



Zadání bakalářské práce

Název:	Řídicí jednotka pro vzdálenou správu domácnosti s ESP32
Student:	Zdeněk Muzika
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

- 1) Prozkoumejte existující řešení.
- 2) Navrhněte vlastní řešení řídicí jednotky pro vzdálenou správu elektrických zařízení v domácnosti.
- 3) Řídicí jednotka bude splňovat tyto požadavky:
 - možnost přístupu k jednotce pomocí pevného i bezdrátového spojení,
 - jednotku bude možné ovládat z internetu,
 - jednotka bude obsahovat displej a tlačítka pro možnost nastavení základních parametrů,
 - jednotka bude umožňovat ovládat zařízení v domácnosti s pomocí relé, sériové linky a rozšiřujícího portu,
 - jednotka bude schopna zjistit aktuální stav externích zařízení s pomocí sériové linky a rozšiřujícího portu.
- 4) Nevržené řešení zrealizujte.
- 5) Pro výsledné zařízení napište obslužný program umožňující přístup k řídicí jednotce přes webové rozhraní.
- 6) Výsledné řešení řádně otestujte.

Bakalářská práce

ŘÍDICÍ JEDNOTKA PRO VZDÁLENOU SPRÁVU DOMÁCNOSTI S ESP32

Zdeněk Muzika

Fakulta informačních technologií
Katedra číslicového návrhu
Vedoucí: Ing. Pavel Kubalík, Ph.D.
10. května 2022

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Zdeněk Muzika. Odkaz na tuto práci.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci: Muzika Zdeněk. *Řídicí jednotka pro vzdálenou správu domácnosti s ESP32*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratk	x
Úvod	1
1 Cíle práce	3
2 Existující řešení	5
2.1 Komerční řešení	5
2.1.1 Immax NEO BRIDGE PRO Smart Zigbee 3.0 v2	5
2.1.2 Philips Hue Bridge	5
2.1.3 Tesla Smart ZigBee Hub	6
2.1.4 Xiaomi Mi Smart Home Hub	6
2.1.5 Porovnání komerčních řešení	7
2.2 Nekomerční řešení	7
2.2.1 Zigbee systém pro domácí automatizaci	8
2.2.2 Systém pro domácí automatizaci na bázi GSM	9
2.2.3 Systém pro domácí automatizaci využívající Bluetooth	9
2.2.4 Shrnutí nekomerčních řešení	9
3 Analýza	11
3.1 Mikropočítač	11
3.2 Rozšiřovač portů	12
3.3 Displej	12
3.4 Tlačítka	13
3.5 Měření teploty	14
3.6 Relé	14
3.7 Obvod reálného času	15
3.8 Napájení obvodu	15
3.9 Sběrnice pro komunikaci mezi moduly	16
3.9.1 SPI – Serial Peripheral Interface	17
3.9.2 I2C – Inter-Integrated Circuit	17
4 Návrh	19
4.1 Návrh propojení modulů	19
4.2 Vývojové prostředí	19
4.3 Návrh programu	19

5	Realizace jednotky	21
5.1	Zapojení	21
5.1.1	Mikropočítač WT32-ETH01	21
5.1.2	Rozšiřovač portů CJMCU-2317 a tlačítka	21
5.1.3	Displej OLED SH1106	23
5.1.4	Senzor teploty a vlhkosti DHT11	23
5.1.5	Relé modul	24
5.1.6	RTC modul PCF8563	24
5.1.7	Napájení obvodu	24
5.2	Plošný spoj	25
5.2.1	Návrh desky	25
5.2.2	Tvorba desky	26
6	Realizace programu	27
6.1	Zapojení WT32-ETH01 pro programování	27
6.2	Použité knihovny	27
6.3	Struktura programu	28
6.3.1	Hlavní program	29
6.3.2	Uživatelské rozhraní	29
6.3.3	Obrazovky	29
6.3.4	Rozšiřovač portů	30
6.3.5	Měření teploty	31
6.3.6	Obvod RTC	31
6.3.7	Připojení	31
6.3.8	Server	32
6.3.9	Paměť	33
7	Testování	35
7.1	Testování modulů	35
7.2	Testování připojení	35
7.2.1	Problém s ethernetovým připojením	36
7.2.2	Problém s vyřizováním HTTP požadavků	37
7.3	Testování provozu	37
8	Závěr	39
	Obsah přiloženého média	43

Seznam obrázků

2.1	Immax NEO BRIDGE PRO Smart Zigbee 3.0 v2 [1]	6
2.2	Philips Hue Bridge [2]	6
2.3	Tesla Smart ZigBee Hub [4]	7
2.4	Xiaomi Mi Smart Home Hub [5]	7
2.5	Prototyp ZigBee systému pro domácí automatizaci [8]	8
3.1	Vývojový kit ESP32-DevKitC ve verzi s 38 piny	11
3.2	Mikropočítač WT32-ETH01 s čipem WT32-S1	12
3.3	Rozšiřovací modul CJMCU-2317 s čipem MCP23017	13
3.4	1,3"OLED s čipem SH1106	14
3.5	Tlačítko TACTS-24R-F [20]	14
3.6	Senzor teploty a vlhkosti DHT11	15
3.7	5V relé modul se čtyřmi kanály	15
3.8	Obvod reálného času s čipem PCF8563	16
4.1	Blokové schéma komunikace mezi moduly	20
5.1	Schéma zapojení WT32-ETH01	22
5.2	Schéma zapojení CJMCU-2317	22
5.3	Schéma zapojení spínače na digitální vstup	23
5.4	Schéma zapojení displeje	23
5.5	Schéma zapojení senzoru DHT11	24
5.6	Schéma zapojení relé modulu	24
5.7	Schéma zapojení RTC modulu PCF8563	25
5.8	Konceptuální schéma zapojení napěťových regulátorů	25
5.9	Návrh desky plošného spoje v software Eagle	26
5.10	Osazená deska plošného spoje	26
6.1	Ukázka zapojení USB-TTL převodníku pro programování WT32-ETH01	28
6.2	Obrazovka s hlavním menu	30
6.3	Obrazovka pro zobrazení statusu jednotky	30
6.4	Obrazovka pro zadávání řetězcového vstupu	30
6.5	Webová stránka jednotky zobrazená v mobilním telefonu	32
7.1	Zapojení s funkční ethernetovou komunikací	36

Seznam tabulek

2.1	Srovnání komerčně dostupných centrálních jednotek pro chytré domovy	8
3.1	Porovnání displejů	13
3.2	Spotřeba použitých 3,3 V modulů	16
6.1	Zapojení pinů pro programování WT32-ETH01	27

Své vřelé poděkování bych chtěl věnovat především svému vedoucímu Ing. Pavlu Kubalíkovi za jeho rady a pomoc při tvorbě této práce. Dále bych chtěl poděkovat za psychickou podporu své rodině a přátelům.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 10. května 2022

.....

Abstrakt

Tato bakalářská práce se zabývá návrhem a vytvořením prototypu jednotky pro vzdálenou správu domácnosti za použití mikropočítače z rodiny ESP32. V práci jsou také prozkoumány existující komerční a nekomerční řešení pro vzdálenou správu domácnosti. Výsledkem této práce je prototyp, který funguje jako server připojený do domácí sítě přes WiFi nebo ethernet a umožňuje uživateli ovládat zařízení v domácnosti přes webové rozhraní.

Klíčová slova internet věcí, chytrá domácnost, domácí automatizace, řídicí jednotka, ESP32, Arduino, PlatformIO

Abstract

This bachelor thesis deals with design and creation of a prototype of a remote home control unit using microcomputer from the ESP32 family. The work also explores existing commercial and non-commercial solutions for remote home control. The result of this work is a prototype, which works as a server, connected to the home network using WiFi or ethernet and allows the user to control devices at home through a web interface.

Keywords internet of things, smart home, home automation, control unit, ESP32, Arduino, PlatformIO

Seznam zkratek

DHCP	Dynamic host configuration protocol
GPIO	General-purpose input/output
GSM	Global system for mobile communications
HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
I2C	Inter-integrated circuit
IP	Internet protocol
IPv4/IPv6	Internet protocol version 4/6
IoT	Internet of things
LCD	Liquid crystal display
NTC	Negative temperature coefficient
NVRAM	Non-volatile random access memory
OLED	Organic light-emitting diode
RGB	Red-green-blue
RTC	Real time clock
SPI	Serial peripheral interface
SSL	Secure socket layer
TFT	Thin-film transistor
TTL	Transistor-transistor logic
UART	Universal asynchronous receiver-transmitter
USB	Universal serial bus

Úvod

Naše domovy jsou místem pro odpočinek a pohodlí. Není tak divu, že se z nich lidé odjakživa snaží vytvořit co nejpříjemnější prostředí ke každodennímu životu. V dávných dobách si udržovali oheň v kamnech a navečer rozsvěcovali svíce. V současnosti pouze stačí na termostatu nastavit teplotu a zapnout osvětlení nástěnným vypínačem. Dnes ale už není žádnou novinkou, aby se tyto činnosti dělaly pouhým klepnutím na chytrém telefonu, nebo i zcela automaticky pomocí senzorů či časových spínačů. Tyto chytré domácnosti dokáží dále zjednodušit a zefektivnit správu našich obydlí.

Tato práce se zabývá návrhem a vytvořením centrální jednotky pro vzdálenou správu domácnosti. Výstupem práce by měl být funkční prototyp sestavený z modulů, vhodný k rozšíření navazujícími projekty.

Práce je strukturována do teoretické části (kapitoly Existující řešení, Analýza a Návrh) a praktické části (kapitoly Realizace jednotky, Realizace programu, a Testování). V teoretické části budou prozkoumány principy a existující řešení pro vzdálenou správu domácnosti, nalezeny vhodné součástky nebo moduly a vytvořen návrh základní struktury prototypu jednotky. Praktická část se bude věnovat vytvoření prototypu na základě analýzy a návrhu, tvorbě obslužného programu a otestování funkčnosti vytvořeného prototypu.

Kapitola 1

Cíle práce

Cílem práce je vytvořit prototyp centrální jednotky pro vzdálenou správu, který bude ovládán přes internet, přičemž základní nastavení bude možno provádět pomocí tlačítek a displeje jednotky. Prototyp bude založen na platformě ESP32, resp. na modulu využívající tuto platformu. Komunikace s prototypem by měla být možná jak přes drátové, tak bezdrátové připojení. Výsledek práce by měl fungovat jako server, na který se lze z internetu připojit a ovládat připojená zařízení. Obslužný program jednotky by měl demonstrovat funkčnost použitých modulů a součástek a umožňovat základní ovládání a nastavování.

Cílem kapitoly Existující řešení je prozkoumat existující řešení pro vzdálenou správu domácnosti a porovnat jejich vlastnosti a funkce.

Cílem kapitoly Analýza je vybrat vhodné součástky a moduly, které budou součástí jednotky a popsat technologie používané těmito moduly.

V kapitole Návrh je cílem vytvořit základní strukturu a blokový návrh jednotky. Bude také popsán postup pro vytváření struktury programu.

Cílem kapitoly Realizace jednotky je vytvořit vhodná zapojení součástek/modulů, které nakonec budou integrovány do desky plošného spoje vytvořeného prototypu.

Cíl kapitoly Realizace programu je popsat strukturu a funkce vytvořeného obslužného programu. Také bude popsáno, jaké knihovny byly v programu použity.

Cíl kapitoly Testování je odzkoušet a otestovat funkčnost sestaveného prototypu.

Kapitola 2

Existující řešení

Pro návrh prototypu je vhodné prozkoumat již existující řešení pro vzdálenou správu domácnosti, čímž se zabývá tato kapitola. Prozkoumána budou komerční i nekomerční řešení. U existujících řešení budou hlavně sledovány používané komunikační technologie pro komunikaci s ostatními částmi systému a pro vzdálený přístup.

2.1 Komerční řešení

Na trhu jsou dostupné řady různých zařízení pro domácí automatizaci, která se mohou značně lišit nabízenou funkcionalitou. K dispozici jsou jak rozsáhlé systémy, vyžadující centrální jednotku a umožňující propojení s celou řadou jiných technologií (např. s hlasovými asistenty), tak i samostatně fungující jednoduchá zařízení jako jsou například chytré zásuvky, meteostanice, kamery apod. Mnoho těchto systémů využívá pro komunikaci bezdrátové připojení, jako je například WiFi, Bluetooth nebo ZigBee.

Z komerčních řešení budou hlavně prozkoumány a porovnány samostatné centrální jednotky, které jsou běžně dostupné v obchodech a e-shopech.

2.1.1 Immax NEO BRIDGE PRO Smart Zigbee 3.0 v2

Jedná se o centrální jednotku chytré domácnosti od společnosti Immax (viz obrázek 2.1). Jednotka dokáže komunikovat přes protokoly WiFi a ZigBee. Funguje pouze jako centrální jednotka a tudíž potřebuje ostatní zařízení, skrze které může domácnost ovládat. Dokáže také spravovat zabezpečovací systém domova, například bezpečnostní kamery, alarm nebo zámky dveří. Uživateli je k dispozici aplikace Immax NEO PRO na mobilní telefon, pomocí které se může na jednotku připojit. [1]

2.1.2 Philips Hue Bridge

Centrální jednotka pro systém chytrého osvětlení Philips Hue (na obrázku 2.2) dokáže ovládat 50 připojených světelných zařízení a 10 dalších příslušenství. Ke komunikaci se zařízeními používá protokol ZigBee, přičemž samotná jednotka se připojuje pomocí ethernetového kabelu k domácímu routeru. Stejně jako předchozí představené zařízení nemá samostatnou funkci a slouží pouze jako srdce systému. Pro ovládání pomocí telefonu je k dispozici aplikace Hue. [2]



■ **Obrázek 2.1** Immax NEO BRIDGE PRO Smart Zigbee 3.0 v2 [1]



■ **Obrázek 2.2** Philips Hue Bridge [2]

2.1.3 Tesla Smart ZigBee Hub

Centrální jednotka od společnosti Tesla (na obrázku 2.3) taktéž používá protokol ZigBee pro komunikaci se zařízeními v chytré domácnosti a taktéž je určena pouze pro účel správy ostatních zařízení. K domácí síti se připojí pomocí WiFi připojení. K ovládání přes telefon slouží aplikace Tesla Smart. [3]

2.1.4 Xiaomi Mi Smart Home Hub

Zařízení od firmy Xiaomi (obrázek 2.4) dokáže komunikovat s ostatními přístroji nejen pomocí ZigBee, ale taky pomocí připojení Bluetooth 5.0. Do domácí sítě se zařadí pomocí WiFi a ovládat lze opět pomocí aplikace na telefon: Mi Home. [5]



■ **Obrázek 2.3** Tesla Smart ZigBee Hub [4]



■ **Obrázek 2.4** Xiaomi Mi Smart Home Hub [5]

2.1.5 Porovnání komerčních řešení

Tabulka 2.1 srovnává představená zařízení. Cenově jsou jednotky ve stejné kategorii, až na Philips Hue Bridge, který je dražší než ostatní. Z porovnání je také patrné, že jsou primárně používány bezdrátové technologie (hlavně ZigBee), především pro komunikaci s ostatními prvky chytré domácnosti. Důvodem je pravděpodobně jednoduchost a cena, jelikož není nutné mezi zařízeními tvořit kabelové propojení, což ocení i uživatel. Limit připojených zařízení se pohybuje v několika desítkách, přičemž jednotka značky Philips umožňuje propojení hlavně se světelnými zařízeními stejné značky (celkem 50) a limit pro ostatní zařízení je pouze 10. Většina zařízení podporuje spojení s hlasovými asistenty, jako jsou například Amazon Alexa nebo Google Home. Všechny porovnávané jednotky mají aplikaci pro mobilní telefon, která umožňuje jejich vzdálené ovládání.

2.2 Nekomerční řešení

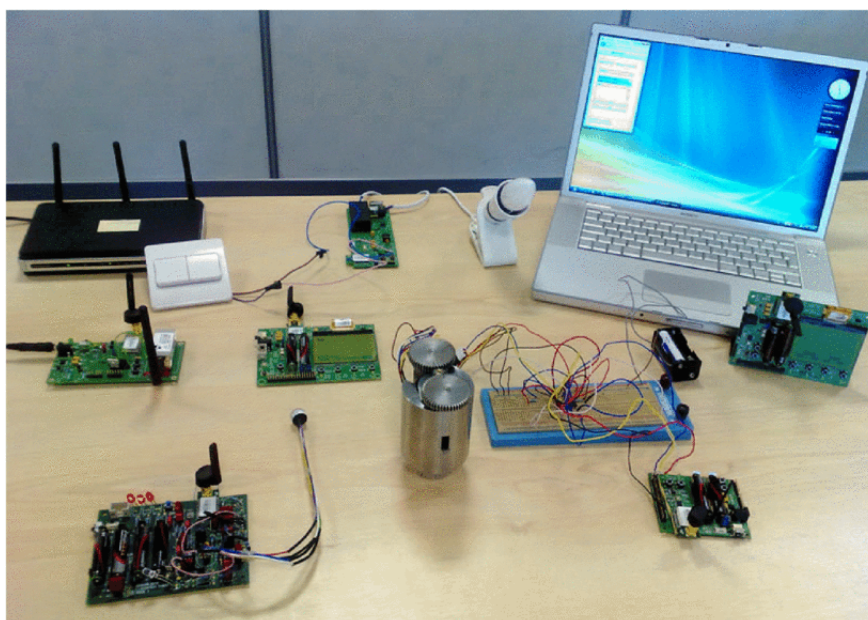
Nekomerční řešení pro vzdálenou správu a automatizaci domácnosti, obdobně jako ta komerční, využívají převážně bezdrátových technologií. Často se jedná o prototypovou výrobu, zaměřenou na výzkum domácí automatizace a IoT, ale lze také nalézt hobby projekty zaměřené na platformy Arduino, Raspberry PI, ESP32/ESP8266 apod.

	Immax Bridge	Philips Hue Bridge	Tesla Hub	Xiaomi Hub
Propojení do sítě	WiFi	Ethernet	WiFi	WiFi
Komunikace	WiFi, ZigBee	ZigBee	WiFi, ZigBee	WiFi, ZigBee, Bluetooth
Limit zařízení	neuveđen	50 světel, 10 ostatní	50	100
Aplikace pro telefon	Immax NEO PRO	Philips Hue	Tesla Smart	Mi Home
Podpora	Hlasoví asistenti	Hlasoví asistenti, osvětlení Philips Hue	Hlasoví asistenti	Apple HomeKit
Cena	699 Kč [1]	1 490 Kč [6]	799 Kč [4]	799 Kč [7]

■ **Tabulka 2.1** Srovnání komerčně dostupných centrálních jednotek pro chytré domovy

2.2.1 Zigbee systém pro domácí automatizaci

Jedno z prozkoumaných řešení se zabývá vytvořením systému založeného na protokolu ZigBee. Autoři v ní vytvořili prototyp systému (obrázek 2.5), tvořený čtyřmi rozdílnými periferními zařízeními, domácí bránou a řízený ZigBee koordinátorem. Síť ZigBee je řízená koordinátorem, který se stará o spuštění celé sítě. Pro tento účel vyhledá vhodného přenosového kanálu a poté vyčká na žádosti o připojení od ostatních zařízení. Komunikace se zařízeními je šifrovaná, aby byla zajištěna bezpečnost systému. Brána se stará o předávání dat mezi sítěmi WiFi a ZigBee. Jako periferní zařízení byl vytvořen ZigBee dálkový ovladač, vypínač světel, regulátor topení a bezpečnostní senzor, který zahrnuje prvky pro měření teploty a detekci kouře, ohně a oxidu uhelnatého. [8]



■ **Obrázek 2.5** Prototyp ZigBee systému pro domácí automatizaci [8]

2.2.2 Systém pro domácí automatizaci na bázi GSM

Další výzkumná práce se zabývá vytvořením jednoduchého prototypu systému pro ovládání zařízení v domácnosti. Systém využívá GSM modem, mikrokontrolér PIC a relé, přes která jsou připojena koncová zařízení. Ovládání relé je zajištěno pomocí mobilního telefonu posláním SMS zpráv s příkazy pro spínání a rozepínání relé a tudíž spouštění/vypínání k nim připojených zařízení. Komunikace mezi modemem a mikrokontrolérem je zařízená pomocí sériového rozhraní RS232, přes které mikrokontrolér posílá příkazy do GSM modemu. [9]

2.2.3 Systém pro domácí automatizaci využívající Bluetooth

Další nastudovanou prací byla práce zabývající se vytvořením systému využívajícím technologii Bluetooth k propojení se s telefonem uživatele. Systém pro ovládání využívá desku Arduino UNO, která je pomocí modulu HC-06 rozšířena o Bluetooth připojení. Obdobně jako předešlé práce je ke spínání domácích spotřebičů využito relé. Systém také disponuje ultrazvukovým senzorem pro měření vzdálenosti a kapacitní sensor pro měření vlhkosti půdy (pro měření pokojových rostlin). Komunikace přes mobilní telefon zajišťuje aplikace fungující jako Bluetooth terminál. Přes tento terminál jsou posílány příkazy pro Arduino, které je pak vykonává. [10]

2.2.4 Shrnutí nekomerčních řešení

Představené nekomerční systémy používají, obdobně jako komerční řešení, převážně bezdrátové technologie a komunikaci. Důvody jsou pravděpodobně stejné: jednoduchost a nižší cena. Bylo představeno i řešení využívající GSM technologii pro ovládání pomocí SMS. Vytvořené systémy využívají bezdrátové komunikační technologie převážně ke komunikaci s uživatelským zařízením nebo pro přístup do internetu. První představený systém využívající Zigbee pak takto komunikuje i se samotnými koncovými zařízeními v systému.

Kapitola 3

Analýza

Tato kapitola se zabývá průzkumem a výběrem vhodných součástí a modulů pro sestavení jednotky. Vybrané součástky budou následně integrovány do návrhu prototypu. Také jsou v této kapitole popsány běžně používané technologie/sběrnice pro komunikaci mezi moduly.

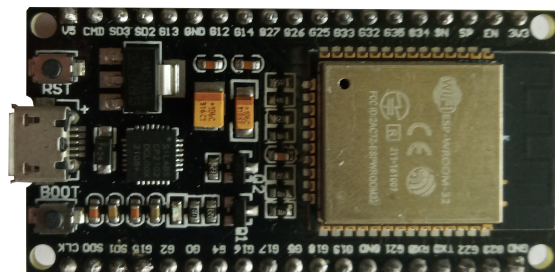
Výběr součástek bude prováděn na základě požadavků jednotky. Součástky budou vybírány na základě jejich funkcí a vzájemné interoperability. Pro odhad výsledné spotřeby všech modulů bude také sledována spotřeba proudu, která pak bude zohledněna při výběru napájecích obvodů.

3.1 Mikropočítač

Mikropočítač bude v jádru celé jednotky. Jeho úkolem je ovládat a spravovat všechny periférie a moduly. Pro tento účel mají mikropočítače k dispozici řadu ovladatelných vstupně-výstupních pinů (GPIO) a sběrnic. Některá zařízení dokáží komunikovat prostřednictvím bezdrátové sítě a značně tak ulehčit propojování s ostatními moduly nebo zařízeními. Rozsah funkcí a výkon mikropočítače je závislý převážně na ceně a velikosti.

Při výběru mikropočítače bylo zvaženo několik možností z rodiny ESP32. Hlavní výhodou těchto mikropočítačů je vestavěná bezdrátová konektivita prostřednictvím WiFi a Bluetooth. Další výhodou je také dostupnost za relativně nízkou cenu. Na trhu jsou dostupné řady vývojových kitů jako například ESP32-DevKitC ve verzi s 38 piny (na obrázku 3.1). Tyto vývojové kity často integrují USB konektory pro programování a komunikaci přes osobní počítač.

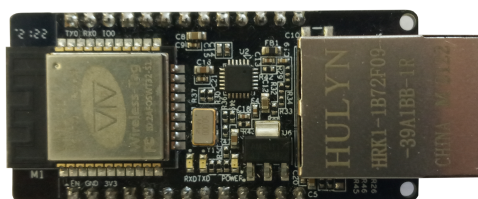
Pro řízení jednotky byl zvolen mikropočítač WT32-ETH01 (obr. 3.2), který je osazen čipem WT32-S1 z řady ESP32. Připojení do bezdrátové sítě je možné pomocí vestavěné 2,4 GHz WiFi nebo přes Bluetooth. Výhodou WT32-ETH01 je, že integruje čip LAN8720, který umožňuje



■ Obrázek 3.1 Vývojový kit ESP32-DevKitC ve verzi s 38 piny

konektivitu do ethernet sítě. Pro připojení kabelem je na mikropočítači vestavěný RJ45 konektor. Mikropočítač má 9 GPIO pinů, nicméně několik z nich je rezervováno pro boot nebo programování a některé jsou pouze vstupní. Výsledný počet dostupných pinů je proto vcelku nízký, v důsledku toho bude výhodné využít moduly a součástky, co komunikují po některé z běžně rozšířených sériových sběrnic, například I2C¹ nebo SPI² (tyto sběrnic jsou podrobněji popsány dále v této kapitole). Doporučený proud zdroje je alespoň 500 mA, přičemž průměrný provozní proud je 80 mA.[11]

Mikropočítač pracuje na napěťové úrovni 3,3 V, i když umožňuje externí napájení i na 5 V, to je nutné vzít v úvahu při komunikaci s dalšími moduly na desce. Pokud by bylo nutné komunikovat s modulem, který funguje pouze na 5 V, je potřeba využít převodník napěťových úrovní. Při volbě modulů/součástek se však bude tato práce snažit vybrat především součástky/moduly fungující na napětí 3,3 V.



■ **Obrázek 3.2** Mikropočítač WT32-ETH01 s čipem WT32-S1

3.2 Rozšiřovač portů

Jelikož WT32-ETH01 nedisponuje dostačujícím množstvím vstupně-výstupních pinů, je nutné do návrhu zahrnout rozšiřovač portů. Pro tento účel byl zvolen modul CJMCU-2317, který disponuje čipem MCP23017 (obr. 3.3). Výhodou tohoto modulu je možnost pracovat na napětí 3,3 V, a také schopnost komunikovat po sběrnici I2C – pro tento účel umožňuje externě nastavit adresu (3 adresní bity pro celkem 8 možností). Čip MCP23017 má 2 porty, které dohromady disponují 16 rozšiřujícími piny. Maximální proud rozšiřujícími piny je 25 mA přičemž maximální celkový proud je 125 mA. [12]

3.3 Displej

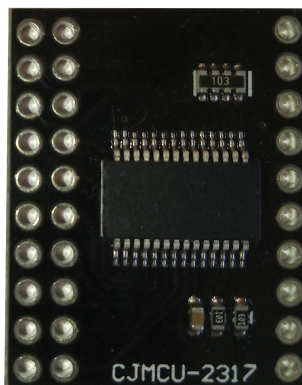
Zvolený displej by v ideálním případě měl komunikovat po sériové sběrnici, aby se omezil počet potřebných vodičů. Vykreslování grafiky by mělo být možné bez omezení po celé ploše. Displej by ideálně měl fungovat na napětí 3,3 V, aby mohla být snadno zřízena komunikace po sériové sběrnici.

Pro výběr displeje bylo porovnáno několik možností: LCD znakový displej, TFT LCD a OLED displej. Z každé kategorie byl vybrán jeden komerčně dostupný zástupce pro porovnání v tabulce 3.1. Principy fungování OLED a LCD displejů jsou popsány níže.

LCD (liquid crystal display) Displeje založené na technologii kapalných krystalů, které reagují na elektrické napětí. Vyvolaná reakce mění optické vlastnosti kapalných krystalů a umožňuje je precizně ovládat. Podsvícením těchto krystalů je poté možné generovat obraz.[13]

¹Inter-Integrated Circuit, někdy také IIC, nebo I²C

²Serial Peripheral Interface



■ **Obrázek 3.3** Rozšiřovací modul CJMCU-2317 s čipem MCP23017

	OLED SH1106 [15]	TFT ILI9341 [16]	LCD2004 [17, 18]
Typ	OLED	LCD TFT	LCD
Úhlopříčka	1,3"	2,2"	3,52"
Rozlišení	128 × 64 pixelů	240 × 320 pixelů	4 × 20 znaků (5 × 8 pixelů na znak)
Barva	bílá	RGB	bílá; modré podsvícení
Rozhraní	I2C (dostupné i verze s SPI)	SPI	I2C (s převodníkem)
Napájecí napětí	3,3–5 V	3,3 nebo 5 V	5 V
Přibližná cena	150 Kč	450 Kč	200 Kč

■ **Tabulka 3.1** Porovnání displejů

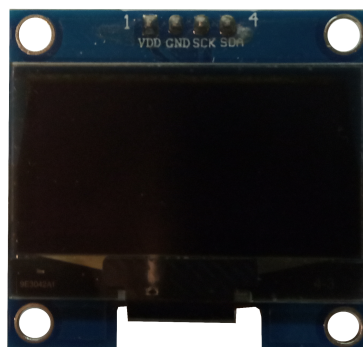
OLED (organic light emitting diode) Tyto displeje jsou tvořeny vrstvením tenkého organického materiálu mezi dvě elektrody. Pokud je na tyto elektrody aplikováno elektrické napětí, reaguje organická vrstva vyzařováním světla. Není nutné podsvícení, jelikož body OLED displeje světlo samy vyzařují. Oproti LCD displejům lze OLED displeje vytvořit v mnohem tenčím provedení. [14]

Z porovnávaných displejů byl vybrán OLED displej s čipem SH1106 (viz obr. 3.4). Oproti znakovému displeji LCD2004 není omezen pouze na samostatné znaky, ale umožňuje nastavovat každý pixel zvlášť. Výhodou proti TFT ILI9341 je podpora komunikace po sběrnici I2C a nižší cena. Nevýhodou je relativně malá uhlopříčka 1,3", proto bude nutné při tvorbě uživatelského rozhraní nutně brát zřetel na čitelnost údajů zobrazovaných na displeji. Běžný provozní proud OLED SH1106 displeje je cca 30 mA [19], ale odvíjí se od množství právě aktivních bodů na displeji.

3.4 Tlačítka

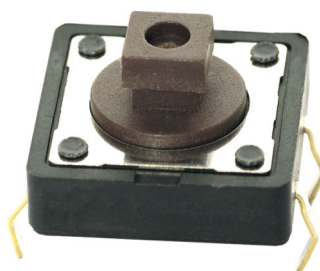
Manuální ovládání jednotky bude uživateli umožňovat klávesnice z tlačítek. Na trhu jsou nabízeny různé druhy tlačítek, lišící se svými rozměry, maximálním proudem, napětím atd. K dispozici jsou také tlačítkové klávesnicové matice různých formátů, které svá tlačítka adresují pomocí řádků a sloupců – tím dokáží omezit počet potřebných vodičů, i když je pak nutné programově tuto adresu správně dekodovat.

Pro uživatele by bylo vhodné mít tlačítek co nejvíce, nicméně naproti tomu je snaha navrháři omezit plochu desky plošného spoje a tudíž osazených součástí. Menší plocha desky je



■ **Obrázek 3.4** 1,3"OLED s čipem SH1106

méně nákladná a menší počet tlačítek je jednodušší na zapojení. Jako kompromis bylo zvoleno 5 tlačítek typu TACTS-24R-F (viz obr. 3.5). Daný typ tlačítek byl zvolen hlavně kvůli velikosti ($12 \times 12 \times 7,3$ mm). Menší tlačítka by mohla být nepříjemná na ovládání, naopak větší by zbytečně zvětšovala plochu desky.



■ **Obrázek 3.5** Tlačítko TACTS-24R-F [20]

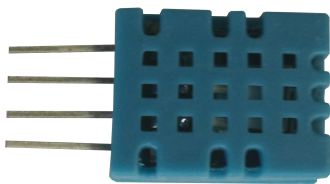
3.5 Měření teploty

Jednotka by měla být schopná provádět základní měření teploty. K tomuto účelu bude osazena senzorem teploty a vlhkosti DHT11 (na obr. 3.6). Senzor DHT11 měří teplotu pomocí tzv. NTC (negative temperature coefficient) termistoru, což je součástka s odporem závislým na teplotě, a pokud se teplota zvyšuje, pak se odpor této součástky snižuje [21]. Modul má rozsah měření teploty $0-50$ °C s přesností ± 2 °C a relativní vlhkosti $20-90$ % s přesností ± 5 %. [22]

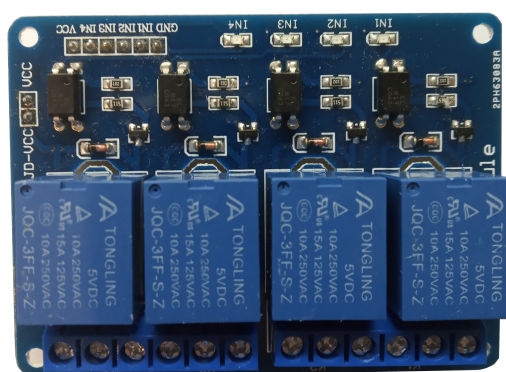
3.6 Relé

Pro ovládání spotřebičů v domácnosti je žádoucí, aby jednotka zvládla spínat napětí rozvodných sítí 230 V. K tomuto účelu se používá magneticky spínaných relé, kde je pomocí elektromagnetů rozpínán resp. spínán řízený obvod. V relé se nachází pružina, která drží spínač v poloze, kde uzavírá jeden obvod, působením magnetické síly se tento spínač přepne do druhé polohy a uzavře druhý obvod. [23]

Vybrán byl 5V relé modul se čtyřmi kanály (obr. 3.7). Osazená relé značky Tongling dokáží spínat střídavé napětí do 250 V s maximálním průchozím proudem 10 A. Stejnoseměrné napětí je možné spínat pouze do 30 V [24]. Je nutné také zajistit napájení cívek 5 V.



■ **Obrázek 3.6** Senzor teploty a vlhkosti DHT11



■ **Obrázek 3.7** 5V relé modul se čtyřmi kanály

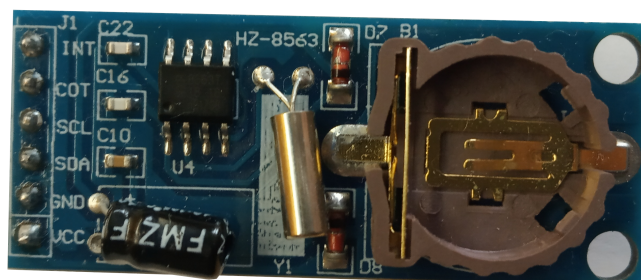
3.7 Obvod reálného času

Aby jednotka dokázala udržet aktuální čas i při ztrátě napájení, je nutné do návrhu zařadit modul reálného času (RTC – real time clock). Modul dokáže sledovat čas a datum, i pokud je jednotka vypojená od napájení, jelikož má integrovanou baterii. Čip tohoto modulu má velmi nízkou spotřebu, aby výdrž baterie byla dostatečná.

Vybraný modul reálného času používá čip PCF8563. Modul dokáže pracovat na napětí v rozmezí 1,0–5,5 V a komunikuje přes sběrnici I2C. Má k dispozici funkci budíku. Běžná spotřeba modulu je malá – v řádech mikroampér. Záložní baterií modulu je knoflíková baterie typu CR2032. [25]

3.8 Napájení obvodu

Pro napájení vybraných součástek bude třeba poskytnout stabilní napětí 3,3 V a 5 V. K regulaci napájecího napětí se používá lineárních nebo spínaných regulátorů napětí. Lineární regulátor napětí reaguje na spotřebu zátěže/napájeného obvodu a reguluje hodnotu výstupního napětí podle potřeby, tak aby bylo drženo na stabilní hodnotě. Spínaný regulátor napětí oproti tomu spíná napájecí napětí tak, aby bylo průměrně dosaženo požadované hodnoty. Rozdíly při sepnutí a rozepnutí jsou pak upravovány filtrem. Výhodou lineárního regulátoru je nízká cena a snazší zapojení. Spínaný regulátor má naopak složitější zapojení, ale dokáže dosáhnout velmi dobré efektivity. Regulátory se také dále dělí na pevné a nastavitelné. Pevné regulátory mají předem



■ **Obrázek 3.8** Obvod reálného času s čipem PCF8563

určenou hodnotu napětí, kterou na svém výstupu budou poskytovat. U nastavitelných regulátorů se naopak výstupní napětí dá v určitém rozsahu měnit.

Vybraný regulátor pro 3,3 V musí mít jmenovitý proud vyšší, než je spotřeba obvodu i s určitou rezervou. Tabulka 3.2 ukazuje spotřebu jednotlivých součástek a výsledný součet proudu. Uvažovaný je vždy nejvyšší možný proud (u WT32-ETH01 je to nejmenší doporučený proud zdroje [11]).

Napájení cívek relé modulu musí být zajištěno zvláště pomocí 5V regulátoru. Pro 4 relé je minimální proud zhruba 300 mA [24].

Dále je uvažováno, že vstupní napětí jednotky bude v rozmezí 9–12 V – napětí běžně poskytované stejnosměrnými napěťovými zdroji do zásuvky. Pro přívod vstupního napětí pomocí těchto zdrojů byl zvolen běžně rozšířený 5,5 × 2,1 mm souosý konektor.

	Spotřeba
WT32-ETH01	500 mA
CJMCU-2317	125 mA
OLED displej	30 mA
DHT11	<5 mA
RTC	<5 mA
Celkem	<665 mA

■ **Tabulka 3.2** Spotřeba použitých 3,3 V modulů

Pro napájecí napětí 5 V i 3,3 V byly vybrány pevné lineární regulátory. Napětí 5 V je regulováno pomocí LF50CV, který má maximální jmenovitý proud 1 A a vstupní napětí až 16 V. [26]. O regulaci napětí 3,3 V se stará LM1085IT-3.3 se jmenovitým proudem až 3 A a maximálním vstupním napětím 30 V [27]. Oba regulátory byly vybrány díky ceně a dostupnosti těchto součástek a s přehledem by měly pokrýt napájecí nároky jednotky.

3.9 Sběrnice pro komunikaci mezi moduly

Jednotlivé moduly a zařízení mohou s mikropočítačem komunikovat různými způsoby. Jedním z těchto způsobů je použití nízkorychlostních sériových sběrnic, na které se periferní zařízení připojují. Výhodou sériových sběrnic je malý počet vodičů potřebný pro komunikaci. Tato pod-

kapitola představuje dvě velmi rozšířené sběrnice: I2C a SPI. Tyto sběrnice jsou určeny pro přenos dat na velmi krátké vzdálenosti v rámci jednoho čipu, modulu nebo desky plošného spoje.

3.9.1 SPI – Serial Peripheral Interface

Čtyřvodičová sběrnice určená pro synchronní přenos dat. SPI obvykle mívá pouze jedno *master* zařízení, které spravuje komunikaci po sběrnici propojené s jedním a nebo více periferními *slave* zařízeními. Pojmenování vodičů se může lišit, ale velmi rozšířené jsou názvy: CS (*chip select* – výběr periferního zařízení), MISO (*master in slave out*), MOSI (*master out slave in*) a CLK (hodinový vodič). Vodič MISO je vstupem pro zařízení *master* a výstupem pro zařízení *slave*. Po vodiči MOSI jsou pak data posílána v opačném směru. [28]

Přenos dat je inicializován *master* zařízením, které generuje hodinový signál a přes CS vybírá periferní zařízení, se kterým chce komunikovat. Data jsou pak posílána po příslušných vodičích pro vstup/výstup. [28]

3.9.2 I2C – Inter-Integrated Circuit

Synchronní sběrnice vyvinutá firmou Philips, tvořená dvěma vodiči: SDA (*serial data* – datový vodič) a SCL (*serial clock* – hodinový vodič). Zařízení připojená na tuto sběrnici musí mít unikátní adresu, která je použita při komunikaci s nimi. I2C umožňuje obousměrnou komunikaci a sběrnice může fungovat jako *multimaster*. Existuje několik rychlostí, na kterých může sběrnice fungovat, nejvyšší obousměrná rychlost může být až 3,4 Mbit/s. Vodiče musí být v klidovém stavu udržovány v logické 1. [29]

Komunikace po I2C začíná startovací podmínkou (sestupná hrana na SDA a SCL na log. 1), poté probíhá vlastní přenos a nakonec je zakončeno ukončovací podmínkou (vzestupná hrana na SDA a SCL na log. 1). Přenos je pak tvořen adresou cílového zařízení, určením jestli se žádá o zápis nebo čtení a poté samotnými daty. Posílané bajty, stejně jako adresa, jsou průběžně potvrzovány přijímajícím zařízením. [29]

V této kapitole bude popsán návrh propojení modulů na desce a obslužného programu. Výsledné navrhované zařízení by mělo fungovat jako centrální jednotka pro vzdálené řízení domácnosti. K dispozici by měly být rozšiřující porty a také vyvedený sériový port. Jednotka se bude připojovat do domácí sítě pomocí WiFi nebo přes ethernet. Pro spínání spotřebičů připojených k jednotce budou využita relé.

4.1 Návrh propojení modulů

Pro komunikaci s moduly bude použita sběrnice I2C, kterou bude řídit mikropočítač. Na tuto sběrnici bude napojený displej, rozšiřovač portů a obvod RTC. Do rozšiřovače portů budou zapojena tlačítka jako vstupy a spouštění relé jako výstupy. Pro připojení dalších zařízení bude z WT32-ETH01 vyvedena sériová linka UART a z rozšiřovače portů zbylé nepoužité porty. Sensor pro měření teploty a vlhkosti DHT11 bude využívat samostatný vodič pro komunikaci s mikropočítačem. Blokové schéma navrženého datového propojení lze nalézt na obrázku 4.1.

Připojení do domácí sítě přes WiFi či ethernetový kabel bude zprostředkováno přímo mikropočítačem WT32-ETH01.

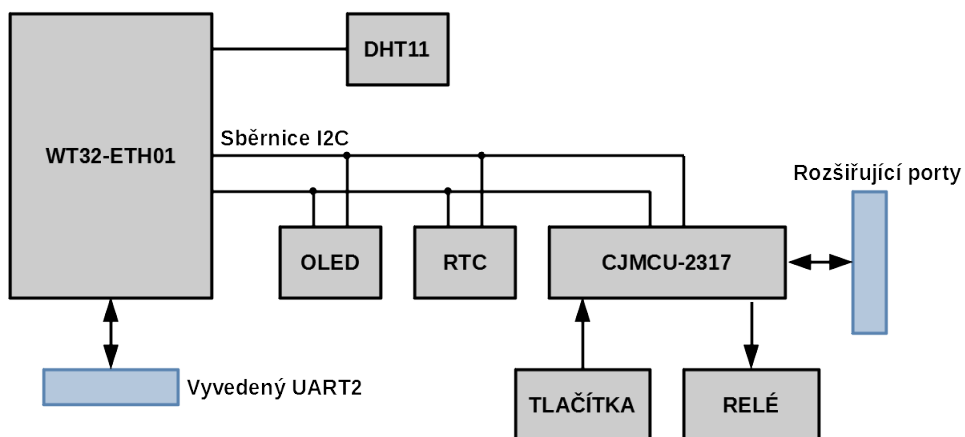
4.2 Vývojové prostředí

Obslužný program bude psán pomocí editoru Visual Studio Code s rozšířením PlatformIO, které umožňuje programovat širokou škálu různých typů mikropočítačů, jako jsou například Arduino, Raspberry PI, WeMos a především v této práci použitý ESP32. V PlatformIO jsou také k dispozici řady volně dostupných knihoven, určených pro ovládání různých modulů a zařízení, nebo jinak poskytující rozšířenou funkcionalitu.

Program bude používat Arduino framework pro systémy ESP32. K dispozici je také oficiální Espressif IoT development framework pro zařízení řady ESP32, který umožňuje komplexnější ovládání, avšak je méně používaný.

4.3 Návrh programu

Program bude obsahovat jednotlivé podprogramy, které budou zprostředkovávat dané funkcionality (například měření teploty, čtení tlačítek, získávání aktuálního času z RTC apod.). Tyto podprogramy budou spouštěny periodicky v hlavní smyčce s ohledem na to, jak často bude nutné je spouštět (např. měření teploty může být prováděno každých několik desítek sekund, obsluha



■ **Obrázek 4.1** Blokové schéma komunikace mezi moduly

čtení tlačítek zase musí proběhnout několikrát za sekundu). Hlavní smyčka samotná by měla sloužit převážně pro předávání dat mezi jednotlivými podprogramy.

Uživatelské rozhraní jednotky bude zprostředkováno pomocí displeje a tlačítek. Bude umožňovat provádění základního nastavení jednotky a také bude možné si zobrazit stav jednotky.

Program pro běh serveru na jednotce by měl zprostředkovávat zabezpečené připojení a také autorizaci uživatele. Na webové stránce serveru by pak mělo být možné zobrazit si aktuální naměřenou teplotu a vlhkost z jednotky a ovládat relé modul.

Realizace jednotky

Tato kapitola se věnuje vytvoření zapojení součástek/modulů a následně desky plošného spoje jednotky. Všechna zapojení jsou podrobně popsána a znázorněna na obrázcích schemat. Z vytvořených zapojení pak bude vytvořen návrh desky, na základě kterého bude následně deska vyrobena.

5.1 Zapojení

Při vytváření zapojení je třeba zohlednit všechny potřebné specifikace použitých modulů a součástek, zejména jestli jsou připojeny k odpovídajícímu napájecímu napětí. K většině modulů se také paralelně k napájení řadí tzv. blokovací kondenzátory. Jejich použitím je zajištěna stabilita napájecího napětí, jelikož kondenzátor „vyhlazuje“ napěťové špičky sítě. Často používané hodnoty blokovacích kondenzátorů jsou v rozsahu 10–100 nF.

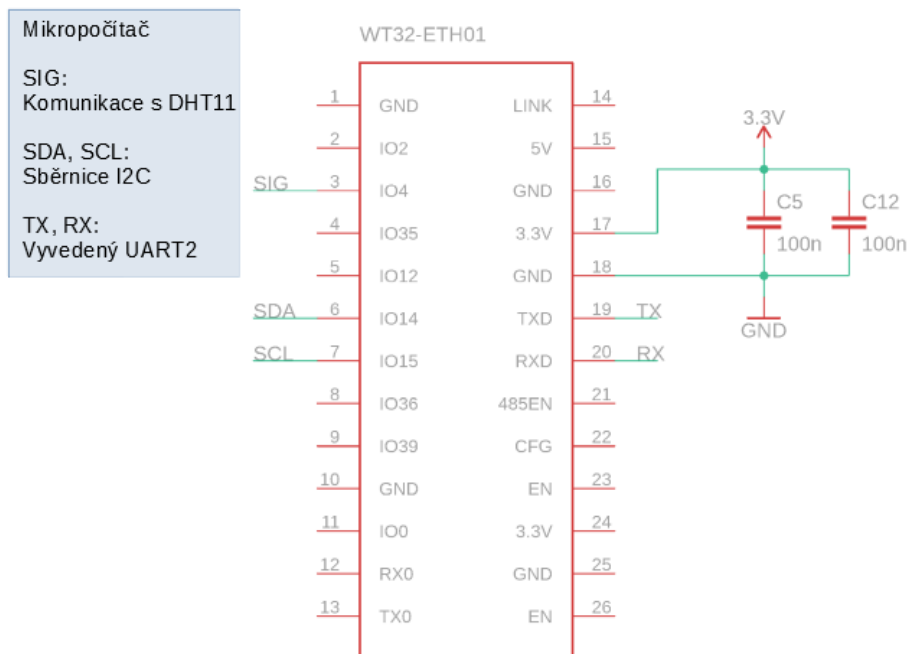
Schémata zapojení byla vytvořena pomocí programu Eagle. Pro některé základní součástky jako rezistory a kondenzátory byly dostupné schematické značky ze standardně poskytované knihovny, ale pro většinu modulů musely být vytvořeny náležitě funkční bloky.

5.1.1 Mikropočítač WT32-ETH01

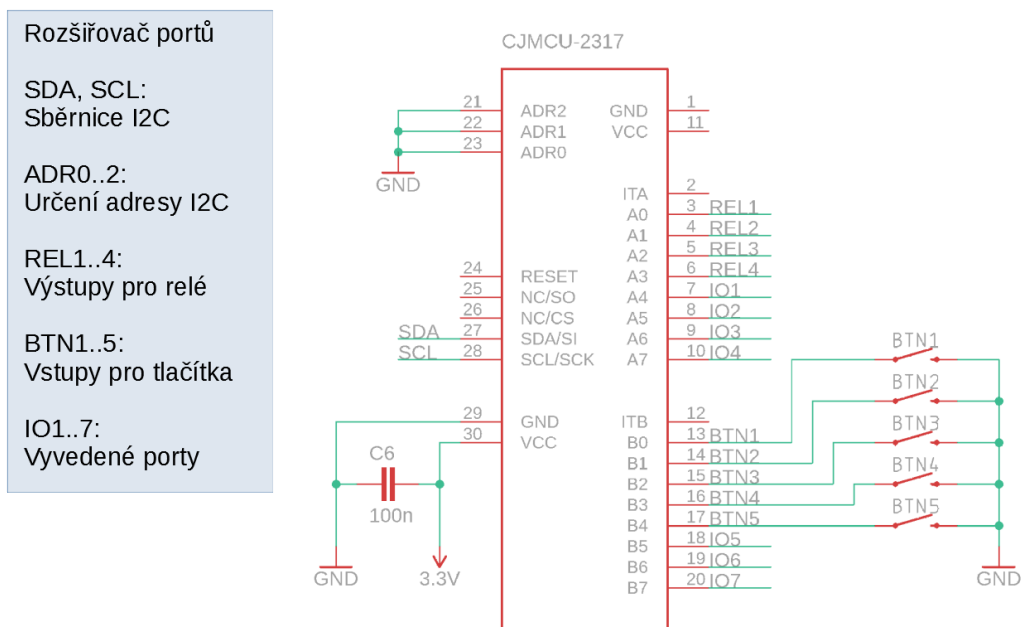
Zapojení mikropočítače WT32-ETH01 ukazuje schéma na obrázku 5.1. Napětí 3,3 V je přivedeno na jeden z několika dostupných napájecích pinů. Z mikropočítače je vyvedena sběrnice UART (piny TXD a RXD, jedná se o sériový port č. 2), která bude dostupná jako jeden z vstupů/výstupů desky. Dále je na pinech IO14 a IO15 vyveden datový a hodinový vodič (SDA a SCL) rozhraní I2C, pomocí kterého bude mikropočítač komunikovat s ostatními moduly. Posledním vyvedeným pinem je IO4, kde přes vodič SIG bude probíhat komunikace s teplotním senzorem DHT11. Ostatní piny mikropočítače nejsou použité a zůstávají nezapojené.

5.1.2 Rozšiřovač portů CJMCU-2317 a tlačítka

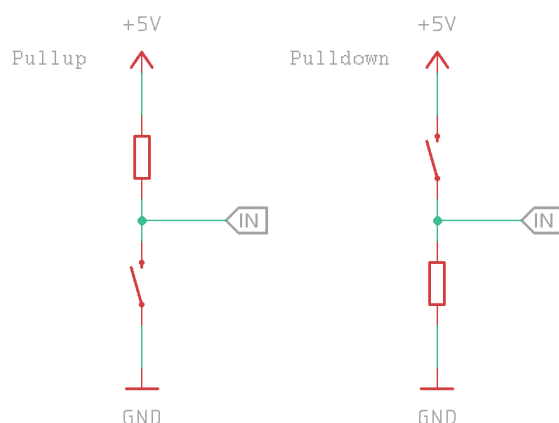
Rozšiřovač portů i se vstupními tlačítky (BTN1–BTN5) je zapojen podle schématu na obrázku 5.2. Přivedené napájecí napětí má hodnotu 3,3 V. Rozšiřovač je napojen na sběrnici I2C. Vstupy určující I2C adresu rozšiřovače jsou uzemněny – je tedy použita výchozí adresa. Z portu A rozšiřovače jsou vyvedeny vodiče A0–A3 jako výstupy řídicí relé. Port B má vyvedeny vodiče B0–B4 jako vstupy pro tlačítka. Zbytek vodičů portu A a B bude vyveden z desky pro rozšiřující vstupy/výstupy.



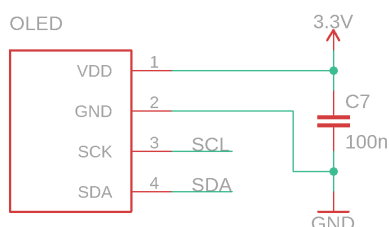
■ Obrázek 5.1 Schéma zapojení WT32-ETH01



■ Obrázek 5.2 Schéma zapojení CJMCU-2317



■ **Obrázek 5.3** Schéma zapojení spínače na digitální vstup



■ **Obrázek 5.4** Schéma zapojení displeje

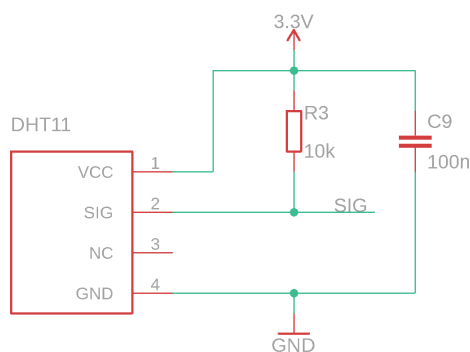
Tlačítka budou zapojena přes rozšiřovač portů jako vstupy. Pro zapojení tlačítek jako vstupu se běžně používá zapojení s rezistorem, který funguje buď jako pullup nebo pulldown. Pullup resp. pulldown zapojení se používá proto, aby na čteném vstupu byla jasně definována napěťová úroveň, i když je tlačítko/spínač rozpojený. Schéma na obrázku 5.3 znázorňuje obě zapojení. Pokud je spínač rozpojený, je vstup stále připojen k definované napěťové úrovni přes rezistor (pullup „vytahuje“ na log. 1, pulldown „stahuje“ na log. 0). Používané hodnoty rezistorů závisí na vlastnostech vstupního portu, časté hodnoty jsou 10–100 k Ω . Použitý rozšiřovač portů s MCP23017 integruje možnost pullup vstupu s hodnotou odporu 100 k Ω [12].

5.1.3 Displej OLED SH1106

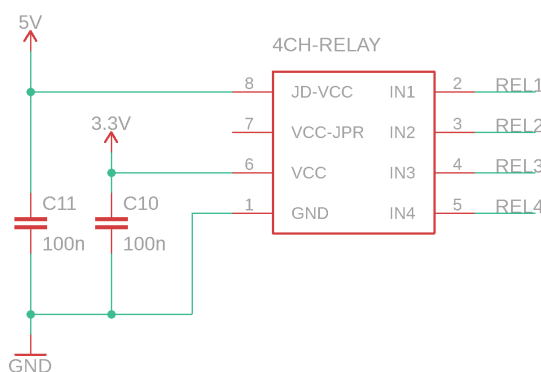
Modul s OLED displejem má jednoduché zapojení podle schématu na obrázku 5.4. Je zajištěno připojení k napájecímu napětí 3,3 V. Pro komunikaci s mikropočítačem po sběrnici I2C má vyvedeny piny SCK (jako SCL) a SDA.

5.1.4 Senzor teploty a vlhkosti DHT11

Schéma na obrázku 5.5 ukazuje zapojení DHT11. Senzor je zapojen do sítě s napájecím napětím 3,3 V. Vyveden je komunikační vodič SIG, který je napojen přímo na WT32-ETH01. SIG musí být připojen k napájecímu napětí v pullup zapojení – byl použit rezistor s hodnotou 10 k Ω . Vodič NC je na senzoru navíc a zůstává proto nezapojený.



■ Obrázek 5.5 Schéma zapojení senzoru DHT11



■ Obrázek 5.6 Schéma zapojení relé modulu

5.1.5 Relé modul

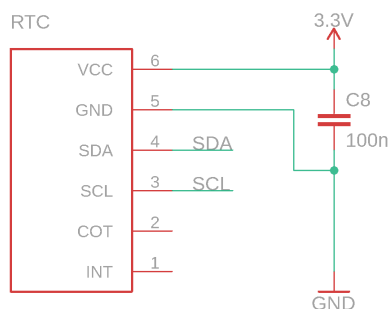
Relé modul je zapojen podle schématu na obrázku 5.6. Je zajištěno jak napájení cívek pomocí 5 V (pin JD-VCC), tak 3,3 V napájení spouštěcího obvodu. Vstupy pro ovládání relé REL1–REL4 jsou napojeny na rozšiřovač portů na porty A0–A3.

5.1.6 RTC modul PCF8563

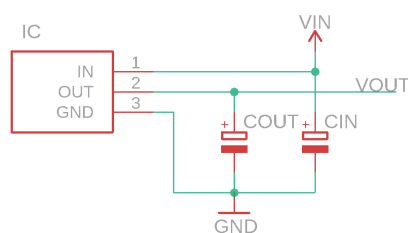
Schéma zapojení RTC modulu lze vidět na obrázku 5.7. Pro napájení modulu je přivedeno 3,3 V. Modul je také napojen na sběrnici I2C. Nevyužité piny pro funkci budíku a výstup hodinové frekvence jsou nezapojeny.

5.1.7 Napájení obvodu

Napěťové regulátory jsou zapojeny podle schéma na obrázku 5.8. Oba regulátory používají stejné zapojení. Vstup regulátorů je dále připojen na koaxialní konektor pro napájení jednotky. Na vstup a výstup regulátorů se umísťují kondenzátory (na schéma jako CIN a COUT), které napomáhají stabilizovat napětí jak na vstupu tak výstupu regulátoru. Použité tantalové kondenzátory mají hodnotu 10 μF .



■ **Obrázek 5.7** Schéma zapojení RTC modulu PCF8563



■ **Obrázek 5.8** Konceptuální schéma zapojení napěťových regulátorů

5.2 Plošný spoj

Pro propojení všech součástek a modulů byla vytvořena deska plošného spoje, která bude všechny moduly, součástky a jiné pomocné obvody integrovat.

Plán desky plošného spoje byl vytvořen na základě schémat v software Eagle. Stejně jako při vytváření schémat propojení musely být připraveny vlastní schématické značky, tak i zde bylo třeba vytvořit pro moduly vlastní stopy (rozložení pájecích plošek, potisků apod. daného modulu).

5.2.1 Návrh desky

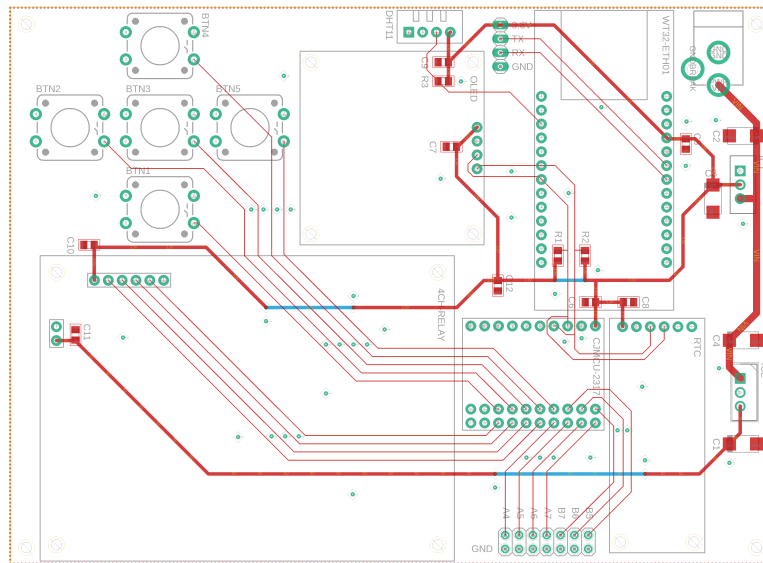
Výsledné rozložení desky ukazuje obrázek 5.9. Navržená deska má rozměry $10,2 \times 14$ cm a je vyrobena ve dvouvrstevném provedení, přičemž všechny součástky jsou pouze na vrchní straně. Při rozmísťování součástek byla snaha řadit veškeré vstupní a výstupní konektory jednotky na okraje desky. Tlačítka byla rozmístěna do kříže na volný kraj desky a displej byl umístěn přímo nad ně. Relé byla umístěna na kraj tak, aby do nich šly snadno zapojit přívodní kabely elektrických zařízení. Regulátory byly umístěny také na kraj desky, aby bylo vytvořeno místo pro případnou montáž chladiče. Blokovací kondenzátory modulů jsou umístěny co nejbližší k nim.

Pro napájecí rozvody byla použita větší šířka spojů, kvůli velkým proudům, které jimi mohou protékat. Přívodní napětí desky má šířku stopy 1,27 mm a rozvody napájení 5 a 3,3 V mají šířku necelých 0,61 mm. Ostatní rozvody mají šířku přibližně 0,15 mm.

Deska má také vrchní servisní potisk, zobrazující místa a označení všech součástek – pro usnadnění následného osazování a identifikaci součástek. Označeny jsou také vyvedené rozšiřující porty CJMCU-2317 a UART sběrnice mikropočítače.

Deska bude mít na svých rozích vyvrtané montážní otvory. Moduly budou také mít pod sebou k dispozici otvory pro uchycení distančními sloupky. Všechny tyto otvory mají jmenovitý průměr

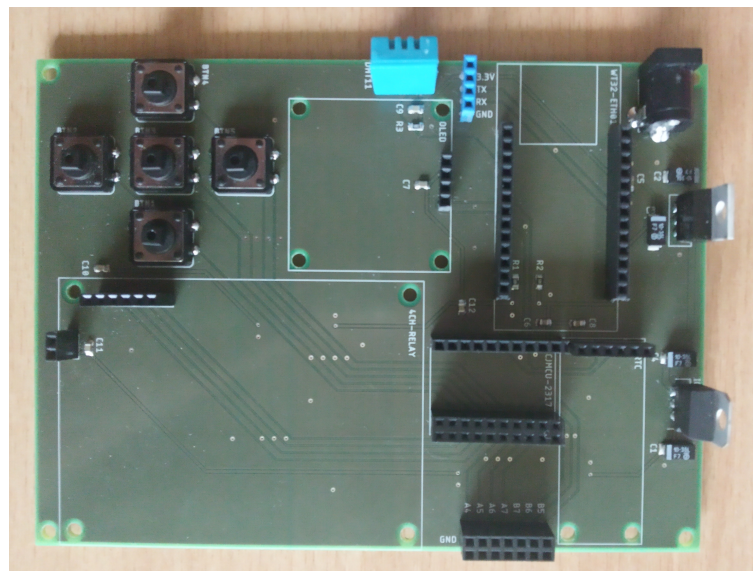
3 mm.



■ **Obrázek 5.9** Návrh desky plošného spoje v software Eagle

5.2.2 Tvorba desky

Navržený plošný spoj byl vyroben ve firmě Pragoboard s.r.o. za cenu 1 210 Kč. Na desku byly pro většinu modulů osazeny 0,1" dutinkové lišty (výjimkou je senzor DHT11, který byl osazen přímo na desku), které umožňují snadné odejmutí z desky pro případné testování, zkoušení nebo výměnu. Osazenou desku ukazuje obrázek 5.10.



■ **Obrázek 5.10** Osazená deska plošného spoje

Realizace programu

Tato kapitola se věnuje programové části jednotky. Následující podkapitoly popisují strukturu a přehled funkcionalit vytvořeného obslužného programu a použité externí knihovny. Také je uveden postup pro programování použitého mikropočítače.

6.1 Zapojení WT32-ETH01 pro programování

Použitý mikropočítač WT32-ETH01 nemá vestavěný USB port na programování, je proto nutné použít externí USB-TTL převodník. Zapojení pinů je popsáno v tabulce 6.1 a také ukázáno na obrázku 6.1. Komunikace po sériové lince je možná pouze na napětí 3,3 V, ale napájení mikropočítače je možné zajistit pomocí 5 V.

Pin WT32-ETH01	Pin USB-TTL převodníku	Poznámka
TX0	RX	sériový port UART 3,3 V
RX0	TX	sériový port UART 3,3 V
IO0	GND	nutné zapojit před napájením
GND	GND	
5V/3V3	VCC	zvolit podle napětí převodníku

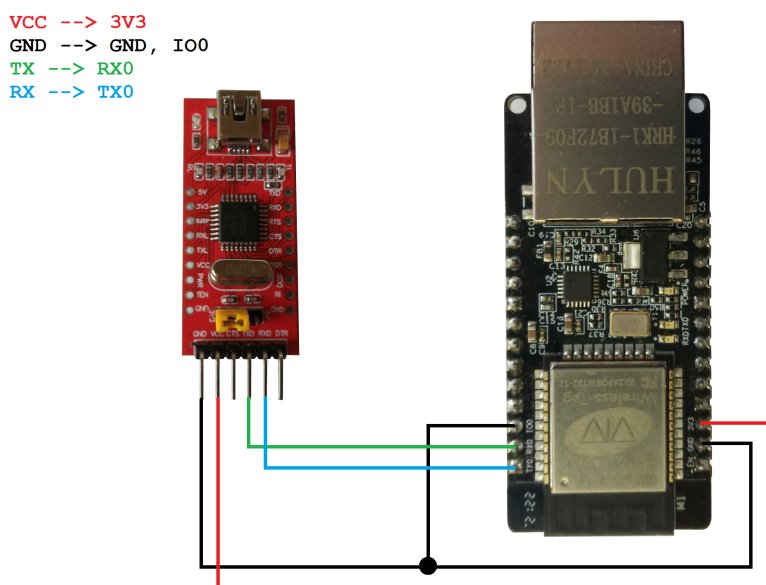
■ **Tabulka 6.1** Zapojení pinů pro programování WT32-ETH01

6.2 Použité knihovny

Pro ulehčení tvorby programu bylo využito několika knihoven, zjednodušující práci a komunikaci s moduly, zajišťující provoz serveru apod. V následujícím seznamu jsou popsány všechny použité pomocné knihovny a jaký je jejich účel a funkce. Nejsou zde uvedeny standardní knihovny Arduino frameworku pro ESP32.

Adafruit MCP23017 Arduino Library Knihovna pro práci s rozšiřovači portů MCP23017. Zajišťuje komunikaci po sériové sběrnici a následně umožňuje jednoduše ovládat použitý rozšiřovač.

DHT sensor library Tato knihovna byla vytvořena pro komunikaci se senzory pro měření teploty a vlhkosti DHT. Pomocí této knihovny lze snadno provádět a odečítat měření ze senzorů DHT11 nebo DHT22.



■ **Obrázek 6.1** Ukázka zapojení USB-TTL převodníku pro programování WT32-ETH01

U8g2 Knihovna pro ovládání široké škály displejů pomocí I2C nebo SPI rozhraní. Nabízí řadu funkcí pro vykreslování grafiky na displej a také integruje mnoho různých fontů pro vykreslování textu.

RTCLib Pomocí této knihovny lze snadno komunikovat s různými RTC čipy. Knihovna také implementuje některé pomocné struktury jako například datum a čas.

esp32_https_server Tato knihovna zprostředkovává obsluhu provozu serveru se zabezpečenou komunikací na mikropočítači. Pomocí knihovny lze detekovat příchozí požadavky od klienta a svázat je s určitou funkcí, která klientovi vrací odpověď. Dále je dostupná funkcionality pro validaci, autentizaci a autorizaci.

Pro správnou kompilaci knihovny *esp32_https_server* při použití Arduino frameworku pro ESP32 verze 3 nebo vyšší je nutné ve hlavičkovém souboru knihovny *HTTPConnection.hpp* změnit inkluzi zastaralé pomocné knihovny *<hwcrypto/sha.h>* na *<esp32/sha.h>* (řádek 9 v souboru). Dále je také nutné provést úpravy *HTTPConnection.hpp* a *HTTPConnection.cpp* podle pull requestu na GitHubu této knihovny: [30] (více v kapitole testování – podkapitola Problém s vyřizováním HTTP požadavků). Upravené soubory knihovny budou také součástí příloženého média (adresář extras).

6.3 Struktura programu

Tato podkapitola popisuje jednotlivé části programu a jejich účel. Pro vytvoření programu byly použity programovací jazyky C a C++ a Arduino framework pro ESP32. Podprogramy jsou děleny na hlavičkové soubory a soubory s implementací. Všechny vytvořené zdrojové kódy budou obsahem příloženého média.

6.3.1 Hlavní program

Hlavní program lze nalézt v souboru *main.cpp*. Úkolem hlavního programu je předávat data mezi ostatními částmi programu a spouštění programových smyček, které obsluhují moduly, uživatelské rozhraní atd.

setup Tato funkce slouží k inicializaci všech knihoven, modulů a proměnných používaných pro sledování smyček a výměnu dat.

loop V této smyčce jsou periodicky spouštěny ostatní podprogramy. Je zde obsluhováno uživatelské rozhraní, měření teploty a vlhkosti, aktualizace času a provozován server.

6.3.2 Uživatelské rozhraní

Podprogram pro správu uživatelského rozhraní je v souborech *user_interface.cpp* a *user_interface.h*. Stará se o správu uživatelského rozhraní, tedy ovládání displeje, přechod mezi jednotlivými obrazovkami a předávání dat k zobrazení, nebo pro výstup z podprogramu.

ui_init Funkce slouží k inicializaci uživatelského rozhraní a s ním spjatých modulů a knihoven.

ui_loop Smyčka uživatelského rozhraní, která se stará o přechody mezi obrazovkami a jejich správné zobrazování. Dále může předávat data do hlavní smyčky, pokud to je nutné (např. nastavení nového času, změna hesla atd.).

ui_update_status Funkce slouží k aktualizaci stavu (struktura status popsána v *status.h*) zobrazovaného na obrazovce status.

6.3.3 Obrazovky

Jednotlivé obrazovky a jejich typy lze nalézt v *screen.cpp* a *screen.h*. Je zde vytvořena pomocná rodičovská třída *Screen* pro všechny obrazovky, které jsou z ní následně odvozené. Tato část programu se také stará o vykreslování obrazovek pomocí grafické knihovny *u8g2*.

screen_init Funkce zajišťuje inicializaci knihovny *u8g2* a pomocných proměnných.

swap_screen Funkce nastavuje příznak pro zmrazení obrazovky. Zmražená obrazovka čeká, dokud nejsou uvolněna všechna tlačítka, až tehdy začne znovu reagovat na vstup. Toho je využíváno, pokud se mění obrazovka, aby uživatel nechtěně neprovedl nějakou akci hned poté, co se změní obrazovka.

Screen Základní třída pro všechny obrazovky. Definuje základní metody pro vykreslení, zpracování vstupů a také pomocné metody pro zmáčknutí jednotlivých tlačítek.

Menu Třída obrazovky pro zobrazování menu. Zobrazuje položky menu jako seznam a umožňuje mezi nimi listovat pomocí tlačítka nahoru/dolu a vybírat pomocí prostředního tlačítka.

StatusBoard Třída obrazovky pro zobrazení stavu jednotky, měřené teploty apod. Pro opuštění obrazovky se zmáčkne prostřední tlačítko.

StringInput Obrazovka, která slouží jako vstup pro zadávání tisknutelných ASCII znaků uživatelem. Umožňuje listování mezi znaky a příkazy pomocí tlačítek nahoru/dolu/vpravo/vlevo a výběr pomocí prostředního tlačítka.

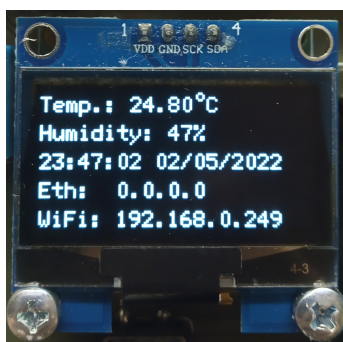
Confirmation Jednoduchá obrazovka, která slouží pro možnosti, kde uživatel vybírá ANO/NE. Výběr je prováděn pomocí tlačítek vpravo/vlevo a potvrzen prostředním tlačítkem.

TimeSet Obrazovka pro nastavování/výběr času a data na sekundy přesně. Umožňuje listovat mezi údaji data a měnit je. Při výběru je zohledněna validita data (dny v měsíci a přestupné roky).

Ukázky některých obrazovek tak, jak se vykreslují na OLED displeji, lze vidět na obrázcích 6.2, 6.3 a 6.4.



■ Obrázek 6.2 Obrazovka s hlavním menu



■ Obrázek 6.3 Obrazovka pro zobrazení statusu jednotky



■ Obrázek 6.4 Obrazovka pro zadávání řetězcového vstupu

6.3.4 Rozšiřovač portů

O správu rozšiřovače portů se stará program v *io_ports.cpp* a *io_ports.h*. Ovládají se zde vstupy a výstupy rozšiřovače (relé a tlačítka). Pro práci s rozšiřovačem MCP23017 je zde použita knihovna *Adafruit MCP23017 Arduino Library*.

io_init Funkce inicializuje komunikaci po I2C sběrnici a také volá níže uvedené inicializační funkce pro tlačítka a relé.

button_init Inicializační funkce pro tlačítka. Nastavuje vstupní piny z tlačítek jako pullup vstupy a inicializuje proměnné používané pro ochranu proti zákmitům tlačítek.

relay_init Funkce inicializuje výstupní piny rozšiřovače pro relé, které uvede do rozepnuté polohy.

buttons_loop Smyčka pro čtení tlačítkových vstupů. Je zde prováděna programová ochrana proti zákmitům tlačítek.

relay_set Funkce slouží k nastavování výstupních relé.

get_relay_state Tato funkce slouží k zjištění aktuálního stavu relé.

6.3.5 Měření teploty

Obsluhu měření teploty a vlhkosti pomocí DHT11 lze najít v souborech *sensor.cpp* a *sensor.h*. Je zde využita knihovna *DHT sensor library*, která zprostředkovává komunikaci se senzorem po jednovodičovém rozhraní přes GPIO 4.

dht_init Funkce pro inicializaci instance senzoru DHT11.

dht_reading Tato funkce změří aktuální teplotu a vlhkost, pokud jsou obě hodnoty validní, přepíše výstupní parametry na naměřené hodnoty.

6.3.6 Obvod RTC

Pro správu obvodu reálného času je využito podprogramu v *rtc.cpp* a *rtc.h*. Podprogram se stará o poskytování času z RTC a o jeho nastavování. Je zde použita knihovna *RTCLib*.

rtc_init Inicializace komunikace s obvodem RTC. Pokud je zjištěno, že obvod RTC přišel o napájení a tudíž o aktuální čas, je zde i nastaven nový výchozí čas.

get_datetime Funkce, která navrácí aktuální datum a čas získané z obvodu RTC.

set_datetime Tato funkce slouží k nastavení nového času obvodu RTC.

6.3.7 Připojení

Podprogram pro správu WiFi a ethernetového připojení lze nalézt v souborech *network.cpp* a *network.h*. Správa připojení používá standardní knihovny *WiFi.h* a *ETH.h*. Ethernet a WiFi se pak chovají jako odlišná rozhraní pro účely IP adresace, ale přístup na server jednotky je možný nezávisle.

Obslužný program pracuje se síťovými adresami IPv4, ale použité knihovny teoreticky mohou podporovat i IPv6 adresaci.

ethernet_init Funkce inicializuje ethernetové rozhraní a nastaví handler pro události na něm.

ethernet_wifi_handler Handler pro zpracovávání událostí na ethernetovém rozhraní (připojení/odpojení, získání IP adresy apod.).

ethernet_connected Funkce pro ověření, zda je ethernetové rozhraní úspěšně připojené.

get_ethernet_ip Vrací IP adresu ethernetového rozhraní.

wifi_init Funkce pro inicializaci WiFi připojení. Nastaví ESP32 jako přístupovou stanicí a pokusí se připojit k zadané WiFi síti.

wifi_connected Funkce pro ověření, zda je rozhraní WiFi úspěšně připojené.

wifi_connect_to Tato funkce odpojí rozhraní WiFi od aktuálně připojené sítě a pokusí se připojit do nově zadané WiFi sítě.

get_wifi_ip Vrací IP adresu rozhraní WiFi.

6.3.8 Server

Rozhraní a funkce serveru zajišťuje podprogram v *server.cpp* a *server.h*. Pro tento podprogram byla použita knihovna *esp32_https_server*. Server používá šifrovanou SSL komunikaci na portu 443.

Pro připojení na stránku jednotky je nutné se přihlásit pomocí uživatelského jména a hesla, které se dají nastavit přímo na jednotce. Stránka pak umožňuje přepínat relé a zobrazuje základní data vyčtená z jednotky (teplota, vlhkost a čas RTC). Ukázkou stránky zobrazené v prohlížeči na mobilním telefonu lze vidět na obrázku 6.5.

Klíč a certifikát stránky byly vygenerovány pro demonstraci a jsou uvedeny v souborech *private_key.h* a *cert.h*. Jelikož byly vytvořeny jen pro účely této práce, nejsou spravované žádnou certifikační autoritou a jsou tedy webovými prohlížeči uvažovány jako nedůvěryhodné.



■ **Obrázek 6.5** Webová stránka jednotky zobrazená v mobilním telefonu

server_setup Funkce pro nastavení a spuštění serveru. Váže cílové stránky s obslužnými rutinami a zavádí autorizační a validační funkce.

server_loop Smyčka běhu serveru pro spuštění v hlavním programu.

server_update_status Aktualizuje status pro zobrazování na hlavní stránce serveru (teplota, vlhkost a čas RTC).

access_set Nastaví autentizační údaje pro přístup na stránky.

handle_root Funkce obsluhující přístup na hlavní stránku. Stará se o vytvoření tlačítek pro přepínání relé a také o zobrazování aktuálních údajů o teplotě, vlhkosti a času RTC.

handle_switch Funkce obsluhující přepínání relé podle HTTP POST požadavků, které jsou poslány pomocí tlačítek na hlavní stránce.

handle_not_found Zajišťuje obsluhu požadavků na neexistující stránky.

validate_relay_id Validační funkce pro id relé. Ověřuje jestli parametr id požadavku správně označuje některé z relé (hodnota id mezi 0–3).

validate_relay_state Validační funkce pro stav relé. Tato funkce ověřuje jestli požadavek správně určuje spuštění nebo vypnutí relé.

middleware_access_control Funkce, která je spouštěna pro každý požadavek. Zajišťuje autorizaci uživatele, případně o ni žádá.

6.3.9 Paměť

Obsluha nevolatilní paměti (data uložená v této paměti jsou zachována i při ztrátě/odpojení napájení) je zajištěna v programové části *memory.cpp* a *memory.h*. Pro práci s pamětí je použita standardní Arduino knihovna *EEPROM.h*, která pro ukládání dat se systémy ESP32 používá předem inicializovaný úsek flash paměti.

Ukládání dat do flash paměti není vhodné dělat často, jelikož paměť má omezený počet zápisových cyklů. Je možné zapisovat data, která se nemění často. Jednotka tedy ukládá jen data pro připojení do WiFi sítě a přihlašovací údaje serveru, jelikož se nepředpokládá, že se budou měnit často, ale zároveň je žádoucí aby se jednotka po ztrátě napájení automaticky napojila do stejné WiFi sítě a aby se nemusely znovu nastavovat přihlašovací údaje pro přístup k webovým stránkám. Naopak by nebylo vhodné provádět například periodické zaznamenávání stavu jednotky nebo ukládat přesný stav výstupů relé.

memory_init Inicializační funkce pro paměťový podprogram. Alokuje se v ní paměť předem definované velikosti, s kterou se dále bude pracovat.

memread... Skupina funkcí *memread* slouží k přečtení požadovaných dat z paměti. Je tvořena funkcemi pro čtení přístupových údajů do WiFi sítě a přihlašovacích údajů na webové stránky.

memwrite... Skupina funkcí pro zápis dat do paměti. Zápis je nutné potvrdit funkcí *memcommit*. Po potvrzení zápisu jsou zapsaná data přístupná přes příslušnou funkci *memread*.

memcommit Účelem této funkce je potvrzení zápisu do paměti funkcemi *memwrite*. Záписы nepotvrzené touto funkcí nejsou do paměti propsány.

Testování

Testování jednotky a obslužného programu spolu s případnou analýzou a opravou zjištěných chyb se věnuje tato kapitola. Také byly testovány samostatné moduly, které jsou součástí jednotky. S výhodou bylo při testování modulů využito dutinkových lišt jednotky, které umožňovaly vytvořit zapojení pouze s některými moduly, které byly zrovna testované.

7.1 Testování modulů

Při návrhu a vývoji obslužného programu, jakožto i jednotky, byly průběžně testovány používané moduly a k nim příslušné knihovny, nejdříve zvlášť a poté ve větších celcích. Testy byly vcelku jednoduché a sloužily hlavně pro ověření funkcionality modulů a částí obslužného programu. Testovány byly především různé obrazovky uživatelského rozhraní a jejich ovládání pomocí tlačítek. Testování modulů při vývoji bylo následující:

Relé Testování reakce na spouštěcí signál. Ověření správného sepnutí relé (uzavření síťového obvodu).

Rozšiřovač portů Ověřování správného čtení a zápisu na porty rozšiřovače. Kontrola správné funkčnosti vnitřního pullup zapojení.

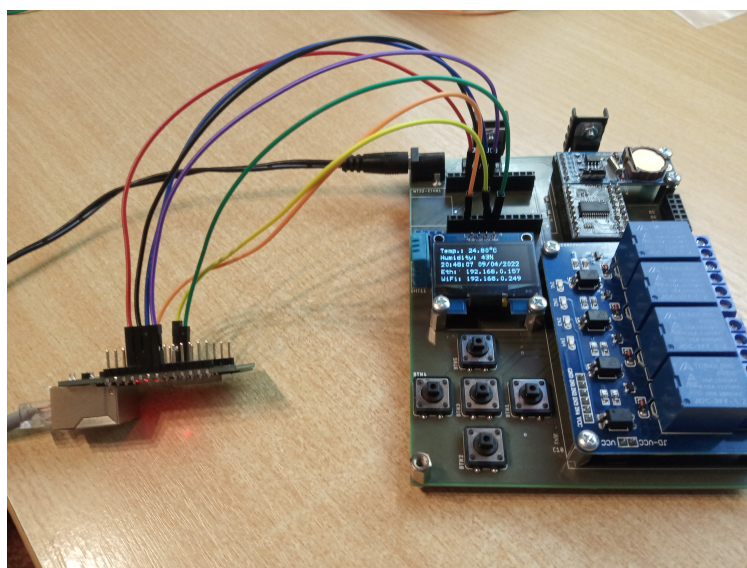
Měření teploty Kontrola správné funkčnosti senzoru DHT11 (nesmyslné hodnoty měření, neúspěšná měření). Reakce na náhlé změny teploty a vlhkosti.

Uživatelské rozhraní Testování funkčnosti různých obrazovek a reakcí na tlačítka. Ověření čitelnosti textu.

RTC Testování nastavování času. Ověření udržování času i po ztrátě napájení.

7.2 Testování připojení

Pro ověření WiFi a ethernetového připojení k jednotce bylo využito domácí sítě spravované routerem s DHCP. Prototyp jednotky fungoval jako server zobrazující jednoduchou stránku a pomocí ostatních zařízení (2 PC, notebook a telefon) v lokální síti byla ověřována zobrazitelnost webové stránky při připojení přes prohlížeč. Při tomto testování byly objeveny dva problémy, kterým se věnují následující podkapitoly.



■ **Obrázek 7.1** Zapojení s funkční ethernetovou komunikací

7.2.1 Problém s ethernetovým připojením

Při použití osazené desky jednotky bylo zjištěno, že není úspěšně inicializováno ethernetové rozhraní mikropočítače WT32-ETH01 spravované čipem LAN8720A, nicméně ostatní funkce byly nedotčeny. Tento problém se projevil až při zapojení mikropočítače do příslušného místa na desce. Při testování mikropočítače samotného probíhala celá ethernetová komunikace v pořádku.

Nejdříve bylo otestováno, zdali není na desce jednotky vytvořené nechtěné propojení nebo zkrat, které by mohlo funkci mikropočítače narušovat. To bylo testováno multimetrem, nejdříve mezi sebou sousedící piny a následně kontrola nechtěných propojení vůči zemi, nebo napájení. Všechny kontrolované piny byly v pořádku.

Následně byly odmontovány všechny možné ostatní moduly a otestováno zapojení s mikropočítačem samotným. Na desce kromě mikropočítače zůstal jen senzor DHT11 a další neoddelitelné součástky (tlačítka, kondenzátory, rezistory apod.). Toto zapojení také nefungovalo a ethernet se neinicializoval.

Dalším postupem bylo zapojení mikropočítače do desky přes propojovací kabely. Propojeny byly nejdříve jen využití piny. Při tomto zapojení se podařilo ethernetové připojení zprovoznit, proto vznikla domněnka, že chybu způsobuje některý z nezapojených pinů.

Pro ověření vzniklé domněnky byly postupně po jednom kabelem propojovány všechny nevyužití vodiče přímo do desky. Ve všech případech však ethernet modul fungoval, proto bylo vyzkoušeno propojit všechny nevyužití piny přímo do desky. Při propojení všech pinů ethernetový modul stále fungoval a bylo tedy nutné ověřit, jestli správně funguje zapojení mikropočítače do dutinkové lišty (hlavně napájecí a zemní pin). Mikropočítač byl zasunut do dutinkové lišty a multimetrem bylo zjišťováno, jestli mezi piny mikropočítače a vývodem příslušné dutinky nevzniká velký přechodový odpor, avšak nebyla zjištěna žádná neobvyklá hodnota.

Příčina této chyby nebyla odhalena. Je možné, že se jedná o nezdokumentovanou chybu v Arduino frameworku pro ESP32, která způsobuje chybu inicializace. Prozatímním řešením se zdá být zapojení mikropočítače na desku pomocí kabelových propojek (viz obr. 7.1).

7.2.2 Problém s vyřizováním HTTP požadavků

Přístup na server byl zkoušen z několika různých zařízení a prohlížečů. Při připojení z PC přes prohlížeč Firefox nebyl zaznamenán žádný problém ani při opakované komunikaci. Pokud však bylo použito jiných prostředků (PC s prohlížečem Opera a telefon s Google Chrome), tak došlo ke kompletnímu zamrznutí programu mikropočítače.

Viníkem tohoto zamrznutí je možnost vzniknutí nekonečné smyčky při zpracovávání komunikace. Chyba leží v knihovně *esp32_https_server* a věnuje se jí pull request této knihovny na GitHubu: [30]. Je nutné opravit funkci pro zpracování HTTP požadavků, aby se program nezacyklil.

Po použití oprav knihovny bylo připojení ze všech zařízení v pořádku, jednotka nepřestala pracovat a spojení fungovalo bez problémů.

7.3 Testování provozu

Tento test se věnoval provozu jednotky po delší dobu. Jednotka byla v neustálém provozu po dobu několika dní v místnosti při pokojové teplotě na dobře větraném místě ve stínu a byla kontrolována její stabilita a funkčnost. Pro napájení jednotky byl použit 9V/1A stejnosměrný zdroj. Připojení bylo zřízeno pomocí domácí WiFi a několikrát denně byla testována funkčnost webového rozhraní (zobrazení údajů a ovládání relé) a manuálního rozhraní (tlačítka a obrazovka).

Jednotka vydržela v nepřetržitém provozu 7 dní, dokud nebyla vypojena ze sítě. Připojení na web jednotky bylo bez problémů. Reakce na uživatelské vstupy přes tlačítka a vykreslování obrazovky fungovalo bez problémů. Bylo pozorováno mírné zahřívání jednotky (obzvláště regulátorů napětí), když byl WiFi signál slabý a nebo pokud byla spuštěna všechna relé. Důvodem zahřívání při spuštění relé je zvýšení potřebného proudu pro sepnutí relé. Zahřívání při slabém WiFi signálu je pak pravděpodobně způsobeno častým opětovným hledáním a připojováním do sítě, které je náročné na napájení. Zahřívání regulátorů bylo kompenzováno chladiči.



Kapitola 8

Závěr

Cílem práce bylo prozkoumat existující řešení pro vzdálenou správu domácnosti, navrhnout a vytvořit funkční prototyp jednotky pro vzdálenou správu, vytvořit obslužný program demonstrující funkčnost tohoto prototypu a otestovat jej.

V kapitole Existující řešení byly prozkoumána komerční a nekomerční řešení pro vzdálené řízení domácnosti. Kapitola Analýza se poté zabývala popisem a výběrem vhodných součástí pro použití při vytváření prototypu jednotky. Následně byl v kapitole Návrh představen plán komunikace mezi jednotlivými moduly a naplánovaná struktura pro realizaci programu. Kapitola Realizace jednotky se pak zabývala samotnými zapojeními součástí a vytvořením desky samotné. Kapitola Realizace programu se zabývala realizací softwarové části jednotky a popisem vytvořeného programu. Nakonec byla jednotka otestována v kapitole Testování, kde se kontrolovala správná funkčnost.

Výsledkem práce je deska plošného spoje představující funkční prototyp jednotky, nastavitelný pomocí manuálního uživatelského rozhraní s tlačítky a displejem. Připojení do domácí sítě je zařízeno pomocí WiFi nebo ethernetu. Jednotka pak pracuje jako server, provozující jednoduché webové stránky po zabezpečeném připojení a s autorizací uživatele. Pomocí webového rozhraní tohoto serveru může uživatel ovládat relé na jednotce a zobrazovat si informace o změřené teplotě a vlhkosti. Pro přidání další funkcionality nebo modulů mohou být použity vyvedené rozšiřující porty nebo sériová linka UART.

Vytvořený prototyp může sloužit pro vzdálené ovládání zařízení v domácnosti pomocí relé a nebo jako podklad pro další práce studentů.

Bibliografie

1. *Immax NEO BRIDGE PRO Smart Zigbee 3.0 v2* [online]. IMMAX WPB CZ, s.r.o., 2021 [cit. 2022-04-22]. Dostupné z: <https://www.immax.cz/immax-neo-bridge-pro-smart-zigbee-3-0-v2-p10213>.
2. *Philips Hue Bridge* [online]. Signify Holding, 2022 [cit. 2022-04-22]. Dostupné z: <https://www.philips-hue.com/cs-cz/p/hue-hue-bridge/8719514342620>.
3. *Tesla Smart ZigBee Hub* [online]. TESLA Solar, s.r.o., 2022 [cit. 2022-04-22]. Dostupné z: <https://www.teslasmart.com/produkty/chytre-centralni-jednotky/tesla-smart-zigbee-hub/>.
4. *Tesla Smart ZigBee Hub* [online]. Alza.cz a.s., 2022 [cit. 2022-04-22]. Dostupné z: <https://www.alza.cz/tesla-smart-zigbee-hub-d6551965.htm>.
5. *Xiaomi Mi Smart Home Hub* [online]. Xiaomi.cz, 2022 [cit. 2022-04-22]. Dostupné z: <https://www.xiaomi.cz/xiaomi-mi-smart-home-hub/>.
6. *Philips Hue Bridge* [online]. CZC.cz, 2022 [cit. 2022-04-23]. Dostupné z: <https://www.czc.cz/philips-hue-bridge/228857/produkt>.
7. *Xiaomi Mi Smart Home Hub* [online]. CZC.cz, 2022 [cit. 2022-04-23]. Dostupné z: <https://www.czc.cz/xiaomi-mi-smart-home-hub/299090/produkt>.
8. GILL, Khusvinder; YANG, Shuang-Hua; YAO, Fang; LU, Xin. A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics*. 2009, roč. 55, č. 2, s. 422–430. Dostupné z DOI: 10.1109/TCE.2009.5174403.
9. TEYMOURZADEH, Rozita; AHMED, Salah Addin; CHAN, Kok Wai; HOONG, Mok Vee. Smart GSM based Home Automation System. In: *2013 IEEE Conference on Systems, Process Control (ICSPC)*. 2013, s. 306–309. Dostupné z DOI: 10.1109/SPC.2013.6735152.
10. ASADULLAH, Muhammad; ULLAH, Khalil. Smart home automation system using Bluetooth technology. In: *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*. 2017, s. 1–6. Dostupné z DOI: 10.1109/ICIEECT.2017.7916544.
11. *WT32-ETH01 datasheet* [online]. Wireless-Tag Technology Co., Ltd., 2019. Ver. 1.1 [cit. 2022-04-15]. Dostupné z: <http://www.wireless-tag.com/wp-content/uploads/2020/08/WT32-ETH01%E8%A7%84%E6%A0%BC%E4%B9%A6V1.2EN%EF%BC%88%E8%8B%B1%E6%96%87%E6%96%89.pdf>.
12. *MCP23017/MCP23S17 datasheet* [online]. Microchip Technology Inc., 2016. Ver. C [cit. 2022-04-15]. Dostupné z: <https://ww1.microchip.com/downloads/en/devicedoc/20001952c.pdf>.

13. WU, Shin-Tson; YANG, Deng-Ke. *Fundamentals of Liquid Crystal Devices*. New York: John Wiley & Sons, Incorporated, 2014. ISBN 9781118751992.
14. GEFFROY, Bernard; ROY, Philippe le; PRAT, Christophe. Organic light-emitting diode (OLED) technology: materials, devices and display technologies. *Polymer International* [online]. 2006, roč. 55, č. 6, s. 572–582 [cit. 2022-05-03]. Dostupné z DOI: <https://doi.org/10.1002/pi.1974>.
15. *1.3"128x64 OLED displej, I2C, bílý* [online]. 2022 [cit. 2022-04-18]. Dostupné z: <https://www.laskakit.cz/oled-displej-bily-128x64-1-3--i2c/>.
16. *2.2"240x320 TFT displej, ILI9341, SPI* [online]. 2022 [cit. 2022-04-18]. Dostupné z: <https://www.laskakit.cz/240x320-barevny-lcd-tft-displej-2-2--spi/>.
17. *20x4 LCD displej 2004 modrý + I2C převodník* [online]. 2022 [cit. 2022-04-18]. Dostupné z: <https://www.laskakit.cz/20x4-lcd-displej-2004-modry-i2c-prevodnik>.
18. *LCD2004A specification* [online]. Shenzhen E-One Electronics Co., Ltd., 2007 [cit. 2022-04-18]. Dostupné z: https://uk.beta-layout.com/download/rk/RK-10290_410.pdf.
19. *SH1106 datasheet* [online]. Sino Wealth Electronic Ltd., 2013. Ver. 2.3 [cit. 2022-04-18]. Dostupné z: <https://datasheetspdf.com/pdf/956102/SINOWEALTH/SH1106/1>.
20. *Mikrospínače řady TACTS-2* [online]. TME Czech Republic s.r.o., 2022 [cit. 2022-05-08]. Dostupné z: https://www.tme.eu/html/CZ/mikrospinace-rady-tacts-2/ramka_1443_CZ_pelny.html.
21. MAŤÁTKO, Jan. Součástky řízené neelektrickými veličinami. In: *Elektronika*. 4. vyd. Praha: IDEA SERVIS, 1997, s. 142–155. ISBN 80-85970-20-1.
22. *DHT11 Humidity & Temperature Sensor datasheet* [online]. D-Robotics, 2010 [cit. 2022-04-19]. Dostupné z: <https://datasheetspdf.com/pdf-file/785590/D-Robotics/DHT11/1>.
23. PRASAD, Leela. What is Relay? How it Works? Types, Applications, Testing. In: *electronicshub.org* [online]. Electronics Hub, 2022 [cit. 2022-04-15]. Dostupné z: <https://www.electronicshub.org/what-is-relay-and-how-it-works/>.
24. *PCB JQC-T73* [online]. Donghai Tongling Electric Appliances Co., Ltd, 2020 [cit. 2022-04-15]. Dostupné z: <https://en.tonglingrelay.com/product/JQC-3FF-S-Z-5VDC-15A-5V-Relay-1065.html>.
25. *SH1106 datasheet* [online]. NXP Semiconductors N.V., 2015 [cit. 2022-04-19]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/PCF8563.pdf>.
26. *LFXX datasheet* [online]. STMicroelectronics, 2017 [cit. 2022-04-23]. Dostupné z: <https://www.st.com/resource/en/datasheet/lfxx.pdf>.
27. *LM1085 3-A Low Dropout Positive Regulators* [online]. Texas Instruments Incorporated, 2015 [cit. 2022-04-23]. Dostupné z: <https://www.ti.com/lit/ds/symlink/lm1085.pdf>.
28. DHAKER, Piyu. *Introduction to SPI Interface* [online]. Analog Devices, Inc, 2018 [cit. 2022-05-07]. Dostupné z: <https://www.analog.com/media/en/analog-dialogue/volume-52/number-3/introduction-to-spi-interface.pdf>.
29. *I2C-bus specification and user manual* [online]. NXP Semiconductors N.V., 2021 [cit. 2022-05-07]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>.
30. VVIGILANTE. *Fix infinite loop when the buffer ends with \r* [online]. github.com, 2021 [cit. 2022-03-05]. Dostupné z: https://github.com/fhessel/esp32_https_server/pull/123.

Obsah přiloženého média

	readme.txt	stručný popis obsahu média	
	thesis.pdf	text práce ve formátu PDF	
	board	adresář se schématem a návrhem desky jednotky	
	extras	adresář obsahující použité úpravy knihovny esp32_https_server	
	project	adresář se zdrojovou formou programu	
		include	adresář s hlavičkovými soubory
		src	adresář se zdrojovými kódy
		platformio.ini	inicializační soubor projektu pro VS Code s PlatformIO
	thesis_src	adresář se zdrojovou formou práce v LaTeX	
		thesis.zip	komprimovaná složka ZIP s kompletní prací včetně šablony
		text	adresář se zdrojovým kódem textu práce