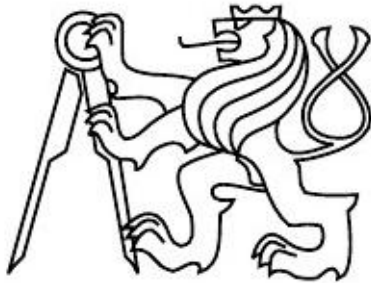


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra řídicí techniky



Diplomová práce
Firmware pro základnovou stanici domácího asistivního systému

Bc. Markéta Pecoldová

Vedoucí práce: Ing. Jan Dvořák

Studijní program: Otevřená informatika

Obor: Počítačové inženýrství

7.května 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Markéta Pecoldová**

Studijní program: Otevřená informatika
Obor: Počítačové inženýrství

Název tématu: **Firmware pro základnovou stanici domácího asistivního systému**

Pokyny pro vypracování:

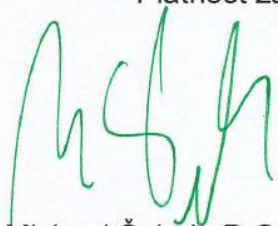
1. Seznamte se s problematikou domácí asistivní péče a s obvyklými požadavky na vzdálené monitorování osob v domácí péči a s hardwarem dodané základnové stanice domácího asistivního systému a s příslušejícími bezdrátovými snímači.
2. Navrhněte firmware pro základnovou stanici domácího asistivního systému, který umožní příjem dat z bezdrátových senzorů, zpracování těchto dat, jejich logování a odesílání na vzdálené dohledové pracoviště.
3. Implementujte navržený firmware a ověřte jeho funkci.

Seznam odborné literatury:

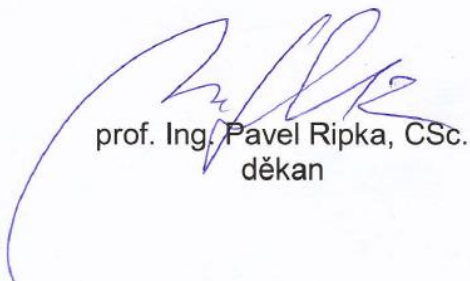
- [1] S. Bakkes, R. Morsch, B. Krose: Telemonitoring for Independently Living Elderly: Inventory of Needs & Requirements,
- [2] Datasheet STM32F207, available from www.st.com
- [3] Application notes for STM32F207, available from www.st.com

Vedoucí: Ing. Jan Dvořák

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 3. 2. 2014

Prohlášení

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

V Praze, dne 7. května 2014

..... Pecaldová

podpis

Poděkování

Ráda bych poděkovala svému vedoucímu práce Ing. Janu Dvořákovi a Ing. Janu Havlíkovi, Ph.D za všestrannou pomoc při zpracování této práce. Dále bych chtěla poděkovat své rodině, která mě po dobu celého studia podporovala.

Abstrakt

Stárnutí populace je současný fenomén ve vyspělých zemích, proto jsou důležité nové technologie a aplikace, které splňují požadavky a potřeby starších lidí na zdravotní péči. Hlavním cílem této práce je navrhnout a realizovat firmware pro základnovou stanici domácího asistivního systému. Za tímto účelem byl proveden rozbor problematiky domácí asistivní péče. V další části je popsán hardware základnové stanice a senzoru. Podrobně je popsán mikrokontrolér STM32F207VGT6. Další část práce se zabývá podrobným popisem knihoven a konfigurací periférií použitých při návrhu firmwaru. Hlavní část práce popisuje návrh a realizaci domácího asistivního systému pro základnovou stanici. V další části práce jsou popsány způsoby testování funkčnosti navrženého firmwaru. V poslední části práce je závěrečné zhodnocení a shrnutí práce.

Abstract

Aging of the population is a current phenomenon in the developed countries, therefore are important new technologies and new applications that cover the requirements and the needs of older people in the health care. The main objective of this thesis is to design and to implement firmware for base station of home assistive system. To this purpose, an analysis of the issue of telemonitoring was carried out. In the next part the hardware of base station is described. You can find there the detailed description of the microcontroller STM32F207VGT6. The next part describes libraries and configuration of peripherals used for the design of the firmware. The main part of this thesis describes the design and the implementation of the firmware for base station of home assistive system. In the next part are described the ways of the designed firmware testing. In the last part are the final summary and evaluation of work.

Obsah

1 Úvod	1
1.1 Cíl práce.....	2
2 Domácí asistivní péče a požadavky na vzdálené monitorování	3
2.1 Monitorování fyziologických funkcí	3
2.2 Monitorování rizikových situací.....	6
2.3 Monitorování každodenních aktivit	8
2.4 Potřeby a požadavky monitorovaných osob.....	9
2.5 Potřeby a požadavky ošetřujících osob.....	11
3 Hardware základnové stanice a pomocných obvodů	13
3.1 Pomocný hardware základnové stanice	13
3.2 Popis hardwaru základnové stanice	14
3.3 Mikrokontrolér STM32F207VGT6	16
3.3.1 Architektura ARM.....	16
3.3.2 ARM Cortex-M3.....	18
3.3.3 Popis mikrokontroléru STM32F207VGT6.....	21
3.4 Napájení základnové stanice	23
3.5 UART	23
3.6 Ethernet	24
3.7 RF modul.....	24
3.8 One-wire bus	25
3.9 Slot SD karty	25
3.10 Bluetooth	26
3.11 Bezdrátový senzor - akcelerometr	27
4 Software	28
4.1 Vývojové prostředí.....	28
4.2 Operační systém FreeRTOS	28
4.2.1 Úlohy (Tasks).....	29
4.2.2 Fronty, semaforey a mutexy	30
4.2.3 Řízení úloh	31
4.3 LwIP.....	31

4.3.1	Podporované protokoly.....	31
4.3.2	Způsoby komunikace.....	32
4.3.3	Správa paměti.....	33
4.3.4	Použití LwIP	33
4.3.4.1	Ethernetové rozhraní.....	33
4.3.4.2	DHCP.....	33
4.3.4.3	TCP rozhraní	34
4.3.4.4	SNTP.....	34
4.4	USART/RS-232.....	35
4.5	Micro SD karta a FatFs	36
4.5	One-wire bus	38
4.6	RF modul.....	42
4.7	EEPROM.....	42
4.8	RTC.....	42
5	Návrh firmwaru	44
5.1	Inicializace základnové stanice	44
5.2	Nastavení statické IP adresy, masky sítě a IP adresy brány.....	45
5.3	Konfigurace parametrů zařízení z DHCP serveru	47
5.4	Nastavení hodin reálného času pomocí SNTP protokolu	47
5.5	Sledování senzorů.....	48
5.5.1	Používané senzory.....	49
5.5.2	Posílání dat na dohledový server Swiftshield	50
5.5.3	Zapsání dat do logu	52
5.6	Posílání live packetu každou hodinu.....	53
5.6.1	Kontrola odeslání dat na dohledový server a zápis dat do logu.....	55
5.6.2	Příklad kontroly a zápisu do logu	56
6	Testování	57
6.1	Testování statických parametrů.....	57
6.1.1	Testování uložení statických parametrů do souboru	57
6.1.2	Testování načtení statických parametrů z neexistujícího souboru.....	58
6.1.3	Testování načtení statických parametrů z existujícího souboru.....	59
6.2	Testování přijímání dat ze senzorů a odesílání dat na dohledový server Swiftshield.....	60

6.3 Testování kontroly logu	64
7 Závěr	69
Literatura.....	71
A Seznam použitých zkratek	79
B Layouty základnové stanice	83
C Obsah přiloženého CD	85

Seznam obrázků

Obr. 3.1 Převodník Asix UDS1 s modulem UMS1B	13
Obr. 3.2 Discovery kit STM32L1	14
Obr. 3.3 Základnová stanice	14
Obr. 3.4 Blokové schéma základnové stanice	15
Obr. 3.5 Blokové schéma procesoru ARM Cortex-M3 [40]	19
Obr. 3.6 Paměťový prostor procesoru ARM Cortex-M3 [40]	20
Obr. 3.7 Obvodové schéma mikrokontroléru	21
Obr. 3.8 Blokové schéma STM32F207 [41]	22
Obr. 3.9 Obvodové schéma napájení	23
Obr. 3.10 Obvodové schéma UARTu	23
Obr. 3.11 Obvodové schéma komunikace Ethernet	24
Obr. 3.12 Koncepce Ethernetu	24
Obr. 3.13 Obvodové schéma RF modulu	25
Obr. 3.14 Obvodové schéma one-wire bus	25
Obr. 3.15 Obvodové schéma slotu micro SD karty	26
Obr. 3.16 Obvodové schéma Bluetooth	26
Obr. 3.17 Obvodové schéma akcelerometru	27
Obr. 3.18 Blokové schéma akcelerometru	27
Obr. 4.1 Stavový diagram úlohy [42]	29
Obr. 4.2 Výpis po DHCP	34
Obr. 4.3 Signály Rx a Tx UARTu pro písmeno “K“	35
Obr. 4.4 Signály Rx a Tx RS-232 pro písmeno “K“	36
Obr. 4.5 FAT File Systém Modul [50]	37
Obr. 4.6 Inicializační sekvence [51]	39
Obr. 4.7 Time sloty pro čtení dat [51]	39
Obr. 4.8 Time sloty pro zápis dat [51]	40
Obr. 4.9 Reset a presence puls, příkaz Read ROM a čtení Family Number	41
Obr. 4.10 Čtení sériového čísla Serial Number	41
Obr. 4.11 Čtení kontrolního součtu CRC	41
Obr. 5.1 Výpis pokud není vložena SD karta	45
Obr. 5.2 Nastavení statické IP adresy, masky sítě a IP adresy brány	46

Obr. 5.3 Výpis při konfiguraci z DHCP serveru	47
Obr. 5.4 Výpis při přiřazení statických parametrů	47
Obr. 5.5 Výpis s aktuálním časem ze serveru	48
Obr. 5.6 Sledování senzorů a zápis do logu	49
Obr. 5.7 Výpis paketů ze senzoru.....	50
Obr. 5.8 Rozhraní dohledového serveru Swiftshield.....	51
Obr. 5.9 Zobrazení systoly a diastoly na serveru Swiftshield	51
Obr. 5.10 Struktura dat zapisovaných do logu	52
Obr. 5.11 Obecná struktura dat zapsaných do logu	53
Obr. 5.12 Kontrola a posílání „data live“ na dohledový server	54
Obr. 5.13 Struktura dat při neúspěšném odeslání dat na dohledový server a záznam o kontrole s odesláním „data live“ na dohledový server	56
Obr. 6.1 Výpis při uložení a přiřazení statických parametrů	58
Obr. 6.2 Výpis při načítání statických parametrů z neexistujícího souboru	59
Obr. 6.3 Výpis při načítání statických parametrů z existujícího souboru	60
Obr. 6.4 Výpis inicializace a nastavení RTC při simulaci druhého scénáře testování	61
Obr. 6.5 Výpis logu po simulaci druhého scénáře testování	62
Obr. 6.6 Záznam událostí na Swiftshiledu po simulaci druhého scénáře testování	63
Obr. 6.7 Záznam hodnot krevního tlaku po simulaci druhého scénáře testování	64
Obr. 6.8 Výpis inicializace a nastavení RTC při simulaci třetího scénáře testování.....	65
Obr. 6.9 Výpis logu po simulaci třetího scénáře testování	66
Obr. 6.10 Záznam událostí na Swiftshiledu po simulaci třetího scénáře testování.....	67
Obr. 6.11 Záznam hodnot krevního tlaku po simulaci třetího scénáře testování	68
Obr. B.1 Layout horní vrstvy základnové stanic	83
Obr. B.2 Layout horní vrstvy základnové stanice	84

Seznam tabulek

Tabulka 1.1 Prognóza průměrného věku obyvatel ČR [2].....	1
Tabulka 1.2 Budoucí vývoj složení obyvatel starších 65 let v ČR [3]	1

1 Úvod

Stárnutí populace je současný fenomén ve vyspělých zemích, který se projevuje úbytkem mladší populace a jejím nahrazením populací starší. Osob ve věku starších 65 let přibývalo zatím pozvolně. Podle Českého statistického úřadu, však bude intenzivněji přibývat v České republice starších obyvatel, počet dětí bude naopak ubývat. Podobné výsledky uvádí i zpráva Evropské unie o stárnutí populace.

Gerontologové obvykle uvádějí jako počátek stáří ukončení pracovního poměru a odchod do důchodu. Věk odchodu do důchodu se většinou pohybuje v rozmezí 60 až 65 let. V tomto věku většinou dochází ke zřetelným fyzickým a psychologickým změnám [1]. Pojmem senior tedy bude dále myšlena osoba starší 65 let.

Průměrný věk obyvatel České republiky vzrostl za posledních 10 let téměř o dva roky. Do roku 2065 se očekává, že průměrný věk obyvatelů České republiky vzroste o dalších 8 let (tab. 1.1). Tento fakt je způsoben nižší porodností a rostoucí střední délkou života.

	2000	2010	2015	2020	2025	2045	2055	2065
Průměrný věk obyvatel ČR	38,8	40,6	41,6	42,7	43,9	47,5	48,3	49

Tabulka 1.1 Prognóza průměrného věku obyvatel ČR [2]

Do roku 2050 bude počet seniorů v České republice dvojnásobný, přičemž se zpětinásobí počet obyvatel starších 85 let (tab. 1.2). Dle výzkumu Českého statistického úřadu by v roce 2055 měl být průměrný věk dožití v České republice u mužů 84,5 let a u žen 89,3 let.

	65-74	75-84	85+
2002	818 702	501 081	98 179
2010	924 610	527 687	144 515
2015	1 165 776	524 561	173 809
2020	1 273 549	627 483	187 301
2025	1 184 048	809 329	207 933
2045	1 520 352	884 345	457 323
2050	1 415 382	1 043 570	497 127

Tabulka 1.2 Budoucí vývoj složení obyvatel starších 65 let v ČR [3]

Podle zprávy Evropské unie z roku 2012 se do roku 2060 v EU rapidně nezvýší počet obyvatel. Populace v EU však bude mnohem starší. Téměř 30% obyvatel bude starších 65 let. Střední délka života by se měla v Evropské unii prodloužit u mužů na 84,6 let a u žen na 89,1 let [4].

Vyšší věk obyvatelstva koresponduje se zvýšeným počtem nemocných. Jedná se o nemoci jako diabetes, srdeční nemoci či demence, proto je nutné neustálé monitorování starších osob. Důležité je však i monitorovat činnosti běžného dne kvůli bezpečnosti seniora. Je tedy důležité vytvořit samostatně žijícím seniorům takové podmínky, aby se ve svých domovech cítili bezpečně a nezávisle.

1.1 Cíl práce

Jak již bylo zmíněno, obyvatelstvo stárne a tím se i zvyšují požadavky na zdravotní péči, proto je nutné vytvořit technologie a aplikace, které by tyto požadavky splňovaly. Tato práce se proto klade za cíl navrhnout firmware, který by se stal základem pro plnohodnotný domácí asistivní systém.

Cílem této práce je seznámit se s problematikou domácí asistivní péče a s obvyklými požadavky na vzdálené monitorování osob v domácí péči. Dále si tato práce klade za cíl seznámit se s hardwarem dodané základnové stanice domácího asistivního systému a s příslušejícími bezdrátovými snímači. Přínosem práce je návrh firmwaru pro základnovou stanici domácího asistivního systému, který umožní příjem dat z bezdrátových senzorů, zpracování těchto dat, jejich logování a odesílání na vzdálené dohledové pracoviště. Navržený firmware bude implementován a bude ověřena jeho funkce.

Diplomová práce bude rozdělena do sedmi kapitol. Druhá kapitola se bude zabývat výzkumy, které se zaměřují na problematiku domácí asistivní péče a na obvyklé požadavky na vzdálené monitorování osob v domácí péči. Třetí kapitola se bude zabývat hardwarem dodané základnové stanice a hardwarem bezdrátových snímačů. Čtvrtá kapitola bude popisovat knihovny využívané při návrhu softwaru a bude zde i popsána konfigurace a použití jednotlivých periférií. V páté kapitole bude popsán návrh firmwaru. Šestá kapitola se bude zabývat testováním navrženého firmwaru. Poslední kapitola shrne všechny zjištěné poznatky, návrh a i funkčnost firmwaru.

2 Domácí asistivní péče a požadavky na vzdálené monitorování

Asistivní technologie zajišťují bezpečný život jejím uživatelům. Především se jedná o uživatele, kteří žijí samostatně ve svých domovech. Mezi asistivní technologie patří jakýkoliv nástroj, zařízení, software nebo systém, který využívá moderní technologie, především se jedná o senzory, informační a komunikační technologie. Cílem je osobám, které využívají asistivní technologie usnadnit každodenní život, zlepšit kvalitu života, podpořit samostatnost a soběstačnost.

Součástí asistivní technologie je i telemedicína. Telemedicína zahrnuje zdravotnické aktivity, služby a systémy, provozované na dálku cestou informačních a komunikačních technologií za účelem podpory zdraví, prevence, zdravotní péče, řízení zdravotnictví a zdravotnického výzkumu [5]. Telemedicína je také jedna z možností, jak reagovat na nové potřeby související se stárnutím populace [6], protože současná zdravotní péče neodpovídá potřebám starší populace [7].

Rychlý technologický pokrok vede k vývoji širokého spektra telemedicínských systémů, které umožňují prevenci, včasnou diagnostiku zdravotního stavu a monitoring chronicky nemocných pacientů, kteří žijí samostatně ve svých domovech. Dálkový monitoring může také snížit počet hospitalizací v nemocnici, například u seniorů často dochází k neplánované a nákladné hospitalizaci v nemocnicích či jiných zdravotnických zařízeních. Monitoring pomocí telemedicínských systémů může také odhalit postupné zhoršování zdravotního stavu, což může znamenat snížení schopnosti seniora žít samostatně. Důležité je proto zvolit vhodný přístup monitorování. Jedná se o monitorování fyziologických funkcí, monitorování rizikových situací a monitorování každodenních aktivit [9].

2.1 Monitorování fyziologických funkcí

Mezi základní fyziologické funkce patří dýchání, vědomí, udržování stálé tělesné teploty, udržování stálého krevního tlaku, srdeční akce a puls [8].

Vědomí je stav, kdy je osoba zcela schopna vnímat podněty a je schopna na ně reagovat. Stav vědomí se především vyhodnocuje podle orientace dané osoby v prostoru, místě a času.

Tělesná teplota je ovlivněna bazálním metabolismem, zvýšenou svalovou aktivitou, zvýšenou teplotou tělových buněk, hormony štítné žlázy a nadledvinek, psychickými procesy, věkem, denní dobou a tělesnou aktivitou. Hodnocení tělesné teploty: zvýšená teplota (37 – 38 °C), horečka (38 – 40°C), hyperpyrexie (teplota nad 40°C) a hypotermie (teplota pod 35,5 °C).

Krevní tlak je síla, kterou vyvíjí krev na stěnu tepen. U krevního tlaku je měřena jeho maximální (systola) a minimální (diastola) hodnota v artériích. Krevní tlak je závislý na objemu krve v krevním řečišti, pružnosti cévní stěny, průsvitem kapilár a viskozitou krve. Dále je krevní tlak ovlivněn věkem, tělesnou námahou, emocemi, pohlavím, denní dobou, tělesnou hmotností, léky, nemocemi srdce a cév, úrazy, nemocemi nervového systému, endokrinními chorobami a prostředím. Hodnocení krevního tlaku: normotenze (120/80 mmHg), mírná hypertenze (140/90 mmHg), střední hypertenze (160/100 mmHg), těžká hypertenze (180/110 mmHg) a hypotenze (85/60 mmHg).

Pulzová vlna je náraz krevního proudu na stěnu tepny při systole. Je rozlišován puls periferní a centrální. Puls je ovlivněn věkem, pohlavím, fyzickou námahou, zvýšenou tělesnou teplotou, krvácením, stresem, strachem, úzkostí a léky. Hodnocen je dle frekvence, plnosti a pravidelnosti.

Dýchání existuje zevní a vnitřní, hrudní + břišní, vdech + výdech. Ovlivňuje ho věk, tělesná aktivita, stres, strach, obavy, nadmořská výška, léky a životní styl. Dýchání je hodnoceno podle frekvence, hloubky dýchání, charakteru dýchání a pravidelnosti.

Při monitorování fyziologických funkcí je důležité se především zaměřit na měření kardiovaskulární aktivity. Kardiovaskulární onemocnění patří mezi civilizační choroby a jsou nejčastější příčinou úmrtí [10]. Většina kardiovaskulárních onemocnění se projevuje po padesátém roce života. Především se jedná o onemocnění jako ischemická choroba srdeční (angina pectoris a infarkt myokardu), vrozené srdeční vady, získané srdeční vady, kardiomyopatie, hypertenze, cévní mozkové příhody, ischemická choroba cév dolních končetin, záněty žil a chronická cévní nedostatečnost.

Bylo provedeno mnoho výzkumů, které monitorovaly srdeční frekvenci [11], krevní tlak [11], [12], EKG [11], [13] či kardiostimulátor [13]. Konkrétně výzkum [11] se zaměřil na trvalé monitorování kardiovaskulární aktivity pomocí telemedicínského bezdrátového zařízení. Monitorování probíhalo v domově pacientů. Do studie bylo

zahrnuto 20 pacientů. Bezdrátové zařízení zaznamenávalo srdeční frekvenci a dechovou frekvenci, krevní tlak, EKG a tělesnou teplotu. Výsledky byly automaticky odesílány na vzdálené dohledové pracoviště. Správnost tohoto monitorování bylo kontrolováno školenou zdravotní sestrou, která naměřené hodnoty ověřovala. Systém byl pro pacienty přijatelný a osvědčil se i jako použitelný v domově pacientů.

Přínosem pro telemedicínu byl i výzkum [13] o prototypu systému, který přenáší EKG pacienta a parametry kardiostimulátoru do vzdáleného monitorovacího pracoviště. Tento monitorovací systém byl především přínosem pro pacienty, hlavně co se týče zvýšení kvality jejich života. Jak studie uvádí, v současné době se neustále zvyšuje počet pacientů s kardiostimulátorem. Každý pacient musí dvakrát za rok na prohlídku do nemocnice. Tato skutečnost může především být nepříjemná starším pacientům. Navržený prototyp by měl tyto nepříjemnosti odstranit.

Do monitorování fyziologických funkcí bezpochyby patří i monitorování metabolismu. Konkrétně se jedná o monitorování bazálního metabolismu [15], hladiny glukózy v krvi [14], tělesné teploty [16] či krevního laktátu [16]. Výzkum [14] zaměřený na měření hladiny glukózy v krvi představuje telemedicínský systém DIABTel. Tento systém je určen pro monitorování diabetických pacientů a měl by zlepšit kvalitu péče o tyto pacienty. Tento telemedicínský systém se skládá z přístroje pro pacienty a z lékařské pracovní stanice, kterou využívají lékaři a sestry. Jak přístroj pro pacienty, tak lékařská stanice umožňuje shromažďování, správu a zobrazování dat a i výměnu dat a zpráv. Výzkum probíhal 6 měsíců a zúčastnilo se ho 10 pacientů s diabetem typu I. Systém zlepšuje dostupnost informací, které jsou důležité pro stanovení léčby. Systém dále přináší i zlepšení kontroly fyziologických funkcí pacienta.

Dále byl vyvinut telemedicínský systém pro vyšetření spirometrie [17]. Spirometrie je vyšetření, kdy lékař zjišťuje funkci pacientových plic. Vyšetření je prováděno pomocí spirometru, kdy je testována schopnost pacientových plic při nádechu a výdechu. Cílem tohoto výzkumu bylo monitorování spirometrie u pacientů s astmatem na dálku. Naměřená data byla posílána na server, kde si tato data mohli prohlédnout lékaři i sestry. Každý pacient s astmatem nejdříve prošel 40 minutovým školením o ovládání systému. Pacienti poté dvakrát denně po dobu 3 týdnů měřili pomocí systému svou plicní ventilaci ve svých domovech. Každého pacienta jednou navštívil 10 až 40 minut po měření spirometrie školený zdravotník, který požádal pacienta o provedení nového měření spirometrie pod jeho dohledem. Tímto opětovným měřením byly ověřeny naměřené hodnoty a byl vyhodnocen postoj pacienta k systému pomocí standardizovaného

dotazníku. Bylo zjištěno, že naměřená data po kontrole zdravotníkem se shodují s daty naměřené pacientem. I když většina zúčastněných pacientů (71 %) nevlastnilo žádný počítač, neměli s testováním téměř žádné problémy. Většina pacientů (87,1 %) mělo zájem o pokračování tohoto monitorování pomocí telemedicínského systému i v budoucnu. Důležitým zjištěním je, že výsledky, které naměřili pacienti ve svých domovech, jsou srovnatelné s výsledky, které byly naměřeny školeným zdravotnickým personálem.

Vyvinuty byly i neurologické systémy pro měření EEG [18] či EMG [19], ale i urologické systémy, které měří intravezikální tlak [20].

2.2 Monitorování rizikových situací

Data získaná z telemedicínského systému, lze použít i pro odhalení rizikové situace. Mezi ně patří situace, které vyžadují bezprostřední péči, dále neočekávané situace. Dále je nutné identifikovat situace, kdy není potřeba lékařský zásah. Příklady těchto rizikových situací, které přinášejí hlavní zdravotní riziko, jsou pády a vycházky.

U starších lidí se zvyšuje počet rizikových situací, jako jsou pády. Senioři mají problémy s rovnováhou i chůzí, proto jsou k pádům více náchylní [21]. Téměř polovina všech pádů jsou nehody, způsobené vycházkami či uklouznutím. Příčinou pádu, ale může být i postižení chůze, rovnováhy a zraku či projev nemoci, například infekce močových cest či hrudníku. Tento stav může být způsoben neurologickým onemocněním, poruchou pohybového systému, postižením způsobeným chronickým onemocněním (mrtvice, Parkinsonova nemoc) či působením léků a alkoholu. Poslední možná příčina pádů jsou záchvaty, kdy se jedná především o cévní příhody (mrtvice, infarkt myokardu, plicní embolie) a o zavrať.

Výzkum [22] zjišťoval, zda telemedicínské systémy pro vzdálené monitorování mohou být použity pro upozornění lékařského personálu v případě pádu pacienta a tím mohou zvýšit pocit bezpečí u starších lidí. Výsledky uvádějí, že použití telemedicínského systému pro detekci pádů zvýšilo u seniorů pocit bezpečí a hlavně jim umožnil zůstat ve svých domovech. Nicméně, někteří ze seniorů uvádějí, že považují systém za zásah do soukromí a měli pocit, že systém neposílá informace do dohledového pracoviště.

Pro detektování pádů používají telemedicínské systémy dva druhy senzorů. První možnost je detekce pádů pomocí senzorů umístěných na těle. Například výzkum [23]

používal akcelerometry, které měli účastníci připevněny k trupu a ke stehnu. Cílem tohoto výzkumu bylo získat data, která by v budoucnu poskytla podklad pro identifikaci pádů. V první části výzkumu pády simulovalo 10 mladých lidí ve věku 21 až 29 let. Každý účastník simuloval osm nejčastějších pádů zaznamenaných u starších osob. Každý tento pád byl zopakován třikrát. Každý účastník tedy nasimuloval 24 pádů. Jednalo se o simulaci pádů, kdy člověk padá dopředu, dozadu, doleva, doprava s nepokrčenými nohama a s pokrčenými koleny. Druhé části výzkumu se zúčastnili osoby ve věku 70 až 83 let. Akcelerometry měli opět umístěny na trupu a na stehně. Cílem této části výzkumu bylo analyzovat pohyby starších osob. Každý účastník musel třikrát zopakovat následující úkoly:

- Sedět na křesle a poté z něho vstát.
- Sedět na kuchyňské židli a poté z ní vstát.
- Sedět na toaletě a poté z ní vstát.
- Sedět na nízké stoličce a poté z ní vstát.
- Sednout si do auta a vystoupit z auta.
- Sedět a poté se zvednou z postele.
- Ležet na posteli a poté z ní vstát.
- 10 m chůze.

Další možností je použít senzory, které nejsou umístěny na těle. Tyto senzory jsou umístěny v předmětech denní potřeby, jako například na toaletě, v židli či posteli. Tyto senzory, ale mohou být umístěny i po bytě sledované osoby. Například se jedná o kamerové systémy. Obvykle se jedná o širokoúhlé kamery [24]. Při umístění kamer se musí dbát na soukromí sledovaných osob [25], protože v místnostech jako toaleta či koupelna může být monitorování nepříjemné. Monitorování těchto místností, je ale velmi důležité. V koupelně mohou být i překážky, které brání monitorování (závěs u sprchového koutu). Nutný je dozor i na toaletě, kde existuje zvýšené riziko uklouznutí či pádu. Proto je vhodné použít v těchto prostorech jiné senzory, jako infračervené senzory [26], senzory pohybu či akustické senzory [27]. Dále je vhodné v těchto místnostech umístit tlačítko pro přivolání pomoci v případě potřeby. Například v koupelně se toto tlačítko umísťuje i na zem pro případ pádu.

Mezi rizikové situace patří i zmatené a popletené chování. Stejně jako pro detekci pádů mohou být použity senzory umístěné na těle. Jedná se například o senzor [28], který je umístěn na těle ve formě náramkových hodinek. Komunikace Bluetooth zajišťuje

posílání dat do mobilního telefonu či do počítače. Tento senzor snímá světlo, pohyb, zvuk a teplotu. Použity mohou být opět obdobné senzory, které nejsou umístěné na těle monitorované osoby [29].

2.3 Monitorování každodenních aktivit

Monitorování každodenních aktivit je velmi obtížné. Důležité je zaměřit se na konkrétní aktivity. V roce 1964 definoval Katz [30] základní každodenní aktivity a Katzův index nezávislosti. Katzův index nezávislosti je používán k vyhodnocení aktuálního stavu sledovaného. Mezi šest základních každodenních aktivit patří:

- koupání;
- oblékání;
- používání toalety;
- přemísťování;
- kontinence;
- stravování.

U těchto aktivit je důležité posoudit závislost či nezávislost monitorované osoby [31]. Pokud při dané aktivitě osoba nepotřebuje žádnou asistenci, je považována za nezávislou, v opačném případě se jedná o závislou osobu. Při koupání je osoba považována za nezávislou pokud se myje sama či potřebuje pomoc jen u mytí jedné části těla (např. záda či ochrnutá končetina). Osoba je při koupání závislá pokud potřebuje pomoc s mytím více částí těla ať ve vaně, ve sprše či na lůžku. Pokud je osoba schopná vyndat oblečení ze skříni a zásuvek, obleče se kompletně včetně kabátu a zapínání, je považována za nezávislou, co se týká oblékání. Dojde-li si osoba na toaletu a dokáže se posadit i vstát a upravit oblečení je osoba nezávislá, co se týče aktivity používání toalety. Pokud ovšem osoba potřebuje pomoci s přemístěním na toaletu či s očistou je osoba závislá. Při přemísťování je osoba nezávislá pokud nepotřebuje asistenci při přesunu z lůžka a do něj či do křesla, osoba může používat mechanické pomůcky. Závislost při přemísťování je, když osoba potřebuje pomoci k přesunu z lůžka do křesla či potřebuje pomoc při kompletním přemísťování. V případě, že osoba má kompletně pod kontrolou vyměšování, je tato osoba nezávislá při kontinenci. Pokud, ale u osoby existuje úplná či částečná inkontinence moči či stolice, jedná se o osobu závislou na asistenci při aktivitě kontinence. Za předpokladu, že osoba je schopná vkládat si stravu z talíře do úst bez asistence, je osoba považována za nezávislou, co se týče stravování a to i v případě, že

přípravu jídla zajišťuje jiná osoba. V případě, že osoba potřebuje při stravování částečnou či kompletní pomoc, jedná se o osobu závislou.

V minulých letech bylo navrženo mnoho telemedicínských systémů pro monitorování každodenních aktivit, např. systémy založené na akcelerometrech, RFID čtečky, kamerové systémy, lokalizační technologie, pohybové infračervené senzory, atd. Z výzkumu [32] vyplývá, že nejlepší výsledky, pro monitorování každodenních aktivit, mají infračervené senzory pohybu. Tyto senzory jsou schopny měřit téměř všechny aktivity, kromě stravování, čtení a práce s mobilním telefonem, protože u těchto aktivit provádí sledovaná osoba minimální pohyb. Tato skutečnost je zapříčiněna i faktem, že například sledování televize probíhá výlučně vždy na gauči. Zatímco výše zmíněné tři aktivity nejsou prováděny na jednom místě.

Naopak bylo zjištěno, že technologie RFID není vhodná pro monitorování každodenních aktivit [32]. RFID tagy byly umístěny na různých předmětech, které jsou nutné pro provádění každodenních aktivit. Monitorovaná osoba měla na zápěstí náramek s RFID čtečkou, která měla tagy na předmětech detekovat. Problémem je, že ne všechny každodenní aktivity vyžadují interakci s předměty, kde jsou umístěny RFID tagy. Mnoho aktivit (např. mytí nádobí) také zahrnuje interakci s předměty, kde nejsou tagy umístěny, protože jsou tyto předměty kovové, či se používají v mikrovlnné troubě. Někdy RFID čtečka nenačetla tag, protože byl náramek umístěn na druhé ruce, než kterou byla aktivita prováděna.

2.4 Potřeby a požadavky monitorovaných osob

Pro použití telemedicínských systémů v praxi je ovšem nejdůležitější vzít v úvahu i požadavky a potřeby samostatně žijících seniorů. Potřeba je pocit nedostatku. Existují primární potřeby, jako jídlo, pití, spánek, atd. a sekundární potřeby jako seberealizace, odpočinek atd. Požadavek je forma uspokojení potřeby. Výzkum [33] ukazuje, že nejdůležitější je monitorovat ty činnosti, které vyvolávají potíže v každodenním životě. Zdravotní omezení jako zapomnětlivost, špatný zrak, sluchová omezení, atd. zvyšují u samostatně žijících seniorů potřebu využívat telemedicínský systém. Stejný výzkum se zabývá i Maslowovou pyramidou potřeb. Tuto hierarchii lidských potřeb definoval v roce 1943 americký psycholog Abraham Harold Maslow [34]. Tato teorie je založená na faktu, že každý člověk má pět základních potřeb:

1. fyziologické potřeby
2. potřeba bezpečí a jistoty
3. sociální potřeby
4. potřeba uznání a úcty
5. potřeba seberealizace

Podle této teorie je nutné, aby byly potřeby uspokojovány od nejnmutnějších fyziologických potřeb. Mezi tyto fyziologické potřeby patří dýchání, regulace tělesné teploty, jídlo, pití, fyzické aktivity, rozmnožování, vylučování a vyměšování. Jsou-li uspokojeny fyziologické potřeby, začne narůstat touha uspokojit potřeby jistoty a bezpečí. Mezi tyto potřeby patří jistota zaměstnání, fyzická bezpečnost, jistota rodiny, jistota zdraví. Dále je potřeba uspokojit potřeby sociální, které se hlavně skládají z citových vztahů jako přátelství, partnerský vztah a potřeby mít rodinu. Po uspokojení sociálních jistot přichází potřeba uznání a úcty, kdy člověk chce být přijímán, oceňován a respektován ostatními. Poslední stupeň Maslowovy pyramidy potřeb je seberealizace, kdy přichází potřeba naplnit své schopnosti a dále snaha být tím nejlepším člověkem.

V souladu s Maslowovou pyramidou výzkum [33] uvádí, že nejdůležitější funkcionalitou telemedicínského systému je alarm, který upozorní dohledovou stanici či pečovatele, o nebezpečné situaci, v níž se sledovaná osoba nachází. Senioři mají největší strach o svojí bezpečnost, především mají strach z rizikových situací, jako jsou pády či jiné úrazy. Teprve po uspokojení potřeby bezpečí a jistoty, sledovaná osoba cítí potřebu uspokojit například sociální potřeby v podobě sociálního kontaktu. Další výzkum [35] uvádí, že klíčové potřeby starších lidí jsou cítit se v bezpečí, žít doma samostatně a být se schopen starat sám o sebe a o své domovy. Výzkum uvádí i obavy vznikající při použití telemedicínských systémů. Mezi tyto obavy patří: ztráta soukromí, zabezpečení nashromážděných dat, finanční náklady na telemedicínský systém, snížení sociální interakce a včasná reakce pečovatelem či jiným zdravotním pracovníkem na nebezpečnou situaci.

Dle výzkumu [9] jsou tedy nejdůležitější potřeby a požadavky starších lidí na plnohodnotný život tyto: celkový zdravotní stav, údržba domácnosti, spánek, osobní hygiena, dušení rovnováha, sociální interakce, příjem léků, bezpečnost, jídlo a pití.

Důležité je při návrhu telemedicínských systémů zohlednit i požadavky starších lidí. Výzkum [36] došel k závěru, že starší lidé požadují, aby tyto systémy byly snadně použitelné a to i při případné změně systému. Dále by měly být systémy integrovány do technologií, které již umějí ovládat a dané systémy by měly podporovat samostatnost

starších lidí. Důležitý požadavek je, aby telemedicínský systém používal senzory, které nejsou umístěné na těle, dále by neměl systém porušovat soukromí sledované osoby, měl by být snadno použitelný a měl by poskytovat validní informace. Systém by měl být i odolný a spolehlivý. Kromě tohoto, by měl být monitoring všudypřítomný a nemělo by být nutné aktivování systému sledovanou osobou.

2.5 Potřeby a požadavky ošetřujících osob

Výzkum [7] provedený v zařízení s asistivní péčí se zaměřil i na požadavky a potřeby monitorování z hlediska ošetřujících osob. Ošetřující osoby přicházejí s telemedicínským systémem často do kontaktu, jedná se o osoby jako zdravotní sestry či zdravotníci, pečovatelé, lékařští specialisté, jako praktičtí lékaři a fyzioterapeuti. Výzkum byl prováděn ve formě rozhovorů, kdy byl použit seznam otázek, který byl zaměřen na nejdůležitější potřeby a požadavky starších lidí. Tyto potřeby a požadavky pro připomenutí jsou: celkový zdravotní stav, údržba domácnosti, spánek, osobní hygiena, duševní rovnováha, sociální interakce, příjem léků, bezpečnost, jídlo a pití.

Z rozhovorů s pečovateli vyplývá, že se především starají o bezpečnost seniorů. Konkrétně se jedná o činnosti jako zabránění pádů a detekce zmatenosti u seniorů. Pečovatelé se dále zaměřují na odchylky v chování seniora. Jedná se například o detekci odchylek při spánku, kdy senior může opouštět lůžko během noci. Dále pečovatel detekuje odchylky při příjmu léků, kdy sleduje dávkování a čas. Samozřejmě pečovatel také sleduje příjem jídla a pití, kdy se zajímá o množství a čas jídla a pití. Důležité je sledování celkového zdravotního stavu seniora, kdy je kontrolován krevní tlak, tělesná teplota, atd., ale je také důležité sledovat například i příznaky demence. Sociální interakce je také důležitý ukazatel celkového stavu seniora, je sledováno jak často a kdy opustil senior dům kvůli sociální interakci.

Lékařští specialisté uvádí, že považují za nejdůležitější monitorovat míru aktivity seniora, sebezanedbávání a sociální interakci. Na základě tohoto, navrhli specialisté měřitelné ukazatele pro tyto činnosti. Například u míry aktivity seniora může být měřen fyzický pohyb, jako ušlá vzdálenost či kolikrát senior opustil dům. Pro zjištění sebezanedbávání jsou monitorovány aktivity, které definoval Katz [30]. Tedy je monitorováno, zda byl jedinec na toaletě, dále koupání, braní léků, vaření, oblékání a jídlo. Sociální interakce je vyhodnocována podle počtu návštěv a podle toho kolikrát

senior opustil svůj domov. Lékařští specialisté navrhnou se zaměřit právě na měření odchylek v těchto činnostech, aby bylo možno na rizikovou situaci včas zareagovat.

Z výsledků rozhovorů vyplývá, že telemedicínské systémy pro monitorování osob jsou důležité hlavně u osob, kde existuje vysoké riziko hospitalizace v nemocnici. Důležité je při monitorování zvážit dopad na soukromí monitorované osoby. Ošetřující osoby se shodli, že by mohl monitoring osob ulehčit v budoucnosti jejich práci a mohl by poskytnout i vyšší kvalitu života pro samostatně žijící seniory. Největší výhodu vidí lékařští odborníci v pravidelném a preventivním monitorování. Z rozhovorů s odborníky vyplývá, že péče je seniorům, z jejich strany, poskytována podle ukazatelů zdravotního stavu a duševní rovnováhy. Ošetřující osoby nemusí být schopny si vytvořit objektivní názor na daného seniora, kvůli nepravidelnému sledování. Pokud ale existují průběžné informace o stavu seniora, může být zabráněno hospitalizaci či dokonce úmrtí seniora.

Jak již bylo zmíněno v kapitole 2.3 Monitorování každodenních aktivit, senioři mají největší strach o svojí bezpečnost. Nicméně odborníci se shodují, že by telemedicínské systémy měly dále monitorovat i míru aktivity seniora, sebezanebávání a sociální interakci.

Aby senioři přijali telemedicínské systémy, je nutné zohlednit i požadavky ošetřujících osob na systém. Požadavky ošetřujících osob jsou velmi podobné požadavkům, které měli senioři (viz. kapitola 2.4 Potřeby a požadavky monitorovaných osob). Z výzkumu, ale vyplývají i nové požadavky. Použité senzory by měly být nenápadně rozmístěny po domově seniora, bez viditelného vedení. Dá se očekávat, že by starší lidé přijali telemedicínské systémy, aby zůstali co nejdéle samostatní, nezávislí a nemuseli by je navštěvovat pečovatelé, protože mnoho seniorů považuje návštěvy ošetřovatelů za nepříjemné.

3 Hardware základnové stanice a pomocných obvodů

V této kapitole je popsána základnová stanice a její hardwarové vybavení umožňující připojení senzorů. Konkrétně je zde popsán bezdrátový snímač – akcelerometr.

3.1 Pomocný hardware základnové stanice

Základnová stanice se skládá z desky a mikrokontroléru STM32F207VGT6. Převodník Asix UDS1 (obr. 3.1) [45] s modulem UMS1B [46] převádí USB na RS-232. Převodník umožňuje připojení k PC pomocí USB. Základnová stanice obsahuje rozhraní RS-232 a pro připojení k PC bez sériového portu je využit právě převodník UDS1 se standardním obvodem od FTDI.



Obr. 3.1 Převodník Asix UDS1 s modulem UMS1B

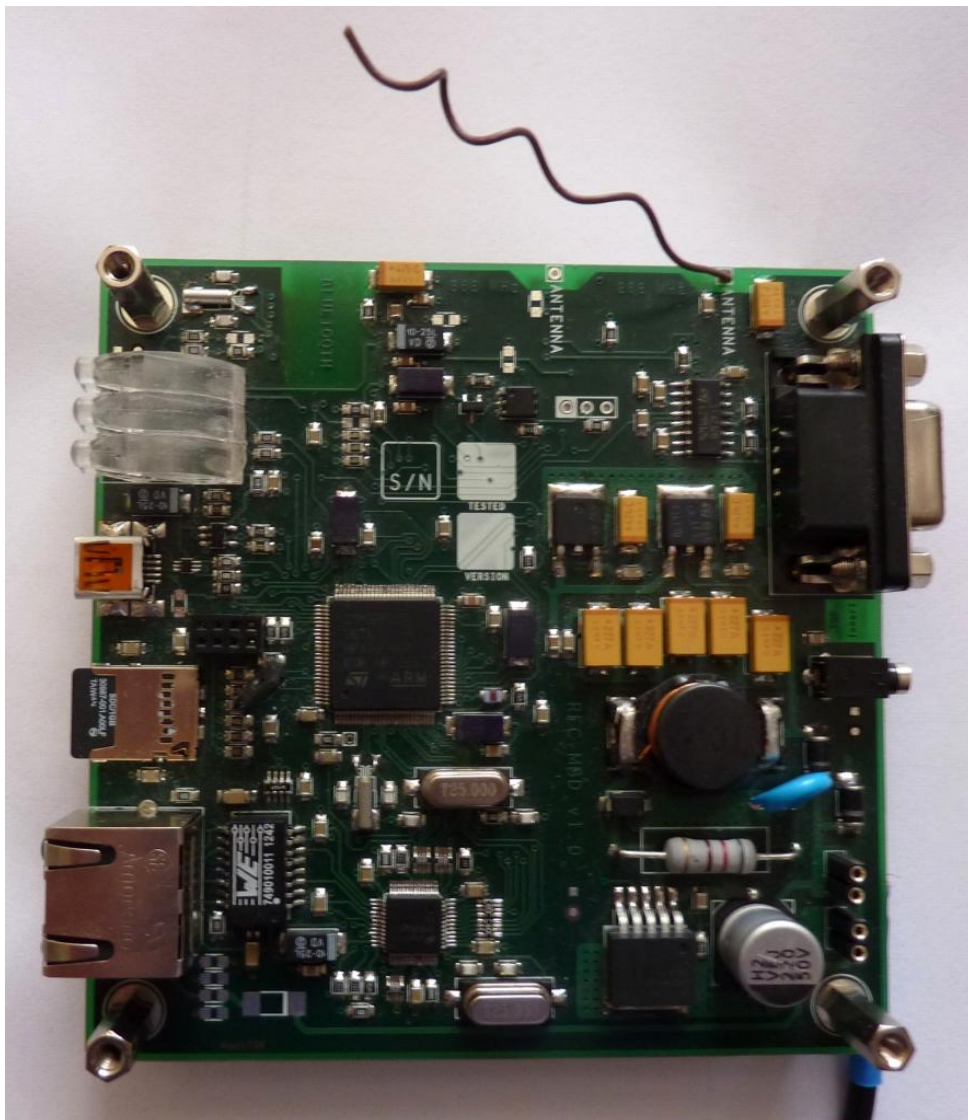
Pro nahrání firmwaru do základnové stanice je použit discovery kit STM32L1 (obr. 3.2). Discovery kit je připojen přes rozhraní SWD k základnové stanici. Spojení PC s kitem je realizováno pomocí USB.



Obr. 3.2 Discovery kit STM32L1

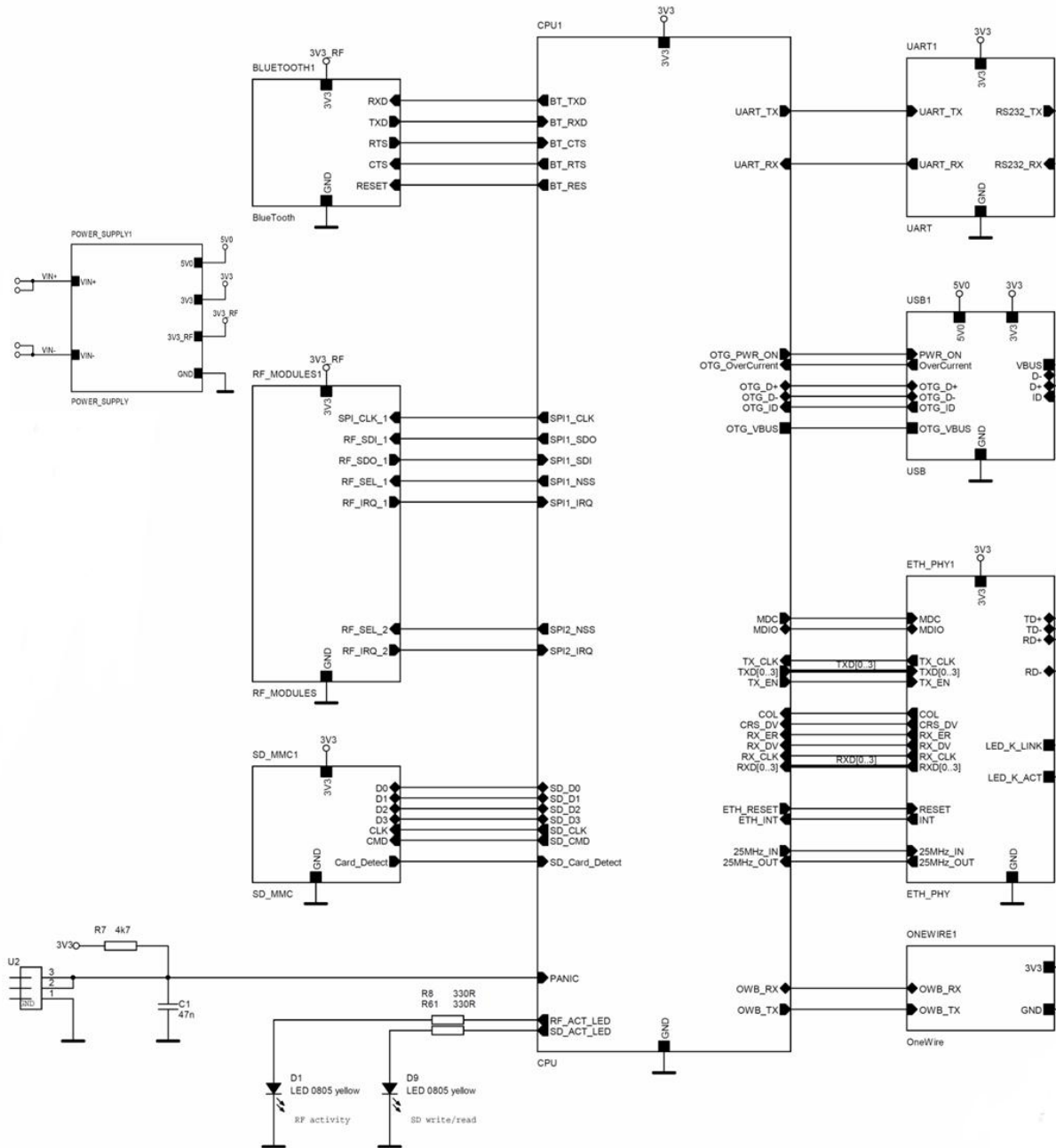
3.2 Popis hardwaru základnové stanice

Základnová stanice (obr. 3.3) je realizována na desce REC-MBD verze 1, která slouží pro vývoj aplikací na katedře Teorie obvodů na ČVUT FEL.



Obr. 3.3 Základnová stanice

Blokové schéma základnové stanice je znázorněno na obr. 3.4. Z blokového schématu je zřejmé připojení jednotlivých periférií k mikrokontroléru STM32F207VGT6. Layouty základnové stanice jsou v příloze B.



Obr. 3.4 Blokové schéma základnové stanice

3.3 Mikrokontrolér STM32F207VGT6

Mikrokontrolér STM32F207VGT6 od firmy STMicroelectronics je nejdůležitějším prvkem základnové stanice. V této kapitole je popsán mikrokontrolér STM32F207VGT6, architektura ARM a rodina ARM Cortex-M3, do které mikrokontrolér patří.

3.3.1 Architektura ARM

Architektura ARM (Advanced RISC Machines) byla vyvinuta v Británii ve firmě ARM Limited. První mikrokontrolér byl vyvinut touto firmou již v roce 1984. Jedná se o 32 bitovou architekturu, která je založena na filosofii RISC. Procesory s touto architekturou obsahují 44 základních instrukcí, kdy je šířka instrukce 32 bitů. Existují čtyři základní režimy, ve kterých může procesor pracovat. Jedná se o uživatelský režimUSR, privilegovaný režim supervizora SUP, privilegovaný režim přerušení IRQ a privilegovaný režim rychlého přerušení FIQ [39]. Dnes je architektura ARM v 90% všech 32 bitových RISC procesorů ve vestavěných zařízeních [37].

Výhoda procesorů s architekturou ARM je ve velkém výpočetním výkonu, malé velikosti kódu a v energeticky úsporných vlastnostech. Využití této architektury je v mobilním odvětví (smartphony, PDA, přenosné herní konzole, kalkulačky, atd.), v průmyslu a vestavěných systémech (bankovní automaty, průmyslová zařízení, automobilové řídicí jednotky, switche, routery, atd.) a ve větších zařízeních jako MID (Mobile Internet Device) či netbookách [37].

Současné mikrokontroléry ARM mohou nejčastěji pracovat s třemi instrukčními sadami najednou, mezi jednotlivými sadami se dá přepínat [38]. Za prvé se jedná o instrukční sadu ARM (instrukce má šířku 32 bitů), instrukční sada Thumb (instrukce má šířku 16 bitů) a instrukční sada Thumb 2 (instrukce má proměnnou délku). Přístup do paměti je možný pouze pomocí instrukcí Load/Store. Tyto instrukce načítají či ukládají obsah pracovních registrů do operační paměti. Tento fakt výrazně zjednodušuje výkonnou jednotku procesoru (Execution Unit). Dalším typem instrukcí jsou aritmetické operace, bitové a logické operace, skoky a instrukce pro práci s registry CPSR či SPSR. CPSR a SPSR jsou stavové registry, kde jsou uloženy bity, které určují, kterou instrukční sadu mikroprocesor právě zpracovává, příznaky používané u SIMD operací, bitové příznaky nastavované ALU a příznak zpracování bajtů (little endian a big endian).

Existuje více rodin architektury ARM, ale mnoho těchto verzí je už zastaralých. Základní vlastnosti těchto rodin [39]:

- ARM1-ARM3 – jsou dnes již zastaralé architektury. Do těchto rodin patří architektury ARMv1 (jádro ARM1), ARMv2 (jádro ARM2) a ARMv2a (jádra ARM250 a ARM2a). Cache o velikosti 4K měl až ARMv2a a dosahoval výkonu 12 MIPS @ 25 MHz.
- ARM6 – dnes se také jedná o zastaralou technologii. Zástupce této rodiny je architektonická verze ARMv3 (jádra ARM60, ARM600 a ARM610). Verze s jádrem ARM610 dosáhla výkonu 17 MIPS @20 MHz.
- ARM7 – Jedná se opět o zastaralou verzi ARMv3 (jádra ARM700, ARM710, ARM710a, ARM7100, ARM7500 a ARM7500FE). Tato verze přinesla 8 KB cache a vyšší výkon než předchozí verze.
- ARM7TDMI – verze ARMv4T (jádra ARM7TDMI(-S), ARM710T, ARM720T a ARM740T). Tato architektura přidala třístupňovou pipeline, MMU, MPU a měla 8kB cache. Dále sem patří verze ARMv5TEJ (jádro ARM7EJ-S), která přinesla pětistupňovou pipeline, DBX a DSP instrukce. Tento procesor byl použit například v Game Boyi, Nintendu DS či iPodu.
- StrongARM – verze ARMv4 (jádra SA-110 a SA-1110) přinesla až 16 kB cache. Procesor byl použit například v PDA.
- ARM8 – opět se jednalo o verzi ARMv4 (jádro ARM810). Tato architektura využívá pětistupňovou pipeline, statickou predikci skoku, paměť s dvojnásobnou propustností a unifikovanou 8 kB cache.
- ARM9TDMI – verze ARMv4T (jádra ARM9TDMI, ARM920T, ARM922T a ARM940T) využívá pětistupňovou pipeline a 16 kB/kB MMU. Zástupcem této rodiny je procesor Samsung S3C2442.
- ARM9E – verze ARMv5TE (jádra ARM946E-S, ARM966E-S, ARM968E-S a ARM996HS) a ARMv5TEJ (jádro ARM926EJ-S). Tento procesor je jeden z nejpoužívanějších, například v mobilních telefonech (Soni Ericsson série K a W) Texas Instruments (OMAP1710, OMAP1610,...) či Freescale i.MX21, i.MX27.
- ARM10E – opět se jedná o verze ARMv5TE (jádra ARM1020E a ARM1022E) a ARMv5TEJ (jádro ARM1026EJ-S), které využívají šestistupňovou pipeline a 32 kB/32 kB cache, 16 kB/16 kB cache nebo variabilní cache.

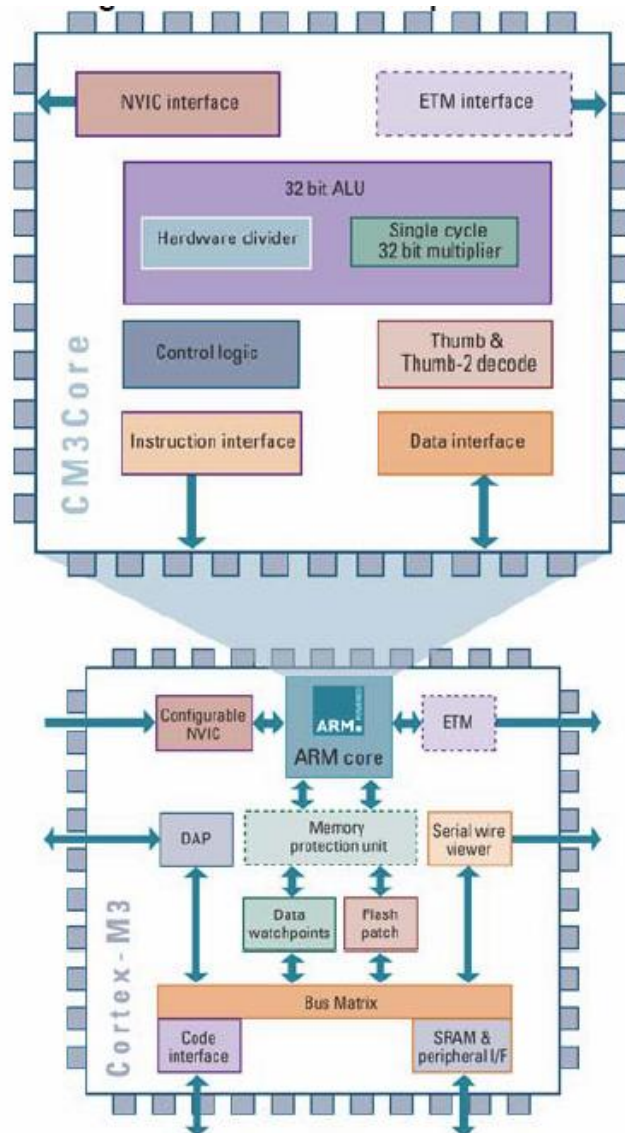
- X Scale – jedná se o jedinou verzi ARMv5TE (jádra 80200/IOP310/IOP315, 80219, IOP321, IOP33x, IOP34x, PXA210/PXA250, PXA255, PXA263, PXA26x, PXA27x, PXA800(E)F, PXA3XX, PXA900, IXC1100, IXP2400/IXP2800, IXP2850, IXP2325/IXP2350 a IXP42x). Nejrozšířenější je procesor s jádrem PXA27x, který využívá 32 kB/32 kB cache a dosahuje výkonu 800 MIPS @ 624 MHz. Toto jádro bylo použito například u Motoroly Rokr E2 či u HTC Universal.
- ARM11 – obsahuje čtyři verze ARMv6 (jádro ARM1136J(F)-S), ARMv6T2 (jádro ARM1156T2(F)-S), ARMv6KZ (jádro ARM1176JZ(F)-S) a ARMv6K (ARM11 MPCore). Tyto procesory se používají v dnešních komunikátorech. Verze s jádrem ARM1136J(F)-S využívá osmistupňovou pipeline, variabilní cache a dosahuje výkonu 700 MIPS @532-665 MHz (iMX31 SoC), 400-528 MHz. Toto jádro používá mnoho zařízení: Texas Instruments OMAP2420, Freescale MXC300-30 či Qualcomm MSM7201A.
- Cortex – tato rodina využívá především verzi ARMv7, kromě některých Cortex-M, které mají ARMv6. Architektura se dále dělí na tři profily:
 - Profil Cortex-A – Používají se hlavně v komplexnějších aplikacích a operačních systémech. Jejich výhodou je ve výkonném procesoru a podpoře virtuální paměti pomocí Memory Management Unit.
 - Profil Cortex-R – Používají se především ve výkonných real-time řídicích systémech (automobilový průmysl, atd.). Mají implementovanou ochranu paměti založenou na Memory Protected Unit.
 - Profil Cortex-M – Využívá tříúrovňovou pipeline. Jedná se o velmi cenově přístupné řešení s vysokým výkonem a snadným použitím. Jejich využití je především vhodné v levných embedded zařízeních.

3.3.2 ARM Cortex-M3

Rodina ARM Cortex-M3, do které patří používaný mikrokontrolér STM32F207VGT6, byla představena roku 2004. Jak již bylo zmíněno, procesory z této rodiny jsou především určeny pro použití v levných embedded zařízeních, jako průmyslové řídicí systémy, drátové a bezdrátové telekomunikační systémy či automobilová elektronika. Jedná se o procesory architektury ARMv7-M, které jsou založeny na harvardské architektuře. Při

návrhu byl kladen důraz na vysoký výkon, snadné použití a nízké náklady. Blokové schéma ARM Cortex-M3 je na obr. 3.5.

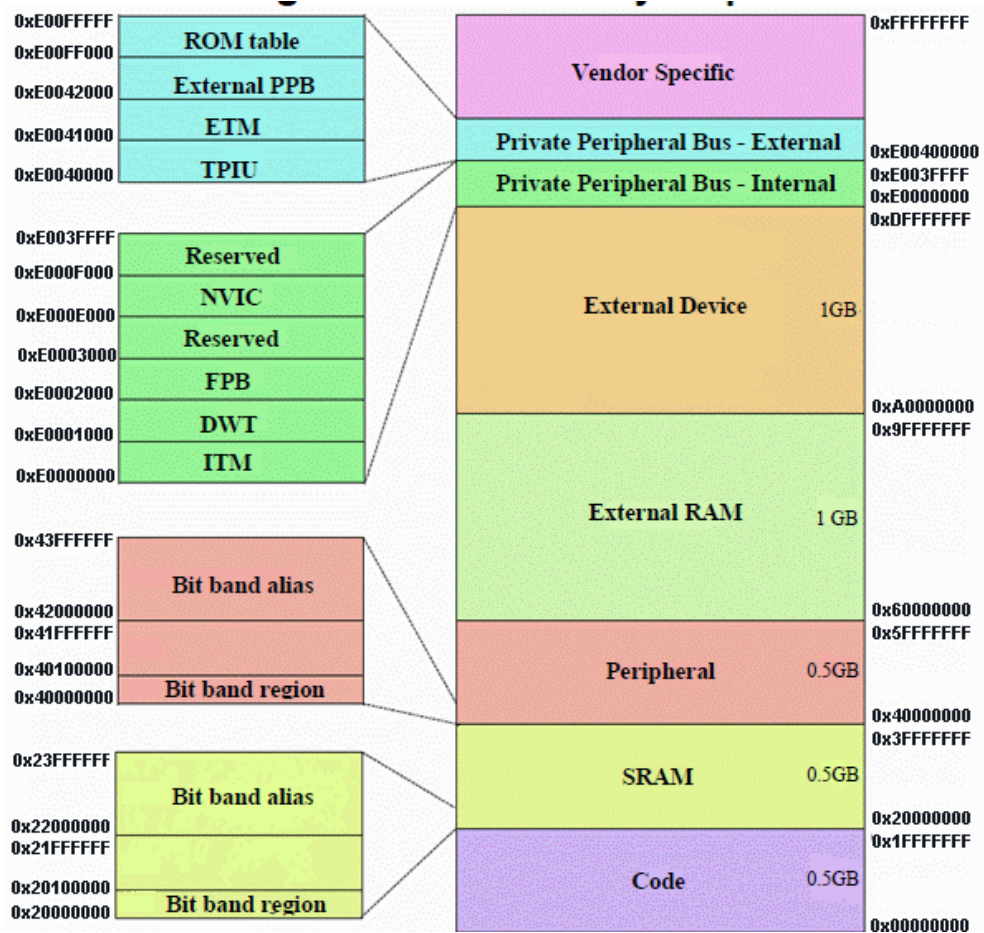
Procesor ARM Cortex-M3 má implementovanou instrukční sadu Thumb i Thumb-2. Sada Thumb-2 je zpětně kompatibilní se sadou Thumb. Instrukční sada Thumb-2 umožňuje dosáhnout o 70 % vyššího výkonu než procesory, které jsou vybaveny pouze instrukční sadou Thumb. Výhoda Thumb-2 je i v tom, že se výrazně (až třikrát) zmenší velikost kódu a zároveň se zvýší i efektivnost zpracování kódu. Jelikož je jádro založeno na harvardské architektuře a využívá třístupňovou pipeline, takže se může vykonávat více instrukcí současně. V jednom hodinovém cyklu může třístupňová pipeline vykonávat tři operace: Fetch, Decode a Execute.



Obr. 3.5 Blokové schéma procesoru ARM Cortex-M3 [40]

Jádro Cortex-M3 má rozšířenou aritmeticko-logickou jednotku (ALU), která podporuje hardwarové násobení a dělení. Dále má jádro 32 bitovou datovou sběrnici. Podporuje dva pracovní režimy (Thread a Handler) a dvě úrovně přístupu ke kódu (privilegovaný a neprivilegovaný).

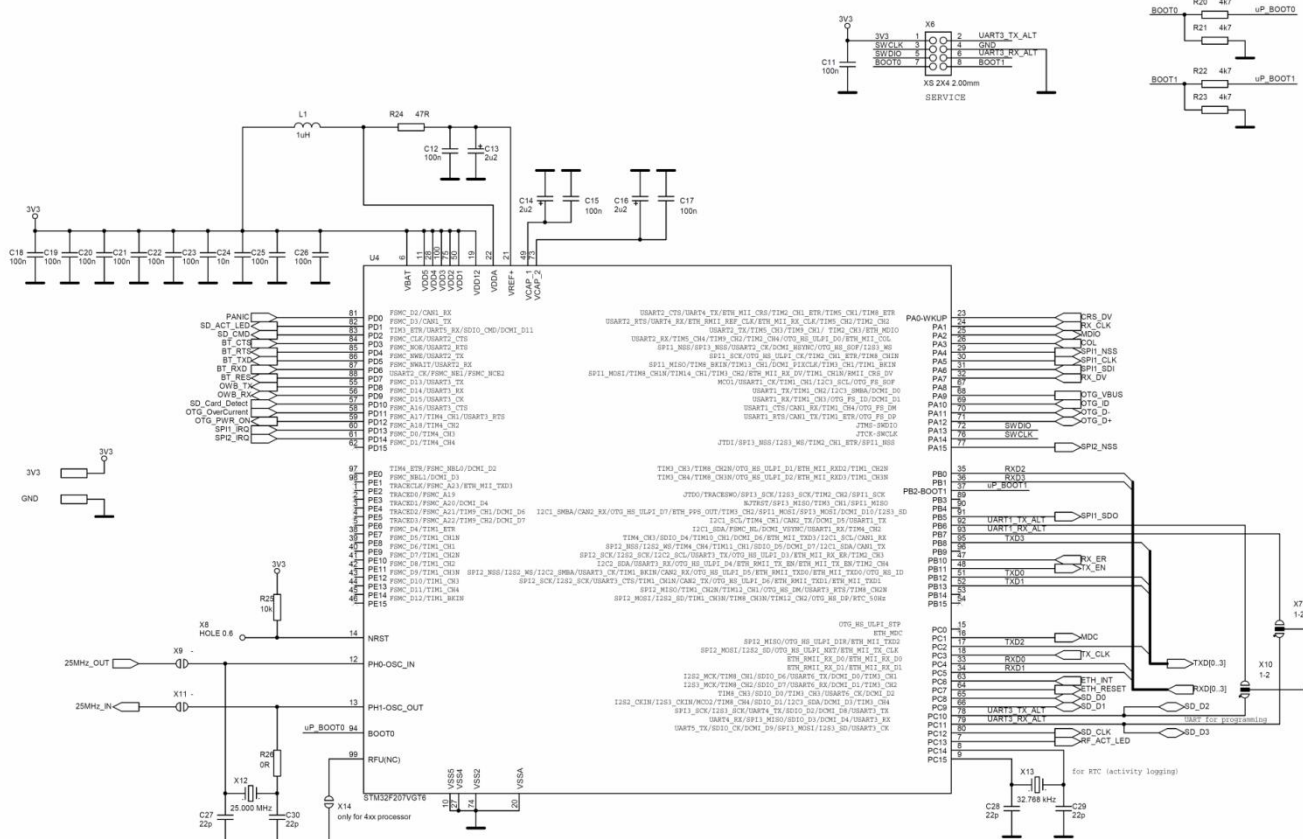
Cortex-M3 má pevnou paměťovou plochu s adresovatelným prostorem až 4 GB. Součástí paměti je pevně daný adresný prostor pro pracovní paměť RAM, externí paměť, programovou paměť FLASH a externí zařízení. Rozdělení paměťového prostoru je zobrazeno na obr. 3.6.



Obr. 3.6 Paměťový prostor procesoru ARM Cortex-M3 [40]

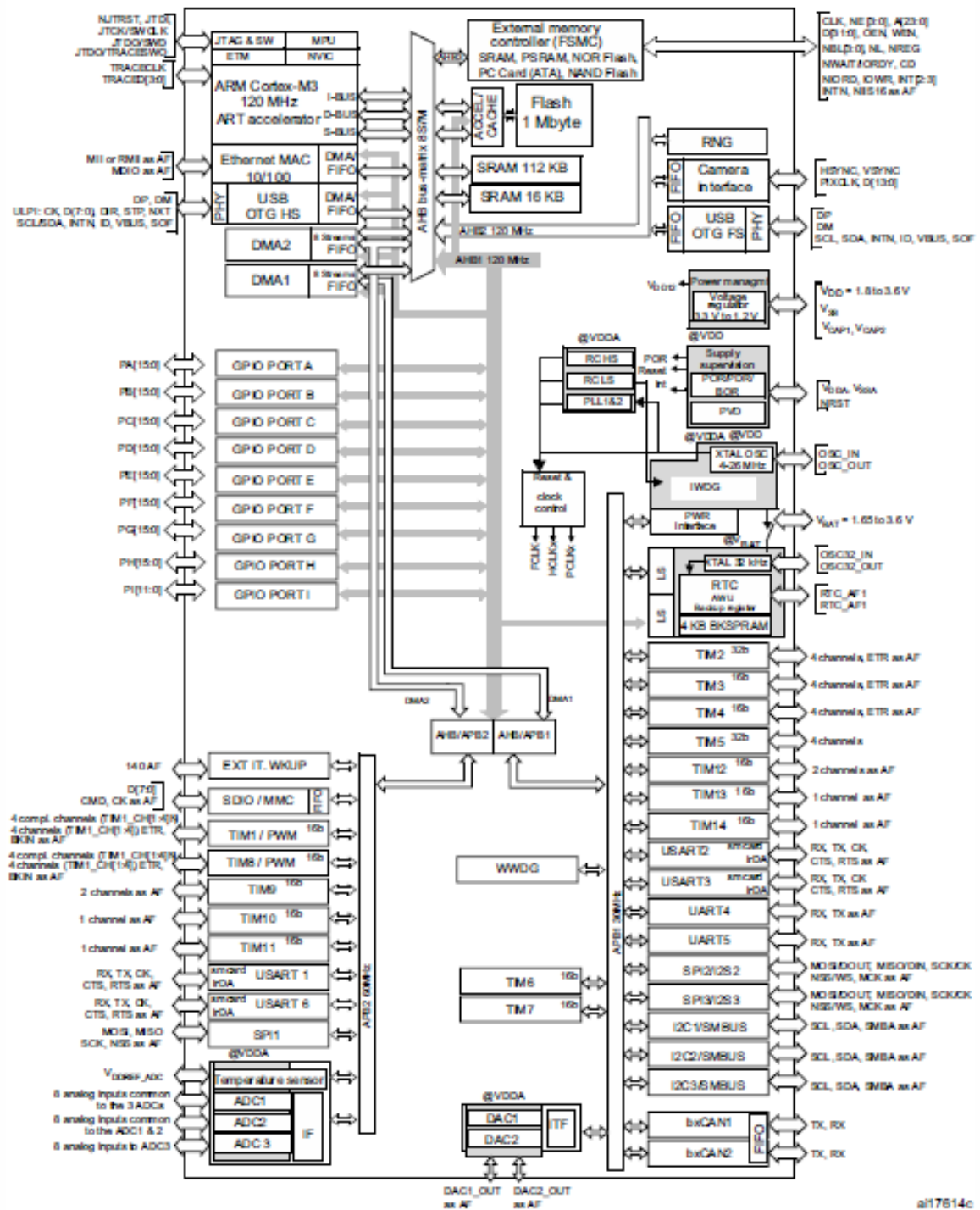
3.3.3 Popis mikrokontroléru STM32F207VGT6

Jak již bylo zmíněno, mikrokontrolér STM32F207VGT6 je založen na jádru ARM Cortex-M3. Jádro pracuje s maximálním hodinovým taktem 120 MHz a dosahuje až 150 MIPS. MCU nabízí velký výpočetní výkon díky akcelerátoru ART (Adaptive Real Time). Jádro podporuje programování přes JTAG a SWD. Mikrokontrolér je napájen napětím 1,8V-3,6V. Disponuje 1 MB Flash pamětí a 128 KB RAM. Dále mikrokontrolér má MAC vrstvu Ethernetu s vlastním DMA, 2.0 Full Speed/High Speed USB i USB Host. Je zde rozhraní SDIO pro připojení SD karty. Mikrokontrolér disponuje i řadou dalších standardních periférií jako čtyřikrát USART a dvakrát UART s maximální rychlostí 7,5 Mbit/s, třikrát SPI s maximální rychlostí 30 Mbit/s, šestnácti kanálové DMA, dvanácti bitový AD převodník s 24 kanály a až 6 MSPS, dvanácti bitový DA převodník, dvě rozhraní CAN, sedmnáct časovačů, RTC s podporou kalendáře a s možností zálohy oddělenou baterií. Mikrokontrolér obsahuje i paralelní rozhraní pro připojení CMOS



Obr. 3.7 Obvodové schéma mikrokontroléru

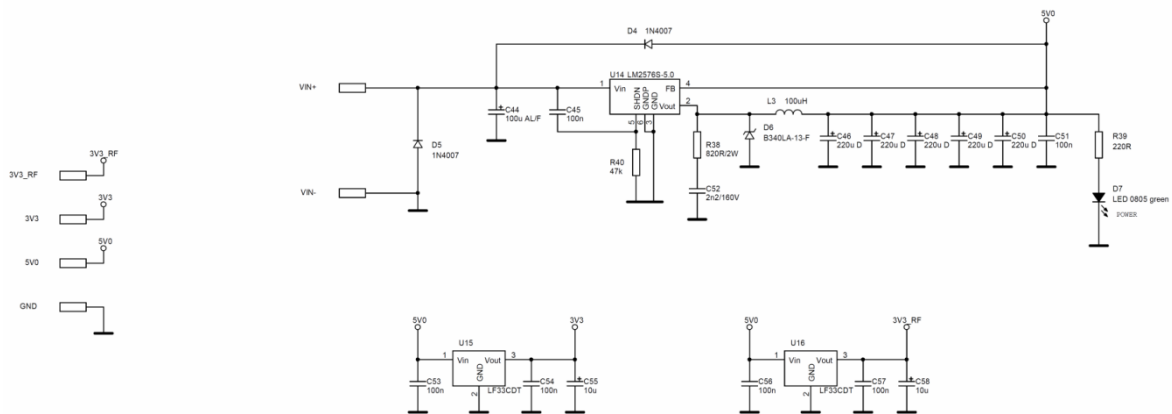
kamery, hardwarového kalkulátoru CRC a generátoru náhodných čísel. Uvedené součásti mikrokontroléru jsou připojeny přímo k jádru přes AHB bus-matrix. Na obr. 3.7 je znázorněno obvodové schéma. Na obr. 3.8 je blokové schéma mikrokontroléru, na kterém jsou zobrazeny všechny periférie a jejich připojení přes sběrnice.



Obr. 3.8 Blokové schéma STM32F207 [41]

3.4 Napájení základnové stanice

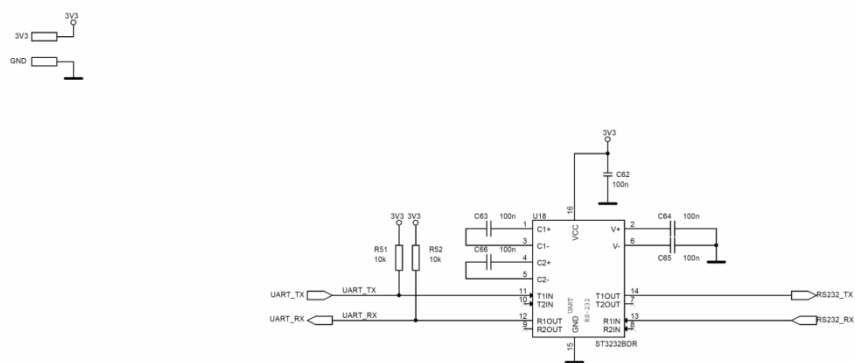
Napájení základnové stanice je zobrazeno na obr. 3.9. Integrovaný obvod U14 (LM2576S-5.0) slouží pro stabilizaci napětí 5.0 V, vstupní napětí obvodu je až 40 V. Toto napětí poté upravují obvody U15 (LF33CDT) a U16 (LF33CDT) na 3.3 V. LF33CDT umožňuje vstupní napětí až 16 V.



Obr. 3.9 Obvodové schéma napájení

3.5 UART

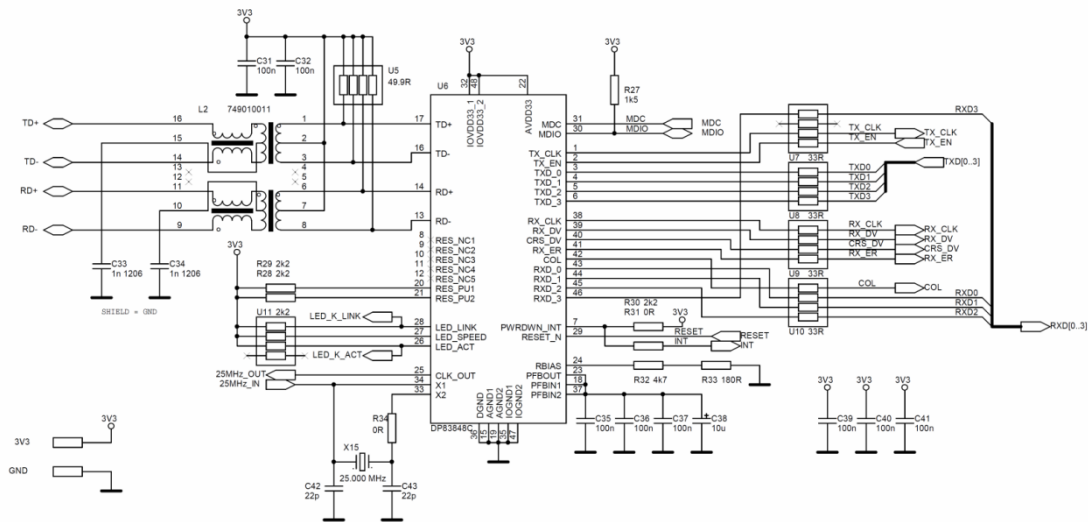
Integrovaný obvod ST3232BDR slouží k převodu UART na RS-232. Základní funkcí tohoto obvodu je převod napětí UARTu na úroveň RS-232 (obr. 3.10).



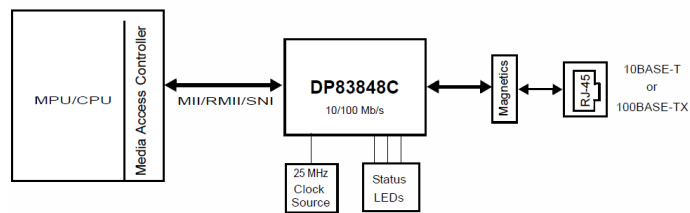
Obr. 3.10 Obvodové schéma UARTu

3.6 Ethernet

Obvodové schéma komunikace Ethernet je na obr. 3.11. Koncepce Ethernetu s obvodem DP83848C je znázorněna na obr. 3.12 [48]. Obvod DP83848C je ETH PHY, který je k mikrokontroléru připojen přes rozhraní MII. Vnější ethernetový kabel je připojen přes konektor RJ-45 na ETH transformátor s převodem 1:1. Pro vytvoření ethernetové komunikace je používán standardní softwarový modul LwIP (kapitola 4.3 LwIP).



Obr. 3.11 Obvodové schéma komunikace Ethernet

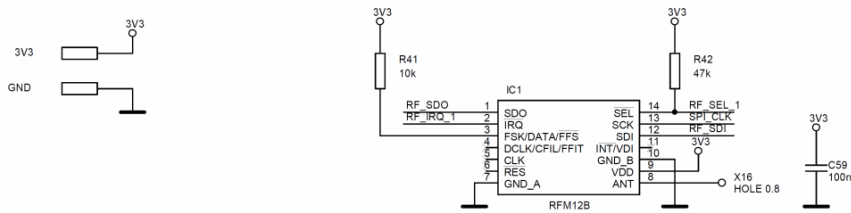


Obr. 3.12 Koncepce Ethernetu

3.7 RF modul

Deska základnové stanice je připravena pro dvojici identických RF modulů. Na desce je však osazen pouze jeden modul (obr. 3.13). Komunikační modul RFM12B – 868/S2

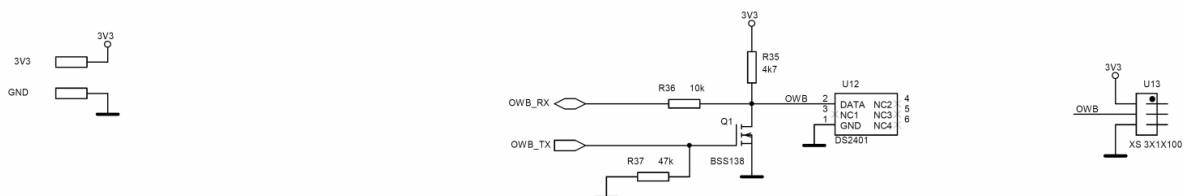
realizuje bezdrátovou komunikaci na frekvenci 868 MHz. Jsou využívány pro přenos dat ze senzorů. Komunikace modulu a mikrokontroléru probíhá přes SPI rozhraní.



Obr. 3.13 Obvodové schéma RF modulu

3.8 One-wire bus

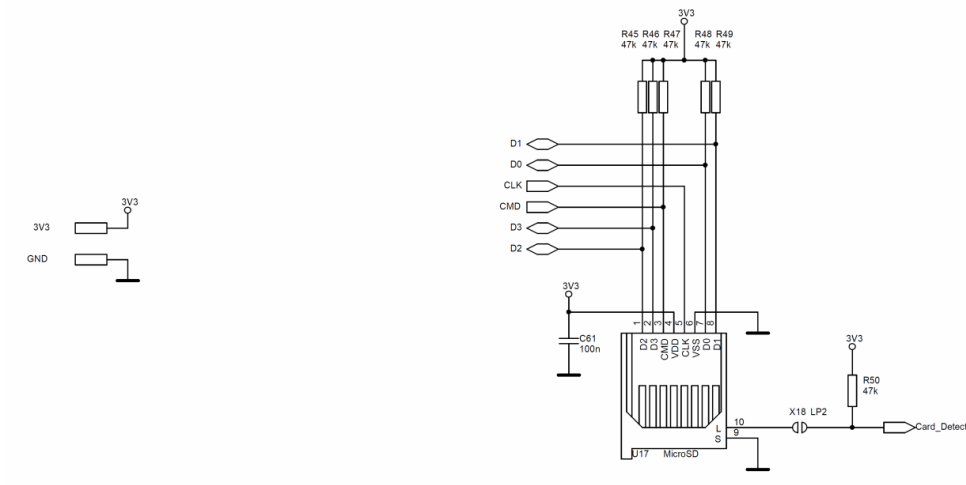
One-wire bus slouží pro jednodrátové připojení jednoduchých zařízení například čidel, pamětí či převodníků na jinou sběrnici. Obvod DS2401 je PROM paměť, která obsahuje sériové číslo modulu, tzv. ROM číslo. Samotnou komunikaci realizují pouze piny OWB_RX a OWB_TX mikrokontroléru, společně s tranzistorem Q1 a pull-up odpor.



Obr. 3.14 Obvodové schéma one-wire bus

3.9 Slot SD karty

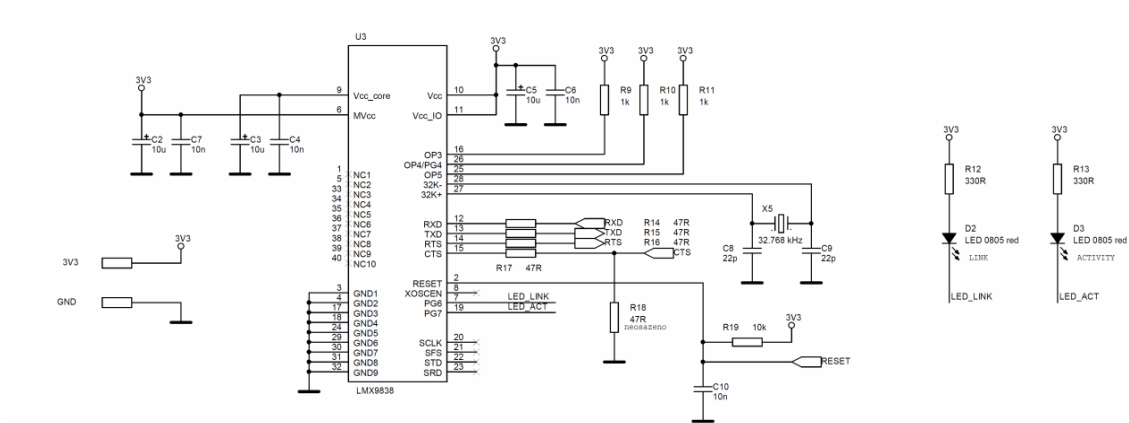
Na základnové stanici je umístěn slot micro SD karty. Vstupy a výstupy obsluhující čtení a zápis na kartu jsou generovány přímo v mikrokontroléru STM32F207 (obr. 3.15).



Obr. 3.15 Obvodové schéma slotu micro SD karty

3.10 Bluetooth

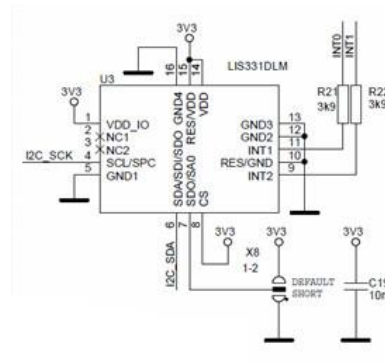
Na základnové stanici je připravena komunikace pomocí Bluetooth (obr. 3.16). Aplikace této komunikace se v této práci neuvažuje, ovšem v budoucnosti může být použita. Komunikace je realizována na základě LMX3838 přes UART 2.



Obr. 3.16 Obvodové schéma Bluetooth

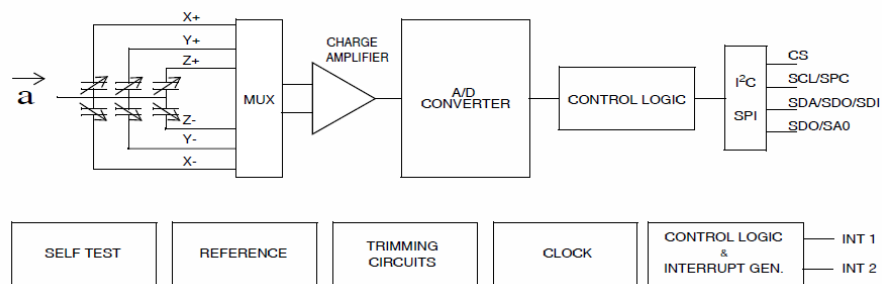
3.11 Bezdrátový senzor - akcelerometr

Základem tohoto bezdrátového senzoru [49] je mikrokontrolér STM32F100C4T6B. Pro bezdrátovou komunikaci se základnovou deskou je použit RF modul, který je identický s modulem na základnové stanici (kapitola 3.7 RF modul). Napájení řešeno dvěma bateriemi typu AAA. Na obr. 3.17 je zobrazeno obvodové schéma akcelerometru LIS331DLM.



Obr. 3.17 Obvodové schéma akcelerometru

Na obr. 3.18 [50] je blokové schéma akcelerometru. Akcelerometr zjišťuje změnu pohybu v osách X, Y a Z. Dále měří rychlost, vibrace či náklon.



Obr. 3.18 Blokové schéma akcelerometru

4 Software

V této kapitole jsou popsány knihovny využívané při návrhu firmwaru. Dále je zde popsána konfigurace a použití jednotlivých periférií, které jsou využívány při návrhu firmwaru.

4.1 Vývojové prostředí

Jako vývojové prostředí byl zvolen Keil MDK verze 5 [53] v kombinaci s debuggrem ST-Link v2. Keil podporuje velkou škálu procesorů i ARM Cortex-M3. Součástí MDK je IDE μ Vision 5. Keil μ Vision 5 podporuje programování v jazyku symbolických adres (assembler) nebo v jazyce C. Vývoj tohoto firmwaru probíhal v jazyce C. Součástí Keil MDK 5 je i ARM C/C++ kompilátor. Použitý debugger ST-Link je od firmy STMicroelectronics. Pro komunikaci s CPU je využito rozhraní SWD.

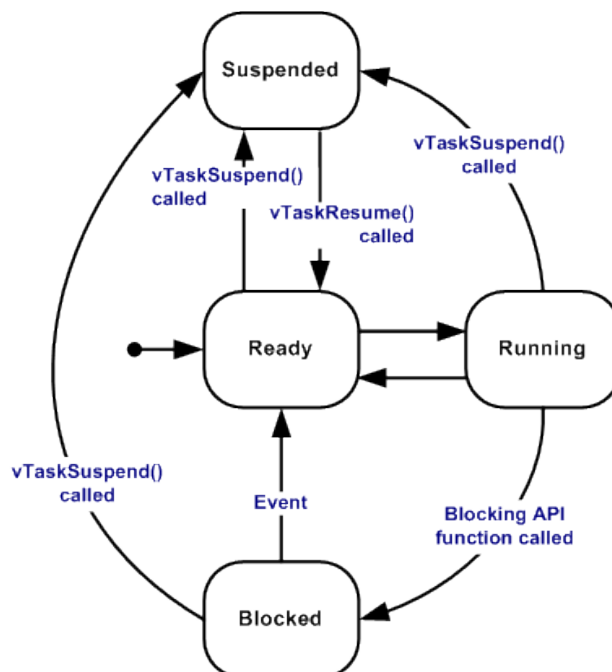
4.2 Operační systém FreeRTOS

FreeRTOS [42] je operační systém reálného času, který určen pro mikrokontroléry a je vyvíjen firmou Real Time Engineers Ltd. Jedná se o open source pod licencí GPL, jehož zdrojový kód lze volně šířit a upravovat. Operační systém je napsán v jazyce C, kromě pár funkcí napsaných v assembleru. Výhoda tohoto operačního systému je v jednoduchém použití, malé velikosti a zároveň je poskytováno velké množství ukázkových aplikací. Systém podporuje mnoho architektur od výrobců jako STMicroelectronics, Texas Instruments, Freescale, atd. Real Time Engineers Ltd. dále nabízí další dvě verze: OpenRTOS a SafeRTOS. OpenRTOS je komerční verze FreeRTOS, která rozšířena o podporu pro práci s TCP/IP, USB a systémem souborů. SafeRTOS je určen pro safety aplikace a splňuje normu SIL3 podle standartu IEC 61508. Při vývoji firmwaru byla použita verze FreeRTOS 6.0.1.

4.2.1 Úlohy (Tasks)

Výsledná aplikace, která používá operační systém FreeRTOS je rozdělena do více úloh neboli tasků. Každá úloha je určena názvem. V daný okamžik se vykonává pouze jedna úloha, ostatní úlohy mohou být v několika různých stavech. Stavy jednotlivých úloh jsou zobrazeny na obr. 4.1. Změny stavů řídí plánovač. Existují dva režimy plánovače a to preemptivní, kdy plánovač přiděluje procesor úloze s nejvyšší prioritou a nepreemptivní režim, kdy přidělování procesoru zajišťují samotné úlohy.

Úloha může být v jednom ze čtyř stavů. Existuje pouze jeden aktivní stav, kdy je úloha vykonávána a tři stavy, kdy je úloha neaktivní. Running je aktivní stav, kdy plánovač přidělil úloze procesor a tato úloha je vykonávána. První neaktivní stav je Ready. V tomto stavu je úloha, když je vytvořena. Úloha je připravena k vykonání, ale čeká na přiřazení procesoru plánovačem, protože je vykonávána úloha s vyšší prioritou. Dalším stavem je Blocked, kdy úloha čeká na synchronizační událost nebo na vypršení časové periody a poté úloha přejde do aktivního stavu. Poslední neaktivní stav je Suspended, kdy je úloha pozastavena funkcí `xTaskSuspend()` a do aktivního stavu úloha přechází pomocí funkce `xTaskResume()`.



Obr. 4.1 Stavový diagram úlohy [42]

Každá úloha je realizována jako funkce s nekonečnou smyčkou, která musí být před použitím vytvořena pomocí funkce `xTaskCreate()` a má následující tvar a parametry:

```
BaseType_t xTaskCreate(  
    TaskFunction_t pvTaskCode,  
    const char * const pcName,  
    unsigned short usStackDepth,  
    void *pvParameters,  
    unsigned BaseType_t uxPriority,  
    TaskHandle_t *pvCreatedTask  
);
```

Parametr `pTaskCode` je ukazatel na funkci s nekonečnou smyčkou. Další parametr `pcName` je string, který obsahuje jméno úlohy. `usStackDepth` je velikost paměťového prostoru pro danou funkci. Parametr `uxPriority` definuje prioritu dané úlohy, zde platí pravidlo: čím vyšší hodnota tím vyšší priorita. Poslední parametr `pvCreatedTask` je ukazatel, ve kterém je uložena vytvořená funkce. Úloha je smazána pomocí funkce `xTaskDelete()`.

4.2.2 Fronty, semaforey a mutexy

Fronty jsou nástroj pro komunikaci mezi úlohami. Obvykle se jedná o zásobník typu FIFO, kdy jsou data ukládána na konec fronty. Fronty jsou specifikovány typem dat a maximální délkou dat. Přenos dat je realizován pomocí kopírování dat. Kopírování dat je náročné pro operační paměť a proto je nutné vytvářet pouze takové fronty, které jsou potřeba. Data jsou z fronty odebírána při příjmu.

Existují dva typy semaforů: binární semafor a čítací semafor. Binární semaforey jsou využívány k synchronizaci přerušení a k synchronizaci úloh. Binární semafor je fronta s jednou položkou, kdy je důležité, zda je fronta prázdná či plná. Semaforey mohou nastavovat časovou periodu pro blokování úlohy a využívají se tedy k blokování neaktivních úloh. Čítací semaforey se využívají pro počítání událostí, které se mají ještě vykonat.

Mutexy jsou binární semaforey, které jsou využívány pro ochranu systémových prostředků a zdrojů. Předcházejí tomu, aby ke komunikační sběrnici nepřístupovalo více úloh najednou, pomocí mechanismu dědění.

4.2.3 Řízení úloh

Důležité je řídit běh vytvořených úloh. Funkce `vTaskDelay()` zpozdí vykonávanou úlohu o nastavenou dobu. Úloha je zablokována od té doby, co byla tato funkce zavolána a po uplynutí doby se stane úloha opět aktivní. Další funkce pro řízení běhu úlohy je `vTaskDelayUntil()`. Tato funkce zpozdí vykonávanou úlohu přesně od určené doby. Funkce `vTaskPrioritySet()` se využívá pro změnu priority dané úlohy. Úloha je pozastavena pomocí funkce `vTaskSuspend()`, kdy úloha nebude vykonávána ani s ohledem na prioritu. Funkce je obnovena v běhu pomocí funkce `vTaskResume()`.

4.3 LwIP

LwIP stack (A Lightweight TCP/IP stack) [43] je implementace protokolu TCP/IP a je určen pro embedded systémy. Vyžaduje zhruba 40 kB nárok na programové paměti a 10 kB nárok na operační paměti. LwIP stack patří pod modifikovanou BSD licenci. Tato licence umožňuje volné šíření licencovaného obsahu. Vyžaduje pouze uvedení autora, informace o licenci a zřeknutí se odpovědnosti. Dále umožňuje komerční využití i v proprietárním softwaru bez zveřejnění zdrojového kódu.

LwIP je napsán v jazyce C a je schopen pracovat i bez operačního systému. V této práci, ale LwIP stack pracuje s operačním systémem FreeRTOS a je použita verze LwIP 1.3.2. Při návrhu firmwaru je používán upravený LwIP stack od STMicroelectronics pro mikrokontrolér STM32F207.

4.3.1 Podporované protokoly

Nejnovější verze LwIP podporuje protokoly:

- Linková vrstva
 - ARP (Address Resolution Protocol)
 - PPP (Point-to-Point Protocol)
- Síťová vrstva
 - IP (Internet Protocol) – podporována verze IPv4 i IPv6
 - ICMP (Internet Control Message Protocol)

- IGMP (Internet Group Management Protocol)
- Transportní vrstva
 - UDP (User Datagram Protocol)
 - TCP (Transmission Control Protocol)
- Aplikační vrstva
 - DHCP (Dynamic Host Configuration Protocol)
 - DNS (Domain namesystem)
 - AutoIP/APIPA
 - SNMP (Simple Network Management Protocol)
 - SNTP (Simple Network Time Protocol)

4.3.2 Způsoby komunikace

LwIP poskytuje tři způsoby komunikace: RAW API, Socket API a Netconn API [44]. RAW API je řízeno událostmi a je navrženo pro použití bez operačních systémů. Aplikace je implementována jako soubor callback funkcí, které jsou volány z LwIP jádra při vzniknuté odpovídající události. Socket API je sekvenční a zaměřuje se na kompatibilitu a přenosnost kódu. Není možné ho používat bez operačního systému. K dispozici jsou funkce socket (vytvoření socketu), bind, listen, connect, accept, read, write a close. Poslední způsob je Netconn API a je nutné ho používat s operačním systémem. V této práci je použit právě tento způsob komunikace s operačním systémem FreeRTOS. Zpracování paketů probíhá pouze ve vláknech. V netconn API jsou k dispozici funkce:

- netconn_new – vytvoří nové spojení
- netconn_delete – smaže existující spojení
- netconn_bind – přiřadí spojení k lokální IP adrese či portu
- netconn_connect – naváže spojení se vzdáleným IP/portem
- netconn_send – pošle data na vzdálený IP/port (není pro TCP)
- netconn_recv – přijímá data z netconn
- netconn_listen – nastaví TCP připojení na příjem
- netconn_accept – přijímá přicházející spojení
- netconn_write – posílá data na připojené TCP netconn
- netconn_close – zavírá TCP spojení bez jeho vymazání

4.3.3 Správa paměti

Správa paměti LwIP se především stará o alokaci a dealokaci paměti. Dále správa paměti stará o využití paměti, aby data z LwIP nezabrala veškerou paměť. Je využívána datová struktura pBuf, která reprezentuje pakety uvnitř LwIP. Tyto struktury mohou být spojeny do řetězců. Existují tři základní pBuf: PBUF_ROM, PBUF_RAM a PBUF_POOL.

4.3.4 Použití LwIP

4.3.4.1 Ethernetové rozhraní

Před inicializací LwIP pomocí funkce LwIP_Init() je nejprve nutné inicializovat Ethernet. Ke konfiguraci Ethernetu byla použita příručka [44] a knihovna STM32F2x7_ETH_Driver od STMicroelectronics. Nejprve je nutné povolit hodiny pro GPIO a Ethernet. Dále je nutné konfigurovat GPIO pro rozhraní MII, přes které komunikuje Ethernet PHY a i mikrokontrolér. Nutné je i konfigurovat přerušení NVIC pro Ethernet. Pro nastavení Ethernet rozhraní je nejprve nutné zavolat funkce ETH_DeInit(), kdy jsou vynulovány všechny registry, které používá Ethernet na defaultní hodnotu. Následuje reset a vynulování interních registrů MAC vrstvy pomocí funkce ETH_SoftwareReset(). Funkce ETH_StructInit() provede konfiguraci samotného Ethernetu pomocí struktury ETH_InitStructure, jejíž parametry jsou použity pro nastavení jednotlivých registrů rozhraní. Jako poslední je nastaveno DMA (Direct Memory Access). Po úspěšné konfiguraci Ethernet rozhraní může být inicializován LwIP stack pomocí funkce LwIP_Init().

4.3.4.2 DHCP

Pro každé zařízení je vytvořena datová struktura netif, která obsahuje mimo jiné přidělenou IP adresu zařízení, masku sítě a IP adresu výchozí brány. LwIP obsahuje protokol DHCP pomocí, kterého byly nakonfigurovány parametry zařízení. Funkce dhcp_start() povoluje na příslušném zařízení konfiguraci. Nastavena je IP adresa, maska sítě a IP adresa výchozí brány (obr. 4.2).

```
thernet Init OK
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
IP assigned by DHCP:      10.0.0.37
Netmask assigned by DHCP: 255.255.255.0
Gateway assigned by DHCP: 10.0.0.138
```

Obr. 4.2 Výpis po DHCP

4.3.4.3 TCP rozhraní

Struktury `tcp_pcb` obsahují stav daného spojení a jsou uloženy v propojeném seznamu. V této práci je využito API `Netconn`, proto je nutné pro vytvoření nového spojení nejdříve zavolat funkci `netconn_new()`, která provádí alokaci nového spojení. Dále funkce `netconn_bind()` provede asociaci struktury `tcp_pcb` s portem a lokální adresou. Poté je nutné nastavit TCP spojení do stavu, aby mohlo přijímat spojení, pomocí funkce `netconn_listen()`. Při přijetí spojení je volána funkce `netconn_accept()`. Při přijetí dat ze sítě je volána funkce `netconn_recv()`. Pro odeslání dat po TCP je volána funkce `netconn_write()`. Pro ukončení spojení je nutno zavolat funkci `netconn_close()` a také je nutné spojení vymazat pomocí funkce `netconn_delete()`.

4.3.4.4 SNTP

Jak již bylo řečeno, `LwIP` podporuje i protokol `SNTP` [43] a tento protokol je v této práci využit. `SNTP` (Simple Network Time Protocol) je zjednodušená forma `NTP` (Network Time Protocol). Tento protokol si nepamatuje stav z předchozí komunikace a ani neuvažuje zpoždění paketů.

4.4 USART/RS-232

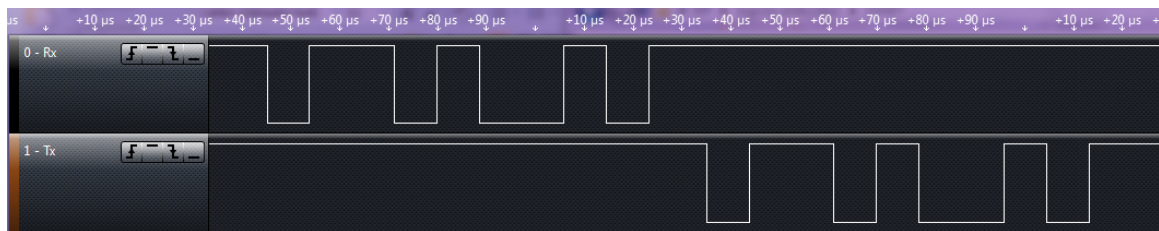
Pro ladící účely byl zprovozněn i USART. Zvolen byl asynchronní režim, který je plně duplexní. Komunikace s PC je realizována pomocí sériového portu a protokolu RS-232. Pro zobrazení komunikace po sériové lince byl zvolen Hyperterminal verze 6.1.

UART přijímá data na pinu RX a vysílá data na pinu TX. V klidové úrovni má UART log.1. Vysílání je zahájeno pomocí tzv. start-bitu a změnou úrovně na log.0. Jako první se pak vysílá LSB a jako poslední MSB. Následuje stop-bit, který změní úroveň na log.1. Po stop-bitu může začít přenos dalšího bajtu.

Pro zprovoznění UARTu byla použita knihovna `STM32F2xx_StdPeriph_Lib_V1.0.0` a její vzorové příklady [54]. Rychlost komunikace byla nastavena na 115 200 b/s, délka slova 8 bitů, bez paritního bitu, jeden stop-bit a bez HW řízení komunikace.

Přesměrována byla funkce `printf()`, tak aby posílala svůj výstup na sériovou linku. Dále byla vytvořena funkce `USART_Scanf()` pro čtení znaků z terminálu a funkce `USART_Scanf_uint()` pro čtení čísel z terminálu. Tento zdrojový kód je umístěn v souboru `serial_debug.c`.

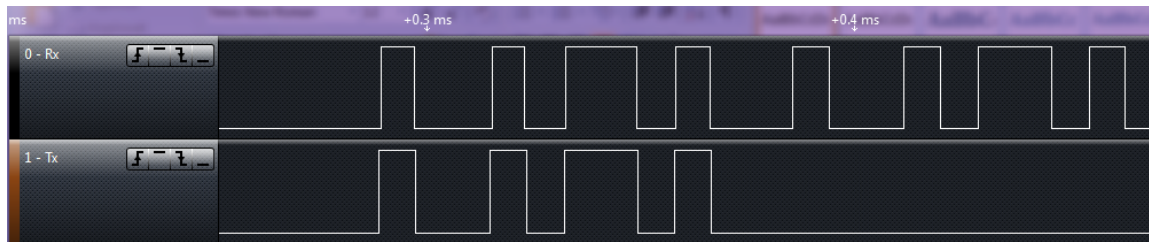
Pomocí logického analyzátoru byly nasnímány signály Rx a Tx UART u pro písmeno “K” (obr. 4.3). Písmeno “K” odpovídá dle ASCII tabulky hodnotě 0x4B (binárně 1001011). Jak je z obrázku patrné nejdříve jsou signály v klidové úrovni (log.1). Poté je odvyslán start-bit (log.0). Následuje vysílání jednotlivých bitů písmena “K” od nejméně významného bitu (1101001). Nakonec je odvyslán stop-bit (log. 0), který nastaví signál opět do klidové úrovně (log. 1).



Obr. 4.3 Signály Rx a Tx UARTu pro písmeno “K”

Nasnímán byl i signály Rx a Tx RS-232 na převodníku Asix UDS1 (obr. 4.4). Převodník Asix UDS1 slouží jako invertor signálů linky UART a upravuje výstupní napětí na úroveň RS-232. Opět byly signály nasnímány pro písmeno “K” (binárně 1001011). Nejdříve jsou signály v klidové úrovni (log. 0), poté následuje start-bit (log. 1) kdy se signál přepne do opačného stavu. Následuje vysílání bitů písmena “K” od nejméně

významného bitu (1101001). Poté je následuje stop-bit (log. 1) a signály jsou opět přepnuty do klidové úrovně (log. 0).



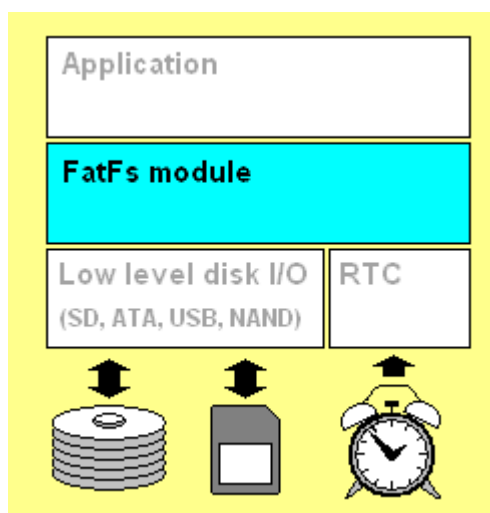
Obr. 4.4 Signály Rx a Tx RS-232 pro písmeno “K“

4.5 Micro SD karta a FatFs

Komunikace s SD kartou je realizována pomocí rozhraní SDIO. Pro komunikaci s micro SD kartou je používán driver `stm322xg_eval_sdio_sd.c`. Driver posílá příkazy SD kartě a zpracovává přijatá data a odpovědi. Tento driver obsahuje funkce pro inicializaci, zápis, čtení a mazání dat atd. Tyto funkce volají driver `stm32f2xx_sdio.c` z knihovny `STM32F2xx_StdPeriph_Driver`, který pracuje přímo s registry SDIO. Pro zprovoznění SD karty je nutné povolit přerušení a zavolat funkci `SD_Init()`.

Pro zápis na micro SD kartu byla použita knihovna FatFs R0.10 [50]. Pomocí této knihovny lze na embedded zařízeních realizovat systém souborů FAT. Knihovna FatFs je napsána v jazyku C, je nezávislá na platformě a má malou velikost. FatFs je oddělen od I/O vrstvy, proto vyžaduje funkce pro zápis a čtení z fyzického disku (obr. 4.5). Nutné je v driveru `stm322xg_eval_sdio_sd.c` implementovat funkce `SD_WriteMultiBlockFIXED()` a `SD_ReadMultiBlocksFIXED()`. Na rozdíl od standardních funkcí driveru `stm322xg_eval_sdio_sd.c` mají tyto funkce v argumentu adresu, která není v bajtech, ale v blocích. Tento fakt umožňuje práci s použitou 1 GB micro SD kartou. Funkce nutné pro zápis a čtení z fyzického disku jsou umístěné v souboru `diskio.c` a jsou následující:

- `disk_status` – získání informací o stavu disku.
- `disk_initialize` – inicializace disku
- `disk_read` – přečtení sektoru
- `disk_write` – zápis do sektoru
- `disk_ioctl` – řídicí zařízení, které je závislé na funkcích
- `get_fattime` – získání aktuálního času



Obr. 4.5 FAT File System Modul [50]

Knihovna FatFs dále poskytuje funkce pro práci s SD kartou. Na základě těchto funkcí byly vytvořeny v souboru FATFS.c funkce, které byly použity při návrhu firmwaru. Funkce poskytované FatFs, které jsou umístěné ff.c jsou následující:

- `f_mount` – funkce při připojení či odpojení souboru systémů. Funkce dále zaregistruje SD kartu jako pracovní jednotku. Musí být volána vždy na začátku a na konci práce se systémem souborů.
- `f_open` – tato funkce otevírá soubor pro čtení či zápis (flagy `FA_READ` a `FA_WRITE`), kontroluje existenci souboru (flag `FA_OPEN_EXISTING`) a vytváří nové soubory (flagy `FA_CREATE_NEW` a `FA_CREATE_ALWAYS`). Dále vytváří objekt pro přístup k souboru.
- `f_close` – funkce uzavírá aktuálně otevřený soubor. V jednu chvíli může být otevřen pouze jeden soubor, jinak dochází k chybám.
- `f_read` – čte z právě otevřeného souboru do bufferu určitý počet bajtů.
- `f_write` – zapisuje do právě otevřeného souboru.
- `f_lseek` – funkce pohybuje ukazatelem pro čtení a zápis v aktuálně otevřeném souboru.
- `f_truncate` – zmenší velikost souboru.
- `f_sync` – vymaže data z cache.
- `f_forward` – přečte data ze souboru a předá je streamovacímu zařízení.
- `f_stat` – kontroluje existenci souboru či adresáře.
- `f_opendir` – otevírá existující adresář.

- `f_closedir` – zavírá aktuálně otevřený adresář.
- `f_readdir` – čte obsah aktuálně otevřeného adresáře.
- `f_mkdir` – vytváří nový adresář.
- `f_unlink` – odstraní konkrétní soubor či adresář.
- `f_chmod` – mění atributy zvolených souborů či adresářů.
- `f_utime` – mění timestamp souboru či adresáře.
- `f_rename` – přejmenovává soubor či adresář.
- `f_chdir` – přepíná aktuální adresář.
- `f_chdrive` – přepíná aktuální jednotku.
- `f_getcwd` – načítá aktuální adresář.
- `f_getfree` – vrátí počet volných clusterů.
- `f_getlabel` – vrací pojmenování jednotky.
- `f_setlabel` – nastavuje jméno jednotky.
- `f_mkfs` – vytvoří souborový systém na disku.
- `f_fdisk` – rozděluje fyzický disk.
- `f_gets` – přečte string.
- `f_putc` – zapíše znak.
- `f_puts` – zapíše string.
- `f_printf` – zapíše formátovaný string.
- `f_tell` – vrací aktuální ukazatel čtení či zápisu.
- `f_eof` – test konce souboru.
- `f_size` – vrací velikost souboru.
- `f_error` – testuje chyby souborů.

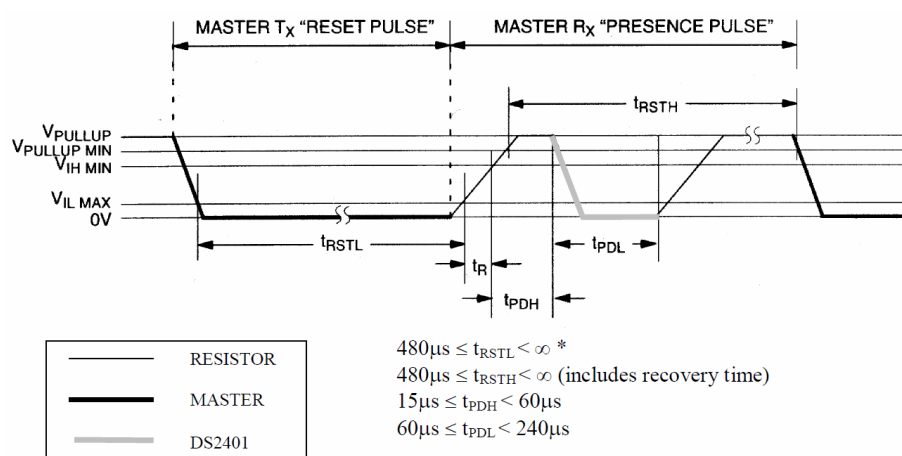
4.5 One-wire bus

Sběrnice one-wire bus funguje na modelu master/slave, kdy k řídicí jednotce (master) lze připojit více zařízení (slave), například čidel. Každé zařízení je adresováno pomocí 64 bitového čísla ROM, které je uloženo v PROM paměti každého zařízení. Nejnižších 8 bitů tohoto čísla je Family Code, který určuje typ zařízení. Zařízení DS2401 má Family Code vždy 0x01. Další 48 bitů čísla ROM je sériové číslo zařízení Serial Number. Nejvyšších

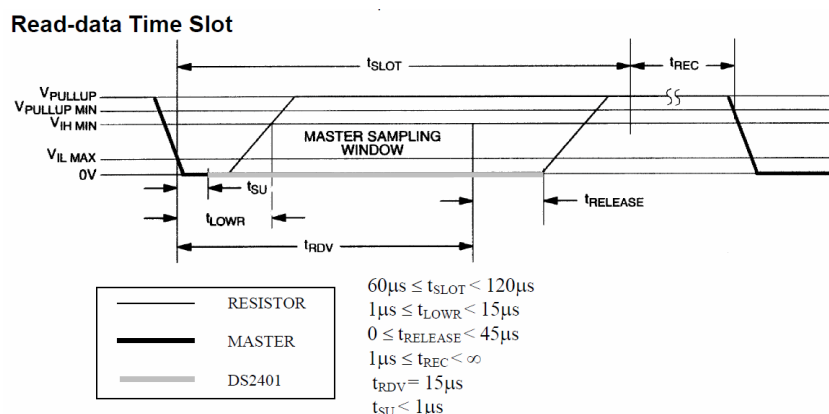
8 bitů čísla ROM je kontrolní součet CRC a je vypočítáváno polynomem $CRC = x^8 + x^5 + x^4 + 1$.

Před zahájením komunikace je nutné, aby master zařízení vyslalo reset puls. Reset puls přepne datový vodič do log.0 a je na této úrovni ponechán alespoň 480 μ s. Poté je sběrnice uvolněna na log.1 a naslouchá. Následuje presence puls, během kterého se má potvrdit přítomnost zařízení na sběrnici. Připojené zařízení je detekováno vzestupnou hranou a po 15 až 60 μ s je přepnuta sběrnice do log. 0 na dobu 60 až 240 μ s (obr 4.6).

Pokud je ohlášeno nějaké zařízení tak můžou být vysílána přijímána data v time slotech. Time sloty jsou 60 až 120 μ s dlouhé a mezi jednotlivými time sloty musí být alespoň 1 μ s mezera. Během jednoho time slotu může být přijat či odeslán pouze jeden bit. Existují čtecí a zapisovací time sloty a jsou zobrazeny na obr. 4.7 a obr. 4.8. Jsou dva druhy zapisovacích slotů: write-one time slot a write-zero time slot.

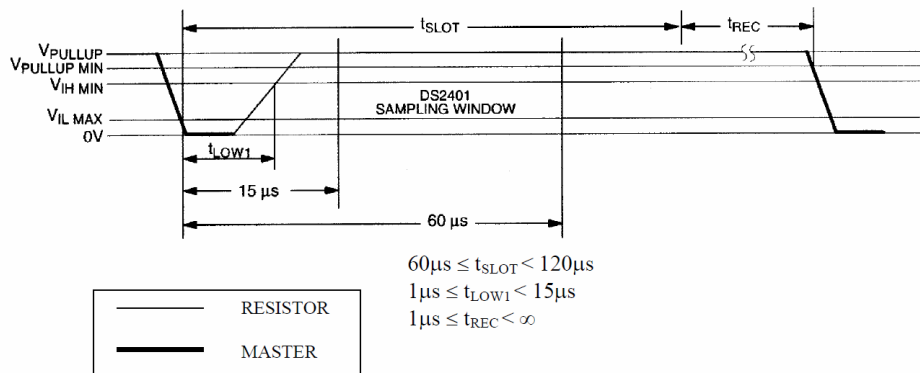


Obr. 4.6 Inicializační sekvence [51]

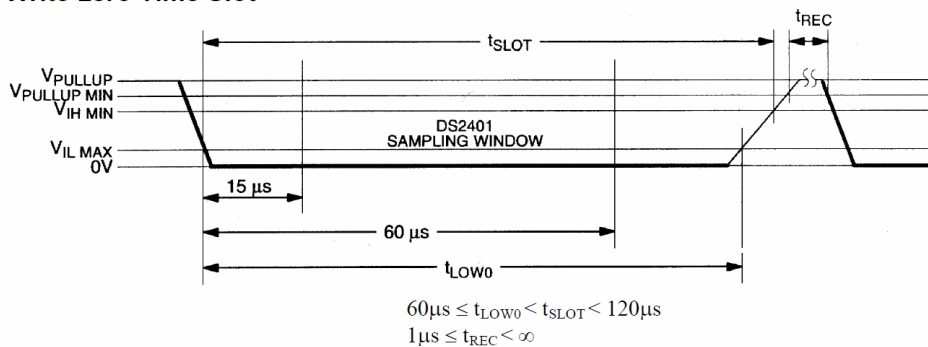


Obr. 4.7 Time sloty pro čtení dat [51]

Write-One Time Slot



Write-zero Time Slot



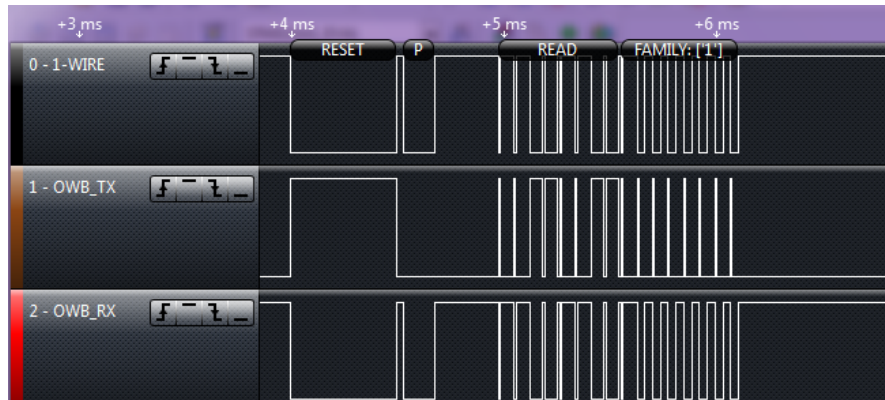
Obr. 4.8 Time sloty pro zápis dat [51]

ROM příkazy slouží k adresaci a komunikaci s konkrétním zařízením. Master vyšle příkaz a adresu zařízení. Poté na příkazy reaguje pouze adresované zařízení až do dalšího reset pulsu. První příkaz je Search ROM (0xF0) a slouží pro zjištění čísla ROM od všech připojených zařízení. Příkaz Read ROM (0x33 nebo 0x0F) slouží k zjištění čísla ROM pouze od jednoho zařízení. Může být použito, pouze pokud je ke sběrnici připojeno pouze jedno zařízení. Příkaz Match ROM (0x55) slouží k porovnání vysílaného ROM čísla masterem s ROM číslem připojených zařízení. Pokud nějaké zařízení má shodné ROM číslo, tak se přepne do režimu, kdy poslouchá a čeká na přijetí funkčního příkazu. Poslední příkaz Skip ROM (0xCC) je vhodné použít, když je připojeno pouze jedno zařízení, které tímto příkazem začne naslouchat a čeká na přijetí funkčního příkazu.

Komunikaci na sběrnici tedy můžeme rozdělit do tří po sobě následujících částí:

- inicializace
- ROM příkazy
- čtení dat

Pomocí logického analyzátoru byl nasnímán signál one-wire busu. Nejdříve proběhne reset puls a presence puls. Poté následuje příkaz Read ROM (0x33). Z ROM čísla je přečteno nejnižších 8 bitů Family Number, které je 0x01 (obr. 4.9).



Obr. 4.9 Reset a presence puls, příkaz Read ROM a čtení Family Number

Následuje čtení 48 bitů sériového čísla Serial Number, které je 381578370 (obr. 4.10). Naposledy je přečteno nejvyšších 8 bitů z ROM čísla kontrolní součet CRC, který má hodnotu 450 (obr. 4.11). Zdrojový kód je umístěn v souboru ds2401.c.



Obr. 4.10 Čtení sériového čísla Serial Number



Obr. 4.11 Čtení kontrolního součtu CRC

4.6 RF modul

RF modul je zařízení pro bezdrátovou komunikaci a zvolen byl modul RFM12b [52]. Při návrhu firmwaru byl modul použit pro přijímání dat z akcelerometru základnovou stanicí. Odesílání dat probíhá pouze na straně akcelerometru, kde je také umístěn RF modul. Komunikace s mikrokontrolérem probíhá přes rozhraní SPI. Po startu je RF modul nutné konfigurovat, protože se po startu naplní ovládací registry defaultními hodnotami. Nutné je nakonfigurovat parametry pro příjem a vysílání (nosný kmitočet, baudrate, atd.) a nastavit jestli se RF modul bude chovat jako přijímač, vysílač či bude v režimu spánku. V režimu vysílače je nutné načíst data do FIFO a příkazem data vyslat. Po dokončení přenosu mohou být vyslána další data. V režimu přijímače je nutné nastavit FIFO na režim příjmu a přijatá data do FIFO registru uložit. Data jsou potom vyčtena z FIFO a před dalším příjmem dat je nutné registr resetovat. RF modul na základnové stanici bude tedy nastaven jako přijímač a bude čekat na data. RF modul v akcelerometru bude v režimu vysílače. Funkce pro ovládání RF modulu na základnové stanici jsou umístěny v souboru rfm12b.c.

4.7 EEPROM

Byla také virtualizována EEPROM paměť do Flash paměti. S využitím se během návrhu firmwaru nepočítá, ale její budoucí využití je možné. Byly vytvořeny pouze jednoduché funkce pro ovládání EEPROM, tyto funkce jsou umístěny v souboru eeprom.c a vycházejí z příkladů knihovny STM32F2xx_StdPeriph_Lib_V1.0.0 [54]. Paměť je reprezentovaná bufferem uint32_t flash[256]. Při čtení dat jsou do tohoto bufferu načteny data z EEPROM paměti. Při zápisu dat jsou data z tohoto bufferu zapsána do EEPROM paměti.

4.8 RTC

Pro možnost zjištění aktuálního času byly zvoleny hodiny reálného času (RTC). Zdrojový kód vychází z knihovny STM32F2xx_StdPeriph_Lib_V1.0.0 a jejich příkladů [54]. Nejdříve je nutné konfigurovat RTC (výběr zdroje RTC čítače, povolení taktování

rozhraní RTC, atd.). Nastavení aktuálního času je realizováno vždy po restartu systému. K zjištění aktuálního času se využívá jeden z podporovaných protokolů LwIP, konkrétně protokol SNTP (kapitola 4.3.4.4). Po zjištění aktuálního času pomocí SNTP protokolu z veřejného SNTP serveru jsou tyto hodnoty (hodiny, minuty, sekundy, měsíce, dny a roky) zapsány do registru RTC čítače. RTC čítač se poté již stará o inkrementování každou jednu sekundu a udržuje právě tak aktuální čas.

5 Návrh firmwaru

Jak již bylo řečeno, dodaná základnová stanice se skládá z desky a mikrokontroléru STM32F207VGT6. Pro komunikaci s mikrokontrolérem je využito rozhraní SWD, zdrojový kód je napsán v jazyce C. Jako vývojové prostředí byl zvolen Keil MDK verze 5.

Návrh firmwaru pro základnovou stanici domácího asistivního systému vychází z dostupné literatury (kapitola 2), z které byly zjištěny běžné požadavky na domácí asistivní systém. Po návrhu byl firmware implementován. Navržený firmware se má stát pouze základem pro plnohodnotný domácí asistivní systém. Tento systém by měl především být využit pro přijímání dat ze senzorů, které jsou v asistenční péči využívány (kapitola 2). Systém by dále měl podporovat odesílání dat ze senzorů na dohledový server, kde by měla být přijatá data monitorována a automaticky vyhodnocována na základě odchylek od běžných zvyklostí uživatele.

Navržený firmware pro základnovou stanici domácího asistivního systému je následující: Po zapnutí základnové stanice jsou volány funkce pro její inicializaci a inicializaci používaných knihoven. Poté jsou pomocí operačního systému FreeRTOS spuštěny úlohy aplikace (kapitola 4.2). Nejprve je spuštěna úloha pro konfiguraci parametrů zařízení pomocí DHCP serveru (kapitola 5.3). Pokud tyto parametry nejsou z DHCP serveru získány, tak jsou použity statické parametry, které byly nastaveny po inicializaci stanice. Dále jsou nastaveny hodiny reálného času (RTC) pomocí SNTP protokolu (kapitola 5.4). Následně je spuštěno sledování senzorů (kapitola 5.5) a posílání live packetu na dohledový server Swiftshield každou hodinu (kapitola 5.6).

5.1 Inicializace základnové stanice

Hlavní soubor firmwaru pro základnovou stanici domácího asistivního systému je `main.c`. Po zapnutí či restartování základnové stanice jsou nejdříve volány funkce ze souboru `main.c`, protože nejdříve musí dojít k inicializaci stanice. Inicializací stanice je myšlena inicializace použitých knihoven a periférií (podrobně kapitola 4). Pro debugovací účely je inicializovaný USART. Dále jsou inicializované LED diody, které se na základnové stanici nacházejí, one-wire bus a časovač pro funkce `delay_us()` a `delay_ms()`. Pro použití LwIP stacku je nejprve nutné inicializovat Ethernet rozhraní a až poté

inicializovat LwIP stack, z kterého jsou použity některé protokoly. Povoleny musí být i přerušení pro periférie. Pro přijímání dat do RF modulu je inicializováno rozhraní SPI a RF modul.

5.2 Nastavení statické IP adresy, masky sítě a IP adresy brány

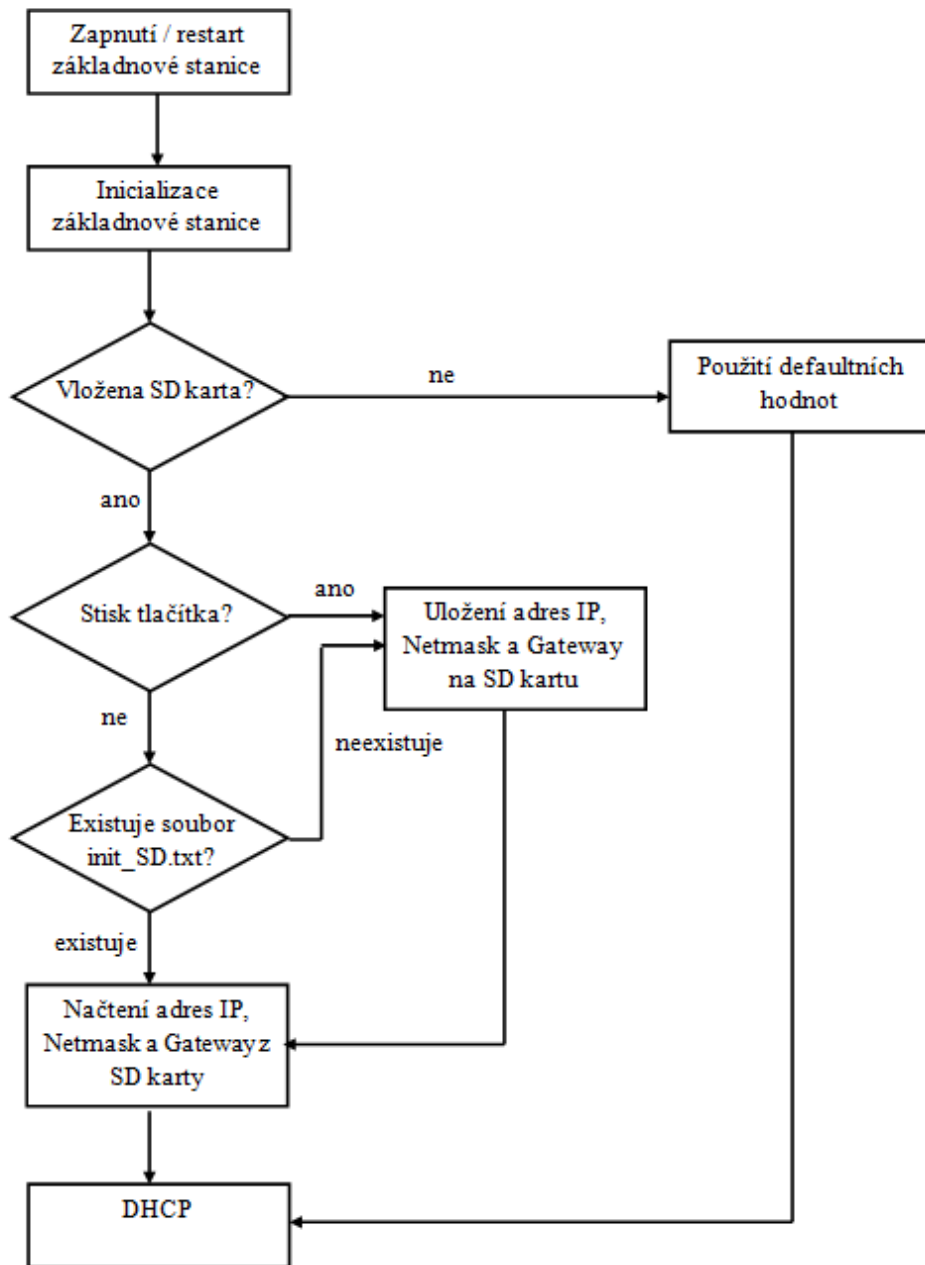
V případě, že se základnová stanice po zapnutí a inicializaci nepřipojí k DHCP serveru (kapitola 5.3) automaticky (např. na síti není DHCP server), cca do jedné minuty, je použita statická IP adresa, maska sítě a IP adresa brány. Po inicializaci základnové stanice je spuštěna cca 9 vteřinová čekací smyčka. V této smyčce se čeká, zda bude stisknuto tlačítko. Pokud je stisknuto tlačítko jsou výchozí hodnoty IP adresa, maska sítě a IP adresa brány uloženy do souboru init_SD.txt na SD kartě, pomocí knihovny FatFs, pokud tento soubor již existuje tak je přepsán. Výchozí hodnota IP adresy je 192.168.0.10, masky sítě 255.255.255.0 a IP adresa brány je 192.168.0.1.

Pokud tlačítko není stisknuto, jsou tyto statické hodnoty načteny ze souboru init_SD.txt a výchozí hodnoty jsou nahrazeny hodnotami ze souboru. V tomto případě je, ale nutné zkontrolovat existenci souboru init_SD.txt na SD kartě. Pokud soubor neexistuje, bude se aplikace chovat, jako kdyby bylo stisknuto tlačítko. Tedy budou do souboru init_SD.txt uloženy výchozí hodnoty. Celý scénář nastavení statické IP adresy, masky sítě a IP adresy výchozí brány je zobrazen na obr. 5.2.

Pokud není vložena SD karta, je tato skutečnost oznámena prostřednictvím výpisu v Hyperterminálu (obr. 5.1). Použity jsou výchozí hodnoty, které jsou uloženy v proměnných souboru netconf.c. Jedná se o stejné hodnoty, které jsou ukládány do souboru init_SD.txt v případě přítomnosti SD karty.

```
thernet Init failed
PHY MICR initialized...
PHY MISR initialized...
CHYBA: Neni vlozena SD karta.
Defaultni hodnoty:
IP adress: 192.168.0.10
Netmask: 255.255.255.0
Gateway: 192.168.0.1
Inicializace OK
Looking for DHCP server ...
DHCP timeout, static IP address: 192.168.0.10
Static Netmask: 255.255.255.0
Static Gateway: 192.168.0.1
```

Obr. 5.1 Výpis pokud není vložena SD karta



Obr. 5.2 Nastavení statické IP adresy, masky sítě a IP adresy brány

Využití výše zmíněného se předpokládá ve startovacím menu, kdy by uživatel mohl nastavit výchozí statické hodnoty IP adresy, masky sítě a IP adresy brány. Toto menu není v této práci implementováno.

5.3 Konfigurace parametrů zařízení z DHCP serveru

Knihovna LwIP podporuje protokol DHCP, který je využit při konfiguraci parametrů zařízení. Konfigurované parametry zařízení jsou IP adresa zařízení, maska sítě a IP adresa výchozí brány. Tento scénář je spuštěn jako úloha pomocí operačního systému FreeRTOS. Pokud základnová stanice najde na síti DHCP tak jsou parametry nakonfigurované dle DHCP serveru (obr. 5.3).

```
thernet Init OK
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
IP assigned by DHCP:      10.0.0.37
Netmask assigned by DHCP: 255.255.255.0
Gateway assigned by DHCP: 10.0.0.138
```

Obr. 5.3 Výpis při konfiguraci z DHCP serveru

V případě, že stanice nenalezne DHCP server, tak jsou po určité době (cca po 1 minutě), přiřazeny zařízení statická IP adresa, maska sítě a IP adresa výchozí brány (kapitola 5.3) (obr. 5.4). Úloha je ukončena po nakonfigurování parametrů dle DHCP serveru nebo po přiřazení statických adres.

```
thernet Init failed
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
DHCP timeout, static IP address: 192.168.0.10
Static Netmask: 255.255.255.0
Static Gateway: 192.168.0.1
```

Obr. 5.4 Výpis při přiřazení statických parametrů

5.4 Nastavení hodin reálného času pomocí SNTP protokolu

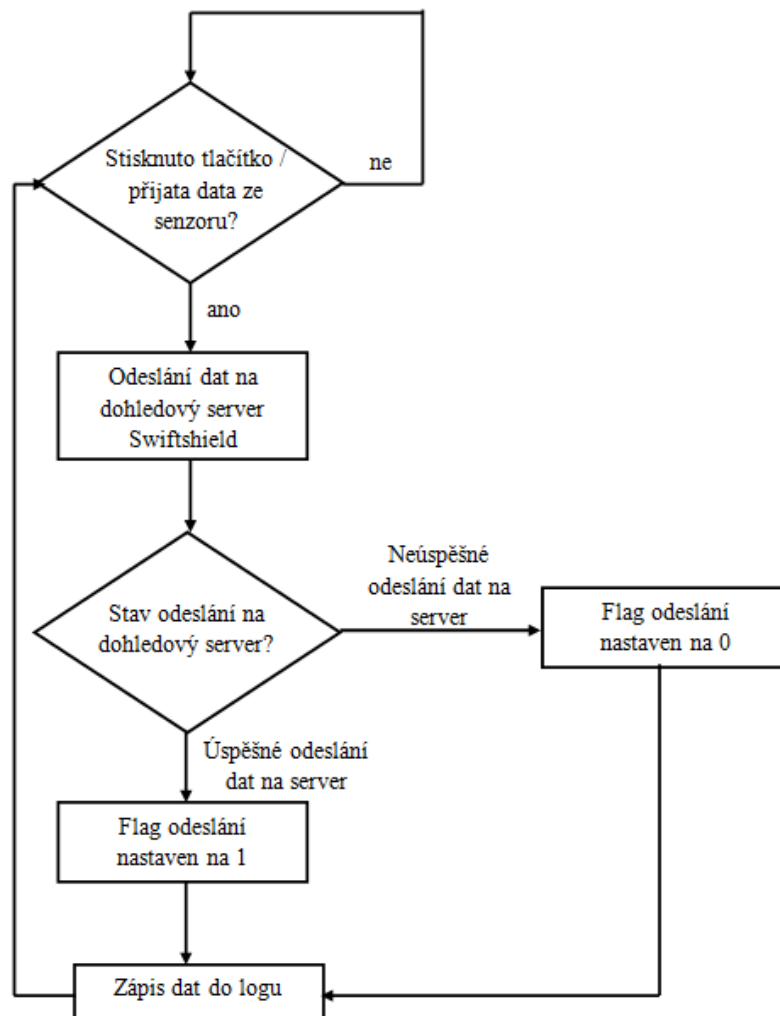
Po ukončení úlohy konfigurace parametrů zařízení z DHCP serveru je spuštěna úloha, která nastaví hodiny reálného času (RTC) pomocí SNTP protokolu, který poskytuje LwIP stack. Pomocí SNTP protokolu je nejdříve zjištěn aktuální čas z veřejného SNTP serveru (obr. 5.5). Tento čas je poté zapsán do registru čítače RTC a úloha je ukončena.

```
therneth Init OK
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
IP assigned by DHCP: 10.0.0.37
Netmask assigned by DHCP: 255.255.255.0
Gateway assigned by DHCP: 10.0.0.138
Aktualni cas: [2014-04-21 13:35:10]
```

Obr. 5.5 Výpis s aktuálním časem ze serveru

5.5 Sledování senzorů

Sledování dat a událostí z čidel je stěžejní úkol této aplikace (obr. 5.6). Při návrhu byl k dispozici pouze jeden senzor s akcelerometrem, jehož využití v asistenční péči je popsán v kapitole 2. Za druhý senzor lze považovat tlačítko, které je připojeno přímo k základnové stanici. Toto tlačítko může například reprezentovat tlačítko pro přivolání pomoci v případě potřeby (kapitola 2). Způsob přijetí dat z těchto dvou senzorů se liší. Získaná data z obou senzorů jsou avšak zapisována podobným způsobem do logu, který je umístěn na SD kartě a zároveň dochází k posílání dat na dohledový server Swiftshield.



Obr. 5.6 Sledování senzorů a zápis do logu

5.5.1 Používané senzory

Senzor s akcelerometrem byl dodán vedoucím práce i se softwarem, takže implementace tohoto softwaru není součástí této práce. Pro potřeby práce bylo pouze nutné se seznámit s daty, která vysílá senzor pomocí RF modulu a přijímá je základnová stanice opět pomocí RF modulu. Přijímání dat ze senzoru je realizováno pomocí úlohy operačního systému FreeRTOS.

Při pohybu senzor posílá tři pakety (obr. 5.7), které jsou ovšem pouze demonstrační, protože senzor neodesílá informace o naklonění, atd. V budoucnu je však možné využít plnou funkčnost akcelerometru. První dva pakety mají zcela stejný formát a to z důvodu, kdyby první paket nebyl přijat základnovou stanicí, nebo lze použít druhý paket jako kontrolu, zda je paket korektní. Dále jsou tedy zpracovávána pouze data z prvního paketu.

Třetí paket je pouze kontrolní a poskytuje informace, že senzor “usíná“ a není nijak dál zpracováván.

```
<>ACC01 INTERRUPT    BAT OK CRC>  
<>ACC01 INTERRUPT    BAT OK CRC>  
Entering sleep mode ...
```

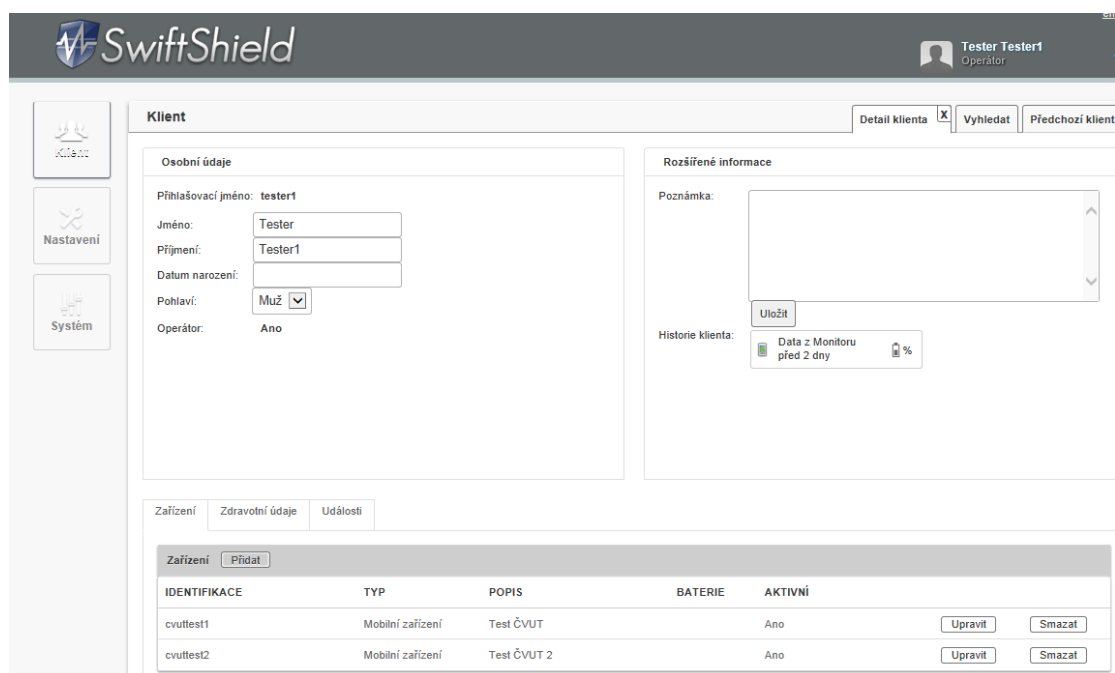
Obr. 5.7 Výpis paketů ze senzoru

Jak již bylo řečeno, tlačítko je přímo připojené k základnové stanici. Při jeho stisku pouze dojde k události přerušení a poté k dalšímu zpracování této události.

5.5.2 Posílání dat na dohledový server Swiftshield

Po přijmutí dat ze senzoru či po stisku tlačítka jsou data, pomocí TCP klienta z LwIP stacku, odeslána na dohledový server Swiftshield [55]. Toto odesílání má pouze demonstrační a kontrolní charakter. Swiftshield je služba, která slouží pro automatizované a nepřetržité monitorování bezpečnosti a zdravotního stavu osob prostřednictvím mobilních monitorovacích a dohledových aplikací, digitálního senzoru či zdravotních přístrojů. Po přihlášení do systému je možné vytvořit nového klienta a přidat monitorovací zařízení. Swiftshield poskytuje pět typů monitorovacích zařízení: Mobilní zařízení, Váha Withings, Dálkové ovládání mobilního telefonu, Dálkové ovládání Care a Care mobilní zařízení. Tato práce využívá typ Mobilní zařízení s identifikací cvuttest1 (obr. 5.8).

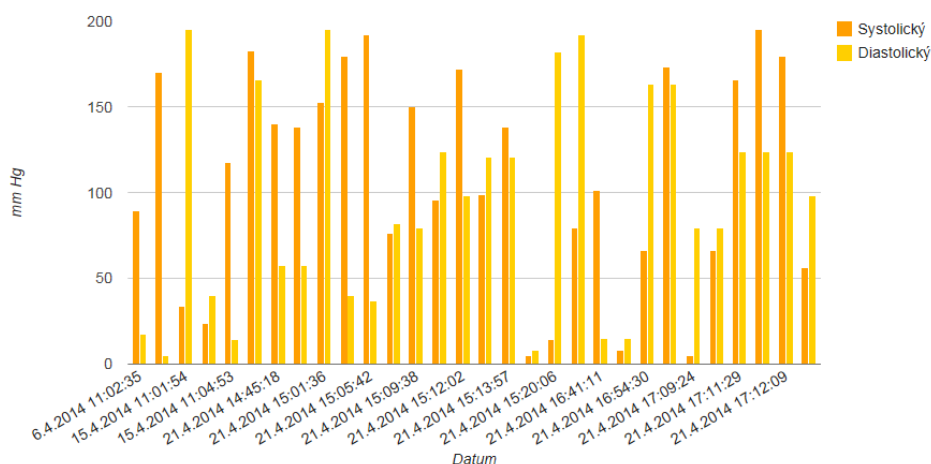
Server poskytuje monitorování několika typů událostí: Přihlášení do systému, Přidělení úkolu, Zobrazení v systému, Nové hodnoty měření Váhy, Nově hodnoty měření tuku, Nová hodnota pozice, Hlášení stavu nouze, Hlášení stavu neaktivity, Hlášen stav aktivity, Nabíjení mobilního telefonu, Nabíjení mobilního telefonu ukončeno, Nové hodnoty měření tlaku, Nové hodnoty měření pulsu, Přidělení úkolu, Nový úkol, Zavření úkolu, Zobrazení klienta, Příchozí hovor, Doma, Mimo domov, Mobilní aplikace je spuštěná, Úkol uživatele přidělen, Úkol uživatele vyřešen, Úkol uživatele uzavřen, Úkol uživatele odložen a Odložení úkolu.



Obr. 5.8 Rozhraní dohledového serveru Swiftshield

Odeslání dat na server ze základnové stanice se projeví, jako dvě události. První událost je “Mobilní aplikace je spuštěná“, která bezprostředně přechází události “Nové hodnoty měření tlaku“. Při druhé operaci server přijímá data od základnové stanice. Tyto data odesílaná na server jsou hodnoty krevního tlaku (systola a diastola). Odesílané hodnoty ovšem nejsou reálné a jedná se pouze o dvě náhodná čísla v rozsahu 0 až 200, které mají jen demonstrační charakter. Tyto hodnoty lze na serveru zobrazit (obr. 5.9).

Po úspěšném odeslání těchto dat na server nastavuje aplikace flag na hodnotu 1, pokud nebylo odeslání úspěšné je flag nastaven na 0. Tato hodnota je dále zpracovávána při zápisu do logu.



Obr. 5.9 Zobrazení systoly a diastoly na serveru Swiftshield

5.5.3 Zapsání dat do logu

Po odeslání dat na server Swiftshield jsou do souboru `senzor.log` na SD kartě, pomocí knihovny `FatFs`, zapsána data, které přijala základnová stanice od senzoru nebo jsou zapsána statické data, která jsou vytvořena vždy po stisku tlačítka. Struktura zapisovaných dat je zobrazena na obr. 5.10. Na prvním řádku je zobrazena struktura dat, která jsou vytvořena a zapsána do logu vždy po stisku tlačítka. Druhý řádek zobrazuje strukturu dat, která jsou přijata ze senzoru.

[2014-04-24 16:39:57]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-24 16:40:10]	00	ACC01 INTERRUPT	255	1	CRC	1

Obr. 5.10 Struktura dat zapisovaných do logu

Data zapisovaná do logu mají vždy stejnou strukturu (obr. 5.11). Nejdříve je do logu zapsán čas, kdy došlo ke zmáčknutí tlačítka či přijetí dat ze senzoru. Jedná se o aktuální čas z hodin reálného času (RTC). Další položka na řádku je dvojmístné číslo, které reprezentuje typ senzoru (00 senzor s akcelerometrem a 99 tlačítko). Následují vlastní data senzoru. Senzor s akcelerometrem posílá pakety s textem, který je při zápisu do logu částečně použit. Při stisku tlačítka jsou vygenerována vždy stejná data, která jsou zapsána do logu. Trojmístné číslo, které následuje po vlastních datech senzoru, má reprezentovat unikátní identifikační číslo zařízení. Tato hodnota je nastavena v obou případech defaultně na 255. V budoucnu by tato hodnota měla například rozlišit dva senzory s akcelerometrem, protože se jedná o stejný typ senzoru. Jednomístné číslo po unikátním identifikačním čísle má reprezentovat stav baterie daného senzoru. Pokud je hodnota 1 je baterie nabitá, při stavu 0 je baterie vybitá. V případě tlačítka je hodnota defaultně nastavena na 1. Senzor s akcelerometrem posílá informace o stavu baterie (obr. 5.7). Informace o stavu baterie senzoru je defaultně nastavena a tudíž není reálná. Tento stav je i přes jeho defaultní nastavení zkontrolován a výsledek je zapsán do logu. Následuje kontrolní součet, kdy zapsaná hodnota při stisku tlačítka je opět defaultní. Kontrolní součet zapsaný do logu při přijetí dat ze senzoru je přebrán z textu přijatých dat, který má reprezentovat kontrolní součet. Poslední hodnota na řádku, která je zapisována do logu, je flag úspěšného či neúspěšného odeslání dat na server Swiftshield.

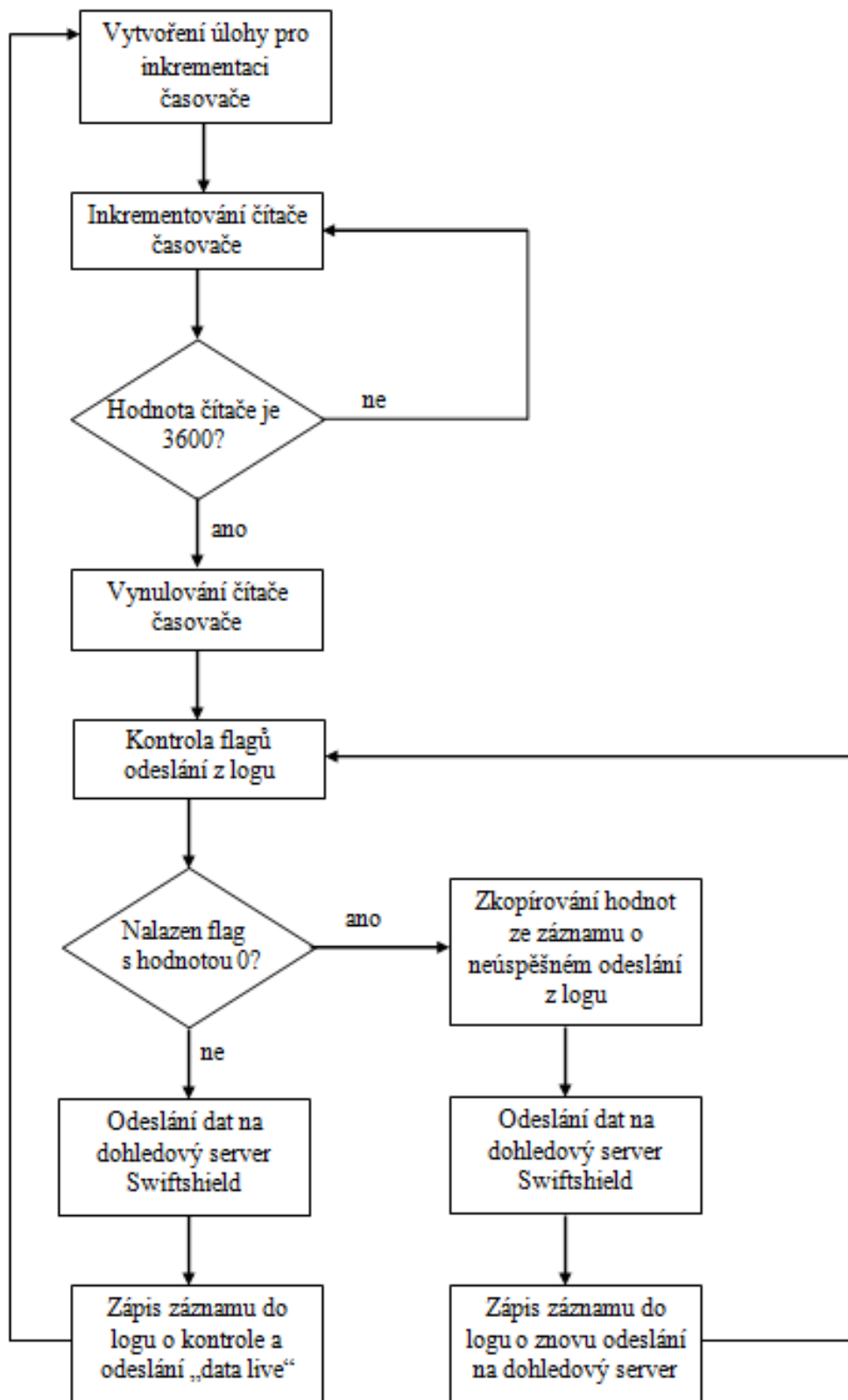


Obr. 5.11 Obecná struktura dat zapsaných do logu

5.6 Posílání live packetu každou hodinu

Přijímání dat ze senzorů či posílání dat na dohledový server Swiftshield nemusí být vždy plně funkční (například z důvodu výpadku internetového spojení u uživatele), proto je nutné zařadit do aplikace i kontrolu těchto činností. Při návrhu firmwaru byla tato kontrola realizována pomocí tzv. live packetu (obr. 5.12). Pod tímto pojmem je myšleno posílání dat na dohledový server každou hodinu a zároveň této činnosti předchází kontrola, zda byla odeslána všechna data od předchozí kontroly či od zapnutí systému.

Základem tohoto scénáře je spuštění úlohy pomocí operačního systému FreeRTOS, která inkrementuje čítač času po vteřině. Tato úloha je spuštěna po inicializaci základnové stanice. Pokud hodnota čítače odpovídá hodnotě 3600 (1 hodina) je spuštěna kontrola, zda byla odeslána všechna data na dohledový server. Kontrola odeslaných dat je realizována pomocí souboru `senzor.log`.



Obr. 5.12 Kontrola a posílání live packetu na dohledový server

5.6.1 Kontrola odeslání dat na dohledový server a zápis dat do logu

Jak již bylo zmíněno v kapitole 5.5.2 je po každém odeslání dat na dohledový server nastaven flag, zda bylo odeslání úspěšné a hodnota flagu je zapsána i do logu. Během kontroly jsou postupně tyto flagy v logu kontrolovány. Pokud je nalezen flag, který indikuje, že nedošlo k úspěšnému odeslání dat na server, jsou odeslána data na dohledový server a do logu je zapsán záznam o chybě.

Před zápisem do logu jsou poslána data na dohledový server Swiftshield. Opět jsou posílána pouze dvě náhodně generována čísla (systola a diastola), které jsou na serveru zaznamenávány jako hodnoty krevního tlaku. Při tomto odeslání je opět nastavován flag, zda došlo k úspěšnému odeslání dat na server.

Po odeslání na dohledový server Swiftshield je zapsán záznam do souboru `init_SD.txt`. Tento záznam v logu má podobnou strukturu jako záznam při přijetí dat ze senzoru či při stisku tlačítka. Na prvním místě na řádce je čas, kdy došlo ke kontrole a k novému poslání dat na dohledový server. Poté následuje typ senzoru, který je zkopírován ze záznamu, kdy došlo k neúspěšnému odeslání dat na server. Následují vlastní data záznamu, která začínají znaky „ER“ a poté časem, kdy došlo k neúspěšnému odeslání dat na server. Hodnoty unikátního identifikačního čísla, stavu baterie a kontrolního součtu jsou opět převzaty ze záznamu, kdy došlo k neúspěšnému odeslání na dohledový server. Poslední hodnota na řádce je flag úspěšného či neúspěšného odeslání.

Pokud již byly zpracovány všechny záznamy o neúspěšném odeslání či nebyl nalezen žádný zájem o neúspěšném odeslání je posílán live packet. Nejprve jsou opět odeslána data (systola a diastola) na dohledový server Swiftshield a je nastaven flag o úspěšném či neúspěšném odeslání. Po odeslání dat na dohledový server je do logu zapsán záznam o kontrole a o odeslání live packetu na dohledový server. Tento záznam má téměř stejnou strukturu jako data zapisována při stisku tlačítka, zápis se liší pouze v sekci data (obr. 5.11). V sekci data jsou místo znaků „ab“ umístěny znaky „<>“. Tyto znaky značí, že byla provedena kontrola záznamů předešlých záznamů a že byl odeslán live packet na dohledový server.

5.6.2 Příklad kontroly a zápisu do logu

Na obr. 5.13 je zobrazen stav logu po kontrole a odeslání live packetu. Jak je z obrázku patrné, tak na prvním a čtvrtém řádku je flag nastaven na hodnotu 0. Z výše zmíněného vyplývá, že nedošlo k úspěšnému odeslání dat na dohledový server. Dojde tedy dvakrát k novému odeslání dat na dohledový server a zapsání do logu, protože dvakrát nebyla data odeslána úspěšně. Po této kontrole jsou odeslána data nazývaná live packet na dohledový server. Po odeslání těchto dat je zapsán do logu záznam o kontrole a odeslání dat. Tento záznam se nachází na posledním řádku obr. 5.13.

[2014-04-25 13:18:59]	99	abcdefghijklmnopqrstu	255	1	255	0
[2014-04-25 13:22:15]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-25 13:29:57]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-25 13:33:10]	00	ACC01 INTERRUPT	255	1	CRC	0
[2014-04-25 13:35:39]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-25 14:04:36]	99	ER2014-04-25 13:18:59	255	1	255	1
[2014-04-25 14:04:41]	00	ER2014-04-25 13:33:10	255	1	CRC	1
[2014-04-25 14:04:42]	99	<>abcdefghijklmnopqrstu	255	1	255	1

Obr. 5.13 Struktura dat při neúspěšném odeslání dat na dohledový server a záznam o kontrole s odesláním live packetu na dohledový server

6 Testování

Na základnové stanici byla provedena řada simulací, které dokázali funkční nastavení jednotlivých periférií a navrženého firmwaru. Pro testování byly navrženy tři scénáře testování. První scénář (kapitola 6.1) testuje načtení statických parametrů ze souboru na SD kartě a uložení statických parametrů do souboru na SD kartě. Druhý scénář (kapitola 6.2) testuje přijmutí dat ze senzoru s akcelerometrem. Dále testuje odeslání dat na dohledový server Swiftshield. Testováno je i uložení záznamu do logu na SD kartě po přijmutí dat ze senzoru či po stisku tlačítka. Třetí scénář (kapitola 6.3) testuje kontrolu logu na SD kartě a následné posílání live packetu na dohledový server Swiftshield.

6.1 Testování statických parametrů

První scénář ověřuje funkčnost načtení a uložení statické IP adresy, masky sítě a IP adresy brány ze souboru `SD_init.log`, který je umístěn na SD kartě (podrobně kapitola 5.2). Dále se scénář zabývá přiřazením těchto statických parametrů v případě, že se základnová stanice po inicializaci nepřipojí k DHCP serveru automaticky (podrobně kapitola 5.3).

Tento scénář je rozdělen na tři části. První část scénáře se zabývá testováním uložení statických parametrů do souboru `SD_init.txt` na SD kartě (kapitola 6.1.1). Druhá část se zabývá testováním načtení dat ze souboru v případě, že soubor `SD_init.txt` neexistuje (kapitola 6.1.2). Poslední část se zabývá testováním načtení hodnot ze souboru, když soubor `SD_init.txt` existuje (kapitola 6.1.3).

6.1.1 Testování uložení statických parametrů do souboru

Scénář testování této části je následující: V čekací smyčce (cca 9 vteřin), po inicializaci základnové stanice, bude stisknuto tlačítko. Po stisku tlačítka budou do souboru `init_SD.txt` uloženy statické parametry. Tyto parametry jsou IP adresa zařízení (192.168.0.10), maska sítě (255.255.255.0) a IP adresa brány (192.168.0.1). Tyto statické parametry jsou převzaty ze souboru firmwaru `netconf.c`, kde jsou uloženy v proměnných. Po uplynutí času (cca 1 minuta), nebude nalezen DHCP server, protože nebude

k základnové stanici připojen ethernetový kabel, proto budou načteny statické parametry, které byly uloženy v souboru init_SD.txt.

```
ethernet Init failed
PHY MICR initialized...
PHY MISR initialized..
Stisk tlačítka
uložení na SD kartu:
IP address: 192.168.0.10
Netmask: 255.255.255.0
Gateway: 192.168.0.1
Inicializace OK
Looking for DHCP server ...
DHCP timeout, static IP address: 192.168.0.10
Static Netmask: 255.255.255.0
Static Gateway: 192.168.0.1
```

Obr. 6.1 Výpis při uložení a přiřazení statických parametrů

Výsledek simulace tohoto scénáře je zobrazen na obr. 6.1. Nejprve byl zobrazen výpis, že není inicializovaný Ethernet, protože k základnové stanici nebyl připojen ethernetový kabel. Poté bylo v čekací smyčce (cca 9 vteřin) stisknuto tlačítko. Tento stisk indikuje uložení statických parametrů (IP adresa, maska sítě a IP adresa výchozí brány) do souboru na SD kartě. Následoval výpis uložených hodnot na SD kartě. Poté začala čekací smyčka (cca 1 minuta) během, které nebyl nalezen DHCP server. Následoval výpis o vypršení čekací smyčky a byly přiřazeny statické parametry, které byly načteny ze souboru na SD kartě a jejich hodnoty byly vypsány.

6.1.2 Testování načtení statických parametrů z neexistujícího souboru

Scénář testování druhé části je následující: V čekací smyčce (cca 9 vteřin), po inicializaci základnové stanice, nebude stisknuto tlačítko. Poté bude zjištěno, zda existuje soubor init_SD.txt na SD kartě. Tento soubor nebude existovat, proto budou do souboru init_SD.txt uloženy statické parametry. Tyto parametry jsou IP adresa zařízení (192.168.0.10), maska sítě (255.255.255.0) a IP adresa brány (192.168.0.1). Po uplynutí času (cca 1 minuta), nebude nalezen DHCP server, protože nebude k základnové stanici připojen ethernetový kabel, proto budou načteny statické parametry, které byly uloženy v souboru init_SD.txt.

```
thernet Init failed
PHY MICR initialized...
PHY MISR initialized...
ulozeni na SD kartu:
IP adress: 192.168.0.10
Netmask: 255.255.255.0
Gateway: 192.168.0.1
Inicializace OK
Looking for DHCP server ...
DHCP timeout, static IP address: 192.168.0.10
Static Netmask: 255.255.255.0
Static Gateway: 192.168.0.1
```

Obr. 6.2 Výpis při načítání statických parametrů z neexistujícího souboru

Na obr. 6.2 je zobrazen výsledek simulace tohoto scénáře. Nejdříve bylo vypsáno, že není inicializovaný Ethernet, protože k základnové stanici nebyl připojen ethernetový kabel. Poté nebylo v čekací smyčce (cca 9 vteřin) stisknuto tlačítko, proto došlo ke kontrole existence souboru `init_SD.txt` na SD kartě. Jelikož tento soubor neexistoval, choval se firmware, jako kdyby bylo stisknuto tlačítko. Došlo tedy k vytvoření souboru na SD kartě a k uložení statických parametrů (IP adresa, maska sítě a IP adresa výchozí brány) do vytvořeného souboru. Následoval výpis uložených hodnot na SD kartě. Poté začala čekací smyčka (cca 1 minuta) během, které nebyl nalezen DHCP server. Následoval výpis o vypršení čekací smyčky a byly přiřazeny statické parametry, které byly načteny ze souboru na SD kartě a jejich hodnoty byly vypsány.

6.1.3 Testování načtení statických parametrů z existujícího souboru

Scénář testování třetí části je následující: V čekací smyčce (cca 9 vteřin), po inicializaci základnové stanice, nebude stisknuto tlačítko. Poté bude zjištěno, zda existuje soubor `init_SD.txt` na SD kartě. Tento soubor bude existovat, proto budou ze souboru `init_SD.txt` načteny statické parametry. Statický parametr IP adresa zařízení bude nastaven v souboru `init_SD.txt` na hodnotu 172.188.0.10, parametry maska sítě (255.255.255.0) a IP adresa výchozí brány (192.168.0.1) budou stejné, jako implicitní hodnoty. Po uplynutí času (cca 1 minuta), nebude nalezen DHCP server, protože nebude k základnové stanici připojen ethernetový kabel, proto budou načteny statické parametry, které jsou uloženy v souboru `init_SD.txt`.

```
ethernet Init failed
PHY MICR initialized...
PHY MISR initialized...
nacteni z SD karty:
IP adress: 172.188.0.10
Netmask: 255.255.255.0
Gateway: 192.168.0.1
Inicializace OK
Looking for DHCP server ...
DHCP timeout, static IP address: 172.188.0.10
Static Netmask: 255.255.255.0
Static Gateway: 192.168.0.1
```

Obr. 6.3 Výpis při načítání statických parametrů z existujícího souboru

Na obr. 6.3 je zobrazen výsledek simulace tohoto scénáře. První byl výpis, že není inicializovaný Ethernet, protože k základnové stanici nebyl připojen ethernetový kabel. Dále nebylo v čekací smyčce (cca 9 vteřin) stisknuto tlačítko, proto došlo ke kontrole existence souboru `init_SD.txt` na SD kartě. Tento soubor existoval, a proto byly načteny hodnoty, které byly uloženy v souboru. Tyto hodnoty byly vypsány, jak je vidět IP adresa odpovídala hodnotě 172.188.0.10. Dále byly vypsány hodnoty maska sítě a IP adresy výchozí brány, které byly stejné jako implicitní hodnoty. Poté začala čekací smyčka (cca 1 minuta) během, které nebyl nalezen DHCP server. Následoval výpis o vypršení čekací smyčky a byly přiřazeny statické parametry, které byly načteny ze souboru na SD kartě a jejich hodnoty jsou vypsány.

6.2 Testování přijímání dat ze senzorů a odesílání dat na dohledový server Swiftshield

Druhý scénář testování ověřuje funkčnost přijímání dat ze senzoru s akcelerometrem základnovou stanicí pomocí RF modulu (podrobně kapitola 5.5.1). Dále se scénář zabývá odesíláním dat na dohledový server po stisku tlačítka či po přijmutí dat ze senzoru s akcelerometrem (podrobně kapitola 5.5.2). Součástí tohoto testování je i zápis záznamu do souboru `senzor.log` po odeslání dat na dohledový server, kterému předchází stisk tlačítka či přijmutí dat ze senzoru (podrobně kapitola 5.5.3). Ověřena je i funkčnost kontroly logu před odesláním live packetu na dohledový server Swiftshield.

Scénář tohoto testování je následující: Po inicializaci základnové stanice, načtení statických hodnot IP adresy zařízení, masky sítě a IP adresy výchozí brány ze souboru `init_SD.txt`, konfigurování parametrů z DHCP serveru a nastavení hodin reálného času (RTC) z veřejného SNTP serveru bude dvakrát stisknuto tlačítko, poté budou dvakrát přijata data ze senzoru s akcelerometrem, opět bude stisknuto tlačítko a nakonec budou jednou přijata data ze senzoru s akcelerometrem. Po každém stisku tlačítka či přijmutí dat ze senzoru s akcelerometrem budou odeslána data na dohledový server Swiftshield a poté budou zapsána data do souboru `senzor.log`, který je umístěn na SD kartě. Jako poslední bude provedena kontrola logu, zda byla všechna data odeslána na dohledový server, poté budou poslána data na dohledový server, která budou reprezentovat live packet a do logu bude uložen záznam o odeslání live packetu. Pro účely testování byla upravena doba kontroly logu a posílání live packetu z 1 hodiny na 10 minut.

Pro vyhodnocení testování bude porovnán čas v souboru `senzor.log` se záznamem událostí na dohledovém serveru Swiftshield. Dále budou porovnány odesílané hodnoty krevního tlaku (systola a diastola) s hodnotami zaznamenanými na dohledovém serveru.

```
thernet Init OK
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
IP assigned by DHCP:      10.0.0.139
Netmask assigned by DHCP: 255.255.255.0
Gateway assigned by DHCP: 10.0.0.138
Aktualni cas: [2014-04-27 15:35:01]
```

Obr. 6.4 Výpis inicializace a nastavení RTC při simulaci druhého scénáře testování

Začátek simulace tohoto scénáře je zobrazen na obr. 6.4. Nejdříve byly načteny statické hodnoty IP adresa zařízení, maska sítě a IP adresa výchozí brány ze souboru `init_SD.txt` (výpis byl vypnut), protože nebylo v čekací smyčce stisknuto tlačítko. Dále je z obrázku patrné, že nejdříve proběhla inicializace Ethernetu, která je potřebná pro inicializaci LwIP stacku, který podporuje používané protokoly DHCP a SNTP. Poté byly pomocí DHCP nakonfigurovány parametry IP adresa zařízení, maska sítě a IP adresa výchozí brány. Po konfiguraci parametrů byly nastaveny hodiny reálného času (RTC) z veřejného SNTP serveru.

Po inicializaci základnové stanice, načtení statických parametrů ze souboru, nakonfigurování parametrů pomocí DHCP serveru a nastavení RTC z veřejného SNTP

serveru bylo stisknuto dvakrát tlačítko, dvakrát byly přijaty data ze senzoru s akcelerometrem, opět bylo stisknuto tlačítko a nakonec byla jednou přijata data ze senzoru. Poté byla po deseti minutách provedena automatická kontrola logu, zda byla všechna data odeslána a následovalo odeslání dat na dohledový server se záznamem do logu o odeslání live packetu na dohledový server. Výsledek této části simulace je zobrazen na obr. 6.5.

[2014-04-27 15:35:18]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 15:35:57]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 15:36:35]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-27 15:37:14]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-27 15:37:51]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 15:38:18]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-27 15:44:46]	99	<>cdefghijklmnopqrstu	255	1	255	1

Obr. 6.5 Výpis logu po simulaci druhého scénáře testování

Z obr. 6.5 je patrné, že na začátku bylo dvakrát stisknuto tlačítko, podle hodnoty typ senzoru nastavené na hodnotu 99 (viz. obr. 5.11). Poté byla podvčetně přijata data ze senzoru s akcelerometrem, protože hodnota typ senzoru je nastavena na hodnotu 00. Z pátého řádku logu je patrné, že bylo opět stisknuto tlačítko a že poté byla přijata opět data ze senzoru. Na posledním řádku logu je záznam o odeslání live packetu na dohledový server, protože text v sekci data (viz. obr. 5.11) je začíná znaky “<>”.

Jak je patrné, hodiny reálného času byly nastaveny v 15:35:01, ale k odeslání live packetu došlo již po 9 minutách a 45 vteřinách, místo pro simulaci nastavených 10 minutách. Tento fakt je způsoben tím, že úloha operačního systému FreeRTOS, která inkrementuje čítač času po vteřině, je spuštěna ve stejné chvíli jako úloha pro konfiguraci parametrů pomocí DHCP serveru. Tato úloha je tedy spuštěna ihned po inicializaci stanice a nastavení hodin reálného času (RTC) a proběhne až po konfiguraci parametrů DHCP serverem.

Po provedení simulace byly nasnímány i data z dohledového serveru Swiftshield. Na obr. 6.6 je seznam událostí, které dohledový server Swiftshield zaznamenal po simulaci tohoto scénáře testování. Jak již bylo řečeno, odeslání dat na server ze základnové stanice se projeví jako dvě události: “Mobilní aplikace je spuštěna“ a “Nové hodnoty měření tlaku“. Z logu je patrné, že vždy došlo k odeslání dat na dohledový server, protože

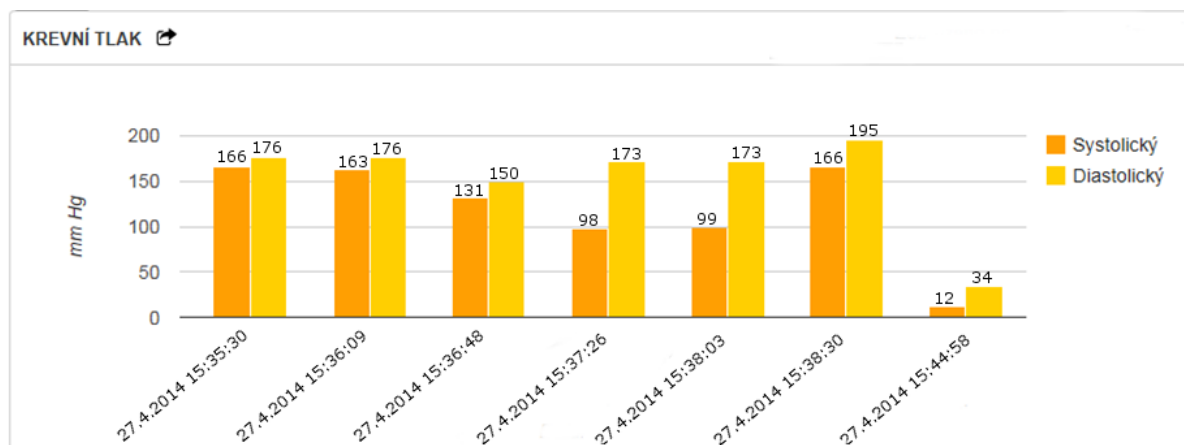
všechny flagy odeslání jsou nastaveny na hodnotu 1 (viz. obr. 5.11) a nedošlo tedy k žádnému neúspěšnému odeslání dat na dohledový server. Proto je správně zaznamenáno na dohledovém serveru čtrnáct událostí, protože bylo provedeno celkem sedm odeslání na dohledový server, které je vždy správně zaznamenáno jako dvě události.

Při porovnání času v logu a času, kdy byla událost zaznamenána na dohledovém serveru, je téměř vždy patrný dvanácti vteřinový rozdíl v těchto časech. Tento fakt je způsoben tím, že v logu je vždy zaznamenán čas, kdy došlo ke stisku tlačítka nebo přijmutí dat ze senzoru a ne čas, kdy došlo k odeslání dat na dohledový server Swiftshield.

DATUM	UDÁLOST	ZDROJ
27.4.2014 15:44:58	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:44:58	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:38:30	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:38:30	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:38:03	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:38:03	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:37:26	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:37:26	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:36:48	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:36:48	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:36:09	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:36:09	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 15:35:30	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 15:35:30	Mobilní aplikace je spuštěná	Zařízení cvuttest1

Obr. 6.6 Záznam událostí na Swiftshiledu po simulaci druhého scénáře testování

Dále byly nasnímány i hodnoty krevního tlaku (systola a diastola), které jsou vždy posílány na dohledový server Swiftshield (obr. 6.7). Tato data jsou dohledovým serverem Swiftshield přijata při druhé události “ Nové hodnoty měření tlaku“. Jak již bylo řečeno, jedná se o dvě náhodná čísla v rozsahu 0 až 200. Při prvním stisku tlačítka byly odeslány hodnoty 166 a 176 a při druhém stisku tlačítka byly odeslány hodnoty 163 a 176. Při prvním přijmutí dat ze senzoru byly na dohledový server poslány hodnoty 131 a 150, při druhém přijetí dat senzoru byly poslány hodnoty 98 a 173. Při třetím stisku tlačítka byly poslány hodnoty 99 a 173, při třetím přijetí dat ze senzoru byly poslány hodnoty 166 a 195. Při posílání live packetu byly odeslány hodnoty 12 a 34. Jak je z obr. 6.7 patrné dohledový server Swiftshield zaznamenal všechny odeslané náhodné hodnoty správně a ve správných časech.



Obr. 6.7 Záznam hodnot krevního tlaku po simulaci druhého scénáře testování

6.3 Testování kontroly logu

Třetí scénář testování ověřuje funkčnost automatické kontroly souboru `senzor.log` na SD kartě, která předchází posílání live packetu na dohledový server Swiftshiled. Tato kontrola detekuje, zda byla úspěšně odeslána všechna data na dohledový server od poslední kontroly či od vytvoření souboru `senzor.log` (podrobně kapitola 5.6.1). Pokud nedošlo k úspěšnému odeslání dat na server je u daného záznamu v logu nastaven flag na hodnotu 0 (viz. obr. 5.11). Součástí tohoto testování tedy je kontrola logu, při které bude zjištěno, že nedošlo k úspěšnému odeslání dat na dohledový server, proto budou na dohledový server poslána nová data a do logu bude zapsán záznam o chybě. Testováno bude následně i odeslání live packetu a uložení záznamu do logu.

Scénář tohoto testování je následující: Nejprve se provede inicializace základnové stanice, načtení statických hodnot IP adresy zařízení, masky sítě a IP adresy výchozí brány ze souboru `init_SD.txt`, konfigurování parametrů z DHCP serveru a nastavení hodin reálného času (RTC) z veřejného SNTP serveru. Poté se po stanoveném čase provede kontrola souboru `senzor.log`, který bude umístěn na SD kartě. Při kontrole bude u dvou záznamů zjištěno, že nedošlo k úspěšnému poslání dat na dohledový server Swiftshield. První neúspěšné odeslání dat bude zjištěno u záznamu, který vznikl po stisku tlačítka. Druhé neúspěšné odeslání bude zjištěno u záznamu, který vznikl po přijetí dat ze senzoru s akcelerometrem. Proto se dvakrát provede odeslání nových dat na dohledový server Swiftshield a následný zápis dat o neúspěšném odeslání do logu. Po skončení kontroly logu budou poslána data na dohledový server, která budou reprezentovat live packet a do

logu bude uložen záznam o odeslání live packetu. Po odeslání live packetu bude jednou stisknuto tlačítko. Po stisku tlačítka budou odeslány data na dohledový server Swiftshield a bude proveden zápis dat do logu. Pro účely testování byla upravena doba kontroly logu a posílání live packetu z 1 hodiny na 3 minuty. Pro simulaci neúspěšného odeslání dat na dohledový server Swiftshield byl upraven a použit soubor senzor.log z předchozího testování.

Pro vyhodnocení testování bude zjištěno, zda bylo správně detekováno, že nebyla úspěšně odeslána data na dohledový server. Dále bude porovnán čas v souboru senzor.log se záznamem událostí na dohledovém serveru Swiftshield. Dále budou porovnány odesílané hodnoty krevního tlaku (systola a diastola) s hodnotami zaznamenanými na dohledovém serveru.

```
thernet Init OK
PHY MICR initialized...
PHY MISR initialized...
Inicializace OK
Looking for DHCP server ...
IP assigned by DHCP:      10.0.0.139
Netmask assigned by DHCP: 255.255.255.0
Gateway assigned by DHCP: 10.0.0.138
Aktualni cas: [2014-04-27 17:15:33]
```

Obr. 6.8 Výpis inicializace a nastavení RTC při simulaci třetího scénáře testování

Na začátku simulace třetího scénáře (obr. 6.8) byly nejdříve načteny statické hodnoty IP adresa zařízení, maska sítě a IP adresa výchozí brány ze souboru init_SD.txt (výpis byl vypnut), protože nebylo v čekací smyčce stisknuto tlačítko. Poté se inicializoval Ethernet, který je nezbytný pro inicializaci LwIP stacku, který podporuje protokol DHCP, který je použit pro nakonfigurování parametrů IP adresy zařízení, masky sítě a IP adresy výchozí brány. Dále je použit protokol SNTP, který také podporuje LwIP stack, pro nastavení hodin reálného času (RTC).

Po inicializaci základnové stanice, načtení statických parametrů ze souboru, nakonfigurování parametrů pomocí DHCP serveru a nastavení RTC z veřejného SNTP bylo počkáno tři minuty na automatickou kontrolu logu. Při kontrole logu byly postupně procházeny všechny záznamy. Z obr. 6.9 je patrné, že podle záznamu v logu nedošlo v 15:35:37 a v 15:37:14 k úspěšnému odeslání dat na dohledový server, protože hodnota flagu je u obou záznamů nastavena na hodnotu 0. Proto nejdříve došlo k odeslání dat na dohledový server Swiftshield po zjištění, že v 15:35:57 nedošlo k úspěšnému odeslání dat

na dohledový server. Poté byl do logu zapsán záznam, že v 15:35:57 došlo k neúspěšnému odeslání dat na dohledový server Swiftshield. Soubor sensor.log byl poté dále procházen a bylo zjištěno, že k neúspěšnému odeslání došlo i v 15:37:14, proto byla opět odeslána data na dohledový server Swiftshield a do logu byl zapsán záznam o neúspěšném odeslání dat, ke kterému došlo v 15:37:14. Poté byla dokončena kontrola logu, při které již nebyl nalezen žádný další záznam o neúspěšném odeslání dat na dohledový server Swiftshield. Po kontrole logu byla odeslána data reprezentující live packet na dohledový server a do logu byl zapsán záznam o odeslání live packetu. Nakonec bylo stisknuto tlačítko, poté následovalo poslání dat na dohledový server a zapsání záznamu do logu.

[2014-04-27 15:35:18]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 15:35:57]	99	abcdefghijklmnopqrstu	255	1	255	0
[2014-04-27 15:36:35]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-27 15:37:14]	00	ACC01 INTERRUPT	255	1	CRC	0
[2014-04-27 15:37:51]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 15:38:18]	00	ACC01 INTERRUPT	255	1	CRC	1
[2014-04-27 15:44:46]	99	abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 17:18:28]	99	ER2014-04-27 15:35:57	255	1	255	1
[2014-04-27 17:18:33]	00	ER2014-04-27 15:37:14	255	1	CRC	1
[2014-04-27 17:18:34]	99	<>abcdefghijklmnopqrstu	255	1	255	1
[2014-04-27 17:18:40]	99	abcdefghijklmnopqrstu	255	1	255	1

Obr. 6.9 Výpis logu po simulaci třetího scénáře testování

Je patrné, že kontrola logu dopadla úspěšně. Záznam o neúspěšném odeslání dat v 15:35:57 byl při kontrole nalezen, došlo k odeslání dat na dohledový server a do logu byl zapsán v 17:18:28 záznam o chybném odeslání dat. Struktura tohoto záznamu je také správná, protože typ senzoru je opět 99 (stisk tlačítka) a v sekci data je zapsán čas, kdy nedošlo k odeslání dat (viz. obr. 5.11). Nalezen byl i druhý záznam o neúspěšném odeslání dat v 15:37:14, došlo k odeslání dat na dohledový server a do logu byl zapsán v 17:18:33 záznam o chybném odeslání dat. Struktura tohoto záznamu v logu je také správná, protože typ senzoru je opět 00 (senzor s akcelerometrem) a v sekci data je zapsán čas, kdy nedošlo k úspěšnému odeslání dat na dohledový server Swiftshield.

Dále je zřejmé, že hodiny reálné času byly nastaveny v 17:15:33, ale k odeslání data live packetu došlo až za 3 minuty a 7 vteřin, zatímco při simulaci druhého scénáře byl live packet odeslán dříve. Úloha operačního systému FreeRTOS, která inkrementuje čítač času

po vteřině, je sice spuštěna již po inicializaci základnové stanice a proto by k odeslání live packetu mělo dojít o něco dříve než 3 minuty od nastavení hodin reálného času. Při simulaci tohoto scénáře, však došlo k nalezení dvou záznamů o neúspěšném odeslání dat na dohledový server, proto se musí zohlednit čas, který je potřeba na zpracování těchto dvou událostí. Až poté můžou být na dohledový server odeslána data reprezentující live packet a do logu může být zapsán záznam o odeslání live packetu.

DATUM	UDÁLOST	ZDROJ
27.4.2014 17:18:48	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 17:18:48	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 17:18:42	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 17:18:42	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 17:18:37	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 17:18:36	Mobilní aplikace je spuštěná	Zařízení cvuttest1
27.4.2014 17:18:32	Nové hodnoty měření tlaku	Zařízení cvuttest1
27.4.2014 17:18:31	Mobilní aplikace je spuštěná	Zařízení cvuttest1

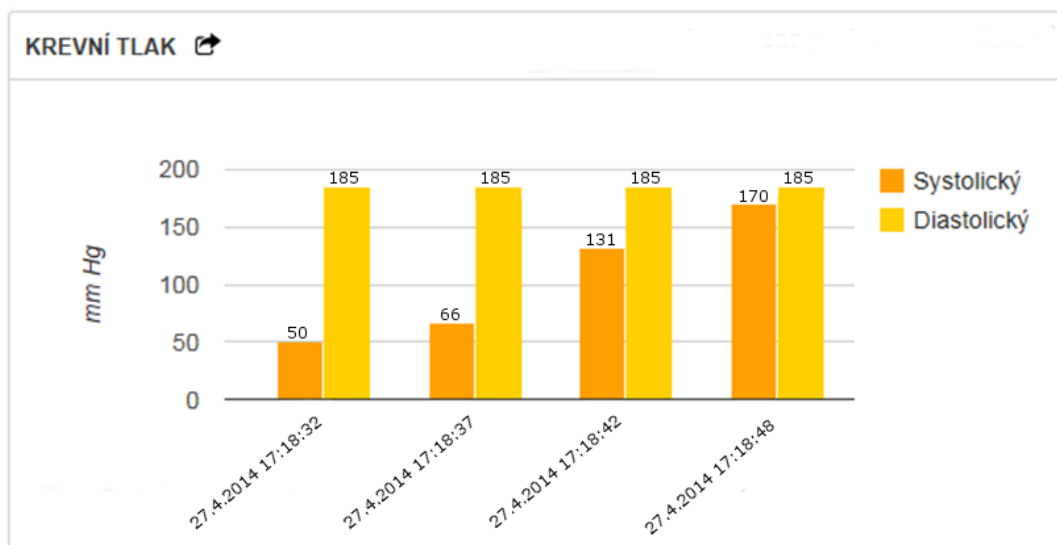
Obr. 6.10 Záznam událostí na Swiftshiledu po simulaci třetího scénáře testování

Po provedení simulace třetího scénáře byl opět nasnímán i seznam událostí, které zaznamenal dohledový server Swiftshield (obr. 6.10). Opět se odeslání dat na server ze základnové stanice projeví jako dvě události: “Mobilní aplikace je spuštěná“ a “Nové hodnoty měření tlaku“. Při simulaci tohoto scénáře došlo celkem čtyřikrát k odeslání dat na dohledový server, proto je na serveru správně zaznamenáno celkem osm událostí, protože jedno odeslání je zaznamenáno na serveru jako dvě události. Z logu je patrné, že všechna odeslání na dohledový server byla úspěšná, protože všechny flagy odeslání u jednotlivých záznamů jsou nastaveny na hodnotu 1.

Při porovnání času v logu s časem, kdy byla událost zaznamenána na dohledovém serveru, je patrný časový rozdíl. Tento fakt je způsoben tím, že v logu je zaznamenán čas, kdy bylo při kontrole zjištěno, že data nebyla úspěšně odeslána na dohledový server. Při stisku tlačítka je zaznamenán čas, kdy došlo ke stisku tlačítka.

Opět byly nasnímány i hodnoty krevního tlaku (systola a diastola), které jsou vždy posílány na dohledový server Swiftshield (obr. 6.11). Opět se jednalo o dvě náhodná čísla v rozsahu 0 až 200. Při prvním zjištění, že nedošlo k úspěšnému odeslání dat na dohledový server, byly odeslány hodnoty 50 a 185. Při druhém zjištění byly odeslány hodnoty 66 a 185. Při posílání live packetu byly odeslány hodnoty 131 a 185 a po stisku tlačítka byly poslány hodnoty 170 a 185. Jak je z obr. 6.11 patrné, dohledový server

Swiftshield zaznamenal všechny odeslané náhodné hodnoty správně a ve správných časech.



Obr. 6.11 Záznam hodnot krevního tlaku po simulaci třetího scénáře testování

7 Závěr

Cílem této práce bylo seznámit se s problematikou domácí asistivní péče a s obvyklými požadavky na vzdálené monitorování osob v domácí péči. Dalším cílem bylo seznámit se s hardwarem dodané základnové stanice domácího asistivního systému a s příslušejícími bezdrátovými snímači. Nedílnou součástí této práce byl i návrh firmwaru pro základnovou stanici domácího asistivního systému, který umožní příjem dat z bezdrátových senzorů, zpracování těchto dat, jejich logování a odesílání na vzdálené dohledové pracoviště. Navržený firmware byl implementován a byla ověřena jeho funkce.

V prvních dvou kapitolách byla popsána problematika stárnutí populace a problematika domácí asistivní péče. Podrobně byl popsán současný fenomén stárnutí populace v České republice i v Evropské unii a s ním spojené zvyšující se požadavky na zdravotní péči. Nejlépe, jak uspokojit zvyšující se požadavky, je zvolit vhodnou formu asistivní technologie, konkrétně telemedicínu. Telemedicína zahrnuje i domácí asistivní systémy, jejímž návrhem a implementací se tato práce zabývala.

Bylo zjištěno, že domácí asistivní systém by měl být schopen monitorovat fyziologické funkce, rizikové situace a každodenní aktivity uživatele. V dostupné literatuře byly identifikovány požadavky uživatelů na domácí asistivní systém. Uživatelé požadují, aby domácí asistivní systémy byly snadně použitelné, integrovány do technologií, které již umějí ovládat či by měly podporovat samostatnost starších lidí. Důležitým požadavkem je, aby telemedicínský systém používal senzory, které nejsou umístěné na těle a dále by tento systém neměl porušovat soukromí sledované osoby. Systém by měl být i odolný a spolehlivý. Kromě tohoto, by měl být monitoring všudypřítomný a nemělo by být nutné aktivování systému sledovanou osobou.

V třetí kapitole byla podrobně popsána dodaná základnová stanice a její hardwarové vybavení umožňující připojení senzorů. Popsán byl i dodaný senzor s akcelerometrem, který byl použit při návrhu systému. V následující kapitole byly popsány knihovny využívané při návrhu firmwaru. Konkrétně se jedná o operační systém reálného času FreeRTOS, implementaci protokolu TCP/IP LwIP stack a FatFs R0.10, pomocí které lze realizovat systém souborů FAT. Dále byla v této kapitole popsána konfigurace a použití jednotlivých periférií, které byly použity při návrhu firmwaru.

Další část této práce se zabývala implementací a návrhem firmwaru pro základnovou stanici domácího asistivního systému, který umožní příjem dat z bezdrátových senzorů, zpracování těchto dat, jejich logování a odesílání na vzdálené dohledové pracoviště. Výsledná aplikace podporuje konfiguraci parametrů zařízení z DHCP serveru, nastavení hodin reálného času (RTC) pomocí SNTP protokolu, příjem a zpracování dat ze senzoru s akcelerometrem, odeslání dat na dohledový server Swiftshield, zápis do logu a kontrolu dat v logu, který je umístěn na SD kartě. Navržený firmware byl i úspěšně otestován.

Do budoucna je možné rozšířit navržený firmware i hardwarovou výbavu. Použito může být například více senzorů, které by mohly monitorovat fyziologické funkce, rizikové situace či každodenní aktivity uživatele. Aplikace by poté mohla data přijímaná z těchto senzorů efektivně zpracovávat a reagovat na ně. Cílem této práce však nebylo vytvořit komplexní systém pro domácí asistivní péči. Cílem bylo navrhnout a implementovat pouze základ pro plnohodnotný domácí asistivní systém, který by splňoval požadavky této práce a umožňoval by rozšiřitelnost.

Literatura

- [1] VOHRALÍKOVÁ, Lenka a Ladislav RABUŠIC (2004). Čeští senioři včera, dnes a zítra. [cit. 2014-3-26].
Dostupné z: http://praha.vupsv.cz/Fulltext/vz_149.pdf
- [2] SVODOVÁ, Kamila (2013). Demografické stárnutí ČR podle výsledků projekce.[online]. [cit. 2014-03-26].
Dostupné z: http://www.demografie.info/?cz_detail_clanku=&artclID=824
- [3] SVODOVÁ, Kamila (2005). Stárnutí populace podle výsledků projekce ČSÚ. [online]. [cit. 2014-03-26].
Dostupné z: http://www.demografie.info/?cz_detail_clanku&artclID=34
- [4] Stárnoucí Evropa? Skutečnost, na kterou je třeba se připravit (2012). [online]. [cit. 2014-03-26].
Dostupné z: http://ec.europa.eu/news/economy/120515_cs.htm
- [5] ŠTĚPÁNKOVÁ, Olga a Pavel SLAVÍK. ČVUT FEL (2014). Asistivní technologie, telemedicína a dohledové systémy, Přednášky předmětu Asistivní technologie a dohledové systémy.
- [6] MEYSTRE, Stephane (2005). The current state of telemonitoring: a comment on the literature, Telemedicine Journal & e-Health, vol. 11, no. 1, pp. 63–69.
- [7] D.WILSON, S. CONSOLVO, K. FISHKIN and M. PHILIPOSE (2005). In-home assessment of the activities of daily living of the elderly, in ExtendedAbstracts of CHI 2005: Workshops-HCI Challenges in Health Assessment.
- [8] Fyziologické funkce [online]. [cit. 2014-03-27]. ISSN 1804-6517. Dostupné z: http://www.wikiskripta.eu/index.php/Fyziologick%C3%A9_funkce

- [9] BAKKES, S.C.J., MORSCH, R. & KRÖSE, B.J.A. (2011). Telemonitoring for Independently Living Elderly: Inventory of Needs & Requirements. In J. Maitland, J.C. Augusto & B. Caulfield (Eds.), Proceedings of the Pervasive Health 2011 conference (pp. 152-159).
- [10] Oslabení kardiovaskulárního systému (2012). In: [online]. Zdravotní tělesná výchova, Fakulta sportovních studií MU [cit. 2014-03-27]. Dostupné z: <http://is.muni.cz/do/fsps/e-learning/ztv/doc/kardio.pdf>
- [11] LUSIGNAN, S., ALTHANS, A., WELLS, S., JOHNSON, P., VANDENBURG, M. and ROBINSON, J. (2000). A pilot study of radiotelemetry for continuous cardiopulmonary monitoring of patients at home, Journal of Telemedicine and Telecare, vol. 6, no. Supplement 1, p. 119.
- [12] JOHNOSON, P., and ANDREWS, D. (1996). Remote continuous physiological monitoring in the home. Journal of telemedicine and telecare, vol. 2, no. 2, p. 107.
- [13] BARBARO, V., BARTOLINI, P. and BERNARDUCCI, R. (1997). A portable unit for remote monitoring of pacemaker patients, Journal of telemedicině and telecare, vol. 3, no. 2, p. 96.
- [14] GÓMEZ, E., HERNANDO, M., GARCIA, A., DEL POZO, F., CERMENO, J., CORCOY, R., BRUGUES, E. and DE LAIVA, A., (2002). Telemedicine as a tool for intensive management of diabetes: the DIABTel experience, Computer Methods and Programs in Biomedicine, vol. 69, no. 2, pp. 163–177.
- [15] NICOGOSSIAN, A., POBER, D. and ROY, S. (2001). Evolution of telemedicine in the space program and earth applications, Telemedicine Journal and e-Health, vol. 7, no. 1, pp. 1–15.

- [16] SATAVA, R., ANGOOD, P., HARNETT, B., MACEDONIA, C. and MERRELL, R. (2000). The physiologic cipher at altitude: telemedicine and real-time monitoring of climbers on Mount Everest, *Telemedicine Journal and e-Health*, vol. 6, no. 3, pp. 303–313.
- [17] FINKELSTEIN, J., CABRERA, M. and HRIPCSAK, G. (2000). Internet-based home asthma telemonitoring: can patients handle the technology? *Chest Chicago*, vol. 117, no. 1, pp. 148–155.
- [18] OOHASHI, T., KAWAI, N., HONDA, M., NAKAMURA, S., MORIMOTO, M., NISHINA, E. and MAEKAWA, T. (2002). Electroencephalographic measurement of possession trance in the field, *Clinical Neurophysiology*, vol. 113, no. 3, pp. 435–445.
- [19] FIELD, M. and GRIGSBY, J. (2002). Telemedicine and remote patient monitoring, *Jama*, vol. 288, no. 4, p. 423.
- [20] GÜLER, N. and ÜBEYLI, E. (2002). Theory and applications of biotelemetry, *Journal of Medical Systems*, vol. 26, no. 2, pp. 159–178.
- [21] BERG, W. P., ALESSIO, H. M., MILLS, E. M. and TONG, C. (1997). Circumstances and consequences of falls in independent community-dwelling older adults. *Age and Ageing*, 26: 261-68.
- [22] HORTON, K., R. RN, and P. RNT (2008). Falls in older people: The place of telemonitoring in rehabilitation. *Journal of rehabilitation research and development*, vol. 45, no. 8, p. 1183.
- [23] BOURKE, A., O'BRIEN, J., and LYONS, G. (2007). Evaluation of a thresholdbased tri-axial accelerometer fall detection algorithm, *Gait & posture*, vol. 26, no. 2, pp. 194–199.
- [24] NAIT-CHARIF, H. and MCKENNA, S. (2004). Activity summarisation and fall detection in a supportive home environment, in *Pattern Recognition. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4.

- [25] VÍTEK, S. and RICHTER J. (2013). Removing barriers from cities the smart way. Smart Homes 2013, 2nd Conference on Innovations in Assistive Technologies and Health Care, pp. 11-13.
- [26] SIXSMITH, A. and JOHNSON, N. (2004). A smart sensor to detect the falls of the elderly, IEEE Pervasive Computing, vol. 3, no. 2, pp. 42–47.
- [27] POPESCU, M., COUPLAND, S. and DATE, S. (2008). A fuzzy logic system for acoustic fall detection.
- [28] MAURER, U., ROWE, A., SMAILAGIC, A., and SIEWIOREK, D. (2006). eWatch: a wearable sensor and notification platform,” in Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on, p. 4.
- [29] HUANG, C., TSENG, Y., and WU, H. (2007). Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network, ACM Transactions on Sensor Networks (TOSN), vol. 3, no. 1, p. 5.
- [30] KATZ, S., DOWNS, T., CASH, H., and GROTZ, R. (1970). Progress in development of the index of ADL, The Gerontologist, vol. 10, no. 1 Part 1, p. 20.
- [31] KALVACH, Zdeněk, Zdeněk ZADÁK, Roman JIRÁK, Helena ZAVÁZALOVÁ, Iva HOLMEROVÁ a Pavel WEBER (2008). Geriatrické syndromy a geriatrický pacient. 1. vydání. Praha: Grada Publishing a.s. ISBN 978-80-247-2490-4.
- [32] LOGAN, B., HEALEY, J., PHILIPOSE, M., TAPIA, E. M. and INTILLE, S. S. (2007)). A long-term evaluation of sensing modalities for activity recognition, in Proc. of Ubicomp07, pp. 483–500.
- [33] SPONSELEE, A., SCHOUTEN, B., and BOUWHUIS, D.(2010). Telecare for elderly users: Needs and benefits, Gerontechnology, vol. 9, no. 2, p. 249.

- [34]FRANĚK, Petr. Maslowova pyramida lidských potřeb. Filozofie úspěchu [online]. 2011 [cit. 2014-03-31]. Dostupné z: <http://www.filosofie-uspechu.cz/maslowova-pyramida-lidskych-potreb/>
- [35]DAMODARAN, L. and OLPHERT, C. (2010). Assisted living technologies: Users' needs and challenges for successful uptake, *Gerontechnology*, vol. 9, no. 2, p. 249.
- [36]DIAZ, U., GARCÍA, A. and URDANETA, E. (2010). What elderly users do not want from technology: A qualitative approach, *Gerontechnology*, vol. 9, no. 2, p. 210.
- [37]TICHÝ, Miroslav. ARM: Představení a vývoj architektury ARM. In: [online]. 2009 [cit. 2014-04-06]. Dostupné z: [http://wh.cs.vsb.cz/mil051/images/4/43/PAP_Architektura_procesoru_ARM_prezentace_\(Miroslav_Tich%C3%BD\).pdf](http://wh.cs.vsb.cz/mil051/images/4/43/PAP_Architektura_procesoru_ARM_prezentace_(Miroslav_Tich%C3%BD).pdf)
- [38]TIŠNOVSKÝ, Pavel. Co se děje v počítači: Pohled programátora na mikroprocesory ARM. In: [online]. Root.cz, 2012 [cit. 2014-04-06]. Dostupné z: <http://www.root.cz/clanky/pohled-programatora-na-mikroprocesory-arm/#ic=serial-box&icc=text-title>
- [39]KUZNÍK, Bořek. ARM architektura - nejen mobilní procesory. *Notebook.cz* [online]. 2010 [cit. 2014-04-06]. Dostupné z: <http://notebook.cz/clanky/technologie/2010/arm-architektura>
- [40]SADASIVAN, Shyam. An Introduction to the ARM Cortex-M3 Processor. [online]. 2006 [cit. 2014-04-08]. Dostupné z: <http://www.arm.com/files/pdf/IntroToCortex-M3.pdf>
- [41]STM32F207xx. Datasheet, STMicroelectronics (www.st.com), [online]. [cit. 2014-04-11]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00237391.pdf>

- [42] FreeRTOS[online]. [cit. 2014-04-11]. Dostupné z: <http://www.freertos.org/>
- [43] A Lightweight TCP/IP stack [online]. [cit. 2014-04-13]. Dostupné z: <http://savannah.nongnu.org/projects/lwip/>
- [44] AN3384 Application note: lwIP TCP/IP stack demonstration for STM32F2x7xx microcontrollers. STMicroelectronics[online]. 2011 [cit. 2014-04-13]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/DM00026013.pdf
- [45] Vývojová deska UDS1: Nejjednodušší cesta k USB. In: Asix [online]. 2009 [cit. 2014-04-16]. Dostupné z: http://asix.cz/products_museum_uds1.htm
- [46] Modul UMS1: převodník USB-UART: Nejjednodušší cesta k USB. In: Asix [online]. 2008 [cit. 2014-04-16]. Dostupné z: http://www.asix.cz/products_museum_ums1.htm
- [47] UM1079 User manual: STM32L1 discovery kits: STM32L-DISCOVERY and 32L152CDISCOVERY. STMicroelectronics[online]. 2013 [cit. 2014-04-16]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00027954.pdf
- [48] DP83848C: DP83848C PHYTER Commercial Temperature Single Port 10/100 Mb/s Ethernet Physical Layer Transceiver. Texas Instruments [online]. 2011 [cit. 2014-04-16]. Dostupné z: <http://www.ti.com.cn/cn/lit/ds/symlink/dp83848c.pdf>
- [49] STM32F100x4, STM32F100x6, STM32F100x8, STM32F100xB. Datasheet, STMicroelectronics (www.st.com), [online]. [cit. 2014-04-17]. Dostupné z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00251732.pdf>

- [50]FatFs - Generic FAT File System Module [online]. [cit. 2014-04-18]. Dostupné z: http://elm-chan.org/fsw/ff/00index_e.html
- [51]DS2401. Datasheet, Maxim Integrated (<http://www.maximintegrated.com>), [online]. [cit. 2014-04-19]. Dostupné z <http://datasheets.maximintegrated.com/en/ds/DS2401.pdf>
- [52]RFM12 Universal ISM Band FSK Transceiver. Hoperf Electronic (<http://www.hoperf.com>)[online]. [cit. 2014-04-19]. Dostupné z <http://www.hoperf.com/upload/rf/RFM12.pdf>
- [53]KEIL – Tools by ARM [online]. [cit. 2014-04-20]. Dostupné z: <https://www.keil.com/download/>
- [54]STM32F2xx_StdPeriph_Lib_V1.0[online]. [cit. 2014-04-20]. Dostupné z: http://subversion.assembla.com/svn/freertos/STM32F2xx_StdPeriph_Lib_V1.0.0/Project/
- [55]Swiftshield[online]. [cit. 2014-04-23]. Dostupné z: <http://swiftshield.com/>

A Seznam použitých zkratek

ALU Arithmetic Logic Unit - Aritmeticko-logická jednotka

API Application Programming Interface

ARM Advanced RISC Machine – 32 bitová architektura RISC od firmy ARM Limited

ARP Address Resolution Protocol - síťový protokol pro zjišťování MAC adres

ART Adaptive Real Time

CAN Controller Area Network - komunikační standard, definující fyzickou a linkovou vrstvu, primárně vyvinut pro automobilový průmysl

CMOS Complementary Metal–Oxide–Semiconductor

CPSR Current Processor Status Register - stavový registr

CPU Central Processing Unit – procesor, mikroprocesor

CRC Cyclic redundancy Chec - Cyklický redundantní součet

DHCP Dynamic Host Configuration Protocol – protokol pro automatickou konfiguraci zařízení připojených do počítačové sítě

DMA Direct Memory Access - přímý přístup do paměti

DNS Domain name system - hierarchický systém doménových jmen

EEG Elektroencefalogram -záznam časové změny elektrického potenciálu způsobeného mozkovou aktivitou

EEPROM Electrically Erasable Programmable Read-Only Memory – elektricky mazatelná nevolatilní paměť

EKG Elektrokardiogram - záznam časové změny elektrického potenciálu způsobeného srdeční aktivitou

EMG Elektromyografie – vyšetřuje funkci svalů pomocí elektrických biosignálů

ETH Ethernet – označení komunikačního standardu IEEE 802.3

EU Evropská unie

FAT File Allocation Table

FatFs Generic FAT File System Module

FIFO First In, First Out - fronta

FIQ Privilegovaný režim rychlého přerušení procesoru

FreeRTOS Free real time operating system

FTDI Future Technology Devices International Ltd.

GPIO General-purpose input/output

GPL GNU General Public License - všeobecná veřejná licence GNU

HW Hardware

ICMP Internet Control Message Protocol

IGMP Internet Group Management Protocol

IP Internet Protocol – protokol síťové vrstvy

IRQ Privilegovaný režim přerušení procesoru

JTAG Joint Test Action Group – komunikační standard definovaný normou IEEE 1149.1

LED Light Emitting Diode – elektronická polovodičová součástka

LSB Least Significant Bit - nejméně významný bit

LwIP A Lightweight TCP/IP stack

MAC Media Access Control – část linkové vrstvy specifikace IEEE 802.3, definuje řízení přístupu ke sdílenému médiu

MID Mobile Internet Device – Mobilní internetové zařízení

MII Media Independent Interface – komunikační rozhraní mezi MAC a transceiverem Ethernet

MMU Memory management unit - Jednotka správy paměti

MPU Memory protection unit – Jednotka ochrana paměti

MSB Most Significant Bit - nejvýznamnější bit

NTP Network Time Protocol

NVIC Nested Vectored Interrupt Controller

PC Personal Computer – osobní počítač

PDA Personal Digital Assistant – osobní digitální pomocník

PHY Fyzická vrstva standardu ISO/OSI

PPP Point-to-Point Protocol - protokol linkové vrstvy pro přímé spojení mezi dvěma síťovými uzly

PROM Programmable Read Only Memory - elektricky "jednorázově" programovatelná permanentní paměť

RAM Random-Access Memory - paměť s přímým přístupem nebo paměť s libovolným výběrem

RISC Reduced Instruction Set Computing - procesory s redukovanou instrukční sadou

RFID Radio Frequency Identification - identifikace na rádiové frekvenci

RTC Real-time clock - hodiny reálného času

SDIO Secure Digital Input Output

SIMD Single Instruction, Multiple Data – instrukce, pomocí nichž se aplikuje jedna operace na vektor dat

SNMP Simple Network Management Protocol – pro potřeby správy sítí

SNTP Simple Network Time Protocol - protokol pro synchronizaci vnitřních hodin zařízení

SPI Serial Peripheral Interface - sériové periferní rozhraní

SPSR Saved Processor Status Register – stavový archivní registr

SUP Privilegovaný režim supervizora procesoru

SWD Serial Wire Debug – dvouodičová alternativa standardu JTAG

TCP Transmission Control Protocol – protokol datových sítí na úrovni transportní vrstvy, poskytuje spojovanou službu

TCP/IP Transmission Control Protocol/Internet Protocol - primární přenosový protokol/protokol síťové vrstvy

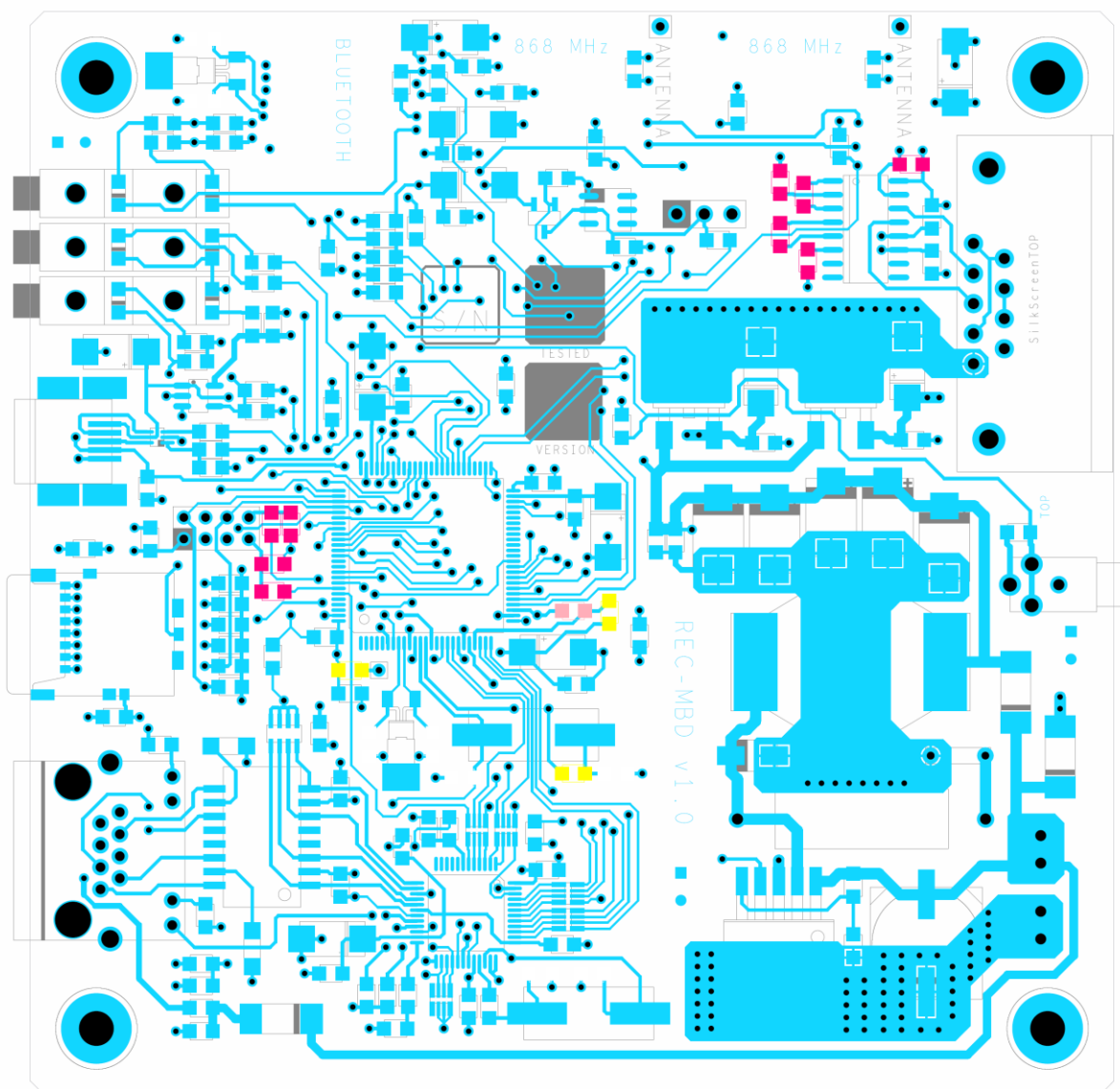
UDP User Datagram Protocol - protokol datových sítí na úrovni transportní vrstvy, poskytuje nespojovanou službu

USART Universal Synchronous / Asynchronous Receiver and Transmitter -Synchronní / Asynchronní sériové rozhraní USART

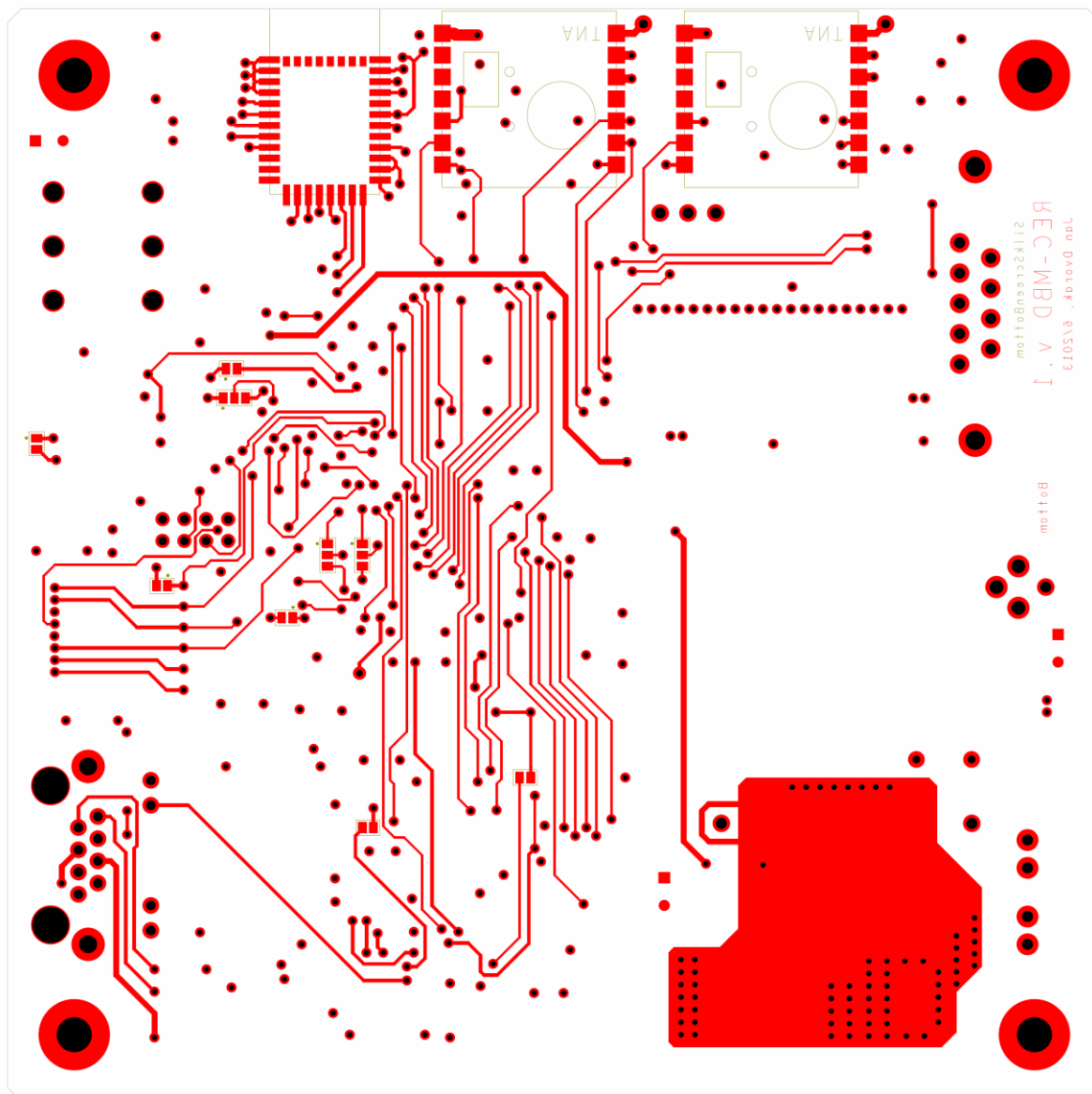
USB Universal Serial Bus – univerzální sériová sběrnice

USR Uživatelský režim procesoru

B Layouty základnové stanice



Obr. B.1 Layout horní vrstvy základnové stanice



Obr. B.2 Layout horní vrstvy základnové stanice

C Obsah příloženého CD

K této práci je přiloženo CD s následujícím obsahem:

- Firmware Složka projektu Keil μ Vision 5, která obsahuje zdrojové kódy navrženého firmwaru a zdrojové kódy používaných knihoven.
- Akcelerometr Složka obsahuje dodané zdrojové kódy senzoru s akcelerometrem.
- Layouty Složka s layouty základnové stanice.
- blokova_schemata.pdf Blokovaná schémata ve formátu PDF použitého hardwaru.
- dp_2014_pecolmar.pdf Text diplomové práce ve formátu PDF.