není příliš bezpečné, ale v případě ponechání senzoru ve vnitřní, bezpečné, síti se nedá k senzoru zvenku dostat, takže ani k souboru s konfigurací. V případě, že je vyžadováno zabezpečené spojení, by šlo použít broker v podobě například Raspberry Pi, a ten teprve nechat zabezpečeně odesílat naměřená data, či přes něj přemostit spojení například VPN tunelem (zkratka VPN znamená Virtual Private Network, neboli virtuální privátní síť). Takže by vlastně šlo o takový zabezpečovací mezistupeň. Sensor by se hlásil Raspberry Pi (byl by to jeho broker) a Raspberry Pi by teprve odesílalo data na hlavní broker, umístěný někde v internetu. Oba postupy jsou ale již poměrně pokročilý způsob použití tohoto obecného wifi senzoru a spíše než se senzorem samotným, souvisí s bezpečností odesílaných dat obecně. Webové rozhraní také neimplementuje žádný způsob autentifikace. Vzhledem k myšlence zařízení se to nejevilo jako nutnost, ale v otázce bezpečnosti je to vcelku problém. Opět ale řešitelný ponecháním zařízení ve vnitřní, bezpečné, síti.

17 Možné způsoby budoucího rozšíření

Firmware popsaný v této bakalářské práci, tak jak je plní zadané podmínky a dokonce je v něm několik věcí pro zpříjemnění práce (přístup ke konfiguračnímu souboru, webové rohraní, OTA updaty, kvalitní a dále rozšiřitelný kód, atd.), ale vývoj firmwaru je nikdy nekončící práce a vždy se najde nějaká funkce, kterou by bylo vhodné postupem času přidat. Níže je popsáno několik myšlenek, jak by bylo možné práci v budoucnu rozšiřovat, vzhledem k tomu, že se jedná o celkem kvalitní základ, na kterém se dá snadno stavět.

17.1 Přesnější čas

Myšlenka přesného času by byla vhodná například pro přidání časové značky do odesílaných dat, či pro lepší měření času od spuštění, intervalu mezi měřeními, atp. Bylo by užitečné přidat externí obvod, který by měřil čas a například při startu, a přístupu k internetu, provést jeho synchronizaci s NTP (Network Time Protocol) servery, které poskytují velmi přesný čas.

17.2 Logický a analogový filtr

V případě zarušených dat by bylo vhodné implementovat filtr, který by se pokusil eliminovat nevalidně naměřená data, případně způsobil co nejmenší odchylku od měření a tím by se docílilo mnohem přesnějším výsledků, hlavně v případě analogových senzorů, kde se hodnoty i beze změny podmínek často mění.

17.3 Záznam dat

V případě ztráty spojení s MQTT brokerem by bylo vhodné nějak ukládat naměřená data a při opětovném navázání spojení uložená data najednou odeslat a pokračovat v normálním chodu. V principu by šlo o nějaký buffer, který by ukládal například maximálně pevně danou historii 10 minut měření. Nebo by šlo využít místo ve SPIFFSu a pokusit se data ukládat tam, například do souboru jehož název by byl id senzoru.

17.4 Záložní baterie

V případě, že by vypadlo napájení, je vysoká šance, že i MQTT broker bude nedostupný. Pokud se jedná o kritické zařízení, které svou lokální akcí dokáže odvrátit katastrofu (například přetečení zásobníku na vodu), bylo by vhodné vymyslet způsob, jak jej udržet v chodu i po výpadku napájení. V nejjednodušším případě, vzhledem k připojení přes micro USB, by se dala zvážit přímo power banka, běžně určená pro prodloužení doby chodu mobilních telefonů. Případně vlastní nabíjecí baterie, s příslušným řídicím obvodem.

18 Závěr

Body zadání bakalářské práce byly splněny. Bylo prozkoumáno množství senzorů, vhodných pro použití, včetně jejich komunikačních rozhraní. Získané poznatky jsou shrnuty v kapitole 3. Pro problém komunikačního protokolu byl po delším zkoumání navržen protokol MQTT. Jeho vlastnosti a výhody, včetně dalších zkoumaných protokolů, jsou shrnuty v kapitole 6. Navržený firmware pomocí přehledného webového rozhraní dovoluje nastavit připojení různých senzorů. V případě potřeby přidat další senzory, které neodpovídají již přednastaveným možnostem, je zdrojový kod vhodně členěn a dostatečně přehledný a je potřeba ovlivnit pouze minimální část kódu. Postup je popsán v kapitole 15. Firmware také prošel nekolikanásobným testováním, byl spuštěn s intervalovým nastavením několik hodin v kuse a neprojevil se žádné chyby.

Ke konci vývoje byl nasazen do pokoje spolu s pohybovým senzorem a zařízení bylo kontinuálně zapojeno, podle dat přijímaných na server zařízení fungovalo dle očekávání.

V případě pohybového čidla je třeba dát pozor na dostatečně kvalitní zdroj. Např. při napájení z počítače se senzor jeví jako v pořádku a při osazení na místo s 5V adaptérem do zásuvky, posílá nesprávná data v podobě spínání, i když před ním není pohyb, případně nesepnutím při pohybu. Chyba je způsobena nekvalitním, zarušeným adaptérem.

Vzhledem k poměrně slabému výpočetnímu výkonu ESP8266 bylo při vývoji zjištěno, že webový server v AP módu má neúměrně vysokou latenci. Trvá velmi dlouho, až 5 vteřin, než se prohlížeč může přepnout z jedné stránky na druhou. Tato chyba se ve staničním módu nikdy neprojevila, proto je doporučeno v AP módu provést jen nejnnutnější nastavení a zbytek nakonfigurovat již ve staničním módu.

Literatura

- [1] [online]. HW Kitchen. [cit. 2019/30/12]. Dostupné z: <https://www.hwkitchen.cz/nodemcu-v3-esp8266-lua-wifi-ch340g/>.
- [2] *Eclipse Mosquitto* [online]. mosquitto.org. [cit. 2020/04/01]. Dostupné z: <https://mosquitto.org/>.
- [3] *ESP-01 WiFi Modul* [online]. Ai Thinker, 2015. [cit. 2019/27/12]. Dostupné z: <http://www.microchip.ua/wireless/esp01.pdf>.
- [4] *ESP-12E WiFi Modul* [online]. Ai Thinker, 2015. [cit. 2019/27/12]. Dostupné z: https://components101.com/sites/default/files/component_datasheet/ESP12E%20Datasheet.pdf.
- [5] *ESP8266EX Datasheet* [online]. Espressif, 2015. [cit. 2019/27/12]. Dostupné z: https://cdn-shop.adafruit.com/product-files/2471/OA-ESP8266__Datasheet__EN_v4.3.pdf.
- [6] *HTTPS* [online]. Wikipedia, 2019/11/09. [cit. 2020/14/01]. Dostupné z: <https://cs.wikipedia.org/wiki/HTTPS>.
- [7] *Hypertext Transfer Protocol* [online]. Wikipedia, 2019/13/09. [cit. 2020/04/01]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol.
- [8] M., L. *Požární senzor* [online]. Arduino Návody, 2016/03/08. [cit. 2019/23/12]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/pozarni-senzor.html>.
- [9] M., L. *Senzor hořlavých plynů* [online]. Arduino Návody, 2016/05/04. [cit. 2019/26/12]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/senzor-horlavych-plynu-mq-2.html>.
- [10] MALÝ, M. *Protokol MQTT: komunikační standard pro IoT* [online]. root.cz, 2016/29/06. [cit. 2020/04/01]. Dostupné z: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>.
- [11] *MQTT* [online]. iot-portal.cz, 2016/24/5. [cit. 2019/30/12]. Dostupné z: <https://www.iot-portal.cz/2016/05/24/mqtt/>.
- [12] *Půdní vlhkoměr* [online]. Eses. [cit. 2019/26/12]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/755/eses1474354607.pdf>.

- [13] PETR VIKTORIN, M. H. *MicroPython na ESP8266* [online]. Python.cz, 2016. [cit. 2019/30/12]. Dostupné z: <https://naucse.python.cz/lessons/intro/micropython/>.
- [14] *Pohybové čidlo* [online]. Eses. [cit. 2019/23/12]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/757/eses1500635995.pdf>.
- [15] *Sběrnice I2C* [online]. 2009/01/08. [cit. 2020/02/05]. Dostupné z: <https://www.root.cz/clanky/komunikace-po-seriove-sbornici-isup2supc/>.
- [16] *Sběrnice OneWire* [online]. 2004/17/11. [cit. 2020/02/05]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/rozhrani/sbornice-1-wiretm.html>.
- [17] *Senzor DHT11* [online]. 2016/02/07. [cit. 2019/23/12]. Dostupné z: <http://www.1cip.radekc.eu/clanek/arduino-a-cidlo-vlkosti-teploty-dht22>.
- [18] *Senzor MPU-6050* [online]. [cit. 2020/02/05]. Dostupné z: <https://www.gme.cz/modul-gyroskop-akcelerometr-i2c>.
- [19] *Ultrazvukový senzor* [online]. ElecFreaks. [cit. 2019/26/12]. Dostupné z: https://www.hwkitchen.cz/user/related_files/ultrazvukovy-meric-vzdalenosti-hc-sr04-datasheet.pdf.
- [20] VODA, Z. *Programujeme Arduino* [online]. Arduino.cz, 2014/14/10. [cit. 2020/14/01]. Dostupné z: <https://arduino.cz/programujeme-arduino/>.

⁰Pro offline použití je literatura na CD ve složce Literatura.