

Obsah

1	Systémový návrh	3
1.1	Požadavky	3
1.2	Blokové schéma	3
1.3	Komunikační rozhraní	5
1.3.1	Výběr datové sběrnice	5
1.3.2	Sběrnice CAN	7
2	Návrh řídicí jednotky	8
2.1	Mikrokontrolér	8
2.2	Návrh zapojení a tvorba DPS	9
2.2.1	Zapojení ESP32 modulu	9
2.2.2	Napájecí obvod	10
2.2.3	Deska plošných spojů	11
2.3	Konektivita	13
3	Obecný modul periferie	15
3.1	Mikrokontrolér	15
3.2	Návrh zapojení a tvorba DPS	15
4	Volba a návrh periférií	17
4.1	LED osvětlení	17
4.1.1	Návrh zapojení	17
4.1.2	Tvorba DPS	21
4.2	Senzor teploty	21
4.2.1	Metody měření teploty	21
4.2.2	Realizace sensoru	22
4.3	Senzor výšky hladiny	23
4.4	Senzor pH	24
4.5	Ovládání 230V periférií	24
5	Software	26
5.1	Architektura	26
5.2	Popis CAN komunikace	27
5.3	Firmware řídicí jednotky	28
5.3.1	FreeRTOS	29
5.3.2	Indikace stavu zařízení	29
5.3.3	30
5.4	Firmware periférií	30

5.5	Webové rozhraní	30
6	Sestavení a testování	32
	Seznam příloh	33
A	Schéma řídicí jednotky	34
A.1	Blokové schéma	34
A.2	Zapojení MCU	35
A.3	Napájecí obvod	36
A.4	Konektory	37
A.5	Sběrnice periférií	38
B	Schéma modulu periférií	39
C	Schéma modulu LED osvětlení	40

1 Systémový návrh

Tato část práce popisuje proces návrhu vlastního zařízení, které by mělo být výstupem této práce. Věnuje se konkretizaci požadavků na zařízení a koncepčnímu návrhu na systémové úrovni, který je zde podpořen blokovým schématem. Po celou dobu tvorby zařízení je kladen důraz na požadavky stanovené v této kapitole a na jejich základě jsou tvořena vhodná technická řešení. Detailně se jednotlivým blokům a jejich návrhu věnují následující kapitoly.

1.1 Požadavky

Cílem je vytvořit zařízení, které umožní co nejvíce automatizovat provoz akvária. Hlavním aspektem by měla být jednoduchost použití pro koncového uživatele, vše by mělo být nanejvýš intuitivní a přehledné. Zařízení musí mít možnost připojení k internetu prostřednictvím sítě Wi-Fi, uživatel tak bude moci zařízení konfigurovat a sledovat z libovolného místa za pomoci webové stránky popř. mobilní aplikace.

Požadavky jednotlivých akvaristů se mohou lišit a zároveň se v čase měnit. Vytvoření dokonalého a všestaránného zařízení, které vyhoví všem účelům použití není v časových ani finančních možnostech bakalářské práce, proto byl stanoven požadavek, aby bylo zařízení co nejvíce modulární a rozšiřitelné. Musí být zvolena taková architektura, aby bylo možné v budoucnu přidat další funkce a periferie bez nutnosti modifikovat stávající hardware.

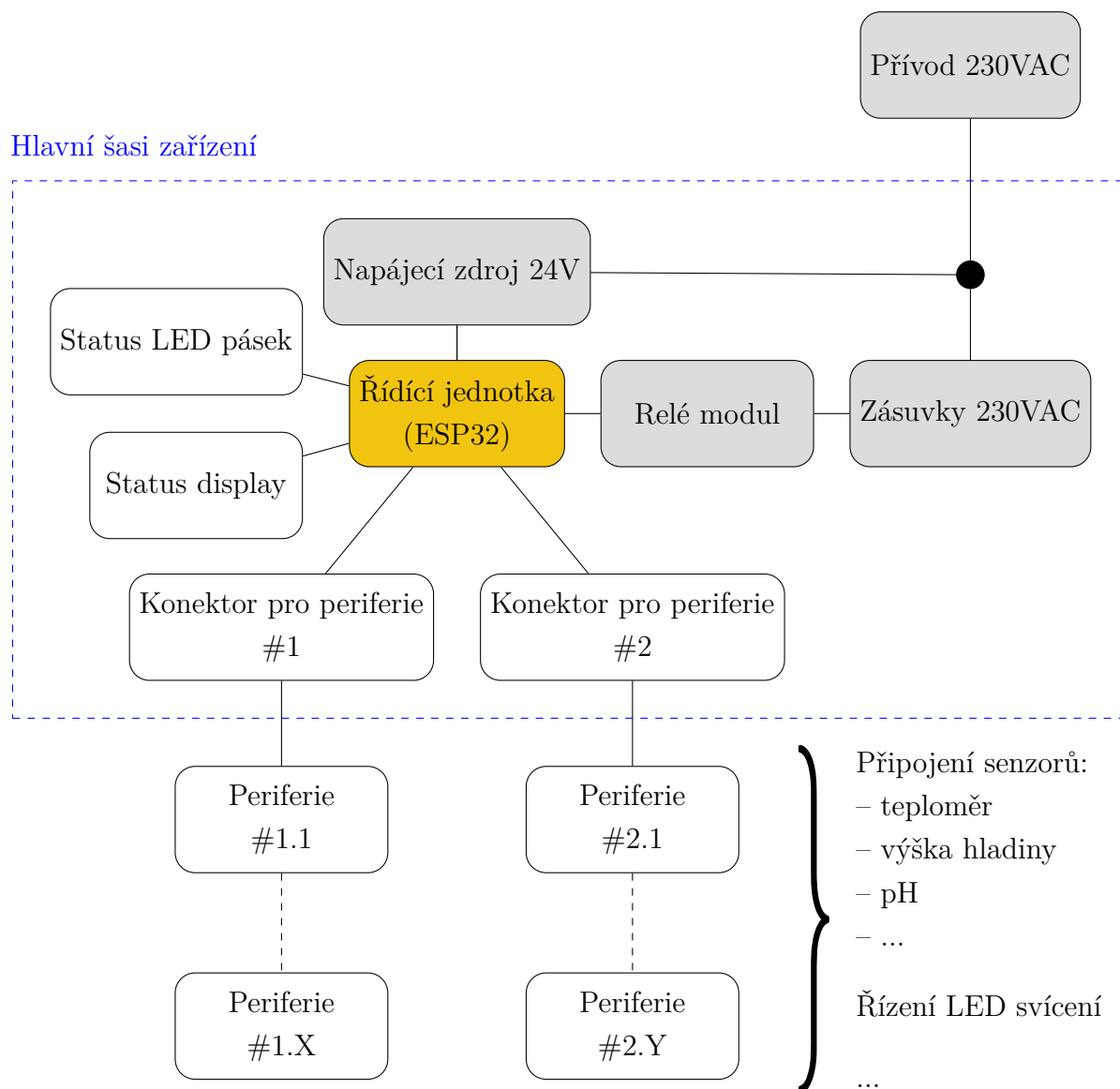
Výstupem bakalářské práce by mělo být zařízení schopné monitorovat některé akvaristické veličiny a na základě jejich hodnoty informovat uživatele a ovládat akvárium. Zařízení bude přímo řídit LED páskové osvětlení na 12 V a spínat popř. vypínat již existující akvaristické přístroje pracující se síťovým napětím 230 V.

Jelikož modulární architektura bude nepochybně vyžadovat použití více než jednoho mikrokontroleru a tedy také více různých firmwarů, je potřeba zajistit jejich vzájemnou kompatibilitu a stabilitu celého systému. Veškerý firmware tak musí být verzovaný a po připojení nové periferie musí řídicí jednotka rozpoznat, o jakou periferii se jedná. V případě připojení nekompatibilní periferie (např. z důvodu zastaralého firmwaru řídicí jednotky) musí být uživatel upozorněn a nesmí být nijak narušena funkce zbytku systému. Aby bylo možné těmto situacím předejít, musí mít řídicí jednotka možnost vzdálené aktualizace firmwaru.

1.2 Blokové schéma

Blokové schéma zařízení se nachází na obr. 1.1. Pro pohodlné použití je hlavní část zařízení soustředěna do jedné krabičky napájené přívodním síťovým kabelem. Uživa-

tel pak dle potřeby připojí příslušenství pracující s napětím 230 V do integrovaných síťových zásuvek a veškeré další periferie za pomoci jednoho z univerzálních konektorů. O stavu zařízení bude uživatel informován sérií notifikačních LED a malým displayem.



Obr. 1.1: Blokové schéma systému.

Pro napájení vlastní elektroniky zařízení je v šasi umístěn hotový modul spínaného zdroje převádějící síťové napětí 230 V na stejnosměrných 24 V se kterými pak zařízení dále pracuje (viz kapitola 2.2.2).

Z hlediska bezpečnosti je potřeba zajistit, aby se uživatel ani samotná nízkonapěťová část obvodu nemohli dostat do kontaktu s nebezpečným napětím. Toho je dosaženo galvanickým oddělením částí zařízení pracujících se síťovým napětím.

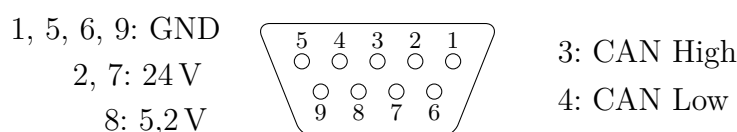
V blokovém schématu (obr. 1.1) jsou všechny tyto části podbarveny šedou barvou. Pro spínání síťových zásuvek jsou použita relé, které samy o sobě tvoří galvanickou izolaci, pro ještě lepší ochranu mikrokontroleru je pak zvolena varianta s optočleny. Pro napájecí zdroj s výstupem 24 V je přítomnost galvanického oddělení kontrolována v dokumentaci výrobce.

1.3 Komunikační rozhraní

Před návrhem jednotlivých částí zařízení je zapotřebí definovat komunikační rozhraní mezi řídicím modulem a periferiemi, protože právě od jeho specifikace se následně odvíjí tvorba zbytku zařízení.

Úkolem rozhraní je obousměrně komunikovat s periferiemi, tedy např. stahovat data z připojených sensorů a zároveň za pomoci příkazů periferie řídit. Kromě datové komunikace musí rozhraní periferie také napájet a to i v případě energeticky náročnějších obvodů jako např. osvětlení.

Pro připojení periférií byl zvolen konektor typu D'sub 9, který je cenově dostupný, disponuje dostatečným množstvím pinů a umožňuje montáž jak do panelu, tak i jako zakončení kabelu. Přiřazení a funkce jednotlivých vodičů jsou vyobrazeny na obr. 1.2. Konektor disponuje dvěma úrovněmi napájení, 5 V slouží k napájení mikrokontrolerů modulů periférií a k nim připojených sensorů nebo jiné nenáročné elektroniky. Je pravděpodobné, že s připojením více periférií za sebe dojde k úbytku napětí v důsledku ztrát na vedení, aby bylo toto částečně kompenzováno, napětí vystupující z měniče řídicí jednotky je přibližně o 0,2 V vyšší. Druhou napájecí linkou je výstup přímo z externího spínaného zdroje, tedy s napětím 24 V, ten slouží pro výkonově náročnější periferie, které si již napětí dále upraví podle potřeby a nebudou neúměrně zatěžovat první zmíněnou napájecí linku.



Obr. 1.2: Přiřazení pinů konektorů D'sub 9 pro připojení periférií.

1.3.1 Výběr datové sběrnice

Existuje celá řada datových sběrnic, které jsou v elektrotechnice hojně využívány. Každá z nich má své výhody a nevýhody stejně jako jisté limitace použití. V tab. 1.1 se nachází výčet různých sběrnic, které byly při výběru uvažovány.

Tab. 1.1: Datové sběrnice, porovnání [1].

Typ	Výhody	Nevýhody	Limitace
SPI	Více zařízení na sběrnici Vysoká rychlost přenosu dat Jednoduchý protokol	Nutný CS pin pro každé zařízení	Určeno na krátkou vzdálenost
I ² C	Pouze 2 piny Více zařízení – 128 adres	Riziko kolize adres Nižší rychlost přenosu dat proti SPI	Určeno na krátkou vzdálenost
CAN	Vysoká spolehlivost Dlouhé propojení	Vyšší náklady na implementaci Nižší rychlost přenosu dat	Nepodporovano běžnými MCU – nutný externí řadič
UART	Jednoduchá implementace Možnost asynchronní komunikace	Nižší rychlost přenosu dat proti SPI Pouze 2 zařízení	Pouze 2 zařízení Určeno na krátkou vzdálenost

Protože hlavní šasi zařízení nabízí dva konektory, ale žádoucí je připojit větší předem nedefinovaný počet periférií, je potřeba, aby sběrnice umožnila připojení více zařízení. Obecný problém všech sběrnic je omezení jejich maximální délky, s rostoucí délkou se sběrnice snáze zaruší, navíc z důvodu parazitních vlastností vedení dochází k zaoblení ostrých hran signálu, dlouhé vedení se chová jako filtr typu dolní propust. V důsledku toho se snižuje maximální rychlost sběrnice.

Sběrnice SPI nebo **i2c!** je obecně doporučeno používat pouze v rámci DPS, tedy na krátké vzdálenosti. Při snížení rychlosti je možné je používat i na větší vzdálenost, ovšem modulární scénář vytvářeného systému teoreticky nestanovuje žádný délkový limit a bylo by velmi obtížné spolehlivě určit, kolik periférií uživatel může za sebe zapojit při zachování spolehlivé komunikace.

UART je výhodný svou jednoduchou implementací a umožňuje obousměrnou asynchronní komunikaci. Nevýhodou je že funguje pouze pro dvě zařízení. Jednou z možností jak tuto limitaci obejít by bylo zavedení řetězového způsobu komunikace, kdy by každé zařízení komunikovalo se dvěma sousedními a informace by se postupně předávala dále až k cílovému zařízení. Tento systém je relativně jednoduchý, ale například v případě poruchy jednoho zařízení se odpojí všechna následující zařízení, což může mít neočekávané následky.

Sběrnice CAN je určena pro provoz v průmyslovém prostředí (zejména je po-

užívána v automobilovém průmyslu) a díky své robustnější konstrukci ji lze bez problému použít i na delší vzdálenosti a pro více zařízení. Při komunikaci je používán diferenční pár vodičů, takže i odolnost proti rušení je výrazně lepší. Nevýhodou je ale její o něco složitější a dražší implementace. Většina běžných mikrokontrolerů nemá pro CAN vestavěnou periférii a je tak potřeba buďto zvolit dražší mikrokontroler nebo připojit externí ovladač řízený např. přes SPI, dále je nutné přidat i řadič, který převede signál na diferenční a zároveň umožní zvýšit provozní napětí na 12 nebo 24 V, čímž dojde k ještě lepšímu potlačení šumu.

1.3.2 Sběrnice CAN

Po důkladné rešerši a zvážení zmíněných kladů a záporů byla zvolena sběrnice CAN. ESP32 jakožto již zvolený mikrokontroler řídící jednotky obsahuje vestavený CAN kontroler a pro moduly periférií byl na základě tohoto rozhodnutí zvolen také vhodný mikrokontroler. Co se týče nutnosti přidání řadiče, jedná se sice o další součástku, která na první pohled navyšuje cenu zařízení, kromě převodu signálu na diferenční ale zajišťuje také ochranu konektorů proti mnoha nežádoucím jevům jako je zkrat, ESD výboj nebo přepětí. Tímto se ve výsledku celé zapojení zlevní a zjednoduší.

2 Návrh řídicí jednotky

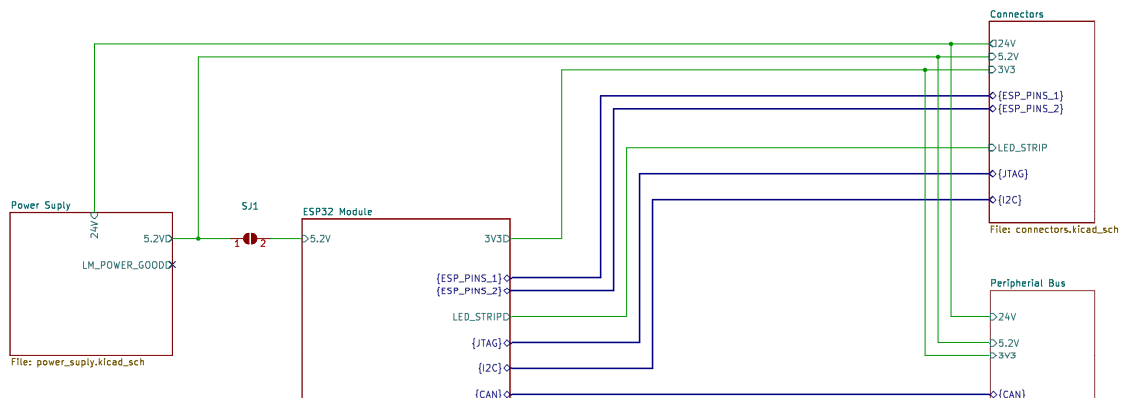
Řídicí jednotka je jádrem celého zařízení. Její funkcí je řízení systému a zároveň komunikace s uživatelem za pomoci Wi-Fi. Musí v sobě nést informaci o konfiguraci systému a na jejím základě zpracovávat data z jednotlivých připojených periferií. Podle uživatelem nastavených scénářů pak dynamicky reaguje na změny hodnot měřených akvaristických veličin a ovládá akční členy (osvětlení, ohřev, filtr vody). Za pomoci displaye a LED pásku také informuje uživatele o momentálním stavu zařízení.

2.1 Mikrokontrolér

Při výběru vhodného mikrokontroleru bylo potřeba zohlednit výše zmíněné požadavky, tedy zejména Wi-Fi konektivitu a dostatečný výkon k její obsluze, periferii CAN a dostatek GPIO pinů pro připojení zbylých modulů v hlavním šasi (viz obr. 1.1). Na trhu existuje vícero výrobců nabízejících mikrokontrolery s vhodnými parametry, z důvodu jednoduchosti použití a nízké ceny byl nakonec zvolen model ESP32 od firmy Espressif, konkrétně modul WROOM-32E [2] s čipem ESP32-D0WDR2-V3 [3]. Tento modul je často využíván v různých hobby projektech, ale také v komerčních aplikacích zejména v oblasti chytré domácnosti. Z tohoto důvodu k němu existuje velká škála softwarových knihoven a v rámci komunity uživatelů je také sdíleno mnoho projektů, kterými je možné se inspirovat.

2.2 Návrh zapojení a tvorba DPS

Řídící jednotka je tvořena jednou speciálně navrženou DPS, která kromě samotného mikrokontroleru obsahuje také měnič napětí typu BUCK ke snížení napájecího napětí externího zdroje na hodnotu 5,2 V. Toto napětí je pak dále používáno pro napájení samotného mikrokontroleru řídicí jednotky a zároveň je vyvedeno na konektor pro připojení periférií. Blokové schéma na úrovni logických bloků v rámci jedné DPS je na obr. 2.1, jednotlivým částem se blíže věnují další kapitoly. Celé schéma je k dispozici v příloze A.



Obr. 2.1: Blokové schéma řídicí jednotky. Vytvořeno v KiCad 7.0.

2.2.1 Zapojení ESP32 modulu

Při tvorbě schématu bylo vycházeno z dokumentace výrobce [2] a také ze schématů různých existujících vývojových desek. K zajištění správné a spolehlivé funkce modulu je potřeba dodržet několik věcí. Výřez schématu obsahující potřebné doplňující obvody pro ESP32 modul je na obr. 2.2.

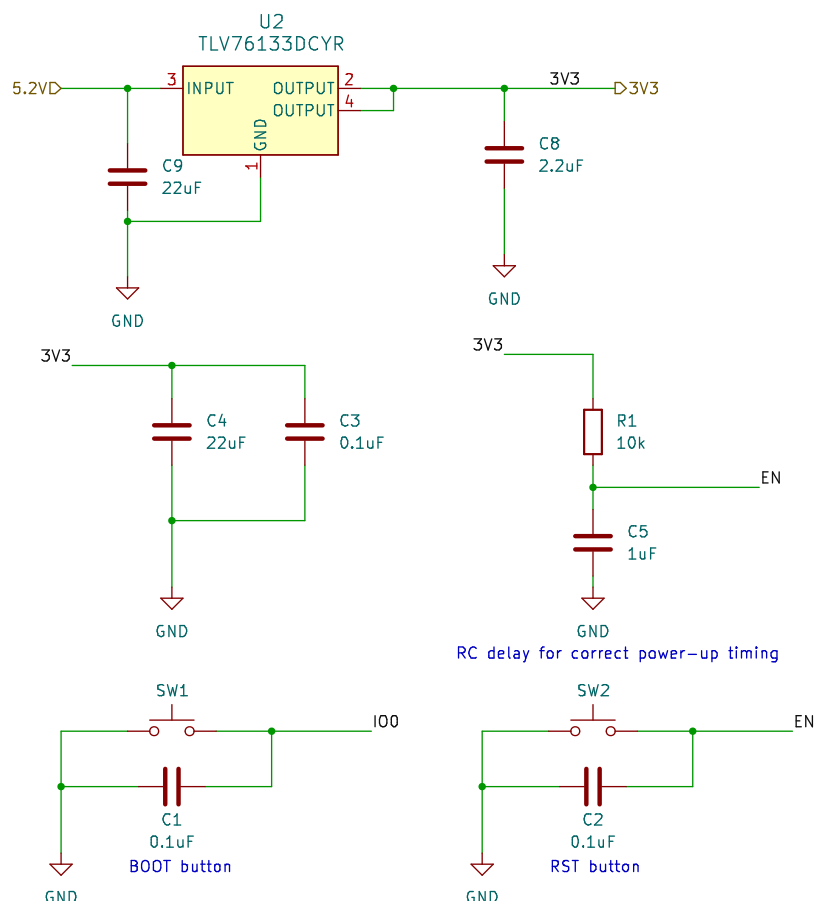
Na napájecí pin (3V3) je třeba přivést stabilní napětí a opatřit ho blokovacími kondenzátory (C1, C3). Ke snížení napětí z původních 5,2 V na požadovaných 3,3 V je použit lineární regulátor TLV76133 (U4).

Dále je potřeba přivést kladné napětí na povolovací pin (EN). Z dokumentace vyplývá, že by mělo být přivedeno až po ustálení napájecí linky. Uvedený čas nutný ke stabilizaci je roven $t_{STBL} = 50 \mu s$ [3]. Požadované zpoždění zajistí RC článek (R1, C2) s časovou konstantou τ :

$$\tau = R_1 C_2 = 10 \text{ k}\Omega \cdot 1 \mu\text{F} = 10 \text{ ms} \quad (2.1)$$

Jak je vidět, byla zvolena dostatečná návrhová rezerva.

Pro možnost resetu zařízení a vstupu do bootloaderu byla doplněna také dvě tlačítka (SW1, SW2).

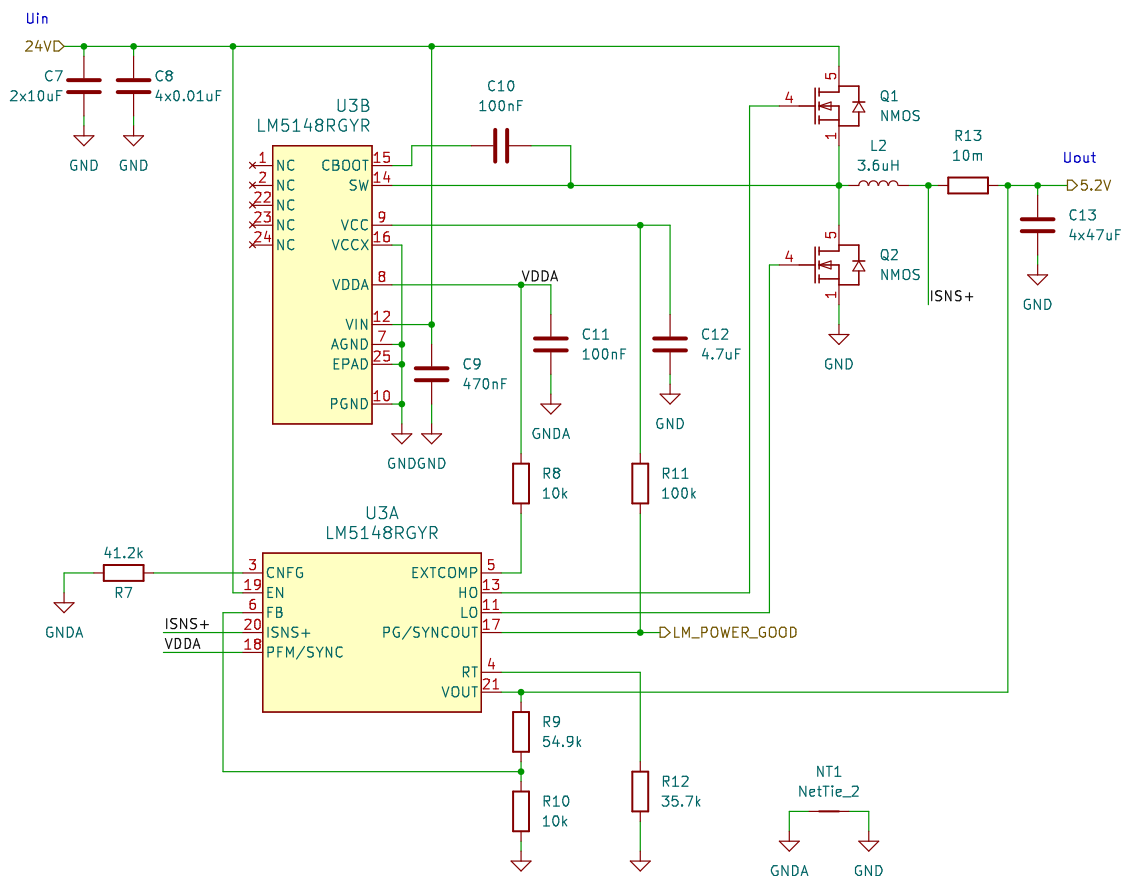


Obr. 2.2: Podpůrné obvody pro modul ESP32-WROOM-E. Vytvořeno v KiCad 7.0.

2.2.2 Napájecí obvod

Pro napájení celého zařízení je použit externí zdroj stejnosměrného napětí 24 V a toto napětí je takto rozvedeno všem připojeným periferiím. Pro většinu komponent je ale nutné napětí snížit. K tomuto účelu byl navržen DC/DC měnič typu buck s požadovaným výstupním napětím 5,2 V. Existuje celá řada čipů vyvinutých pro tento účel. Aplikace v tomto zařízení je specifická svými požadavky na výstupní proud, zatímco samotná řídicí jednotka nebude odebírat velký proud, není jasné, kolik periférií a s jakými výkonovými požadavky uživatel k systému připojí. Navržený měnič tak musí fungovat v širším rozsahu proudů (řádově od desítek mA po jednotky A) a to s co nejlepší účinností.

Aby bylo vyhověno zmíněným požadavkům a zachována návrhová rezerva, byl jako základ buck měniče zvolen čip LM5148 [4]. Jedná se o moderní součástku firmy Texas Instruments s velkou výkonovou rezervou. Tento čip funguje pouze jako buck kontroler a zapojení je potřeba doplnit zapojení dvěma externími MOSFET tranzistory, většina tepelných ztrát vzniká právě na nich, čímž se sníží ohřev samotného



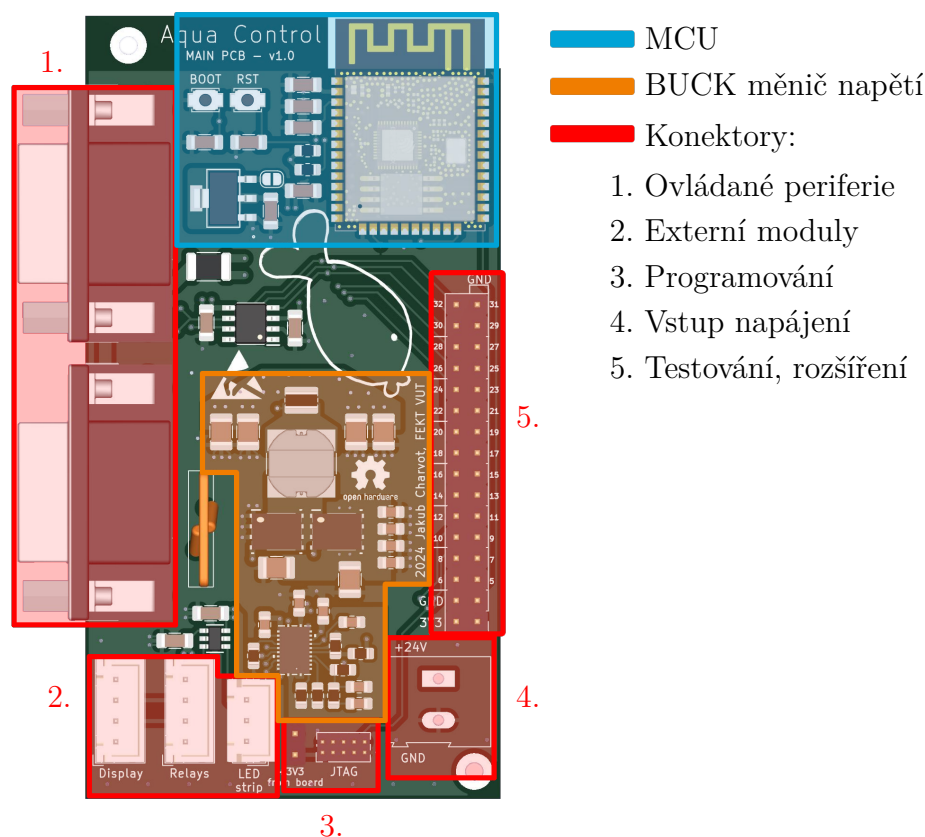
Obr. 2.3: Napájecí obvod řídicí jednotky. Vytvořeno v KiCad 7.0.

čipu a generované teplo se lépe rozloží. Na volbě tranzistorů závisí také výsledná účinnost měniče. Při návrhu zapojení této součástky byl použit nástroj Webench Power Designer [5], který podle zadaných parametrů navrhne konkrétní schéma zapojení, provede simulaci a zobrazí grafy upravené na míru zadaným hodnotám. Tento nástroj uvádí přibližnou účinnost zapojení jako 88 %. V navrženém schématu bylo posléze provedeno několik změn, aby vše odpovídalo požadavkům uvedeným v katalogovém listu součástky [4]. Kompletní schéma zapojení spolu s odkazy k relevantním kapitolám katalogového listu se nachází v příloze A.3, pro přibližnou představu pak postačí zjednodušené schéma na obr. 2.3.

TODO: výpočty by asi bylo dobré uvést co?

2.2.3 Deska plošných spojů

Ačkoliv se jedná o relativně jednoduchou DPS, je potřeba při návrhu dbát jistých pravidel a doporučení. Modul ESP32 je vybaven anténou a volba jeho umístění na DPS je rozhodujícím faktorem pro spřávnou funkci antény. Další částí vyžadující správný návrh rozložení a propojení součástek je pak BUCK měnič.



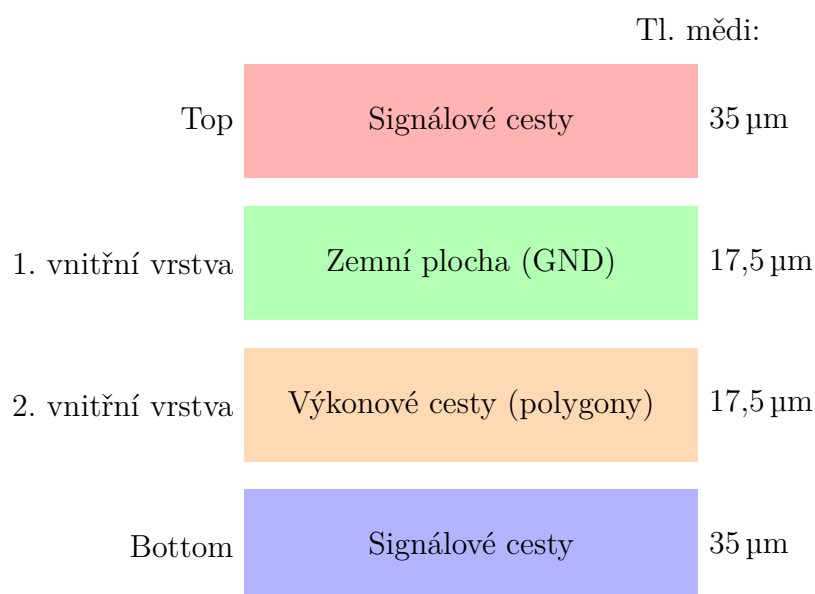
Obr. 2.4: Vizualizace DPS řídicí jednotky s vyznačením jednotlivých částí.

Prvním krokem návrhu je volba počtu vrstev a jejich funkce. Vyšší počet vrstev nabízí více prostoru pro vedení cest a také umožňuje vedení napájecích napětí pomocí rozlitéch měděných polygonů, čímž se zároveň zlepší vlastnosti zařízení z hlediska EMC. Zvolený výrobce (JLC PCB [6]) nabízí výrobu desek s jednou až dvaceti vrstvami mědi. Byla zvolena čtyřvrstvá deska, která je pro danou aplikaci dostatečná a stále se nachází v přijatelné cenové skupině výrobce. Rozložení a funkce vrstev jsou vyobrazeny na obr. 2.5.

Pro optimální funkci Wi-Fi antény výrobce doporučuje umístit ESP32 modul do pravého horního rohu DPS tak, aby se pod anténou nenacházela vrstva mědi a nejlépe ani samotná deska [7]. Na obr. 2.4 je zobrazen výsledný návrh DPS, v modře vyznačené oblasti lze vidět, že tyto požadavky byly splněny. Vedle ESP32 modulu se nachází související součástky popsané v kapitole 2.2.1.

Měnič napětí

TODO: Zde popis návrhu, zdroje, obrázky Q: Jak moc do detailu?



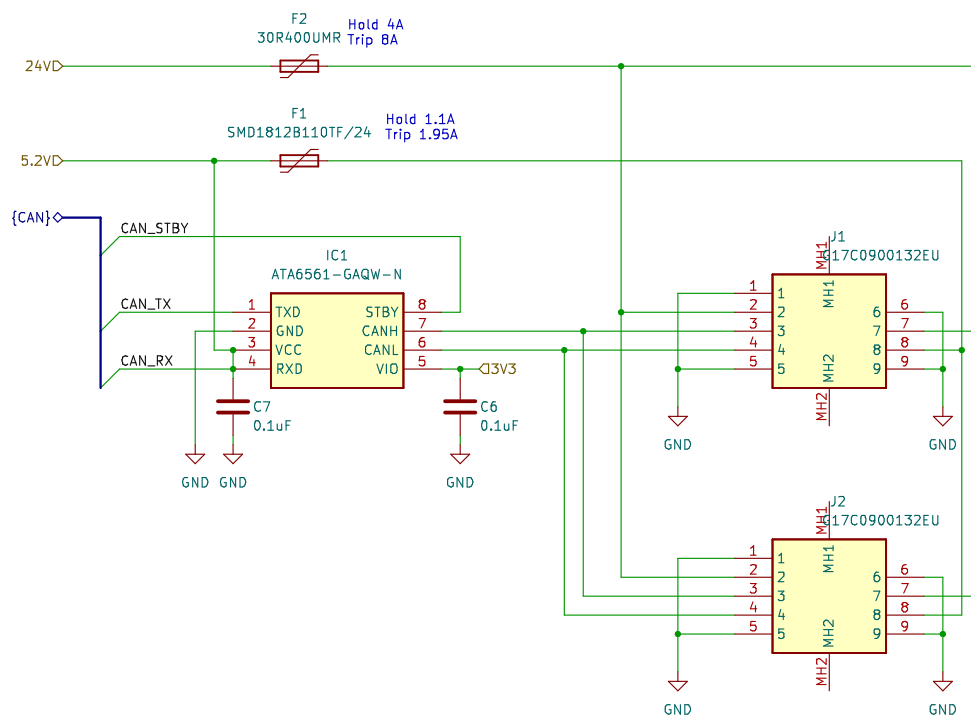
Obr. 2.5: Rozložení vrstev DPS řídicí jednotky.

2.3 Konektivita

Řídicí jednotka je vybavena několika konektory, jak je opět možno vidět na obr. 2.4. Na levé straně DPS (značeno 1.) se nachází konektory pro připojení periférií připravené pro montáž do panelu, kde pak budou přístupné uživateli. Ostatní vyznačené konektory slouží pro připojení externích modulů v rámci hlavního šasi (viz blokové schéma na obr. 1.1) popř. pro programování a testování, nebudou tedy volně přístupné uživateli.

Abychom předešli poškození zařízení při nevhodném zacházení uživatelem, je potřeba pro volně dostupné konektory přidat dodatečnou ochranu [8]. Jak je vidět na obr. 1.2, konektor pro periferie sdružuje jak datovou komunikaci, tak i napájení. Ochranu diferenční datové linky zajistí samotný CAN řadič, který je určen pro průmyslové použití a obsahuje zabudovanou ochranu jak proti zkratu datové linky s napájením či zemí, tak proti ESD [9].

Co se týče napájecích vodičů, každý z nich je ošetřen vratnou pojistkou (ang. polyfuse) dimenzovanou podle předpokládaného maximálního odběru zařízení. Při překročení tohoto proudu, např. z důvodu zkratu v některé z periférií, pojistka sepne a proud v obvodu omezí na minimum. Schéma zapojení konektorů spolu s hodnotami vratných pojistek se nachází na obr. 2.6.



Obr. 2.6: Zapojení a ochrana konektorů. Vytvořeno v KiCad 7.0.

3 Obecný modul periferie

Díky zvolené koncepci systému je možné za periférii považovat jakékoliv zařízení schopné obousměrně komunikovat po navržené sběrnici. Není vyloučeno, aby byla každá periferie navržena zcela odlišně na základě svých vlastních požadavků na výkon, počet pinů nebo dostupná rozhraní daného MCU. Hlavní výhodou této koncepce je to, že periferie mohou být vyvíjeny postupně a přidávány do již funkčního a odladěného systému bez nutnosti modifikovat stávající hardware. V případě chyby v návrhu periferie je také oprava méně náročná, než by tomu bylo v případě zabudování veškeré funkcionality přímo do řídicí jednotky.

Nicméně pokud by byl pro každou periférii zvolen zcela jiný mikrokontroler a vytvořen vlastní návrh DPS, vývoj více periférií by byl zbytečně drahý a časově náročný. Proto byl zvolen koncept „obecného modulu periferie“, tedy jedné DPS s konkrétním mikrokontrolerem zajišťující připojení ke komunikačnímu rozhraní, napájení periferie a rozhraní pro programování. Kromě toho budou na DPS dvě dutinkové lišty, do kterých bude možné vsadit druhou DPS (popř. během vývoje pouze prototypovací desku) ve funkci dceřinné desky (ang. daughterboard). Vložená deska pak bude obsahovat obvody nutné přímo pro danou konkrétní periférii.

3.1 Mikrokontrolér

Kritéria pro výběr mikrokontroleru byla následující:

- Musí nutně splňovat:
 - CAN periferie – pro komunikaci po sběrnici
 - PWM výstup – řízení LED, popř. jiné
 - ADC – pro práci s analogovými sensory
 - Nízká cena
- Je výhodou:
 - Dobrá dokumentace, komunita uživatelů
 - Zkušenost autora s danou platformou
 - Další periferie (**i2c!**, SPI, ...)

Na základě těchto kritérií byl vybrán mikrokontroler **PIC18F26Q83** od firmy Microchip, ten splňuje všechna kritéria a disponuje také množstvím dalších periférií, které by mohly být v budoucnu užitečné [10].

3.2 Návrh zapojení a tvorba DPS

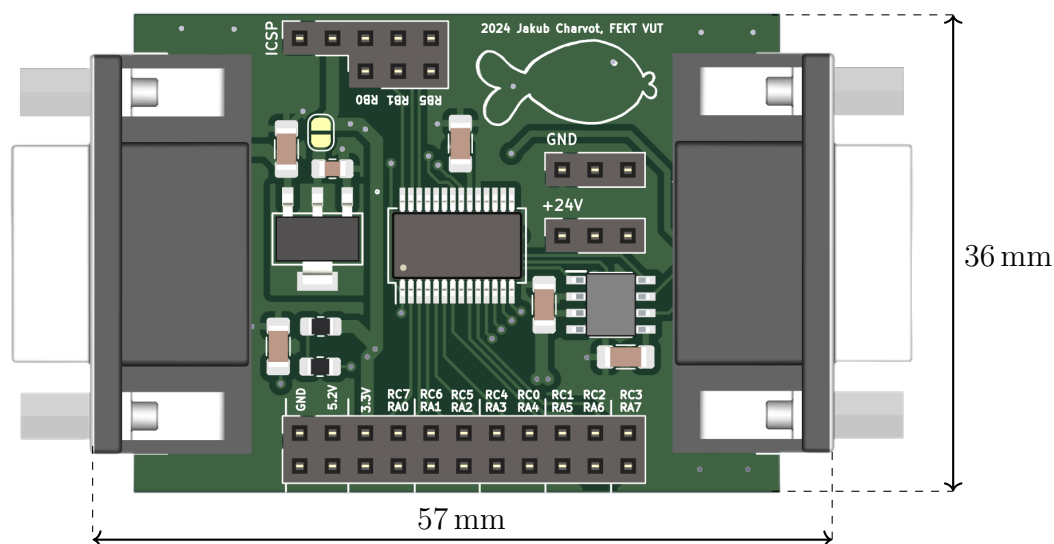
Celé schéma pro obecný modul periferie se nachází v příloze B. Návrh zapojení lze rozdělit na tři části. V prvním kroku je zapojen mikrokontroler tak, aby byly

opět splněny všechny požadavky výrobce. Mikrokontrolery řady PIC se vyznačují značnou jednoduchostí použití a ke správnému chodu jim stačí jen minimum dalších součástek. V případě PIC18F26Q83 postačí připojit blokovací kondenzátor (C6) na napájení (pin VCC) a přivést kladné napětí na resetovací pin (MCLR). MCU je dále doplněn resetovacím tlačítkem pro pohodlnější práci při vývoji a testování firmwaru a regulátorem napětí (U1) s výstupem 3,3 V.

Ve druhém kroku je přidána dvojice D'sub konektorů a CAN řadič ATA6561 obdobně jako u řídicí jednotky popsané v kapitole 2.2. Na závěr jsou všechny dosud nevyužité piny vyvedeny na konektor (dutinkovou lištu), aby byly jednoduše dostupné pro připojenou dceřinnou desku. Pro napájení výkonově náročnějších periferií (např. ovladač LED osvětlení) jsou zvlášť vyvedeny ještě dvě trojice pinů připojené na 24 V a zem (GND).

Tvorba DPS

Hlavním cílem bylo vytvořit DPS co nejmenší. Jedná se o samostatné moduly, kterých bude v systému zapojeno několik a pro manipulaci a rozmístění modulů okolo akvária je menší rozměr výhodou. Hlavní limitací v této snaze není počet součástek, ale spíše rozměry mechanických prvků, zejména konektorů D'sub. Na obr. 3.1 je vidět výsledný návrh DPS s osazenými součástkami a dosaženými rozměry. Je zřejmé že jiným rozložením dutinkových lišt by se návrh dal ještě drobně zmenšit, ale je potřeba ponechat jistou flexibilitu pro tvorbu dceřinných desek.



Obr. 3.1: Vizualizace DPS obecného modulu periferie.

Co se týče rozložení vrstev, byla opět použita čtyřvrstvá deska se strejnou strukturou jako je na obr. 2.5 pro DPS řídicí jednotky. Celý návrh DPS se nachází v příloze TODO.

4 Volba a návrh periférií

Tato kapitola se již věnuje návrhu konkrétních periférií, tedy jednotlivých senzorů a akčních členů. Po technické stránce jsou všechny zmíněné moduly nadstavbou pro „obecný modul periferie“ popsaný detailně v předešlé kapitole. Jelikož je celý systém modulární, je pravděpodobné, že postupem času bude dále rozšiřován o nové typy periférií a i v současné chvíli je jich v plánu více, než je v možnostech této práce. Pro lepší přehlednost se v tab. 4.1 nachází přehled všech realizovaných i v tuto chvíli pouze plánovaných periférií.

Tab. 4.1: Přehled periférií.

Název	Typ	Napájení	Funkce	Realizováno
Sensor teploty	S	5 V	Teplota vody	Ano, kap. 4.2
Sensor hladiny	S	5 V	Výška hladiny (spojitě + skokově)	Ano, kap. 4.3
LED osvětlení	A	24 V	Intenzita osvětlení	Ano, kap. 4.1
Sensor pH	S	5 V	pH vody	Ano, kap. 4.4
Topné těleso	A	230 V	Ohřev vody	Ano, kap. 4.5
Filtr vody	S	230 V	Filtrace vody	Ano, kap. 4.5
Krmítko	A	24 V	Dávkování krmiva	Ne
Sensor průtoku	S	-	Voda tekoucí filtrem	Ne

S = sensor, A = akční člen

4.1 LED osvětlení

Úkolem této periferie je zajistit osvětlení akvária a umožnit jeho ovládání. V porovnání s ostatními moduly je zapojení relativně komplexní a proto byla pro tuto periferii navržena a zhotovena vlastní DPS (fungující jako dceřinná deska, viz. 3). Jako typ svítidla byly zvoleny LED pásy pracující s napětím 12 V. Modul musí být schopen samostatně ovládat 2 LED pásy, kdy za pomoci regulace proudu do LED pásu nastaví intenzitu osvětlení.

4.1.1 Návrh zapojení

Na začátku návrhu je potřeba specifikovat si požadavky na elektrické parametry zapojení. Uvažujme délku každého pásu $l = 1$ m, vstupní napětí získané z konektoru D'sub $U_{in} = 24$ V a výstupní napětí pro které je pásek určen $U_{out} = 12$ V. Pro stanovení maximálního proudu bylo vycházeno z údajů na e-shopu LED Solution [11],

kdy nejvýkonější nabízený LED pásek pro dané napětí má příkon $P_i = 20 \text{ W/m}$. Pak každý kanál musí být schopen dodat proud odpovídající nejnáročnějšímu scénáři:

$$I_{max} = \frac{P_i \cdot l}{U_{out}} = \frac{20 \cdot 1}{12} = 1,66 \text{ A} \quad (4.1)$$

Jelikož se jedná o dceřinnou desku pro obecný modul, rozměr výsledné DPS je omezen a část plochy je navíc využita konektory pro vsazení do obecného modulu. Je tedy potřeba zvolit co nejvíce integrované řešení, které současně slibuje dobrou účinnost a tedy co nejnižší ohřev zařízení během provozu.

Pro řízení LED osvětlení je často používán proudový zdroj, který umožňuje lineárně regulovat výstupní proud a tím i intenzitu osvětlení. Na trhu existuje opět celá řada čipů určena přímo k ovládání LED pásků [12], problém je zde ale v tom, že uživatel velmi pravděpodobně připojí pásek s nižším, předem neznámým příkonem. Maximální proud je tedy specifický danému LED pásku a proudový zdroj by musel zároveň spolehlivě zaručit, že nebude překročeno napětí $U_{out} = 12 \text{ V}$.

Touto funkcí většina čipů nedisponuje a pokud ano, nejsou dostatečně integrované pro použití v této aplikaci. Po důkladné rešerši a několika iteracích návrhu se nakonec jeví jako nejlepší použití napěťového měniče typu BUCK spolu se zesilovačem pro snímání proudu. Snímaný proud je následně zpracován mikrokontrolerem a jsou upraveny poměry ve zpětné vazbě měniče, aby napětí odpovídalo požadovanému proudu.

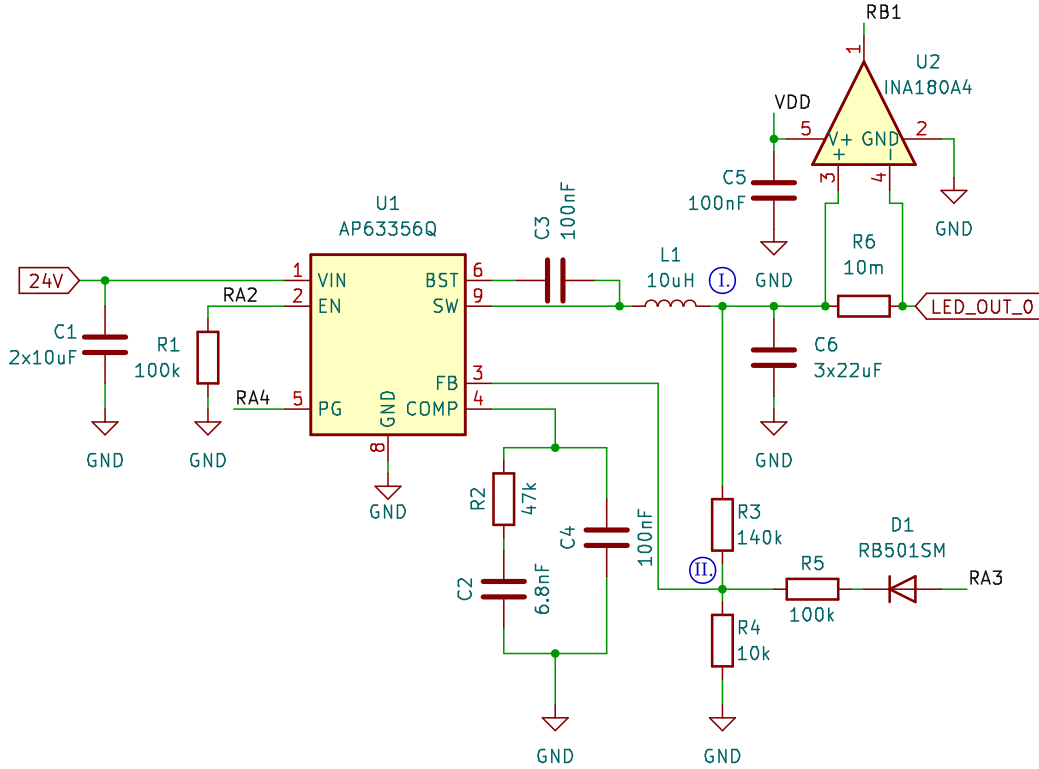
Popis schématu a výpočty hodnot součástek

Zjednodušené schéma pro jeden ovládaný kanál se nachází na obr. 4.1, při výpočtech bude použito označení součástek z tohoto schématu. Úplné schéma modulu pak lze nalézt v příloze C.

Jako BUCK kontroler je zvolen čip AP63356Q vyvinutý společností Diodes incorporated, jedná se o úspornou a velmi malou součástku, která v sobě integruje oba potřebné MOSFET tranzistory a spíná s pevně danou frekvencí $f_{SW} = 450 \text{ kHz}$ [13]. Pro snímání proudu poslouží čip INA180A4 firmy Texas Instruments [14].

Z obrázku je vidět, že pro ovládání jednoho kanálu LED pásků jsou využity 4 piny MCU, RA4 a RB1 fungují jako vstupy, RA2 a RA3 pak jako výstupy. Rezistor R_1 drží BUCK měnič ve vypnutém stavu dokud mikrokontroler nenastaví hodnotu pinu RA2 na logickou 1. Pin RA4 je pak čipem (U_1) stažen k zemi vždy, když na výstupu není odpovídající nastavené napětí.

Nastavení výstupního napětí měniče je dosaženo za pomoci zpětnovazební smyčky mezi výstupním uzlem (I.) a zpětnovazebním pinem FB (uzel II.). V uzlu II. je drženo konstantní napětí $0,8 \text{ V}$ [13], poměrem rezistorů R_3 a R_4 je pak definováno výstupní napětí. Zvolíme hodnotu odporu $R_4 = 10 \text{ k}\Omega$, pro maximální požadované



Obr. 4.1: Zjednodušené schéma ovladače LED. Vytvořeno v KiCad 7.0.

napětí $U_{outMAX} = 12\text{ V}$ platí:

$$R_3 = R_4 \cdot \left(\frac{U_{outMAX}}{0,8} - 1 \right) = 10\text{ k} \cdot \left(\frac{12}{0,8} - 1 \right) = 140\text{ k}\Omega \quad (4.2)$$

Na pin RA3 mikrokontroleru je přivedeno analogové napětí z periferie DAC popř. PWM signál a skrze rezistor R_5 (a ochrannou diodu) pak teče proud do rezistoru R_4 , úbytek napětí na tomto rezistoru je ale konstantní ($0,8\text{ V}$) a tedy je konstantní i proud rezistorem. Z prvního Kirchhoffova zákona pak víme, že proud rezistorem R_3 klesne o hodnotu proudu dodanou z pinu RA3, tím klesne také napětí na výstupu měniče a dojde ke ztlumení jasu LED pásu. Citlivost (nebo také rozsah) změny je definován hodnotou R_5 , snížením jeho hodnoty lze dosáhnout na výstupu ještě nižšího napětí. Pro hodnotu $R_5 = 100\text{ k}\Omega$ zobrazenou ve schématu lze nejnižší možné napětí vypočítat následovně:

$$U_{outMIN} = 0,8 + R_3 \cdot I_3 = 0,8 + R_3 \cdot \left(\frac{0,8}{R_4} - \frac{U_{VDD} - U_{D1}}{R_4 + R_5} \right) \quad (4.3)$$

Kdy $U_{VDD} = 3,3\text{ V}$ je maximální napětí pinu RA3 a $U_{D1} = 0,35\text{ V}$ je prahové napětí zvolené diody. Po dosazení získáme:

$$U_{outMIN} = 0,8 + 140\text{ k} \cdot \left(\frac{0,8}{10\text{ k}} - \frac{3,3 - 0,35}{10\text{ k} + 100\text{ k}} \right) = 8,25\text{ V} \quad (4.4)$$

Toto napětí by mělo být dostatečně nízké k úplnému zhasnutí LED pásku.

TODO: Kompenzace

V dalším kroku je stanovena hodnota induktoru L_1 . Výrobce doporučuje zvolit zvlnění proudu induktorem (ripple) ΔI_L jako 30 až 50 % maximálního proudu čipu. Při zvolení střední hodnoty 40 % získáme:

$$\Delta I_L = 0,4 \cdot I_{IC-max} = 0,4 \cdot 3,5 = 1,4 \text{ A} \quad (4.5)$$

Odpovídající hodnota indukčnosti je vypočtena následujícím vztahem:

$$L_1 = \frac{U_{outMAX} \cdot (U_{in} - U_{outMAX})}{U_{in} \cdot \Delta I_L \cdot f_{SW}} \quad (4.6)$$

Po dosazení získáme:

$$L_1 = \frac{12 \cdot (24 - 12)}{24 \cdot 1,4 \cdot 450 \text{ k}} = 9,52 \text{ } \mu\text{H} \quad (4.7)$$

Zvolíme nejbližší běžně používanou hodnotu $L_1 = 10 \text{ } \mu\text{H}$.

Pro vstupní (C_1) a výstupní (C_6) kapacitu použijeme hodnoty doporučené výrobcem, stejně tak pro bootstrap kondenzátor C_3 .

Poslední součástíou zůstává měřicí rezistor R_6 . Tímto rezistorem protéká celý výstupní proud měniče, v rámci minimalizace ztrátového výkonu by měl mít tedy co nejmenší odpor. Musíme ovšem také brát v potaz rozsah měřicího zesilovače INA180A4. Tato součástka se vyrábí v několika variantách, byla zvolena varianta s nejvyšším ziskem $G_{INA} = 200$ pro zachování co nejnižší hodnoty rezistoru, výstupní napětí zesilovače je v rozsahu 0 až 3,3 V (VDD mikrokontroleru). Při maximálním očekávaném proudu chceme dosáhnout horní hranice rozsahu zesilovače, z této podmínky vyplývá vztah pro výpočet R_6 :

$$R_6 = \frac{U_{VDD}}{I_{max} \cdot G_{INA}} = \frac{3,3}{1,66 \cdot 200} = 9,94 \text{ m}\Omega \quad (4.8)$$

Zvolíme blízkou hodnotu $R_6 = 10 \text{ m}\Omega$.

Očekávané parametry

Pro výpočet výstupního zvlnění (v uzlu I.) chybí údaj o ekvivalentním sériovém odporu (ESR) výstupních kondenzátorů (C_6), který výrobce neuvádí. Vyjdeme tedy z typické hodnoty pro keramický kondenzátor $ESR = 15 \text{ m}\Omega$ [15], kdy počítáme s paralelní kombinací tří kondenzátorů. Očekávané výstupní zvlnění je tedy přibližně:

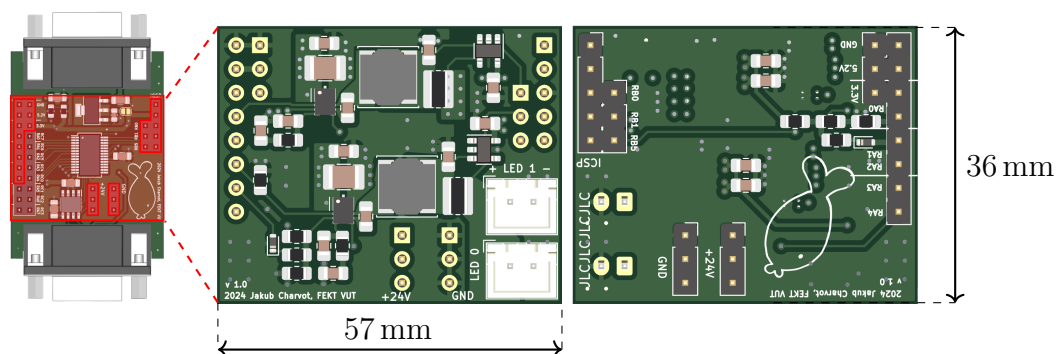
$$\Delta U_{out} = \Delta I_L \cdot \left(\frac{ESR}{3} + \frac{1}{8 \cdot f_{SW} \cdot C_6} \right) \quad (4.9)$$

$$\Delta U_{out} = 1,4 \cdot \left(\frac{15 \text{ m}}{3} + \frac{1}{8 \cdot 450 \text{ k} \cdot 3 \cdot 22 \text{ } \mu} \right) = 13 \text{ mV} \quad (4.10)$$

TODO: Zde velký question: Pokusit se o přesnější výpočet ztrát a účinnosti nebo se odkázat na kalkulačku výrobce, která stejně bude nejpřesnější?

4.1.2 Tvorba DPS

Jak již bylo zmíněno v úvodu kapitoly, jedná se o dceřinnou desku pro obecný modul periferie, čímž jsou jasné určeny její maximální rozměry. Stejně jako u předešlých návrhů, i zde je použita čtyřvrstvá struktura viz obr. 2.5. Vizualizace návrhu se nachází na obr. 4.2 spolu s ukázkou sesazení s obecným modulem periferie.



Obr. 4.2: Vizualizace DPS periferie LED osvětlení.

Za účelem zvětšení plochy pro umístění součástek byly z konektorů obecného modulu vyvedeny pouze některé piny, i přesto bylo nakonec potřeba umístit komponenty také na spodní stranu DPS. Rozmístění součástek je obdobné pro oba měniče napětí a respektuje doporučení výrobce a tedy i obecná pravidla pro návrh měničů napětí [13]. Je kladen důraz na to, aby smyčka ze spínacího uzlu přes výstupní kapacitu a zem zpět do kontroleru byla co nejkratší a vedena za pomoci polygonů. Stejně tak vstupní kapacitor je umístěn přímo vedle vstupních pinů kontroleru.

4.2 Senzor teploty

Cílem tohoto modulu je kontinuálně měřit teplotu vody a naměřená data poskytovat řídící jednotce skrze sběrnici CAN. Při volbě konkrétního teplotního čidla je potřeba vzít v potaz několik faktorů:

- Přesnost a rozsah
- Časová stálost
- Složitost implementace
- Pouzdro určené pro ponoření do vody
- Cena

4.2.1 Metody měření teploty

Nejčastěji používanými součástkami určenými k měření teploty jsou nepochybně termistory a termočlánky [16]. Termistor je rezistor vytvořen z materiálu, který mění

svůj odpor v závislosti na teplotě přičemž rozlišujeme dva základní typy termistorů podle toho, zda s rostoucí teplotou jejich odpor roste (PTC termistor) anebo klesá (NTC termistor). U obou typů lze obecně říci, že závisost odporu na teplotě je značně nelineární, pro zjištění přesné teploty je tedy potřeba buďto měřená data dále zpracovat (např. mikrokontrolerem) anebo využít speciální integrovaný obvod, který výstup z připojeného termistoru linearizuje a dále propaguje buďto v analogové nebo i digitální podobě. Jelikož odpor termistoru a stejně tak i dalších součástek, potřebných k jeho zapojení, má jistou výrobní toleranci, je vhodné sensor před použitím kalibrovat.

Princip termočlánku je odlišný, jedná se o vodivé spojení dvou kovů na kterém díky Seebeckově jevu vzniká termoelektrické napětí. Velikost tohoto napětí je daná použitými materiály a je také teplotně závislá. V praxi se používá nejčastěji několik dvojic materiálů, které svými vlastnostmi a cenou nejvíce vyhovují běžným požadavkům, ty pak získaly také své označení jako termočlánky typu J, K, T nebo E (typů existuje více, uvedeny jsou nejčastěji používané [17]). Termočlánky pracují oproti ostatním sensorům s výrazně větším rozsahem teplot a mohou měřit také teploty velmi vysoké. Nevýhodou je nízké výstupní napětí, které musí být spolehlivě měřeno, tedy ideálně porovnáno s přesnou napěťovou referencí a také je potřeba, aby část zařízení, ke kterému je termočlánek připojen (tzv. studený konec), byla udržována při konstantní referenční teplotě anebo případnou změnu teploty měřila jiným způsobem a kompenzovala výpočtem [16, 17].

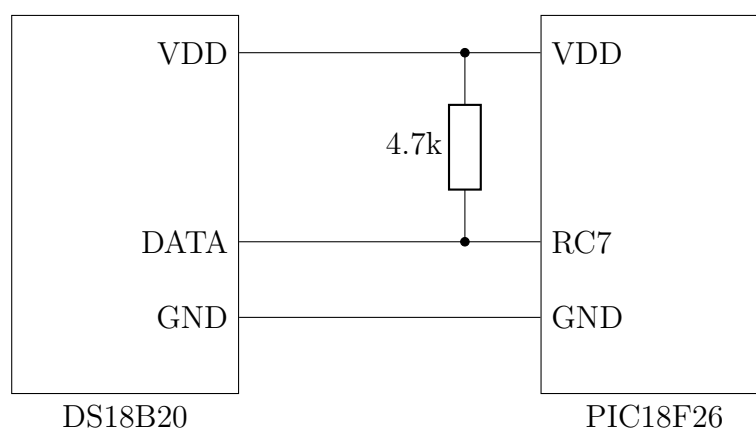
Z hlediska praxe je další často využívanou možností použití zcela integrovaného sensoru s digitálním výstupem. Pro zpracování je sice potřeba mikrokontroler, ale tyto sensory bývají od výroby kalibrovány a také jejich zapojení je velmi jednoduché, což je výhodou.

4.2.2 Realizace sensoru

Při porovnání uvedených metod se použití termočlánku jeví jako nevhodné, zejména kvůli náročnosti implementace, která současně navyšuje také cenu. Zbývá tedy rozhodnutí mezi termistoru a digitálním čidlem. Ve voděodolném pouzdře lze zakoupit jak několik variant termistorů, tak i digitální čidlo (zde DS18B20). Nejlevněji vychází termistor typu NTC, ale v porovnání s cenou celého zařízení je rozdíl v ceně zanedbatelný.

Pro realizaci sensoru bylo zvoleno digitální čidlo DS18B20, které narozdíl od termistoru není potřeba kalibrovat a výrobce garantuje přesnost $\pm 0,5^\circ\text{C}$ na celém teplotním rozsahu od -55°C do 125°C . Rozlišení sensoru je až 12 bitů přičemž minimální měřitelná změna teploty odpovídá $0,0625^\circ\text{C}$. Pro komunikaci s čidlem se využívá protokol 1-Wire, kdy datový vodič funguje obousměrně. Pro propojení čidla

s mikrokontrolerem tedy stačí využít tři vodiče a jeden pull-up rezistor, viz obr. 4.3.



Obr. 4.3: Připojení čidla DS18B20 k MCU.

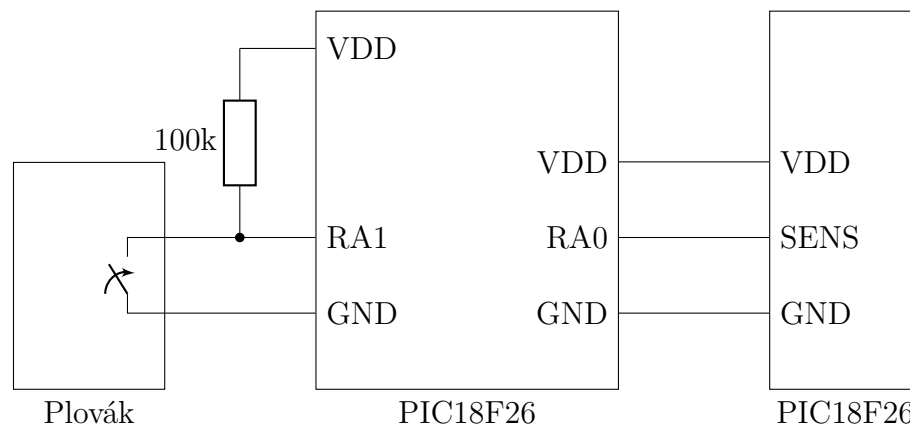
4.3 Senzor výšky hladiny

Voda v akváriu se průběžně odpojuje a je potřeba ji doplňovat. Účelem této periferie je průběžné monitorování hladiny akvária a upozornění uživatele na nutnost doplnění vody. Také může uživatele varovat v případě poškození nádrže a nežádoucího úniku vody do okolí. K realizaci tohoto modulu jsou použity dva jednoduché sensory přičemž každý funguje na jiném principu a má tedy také odlišné přednosti a nedostatky, v kombinaci tedy zvyšují celkovou spolehlivost modulu. Oba sensory se nachází na obr. ??.

První ze sensorů využívá k určení výšky hladiny vodivost (potažmo odpor) vody. Obsahuje dva sety vodivých plošek, které nejsou vodivě spojeny. Při ponoření měřící části do vody začne mezi ploškami procházet slabý proud, který je přibližně úměrný velikosti ponořené části. Tento proud otevírá tranzistor, na jehož výstupu pak vzniká stejnosměrné napětí v rozsahu přiloženého napájení (zde 0 až 3,3 V). Tento signál je přiveden na pin mikrokontroleru a následně zpracován vestaveným převodníkem. PIC18F26 obsahuje integrovanou periferii ADC s rozlišením 12 bitů, teoreticky lze tedy rozlišit 4296 úrovní [10]. Pro převod měřené hodnoty na výšku hladiny je potřeba sensor nejprve nakalibrovat. Byla tedy změřena přibližná výstupní hodnota pro minimální a maximální měřitelné ponoření sensoru a údaj je následně mikroprocesorem převeden na procenta. Úvaz v procentech ponořené části je pro uživatele univerzálním ukazatelem nezávislým na umístění sensoru, pro měření absolutní výšky hladiny by musel uživatel v systému nastavit výšku umístění sensoru a také ji pokaždé měnit v případě změny jeho pozice.

Druhým senzorem je jednoduchý plovák obsahující mechanický spínač, který je při ponoření do vody rozepnut. V případě poklesu hladiny pod zvolenou úroveň je pak spínač sepnut.

Propojení mikrokontroleru s oběma senzory se nachází na obr. 4.4.



Obr. 4.4: Připojení sensorů hladiny k MCU.

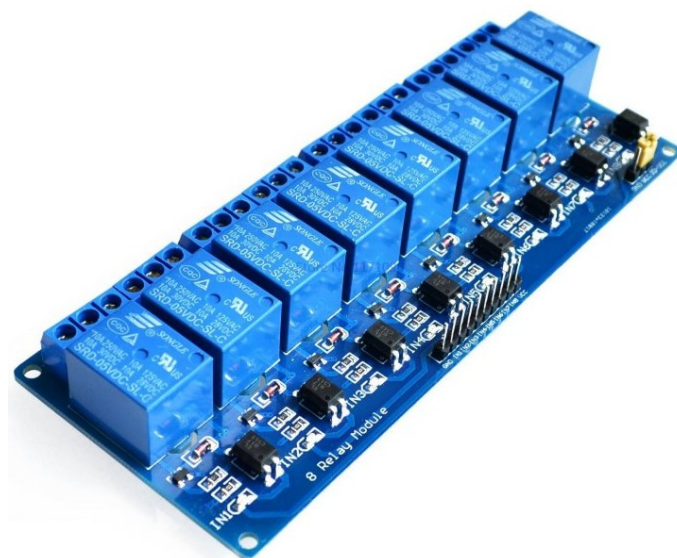
4.4 Senzor pH

4.5 Ovládání 230V periferií

Jak vyplývá z požadavků zařízení a přehledu používané akvaristické techniky, pro automatizovaný provoz akvária je nutné umožnit řídicí jednotce ovládat několik okruhů se síťovým napětím a spínat tak zvláště zakoupené hotové spotřebiče pracující s tímto napětím. Jedná se typicky o ohřev vody, filtraci, popř. některé druhy osvětlení.

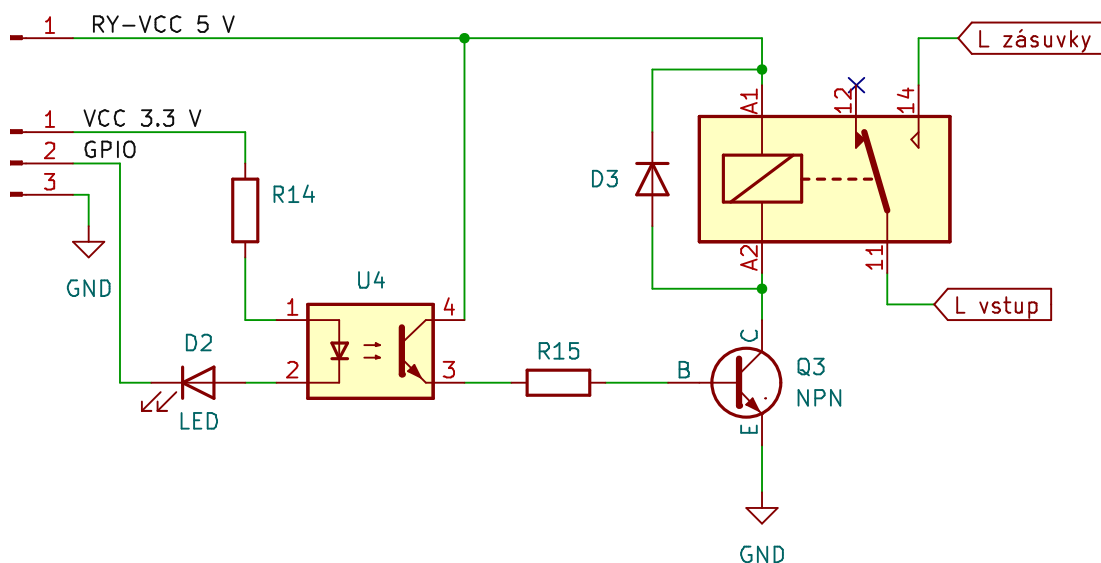
Aby uživatel mohl spínaná zařízení bezpečně připojit bez nutnosti odborné způsobilosti, nachází se na hlavním šasi zařízení čtyři standartní zásuvky (typ E) s jednofázovým napětím 230 V. Fázové vodiče jsou uvnitř zařízení přerušeny spínacími relé. Je použit předpřipravený modul disponující osmi relé [18], čtyři z nich tedy zůstanou nevyužité a slouží jako rezerva pro případ poškození některého z používaných relé nebo při potřebě budoucího rozšíření o další zásuvky.

Z důvodu nedostatku pinů na mikrokontroleru řídicí jednotky (ESP32) je k relé modulu připojen ještě jeden externí modul a to expandér GPIO pinů komunikující přes sběrnici **i2c!** [19]. Z pohledu mikrokontroleru jsou tak všechny zásuvky ovládány pomocí dvou GPIO pinů (SDA, SCL), které je navíc možné dále využít pro připojení jiných periferií jako např. OLED displeje pro zobrazení stavu zařízení.



Obr. 4.5: TODO vyměnit: Relé modul, ilustrační foto. Převzato z [18].

Relé na použitém modulu potřebuje pro spolehlivé sepnutí napětí alespoň 5 V, logické signály řídící jednotky ale pracují s napětím pouze 3,3 V. Ze schématu na obr. 4.6 je vidět, že použitý relé modul je spínán signálem logické nuly, tímto způsobem je problém s rozdílnou úrovní napájení elegantně vyřešen.



Obr. 4.6: Schéma jednoho kanálu relé modulu. Vytvořeno v KiCad 7.0.

Do budoucna by bylo možným zlepšením a rozšířením této práce zahrnutí obou zmíněných modulů přímo na DPS řídící jednotky.

5 Software

Tato kapitola se věnuje popisu návrhu softwaru pro jednotlivé části systému řízení akvária. Jak již bylo zmíněno v předchozích kapitolách, systém se skládá z řídicí jednotky, ke které jsou připojeny jednotlivé periferie a která komunikuje s webovým serverem za pomoci WiFi. Každá ze zmíněných částí pak potřebuje vlastní zdrojový kód.

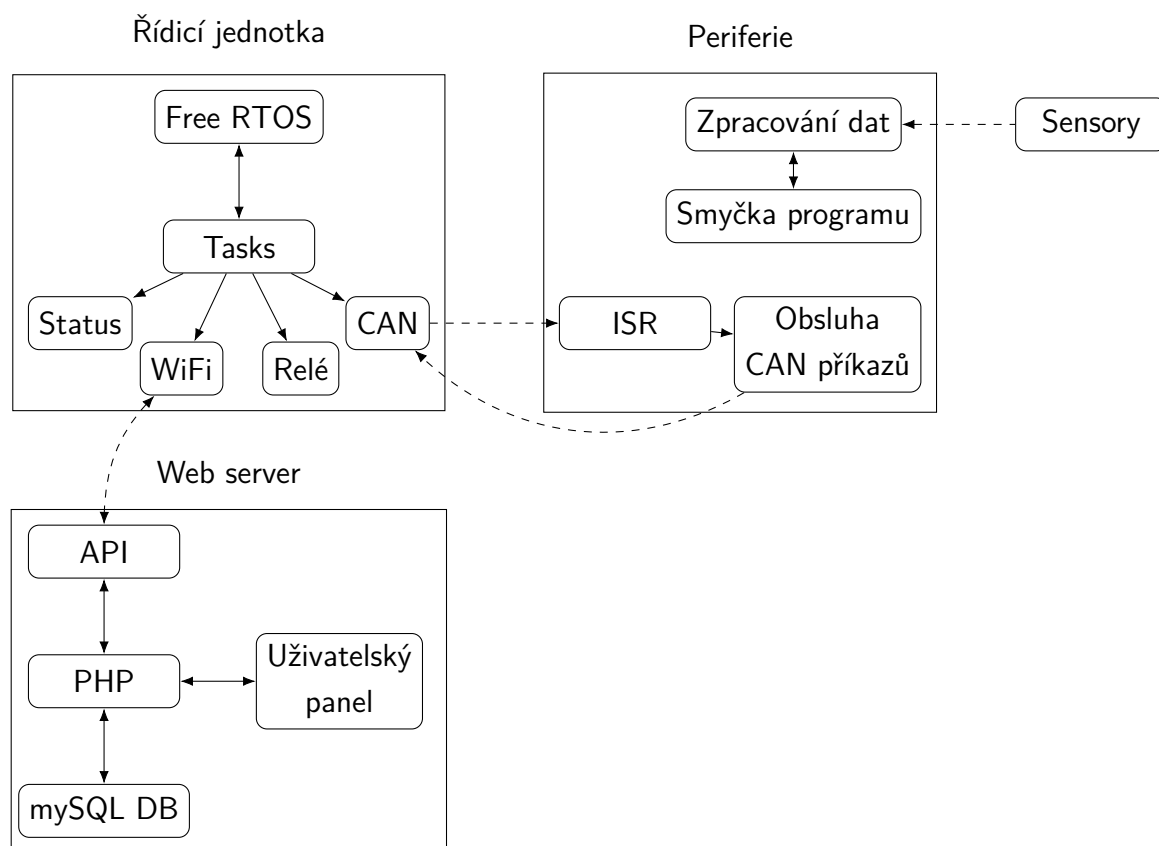
K programování a testování byla použita dvě vývojová prostředí. Visual Studio Code je open source řešení spravované společností Microsoft a díky široké škále doplňků umožňuje velmi univerzální použití. S přidaným rozšířením ESP-IDF je také preferovaným prostředím společnosti Espressif, bylo tedy využito pro tvorbu kódu řídicí jednotky, stejně tak i pro webové rozhraní. Pro programování periférií pak bylo zvoleno prostředí MPLAB X. Jedná se o řešení společnosti Microchip určené speciálně pro programování mikrokontrolerů této firmy. Součástí je kromě samotného editoru také kompilátor, možnost debugování kódu nebo modul zvaný Code Configurator sloužící pro generování jednoduchého HAL (Hardware Abstraction Level) kódu.

V této chvíli software odpovídá podobě zbytku zařízení – tedy jedná se o prototyp určený primárně k demonstraci funkce zařízení. Aby byl kód použitelný v reálné aplikaci a choval se zde spolehlivě je potřeba podrobit jej rozsáhlejšímu testování a také lépe ošetřit chování zařízení v různých nežádoucích stavech.

5.1 Architektura

Na obr. 5.1 se nachází blokové schéma systému z pohledu softwaru. Obrázek slouží primárně pro lepší orientaci čtenáře v této kapitole, obsahuje pouze klíčové části a některé věci zjednodušuje. Podrobněji se jednotlivým blokům věnují další podkapitoly. Obdélník popsáný v obrázku jako „Periferie“ popisuje strukturu kódu platnou pro všechny periferie, je ale samozřejmě že periférií této struktury bude v systému připojeno vícero.

Propojení přerušovanými šipkami v obrázku značí komunikaci mezi dvěma částmi s odlišným programem. Z hlediska internetové komunikace se navržené zařízení chová jako klient, tedy neposlouchá na žádném portu a z vnější sítě není nijak dostupné. Webový server disponuje datovým rozhraním (API), kterého se zařízení v pravidelných intervalech dotazuje na případné změny konfigurace a prostřednictvím kterého zasílá na server data ze svého běhu. Při komunikaci mezi řídicí jednotkou a perifériemi pak řídicí jednotka funguje jako „master“ a periferie odpovídají pouze v reakci na dotaz z její strany.



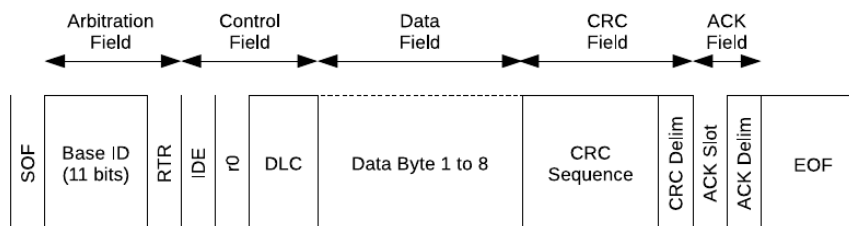
Obr. 5.1: Zjednodušená architektura softwaru systému.

Jelikož jsou řídicí jednotka i periferie programovány ve stejném jazyce (C/C++), lze mezi nimi část kódu sdílet. Tímto způsobem lze částečně předejít chybám v komunikaci modulů mezi sebou. Společná část kódu obsahuje definice datových typů a konstant používaných při komunikaci po sběrnici CAN.

5.2 Popis CAN komunikace

Protokol CAN je poměrně rozsáhlý a velká část organizace komunikace je řešena přímo hardwarovou periferií mikrokontroleru. Pro úspěšnou a efektivní komunikaci bylo potřeba nastavit oba typy mikrokontrolerů stejně a stanovit společný standart komunikace. Systém popsáný v této práci pracuje s frekvencí 125 kHz a používá standartní strukturu rámců s identifikátorem zprávy dlouhým 11 bitů (standart CAN 2.0B umožňuje také délku 29 bitů). Sběrnice CAN má implementovaný princip arbitrace, pokud začne komunikovat více zařízení současně, odešle se zpráva mající identifikátor s nejvyšší prioritou. Logická nula se jeví na sběrnici jako dominantní, jednička naopak jako recesivní. Pokud zařízení odesílá recesivní signál a zároveň čte ze sběrnice signál dominantní, znamená to pro něj ztrátu arbitrace a přestává vysílat,

jelikož na sběrnici je v danou chvíli vysílaná zpráva s vyšší prioritou. Po skončení vysílání pak přerušené zařízení pokus opakuje.



Obr. 5.2: Struktura datového rámce sběrnice CAN. Převzato z [3]

V navrženém systému nese identifikátor zprávy dvě informace. První tři odesílané bity značí typ zprávy. Rozlišena je zpráva určené všem jednotkám (BR – Broadcast), zpráva od řídicí jednotky k perifériím (TS – master To Slave), od periférie zpět k řídicí jednotce (TM – slave To Master) a debug zpráva sloužící k odeslání diagnostických dat nezávisle na ostatní komunikaci. Zbýlých 8 bitů pak tvoří adresu jednotky odesílající zprávu (v případě TM) resp. zprávu přijímající (v případě TS).

Adresy jednotlivých modulů by měly být po startu systému nebo připojení nové jednotky automaticky přiděleny tak, aby nedošlo ke kolizi adres ani v případě připojení několika sensorů stejného typu. Princip adresace spočívá v sérii několika BR zpráv. Po startu systému pošle řídicí jednotka požadavek na adresaci, jako odpověď odešlou jednotky periférií své unikátní sériové číslo přičemž pouze jedna ze zpráv vyhraje arbitraci. Řídicí jednotka odpoví zprávou, která obsahuje sériové číslo úspěšné jednotky a přidělenou osmibitovou adresu. Následně zopakuje požadavek adresace a odpoví již pouze jednotky bez adresy, po dokončení adresaci pak neodpoví žádná jednotka. Pokud je do běžícího systému připojena nová periférie, odešle sama BR zprávu s požadavkem na přidělení adresy.

Ačkoliv je tento princip vymyšlen, v rámci prototypu prozatím není implementován a otestován a každý typ periférie má pevně přidělenou adresu, lze tedy připojit pouze jednu periférii stejného typu. V současné chvíli je toto řešení dostačující.

5.3 Firmware řídicí jednotky

Firma Espressif nabízí pro své mikrokontrolery dva základní frameworky. Oba jsou vyvíjeny jako open-source a jsou tedy také volně dostupné pro jakékoliv použití. Univerzálním řešením vhodným i pro komerční aplikace je ESP-IDF (Espressif Integrated Development Framework). Pro hobby projekty lze využít také Arduino framework, který je taktéž oficiálně podporovaný. Poslední novinkou je pak možnost

programování v jazyce Rust namísto klasického C/C++, tento projekt je vytvářen komunitou uživatelů za podpory společnosti Espressif, prozatím ale nebyla oficiálně vydána stabilní verze.

5.3.1 FreeRTOS






















V rámci této práce je využit framework ESP-IDF, který obsahuje podporu pro jednoduchý operační systém Free RTOS a také drivery pro všechny hardwarové periferie mikrokontroleru [20].

Systém FreeRTOS umožňuje vytvářet tzv. tasky neboli samostatné procesy které běží z pohledu uživatele paralelně. Jelikož má zvolený mikroprocesor dvě jádra, mohou dva tasky běžet skutečně paralelně, více procesů se pak ve svém běhu střídá a běží tak přerušovaně, vůči sobě pseudoparalelně. O tuto režii se stará samotný operační systém a programátor má několik možností jak tento proces ovlivnit. V případě vytížení procesoru systém přiděluje čas na základě nastavených priorit a dá přednost tasku s vyšší prioritou, může se tak stát, že některý proces zůstane pozastaven na dlouhou dobu. Při tvorbě kódu je potřeba mít toto na paměti, vhodně zvolit priority tasků a také průběžně sledovat vytížení procesoru jednotlivými tasky.

Aby byl kód tzv. thread-safe, tedy bezpečný pro přístup z více vláken, je potřeba ošetřit případy, kdy více tasků pracuje se stejnými daty nebo přistupuje ke stejné periférii mikrokontroleru. K tomuto účelu FreeRTOS nabízí synchronizační objekty jako jsou mutexy, semaforey případně fronty.

5.3.2 Indikace stavu zařízení

Šasi řídicí jednotky je vybaveno adresovatelným barevným LED páskem sestávajícím ze čtyř diod jejichž úkolem je signalizovat uživateli stav zařízení. Jednotlivé stavy spolu s popisem diod jsou zobrazeny na obr. 5.3. Každý task, který je součástí procesu diagnostiky má svou vlastní globální proměnnou do které ukládá svůj stav. Dvakrát za vteřinu se pak spustí jednoduchá diagnostická funkce (běží v rámci vlastního tasku), která jednotlivé stavové proměnné přečte, vyhodnotí celkový stav zařízení a aktualizuje barvu stavových LED.

Stav zařízení	Stav WiFi	Alarm 1	Alarm 2
			
Nedefinováno 	Nedefinováno 	Nedefinováno 	Nedefinováno 
Inicializace 	Inicializace 	Upozornění 	Upozornění 
Ok 	Připojeno 	Alarm 	Alarm 
Chyba 	Online 	OK 	OK 
	Chyba 		

Obr. 5.3: Stavové LED řídicí jednotky.

5.3.3

5.4 Firmware periferií

5.5 Webové rozhraní

Aby bylo možné systém konfigurovat a monitorovat bylo zapotřebí navrhnout a vytvořit webové rozhraní. Důležitým krokem v rozhodování byla volba, zda bude MCU řídicí jednotky sloužit přímo jako webový server nebo pouze jako klient. První scénář klade podstatně vyšší nároky na zatížení a paměť MCU, výhodou je ale velmi jednoduchý systém, který funguje samostatně bez nutnosti externího serveru případně také bez připojení k internetu (ESP32 může sloužit přímo jako přístupový bod). Výhodou druhé varianty je větší flexibilita, externí server má nesrovnatelně vyšší výkon a kapacitu úložiště a umožní tak tvorbu mnohem komplexnější webové stránky, která bude (v případě připojení serveru do internetu) dostupná odkudkoliv. Server zároveň může ukládat měřená data a ta tedy budou dostupná i v případě, že samotné zařízení je mimo provoz nebo není připojeno k síti.

V rámci realizace této práce byla zvolena varianta externího serveru. Jádrem vytvořené webové aplikace tvoří program v jazyce PHP, který zpracovává jak požadavky uživatele, tak i samotného zařízení. Na pozadí dále běží databázový server s otevře-

ným systémem MySQL sloužící k uchování provozních dat. V databázi jsou uloženy údaje o uživateli, systémech akvárií (tedy řídicí jednotka a k ní připojené periférie) a jejich konfigurace a data získaná ze senzorů. Struktura tabulek databáze je zobrazena na obr. ???. V záznamu odpovídajícímu danému systému akvária je uložen číselný údaj o aktuální verzi konfigurace. Pokud uživatel konfiguraci modifikuje, toto číslo se inkrementuje na což následně zareaguje zařízení a vyžádá si ze serveru novou verzi konfigurace.

Zařízení komunikuje s webem pomocí aplikačního rozhraní (API), které tvoří nenáročný způsob komunikace využívající formát JSON. Jednotlivé adresy API rozhraní jsou přehledně popsány v tab. ???.

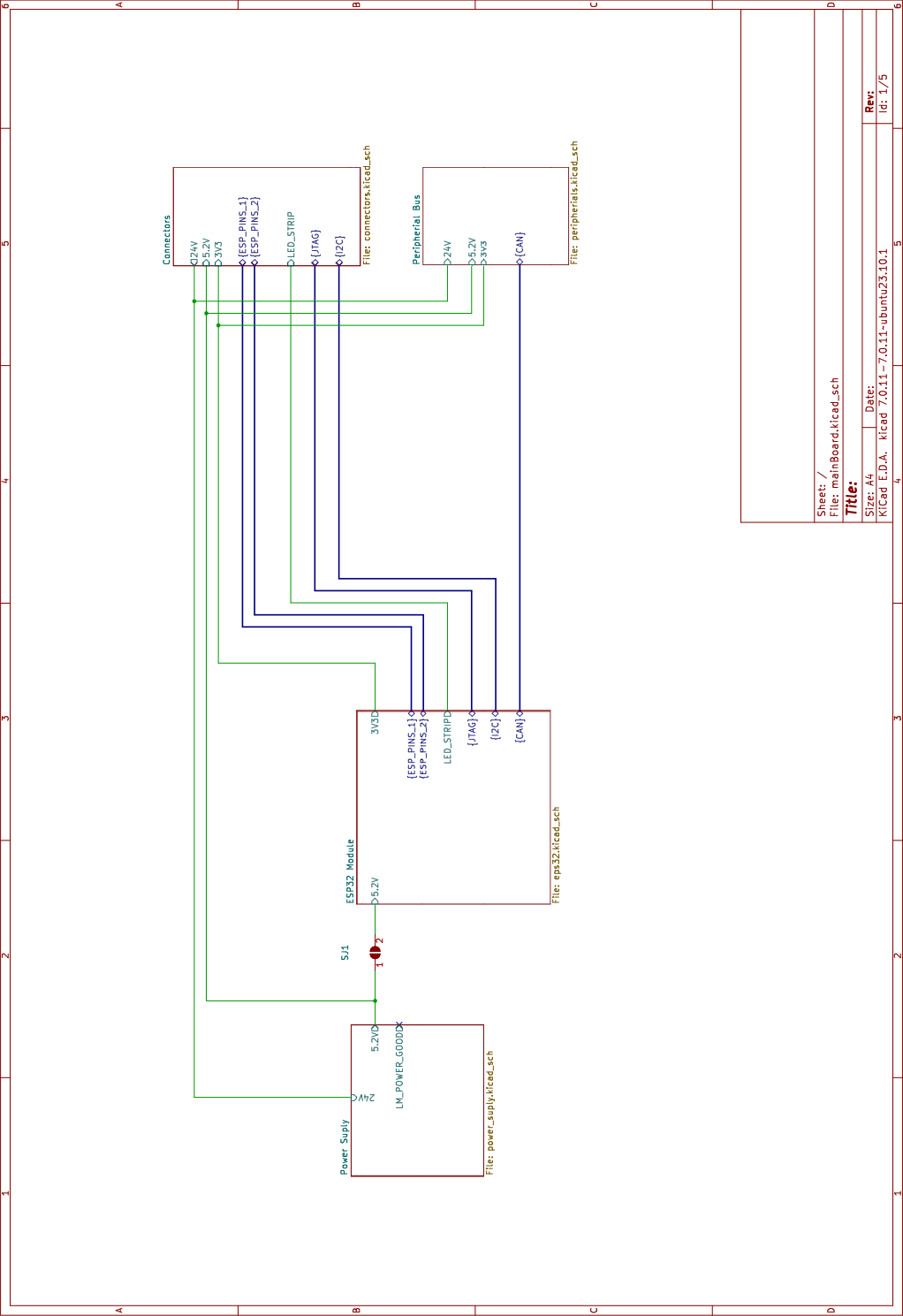
6 Sestavení a testování

Seznam příloh

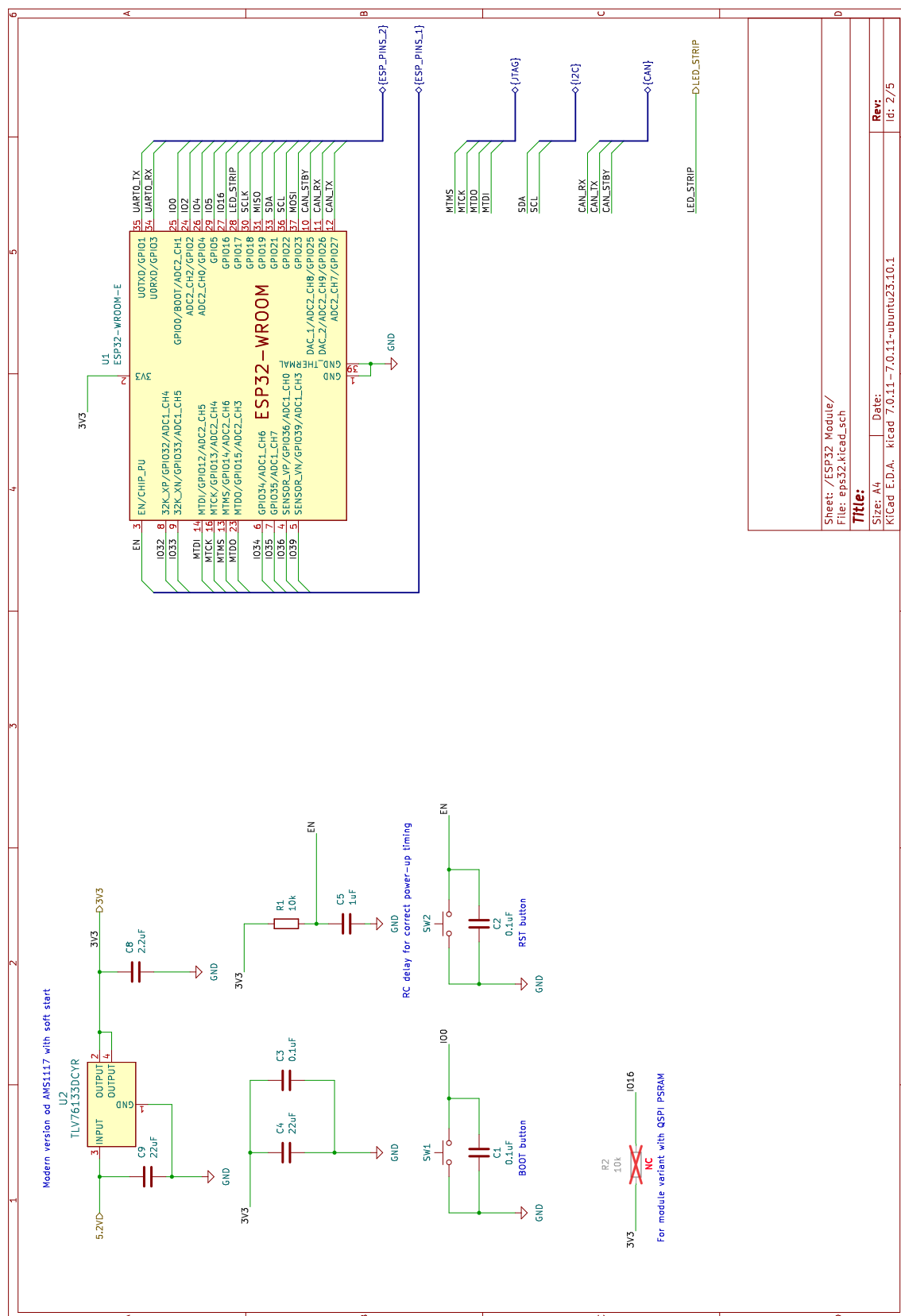
A	Schéma řídicí jednotky	34
A.1	Blokové schéma	34
A.2	Zapojení MCU	35
A.3	Napájecí obvod	36
A.4	Konektory	37
A.5	Sběrnice periférií	38
B	Schéma modulu periférií	39
C	Schéma modulu LED osvětlení	40

A Schéma řídicí jednotky

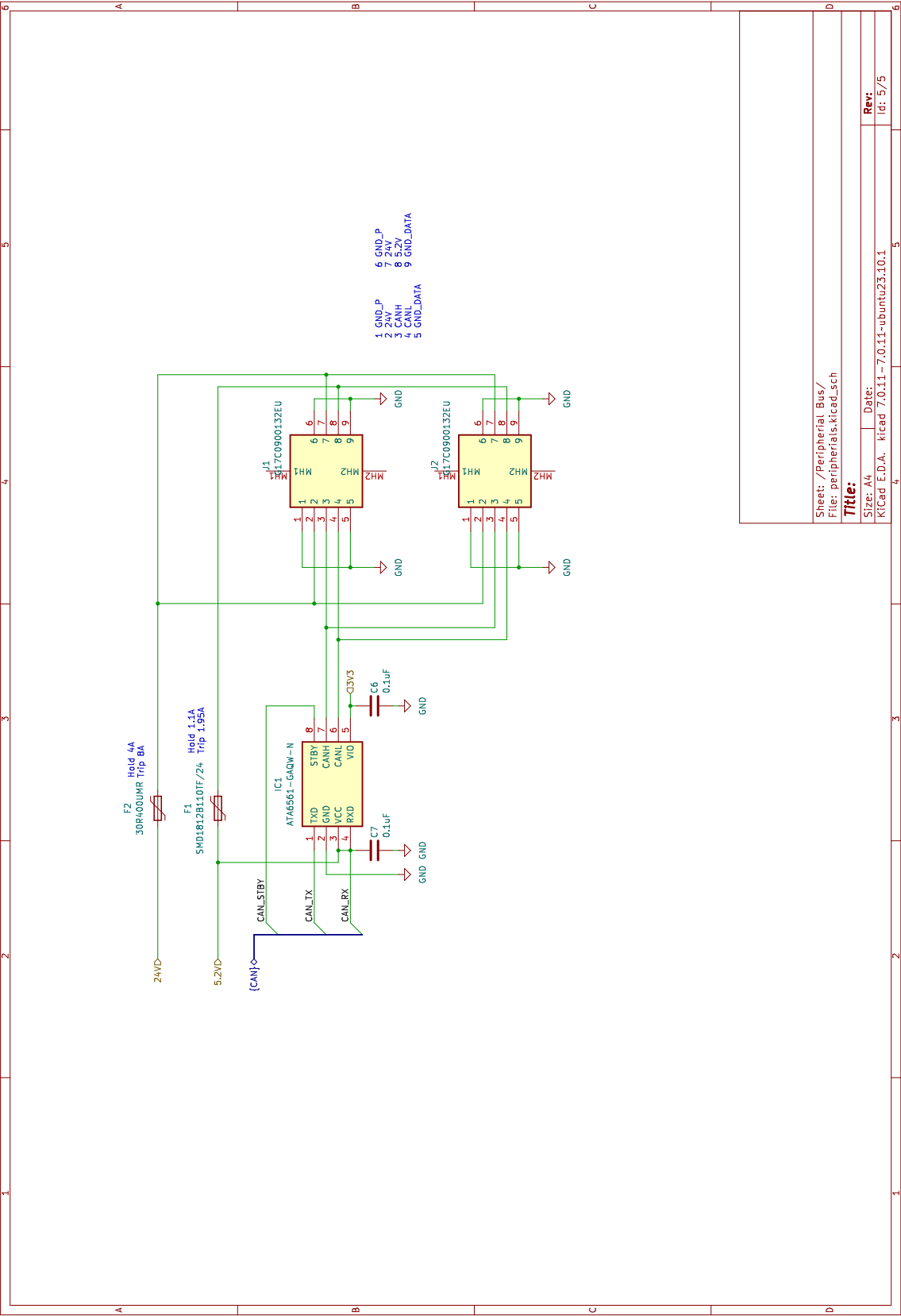
A.1 Blokové schéma



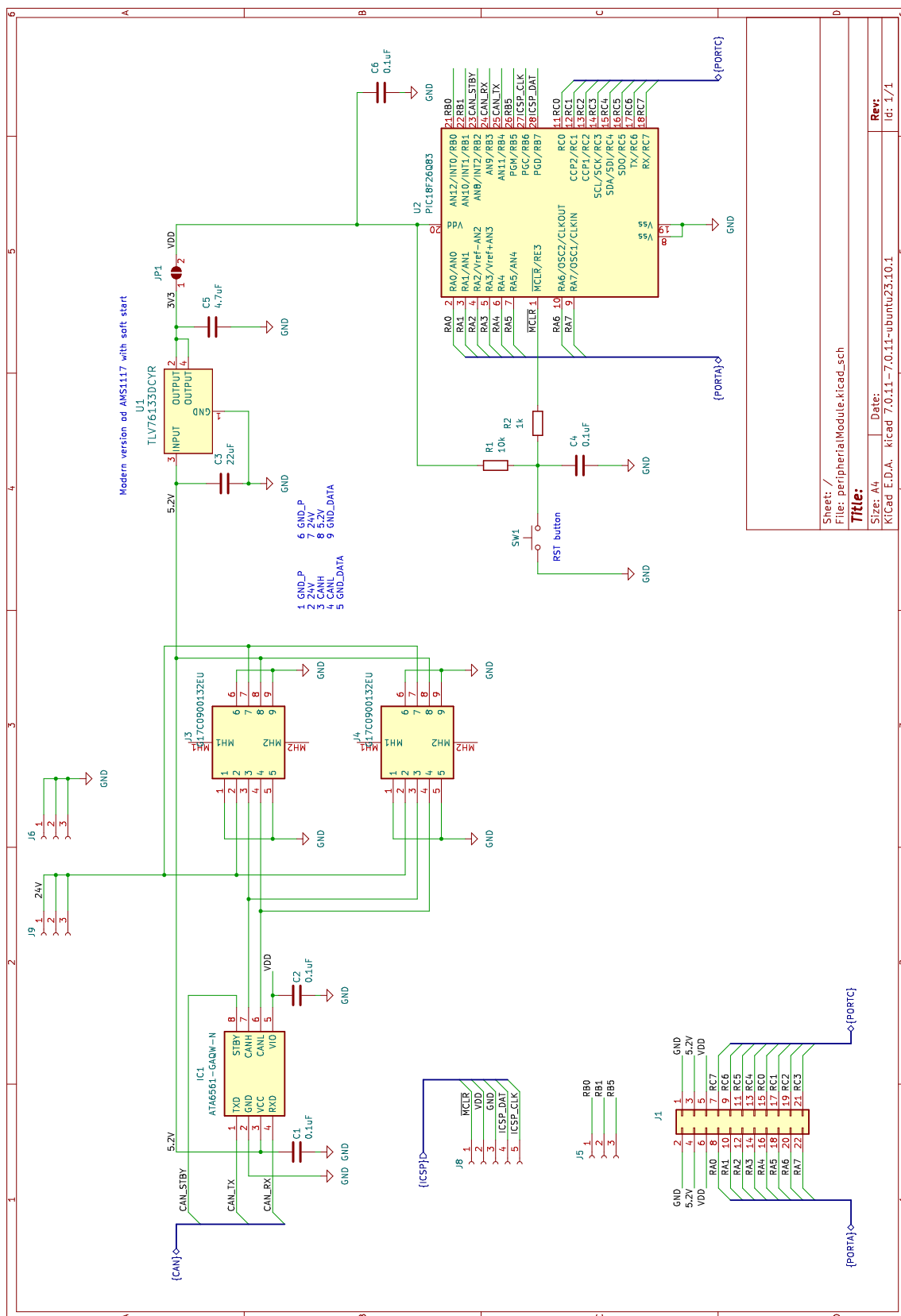
A.2 Zapojení MCU



A.5 Sběrnice periferií



B Schéma modulu periferií



C Schéma modulu LED osvětlení

