

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Tomáš Roško



# 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

## TEPLOTNÍ SENZOR VYUŽÍVAJÍCÍ MODUL (ESPRESSIF) ESP32

TEMPERATURE SENSOR USING (ESPRESSIF) ESP32 MODULE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Tomáš Roško

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2018

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**  
Ústav telekomunikací

**Student:** Tomáš Roško

**ID:** 186804

**Ročník:** 3

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Teplotní senzor využívající modul (Espressif) ESP32

**POKYNY PRO VYPRACOVÁNÍ:**

Navrhněte koncepci bateriově napájeného teplotního senzoru s WiFi modulem ESP32. Senzorová jednotka bude mít navíc univerzální vstup 0-10V a 4-20mA pro připojení libovolného průmyslového čidla. Důraz je kladen na minimální spotřebu senzorové jednotky z důvodu dlouhé životnosti při bateriovém napájení. Senzorová jednotka bude data předávat do vhodného cloudového řešení. Navržené řešení realizujte.

**DOPORUČENÁ LITERATURA:**

[1] MANN, B. C pro mikrokontroléry, BEN, 2003. ISBN 80-7300-077-6

[2] KOSEK, J. PHP tvorba interaktivních internetových aplikací, Grada Publishing 1999, 492 stran, ISBN 80-716-373-1

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 29.5.2018

**Vedoucí práce:** doc. Ing. Vladislav Škorpil, CSc.

**Konzultant:** Ing. Ondřej Pavelka, Honeywell

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato bakalářská práce se zabývá konstrukcí bateriově napájeného teplotního senzoru s Wi-Fi modulem ESP32 s možností připojení libovolného průmyslového čidla fungujícího na analogovém přenosu. Při konstrukci modulu byl kladen speciální důraz na minimální spotřebu sensorové jednotky z důvodu dlouhé výdrže zařízení na jeden nabíjecí cyklus baterie, na způsob uložení naměřených dat a na realizaci webového rozhraní, které data pohodlně zobrazuje koncovému uživateli. Teoretická část práce je věnována popisu oblasti Internetu věcí (Internet of Things, zkráceně IoT), přičemž klade důraz na jeho historii, využití a bezpečnost. Kapitola také pokrývá široké možnosti aplikace různých senzorů a jejich využití jak v průmyslu, tak v běžném životě. Kapitulu o Internetu věcí doplňují podkapitoly věnované popisu nejčastějších komunikačních technologií a protokolů. Praktická část je věnována konstrukci celého zařízení. Je rozdělena do tří hlavních kapitol. První z nich je věnována specifikaci požadavků a návrhu zařízení, přičemž klade velký důraz na volbu vhodného bateriového způsobu napájení modulu a kalkulaci doby životnosti baterie. Probrány jsou také jednotlivé části celého zařízení, jako je výběr samotného teplotního čidla či řešení vstupů pro připojení průmyslových čidel. Druhá kapitola se zabývá praktickou realizací modulu a to jak po hardwarové, tak softwarové stránce. Hardwarová část popisuje realizaci desky s tištěnými spoji a její krytí zhotovené pomocí 3D tiskárny, zatímco softwarová vysvětluje, jak je modul naprogramován a jak funguje webová prezentace. Poslední kapitola práce uvádí průběh testování modelu a celkovou finanční kalkulaci projektu.

## **KLÍČOVÁ SLOVA**

Cloud, DS18B20, ESP32, Internet věcí, Teplotní senzor, Vstup 0-10 V, Vstup 4-20 mA, Webový server, Wi-Fi, WROOM-32



## ABSTRACT

This bachelor thesis focuses on construction of battery powered temperature sensor with Wi-Fi module ESP32 with the possibility of connection to any analogue industrial sensor. The construction of module emphasizes on minimal battery power consumption with taking into account the longest possible duration per one charging battery cycle. It is also stressed on the way of storing measured data and realization of website, which presents those data to the end user in a comfortable manner. The theoretical part describes Internet of Things, accentuates its history, usage and security. The chapter also covers vast possibilities of application of different sensors and their usage in both industry and ordinary life. The Internet of Things chapter is concluded with description of most used communication technologies and protocols. The practical part is concerned with construction of temperature sensor with ESP32 module. It is divided into three main parts. First part deals with the specification of requirements and the design of the device. It also takes into account the choice of best possible way of powering the device by battery and calculation of battery life. Chapter also discusses individual parts of the whole device, such as the choice of temperature sensor or the solution of connectability of industrial sensors. The second practical chapter deals with the realization of device, both in hardware and software way. Hardware part describes realization of the printed circuit board and its 3D printed coverage, whereas software part sheds light on the programming of the device and how the web presentation works. The final chapter presents two main things: how was the module tested and complete financial calculation for the whole project.

## KEYWORDS

0-10 V input, 4-20 mA input, Cloud, DS18B20, ESP32, Internet of Things, Temperature sensor, Web server, Wi-Fi, WROOM-32

ROŠKO, Tomáš. *Teplotní senzor využívající modul (Espressif) ESP32*. Brno, 2018, 52 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Teplotní senzor využívající modul (Espresso) ESP32“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval panu doc. Ing. Vladislavu Škorpilovi, CSc. za odborné vedení této bakalářské práce. Velké díky také patří odbornému konzultantovi panu Ing. Ondřeji Pavelkovi z firmy Honeywell za jeho cenné rady, díky kterým byla cesta k dokončení práce mnohem jednodušší. Na závěr bych chtěl poděkovat své rodině, přítelkyni a přátelům, kteří mě po celou dobu podporovali.

Brno .....

.....

podpis autora

# Obsah

Úvod	11
<b>1 Internet věcí</b>	<b>12</b>
1.1 Obecné informace	12
1.2 Vznik Internetu věcí a jeho budoucnost	12
1.3 Využití v současnosti	14
1.3.1 Chytré osobní doplňky	14
1.3.2 Chytrá domácnost	14
1.3.3 Chytré město	15
1.3.4 Chytrý průmysl	15
1.3.5 Chytrá příroda	16
1.4 Zabezpečení	16
1.5 Komunikační technologie	16
1.5.1 Wi-Fi	17
1.5.2 LTE-M	17
1.5.3 Bluetooth	17
1.5.4 ZigBee	18
1.5.5 Z-Wave	18
1.6 Komunikační protokoly	19
1.6.1 HTTP	19
1.6.2 MQTT	19
1.6.3 CoAP	19
<b>2 Modul ESP8266 a ESP32</b>	<b>20</b>
2.1 Espressif Systems	20
2.2 ESP8266	20
2.3 ESP32	21
<b>3 Specifikace požadavků a návrh zařízení</b>	<b>22</b>
3.1 Řešení systému	22
3.2 Napájení zařízení	23
3.3 Teplotní čidlo	24
3.3.1 Teplotní čidlo DS18B20	24
3.4 Napěťový a proudový vstup	26
3.5 Měření stavu baterie	27
3.6 Nastavení parametrů	28
3.7 Výpočet výdrže baterie	28

3.8	Cloudové řešení . . . . .	29
3.8.1	ThingSpeak . . . . .	30
3.8.2	Thinger.io . . . . .	30
3.8.3	Azure IoT Suite . . . . .	30
3.8.4	Vlastní řešení . . . . .	31
<b>4</b>	<b>Praktická realizace</b>	<b>32</b>
4.1	Hardwarová část . . . . .	32
4.1.1	Deska plošných spojů . . . . .	32
4.1.2	Krytí DPS . . . . .	34
4.2	Softwarová část . . . . .	35
4.2.1	Programování modulu ESP32 . . . . .	35
4.2.2	Webové rozhraní . . . . .	37
<b>5</b>	<b>Testování a cenová kalkulace</b>	<b>39</b>
5.1	Testování funkčnosti finálního produktu . . . . .	39
5.2	Cenová kalkulace . . . . .	40
<b>6</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>
	<b>Seznam zkratk</b>	<b>46</b>
	<b>Seznam příloh</b>	<b>47</b>
<b>A</b>	<b>Návrh zařízení</b>	<b>48</b>
A.1	Celkové schéma zapojení . . . . .	48
A.2	Seznam použitých komponent . . . . .	49
<b>B</b>	<b>Fotografie projektu</b>	<b>50</b>
<b>C</b>	<b>Příložené médium</b>	<b>52</b>
C.1	Struktura obsahu příloženého média . . . . .	52

# Seznam obrázků

1.1	Zájem o Internet věcí podle Google Trends (převzato a upraveno z [7]).	13
1.2	Chytré léky (převzato z [6]).	14
1.3	Detektory monitorující zvířata (převzato z [6]).	16
2.1	WROOM-02 (převzato z [19]).	20
2.2	Vývojová deska ESP32 DevKitC.	21
3.1	Blokové schéma funkčnosti zařízení.	22
3.2	Vybíjecí charakteristika baterie (převzato z [21]).	24
3.3	Teplotní čidlo DS18B20 TO-92.	25
3.4	Zapojení čidla DS18B20 v režimu externího napájení.	26
3.5	Napěťový měnič s integrovaným obvodem MT3608.	27
3.6	USB-Serial převodník WaveShare (převzato z [24]).	28
4.1	Vizualizace horní strany DPS.	32
4.2	Vizualizace spodní strany DPS.	33
4.3	Obvod sloužící k automatickému nahrávání firmwaru.	34
4.4	Vizualizace modelu krabičky a víka.	35
4.5	Logická struktura programu.	36
4.6	Výsledná podoba stránky na různých zařízeních.	38
5.1	Umístění zařízení při testovacím měření.	40
A.1	Celkové schéma zapojení.	48
B.1	Osazená deska tištěných spojů.	50
B.2	Vytištěný model krabičky s víkem se zatavenými závity.	50
B.3	Fotografie finálního produktu.	51

# Seznam tabulek

1.1	Přehled standardů Wi-Fi (převzato a upraveno z [10]). . . . .	17
4.1	Struktura tabulky v MySQL databázi. . . . .	37
5.1	Cenová kalkulace jednotlivých komponent. . . . .	41
A.1	Seznam použitých komponent. . . . .	49

# Úvod

Internet věcí je jedním z nejrychleji rozvíjejících se průmyslů posledních let – některé odhady tvrdí, že v roce 2020 bude existovat padesát miliard IoT senzorů [1]. Od roku 1982, kdy vznikl první senzor Internetu věcí [4], uplynulo více než třicet let a IoT senzory jsou běžnou součástí nejen průmyslových řešení, ale i měst a domácností, kde svým uživatelům značně usnadňují život.

Práce je rozdělena do pěti částí. První část je věnována oblasti Internetu věcí, jeho vzniku, využitím a problematice zabezpečení, jehož význam v dnešní době rapidně roste. Důraz je kladen také na komunikační technologie a protokoly, které senzory nejčastěji využívají. V druhé části práce je stručně představena firma Espressif Systems a její dva nejznámější moduly ESP8266 a ESP32. Třetí část se zabývá specifikací požadavků a návrhu zařízení, přičemž pokrývá mnoho témat. Jedná se například o výběr vhodného způsobu napájení zařízení, výběr teplotního čidla, řešení proudového a napěťového vstupu, komunikaci s uživatelem, měření stavu baterie a výpočet její životnosti. Čtvrtá kapitola se zabývá praktickou realizací celého projektu a je rozdělena na hardwarovou a softwarovou část. Hardwarová část pojednává o realizaci desky plošných spojů a vhodného krytí DPS, zatímco softwarová část se potýká s problematikou programování modulu a tvorbou webového rozhraní. Finální pátá část práce popisuje průběh testování finálního produktu, které probíhalo nepřetržitě po dobu jednoho týdne. Na závěr nechybí finanční kalkulace celého produktu.

Cílem práce je navrhnout řešení teplotního senzoru s Wi-Fi modulem ESP32. Modul bude napájen z baterie, tudíž jsou kladeny vysoké požadavky na co nejmenší spotřebu energie. Dále bude zařízení obsahovat vstupní konektory pro připojení průmyslových čidel pracujících na bázi analogového přenosu 4-20 mA a 0-10 V. Modul také bude schopen odesílat a ukládat data do cloudového řešení a „komunikovat“ tak s koncovým uživatelem.



# 1 Internet věcí

Tato kapitola pojednává o oblasti „Internetu věcí“ (z angl. Internet of Things – zkráceně IoT), jeho vývoji, využití a také zabezpečení. Dále jsou v kapitole popsány komunikační protokoly a technologie, které IoT využívá.

## 1.1 Obecné informace

Internet věcí je relativně nový trend v oblasti informačních technologií s obrovským potenciálem do budoucna. Obecně lze říct, že se jedná o zastřešující pojem pro přístroje, které jsou určitým způsobem připojeny k Internetu a jsou mezi sebou schopny komunikovat [1]. Hlavní úlohou těchto zařízení je sběr dat pomocí senzorů, jejich zpracování a následné sdílení mezi sebou, případně interpretace těchto dat uživatelům. Důležitým požadavkem je také dlouhá výdrž, a to zejména v případě, že jsou tyto zařízení napájena bateriově.

Jedním z mnoha příkladů použití Internetu věcí jsou například „chytré“ termostaty, které dokáží komunikovat s vytápěcím zařízením, jako jsou kotle a bojler, sloužící pro ohřev teplé vody. Dokáží se přizpůsobovat dle denní doby a ročního období, ale i podle teplotních preferencí uživatele. Dalo by se tedy říci, že nám tyto „věci“ usnadňují život, šetří peníze a čas.

## 1.2 Vznik Internetu věcí a jeho budoucnost

Cesta, která vedla nejen k vytvoření pojmu Internet věcí, započala již roku 1832, kdy Carl Friedrich Gauss a Wilhelm Weber použili elektromagnetický telegraf ke komunikaci na vzdálenost přes 1200 metrů [2]. Jejich úspěch byl předznamenáním prudkého rozvoje dalších vynálezů a elektrotechnického průmyslu vůbec.

V roce 1844 Samuel Morse realizoval první úspěšný přenos Morseovy abecedy na území Severní Ameriky z Washingtonu, D.C. do Baltimoru (vzdušnou čarou přibližně 60 km).

O 22 let později Karl Steinbuch řekl: „Během několika desetiletí budou již počítače součástí každého průmyslového výrobku,“ [3].

Roku 1969 byla založena síť ARPANET, kterou financovalo ministerstvo obrany USA a sloužila především jako experimentální síť. Byla zárodkem toho, co dnes označujeme jako internet.

V roce 1982 ještě nebyl termín Internet věcí zaveden, nicméně vzniklo první zařízení, které bychom tak v dnešní době označili. Studenti z Carnegie-Mellonovy univerzity připojili automat na Coca-Colu k internetu [4]. Takto zjišťovali stav zásob a teplotu nápojů.

Roku 1889 anglický fyzik Tim Berners-Lee navrhnul World Wide Web a o dva roky později vytvořil první webovou stránku. V roce 1999 Kevine Ahston, výkonný ředitel firmy AutoID-Labs, poprvé použil termín Internet věcí.

První zmínky o IoT mimo vědecké prostředí se dostaly k veřejnosti v období roku 2003-2004 díky deníku The Guardian, vědeckému časopisu Scientific American či novin The Boston Globe.

Mezi širokou veřejností se však pojem Internet věcí dostal až v roce 2005, díky Mezinárodní telekomunikační unii, která vydala zprávu na toto téma. Na svět bylo také uvedeno Arduino, levný a uživatelsky jednoduše programovatelný mikrokontrolér. V roce 2010 Čínský premiér Wen Ťia-pao označil Internet věcí jako klíčový průmysl pro Čínu.

Z důvodu docházejícího adresního prostoru protokolu IPv4, byl roku 2011 spuštěn protokol IPv6, který ho má postupně nahradit a dokáže adresovat až  $2^{128}$  zařízení v internetu. Steve Liebson tehdy prohlásil: „Rozšíření adresového prostoru protokolem IPv6 znamená, že jsme schopni přiřadit adresu každému atomu na povrchu Země a stále bude dostatek adres dostupných k tomu, abychom tento proces mohli zopakovat nejméně stokrát,“ [5].

Co se bude dít v příštích letech? Jak lze vidět na obrázku 1.1, který ukazuje míru vyhledávání pojmu Internet věcí, relativně vztaženou ke stupnici nula až sto, kde sto značí nejvyšší zájem a nula nejnižší zájem, za období posledních deseti let od konce listopadu 2017, zájem o Internet věcí určitě poroste. Ačkoli se názory mírně liší, předpovídá se, že v roce 2020 bude existovat 50 miliard IoT zařízení a každou sekundu jich přibude dalších sto [1]. V průměru tedy bude připadat 6 zařízení na člověka.



Obr. 1.1: Zájem o Internet věcí podle Google Trends (převzato a upraveno z [7]).

## 1.3 Využití v současnosti

V dnešní době lze ve světě nalézt nepřeberné množství zařízení spadajících do oblasti Internetu věcí, které přináší nové možnosti. Ať už se jedná o produkty, které dokáží sledovat kondici lidského těla či to, jestli jsme doma vypnuli troubu a jestli máme dostatek potravin v lednici, existují i přístroje monitorující míru znečištění, kvalitu vzduchu a vody ve městech [6]. Nesmíme zapomínat také na průmysl – i zde hraje IoT velkou roli. V následující části práce si ukážeme praktické využití IoT v každé z uvedených oblastí.

### 1.3.1 Chytré osobní doplňky

Nositelná elektronika (z angl. Wearable Devices) je pojem označující miniaturizované elektronické zařízení uzpůsobené tak, aby mohlo být běžně nošené na lidském těle. Jedná se například o hodinky, které jsou schopny měřit počet ušlých kroků, spálených kalorií, teplotu pokožky, míru pocení a mnoho dalších informací. Hodinky se dokáží spárovat s telefonem a všechna tato data sdílet. V telefonu se poté ze sesbíraných dat automaticky vytvoří grafy a uživatel je schopen zaznamenávat svůj pokrok.

Za zmínku také stojí speciální obaly na léky, které jsou propojené s mobilním telefonem a jsou schopné jejich uživatele informovat o tom, zda si vzali či nevězali dávku léků.



Obr. 1.2: Chytré léky (převzato z [6]).

### 1.3.2 Chytrá domácnost

Internet věcí je v poslední době často integrovaný i do celých domácností, které jsou poté označovány jako tzv. chytré domácnosti. Síť elektronických zařízení zabudovaných v domě je schopna monitorovat zapnuté spotřebiče, žárovky nebo dálkově ovládat klimatizaci. Ke všem těmto údajům mají obyvatelé domu přístup ze svých

mobilních zařízení a tak mohou z jakéhokoli místa na světě dům nejen kontrolovat, ale také ovládat. Například o rostliny se mohou starat zavlažovací stanice, které je zalévají na základě měření vlhkosti půdy.

Dům je také možné zabezpečit speciálními kamerami, které uživatele informují při narušení prostoru, a ti tak mají čas rychle zareagovat. Pokud by náhodou došlo k vodovodní havárii, informuje obyvatele domu o nehodě chytrý hlásič umístěný ve sklepě.

Bezdrátová zařízení umí také řešit každodenní problémy, jako jsou ztracené klíče. Mikročip v klíčích ve spojení s mobilním telefonem či počítačem dokáže lokalizovat klíče s přesností na několik decimetrů.

### 1.3.3 Chytré město

Internet věcí dokáže také velmi zjednodušit běžný život, například v dopravním provozu. Chytré kamery, sledující pohyb automobilů a dalších dopravních prostředků po městě dokáží řidičům poradit, kudy jet, pokud je špička a město je ucpané. Také dokáží monitorovat volná a obsazená parkovací místa v okolí a prostřednictvím mobilního zařízení jsou schopny informovat řidiče o volných parkovacích místech.

Chytrá zařízení, ovlivňující veřejný prostor, jsou například pouliční lampy. Dokáží šetřit energii tím, že mění svoji intenzitu osvětlení podle počasí či denní doby.

Existují i takzvané chytré koše na odpadky, které mohou díky sensorům informovat úklidové firmy, pokud se blíží hranice jejich zaplnění.

### 1.3.4 Chytrý průmysl

Zařízení, kterých se využívá v IoT, najdou široké uplatnění i v průmyslu. Jedná se například o senzory, které sledují stav výrobních strojů a také to, zda některé výrobní části nepřekročily limity svého opotřebení.

Dalšími příklady jsou čidla monitorující vlhkost půdy, předpověď počasí, ale také detektory sledující použití pesticidů v okolí. Farmáři mohou poté tato data využít a přizpůsobit se jim.

Nejen, že všechna tato zařízení usnadňují průmyslové procesy, ale dle výzkumu z roku 2016 provedeného firmou Hewlett-Packard Enterprise až 72 % firem z celkového vzorku 3100 firem s více než pěti sty zaměstnanci zavedlo IoT (ať už v podobě klimatizace či osobních mobilních zařízení) do běžného pracovního prostředí svých zaměstnanců. Přes 78 % dotázaných společností uvedlo, že zavedení těchto zařízení zlepšilo efektivitu práce firmy, zatímco 75 % také sdělilo, že z tohoto důvodu se také zvýšily jejich zisky [8].

### 1.3.5 Chytrá příroda

Internet věcí nachází své uplatnění i v oblasti ochrany přírody. Ilegální kácení stromů je hlídáno speciálními detektory, které jsou instalovány na náhodné stromy v pralesích. Jakmile je toto zařízení v pohybu, vysílá signál příslušným institucím, které na tuto informaci reagují. Na různé druhy zvířat se připínají obojky za účelem monitorování jejich pohybu a případné ochrany proti pytlačení.

Varovná čidla, která nás informují o blížícím se nebezpečí jako je výbuch sopky, sesuv půdy, zemětřesení a dalších přírodních katastrof nejsou dnes výjimkou.



Obr. 1.3: Detektory monitorující zvířata (převzato z [6]).

Očekává se, že IoT bude mít vliv také na vládu, vzdělání, dopravu a finance. Existuje téměř nekonečná kombinace aplikací těchto chytrých zařízení.

## 1.4 Zabezpečení

S IoT mimo jiné přichází také velká míra nebezpečí. Před dvaceti lety bylo téměř nemyslitelné, že by bylo možné ukrást heslo k bankovnímu účtu pomocí telefonu. V dnešní době, při používání chytrých zařízení, se toto děje čím dál tím častěji. Čím více připojených zařízení ve světě je, tím více je možností pro útočníky získat citlivá data. Naštěstí se s rozvíjejícím počtem těchto zařízení začíná problém zabezpečení řešit více a více [9].

## 1.5 Komunikační technologie

IoT v dnešní době nabízí širokou škálu možností komunikace. Mezi nejdůležitější aspekty při výběru vhodné přenosové technologie přenosu patří rychlost přenosu dat, dosah, energetická specifikace, bezpečnost a způsob zpracování dat.

### 1.5.1 Wi-Fi

Wi-Fi neboli Wireless Fidelity je v informatice označení pro několik standardů IEEE 802.11x (x označuje danou verzi) popisujících bezdrátovou komunikaci v počítačových sítích. Tato technologie využívá tzv. bezlicenčního frekvenčního pásma, proto je ideální pro budování levné, ale výkonné sítě, bez nutnosti pokládky kabelů [10]. Každý standard je definován maximální přenosovou rychlostí a frekvencí, na které pracuje. Nejčastěji je využíváno pásmo 2,4 a 5 GHz. Přehled základních verzí je zobrazen v tabulce 1.1.

Poslední dobou se začíná do popředí dostávat také PoWi-Fi (Power over Wi-Fi), která umožňuje přenášet i energii potřebnou k napájení nízkoenergeticky náročných zařízení pomocí signálu Wi-Fi. Tato technologie však není v současnosti moc rozšířená a je teprve ve vývoji [11].

Tab. 1.1: Přehled standardů Wi-Fi (převzato a upraveno z [10]).

Standard IEEE	Pásmo [GHz]	Přenosová rychlost [Mbit/s]	Rok vydání
802.11a	5	54	1999
802.11b	2,4	11	1999
802.11g	2,4	54	2003
802.11n	2,4 nebo 5	600	2009
802.11ac	2,4 nebo 5	1800	2013

### 1.5.2 LTE-M

LTE-M je relativně nová, energeticky nenáročná technologie určená na delší vzdálenosti, která je speciálně navržena pro IoT zařízení. Mezi její hlavní přednosti patří nízké provozní náklady a výdrž zařízení na baterii až 10 let. Přenosová rychlost je této technogíe je menší, typicky 1 Mbit/s [12].

### 1.5.3 Bluetooth

Bezdrátová technologie Bluetooth je definovaná standardem IEEE 802.15.1. Vytvořena byla v roce 1994 firmou Ericsson jako náhrada za drátové rozhraní RS-232 a také za účelem redukce množství kabelů. Je určena pro komunikaci na kratší vzdálenosti, maximálně do 100 metrů při plné viditelnosti, reálně je však tato vzdálenost menší. K přenosu využívá, podobně jako Wi-Fi, bezlicenční pásmo 2,4 GHz, avšak frekvence se v některých státech může lišit.

Technologie Bluetooth si za dobu své existence prošla mnoha změnami a vylepšeními. V roce 1999 se na svět dostala první verze s označením 1.0 a 1.0B. Tato verze

měla mnoho nedostatků, proto byla roku 2001 představena verze 1.1, která tyto nedostatky upravovala a byla první plně funkční verzí. O 2 roky později vznikla verze 1.2, která oproti verzi 1.1 uměla rychleji hledat a připojovat se k blízkým zařízením.

V roce 2004 byla představena verze 2.0+EDR. V této verzi došlo k navýšení rychlosti přenosu dat. Další verze 2.1+EDR vznikla v roce 2007. Hlavním rysem této verze je lepší zabezpečení při párování zařízení.

Bluetooth 3.0+HS, který byl uveden roku 2009 opět zvýšil přenosovou rychlost, tentokrát na 24 Mbit/s.

V roce 2010 vznikla verze Bluetooth 4.0+LE (BLE – Bluetooth Low Energy), která byla vhodná pro zařízení Internetu věcí. Jejím myšlenkou bylo snížení energetické náročnosti za účelem možnosti dlouhodobého provozu.

Zatím poslední verzí je Bluetooth 4.2 představený roku 2014, taktéž vhodný pro IoT zařízení. V této verzi došlo k dalším vylepšením a také ke zlepšení bezpečnosti komunikace [13].

#### **1.5.4 ZigBee**

Technologie ZigBee, která vznikla roku 2004, je vystavěna na standardu IEEE 802.15.4 a je velmi podobná technologii Bluetooth [14]. Mezi její hlavní výhody patří nízká energetická náročnost. Komunikuje až do vzdálenosti 100 metrů a její přenosová rychlost dosahuje hodnot 20, 40 nebo 250 Kbit/s. Není tedy vhodná pro přenos objemově větších dat.

Stejně jako Wi-Fi a Bluetooth využívá pásmo 2,4 GHz. Ochrana dat je zajištěna pomocí symetrické kryptografie a maximální počet připojených zařízení do jedné sítě je 1024.

#### **1.5.5 Z-Wave**

Z-Wave je bezdrátová technologie vyvinutá firmou Zensys v roce 2001 [15]. Mezi její nejdůležitější vlastnosti patří nízká spotřeba a několika-úrovňové zabezpečení. Díky tomu, že pracuje na málo využívané frekvenci 868 MHz je velmi odolná proti rušení. Maximální udávaný dosah je 100 metrů a komunikační rychlost činí 100 Kbit/s.

Princip fungování sítě Z-Wave je založen na jednom centrálním prvku, tzv. Z-Wave Controller, ke kterému se pomocí procesu párování připojují další zařízení. Každá taková síť má své identifikační číslo o délce 32 bitů, stejně tak jako každé zařízení v síti (8 bitů).

## 1.6 Komunikační protokoly

Komunikační protokol je standard, podle kterého se řídí komunikace mezi dvěma zařízeními. Nelze jasně určit, který protokol je pro výměnu dat nejlepší, protože každý je využíván k jinému účelu.

### 1.6.1 HTTP

Jedním z nejpobulárnějších protokolů pro přenos dat je HTTP (Hypertext Transfer Protocol) využívající TCP port 80 [16]. Byl založen roku 1989 a funguje na principu klient-server. To znamená, že klient pošle požadavek na server, který jej zpracuje a vrátí klientovi odpověď.

Jeho zabezpečenou variantou je protokol HTTPS (TCP port 433), který společně s HTTP využívá protokol SSL nebo TLS. Zajišťuje autentizaci, integritu a důvěrnost přenášených dat.

### 1.6.2 MQTT

MQTT (Message Queuing Telemetry Transport) byl uveden na svět v roce 1999 společností IBM [16]. Své uplatnění najde především u méně výkonných zařízení. U protokolu MQTT jsou data přenášena pomocí TCP protokolu na bázi publish (vydavatel) a subscribe (odběratel), mezi kterými je tzv. broker (zprostředkovatel). V praxi to znamená, že klient (vydavatel) pošle data zprostředkovateli, který je zpracuje a podle toho se rozhodne, kterému klientovi (odběratel) je pošle.

Společně s protokolem CoAP se jedná o nejpoužívanější protokol v Internetu věcí a lze se s ním setkat například v aplikaci Facebook Messenger, kterou měsíčně využívá více než 1,3 miliardy lidí [17].

### 1.6.3 CoAP

Protokol CoAP (Constrained Application Protocol) je určený pro komunikaci mezi zařízeními s nízkou spotřebou energie [16]. Na rozdíl od MQTT a HTTP využívá protokol UDP, je tedy rychlejší, avšak nedochází k ověření doručení dat. Funguje na principu požadavku a odpovědi, podobně jako HTTP.



## 2 Modul ESP8266 a ESP32

Tato kapitola má za cíl představit mikrokontroléry ESP8266 a jeho nástupce ESP32. Na začátku kapitoly je uveden popis firmy, která tyto čipy uvedla na trh. Následně jsou popsány hlavní parametry těchto mikrokontrolérů.

### 2.1 Espressif Systems

Firma Espressif Systems se sídlem v Šanghaji se zabývá produkcí nízkoenergetických, avšak výkonných Wi-Fi a Bluetooth mikrokontrolérů pro užití ve sféře Internetu věcí. Kromě toho také poskytují open-source vývojovou platformu, speciálně určenou pro jejich zařízení [18]. Mezi jejich nejznámější produkty řadíme modul ESP32 a ESP8266, které mají spoustu verzí lišících se podle potřeby použití.

### 2.2 ESP8266

ESP8266 je cenově dostupný model, který se dočkal své pozornosti v roce 2014. I přes jeho poměrně stáří se s ním lze velmi často setkat, a to ať už v podobě holého čipu od firmy Espressif Systems (například WROOM-02 [19]), nebo v podobě různých vývojových kitů a desek upravených jinými firmami.



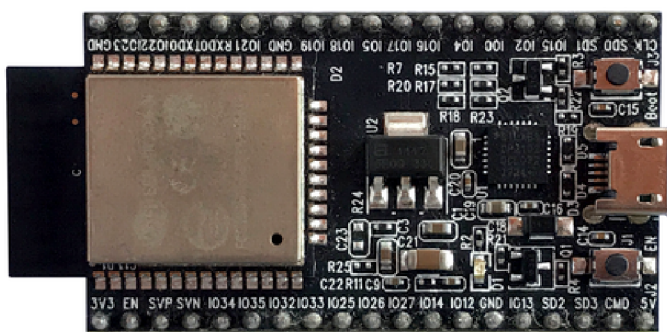
Obr. 2.1: WROOM-02 (převzato z [19]).

Využívá logiku 3,3 V a jeho průměrná spotřeba je 80 mA, která však ve špičkách může dosahovat až 300 mA. Je osazen 32bitovým mikroprocesorem Xtensa L106 běžícím na frekvenci 80 MHz. Obsahuje anténu pro připojení k Wi-Fi, 16 GPIO pinů, rozhraní I<sup>2</sup>C, UART a 10 bitový AD převodník (ADC).

## 2.3 ESP32

ESP32 je novější verzí modulu ESP8266 z roku 2016. Na rozdíl od ESP8266 je o několik desítek korun dražší, avšak výrazně výkonnější. Díky rozsáhlé technické dokumentaci, která je plně přeložena do angličtiny, vznikají nejrůznější zařízení založené na tomto modulu, a to ať už v rukou domácích kutilů či profesionálů.

Stejně jako u jeho předchůdce existuje mnoho odvozenin a verzí, které se liší v závislosti na účelu použití. Jedním z nich je například vývojová deska, která již obsahuje regulátor vstupního napětí na 3,3 V a značně tak usnadňuje práci. Kromě regulace napětí také zařizuje sériovou komunikaci, obsahuje tlačítka umožňující ruční nahrání nového softwaru či restart zařízení a mnoho dalších funkcí.



Obr. 2.2: Vývojová deska ESP32 DevKitC.

ESP32, stejně jako ESP8266 pracuje s napětím 3,3 V. Jeho běžná spotřeba se od ESP8266 moc neliší, musíme však vzít v potaz, že výkonově je na tom podstatně lépe. Záleží ovšem na mnoha faktorech; kupříkladu s klesající silou signálu Wi-Fi spotřeba roste. Stejně jako ESP8266 však dokáže běžet v několika druzích úsporných režimů, které naopak šetří energii.

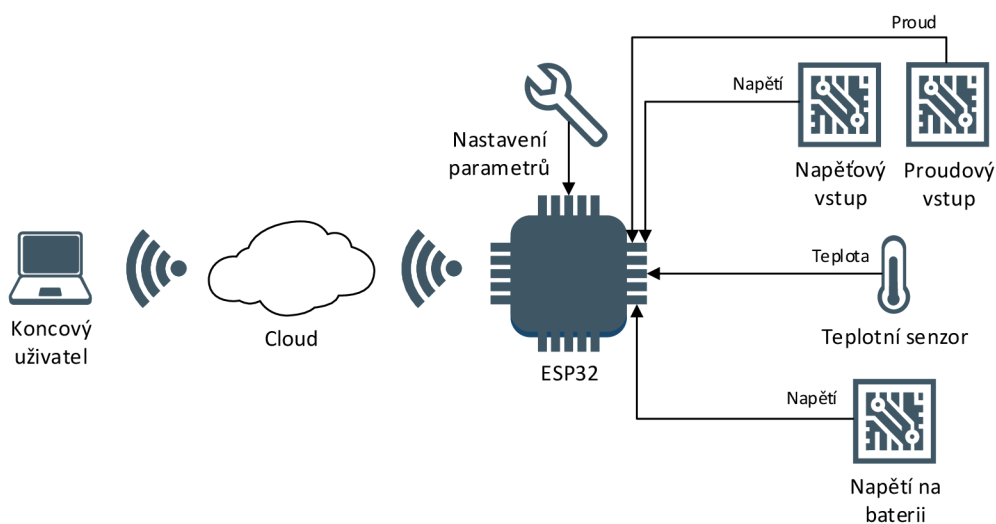
Modul je vybaven procesorem Xtensa LX6, který operuje na frekvenci 160 až 240 MHz. Obsahuje také jeden ULP procesor, který najde své využití v úsporných módech. Anténa pro připojení k Wi-Fi a Bluetooth (BLE) je součástí každého modulu. Oproti ESP8266 obsahuje větší množství GPIO pinů, větší počet rozhraní pro připojení periférií a 12 bitové AD převodníky. Nechybí ani zabudovaný teplotní senzor či Hallova sonda.

## 3 Specifikace požadavků a návrh zařízení

V této kapitole jsou specifikovány jednotlivé požadavky a cíle této práce. Nejprve je popsáno, jak by mělo celé zařízení fungovat. Dále je vysvětleno, jak je zařízení napájeno, které teplotní čidlo je použito, jak je řešen napěťový a proudový vstup, měření stavu baterie, komunikace s uživatelem a nastínění výpočtu výdrže zařízení na baterii. Ke konci kapitoly jsou zmíněny dostupné cloudové služby pro ukládání dat z IoT zařízení.

### 3.1 Řešení systému

Cílem této práce je navrhnout zařízení spadající do kategorie Internetu věcí. Takové zařízení by mělo splňovat několik kritérií. Jde především o jeho kompaktnost, nízkou energetickou náročnost a schopnost komunikovat s okolím. Také by neměly být opomíjeny náklady na výrobu tohoto zařízení.



Obr. 3.1: Blokové schéma funkčnosti zařízení.

Zařízení má být schopno měřit okolní teplotu pomocí teplotního čidla, měřit hodnotu na napěťovém a proudovém vstupu a informovat uživatele o stavu baterie. Všechna tato data by měla být změřena v co nejkratším časovém úseku za účelem úspory baterie. Při výběru jednotlivých komponent je proto nutné dbát na jejich spotřebu v režimu nečinnosti. Zařízení by také mělo mít možnost připojit konektory pro sériovou komunikaci, aby uživatel mohl pohodlně nastavit potřebné údaje pro připojení k Wi-Fi a časovou periodicitu pro opakování měření.

Hlavním komponentem celého zařízení bude již zmíněný modul ESP32. Po zpracování dat modulem je tato data potřeba vhodně odeslat do vybraného cloudového

řešení, kde budou zobrazena uživateli. Uživatel by měl mít možnost vybrat časový úsek podle toho, jaká data si přeje vidět. Přenos dat musí být zabezpečen, aby nemohlo dojít k odposlechnutí.

Všechnu elektroniku, konektory pro připojení externích čidel a držák na baterii je potřeba osadit na vytvořenou desku plošných spojů. Tato deska musí být umístěna do vhodně zvolené krabičky, která bude vytisknuta na 3D tiskárně.

## 3.2 Napájení zařízení

Celé zařízení by mělo být napájeno pouze bateriově. Na trhu existuje nepřeberné množství baterií, ze kterých je možno vybírat. Z důvodu úspory místa však nepřichází v úvahu použití více článků. Je proto potřeba zvolit baterii jednu, která dokáže poskytnout největší možnou kapacitu a zároveň potřebné napětí pro napájení zařízení.

První myšlenkou bylo použít baterii typu Li-Ion. Její největší výhodou je poměrně velká kapacita vzhledem k její velikosti. Nejčastěji jsou vyráběny ve velikosti typu 18650 (18 mm průměr, 65 mm délka), která se používá například v power bankách, elektromobilech či bateriích notebooků [20]. Li-Ion baterie se vyrábí v nominální hodnotě 3,7 V, napětí však může dosahovat hodnoty až 4,2 V při plném nabití.

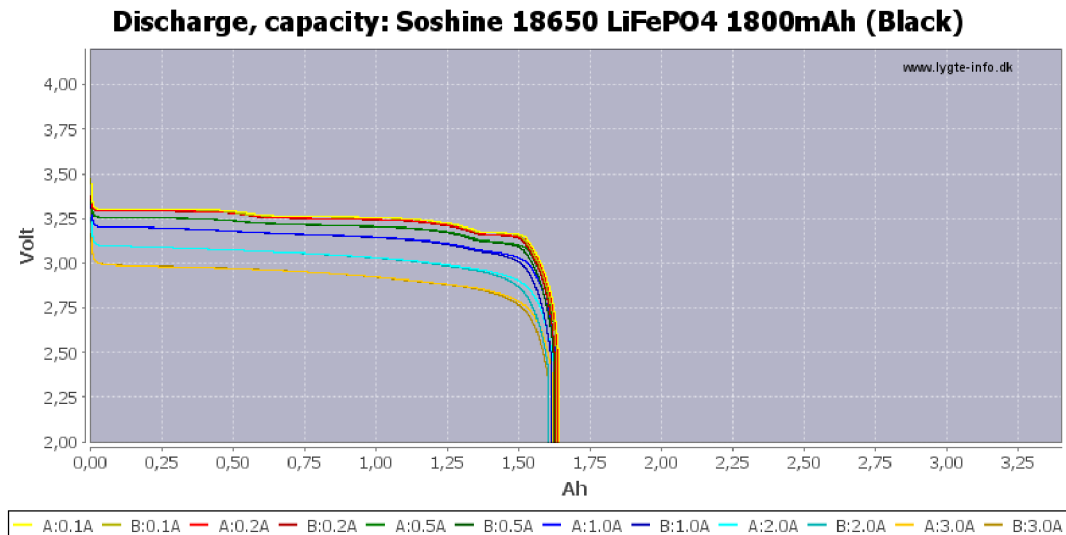
Modul ESP32 pracuje s napětím v mezích od 2,3 do 3,6 V. Proto by bylo v případě Li-Ion baterie nutné použít stabilizátor, který by napětí dodávané baterií stabilizoval na hodnotě 3,3 V (nominální hodnota ESP32). Výběr stabilizátoru probíhal dle následujících kritérií:

- Velikost napětí, na které se stabilizuje.
- Maximální velikost dodávaného proudu.
- Velikost klidového proudu, odebíraného za chodu stabilizátoru.
- Napěťový pokles neboli hodnota napětí, o kterou musí být zdrojové napětí větší než napětí, na které chceme stabilizovat.

Po důkladné rešerši existujících stabilizátorů dostupných na trhu bylo zjištěno, že téměř žádný plně neodpovídá vytyčeným kritériím tak, aby zařízení fungovalo správně a zároveň nedocházelo ke zbytečnému nevyužití kapacity baterie. Proto bylo od tohoto řešení upuštěno.

Jako nejlepší varianta se nakonec ukázalo použít baterii typu LiFePo<sub>4</sub>. Od Li-Ion baterie se liší především menším nominálním napětím, které je 3,2 V, ale také kapacitou, která je o několik desítek procent menší při použití stejné velikosti baterie. Hodnota napětí baterie se pohybuje v rozmezí od 2,5 do 3,6 V, což je pro použití s modulem ESP32 ideální. Byla proto zvolena baterie LiFePo<sub>4</sub> od firmy Soshine typu 18650 s kapacitou 1800 mAh. Reálná kapacita je však o něco menší, přibližně 1600

mAh. Baterie navíc obsahuje ochranné obvody proti přebíjení, podbíjení a zkratu. Udávaná životnost je přes 1000 cyklů. Na obrázku 3.2 je znázorněna vybíjecí charakteristika baterie. Lze vidět, že při 2,8 V je kapacita již skoro vyčerpána, a proto budeme toto napětí považovat jako maximální možnou hranici, na kterou necháme baterii vybít. Poté se zařízení trvale uspí, dokud uživatel baterii znovu nenabije. Další vybíjení pod tuto hranici by mohlo vést k poškození baterie.



Obr. 3.2: Vybíjecí charakteristika baterie (převzato z [21]).

### 3.3 Teplotní čidlo

Abychom mohli měřit okolní teplotu, potřebujeme teplotní snímač. V našem případě připadají v úvahu dvě varianty. První variantou je využít integrované čidlo v modulu ESP32. To s sebou ovšem přináší jeden zásadní problém. Při běhu modulu totiž dochází k jeho ohřívání a tím pádem by mohlo dojít ke zkreslení reálné teploty. Druhou variantou je použít některá z dostupných čidel na trhu. Existuje jich široká řada, avšak každé z nich nabízí jiné parametry. Některá čidla dokáží měřit jen teplotu, jiná jsou kombinovaná a dokáží snímat i vlhkost. Záleží i na přesnosti měření, rozsahu teplot, velikosti napájecího napětí, ceně či oblasti použití.

#### 3.3.1 Teplotní čidlo DS18B20

Jedním z nejpoužívanějších čidel pro „domácí využití“ je čidlo DS18B20 od firmy Maxim Integrated. Je populární především mezi domácími kutily, avšak dokáže posloužit i v náročnějších aplikacích. Zvládne měřit teplotu v rozsahu od  $-55$  do

+125°C. V rozmezí od -55 do +85°C měří s odchylkou  $\pm 0,5^\circ\text{C}$  [22]. Oblíbené je díky své příznivé ceně a lze se s ním setkat i ve vodotěsné verzi.

Čidlo funguje na digitálním přenosu a nabízí programovatelné rozlišení, které lze měnit od 9 do 12 bitů. 9 bitů zaokrouhluje teplotu na  $0,5^\circ\text{C}$ , 10 bitů na  $0,25^\circ\text{C}$ , 11 bitů na  $0,125^\circ\text{C}$  a 12 bitů na  $0,0625^\circ\text{C}$ . Stále je však nutné počítat s odchylkou  $\pm 0,5^\circ\text{C}$ . Toto rozlišení má vliv na čas potřebný ke konverzi teploty, tedy čas, za jak dlouho je nám čidlo schopno poskytnout naměřenou hodnotu od momentu, kdy jsme si ji vyžádali. Při 9bitovém rozlišení je tento čas přibližně 94 ms, při 12bitovém 750 ms. Výchozí nastavení rozlišení je 12 bitů, avšak v našem případě bude naprosto stačit 9bitové.

K dostání je ve více pouzdrech, avšak v naší práci bude využito ve verzi TO-92 (obr. 3.3). V této verzi jej lze připojit pomocí tří pinů. První zleva je zemnicí, druhý datový a třetí napájecí.



Obr. 3.3: Teplotní čidlo DS18B20 TO-92.

Díky tomu, že každé čidlo má unikátní identifikační číslo o délce 64 bitů, lze jich pomocí sběrnice 1-Wire připojit více na jeden datový kabel a jednotlivě s nimi komunikovat.

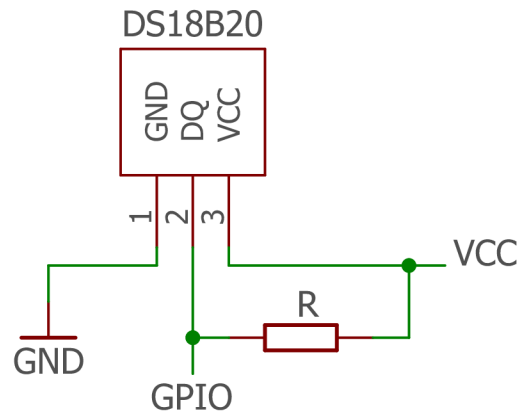
Napájení čidla lze realizovat dvěma způsoby. První z nich je využití tzv. parazitního napětí, kdy je napětí přivedeno na datový pin teploměru. To je realizováno pomocí tranzistoru, který má bázi připojenou na pin mikrokontroléru. Obstarává tak napětí čidla v době, kdy nedochází k přenosu dat. Tím se nabíjí vnitřní kondenzátor, který dodává čidlu napětí při komunikaci. Zároveň je připojen pull-up rezistor<sup>1</sup> mezi napájecí a datový pin. Napájecí pin čidla je spojen s pinem zemnicím.

Výhodou této možnosti je vedení pouze dvou kabelů k čidlu. Nevýhoda tkví v nutnosti použít další elektrotechnickou součástku a ve využití dalšího pinu mikrokontroléru pro připojení tranzistoru. Tato možnost by měla být využita pouze při použití menšího počtu čidel a na kratší vzdálenost komunikace.

<sup>1</sup>Rezistor definující logickou jedničku v dané části obvodu.



Druhou možností je napájet čidlo pomocí externího zdroje napětí. V tomto případě je nutné vložit pouze pull-up rezistor mezi napájecí a datový pin. Toto zapojení lze vidět na obrázku 3.4.



Obr. 3.4: Zapojení čidla DS18B20 v režimu externího napájení.

Výhodou je odpadnutí nutnosti použít další součástku, avšak je třeba vést k čidlu tři kabely namísto dvou. Oproti první možnosti také odpadá potřeba využít další pin mikrokontroléru. V této práci bude využito druhé možnosti, tedy použití externího zdroje napájení.

### 3.4 Napěťový a proudový vstup

Jedním z požadavků této práce je možnost k zařízení připojit libovolná průmyslová čidla. Do zařízení budou připojeny externě, pomocí univerzálních vstupů (konektorů). Analogová hodnota, kterou čidlo „generuje“, se mění podle velikosti měřené veličiny. Například u teplotního čidla fungujícího na principu 0-10 V s rozsahem  $-20$  až  $+80$  °C bude hodnota  $+30$  °C reprezentována jako 5 V. Při použití principu 4-20 mA by analogová hodnota byla 10 mA.

Analogová hodnota bude měřena pomocí AD převodníku modulu ESP32 a uživateli bude zobrazena jako taková, tedy bez převodu zpět na měřenou veličinu.

Aby připojená čidla mohla správně pracovat, musí být napájena dostatečným napětím. Standardně pracují s 9, 12 či 24 V. Z tohoto důvodu je potřeba připojit měnič napětí, jelikož ESP32 nedokáže poskytnout více než 3,3 V. V našem případě lze využít měnič s integrovaným obvodem MT3608, který je schopen dodat napětí až o velikosti 28 V [23]. Toto napětí lze regulovat přímo na desce pomocí potenciometru, viz obrázek 3.5.

Hodnotu rezistoru, na kterém bude probíhat měření při použití vstupu 4-20 mA je vhodné volit tak, aby byl využit co možná největší rozsah AD převodníku. Při použití



Obr. 3.5: Napěťový měnič s integrovaným obvodem MT3608.

rozsahu 3,3 V lze pomocí maximálního proudu (0,02 A) vypočíst, že vhodný odpor je  $165 \Omega$ . Při použití vstupu 0-10 V je potřeba vzít v potaz, že maximální rozlišení AD převodníku dokáže zpracovat pouze napětí do 3,3 V, proto je nutné využít dělič napětí, nejlépe v poměru  $3,0\bar{3}:1$ . Takto nám z maximálních 10 V vzniknou maximálně 3,3 V. Z důvodu, že AD převodník měří napětí, je potřeba zpětně vypočítat proud u vstupu 4-20 mA a správnou hodnotu napětí u vstupu 0-10 V, protože je využit dělič napětí.

### 3.5 Měření stavu baterie

Aby uživatel věděl, kdy se blíží kapacita baterie ke svému konci, je potřeba ho informovat o jejím stavu. Jelikož se nedá změřit kapacita baterie na přímo, je potřeba ji zjistit pomocí hodnoty napětí dodávaného baterií. Z tohoto napětí lze poté přibližně pomocí vybíjecí křivky (obr. 3.2) určit její kapacitu.

Jelikož je baterie schopna dodávat vyšší napětí, než AD převodník ESP32 dokáže zpracovat (maximálně 3,3 V), je potřeba využít dělič napětí. K tomuto děliči napětí bude připojen AD převodník modulu ESP32, pomocí kterého lze úbytek napětí na rezistoru měřit a poté zpětně dopočítat napětí baterie. Zde však nastává problém. Pokud by bylo napětí na baterii například 3,5 V a byl by použit dělič napětí o velikosti 10 a 10 k $\Omega$  (celkově tedy 20 k $\Omega$ ), proud takovýmto obvodem by činil přibližně 0,175 mA. Proud teče obvodem i ve chvíli, kdy je zařízení v úsporném režimu, a proto toto řešení nepřipadá do úvahy.

Dalším řešením by mohlo být tyto odpory řádově zvýšit, například na několik M $\Omega$ , aby byl proud protékající obvodem co nejmenší. V takovémto případě však dochází ke zkreslení napětí měřeného AD převodníkem.

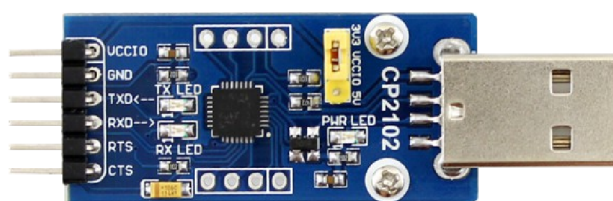
Je tedy nutné navrhnout obvod, který bude sepnut pouze ve chvíli, kdy je to potřeba. Toho lze docílit použitím tranzistorů, které budou spínat či rozepínat obvod podle naší potřeby tak, aby nedocházelo ke zbytečným ztrátám kapacity baterie.



## 3.6 Nastavení parametrů

V IoT je kladen důraz také na přenositelnost. Může se tedy stát, že nějakou dobu bude chtít uživatel používat zařízení doma, poté v práci. Pokaždé je potřeba jiných přihlašovacích údajů k Wi-Fi. Z tohoto důvodu by měl mít uživatel možnost nastavit parametry pro připojení podle své potřeby. Dalším požadavkem je možnost uživatele nastavit časovou hodnotu, která udává, jak často bude probíhat měření.

Pro navázání komunikace mezi počítačem a zařízením lze využít například převodník WaveShare (obr. 3.6), obsahující integrovaný obvod CP2102, který převádí komunikaci z USB na sériovou (UART) [24]. Tento převodník nebude instalován uvnitř krabičky, pouze konektory pro jeho připojení.



Obr. 3.6: USB-Serial převodník WaveShare (převzato z [24]).

Převodník zajišťuje komunikaci pomocí pinů TX (Transmit Data) a RX (Receive Data). Tyto piny jsou nezbytné pro úspěšnou sériovou komunikaci. Dále nabízí využití pinů RTS (Request to Send), CTS (Clear to Send) či DTR (Data Terminal Ready), ale také dalších pinů, které slouží k řízení sériové komunikace. Hodnoty na těchto pinech jsou určovány vůči zemi, tedy pinu GND. Převodník nabízí i možnost zařízení napájet, a to o velikosti 3,3 nebo 5 V. Velikost lze měnit pomocí jumperu<sup>2</sup>, který však zároveň ovlivňuje i logiku komunikačních pinů. Proto musí být nastaven na hodnotu 3,3 V, aby nedošlo k poškození zařízení. Převodník také obsahuje signalizační diody.

V praxi bude zadávání údajů vypadat následovně: Uživatel otevře krabičku zařízení a zapojí převodník pomocí kabelů na vyvedené piny uvnitř krabičky, zasune USB port převodníku do počítače a vloží baterii. Po vložení baterie se zařízení nainstaluje a uživatel může na počítači otevřít sériový terminál, kde provede požadované nastavení všech údajů.

## 3.7 Výpočet výdrže baterie

Jak již bylo zmíněno, při návrhu zařízení Internetu věcí je kladen velký důraz na jeho energetickou náročnost. Proto je potřeba zvolit vhodné komponenty, které budou mít

<sup>2</sup>Mechanická spojka vodičů.

co nejmenší spotřebu.

Jelikož bude zařízení měřit pouze v určitých časových intervalech, je zbytečné, aby běželo i v době, kdy to není potřeba. Při použití ESP32 lze využít jeden z režimů šetření energie, které modul nabízí. Tím je režim hlubokého spánku, při kterém je deklarovaná spotřeba  $5\mu\text{A}$ . Spotřeba ostatních komponent v době, kdy je ESP32 v režimu spánku je zanedbatelná.

Při běhu modulu se spotřeba pohybuje kolem  $80\text{mA}$ , se zapnutou Wi-Fi přibližně  $130\text{mA}$ . Spotřeba u prvků v obvodu jako je teplotní čidlo, napěťový měnič a případná připojená čidla do vstupů se bude pohybovat v řádu jednotek (maximálně pár desítek) mA. Celkově by se tedy v době, kdy modul není v režimu spánku, mohla spotřeba pohybovat kolem  $120\text{mA}$ .

Ze vzorce pro výpočet počtu cyklů měření

$$P_C = 0,85 \cdot \frac{K_B \cdot 3600}{T_S \cdot S_S + T_B \cdot S_B}, \quad (3.1)$$

kde  $K_B$  je kapacita baterie (mAh),  $T_S$  délka spánku zařízení (s),  $S_S$  spotřeba v režimu spánku (mA),  $T_B$  délka doby běhu zařízení v klasickém režimu (s) a  $S_B$  celková spotřeba v klasickém režimu (mA) lze dopočítat počet dní

$$P_D = P_C \cdot \frac{T_C}{24}, \quad (3.2)$$

kde  $T_C$  je celková doba opakování měření v sekundách ( $T_S + T_B$ ).

V případě výše uvedené spotřeby při měření co půl hodiny s kapacitou baterie  $1800\text{mAh}$  a délkou měření 3 sekundy by zařízení mělo vydržet běžet 311 dní na jeden nabíjecí cyklus baterie. Tento výsledek je vynikající, avšak pouze teoretický. V realitě by se výdrž mohla lišit i o několik desítek dnů. To závisí především na reálné spotřebě zařízení, která může variovat v závislosti na kvalitě signálu připojení k Wi-Fi, na kvalitě baterie, okolní teplotě a mnoha dalších faktorech.

## 3.8 Cloudové řešení

Zařízení Internetu věcí sbírají velké objemy dat, které je nutno někde ukládat. Jedním z nejefektivnějších řešení mohou být cloudová úložiště, která jsou schopna denně přijímat obrovské množství dat. Propojení jednotlivých zařízení s cloudem uživatelům umožňují také jejich ovládní na dálku, například pomocí mobilu nebo počítače.

Uložená data je vhodné vizualizovat, popřípadě je dále zpracovat a na základě toho vyvolat další akci. V dnešní době se lze setkat se spoustou služeb, které nám toto umožní. V jádru jsou všechny podobné, lišit se mohou malými detaily. Některé jsou placené, jiné zdarma, avšak s omezenou funkcí.

### 3.8.1 ThingSpeak

ThingSpeak je cloudová služba od společnosti MathWorks, vytvořená speciálně pro sběr dat ze zařízení Internetu věcí [25]. V základní verzi je zdarma, ačkoliv některé funkce jsou omezené. Naměřena data lze jednoduše vizualizovat a díky tomu, že je propojena se službou MATLAB, lze vytvářet pokročilé analytické funkce. Po založení účtu na tomto portálu lze vytvořit veřejné či soukromé kanály, přístupné skrze webové rozhraní a API, do kterých můžeme ukládat data.

### 3.8.2 Thinger.io

Thinger.io, podobně jako ThingSpeak, je nabízen v placené i neplacené verzi [26]. V neplacené verzi lze připojit pouze 2 zařízení. V placených verzích se tento počet zvyšuje. Připojit se k této platformě lze pomocí speciální knihovny vytvořené pro usnadnění práce s navazováním spojení. Příkazy z této knihovny poté použijeme v kódu, vyplníme potřebné údaje, jako uživatelské jméno, heslo a ověřovací číslo zařízení, a můžeme sdílet data.

Data lze převádět do různých grafů, tabulek, přepínačů a mnoha dalších podob. Dále lze sledovat, zda je zařízení připojeno, popřípadě jak dlouho a odkud, jestli probíhá komunikace, jaká je velikost datového toku a mnoho dalších užitečných věcí.

### 3.8.3 Azure IoT Suite

Cloudové služby jsou běžně nabízeny i technologickými giganty jako například Microsoft, Google či IBM. Řešení u takto velkých firem nám přináší mnoho výhod. Oproti menším cloudům poskytují například obousměrnou komunikaci, záruku mnohem vyšší kvality, zejména stabilitu dostupných služeb a rychlost odezvy.

Jedním z takových cloudů je Azure IoT Suite [27], nabízený i v neplacené verzi. V této verzi je omezen počet dat, jež můžeme do cloudu poslat. Konkrétně na 8000 odeslaných zpráv na den na jedno zařízení. I zde je pro navázání komunikace využito knihovny, poskytované Microsoftem, která nám značně usnadňuje práci.

Hlavní výhodou oproti cloudům zmíněných výše je možnost obousměrné komunikace mezi uživatelem a zařízením. Zařízení tedy lze ovládat na dálku, což přináší velké spektrum možností dalšího využití. Umožňuje také komunikaci mezi jednotlivými připojenými zařízeními a dovoluje tak vytvářet propojené systémy, kde jedno zařízení ovládá druhé.

### 3.8.4 Vlastní řešení

Jednou z variant je také použití vlastního úložiště. Jedná se například o domácí server či server hostingové služby. Výhodou této možnosti je především fakt, že nejsme v ničem limitováni. Lze připojit velké množství zařízení, upravovat vzhled zobrazovaných dat podle vlastního uvážení či odesílat tolik dat, kolik server zvládne přijmout.

Při použití této varianty lze využít HTTP(S) protokol pro přenos dat z modulu. Po přijetí dat serverem jsou tato data zpracována a uložena. Uložená data lze poté uživateli zobrazit, například pomocí grafů.

Varianta hostingového serveru bude využita v našem případě. Důvodem je především možnost navrhnout si vlastní design zobrazovaných dat. Lze využít vlastních grafů, ve kterých budeme mít možnost vybrat si časový úsek zobrazované oblasti podle naší libosti. Dalším důvodem je také přístup na stránku pomocí námi vybraného doménového jména.

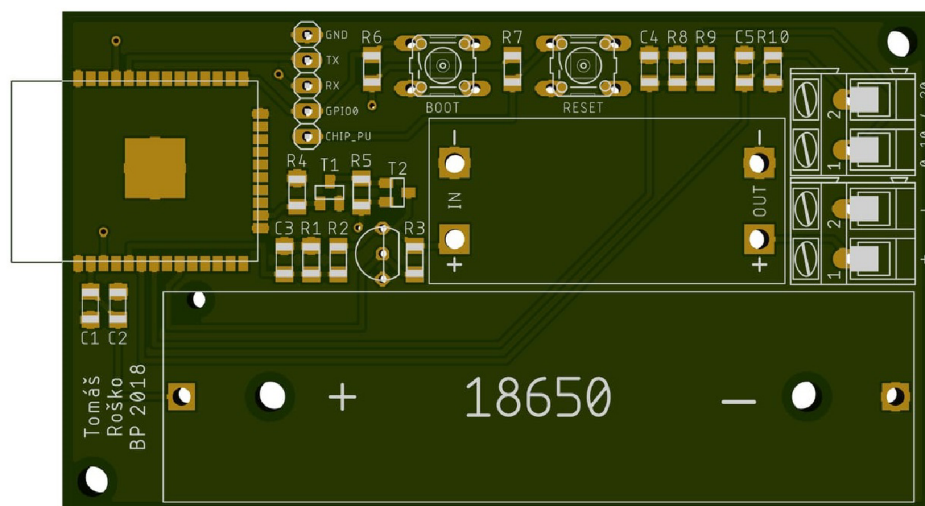
## 4 Praktická realizace

Tato kapitola se zabývá praktickou realizací celého projektu. V prvním oddílu je popsána hardwarová část, tedy to, jak vznikaly jednotlivé díly zařízení, jako je deska plošných spojů a její krytí. Ve druhé části je vysvětleno, jak je celé zařízení naprogramováno, včetně popisu jednotlivých částí kódu. Není také opomenuta webová část, ve které je vysvětleno, jak jsou přijatá data ukládána a zobrazována uživateli.

### 4.1 Hardwarová část

#### 4.1.1 Deska plošných spojů

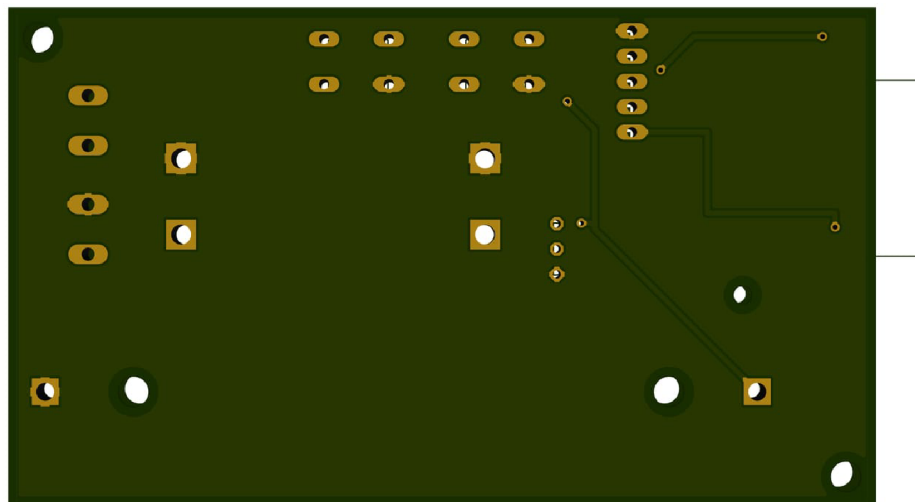
Deska plošných spojů (DPS) byla navržena v programu Eagle s využitím několika dalších knihoven. Jedná se o dvouvrstvou verzi, spoje tedy vedou jak na horní, tak spodní straně desky. Ty jsou propojeny pomocí prokovených děr. Při návrhu desky plošných spojů byl kladen důraz na prostorovou minimalizaci. Po několika úpravách rozmístění prvků na desce má finální deska rozměry  $87 \times 50$  mm (š  $\times$  v). S přičtením délky antény modulu ESP32 délka činí  $93 \times 50$  mm. Finální výrobek však bude větší, jelikož bude deska umístěna do krabičky. Deska byla zhotovena na zakázku v Číně o standardní tloušťce 1,6 mm. Vizualizace horní a dolní strany DPS je zobrazena na obrázku 4.1 a 4.2. Schéma, podle kterého byla deska zhotovena lze vidět v příloze A.1.



Obr. 4.1: Vizualizace horní strany DPS.

Hlavním prvkem je modul ESP32 (WROOM-32), který je umístěn na levé straně tak, aby se anténa modulu nepřekrývala s deskou. K modulu je na pin 3V3 připojena baterie, která na desce zabírá nejvíce místa. Tento pin se stará o napájení celého

modulu. K pinu jsou připojeny dva keramické kondenzátory, které slouží pro pokrytí proudových špiček. Ty mohou dosahovat až 0,5 A, například při připojování k Wi-Fi. K zajištění správné funkce modulu je nutné přivést napětí i na pin EN. Tento pin určuje, zda je modul vypnut, nebo zapnut.



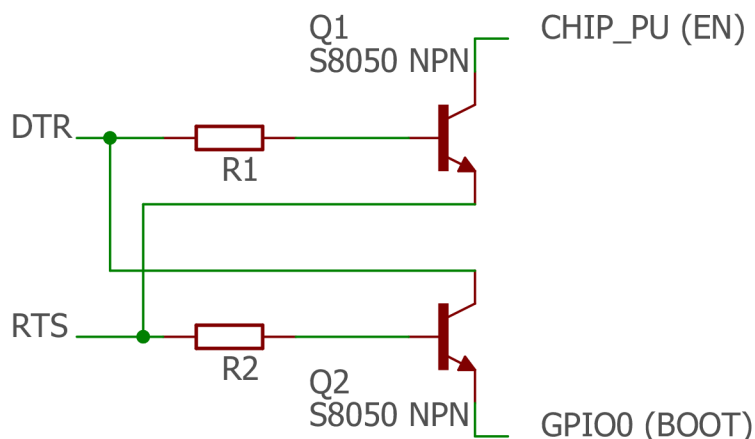
Obr. 4.2: Vizualizace spodní strany DPS.

Další částí obvodu jsou tlačítka. Při jejich zmačknutí dochází k uzemnění daného pinu a tím k vyvolání určité reakce. Při zmačknutí tlačítka RESET (obr. 4.1) dojde k resetování modulu. To může posloužit v případě, kdy uživatel zadá špatné přihlašovací údaje. Může tedy namísto odebrání a opětovného vložení baterie využít právě tohoto tlačítka.

Tlačítko RESET lze také využít v kombinaci s tlačítkem BOOT k nahrání programu do modulu. Postupuje se tak, že se zmačkne tlačítko RESET a následně se zmačkne a pustí tlačítko BOOT. Poté se pustí i tlačítko RESET. Tím se modul uvede do tzv. stahovacího módu, ve kterém lze firmware nahrát.

Program lze nahrát pomocí sériové linky. Z tohoto důvodu jsou na desce vyvedeny piny pro připojení převodníku, který zároveň může sloužit jako prostředek pro zadávání parametrů uživatelem, viz sekce 3.6. Modul má vyvedené piny GND, RX, TX. Pin RX modulu ESP32 se připojuje k převodníku na pin TX, stejně jako TX na RX. Na desce jsou vytisknuty popisy pinů tak, jak se mají připojit k převodníku, tedy pin TX se připojí na pin TX, RX na RX.

Piny GPIO0 (BOOT) a CHIP\_PU (EN) slouží k automatickému nahrávání programu do modulu. To znamená, že není potřeba mačkat tlačítko EN a BOOT, jak je uvedeno výše. K možnosti využít automatické nahrávání je však potřeba připojit externí obvod, který není součástí desky. Toto zapojení lze vidět na obrázku 4.3. Vyvedení těchto pinů sloužilo hlavně k většímu pohodlí při programování a ladění modulu.



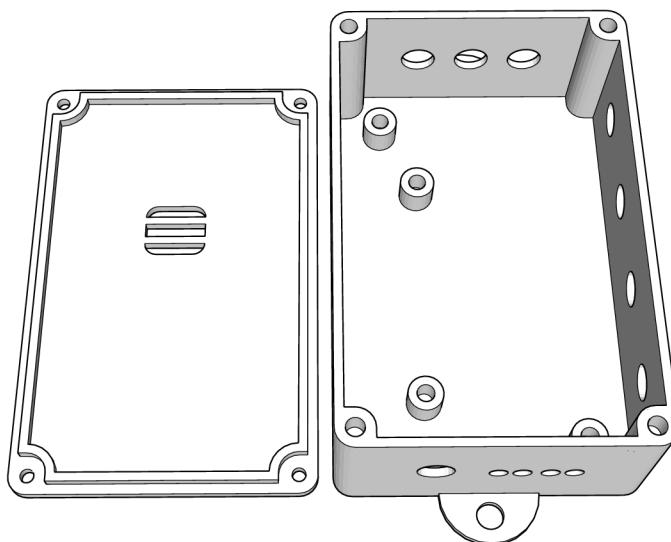
Obr. 4.3: Obvod sloužící k automatickému nahrávání firmwaru.

Další částí obvodu je spínací obvod, který lze spínat pomocí GPIO pinu a tím napájet potřebné části obvodu pouze v době, kdy má zařízení měřit. Hlavním důvodem použití tohoto obvodu bylo šetření kapacity baterie v případě měření napětí na baterii. Nakonec se však ukázalo jako ideální varianta použít toto zapojení i pro ostatní prvky, které se spínají jednou za určitou časovou jednotku (teplotní čidlo, napěťový měnič MT3608).

#### 4.1.2 Krytí DPS

Desku plošných spojů je potřeba umístit do vhodného krytí. Na trhu se dají najít krabičky různého tvaru a různých materiálů, avšak nejlepším možným řešením je vymodelovat a vytvořit si krabičku svou. Při modelování je potřeba myslet na to, že prostor uvnitř krabičky by měl být přizpůsoben velikosti desky, aby nedošlo ke zbytečnému zvětšení celého zařízení. V této krabičce je také potřeba navrhnout uchycení pro DPS a držák baterie. Důležitým faktem také je, že zařízení měří teplotu a mělo by mít možnost připojit další průmyslová čidla. Proto je nutné do krabičky vymodelovat několik děr. Tyto díry sice zamezí tomu, aby zařízení mohlo být vystavené lehkému dešti, avšak modul by neměl být vystaven přímému slunečnímu záření, aby nedocházelo ke zkreslení měřené teploty. Lze proto předpokládat, že krabička bude schována pod nějakým přístřeškem.

K modelování byl použit program SketchUp od společnosti Trimble. Celý model je složen ze dvou částí, a to z krabičky a víka. Vizualizaci modelu lze vidět na obrázku 4.4. Uchycení víka, desky a držáku baterie je realizováno pomocí M3 šroubků a závitů, které byly do plastu zataveny, aby se zabránilo jejich protáčení. Ke spodní části krabičky byly také domodelovány dva úchyty sloužící ke snadnějšímu přichycení na zed.



Obr. 4.4: Vizualizace modelu krabičky a víka.

Model byl vytištěn na 3D tiskárně na Masarykově univerzitě pomocí tiskárny BE3D EDEE. Je vyroben z bílého PLA filamentu certifikovaného firmou Y Soft. Celková doba tisku činila přibližně 9 hodin. Velikost vytisknutého modelu včetně víka a držáků činí přibližně  $127,5 \times 37,5 \times 68,5$  mm (š × v × h).

## 4.2 Softwarová část

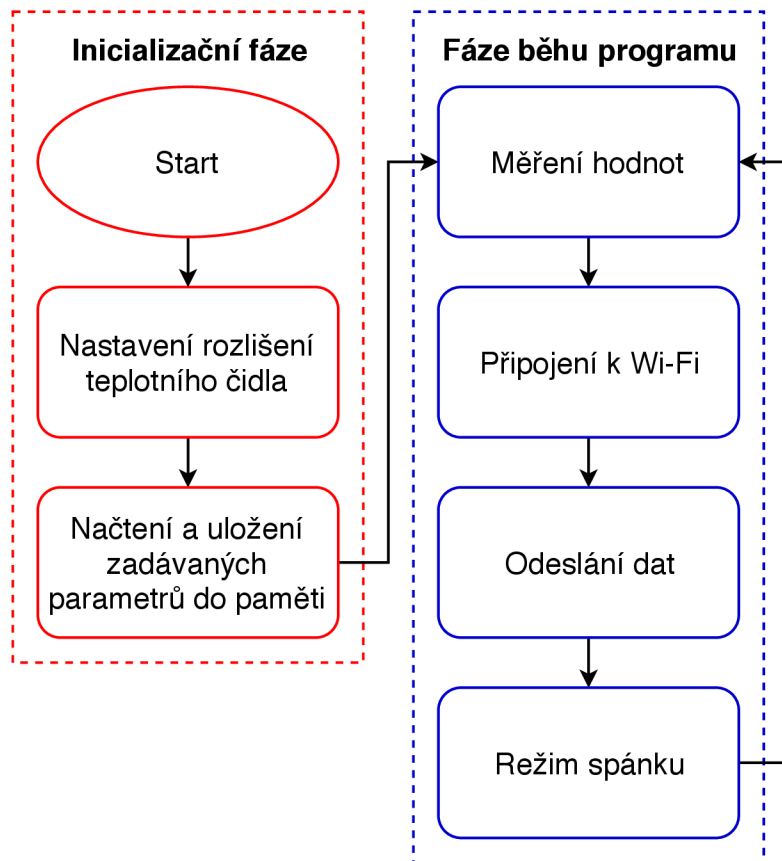
### 4.2.1 Programování modulu ESP32

Program pro modul ESP32 byl vytvořen pomocí editoru Atom, do kterého bylo přidáno vývojové prostředí PlatformIO. Jedná se o vývojové prostředí, které je speciálně určeno pro vývoj zařízení Internetu věcí. Mezi jeho hlavní výhody patří například jednoduchá správa knihoven, našeptávání a dokončování příkazů či možnost výběru desky, pro kterou chceme vytvořený program zkompilovat. Při programování zařízení bylo na výběr z několika frameworků<sup>1</sup>. V našem případě byl zvolen framework Arduino, a to především kvůli jeho dobré technické dokumentaci a jednoduchosti. Celý kód se skládá ze dvou hlavních částí, viz obrázek 4.5.

První z nich je část inicializační, která se provede pouze při prvním spuštění zařízení. Tato část má za úkol nastavit rozlišení pro teplotní čidlo na 9 bitů, jak již bylo zmíněno v části 3.3.1. Dalším krokem je načtení parametrů zadávaných uživatelem a jejich uložení do paměti zařízení. Druhá část kódu se stará o měření dat, jejich zpracování a odeslání. Na rozdíl od první části kódu je vykonávána při každém běhu zařízení. Obsahuje následující funkce:

<sup>1</sup>Softwarová struktura sloužící pro usnadnění programování.





Obr. 4.5: Logická struktura programu.

### Měření hodnot – `getValues()`

V této funkci probíhá měření dat. Pro zajištění správného měření je nejprve nutné přivést napětí ke všem částem obvodu. To je docíleno sepnutím tranzistorů pomocí GPIO pinu modulu ESP32. Po přivedení napětí je nutné počkat alespoň půl sekundy, než se napětí ustálí. Po ustálení napětí je nejprve změřena teplota a následně všechny analogové hodnoty, které jsou měřeny pomocí speciální kalibrační funkce, jež zaručuje jejich větší přesnost. Naměřené analogové hodnoty jsou také vhodně upraveny před samotným odesláním (vynásobení poměrem děliče, nastavení nuly). V případě naměření hodnoty baterie menší než 2,8 V se ve vykonávání další části kódu nepokračuje a dojde k uspání celého zařízení na dobu neurčitou.

### Připojení k Wi-Fi – `connectToWiFi()`

Tato funkce má za úkol připojit zařízení k Wi-Fi. V případě připojování, které trvá déle než 5 sekund, dojde k resetování zařízení a opětovnému pokusu o připojení. Pokud se stane, že se opětovné připojení nepovede více než na třetí pokus, je zařízení na jednu hodinu uspáno. Toto řešení je využito hlavně z důvodu úspory baterie v případě, že by došlo k výpadku Wi-Fi routeru.

## Odeslání dat – sendValues()

Po úspěšném připojení k Wi-Fi je potřeba naměřená data odeslat. V našem případě jsou data odesílána využitím protokolu HTTP v jeho šifrované verzi (HTTPS). To je zajištěno pomocí ověření certifikátu certifikační autority daného serveru. Samotná funkce také testuje, zda se povedlo úspěšně připojit k serveru. V případě úspěchu jsou naměřená data předána serveru HTTP metodou POST. V případě neúspěšného připojení se naměřená data neodesílají.

Po provedení těchto funkcí je zařízení uspáno na dobu zvolenou uživatelem. Po uplynutí této doby dojde k jeho probuzení a opětovnému provedení výše zmíněných funkcí.

### 4.2.2 Webové rozhraní

Jak již bylo uvedeno v části 3.8.4, v tomto projektu je využit vlastní hostingový server, který lze nalézt na adrese [www.homesp.cz](http://www.homesp.cz). Po přijetí dat, jsou data zpracována a uložena do MySQL databáze, která je součástí hostingového serveru. K přijatým hodnotám je také přidáno časové razítko, které zaznamenává čas přijetí (měření) dat. Struktura tabulky v MySQL databázi je zobrazena v tabulce 4.1. Hodnota „id“ má nastaven primární klíč, to znamená, že nemůžou existovat dva záznamy se stejnou hodnotou „id“.

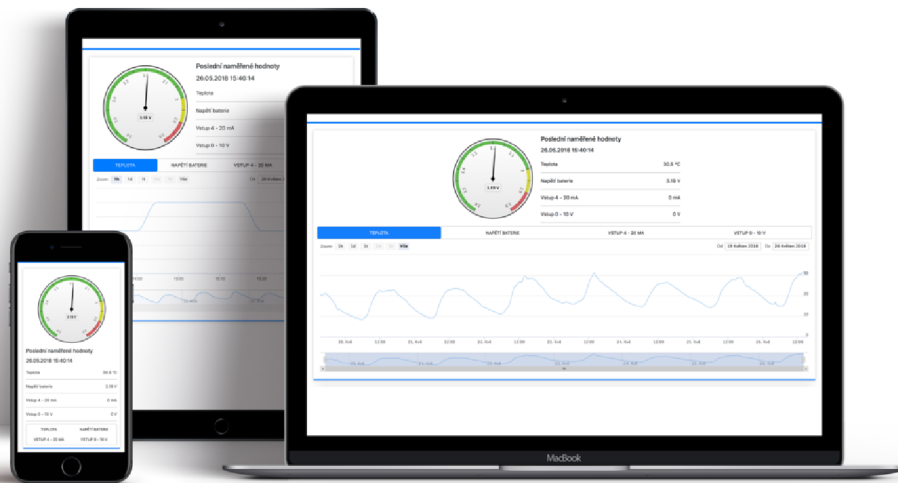
Tab. 4.1: Struktura tabulky v MySQL databázi.

esp32Log			
Název	Typ	Výchozí	Další
id	int(32)	Žádná	AUTO_INCREMENT
temperature	float	Žádná	
battery	float	Žádná	
current	float	Žádná	
voltage	float	Žádná	
timestamp	timestamp	CURRENT_TIMESTAMP	

Po zadání adresy serveru do prohlížeče je uživateli zobrazen obsah souboru `index.php`, ve kterém se nachází HTML předpis, podle kterého je dána stránka zobrazena. V tomto souboru jsou také načteny všechny potřebné knihovny a styly pro správný chod celé stránky, jako například JavaScriptová knihovna Highcharts (Highstock), která slouží k vytvoření zobrazovaných grafů. Aby mohla být data v grafech zobrazena, je potřeba je vhodně naformátovat do podoby, které knihovna rozumí. O to se starají PHP skripty v souboru `functions.php`.

Stránka je rozdělena na dvě části. V horní části se nachází tabulka a graf, který slouží pro rychlý přehled o stavu baterie a je barevně rozdělen tak, aby uživatel věděl, kdy se baterie blíží do stavu vybití. Tento stav udává červená barva v grafu. V tabulce napravo lze vidět všechny naměřené hodnoty při posledním měření. Pod horní částí se nachází čtyři tlačítka, která po kliknutí zobrazí graf podle zvoleného výběru. V grafech lze vidět změny jednotlivých hodnot v závislosti na čase. Pro lepší orientaci v grafu existuje také možnost vybírat libovolné časové období.

Celá stránka je responzivní, přizpůsobuje tedy svůj obsah podle toho, na jakém zařízení je zobrazována. Výslednou podobu stránky na různých zařízeních lze vidět na obrázku 4.6.



Obr. 4.6: Výsledná podoba stránky na různých zařízeních.

## 5 Testování a cenová kalkulace

V této kapitole jsou popsány způsoby a výsledky testování funkčnosti zařízení, včetně jeho energetické náročnosti a přesnějšího výpočtu výdrže baterie na jeden nabíjecí cyklus. Dále je zde shrnuta finanční stránka projektu, jelikož jedním z nepsaných cílů práce bylo vytvořit zařízení co nejlevněji.

### 5.1 Testování funkčnosti finálního produktu

Testování probíhalo ve dvou fázích. První fází bylo měření reálné spotřeby zařízení. To ukázalo, že odběr zařízení se pohybuje okolo  $5 \mu\text{A}$  v době, kdy je zařízení v režimu spánku. Hodnota spotřeby zařízení v režimu měření byla zprůměrována na hodnotu  $130 \text{ mA}$ , a to z důvodu, že zařízení mění svoji spotřebu podle toho, kterou část kódu zrovna vykonává (například pouze zapnuté zařízení má menší spotřebu, když hodnoty měří, než když naměřené hodnoty odesílá). Celková doba běhu zařízení se pohybuje kolem  $2500 \text{ ms}$  ( $2,5 \text{ s}$ ). Tento časový údaj se však může občas lišit, ne vždy se totiž povede připojit k Wi-Fi ve stejném časovém úseku. Toto platí i pro čas potřebný k odeslání dat.

Se znalostí těchto hodnot lze vypočítat výdrž baterie zařízení přesněji, než jen s teoretickými znalostmi. Pokud zařízení bude měřit co půl hodiny, dosazením do vztahů 3.1 a 3.2 nám vyjde, že zařízení by mělo vydržet přibližně 344 dní, což je hodnota o 11 % vyšší než podle prvních výpočtů.

Druhá fáze spočívala ve zjišťování přesnosti měřených hodnot. Měření probíhalo jeden týden, v období od 19. do 26. května 2018. Po zadání přihlašovacích údajů k Wi-Fi a nastavení časové periodicity měření, která byla zvolena na 5 minut, bylo zařízení umístěno pod stříšku pergoly domu. Měření referenční teploty, od které se námi měřená teplota posuzovala, bylo prováděno pomocí teplotního čidla od firmy Siemens, které pomáhá řídit domácí kotel (obr. 5.1). Odchylka činila přibližně  $0,5 \text{ }^\circ\text{C}$ . Teplota byla také srovnávána s měřenými hodnotami volně dostupných portálů pro předpovědi počasí. V tomto případě byla odchylka o něco větší, přibližně  $+2 \text{ }^\circ\text{C}$ . Je však třeba brát v potaz, že portály ukazovaly teplotu v jiném místě, než ve kterém probíhá naše měření. Dalším důvodem může být, že zařízení, ani teplotní čidlo kotle, není umístěno tak, jak by správně měly být umístěny čidla pro měření teploty (2 metry ve výšce ve speciálním zákrytu). Zkreslení také mohlo způsobovat sluneční záření, které ohřívalo černou střechu pergoly, pod kterou se v její těsné blízkosti nacházelo samotné zařízení. V neposlední řadě může být zdrojem nepřesnosti samotný čip ESP32, který se může v určitých chvílích více zahřívat.

V případě měření přesnosti AD převodníků (měření baterie, vstupu  $4\text{-}20 \text{ mA}$  a vstupu  $0\text{-}10 \text{ V}$ ) činila odchylka maximálně  $\pm 3 \%$ . Této přesnosti je dosaženo díky



Obr. 5.1: Umístění zařízení při testovacím měření.

použití kalibrační funkce a paralelnímu umístění kondenzátorů ke měřeným rezistorům. Měření bylo prováděno samostatně, před umístěním zařízení ven k dalšímu testování. Velikost měřených hodnot byla regulována pomocí potenciometru, kterým byl regulován proud, případně napětí, které bylo měřeno na daných rezistorech.

Za celou dobu měření se stalo pouze několikrát, že data nebyla odeslána. Rozestupy mezi těmito výpadky byly však maximálně 20 minut. Lze tedy předpokládat, že došlo k výpadkům serveru, než ke špatnému připojení k Wi-Fi. Pokud by totiž došlo k výpadku Wi-Fi, zařízení by se na hodinu uspalo, proto by byl rozestup větší.

## 5.2 Cenová kalkulace

Jedním z důležitých parametrů zařízení Internetu věcí je také jeho cena. Celkové náklady na výrobu finálního produktu se v našem případě skládají z několika položek:

- Cena jednotlivých komponent.
- Náklady na výrobu DPS.
- Náklady na tisk 3D modelu.

V tabulce 5.1 lze vidět ceny za jednotlivé komponenty. V příloze A.2 jsou jednotlivé komponenty popsány detailněji. Uvedené ceny jsou kombinovány jak z českých, tak zahraničních e-shopů, tak aby výsledná cena za kus byla co nejmenší. Ceny jsou udávány včetně DPH.

Celková cena za komponenty k DPS činí **341 Kč**. K této ceně je potřeba připočítat

Tab. 5.1: Cenová kalkulace jednotlivých komponent.

Typ	Označení	Cena/ks [Kč]	Cena celkem [Kč]
Rezistor	R1, R2, R4, R5, R6, R7	1	6
Rezistor	R3	1	1
Rezistor	R8	1	1
Rezistor	R9	1	1
Rezistor	R10	1	1
Ker. kondenzátor	C1	10	10
Ker. kondenzátor	C2	2	2
Ker. kondenzátor	C3, C4, C5	1	3
MOSFET N	T1	4	4
MOSFET P	T2	4	4
Mikrospínač	S1, S2	2	4
Baterie		120	120
Držák baterie		23	23
Teplotní čidlo		35	35
Měníč napětí		30	30
ESP32		80	80
Svorkovnice		7	14
Oboustranné kolíky		2	2
			<b>341</b>

ještě cenu za výrobu DPS a cenu za výrobu 3D modelu krabičky a víka. Jak bylo uvedeno v části 4.1.1, deska byla v našem případě vyrobena v Číně. Cena činila 5 \$ při minimálním odběru 10 kusů. Jeden kus tedy vychází v přepočtu na **11 Kč**. Cena za 3D tisk byla přibližně **150 Kč**.

Celková cena finálního produktu je tedy **502 Kč**. Mohla by však být mnohem nižší, například pokud by byly jednotlivé komponenty kupovány ve větším množství. K ceně není připočteno poštovné, které variuje nejen v závislosti na zemi původu, ale i na rychlosti doručení a ceně objednávaných komponent.

## 6 Závěr

Cílem této práce bylo navrhnout řešení pro sestavení teplotního senzoru s Wi-Fi modulem ESP32 včetně jeho praktické realizace. Důraz byl kladen především na co nejmenší spotřebu kapacity baterie a možnost připojit jakékoliv průmyslové čidlo fungující na principu analogového přenosu 4-20 mA a 0-10 V. Dalším cílem práce bylo navrhnout vhodné ukládání měřených dat, preferovaně pomocí cloudu, a vytvořit webové rozhraní, které tato data pohodlně zobrazí koncovému uživateli.

První část práce představila rozvíjející se oblast Internetu věcí, včetně výčtu možností uplatnění jednotlivých senzorů. Teoretická část se také věnovala nejčastějším komunikačním technologiím a protokolům. Druhá kapitola popisuje firmu Espressif Systems a popisuje dva jejich nejpoužívanější moduly ESP8266 a ESP32, ze kterých je ESP32 použito v našem řešení. Třetí kapitola práce se zabývá specifikací požadavků a celkovému návrhu zařízení. Diskutuje také jednotlivé součásti zařízení a problémy s nimi spojené, jako je například zhodnocení nejideálnějšího řešení pro napájení zařízení. Probrány jsou také dostupné možnosti pro ukládání a prezentaci dat. Ve čtvrté kapitole dochází k praktické realizaci modulu a to jak po hardwarové, tak softwarové stránce. V hardwarové části je popsána realizace desky plošných spojů a jejího krytí, které bylo zhotovené pomocí 3D tisku z plastového filamentu. Softwarová část představuje, jak bylo zařízení naprogramováno, včetně popisu jednotlivých funkcí. V této části je také představeno webové rozhraní sloužící koncovému uživateli k zobrazování měřených dat. Následovné testování modulu je popsáno v závěrečné páté kapitole. Testování probíhalo nepřetržitě po dobu jednoho týdne a ukázalo, že navržený produkt je plně funkční. Odhadovaná odchylka měření teploty je 0,5 °C, v případě analogových hodnot  $\pm 3\%$ . Modul by měl vydržet až 344 dnů na jeden nabíjecí cyklus baterie. Celková cena finálního produktu bez poštovného pro dodání jednotlivých komponent zařízení činí 502 Kč.

Do budoucna lze zařízení rozšířit o centrální prvek, ke kterému by bylo připojeno více takovýchto zařízení. Centrální prvek by naměřená data z připojených senzorů sbíral a zpracovával je. Na základě toho by mohl vyvolávat různé akce. Připojená zařízení k centrálnímu prvku by také mohla přenášet naměřená data efektivnějším způsobem vzhledem k energetické náročnosti.

# Literatura

- [1] HANES, David a kol. *IoT fundamentals: networking technologies, protocols, and use cases for the Internet of things*. Indianapolis: Cisco press, 2017. ISBN 1-58714-456-5.
- [2] Postscapes. *History of Internet of Things* [online]. 2016 [cit. 2017-11-15]. Dostupné z: <<https://www.postscapes.com/internet-of-things-history>>.
- [3] Kelton. *Internet of Things Timeline* [online]. 2014 [cit. 2017-11-15]. Dostupné z: <<http://internetofthings.keltonglobal.com>>.
- [4] The Future of Smart. *The First IoT Device* [online]. 2014 [cit. 2017-11-15]. Dostupné z: <<https://thefutureofsmart.wordpress.com/2014/10/04/the-first-iot-device-a-coke-machine>>.
- [5] FOOTE, Keith. *A Brief History of the Internet of Things* [online]. 2016 [cit. 2017-11-15]. Dostupné z: <<http://www.dataversity.net/brief-history-internet-things>>.
- [6] Postscapes. *Internet Things Examples* [online]. 2016 [cit. 2017-11-19]. Dostupné z: <<https://www.postscapes.com/internet-of-things-examples>>.
- [7] Google Trends. *Zájem o Internet věcí během posledních 10 let* [online]. Kalifornie: Google, 2017 [cit. 2017-11-17]. Dostupné z: <<https://trends.google.com/trends/explore?date=2007-01-01%202017-11-30&q=Internet%20of%20Things>>.
- [8] Hewlett-Packard Enterprise. *The Internet of Things: Today and Tomorrow* [online]. 2016 [cit. 2017-11-19]. Dostupné z: <[http://www.arubanetworks.com/assets/eo/HPE\\_Aruba\\_IoT\\_Research\\_Report.pdf](http://www.arubanetworks.com/assets/eo/HPE_Aruba_IoT_Research_Report.pdf)>.
- [9] RUSSELL, Brian a Drew VAN DUREN. *Practical Internet of Things Security*. Birmingham: Packt Publishing, 2016. ISBN 978-1-78588-963-9.
- [10] Wikipedia, The Free Encyclopedia. *Wi-Fi* [online]. 2017 [cit. 2017-11-23]. Dostupné z: <<https://cs.wikipedia.org/wiki/Wi-Fi>>.
- [11] TALLA, Vamsi a kol. *Powering the Next Billion Devices with Wi-Fi* [online]. 2015 [cit. 2017-11-23]. Dostupné z: <<https://homes.cs.washington.edu/~gshyam/Papers/powifi.pdf>>.
- [12] RAY, Brian. *What is LTE-M?* [online]. 2017 [cit. 2017-11-23]. Dostupné z: <<https://www.link-labs.com/blog/what-is-lte-m>>.



- [13] Wikipedia, The Free Encyclopedia. *Bluetooth* [online]. 2017 [cit. 2017-11-23]. Dostupné z: <<https://en.wikipedia.org/wiki/Bluetooth>>.
- [14] ANKUR, Tomar. *Introduction to Zigbee Technology* [online]. 2011 [cit. 2017-11-23]. Dostupné z: <<http://www.cs.odu.edu/~cs752/papers/zigbee-001.pdf>>.
- [15] GRAZIER, Mark. *The future of home automation – ZigBee or Z-Wave?* [online]. 2014 [cit. 2017-11-23]. Dostupné z: <<https://www.edn.com/Pdf/ViewPdf?contentItemId=4426050>>.
- [16] WAHER, Peter. *Learning Internet of Things*. Birmingham: Packt Publishing, 2015. ISBN 978-1783553532.
- [17] CONSTINE, John. *Facebook Messenger Day hits 70M daily users as the app reaches 1.3B monthlies* [online]. 2017 [cit. 2017-11-29]. Dostupné z: <<https://techcrunch.com/2017/09/14/facebook-messenger-1-3-billion>>.
- [18] Espressif Systems. *Our Company* [online]. Shanghai: Espressif Systems, 2017 [cit. 2017-12-03]. Dostupné z: <<http://espressif.com/en/company>>.
- [19] Espressif Systems IOT Team. *ESP-WROOM-02 Datasheet* [online]. 2017 [cit. 2017-12-03]. Dostupné z: <[http://espressif.com/sites/default/files/documentation/0c-esp-wroom-02\\_datasheet\\_en.pdf](http://espressif.com/sites/default/files/documentation/0c-esp-wroom-02_datasheet_en.pdf)>.
- [20] MIKOLAJCZAK, Celina a kol. *Lithium-Ion Batteries Hazard and Use Assessment* [online]. 2011 [cit. 2017-12-05]. Dostupné z: <[http://www.prba.org/wp-content/uploads/Exponent\\_Report\\_for\\_NFPA\\_-\\_20111.pdf](http://www.prba.org/wp-content/uploads/Exponent_Report_for_NFPA_-_20111.pdf)>.
- [21] JENSEN, Henrik. *Soshine 18650 LiFePO<sub>4</sub> 1800mAh (Black)* [online]. 2012 [cit. 2018-04-15]. Dostupné z: <[https://lygte-info.dk/review/batteries2012/Soshine%2018650%20LiFePO4%201800mAh%20\(Black\)%20UK.html](https://lygte-info.dk/review/batteries2012/Soshine%2018650%20LiFePO4%201800mAh%20(Black)%20UK.html)>.
- [22] Maxim Integrated Products. *Programmable Resolution 1-Wire Digital Thermometer DS18B20 Datasheet* [online]. 2015 [cit. 2017-12-09]. Dostupné z: <<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>>.
- [23] Aerosemi Technology. *High Efficiency Step Up Converter MT3608 Datasheet* [online]. 2011 [cit. 2017-12-10]. Dostupné z: <<https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf>>.
- [24] WaveShare. *WaveShare CP2102 USB UART Board (type A)* [online]. Shenzhen: WaveShare, 2017 [cit. 2017-12-11]. Dostupné z: <<https://www.waveshare.com/cp2102-usb-uart-board-type-a.htm>>.

- [25] ThingSpeak. *Learn More About ThingSpeak* [online]. Natick: MathWorks, 2017 [cit. 2017-12-12]. Dostupné z: <[https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more)>.
- [26] Thingier.io Platform. *About Thingier* [online]. Madrid: Thingier, 2017 [cit. 2017-12-12]. Dostupné z: <<https://thingier.io>>.
- [27] Microsoft Azure IoT Suite. *About Microsoft Azure IoT Suite* [online]. Redmond: Microsoft, 2017 [cit. 2017-12-12]. Dostupné z: <<https://www.azureiotsuite.com>>.

## Seznam zkratek

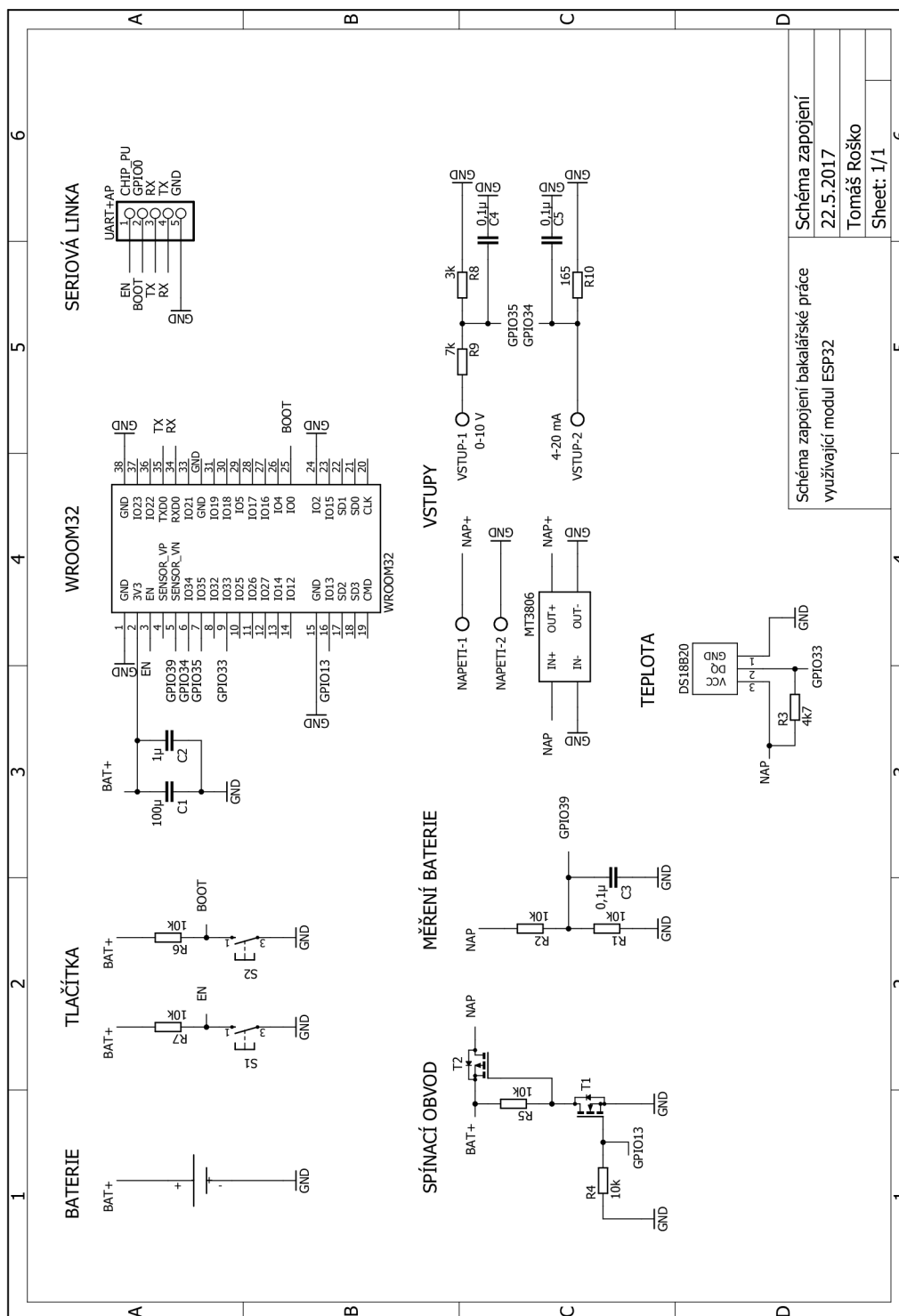
<b>ADC</b>	Analog-to-Digital Converter
<b>API</b>	Application Programming Interface
<b>ARPANET</b>	Advanced Research Projects Agency NETwork
<b>BLE</b>	Bluetooth Low Energy
<b>CoAP</b>	Constrained Application Protocol
<b>DPS</b>	Deska tištěných spojů
<b>GPIO</b>	General-Purpose Input/Output
<b>HTTP(S)</b>	Hypertext Transfer Protocol (Secure)
<b>I<sup>2</sup>C</b>	Inter-Integrated Circuit
<b>IEEE</b>	Electrical and Electronics Engineers
<b>IoT</b>	Internet of Things
<b>IP</b>	Internet Protocol
<b>LTE-M</b>	Long Term Evolution Category M1
<b>MQTT</b>	Message Queuing Telemetry Transport
<b>PLA</b>	Polylactic Acid – kyselina polymléčná
<b>PoWi-Fi</b>	Power over Wi-Fi
<b>SSL</b>	Secure Sockets Layer
<b>TCP</b>	Transmission Control Protocol
<b>TLS</b>	Transport Layer Security
<b>UART</b>	Universal Asynchronous Receiver-Transmitter
<b>UDP</b>	User Datagram Protocol
<b>ULP</b>	Ultra Low Power
<b>USB</b>	Universal Serial Bus
<b>Wi-Fi</b>	Wireless Fidelity

# Seznam příloh

<b>A</b>	<b>Návrh zařízení</b>	<b>48</b>
A.1	Celkové schéma zapojení . . . . .	48
A.2	Seznam použitých komponent . . . . .	49
<b>B</b>	<b>Fotografie projektu</b>	<b>50</b>
<b>C</b>	<b>Přiložené médium</b>	<b>52</b>
C.1	Struktura obsahu přiloženého média . . . . .	52

# A Návrh zařízení

## A.1 Celkové schéma zapojení



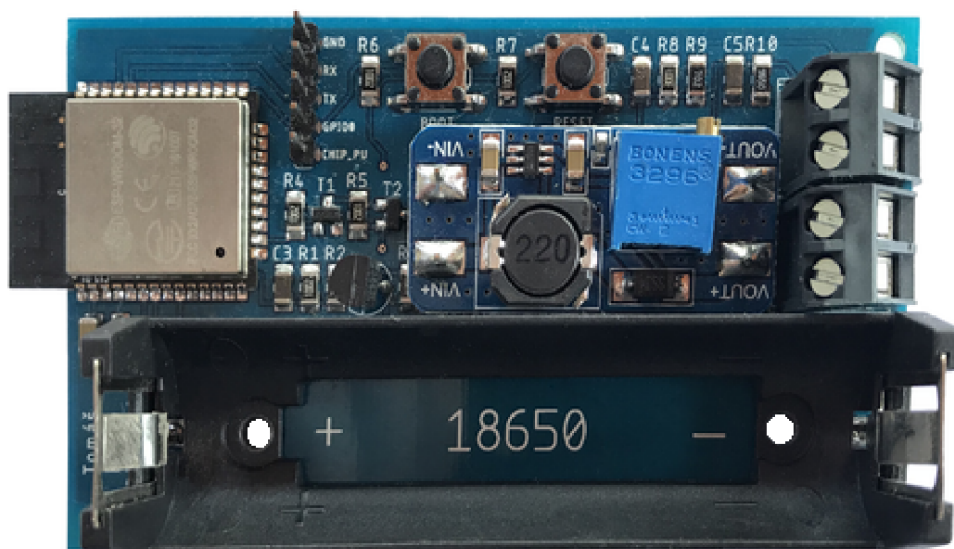
Obr. A.1: Celkové schéma zapojení.

## A.2 Seznam použitých komponent

Tab. A.1: Seznam použitých komponent.

Typ	Hodnota	Označení
Rezistor	10 k $\Omega$ 0,25 W 1 % 1206	R1, R2, R4, R5, R6, R7
Rezistor	4700 $\Omega$ 0,25 W 1 % 1206	R3
Rezistor	3 k $\Omega$ 0,25 W 1 % 1206	R8
Rezistor	7 k $\Omega$ 0,25 W 1 % 1206	R9
Rezistor	165 $\Omega$ 0,25 W 1 % 1206	R10
Ker. kondenzátor	100 $\mu$ F 6,3 V X5R 20 % 1206	C1
Ker. kondenzátor	1 $\mu$ F 50 V X7R 20 % 1206	C2
Ker. kondenzátor	0,1 $\mu$ F 50 V X7R 10 % 1206	C3, C4, C5
MOSFET N	BVSS123LT1G SOT-23	T1
MOSFET P	NDS356AP SOT-23	T2
Mikrospínač	TC-0102-T	S1, S2
Baterie	Soshine 18650 1800 mAh	
Držák baterie	COMF BHC-18650-1P	
Teplotní čidlo	DS18B20 TO-92	
Měnič napětí	MT3608	
ESP32	WROOM-32 rev. 1	
Svorkovnice	2 $\times$ PTR AK300/2-5.0	
Oboustranné kolíky	S1G05 2,54 mm	

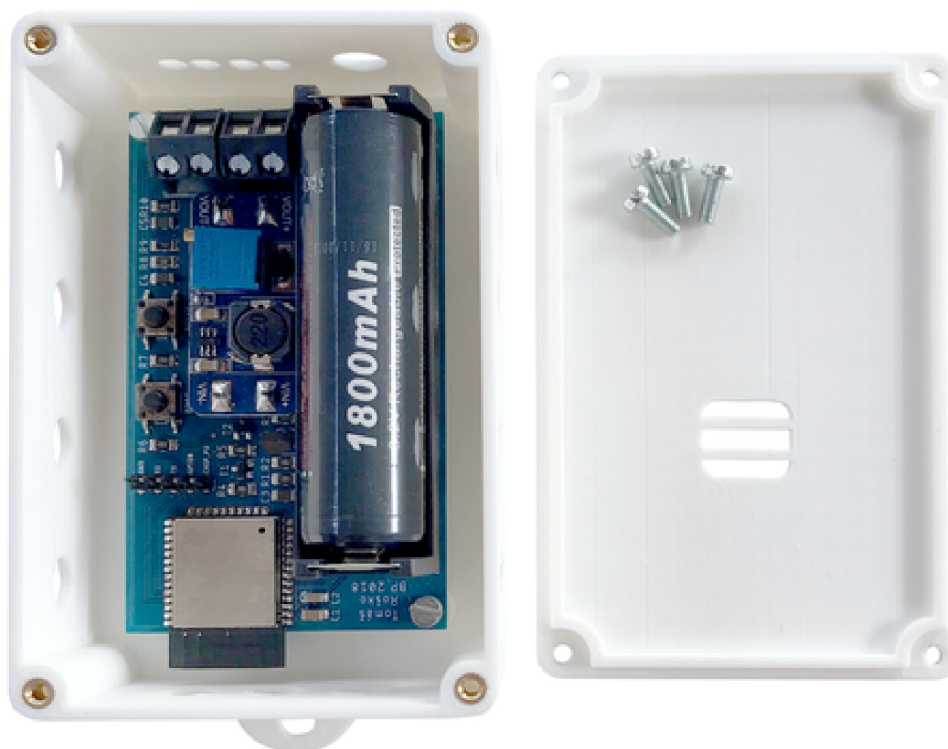
## B Fotografie projektu



Obr. B.1: Osazená deska tištěných spojů.



Obr. B.2: Vytisknutý model krabičky s víkem se zatavenými závity.



Obr. B.3: Fotografie finálního produktu.



# C Přiložené médium

## C.1 Struktura obsahu přiloženého média

```
/. ..... Kořenový adresář přiloženého CD
├── Fotografie projektu
│   ├── deska.png
│   ├── finalni_produktn.png
│   └── modely.png
├── Hardwarová část
│   ├── Deska plošných spojů ..... Eagle 8.4.1
│   │   ├── DPS.brd ..... Deska tištěných spojů
│   │   ├── DPS.sch ..... Schéma zapojení
│   │   ├── libraries.zip ..... Knihovny pro Eagle
│   │   └── schema_zapojeni.pdf
│   ├── Modely ..... SketchUp Pro 2016
│   │   ├── krabicka.skp
│   │   └── viko.skp
├── Softwarová část
│   ├── Program ..... PlatformIO 3.5.3rc1 – Atom 1.27.0
│   │   ├── main.txt ..... Hlavní část kódu ve formátu txt
│   │   └── projekt.zip ..... Vyexportovaný projekt s pokyny ke spuštění
│   ├── Webové rozhraní
│   │   └── web.zip ..... Soubory k webové aplikaci
```