VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Alžbeta Kostelanská

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

IOT WIFI SENSOR VLHKOSTI PŮDY A TEPLOTY
IOT WIFI SOIL MOISTURE AND TEMPERATURE SENSOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Alžbeta Kostelanská**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Vladislav Škorpil, CSc.**
SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Studentka: Alžbeta Kostelanská ***ID:*** 214078

Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

IoT WiFi sensor vlhkosti půdy a teploty

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci jednoduchého bateriově napájeného sensoru/měřiče relativní vlhkosti půdy a teploty okolí pomocí WiFi modulu Espressif ESP32. Sensor bude napájen bateriově a připojen do libovolné WiFi sítě. Bude reportovat emailem překročení mezních nastavených hodnot a umožňovat plynulý zápis naměřených hodnot do vlastní cloudové služby (NodeRed, Domoticz, …) realizované na počítači Raspberry se zobrazením aktuálních hodnot a podrobným zpětným prohlížením hodnot a grafů. Důraz při návrhu je přenositelnost zařízení, bezpečnost přenosu dat a dlouhá provozní doba. Realizujte navržené řešení.

DOPORUČENÁ LITERATURA:

[1] Baur, A.: Soil Moisture Tester for houseplants, Elektor magazine 4/2001. ISSN 1757-0875

[2] Claussen, Mathias: MQTT Sensor Hub, Elektor magazine 5/2019. ISSN 1757-0875

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.
Konzultant: Ing. Ondřej Pavelka (Ademco CZ s.r.o.)

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Bakalárska práca sa zaoberá problematikou návrhu IoT WiFi senzora vlhkosti pôdy, vzduchu a teploty. Pre navrhnuté zariadenie sa predpokladá možnosť jednoduchého prenosu zariadenia, nízka spotreba a zabezpečená komunikácie.
V teoretickej časti práce sú popísané vybrané typy senzorov a ich fungovanie. Je predstavený modul ESP32 a jeho vlastnosti, ktorú sú pre navrhované zariadenie dôležité. Zahrnuté je aj predstavenie vybraných typov komunikačných protokolov využívajúcich WiFi rozhranie, ktoré sú vhodné pre komunikáciu v IoT.
V praktickej časti je zhrnutá realizácia návrhu zariadenia. Pri návrhu bol kladený dôraz na nízku spotrebu zariadenia. Problematika nízkej spotreby je riešená na úrovni hardwaru aj softwaru. Je kladený dôraz na čo najvyššiu nezávislosť zariadenia od užívateľa a zabezpečenú komunikáciu pri prenose dát.
Súčasťou praktickej časti návrhu je taktiež server pre spracovanie nameraných dát na RaspberryPi, grafické rozhranie pre užívateľa a systém upozornení na namerané hodnoty.

KĽÚČOVÉ SLOVÁ

ESP32, senzor, kapacitný senzor vlhkosti pôdy, DHT22, RaspberryPi, MQTT, TLS, deep-sleep

ABSTRACT

Bachelor thesis addresses the subject of IoT WiFi soil moisture sensor and temperature sensor based on ESP32 platform. The primary focus of the design is on low power consumption of the device and secure communication. The theoretical part of the thesis describes selected types of sensors. The ESP32 module and some of its features are also introduced. Lastly some communication protocols which are suitable for IoT and can be used with WiFi peripheral are described. The practical part of the bachelor thesis deals with hardware design of the sensor device and implementation of this design. The problem of low power consumption is implemented in hardware and in software solution. The focus is on independence of the device and secure communication. Lastly the concept of webserver used for storing and processing collected data on RaspberryPi platform and concept of sending the measured data to the end-user is introduced.

KEYWORDS

ESP32, sensor, capacitive soil moisture sensor, DHT22, RaspberryPi, MQTT, TLS, deep-sleep

KOSTELANSKÁ, Alžbeta. *IoT WiFi sensor vlhkosti půdy a teploty*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 77 s. Bakalárska práca. Vedúci práce: doc. Ing. Vladislav Škorpil, CSc.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Alžbeta Kostelanská
VUT ID autora:	214078
Typ práce:	Bakalárska práca
Akademický rok:	2020/21
Téma závěrečnej práce:	IoT WiFi sensor vlhkosti pôdy a teploty

Vyhlasujem, že svoju záverečnú prácu som vypracovala samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autorka uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomá následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Rada by som poďakovala vedúcemu práce Ing. Vladislavovi Škorpilovi, CSc., odbornému konzultantovi Ing. Ondřejovi Pavelkovi za konzultácie a podnety pri riešení práce a Ing. Ondřejovi Krajsovi, Ph.D. za konzultácie a pomoc pri hardwarovej časti návrhu.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam výpisov

Úvod

Siete IoT zažívajú v aktuálnom období dlhodobo veľký nárast. Ide o systém alebo sieť zariadení, ktoré spolu komunikujú cez sieťové rozhranie, či už s využitím licenčného alebo bezlicenčného pásma.

Zariadenia, ktoré sú súčasťou IoT sietí sú často riadené rôznymi typmi senzorov, napríklad senzormi reagujúcimi na pohyb, svetlo, zmeny teploty, tlaku alebo na vlhkosť. Využívajú sa v rôznych aplikáciách, či už v domácnostiach, pri koncepte tzv. chytrej domácnosti, ale aj v priemysle, a to najmä pri jeho automatizácii. Taktiež sa využívajú v riadení infraštruktúry miest, pri riadení dopravy a pri správe a distribúcii energie v priemysle aj v domácnostiach. Dôležitú časť využitia IoT systémov tvorí aj monitorovanie životného prostredia.

Bakalárska práca je zameraná na návrh IoT WiFi senzora na meranie vlhkosti pôdy, vzduchu a teploty vonkajšieho prostredia. Základom pre návrh je mikrokontrolér ESP32, senzory na meranie vlhkosti, teploty a platforma RaspberryPi s vlastným backendom a užívateľským rozhraním.

Navrhnuté zariadenie je prispôsobené umiestneniu vo vonkajšom prostredí. Vytvorené prepojenie s webovým rozhraní dbá na zabezpečenie prenášaných dát, a taktiež na prehľadné a užívateľsky priateľské rozhranie s možnosťou priameho zasielania nameraných dát užívateľovi v pravidelných intervaloch. Rovnako je v práci zahrnutá aj možnosť ukladania a zberu nameraných dát. Rieši sa možnosť ich ďalšie spracovanie do grafov pre lepšiu kontrolu a správu nameraných výsledkov.

V teoretickej časti je najskôr zhrnutá teória merania rôznych typov senzorov vhodných na použitie v navrhnutej aplikácii a využitý princíp merania vlhkosti pôdy a vzduchu. Je predstavená platforma ESP32 a niektoré jej vybrané vlastnosti dôležité pre navrhnuté zariadenie. Taktiež sú zhrnuté využité protokoly MQTT a HTTP spolu s možnosťami ich zabezpečenia.

V praktickej časti bakalárskej práce je popísaný výber jednotlivých senzorov, návrh zariadenia a jeho riadiacej PCB, návrh napájania mikrokontroléru, návrh krytia a princíp pre izoláciu zariadenia proti vlhkosti, keďže sa predpokladá umiestnenie zariadenia do vonkajšieho prostredia.

Nasleduje popis softvérového riešenia, systém riadenia a napájania senzorov, pripojenie zariadenia do siete a správa prihlasovacích údajov, odosielanie nameraných dát, distribúcia aktualizácií a softvérové zníženie spotreby zariadenia.

Poslednú časť tvorí popis webového a užívateľského rozhrania, rovnako ako spôsob poskytnutia nameraných údajov užívateľovi.

1 Teoretické poznatky

1.1 Senzory a princíp ich merania

Bakalárska práca je zameraná na návrh zariadenia na meranie vlhkosti a teploty. Nasledujúce kapitoly sa venujú zhrnutiu základných poznatkov o senzoroch schopných merať tieto veličiny.

Spomínané senzory využívajú nepriamy spôsob merania. Hodnotu meranej veličiny získajú jej prepočtom z inej hodnoty. V prípade spomínaných senzorov ide o prepočet hodnôt z permitivity a z jej závislosti na zmenách teploty, alebo z kapacity, ktorá sa mení v závislosti na vlhkosti prostredia.

1.1.1 Kapacitný senzor

Kapacitor je objekt zložený z dvoch elektricky vodivých platní, medzi ktorými je vzduchová medzera. Ako kapacitor môžeme z fyzikálneho hľadiska označiť takýto objekt, ak aj po jeho odpojení od zdroja elektrického napätia, dokáže medzi platňami udržať elektrický náboj. Ideálny kapacitor je teda charakterizovaný veľkosťou náboja platní q a rozdielom elektrického potenciálu medzi nimi, charakterizujúceho hodnotu napätia U. Kapacita C teda predstavuje pomer dvoch veličín

$$C = \frac{q}{U}. \tag{1.1}$$

V prípade kapacitného senzoru vychádzame zo zjednodušeného vzťahu pre kapacitor vo vákuu, pre ktorý platí

$$C = \frac{\epsilon_0 \cdot A}{d}, \tag{1.2}$$

kde veličina A predstavuje plochu platní kapacitora a d vzdialenosť medzi nimi, definujúce geometrické vlastnosti daného kapacitora. ϵ_r je relatívna permitivita vákua. Uvedený vzťah platí pre kapacitor s dvoma paralelnými platňami. V prípade kapacitora iného tvaru je potrebné zahrnúť do vzťahu špecifické geometrické vlastnosti.

Keďže geometrické vlastnosti kapacitného senzoru ostávajú počas celej doby jeho merania nemenné, zo vzťahu vyplýva, že jeho fungovanie je závislé od hodnoty permitivity.

Permitivita ϵ je ovplyvnená tým, aké dielektrické vlastnosti má materiál zaplňujúci priestor medzi platňami kapacitora. Dá sa teda hovoriť o relatívnej permitivite látky ϵ_r medzi platňami kapacitora, ktorá je od hodnoty ϵ_0 rozdielna násobkom dielektrickej konštanty danej látky.

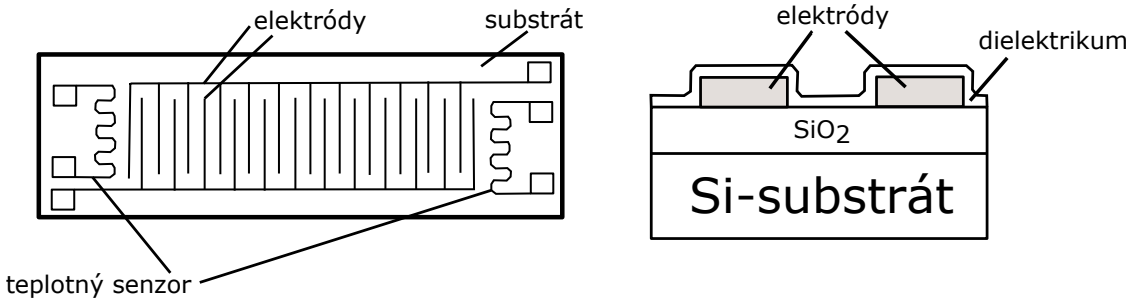
V kontexte kapacitného senzoru je daný dôraz na nestabilitu kapacitora vzhľadom na zmeny podmienok okolia, teda na ovplyvnenie jeho elektrických vlastností zmenou teploty alebo vlhkosti prostredia.

V prípade kapacitného senzoru vlhkosti sú teda využívané zmeny elektrických vlastností pôdy ako dielektrika na základe rozdielneho pomeru pevných častíc pôdy a vody.

Pôdu a senzor v tomto prípade vnímame ako jeden objekt (kapacitor) s určitými elektrickými a geometrickými vlastnosťami. Pri zmenách pomeru vody a pôdy sa menia dielektrické vlastnosti prostredia a senzoru. Z toho vyplývajú hodnoty kapacity. Táto závislosť je takmer lineárna.

Pre takto navrhnutý kapacitný senzor ide o zmeny v jednotkách až desiatkach pF [20], ktoré sa prejavia zmenou elektrických vlastností obvodu, ktorého súčasťou je senzor. Zmeny sú následne zaznamenané a sú z nich vypočítané hodnoty relatívnej vlhkosti prostredia.

Kapacitný senzor využíva nepriamy spôsob merania fyzikálnych veličín.



Obr. 1.1: Možné prevedenie kapacitného senzoru[20].

1.1.2 Odporový senzor

Elektrický odpor R je fyzikálna veličina popisujúca schopnosť vodiča viesť elektrický prúd. Je vyjadrený Ohmovým zákonom. Jeho hodnota je úmerná pomeru prúdu a napätia, ktoré vodičom prechádzajú.

Schopnosť samotnej látky viesť elektrický prúd popisuje merný elektrický odpor (rezistivita) ρ. Ten popisuje rezistívne vlastnosti vodiča dĺžky 1 meter s jednotkovou plochou prierezu. Ide o látkovú konštantu. Pre hodnotu odporu R v tomto prípade platí všeobecný vzťah

$$R = \rho \frac{l}{A}, \tag{1.3}$$

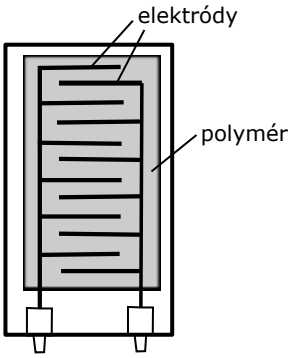
kde ρ je rezistivita materiálu, l a A vyjadrujú geometrické vlastnosti vodiča, jeho dĺžku a plochu prierezu.

Keďže závislosť odporu na zmenách teploty nie je lineárna je potrebné pre použitie v senzoroch aproximovať funkciu vhodnou krivkou. Zároveň je potrebné brať do úvahy aj chemickú čistotu materiálu, ktorý je na výrobu senzoru použitý.

Pri využití odporového senzoru na meranie teploty berieme ohľad na zmeny fyzikálnych vlastností spôsobené zmenou teploty prostredia. V prípade senzoru sa vytvára rezistor s nestabilnými obvodovými vlastnosťami, teda vysokou závislosťou zmien na teplote. Podľa [20] permitivita takto navrhnutého senzora stúpa s rastúcou teplotou v desiatkach $\Omega \cdot m$.

V prípade odporového senzoru na meranie relatívnej vlhkosti je kladený dôraz na výber materiálu, ktorého permitivita je silne ovplyvnená jeho schopnosťou pohltiť molekuly vody. Fyzikálne a elektrické vlastnosti tohto materiálu sú silne závislé od vlhkosti prostredia.

Typický odporový senzor vlhkosti pozostáva z keramického substrátu a elektródových vodičov obalených vo vrstve gélu, zväčša na báze organickej zlúčeniny. Príklad odporového senzoru sa nachádza na obrázku 1.2. Elektródy tvoria rezistor, ktorého zmeny odporu v závislosti na zmene vlhkosti sa zaznamenávajú.



Obr. 1.2: Typický odporový senzor

Odporové senzory sú veľmi pomalé (zväčša majú odozvu 10 až 30 s)[20]. Závislosť odporu na vlhkosti s rastúcou vlhkosťou klesá v jednotkách kΩ až stovkách MΩ.

Ide o aktívne senzory, musia byť napájané prúdom. Sú veľmi citlivé na jednosmerný prúd, preto sa pri ich napájaní odporúča striedavý prúd so symetrickým priebehom.

1.1.3 Senzory na meranie relatívnej vlhkosti

Vlhkostné senzory merajú relatívnu vlhkosť prostredia RH%, z anglického relative humidity. Tá vyjadruje pomer tlaku spôsobeného vodnou parou a jeho maximálnej dosiahnuteľnej hodnoty v danom prostredí.

Kalibrácia senzora vlhkosti je závislá na prostredí, v ktorom bude použitý. Senzor vlhkosti pôdy musí byť kalibrovaný na určitý typ pôdy. Postup kalibrácie takéhoto senzora sa nachádza v kapitole 1.1.5.

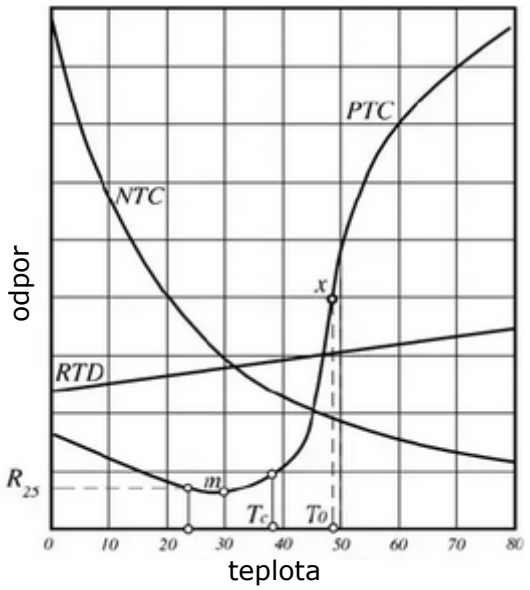
V práci je ako na meranie vlhkosti pôdy aj na meranie vlhkosti vzduchu použitý kapacitný senzor vlhkosti.

Medzi najčastejšie typy patrí polymérový a keramický kapacitný senzor na meranie vlhkosti. Polymérový senzor pracuje na princípe dvoch elektród, medzi ktorými sa nachádza polymérové dielektrikum. Molekuly vody v meranom prostredí zaplnia prázdne miesta v štruktúre polyméru a tým spôsobia zmenu jeho relatívnej permitivity. V prípade bežne využívaných polymérov ide o $\epsilon_r = 5$ pri normálnych podmienkach, kde ϵ_r vody je okolo 80. Táto zmena má lineárny priebeh a nie je nutná ďalšia aproximácia.

1.1.4 Senzory na meranie teploty

Pre meranie teploty prostredia sú najrozšírenejšie tri typy senzorových snímačov. Termočlánky, ktoré sú charakteristické veľmi veľkým meracím rozsahom, dosahujúcim tisícky °C, RTD (resistance temperature detector), alebo odporové teplotné detektory, ktoré sú charakterizované lineárnou závislosťou odporu na teplote pre veľký rozsah teplôt. Nasledujúca kapitola bude bližšie rozoberať tretí typ snímačov, teda odporové teplotné senzory – termistory.

Termistor je senzorom, ktorý meria teplotu nepriamo. Sleduje reakcie materiálu pri zmenách teploty. Meraná hodnota teploty je absolútnou hodnotou, čo znamená hodnotou na absolútnej stupnici. Rozlišujeme dva základné typy termistorov: NTC a PTC.



Obr. 1.3: Porovnanie teplotných charakteristík odporových senzorov[20].

Zatiaľ čo pre meranie precíznejšie NTC termistory sú charakterizované klesajúcim odporom pri rastúcej teplote. Pre PTC platí hodnota odporu rastúca zároveň s teplotou prostredia. Z grafu vyplýva, že zatiaľ čo RTD senzory sú vhodné na meranie veľkých teplotných rozsahov, pre NTC a PTC senzory platí, že ich presnosť závisí od využitej aproximácie, teda aj od rozsahu meraných teplôt.

Pre termistory je charakteristická veľmi vysoká citlivosť, ktorá je ale obmedzená malým rozsahom. Pre aplikácie v poľnohospodárstve alebo pri sledovaní vonkajšej teploty prostredia je rozsah dostačujúci. Pri vyššom meranom rozsahu sa presnosť znižuje. Zároveň trpia prehrievaním. Ich cena je v porovnaní s ostatnými spomínanými teplotnými senzormi veľmi nízka, čo má vplyv na ich široké využitie aj s prihliadnutím na ich kratšiu životnosť.

1.1.5 Kalibrácia kapacitného senzora na meranie vlhkosti pôdy

Pri kalibrácii v práci použitého kapacitného senzora môžeme vychádzať z niekoľkých možností.

Základným a najspoľahlivejším spôsobom je vytvorenie kontrolnej vzorky, s dopredu známou hodnotou relatívnej vlhkosti. Napríklad pri meraní vlhkosti pôdy je možné podľa [26] postupovať vysušením pôdy tepelnou úpravou na hodnotu blízku alebo rovnú 0 RH% a odmeraním vlhkosti vzorky nenakalibrovaným senzorom.

Následne je možné pridaním dopredu známeho množstva vody určiť meraný pomer pôdy a vody. Týmto je možné nepriamo určiť hodnotu RH% ešte pred samotným meraním prostredníctvom senzora. Hodnota nameranú nekalibrovaným senzorom je možné priradiť známej hodnote relatívnej vlhkosti.

Jednoduchším spôsobom je približné prirovnanie požadovaných hodnôt ku hodnotám im veľmi podobným. Pri postupe kalibrácie kapacitného senzora sa potom vychádza z predpokladu, že hodnota relatívnej permitivity vzduchu je veľmi podobná relatívnej permitivite suchej pôdy. Preto je postup nasledovný:

1. Nekalibrovaný senzor sa nechá voľne vo vzduchu. Takto nameraná hodnota sa použije ako vzorová hodnota a priradí sa jej hodnota vlhkosti 0 % .
2. Senzor sa vloží do nádoby s vodou. Takto nameraná hodnota sa použije ako vzorová pre hodnotu vlhkosti 100 RH%.

Namerané hodnoty pre 100 RH% a 0 RH% sa zahrnú do prepočtu, kde nameraná hodnota predstavuje hodnotu napätia obvodu, ktoré je ovplyvnené zmenou relatívnej vlhkosti a teda permitivity prostredia. Namerané údaje sa následne prepočítajú na relatívnu vlhkosť prostredia pomocou týchto dvoch hodnôt.

Pri kalibrácii je dôležité zahrnúť aj typ pôdy, ktorá sa nachádza v meranom prostredí. Rôzne typy pôdy sú schopné pohltiť rôzne množstvo vody.

1.2 ESP32

ESP32 je platforma vyvíjaná spoločnosťou Espressif. Vyznačuje sa nízkou spotrebou, hlavne v kombinácií s možnosťou využitia tzv. deep-sleep módu (1.2.3), ktorý prevedie mikrokontrolér do spánku v čase, keď nie je potrebné, aby spracovával dáta.

1.2.1 Základné vlastnosti ESP32

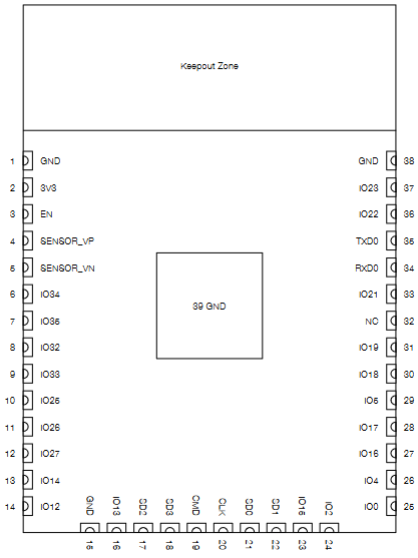
V [?] je uvedených niekoľko jeho základných vlastností a príklady ich implementácie v rôznych aplikáciách.

Platforma ESP32 umožňuje implementáciu OTA aktualizácií, rôznych komunikačných protokolov a šifrovanie vnútornej pamäte aj prijímanej a odosielanej komunikácie.

Ďalej vlastným integrovaným rozhraním pre WiFi protokol 802.11b/g/n, Bluetooth (BT) a Bluetooth Low Energy (BLE).

Vďaka vysokému výpočtovému výkonu, pomerne nízkej cene a ďalšími vlastnosťami sú mikrokontroléry pracujúce s modulom ESP32 je dobrým základom pre rôzne odvetvia IoT.

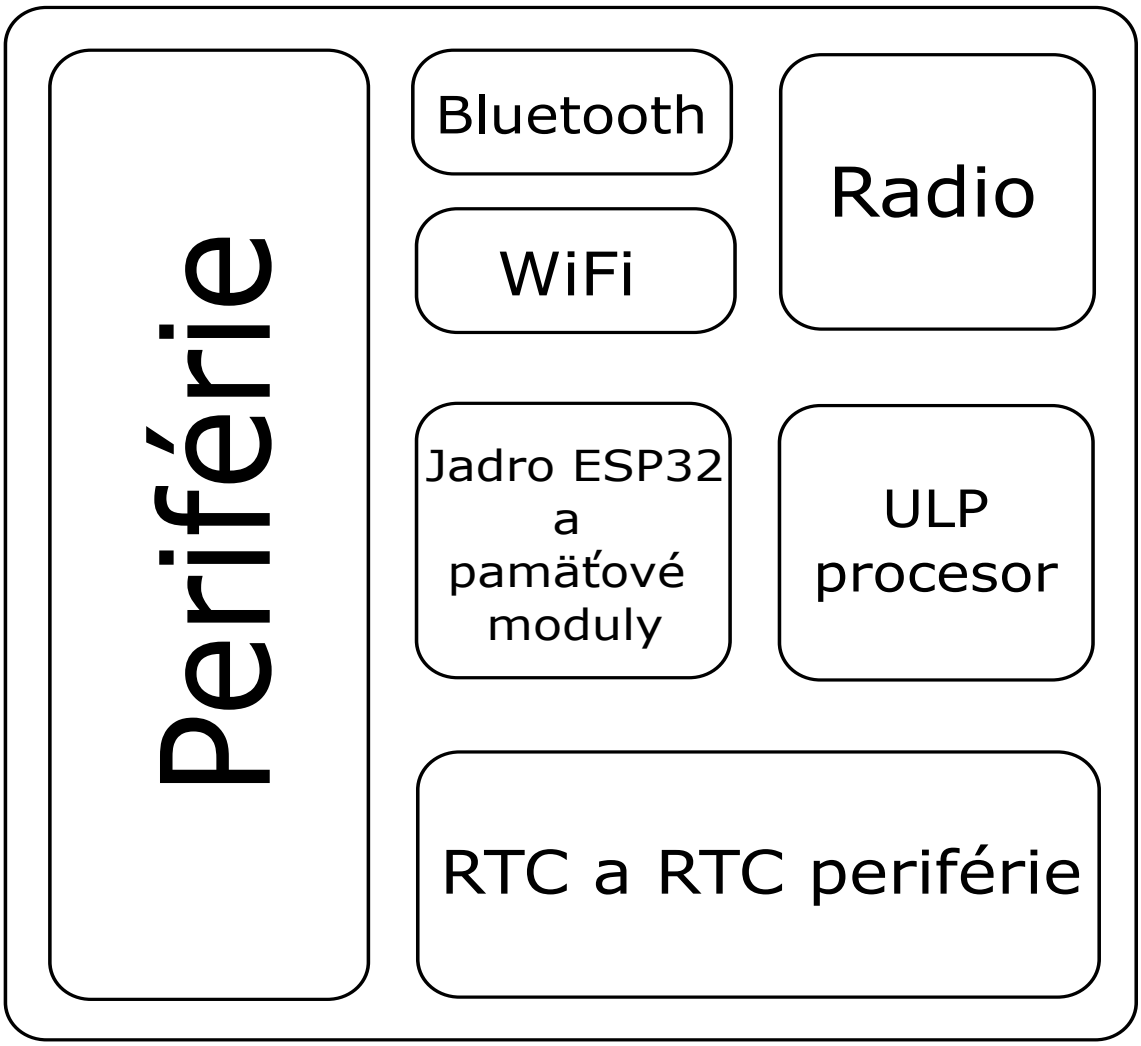
Vďaka knižnici vytvorenej samotnou spoločnosťou Espressif, sú mikrokontroléry navrhnuté na platforme ESP32 kompatibilné s obvykle používanými modulmi a vývojovým prostredím pre Arduino.



Obr. 1.4: Pinout ESP32[18]

Na obrázku 1.4 sa nachádza schéma dostupných pinov pre pripojenie rôznych modulov. Súčasťou štandardne predávaného modulu je 34 programovateľných GPIO.

V závislosti od modelu mikrokontroléru obsahuje až 18 12-bitových ADC prevodníkov a dva 8-bitové DAC prevodníky, 10 vstupov s možnosťou pripojenia dotykového senzoru, ktorý je možné použiť na prebudenie zariadenia z úsporného režimu. Niektoré moduly disponujú vnútorným teplotným senzorom. Modul taktiež poskytuje možnosť využitia PWM pre väčšinu z GPIO.



Obr. 1.5: Blokový diagram mikrokontroléru ESP32

1.2.2 ESP32-WROOM-32

Jednou z predávaných verzií mikrokontroléru platformy ESP32 je v práci použitý ESP32-WROOM-32. Jeho základom je čip ESP32-D0WDQ6. Ide o čip s dvojjadrovým procesorom, s maximálnou frekvenciou 240 MHz, podporou pre WiFi, BT a BLE.

Súčasťou čipu je low-power co-processor s veľmi nízkou spotrebou, schopný monitorovať periférie v čase, keď nie je potrebný vysoký výpočtový výkon. Hlavným príkladom jeho využitia je riadenie deep-sleep módu a hibernácie.

Rozhranie WiFi podporuje frekvenciu 2,4 GHz a protokoly 802.11b/g/n.

Procesor umožňuje využitie 4 MB SRAM, ktorá je pripojená na GPIO6 až GPIO11. Tieto piny nie je možné použiť na pripojenie modulov, ako bežné GPIO.

Podľa [18] je odporúčané napájanie minimálne 0,5 A zdrojom s napätím minimálne 3,0 V a maximálne 3,6 V. Typické napätie pre zabezpečenie správneho fungovania mikrokontroléru je 3,3 V, pri prekročení tejto hodnoty môže nastať tzv. brown-out efekt a následne opakovaný reštart zariadenia. Maximálny prúd privedený na výstupy mikrokontroléru môže byť 1,1 A. Pri prekročení tejto hodnoty sa riskuje poškodenie zariadenia.

Súčasťou mikrokontroléru sú 12-bitové ADC prevodníky. [18] uvádza ich maximálnu odchýlku 6 %, odchýlka sa znižuje na základe rozsahu, pri znížení rozsahu môže podľa [18] byť minimálne 2,3 %.

Súčasťou mikrokontroléru je aj mechanizmus šifrovania. Mikrokontrolér ponúka možnosť šifrovania Flash pamäte a bezpečného spustenia.

Podporovaný je 1024-bitový OTP založený na Vernamovej šifre a šifrovacie mechanizmy AES, RSA a iné. Mikrokontrolér vie pracovať s hashovacími funkciami, presnejšie využíva SHA-2. Šifrovanie je možné použiť pre dáta uložené v internej pamäti, ale aj pre komunikáciu prostredníctvom sieťového rozhrania.

1.2.3 Režimy zníženia spotreby

ESP32 ponúka niekoľko úrovní napájacích módov. Základným je aktívny mód, kedy sú napájané všetky periférie. Všetky vlastnosti mikrokontroléru sú aktívne. Spotreba zariadenia je v tomto prípade najvyššia, [18] uvádza hodnotu prúdu 160 až 260 mA.

Ďalším módom je režim modemu. V tomto prípade sú vypnuté všetky sieťové rozhrania a periférie. Oba procesory a RTC rozhranie sú aktívne. Spotreba v tomto režime je podstatne nižšia, podľa [18] 3 až 20 mA.

Z porovnania prvých dvoch uvedených módou vyplýva, že pri použití sieťových rozhraní spotreba zariadenia vzrastie približne 10 násobne. V tomto móde je možné aj po vypnutí sieťových rozhraní stále ponechať možnosť pripojenia cez WiFi. Toto je zabezpečené časovačom, ktorý prebudí sieťové rozhranie vo chvíli, keď je potrebné prijať packet vyslaný lokálnym AP s informáciami o stave siete a možnosti pripojenia.

Light sleep mód je rozšírením módu modemu. Okrem sieťových rozhraní a periférií je v ňom pozastavený aj samotný ESP32 procesor a vlastná pamäť zariadenia. Spotreba v tomto prípade klesne pod 1 mA.

V režime deep-sleep je napájaný ULP (ultra-low power) procesor a RTC rozhranie a pamäť. Ak je ULP procesor používaný na spracovanie dát je spotreba 0,15 mA. Prerušením jeho funkcií klesá spotreba samotného ESP32 čipu na 10 A. Pri prechode do deep-sleep módu sú všetky lokálne premenné uložené mimo RTC pamäť stratené. Preto je pri jeho používaní potrebné ukladať prihlasovacie údaje do pamäte RTC. To zabezpečí ich zachovanie aj po prebudení zariadenia.

Režimom s najnižšou spotrebou je režim hibernácie. Pri jeho využití sa nenapája žiadne z rozhraní mikrokontroleru. Napájaný je iba RTC časovač, ktorý zabezpečí prebudenie zariadenia po preddefinovanom čase. V tomto prípade je spotreba zariadenia okolo 2,5 μA.

Light a deep-sleep módy

ESP32 ponúka možnosť nastavenia deep-sleep módu[**?**], teda definíciu časových intervalov, kedy je zariadenie prevedené do spánku spolu s procesormi a veľkou časťou pamäte RAM. Výstupné a vstupné rozhrania sú vypnuté. V činnosti ostáva iba riadenie a udržiavanie systémového času. Prebudenie z tohoto režimu sa dá zariadiť rôznymi spôsobmi. ESP32 podporuje prebudenie v určitých časových intervaloch, použitím dotykového senzoru a externé prebudenie závislé od signálu poslaného do zariadenia.

ESP32 poskytuje taktiež možnosť tzv. light-sleep módu. Ten prevedie rozhrania do spánkového režimu, navonok rovnakého ako deep-sleep mód, stav pred prechodom do tohto režimu je zachovaný.

Na riadenie sa v tomto prípade používa zmena clockových frekvencií zariadenia. Zariadenie potom môže pokračovať v bode, kedy bolo prevedené do spánkového režimu. V prípade light-sleep módu sa ponúkajú rovnaké možnosti prebudenia ako pri pre deep-sleep móde. Pri porovnaní s deep-sleep módom je spotreba zariadenia vyššia, možnosť uchovanie pamäte je ale výhodou.

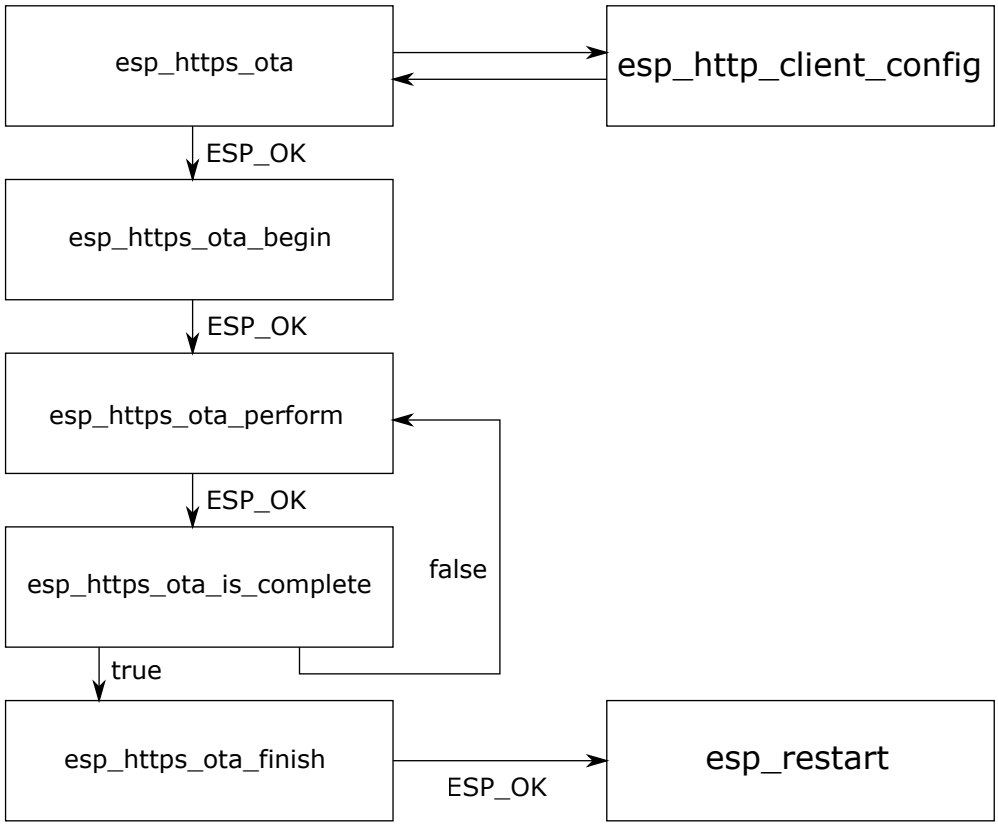
Pred začatím spánkového režimu musí byť vypnuté rozhranie WiFi. Na správne fungovanie pripojenia musí byť ale spojenie medzi AP a zariadením udržiavané. Podľa [18] ESP32 ponúka implementáciu automatickej funkcie zariadenia, ktorá kontroluje, či je potrebné nadviazať spojenie s AP a po dokončení tejto operácie prevedie zariadenie automaticky do light-sleep módu.

Pre oba režimy platí, že v prípade zabezpečenej komunikácie sa spotreba bezprostredne po prebudení zo spánkového režimu zvyšuje, a to hlavne z dôvodu zvýšených nárokov na zariadenie, ktoré sú spojené s vytvorením zabezpečenej komunikácie medzi AP a samotným zariadením.

1.2.4 Systém Over-The-Air aktualizácií

ESP32 je jednou z platforiem, ktoré podporujú systém OTA, teda systém bezdrôtových aktualizácií ("Over the Air"), a to použitím svojho vlastného sieťového rozhrania. Podľa [17] zariadenie v tomto prípade pracuje s internou pamäťou rozdelenou do segmentov alebo sekcií. Stará verzia firmwaru je načítaná v jednom segmente, nová je súčasne prijatá do ďalšieho práve nevyužitého, bez potreby prerušenia fungovania zariadenia.

Samotný systém OTA aktualizácií s použitím HTTPS protokolu prebieha podľa schémy na obrázku 1.6. Na začiatku je volaná funkcia pre samotný OTA proces. Následne sa naviaže HTTPS spojenie so serverom, kde sa nachádza nová verzia firmwaru. Prevedie sa proces aktualizácie a po jeho úspešnom dokončení sa prevedie reštart zariadenia. V každej fáze procesu systém čaká na prijatie správy ESP_OK. V prípade prijatia ESP_ERROR sa proces preruší. Zariadenie v tomto prípade pokračuje v bežnom fungovaní s prechádzajúcou verziou firmwaru.



Obr. 1.6: Mechanizmus OTA aktualizácií[**?**]

V tabuľke 1.1 je uvedený príklad možného rozdelenia pamäte na oddiely ("partitions"). Schéma uvedená v tabuľke je použitá v navrhnutom zariadení. Pozostáva

z dvoch 2 MB oddielov pre ukladanie firmwaru a 200 kB oddielu pre SPIFFS dáta.

Názov oddielu	**Typ**	**Popis**	**Offset**	**Veľkosť**
NVS	Data	NVS	0x9000	0x5000
OTAdata	Data	OTA	0xe000	0x2000
App0	App	OTA0	0x10000	0x1E0000
App1	App	OTA1	0x1F0000	0x1E0000
SPIFFS	Data	SPIFFS	0x3D0000	0x30000

Tab. 1.1: Schéma oddielov pre použitie OTA aktualizácii

1.3 MQTT a HTTP protokol

Na komunikáciu IoT zariadení medzi sebou alebo so serverom alebo brokerom je potrebné využiť vhodný komunikačný protokol.

IoT zariadenia často používajú protokoly pracujúce v bez licenčnom pásme. Vzhľadom na použitie platformy ESP32 je možné použiť aj protokoly MQTT a HTTP. Hlavne HTTP je všeobecne rozšírené na komunikáciu v sieti Internet.

Dôležitou súčasťou komunikačných protokolov je aj ich zabezpečenie. Jeho prevedenie by malo byť na princípe AAA, z anglického authentication, authorization and accounting (autentizácia, autorizácia a účtovanie.

Všetka komunikácia medzi klientskými zariadením a serverom by mala prebiehať zabezpečene, server by mal naväzovať spojenie len s overenými klientmi, ktorých prihlasovacie údaje sú mu známe. Iba užívatelia s poverením by mali byť schopný pristupovať ku zdrojom. Všetky akcie serveru a klientov by mali byť zaznamenávané.

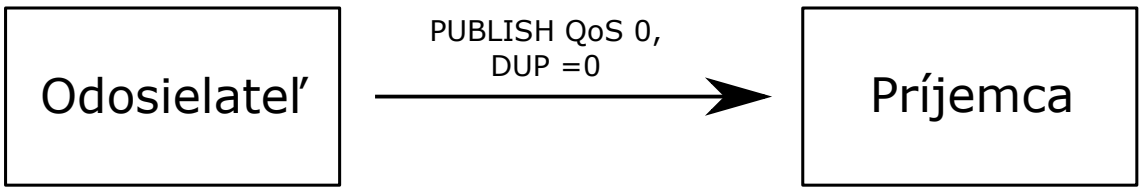
1.3.1 MQTT

MQTT (Message Queuing Telemetry Transport) protokol je navrhnutý pre komunikáciu v rámci IoT sietí. Je založený na princípe publish/subscribe, zverejňovania a odberu ku téme. Je výhodný hlavne pre siete, kde sa prenáša malé množstvo dát, majú obmedzenú šírku pásma, alebo je funkcionalite založená na jednoduchých úkonoch.

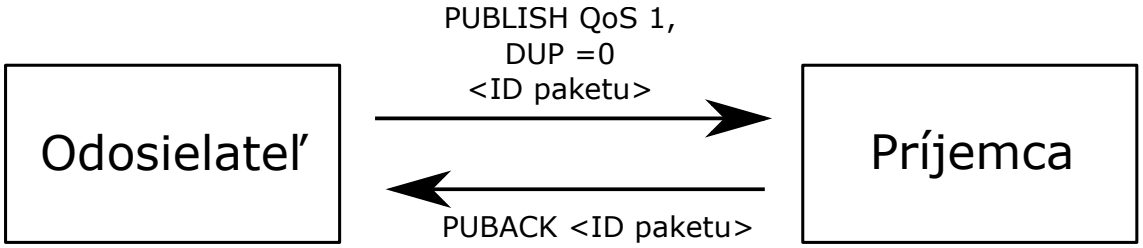
MQTT umožňuje obojstrannú komunikáciu. Správy publish (zverejnenie) sú posielané od klienta na server. V prípade správ subscribe, sa klient prihlasuje k odberu správ z danej témy. V prípade MQTT je serverom broker, ktorý riadi komunikáciu medzi jednotlivými zariadeniami, rozhraniami a aplikáciami. Najrozšírenejším brokerom je Mosquitto, pod správou nadácie Eclipse.

Obrázky 1.7 1.8 a 1.9 [25] ilustrujú úrovne kvality služieb pre protokol MQTT. Protokol rozlišuje tri úrovne. Najmenšie zabezpečenie predstavuje QoS 0, v tomto nastavení klient posiela správy bez požiadavky potvrdenia o doručení. Klient v tomto prípade nemá žiadnu správu o tom, či odoslaný paket bol prevzatý brokerom. Pri použití QoS 1 server odosiela potvrdenie o doručení paketu. Najvyššiu úroveň kvality služieb predstavuje QoS 2, kedy dochádza k obojsmernému potvrdeniu prijatia správy.

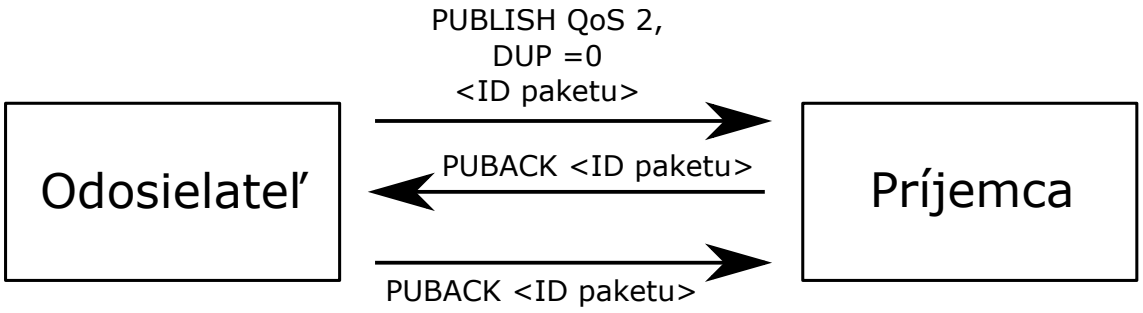
V súčasnej dobe sú používané tri verzie MQTT. Ide o štandard QASIS MQTT 5, MQTT 3.1.1, štandardizovaný organizáciou ISO a OASIS. Posledným je zastaralý, ale niektorými aplikáciami stále podporovaný MQTT 3.1.



Obr. 1.7: MQTT QoS 0



Obr. 1.8: MQTT QoS 1



Obr. 1.9: MQTT QoS 2

Pre MQTT sú rezervované porty 1883 pre nezabezpečenú a 8883 pre zabezpečenú komunikáciu. Protokol funguje na princípe požiadavka/odpoveď (request/response).

1.3.2 HTTP

HTTP (Hypertext Transfer Protocol) je protokol vytvorený pre komunikáciu medzi serverom a klientmi. Protokol funguje na princípe požiadavka/odpoveď (request/response). Primárne používa port 80 pre nezabezpečenú a 443 pre zabezpečenú komunikáciu.

Pri komunikácií je najskôr zo strany klienta poslaná žiadosť o dáta. Dáta môžu predstavovať rôzne typy formátov ako napríklad HTML súbory, žiadosti o výber dát z databázy, audio, video zdroje a podobne. Súčasťou požiadavky klienta musí byť definícia verzie použitého protokolu, URI s informáciou o type požadovanej služby a typ metódy požiadavky na základe požadovanej akcie (GET, POST, PUT, DELETE a pod.).

Server odpovedá na požiadavku správou, v ktorej hlavičke sa nachádza informácia o použitom protokole a kóde. Ten zastupuje kladnú odozvu serveru alebo chybový kód.

Komunikácia začína požiadavkou zo strany klienta, server požiadavku prijme a spracuje jej obsah. Výsledok tejto akcie následne odošle klientovi ako odpoveď. Klient prijme odpoveď a spracuje ju podľa svojich potrieb.

HTTP je určený pre širokú škálu aplikácií. Jeho využitie je možné v jednoduchých systémoch, kde nie je potrebný vysoký výkon, ale aj v systémoch, kde dochádza ku veľkému množstvu odoslaných a prijatých správ.

1.3.3 Zabezpečenie

Pre IoT aplikácie je otázka zabezpečenia komunikácie dôležitou súčasťou návrhu zariadení.

Jedným z dôvodov je ochrana zariadenia pred škodlivým softwarom a útokmi. V krajnom prípade by mohlo dôjsť ku poškodeniu samotného zariadenia.

Ďalej je dôležitá aj ochrana a zabezpečenie spracovaných dát. V prípade zariadenia navrhnutého na riadenie senzorov by mohlo dôjsť ku zámene nameraných dát, prípadne ich pozmeneniu. V prípade použitia senzora na riadenie iných aplikácií, napríklad zavlažovania alebo klimatizácie by týmto spôsobom mohlo dôjsť ku škodám.

Zabezpečiť nie je potrebné iba namerané dáta. Pri použití mechanizmu OTA aktualizácií je dôležité zabezpečiť aj proces distribúcie firmwaru medzi serverom alebo úložiskom a samotným zariadením. V prípade útoku by mohol byť útočníkom podvrhnutý firmware, ktorý by zariadenie považovalo za novú aktualizáciu. V tomto prípade by mohol byť narušený celý systém senzorov alebo zariadení v sieti.

MQTT aj HTTP protokoly podporujú šifrovanie komunikácie pomocou TLS a SSL. V prípade zabezpečenej komunikácie je odporúčané okrem samotného šifrovania používať u klientov a serveru aj platné certifikáty.

Tie v prvom rade obmedzia hrozbu útoku Man-In-The-Middle. V tomto prípade sa zariadenie útočníka vystupuje ako zariadenie serveru alebo klienta v sieti. Útočník nepozná certifikát zariadenia, za ktoré sa vydáva. Pri naviazaní spojenia sa kontroluje platnosť certifikátu, komunikácia so zariadením útočníka sa na základe neplatného certifikátu preruší.

Zariadenie môže využívať rôzne typy certifikátov. Najvýhodnejší z dlhodobého hľadiska a udržateľnosti systému je CA certifikát (certifikát certifikačnej autority). Certifikát má v tomto prípade platnosť niekoľko desiatok rokov. Nie je ho potrebné obnovovať ručne, o jeho obnovenie sa stará certifikačná autorita.

Najjednoduchším spôsobom je generovanie certifikátu s vlastným podpisom. Pri generovaní certifikátu sa najprv vygeneruje súkromný kľúč, ktorý sa použije pri šifrovaní vygenerovaného certifikátu.

Pri generovaní samotného certifikátu je dôležité, aby sa doménové meno zariadenia zhodovalo s menom uvedeným v certifikáte. Certifikát sa generuje obvykle so životnosťou jeden rok, životnosť si ale užívateľ volí pri generovaní sám.

Pred vypršaním jeho platnosti je preto dôležité myslieť na jeho vygenerovanie nanovo. Po vypršaní platnosti sa preruší komunikácia so zariadením, keďže pri jej naviazaní certifikát neprejde kontrolou na jeho pravosť a platnosť.

2 Hardwarova časť návrhu

2.1 Výber senzorov

2.1.1 Senzory pre meranie relatívnej vlhkosti pôdy

Pre senzory na meranie relatívnej vlhkosti pôdy existuje na trhu niekoľko možností. Ide najmä o analógové senzory.

Veľmi časté sú senzory generujúce vlny alebo pulzy a čítajúce hodnotu ich odrazu v prostredí. Vychádzajú z princípu vlhkej pôdy ako vodivého prostredia a odčítavajú buď vplyv generovaného elektrického poľa na toto prostredie alebo dobu odrazu pulzov v prostredí. Vyznačujú sa veľmi dobrou životnosťou, sú vhodné na nasadenie nie len v poľnohospodárstve, ale aj pre ťažký priemysel a aplikácie, kde je potrebná vysoká presnosť, ako napríklad baníctvo. Hlavne pri kalibrácii na daný typ pôdy v prostredí dosahujú veľmi dobré výsledky. Nevýhodou je ich vyššia cena.

Veľmi rozšírenými, najmä kvôli nízkej cene, sú kapacitné a odporové sondy. Ako už bolo spomenuté v 1.1.1 využívajú fyzikálne vlastnosti pôdy ako dielektrika. Ich hlavné vlastnosti zhrňuje tabuľka 2.1.

Vybraná vlastnosť	Kapacitný senzor[9]	Odporový senzor[10]
Typ senzoru	analógový	analógový
Merací princíp	zmeny kapacity	zmeny odporu
Napájacie napätie	3,3 až 5,5 V	3,3 V alebo 5 V
Napätie na výstupe	0 až 3 V	0 až 4,2 V
Napájací prúd	5 mA	35 mA
Počet pinov	3 (VCC, GND, AOUT)	3 (VCC, GND, AOUT)

Tab. 2.1: Porovnanie vlastností senzorov vlhkosti pôdy

Keďže sonda odporového senzoru je obvykle odkrytá, senzor je náchylný na koróziu. Tá by mohla poškodiť rastliny v pôde, kde je senzor použitý. V prípade kapacitného senzora je plocha, ktorá sa vkladá do pôdy pokrytá ochrannou vrstvou.

Oba senzory majú odkrytý riadiaci obvod, preto je potrebné pred ich použitím hlavne v exteriéry tento obvod zabezpečiť izoláciou.

Hlavne z pohľadu dlhšej životnosti a nízkej ceny bol pre návrh vybratý kapacitný senzor vlhkosti pôdy. Jeho ďalšou výhodou oproti odporovému senzoru je nižšia spotreba, keďže jeho prúdový odber je podľa [9] 5 mA. Nižší prúdový odber je výhodou pri napájaní zariadenia z batérie. Potreba kalibrácie na základe meranej pôdy sa dá vykompenzovať na softwarovej úrovni.

Vybratý senzor je označený ako Capacitive soil moisture sensor V2.0. Jeho základom je 555 časovač generujúci signál. Zmena kapacity spôsobená zmenou vlhkosti prostredia sa prejaví na odozve senzoru na generovaný signál. Senzor meria relatívnu vlhkosť pôdy. S pridaním senzoru teplotu a tlaku by bolo možné dopočítať absolútnu hodnotu vlhkosti prostredia, táto možnosť ale v práci nie je zahrnutá.

2.1.2 Senzor teploty a relatívnej vlhkosti vzduchu

V súčasnej dobe je v IoT aplikáciach čoraz viac rozšírené použitie digitálnych teplotných senzorov, ktorých hlavnou výhodou je možnosť odčítania meranej hodnoty priamo zo signálového pinu senzoru.

Na rozdiel od analógových senzorov nie je potrebné myslieť na dodatočný ADC prevodník a následné softwarové prepočítavanie odčítanej hladiny signálu. Využitie digitálnych senzorov z tohto hľadiska predstavuje pre návrh zariadenia značné zjednodušenie.

Digitálne senzory majú taktiež niekoľko ďalších výhod, ktoré napomáhajú ich rozšírenie do rôznych aplikácii. Hlavnou výhodou je vyššia presnosť, keďže pri odčítaní nie je potrebný dodatočný prevodník, na ktorom by mohli vzniknúť straty. Výsledky merania analógovým senzorom môžu byť ovplyvnené starnutím obvodových súčiastok. Digitálne senzory vykazujú nižšiu spotrebu, ktorá rastie v čase merania a spracovania dát a nižším prehrievaním samotného senzoru. Nevýhodou oproti analógovým senzorom je ich menší merací rozsah.

Pri výbere senzora na meranie teploty a vlhkosti vzduchu sú možné dva postupy. Prvou možnosťou je vyber dvoch osobitných čidiel. Druhou je výber jedného modulu obsahujúceho senzory pre obe funkcie.

Modul senzoru	DHT11[11]	DHT22[4]	BME280[12]
Merané veličiny	teplota a vlhkosť	teplota a vlhkosť	teplota, vlhkosť a tlak
Napájacie napätie	3 až 5,5 V	3 až 6 V	1,71 V až 3,6 V
Napájací prúd	max. 1 mA	max. 1,5 mA	max. 2,8 mA
Teplota			
Rozsah	0 až 50 °C	40 °C až 80 °C	40 °C až 85 °C
Relatívna odchýlka	2 °C	max. 0,5 °C	max. 1,0 °C
Rozlíšenie	1 °C	0,1 °C	0,01 °C
Vlhkosť vzduchu			
Relatívna odchýlka	±4 %	max. ±5 %	±3 %
Rozlíšenie	1 %	±1 %	0,008 %

Tab. 2.2: Dostupné modul senzorov teploty a vlhkosti

V práci bol zvolený druhý prístup. Tabuľka 2.2 obsahuje údaje k dostupným senzorom, ktoré by bolo možné pri návrhu použiť.

Pre navrhnuté zariadenie bol vybratý modul DHT22. V porovnaní s ostatnými senzormi netrpí vysokým prehrievaním a je pomerne presný. DHT22 je pomalý senzor. Podľa [4] je perióda pre odčítavanie dát 2 s. Senzor má pomerne malú spotrebu. V prípade, že číta dáta je napájací prúd maximálne 1,5 mA.

Keďže sa predpokladá umiestnenie senzora vo vonkajšom prostredí je potrebné senzor zabezpečiť proti vlhkosti. Dokumentácia ku senzoru uvádza hmlu a hmlisté prostredie ako rizikový faktor, ktorý by ho mohol poškodiť.

Senzory sú pripájané ku dvom pinom mikrokontroléru. Jeden z pinov je vždy využitý na čítanie dát, druhý na napájanie. Kapacitný senzor používa analógový GPIO36 na čítanie dát GPIO33 ako digitálny výstup na riadenie napájania. DHT22 využíva digitálny GPIO2 a GPIO25 na napájanie. Na prototype zariadenia sú senzory napájané priamo z batérie. Pre zníženie spotreby je potrebné realizovať úpravu tak, aby napájanie mohlo byť riadené vysokou hladinou signálu z GPIO pinov. Odovzdané súbory už túto zmenu zahŕňajú.

2.2 Návrh PCB

Keďže v práci je kladený dôraz na zabezpečenie dlhej životnosti batérie je potrebné tomu prispôsobiť aj samotné obvod s ESP32.

Pri návrhu samotnej dosky, do ktorej je mikrokontrolér osadený, sa vychádzalo z podmienky na čo najnižšiu spotrebu. Základ pre návrh tvorili bežne dostupné moduly Wemos Lolin 32 a Wemos D1 mini, ktorý ale používa starší mikrokontrolér spoločnosti Espressif ESP8266.

Ďalším zdrojom pre návrh výsledného prototypu je sprievodca návrhom hardwaru s platformou ESP32 vydaného spoločnosťou Espressif[19].

V prvom rade bol odstránený USB-UART prevodník, keďže pre navrhované zariadenie po prvom nahraní firmwaru už nie je potrebný. Zariadenie je navrhované s plným využitím systému OTA aktualizácii. Rovnako sa z návrhu odstránila dvojica tranzistorov slúžiaca pri programovaní zariadenia. Keďže po nasadení zariadenia do prostredia sa nepredpokladajú časté zmeny firmwaru, nie je ich prítomnosť v návrhu nutná.

Podľa [19] je pre zabezpečenie napájacieho pinu mikrokontroleru pred poškodením vysokými skokovými hodnotami prúdu pri komunikácii cez WiFi je potrebné čo najbližšie ku pinu umiestniť dostatočne veľké kondenzátory. Sprievodca návrhom odporúča použiť hodnoty 100nF a 10µF.

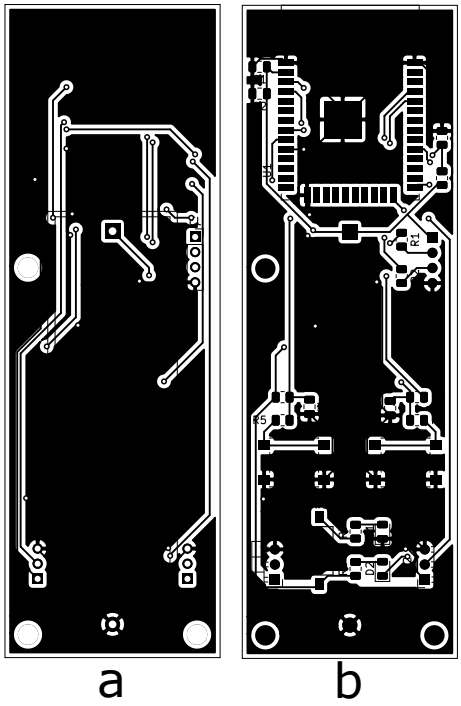
Diódy, ktoré sú súčasťou bežných modulov sa v návrhu zachovali, sú nahradené nízko prúdovými. Ich hlavnou funkciou je signalizácia pri prvotnom oživení zariade-

nia, keďže zariadenie bude ponechané v puzdre a vyberané iba v prípade poruchy alebo za účelom výmeny batérie.

Po oživení a pripravení zariadenia na meranie je možné obe diódy vypnúť. Pri dióde pripojenej na digitálny pin mikrokontroleru je toto zabezpečené jeho prepnutím na nízku úroveň, pre signalizačnú diódu napájania fyzický prerušením obvodu predradeným prepínačom.

Možným vylepšením, ktoré ale nie je súčasťou finálneho prototypu zariadenia je možnosť nabíjacieho obvodu pre použitú batériu.

Na obrázku 2.1 je výsledný nákres vyrobeného PCB. Obrázok a predstavuje jeho zadnú stranu, b je predná strana vyrobeného prototypu.



Obr. 2.1: Nákres PCB zariadenia

Zapojenie samotného ESP32 sa nachádza na obrázku 2.2. Dátové piny senzorov sú privedené na analogový výstup pre senzor GPIO36 a na digitálny GPIO2. Na sledovanie napätia na batérii slúži ADC pin 32.

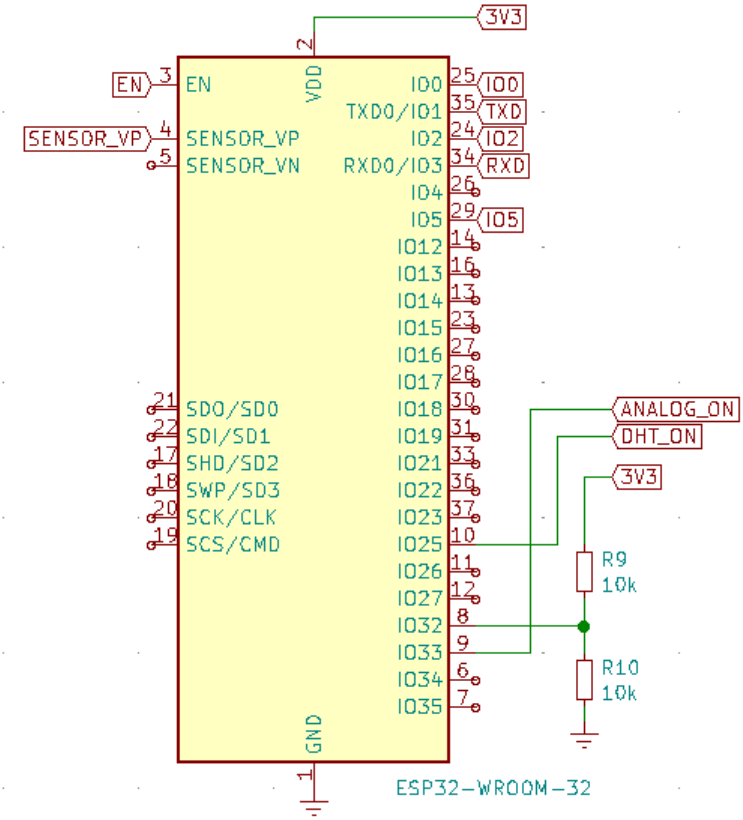
GPIO1 a 3 sú privedené na USB-UART prevodník. Ten slúži na prvotné nahranie firmwaru na mikrokontrolér. Prevodník nie je pevnou súčasťou obvodu, je použitý iba v prípade prvého spustenia dosky, prípadnej závažnej chyby vzniknutej pri OTA aktualizovaní.

Na GPIO5 je privedená dióda, ktorá hlási prebudenie zariadenia z režimu spánku. Za iných okolností nie je napájaná.

Pin EN a GPIO0 slúžia na ručné spustenie reštartu a načítania zariadenia. Sú na nich pripojené programovacie tlačidlá.

V zapojení na obrázku 2.2 je zahrnuté pripojenie napájania senzorov na digitálne GPIO. Výsledný prototyp zariadenia má napájanie senzorov vedené priamo na batériu. Pre výsledné znženie spotreby je ale potrebné previesť túto zmenu. Rovnako aj delič pripojený na GPIO32, ktorý slúži na odčítanie napätia batérie by mohol byť napájaný z pinu, nie priamo z batérie zariadenia.

Kompletná schéma zapojenia je súčasťou prílohy.



Obr. 2.2: Zapojenie ESP32

2.2.1 Napájanie

Jednou z dôležitých vlastností navrhnutého zariadenia je možnosť jeho prenosu a umiestnenia v exteriéry, teda z toho vyplývajúce batériové napájanie. Keďže častá výmena batérie nie je žiadúca, pri návrhu sa musí vziať do úvahy aj minimalizácia spotreby zariadenia.

V IoT aplikáciách je využívaných niekoľko, najmä dobíjateľných, typov batérií. Medzi veľmi rozšírené patria dobíjateľné Li-Ion a Li-Pol články. Vďaka napätiu 3,3 V sa ako výhodná možnosť pre navrhované zariadenie ponúka využitie LiFePo4 batérií.

Ďalej je možné využiť batérie vyrobené z alkalických zemín. Tie majú ale niekoľko nevýhod. Hlavnou je nemožnosť ich dobíjania a výstupné napätie 1,5 V. Na napájanie zariadenia s napájacím napätím 3,3 V by bolo potrebné použiť tri takéto články a výsledné napätie upraviť ochranným obvodom. Ich cena je ale v porovnaní s ostatnými článkami veľmi nízka a na rozdiel od ostatných článkov, ktoré nie je možné nabíjať majú pomerne vysokú kapacitu.

Z vyššie uvedeného vyplýva, že pre aplikácie pracujúce s mikrokontrolérmi s 3,3 V logikou je výhodné využitie LiFePo4 článkov. Ich výstupné napätie je 3,2 až 3,3 V. Pri využití ich nie je potrebné riešiť ochranný obvod a môžu nimi byť napájané priamo piny mikrokontroléra. Výhodou je možnosť dobíjania, podobne ako Li-Ion a Li-Pol články sa predávajú v rôznych veľkostiach, tvaroch, prevedeniach a kapacitách. V porovnaní s Li-Ion článkami majú vyššie samovybíjanie, najmä bezprostredne po prvom použití, sú ale oveľa stabilnejšie.

Samotná minimalizácia spotreby je prevedená ako na hardwarovej, tak aj na softwarovej úrovni. Zariadenie pravidelne prechádza do deep-sleep módu. Ostáva teda napájaný iba ULP procesor, RTC pamäť a RTC periférie, a to hlavne z dôvodu uloženia prihlasovacích údajov pre AP do RTC pamäte, teda možnosti znova pripojenia sa do lokálnej siete. Užívateľ nemusí pri každom prebudení zariadenia realizovať pripojenie k AP ručne. Bližší princíp fungovania napájacích módov na platforme ESP32 bol popísaný v predchádzajúcej kapitole 1.2.3.

Z režimu spánku je zariadenie prebudené každých 10 minút, po odmeraní potrebných údajov prechádza do režimu spánku. Spolu so zariadením sú uspávané aj oba senzory, keďže ich napájanie je riešené napájaním z pinov mikrokontroléra. Najväčšiu spotrebu zariadenie vykazuje pri vytváraní WiFi spojenia a posielaní packetov. Podľa katalógových údajov by sa spotreba pri bežnej prevádzke s použitím sieťových rozhraní mala pohybovať okolo 160 až 260 mA. V prípade prechodu do deep-sleep módu je spotreba výrazne nižšia.

Do návrhu je zahrnuté sledovanie napätia na batérii cez ADC prevodník mikrokontroléru.

ESP32 používa 12-bitový prevodník. Odčítava hodnoty v rozsahu 0 až 4095, kde hodnota 0 reprezentuje 0V a 4095 je možné priradiť hodnote 3,3V. Pred pin je zaradený napäťový delič, a to z dôvodu nelinearity prevodovej funkcie pre hodnoty blízke maximu a minimu rozsahu[15].

Pre výpočet kalibračnej konštanty na výpočet napätia batérie platí vzťah

$$k = \frac{U_{voltmeter}}{ADC}. \tag{2.1}$$

$U_{voltmeter}$ predstavuje napätie batérie nemerané voltmetrom a ADC je hodnota odčítaná z ADC prevodníka. Hodnota z pinu sa na napätie batérie prevedie vynásobením tejto hodnoty kalibračnou konštantou. Následne je potrebné túto hodnotu ešte

zdvojnásobiť, keďže pred pinom je pripojený delič, ktorý znižuje napätie privedené na pin o polovicu.

Pre stredné hodnoty je priebeh prevodovej funkcie lineárny, preto je možné po vynásobení hodnoty kalibračnou konštantou považovať výsledok za primerane presný.

Nabíjanie

Vhodným doplnením, ktoré ale nie je realizované vo výslednom prototype zariadenia je doplnenie o nabíjací obvod. Nabíjací obvod je možné realizovať za pomoci jednotky na riadenia nabíjania LiFePo4 batérie.

Keďže LiFePo4 batérie majú pri plnom nabití napätie maximálne 3,6 V je potrebné vybrať riadiacu jednotku nabíjania na základe tohto parametru.

Z predávaných riadiacich jednotiek je možné použiť napríklad CN3058E alebo MCP73123. V predávaných nabíjacích moduloch sa často využíva jednotka TP5000.

Pre jednotku MCP73123 je napájací prúd pri nabíjaní je typicky 0,7 mA. Jeho maximálna hodnota je 1,5 mA. Po dosiahnutí napätia 3,6 V na nabíjanej batérii jednotka prejde do úsporného režimu. Jej napájací prúd je 4 až 5,5 μA[24].

Keďže pri návrhu zariadenia bol kladený dôraz na zníženie spotreby je potrebné povedať, že pridanie nabíjacieho obvodu zvýši jeho celkovú spotrebu. Nabíjací obvod uľahčí situáciu, kedy dôjde k vybitiu batérie, samotná výdrž batérie sa ale zníži.

V práci nie je nabíjací obvod realizovaný.

2.2.2 Krytie

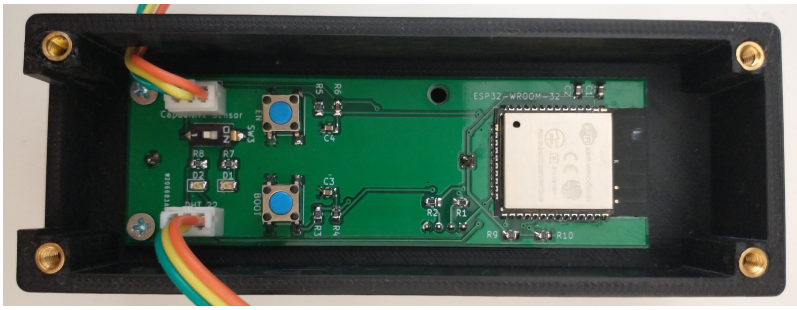
Pri návrhu sa predpokladalo s prenosom zariadenia, a taktiež s jeho umiestnením do vonkajšieho prostredia. Pri kombinácii s inými systémami, ako napríklad s automatizovaným zavlažovaním, je vhodné, aby bolo zariadenie dostatočne kryté. Podľa platnej normy pre stupne ochrany elektronického zariadenia[6] je vhodné použiť pre stupeň krytia pred vniknutím cudzích predmetov stupeň krytia IP 6x, teda krytie pred úplným vniknutím prachu. V prípade ochrany pred vniknutím vody je vhodný minimálny stupeň ochrany IP x6, teda ochrana pred tryskajúcou vodou. Tento stupeň môže byť nižší v závislosti na použití zariadenia vo väčšom systéme.

Bežne predávaný kapacitný senzor, použitý v navrhnutom zariadení, má nevýhodu spočívajúcu v odkrytom riadiacom obvode. Tento obvod je potrebné zaizolovať pred vlhkosťou, aby nemohol byť vlhkosťou poškodený a predĺžila sa jeho životnosť. [30] udáva niekoľko bežných spôsobov izolácie. Jedným so spôsobov je obaliť riadiaci obvod senzora vrstvou epoxidu a polyesteru. Toto riešenie síce z časti môže znížiť senzitivitu senzora, ale zabezpečí ho proti poškodeniu vlhkosťou. Ďalšou možnosťou je použitie gumového náteru alebo gumového spreja. Taktiež je možné na izoláciu použiť silikónový tmel.

Pre PCB s ESP32 sa ponúka možnosť použitia 3D tlače a vytvorenia krabičky, ktorá by plnila funkciu krytia. Rovnako aj pre kapacitný senzor bola vytvorená krabička, ktorá v kombinácii s izoláciou epoxidom a silikónom môže zlepšiť izoláciu zariadenia pred vlhkosťou.

Rovnako ako kapacitný senzor, aj pri DHT22 je odporúčané nevystavovať senzor prílišnej vlhkosti. V jeho dokumentácii je zahrnuté hmlisté prostredie ako rizikový faktor, ktorý môže spôsobiť poškodenie riadiacich obvodov a značne tak znížiť životnosť senzora. V prípade umiestnenia do vonkajšieho prostredia je teda potrebné aj tu realizovať dodatočnú izoláciu, aby sa predišlo poškodeniu senzora. V práci krytie pre DHT22 zabezpečuje krabička vytvorená 3D tlačou s otvormi obalenými v jemnej kovovej tkanine. Toto riešenie z časti znižuje citlivosť senzora. Spôsobená chyba merania nemá ale na výsledok veľký vplyv.

V práci je krytie zariadenia riešené kombináciou 3D tlače a silikónovej izolácie. Kapacitný senzor vlhkosti je obalený silikónom a je pre neho navrhnutá krabička, spoje krabičky sú zakryté vrstvou epoxidového lepidla. Na prepojenie senzora a mikrokontroléra je využitý UTP kábel pre vonkajšie použitie s dodatočnou izoláciou. Na pripojenie senzora k doske budú použité jeho tri vodiče.



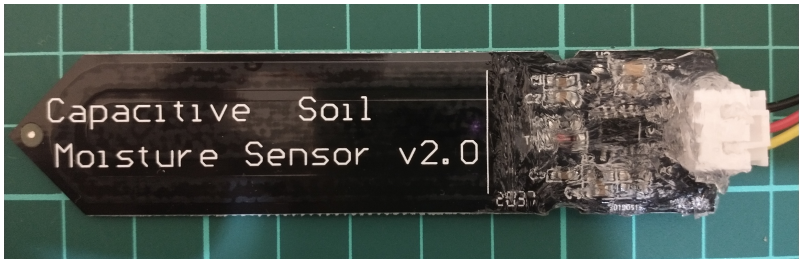
Obr. 2.3: Krytie pre ESP32

Pre DHT22 je vytvorené puzdro, ale keďže ide o senzor merajúci teplotu prostredia a vlhkosť vzduchu, nemôže byť od prostredia úplne izolovaný. Jeho ochrana je teda prvotne zabezpečená umiestnením v prostredí tak, aby sa zamedzilo prístupu tečúcej alebo striekajúcej vody k elektrickým obvodom. Druhotným zabezpečením je zakrytie otvorov jemnou kovovou tkaninou.

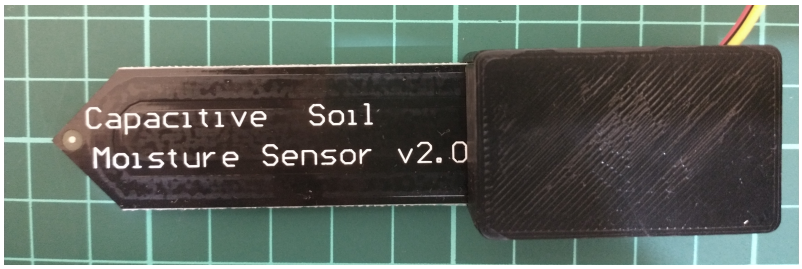
Keďže je puzdro v priestore umiestnené pod krabičkou pre PCB s ESP32 a je prichytené k jej spodnej časti, krabička na riadiacu PCB v tomto prípade slúži ako spôsob tienenia.

Samotný obal pre PCB je vytvorený 3D tlačou v spojení s dodatočnou izoláciou epoxidom, hlavne pre výstup napájania kapacitného senzoru a na spojoch s puzdrom pre DHT22.

Navrhnuté puzdrá pre 3D tlač sú súčasťou prílohy. Zariadenie a krabička sa nachádza na obrázkoch 2.3. Na obrázku 2.4 je výsledná izolácia kapacitného senzora, na obrázku 2.5 je výsledné krytie kapacitného senzora.



Obr. 2.4: Zaizolovaný kapacitný senzor



Obr. 2.5: Krytie pre kapacitný senzor

3 Softwarové riešenie

Sofwarové riešenie je postavené na knižnici pre ESP32, ktorej jadro tvorí vývojové prostredie pre Arduino. Je využitých niekoľko knižníc, ich funkcia je popísaná v tabuľke 3.1.

Knižnica	Úloha
WiFi.h	Správa sieťového rozhrania, riadenie komunikácie
WiFiClientSecure.h	Klienta pre zabezpečenú TLS komunikáciu
PubSubClient.h	MQTT protokol
WiFiManager.h	Správa hesiel a SSID na opätovné pripojenie k sieti
DHT.h	riadenie DHT22
ArduinoJson.h	Knižnica pre prácu s formátom JSON
esp32fota.h	Dedí z OTAUpdate.h, zabezpečuje OTA aktualizácie

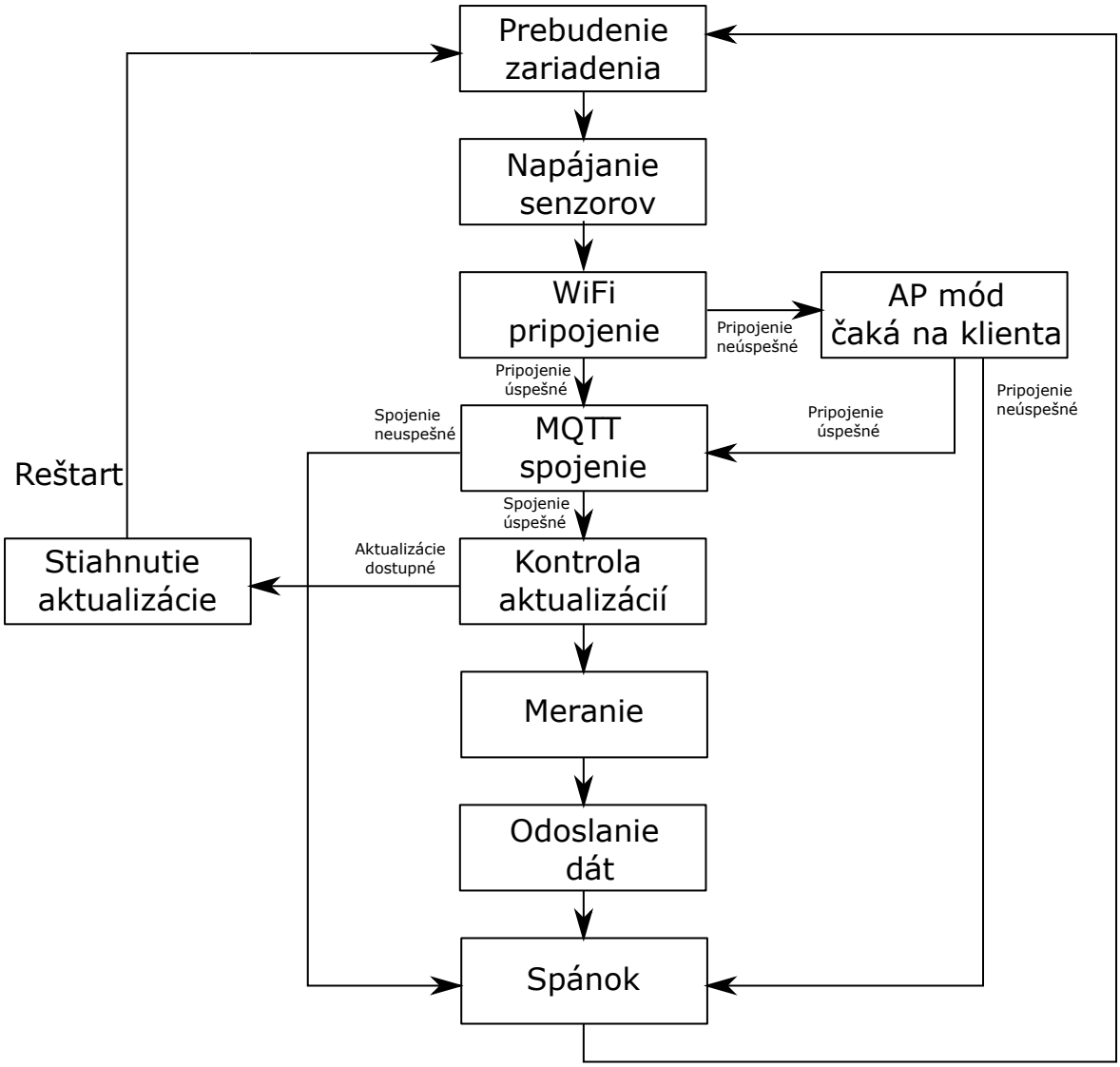
Tab. 3.1: Použité knižnice

3.1 Základný koncept

Základný koncept softwarového riešenia senzora je zhrnuté na schéme na obrázku 3.1.

Ide o princíp jednoduchého stavového automatu. Keďže oba použité senzory sú veľmi pomalé a dosahujú uspokojivé výsledky až po určitom čase od spustenia, je ich napájanie spustené pred sieťovým rozhraní. Zariadenie v tejto fáze čaká 20s, následné prebudí WiFi rozhranie. Prebehne pokus o pripojenie do siete, po úspešnom pripojení prebehne mechanizmus vyhľadania možných aktualizácií firmwaru.

Ďalším krokom je naviazanie spojenia s MQTT brokerom, nameranie potrebných dát a ich odoslanie na MQTT broker. Zariadenie nečaká na potvrdenie o doručení packetov na broker, kontroluje iba ich samotné odoslanie z vlastného rozhrania. Po odoslaní je prerušené napájanie senzorov, je prevedené odpojenie od MQTT brokeru, vypnuté rozhranie pre WiFi a zariadenie prechádza do režimu deep-sleep. V režime deep-sleep ostáva po dobu 10 minút, následne tento proces opakuje pre nové meranie.



Obr. 3.1: Základný koncept firmwaru

3.2 Riadenie senzorov

Z dôvodu čo najefektívnejšieho využitia batériového napájania sú senzory napájané vysokou hladinou signálu na pinoch s digitálnym výstupom, presnejšie z GPIO33 pre kapacitný senzor a GPIO25 pre senzor DHT22.

V prvom kroku sa funkciou pinMode(GPIO, OUTPUT) zadefinuje dané GPIO ako výstup. Systém riadenia napájania následne prevedie príkazom digitalWrite(pin, HIGH) hladinu výstupu definovaného pinu na vysokú hladinu, teda na 3,3 V.

V poslednom kroku sa prevádza volanie funkcie dht.begin(), ktorá je súčasťou knižnice DHT.h. Tá riadi odčítanie hodnoty z digitálneho výstupu senzoru. Jej súčasťou je prispôsobenie nastavenej frekvencii procesoru na ESP32, z čoho vyplýva, že túto hodnotu nie je potrebné definovať pri volaní funkcie.

Pre analógový kapacitný senzor sa volá funkcia analogRead(GPIO), ktorá odčítava hodnotu z ADC prevodníku.

Pre odčítanie hodnôt z kapacitného senzora je použité prepočítanie odčítanej hodnoty z ADC prevodníku na základe funkcie map(). Jej definícia je zložená s piatich hodnôt, a to odčítaná hodnota za pomoci analogRead(), maximálna možná dosiahnuteľná hodnota, teda hodnota nameraná pri kalibrácii pre senzor vo vzduchu(air_value), minimálna možná dosiahnuteľná hodnota, teda hodnota nameraná po ponorení senzora do vody(water_value) a hodnoty, ktorým sú tieto dve kalibračné konštanty priradené, teda 100 pre water_value a 0 pre air_value. Funkcia vracia celočíselnú premennú typu int, ktorá je priradená odčítanej hodnote na základe uvedených maximálnych a minimálnych hodnôt. S prihliadnutím na to, že ADC prevodník odčítava celočíselné hodnoty hladín a na kalibráciu môžeme takto získaný výsledok pokladať za dostatočne presný.

Pre odčítanie hodnôt z DHT22 je použitá knižnica DHT.h. Pre hodnotu teploty sa volá funkcia readTemperature(), pre relatívnu vlhkosť vzduchu funkcia readHumidity().

3.3 Pripojenie k sieti

ESP32 môže fungovať v troch sieťových režimoch. Je nimi režim modemu (WIFI_AP), kedy ESP32 zastupuje v sieti rolu vysielača, režim stanice (WIFI_STA) a režim, v ktorom je stanicou aj modemom zároveň (WIFI_STA+AP). Špeciálnym prípadom je stav WIFI_STOP, kedy je WiFi rozhranie vypnuté.

O pripojenie senzoru k sieti sa stará knižnica WiFiManager.h. Keďže ESP32 v základnom nastavení funguje v móde STA+AP, teda ako stanica aj prístupový bod zároveň, nastaví sa v prvom kroku explicitne do režimu stanice, a to volaním funkcie WiFi.mode(WIFI_STA), ktorá je súčasťou Arduino jadra pre mikrokontroléry ESP32. Následne je volaný WiFimanager a sú definované niektoré jeho funkcie. Medzi základné patrí nastavenie timeout pre prípad, že sa nepodarí pripojiť k uloženému AP z RTC pamäte volaním funkcie setConnectTimeout(čas).

Čas pre tento timeout je stanovený na 60 sekúnd. ESP32 sa na základe uložených prihlasovacích údajov pokúsi pripojiť ku AP, ak je spojenie úspešné pokračuje do ďalšieho kroku.

Ak sa pripojenie nepodarí uskutočniť po uplynutí doby jednej minúty, uložené prihlasovacie údaje sú z RTC pamäte vymazané a zariadenie prechádza do režimu spánku.

Po nasledujúcom prebudení sa znova pokúsi pripojiť, keďže ale nemá v pamäti žiadne údaje pre pripojenie, volá funkciu autoConnect(), ktorá je súčasťou knižnice WiFiManager.h. Tá najskôr prevedie prevedenie ESP32 do AP módu, následne

spustí nastavený časovač pre timeout a čaká na pripojenie klienta. Súčasťou tejto funkcie je tzv. „captive“ portál

Ak sa pripojenie nepodarí uskutočniť po uplynutí doby jednej minúty, uložené prihlasovacie údaje sú z RTC pamäte vymazané a zariadenie prechádza do režimu spánku.

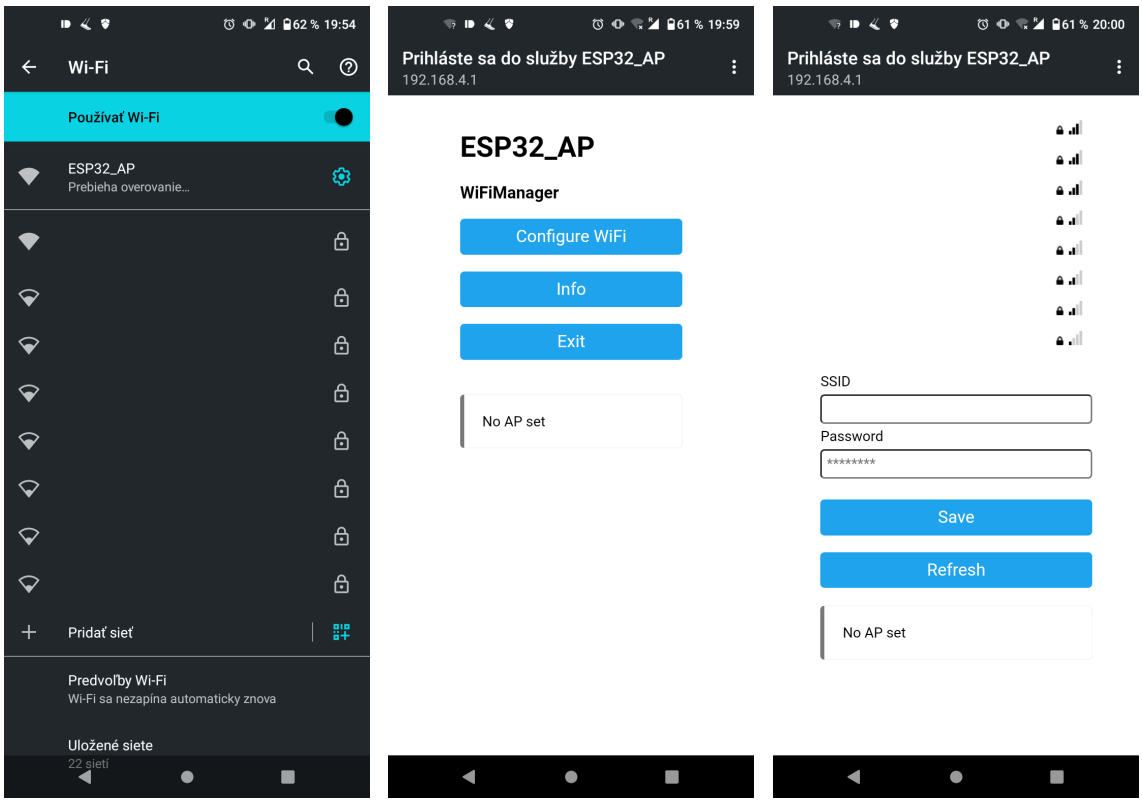
Výpis 3.1: Nastavenie pre WiFiManager

```
void setup() {
...

    WiFiManager wifiManager;
    /*Pokúsi sa o pripjenie s uloženými údajmi*/
    wifiManager.setConnectTimeout(60);

    /*Dark mód*/
    wifiManager.setClass("invert");
    /*Nastaví timeout pre konfiguračný portál*/
    wifiManager.setConfigPortalTimeout(180);

    /*Kontroluje, či sa k nemu pripojilo
    zariadenie klienta*/
    wifiManager.setAPClientCheck(true);
    /*Nastaví minimálnu silu signálu,
    ktorá bude v ponuke medzi prístupnými AP*/
    wifiManager.setMinimumSignalQuality(20);
    /*Po pridaní prihlasovacích údajov
    automaticky odpojí klienta*/
    wifiManager.setBreakAfterConfig(true);

    if(!wifiManager.autoConnect("ESP32_AP", "grimaska")){
        Serial.println("Failed to connect. Timeout.");
        delay(1000);
        /*V prípade nenaviazania spojenia sa uspí*/
        sleep();
}
```

Po nasledujúcom prebudení sa znova pokúsi pripojiť, keďže ale nemá v pamäti žiadne údaje pre pripojenie, volá funkciu autoConnect(), ktorá je súčasťou knižnice WiFiManager.h. Tá najskôr prevedie prevedenie ESP32 do AP módu, následne spustí nastavený časovač pre timeout a čaká na pripojenie klienta. Súčasťou tejto funkcie je tzv. „captive“ portál.

Realizované riešenie využíva vzhľad konfiguračného portálu preddefinovaný v kniž-

nici WiFiManager a nijak ho neupravuje. Konfiguračný portál sa nachádza na obrázku 3.2.



Obr. 3.2: Spracovanie dát na rozhraní Node-RED

Po pripojení klienta na ESP32 v móde AP je klient automaticky presmerovaný na konfiguračný portál. Po zadaní prihlasovacích údajov, SSID a hesla danej sieť, ku ktorej sa má ESP32 pripojiť, si zariadenie zadané údaje uloží do RTC pamäte a prevedie reštart. Klient je z konfiguračného portálu automaticky odpojený. Zariadenie po reštarte prejde rovnakým procesom ako pri prebudení z deep-sleep módu.

V prípade, že sa k zariadeniu v režime AP po dobu uplynutia nastaveného časovaču nepripojí žiaden klient, prechádza zariadenie do režimu spánku. Po prebudení pokus o získanie prihlasovacích údajov opakuje.

V prípade neúspešného pripojenia na nakonfigurované SSID sa zariadenie uspí volaním funkcie sleep() 3.4. Pri nasledujúcom prebudení sa proces WiFiManageru zopakuje.

3.4 Odosielanie a prijímanie nameraných dát

3.4.1 Prijímanie na strane MQTT brokera

Pre MQTT protokol je potrebné definovať tzv. MQTT broker, teda rozhranie, ktoré spracováva publish (vydané) a subscribe (odoberané) požiadavky pre jednotlivé témy. Pre navrhnutý systém bol vybraný všeobecne rozšírený MQTT broker Mosquitto spravovaný spoločnosťou Oracle.

Broker je vytvorený na Raspberry Pi s pevne definovanou IP adresou, ktorá je oznámená jednotlivým klientom. Je využitý port 8883. Príklad základnej konfigurácie zabezpečeného Mosquitto brokeru je nasledovný.

Výpis 3.2: Konfigurácia MQTT brokeru

```
persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

allow_anonymous false
password_file /etc/mosquitto/pwfile

listener 8883 0.0.0.0
cafile /etc/mosquitto/certs/mosquitto-ca.crt
keyfile /etc/mosquitto/certs/broker.key
certfile /etc/mosquitto/certs/broker.crt
tls_version tlsv1.2

log_type error
log_type notice
log_type warning

listener 1883 0.0.0.0
```

Komunikácia je šifrovaná, používa TLSv1.2. Broker má vygenerovaný certifikát s vlastným podpisom, pre navrhnutý systém ale tento certifikát nie je zo strany klientov pri komunikácii kontrolovaný. To síce prináša riziko útoku man-in-the-middle, systém ale funguje plne na lokálnej sieti.

Ďalšou formou zabezpečenie je zakázanie pripojenia anonymných klientov. Klient sa teda pri pripojení musí preukázať svojim menom a heslom, uloženým v šifrovanom súbore hesiel, ktoré broker pozná. V prípade, že je meno a heslo v zozname, klient

môže pokračovať požiadavkou na publish alebo subscribe. V opačnom prípade je spojenie s klientom prerušené.

3.4.2 Odosielanie dát zo strany klienta

V navrhnutom systéme hrajú rolu dvaja klienti. Prvým z nich je samotné ESP32, to s brokerom komunikuje na základe knižnice PubSubClient.h a knižnice WiFiClientSecure.h.

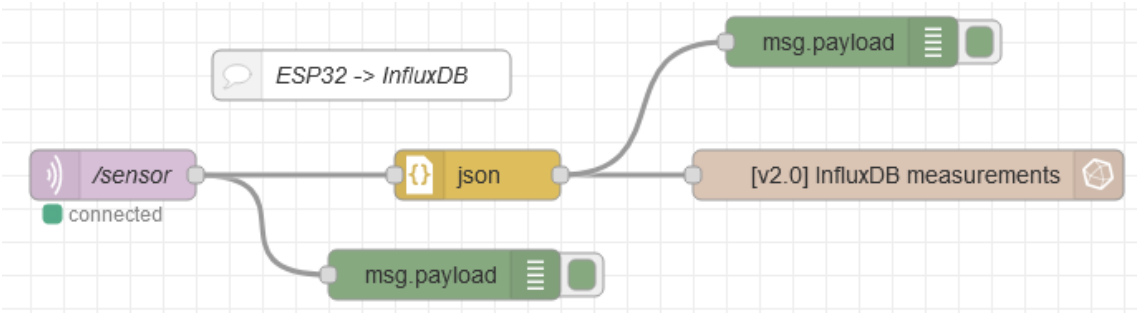
Na základe objektu WiFiClientSecure je vytvorený klient esp32_client. Ten je následne volaný pri vytvorení PubSub klienta objektom PubSubClient client() s parametrami adresa brokeru, port a samotný WiFi klient.

WiFiClientSecure zabezpečuje, že komunikácia medzi brokerom a mikrokontrolérom bude šifrovaná protokolom TLSv1.2.

Pre úplne zabezpečenie spojenia je potrebné kontrolovať pri spojení s brokerom jeho certifikát. Vzhľadom na to, že pre lokálne riadenú sieť je obtiažne získať CA certifikát a doba platnosti certifikátu s vlastným podpisom je pomerne krátka, volá esp32_client pred pripojením funkciu setInsecure(). Tá vynechá krok kontroly platnosti certifikátu serveru, stále ale šifruje komunikáciu a komunikuje so zabezpečeným portom 8883.

Druhým klientom pripojeným na broker je rozhranie Node-RED, ktoré zabezpečuje riadenie a prenos dát z brokeru do databázy. Aj táto komunikácia je zabezpečená na základe certifikátov s vlastným podpisom. Je potrebné kontrolovať ich platnosť. Po uplynutí doby ich platnosti sa preruší spojenie medzi brokerom a databázou.

Na obrázku 3.3 je funkcia rozhrania Node-RED riadiaca posielanie nameraných dát do databázy.



Obr. 3.3: Spracovanie dát na rozhraní Node-RED

ESP32 v prvom kroku uskutoční spojenie s brokerom volaním funkcie reconnect() pod podmienkou, že neexistuje spojenie medzi brokerom a mikrokontrolérom.

Funkcia reconnect() volá funkciu client.connect() s parametrami ID klienta a jeho používateľského mena a hesla uloženého v zozname, ktorý má k dispozícii broker. V prípade neúspešného naviazania spojenia sa pokúsi o jeho znovu vytvorenie po uplynutí doby 5 sekúnd.

Možnosť upozornenia užívateľa na nevytvorené spojenie práca nerieši, možnosťou by bolo odoslanie upozornenia na mail, Node-RED alebo podobné rozhranie, keďže zariadenie už v bode vytvárania spojenia s brokerom disponuje sieťovým pripojením.

V prípade úspešného spojenia zariadenie pokračuje na odčítanie dát zo senzorov. Tie následne vyformátuje do správy v JSON formáte a volá funkciu client.publish(), ktorej parametre sú téma, do ktorej bude správa vydaná a premenná output, ktorá obsahuje samotnú správu. Funkcia publish() vracia hodnotu typu bool, kontroluje sa teda, či je vrátená hodnota TRUE, teda správa úspešne odišla z rozhrania ESP32.

Keďže knižnica PubSubClient v aktuálnom stave podporuje pre správy typu publish len QoS 0, nie je možné riešiť potvrdzovanie prijatia správ brokerom. Zariadenie posiela 5 po sebe idúcich správ z nameranými hodnotami, a to z dôvodu získania nameraných údajov aj v prípade, že niektorý z posielaných packetov bude poškodený alebo stratený, a zároveň ako kontrolu stability nameraných hodnôt.

Výpis 3.3: Formátovanie a odosielanie správy

```
char output[100];
StaticJsonDocument <200> doc;
  doc["version"] = firmware_version;
  doc["soil_humidity"] = humidity;
  doc["air_humidity"] = dht.readHumidity();
  doc["temperature"] = dht.readTemperature();
  doc["battery"] = battery;
 serializeJson(doc, Serial);
 Serial.println();
 size_t n = serializeJson(doc, output);


 if(client.publish("/sensor", output, n) == true){
    Serial.println("Message sent succesfully.");
 }else{
   Serial.println("Error while sending message.");
 }
 delay(100);
}
```

3.5 Napájanie

S prihliadnutí na batériové napájanie a zabezpečenie čo najdlhšieho fungovania zariadenia je potrebné minimalizáciu spotreby riešiť aj na softwarovej úrovni. Zariadenie na toto využíva prechod do režimu hlbokého spánku, teda deep-sleep mód.

Platforma ESP32 ponúka aj možnosť tzv. hibernácie zariadenia, v tomto režime sú ale odpojené všetky periférie a rozhrania mikrokontroléru, okrem RTC časovača, ktorý sa stará o kontrolu času prebudenia. V prípade použitia hibernácie je odpojená aj RTC pamäť zariadenia. Po prebudení by teda zariadenie muselo byť stále nanovo pripojované ku AP užívateľom, zadaním SSID a heslom siete, keďže prihlasovacie údaje nie sú zadané ako pevná premenná.

Pevné zadanie prihlasovacích údajov by sťažilo možnosť prenosu zariadenia do inej lokálnej siete, keďže po jeho prenose by bolo nutné aktuálne prihlasovacie údaje nahrávať do zariadenia spolu s firmwarom, a takto zabezpečiť možnosť pripojenia.

Deep-sleep mód je definovaný volaním funkcie sleep(). Tá je povolaná na konci, po odoslaní správ na MQTT broker a slúži na prípravu zariadenia pre prechod do režimu spánku. Pri uspávaní zariadenia je potrebné definovať dve konštanty, a to TIME_TO_SLEEP, predstavujúcu čas, počas ktorého bude zariadenie v režime spánku a uS_TO_M_FACTOR, ktorý slúži ako konštanta na prepočet času s mikrosekúnd na minúty spánku.

Výpis 3.4: Prechod zariadenia do režimu spánku

```
void sleep(){                                                    1
  client.disconnect();  /*Odpojí sa od MQTT brokeru*/            2
  WiFi.disconnect(true);  /*Odpojí WiFi*/                        3
  WiFi.mode(WIFI_OFF);  /*Vypne WiFi rozhranie*/                 4
  esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP *                  5
                                uS_TO_M_FACTOR);                 6
  Serial.println("Going␣to␣sleep␣now.");                        7
  delay(1000);                                                   8
  Serial.flush(); /*debugging*/                                  9
  esp_deep_sleep_start();                                        10
}                                                                11
```

Funkcia client.disconnect() zabezpečí bezpečné odpojenie klienta od MQTT brokeru, WiFi.disconnect() s parametrom true a WiFi.mode s parametrom WIFI_OFF vypnú a odpoja WiFi periférium. Volá sa funkcia esp_sleep_enable_timer_wakeup(čas) s časovým parametrom predstavujúcim dobu trvania deep-sleep režimu. Nakoniec je volaná funkcia esp_deep_sleep_start(), ktorá realizuje zapnutie deep-sleep režimu na stanovenú dobu.

Pre kontrolu napájania je vytvorená funkcia čítajúca hodnotu napätia batérie

pripojenej na ADC prevodník. Pre túto funkciu je využitý ADC pin GPIO32 a funkcia analogRead(pin). Vzhľadom na predradný delič pripojený na použitý pin je potrebné previesť kalibráciu nameranej hodnoty napätia podľa kapitoly 2.2.1.

3.6 Aktualizácia

Pri navrhnutom zariadení sa ráta s jeho umiestnením do vonkajšieho prostredia a čo najmenšou potrebou jeho otvárania a prevádzania úprav a údržby. Tomuto musí byť prispôsobený aj systém distribúcie aktualizácií.

Zariadenie plne využíva možnosť automatických OTA aktualizácií. Systém aktualizácii je založený na princípe vyhľadania dostupnej aktualizácie, jej následného stiahnutia a uloženia do pamäte, reštartu zariadenia a pokusu načítania novej verzie firmwaru.

Po úspešnom pripojení do lokálnej siete zariadenie najprv požiada na základe HTTP GET žiadosti distribučný server alebo cloudové úložisko o JSON súbor obsahujúci informácie o najnovšej dostupnej verzii firmwaru a adresu úložiska, kde je možné tento firmware získať. Zo získaného súboru vyberie údaj o verzii firmwaru, jeho hodnotu porovná s hodnotou definovanou pre aktuálne bežiaci firmware. Ak je novozískaná hodnota nižšia alebo rovná pokračuje ďalej bez aktualizácie.

Výpis 3.5: Príklad JSON súboru s informáciami o aktualizácii

```
{
  "type": "esp32-sensor",
  "version": 6,
  "host": "192.168.17.241",
  "port": 8080,
  "bin": "/esp32_v4.ino-esp32.bin"
}
```

Ak je hodnota aktuálne načítaného firmwaru nižšia, pošle GET požiadavku na adresu uvedenú v JSON súbore. Po získaní odozvy porovná, či hlavička odpovede obsahuje kód 200, teda kladnú odpoveď, následne porovná, či typ obsahu doručeného v packete zodpovedá typu application/octet-stream, teda doručený packet obsahuje spustiteľný súbor s príponou .bin.

Tento súbor následne uloží do oddielu pamäte určeného pre ukladanie súborov získaných z OTA aktualizácií a po reštarte sa pokúsi sa o jeho načítanie. Po úspešnom načítaní sa spustí aktualizovaný firmware. Po neúšpešnom načítaní prevedie downgrade predchádzajúcu verziu a pokračuje v predchádzajúcej verzii. V aktuálnej verzii nie je upozornenie užívateľa na nefunkčný firmware riešené.

Pre server, kde sú uložené aktualizácie sa ponúkajú dve možnosť, a to lokálny

http server, alebo vzdialené cloudové úložisko. V práci je použité riešenie na základe lokálneho http serveru s certifikátom s vlastným podpisom, načúvajúcom na porte 8080. Možnosťou na distribúciu firmwaru by bolo využitie platformy Github. V tomto prípade by bola zabezpečená šifrovaná komunikácia a zároveň dostupnosť pre pridávanie nových aktualizácií bez závislosti na prístupe na lokálnu sieť.

Pre zabezpečenie správneho fungovania systému aktualizácií je potrebné dodržovať niekoľko zásad. Najdôležitejšou je zabezpečiť, aby každá nová verzia firmwaru definovala použitie OTA aktualizácií, keďže po načítaní spustiteľného súboru bez tejto definície nie je možné naďalej túto funkciu využívať.

Druhu dôležitou zásadou je uvádzanie správnej a zhodnej verzie firmwaru v JSON súbore aj v distribuovanom .bin súbore. Nesprávne uvedenie týchto hodnôt by mohlo viesť k nestiahnutiu aktualizovanej verzie firmware alebo naopak k opakovanému sťahovaniu tej istej verzie.

4 Užívateľské rozhranie

4.1 Spracovanie nameraných dát

Webové rozhranie zariadenia využíva často používaný prístup pomocou Node-RED, ako základ pre fungovanie, nastavenie, úpravu a správu, InfluxDB na ukladanie dát a nástroj Grafana pre ich spracovanie a vizualizáciu pre užívateľa. Webový server je zriadený lokálne na RaspberryPi.

Prípadná možnosť pripojenia k webovému rozhraniu mimo lokálnu sieť nie je v návrhu riešená. Prepojenie medzi nameranými dátami a užívateľom mimo lokálnu sieť je riešené botom pre platformu Telegram a notifikáciami pomocou aplikácie Pushbullet.

Ako už bolo v predchádzajúcich kapitolách povedané, prepojenie medzi senzormi a webovým rozhraním je zriadený pomocou MQTT protokolu s TLS šifrovaním. O prepojenie samotného užívateľa a zdroja dát sa stará platforma Node-RED.

V tabuľke 4.1 sa nachádza prehľad použitých nástrojov, ich verzia a funkcia.

Nástroj	Verzia	Funkcia
Node-RED	v1.3.4	riadenie komunikácie medzi platformami
Mosquitto	v2.0	riadenie MQTT komunikácie
InfluxDB	v2.0	databáza
Grafana	v7.5.7	grafické rozhranie a zobrazenie nameraných údajov
Pushbullet	v362	Upozornenia posielané na mobilný telefón

Tab. 4.1: Použité nástroje

4.2 Grafické rozhranie

O grafické rozhranie sa stará nástroj Grafana. Je dostupná z lokálnej siete na IP adrese RaspberryPi na porte 3000.

Grafana berie zobrazované dáta priamo z InfluxDB a zobrazuje ich podľa požiadavky užívateľa v reálnom čase. Na jej rozhraní je možné nastaviť časový interval pre zobrazované dáta od posledných 5 minút až po dva dni. Taktiež je možné zadefinovať vlastný časový úsek pre zobrazovanie.

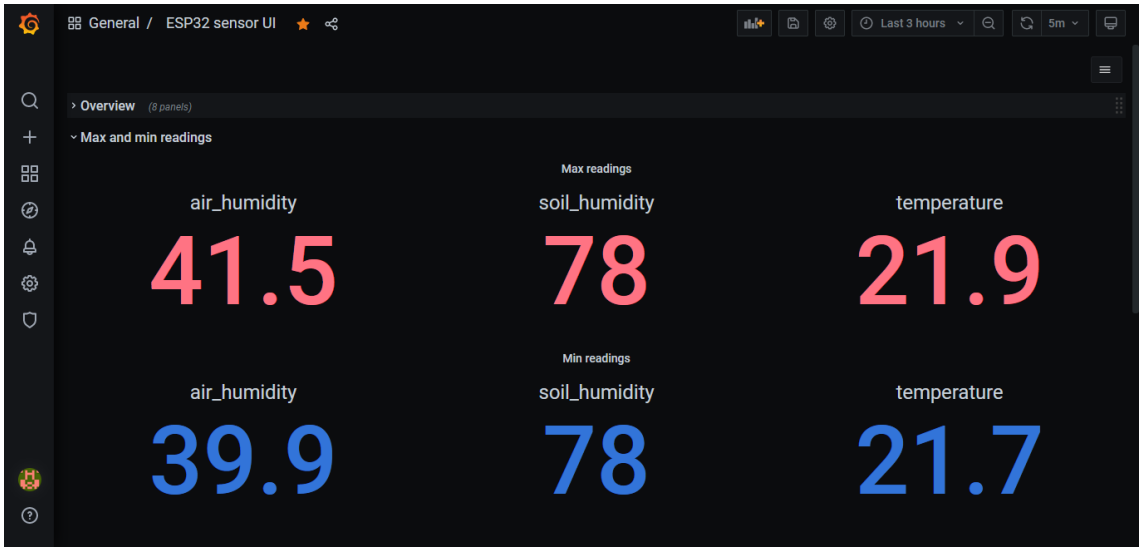
Jednou z ponúkaných možností je aj nastavenie upozornení na užívateľom definované stavy. Takto je možné upozorniť užívateľa napríklad na nízku vlhkosť pôdy, alebo neobvyklé hodnoty merania.

Príklad grafického prostredia pre senzor realizovaný v práci sa nachádza na obrázku 4.1. Rozhranie zahŕňa grafy pre zobrazenie aktuálnej teploty, vlhkosti pôdy

a vzduchu a ich časový priebeh. Zobrazuje aktuálny stav batérie, verziu firmwaru nahraného na ESP32 a upozornenia na neobvyklé stavy.

Obr. 4.1: Grafické rozhranie

Obrázok 4.2 ukazuje rozšírené užívateľské rozhranie. To vypisuje maximálne a minimálne namerané hodnoty pre zadefinovaný časový interval.



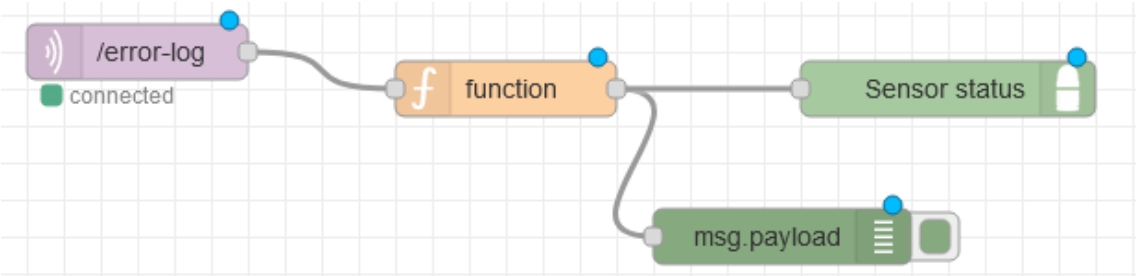
Obr. 4.2: Rozšírené grafické rozhranie

4.3 Komunikácia s užívateľom

Priama komunikácia medzi databázou a užívateľom je zabezpečená pomocou aplikácie Pushbullet, kde je trikrát denne odoslaná správa o meraniach vo forme notifiká-

cie. Rovnako sú na túto platformu preposielané kritické správy, napríklad nízky stav batérie alebo neschopnosť vytvorenia spojenia s MQTT brokerom. V prípade dodatočnej implementácie upozornení na poškodený firmware pre aktualizáciu systému by mohla byť aplikácia využitá aj pre hlásenie tejto chyby.

V práci je takýmto spôsobom už ošetrený prípad neobvyklých hodnôt nameraných senzorom, prípadne ich nesprávneho fungovania. Príkaz pre správu týchto upozornení je na obrázku 4.3.



Obr. 4.3: Príkaz pre zasielanie upozornenia na chybu senzoru

Prijatá správa sa prevedie z JSON Stringu na obyčajný String(). Ten sa následne odošle ako upozornenie na mobilný telefón používateľa.

Aj keď je v súčasnej dobe je v Česku a na Slovensku najvyužívanejšou platformou na komunikáciu Messenger, Facebook pre vytváranie botov na komunikáciu kladie dôraz viac na sprístupnenie tejto funkcie pre obchod a služby. Bot pre túto platformu nie je zameraný na preposielanie automatických správ, je vhodný skôr na komunikáciu so zákazníkom, kde jeho základ tvorí databáza často kladených otázok a odpovedí, ktoré by mali týmto spôsobom uľahčiť prvotnú komunikáciu so zákazníkom.

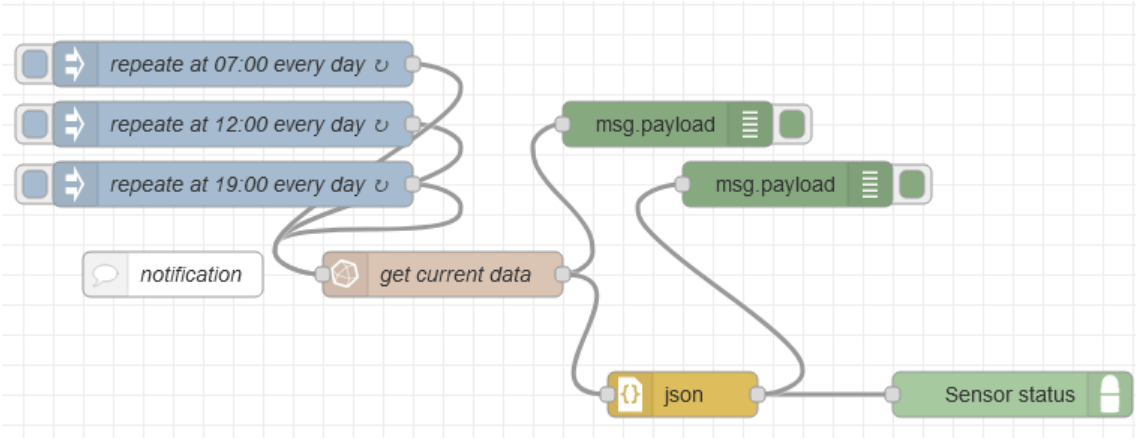
Pre potreby reportovania nameraných dát užívateľovi je vhodnejšie využiť platformu, kde je možné na určitú databázu kontaktov zasielať automatickú správu, bol ako druhá možnosť komunikácie senzoru s užívateľom vybraný bot pre telegram.

Platforma Telegram je prispôsobená na vytváranie jednoduchých botov s rôznymi funkciami, vytvorený bot je následne prepojený s Node-RED. Node-RED v tomto prípade slúži na riadenie akcií vytvoreného bota.

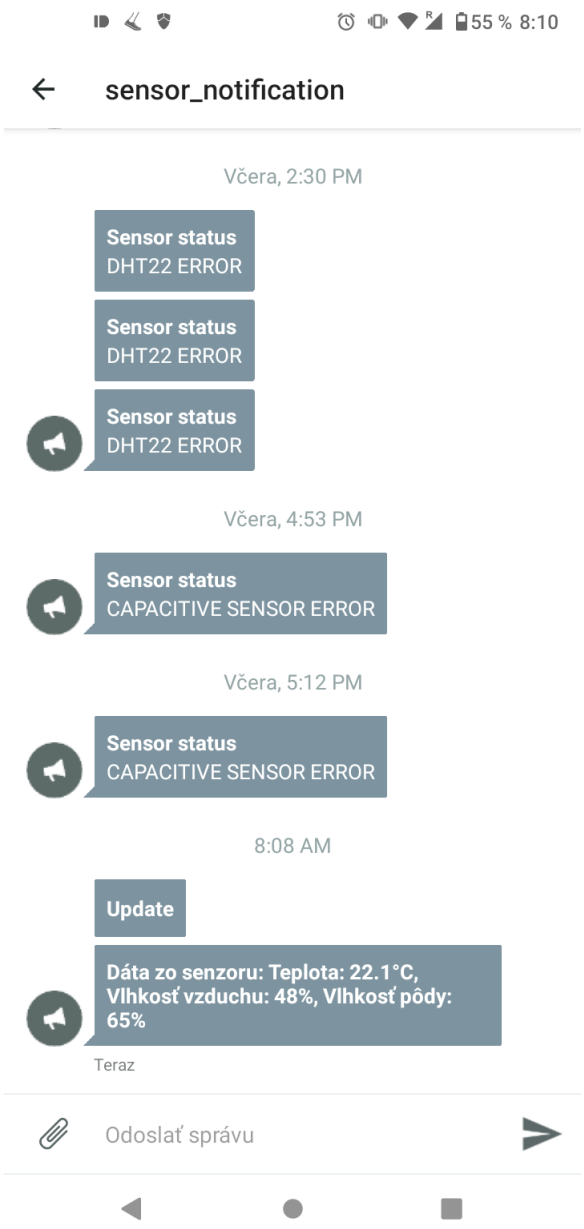
Node-RED najskôr zašle do InfluxDB žiadosť o výpis potrebných údajov. V prípade telegramu ide buď o priemerné hodnoty za daný deň, predchádzajúci deň, alebo o aktuálne hodnoty na základe požiadavky užívateľa. Odpoveď z databázy vo forme javascriptového objektu je prevedená na JSON String. Následne ho preformátuje na správu s požadovanými údajmi a pošle na rozhranie bota, kde si ju užívateľ môže prečítať.

Pri preposielaní notifikácií na platformu Pushbullet je postup rovnaký, ale spúšťačom je automatický časovač nastavený na 7:00, 12:00 a 19:00. Údaje sú v daných

časoch posielané vo forme upozornenia na mobilný telefón užívateľa. Príkaz pre zasielanie upozornení je riadení z Node-RED a je zobrazený na obrázku 4.4.



Obr. 4.4: Príkaz pre zasielanie upozornení v stanovených časoch



Obr. 4.5: Pushbullet upozornenia

Závěr

Bakalárska práca bola zameraná na návrh IoT zariadenia na meranie vlhkosti pôdy, vzduchu a teploty prostredia. Základ návrhu tvorila platforma ESP32.

V prvej časti práce sú zhrnuté teoretické poznatky z oblasti senzorov relatívnej vlhkosti a teploty. Sú spomenuté dva základné typy týchto senzorov. Bližšie je popísaný princíp merania pomocou kapacitného (kapitola 1.1.1) a odporového 1.1.2 senzora. V 1.1.5 je bližšie popísaný spôsob kalibrácie použitého senzora relatívnej vlhkosti pôdy.

Nasleduje predstavenie vybraných vlastností platformy ESP32 (1.2) a hlavne jej verzie ESP32-WROOM-32 použitej v práci.

V prvej časti práce sú taktiež priblížené dve funkcie platformy ESP32 dôležité pre následný návrh zariadenia. Ide o deep a light-sleep mód v 1.2.3 a systém aktualizácií cez OTA v 1.2.4.

Nakoniec sú v 1.3.1 a 1.3.2 predstavené použité komunikačné protokoly a spôsob ich zabezpečenia 1.3.3.

Časti venované samotnému návrhu zariadenia sú rozdelené na návrh hardwaru zariadenia, softwarové riešenie návrhu a návrh webového rozhrania a komunikácie s užívateľom.

Na úvod praktickej časti bakalárskej práce sú popísané vybrané senzory (kapitola 2.1). Kapitola 2.2 sa zaoberá návrhom dosky plošného spoja na základe platformy ESP32. Pri návrhu sa dbá na zníženie spotreby, keďže je výsledné zariadenie napájané z batérie. Riešeniu napájania realizovaného zariadenia je venovaná kapitola 2.2.1. V 2.2.1 je načrtnutá možnosť nabíjacieho obvodu pre navrhnuté zariadenie. Nabíjací obvod nie je v práci realizovaný.

Kapitola 2.2.2 zhrňuje potrebu krytia navrhnutého obvodu, a to najmä z dôvodu umiestnenia zariadenia do vonkajšieho, prípadne pravidelne zavlažovaného prostredia. Na obrázku 2.3 a 2.4 je predstavené konkrétne riešenie izolácie a krytia použitého v práci.

V nasledujúcej časti práce je popísaná softwarová stránka návrhu. Je predstavený základný koncept pre softwarové riešenie návrhu (kapitola 3.1), doplnená o orientačný diagram (obr. 3.1). Ďalej je predstavená problematika napájania riadenia senzorov a ich merania.

Práca ďalej rieši problém pripojenia zariadenia do ľubovoľnej siete (3.3) a odosielanie nameraných dát pomocou protokolu MQTT a jeho zabezpečenie (3.4). V 3.5 je predstavený spôsob implementácie spánkového režimu zariadenia. Poslednou časťou softwarového návrhu je využitie OTA aktualizácií a ich prevedenie zhrnuté v kapitole 3.6.

V poslednej časti (4) bolo predstavené a popísané riešenie webového rozhrania

spojeného so zariadením. Webové rozhranie bolo navrhnuté na platforme RaspberryPi, pomocou rozhrania NodeRed. Na ukladanie dát je využitá databáza InfluxDB určená na prácu s dátami s časovou postupnosťou. Grafické rozhranie pre užívateľa je vytvorené s pomocou nástroja Grafana.

Nakoniec je predstavený spôsob odosielania dát užívateľovi, a to na základe upozornení zasielaných cez platformu Pushbullet.

Pri návrhu zariadenia bol kladený dôraz na prenos a ochranu samotného zariadenia vplyvom prostredia, v ktorom je umiestnené. Návrh zahrňuje zabezpečenie sieťovej komunikácie proti útokom, a z toho vyplývajúce obmedzenie možnosti úpravy nameraných dát alebo aktualizácií firmwaru treťou stranou. V neposlednom rade bolo potrebné vytvorenie prehľadného webového rozhrania s možnosťou ľahkého sledovania nameraných údajov a jednoduchý spôsob komunikácie s užívateľom.

V práci sa predpokladá, že serverové rozhranie na RaspberryPi a senzor sa budú vždy nachádzať na jednej lokálnej sietí. Komunikácia mimo lokálnu sieť nie je v práci rozvinutá.

Literatúra

[1] Adafruit. *DHT11, DHT22 and AM2302 Sensors* [online]. 29. 07. 2012. Posledná zmena: 29. 07. 2012 [cit. 28. 05. 2021]. Dostupné z: <https://learn.adafruit.com/dht>

[2] Adafruit Industries. *Adafruit Unified Sensor Driver*. In: GitHub [online]. Posledná zmena: 29. 06. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/adafruit/Adafruit_Sensor>

[3] Adafruit Industries. *DHT sensor library*. In: GitHub [online]. Posledná zmena: 02. 12. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/adafruit/DHT-sensor-library>

[4] Aosong (Guangzhou) Electronics Co. [Online katalógový list]. *Digital-output relative humidity & temperature senzor/module AM2303*. [cit. 28. 05. 2021]. Dostupné z: <https://www.electroschematics.com/wp-content/uploads/2015/02/DHT22-datasheet.pdf>

[5] BLANCHON, Benoît. *ArduinoJSON: API Reference* [online]. 2014. Posledná zmena: 2021 [cit. 28. 05. 2021]. Dostupné z: <https://arduinojson.org/v6/api/>

[6] ČSN EN 60 529. *Stupně ochrany krytem (krytí - IP kód)*. Federální úřad pro normalizaci a měření, 1993.

[7] CHAOWANAN, J., PREECHA, K., CHANON, F. a WIPA K. An Intelligent Irrigation Scheduling System Using Low-Cost Wireless Sensor Network Toward Sustainable and Precision Agriculture. *IEEEAccess* [online]. 21. 09. 2020. Posledná zmena: 01. 10. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9201441>

[8] Chrisjoyce911 . *esp32FOTA library for Arduino*. In: GitHub [online]. Posledná zmena: 24. 04. 2021. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/chrisjoyce911/esp32FOTA>

[9] DFRobot [Online katalógový list]. *Capacitive Soil Moisture Sensor*. [cit. 05. 12. 2020]. Dostupné z: <https://www.sigmaelectronica.net/wp-content/uploads/2018/04/sen0193-humedad-de-suelos.pdf>

[10] DFRobot [Online katalógový list]. *MoistureSensor*. [cit. 05. 12. 2020]. Dostupné z: <https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0114_Web.pdf>

[11] D-Robotics UK [Online katalógový list]. *DHT11 Humidity Temperature Sensor*. 2017. [cit. 28. 05. 2021]. Dostupné z: <https://datasheetspdf.com/pdf-file/785590/D-Robotics/DHT11/1>

[12] Eclipsera s.r.o. [Online katalógový list]. *Precizní modul BME280 – měření teploty, vlhkosti a barometrického tlaku*. 2017. [cit. 28. 05. 2021]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/91/1469214547.pdf>

[13] EICKEN, T. von. *ESP32 Deep-Sleep Connecting to MQTT* [online]. TvE's Blog, 30. 11. 2018. [cit. 28. 05. 2021]. Dostupné z: <https://blog.voneicken.com/2018/lp-wifi-esp32-mqtt/>

[14] Espressif Systems *Arduino core for the ESP32*. In: GitHub [online]. Posledná zmena: 02. 12. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/espressif/arduino-esp32>

[15] Espressif Systems CO., LTD. ESP-IDF Programming Guide. *Analog to Digital Converter* [online]. 2016. Posledná zmena 2021. [cit. 28. 05. 2021]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/adc.html#adc-calibration>

[16] Espressif Systems CO., LTD. ESP-IDF Programming Guide. *Sleep Modes* [online]. 2016. Posledná zmena 2020. [cit. 28. 05. 2021]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/sleep_modes.html>

[17] Espressif Systems CO., LTD. ESP-IDF Programming Guide. *Over The Air Updates (OTA)* [online]. 2016. Posledná zmena 2020. [cit. 28. 05. 2021]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html>

[18] Espressif Systems CO., LTD [Online katalógový list]. *ESP32-WROOM-32*. [cit. 28. 05. 2021]. Dostupné z: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>

[19] Espressif Systems CO., LTD. *ESP32 Hardware Design Guidelines* [online]. 2021. Posledná zmena 30. 04. 2021. [cit. 28. 05. 2021]. Dostupné z: <https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf>

[20] FRADEN, Jacobs. *Handbook of modern sensors: physics, designs, and applications*. New York: Springer-Verlag, 2004. 589 s. ISBN : 0-387-00750-4

[21] Grafana Labs. Grafana documentation. *Configuration* [online]. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://grafana.com/docs/grafana/latest/administration/configuration/>

[22] InfluxData. *InfluxDB 1.8 documentation* [online]. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://docs.influxdata.com/influxdb/v1.8/#>

[23] Knolleary. *Arduino Client for MQTT*. In: GitHub [online]. Posledná zmena: 20. 05. 2021. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/knolleary/pubsubclient>

[24] Microchip [Online katalógový list]. *Lithium Iron Phosphate (LiFePO4) Battery Charge Management Controller with Input Overvoltage Protection*. 2013. [cit. 28. 05. 2021]. Dostupné z: <https://ww1.microchip.com/downloads/en/DeviceDoc/22191E.pdf>

[25] OASIS. *MQTT Version 3.1.1* [online]. 29. 10. 2014. Posledná zmena: 29. 10. 201 [cit. 28. 05. 2021]. Dostupné z: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>

[26] RADI, MURTININGRUM, NGADISIH, MUZDRIKAH, F. S., NUHA, M. S., RIZQI, F. A. *Calibration of Capacitive Soil Moisture Sensor (SKU:SEN0193)* [online]. 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2018. [cit. 28. 05. 2021]. Dostupné z: <https://www.researchgate.net/publication/329492087_Calibration_of_Capacitive_Soil_Moisture_Sensor_SKUSEN0193>

[27] RFC 2616. *Hypertext Transfer Protocol – HTTP/1.1)*. The Internet Society, 1999. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc2616>

[28] TE Connectivity. *Digital vs. Analog Sensing* [online]. Posledná zmena: 2021. [cit. 28. 05. 2021]. Dostupné z: <https://www.te.com/usa-en/industries/sensor-solutions/insights/digital-over-analog.html>

[29] Tzapu. *WiFiManager*. In: GitHub [online]. Posledná zmena: 24. 04. 2021. [cit. 28. 05. 2021]. Dostupné z: <https://github.com/tzapu/WiFiManager>

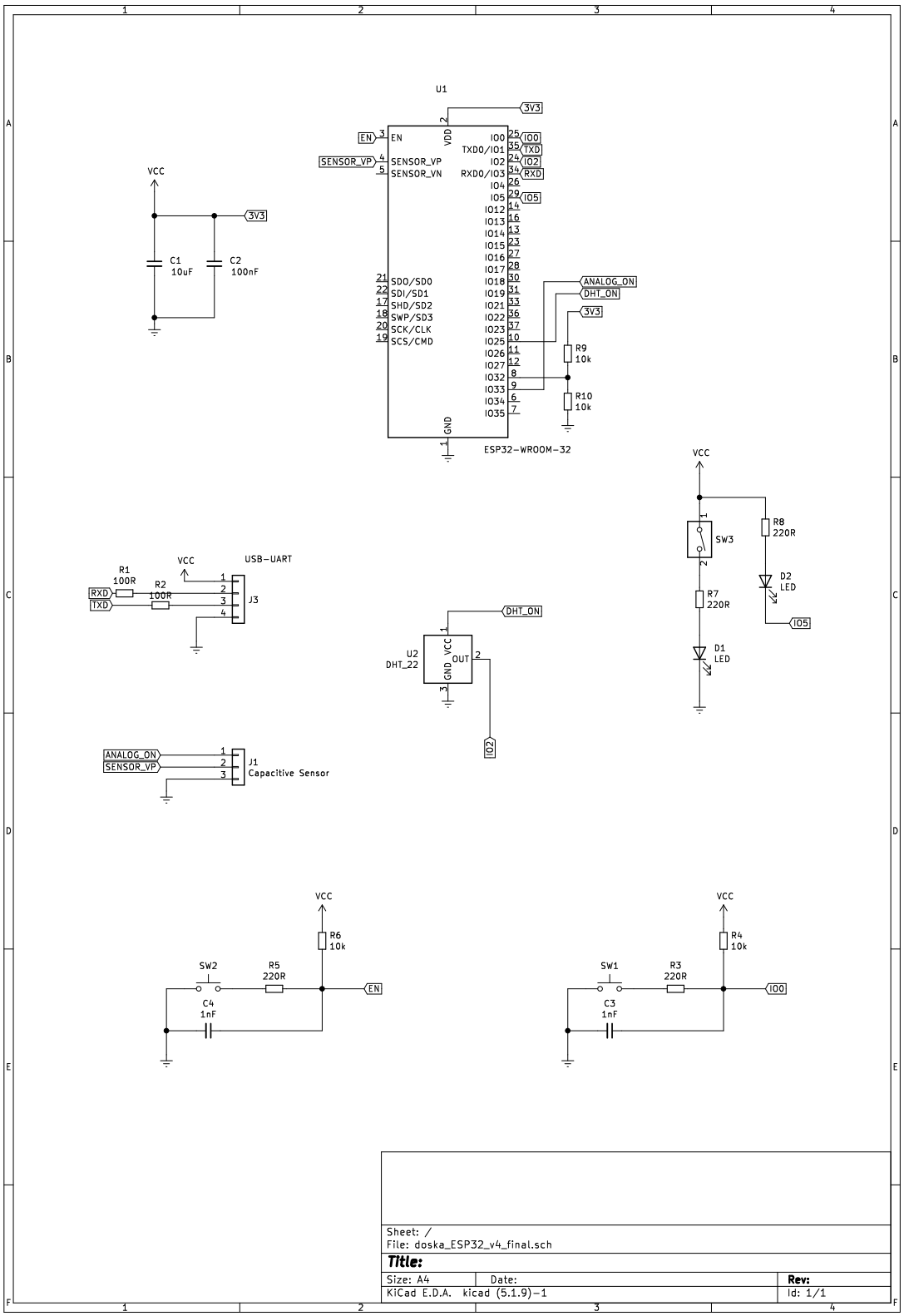
[30] SwitchDoc Labs. *Waterproofing your Capacitive Moisture Sensors* [online]. 16. 07. 2020. [cit. 28. 05. 2021]. Dostupné z: <https://www.switchdoc.com/2020/07/tutorial-waterproofing-capacitive-moisture-sensors/>

Zoznam symbolov a skratiek

RH% relatívna vlhkosť – Relative Humidity
RTD odporový teplotný senzor – Resistance Temperature Detector
NTC Negative Temperature Coefficient
PTC Positiv Temperature Coefficient
RTC hodiny reálneho času – Real Time Clock
ULP Ultra Low Power
BLE Bluetooth Low-Energy
AP prístupový bod – Access Point
ADC analógovo-digitálny prevodník – Analog to Digital Converter
OTA Over the Air
HTTP hypertextový prenosový protokol – Hypertext Transfer Protocol
MQTT Message Queuing Telemetry Transport

Zoznam príloh

A Schéma zapojenia



Obr. A.1: Schéma zapojenia realizovaného zariadenia.

B Elektronická príloha

/
- ESP32_Firmware
  - esp32_firmware.ino ..... Súbor pre Arduino IDE
  - esp32_firmware.ino.esp32.bin ..... Súbor pre OTA aktualizácie
- schéma_zapojenia.pdf
- doska_ESP32 ..... Súbory programu KiCad
  - esp32_gerber ..... Gerber súbory pre výrobu PCB
  - doska_ESP32.kicad_pcb
  - doska_ESP32.sch
- ESP32_senzory_krytie ..... Súbory programu Fusion360
  - Case_main v21.f3d
  - Moisture_sensor_case v6.f3d