



Zadání diplomové práce

Název:	Datalogger pro zařízení měřící spektrum ionizujícího záření
Student:	Bc. Libor Kuchař
Vedoucí:	Ing. Filip Štěpánek
Studijní program:	Informatika
Obor / specializace:	Návrh a programování vestavných systémů
Katedra:	Katedra číslicového návrhu
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Navrhněte a realizujte datalogger (zařízení, které zaznamenává data v čase), který připojíte k reálné či simulované analogové měřící jednotce ionizujícího záření. Návrh a realizace bude obsahovat i obslužnou aplikaci pro PC.

Řešení se bude skládat z těchto částí:

- dataloggeru na bázi mikrokontroleru STM32 a wifi modulu ESP8266,
- obslužné aplikace pro PC, která bude vzdáleně ovládat datalogger za účelem konfigurace zařízení, stahování naměřených dat a jejich vizualizaci.

Postupujte následovně:

- Analyzujte možnosti měření či simulace měření ionizujícího záření a požadavky na datalogger
- Na základě analýzy navrhněte hardware (deska plošných spojů pro propojení mikrokontroléru STM32, modulu ESP8266, analogové měřící jednotky ionizujícího záření) a firmware dataloggeru, který bude komunikovat se zvoleným detektorem ionizujícího záření, ukládat data na paměťové úložiště a komunikovat s PC aplikací.
- Navrhněte a otestujte obslužnou PC aplikaci.
- Návrh dataloggeru realizujte a otestujte.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Datalogger pro zařízení měřící spektrum ionizujícího záření

Bc. Libor Kuchař

Katedra číslicového návrhu

Vedoucí práce: Ing. Štěpánek Filip

22. dubna 2021

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Filipu Štěpánkovi za věnovaný čas, náměty a rady. Také bych mu rád poděkoval za trpělivost, kterou se mnou měl.

Dále bych rád poděkoval Ing. Martinu Kohlíkovi PhD. za poskytnuté rady ohledně diplomové práce a veškerou pomoc a podporu, kterou mi během celého studia poskytl.

Rád bych také poděkoval RNDr. Ivanu Hendrychovi za pomoc a rady ohledně fotonásobiče.

Také bych chtěl poděkovat své rodině, která mě poskytovala ideální podmínky pro studium a všem svým přátelům, kteří mě podporovali a stáli po mém boku po celou dobu studia. Děkuji svým rodičům za zázemí, které mi během psaní diplomové práce poskytovali.

Jsem rád, že jsem mohl studovat na katedře číslicového návrhu. Oceňuji přátelský, avšak profesionální přístup ke studentům od vyučujících z této katedry. Děkuji, že jsem se mohl účastnit konferencí pořádaných KČN a výjezdních zasedání.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 22. dubna 2021

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Libor Kuchař. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Kuchař, Libor. *Datalogger pro zařízení měřící spektrum ionizujícího záření*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Datalogger je zařízení zachycující data v čase. V rámci textu závěrečné práce je rozebíráno, jak vhodně navrhnout a realizovat datalogger v domácích podmínkách vhodný pro měření spektra ionizujícího záření. V rámci tohoto zadání jsou rozebrány požadavky na funkčnost a technické vybavení / návrh daného zařízení. Realizace se skládá z navržení desky plošných spojů, vestavného FW pro mikrokontrolér STM32 a Wi-Fi modul ESP-01. Součástí je též návrh realizace obslužné PC aplikace pro Windows 10. Text diskutuje, jak fyzikální pozadí měření daného spektra záření, tak i podrobnosti z návrhu, realizace a testování daného vestavného systému.

Klíčová slova Datalogger, Spektrum ionizujícího záření, STM32, ESP8266, Windows 10 aplikace, Simulátor detektoru ionizujícího záření

Abstract

Datalogger is a device that records data over time. The content of this thesis discusses how to appropriately design and implement a homemade datalogger suitable for measuring the spectrum of ionizing radiation. Within the assignment, the requirements for functionality and technical equipment / design of the product are discussed. The realisation consists of designing of a printed circuit board, embedded firmware for the STM32 microcontroller and the Wi-Fi module ESP-01. It also includes implementation of the service PC application for Windows 10. Physical background of the measurement of the ionizing radiation spectrum as well as details regarding the design, realization, and testing of the final product are described in the text.

Keywords Datalogger, Ionizing radiation spectrum, STM32, ESP8266, Windows 10 application, Ionizing radiation detector simulator

Obsah

Úvod	1
1 Analýza	3
1.1 Ionizující záření	3
1.1.1 Fotonové záření (gama, X)	4
1.1.2 Interakce gama záření s látkou	5
1.2 Count rate	7
1.3 Elektronvolt	7
1.4 Spektroskopie ionizujícího záření	7
1.4.1 Scintilační detektor	7
1.4.2 Zpracování signálu z detektoru	9
1.4.3 Existující řešení pro zpracování signálu z detektoru	10
1.4.4 Požadavky na datalogger	11
1.5 Spektrum ionizujícího záření	12
1.6 Simulace detektoru ionizujícího záření	17
2 Návrh	19
2.1 Návrh Dataloggeru	19
2.1.1 Architektura dataloggeru	21
2.1.2 Řídící jednotka	21
2.1.2.1 Vývojová deska Blue Pill	23
2.1.2.2 Vývojová deska WeAct Black Pill	25
2.1.3 Wi-Fi modul	26
2.1.4 Hodiny reálného času	27
2.1.5 Výběr hardwaru pro non-volatilní paměť	27
2.1.6 Výběr senzoru teploty, vlhkosti a tlaku	28
2.1.7 Návrh specifických komunikačních protokolů	29
2.1.7.1 Request protokol	30
2.1.7.2 Command protokol	31

2.1.7.3	UDPComm protokol	35
2.1.8	Napájení dataloggeru	36
2.1.9	Zajištění spolehlivého provozu	37
2.2	Návrh počítačové aplikace	39
2.3	Plán realizace	40
3	Realizace dataloggeru	41
3.1	Konfigurace hodinového signálu	41
3.2	Konfigurace dataloggeru	43
3.2.1	Komunikace s EEPROM	46
3.2.2	Stavové hodnoty dataloggeru	46
3.2.3	Konfigurace bezdrátových sítí	48
3.3	Měření reálného času	49
3.3.1	Synchronizace reálného času	49
3.3.2	Čtení systémového času	50
3.4	Ukládání a čtení naměřených dat	51
3.4.1	Operace s fyzickým úložištěm	51
3.4.2	Formát souboru měření	53
3.5	Měření spektra ionizujícího záření	54
3.5.1	Konverze pomocí spouště	54
3.5.2	Pokračované měření signálu	55
3.6	Měření CPM pomocí pomocných měřicích zařízení	57
3.7	Komunikace mezi PC a dataloggerem	58
3.8	Příkazový subsystém	59
3.8.1	Jednocyklové operace	59
3.8.2	Operace: aktualizace dat v mezipaměti	60
3.8.3	Operace: připojení k Wi-Fi síti	61
3.8.4	Operace: start přístupového bodu	61
3.8.5	Operace: poslušání na daném portu	63
3.8.6	Operace: získávání seznamu souborů měření	63
3.8.7	Operace: čtení části souboru	65
3.8.8	Operace: získávání informací o připojení	65
3.8.9	Operace: skenování sítí	65
3.9	Řadič měření	66
3.9.1	Počáteční stav (zastaveno)	67
3.9.2	Stav: inicializace měření	68
3.9.3	Stav: měření	70
3.9.4	Stav: deinicializace měření	70
3.9.5	Stav: připojování k síti	71
3.9.6	Stav: kontrola sítě	72
3.9.7	Stav: aktualizace mezipaměti ESP	72
3.10	Konfigurace čítačů/časovačů	72
3.10.1	Sekundový časovač	74
3.10.2	Mikrosekundový časovač	74

3.10.3	Časovač zajišťující počítání CPM	75
3.10.4	Časovač pro A/D převodník	75
3.10.5	Nastavování čítače pro jednotku RTC	76
3.11	Priority přerušení	77
3.12	Výpočet CRC32	79
3.13	Chybové stavy dataloggeru	80
3.14	ESP-01 firmware	80
3.14.1	Zapojení ESP-01	81
3.14.2	Selektivní opakování datagramu	82
3.15	Výroba plošného spoje	83
3.16	Cena dataloggeru	88
4	Realizace počítačové aplikace	91
4.1	Komunikace s dataloggerem	91
4.1.1	Komunikační most	92
4.1.2	UDP most	93
4.1.3	Sériový most	94
4.1.4	Fronta požadavků	96
4.2	Správce souborů	97
4.2.1	Reprezentace dat měření v aplikaci	97
4.2.2	Načítání dat ze souboru	98
4.2.3	Ukládání dat do souboru	99
4.2.4	Oprava souborů	99
4.3	Okna aplikace	101
4.3.1	Hlavní okno	101
4.3.2	Správce dat	104
4.3.3	Grafické zobrazení dat	105
5	Testování systému	109
5.1	Simulátor detektoru ionizujícího záření	109
5.1.1	Generování impulzu gaussovského tvaru	110
5.1.2	Simulace zářiče a detektoru	111
5.2	Zpracování analogového signálu dataloggerem	113
5.2.1	Zpracování signálu pomocí konverze se spouští	113
5.2.2	Zpracování signálu pomocí pokračovaného měření	115
5.3	Test se simulovaným zářičem	119
5.4	Test funkcionalit systému	124
5.4.1	Manipulace se soubory měření	125
5.4.2	Selhání modulů dataloggeru během měření	127
5.4.3	Kombinovaný test funkcionalit	129
	Závěr	133
	Literatura	135

A	Seznam použitých zkratek	143
B	Obsah příloženého CD	145
C	Nástroje použité pro vývoj a testování systému	147
D	Uživatelský manuál	151
D.1	Datalogger	151
D.2	Obslužná počítačová aplikace	152
D.2.1	Navázení spojení s dataloggerem	153
D.2.2	Čtení stavu dataloggeru a jednotlivých modulů	155
D.2.3	Spuštění/zastavení měření	156
D.2.4	Čtení aktuálních dat, záznam dat	156
D.2.5	Manipulace se soubory v dataloggeru	157
D.2.6	Konfigurace bezdrátových sítí	160
D.2.7	Konfigurace dataloggeru	161
D.2.8	Správce dat	163
D.2.9	Grafické zobrazení dat	164
D.3	První spuštění	165

Seznam obrázků

1.1	Schéma vzniku gama záření	4
1.2	Pronikavost alfa, beta a gama záření	5
1.3	Interakce gama záření s látkou	6
1.4	Jedna z možných variant zapojení fotonásobiče	8
1.5	Impulz gaussovského tvaru	9
1.6	Obvod na zpracování pulzu z fotonásobiče	10
1.7	Existující řešení pro měření spektra ionizujícího záření – Řešení od RHElectronics	11
1.8	Typický tvar spojitého spektra beta mínus a beta plus záření	13
1.9	Typický tvar diskrétního spektra alfa zářičů	13
1.10	Scintilační spektrum Cesia-137	14
1.11	Scintilační spektrum neznámého gama zářiče	16
1.12	Scintilační spektrum Kobaltu-60	16
1.13	Generování signálu podle spektra pomocí emulátoru CEAN	17
1.14	Blokové schéma emulátoru DT5810	18
2.1	Základní části systému	20
2.2	Blokové schéma desky plošných spojů dataloggeru	22
2.3	Porovnání Blue Pillu a WeAct Black Pllu	24
2.4	Výstup STM32 ST-Link utility pro CS32F103C8T6 a STM32F103C8T6	25
2.5	BME280 modul	29
2.6	Komunikační schéma systému	30
2.7	Ukázka command protokolu	35
3.1	Konfigurace hodin - výběr oscilátoru	43
3.2	Konfigurace hodin - Nastavení systémových hodin	44
3.3	Konfigurace hodin - Nastavení hodin sběrnic	44
3.4	Výsledná konfigurace hodin v STM32CubeMX	45
3.5	Jak funguje Chan FatFs	52
3.6	Průběh konverze pomocí spouště	56

3.7	Pokračované měření signálu	57
3.8	Diagram jednocyklové operace	60
3.9	Diagram operace aktualizace dat v mezipaměti	62
3.10	Diagram operace připojení k Wi-Fi síti	62
3.11	Diagram operace start přístupového bodu	63
3.12	Diagram operace poslouchání na daném portu	64
3.13	Diagram operace získávání seznamu souborů měření	64
3.14	Diagram operace čtení části souboru	66
3.15	Diagram operace získávání informací o připojení	67
3.16	Diagram operace skenování sítí	68
3.17	Diagram řadiče měření	69
3.18	Vztah mezi základními typy časovačů	73
3.19	Rozložení pinů ESP-01	81
3.20	Schéma dataloggeru - připojení ESP-01	82
3.21	Průběh posílání souborů pomocí selektivního opakování	84
3.22	Schéma dataloggeru - napájení	85
3.23	Schéma dataloggeru - hlavní část	86
3.24	První verze desky plošných spojů	87
3.25	Druhá verze desky plošných spojů - neosazená	88
3.26	Druhá verze desky plošných spojů - osazená	89
4.1	Výpočet latence komunikace probíhající přes všechny uzly systému	95
4.2	Výpočet latence komunikace probíhající pouze přes Wi-Fi modul	95
4.3	Diagram opravy souborů	102
4.4	Ukázka hlavního okna programu	103
4.5	Ukázka okna správce dat	105
4.6	Ukázka okna detail záznamu	106
4.7	Ukázka uloženého grafu CPM ve formátu png	107
4.8	Ukázka okna grafické zobrazení dat	108
5.1	Vygenerovaný pulz gaussovského tvaru včetně spouště	111
5.2	Ukázka běhu simulátoru detektoru	113
5.3	Referenční spektrum změřené za použití spouště	116
5.4	Energetické spektrum změřené pomocí pokračovaného měření 1	117
5.5	Energetické spektrum změřené pomocí pokračovaného měření 2	117
5.6	Energetické spektrum změřené pomocí pokračovaného měření 3	118
5.7	Referenční energetické spektrum zářiče Kobalt-60	120
5.8	Změřené spektrum simulovaného zářiče Kobalt-60, zašuměné	121
5.9	Změřené spektrum simulovaného zářiče Kobalt-60, s aplikovaným filtrem	122
5.10	Neinvertující sčítací zesilovač	122
5.11	Změřené spektrum simulovaného zářiče Kobalt-60 posunuté o 150 kanálů doprava	123
5.12	Spektrum Kobaltu-60 změřené pomocí pokračovaného měření	124

5.13	Dialogové okno indikující provedené opravy souboru	127
5.14	Výchozí stav měření – všechny moduly jsou funkční	128
5.15	Status dataloggeru v obslužné aplikaci – Selhání modulu BME280	129
5.16	Status dataloggeru v obslužné aplikaci – Selhání Wi-Fi modulu, externí RTC a paměťové karty 1	130
5.17	Status dataloggeru v obslužné aplikaci – Odebrání zbylé paměťové karty	130
C.1	Ukázka prostředí, kde byl systém testován	149
D.1	Zapojení dataloggeru	152
D.2	Hlavní okno aplikace	154
D.3	Stav dataloggeru a jednotlivých modulů	155
D.4	Záložka Aktuální data	158
D.5	Záložka Soubory	159
D.6	Záložka Konfigurace sítí	160
D.7	Záložka Konfigurace dataloggeru	162
D.8	Správce dat	163
D.9	Okno grafického zobrazení dat	166

Seznam tabulek

2.1	Základní parametry STM32F103C8T6	24
2.2	Základní parametry STM32F401CCU6	25
2.3	Formát paketu request protokolu	31
2.4	Formát paketu request protokolu – nastavování statické IP adresy	32
2.5	Formát paketu request protokolu – konfigurace bezdrátové sítě	32
2.6	Formát paketu command protokolu	33
2.7	Formát paketu odpověď na příkaz (command response)	34
2.8	Stav vyřízení paketu command protokolu	34
2.9	Formát paketu UDPComm protokolu	36
2.10	Spotřeba jednotlivých modulů dataloggeru	37
3.1	Konfigurační registry	47
3.2	Konfigurace bezdrátových sítě	48
3.3	Formát záznamu	53
3.4	Priority přerušení	79
3.5	Cena jednotlivých součástí dataloggeru	90
5.1	Závislost délky prodlevy a počtu detekovaných impulzů	114
5.2	Pozorování vlivu délky prodlevy mezi impulzy na aliasing	119
5.3	Rychlost stahování souborů měření z dataloggeru	125
C.1	Parametry testovací sestavy	148

Úvod

Datalogger je zařízení, které sbírá binární, či analogová data ze senzorů a ukládá je na paměťové úložiště. Dataloggery mají široké uplatnění a liší se svojí složitostí a funkcionalitou. Dataloggery mohou být univerzální, či specifické pro určitou aplikaci. Jednoduché, které snímají data z jednoho senzoru, či složité snímající data z mnoha senzorů. Nejjednodušší datalogger je např. zařízení, které průběžně zaznamenává teplotu vzduchu. Příkladem složitějšího dataloggeru je černá skříňka v letadle, která je navržena takovým způsobem, aby přežila i extrémní podmínky.

Dataloggerem je možné zaznamenávat i ionizující záření. Kromě samotné intenzity ionizujícího záření, lze zaznamenávat i energie kvant záření. Zakreslením jednotlivých energetických hodnot do histogramu vznikne spektrum ionizujícího záření. Pomocí spektra ionizujícího záření lze identifikovat konkrétní radionuklid/y, což se může hodit pokud např. zkoumáme-li neznámé radioaktivní látky, či pokud měříme radioaktivní kontaminaci. Diplomová práce volně navazuje na bakalářskou práci, kde byla měřena intenzita ionizujícího záření. Výsledné zařízení z bakalářské práce ale neumělo dlouhodobě sbírat data a nebylo schopné měřit energii ionizujícího záření.

Úkolem diplomové práce je vytvořit datalogger, který je specializovaný na měření energetického spektra ionizujícího záření. Datalogger je možné ovládat bezdrátově pomocí Wi-Fi přes obslužnou aplikaci pro osobní počítač s Windows 10. Výhodou ovládání dataloggeru přes Wi-Fi je to, že může být umístěn na špatně přístupném místě. Cíle práce jsou následující:

- Analýza možnosti měření či simulace měření ionizujícího záření a požadavků na datalogger.
- Návrh a realizace dataloggeru. Součástí realizace je nejen samotný firmware pro mikrokontrolér, ale i návrh desky plošných spojů.
- Návrh a realizace obslužné počítačové aplikace pro Windows 10.

- Testování dataloggeru a obslužné počítačové aplikace.

Práce se skládá z 5 částí – Analýza, Návrh, Realizace dataloggeru, Realizace počítačové aplikace a Testování systému. Kapitola Analýza se zabývá lehkým fyzikálně-chemickým úvodem, nutným pro pochopení měření spektra ionizujícího záření. Dále se zabývá detektory pro měření ionizujícího záření (scintilační detektor) a možnostmi simulace detektoru ionizujícího záření. Součástí analýzy je i popis komerčního emulátoru detektoru.

Kapitola Návrh se zabývá výběrem jednotlivých modulů pro úspěšnou realizaci dataloggeru. Kromě samotného výběru modulů se v této kapitole popisují i navržené protokoly, které využívají ke komunikaci jednotlivé prvky systému. V kapitole Návrh se také popisuje výběr knihoven pro realizaci obslužné počítačové aplikace, či plán realizace diplomové práce.

Další kapitolou je Realizace dataloggeru, která podrobně popisuje řešení klíčových částí firmwaru pro mikrokontrolér STM32. Kromě popisu řešení firmwaru pro STM32 se v této kapitole také píše o realizaci firmwaru pro Wi-Fi modul ESP-01. Součástí kapitoly Realizace dataloggeru je také popis výroby desky plošných spojů od zakreslení schématu dataloggeru až po výsledné osazení desky součástkami.

Čtvrtá kapitola – Realizace počítačové aplikace podrobně popisuje, jakým stylem byla realizována obslužná počítačová aplikace pro Windows 10. Podobně jako v případě kapitoly Realizace dataloggeru se zde popisuje řešení klíčových částí počítačové aplikace. Kromě realizace důležitých tříd je zde i blíže diskutována realizace grafického uživatelského rozhraní.

Poslední kapitola Testování systému se zabývá testováním dataloggeru a obslužné počítačové aplikace. Nejprve je popsána realizace simulátoru detektoru ionizujícího záření, který je vypracován na základě analýzy. Poté je otestováno jakým způsobem datalogger zpracovává analogový signál ze simulátoru. Na závěr je důkladně otestována obslužná počítačová aplikace a datalogger v testu jednotlivých funkcionalit.

Analýza

1.1 Ionizující záření

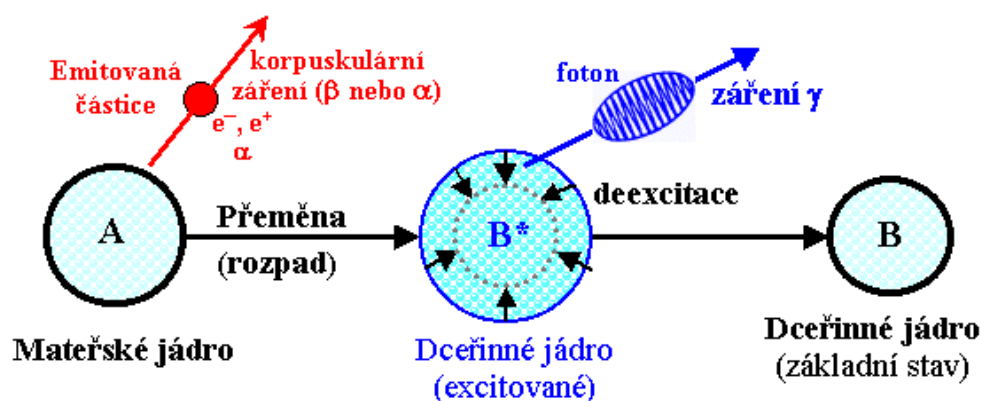
Ionizující záření vystihuje definice ze zdroje [1], která zní následovně: „*Ionizujícím zářením nazýváme takové záření, jehož kvanta mají natolik vysokou energii, že jsou schopna vyrážet elektrony z atomového obalu a tím látku ionizovat.*“

Všechny druhy ionizujícího záření mají společnou vlastnost a tou je ionizace atomů způsobená tím, že se z atomu vyráží elektron. Ionizace znamená, že se z neutrálních atomů stává elektricky nabitá částice – iont [2]. Způsob jakým může ionizační záření interagovat s hmotou se liší druh od druhu ionizujícího záření. Rozlišujeme dva druhy mechanismů interakce záření s hmotou:

- Přímé ionizující záření
- Nepřímé ionizující záření

Přímé ionizující záření v sobě nese elektrický náboj a je schopné přímo vyrážet z atomů elektrony. Jedná se např. o záření alfa, beta plus a beta minus. Nepřímé ionizující záření nenesé žádný náboj a je tedy nenabité. Tento typ ionizující záření předává svou kinetickou energii nabitým částicím v atomu (elektrony, v některých případech atomové jádra) a ty až látku ionizují. Do nepřímého ionizujícího záření patří např. gama záření, nebo rentgenové záření.

Další způsob, jakým se ionizující záření rozděluje je ten, jestli se jedná o záření vlnové, či korpuskulární. Kvanta vlnového záření mají nulovou klidovou hmotnost, pohybují se rychlostí světla a pokud se zabrzdí, odevzdají veškerou svoji energii a zaniknou. Jedná se o elektromagnetické záření, jehož kvanta jsou fotony. Do tohoto typu záření patří např. gama záření, či rentgenové záření. Kvanta korpuskulárního záření mají naopak nenulovou klidovou hmotnost, pokud jsou zabržděny, zachovávají si svoji existenci. Pohybují se rychlosti nižší, než je rychlost světla. Jedná se o proud částic např. jádra hélia (alfa záření), elektrony (záření beta minus), nebo pozitrony (záření beta plus).



Obrázek 1.1: Schéma vzniku gama záření [5].

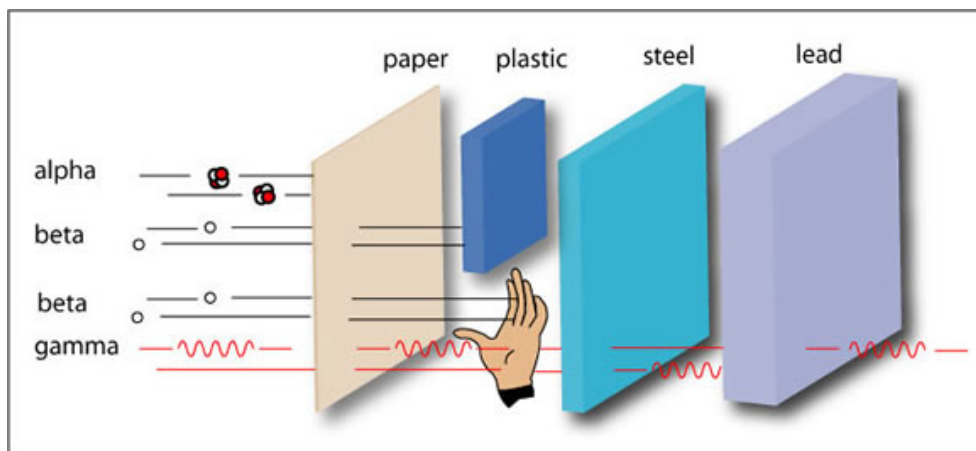
V diplomové práci je blíže diskutováno pouze fotonové záření, konkrétně záření γ a rentgenové záření. Informace o částicových záření alfa, beta jsou k dispozici ve volně dostupných zdrojích [3], [4], [5], [6], [7].

Zdroje ionizujícího záření jsou označovány jako zářič. Zářiče je možné rozdělit buď na přírodní a umělé, nebo podle mechanismu vzniku záření. Více informací o ionizujícím záření je dostupných na webové stránce [1], odkud byly čerpány veškeré informace uvedené v předchozích odstavcích.

1.1.1 Fotonové záření (gama, X)

Zástupcem fotonového záření je gama (γ) záření a rentgenové záření, které se dříve nazývalo záření X. Gama záření je vysokoenergetické elektromagnetické záření, jehož kvanta jsou fotony, které mají vyšší energii než 10 keV. Gama záření vzniká během alfa, nebo beta přeměny, protože po této přeměně může být jádro v excitovaném stavu (jádro není v energeticky základní hodnotě). Přebytková energie se vyzáří vysokoenergetickým elektromagnetickým zářením (kvantem fotonů) a jádro se tím dostane do základního energetického stavu. Vyprodukované elektromagnetické záření odpovídá gama záření. Proces vzniku gama záření popisuje Obrázek 1.1. Gama záření nemění protonové číslo atomu.

Stejně jako alfa záření i gama záření má diskrétní energetické spektrum. Jádro, ale může mít více excitovaných hladin, spektrum proto může obsahovat více diskrétních čar. Je to z důvodu, že se emituje více monoenergetických gama fotonů, které mají různou energii. Jak ukazuje Obrázek 1.2, gama záření je velmi pronikavé. Ke stínění se využívají materiály, které mají vyšší protonové číslo a vysokou hustotu. Využívá se např. olovo či beton. Pokud není možné zářič stínit, je dobré si od zářiče udržovat maximální vzdálenost. Intenzita gama záření nepřímo úměrně klesá s druhou mocninou vzdálenosti



Obrázek 1.2: Pronikavost alfa, beta a gama záření [9].

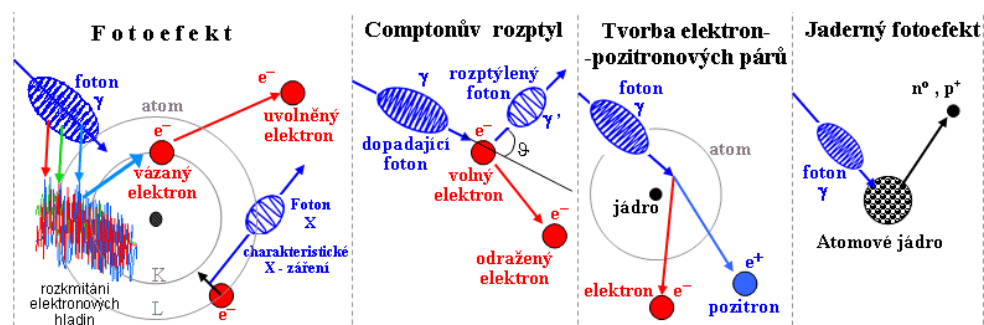
[8].

Rentgenové záření má stejnou fyzikální povahu jako gama záření. Dokonce do spektra gama záření zasahuje i velmi tvrdé rentgenové záření. Gama a X záření proto rozlišujeme podle jejich původu. Gama záření vzniká v atomových jádrech. Rentgenové záření vzniká v atomovém obalu, či je to brzdné záření elektronů. Více informací o gama záření a záření X je na webových stránkách [5], [10], [11].

1.1.2 Interakce gama záření s látkou

Fotony gama záření nemají elektrický náboj, nemohou proto přímo ionizovat atomy. Foton je ale kvantem rychle kmitajícího elektrického pole, když se do blízkosti tohoto pole přiblíží elektron, může obdržet energii fotonu. Interakce gama záření s látkou může probíhat čtyřmi způsoby, které zobrazuje Obrázek 1.3. Prvním způsobem interakce gama záření s látkou je fotoelektrický jev (fotoefekt). Během fotoefektu foton interaguje s elektronem, který je vázaný v atomovém obalu. Foton předá elektronu veškerou svojí energii a zanikne. Zasažený elektron se uvolní z atomového obalu. Kinetická energie uvolněného elektronu je dána rozdílem energie fotonu a vazebnou energií elektronu v atomu. Aby bylo uvolnění elektronu možné, dopadající foton musí mít vyšší energii než je vazební energie elektronu na příslušné slupce. K fotoefektu dochází hlavně u rentgenového záření a gama záření s energií pod 50 keV.

Dalším způsobem interakce je Comptonův rozptyl, ke kterému dochází pokud se foton srazí s volným elektronem, či slabě vázaným elektronem. Foton předá elektronu pouze část svojí energie. Od tohoto elektronu se foton odrazí, změní svůj směr pohybu a ztratí část energie, kterou předal elektronu. Pro



Obrázek 1.3: Interakce gama záření s látkou [5].

výpočet energie odraženého (rozptýleného) fotonu platí následující vzorec:

$$E_{\gamma'} = \frac{E_{\gamma}}{1 + \left(\frac{E_{\gamma}}{m_{0e}c^2}\right) * (1 - \cos\theta)} \quad (1.1)$$

$E_{\gamma'}$ odpovídá energii rozptýleného fotonu, $m_{0e}c$ je klidová hmotnost elektronu, θ je úhel mezi dopadajícím a rozptýleným fotonem. Čím je úhel větší, tím je větší energie, kterou foton ztratí. K největší ztrátě energie fotonu dochází při zpětném rozptylu tedy $\theta = 180^\circ$. Comptonův rozptyl je typický pro fotony o energiích v rozsahu 100 keV až 10 MeV a může se v látce několikrát opakovat, dokud foton neopustí látku, či nezanikne fotoefektem.

Třetím způsobem interakce je tvorba elektron-pozitronových párů, který nastává při průletu fotonu v dosahu coulombické síly jádra, kdy se foton přeměňuje na dvojici částic pozitron a elektron. Aby se vytvořil elektron-pozitronový pár je potřeba, aby foton měl vyšší energii než 1,022 MeV, která se využívá na tvorbu páru, přebývající energie se změní v kinetickou energii elektronu a pozitronu. Elektron v jádře zůstává a pozitron anihiluje s volným elektronem za vzniku dvou gama fotonů o energii 511 keV. Tvorba elektron-pozitronových párů se objevuje při vysokých energiích gama záření u látek s vysokým protonovým číslem.

Posledním způsobem interakce je jaderný fotoefekt, který nastává pokud má gama záření větší energii, než je vazební energie nukleonů v jádře. Může nastat jaderná reakce v jádře, při níž je vyražen neutron či proton. Potřebná energie pro fotojadernou reakci je daná každým jádrem. Nejjednodušší fotojadernou reakcí je vyražení neutronu z deuteria, který má prahovou energii 2,23 MeV. Těžší jádra vyžadují vyšší energie než 8 MeV. Zasažené jádro může být po fotojaderné reakci radioaktivní, dochází ke gama aktivaci. Jaderný fotoefekt není pro účely této diplomové práce podstatný a jedná se o zajímavost.

Rentgenové záření může s látkou reagovat fotoefektem, či Comptonovým rozptylem. Pro ostatní interakce s látkou má typicky rentgenové záření nižší

hodnoty energií, než je potřeba. Více informací o interakci gama a X záření je k nalezení zde [5], [11].

1.2 Count rate

Count rate je veličinou se kterou pracují Geiger-Müllerovy čítače. Count rate udává kolik částic/fotonů Geiger-Müllerův čítač detekoval za časový interval. Jednotkou je buď počet detekovaných událostí za sekundu (CPS), či počet detekovaných událostí za minutu (CPM). Count rate nic neříká o energii záření a hodnoty CPS/CPM jsou různá pro každou trubici. Pomocí CPM/CPS lze vypočítat i dávkový ekvivalent, avšak je nutné provést kalibraci s referenčním zářičem. Více informací o count rate zde [12].

1.3 Elektronvolt

V určování spektra ionizujícího záření se zajímáme o energii fotonu/částice. SI jednotkou energie je Joule. V jaderné fyzice je ale jednotka joule příliš velká, z tohoto důvodu se používá jednotka, která není v soustavě SI – elektronvolt (zkratka eV). Jeden elektronvolt odpovídá kinetické energii, kterou získá elektron urychlený ve vakuu napětím jednoho voltu [13]. Pro převod elektronvoltage na joule platí následující vztah:

$$1 \text{ eV} = 1,602176634 * 10^{-19} \text{ J} \quad (1.2)$$

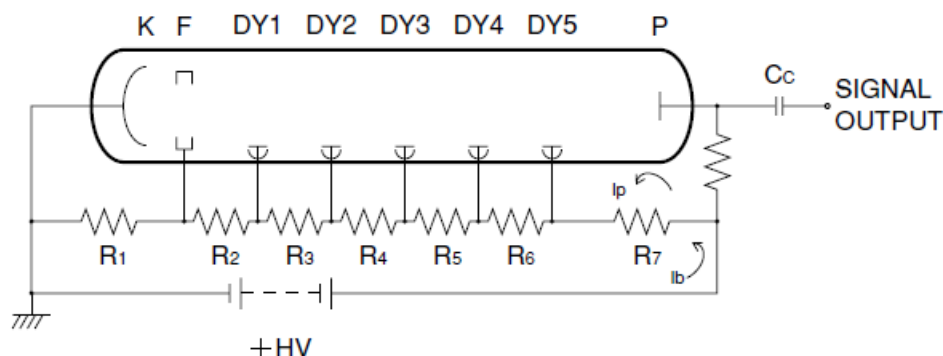
Jednotka elektronvolt je však velmi malá, proto se v praxi používají její násobky – 1 keV odpovídající tisíci eV, či 1 MeV odpovídající milionu eV. Lze použít i větší násobky jako GeV (miliarda eV), či TeV (bilion eV), avšak pro účely této diplomové práce se jedná o příliš velké energie. Více informací o elektronvoltage je k dispozici ve volně dostupných zdrojích [13].

1.4 Spektroskopie ionizujícího záření

Ve spektroskopii ionizujícího záření se využívá toho, že kromě samotné intenzity záření se měří i energie jednotlivých kvant [14]. Poté se zaznamenané hodnoty vynáší na graf, kam se na vodorovnou osu vynáší energie a na svislou osu relativní četnost kvant příslušné energie. Tento graf se nazývá spektrem ionizujícího záření. K měření energií kvant je potřeba mít vhodný detektor. Mezi nejběžnější detektory patří scintilační detektor a polovodičový detektor.

1.4.1 Scintilační detektor

Scintilační detektor je založen na principu, že některé látky reagují na pohlcení kvant ionizujícího záření tím, že vytváří světelné záblesky (latinsky *scintilla*).

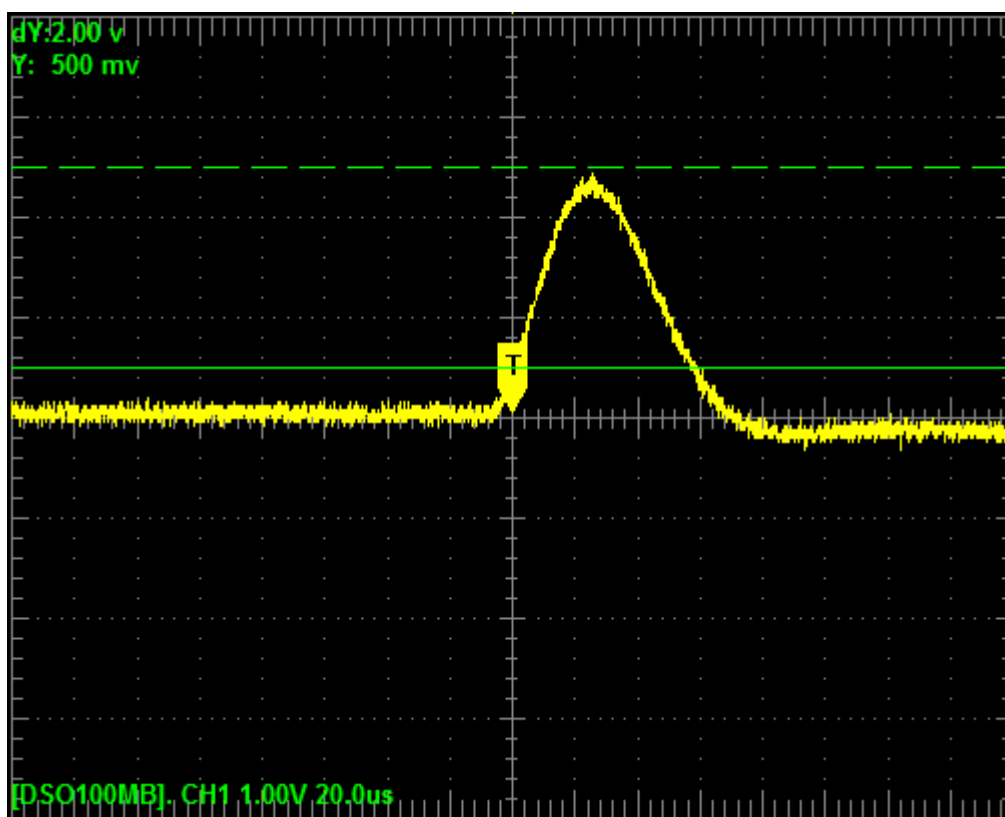


Obrázek 1.4: Jedna z možných variant zapojení fotonásobiče [15].

Záblesky se registrují vhodnou elektronikou – fotonásobičem. Látka, která je schopná kvanta ionizujícího záření převést na záblesky se nazývá scintilátor (či scintilační krystal). Scintilátor může být anorganický krystal, organické plastické materiály, kapalně roztoky organických látek, nebo i vzácné plyny. Nejčastěji se používá krystal jodidu sodného aktivovaného thaliem – NaI(Tl).

Záblesky ze scintilátoru jsou světlovdným kontaktem přivedeny na fotokatodu fotonásobič. Fotonásobič je schopný převést i slabý světelný signál na detekovatelný elektrický impulz. Klasické fotonásobiče jsou vakuové skleněné trubice, které obsahují fotokatodu, fotoanodu a několik dynod. Na každou dynodu je přivedeno napětí několik set voltů. Napětí na každé dynodě je ale jiné, toho se docílí pomocí napěťového děliče. Ke správné funkci klasického fotonásobiče je tedy potřeba vysokonapěťový zdroj, který musí mít stabilní napětí. Ke všemu se jedná o velmi citlivou součástku, aby nedošlo ke zničení fotonásobiče musí být během jeho provozu zaručena světlotěsnost. Kromě klasických fotonásobičů existují i polovodičové násobiče, které jsou kompaktnější, odolnější a nevyžadují takové napětí jako klasické fotonásobiče. Scintilační sondou se nazývá taková součástka, která obsahuje scintilační krystal, fotonásobič a odporový dělič pro napájení dynod. Scintilační sonda je umístěna ve světlotěsném pouzdře. Schéma scintilační sondy je na Obrázku 1.4. Více informací o scintilačních detektorech zde [14].

Signál vycházející z fotonásobiče je krátký (stovky nanosekund) a má omezenou amplitudu (desítky až stovky milivoltů). Proto se využívají zesilovače, který signál z fotonásobiče převedou na impulz, který je lépe čitelný pomocí A/D převodníku. První zesilovač je *Charge-sensitive Preamplifier* (nábojově citlivý předzesilovač), který integruje hodnotu proudu a produkuje na výstup napětí, jehož amplituda odpovídá intenzitě proudu. Dalším zesilovačem je *Shaping Amplifier*, který impulz převede do požadovaného tvaru. Nejčastěji



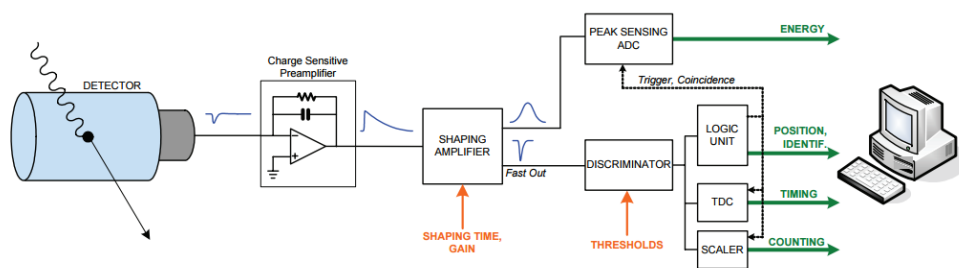
Obrázek 1.5: Impulz gaussovského tvaru [15].

se využívá impulz gaussovského tvaru, který má uživatelem definované vlastnosti jako je délka pulzu. Impulz gaussovského tvaru ukazuje Obrázek 1.5. Vytvarovaný signál se poté již může poslat do zařízení, které snímá maximální hodnotu napětí impulzu pomocí A/D převodníku. Obvod zpracování signálu z fotonásobiče popisuje Obrázek 1.6. Více informací o zpracování signálu z fotonásobiče zde [15].

1.4.2 Zpracování signálu z detektoru

Vytvarovaný signál z detektoru je veden na A/D převodník počítače, či mikrokontroléru, který z napětí udělá digitální hodnotu. Tato digitální hodnota odpovídá indexu pole. Z indexu se získá přesná adresa v paměti, jejíž hodnota je inkrementována o jedna. Takový počítač/mikrokontrolér má roli mnohokanálového analyzátoru [14]. Počet kanálů je dán rozlišením A/D převodníku.

Zapojení detektoru do mikrokontroléru není problém, protože většina mikrokontroléru obsahuje A/D převodník. V případě zapojení detektoru do počítače se může využít zvuková karta, která v sobě má zabudovaný A/D pře-



Obrázek 1.6: Obvod na zpracování pulzu z fotonásobiče [15].

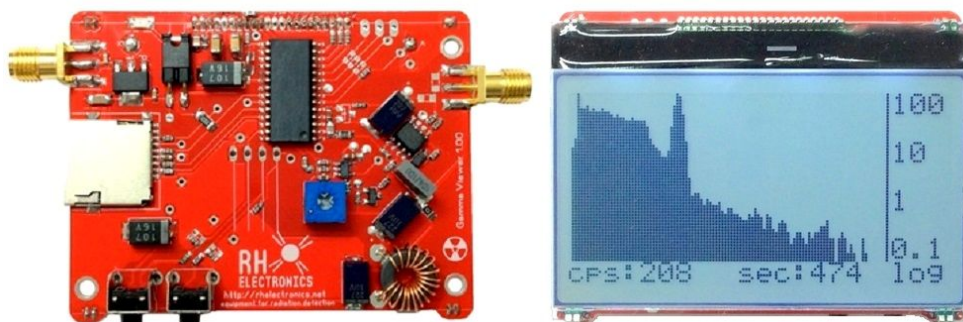
vodník. K tomuto účelu byl vytvořen i software – Theremino MCA. Pomocí tohoto programu je možné měřit a zobrazovat jednotlivé pulzy přicházející z detektoru. Program umí i pulzy zpracovávat a vytvářet energetické spektrum ionizačního záření. Více informací o Theremino MCA je k dispozici na stránkách softwaru [16].

1.4.3 Existující řešení pro zpracování signálu z detektoru

Pokud není možné využít zvukovou kartu ke zpracování signálu z detektoru, existuje komerční řešení, které zvukovou kartu nahrazuje. Jedná se o zařízení s jménem GS-USB-PRO od firmy GammaSpectacular. Zařízení se k počítači připojuje pomocí USB a obsahuje zabudovaný audio kodek. Kromě toho obsahuje i nastavitelný vysokonapěťový zdroj v rozsahu 0–2000 V. Nejedná se o datalogger, ale pouze adaptér pro detektor. Aby bylo možné data dlouhodobě sbírat musí být zařízení připojené k počítači. Cena zařízení je přibližně 9000,-Kč bez DPH. Více informací o GS-USB-PRO zde [17].

Dalším komerčním zařízením je Quantum MCA2500R, které zpracovává signál pomocí 50 MHz 14bitového A/D převodníku. MCA2500R podporuje rozlišení až 1024 kanálů, maximální počet pulzů na kanál je $2^{31} - 1$ (více než 2 miliardy). Maximální propustnost systému je více než 75000 CPS. Podobně jako GS-USB-PRO i MCA2500R obsahuje vysokonapěťový zdroj s rozsahem 0–1200 V. Zařízení se k počítači připojuje buď přes RS-232 (sériová linka), či přes Ethernet. Zařízení obsahuje non-volatilní paměť, ale pouze pro uložení nastavení. Spektrum se nejspíše ukládá do volatilní paměti, takže pro dlouhodobější sběr dat je nutné zajistit, že nedojde k výpadku napájení, či resetu zařízení. Cena zařízení je neznámá. Více informací o Quantum MCA2500R zde [18].

Komerční zařízení, které splňuje parametry dataloggeru, je DIY PIC18 MCA Module for Gamma Spectroscopy od RHElectronics (je zobrazeno na Obrázku 1.7). Zařízení je miniaturní a jako mikrokontrolér využívá PIC18. Ke zpracování analogového signálu se využívá 10bitový A/D převodník. Rozsah



Obrázek 1.7: Existující řešení pro měření spektra ionizujícího záření – Řešení od RHElectronics [19].

měřeného spektra je 1024 kanálů. Požadovaná amplituda signálu je 20–3000 mV a impuls musí mít minimální náběžný čas 2–10 μs a sestupný čas 30–50 μs . Count rate by neměl převyšovat 1000 CPS. Narozdíl od předchozích existujících řešení, toto zařízení neobsahuje vysokonapěťový zdroj. Naměřená data se ukládají na microSD paměťovou kartu. Zařízení obsahuje 128x64 podsvícený LCD displej, na kterém je možné zobrazit spektrum. Zařízení se k počítači připojuje UART-USB převodník. Jedná se o zařízení určené pro amatérskou gama spektroskopii, odpovídá tomu i nízká cena přibližně 1750,-Kč bez DPH. Více informací o DIY PIC18 MCA Module for Gamma Spectroscopy zde [19].

1.4.4 Požadavky na datalogger

Datalogger musí umět zpracovávat analogový signál z detektoru ionizujícího záření. V signálu jsou důležitá jeho maxima, které určují energii zachyceného kvanta záření. Jeden z požadavků na datalogger je tudíž A/D převodník, který snímá maximální hodnoty signálu. A/D převodník by měl mít dostatečné rozlišení, zařízení DIY PIC18 MCA Module obsahuje 10bitový A/D převodník. Minimální rozlišení A/D převodníku by mělo být minimálně 10 bitů, aby realizovaný datalogger dosahoval alespoň takové přesnosti jako nejlevnější nalezené komerční řešení. Rozlišení A/D převodníku udává maximální počet kanálů. Pro 10 bitů je maximální počet kanálů 2^{10} , tedy 1024 kanálů.

Aby byl signál z detektoru dobře interpretován, je nutné číst jeho maximální amplitudu. Prvním způsobem, jakým je možné číst maximální napětí, je průběžné vzorkování signálu a hledání maxim. Rychlost vzorkování určuje, jak moc rychlé impulsy umí datalogger detekovat. Ke zpracování signálu z detektoru se může využívat i zvuková karta. Běžná vzorkovací frekvence zvuku je 44100 Hz [20], A/D převodník dataloggeru by tedy měl mít vzorkovací rychlost alespoň 44100 vzorků/s.

Další možný způsob, jakým je možné číst maximální napětí signálu, je pomocí „spouště“. Spoušť je digitální signál, který určuje, kdy může datalogger číst analogovou hodnotu. Spoušť je nutné generovat pomocí externího obvodu. Konverze A/D převodníku musí být spuštěna neprodleně po spuštění spouště. Nejrychlejší zpracování zajišťuje přerušování s vysokou prioritou od pinu, ke kterému je připojen signál spouště. Mikrokontrolér dataloggeru proto musí disponovat pinu, které podporují externí přerušování.

Spektrum je potřeba za běhu udržovat ve volatilní paměti (RAM). Datalogger proto musí mít k dispozici dostatečnou RAM. Každý kanál zabere 4 bajty – maximální počet pulzů na kanál je více než 4 miliardy. Paměť v bajtech pro uložení celého spektra je daná počtem kanálů vynásobeno hodnotou 4. Během výběru mikrokontroléru je nutné brát zřetel kromě na rozlišení A/D převodníku i na velikost RAM. Zvýšení přesnosti A/D převodníku o jeden bit znamená zdvojnásobení požadavků na RAM.

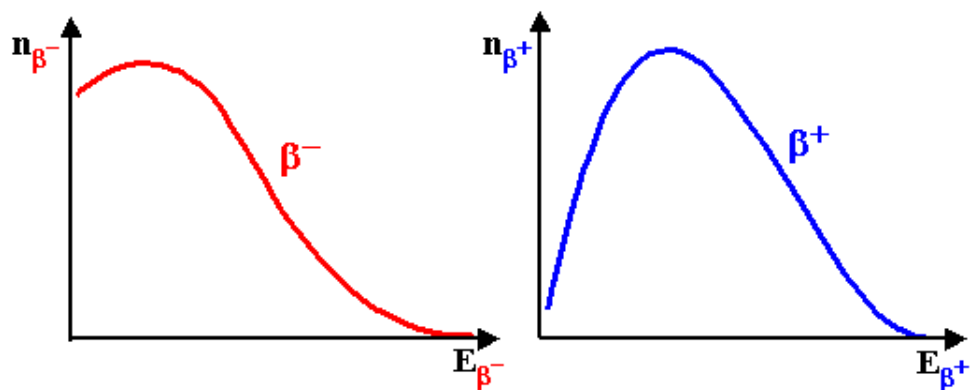
1.5 Spektrum ionizujícího záření

Veškeré informace o spektru ionizujícího záření byly čerpány z volně dostupných zdrojů [5]. Většina radionuklidů emituje záření, jehož kvanta mají různé energie. Energie záření je charakterizována energetickým spektrem, které udává četnost energetických hodnot. Energie záření se zakresluje do histogramu, kde se udává relativní četnost jednotlivých energetických hodnot. Na vodorovnou osu se zanáší energie a na svislou osu četnost kvant s touto energií. Pokud není k dispozici oceňovaný detektor, tak se na vodorovnou osu místo energie zanáší jednotlivé kanály a na svislou osu počet detekovaných impulzů odpovídající příslušnému kanálu.

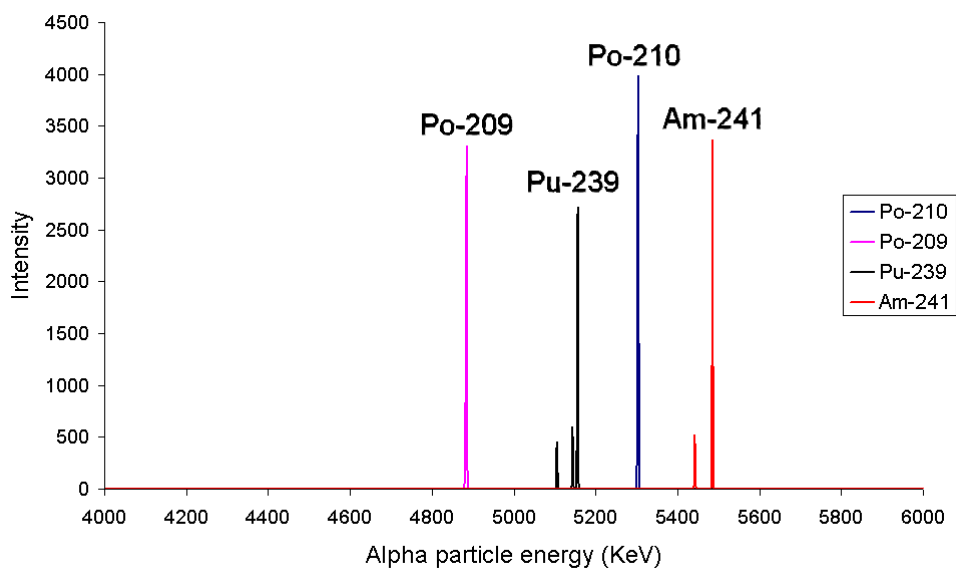
Spektrum ionizujícího záření může být buď spojité, či diskrétní (čárové). Spojité spektrum popisuje Obrázek 1.8. Spojité spektrum mají radionuklidy, které emitují kvanta částic o všech energiích v určitém intervalu. Interval je charakteristický pro každý druh radionuklidu. Pokud se četnost jednotlivých energií zakreslí do grafu vychází spojitá křivka. Spojité spektrum mají např. beta zářiče.

Diskrétní (čárové) spektrum mají zářiče emitující kvanta záření o jedné nebo více určitých energiích. Energie je charakteristická pro každý radionuklid. Graf diskrétního spektra obsahuje vrcholy v přesně definovaných energiích, tyto vrcholy se označují jako píky. Mimo píky jsou hodnoty nižší než v píkách. Diskrétní spektrum zobrazuje Obrázek 1.9. Diskrétní spektrum je typické pro alfa, nebo gama zářiče.

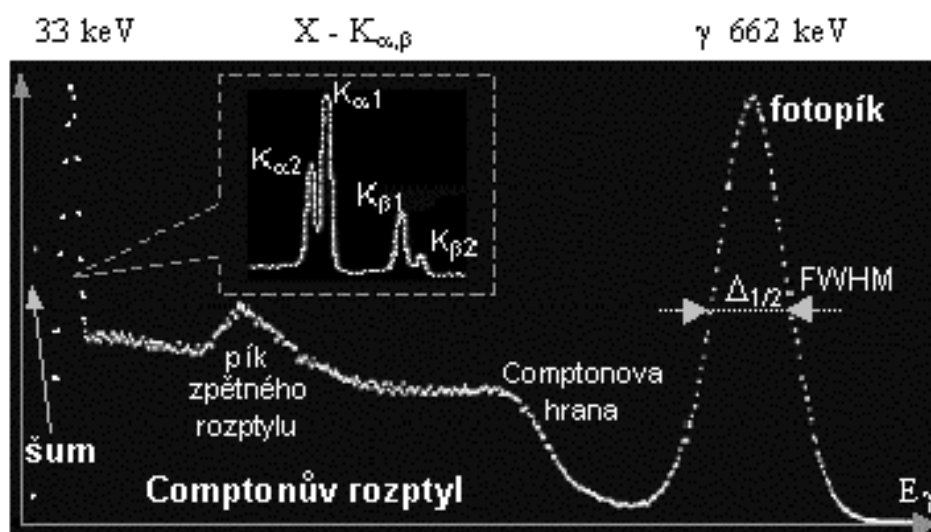
V diplomové práci se zaměřuji konkrétně na gama spektroskopii pomocí scintilačního detektoru. Zaměřím se tedy právě na diskrétní energetické spektra gama záření. Jak bylo napsáno v minulém odstavci, diskrétní spektrum záření by mělo mít výrazné píky v energiích, které charakterizují konkrétní gama zářič. Např. gama zářič Cesium-137 emituje gama záření s energií 0,6617



Obrázek 1.8: Typický tvar spojitého spektra beta mínus a beta plus záření [5].



Obrázek 1.9: Typický tvar diskrétního spektra alfa zářičů [21].



Obrázek 1.10: Scintilační spektrum Cesia-137 [14].

MeV [22]. Očekávané spektrum by mělo tedy mít tvar, že je jediný ostrý pík v energii 0,6617 MeV, na jiných místech jsou hodnoty blízké nule. Spektrum naměřené scintilačním detektorem ale vypadá úplně jinak a popisuje ho Obrázek 1.10. Aby bylo možné spektrum pochopit je nutné znát, jak gama záření interaguje s látkou (viz 1.1.2 Interakce gama záření s látkou).

Na Obrázku 1.10 je jeden výrazný vrchol (pík) v hodnotě 662 keV, který se nazývá fotopík. Fotopík odpovídá gama fotonům, které krystalu odevzdaly veškerou svojí energii buď fotoefektem, vícenásobným comptonovským rozptylem, či kombinací obojího. Gama záření má diskrétní spektrum, očekávaný tvar fotopíku je jedna čára v hodnotě 662 keV. Skutečný tvar fotopíku avšak připomíná gaussovu křivku. Je to z několika různých důvodů. Jedním z důvodů je to, že pokud ke scintilaci dojde v periférní části krystalu, tak fotonásobič zachytí menší počet fotonů, než kdyby ke scintilaci došlo ve středu krystalu blízko fotokatody. Další z důvodů může být nelinearita světelného výtěžku scintilátoru, či statistická fluktuace v počtu uvolněných fotonů v krystalu. Další důvody jsou blíže popsány zde [14].

Čím je fotopík širší, tím je nižší energetické rozlišení detektoru, protože dochází ke spojování fotopíků různých energií. Pro energetické rozlišení detektoru se používá pološířka fotopíku (FWHM), která udává šířku v poloviční výšce píku. Rozlišení je možné vyjádřit buď absolutně v keV, nebo relativně jako podíl pološířky k hodnotě energie středu fotopíku. Je zvykem používat jako energii středu fotopíku hodnotu pro radionuklid Cesium-137 (662 keV).

Nalevo od fotopíku je spojitě Comptonovsky rozptýlené spektrum, které je zprava ohraničeno Comptonovou hranou. Comptonovské rozptýlené spektrum

odpovídá gama fotonům, které v krystalu ztratily jen část své energie interakcí zvanou Comptonovský rozptyl. Comptonova hrana odpovídá maximální hodnotě, kterou může foton Comptonovským rozptylem elektronu předat – dochází tak při zpětném rozptylu. Pokud dojde k vícenásobnému comptonovskému rozptylu, tak je krystalu předána vyšší energie, která může zasahovat i do fotopíku a působit rušivě. Další pík, který lze na Obrázku 1.10 pozorovat je pík zpětného rozptylu. Zpětný rozptyl nastává tehdy, když foton ztratí část svojí energie v okolním materiálu a až následně je detekován.

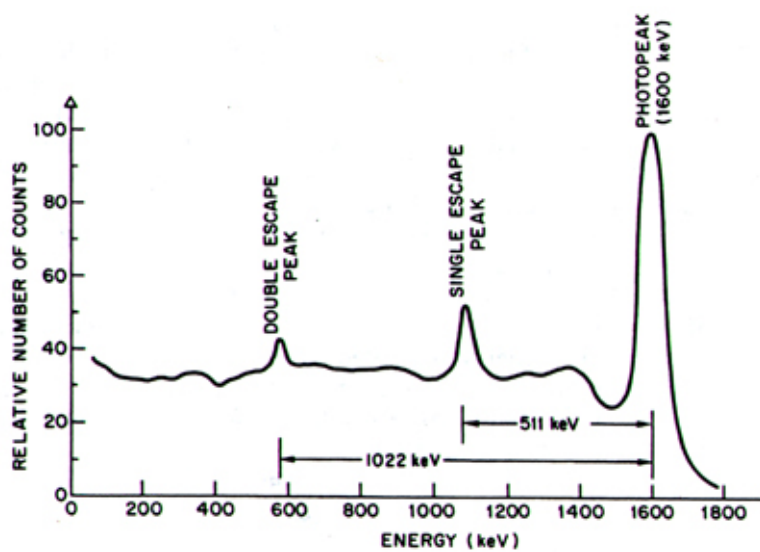
Posledním, avšak výrazným píkem je šum, který se nachází v nízkých energetických hodnotách. Šum může nastávat v elektrických obvodech detektoru, ale i ve fotonásobiči. Šum komplikuje měření nízkenergetických záření. Šum ale lze zredukovat např. chlazením fotonásobiče nebo lepším odstíněním elektroniky.

Spektrum zkresluje i tvorba pozitron-elektronových párů. Nutná podmínka k vytvoření pozitron-elektronových párů je minimální energie 1022 keV. Fotony Cesia-137 mají nedostatečnou energii ke tvorbě elektron-pozitronových párů. Aby se projevila tvorba elektron-pozitronových párů, je potřeba prozkoumat spektrum zářiče, jehož kvanta záření mají vyšší energii než 1022 keV. Obrázek 1.11 zobrazuje spektrum gama zářiče, který má fotopík v hodnotě 1600 keV. Fotony interagují s detektorem tvorbou pozitron-elektronových párů. Pozitron se anihiluje s elektronem a během anihilace se vytvoří dva fotony o energii 511 keV. Pokud jsou oba anihilační fotony detekovány úplnou absorbcí, přispívají fotopíku. Může ale nastat to, že jeden, nebo oba fotony z detektoru uniknou. Pokud unikne jeden foton, je detektorem zaznamenaná energie snížena o energii anihilačního fotonu (511 keV), který unikl z detektoru bez detekce. V oblasti fotopíku mínus 511 keV se objevuje únikový pík (*single escape peak*). Pokud uniknou oba anihilační fotony, tak je zaznamenaná energie snížena o energii dvou anihilačních fotonů (1022 keV). V oblasti fotopíku mínus 1022 keV se objevuje dvojitý únikový pík (*double escape peak*).

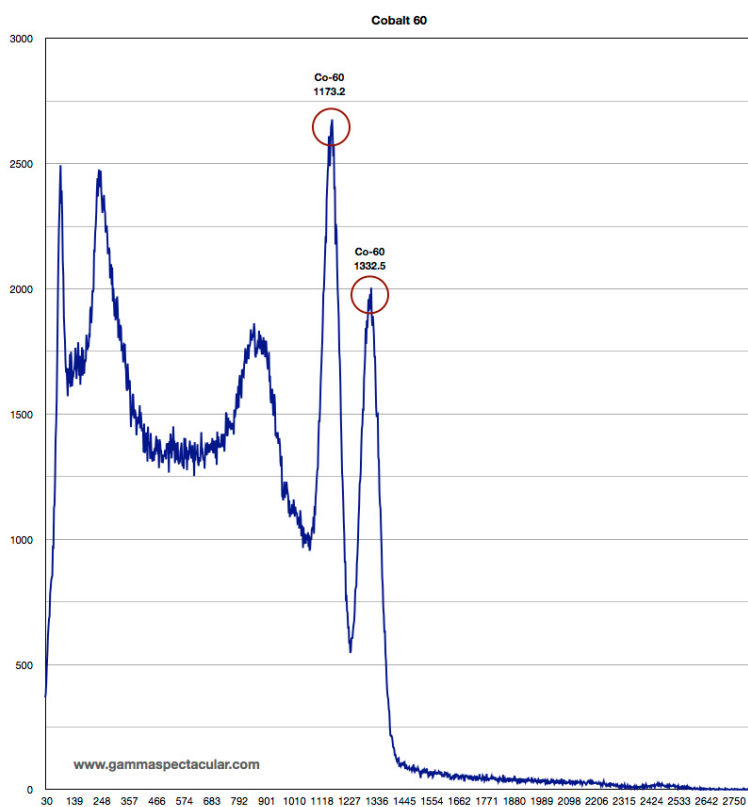
Kromě toho se může ve spektru projevit i *Pile-up efekt*. K Pile-up efektu dochází, když do scintilátoru vlétnou dvě kvanta záření krátce za sebou. Detektor energii těchto dvou kvant sečte. Pokud k Pile-up efektu dochází často, může se vytvořit pík napravo od fotopíku.

Některé radionuklidy emitují záření, které má více energetických hodnot. Takovým příkladem je Kobalt-60, který emituje gama záření s energií 1,1732 MeV a 1.3325 MeV [24]. Dobrý detektor tyto dvě energie umí rozlišit a ve spektru se zobrazí dva výrazné fotopíky. Tuto skutečnost ukazuje Obrázek 1.12. Více informací o vlastnostech spektra gama záření získané pomocí scintilačního detektoru zde [14].

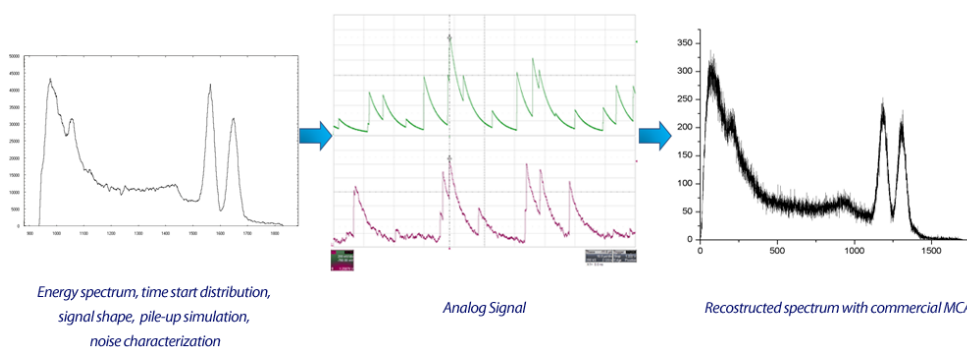
1. ANALÝZA



Obrázek 1.11: Scintilační spektrum neznámého gama zářiče [23].



Obrázek 1.12: Scintilační spektrum Kobaltu-60 [25].



Obrázek 1.13: Generování signálu podle spektra pomocí emulátoru CAEN [26].

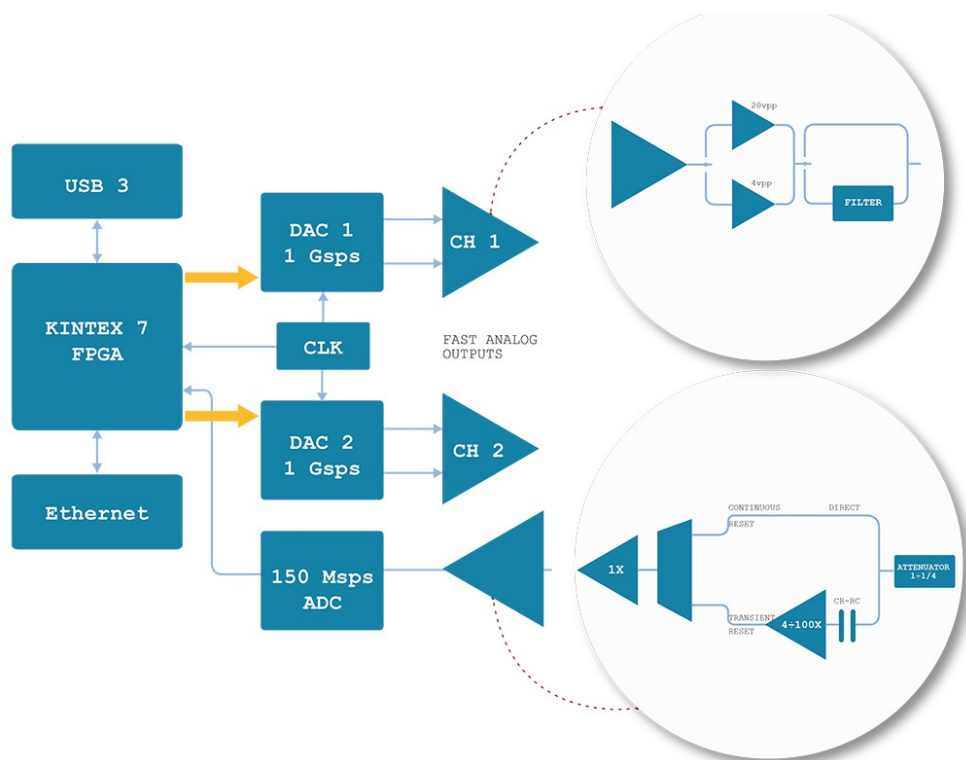
1.6 Simulace detektoru ionizujícího záření

Výsledné zařízení je nutné otestovat. Testování může probíhat buď pomocí skutečného detektoru záření a zářiče, nebo pomocí simulátoru, který simuluje chování detektoru. Realizace skutečného detektoru vyžaduje méně obvyklé a drahé součástky jako fotonásobič a scintilační krystal. Ke správnému chodu scintilační sondy je potřeba i stabilní vysokonapěťový zdroj. Kromě toho je během realizace potřeba mít k dispozici zářič, který může být při špatné manipulaci a skladování nebezpečný.

Kvůli těmto důvodům se k testování používá simulátor detektoru záření. Existuje komerční řešení od firmy CAEN, které nabízí emulaci detektoru záření. Hlavní funkcionalitou tohoto zařízení je generovat analogový signál, který má určité fyzikální a statistické vlastnosti. Uživatel nastaví v softwaru zařízení spektrum zářiče a vlastnosti jednotlivých impulzů. Zařízení poté postupně vygeneruje pulzy zadaných vlastností, amplituda impulzů odpovídá uživatelem zadanému spektru. Tuto funkcionalitu popisuje Obrázek 1.13. Impulzy nejsou předpočítány, ale generují se za běhu emulátoru. Z tohoto důvodu může emulátor reprodukovat fyzikální jevy jako Poissonova distribuce času, provádět korelaci signálu mezi dvěma výstupními kanály, či simulovat několik zdrojů šumu.

Společnost CAEN nabízí více emulátorů podle jejich vybavení a vlastností. Zaměřím se konkrétně na zařízení s označením DT5810. Všechny emulační procesy zajišťuje FPGA Kintex 7. DT5810 obsahuje dva extrémně rychlé 16bitové D/A převodníky, které mají vzorkovací frekvenci 1 Gsps. D/A převodník s takovou rychlostí dovoluje generovat impulzy, které mají náběžnou dobu 1 ns. Každý z D/A převodníků buď může být spuštěn v módu optimalizovaný na rychlost – 1 ns náběžná doba, 4 Vpp (peak-to-peak napětí), či v módu optimalizovaný na amplitudu – 30 ns náběžná doba impulzu, 18 Vpp. Zařízení

1. ANALÝZA



Obrázek 1.14: Blokové schéma emulátoru DT5810 [26].

také obsahuje 14bitový A/D převodník 125 Msps¹ s nastavitelným rozsahem napětím (10 V, 5 V, 2.5 V, 1.25 V). A/D převodník umožňuje kombinovat emulaci s externím analogovým signálem, nebo analýzu signálu ze skutečného detektoru záření za účelem inicializace emulace. Blokové schéma zařízení je na Obrázku 1.14.

Další funkcionalitou emulátorů je generování impulzů uživatelem definovaného tvaru. Zařízení je dodáváno se softwarem, který slouží k ovládnání zařízení. Software obsahuje i databázi izotopů, pomocí níž je možné generovat spektrum jednotlivých radionuklidů, či kombinovat více radionuklidů. Více informací o emulátoru DT5810 je na stránkách výrobce [26].

Emulátor DT5810 je profesionální nástroj pro simulaci detektorů záření. K testování výsledného zařízení ho však nelze použít, protože není k dispozici. Část základních funkcionalit tohoto emulátoru je realizován na dostupném hardwaru (vývojová deska Nucleo-144 L4R5ZI) a je použita k testování zařízení.

¹V textu [26] se uvádí 125 Msps. Blokové schéma na Obrázku 1.14 a text [27] uvádí 150 Msps.

Návrh

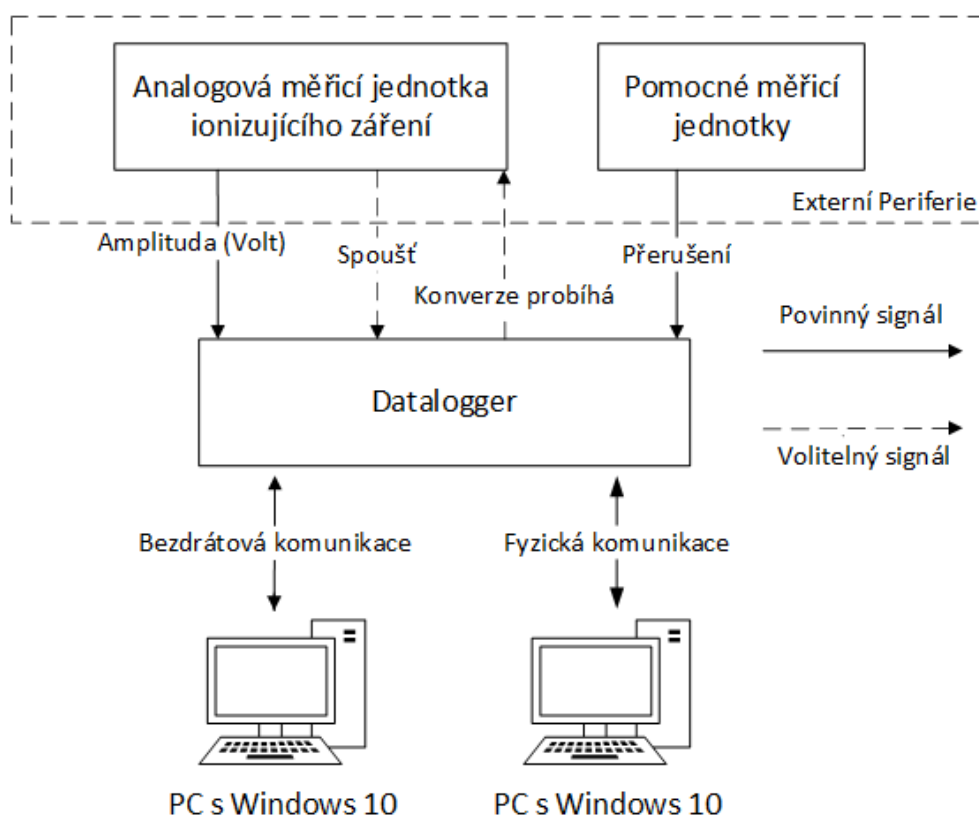
Systém se skládá ze tří prvků. Prvním prvkem je analogová měřicí jednotka spektra ionizujícího záření (dále jako měřicí část), která zajišťuje detekci ionizujícího záření. Měřicí část má také za úkol zpracování a zesílení daného pulzu, aby byl měřitelný A/D převodníkem dataloggeru. Aby byl pulz měřitelný A/D převodníkem musí trvat dostatečně dlouhou dobu (desítky až stovky mikrosekund) a měl by být dostatečně zesílen. Výstupem tohoto prvku je elektrické napětí udávající amplitudu (odpovídá energii zachycené částice/fotonu) a volitelná spoušť (trigger), která datalogger upozorňuje, že signál je možné číst A/D převodníkem. Vstupem je signál, který signalizuje, zdali A/D převodník provádí konverzi. Tento signál je možné využít např. k vybití kondenzátorů, udržující maximální hodnotu. Měřicí část je realizována jako černá skříňka modulem třetí strany. V rámci této diplomové práce je simulována pomocí D/A převodníku. Druhým prvkem systému je datalogger, který slouží jako analyzátor impulzů z měřicí částí. Analyzátor impulzů třídí impulzy podle amplitudy a vytváří spektrum. Toto spektrum je následně ukládáno na paměťové médium. K dataloggeru je možné připojit i další pomocné měřicí periferie jako třeba Geiger-Müllerův detektor. Datalogger se skládá z mikrokontroléru STM32 a Wi-Fi modulu. Posledním prvkem je obslužná stanice, neboli osobní počítač s operačním systémem Windows 10 na kterém je spuštěna aplikace. Tato aplikace slouží ke konfiguraci dataloggeru a vizualizaci naměřených dat. Počítač může s dataloggerem komunikovat buď fyzicky pomocí USB nebo bezdrátově pomocí Wi-Fi modulu. Celý systém je zobrazen na Obrázku 2.1.

2.1 Návrh Dataloggeru

Požadavky na funkcionalitu dataloggeru jsou následující:

- Záznam dat v řádu hodin až dnů.
- Perzistentní uložení konfigurace.

2. NÁVRH



Obrázek 2.1: Základní prvky systému, plné šipky značí komunikaci mezi jednotlivými prvky systému. Čárkované šipky značí komunikaci, která je volitelná (spoušť, ani konverze probíhá nemusí být použita)

- Měření ukládat na externí úložiště.
- Fyzická a bezdrátová komunikace.
- Požadavky na spolehlivý provoz.

Datalogger zaznamenává data ze senzorů po delší dobu bez vnějšího zásahu uživatele. V pravidelných intervalech zapisuje aktuální data ze senzorů na velkokapacitní externí úložiště. Důležitá je i komunikace s dataloggerem, která probíhá za účelem stažení dat, či konfigurace zařízení. Vzhledem k tomu, že zařízení může být umístěno i na špatně přístupném místě (např. jeskyně, střecha domu atd..) je potřeba i bezdrátová komunikace. Aby byl zajištěn spolehlivý provoz, tak některé moduly jsou zdvojené (duplex).

2.1.1 Architektura dataloggeru

Ke splnění všech požadavků na funkcionalitu je nutné využít různých částí, které jsou navzájem propojeny na desce plošných spojů. Toto propojení popisuje blokové schéma na Obrázku 2.2. Jedná se o následující části:

- Řídicí jednotka
- Bezdrátová komunikační jednotka
- Fyzická komunikační jednotka
- Hodiny reálného času
- Konzistentní úložiště
- Senzorová část

Řídicí jednotka zajišťuje ukládání dat, zpracovává data z měřicí části a dalších senzorů. Řídicí jednotka taktéž řídí veškerou komunikaci s periferiemi v dataloggeru. Je realizována pomocí mikrokontroléru STM32.

Bezdrátová komunikační jednotka zajišťuje bezdrátovou komunikaci s PC pomocí Wi-Fi. Bezdrátová komunikační jednotka se využívá takovým způsobem, aby byla komunikace této jednotky s řídicí jednotkou redukována na minimum. Z tohoto důvodu se na tuto jednotku ukládají aktuální naměřené hodnoty, či se využívá jako mezipaměť pro přenášení většího objemu dat mezi řídicí jednotkou a počítačem.

Fyzická komunikační jednotka poskytuje fyzickou komunikaci s počítačem pomocí USB. Je realizována pomocí UART-USB převodníku CP2102. Hodiny reálného času zajišťují řídicí jednotce přesný čas. Vzhledem k povaze dat je nutné data ukládat i s přesným časem měření.

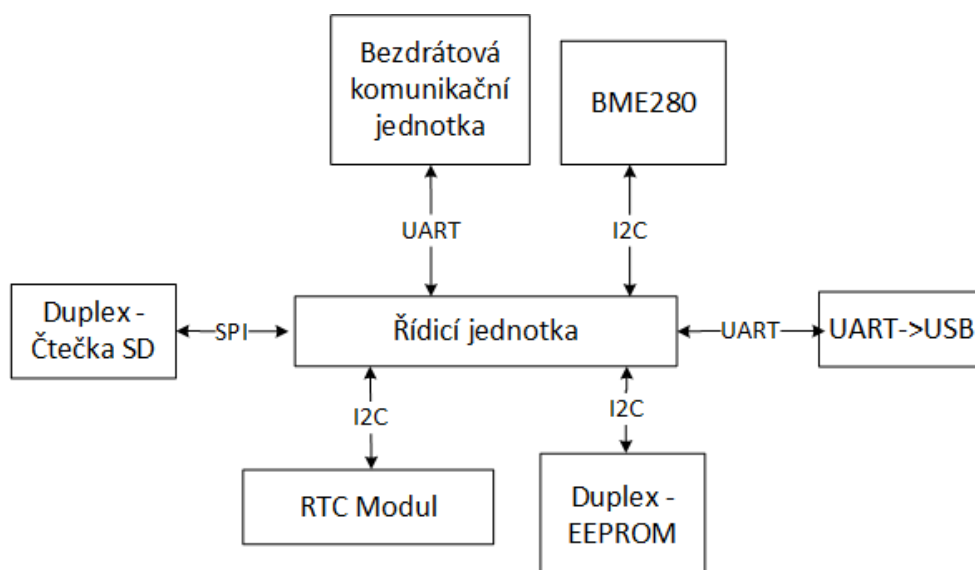
Konzistentní úložiště se používá jednak k uložení naměřených dat, ale i k uložení konfigurace dataloggeru, či sítě. Tyto hodnoty je nutné udržet i po ztrátě napájení. Senzorová část zahrnuje snímání dat, které mohou upřesnit měření. Jedná se o snímání aktuální teploty, vlhkosti a atmosférického tlaku. Pomocí těchto veličin je možné pozorovat vliv počasí, či nadmořské výšky na ionizující záření.

2.1.2 Řídicí jednotka

Řídicí jednotka je hlavní část celého dataloggeru. Jsou k ní připojené všechny periferie dataloggeru a řídí komunikaci mezi jednotlivými částmi dataloggeru. Požadavky na řídicí jednotku jsou následující:

- A/D převodník s minimálním rozlišením 10 bitů.
- Dvě komunikační rozhraní UART.

2. NÁVRH



Obrázek 2.2: Blokové schéma desky plošných spojů dataloggeru, včetně způsobů komunikace mezi řídicí jednotkou a ostatními prvky dataloggeru.

- Dvě komunikační rozhraní I²C.
- Dvě komunikační rozhraní SPI.

A/D převodník je potřeba pro vyčtení hodnot z analogové měřicí části. V amatérské spektrometrii se také využívá vzorkování pomocí zvukové karty. Z tohoto důvodu by vzorkovací frekvence A/D převodníku měla být alespoň 44100 vzorků za sekundu. Tato hodnota odpovídá běžné vzorkovací frekvenci zvuku [20]. Platí ale pravidlo, že čím vyšší vzorkovací frekvence, tím lépe. Dvě komunikační rozhraní UART jsou potřeba, protože je nutné k řídicí jednotce připojit bezdrátovou komunikační jednotku a fyzickou komunikační jednotku. Každá tato jednotka potřebuje svoje vlastní komunikační rozhraní UART. Komunikační rozhraní I²C se používá pro připojení některých modulů (EEPROM, RTC modul, BME280). Paměti EEPROM jsou implementované v duplexu, z tohoto důvodu je nutné mít dvě různé komunikační rozhraní I²C. Komunikační rozhraní SPI se využívá pro připojení SD čtečky. Je sice možné na jedno SPI rozhraní připojit více SD čteček (daný modul je adresován pomocí signálu slave select), avšak k dosažení úplného duplexního zapojení je ale žádoucí zapojit každou čtečku na jiné SPI rozhraní.

Mikrokontroléry STM32 nabízí alespoň jeden A/D převodník. Většina STM32 má 12bitový A/D převodník typu *Successive-approximation*. Existují varianty STM32, které mají dokonce 16bitový A/D převodník typu *Delta-*

sigma modulation, který je přesnější. Více informací o A/D převodnících STM32 zde [28]. Delta-sigma modulace je blíže popsána zde [29].

S větším rozlišením A/D převodníku rostou i paměťové nároky (jak na volatilní paměť, tak i na non-volatilní paměť), ale zvyšuje se i objem dat, které je nutné přenášet mezi řídicí jednotkou a Wi-Fi modulem (popř. mezi Wi-Fi modulem a počítačem). Z tohoto důvodu je použit mikrokontrolér, který nabízí pouze 12bitový A/D převodník. Pro účely amatérského dataloggeru je to dostatečné rozlišení.

Pro potřeby vývoje je použita vývojová deska WeAct Black Pill. Tato vývojová deska obsahuje kromě samotného mikrokontroléru STM32 také např. externí oscilátory, či regulátor napětí. Tuto vývojovou desku lze snadno zapojit do nepájivého pole, což je využitelné během fáze vývoje dataloggeru, kdy ještě není navrhnutá a vytvořena deska plošných spojů. Tato vývojová deska se využívá i v pozdějších fázích vývoje, kdy je nasazena do desky plošných spojů. Přehled dalších vývojových desek STM32 je k nalezení z volně dostupných zdrojů [30].

Během návrhu bylo kromě použití vývojové desky WeAct Black Pill zvažováno i použití méně výkonné desky Blue Pill. V další podkapitole jsou tyto desky blíže představeny. Desky ale nemají kompatibilní piny, takže není možné desku změnit ve fázi, kdy je navrhnutá a vyrobená deska plošných spojů. Rozdíly v pinech ukazuje Obrázek 2.3. Ke konfiguraci mikrokontroléru je využit program STM32CubeMX, k psaní kódu STM32CubeIDE využívající GNU C/C++ for Arm® toolchain. Kvůli snadné přenositelnosti kódu mezi různými mikrokontroléry STM32 je většina (s výjimkou kritických částí) kódu psána za použití Hardware Abstraction Layer (HAL) knihoven. K debugování a programování je využit programátor ST-Link.

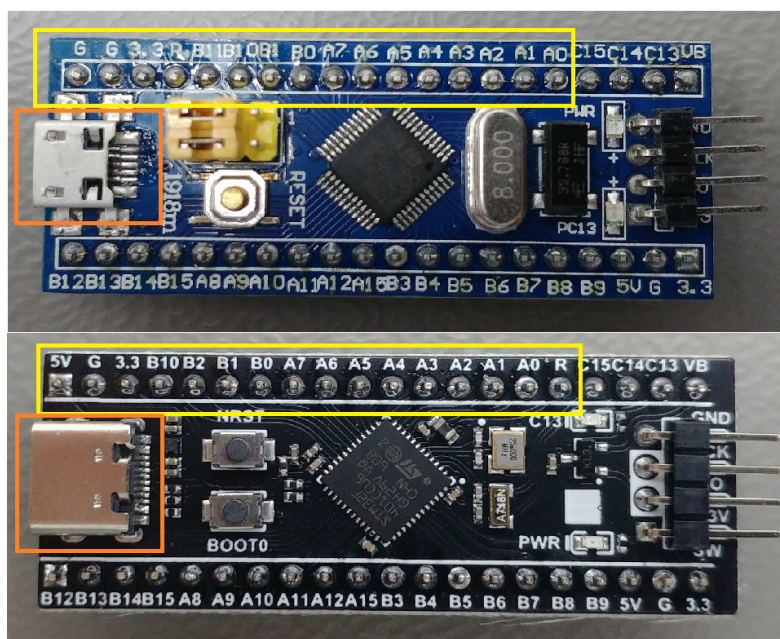
2.1.2.1 Vývojová deska Blue Pill

Informace o vývojové desce Blue Pill byly čerpány z volně dostupných zdrojů [31], informace o mikrokontroléru STM32F103C8T6, jímž je tato deska osazena byly čerpány z webové stránky [33] a referenčního manuálu mikrokontroléru [34]. Základní parametry tohoto mikrokontroléru popisuje Tabulka 2.1.

Vzhledem k nedostatečné velikosti RAM je nutné u STM32F103C8T6 používat A/D převodník s rozlišením pouze 11 bitů. A/D převodník tohoto mikrokontroléru nenabízí nastavitelné rozlišení. Redukce rozlišení se provádí softwarově po konverzi.

Během nákupu vývojové desky jsem narazil na problém, že místo originálního STM32F103C8T6 obsahovala padělek CS32F103C8T6. Dokonce pouzdro mikrokontroléru vypadalo stejně jako v případě originálního STM32. Padělek se překvapivě hlásí se stejným Device ID. Jediný znatelný rozdíl byla velikost paměti FLASH. Originální STM32F103C8T6 po připojení k ST-Link Utility ukazuje, že disponuje 64KB paměti, CS32F103C8T6, že disponuje 128KB pa-

2. NÁVRH



Obrázek 2.3: Porovnání vývojové desky Blue Pill (nahore) a WeAct Black Pill (dole). Obě dvě desky mají zapojení typu DIP-40. Spodní řada pinů je totožná, horní se ale výrazně liší. Další rozdíl je, že Blue Pill obsahuje MicroUSB konektor, WeAct Black Pill USB-C konektor [31], [32].

Tabulka 2.1: Základní parametry STM32F103C8T6

Typ rozhraní	Počet
UART:	3
SPI:	2
I ² C:	2
A/D převodník:	2 nezávislé 12bitové A/D převodníky s vzorkovací frekvencí 1 MSample/s
Ostatní	
CPU jádro:	ARM Cortex-M3, maximální frekvence: 72MHz
Paměť SRAM:	20KB
Paměť ROM:	64KB

CS32F103C8T6 (padělek) STM32F103C8T6

Device	STM32F10xx Medium-density	Device	STM32F10xx Medium-density
Device ID	0x410	Device ID	0x410
Revision ID	Rev X	Revision ID	Rev X
Flash size	128KBytes	Flash size	64KBytes

Obrázek 2.4: Výstup STM32 ST-Link utility pro CS32F103C8T6 a STM32F103C8T6. Vlevo informace o padělků CS32F103C8T6, vpravo originální STM32F103C8T6. Device, Device ID a Revision ID je stejné, liší se pouze velikost flash paměti.

Tabulka 2.2: Základní parametry STM32F103C8T6

Typ rozhraní	Počet
UART:	3
SPI:	4
I ² C:	3
A/D převodník:	1x 12bitový (nastavitelný rozlišení na 10, nebo 8, nebo 6 bitů) A/D převodník s vzorkovací frekvencí 2,4 MSample/s
Ostatní	
CPU jádro:	ARM Cortex-M4, maximální frekvence: 84MHz
Paměť SRAM:	64KB
Paměť ROM:	256KB

měti [35]. Výstup z ST-Link Utility pro oba mikrokontroléry ukazuje Obrázek 2.4.

2.1.2.2 Vývojová deska WeAct Black Pill

Informace o této vývojové desce byly čerpány zde: [32]. Tato vývojová deska je osazena mikrokontrolérem STM32F401CCU6, informace o tomto mikrokontroléru byly čerpány z referenčního manuálu mikrokontroléru [36] a oficiálních stránek výrobce [37]. Základní parametry tohoto mikrokontroléru popisuje Tabulka 2.2.

Výhodou mikrokontroléru STM32F401CCU6 je, že všechny I/O piny jsou 5V tolerantní (platí pouze pro konfiguraci pinu jako digitální vstup, neplatí v případě analogového vstupu). Toto se hodí např. pokud se k dataloggeru připojuje Geiger-Müllerův detektor, který funguje na 5V logice. Kdyby nebyly k dispozici 5V tolerantní piny, bylo by nutné přidávat HW navíc (logic shifter).

Kromě toho nabízí dostatečnou ROM i RAM a rychlý A/D převodník. Z těchto důvodů je použita tato vývojová deska ve finálním řešení dataloggeru.

2.1.3 Wi-Fi modul

Vývojová deska WeAct Black Pill neobsahuje Wi-Fi modul. Aby mohl datalogger komunikovat přes Wi-Fi je nutné použít externí modul. Jako Wi-Fi modul je využit ESP-01, který využívá čip ESP8266. Jedná se levný plnohodnotný mikrokontrolér, který kromě plného TCP/IP stacku obsahuje 32bitový RISC procesor L106. ESP-01 je možné buď naprogramovat jako jiné mikrokontroléry, nebo využít základní firmware, který umožňuje se připojit k síti pomocí Hayes-style příkazů (AT příkazy). ESP-01 se k jinému mikrokontroléru připojuje pomocí sériové linky. Více informací o ESP8266 a ESP-01 je k nalezení z volně dostupných zdrojů [38].

Původně bylo zamýšleno ESP-01 využívat pouze jako komunikační most mezi řídicí jednotkou a počítačem. Tím by se usnadnila implementace dataloggeru, protože by bylo možné použít demonstrační firmware od výrobce ESP-01, či by bylo nutné implementovat pouze jednoduchý firmware. Tím by ale nebyly využity výpočetní prostředky ESP-01 v efektivní formě. Docházelo by k zatěžování řídicí jednotky tím, že by tato jednotka musela řešit každou žádost od počítače a nespolehlivost UDP protokolu. Z tohoto důvodu byl navržen speciální firmware pro ESP-01, který umožňuje ukládání dat ze senzorů do mezipaměti ESP-01. Kromě toho firmware podporuje protokol, který usnadňuje přesun většího objemu dat a řeší nespolehlivost UDP protokolu. Tento firmware snižuje zátěž na řídicí jednotku. Je to z důvodu, že část požadavků z počítače vyřeší samostatně Wi-Fi modul bez nutnosti zásahu řídicí jednotky. Tento firmwaru ale kromě toho i zrychluje komunikaci mezi počítačem a dataloggerem. Bez tohoto firmwaru by bylo nutné data vždy číst z řídicí jednotky, byla by tedy nutná komunikace: PC - Wi-Fi modul - Řídicí jednotka - Wi-Fi modul - PC, s tímto firmware se část dat nachází přímo ve Wi-Fi modulu, stačí pouze následující komunikace: PC - Wi-Fi modul - PC.

Pro účely tohoto systému byly navrženy komunikační protokoly. Tyto protokoly podrobně popisuje podkapitola 2.1.7 Návrh specifických komunikačních protokolů. Pro programování ESP-01 byla využita knihovna Arduino core for ESP8266, která je včetně zdrojových kódů dostupná z repozitáře [39]. Firmware Wi-Fi modulu podporuje připojení k existujícímu přístupovému bodu, tak i vytvoření přístupového bodu.

Nevýhodou ESP-01 je jeho velká spotřeba, která může chvilkově dosahovat až 450mA [40]. To je řešeno použitím vhodného regulátoru. Napájení dataloggeru je podrobně řešeno v podkapitole 2.1.8 Napájení dataloggeru.

2.1.4 Hodiny reálného času

Přesný čas a datum je pro datalogger nutný. Každý STM32 má integrovanou jednotku reálného času [28]. Aby jednotka reálného času udržovala čas i po ztrátě napájení, je potřeba aby k pinu VBAT byl připojen alternativní zdroj energie (např. knoflíková baterie CR2032), dále je nutné aby zdroj hodin pro hodiny reálného času byl LSE oscilátor. Informace o správné konfiguraci RTC (zde konkrétně pro STM32F401) jsou k nalezení v referenčním manuálu mikrokontroléru [36].

Kromě této integrované jednotky je využit i další zdroj hodin reálného času. Jedná se o modul reálného času založený na čipu DS3231. Tento modul se připojuje přes rozhraní I²C. A kromě samotného RTC, který využívá velmi přesný teplotně kompenzovaný oscilátor, obsahuje i 4kilobajtovou EEPROM AT24C32. Tento modul je při ztrátě napájení napájen knoflíkovou baterií (nabíjecí LIR2032, popř. CR2032, či CR2025, je ale nutné odstranit nabíjecí obvod na modulu). Informace o tomto modulu jsou k nalezení zde: [41].

2.1.5 Výběr hardwaru pro non-volatilní paměť

Pro správný chod dataloggeru je potřeba aby obsahoval paměť, která udrží informace i po ztrátě napájení. Do této paměti se ukládají data z měření. Každé měření může mít velikost až stovky megabajtů (v závislosti na tom, jak dlouho měření probíhá). Kromě samotných dat z měření je nutné mít uloženou i konfiguraci dataloggeru. Tato konfigurace má velikost v řádech desítek bajtů. Dále je nutné mít uloženou konfiguraci bezdrátových sítí (SSID, heslo a další údaje). Každá konfigurace sítě má velikost přibližně 100 bajtů.

Během návrhu byly zvažovány tři způsoby, jak docílit ukládání dat a konfigurace. Prvním způsobem je ukládání dat i konfigurace na stejné velkokapacitní úložiště - paměťová karta. Výhodou tohoto řešení je, že nevyžaduje dodatečný HW a SW navíc. Nevýhodou je, že pro správnou funkci je nutné, aby vždy byla přítomná SD karta. Záměrem bylo, aby datalogger v omezeném režimu (sledování aktuálních dat) fungoval i bez přítomnosti SD karty, což by tento způsob non-volatilní paměti komplikoval. Další nevýhodou je složitější výměna SD karet - při výměně SD karty je nutné konfiguraci manuálně zkopírovat z jedné karty na druhou, jinak by byla nutná rekonfigurace zařízení.

Druhý způsob je pro uložení dat použít velkokapacitní úložiště. A pro uložení konfigurace interní non-volatilní paměť. Mikrokontroléry STM32 nemají interní EEPROM jako má např. Arduino [42]. V STM32 se využívá emulace EEPROM v interní flash paměti. Některé STM32 kromě toho nabízí i baterií napájenou záložní SRAM, kterou lze využít také jako non-volatilní paměť. Záložní SRAM není pro účely tohoto dataloggeru využito, protože by to značně snižovalo přenositelnost kódu mezi různými STM32. Výhodou emulované EEPROM je, že nevyžaduje další HW navíc, odbourává to i potřebu SD karty.

Nevýhodou je, že emulovaná EEPROM vyžaduje před zápisem provést mazací operaci, toto je nutné dělat softwarově. Detailní popis emulované EEPROM ve flash (konkrétně pro STM32F40x/STM32F41x) je popsáno v aplikační poznámce [43].

Třetí způsob je pro uložení dat použít velkokapacitní úložiště a pro uložení konfigurace externí EEPROM, které se připojují např. přes I²C. Výhodou je, že je možné současně připojit více externích EEPROM (a tím docílit redundance). Nevýhodou oproti ostatním řešením je, že je potřeba HW navíc.

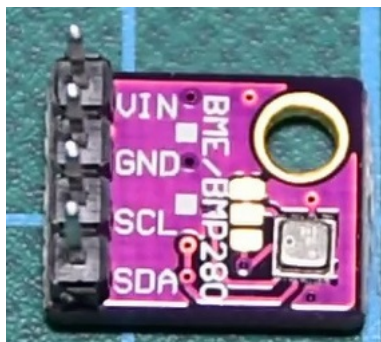
Konfigurace dataloggeru je pro jeho správný chod důležitá, byl tedy zvolen třetí způsob (velkokapacitní úložiště pro data + externí EEPROM), aby bylo možné dosáhnout redundance. Využívají se dvě externí EEPROM STM M24C16-WBN6P, každá z nich je schopna uchovat až 2 kilobajty dat. Tyto moduly se připojují přes I²C, mají fixní adresu, kterou nelze změnit. Není tedy možné mít na jedné I²C sběrnici dva stejné moduly. Z tohoto důvodu byl jeden z požadavků na řídicí jednotku alespoň dvě I²C rozhraní. Veškeré informace o tomto modulu je k nalezení v datasheetu [44]. Konfigurace je uložena v režimu zrcadlení dat (na obou EEPROM jsou stejná data), využívá se duplex.

Jako velkokapacitní úložiště je použita microSD paměťová karta. Data z měření jsou taktéž důležitá, proto i u velkokapacitního úložiště je využito zrcadlení naměřených dat na více nezávislých paměťových médiích. Datalogger obsahuje dvě SPI microSD čtečky. Každá čtečka je připojena k jiné SPI sběrnici, protože kromě selhání čtečky/karty může dojít i k poškození spoje SPI vodičů (studený spoj, poškození PCB atd.), pokud by obě čtečky byly připojeny k jedné SPI sběrnici došlo by současně k selhání obou čteček.

2.1.6 Výběr senzoru teploty, vlhkosti a tlaku

Jak již bylo zmíněno v úvodu této kapitoly, pomocí těchto veličin je možné zkoumat např. nadmořskou výšku. Znalost nadmořské výšky se může hodit pokud je datalogger využit k měření spektra kosmického záření pomocí stratosférického balonu. Takové lety pravidelně pořádá oddělení dozimetrie záření, více informací o těchto letech je k nalezení na webových stránkách ústavu [45]. Znalost těchto veličin je také vhodná pokud je datalogger použit ke sledování vlivu počasí (déšť, bouřka, změny atmosférického tlaku) na spektrum ionizujícího záření.

Měření všech těchto veličin nabízí senzor BME280. Senzor se k mikrokontroléru připojuje pomocí sběrnice I²C. BME280 nabízí mnoho módů měření (např. monitorování počasí, měření nadmořské výšky atd.), které nastavují přesnost jednotlivých čidel, minimální periodu mezi čtením a spotřebu modulu. K tomuto senzoru je dodáno API [46], které usnadňuje vývoj aplikací založené na tomto senzoru. Je nutné pouze doplnit funkce pro čtení/zápis z I²C sběrnice a mikrosekundové zpoždění. Používání tohoto API je přímo doporučeno výrobcem senzoru, z tohoto důvodu je ve firmwaru dataloggeru využito. Více informací o tomto senzoru je k nalezení v datasheetu senzoru



Obrázek 2.5: BME280 modul [48].

[47]. V dataloggeru je využit modul, který tento senzor obsahuje. Kromě senzoru obsahuje i např. pull-up rezistory potřebné pro I²C. Modul BME280 je zobrazen na Obrázku 2.5. Vzhledem k tomu, že se nejedná o klíčové veličiny pro samotné měření spektra ionizujícího záření není využito redundance.

2.1.7 Návrh specifických komunikačních protokolů

Složitější komunikace probíhá mezi počítačem a dataloggerem, ale i v rámci dataloggeru mezi řídicí jednotkou a Wi-Fi modulem. Z tohoto důvodu jsou navrženy 3 protokoly:

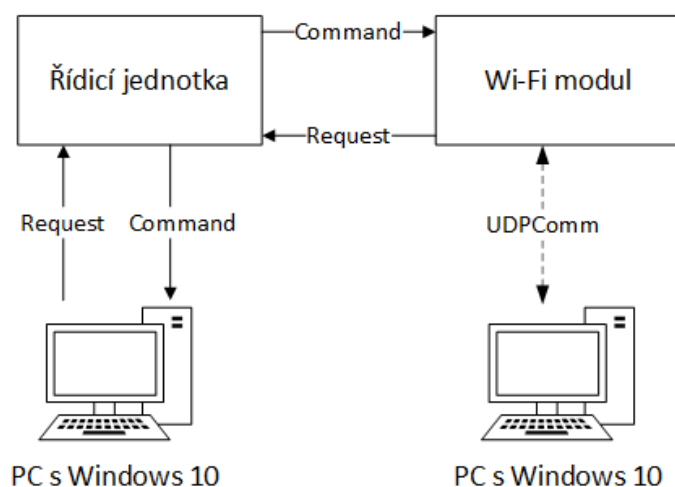
- **Request protokol**
- **Command protokol**
- **UDPComm protokol**

Request protokol se využívá ke komunikaci směrem k řídicí jednotce. Hlavní vlastností tohoto protokolu je, že nemá garanci vyřízení řídicí jednotkou. V případě, že řídicí jednotka je zatížená měřením, může tyto žádosti ignorovat, či je vyřídit až přijde vhodný čas. Řídicí jednotka nepotvrzuje komunikaci.

Command protokol využívá řídicí jednotka směrem k ostatním částem systému. Na rozdíl od Request protokolu musí daná část systému na tento příkaz ihned reagovat. Daná část systému musí řídicí jednotce potvrdit úspěšné přijetí (a vyřízení příkazu), či jí upozornit na chybu.

UDPComm protokol se využívá ke komunikaci mezi Wi-Fi modulem dataloggeru a počítačem. Jedná se o aplikační vrstvu postavená na protokolu UDP. Wi-Fi modul tento příkaz vyřizuje neprodleně, potvrzení je dané tím o jaký typ příkazu se jedná.

Všechny protokoly jsou navrženy tak, aby byly co nejvíce jednoduché a v největší možné míře využívaly tisknutelné znaky (kvůli snadnému ladění). Paket každého protokolu obsahuje kontrolní součet (CRC32) podle kterého



Obrázek 2.6: Komunikační schéma systému, plné šipky značí fyzickou komunikaci, čárkovaná šipka bezdrátovou komunikaci.

se ověřuje konzistence dat. Jako generující polynom je použit $0x04C11DB7$, který se používá např. v GZIPu či ho využívá Ethernet [49]. Komunikace dataloggeru s počítačem může probíhat buď pomocí Wi-Fi, v tomto případě počítač komunikuje přímo pouze s Wi-Fi modulem pomocí UDPComm, nebo přes USB port pomocí Request protokolu. Řídicí jednotka odpovídá Command protokolem. Komunikaci mezi jednotlivými částmi systému ukazuje Obrázek 2.6. Následující podkapitoly blíže popisují jednotlivé protokoly.

2.1.7.1 Request protokol

Request protokol umožňuje Wi-Fi modulu, či počítači připojenému přes USB komunikovat s řídicí jednotkou. Každý paket tohoto protokolu je označován jako žádost. Využívá se k získávání dat ze senzorů, stahování uložených souborů, ovládání měření, získávání stavu řídicí jednotky, či konfigurace zařízení. Není garantovaná doba, za jakou řídicí jednotka daný paket vyřídí, ale ani to jestli daný paket vůbec vyřídí. Je to z důvodu, že komunikace s ostatními částmi systému nesmí nijak narušit měření ionizujícího záření, zpracování žádostí je tedy vedlejší. Řídicí jednotka nepotvrzuje přijetí žádosti. Řídicí jednotka na žádost odpovídá buď příkazem (Command protokol), nebo nijak v závislosti na typu žádosti. Tento protokol využívá pouze tisknutelné znaky. Každý paket je ukončen znakem nového řádku ($0x0A$). Formát paketů popisuje Tabulka 2.3.

Některé typy žádostí si nevystačí pouze s dvěma 32bitovými operandy. Jedná se o žádosti, které mají za úkol nastavovat konfigurace sítě. V případě nastavování statické IP adresy je potřeba přenést dvanáct oktetů (IP adresa,

Tabulka 2.3: Formát paketu request protokolu

Formát paketu: 0 CODE ID [OPERAND1] [OPERAND2] CRC32\n	
Typ	Popis
0	Jednoznačně identifikuje, že se jedná o paket Request protokolu.
CODE	Kladné číslo které definuje o jaký typ požadavku se jedná.
ID	Kladné číslo které identifikuje paket. Toto ID se použije i jako ID příkazu, pokud řídicí jednotka na daný typ žádosti odpovídá příkazem.
OPERAND1	Kladné číslo. OPERAND1 je povinný pouze pro některé typy žádosti.
OPERAND2	Kladné číslo. OPERAND2 je povinný pouze pro některé typy žádosti.
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo obsahující vždy 8 znaků. Používá se zarovnání nulami.

maska podsítě, výchozí brána), v případě nastavování identifikačních údajů - SSID, heslo a dodatkové informace (např. typ šifrování). Formát paketu žádosti nastavování statické IP adresy popisuje Tabulka 2.4. Formát paketu konfigurace identifikačních údajů sítě má popisuje Tabulka 2.5:

Request protokol umožňuje plnou konfiguraci dataloggeru a stahování naměřených dat. Funkce tohoto protokolu jsou následující:

- Získávání všech aktuálních dat - ať už se jedná o spektrum ionizujícího záření, ale i data z ostatních senzorů.
- Konfigurace dataloggeru - nastavování módu jednotlivých senzorů, konfigurace sítě a ukládání konfigurací do non-volatilní paměti.
- Správa uložených měření - stahování těchto měření na počítač, mazání těchto měření.
- Správa měření - vzdálené zapínání/vypínání měření.

2.1.7.2 Command protokol

Command protokol využívá řídicí jednotka ke komunikaci s Wi-Fi modulem, či počítačem připojeným přes USB. Paket tohoto protokolu je označován jako příkaz. Řídicí jednotka tento protokol používá k aktualizaci naměřených dat, konfigurace sítě, či přenosu uložených souborů. Wi-Fi modul, či počítač musí příkaz neprodleně vyřídit a potvrdit úspěšné vyřízení (či chybu) zasláním

2. NÁVRH

Tabulka 2.4: Formát paketu request protokolu – nastavování statické IP adresy

Formát paketu: 0 CODE ID IP1 IP2 IP3 CRC32\n	
Typ	Popis
0	Jednoznačně identifikuje, že se jedná o paket Request protokolu.
CODE	Kladné číslo které definuje o jaký typ požadavku se jedná.
IP1	Požadovaná statická IP adresa. IP adresa se zapisuje v desítkové soustavě, bez teček. IP adresa 192.168.1.100 je zapsána jako 192 168 1 100.
IP2	Maska podsítě. Masku podsítě se zapisuje v desítkové soustavě, bez teček. IP adresa 255.255.255.0 je zapsána jako 255 255 255 0.
IP3	Výchozí brána sítě. Výchozí brána sítě se zapisuje v desítkové soustavě, bez teček. IP adresa 192.168.1.1 je zapsána jako 192 168 1 1.
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo obsahující vždy 8 znaků. Používá se zarovnání nulami.

Tabulka 2.5: Formát paketu request protokolu – konfigurace bezdrátové sítě

Formát paketu: 0 CODE ID OPERAND1 SSID_LEN SSID PASS_LEN PASS CRC32\n	
Typ	Popis
0	Jednoznačně identifikuje, že se jedná o paket Request protokolu.
CODE	Kladné číslo které definuje o jaký typ požadavku se jedná.
OPERAND1	Úpřesňuje např. typ šifrování v případě konfigurace klienta, maximální počet klientů, či kanál v případě konfigurace přístupového bodu.
SSID_LEN	Udává počet znaků SSID.
SSID	SSID sítě (délka se musí shodovat s SSID_LEN).
PASS_LEN	Udává počet znaků hesla.
PASS	Heslo sítě (délka se musí shodovat s PASS_LEN).
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo obsahující vždy 8 znaků. Používá se zarovnání nulami.

Tabulka 2.6: Formát paketu command protokolu

Formát paketu: <i>PACKET_LEN</i> CODE ID [<i>OPERANDS</i> <i>DATA</i>] <i>CRC32</i>	
Typ	Popis
PACKET_LEN	Udává délku paketu včetně CRC32. Délka je vždy 3znaková hodnota. Teoretický rozsah délky paketu je 001-999 bajtů (v případě menších paketů je nutné doplnit počáteční nuly). Délka je uvedena bez prvních třech bajtů, které slouží pouze ke správnému načtení paketu.
CODE	Kladné číslo zapsané dekadicky pomocí tisknutelných znaků. CODE udává o jaký typ příkazu se jedná.
ID	Kladné číslo zapsané dekadicky pomocí tisknutelných znaků. Toto číslo identifikuje paket.
OPERANDS DATA	Jedná se o kombinaci operandů a dat (v libovolném pořadí). Operandy jsou 32bitová čísla (jak znaménková, tak i bezznaménková), které jsou zapsané pomocí tisknutelných znaků. Data jsou binární (nemusí se jednat o tisknutelné znaky). Jeden příkaz může obsahovat až 8 operandů (z toho 4 znaménkové a 4 bezznaménkové) a 320 bajtů dat. Operandy a data upřesňují příkazy.
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo zapsané tisknutelnými znaky obsahující vždy 8 znaků. Používá se zarovnání nulami.

Command Response (odpověď na příkaz) řídicí jednotce. Pomocí tohoto protokolu je nutné přenášet i binární data, pakety tedy mohou obsahovat i netisknutelné znaky. Paket není zakončen znakem nového řádku jako v případě request protokolu, ale obsahuje informaci o délce paketu v prvních třech znacích. Formát jednotlivých typů příkazů je různorodější než v případě žádostí, ale základní formát je stejný pro všechny pakety a popisuje ho Tabulka 2.9

Příkazy lze rozdělit na tři typy podle jejich účelu:

1. **Síťové příkazy:** Příkazy zajišťující správu Wi-Fi a UDP serveru. Tyto příkazy umožňují např. připojení k přístupovému bodu, poslouchání na daném portu, scan blízkých Wi-Fi sítí atd.. Síťové příkazy se nevyužívají pokud řídicí jednotka komunikuje přímo s počítačem přes USB. Tyto příkazy mají CODE v rozsahu 1-14.
2. **Datové příkazy:** Příkazy pro přesun dat mezi řídicí jednotkou a Wi-Fi modulem (popř. počítačem připojený přes USB). Tato data se poté

2. NÁVRH

Tabulka 2.7: Formát paketu odpověď na příkaz (command response)

Formát paketu: CODE ID STATUS [OPT_STATUS] CRC32\n	
Typ	Popis
CODE	Kladné číslo udávající CODE příkazu. CODE musí být totožný s CODE příkazu.
ID	Kladné číslo udávající ID příkazu. ID musí být totožné s ID původního příkazu.
STATUS	Znaménkové číslo, udávající stav zpracování paketu příkazu. Úspěšné zpracování paketu negarantuje úspěšné vyřízení příkazu. Všechny možné stavové kódy popisuje Tabulka 2.8.
OPT_STATUS	Pomocné příznaky určující stav zpracování příkazu.
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo obsahující vždy 8 znaků. Používá se zarovnání nulami.

Tabulka 2.8: Stav vyřízení paketu command protokolu

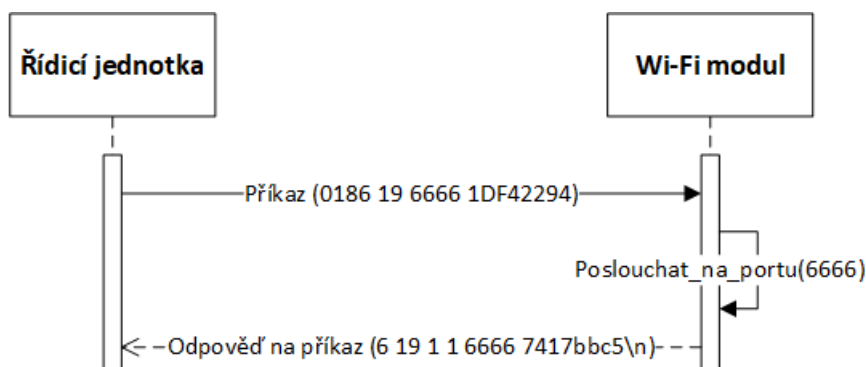
Kód:	Význam
1	Zpracování paketu proběhlo v pořádku, pro ověření správnosti vykonání příkazu je třeba vyhodnotit pomocné příznaky.
-1	Neznámá hodnota CODE.
-2	Paket obsahuje příliš mnoho binárních dat.
-3	Chybné CRC32 (CRC32 paketu se neshoduje s udávaným CRC32).
-4	Špatný formát paketu pro daný typ příkazu.

ukládají do mezipaměti daného zařízení. Rozsah CODE je 101-107.

- Speciální příkazy:** Příkazy sloužící pro přímou komunikaci s počítačem. Pokud se komunikuje s počítačem bezdrátově, tak se dané příkazy ve Wi-Fi modulu překládají na UDP pakety. Rozsah CODE je 201-205.

Wi-Fi modul, či počítač na každý příkaz odpovídá odpovědí na příkaz (command response). Podobně jako žádost, tak i odpověď na příkaz je složen pouze z tisknutelných znaků a je ukončen pomocí znaku newline (0x0A). Formát odpovědi na příkaz popisuje Tabulka 2.9.

Obrázek 2.7 ilustruje komunikaci mezi řídicí jednotkou a Wi-Fi modulem pomocí Command protokolu.



Obrázek 2.7: Obrázek ukazuje, jak řídicí jednotka nastaví Wi-Fi modul, aby poslouchal UDP komunikaci na portu 6666. Řídicí jednotka nejprve pošle příkaz, který je dlouhý 18 bajtů (018), s CODE 6, ID 19 a operandem 6666 (operand pro tento typ příkazu definuje port), CRC32 paketu je 1DF42294. Wi-Fi modul daný paket zpracuje, začne poslouchat na portu 6666 a řídicí jednotce odpoví odpovědí na příkaz, se stejným CODE a ID jako měl příkaz na který odpovídá, kromě toho potvrdí i úspěšné zpracování paketu (1), úspěšné otevření portu (1) a zřekapituluje na jakém portu poslouchá (6666), paket zabezpečí CRC32 a zakončí znakem newline. Jedná se o ideální případ komunikace, kde nic neselhalo.

2.1.7.3 UDPComm protokol

UDPComm protokol je využit pro vzdálenou komunikaci mezi počítačem a dataloggerem. Jak již název napovídá UDPComm protokol využívá UDP protokol. UDP bylo vybráno, protože v porovnání s TCP vyžaduje minimální režii. Jakýkoliv počítač v síti může kdykoliv přečíst aktuální hodnoty senzorů. Nespolehlivost doručení datagramu je řešena na aplikační vrstvě. Na straně počítače opakovaným odesláním datagramu, na straně dataloggeru např. protokolem se selektivním opakováním, či indexem udávající pořadí datagramu. Některé operace vyžadují stavovost, např. je žádoucí aby konfiguraci dataloggeru v jednu chvíli prováděl pouze jeden uživatel. UDPComm protokol proto zavádí pojem Master. Master je daný IP adresou a portem, tyto hodnoty si Wi-Fi modul pamatuje. V jednu chvíli může být pouze jeden Master, který má přístup k neomezenému ovládní Dataloggeru. Ostatní klienti mají přístup pouze k omezeným funkcím dataloggeru. UDP datagram obsahuje CRC16, v případě IPv4 se ale jedná o volitelnou položku. z tohoto důvodu paket UDPComm protokolu obsahuje CRC32. Podrobné informace o UDP protokolu jsou k nalezení na webových stránkách [50].

UDPcomm protokol se využívá ke vzdálenému ovládní Dataloggeru, získávání aktuálně naměřených dat, či stahování souborů. Paket UDPComm, který se využívá směrem od počítače k dataloggeru se skládá pouze z tisknutelných

Tabulka 2.9: Formát paketu UDPComm protokolu

Formát paketu: ID CODE OPERAND [OPT_OPERANDS] CRC32	
Typ	Popis
ID	Kladné číslo udávající ID UDPComm paketu.
CODE	Kladné číslo udávající typ UDPComm paketu. CODE instrukce nevyžadující master přístup jsou 0 (žádost o data v mezipaměti Wi-Fi modulu) a 1 (žádost o aktualizaci dat v mezipaměti Wi-Fi modulu). Všechny ostatní CODE vyžadují master přístup. Pokud počítač bez master přístupu pošle Wi-Fi modulu UDPComm paket s CODE vyšším než 1, Wi-Fi modul mu na to odpoví paketem oznamující přístup byl odepřen.
OPERAND	Kladné číslo upřesňující UDPComm paket.
OPT_OPERANDS	Volitelné operandy povinné pouze pro určitý typ UDPComm paketu. Může se jednat o číslo, ale i textový řetězec.
CRC32	Kontrolní součet paketu CRC32. Hexadecimální číslo obsahující vždy 8 znaků. Používá se zarovnání nulami.

znaků. Formát paketu UDPComm směrem od počítače k dataloggeru popisuje Tabulka 2.9.

Formát paketu UDPComm směrem od dataloggeru k počítači již nemá jednotný formát a liší se typ od typu, pro komunikaci využívá i netisknutelné znaky.

2.1.8 Napájení dataloggeru

Vývojová deska WeAct Black Pill obsahuje USB-C, kterým je možné mikrokontrolér napájet. Na desce se také nalézá regulátor napětí AP7343, který 5 V z USB převádí na 3,3 V. Tento regulátor ale povoluje maximální odběr proudu 300 mA [32], což je pro datalogger nedostatečné (samotné ESP-01 má občasnou spotřebu až 450 mA viz 2.1.3 Wi-Fi modul). Z tohoto důvodu je nutné použít jiný styl napájení. Před samotným návrhem napájení je třeba zjistit jaký je maximální odběr proudu celého dataloggeru. Bohužel ne vždy je možné zjistit maximální spotřebu daného modulu, Tabulka 2.10 uvádí odhad maximální spotřeby každé části dataloggeru. Samotný odhad je ale nadhodnocen, je to z důvodu, že je lepší mít napájení, které plně nevyužije svůj potenciál, než mít napájení, které selže v případě neobvyklé zátěže.

Maximální odhadovaná spotřeba (jedná se o špičky, většinu času bude spo-

Tabulka 2.10: Spotřeba jednotlivých modulů dataloggeru

Součástka:	Odhadovaná maximální spotřeba (mA)
WeAct Black Pill	120 (Odhad dle kalkulačky spotřeby v STM32CubeMx - cca. 30mA, zbylých 90mA je rezerva (vývojová deska obsahuje LED, které odebírají proud, GPIO taktáž)
BME280	3 [47]
M24C16	10 [44]
Čtečka microSD karet	300 (Datasheet k modulu, který je použit v Dataloggeru není k dispozici, informace převzaty z jiného SPI modulu [51]. V jednu chvíli se vždy pracuje pouze s jedním modulem, druhý je ve stand-by módu s minimální spotřebou.)
RTC Modul	40 [52], [53]
ESP-01	450 [40]

třeba mnohem nižší) je kolem 950mA. Všechny komponenty fungují s napětím 3,3 V (některé to dokonce vyžadují). Jako napájecí konektor je využit USB-B. Maximální proud, který může tímto konektorem téct je 1,5 A [54]. Tento konektor bude umožňovat datalogger napájet např. pomocí powerbanky. Datové vodiče konektoru budou odpojeny a využívat se budou pouze napájecí vodiče (VBUS, GND). Vzhledem k tomu, že standardní napětí USB je 5 V, tak je nutné napětí převést na 3,3V. K tomuto účelu se bude využívat napěťový regulátor LD1117V33, každý takový regulátor je schopný dodávat 3,3 V s maximálním výstupním proudem 800mA [55]. Datalogger bude využívat dva napěťové regulátory LD1117V33. Jeden regulátor je určen pouze pro ESP-01, druhý zajišťuje napájení zbytku systému.

2.1.9 Zajištění spolehlivého provozu

Jeden z požadavků na funkcionalitu dataloggeru je spolehlivý provoz. Sběr dat může probíhat delší dobu, uživatel se musí spolehnout, že změřená data jsou správná. Pokud charakter chyb neumožňuje správnost dat, uživatel musí vědět, že naměřená data mohou být chybná. K dosažení vyšší spolehlivosti systému se využívá redundance, kterou lze rozdělit do dvou kategorií:

- Hardwarová redundance (redundance v prostoru)

2. NÁVRH

- Informační redundance

Hardwarová redundance využívá duplikaci modulů. V ideálním případě by bylo vhodné zduplikovat každou část systému, avšak duplikace např. řídicí jednotky způsobí neúměrný nárůst kódu, kde se můžou vyskytovat chyby. Kromě zvýšené náročnosti kódu, duplikace modulů způsobuje i to, že datalogger je složitější a obsahuje více modulů, které mohou selhat. Dalším problémem je růst spotřeby dataloggeru, což způsobuje, že se regulátory napětí více zahřívají a tím klesá jejich životnost. Proto byl u hardwarové redundance volen kompromis, aby došlo k maximálnímu zvýšení spolehlivosti za cenu minimálního zesložiténí (SW a HW) dataloggeru.

Pro měření je potřeba přesný čas. Datalogger obsahuje dvě nezávislé jednotky hodin reálného času. Každá jednotka je napájena vlastní knoflíkovou baterií CR2032. Další zduplikovaný modul je čtečka paměťových karet. Každá čtečka je zapojena na vlastní SPI rozhraní. Zápis dat probíhá na obě paměťové karty v režimu zrcadlení. Lze tedy v dodatečně ověřit, zdali jsou soubory stejné. V případě selhání jedné ze dvou čteček (či paměťových karet) jsou data k dispozici na druhé kartě. Posledním zduplikovaným modulem je EEPROM, kam se ukládá konfigurace dataloggeru a bezdrátových sítí.

Zduplikovaný není modul BME280, protože měření teploty, vlhkosti a tlaku nebylo vyhodnoceno jako důležité. Jak již bylo zmíněno výše, zduplikovaná není ani řídicí jednotka. V případě duplikace řídicí jednotky nelze využít přímočarého duplexu, ale bylo by nutné zajistit přepínání mezi aktivní a záložní řídicí jednotkou. Není možné, aby v jednu dobu byly aktivní dvě řídicí jednotky. Datalogger by musel obsahovat další obvody (či další mikrokontrolér), který by rozhodoval, která řídicí jednotka je ve vedení. Tím by sice byla zajištěna spolehlivost řídicí jednotky, avšak bylo by nutné řešit spolehlivost těchto řídicích obvodů, což by přineslo podobné problémy. Nárůst složitosti (jak SW, tak i HW) v tomto případě je neúměrný a zanesl by do dataloggeru více chyb, než by byl vůbec schopný řešit.

Zduplikovaný není přímo ani Wi-Fi modul. Datalogger kromě bezdrátové komunikace nabízí i fyzickou komunikaci. V případě, že selže Wi-Fi modul je možné s datalogger nadále komunikovat i fyzicky. Selhání Wi-Fi modulu nijak neovlivňuje měření.

Informační redundance využívá použití kódů pro detekci (a případnou opravu chyb). Informační redundance se využívá ve všech protokolech, které se využívají v systému. Každý paket obsahuje kontrolní součet CRC32, pomocí něho lze zjistit, zda paket na straně příjemce je stejný jako na straně odesílatele. V případě obdržení paketu s chybným CRC32 se posílání paketu opakuje. Hlavička souboru měření obsahuje informace o tom, jakým způsobem se data mají číst. Chybná hlavička může způsobit špatnou interpretaci dat. Z tohoto důvodu hlavička souboru měření obsahuje součet CRC32, aby bylo možné detekovat poškození hlavičky.

2.2 Návrh počítačové aplikace

Druhým prvkem systému je osobní počítač s operačním systémem Windows 10, na kterém je spuštěna obslužná aplikace, která komunikuje s dataloggerem buď fyzicky pomocí USB (virtuální sériový port), či bezdrátově pomocí Wi-Fi. Tato aplikace má následující požadavky na funkcionalitu:

- Konfigurace dataloggeru
- Správa měření
- Stahování dat z dataloggeru
- Zpracování a vizualizace stažených dat

Pomocí obslužné aplikace musí být možné intuitivně konfigurovat samotný datalogger, ale i spravovat bezdrátové sítě ke kterým se datalogger připojuje. Kromě toho je nutné, aby pomocí aplikace bylo možné sledovat i současný stav dataloggeru a chyby jednotlivých částí. Aplikace taktéž musí podporovat správu měření, to znamená, že aplikace musí mít možnost spustit či vypnout měření. Dalším úkolem aplikace je stahování dat z dataloggeru a to buď aktuálních naměřených dat, ale i dat, které jsou uchovány na paměťové kartě. Tyto data musí aplikace zpracovat a vizualizovat – pomocí tabulek a grafů. Uložené soubory na paměťové kartě mohou být nějakým způsobem poškozeny. Aplikace musí poskytovat i možnost soubor opravit, pokud není nenávratně poškozen.

První otázka, co během návrhu počítačové aplikace vyvstala, je, jestli je potřeba mít aplikaci s grafickým uživatelským rozhraním (GUI), či bude postačovat konzolová aplikace. Nevýhoda aplikace s grafickým uživatelským rozhraním je to, že je nutné toto grafické rozhraní vytvářet. Avšak existují návrháře, ve kterých se grafické rozhraní nakliká a tyto návrháře vytvoří příslušný kód. Vzhledem k velkému množství funkcionalit, co počítačová aplikace nabízí, je vhodnější použití grafického rozhraní. Aplikaci by bylo možné sice realizovat i pomocí konzolové aplikace. Její realizace by byla komplikovanější a výsledná aplikace by byla pro uživatele méně intuitivní a složitější na ovládání.

Když bylo známo, že počítačová aplikace bude obsahovat grafické uživatelské rozhraní, tak bylo nutné vybrat technologii/knihovny pro práci s GUI. Aplikace musí běžet na Windows 10, není třeba řešit přenositelnost kódu mezi různými operačními systémy. Z tohoto důvodu je možné využít některou z knihoven, kterou nabízí .NET Framework. Jedná se buď o Windows Forms (WinForms) nebo Windows Presentation Foundation (WPF). Kromě toho ještě existuje Universal Windows Platform (UWP), který je součástí Windows UI knihovny, nebo realizace GUI pomocí WinApi. S WPF, UWP, či WinApi mám minimální zkušenosti. Naproti tomu ve WinForms jsem realizoval část bakalářské práce a mnoho dalších aplikací, takže s ním zkušenosti mám. Z tohoto důvodu byl k realizaci grafického uživatelského rozhraní zvolen Windows

Forms, které nabízí dostatečnou funkcionalitu a výkon pro tento typ aplikace. Windows Forms má i grafický návrhář, který je součástí Microsoft Visual Studio. Více informací o knihovnách pro tvorbu uživatelského grafického rozhraní jsou na webu Microsoftu [56], který za těmito knihovnamy stojí.

Obslužná aplikace pro datalogger je tedy realizována jako aplikace s grafickým rozhraním. Využito je .NET frameworku 4.0 s knihovnou WinForms. Programovacím jazykem je C#. NET framework nabízí dobře zdokumentované knihovny pro práci se sériovou linkou a i UDP. Není tedy nutné používat knihovny třetích stran pro komunikaci s dataloggerem.

2.3 Plán realizace

Plán realizace byl uzpůsoben probíhající pandemii, která plně udeřila v roce 2020–2021. Z tohoto důvodu se začínalo s úkoly, které touto situací mohly být vážně ovlivněny. Jedná se zejména o nákup součástek. Některé součástky nebyly dostupné v České republice, dokonce ani v rámci Evropské unie a musely být objednávány z Číny. Po nákupu součástek se realizoval datalogger. Bylo potřeba co nejdříve navrhnout a vytvořit plošný spoj, protože se podobně jako u nákupu součástek jedná o něco, co mohlo být pandemií taktéž ovlivněno. Návrh a výroba plošného spoje, ale byla možná v době, kdy byl datalogger realizovaný na nepájivém poli. K ověření funkčnosti obvodu bylo nutné, aby byla realizována základní kostra firmwaru. Po úspěšné realizaci dataloggeru (včetně výroby plošného spoje) přišla na řadu PC aplikace. Tato aplikace byla sice na realizaci časově náročná, ale neobsahovala žádná úskalí, která by mohla způsobovat komplikace. Na závěr probíhalo testování celého systému. Plán realizace byl tedy následující:

- **Březen–Srpen 2020:** Analýza možností, jak měřit/simulovat spektrum ionizujícího záření, výběr a nákup všech potřebných součástek.
- **Září 2020:** Návrh systému, realizace firmwaru na Wi-Fi modul.
- **Říjen–Listopad 2020:** Realizace firmwaru na řídicí jednotku.
- **Prosinec 2020:** Návrh a výroba plošného spoje.
- **Leden–Únor 2021:** Osazení desky plošných spojů součástky, realizace PC aplikace.
- **Březen 2021:** Testování jednotlivých částí a systému jako celku.

Realizace dataloggeru

Realizaci dataloggeru lze rozložit na tři dílčí úkoly. Prvním úkolem je implementace firmwaru pro řídicí jednotku. Tento úkol zahrnuje i implementaci podpůrných funkcí pro práci s periferiemi, či výběrem vhodných knihoven třetí strany. Druhým úkolem je implementace firmwaru pro Wi-Fi modul. Posledním úkolem je návrh a výroba plošného spoje. Následující podkapitoly blíže popisují jednotlivé dílčí úkoly.

3.1 Konfigurace hodinového signálu

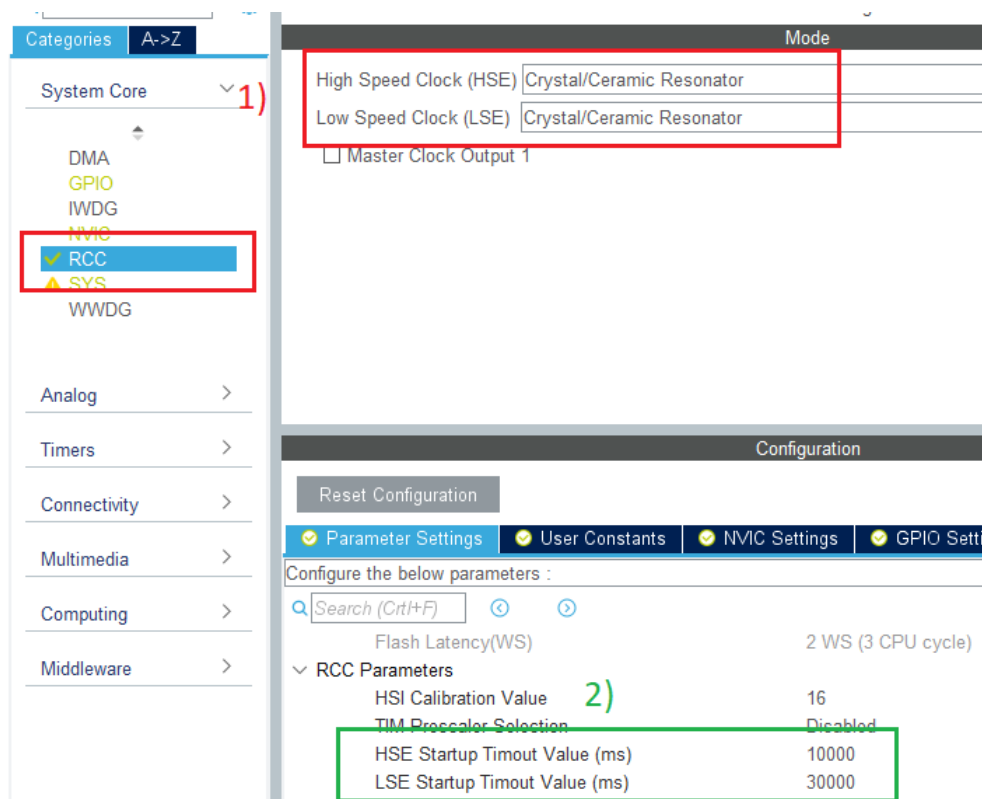
Jako první je nutné nastavit hodinový signál, aby měl požadovanou frekvenci. Mikrokontroléry STM32 mají poměrně složitý rozvod hodin, který se označuje jako clock tree. Složitost rozvodu hodin se liší v závislosti na typu mikrokontroléru STM32. Clock tree má několik fází, které jsou konfigurovatelné. Aby byla konfigurace úspěšná, je potřeba všem používaným perifériím, periferním sběrnicím a jádru CPU dodat hodinový signál se správnou frekvencí. Většinou je tato frekvence daná povoleným rozsahem, ale některé periferie, jako např. USB vyžadují přesně definovanou frekvenci hodin (48 MHz). Konfigurace clock tree je možné provádět přímo v programu STM32CubeMX v záložce Clock Configuration. Tento konfigurátor je schopný označit chybně nastavené větve clock tree. Pokud si s konfigurací hodinového signálu skutečně nevíme rady, je možné využít řešič problémů (Resolve Clock Issues), který se pokusí tyto problémy vyřešit automaticky.

Ještě před samotnou konfigurací je nutné pochopit jakými zdroji hodinového signálu mikrokontrolér STM32F401 disponuje. Prvním oscilátorem je interní vysokorychlostní RC oscilátor označován jako HSI (High Speed Internal). Dalším oscilátorem je dedikovaný externí vysokorychlostní krystalový oscilátor označován jako HSE (High Speed External). HSE krystal není součástí mikrokontroléru, je ho proto třeba dodat na PCB a připojit na piny pro tento krystal určené (vývojové desky tento krystal mohou obsahovat). Dalším

typem je nízkorychlostní RC oscilátor, označován jako LSI (Low Speed Internal). Posledním typem je nízkorychlostní externí oscilátor, označován jako LSE (Low Speed External). Podobně jako HSE krystal i LSE krystal je třeba mikrokontroléru dodat. Vysokorychlostní zdroj hodinového signálu se používá pro chod mikrokontroléru a periférií. Nízkorychlostní zdroj hodinového signálu zajišťuje jednotku hodin reálného času, či watchdogy. Aktuálně vystává otázka, proč jsou vůbec potřeba externí oscilátory, které pouze zabírají místo na desce plošných spojů, když STM32 nabízí alternativu v podobě interních zdrojů hodinového signálu. Výhodou externích oscilátorů je to, že jsou přesnější, což může být pro některé aplikace klíčové (např. USB). Nevýhodou je, že po napájení nabíhají delší dobu než interní oscilátory (vlastní pozorování během vývoje dataloggeru).

Kvůli vyšší přesnosti oscilátorů se v dataloggeru používají externí zdroje hodin - jak HSE, tak i LSE. Použití LSE je dokonce nutné, pokud se využívá interní RTC napájená knoflíkovou baterií viz 3.10.5 Nastavování čítače pro jednotku RTC. Před konfigurací clock tree je potřeba zjistit jakou frekvenci externí oscilátory mají. Externí oscilátory jsou dané vývojovou deskou, proto se nahlíží do dokumentace WeAct Black Pill [32], kde je informace že HSE má frekvenci 25 MHz a LSE 32768 Hz. Nyní je možné konfigurovat hodiny a postup je následovný:

1. V STM32CubeMX v Pinout & Configuration se v položce RCC (která je k nalezení v System Core) vybere jako zdroj HSE a LSE Crystal/Ceramic Resonator, viz Obrázek 3.1.
2. Jako RCC Parameters se zvýší startovací čas oscilátoru (HSE popř. LSE Startup Timeout Value), alespoň na 5000 ms, viz Obrázek 3.1. Bez změny startovací času oscilátorů se stávalo to, že datalogger nenaskočil korektně.
3. V RTC (položka Timers) se povolí jednotka hodin reálného času. Je to z důvodu, že kdyby nebyla povolena, nebylo by možné v Clock Configuration měnit zdroj hodin pro tuto jednotku.
4. V záložce Clock Configuration se Input Frequency upraví na 25 MHz (frekvence HSE), v PLL Source Select se vybere HSE. Frekvence se za výstupem multiplexoru vydělí 25 (s 1 MHz se lépe počítá), krok popisuje Obrázek 3.2.
5. V Main PLL se frekvence vynásobí 168 (tím se získá frekvence 168 MHz) a následně vydělí 2 (frekvence 84 MHz). Poté se v System Clock Mux vybere jako vstup PLLCLK (na SYSCLK se přivede 84 MHz který je nastaven pomocí Main PLL), viz Obrázek 3.2.
6. Poté se upraví AHB prescaler na 1, APB1 prescaler na 2 a APB2 prescaler na 1. Krok popisuje Obrázek 3.3.



Obrázek 3.1: Konfigurace hodin - výběr oscilátoru, nejprve se vybere zdroj hodin, poté se zvýší startovací timeout.

7. RTC Clock MUX se upraví na LSE.

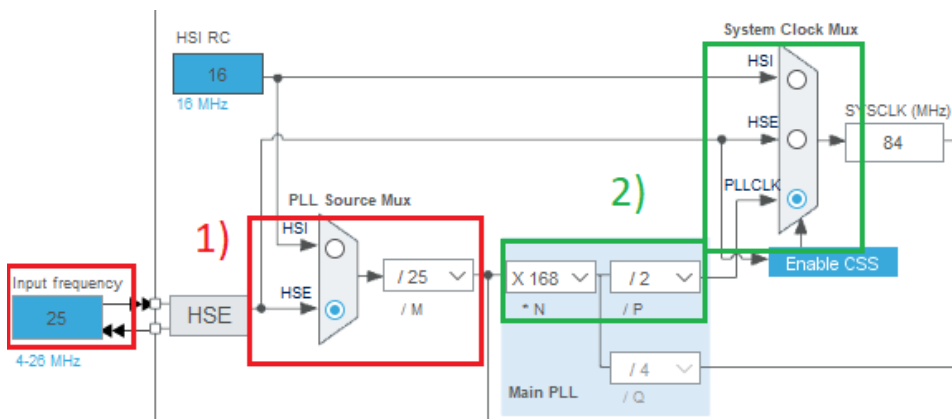
Tímto je konfigurace hodinového signálu hotová, využívá se maximální možná povolené frekvence mikrokontroléru 84 MHz. Frekvenci je možné kdykoliv stejným postupem upravit – nejjednodušší způsob je změna Main PLL. Výslednou Clock Configuration ukazuje Obrázek 3.4.

Další informace o clock tree a konfiguraci hodin je k nalezení v knize [28].

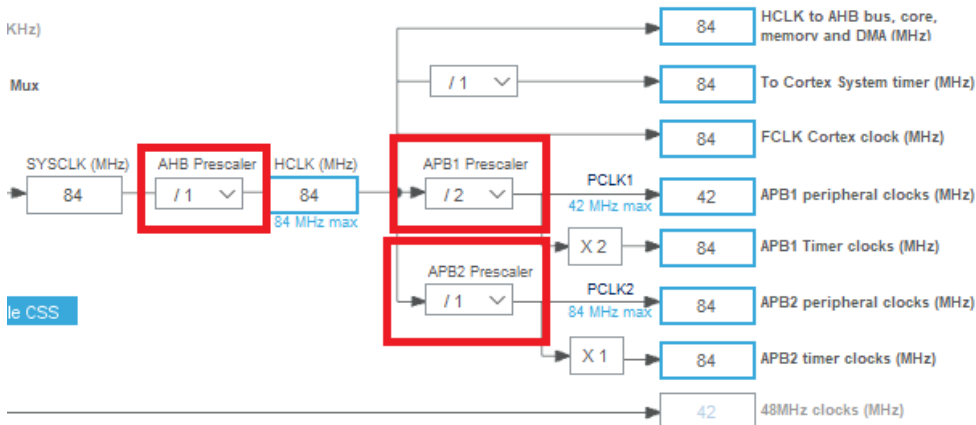
3.2 Konfigurace dataloggeru

Datalogger nabízí možnost konfigurace uživatelem pomocí PC aplikace. Uživatel může nastavit funkce a vlastnosti jednotlivých prvků zařízení a spravovat bezdrátové síť uložené v paměti dataloggeru. Nastavení funkcí a vlastností dataloggeru je uchováno v patnácti 4bajtových registrech. Každý registr má pevně definovaný formát bitů. Tyto registry jsou blíže popsány v podkapitole 3.2.2 Stavové hodnoty dataloggeru. Aby hodnoty registrů byly zachovány i

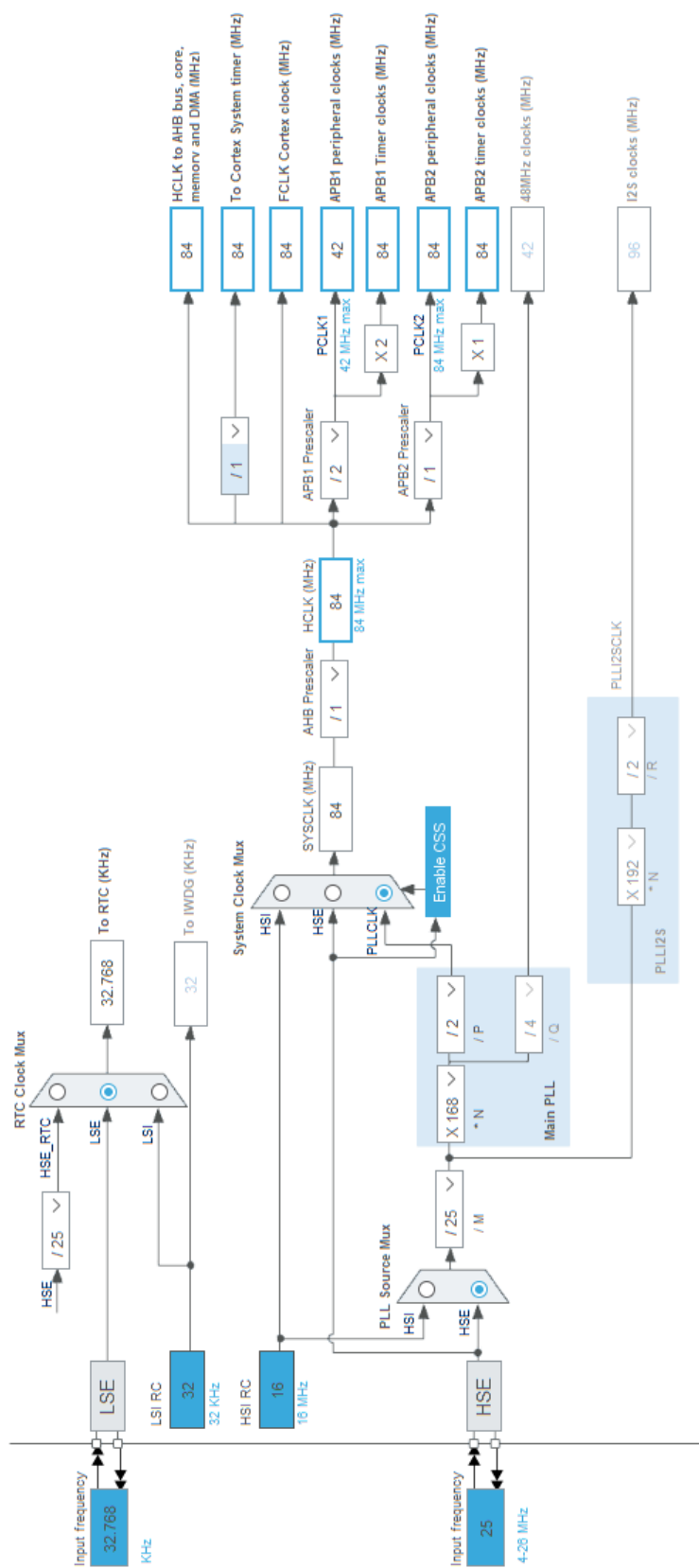
3. REALIZACE DATALOGGERU



Obrázek 3.2: Konfigurace hodin - Nastavení systémových hodin. Nejprve se vybere jako zdroj hodin HSE, frekvence se vydělí 25. Poté se v hlavním PLL nastaví násobení 168 a dělením 2. Jako zdroj hodinového signálu pro systémové hodiny se vybere PLLCLK.



Obrázek 3.3: Konfigurace hodin - Nastavení hodin sběrnic. To spočívá v tom, že se nastaví AHB a APB2 prescaler na 1 a APB1 prescaler na 2.



Obrázek 3.4: Výsledná konfigurace hodin v STM32CubeMX.

po ztrátě napájení, ukládají se do EEPROM. Datalogger má k dispozici dva moduly EEPROM STM M24C16-WBN6P, do kterého ukládá konfigurační registry. Registry se ukládají v režimu zrcadlení. Na obou EEPROM jsou stejná data.

3.2.1 Komunikace s EEPROM

Modul EEPROM M24C16-WBN6P se k mikrokontroléru STM32 připojuje pomocí I²C rozhraní. Adresování modulu probíhá tím způsobem, že řídicí jednotka na I²C sběrnici vystaví 7bitovou I²C adresu EEPROM, poté 8bitovou datovou adresu, která určuje adresu dat s kterými se manipuluje. M24C16-WBN6P má kapacitu 2048 bajtů, 8bitová datová adresa je tedy nedostatečná. Z tohoto důvodu se nejvyšší tři bity datové adresy vysílají jako součást I²C adresy, která má fixní prefix 0b1010. Nejméně významné tři bity I²C adresy odpovídají nejvíce významným třem bitům datové adresy.

Aby bylo možné s EEPROM pracovat, bylo nutné implementovat funkci pro čtení a zápis za použití HAL_I2C funkcí. Sekvenční čtení dat z EEPROM se provádí tím způsobem, že se nejprve provede tzv. dummy write. To spočívá v tom, že se na I²C sběrnici zapíše adresa dat, odkud se mají číst. Tento zápis se dělá bez posláni stop podmínky. Poté je nutné vyčkat, než EEPROM tato data načte. Po načtení je možné data číst pomocí I²C čtecí operace. EEPROM v sobě obsahuje interní adresní čítač, takže se data postupně odesílají přes rozhraní do doby, dokud řídicí jednotka nevyšle stop podmínku.

Zápis do EEPROM je možný buď po bajtech, nebo po stránkách. Psaní po stránkách umožňuje v jednom psacím cyklu zapsat celou stránku. Stránka má velikost 16 bajtů a začíná vždy na adrese, která je dělitelná 16. V M24C16-WBN6P je celkově 128 stránek. Zápis po stránce se provádí tak, že se postupně na sběrnici pošle 16 bajtů, který je zakončený stop podmínkou. Po odeslání 16 bajtů je nutné vyčkat do doby, než se data v EEPROM uloží. To se dělá tím způsobem, že se EEPROM modul vybere – na sběrnici se vystaví jeho adresa a čeká se do doby kdy výběr potvrdí. Firmware dataloggeru umožňuje zápis pouze po stránkách. Data, která jsou potřeba zapsat se rozdělí na 16bajtové fragmenty, které se postupně zapíší do EEPROM. Pokud velikost dat, která se zapisují nejsou dělitelná 16, tak se data zarovnají nulami.

Během implementace kódu pro práci z EEPROM jsem vycházel z datasheetu EEPROM [44] a knihy *Mastering STM32* [28], kde je popsán princip práce s jiným typem EEPROM.

3.2.2 Stavové hodnoty dataloggeru

Stavové hodnoty dataloggeru určují nastavení senzorů, Wi-Fi modulu, či měření. Stavové hodnoty se ukládají do tzv. konfiguračních registrů. Datalogger má k dispozici celkově 15 konfiguračních registrů, každý registr odpovídá určité části dataloggeru. Tři nejvyšší bity každého registru jsou chybové pří-

Tabulka 3.1: Konfigurační registry

Index:	Typ registru	Popis
0	Nastavení ESP-01	Obsahuje, zdali je ESP-01 vypnut a UDP port
1	Nastavení startu měření	Obsahuje informaci o typu startu a určuje čas prodlevy mezi startem dataloggeru a měřením
2,3	Nastavení interních popř. externích RTC	Obsahuje informaci o tom, zdali je daný RTC vypnuto a zdali je napájeno baterií.
4,5	Nastavení SD čtečky 0 popř. 1	Obsahuje informaci o tom, zdali je daná SD-čtečka připojena.
6	Příznaky EEPROM	Obsahuje příznak, zdali se změnila konfigurace, či je vyžadován reset.
7	Nastavení pravidelných akcí	Obsahuje časy pravidelných akcí - kontrola sítě, aktualizace mezipaměti ESP-01 a zápis na paměťovou kartu.
8	Nastavení měřicí jednotky	Obsahuje nastavení měřicího zařízení jako třeba typ spouště, přesnost A/D převodníku atd..
9	Nastavení pomocných měřicích jednotek	Obsahuje nastavení obou pomocných měřicích jednotek - typ spouště, či zdali se počítá CPM.
10	Nastavení hodin vypínání ESP-01	Obsahuje informace o tom v jakých hodinách se má ESP-01 vypnout.
11	Nastavení dnů vypínání ESP-01	Obsahuje informace o tom v jakých dnech se má ESP-01 vypnout. Je nadřazené vypínacím hodinám.
12	Nastavení BME280	Obsahuje zdali je BME280 vypnuto, či konfiguraci senzoru tlaku/teploty/vlhkosti.
13,14	Rezervováno	Nevyužité registry

znaky, které indikují chyby části dataloggeru, které s daným registrem souvisí. Zbývající bity se již liší typ od typu registru a nejsou jednotné. V dataloggeru se s těmito registry pracuje jako s 15prvkovým polem bezznaménkových 32bitových čísel. Význam jednotlivých registrů popisuje Tabulka 3.1.

Konfigurační registry se načítají z EEPROM do paměti mikrokontroléru během inicializace zařízení. Načítání probíhá tím stylem, že se přečtou registry z obou EEPROM a uloží do paměti. Ne vždy se využívá všech 32 bitů registru, z tohoto důvodu se na tyto registry použije maska, která vynuluje nevyužité bity. Poté se postupně porovnají všechny registry načtené z první a

Tabulka 3.2: Konfigurace bezdrátových sítě

Offset:	Popis
0x00	Příznaková proměnná. Určuje zdali je konfigurace aktivní, v případě konfigurace přístupového bodu obsahuje kanál a maximální počet klientů. V případě konfigurace bezdrátové sítě typ šifrování.
0x02	SSID, délka je 33 bajtů. Maximální povolená délka SSID je 32 a 1 bajt je pro ukončovací nulu.
0x23	Heslo, délka je 65 bajtů. Maximální povolená délka hesla je 64 bajtů a 1 bajt je pro ukončovací nulu.
0x64	Statická IP adresa.
0x68	Maska podsítě.
0x6C	Výchozí brána.

druhé EEPROM. Pokud je hodnota registru z obou EEPROM stejná je použita a nahrána do paměti řídicí jednotky. Pokud hodnoty totožné nejsou, tak některý z bitů je poškozený. Pomocí duplexu není možné chybu opravit, ale pouze detekovat, takže se používá výchozí nastavení registru. Registry se maskují z toho důvodu, aby chyba v nevyužitém bitu nezpůsobila chybné načtení registru.

Aby bylo omezeno opotřebování EEPROM, tak se po dobu běhu dataloggeru pracuje (zápis/čtení) výhradně s registry načtenými v RAM. Uživatel ale může explicitně vynutit zápis registrů z RAM do EEPROM. Před samotným zápisem do EEPROM se kontroluje jestli se změnila konfigurace. Pokud se konfigurace nezměnila, k žádnému zápisu nedochází. Opět se jedná o opatření, které má zabránit opotřebování EEPROM. Registry se do EEPROM ukládají bez chybových příznaků a zápis se provede na oba paměťové moduly.

3.2.3 Konfigurace bezdrátových sítí

Datalogger umožňuje uložit konfiguraci až 3 bezdrátových sítí a jednu konfiguraci přístupového bodu. Každá konfigurace sítě má velikost 112 bajtů, mezi jednotlivými konfiguracemi je výplň 16 bajtů. Konfiguraci bezdrátové sítě a přístupového bodu popisuje Tabulka 3.2.

Pořadí konfigurace bezdrátových sítí je důležité. Podle pořadí se určuje pořadí jak se bude datalogger zkoušet připojit k sítím. První tři konfigurace odpovídají tomu s jakými parametry se bude Wi-Fi modul připojovat k bezdrátové síti. Poslední konfigurace odpovídá tomu s jakými parametry spouští Wi-Fi modul přístupový bod. Přístupový bod se spouští jen tehdy, pokud připojení k ostatním sítím skončí neúspěchem. Konfigurace bezdrátových sítí je stejně jako konfigurační registry uložena v EEPROM a načítají se během inicializace dataloggeru. Podobně jako u konfiguračních registrů se porovnávají

konfigurace z obou EEPROM a pokud nejsou totožné, tak se nevyužívají. Pokud jsou totožné načítají se do paměti mikrokontroléru. Stejně jako v případě konfiguračních registrů se po dobu běhu Dataloggeru pracuje přímo pouze s konfiguracemi bezdrátových sítí uložených v RAM. Pokud chce uživatel změnu konfigurace sítí uložit do EEPROM musí to explicitně vynutit.

3.3 Měření reálného času

Datalogger má k dispozici 2 nezávislé jednotky hodin reálného času (interní v mikrokontroléru a externí DS3231 modul). Každá jednotka má vlastní záložní baterii CR2032, která zajišťuje, že jednotka neztratí přesný čas i po výpadku hlavního napájení. S interní se komunikuje pomocí HAL_RTC funkcemi. Konfigurace interní jednotky RTC je komplikovanější a blíže jí popisuje podkapitola 3.10.5 Nastavování čítače pro jednotku RTC. Externí jednotka RTC je připojena přes I²C, funkce pro externí RTC jsem implementoval sám dle dokumentace DS3231 [53] a HAL_I2C funkcí. Hlavním úkolem těchto jednotek je tzv. synchronizace reálného času. Pokud synchronizace reálného času proběhne v pořádku, tak se přidává třetí způsob měření času a to je měření reálného času pomocí časovače se sekundovým přerušením (dále v textu jako interní hodiny). Interní hodiny mají čas uložený ve formátu Unixového času. Jedná se o pojistku, kdyby některá (či dokonce obě) z jednotek selhaly. Interní hodiny udržují reálný čas do doby, dokud není odpojeno napájení, či se datalogger neresetuje.

3.3.1 Synchronizace reálného času

Synchronizace reálného času nastavuje přesný systémový čas dataloggeru. Synchronizace probíhá během inicializace dataloggeru (během startování dataloggeru) pomocí jednotek hodin reálného času, či kdykoliv za běhu pomocí počítače. Pokud se synchronizuje čas pomocí počítače, tak nedochází k žádným kontrolám zadaného času, pouze se nastaví obě jednotky a interní hodiny na daný čas.

Synchronizace pomocí jednotek hodin reálného času je komplikovanější. Probíhá ve třech fázích. V první fázi se čtou údaje (datum a čas) z obou jednotek. Během této fáze probíhá triviální test, který je schopný detekovat selhání jednotky (nelze s danou jednotkou komunikovat), či vybití baterie. Vybitá baterie se kontroluje tím způsobem, že se zkontroluje rok, v případě, že došlo k vybití baterie (či odebrání baterie) tak přečtený rok je menší než je aktuální rok (v době psaní firmwaru to byl rok 2020). Pokud triviální test proběhne v pořádku, tak se daná jednotka označí jako důvěryhodná.

Druhá fáze je porovnání časů obou jednotek. Porovnání lze dělat pouze za předpokladu, že obě jednotky jsou triviálním testem označeny jako důvěryhodné. Pokud ani jedna není důvěryhodná tak synchronizace času končí neúspěchem. Pokud pouze jedna z jednotek je důvěryhodná, tak se bere jako

referenční a pomocí ní se nastaví nedůvěryhodná jednotka a interní hodiny. Čas a datum ale nemusí být správný, synchronizace tedy končí s chybou. Pokud jsou obě jednotky důvěryhodné probíhá porovnání času. To se dělá tím způsobem, že se datum a čas z obou jednotek převede na Unixový čas a porovná se absolutní hodnota jejich rozdílu s tolerancí. Pokud je hodnota v toleranci (tolerance je nastavena na 15 minut) tak čas lze považovat za správný. Pokud není, synchronizace končí neúspěchem, protože není možné zjistit, která jednotka ukazuje špatný čas.

Třetí fází je nastavení interních hodin. Pro nastavení se používá externí hodiny reálného času (RTC modul). Je to z důvodu, že tato jednotka je přesnější než interní hodiny reálného času. Po nastavení interních hodin synchronizace končí úspěchem.

Synchronizace reálného času může skončit třemi různými výsledky. První výsledek je, že synchronizace proběhla v pořádku. To znamená, že během synchronizace nenastala žádná chyba. Čas je ověřený pomocí dvou různých a nezávislých jednotek hodin reálného času.

Druhý výsledek, jakým může synchronizace skončit, je synchronizace proběhla s chybou. To znamená, že během synchronizace nastala chyba, která znemožňuje ověření, zda je čas správný. Systémový čas je nastaven, ale nemusí být korektní (není ověřen oběma jednotkami RTC).

Poslední výsledek je selhání synchronizace, což znamená, že synchronizaci není možné provést. Systémový čas zůstává nenastaven. Selhání synchronizace reálného času je vážný problém, který znemožňuje start měření do doby, dokud není chyba odstraněna.

3.3.2 Čtení systémového času

Po úspěšné synchronizaci reálného času je možné číst systémový čas, který odpovídá reálnému času. Systémový čas se používá převážně během ukládání aktuálně naměřených dat na paměťové médium. Kromě samotných dat se ke každému záznamu přidává i časová značka. Ke čtení systémového času se využívají obě jednotky reálného času a interní hodiny. Využívá se triple modular redundancy, to znamená, že o systémovém čase hlasují všechny hodiny. Podobně jako u synchronizace reálného času se porovnává absolutní hodnota rozdílu unixových časů s maximální tolerancí 15 minut.

Pokud čtení systémového času neproběhne v pořádku (hodiny se neshodnou na čase, selžou obě jednotky RTC atd.), tak se použije datum a čas z interních hodin. Podobně jako selhání synchronizace reálného času i chybné čtení systémového času je vážný problém, který znemožňuje start měření. Avšak pokud čtení systémového času selže během měření, tak se měří dál a důvěřuje se interním hodinám. Tento fakt je ale zaznamenán do daného souboru měření a během vyhodnocování dat je potřeba na selhání hodin brát zřetel.

3.4 Ukládání a čtení naměřených dat

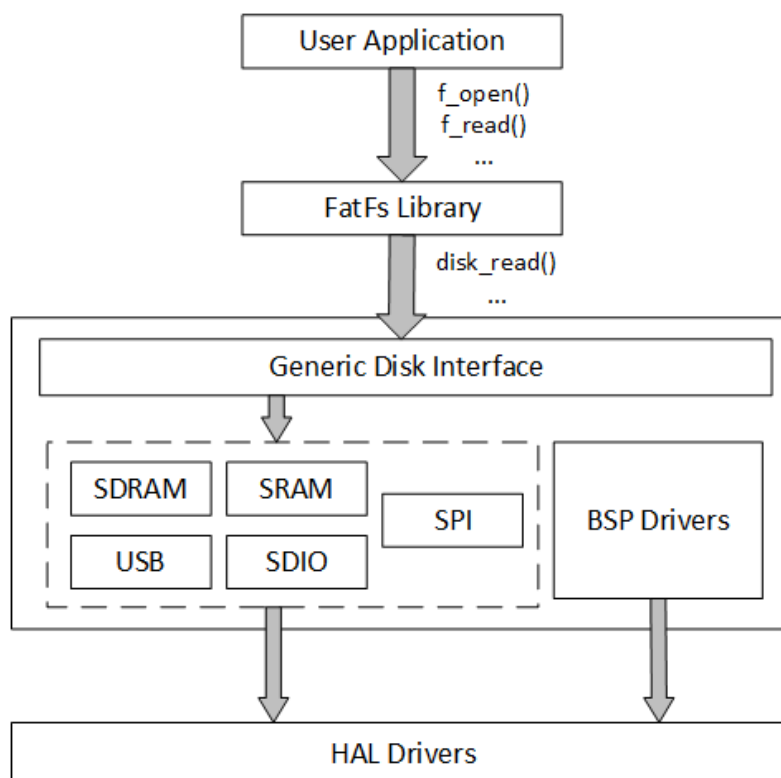
Data z měření se ukládají do binárních souborů, které se ukládají na microSD kartu naformátovanou na FAT32 filesystem. Formát těchto souborů je blíže popsán v podkapitole 3.4.2 Formát souboru měření. Datalogger disponuje dvěma microSD čtečkami, je tedy možné data z měření ukládat na dvě microSD karty a během vyhodnocování dat používat druhý soubor pro ověření konzistence. Datalogger je ale plně funkční i s pouze jednou SD kartou, nelze ale ověřit, jestli data nejsou poškozena, či v případě selhání karty dojde k částečné (či úplné) ztrátě dat.

3.4.1 Operace s fyzickým úložištěm

informace o FatFS (generický FAT filesystem pro malé vestavěné systémy) jsou převzaty z následujících z knihy [28] a oficiální dokumentace Chan FatFs [57]. Aby bylo možné pracovat s microSD kartou, která je zformátovaná na FAT32, je potřeba mít funkce, které umožňují pracovat s tímto filesystemem. STM32CubeMX umožňuje programátorům do jejich projektu zahrnout knihovnu FatFs. Knihovna FatFs nabízí funkce pro práci s filesystemem. Tato knihovna je kompletně separovaná od I/O vrstvy, aby bylo možné tuto knihovnu použít, musí se doimplementovat funkce komunikující s daným paměťovým médiem. ST nabízí implementované knihovny pro některé typy paměťových úložišť, bohužel SPI čtečky microSD karet mezi to nepatří. Je proto nutné v STM32CubeMX vybrat user-defined filesystem a tuto I/O vrstvu získat svépomocí. Fungování Chan FatFs popisuje Obrázek 3.5. U Chan FatFs knihovny je možné odpojit část funkcí a získat tím kompaktnější velikost kódu. Datalogger ale vyžaduje většinu funkcí knihovny, využívá proto nejnáročnější variantu této knihovny.

Implementace I/O vrstvy se ukázala jako poměrně náročná (ale i zbytečná) práce, ke všemu vyvíjení vlastního řešení může do firmwaru dataloggeru vnést zbytečné chyby. Proto je použita knihovna třetí strany [58]. Knihovna ale počítá pouze s jednou SPI čtečkou, avšak datalogger obsahuje dvě stejné čtečky. Bylo nutné danou knihovnu upravit tak, aby fungovala s více čtečkami, postup úpravy je následující:

1. V konfiguraci uživatelem definovaného FatFS v STM32CubeMX změnit počet logických disků na 2.
2. Upravit funkce `sd_disk_initialize`, `sd_disk_status`, `sd_disk_write`, `sd_disk_ioctl` ve `fatfs_sd.c`, aby komunikovaly se správnou čtečkou. V případě firmwaru dataloggeru je to řešeno, že nejsou přímo volané již zmíněné `sd` funkce, ale pro každou čtečku je implementována funkce, která nastaví správné SPI a CS a až poté se zavolá požadovaná `sd` funkce dodaná s knihovnou.



Obrázek 3.5: Jak FatFs knihovna komunikuje s hardwarem. Pro správnou funkci je třeba doplnit Low-level Disk I/O. Zdroj [28].

3. Posledním krokem je nově implementované funkce pro výběr dané čtečky přilinkovat FatFs knihovně. Linkování se dělá ve funkci `MX_FATFS_Init` v souboru `fatfs.c`. Linkování spočívá v tom, že k logické jednotce se přiřadí funkce (ty, jež byly implementovány v minulém kroce) pro práci s příslušnou fyzickou jednotkou.
4. Nyní je možné používat dvě různé čtečky SD. První se adresuje pomocí logické adresy "0:/", druhá pomocí logické adresy "1:/". Každá čtečka má vlastní SPI sběrnici. Postup by fungoval i v případě, že by obě čtečky byly na jednom SPI.

Kromě toho bylo potřeba implementovat funkci `get_fattime`, která se také nalézá v souboru `fatfs.c`. Funkce `get_fattime` je nutná, pokud v konfiguraci uživatelem definovaného FatFs je nastaveno `FS_NORTC` na dynamical timestamp. Funkce `get_fattime` se využívá k získání reálného času k nastavení časové značky souboru. Vzhledem k tomu, že poslední čas úpravy souboru není pro měření důležitý (časová značka je součástí záznamů v daném souboru), tak se tento čas získává pouze pomocí interní jednotky hodin reálného času. Ne-

Tabulka 3.3: Formát záznamu

Offset:	Velikost(B)	Hodnota
0x0	8	Datum a čas
0x8	8	Chybové příznaky
0x10	4	Teplota
0x14	4	Tlak
0x18	4	Vlhkost
0x1C	16	Data z pomocných měřicích zařízení
0x2C	0 – 16384	Data z analogové měřicí jednotky (v závislosti na počtu kanálů)

probíhá žádná kontrola správnosti času. Po doplnění funkce `get_fattime` už knihovna FatFs funguje dle očekávání.

3.4.2 Formát souboru měření

Původně jsem pro ukládání dat z měření chtěl využít csv formát. Tento formát je pro datalogger datově objemný, což není vhodné. Je to z důvodu, že operace s microSD kartou jsou časově náročné, čím víc dat je třeba zapsat, tím to déle trvá. Podobně i přenos větších souborů trvá déle. Z tohoto důvodu je navržen binární datový formát pro ukládání dat z měření. V binárním formátu se měření ukládají na straně dataloggeru. Počítačová aplikace binární formát umí číst a následně převést na csv.

Soubor se skládá z hlavičky a záznamů. Hlavička se nalézá na začátku souboru a obsahuje informace o měření. Pomocí této hlavičky je možné vypočítat velikost každého záznamu. Hlavička má velikost 64 bajtů a je zabezpečena CRC32, který je umístěn na konci hlavičky.

Záznam má formát který popisuje Tabulka 3.3. Offset je daný v rámci daného záznamu. Velikost každého záznamu je proměnlivá v závislosti na počtu kanálů analogové měřicí jednotky spektra ionizujícího záření. Každý záznam odpovídá aktuální hodnotě všech senzorů v době, kdy se záznam prováděl.

Počet záznamů je omezený pouze limitem filesystému pro maximální velikost souboru, či volnou kapacitou SD karty. Výhodou tohoto formátu je, že všechny záznamy v jednom měření (během měření nelze měnit nastavení konfigurace) mají stejnou velikost. Je tedy jednoznačně daný offset, kde se nachází jakýkoliv záznam. V případě csv má každý záznam proměnlivou velikost. Jednotná velikost záznamů je vykoupena obrovským množstvím zbytečných dat navíc. Zpracování takového souboru je časově náročnější.

Kvůli přehlednosti obsahují soubory měření ve jménu datum a čas vytvoření souboru (RRMMDD_hhmmss.DAT²). Koncovka DAT identifikuje, že se

²R = Rok, M = Měsíc, D = Den, h = hodina, m = minuta, s = sekunda. Každý údaj je dlouhý dva znaky, používá se zarovnání nulou a rok je brán modulo 100.

jedná o soubor měření. Datalogger umožňuje mazat pouze soubory s koncovkou .DAT.

3.5 Měření spektra ionizujícího záření

Energie ionizujícího záření se měří pomocí analogové měřicí jednotky, která je připojena na A/D převodník řídicí jednotky. Pomocí A/D převodníku se zjistí maximální hodnota signálu. Změřená hodnota odpovídá energii, kterou kvanta záření ztratila v měřicí jednotce. Každá možná hodnota rozsahu A/D převodníku odpovídá jednomu kanálu, který se inkrementuje o jedna v případě, že je kvanta této energie detekována. Zanesením kanálů do grafu na osu x a počtu impulzů daného kanálu na osu y získáme spektrum ionizujícího záření.

STM32F401CCU6 má k dispozici jeden A/D převodník s maximální vzorkovací frekvencí 2,4 MSample/s. Konverze s 12bitovým rozlišením trvá 15 ADC cyklů. Jeden ADC cyklus odpovídá taktu hodin A/D převodníku, který odpovídá v případě mikrokontroléru dataloggeru taktu sběrnice APB2 vydělený hodnotou 2, 4, 6, či 8. Sběrnice APB2 je taktována na 84 MHz. S frekvencí sběrnice APB2 nelze dosáhnout vzorkovací frekvence 2,4 MSample/s, ale pouze 1,4 MSample/s³. Délka konverze je cca. 0,714 μ s. Používá se taktovací frekvence APB2 vydělená 4. Hodiny A/D převodníku mají frekvenci 21 MHz.

Pro práci s A/D převodníkem se využívají funkce HAL_ADC. Používají se pouze varianty těchto funkcí, které pracují v neblokujícím režimu. To znamená, že se spustí konverze, která neblokuje procesor od vykonávání dalších instrukcí. Po skončení konverze je vyvoláno přerušení, které spustí callback, kde se s přečtenou hodnotou dále pracuje.

Datalogger podporuje dva způsoby, které určují jakým způsobem se spouští konverze. Prvním způsobem je spouštění konverze pomocí spouště. Druhý způsob je pokračované měření signálu. V případě pokračovaného měření se konverze spouští v pravidelných intervalech a průběžně se hledají maxima. V následujících podkapitolách budou oba způsoby blíže popsány.

3.5.1 Konverze pomocí spouště

Prvním způsobem, kterým datalogger spouští konverzi, je konverze pomocí spouště. To znamená, že měřicí jednotka má kromě analogového výstupu ještě digitální výstup, který signalizuje, že je signál připraven na přečtení A/D převodníkem. Tento výstup se nazývá spoušť. Spoušť je připojena na pin podporující externí přerušení. Pokud v signálu proběhne uživatelem definovaná hrana spouště (náběžné, sestupná, obě), vyvolá se přerušení v jehož obsluze se spustí konverze A/D převodníku. Po provedení konverze se zkonvertovaná hodnota přečte, popř. logicky posune doprava o uživatelem definovaný posun

³Maximální povolené frekvence hodin A/D převodníku je 36 MHz.

(v rozsahu 0–12), který softwarově snižuje rozlišení A/D převodníku. Tato hodnota určuje kanál, který se má inkrementovat o 1.

V režimu konverze pomocí spouště je možné aktivovat volitelný signál, který určuje zdali probíhá konverze. Signál se nastavuje do logické 1 v době, kdy A/D převodník začne konverzi. V tuto dobu už je hodnota napětí držena na interním kondenzátoru A/D převodníku a změna napětí v signálu konverzi nijak neovlivní⁴. Signál přechází do logické 0 v době kdy se zkonvertovaná hodnota převede na kanál a daný kanál se inkrementuje o 1. Po přivedení signálu *konverze probíhá* na osciloskop je možné pozorovat, že trvá déle než 0,714 μs . Je to z důvodu, že tento signál neodpovídá přímo konverzi ADC, ale době, kdy není datalogger schopný zpracovat další impuls. To odpovídá době, dokud není dokončen callback určující konec konverze. Signál *konverze probíhá* se může využít k vybití kondenzátorů v měřicím zařízení, pokud obsahuje např. obvod detekující vrchol signálu. Kromě toho se signál *konverze probíhá* využívá k experimentálnímu testování vlastností dataloggeru jako třeba prodleva mezi spouští a začátkem konverze, délka mrtvé doby atd..

Právě signál *probíhá konverze* byl využit během realizace dataloggeru ke změření vlastností konverze pomocí spouště. Byl využit osciloskop, první sonda byla připojena na signál spoušť a druhá sonda na signál konverze probíhá. Spoušť byla nastavena na sestupnou hranu. Prodleva mezi spouští a začátkem vzorkování signálu byla přibližně 2,8 μs . Délka vytíženosti A/D převodníku byla přibližně 4,6 μs . Aby byl signál dataloggerem dobře přečten, je nutné, aby maximální hodnota vydržela po aktivaci spouště alespoň 3 μs . Mrtvá doba je v nejlepším případě 8 μs , která odpovídá době mezi spouští a konverzí a pak samotným vzorkováním a zpracováním dat. Pokud je spoušť spuštěna v době, kdy právě probíhá konverze, je spoušť ignorována. Průběh obou signálů popisuje Obrázek 3.6.

3.5.2 Pokračované měření signálu

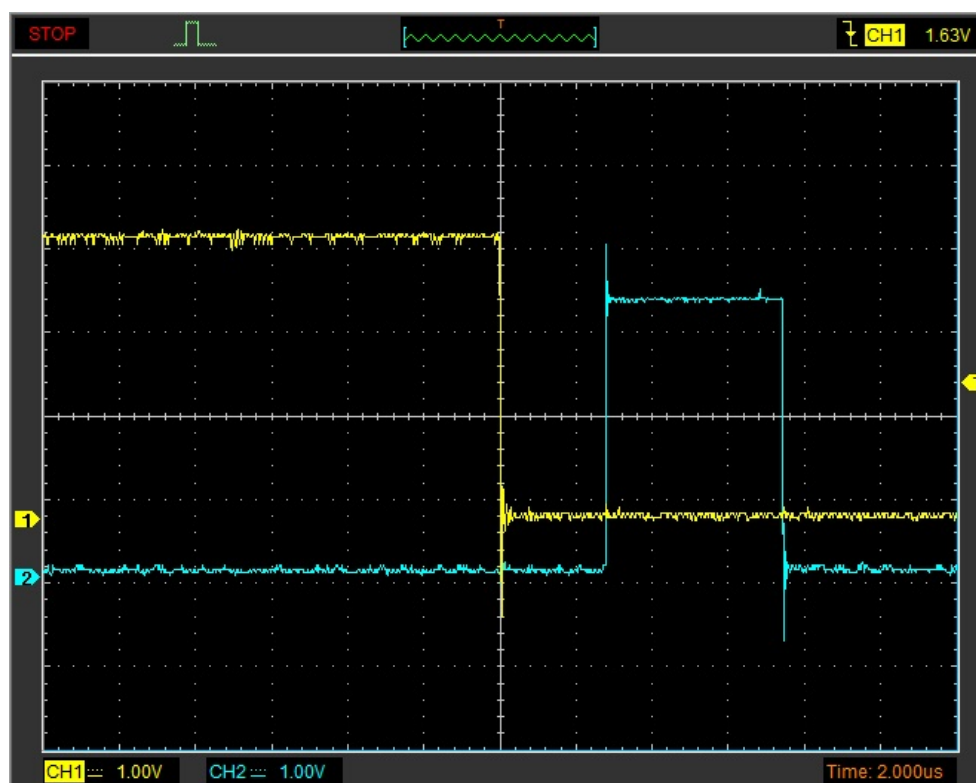
Pokračované měření signálu funguje na principu, že analogový signál z měřicí jednotky je v pravidelných intervalech vzorkován a průběžně jsou hledaná maxima impulsů. Vzorkování probíhá s frekvencí 48 kHz. Výběr vzorkovací frekvence se probírá v podkapitole 3.10.4 Časovač pro A/D převodník. Pokračované měření nevyužívá spoušť, ale automaticky detekuje impulsy. Signál ze spouště je ignorován. Během pokračovaného měření není možné spouštět signál *probíhá konverze*.

Pokračované měření funguje na principu, že pokud není měřicím zařízením detekováno kvantum záření, na jeho výstupu je napětí, které nepřekročí určitý práh⁵. V případě detekce kvanta ionizujícího záření se vytvoří impuls

⁴V A/D převodníku se používá tzv. Sample and Hold obvod.

⁵Napětí by bylo v ideálním případě konstantní, ale v reálném světě má toto napětí šum, který může být až několik desítek mV.

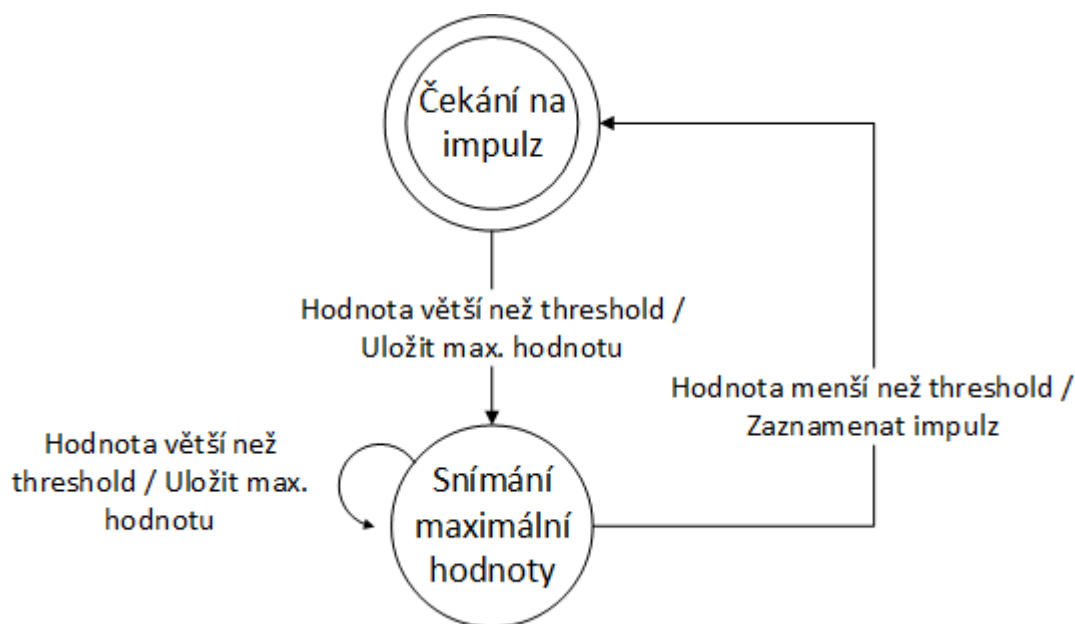
3. REALIZACE DATALOGGERU



Obrázek 3.6: Průběh konverze pomocí spouště. Žlutý signál odpovídá spoušti, tyrkysový signálu *probíhá konverze*.

(pro účely této práce Gaussovského tvaru), který tento práh překročí. Během pokračovaného měření datalogger čeká na to, až napětí, převedené na digitální hodnotu, překročí uživatelem definovaný práh (threshold). Poté se signál v pravidelných intervalech dále vzorkuje a ukládá se maximální detekovaná hodnota do doby, dokud napětí neklesne pod hodnotu práh. Když napětí klesne pod práh, tak se použije maximální detekovaná hodnota, která se logicky posune doprava o uživatelem definovaný posun. Tato hodnota určuje kanál, který se má inkrementovat o jedna. Pokračované měření popisuje diagram na Obrázku 3.7.

Prvním problémem pokračovaného měření signálu je aliasing, který nastává v případě, že impulzy nastanou těsně za sebou a datalogger je není schopný odlišit. Aliasing nastává v případech, kdy impulzy mají mezi sebou prodlevu kratší než je perioda vzorkování. Dalším problémem je to, že datalogger není schopný správně detekovat vrchol. Chybná detekce vrcholu nastává, když signál není v maximální hodnotě napětí po dobu periody vzorkování. Oba tyto problémy lze částečně potlačit rychlejším vzorkováním, ale ani se se-



Obrázek 3.7: Pokračované měření signálu, počátečním stavem je čekání na impuls. `adcValue` odpovídá hodnotě přečtené A/D převodníkem, `threshold` je práh, `peakValue` je proměnná pro ukládání maximální hodnoty.

berychlejší vzorkováním je nelze zcela odstranit. Pokračované měření signálu je doporučeno pro méně aktivní vzorky – stovky až tisíce pulsů za sekundu.

Posledním problémem je správné určení hodnoty prahu. Pokud je hodnota příliš nízká, měření může být přehlušeno šumem. Pokud je naopak příliš vysoká, datalogger nemusí být schopný zachytit nízkoenergetické částice/fotony. Před samotným měřením je potřeba změřit pomocí osciloskopu měřicí zařízení, jakou má výchozí napěťovou hodnotu a jak moc velký je šum. Podle těchto hodnot se nastaví práh na takovou hodnotu, aby šum nevytvářel falešné pulzy.

3.6 Měření CPM pomocí pomocných měřicích zařízení

K dataloggeru je možné připojit až dvě přídavná měřicí zařízení. Tyto měřicí zařízení nejsou schopné měřit energii částice/fotonu, ale umí detekovat přítomnost ionizujícího záření. Jedná se např. o detektor založený na Geiger-Müllerově trubici. Výstupem těchto měřicích zařízení je jedna logická úroveň v případě detekce, jinak druhá logická úroveň. Pomocná měřicí zařízení se připojují na piny, který podporují externí přerušení. Spouštěcí hrana přerušení je

nastavitelná v konfiguraci dataloggeru. Obsluha přerušení je jednoduchá, nejprve se zresetují příznaky přerušení, poté se zvýší čítač impulzů o jedna. Čítač určuje počet impulzů z dané pomocné měřicí jednotky od počátku měření.

Kromě toho je možné u pomocných měřicích zařízeních měřit počet impulzů za minutu (CPM). Problematikou přesného počítání impulzů za minutu jsem se podrobněji zabýval ve svojí bakalářské práci, kde se osvědčilo ukládání mezihodnot do kruhového bufferu. V pravidelných intervalech (v případě dataloggeru je to 3,75 sekund) se do kruhového bufferu ukládá počet zaznamenaných impulzů, které nastaly ve zkoumaném intervalu. Přesné časování zajišťuje přerušení od časovače. Buffer má kapacitu 16 prvků, vždy se od CPM odečte nejstarší hodnota (hodnota byla aktuální před 60 sekundami, je tedy už zastaralá), poté se k CPM přičte hodnota počtu impulzů z posledního intervalu a starý záznam se přepíše a posune se ukazatel, ukazující na nejstarší záznam kruhového bufferu. Kapacita 16 prvků nebyla vybrána náhodně, práce s kruhovým bufferem vyžaduje operaci modulo. Pokud se jedná o modulo mocninu dvou lze operaci nahradit operací AND (modulo 16 lze udělat pomocí AND 0x0F). Dělení trvá na procesoru Cortex-M4 až 12 cyklů, operace AND pouze 1 cyklus [59]. Více informací o využití kruhového bufferu pro počítání CPM je k nalezení v mé bakalářské práci [8].

3.7 Komunikace mezi PC a dataloggerem

Řídicí jednotka komunikuje s ESP-01 a počítačem pomocí sériové linky. ESP-01 je k řídicí jednotce připojeno přímo přes sériovou linku. Počítač se k dataloggeru připojuje pomocí USB, součástí dataloggeru je převodník UART na USB CP2102.

Vývojová deska WeAct Black Pill obsahuje USB-C port. Pokud by se použil vestavěný USB-C port, nebylo by nutné řešit USB převodník. Výrobce mikrokontroléru STM32 dodává knihovnu pro obsluhu USB portu v podobě middleware. Middleware pro USB spotřebovává zbytečně moc paměti ROM a RAM. Vzhledem k tomu, že je potřeba pouze virtuální sériová linka, tak se jedná o zbytečně složité řešení. Ke všemu by použití integrovaného USB-C portu způsobovalo omezení v rozvržení komponent na desce plošných spojů. Vývojová deska by musela být umístěna tak, aby byl USB port fyzicky přístupný. Kvůli těmto důvodům byla zvolena realizace pomocí převodníku CP2102. Výhodou oproti použití zabudovanému USB je to, že je možné použít stejný kód jako se využívá pro komunikaci s Wi-Fi modulem. Převodník CP2102 je připojen na UART2. Baudrate je nastaven 921600, což odpovídá maximální možné hodnotě baudrate pro CP2102 [60].

Wi-Fi modul je k řídicí jednotce připojen pomocí UART1. Baudrate je nastaveno na 1000000. U vyšší hodnoty by mohlo docházet k tomu, že by se bajty ztrácely z důvodu častého přerušení od přijímače sériové linky. Nižší hodnoty měly za důsledek to, že např. pravidelná aktualizace dat v mezipaměti

ESP-01 trvala i několik sekund.

3.8 Příkazový subsystém

Příkazový subsystém se stará o to, aby příkazy probíhaly korektně (dle návrhu Command protocolu). Kromě toho umožňuje složitější operace, které nelze realizovat pomocí jednoho příkazu, ale sérií několika příkazů, např. aktualizace dat na straně Wi-Fi modulu. Příkazový subsystém je realizován pomocí příkazového řadiče. Příkazový řadič také opakuje vyslání posledního příkazu v případě neúspěšného zpracování paketu na straně Wi-Fi modulu, či PC. V dataloggeru je pouze jeden příkazový řadič, který je sdílený pro Wi-Fi modul a počítač připojený přes USB. Pokud dorazí žádost, která vyžaduje příkazový řadič v době, kdy je tento řadič zaneprázdněn, je daná žádost zahozena (Request protokol dovoluje nevyřízení žádosti).

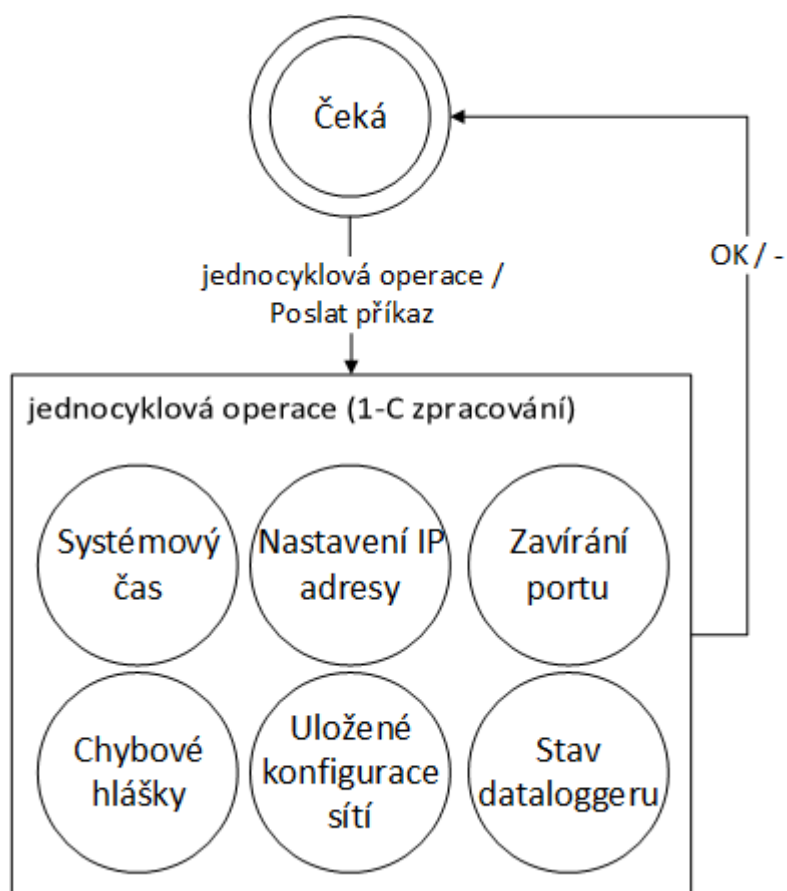
Komunikace s Wi-Fi modulem, či počítačem přes USB je nespolehlivá, druhá strana může kdykoliv selhat a tato komunikace je asynchronní, příkaz sice musí být vyřízen neprodleně, ale zpracování daného příkazu může trvat déle. Příkladem takové operace je připojování se k Wi-Fi síti, která může trvat až několik sekund. Aby příkazový řadič neuváznul, tak je nastaven timeout na 30 sekund, v případě, že ani za dobu timeoutu není řadič ve výchozím, stavu proběhne reset řadiče. 30 sekund je zcela dostačující pro všechny operace. Reset řadiče proběhne i tehdy, pokud řídicí jednotka obdrží třikrát po sobě neúspěšné zpracování paketu.

Příkazový řadič je buzen interními žádostmi. Interní žádosti se vytváří buď překladem paketu z Request protokolu, překladem Command response, nebo je vytváří hlavní řadič. Hlavní řadič může příkazový řadič zamknout, to má za důsledek, že příkazový řadič ignoruje interní žádosti, které nevygeneroval hlavní řadič. Zamknutí je možné pouze, pokud se příkazový řadič nachází v počátečním stavu (stav *Čeká*).

V následujících podkapitolách jsou probrány jednotlivé operace tohoto řadiče. Vzhledem k velikosti schématu (řadič obsahuje celkově 16 stavů), je vždy zobrazena pouze relevantní část daného řadiče, která zpracovává daný typ operace. Operace příkazového řadiče vždy začíná ve stavu *Čeká*, pokud je řadič ve stavu *Čeká*, tak to znamená, že je volný a připraven vykonat další operaci.

3.8.1 Jednocyklové operace

Jedná se o nejjednodušší operace co může příkazový řadič obsluhovat. jednocyklové obsahují pouze jeden příkaz a command response neobsahuje žádné pomocné statusy. Jednocyklové operace tedy probíhají tím způsobem, že se vyšle příkaz a čeká se pouze na úspěšné potvrzení paketu. Poté se řadič vrací do výchozího stavu. Jednocyklové operace zajišťují následující funkce a jsou posány diagramem na Obrázku 3.8:



Obrázek 3.8: Diagram jednocyklové operace, počátečním stavem je stav Čeká. Stavy v obdelníku jednocyklové operace jsou pouze symbolické a představují možnosti jednocyklových operací.

- Přesun systémového času mezi dataloggerem a Wi-Fi modulem.
- Nastavování IP adresy.
- Ukončování poslouchání na portu.
- Vypisování chybových hlášek.
- Posílání informací o uložených konfiguracích sítí.
- Posílání informací o stavu dataloggeru.

3.8.2 Operace: aktualizace dat v mezipaměti

Operace aktualizace dat v mezipaměti má za úkol přesunout data z paměti řídicí jednotky do paměti Wi-Fi modulu, či do paměti PC připojený přes USB.

Tato operace se používá pro přesun aktuálně naměřených dat. Každá operace aktualizace dat v mezipaměti umožňuje přesouvat pouze jeden typ dat – spektrum ionizujícího záření, data z BME280, data z přídatných měřicích zařízení, nebo aktuální konfiguraci dataloggeru. Data není vždy možné poslat pomocí jednoho příkazu. Příkaz pro přesun dat umožňuje v jednom příkazu přenést pouze 256 bajtů dat, spektrum ionizujícího záření může ale obsahovat až 4096 4bajtových hodnot. Z tohoto důvodu jsou data rozdělena na 256bajtové fragmenty a je vypočítána hranice, která určuje, kolik je nutno vyslat příkazů.

Příkaz pro přesun dat taktéž vyžaduje adresu v paměti Wi-Fi modulu (či PC), kam se mají data přesunout. Adresy na Wi-Fi modulu odpovídají skutečným adresám, kam se data ukládají. V případě PC aplikace se jedná pouze o symbolické adresy, které jsou přesně dané a jednoznačně identifikují o jaký typ dat se jedná. Nejedná se o logické (ani fyzické) adresy v paměti. Symbolické adresy v PC aplikaci jsou fixní a jsou natvrdo zakódovány do firmwaru dataloggeru. U adres Wi-Fi modulu to tak není (mohou se měnit při kompilaci firmwaru ESP-01). Z tohoto důvodu operace aktualizace dat v mezipaměti obsahuje i stav, který má za úkol získat adresy, kam data uložit. Stav pro získávání adres se volá pouze za předpokladu, když datalogger nemá k dispozici správné adresy. Operace aktualizace dat v mezipaměti je popsána diagramem na Obrázku 3.9.

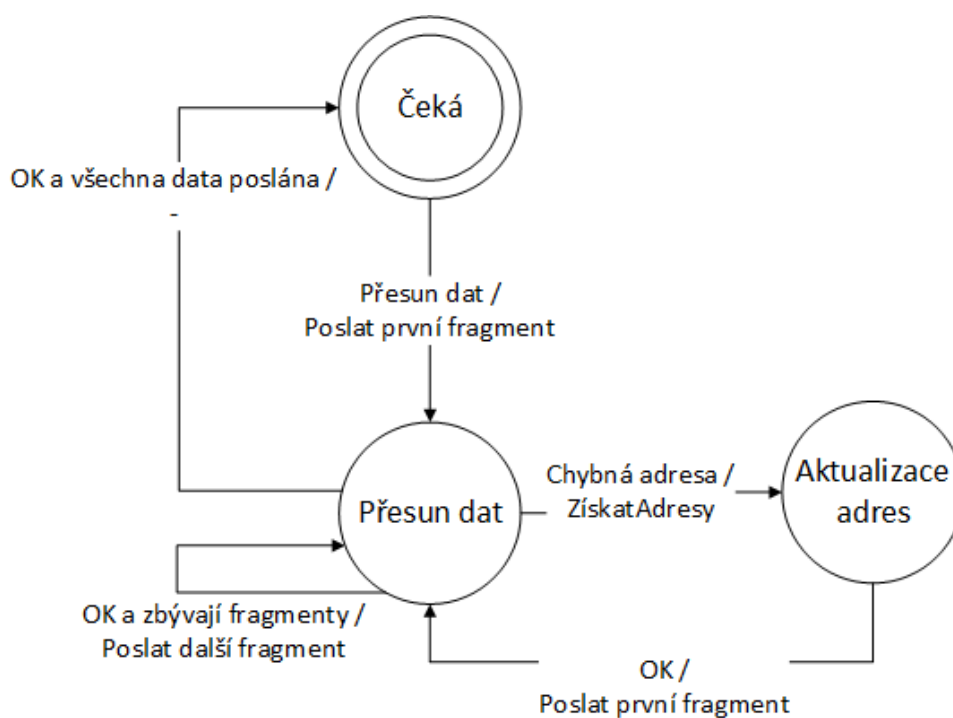
3.8.3 Operace: připojení k Wi-Fi síti

Operace připojení k Wi-Fi sítím se využívá k tomu, aby se Wi-Fi modul připojil k přístupovému bodu. V podstatě se jedná o jednocyklovou operaci, s tím rozdílem, že při úspěšném zpracování příkazu na straně Wi-Fi modulu se do paměti řídicí jednotky ukládají stavové informace o síti (návratový status Wi-Fi modulu, popř. přidělená IP adresa). Připojování k síti může trvat několik sekund. Timeout na připojování k síti je nastaven na 15 sekund. Operace je popsána diagramem na Obrázku 3.10.

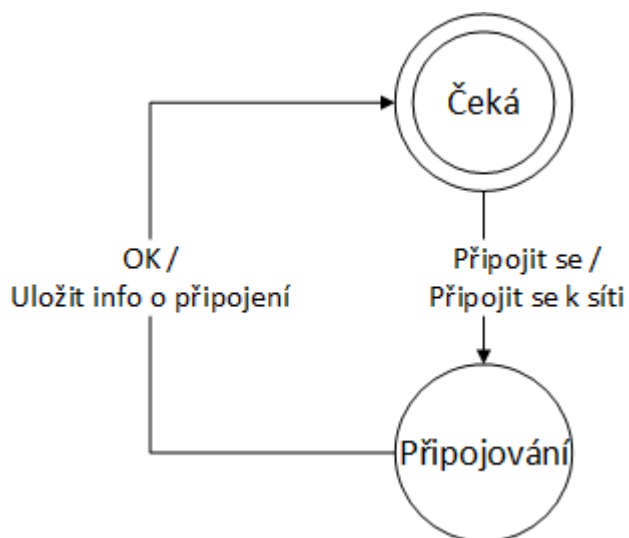
3.8.4 Operace: start přístupového bodu

Pomocí operace start přístupového bodu se Wi-Fi modul pustí v režimu přístupového bodu. Jedná se o velmi podobnou operaci jako připojení Wi-Fi síti, avšak ukládají se jiné stavové informace o síti, z tohoto důvodu není možné použít operaci připojení Wi-Fi modulu, ale je nutné mít vlastní stav pro tuto operaci. Narozdíl od připojování ke stanici je vytvoření přístupového bodu téměř okamžité. Start přístupového bodu je popsán diagramem na Obrázku 3.11.

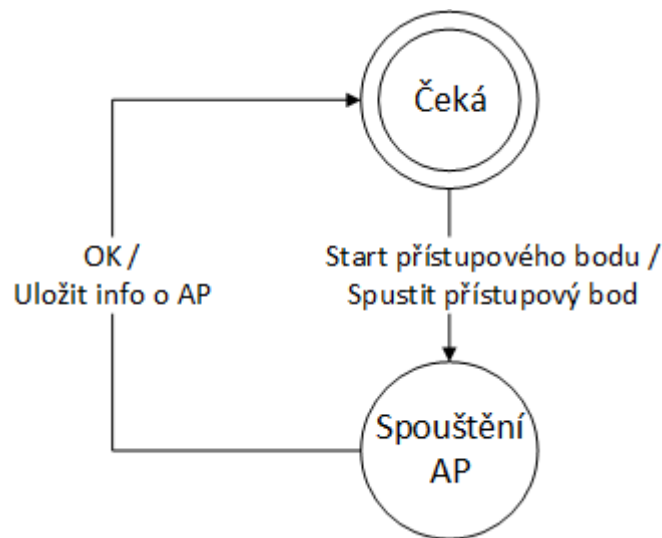
3. REALIZACE DATALOGGERU



Obrázek 3.9: Diagram operace aktualizace dat v mezipaměti, počátečním stavem je stav Čeká.



Obrázek 3.10: Diagram operace připojení k Wi-Fi síti, počátečním stavem je stav Čeká. OK znamená úspěšné potvrzení paketu command protokolu.



Obrázek 3.11: Diagram operace start přístupového bodu, počátečním stavem je stav Čeká.

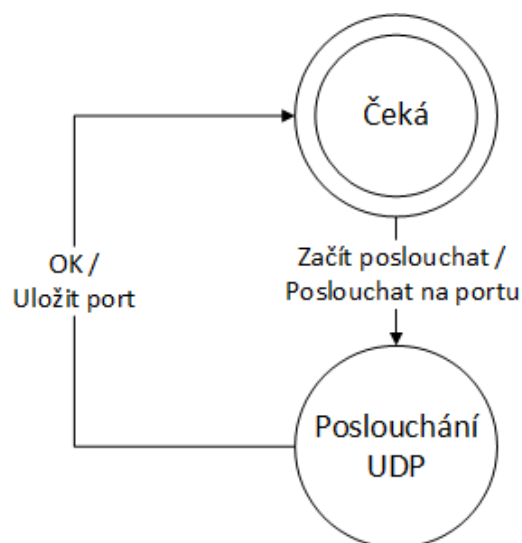
3.8.5 Operace: poslouchání na daném portu

Operace poslouchání na daném portu, aby Wi-Fi modul začal poslouchat UDP komunikaci na daném portu. Do paměti řídicí jednotky se ukládá informace o tom, na jakém portu Wi-Fi modul poslouchá. Operaci popisuje diagram na Obrázku 3.12.

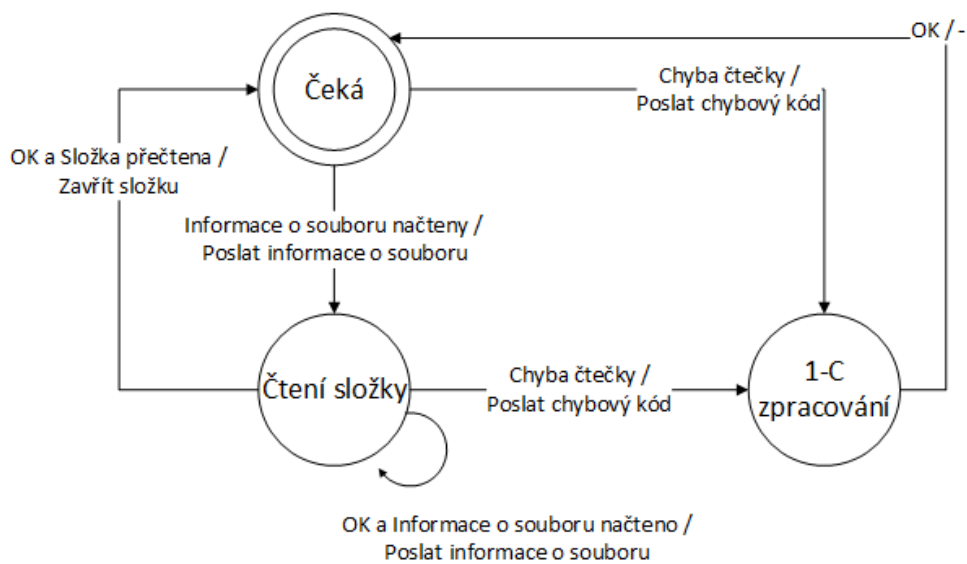
3.8.6 Operace: získávání seznamu souborů měření

Pomocí operace získávání seznamu souborů měření se získává seznam uložených souborů měření na microSD kartě. Každému souboru je přiřazeno jednoznačné ID počínaje 0, pomocí něho je možné daný soubor stáhnout, či smazat. ID se používá i jako identifikace UDP paketu (počítač si tedy může ověřit, že dorazily informace o všech souborech). Informace (jméno a ID) o každém souboru jsou posílány vlastním příkazem, který je následně na Wi-Fi modulu převeden na UDP paket a poslán počítači. Počet příkazů v operaci je tedy dán počtem uložených souborů na microSD kartě. Pokud se narazí na konec složky, tak se vyšle příkaz, který neobsahuje jméno souboru, který má ale vlastní ID. Tímto se dá najevo počítači, že složka byla úspěšně načtena a počítač by měl mít k dispozici seznam všech souborů na dané microSD kartě, tímto krokem operace končí. V případě chyby microSD karty se používá jednocyklová operace a chybový kód je poslán počítači. Operace je popsána diagramem na Obrázku 3.13.

3. REALIZACE DATALOGGERU



Obrázek 3.12: Diagram operace poslouchání na daném portu, počítačným stavem je stav Čeká.



Obrázek 3.13: Diagram operace získávání seznamu souborů měření, počítačným stavem je stav Čeká.

3.8.7 Operace: čtení části souboru

Operací čtení části souboru se kopíruje část souboru mezi řídicí jednotkou a počítačem. Vzhledem k tomu, že soubor může být velký (teoreticky až 4 GiB) a posílání celého souboru najednou nepřipadá v úvahu, tak se pomocí operace pošle vždy pouze část souboru (tisíce bajtů). Část souboru je jednoznačně daná ID souboru a indexem. Soubor je nutné číst sekvenčně, mezi čtením jednotlivých částí může být libovolná časová prodleva (jen se nesmí začít číst jiný soubor). Operace plně využívá příkazy, které umožňují využití protokolu se selektivním opakováním a posílání dávek UDP paketů. Operaci lze rozdělit na dvě větve, první větev je taková, když se čte dosud neotevřený soubor (index části souboru je 0), v tomto případě se nejprve inicializuje mezipaměť ve Wi-Fi modulu. Poté se řadič dostane do druhé větve, která je volána, pokud se čte již otevřený soubor. V této větvi se přečte daná část souboru (maximálně 7140 bajtů). Část je následně rozdělena na fragmenty o velikosti 255 bajtů. Fragmenty jsou poté přeposlány na Wi-Fi modul, kam se ukládají do mezipaměti. Po přesunu všech fragmentů je poslán příkaz, který všechny uložené fragmenty pošle na počítač, k tomu se používá posílání UDP paketu po dávkách. Ztrátu paketů řeší Wi-Fi modul a využívá se protokolu se selektivním opakováním. To znamená, že počítač nahlásí jaké pakety mu chybí a Wi-Fi modul je znovu odešle. Operace se beze změny využívá i když se část souboru posílá počítači připojený přes USB. Příkaz k vyslání dávky UDP paketů se v takovém případě bere jako potvrzovací signál, že veškeré fragmenty dané části byly poslány. Protokol se selektivním opakováním je blíže popsán v podkapitole 3.14.2 Selektivní opakování datagramu.

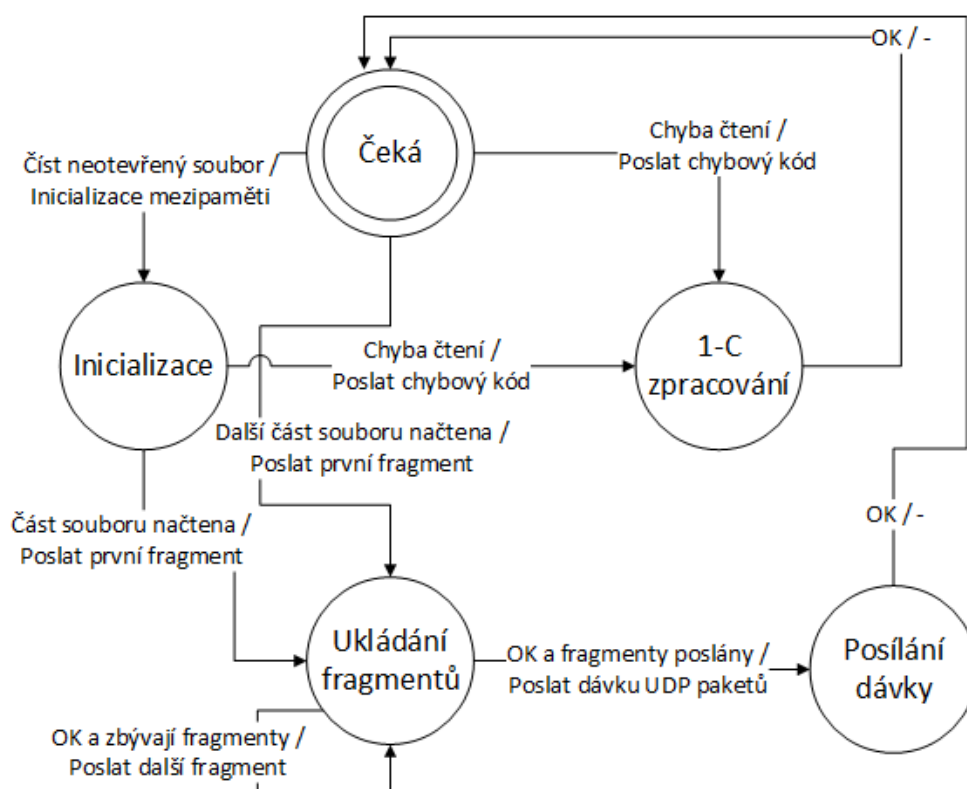
V případě neúspěchu (např. soubor s daným ID neexistuje, chyba microSD karty atd.) se používá jednocyklová operace a vyslání chybového kódu. Operace čtení části souboru popisuje diagram na Obrázku 3.14.

3.8.8 Operace: získávání informací o připojení

Operace, která se využívá k získání IP adresy Wi-Fi modulu a portu, na kterém právě poslouchá UDP komunikaci. Nejprve se Wi-Fi modulu pošle příkaz pro zjištění IP adresy. Tato IP adresa je následně uložena do paměti řídicí jednotky. Poté se vysílá příkaz, který získá port na kterém ESP-01 poslouchá. Tento port je taktéž uložen do paměti řídicí jednotky. Tímto krokem operace končí. Tuto operaci popisuje diagram na Obrázku 3.15.

3.8.9 Operace: skenování sítí

Operace skenování sítě slouží k tomu, aby Wi-Fi modul zjistil všechny dostupné bezdrátové sítě v okolí, poté se tyto sítě postupně posílají počítači připojený přes USB. Operaci může spustit pouze počítač připojený přes USB. Počítač připojený přes Wi-Fi taktéž může získat seznam okolních sítí, v takovém případě se ale vůbec nezatažuje řídicí jednotka a komunikace probíhá

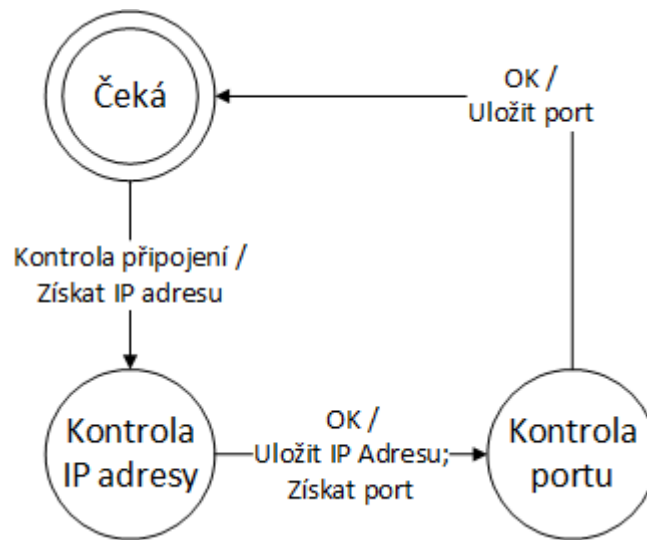


Obrázek 3.14: Diagram operace čtení části souboru, počátečním stavem je stav Čeká. OK znamená úspěšné potvrzení paketu command protokolu.

pouze mezi počítačem a Wi-Fi modulem. Operace skenování sítě začíná tím, že se Wi-Fi modulu pošle příkaz aby zjistil sítě v blízkosti. Po úspěšném vyřízení (součástí odpovědi na příkaz je i počet naskenovaných sítí) se postupně získávají informace o každé síti (SSID, typ šifrování, kanál, RSSI). Následně jsou tyto informace přeposílány do PC připojeného přes USB. Pokud nejsou žádné sítě k dispozici řadič se vrací do stavu *Čeká*. Operace je popsána diagramem na Obrázku 3.16.

3.9 Řadič měření

Řadič měření se stará o chod dataloggeru. Buzen je buď interními podněty, nebo externími podněty od počítače. Mezi interní podněty patří pravidelné události, jako např. aktualizace dat v mezipaměti ESP-01, kontrola připojení, nebo zápis dat na paměťové úložiště. Řadič měření je popsán diagramem na Obrázku 3.17. Jedná se o velmi zjednodušený diagram a stavy a přechody jsou zobecněné, jednotlivé stavy jsou blíže upřesněny v následujících podka-



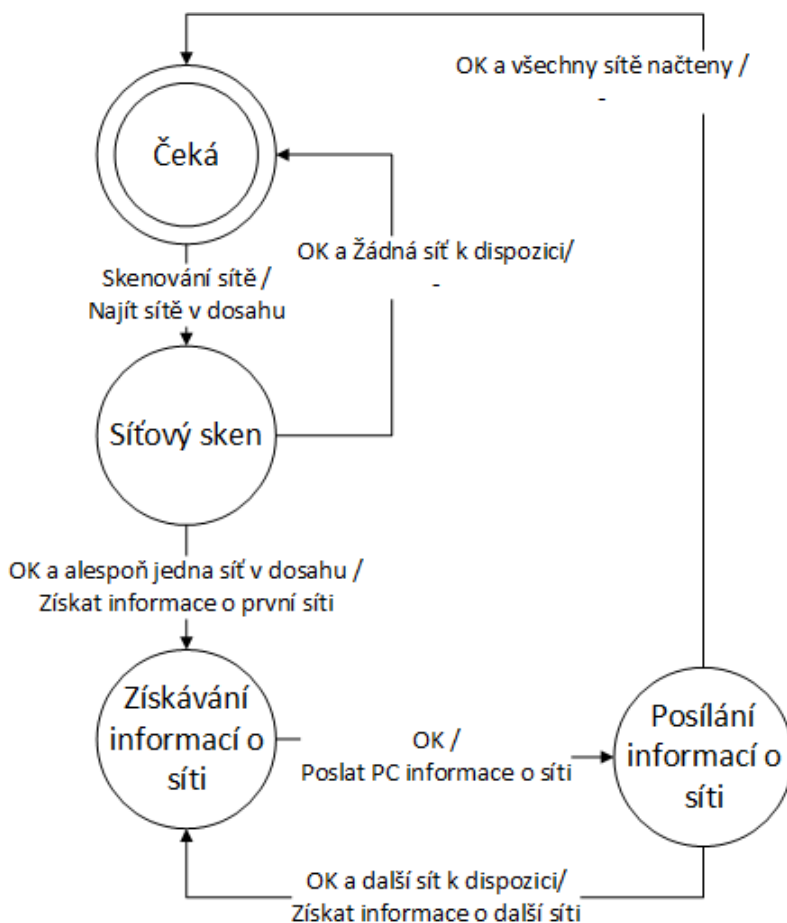
Obrázek 3.15: Diagram operace získávání informací o připojení, počátečním stavem je stav Čeká.

pitolách.

Po spuštění dataloggeru probíhá zahajovací procedura, která uvádí datalogger do chodu. V tuto chvíli není řadič měření ještě aktivní. Zahajovací procedura začíná nastavováním systémových hodin a inicializace periférií STM32. Tato část kódu je automaticky generovaná pomocí STM32CubeMX. Dalším krokem je konfigurace ostatních periférií během kterého také dochází k synchronizaci reálného času (viz 3.3.1 Synchronizace reálného času). Dále se načítá konfigurace dataloggeru a bezdrátových sítí z EEPROM. V tomto kroku dochází k připojování microSD karet a připojení filesystému. Pokud dojde po tomto kroku k fyzické manipulaci s pamětovou kartou, je nutné provést reset dataloggeru. Poté se povolí přerušení od obou sériových linek a sekundového časovače, zresetuje se řadič měření a příkazový řadič. Tímto krokem končí zahajovací procedura. Řízení dataloggeru přebírá řadič měření a datalogger je v plném provozu.

3.9.1 Počáteční stav (zastaveno)

Jedná se o výchozí stav řadiče. Ve stavu *zastaveno* může počítač komunikovat s dataloggerem. V tomto stavu se provádí pravidelné kontroly (interval je daný konfigurací dataloggeru), jestli Wi-Fi modul funguje dle očekávání přechodem do stavu *Kontrola sítě*. Je to z důvodu, že Wi-Fi modul se může od sítě kdykoliv odpojit (slabý signál, výpadek přístupového bodu). Pokud není Wi-Fi modul připojen k síti, tak se přechází do stavu *Připojování k síti*, ještě předtím ale proběhne hardwarový reset Wi-Fi modulu. Hardwarový reset Wi-Fi modulu se dělá tím způsobem, že řídicí jednotka na 10 milisekund nastaví pin

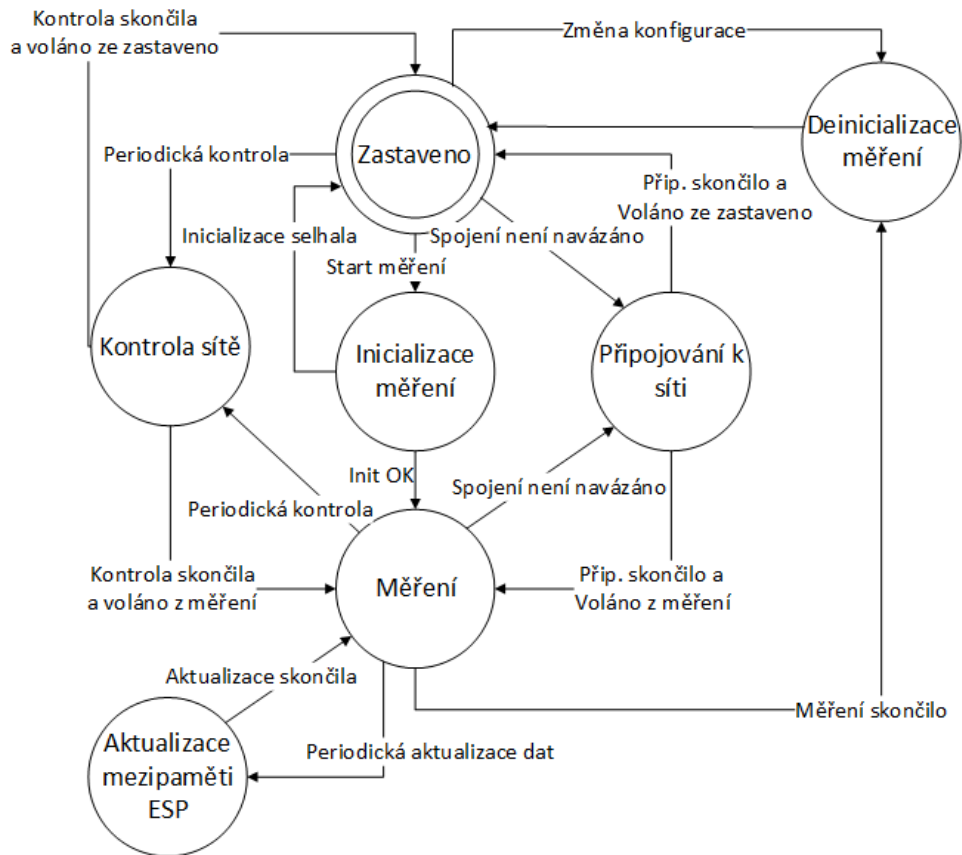


Obrázek 3.16: Diagram operace skenování sítí, počátečním stavem je stav Čeká.

CH_PD na ESP-01 na logickou 0, tímto se ESP-01 vypne. Poté řídicí jednotka nastaví pin CH_PD na logickou 1, počká 250 milisekund, aby ESP-01 stihlo naskočit a vymaže se sériový buffer. Ve stavu *Zastaveno* je také možné měnit konfiguraci dataloggeru, či sítí, danou konfiguraci uložit do EEPROM, či softwarově resetovat datalogger pomocí přechodu do stavu *Deinicializace měření* s danými parametry. Měření se spouští buď automaticky po připojení k síti, po určité době od startu dataloggeru, či pokud datalogger obdrží žádost na start měření. Dochází k přechodu do stavu *Inicializace měření*. Typ startu měření je daný konfigurací dataloggeru.

3.9.2 Stav: inicializace měření

Ve stavu *inicializace měření* se nastavuje měření. Aby bylo možné spustit měření, nesmí proběhnout konfigurace, která vyžaduje reset dataloggeru (např.



Obrázek 3.17: Diagram řadiče měření, počátečním stavem je *Zastaveno*.

manipulace s SD kartou). Datalogger je schopný změnu vyžadující reset detekovat a inicializace skončí neúspěchem a následným návratem do *Zastaveno*. Inicializace měření probíhá v pěti krocích:

1. Čtení konfigurace dataloggeru z RAM.
2. Kontrola systémového času. Pokud je problém se získáním systémového času, inicializace končí neúspěchem.
3. Vytváření souboru, do kterého se budou ukládat záznamy z měření. Během vytváření souboru se do něho přidá i hlavička. Na tento soubor se přidává zámek, který zabráňuje smazání souboru, dokud měření neskončí.
4. Příprava měřicí jednotky. Dochází k nulování hodnot všech kanálů. Dle módu měření se nastavuje spouštěcí hrana pro přerušení, v případě *po-*

3. REALIZACE DATALOGGERU

kračovaného měření spouštění časovače, které generuje vzorkovací frekvenci pro A/D převodník.

5. Příprava pomocných měřicích jednotek – nastavování spouštěcí hrany pro přerušování, resetování počítačové jednotky za minutu.

Po úspěšné inicializaci probíhá přechod do stavu *Měření*.

3.9.3 Stav: měření

Ve stavu *měření* datalogger zpracovává data z měřicí jednotky a pomocných měřicích jednotek. Je možné s dataloggerem komunikovat, konfigurovat datalogger a bezdrátové síť. Změna konfigurace neovlivňuje současné měření. Podobně jako ve stavu *Zastaveno* dochází k pravidelným kontrolám, jestli se datalogger neodpojil od sítě (přechod do stavu *Kontrola sítě*). Pokud datalogger není připojen k síti, tak se k ní připojí (postup je stejný jako ve stavu *Zastaveno*, včetně HW resetu ESP-01). Kromě toho ve stavu *měření* dochází k vypínání Wi-Fi modulu v závislosti na čase. Pravidelné vypínání v daných časech může uživatel nastavit v PC aplikaci. Uživatel např. ví že se o víkendech nebude k dataloggeru připojovat, je tedy možné nastavit automatické vypínání Wi-Fi modulu v sobotu a neděli. Vypínání Wi-Fi modulu značně snižuje spotřebu dataloggeru, což je i potvrzeno v rámci testování.

Ve stavu *měření* probíhá také v pravidelných intervalech (opět dané konfigurací dataloggeru) zápis naměřených dat na paměťové karty. Nejprve probíhá diagnostika dataloggeru, poté se aktuální data zapisují na paměťové karty. Diagnostika dataloggeru provádí test SD čteček, jednotek reálného času a senzoru BME280.

V pravidelných intervalech se provádí aktualizace dat na Wi-Fi modulu (přechod do stavu *Aktualizace mezipaměti ESP*). Měření se ukončuje pomocí žádosti ukončit měření. To probíhá tím způsobem, že se data uloží na paměťové karty a přechází se do stavu *deinicializace měření*.

3.9.4 Stav: deinicializace měření

Stav *deinicializace měření* slouží primárně k deinicializaci měření. Jedná se o opačný proces k inicializaci. To znamená, že se ze souboru se odstraňuje zámeček a zakazují se přerušování, které byly povoleny v souvislosti s měřením. Pomocí stavu *deinicializace měření* se překlápí konfigurace dataloggeru a sítě do EEPROM. V tomto stavu je možné vynutit smazání všech chybových příznaků, které indikují chyby jednotlivých součástí dataloggeru. Z tohoto stavu je také možné provést softwarový reset zařízení. Z těchto důvodů je možné do tohoto stavu přejít i ze stavu *Zastaveno*. Ze stavu *deinicializace měření* se vždy přechází do stavu *Zastaveno*.

3.9.5 Stav: připojování k síti

Stav *připojování k síti* se využívá, aby se Wi-Fi modul připojil k síti (či vytvořil přístupový bod) a začal poslouchat UDP komunikaci na portu daném konfigurací dataloggeru. V tomto stavu řadič měření úzce spolupracuje s příkazovým řadičem, protože připojení k síti a spuštění serveru na daném portu je sekvence příkazů od řídicí jednotky k Wi-Fi modulu. Postup je následující:

1. Načtení konfigurace sítě, pokud je nějaká k dispozici, jinak připojení končí s neúspěchem. Zamyká se příkazový řadič.
2. Vyslání příkazu, aby Wi-Fi modul používal danou statickou IP adresu. I pokud se nepoužívá statická IP adresa, tak se tento krok dělá. Jako IP adresa se v takovém případě nastavuje na 0.0.0.0 a Wi-Fi modul využívá IP adresu přidělenou DHCP, či výchozí adresu v případě přístupového bodu.
3. Čeká se na odpověď od Wi-Fi modulu. Poté se vysílá příkaz, aby se Wi-Fi modul připojil k síti (či vytvořil přístupový bod) s danými parametry. Využívá se operace *připojení k Wi-Fi síti*, nebo *start přístupového bodu*.
4. Čeká se na odpověď od Wi-Fi modulu (připojování k síti může trvat i několik sekund). Prakticky to znamená, že v tomto mezistavu řadič čeká dokud se příkazový řadič nedostane do stavu *Čeká*.
5. Kontrola jestli bylo připojení úspěšné, pokud ano pokračuje se k dalšímu bodu, pokud ne tak návrat na první bod.
6. Vyslání příkazu Wi-Fi modulu, aby přestal poslouchat na všech portech.
7. Čeká se na odpověď od Wi-Fi modulu.
8. Vyslání příkazu Wi-Fi modulu, aby začal poslouchat UDP komunikaci na daném portu. Využívá se operace *poslouchání na daném portu*.
9. Čeká se na odpověď od Wi-Fi modulu. Uvolňuje se zámek, nastavuje se čas další kontroly připojení a připojování končí s úspěchem.

Načítání konfigurace sítě má jasně dané pořadí. Je možné uložit konfiguraci až tří bezdrátových sítí, ke kterým se datalogger připojuje jako klient a konfiguraci přístupového bodu. Postupně se nejprve načítají první tři sítě, poté se zkusí vytvořit přístupový bod daný uživatelem. Pokud i to selže, vytváří se přístupový bod s výchozí konfigurací. Pokud se nepovede vytvořit ani přístupový bod s výchozí konfigurací, tak se jedná o selhání Wi-Fi modulu. Modul se proto vypíná (pin CH_PD se nastavuje na logickou 0), aby zbytečně nespotřeboval energii. Připojování končí u první úspěšně připojení sítě, tzn. pokud je úspěšné připojení k uživatelem definované síti 1, nezkouší se další síť. V každém případě se uvolňuje příkazový zámek a přechází do stavu, z kterého byl stav volán (*Zastaveno*, či *Měření*).

3.9.6 Stav: kontrola sítě

Stav *kontrola sítě* slouží k ověření zdali je datalogger připojený k síti a na jakém portu poslouchá. V tomto stavu řadič měření využívá operaci *získávání informací o připojení*. V případě odpojení od sítě má Wi-Fi modul IP adresu 0.0.0.0. Kontrola se provádí podle nejvyššího oktetu, který by měl být nullový v případě, že je datalogger korektně připojen k síti. Kromě IP adresy se kontroluje i port na kterém Wi-Fi modul poslouchá, jestli se shoduje s portem daný konfigurací dataloggeru. Pokud se port neshoduje skončí kontrola sítě s tím, že Wi-Fi modul není korektně připojený. Po kontrole portu se přechází zpátky do stavu odkud byl stav *Kontrola sítě* volán.

3.9.7 Stav: aktualizace mezipaměti ESP

Aby byl využit plný potenciál Wi-Fi modulu měl by mít co nejaktuálnější data z měření. Z tohoto důvodu existuje tento stav, řídicí jednotka pravidelně aktualizuje data na straně Wi-Fi modulu. Do stavu *aktualizace mezipaměti ESP* se řídicí jednotka dostává v pravidelných intervalech. Firmware dataloggeru s tím počítá, takže nedochází k žádné degradaci měření. Podobně jako u připojení k síti se jedná o sekvenci příkazů, které mají za úkol přenášet data mezi řídicí jednotkou a Wi-Fi modulem. V každém kroku se čeká na to, kdy příkazový řadič bude ve stavu *Čeká*. Postup je následující:

1. Zamknutí příkazového řadiče, aktualizace spektra ionizujícího záření na straně Wi-Fi modulu za použití operace *aktualizace dat v mezipaměti*.
2. Přečtení hodnot z BME280 a následné poslání hodnot Wi-Fi modulu za použití operace *aktualizace dat v mezipaměti*.
3. Poslání dat z pomocných přídavných měřicích jednotek za použití operace *aktualizace dat v mezipaměti*.
4. Načtení systémového času a odeslání času Wi-Fi modulu.
5. Odemčení příkazového řadiče, přechod do stavu *Měření*.

Pokud se nepoužívá vynucená aktualizace dat ze strany počítače, tak uložený čas v paměti na straně Wi-Fi modulu odpovídá času aktualizace dat.

3.10 Konfigurace čítačů/časovačů

V dataloggeru probíhají periodické děje. Ať už se jedná o kontroly připojení k síti, počítání CPM, vzorkování analogového signálu z měřicí části s frekvencí 44,1 KHz, či třeba mikrosekundové zpoždění potřebné pro správný chod BME280. Všechny pravidelné děje jsou časovány pomocí přerušení od čítače. Mikrokontroléry STM32 nabízí více nezávislých časovačů (počet časovačů se



Obrázek 3.18: Vztah mezi základními typy časovačů. Zdroj [28].

liší podle konkrétního mikrokontroléru). Časovače jsou rozděleny na tři typy: základní časovače (*Basic Timers*), časovače pro obecný účel (*General Purpose Timers*) a pokročilé časovače (*Advanced Timers*). Základní časovače neobsahují I/O kanály, není je tedy možné pro input capture, či generování PWM. Naproti tomu časovače pro obecný účel mají až 4 nezávislé kanály, mohou být 32bitové. Pokročilé časovače rozšiřují časovače pro obecný účel ještě o další funkcionality (např. generování komplementárního PWM signálu). Jako základní časovač lze využít časovač pro obecný účel, či pokročilý časovač, stejně tak lze použít jako časovač pro obecný účel pokročilý časovač. Hierarchii časovačů popisuje Obrázek 3.18. Pro účely dataloggeru jsou potřeba pouze základní 16bitové časovače, může tedy být použit jakýkoliv časovač, který mikrokontrolér nabízí, není třeba se zabývat o jaký typ časovače se jedná.

Pro správnou konfiguraci časovače je nezbytné znát frekvenci vybraného časovače. Frekvence je daná frekvencí APB sběrnice ke které je časovač připojen. Před výpočtem správné hodnoty děličky (prescaler) a periody je nutné zjistit ke které APB sběrnici je časovač připojen. V případě konfigurace hodinového signálu pro datalogger má APB1 i APB2 Timer clocks stejnou frekvenci 84 MHz, takže tento krok odpadá, ale v obecném případě to nemusí být pravda a časovače mohou mít různé frekvence. Když je známá frekvence je možné spočítat hodnoty, aby přerušení probíhalo ve správných intervalech. Pro výpočet se využívá následující vzoreček:

$$f_{\text{int}} = \frac{f_{\text{clock}}}{(\text{prescaler} + 1) * (\text{period} + 1)} \quad (3.1)$$

Prescaler je 16bitová hodnota určující děličku hodin. Dělička zpomaluje

hodiny a nemůže být 0, z tohoto důvodu se čísluje sice od 0, ale hodnota je vždy zvýšena o jedna. Rozsah tedy je 1 – 65536. Period určuje kolik hodinových taktů je potřeba aby došlo k přetečení (vyvolá se přerušení). Podobně jako u prescaler, perioda 0 nedává smysl, proto je tato hodnota vždy zvýšena o 1. Period může být buď 16bitová hodnota, či 32bitová hodnota v závislosti na konkrétním časovači. V následujících podkapitolách jsou blíže popsány jednotlivé časovače, jejich využití a nastavení konkrétních hodnot prescaler a period, aby docházelo ke správné frekvenci přerušení. Obsluha přerušení jakéhokoliv časovače v dataloggeru neobsahuje volání automaticky generované HAL funkce. Automaticky generované HAL funkce přináší zbytečnou zátěž navíc, což je u rychlých časovačů nežádoucí. Příznak přerušení se resetuje manuálně v obsluze přerušení. Speciální případ časovače je tzv. SysTick časovač. Je to interní časovač Cortex-M, který převážně slouží jako časovač pro RTOS (je-li použit). SysTick časovač má periodu 1 ms. SysTick v dataloggeru využívá knihovna FatFs která pracuje s I/O vrstvou. Informací o časovačích byly čerpány z následujících zdrojů [28], [61]. Informace o tom na jaké APB sběrnici je konkrétní časovač připojen je k nalezení v referenčním manuálu mikrokontroléru [36].

3.10.1 Sekundový časovač

Tento časovač má periodu 1 sekundu. Obsluha přerušení od sekundového časovače má za úkol inkrementovat o jedna interní hodiny a proměnnou, která reprezentuje počet sekund od startu dataloggeru. Tato proměnná se využívá jednak ke kontrole timeoutu příkazového řadiče, ale i např. toho, jestli se nemá zkontrolovat stav sítě, aktualizovat data v mezipaměti ESP-01, či zapsat data z měření na paměťovou kartu. Sekundový časovač běží za každých okolností. Pro sekundový časovač bude využit časovač pro obecný účel TIM4. Požadovaná frekvence přerušení je 1 Hz, je potřeba najít parametry prescaler a period aby byla dosažena požadovaná frekvence:

$$f_{\text{int}} = \frac{84000000 \text{ Hz}}{(8399 + 1) * (9999 + 1)} = 1 \text{ Hz} \quad (3.2)$$

Hodnotu Prescaler pro časovač TIM4 v STM32CubeMX je nutné nastavit na 8399 a Counter Period na 9999. Po spuštění daného časovače a povolení přerušení získáváme přerušení každou sekundu.

3.10.2 Mikrosekundový časovač

Časovač s periodou 1 mikrosekunda. Obsluha tohoto přerušení dekrementuje o jedna mikrosekundový čítač, pokud je jeho hodnota různá od nuly. Tento časovač běží pouze tehdy, když je potřeba mikrosekundové zpoždění. Mikrosekundové zpoždění se používá pouze pokud se pracuje se senzorem BME280. Jako mikrosekundový časovač se využívá časovač pro obecný účel TIM3. Frekvence

přerušení je 1000000 Hz, nyní je třeba zjistit hodnotu prescaler a counter:

$$f_{\text{int}} = \frac{84000000 \text{ Hz}}{(0 + 1) * (83 + 1)} = 1000000 \text{ Hz} \quad (3.3)$$

Pro přerušení s frekvencí 1 MHz je nutné nastavit prescaler pro časovač TIM3 na 0 a hodnotu Counter Period na 83. Daný časovač se spouští poté co se nastaví hodnota reprezentující zpoždění v mikrosekundách, vypíná se a zakazuje přerušení poté co odpočet skončí.

3.10.3 Časovač zajišťující počítání CPM

CPM se počítá tím způsobem, že se v pravidelných intervalech přidává do kruhového bufferu počet zaznamenaných impulzů, které v daném intervalu nastaly. O pravidelné intervaly se stará právě časovač, který vyvolává přerušení s periodou 3,75 sekund. Frekvence přerušení je tedy 4/15 Hz. Pro tento časovač byl využit časovač pro obecný účel TIM9, časovač je spuštěn pouze v době, když probíhá měření a je povoleno počítání CPM alespoň u jednoho přídavného měřicího zařízení. Opět je potřeba zjistit nastavení prescaler a counter, aby bylo dosaženo požadované frekvence:

$$f_{\text{int}} = \frac{84000000 \text{ Hz}}{(8399 + 1) * (37499 + 1)} = \frac{4}{15} \text{ Hz} \quad (3.4)$$

Podobně jako u předchozích časovačů stačí tyto hodnoty nastavit pro TIM9 v STM32CubeMX. Pokud se časovač spustí (a povolí přerušení), dostáváme přerušení s požadovanou frekvencí 4/15 Hz.

3.10.4 Časovač pro A/D převodník

Časovač se využívá pokud se využívá *pokračované měření signálu* analogové měřicí jednotky. Signál se vzorkuje s frekvencí 44100 Hz a průběžně se hledají vrcholy signálu z detektoru. Vzorkování s požadovanou frekvencí zajišťuje časovač pro obecný účel TIM10. V obsluze přerušení se spustí pouze konverze analogového signálu na digitální. Aby nedocházelo ke zbytečnému uváznutí v přerušení, tak se volá neblokující funkce konverze. Konverze běží na pozadí a v případě dokončení je vyvoláno přerušení od A/D převodníku. Časovač pro A/D převodník je spuštěn jen tehdy pokud probíhá měření a pokud je použito *pokračovaného měření*. Podobně jako u předchozích časovačů, i v případě časovače pro A/D převodníku je nutné vypočítat parametry prescaler a counter period:

$$f_{\text{int}} = \frac{84000000 \text{ Hz}}{(\text{prescaler} + 1) * (\text{period} + 1)} = 44100 \text{ Hz} \quad (3.5)$$

$$\text{period} = \frac{39979 - 21 * \text{prescaler}}{21(\text{prescaler} + 1)} \quad (3.6)$$

Rovnice nemá v oboru přirozených čísel řešení. Není možné při frekvenci hodin 84 MHz najít hodnotu prescaler a period, aby požadovaná frekvence přerušení byla 44100 Hz. Aby bylo možné frekvence 44100 Hz dosáhnout je třeba změnit nastavení hodin sběrnice. Tohle není příliš žádoucí, protože by to zpomalilo některé periferie a bylo by nutné přepočítávat časovače, které využívají stejnou sběrnici jako TIM10. Z tohoto důvodu je využita jiná vzorkovací frekvence zvuku a sice 48000 Hz. Požadovaná frekvence 48000 Hz se dosadí do rovnice:

$$f_{\text{int}} = \frac{84000000 \text{ Hz}}{(13 + 1) * (124 + 1)} = 48000 \text{ Hz} \quad (3.7)$$

Při vzorkovací frekvenci 48000 Hz již existuje kombinace prescaler (13) a period (124), která splňuje požadovanou rovnici. Nalezené hodnoty se nastaví u TIM10. Tímto je nastavování všech časovačů potřebných pro chod dataloggeru hotové.

3.10.5 Nastavování čítače pro jednotku RTC

Jednotka reálného času je také čítač. Při každé náběžné hraně hodin zvýší svojí vnitřní hodnotu o 1 (tím se zvýší čas o sekundu). Aby jednotka RTC fungovala dle očekávání je nutné jí dodat hodiny se správnou frekvencí – 1Hz. DS3231 modul již toto vyřešil za nás, ovšem u interní jednotky hodin reálného času (ty které jsou součástí mikrokontroléru STM32) je potřeba to nastavit v STM32CubeMX. Informace o správné konfiguraci jsou čerpány z referenčního manuálu mikrokontroléru [36]. Podobně jako u jiných časovačů, je nutné znát frekvenci hodin. RTC jednotka je připojena na LSE oscilátoru, který má frekvenci 32768 Hz.

Když je známá frekvence oscilátoru tak je potřeba dané hodiny zpomalit na frekvenci 1 Hz. Podobně jako u časovačů i u jednotky hodin reálného času je možné nastavit zpomalení hodin pomocí tzv. *Asynchronous Predivider value* (PREDIV_A) a *Synchronous Predivider value* (PREDIV_S). PREDIV_A je 7bitová hodnota, kvůli snížení spotřeby je výrobcem mikrokontroléru doporučeno, aby tato hodnota byla maximální (127), PREDIV_S je 15bitová hodnota. Využívá se následující vzoreček:

$$f_{\text{sec}} = \frac{f_{\text{RTCCLK}}}{(\text{PREDIV}_S + 1) * (\text{PREDIV}_A + 1)} \quad (3.8)$$

Vzhledem k tomu, že jeden parametr je daný doporučením (127), tak výpočet PREDIV_S je jednoduchý a vychází na 255. Nalezené hodnoty se zadávají v STM32CubeMX v položce RTC. Nyní je interní jednotka RTC správně nastavena a pokud se napájí knoflíkovou baterií drží čas i po ztrátě napájení dataloggeru.

3.11 Priority přerušeni

Datalogger využívá mnoho přerušeni od různých periférií. Některá přerušeni jsou pro měření důležitá (přerušeni od spouště, časovač pro výpočet CPM), některá méně (přerušeni od sériové linky). Z tohoto důvodu je vhodné nastavit každému přerušeni prioritu. Procesory Cortex M4 v souvislosti s prioritami přerušeni definují tzv. *Preemption priority* a *Sub Priority*. Preemption priority zajišťuje to, že pokud se během vykonávání přerušeni vyskytne jiné přerušeni s vyšší preemption priority, tak je aktuálně vykonávané přerušeni pozastaveno a začne se vykonávat přerušeni s vyšší preemption priority. Sub priority zajišťuje to, že pokud čeká na zpracování více přerušeni se stejnou preemption prioritou, tak se jako první vykonává přerušeni s vyšší sub priority. Jak u preemption priority tak i sub priority platí, že čím nižší číslo mají tím je vyšší priorita – přerušeni, které má preemption priority 0 a sub priority 0 má nejvyšší prioritu. Více informací o prioritách přerušeni je k dispozici v knize *Mastering STM32* [28].

Během určování priorit přerušeni se osvědčilo jako první sepsat všechna přerušeni, která mohou v dataloggeru nastat:

- Přerušeni od spouště měřící jednotky.
- Přerušeni od A/D převodníku.
- Přerušeni od spouště pomocné měřící jednotky 1 a 2.
- Přerušeni od přijímače UART1 a UART2.
- Přerušeni od časovačů 3,4,9,10.

Chybová přerušeni jako Hard fault, Memory management fault byly ze seznamu vynechány a použity výchozí hodnoty⁶. V případě každého přerušeni se pokládaly otázky typu: jak rychle musí být přerušeni obslouženo? Jak dlouho trvá zpracování přerušeni? Jak často přerušeni nastává? Jak moc je vážné, když se přerušeni zmešká? Pomocí těchto základních otázek bylo možné určit prioritu. Nejvyšší prioritu má přerušeni, které je nutné rychle obsloužit. Poté mají vyšší prioritu přerušeni, která jsou krátká, či nenastávají často. Nejnižší prioritu mají přerušeni, u kterého zmeškání neohroží chod měření, či funkci dataloggeru.

Přerušeni od spouště měřící jednotky musí být zpracováno co nejrychleji. Jakékoliv zpoždění může znamenat to, že se analogový signál začne zpracovávat pozdě a datalogger přijde o jeho maximální hodnotu. Pomalé vykonávání přerušeni od spouště může prodlužovat mrtvou dobu detektoru. Obsluha přerušeni od spouště je oproti ostatním přerušením dlouhá, protože se během ní spouští konverze A/D převodníku. Frekvence přerušeni není vyšší než 30000

⁶Nemaskovatelné přerušeni a hard fault přerušeni mají nejvyšší prioritu, která je neměnná.

– 40000 Hz. Při vyšších hodnotách už bude impuls příliš krátký a nelze ho zpracovat. I přes dlouhou obsluhu přerušení se jedná o nejdůležitější přerušení, co může v dataloggeru nastat, proto má nejvyšší prioritu.

Rychlost zpracování přerušení od A/D převodníku určuje to, za jak dlouho je A/D převodník připraven k další konverzi. Pokud není A/D převodník připraven a přijde od měřicí části další impuls, tak je impuls ignorován. Vzhledem k tomu, že samotná měřicí část nemá účinnost 100%, tak ztráta impulsu není takový problém, jako čtení signálu mimo jeho maximum. Přerušení od A/D převodníku má také delší obsluhu přerušení, četnost přerušení odpovídá četnosti konverze, maximální četnost je tedy 48000 Hz. Priorita tohoto přerušení je nižší.

Zmeškání přerušení od spouště pomocné měřicí jednotky znamená to, že se nezapočte impuls. Podobně jako u analogové měřicí jednotky, ani pomocné měřicí jednotky nemají 100% účinnost, zmeškání přerušení proto není takový problém. Obsluha přerušení je ale krátká, mikrokontrolér obsluhu přerušení zpracovává rychle. Četnost přerušení odpovídá aktivitě radionuklidu a závisí na účinnosti a mrtvé době pomocné měřicí jednotky. Frekvence přerušení se může pohybovat od jednotek Hz až po stovky kHz. Pokud by přerušení od pomocných měřicích jednotek mělo vysokou prioritu, hrozilo by v případě vysoké aktivity radionuklidu k vyhladovění ostatních přerušení.

Četnost přerušení od přijímače sériové linky je daná baudrate. Maximální baudrate, který datalogger využívá, je 1000000, což odpovídá maximální frekvenci přerušení 100 kHz (za sekundu je přijmuto maximálně 100000 bajtů). Zmeškání tohoto přerušení způsobí to, že datalogger ztratí bajt/bajty. Ztráta bajtu reálně znamená ztrátu celého paketu, avšak se ztrátou paketu systém počítá a umí na to reagovat. Obsluha přerušení od přijímače sériové linky je velmi krátká. Komunikace s PC by ale neměla nijak narušovat měření, proto je priorita těchto přerušení nastavena na nejnižší úroveň.

V dataloggeru jsou celkově 4 časovače, každý má svojí obsluhu přerušení, které je vyvoláno během přetečení čítače. Ze všech časovačů má nejvyšší prioritu sekundový časovač. Sekundový časovač udržuje přesný čas v případě, že obě jednotky RTC selžou, kromě toho četnost přerušení od sekundového časovače je nízká – pouze 1 Hz. Vyšší priorita nemůže způsobovat vyhladovění ostatních časovačů. Další z časovačů je mikrosekundový časovač, který má frekvenci přerušení 1 MHz. U tohoto časovače už hrozí, že způsobí vyhladovění ostatních přerušení. Ztráta přerušení není pro datalogger kritická, časovač se využívá pouze k mikrosekundovým zpožděním, kde se příliš neřeší přesnost. Z tohoto důvodu je priorita přerušení od mikrosekundového časovače nižší než u ostatních časovačů. Další z časovačů zajišťuje počítání CPM, který má nejnižší četnost přerušení - pouze 4/15 Hz. Zmeškání tohoto přerušení způsobí to, že se v následujících 60 sekundách špatně počítá hodnota CPM. Poslední z časovačů je časovač, který zajišťuje vzorkování signálu. Obsluha přerušení je dlouhá (spouští se konverze A/D převodníku), avšak zmeškání přerušení znamená to, že datalogger nevzorkuje se správnou frekvencí. Z tohoto důvodu

Tabulka 3.4: Priority přerušení

Přerušení:	Preemption priority	Sub priority
Spoušť měřící jednotky	0	0
Sekundový časovač	1	0
Časovač pro A/D převodník	1	1
Časovač pro výpočet CPM	1	1
A/D převodník	2	0
Mikrosekundový časovač	2	1
Pomocné měřící zařízení	2	1
Přijímače sériové linky	3	0

je nutné mít u tohoto přerušení vyšší prioritu.

Po rozvržení stupně priorit přerušení je nutné priority nastavit i softwarově. Priority se nastavují v STM32CubeMX v položce NVIC (*nested vector interrupt controller*). Jako první je potřeba nastavit priority group na 2 bity pro preemption a 2 bity na sub priority. Poté se postupně nastaví preemption priority a sub priority podle Tabulky 3.4.

3.12 Výpočet CRC32

V celém systému se často využívá výpočet CRC32, který se využívá k detekci chyb v datech. Kontrolní součet CRC32 používají všechny komunikační protokoly, ale používá se i k zabezpečení hlavičky souboru měření. Výpočet CRC32 je tedy nezbytný pro správný chod systému. STM32 nabízí jednotku na výpočet CRC32, výhodou dedikované jednotky je, že je mnohem rychlejší než v případě výpočtu softwarového. Nevýhodou je, že STM32F4xx nenabízí možnost zreflektovat (zrcadlově obrátit bity viz [62]) mezivýsledky (vstup, výstup). V celém systému se ale počítá s reflektovanými bity (dokonce i generující polynom je zreflektovaný), bylo by nutné data vždy softwarově reflektovat a po výpočtu reflektovat zpět. Časová úspora by byla minimální, složitost kódu by ale výrazně vzrostla. Informace o CRC32 na platformě STM32 je k nalezení v knize [28].

V celém systému (řídící jednotka, ESP-01, počítač) se využívá softwarový výpočet CRC32. Jako generující polynom se využívá 0x4C11DB7. Jako algoritmus se používá základní algoritmus na výpočet CRC32 [63]. Vyhledávací tabulka pro CRC32 je uložena v paměti jako konstanta, tabulka je zreflektovaná.

3.13 Chybové stavy dataloggeru

Každá část dataloggeru má vlastní chybový příznak, který určuje zdali v dané části nastala chyba či nikoliv. Chyby ale nemají stejnou váhu a např. selhání BME280 není tak závažné jako selhání jednotek hodin reálného času. Z tohoto důvodu jsou zavedeny tzv. chybové stavy dataloggeru, které mají převážně informační charakter pro uživatele. Výjimku tvoří stav Selhání, který zabraňuje spuštění měření. Stavů jsou odstupňované podle vážnosti:

1. OK
2. Varování
3. Chyba
4. Selhání

OK znamená, že v dataloggeru funguje vše dle očekávání a žádná chyba nebyla po dobu běhu detekována. *Varování* znamená, že měření může být nepřesné. *Varování* nastává, když selže BME280, či když uživatel dobrovolně nevyužívá duplexu (např. používá pouze jedné paměťové karty).

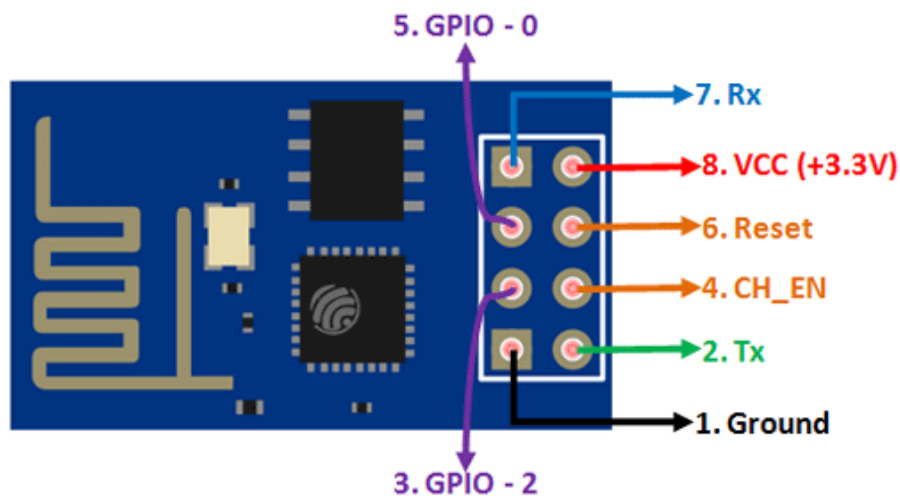
Chyba nastává, že některý z důležitých modulů selhal, avšak měření může být stále správné, tj. když selže jeden ze dvou jednotek RTC, Wi-Fi modul či paměti EEPROM. *Chyba* nastává i v době, kdy selžou obě paměťové karty, ale stále funguje Wi-Fi modul, či když selže jedna ze dvou SD čteček a Wi-Fi modul.

Nejzávažnější stav je *Selhání*. Tento stav indikuje selhání takové části dataloggeru, že měření bude špatné. Pokud je datalogger v chybovém stavu selhání není možné spustit nové měření. Měření, které probíhá, ale není ukončeno. *Selhání* je vyvoláno v případě, když selžou všechny RTC moduly (nelze číst systémový čas), selžou obě čtečky paměťových karet a Wi-Fi modul, nebo v případě, kdy není možné synchronizovat čas.

3.14 ESP-01 firmware

V rámci realizaci dataloggeru byl realizován i firmware pro ESP-01. Firmware je udělán tak, aby snižoval zatížení řídicí jednotky. V paměti Wi-Fi modulu se uchovávají aktuální hodnoty ze senzorů dataloggeru. Pokud je chce počítač číst, není třeba zatěžovat řídicí jednotku. ESP-01 firmware podporuje protokol se selektivním opakováním paketu, který usnadňuje přenos většího objemu dat. Protokol je blíže rozebrán v 3.14.2 Selektivní opakování datagramu.

Firmware funguje tak, že v nekonečné smyčce kontroluje příchozí UDP-Comm pakety od PC a pakety command protokolu od řídicí jednotky. UDP-Comm pakety se ve Wi-Fi modulu zpracovávají, kontroluje se jejich správnost a následně se vyřizují. Pokud se jedná o pakety, který vyžadují součinnost s



Obrázek 3.19: Rozložení pinů ESP-01 [64].

řídící jednotkou, tak se vytváří příslušný paket request protokolu, který je následně zaslán řídicí jednotce. Pakety command protokolu se zpracovávají neprodleně po obdržení.

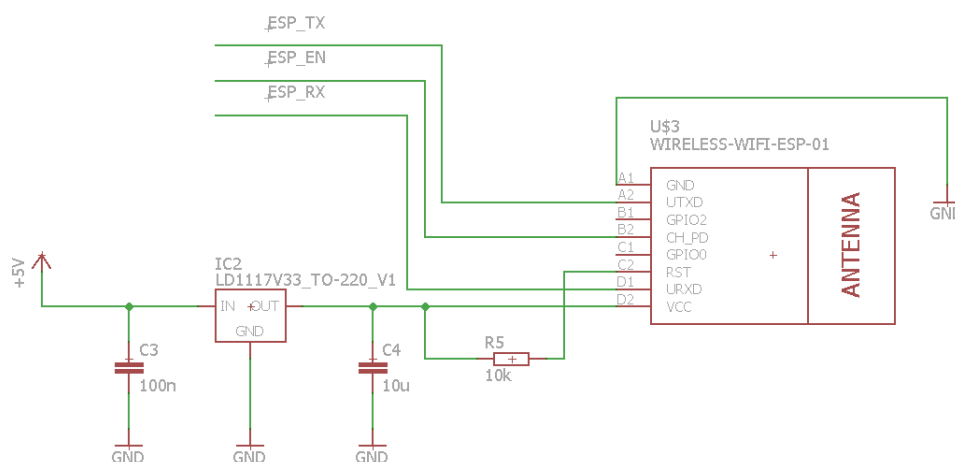
3.14.1 Zapojení ESP-01

Wi-Fi modul ESP-01 obsahuje celkově 8 pinů. Rozložení jednotlivých pinů popisuje Obrázek 3.19. Dva piny jsou napájecí - VCC a Ground. Na pin VCC je přivedeno napájecí napětí 3,3 V. Datalogger je napájen USB s napětím 5 V, k získání požadovaných 3,3 V je použit regulátor LD1117V33 (napájení je podrobně probíráno v kapitole 2.1.8 Napájení dataloggeru). Na pin Ground je přivedena zem.

Dalším pinem je reset. Pomocí pinu reset je možné zresetovat mikrokontrolér ESP8266, pokud je na něj přivedena logická 0. Pin reset řídicí jednotka nevyužívá, tudíž je natrvalo přes 10k resistor připojen na VCC (logickou 1). Hardwarový reset je možné vykonat i pomocí jiného pinu – CH_PD⁷. Pokud je na tento pin přivedena logická 0, tak je mikrokontrolér ESP8266 vypnut. Pokud je na něj přivedena logická 1 je zapnut. Řídící jednotka má možnost Wi-Fi modul vypnout právě pomocí pinu CH_PD, který je připojen na pin řídicí jednotky. Hardwarový reset ESP-01 se provádí tím způsobem, že na pin CH_PD je na několik desítek milisekund přivedena logická 0. Poté se na pin přivede logická 1, vyčká se 250 ms. Během této doby startuje Wi-Fi modul.

⁷V některých zdrojích je tento pin pojmenován jako CH_EN.

3. REALIZACE DATALOGGERU



Obrázek 3.20: Schéma dataloggeru - připojení ESP-01

Po vyčkáání je Wi-Fi modul plně funkční a lze s ním komunikovat přes sériovou linku.

Poslední 4 piny jsou piny pro obecný účel. Z toho dva piny jsou přijímač (RX), resp. vysílač (TX) sériové linky. Přijímač je připojen k vysílači sériové linky řídicí jednotky. Vysílač je připojen k přijímači sériové linky řídicí jednotky. Pomocí těchto dvou pinů ESP-01 komunikuje s řídicí jednotkou. Zbylé dva piny pro obecný účel nejsou využity a jsou nezapojeny. Zapojení ESP-01 popisuje Obrázek 3.20. Více informací o rozložení pinů je k dispozici ve volně dostupných zdrojích [64].

3.14.2 Selektivní opakování datagramu

Nejnáročnější komunikační operací mezi dataloggerem a počítačem připojený přes Wi-Fi je přenos souboru. Soubory jsou typicky velké a není přípustná ztráta jakékoliv části. Komunikace probíhá pomocí nespolehlivého UDP, který negarantuje doručení datagramů. Čtení souboru je taktéž časově náročnější a je výhodnější číst soubor po větších blocích (několik tisíc bajtů). Z tohoto důvodu je na straně Wi-Fi modulu realizován upravený protokol se selektivním opakováním [65].

V jednu chvíli se posílá pouze část souboru, která je velká maximálně 7140 bajtů. Část souboru je na straně řídicí jednotky rozdělena na 28 fragmentů po 255 bajtech. Rozdělení se provádí z důvodu, že command protokol nemůže přenést v jednom paketu 7140 bajtů dat.

Po rozdělení části na maximálně 28 fragmentů probíhá přenos jednotlivých fragmentů mezi řídicí jednotkou a Wi-Fi modulem. Wi-Fi modul si fragmenty ukládá do mezipaměti. Po uložení všech fragmentů do mezipaměti Wi-Fi mo-

dulu, vysílá řídicí jednotka příkaz, který všechny data z mezipaměti pošle PC pomocí UDP. Po tomto kroku již přenášení části souboru má plně pod kontrolou Wi-Fi modul a řídicí jednotka může pokračovat ve svojí činnosti.

Wi-Fi modul vysílá 28 UDPComm paketů. Každý paket je označený indexem, který odpovídá pořadí. Počítač pakety přijímá a podle indexů sestavuje pořadí. UDP negarantuje doručení datagramu, může tedy nastat situace, že některý z paketů není doručen. Pomocí indexů je však možné určit, které konkrétní pakety byly ztraceny a o chybějící fragmenty si počítač znova zažádá (selektivní opakování). Wi-Fi modul má chybějící fragmenty v mezipaměti, není nutné znova žádat řídicí jednotku o daný fragment souboru.

Mezipaměť pro části je implementována pomocí kruhového bufferu. Kromě toho má Wi-Fi modul v paměti indexy fragmentů, které má k dispozici a jednoznačný identifikátor souboru, ke kterému fragmenty patří. Podle toho může rozhodnout, jestli má daný fragment uložený v mezipaměti, či o něj musí žádat řídicí jednotku. Kruhový buffer podporuje pouze sekvenční přidávání fragmentů a při čtení prvního fragmentu souboru (index 0) je nutné tento kruhový buffer inicializovat a přiřadit mu identifikátor souboru. Jakým způsobem probíhá posílání části souboru na straně řídicí jednotky, je popsáno v podkapitole 3.8.7 Operace: čtení části souboru. Průběh posílání souborů pomocí selektivního opakování popisuje obrázek 3.21.

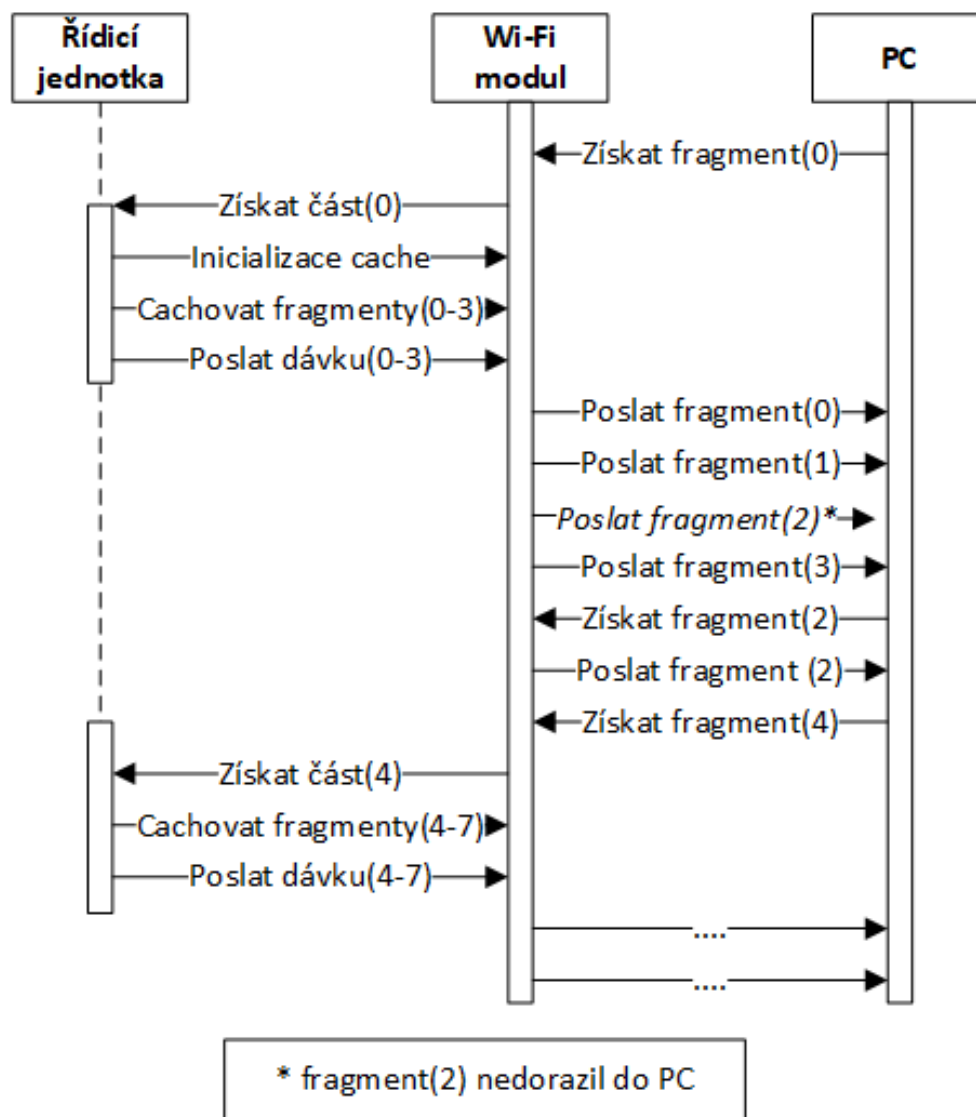
3.15 Výroba plošného spoje

Posledním krokem během realizace byl návrh a výroba plošného spoje. Výroba a doručení plošného spoje mohla v pandemické situaci trvat dlouho. Z tohoto důvodu byl plošný spoj zadán do výroby, co nejdříve to šlo⁸. Kromě toho se počítalo, že výroba nebude úspěšná na první pokus a bude potřeba chyby opravit a nechat vyrobit další plošný spoj.

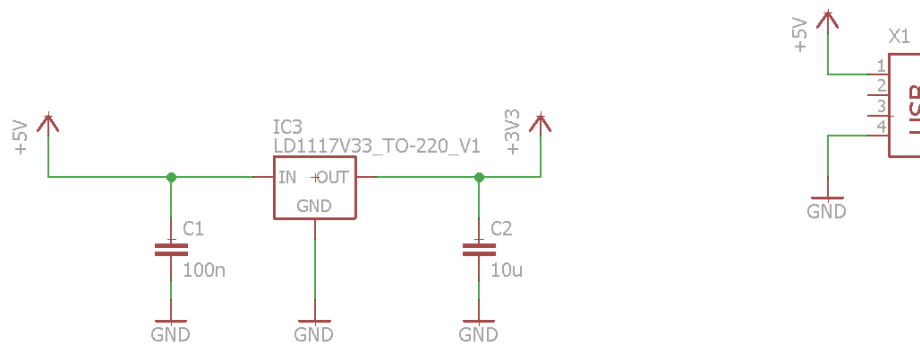
K návrhu plošného spoje byl využit program Eagle. Jako první bylo nutné datalogger zakreslit do schématu. Prvním problémem bylo, že nebyla k nalezení součástka WeAct Black Pill. Bylo ale využito znalosti, že WeAct Black Pill má stejné rozměry jako Blue Pill, ale jiné rozložení pinů. Byla proto stažena knihovna pro Blue Pill a přemapovaly se piny. Dalším problémem bylo, že některé typy součástek nebyly v programu EAGLE k dispozici. To bylo řešeno využitím knihoven třetích stran, které se do EAGLE importovaly. Výsledné schéma je na Obrázku 3.20, Obrázku 3.22 a Obrázku 3.23.

Když bylo zakresleno schéma, bylo možné rozvrhovat součástky a spoje na desce plošných spojů. K tomuto byl také využit program EAGLE. Nejprve se rozvrhovalo rozložení součástek. Během rozvrhování součástek bylo potřeba brát zřetel na to, že součástky jako čtečka SD karet, napájecí USB-B a komunikační microUSB musí být na kraji plošného spoje, aby v případě zapouzdření dataloggeru byly přístupné uživateli. Po rozvržení součástek se

⁸V době, kdy bylo jisté rozvržení součástek na desce.



Obrázek 3.21: Průběh posílání souborů pomocí selektivního opakování. Fragment 2 se během přenosu ztratí, PC o něj znovu požádá, ale zbylé fragmenty má už v paměti, takže po úspěšném obdržení žádá o fragment 4. Komunikace je zjednodušená.



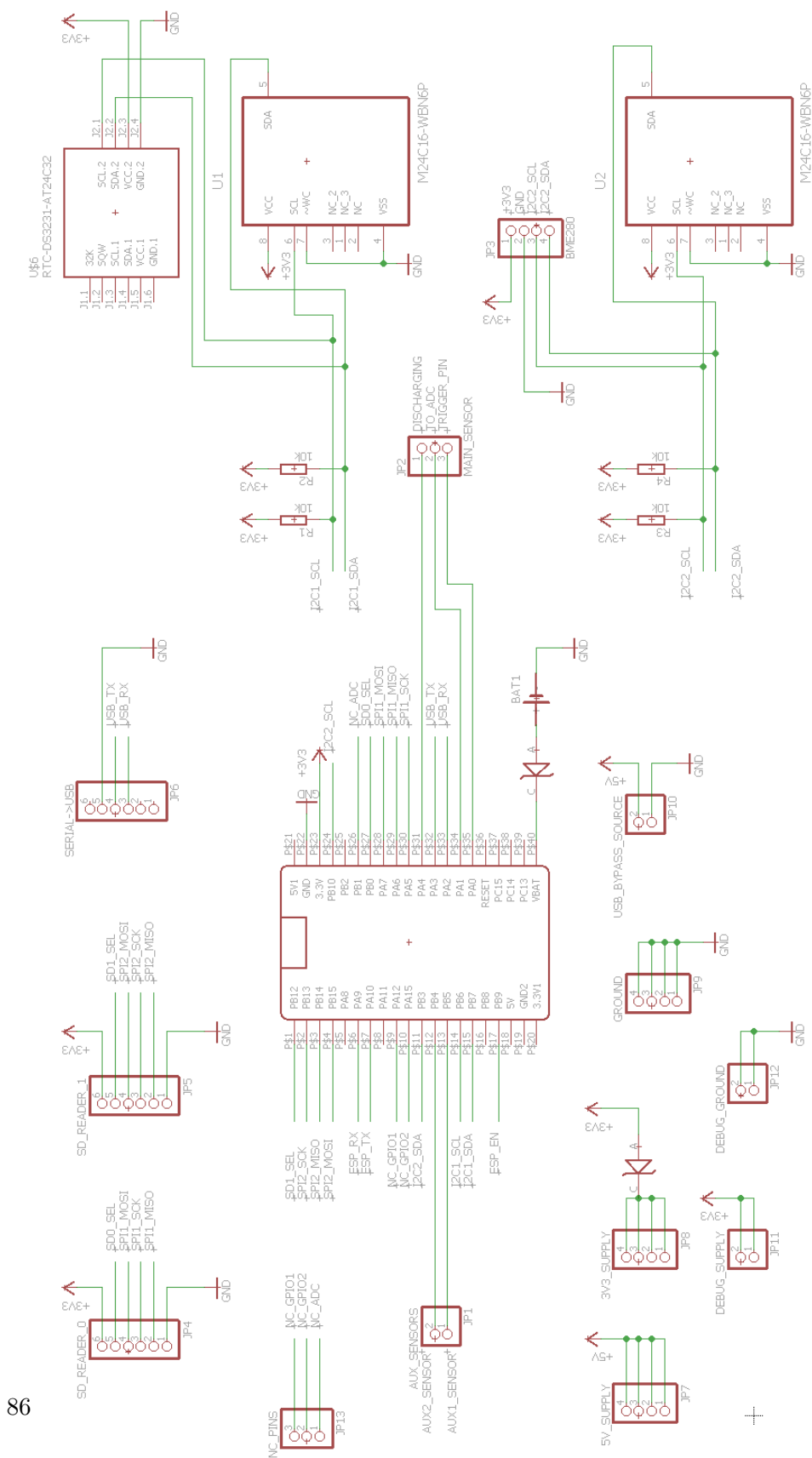
Obrázek 3.22: Schéma dataloggeru - napájení

rozvrhovaly cesty. Rozvrhování cest probíhalo nejdříve tím způsobem, že se nejdříve rozvedly napájecí cesty (5 V, 3,3 V a GND). Poté cesty pro ostatní součástky. Během rozvrhování cest se využíval i autorouter, ale pouze jako inspirace a vždy se automaticky vygenerované cesty překreslovaly, aby cesty byly optimálnější.

Celkově byly navrženy a vyrobeny dva typy desky plošných spojů. První měl obdélníkový tvar s rozměry cca. 12 x 8 cm. Deska obdelníkového tvaru byla navrhována v době onemocnění COVIDem-19, kvůli zdravotnímu stavu bylo uděláno mnoho chyb. První chybou byla špatné mezery mezi jednotlivými cestami, které byly menší než 0,254 mm. Továrna, kde se deska plošných spojů vyráběla, ale vyžadovala, aby byly mezery větší než 0,254 mm jinak dochází ke slévání dvou cest do jedné. Dalším problémem, avšak ne fatálním, byla malá izolace cest a součástek. Malá izolace by komplikovala osazování součástek, protože při neopatrném pájení mohlo snadno dojít ke zkratu. Poslední a zásadní chybou bylo zadání desky do výroby se špatně nastavenými parametry. Plošný spoj byl dvouvrstvý, kvůli neopatrnosti byla ale objednána výroba pouze jedné vrstvy. Plošný spoj byl ale objednan v České republice, takže doručení bylo velmi rychlé a na chybu se přišlo brzy. Výsledný plošný spoj je na Obrázku 3.24

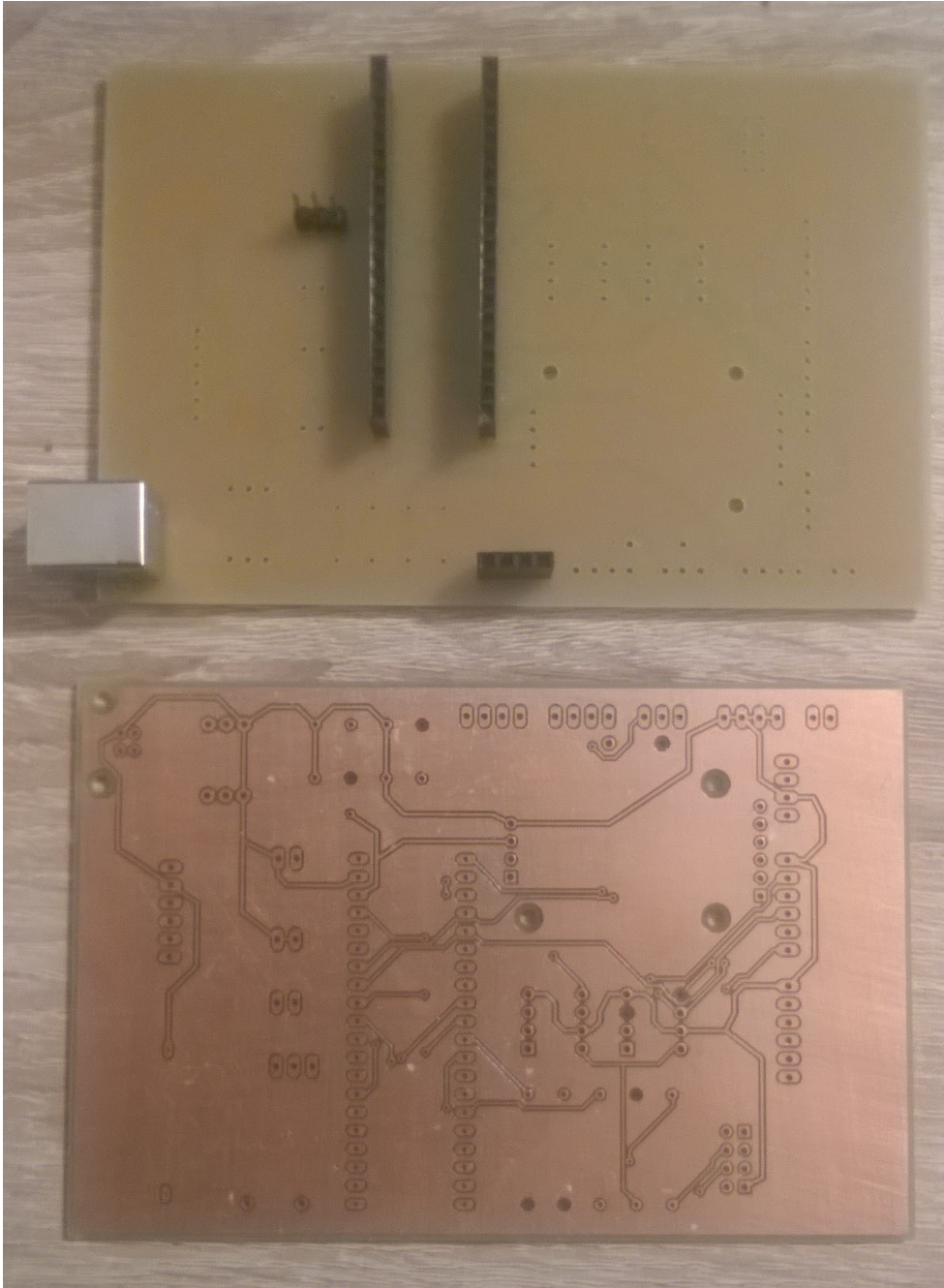
Po neúspěšné první desce plošných spojů byla navrhována druhá. Druhá deska plošných spojů má rozměry 99,1 x 95,3 mm. Použití takového rozměru bylo zvoleno z ekonomických důvodů, protože výroba 5 kusů PCB stála pouze 2 dolary. Pět desek plošných spojů s rozměry první desky by stály přes 20 dolarů. Druhá deska plošných spojů se i vyráběla v jiné továrně, která se nacházela v Číně. Během navrhování druhé desky jsem se poučil z chyb, které jsem udělal během návrhu první desky. Byla zvýšena velikost izolace. Dále bylo zajištěno, aby rozstup mezi cestami byl vyšší než 0,254 mm. Zvětšování rozstupů ale nebylo nutné, protože v továrně, kde byl vyrobený druhý plošný spoj, zvládali i menší mezery. Kromě toho byla zvýšena opatrnost bě-

3. REALIZACE DATALOGGERU



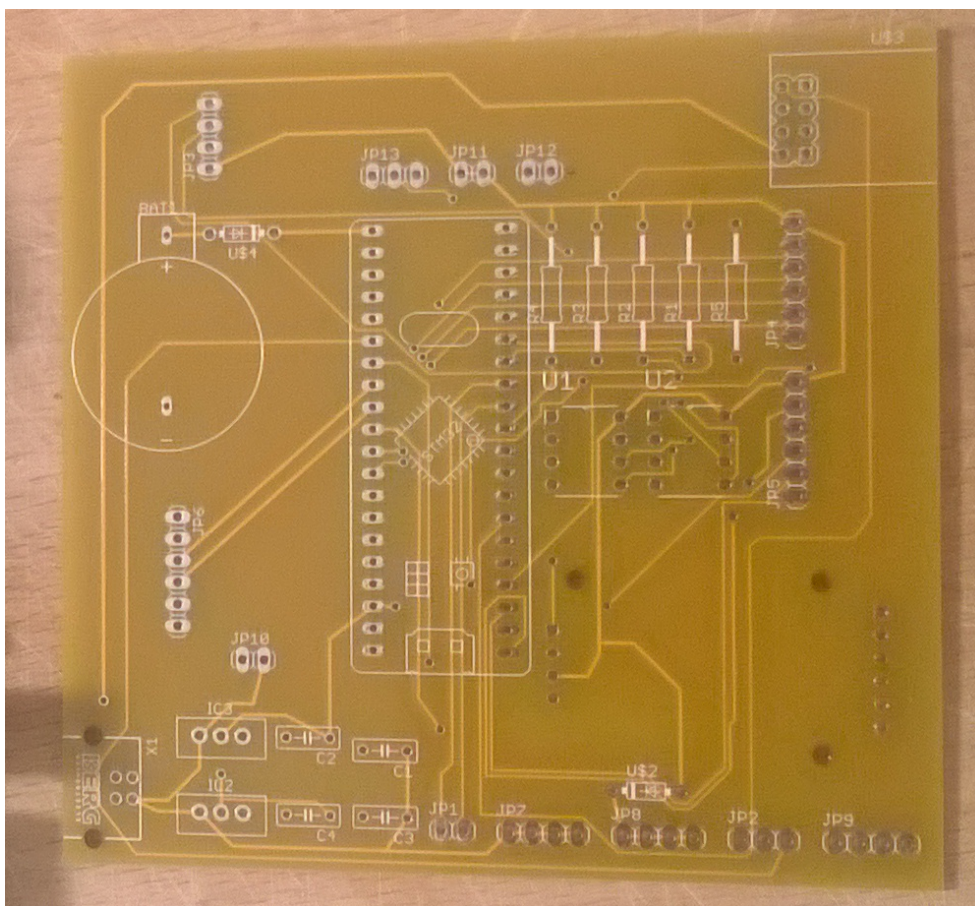
86

Obrázek 3.23: Schéma dataloggeru - hlavní část



Obrázek 3.24: První verze desky plošných spojů. Nahoře je vrchní vrstva, částečně osazena součástkami. Dole je spodní vrstva.

3. REALIZACE DATALOGGERU



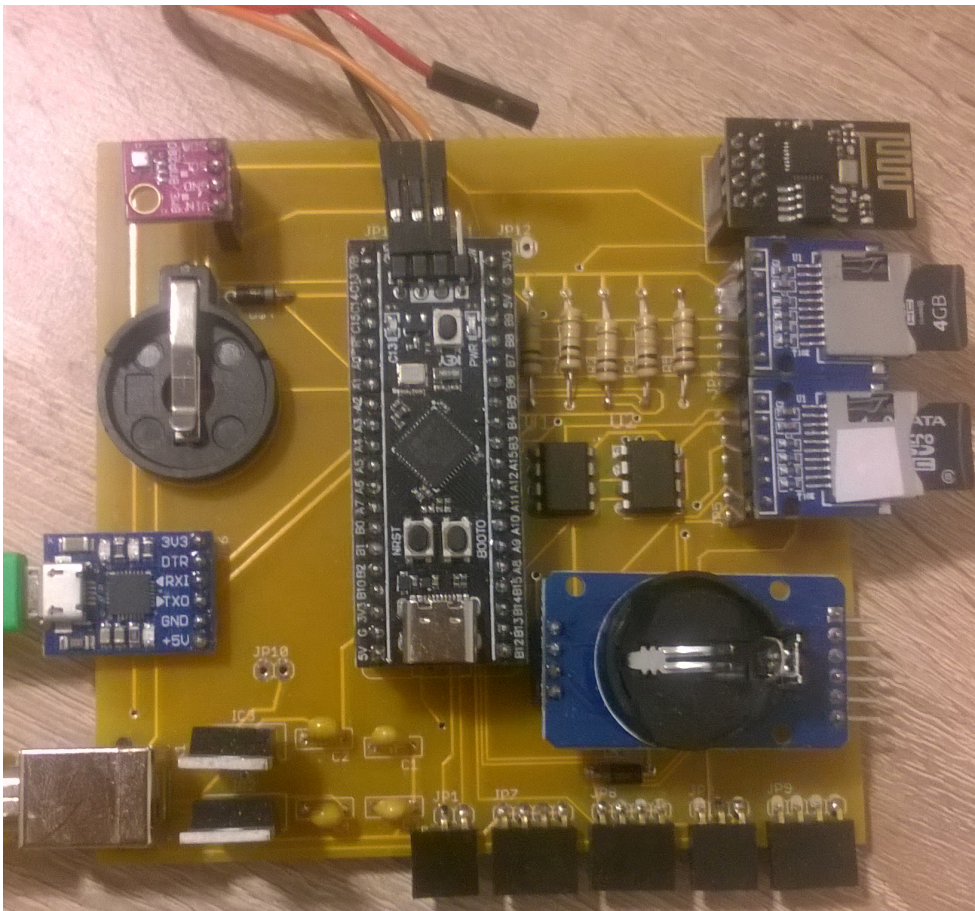
Obrázek 3.25: Druhá verze desky plošných spojů - neosazená.

hem zadávání do výroby, aby deska měla správný počet vrstev. Druhá deska plošných spojů už byla zcela funkční. Neosazená druhá deska plošných spojů je na Obrázku 3.25. Osazování desky plošných spojů probíhalo v domácích podmínkách pomocí levné pájecí stanice. Výsledek není zcela profesionální, avšak je plně funkční. Osazená deska plošných spojů je k vidění na Obrázku 3.25.

3.16 Cena dataloggeru

Po kompletní realizaci dataloggeru bylo možné vypočítat i výslednou cenu. Jako cena komponent je brána cena, za kterou byly skutečně koupeny. Do ceny komponent není započtena doprava. Cenu jednotlivých komponent popisuje Tabulka 3.5.

Výsledná cena dataloggeru je přibližně 420,- Kč. Ceny jednotlivých kom-



Obrázek 3.26: Druhá verze desky plošných spojů - osazená. K řídicí jednotce je připojen ST-Link a je připojen napájecí USB kabel a komunikační kabel.

ponent, ale nejsou optimální a ne vždy se součástky kupovaly za nejnižší možnou cenu. Existující řešení od firmy RHElectronics, které je blíže představeno v 1.4.3 Existující řešení pro zpracování signálu z detektoru stojí 1750,- Kč. Realizovaný datalogger je tedy přibližně 4x nižší než je nejlevnější nalezené komerční řešení. Kdyby se datalogger prodával komerčně, bylo by nutné započítat i délku realizace zařízení. Práce na dataloggeru včetně návrhu je odhadována přibližně na 400 hodin. Kromě toho se do času realizace musí započítat i realizace obslužné počítačové aplikace, která zabrala přibližně 120 hodin.

3. REALIZACE DATALOGGERU

Tabulka 3.5: Cena jednotlivých součástek dataloggeru

Součástka:	Počet	Cena za kus (Kč)	Výsledná cena (Kč)
Vývojová deska WeAct BlackPill	1	79	79
ESP-01	1	43,4	43,4
DS3231 modul	1	68	68
Čtečka paměťových karet	2	28	56
EEPROM	2	5,7	11,4
BME280	1	40,8	40,8
Držák na baterii	1	17	17
UART-USB převodník	1	27	27
Regulátor napětí	2	12,5	25
Plošný spoj	1	8,7	8,7
Pasivní součástky, ostatní	-	-	40
Celkem			416,3

Realizace počítačové aplikace

Obslužná počítačová aplikace se skládá z frontendu, který představují formuláře (okna) Windows Forms a backendu, který představují obslužné třídy pro komunikaci s dataloggerem a práci s daty. Backend je znovupoužitelný v případě, že by se předělávalo grafické rozhraní, či by se používala jiná frontendová technologie (WPF, UWP). Realizace počítačové aplikace probíhala tím způsobem, že byl nejprve implementován backend, který byl vytvářen nezávisle na frontendu. Po úspěšné realizaci backendu se realizoval frontend. Realizace frontendu spočívala hlavně v naklikávání uživatelského rozhraní v grafickém návrháři a následném propojení frontendu s backendem.

4.1 Komunikace s dataloggerem

Počítač komunikuje s dataloggerem buď bezdrátově pomocí UDP, nebo fyzicky přes USB. Bezdrátová komunikace využívá pakety UDPComm protokolu, fyzická komunikace pakety request protokolu a datalogger odpovídá paketem command protokolu. V počítačové aplikaci se pro komunikaci s dataloggerem využívá tzv. komunikační most. Komunikační most je rozhraní garantující určité funkce, které lze volat z frontendu. Funkce komunikačního mostu jsou následně namapovány na funkce pracující s příslušným protokolem a komunikačním rozhraním. Použití komunikačního mostu usnadňuje implementaci aplikace, protože se není nutné zabývat, jakým způsobem je uživatel připojen k dataloggeru, ale stačí obecné funkce komunikačního mostu.

Rozhraní představuje abstraktní třída DataloggerBridge z které dědí tři další třídy. První třídou je UDPBridge, která zajišťuje funkce pro komunikaci pomocí UDP. Další třídou je SerialBridge, která zajišťuje komunikaci pomocí USB přes virtuální sériovou linku. Poslední třídou je tzv. DummyBridge, která se využívá pouze po startu aplikace v době, kdy nebylo navázáno spojení. Pokud by v budoucnu datalogger podporoval další protokoly (např. TCP), tak by stačilo doplnit funkce pro práci s daným protokolem a nebyl by nutný

zásah do frontendu.

4.1.1 Komunikační most

Komunikační most zajišťuje abstraktní třída `DataloggerBridge`, která představuje rozhraní mezi frontendem a backendem aplikace. Komunikační most garantuje následující funkce:

- **Obecné:** Ukončit spojení s dataloggerem, vyčistit příchozí/odchozí buffery, Požádat o hlavní (master) přístup.
- **Práce s daty ze senzorů:** Získat aktuální data ze senzorů, vynutit aktualizace dat v mezipaměti.
- **Konfigurace dataloggeru:** Získat nastavení dataloggeru, získat stav dataloggeru, nastavit čas dataloggeru, změnit konfiguraci dataloggeru.
- **Konfigurace bezdrátových sítí:** Získat uložené konfigurace sítí, přidat konfiguraci bezdrátové sítě, změnit pořadí bezdrátové sítě, odstranit konfiguraci bezdrátové sítě, skenovat bezdrátové sítě v dosahu.
- **Práce se soubory:** Získat seznam souborů, stáhnout soubor, odstranit soubor.
- **Správa měření:** Spustit měření, zastavit měření.

Frontend aplikace neřeší jakým způsobem je právě připojena k dataloggeru, ale pouze zavolá příslušnou funkci z rozhraní. Samotnou komunikaci s dataloggerem zprostředkovává příslušný most (`UDPBridge`, `SerialBridge`). Komunikační most může být ve třech stavech - odpojeno, připojeno a fatální chyba. Komunikace s dataloggerem je možná pouze tehdy, pokud je komunikační most ve stavu připojeno. Fatální chyba znamená, že spojení nebylo korektně navázáno. Pokud je spojení úspěšně navázáno (stav připojeno), tak komunikační most počítá počet chyb, které během spojení nastaly. Chyba nastává, když datalogger neodpovídá na požadavek. Počet chyb slouží pouze jako informace pro uživatele, která může uživatele informovat o případném problému v komunikaci. Občasná chyba je ale zcela přirozená a spočívá v tom, že datalogger může zahazovat požadavky, pokud je právě zaneprázdněn. Komunikační most si drží informaci o tom, jestli má uživatel hlavní přístup.

Komunikační most si také uchovává aktuální data z dataloggeru. Jedná se o spektrum ionizujícího záření, data z pomocných měřicích zařízení, teplotu, vlhkost, tlak a čas. Dále i aktuální konfiguraci dataloggeru, konfiguraci bezdrátových sítí, zdali probíhá měření, název souboru, do kterého se ukládá, či chybový stav dataloggeru. Pokud frontend aplikace chce některé z hodnot číst, tak je čte přímo z komunikačního mostu.

4.1.2 UDP most

UDP most zajišťuje bezdrátovou komunikaci s dataloggerem pomocí UDP-Comm protokolu. UDPComm protokol je postaven na nespolehlivém UDP, který negarantuje doručení datagramu. Nespolehlivost tohoto protokolu je řešena na straně UDP mostu tím způsobem, že pokud PC aplikace neobdrží očekávanou odpověď, či obdrží poškozený datagram, tak je žádost opakována. Aby nedošlo k uvážnutí, tak je nastaven maximální počet opakování. Pokud i po maximálním počtu opakování neskončí daná žádost úspěšně, zvyšuje se počet chyb mostu o jedna a žádost se přeskakuje.

UDP most se inicializuje čtyřmi parametry. První parametr udává lokální port, na kterém aplikace poslouchá UDP komunikaci. Daný port musí být volný, nesmí na něm poslouchat jiná aplikace UDP komunikaci, jinak končí inicializace neúspěchem. Dalšími parametry jsou IP adresa a port dataloggeru. Posledním parametrem je timeout v milisekundách, který určuje, po jaké době se opakuje odesílání datagramu. V první verzi programu byl timeout nastaven jako konstanta ve zdrojovém kódu. Během realizace se ale ukázalo, že je občas síť pomalejší a datagramy přichází s větším zpožděním, než za normálních okolností. Proto má uživatel možnost timeout nastavit podle parametrů sítě, kde je datalogger připojen. Správný výběr timeoutu má vliv na kvalitu spojení. Pokud je zvolen příliš krátký timeout, žádosti zbytečně končí s chybou, protože datagramy nestihnou dorazit včas. Pokud naopak dlouhý, komunikace je pomalejší, protože na ztrátu datagramu je reagováno později. Během inicializace probíhá spuštění UDP klienta na uživatelem definovaném portu. Poté se provádí test, jestli na IP adrese a portu skutečně poslouchá datalogger. To je uskutečněno tím způsobem, že se vyšle UDPComm paket, žádající Wi-Fi modul, aby poslal informace o tom, kdy byla naposledy provedena aktualizace dat (v podobě milisekund od startu Wi-Fi modulu). Pokud na uživatelem zadané IP adrese a portu skutečně poslouchá datalogger, Wi-Fi modul na paket odpoví a UDP most je ve stavu připojeno. Pokud aplikace odpověď na UDP-Comm paket nezíská, UDP most přejde do stavu fatální chyba.

Po úspěšné inicializaci mostu je možné s dataloggerem pomocí UDPComm protokolu komunikovat. Existují tři způsoby, jakým datalogger na UDPComm pakety odpovídá. První způsob je, že odpovídá jedním paketem, další způsob je, že odpovídá dávkou paketů – předem neurčené množství paketů, které následují po sobě v krátkých časových intervalech. Posledním způsobem je, že na daný UDPComm paket datalogger vůbec neodpovídá. Způsob odpovídání je závislý na typu UDPComm paketu.

Nejkomplikovanější na realizaci jsou ty typy UDPComm paketů, na které datalogger nijak neodpovídá. Nelze totiž přímo zjistit jestli paket vůbec dorazil, či jestli ho datalogger zpracoval. Tento typ paketů ale ve většině případů mění stav dataloggeru. Je tedy možné před odesláním příslušného, získat aktuální stav toho, co se má měnit. Poté vyslat paket, který nevrací odpověď. Chvilí vyčkat a znova přečíst stav toho, co se měnilo. Pokud paket dorazil

změna se projeví.

Naopak nejméně komplikovaný na realizaci jsou ty typy UDPComm paketů, který odpovídají přesně jedním paketem. Úspěšné doručení a vyřízení je dané odpovědí Wi-Fi modulu. Pokud most obdrží odpověď, kterou předpokládal, lze konstantovat, že komunikace proběhla úspěšně.

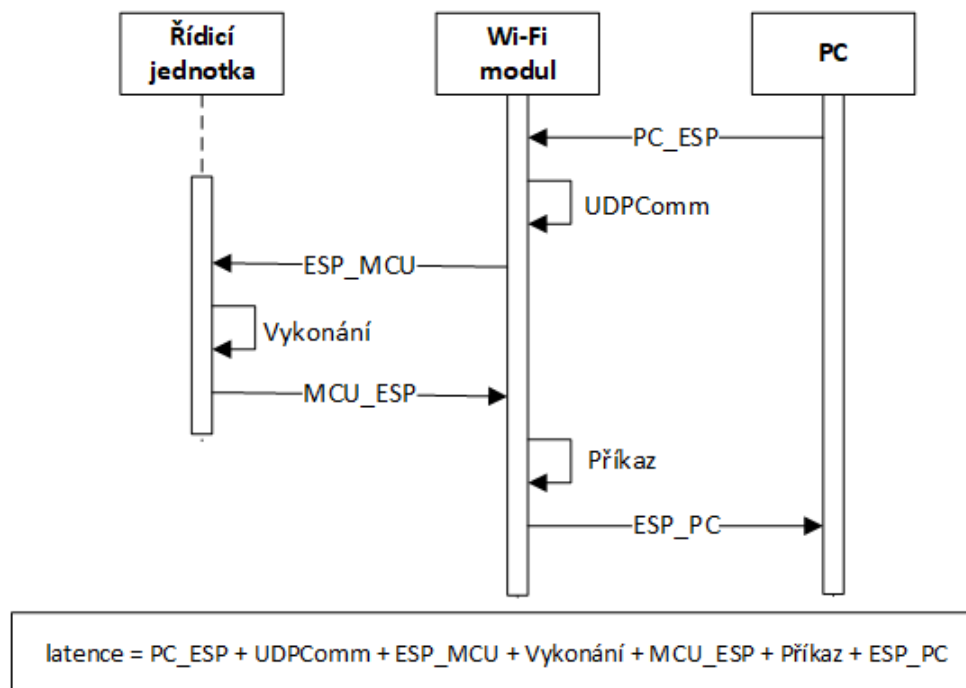
Poslední typ UDPComm paketu je ten, který odpovídá dávkou paketů. Zde je problém, že most předem neví, kolik paketů v dávce obdrží. Teoreticky to může být 0, ale i nekonečno⁹. Časový interval mezi jednotlivými pakety dávky také není konstantní a část paketů dávky se může ztratit. Dávka paketů se přijímá tím způsobem, že se ve smyčce volá funkce na přijetí jednoho paketu do doby, dokud most přijímá pakety. Pokud most nepřijme žádný paket v daném časovém intervalu, přijímání dávky končí. Většina dávkových přenosů má v sobě index paketu, aby bylo možné zjistit jestli dávka byla přijata celá. Výjimku tvoří sken sítí, kde ale ztráta paketu není závažná.

Během realizace jednotlivých funkcí pro most bylo nutné počítat i s celkovou latencí. V nejhorsím možném případě komunikace probíhá stylem PC – ESP01 – Řídicí jednotka – ESP01 – PC. Zpracování žádosti na straně řídicí jednotky může trvat i několik stovek milisekund (např. čtení z paměťové karty). Nelze vždy spoléhat na uživatelem definovaný timeout. Proto se u žádostí s vyšší latencí používá dočasné uspávání vlákna mostu po odeslání UDPComm paketu. Do latence je potřeba započítat i samotné zpracování paketu na každém uzlu, avšak zpracování je ve většině případů zanedbatelné (desítky mikrosekund). Výpočet latence, kde je nutné zapojovat i řídicí jednotku, je na Obrázku 4.1. Výpočet latence, kde probíhá komunikace pouze s Wi-Fi modulem (ESP-01) je na Obrázku 4.2.

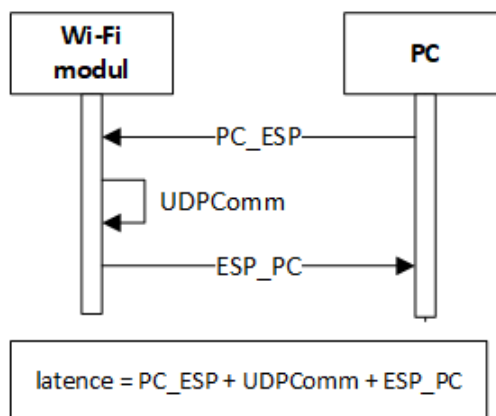
4.1.3 Sériový most

Sériový most zajišťuje fyzickou komunikaci pomocí USB. Datalogger zapojený přes USB počítač vidí jako virtuální sériovou linku. Není potřeba instalovat dodatečné ovladače, využívá se výchozí ovladač pro virtuální sériovou linku, který je součástí Windows 10. Aplikace pro komunikaci s datalogger využívá request protokol, datalogger odpovídá pakety command protokolu. Oproti UDP je komunikace přes sériovou linku spolehlivá a nedochází ke ztrátě paketů. Může ale dojít k zahození žádosti na straně dataloggeru (request protokol to umožňuje), či ztrátě některého z bajtů. Zahození žádosti je řešeno na straně PC aplikace tím způsobem, že se vyslání žádosti v případě nevyřízení opakuje. Aby nedošlo k uvážnutí, tak je nastaven maximální počet opakování, pokud není žádost vyřízena ani po limitu opakování zvyšuje se chyba mostu o jedna a pokračuje se dál. Ztráta bajtů je řešena pomocí CRC32. Pokud se některý z bajtů ztratí, tak kontrolní součet nevyjde správně a paket je zahozen.

⁹Dříve by ale aplikace spadla na nedostatek paměti.



Obrázek 4.1: Výpočet latence probíhající přes všechny uzly systému. UDPComm odpovídá době zpracování UDPComm paketu a Příklad délce zpracování příkazu.



Obrázek 4.2: Výpočet latence komunikace probíhající pouze přes Wi-Fi modul. UDPComm odpovídá době zpracování UDPComm paketu.

Sériový most se inicializuje pomocí dvou parametrů. Prvním parametrem je identifikátor sériového portu. Identifikátor je řetězec, který uživatel nastavuje v aplikaci. Druhým parametrem je baudrate. Baudrate je fixně daný nastavením dataloggeru, proto je i v aplikaci baudrate jako konstanta a uživatel jí nemůže nijak měnit. Vzhledem k tomu, že chování sériové linky je lépe predikovatelné než chování bezdrátové sítě, tak uživatel u tohoto mostu nemůže nastavit hodnotu timeoutu, ale je pevně daná. Během inicializace se otevírá sériový port. Pokud je port otevřen již jinou aplikací, nebo není dostupný, tak inicializace končí fatální chybou mostu. Pokud je otevření sériového portu úspěšné, tak se na straně dataloggeru vymaže příchozí buffer. To se dělá tím způsobem, že se dataloggeru pošle znak nového řádku. Poté se kontroluje zdali datalogger je funkční a komunikuje pomocí zaslání žádosti o status dataloggeru. Pokud datalogger funguje, na žádost odpoví.

Sériový most komunikuje s dataloggerem pomocí žádostí (paket request protokolu). Na některé typy žádostí datalogger vůbec neodpovídá, na některé odpovídá pomocí příkazu (paket command protokolu). Sériový most musí příkaz vyřídit a odpovědět odpovědí na příkaz. Pokud tak sériový most neučiní, dojde k dočasnému uváznutí příkazového řadiče na straně dataloggeru. Komunikaci vždy zahajuje sériový most pomocí žádosti. V případě, že se jedná o žádost, na kterou datalogger neodpovídá, je nutné použít stejný mechanismus ověřování doručení paketu jako v případě paketů UDPComm, na které Wi-Fi modul neodpovídá. To znamená, nejprve přečíst hodnotu, kterou se žádostí mění, poté poslat samotnou žádost a poté opět přečíst tuto hodnotu, jestli se změnila.

Žádosti, na které datalogger odpovídá příkazem, jsou sice inicializované sériovým mostem, avšak komunikaci řídí datalogger pomocí příkazového řadiče. Vzhledem k tomu, že komunikaci řídí příkazový řadič, tak je předem daný postup, jakým stylem bude každá komunikace probíhat. Pro jednodušší práci s příkazy je na straně počítačové aplikace implementován parser příkazů, který z každého příkazu vytáhne veškeré informace a zkontroluje konzistenci příkazu za využití kontrolního součtu CRC32. Parser vygeneruje i odpověď na příkaz, který se následně posílá nazpět dataloggeru.

4.1.4 Fronta požadavků

Po úspěšné implementaci komunikačních mostů bylo potřeba vyřešit, jakým stylem se tyto funkce budou volat z frontendu. V jednu dobu je možné vykonávat pouze jednu funkci komunikačního mostu, proto se jako řešení nabízí volat funkce z UI vláknem. Tento krok by skutečně zamezil tomu, že by se v jednu dobu volaly dvě funkce najednou. Uživatelské rozhraní by se po dobu vykonávání funkce pozastavilo a nebylo by možné spustit další funkci. Mělo by to, ale za důsledek to, že by program dočasně přestal odpovídat, což nevypadá dobře a uživatel má tendenci takový program násilně ukončit.

Z tohoto důvodu bylo rozhodnuto, že se požadované funkce vkládají do fronty, která je označena jako fronta požadavků. Funkce z fronty požadavků jsou postupně volány asynchronně. V kódu je zajištěno, že se v jednu chvíli může zpracovávat pouze jedna funkce. Asynchronní volání bylo prvotně řešeno tím způsobem, že se pro každou funkci vytvořilo vlastní vlákno. Vlákno se spustilo v době, kdy na něj přišla řada a bylo ukončeno poté co vykonalo danou funkci komunikačního mostu. Životnost takových vláken byla krátká (maximálně jednotky sekund) a byla vytvářena často. Vytváření vláken je ale pro operační systém náročná operace [66] a může mít za důsledek horší výkon aplikace.

NET framework nabízí lepší způsob, jak vytvářet vlákna s krátkou životností tzv. Thread pool. Thread pool zajišťuje, že proces má přidělený určitý počet vláken, které se recyklují po dobu běhu aplikace [67]. Odpadá režie vzniku a zániku vláken. Stejně jako u přístupu vyřizování požadavků pomocí samostatných vláken se i v případě thread pool využívá fronta. Do fronty se nepřidávají vlákna, ale akce (Action). Každá akce odpovídá požadované funkci komunikačního mostu. Pomocí akcí se spouští vlákna (označené jako tasky) v thread poolu. Z fronty se aplikace snaží odebírat akce v pravidelných intervalech. Aby bylo zajištěno, že se v jednu chvíli nespustí více funkcí komunikačního mostu najednou, tak se před samotným odebráním akce z fronty kontroluje, zda předchozí task již skončil. Pokud ano, odebere se z fronty vrchní akce a spustí se v thread poolu.

4.2 Správce souborů

Datalogger ukládá data ze senzorů do binárních souborů, jejichž formát je specifický pro účely tohoto dataloggeru. Aby uživatel s daty mohl pracovat, musí je aplikace zobrazit v čitelné podobě. Datalogger má k dispozici dvě nezávislé čtečky SD karet. Měření je možné ukládat na dvě nezávislé paměťové karty. Z tohoto důvodu aplikace podporuje porovnání dvou souborů. V případě, že soubory nejsou shodné, tak se je aplikace může pokusit opravit. Kvůli přenositelnosti dat, je možné data zkonvertovat do formátu csv.

4.2.1 Reprezentace dat měření v aplikaci

Pro jednodušší práci s daty z měření aplikace využívá třídu MeasurementData, která byla pro tyto účely realizována. Do objektu MeasurementData se načítají data ze souborů, či ho aplikace vytváří sama např. pokud průběžně zaznamenává data. Data z objektu MeasurementData lze zpětně uložit buď do binárního formátu měření, či do formátu csv. Aplikace umí pracovat pouze s třídou MeasurementData. Data není možné graficky zobrazit do doby, dokud nejsou data načtena v tomto objektu.

Třída MeasurementData obsahuje čas začátku a konce měření. Dále obsahuje nastavení měření, počet kanálů spektra, počet záznamů a samotný

seznam všech záznamů. Záznam je realizovaný třídou Record. Záznam odpovídá datům ze všech senzorů v době, kdy datalogger prováděl zápis do souboru. Kromě toho záznam obsahuje i čas záznamu a chybové příznaky. Záznam je jednoznačně určen časem, který slouží jako identifikátor. Pro uložení záznamů ve třídě se využívají dvě generické třídy .NET frameworku. První generickou třídou je SortedSet, který využívá jako parametr čas záznamu. V seznamu se udržují časy všech záznamů, které jsou seřazené. Záznamy se ukládají do druhé generické třídy Dictionary. Jako klíč se bere čas záznamu a jako hodnota záznam. Jak SortedSet, tak i Dictionary vyžadují unikátní klíče. Není možné přidat dva různé záznamy se stejným časem. Avšak takové chování je žádoucí, protože nedává smysl přidávat dva záznamy, které mají stejnou časovou značku, ale obsahují jiná data.

MeasurementData umožňuje spojování dvou různých objektů do jednoho. Taková funkcionalita je využita v případech, že se zápis dat ze senzorů provádí jak na paměťovou kartu, tak i do paměti počítače. Podmínkou spojení objektů je to, že musí mít totožné nastavení měření. Spojení probíhá tím způsobem, že se vezmou záznamy z jednoho objektu a postupně se přidávají do druhého objektu. Platí zde ale časová unikátnost záznamů. Záznamy, které obsahují stejný čas, jsou ze spojování vynechány.

4.2.2 Načítání dat ze souboru

Aby mohla aplikace s daty ze souboru pracovat, musí je nejprve načíst do objektu MeasurementData. Aplikace umí načítat soubory ve dvou formátech. Prvním formátem je binární formát, který používá pro ukládání dat datalogger. Binární formát obsahuje v prvních 64 bajtech hlavičku, která odpovídá nastavení měření (prvních 60 bajtů samotné nastavení, 4 bajty CRC32). Pomocí hlavičky je možné zjistit velikost každého záznamu. Jako první se načítá samotná hlavička. Poté se porovná, jestli CRC32 nastavení měření odpovídá CRC32, které je uloženo na konci hlavičky. Pokud kontrolní součet není totožný, používá se na nastavení měření maska, která odstraňuje chybu v nevyužitých bitech hlavičky. Poté se opět porovná CRC32. Pokud ani v tomto případě není porovnání úspěšné, soubor je považován za poškozený a načítání končí. Pokud je CRC32 hlavičky totožné, čte se název souboru. Datalogger ukládá do názvu souboru informaci o tom, kdy bylo započato měření. Pokud soubor neobsahuje ve jménu čas začátku měření, tak se používá jako začátek měření čas prvního záznamu. Poté se postupně načítají všechny záznamy do objektu MeasurementData.

Druhým formátem se kterým aplikace pracuje je csv. Na prvním řádku je explicitně definovaný separátor hodnot. Separátor je přečten a používá se v oddělování jednotlivých hodnot csv. Další řádek obsahuje informace o měření. Jako první hodnota je uložen čas počátku měření, další hodnoty na tomto řádku odpovídají nastavení měření. Pomocí těchto hodnot se inicializuje objekt MeasurementData. Třetí řádek souboru obsahuje záhlaví záznamů, která

slouží pouze pro přehlednost, pokud uživatel data otevře v programu třetí strany (např. Microsoft Excel). Další řádky odpovídají jednotlivým záznamům, které jsou postupně načteny do objektu MeasurementData.

4.2.3 Ukládání dat do souboru

Aplikace kromě načítání dat ze souboru umí data z objektu MeasurementData uložit do souboru buď ve formátu csv, či do binárního formátu, který využívá datalogger. Uložený soubor v binárním formátu má stejný tvar, jako by ho vytvářel datalogger. To znamená, že v prvních 64 bajtech se nachází hlavička, která obsahuje nastavení měření (60 bajtů) a kontrolní součet CRC32 nastavení (4 bajty). Po uložení hlavičky se postupně ukládají jednotlivé záznamy. Uživatel má možnost soubor uložit s doporučeným jménem, které obsahuje ve jméně čas začátku měření, či ho pojmenovat libovolně. Pokud ale uživatel nepoužije doporučené jméno, ztratí informaci o času začátku měření.

Soubor formátu csv na prvním řádku obsahuje explicitně daný separátor hodnot. Aplikace byla vyvíjena na české verzi Windows 10, proto se jako separátor využívá znak středníku. Na dalším řádku obsahuje čas začátku měření a nastavení měření. Třetí řádek tvoří záhlaví záznamů, které má pouze informační charakter. Další řádky odpovídají jednotlivým záznamům. Jak již bylo zmíněno, aplikace byla vyvíjena na české verzi Windows 10. Aplikace očekává, že soubor ve formátu csv používá český zápis. To znamená, že desetinná značka odpovídá znaku čárky (nikoliv znaku tečky), datum a čas je zapsán v „českém formátu“ DD.MM.RRRR hh:mm:ss.

V případě, že dojde k opakované konverzi formátu souboru (binární -> csv -> binární), tak hash výsledného binárního souboru nemusí být stejný jako hash souboru původního. Je to z důvodu, že desetinná čísla jsou ve formátu csv reprezentovaná omezeným počtem desetinných míst. Teplota se zaokrouhluje na dvě desetinná místa, tlak na čtyři desetinná místa a vlhkost na pět desetinných míst. Zaokrouhlení nemá vliv na výsledná data, ale mění binární hodnotu desetinných čísel. Aby měly dva soubory stejný hash, je potřeba, aby byly totožné. Zaokrouhlení ale způsobí, že některé bity mohou být odlišné.

4.2.4 Oprava souborů

Datalogger může data z měření ukládat na dvě nezávislé paměťové karty. Smyslem ukládání na více paměťových karet bylo to, aby aplikace mohla soubory porovnat a detekovat případnou chybu v datech. Chybou se v tomto případě rozumí nechtěné přepsání bitů/bajtů v souboru. Pokud je chyba detekována, aplikace se jí může pokusit opravit. Porovnání je možné pouze u souborů, které jsou uloženy v binárním formátu. Porovnání a oprava csv souborů není možná. Pro porovnání obsahu souborů se využívá hashovací funkce SHA-256. Pomocí SHA-256 se vypočte hash obou souborů, který se následně porovná. Porovnání souborů probíhá ve čtyřech úrovních. Každá úroveň na-

bízí jinou mírou nebezpečí v poškození výsledných dat způsobené opravou. Čím vyšší úroveň tím je nižší riziko poškození dat samotnou opravou. Aby si uživatel mohl vybrat, jakou úroveň oprav je schopný tolerovat, aplikace nabízí tzv. přísnost¹⁰ porovnání dat. Celkově jsou 4 přísnosti:

1. Nejvyšší
2. Vysoká
3. Střední
4. Nízká

Nejvyšší přísnost znamená to, že se hash souborů pouze porovná a neprovádí se žádné opravy. Pokud hashe nejsou stejné, porovnání končí neúspěchem a data se nenačtou. Pokud se hashe shodují, načítají se data a porovnání končí úspěchem.

Vysoká přísnost umožňuje opravovat soubor pomocí toho, že se delší soubor zkrátí na velikost kratšího souboru. Tento typ opravy je možný aplikovat pouze na ty případy, kdy jeden ze souborů je delší než ten druhý. Po zkrácení souboru se opět porovnají hashe, pokud nejsou stejné, porovnání končí neúspěchem. V případě shodnosti hashů se data načítají a porovnání končí úspěchem.

Pokud je nastavena střední přísnost, tak se kromě samotného zkracování souboru provádí i oprava hlavičky souboru. Oprava hlavičky je komplikovaný proces, protože může nastat mnoho scénářů, jakým způsobem může být hlavička poškozena. Chyba může být v užitečných bitech nastavení, v neužitečných (nepoužívaných) bitech nastavení nebo v CRC32. Chyby se mohou i kumulovat, je tedy možné, že jsou chybné bity jak v užitečných a neužitečných bitech nastavení, tak i v CRC32. Různé chyby mohou současně nastat na obou paměťových kartách.

Během opravy hlavičky se nejprve kontroluje, jestli v jednom ze dvou souborů je nepoškozená hlavička. To se kontroluje pomocí CRC32, které je uloženo na konci hlavičky. Pokud je detekována nepoškozená hlavička, použije se jako referenční hlavička a přepíše se jí poškozená hlavička druhého souboru. Taková oprava je vhodná, když je poškozená hlavička pouze jednoho souboru.

Pokud je tento typ opravy hlavičky neúspěšný, používá se maska, která vynuluje neužitečné bity. Poté se opět zkusí, jestli alespoň jeden ze souborů nemá CRC32 nastavení měření shodné s uloženým CRC32. Pokud ano, použije se jako referenční hlavička, která se přepíše i do druhého souboru a oprava hlavičky končí. Maskování dokáže opravit, pokud je poškozená hlavička v obou souborech, ale chyba se nalézá pouze v neužitečných bitech.

V případě neúspěšnosti maskování se využívá toho, že program porovná vymaskované hodnoty nastavení měření z obou souborů. Pokud je nastavení v

¹⁰V aplikaci se přísnost označuje slovem striktnost.

obou souborech shodné, vypočte se z tohot nastavení měření nové CRC32, kterým se přepíše chybné CRC32 v hlavičce. Tento typ opravy se hodí v případě, že je chyba v neužitečných bitech a v CRC32. Pokud ani tento typ opravy není úspěšný, tak oprava hlavičky končí neúspěchem. Neúspěch během opravy hlavičky znamená to, že soubor je nečitelný a porovnání končí chybou.

Po opravě hlavičky se naposledy porovnává hash obou souborů. Pokud není stejný a uživatel nastavil střední přísnost porovnávání souboru končí neúspěchem. Pokud je hash stejný, tak data se načtou a porovnání končí úspěchem. Oprava hlavičky se volá pouze v případě, že některá ze dvou hlaviček je poškozená.

Nízká úroveň přísnosti kromě zkracování souboru a opravy hlavičky porovnává i jednotlivé záznamy. Nutnou podmínkou, aby se provádělo porovnání záznamů je to, že musí být úspěšně přečtena hlavička. Porovnání záznamů probíhá tím způsobem, že se záznamy postupně načítají z obou souborů a každý záznam je porovnán se záznamem z druhého souboru. Pokud nejsou záznamy stejné, zahazují se. V nízké úrovni přísnosti se už hashe souborů neporovnávají a porovnání vždy končí úspěchem. Úspěch ale může znamenat, že jsou veškeré záznamy zahozeny. Průběh opravy souboru popisuje Obrázek 4.3.

Během porovnávání souborů dochází k přepisování souborů na disku. Proto je možné porovnání provádět buď na místě, či za použití pomocných souborů. Porovnání na místě znamená, že případné úpravy se dělají přímo do souborů, které se porovnávají. Porovnání za použití pomocných souborů se provádí tím způsobem, že se soubory, které se porovnávají nejprve zduplikují a úpravy se provádí na těchto kopiích. Po porovnání souborů se kopie souborů mažou.

4.3 Okna aplikace

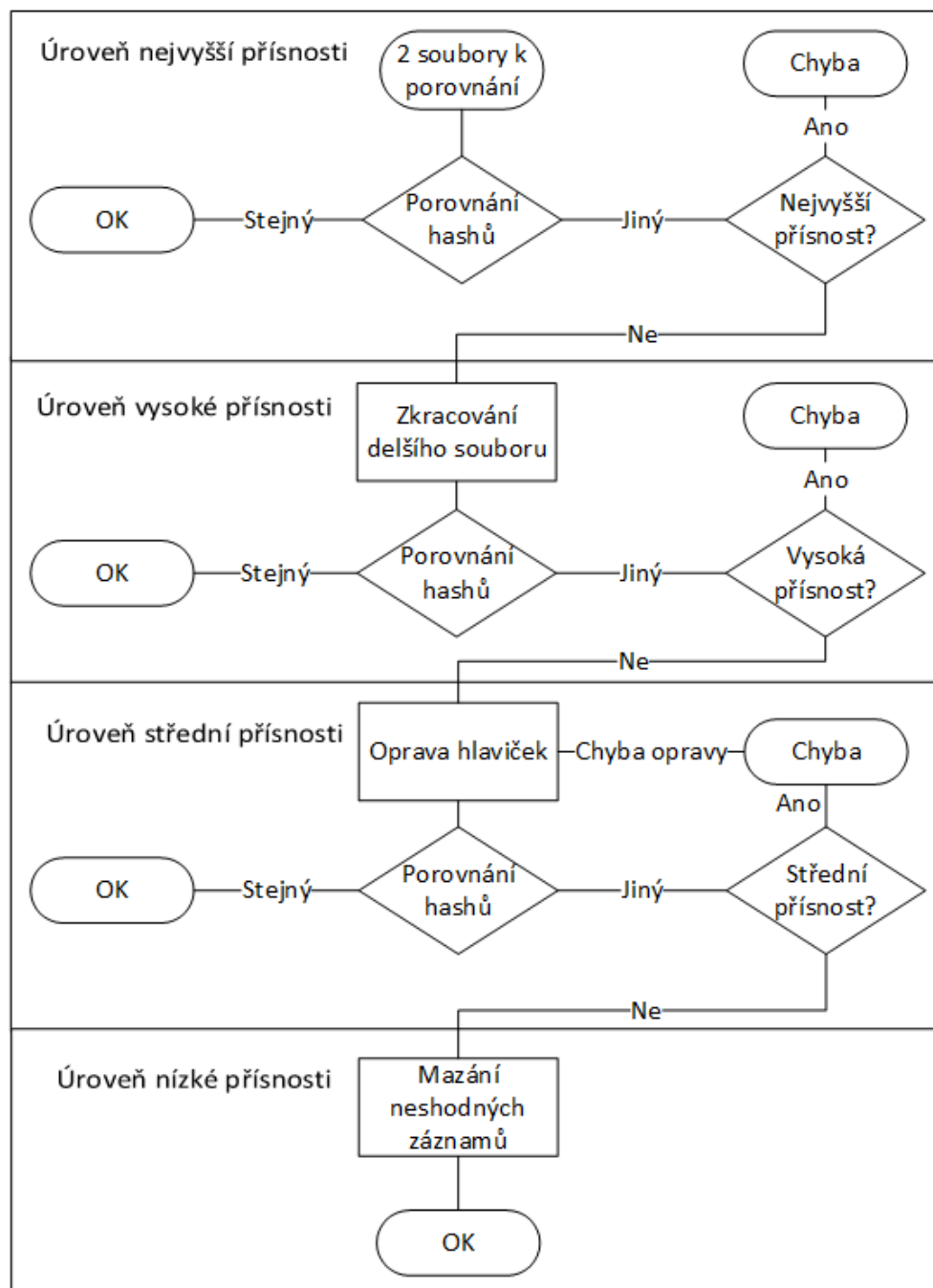
Okna aplikace zprostředkovávají uživateli grafické rozhraní. Každé okno je realizované jako formulář WinForms. Kromě rozložení jednotlivých ovládacích prvků, formuláře obsahují i obsluhy událostí (Event Handler). Obsluhy událostí jsou metody, které jsou vyvolané událostí v aplikaci jako např. kliknutím na tlačítko. Aplikace využívá celkově 4 různé fomuláře, každý z nich je blíže představen v dalších podkapitolách.

4.3.1 Hlavní okno

Hlavní okno je výchozím a zároveň hlavním oknem. Pokud se hlavní okno zavře, zavřou se i další okna, které aplikace vytvořila. V hlavním okně se může uživatel připojit k dataloggeru, sledovat aktuální data ze senzorů, spravovat soubory uložené na dataloggeru a konfigurovat datalogger. Ukázka hlavního okna je na Obrázku 4.4.

Uživatel v hlavním okně vybírá způsob připojení k dataloggeru a vyplňuje údaje potřebné pro připojení. V okně se zobrazují také informace o připojení

4. REALIZACE POČÍTAČOVÉ APLIKACE



Obrázek 4.3: Diagram opravy souborů.

4.3. Okna aplikace

Hlavní okno - Datalogger App

Správce dat | Uložit konfiguraci do EEPROM | Aktuální data | Soubory | Konfigurace sítě | Konfigurace dataloggeru | **Připojeno k dataloggeru**

Status dataloggeru
Měření probíhá: ANO
Stav dataloggeru: OK
Aktuální soubor: 210219_211715.DAT
Reset vyžadován: NE

Stav modulu
SD Čtečka 0: OK **Wi-Fi modul:** OK
SD Čtečka 1: OK **AUX1:** OFF
Interní RTC: OK **AUX2:** OK
Externí RTC: OK **Hlavní senzor:** OK
BME280: OK **EEPROM:** OK

Připojení
Hlavní přístup: ANO **Požádat o hlavní přístup**
Počet chyb: 22 **Vynutit přístup**
Požadavků ve frontě: 0 **Vyčistit frontu**

Konfigurace
 UDP USB (Virtual Com Port) **Odpojit se**

IP Adresa dataloggeru: 192.168.1.96
Port dataloggeru: 44666
Lokální port: 1
Timeout (ms): 250 **Připojit se**

Záznam dat
Nastavení:
Interval záznamu aktuálních dat (s): 60
Začít nové zaznamenávání
Zastavit zaznamenávání

Průběh záznamu:
Zaznamenávání probíhá: ANO
Start zaznamenávání: 19.02.2021 21:17:33
Počet záznamů: 109
Interval záznamů: 60
Otevřít záznam ve správci dat
Zahodit záznam

Spektrum:

Kanál	Počet pulzů
480	620193
481	397969
482	354191
483	273370
484	299420
485	234925
486	191329
487	131983
488	166253
489	107031
490	82686
491	60802
492	53267
493	38650

Otevřít aktuální data ve správci dat

Ostatní:
Čas dat: 19.02.2021 23:06:05
Teplota (°C): 22.86
Tlak (hPa): 994.072
Vlhkost (%): 31,105

AUX1 (celkový počet pulzů): 0
AUX1 CPM: 0
AUX2 (celkový počet pulzů): 3074
AUX2 CPM: 23

Nastavení aktualizace dat
Interval aktualizace dat (s): 60
Nový interval aktualizace: 60
Změnit interval
Aktualizovat data hned

Vynucení aktualizace dat v mezipaměti Wi-Fi modulu
 Aktualizovat spektrum **Vynutit aktualizaci**

Obrázek 4.4: Ukázka hlavního okna programu.

jako např. počet chyb mostu, počet požadavků ve frontě, nebo informace o stavu dataloggeru a jednotlivých modulů a periférií.

Hlavní okno využívá 4 záložky. Každá záložka zajišťuje určitou část funkcionalit. První záložkou je záložka *Aktuální data*, kde se zobrazují aktuální data ze všech senzorů. Spektrum se zobrazuje jako tabulka. Pokud si uživatel přeje aktuální spektrum zobrazit graficky, či aktuální hodnoty uložit do souboru, může tak učinit stiskem tlačítka *Otevřít aktuální data ve správci dat*. V záložce *Aktuální data* je možné nastavit i interval čtení dat z dataloggeru, či vynutit okamžité čtení dat. Uživatel může aktuální data v pravidelných časových intervalech zaznamenávat do paměti počítače a tím nahrazovat, či doplňovat paměťovou kartu. Aby bylo zaznamenávání dat na počítač účinné, je nutné, aby datalogger měl stabilní připojení k počítači.

Druhou záložkou je záložka *Soubory*. V záložce je možné spravovat soubory, které se nalézají v dataloggeru na paměťových kartách. Kromě mazání souborů měření je možné i soubory v záložce *Soubory* stahovat do počítače.

Třetí záložkou je *Konfigurace sítí*, kde probíhá správa sítí. V záložce je možné sítě přidat, odebrat, či změnit jejich pořadí. Také je zde možné provést skenování dostupných sítí.

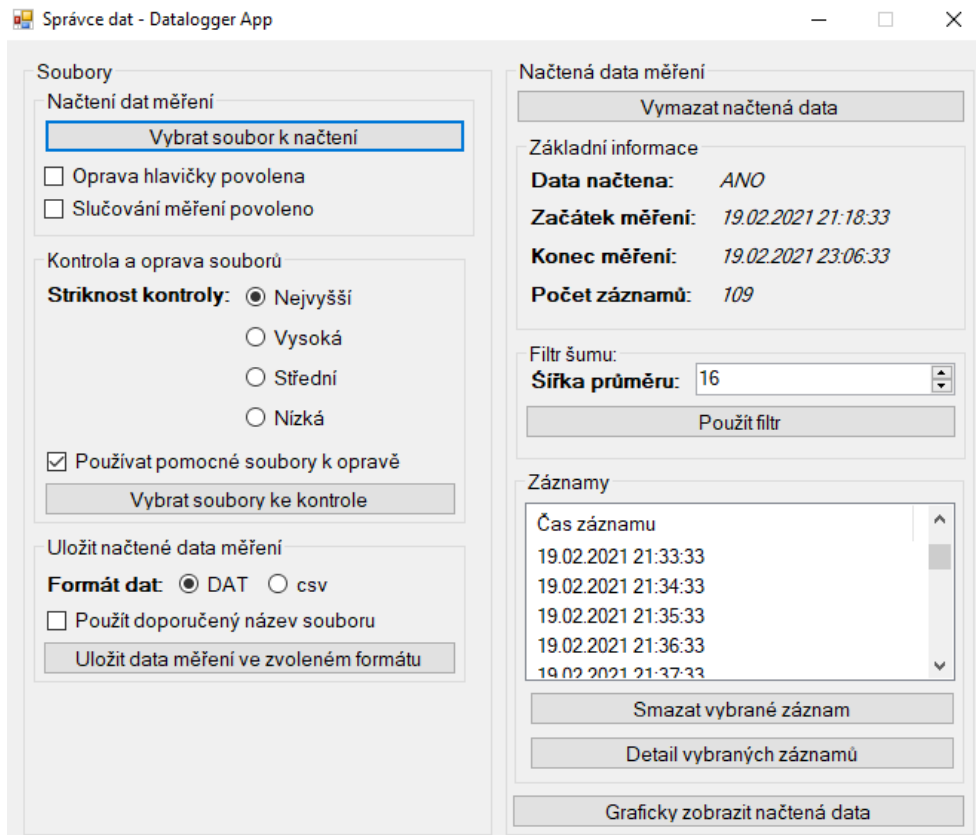
Poslední záložkou je *Konfigurace dataloggeru*, kde je možné provádět plnou konfiguraci dataloggeru. V záložce se i zobrazuje aktuální konfigurace dataloggeru. Po provedení změn je nutné konfiguraci uložit do paměti EEPROM pomocí tlačítka *Uložit konfiguraci do EEPROM*, které se nalézá v horní liště okna.

4.3.2 Správce dat

V okně správce dat uživatel pracuje s daty, které jsou načteny v objektu *MeasurementData*. Zobrazují se zde informace o měření jako např. začátek měření, konec měření, nebo počet záznamů. Ve správci dat si uživatel může zobrazit i detail jednotlivých záznamů, či část záznamů smazat. Načtená data si může zobrazit pomocí grafů po kliknutí na tlačítko *Graficky zobrazit načtená data*. Pokud je spektrum zatíženo šumem, může uživatel pomocí filtru šum odstranit. Filtr šumu využívá jednoduchý klouzavý průměr, pomocí něhož se vždy vypočítá počet pulzu daného kanálu podle předchozích x hodnot. Počet předchozích kanálů, které se počítají do průměru si může uživatel vybrat pomocí ovládací prvku *Šířka průměru*.

Kromě práce s již načtenými daty, okno správce dat nabízí i možnost načtená data uložit do souboru formátu csv, či binárního formátu (.DAT). Uživatel může v tomto okně načíst data ze souboru, či využít spojení dat z více souborů. V okně správce dat se také provádí kontrola a oprava souborů.

Okno správce dat se spouští z hlavního okna. Okno je buď možné spustit bez načtených dat, v takovém případě se využívá konstruktor, který nemá žádný parametr. Druhým způsobem, jakým se může správce dat spouštět, je již s načtenými daty z měření. V tomto případě se využívá konstruktor, který



Obrázek 4.5: Ukázka okna správce dat.

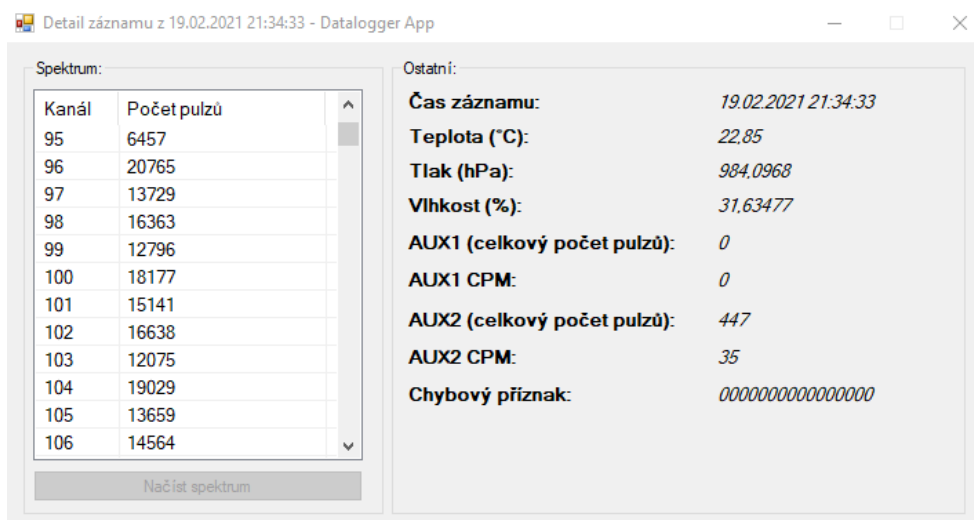
má jako argument objekt `MeasurementData`. V jednu dobu může být spuštěno více oken správce dat. Ukázka správce dat je na Obrázku 4.5.

Třetím formulářem, který se spouští ze správce dat pomocí tlačítka *Detail vybraných záznamů*, je okno detail záznamu. Okno detail záznamů má jedinou funkcionalitu a to je zobrazení detailu záznamu. Detailem záznamu se rozumí hodnoty spektra ionizujícího záření, hodnoty z pomocných měřicích zařízení, teplota, vlhkost, tlak, čas záznamu a chybové příznaky. Spektrum se při spuštění okna automaticky nenačítá. K načtení spektra v tabulce je nutné kliknout na tlačítko *Načíst spektrum*. Je to z důvodu, že načítání spektra trvá delší dobu a pokud se načítalo během spouštění formuláře tak se snižovala plynulost aplikace. Ukázka okna detailu záznamu je na Obrázku 4.6.

4.3.3 Grafické zobrazení dat

Posledním oknem aplikace je grafické zobrazení dat, které slouží k vizualizaci naměřených dat. Uživatel zde může nastavit rozsah záznamů, který se

4. REALIZACE POČÍTAČOVÉ APLIKACE

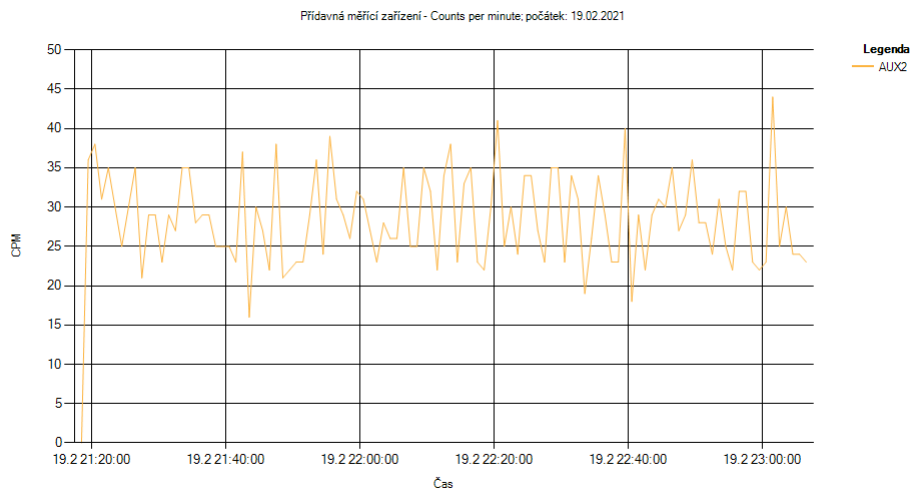


Obrázek 4.6: Ukázka okna detail záznamu.

v grafech zobrazuje. Rozsah je určen počátečním časem a koncovým časem záznamů. Hodnoty jednotlivých kanálů spektra se určují tím způsobem, že se u každého kanálu použije hodnota záznamu s koncovým časem a odečte se od toho hodnota záznamu, která předchází záznamu počátečního času. Pokud je jako počáteční záznam vybrán první záznam, odečítá se od hodnoty všech kanálů posledního záznamu hodnota 0. Uživatel také může vybrané grafy uložit do formátu jpg, png, či bmp. Výsledek uložení grafu hodnot CPM ve formátu png ukazuje Obrázek 4.7.

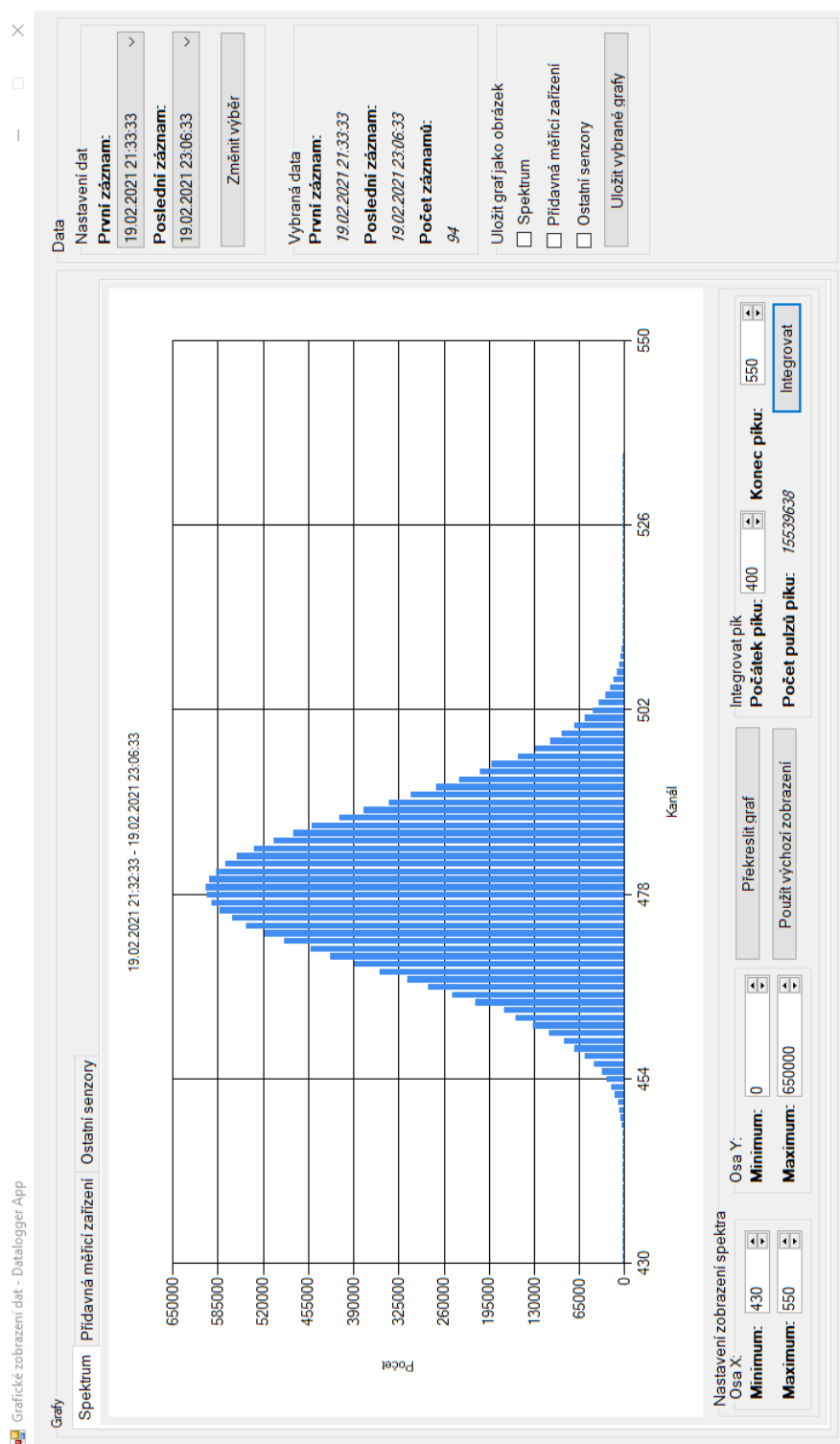
V okně se nalézají tři záložky. První záložkou je *Spektrum*, ve které se pomocí histogramu zobrazuje spektrum ionizujícího záření. Na ose x jsou vyneseny jednotlivé kanály, na ose y četnost pulzů každého kanálu. Kromě grafu se v záložce *Spektrum* vyskytuje i nastavení grafu, uživatel může manuálně nastavit rozsah hodnot na ose x a y. Dále je možné provést integraci plochy píku, která se provádí tím způsobem, že se sečte počet impulzů v rozsahu, který je určen počátečním kanálem, kde pík začíná a koncovým kanálem kde pík končí.

Další záložkou je záložka pro pomocné měřicí zařízení (v aplikaci označeno jako přídatná měřicí zařízení). V záložce *Přídatná měřicí zařízení* se zobrazuje graf hodnot pomocných měřicích zařízení v závislosti na čase. Na grafu je možné zobrazit buď celkový počet pulzů, nebo hodnoty impulzů za minutu (CPM) v čase. Poslední záložka má označení *Ostatní senzory*, ve které se zobrazují grafy teploty, vlhkosti a tlaku v závislosti na čase. Ukázka okna grafického zobrazení dat, konkrétně záložku zobrazující graf spektra ionizujícího záření je na Obrázku 4.8.



Obrázek 4.7: Ukázka uloženého grafu CPM ve formátu png.

4. REALIZACE POČÍTAČOVÉ APLIKACE



Obrázek 4.8: Ukázka okna grafické zobrazení dat.

Testování systému

Testování systému lze rozdělit do čtyř kroků. Prvním krokem byla realizace simulátoru. Simulátor má za úkol generovat signál gaussovského tvaru, který by přicházel i ze skutečného detektoru. Signál musí být generován s vlastnostmi, který zadá uživatel, či je vygenerován podle energetického spektra zkoumaného zářiče. Druhý a třetí krok se zaměřoval na to, jakým způsobem datalogger zpracovává analogový signál. Druhý krok se zaměřoval na zpracování impulzů v závislosti na jejich délce a prodlevách mezi nimi. Třetí krok se zaměřoval na zpracování impulzů, které mají stejnou délku, avšak jejich napětí je dané energetickým spektrem uživatelem zadaného zářiče. Impulzy přichází náhodně s různou prodlevou. Posledním krokem bylo testování funkcionalit dataloggeru a obslužné aplikace. Použité nástroje pro testování jsou popsány v příloze C Nástroje použité pro vývoj a testování systému.

5.1 Simulátor detektoru ionizujícího záření

Simulátor detektoru ionizujícího záření¹¹ se skládá ze dvou částí. První částí je osobní počítač s obslužnou aplikací, která zajišťuje konfiguraci simulátoru. Druhá část je vývojová deska Nucleo-144 L4R5ZI s mikrokontrolérem STM32L4R5, která simuluje fyzické chování obvodu detektoru. Výstupem mikrokontroléru je analogový signál udávající energii detekovaného fotonu/částice a spoušť. Mikrokontrolér komunikuje s počítačem přes USB. Pro účely diplomové práce byly realizované dva firmwary pro vývojovou desku Nucleo-144. První firmware zajišťuje generování impulzů gaussovského tvaru s vlastnostmi (délka pulzu, intenzita, prodleva mezi pulzy), které definuje uživatel. Druhý firmware zajišťuje skutečné chování detektoru, uživatel definuje energetické spektrum zářiče a simulátor vygeneruje impulzy s rozložením zadaného energetického spektra.

¹¹V předchozí části textu se detektorem ionizujícího záření rozumí měřicí část.

5.1.1 Generování impulsu gaussovského tvaru

Jako první bylo potřeba vymyslet, jakým stylem se bude generovat impuls gaussovského tvaru. Kromě samotného generování impulsu musí simulátor generovat signál spouště, který přechází do logické 1 v době, kdy impuls začíná a přechází do logické 0 tři μs před dosažením vrcholu křivky (důvod proč 3 μs se popisuje v 3.5.1 Konverze pomocí spouště). Vývojová deska Nucleo-144 L4R5ZI obsahuje 12bitový 1 MSample/s D/A převodník [68], který je využit ke generování analogového signálu. K vygenerování „hladkého signálu“ je potřeba alespoň 100 vzorků signálu, z tohoto důvodu je minimální délka impulsu 100 μs . Firmware pro generování pulzů používá 100 vzorků, firmware simulující skutečné chování detektoru používá 200 vzorků. Průběh signálu se generují podle následujících vzorečků [69]:

$$f(x) = e^{-x^2} * \textit{intenzita} \quad (5.1)$$

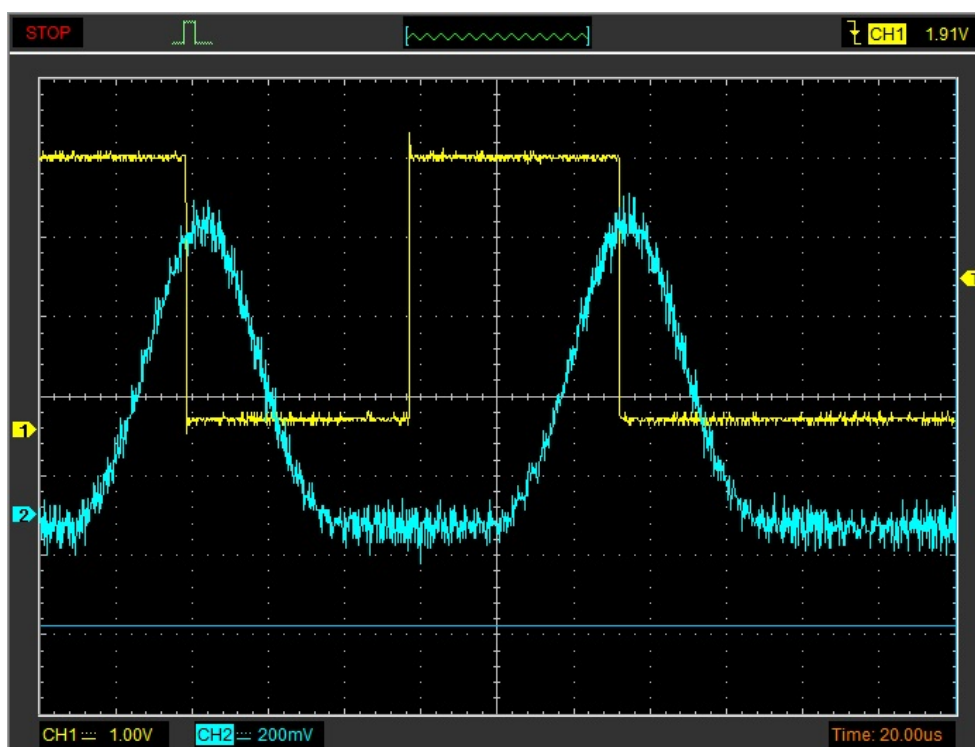
$$x = \frac{\textit{index} - \frac{\#vzorku}{2}}{y} \wedge y > 0 \quad (5.2)$$

Intenzita udává maximální napětí impulsu vyjádřeno jako 12bitové číslo. *Index* odpovídá indexu pole, do kterého se předpočítané hodnoty pro D/A převodník ukládají, *#vzorku* definuje velikosti tohoto pole. Počet vzorků musí být sudé číslo. Proměnná *y* nastavuje rychlost růstu signálu. Osvědčilo se *y* v rozsahu 40–75.

První firmware (firmware generující impulsy) hodnoty signálu předpočítává pomocí zmíněných vzorečků za běhu simulátoru. V případě tohoto firmwaru totiž nevdí počáteční zpoždění, způsobené výpočty, než se spustí generování pulzů. Avšak ve firmwaru, který simuluje skutečné chování detektoru, nejsou výpočty za běhu možné, protože by docházelo ke zpoždění impulsů. Tento firmware obsahuje předpočítanou tabulku, která je uložena v ROM. Předpočítaná tabulka obsahuje hodnoty vzorků signálu pro všechny možné hodnoty intenzity. Nevýhodou předpočítané tabulky je to, že roste velikost kódu¹². Roste čas nahrávání firmwaru do mikrokontroléru, což zdržuje testování, protože je nutné firmware pravidelně přehrávat. Z tohoto důvodu se u prvního firmwaru, nevyužívá vyhledávací tabulka.

Analogový signál se generuje pomocí D/A převodníku. D/A převodník v pravidelných intervalech, který udává časovač, provádí konverzi digitálního signálu na analogový. Je využita konverze za použití DMA a externí spouště, kterou představuje časovač. Konverze probíhají bez zásahu CPU. Délka impulsu je daná nastavením časovače, který spouští konverzi. Signál spouště je také řešen pomocí časovače, který vyvolá přerušení 3 μs před vrcholem impulsu. V obsluze přerušení se nastaví spoušť do logické 0.

¹²Tabulka je velká přes 1,6 MB.



Obrázek 5.1: Vygenerovaný pulz gaussovského tvaru včetně spouště. Žlutý signál zachycuje spoušť, tyrkysový signál zachycuje amplitudu.

Generování impulzu včetně spouště ukazuje Obrázek 5.1. Během zachycování průběhu signálu nebyl využit žádný filtr na straně osciloskopu. Oba signály jsou proto značně zašuměné, odpovídají ale reálnému signálu, který bude číst i datalogger. Signál je zašuměn z toho důvodu, že se pro propojení simulátoru a dataloggeru využívají nestíněné vodiče, které fungují jako anténa. Zašuměný signál představuje nejhorší možnou variantu signálu a takovýto signál se může objevit i během měření se skutečným detektorem. Po celou dobu testování se využívá zašuměný signál, představující právě tu nejhorší variantu a využívá se filtr šumu, který nabízí obslužná počítačová aplikace.

5.1.2 Simulace zářiče a detektoru

Generování pulzů o určité amplitudě může prozradit některé parametry dataloggeru, jako např. mrtvá doba dataloggeru a minimální délka impulzu, aby bylo napětí dobře přečteno. V reálném světě do vícekanalového analyzátoru přicházejí impulzy, jejichž amplituda je daná energetickým spektrem daného zářiče (či daných zářičů) – viz kapitola 1.5 Spektrum ionizujícího záření. Z tohoto důvodu byl realizován firmware pro Nucleo-144, který simuluje skutečné

chování detektoru a zářiče. Mechanismy simulace detektoru byly převzaty a upraveny z uživatelského manuálu [70] již diskutovaného emulátoru (viz 1.6 Simulace detektoru ionizujícího záření).

Kvanta ionizujícího záření přichází do detektoru s poissonovým rozdělením. Rozdělení je určeno hodnotou count per second (CPS), které určuje počet přichozích kvant do detektoru za sekundu. Poissonovo rozdělení je aproximováno pomocí často se opakujícího Bernoulliho pokusu. Nejprve je vypočítán práh úspěšnosti pomocí následujícího vzorečku:

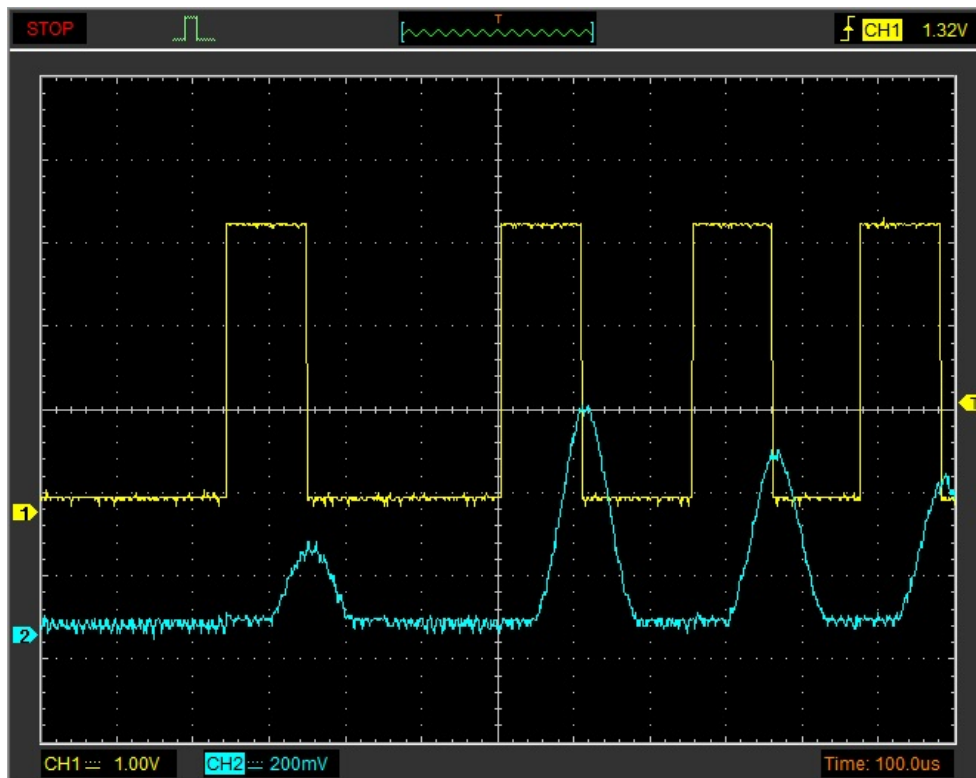
$$prah = \left\lceil \frac{CPS}{f_{genTimer}} * 1048575 \right\rceil \quad (5.3)$$

$f_{genTimer}$ odpovídá frekvenci generování náhodného čísla. CPS musí být výrazně menší než je frekvence generování náhodného čísla. Hodnota 1048575¹³ převádí desetinné číslo v rozsahu 0–1 na celé číslo, se kterým se lépe pracuje. Simulátor generuje náhodné číslo každých 10 μ s (frekvence je 100 kHz). Ke generování náhodného čísla se využívá generátor náhodných čísel, který je součástí mikrokontroléru STM32L4R5. Vygenerované číslo je v rozsahu 0–1048575 a porovnává se s hodnotou prahu úspěšnosti. Pokud je vygenerované číslo menší než práh úspěšnosti, znamená to, že se do detektoru dostala kvanta záření.

Kvanta přichází náhodně a mohou přijít i v době, kdy detektor zpracovává předchozí foton. I v případě simulátoru se tomu tak děje. Pokud se do detektoru dostane kvantum záření v mrtvé době, tak je tato událost ignorována a neprodlužuje mrtvou dobu. Pokud ale simulátor negeneruje žádný pulz, tak to znamená, že je dané kvantum zachyceno. Nyní je potřeba určit energii kvanta. Energie se generuje náhodně pomocí energetického spektra, které dodá uživatel před začátkem simulace. Spektrum je do simulátoru posláno jako četnost výskytu jednotlivých energií. Energie jsou v simulátoru převedené na rozsah D/A převodníku. Četnosti jsou namapovány do pole velkého přibližně 250 000 prvků. Četnost určuje, kolik prvků pole bude daná energie (převedená na rozsah DAC) zabírat. Energie detekovaného kvanta se určí tím způsobem, že se vygeneruje náhodné číslo, které udává index pole. Položka s vygenerovaným indexem odpovídá energii detekovaného kvanta.

Simulátor generuje pulzy gaussovského tvaru za použití předpočítané tabulky hodnot signálu. Délka impulzů je násobkem 200 μ s. O jaký násobek se jedná se určuje nastavením časovače spouště DAC. Délka impulzu odpovídá mrtvé době detektoru, kdy není možné detekovat další kvanta záření. Během generování impulzu se generuje i spoušť. Průběh generovaného signálu ukazuje Obrázek 5.2. Jako simulovaný zářič se použil Kobalt-60 s intenzitou 2500 CPS. Simulátor detektoru zachytil ve zkoumaném intervalu (1,2 ms) celkově 4 kvanta záření. Prodleva mezi impulzy je různá, stejně tak amplituda jednotlivých impulzů.

¹³Číslo 1048575 je zvoleno tak, aby bylo možné jednoduše zkrátit rozsah generovaných čísel bez použití operace modulo. Ke zkrácení se používá AND 0xFFFFF.



Obrázek 5.2: Ukázka běhu simulátoru detektoru. Žlutý signál zachycuje spoušť, tyrkysový signál zachycuje amplitudu.

5.2 Zpracování analogového signálu dataloggerem

Nejprve byl datalogger testován jakým způsobem reaguje na analogový signál z měřicí jednotky. Pro tyto testy byl využit simulátor s firmwarem, který generuje pulzy gaussovského tvaru zadaných vlastností. Datalogger umožňuje zpracování analogového signálu pomocí spouště, či *pokračované měření*. Oba způsoby jsou v následujících podkapitolách otestovány.

5.2.1 Zpracování signálu pomocí konverze se spouští

U konverze signálu za použití externí spouště je nutné zjistit jaká je minimální prodleva mezi spouštěmi, aby byl signál korektně přečten. Prodleva se nazývá jako mrtvá doba dataloggeru. V podkapitole 3.5.1 Konverze pomocí spouště bylo pomocí osciloskopu změřeno, že minimální mrtvá doba je 8 μs . Tvrzení je potřeba experimentálně ověřit.

Během experimentu byl datalogger spuštěn a probíhalo měření. Využívala se konverze pomocí spouště a jako spoušť byla nastavena sestupná hrana.

Tabulka 5.1: Závislost délky prodlevy a počtu detekovaných impulzů

Prodleva (μs)	Počet detekcí	Účinnost (%)
1	58666	5,87
2	199619	19,96
3	333229	33,32
4	333551	33,36
5	499232	49,92
6	500000	50
7	500001	50
8	995656	99,57
9	999999	100
10	1000000	100

Spoušť byla generována pomocí vývojové desky Nucleo-144 a vždy bylo vygenerováno milion rychlých spouští s prodlevou – 1–10 μs . Poté bylo zkoumáno, kolik spouští datalogger úspěšně zachytil a zpracoval. V tomto testu nebylo zkoumáno, jaké konkrétní napětí datalogger přečte, ale pouze počet detekovaných událostí. Závislost prodlevy mezi jednotlivými impulzy a počet detekovaných událostí popisuje Tabulka 5.1.

Jak ukazuje Tabulka 5.1, k detekci všech milionů spouští byla potřeba prodleva alespoň 10 μs . U prodlevy 9 μs se projevují přerušení od sekundového časovače a CPM časovače, které mají vysokou prioritu a předbíhají callback od A/D převodníku, který po vykonání nastavuje A/D převodník do stavu připraven. Tvrzení lze ověřit tím, že se místo milionu spouští generuje nižší počet spouští (např. 1000), aby se neprojevil vliv přerušení. Pokud je počet vygenerovaných spouští menší a prodleva nastavena na 9 μs , jsou detekované všechny události. Podobně tomu je i u prodlevy 8 μs , kde se kromě přerušení s vyšší prioritou projevují i spouště které jsou vygenerované dříve. Pro časování spouští se sice využívá časovač, avšak může docházet k tomu, že se některé spouště generují o pár desítek či stovek nanosekund dříve/později. Pokud je počet vygenerovaných spouští nižší než 200 a prodleva je nastavena na 8 μs , tak jsou detekované všechny spouště.

Pokud je prodleva mezi spoušti 5 až 7 μs , je detekována každá druhá spoušť, což odpovídá předpokladu, že mrtvá doba odpovídá 8 μs . Pro 3 a 4 μs je detekována každá třetí spoušť a pro 2 μs každá pátá. U prodlevy 1 μs se projevuje to, že přerušení od spouště má nejvyšší prioritu. Dochází k hladovění ostatních přerušení, protože procesor značnou část výpočetního času¹⁴ obsluhuje přerušení od spouště a zpracování dat z A/D převodníku se tudíž prodlužuje. Teoreticky kdyby prodleva mezi spouštěmi odpovídala délce obsluhy přerušení docházelo by k totálnímu vyhledování všech ostatních pře-

¹⁴1 μs odpovídá pouze 84 taktům procesoru dataloggeru a samotná obsluha přerušení pár desítek taktů zabere.

rušení a detekována by byla pouze jedna spoušť. Naměřené hodnoty ukazují, že mrtvá doba dataloggeru skutečně odpovídá minimálně 8 μs .

Během experimentu byla odhalena chyba v kódu firmwaru dataloggeru. Pro indikování toho, že probíhá konverze, se využívá globální proměnná. Globální proměnná se ale nikde nanastavovala a to mělo za důsledek, že se konverze spouštěla před tím, než skončila předchozí konverze. Pokaždé když byla vyvolána spoušť, vykonávala se větev obsluhy přerušení, kde se spouští konverze A/D převodníku. Tato větev je ale časově náročná (odpovídá délce zpracování přibližně 2–3 μs). Vyhladovění se projevilo právě u prodlevy spouště 3 μs , kdy bylo detekováno pouze 0,15 % spouští. Kvůli podivnému chování byla chyba detekována a opravena.

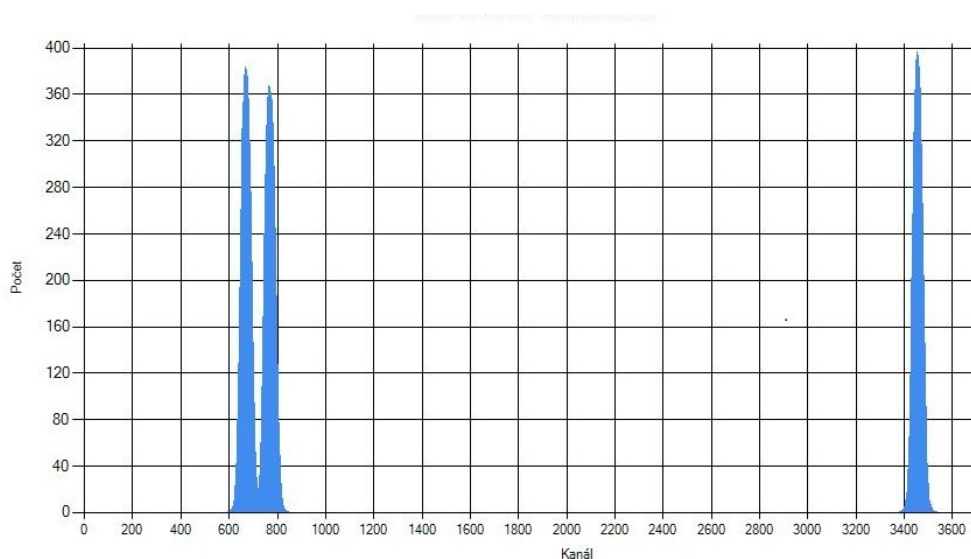
Experiment ukázal, že pokud je potřeba dataloggerem číst amplitudu více než 99 % detekovaných kvant, je nutná prodleva mezi spouštěmi alespoň 8 μs . Experiment taktéž odhalil, že pokud je prodleva mezi spouští 1 μs a nižší, snižuje se výrazně účinnost dataloggeru. Je to z důvodu, že dochází k vyhladovění ostatních přerušení a hlavního kódu na úkor přerušení od spouště. Platí ale pravidlo, že analogový signál musí být stabilní a držet maximální hodnotu po dobu alespoň 3 μs po spuštění spouště. Pokud je toto pravidlo porušeno dochází ke čtení špatných hodnot, které se nepříznivě projeví na výsledném spektru.

5.2.2 Zpracování signálu pomocí pokračovaného měření

Pokračované měření využívá toho, že je analogový signál z měřicí části vzorkován s frekvencí 48 kHz a průběžně jsou hledány vrcholy signálu. U *pokračovaného měření* dochází ke dvěma problémům. První problém je aliasing, kdy jsou dva (či více impulzů) čtený jako jeden. Druhým problémem je, že se nesprávně určí vrchol impulzu. Je potřeba zjistit, jaké parametry má mít impulz, aby nedocházelo k aliasingu či nesprávné interpretace vrcholu signálu.

Během tohoto experimentu byl datalogger puštěn a měření probíhalo v režimu *pokračovaného měření*. Práh byl nastaven na 300 (240 mV). Jako první se pozoroval vliv délky impulzu gaussovského tvaru na výsledné spektrum. Prodleva mezi impulzy byla nastavena takovým způsobem, aby nemohlo docházet k aliasingu – 100 μs . Generovaly se tři typy impulzů. První typ impulzů měl amplitudu nastavenou na 700 (odpovídá přibližně 560 mV), druhý typ impulzu měl amplitudu 800 (640 mV). Poslední typ impulzů měl amplitudu 3500 (2,82 V). Od každého typu bylo vygenerováno 20000 impulzů. Amplitudy byly zvoleny tím způsobem, aby v případě špatného měření došlo ke splnutí dvou píků do jednoho (amplituda 700 a 800). Impulzy s amplitudou 3500 mají strmější růst a klesání než impulzy s nízkou amplitudou, při špatném čtení dochází k velkému rozptylu. Impulzy s amplitudou 700 a 800 představují nízkoenergetické kvanta záření, impulzy s amplitudou 3500 představují vysokoenergetická kvanta záření. Simulátor byl připojen k dataloggeru za využití 10k pull-down rezistoru. Pokud nebyl rezistor připojen, vytvářel se další pík kolem 1000. ka-

5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.3: Referenční spektrum změřené za použití konverze pomocí spouště.

nálu nejspíše z důvodu nedostatečného vybití sample & hold kondenzátoru. Byly vyzkoušeny impulzy s různou délkou a byl sledován vliv délky impulzu na výsledné spektrum. Všechna zobrazená spektra v této kapitole mají vyfiltrováný šum pomocí klouzavého průměru o velikosti 16.

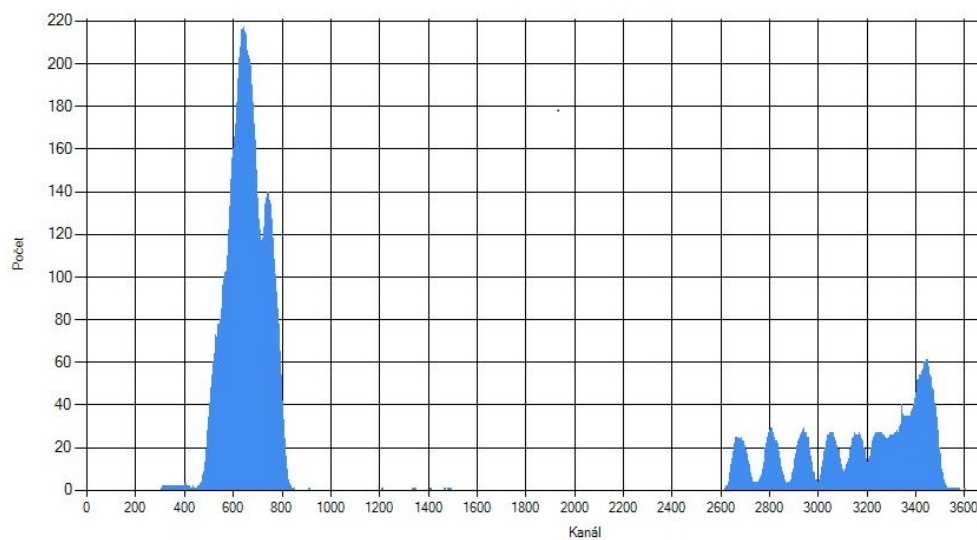
Referenční spektrum zobrazuje Obrázek 5.3. Spektrum bylo změřeno za ideálních podmínek, nevyužívalo se *pokračované měření*, ale měření pomocí spouště s délkou impulzu 200 μs . Ve spektru se objevují tři píky, které mají vrcholy v kanálech 669, 765 a 3451. Píky jsou úzké (cca. 50 kanálů FWHM) a jsou jednoznačně oddělitelné.

Energetické spektrum, pokud se využívá *pokračované měření* a délka impulzů je 100 μs , zachycuje Obrázek 5.4. Nízkoenergetické píky téměř splývají, vysokoenergetický pík je rozdělen na 7 menších píků s rozsahem přibližně 2700–3500. Impulzy s délkou 100 μs jsou příliš krátké na to, aby se daly změřit pokračovaným měřením s vzorkovací frekvencí 48 kHz.

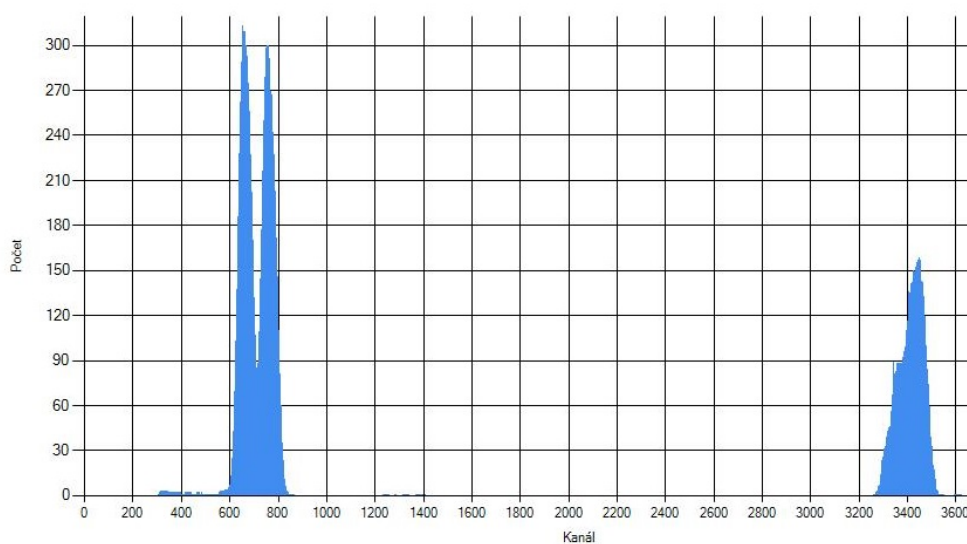
S rostoucí délkou impulzů klesá i šířka píků a vysokoenergetický pík se přibližuje k 3500. kanálu. Při délce impulzů 250 μs je spektrum již poměrně čitelné. Nízkoenergetické píky jsou již zřetelně odděleny a v případě vysoké energie se tvoří pouze jeden pík. Píky jsou však stále široké. Změřené spektrum s délkou impulzu 250 μs zobrazuje Obrázek 5.5.

Při délce impulzů 550 μs dochází k tomu, že nízkoenergetické píky mají podobnou šířku jako v případě referenčního spektra – přibližně 50 kanálů FWHM. Nízko energetické píky mají vrcholy v kanálech 683 a 783, vrcholy

5.2. Zpracování analogového signálu dataloggerem

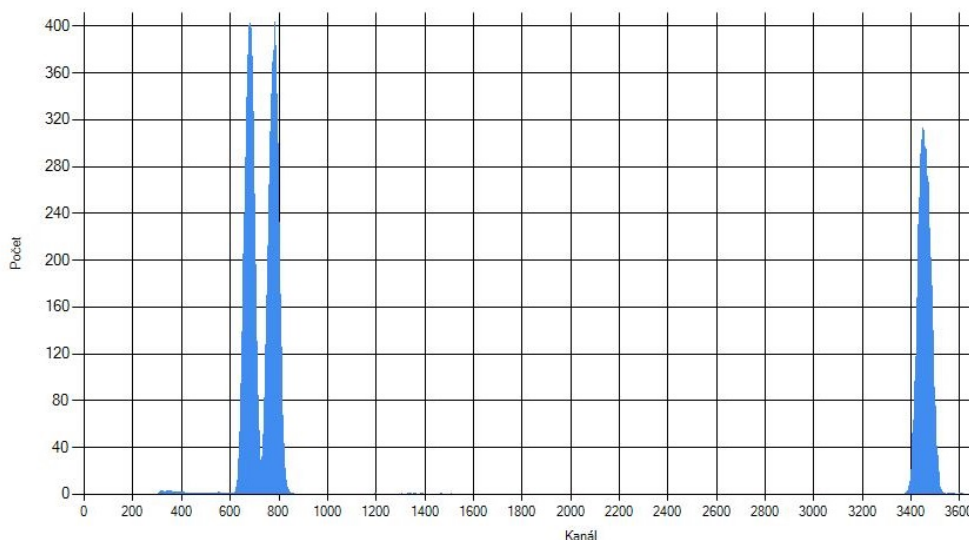


Obrázek 5.4: Energetické spektrum změřené pomocí *pokračovaného měření* – délka impulzů 100 μs.



Obrázek 5.5: Energetické spektrum změřené pomocí *pokračovaného měření* – délka impulzů 250 μs.

5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.6: Energetické spektrum změřené pomocí *pokračovaného měření* – délka impulzů 550 μs .

píků jsou blíže generovanému napětí než v případě referenčního spektra. Vysokoenergetický pík má vrchol v kanálu 3447, avšak FWHM je 62 kanálů. Je tedy přibližně o jednu třetinu širší. Změřené spektrum s délkou impulzů 550 μs ukazuje Obrázek 5.6. Pokud se dále prodlužuje trvání impulzu tak dochází k zužování vysokoenergetického píku. Nízkoenergetické píky již zůstávají stejné, 50 kanálu FWHM odpovídá nejlepší možné šířce při úrovni šumu způsobené vedení signálem mezi simulátorem a dataloggerem. Pro nízké a střední energie však platí, že délka impulzů 500–550 μs je dostačující. Pokud je potřeba měřit pouze vysoké energie je vhodné zvolit nižší úroveň zesílení napětí v předzesilovači.

Když je známá vhodná délka impulzu, je třeba změřit jaká musí být minimální prodleva mezi impulzy, aby nedocházelo k aliasingu. Vzorkovací perioda je přibližně 20,8 μs . Hypotéza je následující: aby nedocházelo k aliasingu musí být napětí mezi impulzy drženo pod hodnotu uživatelem určeného prahu po dobu minimálně 20,8 μs . Pro testování hypotézy se nevyužíval pulz gaussovského tvaru, protože by docházelo v nepřesnostech určování doby, kdy je impulz pod hodnotou prahu. Využíval se digitální signál nabývající 0 V, nebo 3,3 V. Digitální signál byl přiveden na analogový vstup dataloggeru. Logická 1 reprezentovala impulz (napětí nad hodnotou prahu), logická 0 prodlevu mezi impulzy (napětí pod hodnotou prahu). V tomto experimentu nezáleželo na výsledném spektru, ale celkovém počtu detekovaných impulzů. Zkoumal se počet detekovaných impulzů v závislosti na délce prodlev mezi impulzy. Vždy bylo vygenerováno 100 000 impulzů, které byly odděleny prodlevou s definovanou

Tabulka 5.2: Pozorování vlivu délky prodlevy na aliasing

Délka prodlevy (μs)	Počet detekcí	Účinnost (%)
0	1	0
5	23097	23,1
10	47153	47,2
15	71152	71,2
19	90358	90,4
20	95146	95,1
20,8	99545	99,5
21	99901	99,9
22	100000	100

délkou. Výsledky experimentu popisuje Tabulka 5.2.

Pokud mezi jednotlivými impulzy není žádná prodleva a napětí je stále nad hodnotou prahu, všechny impulzy splývají do jednoho. S přibývajícím délkou prodlevy se snižuje riziko, že nastane aliasing. Hypotéza byla potvrzena, protože pokud je délka prodlevy nižší než 20,8 μs , dochází k aliasingu. Aliasing se vyskytoval dokonce i u prodlevy 21 μs . K aliasingu nedochází pokud je prodleva mezi impulzy 22 μs a vyšší.

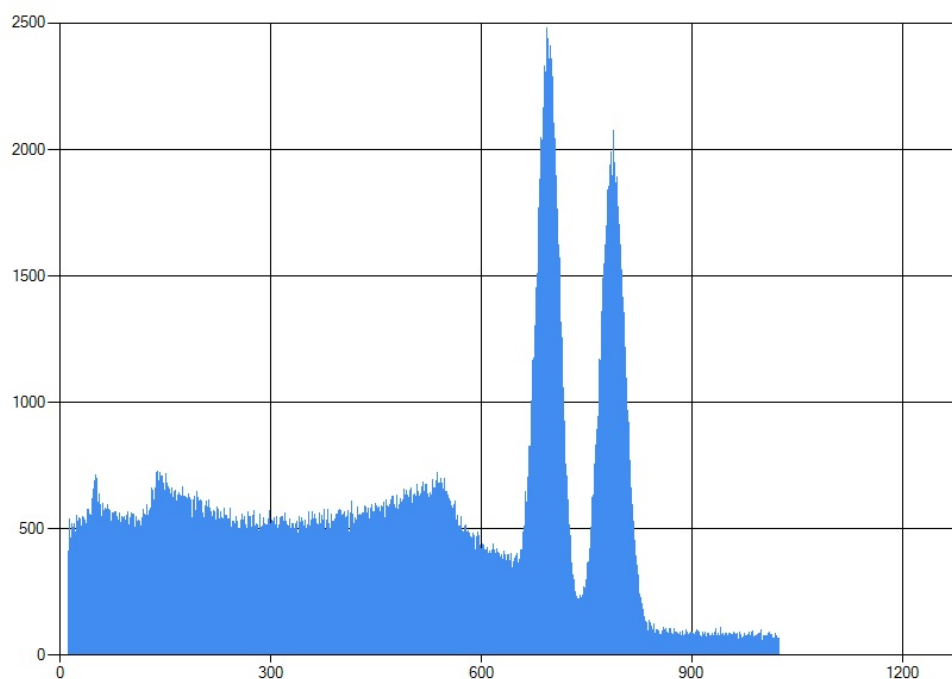
Experimenty ukázaly, že vhodná délka impulzů je 500–550 μs , pokud je kratší, může docházet ke splývání píků, které mají podobné energetické hodnoty. Pokud není minimální prodleva mezi impulzy alespoň 22 μs , může docházet k aliasingu, který způsobí, že dva, či více impulzů jsou interpretovány jako jeden impulz.

5.3 Test se simulovaným zářičem

K testu se simulovaným zářičem se využívá firmware pro Nucleo-144, který simuluje skutečný detektor. Jako referenční spektra byla použita energetická spektra, která byla změřena pomocí scintilačního detektoru s NaI(Tl) krystalem. Spektra ionizujícího záření jsou volně ke stažení z webových stránek [71]. Test se simulovaným zářičem měl otestovat, jakým způsobem datalogger zpracovává impulzy, které by přicházely ze skutečného detektoru ionizujícího záření. Jako zářič se využíval Kobalt-60, jehož energetické spektrum je na Obrázku 5.7. Simulátor detektoru generoval impulzy s rozdělením dle tohoto spektra. Count rate zářiče bylo nastaveno na 1500 CPS.

Během prvních dvou testů se simulovaným zářičem byl datalogger spuštěn v módu konverze analogového signálu za využití spouště. Délka impulzů byla 200 μs . První měření trvalo 30 minut a 39 sekund. Během této doby prolétlo simulovaným detektorem 2757164 kvant záření. Detekováno jich bylo 2095917. Zbylá kvanta prolétla detektorem v jeho mrtvé době. Pokud se vydělí celkový počet kvant s délkou měření, vychází hodnota 1499,27 CPS, což odpovídá

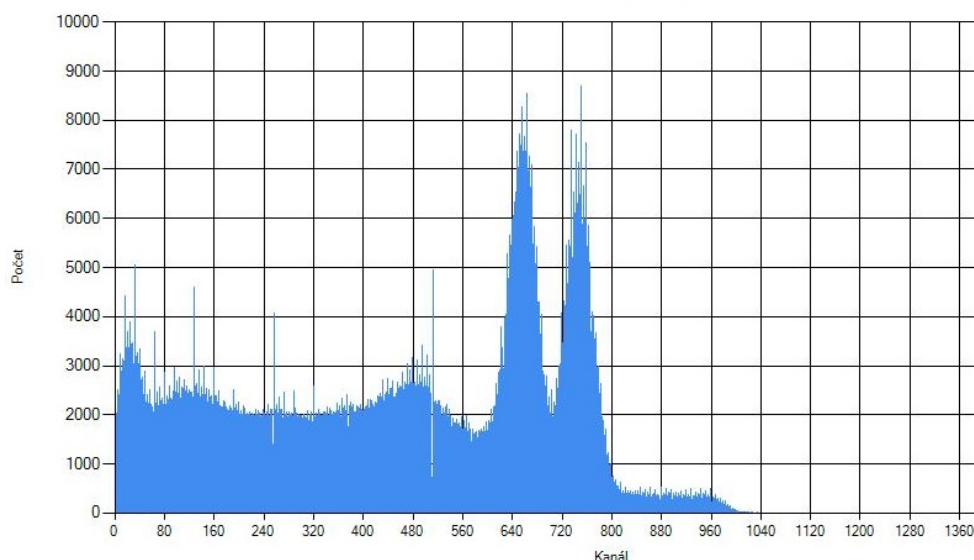
5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.7: Referenční energetické spektrum zářiče Kobalt-60. Zdroj spektra [71].

nastavené hodnotě count rate 1500 CPS. Datalogger detekoval všech 2095917 impulzů, což se dalo očekávat, protože prodleva mezi spouštěmi byla delší než minimálních 10 μ s, která byla změřena v 5.2.1 Zpracování signálu pomocí konverze se spouští.

Změřené spektrum bez úprav zobrazuje Obrázek 5.8. Spektrum je sice čitelné a píky jsou oddělitelné, avšak obsahuje šum, který způsobuje úzké čáry. Tyto čáry působí rušivě, proto se využívá filtr s klouzavým průměrem 4, který tento šum odstraní. Výsledné spektrum je zobrazeno na Obrázku 5.9. Ve spektru jsou dva výrazné a dobře oddělené fotopíky odpovídající energiím 1,17 MeV a 1,33 MeV. Dalším výrazným píkem je Comptonova hrana v energii 963,4 keV. Ve spektru Kobalt-60 by měla být i druhá Comptonova hrana v energii 1118,1 keV. Tato hrana ale není zřetelná ani v referenčním spektru na Obrázku 5.7. Dalším píkem je pík zpětného rozptylu. V nízkých kanálech dochází k šumu. Ostrý a vysoký pík je v kanálu 0. Pokud se nepoužije filtr, tak je tento pík dokonce nejvyšší. Je to způsobené tím, že šum způsobuje, že amplitudy s nízkým napětím se mají dočasně napětí i pod 0 V (jak ukazuje Obrázek 5.1). A/D převodník při použité konfiguraci není schopný měřit záporné napětí a dochází k interpretaci tohoto napětí jako hodnota 0. Pokud by na A/D převodník bylo přivedeno vyšší negativní napětí, mohlo by dojít ke



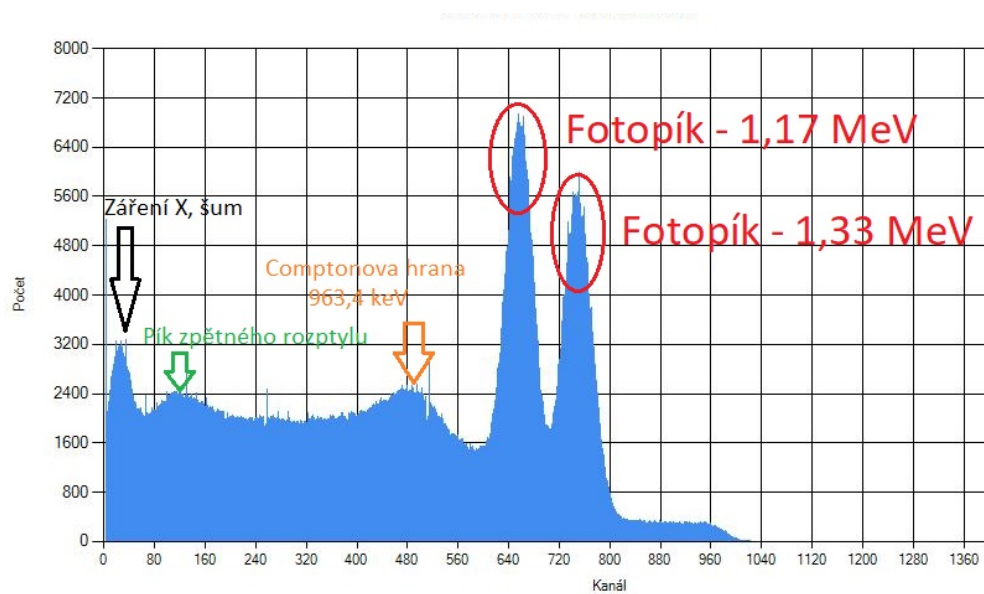
Obrázek 5.8: Změřené spektrum simulovaného zářiče Kobalt-60, zašuměné.

zničení mikrokontroléru. Informace o spektru Kobaltu-60 byla čerpána z volně dostupných zdrojů [25].

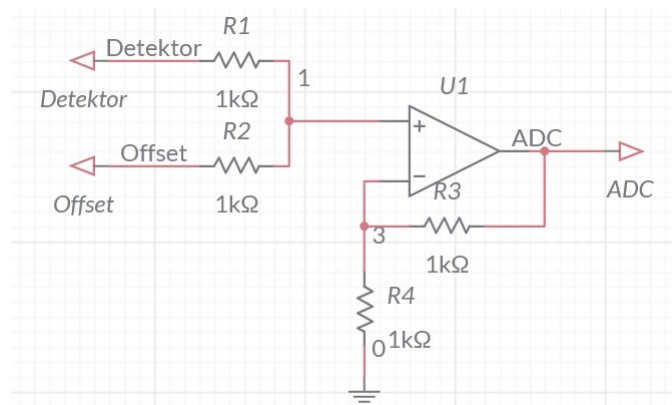
Pík v kanálu 0 působí rušivě. Ke všemu z důvodu šumu může dojít i ke ztrátě nízkoenergetických píků, protože negativní napětí je vždy interpretováno jako 0. Měření lze zlepšit tím způsobem, že se na vstup A/D převodníku přivede napětí, které bude o pár desítek (či stovek) mV zvýšené. Během testování se tohoto offsetu dosahovalo tím způsobem, že se celé spektrum posune o fixní počet kanálů doprava. V případě hardwarové implementace je možné využít např. neinvertující sčítací zesilovač [72]. Ukázkou zapojení zobrazuje Obrázek 5.10. Detektor odpovídá výstupu detektoru ionizujícího záření, Offset udává, o jaké napětí je amplituda z detektoru posunuta. Posun napětí může být proměnlivý pokud je rezistor R2 nahrazen potenciometrem. Pokud se využívá posunu napětí, je potřeba si dát pozor na to, aby výsledné napětí nepřekročilo 4 V. Toto napětí je maximální povolené napětí pro analogový vstup pro mikrokontrolér STM32F401 [73]. Maximální napětí, co může mikrokontrolér přesně přečíst, odpovídá hodnotě V_{REF+} , která je 3,3 V.

Další test zkoumal, jak se změní výsledné spektrum, pokud se použije zvýšené napětí o konstantní hodnotu na vstupu A/D převodníku dataloggeru. Napětí se zvyšovalo softwarově tím způsobem, že bylo generované spektrum v simulátoru posunuto o 150 kanálů doprava – napětí je zvýšeno přibližně o 120 mV. Měření probíhalo po dobu 36 minut a 10 sekund. Za tuto dobu prolétlo detektorem 3252339 kvant. Detekováno jich bylo 2474384. Průměrná hodnota

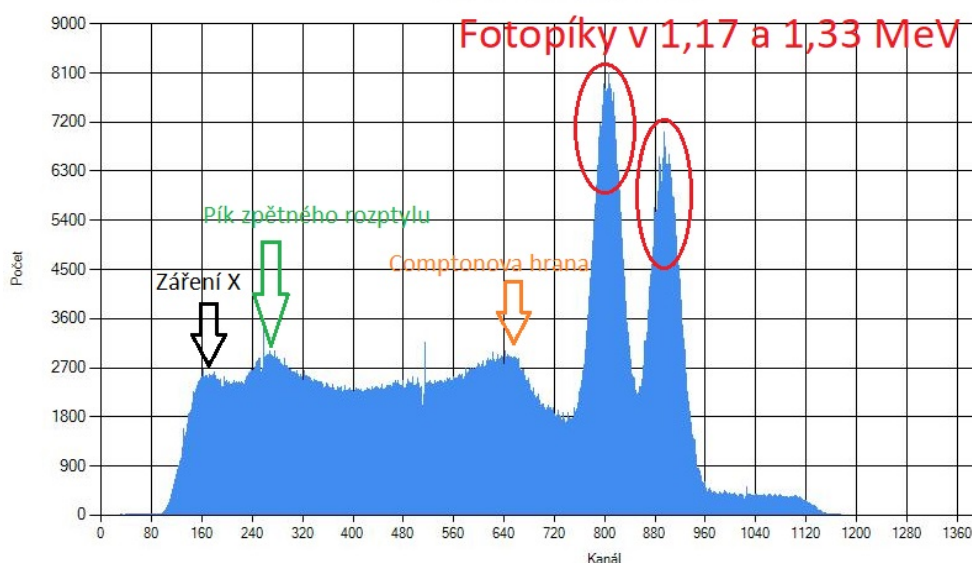
5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.9: Změřené spektrum simulovaného zářiče Kobalt-60, s aplikovaným filtrem.



Obrázek 5.10: Neinvertující sčítací zesilovač.

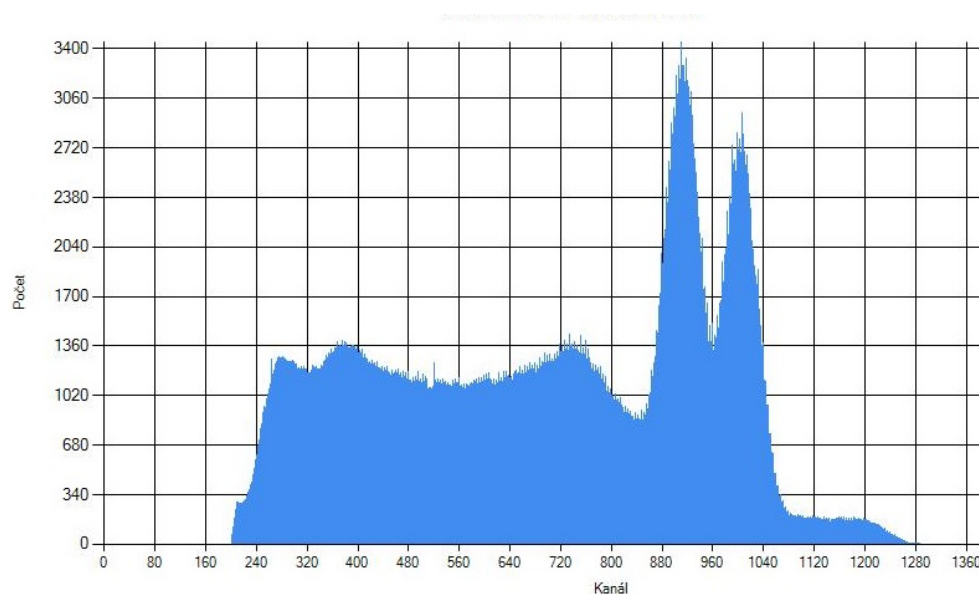


Obrázek 5.11: Změřené spektrum simulovaného zářiče Kobalt-60 posunuté o 150 kanálů doprava, s aplikovaným filtrem.

CPS vychází na 1498,7. Výsledné spektrum po aplikaci filtru s klouzavým průměrem o velikosti 4 zobrazuje Obrázek 5.11. Celé spektrum je skutečně posunuto o 150 kanálů doprava. Zřetelné zůstaly oba fotopíky, Comptonova hrana a i pík zpětného rozptylu. Výrazný pík, který se objevoval v nízkých energiích ve spektru na Obrázku 5.9, je v posunutém spektru odstraněn a v nízkých energiích se objevuje malý pík, který odpovídá nízkoenergetickému záření X. Zvýšení napětí na vstupu může pomoci při detekci nízkoenergetických kvant.

Posledním testem se simulovaným zářičem byl test, kdy byl datalogger spuštěn v režimu *pokračovaného měření*. Práh měření byl nastaven na 200. kanál. Délka impulzů simulátoru detektoru byla nastavena na 400 μs , protože 200 μs by byla nedostatečná (jak vysvětluje kapitola 5.2.2 Zpracování signálu pomocí pokračovaného měření). Vzhledem k prodloužení délky impulzů byla snížena hodnota CPS pro test na 800. Referenční spektrum v simulátoru bylo posunuto o 250 kanálů doprava (přibližně 200 mV). Simulace probíhala po dobu 31 minut a 45 sekund. Během této doby proletělo detektorem celkově 1525587 kvant záření, detekováno jich bylo 1149180. Průměrná hodnota CPS vychází na 800,97. Dataloggerem bylo detekováno 1168988 impulzů, což je přibližně o 20000 impulzů více, než simulátor vygeneroval. Dochází k tomu, že jeden impulz s nízkou amplitudou je dataloggerem detekován víckrát z důvodu šumu signálu. Výsledné spektrum je zašuměné více než v případě použití konverze pomocí spouště. Je to způsobené tím, že impulzy o délce 400 μs jsou

5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.12: Spektrum Kobaltu-60 změřené pomocí *pokračovaného měření*, s aplikovaným filtrem šumu.

stále příliš krátké, aby *pokračované měření* dosahovalo stejných kvalit jako v případě konverze pomocí spouště. Aby byl šum odstraněn využíval se filtr o velikosti klouzavého průměru 10. Výsledné spektrum zobrazuje Obrázek 5.12. Fotopíky jsou odděleny, avšak ne tak zřetelně jako v případě *pokračovaného měření*. Ostatní významné píky jako Comptonova hrana, zpětný pík, či pík X záření jsou viditelné.

Test se simulovaným zářičem ukázal, že datalogger je schopný zpracovávat i signály z detektoru ionizujícího záření. Impulzy přicházely do dataloggeru s takovým rozdělením (jak časovým tak i amplitudou), jako by přicházely ze skutečného detektoru. V této kapitole byl také nastíněn způsob, jak lze přesněji měřit nízkoenergetické hodnoty pomocí konstatního zvýšení napětí na vstupu dataloggeru.

5.4 Test funkcionalit systému

V předchozích kapitolách se testovalo, jakým způsobem reaguje datalogger na příchozí signál z měřicí části. Během předchozích testů se nepřímě testovaly i funkcionality obslužné počítačové aplikace a dataloggeru, avšak primární důraz byl kladen na změřené spektrum záření. V testu funkcionalit systému se naopak neřeší již samotné spektrum, ale jednotlivé funkcionality systému.

Tabulka 5.3: Rychlost stahování souborů měření z dataloggeru. Rychlosti stahování jsou uvedené v kB/s.

Bezdrátové připojení			
Rychlost (kB/s)	Minimální	Maximální	Průměrná
Soubor 1:	5,041	7,453	6,576
Soubor 2:	6,171	7,416	6,979
Soubor 3:	6,836	7,674	7,356
Fyzické připojení (USB)			
Rychlost (kB/s)	Minimální	Maximální	Průměrná
Soubor 1:	7,032	7,214	7,111
Soubor 2:	8,111	8,160	8,140
Soubor 3	8,209	8,247	8,225

5.4.1 Manipulace se soubory měření

Prvním testem funkcionalit byl test, který se zaměřoval na práci se soubory měření, které jsou uloženy na paměťových kartách. Test byl rozdělen na dvě fáze. První fází bylo testování, jakou rychlostí probíhá stahování dat z dataloggeru do počítače. Druhou fází byl test funkce, která opravuje poškozené soubory. Manipulace se soubory měření se zaměřovalo na celý proces, jakým způsobem se uživatel dostane k souborům měření. Testy pokrývaly jednak samotný přenos mezi dataloggerem a počítačem, ale i kontrolu a případnou opravu stažených souborů.

Rychlost stahování byla změřena pro oba typy připojení počítače k dataloggeru – jak bezdrátové připojení, tak i fyzické připojení pomocí USB. Stahování souboru probíhalo v době, kdy neprobíhalo měření. Stahovaly se tři soubory o různých velikostech – 16492 (Soubor 1), 131488 (Soubor 2) a 246484 (Soubor 3) bajtů. Každý soubor byl stažen 10x za sebou z obou čteček a měřilo se, za jak dlouho se soubor stáhne. Tabulka 5.3 ukazuje minimální, maximální a průměrnou rychlost stahování souborů z dataloggeru.

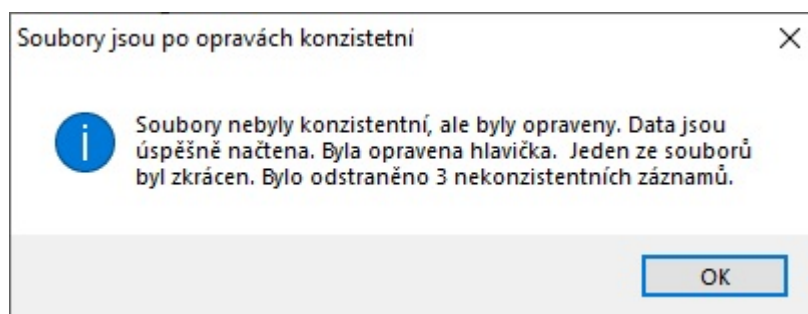
V případě bezdrátového přenosu je znatelný rozdíl mezi naměřenou minimální a maximální rychlostí. Rychlost přenosu je v případě bezdrátové komunikace proměnlivá a záleží na vytíženosti sítě. U přenosu souborů přes USB je rychlost stahování stabilní. Dochází ale k situaci, že menší soubor má nižší přenosovou rychlost než soubor větší. Je to dané tím, že na začátku posílání souboru musí datalogger provést inicializaci daného souboru. Jedná se o nalezení souboru pomocí ID a otevření samotného souboru. Soubor se hledá sekvenčně, pokud se čte soubor s vysokým ID, tak datalogger musí „prolistovat“ většinu obsahu paměťové karty, což je časově náročné. Pokud se stahuje větší soubor, režijní doba inicializace je oproti celému stahování zanedbatelná. Pokud se ale stahuje malý soubor, tak režie má znatelný podíl na celkové délce stahování a rychlost stahování tak klesá.

Přenosové rychlosti souborů jsou nízké. Pro bezdrátovou komunikaci je to průměrně kolem 7 kB/s a pro fyzickou komunikaci cca. 8 kB/s. Je to dané tím, že obslužná počítačová aplikace využívá méně agresivní timeouty, které zaručují vyšší spolehlivost přenosu, ale snižují přenosovou rychlost. Další z důvodů je, že se soubor posílá po částech a datalogger vždy čte pouze tu část souboru, o který počítačová aplikace žádá. Čtení souboru je časově náročné, vzniká proto prodleva mezi žádostí o část souboru a samotného posílání dat z dataloggeru do počítače. Tato čekací doba výrazně snižuje rychlost přenosu, protože zlomek času se žádná data neposílají. Pokud by byl použit mikrokontrolér s větší RAM, bylo by možné si části souboru spekulativně přednačítat a tím by klesala prodleva mezi žádostí a posílání dat. Pro čtení dat z měření je možné využít i externí čtečku paměťových karet třetí strany, které nabízí mnohem vyšší přenosové rychlosti. K tomu je ale potřeba mít k dataloggeru fyzický přístup, aby bylo možné vyjmout paměťové karty.

Druhá fáze testu manipulace se soubory se zaměřovala na testování, zdali korektně fungují funkce, které kontrolují, popř. opravují soubory. Funkce jsou součástí obslužné počítačové aplikace. Chyby do souborů měření byly vneseny manuálně pomocí binárního editoru. Nejběžnější chyba při kontrole dvou souborů je to, že jeden ze souborů je delší než druhý. Tato chyba může nastat, když se stahuje za běhu měření soubor, do kterého se ukládají data z měření. Pro kontrolu souboru je potřeba mít stažené soubory z obou paměťových karet. Mezi stahováním souboru z první a druhé karty může proběhnout zápis nových dat na obě karty. Později stažený soubor bude obsahovat data, která během stahování souboru z první karty ještě neexistovala. Pokud se do správce dat vloží dva soubory z měření, které nemají stejnou velikost a nastaví se striktnost kontroly na jinou úroveň než na nejvyšší, tak je soubor opraven a načten. Zkracování souboru funguje dle očekávání.

Další chybou, která může nastat, je přepsání bajtů v uloženém souboru, či zapsání špatných dat do paměťové karty na straně dataloggeru. Tato chyba může poškodit buď hlavičku souboru, či některý ze záznamů. Obslužná aplikace nabízí funkci, která umí detekovat poškození hlavičky a pokusit se jí opravit. Hlavička obsahuje užitečné konfigurační bity, neužitečné (nevyužité) konfigurační bity a kontrolní součet hlavičky. Pro testování funkce na opravu hlaviček byly použity tři různé dvojice souborů:

1. Soubor z první paměťové karty obsahuje chybu v užitečných bitech hlavičky, soubor z druhé karty obsahuje chybu v neužitečných bitech hlavičky.
2. Soubor z obou paměťových karet obsahuje různou chybu v neužitečných bitech hlavičky a kontrolním součtu.
3. Soubor z obou paměťových karet má nenávratně poškozenou hlavičku (chyba v užitečných bitech hlavičky).



Obrázek 5.13: Dialogové okno indikující provedené opravy souboru.

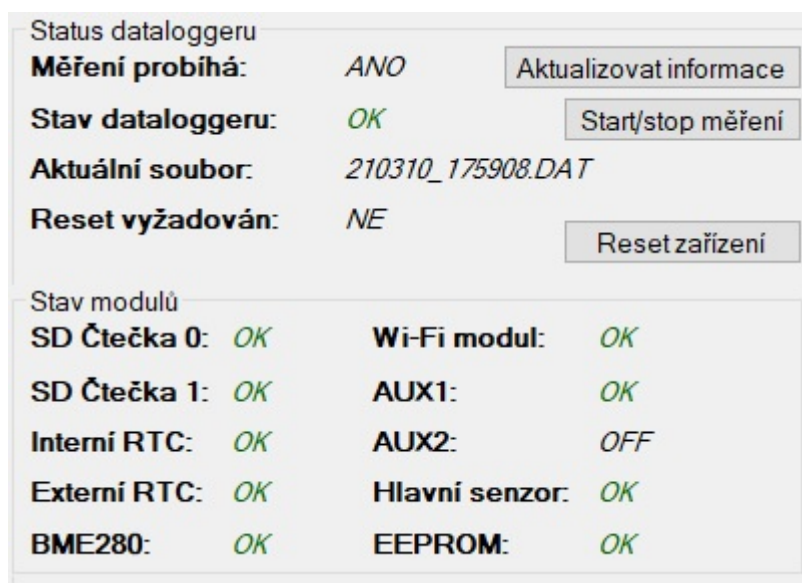
Dvojice byly zvoleny takovým způsobem, aby otestovaly všechny možné cesty opravy hlavičky, které mohou nastat. První dvě dvojice souborů by měly být opravitelné, poslední dvojice je neopravitelná. Kromě toho jsou soubory různě dlouhé, projeví se i zkracování souborů. K povolení opravy hlavičky je nutné ve správci dat mít nastavenou striktnost kontroly na střední, či nízkou. Test dopadl dle očekávání, první dvojice souborů byla úspěšně opravena, poslední dvojici souborů obslužná aplikace neopravila.

Přepsání bajtů se může kromě hlavičky projevit i v jednotlivých záznamech. Poslední testovací dvojice obsahovala opravitelnou chybu v hlavičce a 3 různé záznamy. Kromě toho byl jeden ze souborů delší než ten druhý. K opravě záznamů je potřeba mít v aplikaci nastavenou striktnost kontroly na nízkou. Pokud se dvojice takto poškozených souborů vloží do správce dat v aplikaci, tak se uživateli zobrazí dialogové okno, které informuje o úspěšné opravě souboru a jaké akce byly provedeny. Dialogové okno ukazuje Obrázek 5.13.

Paměťová karta může úplně selhat, či se na ní může rozbít filesystém. Rozbití filesystému nastávalo často během realizace a testování při využití levné paměťové karty. Obslužná počítačová aplikace neobsahuje nástroje na opravu poškozeného filesystému. K opravě filesystému je třeba využít programy třetích stran. Z tohoto důvodu datalogger obsahuje dvě čtečky paměťových karet, aby v případě selhání jedné karty byla data uložena i na kartě druhé. Při selhání jedné z karet není možné využít kontrolu správnosti dat.

5.4.2 Selhání modulů dataloggeru během měření

Další z testů se zaměřoval na to, jakým způsobem se datalogger zachová, když selžou moduly zařízení během probíhajícího měření. Selhávání moduly dataloggeru bude simulováno tím způsobem, že odpojitelné moduly byly postupně odpojovány z desky plošných spojů. Během testu selhávání modulů byl datalogger k počítači připojen pomocí USB, aby bylo možné pozorovat i vliv selhání Wi-Fi modulu. Očekávané chování je to, že datalogger detekuje selhání dané



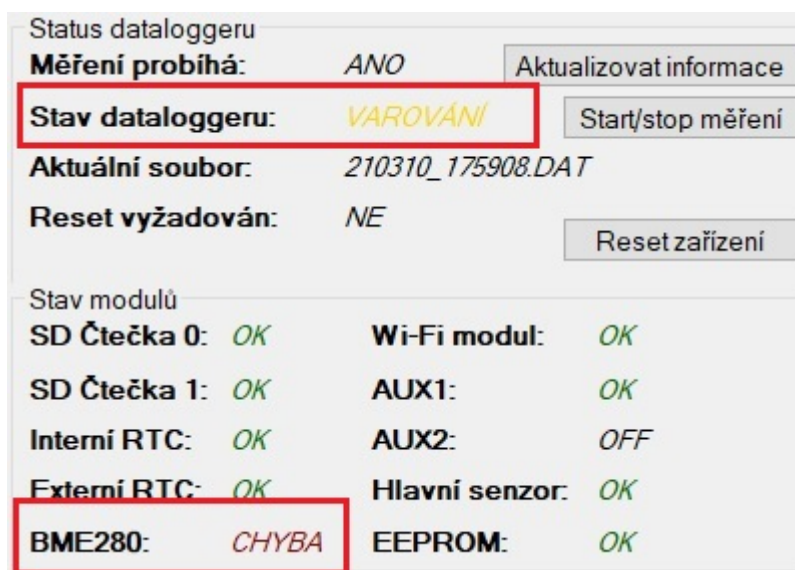
Obrázek 5.14: Výchozí stav měření – všechny moduly jsou funkční.

části systému a uživateli to dá najevo v obslužné aplikaci. Selhání části systému by nemělo přerušit měření.

Měření začínalo ve stavu, kdy byly připojeny všechny moduly na desce plošných spojů a byly v provozu. K dataloggeru byl připojen simulátor detektoru, jako záříč byl nastaven Kobalt-60. Kromě simulátoru byl jako pomocné měřicí zařízení připojen detektor založený na Geiger-Müllerové trubici, který se využíval v bakalářské práci [8]. Druhé pomocné měřicí zařízení je vypnuto. Status dataloggeru po začátku měření v obslužné počítačové aplikaci ukazuje Obrázek 5.14.

Jako první byl odebrán modul měřící teplotu, vlhkost a tlak (BME280). Po odebrání tohoto modulu přešel datalogger do stavu varování a v obslužné aplikaci je indikované selhání modulu jako chyba u položky BME280. Měření nadále probíhá a jako teplota, vlhkost a tlak se zobrazují poslední hodnoty, které byly změřeny před selháním modulu. Status dataloggeru po selhání BME280 je zobrazen na Obrázku 5.15.

Dále bylo postupně odpojen externí RTC modul, Wi-Fi modul a jedna ze dvou karet byla vytažena ze čtečky (simulace selhání paměťové karty). Po odpojení těchto částí přechází datalogger do stavu chyba, měření nadále probíhá. Podobně jako u selhání BME280 je i selhání vyjmenovaných částí zobrazeno v obslužné počítačové aplikaci. Po odebrání Wi-Fi modulu nebylo možné s dataloggerem přibližně 5-10 minut komunikovat přes USB. Je to dané tím, že datalogger zjišťuje selhání Wi-Fi modulu tím způsobem, že přestane odpovídat na příkazy. Datalogger příkazů vysílá více a až po několikanásobné neodpovědi vyhodnocuje, že je Wi-Fi modul poškozen. Timeout každého příkazu je



Obrázek 5.15: Status dataloggeru v obslužné aplikaci – Selhání modulu BME280.

30 sekund. Během čekání na odpověď na příkaz dojde k vytížení příkazového řadiče a žádosti od počítače připojeného přes USB jsou zahazovány. Status dataloggeru po selhání externího RTC modulu, Wi-Fi modulu a jedné ze dvou paměťových karet je zobrazen na Obrázku 5.16.

Jako poslední byla odebrána zbylá paměťová karta. Datalogger přešel do stavu selhání. Ve stavu selhání sice měření nadále probíhá, ale data se již nikam neukládají. Jediná možnost jak číst aktuální data z dataloggeru je pomocí USB. Status dataloggeru po odebrání zbylé paměťové karty je zobrazen na Obrázku 5.17.

Pořadí „selhávání“ modulů bylo vybráno tím způsobem, aby se projeví všechny chybové stavy, které datalogger má (o chybových stavech více v kapitole 3.13 Chybové stavy dataloggeru). Test selhávání modulů měl ověřit, jestli datalogger správně reaguje na selhání jednotlivých modulů. To znamená, zda při selhání modulu neskončí měření a jestli je selhání modulu možné vyčíst z obslužné počítačové aplikace. Test skončil úspěšně, protože i po selhání většiny modulů měření nadále probíhalo a bylo možné pomocí počítače připojeného přes USB vyčíst naměřené hodnoty.

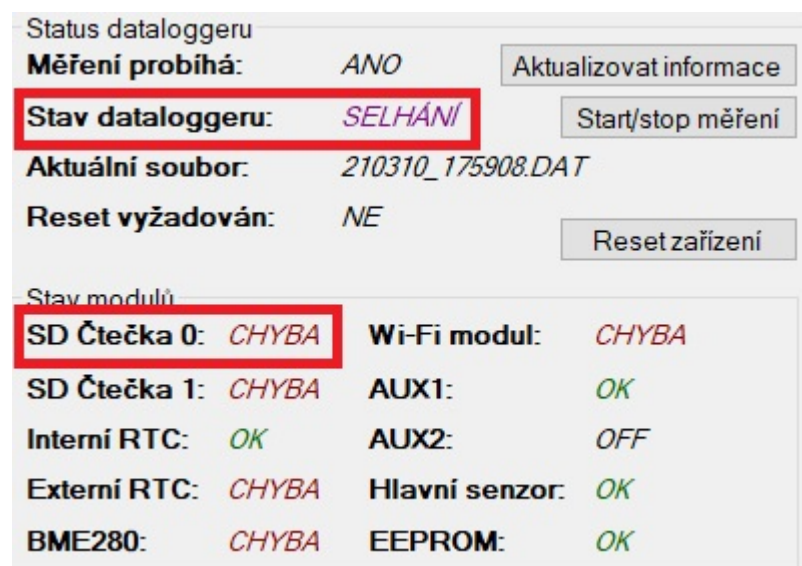
5.4.3 Kombinovaný test funkcionalit

V předešlých testech a experimentech byl kladen důraz vždy na určitou část, či funkcionalitu systému. Kombinovaný test ověřuje více funkcionalit tím způsobem, že se s dataloggerem pracuje takovým způsobem, jakým by se pracovalo

5. TESTOVÁNÍ SYSTÉMU



Obrázek 5.16: Status dataloggeru v obslužné aplikaci – Selhání Wi-Fi modulu, externí RTC a paměťové karty 1.



Obrázek 5.17: Status dataloggeru v obslužné aplikaci – Odebrání zbylé paměťové karty.

ve reálných podmínkách. Ke komunikaci s dataloggerem se využívá pouze obslužná počítačová aplikace. Průběh testu je následující:

1. Konfigurace dataloggeru a sítí
2. Měření po dobu několika hodin
3. Stažení naměřených dat

Během testů byl měřen i odběr proudu dataloggeru pomocí testeru¹⁵. Ukázalo se ale, že tester není dostatečně citlivý a naměřené hodnoty jsou chybné. Proto se měřil okamžitý odběr proudu při vypnutém a zapnutém Wi-Fi modulu pomocí multimetru. Celkově byly provedeny tři kombinované testy. První test byl s vypnutým Wi-Fi modulem, s dataloggerem se komunikovalo pomocí USB. Tento test měl za úkol otestovat sériový most v obslužné počítačové aplikaci. Druhý test měl nastaveno vypínání Wi-Fi modulu v závislosti na čase/dnu. Účelem druhého testu bylo otestování funkcí, které zajišťují zapínání/vypínání Wi-Fi modulu v závislosti na hodinách, či dnech. Poslední test měl zapnutý Wi-Fi modul po celou dobu měření. V posledním kombinovaném testu s permanentně zapnutým Wi-Fi modulem probíhalo ukládání aktuální dat na straně počítače s intervalem 5 minut. Poslední test měl otestovat UDP most a zjistit, zda správně funguje záznam dat na straně počítače.

Během kombinovaných testů byl k dataloggeru připojen simulátor, v prvním a třetím testu bylo jako zářič využito Cesium-137 s 2000 CPS. Ve druhém kombinovaném testu se využíval jako zářič Kobalt-60 s 2000 CPS. Dále bylo k dataloggeru připojeno pomocné měřicí zařízení na bázi Geiger-Müllerovy trubice. Během druhého a třetího testu bylo v blízkosti tohoto detektoru přiloženo uranové sklo. Interval zápisu na paměťovou kartu byl 15 minut, kontrola sítě probíhala každé dvě minuty, mezipaměť ve Wi-Fi modulu se aktualizovala každou minutu.

První kombinovaný test začínal konfigurací dataloggeru. Wi-Fi modul byl nastaven na vypnutý, typ startu bylo nastaveno na *čekání na žádost*. Měření tedy bylo nutné spustit manuálně pomocí aplikace a počítače připojeného k dataloggerem přes USB. Sítě nebylo nutné konfigurovat, protože Wi-Fi modul byl stejně vypnutý. Poté proběhlo uložení konfigurace do EEPROM a datalogger byl odpojen od napájení, aby se otestovalo, zda konfigurace zůstane uložena i po ztrátě napájení. Potom byl datalogger připojen k napájení a bylo manuálně spuštěno měření, počítač byl následně od dataloggeru odpojen. Měření probíhalo po dobu přibližně 270 minut. Po ukončení měření byla stažena data z měření a otevřena. Tímto krokem končil první kombinovaný test.

Podobně jako první kombinovaný test i druhý kombinovaný test začínal konfigurací dataloggeru. Wi-Fi modul byl nastavený na zapnutý a bylo nastaveno automatické vypínání Wi-Fi modulu v intervalu 13-15 hodin. Jako

¹⁵<https://www.ebay.com/itm/USB-Charger-Doctor-Voltage-Current-Meter-Detector-Voltmeter-Ammeter-Tester/114345260039>

typ startu se použil *start se zpožděním* 120 sekund. Kromě konfigurace dataloggeru proběhla i konfigurace sítě. Poté proběhlo uložení konfigurací do EEPROM a datalogger byl odpojen od napájení. Potom se datalogger opět připojil k napájení a byl poznamenán čas, kdy se připojilo napájení. Vyčkalo se 5 minut a počítač se k dataloggeru připojil přes Wi-Fi. Proběhla kontrola, zda měření probíhá a zdali se měření spustilo dvě minuty po napájení (startu) dataloggeru. Měření probíhalo celkově 310 minut, z toho 180 minut byl vypnut Wi-Fi modul. Druhý kombinovaný test byl ukočen stažením a otevřením naměřených dat.

Poslední kombinovaný test začínal stejně jako druhý kombinovaný test – nastavením dataloggeru a sítě. Jako typ startu se použil *okamžitý start*. Konfigurace byla uložena do EEPROM a podobně jako u předchozích testů byl datalogger na chvíli odpojen od napájení. Po startu datalogger se počítač připojil k dataloggeru přes Wi-Fi a zkontrolovalo se, zdali se měření spustilo automaticky. Poté se v aplikaci nastavil záznam dat s intervalem 5 minut. Měření probíhalo 320 minut. Během tohoto testu se počítač přibližně na hodinu a půl nečekaně odpojil od sítě, proto v tuto dobu nedocházelo ke sběru dat. Nechtěné odpojení od sítě umožnilo otestovat další funkce obslužné aplikace. Bylo nutné promazat záznamy, ve kterých nedocházelo k aktualizaci dat. Poté byly staženy soubory daného měření z obou paměťových karet, zkontrolovány, zda jsou konzistentní, a spojeny se záznamy z počítače.

Kombinované testy ověřily zbylé funkcionality, které datalogger a obslužná počítačová aplikace nabízí a nebyly důkladně otestovány předchozími testy. Během kombinovaných testů se měřila i spotřeba dataloggeru, ale ukázalo se, že použitý tester měří chybně. Proto byl alespoň změřen okamžitý odběr proudu pomocí multimetru. Měření proudu probíhalo s odpojenými měřicími zařízeními, takže byla měřena spotřeba samotného dataloggeru. Při měření proudu shořela 200 mA pojistka, která se nachází v multimetru. Proud byl proto měřen na vstupu multimetru, který není chráněn pojistkou ale umožňuje bezpečně měřit proud pouze po dobu 10 sekund. Shoření pojistky signalizuje, že po dobu běhu dataloggeru může být chvilkový odběr proudu vysoký. Při vypnutém Wi-Fi modulu se odběr proudu pohyboval mezi 40–60 mA. Při zapnutém Wi-Fi modulu byl odběr proudu 110–130 mA.

Největší spotřebu má Wi-Fi modul, který sám o sobě má proudový odběr průměrně 70 mA. Osvědčila se funkce vypínání Wi-Fi modulu v závislosti na hodinách/dnech, která výrazně šetří energii. Při vhodném použití může prodloužit délku provozu dataloggeru bez zásahu uživatele o několik stovek hodin. Pro dosažení optimální spotřeby by bylo potřeba odpájet všechny LED z modulů dataloggeru, které během realizace a testování sloužily pro ladící účely. V „ostrém“ běhu však postrádají smysl a jejich odpájení může snížit spotřebu o přibližně 10–20 mA. Během výběru kapacity powerbanky je nutné vzít v potaz i to, že do celkové spotřeby je potřeba započítat i všechny připojené měřicí zařízení (včetně pomocných měřicích zařízení). Každé měřicí zařízení může mít spotřebu až několik desítek mA.

Závěr

Na začátku práce byly stanoveny cíle, které se povedly splnit. Prvním cílem byla analýza možností měření či simulace ionizujícího záření a požadavků na datalogger. Analýza detektorů ukázala, že k realizaci detektoru jsou potřeba dražší a méně obvyklé komponenty. Z tohoto důvodu byla zvolena možnost simulace detektoru. V rámci analýzy byl nalezen komerční emulátor detektoru od firmy CEAN, který však byl pro testování nedostupný. V rámci testování byl implementován simulátor na vývojové desce Nucleo-144 s mikrokontrolérem STM32, který obsahuje část funkcionalit komerčního emulátoru. Analýza také ukázala, že datalogger potřebuje pro čtení hodnot z detektoru rychlý A/D převodník.

Dalším cílem byl návrh a realizace dataloggeru. V rámci návrhu byly zvoleny jednotlivé komponenty a konkrétní mikrokontrolér STM32, kromě toho byly navrženy i protokoly pro komunikaci prvků v rámci celého systému. Výsledný datalogger byl realizován takovým způsobem, aby jeho provoz byl spolehlivý. Datalogger obsahuje dva módy měření analogového signálu z detektoru ionizujícího záření. Prvním módem je *pokračovaného měření*, kdy je signál vzorkován s frekvencí 48 kHz a jsou hledána maxima. Druhým módem je měření se spouští, při kterém se konverze A/D převodníku spouští pomocí externí spouště. Kromě toho je k dataloggeru možné připojit i dvě tzv. „pomocná měřicí zařízení“, které neměří energii kvant záření, ale detekují přítomnost ionizujícího záření. K úspěšné realizaci dataloggeru bylo nutné kromě samotného firmwaru pro STM32, potřeba realizovat i firmware pro Wi-Fi modul ESP-01. Součástí realizace dataloggeru byl i návrh a realizace desky plošných spojů. Deska plošných spojů byla osazena komponenty v domácích podmínkách pomocí levné pájecí stanice.

Třetím cílem byl návrh a realizace obslužné počítačové aplikace pro Windows 10. V rámci návrhu byla zvolena knihovna Windows Forms, která umožňuje tvořit aplikace s grafickým rozhraním. Jako programovací jazyk obslužné aplikace se používal C#. Pomocí aplikace lze konfigurovat datalogger, sledovat aktuální hodnoty, stahovat naměřená data a data graficky zobrazit. Počítač,

na kterém obslužná aplikace běží, komunikuje s dataloggerem buď bezdrátově pomocí Wi-Fi, či fyzicky pomocí USB.

Posledním cílem bylo testování dataloggeru a obslužné aplikace. Jak již bylo zmíněno, součástí testování byla i realizace simulátoru detektoru ionizujícího záření. Testování dataloggeru probíhalo za využití tohoto simulátoru. Jako referenční energetické spektrum, které bylo simulátorem generováno se využívalo spektrum Kobaltu-60, či Cesia-137, které bylo změřeno scintilačním detektorem. Kromě simulátoru se k testování využíval detektor na bázi Geiger-Müllerovy trubice a uranové sklo. Obslužná aplikace byla testována pomocí testu ostatních funkcionalit, které se zaměřovaly na důkladné testování jednotlivých funkcionalit systému. Součástí testování byl i test, jakým způsobem datalogger reaguje na selhávající prvky.

Výsledkem diplomové práce je datalogger včetně zdrojového kódu firmwaru, schématu a návrhu desky plošných spojů. Datalogger lze do budoucna vylepšit tím, že se optimalizuje firmware. Většina firmwaru je napsaná za využití HAL knihoven, které zvyšují přenositelnost mezi mikrokontroléry STM32, avšak nejsou optimální co se týče rychlosti. Dalším možným vylepšením je akcelarovat zpracování analogového signálu pomocí FPGA. Rychlost vzorkování A/D převodníku je v dataloggeru značně omezena, protože při vyšších frekvencích vzorkování by nebylo možné data zpracovat. Akcelerace pomocí FPGA by mohla zlepšit výslednou vzorkovací frekvenci, tudíž by se zvýšila přesnost měření. Datalogger lze vylepšit i tím, že se pro připojení detektoru bude využívat např. koaxiální kabel, který je stíněný proti elektromagnetickému záření.

Další výsledek diplomové práce je obslužná počítačová aplikace pro Windows 10. Aplikaci lze vylepšit tím způsobem, že se do ní přidají další funkce, jako např. efektivnější filtr šumu, lepší vizualizace dat atd. Aplikace není responzivní a její okna mají fixní velikost, další možné vylepšení je aplikaci udělat responzivní na různá rozlišení monitorů. Dalším možným vylepšením aplikace je přenos aplikace na jiný operační systém, či vytvoření mobilní aplikace.

Výsledkem diplomové práce je kromě dataloggeru a obslužné počítačové aplikace i jednoduchý simulátor ionizujícího záření. Simulátor je omezen pouze na pulzy gaussovského tvaru a nepodporuje kumulativní mrtvou dobu, což je omezující. Simulátor lze v budoucnu vylepšit podporou dalších tvarů pulzů, či implementace kumulativní mrtvé doby detektoru. Dalším vylepšením je realizace simulátoru na FPGA, jako je to v případě komerčního emulátoru, ze kterého realizovaný simulátor vycházel. Realizace plnohodnotného simulátoru by vydalo za samostatné téma diplomové práce.

Literatura

- [1] Ullmann, V.: Ionizující záření. [online], [cit. 2021-02-20]. Dostupné z: <https://astronuklfyzika.cz/JadRadFyzika6.htm>
- [2] Ionizace – WikiSkripta. [online], Poslední aktualizace 2018-06-15, [cit. 2021-02-20]. Dostupné z: <https://www.wikiskripta.eu/w/Ionizace>
- [3] Částice alfa – Wikipedie. [online], Poslední aktualizace 2020-10-15, [cit. 2021-02-21]. Dostupné z: https://cs.wikipedia.org/wiki/%C4%8C%C3%A1stice_alfa
- [4] Záření alfa – WikiSkripta. [online], Poslední aktualizace 2020-08-02, [cit. 2021-02-21]. Dostupné z: https://www.wikiskripta.eu/w/Z%C3%A1%C5%99en%C3%AD_alfa
- [5] Ullmann, V.: Radioaktivita, radionuklidy, záření. [online], [cit. 2021-02-20]. Dostupné z: <https://astronuklfyzika.cz/JadRadFyzika2.htm>
- [6] Záření beta – Wikipedie. [online], Poslední aktualizace 2020-10-08, [cit. 2021-02-21]. Dostupné z: https://cs.wikipedia.org/wiki/Z%C3%A1%C5%99en%C3%AD_beta
- [7] Záření beta – WikiSkripta. [online], Poslední aktualizace 2018-02-26, [cit. 2021-02-21]. Dostupné z: https://www.wikiskripta.eu/w/Z%C3%A1%C5%99en%C3%AD_beta
- [8] Kuchař, L.: *Zařízení pro měření radiačního záření*. Bakalářská práce, ČVUT. Fakulta informačních technologií, 2018.
- [9] Gamma radiation | ARPANSA. [online], [cit. 2021-02-21]. Dostupné z: <https://www.arpansa.gov.au/understanding-radiation/what-is-radiation/ionising-radiation/gamma-radiation>

- [10] Záření gama – WikiSkripta. [online], Poslední aktualizace 2018-02-26, [cit. 2021-02-21]. Dostupné z: https://www.wikiskripta.eu/w/Z%C3%A1%C5%99en%C3%AD_gama
- [11] Záření beta – Wikipedie. [online], Poslední aktualizace 2020-02-14, [cit. 2021-02-21]. Dostupné z: https://cs.wikipedia.org/wiki/Z%C3%A1%C5%99en%C3%AD_gama
- [12] What Is CPM in Radiation? | SOEKS USA. [online], Publikováno 2016-04-06, [cit. 2021-02-22]. Dostupné z: <https://soeks-usa.com/blogs/radiation/blog-what-is-cpm-in-radiation>
- [13] Elektronvolt – Wikipedie. [online], Poslední aktualizace 2020-06-08, [cit. 2021-02-22]. Dostupné z: <https://cs.wikipedia.org/wiki/Elektronvolt>
- [14] Ullmann, V.: Detekce a spektrometrie ionizujícího záření. [online], [cit. 2021-02-22]. Dostupné z: <https://www.astronuklfyzika.cz/DetekceSpektrometrie.htm>
- [15] PMT Pulse Processing | PhysicsOpenLab. [online], Publikováno 2016-04-21, [cit. 2021-02-22]. Dostupné z: <http://physicsopenlab.org/2016/04/21/pmt-pulse-processing/>
- [16] Gamma Spectrometry | theremino. [online], [cit. 2021-02-22]. Dostupné z: <https://www.theremino.com/en/blog/gamma-spectrometry>
- [17] USB Gamma Spectrometer. [online], [cit. 2021-03-22]. Dostupné z: <https://www.gammaspectacular.com/blue/gamma-spectroscopy/sound-card-spectrometry-drivers/gs-usb-pro>
- [18] Quantum™ MCA2500R SYSTEM MANUAL. [online], [cit. 2021-03-22]. Dostupné z: <https://www.berkeleyneutronics.com/sites/default/files/products/resources/mca2500r-manual.pdf>
- [19] DIY PIC18 MCA Module for Gamma Spectroscopy 1024 Channels 10bit ADC w/SD logs RH Electronics. [online], [cit. 2021-03-22]. Dostupné z: <https://rhelectronics.net/store/diy-pic18-mca-kit-for-gamma-spectroscopy.html>
- [20] Brown, G.; iZotope Content Team: Digital Audio Basics: Sample Rate and Bit Depth. [online], Poslední aktualizace 2019-07-15, [cit. 2020-12-22]. Dostupné z: <https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html>
- [21] Alpha-particle spectroscopy - Wikipedia. [online], Poslední aktualizace 2018-09-13, [cit. 2021-02-23]. Dostupné z: https://en.wikipedia.org/wiki/Alpha-particle_spectroscopy

-
- [22] Caesium-137 - Wikipedia. [online], Poslední aktualizace 2021-02-16, [cit. 2021-02-23]. Dostupné z: <https://en.wikipedia.org/wiki/Caesium-137>
- [23] The Analysis Pulse-Height Spectrometry. [online], [cit. 2021-02-23]. Dostupné z: <http://www.people.vcu.edu/~mhcrosthwait/clrs322/Pulseanalysis.htm>
- [24] Cobalt-60 - Wikipedia. [online], Poslední aktualizace 2021-02-06, [cit. 2021-02-23]. Dostupné z: <https://en.wikipedia.org/wiki/Cobalt-60>
- [25] Co 60 Gamma spectrum. [online], [cit. 2021-02-23]. Dostupné z: https://www.gammaspectacular.com/blue/gamma_spectra/Co-60-gamma-spectrum
- [26] CAEN Detector Emulator. [online], [cit. 2021-02-24]. Dostupné z: <http://www.detectoremulator.com/emulationprinciple.html>
- [27] CAEN Detector Emulator. [online], [cit. 2021-02-24]. Dostupné z: <http://www.detectoremulator.com/comparison.html>
- [28] Noviello, C.: *Mastering STM32*. Květen 2018, [cit. 2021-03-27]. Dostupné z: <https://leanpub.com/mastering-stm321>
- [29] Delta-sigma modulation - Wikipedia. [online], Poslední aktualizace 2020-12-07, [cit. 2020-12-22]. Dostupné z: https://en.wikipedia.org/wiki/Delta-sigma_modulation
- [30] Development boards | STM32-base project. [online], [cit. 2020-12-22]. Dostupné z: <https://stm32-base.org/boards/>
- [31] STM32F103C8T6 - Blue Pill | STM32-base project. [online], [cit. 2020-12-22]. Dostupné z: <https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill>
- [32] STM32F401CCU6 - WeAct Black Pill V1.2 | STM32-base project. [online], [cit. 2020-12-22]. Dostupné z: <https://stm32-base.org/boards/STM32F401CCU6-WeAct-Black-Pill-V1.2>
- [33] STM32F103C8 - Mainstream Performance line, Arm Cortex-M3 MCU with 64 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN - STMicroelectronics. [online], [cit. 2020-12-28]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>
- [34] STMicroelectronics: *RM0008 Reference manual*. [cit. 2020-12-27]. Dostupné z: https://www.st.com/resource/en/reference_manual/cd00171190-stm32f101xx-stm32f102xx-stm32f103xx-

stm32f105xx-and-stm32f107xx-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

- [35] Strange unknown STM32F103 clone: CS32F103. Anyone has info or tested it? - Sonsivri. [online], [cit. 2020-12-27]. Dostupné z: <http://www.sonsivri.to/forum/index.php?topic=66427.0>
- [36] STMicroelectronics: *RM0368 Reference manual*. [cit. 2020-12-27]. Dostupné z: https://www.st.com/resource/en/reference_manual/dm00096844-stm32f401xb-c-and-stm32f401xd-e-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf
- [37] STM32F401CC - High-performance access line, Arm Cortex-M4 core with DSP and FPU, 256 Kbytes of Flash memory, 84 MHz CPU, ART Accelerator - STMicroelectronics. [online], [cit. 2020-12-29]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32f401cc.html>
- [38] ESP8266 - Wikipedia. [online], Poslední aktualizace 2020-12-25, [cit. 2020-12-29]. Dostupné z: <https://en.wikipedia.org/wiki/ESP8266>
- [39] esp8266/Arduino: ESP8266 core for Arduino. [online], [cit. 2020-12-29]. Dostupné z: <https://github.com/esp8266/Arduino>
- [40] The best battery for ESP8266 microcontrollers - DIY10T. [online], [cit. 2020-12-29]. Dostupné z: <https://diy10t.com/best-battery-for-esp8266/>
- [41] RTC hodiny reálného času DS3231 + paměť EEPROM AT24C32 - HW Kitchen. [online], [cit. 2020-12-27]. Dostupné z: <https://www.hwkitchen.cz/rtc-hodiny-realneho-casu-ds3231-pamet-eprom-at24c32/>
- [42] EEPROM on Flash - Espruino. [online], [cit. 2020-12-25]. Dostupné z: <https://www.espruino.com/FlashEEPROM>
- [43] STMicroelectronics: *AN3969*. [cit. 2020-12-25]. Dostupné z: https://www.st.com/resource/en/application_note/dm00036065-eprom-emulation-in-stm32f40x-stm32f41x-microcontrollers-stmicroelectronics.pdf
- [44] STMicroelectronics: *M24C16, M24C08, M24C04, M24C02, M24C01*. [cit. 2020-12-27]. Dostupné z: <https://www.svetsoucastek.cz/media/6626/datasheet-414-064.pdf?&key=ZGpmIyQwNUZfNjYyNg==>
- [45] Lety Fík - Ústav jaderné fyziky AV ČR. [online], [cit. 2020-12-25]. Dostupné z: <http://www.ujf.cas.cz/cs/oddeleni/oddeleni-dozimetrie-zareni/lety-fik/>

-
- [46] Sensortec, B.: BME280 driver. [online], [cit. 2020-12-25]. Dostupné z: https://github.com/BoschSensortec/BME280_driver
- [47] Bosch Sensortec: *BME280 datasheet*. [cit. 2020-12-25]. Dostupné z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf>
- [48] Arduino Easy Weather Station With BME280 Sensor : 6 Steps (with Pictures) - Instructables. [online], [cit. 2021-02-10]. Dostupné z: <https://www.instructables.com/Arduino-Easy-Weather-Station-With-BME280-Sensor/>
- [49] Cyclic redundancy check - Wikipedia. [online], Poslední aktualizace 2020-12-27, [cit. 2020-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Cyclic_redundancy_check
- [50] User Datagram Protocol - Wikipedia. [online], Poslední aktualizace 2020-12-10, [cit. 2021-01-01]. Dostupné z: https://en.wikipedia.org/wiki/User_Datagram_Protocol
- [51] *Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino*. [cit. 2021-01-02]. Dostupné z: <http://datalogger.pbworks.com/w/file/attach/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>
- [52] Atmel Corporation: *AT24C32, AT24C64*. [cit. 2021-01-02]. Dostupné z: https://www.hwkitchen.cz/user/related_files/pamet-eprom-at24c32.pdf
- [53] Maxim Integrated Products: *DS3231*. [cit. 2021-01-02]. Dostupné z: https://www.hwkitchen.cz/user/related_files/rtc-hodiny-realneho-casu-ds3231.pdf
- [54] Connfly electronic CO. Ltd.: *Connfly*. [cit. 2021-01-02]. Dostupné z: <https://www.svetsoucastek.cz/media/6690/datasheet-832-120.pdf?&key=ZGpmIyQwNUZfNjY5MA==>
- [55] STMicroelectronics: *LD1117*. [cit. 2021-01-02]. Dostupné z: <https://www.st.com/resource/en/datasheet/ld1117.pdf>
- [56] Choose your Windows app platform - Windows applications | Microsoft Docs. [online], Poslední aktualizace 2021-02-03, [cit. 2021-02-15]. Dostupné z: <https://docs.microsoft.com/en-us/windows/apps/desktop/choose-your-platform>
- [57] FatFs - Generic FAT Filesystem Module. [online], [cit. 2021-01-05]. Dostupné z: http://elm-chan.org/fsw/ff/00index_e.html

- [58] eziya: STM32 SPI SDCARD. [online], [cit. 2021-01-05]. Dostupné z: https://github.com/eziya/STM32_SPI_SDCARD
- [59] Cortex-M4 Technical Reference Manual. [online], [cit. 2021-01-07]. Dostupné z: <https://developer.arm.com/documentation/ddi0439/b/Programmers-Model/Instruction-set-summary/Cortex-M4-instructions?lang=en>
- [60] Silicon Laboratories: *CP2102/9*. [cit. 2021-02-13]. Dostupné z: <https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf>
- [61] Shahryiar, S.: STM32 Timers | Embedded Lab. [online], Publikováno 2015-12-01, [cit. 2021-01-10]. Dostupné z: <http://embedded-lab.com/blog/stm32-timers/>
- [62] Sunshine's Homepage - Understanding CRC. [online], Poslední úprava 2019-07-02, [cit. 2021-01-11]. Dostupné z: http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html
- [63] Cyclic redundancy check - Wikipedia. [online], Poslední úprava 2020-12-27, [cit. 2021-01-11]. Dostupné z: https://en.wikipedia.org/wiki/Cyclic_redundancy_check#CRC-32_algorithm
- [64] ESP8266 Pinout, Pin Configuration, Features, Example Circuit & Datasheet. [online], Zveřejněno 2018-03-20, [cit. 2021-02-14]. Dostupné z: <https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet>
- [65] Selektivní opakování – Wikipedie. [online], Poslední aktualizace 2020-06-05, [cit. 2021-02-14]. Dostupné z: https://cs.wikipedia.org/wiki/Selektivn%C3%AD_opakov%C3%A1n%C3%AD
- [66] c# - Many short-lived threads approach? - Stack Overflow. [online], [cit. 2021-02-17]. Dostupné z: <https://stackoverflow.com/questions/33380914/many-short-lived-threads-approach>
- [67] Programming the Thread Pool in the .NET Framework | Microsoft Docs. [online], Poslední aktualizace 2007-04-30, [cit. 2021-02-17]. Dostupné z: [https://docs.microsoft.com/en-us/previous-versions/dotnet/articles/ms973903\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/dotnet/articles/ms973903(v=msdn.10)?redirectedfrom=MSDN)
- [68] STMicroelectronics: *RM0432*. [cit. 2021-03-02]. Dostupné z: https://www.st.com/resource/en/reference_manual/dm00310109-stm3214-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
- [69] Gaussian filter - Wikipedia. [online], Poslední aktualizace 2021-01-21, [cit. 2021-03-02]. Dostupné z: https://en.wikipedia.org/wiki/Gaussian_filter

-
- [70] CAEN SpA: *User Manual UM5312*. [cit. 2021-03-04]. Dostupné z: http://www.nuclearinstruments.eu/cdn/UM5312_Fast_Digital_Detector_Emulator_r0w.pdf
- [71] Siegel, P.: Cal Poly Pomona Physics Department Gamma Detector Data Files. [online], [cit. 2021-03-07]. Dostupné z: <https://www.cpp.edu/~pbsiegel/nuclear.html>
- [72] Summing Amplifier is an Op-amp Voltage Adder. [online], [cit. 2021-03-08]. Dostupné z: https://www.electronics-tutorials.ws/opamp/opamp_4.html
- [73] STMicroelectronics: *STM32F401xB STM32F401xC*. [cit. 2021-03-08]. Dostupné z: <https://www.st.com/resource/en/datasheet/stm32f401cb.pdf>

Seznam použitých zkratk

- ADC** Analog-to-digital converter (Analogově-digitální převodník)
- A/D** Analog-to-digital (Analogově-digitální)
- D/A** Digital-to-analog (Digitálně analogový)
- DAC** Digital-to-analog converter (Digitálně-analogový převodník)
- FWHM** Full width at half maximum (šířka v polovině maxima)
- HAL** Hardware abstraction layer
- HSE** High speed external
- HSI** High speed internal
- LSE** Low speed external
- LSI** Low speed internal
- RAM** Random access memory
- ROM** Read-only memory
- RTC** Real time clock (Hodiny reálného času)
- UDP** User datagram protocol
- USB** Universal serial bus

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
exe	adresář se spustitelnou formou implementace
├─ DataloggerApp	obslužná počítačová aplikace
├─ SimulatorApp	aplikace pro obsluhu simulátoru
data	data z měření kombinovaných testů
HW	schéma a návrh desky dataloggeru
src		
├─ Datalogger		
│ └─ STM32 Firmware	zdrojové kódy firmwaru pro řídicí jednotku
│ └─ ESP01 Firmware	zdrojové kódy firmwaru pro Wi-Fi modul
├─ DataloggerApp	zdrojové kódy obslužné počítačové aplikace
├─ Simulator	zdrojové kódy simulátoru
│ └─ PulseGenerator	zdrojové kódy pro firmware generující pulzy gaussovského tvaru uživatelem definovaných vlastností
│ └─ SpectrumGenerator	...	zdrojové kódy pro firmware generující pulzy dle zadaného spektra
│ └─ SimulatorApp	zdrojové kódy aplikace pro obsluhu simulátoru
└─ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
└─ thesis.pdf	text práce ve formátu PDF

Nástroje použité pro vývoj a testování systému

- Pro testování obslužné počítačové aplikace a konfigurace simulátoru byl využit počítač střední třídy, jehož parametry udává Tabulka C.1. Pro bezdrátovou komunikaci se využívá externí Wi-Fi karta TP-Link Archer T4U AC1200¹⁶.
- Fyzické chování simulátoru detektoru ionizujícího záření byla využita vývojová deska Nucleo-144 L4R5ZI s mikrokontrolérem STM32L4R5.
- Jako simulované zářiče pro simulátor byly využity Cesium-137 a Kobalt-60, jejichž spektrum bylo staženo z [71].
- Použitý osciloskop byl Hantek 2D72¹⁷.
- Použitý multimetr byl UNI-T UT39B.
- Pro měření proudu se využíval USB Charger Doctor Voltage Current Meter Detector Voltmeter Ammeter Tester¹⁸.
- Jako pomocné měřicí zařízení byl využit detektor na bázi Geiger-Müllerovy trubice ¹⁹.
- Jako zářič pro pomocné měřicí zařízení se využívalo uranové sklo s ekvivalentním dávkovým příkonem 4,5 $\mu\text{Sv/h}$.

¹⁶<https://www.zbozi.cz/vyrobek/tp-link-archer-t4u-ac1200-wifi-dual-b-usb-adapter/>

¹⁷<https://www.aliexpress.com/item/32968939126.html>

¹⁸<https://www.ebay.com/itm/USB-Charger-Doctor-Voltage-Current-Meter-Detector-Voltmeter-Ammeter-Tester/114345260039>

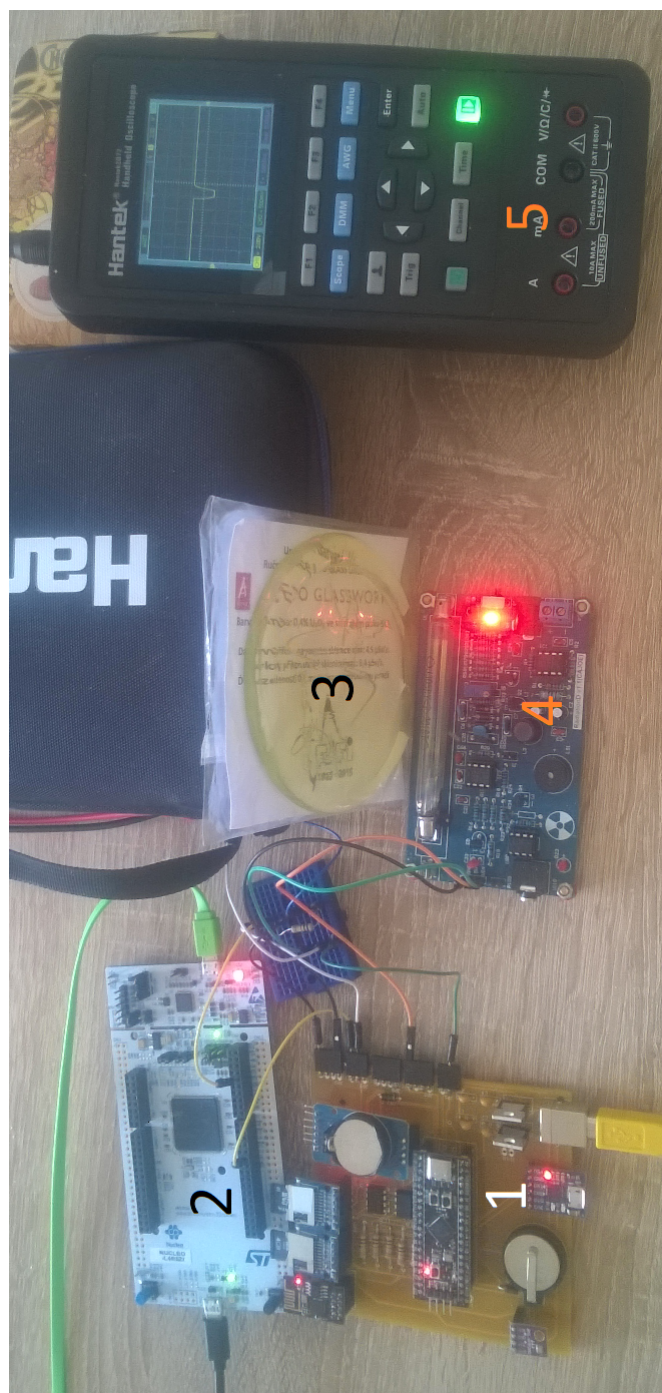
¹⁹<https://www.ebay.com/itm/Assembled-Geiger-Counter-DIY-Kit-Nuclear-Radiation-Detector-Gamma-Ray-Board/363238244426>

Tabulka C.1: Parametry testovací sestavy

CPU:	AMD Ryzen 5 1600; 6 jader / 12 vláken; frekvence jádra: 3,2 GHz
GPU:	Geforce GTX 1080 Sea Hawk X; frekvence jádra: 1822 MHz; 8192 MB GDDR5X 10010 MHz
RAM:	G.SKILL 16GB KIT DDR4 3200MHz CL14 Flare X for AMD
Operační systém:	Windows 10 Pro 64-bit

- Jako pájecí stanice byla použita pájecí stanice Zhongdi ZD-99²⁰.

²⁰<https://www.svetsoucastek.cz/pajeci-stanice-zhongdi-zd-99-p55397/>



Obrázek C.1: Ukázka prostředí, kde byl systém testován. 1 – Datalogger; 2 – Vývojová deska simulující detektor ionizujícího záření; 3 – Uranové sklo; 4 – Pomocné měřicí zařízení; 5 – Osciloskop připojený na výstup pomocného měřicího zařízení.

Uživatelský manuál

D.1 Datalogger

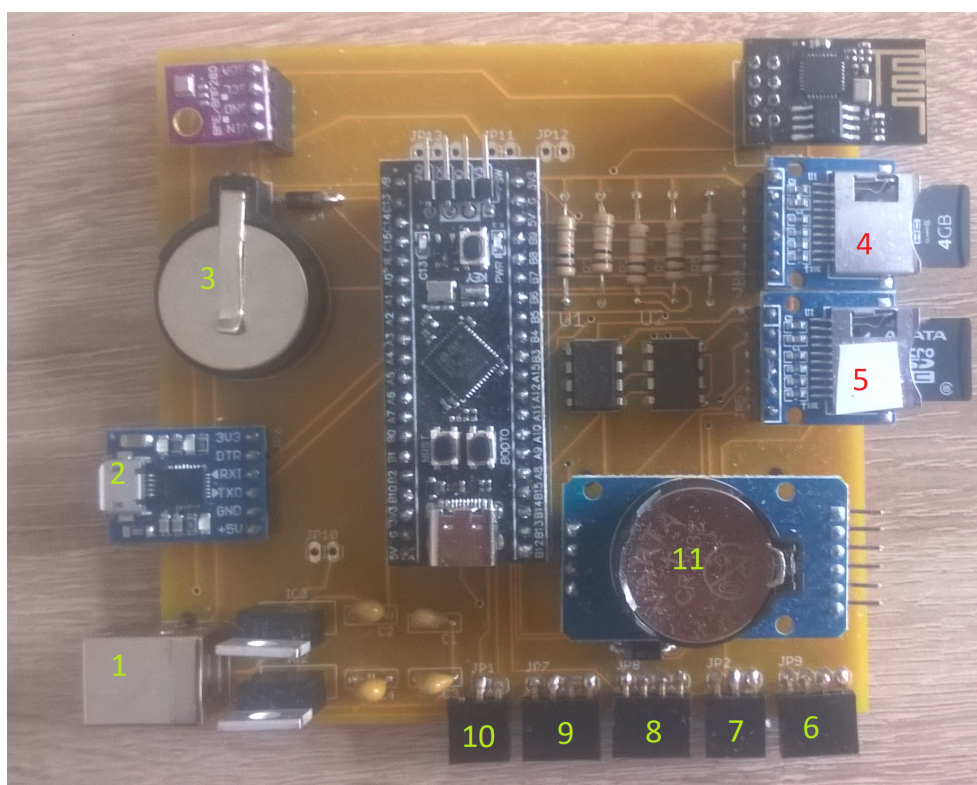
Prvky, s kterými uživatel přímo interaguje, popisuje Obrázek D.1. Datalogger je napájen z USB (na Obrázku D.1 jako 1), jako zdroj může být použit počítač, či powerbanka s vhodnou kapacitou. S dataloggerem lze komunikovat buď pomocí Wi-Fi, či pomocí USB kabelu, který se zapojuje do portu označený na Obrázku D.1 jako 2.

Datalogger obsahuje dvě jednotky hodin reálného času, které ke správné funkci potřebují knoflíkovou baterii (CR2032). Místa, kam se knoflíkové baterie vkládají jsou na Obrázku D.1 označeny jako 3 a 11. Datalogger také obsahuje dvě čtečky paměťových karet (microSD), první čtečka je označena jako čtečka paměťových karet 0 (na Obrázku D.1 jako 4), druhá jako čtečka paměťových karet 1 (na Obrázku D.1 jako 5).

Datalogger obsahuje napájecí piny, kterými lze napájet externí periferie. Na desce se nachází 4 piny, které jsou připojeny k zemi (na Obrázku D.1 jako 6), 4 piny poskytující napětí 5 V (na Obrázku D.1 jako 9) a 4 piny poskytující napětí 3,3 V (na Obrázku D.1 jako 8).

K připojení analogové měřicí jednotky ionizujícího záření se využívá trojice pinů, která je na Obrázku D.1 označena číslem 7. Pin, který je nejbližší napájecímu USB (z trojice pinů je umístěn vlevo) indikuje, kdy je ADC vytíženo (signál *A/D vytížení*, v textu je tento pin oznčen jako *Konverze probíhá*). Prostřední pin z této trojice je pin, ke kterému se připojuje analogový signál z měřicí jednotky. Poslední pin z trojice je pin pro spoušť.

Kromě analogové měřicí jednotky je možné k dataloggeru připojit i dvě nezávislé pomocné měřicí jednotky (v obslužné aplikaci také pod označením přídavné měřicí jednotky). Tyto jednotky se připojují do trojice pinů, která je na Obrázku D.1 označena číslem 10. Pin z trojice, který je nejbližší napájecímu USB, není připojen k mikrokontroléru a nemá žádný význam. K prostřednímu pinu se připojuje pomocné měřicí zařízení 1 (*AUX1*), ke zbylému pinu se při-



Obrázek D.1: Zapojení dataloggeru. 1 – napájecí USB; 2 – Komunikační USB; 3 – Knoflíková baterie pro napájení interního RTC; 4 – Čtečka paměťových karet 0; 5 – Čtečka paměťových karet 1; 6 – Zem (Ground); 7 – 3x pin pro připojení měřicí zařízení, vpravo spoušť, uprostřed analogový signál, vlevo A/D vytížen signál; 8 – Výstupní napájení 3,3 V; 9 – Výstupní napájení 5 V; 10 – 3x pin pro připojení pomocných měřicích zařízení, vpravo AUX2, uprostřed AUX1, vlevo nepřipojeno; 11 – Knoflíková baterie pro napájení externího RTC

pojuje pomocné měřicí zařízení 2 (*AUX2*).

D.2 Obslužná počítačová aplikace

S dataloggerem se komunikuje přes obslužnou počítačovou aplikaci. K dataloggeru se počítač připojuje buď přes Wi-Fi, nebo přes USB pomocí virtuální sériové linky. Obslužná aplikace obsahuje více oken. Po spuštění aplikace se objeví okno, které zobrazuje Obrázek D.2. Hlavní okno se skládá ze čtyř hlavních částí.

První část je horní lišta, která je na Obrázku D.2 ve žlutém a oranžovém rámečku. Část lišty, která je ve žlutém rámečku obsahuje tlačítka pro ote-

vření správce dat (další okno aplikace), a uložení konfigurace do EEPROM. Část lišty, která je v oranžovém rámečku obsahuje informace o připojení k dataloggeru a zdali probíhá komunikace.

V červeném rámečku se nachází panel, kde se zobrazují aktuální data, manipuluje se soubory a konfiguruje datalogger a síť. V modrém rámečku jsou informace o dataloggeru. Nachází se zde stav jednotlivých modulů, status dataloggeru, zdali je vyžadován reset a informace o aktuálním měření. Zelený rámeček obsahuje nastavení a informace o připojení.

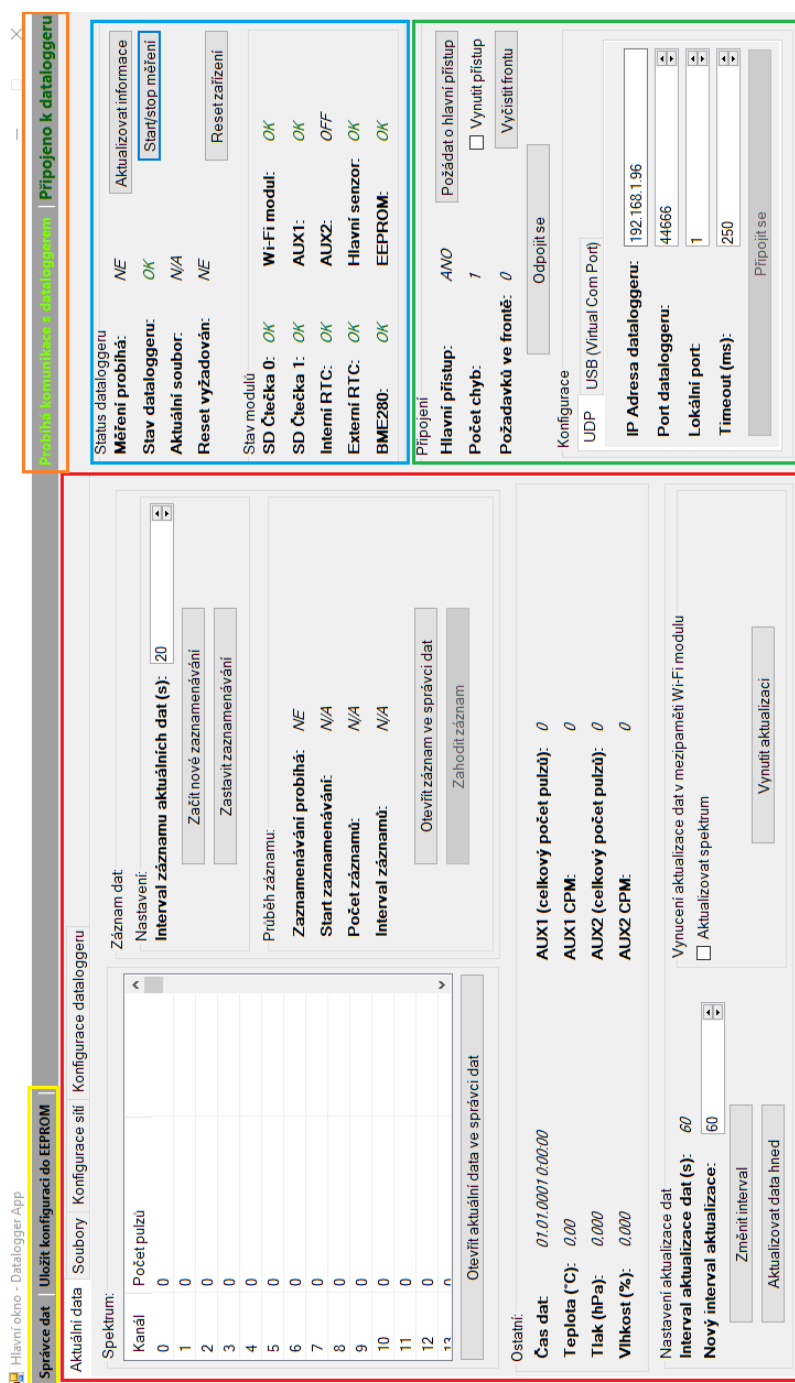
D.2.1 Navázání spojení s dataloggerem

Před jakoukoliv prací s dataloggerem je nutné nejprve navázat spojení. Spojení je možné navázat buď pomocí Wi-Fi a protokolu UDP, nebo USB. Pro připojení přes Wi-Fi je postup následující:

1. Počítač připojit ke stejné síti, ve které se nachází datalogger.
2. V panelu *Připojení* (na Obrázku D.2 jako zelený rámeček) v podpanelu *Konfigurace* vybrat záložku *UDP*.
3. Nastavit IP adresu dataloggeru, port dataloggeru, lokální port a timeout v milisekundách. Aby bylo možné navázat spojení, je potřeba, aby lokální port nebyl obsazen jinou aplikací pro UDP komunikaci.
4. Kliknout na tlačítko *Připojit se*.
5. Pokud je připojení úspěšné, v horní liště (na Obrázku D.2 jako oranžový rámeček) je *Připojeno k dataloggeru*. Pokud je spojení neúspěšné, v tomto rámečku je *Fatální chyba připojení*. Fatální chyba připojení znamená, že je nutné se odpojit pomocí tlačítka *Odpojit se*, opravit údaje a znova se připojit.
6. O hlavní přístup se žádá v panelu *Připojení* kliknutím na tlačítko *Požádat o hlavní přístup*. Skončí-li žádost o hlavní přístup neúspěchem, je možné vynutit hlavní přístup pomocí zaškrtnutí políčka *Vynutit přístup*.

Připojení přes USB má vždy status hlavního přístupu. Pro připojení přes USB je postup následující:

1. Datalogger připojit k počítači pomocí USB kabelu k portu, který je na Obrázku D.1 pod číslem 2.
2. V panelu *Připojení* (na Obrázku D.2 jako zelený rámeček) v podpanelu *Konfigurace* vybrat záložku *USB (Virtual Com Port)*.
3. Ze seznamu vybrat sériový port dataloggeru. Pokud v seznamu port není, zkontrolovat, zdali je datalogger korektně připojen, a aktualizovat porty pomocí tlačítka *Aktualizovat dostupné sériové porty*.



Obrázek D.2: Hlavní okno aplikace. Zelený rámeček obsahuje konfiguraci a status připojení. Modrý rámeček status a ovládací prvky dataloggeru. Oranžový rámeček zobrazuje stav připojení k dataloggeru a zdali právě probíhá komunikace s dataloggerem. Žlutý rámeček obsahuje prvky pro otevření správce dat a uložení konfigurace do EEPROM. Červený rámeček je hlavním ovládacím prvkem tohoto okna. Obsahuje 4 záložky.



Obrázek D.3: Stav dataloggeru a jednotlivých modulů. V červeném rámečku jsou ty ovládací prvky, které vyžadují hlavní přístup.

4. Kliknout na tlačítko *Připojit se*.
5. Pokud je připojení úspěšné, v horní liště (na Obrázku D.2 jako oranžový rámeček) je *Připojeno k dataloggeru*. Pokud je spojení neúspěšné, v tomto rámečku je *Fatální chyba připojení*. Fatální chyba připojení znamená, že je nutné se odpojit pomocí tlačítka *Odpojit se*, opravit údaje a znova se připojit.

D.2.2 Čtení stavu dataloggeru a jednotlivých modulů

Ke čtení stavu modulů je potřeba úspěšně navázané spojení. Ke čtení stavu dataloggeru je potřeba mít hlavní přístup. Informace o jednotlivých modulech a stavu dataloggeru je k nalezení v panelu *Status dataloggeru*, který je na Obrázku D.2 v modrém rámečku. Detail panelu ukazuje Obrázek D.3. Pokud není informace k dispozici, je u položky hodnota *N/A*. Kromě samotných informací o dataloggeru a jednotlivých modulů je možné v tomto panelu začít či zastavovat měření (tlačítko *Start/stop měření*), softwarově resetovat datalogger (tlačítko *Reset zařízení*) či aktualizovat informace o dataloggeru a jednotlivých modulů (tlačítko *Aktualizovat informace*).

D.2.3 Spuštění/zastavení měření

Pokud není datalogger nastaven v módu *okamžité měření*, či *měření se zpožděním*, je potřeba měření spustit manuálně. Ke spuštění, či zastavení měření musí mít aplikace hlavní přístup k dataloggeru. Spouštění/zastavení měření probíhá v panelu *Status dataloggeru* (na Obrázku D.2 v modrém rámečku) kliknutím na tlačítko *Start/stop měření*.

Spuštění měření v módu měření na žádost (či měření se zpožděním) se dělá stiskem tlačítka *Start/stop měření*. Aby bylo možné spustit měření, nesmí být vyžadován reset, či datalogger nesmí být ve stavu selhání. Po stisku tlačítka je potřeba počkat přibližně 5 sekund. O úspěchu spuštění měření je uživatel informován dialogovým oknem. Pokud není spuštění úspěšné, je doporučeno nejprve aktualizovat status dataloggeru pomocí tlačítka *Aktualizovat informace* a tím zkontrolovat, zda se měření nespustilo dodatečně, či se nevyskytla chyba znemožňující start měření. Pokud ani toto neodhalí důvod nespustění měření, je doporučeno start měření pomocí stisku tlačítka opakovat.

Pro zastavení měření se používá tlačítko *Start/stop měření*. Pokud je měření úspěšně zastaveno, je uživatel informován dialogovým oknem. Ve výjimečných případech se stává, že měření je zastaveno, ale aplikace to nestihne zachytit v timeoutu. Pokud tedy nevyskočí po kliknutí na tlačítko *Start/stop měření* dialogové okno, je doporučeno aktualizovat status dataloggeru pomocí tlačítka *Aktualizovat informace* a zkontrolovat, zda měření nebylo ukončeno. V případě neúspěchu ukončení měření opakovat.

D.2.4 Čtení aktuálních dat, záznam dat

Čtení aktuálních dat probíhá v záložce *Aktuální data*, která je v panelu na Obrázku D.2 označená červeným rámečkem. Aby bylo možné číst spektrum záření a data z pomocných měřicích zařízení, je potřeba, aby probíhalo měření. Čas dat (systémový čas), teplotu, tlak a vlhkost je možné číst i v době, kdy měření neprobíhá. Ke čtení teploty, tlaku a vlhkosti musí být povolen senzor BME280. Aby bylo možné data číst, je potřeba být k dataloggeru připojený.

Data se aktualizují pravidelně v intervalu, který lze měnit. Změna intervalu aktualizace dat probíhá v panelu *Nastavení aktualizace dat*, který je na Obrázku D.4 v červeném rámečku. Minimální interval aktualizace je 20 sekund, maximální 86400 sekund. Pro změnu intervalu aktualizace dat se nastavuje hodnota intervalu v ovládacím prvku *Nový interval aktualizace dat* a potvrzuje se tlačítkem *Změnit interval*. Pokud je počítač k dataloggeru připojený přes Wi-Fi, tak je doporučeno mít interval stejný, nebo vyšší jako je interval aktualizace mezipaměti v dataloggeru. Data lze aktualizovat hned pomocí tlačítka *Aktualizovat data hned*. Pokud je datalogger připojen přes Wi-Fi, tak se data čtou z mezipaměti Wi-Fi modulu. Pro vynucení aktualizace mezipaměti Wi-Fi modulu se používá tlačítko *Vynutit aktualizaci*, kde je možné nastavit, zdali se aktualizuje i spektrum či nikoliv. V případě připojení přes USB toto

tlačítko nemá žádný efekt, protože data se aktualizují přímo z paměti řídicí jednotky.

Aktuální data se zobrazují v panelu *Spektrum* a *Ostatní* (zelený rámeček na Obrázku D.4). Data se zobrazují textově. Pro grafické zobrazení dat, či uložení těchto dat do soubory, je nutné kliknout na tlačítko *Otevřít aktuální data ve správci dat*, který otevře okno *Správce dat* s nahranými aktuálními daty.

Data je možné průběžně ukládat do paměti počítače v panelu *Záznam dat* (modrý rámeček na Obrázku D.4). Záznam dat se zapíná tím způsobem, že se nejprve vybere interval zaznamenávání dat a následně klikne na tlačítko *Začít nové zaznamenávání*. Interval záznamu dat by měl být stejný, či větší, než je interval aktualizace dat. O průběhu záznamu se může uživatel informovat v panelu *Průběh záznamu*, kde nalezne, zda zaznamenávání probíhá, start zaznamenávání, počet záznamů a nastavený interval záznamů. Zaznamenaná data je možné graficky zobrazit, či uložit ve správci dat pomocí tlačítka *Otevřít záznam ve správci dat*. Záznam končí stiskem tlačítka *Zastavit zaznamenávání*. Poté je možné zaznamenaná data zahodit pomocí tlačítka *Zahodit záznam*.

D.2.5 Manipulace se soubory v dataloggeru

Manipulace se soubory v dataloggeru probíhá v záložce *Soubory*, která je v panelu na Obrázku D.2 označená červeným rámečkem. K manipulaci se soubory v dataloggeru je potřeba hlavní přístup. Datalogger obsahuje dvě čtečky paměťových karet. Jako SD čtečka 0 je označena čtečka, která je na Obrázku D.1 pod číslem 4, SD čtečka 1 je na Obrázku D.1 pod číslem 5.

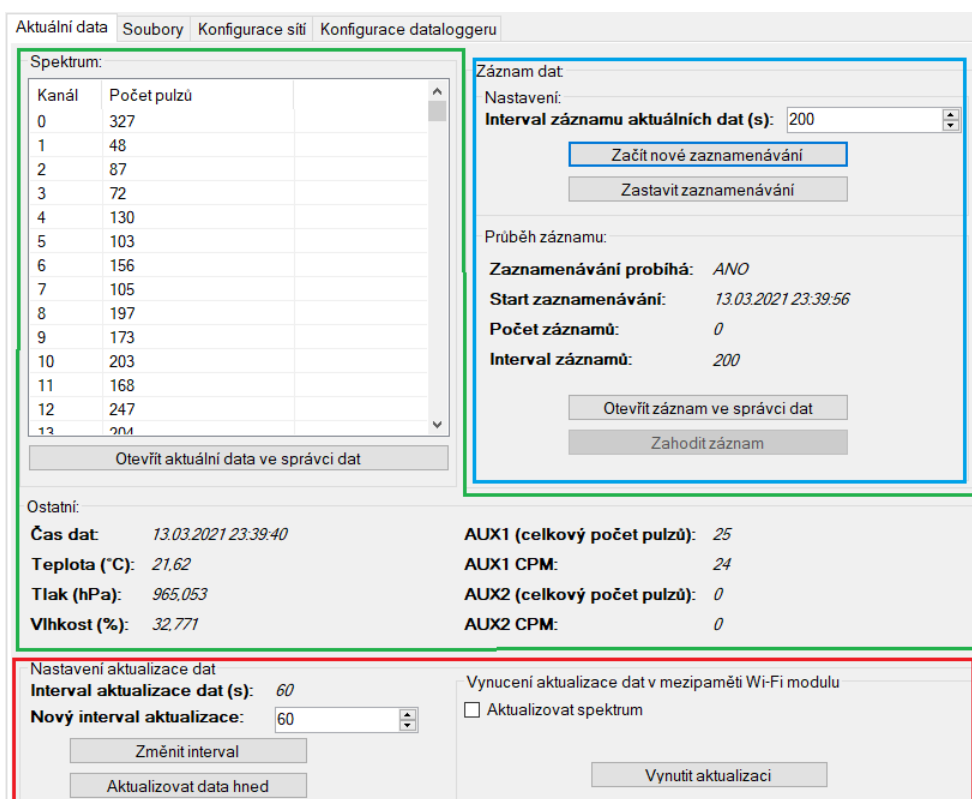
Před mazáním či stahováním souboru je potřeba prvně získat seznam souborů uložených na paměťových kartách. To se dělá následujícím způsobem:

1. Vybrat čtečku v panelu *Aktivní čtečka*, která je na Obrázku D.5 v červeném rámečku.
2. Kliknout na tlačítko *Aktualizovat seznam souborů*, které je na Obrázku D.5 v zeleném rámečku.

Po úspěšném stažení seznamu souborů na paměťové kartě je možné tyto soubory buď mazat, či stahovat. Průběh mazání souborů je následující:

1. Vybrat paměťovou kartu, kde se mají mazat soubory v panelu *Aktivní čtečka*, která je na Obrázku D.5 v červeném rámečku.
2. V seznamu pro vybranou paměťovou kartu (Panely na Obrázku D.5 v modrém rámečku) vybrat soubory, které mají být smazány. Lze vybrat více souborů najednou pomocí přidržení klávesy Ctrl, či vybrat soubory v rozsahu přidržením klávesy Shift.
3. Kliknout na tlačítko *Smazat vybrané soubory*, které je na Obrázku D.5 v oranžovém rámečku.

D. UŽIVATELSKÝ MANUÁL



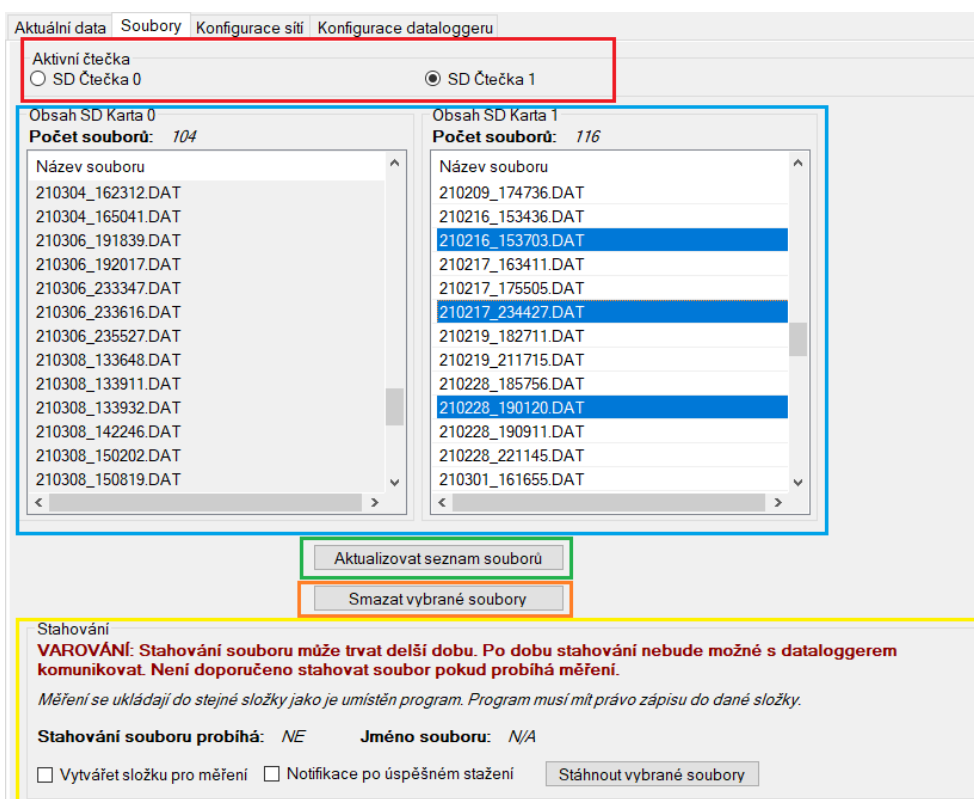
Obrázek D.4: Záložka Aktuální data. Červený rámeček obsahuje nastavení aktualizace dat. Zelený rámeček zahrnuje aktuální data z dataloggeru. Modrý rámeček obsahuje nastavení záznamu dat a samotný průběh záznamu.

Pokud je mazání některého z vybraných souborů neúspěšné, je o tom uživatel informován dialogovým oknem. Nelze mazat soubory, do kterých se ukládají data z probíhajícího měření.

Průběh stahování souborů je následující:

1. Vybrat paměťovou kartu, kde se mají mazat soubory v panelu *Aktivní čtečka*, která je na Obrázku D.5 v červeném rámečku.
2. V seznamu pro vybranou paměťovou kartu (Panely na Obrázku D.5 v modrém rámečku) vybrat soubory, které mají být staženy. Lze vybrat více souborů najednou pomocí přidržení klávesy Ctrl, či vybrat soubory v rozsahu přidržením klávesy Shift.
3. V panelu *Stahování* (na Obrázku D.5 ve žlutém rámečku) vybrat, zdali se má vytvářet složka pro data z měření, která má název RMMDD²¹ a

²¹R = Rok, M = Měsíc, D = Den



Obrázek D.5: Záložka Soubory. Červený rámeček obsahuje výběr aktivní čtečky paměťových karet. Modrý rámeček obsahuje seznam souborů měření a jejich počet pro obě paměťové karty. Zelený rámeček obsahuje tlačítko pro aktualizaci seznamů souborů. Oranžový rámeček obsahuje tlačítko pro smazání vybraných souborů. Žlutý rámeček zahrnuje informace o aktuálním stahování a nastavení stahování.

zda má být uživatel informován o úspěšném stažení dialogovým oknem. Stahování začíná kliknutím na tlačítko *Stáhnout vybrané soubory*.

Pokud není soubor úspěšně stažen, je o tom uživatel informován dialogovým oknem. Průběh stahování lze také pozorovat v panelu *Stahování* (na Obrázku D.5 ve žlutém rámečku), kde je informace o tom, zdali stahování aktuálně probíhá a jaký soubor se stahuje. Stažené soubory vždy přepisují soubory, které mají stejný název. Stahování souborů může trvat delší dobu (až desítky hodin pro velké soubory) a není doporučeno stahovat soubory během probíhajícího měření.

Aktuální data Soubory Konfigurace sítě Konfigurace dataloggeru

Uložené nastavení sítě

Pořadí	SSID	Statická IP Adresa	Maska sítě	Výchozí brána
1	dalsi_sit	192.168.1.100	255.255.255.0	192.168.1.1
2	esp8266_network	0.0.0.0	0.0.0.0	0.0.0.0
3	-	-	-	-
4	DATALOGGER_NETWORK	0.0.0.0	0.0.0.0	0.0.0.0

Aktualizovat uložené sítě

Smazat vybrané sítě

Prohodit vybrané sítě

Dostupné sítě

SSID	Kanál	RSSI	Zabezpečení
esp8266_network	3	-50	WPA1/WPA2
XXXXXXXXXX	3	-50	WPA1/WPA2
XXXXXXXXXX	1	-70	WPA1/WPA2
XXXXXXXXXX	2	-70	WPA1/WPA2
XXXXXXXXXX	11	-73	Auto
XXXXXXXXXX	2	-76	Otevřená
XXXXXXXXXX	10	-77	WPA1/WPA2
XXXXXXXXXX	13	-84	WPA1/WPA2
XXXXXXXXXX	4	-90	Auto
XXXXXXXXXX	4	-91	Auto

Skenovat dostupné sítě

Spravovat uložené sítě

IP adresa: 192.168.1.100

Maska sítě: 255.255.255.0

Výchozí brána: 192.168.1.1

SSID: dalsi_sit

Heslo: 123456789

Stanice:

Typ zabezpečení:

WPA1/2 Otevřená Auto

Zabezpečení WEP není podporováno!

Přidat stanici s danou konfigurací

Přístupový bod

Kanál: 1

Maximální počet klientů: 1

Změnit konfiguraci přístupového bodu

Obrázek D.6: Záložka Konfigurace sítě. Červený rámeček obsahuje uložené konfigurace sítě v dataloggeru. Modrý rámeček obsahuje ovládací prvky pro přidávání nových sítí, či konfiguraci přístupového bodu. Zelený rámeček obsahuje seznam dostupných sítí v okolí. SSID sítí je na obrázku anonymizováno.

D.2.6 Konfigurace bezdrátových sítí

Konfigurace bezdrátových sítí probíhá v záložce *Konfigurace sítě*, která je v panelu na Obrázku D.2 označena červeným rámečkem. Skenování dostupných sítí je možné provádět i bez hlavního přístupu. Pro všechny ostatní operace je nutné mít hlavní přístup. Skenování dostupných sítí se provádí v panelu *Dostupné sítě* (na Obrázku D.6 v zeleném rámečku) stisknutím tlačítka *Skenovat dostupné sítě*. Poté je potřeba chvíli počkat, než se provede sken sítí. Naskenované sítě se objeví v seznamu v panelu *Dostupné sítě*. Kromě SSID se zobrazuje i kanál, na které daná síť běží, síla signálu (RSSI) a typ zabezpečení.

Uloženou konfiguraci sítí lze pozorovat v panelu *Uložené nastavení sítě* (na Obrázku D.6 v červeném rámečku). Pořadí sítí určuje, v jakém pořadí se datalogger bude zkoušet připojit. První tři pozice pořadí jsou pro konfiguraci sítí, ke kterým se datalogger připojuje jako klient, poslední pořadí odpovídá

nastavením přístupového bodu. U konfigurace přístupového bodu nelze změnit pořadí a vždy má pořadí číslo 4. K uložené konfiguraci sítí se lze dostat pomocí stisku tlačítka *Aktualizovat uložené sítě*. Uložené konfigurace lze mazat pomocí vybraní dané sítě v seznamu uložených sítí a stiskem tlačítka *Smazat vybrané sítě*. Pořadí lze měnit tím způsobem, že se vyberou dvě sítě v seznamu (pomocí přidržení tlačítka Ctrl) a stiskem tlačítka *Prohodit vybrané sítě*. Po mazání či prohození pořadí sítí je nutné konfiguraci uložit do EEPROM pomocí stisknutí tlačítka *Uložit konfiguraci do EEPROM*, které se nachází v horní liště okna (na Obrázku D.2 označená žlutým rámečkem).

Změna konfigurace přístupového bodu se provádí v panelu *Spravovat uložené sítě* (na Obrázku D.6 v modrém rámečku). Nastaví se SSID, heslo, IP adresa, maska sítě a výchozí brána, kanál a maximální počet klientů a klikne se na tlačítko *Změnit konfiguraci přístupového bodu*. K použití výchozí konfigurace IP adres/masky sítě/výchozí brány je možné tuto položku nechat nevyplněnou. SSID přístupového bodu musí být v rozsahu 8 – 31 znaků.

Změna konfigurace bezdrátových sítí, ke kterým se datalogger připojuje jako klient není možná. Změna konfigurace se provádí tím způsobem, že je předchozí konfigurace smazána a přidána nová. Nová konfigurace se přidává v panelu *Spravovat uložené sítě* (na Obrázku D.6 v modrém rámečku), kde se nastaví SSID, heslo, IP adresa, maska sítě a výchozí brána. Poté se vybere typ zabezpečení a konfigurace se přidá pomocí stisknutí tlačítka *Přidat stanici s danou konfigurací*. Pokud není potřeba nastavovat statickou IP adresu, nechá se položka *IP adresa*, *Maska sítě* a *Výchozí brána* nevyplněná. Aby bylo možné přidat konfiguraci sítě, je potřeba, aby alespoň jeden slot byl volný. V případě neúspěchu přidání sítě, se zobrazí dialogové okno. Po provedení změny konfigurace přístupového bodu, či při přidání nové konfigurace je nutné konfiguraci uložit do EEPROM pomocí stisknutí tlačítka *Uložit konfiguraci do EEPROM*, které se nachází v horní liště okna (na Obrázku D.2 označená žlutým rámečkem).

D.2.7 Konfigurace dataloggeru

Konfigurace dataloggeru probíhá v záložce *Konfigurace dataloggeru*, která je v panelu na Obrázku D.2 označená červeným rámečkem. Ke konfiguraci dataloggeru je potřeba hlavní přístup. V této záložce se také nastavuje systémový čas pomocí tlačítka *Synchronizovat čas dataloggeru s lokálním časem*, které je na Obrázku D.7 v oranžovém rámečku.

Konfigurace dataloggeru je rozdělena na 4 nezávislé části, každá část konfigurace se potvrzuje vlastním tlačítkem. Vedle ovládacího prvku nastavující konfiguraci je kurzívou napsaná aktuální hodnota daného nastavení. Obecné nastavení (na Obrázku D.7 v červeném rámečku) se potvrzuje pomocí tlačítka *Změnit obecná nastavení*. Nastavení měření (na Obrázku D.7 v modrém rámečku) pomocí tlačítka *Změnit nastavení měření*. Časy pravidelných akcí (na Obrázku D.7 v zeleném rámečku) se potvrzují tlačítkem *Změnit časy*. Položka

D. UŽIVATELSKÝ MANUÁL

Aktuální data | Soubory | Konfigurace sítě | Konfigurace dataloggeru

Obecné nastavení

Wifi modul vypnut *NE*

Port: 44666 *44666*

Typ startu: *Měření na žádost*

Čas opožděného startu (s): 0 *0*

SD Reader 0 zakázán *NE*

SD Reader 1 zakázán *NE*

Změnit obecné nastavení

Synchronizovat čas dataloggeru s lokálním časem

Časy pravidelných akcí

Kontrola sítě (s): 1 *120*

Aktualizace cache (s): 2 *60*

Zápis na SD kartu (s): 4 *900*

Změnit časy

Nastavení měření

Hlavní senzor

Mód spouště: *Sestupná hrana*

Práh: 0 *0*

A/D vytižen signál zakázán *Zakázán*

Posun spektra: 0 *0*

Přidavné měřicí zařízení 1 (AUX1)

Mód spouště: *Náběžná hrana*

Počítání CPM povoleno *ANO*

Přidavné měřicí zařízení 2 (AUX2)

Mód spouště: *Vypnuto*

Počítání CPM povoleno *NE*

BME280 vypnut *NE*

Změnit nastavení měření

Vypínání WiFi modulu

Hodiny	Hodina	Vypnuto
<input type="checkbox"/> 0	0	NE
<input type="checkbox"/> 1	1	NE
<input type="checkbox"/> 2	2	NE
<input type="checkbox"/> 3	3	NE
<input type="checkbox"/> 4	4	NE

Dny:	Den	Vypnuto
<input type="checkbox"/> PO	Pondělí	NE
<input type="checkbox"/> UT	Úterý	NE
<input type="checkbox"/> ST	Středa	NE
<input type="checkbox"/> ČT	Čtvrtek	NE
<input type="checkbox"/> PÁ	Pátek	NE
<input type="checkbox"/> SO	Sobota	NE
<input type="checkbox"/> NE	Neděle	NE

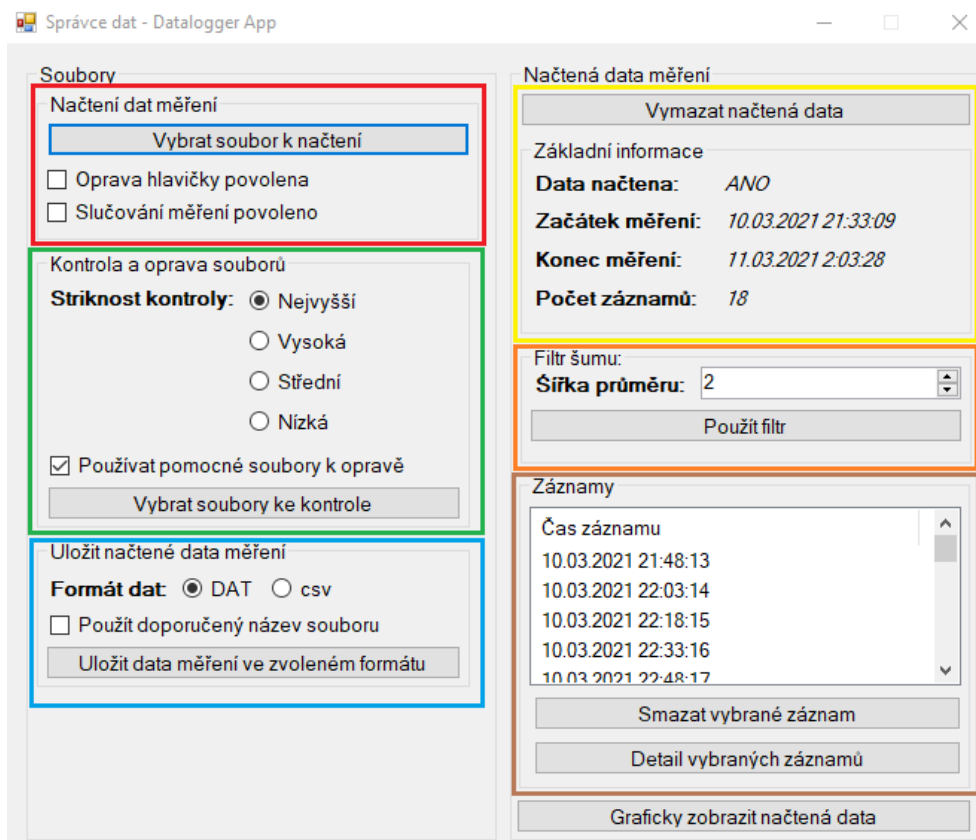
Načíst tabulky vypínání

Změnit nastavení vypínání WiFi modulu

Obrázek D.7: Záložka Konfigurace dataloggeru. Červený rámeček obsahuje ovládací prvky obecnou konfiguraci dataloggeru. V oranžovém rámečku je tlačítko na synchronizaci systémového času dataloggeru s lokálním časem počítače. V zeleném rámečku je nastavení časů pravidelných akcí. V modrém rámečku je konfigurace měření a ve žlutém rámečku vypínání Wi-Fi modulu v závislosti na hodinách/dnech.

Aktualizace cache (s) může obsahovat pouze násobky 2, *Zápis na SD kartu (s)* násobky 4. Vypínání Wi-Fi modulu (na Obrázku D.7 ve žlutém rámečku) se potvrzuje pomocí tlačítka *Změnit nastavení vypínání Wi-Fi modulu*. Tabulky vypínání nejsou ve výchozím stavu načteny a je nutné je načíst pomocí tlačítka *Načíst tabulky vypínání*.

Po změně jakékoliv konfigurace (kromě nastavování systémového času) je potřeba změnu konfigurace uložit do EEPROM pomocí stisknutí tlačítka *Uložit konfiguraci do EEPROM*, které se nachází v horní liště okna (na Obrázku D.2 označená žlutým rámečkem).



Obrázek D.8: Správce dat. Červený rámeček obsahuje ovládací prvky pro načtení souborů měření. V zeleném rámečku jsou ovládací prvky pro kontrolu, opravu a následné načtení souborů. V modrém rámečku jsou ovládací prvky pro uložení dat z měření do souboru csv, či .DAT. Ve žlutém rámečku je stav načtení dat. V oranžovém rámečku jsou ovládací prvky pro filtr dat. Hnědý rámeček obsahuje seznam jednotlivých záznamů.

D.2.8 Správce dat

Správce dat je další okno aplikace, které je možné spustit buď pomocí tlačítka *Správce dat*, které se nachází v horní liště hlavního okna (na Obrázku D.2 označená žlutým rámečkem). V tomto případě se správce dat spouští bez načtených dat. Okno správce dat lze otevřít i s načtenými aktuálními daty, či záznamem, jak se popisuje v D.2.4 Čtení aktuálních dat, záznam dat. Správce dat funguje i v offline režimu, není potřeba být připojený k dataloggeru.

Aby bylo možné s daty pracovat, je nutné je nejprve načíst. Data jsou buď načtena automaticky (viz D.2.4 Čtení aktuálních dat, záznam dat), nebo je třeba je manuálně načíst. Data lze načíst dvěma způsoby. První způsob je

načtení bez kontroly, druhý je načtení s kontrolou. Načtení bez kontroly se provádí v panelu *Načtení dat měření* (na Obrázku D.8 v červeném rámečku). V tomto panelu je možné zapnout slučování měření (slučovat lze pouze měření se stejnou hlavičkou) a případnou opravu hlavičky. Po stisknutí tlačítka *Vybrat soubor k načtení* se otevře dialogové okno, kde uživatel vybere cestu k souboru. Povolené formáty jsou .DAT (binární formát dataloggeru) a .csv.

Druhý způsob načtení dat je načtení s kontrolou a případnou opravou. To se dělá v panelu *Kontrola a oprava souborů* (na Obrázku D.8 v zeleném rámečku), kde se vybere striktnost kontroly (více v kapitole 4.2.4 Oprava souborů) a nastaví se, zda se k opravě používají pomocné soubory. Pokud se pomocné soubory nepoužijí, může docházet k úpravě zadaných souborů. Poté se klikne na tlačítko *Vybrat soubory ke kontrole* a následně vyskočí dialogová okna, kde uživatel vybere první a druhý soubor k porovnání. Provede se kontrola, případná oprava souboru a následné načtení (pokud je to možné). O výsledku je uživatel informován dialogovým oknem.

Po načtení je možné s daty pracovat. Detail měření je v panelu *Základní informace* (na Obrázku D.8 v žlutém rámečku). Zde je možné i načtená data zahodit pomocí tlačítka *Vymazat načtená data*. Načtená data je také možné uložit buď ve formátu csv, nebo původního formátu DAT. Uložení je možné v panelu *Uložit načtená data měření* (na Obrázku D.8 v modrém rámečku). Je možné použít doporučený název měření, či vlastní. Po kliknutí na tlačítko *Uložit data měření ve zvoleném formátu* se otevře dialogové okno s výběrem cesty, kam soubor uložit.

Pokud načtená data obsahují šum, který ruší grafickou interpretaci, je možné použít filtr šumu. Vlastnosti filtru šumu se nastavují v panelu *Filtr šumu* (na Obrázku D.8 v oranžovém rámečku). Vybrat lze šířku klouzavého průměru. Filtr se aplikuje tlačítkem *Použít filtr*. Uživatel je o úspěšné aplikaci filtru informován dialogovým oknem.

S jednotlivými záznamy je možné manipulovat v panelu *Záznamy* (na Obrázku D.8 v hnědém rámečku). Jednotlivé záznamy je možné buď smazat tím způsobem, že se záznamy vyberou v seznamu a klikne se na tlačítko *Smazat vybrané záznamy*. Je možné zobrazit detail jednotlivých záznamů tím způsobem, že se záznamy, které uživatele zajímají, vyberou v seznamu a otevřou se tlačítkem *Detail vybraných záznamů*.

Načtená data lze graficky zobrazit pomocí stisknutí tlačítka *Graficky zobrazit načtená data*.

D.2.9 Grafické zobrazení dat

Okno grafické zobrazení dat se otevírá ze správce dat. Grafické zobrazení dat funguje i v offline režimu. Rozsah záznamů, které se zobrazují, se nastavuje v panelu *Nastavení dat* (na Obrázku D.9 v zeleném rámečku). Ve výchozím stavu se zobrazují všechna načtená data (počátek je první záznam, konec

poslední záznam). Jaké data jsou vybrána se zobrazují v panelu *Vybraná data* (na Obrázku D.9 v žlutém rámečku).

Vybraná data se graficky zobrazují v panelu *Grafy* (na Obrázku D.9 v červeném rámečku). Tento panel obsahuje 3 záložky. První záložkou je *Spektrum*, ve které se zobrazuje graf spektra ionizujícího záření. Zobrazení grafu se mění v panelu *Nastavení zobrazení spektra* (na Obrázku D.9 v modrém rámečku). Lze změnit rozsah kanálů (panel *Osa X*) a rozsah počtu pulzů (panel *Osa Y*). Změna se provádí tlačítkem *Překreslit graf*. Výchozí zobrazení grafu spektra – tedy osa X od 0. do maximálního kanálu dat, osa Y od 0. do maximálního počtu pulzů – se zobrazuje pomocí tlačítka *Použít výchozí zobrazení*. Další funkcí vizualizéru dat je integrování píku. Integrace píků se provádí v panelu *Integrovat pík*. Integrace provede součet impulzů v daném rozsahu. Integrace proběhne po stisku tlačítka *Integrovat*.

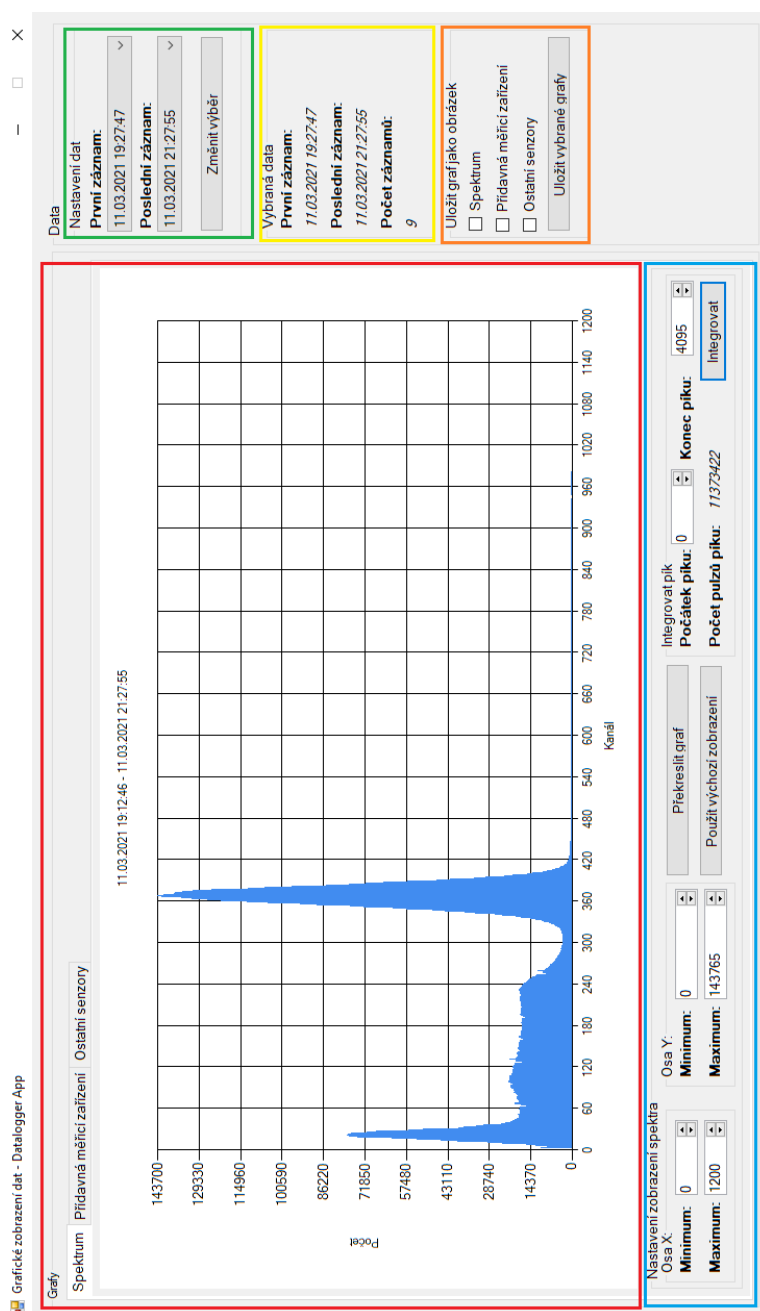
Druhou záložkou je *Přídavná měřicí zařízení*. V této záložce se zobrazuje graf průběhu počtu pulzů detekovaných pulzů pomocnými měřicími zařízeními. V panelu *Nastavení zobrazení spektra* v podpanelu *Zobrazit* je možné vybrat, zda se bude zobrazovat graf počtu impulzů za minutu (CPM), či celkový počet detekovaných pulzů. V panelu *Nastavení zobrazení spektra* v podpanelu *Povolit* se nastavuje, zda se v grafu bude zobrazovat pomocné měřicí zařízení 1 (AUX1), pomocné měřicí zařízení 2 (AUX2) či oboje pomocí zaškrtnutých políček *AUX1* popř. *AUX2*.

Poslední záložkou je záložka *Ostatní senzory*, kde se v jednotlivých grafech zobrazuje vývoj teploty, tlaku a vlhkosti v závislosti na čase. Narozdíl od grafu spektra a přídavných měřicích zařízení nelze měnit zobrazení jednotlivých grafů. Lze pouze změnit rozsah dat v panelu *Nastavení dat* (na Obrázku D.9 v zeleném rámečku).

Grafy je možné uložit jako obrázek formátu JPG, PNG či BMP. Uložení grafů probíhá v panelu *Uložit graf jako obrázek* (na Obrázku D.9 v oranžovém rámečku). Grafy se ukládají pomocí stisku tlačítka *Uložit vybrané grafy*. Uloží se ty grafy, které jsou zaškrtnuté v panelu *Uložit graf jako obrázek*. Po stisku tlačítka se objeví dialogové okno, kde uživatel vybere lokaci, kam soubor uložit, jméno souboru a požadovaný formát. Za jméno souboru se automaticky přidá `_SPECTRUM` pro graf spektra, `_AUX` pro graf pomocných měřicích jednotek a `_OTHER` pro ostatní senzory. Grafy se ukládají v podobě, ve které jsou zobrazené (uloží se tedy to, co uživatel vidí).

D.3 První spuštění

Před prvním použitím dataloggeru je třeba provést konfiguraci sítě a dataloggeru. Datalogger obsahuje výchozí konfiguraci, která se načítá v době, kdy není inicializovaná EEPROM. Výchozí konfigurace se načítá i v době, kdy není možné provést načtení konfigurace z EEPROM. Kromě výchozí konfigurace je i výchozí nastavení přístupového bodu, který datalogger vytvoří, pokud nemá



Obrázek D.9: Okno grafického zobrazení dat. Červený rámeček obsahuje grafy. Je rozdělen do 3 záložek, první záložkou je graf spektra, druhou průběh pomocných měřicích zařízení a třetí zahrnuje senzory teploty, tlaku a vlhkosti. Modrý rámeček obsahuje ovládací prvky pro nastavení zobrazení dat. Zelený rámeček obsahuje ovládací prvky na výběr rozsahu záznamů. Žlutý rámeček obsahuje informace o vybraných záznamech. V oranžovém rámečku jsou ovládací prvky pro uložení grafů jako obrázek.

k dispozici jiné údaje o bezdrátových sítích. Výchozí SSID přístupového bodu je **DATALOGGER_NETWORK** a heslo je **MYT69IFBRceTN89k**.

Postup konfigurace dataloggeru po prvním spuštění je následující:

1. Vložení do držáků baterií knoflíkové baterie CR2032 (či jejich ekvivalent). Místa, kam se knoflíkové baterie vkládají jsou na Obrázku D.1 označeny jako 3 a 11.
2. Připojit do napájecího USB (na Obrázku D.1 jako 1) zdroj (např. powerbanka).
3. Pomocí obslužné aplikace se k dataloggeru připojit. Připojit se lze buď pomocí komunikačního USB (na Obrázku D.1 jako 2), nebo bezdrátově po připojení počítače k přístupovému bodu vytvořený dataloggerem. Přístupový bod má SSID **DATALOGGER_NETWORK** a heslo **MYT69IFBRceTN89k**. Jak se připojit k dataloggeru popisuje D.2.1 Navázení spojení s dataloggerem.
4. Synchronizovat čas v záložce Konfigurace dataloggeru stiskem tlačítka *Synchronizovat čas dataloggeru s lokálním časem*.
5. Provést konfiguraci dataloggeru (více o konfiguraci dataloggeru v kapitole D.2.7 Konfigurace dataloggeru).
6. Provést konfiguraci sítě (více o konfiguraci sítě v kapitole D.2.6 Konfigurace bezdrátových sítí).
7. Uložit konfiguraci do EEPROM pomocí stisknutí tlačítka *Uložit konfiguraci do EEPROM*, které se nachází v horní liště okna (na Obrázku D.2 označený žlutým rámečkem).
8. Resetovat zařízení buď pomocí vytažení napájecího USB, či pomocí tlačítka *Reset zařízení*, které se nachází na Obrázku D.2 v modrém rámečku.
9. Tímto je prvotní konfigurace hotová a zařízení je možné používat.