



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**SYSTÉM PRO VZDÁLENÝ MONITORING A SPRÁVU  
ZAHRADNÍCH BAZÉNŮ**

SYSTEM FOR REMOTE MONITORING AND MANAGEMENT OF GARDEN POOLS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MAREK ŠŤASTNÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÁCLAV ŠIMEK,**

BRNO 2022

## Zadání bakalářské práce



Student: **Šťastný Marek**  
Program: Informační technologie  
Název: **Systém pro vzdálený monitoring a správu zahradních bazénů**  
**System for Remote Monitoring and Management of Garden Pools**  
Kategorie: Vestavěné systémy

### Zadání:

1. Připravte stručnou rešerši existujících IoT systémů pro monitoring zahradních bazénů. Dále prostudujte aktuálně používané IoT protokoly v této oblasti.
2. Zpracujte stručný přehled dostupných komunikačních modulů Bluetooth a WiFi, které jsou vhodné pro napájení z baterie.
3. Na základě poznatků z bodů 1) a 2) navrhnete architekturu modulárního systému pro komplexní monitoring a obsluhu bazénu.
4. Zvolte vhodné komponenty pro jednotlivé moduly dle vytvořeného návrhu. Proveďte realizaci modulů na obvodové úrovni včetně základního oživení.
5. Implementujte obslužný firmware zajišťující požadovanou funkcionalitu jednotlivých modulů systému a to včetně komunikace s řídicí jednotkou.
6. Implementujte grafické uživatelské rozhraní umožňující zobrazování informací o stavu systému a konfiguraci jeho funkčních prvků.
7. Ověřte funkčnost realizovaného řešení. Zhodnoťte dosažené výsledky a pokuste se navrhnout případná vylepšení či budoucí směr vývoje.

### Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**  
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.  
Datum zadání: 1. listopadu 2021  
Datum odevzdání: 11. května 2022  
Datum schválení: 29. října 2021

## Abstrakt

Tato práce se zabývá návrhem a následnou realizací bezdrátového systému pro automatické monitorování a ovládání zahradního bazénu. V práci je čtenář nejprve seznámen se základními měřeními kvality vody. Dále je v této práci věnována pozornost analýze bezdrátových technologií a komunikačních protokolů. Na základě získaných poznatků byl proveden návrh a následná realizace systémových modulů a webové aplikace, kterou je systém řízen. V závěru práce je popsán způsob, jakým byly testovány jednotlivé moduly a ověřena funkčnost celku.

## Abstract

This thesis deals with the design and subsequent implementation of a wireless system for automatic monitoring and control of a garden pool. In this work, the reader is first introduced to basic water quality measurements. Next, the analysis of wireless technologies and communication protocols is analyzed in this thesis. Based on the knowledge gained, the design and subsequent implementation of the system modules and the web application that controls the system has been carried out. The thesis concludes with a description of how the modules were tested and the functionality of the whole system was verified.

## Klíčová slova

automatizace, chytrý bazén, kvalita vody, mikrokontrolér, IoT, PCB, ESP32, ESP-Now, MQTT, Flask, Angular

## Keywords

automation, smart pool, water quality, microcontroller, IoT, PCB, ESP32, ESP-Now, MQTT, Flask, Angular

## Citace

ŠŤASTNÝ, Marek. *Systém pro vzdálený monitoring a správu zahradních bazénů*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek,

# System pro vzdálený monitoring a správu zahradních bazénů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Marek Šťastný  
11. května 2022

## Poděkování

Děkuji vedoucímu práce Ing. Václavu Šimkovi za jeho cenné rady, vstřícnost při konzultacích a za pomoc s pájením desek plošných spojů. Dále bych chtěl poděkovat svým kamarádům Ing. Viktoru Kovaříkovi za praktické rady zejména v oblasti vývoje firmwaru, Vlastimilu Drškovi za přeložení abstraktu do anglického jazyka a Michaelu Kincovi za diskuze nad touto prací a motivaci k usilovné práci. Poděkování patří také mé přítelkyni Ivě Kašparové za pomoc s kontrolou gramatiky technické zprávy a podporu při studiu.

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Měření spojená s bazény a kvalitou vody</b>	<b>4</b>
2.1 Hodnota pH a její význam . . . . .	4
2.2 Význam hodnoty ORP . . . . .	5
2.3 Význam měření teploty . . . . .	6
2.4 Vodní hladina . . . . .	6
<b>3 Přehled dostupných technologií</b>	<b>8</b>
3.1 Sériové komunikační protokoly . . . . .	8
3.2 Přehled komunikačních protokolů vhodných pro IoT . . . . .	11
3.3 Přehled vhodných komunikačních modulů pro napájení z baterie . . . . .	15
<b>4 Návrh architektury</b>	<b>18</b>
4.1 Stávající řešení . . . . .	18
4.2 Definice požadavků . . . . .	18
4.3 Architektura . . . . .	19
4.4 Řídící server . . . . .	21
4.5 Návrh grafického uživatelského rozhraní . . . . .	23
<b>5 Technická realizace řídicího systému</b>	<b>24</b>
<b>6 Implementace</b>	<b>32</b>
6.1 Sestavení bazénových modulů . . . . .	32
6.2 Vytváření firmwaru pro bazénové moduly . . . . .	33
6.3 Implementace REST API serveru . . . . .	35
6.4 Implementace grafického uživatelského rozhraní . . . . .	35
<b>7 Testování</b>	<b>39</b>
7.1 Ověření výdrže baterie plovacího modulu . . . . .	39
7.2 Testování celého systému . . . . .	41
7.3 Další možný vývoj . . . . .	43
<b>8 Závěr</b>	<b>44</b>
<b>Literatura</b>	<b>45</b>
<b>A Schémata zapojení navržených obvodů</b>	<b>47</b>

<b>B</b>	<b>Návrhy desek plošných spojů</b>	<b>52</b>
<b>C</b>	<b>Obsah přiložené SD karty</b>	<b>56</b>

# Kapitola 1

## Úvod

Lidé si mnohdy pořizují zahradní bazén s představou příjemných chvil strávených ve vodě během teplého léta, avšak s bazény je také spojená práce s udržováním kvality a zdravotní nezávadnosti vody. V případě, že se o ni lidé nedokážou adekvátně postarat, může se její stav zhoršit natolik, že je potřeba buďto její kompletní obměna, nebo obnovení chemické rovnováhy za pomoci chemických přípravků. Automatizace se s postupujícím technickým vývojem dostává z původní domény průmyslu stále ve vyšší míře také do lidských domácností. Zároveň se rozvíjí trend, kdy jsou různé zařízení spolu propojena a sdílí mezi sebou informace. Na základě sdílených informací zařízení mohou vykonávat kvalifikovanější rozhodnutí.

Tato práce se pokouší oba zmíněné trendy využít a vytvořit systém zařízení, který bude mít potenciál nechtěné chemické jevy ve vodě odhalit, případně jim zabránit a celkově zvýšit komfort použití bazénu. Navíc by měl na základě intenzity slunečního svitu a teploty vody umožňovat skrze podmíněné spouštění čerpadla maximalizovat efektivitu solárního ohřevu. Uživatel bude mít přístup k naměřeným hodnotám skrze webovou aplikaci.

Práce je rozdělena na několik částí. V kapitole 2 jsou popsány měřené parametry vody, které mají vliv na její kvalitu. Zároveň jsou zde popsány i způsoby jejich měření. Obsahem kapitoly 3 je přehled dostupných sériových protokolů, bezdrátových technologií a modulů vhodných pro bezdrátovou komunikaci při napájení z baterie. Kapitola 4 popisuje navrženou architekturu která vznikla na základě informací z předchozích kapitol a na základě rešerše existujících řešení. V kapitole 5 je popsán návrh jednotlivých modulů bazénového systému a jejich následná transformace do návrhu desek plošných spojů. Obsahem kapitoly 6 je popis implementace navrženého řešení. Kapitola 7 uvádí, jakým způsobem byly validovány senzory a popisuje testování jednotlivých částí systému i celku. Na konci je navrženo, jakým způsobem je možné vytvořený systém dále rozvíjet a zdokonalovat.

## Kapitola 2

# Měření spojená s bazény a kvalitou vody

Pro to, aby byla voda pro koupání zdravotně nezávadná, musí splňovat určité parametry. Například pro veřejná koupaliště jsou definovány zákonem 238/2011 Sb. Mezi tyto parametry patří například hodnota pH, množství volného a vázaného chlóru, průhlednost, přítomnost různých typů bakterií a další.[21] Ačkoli se u zahradních bazénů většina těchto měření neprovádí (kvůli jejich složitosti a potřebné odbornosti), některé z nich je možné uskutečnit i za pomoci běžně dostupných senzorů, které mají potenciál ve spojení s výpočetní technikou činnost měření automatizovat.

### 2.1 Hodnota pH a její význam

Měření pH je spolehlivá metoda pro určování acidity chemických látek. Hodnota pH je definována jako záporný dekadický logaritmus počtu molů oxoniových kationtů  $H_3O^+$  v jednom litru chemického roztoku.[13]

Definice pH:

$$pH = -\log_{10}[H_3O^+] \quad (2.1)$$

Platí, že čím je pH nižší, tím má látka větší aciditu. Látky podle pH dělíme do těchto základních skupin:

Tabulka 2.1: Dělení látek podle kyselosti

označení	rozmezí	příklad
silně kyselé	$pH \in (0, 1)$	kyselina chlorovodíková
slabě kyselé	$pH \in (1, 7)$	ocet
neutrální	$pH = 7$	destilovaná voda
slabě zásadité	$pH \in (7, 13)$	jedlá soda
silně zásadité	$pH \in (13, 14)$	hydroxid sodný

pH vody na koupání by se dle vyhlášky mělo pohybovat v rozmezí 6,5–7,2 [21], což dle tabulky 2.1 odpovídá téměř neutrálnímu roztoku. Hodnotu pH je důležité sledovat hned z několika důvodů. Při změně pH se mění chemické vlastnosti látek. Pokud má voda nižší hodnotu než 7,2, bývá považována za korozivní, což může způsobit škody na



technickém vybavení. Dále při nesprávné kyselosti vody může docházet k mléčnému zákalu vody, podráždění kůže, zrychlenému stárnutí materiálů, nebo snižování účinnosti chlóru. [18]

### 2.1.1 Základní principy měření

#### Acidobazické indikátory

Jedná se o látky, které při styku s vodným roztokem, který překročil určitou hodnotu pH, mění svoji barvu.[6] Takovými látkami jsou obvykle napuštěny indikační papírky. Po namočení indikátoru do vody je uživatel schopen hodnotu pH z papírku odečíst po porovnání s barevnou indikační stupnicí. Zmiňovaná metoda je velice levná, avšak má malou přesnost a její potenciál pro automatizaci je minimální.



Obrázek 2.1: Indikátorové papírky od firmy Lach-Ner<sup>1</sup>

#### Potenciometrické měření

Toto měření spočívá v měření napětí mezi dvěma elektrodami. U běžně používaných skleněných sond se využívá jevu, kdy se napětí na stranách tenké skleněné membrány mění v závislosti na rozdílném poměru koncentrací  $H_3O^+$ . [5] U komerčních řešení bývají pro jednoduchost užívání měřicí elektrody kompaktně uloženy do jedné sondy, viz obr. 2.2. Pro zajištění vyšší přesnosti měření je vhodné sondu před použitím kalibrovat.

## 2.2 Význam hodnoty ORP

ORP (*oxidačně-redukční potenciál*) je hodnota udávající tendenci látek v roztoku odevzdávat elektrony (oxidace), či přijímat elektrony (redukce). Měření ORP je analogické k pH, jen se v roztoku namísto aktivity protonů měří aktivita elektronů, přičemž se naměřená hodnota nepřevádí na škálu od nuly do čtrnácti, ale hodnota se ponechává v naměřených mV. [12]

<sup>1</sup>Převzato z <https://www.verkon.cz/papirky-indikatorove-detail/>



Obrázek 2.2: Ukázka skleněné pH sondy

Bakterie, řasy a škodlivé látky vyskytující se ve vodě jsou pro nás škodlivé kvůli chemické reaktivitě, která je možná díky volným elektronům. Po přidání chlóru do vody nastává chemická reakce, při které většina nebezpečných látek interaguje s chlórem a vytváří neškodné vedlejší produkty. Díky hodnotě ORP jsme tedy schopni nepřímo určit bezinfekčnost vody. Dále jsme schopni určit, zdali je ve vodě dostatečné množství chlóru. Zde je však potřeba dbát na to, že účinnost chlóru je závislá i na hodnotě pH, takže je vhodné tyto měření kombinovat. Ideálně by se ORP měla u vody na koupání pohybovat v rozmezí 700mV–750mV.[16]

Senzory na měření ORP jsou velice podobné sensorům na měření pH. Hlavní rozdíl je, že ORP sonda má elektrodu z ušlechtilých kovů a že nebývá kalibrována v závislosti na teplotě, jelikož takováto kalibrace je v praxi téměř nemožná.[12]

### 2.3 Význam měření teploty

Komerční provozovatelé koupališť nebo veřejných bazénů mají povinnost tento parametr kontrolovat třikrát denně. U soukromých bazénů je tento údaj spíše orientační. Teplota vody má velký vliv na komfort užívání bazénu. Bazény s regulací teploty mají potenciál prodloužit pokud má bazén možnost regulovat teplotu. Užitečnost této hodnoty spočívá i ve faktu, že je s její pomocí možné zpřesnit měření pH a některých dalších parametrů.

Teplota se dá měřit několika způsoby. Nejjednodušší je měření na základě teplotní roztažnosti kapalin, kdy při zvyšování teploty kapalina zvětšuje svůj objem a vtlačuje se do tenké trubičky. Podle zaplnění trubičky d rýskou a pohledem jde snadno vyčíst aktuální teplota. Pro elektrické měření se mnohem více hodí senzory teploty, které jsou založeny na principu změny odporu termistoru s měnící se teplotou.

### 2.4 Vodní hladina

Výšku vodní hladiny je u bazénů vhodné kontrolovat ze třech základních důvodů. Za prvé se jedná o prostý fakt, že uživatelé chtějí mít bazén plný. Za druhé je možné pomocí výšky vodní hladiny kontrolovat, zdali nedochází k nadměrnému úbytku vody. Konečně, v pří-

padě, že je u bazénu implementována funkce automatického spouštění čerpadla, je potřeba kontrola, že vodní hladina dosahuje alespoň do výšky skimmru. V opačném případě by čerpadlo nenasávalo vodu, docházelo by k jeho přehřátí a možnému destrukci.

#### 2.4.1 Plovákové snímače

Jedná se o jednoduchou a levnou variantu, jak kontrolovat výšku vodní hladiny. Plovákové snímače jsou sestaveny ze dvou hlavních částí – těla a plováku. Tělo je připevněno ke stěně nádoby v takovém místě, aby jeho na volno připevněný plovák měl přístup k hladině. Při změně hladiny dojde vlivem hydrostatického tlaku ke změně horizontální polohy plováku a následně k sepnutí, nebo rozepnutí obvodu. Nevýhodou tohoto měření je fakt, že plovákový snímač zabírá vodní plochu a kazí estetický dojem. Dále je měření náchylné na vlny a mechanické poškození.

#### 2.4.2 Hydrostatické snímače

Výšku vodní hladiny lze dopočítat i díky senzoru tlaku. Senzor lze umístit buďto přímo do vody, nebo tlak přenést pomocí hadičky mimo bazén a měření provést tam. Naměřený tlak lze pomocí vzorce pro hydrostatický tlak 2.2 a pomocí dalších dobře známých hodnot převést na výšku vodního sloupce nad senzorem.

Vzorec pro výpočet vodního sloupce nad senzorem 2.3 dostaneme jednoduchou úpravou vzorce 2.2. Tato hodnota však sama o sobě nestačí, jelikož senzor nenaměřil pouze hydrostatický tlak, ale i tlak atmosferický. Jedná se o proměnlivou veličinu, která je schopna přesnost měření ovlivnit v rámci desítek centimetrů. Proto je potřeba hodnotu atmosferického tlaku dynamicky zjišťovat a následně odečíst, jak je ukázáno v rovnici 2.4.

Rovnice pro výpočet vodního sloupce nad senzorem tlaku:

$$p = \rho \cdot g \cdot h \quad (2.2)$$

$$h = \frac{p}{\rho \cdot g} \quad (2.3)$$

$$h_r = \frac{p}{\rho \cdot g} - p_a \quad (2.4)$$

Kde:

$p$ : tlak [ $Pa$ ]

$\rho$ : hustota kapaliny [ $kg/m^3$ ]

$g$ : tíhové zrychlení [ $m/s^2$ ]

$h$ : výška vodního sloupce nad senzorem ve vakuu [ $m$ ]

$h_r$ : výška vodního sloupce nad senzorem [ $m$ ]

$p_a$ : atmosferický tlak [ $Pa$ ]

## Kapitola 3

# Přehled dostupných technologií

### 3.1 Sériové komunikační protokoly

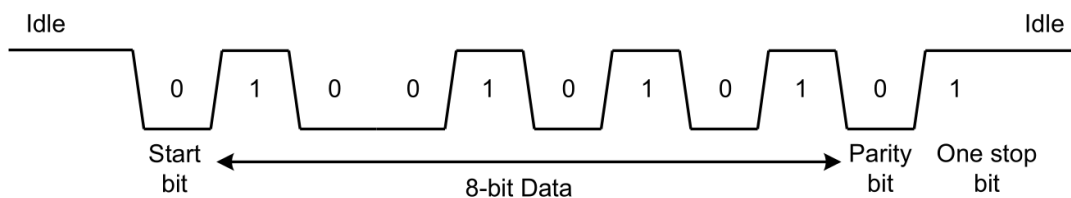
V následující sekci jsou rozebrány významné sběrnice komunikační protokoly, prostřednictvím kterých může být zprostředkována komunikace mezi mikrokontroléry a periferními zařízeními. Tato sekce čerpá z [17].

#### 3.1.1 UART

UART (*Universal Asynchronous Receiver/Transmitter*) je sériový<sup>1</sup> komunikační protokol pro komunikaci mezi dvěma zařízeními (point to point), který poskytuje plně duplexní spojení<sup>2</sup>.

Propojení zařízení se utváří pomocí minimálně třech vodičů. Jeden vodič je zemnicí a další dva datové, kdy každý z datových vodičů posílá data v opačném směru. Datové vývody na zařízení se označují U\_TX a U\_RX. Data proudí vždy ve směru od vývodu U\_TX na jednom zařízení k U\_RX na zařízení druhém. Proto je třeba dbát na „překřížení“ vodičů tak, aby měl datový vodič vždy na jednom konci U\_TX a U\_RX na druhém. Tento protokol lze rozšířit o další dva vodiče pro kontrolu toku dat. Tyto vodiče jsou na obou stranách připojeny k vývodům RTS (request to send) a CTS (clear to send).

Data jsou posílána po rámcích. Rámec se skládá z jednoho start bitu, pěti až devíti datových bitů, jednoho paritního a jednoho až dvou stop bitů. Paritní bit může být buďto sudý, nebo lichý a jeho účel je kontrola jednobitových chyb.



Obrázek 3.1: Ukázka UART rámce s jedním start bitem, osmi datovými bity, jedním (sudým) paritním bitem, a jedním stop bitem. Převzato z [17]

<sup>1</sup>U sériové komunikace se používá pouze jeden datový vodič pro daný směr komunikace. U paralelní komunikace se využívá více datových vodičů pro daný směr přenosu dat. Sériová komunikace mívá na rozdíl od paralelní zpravidla menší datový tok, zato bývají data zasílána na delší vzdálenost.

<sup>2</sup>Zařízení mohou zároveň zasílat i přijímat data

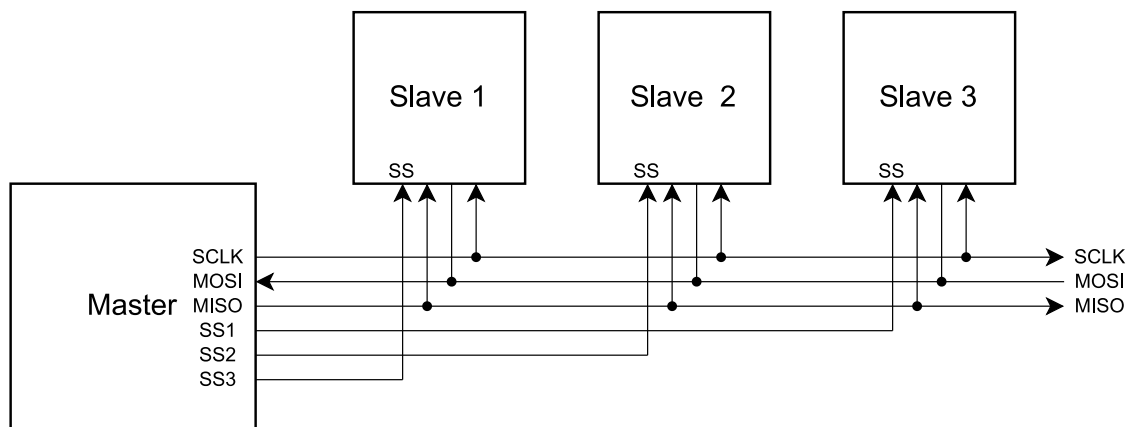
U tohoto protokolu se nepoužívá hodinový signál pro synchronizaci zařízení, proto je označován jako asynchronní. Je potřeba, aby byla rychlost přenosu (baud rate<sup>3</sup>) oběma zařízení předem známá a její hodnota se rovnala. Přijímající zařízení se synchronizuje s vysílajícím pomocí interních hodin, které mají šestnáctkrát vyšší frekvenci než je baud rate. Při začátku start bitu začne přijímací zařízení na čítači odpočítávat od nuly do patnácti a vždy při osmém tiku odečítá hodnotu na datovém vodiči.

### 3.1.2 SPI

SPI (*Serial peripheral interface*) je stejně jako UART 3.1.1 plně duplexní protokol. Na rozdíl od UART využívá master-slave<sup>4</sup> koncept. Jedno centrální zařízení (master) je připojeno pomocí sběrnice k dalším zařízením (slaves) a v případě potřeby iniciuje komunikaci právě s jedním z nich. Typické využití tohoto protokolu je přenos dat mezi mikrokontrolérem a periferními zařízeními. Společná sběrnice se skládá ze třech vodičů:

- SCLK (serial clock) společný hodinový signál,
- MOSI (master out, slave in) vodič pro přenos dat od mastera k slave zařízení,
- MISO (master in, slave out) vodič pro přenos dat od slave zařízení do mastera.

Jak je vidět na obr. 3.2, master má pro každé slave zařízení vlastní vodič označovaný SS (slave select). Pomocí něj vybírá zařízení pro komunikaci. Z výše zmíněného plyne, že pro komunikaci s  $n$  zařízeními je potřeba  $3 + n$  vodičů. Pro UART je potřeba  $2n$  vodičů. Pro větší počet zařízení dostáváme u SPI značnou úsporu výstupů pro master zařízení.



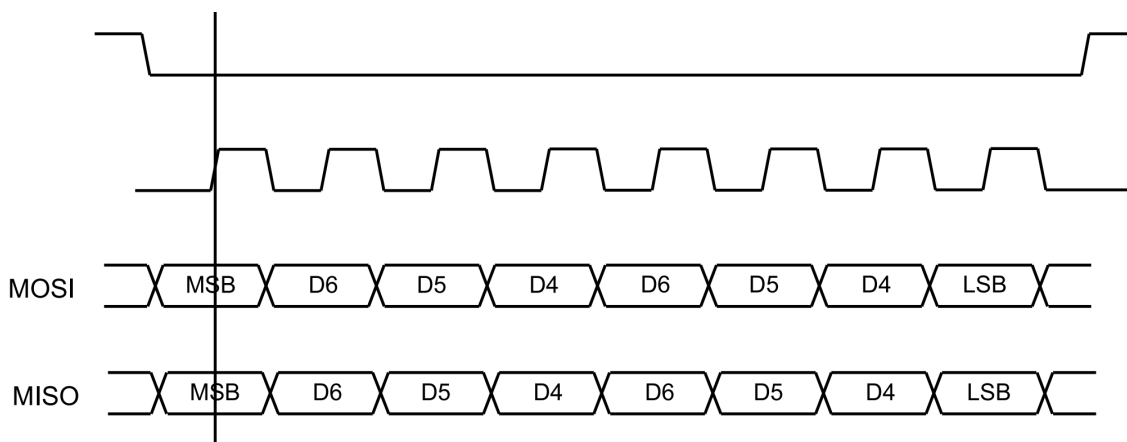
Obrázek 3.2: SPI sběrnice propojující tři slave zařízení s masterem. Šipky značí směr toku dat.

Přenos dat inicializuje master a to tak, že stáhne napětí na nulu u vodiče SS pro zařízení, se kterým chce komunikovat. Poté se začínají odečítat hodnoty na datových vodičích - vždy při náběžných, nebo sestupných hranách hodinového signálu. To, zda se odečítá na náběžnou či sestupnou hranu, závisí na tom, v jakém módu je nastavený přenos dat. Módy se liší podle toho, zda je hodinový signál implicitně v logické nule, nebo logické jedničce, a také

<sup>3</sup>Modulační rychlost - udává počet změn stavu za jednotku času.

<sup>4</sup>Asymetrický model komunikace, kdy jeden prvek (master) převezme řídicí roli a diktuje ostatním prvkům (slave), jak má probíhat komunikace.

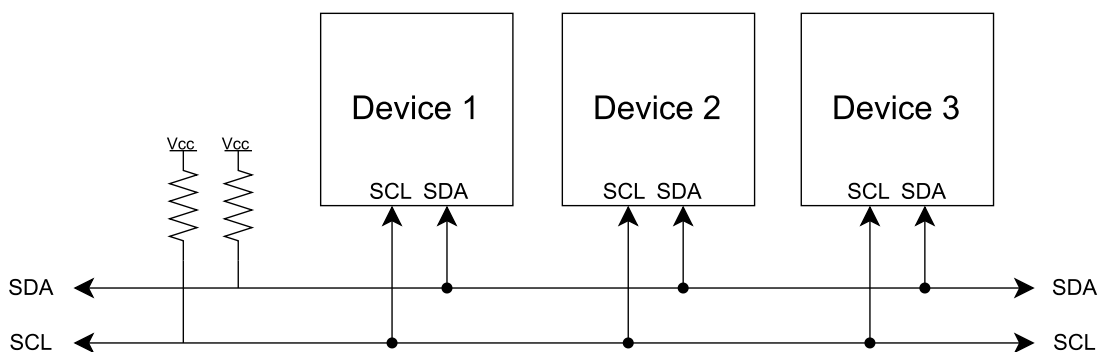
podle toho, zdali se data odečítají na náběžnou hranu, či sestupnou hranu. Po přenesení posledního bitu, hodinový signál zůstává v implicitní hodnotě a nakonec se signál SS vrací do logické jedničky.



Obrázek 3.3: Schéma přenosu dat pomocí SPI. Hodinový signál je implicitně v logické nule a měření se provádí při náběžné hraně. Převzato z [17]

### 3.1.3 I<sup>2</sup>C

I<sup>2</sup>C (*Inter-Integrated Circuit*) je synchronní half-duplexní protokol, který využívá pro propojení několika zařízení pouze dva vodiče. Hodinový signál SCL (serial clock) a SDA (serial data). Oba tyto vodiče jsou implicitně v logické jedničce, proto je k nim potřeba připojit pull-up rezistory, jak je zobrazeno na obr. 3.4. Jak název napovídá, protokol je vhodný především k propojení zařízení v rámci integrovaných obvodů. Délka sběrnice je (kvůli maximální možné kapacitanci (400 pF) a kvůli nízké odolnosti vůči šumu) omezena na maximálně jednotky metrů. Stejně jako SPI 3.1.2 využívá master-slave koncept, avšak v tomto případě může být master kterékoli zařízení připojené ke sběrnici.



Obrázek 3.4: I<sup>2</sup>C sběrnice propojující tři zařízení. Vlevo jsou k vodičům připojeny dva pull-up rezistory zajišťující implicitní napětí na vodičích.

Master zařízení může začít komunikaci kdykoli na sběrnici neprobíhá komunikace mezi jinými zařízeními. Přenos má několik fází. V první fázi probíhá vyhrazení sběrnice pro

komunikaci pomocí tzv. start condition – SCL signál je v implicitní hodnotě logické jedničky a přitom dojde ke stažení SDA hodnoty na nulu. V druhé fázi master adresuje slave zařízení. Každé slave zařízení musí mít v rámci linky unikátní sedmi bitovou adresu. Pomocí ní je master zařízením adresován slave, se kterým probíhá komunikace.

Následně master zasílá R/S (receive/send) bit. Tím určí, zdali má slave data přijímat, nebo vysílat.

Ve čtvrté fázi slave, jehož adresa se shoduje s adresou zapsanou na sběrnici, ví, že je s ním komunikováno, a zapíše na SDA sběrnici ACK bit.

Dále v případě, že slave má data přijímat, zapíše příchozích 8 bitů do registru a poté pošle ACK bit. V případě, že má data odesílat, pošle 8 bitů na sběrnici a poté očekává ACK bit od master zařízení. Tento proces se opakuje, dokud nejsou poslány veškerá data. Při zasílání dat od slave zařízení je poslední byte potvrzen NACK bitem.

V poslední fázi master pro ukončení komunikace zasílá tzv. STOP condition – ve chvíli, kdy se SCL signál vrací do implicitní hodnoty, se na SDA signálu vytvoří náběžná hrana a SDA signál zůstává v implicitní hodnotě. Od této doby je sběrnice opět volná pro komunikaci s dalšími zařízeními.

Tento způsob komunikace je z výše zmiňovaných protokolů nejefektivnější co se týče potřeby fyzických propojení. Počet nutných vodičů pro propojení libovolného množství zařízení je konstantní (vždy 2).

## 3.2 Přehled komunikačních protokolů vhodných pro IoT

bezdratove zdratove pridatlorawan nebo neco s delsim dosahem

### 3.2.1 Wi-Fi

Jedná se patrně o nejúspěšnější sadu standardů pro vytváření bezdrátových LAN sítí spadající pod organizaci IEEE<sup>5</sup>. Tyto standardy, jejichž název vždy začíná označením 802.11, sdílí některé společné vlastnosti:

- využití přenosového pásma 2,4–2,485 GHz nebo 5,1–5,8 GHz,
- zařízení se starším standardem mohou komunikovat s novějšími (zpětná kompatibilita),
- stejný formát rámců na linkové vrstvě,
- schopnost snížit přenosovou rychlost v zájmu vyššího dosahu,

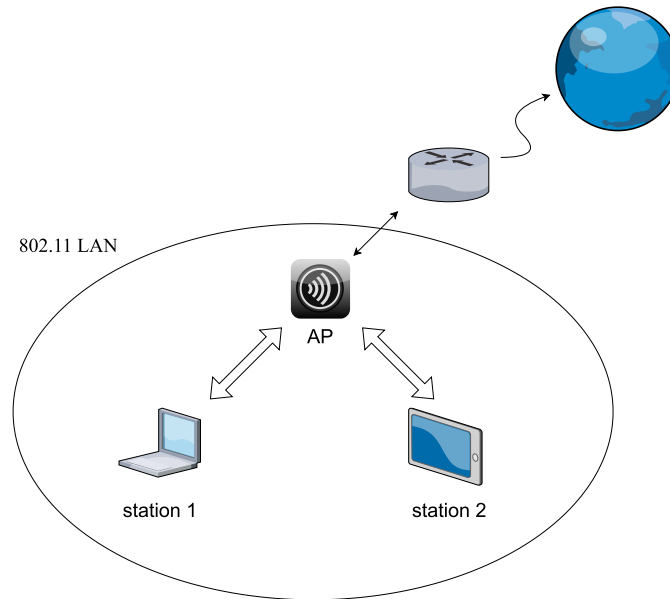
Dosah zařízení se liší v závislosti na konkrétní variantě standardu. Obecně lze říci, že za ideálních podmínek dosah při přenosu dat ve vnitřních prostorách nepřesahuje 70 metrů a venku nepřesahuje 250 metrů. Zařízení využívající 5 GHz pásmo jsou schopna komunikovat na kratší vzdálenost, zato mají vyšší přenosovou rychlost.

Základní jednotka architektury Wi-Fi sítě je BSS<sup>6</sup>. Ta se skládá z bezdrátových stanic a jednoho AP<sup>7</sup>. Umožňuje bezdrátovým zařízením spolu komunikovat skrze společné medium – AP. Jak je ukázáno na obr. 3.5, v domácnostech bývá jedna (nebo více) BSS skrze AP dále napojena na router s konektivitou k internetu.[11]

<sup>5</sup>Institute of Electrical and Electronics Engineers (Institut elektrotechnických a elektronických inženýrů)

<sup>6</sup>basic service set (základní servisní sada)

<sup>7</sup>access point (přístupový bod)



Obrázek 3.5: Ilustrace jednoduché WiFi sítě. Dvě stanice komunikují s AP. Pro přístup k internetu AP dále zasílá data routeru.

### 3.2.2 Bluetooth

Původním záměrem technologie Bluetooth (standard IEEE 802.15) bylo vytváření ad hoc PAN sítí pro přenos dat na krátkou vzdálenost. Jedním z cílů byla možnost eliminace množství kabelů potřebných pro užívání příslušenství k počítači (klávesnice, sluchátka). Důležitými aspekty byla podpora pro přenos hlasu a využití levného a energeticky úsporného vysílače.

Bluetooth využívá pásmo 2.4 GHz stejně jako Wi-Fi nebo ZigBee. U tohoto standardu byla věnována pozornost odolnosti vůči interferenci. Toho je dosaženo především díky posílání malých paketů pomocí FHSS<sup>8</sup> s vysokou frekvencí přeskoků. Bluetooth také nabízí detekci chyb a automatické zasílání požadavku na opakované přeposlání.

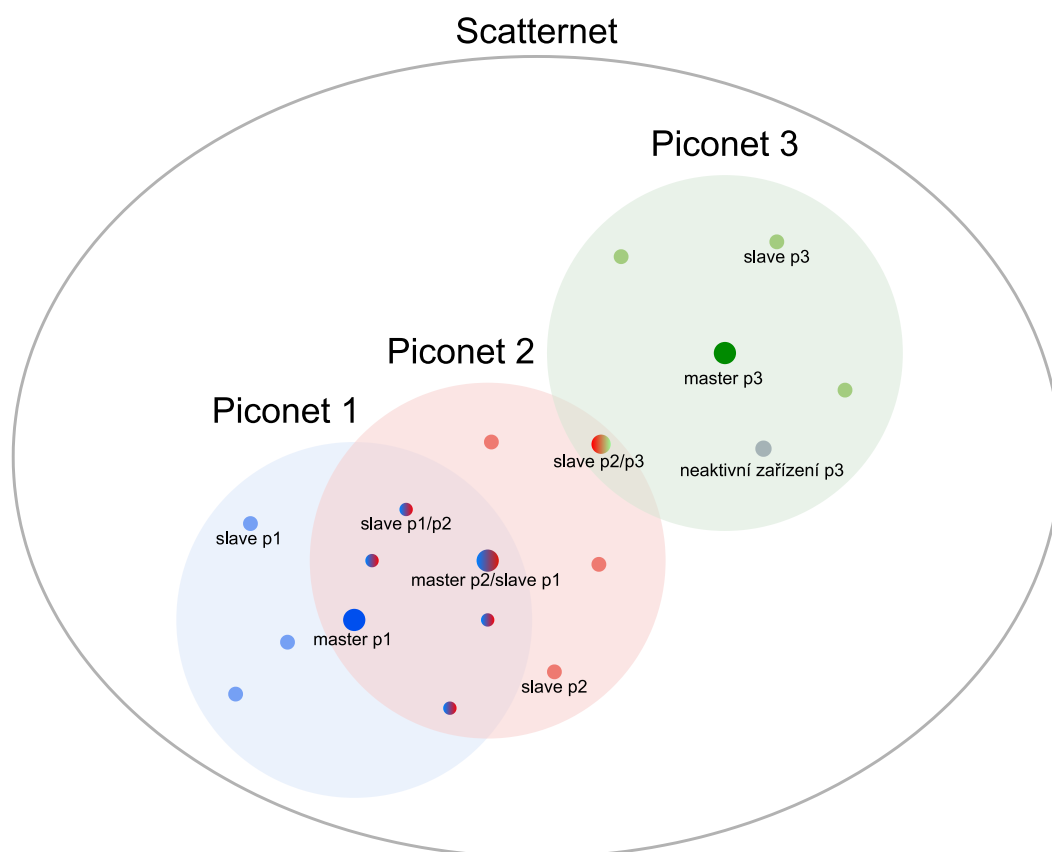
Základní jednotkou sítě je „piconet“ – 2 až 8 aktivních zařízení, z nichž jedno přijme roli master a ostatní slave. V piconetu může být až 255 neaktivních zařízení. Rozvržení rolí je náhodné a po ukončení komunikace zařízení role ztrácí. Úkolem mastera je pro svůj piconet řídit komunikaci. Pro vytváření větších sítí se spojuje více překrývajících se piconetů a utváří „scatternet“. V rámci scatternetu komunikuje každý piconet na jiném FHSS kanálu a zařízení mohou mít více rolí. Například slave v rámci jednoho piconetu a master u jiného, viz obr. 3.6.[15]

V dnešní době původní verzi nazýváme Bluetooth Classic, jelikož s příchodem verze Bluetooth 4 už bylo potřeba rozlišovat tři varianty:

- Bluetooth Classic,
- Bluetooth Low Energy,

<sup>8</sup>Frequency Hopping Spread Spectrum – Metoda, u které je přenosové pásmo rozděleno na několik menších pásem. Odesílatel odesílá po stanovenou dobu část dat na dané frekvenci a poté přeskočí na jinou. Doba přenosu v daném pásmu je konstantní a k přeskoku dochází zpravidla několikrát za sekundu. Posloupnost frekvencí musí vypadat náhodně, avšak odesílatel i příjemce na ní musejí být předem domluveni. Tato posloupnost vlastně vytváří kanál, na kterém zařízení komunikují.





Obrázek 3.6: Ukázka přiřazení rolí Bluetooth zařízení v rámci piconetů. Piconety vytváří jeden větší scatternet.

- Bluetooth High Speed.

Verze Bluetooth 5.0 nabízí velice zajímavé parametry pro IoT aplikace. Oproti předchozí verzi zprostředkovává dvakrát rychlejší párování, až čtyřnásobný dosah (200 m) a až osmkrát rychlejší přenos dat (50 mbps). Základní parametry jsou zobrazeny v tabulce 3.1.

Tabulka 3.1: Přehled hlavních Bluetooth verzí

Verze	dosah [m]	Maximální přenosová rychlost [Mbit/s]
Bluetooth 1.2	10	1
Bluetooth 2.0	30	3
Bluetooth 3.0	30	25
Bluetooth 4.0	60	25
Bluetooth 5.0	240	50

Verze Low Energy je navržena pro energeticky úsporný přenos krátkých zpráv a umožňuje vytvářet různé komunikační topologie. Například pro fitness zařízení lze navazovat spojení

point-to-point. Pro hledání věcí umožňuje broadcast a pro automatizaci budov vytváří mesh síť.<sup>9</sup>

### 3.2.3 ESP-NOW

ESP-NOW je protokol, který vyvinula firma Espressif jako součást své vývojové platformy. Zprostředkovává nespojovanou peer to peer (P2P) i broadcastovou komunikaci, která může být šifrována pomocí CCMP. Tento vícevrstevní protokol, je založen na standardu IEEE 802.11 (viz 3.2.1), avšak využívá výlučně 2,4 GHz pásmo.

Další z rozdílů oproti Wi-Fi je, že zařízení pro komunikaci nepotřebují AP, ale mohou komunikovat v režimu peer to peer. V long range módu je možné data posílat až na vzdálenost 480 m.[1]

ESP-NOW vychází sice z Wi-Fi, avšak pro komunikaci zasílá pouze „vendor-specific action“ rámce. Data vkládá do pole Body. Z toho plyne omezení pro délku zasílaných zpráv – toto pole má maximální délku 250 bytů.

Zařízení si pro unicastovou komunikaci uchovává informace o peerech v interní paměti. Jeden z limitů této technologie je, že je možné uchovávat informace o maximálně 20 peerech. Tento limit je omezen dokonce pouze na 10, pokud jsou zprávy šifrovány.

Tento protokol neprovádí handshaking. To jej v kombinaci s malým objemem přenášených dat činí oproti Wi-Fi energeticky úsporným. Navíc je možné v long range módu data posílat až na vzdálenost 480 m. V kombinaci se šifrováním komunikace tyto vlastnosti činí danou technologii vhodnou pro vytváření sítí u baterií napájených zařízeních.[3]

### 3.2.4 MQTT

MQTT je klient-server publish/subscribe protokol, který klade důraz na jednoduchou implementaci. Využívá TCP/IP a díky nízkému overheadu paketů je vhodný k použití u případů, kde je potřeba šetřit s energií pro přenos dat nebo s datovým tokem. Další výhodou je, že u jednotlivých zpráv je možné nastavit jednu ze tří úrovní kvality služby.

Základní myšlenka návrhového vzoru publish/subscribe spočívá v tom, že existuje centrální bod (broker), přes který proudí veškerá komunikace. Dochází tedy k oddělení odesílatele a příjemce. Broker přijímá zprávy od odesílatele (publisher), tyto zprávy zpracovává a následně je pošle zařízením, které se přihlásilo k jejich odběru (subscriber). Díky tomuto konceptu se publisher ani subscriber nemusí mezi sebou synchronizovat pro přenos dat ve stejnou dobu, a dokonce nepotřebují informaci o existenci toho druhého.[8]

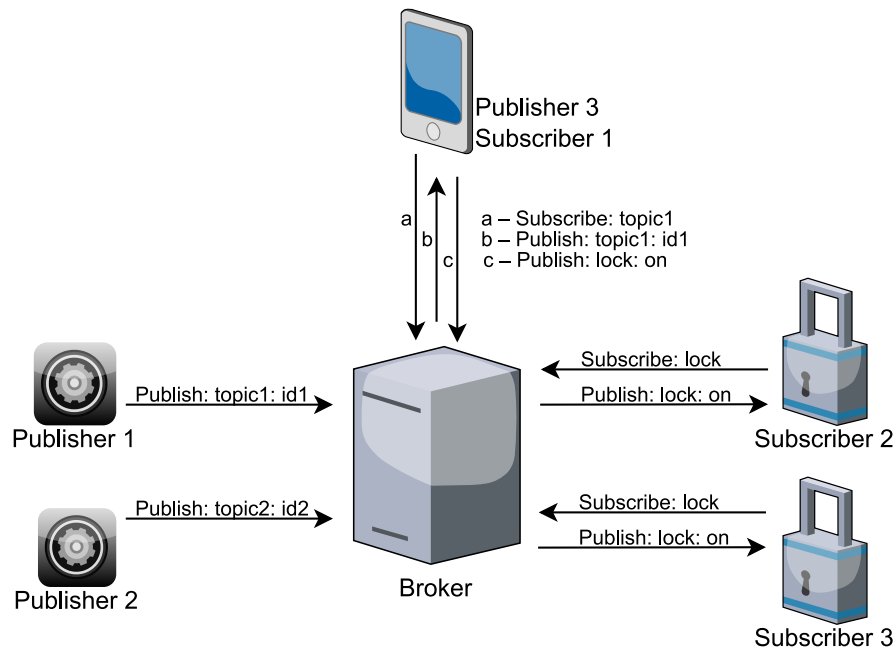
Jeden z úkolů brokera je filtrování zpráv. Tím je zajištěno, že subscriber dostává pouze vyžádané zprávy. Filtrování je prováděno na základě témat (topic). Jak je ukázáno na obr. 3.7, zařízení, které chce dostávat zprávy na určité témata, pošle brokerovi subscribe zprávu s tématy a ten je následně zodpovědný, za jejich doručení.

Standard MQTT 3.1 byl v roce 2010 vydán firmou IBM. Jednalo se o první veřejný MQTT standard, který byl navíc zdarma. Mohl jej tedy implementovat a využívat kdokoli. Následně byl v roce 2014 potvrzen standard MQTT 3.1.1 tentokrát už organizací OASIS<sup>10</sup>. Tato verze měla oproti předchozí spíše zanedbatelné změny. Verze MQTT 5 z roku 2019 naproti tomu reaguje na překotný vývoj na poli IoT. Zohledňuje vlastnosti pro použití na cloudových platformách, stanovuje koncept kritických zpráv a další.[9]

prehled rest serveru   prehled forntend frameworku

<sup>9</sup><https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>

<sup>10</sup>url: <https://www.oasis-open.org>



Obrázek 3.7: Ukázka komunikace prvků sítě MQTT. Subscriber1 odebírá zprávy na téma topic1. Subscriber 2 a 3 odebírají zprávy na téma lock. Publisher 1 zasílá zprávu brokerovi, ten ji přeposílá Subscriberovi 1. Následkem čehož publikuje zprávu on na téma lock, kterou přebírá broker a přeposílá ji zařízením Subscriber 2 a 3.

### 3.3 Přehled vhodných komunikačních modulů pro napájení z baterie

#### 3.3.1 ESP32

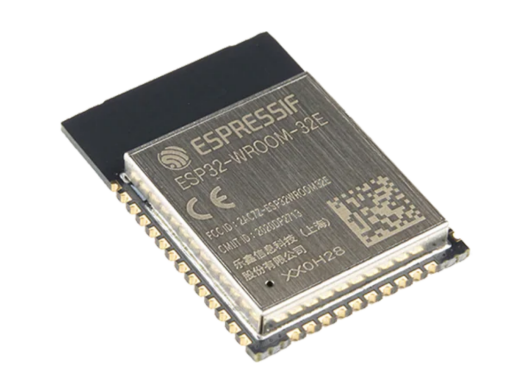
Jedná se o system on chip (SoC) řadu mikrokontrolérů, které se těší velké oblibě mezi nadšenci pro domácí automatizaci. S cenou okolo 120 korun, v kombinaci s univerzálností a poměrně vysokým výkonem se jedná o zařízení s rozličnými možnostmi využití.

Tento mikrokontroler je poháněn mikroprocesory Tensilica Xtensa LX6 v jedno- nebo dvoujádrovém provedení. Další varianty jsou dvoujádrový mikroprocesor Tensilica Xtensa LX7, nebo jednojádrový mikroprocesor založený na architektuře RISC-V. Tyto mikroprocesory mají nastavitelnou frekvenci hodin s tím, že jejich maximální frekvence se pohybuje v rozmezí od 80 MHz do 240 MHz.<sup>11</sup> Čip obsahuje jednotku pro bezdrátovou komunikaci Bluetooth a také pro Wi-Fi (blíže popsané v sekci 3.2.2 a 3.2.1). Tento mikrokontrolér je prodáván i jako modul s externí PCB anténou, flash pamětí a oscilátorem zakrytý stíněným obalem (viz 3.8).

Tento SoC se interně skládá z několika funkčních bloků, které jsou ukázány na obr. 3.8. Za běžného používání (active mode) se spotřeba dle výrobce na dvoujádrovém mcu o frekvenci 240 MHz může pohybovat v rozmezí 30 mA–68 mA. Ovšem pokud jsou používány bloky pro Bluetooth a Wifi, spotřeba může ve špičkách stoupnout až k hodnotě 790 mA. Naopak pro použití v bateriích napájených senzorech je možné využít light sleep mód, ve kte-

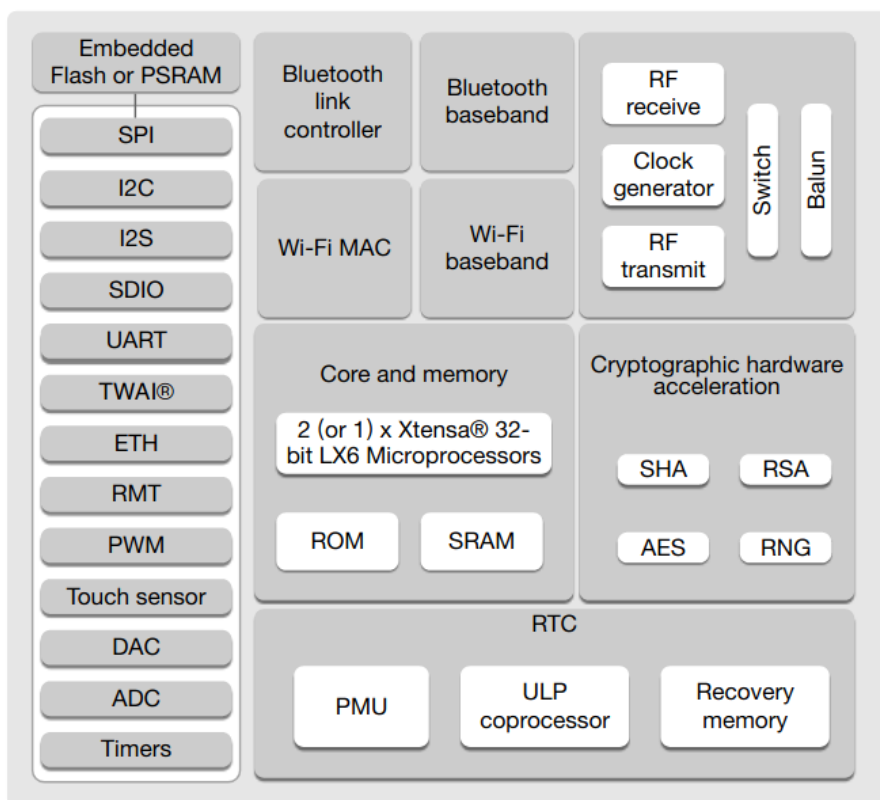
<sup>11</sup><https://www.espressif.com/en/products/socs>

<sup>12</sup>Staženo z <https://cz.mouser.com/images/marketingid/2021/img/154956428.png?v=031122.0722>



Obrázek 3.8: ESP32-WROOM-32E <sup>12</sup>

rém jsou vypnuty Wi-Fi, Bluetooth a rádio moduly. Cpu a přidružená paměť jsou odpojeny od hodinového signálu. Spotřeba v tomto režimu se pohybuje okolo 0.8 mA. V hibernation módu, kdy je zapnutý pouze Real Time Clock (RTC) časovač, je spotřeba pouze 5  $\mu$ A



Obrázek 3.9: Diagram funkčních bloků ESP32 [2]

ESP32 nabízí také širokou škálu periférií včetně 34 GPIO pinů, dvou 12 bitových ADC převodníků, jednoho DAC převodníku a Touch senzoru. U ADC je potřeba mít na paměti, že mikrokontrolér je napájen napětím 3.3 V a samotný ADC měří s určitou přesností pouze v rozsahu 100 mV–2450 mV. V neposlední řadě je možné použít sběrnice SPI, I2C, I2S.

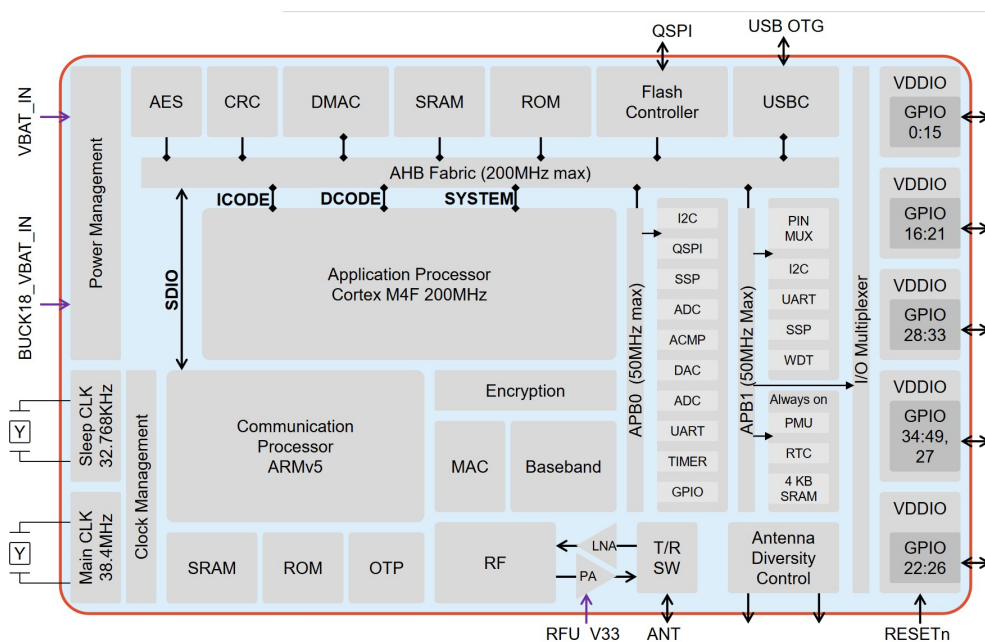
Mikrokontrolér dále obsahuje kontrolery pro podporu komunikačních protokolů jako třeba UART, SDIO a TWAI.[2]

### 3.3.2 NXP – 88MW322

88MW322 je výkonný SoC od firmy NXP s cenovkou okolo 160 Kč. Jeho CPU je postavený na návrhu 32 bitového procesoru ARM Cortex-M4F. Procesory z rodiny Cortex-M potřebují napájecí napětí 3.3 V a vynikají nízkou cenou a nízkou spotřebou energie. Tento mikrokontrolér byl navržen pro použití v doménách jako například: chytrá domácnost, automatizace budov, bridge nebo herní příslušenství. Základní specifikace tohoto mikrokontroléru nabývají hodnot:

- frekvence CPU 200 Mhz,
- CPU poskytuje 512 KB SRAM paměti,
- 128KB ROM paměti.

Pro bezdrátovou komunikaci je k dispozici Wi-Fi 4 3.2.1 modul. Pro jeho využití je nutné připojit ještě dolní propust a anténu. Pro snížení spotřeby je možné u tohoto modulu aktivovat standby mód, nebo uspat do deep sleep módu.



Obrázek 3.10: Diagram funkčních bloků 88MW322 <sup>13</sup>

Jak je vidět na schématu 3.10, 88MW322 vyniká v možnostech připojení periferních zařízení. Nabízí připojení až padesáti GPIO pinů. Dále lze připojit 3x sběrnici I<sup>2</sup>S, SPI a UART, dvakrát sběrnici I<sup>2</sup>C a dva piny pro probuzení z režimu spánku. Z analogových rozhraní nabízí ADC i DAC převodníky, dva analogové komparátory a obvod pro měření napětí baterie.[14]

<sup>13</sup>Staženo z <https://www.nxp.com/docs/en/application-note/AN12962.pdf>

## Kapitola 4

# Návrh architektury

### 4.1 Stávající řešení

Chytrých zařízení pro zahradní bazény existuje poměrně velké množství. Jako příklad lze uvést výrobek ONDILO ICO<sup>1</sup>. Tento výrobek nabízí automatizované měření teploty, pH, ORP a celkové množství rozpuštěných pevných látek. Následně provádí analýzu naměřených dat v mobilní aplikaci a předkládá uživateli doporučení. Slabinou tohoto systému je nemožnost rozšíření o další funkce. Také nenabízí žádné akční členy (aktuátory), které by prováděly obsluhu bazénu. Aktivní prvky u bazénu plní automatické spínače nebo dávkovací stanice. Dávkovače, které nabízí firma Aseko<sup>2</sup>, umí dávkovat až čtyři různé roztoky. Nastavení se provádí buďto pomocí dotykového displeje, nebo pomocí aplikace. Produkty těchto firem sdílejí společné nevýhody a to:

- nemožnost sdílet data s dalšími systémy,
- nemožnost systém ovládat z cizích automatizačních systémů,
- vysoká cena (u druhé firmy se náklady na systém pohybují v řádu stovek tisíc).

Pro automatické spínání čerpadla se používají spínače s časovačem, kdy se uživatel pro nastavení spínání musí fyzicky dostavit k časovači. Pokročilejší varianta je využití chytrého spínače implementujícího obecně rozšířený IoT protokol. Ovládání takového prvku je poté snadno začlenitelné do zdarma dostupných IoT aplikací. Z chytrých senzorů a aktuátorů lze takto automatizovat spoustu činností, avšak tato varianta vyžaduje technicky zdatného uživatele, nemalou investici času a značné úsilí.

### 4.2 Definice požadavků

Pro správné navržení systému je potřeba nejdříve vytyčit cíle, ke kterým je práce směřována. Tato práce se zaměřuje na tři hlavní cíle. První cíl je měření teploty. Jak bylo popsáno v sekci 2.3 teplota vody je jeden z nejdůležitějších parametrů, z hlediska uživatelského komfortu. Druhý cíl je automatická regulace teploty. Toho bude dosaženo pomocí automatického spouštění solárního ohřevu pouze za optimálních podmínek. Tato funkcionality má potenciál snížit spotřebu energie a zároveň zjednodušit uživateli obsluhu bazénu. Třetí cíl

---

<sup>1</sup><https://ondilo.com/>

<sup>2</sup><https://www.asekopool.com/en/dosing-and-control-system-for-automatic-pool-water-treatment-with-internet-connection/>

je kontrola kvality vody skrze měření pH. Toto měření bylo vybráno, jelikož představuje poměrně spolehlivý a dostupný souhrnný indikátor kvality vody. Dále bylo na základě výše popsané analýzy existujících řešení 4.1 stanoveno pět základních cílů, které byly zohledněny při návrhu systému:

- systém by měl být funkční a přínosný sám o sobě,
- obsluha a monitorig by měla být přístupná z internetu,
- systém umožňuje nastavení automatických funkcí,
- generovaná data by měla být přístupná jiným systémům,
- náklady na výrobu by neměly přesáhnout 15 000 Kč.

### 4.3 Architektura

Celkové řešení lze rozdělit na dva základní stavební bloky. Bazénový systém (modrá část vlevo na obr. 4.1) a informační systém (dvě části vpravo na obr. 4.1). Tyto dva bloky jsou mezi sebou propojeny zasíláním MQTT zpráv. MQTT zprávy slouží pro odesílání informací o naměřených hodnotách a také pomocí nich aktuátory dostávají příkazy k činnosti. Bazénový systém je tvořen čtyřmi prvky: Thermometer, Luxmeter, Switch a Bridge. V následujících čtyřech odstavcích budou zevrubně popsány.

#### Thermometer

Senzor plovoucí na hladině, který měří teplotu vody a pH. Kvůli bezpečnosti a komfortu bude napájen baterií. Zařízení by na jedno nabití mělo vydržet alespoň 6 měsíců, tedy celou koupací sezónu. Teplotní sonda je ponořena hlouběji do vody, aby naměřená hodnota nebyla zkreslená vrstvou teplejší vody u povrchu.

#### Luxmeter

Senzor umístěný v těsné blízkosti solárního ohřevu, který měří intenzitu slunečního světla. Naměřená hodnota se dá dále využít pro vyhodnocení optimálního spouštění solárního ohřevu. Součástka pro měření světelného toku by potenciálně mohla být umístěna i na thermometeru, avšak hrozilo by zkreslení měření, protože solární ohřev může být obecně v jiných světelných podmínkách než bazén (sklon, stínění cizími objekty). Tento senzor je také napájen baterií, a požadavky na ni jsou zde totožné jako u thermometeru.

#### Switch

Aktuátor pro spínání čerpadla. Toto zařízení má zároveň funkci senzoru, který měří výšku vodní hladiny a průtok vody čerpadlem. Jako bezpečnostní pojistka zde slouží funkce, kdy bude chvíli po sepnutí čerpadla zkontrolován průtok a v případě podlimitní hodnoty bude čerpadlo opět vypnuto. Měření výšky vodní hladiny bude měřeno na základě hydrostatického tlaku, jak je popsáno v podsekcí. Energie pro tento prvek je čerpána z přívodu energie pro čerpadlo. 2.4.2.

## Bridge

Uzel připojený pomocí WiFi k internetu. V případě, že se mu nedaří připojit k žádné z dostupných sítí, vytvoří captive portal, skrze který je třeba nakonfigurovat připojení k WiFi. Účel tohoto zařízení je komunikace s jednotlivými zařízeními v rámci bazénového systému a vytváření rozhraní pro externí aplikace. Jelikož toto zařízení musí mít neustále aktivní komunikační modul, zvyšuje se jeho energetická náročnost, a tedy je nutné jej napájet z elektrické sítě.

Komunikace těchto zařízení je zprostředkována protokolem ESP-Now, který je popsán v podkapitole 3.2.3. Síť složená z výše popsaných zařízení tvoří topologii star. Thermometer, Switch a Luxmeter komunikují pouze se zařízením Bridge, které plní roli hubu. Bridge následně tyto zprávy překládá na zprávy protokolu MQTT a pod různými MQTT tématy je vydává skrze MQTT brokera komukoli, kdo je odebírá. Přehled a formát zpráv je zobrazen v tabulce 4.1. Rozdělení naměřených hodnot na jednotlivé témata poskytuje výhodu, že externí systém se může přihlásit k odběru pouze konkrétních dat - tedy dat, která ho zajímají.

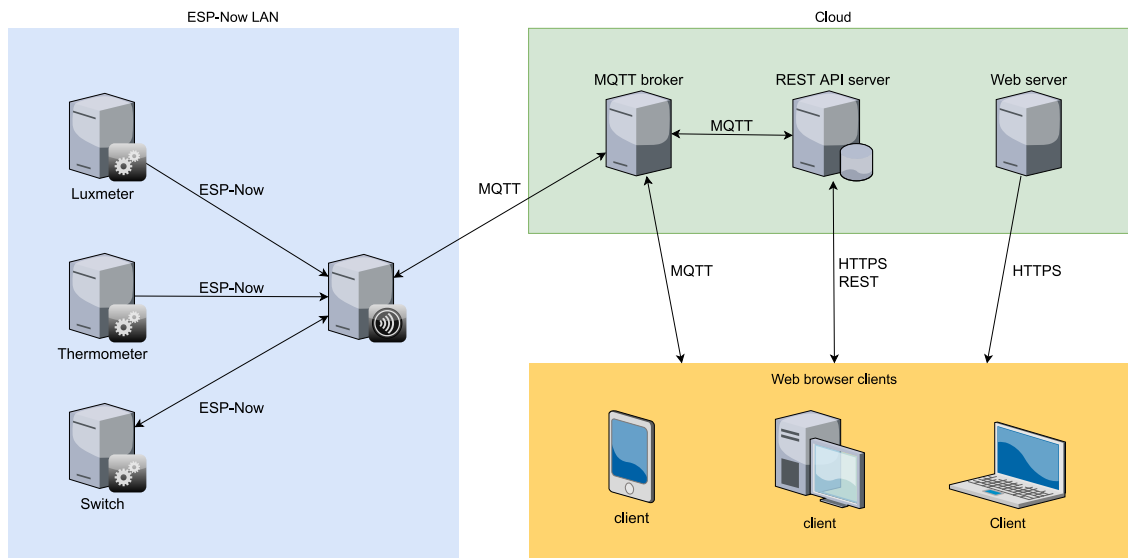
Tabulka 4.1: Popis MQTT rozhraní bazénového systému

téma	datový typ	popis
thermometer/temperature	float	naměřená teplota vody [°C]
thermometer/ph	float	naměřené pH vody
thermometer/bat	u_int8_t	zbývající procenta baterie
luxmeter/lux	u_int16_t	světelný tok [lx]
luxmeter/bat	u_int8_t	zbývající procenta baterie
switch/status	char[4]	„ON_“ - zapnuto, „OFF“ - vypnuto
switch/pressure	u_int16_t	hydrostatický tlak [Pa]
switch/flow	u_int8_t	průtok vody čerpadlem [l/hod]
switch/comands	char[4]	„ON_“ - zapnout, „OFF“ - vypnout

\* každé téma začíná prefixem „SMpool/pool/“

Informační systém lze rozdělit na dvě části: řídicí server a webovou aplikaci. Server přebírá primární data z bazénového systému a vytváří nad nimi vyšší logiku. Konkrétně se jedná o přepočítání hydrostatického tlaku na výšku vodní hladiny oproti normálu, automatizaci spouštění čerpadla a ukládání naměřených dat. Druhá funkce serveru je perzistentní uložení dat pro webovou aplikaci. Server pro aplikaci uchovává data o uživatelích, konfiguraci uživatelských účtů a nastavení systému. Tyto informace jsou poskytovány skrze REST API webové aplikaci. Webová aplikace má na starost interakci s uživatelem skrze grafické uživatelské rozhraní (GUI). Komunikace mezi všemi částmi systému je zobrazena na obr. 4.1.

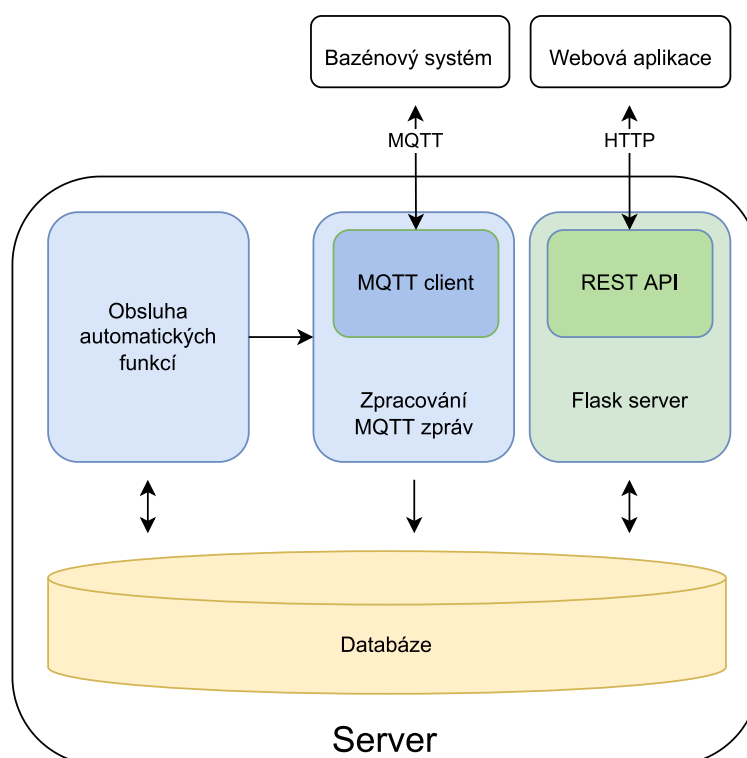




Obrázek 4.1: Přehled komunikace mezi jednotlivými částmi navrhovaného systému.

## 4.4 Řídící server

Řídící server je na nejvyšší úrovni složen z REST API serveru, části pro zpracování MQTT zpráv, obsluhy automatických funkcí a databáze. Schéma je zobrazeno na obrázku 4.2.



Obrázek 4.2: Základní blokové schéma serveru obsluhující bazénový systém a webovou aplikaci.

## Obsluha automatických funkcí

Obsluha automatických funkcí je proces, který kontroluje stav bazénového systému, a podle uživatelem nastavené konfigurace reaguje na vzniklé události. V současném stavu jsou implementovány dvě automatické funkce. První funkce umožňuje nastavení dvou časů během dne, přičemž v jednom čase dojde ke spuštění čerpadla a ve druhém k jeho vypnutí. Druhá funkce také provádí automatické ovládání čerpadla, avšak tentokrát je pro spuštění potřeba splnit dvě podmínky – intenzita světla musí překročit uživatelem nastavený práh a teplota vody musí být pod požadovanou teplotní hranicí. Kdykoli dojde k porušení jedné z podmínek, čerpadlo se vypne.

## Zpracování MQTT zpráv

Tento blok je zodpovědný za příjem a odesílání zpráv bazénovému systému. Při obdržení nové informace od senzoru, tuto informaci ukládá do databáze. V případě, kdy dostává zprávu od spínacího modulu ohledně naměřeného tlaku, využívá data z databáze, získanou hodnotu převádí na výšku vodního sloupce a vysílá zprávu pod novým MQTT tématem pro webovou aplikaci.

## Návrh databáze

Databáze slouží jako jediné permanentní uložení v celém navrhovaném systému. Uchovávají se v ní data o uživateli, konfigurace automatických funkcí a poslední naměřená data. Uložení posledních naměřených dat je nutností zejména z toho důvodu, že uživatel po zapnutí GUI dostane okamžitě informace o stavu bazénu a nebude muset čekat na poskytnutí informací od senzorů, což by mohlo trvat až minuty. Níže je možné si prohlédnout ER diagram na obrázku 4.3.

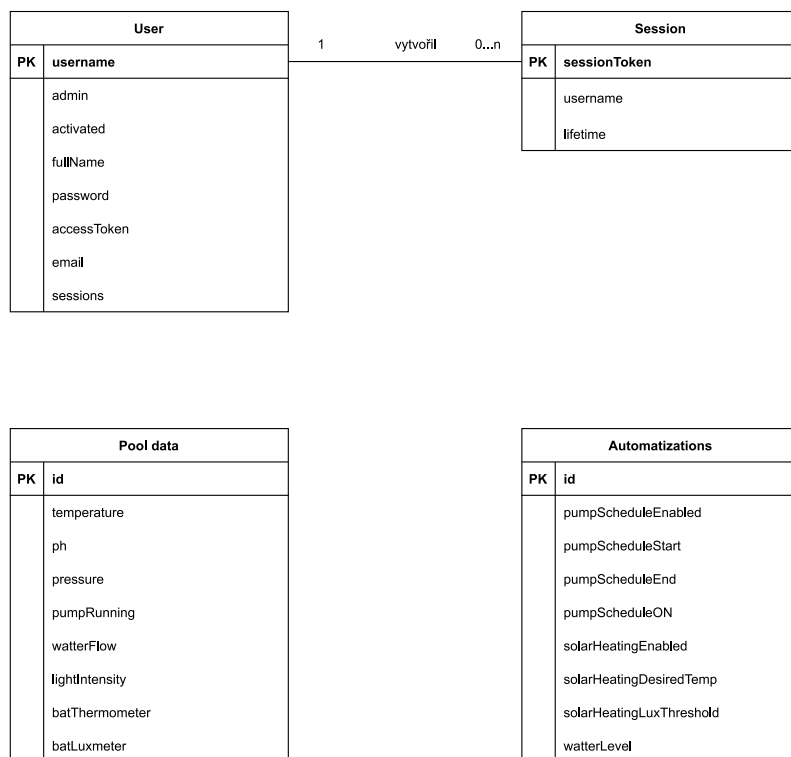
## REST API server

Tento server vytváří webové aplikace webovou službu, která pro GUI, která je běží uživateli v prohlížeči, poskytuje skrze REST API informace o aktuálním stavu celé aplikace. Toto API také uživatelům umožňuje provádět nastavení automatických funkcí a editovat vlastní uživatelská data.

API je tvořeno několika přístupovými body, které jsou definovány pomocí URI. Přehled těchto koncových bodů a jejich význam je popsán v tabulce 4.2

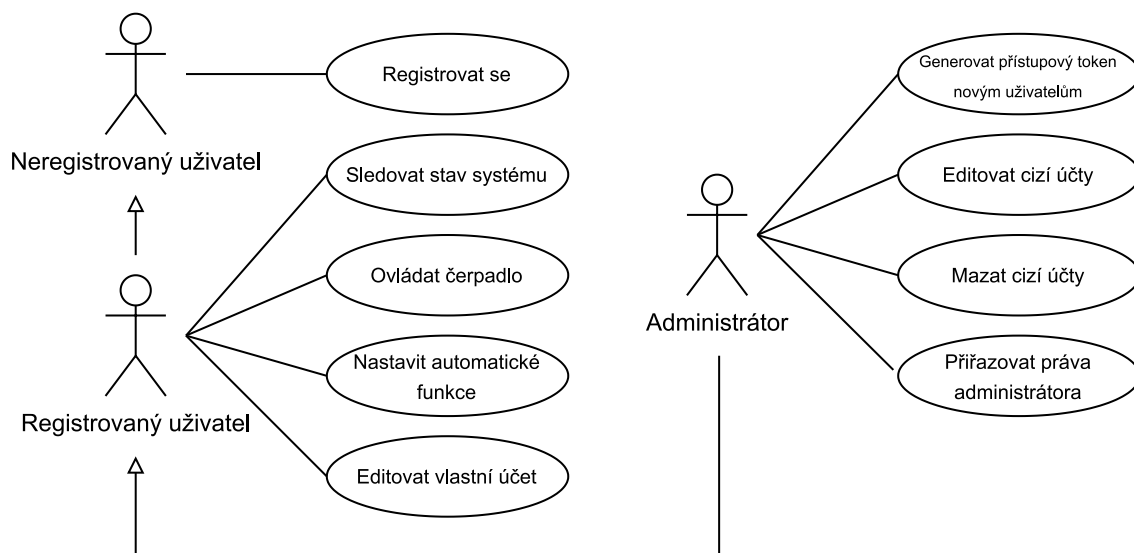
Tabulka 4.2: Přehled přístupových bodů serveru.

Endpoint	Význam
/api/infoAdmin	Základní informace o administrátorovi aplikace
/api/registerAdmin	Registrace administrátora pokud neexistuje
/api/registration	Registrace nového uživatele
/api/login	Po autentizaci uživatel získá práva pro chráněné endpointy
/api/admin/createUser	Administrátor vytváří pro uživatele přístupový token
/api/admin/allUsers	Administrátor získá informace o všech uživateli
/api/pool/overview	Přihlášený uživatel získá informace o stavu bazénu
/api/pool/automaticFunctions	Informace a nastavení automatických funkcí



Obrázek 4.3: ER diagram databáze.

## 4.5 Návrh grafického uživatelského rozhraní



Obrázek 4.4: Diagram případu užití webové aplikace.

## Kapitola 5

# Technická realizace řídicího systému

V této kapitole je na začátku popsán online nástroj, ve kterém probíhal návrh desek plošných spojů (PCB). Dále je navržen způsob krytí obvodů vůči vodě a dalším vlivům prostředí. Poté jsou v jednotlivých sekcích pomocí části schémat prezentovány logické celky obvodů, které jsou společné pro více modulů. Další sekce popisují vznik modulů spojením předem definovaných obvodů a popisují obvody specificky navržené pro konkrétní modul. Kompletní schémata obvodů jsou k nahlédnutí v příloze A. V poslední sekci je popsán proces tvorby návrhu desek plošných spojů na základě schémat.

### Použitý designer desek plošných spojů

Pro tvorbu PCB byl zvolen nástroj EasyEDA <sup>1</sup>. EasyEDA je online nástroj zdarma, který kromě zmíněného umožňuje generovat 3D modely návrhu, automatické navržení vodivých cest mezi součástkami a také poskytuje komunitní knihovnu se součástkami – zde je však třeba brát zřetel na to, že do ní mohl díl umístit kdokoli a je třeba jej překontrolovat. EasyEDA byla vybrána zejména kvůli jednoduchosti použití a rozsáhlé knihovně se součástkami.

### Způsob ochrany desek plošných spojů před vnějšími vlivy

Jako pouzdro na usazení obvodů byla původně zvažována varianta tisku na 3D tiskárně. Pouzdro by mohlo mít originální design a přesně by pasovalo na dané PCB. Od této varianty bylo ustoupeno z důvodu, že některé zařízení budou v přímém kontaktu s vodou a 3D tištěné obaly mají tendenci propouštět vodu. Nakonec byla zvolena varianta krytí polykarbonátovými krabičkami s certifikací IP68 <sup>2</sup>. Jelikož bylo skrz luxmeter potřeba, aby do interiéru krabičky dopadalo sluneční světlo kvůli inspekci stavu zařízení, byla u krabiček vybrána varianta s průhledným víkem.

Pro průchod sond a kabelů stěnou krabičky byly zvoleny panelové průchodky různých průměrů - opět s certifikací IP68. Průchodky je možné vidět na obrázku 6.2.

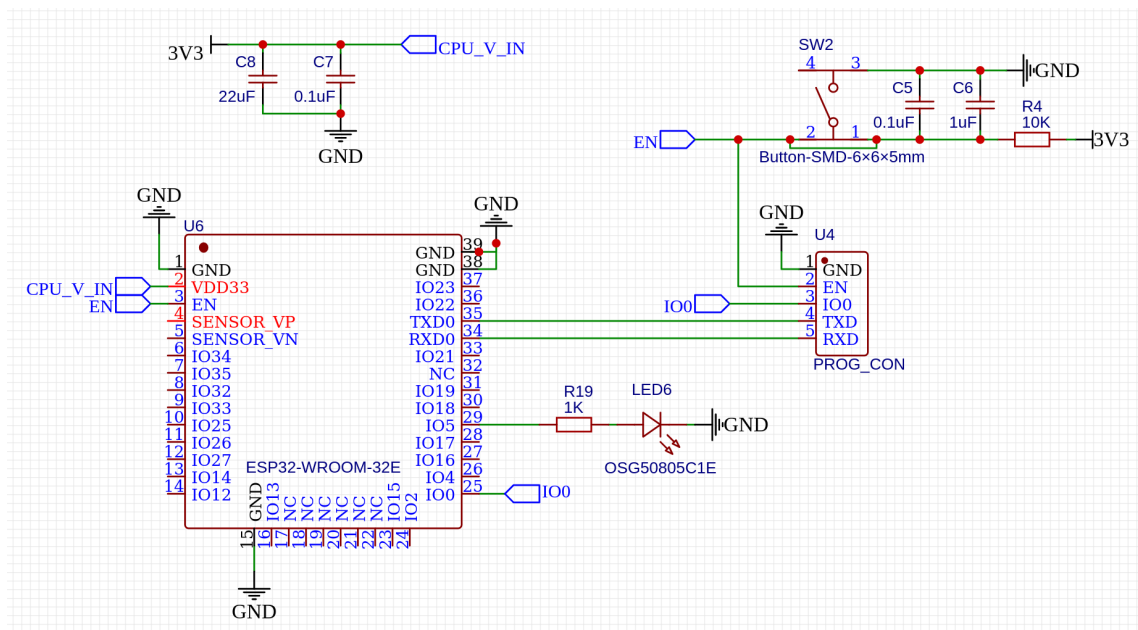
---

<sup>1</sup><https://easyeda.com/>

<sup>2</sup>Ochrana proti prachu a trvalému ponoření do vody za podmínek stanovených výrobcem.

## Návrh obvodu základní obsluhy mikrokontroléru

Pro všechny zařízení byla vybrána řídicí jednotka ESP32-WROOM-32E a to především díky výbornému poměru cena/výkon a vývojovému frameworku firmy ESP. Tento modul je z řady výrobků ESP32, jež je popsána v sekci 3.3.1. Z datasheetu pro vybraný modul bylo vyčteno minimální nutné zapojení – čtyři zemnicí vodiče a jeden vodič s napětím 3,3 V. Pro zajištění ochrany před fluktuacemi napětí byly přidány dva blokovací kondenzátory s hodnotami  $22\ \mu\text{F}$  a  $0.1\ \mu\text{F}$  (na schématu 5.1 nesou označení C8 a C7). Dále bylo třeba zajistit ukládání firmwaru do paměti mikroprocesoru. Nejprve je potřeba uvést zařízení do flash módu. Toho je docíleno sekvencí tří kroků. Výstup IO0 je třeba propojit se zemí a ponechat jej v tomto stavu, následně propojit se zemí výstup EN a nakonec vrátit do původního stavu IO0. Tento postup lze provádět manuálně za pomoci tlačítek nebo automaticky pomocí tranzistorů připojených k USB–UART převodníku. Pro snížení spotřeby a zjednodušení obvodu byla zvolena varianta, kdy se pro flashování použije zařízení s USB–UART převodníkem a zabudovanými tranzistory. Pro tuto variantu stačí na navrhovaném obvodu vyvést konektor pro UART rozhraní, zem, IO0 a EN. Na schématu 5.1 je tento konektor označený PROG\_CON. Pro základní interakci s mikrokontrolérem byla ještě přidána LED dioda a resetovací tlačítko.



Obrázek 5.1: Schéma zapojení mikroprocesoru spolu se zapojením led diody a resetovacím tlačítkem.

## Návrh bateriového obvodu

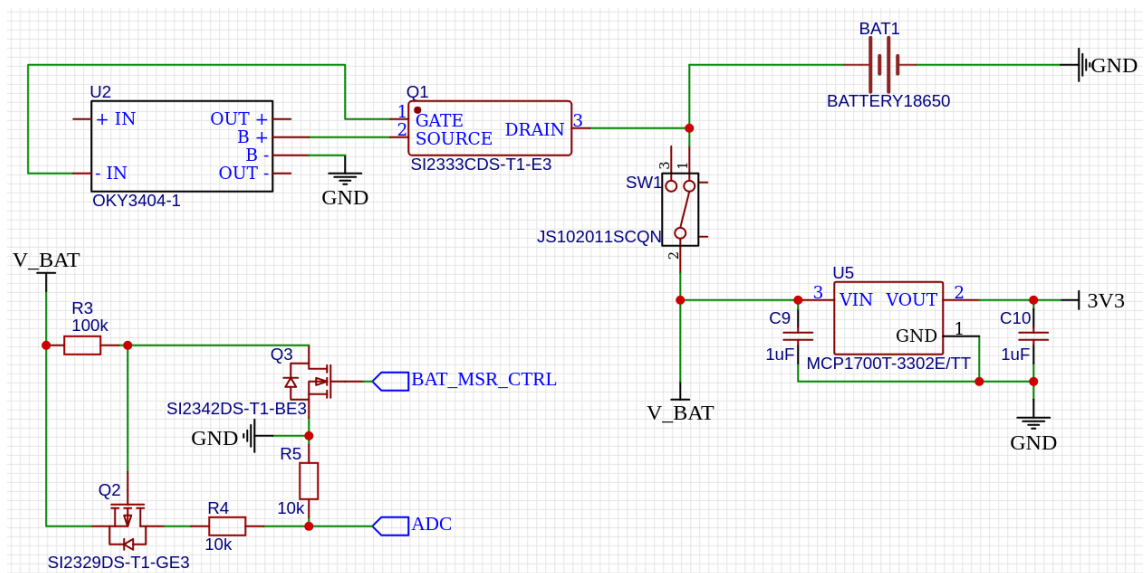
Pro Thermometer a Luxmeter bylo rozhodnuto, že příjem energie zajistí napájení z baterie. Vybrán byl Lithium-iontový (Li-ion) akumulátor INR18650-M36 s kapacitou 3450 mAh. V souvislosti s akumulátorem bylo potřeba vyřešit tři záležitosti – regulace napětí, nabíjení a měření zbývající kapacity akumulátoru.

Li-ion akumulátory poskytují napětí v rozmezí 4.2 V–3.5 V (v případě vybití pod hodnotu 3.5 V dochází k nenávratnému poškození článku). Zvolený mikrokontrolér potřebuje

vstupní napětí 3.3 V. Pro vyrovnání rozdílů a stabilizaci napětí byl použit LDO (Low Dropout) regulátor. Tento regulátor je možné najít ve schématu 5.2 pod označením U5. Význam tohoto typu regulátoru spočívá v tom, že dokáže regulovat napětí i pro malé rozdíly mezi vstupním a výstupním napětím.

Pro nabíjení byl vybrán obvod OKY3404-1. Ten skrze USB-C konektor umožňuje nabíjení pomocí běžně používaných nabíjecích adaptérů pro mobilní telefony. Tento obvod nabíjí rychlostí 1000 mA. Aby nedocházelo k postupnému vybíjení baterie klidovým odběrem tohoto obvodu, byl připojen k baterii přes tranzistor, který se sepne pouze v případě, že je připojen nabíjecí konektor.

Měření zbývající kapacity baterie je odvozeno z napětí, které baterie poskytuje. Vestavěný ADC převodník mikrokontroléru měří s garantovanou přesností v maximálním rozsahu 150 mV–2450 mV. Je tedy třeba napětí baterie analogicky převést do tohoto rozsahu. Za tímto účelem byl navržen dělič napětí, který snižuje měřené napětí o polovinu. Je realizovaný dvěma rezistory s odporem 10000 Ω. Na schématu 5.2 jsou k nalezení pod označením R4 a R5. Ačkoli tímto obvodem neprochází vysoký proud, v delším časovém horizontu by přesto měl vliv na vybíjení akumulátoru. Mezi tento obvod a akumulátor byl tedy vložen P-gate MOSFET tranzistor, který je spínáný přes N-gate MOSFET tranzistor mikroprocesorem. Ten tak činí pouze na dobu nezbytně nutnou pro měření napětí.

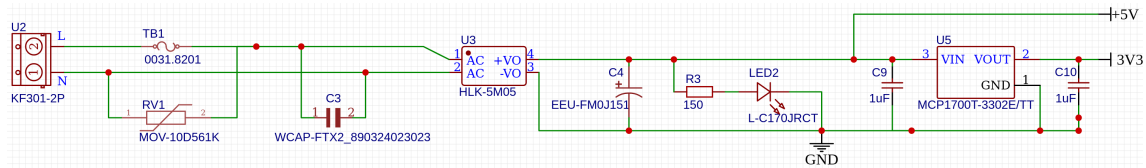


Obrázek 5.2: Schéma zapojení obvodu, obsluhující baterii.

## Návrh síťového obvodu

Zařízení Switch a Bridge musí mít neustále zaplé komunikační moduly, což zvyšuje jejich spotřebu a je vhodné napájet ze sítě. Switch spíná obvod, který vede elektrickou energii pod napětím 230 V a může být poháněn z tohoto zdroje. V tomto zařízení jsou zároveň součástky, které vyžadují napětí 5 V. Regulace napětí je tedy navržena tak, že 230 V je zregulováno na 5 V pomocí zdroje Hi-Link HLK-5M05, z této větve jsou napájeny potřebné součástky a dále je 5 V zregulováno na 3 V stejným způsobem, který je popsán v druhém odstavci předchozí sekce 5. Pro zvýšení bezpečnosti byly před zdroj dle datasheetu [7] předřazeny bezpečnostní prvky: varistor, proudová pojistka a kondenzátor. Dále byla za

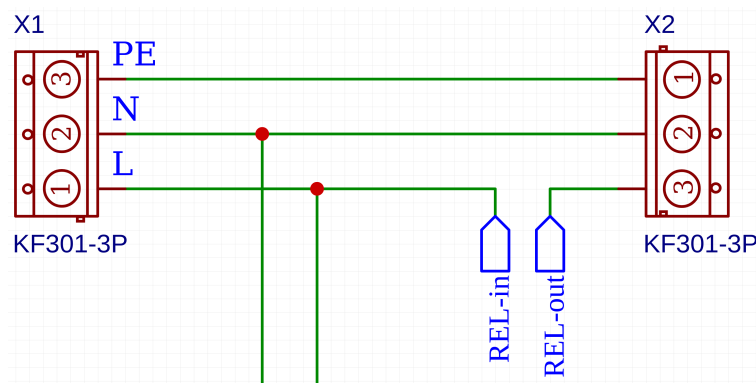
zdroj umístěna červená led dioda pro explicitní indikaci, že je obvod pod proudem. Obvod pro regulaci napětí je zakreslen ve schématu 5.3.



Obrázek 5.3: Schéma obvodu, regulujícího napětí z 230 V na 3.3 V

## Návrh spínacího modulu

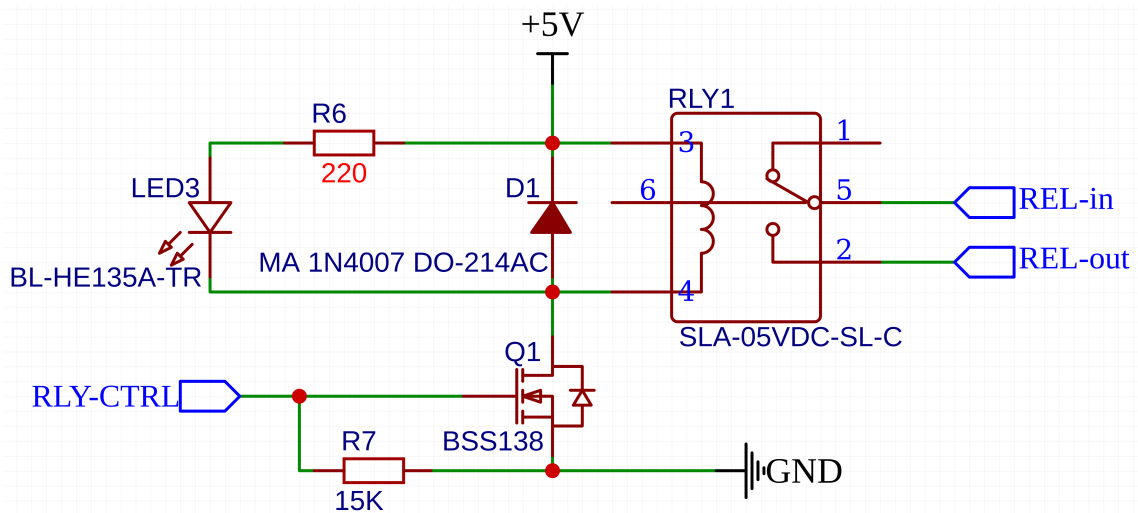
Spínání čerpadla pomocí switche je realizováno tak, že je switch napájen z elektrické sítě a dále napájí v případě potřeby čerpadlo. Vstupní a výstupní vodiče jsou připojeny k desce pomocí svorkovnic TB002-500-03BE, které jsou vyrobeny pro jmenovité napětí 300 V a jmenovitý proud 15 A. Fázový vodič je veden skrze relé SLA-05VDC-SL-C, které je dimenzováno až na 30 A při 250 V. Toto relé je zapojeno tak, že je ve výchozím stavu vodič rozepnutý a v případě potřeby sepne obvod. Pokud si spočítáme (dle vzorce pro výpočet elektrického výkonu  $W = I \cdot V$ ) výkon mezních hodnot, vyjde nám, že svorkovnice je schopna přenášet výkon až 4500 W a relé je schopno přenášet výkon až 7500 W. Obvod byl navrhován pro referenční čerpadlo o výkonu 500W. Svorkovnice i relé jsou tedy značně naddimenzovány a k překročení mezních hodnot tedy nedochází.



Obrázek 5.4: Ukázka propojení svorkovnic. Z vodiče vedoucího fází a nulového vodiče je napájen obvod. Ke spínání obvodu je provedeno na fázovém vodiči pomocí relé.

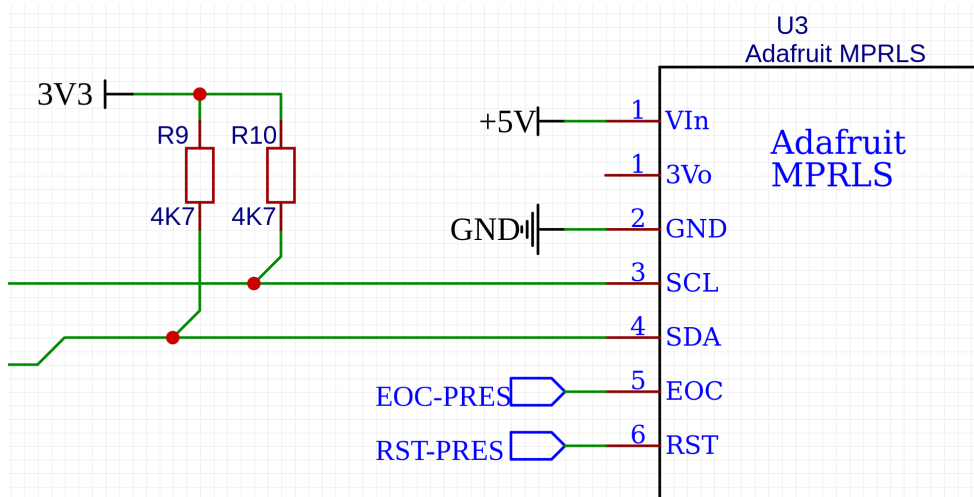
Pro sepnutí zmíněného relé je potřeba na vývod k jeho interní cívce přivést napětí 5 V. Napětí je přiváděno z 5 V větve síťového obvodu popsaného výše. Uzemnění je provedeno skrze N-gate MOSFET tranzistor BSS138, který je ovládaný GPIO vývodem mikrokontroléru. Při vypínání a spínání relé vznikají napěťové špičky, které je třeba tlumit, aby nedošlo k poškození tranzistoru, toho je dosaženo připojením diody (ve schématu 5.5 nese označení D1) paralelně k cívce s opačnou polaritou. K relé byla ještě připojena paralelně dioda. Díky tomu uživatel pozná, zdali je obvod sepnutý, či nikoli.

Pro měření hydrostatického tlaku bylo potřeba vybrat součástku, kterou bude možné připojit na hadičku, která bude dimenzována na dostatečný tlak a bude odolná vůči vlhkosti. Vybrána byla součástka MPRLS od firmy Adafruit. U této součástky je k měřicímu čipu



Obrázek 5.5: Schéma zapojení relé spínajícího čerpadlo.

od výrobce připevněno kovové hrdlo, na které je možné připevnit hadičku. Tato součástka je schopna měřit tlak až do hodnoty 1720 hPa, což po odečtení atmosferického tlaku a po přepočtu odpovídá zhruba tlaku sedmimetrového sloupce vody. Výrobce navíc daný díl navrhl pro styk s vodou a senzor uvnitř hrdla ochránil silikonovým gelem. [10] Z čipu se získávají data pomocí sběrnice I<sub>2</sub>C. Jak bylo popsáno v sekci 3.1.3, pro tuto sběrnici jsou potřeba dva vodiče s pull-up rezistory. Byly zvoleny rezistory s odporem 4200 Ω, které se pro daný účel obvykle používají.



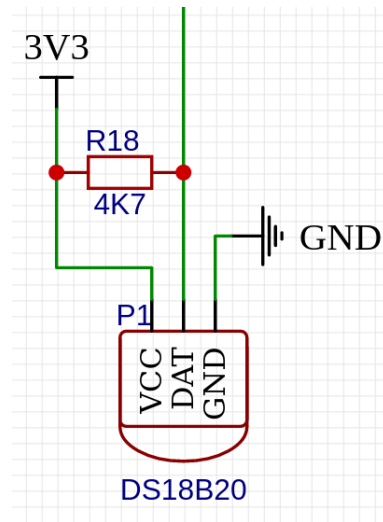
Obrázek 5.6: Schéma zapojení senzoru Adafruit MPRLS. Tento senzor s vysokou přesností měří tlak. K mikrokontroléru je připojen přes sběrnici I<sub>2</sub>C

### Návrh plovacího modulu

Pro měření teploty byl zvolen digitální senzor DS18B20 s vodotěsným pouzdem. Tento senzor je připojen pomocí tří vodičů. Jeden vodič je zemní, druhý napájecí a třetí datový.



Z dokumentace [19] bylo zjištěno, že k přenosu dat se používá technologie 1-Wire. Ta vyžaduje připojený pull-up rezistor k datovému vodiči. Byl tedy přidán rezistor spojující napájecí a datový vodič. Toto zapojení je zobrazeno na schématu 5.7



Obrázek 5.7: Schéma zapojení senzoru teploty DS18B20. Pro datový vodič musí být použit pull-up rezistor

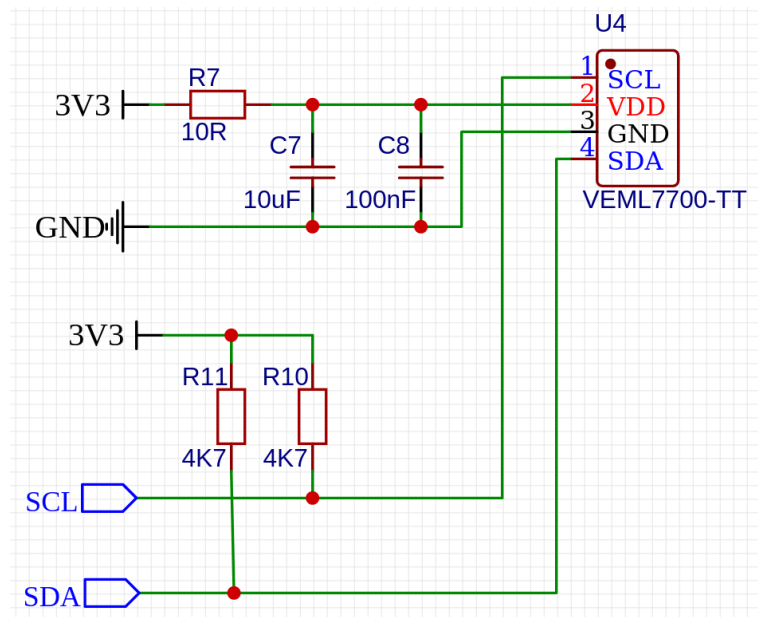
Měření pH je zajištěno pomocí sady Gravity: Analog pH Sensor / Meter Pro Kit V2. Tento typ měření byl zvolen z důvodu finanční dostupnosti a dostatečné přesnosti. Konkrétně tato sada byla vybrána, jelikož, na rozdíl od konkurenčních sad, které potřebují napájet napětím 5 V, této sadě stačí napětí 3,3 V. Tím pádem nebylo potřeba převádět napětí z 3,3 na vyšší a došlo ke zjednodušení návrhu. Druhým důvodem je, že tato sada má levnější variantu, která má stejné rozhraní. Levnější varianta má menší životnost, avšak pro vývoj, během kterého je vyšší šance jejího poškození, bude postačovat. Sada se skládá ze dvou částí - sondy a analogového převodníku. Připojení převodníku je velice jednoduché. Kromě napájení a zemnění stačí připojit výstupní vodič převodníku na výstup mikrokontroléru, který umožňuje použití ADC převodníku.

### Popis vybraného luxmetru

Pro měření sluneční intenzity bylo třeba vybrat senzor, který dokáže naměřit hodnotu maximálního slunečního osvětlení, které může nabývat hodnoty až 120 000 lx. Vybrán byl senzor VEML7700-TT, jenž poskytuje naměřenou hodnotu s 16 bitovou přesností. Komunikace probíhá stejně jako u senzoru tlaku pomocí I<sub>2</sub>C sběrnice. U napájení byly přidány blokovací kondenzátory dle doporučení technické dokumentace [20].

### Návrh desek plošných spojů

Nejprve bylo potřeba ke všem dílům ze schématu přiřadit jejich odpovídající model (footprint) pro návrh PCB. Většina dílů z knihovny má tyto footprints již přiřazeny, avšak některé byly chybné, nebo chyběly úplně. V takovém případě je potřeba jejich vlastní modelace. V dalším kroku EasyEda po navolení základních parametrů návrhu vyskládá všechny díly do prostoru a uživatel rozhoduje o jejich rozmístění dle potřeby. Pro vytvoření cest mezi



Obrázek 5.8: Na tomto schématu je zapojen senzor VEML7700-T pomocí sběrnice I<sub>2</sub>C. U této sběrnice je třeba jak na vodiči SCL, tak na SDA použít pull-up rezistory.

díly byl původně použit nástroj Auto Router. Ten propojování dílů automatizuje, avšak výsledek nebyl moc uspokojivý. Spojům chybí logika uspořádání, přehlednost a jednoduchost. Nakonec byly cesty vytvořeny manuálně.

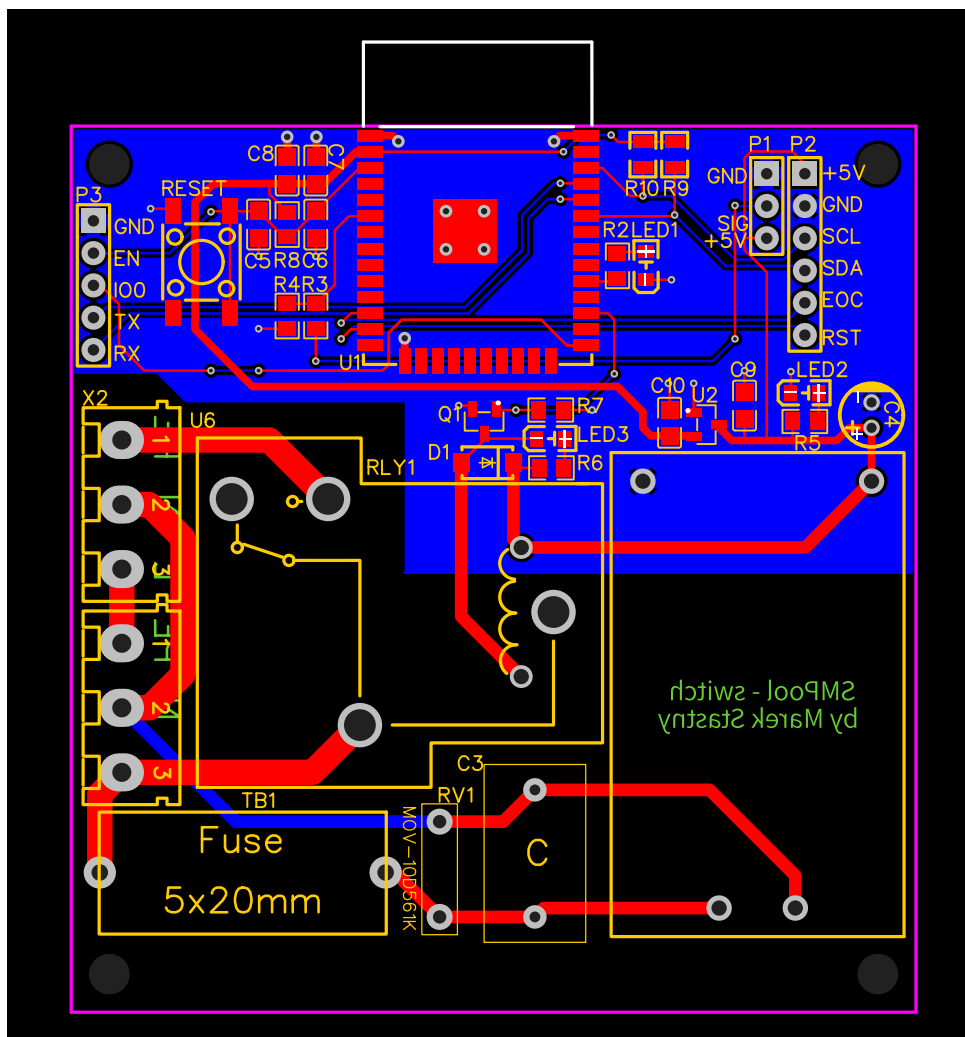
Při návrhu částí, kde je veden proud pod napětím 230 V bylo kvůli bezpečnosti třeba dbát na dostatečnou šířku cest a jejich vzdálenost od sebe. Minimální hodnoty pro tyto parametry definuje standard IPC-2221<sup>3</sup>. Pro výpočet šířky cest byl použit online nástroj PCB Trace Width Calculator<sup>4</sup>. Pro výpočet minimální vzdálenosti byla použita online kalkulačka<sup>5</sup>. Oba tyto nástroje počítají hodnoty na základě zmiňovaného standardu IPC-2221. Po zadání hodnot vyšla minimální nutná šířka cest při výšce vrstvy 35  $\mu\text{m}$  na 1,61 mm a mezera mezi cestami na 2,5 mm.

U blokovacích kondenzátorů je třeba dodržovat dvě základní pravidla. Délka cesty mezi kondenzátory a součástkou by měla být co nejkratší a pořadí kondenzátorů u součástky by mělo být řazeno sestupně dle kapacity (kondenzátor s nejnižší kapacitou těsně u součástky). Na obrázku 5.9 jsou takto umístěny například kondenzátory C7 a C8. Návrhy všech čtyř modulů jsou k nahlédnutí v příloze B.

<sup>3</sup>[http://www-eng.lbl.gov/~shuman/NEXT/CURRENT\\_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](http://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf)

<sup>4</sup><https://www.digikey.cz/en/resources/conversion-calculators/conversion-calculator-pcb-trace-width>

<sup>5</sup><https://www.smps.us/pcbtracespacing.html>



Obrázek 5.9: Návrh desky plošných spojů pro spínací jednotku. Ve spodní části návrhu jsou součástky, které pracují s vysokým napětím, zde bylo třeba dbát na dostatečnou vzdálenost mezi cestami. V horní části obrázku je umístěn mikrokontrolér s výstupy po periferní zařízení.

# Kapitola 6

## Implementace

Tato kapitola je rozdělena do čtyř sekcí. V sekci 6.1 je popsána fyzická kompletace bazénových modulů. V sekci 6.2 jsou pro každý modul popsány použité knihovny, důležité koncepty a ukázány části nejdůležitějších funkcí. V sekci 6.3 je věnována pozornost implementaci serverové části a v sekci 6.4 je popsáno grafické uživatelské rozhraní vytvořené frameworkem Angular.

### 6.1 Sestavení bazénových modulů

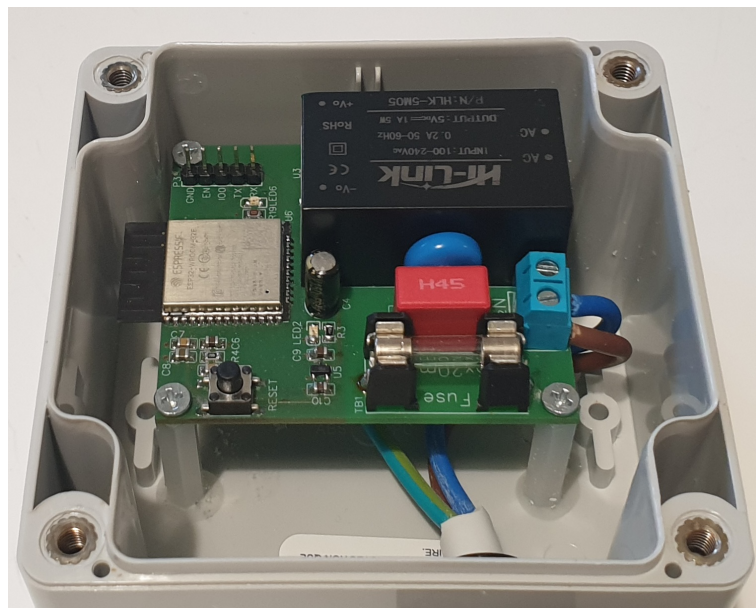
Na základě návrhů z kapitoly 5 byly firmou Gatema <sup>1</sup> vytvořeny čtyři desky plošných spojů. Ty byly následně za pomoci pájení pájecí pastou osázeny součástkami. U modulu bridge bylo zjištěno, že varistor (na návrhu 6.1 označený RV1) umístěný těsně mezi sousední součástky je ve skutečnosti mírně vyrotovaný a je velice obtížné jej umístit do připraveného obdélníkového prostoru. Na základě tohoto poznatku byl návrh upraven.

Pro uchycení desek do ochranné krabičky bylo rozhodnuto o připevnění desek pomocí distančních sloupků s vnitřním závitem. Z toho důvodu byly do rohů desek navrhnuté otvory pro šroubky o průměru 3 mm. K desce byly pomocí šroubků připevněny sloupky, na jejich základnu bylo nanášeno epoxidové lepidlo a deska i s nožičkami byla postavena na požadované místo v krabičce. Takto došlo k pevnému spojení desky s krabičkou. Po zaschnutí lepidla bylo potřeba odšroubovat rohové šrouby pro vyjmutí desky.

Pro prostup kabelů a senzorů krabičkou bylo nejprve potřeba vyvrtat otvory pomocí stromečkového vrtáku. Do vzniklých otvorů byly umístěny průchodky do panelu, těmi protaženy kabely a následně utaženy. Speciálním případem je senzor teploty na plovacím modulu. Pro udržení ve větší vzdálenosti od modulu, byla do průchodky umístěna plastová trubička s vnitřním průměrem shodným s průměrem senzoru. Tou byl poté protažen kabel senzoru a samotný senzor byl u paty oblepen epoxidovým lepidlem a na těsně vsazen do trubičky. Pro zakrytí hrany přechodu mezi senzorem a trubičkou byl přechod ještě přetažen smršťovací bužírkou s lepidlem. Výsledek je možné prohlédnout na obrázku 6.2

---

<sup>1</sup><https://www.gatema.cz/>



Obrázek 6.1: Ukázka usazení bridge do ochranné krabičky pomocí distančních sloupků.



Obrázek 6.2: Ukázka prostupu senzoru teploty a pH sondy stěnou krabičky u plovacího modulu.

## 6.2 Vytváření firmwaru pro bazénové moduly

### ESP-IDF

Jako první je třeba zvolit programovací jazyk. Jeho volba byla ovlivněna převážně dvěma aspekty. Za prvé bylo třeba zvolit jazyk ve kterém lze najít knihovny pro všechny použité senzory. Druhý aspekt, který ovlivňuje použitý jazyk je ten, že vývojový framework musí daný jazyk podporovat. Nakonec byla zvolena kombinace jazyků C++/C, které oba

požadavky splňují. Tedy pro všechny použité senzory v nich existují knihovny pro jejich obsluhu.

Pro vývoj byl vybrán framework ESP-IDF. Toto vývojové prostředí <sup>2</sup> je oficiální framework firmy Espressif. Nabízí kompletní sadu nástrojů potřebnou pro vývoj firmwaru mikrokontroléru. Překlad je v tomto frameworku zajištěn pomocí CMake a Ninja Build. Velkou výhodou práce s tímto frameworkem je, že obsahuje sadu připravených vzorů kódu, které jsou určeny jako výchozí bod pro psaní firmwaru a použití různých technologií.

## ESP-Now

Výchozím bodem pro implementaci komunikace mezi moduly bylo využití výchozího kódu <sup>3</sup> pro využití technologie ESP-Now, který je součástí použitého vývojového frameworku. Tento kód byl následně značně přepracován. Z tohoto kódu byla vytvořena komponenta s názvem `my_espnow`, která je využita ve všech čtyřech modulech. Pro nastavení komunikace touto komponentou je nejprve třeba zavolat funkci `myEspNowSetup`. Tato funkce provede nastavení Wi-Fi na mikrokontroléru a inicializuje veškeré proměnné. Této funkci je potřeba předat odkaz na strukturu `send_param`, která obsahuje veškeré nastavení pro posláni konkrétní zprávy, včetně ukazatele na samotný obsah zprávy a informaci o délce této zprávy. Tato struktura musí být předávána každé funkci z komponenty `my_espnow`. Pro posláni zprávy následně stačí zprávu uložit do struktury `send_param` a zavolat funkci `my_send`. Pro příjem zpráv je třeba vytvořit nové vlákno, ve kterém bude spuštěna funkce `my_espnow_task`. Tuto funkci je třeba si před použitím upravit a napsat vlastní kus kódu, který reaguje na příjem zprávy od konkrétního typu zařízení.

## Captive portal

System je připojen k internetu skrze modul bridge, který se po zapnutí musí připojit k Wi-Fi. Problém ohledně připojení se ke správnému Wi-Fi síti je řešen komponentou `WifiManager` <sup>4</sup>. Ta se pokusí připojit k dostupným Wi-Fi sítím a v případě neúspěchu vytváří vlastní AP (Access Point 3.2.1), ke kterému je možné se například pomocí chytrého telefonu připojit a skrze grafické uživatelské rozhraní nakonfigurovat připojení. Po úspěšném připojení ukládá konfiguraci do nevolatilní paměti, takže se po restartu zařízení opět může připojit. Základní použití `WiFiManageru` je zobrazeno ve výpisu 6.1.

```
1   initialize_output_pin(INFO_LED_PIN);
2   gpio_set_level(INFO_LED_PIN, LED_ON);
3   initArduino();
4   WiFiManager wm;
5   bool res;
6   res = wm.autoConnect("SMpool", "SMpoolpass1");
7   if(!res) {
8       ESP_LOGI("WiFiManager", "Failed to connect");
9       esp_restart();
10  }
11  else {
```

<sup>2</sup><https://www.espressif.com/en/products/sdks/esp-idf>

<sup>3</sup>[https://github.com/espressif/esp-idf/blob/master/examples/wifi/espnow/main/espnow\\_example\\_main.c](https://github.com/espressif/esp-idf/blob/master/examples/wifi/espnow/main/espnow_example_main.c)

<sup>4</sup><https://github.com/tzapu/WiFiManager/blob/master/WiFiManager.cpp>

```

12     ESP_LOGI("WiFiManager", "Connected to WiFi");
13     WiFi.mode(WIFI_AP_STA);
14     gpio_set_level(INFO_LED_PIN, LED_OFF);
15     bridge_core(); //Start bridge main job
16 }

```

Výpis 6.1: Fragment kódu pro spuštění WIFI manageru. Tato komponenta využívá Arduino komponentu, na řádce 3 je tedy inicializována. Aby uživatel poznal, že zařízení není připojeno, je zde implementována funkcionalita, kdy se po restartu rozsvítí informační LED dioda a je vypnuta až po úspěšném vytvoření spojení. Pro spuštění manageru je třeba nad instancí třídy `WiFiManager` zavolat funkci `autoConnect()`. Tato funkce přijímá jako volitelné parametry SSID a heslo vytvářeného AP.

### 6.3 Implementace REST API serveru

Tento server byl implementován pomocí mikroframeworku Flask <sup>5</sup>. Tento mikroframework je implementován v jazyce Python. Osobně jej rád používám díky přehledné dokumentaci a jednoduchosti použití. Pro vytvoření REST serveru, stačí do Flask aplikace importovat rozšíření `Flask-RESTful`. Po vytvoření objektu typu API je možné si nadefinovat přístupové URL. Ve výpisu 6.2 je možné si prohlédnout vytvoření koncového bodu `/api/pool/automaticFunctions`. Tento koncový bod je obsluhován objektem třídy `AutomaticFunctions`, kterou je třeba si nadefinovat. Tato třída musí mít pro každý typ HTTP požadavku metodu, která jej obsluhuje. Název metody se musí shodovat s typem požadavku, jen musí být psaný malými písmeny (tedy pro požadavek HTTP GET se obslužná metoda jmenuje `get`).

```

1     api.add_resource(AutomaticFunctions, '/api/pool/automaticFunctions')

```

Výpis 6.2: Ukázka vytvoření nového REST API koncového bodu. Tento endpoint je obsluhován objektem Třídy `User`.

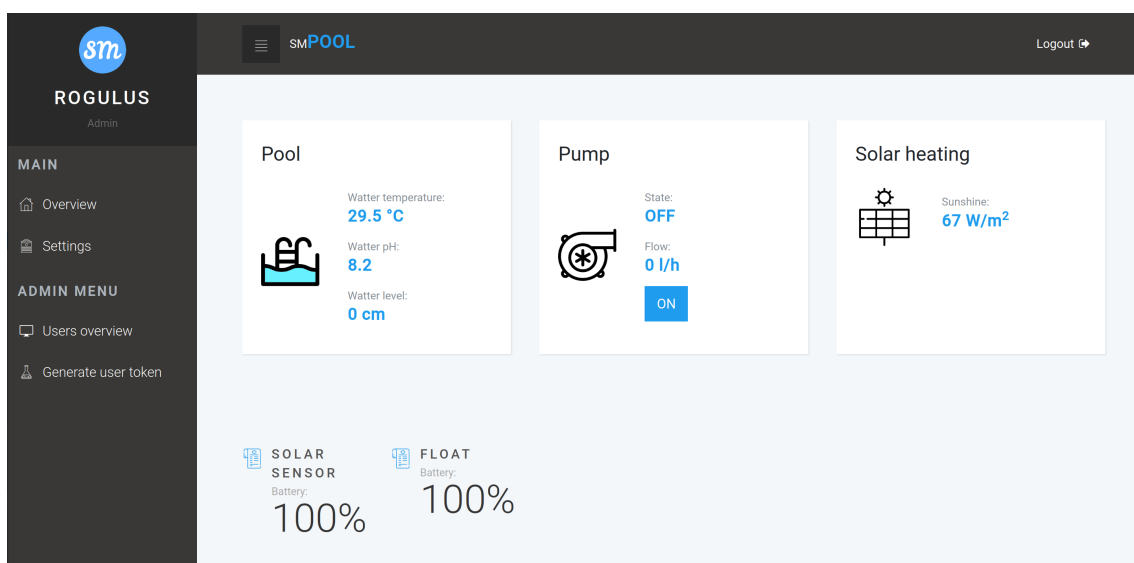
### 6.4 Implementace grafického uživatelského rozhraní

Webová aplikace pro interakci s bazénovým systémem je psána v open-source frameworku Angular <sup>6</sup> využívající jazyk TypeScript. Základním stavebním blokem aplikace v Angularu je komponenta. Každá komponenta má svoji vlastní HTML šablonu, pomocí které je pak v kombinaci s CSS stylováním vykreslována. Pro Design stránky byla použita již vytvořená šablona dostupná z této stránky: <https://bootstrapious.com/p/bootstrap-4-dashboard>.

Aplikace byla udělána formou dashboardu. Po přihlášení do aplikace je uživatel přesměrován na hlavní přehledovou stránku, kde se mu zobrazují hlavní data o systému. Tato úvodní stránka je k nahlédnutí na obrázku 6.3. Na obrázku je také vidět boční menu, které je horizontálně rozděleno na dvě části. Položky v horní části jsou přístupné všem uživatelům a položky v dolní části jsou pouze pro administrátora, který má právo editovat data veškerých uživatelů. Editace uživatelů je zobrazena na obrázku 6.5. V části menu pro všechny uživatele je položka `Settings`, kde lze nastavovat funkce pro automatizaci bazénu. Stránku s nastavováním si lze prohlédnout na obrázku 6.4.

<sup>5</sup><https://flask.palletsprojects.com/en/2.1.x/>

<sup>6</sup><https://angular.io/>



Obrázek 6.3: Obrázek hlavní stránky aplikace. Ve středovém prostoru jde vidět tři dlaždice a každá z nich informuje o naměřených hodnotách u různých prvků bazénu. Ve druhé řadě centrální plochy jsou zobrazeny technické informace, které podávají informaci o stavu modulů.



### Automatic functions

Pump schedule

Solar heating

Desired temperature: 32°C

Light intensity threshold: 236W/m<sup>2</sup>

Water level

Current water level: -2cm

Obrázek 6.4: Na tomto obrázku je vidět nastavování automatických funkcí. Funkce Pump schedule je vypnutá. U funkce Solar heating lze nastavovat požadovanou teplotu v bazénu, a minimální intenzitu světla, při kterém je ohřev možné spustit. U funkce Water level lze upravit výchozí výšku hladiny vody.

Users

Index	Username	Full name	Email	Admin	Activated		
0	iva55	iva kaspar	ivuska.kasparova@gmail.com	true	true	Edit	Delete
1	Rogulus	Marek Šťastný	marek.stny@gmail.com	true	true	Edit	Delete
2	ZjfdTQ			false	false	Edit	Delete

Edit user close

Username

Name and surname

Email

New password

Confirm new password

Obrázek 6.5: Přehled uživatelů poskytnutý administrátorům. Administrátor může smazat, či editovat všechny uživatele.

# Kapitola 7

## Testování

Testování je nedílnou součástí vývoje softwaru. V průběhu vývoje se provádí tzv. unit testy, čili testování menších celků kódu. Na konci vývoje, nebo po dokončení iterace vývoje se testuje systém jako celek. Během testování celku se ověřuje kompatibilita jednotlivých již otestovaných částí a hledají se poruchy nové, které vznikají například při vývoji neuvažovanou sekvencí kroků.

Při vývoji tohoto systému bylo postupováno obdobně. Navržené schémata obvodů byla nejprve přenesena na nepájivé pole, kde byla ověřena funkčnost konceptu. Aby bylo možné periferní zařízení připojit k mikrokontroléru, byla použita vývojová deska WeMos D1 R32 UNO ESP32. Díky těmto prototypům bylo možné začít psát firmware a ihned ověřovat jeho funkčnost.

Jelikož u měření senzorů pro dané použití není vyžadována vysoká přesnost, pro ověřování přesnosti měření senzorů byly použity poměrně jednoduché nástroje. Naměřená teplota na teploměru byla porovnána s teplotou naměřenou domácí meteostanicí. Průtokoměr byl validován experimentem, kdy byla napouštěna nádoba o známém objemu a při tom měřen čas, za který byla naplněna. Dle vzorce 7.1 byl pak snadno dopočítán reálný průtok a porovnán se skutečným. Z výsledků zaznamenaných v tabulce 7.1 lze vyčíst, že odchylka naměřených hodnot oproti skutečnému průtoku je do dvou procent, což považuji vzhledem k účelu měření za dostatečnou přesnost.

$$Q = \frac{V}{t} \quad (7.1)$$

Kde:

- $Q$ : průtok
- $V$ : objem kapaliny
- $t$ : čas

### 7.1 Ověření výdrže baterie plovacího modulu

Pro ověření výdrže baterie byl proveden pokus, jehož cílem bylo dokázat předpoklad, že akumulátorem poháněné moduly vydrží celou sezónu (180 dní) bez dobíjení. Dalším cílem bylo zjistit vývoj hodnoty napětí baterie během vybíjení. Tyto hodnoty byly následně využity pro převod napětí na procenta kapacity baterie.

Měření	Čas [s]	Reálný průtok [l/h]	Naměřený průtok [l/h]	odchylka
1.	64,8	88.8	90	101,35 %
2.	64,7	89.0	90	101,12 %
3.	64,8	88.8	90	101,35 %

Tabulka 7.1: Tabulka zaznamenaných hodnot během porovnávání naměřeného průtoku a skutečného průtoku. Z posledního sloupce lze vyčíst, že naměřený průtok má od reálného průtoku odchylku do dvou procent.

Výpočet výsledné výdrže modulu vychází z předpokladu, že během jednoho cyklu (měření modulu a spánek do dalšího měření) dochází k vybití baterie vlivem spotřeby modulu při měření, spotřeby modulu během spánku a samovolného vybíjení baterie. Dále předpokládá, že samovolné vybíjení baterie během měření, které trvá jednotky sekund je zanedbatelné. Na základě těchto předpokladů vznikl vzorec 7.2, který dává do vztahu vybití 100 % baterie počet cyklů a různé parametry spotřeby. z něj byl jednoduchými matematickými operacemi vyvozen vzorec 7.3, který je použit pro výpočet celkové výdrže modulu na základě naměřených hodnot.

Baterie byla nejdříve plně nabitá a poté byla 24 h volně ponechána. Během této doby dochází k intenzivnímu poklesu napětí, [4] které by jinak zkracovalo naměřené hodnoty. U prvního typu měření byla u senzoru nastavena doba pro hluboký spánek na 0 ms a poté se nechal senzor opakovaně provádět celý cyklus od probuzení, přes měření a odesílání dat, až po usnutí. Bridge při tom přijímal zprávy s naměřeným napětím baterie senzoru a tyto zprávy ukládal do souboru. Po vybití baterie je snadné sečíst uložené záznamy a získat tak počet probuzení senzoru. Samovolné vybíjení baterie je během tohoto měření zanedbatelné, a tak je možné odvodit kolik procent baterie se vybití během jednoho probuzení. Naměřená hodnota 144219 cyklů odpovídá vybití během jednoho cyklu o 0.000693389 %.

Dále bylo provedeno měření spotřeby celého obvodu během režimu spánku. Z naměřené hodnoty ( $323 \mu\text{A}$ ), známé kapacity baterie (3450 mAh) a doby cyklu lze opět snadno dopočítat procentuální úbytek kapacity pro jeden interval spánku (10 min). Spočtená hodnota odpovídá vybití 0.00001560386 % během jednoho usnutí.

Konečně je ještě potřeba spočítat samovolný úbytek baterie. Pro dobu, kdy je senzor aktivní, je tato hodnota zanedbatelná. Je tedy vypočtena jen pro dobu, kdy je senzor v režimu spánku. Pro použité Li-ion články se udává měsíční samovolný úbytek kapacity 1–2 % [4]. Budeme-li uvažovat 30 denní měsíc a počítat s pesimističtější hodnotou 2 %, dostáváme pro desetiminutový spánek samovolné vybití baterie 0.00046296296 %.

Výše vypočítané hodnoty lze nyní dosadit do rovnice 7.3. Po vypočtení rovnice byl získán výsledek 85327 cyklů což při desetiminutovém intervalu spánku odpovídá 592 dnům. Předpoklad byl tedy potvrzen a dokonce bylo zjištěno, že v případě potřeby je možné dobu spánku ještě zásadně zkrátit.

Rovnice pro výdrž senzoru:

$$c \cdot (u_m + u_s + u_b) = 100 \quad (7.2)$$

$$c = \frac{100}{u_m + u_s + u_b} \quad (7.3)$$

Kde:

$c$ : počet měřících cyklů (měření + spánek)

$u_m$ : procentuální úbytek kapacity baterie během jednoho měření

$u_s$ : procentuální úbytek kapacity baterie během jednoho intervalu spánku vlivem spotřeby senzoru

$u_b$ : procentuální úbytek kapacity baterie během jednoho intervalu spánku vlivem samovolného vybíjení

Pro ověření funkčnosti MQTT komunikace byl využit obline nástroj HiveMQ Websocket Client.<sup>1</sup> Pomocí něj je možné se přihlásit k jakémukoli MQTT tématu pro vybraný cluster a následně provádět inspekci zasílaných zpráv. Pomocí tohoto nástroje lze také posílat zprávy vlastní. Díky tomu bylo možné simulovat scénáře, které jsou možné, avšak se buďto dějí vyjimečně, nebo by jejich kontrolované vytvoření pomocí modulů vyžadovalo úpravu firmwaru.

REST API server byl testován za pomoci nástroje Postman<sup>2</sup>. Postman je komplexní aplikace, která mimo jiné poskytuje API klienta který přehledně zobrazuje veškeré nastavení komunikace a obsah zpráv. Pro testování bylo využito možnosti uložit všechny potřebné serverové požadavky a poté jejich jejich dotazováním se na server vytvářet scénáře.

## 7.2 Testování celého systému

Jelikož v době dokončování práce kvůli počasí nemohl být systém otestován v reálných podmínkách, byl za pomoci skleněné nádoby vytvořen alespoň model bazénu. Na tomto modelu byl systém testován. Celou testovací sadu je možné si prohlédnout na obrázku 7.1. Systém byl spuštěn pět dní a během této doby bylo pozorováno, zda nedojde k výpadku systému. Dále byly moduly namáčeny do vody a bylo zkoumáno, zda nedochází k průsaku vody. Zároveň byla spolubydlícím předložena ovládací aplikace a byla jim dána možnost si systém vyzkoušet. Byli instruováni, aby plnili různé scénáře, které pokrývali veškerou funkcionalitu aplikace a bylo pozorováno, zdali se v aplikaci orientují. Po dokončení těchto úkolů s nimi byl proveden rozhovor na téma, co by se dalo zjistit. Díky této proceduře byly zjištěny následující poznatky:

1. po registraci je vhodné uživatele automaticky přesměrovat na přihlašovací stránku, nebo rovnou do aplikace.
2. po vypnutí veškerých automatických funkcí neexistuje možnost, jak volbu potvrdit
3. nastavování času pomocí slideru je vhodné maximálně pro hodiny, jemnější jednotky je velice obtížné nastavit přesně
4. přehled naměřených hodnot tvrdí, že průtok je v litrech za minutu, ale ve skutečnosti je v litrech za hodinu

Chyby 1, 2 a 4 byly následně opraveny.

---

<sup>1</sup><https://websocketclient.hivemq.cloud>

<sup>2</sup><https://www.postman.com/>



Obrázek 7.1: Ukázka testování kompletního systému.

### 7.3 Další možný vývoj

Tento systém je dále možné rozšiřovat v několika různých směrech. V současném stavu se jedná o minimální možný použitelný produkt, avšak pokud by měl být uveden do komerční produkce, bylo by potřeba provést druhou iteraci vývoje a systém dovyvinout. Jeho slabiny spatřuji zejména ve třech oblastech – design bazénových modulů, potřeba instalace serveru a bezpečnost celého systému. Pro komerční použití by bylo potřeba navrhnout design obalu modulů a nejspíše jím i přizpůsobit tvar navržených desek. V současné době je celý server vázaný pouze na jeden bazénový systém. Do budoucna je možné celý systém rozvinout v cloudovou službu, přičemž každý zákazník bude mít na serveru vlastní databázi, automatizační logiku a MQTT témata pro komunikaci se zakoupenými moduly. Co se týče bezpečnosti, tak pro další použití považují téměř za nutnost komunikaci mezi serverem a webovou aplikací převést z protokolu HTTP na protokol HTTPS.

Tento systém by bylo dále vhodné rozšířit o další senzory a aktuátory. Jako konkrétní možné příklady zde uvedu měření ORP a automatické dávkování chemických roztoků. V návaznosti na přidávání nových modulů, by bylo vhodné, kdyby uživatelské rozhraní poskytovalo možnost přidávat a odebírat bazénové moduly, podle toho, které moduly jsou dostupné.

Další možnost vývoje spatřuji v navázání na MQTT rozhraní bazénového systému. Tento systém by mohl být, například pomocí aplikace Home Assistant, integrován do stávající sítě chytrých věcí v domácnosti a mohly by tak vznikat nové interakce mezi zařízeními.

## Kapitola 8

# Závěr

Tato práce vychází z vytyčeného cíle vytvořit funkční systém, který bude možné zaintegrovat do již koupených a používaných zahradních bazénů. Z počátku je čtenář seznámen se základními měřeními kvality vody – především měření pH a ORP. Dále je vypracován přehled dostupných sériových protokolů používaných ve vestavěných systémech a bezdrátových komunikačních protokolů používaných v oblasti IoT, které je pro daný systém možné použít. Na základě získaných poznatků a provedené rešerše stávajících řešení byly navrhnuty čtyři moduly založené na mikrokontrolérech řady ESP32. Tyto moduly obsluhují různé senzory a aktuátory a jako celek vytváří systém umožňující provádět některé nutné úkony pro údržbu bazénu. Tento systém využívá pro bezdrátovou komunikaci protokoly Wi-Fi a ESP-Now.

Na bazénový systém následně navazuje vytvořená obslužná aplikace, pomocí které uživatel interaguje s bazénovým systémem. Tato aplikace zajišťuje ukládání perzistentních dat, umožňuje zjišťovat aktuální naměřené hodnoty, ovládat aktuátory systému a nastavovat automatické funkce. Tato aplikace se skládá ze dvou částí REST API serveru implementovaného pomocí mikro frameworku Flask a z webové aplikace implementované pomocí frameworku Angular. Proces návrhu a implementace je popsán v druhé polovině této práce.

V závěru tohoto textu je popsán proces testování jednotlivých částí systému a následně testování celku. Tímto testováním je prokázána funkčnost výsledného produktu. Navíc bylo odhaleno několik menších chyb ve webové aplikaci, které byly následně odstraněny. Také bylo potvrzeno, že moduly poháněné baterií vydrží v provozu požadovaných 180 dní, a dokonce bylo vypočteno, že jejich výdrž je téměř dva roky. Na konci práce jsou navrženy různé postupy, kterými je možné vzniklý systém dále rozvíjet.



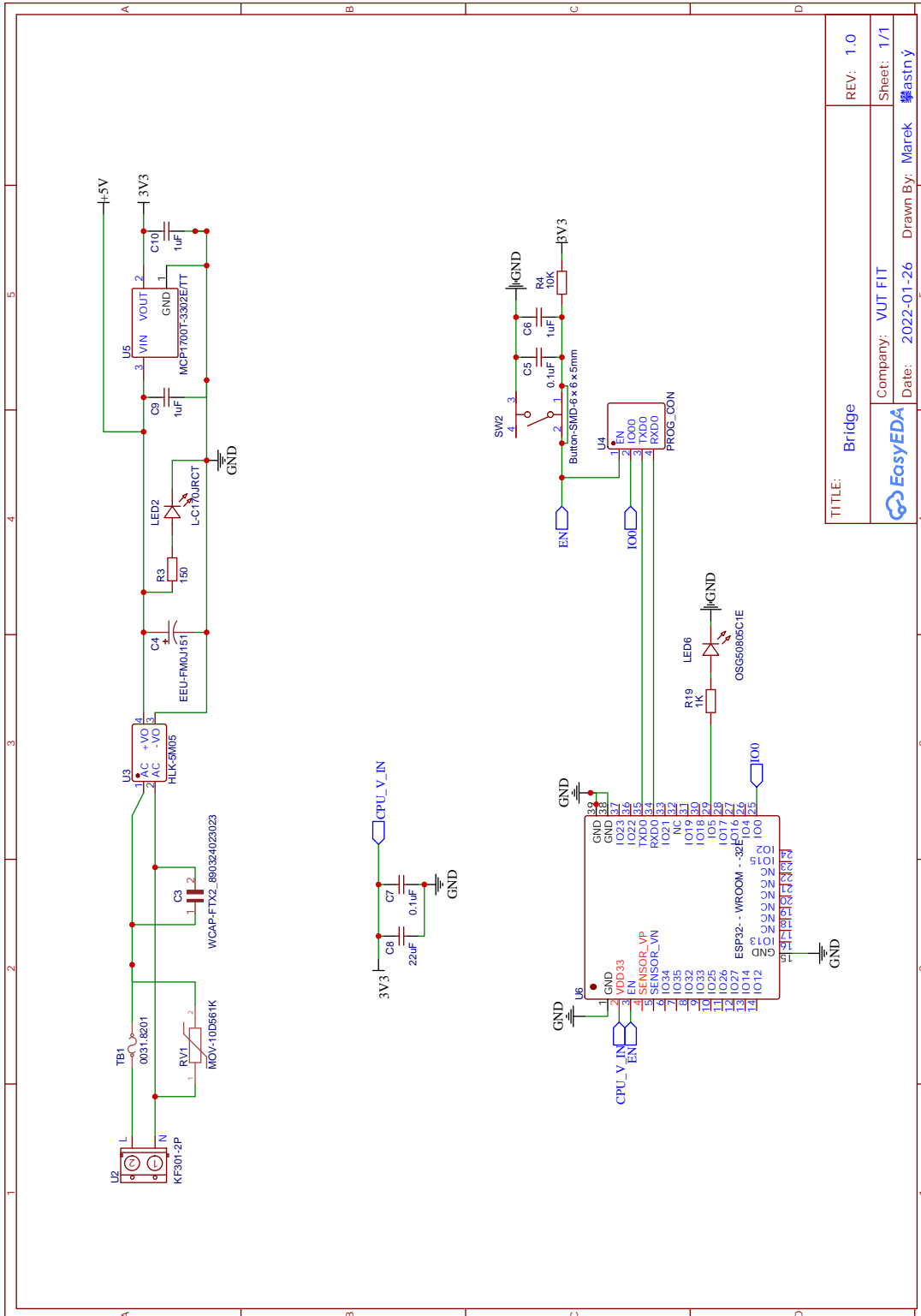
# Literatura

- [1] CHMELAŘOVÁ, G. *Návrh úsporných IoT senzorů a sítě chytré domácnosti*. Brno, CZ, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/21338>.
- [2] ESPRESSIF. *ESP32 Series Datasheet* [online]. [cit. 2022-04-08]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
- [3] ESPRESSIF. *ESP-NOW User Guide* [online]. Espressif, 2016 [cit. 2022-04-06]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp-now\\_user\\_guide\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp-now_user_guide_en.pdf).
- [4] GROUP, B. U. *What does Elevated Self-discharge Do?* [online]. 2021 [cit. 2022-04-21]. Dostupné z: <https://batteryuniversity.com/article/bu-802b-what-does-elevated-self-discharge-do>.
- [5] HARVEY, D. *Potentiometric Methods* [online]. 2021 [cit. 2022-04-04]. Dostupné z: [https://chem.libretexts.org/Bookshelves/Analytical\\_Chemistry/Analytical\\_Chemistry\\_2.1\\_\(Harvey\)/11%3A\\_Electrochemical\\_Methods/11.02%3A\\_Potentiometric\\_Methods](https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Analytical_Chemistry_2.1_(Harvey)/11%3A_Electrochemical_Methods/11.02%3A_Potentiometric_Methods).
- [6] HELMENSTINE, A. M. *Definition and Examples of Acid-Base Indicator* [online]. 2020 [cit. 2022-04-03]. Dostupné z: <https://www.thoughtco.com/definition-of-acid-base-indicator-604738>.
- [7] HI LINK. *5W Ultra small series power module, 5M03/5M05/5M09/5M12* [online]. [cit. 2022-04-023]. Dostupné z: [https://pajenicko.cz/index.php?route=product/product/get\\_file&file=1912111437\\_HI-LINK-HLK-5MXX.pdf](https://pajenicko.cz/index.php?route=product/product/get_file&file=1912111437_HI-LINK-HLK-5MXX.pdf).
- [8] HILLAR, G. C. *MQTT Essentials - A Lightweight IoT Protocol*. 1. vyd. Packt Publishing Ltd., 2017. ISBN 978-1-78728-781-5.
- [9] HIVEMQ. *MQTT & MQTT 5 Essentials*. 1. vyd. HiveMQ, 2020. ISBN 978-3-00-067913-1.
- [10] HONEYWELL. *MPR Series, MicroPressure Board Mount Pressure Sensors Compact, High Accuracy, Compensated/Amplified* [online]. [cit. 2022-04-025]. Dostupné z: <https://docs.rs-online.com/174f/A700000006835400.pdf>.
- [11] KUROSE, J. F. a ROSS, K. W. *Computer Networking*. 7. vyd. Pearson, 2017. ISBN 978-0-13-359414-0.
- [12] LIPTÁK, B. G. a VENCZEL, K. *Analysis and Analyzers: Volume II*. Boca Raton: CRC Press, 2017. ISBN 1498727646.

- [13] MALONE, L. J. a DOLTER, T. *Basic Concepts of Chemistry*. 8. vyd. John Wiley and Sons, Inc., 2010. ISBN 978-0-471-74154-1.
- [14] NXP. *88MW320-88MW322* [online]. [cit. 2022-04-08]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/88MW32X-DS.pdf>.
- [15] ROBERTAZZI, T. G. *Introduction to Computer Networking*. 1. vyd. Springer, 2017. ISBN 978-3-319-53102-1.
- [16] SCIENTIFIC, A. *Understanding ORP Measurements for Chlorine in Pools* [online]. [cit. 2022-04-21]. Dostupné z: <https://atlas-scientific.com/blog/orp-measurement-for-chlorine/>.
- [17] TAHIR, M. a JAVED, K. Serial Communication Interfaces. In: *ARM Microprocessor Systems*. CRC Press, 2017, s. 367–438. ISBN 9781482259384.
- [18] TEAM, E. B. P. *The Top 5 Reasons Pool pH is Important* [online]. 2021 [cit. 2022-04-03]. Dostupné z: <https://executivebluepools.com/the-top-5-reasons-pool-ph-is-important/>.
- [19] UMW. *UMW DS18B20* [online]. [cit. 2022-04-023]. Dostupné z: [https://www.laskakit.cz/user/related\\_files/umw\\_ds18b20\\_en.pdf](https://www.laskakit.cz/user/related_files/umw_ds18b20_en.pdf).
- [20] VISHAY. *High Accuracy Ambient Light Sensor With I 2C Interface* [online]. [cit. 2022-04-025]. Dostupné z: <https://www.vishay.com/docs/84323/designingveml7700.pdf>.
- [21] ČESKO. *Vyhláška č. 238/2011 Sb.* [online]. 2011 [cit. 2022-04-03]. Dostupné z: <https://www.zakony.cz/zakony/2011/201/zakon-238-2011-Sb-vyhlaska-o-stanoveni-hygienickych-pozadavku-na-koupaliste-sauny-a-hygienicke-limity-pisku-v-piskovi-SB2011238>.

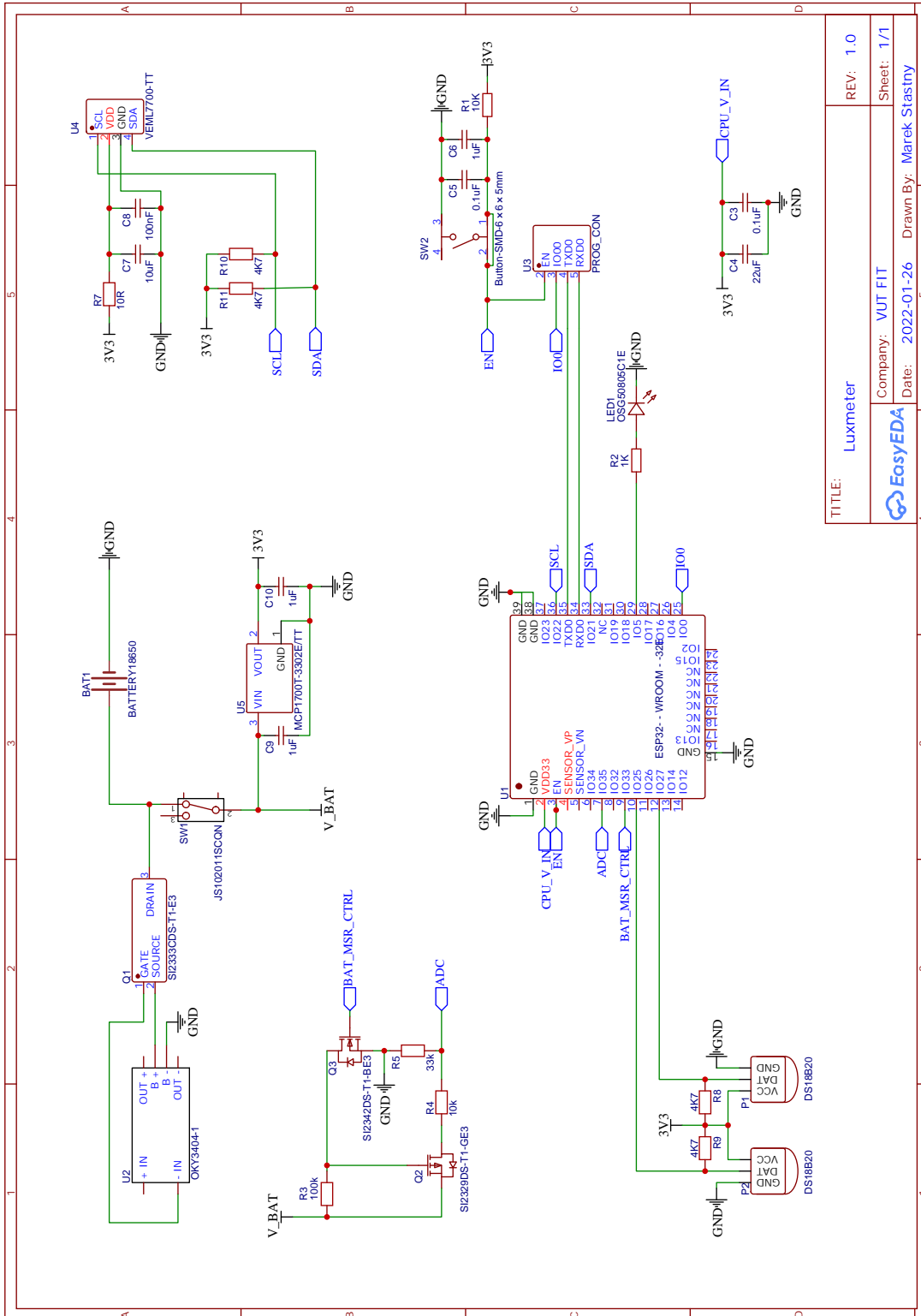
## Příloha A

# Schémata zapojení navržených obvodů



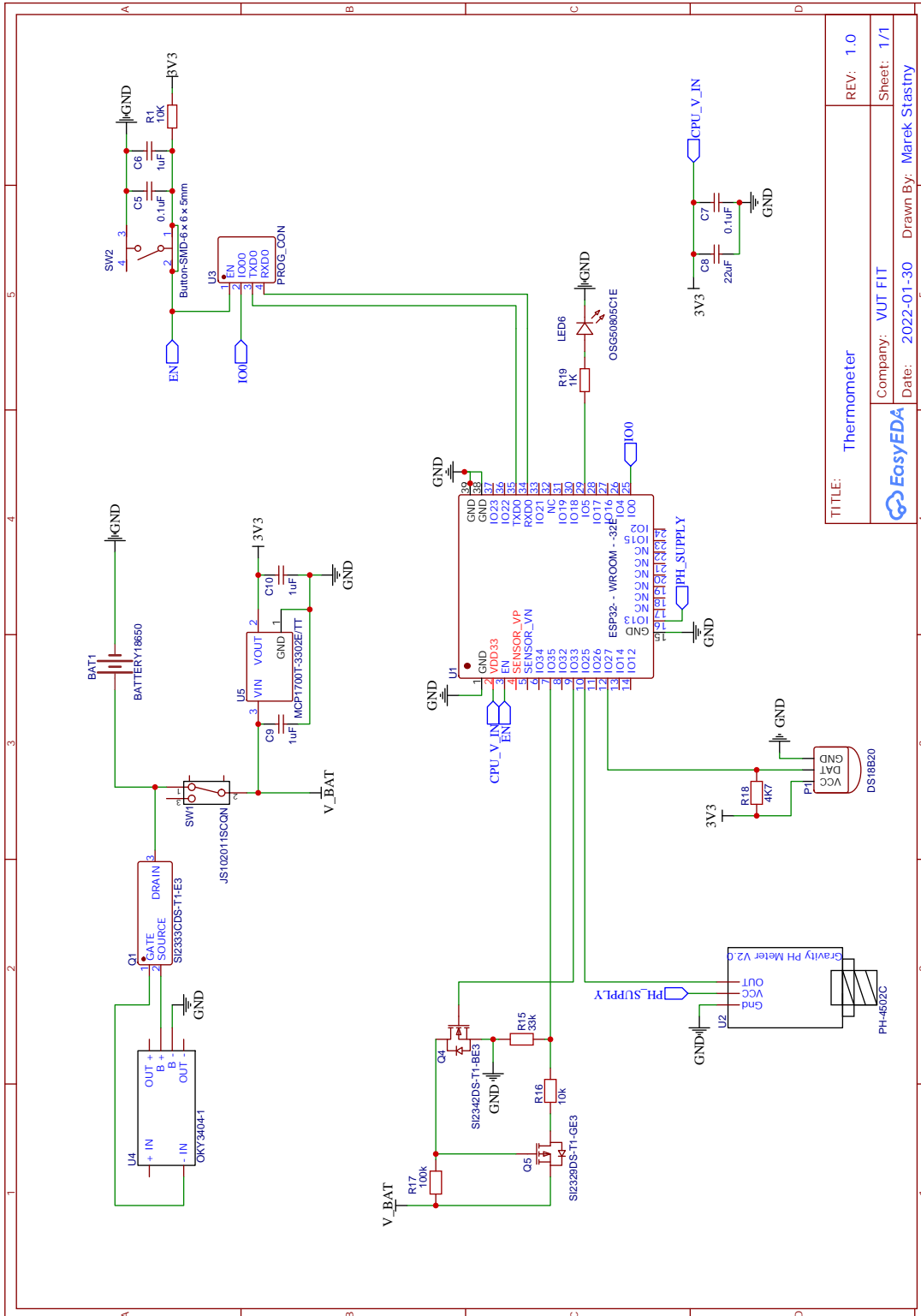
TITLE: Bridge	REV: 1.0
Company: VUT FIT	Sheet: 1/1
Date: 2022-01-26	Drawn By: Marek

Obrázek A.1: Schéma zapojení bridge.



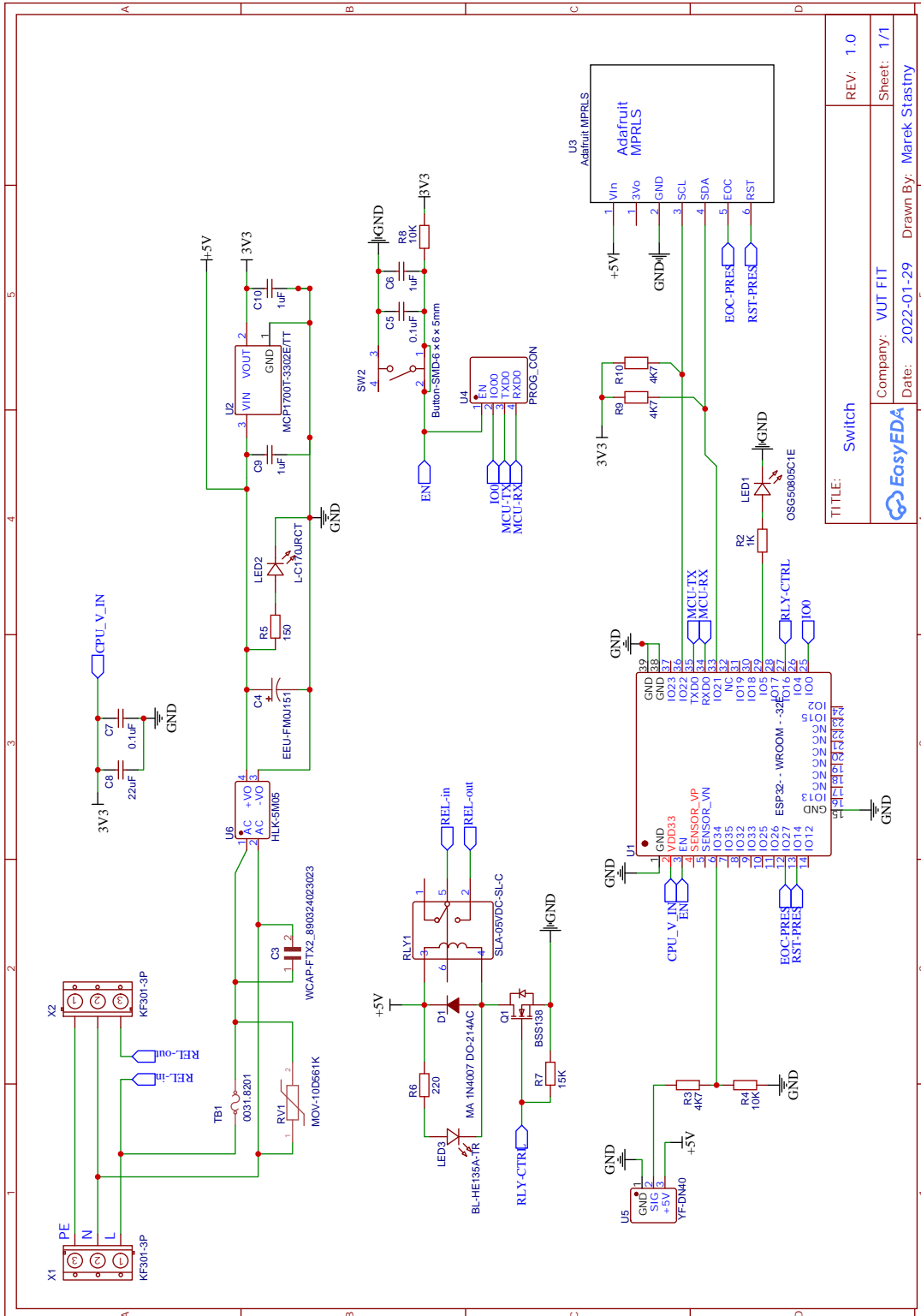
TITLE: Luxmeter	REV: 1.0
Company: VUT FIT	Sheet: 1/1
Date: 2022-01-26	Drawn By: Marek Stastrny

Obrázek A.2: Schéma zapojení senzoru osvětlení.



TITLE: Thermometer	REV: 1.0
Company: VUT FIT	Sheet: 1/1
Date: 2022-01-30	Drawn By: Marek Stastny

Obrázek A.3: Schéma zapojení plovací jednotky.



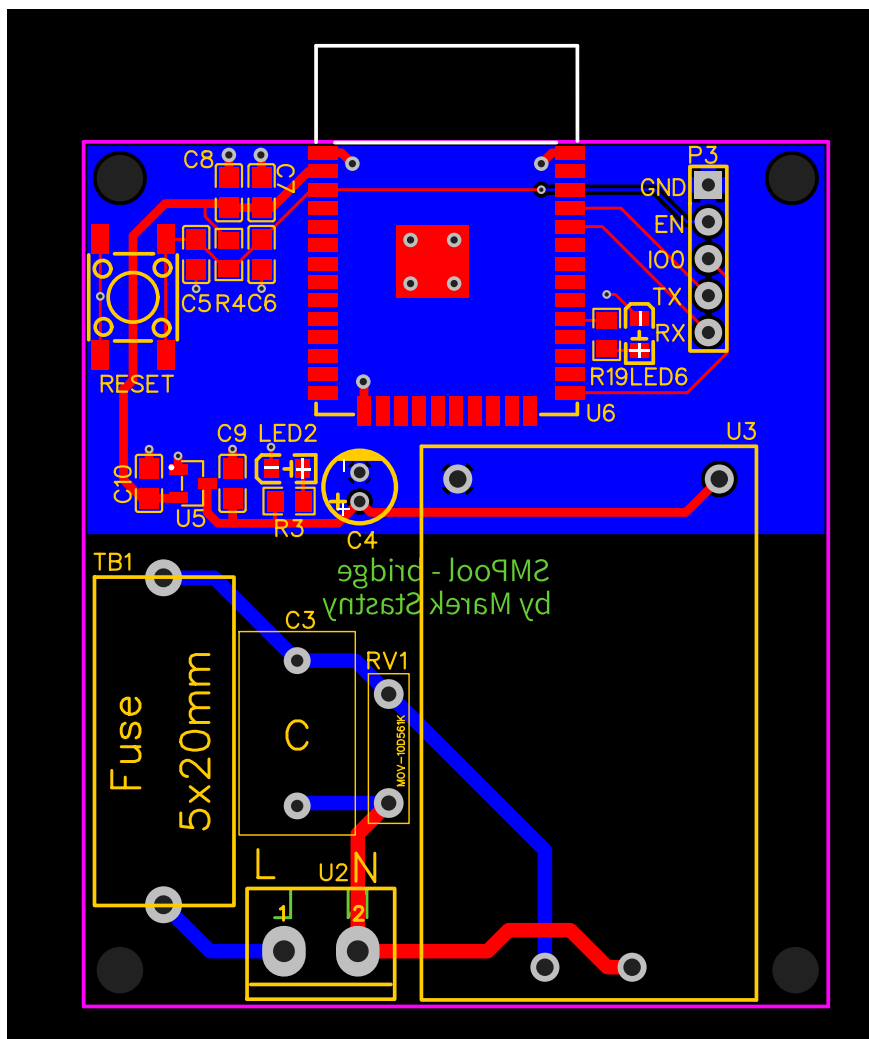
TITLE: Switch	REV: 1.0
Company: VUT FIT	Sheet: 1/1
Date: 2022-01-29	Drawn By: Marek Stastny

Obrázek A.4: Schéma zapojení spínací jednotky.

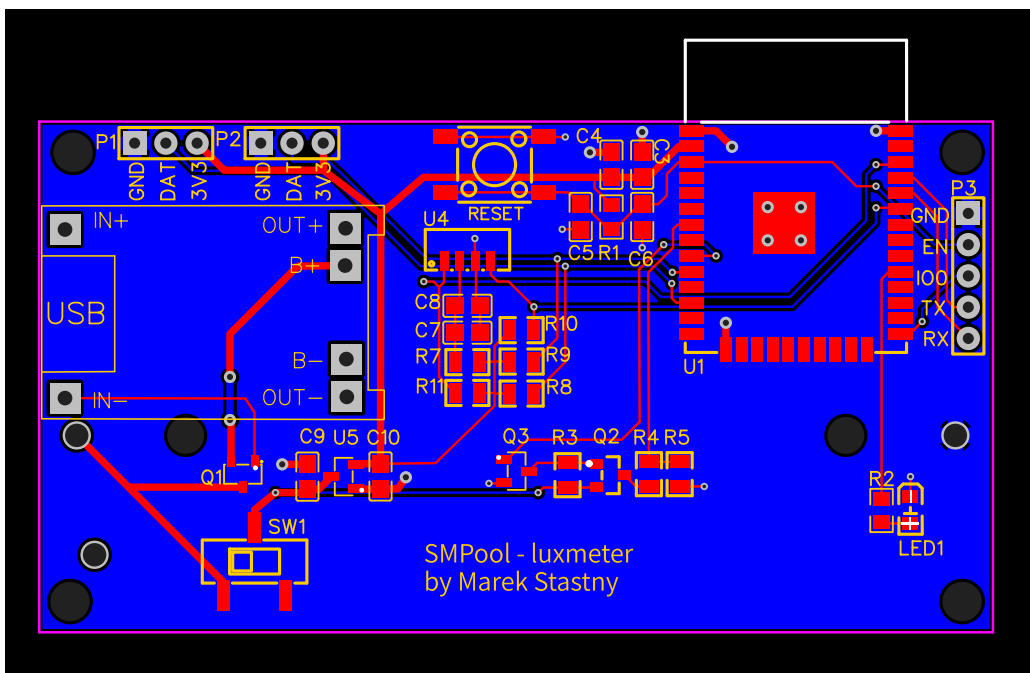
**Příloha B**

**Návrhy desek plošných spojů**

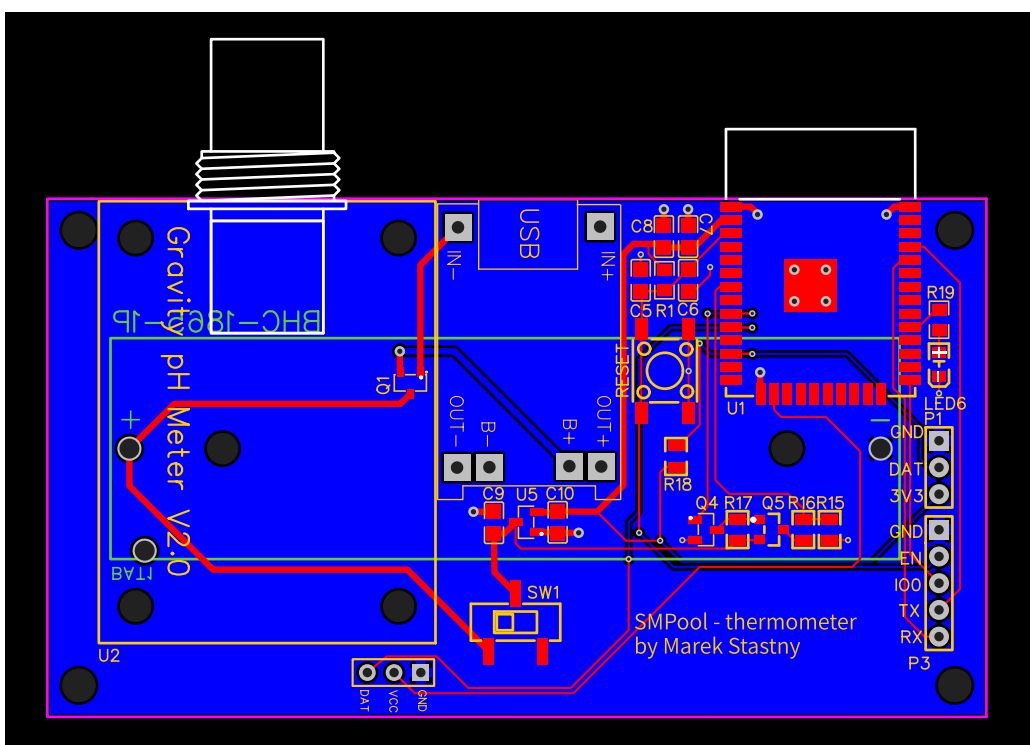




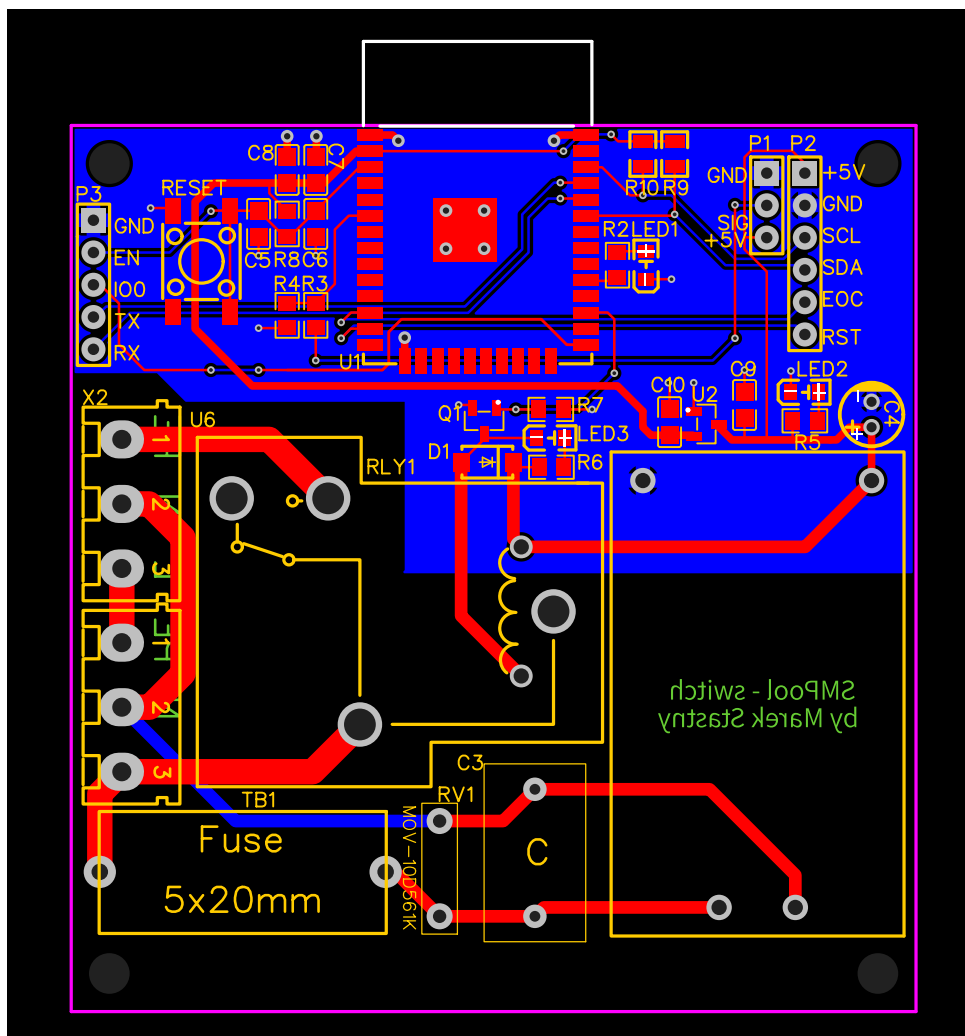
Obrázek B.1: Návrh desky plošných spojů pro bridge.



Obrázek B.2: Návrh desky plošných spojů pro senzor osvětlení.



Obrázek B.3: Návrh desky plošných spojů pro plovací jednotku.



Obrázek B.4: Návrh desky plošných spojů pro spínací jednotku.

## Příloha C

# Obsah přiložené SD karty

/	
	— README.txt.....návod k instalaci
	— docs
	— thesis ..... technická zpráva a její zdrojové soubory
	— diagrams ..... schémata zapojení HW modulů
	— pcb_layouts ..... schémata PCB
	— src
	— firmware ..... zdrojové kódy pro HW moduly
	— web_app ..... zdrojové kódy webové aplikace
	— diagrams ..... zdrojové kódy schémat zapojení HW modulů
	— pcb_layouts ..... zdrojové kódy schémat PCB