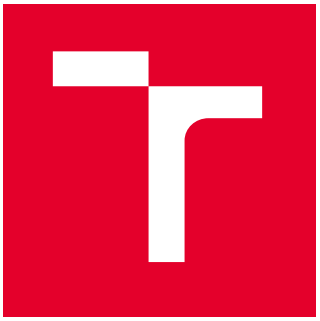


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

VZDÁLENÝ DOHLED PRO SOLÁRNÍ STŘÍDAČ

REMOTE CONTROL OF SOLAR INVERTER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Ondřej Pohorský**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. František Burian, Ph.D.**

SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Ondřej Pohorský ***ID:*** 221011

Ročník: 2 ***Akademický rok:*** 2023/24

NÁZEV TÉMATU:

Vzdálený dohled pro solární střídač

POKYNY PRO VYPRACOVÁNÍ:

Úkolem studenta je navrhnout vzdálený dohled pro solární střídač. Vizualizace čtených hodnot bude prováděna za pomocí jednoduchých webových stránek dostupných přes wifi. Součástí dohledu je i jednoduché ovládání spotřebičů pomocí 12V signálů.

1. Nastudujte komunikační protokoly sloužící pro komunikaci se střídačem Sofar.
2. Proveďte rešerši v oblasti dostupných mikrokontrolerů a vyberte vhodný.
3. Navrhněte jednoduchou modulovou elektroniku, která bude schopna komunikovat se střídačem.
4. Realizujte ovládací elektroniku měniče.
5. Implementujte webové stránky, které zobrazují stav zařízení a umožňují nastavovat 12V výstupy.
6. Implementujte funkci archivace historických měření do FLASH a její vizualizaci na webu.

DOPORUČENÁ LITERATURA:

RAJIV K., Varma. Smart Solar PV Inverters with Advanced Grid Support Functionalities. Wiley, 2021. ISBN 978-1119214182.

Termín zadání: 5.2.2024 ***Termín odevzdání:*** 15.5.2024

Vedoucí práce: Ing. František Burian, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato práce se zabývá návrhem a realizací zařízení pro vzdálený dohled solárního střídače SOFAR HYD-10-KTL-3PH. Zadání této práce vzniklo na základě poznatků a požadavků autora práce. Cílem práce je vytvořit zařízení komunikující se střídačem. Zařízení bude zobrazovat podstatné parametry na webových stránkách. Z webových stránek je také možné ovládání výstupů zařízení, kterými lze spínat spotřebiče.

KLÍČOVÁ SLOVA

SOFAR, HYD-10-KTL, vzdálený dohled, solární střídač, webové stránky, HTTP, Wi-Fi

ABSTRACT

This thesis deals with the design and implementation of a remote monitoring device for the solar inverter SOFAR HYD-10-KTL-3PH. The assignment of this thesis was based on the knowledge and requirements of the author. The aim of this thesis is to create a device communicating with the inverter. The device will display the essential parameters on the website. From the website it should also be possible to control the outputs of the device, which can be used to switch appliances.

KEYWORDS

SOFAR, HYD-10-KTL, remote control, solar inverter, web pages, HTTP, Wi-Fi

POHORSKÝ, Ondřej. *Vzdálený dohled pro solární střídač*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2024. Vedoucí práce: Ing. František Burian, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Ondřej Pohorský
VUT ID autora:	221011
Typ práce:	Diplomová práce
Akademický rok:	2023/24
Téma závěrečné práce:	Vzdálený dohled pro solární střídač

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Františku Burianovi, Ph.D. za účinnou a metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Brno . . . .

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Tato diplomová práce se věnuje návrhu zařízení, které bude zajišťovat vzdálený dohled pro solární střídač SOFAR HYD-10-KTL-3PH. Konkrétněji tedy prostudování způsobu, kterým je možné s tímto střídačem komunikovat. Na základě zjištěného způsobu komunikace bude proveden návrh zařízení, které bude komunikaci obsluhovat.

Návrh bude spočívat v první řadě ve výběru vhodného mikrokontroléru. Následně k hardwerovému návrhu vhodného komunikačního rozhraní pro komunikaci se střídačem. Zařízení bude také obsahovat digitální výstupy určené ke spínání spotřebičů. Spínání muže být provedeno například na základě aktuálního výkonu střídače a aktuální spotřeby.

Další funkcí zařízení budou výstupy ve standardní napěťové úrovni 0 až 10 V. Tyto výstupy budou sloužit k ovládání spotřebičů, které mají možnost přizpůsobit hodnotu aktuálního příkonu. Jedná se například o spotřebiče jako klimatizace, nabíječka elektromobilu, tepelné čerpadlo apod. Signálem 0 až 10 V je možné také ovládat SSR relé, kterým lze přizpůsobovat příkon odporových zátěží typu elektrický bojler nebo přímotop.

Téma této práce vzniklo na základě poznatků autora práce, který vlastní solární elektrárnu se střídačem SOFAR HYD-10-KTL-3PH. Společnost SOFAR nabízí vzdálený dohled založený na platformě SOLARMAN. Jedná se o cloudové řešení, kdy jsou data ukládána na server pomocí Wi-Fi datalogeru v intervalu pěti minut. Návrh vlastního řešení tedy vychází zejména z těchto nedostatků ve vzdáleném dohledu od výrobce střídače. Cílem je dosáhnout zobrazení aktuálních hodnot výroby, spotřeby, odběru z distribuční sítě, a informací o baterii. Tyto hodnoty budu aktualizovány v podstatně kratším intervalu (v řádu sekund). Hodnoty budou zobrazované na webové stránce. Na této stránce také bude možné ovládat digitální a analogové výstupy zařízení.

Nejpodstatnější data budou v rozumném časovém intervalu uchovávána v paměti, která bude součástí zařízení. Vlastní řešení je také výhodné z důvodu bezpečnosti a ochrany dat. Na základě údajů o spotřebě lze například velmi snadno rozpoznat nepřítomnost osob v objektu.

Práce je rozdělena na teoretickou a praktickou část. Teoretická část obsahuje veškeré podklady podstatné pro úspěšnou realizaci zařízení dle zadání práce. Samotná realizace je podrobně rozebrána právě v teoretické části práce.

1 Teoretická část studentské práce

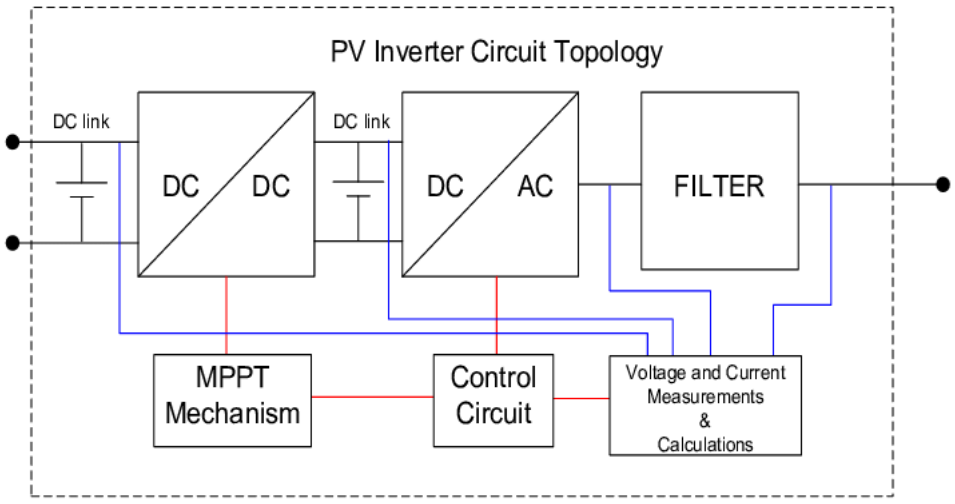
Tato část semestrální práce se zabývá teoretickým rozborem základních částí práce vyplývajících ze zadaní semestrální práce.

1.1 Solární střídače SOFAR

Tato kapitola v první podkapitole ve stručnosti popisuje co jsou solární střídače a jejich dělení. Ve druhé podkapitole je představena společnost SOFARSOLAR zabývající se výrobou solárních střídačů.

1.1.1 Solární střídače

Fotovoltaické panely jsou zdrojem stejnosměrného proudu. Ovšem pro použitelnost v domácnostech, případně dodávek do distribuční sítě je nutné stejnosměrný proud přeměnit na střídavý a upravit hodnotu napětí tak, aby odpovídala parametrům distribuční sítě, případně požadavkům připojovaných elektrických zařízení. [1] Na následujícím obrázku je znázorněné jednoduché blokové schéma solárního střídače.



Obr. 1.1: Blokové schéma solárního střídače [3]

Z předchozího obrázku vyplývá, že solární střídač je složen ze 6 základních částí. Na vstupu fotovoltaických panelů DC/DC měnič napětí. Tento měnič napětí je řízen MPPT mechanizmem. MPPT (Maximum Power Point Tracker) jak již název napovídá zajišťuje maximální výtěžnost solárních panelů. Ke sledování maximálního výkonu se používá metoda přírůstkové vodivosti s integračním regulátorem pro minimalizaci chyby při sledování MPP. Další nedílnou součástí solárních střídačů je

DC/AC měnič napětí. Tento měnič napětí zajišťuje přeměnu stejnosměrného napětí na střídavé a zabezpečuje požadované parametry výstupního napětí. Jako je například v České republice běžně používané napětí AC 230 V / 50 Hz. Výstup tohoto napěťového měniče je opatřen filtračními obvody, které slouží k odstranění nežádoucích harmonických složek. Velmi důležitou součástí solárních střídačů jsou napěťové a proudové měřící obvody. Tyto obvody jsou umístěny na vstupech a výstupech avizovaných měničů a filtrů a zajišťují vstupní data pro správnou regulaci všech součástí solárního střídače. Nad všemi předchozími částmi stojí řídící obvody zajišťující správný chod a funkčnost solárního střídače. Řídící obvody solárních střídačů jsou obvykle složeny z digitálních signálových procesorů.[3]

Z hlediska aplikace rozlišujeme tři hlavní typy měničů.

- **Síťové měniče:** Tyto měniče jsou určeny pro připojení k distribuční síti a jsou nezbytné pro distribuci elektrické energie do sítě. Podstatné je, že tyto měniče se při výpadku distribuční sítě odpojí a nevyrábí elektrickou energii. Odpojení od distribuční sítě je nezbytné pro ochranu života a zdraví pracovníků při opravách na distribuční síti.[1]
- **Ostrovní měniče:** Ostrovní měniče jsou zejména využívány v oblastech, kde není zavedena distribuční síť. Jsou tedy schopné fungovat nezávisle na distribuční síti, ale ke správnému chodu je ve většině případů potřeba, aby byl ostrovní měnič doplněn o baterii.[1]
- **Hybridní měniče** Sloučením předchozích dvou typů měničů jsou hybridní měniče. Zpravidla mají tyto měniče dva výstupy, síťový a zálohovaný. Za normálních okolností, tedy pokud je napětí distribuční sítě v mezích určených v připojovacích podmínkách distribuční soustavy jsou oba vývody interně propojeny a fungují jako síťové měniče. Při výpadku distribuční sítě se síťový výstup odpojí od distribuční sítě, měnič se přepne do režimu EPS a napájí elektrická zařízení připojená na zálohovaný výstup.[1]

Dalším možným dělením solárních měničů je podle výstupu obvodu DC/AC měniče.

- **Transformátorové měniče:** Jejich hlavní výhodou je galvanické oddělení od distribuční sítě. Nevýhody pak tvoří vyšší cena a hmotnost měničů.
- **Beztransformátorové měniče:** Tyto měniče nejsou galvanicky odděleny od distribuční sítě. Zpravidla je nutné výstup těchto měničů chránit proudovým chráničem typu B.

Měniče také dělíme podle střídavého výstupu.

- **Jednofázové měniče:** Tento typ má pouze jednofázový výstup a je tedy vhodný zejména pro malé rezidenční fotovoltaické elektrárny. Některé měniče umožňují vytvoření třífázového výstupu ze 3 jednofázových měničů, které spolu komunikují a vytváření výstupní napětí se správným fázovým posuvem oproti

ostatním měničům.

- **Třífázové měniče:** Třífázový výstup poskytuje vetšinou vetší výstupní výkon. Tyto měniče se dále dělí na symetrické, které dodávají vždy stejný výkon na všechny fáze a asymetrické, které mohou dodávat rozdílný výkon na jednotlivé fáze. [1]

1.1.2 Společnost SOFARSOLAR

Společnost SOFARSOLAR, která byla založena v roce 2012, představuje dceřinou firmu renomované skupiny SOFAR. Tato společnost se řadí mezi pět největších výrobců střídačů v Číně a zaměřuje se na komplexní oblasti výzkumu, vývoje, výroby, distribuce a poskytování servisu v oblasti širokého spektra střídačů. To zahrnuje střídače navržené pro obytné a komerční budovy s výkonem od 1 kW až do 255 kW, hybridní střídače s rozsahem od 3 kW do 20 kW a inovativní akumulační baterie AMASS. [2]

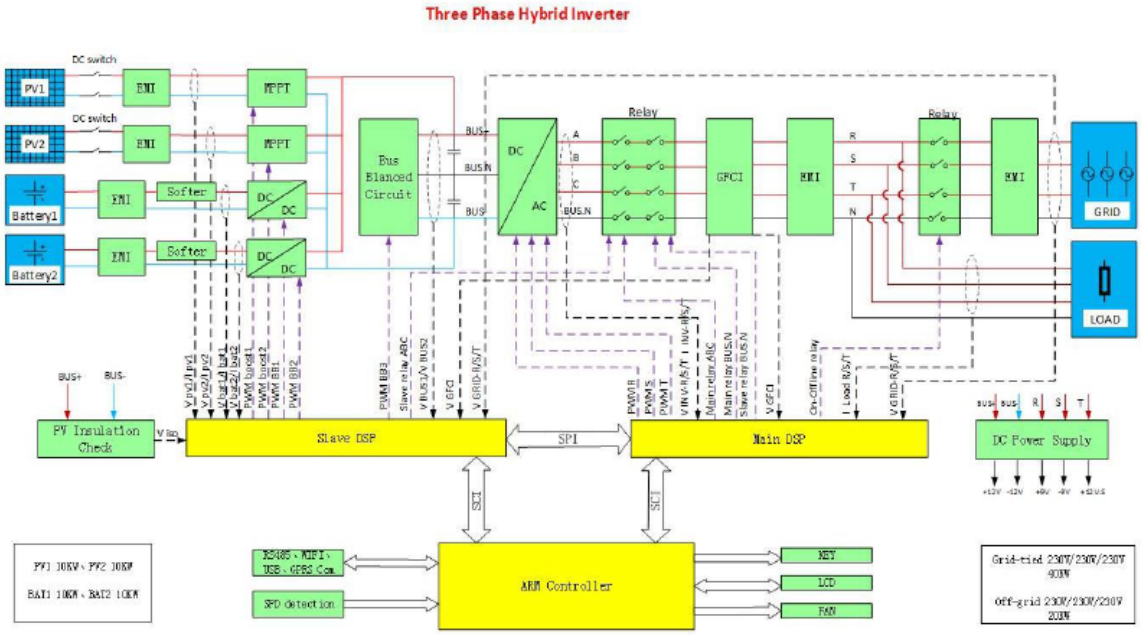
Evropská centrála SofarSolar GmbH se nachází v německém mestě Reutlingen. Tato centrála se stará o rozvoj trhů v regionu EMEA (Evropa, Střední východ a Afrika) a zajišťuje marketing a servis výrobků SofarSolar.[2]

1.1.3 Střídač HYD 10-KTL-3PH

Jedním ze stěžejních bodů této práce je zjistit, jakým způsobem lze komunikovat se střídačem HYD 10-KTL-3PH.

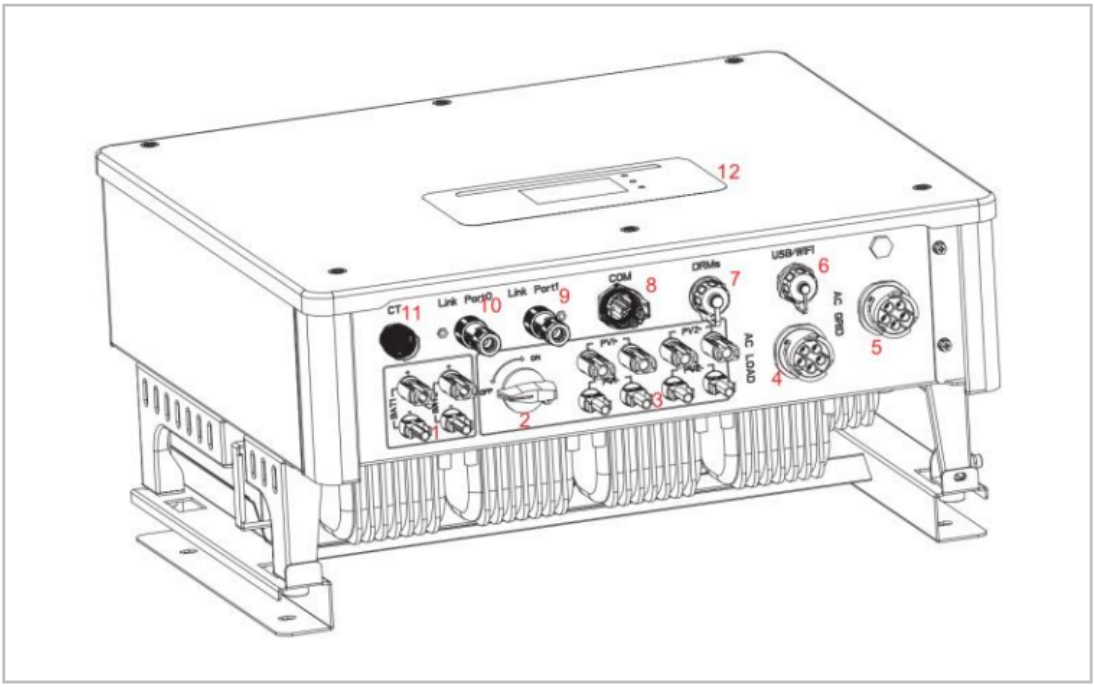
Jedná se o hybridní asymetrický třífázový střídač o výkonu 10 kW. Tento střídač je vhodný zejména pro rezidenční instalace na rodinné domy. Blokové schéma střídače je nastíněno na následujícím obrázku. Ze schématu lze vidět následující: Střídač obsahuje dva MPPT regulátory, na každý z nich je možné připojit jeden řetězec panelů. Ve stejnosměrné části se také nachází dva DC/DC měniče. Ke střídači je tedy možné připojit až dvě vysokonapěťové baterie. Výstupy všech těchto stejnosměrných prvků jsou připojeny na DC sběrnici, která je opatřeny vyrovnávacím obvodem. DC sběrnice je připojena na AC/DC měnič (střídač), který přeměňuje stejnosměrné napětí na střídavé napětí s parametry síťového napětí. Výstup střídače je připojen na sérii dvou stykačů z nichž každý je ovládán jedním signálovým procesorem. Za stykači následuje proudový chránič, který reaguje i na stejnosměrné proudy. Za proudovým chráničem je výstup střídače rozdělen na dvě větve. Ongrid větev je připojena přes stykač který je sepnut pouze v případě, kdy je přítomno napětí ze sítě. Větev LOAD je napájena trvale tzn. je funkční i při výpadku distribuční sítě. Na tento výstup se připojují spotřebiče, jejichž funkčnost je třeba zachovat i při výpadku distribuční sítě. Obě tyto větve disponují výstupními filtry.

O řízení všech těchto výkonových prvků a obvodů se starají dva signálové procesory. Master DSP a Slave DSP mezi sebou komunikují pomocí SPI sběrnice Nad těmito signálovými procesory stojí hlavní ARM procesor, který komunikuje se signálovými procesory pomocí standardního sériového rozhraní. Cílem bude tedy komunikovat s tímto ARM procesorem. Na blokovém schématu jsou naznačeny rozhraní ke kterým je tento procesor připojen. Jmenovitě to je RS485 sběrnice, WIFI, USB, GPRS. Z těchto rozhraní je z hlediska spolehlivosti, robustnosti a odolnosti proti rušení nejvhodnější sběrnice RS485. Uvedené blokové schéma neobsahuje všechny podstatné vstupy a výstupy. Z tohoto důvodu je na obrázku Obr. 1.3 zobrazeno, jakými konektory je střídač obdařen.



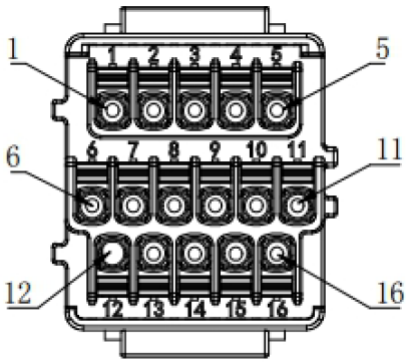
Obr. 1.2: Blokové schéma střídače HYD 10-KTL-3PH [4]

Na obrázku 1.3 lze vidět veškeré konektory střídače HYD-10-KTL-3PH. Konektory označené číslem 1 jsou vstupy pro baterie. Pod číslem 2 se nachází dc vypínač určený k odpojení solárních panelů. Číslo 3 označuje konektory pro připojení solárních panelů. Nachází se zde 4 páry, přičemž každé dva páry jsou paralelně spojeny a přivedeny na jeden MPPT regulátor. Konektor 4 (AC LOAD) slouží pro přípojení zálohovaných spotřebičů. Konektor číslo 5 je síťový výstup střídače (AC GRID). Konektor číslo 6 je USB konektor pro připojení wifi logeru případně pro připojení flash disku pro aktualizaci firmwaru. Pod číslem 7 se nachází konektor RJ45, který



Obr. 1.3: Ukázka konektorů střídače HYD-10- KTL-3PH [4]

je určen pro řízení výkonu střídače. Konečně konektor číslo 8 je komunikační konektor. Tento konektor bude následně podrobně rozebrán. Pro úplnost konektory 9 a 10 slouží pro komunikaci s dalšími střídači při paralelním provozu více střídačů. Konektor 11 je určen pro připojení měřících transformátorů proudu na přívodu DS do objektu. Na obrázku 1.4. je zobrazen multifunkční komunikační konektor. Z tohoto obrázku lze vyčíst kolika svorkami je konektor opatřen a jak jsou očíslovány. Funkce jednotlivých svorek je popsána v následující tabulce Tab 1.1. Tento konektor je pro účely této práce nejpodstatnější. Z tabulky Tab 1.1. lze vyčíst, že multifunkční komunikační konektor obsahuje 2 komunikační sběrnice RS485, přičemž první z nich je v konektoru vyvedena 2x. Tato sběrnice je určena pro komunikaci mezi střídači při kaskádovém zapojení více střídačů nebo je možný tuto sběrnice využít pro monitoring a řízení střídače. Na tuto sběrnici bude tedy připojena modulová elektronika vytvořená na základě zadání této práce. Druhá sběrnice RS485 je vyhrazena pro komunikaci s elektroměrem umístěném na patě objektu. Na pinech 7 až 11 jsou sběrnice pro komunikace s BMS litiových baterií. Konkrétně tedy sběrnice CAN a opět sběrnice RS485. Tato sběrnice je střídačem automaticky identifikována. Na piny 12 a 13 je možné připojit teplotní čidlo pro baterie, které nekomunikují se střídačem. Poslední svorky jsou určeny ke spínaní signálu 12 V. Signál lze použít přímo ze střídače na svorce 16, případně spínat externí signál pomocí pinů 14 a 15.



Obr. 1.4: Multifunkční komunikační konektor[4]

Pin	Definice	Funkce	Poznámka
1	RS-485A1-1	RS-485 differenciální signál +	Monitorování a řízení střídače, propojení monitoringu při kaskádovém zapojení více střídačů
2	RS-485A1-2	RS-485 differenciální signál +	
3	RS-485B1-1	RS-485 differenciální signál -	
4	RS-485B1-2	RS-485 differenciální signál -	
5	RS-485A2	RS-485 differenciální signál +	Komunikace s elektroměry
6	RS485B2	RS485 differenciální signál -	
7	CAN0H	CAN high data	Komunikace s BMS lithiovými bateriemi, střídač automaticky identifikuje komunikační sběrnici baterie (CAN nebo RS485)
8	CAN0L	Can low data	
9	GND.S	zem pro komunikaci s BMS	
10	485TX0+	RS-485 differenciální signál +	
11	485TX0-	RS-485 differenciální signál -	
12	GND.S	Signálová zem	Vnitřní teplota baterie
13	BATTemp	Teplota baterie	
14	DCT1	Beznapěťový kontakt1	Elektronická spínací funkce
15	DCT2	Beznapěťový kontakt2	
16	VCC12V	VCC12V/0,5 A	Ovládací signál 12 V

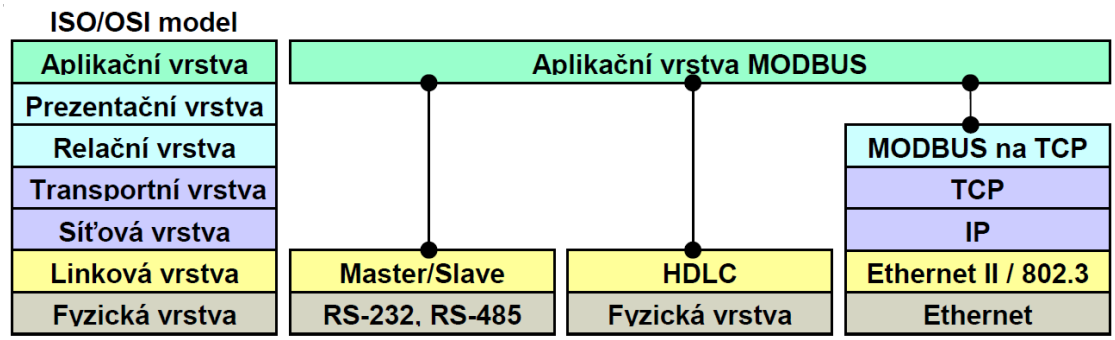
Tab. 1.1: Popis komunikačního konektoru střídače SOFAR HYD-10-KTL [4]

1.2 Komunikační protokol MODBUS

Pro komunikaci se střídačem HYD-10-KTL je nutné využít protokol Modbus RTU. Tato kapitola popisuje samotný komunikační protokol MODBUS.

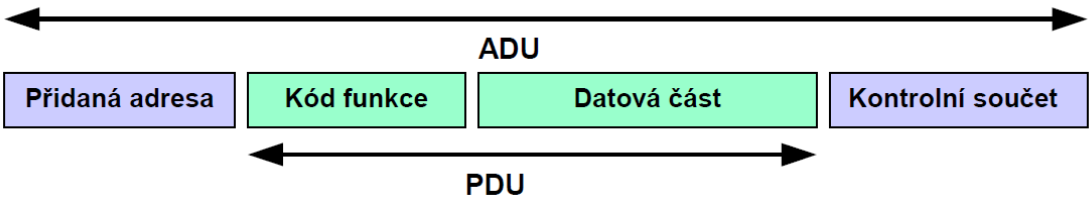
1.2.1 Popis protokolu

MODBUS je komunikační protokol pracující na aplikační (7.) vrstvě ISO/OSI modelu, který umožňuje komunikaci typu klient server mezi zařízení připojenými pomocí různých typů komunikačních sběrnic nebo sítích. V roce 1979 se stal standardem pro komunikaci v průmyslu. V současné době podporuje celou řadu komunikačních medií např. sériové linky RS-232, RS-422 a RS-485, optické a rádiové sítě nebo síť Ethernet. Pro přístup pomocí sítě Ethernet je rezervován port 502 v protokolu TCP/IP. Komunikace probíhá způsobem požadavek-odpověď a požadovaná funkce je určena pomocí kódu funkce, který je součástí PDU. [6]

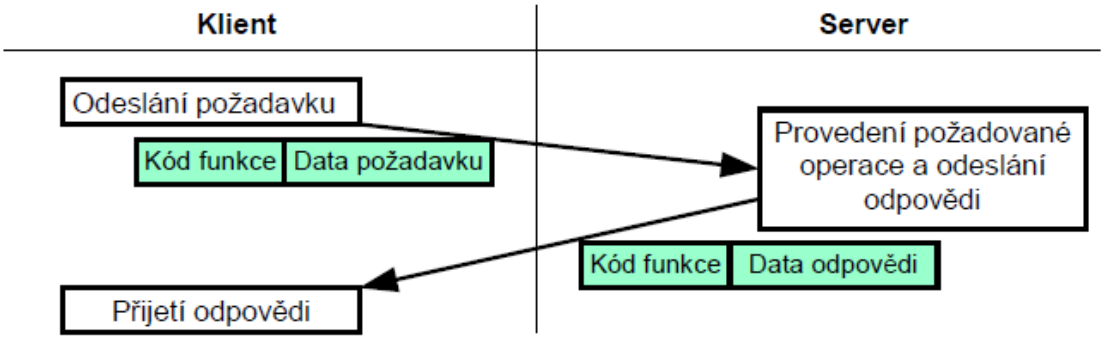


Obr. 1.5: Příklady implementace MODBUS protokolu [5]

Protokol MODBUS definuje základní strukturu zprávy jako Protocol Data Unit (PDU). PDU je nezávislá na typu komunikační vrstvy. Na základě typu použité sítě je PDU rozšířena o další části, čímž tvoří zprávu na aplikační úrovni (ADU). [6]



Obr. 1.6: Základní tvar MODBUS zprávy [5]



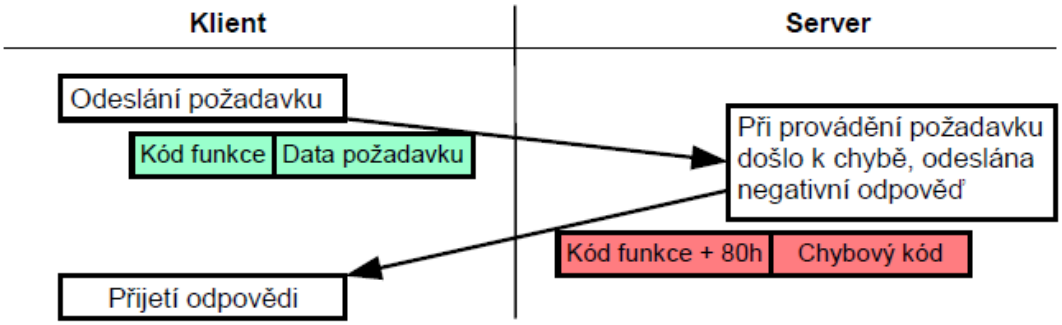
Obr. 1.7: Modbus transakce [5]

Application data unit vytváří klient, který inicializuje MODBUS transakci. Kód funkce říká serveru jakou operaci má provést. Formát žádosti určuje protokol MODBUS. Kód žádosti má velikost jeden bajt a nabývá hodnot 1 až 255, přičemž kódy 128 až 255 jsou využívány pro oznámení chybových stavů. K některým kódům funkce jsou přidány kódy podfunkcí pro definici více akcí. Datová část zprávy obsahuje další informace, které server používá při vykonávání operace definované kódem funkce. Tato část zahrnuje zejména adresy a množství registrů případně vstupů. U některých typů operací nemusí být tyto dodatečné informace potřebné a proto může datová část ve zprávě chybět. [6]

MODBUS protokol definuje 3 typy zpráv PDU:

- Požadavek = (kód funkce, data požadavku)
  - kód funkce určující operaci která se má vykonat - jeden bajt
  - data požadavku určená k vykonání operaci dle funkčního kódu jako například adresa, proměnné, počet proměnných, kód podfunkce, offset atd. - 0 až 253 bajtů
- Odpověď = (kód funkce, data odpovědi)
  - kopie kódu funkce z požadavku - jeden bajt
  - data odpovědi například přečtené vstupy nebo registry atd. - 0 až 253 bajtů
- Záporná odpověď = (kód funkce + 80h, data odpovědi)
  - kopie kódu funkce z požadavku sečtená s 80h (indikace neúspěchu) - jeden bajt
  - chybový kód (Exception code) dle definovaných chybových kódů.

V případě, že nedojde při provádění operace k žádné chybě, server odpoví zprávou, která obsahuje kód funkce stejný jako v žádosti od klienta. Tím je indikováno úspěšné vykonání požadované operace. V datové části jsou serverem předána požadovaná data, pokud nějaká jsou. ukázka provedení MODBUS transakce bez chyby je znázorněna na obrázku 1.7. [6]



Obr. 1.8: Modbus transakce s chybou [5]

V případě, kdy při zpracování instrukce dojde k chybě je vrácen kód funkce s nastaveným nejvyšším bitem, což indikuje chybu. V datové části je vrácen chybový kód, který upřesňuje důvod neúspěchu. Ukázka průběhu této transakce je na obrázku 1.8. [6]

Jelikož je možná ztráta odpovědi je vhodné nastavit časový limit pro její přijetí. [6]

Maximální velikost PDU je odvozena od první aplikace MODBUS protokolu na sériové lince RS-485, kde byla maximální délka ADU 256 bajtů. Z čehož vyplývá maximální délka PDU 253 bajtů pro sériovou linku. [6]

Velikost ADU na RS-485 = 253 bajtů PDU + 1 bajt pro adresu + 2 bajty CRC = 256 bajtů. Velikost ADU na TCP/IP = 253 bajtů PDU + 7 bajtů MBAM = 260 bajtů.[6]

1.2.2 Způsob kódování

MODBUS využívá Big-endian reprezentaci dat. Tedy při posílání datových bloků větších než jeden bajt je nejprve poslán nejvyšší bajt, zatímco nejnižší bajt je poslán jako poslední. [6]

Například při zaslání 16 bitového registru 0x12AB bude jako první zaslán bajt 0x12 a jako druhý bajt 0xAB.

1.2.3 Datový model MODBUS

MODBUSem definovaný datový model je rozdělen do čtyř charakteristických tabulek. Seznam těchto datových tabulek se nachází v tabulce 1.2. Dle protokolu může mít každá z tabulek až 65536 hodnot. Kvůli zpětné kompatibilitě bývá adresní velikost tabulek omezena na 10000 hodnot. Takto definované tabulky nijak nesouvisí s konkrétní aplikací. Mapovaní těchto tabulek je závislé na konkrétním zařízení. Může se stát, že se tyto tabulky překrývají kvůli nedostatku paměti . [6]

Název tabulky	Velikost položky	Přístup	Adresa
Cívky	1 bit	Čtení/zápis	0 - 9999
Diskrétní vstupy	1 bit	Pouze čtení	10000 - 19999
Vstupní registry	16 bitů	Pouze čtení	30000 - 39999
Uchovávací registry	16 bitů	Čtení/zápis	40000 - 49999

Tab. 1.2: Datový model MODBUS [6]

1.2.4 Zpracování MODBUS požadavku

Po přijetí požadavku od klienta dochází k jeho zpracování na straně serveru. Obecný postup zpracování MODBUS požadavku je znázorněn na následujícím stavovém diagramu (Obr 1.9.). Po přijetí požadavku server otestuje zda je obsažen platný kód funkce. Následně ověří, jestli adresa dat odpovídá rozsahu. A nakonec ověří, zda mají data platnou hodnotu. V případě, že je vše v pořádku pokračuje ke zpracování požadavku. V opačném případě, v závislosti na typu chyby vygeneruje chybový kód který přičte ke kódu funkce. Pokud zpracování požadavku proběhlo úspěšně, odešle se odpověď klientovi. Pokud ne vygeneruje se chybový kód odpovídající chyby a odešle se záporná odpověď. [6]

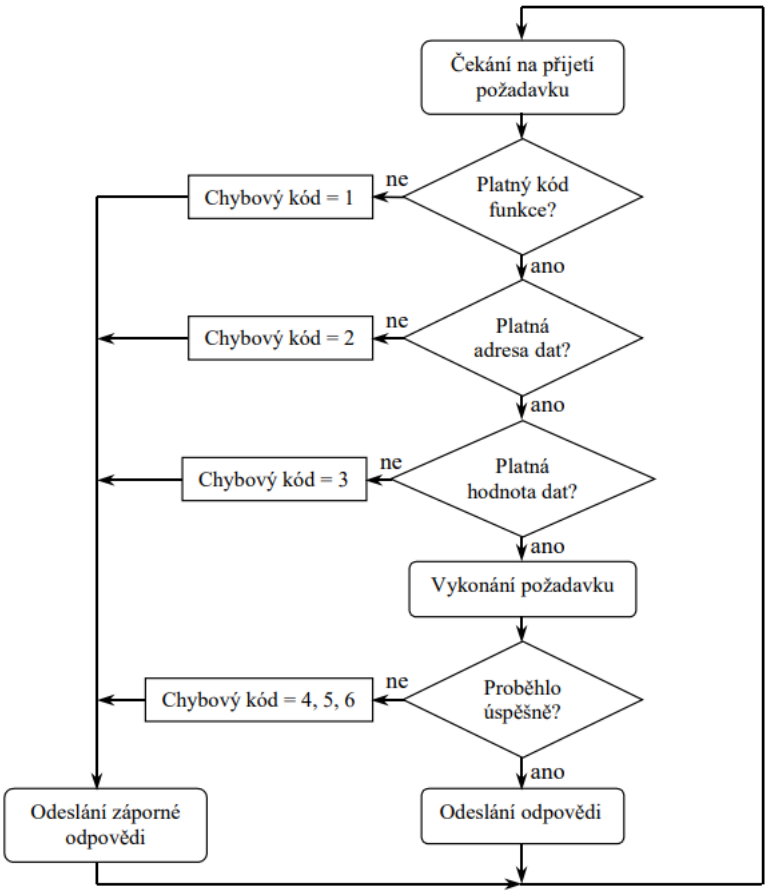
1.2.5 Popis vybraných funkcí MODBUS

Čti cívky (Read coils) - 0x01 - slouží ke čtení stavu 1 až 2000 cívek. V požadavku je specifikována adresa a početek cívek. V odpovědi je nejnižší bit informace o stavu první cívky. [6]

- Požadavek
  - Kód funkce: 0x01
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet cívek: 1 - 2000 (0x7D0)
- Odpověď
  - Kód funkce: 0x01
  - Počet bajtů (počet cívek/8)
  - Stavy cívek

Čti diskrétní vstupy (Read discrete inputs) - 0x02 - slouží ke čtení stavu 1 až 2000 diskrétních vstupů. V požadavku je specifikována adresa prvního vstupu a počet vstupů. V odpovědi je nejnižší bit informace o stavu prvního vstupu. [6]

- Požadavek
  - Kód funkce: 0x02
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet vstupů: 1 - 2000 (0x7D0)



Obr. 1.9: Zpracování požadavku na straně serveru [5]

- Odpověď
  - Kód funkce: 0x02
  - Počet bajtů (počet vstupů/8)
  - Stavy vstupů

Čti uchovávací registry (Read holding registers) - 0x03 - slouží ke čtení hodnot až 125 registrů v souvislém bloku. V požadavku je specifikována adresa prvního registru a počet registrů. V odpovědi jsou pro každý registr 2 bajty prezentující hodnoty jednotlivých registrů. [6]

- Požadavek
  - Kód funkce: 0x03
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet vstupů: 1 - 125 (0x7D)
- Odpověď
  - Kód funkce: 0x03
  - Počet bajtů (2x počet registrů)

  - Stavy registrů

Čti vstupní registry (Read holding registers) - 0x04 - slouží ke čtení hodnot až 125 vstupních registrů v souvislém bloku. V požadavku je specifikována adresa prvního registru a počet registrů. V odpovědi jsou pro každý registr 2 bajty prezentující hodnoty jednotlivých registrů. [6]

- Požadavek
  - Kód funkce: 0x04
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet vstupů: 1 - 125 (0x7D)
- Odpověď
  - Kód funkce: 0x04
  - Počet bajtů (2x počet registrů)
  - Stavy registrů

Zapiš jednu cívku (Write single coil) - 0x05 - slouží k nastavení jednoho výstupu na logickou 1 nebo logickou 0. V požadavku je specifikována adresa výstupu a hodnota na kterou se má nastavit. Pro logickou 0 je zaslána hodnota 0x0000 a pro logickou 1 je zaslána hodnota 0xFF00. Pokud je hodnota úspěšně zapsána odpověď je kopie požadavku. [6]

- Požadavek
  - Kód funkce: 0x05
  - Adresa výstupu: 0x0000 - 0xFFFF
  - Hodnota výstupu: 0x0000 nebo 0xFF00
- Odpověď
  - Kód funkce: 0x05
  - Adresa výstupu: 0x0000 - 0xFFFF
  - Hodnota výstupu: 0x0000 nebo 0xFF00

Zapiš jeden registr (Write single register) 0x06 - slouží k zápisu dat do jednoho uchovávacího registru. Požadavek specifikuje adresu registru a hodnotu která se má zapsat. [6]

- Požadavek
  - Kód funkce: 0x06
  - Adresa registru: 0x0000 až 0xFFFF
  - Hodnota registru: 0x0000 až 0xFFFF
- Odpověď
  - Kód funkce: 0x06
  - Adresa výstupu: 0x0000 - 0xFFFF
  - Hodnota výstupu: 0x0000 nebo 0xFF00

Zapiš více registrů (Write multiple registers) 0x10 - slouží k zápisu dat do až 120 uchovávacích registrů). Požadavek specifikuje adresu registru počet registrů

a hodnoty které se mají zapsat. [6]

- Požadavek
  - Kód funkce: 0x10
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet registrů: 1 až 120 (0x78)
  - Počet bytů: počet registrů x 2
  - Hodnoty registrů:počet registrů x (0x0000 až 0xFFF)
- Odpověď
  - Kód funkce: 0x10
  - Počáteční adresa: 0x0000 - 0xFFFF
  - Počet registrů: 1 až 120 (0x78)

1.3 Sběrnice RS-485

Z první kapitoly vyplývá, že se střídačem je možné komunikovat po sběrnici RS-485. V této kapitole je tedy podrobně teoreticky rozebrán tento komunikační standard, pro snadnou následnou implementaci.

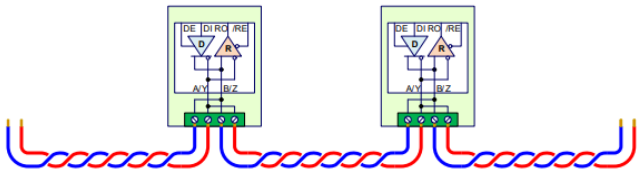
Sběrnice RS-485 se používá především v průmyslovém prostředí a využívá jí mnoho komunikačních protokolů v průmyslu, jako například PROFIBUS, MODBUS atd. Tento standard byl definován v roce 1983 a definuje pouze elektrické vlastnosti pro rozhraní přijímačů a vysílačů. Jedná se o sériovou komunikaci a vychází z jejího předchůdce RS-232. Hlavním cílem RS-485 je poskytnout spolehlivou a odolnou komunikaci mezi průmyslovými zařízeními. [7]

1.3.1 Základní vlastnosti RS-485

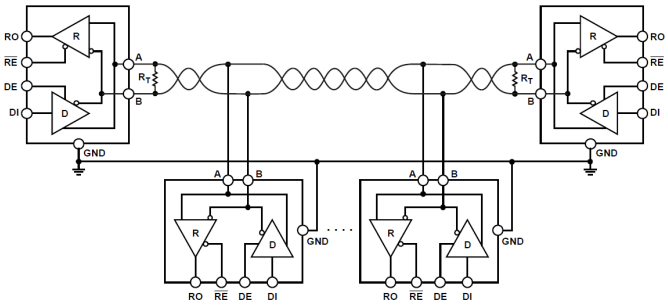
- Až 32 zařízení na sběrnici
- rozsah napětí -7 až +12 V
- Vícebodové operace s jedním 5V napájecím zdrojem
- Maximální rychlost až 10 Mb/s (při vzdálenosti maximálně 12 cm)
- Maximální délka vedení 1200 m (pro rychlosti do 100 kb/s) [8]

1.3.2 Topologie sítě

RS-485 podporuje několik topologií. Jako například propojení bod-bod (point-to-point) pokud jsou připojena pouze 2 zařízení. Pro více zařízení je možné zapojení se společnou sběrnicí, případně řetězové zapojení. Výběr typu zapojení záleží zejména na aplikaci. Pokud je na sběrnici připojeno více zařízení je nutné zabezpečit, aby



Obr. 1.10: RS-485 Zapojeni do řetězce [8]



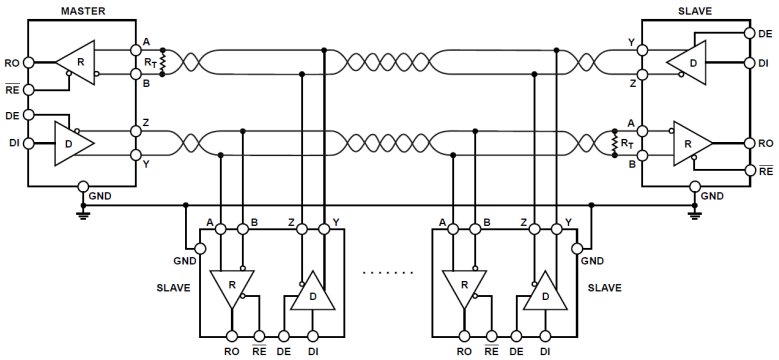
Obr. 1.11: Half Duplex RS-485 [10]

každé zařízení mělo unikátní adresu. Ukázka zapojení do řetězce je na obrázku 1.10. [8]

Sběrnice RS-485 je navržena jako half-duplex (k přenosu dat dochází v jednom časovém okamžiku pouze jedním směrem). Zdvojením signálových vodičů lze docílit full-duplex přenosu, tedy v jednom čase může master a slave zároveň přijímat i vysílat. Ukázka možného zapojení half-duplex je na obrázku 1.11. Toto zapojení je také známo pod pojmem dvouvodičové zapojení, na obrázku je však naznačeno propojení zemí zařízení. Toto propojení je důležité vytvořit z důvodu možných rozdílů v potenciálu země u komunikujících zařízení a to zejména při větších vzdálenostech mezi zařízeními. Po přidání zemnícího vodiče je tedy zapojení reálně třívodičové. Na obrázku 1.12. je znázorněno zapojení pro full-duplex přenos známé také jako čtyřvodičové zapojení, se zemnícím vodičem tedy pětivodičové. [10]

1.3.3 Signálové úrovně

Logické stavy jsou reprezentovány rozdílovým napětím mezi oběma vodiči. Detekce logického stavu založená na rozdílovém napětí mezi oběma vodiči je výhodná zejména kvůli eliminaci indukovaného rušivého signálu, který se přičítá k oběma vodičům stejně. Přijímač rozlišuje logický stav 1 při rozdílu napětí A B $<$ -200 mV. Logický stav 0 při rozdílu napětí A - B $>$ +200 mV. Vysílač by měl na výstupu při logické 1 (klidový stav linky) generovat na vodiči A napětí alespoň 1,5 V, na vodiči B +1,5 V, při logické 0 by měl na vodiči A generovat +1,5 V, na vodiči B -1,5 V. [8]



Obr. 1.12: Half Duplex RS-485 [10]

1.3.4 Typ kabelu

Pro udržení výhody RS-485 komunikace, tedy detekce stavu z rozdílového napětí, je nutné zvolit vhodný typ kabelu. Jako komunikační kabel je možné použít kabely s krouceným párem. U těchto kabelů je velmi pravděpodobné, že se rušení přičte k oběma signálům stejně. Kabel by měl mít impedanci 120 ohmů. Kabely s krouceným párem se vyrábí s různým počtem párů a se stíněním jak celého kabelu tak i jednotlivých vodičů. Výběr vhodného typu kabelu s krouceným párem závisí na aplikaci a prostředí kde bude umístěn. [9]

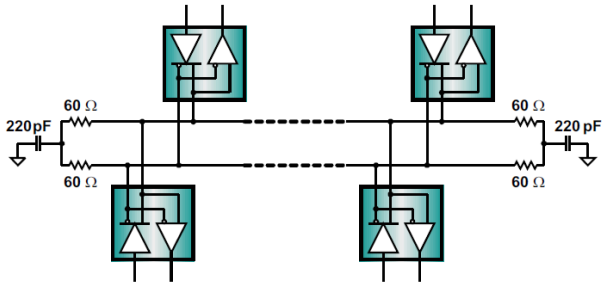
1.3.5 Terminace

Aby nedocházelo k odrazům signálu měly by se konce komunikační linky zakončit terminačním odporem. Tyto odpory by měly mít hodnotu 120 ohmu a umístěny na každém konci komunikační linky. Pokud je sběrnice v prostředí s vysokým výskytem šumu je možné 120 ohmový rezistor nahradit dvěma 60 ohmovými dolnopropustnými filtry, které zajistí dostatečnou filtraci šumu. Tyto rezistory by také měly být co nejpřesnější, nejlépe s 1% tolerancí hodnoty. A to z důvodu aby byla zajištěna podobná hodnota mezní frekvence filtrů. Pokud by se mezní frekvence výrazně lišila, tak se bude společný šum převádět na diferenciální šum. Ukázka terminace s dvěma dolnopropustnými filtry je na obrázku 1.13. [9]

1.4 Požadavky na zařízení

Na základě práce a poznatků z předchozích kapitol lze v této kapitole zformulovat základní požadavky na vyvíjené zařízení. Pro vzdálený dohled solárního střídače jsou stěžejní následující funkce a parametry zařízení:

1. Zařízení musí mít schopnost připojení Wi-iI



Obr. 1.13: Terminace RS-485 [9]

2. Implementace komunikace pomocí protokolu modbus RTU přes sběrnici RS-485
3. Zařízení bude opatřeno dostatečnou pamětí pro logování dat
4. Pro ovládání digitálních výstupů budou vytvořeny 2 galvanicky oddělené výstupy pro relátka/stykače
5. Analogové výstupy budou v průmyslovém standardu rozsahu napětí 0 až 10 V
6. Zařízení bude opatřeno displejem pro zobrazovaní parametrů a úpravu nastavení
7. Pro ovládání budou použita tlačítka
8. Zařízení bude mít uchycení na DIN lištu a bude v rozměrech modulových přístrojů
9. Pro programovaní a případné aktualizace softwaru bude použit USB konektor
10. Napájení zařízení bude 12 V DC
11. Zařízení musí být odolné proti rušení, přepětí, přepólování a nadproudu

1.5 Požadavky na mikrokontrolér

Z předchozí kapitoly vyplývají následující parametry mikrokontroléru, který bude použit pro vzdálený solární dohled.

- Sériová komunikace UART, která bude pro komunikaci se střídačem převedena na RS-485
- Možnost připojení Wi-Fi a vytvoření WEB serveru
- Sběrnice SPI pro komunikaci s externí flash paměti
- Sběrnice I2C pro komunikaci s displejem
- Minimálně 4 digitální výstupy
- Minimálně 4 digitální vstupy
- Dostatečný výpočetní výkon a rychlost pro bezproblémovou obsluhu všech periferii, komunikaci a obsluhu webserveru

Obr. 1.14: RTL8710 [11]

- Přijatelná cena

1.6 Průzkum trhu dostupných mikrokontrolérů

Z předchozí kapitoly, která sumarizuje základní požadavky na mikrokontrolér vyplývá, že hlavním kritériem pro výběr bude schopnost mikrokontroléru komunikovat pomocí Wi-Fi, dostatečný výkon a cena mikrokontroléru. Ostatní požadované parametry zpravidla splňuji běžné mikrokontroléry. V následujících podkapitolách budou popsány nalezené mikrokontroléry, jakožto výsledek provedeného průzkumu trhu.

1.6.1 RTL8710

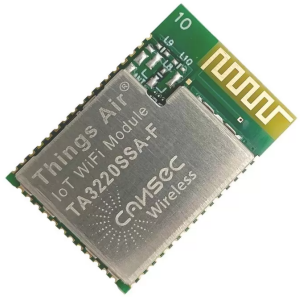
Prvním nalezeným možným mikrokontrolérem je RTL8710 od společnosti Realtek. Jádrem tohoto mikrokontroléru je ARM Cortex-M3 Processor běžící na frekvenci 166MHz. [11]V tabulce 1.3 jsou uvedeny nejpodstatnější parametry. Ukázka modulu RTL8710 je na obrázku 1.14.

1.6.2 CC3220

Další variantou mikrokontroléru je CC3220 SimpleLink od společnosti Texas Instrument. Ukázka tohoto mikrokontroléru je na obrázku 1.15. Jedná se o jediný čip, který zahrnuje dvě oddělené prostředí. Obsahuje mikrokontrolér založen na procesoru ARM Cortex M4 běžící na frekvenci 80 Mhz a síťový procesor pro provoz všech logických vrstev Wi-Fi a internetu.[12] Důležité parametry tohoto čipu jsou obsaženy v tabulce 1.4.

Kategorie	Funkce	Specifikace
Wireles	standard	802.11 b/g/n
	frekvence	2.4 GHz
Hardware	CPU	ARM Cortex M3 32 Bit 166 MHz
	ROM/RAM/FLASH	1MB/512KB/1MB
	SPI	až 2
	I2C	3
	GPIO	17
	Pracovní napětí	3.3 V (3.0 - 3.6)
	Pracovní teplota	-20 až +85 °C
	Wireles network model	ST/AP
	Cena	120 Kč

Tab. 1.3: Parametry mikrokontroléru RTL8710 [11]



Obr. 1.15: CC3220 [13]

1.6.3 ESP8266

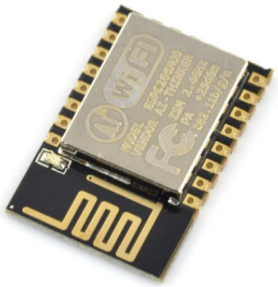
ESP8266 od společnosti Espressif poskytuje integrované řešení Wi-Fi SoC (System on Chip) splňující požadavky uživatelů v oblasti internetu věcí, jako jsou například: spolehlivý výkon kompaktní design a nízká spotřeba energie. S plnou a samostatnou schopností Wi-Fi sítě může ESP8266 fungovat buď jako samostatná aplikace, nebo jako podřízená jednotka hostitelskému MCU. Vyrábí se také již jako hotový modul s anténou. [15] Podstatné parametry modulu ESP8266 jsou v tabulce 1.5. Ukázka tohoto modulu je na obrázku 1.16.

1.6.4 ESP32

Na obrázku 1.17. je modul ESP32 jedná se o novější verzi Wi-Fi SoC avizovaného ESP8266. Zásadní rozdíl je v použitém 32 bitového procesoru Tenselica LX6,

Kategorie	Funkce	Specifikace
Wireles	standard	802.11 b/g/n
	frekvence	2.4 GHz
Hardware	CPU	ARM Cortex M4 32 bit 80 MHz
	ROM/RAM/FLASH	1MB/256KB/1MB
	SPI	1
	I2C	1
	GPIO	až 27
	Pracovní napětí	3.3 V (napájení - 2.1 - 3.6)
	Pracovní teplota	-40 až +85 °C
	Wireles network model	ST/AP
	Cena	200 Kč

Tab. 1.4: Parametry mikrokontroléru CC3220 [12]



Obr. 1.16: ESP8266[14]

který je dvoujádrový oproti jednojádrovému procesoru Tenselica L106 používaném v ESP8266. ESP32 také oproti ESP8266 podporuje Wi-Fi na pásmu 5 Ghz. Dále disponuje nižší spotřebou. Nevýhodou je vyšší cena oproti ESP8266.[17] Podstatné parametry modulu ESP8266 jsou v tabulce 1.6.

1.7 Programování mikrokontroléru ESP32

Pro programovaní mikrokontroléru ESP32 je možné využít několik cest.

Pro vývoj software ESP32 je možné využít následující dva frameworky. Arduino ESP32 frmaework a ESP-IDF framework.

Nejjednodušší cestou je vývoj firmware ve vývojovém prostředí Arduino IDE. Pro vývoj software v Arduino IDE stačí přidat mikrokotrolér ESP32 a následně lze využít programovací jazyk wiring a knihovny určené pro esp32 vytvořené v arduino IDE, tedy Arduino framework.

Kategorie	Funkce	Specifikace
Wireles	standard	802.11 b/g/n
	frekvence	2.4 GHz
Hardware	CPU	Tensilica L106 32-bit
	RAM	32KB(instrukce), 80KB(data)
	FLASH	512KB-4MB
	UART	2
	SPI	2
	I2C	1
	GPIO	až 17
	Pracovní napětí	3.3 V (2.5 - 3.6)
	Pracovní teplota	-40 až +125 °C
	Wireles network model	ST/AP
	Cena	80 Kč

Tab. 1.5: Parametry mikrokontroléru ESP8266 [15]

Obr. 1.17: ESP32[16]

Framework ESP-IDF je oficiální IoT vývojový framework vytvořený přímo společností Espessif. ESP-IDF používá programovací jazyk c a c++. Jedná se o open-source framework, který je volně dostupný na GitHub.

ESP-IDF je možné nainstalovat několika způsoby. Doporučeným způsobem je instalace rozšíření ve vývojovém prostředí Eclipse nebo Visual Studio code. Druhou možností je manuální instalace přímo v operačním systému PC.

Firmware pro ESP32 je také možné vyvíjet na platformě PlatformIO. Jedná se o profesionální platformu pro vývoj vestavěných systémů. PlatformIO podporuje jak Arduino32 framework, tak ESP-IDF framework. Také zahrnuje mnoho nástrojů pro nejběžnější vývojové úkoly, jako je ladění, unit test a statická analýza kódu. [18]

Kategorie	Funkce	Specifikace
Wireles	standard	802.11 b/g/n
	frekvence	2.4 GHz
Hardware	CPU	Tensilica LX6 32-bit 240 Mhz
	RAM	520 KB
	FLASH	4 MB - 16 MB
	UART	3
	SPI	3
	I2C	2
	GPIO	až 34
	Pracovní napětí	3.3 V (2.5 - 3.6)
	Pracovní teplota	-40 až +125 °C
	Wireles network model	ST/AP
	Cena	100 Kč

Tab. 1.6: Parametry mikrokontroléru ESP32[17]

1.8 Wi-Fi

Wi-Fi je jedna z nejrozšířenějších technologií pro lokální bezdrátové sítě, které jsou často označované jako WLAN (Wirelles Local Area Network). V ISO/OSI modelu patří Wi-Fi do fyzické (první) vrstvy. Označení Wi-Fi je pouze obchodní název. Standard síťových bezdrátových protokolů nese označení IEEE 802.11.[19]

Technologie Wi-Fi je nejčastěji využívána díky jednoduché instalace, přijatelné ceně, dostatečné přenosové rychlosti (nejnovější verze se přibližují rychlostem fyzickým kabelovým spojům). Dosah signálu může být až 100 m, tím je zajištěno pokrytí celé domácnosti, což je dalším důvodem pro oblíbenost Wi-Fi.[20]

Dosah signálu závisí na využité frekvenci signálu. Nejběžnější Wi-Fi varianty 802.11b/g/n pracují v pásmu 2,4 Ghz. Modernější verze jako 802.11ac využívají pásmo 5 GHz. Aktuálně vyvíjený standard 802.11be využívá frekvenční pásma 2,4 Ghz i 5 GHz a ještě navíc 6 GHz. [20]

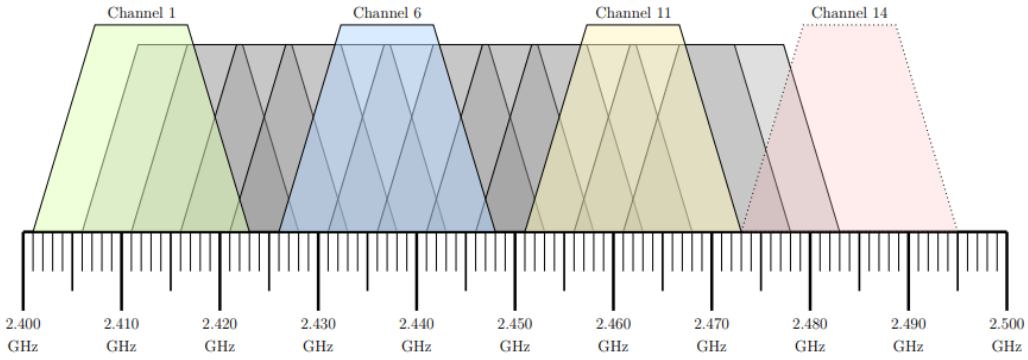
Shrnutí standardů 802.11 včetně jejich frekvenčních pásem a teoretických maximálních přenosových rychlosti zobrazuje přehledová tabulka č. 1.7. [20]

Wi-Fi je ochranná značka spravovaná neziskovou organizací Wi-Fi Alliance, která omezuje používání označení Wi-Fi Certified jen na ty výrobky, které úspěšně splní certifikační testy interoperability.[19]

Z pohledu frekvence, každé spojení nezabírá celé pásmo, ale pouze jeho malou část. Frekvenční pásma jsou rozdělena na několik kanálů. Rozdělením frekvenčního pásma je redukována interference mezi sousedními sítěmi. V pásmu 2,4 GHz je de-

IEEE standard	Rok	Frekvence [GHz]	Max. Přenosová rychlost [Mbit/s]
802.11	1997	2,4	1 - 2
802.11b	1999	2,4	1 - 11
802.11a	1999	5	6 - 54
802.11g	2003	2,4	6 - 54
802.11n	2008	2,4/5	72 - 600
802.11ac	2014	5	433 - 6933
802.11ax	2019	2,4/5	574 - 9608
802.11be	2024	2,4/5/6	1376 - 46120

Tab. 1.7: Porovnání standardů 802.11[20]



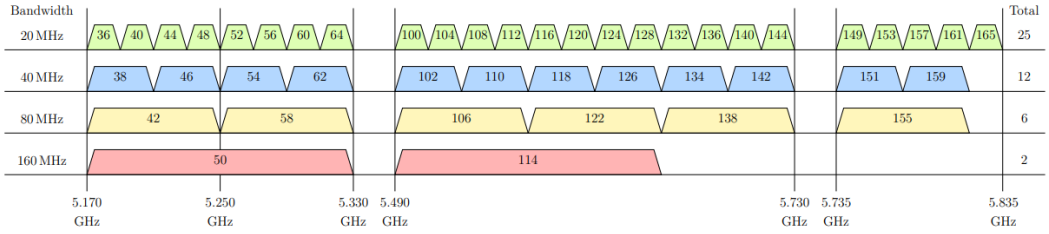
Obr. 1.18: Vizualizace frekvenčních pásem 14 kanálů Wi-Fi v pásmu 2,4 GHz[20]

finováno 14 kanálů pro standardy 802.11b/g/n. Kanál č. 14 se používá pouze v Japonsku pro standard 802.11b. Šířka pásma jednoho kanálu je 20 MHz. Posun mezi středy jednotlivých kanálů je 5 MHz. To znamená, že pásma jednotlivých kanálů se překrývají. Vizualizace frekvenčních pásem kanálů je na obrázku č.1.18. Zvýrazněné kanály ukazují možnost využití bez překrytí frekvenčních pásem.

Pásmo 5 GHz využívané standardy 802.11ac nebo 802.11ax je mnohem širší a sítě využívající toto pásmo mohou použít kanály z různými šířky pásma kanálu. Šířka pásma kanálů v pásmu 5 GHz se pohybuje od 20 MHz do 160 MHz. Zobrazení všech kanálů frekvenčního pásma 5 GHz je na obrázku č. 1.19.

1.8.1 Přístupový bod (Access Point)

Access point (AP) je zařízení, které vytváří bezdrátovou síť Wi-fi. Slouží jako centrální bod ke kterému se mohou připojovat další zařízení (stanice) a komunikovat mezi sebou nebo se připojit k síti internet.



Obr. 1.19: Vizualizace frekvenčních pásem standardu 802.11ac v pásmu 5 GHz[20]

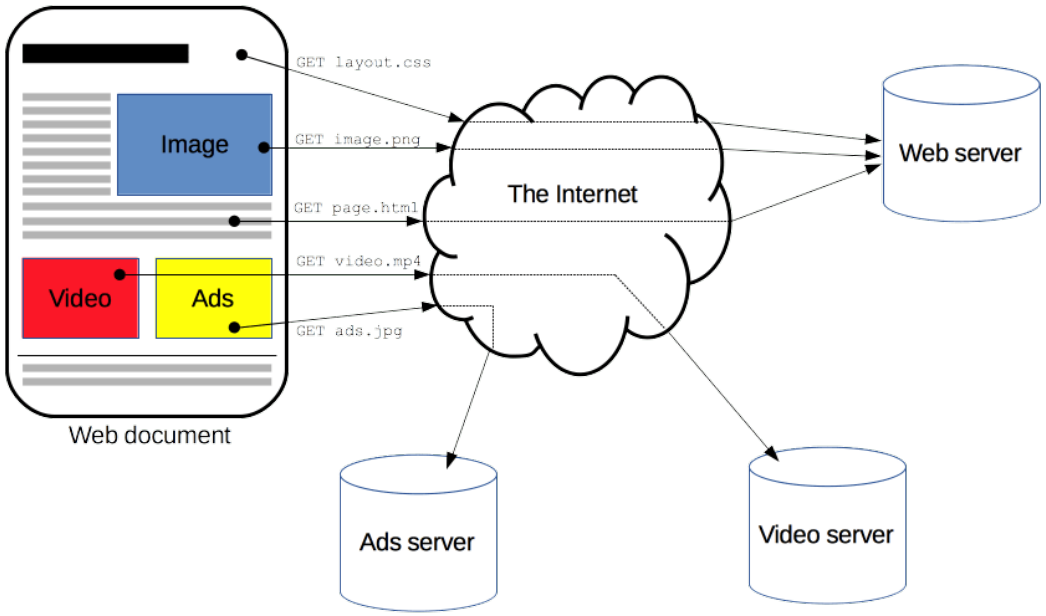
Access point obvykle funguje jako most mezi bezdrátovou sítí a drátovou síťovou infrastrukturou. Každý AP má vlastní identifikátor síťě SSID (Service Set Identifier). [21]

Přístupové body mohou být zabezpečeny různými metodami. Nejběžnější metody jsou popsány v následujícím seznamu.

- **WEP (Wired Equivalent Privacy)** byl jedním z prvních zabezpečovacích mechanismů používaných v bezdrátových sítích. Používá sdílený klíč pro šifrování dat. WEP má však mnoho bezpečnostních nedostatků a dnes se již nedoporučuje používat.
- **WPA (Wi-Fi Protected Access)** byl vyvinut jako náhrada za WEP a poskytuje silnější zabezpečení pomocí TKIP (Temporal Key Integrity Protocol) nebo AES (Advanced Encryption Standard). WPA používá předem sdílený klíč (PSK) nebo autentizaci pomocí EAP (Extensible Authentication Protocol).
- **WPA2 (Wi-Fi Protected Access 2)** WPA2 je další generací protokolu WPA a poskytuje ještě silnější zabezpečení pomocí šifrování AES-CCMP (Advanced Encryption Standard-Counter Mode Cipher Block Chaining Message Authentication Code Protocol). WPA2 je doporučovaný standard pro zabezpečení bezdrátových sítí a je vhodný pro domácí i podnikové nasazení. [21]

1.8.2 Stanice

Station (STA) je jakékoliv zařízení, které se připojuje k Access Pointu (nebo k jiné stanici) a využívá bezdrátovou síť Wi-Fi k přenosu dat. Stanicí může být například laptop, chytrý telefon, tablet nebo jiné zařízení s Wi-Fi funkcionalitou. Stanice může být jak aktivní (přijímání a odesílání dat), tak pasivní (pouze přijímání dat), v závislosti na její funkci v síti. [21]



Obr. 1.20: Znázornění rekonstrukce dokumentu ve webovém prohlížeči pomocí HTTP protokolu[22]

1.9 HTTP (HyperText Transfer Protocol)

HTTP je protokol určený pro načítání zdrojů, jako například dokumentů HTML. Jedná se o protokol typu klient-server. Požadavky jsou iniciovány příjemcem (klientem), což je obvykle webový prohlížeč. HTTP je základem jakékoli výměny dat na webu. Kompletní dokument je rekonstruován z různých načtených dílčích dokumentů, například textu, popisu rozvržení, obrázků, videa, skriptů a dalších. [22]

Protokol HTTP operuje na aplikační (sedmé) vrstvě referenčního modelu ISO/OSI, nebo také na čtvrté vrstvě modelu TCP/IP. HTTP protokol je defaultně přístupný na TCP portu 80, ale existují i varianty s UDP, kde lze číslo portu změnit. Skládá se z metody, cesty k tázanému souboru a verze protokolu HTTP. Poté následuje konec řádku a hlavičky, což jsou další rozšiřující data. Po hlavičkách následují dva konce řádku a samotná data souboru.[23]

1.9.1 Metody HTTP

HTTP protokol využívá několik metod pro komunikaci mezi klientem a serverem. V následujícím výčtu jsou popsány nejpoužívanější metody HTTP protokolu.

GET metoda nejčastěji slouží pro získání dat ze serveru. Její parametry jsou přímo v URL adrese. Z tohoto důvodu není příliš vhodná pro zasílání citlivých údajů, jako jsou například hesla. [24]

POST metoda slouží pro odesílání dat na server. Na rozdíl od metody GET neposílá své parametry v adrese URL. Hodí se tedy pro formuláře a manipulaci s daty. Post metodu je vhodné použít například pro registraci uživatelů, případně pro jejich přihlašování.[24]

PUT metoda se používá pro aktualizaci nebo vytvoření zdroje na serveru podle poskytnutých dat. PUT by mělo být idempotentní, což znamená, že opakované volání s týmiž daty by nemělo měnit stav serveru.[24]

DELETE metoda se používá pro smazání určeného zdroje na serveru.[24]

PATCH metoda Používá se pro částečnou aktualizaci určeného zdroje na serveru. Tato metoda umožňuje aktualizaci pouze určitých částí zdroje, což může být užitečné, pokud není třeba posílat celý objekt zpět na server.[24]

HEAD metoda v HTTP žádá hlavičky, které by byly vráceny, pokud by se URL požadované metodou HEAD místo toho požadovalo metodou GET. Například, pokud by URL mohlo vytvořit stahování velkého souboru, HEAD požadavek by mohl přečíst jeho Content-Length hlavičku, aby zkontroloval velikost souboru, aniž by skutečně stahoval soubor.[24]

1.9.2 Verze HTTP protokolu

HTTP protokol byl vyvinut Timem Berners-Lee a jeho týmem mezi lety 1989 a 1991. Od té doby HTTP prošel mnoha změnami, které pomohly udržet jeho jednoduchost a zároveň formovaly jeho flexibilitu.[25]

První verze byla zpětně označena číslem 0.9. **HTTP/0.9** byl velmi jednoduchý. Požadavky se skládaly z jediného řádku a začínaly jedinou možnou metodou GET, po níž následovala cesta ke zdroji. Úplná adresa URL se neuváděla, protože protokol, server a port nebyly po připojení k serveru nutné.[25]

Odpověď byla také velmi jednoduchá. Obsahovala pouze vlastní HTML soubor.

Kvůli svým omezením byl HTTP/0.9 rychle nahrazen verzí 1.0. **HTTP/1.0** přidává informaci o verzi protokolu na konec řádku v každém požadavku. Na začátek odpovědi byl přidán stavový kód, který umožnil prohlížeči rozpoznat úspěch, nebo chybu žádosti. Dále byl zaveden koncept HTTP hlaviček pro požadavky i odpovědi. To umožnilo přenos metadat a protokol se stal extrémně flexibilním a rozšiřitelným. V roce 1996 byl vydán standard RFC 1945 , který definuje protokol HTTP/1.0. [25]

Mezitím, co probíhala standardizace HTTP/1.0, již vznikala verze další verze. **HTTP/1.1** byl zveřejněn na počátku roku 1997 (standard RFC 2068), pouze pár měsíců po HTTP/1.0. Tato verze umožňuje přenos více souborů po sobě v rámci jednoho spojení. Byly podporovány také chunked odpovědi, tedy odesílání dat klientovi rozdělených na různě velké části. Dále umožňuje udržování spojení TCP (keep-alive-connection). Byl přidán pipelining, což umožnilo odeslat druhý požadavek dříve, než

byla zcela přenesena odpověď na první, tím byla snížena latence komunikace. Díky nově přidané hlavičce host bylo možné hostit různé domény ze stejné IP adresy. HTTP/1.1 byl zdokonalen dvěma revizemi. V roce 1999 standardem RFC 2616 a v roce 2014 standardem RFC 7235. [25]

V průběhu let se webové stránky staly podstatně složitějšími. Mnohdy webové stránky představovaly aplikaci samy o sobě. Začalo být zobrazováno více vizuálních medií. To způsobilo výrazný růst objemu posílaných dat. Z tohoto důvodu zavedla společnost Google v roce 2010 experimentální protokol SPDY. Tento protokol zvyšuje odezvu a řeší problém zdvojování přenosu dat. Protokol SPDY se stal základem pro HTTP/2. [25]

HTTP/2 se liší od protokolu HTPP/1.1 následujícími principy.

Je to binární protokol namísto textového protokolu. Nelze jej číst a vytvářet ručně. Přesto umožňuje implementaci vylepšených optimalizačních technik. Paralelní požadavky mohou být provedeny přes stejné spojení, což odstraňuje omezení protokolu HTTP/1.x. Dochází ke kompresi hlaviček, tím je odstraněna duplicita a režie přenášených dat. [25]

Oficiální standardizace HTTP/2 proběhla v květnu 2015, používání HTTP/2 dosáhlo svého vrcholu v lednu 2022, kdy jí využívalo 46,9% všech webových stránek. [25]

1.9.3 REST API

V roce 2000 byl navržen nový vzor používání protokolu HTTP: reprezentativní přenos stavu (REpresentative State Transfer neboli REST). Tato rozhraní API nebyla založena na nových metodách HTTP, ale spíše se spoléhala na přístup k určitým URI s pomocí základních metod HTTP/1.1. To umožnilo jakékoliv webové aplikaci umožnit API načítat a měnit svá data bez nutnosti aktualizace prohlížečů nebo serverů. Veškeré potřebné informace byly vloženy do souborů, které webové stránky poskytovaly prostřednictvím standardního HTTP/1.1.[25]

Mezi základní principy REST API patří:

- **Zdroje (Resources):** Každý zdroj je identifikován unikátním URI (Uniform Resource Identifier). To může být například adresa URL, která reprezentuje určitý objekt nebo datový záznam. Všechny požadavky se tedy neposílají na jeden centrální bod, ale každý zdroj má svůj koncový bod.[27]
- **Operace (Operations):** REST API definuje standardní operace, které mohou být prováděny nad zdroji. Tyto operace jsou obvykle mapovány na standardní HTTP metody, jako jsou GET (pro čtení dat), POST (pro vytváření nových zdrojů), PUT (pro aktualizaci existujících zdrojů) a DELETE (pro odstranění zdrojů).[27]

- **Reprezentace (Representation):** Data spojená se zdrojem jsou přenášena mezi klientem a serverem ve formě reprezentace zdroje, často v formátu JSON nebo XML. Klienti mohou požadovat určité formáty reprezentace pomocí hlaviček HTTP Accept.[27]
- **Stav (State):** Každý požadavek na server je nezávislý a neuchovává žádný stav na straně serveru. To znamená, že každý požadavek obsahuje veškeré informace potřebné k jeho zpracování, a server nemusí uchovávat žádné kontextové informace mezi požadavky.[27]
- **(Hypermedia as the Engine of Application State - HATEOAS):** Princip HATEOAS umožňuje klientovi dynamicky objevovat a navigovat skrz API pomocí hypertextových odkazů obsažených v odpovědích serveru. Zjednodušeně jde o to, že známe pouze základní koncový bod, který vrací spolu s daty také odkazy na další zdroje, ty zase na další a tak dále. [27]

1.9.4 Stavové kódy

Součásti hlavičky odpovědi v HTTP/1.1 je stavový kód. Nejpoužívanější stavové kódy v rámci REST jsou vypsány v následujícím seznamu.

- **200 OK** - Požadavek proběhl v pořádku.
- **201 Created** - Při POST, pokud byl vytvořen nový obsah.
- **202 Accepted** - Žádost byla přijata, ale zatím nebyla vyřízena.
- **204 No Content** - Když požadavek na server proběhne v pořádku, ale server nic nevrátí.
- **304 Not Modified** - Pokud od posledního požadavku nebyl změněn obsah – používá se pro nativní HTTP cache.
- **400 Bad Request** - Požadavek na server je nějakým způsobem nečitelný (např. špatný formát JSON).
- **401 Unauthorized** - Klient není ověřen.
- **403 Forbidden** - Klient nemá přístup k danému obsahu.
- **404 Not Found** - Zdroj není nalezen.
- **405 Method Not Allowed** - Zdroj není dostupný pro tuto metodu.
- **410 Gone** - Zdroj není na této adrese dostupný - používá se při verzování.
- **414 URI Too Long** - URI požadovaný klientem je delší, než je server ochoten interpretovat.
- **422 Unprocessable Entity** - Chyba validace dat - např. formuláře.
- **429 Too Many Requests** - Pokud klient překročil maximální počet požadavků.[26] [27]

2 Praktická část

Obsahem této kapitoly je výběr vhodného mikrokontroléru a následný návrh celého zařízení. Navržené schéma je rozděleno na jednotlivé funkční celky, které jsou podrobně rozebrány.

2.1 Výběr vhodného mikrokontroléru

V kapitole 1.6 byl vytvořen výběr několika možných mikrokontrolérů vyhovujících požadavkům na vyvíjené zařízení. Z těchto možností byl zvolen mikrokontrolér ESP32 z následujících důvodů. ESP32 disponuje dobrým poměrem cena/výkon. ESP32 je velmi známý a používaný mikrokontrolér tím pádem je k němu vytvořena spousta knihoven. Z tohoto důvodu bude snadnější následující vývoj softwaru. Další výhodou je, že firma espressif se zavazuje k výrobě ESP32 do roku 2028.

2.2 Návrh modulové elektroniky

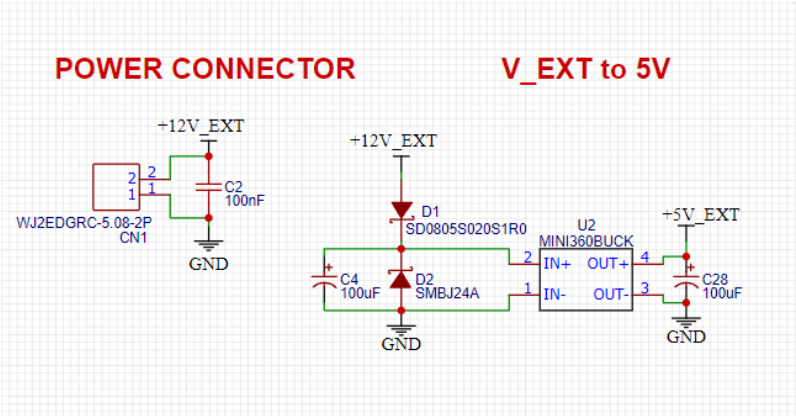
Obsahem této kapitoly je popis navrženého schéma zapojení a desky plošného spoje. Kompletní schéma zapojení je přiloženo v příloze č.1. V následujících podkapitolách jsou popsány jednotlivé části schéma.

2.2.1 Napájení zařízení

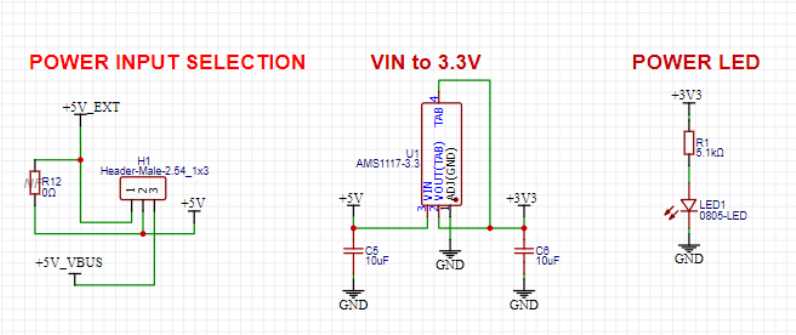
Základem pro správnou funkčnost zařízení je návrh vhodného napájení. Pro napájení bude použit externí 12V zdroj, který bude připojen pomocí konektoru EDGRC-5mm. U tohoto konektoru je připojen oddělovací kondenzátor 100nF. Pro napájení některých částí obvodu je nutné vytvořit napětí o jmenovité hodnotě 5V. Toho je docíleno pomocí hotového modulu MINI-360, jehož srdcem je integrovaný obvod MP1484. Jedná se o step-down měnič se vstupním napětím od 4,75 V do 18 V. Výstupní napětí tohoto čipu je nastavitelné od 1 V až do 17 V. Výstupní proud tohoto obvodu jsou 3 A. Na vstup tohoto modulu je tedy připojeno externí napájecí napětí z konektoru CN1 a to přes schottkyho diodu označenou D1, která slouží jako ochrana proti přepólování. Dále je zde připojen transil označený D2 sloužící pro ochranu modulu proti přepětí. Zapojení napájecí části obvodu je na obrázku 2.1.

Většina komponentů je napájena napětím o hodnotě 3,3 V. Pro vytvoření napětí o této hodnotě je využit lineární stabilizátor AMS1117-3.3. Na jeho vstup je přivedeno buď napětí 5 V z výstupu step-down měniče MINI360 nebo napětí 5 V z konektoru USB. Výběr tohoto napájení je realizován pomocí jumperu. Možnost napájení z USB je především pro pro vývoj softwaru. Pro signalizaci napájení je na

výstupu lineárního stabilizátoru umístěna LED s předřadným rezistorem o hodnotě 5,1 kΩ. Schéma zapojení lineárního stabilizátoru je na obrázku 2.2.



Obr. 2.1: Schéma napájení 5 V

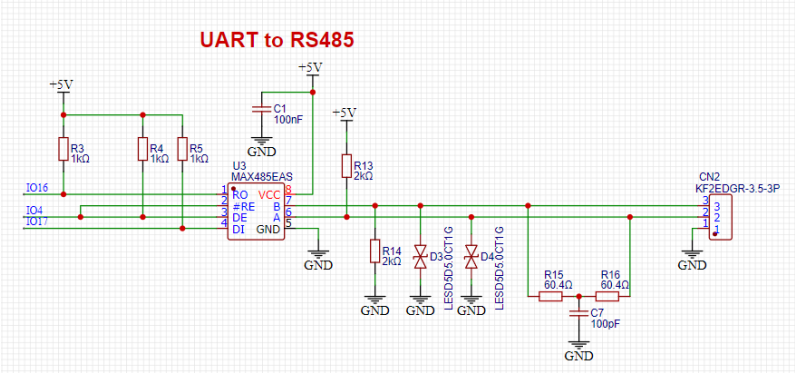


Obr. 2.2: Schéma lineárního stabilizátoru pro napájení 3,3 V

2.2.2 Převod UART na RS-485

Pro převedení komunikačního standardu UART na standard RS-485 slouží integrovaný obvod MAX485, který obsahuje dva operační zesilovače. MAX485 je napájený napětím 5 V. U vstupu napájení je připojen blokovací kondenzátor 100nF. Na výstup RO (Reciever output) je připojen signál RX z ESP32, který je na pinu GPIO 16. Negovaný výstup RE (Reciever enable) a vstup DE (Driver enable) jsou spojeny, protože komunikace bude probíhat vždy pouze jedním směrem (Half-duplex). K těmto dvoum vývodům je připojen pin ESP32 GPIO 4. Vstup DI (Driver input) je připojen k ESP TX pin ESP32, který je vyveden na GPIO 17. Na signál A je připojen pullup rezistor o hodnotě 2 kΩ, který definuje klidovou úroveň linky. Stejnou funkci plní pulldown rezistor o hodnotě 2kΩ připojený k signálu B. K těmto signálům jsou připojeny transily sloužící jako ochrana proti přepětí, které může být

naindukováno na linku. Mezi signály jsou připojeny 2 filtry typu dolní propust, složené z rezistorů o hodnotě 60,1 Ω a kondenzátoru 100pF. Toto zapojení nahrazuje terminační odpor o hodnotě 120 Ω, která odpovídá impedanci kabelu. Takto zapojená terminace linky zabraňuje odrazům signálu a filtruje vysokofrekvenční rušení. Schéma zapojení obvodu s MAX485 je na obrázku 2.3.



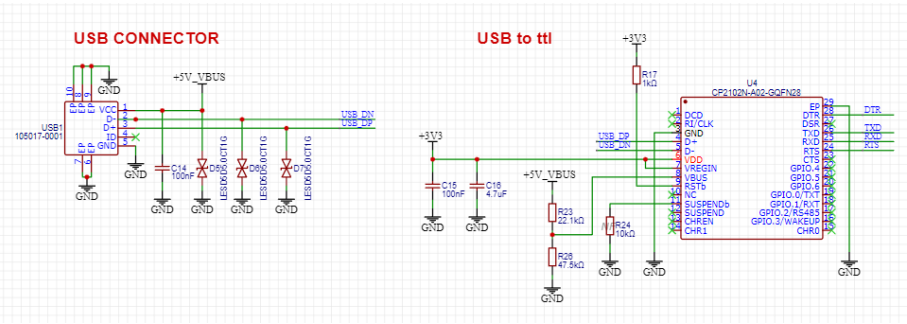
Obr. 2.3: Schéma zapojení převodníku MAX485

2.2.3 Konektor USB a převod na ttl

V kapitole 1.4 bylo mimo jiných funkcí určeno, že zařízení bude pro programování a případné aktualizace softwaru vybaveno USB konektorem. Jako USB konektor byl zvolen typ micro usb. Tento typ je z hlediska přenosové rychlosti dostačující. Dalším důvodem výběru typu micro USB jsou jeho rozměry, protože budou využit pouze při programování je žádoucí aby zabíral co nejméně místa. Na obrázku 2.4 je znázorněno schéma zapojení konektoru a převodníku. Na výstup napájení označený +5V_VBUS je připojen blokovací kondenzátor 100 nF. Dále je zde připojen transil LESD5D5.0CTG1 zajišťující ochranu proti přepětí. Stejné transily jsou umístěny také na signálových výstupech.

Pro převedení standardu USB na ttl byl zvolen čip CP2102 v pouzdře QFN28. Převodník CP2102 je napájen napětím 3,3 V z AMS117 lineárního regulátoru popsaném v kapitole 2.2.1. Na napěťový výstup z USB konektoru označený +5V_VBUS je tedy vhodné připojit napěťový dělič z rezistorů o hodnotách 47,5 kΩ a 22,1kΩ, který vytvoří napětí o hodnotě přibližně 3,4 V. Toto opatření zajistí nepřekročení maximálního napětí v případě, kdyby nebyl napájen vnitřní na napěťový regulátor. Hodnota rezistorů použitých pro napěťovým dělič musí být dostatečně přesná, ideálně s 1% tolerancí. Signálové vstupy jsou připojeny přímo z USB konektoru. Vstup pro reset je připojen přes pull-up rezistor o hodnotě 1 kΩ k napájecímu napětí 3,3 V.

Signály DTR (Data Terminal Ready) a RTS (Request To Send) jsou pomocí tranzistorů připojeny k pinům EN a IO0 ESP32. Toto zapojení zajistí restart ESP32 před nahráním programu a následně aktivaci signálů EN a IO0, což je nutné pro úspěšné nahrání programu. Transistory by měli tedy nahradit držení tlačítek reset a boot při nahrávání programu.



Obr. 2.4: Digitální výstupy

2.2.4 Flash paměť

Pro uchovávání dat vybraných parametrů střídače je potřeba zařízení opatřit externí pamětí s dostatečnou velikostí. Pro tyto účely byl zvolen integrovaný obvod W25Q512JV. Jedná se o nor flash paměť o velikosti 512 Mb. Napájecí napětí tohoto čipu je 3,3 V. Paměť integrovaného obvodu je rozdělena na 262 144 stránek o velikosti 256 bajtů. V jednom okamžiku může být zapsáno až 256 bajtů. Stránky mohou být mazány po skupinách o 16 stránkách (4 kB), po skupinách o 128 stránkách (32 kB), po skupinách o 256 stránkách (256 kB) nebo může být vymazána celá paměť. W25Q512JV podporuje standardní sériové periferní rozhraní (SPI), Dual/Quad I/O SPI: sériový hodinový signál, výběr čipu, sériový datový vstup/výstup (DI), I/O1 (DO), I/O2 a I/O3. Jsou podporovány SPI hodinové frekvence W25Q512JV až 133MHz, což umožňuje ekvivalentní hodinové frekvence 266MHz (133MHz x 2) pro Dual I/O a 532MHz (133MHz x 4) pro Quad I/O při použití rychlého čtení Dual/Quad I/O. Tyto přenosové rychlosti mohou předčit běžné asynchronní 8 a 16bitové paralelní paměti Flash.

2.2.5 Digitální výstupy

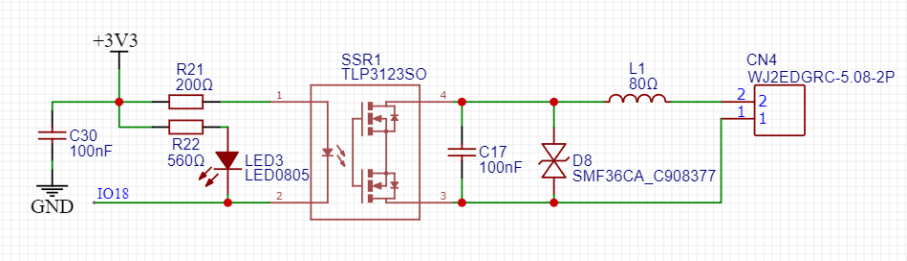
Jako digitální výstupy jsou vytvořeny beznapěťové kontakty ke kterým se připojí relé s s externím napájením 12 V. Pro vytvoření beznapěťového kontaktu je využit optočlen TLP3123, který zajistí galvanické oddělení mikrokontroléru od kontaktů. K diodě optočlenu je nutné spočítat hodnotu předřadného odporu. K tomuto výpočtu

je nutné v datasheetu najít doporučený proud diodou a napětí na diodě. Pro diodu v optočlenu TLP3123 je doporučený proud LED 10 mA a napětí na LED je 1,33 V. Výpočet hodnoty odporu je v následující rovnici.

$R = \frac{U - U_d}{I_d} = \frac{3{,}3 - 1{,}33}{0{,}01} = 197\Omega$

Vypočtená hodnota odporu je 197 Ω, byl zvolen tedy rezistor o hodnotě 200 Ω, která odpovídá řadě vyráběných rezistorů.

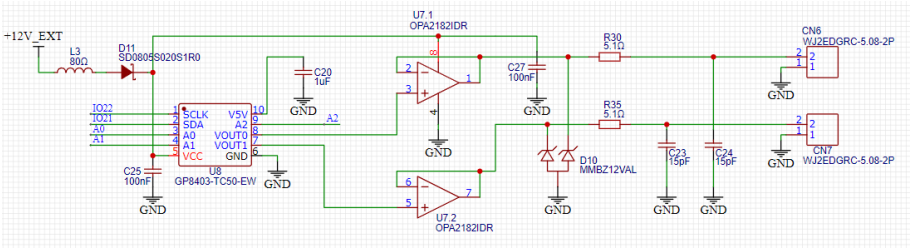
Pro signalizaci sepnutí digitálního výstupu je na vstupu optočlenu umístěna LED. Pro tuto LED byl vypočten předřadný odpor stejným způsobem jako pro LED v optočlenu. Na vstupu i výstupu optočlenu jsou umístěny blokovací kondenzátory 100nF. Na výstupu je dále umístěn transil sloužící jako ochrana proti přepětí a feritový korálek pro odfiltrování rušení. Schéma zapojení digitálních výstupů je zobrazeno na obrázku číslo 2.4.



Obr. 2.5: Digitální výstupy

2.2.6 Analogové výstupy

Pro vytvoření analogových výstupů v průmyslovém standardním rozsahu 0 až 10 V byl zvolen integrovaný obvod GP8403. Tento čip disponuje dvěma analogovými výstupy v rozsahu 0 až 10 V. Nastavení hodnot těchto výstupů probíhá pomocí sběrnice I2C. Adresa pro sběrnici I2C je nastavena pomocí pullup rezistorů o hodnotě 4,7 kΩ připojených k napětí 5V. Hodnota této adresy je 0x5F. Integrovaný obvod GP8403 je napájen napětím z externího zdroje o hodnotě 12 V. Na vstupu napájení je umístěn feritový korálek jako ochrana proti rušení a schottkyho dioda sloužící jako ochrana proti přepólování. Také je zde umístěn blokovací kondenzátor 100 nF. Výstupní signály jsou posíleny operačními zesilovači zapojenými jako sledovač napětí. Operační zesilovače jsou umístěny v jednom pouzdře a jedná se o operační zesilovače OPA2182 od společnosti Texas Instruments. Na výstupech operačních zesilovačů jsou umístěny transily pro ochranu proti přepětí. Za transily jsou umístěny rezistory o hodnotě 5,1 Ω, které zde slouží jako pojistka. Dále jsou na výstupech operačních zesilovačů umístěny filtrační kondenzátory 15 pF. Schéma zapojení digitálních výstupů je zobrazeno na obrázku 2.5.



Obr. 2.6: Analogové výstupy

2.2.7 ESP32

Jak již bylo řečeno v kapitole 2.1 jádrem celého zařízení je mikrokontrolér ESP32. Modul ESP32 je napájen napětím 3,3 V z výstupu lineárního stabilizátoru AMS117 popsaném v kapitole 2.2.1. Bude použit již hotový modul obsahující základní elektronické součástky a anténu na plošném spoji. V zařízení je využito dvou asynchronních sériových komunikačních rozhraní UART. První UART na pinech označených TXD a RXD0 je využit pro programování mikrokontroléru a je tedy připojen pomocí převodníku CP2102 ke konektoru USB, jak je blíže popsáno v kapitole 2.2.3. Druhý UART umístěný na pinech 16 a 17 je využit pro komunikaci se střídačem. Pro tento účel je nutný převod UART na RS485, což zajišťuje integrovaný obvod MAX485 popsaný v kapitole 2.2.2.

Sběrnice I2C je vyvedena na pinech 21 a 22. Na pinu 21 se nachází datový signál SDA a na pinu 22 hodinový signál SCL. Sběrnice I2C je využita pro OLED displej a také pro ovládání integrovaného obvodu GP8403, který obsluhuje analogové výstupy 0 až 10 V.

Sběrnice SPI je vyvedena na pinech 12 až 15. Na pinu 12 je signál MISO, na pinu 13 je signál MOSI, na pinu 14 je hodinový signál SCLK a pin 15 slouží pro výběr slave zařízení (SS). Sběrnice SPI je využita pro přenos dat mezi mikrokotrolérém a externí flash pamětí.

Vstupy 25, 26, 27 a 34 jsou využity pro tlačítka určená k ovládání zařízení.

2.2.8 HMI

Pro vizualizaci požadovaných, případně nastavení může být využit displej, který komunikuje pomocí sběrnice I2C. Pro ovládaní zařízení je možné využít až čtyři tlačítka, která zajistí pohyb v menu. Tlačítka musí být umístěna na samostatném plošném spoji, který se k základnímu plošnému spojem připojí plochým kabelem. Na základní desce plošného spoje je tedy pro tlačítka a displej umístěn konektor označený CN3. Jedná se o standardní kolíkový konektor s roztečí 2,54 mm. Tento konektor je dvouřadý a každá řada disponuje 10 piny. Na jedno řadu je připojen

pouze zemní potenciál. Na druhou řadu jsou připojeny vstupy mikrokontroléru určené pro tlačítka, piny I2C sběrnice, napájecí napětí +5V, napájecí napětí +3,3 V a dva GPIO vývody mikrokontroléru jako rezerva.

2.3 Návrh desky plošného spoje

Po vytvoření schéma zapojení bylo možné přistoupit k návrhu desky plošného spoje. Pro návrh plošného spoje byl stejně jako pro vytvoření schéma zapojení využit návrhový nástroj Easy EDA.

2.3.1 Rozmístění součástek

Jako první krok k úspěšnému návrhu schématu bylo rozmyslet si rozmístění součástek. Nejprve byl umístěn mikrokontrolér ESP 32 zhruba doprostřed k levému okraji desky. Umístění mikrokontréleru ke kraji desky je důležité z důvodu kvality šíření signálu Wi-Fi. Anténa na plošném spoji modulu mikrokontroléru ESP32 je umístěna ve výřezu desky plošného spoje celého zařízení. Tím je zajištěno maximální možné šíření signálu.

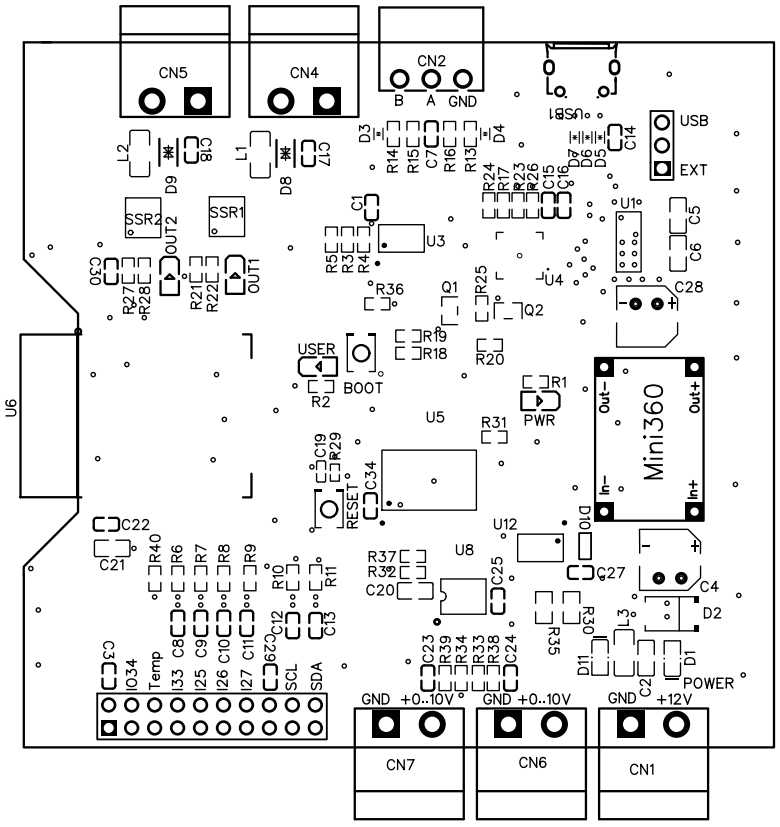
S ohledem na rozmístění pinů jednotlivých periferií na mikrokontroléru byly postupně umístěny ostatní součástky.

V blízké vzdálenosti vývodů UART pinů byl umístěn integrovaný obvod MAX485 (označený U3 na obrázku 2.7), zajišťující převod na komunikační sběrnici RS-485. K vývodům druhého UART byl umístěn integrovaný obvod CP2102 (označený U4 na obrázku 2.7) zajišťující převod na USB sběrnici.

K vývodům SPI sběrnice byl umístěn integrovaný obvod flash paměti W25Q512JV (označený U5 na obrázku 2.7). Nad mikrokontrolér byly umístěny optočleny zajišťující posílení digitálních výsutpů.

V horní části byly umístěny také konektory EDGRC 5 mm pro digitální výstupy (CN4 a CN5). Vedle tohoto konektoru je umístěn konektor EDGRC 3,5 mm (CN2) pro připojení sběrnice RS-485. Posledním konektorem umístěným v horní části plošného spoje je micro USB konektor, sloužící pro programování mikrontroléru. Ve spodní části desky plošného spoje je umístěn konektor pro připojení tlačítek a displeje. Vedlo tohoto konektoru jsou konektory EDGRC 5 mm (CN6 a CN7) sloužící pro připojení analogových výstupů 0-10 V. V pravém spodním rohu je konektor (CN1) sloužící pro napájení celého zařízení.

Nad napájecím konektorem je umístěn step-down měnič MINI-360. Nad tímto měničem je umístěn lineární regulátor AMS1117-3.3. V pravém horním rohu jsou umístěny piny, kterými lze pomocí zkratovací propojky zvolit napájení zařízení z externího zdroje, nebo z USB konektoru.



Obr. 2.7: Rozmístění součástek

Po rozmístění konektorů a integrovaných obvodů na desce plošných spojů následovalo rozmístění dalších součástek nezbytných pro správný chod všech komponentů. Mezi tyto součástky patří například rezistory, které slouží k nastavení napětí nebo adres, a které jsou klíčové pro správnou funkci celého zařízení.

Další důležitou fází bylo umístění ochranných prvků, které zajišťují bezpečnost a spolehlivost zařízení i za neideálních podmínek. Mezi tyto ochranné prvky patří například transilové diody, známé také jako transily, které slouží k ochraně proti přepětí. Tyto diody dokáží rychle a efektivně odvádět nadměrné napětí do země, čímž chrání citlivé součástky před poškozením.

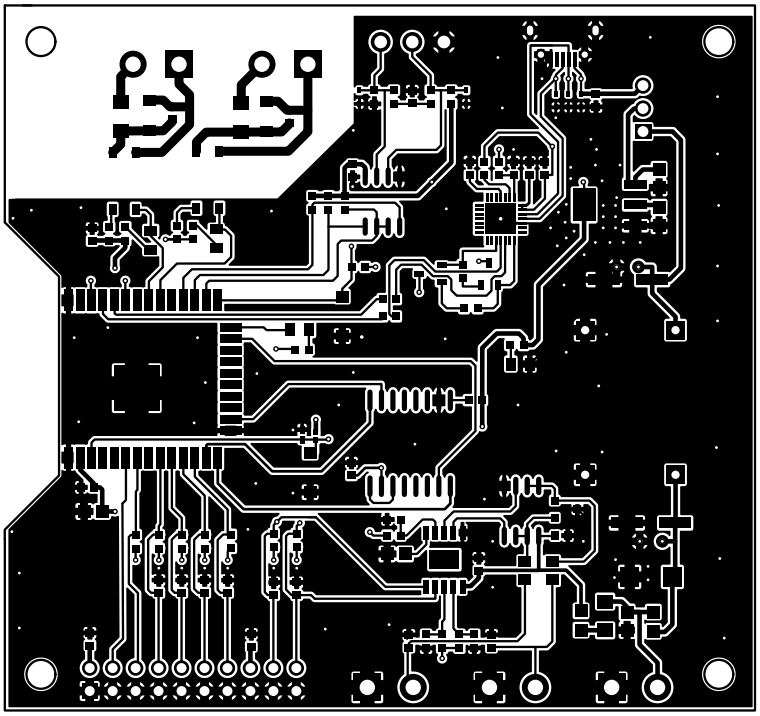
Kromě toho byly do rozmístění začleněny i kondenzátory, které mají několik důležitých funkcí. Jedna z hlavních funkcí kondenzátorů je blokování rušivých signálů a šumu, což pomáhá zachovat čistotu signálu a minimalizovat interferenci v elektronickém zařízení. Dále kondenzátory mohou sloužit k vyrovnávání napěťových poklesů a k zajištění stability napájecích obvodů.

2.3.2 Vytvoření vodivých cest

Po dokončení návrhu rozmístění součástek bylo možné přistoupit k vytvoření vodivých cest plošného spoje.

Plošný spoj byl navržen jako dvouvrstvý, přičemž spodní vrstva byla využita zejména pro spoje, které se křížily. Pro signály byla zvolena tloušťka 10 mils (0,254 mm), a to s ohledem na minimální tloušťku výrobců desek plošných spojů a vhodnou impedanci. Pro cesty, které slouží jako napájení byla zvolena tloušťka 20 mil (0,508 mm). Tím je zajištěna dostatečná rezerva pro proudové zatížení. Cesty na výstupech optočlenů mají tloušťku 1 mm pro zajištění dostatečné proudové zatížitelnosti spínaných cívek připojovaných externích relé.

Vodivá cesta pro zem (GND) byla vytvořená jako rozlitá měď a to ve spodní i vrchní vrstvě. Rozlití vodivé vrstvy GND bylo vyříznuto z oblasti, kde jsou digitální výstupy oddělené optočleny, aby došlo k optickému a izolačnímu oddělení na desce plošného spoje.



Obr. 2.8: Vrchní vrstva desky plošného spoje

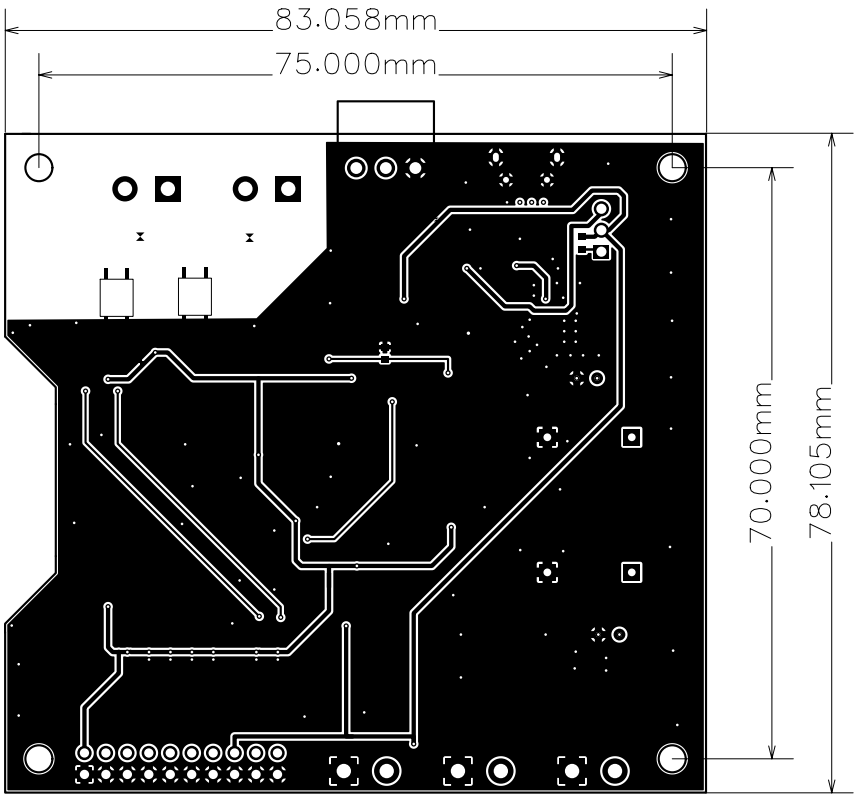
2.3.3 Rozměry desky plošného spoje

Jelikož je navrhované zařízení zamýšleno umístit do modulové rozvodnice, bylo nutné již pří návrhu desky plošného spoje brát ohled na rozměry modulových přístrojů, které jsou do modulových rozvodnic běžně instalovány.

Instalační přístroje, jako například jednofázový jistič, mají zpravidla tyto rozměr: Výška modulu 90 mm a šířka jednoho modulu 17,5 mm.

Výšku desky plošného spoje bylo tedy nutné zvolit maximálně 85 mm, aby byla dostatečná rezerva pro krabičku zařízení. Šířku desky plošného spoje bylo vhodné alespoň přibližně přizpůsobit rozměrům v násobcích velikosti modulu instalačních přístrojů a to opět s rezervou pro krabičku zařízení. Rozměry desky plošného spoje zobrazují kóty na obrázku č. 2.9.

Na desce plošného spoje byly vytvořeny otvory pro upevnění desky plošného spoje do krabičky. Tyto otvory mají průměr 3,2 mm. Jsou tedy určené pro šrouby o velikosti M3. Vzdálenost mezi montážními otvory je v rastru 5 mm pro snazší budoucí návrh krabičky zařízení.



Obr. 2.9: Spodní vrstva desky plošného spoje

2.4 Výroba zařízení

Výrobu zařízení lze rozdělit na čtyři části. První část byla výroba desky plošného spoje, druhá část osazení desky plošného spoje součástkami, třetí část otestovaní funkčnosti všech komponentů a nakonec návrh a výroba krabičky zařízení.

2.4.1 Výroba desky plošného spoje

Pro výrobu desky plošného spoje byla zvolena společnost JLCPCB sídlící v Číně, konkrétně v městě Shenzhen, které je známé jako hlavní centrum elektronického průmyslu v Číně.

Pro výrobu desky plošného spoje bylo nutné vygenerovat několik výrobních podkladů. Výrobní podklady se generují v gerber souborech. Gerber formát je otevřený, ASCII , vektorový formát pro návrhy desek plošných spojů. Jedná se o průmyslový standard pro popis jednotlivých výrobních podkladů, jako například vrstvy mědi, pájecí masky, data o vrtání apod.

Vygenerování všech potřebných gerber souborů zajišťuje vývojový software Easy EDA. Před vytvořením gerber souborů Easy EDA nejprve provede kontrolu, zda jsou všechny součástky zapojeny a následně ještě kontrolu DRC (Design Rule Check).DRC ověří, zda jsou dodržena představená pravidla jako například minimální šířka vodivých cest, minimální vzdálenost mezi cestami, minimální průměr prokovů apod. Po těchto kontrolách Easy EDA vygeneruje veškeré gerber soubory potřebné pro výrobu a uloží je do jednoho zip souboru.

Vygenerovaný zip soubor následně stačí nahrát na webové stránky JLCPCB. Poté stačí zvolit parametry desky plošného spoje, jako je například barva desky plošného spoje, povrchová úprava, tloušťka DPS atd. Na základě těchto podkladů se vytvoří objednávka do výroby. Minimální počet vyrobených desek je 5 kusů. Výroba desky plošného spoje trvala zhruba týden.

2.4.2 Osazení desky plošného spoje

Protože navržené zařízení obsahuje součástky v pouzdře GQFN28 bylo nutné zvolit osazení automatickým osazovačem. Osazení desky bylo tedy provedeno ve společnosti JLCPCB. A to i včetně THT součástek, které byly osazeny ručně v JLCPCB.

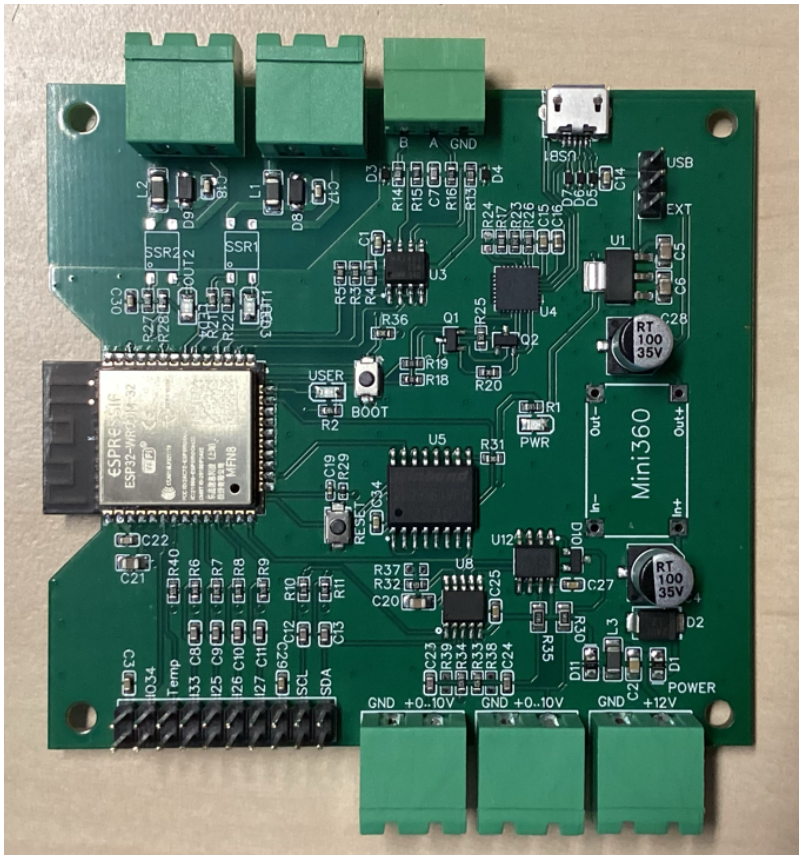
Pro osazení desky plošného spoje bylo nutné vygenerovat další dva soubory. Společnost JLPCB vyřizuje i objednávky jednotlivých součástek. Proto bylo nutné již při návrhu schématu zapojení vybírat takové součástky, které jsou dostupné u dodavatele spolupracujícího se společností JLCPCB. Tento dodavatel je společnost LCSC Electronics.

Pro objednávku součástek bylo nutné vygenerovat BOM soubor (Bill Of Materials), neboli kusovník. Jedná se o textový soubor obsahující: název součástky, její označení ve schématu, typ pouzdra, množství, výrobce, partnumber a cenu. Tyto údaje jsou uspořádány v tabulce. Na základě kusovníku tedy JLCPCB objedná potřebné součástky.

Pro samotné osazení je potřeba vytvořit soubor Pick and Place. Jedná se o soubor v ASCII formátu, který obsahuje pro každou součástku následující data. Designator (označení součástky ve schématu), Mid x a Mid y (souřadnice středu součástky), Layer (vrchní nebo spodní vsrstva) a Rotation (natočení součástky ve stupních).

Výše uvedené soubory vygeneruje Easy EDA. Soubory stačí následně nahrát opět na webové stránky JLCPB a provést kontrolu natočení součástek, poté už jen objednat a zaplatit objednávku. Pro osazení požaduje JLCPCB minimálně 2 kusy hotových osazených desek. Z tohoto důvodu byly objednány dvě kompletně osazené DPS a tři neosazené.

Výroba včetně osazení součástek trvala zhruba měsíc, poté ještě týden doprava z Číny.



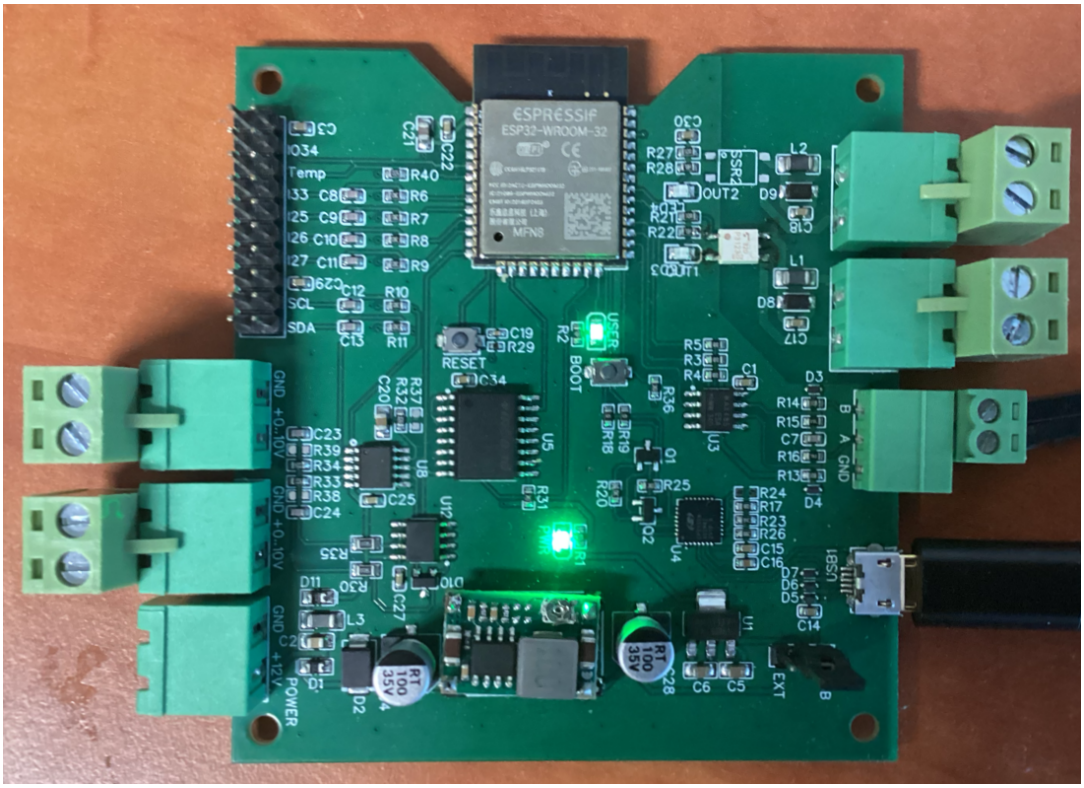
Obr. 2.10: Osazená deska plošného spoje

2.4.3 Testování funkčnosti modulové elektroniky

Po obdržení osazené desky plošného spoje přišlo na řadu testování funkčnosti. První krokem byla optická kontrola správné orientace součástek.

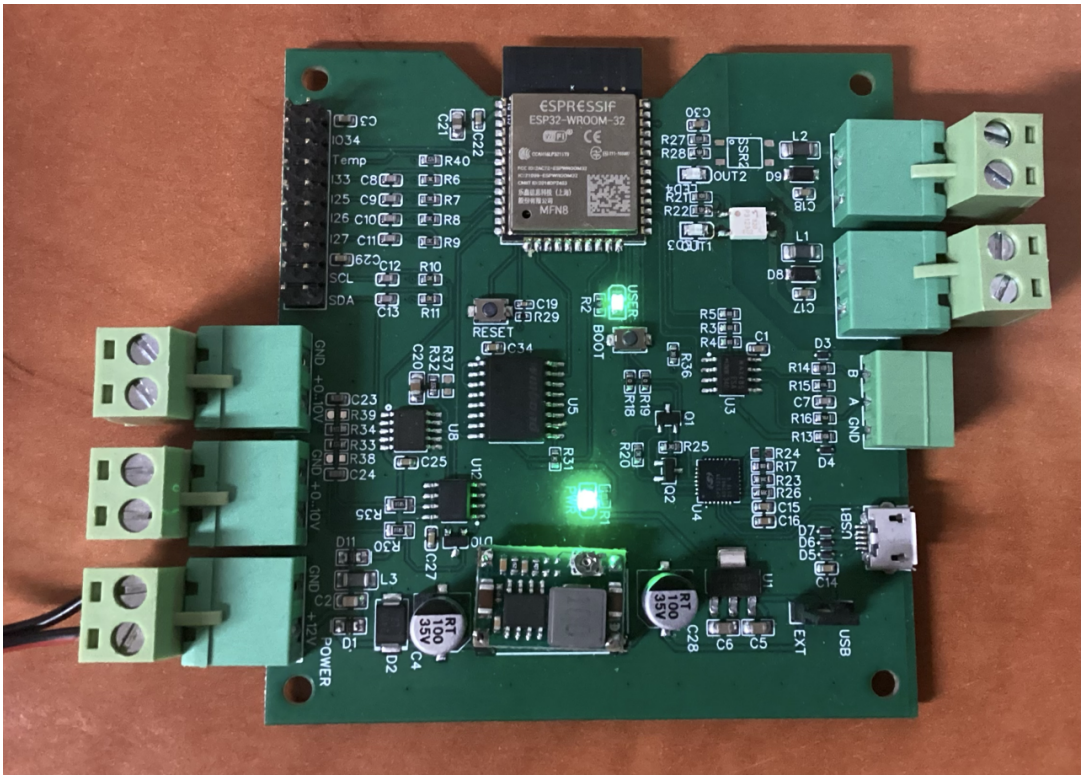
Následně byl pomocí multimetru zkontrolován odpor mezi napájecími kontakty a zemí. Tím bylo ověřeno, že na desce plošného spoje není zkrat a je možné připojit napájení. Před připojením USB konektoru pro napájení byla umístěna zkratovací propojka na pin sloužící pro napájení z USB viz obrázek 2.11. Následně bylo zařízení připojeno pomocí USB kabelu k PC. Po připojení proběhla pomocí voltmetru kontrola napětí v jednotlivých částek obvodu. Touto kontrolou bylo zjištěno, že jsou veškeré napěťové úrovně v pořádku. Funkčnost napájení také signalizuje rozsvícená LED označená PWR.

Dalším krokem bylo otestovaní funkčnosti mikrokontroléru ESP32. Test funkčnosti ESP32 byl proveden nahráním testovacího programu, který bliká LED diodou označenou USER. Tímto testem se také ověřila část obvodu obstarávající převod z USB na UART. Digitální výstupy byly otestovány stejným programem, s tím rozdílem, že v programu byl upraven GPIO, který byl přepínán ze stavu LOW do stavu HIGH. Další části obvodu kromě napájení zařízení budou otestovány při vývoji firmware.



Obr. 2.11: Testování desky plošného spoje

Pro otestování funkčnosti napájení z externího zdroje bylo nutné osadit modul MINI 360 obsahující sted-down měnič. Před osazením tohoto modulu bylo potřeba nastavit pomocí trimeru na modulu MIN 360 výstupní napětí na 5V. Po osazení modulu MINI 360 bylo možné připojit externí napájení 12 V a vyzkoušet funkčnost modulové elektroniky napájené z externího zdroje. Před tímto testem bylo nutné přepojit zkratovací propojku na externí napájení. Ukázka funkčnosti modulové elektroniky je na obrázku 2.12. Po připojení externího napájení 12 V byly opět pomocí voltmetru zkontrolovány napěťové úrovně napájecích větví částí obvodu.



Obr. 2.12: Testování desky plošného spoje

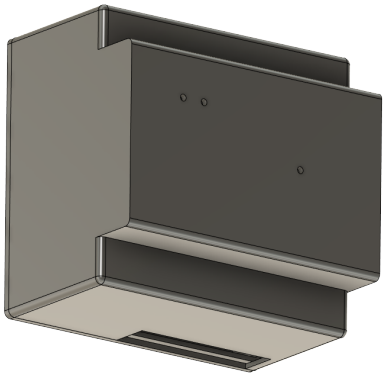
2.5 Návrh krabičky zařízení

Jak již bylo zmíněno v kapitole 2.3.3, zařízení bude umístěné v modulové rozvodnici. Proto bylo nutné při návrhu krabičky zařízení zohlednit standardní rozměry modulových přístrojů. V kapitole 2.3.3 jsou také popsány rozměry desky plošného spoje. Na základě těchto rozměrů a rozměru standardních modulových přístrojů byly určeny následující rozměry: výška 82 mm, šířka byla přizpůsobena rozměru jednoho modulu, tedy v násobcích 17,5 mm. S ohledem na rozměr desky plošného spoje vyšla šířka krabičky zařízeni na velikost pěti modulů, tedy 87,5 mm. Hloubka krabičky je 60 mm.

Pro návrh krabičky byl využit software Fusion 360 od společnosti Autodesk. Jedná se o 3D CAD systém vhodný pro potřeby 3D tisku.

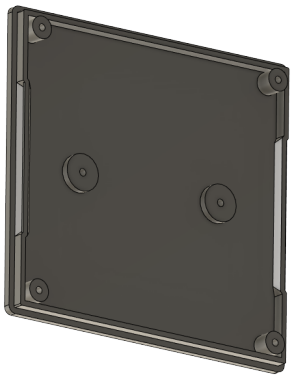
Tvar krabičky je odvozen od tvaru standardních modulových přístrojů. Ukázka kompletní krabičky je na obrázku 2.13.

Na spodní a vrchní straně víčka krabičky jsou v místech konektorů vytvořeny výřezy, aby bylo možné připojení napájení, komunikace a výstupů zařízení. Ve víčku jsou také otvory v místech naproti LED indikující aktivní digitální výstupy a také naproti LED signalizující napájení zařízení.



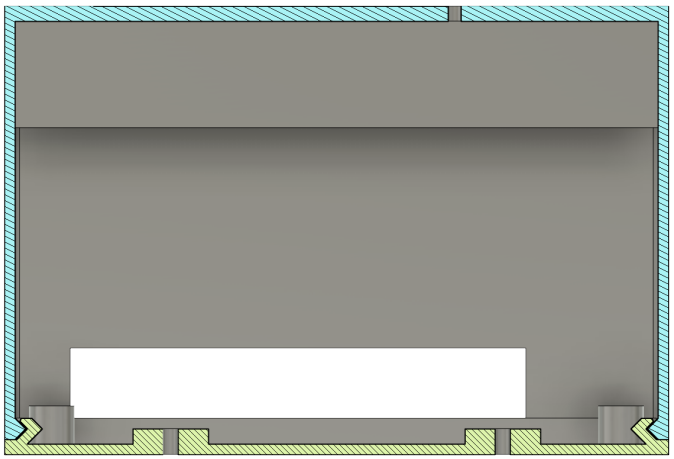
Obr. 2.13: Náhled kompletní navržené krabičky

Ve spodní části krabičky jsou v rozích vytvořeny distanční sloupky, do kterých bude uchycena deska plošného spoje. V místech, kde budou pomocí vrutu připevněny držáky na DIN lištu jsou vytvořeny kruhové výztuže s otvorem pro vrut.



Obr. 2.14: Náhled spodní části navržené krabičky

Spojení spodní části krabičky s víčkem krabičky je provedeno zacvakávacími klipsy na stranách krabičky. Na obrázku 2.15 je zobrazen řež krabičky ze kterého je patrný mechanizmus zacvaknutí dílů krabičky do sebe.



Obr. 2.15: Řez navrženou krabičkou

2.6 Vývoj firmware

Pro vývoj firmware mikrokontroléru ESP32 byl zvolen framework ESP-IDF, což je oficiální IoT framework pro vývoj firmware mikrokontroléru od společnosti Espressif. Framework ESP-IDF byl nainstalován jako rozšíření do vývojového prostředí Visual studio code. Důvodem pro výběr vývoje firmware v prostředí VS code a ESP-IDF frameworku byla hlavně možnost psaní kódu ve standardních knihovnách jazyka C, případně C++, což umožňuje vytvořit kód na profesionálnější úrovni oproti Arduino frameworku. Další výhodou ESP-IDF je, že se jedná o oficiální framework přímo od výrobce a je tedy pravděpodobné, že bude obsahovat méně chyb v knihovnách a s tím spjatá kvalitně provedená dokumentace knihoven.

2.6.1 Struktura programu

Program je rozdělený do souborů podle jednotlivých funkční celků. Například kód obsluhující Modbus komunikaci je v samostatném souboru, respiktive ve dvou soborech: modbus.h a modbus.c.

Jelikož ESP-IDF podporuje freeRTOS byl pro vývoj firmware využit tento operační systém reálného času. Pokud je obsluha některé z periferií potřeba volat opakovaně v daných intervalech je pro ni vytvořena vlastní úloha. Například pro Modbus je vytvořena úloha mb_master_task ve které je nekonečná smyčka. Definice parametrů jednotlivých úloh je v souboru tasks_common.h

2.6.2 Program obsluhující protokol Modbus RTU

K vytvoření kódu obsluhujícího protokol Modbus RTU byla využita knihovna esp_modbus_master.h

Prvním krokem bylo nastavení portu a rozhraní využívaného pro Modbus komunikaci. Pro tento účel byla vytvořena funkce master_init. Tato funkce se nachází v souboru modbus.c na řádku 180. Na začátku této funkce byla vytvořena struktura mb_communication_into_t obsahující parametry sériové komunikace. Následně je volána funkce mbc_master_init, která inicializuje Modbus port pro sériovou komunikaci a nastaví mikroprocesor jako master. Funkce mbc_master_setup, které je předán jako parametr struktura comm, nastaví parametry komunikace. Pro nastavení GPIO pinů je volána funkce uart_set_pin a pomocí funkce uart_set_mode je nastaven UART na mód RS485 half duplex.

Výpis 2.1: Ukázka inicializace modbus komunikace

```
mb_communication_info_t comm = {
    .port = MB_PORT_NUM, // 2
    .mode = MB_MODE_RTU,
    .baudrate = MB_DEV_SPEED, // 9600
    .parity = MB_PARITY_NONE
};
```

Architektonický přístup ESP_Modbus zahrnuje jednu úroveň nad standardním IO ovladačem Modbus. Přidaná vrstva se nazývá Modbus controller a přidává abstrakci characterstic identifier (CID). CID je spojen s odpovídajícími registry Modbus pomocí tabulky nazvané Datový slovník a reprezentuje fyzický parametr zařízení (například napětí, proud atd.) v konkrétním Modbus slave zařízení.[28]

Datový slovník je definován jako pole typu mb_parameter_descriptor_t, kde každý prvek reprezentuje popis jedné fyzické charakteristiky. Tato struktura obsahuje následující prvky. CID, název parametru, jednotky, adresu slave zařízení, typ Modbus registru, adresu registru, velikost registru, offset, datový typ, velikost dat, limity parametrů a mód přístupu k parametru. Detailní popis těchto parametrů sumarizuje tabulka B.1. Ukázku definice datového slovníku zobrazuje výpis 2.3. Způsobem, který je naznačen v této ukázce kódu, jsou definovány veškeré parametry, které jsou z pohledu autora práce vhodné k zobrazení, případně ukládání. Seznam všech parametrů uložených v Modbus uchovávacích registrech solárního střídače Sofar HYD-10KTL byl dodán výrobcem ve formě excelovské tabulky.

Výpis 2.2: Definice datového slovníku modbus

```
const mb_parameter_descriptor_t sofar_parameters[] = {
    // { CID, Param Name, Units, Modbus Slave Addr,
    Modbus Reg Type, Reg Start, Reg Size, Instance Offset,
    Data Type, Data Size, Parameter Options, Access Mode }

    { CID_HOLD_DATA_0, STR("Power_PV1"), STR("kW"),
        MB_DEVICE_ADDR1, MB_PARAM_HOLDING, 0x0586,
        1, 0, PARAM_TYPE_U16, 1,OPTS( 0, 1000, 1 ),
        PAR_PERMS_READ_WRITE_TRIGGER },

    { CID_HOLD_DATA_1, STR("Power_PV2"), STR("kW"),
        MB_DEVICE_ADDR1, MB_PARAM_HOLDING, 0x0589,
        1,0, PARAM_TYPE_U16, 1, OPTS( 0, 1000, 1 ),
        PAR_PERMS_READ_WRITE_TRIGGER },

    { CID_HOLD_DATA_3, STR("SOC_Bat1"), STR("%"),
        MB_DEVICE_ADDR1, MB_PARAM_HOLDING, 0x0608,
        1, 0, PARAM_TYPE_U16, 1,OPTS( 0, 100, 1 ),
        PAR_PERMS_READ_WRITE_TRIGGER },
}
```

Čtení hodnot všech parametrů probíhá ve funkci master_operation_func, která se nachází v souboru modbus.c na řádku 125. Tato funkce v cyklu postupně čte všechny parametry definované v datovém slovníku. Před čtením parametru je provedena kontrola existence parametru odpovídajícího CID. Následně je pomocí funkce mbc_master_get_parameter vyčtena hodnota parametru odpovídajícího CID. Pokud čtení parametru proběhlo v pořádku dojde k uložení hodnoty do dočasné struktury write_memory. Tato struktura obsahuje pouze dvě položky, kterými jsou: memory a flags. V případě neúspěšného čtení hodnoty parametru dojde k nastavení odpovídajícího bitu flagu. Po projití všech CID se struktura write_memory překlopí do struktury actual_memory, což zajišťuje funkce publish_data.

Program obsluhující Modbus komunikaci je obsluhován ve svém vlastním tasku mb_master_task. Na začátku tasku se zavolá funkce master_init a poté je v nekonečné smyčce volána výše popsaná funkce master_operation_func.

Výpis 2.3: funkce pro čtení hodnot modbus slave

```
void master_operation_func(void *arg)
{
    uint16_t value = 0;
    const mb_parameter_descriptor_t* pd = NULL;
    write_data->flags = 0;

    for (...all parameters...)
    {
        mbc_master_get_cid_info(cid, &pd);
        if(ok...)
        {
            mbc_master_get_parameter(cid, pd->param_key,
                &value, ...);

            if (ok...)
                write_mb_data(cid, value);
            else
                write_data->flags |= 1<<cid;

            vTaskDelay(POLL_TIMEOUT_TICS);
        }
    }
    vTaskDelay(UPDATE_CIDS_TIMEOUT_TICS);
    publish_data();
}
```

2.6.3 Program obsluhující Wi-Fi

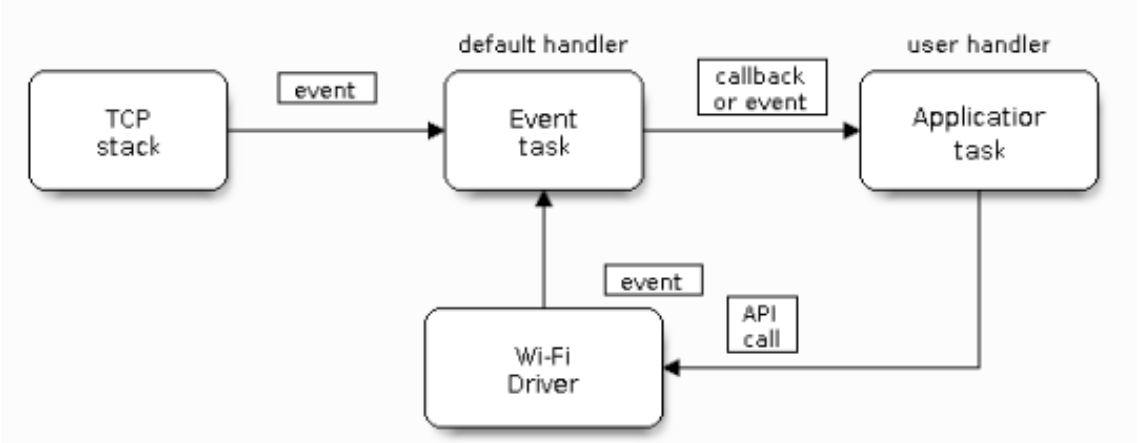
K obsluze Wi-Fi periférie byla využita ESP-IDF knihovna esp_wifi.h. Pro požadovanou funkcionalitu zařízení bylo nutné nakonfigurovat Wi-Fi v režimu přístupového bodu i v režimu stanice. Definice parametrů Wi-Fi připojení je zobrazena na obrázku 2.16.

Ovladač Wi-Fi lze považovat za černou skříňku, která neví nic o kódu vyšší vrstvy, jako je zásobník TCP/IP, úloha aplikace a úloha událostí. Úloha aplikace (kód) obvykle volá rozhraní API ovladače Wi-Fi k inicializaci Wi-Fi a v případě potřeby zpracovává události Wi-Fi. Ovladač Wi-Fi přijímá volání API, zpracovává je a odesílá události do aplikace. [29]

Pro obsluhu Wi-Fi byla vytvořena vlastní úloha s názvem wifi_app_task. Na za-

```
// Wifi application setings
#define WIFI_AP_SSID                "Sofar remote control"      // AP name
#define WIFI_AP_PASSWORD            "password"                  // AP Password
#define WIFI_AP_CHANNEL             1                           // AP channel
#define WIFI_AP_SSID_HIDDEN         0                           // AP visibility
#define WIFI_AP_MAX_CONNECTIONS     5                           // AP max clients
#define WIFI_AP_BEACON_INTERVAL     100                         // AP beacon: 100 ms recommended
#define WIFI_AP_IP                  "192.168.0.1"               // AP default IP adress
#define WIFI_AP_GATEWAY             "192.168.0.1"               // AP default gateway
#define WIFI_AP_NETMASK             "255.255.255.0"             // AP default mask
#define WIFI_AP_BANDWIDTH           WIFI_BW_HT20                // AP bandwith 20 MHz
#define WIFI_STA_POWER_SAVE         WIFI_PS_NONE                // Power save is not use
#define MAX_SSID_LENGTH             32                          // IEEE standard maximum
#define MAX_PASSWORD_LENGTH         64                          // IEEE standard maximum
#define MAX_CONNECTIONS_RETRIES     5                           // Retry number on disconect
```

Obr. 2.16: Definice parametrů Wi-Fi



Obr. 2.17: ESP32 Wi-Fi programovací model [29]

čátku této úlohy je definována proměnná msg datového typu wifi_app_queue. Jedná se o výčtový datový se zprávami pro obsluhu Wi-Fi událostí. Definice těchto zpráv je ve výpisu 2.4 Následuje funkce, která inicializuje obsluhu událostí wifi_app_event_handler_init. Poté je volána funkce wifi_app_default_wifi_init, která inicializuje TCP zásobník a vytvoří výchozí nastavení pro stanici a přístupový bod. Pro nastavení konfigurace přístupového bodu je volána funkce wifi_app_soft_ap_config. K tomuto nastavení jsou využit hodnoty, jejichž definice je na obrázku 2.16.

Po inicializaci a nastavení konfigurace je spuštěna Wi-Fi poomocí funkce esp_wifi_start. Následuje odeslání první zprávy do fronty, jejímž obsahem je následující parametr. „WIFI_APP_MSG_LOAD_SAVED_CREDENTIALS“ do fronty. Poté je spuštěna nekonečná smyčka, která kontroluje příchozí zprávy z fronty, na základě nichž provádí další akce. Například pokud přijde zpráva, která byla odeslána na začátku úlohy: „WIFI_APP_MSG_LOAD_SAVED_CREDENTIALS“ pokusí se načíst ssid a heslo pro připojení k přístupovému bodu. V případě, kdy je načtení těchto

údajů úspěšné dojde k připojení k přístupovému bodu.

Výpis 2.4: Definice zpráv pro obsluhu Wi-Fi událostí

```
typedef enum wifi_app_message
{
  WIFI_APP_MSG_START_HTTP_SERVER = 0,
  WIFI_APP_MSG_CONNECTING_FROM_HTTP_SERVER,
  WIFI_APP_MSG_STA_CONNECTED_GOT_IP,
  WIFI_APP_MSG_USER_REQUESTED_STA_DISCONNECT,
  WIFI_APP_MSG_LOAD_SAVED_CREDENTIALS,
  WIFI_APP_MSG_STA_DISCONNECTED,
} wifi_app_message_e;

typedef struct wifi_app_queue_message
{
   wifi_app_message_e msgID;
} wifi_app_queue_message_t;
```

2.6.4 Úloha obsluhující HTTP server

HTTP server byl vytvořen pomocí knihovny esp_http_server.h. Pro spuštění serveru byla vytvořena funkce http_server_start, která se nachází v souboru http_server.c na řádku 899. Tato funkce je volána úlohou obsluhující Wi-Fi.

Funkce http_server_start volá funkci http_server_configure (soubor http_server.c řádek 694), která vytvoří konfigurační proměnnou config typu httpd_config_t a inicializuje ji s výchozí konfigurací. Poté upraví některá nastavení, jako například velikost zásobníku, prioritu úlohy a jádro procesoru na kterém je úloha spouštěna. Následně nastaví maximální počet URI (Uniform Resource Identifier – „jednotný identifikátor zdroje“) zpracovatelů a nastaví timeouty pro příjem a odeslaní. Dále funkce http_server_configure vytvoří úlohu a frontu zpráv pro správu a sledování stavu HTTP serveru. Po nastavení konfigurace je za pomocí funkce httpd_start spuštěn HTTP server. Po spuštění HTPP serveru jsou registrováni zpracovatelé URI. Ve výpisu 2.5 zobrazeno jak vypadá registrace zpracovatele URI. Tedy nejprve je vytvořena konfigurační struktura typu httpd_uir_t s názvem index_html, která má parametry uri, HTTP metodu, zpracovatele, neboli funkci která zpracuje požadavek klienta. Funkce, která zpracovává tento konkrétní požadavek, s nazvem http_server_index_html_handler, odešle v HTTP odpovědi celou webovou stránku v HTML.

Výpis 2.5: Funkce obsluhující požadavek index_html

```
esp_err_t http_server_index_html_handler(httpd_req_t *req)
{
  ESP_LOGI(TAG, "index.html requested");

  httpd_resp_set_type(req, "text/html");
  httpd_resp_send(req, (const char *)index_html_start,
        index_html_end - index_html_start);

  return ESP_OK;
}
```

2.6.5 Úloha obsluhující SNTP protokol

Funkce sntp_time_sync_task_start vytvoří úlohu obsluhující SNTP protokol.

V úloze se každých deset sekund volá funkce sntp_time_sync_obtain_time. Tato funkce načte čas a pokud není aktuální (aktuálnost času se posuzuje pouze dle roku) zavolá se funkce sntp_time_sync_init_sntp, která nastaví adresu sntp serveru a zajistí inicializaci. Následně je volána funkce setenv s parametrem „CET-1CEST,M3.5.0,M10.5.0/3“, čímž je zajištěno nastavení časové zóny platnou pro Českou republiku. Voláním funkce tzset dojde k aktualizaci běhových dat standardní knihovny C pro novou časovou zónu. Po tomto nastavení je možné načíst čas pomocí standardních C funkcí pro práci s časem. Například ve funkci sntp_get_time_in_sec je použita funkce localtime_r pro získání aktuálního času. Ukázka funkce sntp_time_sync_obtain_time je zobrazena ve výpisu 2.6.

Výpis 2.6: Funkce pro synchronizaci času pomocí SNTP

```
static void sntp_time_sync_obtain_time(void)
{
  time_t now = 0;
  struct tm time_info = {0};

  time(&now);
  localtime_r(&now, &time_info);

  if (time_info.tm_year < (2016 - 1900))
  {
    sntp_time_sync_init_sntp();
    setenv("TZ", "CET-1CEST,M3.5.0,M10.5.0/3", 1);
    tzset();
  }
}
```

2.6.6 Ukládání historických hodnot do externí flash paměti

Pro ukládání historických hodnot bylo zařízení osazeno externí flash pamětí W25Q512JV. Tato flash paměť komunikuje pomocí sběrnice SPI. Flash pamět je v režimu slave a mikrokontrolér je tedy nutné nakonfigurovat v režimu master.

Prvním krokem bylo tedy nastavení sběrnice SPI a zprovoznění komunikace s externí flash pamětí W25Q512JV. K tomuto účelu slouží funkce external_spiflash_init. Nastavení parametrů sběrnice SPI je uloženo ve struktuře bus_config V této struktuře jsou zapsány čísla pinů pro signály MISO (Master In Slave Out), MOSI (Master Out Slave In), a SCLK (Serial Clock). Ukázka struktury bus_config je ve výpisu 2.7.

Výpis 2.7: Nastavení sběrnice SPI v režimu Master

```
spi_bus_config_t bus_config =
{
    .mosi_io_num = SPI2_IOMUX_PIN_NUM_MOSI, // GPIO 13
    .miso_io_num = SPI2_IOMUX_PIN_NUM_MISO, // GPIO 12
    .sclk_io_num = SPI2_IOMUX_PIN_NUM_CLK,  // GPIO 15
    .quadhd_io_num = -1,
    .quadwp_io_num = -1,
};
```

Dále bylo nutné nastavit kanál DMA (Direct Memory Acces). Pro nastavení

DMA kanálu byla vytvořena proměnná spi_dma_chan, která je datového typu enum. Proměnná spi_dma_chan byla nastavena na hodnotu SPI_DMA_CH_AUTO, což zajistí automatickou volbu kanálu ovladačem. Definici proměnné lze vidět ve výpisu 2.8.

Nastavení parametrů pro komunikaci s flash pamětí je uloženo ve struktuře spi_device_config, kde jsou uloženy následující parametry: ID sběrnice ESP32, ID chipu, GPIO pin pro signál CS (Chip Select), mód SPI, který je nastaven na duální mód a frekvence, která je nastavena na 40 Mhz. Struktura spi_device_config je zobrazena ve výpisu 2.8.

Výpis 2.8: Nastavení SPI parametrů pro čip W25Q512JV

```
spi_dma_chan_t spi_dma_chan = SPI_DMA_CH_AUTO;

const esp_flash_spi_device_config_t spi_device_config =
{
    .host_id = host_id,
    .cs_id = 0,
    .cs_io_num = SPI2_IOMUX_PIN_NUM_CS, // GPIO 15
    .io_mode = SPI_FLASH_DIO,
    .freq_mhz = ESP_FLASH_40MHZ,
};
```

Po uložení konfigurace SPI sběrnice do odpovídajících struktur bylo možné zavolat funkci external_spiflash_init, ta uvnitř volá následující funkce pro korektní inicializaci čipu W25Q512JV. Nejdříve je volána funcke spi_bus_initialize, která inicializuje SPI sběrnici nastavenou v parametru této funkce. Dále je volána funkce spi_bus_add_flash_device, která zajistí inicializaci parametrů pro čip W25Q512JV.

Pro práci s externí flash pamětí byla využita knihovna esp_flash_api.h. Tato knihovna integruje externí paměti flash od výrobce Winbond. Inicializaci z této knihovny provádí funkce esp_flash_init. Ukázka funkce external_spiflash_init je zobrazena ve výpisu 2.9

Výpis 2.9: Inicializace SPI pro čip W25Q512JV

```
void external_spiflash_init()
{
    // Initialize the SPI bus
    spi_bus_initialize(host_id, &bus_config, spi_dma_chan));

    // Add device to the SPI bus
    spi_bus_add_flash_device(&ext_flash, &spi_device_config));

    esp_err_t err = esp_flash_init(ext_flash);
```

ESP-IDF používá pro správu flash pamětí vyšší vrstvu, která je nazvána jako tabulka oddílů. Tato tabulka uchovává informace o různých paměťových oblastech, jako jsou například zavaděč, paměťové prostory pro více druhů aplikací, různé binární soubory aplikací, data, souborové systémy apod.

Tabulku oddílů, je možné nastavit pomocí konfiguračního souboru ve formátu csv, ale lze ji také editovat za běhu programu.

Pro ukládání historických hodnot do externí flash paměti je tedy vhodné využít této vyšší vrstvy a přidat externí flash paměť do tabulky oddílů.

Toho lze docílit zavoláním funkce esp_partition_register_external. Jedná se o funkci z knihovny partition.h. Parametry této funkce jsou: ukazatel na strukturu identifikující flash čip, offset adresy, kde má oddíl začínat, velikost oddílu v bajtech, název oddílu, typ oddílu, podtyp oddílu a výstup (ukazatel na výslednou strukturu esp_partition_t.

Z důvodu vyrovnání opotřebení externí flash paměti bylo rozhodnuto pro ukládání historických hodnot do souboru pomocí souborového systému SPIFFS, který zajišťuje kýžené vyrovnání opotřebení, ale i například kontrolu konzistence souborového systému.

Pro vytvoření souborového systému SPIFFS je potřeba nejprve vytvořit strukturu s konfigurací datového typu esp_vfs_spiffs_conf_t. Tato struktura obsahuje následující položky: základní cesta, název oddílu v tabulce oddílů, maximální počet souborů otevřených v jeden okamžik, a nastavení formátování souborového systému při selhání připojení. Dalším krokem je zavolat funkci esp_vfs_spiffs_register, které je předán ukazatel na konfigurační strukturu typu esp_vfs_spiffs_conf_t.

Ukázka definice konfigurační struktury a volání funkce esp_vfs_spiffs_register je na výpisu 2.10

Výpis 2.10: Inicializace SPIFFS souborového systému

```
esp_err_t spiffs_init()
{
    esp_err_t ret;
    esp_vfs_spiffs_conf_t extspiffsConf = {
        .base_path="/spiffs",
        .partition_label = extern_flash_spiffs,
        .max_files=3,
        .format_if_mount_failed=true
    };

    ret = esp_vfs_spiffs_register(&extspiffsConf);
    return ret;
}
```

Zápis historických dat do flash paměti je navržen tak, že se vždy každý den vytvoří nový soubor s názvem ve formátu: dd.mm.yyyy (den, měsíc, rok). Samotný zápis historických dat do flash paměti je realizován funkcí nazvanou spiffs_write. Tato funkce nejprve ověří, zdali již byl vytvořen soubor pro aktuální den, v opačném případě vytvoří nový soubor. Následně ověří, zdali od poslední zápisu uběhlo pět minut a zapíše data do souboru.

Data jsou ukládána ve struktuře dataLog, která obsahuje tři prvky. Čas zápisu, klíč uchovávající typ a název parametru a samotnou hodnotu parametru. Velikost prvků struktury dataLog byla přizpůsobena tak, aby celková velikost struktury měla rozměr 2^n. Důvodem přizpůsobení rozměru této struktury je, aby při zápisu do flash paměti nezasahovala do dvou stránek najednou (velikost jedné stránky je 256 bajtů). Struktura dataLog je zobrazena ve výpisu 2.11.

Výpis 2.11: Struktura ukládaných dat

```
typedef struct
{
    uint16_t  time;
    uint16_t  key;
    uint32_t  value;
} dataLog;
```

Pro každou položku struktury byla vytvořena vlastní funkce, která nastaví odpovídající hodnotu. Uložení hodnoty času obstarává funkce set_data_log_time. Tato funkce na základě předem definovaného počtu zapisovaných parametrů zapíše aktuální hodnotu času ke všem parametrům. Zápis klíče zajišťuje funkce set_data_log_keys,

ta přiřadí k zápisům čísla, podle kterých lze z tabulky datového slovníku definovaného v souboru modbus.c zjistit informace o parametru, tedy například název a jednotky. Zápis samotných hodnot parametrů provádí funkce set_data_log_values. Ta na základě pole data_index zapíše odpovídající hodnoty parametrů, které čte ze struktury actual_data definované v kódu obsluhující Modbus.

2.7 Webové stránky

Webové stránky byly vytvořeny v jazycích html, css a JavaScript. Jsou tedy složeny ze tří souborů: index.html, app.css a app.js. Tyto soubory jsou uloženy ve flash paměti mikrokontroléru ESP32. Strukturu a vzhled webové stránky definují soubory index.hrml a app.css. Funkcionalitu webové stránky definuje soubor app.js. Tyto soubory lze najít ve složce se zdrojovým kódem. Ve složce main se nachazí podsložka webpage obsahující zmíněné soubory. Ukázka vzhledu webové stránky je v příloze D. Při vytváření JavaScriptových funkcí byla využita knihovna JQuery, která byla uložena do flash paměti ESP32.

Na začátku JavaScriptového kódu je funkce $(document).ready, která po načtení DOM (document object model) nastaví časový interval pro funkci getSofarParameters a funkci getHstoricalData. Dále nastaví tlačítku pro připojení k Wi-Fi událost on.click, která spustí funkci checkCredentials.

Funkce gatSofarParameters, která je volána každých pět sekund, pošle požadavek na HTTP server s URI: „/SofarParameters.json“. Po příchozí odpovědi od serveru obsahující JSON se všemi parametry je zobrazí na webové stránce.

Volání funkce getHistoricalData je provedeno po načtení webové stránky a následně každou minutu. Funkce getHistoricalData pošle postupně HTTP GET požadavky pro historická data uložená v externí flash paměti. Následně jsou z dat v odpovědích vytvořeny grafy pomocí knihovny Chart.js.

Při stisku tlačítka „Připojit“ se zavolá funkce checkCredintials, která z textových polí načte hodnotu ssid a hesla. Následně ověří zda se nejedná o prázdné řetězce. Pokud ano vypíše na webovou stránku upozornění pro uživatele. V případě, že se nejedná o prázdné řetězce zavolá se funkce connectWifi. Tato funkce odešle POST požadavek s SSID a heslem uložených v hlavičkách a zavolá funkci startWifiConnectStatus se zpožděním 2,8 sekundy. Tato funkce odešle požadavek „/wifiConnectStatus“ a následně podle odpovědi vypíše stav připojení k Wi-Fi.

2.8 Testování funkčnosti zařízení

Testování zařízení bylo rozděleno na testy jednotlivých funkčních celků obdobně jako vývoj firmware. Způsoby testování jednotlivých funkčních celků jsou popsány v následujících podkapitolách.

2.8.1 Testování komunikace přes Modbus protokol

Komunikace přes protokol Modbus využívající sběrnici RS485, byla nejprve testována pomocí PC softwaru pyModSlave. Jedná se o volně dostupný Modbus slave simulátor vytvořený v Pythonu.

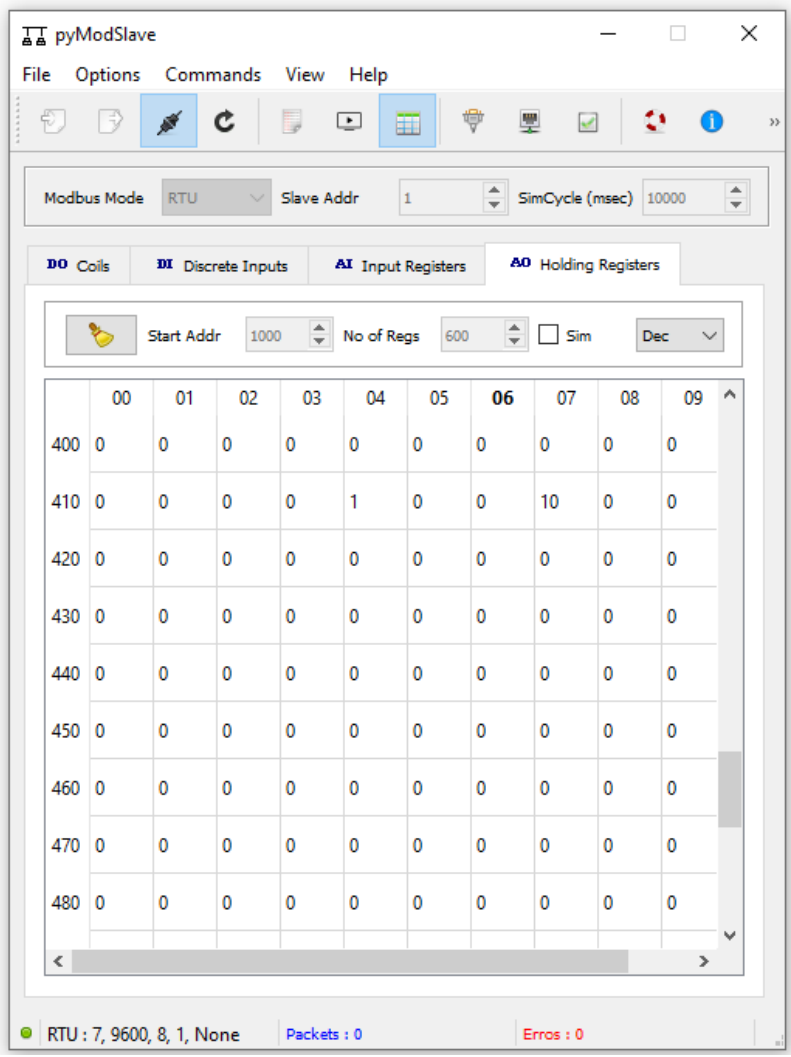
Pro testování Modbus komunikace bylo potřeba připojit výstup sběrnice RS485 zařízení k PC pomocí převodníku z RS485 na USB.

Obr. 2.18: Převodník RS485 na USB

V softwaru pyModSlave bylo potřeba nastavit parametry komunikace, jako je COM port, rychlost komunikace a parita. Následně vybrat záložku Holding registers a zadat počáteční adresu a počet registrů. Pro účel otestování zařízení postačí počáteční adresa 1000 a 600 registrů. Tím bude pokryta celá oblast definovaného datového slovníku dle kapitoly 2.6.2. Poté již stačí kliknout na tlačítko connect. Po připojení je možné zapisovat do tabulky simulovaných registrů.

Na obrázku 2.19 je ukázka okna pyModSlave, kde je na adrese 1414 (hodnota výkonu na prvním stringu v kW vydělená 100) zapsána hodnota 1 a na adrese 1417 (hodnota výkonu na druhém stringu) je zapsána hodnota 10.

Pro ověření funkčnosti Modbus komunikace byly čtené hodnoty vypisovány do terminálu ve Visual Studio Code. Následně byla ověřena funkčnost zobrazení vyčítaných hodnot na webových stránkách a tím i ověřena funkčnost zpracování HTTP požadavku těchto hodnot. Na obrázku 2.20 je výstřižek z webové stránky, kde jsou zobrazeny vyčítané hodnoty.



Obr. 2.19: pyModSlave

2.8.2 Testování Wi-Fi a HTTP serveru

Prvním testem funkčnosti Wi-Fi rozhraní byl pokus o připojení se k ESP32 v režimu STA. Připojení k ESP32 pomocí Wi-Fi bylo provedeno z notebooku pomocí SSID a hesla uvedeného v kapitole 2.6.3. Připojení k ESP32 proběhlo úspěšně, následně byla otestována funkčnost HTTP serveru.

Ve webovém prohlížeči byla zadána IP adresa 192.168.0.1, která byla nastavena jako výchozí IP adresa ESP32. Po zadání IP adresy došlo k načtení webové stránky uložené ve flash paměti ESP32. Tím byla ověřena funkčnost HTTP serveru, na požadavek s URI „/“ byla vrácena odpověď v podobě souboru index.html a došlo k zobrazení webové stránky.

Na webové stránce se nachází textová pole, sloužící pro zadaní SSID a hesla. Pod těmito textovými poli je tlačítko pro připojení k přístupovému bodu. Zadáním

Obr. 2.20: Zobrazení hodnot výkonů stringů na webových stránkách

správných údajů pro připojení a kliknutím na tlačítko „Připojit“ dojde k pokusu o připojení ESP32 k přístupovému bodu. Pokud se pokus o připojení nezdařil, dojde k opětovnému pokusu o připojení dokud se ESP32 nepřipojí, nebo dokud nedojde k překročení maximálního počtu pokusů, tedy 5 opakovaných připojení.

Obr. 2.21: Připojení k Wi-Fi AP

Pro účel otestovaní zařízení byl pomocí mobilního telefonu vytvořen přístupový bod a provedlo se úspěšné připojení. Po úspěšném připojení se pod textovými poli zobrazila IP adresa, kterou přístupový bod přidělil ESP32. Následně byla v jiném zařízení, které bylo připojeno ke stejnému přístupovému bodu jako ESP32 ve webovém prohlížeči zadána přidělená IP adresa. Po zadaní IP adresy došlo k načtení webové stránky uložené v mikrokontroléru ESP32, čímž byla ověřena funkčnost rozhraní Wi-Fi v režimu stanice.

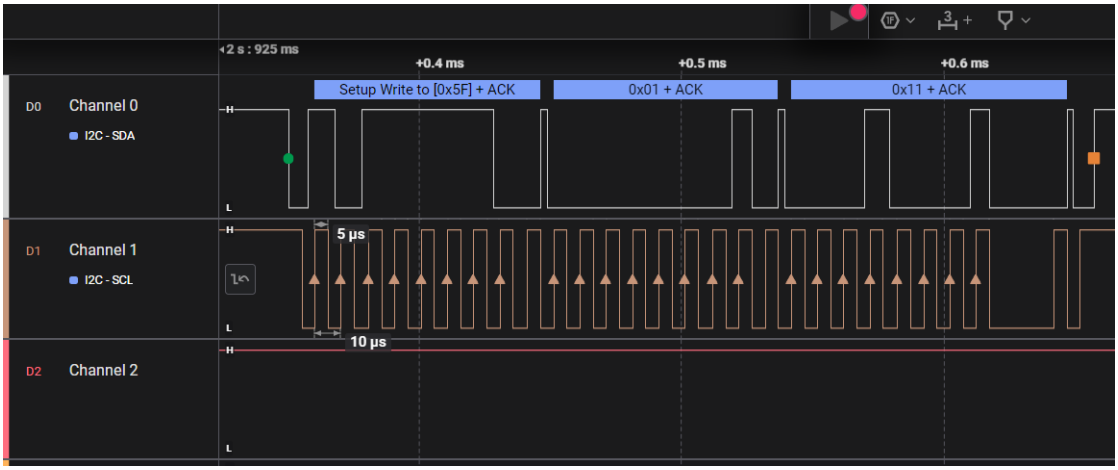
2.8.3 Testování flash paměti

Pro ověření funkčnosti externí flash paměti byly ve funkci external_spiflash_init vytvořeny výpisy do terminálu. Díky těmto výpisům bylo při testování zjištěno, že komunikace SPI s čipem je nefunkční. Zpětnou kontrolou a měřením pomocí logického analyzátoru bylo zjištěno, že jeden z pinů byl nekvalitně připájen k desce plošného spoje. Po odstranění této závady již byla komunikace s externí flash pamětí funkční. Následně byla ověřena funkčnost zápisu do souboru a čtení ze stejného souboru.

2.8.4 Testování analogových výstupů

Při testování funkčnosti analogových výstupů 0 až 10 v byl použit logický analyzátor, kterým byla ověřena funkčnost sběrnice I2C.

Nejprve byl vytvořen zkušební program, který nastavil integrovaný obvod GP8403 do módu 0 až 10 V. Ukázka zaslané zprávy pro nastavení tohoto módu je na obrázku 2.22. Následně bylo v cyklu postupně zvyšováno požadované napětí od 0 do 10 V. Na výstupy byly připojeny voltmetry, kterými byla ověřena funkčnost tohoto programu.



Obr. 2.22: Připojení k Wi-Fi AP

Po dokončení vývoje firmwaru a webových stránek byl ještě proveden test nastavení analogových výstupů přes webovou stránku, kdy byla nastavená hodnota ověřena voltmetrem. Tyto testy prokázali funkčnost analogových výstupů, ale napětí analogových výstupů je zhruba o desetinu voltu nižší než požadované.

2.9 Spotřeba zařízení

Spotřeba zařízení byla změřena pomocí měřícího přístroje SMU (Source Meter Unit) KEITHLEY 2400 . Při měření spotřeby byly zapnuty všechny funkční části zařízení, aby byl změřen maximální odběr zařízení. Na digitální výstupy byly připojeny rezistory o hodnotě 2,2 kΩ, které simulují vyhodnocovací obvody digitálních vstupů jiných zařízení.

Naměřená hodnota odběru zařízení při napájení 12 V byla $(84,1 \pm 0,1)$ mA Po vypnutí digitální a analogových výstupů klesl odběr zařízení na $(65,5 \pm 0,1)$ mA

Rozsah napětí byl nastaven na 20 V a rozsah proudu 1 A.

Závěr

Cílem této diplomové práce bylo navrhnout jednoduchou modulovou elektroniku, která bude realizovat komunikaci se střídačem SOFAR HYD-10-KTL-3PH.

Z tohoto cíle vyplývá šest základních bodů zadání práce, které bylo nutné splnit. Prvním bodem bylo nastudování komunikačních protokolů určených pro komunikaci s tímto střídačem. Tomuto bodu zadání se věnuje kapitola 1.1 a 1.2. V kapitole 1.1 je popsán střídač SOFAR HYD-10-KTL-3PH. V této kapitole bylo zjištěno, že pro komunikaci se střídačem je nutné využít komunikační protokol MODBUS RTU, který podrobně popsán v kapitole 1.2. Protokol MODBUS RTU komunikuje po sběrnici RS-485, kterou se zabývá kapitola 1.3.

Dalším bodem zadání bylo provést rešerši vhodných mikrokontrolérů pro vyvíjené zařízení. Pro splnění tohoto bodu zadání bylo nejdříve nutné specifikovat požadavky na zařízení, čemuž se věnuje kapitola 1.4. Na základě těchto požadavků bylo možné specifikovat požadavky na mikrokontrolér, který řídí celé zařízení. Specifikace požadavků na mikrokontrolér je obsahem kapitoly 1.5. Dle požadavků shrnutých v kapitole 1.5 byl proveden průzkum trhu dostupných mikrokotrolérů. Vhodné mikrokontroléry jsou popsány v kapitole 1.6.

Po dokončení prvních dvou bodů zadání bylo možné přistoupit k samotnému návrhu zařízení. Jako řídící mikrokontrolér byl zvolen ESP32 od společnosti Espressif. Důvody volby tohoto mikrokontroléru jsou popsány v kapitole 2.1.

Po výběru mikrokontrélu následoval návrh schématu zapojení. Návrh obsahoval několik stěžejních částí: návrh napájení celého zařízení, převod UART na RS-485, převod USB na ttl, flash paměť, digitální a analogové výstupy, a nakonec zapojení samotného ESP32. Všechny tyto části jsou popsány v kapitole 2.2 Celkové schéma je přiloženo v příloze A.

Čtvrtým bodem byla realizace zařízení, tedy návrh desky plošného spoje a její následná výroba. Popis postupu návrhu desky plošného spoje je obsahem kapitoly 2.3. Po dokončení návrhu bylo přistoupeno k výrobě desky plošného spoje. Výroba byla řešena dodávkou kompletně osazené desky od společnosti JLCPCB. Způsob objednávky této dodávky je popsán v kapitole 2.4.

K návrhu zařízení byla navíc oproti zadání navržena také krabička na zařízení, jejíž popis je obsahem kapitoly 2.5.

Dalším velmi důležitým krokem byl vývoj firmwaru. Vývoj firmwaru byl rozdělen na jednotlivé funkční celky, které jsou podrobně popsány v kapitole 2.6.

Pátým bodem zadání byla implementace webových stránek, které zobrazují požadované parametry zařízení. Implementaci webových stránek je věnována kapitola 2.7. Vzhled webových stránek je zobrazen v příloze D.

Posledním bodem zadání je implementace archivace historických měření do FLASH

paměti. Pro splnění tohoto bodu bylo nutné učinit několik kroků. Nejprve bylo nutné zvolit čip externí FLASH paměti, který bude komunikovat s mikrokotrolérem ESP32 a následně jej začlenit do desky plošného spoje. Posledním krokem bylo implementovat firmware, který bude samotné archivování dat provádět. To je popsáno v kapitole 2.6.6.

Po dokončení všech bodů zadání bylo provedeno kompletní testování celého zařízení, které je popsáno v kapitole 2.8.

Nakonec bylo ještě nad rámec zadání práce provedeno měření spotřeby zařízení.

Literatura

[1] LIBRA, M. a POULEK, V. *Fotovoltaika: Teorie i praxe využití solární energie.* 1. Praha: Česká zemědělská univerzita, 2009. ISBN 978-80-904311-5-7.

[2] SOFARSOLAR GMBH. *SOFARSOLAR.* Online. C2020-2021. Dostupné z: https://sofarsolar.eu/. [cit. 2023-10-16].

[3] PATSALIDES, Minas; GEORGHIOU, George; STAVROU, Andreas a VENIZELOS, Efthymiou. Voltage regulation via photovoltaic (PV) inverters in distribution grids with high PV penetration levels. Online. *IET Conference Publications*. S. 6. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6521893. [cit. 2024-01-02].

[4] SOFARSOLAR CO. *Uživatelský manuál HYD 5-20-KTL-3PH.* 2023. Dostupné z: https://www1.sofarsolar.com/upload/file/20231223/1703315512994082361.pdf. [cit. 2023-12-05].

[5] Ronešová, A. *Přehled protokolu Modbus [online].* Dostupné z: https://home.zcu.cz/~ronesova/bastl/files/modbus.pdf [cit. 2023-11-6].

[6] MODBUS Organization, *MODBUS Application Protocol 1 1 b. Online. In:* . S. 50. Dostupné z: https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. [cit. 2023-12-17].

[7] WIKIPEDIE. *RS-485.* Online. Dostupné z: https://cs.wikipedia.org/wiki/RS-485. [cit. 2024-11-26].

[8] RENESAS. *RS-485 Design Guide Application Note.* Online. In: . Dostupné z: https://www.renesas.com/us/en/document/apn/rs-485-design-guide-application-note. [cit. 2023-11-26].

[9] TEAXAS INSTRUMENTS. *The RS-485 Design Guide.* Online. In: . Dostupné z: https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1704173061831. [cit. 2023-11-26].

[10] ANALOG DEVICES. *AN-960 RS-485/RS-422 Circuit Implementation Guide Application Note.* Online. In: . Dostupné z: https://www.analog.com/media/en/technical-documentation/application-notes/an-960.pdf. [cit. 23-11-26].

[11] Seed studio. *RTL8710.* Online. Dostupné z: https://www.seeedstudio.com/RTL8710-WiFi-Module-p-2793.html. [cit. 2023-12-17].

[12] TEXAS INSTRUMENTS. *CC3220R, CC3220S, and CC3220SF SimpleLink™ Wi-Fi® Single-Chip Wireless MCU Solutions: datasheet.* Dostupné také z: https://www.ti.com/lit/ds/symlink/cc3220r.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1703853891010&ref_url=https%253A%252F%252Fcz.mouser.com%252F. [cit. 2024-12-20].

[13] *Ti CC3220 Module IPV4 a IPV6 Wifi Transmitter And Receiver Module.* Online. In: . Dostupné z: https://www.wifiblemodule.com/sale-13871302-ti-cc3220-module-ipv4-ipv6-wifi-transm html. [cit. 2024-01-03].

[14] *WiFi modul ESP-12E ESP8266.* Online. In: Bootland. Dostupné z: https://botland.cz/moduly-wifi-esp8266/5463-wifi-modul-esp-12e-esp8266-cerny-11-gp html. [cit. 2024-01-03].

[15] ESPRESSIF. *ESP8266EX.* Online. Datasheet. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf [cit. 2023-12-21].

[16] *Espressif Systems ESP32-WROOM-32.* Online. In: MOUSER ELECTRONICS. Dostupné z: https://cz.mouser.com/ProductDetail/Espressif-Systems/ESP32-WROOM-32M103QH2800PH3Q0?qs=W%2FMpXkg%252BdQ7IcgHFTTq3Ig%3D%3D. [cit. 2024-01-03].

[17] ESPRESSIF. *ESP32 WROOM 32E. Online.* Datasheet. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf. [cit. 2023-12-21].

[18] ESP32 DOCUMENTATION *ESP32 Arduino Core's documentation..* Online. c2016 - 2024. Dostupné z: https://docs.espressif.com/. [cit. 2024-04-26].

[19] *Wi-Fi.* Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024, Dostupné z: https://cs.wikipedia.org/wiki/Wi-Fi#cite_note-3. [cit. 2024-04-28].

[20] *EXPLOITING WIRELESS COMMUNICATIONS FOR LOCALIZATION: BEYOND FINGERPRINTING* Online, Disertace. CASTELLÓN DE LA PLANA: VUT FEKT, 2023. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_

id=260441#page=145&zoom=100,132,778 [cit. 2024-04-28].FOROUZAN, Behrouz A. Data Communications and Networking. Online. 5. The Mc Graw Hill, 2012. Dostupné z: https://elcom-hu.com/Subjects/Computer/Compulsory/Communication/Data-Communications-and-Network-5e.pdf. [cit. 2024-04-28].

[21] *FOROUZAN, Behrouz A. Data Communications and Networking.* Online. 5. The Mc Graw Hill, 2012. Dostupné z: https://elcom-hu.com/Subjects/Computer/Compulsory/Communication/Data-Communications-and-Network-5e.pdf. [cit. 2024-04-28].

[22] *An overview of HTTP.* Online. MDN Web Docs. C1998–2024. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview. [cit. 2024-04-28].

[23] *APLIKACE PRO DOBROVOLNÉ HASIČE* Online, Bakalářská. Brno: VUT FEKT, 2022. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=242545. [cit. 2024-04-28].

[24] *HTTP request methods* Online. MDN Web Docs. C1998–2024. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods [cit. 2024-04-29].

[25] *Evolution of HTTP* Online. MDN Web Docs. C1998–2024. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP [cit. 2024-04-30].

[26] *HTTP response status codes* Online. MDN Web Docs. C1998–2024. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status [cit. 2024-04-30].

[27] *Stopařův průvodce REST API* Online. IT Network. C2024. Dostupné z: https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api [cit. 2024-05-04].

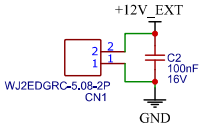
[28] *Modbus Master API Overview* Online. Espressif. C2019 - 2024. Dostupné z: {https://docs.espressif.com/projects/esp-modbus/en/latest/esp32/master_api_overview.html [cit. 2024-05-05].

[29] *Wi-Fi Driver* Online. Espressif. C2019 - 2024. Dostupné z: {https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/wifi.html [cit. 2024-05-05].

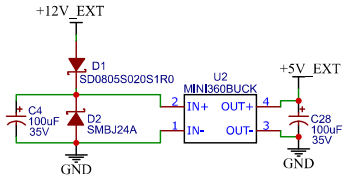
A Schéma zařízení

Na následujících dvou stranách se nachází schéma celého zařízení

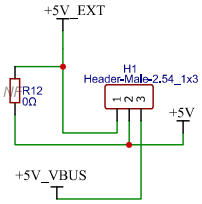
POWER CONNECTOR



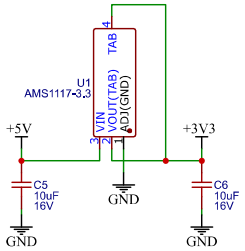
V_EXT to 5V



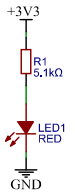
POWER INPUT SELECTION



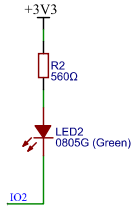
VIN to 3.3V



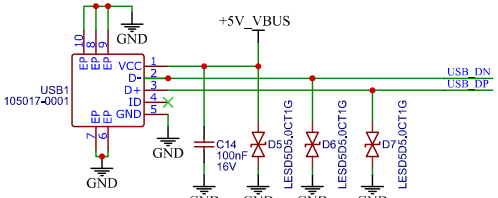
POWER LED



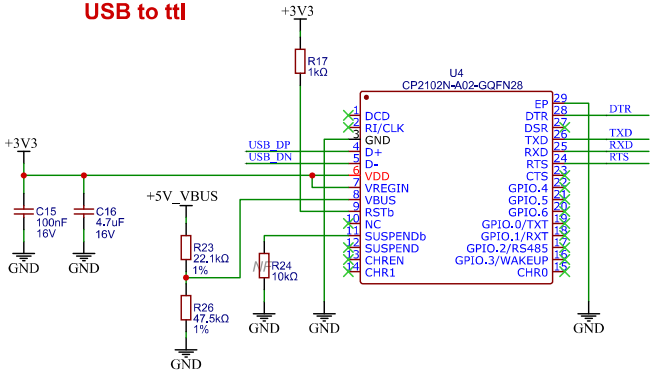
USER LED



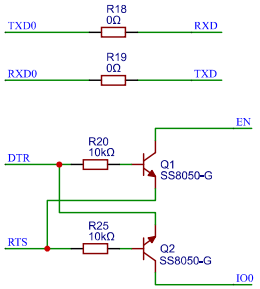
USB CONNECTOR



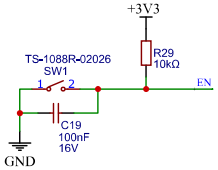
USB to ttl



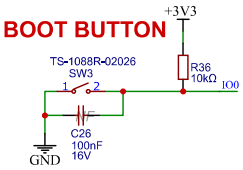
SERIAL SIGNALS HANDLING



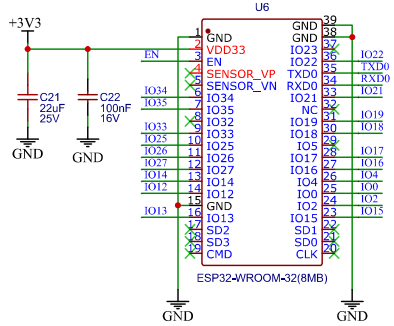
RESET BUTTON



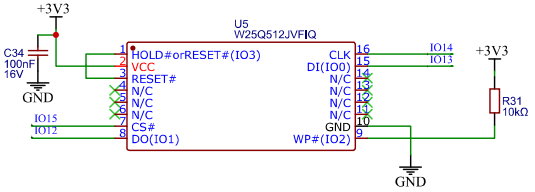
BOOT BUTTON



ESP32 MODULE

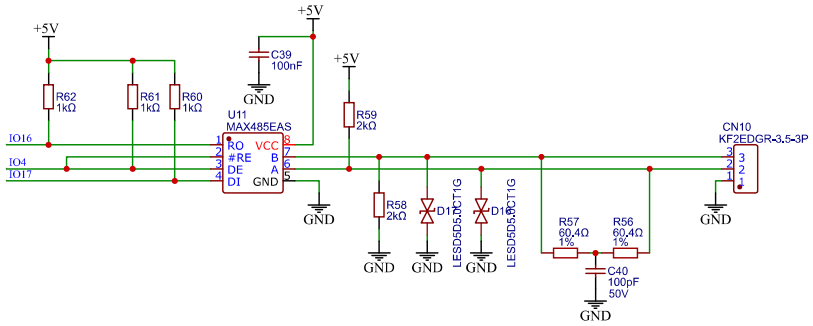


FLASH MEMORY

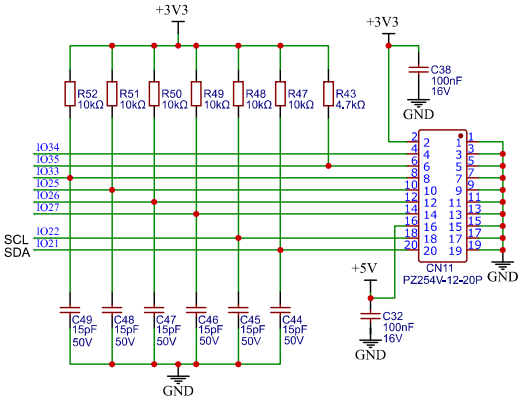


TITLE: Remote control for sofar inverter	REV: 1.0
Company: VUT Brno	Sheet: 1/2
Date: 2023-11-03	Drawn By: Ondřej Pohorský

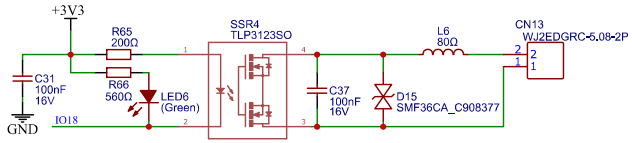
UART to RS485



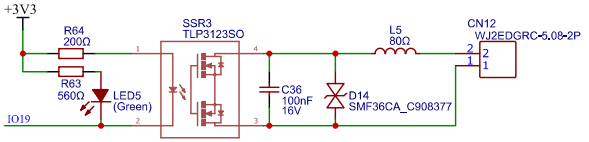
DISPLAY and BTNs CONNECTOR



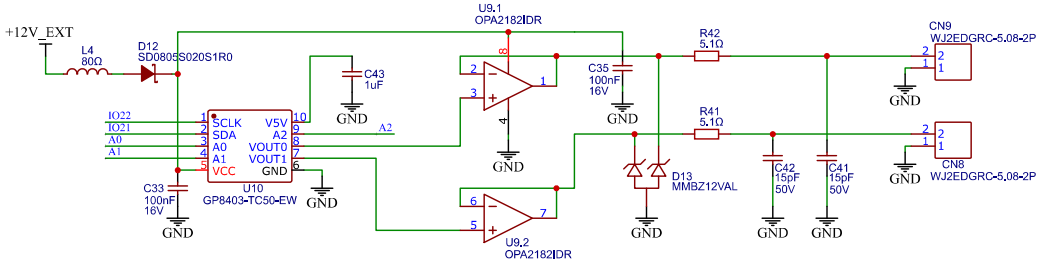
DIGITAL OUTPUT 1



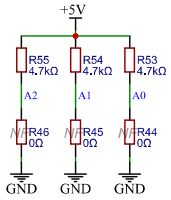
DIGITAL OUTPUT 2



ANALOG OUTPUTS



ADRESS SET



TITLE: Remote control for sofar inverter		REV: 1.0
	Company: VUT Brno	Sheet: 2/2
	Date: 2023-11-03 Drawn By: Ondřej Pohorský	

B Popis parametrů Datového slovníku

Název položky	**Popis**	**Detailní informace**
cid	ID charakterstiky	identifikátor charakteristiky (musí být unikátní)
param_key	Název charakteristiky	String udávající název charakteristiky
param_uints	Jednotky charakteristiky	Fyzikální jednotky charakteristiky
mb_slave_addr	Adresa Modbus slave	Adresa zařízení s korespondujícím CID paramterem
mb_param_type	Typ Modbus registru	Typ Modbus registru. MB_PARAM_INPUT, MB_PARAM_HOLDING, MB_PARAM_COIL, MB_PARAM_DISCRETE.
mb_reg_start	Začátek modbus registru	Relativní adresa registru charakteristiky
mb_size	Velikost Modbus registru	Délka charakteristiky v registrech (dva bajty)
param_offset	Offset instance	Offset instance charakteristiky v bajtech. Používá se k výpočtu absolutní adresy charakteristiky ve struktuře úložiště. Je to nepovinné pole a může být nastaveno na nulu, pokud se parametr v aplikaci nepoužívá.
param_type	Datový typ	Specifikace datové typu charakteristiky
param_size	Velikost dat	Velikost charakteristiky (v bajtech) popisuje velikost dat, která se mají při mapování uchovávat v datové instanci. Toto umožňuje vytvářet datové kontejnery.
param_opts	Možnosti parametru	Limity charakteristiky použité při zpracování alarmu v uživatelské aplikaci (volitelné)
access	Způsob přístupu	Lze použít v uživatelské aplikaci k definování chování charakteristiky při zpracování dat v uživatelské aplikaci

Tab. B.1: Popis Modbus master Datového slovníku[28]

C Obsah elektronické přílohy

Obsahem elektronických příloh je schéma zařízení ve formátu PDF. Dále je zde složka s výrobními podklady DPS, včetně seznamu součástek.

V adresáři „Krabička“ se nachází výkresy krabičky na zařízení, 3D modely krabičky ve dvou různých formátech a vyexportovaný projekt ze softwaru Fusion 360.

Ve složce Sofar_remote_control je kompletní projekt ESP-IDF, včetně konfiguračních souborů. Projekt byl vytvořen a testován ve verzi v5.2.1 ESP-IDF frameworku. Veškeré zdrojové kódy se nachazí v podsložce main.

- `/` ..... kořenový adresář přiloženého archivu
  - `Vzdálený dohled pro solární střídač.pdf` ..... text diplomové práce
  - `Přílohy` ..... přílohy diplomové práce
    - `Sofar_remote_control_schema.pdf`
    - `DPS` ..... výrobní podklady pro DPS
      - `Gerber_PCB_Sofar_esp32_datalogger.zip`
      - `PickAndPlace_PCB_Sofar_esp32_datalogger.csv`
      - `BOM_PCB_Sofar_esp32_datalogger.csv`
    - `Krabička`
      - `vykres_horni_cast.pdf`
      - `vykres_spodni_cast.pdf`
      - `Sofar_remot_contol_case.obj` ..... 3D model krabičky
      - `Sofar_remot_contol_case.3mf` ..... 3D model krabičky
      - `Sofar_remot_contol_case.f3z` ..... export projektu z Fusion 360
    - `Sofar_remote_control` ..... ESP-IDF Projekt
      - `main` ..... zdrojové kódy
        - `webpage` ..... zdrojové kódy webových stránek

D Ukázka kompletní webové stránky

Na následující čtyřech stranách je zobrazena vytvořená webová stránka exportovaná do PDF.

Vzdálený dohled pro solární střídač SOFAR HYD-10-KTL

Výkon jednotlivých stringů

Výkon na stringu 1: 2.23 kW
Výkon na stringu 2: 4.12 kW

Výkon sítě

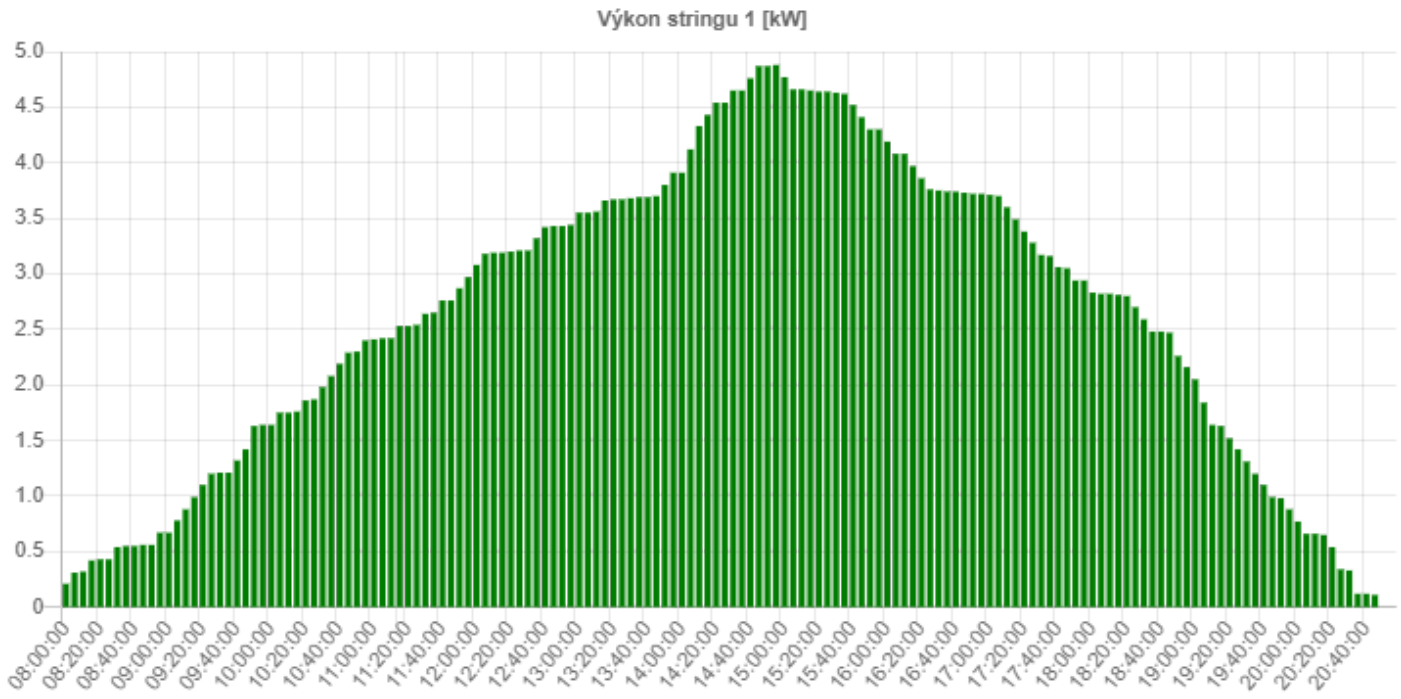
Celkový výkon sítě: 1.83 kW
Výkon L1: 0.86 kW
Výkon L2: 0.55 kW
Výkon L3: 0.42 kW

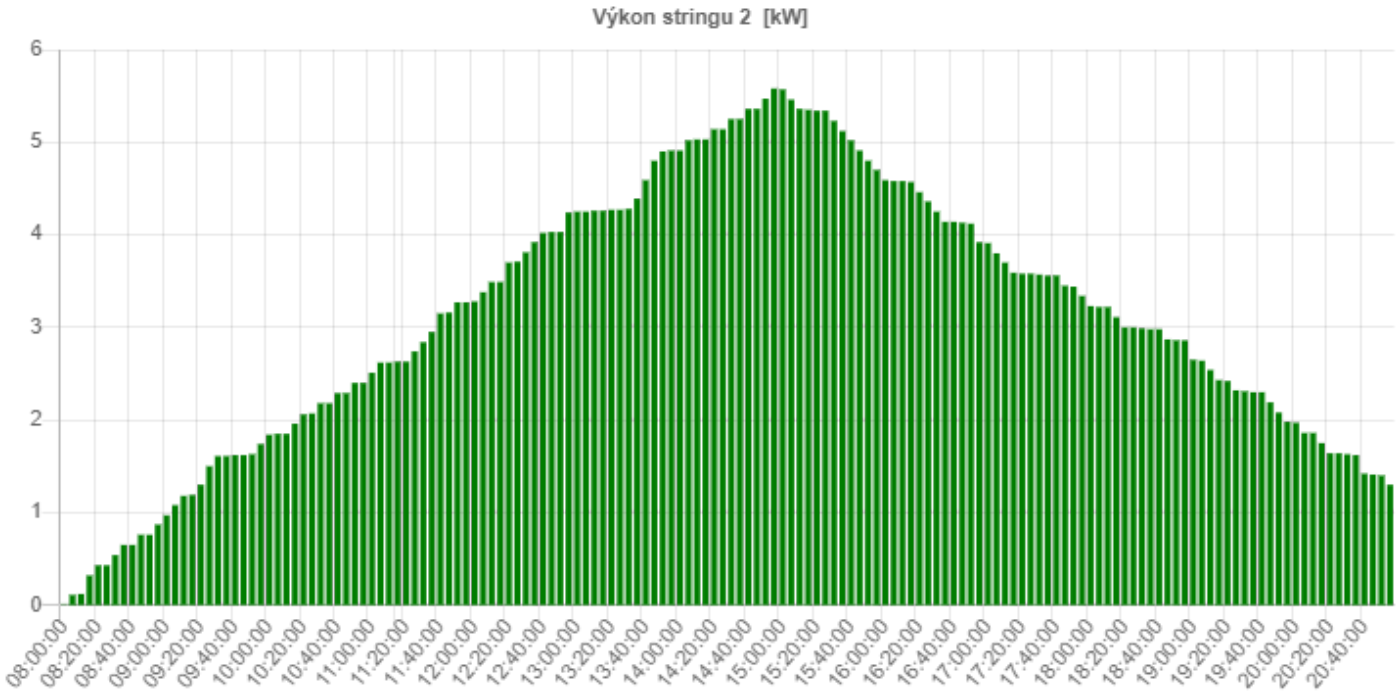
Spotřeba domu

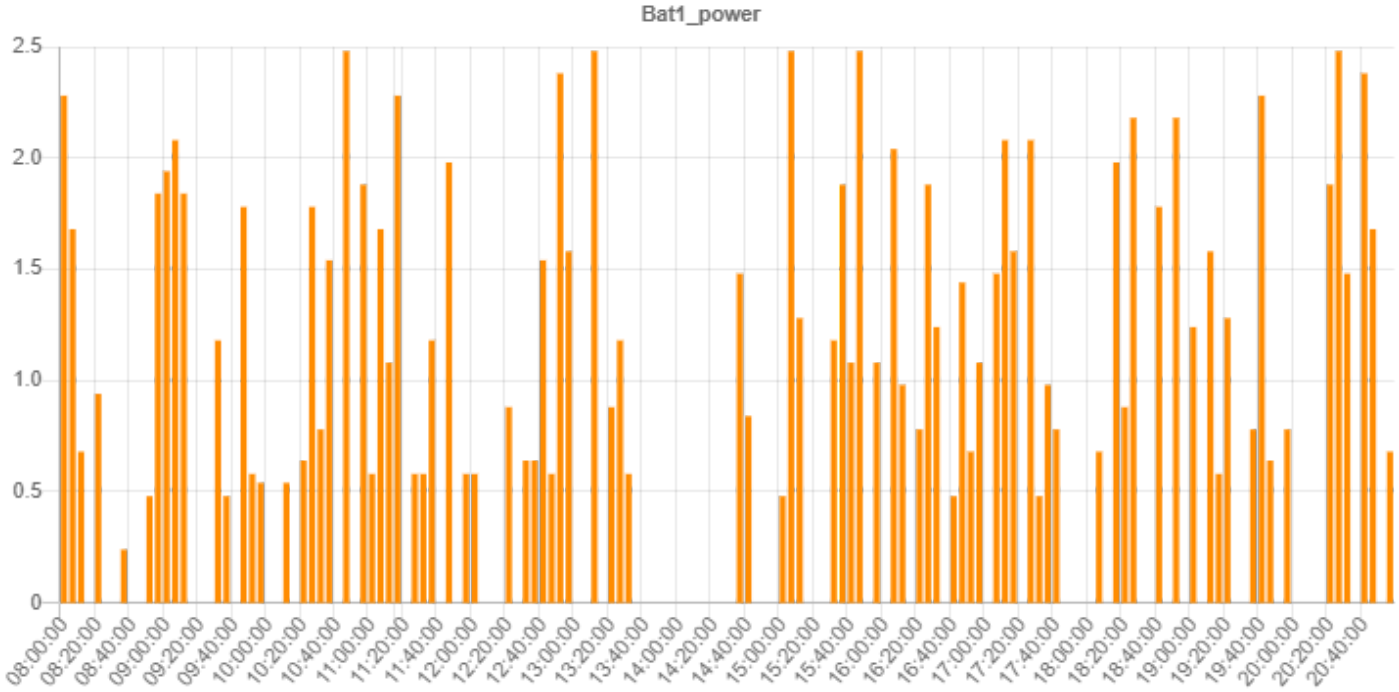
Celková spotřeba: 4.59 kW
Spotřeba na L1: 1.12 kW
Spotřeba na L2: 2.48 kW
Spotřeba na L3: 2.48 kW

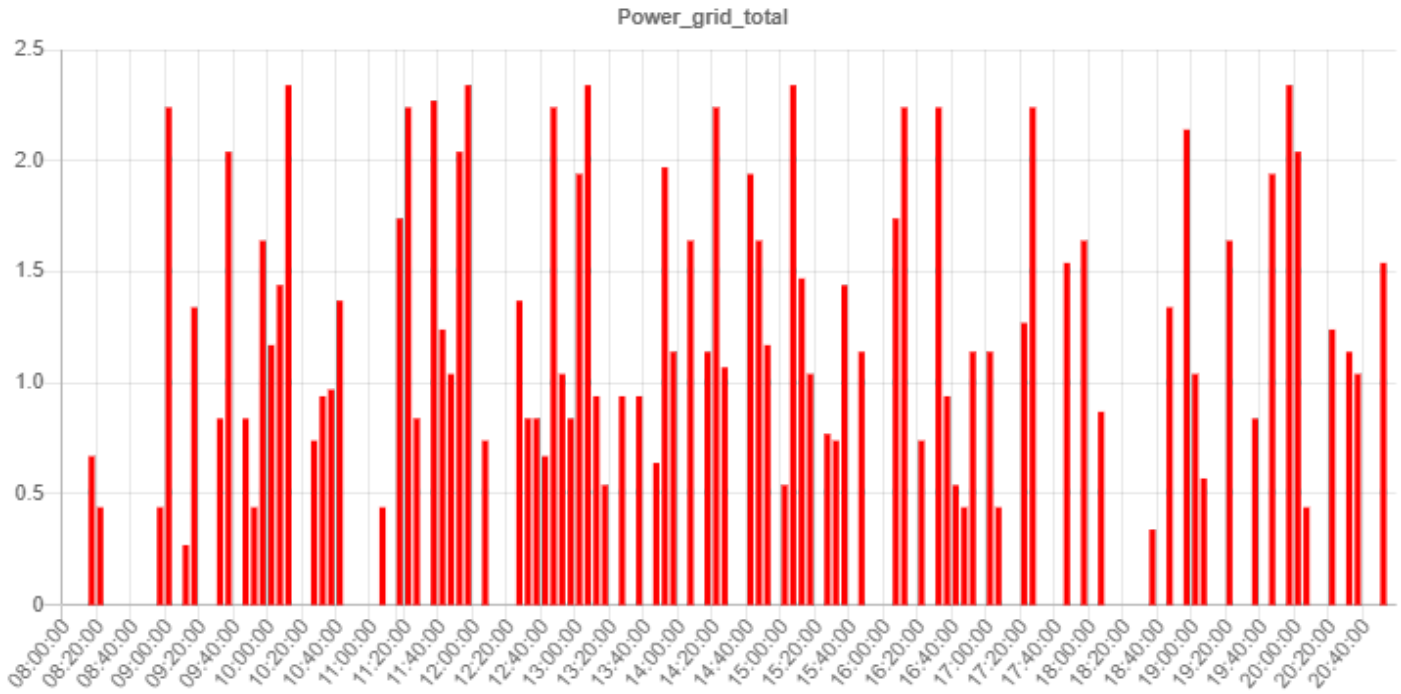
Baterie

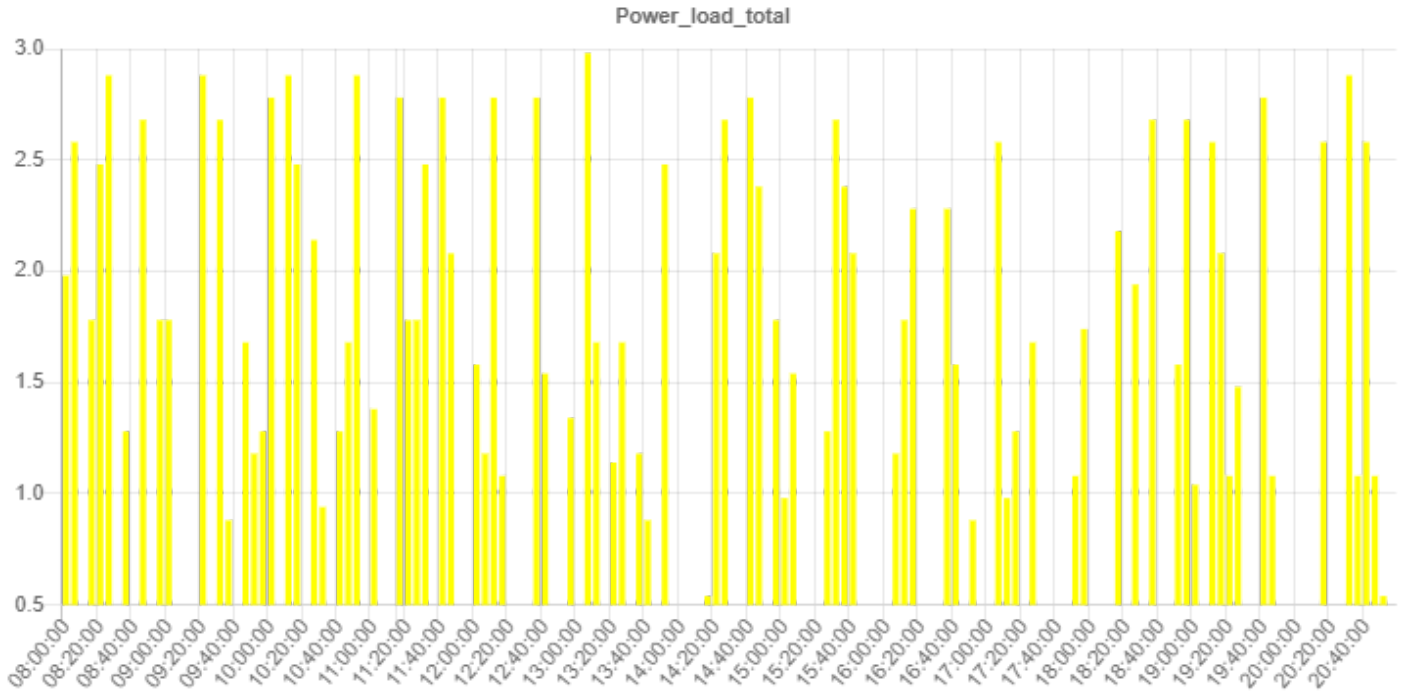
Stav nabití baterie: 83 %
Teplota baterie: 34 °C
Výkon baterie: 1.23 kW




Výkon stringu 2 [kW]
6
5
4
3
2
1
0
08:00:00
08:20:00
08:40:00
09:00:00
09:20:00
09:40:00
10:00:00
10:20:00
10:40:00
11:00:00
11:20:00
11:40:00
12:00:00
12:20:00
12:40:00
13:00:00
13:20:00
13:40:00
14:00:00
14:20:00
14:40:00
15:00:00
15:20:00
15:40:00
16:00:00
16:20:00
16:40:00
17:00:00
17:20:00
17:40:00
18:00:00
18:20:00
18:40:00
19:00:00
19:20:00
19:40:00
20:00:00
20:20:00
20:40:00


Bat1_power
2.5
2.0
1.5
1.0
0.5
0
08:00:00
08:20:00
08:40:00
09:00:00
09:20:00
09:40:00
10:00:00
10:20:00
10:40:00
11:00:00
11:20:00
11:40:00
12:00:00
12:20:00
12:40:00
13:00:00
13:20:00
13:40:00
14:00:00
14:20:00
14:40:00
15:00:00
15:20:00
15:40:00
16:00:00
16:20:00
16:40:00
17:00:00
17:20:00
17:40:00
18:00:00
18:20:00
18:40:00
19:00:00
19:20:00
19:40:00
20:00:00
20:20:00
20:40:00





Spínaní digitálních výstupů

Relé 1 **Relé 2**

Nastavení analogových výstupů

Analogový výstup 1 (0-10 V): Nastavit

Analogový výstup 2 (0-10 V): Nastavit

Připojení k WiFi

SSID Heslo Zobrazit heslo

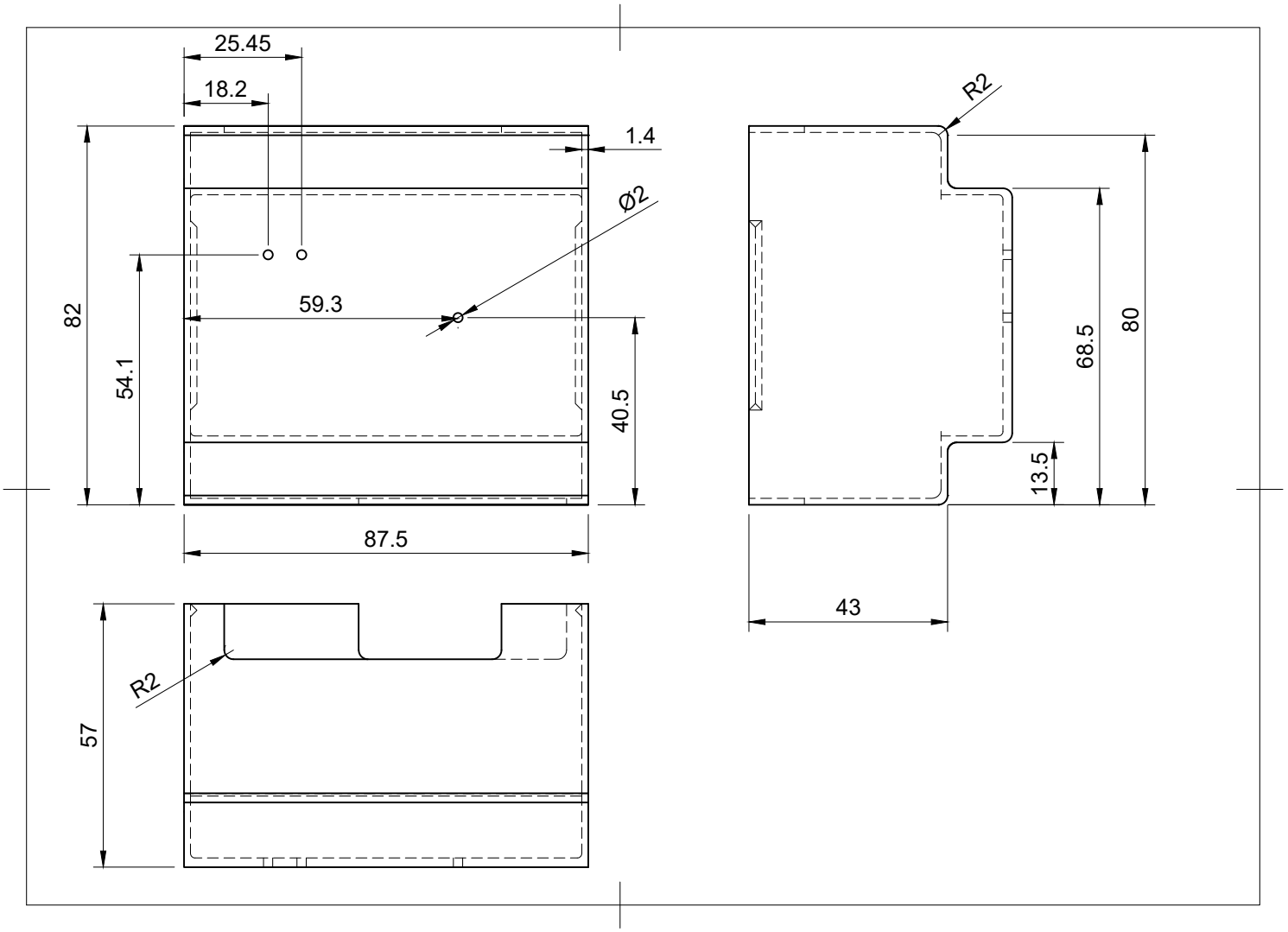
Připojit

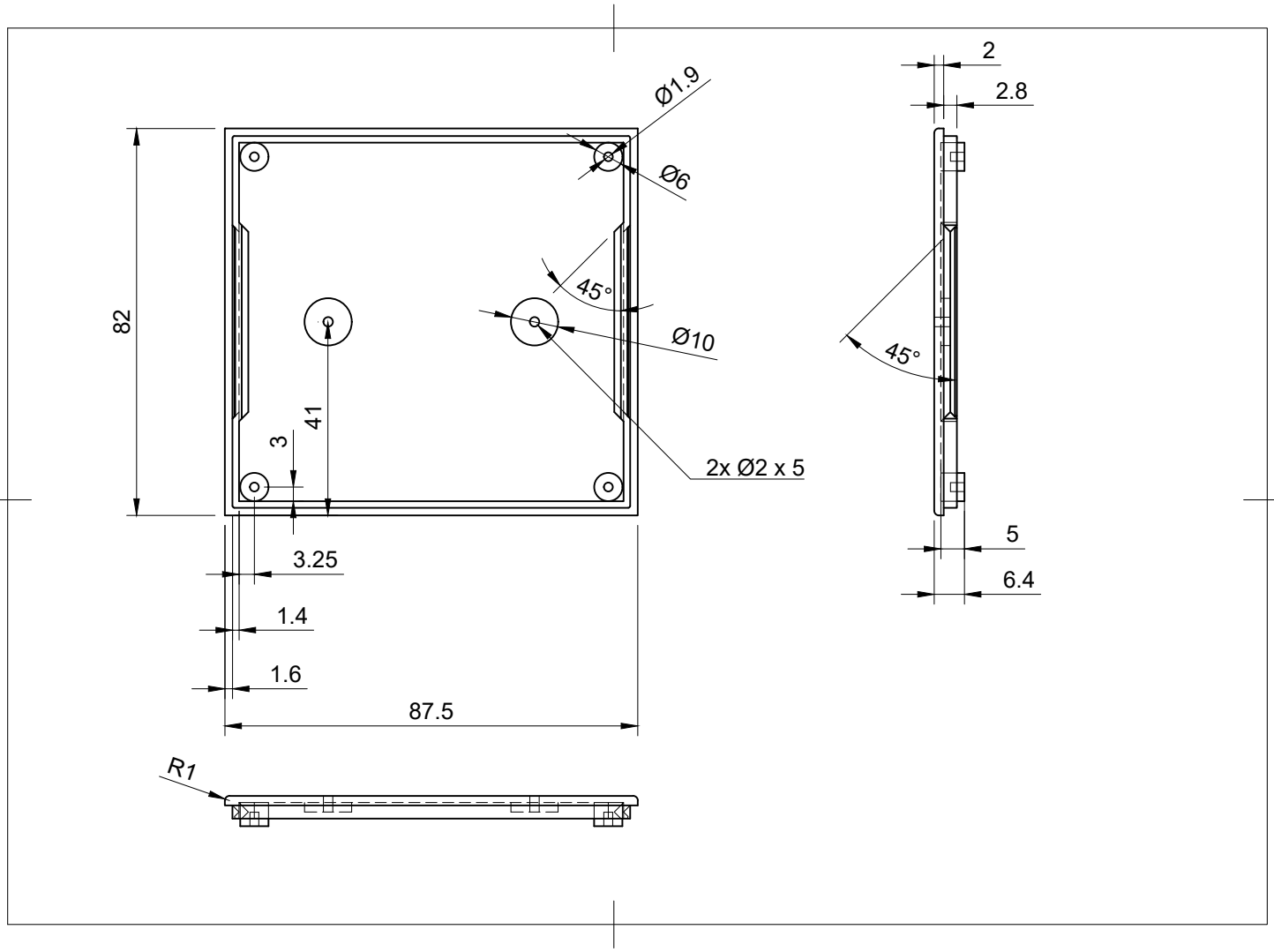
IP Adresa

IP adresa přidělená přístupovým bodem: 192.168.158.163

E Výkresy krabičky

Výkres krabičky je zobrazen na následujících dvou stranách. Výkres je rozdělen na horní část a spodní část.


25.45
18.2
1.4
ϕ2
82
54.1
59.3
40.5
87.5
R2
57
R2
68.5
80
13.5
43


Ø1.9
Ø6
45°
Ø10
2x Ø2 x 5
82
41
3
3.25
1.4
1.6
87.5
R1
2
2.8
45°
5
6.4