

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

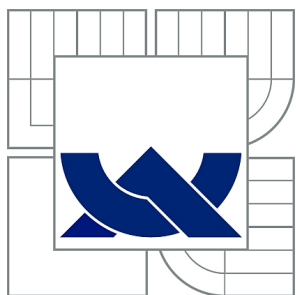
VZDÁLENĚ ŘÍZENÝ TERMOSTAT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

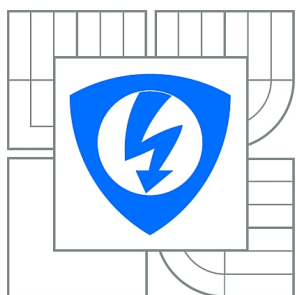
Bc. VILÉM ZÁVODNÝ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VZDÁLENĚ ŘÍZENÝ TERMOSTAT

REMOTELY CONTROLLED THERMOSTAT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VILÉM ZÁVODNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PAVEL ŠTEFFAN, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Vilém Závodný

ID: 125710

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Vzdáleně řízený termostat

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte termostat pro regulaci vytápění, který bude možné ovládat vzdáleně přes webové rozhraní. Při návrhu mezi sebou porovnejte různá řešení, srovnajte jejich výhody a nevýhody. Vybrané řešení realizujte.

DOPORUČENÁ LITERATURA:

[1] HRBÁČEK, J. Komunikace mikrokontroléru s okolím. Praha: BEN - technická literatura, 1999. 159 s. ISBN 80-86056-42-2.

[2] MATOUŠEK, D. Práce s mikrokontroléry ATMEL AVR AT90S. Praha: BEN - technická literatura, 2006. 376 s. ISBN 80-7300-209-4.

Termín zadání: 10.2.2014

Termín odevzdání: 28.5.2014

Vedoucí práce: doc. Ing. Pavel Šteffan, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Cílem práce je porozumět vyráběným termostatům a navrhnout vlastní termostat tak, aby umožňoval většinu základních možností klasických termostatů a navíc nastavení a správu přes webové rozhraní. Jedná se tedy o realizaci malého webového serveru, který úzce spolupracuje s hardwarovými periferiemi mikrokontroléru. Kromě webového rozhraní je ještě možné spravovat webový server z LCD TFT displeje nebo aplikací pro chytrý telefon s Androidem. Navíc termostat umožňuje čtení teplot z bezdrátových BLE modulů. Všechny teploty jsou ukládány průběžně do databáze a graficky zpracovávány.

KLÍČOVÁ SLOVA

Embedded webový server, termostat, měření teploty, měření vlhkosti, STM32F4 Discovery, BLE moduly, DHT11

ABSTRACT

The aim of my thesis is to understand the manufactured thermostats and design own thermostat to allow the most basic possibilities classical thermostats plus setup and management via web interface. It is a realization of a small web server that works closely with hardware peripherals. In addition, the web interface is still possible to manage a Web server of TFT LCD screen or applications for smart phone with Android. In addition, the thermostat allows the temperature reading from the BLE modules. All temperatures are continuously stored in a database and graphically processed.

KEYWORDS

Embedded web server, thermostat, measure temperatur, measure humidity, STM32F4 Discovery, BLE moduls, DHT11

ZÁVODNÝ, Vilém *Vzdáleně řízený termostat*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 70 s. Vedoucí práce byl doc. Ing. Pavel Šteffan, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vzdáleně řízený termostat“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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)



Děkuji Doc. Ing. Pavlovi Šteffanovi, Ph. D. za metodickou pomoc, připomínky, cenné rady při zpracování práce a zakoupení potřebných modulů a Ing. Ladislavu Macháňovi za příkladnou pomoc v laboratoři.

Dále děkuji firmám STMicroelectronics, Spezial electronic, ConnectBlue a Conrad za pomoc při výběru součástek a materiální i technickou podporu.



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	12
1 Teoretický úvod	13
1.1 Běžně prodávané termostaty	13
1.1.1 Jednoduchý termostat	13
1.1.2 Klasický programovatelný termostat	13
1.1.3 Inteligentní programovatelný termostat	14
1.1.4 MAX! Cube	15
1.2 Vývojový kit STM32F4 Discovery	16
1.2.1 Mikrokontrolér STM32F407VG	17
1.2.2 FreeRTOS	18
1.2.3 LwIP stack	19
1.2.4 Datové sběrnice	20
1.2.5 USART	20
1.2.6 I ² C (TWI)	20
1.2.7 SPI	21
1.2.8 One-wire	22
1.3 Ethernet	24
1.3.1 ENC28J60	24
1.3.2 Lantronix XPORT	24
1.3.3 LAN8720	26
1.4 LCD TFT displej	26
1.4.1 Komunikace s displejem	27
1.4.2 Generování souborů s písmem	28
1.4.3 Generování souborů s obrázkem	28
1.5 Bezdrátové moduly	29
1.5.1 OBS421	29
1.5.2 OLP425	30
1.5.3 OLS426	31
1.5.4 Nastavení a propojení jednotlivých modulů	31
1.6 Měření teploty a vlhkosti	33
1.6.1 Teplotní čidlo DS18B20	33
1.6.2 Čidlo teploty a vlhkosti DHT11	34
1.6.3 Teplotní čidlo na bluetooth modulu	35
1.7 Datum a čas	36
1.8 Android	36

2 Praktická realizace	38
2.1 Termostat s mikrokontrolérem ATMEL	38
2.2 Termostat s vývojovým kitem STM32F4 Discovery	41
2.2.1 Popis procesů	42
2.2.2 Příjem POST a GET parametrů	43
2.2.3 Zabezpečení	44
2.2.4 Způsoby ovládání termostatu	45
2.2.5 Termostat s mikrokontrolérem STM32F417VG	48
2.3 Ukládání dat a zobrazování statistik	50
2.3.1 Ukládání aktuálních hodnot do databáze	50
2.3.2 Zobrazování statistik	50
3 Závěr	52
Literatura	54
Seznam symbolů, veličin a zkratk	56
Seznam příloh	57
A Komunikace bluetooth modulů	58
B Schémata	60
C PCB	67

SEZNAM OBRÁZKŮ

1.1	Jednoduchý termostat [2]	13
1.2	Klasický programovatelný termostat [3]	14
1.3	Inteligentní programovatelný termostat [4]	14
1.4	MAX! Cube - Lan Gateway	15
1.5	MAX! Cube - bezdrátová hlavice	15
1.6	MAX! Cube - bezdrátový termostat	16
1.7	Vývojový kit STM32F4 Discovery	17
1.8	Datový rámec USART – asynchronní přenos dat [12]	20
1.9	Blokové schéma zapojení několika zařízení na I ² C [13]	21
1.10	Přenos jednoho 8bitového rámce po I ² C [13]	21
1.11	Přenos jednoho 8bitového rámce oběma směry po sběrnici SPI [13]	22
1.12	Spojení dvou obvodů pomocí sběrnice SPI [13]	22
1.13	Časový průběh zahájení komunikace pomocí RESET a PRESET sekvence [7]	23
1.14	Časový průběh zápisu log. 0 [7]	23
1.15	Časový průběh zápisu log. 1 [7]	23
1.16	Časový průběh čtení logické úrovně [7]	24
1.17	Základní zapojení obvodu ENC28J60	25
1.18	Lantronix XPORT	25
1.19	LCD TFT displej s řadičem SSD1289	27
1.20	Font Converter for emWin V5.22	28
1.21	Modul OBS421	29
1.22	Blokové schéma připojení modulu OBS421 k počítači	30
1.23	Modul OLP425	30
1.24	Modul OLS426	31
1.25	Spojení bezdrátových modulů do sítě	31
1.26	Senzor teploty a vlhkosti DHT11	34
1.27	Datový rámec modulu DHT11	35
1.28	Modul s obvodem ds1307 z obou stran	36
2.1	Zobrazení nastavené a změřené teploty na TFT LCD dotykovém displeji	38
2.2	Blokové schéma vytvořeného zřízení s mikrokontrolérem Atmel	39
2.3	Webová stránka pro nastavení teploty u termostatu s mikrokontrolérem Atmel	40
2.4	Blokové schéma termostatu realizovaného s vývojovým kitem STM32F4 Discovery	41
2.5	Přihlašovací stránka	45
2.6	Stránka s přehledem teplot	46

2.7	Stránka s týdenním programem	46
2.8	Stránka s nastavením	47
2.9	Stránka s daty ve formátu JSON	47
2.10	Aplikace pro Android	49
2.11	Hotové zařízení	50
2.12	Webová stránka s grafy	51
A.1	Spárování bluetooth modulů	58
A.2	Čtení a nastavování modulu OLP425 pomocí GATT příkazů z modulu OBS421	59
B.1	Schéma řídicí desky pro LCD TFT displej	60
B.2	Schéma webového serveru s mikrokontrolérem Atmel	61
B.3	Část schématu s připojením LCD	62
B.4	Část schématu s připojením bluetooth modulu	63
B.5	Část schématu s mikrokontrolérem a napájením	64
B.6	Část schématu s připojením ethernet modulu	65
B.7	Část schématu se senzory	66
C.1	Deska plošného spoje – pohled zespodu	67
C.2	Deska plošného spoje – pohled zvrchu	68
C.3	Deska plošného spoje – pohled zespodu – rozmístění součástek	69
C.4	Deska plošného spoje – pohled zvrchu – rozmístění součástek	70

SEZNAM TABULEK

1.1	Základní nastavení portu pro komunikaci s bluetooth modulem	30
1.2	Závislosti doby převodu a přesnosti na počtu bitů [15]	33
1.3	Závislosti doby převodu a přesnosti na počtu bitů [18]	35

ÚVOD

Téměř v každé budově je dnes kotel nebo jiné zařízení, které vyhřívá či klimatizuje příslušné místnosti uvnitř budovy. Toto zařízení je potřeba zapínat a vypínat podle teploty v místnosti. Pro usnadnění této činnosti byl vynalezen termostat, který se řídí teplotou uvnitř domu a podle toho ovládá kotel.

V případě, že nám nastavená teplota nevyhovuje, můžeme ji snadno změnit otočením kolečka termostatu nebo nastavením tlačítka. Pro pohodlné používání existují termostaty s týdenním programem. Ty umožňují nastavit např. denní a noční teploty podle času na každý den. Tím se dá snadno nastavit termostat tak, aby vytápěl dům jen pár hodin před tím, než se vrátíme z práce nebo z dovolené. A zároveň vypnul topení např. v noci nebo když jsme v práci.

Ale může nastat situace, kdy dojíždíme či odjíždíme nepravidelně. Pak nám pevný program v termostatu nestačí a je potřeba vzdáleně ovládat vyhřívání domu. Pro tento scénář je vhodný termostat ovládaný mobilním telefonem přes GSM síť. Pošleme SMS zprávu s teplotou a termostat pak reguluje vyhřívání domu podle nastavené teploty.

Navíc ještě může nastat třetí možnost, kdy dům stojí na místě, kde není GSM signál a nebo je velice nízká úroveň signálu. I tak dům stále nemusí být úplně mimo veškerou komunikaci s okolním světem. Na místech, kam se nedostane GSM signál se může dostat poskytovatel internetu. Pak by nám stačil termostat řízený přes internet, ale takové zařízení není možné sehnat s nízkou cenou. Většinou se používá centrální pult na řízení inteligentního domu nebo zabezpečovací zařízení, na které je možné napojit kotel. Ale obyčejný termostat s internetovým rozhraním neexistuje.

Proto jsem se v mé práci zaměřil na termostat vzdáleně řízený přes internet. Mimo to, jsem využil výkonu mikrokontroléru na další funkce, které doplňují možnosti klasického termostatu.

Před realizací bylo nutné vyzkoušet a prověřit spoustu senzorů, modulů a mikrokontrolérů tak, aby vyhovovaly maximálně požadavkům na cenu, výkon i jednoduchost.

Pro dostatečné vyzkoušení všech modulů jsem termostat realizoval celkem dvakrát. Pokaždé s jiným mikrokontrolérem a jinými moduly, abych vše mohl porovnat a najít vhodné řešení. Oba termostaty popisuji v této práci a v závěru srovnávám jejich vhodnost praktického použití.

1 TEORETICKÝ ÚVOD

1.1 Běžně prodávané termostaty

Jak bylo řečeno v úvodu, tak existuje celá řada termostatů. Od jednoduchých až po složité inteligentní, které disponují funkcemi jako optimální spouštění, aby uspořily, co nejvíce energie. Vybral jsem několik zástupců, které porovnáám v následujících kapitolách.

1.1.1 Jednoduchý termostat

Prvním představitelem je jeden z jednodušších termostatů. Toto zařízení umožňuje pouze nastavit kolečkem teplotu, na kterou se má vyhřívat daná místnost. Je možné jej úplně vypnout či zapnout malým vypínačem. A navíc zobrazuje stav indikační LED diodou. Termostat je vyobrazen na obrázku 1.1. [2]



Obr. 1.1: Jednoduchý termostat [2]

1.1.2 Klasický programovatelný termostat

Dalším představitelem je programovatelný termostat obvykle dodávaný firmou společně s kotlem. Toto zařízení umožňuje nejen nastavení současné teploty, ale má několik režimů, které si uživatel může sám nastavit. První užitečná funkce dává uživateli možnost si nastavit jinou denní a jinou noční teplotu. Další využívanou funkcí je nastavení vlastního programu na jednotlivé dny v týdnu. Je možné určit v jednotlivých dnech a hodinách, jestli termostat bude vytápět podle noční nebo denní teploty.

Jak je vidět, termostat je vybaven základními funkcemi vhodnými do rodinného domu, ale problém nastává, když uživatel chce tyto funkce opravdu využít a nastavit

je. K nastavení je totiž potřeba spousta tlačítek a jejich kombinace. To je velkou nevýhodou takového typu termostatů.

Výhodou oproti předchozímu zařízení je displej, na kterém je zobrazena aktuální teplota, čas a zvolený program.



Obr. 1.2: Klasický programovatelný termostat [3]

1.1.3 Inteligentní programovatelný termostat

Třetím zařízením, které jsem vybral a nechal se jím mírně inspirovat ve své práci je termostat umožňující nejen nastavování programů a teplot, ale také se sám učí a šetří tak energii. Vychází z doby, jak dlouho trvá vyhřát místnost na požadovanou teplotu a podle toho spouští kotel v čase, kdy je nutné. Navíc termostat umožňuje snadnější nastavení programů, které jsou již předem předvolené na funkce dovolená (delší nepřítomnost), party (kratší nepřítomnost) a další. [4]



Obr. 1.3: Inteligentní programovatelný termostat [4]

1.1.4 MAX! Cube

Poslední zařízení, které jsem objevil až v průběhu psaní mé práce mě oslovilo tím, že umožňuje připojení do sítě LAN. Je to tedy jedno z mála levnějších řešení, které by se mohlo podobat mé práci. Zařízení disponuje obrovským množstvím dobrých funkcí, ale i přesto má nedostatky.



Obr. 1.4: MAX! Cube - Lan Gateway

Celý systém se skládá z několika dílčích modulů. Jedná se o Lan-Gateway (obr. 1.4), bezdrátový nástěnný termostat (obr. 1.6) a bezdrátové programovatelné termostatické hlavice (obr. 1.5).



Obr. 1.5: MAX! Cube - bezdrátová hlavice

Systém funguje tak, že pouze vzdáleně spravuje termostatické hlavice, které podle teploty ovládají ventil na radiátoru. Hlavice je možné nastavovat přes internet pomocí zařízení Lan-Gateway a nebo pomocí nástěnného termostatu. V obou případech se jedná pouze o nastavení hlavic vzdálenou správou a pak už nejsou tyto zařízení

potřeba. Pouze pro případ změny nastavení nebo sledování aktuálních naměřených teplot. To znamená, že nemají ani možnost ovládat kotel podle nastavených teplot.



Obr. 1.6: MAX! Cube - bezdrátový termostat

Takže v případě, že v místnosti bude nastavena na hlavici větší teplota, než na kterou vytápí kotel, pak se v té místnosti nikdy této teploty nedosáhne. Proto je tento systém především vhodný do bytů, kde nemáme možnost regulovat spínání kotle.

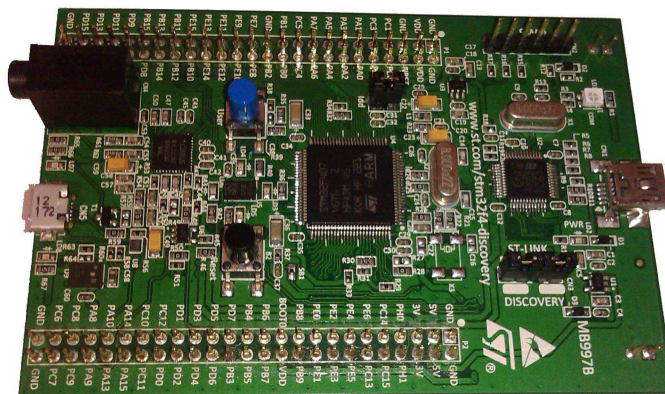
I přes tuto nevýhodu celý systém dokáže výrazně snížit náklady na vytápění.

1.2 Vývojový kit STM32F4 Discovery

Pro vlastní realizaci mé práce byl vybrán vývojový kit STM32F4 Discovery (obr. 1.7) od firmy STMicroelectronics a k němu mi firma ST zapůjčila i rozšiřující desku, která obsahuje další periferie na připojení k síti LAN, komunikaci s LCD TFT barevným dotykovým displejem, práci s SD kartou a spoustu dalších.

Celý vývojový kit řídí mikrokontrolér STM32F407VG. Kromě mikrokontroléru kit obsahuje ještě programátor ST-LINK s možností debuggeru, převodník na rs232, akcelerometr, uživatelské tlačítko, port pro USB OTG komunikaci a konektor na připojení zvukového zařízení.

K vývojovému kitu firmy STMicroelectronics a Farnell dodávají zdarma ukázkové programy, hotové knihovny a spoustu návodů k programování. Navíc je možné na kitu rozjet veřejný balík knihoven A Lightweight TCP/IP stack (LwIP stack), operační systém FreeRTOS a STemWin knihovnu pro práci s grafickými displeji.



Obr. 1.7: Vývojový kit STM32F4 Discovery

1.2.1 Mikrokontrolér STM32F407VG

Základem celého vývojového kitu je mikrokontrolér STM32F407VG od firmy STMicroelectronics. Mikrokontrolér obsahuje oproti konkurenčním spoustu hardwarových periferií a některé i ve velkém počtu. Jádrem mikrokontroléru je založené na ARM 32-bit Cortex-M4, které je vhodné pro embedded aplikace a navíc disponuje i jednotkou pro počítání s pohyblivou řádovou čárkou. Další výhodou oproti ostatním mikrokontrolérům v této kategorii je možnost použití až na 168 MHz. [17]

Základní vlastnosti mikrokontroléru

- Maximální frekvence 168 MHz.
- Velikost paměti FLASH je 1 MB.
- Velikost paměti SRAM je 192+4 KB.
- Obsahuje 3x 12bitový A/D převodník s 24 kanály.
- Obsahuje 2x 12bitový D/A převodník.
- Obsahuje 17x časovač.
- Obsahuje 140 vstupně/výstupních pinů.
- Disponuje 15 komunikačními rozhraními:
 - 3x I²C
 - 4x USART (10,5 Mbps)
 - 3x SPI (37,5 Mbps)
 - 2x CAN
 - 1x SDIO
 - 1x USB 2.0
 - 1x Ethernet MAC 10/100
- Umožňuje připojit 8 až 14bitovou kameru.

- Generátor náhodných čísel.
- Obsahuje obvod reálného času. [17]

1.2.2 FreeRTOS

Diplomová práce z pohledu softwaru je založena na real-time operačním systému FreeRTOS, který je volně ke stažení a použitelný i pro komerční aplikace bez nutnosti zveřejnění kódu. Pro tento operační systém mnozí výrobci dodávají přímo ukázkové aplikace pro použití s jejich mikrokontroléry. Je tomu tak i u mikrokontroléru STM32F4.

Výhodou použití real-time operačního systému je, že se neprovádí úkony v nekonečné smyčce, jak tomu bývá u klasického programu v mikrokontroléru, ale jsou rozděleny do jednotlivých procesů. Ty se pak volají podle priorit a jsou jim přiděleny určité prostředky.

Další výhodou FreeRTOS má v tom, že pro něj existuje spousta rozšíření např. pro komunikaci s LAN, USB nebo pamětovými médii. [6]

Praktické použití FreeRTOS

Pro základní práci s FreeRTOS stačí znát jen několik pravidel a funkcí. Hlavní činností je vytváření procesů. Pro založení nového procesu je potřeba nejprve vytvořit funkci procesu. Konstrukce funkce je zde:

```
void vATaskFunction( void *pvParameters )
{
    for( ;; )
    {
        -- Libovolný kód procesu. --
        vTaskDelay(250);
    }
}
```

Ve funkci může být libovolně ještě funkce `vTaskDelay`, která určí, že se daný proces bude provádět jen každých 250 ms v tomto případě. Po vytvoření funkce je potřeba ji zapsat do seznamu procesů. To provedeme funkcí:

```
xTaskCreate( vATaskFunction, "NAME", STACK_SIZE,
&ucParameterToPass, tskIDLE_PRIORITY, &xHandle );
```

Kde první parametr značí název volané funkce, která je vykonávána jako jeden proces. Druhý parametr označuje název procesu. Třetím parametrem je velikost

zásobníku pro vykonání funkce. Další parametr je ukazatel na volitelný parametr předávaný volané funkci. Pátý parametr určuje prioritu procesu. A poslední parametr je předáván jako ukazatel na vytvořený proces.

Jako poslední je potřeba zavolat funkci, která spustí plánovač úloh. To provedeme funkcí `vTaskStartScheduler()`, která se volá bez parametrů.

1.2.3 LwIP stack

Spolu s FreeRTOS je v mém projektu využit balíček LwIP stack pro implementaci potřebných protokolů ke komunikaci po síti LAN do mikrokontroléru. Zmíněný balíček realizuje protokoly IP, ICMP, UDP, TCP, IGMP, ARP, DNS, DHCP, HTTP a spoustu dalších, které nejsou v mé práci nevyužity. [10]

Základem celého balíčku je komunikace s ethernet rozhraním a zprostředkovávat komunikaci vyšším vrstvám, které balíček také realizuje. Vyšší vrstvy poté řeší každá své protokoly.

Praktické použití LwIP stack

Pro základní nastavení a navázání spojení je několik funkcí, které nyní popíšu. Nejprve je nutné vytvořit nové spojení funkcí `netconn_new(NETCONN_TCP)`, kde parametr určuje typ spojení a funkce vrací ukazatel na spojení. Spojení může být TCP a UDP. Poté je nutné nastavit spojení funkcí `netconn_bind(conn, NULL, 80)`, kde první parametr je ukazatel na vytvořené spojení, druhý určuje IP adresu a třetí port, na kterém bude spojení naslouchat. Když je IP adresa `NULL`, jako v tomto případě, znamená to, že bude naslouchat na všech IP adresách. Funkce vrací chybový kód. Pokud vrátí `ERR_OK`, pak nastavení proběhlo bez chyby. Nyní už můžeme spustit naslouchání serveru na nastavených parametrech a to zavoláním funkce `netconn_listen(conn)`, kde parametrem je ukazatel na vytvořené spojení.

V tomto případě už nám server funguje, ale nepřijímá žádná data. Proto ve smyčce voláme funkci `netconn_accept(conn)` s parametrem ukazatele na vytvořené spojení, která přijímá všechna příchozí spojení a vrací ukazatel na toto spojení. Následně je možné přijmout data z tohoto spojení funkcí `netconn_recv(connx)`, kde parametr je ukazatel na přijaté spojení. Nikoliv na spojení vytvořené. Funkce vrací přijatá data, které je vhodné ještě zpracovat funkcí

```
netbuf_data(inbuf, (void**)&buf, &buflen).
```

Po zpracování dostaneme ve druhém parametru funkce ukazatel na začátek dat ve struktuře `inbuf`.

Po zpracování přijatých dat pošleme data zpátky pomocí

```
netconn_write(connx,data,len,NETCONN_NOCOPY)
```

kde první parametr určuje ukazatel na přijaté spojení, druhý odesílaná data, třetí velikost dat a poslední určuje jestli se data během přenosu mění nebo ne. Nakonec odstraníme z paměti vytvořené přijaté spojení funkcí `netconn_delete(connx)`. [11]

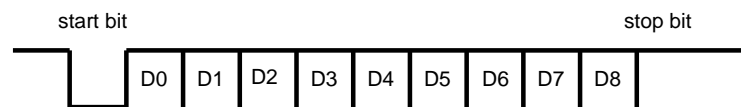
1.2.4 Datové sběrnice

Při praktické realizaci bylo využito několik komunikačních sběrnic.

1.2.5 USART

USART je synchronní a asynchronní sériové rozhraní pro přenos dat mezi zařízeními v obou směrech (plný duplex). Někdy je také nazývaný pouze jako UART a to v případě, že není možné použít synchronní linku. Používá se pro komunikaci mezi mikrokontroléry, počítači a dalšími zařízeními podporující tento standard.

Pro přenos se používají rámce, které mohou mít 5 až 9 bitů a jsou od sebe odděleny jedním start bitem a jedním nebo dvěma stop bity (obr. 1.8). Každý rámec může obsahovat ještě paritní bit pro kontrolu rámce. Dále je možné nastavit rychlost přenosu dat od 1 200 bps až do 10,5 Mbps.

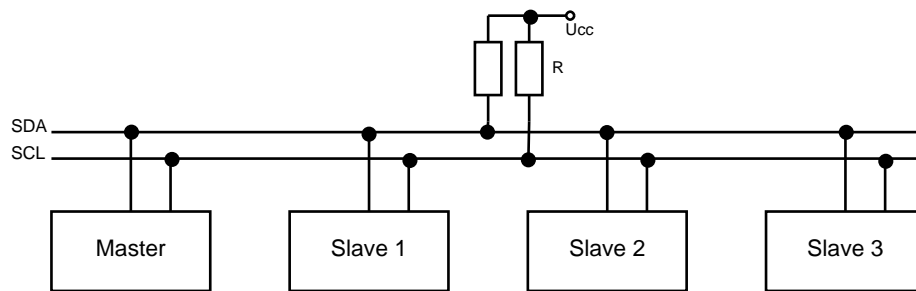


Obr. 1.8: Datový rámec USART – asynchronní přenos dat [12]

Použitý mikrokontrolér STM32F407VG disponuje celkem 6 sériovými rozhraními USART, ale pouze dvě rozhraní mohou používat maximální rychlost 10,5 Mbps. Ostatní mají maximální rychlost 5,25 Mbps.

1.2.6 I²C (TWI)

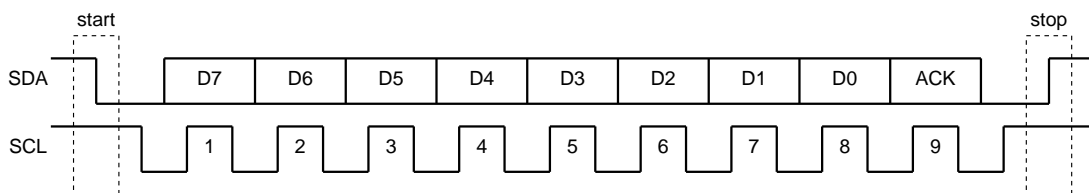
Jako další uvádím sběrnici I²C, někdy nazývanou také jako TWI (two-wire), která je běžně používaná pro připojení až 128 zařízení po dvou vodičích (obr. 1.9). Sběrnice je obousměrná, ale v jeden okamžik může vysílat pouze jedno zařízení (poloviční duplex). Oba vodiče musí být připojeny přes rezistor na napájecí napětí (tzv. pull-up rezistory), tím se předejde poškození vysílacích a přijímacích obvodů při kolizi a poškodí se pouze úroveň signálů.



Obr. 1.9: Blokové schéma zapojení několika zařízení na I²C [13]

Přenos dat

Na linku je připojen jeden master (řídící) a několik slave (podřízených) obvodů. Pro posláání datového rámce některému slave zařízení, master vyšle v prvních 7 bitech adresu slave zařízení, v osmém bitu log. 1 a čeká na příjem potvrzovacího bitu ACK. Poté master vyšle data po lince SDA. V případě, že je potřeba získat data ze slave zařízení, pak master opět pošle adresu, ale v osmém bitu pošle log. 0, aby slave poznal, že má volno a může poslat data. Synchronizace přenosu dat probíhá impulsy na lince SCL o frekvenci 10 kHz až 400 kHz. Průběh dat na linkách SDA a SCL je zřejmý z obrázku 1.10.

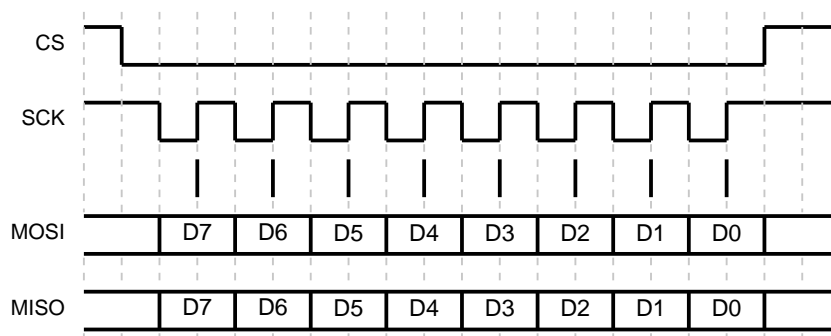


Obr. 1.10: Přenos jednoho 8bitového rámce po I²C [13]

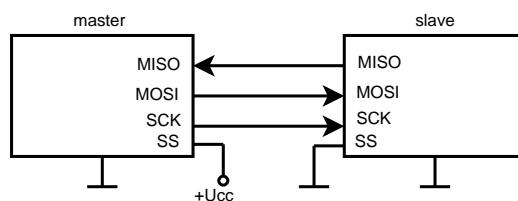
1.2.7 SPI

Další použitou sběrnici v mé práci je SPI, která se často používá pro přenos většího objemu dat, protože dosahuje vysokých přenosových rychlostí i u mikrokontrolérů a modulů. Přenos dat u SPI je obousměrný (plný duplex) díky linkám MOSI (master out slave in) a MISO (master in slave out), po kterých se přenáší data synchronizované linkou CLK. Na lince CS (někdy uváděno SS) je pouze informace pro slave obvod, aby bylo rozlišeno, kterému obvodu jsou data určena. Datový rámec je zobrazen

na obrázku 1.11, ze kterého je vidět jak jsou data závislá na synchronizačních pulsech. Na obrázku 1.12 je zobrazeno klasické zapojení mezi obvodem master a slave, které je použito v mé práci. Dále existují ještě dva způsoby propojení pro případ více slave obvodů řízených jedním masterem, který pak řídí celou komunikaci buď tím, že přiděluje jednotlivým slave zařízením vysílací čas, nebo si všechny obvody přeposílají data do kruhu.



Obr. 1.11: Přenos jednoho 8bitového rámce oběma směry po sběrnici SPI [13]

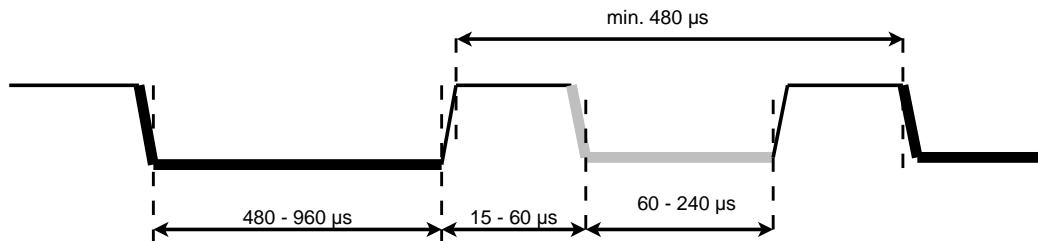


Obr. 1.12: Spojení dvou obvodů pomocí sběrnice SPI [13]

1.2.8 One-wire

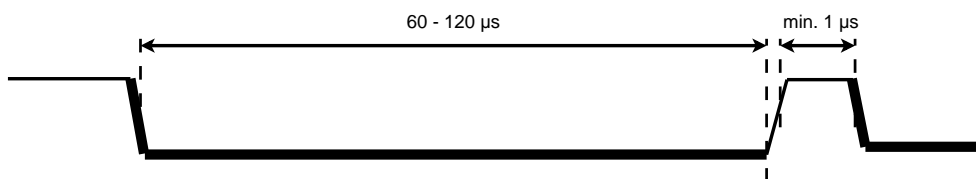
Jednovodičová sběrnice one-wire (1-wire) vyvinutá společností DALLAS umožňuje komunikovat s prvky po jednom vodiči a rozlišovat je pouze vnitřní 48bitovou adresou, která je dána z výroby. Tato sběrnice se nejčastěji používá u obvodů realizující elektronickou identifikaci, adresovatelných spínačů, dotykových pamětí, teplotních senzorů a dalších.

Základními sekvencemi komunikace po této sběrnici jsou RESET/PRESENCE, ZÁPIS BITU a ČTENÍ BITU. Na obrázku 1.13 je zobrazen průběh pro RESET a čtení PRESENCE. Tím celá komunikace začíná. Master pošle RESET a čeká na PRESENCE. Poté může zahájit vlastní přenos bitů a čtení dat z připojeného obvodu.

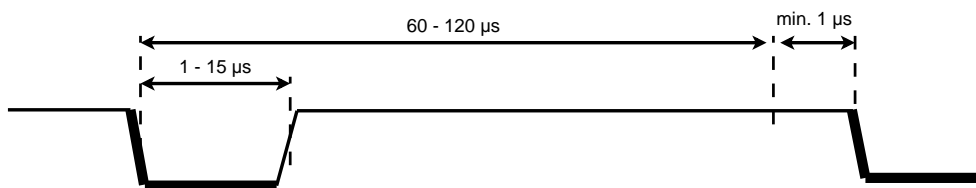


Obr. 1.13: Časový průběh zahájení komunikace pomocí RESET a PRESET sekvence [7]

Dalšími důležitými sekvencemi jsou odeslání logické úrovně nula (obr. 1.14) nebo jedna (obr. 1.15) a taky jejich příjem (obr. 1.16). Pro odeslání logické úrovně, se linka přepne z úrovně H do úrovně L na dobu 1 až 15 μs , poté se nastaví podle logické úrovně (pro log. 1 je to úroveň H, pro log. 0 úroveň L) na dobu 45 až 105 μs . Nakonec se minimálně na 1 μs nastaví linka do úrovně H a přejde zpět do úrovně L. Pro příjem logické úrovně je situace podobná. Jen logickou úroveň nenastavuje mikrokontrolér, ale vysílací obvod a mikrokontrolér úroveň čte.



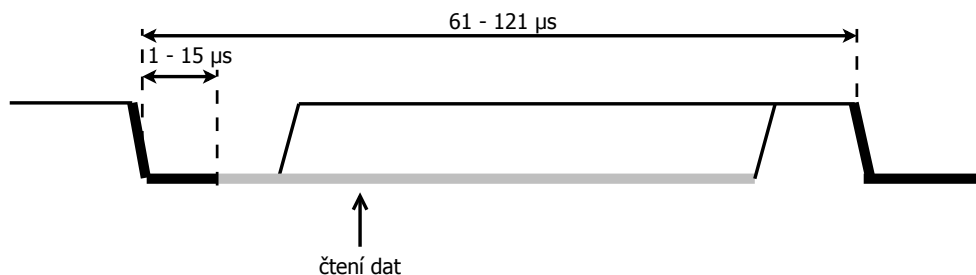
Obr. 1.14: Časový průběh zápisu log. 0 [7]



Obr. 1.15: Časový průběh zápisu log. 1 [7]

Výhodou této sběrnice je možnost použití datového vodiče i jako napájení. Oproti tomu nevýhodné je, že sběrnice umožňuje pouze poloviční duplex a tak není vhodná pro větší objem dat. [7]

Tato sběrnice se často používá i ke komunikaci s SD kartou.



Obr. 1.16: Časový průběh čtení logické úrovně [7]

1.3 Ethernet

Existuje mnoho hardwarových možností pro připojení zařízení do sítě LAN. Vybral jsem několik z nich a popíšu jejich vlastnosti, výhody a nevýhody.

1.3.1 ENC28J60

První vyzkoušenou možností pro připojení zařízení do sítě LAN je IO ENC28J60 od firmy Microchip. Tento obvod realizuje pouze fyzickou vrstvu síťového modelu ISO OSI a ostatní vrstvy je nutné doprogramovat nebo použít již naprogramované knihovny.

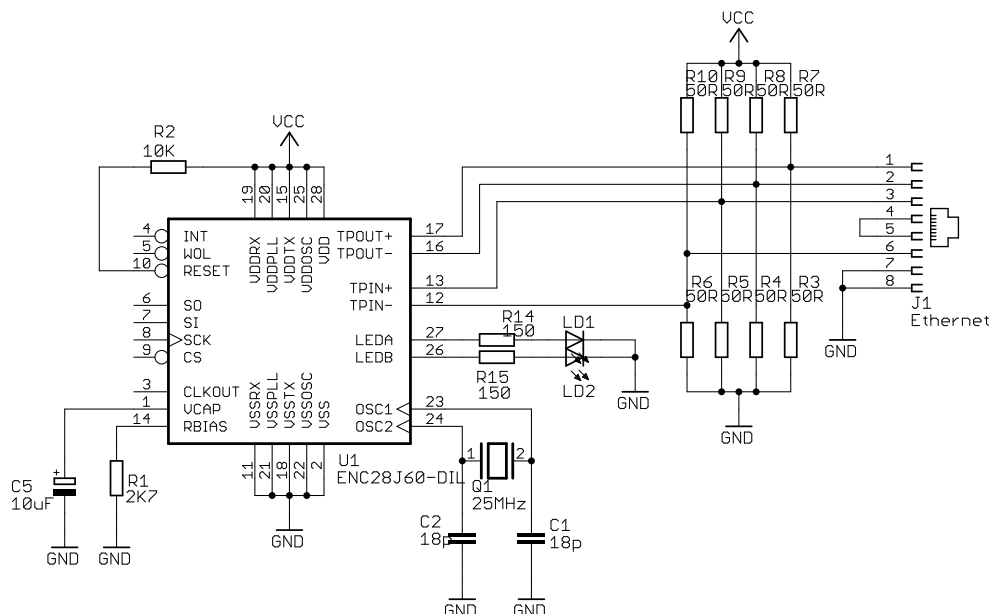
Pro komunikaci s mikrokontrolérem slouží rozhraní SPI, přes které se posílají data do jednotlivých registrů integrovaného obvodu ENC28J60.

Pro vyzkoušení obvodu v praxi byl vytvořen veřejný projekt z literatury [19], který realizuje základní protokoly pro síťovou komunikaci. Jednalo se o protokoly IP, TCP, UDP a HTTP.

Velkou výhodou je, že je velice levný a také je dostupný v pouzdře DIL, které je vhodné do nepájivého kontaktního pole pro testování. Navíc ke své funkci nepotřebuje příliš mnoho dalších součástek, což je patrné i ze schématu na obrázku 1.17. Další výhodou tohoto obvodu je možné použití i s mikrokontroléry, které nemají hardwarově implementovanou fyzickou vrstvu MAC, protože komunikace probíhá výhradně přes SPI a vše ostatní už řeší obvod ENC28J60.

1.3.2 Lantronix XPORT

Další možností, jak připojit zařízení do sítě LAN, je využití hotového modulu od firmy Lantronix. Tato firma nabízí velké množství produktů zaměřených právě na konektivitu. V nabídce můžeme najít již hotové převodníky z rozhraní RS232 na ethernet, ale také základní moduly pro komunikaci po síti LAN. Moduly mohou



Obr. 1.17: Základní zapojení obvodu ENC28J60

být různé. Některé disponují napájením po ethernetu (PoE) a některé mají dokonce paměť na uchování malé webové stránky a chovají se jako webový server.

Jedním z řady modulů, které firma Lantronix nabízí, je malý webový server. V nabídce je možné jej najít pod označením XPORT (obr. 1.18). Libovolné zařízení k němu můžeme připojit snadno pomocí rozhraní UART a přenášet data z webové stránky, ke které bude přistupovat uživatel.



Obr. 1.18: Lantronix XPORT

Výhodou je snadné a rychlé nasazení modulu. Není potřeba nic složité programovat. Ale velkou nevýhodou je jeho cena, která u většiny modulů převyšuje 1 000 Kč a tím se podstatně zvyšuje i cena celého zařízení. Výhodou je také velikost modulu, která je jen o málo větší než ethernet konektor RJ45.

Vlastnosti modulu XPORT

- Modul integrovaný v konektoru RJ45
- Webový server
- Snadná konfigurace přes webové rozhraní
- Jednoduchá úprava HTML stránek
- Zabezpečení heslem
- Napájení 3,3 V
- Převodník UART na Ethernet
- Podpora velkého množství protokolů (ARP, UDP, TCP, ICMP, Telnet, TFTP, DHCP, HTTP, SNMP, SMTP)
- Podpora posílání E-mailů [9]

1.3.3 LAN8720

Poslední obvod, který jsem vyzkoušel na připojení zařízení do sítě LAN byl LAN8720 od firmy SMSC. Obvod byl vyzkoušen na základové desce ke kitu STM32F4 Discovery.

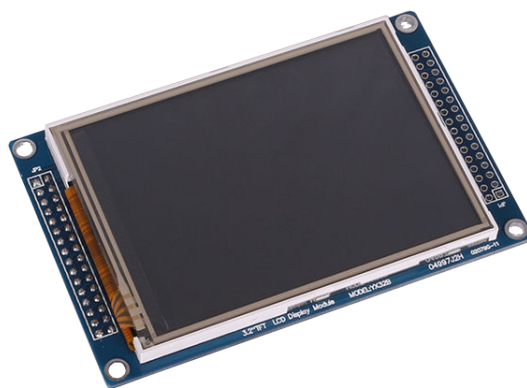
Stejně jako IO ENC28J60 realizuje tento obvod jen fyzickou vrstvu síťového modelu OSI ISO, ale rozdílem je komunikace s mikrokontrolérem, která probíhá pomocí speciálního sériového rozhraní MDIO. Proto je také potřeba, aby mikrokontrolér měl hardwarovou podporu ethernet vrstvy a dokázal komunikovat po rozhraní MDIO.

Obvod se používá hodně i v praxi např. v síťových tiskárnách, moderních televizorech, VoIP telefonech, herních konzolách, modemech, routerech a ve spoustě dalších zařízeních, kde je potřeba komunikovat po síti LAN.

Pro nastavení komunikace a celé fyzické vrstvy je v obvodu několik registrů, které jsou blíže specifikováni v literatuře [16].

1.4 LCD TFT displej

K vývojovému kitu byl dodáván displej s řadičem SSD2119, pro který firma STM dodávala i příslušné knihovny na ovládání. V mé práci byl však zvolen jiný displej s řadičem SSD1289 (obr. 1.19). Důvodem byla především jeho o dvě třetiny nižší cena. Tento displej byl použit i při testování webového serveru s mikrokontrolérem Atmel, ale pro tento mikrokontrolér je použití takového displeje nevhodné. Zvláště z důvodu malé vnitřní paměti. Pro displej je totiž nutné vygenerovat soubory s písmem a pro každou velikost a typ písma je nový soubor, který je plný bitmapových popisů jednotlivých písmen. Proto v případě použití displeje s mikrokontrolérem Atmel, byly vybrány pouze znaky, které jsou potřeba k výpisu.



Obr. 1.19: LCD TFT displej s řadičem SSD1289

Oba použité displeje jsou dotykové s odporovým snímáním a liší se mezi sebou ještě řadičem dotykové plochy. Displej dodávaný firmou STM obsahuje řadič, který komunikuje s okolím pomocí sběrnice I²C a displej, který jsem zvolil já disponuje řadičem s komunikací po SPI sběrnici.

Řadiče obrazu jsou velice podobné a tak zaměnění displeje nebyl takový problém. Stačilo jen trochu poupravit registry.

1.4.1 Komunikace s displejem

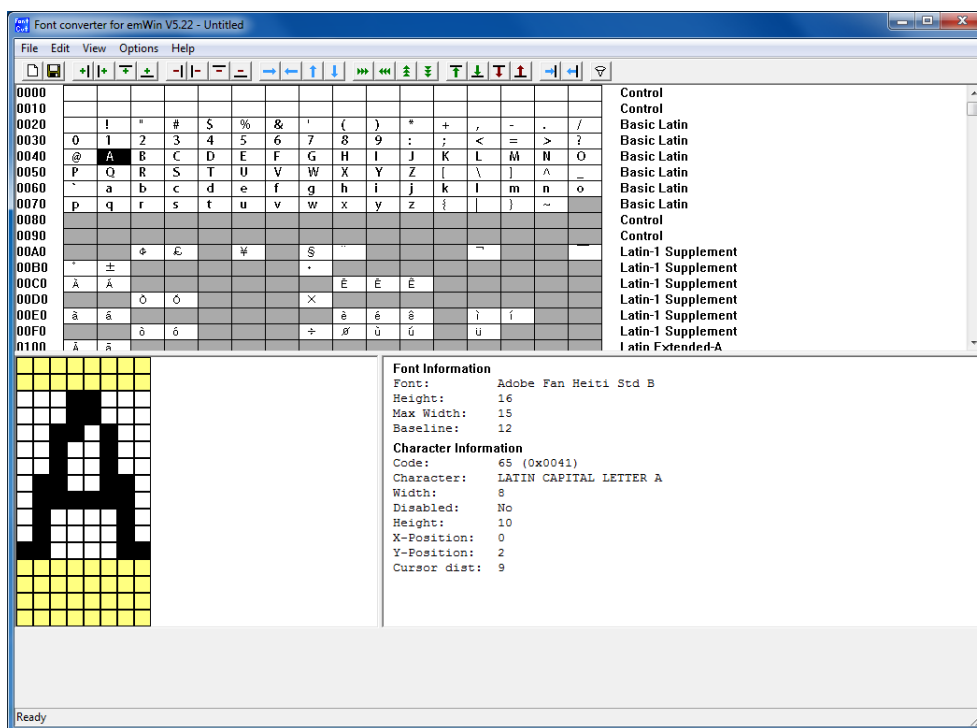
Komunikace s displejem probíhá paralelně po 16 datových vodičích a šesti vodičích pro nastavení komunikace. Jedná se o piny RESET, WR, RD, RS, CS a BL_CNT. Pin RESET slouží pro reset čipu a je aktivní v log. 0. Pro zápis dat do paměti čipu slouží pin WR, který je aktivní také v log. 0. Dalším pinem je RD, který slouží pro čtení dat z paměti čipu. Tento pin je aktivní v log. 1. Pro výběr aktivního čipu slouží CS pin, který má být nastaven na log. 0 při komunikaci s čipem. Dále máme možnost číst a zapisovat buď data nebo příkazy. Pro odlišení slouží pin RS. Pro data je v log. 1 a pro příkazy v log. 0. Posledním významným pinem je BL_CNT, kterým se pomocí PWM nastavuje jas podsvícení displeje.

Pro komunikaci mikrokontroléru s čipem na LCD displeji je možné použít rozhraní pro přístup k 16bitovým pamětem. Jedná se o rozhraní Flexible Static Memory Controller (FSMC), které má mikrokontrolér hardwarově implementované na čipu a tím je celá komunikace o mnoho jednodušší a rychlejší. V mikrokontroléru se předpřipraví celý blok dat do paměti, která je na to dostatečně velká a poté se jen pošle celý blok po 16 vodičích přímo do čipu na LCD displeji a tím se vykreslí.

1.4.2 Generování souborů s písmem

Jak již bylo řečeno v předchozích odstavcích, tak pro výpis textu na grafický displej je nutné mít vygenerované soubory s písmem, které popisují bitmapově jednotlivé znaky. Tyto soubory je možné snadno vytvořit programem v MS Windows a poté je nainportovat do vlastního projektu. Takových programů je na internetu spousta. V mé práci byl použit Font Converter for emWin V5.22 (obr. 1.20), který je také dodáván ke grafické knihovně STemWin.

Program umožňuje vybrat libovolné písmo z MS Windows, zvolit jestli má být tučné nebo kurzíva a vybrat znaky, které se mají generovat do souboru s písmem. Také je možné každý znak písma libovolně upravit, jak je vidět na obrázku 1.20. Program nám vygeneruje soubor *.c, který nainportujeme do vývojového prostředí. Poté máme možnost písmo vypsat na připojeném displeji.



Obr. 1.20: Font Converter for emWin V5.22

1.4.3 Generování souborů s obrázkem

Stejně jako soubory s písmem, můžeme vygenerovat soubor s obrázkem, který chceme vykreslit na displeji. Postup je podobný, jen k tomu slouží jiný program. Také je dostupný s knihovnou STemWin a jmenuje se Bitmap Converter for emWin V5.22. Nástroj umožňuje ještě před uložením souboru *.c obrázek libovolně upravit. Před

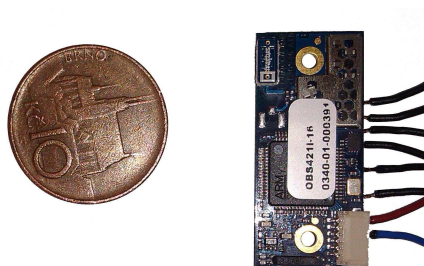
vygenerováním samotného souboru se nás program zeptá v jakém formátu má obrázek uložit. Umožňuje formáty s různým počtem barev.

1.5 Bezdrátové moduly

V mé práci jsem se setkal se třemi bezdrátovými bluetooth moduly. Tyto moduly slouží ke sběru dat z jednotlivých místností do řídicího termostatu. Mají za úkol posílat teplotu.

1.5.1 OBS421

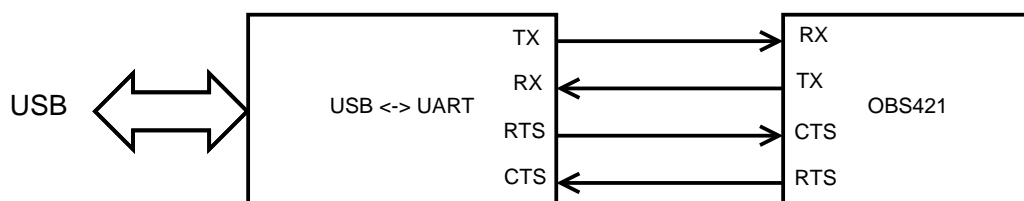
Základním článkem celého sběrného systému je modul OBS421 (obr. 1.21). Komunikace s modulem probíhá po sériové sběrnici UART, po které se posílají AT příkazy pro nastavování modulu a GATT příkazy pro komunikaci s ostatními moduly. Modul umožňuje komunikaci se dvojím typem bluetooth modulů. Prvním typem je klasický bluetooth používaný v mobilních telefonech a notebookech, druhý typ je označován jako Bluetooth Low Energy (BLE). Velkou výhodou těchto modulů je jejich spotřeba, která nepřesahuje 18 mA a v průměru dosahuje pouze 8 mA.



Obr. 1.21: Modul OBS421

Pro základní nastavení modulu je potřeba jej připojit k počítači pomocí USB nebo rs232 portu. Na obrázku 1.22 je blokově naznačeno připojení modulu k počítači pomocí USB. Pro připojení je také nutné znát základní nastavení portu jako rychlost datového toku a řízení toku. To je uvedeno v tabulce 1.1.

Modul může pracovat v několika různých režimech a je možné nastavit i různý počet klientů, ke kterým se připojí. V defaultním režimu umožňuje pouze přenos dat mezi dvěma klasickými bluetooth moduly. Ale v mé práci jsem potřeboval s modulem vytvořit síť tak, že bude jako master a ostatní budou klienti, které bude spravovat a přijímat od nich informace. To je možné dvojím způsobem. Buď na dotaz každého klienta nebo každý klient pošle data vždy při změně.



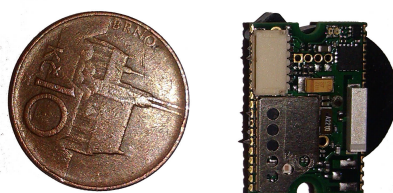
Obr. 1.22: Blokové schéma připojení modulu OBS421 k počítači

Tab. 1.1: Základní nastavení portu pro komunikaci s bluetooth modulem

Baud rate	57 600 Bd
datových bitů	8 bitů
Parita	žádná
Řízení toku	HW RTS/CTS

1.5.2 OLP425

Dalším modulem v mé práci je OLP425 (obr. 1.23), který se dost liší od OBS421. Je to také BLE modul, ale neumožňuje připojení ke klasickému bluetooth. Navíc však disponuje mikrokontrolérem od firmy Texas Instruments, ke kterému je připojeno teplotní čidlo, akcelerometr a dvě LED diody. Mikrokontrolér obsahuje defaultně demo program, který postačí pro mé řešení. Jinak je možné program přepsat a upravit. Mimo to, je modul napájen z knoflíkové baterie s napětím 3V.

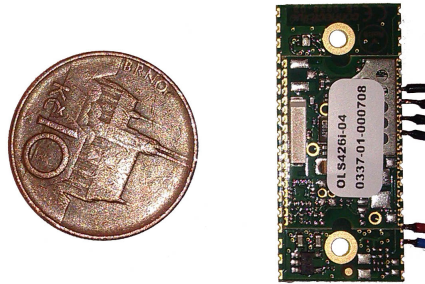


Obr. 1.23: Modul OLP425

Demo program umožňuje číst hodnoty akcelerometru, teplotního senzoru i LED diod pomocí GATT příkazů a nastavovat hodnoty LED diod. Také umožňuje nastavit pravidelné posílání teploty při změně. Celý postup komunikace mezi jednotlivými moduly je popsán v kapitole 1.5.4.

1.5.3 OLS426

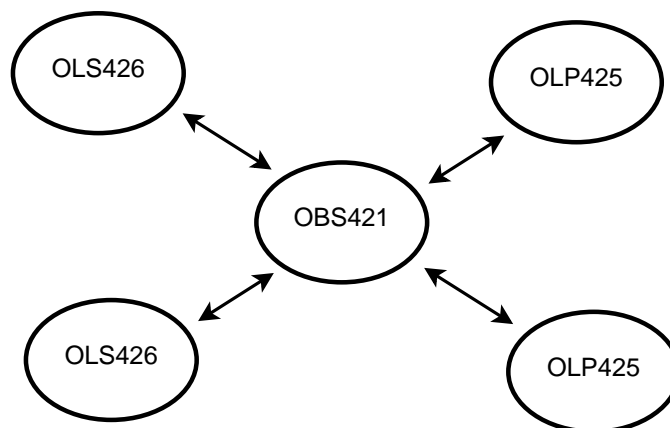
Posledním použitým modulem je OLS426 (obr. 1.24), který je prakticky totožný s OLP425 jen na desce nemá osazené teplotní čidlo, akcelerometr a pouzdro pro baterii.



Obr. 1.24: Modul OLS426

1.5.4 Nastavení a propojení jednotlivých modulů

Na obrázku 1.25 je vidět, jak jsou jednotlivé moduly propojeny a jak probíhá komunikace mezi nimi. Veškerou komunikaci řídí master modul OBS421 a ostatní moduly jsou mu podřízeny. Aby jsme dosáhli tohoto propojení je nutné modul OBS421 softwarově nastavit pomocí AT a GATT příkazů.



Obr. 1.25: Spojení bezdrátových modulů do sítě

Postup vytvoření sítě z bluetooth modulů

Pro spárování se s bluetooth moduly BLE je nutné mít v modulu OBS421 verzi firmwaru 5.1.0 nebo vyšší. Verzi firmwaru je možné zjistit pomocí AT příkazu `AT*AILVI?<CR>`. Nyní popíšu, jak spárovat dva moduly. Následující postup doplňuje obrázek A.1 v příloze.

1. Propojení modulu OBS421 s PC nebo jiným zařízením pomocí sériového portu.
2. Připojení k modulu (viz. kapitola 1.5.1).
3. Aktivace AT příkazů posláním `///` v čase kratším než 200 ms
4. Přepnutí modulu do BLE módu příkazem `AT*AGLE=1,1<CR>`.
5. Restart modulu (odpojení a znovu připojení napájení).
6. Znovu aktivace AT příkazů posláním `///` v čase kratším než 200 ms
7. Vyhledání okolních BLE zařízení příkazem `AT*AGI=6,4,0<CR>` (vrátí adresy v následujícím formátu: `*AGI:78c5e56c8979,0,1,"'OLP425-8979"',53`).
8. Spárování nalezeného zařízení příkazem `AT*ADCP=78c5e56c8979,17,0,0<CR>` (adresa `78c5e56c8979` je závislá na přijaté adrese z předchozího bodu).
9. V předchozím bodu bude vrácen ukazatel na spojení následovně `*ADCP:0`, kde 0 je tím ukazatelem.

Komunikace se spárovanými moduly pomocí GATT příkazů

Jak již bylo zmíněno v předchozích kapitolách, tak s BLE moduly je možné komunikovat pomocí GATT příkazů. V následujícím textu je popsáno několik příkladů. Některé z nich jsou graficky vyjádřeny na obrázku A.2 v příloze.

1. Po spárování modulů z předchozího příkladu přejdeme do GATT módu příkazem `AT*GATTD=0<CR>`, kde 0 je ukazatel na spojení s modulem.
2. Příkaz vypíše dostupné služby (při ponechání demo programu jsou ID služeb u všech modulů stejné jen se liší dostupné služby u modulů OLP425 a OLS426).
3. Nyní máme možnost přistupovat k jednotlivým periferiím.
4. **Rozsvítit zelenou LED:** `AT*GATTW=0,59,01<CR>`, kde první 0 je ukazatel na spojení s modulem.
5. **Zhasnout zelenou LED:** `AT*GATTW=0,59,00<CR>`, kde první 0 je ukazatel na spojení s modulem.
6. **Rozsvítit červenou LED:** `AT*GATTW=0,57,01<CR>`, kde první 0 je ukazatel na spojení s modulem.
7. **Zhasnout červenou LED:** `AT*GATTW=0,57,00<CR>`, kde první 0 je ukazatel na spojení s modulem.
8. **Číst aktuální teplotu:** `AT*GATTRU=0,51,54,FFE1<CR>`, kde první 0 je ukazatel na spojení s modulem. Modul vrátí `*GATTRU:0,53,19`, kde číslo 19 je

teplota v hexadecimálním kódu. Po převedení do desítkové soustavy to vychází na 25 °C.

9. **Číst aktuální stav baterie:** AT*GATTRU=0,27,31,2A19<CR>, kde první 0 je ukazatel na spojení s modulem. Modul vrátí *GATTRU:0,29,41, kde číslo 41 určuje stav baterie v hexadecimálním kódu. Po převedení do desítkové soustavy to vychází na 65 %.
10. **Nastavit automatické posílání teploty:** AT*GATTWC=0,54,01<CR>, kde první 0 je ukazatel na spojení s modulem. Modul poté vrátí teplotu ve stejném tvaru jako při dotazu, ale s tím rozdílem, že teplotu bude posílat při každé změně.

1.6 Měření teploty a vlhkosti

Pro realizaci termostatu bylo nutné měřit teplotu. Měření teploty probíhá několika teplotními senzory, které jsou popsány v následujících kapitolách. Důvodem využití více senzorů je vhodnost každého senzoru pro jinou část aplikace.

1.6.1 Teplotní čidlo DS18B20

Prvním zvoleným teplotním čidlem je DS18B20 od firmy Dallas, které komunikuje po sběrnici 1-wire. U obvodu je možné nastavit rozlišení teploty od 9bitového po 12bitové. Podle toho se odvíjí přesnost měřené teploty, ale také čas potřebný k převodu teploty na binární číslo. Závislosti doby převodu a přesnosti na počtu bitů jsou v tabulce 1.2.

Tab. 1.2: Závislosti doby převodu a přesnosti na počtu bitů [15]

Počet bitů	Doba převodu	Přesnost
9	93,75 ms	0,5 °C
10	187,5 ms	0,25 °C
11	375 ms	0,125 °C
12	750 ms	0,0625 °C

Dále je třeba určit typ komunikace, který může být dvojitý. První je závislý na adrese teplotního čidla. Používá se obvykle při připojení více zařízení na jednu linku 1-wire. Druhý je možné použít pouze v případě, když na lince je připojeno jen jedno zařízení a pak není nutné adresovat, ale komunikujeme pouze s jedním čidlem.

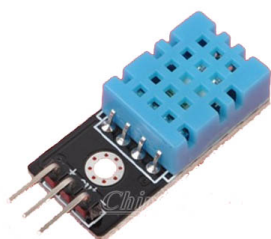
V mé práci je připojeno pouze jedno teplotní čidlo na sběrnici 1-wire, proto je možné využít jednodušší způsob komunikace bez adresování. [15]

Popis komunikace s obvodem DS18B20 pro přechtení teploty

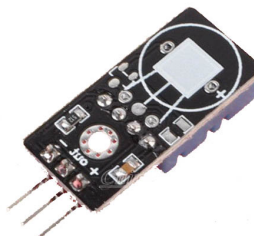
1. Provedeme inicializaci sběrnice 1-wire podle kapitoly 1.2.8.
2. Pošleme bajt `0xCCh`, který přeskočí adresaci.
3. Dále provedeme převod teploty na binární číslo posláním bajtu `0x44h`.
4. Poté je nutné znovu provést inicializaci podle bodu 1 a přeskočit adresaci podle bodu 2.
5. Posláním bajtu `0xBEh` oznámíme obvodu, že má poslat změřenou teplotu.
6. Obvod pošle ve dvou bajtech naměřenou teplotu. [15]

1.6.2 Čidlo teploty a vlhkosti DHT11

Pro měření vlhkosti byl zakoupen ještě modul s teplotním čidlem DHT11 (obr. 1.26). Hlavním důvodem zakoupení tohoto čidla bylo, že umožňuje měřit i vlhkost. Nevýhodou teplotního čidla je, že má přesnost naměřené teploty pouze 2°C a rozsah měřené teploty je od 0 do 50°C . Pro senzor vlhkosti je přesnost 5% a rozsah od 20 do 90% . Tyto hodnoty jsou pro moji práci dostačující.



(a)



(b)

Obr. 1.26: Senzor teploty a vlhkosti DHT11

Jak by se na první pohled zdálo, tak komunikace senzoru s mikrokontrolérem neprobíhá pomocí sběrnice 1-wire, ale je velice podobná. V zásadě se liší pouze v posílání log. 1 a log. 0. V obou případech totiž signál přechází do úrovně H a logické úrovně se odlišují pouze délkou trvání úrovně H. Pro log. 1 je délka trvání úrovně H přibližně $70\ \mu\text{s}$ a pro log. 0 mezi 26 a $28\ \mu\text{s}$.

Dalším rozdílem oproti obvodům podobného typu je adresace. Obvod totiž žádnou adresaci nepodporuje. Mikrokontrolér pošle pouze start signál a poté modul

DHT11 odešle celý rámeček dat, který obsahuje naměřenou vlhkost, teplotu a kontrolní součet. Rámeček je zachycen na obrázku 1.27. Výpočet kontrolního součtu se provádí tím, že se sečtou 8bitové bloky vlhkosti a teploty. [5]

data vlhkosti (jednotky) 8 bitů	data vlhkosti (desítky) 8 bitů	data teploty (jednotky) 8 bitů	data teploty (desítky) 8 bitů	kontrolní součet 8 bitů
------------------------------------	-----------------------------------	-----------------------------------	----------------------------------	----------------------------

Obr. 1.27: Datový rámeček modulu DHT11

1.6.3 Teplotní čidlo na bluetooth modulu

Poslední teplotní čidlo bylo zvoleno již výrobcem bluetooth modulu OLP426. Modul má na desce přímo z výroby čidlo TMP112. Výrobcem senzoru je firma Texas Instruments. Výrobce zvolil právě toto čidlo, protože má velmi nízkou spotřebu energie a tak je vhodné pro bluetooth modul.

Teplotní čidlo komunikuje po sběrnici I²C s mikrokontrolérem na bluetooth modulu. U čidla je možné nastavit přesnost na 12 nebo 13 bitů.

Čtení teploty

Komunikace probíhá standardně jako s většinou zařízení připojených na I²C sběrnici. Pro čtení teploty nejdříve master obvod pošle adresu zařízení, se kterým chce komunikovat a čeká na ACK od teplotního senzoru. U teplotních senzorů TMP112 je možné nastavit až 4 adresy podle tabulky 1.3 pinem A0. Poté master obvod zasílá identifikaci registru, který chce číst a znovu čeká na ACK od slave. Tím dá master obvodu slave najevo, že chce informaci z určitého registru. Dále master musí poslat znovu adresu slave obvodu a počkat na potvrzení ACK. V tomto okamžiku slave posílá dva bajty ze dvou registrů. Přijetí každého bajtu musí master potvrdit posláním ACK. [18]

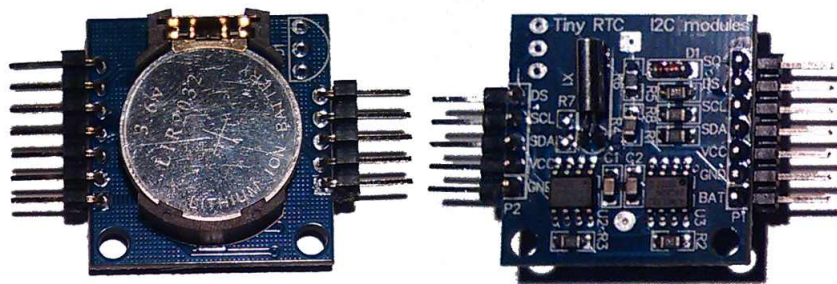
Tab. 1.3: Závislosti doby převodu a přesnosti na počtu bitů [18]

Adresa	Připojení pinu A0
1001000	GND
1001001	VCC
1001010	SDA
1001011	SCL

1.7 Datum a čas

Pro zjišťování reálného času jsem použil obvod DS1307, který je připojený k mikrokontroléru pomocí I²C sběrnice. K obvodu reálného času je navíc připojen akumulátor s napětím 3 V pro udržení a běh času i po odpojení napájení.

Hodnoty jsou uloženy v registrech, které mikrokontrolér může kdykoliv číst pomocí jednoduchých příkazů po sběrnici I²C .



Obr. 1.28: Modul s obvodem ds1307 z obou stran

Na vyzkoušení a praktickou realizaci jsem zakoupil hotový modul k Arduino (obr. 1.28), na kterém je realizováno základní schéma obvodu DS1307, připojen akumulátor a navíc obsahuje i externí paměť. [14]

Popis komunikace s obvodem DS1307

1. Master pošle 7bitovou adresu obvodu ds1307, která je 1101000 a k ní přidá jeden bit, který značí jestli se jedná o čtení nebo zápis. V případě výběru registru se jedná o zápis. Pro zápis se posílá log. 0.
2. Master posílá 8bitovou adresu registru do kterého chce zapisovat nebo číst.
3. Master znovu pošle 7bitovou adresu obvodu a přidá bit pro čtení/zápis.
4. Tento krok je odlišný pro zápis a čtení. Při zápise, master posílá hodnotu, kterou chce poslat do registru a při čtení slave posílá hodnu registru. [14]

1.8 Android

V průběhu realizace a testování mé práce jsem se rozhodl celý termostat řídit i mobilním telefonem se systémem Android. Požadavkem bylo implementovat jednoduše základní funkce termostatu do programu, který bude mít uživatel nainstalovaný ve svém mobilním telefonu nebo tabletu.

Operační systém android pro mobilní telefony a tablety vyvinula společnost Google a pro programování zdarma dává jak základní balíčky funkcí, tak i návody k nim pro programovací jazyk JAVA a vývojové prostředí Eclipse.

Balíčky umožňují snadný přístup ke všem funkcím operačního systému Android a tak je programování aplikace jednodušší. Pro přístup k některým funkcím je ovšem nutné ještě požádat o svolení uživatele při instalaci aplikace, jinak funkce nebude fungovat správně. Jedná se např. o funkce připojení k internetu, posílání SMS zpráv, zjišťování stavu připojení a další.

Změnou oproti klasickému programování je použití procesů na pozadí. Např. při připojení k internetu se veškerá komunikace a přenosy dat provádějí na pozadí a mezitím může aplikace provádět další úkony nebo uživatel má možnost do aplikace libovolně zasahovat.

2 PRAKTICKÁ REALIZACE

Prakticky byla vyzkoušena řada modulů a možností realizace, ale jako kompletní produkt termostatu řízeného přes internet byly vytvořeny dva prototypy. Jeden je s mikrokontrolérem Atmel a druhý s mikrokontrolérem STM32F417VG (původně testováno s vývojovým kitem STM32F4 Discovery). V následujících kapitolách jsou popsány obě zařízení a zhodnoceny výhody i nevýhody obou produktů.

2.1 Termostat s mikrokontrolérem ATMEL

Pro první pokus realizace termostatu řízeného přes internet byl zvolen mikrokontrolér Atmel, protože je cenově dostupný a snadno programovatelný. Mikrokontrolér nemá hardwarovou implementaci ethernet rozhraní, tak bylo nutné využít nějaký převodník. K dispozici byl modul Lantronix XPORT, který byl cenově nedostupný a integrovaný obvod ENC20J60. Modul ENC20J60 byl nakonec i použit. Jak již bylo uvedeno v úvodu, obvod komunikuje s mikrokontrolérem pomocí rozhraní SPI.

Zmíněný obvod realizuje pouze fyzickou vrstvu internetové komunikace a proto bylo nutné implementovat softwarově protokoly potřebné ke komunikaci. Pro vyzkoušení byl využit veřejný projekt z webu [19] realizující protokoly TCP, UDP, IP, ARP, HTTP a webový server.

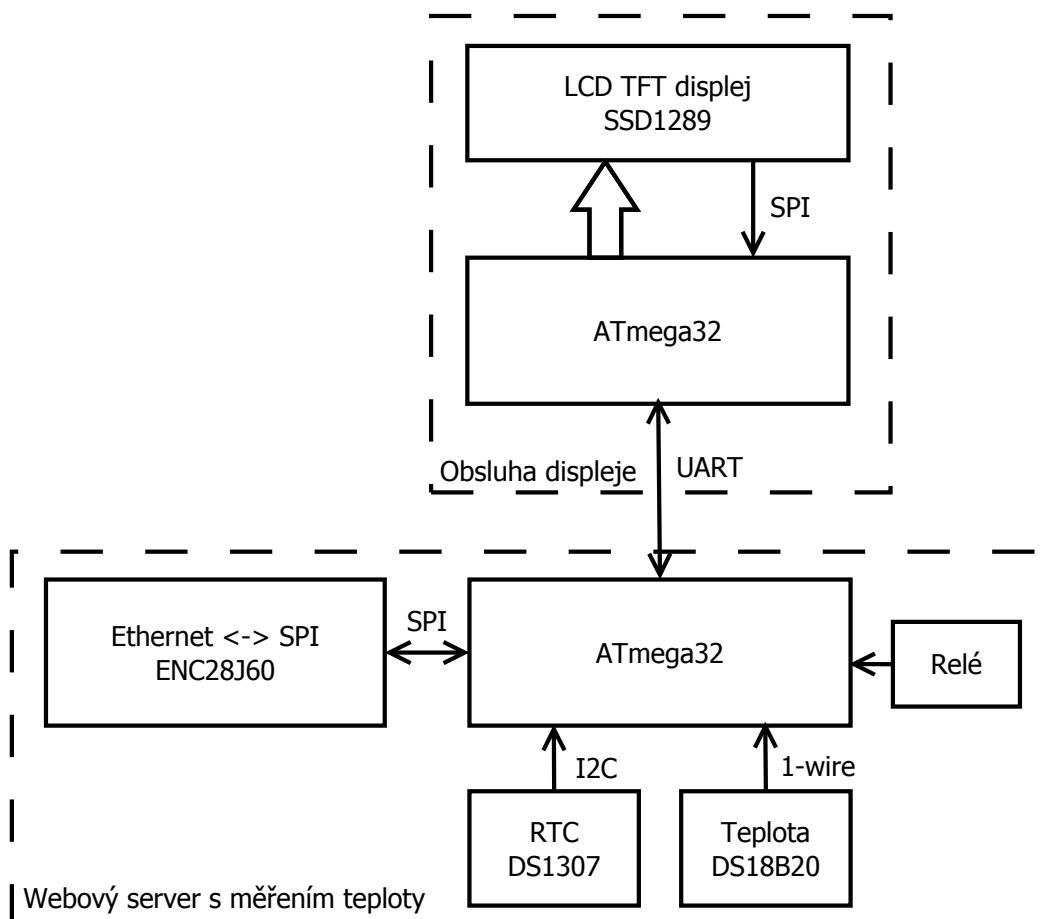


Obr. 2.1: Zobrazení nastavené a změřené teploty na TFT LCD dotykovém displeji

K webovému serveru s mikrokontrolérem bylo nutné připojit a doprogramovat další moduly. Nejdůležitějším prvkem bylo měření teploty. Byl použit senzor DS18B20 popsáný v úvodu a k němu napsána knihovna pro komunikaci. V programu kromě měření teploty byla napsána rutina pro porovnávání teploty nastavené (z webové stránky nebo displeje) a změřené. Jestliže změřená teplota klesla o 1 °C na více než jednu minutu, poté se sepnulo relé, které má sepnout kotel. V případě, že

místnost se vyhřála na teplotu o 1 °C vyšší než byla nastavená, pak se relé vypnulo a kotel přestal topit.

Další důležitou částí termostatu je informace o datu a čase pro nastavení různých teplot pro různé dny a hodiny. K tomu byl použit integrovaný obvod DS1307, který udržuje datum a čas i po odpojení napájení díky připojenému akumulátoru. Ale týdenní program stejně nebylo možné realizovat, protože mikrokontrolér už neměl dostatek volné paměti na celou tabulku hodnot.

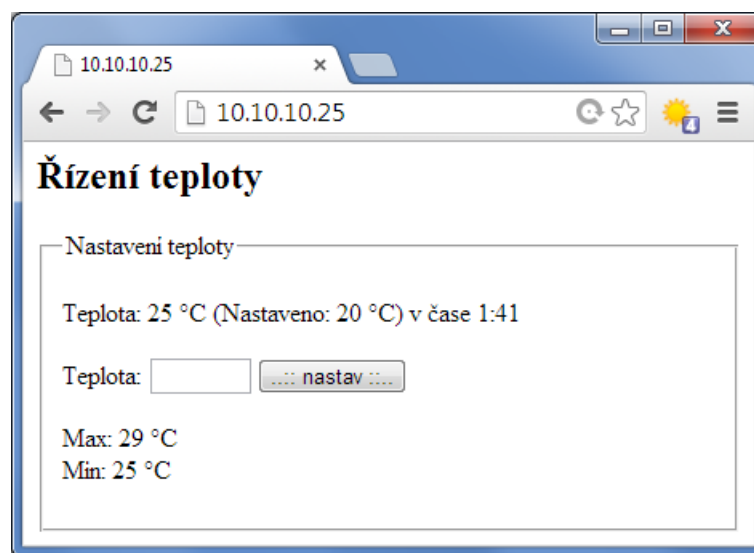


Obr. 2.2: Blokové schéma vytvořeného zřízení s mikrokontrolérem Atmel

Pro případ, že by uživatel nechtěl využít webové rozhraní pro nastavení teploty, byl připojen displej k rychlému zjištění naměřené a nastavené teploty. Nejprve byl použit znakový LCD displej, ale pro rychlé nastavení by bylo nutné k zařízení připojit ještě tlačítka. Proto byl nakonec použit dotykový LCD TFT displej s řadičem SSD1289 (Obr. 2.1). Při objednání z Číny je jeho cena nízká a tak se celková cena zařízení zvýší jen nepatrně. Problém byl, že komunikace s displejem probíhá paralelně po 16 vodičích a informace z dotykové plochy je navíc posílána po SPI sběrnici, proto bylo nutné využít ještě jeden mikrokontrolér. Stávající neměl dostatek vývodů

a také mu chyběla paměť, protože pro výpis znaků na displeji je potřeba vytvořit knihovnu s grafickým popisem písma a dalších znaků, která je příliš velká a spolu s programem pro webový server se do jednoho mikrokontroléru nevezla.

Napájení celého zařízení bylo realizováno po zbylých dvou párech UTP kabelu. K tomu bylo potřeba ještě vytvořit vlastní přizpůsobovací článek, který obsahoval stabilizaci napětí na 5 V a přizpůsobení na UTP kabel.



Obr. 2.3: Webová stránka pro nastavení teploty u termostatu s mikrokontrolérem Atmel

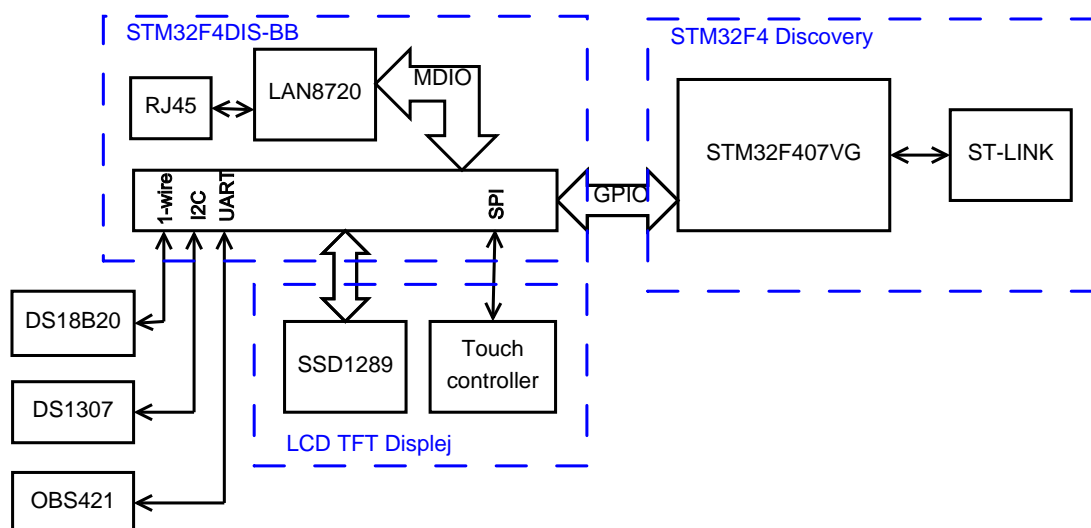
Celé zařízení bylo prakticky vytvořeno a vyzkoušeno. Na obrázku 2.2 je zobrazeno blokové schéma zařízení, které je rozděleno fyzicky na dvě části. První je část webového serveru a měření teploty a druhá část obsahuje komunikaci s displejem a obsluhu dotykové plochy. Pro každou část zařízení byla navržena a vyrobena vlastní deska. Tyto desky jsou propojeny sběrníci UART. Schémata obou desek jsou na obrázku B.2 a B.1 v příloze.

Zařízení bylo sice realizováno s minimálními náklady, ale má i několik chyb. Základním nedostatkem je interní paměť mikrokontroléru, která je na webové stránky nedostatečná. Bylo možné zobrazit pouze malou webovou stránku (obr. 2.3). Problém byl také v tom, že server umožňuje odeslat pouze jeden TCP paket s daty. Tím se velikost webové stránky značně zmenšila.

2.2 Termostat s vývojovým kitem STM32F4 Discovery

Po ne zcela úspěšném pokusu termostatu s mikrokontrolérem Atmel, se má práce ubírala zcela jiným směrem. Byl použit vývojový kit STM32F4 Discovery s mikrokontrolérem ATM32F407VG. Důvodem použití tohoto mikrokontroléru je především jeho rychlost, velikost paměti a hardwarová implementace spousty komunikačních rozhraní a periférií. Nejvíce záleželo na MAC vrstvě, kterou má mikrokontrolér implementovanou. Tím se celé zařízení zjednodušuje.

Pro jednoduchost testování a realizace první verze termostatu byla připojena základovou desku pro vývojový kit a LCD TFT displej. Blokové schéma celého zařízení včetně rozšiřujících desek je na obrázku 2.4. Ve schématu je i vyznačeno, které části jsou realizovány z vývojových desek a které jsou externě připojeny. Navíc hardwarové prvky, které nejsou v mé práci využity a vývojový kit je obsahuje, v blokovém schématu uvedeny nejsou.



Obr. 2.4: Blokové schéma termostatu realizovaného s vývojovým kitem STM32F4 Discovery

Blokové schéma na obrázku 2.4 není nijak složité a to především díky tomu, že mikrokontrolér obsahuje spoustu hardwarových periférií uvnitř a také je hodně realizováno pomocí softwaru v mikrokontroléru.

Celý program v mikrokontroléru je založen na real-time operačním systému FreeRTOS (viz. kapitola 1.2.2), který řeší základní funkce jako např. práci s procesy a vlákny. Vybrané procesy jsou popsány v kapitole 2.2.1. Jako jedno vlákno běží

webový server, který je realizován pomocí souboru knihoven LwIP stack (viz. kapitola 1.2.3).

Pro další periferie, jako jsou teplotní čidla, obvod hodin, bluetooth moduly a displej bylo nutné napsat separátní knihovny, které komunikují s vybranými zařízeními pomocí protokolů a sběrnic popsaných v úvodu.

2.2.1 Popis procesů

Funkčnost celého zařízení je rozdělena na několik základních částí, které jsou od sebe odděleny a vykonávají se nezávisle na sobě. Jen jejich výstup může být používán i v dalších částech. Tyto části jsou vykonávány v takzvaných procesech. Každý proces má svoji prioritu a vyhrazenou paměť. Procesy jsou vykonávány v různém pořadí a vytvářejí dojem multitaskingu. Kromě procesů jsou ještě složitější části programu, které jsou vykonávány ve vláknech. Ty mohou běžet zároveň s jednotlivými procesy. Funkce jednotlivých procesů jsou popsány v následujících kapitolách. Komunikace mezi procesy probíhá v celém programu pomocí globálních proměnných.

Proces blikání LED

Prvním nejjednodušším procesem je blikání LED diody. Jedná se o jednoduchou změnu stavu portu, na kterém je LED dioda. Proces slouží pro kontrolu funkčnosti celého zařízení. Vykonávání procesu se děje každých 250 ms.

Blikání LED je vidět pouze na vývojovém kitu a v hotovém zařízení již LED není připojena.

Proces měření teploty

Každé 2s se provádí proces čtení teploty z teplotního čidla a uložení teploty do proměnné. Zároveň se čtou hodnoty z vlhkostního čidla.

Proces porovnávání teplot

V tomto procesu se porovnávají naměřené a nastavené teploty a spíná se relé podle nastaveného režimu. Záleží jestli se jedná o týdenní program a také podle kterého pokoje se má relé spínat.

Naměřené teplota musí být o 0,5 °C vyšší než nastavená, aby se relé sepnulo. Při rozepnutí to funguje stejně.

Proces se vykonává každých 400 ms.

Proces kontrola času

Pro zjišťování data a času je potřeba v pravidelných intervalech číst data z obvodu DS1307 po sběrnici I²C . To má za úkol tento proces, který se vykonává každou sekundu a ukládá datum i čas do globální proměnné.

Proces vypisování na displeji

Dalším důležitým procesem je vypisování na displeji. Každých 300 ms se na displeji vykreslují všechny prvky včetně data, času, teploty a vlhkosti. Kromě vypisování má proces ještě za úkol snížit jas displeje po několika vteřinách a obsluhovat dotykovou plochu displeje.

Proces komunikace s bluetooth

V tomto procesu se komunikuje s bluetooth moduly a přijímají se data obsahující naměřenou teplotu. Proces se provádí každé 2 s.

Proces generování výstupních souborů

Posledním procesem je generování výstupních souborů pro webové stránky. Tento proces má urychlit komunikaci webového prohlížeče s webovým serverem. Server totiž není schopný soubor vygenerovat a poslat v dostatečně nízkém čase a prohlížeč mezitím vyhodnotí server jako nedostupný. Proto proces vygeneruje všechny webové stránky do paměti mikrokontroléru a server po vyžádání webové stránky pouze přečte z paměti data a pošle je prohlížeči. Webové stránky jsou obnovovány v paměti každých 300 ms.

2.2.2 Příjem POST a GET parametrů

Pro nastavování teplot a dalších možností termostatu bylo nutné v programu webového serveru dopsat funkce pro zpracování parametrů POST a GET.

Hlavním rozdílem těchto dvou parametrů je to, jakým způsobem se posílají a jak je vidí uživatel. V případě odeslání GET parametru, se objeví všechny hodnoty v adrese webu za symbolem otazníku. Jednotlivé parametry jsou poté odděleny symbolem & a hodnoty parametrů jsou přiřazeny za rovná se. Oproti tomu POST parametry běžný uživatel nevidí. Jsou totiž skryté v hlavičce, ale jinak je řetězec s daty úplně stejný.

V mém programu bylo nutné vytáhnout tyto parametry z přijatých dat a rozdělit je tak, aby hodnoty seděly k jednotlivým parametrům. K tomu byly napsány dvě funkce, které se liší jen tím, kde vyhledávají řetězec parametrů. U GET parametru je

řetězec ihned za adresou stránky, která data poslala a tato adresa je úplně na začátku hlavičky přijaté webovým serverem. Oproti tomu parametr POST je posílán až na konci přijaté hlavičky a nezačíná znakem otazníku.

Popis funkcí pro rozdělení parametrů

Pro získání GET parametrů musí funkce nejprve získat řetězec s parametry z hlavičky. Proto nejprve odřízneme vše, co je před otazníkem a vše, co je za následující mezerou. Tím zůstane řetězec podobný tomuto `?param1=10¶m2=50`. Poté je potřeba ještě odstranit znak otazníku. V dalším kroku se ve smyčce rozdělí celý řetězec podle znaku `&` a zároveň se rozdělené bloky rozdělí ještě podle rovná se. Tím dostaneme v každé iteraci funkce dvě hodnoty. Jedna je název parametru a druhá je jeho hodnota. Každou z hodnot ukládáme zvlášť do pole, ze kterého pak parametry zpracováváme.

U parametru POST se liší pouze získání celého řetězce parametrů. Ten získáme oddělením dvou řetězců, které jsou oddělené mezerou a oddělením znaků před symbolem `\r`. Poté ještě odebrání prvních 4 znaků, které jsou mezery. Stejně jako u parametru GET ještě oddělíme vše za následující mezerou. Tím získáme jen řetězec parametrů oddělených znakem `&`.

2.2.3 Zabezpečení

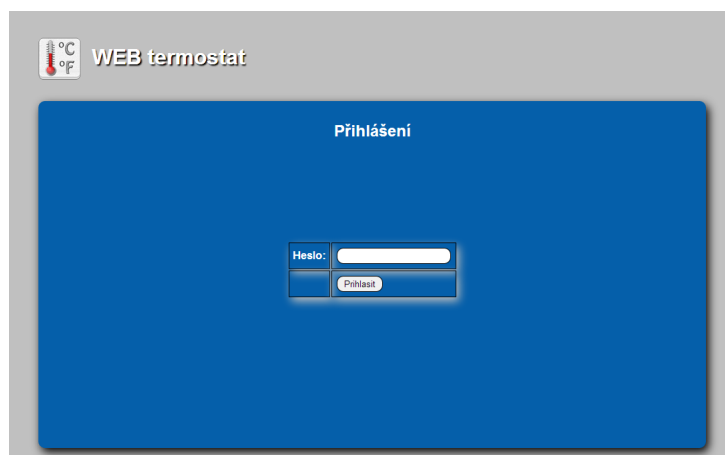
Přístup k webovému serveru i ke všem jeho službám je zabezpečen heslem, které je možné si nastavit na stránce s nastavením nebo na dotykovém displeji. Heslo je uloženo v paměti serveru a také v paměti obvodu DS1307, aby nedošlo ke ztrátě po vypnutí napájení. Pro přístup na každou webovou stránku serveru je tedy nutné znát nastavené heslo, ale heslo stačí zadat pro každé připojení jen jednou a poté už probíhá ověřování pomocí `session`.

Postup ověřování a zabezpečení pomocí `session` je velice jednoduchý v případě použití PHP serveru, ale můj webový server programovací jazyk PHP nepodporuje. Proto bylo nutné naprogramovat metodu, která umožňuje ověřit správnost `session`.

Celý proces ověřování začíná přihlášením. Po zadání hesla a odeslání formuláře webový server ověří správnost hesla. Jestliže se heslo shoduje s uloženým, poté pošle prohlížeči webovou stránku, která navíc v hlavičce obsahuje hash, pod kterým má server zaregistrovanou proměnnou. Přesné znění zaregistrování hash je zde: `Set-Cookie:PHPSESSID=123456789;path=/`, kde číslo 123456789 je posílaný hash. Poté už webový prohlížeč automaticky posílá stejný hash v každém dotazu o webovou stránku. Tím server pozná, že se jedná stále o jedno ověřené připojení a může poslat požadovanou stránku. V opačném případě server vrací stránku s přihlášením.

2.2.4 Způsoby ovládání termostatu

Aktuální verze termostatu umožňuje celkem tři způsoby nastavení teploty a získávání informací o teplotách, datu a času. Všechny tři způsoby nejsou úplně tradiční k ovládní termostatu, ale v dnešní době se stávají stále více běžnější a dostávají se i mezi uživatele, kteří využívají techniku méně. Prvním způsobem je nastavení z webového prohlížeče počítače, tabletu nebo mobilního telefonu. Jedná se o klasickou webovou stránku. Druhý způsob je za použití dotykového LCD TFT displeje, kde je možné provádět základní úkony s termostatem. A třetím způsobem je aplikace napsaná pro operační systém Android, který je v některých mobilních telefonech a tabletech.

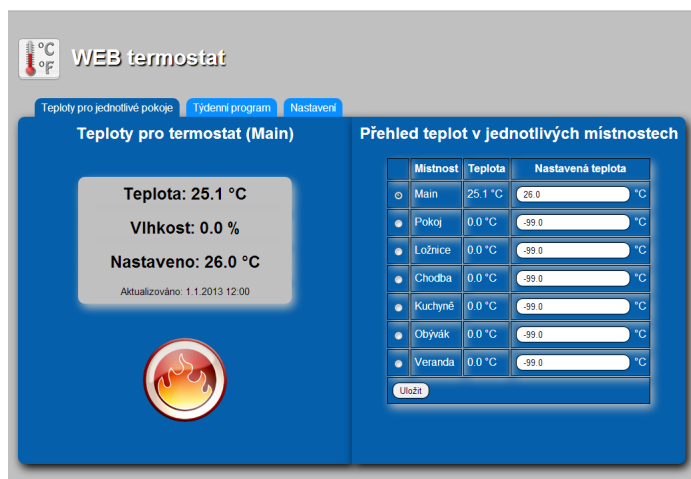


Obr. 2.5: Přihlašovací stránka

Webová stránka

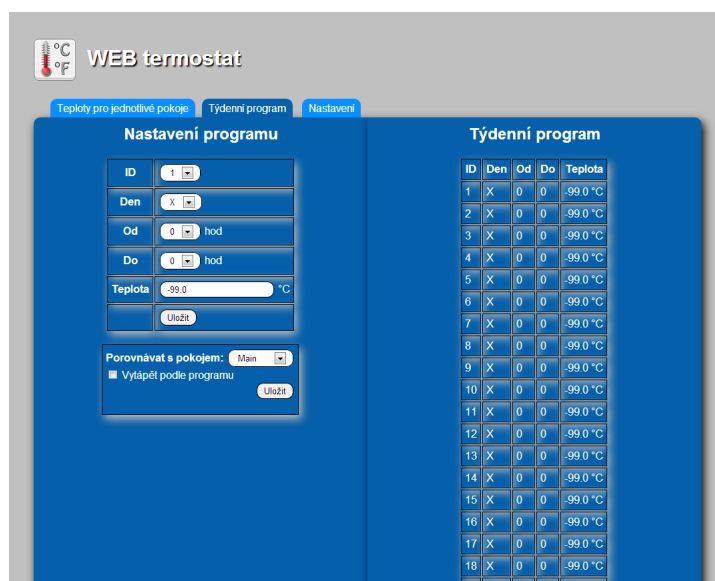
Základním rozhraním pro nastavení teplot v jednotlivých pokojích a zobrazení naměřených teplot je webová stránka. Na tu se dostaneme zadáním IP adresy webového serveru do webového prohlížeče, která je defaultně nastavena na **10.10.10.5**, ale je možné ji kdykoliv změnit na stránce s nastavením. Webový server obsahuje pouze několik stránek, které jsou potřeba k ovládní celého zařízení.

První stránkou, která se objeví po zadání IP adresy je přihlašovací okno (obr. 2.5). Je potřeba zadat heslo, které je nastaveno na **1234**, ale je možné ho v nastavení kdykoliv změnit. Po přihlášení se objeví stránka s přehledem (obr. 2.6), na kterou je možné se dostat zadáním adresy **/pokoje.html**. Vlevo je zobrazena naměřená a nastavená teplota v aktuálním pokoji a vlhkost naměřená termostatem, vpravo pak přehled naměřených a nastavených teplot ve všech pokojích. Navíc je napravo umístěn formulář, kterým je možné teploty nastavovat. Také je možné zvolit, který pokoj bude aktuální, což znamená, podle kterého pokoje se bude vytápět. Nalevo je



Obr. 2.6: Stránka s přehledem teplot

ještě zobrazen čas, kdy byly data vygenerovány, aby jsme měli jistotu, že jsou data aktuální. Pod datem a časem je zobrazena ikona ohně, která indikuje zda v tomto čase je sepnuté relé pro ovládání kotle.



Obr. 2.7: Stránka s týdenním programem

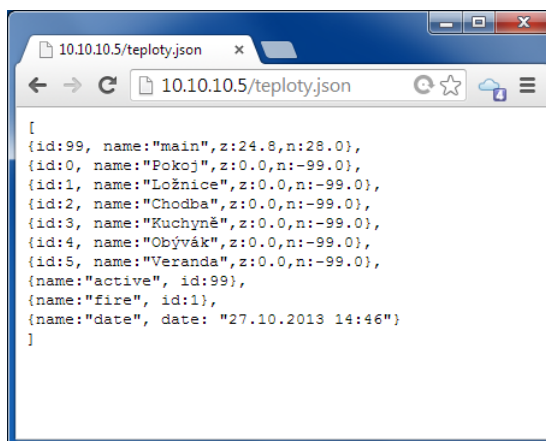
Další stránka, kterou můžeme najít pod adresou `/program.html`, je týdenní program (obr. 2.7). Je možné si navolit teplotu, den v týdnu a čas od kdy a do kdy má být zvolená teplota aktuální. Toto můžeme přidávat do tabulky celkem 20x. Důležité je, že týdenní program je vždy aktuální pouze na aktuální pokoj. Na stránce týdenního programu je také možné zvolit jestli vytápět podle zvoleného programu

nebo jen manuálním režimem na stránce s pokoji. Manuální režim se také nastaví automaticky vždy, když bude na stránce s pokoji nastavena ručně teplota.



Obr. 2.8: Stránka s nastavením

Na adrese `/nastaveni.html` je stránka s nastavením celého termostatu (obr. 2.8). Toto je jediná možnost, kde můžeme měnit veškerá nastavení. Ostatní způsoby ovládní termostatu neumožňují měnit všechna nastavení. Na stránce s nastavením je možné měnit bezpečnostní heslo, IP adresu zařízení, síťovou bránu, názvy pokojů, datum, čas a další možnosti.



Obr. 2.9: Stránka s daty ve formátu JSON

Důležitými stránkami především pro mobilní aplikaci a aplikaci pro sběr dat jsou stránky `/teploty.json` a `/set_teploty.json`. Obě stránky vrací stejné data, kterými jsou všechny teploty a pokoje v JSON formátu (obr. 2.9). Na obě stránky je možné se dostat pouze se správným heslem a to se přenáší GET parametrem

pass. Stránka s předponou set zpracovává ještě parametr `json`, ve kterém je možné předat nastavení teploty pro jeden pokoj. JSON pro nastavení musí vypadat takto: `[{id:1,n:20.5}]`, kde `id` je identifikační číslo pokoje a `n` je hodnota teploty, kterou chceme nastavit.

Poslední stránkou, se kterou se můžeme setkat je chybové hlášení, že požadovaná stránka neexistuje. A to nastane v případě, když požadujeme jinou stránku než je napsané v předchozích odstavcích.

Grafické rozhraní na displeji

V případě, že nechceme zapínat počítač a přesto potřebujeme nastavit teplotu nebo jen zjistit aktuální teplotu, postačí nám připojený dotykový TFT LCD displej. V hlavním zobrazení můžete vidět jen aktuální teplotu, naměřenou vlhkost, nastavenou teplotu, několik dotykových tlačítek, datum a čas. Displej umožňuje nastavení teploty tlačítky plus a mínus. Teplota je okamžitě akceptována.

Displej je možné přepnout do rozšířeného zobrazení, které zobrazí několik pokojů s naměřenými teplotami.

Navíc je možné na displeji nastavit heslo pro přístup a jas.

Aplikace pro Android

Poslední možností jak si nastavit teploty termostatu a zobrazit naměřené teploty je pomocí aplikace v chytrém mobilním telefonu s operačním systémem Android. V případě, že je aplikace správně nastavená (adresa k JSON stránce a heslo), poté se po spuštění zobrazí seznam pokojů s naměřenými a nastavenými teplotami podle obrázku 2.10a. Vždy první teplota je naměřená a druhá nastavená. Navíc před názvem aktivního pokoje se zobrazí šipka. Pro změnu nastavené teploty stačí kliknout na řádek s určitým záznamem a objeví se dialogové okno (obr. 2.10b), ve kterém je možné zapsat do textového pole teplotu a potvrdit tlačítkem OK. Poté se odešle nastavený údaj do termostatu a zároveň se nastaví zvolený pokoj jako aktuální. Po nastavení se v aplikaci obnoví seznam pokojů i s teplotami. Seznam je možné aktualizovat i ručně, kliknutím na menu a položku obnovit.

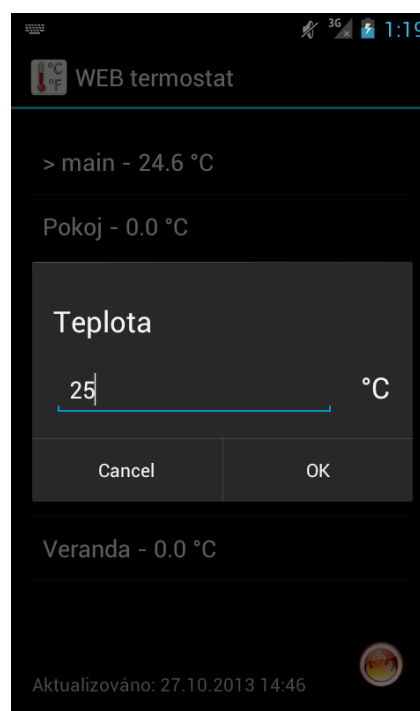
V menu je možné ještě otevřít nastavení, ve kterém můžeme měnit adresu JSON souboru a heslo.

2.2.5 Termostat s mikrokontrolérem STM32F417VG

Po vyzkoušení celého termostatu na vývojovém kitu bylo celé zařízení realizováno na jednu desku. Nejprve bylo nakresleno výsledné schéma v softwaru Eagle, které je rozděleno celkem do 5 částí a poté byla navržena výsledná deska. Deska byla



(a) Přehled teplot



(b) Nastavení teploty

Obr. 2.10: Aplikace pro Android

vyrobena v Praze ve společnosti PragoBoard. Po osazení a nahrání programu vzniklo výsledné zařízení, které je na obrázku 2.11. Oproti původnímu návrhu bylo ještě vyměněno vlhkostní čidlo DHT11 za DHT22 pro jeho větší přesnost. Komunikace zůstala stejná.

Schémata

Všechny části schématu jsou přiloženy v příloze. Na první části je mikrokontrolér STM32F417VG, zapojení programovacího konektoru a napájecí část celého zařízení. První část je na obrázku B.5. Druhá část (obr. B.6) zobrazuje připojení mikrokontroléru k obvodu pro komunikaci s internetem. Na další části schématu (obr. B.3) je znázorněno připojení LCD TFT displeje k mikrokontroléru. Čtvrtá část schématu (obr. B.7) obsahuje zapojení všech použitých senzorů, obvodu reálného času a výstupního relé. Na poslední části schématu (obr. B.4) je připojení bluetooth modulu k mikrokontroléru.

PCB

Deska plošného spoje z obou stran je na obrázku C.1 a C.2 a osazovací pláněk je na obrázcích C.3 a C.4.



Obr. 2.11: Hotové zařízení

2.3 Ukládání dat a zobrazování statistik

Separátní částí od termostatu jsou statistiky naměřených teplot. Pro jednoduchost jsem zvolil pro statistiky veřejný webový server, na kterém je uloženo několik PHP souborů.

2.3.1 Ukládání aktuálních hodnot do databáze

Pro statistiky je nejprve nutné mít uložené hodnoty teplot v jednotlivých okamžicích. Proto je na veřejném serveru uložen skript, který se spouští pomocí cronu každých 15 minut. Skript je naprogramovaný tak, aby přečetl JSON data z termostatu, dekodoval je a poté uložil do MySQL databáze s aktuálním datem a časem.

2.3.2 Zobrazování statistik

Pro zobrazení statistik je nejprve nutné přičíst data z databáze za dané období. Časové období, za které chceme statistiku zobrazit, je možné si zvolit pomocí formuláře na internetové stránce. Po přečtení dat, dostaneme pole dat, ve kterém jsou všechny teploty a pokoje. Data je poté možné vykreslit do grafu. V prvním grafu jsou vidět teploty ve všech místnostech a na dalším grafu jsou vlhkosti v jednotlivých místnostech. Je možné zobrazit graf teplot a vlhkostí pro jeden den, jeden měsíc a jeden rok. Webová stránka s grafy je zobrazena na obrázku 2.12.

Adresa internetové stránky s grafy je <http://termostat.vzap.eu/>.



Termostat - stats

Home Filtry

DEN: 1 MĚSÍC: 4 ROK: 2014 Zobraz



Obr. 2.12: Webová stránka s grafy

3 ZÁVĚR

Cílem mé práce bylo vytvořit termostat pro ovládání kotle v rodinném domku tak, aby byl řízený přes internet nebo počítačovou síť LAN. V první části mé práce bylo analyzováno několik možných řešení, jak připojit libovolné zařízení k internetu a všechny tyto možnosti jsou popsány v úvodu mé práce. Kromě připojení k internetu bylo vyřešeno měření teploty v pokoji a navíc bylo implementováno i měření vlhkosti. Výběr teplotního čidla nebyl nikterak náročný. Všechny zmíněné senzory byly vyzkoušeny a poté použity v mé práci.

Prakticky byly realizovány celkem dvě zařízení popsané v praktické části. První zařízení bylo se slabším mikrokontrolérem, na kterém byla ověřena funkčnost některých senzorů a modulů pro další použití s mikrokontrolérem STM32F417VG. Vyskytlo se několik problémů při realizaci prvního zařízení. Především se týkali nedostatku výkonu mikrokontroléru a také nedostatku paměti. Tu by bylo možné rozšířit např. SD kartou, ale výkon mikrokontroléru na to už nedostačuje. Proto první zařízení posloužilo jako testovací pro většinu modulů a pro porovnání s výsledným termostatem.

Pro nedostatky mikrokontroléru Atmel byl zvolen mnohem výkonnější mikrokontrolér STM32F407VG, který má jádro ARM Cortex-M4 a mnohonásobně více paměti. Další výhodou mikrokontroléru STM32F407VG je velké množství hardwarových periférií, které zrychlují činnost celé aplikace a zjednodušují program uložený v mikrokontroléru.

K vyzkoušení a porovnání s první verzí webového termostatu byl zvolen k realizaci vývojový kit STM32F4 Discovery a k němu rozšiřující desky pro komunikaci. Na tomto hardwaru je aplikován operační systém FreeRTOS s balíkem knihoven LwIP. Tento balík slouží k realizaci webového serveru, který naslouchá na portu 80 a podle webové adresy posílá prohlížeči danou webovou stránku. Na stránce je poté vidět nastavené a změřené teploty případně i vlhkost. Také je možné přes webovou stránku nastavovat tyto teploty a další hodnoty termostatu a posílat je webovému serveru.

Výsledkem je tedy zařízení spojující vlastnosti běžného termostatu a inteligentního řízení teploty s webovým serverem. Zařízení umožňuje jak spínání relé podle nastavené teploty, tak i nastavení teploty několika způsoby. Prvním způsobem, tím nejběžnějším, je dotykový displej. Dále je možné teploty nastavovat pomocí mobilního telefonu s operačním systémem Android. Třetí možností je webové rozhraní. To umožňuje kompletní správu termostatu, čímž je myšleno nastavování teploty, správa týdenního programu, změna síťových adres, nastavení data a času a další.

Termostat navíc umožňuje připojení bezdrátových teplotních senzorů pro měření teploty v jednotlivých pokojích nebo ve vnějších prostorách.

Všechny naměřené teploty a vlhkosti jsou ukládány na server do MySQL databáze a zpracovávány do grafu. To je vhodné především pro analýzu nákladů na vytápění a porovnání jak se mění vnitřní teplota oproti venkovní.

Na závěr, když porovnáme běžně prodávané termostaty s termostatem vytvořeným, tak je vidět rozdíl nejen v ovládnání webovým rozhraním a mobilním telefonem, ale také byla snaha o co nejjednodušší rozhraní. Některé termostaty jsou příliš složité nejen pro technicky vzdělaného člověka, ale také pro laika. Proto byla snaha ovládnání co nejvíce zjednodušit.

LITERATURA

- [1] Atmel Corporation *ATmega32* [online datasheet]. 2011, poslední aktualizace 26. 2. 2011 [cit. 4. 11. 2013]. Dostupné z URL: <<http://www.atmel.com/images/doc2503.pdf>>.
- [2] *CONRAD* [online]. 2013, [cit. 4. 11. 2013]. Nástěnný pokojový termostat, 10 až 30 °C, bílá. Dostupné z URL: <<http://www.conrad.cz/nastenny-pokojovy-termostat-10-az-30-c-bila.k560102>>.
- [3] *CONRAD* [online]. 2013, [cit. 4. 11. 2013]. Pokojový termostat s týdenním programem TS-101, bílá. Dostupné z URL: <<http://www.conrad.cz/pokojovy-termostat-s-tydennim-programem-ts-101-bila.k610630>>.
- [4] *CONRAD* [online]. 2013, [cit. 4. 11. 2013]. Pokojový termostat pod omítku Eberle FITNP-3L, 10 - 40 °C, bílá. Dostupné z URL: <<http://www.conrad.cz/pokojovy-termostat-pod-omitku-eberle-fitnp-3l-10-40-c-bila.k560528>>.
- [5] *DHT11 Humidity & Temperature Sensor* [online datasheet]. 2010, poslední aktualizace 30. 7. 2010 [cit. 10. 12. 2013]. Dostupné z URL: <<http://www.micro4you.com/files/sensor/DHT11.pdf>>.
- [6] *FreeRTOS™* [online]. 2013, [cit. 20. 11. 2013]. Dostupné z URL: <<http://www.freertos.org/>>.
- [7] HRBÁČEK, J. *Komunikace mikrokontroléru s okolím*. Odpovědný redaktor Libor Kubica. 1. vydání. Praha: BEN - technická literatura, 1999. 159 s. ISBN 80-86056-42-2.
- [8] *Květákov.NET* [online]. 2008, poslední aktualizace 11. 1. 2008 [cit. 4. 11. 2013]. Sběrnice I2C. Dostupné z URL: <<http://www.kvetakov.net/clanky/avr/79-sbernice-i2c.html>>.
- [9] LANTRONIX *XPort™ Data Sheet* [online datasheet]. 2013, [cit. 3. 12. 2013]. Dostupné z URL: <www.lantronix.com/pdf/XPort_DS.pdf>.
- [10] *lwIP - A Lightweight TCP/IP stack* [online]. 2013, [cit. 20. 11. 2013]. Dostupné z URL: <<http://savannah.nongnu.org/projects/lwip/>>.
- [11] *lwIP Wiki* [online]. 2013, [cit. 2. 12. 2013]. Dostupné z URL: <http://lwip.wikia.com/wiki/Netconn_API>.

- [12] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR AT90S*. Odpovědná redaktorka Iveta Kubicová. 2. vydání. Praha: BEN - technická literatura, 2006. 376 s. ISBN 80-7300-209-4.
- [13] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL ATmega16*. Odpovědná redaktorka Iveta Kubicová. 1. vydání. Praha: BEN - technická literatura, 2006. 320 s. ISBN 80-7300-174-8.
- [14] maxim integrated *DS1307* [online datasheet]. 2008, poslední aktualizace 2. 10. 2008 [cit. 10. 12. 2013]. Dostupné z URL: <<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf> >.
- [15] maxim integrated *DS18B20* [online datasheet]. 2008, poslední aktualizace 22. 4. 2008 [cit. 10. 12. 2013]. Dostupné z URL: <<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> >.
- [16] SMSC *LAN8720/LAN8720i* [online datasheet]. 2013, poslední aktualizace 4. 6. 2013 [cit. 4. 11. 2013]. Dostupné z URL: <http://www.smc.com/Downloads/SMSC/Downloads_Public/Data_Briefs/8720adb.pdf >.
- [17] ST *STM32F405xx* [online datasheet]. 2013, poslední aktualizace 4. 6. 2013 [cit. 4. 11. 2013]. Dostupné z URL: <<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf> >.
- [18] Texas Instruments *TMP112* [online datasheet]. 2009, poslední aktualizace 10. 6. 2009 [cit. 10. 12. 2013]. Dostupné z URL: <<http://www.ti.com/lit/ds/symlink/tmp112.pdf> >.
- [19] tuxgraphics.org *Using the AVR microcontroller based web server* [online]. 2006 [cit. 23. 11. 2013]. Dostupné z URL: <<http://tuxgraphics.org/electronics/200612/article06121.shtml> >.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ARM Advanced RISC Machine

ARP Address Resolution Protocol

BLE Bluetooth Low Energy

CAN Controller Area Network

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

FreeRTOS Free Real-Time Operating System

FSMC Flexible Static Memory Controller

GATT Generic Attribute Profile

GSM Global System for Mobile Communications

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

IGMP Internet Group Management Protokol

I²C Inter-Integrated Circuit

IP Internet Protocol

JSON JavaScript Object Notation

LAN Local Area Network

LCD TFT Liquid Crystal Display

LED Light-Emitting Diode

LwIP stack A Lightweight TCP/IP stack

MAC Media Access Control

OTG On-The-Go

PCB Printed circuit board

RISC Reduced Instruction Set Computing

RTC Real-Time Clock

SD Secure Digital

SDIO Secure Digital Input Output

SMS Short Message Service

SMTP Simple Mail Transfer Protocol

SNMP Simple Network Management Protocol

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

TCP Transmission Control Protocol

TFT Thin-Film Transistors

TFTP Trivial File Transfer Protocol

TWI Two-Wire

UART Universal Asynchronous Receiver and Transmitter

UDP User Datagram Protocol

USART Universal Synchronous / Asynchronous Receiver and Transmitter

USB Universal Serial Bus

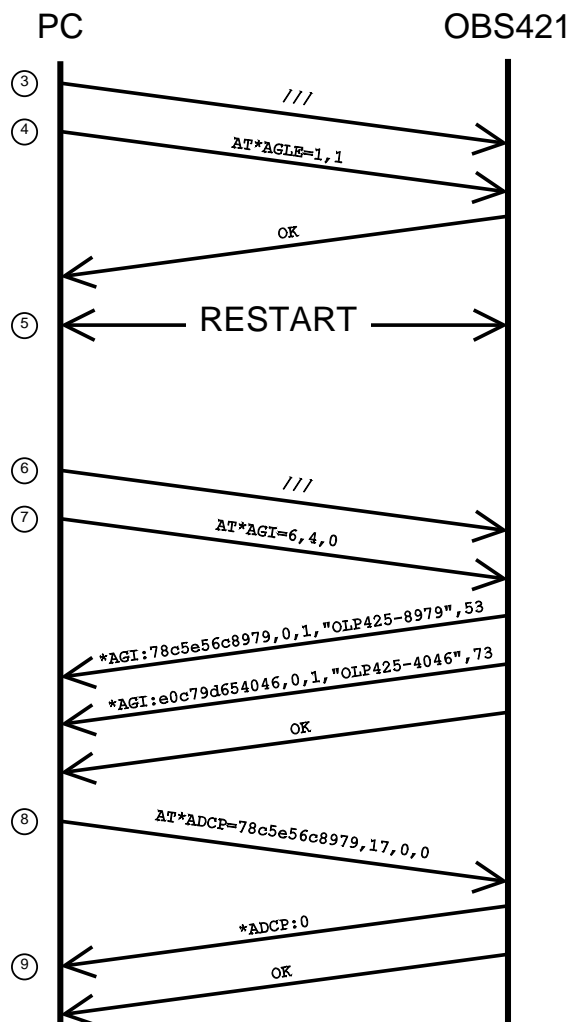
VoIP Voice over Internet Protocol

SEZNAM PŘÍLOH

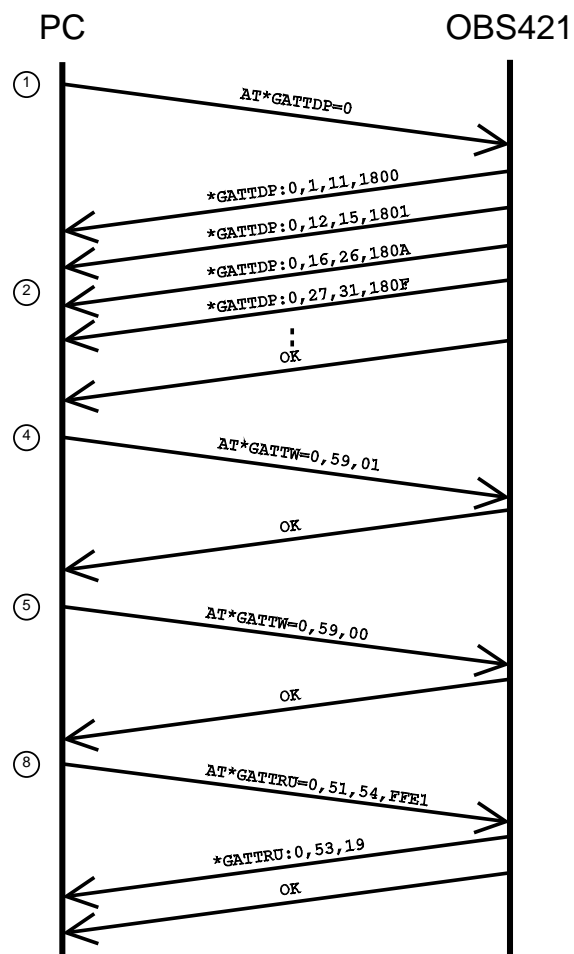
A	Komunikace bluetooth modulů	58
B	Schémata	60
C	PCB	67

A KOMUNIKACE BLUETOOTH MODULŮ

Na následujících grafech je zachycena komunikace master modulu s počítačem.

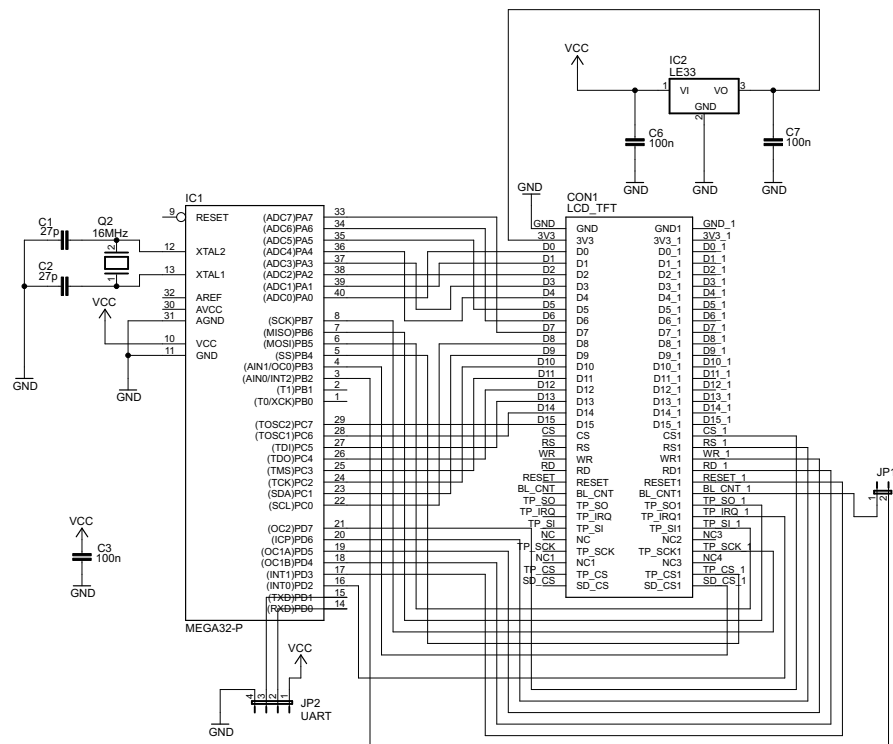


Obr. A.1: Spárování bluetooth modulů

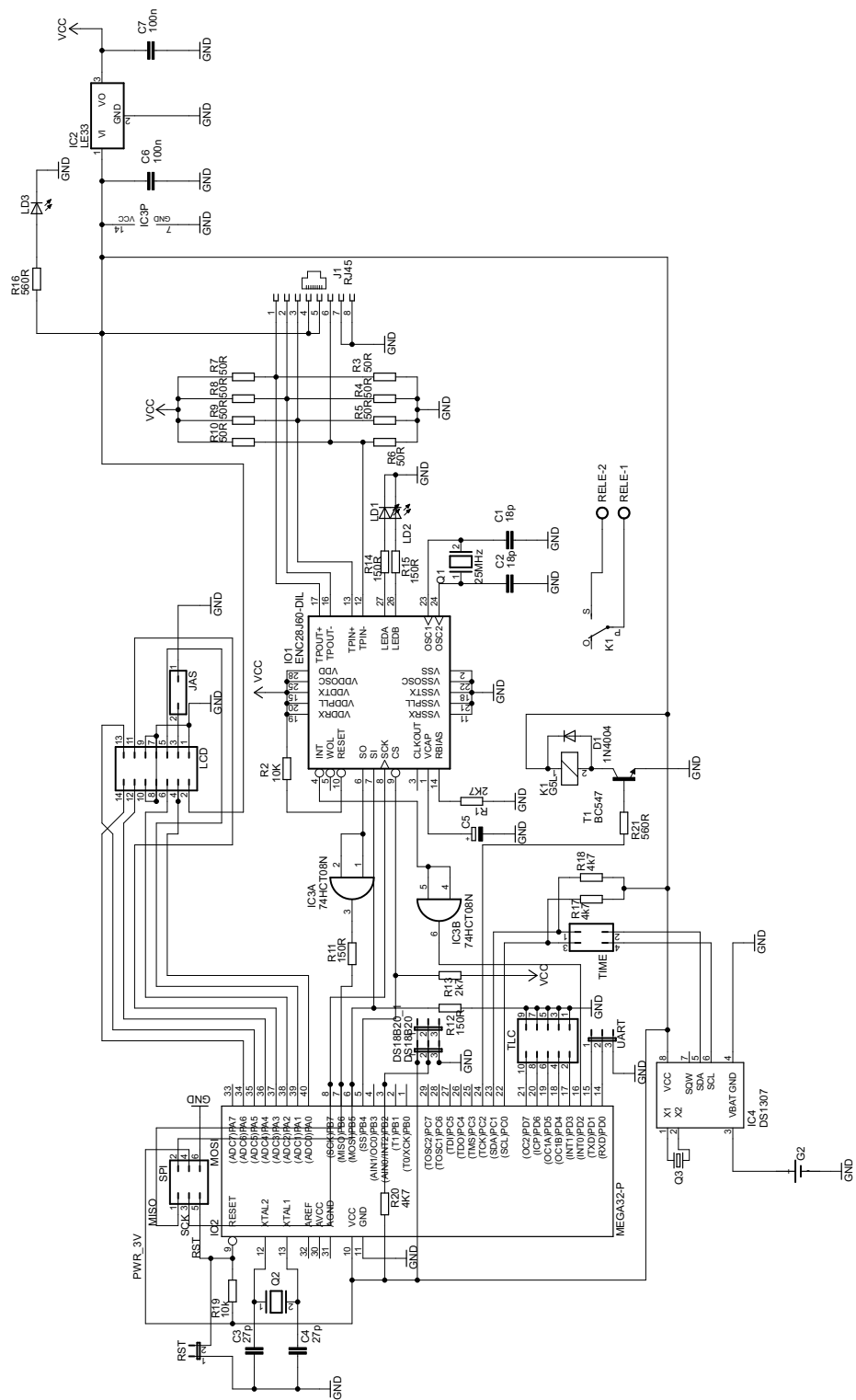


Obr. A.2: Čtení a nastavování modulu OLP425 pomocí GATT příkazů z modulu OBS421

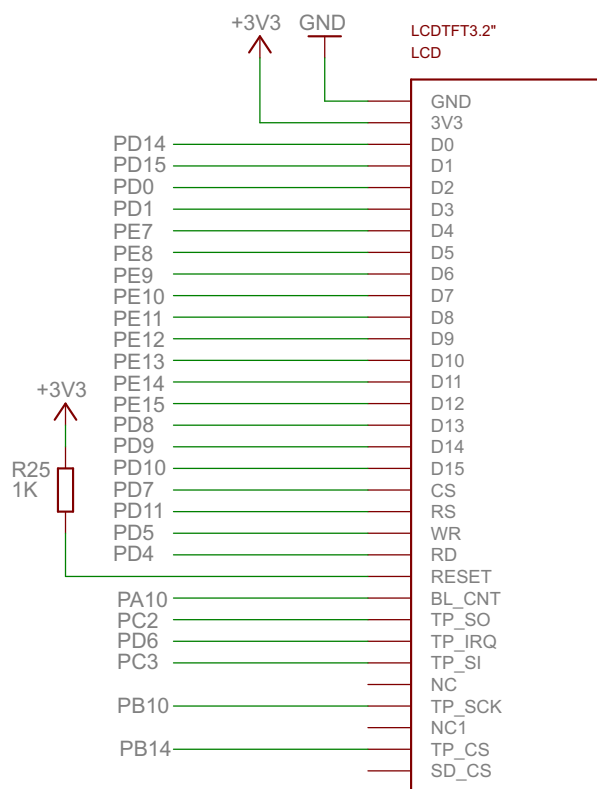
B SCHÉMATA



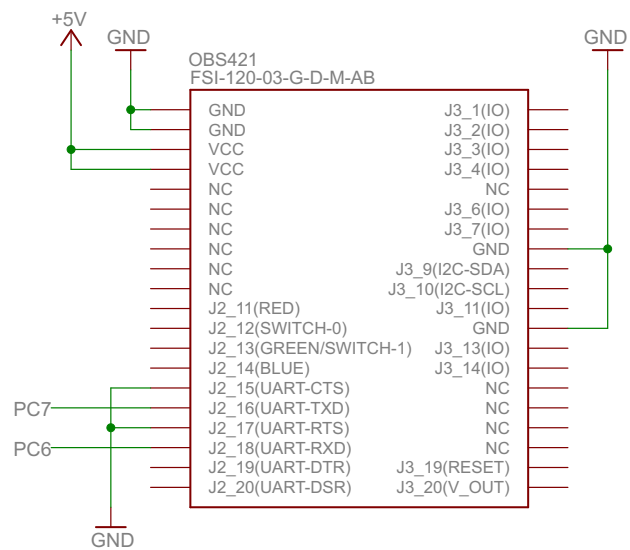
Obr. B.1: Schéma řídicí desky pro LCD TFT displej



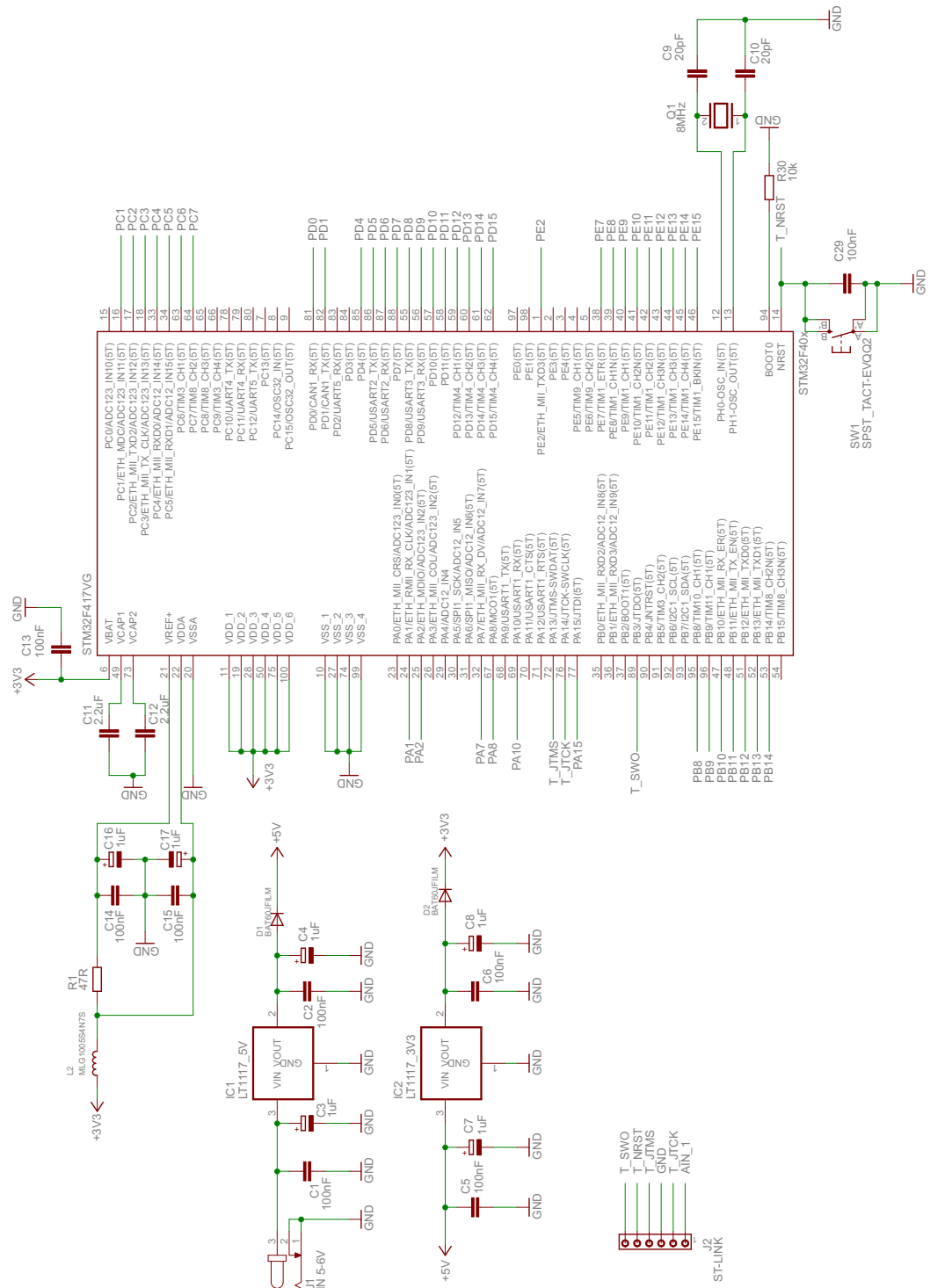
Obr. B.2: Schéma webového serveru s mikrokontrolérom Atmel



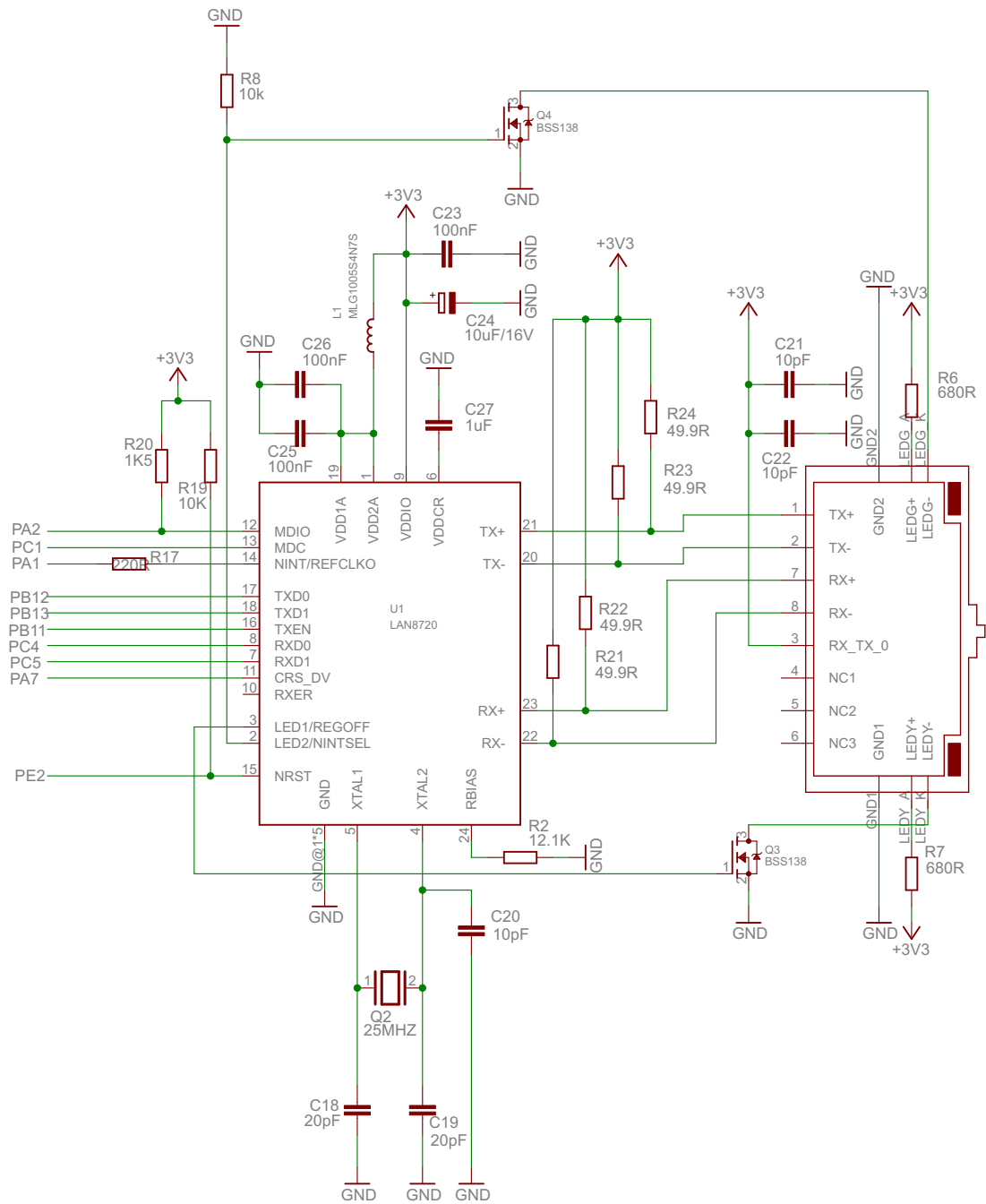
Obr. B.3: Část schématu s připojením LCD



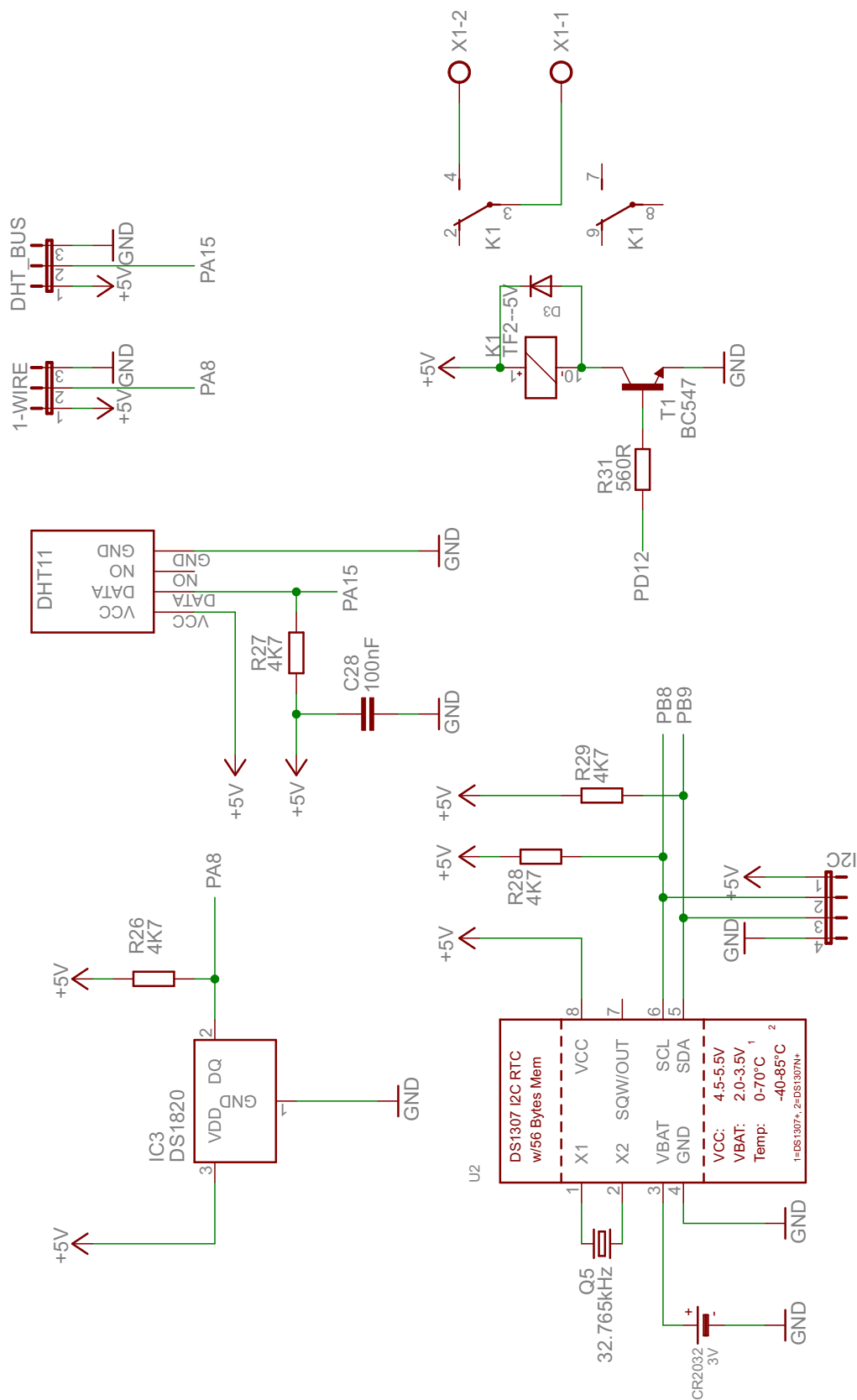
Obr. B.4: Část schématu s připojením bluetooth modulu



Obr. B.5: Část schématu s mikrokontrolérem a napájením

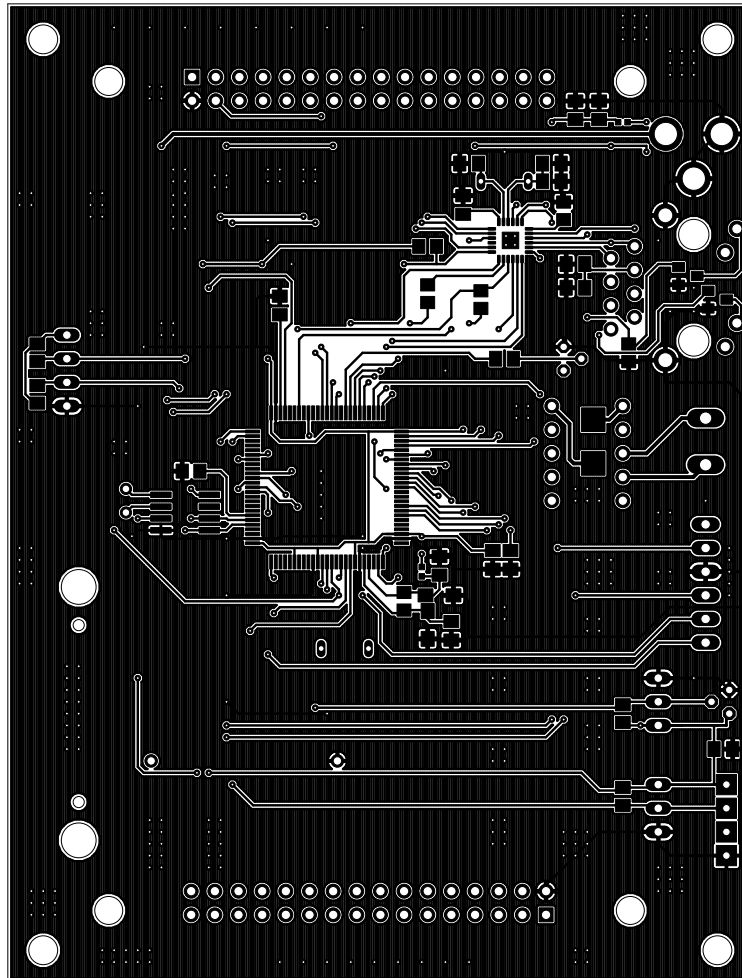


Obr. B.6: Část schématu s připojením ethernet modulu

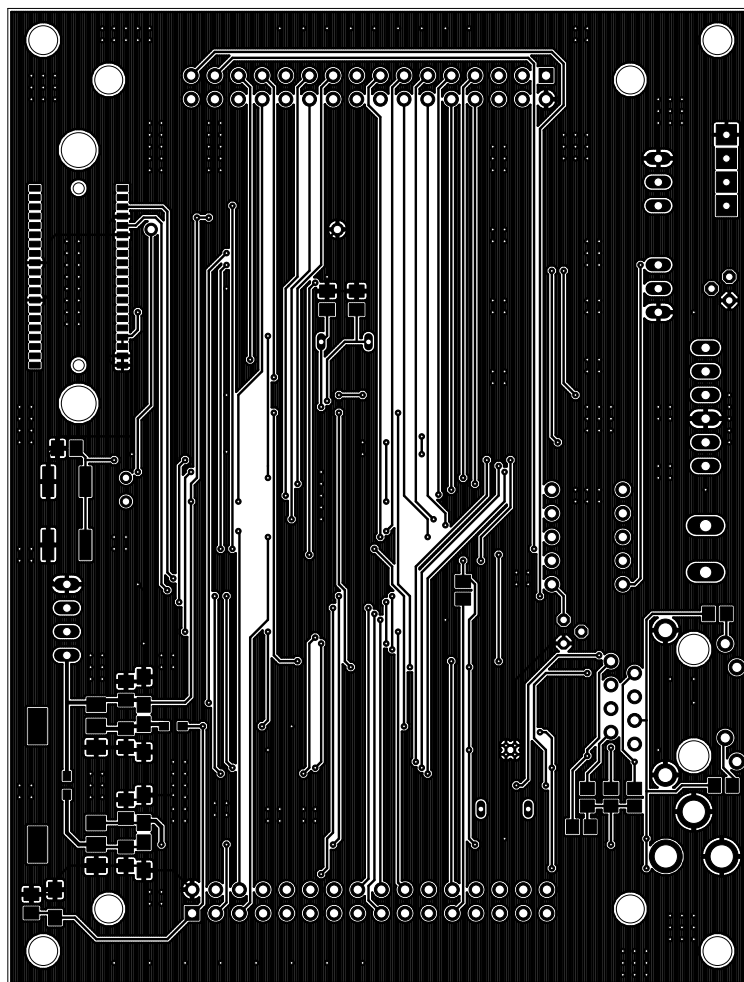


Obr. B.7: Část schématu se senzory

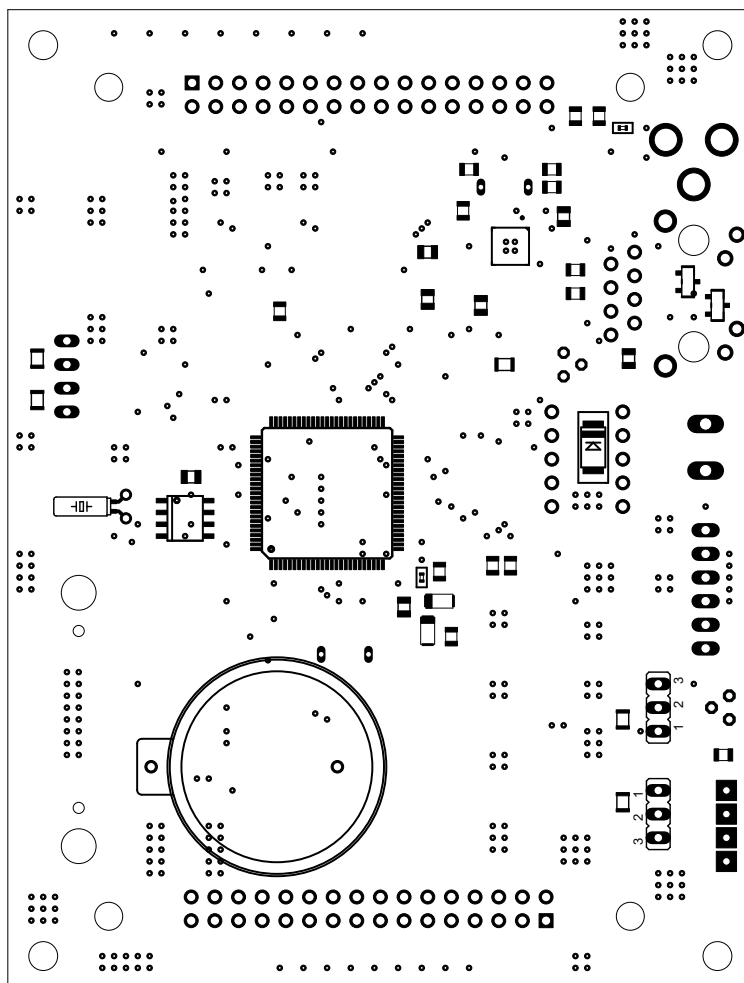
C PCB



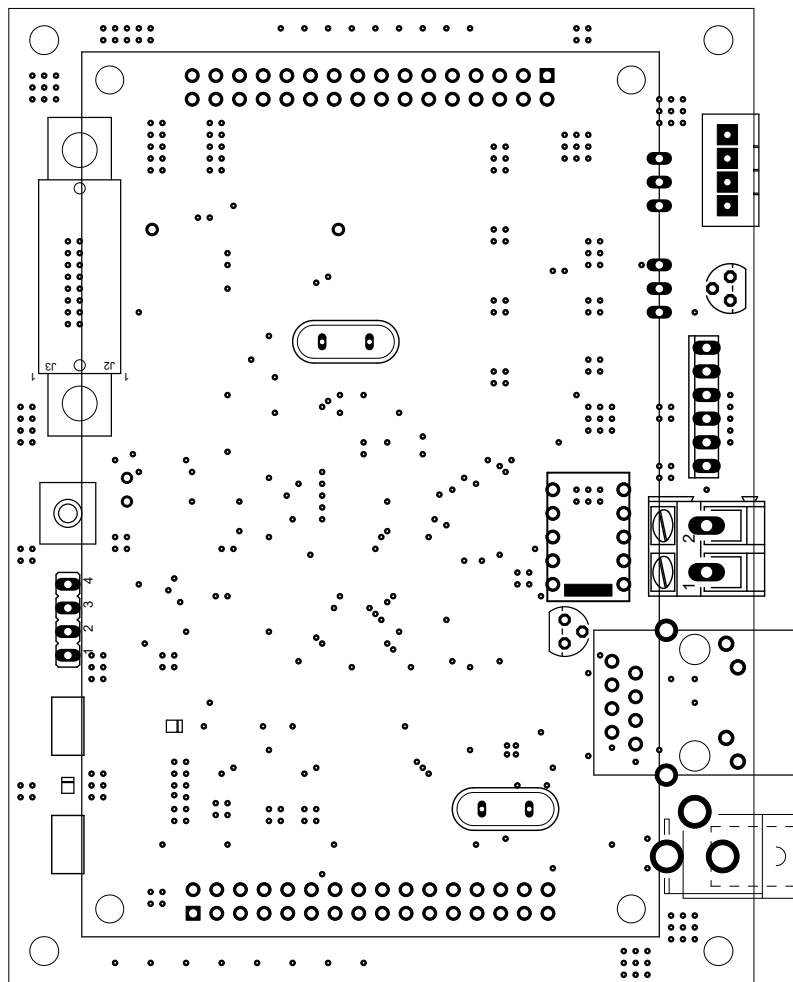
Obr. C.1: Deska plošného spoje – pohled zespodu



Obr. C.2: Deska plošného spoje – pohled zvrchu



Obr. C.3: Deska plošného spoje – pohled zespodu – rozmístění součástek



Obr. C.4: Deska plošného spoje – pohled zvrchu – rozmístění součástek