



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**  
DEPARTMENT OF COMPUTER SYSTEMS

## **DIGITÁLNÍ ZÁZNAMOVÁ JEDNOTKA S PREVENCÍ ZTRÁTY DAT PŘI VÝPADKU NAPÁJENÍ**

DIGITAL DATA LOGGER WITH INFORMATION LOSS PREVENTION IN CASE OF POWER  
FAILURE

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**TOMÁŠ DOLÁK**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. VÁCLAV ŠÍMEK**

**BRNO 2025**

## Zadání bakalářské práce



Ústav: Ústav počítačových systémů (UPSY) 164166  
Student: **Dolák Tomáš**  
Program: Informační technologie  
Název: **Digitální záznamová jednotka s prevencí ztráty dat při výpadku napájení**  
Kategorie: Vestavěný systémy  
Akademický rok: 2024/25

### Zadání:

1. Připravte rešerši existujících přístupů ke sběru a ukládání dat v oblasti vestavěných systémů a to včetně vkládání synchronizačních značek (např. časových razítok) do pořizovaného záznamu.
2. Navrhnete koncepci vestavěného zařízení pro záznam dat (dále jen „záznamníku“), přičemž dbejte následujících požadavků: **A)** autonomní dokončení záznamu dat při detekci krátkodobého výpadku napájení, **B)** doplnění ukládaných dat o synchronizační značky umožňující jejich další analýzu a zpracování, **C)** možnost vyčtení uložených dat bez nutnosti vyjmutí paměťového média ze záznamníku, **D)** podpora USB komunikace se záznamníkem a možnost jeho přímé obsluhy skrze ovládací prvky.
3. V návaznosti na koncepci z bodu 2) zadání vytvořte podrobné blokové schéma záznamníku. Dále provedte výběr vhodných komponent pro jeho technickou realizaci.
4. Ve vhodném editoru navrhnete obvodové zapojení potřebných částí záznamníku a vytvořte k nim desku plošných spojů. Pro zvolený typ řídicího mikrokontroleru lze využít dostupné evaluační desky.
5. Implementujte obslužný firmware zajišťující požadované funkční vlastnosti zařízení. Dbejte na modulární strukturu programového řešení a jeho pečlivé zdokumentování.
6. Vlastnosti realizovaného záznamníku vhodným způsobem otestujte v praxi. Pečlivě vyhodnotěte zjištěné skutečnosti.
7. Diskutujte možné směry úprav záznamníku a blíže rozveďte ty, které považujete za nejperspektivnější.

### Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1.11.2024

Termín pro odevzdání: 14.5.2025

Datum schválení: 31.10.2024

## **Abstrakt**

Tato závěrečná práce se zabývá návrhem a vývojem digitálního záznamníku s prevencí ztráty dat při výpadku napájení. Hlavní pozornost je věnována výběru vhodných komponent, návrhu a implementaci firmwaru záznamníku, konstrukci expanzní desky s potřebnými součástkami a testování výsledného zařízení. Vyvinutý záznamník bude využíván v Qi certifikačních laboratořích pro záznam dat z bezdrátových nabíječek podporovaných společností NXP Semiconductors. Díky své univerzálnosti však může být po úpravě použit i pro záznam dalších typů dat v různých aplikacích.

## **Abstract**

This thesis focuses on the design and development of a digital recorder with information loss prevention in case of power failure. The main attention is given to the selection of suitable components, design and implementation of the logger firmware, construction of the expansion board with the necessary components and testing of the resulting device. The developed logger will be used in Qi certification labs to record data from wireless chargers supported by NXP Semiconductors. However, due to its versatility, it can also be used to record other types of data in various applications after modification.

## **Klíčová slova**

vestavěné zařízení, digitální záznamník, návrh expanzní desky, zálohované napájení, prevence ztráty dat, SDHC karta, FAT souborový systém

## **Keywords**

embedded device, digital recorder, expansion board design, power backup, data loss prevention, SDHC card, FAT file system

## **Citace**

DOLÁK, Tomáš. *Digitální záznamová jednotka s prevencí ztráty dat při výpadku napájení*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šímek

# **Digitální záznamová jednotka s prevencí ztráty dat při výpadku napájení**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením panem Ing. Václavem Šímkem a konzultantem Ing. Martinem Moškem PhD. Další informace mi poskytli mý kolegové z firmy NXP Semiconductors. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Tomáš Dolák  
12. května 2025

## **Poděkování**

Rád bych vyjádřil své upřímné poděkování vedoucímu mé bakalářské práce, Ing. Václavu Šímkovi, za jeho odborné vedení, cenné rady a aktivní podporu během celého procesu. Velmi si cením jeho pomoci při návrhu a realizaci desky plošných spojů, zajištění její výroby a také poskytnutí potřebného vybavení pro měření spotřeby, které bylo klíčové pro finální návrh zařízení.

Dále bych chtěl poděkovat svému firemnímu vedoucímu Martinu Mostkovi za to, že tuto práci inicioval a převzal nad ní odborný dohled. Jeho rady, konstruktivní kritika a čas, který mi věnoval, významně přispěly k úspěšnému dokončení projektu.

Mé díky patří také kolegům z firmy NXP Semiconductors za jejich podporu a cenné podněty během vývoje digitálního záznamníku. Zvláštní poděkování si zaslouží i samotná firma NXP Semiconductors za její dlouhodobou podporu studentů, a to nejen v rámci odborné spolupráce, ale i při psaní závěrečných prací. Právě díky této podpoře mohla tato práce vzniknout.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Záznam dat</b>	<b>7</b>
2.1	Historické pozadí záznamu dat . . . . .	7
2.2	Záznam dat v počátcích elektrotechniky . . . . .	8
2.3	Digitální zpracování dat . . . . .	9
2.3.1	Princip digitálního záznamu dat . . . . .	9
2.3.2	Digitální záznamník . . . . .	10
2.3.3	Digitální záznam v počítačovém systému . . . . .	12
2.3.4	Digitální záznam na platformě mikrořadiče . . . . .	13
2.4	Klíčové koncepty digitálních záznamníku . . . . .	15
2.4.1	Vícenásobná vyrovňávací paměť (multiple-buffering) . . . . .	15
2.4.2	Dávkové ukládání (batch saving) . . . . .	17
2.4.3	Cirkulární buffer . . . . .	18
2.4.4	Nízkoenergetické režimy . . . . .	19
2.5	Koncepty vkládání synchronizačních značek do záznamu . . . . .	20
2.6	Způsoby ukládání dat . . . . .	21
2.6.1	Přímý zápis na permanentní uložiště . . . . .	21
2.6.2	Zápis na permanentní uložiště přes mezivrstvu s FeRAM paměti . . . . .	22
2.6.3	Zápis na vzdálené uložiště . . . . .	23
<b>3</b>	<b>Návrh digitálního záznamníku</b>	<b>25</b>
3.1	Existující řešení digitálních záznamníků . . . . .	25
3.1.1	Keeylog AirDrive Serial Logger . . . . .	25
3.1.2	Anticyclone Systems AntiLog Data Logger Pro . . . . .	26
3.1.3	Shrnutí představených řešení . . . . .	27
3.2	Výběr vhodné platformy . . . . .	27
3.2.1	NXP FRDM-MCXN947 . . . . .	28
3.2.2	Raspberry PI . . . . .	29
3.3	Možnosti správy dat – souborové systémy . . . . .	30
3.3.1	FATFS . . . . .	31
3.3.2	LittleFS . . . . .	32
3.4	Výběr řízení přístupu k získaným datům . . . . .	33
3.4.1	USB Mass Storage . . . . .	34
3.4.2	Media Transfer Protocol . . . . .	35
3.5	Výběr zdroje času pro časové značky . . . . .	36
3.5.1	Interní časovač . . . . .	36
3.5.2	Obvod reálného času . . . . .	37

3.5.3	Vzdálené zdroje času . . . . .	38
3.6	Výber přístupu řízení běhu aplikace . . . . .	41
3.6.1	Bare-Metal . . . . .	41
3.6.2	Operační systém reálného času . . . . .	42
3.7	Architektura systému digitálního záznamníku . . . . .	44
<b>4</b>	<b>Realizace hardwaru</b>	<b>46</b>
4.1	Expanzní deska . . . . .	46
4.1.1	Zálohované napájení . . . . .	46
4.1.2	Obvod reálného času . . . . .	50
4.1.3	GPS pro synchronizaci času . . . . .	51
4.1.4	Signalizace stavu systému . . . . .	52
4.1.5	Layout expanzní desky . . . . .	53
4.1.6	Cenová kalkulace součástek expanzní desky . . . . .	54
4.1.7	Výsledná expanzní deska . . . . .	55
<b>5</b>	<b>Přístup k realizaci obslužného firmware</b>	<b>56</b>
5.1	Záznamová úloha . . . . .	59
5.2	USB Mass Storage úloha . . . . .	62
5.3	Detekce ztráty napájejícího napětí . . . . .	62
5.4	Signalizace stavu systému . . . . .	64
<b>6</b>	<b>Testování systému</b>	<b>66</b>
6.1	Funkcionální testování systému . . . . .	66
6.2	Kontrola bezpečnosti kódu . . . . .	67
6.3	Limitace systému . . . . .	68
6.4	Možná vylepšení a úpravy digitálního záznamníku . . . . .	68
<b>7</b>	<b>Závěr</b>	<b>69</b>
<b>Literatura</b>		<b>71</b>
<b>A Obsah přiloženého paměťového média</b>		<b>77</b>
<b>B Projekt digitálního záznamníku</b>		<b>79</b>
<b>C Použité nástroje</b>		<b>80</b>

# Seznam obrázků

2.1	První grafický záznamník (chart recorder) patentovaný Williamem Henrym Bristolem v roce 1888 [38] . . . . .	7
2.2	Ukázka zařízení patřící do skupiny elektromechanických záznamníků – poly- graf [58] . . . . .	8
2.3	Průběh analogového a digitálního signálu . . . . .	9
2.4	Obecné schéma digitálního záznamníku . . . . .	11
2.5	Tok logovacích dat od aplikací a procesů přes Syslogd [41] . . . . .	12
2.6	Příklad architektury digitálního záznamníku pro měření teploty . . . . .	14
2.7	Schéma principu práce s vícenásobnou vyrovnávací pamětí – sběr dat . . . . .	15
2.8	Schéma principu práce s vícenásobnou vyrovnávací pamětí – prohození uka- zatelů . . . . .	16
2.9	Příklad organizace nevolatilní paměti typu NAND [55] . . . . .	17
2.10	Cirkulární vyrovnávací paměť . . . . .	18
2.11	Graf znázorňující dynamiku spotřeby mikrokontroléru v průběhu času při využití aktivního a nízkoenergetického režimu [62] . . . . .	19
2.12	Přímý zápis na permanentní uložiště s SD kartou za pomocí čtyř pinové datové sběrnice . . . . .	21
2.13	Struktura 1T-1C feroelektrické RAM paměti [43] . . . . .	22
2.14	Architektura pokročilého systému digitálních záznamníků s cloudovým úlo- žištěm a platformou Kafka [70] . . . . .	23
3.1	Anticyclone Anti-Log Pro [2] . . . . .	26
3.2	Vývojová deska NXP FRDM-MCXN947 [8] . . . . .	29
3.3	Vývody Raspberry Pi Zero 2 W [17] . . . . .	30
3.4	Souborový systém s alokační tabulkou souborů [45] . . . . .	31
3.5	Oblasti souborového systému FAT [45] . . . . .	32
3.6	Žurnálování při zápisu do souboru [68] . . . . .	32
3.7	Stavový diagram USB Mass Storage protokolu [19] . . . . .	34
3.8	Stavový diagram Media Transfer Protocol [16] . . . . .	35
3.9	Blokový diagram časovače [12] . . . . .	37
3.10	Zapojení obvodu reálného času v podobě externího modulu připojeného k mi- krokontroléru [11] . . . . .	38
3.11	Minimální nutná sestava čtyř GPS satelitů pro výpočet lokace a času přijí- mače [10] . . . . .	39
3.12	Komunikace pomocí Network Time Protocol [24] . . . . .	40
3.13	Příklad vestavěnné aplikace využívající bare-metal přístup . . . . .	41
3.14	Graf znázorňující rozdělení výpočetního času, při řízení programu operačním systémem reálného času s plánovačem [37] . . . . .	43

3.15 Výsledná architektura digitálního záznamníku . . . . .	45
4.1 Výsledek měření spotřeby implementovaného záznamníku pomocí Power Profiler Kit II . . . . .	46
4.2 Obvod pro testování kondenzátoru . . . . .	47
4.3 Průběh napětí na kondenzátoru při vybíjení zachycený osciloskopem . . . . .	48
4.4 Schéma obvodu pro detekci ztráty napájení a zálohovaného napájení . . . . .	49
4.5 Zapojení obvodu reálného času – DS3231M+ . . . . .	50
4.6 Připojení GPS modulu NEO-7M . . . . .	51
4.7 Ukázka zapojení jedné z kontrolních diod typu LED . . . . .	52
4.8 Layout expanzní desky – přední strana . . . . .	53
4.9 Layout expanzní desky – spodní strana . . . . .	53
4.10 Výsledná expanzní deska – přední strana . . . . .	55
4.11 Výsledná expanzní deska – spodní strana . . . . .	55
5.1 Zjednuduššený diagram běhu firmwaru digitálního záznamníku, včetně přepínání úloh . . . . .	58
5.2 Detekce ztráty napájení, zachycena pomocí logického analyzátoru Saleae Logic Pro 16 . . . . .	64
5.3 Signalační diody digitálního záznamníku . . . . .	64
7.1 Pohled na vnitřní uspořádání digitálního záznamníku vyvinutého pro společnost NXP Semiconductors . . . . .	70

# Kapitola 1

## Úvod

Tato bakalářská práce je věnována návrhu a implementaci digitálního záznamníku s autonomním dokončením záznamu dat při výpadku napájecího napětí. Požadavek na zařízení vznikl od firmy NXP Semiconductors, konkrétně od týmu zaměřeného na bezdrátové nabíjení, ve kterém pracuji. Tento tým působí v České republice, jak v Rožnově pod Radhoštěm, tak i v Brně, a zároveň má své zastoupení v Asii a Severní Americe. NXP Semiconductors je jedním z předních členů WPC (Wireless Power Consortium), organizace zodpovědné za definování standardu Qi pro bezdrátové nabíjení. Primární zaměření NXP v této oblasti spočívá ve vývoji referenčních designů pro automotive sektor, kde zákazníkům poskytuje řešení určená pro integraci do vozidel.

Zákazníci, kteří využívají referenční designy NXP, pocházejí z celého světa a dostávají téměř hotový produkt, který lze následně certifikovat v Qi certifikačních laboratořích. Nicméně i přesto, že jsou referenční designy navrženy podle nejnovějších standardů, často dochází k jejich úpravám podle specifických požadavků zákazníků, zejména s ohledem na konkrétní požadavku koncového zákazníka (OEM – Original Equipment Manufacturer). Tyto požadavky jsou obvykle shrnutы v RFP (Request for Proposal), kde zákazník specifikuje konkrétní požadavky na systém. Tyto úpravy mohou být například realizovány z důvodu snížení ceny nebo zlepšení výkonu, EMC charakteristik a nebo dosažení specifického chování bezdrátové nabíječky v krajních situacích.

Při jakýchkoli úpravách však vznikají nové technické výzvy, a proto NXP poskytuje zákazníkům plnou technickou podporu až do úspěšné certifikace. Certifikace probíhá v různých laboratořích po celém světě, avšak ne vždy může být přítomen zaměstnanec NXP, který by dohlížel na celý proces a zajistil, že certifikace proběhne hladce. V těchto případech se momentálně tým pro bezdrátové napájení spoléhá pouze na záznamy poskytnuté operátorem certifikační laboratoře. Tyto záznamy však pocházejí pouze ze strany přijímače, tedy certifikačního zařízení, která jsou zpravidla od výrobců nok9 nebo Granite River Labs (GRL). Certifikační zařízení poskytuje některé z důležitých informací, nicméně tyto nabídnuté záznamy nezahrnují explicitní informace o chování vysílače. Pokud tedy bezdrátová nabíječka, respektive bezdrátový vysílač, neprojde některým z testů, bývá často obtížné zpětně určit příčinu problému. Právě z tohoto důvodu vznikla potřeba záznamníku dat, který umožňuje zaznamenávat data přímo ze strany bezdrátové nabíječky a následně je porovnat s údaji získanými z testovacího zařízení.

Návrh a implementace záznamníku musí reflektovat požadavky na snadnou obsluhu, neboť zařízení bude poskytováno zákazníkům pro již zmíněné účely certifikace. V klasickém scénáři zákazník předá nabíječku i se záznamníkem operátorovi certifikační laboratoře, ten si záznamník připojí k testovanému zařízení. Po skončení testování operátor záznamník vrátí

zákazníkovi, který jej následně připojí k počítači a odešle společnosti NXP Semiconductors získané záznamy.

Z hlediska textové struktury se tato bakalářská práce v kapitole 2 zabývá problematikou záznamu dat z historického i technického hlediska. Popisuje vývoj záznamových systémů od mechanických a analogových zařízení až po moderní digitální řešení. Dále se zaměřuje na principy digitálního záznamu, popisuje základní architekturu digitálních záznamníků a klíčové koncepty, které se uplatňují při jejich návrhu a implementaci.

Kapitola 3 je věnována procesu návrhu digitálního záznamníku. V úvodu kapitoly jsou rozebrána existující řešení z oblasti záznamu dat, která jsou podobná zařízení vyvíjenému v této bakalářské práci, a zároveň poskytují přehled o jejich výhodách a omezeních. Následně jsou představeny jednotlivé hardware a softwarové komponenty, které byly brány v úvahu pro implementaci digitálního záznamníku, přičemž u každé části systému je uvedeno více alternativ (byť nejde o jediná možná řešení). V závěru kapitoly je pak popsána výsledná architektura záznamníku.

Kapitola 4 se zaměřuje na realizaci hardware části digitálního záznamníku v podobě dedikovaného zařízení. Klíčovým prvkem této části je návrh a realizace rozšiřující expanzní desky, která byla navržena speciálně pro potřeby digitálního záznamníku.

Kapitola 5 je věnována implementaci firmwaru digitálního záznamníku. Nejprve je popsána celková struktura firmwaru, který zajišťuje funkci zařízení jako autonomního záznamového systému. Následně jsou rozebrány hlavní funkční celky, včetně mechanismu záznamu dat, přístupu k uloženým datům bez fyzického vyjmutí paměťového média, detekce výpadku napájení a signalizace stavu zařízení vůči uživateli.

Kapitola 6 popisuje proces, jakým bylo zařízení testováno a validováno. Primárně se zaměřuje na funkcionální testování zařízení, tj. ověření, že systém vykazuje správné chování. Další část je věnována kontrole bezpečnosti a spolehlivosti implementovaného kódu. V závěru kapitoly jsou zmíněna možná vylepšení realizovaného digitálního záznamníku.

# Kapitola 2

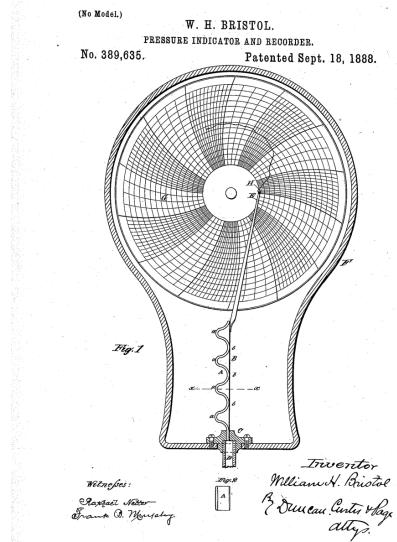
## Záznam dat

### 2.1 Historické pozadí záznamu dat

Lidstvo již od svých počátků mělo potřebu zaznamenávat data, neboť člověk mnohdy dokáže datům přiřadit sémantiku – jejich význam, a proměnit je tak v informace. Právě díky informacím se lidé mohou učit z minulých zkušeností, předávat znalosti dalším generacím, organizovat a podpořit tak neustálý lidský pokrok.

Po staletí byl záznam dat výhradně manuální. Informace se uchovávaly v rukopisech, knihách či na papírových svitcích, ať už formou psaného textu, nebo ručně zapisovanými výsledky pozorování. Zaznamenaná data však byla limitována svou přesností, která byla do značné míry ovlivněna lidskou obsluhou a možnostmi mechanických nástrojů, tak i svou dostupností, protože k nim měl přístup jen omezený okruh lidí. [27]

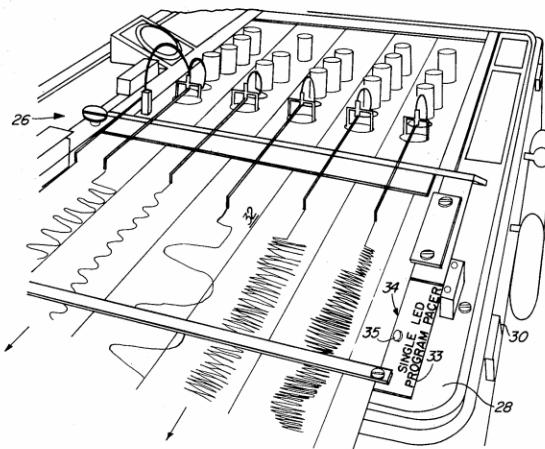
Stěžejní změna přišla na konci 19. století s rozvojem elektrotechniky a nástupem automatických záznamníků, které umožnily jednoduchý sběr dat bez nutnosti lidského zásahu. Namísto ručně zapisovaných měření začaly být hodnoty přenášeny přímo do záznamových zařízení, která je dokázala systematicky uchovávat. [26]



Obrázek 2.1: První grafický záznamník (chart recorder) patentovaný Williamem Henrym Bristolom v roce 1888 [38]

## 2.2 Záznam dat v počátcích elektrotechniky

Prvními specializovanými záznamníky byly mechanické či elektromechanické zařízení, využívající principu analogového záznamu dat. Jejich primáním účelem bylo zaznamenávání fyzikálních veličin, jako je například teplota, tlak, vlnkost nebo vibrace. Tyto přístroje využívaly myšlenky mechanického pohyblivého pera, které převádělo naměřenou hodnotu fyzikální veličiny na samočinný pohyb. Pro realizaci tohoto pohybu bylo nutné nejprve převést měřenou fyzikální veličinu na mechanický posun, přičemž konkrétní způsob převodu závisel na typu měřené veličiny. Například pro měření teploty se běžně využíval bimetalový pásek, složený ze dvou kovových materiálů s různou hodnotou teplotní roztažitelnosti. Při změně teploty docházelo k prohnutí pásku v důsledku rozpínání kovu, čímž bylo rozpohybováno mechanické pero, které zapisovalo hodnotu na paměťové médium. [21]



Obrázek 2.2: Ukázka zařízení patřící do skupiny elektromechanických záznamníků – polygraf [58]

Tyto přístroje jsou běžně používány od druhé poloviny 19. století. Pro již zmíněný záznam teploty je možné například využít přístroj zvaný cirkulární grafový záznamník (circular chart recorder). Zajímavým přístrojem je také polygraf, který lze využít jako detektor lží. Značnou nevýhodou těchto záznamníků bývá typ paměťového média, na které probíhá zápis hodnot, nejčastěji jim je papírová pánska nebo papírový buben. Nevýhodou těchto paměťových médií je nutnost jejich časté výměny za nové, ještě nepopsané, jelikož výsledné záznamy by se jinak staly značně nepřehlednými, pokud by byly popsány vícekrát. [26]

Hlavním omezením analogových záznamových systémů je jejich vysoká specializace pro jediný konkrétní typ záznamu. Tyto záznamníky nejsou snadno upravitelné pro jiné účely, na rozdíl od digitálních řešení, která umožňují flexibilnější přizpůsobení (například pouhou úpravou softwarového programu) k sledování monitorované soustavy. Často je v těchto případech nutné využít jiné řešení. [44]

Další limitací těchto přístrojů je ruční vyhodnocování dat, což je mnohdy časově zdlouhavé a také náchylné k chybám. K správné interpretaci dat je pak často potřeba zkušená obsluha a v některých případech i pomocné měřící pomůcky. Přenos dat a automatizace jsou také komplikované, proto jakmile se v polovině 20. století začaly na trh dostávat číslicové systémy, začaly tyto digitální technologie analogové záznamové systémy postupně nahrazovat. [27, 61]

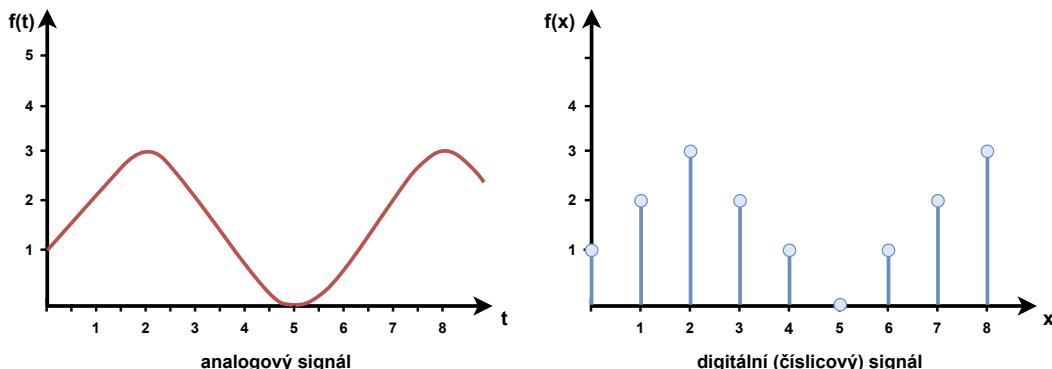
## 2.3 Digitální zpracování dat

S nástupem číslicových systémů v polovině 20. století došlo k velkému pokroku ve způsobu, jakým byla data zaznamenávána, zpracovávána a uchovávána. Digitální zpracování dat v porovnání s analogovým zpracováním se liší způsobem, jakým se v systému obecně pracuje se signály. Zatímco analogový záznam pracuje se spojitými (kontinuálními) signály, digitální záznam využívá diskrétní (číslicové) hodnoty, které jsou uchovávány v binární podobě.

### 2.3.1 Princip digitálního záznamu dat

Z předchozího textu je tedy zřejmé, že digitální zpracování dat (tedy i záznam dat) vyžaduje číslicové hodnoty a z toho plyne, že již na vstupu musí být příchozí signály v digitální podobě. Data tedy musí být generována číslicovými zdroji nebo musí být převedena do digitálního tvaru pomocí komponenty k tomu určené, kterou je například analogově-digitální převodník, aby následně mohly být zpracovány výpočetním strojem a případně uloženy na paměťové médium.

Pro převod analogových signálů do jejich digitální podoby se využívá proces digitalizace, který zahrnuje tři základní kroky. V počátku dochází ke vzorkování, při kterém je původní spojitý signál snímán až už v pravidelných či nepravidelných časových intervalech. Následně dochází ke kvantizaci, při níž jsou vzorkované hodnoty zaokrouhleny na nejbližší úroveň amplitudy v omezeném rozsahu. Vzorky jsou tedy rozděleny do segmentů (tříd) diskrétních hodnot, přičemž tento krok s sebou nese ztrátu přesnosti závislou na jemnosti stupnice, která se nazývá kvantizační chyba. Nakonec je kvantizovaný signál kódován do binární podoby, ve které jej může výpočetní stroj dále zpracovávat, ukládat nebo přenášet. [52, 53]



Obrázek 2.3: Průběh analogového a digitálního signálu

U digitálních signálů je proces záznamu výrazně jednodušší, protože již nevyžaduje žádoucí digitalizaci. Digitální data vstupující do záznamníku obvykle v podobě datového toku (data stream) skrz přijímací periferie. Tyto periferie nejsou navrženy pro přímé měření různých fyzikálních veličin, ale pro příjem napěťového signálu reprezentujícího data dle přesně definovaného formátu. Typicky se jedná o digitální rámce, v nichž jsou data zakódována podle určitého přenosového schématu, například rámec může obsahovat jeden start bit, osm datových bitů a jeden stop bit, jako to může být v případě UART přenosu. Tedy jedna periferie může přenášet jak údaje o teplotě, vlhkosti, tak i cokoliv jiného, pokud je dodržen rámec, přičemž povaha přenášených dat závisí na senzorech či zařízeních připojených k této periferii. Přijímaná data jsou tedy v přijímací periferii zpracovávána přímo ve své binární podobě a mohou být bez nutnosti další konverze zpracovávána, přenášena či ukládána. [57]

Digitální záznam poskytuje mnoho výhod oproti svému analogovému protějšku. Primárně je to jeho flexibilita a efektivita při zpracování, ukládání a přenosu dat. Digitální data lze snadněji kopírovat, přenášet, zpracovávat a ukládat, což je pro záznamové systémy zásadní. Díky digitálnímu záznamu můžeme dnes i jednoduše analyzovat data v porovnání s analogovým záznamem. Významnou výhodou digitálního záznamu je také vyšší přesnost, neboť odstraňuje chyby způsobené lidskou interpretací a umožňuje zachycení i těch nejmenších odchylek. [27]

K digitálnímu záznamu je však vyžadován spolehlivý zdroj elektrické energie, případně i stabilní připojení k síti, což může výrazně omezit jeho využití v odlehlych nebo energeticky nestabilních lokalitách. Zároveň je citlivý na výpadky napájení, selhání hardwaru či softwaru a v neposlední řadě i na kybernetické hrozby, při kterých může dojít ke ztrátě či poškození všech uložených dat. [27]

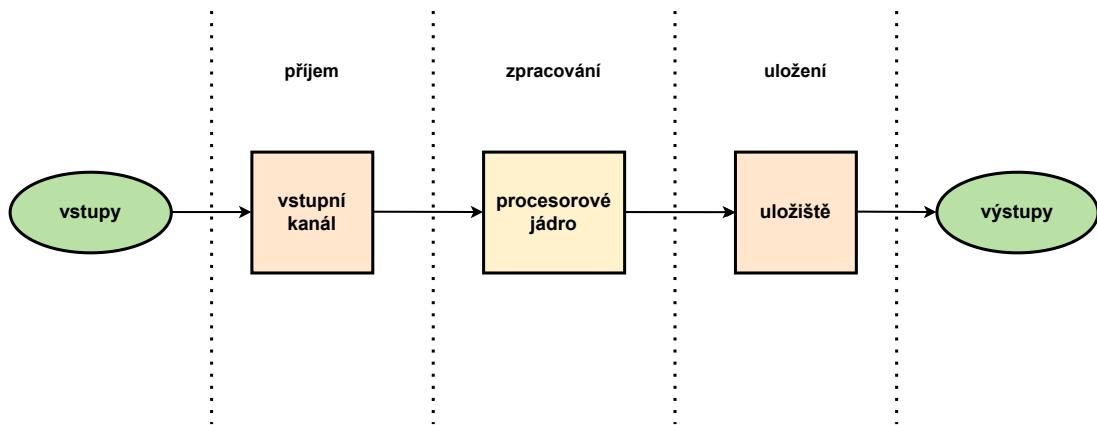
### 2.3.2 Digitální záznamník

Digitální záznamník je dedikované zařízení nebo softwarový program určený ke sběru, zpracování a ukládání dat ve formě digitálního záznamu. Na základě analýzy běžně používaných digitálních záznamníků lze jejich architekturu zpravidla členit do tří základních komponent, kterými jsou vstupní kanál, procesorové jádro a úložiště (viz. obrázek 2.4). [51, 52, 54]

Prvním klíčovým prvkem digitálního záznamníku je vstupní kanál, kterým může být periferie či rozhraní, které slouží ke sběru vstupních dat. Vstupním kanálem může tedy být například periferie zprostředkovávající sériovou komunikaci (UART, SPI, I<sup>2</sup>C a další), síťovou komunikaci (Ethernet, Wi-Fi, LoRa a další) případně analogově-digitální převodníky. Může se jednat, ale také o softwarová rozhraní, jako jsou například různá API (Application Programming Interface). [51, 52]

Druhým hlavním prvkem digitálního záznamníku je procesorové jádro (případně procesorová jádra) centrální výpočetní jednotky (CPU), zajišťující zpracování vstupních dat. Procesorové jádro může být součástí mikrokontroléru, mikroprocesoru, tak i stolního počítače, které může mít v tomto případě na starosti jednoduché operace, jako je například přepočet hodnoty z analogově-digitálního převodníku na teplotu podle kalibrační křivky senzoru, přes filtrování šumu a doplňování časových značek k naměřeným vzorkům, až po pokročilé zpracování dat. [36, 51]

Poslední a také jednou z nejdůležitějších všeobecnou částí digitálního záznamníku je úložiště, kde jsou data ukládána pro pozdější přenos a zpracování (post-processing). Volba tohoto úložného prostoru závisí na požadavcích aplikace a rozsahu jejího využití, od osobních "hobby" projektů až po zařízení využívaná ve velkopodnikových prostředích, nasazována ve vysokých počtech. V závislosti na tom lze využít různé strategie, které jsou v koherenci s různými technologiemi od paměťových karet SD a eMMC přes interní RAM či flash paměti až po síťová úložiště, včetně cloudových služeb. V mnoha případech je také využíván hybridní přístup. Nashromžděná data mohou být nejprve ukládána do interní volatilní paměti záznamníku, jakou je třeba RAM úložiště a následně dávkově přenášena na trvalé médium (viz. kapitola 2.4.2) v podobě lokálního či vzdáleného úložiště. Lze také přidat různé mezi-vrstvy pamětí, například feroelektrickou paměť s náhodným přístupem (FeRAM), která je podrobněji rozebrána v kapitole 2.6.2, nebo jiné varianty nevolatilních pamětí s náhodným přístupem, případně další typy pamětí. [50, 51, 63]



Obrázek 2.4: Obecné schéma digitálního záznamníku

Digitální záznamník poskytuje výstupy, obvykle jimi jsou organizovaná data, která mohou být dále analyzována, vizualizována nebo zpracovávána v rámci stejného systému, po případě i jinými systémy. Jakou podobu mají výstupní data, tedy jaký je jejich formát, opět závisí na konkrétních požadavcích aplikace. Jedním z požadavků je volba dle typu úložiště. Jedná-li se o lokální úložný prostor, například paměťovou kartu, využívají se velmi často typy formátů, jako třeba formát prostého textu (plain text) nebo binární formy. Datatabázová řešení typicky využívají formáty vycházející z relační nebo objektové reprezentace dat. Cloudová řešení naopak nabízejí daleko širší výběr, lze využít jak zmíněný prostý text, i další textové formáty, také je možné využít objektové, relační formáty a další. Dále záleží, jakým způsobem bude proveden přenos dat ze záznamníku dat. Pokud bude využita síťová komunikace, třeba pomocí MQTT či HTTP, je vhodné data uspořádat do serializované podoby, zatímco při zvolení přenosu přes USB může být vhodnější data organizovat do souborů s textovým formátem. Důležitou roli hrají i požadavky na následný post-processing a interpretaci dat v jiných systémech či aplikacích k tomu určeným. Podobu výstupních dat digitálního záznamníku tedy ovlivňuje celá řada faktoriů. [16, 48, 49, 51, 65]

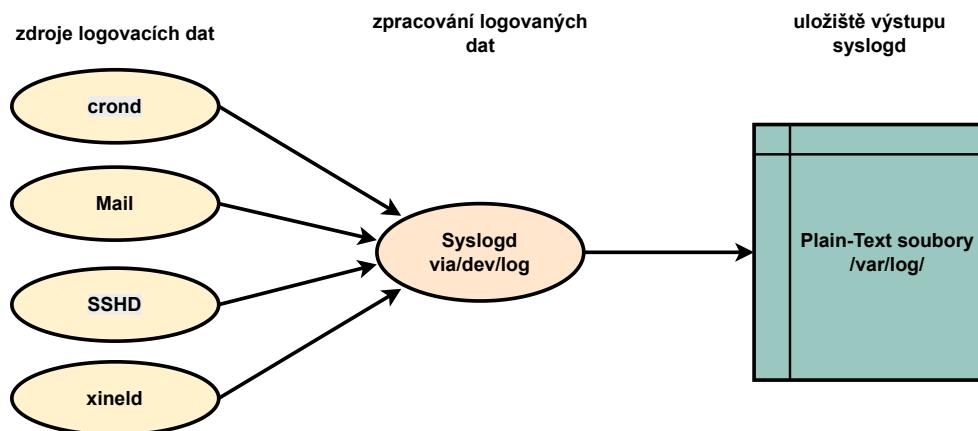
### 2.3.3 Digitální záznam v počítačovém systému

Digitální záznamníky lze implementovat na počítačových systémech, jako jsou osobní počítače či servery, v podobě softwarových řešení, sloužících ke sběru, zpracování a potenciální archivaci dat. Tyto systémy obvykle také vycházejí ze struktury obecného digitálního záznamníku, popsaného v kapitole 2.3.2.

Vstupní data mohou například pocházet z periferií, jako jsou například síťová karta či sériové porty, ale mohou také pocházet například z "pseudo" zařízení obsahujících stavy aplikací běžících na daném počítači nebo z "pseudo" souborů obsahujících systémové informace.<sup>1</sup> Tato data lze obecně získat pomocí speciálních API (například kernel API, libprocps,...) nebo v některých případech i pouhým čtením příslušných souborů, jako je to v případě souborů uložených v adresáři `proc/`. Procesor, který tato data přijímá, je obvykle nejen zpracovává, ale může je i určitým způsobem vyhodnocovat, jelikož disponuje dostatečným výpočetním výkonem pro pokročilé operace. Výsledkem těchto úkonů procesoru nad daty bývají informace o aktuálním stavu sledovaného systému, které lze využít k monitorování a dalším rozhodovacím procesům. [32, 41]

Podle požadavků aplikace a jejího zaměření se liší i způsob, jakým jsou data uchovávána. Některé záznamníky ukládají data přímo na trvalé úložiště, jiné záznamníky si naopak tato data dočasně ukládají do operační paměti RAM, kde tato zaznamenaná data slouží k sledování aktuálního nebo krátkodobého stavu. Pro sledování dlouhodobého stavu je pak možné tato zaznamenaná data v paměti RAM uložit či exportovat na lokální, popřípadě vzdálená dlouhodobá úložiště, a to ve formátech relevantních dané aplikaci. [9, 41]

Za záznamník v počítačovém systému lze považovat například daemon Syslogd v systému Linux (viz. obrázek 2.5), který po desetiletí zajišťoval správu systémových záznamů. Typicky procesy a aplikace odesílají logovací zprávy do pseudo-zařízení `/dev/log`, kde je přijímá právě syslogd. Ten tyto zprávy dále zpracovává a na základě přiložených metadat, například časových značek, informací o původu a prioritě zprávy a dalších částí metadat je ukládá do příslušných textových souborů v adresáři `/var/log/`. [41]



Obrázek 2.5: Tok logovacích dat od aplikací a procesů přes Syslogd [41]

<sup>1</sup>Mezi zmíněné "pseudo" soubory lze zařadit `/proc`, obsahující systémové informace za běhu, jakými jsou například systémová paměť, připojená zařízení, konfigurace hardwaru,...

Je nutné si uvědomit, že tyto záznamníky jsou především implementovány na strojích s relativně výkonnými hardwarevými prostředky, což je činí mnohdy až překvalifikovanými pro pouhé monitorování a představují spíše doplněk, zatímco hlavním účelem stroje může být například vývoj či provozování aplikace či služby. Přičemž tyto záznamníky mohou sloužit například k monitorování zmíněného síťového provozu, monitorování procesů běžících na daném stroji, pokročilé sériové záznamníky a terminály, či záznamníky speciálních rozhraní, jako je například HID (Human Interface Device).<sup>2</sup> [6, 41, 57]

Před implementací či použitím takového záznamníku je tedy vhodné si promyslet účel záznamníku. Klíčovou otázkou je, zda bude pro záznam nezbytné využít výpočetně výkonné zařízení (jakým je například stolní počítač), a zda zařízení bude sloužit výhradně k pořizování záznamů, nebo bude využito i k jiným úlohám. Jako alternativu k digitálnímu záznamníku implementovanému pro počítačový systém lze totiž využít záznamník v podobě dedikovaného zařízení (viz. kapitola 2.3.4).

### 2.3.4 Digitální záznam na platformě mikrořadiče

Digitální záznamníky nemusí být nutně implementovány na výkonných počítačových systémech, ale mohou být také realizovány jako vestavěné systémy postavené na mikrokontrolerech (MCU). Na rozdíl od záznamníků realizovaných na PC jsou však zaměřeny na oblasti, kde je potřeba určitým způsobem sbírat a ukládat data s důrazem na malou spotřebu energie, snadnou přenosnost a mnohdy nižší pořizovací cenu. Proto své uplatnění nacházejí v průmyslové automatizaci, IoT aplikacích, zdravotnických zařízeních a dalších oblastech. [48, 51]

I tyto záznamníky lze obecně složit ze tří základních komponent obecného záznamníku (viz. obrázek 2.6) představené v kapitole 2.3.2, které je možno libovolně rozšířit o další funkční celky, jako je například zálohované napájení, signalizace stavu záznamníku. Data vstupují do záznamníku prostřednictvím vstupního kanálu, která mohou pocházet z různých senzorů, ať už analogových či digitálních (tedy teplotních čidel, akcelerometrů nebo proudových snímačů a dalších) nebo z jiných pozorovaných zařízení. Vstupním kanálem mohou být například periferie klasických sériových komunikačních rozhraní (UART, SPI, I<sup>2</sup>C či I<sup>3</sup>C,...), bezdrátových rozhraní v podobě Wi-Fi, Bluetooth, ale může jím být například analogově-digitální převodník.<sup>3</sup> Tyto vstupní periferie mohou být přímo integrovány v mikrokontroléru, nebo mohou být připojeny externě ve formě samostatných modulů, které komunikují s MCU prostřednictvím již standardních rozhraní. [52]

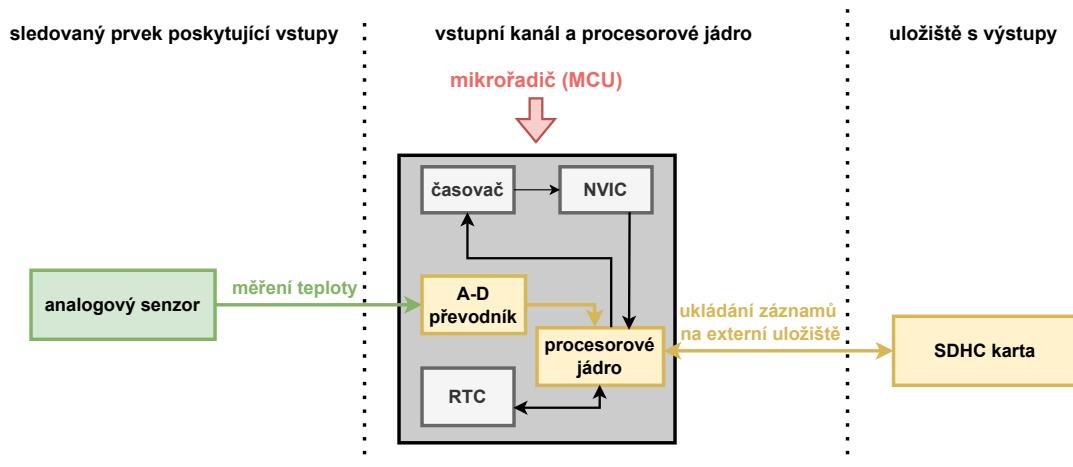
Vstupní data jsou zpracována procesorovým jádrem mikrokontroléru, které zajišťuje jak základní, tak i pokročilé výpočetní operace v reálném čase. Výběr vhodného procesoru hraje zásadní roli, protože celková účinnost systému závisí především na výsledcích mikrokontroléru. Při výběru procesoru je důležité zohlednit jeho architekturu (typicky RISC nebo CISC), velikost paměti, počet a typ I/O pinů, pracovní frekvenci a energetickou efektivitu, popřípadě další parametry. [36, 52]

Po zpracování vstupních dat jádrem procesoru následuje jejich uložení do paměťového média. K ukládání dat se běžně využívá lokální úložiště tvořené sekundárními nevolatilními pamětími, která zaručují perzistence dat i po vypnutí záznamníku. Typickými zástupci těchto pamětí jsou například různé typy SD karet, USB flash disky a další technologie využívající

<sup>2</sup>Tyto záznamníky nemusí sloužit pouze k monitorování samotného stroje na kterém jsou spuštěny, ale mohou také sledovat další připojená zařízení (a periferie), jakým je například sledování komunikace s mikrokontrolérem prostřednictvím sériového terminálu, tiskárny,...

<sup>3</sup>Vstupních kanálů je vícero, zde jsou zmíněna jen ty nejzákladnější.

například technologií NAND nebo NOR flash paměti. Kromě lokálního paměťového média je však možné zvolit přístup se vzdáleným uložištěm v podobě databáze či cloudového systému. Volatilní paměti, jako je například RAM, se v těchto aplikacích rovněž využívají, avšak výhradně pro krátkodobé uchování dat. V některých případech, například pokud jsou nevolatilní paměti organizovány do bloků (viz. 2.4.2), nebo je nutné data nějakým způsobem zpracovat před jejich uložením. [50, 52]



NVIC (Nested Vector Interrupt Controller) - řadič přerušení s podporou priorit a vnoření  
RTC (Real-Time Circuit) - obvod reálného času

Obrázek 2.6: Příklad architektury digitálního záznamníku pro měření teploty

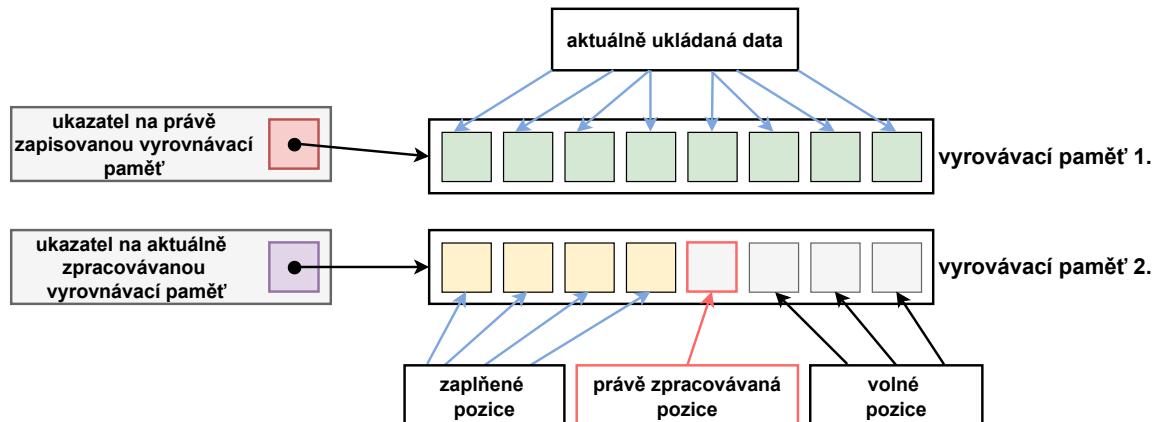
## 2.4 Klíčové koncepty digitálních záznamníku

Digitální záznamníky implementované v podobě dedikovaného zařízení na platformě MCU, dokáží poskytnout dobrý kompromis mezi cenou a výkonem. Tyto záznamníky, na rozdíl od záznamníků realizovaných na výkonnějších platformách, jako jsou osobní počítače nebo jednodeskové počítače, čelí určitým omezením daným dostupnými hardwarovými prostředky. Proto lze využít některé z optimalizačních technik, zaměřených na vhodnější využití dostupných zdrojů zařízení, které mohou vést k vyššímu výkonu, lepšímu využití dostupného paměťového prostoru nebo například i k lepší spotřebě.

### 2.4.1 Vícenásobná vyrovnávací paměť (multiple-buffering)

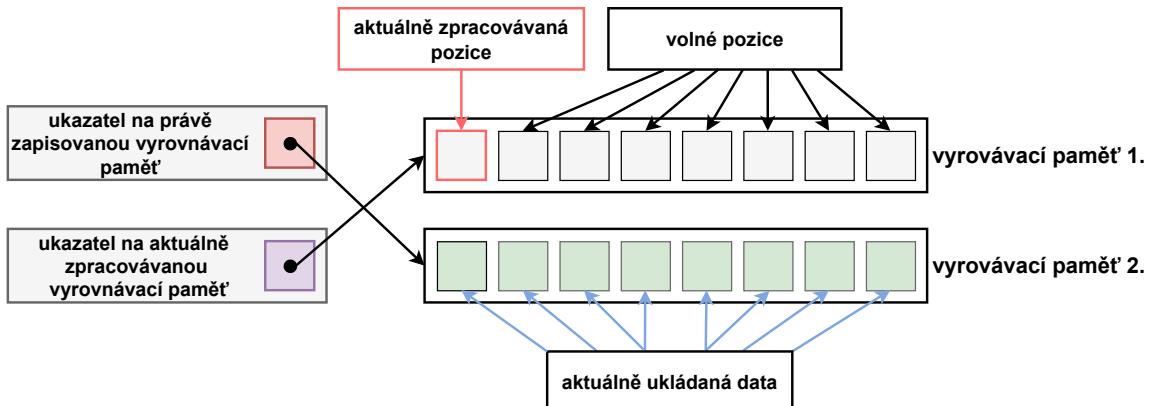
Jedním z častých konceptů využívaných v implementaci digitálních záznamníků je použití vícenásobné vyrovnávací paměti. Tento koncept je převážně známý díky algoritmu využívaným v oboru počítačové grafiky. Grafický čip musí zpracovat velké množství dat za krátký časový úsek, proto algoritmus zpracování dat využívá dvě vyrovnávací paměti, přední vyrovnávací paměť, takzvaný Front Buffer, jež je využívána pro zobrazení aktuálního snímku, a zadní vyrovnávací paměť, Back Buffer, ve které čip připravuje nový obsah. Jakmile je nový snímek kompletní, vyrovnávací paměti se vzájemně prohodí a výsledný obsah tak může být plynule vykreslen bez artefaktů a trhání. [60, 64]

Obdobný mechanismus se využívá i v digitálních záznamnících, kde slouží k zajištění kontinuálního sběru dat bez výpadků (viz. obrázek 2.7). Zatímco Back Buffer přijímá nová digitální data ze vstupní periferie a popřípadě je zpracovává, Front Buffer je současně ukládán na úložné médium. Jakmile je Back Buffer naplněn daty a obsah Front Bufferu je uložen, dojde k jejich prohození. [60, 64]



Obrázek 2.7: Schéma principu práce s vícenásobnou vyrovnávací pamětí – sběr dat

Důležitou vlastností tohoto přístupu je jeho nízká operační režijní náročnost. Plynulý chod zpracování dat je zajištěn bez nutnosti fyzického přenosu obsahu mezi vyrovnávacími paměti. Místo toho se využívají ukazatele (pointery), které směřují na počáteční adresy jednotlivých bufferů. Jakmile je Back Buffer naplněn, ukazatele se prohodí, čímž dojde k záměně rolí jednotlivých bufferů, buffer mající roli Back Buffer je přiřazena role Front Buffer a naopak bufferu s původní rolí Front Buffer je přidělena role Back Buffer, jak je zobrazeno na následujícím obrázku 2.8. [60, 64]



Obrázek 2.8: Schéma principu práce s vícenásobnou vyrovnávací pamětí – prohození ukazatelů

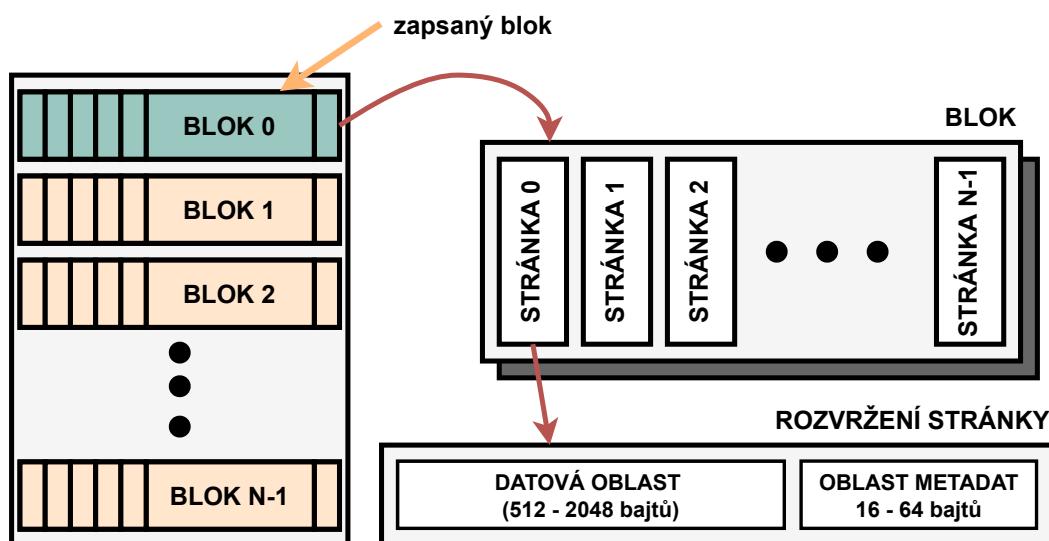
Tato metoda nachází významné uplatnění zejména v systémech pracujících v reálném čase, kde dochází k příjmu velkého objemu dat v krátkých časových intervalech. Použití více vyrovnávacích pamětí zvyšuje celkovou propustnost systému a pomáhá předcházet vzniku úzkých míst nebo ztrátě dat. [60, 64]

Naopak použití vyrovnávacích pamětí obecně zvyšuje nároky na systémové prostředky a tím i celkové náklady. Softwarová implementace bufferů vyžaduje dostatečnou kapacitu volatilní paměti RAM, což může být problém zejména u starších nebo paměťově omezených vestavěných zařízení. Alternativně v takovýchto případech lze využít hardwarové paměťové buffery, které však dále navyšují cenu zařízení. [60, 64]

## 2.4.2 Dávkové ukládání (batch saving)

Další princip, využívaný v implementacích digitálních záznamníků, souvisí s typem uložišť, na které jsou získaná data zaznamenávána. Data jsou standardně dlouhodobě ukládána na některý z typů nevolatilních pamětí, které umožňují uchování dat i po odpojení napájení. Takovými uložišti jsou například NAND či NOR flash paměti (viz. obrázek 2.9). NAND flash paměti jsou fyzicky organizovány do bloků, které se dále dělí na menší jednotky nazývané stránky. Zápis a čtení dat probíhá na úrovni jednotlivých stránek, zatímco mazání je možné pouze na úrovni celých bloků. Naproti tomu NOR flash paměti umožňují přímý náhodný přístup k jednotlivým paměťovým buňkám a zápis jednotlivých slov, avšak mazání dat probíhá stejně jako u NAND pamětí vždy po celých blocích. [47, 67]

U dávkového ukládání se využívá podobná zásada jako u mazání dat ve výše zmíněných nevolatilních pamětech. Data jsou shromažďována ve volatilní paměti, například RAM, a teprve po naplnění určitého objemu (celého bloku či jeho násobku) dojde k jejich zápisu na dlouhodobé paměťové médium. Tím se omezuje počet zápisových operací a zvyšuje životnost paměťového média. Současně dávkové ukládání zrychluje proces ukládání dat, protože místo častých jednotlivých zápisů dochází k efektivnějšímu hromadnému zápisu většího objemu dat najednou. [47, 52, 67]

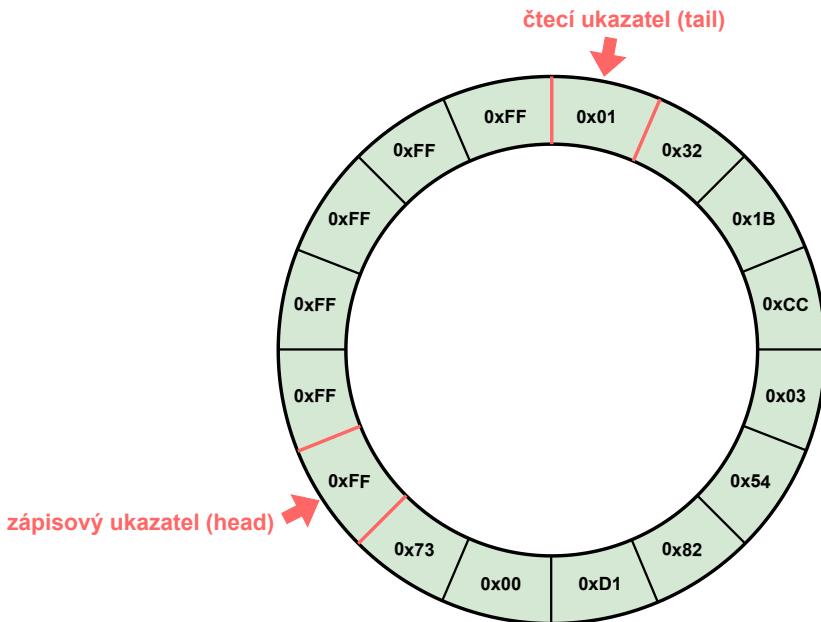


Obrázek 2.9: Příklad organizace nevolatilní paměti typu NAND [55]

### 2.4.3 Cirkulární buffer

Cirkulární buffer (circular buffer), někdy také označovaný jako kruhový nebo cylindrický buffer, je datová struktura, která funguje na principu FIFO fronty (First-In, First-Out), kde je paměť organizována jako logický kruh. Tento přístup je často využíván k řešení problému typu jednoho producenta a konzumenta (producer-consumer problem), kde jedno vlákno je zodpovědné za produkci dat, a druhé vlákno je zodpovědné za konzumaci dat. Například ve vestavěných zařízeních, jedním z vláken je rutina obsluhy přerušení, která čte data ze senzoru, a druhým vláknem je hlavní smyčka události, která data určitým způsobem zpracovává. [40]

Princip činnosti cirkulárního bufferu (viz. obrázek 2.10) spočívá v použití dvou ukazatelů, jedním z nich je zápisový ukazatel (head) a druhým z nich je čtecí ukazatel (tail). Ukazatel head vždy směruje na pozici, kam bude zapisován následující prvek, zatímco ukazatel tail ukazuje na pozici, ze které bude čtena následující hodnota. Pokud ukazatel head dosáhne konce pole, vrací se na jeho začátek, čímž je zajištěna kruhová povaha struktury. Při plném bufferu pak lze zvolit dvě strategie, buď mohou být přepsána nejstarší data, nebo lze nová data odmítout, přičemž výběr závisí na konkrétní aplikaci. [40, 46]



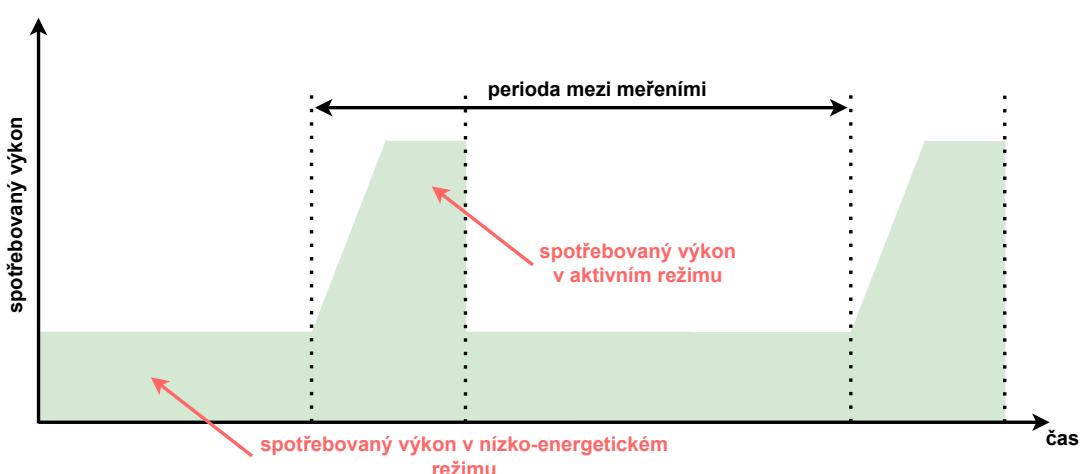
Obrázek 2.10: Cirkulární vyrovnávací paměť

Z hlediska časové složitosti nabízí cirkulární buffer konstantní časovou složitost  $\mathcal{O}(1)$  pro základní operace typu zápis (enqueue) a čtení (dequeue). Tato efektivita vyplývá z absence potřeby přesouvat prvky v paměti při zápisu a čtení dat, místo toho se pouze inkrementují ukazatele s využitím operace modulo. Pokud jde o prostorovou složitost, velikost cirkulárního bufferu je určena předem, zpravidla jde totiž o staticky alokované pole, což odpovídá složitosti  $\mathcal{O}(n)$ , kde  $n$  je maximální počet prvků, které může buffer pojmet. [59]

#### 2.4.4 Nízkoenergetické režimy

Energetická efektivita je jedním z klíčových parametrů obecně vestavěných zařízení, tedy i digitálních záznamníků implementovaných na platformě MCU, zejména pokud jsou napájeny z baterií či jiných omezených zdrojů energie (například energy harvesting). Minimalizace spotřeby bývá v těchto případech realizována využitím nízkoenergetických režimů (low-power modes), které umožňují zařízení přejít do stavu s minimální energetickou náročností během nečinných period. V praxi mnoha digitálních záznamníků nemusí provádět záznam dat nepřetržitě. Například záznamník teploty může provést měření, uložit naměřenou hodnotu, přejít do režimu nízké spotřeby a po uplynutí definovaného časového intervalu nebo při výskytu speciální události přejít zpět do aktivního režimu. [62]

Průběh takového cyklického chování spotřeby mikrokontroléru, kde se střídají fáze měření a spánku s pravidelnou periodou měření teploty, je znázorněn na obrázku 2.11 níže.



Obrázek 2.11: Graf znázorňující dynamiku spotřeby mikrokontroléru v průběhu času při využití aktivního a nízkoenergetického režimu [62]

Ačkoliv nízkoenergetické režimy přinášejí značné úspory energie a jsou mnohdy nezbytné pro zařízení napájená z baterií, u digitálních záznamníků zaznamenávající velký objem dat mohou představovat významná omezení. Tyto režimy sice snižují energetickou náročnost systému, avšak zároveň omezují schopnost mikrokontroléru rychle reagovat na události. Spánkové stavy, které minimalizují spotřebu energie, často vedou k delšímu zpoždění při probuzení a nižší dostupnosti kritických periferií. V aplikacích, kde je vyžadována okamžitá odezva na externí podněty nebo nepřetržité zpracování velkého množství dat, může tento koncept negativně ovlivnit spolehlivost a efektivitu záznamníku. [22]

V těchto případech je proto nutné zvážit provozní podmínky a očekávanou dostupnost systému. Pokud záznamník pracuje s velkým datovým tokem a má možnost být připojen po dobu záznamu stále k externímu napájení, může být výhodnější upustit od implementace nízkoenergetických režimů a místo toho optimalizovat architekturu systému pro nepřetržitý provoz s důrazem na výkon a rychlou odezvu. [62]

## 2.5 Koncepty vkládání synchronizačních značek do záznamu

Součástí funkce digitálního záznamníku může být také vkládání synchronizačních značek do zaznamenaných dat. Synchronizační značky poskytují časový kontext jednotlivým datům a jejich přítomnost může být důležitá při jejich následném zpracování a analýze. Existuje několik přístupů, jak vkládání synchronizačních značek v systému záznamu implementovat, přičemž konkrétní řešení závisí na charakteru zaznamenaných dat a způsobu jejich vzniku. [51, 57]

Jeden z možných přístupů spočívá ve vkládání časových značek při výskytu jednotlivých událostí, například překročení prahové hodnoty, uplynutí pevně definovaného intervalu nebo výskyt externího podnětu, jako je přerušení od senzoru. Časová značka se v těchto případech přidává pouze v okamžiku výskytu takové události do zaznamenaných dat. [51, 54]

Následující ukázka 2.1 v jazyce C demonstruje jednoduchý princip vkládání synchronizačních značek během pravidelného výskytu události, při kterém je potřeba změřit teplotní veličinu. Záznamník každých například pět minut změří aktuální teplotu, získá aktuální čas ze systémového časovače a uloží hodnotu ve formátu, kde je časová značka uvedena v hranatých závorkách před naměřenou hodnotou.

```
while (1)
{
    float teplota = MeasureTemperature();
    uint32_t timestamp = GetTimestamp();
    printf("[%d] Teplota: %.2f \n", timestamp, temperature);
    DelayMinutes(5);
}
```

Výpis 2.1: Vkládání časové značky zaznamenané události

V případě, že digitální záznamník slouží například ke záznamu sériové komunikace, nemusí být okamžik vkládání časové značky jednoznačně určen jednotlivými událostmi, jakým je například přerušení. Je proto nutné implementovat mechanismus, který detektuje vhodný okamžik pro vložení synchronizační značky do přijímaného datového toku. [57]

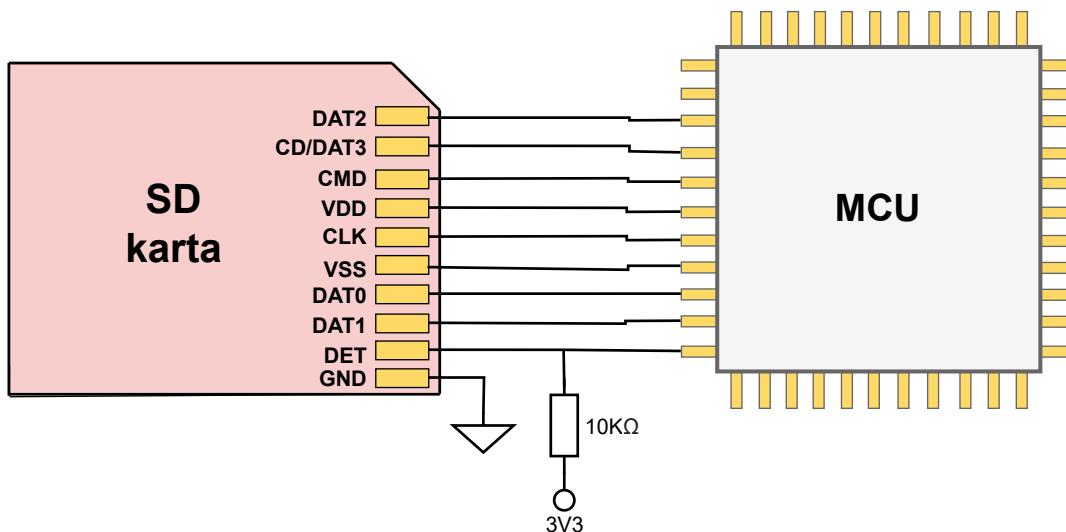
Jedním ze způsobů může být sledování časových prodlev mezi přijatými daty. Pokud mezi dvěma bloky dat přijde prodleva delší než stanovený interval (například 50 ms), počítá se tato prodleva za ukončení bloku dat, a následně se na začátku dalšího bloku vloží nová časová značka. Další možností je vkládání synchronizačních značek po fixní velikosti přijatých dat, například každých 64 bajtů. Alternativně lze využít dohodnutý ukončovací znak nebo sekvenci, která signalizuje konec jednoho logického bloku dat. Typickým příkladem může být znak konce řádku, nulový bajt nebo definovaná specifická posloupnost bajtů. Jakmile je taková sekvence detekována, je po její zpracování připojena časová značka, případně je vložena před začátek následujícího bloku. [57]

## 2.6 Způsoby ukládání dat

Záznamníky dat musí být navrženy tak, aby umožnily spolehlivé ukládání získaných dat, které by mělo být efektivní ve smyslu rychlosti a šetrné pro zvýšení životnosti úložiště. V této kapitole jsou popsány tři různé způsoby ukládání dat, přímý zápis na lokální úložiště, ukládání prostřednictvím mezivrstvy s FeRAM pamětí a využití vzdálených úložišť. Každá z těchto metod má své specifické výhody a omezení, které určují její vhodnost pro konkrétní aplikace.

### 2.6.1 Přímý zápis na permanentní uložiště

Přímý zápis na permanentní uložiště představuje nejpřímější metodu ukládání dat. V tomto případě jsou zaznamenaná data ihned zapisována na nevolatilní paměťové médium, jako je SD karta, eMMC, USB flash disk nebo jiná lokální paměťová uložiště. Přenos dat na tato média obvykle probíhá prostřednictvím specializovaných rozhraní, jakými jsou například SDIO nebo SPI pro SD karty, MMC pro eMMC paměti a USB Host Controller pro připojení USB flash disků. Přičemž použité rozhraní může významně ovlivnit rychlosť, jakou jsou data zapisována a případně čtena. [50, 57]



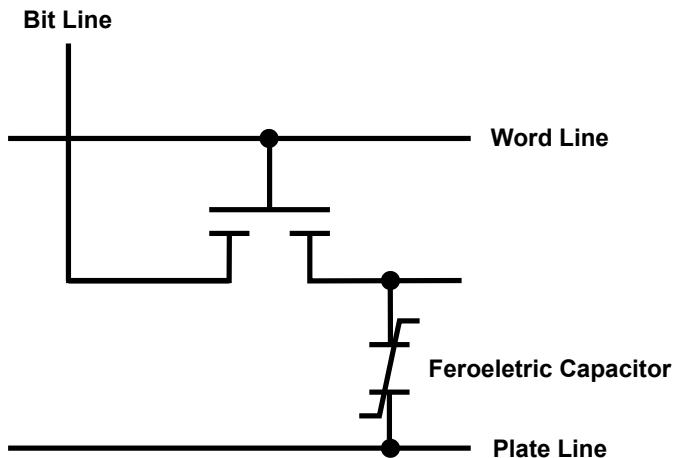
Obrázek 2.12: Přímý zápis na permanentní uložiště s SD kartou za pomocí čtyř pinové datové sběrnice

Tento způsob eliminuje potřebu mezivrstvy mezi záznamníkem a úložištěm, čímž se minimalizuje latence a zjednoduší celková implementace. Naopak nevýhodou tohoto principu může být častý zápis na paměťové médium. Proto, aby se předešlo nadmernému opotřebení uložiště a zvýšila se efektivita zápisu, využívá se mnohdy současně metoda dávkového ukládání, která již byla zmíněna v kapitole 2.4.2. Data jsou krátkodobě uložena ve volatilním uložišti, a jakmile jich je nashromážděno dostatek, tak jsou přepsána do dlouhodobé nevolatilní paměti. To ale přináší i nové problémy, kdy hodnoty uložené v neperzistentním uložišti jsou vystavena riziku ztráty v případě ztráty napájecího napětí. [52]

## 2.6.2 Zápis na permanentní uložiště přes mezivrstvu s FeRAM paměti

Alternativní volbou k přímému zápisu na permanentní uložiště je využití mezivrstvy založené na paměti FeRAM (Ferroelectric Random Access Memory). FeRAM je nevolatilní paměť, jež kombinuje výhody rychlé volatilní RAM paměti a perzistentního úložiště.

Ve feroelektrické RAM paměti jsou data ukládána pomocí změny polarizace feroelektrického materiálu v paměťové buňce. Jednotlivé buňky se skládají podobně jako je tomu u dynamické RAM (DRAM), z jednoho tranzistoru a jednoho feroelektrického kondenzátoru (1T-1C). Na rozdíl však od DRAM, kde je informace uchovávána jako elektrický náboj v lineárním dielektriku, FeRAM využívá feroelektrický materiál, jakým je třeba titaničitan olovnatý (PZT), který vykazuje hysterezní chování. Jakmile je aktivní elektrické pole, dipóly se v krystalové mřížce přeuspořádají do jednoho ze dvou stabilních stavů odpovídajících binárním hodnotám nula či jedna a tento stav zůstává zachován i po odeznění elektrického pole. K samotnému čtení a zápisu do konkrétní paměťové buňky jsou využívány tři hlavní signálové linky Word Line a Bit Line určující konkrétní buňku a Plate Line, která vytváří elektrické pole potřebné k přepolování feroelektrického materiálu během zápisu, jak je znázorněno na obrázku 2.13. [43]



Obrázek 2.13: Struktura 1T-1C feroelektrické RAM paměti [43]

FeRAM lze v dedikovaném digitálním záznamníku využít jako takzvanou mezivrstvu neboli vyrovnávací paměť, pomocí které lze optimalizovat zápsky na konečné dlouhodobé uložiště. Jak jsou tedy data záznamníkem postupně sbírána, tak mohou být postupně či po blocích zapisována do této mezivrstvy. Pokud je následně vyrovnávací paměť FeRAM dostatečně zaplněna, její obsah je načten do RAM paměti mikrokontroléru a následně zapsán na trvalé uložiště, a tento cyklus se opakuje. [43, 47]

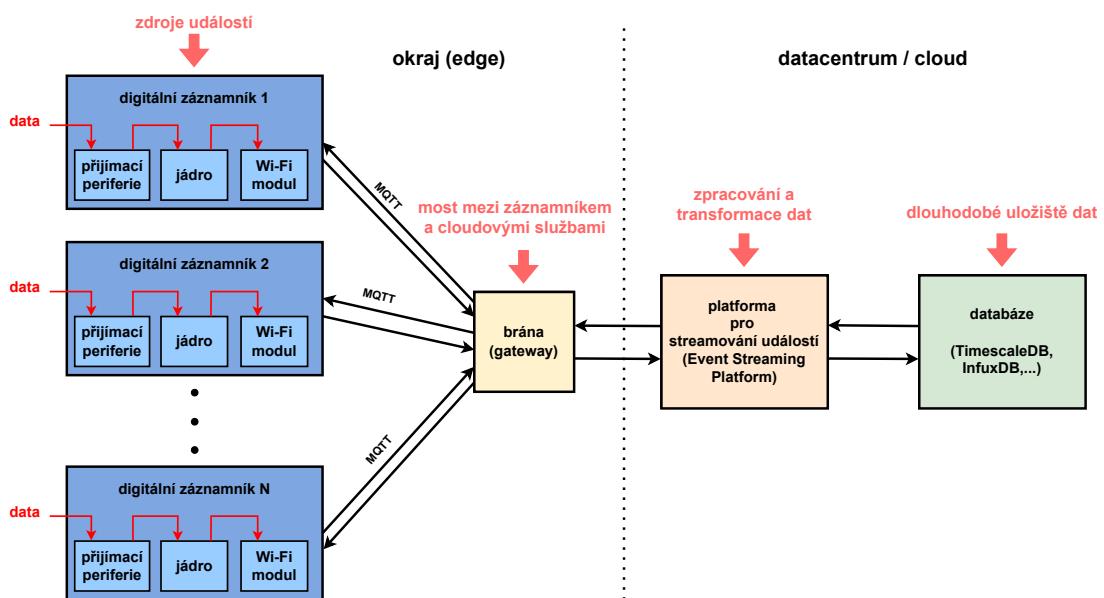
Použití FeRAM jako vyrovnávací paměti přináší několik výhod. Jedním z hlavních přínosů je snížení opotřebení hlavního uložiště, eliminován je totiž častý zápis po malých blocích dat, který zbytečně opotřebovává koncová nevolatilní uložiště typu flash, která mají omezený počet přepisovacích cyklů. Další výhodou je velice rychlý zápis oproti flash pamětem, obvykle trvá zápis na FeRAM v řádu nanosekund, což je rádově rychlejší než srovnávané flash paměti, na kterých zápis trvá typicky v řádu mikrosekund až milisekund. FeRAM je navíc nevolatilní, což znamená, že i v případě výpadku napájecího napětí zůstávají zazna-

menaná data na feroelektrickém uložišti zachována, čímž se eliminuje potřeba dodatečných opatření k ochraně dat, jako jsou záložní baterie nebo kondenzátory. [43, 47]

### 2.6.3 Zápis na vzdálené uložiště

Pro dlouhodobé uložení dat lze také využít vzdálená uložiště, jakými jsou databázová či cloudová uložiště. Lze tak využít jednotného uložiště pro velké množství digitálních záznamníků a eliminovat potřebu lokálních, nevolatilních paměťových médií. Zmiňované jednotné uložiště může být jak centrální server, tak i distribuovaná síťová soustava, umožňující uložení daleko většího množství dat, než může nabídnout lokální uložiště. [42, 48]

V praxi dedikovaný záznamník, který data bud přímo zpracovává, nebo je dočasně uchovává ve volatilní paměti, využívá síťové rozhraní k jejich přenosu do vzdáleného úložiště. Přenos probíhá obvykle prostřednictvím aplikačních síťových protokolů postavených nad transportním protokolem TCP, jakým je například MQTT, nebo nad protokolem UDP, nad kterým je postaven třeba protokol CoAP. Každý z uvedených zástupců poskytuje trošku odlišnou funkcionality. MQTT je vhodnější pro komunikaci, kde odesílatel a příjemce nejsou synchronizováni a dialog probíhá asynchronně. Vysílatel tedy odesílá data a ostatní zařízení tato data mohou začít přijímat. Zato CoAP zase prosazuje formát komunikace typu dotaz-odpověď, kterým mimikuje HTTP. [31, 36, 48]



Obrázek 2.14: Architektura pokročilého systému digitálních záznamníků s clouдовým úložištěm a platformou Kafka [70]

Databázové servery, na které mohou být data ukládána, mohou být postaveny na různých technologiích v závislosti na typu dat a požadavcích na jejich zpracování. Často jsou využívány systémy, které umožňují pracovat s daty ve formátu časových řad, což jsou sekvence datových bodů zaznamenávaných ať už v pravidelných či nepravidelných časových intervalech. Na taková data může být tedy vhodné využít například InfluxDB nebo TimescaleDB.<sup>4</sup> Clouдовá řešení pak zpravidla nabízejí různé typy úložišť od objektových (např.

<sup>4</sup>Možné je také zvolit relační databáze, ale ty jsou u digitální záznamníků méně časté.

pro ukládání datových souborů a binárních objektů) přes databázové služby až po streamovací platformy. Kromě samotného ukládání umožňují také přidat analytickou vrstvu pro zpracování dat v reálném čase, díky které lze reagovat na aktuální události. K tomuto účelu pak lze využít třeba streamovací platformy, jakými jsou Apache Kafka (viz. obrázek 2.14 a AWS Kinesis. [42]

Výhodou tohoto přístupu je především možnost centralizovaného ukládání a zpracování dat, to se hodí při záznamu dat z velkých skupin digitálních záznamníků, v takovémto případě může být obtížné manuálně kontrolovat všechna zařízení a postupně z nich extrahouvat získaná data. Další výhodou je schopnost systému reagovat na aktuální stav prostředí nebo na základě dat odvozovat vhodná opatření. Tento princip lze efektivně využít například v chytrých domácnostech nebo průmyslových provozech, kde lze automaticky upravovat provoz zařízení, jako je klimatizace, vytápění či osvětlení, na základě hodnot získaných ze senzorů. [42, 48, 51]

Přístup se zápisem na vzdálené uložiště hodí pro digitální záznamníky operující ve známých prostředích, kterým je třeba zmiňovaná chytrá domácnost či továrna, jelikož je potřeba zaručit stabilní připojení k síti. Pokud by měl záznamník v různých lokalitách napříč světem, bylo by nutné jej na každém novém místě připojit k Wi-Fi síti či Ethernetu, nebo by jej bylo nutné koncipovat jako digitální záznamník se SIM kartou, pomocí které by byl zajištěn přístup k mobilní síti. Dále je důležité mít i koncipovanou komplexní infrastrukturu, v níž je zajištěn bezpečný přenos dat. Pro tento síťový přenos je třeba přidat další úroveň zabezpečení, která zajistí autentizaci, šifrování, integritu dat a případně další bezpečnostní prvky. S bezpečností souvisí také nezbytnost, aby zařízení poskytovalo zdroje pro běh plnohodnotného TCP/IP modulu (TCP/IP stack), díky kterému bude možné využít bezpečnostní prvky, jako je například SSL/TLS. [42, 48, 51, 56]

## Kapitola 3

# Návrh digitálního záznamníku

### 3.1 Existující řešení digitálních záznamníků

Digitální záznamníky dat jsou běžně využívány v praxi a existuje řada produkčních řešení zaměřených na záznam různých typů dat. Tato kapitola je věnována popisu některých dostupných řešení, konkrétně záznamníkům zaměřeným na záznam sériových dat, mezi které se bude řadit i výsledné zařízení, které jsem navrhl, implementoval a popsal v této bakalářské práci.

#### 3.1.1 Keeylog AirDrive Serial Logger

Jedním z řešení digitálního záznamníku je AirDrive Serial Logger, vyvinutý společností Keelog. Zařízení AirDrive je moderní digitální záznamník, který umožňuje záznam sériových dat pomocí komunikačního standardu RS-232 či RS-485. Na rozdíl od tradičních řešení, která ukládají data pouze na lokální úložiště, nabízí tento záznamník konektivitu k Wi-Fi síti, čímž umožňuje ukládání dat na vzdálené uložiště (viz. kapitola 2.6.3), kam mohou být data pravidelně odesílána a dále zpracovávána či analyzována. Záznamník AirDrive podporuje konfiguraci prostřednictvím webového rozhraní, přičemž lze tímto způsobem nastavit například přenosovou rychlosť, počet datových bitů, paritu a počet stop bitů. [1]

Společnost Keelog poskytuje vícero variant AirDrive záznamníků, které se liší poskytnutými funkcionalitami. Hlavním rozdílem mezi jednotlivými verzemi je odlišný přístup k získaným datům, základní verze pracuje jako Wi-Fi hotspot, zatímco verze Pro a Max umožňují připojení do existující Wi-Fi sítě a také odesílání e-mailových reportů, časové razítkování záznamů nebo dokonce živé streamování dat. V čem se naopak zmíněné verze nelíší, je velikostí interní paměti, která činí 16 GB, jež je uživatelsky přístupná i jako USB flash disk s rychlosťí až 480 Mbps. Zařízení jako AirDrive nachází uplatnění zejména v průmyslovém monitorování, zpětném inženýrství sériových protokolů, zálohování dat z platebních terminálů nebo sběru dat ze senzorových systémů. [1]

Navzdory svým pokročilým funkcím má AirDrive Serial Logger i několik nevýhod. Jednou z nich je omezení na standardy RS-232 a RS-485, které sice stále nacházejí uplatnění v průmyslových a v řadě dalších systémů, avšak v některých ostatních zařízeních je pro připojení k jiným rozhraním nutné využít sériové převodníky. Druhou limitací je omezení maximální přenosové rychlosti zaznamenaných dat, která může činit nejvýše 230400 baudů. To je například nevyhovující pro monitorování systémů bezdrátového nabíjení společnosti NXP Semiconductors, kde se komunikace probíhá s vyšší komunikační rychlosťí. [1]

### 3.1.2 Anticyclone Systems AntiLog Data Logger Pro

Druhým zmíněným řešením digitálního záznamníku je AntiLog Data Logger od společnosti Anticyclone Systems (viz. obrázek 3.1), který lze klasifikovat jako vysoko výkonné digitální záznamníky určené pro záznam sériových dat v průmyslových a vývojových aplikacích.<sup>1</sup> Podobně jako u řešení od společnosti Keeylog (viz. kapitola 3.1.1) umožňuje data přijímat pomocí standardu RS-232 a také plnohodnotně zaznamenávat obousměrné sériové přenosy s vysokými přenosovými rychlostmi až 921600 baudů. Zařízení umožňuje dlouhodobé zaznamenávání díky podpoře velkokapacitních nevolatilních úložišť až do velikosti 1 TB. AntiLog Data Logger je dostupný ve verzích AntiLog, AntiLog Pro a OEM modul pro přímou integraci do jiných systémů. Nejpokročilejší verze Pro podporuje funkce jako časové razítkování, podpora GNSS/NMEA dat a možnost vícekanálového záznamu, což jej činí vhodnou volbou pro aplikace, kde je potřeba přesné a rozsáhlé monitorování sériových přenosů. [2]



Obrázek 3.1: Anticyclone Anti-Log Pro [2]

Výhodou je také nízká spotřeba<sup>2</sup> a možnosti připojení baterie, které umožňují použití jak ve vnitřních prostředích, tak i v exteriérech, tedy například v přírodě. Záznamník podporuje pokročilé časové razítkování s rezolucí až jednu milisekundu, usnadňující synchronizaci a post-processing dat. Záznam lze rozšířit o měření veličin, jako je teplota, vlhkost či tlak, prostřednictvím podporovaných senzorů komunikujících po sběrnici I<sup>2</sup>C, a to paralelně se záznamem až dvou datových kanálů využívajících standard RS-232. Možné je také propojení až 255 jednotek do jednoho vícekanálového záznamníku, které pak umožňuje komplexní monitorování více zařízení současně. [2]

Hlavním nevýhodou Anti-Log záznamníku je jeho vysoká pořizovací cena, ta se aktuálně pohybuje u zařízení Anticyclone Anti-Log Pro od 229 £ do 366 £. Kromě ceny může být překážkou nutnost použití speciální aplikace AntiTermPro RS-232 terminálový software pro

<sup>1</sup>Společnost Anticyclone Systems nabízí tři varianty této záznamníků, v tomto textu je primárně popsána verze Pro, jež je svými parametry a funkcionalitou nejblíže záznamníkům, které jsou předmětem této bakalářské práce.

<sup>2</sup>Společnost uvádí spotřebu 58 mW při přenosové rychlosti 9600 baudů a při přenosové rychlosti 115000 baudů spotřeba stoupne na 90 mW

konfiguraci a přehrávání zaznamenaných dat, což znesnadňuje práci se zařízením nezkušené obsluze, která přijde do kontaktu s digitálním záznamníkem poprvé. [2]

### 3.1.3 Shrnutí představených řešení

Představená řešení AirDrive a AntiLog reprezentují různé přístupy k problematice digitálního záznamu sériových dat. Navzdory svým výhodám však žádné z uvedených řešení plně neodpovídá požadavkům pro záznam dat společnosti NXP Semiconductors.

Požadovaný digitální záznamník musí zvládat zaznamenávat sériovou komunikaci o rychlosti mezi 230000-400000 baudů, dále je potřeba vkládat časová razítka, jelikož výsledné záznamy musí být možné synchronizovat s dalším systémem. Nezbytně nutné je, aby záznamník disponoval mechanismem prevence proti ztrátě dat při výpadku napájecího napětí, získaná data musí být možné vycist bez nutnosti vyjmutí fyzického média a navíc to vše za rozumnou pořizovací cenu.

Hlavním společným problémem obou představených řešení je prevence ztráty dat při výpadku napájecího napětí, kterou ani jedno řešení neřeší. Společnou nevýhodou záznamníků Keelog a Anticyclone Systems je jejich zaměření výhradně na standard RS-232, což také není vhodné pro použití, digitální záznamník bude použit k monitorování bezdrátového nabíjení využívající kanály UARTu a musel by tak být poskytován společně se záznamníkem převodník. Dalším omezením obou zařízení je nemožnost přizpůsobit formát zaznamenaných dat konkrétním potřebám, neboť výstup je pevně dán výrobcem a nelze jej upravit.

Obě řešení sice nabízejí možnost odesílání dat na vzdálené úložiště, tato funkcionality je však momentálně nevyužitelná. Vývojový tým společnosti NXP Semiconductors, zaměřený na oblast bezdrátového napájení, v současnosti nedisponuje žádnou infrastrukturou pro vzdálené ukládání dat a současně není povoleno získaná data ukládat do externích vzdálených uložišť.

Základní požadavky splňuje pouze záznamník od společnosti Anticyclone, ten je však drahý a pro výčtení dat je nutné mít v počítači nainstalovanou speciální aplikaci, požadavkem na digitální záznamník je také jednoduchá obsluha bez instalace dodatečných aplikací, aby se usnadnila práce operátorům testovacích laboratoří, kteří se záznamníkem přijdou do kontaktu poprvé.

Naopak záznamník od společnosti Keelog je sice cenově dostupnější, ale jeho maximální přenosová rychlosť je omezena pouze na 230400 baudů.

Rozhodl jsem se tedy navrhнуть a implementovat vlastní řešení digitálního záznamníku, které bude možné využít pro záznam digitálních dat z bezdrátových nabíječek a nebude mít výše zmíněné nedostatky.

## 3.2 Výběr vhodné platformy

Shrnutím poznatků z předchozí kapitoly 2 vyplývá, že pro realizaci digitálního záznamníku se jako nevhodnější jeví mikrořadič či jednodeskový počítač, jelikož požadavkem na koncový systém ze strany NXP Semiconductors je přenosnost, jednoduchost použití a nízká cena. K dispozici je široká škála platform, které lze pro tento účel využít, přičemž každá z nich nabízí různé výhody a omezení.

### 3.2.1 NXP FRDM-MCXN947

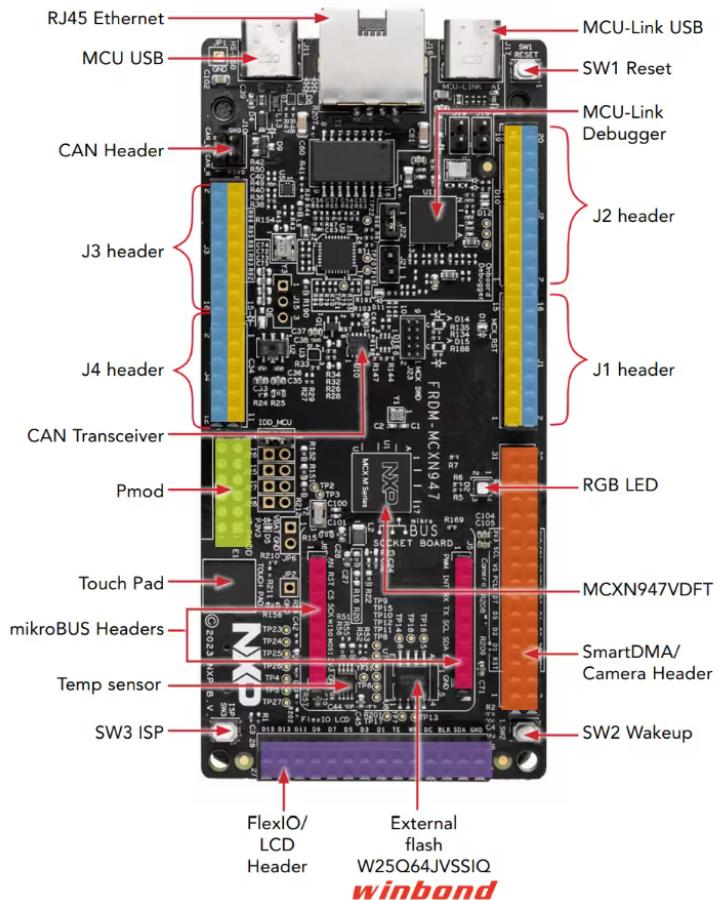
Digitální záznamník lze vyvíjet na vývojové desce s klasickým mikrokontrolérem pomocí C/C++ jazyků. Mikrokontrolér poskytuje nízkoúrovňový přístup k systémovým prostředkům, včetně přímého řízení periferií, časovačů, pamětí a komunikačních rozhraní. Mnohdy lze také kód pro mikrokontrolér optimalizovat z hlediska výkonu, velikosti i spotřeby, stejně jako zvýšit míru zabezpečení samotného firmwaru, neboť je možné přesně řídit chování jednotlivých komponent bez zásahu dalších softwarových vrstev.

Vhodný zástupce vývojové desky se jeví FRDM-MCXN947 od společnosti NXP Semiconductors s mikrokontrolerem MCXN947, která je navržena pro rychlé prototypování vestavných systémů. Tato deska obsahuje nespočet komponent usnadňujících vývoj široké škály aplikací, přítomna je například Q-SPI flash paměť s 8 MB úložným prostorem, FRDM-MCXN947 obsahuje dedikovaný slot pro SD kartu typu SDHC, kterým lze paměťový prostor rozšířit o 32 GB paměti. FRDM-MCXN947 je dále vybavena integrovaným digitálním teplotním senzorem P3T1755DP s přesností 0,5 teplotního stupně, kolíkovými lištami typu Arduino a microBus pro možné rozšíření v podobě expanzních desek plošných spojů. Významnou výhodou je přítomnost vestavěného debuggeru NXP LPC55569, který umožňuje krokové ladění kódu a sledování hodnot proměnných pomocí MCU LinkServer nebo pomocí J-Link programů. K dispozici jsou také dva porty USB typu C, první z nich MCU-Link USB port je určen pro již zmíněné debuggrovací účely, nahrávání programů a sériovou komunikaci, a druhý port, který je na obrázku 3.2 pojmenovaný jako MCU USB, slouží pro aplikační účely a lze ho například využít pro přenos dat pomocí standardů USB Mass Storage, HID (Human Interface Device), USB CDC (Communications Device Class), USB MTP (Media Transfer Protocol) a další. [8, 15]

Mikrokontrolér MCXN947 poskytuje dvě jádra nové generace Arm® Cortex-M33, pracující až na frekvenci 150 MHz. Jádro Cortex-M33 oproti předchozí generaci Cortex-M3 je postaveno s novější instruční sadou ARMv8-M s rozšířením o DSP (Digital Signal Processor), které umožňuje efektivnější zpracování digitálních signálů a numerických výpočtů. Dále jádro nabízí větší míru bezpečnosti v podobě TrustZone, jež umožňuje rozdělení paměťového prostoru na zabezpečenou (secure world) a nezabezpečenou oblast (non-secure world). [15]

Z hlediska paměťových zdrojů nabízí MCXN947 512 KB paměti SRAM s podporou detekce a opravy chyb (Error Correction Code či zkráceně ECC). Pro zvýšení výkonu systému je čip vybaven několika typy rychlé paměti cache. Mikrokontrolér dále poskytuje 2 MB integrované flash paměti a pro programy s vyššími požadavky na úložiště je možné využít zmíněnou Q-SPI flash paměť. [15]

Pro implementaci různorodých aplikací a variabilitu systému postavených na tomto čipu, MCXN947 nabízí širokou škálu periferií, které mohou být zajímavé i z pohledu vývoje digitálního záznamníku. K dispozici jsou například časovače CTIMER a CSTIMER, pro měření časových intervalů či generování periodických událostí. Z hlediska komunikačních rozhraní poskytuje čip osm instancí periferie LP\_FLEXCOMM, jež může být nastavena pro komunikační rozhraní I<sup>2</sup>C, UART či SPI. Nechybí ani podpora síťové komunikace prostřednictvím Ethernetu nebo vysokorychlostního přenosu dat díky rozhraní USB 2.0 s řadičem USBFS (Full-Speed Host and Device Controller), který podporuje také protokol OTG. Pro práci s paměťovými kartami je do čipu integrován Ultra Secured Digital Host Controller (uSDHC), který zprostředkovává komunikaci se SD, SDIO a MMC kartami. Čip MCXN947 také poskytuje tři instance A-D převodníku a taktéž tři analogové komparátory LPCMP. [15]



Obrázek 3.2: Vývojová deska NXP FRDM-MCXN947 [8]

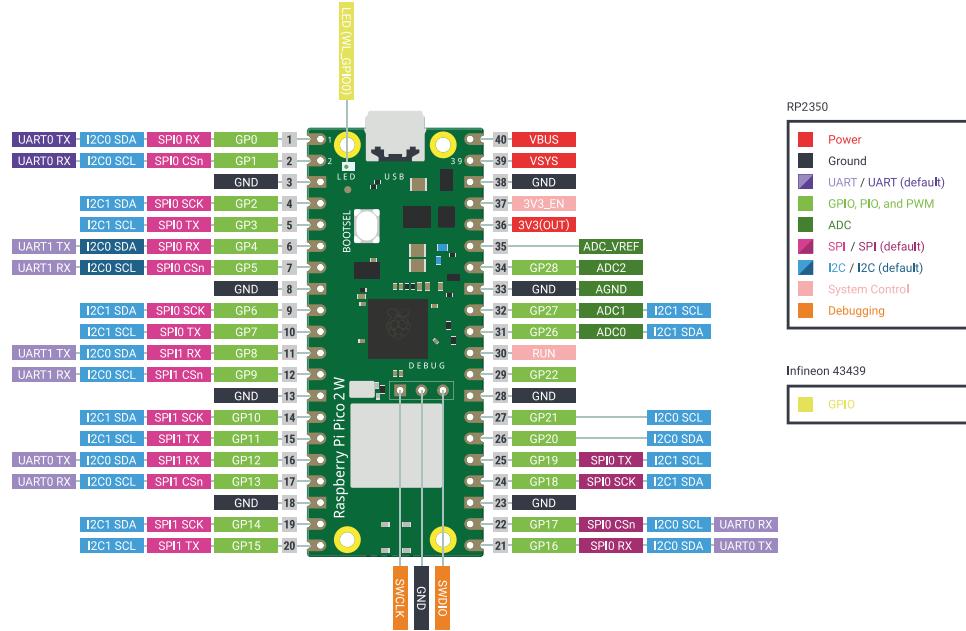
### 3.2.2 Raspberry PI

Jednou z nejvýkonnějších možností pro implementaci digitálního záznamníku v podobě dedikovaného zařízení je využití mikroprocesorových platform s operačním systémem, jakým je například Linux, přičemž nejznámějším zástupcem této kategorie je Raspberry Pi. Tato platforma spadá do kategorie jednodeskových počítačů (SBC – Single Board Computer), které kombinují výpočetní výkon a pružnost standardních počítačových systémů s rozhraními vhodnými pro práci s externími periferiemi. [17]

Raspberry Pi je nízkonákladový minipočítač, dnes už de facto rodina počítačů, postavených na procesorech společnosti ARM, které podporují běh různých operačních systémů. Raspberry Pi oproti klasickým mikrokontrolérům poskytuje vyšší výpočetní výkon, také i větší množství operační paměti a podporuje také multitasking, umožňující běh více aplikací či procesů současně. [17]

Raspberry Pi existuje v několika verzích, následující text je zaměřen na model Raspberry Pi Zero 2 W, který představuje rozumné parametry pro implementaci digitálního záznamníku v podobě dedikovaného zařízení. Model Zero 2 W je vybaven čtyřjádrovým procesorem Broadcom BCM2710A1 založeným na jádru ARM Cortex-A53 s taktem 1 GHz a disponuje 512 MB nízkoenergetické operační paměti SDRAM typu LPDDR2 a jako primární úložiště, na kterém je uložen i operační systém, slouží microSD karta. Raspberry Pi Zero 2 W disponuje jedním USB portem typu micro-B, jenž podporuje režim OTG (On-The-Go)

a umožňuje tak připojení externích zařízení, jako jsou USB flash disky či další periferie. Model Zero 2 W obsahuje také integrovaný bezdrátový komunikační modul podporující Wi-Fi standard 802.11n pracující v pásmu 2,4 GHz, který lze využít k ukládání zaznamenaných dat na vzdálené úložiště (viz. kapitola 2.6.3). [17]



Obrázek 3.3: Vývody Raspberry Pi Zero 2 W [17]

Naopak omezením Raspberry Pi Zero 2 W může být počet komunikačních periferií, platforma totiž disponuje pouze dvěma moduly UART, SPI a I<sup>2</sup>C. Již dvě instance UART periferie mohou vývoj zkomplikovat, pokud by byl jeden UART využit pro ladící výstup a druhý pro příjem sériových dat, nebylo by již možné připojit další komponentu komunikující přes UART, což by mohlo výrazně omezit možnosti budoucího rozšíření systému digitálního záznamníku. Raspberry Pi Zero 2 W rovněž nedisponuje žádným analogovým komparátorem, čímž se zmenšuje výběr jak vyřešit detekci ztráty napájecího napětí, ta by v tomto případě musela být vyřešena pomocí GPIO, ADC nebo pomocí externí komponenty. Zero 2 W taktéž nedisponuje interním obvodem reálného času, nabízelo by se tedy využít jako referenční zdroj času pro časové značky například časovač se softwarovým přešřením. [17]

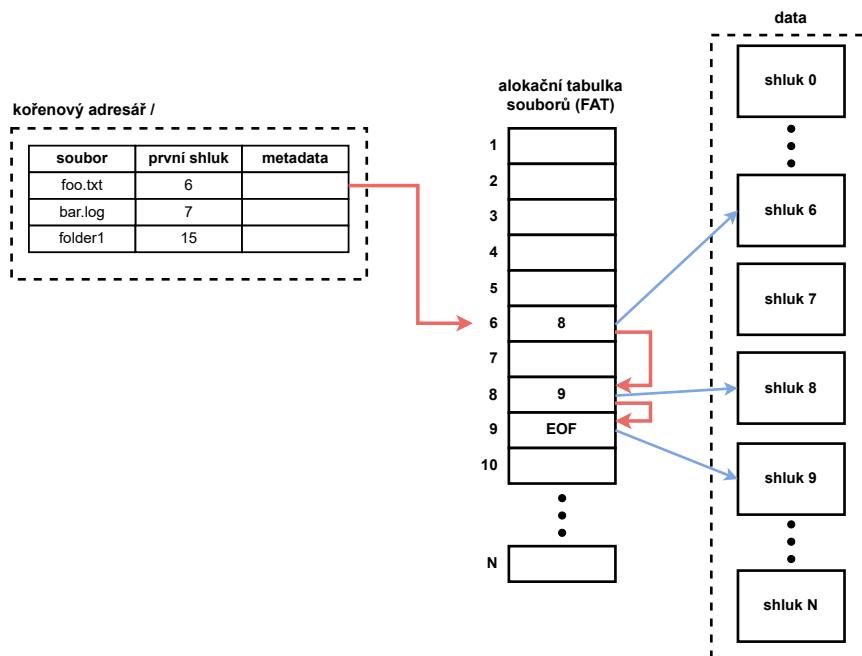
### 3.3 Možnosti správy dat – souborové systémy

Zaznamenaná data na externím úložišti je potřeba organizovat a spravovat, aby mohla být v budoucnu určitým způsobem interpretována a zároveň je vhodné zajistit konzistence a perzistence těchto dat. K tomu účelu lze využít například souborový systém, databázový systém nebo navrhnout vlastní způsob organizace a správy dat. Každý z těchto přístupů má svá specifika, nicméně data, která budou zaznamenávána vyvíjeným záznamníkem, jsou nestrukturovaná. Pod pojmem nestrukturovaná data lze rozumět taková data, která nejsou organizována jednotně v tabulkách (relacích) nebo jiných databázových strukturách, ale

mohou být ukládána do více typů souborů v různých formátech a umístěna do různorodých adresářových struktur. Z tohoto důvodu byl zvolen souborový systém. [23, 20]

### 3.3.1 FATFS

Prvním z představených souborových systémů je FAT File System (FATFS), který je implementován v podobě lehké softwarové knihovny pro mikrokontroléry a vestavěné systémy implementující podporu souborového systému FAT/exFAT. FATFS se řadí mezi hierarchické souborové systémy založené na alokační tabulce souborů (File Allocation Table — FAT), ve kterých jsou data organizována do logických jednotek označovaných jako shluky (clusters). Každý soubor uložený na úložném médiu se tak skládá z jednoho nebo více těchto shluků, přičemž informace o jejich návaznosti jsou uloženy právě v alokační tabulce FAT. [29, 45]



Obrázek 3.4: Souborový systém s alokační tabulkou souborů [45]

Fyzické úložiště v systému FAT je rozděleno do několika oblastí, konkrétně do zaváděcího záznamu (boot record) ležícího v rezervované oblasti (reserved area), který obsahuje informace nutné pro inicializaci a načtení souborového systému, alokační tabulky FAT1 a FAT2, které spravují umístění jednotlivých shluků. Za nimi se nachází oblast kořenového adresáře (root directory), ve které jsou umístěny záznamy o souborech a adresářích nejvyšší úrovni. Poslední část tvoří datová oblast (data area), kde jsou samotné soubory a podadresáře fyzicky uloženy. [45]



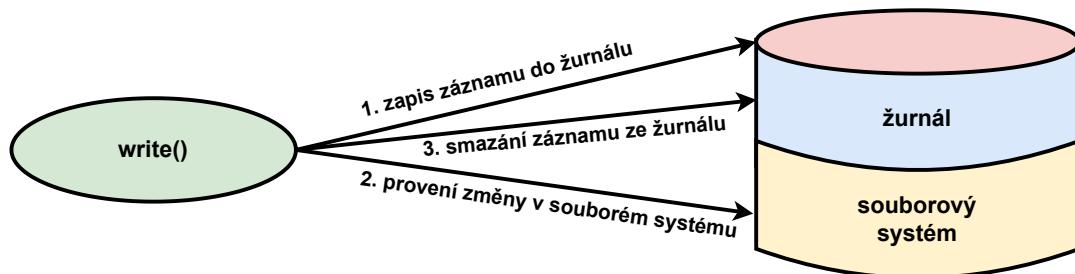
Obrázek 3.5: Oblasti souborového systému FAT [45]

V dnešní době existuje několik variant systému FAT, mezi nejznámější patří FAT12, FAT16, FAT32 a exFAT (FAT64). Vzhledem k rostoucí kapacitě paměťových médií se dnes standardně používají varianty FAT32 a exFAT. Například starší varianta FAT16 s 16bitovým adresním prostorem pro identifikaci jednotlivých clusterů umožňovala velikost uložiště pouze do 4 GB. Z tohoto důvodu i dnešní implementace FATFS podporují 32-bitovou či 64-bitovou variantu adresování jednotlivých bloků dat (LBA – Logical Block Addressing), které podporují mnohem větší kapacitu uložiště. [30]

FATFS v současnosti představuje jeden z nejrozšířenějších souborových systémů pro použití ve vestavěných aplikacích, a to zejména díky své nízké paměťové náročnosti a snadné přenositelnosti mezi různými hardwarovými platformami. Významnou předností tohoto souborového systému je také široká možnost konfigurace, která umožňuje volby krátkých a dlouhých názvů souborů (LFN – Long File Name), navíc v různých formátech ANSI/OEM či Unicode. Pro velká uložiště lze využít podporu exFAT s 64-bitovým adresováním bloků (LBA) a GPT (GUID Partition Table) pro práci s diskovými oddíly o velké kapacitě. FATFS také podporuje více fyzických jednotek či oddílů současně, variabilní velikost sektorů a různé kódové stránky. Knihovna rovněž obsahuje podporu pro volitelná rozhraní API, I/O buffering a režim pouze pro čtení. V případě použití v aplikaci běžící nad operačními systémy reálného času (RTOS) je garantována bezpečnost při práci ve vícevláknovém prostředí.<sup>3,4</sup>

### 3.3.2 LittleFS

Alternativou k souborovému systému FATFS může být žurnálovací souborový systém LittleFS, který je podobně jako FATFS implementován ve formě odlehčené softwarové knihovny určené primárně pro vestavěné systémy a mikrokontroléry. Tento souborový systém byl vyvinut s ohledem na specifika paměti typu flash, aby se rovnoměrně rozdělil zápis na jednotlivé bloky a nebyly první bloky vytíženy více než ty s většími indexy. [14]



Obrázek 3.6: Žurnálování při zápisu do souboru [68]

<sup>3</sup>Knihovna FATFS je takzvaně Thread Safe.

<sup>4</sup>[http://elm-chan.org/docs/fat\\_e.html](http://elm-chan.org/docs/fat_e.html)

Souborový systém LittleFS využívá pokročilý mechanismus atomického zpracování známého například z databázových systémů, kdy je každá operace buď kompletně dokončena, nebo v případě selhání či výpadku napájení zcela zrušena. K tomu je využit mechanismus žurnálování (journaling). Operace nad souborovým systémem jsou nejprve zaznamenány do speciální struktury žurnálu před jejich fyzickým provedením. Pokud by došlo k selhání, souborový systém se může spolehlivě vrátit do konzistentního stavu. [14]

K žurnálování souborový systém LittleFS využívá technologii páry metadat (metadata pairs). Při každé operaci jsou atomicky měněna metadata souborového systému a tedy ke změně jsou využity dvě speciální oblasti, označované jako metadata páry, které jsou opatřeny kontrolními součty (checksum) a čísla revizí (revision count). Při každém zápisu se střídavě aktualizuje jeden ze dvou bloků metadat, přičemž druhý blok vždy uchovává předchozí konzistentní stav. Pokud by tedy došlo k selhání zápisu, systém při následném spuštění zkонтroluje kontrolní součty, revizní čísla obou bloků a rozhodne, který blok obsahuje platná data. [14]

Pro samotná data (non-meta data) souborový systém LittleFS používá techniku Copy-on-Write (COW). Tato technika zajišťuje konzistenci dat tak, že při každém zápisu jsou aktuální (nová) data nejprve zapsána do volných bloků a teprve po úspěšném dokončení zápisu jsou bloky s neaktuálními/starými daty označeny jako volné. Pokud by tedy došlo během zápisu k chybě nebo výpadku napájení, systém se vrátí k původním, neporušeným blokům, a tím zabrání ztrátě nebo poškození dat. [14]

Hlavními výhodami souborového systému LittleFS jsou tedy šetrnost k paměťovým médiím, nad kterými souborový systém operuje, velmi nízká paměťová náročnost jak na velikost programového kódu v paměti flash, tak na množství operační paměti RAM potřebné k jeho provozu. Dalším významným přínosem jsou již výše zmíněné pokročilé mechanismy známé z databázových systémů či pokročilých souborových systémů, zejména žurnálování a atomické operace, které umožňují zachovat souborový systém v konzistentním stavu i v případě neočekávaných selhání nebo výpadků napájení. [14]

Žurnálovací souborový systém LittleFS má ale také své nevýhody. Jednou z nejvýznamnějších je absence kompatibility se standardními operačními systémy, jako jsou Windows, Linux nebo macOS. Data uložená pomocí LittleFS nejsou přímo čitelná na běžném počítači bez použití dodatečného softwarového nástroje (tzv. wrapperu).<sup>5</sup>

### 3.4 Výběr řízení přístupu k získaným datům

Jedním z požadavků na realizaci digitálního záznamníku v této bakalářské práci je přístup k zaznamenaným datům bez nutnosti vyjmutí fyzického média. V úvahu přicházejí varianty využívající síťové technologie či lokální přístupová rozhraní, kam spadá například USB. Nicméně vzhledem k bezpečnostním restrikcím společnosti NXP Semiconductors není možné zařízení připojit k síti například pomocí Ethernetového rozhraní. Z tohoto důvodu se přenos dat omezuje především na protokoly fungující nad USB rozhraním, které je dostupné na všech vývojových deskách popsaných v kapitole 3.2.

---

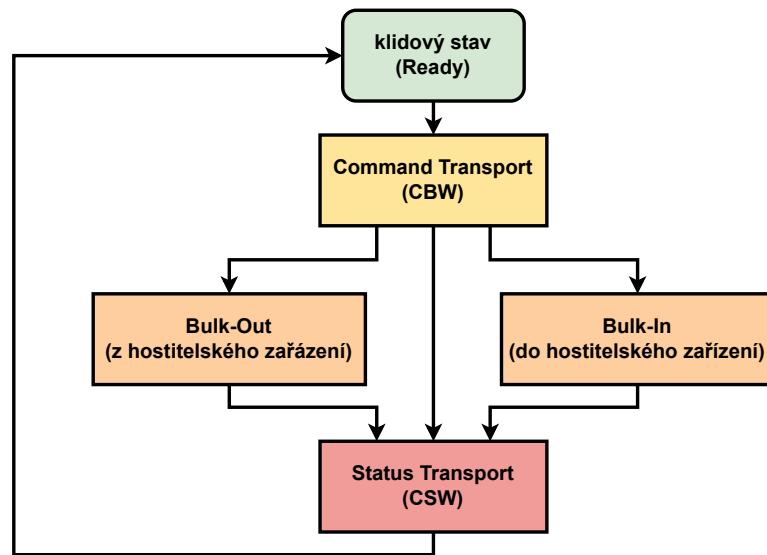
<sup>5</sup><https://www.cnx-software.com/2018/01/02/littlefs-is-an-open-source-low-footprint-resilient-file-system-designed-for-tiny-devices/>

### 3.4.1 USB Mass Storage

Jednou z možností pro zajištění přístupu k uloženým datům představuje použití standardu USB Mass Storage (MSC), vypracovaného v roce 1998 organizací USB-IF (USB Implementers Forum). Tento standard umožňuje, aby zařízení vystupovalo vůči hostitelskému systému jako běžné blokové úložiště (block device), tedy například jako paměťová karta nebo externí disk. Velkou výhodou je, že MSC je dodnes podporován většinou moderních operačních systémů, včetně Microsoft Windows, distribucí Linuxu i macOS. Díky tomu lze data snadno vyčítat a zapisovat téměř z jakéhokoliv osobního počítače bez nutnosti speciálních ovladačů. [18, 19]

USB Mass Storage rozlišuje dva komunikační prvky, prvním z nich je Mass Storage host, tedy zařízení, které aktivně řídí přístup k úložišti a provádí operace čtení a zápisu dat, tím je typicky osobní počítač nebo jiný hostitelský systém. Druhým prvkem je Mass Storage zařízení (device), které je koncovým zařízením vystupující jako externí paměťové médium. [18, 19]

Z hlediska přenosových protokolů definuje USB Mass Storage dvě hlavní metody komunikace, kterými jsou Control/Bulk/Interrupt (CBI) a Bulk-Only Transport (BOT). CBI je přenosový protokol definovaný standardem USB 1.1, který kombinuje tři typy USB přenosů – řídicí (control), blokové (bulk) a přerušovací (interrupt). Přenos se dále dělí na protokol přenosu dat, který využívá přenos přerušení, a protokol, který přenos přerušení nevyužívá. V současnosti nejrozšířenější metodou přenosu v rámci USB Mass Storage je transport pomocí BOT, v tomto režimu jsou data přenášena po blocích. Součástí USB Mass Storage bývá také USB Floppy Interface (UFI), což je sada příkazů založená na SCSI-2 a SFF-8070 sadách příkazů, navržená pro jednoduchá bloková zařízení, původně určená pro USB disketové mechaniky (floppy disk), ale dnes běžně používaná pro základní interakci s jakýmkoliv typem Mass Storage zařízení. UFI definuje sadu příkazů pro čtení, zápis a správu paměťového média. [18]



Obrázek 3.7: Stavový diagram USB Mass Storage protokolu [19]

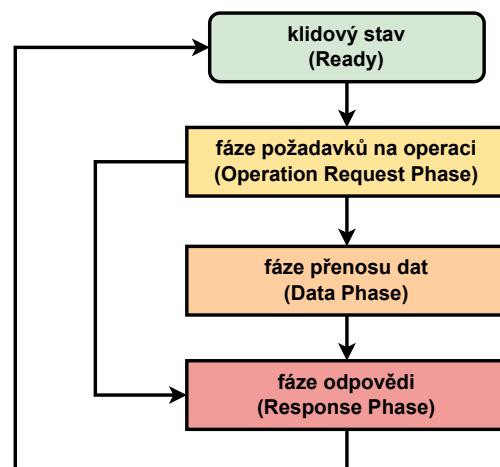
Komunikace mezi hostitelem a zařízením v rámci protokolu Bulk-Only Transport (BOT) probíhá ve třech hlavních fázích, zajišťují řízený přenos příkazů, dat a zpětné vazby o provedení operace. První fází je přenos příkazu (Command Transport), během které host odešle

zařízení požadavek na operaci v podobě struktury Command Block Wrapper (CBW). Tato struktura obsahuje informace o typu požadované operace (jakou je třeba inicializace, čtení, zápis), adresaci cílového sektoru na médiu a velikost přenášených dat. Pokud příkaz vyžaduje přenos dat, následuje druhá fáze – přenos dat (Data Transport). Data mohou být posílána ze zařízení do hostitele (například při čtení dat ze zařízení) nebo z hostitele do zařízení (například při zápisu na SD kartu). Přenos probíhá výhradně pomocí bulk přenosů. Poslední fází je Status Transport, v této fázi zařízení odešle zpět hostiteli strukturu CSW, která obsahuje informaci o výsledku provedené operace<sup>6</sup>. Ne všechny příkazy však vyžadují přenos dat (například příkazy pro kontrolu stavu zařízení), v těchto případech je fáze Data Transport vynechána a zařízení odesílá CSW ihned po přijetí CBW. [19]

### 3.4.2 Media Transfer Protocol

Media Transfer Protocol neboli zkráceně MTP je druhým významným protokolem umožňujícím přenos multimediálních dat mezi hostitelem a cílovým zařízením pomocí USB. MTP protokol byl navržen firmou Microsoft na základě protokolu Picture Transfer Protocol (PTP), se kterým je dodnes zpětně kompatibilní. MTP je od roku 2008 součástí USB standardu a od roku 2011 se stal standardem pro transfer souborů u zařízení využívajících operační systém Android. [16, 39]

MTP komunikace probíhá ve formě transakcí, přičemž role komunikujících zařízení jsou iniciátor (initiator) a odpovídající (responder), zároveň je možné, aby si zařízení role mohly vyměnit. Transakce probíhají vždy ve třech fázích definovaných standardem, tou první je fáze požadavku na operaci (Operation Request Phase), kdy iniciátor vyšle požadavek na provedení určité operace. Následuje volitelná druhá fáze přenosu dat (Data Phase), ve které proběhne přenos dat, pokud to daná operace vyžaduje. Poslední je fáze odpovědi, ve které odpovídající odešle iniciátoru odpověď, ve které informuje o výsledku operace. Probíhající komunikace je vždy jednosměrná (unidirectional), tedy v rámci jedné operace mohou data být pouze přenášena od iniciátora k odpovídajícímu či naopak, nikoliv však oběma směry současně. [16]



Obrázek 3.8: Stavový diagram Media Transfer Protocol [16]

<sup>6</sup>v případě chyby může obsahovat další doplňující informace jako třeba množství nepřenesených dat (Data Residue)

Na rozdíl od USB Mass Storage, kde je médium připojeno přímo jako blokové zařízení a hostitel má přímý přístup k souborovému systému, u MTP k tomuto přímému přístupu nedochází. Iniciátor tedy není zodpovědný za přímé operace se souborovým systémem, ale pouze zasílá žádosti o přenos nebo manipulaci se soubory. To znamená, že data zde nejsou interpretována připojeným zařízením (iniciátorem), ale interpretaci má na starosti odpovídající zařízení, a připojenému iniciátoru předává jen celé soubory. To přináší výhodu, obě zařízení mohou přistupovat k datům současně. Naopak to ale vytváří nové problémy. Přes počítač nelze opravit poškozené soubory uložené na zařízení, v případě úpravy souboru je třeba nejprve soubor stáhnout do počítače, zde upravit a následně přenést zpět do zařízení. [16]

### 3.5 Výběr zdroje času pro časové značky

Požadavkem na implementovaný digitální záznamník je také vkládání časových značek do záznamu, které budou využity pro následnou synchronizaci s testovacím zařízením. Způsobů, jak generovat časové značky, je mnoho, a je tedy nutné si na začátku stanovit základní požadavky, zda časové značky mohou být relativní (tedy například od času spuštění zařízení) či absolutní. Zároveň je však nutné při návrhu systému zvážit i další parametry, jakým je jednak granularita časových značek, tedy časové rozlišení, s jakým budou události zaznamenávány, a také, zda zdroj času bude lokální nebo vzdálený. Tato kapitola představí některé ze způsobů, jak generovat časové značky, a dále rozebere jejich výhody a problémy, kterým konkrétní volby čelí. [57]

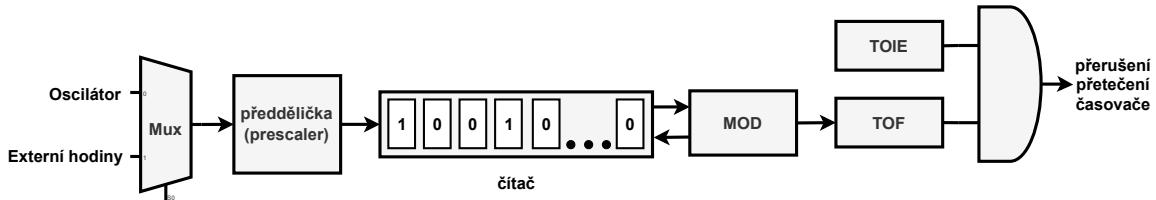
#### 3.5.1 Interní časovač

Jedním z nejjednodušších způsobů, jak generovat časové značky, je využití interního časovače, který je běžně dostupný na většině mikrokontrolérů. Interní časovač pracuje na základě přetečení čítače, který je taktován z hodinového signálu, jenž je generován interním oscilátorem nebo zvoleným hodinovým zdrojem mikrokontroléru. [12]

Schéma na následujícím obrázku 3.9 znázorňuje základní strukturu interního časovače. Na vstupu se nachází multiplexor (Mux), pomocí kterého je třeba zvolit vhodný zdroj hodinového signálu, na základě kterého bude časovač reagovat, tedy inkrementovat. Tuto reakci můžeme dále ještě upravit pomocí děličky kmitočtu (prescaler). Pomocí prescaleru lze snížit rychlosť příchodu pulzů do čítače, a tím ovlivnit rychlosť, s jakou časovač bude inkrementovat svou hodnotu. Na výstupu z prescaleru je napojen čítač (Module Counter), který s každým přijatým impulzem zvýší svou hodnotu o jedničku. Tento čítač je navíc svázán s modulo registrem (MOD), který definuje mezní hodnotu čítače. Jakmile čítač dosáhne hodnoty uložené v registru MOD, dojde k jeho přetečení (overflow), čítač se automaticky vynuluje<sup>7</sup> a je vygenerován příznak přetečení (TOF – Timer Overflow Flag). Pokud je zároveň nastaven příznak TOIE (Timer Overflow Interrupt Enable), tak je do procesoru odeslán požadavek na vyvolání přerušení. V rámci obslužné rutiny tohoto přerušení je následně možné aktualizovat čítač časových značek a udržovat tak aktuální hodnotu pro následné doplnění do záznamů. [12]

---

<sup>7</sup>V nejjednodušším případě je čítač vynulován, některé časovače, umožňují nastavit i jiné reakce, jakou je třeba dekrementace zpět na počáteční hodnotu.



Obrázek 3.9: Blokový diagram časovače [12]

Výhodou takového způsobu je možnost přizpůsobení časového rozlišení podle potřeb aplikace. Časové rozlišení lze ovlivnit volbou hodinového zdroje, hodnotou děličky kmitočtu a nastavením mezní hodnoty MOD. Lze tak snadno generovat přerušení s periodou i v řadě jednotek mikrosekund, sekund až po výrazně delší časové úseky. V základní podobě poskytuje tento mechanismus měření relativního času, často od začátku běhu systému, pokud není použit doplňující prvek, který by umožňoval uchování či zjištění aktuálního času, jako je doplňující obvod reálného času nebo GPS modul. Právě skutečnost, že se jedná o relativní čas, přináší i určité výhody, odpadá totiž nutnost řešit komplexní problémy spojené s absolutním časem, jako jsou přechody mezi letním a zimním časem, nebo postupná odchylka interního časového základu od reálného času. [57]

Naopak mezi nevýhody použití interního časovače patří skutečnost, že takto získaný čas je pouze relativní, což v některých aplikacích nemusí být dostačující. Další nevýhodou je, že zpracování přerušení generovaných časovačem probíhá přímo jádrem mikrokontroléru, a tím odebírá výpočetní kapacitu, která by jinak mohla být věnována jiným úlohám. Platí přitom, že čím jemnější časové rozlišení je požadováno, tím vyšší je frekvence generovaných přerušení a tím větší je dopad na výpočetní výkon mikrokontroléru. [57]

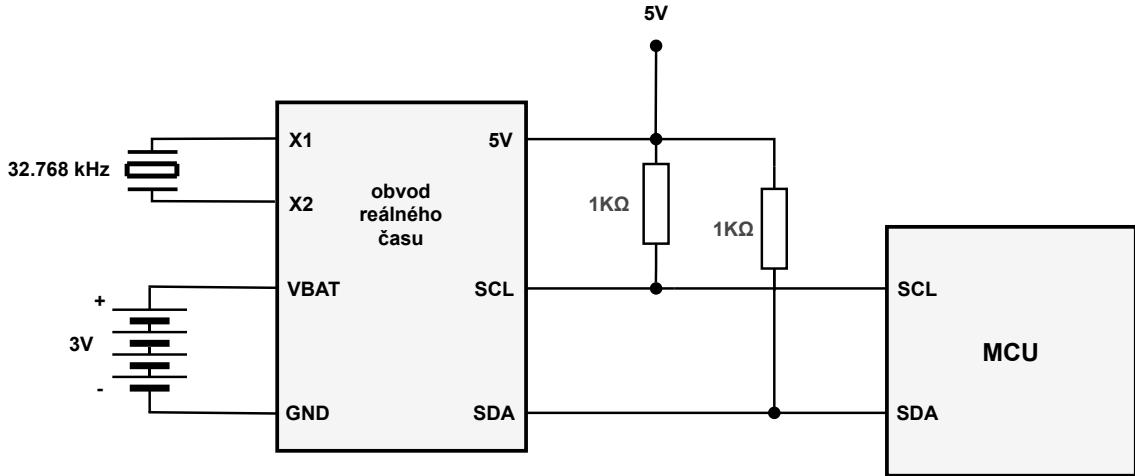
### 3.5.2 Obvod reálného času

Možné je také využít obvod reálného času (RTC – Real-Time Circuit), který je součástí většiny moderních mikrokontrolérů nebo dostupný jako samostatný externí integrovaný obvod. RTC je specializovaný časový modul navržený pro dlouhodobé uchování reálného času, tedy hodin, minut, sekund i datumů a dalších kalendářních informací.

Obvod reálného času je typicky taktován externím či integrovaným krystalem<sup>8</sup>, obvykle s rezonanční frekvencí 32,768 Hz. Tato hodnota není zvolena náhodně – odpovídá hodnotě  $2^{15}$ , tedy násobku dvou a umožňuje tak pomocí jednoduché děličky frekvence vydělit frekvenci na 1 Hz. Jeden kmit tak představuje jednu sekundu. Mnohé obvody reálného času také poskytují možnost připojení záložní baterie<sup>9</sup>, která slouží k udržení chodu interního časového obvodu i v případě, kdy je hlavní napájení zařízení odpojeno. Díky tomu může RTC kontinuálně uchovávat čas i po dlouhou dobu, kdy je zařízení neaktivní a není nutné čas nastavovat po každém startu zařízení. Jak již bylo zmíněno, RTC bývá často v podobě samostatných integrovaných obvodů nebo bývá integrován přímo v mikrokontroléru. Pro komunikaci mikrokontroléru s RTC jako samostatným integrovaným obvodem se obvykle využívá komunikační rozhraní I<sup>2</sup>C, popřípadě SPI. Naopak k RTC integrovanému přímo do mikrokontroléru se přistupuje přímo prostřednictvím speciálních systémových registrů. [4, 69]

<sup>8</sup>Na obrázku 3.10 je krystal připojen pomocí vývodu X1 a X2 k RTC modulu.

<sup>9</sup>Na obrázku 3.10 je baterie připojena pomocí vývodu VBAT a GND k RTC modulu.



Obrázek 3.10: Zapojení obvodu reálného času v podobě externího modulu připojeného k mikrokontroléru [11]

Obvod reálného času oproti běžnému časovači udržuje absolutní čas a dokáže jej udržovat dlouhodobě, a to i při přerušení napájení, pokud je tedy RTC vybaveno záložním napájením (například pomocí knoflíkové baterie).

Využití RTC má také své limitace či nevýhody. Zdrojem problémů může být například krystal, pomocí kterého je čas udržován, může například dojít k jeho poruše (například se krystal zastaví). Může také nastat situace, kdy se vybije bateriový článek, poskytující napětí v době, kdy je hlavní napájení zařízení odpojeno, a je tak ztracena informace o aktuálním čase. Jedním z omezení je také minimální časové rozlišení, které je typicky nastaveno na sekundy. Mnoho standardních RTC modulů tak neumožňuje měřit a zaznamenávat čas s vyšší přesností, například v rádu milisekund či setin sekundy, což může být problémem u aplikací vyžadujících jemnější časovou osu. Také operování s absolutním časem vznikají specifické situace, jako například přechod mezi letním a zimním časem, změny časových pásem, navíc čas měřený obvodem reálného času se bude odchylovat od reálného času s narušující dobou svého běhu v důsledku výrobní tolerance, teplotní charakteristiky a stárnutí krystalového oscilátoru, působením elektromagnetického rušení a dalších faktorů. [4, 11, 69]

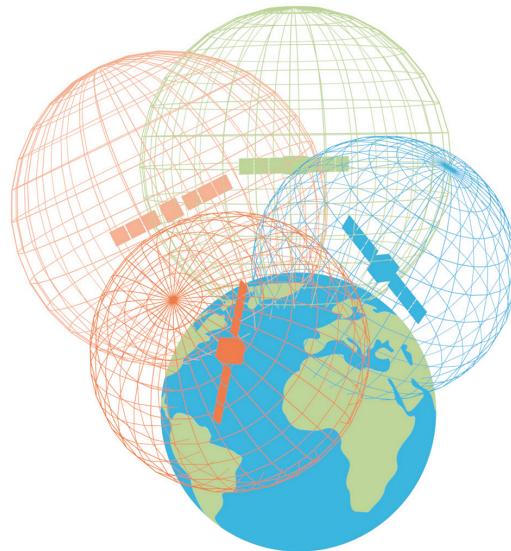
### 3.5.3 Vzdálené zdroje času

Další možností, jak získat časové značky, je využití vzdálených zdrojů času, které poskytují aktuální absolutní čas prostřednictvím komunikačního rozhraní. Takovými zdroji mohou být GPS satelity, NTP servery, rádiové stanice (například DCF77) a další specializované systémy. Přestože se tyto technologie mohou lišit ve způsobu přenosu a fyzikální povaze signálu, sdílí některé principy pro určení aktuálního času. Standardně tyto systémy poskytují čas synchronizovaný vůči mezinárodnímu standardu UTC (Coordinated Universal Time) a zároveň pracují s hierarchickým modelem přesnosti označovaným jako stratum. [3, 24]

Koncept stratum slouží k vyjádření vzdálenosti daného časového zdroje od primárního referenčního času. Hodnotou stratum 0 jsou označeny primární zdroje času, takovými jsou například atomové hodiny. Následovné zdroje s hodnotou stratum 1 jsou přímo synchronizována se zařízeními s hodnotou stratum 0, přičemž další úrovně (stratum 2, 3 a vyšší) postupně navazují na předchozí zařízení s hodnotou stratum o jednu jednotku nižší. [24]

## Globální polohovací systém (GPS)

Hodnotu stratum 0 mají například satelity globálního polohovacího systému (GPS), které obsahují atomové hodiny. Přesný čas satelitů je v systému GPS naprosto zásadní, neboť na jeho základě probíhá výpočet polohy přijímače. Bez znalosti velmi přesného času by nebylo možné určit, za jak dlouhou dobu doputoval signál od satelitu k přijímači, a tím pádem by vznikaly chyby ve výpočtu vzdálenosti, respektive polohy.<sup>10</sup> GPS satelity tedy pravidelně vysílají rádiový signál obsahující identifikátor satelitu, hodinový signál (tj. čas) a části almanachu obsahující například rovnici oběžné dráhy satelitu. Přijímač potřebuje tato data alespoň ze čtyř satelitů<sup>11</sup>, aby mohl dopočítat řešení čtyř rovnic o čtyřech neznámých, jehož součástí je i údaj o přesném čase. [3]



Obrázek 3.11: Minimální nutná sestava čtyř GPS satelitů pro výpočet lokace a času přijímače [10]

Z pohledu přijímače má na starosti veškeré výpočty speciální GPS čipset, který integruje řadu funkcí potřebných pro správnou činnost systému. Kromě výpočtů spojených s určením polohy zpracovává i analogový rádiový signál přijatý anténou, provádí konverzi signálu do digitální podoby, spravuje řízení napájení a poskytuje rozhraní pro interakci s uživatelem. Rozdíly mezi čipovými sadami obvykle spočívají v rovnováze mezi spotřebou energie, časy akvizice a dostupností hardwaru. [3]

Jedna z nevýhod globálního polohovacího systému může nastat u zařízení využívaných v interiérech. Ve vnitřních prostorech může nastat problém se samotným průchodem signálu. Moderní budovy často obsahují betonové stěny, kovové konstrukce, skla s metalizovanými vrstvami nebo střechy pokryté solárními panely, které představují překážky pro rádiový signál a mohou způsobit nepřesnosti ve vypočtených údajích v důsledku změny trajektorie signálu. Mimo jiné může také dojít k výraznému zeslabení nebo úplnému pohlcení signálu. [3]

<sup>10</sup>Chyba jedné nanosekundy v tranzitním čase znamená chybu 30cm ve vzdálenosti.

<sup>11</sup>GPS přijímač, může pro dopočítání využívat data i více než ze čtyř satelitů, aby se eliminovaly chyby vzniklé například z odchylek atomových hodin jednotlivých satelitů, či ze zanedbání faktu, že orbity satelitů nejsou zcela dokonalé kružnice.

## Network Time Protocol (NTP)

Další široce používanou technologií pro získání přesného času je síťový protokol NTP (Network Time Protocol). Tento protokol slouží k synchronizaci systémového času počítačových zařízení s časovými servery prostřednictvím sítí TCP/IP. NTP servery jsou organizovány do hierarchického systému vrstev (stratum), který zároveň umožňuje získání času i v případě nedostupnosti některých serverů.

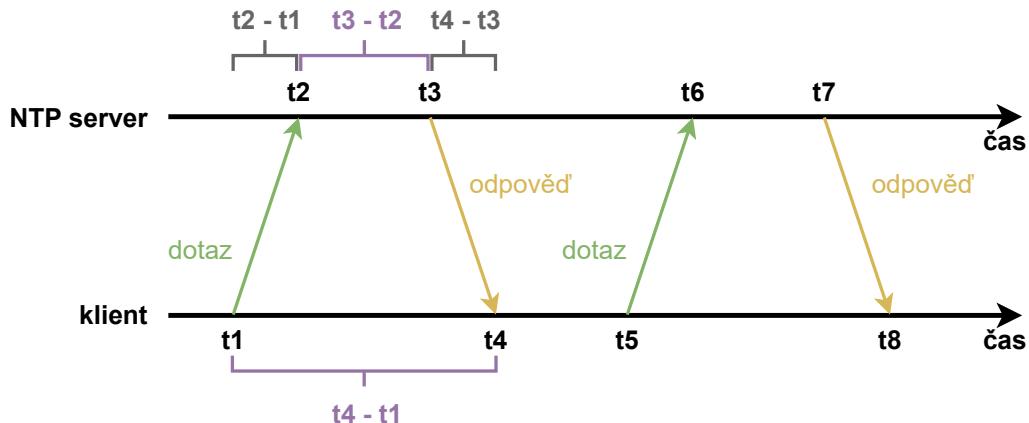
Princip spočívá ve výměně časových značek mezi klientem a serverem. Klient odešle dotaz na NTP server ve chvíli  $T_1$  a server tento dotaz přijme v čase  $T_2$ . Po zpracování server odešle odpověď v čase  $T_3$ , kterou klient přijme v čase  $T_4$ . Na základě těchto čtyř časových údajů lze vypočítat celkové síťové zpoždění  $\delta$  a časový rozdíl (offset)  $\theta$  mezi hodinami klienta a serveru. Zpoždění  $\delta$  lze určit jako:

$$\delta = (T_4 - T_1) - (T_3 - T_2)$$

Zatímco rozdíl mezi klientským a serverovým časem čili offset ( $\theta$ ), lze vypočítat dle vztahu:

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

Získané hodnoty zpoždění a rozdílu pak klientovi umožní upravit systémový čas tak, aby co nejpřesněji odpovídal referenčnímu času. Protokol NTP navíc provádí více opakových měření a volí ty s nejnižším zpožděním, čímž zvyšuje stabilitu synchronizace i při kolísajícím síťovém připojení. [24]



Obrázek 3.12: Komunikace pomocí Network Time Protocol [24]

NTP poskytuje přesný čas, nicméně také není vhodný pro každé vestavné zařízení vyžadující absolutní čas. Pro jeho fungování je nutná přítomnost TCP/IP zásobníku a zároveň aktivní síťové připojení k dostupnému NTP serveru. V prostředí, kde není možné garantovat trvalou konektivitu (například v mobilních, autonomních nebo izolovaných systémech), může být použití NTP omezené nebo zcela nevhodné. V takových případech je výhodnější spolehnout se na jiné zdroje času, jako je například GPS modul, nebo využít NTP pouze pro synchronizaci času, a následně se spolehnout na lokální zdroj času, například časovač či RTC.

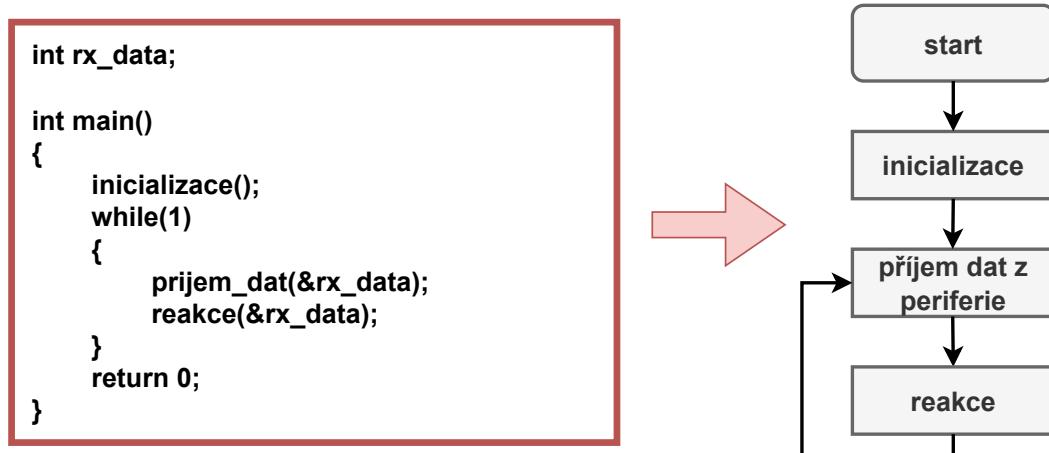
## 3.6 Výber přístupu řízení běhu aplikace

Důležitou součástí digitálního záznamníku je způsob řízení běhu aplikace, příčemž tato volba zásadně ovlivňuje architekturu výsledného řešení. U aplikací vestavných zařízení se zpravidla rozhoduje mezi dvěma směry. Prvním z nich je implementace na holém železe, obvykle se pro tento způsob využívá anglický název bare-metal a druhou možností může být stavět program nad operačním systémem reálného času (Real-Time Operating System, zkráceně RTOS). V následujících podkapitolách budou oba přístupy rozebrány, včetně jejich výhod a nevýhod.

### 3.6.1 Bare-Metal

Bare-metal přístup znamená přímé řízení programu bez operačního systému, typicky s hlavní smyčkou (tzv. *superloop*). Celý systém je dedikován pouze jedné aplikaci, která tak získává přímou kontrolu nad hardwarem. Program tedy přímo přistupuje k hardwarovým registrům mikrokontrolérů. [25]

Program se obvykle skládá ze dvou hlavních fází. První fází je inicializace (viz. obrázek 3.13), během které se nastavují systémové zdroje, konfiguruje se taktování, inicializují se periferie a případně se nastavují přerušení včetně přiřazení obslužných rutin a konfigurace řadiče přerušení. Tato inicializace probíhá pouze jednou při startu systému. Po dokončení inicializace přechází aplikace do hlavní smyčky, kde v nekonečném cyklu vykonává logiku programu, například zpracovává příchozí události a obsluhuje periferie. Řízení programu standardně probíhá sekvenčně, nicméně od tohoto sekvenčního provádění se odchylí pouze tehdy, když dojde k události přerušení. [25]



Obrázek 3.13: Příklad vestavěnné aplikace využívající bare-metal přístup

Výhodou programování na bázi bare-metal je, že vestavné softwarové programy lze podrobně naplánovat pro konkrétní případ použití na co nejmenší úrovni, aniž by bylo nutné akceptovat režii navíc a možné chyby operačního systému. V optimálním případě tak vzniká řešení na míru, které spolehlivě plní úkoly a šetří zdroje. To má smysl vždy, pokud jsou úlohy, které mají být prováděny, zvládnutelné, takže lze zajistit co největší míru deterministického chování systému. [25]

Na druhou stranu, při práci s bare-metal přístupem se mohou vyskytnout obtíže s zvládáním vysoké úrovně složitosti. Pokud se již tak nemalé projekty v průběhu implementační fáze nečekaně zvětší, například pokud přibudou nové funkce, které nebyly při předběžném plánování zohledněny, může být obtížné udržet si přehled. Tato obtíž může ještě narůst, pokud chybí hardwareová abstrakční vrstva nebo je navíc aplikace vyvíjená v assembleru a dokumentace k vývojové desce je nedostatečná, to může zkomplikovat vývoj i zkušeným programátorem. Nelze také opomenout, že manuální správa systémových prostředků, například ruční blokování prostředků pomocí semaforů, může být časově náročná a náchylná k chybám. [25]

### 3.6.2 Operační systém reálného času

Operační systém reálného času je typem počítačového operačního systému, který je navržen tak, aby byl malý a deterministický, tedy aby opakováný vstup vedl ke stejnemu výstupu. Tyto operační systémy se běžně používají ve vestavných systémech, jako jsou lékařské přístroje a automobilové řídicí jednotky, které musí reagovat na vnější události v přesně omezeném čase. [33]

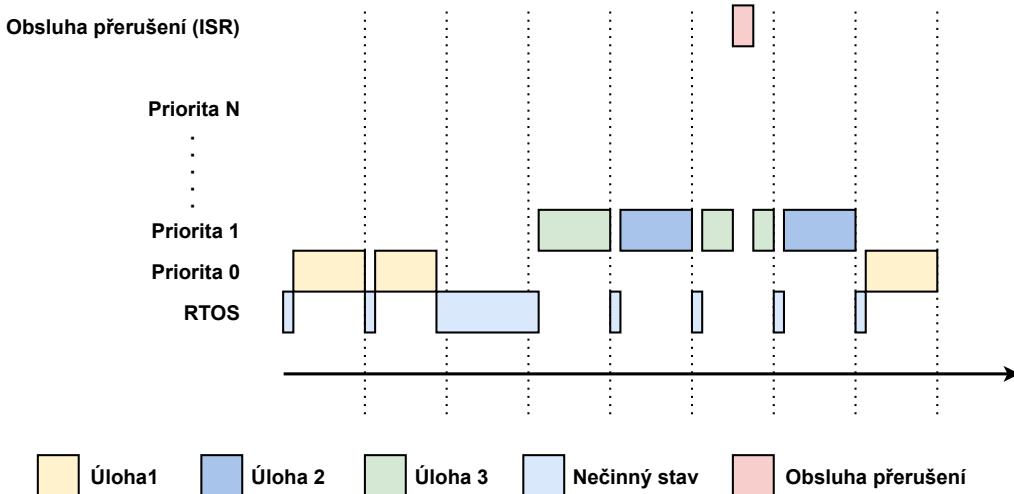
RTOS podporuje multitasking, tedy schopnost spouštět více logických jednotek současně. Zmiňovanou nejmenší logickou jednotkou je obvykle vlákno (thread) nebo úloha (task), které soutěží o čas procesoru a jsou vykonávána plánovačem. Na rozdíl od obecných operačních systémů, jako je Linux, které typicky využívají procesy (složené z jednoho či více vláken) izolované pomocí virtuální paměti, mnoho RTOS, zejména těch určených pro malé vestavné systémy, pracuje bez této vrstvy abstrakce. V takovém případě všechny úlohy sdílejí stejný fyzický adresní prostor a nejsou mezi sebou paměťově izolovány. Existují však i výkonnější RTOS, jako například Zephyr RTOS, které virtuální paměť a izolaci procesů podporují. [33]

Plánovač je součástí jádra (kernelu), které představuje centrální komponentu RTOS a má na starosti správu běhu jednotlivých úloh (viz. obrázek 3.14) a přidělování procesorového času. Na rozdíl od jednoduchého sekvenčního provádění kódu v nekonečné smyčce mohou tyto úlohy běžet paralelně na základě priorit, časování a dalších parametrů. Kromě samotného plánování a přepínání mezi úlohami (tzv. přepínání kontextu) zajišťuje jádro také podporu pro synchronizaci a komunikaci mezi úlohami (např. semafory, fronty, mutexy), správu systémového času a obsluhu přerušení. [37]

Základními technickými výhodami RTOS jsou především deterministické chování, tedy schopnost reagovat na události ve striktně definovaném čase, a přesné řízení časování jednotlivých operací. To umožňuje systémům garantovat, že kritické úlohy budou vykonány ve stanovených časových limitech, bez ohledu na zátěž systému. Dále již byly nastíněny oblasti, ve kterých se operační systémy reálného času běžně uplatňují, mohou jimi být kritické systémy, jejichž selhání může mít katastrofální následky. Proto různé standardizované RTOS garantují vyšší bezpečnostní standardy a spolehlivější bezpečnostní funkce, například definované normami ISO 26262, DO-178C nebo IEC 61508. [34]

Naopak jedním z častých argumentů proti jejich použití je větší režie (overhead), která může snížit výkon a zvýšit paměťové nároky, čímž se zvyšují nároky na hardware. RTOS vnáší do systému složitost, která může ztížit jeho pochopení a ladění. Vývojáři musí být obeznámeni s rozhraním RTOS API a rozumět tomu, jak fungují úlohy, zdroje a komunikace mezi úlohami. Získání těchto znalostí může trvat delší dobu, což prodlužuje dobu vývoje a zvyšuje náklady. Použití RTOS může také zvýšit náklady na systém. Mnoho ko-

merčních dodavatelů RTOS si účtuje licenční poplatky, které mohou být pro malé projekty nákladné. [25]



Obrázek 3.14: Graf znázorňující rozdělení výpočetního času, při řízení programu operačním systémem reálného času s plánovačem [37]

## FreeRTOS

Jednou z implementací operačního systému reálného času je open-source řešení FreeRTOS, které je podobné vývoji s bare-metal přístupem. FreeRTOS podporuje mnoho různých architektur a komplikátorů a je navržen tak, aby byl malý, jednoduchý a snadno použitelný. Jeho jádro tvoří v základu tři oblasti – správa úloh, komunikace a propojení s hardwarem. Správa úloh se stará o vytváření úloh, jak statických, tak dynamických, tak i jejich plánování a správu životního cyklu. Dále FreeRTOS poskytuje oblast zaměřenou na komunikaci mezi jednotlivými úlohami, k tomu slouží fronty (queue). Úlohy a přerušení používají fronty k vzájemnému odesílání dat a k signalizaci použití kritických prostředků pomocí semafóru a mutexů. Poslední oblast slouží jako mezistupeň mezi hardwarově nezávislým jádrem FreeRTOS a hardwarově závislým kódem. [66]

Mezi hlavní výhody FreeRTOS patří jeho nízké nároky na systémové prostředky, což jej činí vhodným pro mikrokontroléry s omezenou pamětí a výpočetním výkonem. FreeRTOS poskytuje základní mechanismy pro plánování úloh, a to včetně prioritního plánovače s preempcí, který umožňuje přerušení aktuálně běžící úlohy ve prospěch jiné úlohy s vyšší prioritou. Velkou výhodou FreeRTOS je také jeho historie trvající přes 20 let, k dispozici je rozsáhlá dokumentace, příklady použití a současně mnoho výrobců, jako například NXP Semiconductors, poskytuje profesionální podporu. [37, 66]

Z hlediska bezpečnosti FreeRTOS dodržuje přísný standard psaní kódu v souladu s normou MISRA-C. Vybrané knihovny FreeRTOS byly navíc formálně ověřeny z hlediska paměťové bezpečnosti. Důraz na bezpečnost potvrzuje i certifikace, které hodnotí zabezpečení IoT platform z hlediska architektury, implementace a odolnosti vůči útokům. [35]

Přestože je FreeRTOS široce používán a oblíbený díky své jednoduchosti a nízké režii, nelze opomenout jeho omezení ve srovnání s robustnějšími RTOS, jakým je například Zephyr. Nevýhodou může být omezená podpora knihoven a celkově nižší flexibilita. Není

jednoduché jen tak přidat podporu na nový mikrokontrolér, jelikož portace je do značné míry jedinečná a velmi závislá na použitém procesoru a nástrojích. [5, 28]

### 3.7 Architektura systému digitálního záznamníku

V předchozích kapitolách byly představeny vybrané komponenty, které lze využít k sestavení digitálního záznamníku v podobě dedikovaného zařízení. Z představených vývojových desek byla jako nevhodnější zvolena deska FRDM-MCXN947 od společnosti NXP Semiconductors (viz. kapitola 3.2.1), která poskytuje dostatečný výkon, paměťový prostor a periferie, které lze využít k realizaci záznamníku.

Pro řízení činnosti záznamníku je využit operační systém reálného času FreeRTOS, přičemž architektura firmwaru je rozdělena do dvou hlavních úloh. První z nich představuje záznamovou úlohu, která zajišťuje průběžné získávání dat z UART periferie, jejich zpracování a jejich ukládání na paměťové médium. Druhou úlohou je komunikační úloha, která zodpovídá za zpřístupnění uložených dat prostřednictvím rozhraní USB ve standardu Mass Storage Class (MSC), címž odpadá nutnost fyzického vyjmutí paměťového média pro vyčtení zaznamenaných dat.

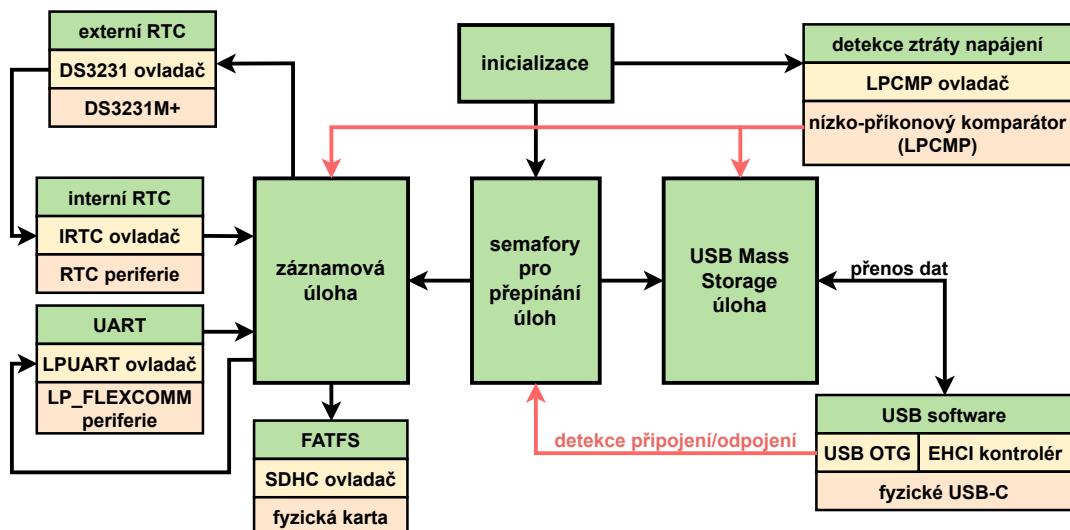
Firmware je koncipován tak, aby v každém okamžiku byla aktivní vždy pouze jedna z hlavních úloh, aby se minimalizovalo riziko konfliktu při přístupu k paměťovému médiu. Přepínání mezi úlohami bude řízeno detekcí připojení nebo odpojení USB rozhraní. V běžném režimu bude systém pracovat v záznamovém režimu a uchovávat příchozí data. Jakmile dojde k připojení zařízení k hostitelskému počítači, záznamová úloha se zastaví a aktivuje se komunikační úloha, která zpřístupní obsah úložiště jako běžný USB disk. Po odpojení USB se firmware automaticky vrátí do režimu záznamu.

Po spuštění firmwaru nejprve proběhne základní inicializace sdílených klíčových komponent. V rámci této fáze se nakonfigurují všechny piny využívané digitálním záznamníkem, inicializuje se ladící konzole a jednotlivým periferiím se přiřadí odpovídající hodinové signály. V této fázi se také nastaví analogový komparátor, který bude využit pro detekci ztráty napájecího napětí.

Po úspěšné inicializaci bude spuštěna jako výchozí a dominantní úloha, záznamová úloha, která zajistí příjem dat z externího zdroje. V úvodu si tato úloha nejprve inicializuje relevantní komponenty, které jsou využívány pouze v jejím kontextu. Mezi tyto komponenty patří zejména komunikační rozhraní, kterým vstupují data určená k zaznamenání, v případě varianty digitálního záznamníku pro společnost NXP Semiconductors se jedná o modul LP\_FLEXCOMM, nakonfigurovaný v režimu UART. V této inicializační fázi úlohy se mimo jiné naváže komunikace s externím RTC modulem prostřednictvím sběrnice I<sup>2</sup>C, ze kterého si mikrokontrolér přečte aktuální čas. Konkrétně se jedná o modul DS3231, který dokáže lépe uchovat čas, neboť obsahuje teplotně kompenzovaný oscilátor. Zjištěný čas bude následně uložen z externího do interního RTC modulu, jenž je součástí samotného mikrokontroléru. Práce s interním obvodem reálného času zrychlí přístup k systémovému času, jelikož další komunikace již nebude probíhat přes komunikační rozhraní I<sup>2</sup>C, ale přímo prostřednictvím registrů v rámci interní periferie. Po inicializační fázi se spustí již záznamová část, ve které budou přijatá data zpracovávána a ukládána do paměťového úložiště s využitím postupů popsaných v kapitole 2.4, přičemž bude kláden důraz na efektivní využití prostředků a minimalizaci rizika ztráty dat i při vyšších přenosových rychlostech. Správu a organizaci dat v paměťovém médiu bude zajišťovat hierarchický souborový systém FATFS, který je na rozdíl od LittleFS kompatibilní s většinou operačních systémů bez nutnosti dodatečného wrapperu.

Samotná zaznamenaná data budou ukládána na SD kartu typu SDHC, která byla zvolena z důvodu dostatečné paměti (až 32 GB), možnosti vytvoření standardního souborového systému a snadné zpětné dostupnosti dat prostřednictvím USB Mass Storage rozhraní. Přístup pomocí USB Mass Storage byl zvolen z důvodu, jelikož umožní hostitelskému systému přímý přístup k souborovému systému digitálního záznamníku, který bude reprezentován jako standardní externí úložiště. Uživatel tak bude mít zejména možnost přímé manipulace se soubory na paměťovém médiu a bude je moct snadno kopírovat, mazat nebo upravovat. V případě poškození souborového systému bude možné médium jednoduše připojit k hostitelskému zařízení a opravit ho pomocí běžně dostupných nástrojů pro správu disků.

Přepnutí ze záznamové úlohy do USB Mass Storage úlohy nastane, jakmile bude detekováno připojení k hostitelskému počítači a firmware pomocí mechanismu USB OTG ověří, že došlo ke skutečnému připojení a navázání komunikace. V reakci na tuto událost se zařízení automaticky přepne do režimu Mass Storage, ve kterém bude vnitřní paměťové úložiště zpřístupněno jako běžný externí disk. Díky tomuto automatickému přepínání není zapotřebí využít žádné ovládací prvky, jakými jsou tlačítka či přepínače. Tím se výrazně zjednoduší celková obsluha záznamníku, který bude často využíván nezkušenou obsluhou.



Obrázek 3.15: Výsledná architektura digitálního záznamníku

Důležitou součástí návrhu systému je rovněž detekce ztráty napájecího napětí, která umožní včasnu reakci na neočekávané odpojení napájení. K tomuto účelu bude využita periferie LPCMP (Low-Power Comparator), která je součástí mikrokontroléru MCXN947. Jakmile napájecí napětí poklesne pod stanovenou prahovou hodnotu, komparátor tento stav vyhodnotí a vygeneruje přerušení, které informuje aplikační vrstvu o hrozícím výpadku. Od tohoto okamžiku bude úkolem aplikace zajistit uložení všech rozpracovaných dat, která dosud nebyla zapsána do permanentního paměťového média, a také korektní uzavření otevřených souborů, aby bylo možné firmware bezpečně ukončit a předejít ztrátě nebo poškození dat.

# Kapitola 4

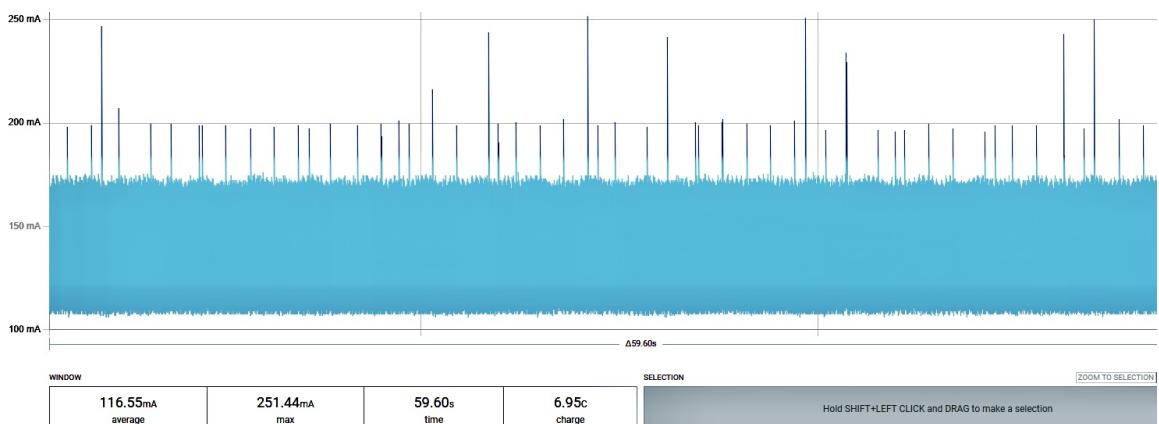
## Realizace hardwaru

### 4.1 Expanzní deska

Vybraná základová deska FRDM-MCXN947 neobsahuje všechny potřebné komponenty k realizaci digitálního záznamníku dle představené architektury v kapitole 3.7 a pro zajištění kompletní funkcionality je nutné doplnit celý systém o expanzní desku plošných spojů, která umožní integraci všech externích komponent a rozšíří tak funkciálnitu základové desky.

#### 4.1.1 Zálohované napájení

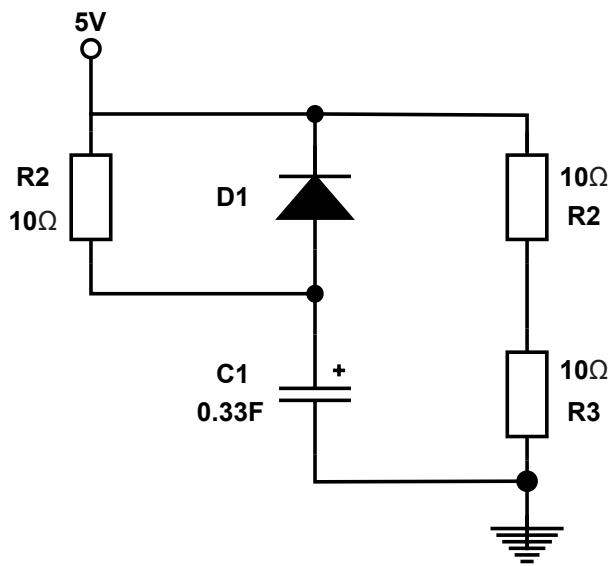
Důležitou součástí realizovaného digitálního záznamníku je zálohované napájení s detekcí ztráty napájecího napětí. Zálohované napájení lze realizovat několika způsoby, například pomocí Li-Po baterie, knoflíkového článku nebo pomocí kondenzátoru. Nejschůdnější variantou pro digitální záznamník se ukázal být kondenzátor, respektive superkondenzátor. Na rozdíl od baterií je schopen okamžitě dodat relativně vysoký vybíjecí proud, který byl naměřen při testování spotřeby pomocí zařízení Power Profiler Kit II od společnosti Nordic Semiconductors (viz. obrázek 4.1), při kterém se ukázalo, že vstupní proud může dosáhnout hodnot až 250 mA. Běžné knoflíkové články nebo malé Li-Po baterie by v tomto ohledu nemusely zajistit dostatečný proud a případně by při takové zátěži vedly k velmi krátké výdrži.



Obrázek 4.1: Výsledek měření spotřeby implementovaného záznamníku pomocí Power Profiler Kit II

Z tohoto důvodu bylo třeba vybrat superkondenzátor, který bude vyhovovat parametry pro využití v digitálním záznamníku. Mezi takové parametry patří zejména kapacita, která určuje, jak dlouho je zařízení schopno fungovat po ztrátě napájení. Dále bylo nutné zohlednit jmenovité napětí, které musí být vyšší než maximální napájecí napětí systému, aby nedošlo k poškození součástky.<sup>1</sup> Dalším důležitým faktorem je ekvivalentní sériový odpor (ESR), jenž ovlivňuje schopnost superkondenzátoru dodávat vysoký proud bez výrazného poklesu napětí, čímž nižší ESR, tím lépe je tedy kondenzátor schopen poskytovat vyšší proud. Vzhledem k omezenému prostoru na expanzní desce bylo zároveň nezbytné zohlednit fyzické rozměry kondenzátoru tak, aby jej bylo možné snadno integrovat a popřípadě umožnit, aby bylo možné využít dva kondenzátory paralelně pro zvýšení spolehlivosti systému v případě selhání jednoho z kondenzátorů.

Jako vhodný kondenzátor byl tedy vybrán model SCMR14G334SRBB0 od společnosti KYOCERA AVX s kapacitou 0.33 F, jmenovitým napětím 7.5 V při stejnosměrném proudu a nízkým ESR, které má konkrétní hodnotu 450 mΩ. Vzhledem k relativně nízké ceně této komponenty bylo třeba jej nejprve experimentálně ověřit v provizorním zapojení (viz. schéma 4.2) na nepájivém poli. Kondenzátor byl v tomto testu nejprve nabíjen z laboratorního zdroje nastaveného na 5 V odpovídající reálným provozním podmínkám v cílovém zařízení. Po jeho odpojení byla sledována vybíjecí křivka pomocí osciloskopu.



Obrázek 4.2: Obvod pro testování kondenzátoru

---

<sup>1</sup>Běžně je doporučováno volit dvojnásobek či dokonce trojnásobek napětí při kterém bude kondenzátor operovat.

Aby bylo možné vyhodnotit vhodnost kondenzátoru pro napájení digitálního záznamníku při výpadku napájecího napětí, je třeba nejprve vysvětlit základní skutečnosti související s vývojovou deskou FRDM-MCXN947. Na této desce je vstupní napětí 5 V vedené přes LDO (Low-Dropout Regulator), které upravuje napětí na 3.3 V pro napájení hlavního mikrokontroléru. LDO však ke své funkci vyžaduje minimální rozdíl mezi vstupním a výstupním napětím, tzv. Dropout Voltage ( $V_{Dropout}$ ), při kterém je tento integrovaný obvod ještě schopen stabilně regulovat výstupní napětí. Minimální vstupní napětí ( $V_{LDO\ In}$ ) potřebné pro stabilní výstup 3.3 V ( $V_{LDO\ Out}$ ) je dáno vztahem:

$$V_{LDO\ Min} = V_{LDO\ Out} + V_{Dropout}$$

V případě použitého stabilizátoru, který je využit na vývojové desce FRDM-MCXN947, je hodnota Dropout Voltage rovna 0.2 V a minimálním napětím, kterým tedy tato deska musí být napájena, činí 3.5 V. Součástí zapojení na schématu je také Schottkyho dioda, na které vzniká úbytek napětí, u použité součástky činí tento úbytek přibližně 0.4 V. Z těchto hodnot vyplývá, že je nutné sledovat napětí vybíjejícího se kondenzátoru v intervalu od 4.6 V do 3.5 V, které definuje časový interval, ve kterém by měl být provoz vývojové desky stabilní.

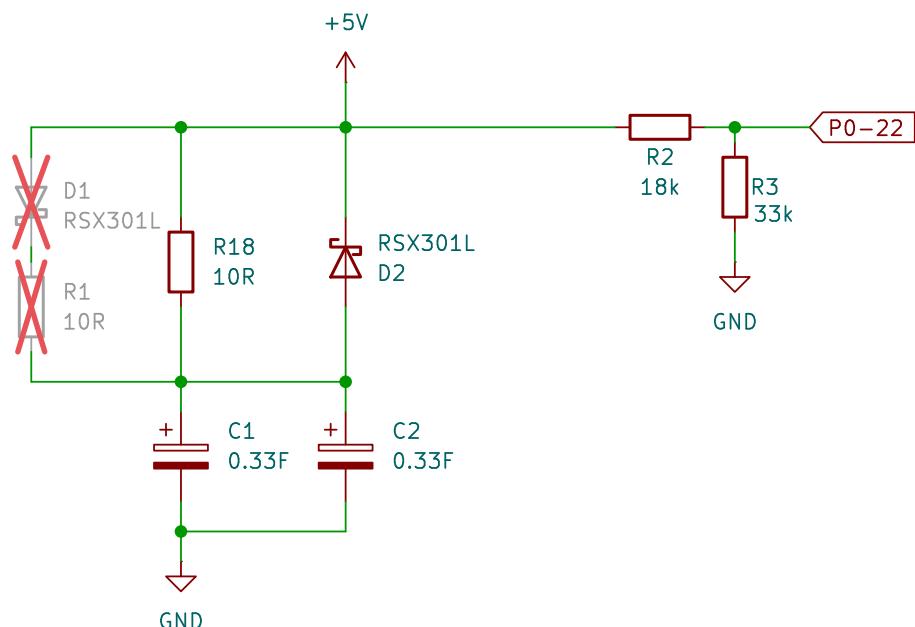
Následující obrázek 4.3, zachycuje průběh napětí z výše zmíněného testu, zaznamenaného pomocí osciloskopu, ze kterého je patrné chování napájecího napětí v čase. Z naměřených dat vyplývá, že napětí na kondenzátoru zůstává v požadovaném funkčním rozsahu od 4.6 V do 3.5 V po dobu přibližně 1.3 sekundy při odporu 20 Ω. Pokud tedy dojde ke včasné detekci ztráty napájejícího napětí, může dojít k bezpečnému ukončení programu a korektnímu uložení všech potřebných dat, která jsou dočasně uložena ve volatilní paměti RAM, což potvrzuje vhodnost zvoleného superkondenzátoru pro danou aplikaci zálohovaného napájení.



Obrázek 4.3: Průběh napětí na kondenzátoru při vybíjení zachycený osciloskopem

Na následujícím obrázku 4.4 je pak znázorněno výsledné schéma obvodu určeného k realizaci zálohovaného napájení včetně detekce ztráty hlavního napájecího napětí. Obvod zahrnuje zmíněnou Schottkyho diodu, která zajistuje zpětný tok proudu z kondenzátoru, a nabíjecí odpór s hodnotou 10 Ω, která byla zvolena tak, aby nabíjecí proud nepřekročil maximální proud povolený základním profilem napájení standardu USB-C, a zároveň aby nabíjení kondenzátoru probíhalo v přiměřeném čase.

Součástí navrženého obvodu zálohovaného napájení je také detekce ztráty napájecího napětí. Monitorování je realizováno přes napěťový dělič, protože sledovaná 5 V větev vývojové desky překračuje maximální přípustné napětí pro přímé připojení na vstupy mikrokontroléra pracujícího s logikou 3.3 V. Výstup z napěťového děliče je připojen na pin mikrokontroléra P0\_22, který umožňuje využití funkcí GPIO, komparátoru i analogově-digitálního převodníku.



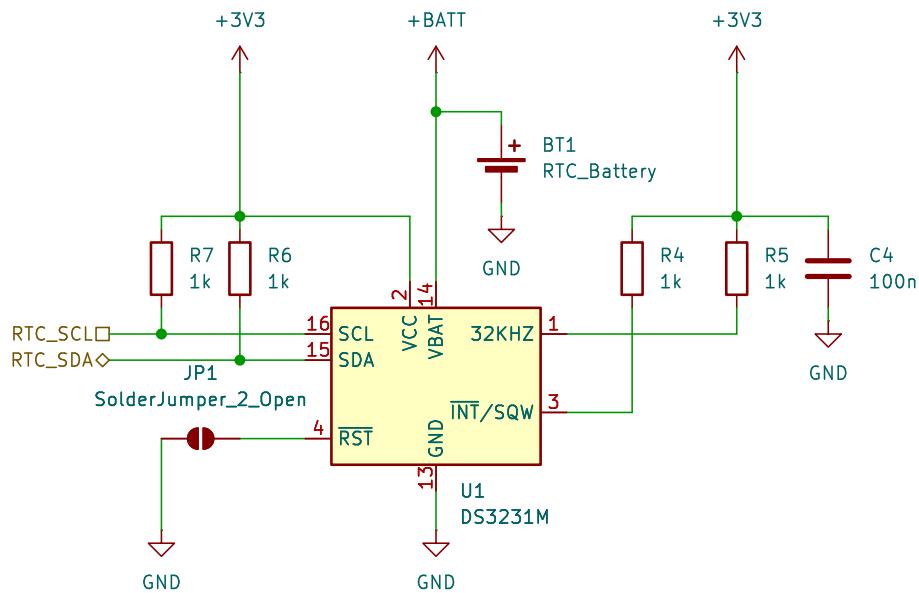
Obrázek 4.4: Schéma obvodu pro detekci ztráty napájení a zálohovaného napájení

#### 4.1.2 Obvod reálného času

Pro uchovávání informací o čase a datu je využit externí obvod reálného času, konkrétně čip DS3231M+ od společnosti Analog Devices. DS3231M+ patří mezi vysoko přesné RTC čipy obsahující integrovaný oscilátor s teplotně kompenzovaným krystalem (TCXO), který zajišťuje minimální odchylku chodu v rádu několika sekund za měsíc bez nutnosti externí kalibrace. Tato odchylka specifikovaná výrobcem činí  $\pm 2$  minut za rok v teplotním rozsahu od  $-40^{\circ}\text{C}$  do  $85^{\circ}\text{C}$ . [7]

Obvod reálného času s čipem DS3231M+ je integrován do expanzní desky plošných spojů (viz. obrázek 4.5) podle vzorového zapojení znázorněného v datasheetu k tomuto čipu. Pro komunikaci s tímto čipem slouží rozhraní I<sup>2</sup>C s využitím dvou signálových linek, kterými jsou SDA (Serial Data Line) a SCL (Serial Clock Line). Linka SDA slouží k přenosu dat mezi mikrokontrolérem a RTC obvodem, zatímco linka SCL zajišťuje synchronizaci komunikace pomocí hodinového signálu. Kromě komunikačních linek je nutné k čipu připojit napájení. Pin V<sub>CC</sub> slouží pro napájení čipu v běžném provozu, kdy je deska aktivně napájena. Pro zachování aktuální časové informace i během neaktivního stavu zařízení nebo během výpadku napájecího napětí je k dispozici pin V<sub>BAT</sub>, který je připojen ke knoflíkové baterii CR2032. [7]

Dále čip DS3231M+ obsahuje vývod 32KHZ, který slouží jako výstup stabilního hodinového signálu o frekvenci 32 kHz. Dále je k dispozici multifunkční pin INT/SQW, který může pracovat buď jako výstup čtvercového signálu (square wave), nebo jako přerušovací výstup v případě shody časových registrů s nastaveným alarmem. Signály 32KHZ a INT/SQW nejsou v implementaci digitálního záznamníku využívány, proto jsou oba připojeny pull-up odpory k zemi. Posledním relevantním signálem je RST, který slouží k resetování RTC obvodu. V návrhu expanzní desky je tento vývod signálu připojen k jumperu, jehož sepnutím lze čip ručně resetovat. [7]

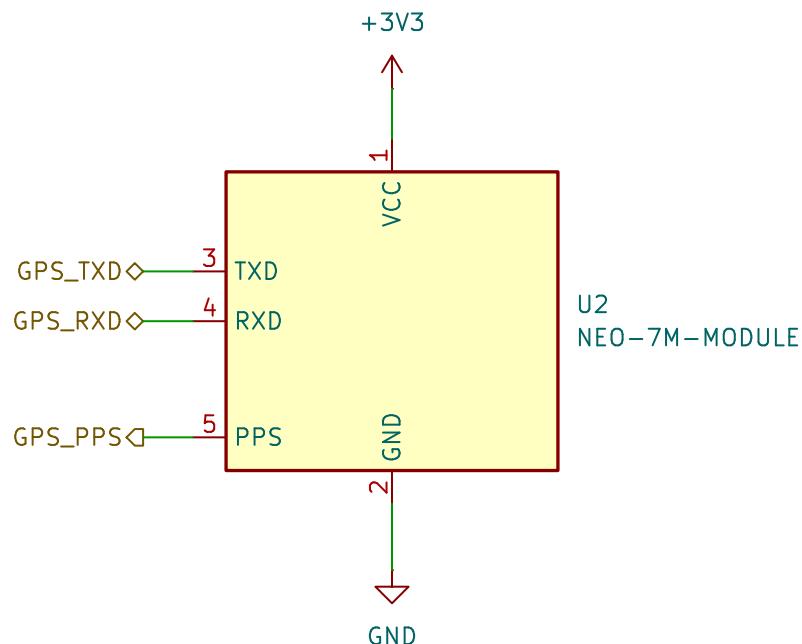


Obrázek 4.5: Zapojení obvodu reálného času – DS3231M+

#### 4.1.3 GPS pro synchronizaci času

Přestože integrovaný obvod reálného času DS3231M+ nabízí velmi dobrou dlouhodobou přesnost, stále jeho odchylka může činit až  $\pm 2$  minuty za rok. Mnoho aplikací vyžaduje ještě vyšší přesnost časové synchronizace, proto je expanzní deska připravena pro možnost připojení externího GPS modulu. Konkrétně se počítá s využitím modulu NEO-7M, který je schopen poskytovat přesné údaje o čase díky synchronizaci s globálním navigačním systémem. [13]

Modul NEO-7M obsahuje integrovaný přijímač, pasivní keramickou anténu a antenní zesilovač. Pokud zařízení má vyšší nároky na příjem signálu, je popřípadě možné pomocí SMA konektoru připojit aktivní anténu. Z následujícího obrázku 4.6 je patrné, že GPS modul NEO-7M je možné připojit k mikrokontroléru pomocí vývodů TXD a RXD, které slouží pro komunikaci prostřednictvím sériového rozhraní UART. Prostřednictvím tohoto rozhraní poskytuje modul datový výstup ve formátu NMEA (National Marine Electronics Association), který je široce používaný standardem pro přenos informací z navigačních zařízení. Mezi přenášené údaje patří aktuální čas v UTC, zeměpisná šířka a délka, nadmořská výška, rychlosť pohybu, směr pohybu, počet detekovaných satelitů, kvalita signálu a další navigační parametry. Modul NEO-7M poskytuje také speciální vývod PPS (Pulse Per Second), který generuje přesný časový impuls synchronizovaný s GPS systémem a umožňuje kompenzovat rozdíly v časování mezi různými zařízeními. V rámci návrhu této expanzní desky však vývod PPS není připojen k mikrokontroléru, neboť se počítá pouze s využitím GPS modulu ke synchronizaci času v RTC na základě dat přenášených přes rozhraní UART. [13]

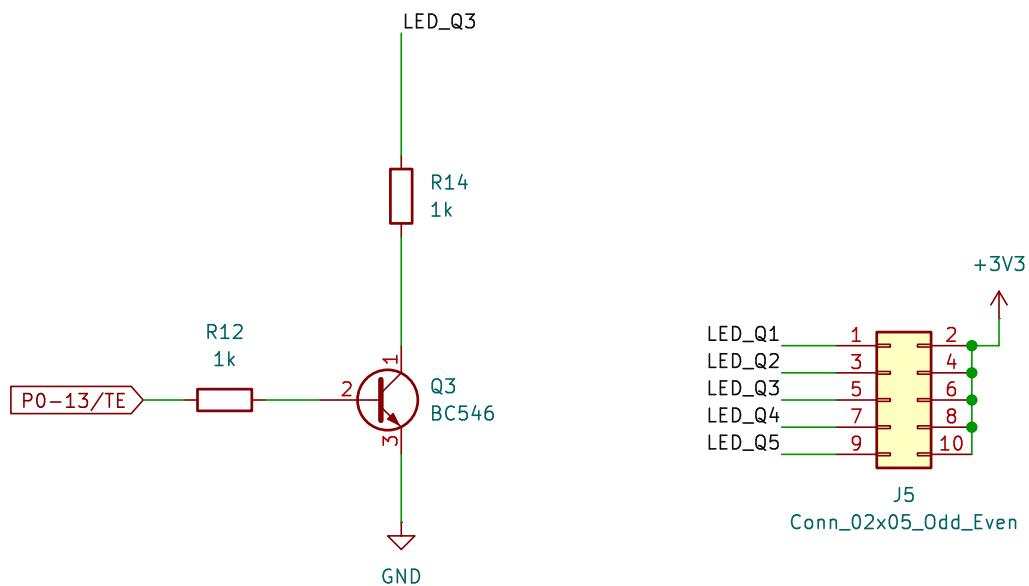


Obrázek 4.6: Připojení GPS modulu NEO-7M

#### 4.1.4 Signalizace stavu systému

Důležitou součástí digitálního záznamníku je také informování uživatele o běžném stavu systému, mezi které například patří informace o probíhajícím záznamu nebo dosažení bezpečného stavu pro odpojení zařízení od napájecího napětí. Signalizace je také nutná z důvodu výskytu případných chyb, které mohou nastat. K informování o těchto stavech slouží sada pěti světelných diod typu LED.

Každá dioda je připojena k bipolárnímu tranzistoru typu NPN (konkrétně BC546 od společnosti ON Semi), který slouží jako spínací prvek. Tranzistor je řízen malým proudem z mikrokontroléru prostřednictvím příslušného GPIO pinu, který umožňuje spínat proud tekoucí přes LED diodu. Pokud je tedy na bázi tranzistoru přítomen signál s logickou úrovní „1“, tranzistor se otevře a umožní průtok proudu skrze diodu a sériově zapojený omezovací rezistor, čímž dojde k jejímu rozsvícení. Schéma zapojení jedné z těchto LED diod je znázorněno na obrázku 4.7. Všechny diody jsou napájeny napětím 3.3 V a proud tekoucí jednotlivými LED je omezen odpovídajícími rezistory, přičemž jednotlivé diody je pak možné připojit do pinové lišty (pin header).



Obrázek 4.7: Ukázka zapojení jedné z kontrolních diod typu LED

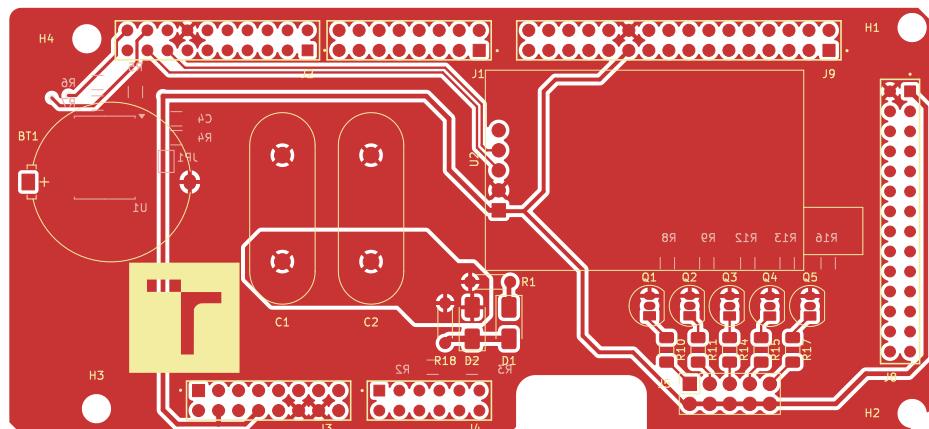
#### 4.1.5 Layout expanzní desky

Výše zmíněné obvody byly navrženy na dvouvrstvou desku plošného spoje v návrhovém editoru KiCAD 8, jejíž rozměry i rozmístění pinových lišť přesně odpovídají vývodům konektorů vývojové desky FRDM-MCXN947 od společnosti NXP Semiconductors.

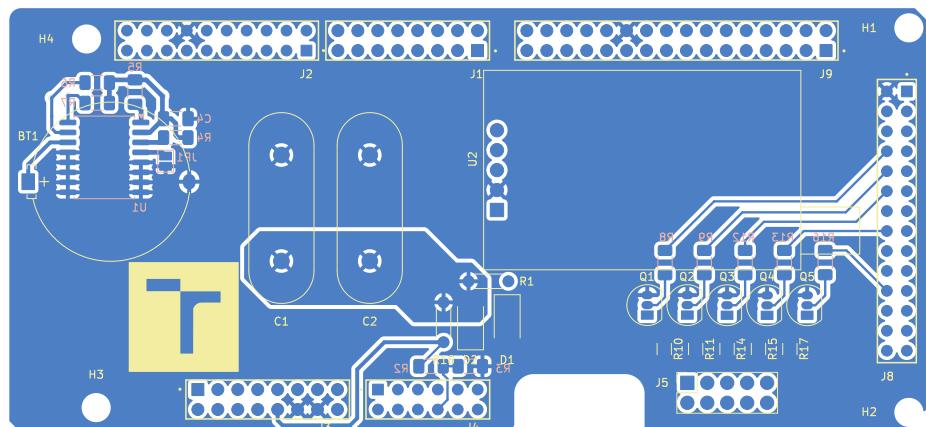
Ve středu desky se nachází logika zálohovaného napájení s detekcí ztráty napájecího napětí a také jsou zde otvory pro přichycení GPS modulu NEO-7M. Připojení GPS modulu je zamýšleno tak, aby mohl být modul připojen pomocí 90°zahnuté pinové lišty a uložen rovnoběžně s rovinou expanzní desky a to z důvodu omezení rozměrů celé sestavy v krytu. Modul NEO-7M sice poskytuje pasivní anténu, nicméně zařízení bude využíváno ve vnitřních prostorách, kde může být nedostačující signál, a proto se počítá s připojením aktivní antény, která má větší rozměry a bude vyvedena skrz kryt zařízení.

V levém spodním rohu se pak nachází pinová lišta pro umístění signalizačních diod typu LED spolu s obvody pro jejich spínání. Naopak v pravém horním rohu se nachází obvod s RTC a vývrtkami pro umístění držáku knoflíkové baterie.

Krom samotných obvodů deska plošného spoje obsahuje čtyři montážní otvory (mounting holes), které slouží k přichycení distančních sloupků, pomocí kterých může být deska fixně zasazena do krytu záznamníku.



Obrázek 4.8: Layout expanzní desky – přední strana



Obrázek 4.9: Layout expanzní desky – spodní strana

#### 4.1.6 Cenová kalkulace součástek expanzní desky

Důležitým parametrem expanzní desky je i její cena. Celková cena komponent použitých na realizaci expanzní desky k datu 25.2.2025 činí přibližně 880 Kč. V následující tabulce 4.1 je uveden přehled součástek použitých při návrhu digitálního záznamníku, včetně jejich orientačních cen.

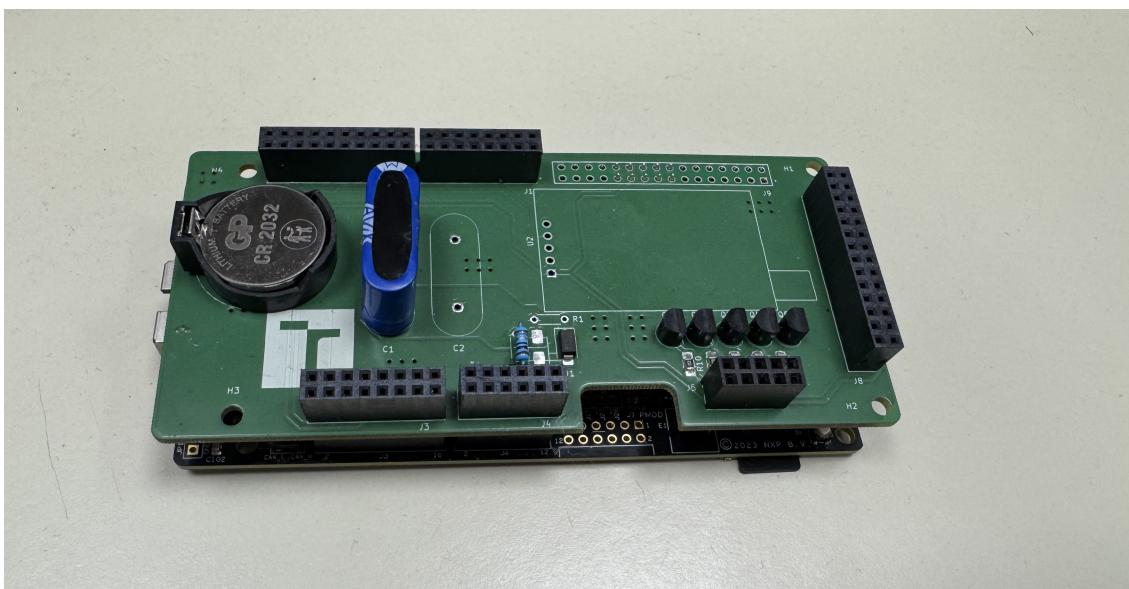
Nutno dodat, že doporučené pin headery od výrobce Samtec nejsou nezbytně nutné a lze je nahradit levnějšími alternativami. Všechny komponenty byly vybírány tak, aby je bylo možné objednat u jednoho distributora elektronických součástek z důvodu zjednodušení procesu objednávky a zároveň minimalizace nákladů na dopravu.

Tabulka 4.1: Přehled použitých součástek

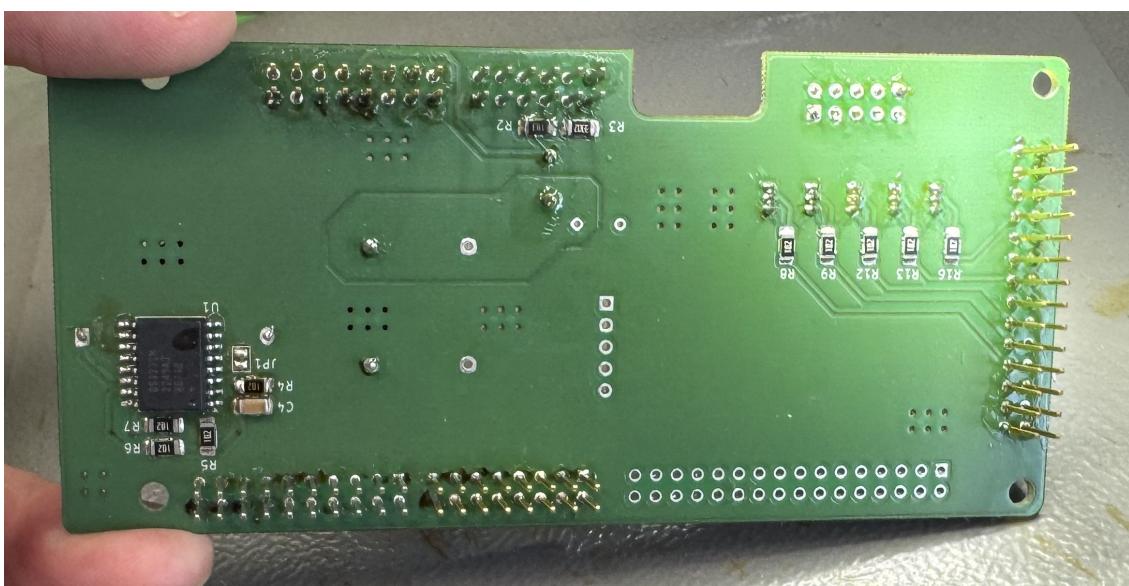
Součástka	Hodnota / Popis	Počet	Cena za kus
SSQ-108-03-T-D	16-pinová lišta (Samtec)	2	46,61 Kč
SSQ-106-03-T-D	12-pinová lišta (Samtec)	1	40,60 Kč
SSQ-114-03-T-D	28-pinová lišta (Samtec)	1	81,70 Kč
SSQ-116-03-T-D	32-pinová lišta (Samtec)	1	93,22 Kč
SSQ-110-03-T-D	20-pinová lišta (Samtec)	1	58,39 Kč
M20-7830342	10-pinová lišta (Harwin)	1	34,33 Kč
DS3231M+	Obvod reálného času	1	284,68 Kč
SCMR14G334SRBB0	Superkondenzátor (KYOCERA AVX)	1	141,09 Kč
RSX301L-30DDTE25	Schottky dioda (ROHM)	1	23,58 Kč
BC546	NPN tranzistor (ON Semi)	5	5,71 Kč
Bateriový box CR2032	–	1	–
THT rezistor	10 Ω	1	–
SMD rezistor R1206	18 kΩ	1	–
SMD rezistor R1206	33 kΩ	1	–
SMD rezistor R1206	1 kΩ	14	–
SMD kondenzátor R1206	1 nF	1	–

#### 4.1.7 Výsledná expanzní deska

Výsledkem je zhotovená expanzní deska, na kterou byly připájeny všechny navržené komponenty. SMD součástky, jako jsou rezistory a kondenzátory, byly osazeny pomocí pájecí pasty, která byla nanесена на příslušná pájecí pole. Po jejich umístění byla deska zapájena reflow procesem v pájecí peci. Zbylé součástky s vývody typu THT (Through-Hole Technology), jako jsou pinové lišty, nabíjecí odpor, držák knoflíkové baterie a superkondenzátory pro zálohované napájení, byly připájeny ručně pomocí páječky.



Obrázek 4.10: Výsledná expanzní deska – přední strana



Obrázek 4.11: Výsledná expanzní deska – spodní strana

## Kapitola 5

# Přístup k realizaci obslužného firmware

Obslužný firmware digitálního záZNAMníku byl implementován v jazyce ISO C99, přičemž během vývoje firmwaru byl kladen rovněž důraz na dokumentaci zdrojového kódu. Všechny funkce, globální proměnné a makra jsou opatřeny komentáři v anglickém jazyce ve formátu *Doxxygen*, ze kterých je vygenerována HTML dokumentace v adresáři `application/doc/`. Krom *Doxxygen* komentářů je i samotný kód doplněn o kontextové komentáře, které objasňují klíčové části implementace.

Zároveň pro podporu dobré čitelnosti kódu a snazší orientaci byla kromě dokumentace kladena pozornost také na jednotný systém pojmenovávání proměnných. Pojmenování proměnných reflektuje datový typ, rozsah a případnou strukturu. Globální proměnné jsou označeny prefixem `g_`, címž se jednoznačně odlišují od lokálních. Datový typ proměnné je dále specifikován typovým prefixem, například proměnná typu `uint32_t` obsahuje prefix `u32`, `boolean` je označen prefixem `b`, ukazatele jsou označeny prefixem `p` a pole jsou označeny prefixem `a`. Proměnné typu struktura (pokud nejsou globální) jsou pojmenovávány bez prefixu. Tyto prefixy jsou vzájemně kombinovány, aby správně reprezentovaly danou proměnnou. Příklad pojmenování proměnných může vypadat následovně:

```
/**  
 * @brief Globální proměnná, ukazatel na hodnotu typu uint32_t  
 * @details g_ = globální, p = pointer, u32 = uint32_t  
 */  
static uint32_t* g_pu32CurrentCount;  
  
/**  
 * @brief Lokální proměnná typu struktura  
 * @details bez prefixu  
 */  
recordingStatus_t recordingStatus;
```

Samotný firmware digitálního záznamníku je postaven na operačním systému reálného času FreeRTOS, který zajišťuje deterministické plánování jednotlivých úloh. Firmware pro zvýšení spolehlivosti a predikovatelnosti využívá výhradně statickou alokaci paměti. Veškeré úlohy i synchronizační prostředky jsou vytvářeny s předem definovanou velikostí a dynamická alokace paměti je zcela vypnuta na úrovni FreeRTOS. Současně pro umožnění provozu systému výhradně se statickou alokací bylo nezbytné implementovat specifické funkce pro přidělení paměti systémovým úlohám, kterými jsou Idle Task a Timer Task. [37]

Aplikační vrstva firmwaru je strukturována do tří hlavních souborů. Inicializaci hardwaru a konfiguraci systému zajišťuje *app\_init.c*. Dále všechny aplikační FreeRTOS úlohy jsou implementovány v *app\_task.c*, přičemž samotná inicializace, vytvoření úloh a jejich spuštění je implementováno v souboru *main.c*.

Firmware digitálního záznamníku je navržen modulárně a je tedy rozdelen do několika funkčních celků, například logika záznamu, vyčtení zaznamenaných dat pomocí Mass Storage, detekce ztráty napájení, signalizace stavu systému a další, přičemž mnoho z nich je možné dle potřeby samostatně aktivovat či deaktivovat. Tohoto chování je dosaženo pomocí podmíněného překladu v jazyce ISO C, kdy jsou jednotlivé části kódu překládány pouze v případě, že jsou odpovídajícím způsobem povoleny. K centrální správě těchto voleb slouží aplikační konfigurační soubor *defs.h*.

Součástí konfiguračního souboru *defs.h* je rovněž možnost zvolit si, jakou mírou informačního výstupu bude firmware komunikovat k vývojáři.<sup>1</sup> Firmware digitálního záznamníku podporuje tři úrovně výstupu zpráv. První, nejnižší úrovní, jsou chybová hlášení (zprávy začínající řetězcem **ERR:**), která informují o kritických stavech. Tato vrstva je automaticky aktivní a nelze ji vypnout. Druhou vrstvou je informační režim (zprávy začínající řetězcem **INFO:**), ve kterém aplikace poskytuje podrobnější výpis o právě probíhajících událostech, jako je například spuštění záznamu, nalezení konfiguračního souboru či inicializace periferií. Třetí, nejpodrobnější úrovní, je ladící režim (zprávy začínající řetězcem **DEBUG:**), ve kterém aplikace rozšiřuje výstup o doplňující technické detaily, které mohou být užitečné zejména při ladění během vývoje. Veškeré tyto výstupy jsou poskytovány skrze sériové rozhraní UART. K tomuto účelu je využívána konkrétně instance periferie LP\_FLEXCOMM4, která je nakonfigurována na přenosovou rychlosť 115200 baudů, 8 datových bitů, 1 stop bit a bez použití parity.

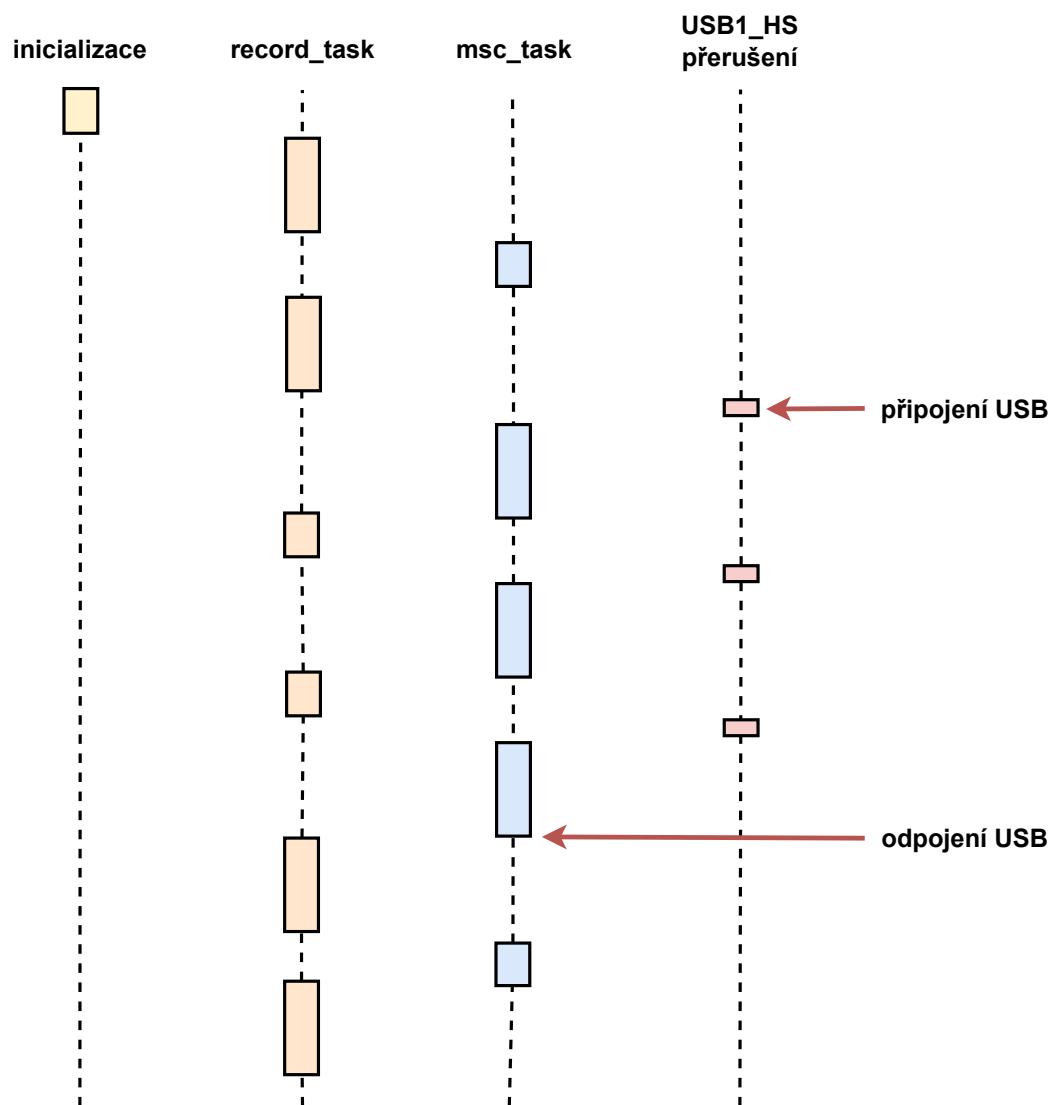
Taktéž je možné v konfiguračním souboru *defs.h* nastavit pevné velikosti zásobníků jednotlivých statických úloh, priority využívaných přerušení či výchozí parametry záznamu, které budou dále rozvedeny v kapitole 5.1.

Po inicializaci firmwaru jsou vytvořeny dvě hlavní úlohy, mezi které patří *record\_task* určený k záznamu dat a *msc\_task* pro vyčtení zaznamenaných dat pomocí USB Mass Storage. Tyto úlohy jsou vzájemně přepínány na základě dvou binárních semaforů. Po startu systému obě úlohy běží paralelně a žádná z úloh není explicitně uspávána. V každém okamžiku je plná funkcionality umožněna pouze záznamové úloze, přičemž úloze *msc\_task* je tento přístup omezen pomocí semaforu. Plná funkcionality záznamové úlohy je umožněna až do situace připojení zařízení k hostitelskému počítači prostřednictvím USB-C, kdy dojde k vyvolání přerušení od *USB1\_HS*, které je zpracováno příslušnou obsluhou přerušení. V rámci této rutiny je pomocí stavových registrů USB OTG řadiče ověřeno, že skutečně došlo k připojení k hostiteli.

---

<sup>1</sup>Pro koncového uživatele je naopak stav zařízení signalizován jednoduše pomocí LED indikátorů.

Je-li připojení potvrzeno, příslušný semafor je uvolněn ve prospěch *msc\_task* úlohy, která následně zpřístupní obsah paměťové karty jako externí úložiště ve formátu USB Mass Storage. Během této doby je záznamová úloha upozaděna a čeká na opětovné získání semaforu. Po ukončení spojení s hostitelským zařízením je stav odpojení detekován opět pomocí registrů USB OTG, tentokrát přímo v rámci běžící úlohy *msc\_task*. Jakmile je odpojení potvrzeno, řízení je převedeno zpět na záznamovou úlohu.



Obrázek 5.1: Zjednudušený diagram běhu firmwaru digitálního záznamníku, včetně přepínání úloh

## 5.1 Záznamová úloha

Samotná záznamová úloha je implementována v souboru *app\_tasks.c*, přičemž využívá modul, poskytující funkce k záznamu dat, který je implementován v souborech *record.c* a *record.h*.

Inicializační fáze záznamové úlohy v prvním kroku zajišťuje inicializaci SD karty pomocí funkce **USB\_DeviceModeInit()**, která v případě potřeby zajistí i její formátování. Bezprostředně po této inicializaci proběhne synchronizace systémového času, při které je aktuální čas získán z externího RTC modulu a uložen do interního RTC modulu mikrokontroléru, jak je podrobně popsáno v kapitole 3.7. Po úspěšném přenosu časových údajů je externí RTC modul odpojen, a dále je již systémový čas řízen výhradně pomocí interního RTC. Následně funkce **CONSOLELOG\_Init()** provede připojení SD karty, nastaví aktuální pracovní jednotku a ověří existenci souborového systému. Pokud není souborový systém nalezen, automaticky provede jeho vytvoření. Poté je vytvořen hlavní relační adresář (session directory) pomocí funkce **CONSOLELOG\_CreateDirectory()**, do kterého budou ukládány záznamy z dané relace, která trvá až do vypnutí digitálního záznamníku.

Během inicializační fáze záznamové úlohy jsou také nastaveny parametry záznamu prostřednictvím konfiguračního souboru *config*, který je očekáván v kořenovém adresáři souborového systému přímo na SD kartě digitálního záznamníku. Jak již bylo zmíněno, zařízení je určeno pro záznam sériových dat, respektive jedno-bajtových znaků přenášených prostřednictvím rozhraní UART, přičemž lze nastavit rychlosť přenosu přes rozhraní (parametr **baudrate**), počet datových bitů (**data\_bits**), počet stop bitů (**stop\_bits**) a paritu (**parity**). Parametr **baudrate** podporuje typické hodnoty přenosové rychlosti standardní pro UART komunikaci, jako například 9600, 19200, 38400, 57600 či 115200 a další. Počet datových bitů je možné volit mezi 7 a 8, stop bity mezi 1 a 2. Parametr **parity** může nabývat hodnot **none**, **even** nebo **odd**.

Kromě nastavení komunikačních parametrů je možné nastavit maximální velikost záznamu (záznamového souboru) prostřednictvím parametru **file\_size**. U velikosti záznamu se počítá s násobkem 512 bajtů, neboť data jsou ukládána na SD kartu po blocích (viz. kapitola 2.4.2). Nicméně je možné zvolit i jinou hodnotu, přičemž firmware ji automaticky zaokrouhlí směrem nahoru na nejbližší přípustný násobek.

Posledním parametrem je **free\_space**, který slouží k definování minimálního množství volného místa (v MB) na SD kartě. Pokud aktuální volná kapacita klesne pod definovanou hodnotu, digitální záznamník tuto skutečnost signalizuje rozsvícením příslušné diody typu LED. Digitální záznamník tak informuje uživatele o blížícím se zaplnění paměti. Pokud je parametr **free\_space** nastaven na hodnotu 0, je tato signalizace deaktivována.

Parametry v konfiguračním souboru nemají pevně dané pořadí, přičemž ukázkový obsah může vypadat následovně:

```
baudrate=115200
file_size=2048
stop_bits=1
data_bits=8
parity=none
free_space=50
```

V případě, že konfigurační soubor *config* není nalezen v kořenovém adresáři nebo se jeho obsah nepodaří záznamníku zpracovat, záznam zařízení proběhne dle předdefinovaných parametrů uvedených v hlavičkovém souboru *defs.h*. Předdefinované parametry z hlavičko-

vého souboru jsou automaticky zvoleny i v případě, kdy je konfigurační soubor nalezen, ale některý z očekávaných parametrů v něm chybí. Přehled všech výchozích hodnot shrnuje následující tabulka 5.1.

Parametr	Výchozí hodnota	Význam
DEFAULT_BAUDRATE	230400	Výchozí přenosová rychlosť sériové komunikace.
DEFAULT_DATA_BITS	8 datových bitů	Výchozí počet datových bitů pro sériovou komunikaci.
DEFAULT_STOP_BITS	1 stop bit	Výchozí počet stop bitů pro sériovou komunikaci.
DEFAULT_PARITY	Bez parity	Výchozí nastavení parity.
DEFAULT_MAX_FILESIZEx	8 MB	Maximální velikost záznamového souboru v bajtech.
DEFAULT_FREE_SPACE	50 MB	Prahová hodnota volné kapacity SD karty, pod kterou je signalizováno docházející místo.

Tabulka 5.1: Výchozí hodnoty konfiguračních parametrů

Pro zpracování konfiguračního souboru slouží vlastní parser implementovaný v souborech *parser.c* a *parser.h*, a je možné jej snadno rozšířit o podporu dalších parametrů. Každý parametr je zpracováván samostatnou funkcí. V případě rozšíření stačí deklarovat funkci v hlavičkovém souboru *parser.h*, implementovat ji ve zdrojovém souboru *parser.c* a následně ji zavolat ve funkci **CONSOLELOG\_ProccessConfigFile()**, která konfigurační soubor zpracovává při inicializaci záznamové úlohy.

Po přípravné fázi následuje samotná logika záznamu, ta je složena ze vnější a vnitřní smyčky. Vnější smyčka reprezentuje řídící smyčku úlohy, která čeká na aktivaci pomocí synchronizačního mechanismu, tedy binárního semaforu, zatímco vnitřní smyčka zprostředkovává záznam dat, ukládání dat do SD karty a případně signalizuje stav záznamu uživateli.

Samotný záznam dat je založen na konceptech představených v kapitole 2.4. V prvé řadě je jako vstupní kanál, kterým data vstupují do digitálního záznamníku, využívána periférie LP\_FLEXCOMM (Low-Power Flexible Communications Interface) s přerušením. Zaznamenané jednotlivé znaky jsou v obslužné rutině přerušení vkládány do cirkulární vyrovnávací paměti (viz. kapitola 2.4.3), se kterou je aktualizována hodnota zápisového ukazatele a také hodnota proměnné, která v sobě uchovává počet tiků, ve kterém data byla zaznamenaná na základě systémového časového údaje vyjádřeného v tikách operačního systému.

Data jsou následně v běhu záznamové úlohy vkládána z cirkulárního bufferu do jedné ze dvou vyrovnávacích pamětí, u kterých se využívá koncept vícenásobné vyrovnávací paměti (viz. kapitola 2.4.1), přičemž jedna paměť vždy slouží ke sběru a předzpracování přicházejících dat (Back Buffer), zatímco druhá slouží k jejich ukládání na paměťovou kartu prostřednictvím řadiče SDHC (Front Buffer). Role těchto dvou pamětí jsou mezi sebou cyklicky prohazovány na základě jednoduché výměny ukazatelů. Znaky jsou tedy vkládány do Back Buffer a při vkládání je každý nový znak porovnáván s předchozím tak, aby bylo možné detektovat přítomnost znakové dvojice CR+LF z prostředí operačního systému Windows CR+LF (Carriage Return Line Feed). Tento bílý znak je detekován z toho důvodu, že monitorovaná bezdrátová nabíječka používá znakové dvojice CR+LF pro ukončení řádku. Díky jeho rozpoznání je tak možné jednoznačně identifikovat začátek nového řádku. V případě, že je tato znaková sekvence zachycena, je z interního obvodu reálného času mikrokontroléru MCXN947 získán aktuální čas, který je následně vložen jako časová značka na začátek právě detekovaného řádku. Výsledný formát zaznamenaného textu po doplnění časových značek vypadá následovně:

```
(12:34:56) Starting system diagnostics...
(12:34:57) Sensor reading: 23.5°C
(12:34:58) Data transmission OK
(12:34:59) System idle
```

Back Buffer je tímto způsobem postupně plněn záznamovými daty a doplňován o časové značky s aktuálním časem, až do okamžiku, kdy je dosažena jeho kapacitní hranice. V ten moment dojde k záměně rolí vyrovnávacích pamětí. Paměť, která dosud sloužila ke sběru dat a vkládání časových značek, je nyní určena k zápisu dat na SD kartu, a naopak paměť, která byla vyhrazená pro zápis, je určena pro naplnění novými daty. Samotný zápis dat do paměťového média je realizován neblokujícím způsobem prostřednictvím transportního algoritmu s přímým přístupem do paměti ADMA (Advanced Direct Memory Access). [15]

Pokud dojde zaplnění záznamového souboru, jehož velikost je definována pametrem `file_size`, je tento soubor uzavřen, a následně je vytvořen nový soubor pomocí funkce `CONSOLELOG_CreateFile()`. Funkce `CONSOLELOG_CreateFile()` vytváří unikátní název souboru na základě aktuálního data a času ve formátu `YYYYMMDD_HHMMSS_X.txt`, kde `X` je inkrementální čítač a následně tento soubor otevře pro zápis, nastaví jeho metadata (čas vytvoření) pomocí dat z interního RTC a záznam tak může pokračovat.

Součástí návrhu je také ošetření okrajových stavů, které by mohly ohrozit konzistenci vkládání časových značek do záznamu. Typickým případem je rozdelení sekvence znaků, kdy první speciální znak CR se může dostat na konec vyrovnávací paměti plnící roli Back Buffer, následně dojde k prohození vyrovnávacích pamětí a druhý speciální znak LF se ocitne již na začátku nového Back Bufferu. V takové situaci by se mohlo stát, že začátek řádku by nebyl detekován a časová značka by nebyla vložena. Aby bylo této chybě zabráněno, jsou uchovávány jakékoli po sobě jdoucí znaky, a tedy k této situaci nedojde. Druhou situací je případ, kdy se sekvence CR+LF nachází na samotném konci aktivního bufferu a do dané paměti se již nevejde celá časová značka. Časová značka je v takovém případě rozdělena, část je uložena do konce původního bufferu a zbývající část je po výměně bufferů doplněna do nově aktivní paměti.

Další specifickou situací je potřeba korektního ukončení záznamu, tu lze například řešit pomocí tlačítka, nicméně rozhodl jsem se tuto situaci řešit obdobně jako situaci, kdy dojde ke ztrátě spojení s monitorovaným zařízením, která je z pohledu popisovaného algoritmu kritická. Pokud v takovém případě není Back Buffer plně zaplněn, jeho obsah se neuloží do permanentního úložiště, jelikož obsah Back Bufferu uvázne ve volatilní paměti digitálního záznamníku. Pro obě zmíněné situace je využívána funkce `CONSOLELOG_Flush()`, která je periodicky volána a zajíšťuje pravidelnou kontrolu. Ta je určena pomocí porovnání aktuálního systémového času, získaného funkcí `xTaskGetTickCount()`, s hodnotou času, kdy byl přijat poslední znak. Pokud rozdíl těchto hodnot přesáhne mez definovanou makrem `FLUSH_TIMEOUT_TICKS`, které je v základě nastaveno na dobu tří sekund, zbývající volná část Back Bufferu je doplněna bílými znaky v podobě mezery a jeho obsah je uložen na SD kartu a záznamový soubor je zároveň uzavřen.

## 5.2 USB Mass Storage úloha

Druhou úlohou je USB Mass Storage, která slouží ke zpřístupnění obsahu paměťové SD karty připojenému hostitelskému zařízení prostřednictvím rozhraní USB. Tato úloha se aktivuje ve chvíli, kdy je zařízení připojeno k hostiteli přes rozhraní `USB1_HS`.

Funkcionalita USB Mass Storage je mnohdy poskytována samotnými výrobci mikrokontrolérů jako součást vývojových balíků s cílem usnadnit vývoj aplikací. Společnost NXP Semiconductors nabízí v rámci svého MCUXpresso SDK (Software Development Kit) USB Mass Storage se zpřístupněním obsahu SD karty připojenému hostitelskému zařízení.

USB Mass Storage bylo třeba zaintegrovat do firmwaru digitálního záznamníku, ve kterém je veškerá paměť předalokována staticky. Samotná implementace tohoto protokolu (tedy software třetí strany) je v rámci projektu umístěna v adresáři `msc/`. Nad tímto základem je pak postavena aplikační část, která je umístěna v rámci složky `source/` soubor `mass_storage.c`, obsahující aplikaci USB Mass Storage, kterou je možné rozšířit. Z pohledu Mass Storage je také důležitá obslužná rutina přerušení `USB1_HS_IRQHandler()`, která zprostředkovává komunikaci a přenos dat mezi hostitelským zařízením.

USB Mass Storage využívá obecný USB modul implementovaný se záměrem kompatibility s velkou řadou mikrokontrolérů. Z tohoto důvodu jsem se rozhodl modul optimalizovat. Poskytnutý Mass Storage poskytuje vícero rozhraní hostitelských řadičů (Host Controller Interface – HCI) jakým je například KHCI, a speciální rozhraní pro rodinu LPC, které jsou nekompatibilní s mikrokontrolérem MCXN947, proto byla odstraněna, a v případě přechodu na jiné MCU lze rozhraní zaměnit.

## 5.3 Detekce ztráty napájecího napětí

Z důvodu prevence ztráty dat a minimalního rizika poškození souborového systému v důsledku náhlého výpadku napájení je v systému implementován mechanismus pro detekci poklesu napájecího napětí, který je realizován pomocí analogového komparátoru a pomocného časovače.

Analogový komparátor je přítomen na MCU MCXN947 v podobě periferie LPCMP a ve firmwaru slouží ke sledování poklesu na 5 V větvi, na které je respektive při napájení hodnota napětí 4.6 V, z důvodu úbytku napětí na Schottky diodě (viz. kapitola 4.1.1).

Napětí z této větve je následně přivedeno na vstup analogového komparátoru prostřednictvím napěťového děliče. Komparátor musí reagovat co nejdříve, aby MCU dostalo co nejvíce času na uložení dat do SD karty, proto je hodnota napětí, při které bude komparátor reagovat, zvoleno napětí 4.5 V, pokud je tato hodnota přepočítána na hodnotu napětí, které je za napěťovým děličem, vyjde hodnota 2.911 76 V, jelikož:

$$V_{diff} = 4.5 \text{ V} * \frac{33 \text{ k}\Omega}{18 \text{ k}\Omega + 33 \text{ k}\Omega} = 2.91176 \text{ V}$$

Perifie LPCMP obsahuje interní DAC (Digital-Analog Converter), díky kterému je možné porovnávat vstupní analogový signál s 256 respektive 255 úrovněmi referenčního napětí, kterým je signál **VDD\_ANA** s napětím 3.3 V. Z tohoto důvodu je hodnota v DAC, při které bude komparátor reagovat, nastavena na hexadecimální hodnotu 0xE1, jelikož:

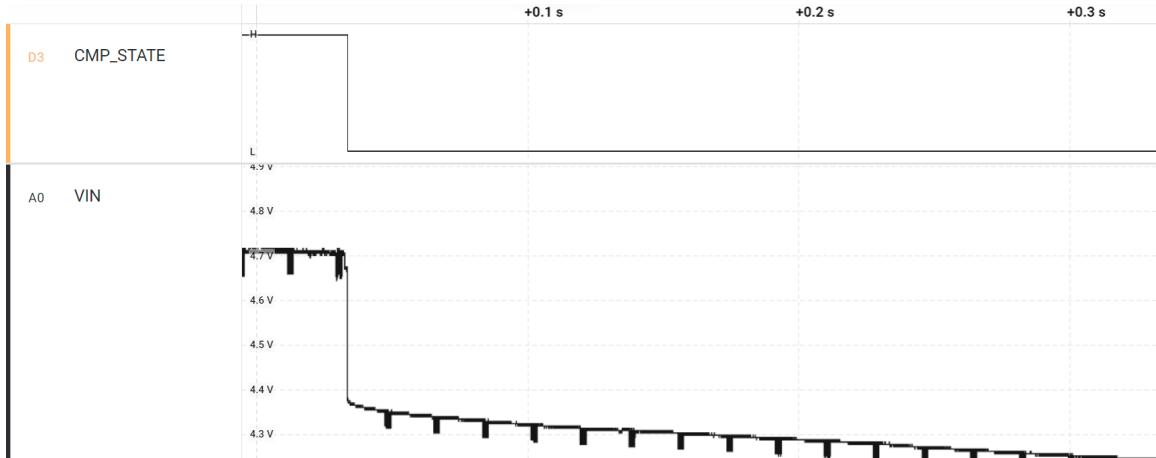
$$DAC_{level} = 255 * \frac{2.91176 \text{ V}}{3.3 \text{ V}} = 225.125 \approx 0xE1$$

Reakce komparátoru je rovněž ovlivněna hysterézí, která je nastavena na hodnotu 30 mV aby byly omezeny falešné detekce. Aby bylo zajištěno, že detekce ztráty napájení bude aktivní až po dostatečném nabité zálohovacího kondenzátoru, je součástí systému také časovač, který zpozdí povolení detekce. Funkce komparátoru je aktivována po uplynutí doby odpovídající přibližně  $5\tau$ , tedy v okamžiku, kdy je kondenzátor nabity přibližně na 99 % své kapacity a je již schopen dokončit záznam dat v případě ztráty zálohovaného napájení. Výsledná hodnota  $5\tau$  činí 16.5 s, což odpovídá následujícímu výpočtu:

$$5\tau = 5 \cdot R \cdot C = 5 \cdot 10 \Omega \cdot 0.33 \text{ F} = 16.5 \text{ s}$$

Jakmile je záznamník připraven na bezpečné odpojení napájení a k odpojení skutečně dojde, je pokles napětí pod stanovenou mez detekován analogovým komparátorem. V takovém případě je vyvoláno přerušení s nejvyšší prioritou. Během obslužné rutiny přerušení jsou uložena všechna data, která jsou dočasně uložená ve volatilní paměti RAM, do souboru uloženém na SD kartě prostřednictvím funkce **CONSOLELOG\_PowerLossFlush()**, která zajišťuje také korektní uzavření souboru. Následně se systém dostává do stavu, kdy je připraven na nedostatečné napětí, při kterém již není možný provoz záznamníku bez ohrožení integrity dat.

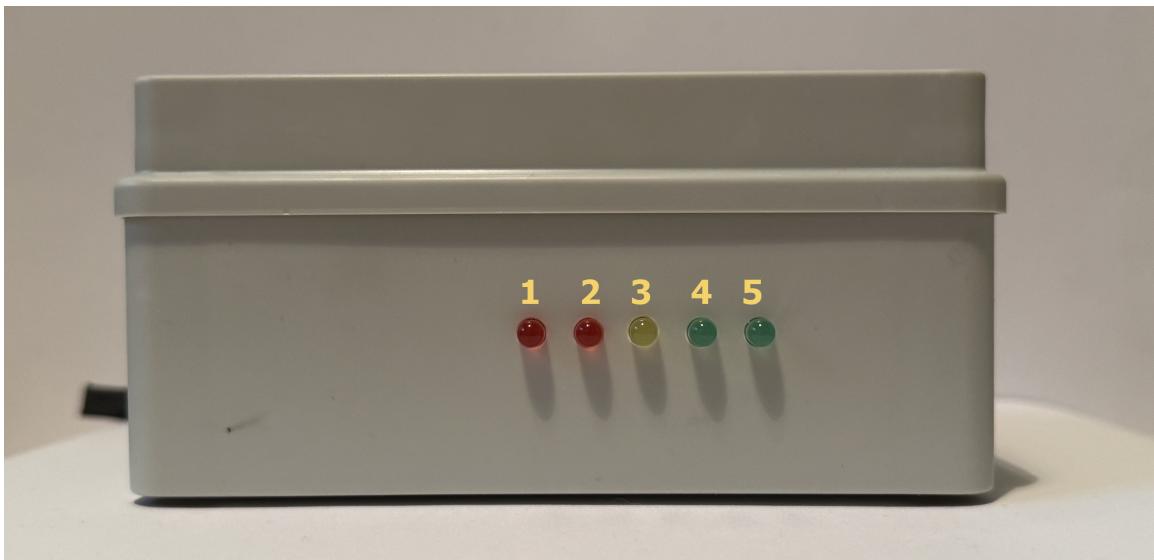
Na následujícím obrázku 5.2 je možné vidět reakci komparátoru v případě odpojení napájejícího napětí. Horní průběh představuje výstup komparátoru, který se při detekci poklesu napětí změní z logické úrovně „1“ na „0“. Spodní průběh pak zachycuje napětí na sledované 5 V napájecí větvi.



Obrázek 5.2: Detekce ztráty napájení, zachycena pomocí logického analyzátoru Saleae Logic Pro 16

## 5.4 Signalizace stavu systému

Digitální záznamník využívá sadu pěti LED diod, které slouží k signalizaci stavu digitálního záznamníku uživateli. Na následujícím obrázku 5.3 je znázorněno umístění jednotlivých LED diod na těle zařízení. Diody jsou číselně označeny 1 až 5, aby bylo možné v následujícím textu i tabulce 5.2 jednoznačně popsat jejich význam a způsob indikace.



Obrázek 5.3: Signalizační diody digitálního záznamníku

Dioda 5 signalizuje, zda je možné digitální záznamník bezpečně odpojit od napájejícího napětí a rozsvítí se za dobu  $5\tau$  (16.5 s od připojení záznamníku k napájení), kdy je zajištěno dostatečné nabité kondenzátoru pro zálohované napájení. Dioda 4 signalizuje blikáním stav, kdy digitální záznamník zaznamenává a ukládá data. Pokud záznam skončí, tedy například monitorované zařízení přestane odesílat data, nebo je digitální záznamník odpojen od monitorovaného zařízení, rozsvítí se dioda 3. Pokud volné místo na paměťové kartě klesne pod

hodnotu nastavenou parametrem `free_space`, který je definován v konfiguračním souboru, rozsvítí se dioda 1.

Během užití digitálního záznamníku může nastat situace, kdy konfigurační soubor na paměťové kartě nebude správně přečten z důvodu špatného obsahu, nebo dojde k chybě během záznamu dat, v takovém případě je rozsvícená dioda 2.

Dioda	Význam	Režim indikace
1	Volná paměť SD karty je pod hranicí definovanou parametrem <code>free_space</code>	Svítí trvale
2	Chybový stav během záznamu	Svítí trvale
3	Ukončení záznamu dat <sup>2</sup>	Svítí trvale
4	Aktivní záznam dat	Bliká při záznamu dat
5	Bezpečné odpojení	Svítí trvale

Tabulka 5.2: Přehled funkce jednotlivých signalizačních LED diod

# Kapitola 6

## Testování systému

### 6.1 Funkcionální testování systému

Funkcionalita digitálního záznamníku byla prověřena pomocí skriptu využívajícího objektově orientovaný přístup napsaném v jazyce Python. Tento skript simuluje monitorované zařízení tak, že načítá testovací textové soubory a jejich obsah postupně odesílá ve formě sériových dat přes COM port do digitálního záznamníku. Testovací skript se nachází v adresáři `tests/` pod názvem `serial_test.py` a pro jeho korektní funkcionality je potřeba správně nastavit hodnoty definující COM port přes který budou data posílána, sadu testovacích souborů a přenosovou rychlost.

Testování s pomocí skriptu bylo provedeno s různými přenosovými rychlostmi, přičemž testování bylo primárně zaměřeno na přenosové rychlosti odpovídající skutečným scénářům nasazení. Po ukončení testu byly zaznamenané výstupy porovnány s původními testovacími soubory. Výsledky porovnání potvrdily, že ve všech testovaných případech proběhl přenos korektně a nedošlo k žádné ztrátě ani poškození dat.

Digitální záznamník byl také úspěšně ověřen pomocí skriptu `stress_test.py` obsahující zátežový test. Při tomto testu byla data do digitálního záznamníku odesílána s rychlostí 921,600 baudů. Data byla nejprve shromažďována do cirkulárního bufferu o velikosti 4 KiB, z něhož se následně po blocích o velikosti 4 KiB ukládala na SD kartu. Výsledné záznamové soubory mely velikost 1 MiB.

Součástí funkčního testování bylo také ověření chování digitálního záznamníku v hraničních či nestandardních situacích, které mohou nastat při reálném nasazení digitálního záznamníku. Testy byly zaměřeny na případy, jako je absence nebo nekompletní obsah konfiguračního souboru, použití nové SD karty bez vytvořeného souborového systému, nebo ověření funkčnosti s různě velkými záznamovými soubory. Výsledky testování jsou uvedeny v následující tabulce 6.1.

Tabulka 6.1: Výsledky testování hraničních stavů digitálního záznamníku

Popis testu	Očekávaný výsledek	Splněno
Absence konfiguračního souboru	Signalizace chyby pomocí LED diody 2 a spuštění záznamu se základním nastavením	Ano
Chybějící hodnota <code>baudrate</code> v konfiguraci	Spuštění záznamu se základní hodnotou <code>baudrate</code>	Ano

Popis testu	Očekávaný výsledek	Splněno
Záznam na novou nezformátovanou SD kartu	SD karta je zformátována, je inicializován souborový systém a zařízení následně zaznamenává příchozí data	Ano
Záznam s velikostí souboru 512 B	Vytvoření souboru a správné ukončení po dosažení limitu	Ano
Záznam s velikostí souboru 1 MB	Správné naplnění a ukončení souboru bez ztráty dat	Ano
Volná kapacita SD karty pod hodnotou definovanou parametrem <b>free_space</b>	Signalizace pomocí LED diody 1	Ano
Přenos zaznamenaných dat z digitálního záznamníku	Přeneseny záznamy do hostitelského zařízení z digitálního záznamníku	Ano

Digitální záznamník byl dále testován i s reálným monitorovaným zařízením po dobu delší než 168 hodin provozu. Ani během tohoto dlouhodobého testu nebyly zjištěny žádné projevy nesprávné funkce nebo ztráty dat, čímž byla taktéž potvrzena stabilita a spolehlivost zařízení.

## 6.2 Kontrola bezpečnosti kódu

Kromě funkcionálního testování byla provedena statická analýza zdrojového kódu pomocí pravidel specifikace MISRA (Motor Industry Software Reliability Association), respektive MISRA C:2012. MISRA C:2012 je standard pro vývoj softwaru v programovacím jazyku ISO C, který má za cíl zvýšení bezpečnosti kódu, přenositelnosti a spolehlivosti v kontextu vestavěných systémů.

K tomuto účelu byl využit nástroj PC-lint Plus ve verzi 2.2 od společnosti Vector s 14denní licencí zdarma. Konfigurační soubory `lint_config.lts` a `sources.lts` spolu s výstupními reporty nástroje jsou uloženy ve složce `test/static-analysis/`.

Pozornost byla věnována primárně pravidlům spadajícím do kategorie Required a Mandatory.<sup>1</sup> Všechna nalezená porušení pravidel patřící do kategorií Required a Mandatory byla opravena, případně obhájena pomocí komentářů ve zdrojovém kódu, kde je uveden odkaz na příslušné pravidlo i odůvodnění výjimky (viz. ukázka 6.1). Obhájená porušení se týkají zejména kvůli architektuře MCUXpresso SDK, FreeRTOSu, či z důvodu, že PC-lint Plus nedokázal například najít souvislosti mezi zdrojovými a hlavičkovými soubory.

---

<sup>1</sup>Standard MISRA C rozlišuje kategorie Advisory, Mandatory a Required. Pravidla ze skupiny Advisory nebyly brány v potaz, porušení pravidel této kategorie nijak nenarušují fukcionalitu firmwaru.

```

/**
 * MISRA Deviation: Rule 10.3 [Required]
 * Suppress: Conversion From enum To unsigned32.
 * Justification: The Enum Value is Part of The NXP SDK
 * and is Intentionally Used in Context as a Bitmask Flag
 * For Hardware Status Registers.
 */

/*lint -e9034 */
CTIMER_ClearStatusFlags(CTIMER4, kCTIMER_Match0Flag);
/*lint +e9034 */

```

Výpis 6.1: Ukázka odůvodněného porušení pravidla MISRA

### 6.3 Limitace systému

Během testování nebyly nalezeny žádné skutečnosti limitující provoz digitálního záznamníku.

### 6.4 Možná vylepšení a úpravy digitálního záznamníku

Navržený digitální záznamník je koncipován jako modulární a dále rozšířitelný systém, který lze v budoucnu snadno doplnit o nové funkce či upravit pro záznam jiných typů dat podle potřeb konkrétní aplikace.

Jednou z připravených, avšak dosud nerealizovaných funkcionalit je synchronizace reálného času pomocí GPS modulu. Hardwarové rozhraní je již implementováno a připojeno, zbývá pouze doplnit softwarovou část, která bude pravidelně aktualizovat čas v RTC obvodu na základě časových údajů získaných z GPS modulu.

Z hlediska možných úprav hardwarové části digitálního záznamníku by za zvážení stálo nahrazení stávající Schottky diody v obvodu zálohovaného napájení tzv. ideální diodou na expanzní desce, která by snížila napěťový úbytek na 5 V větvi a díky níž by se zlepšila účinnost zálohovaného napájení.

Dalším potenciálním rozšířením je zavedení mechanismů pro zabezpečení zaznamenaných dat uložených na SD kartě, zahrnujících šifrování, ověřování integrity dat a řízení přístupu. Jednou z možností, jak zmíněné zabezpečení realizovat, je použití speciálních microSD karet s integrovaným kryptografickým prvkem. Tyto karty umožňují přímo na úrovni paměťového média realizovat funkce jako automatické šifrování uložených dat, digitální podepisování a ověřování integrity, nebo ochranu přístupu na základě identity zařízení a to přímo na úrovni paměťového média. Pro využití funkcí těchto karet by bylo možné rozšířit firmware digitálního záznamníku o rozhraní komunikující s bezpečnostním prvkem paměťové karty, počemž by bylo možné zmíněné microSD karty využít.

Digitální záznamník by rovněž mohl být vybaven volitelnou možností odesílání dat na vzdálené úložiště místo ukládání na SD kartu. Pomocí této možnosti by mohl být digitální záznamník ještě univerzálněji využitelný, neboť data by mohla být na vzdáleném úložišti dále zpracovávána, automaticky analyzována a dlouhodobě archivována.

# Kapitola 7

## Závěr

Cílem bakalářské práce bylo navržení a realizace digitálního záznamníku pro účely společnosti NXP Semiconductors, a tohoto cíle bylo dosaženo.

Práce se věnuje historickému pozadí záznamu dat, počátkům záznamu dat v počátcích elektrotechniky a digitálnímu záznamu dat včetně záznamu dat na osobním počítači a na mikrokontrolérech. Rozvedeny jsou také klíčové koncepty nevyjímaje vkládání synchronizačních značek, které jsou mnohdy využívány digitálními záznamníky a způsoby ukládání dat, přičemž je rovněž zmíněn i způsob využívající FeRAM paměť pro prevenci ztráty dat při výpadku napájení.<sup>1</sup>

Navazující část práce je zaměřena na analýzu existujících řešení dostupných na trhu a identifikaci jejich klíčových vlastností. Na tuto analýzu navazuje podrobná rešerše jednotlivých komponent, které byly brány v úvahu při návrhu architektury digitálního záznamníku a samotný návrh této architektury.

Architektura digitálního záznamníku neobsahuje klasické ovládací prvky, které byly zmíněny v zadání, jelikož ovládání digitálního záznamníku probíhá na základě připojení/odpojení aplikačního USB s ověřením pomocí USB OTG. Díky tomuto přístupu je zjednodušena obsluha zařízení, neboť digitální záznamník bude často používán nezkušenou obsluhou, která s tímto zařízením přijde do kontaktu poprvé.

Součástí práce je návrh a realizace hardwarové expanzní desky, která rozšiřuje funkcionality vývojové desky, a taktéž je popsán i návrh a implementace firmwaru. Následně je rozbrán způsob, jakým probíhalo testování digitálního záznamníku, při kterém byla ověřena správná funkcionalita digitálního záznamníku a dále byla ověřena bezpečnost a spolehlivost kódu firmwaru vůči pravidlům standardu MISRA C.

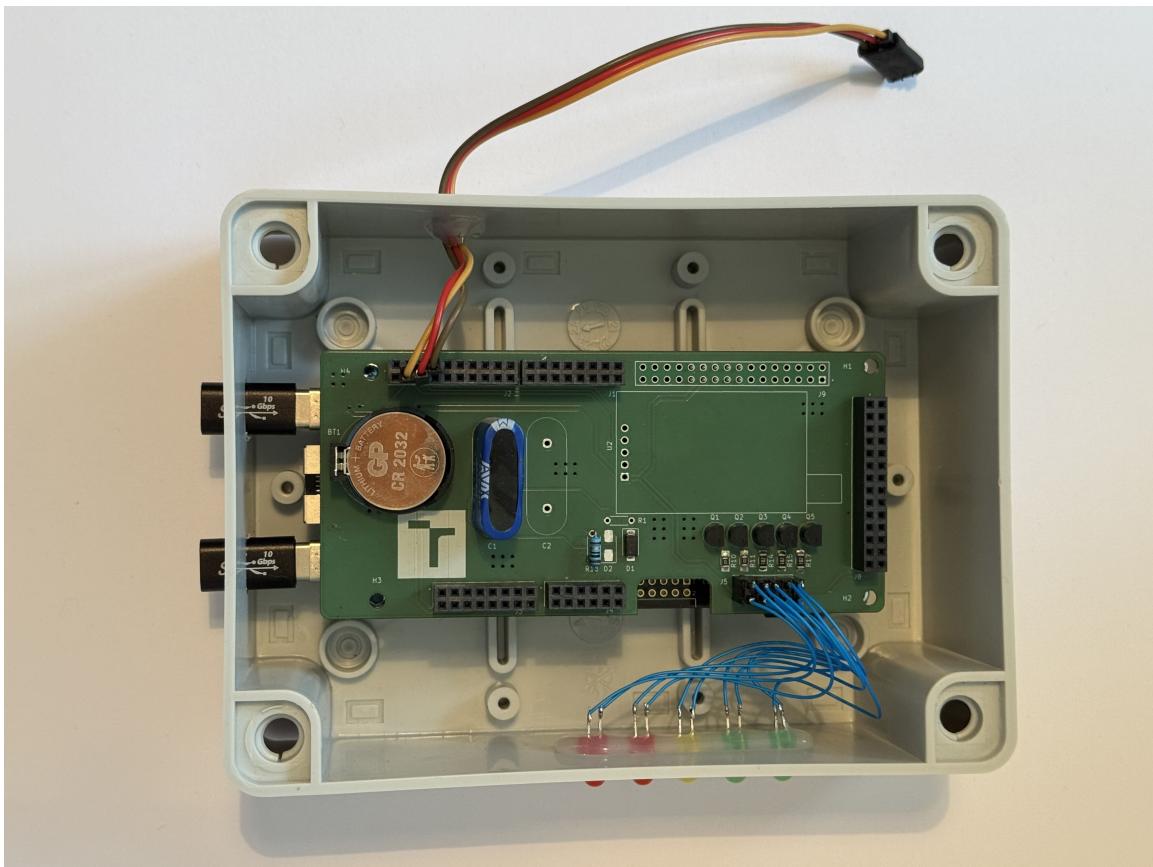
Výsledkem práce je zařízení, které umožňuje záznam sériových dat z rozhraní UART včetně vkládání časových značek do záznamů, systematické ukládání záznamů a pozdější přístup k těmto datům bez nutnosti vyjmout paměťového média. Záznamník je zároveň schopen bezpečně reagovat na výpadky napájecího napětí a zabezpečit tak případnou ztrátu či poškození zaznamenaných dat a souborového systému.

Digitální záznamník také obsahuje přípravu na možná rozšíření v připojení GPS modulu pro synchronizaci času. Firmware digitálního záznamníku je díky své modularitě možné nadále rozšiřovat, a nebo upravit pro záznam jiného typu dat.

Výhodou vyvinutého záznamníku jsou i nižší náklady na realizaci vůči konkurenci, která byla zmíněná v kapitole 3.1. Celková cena zařízení činí přibližně 1473 Kč. Například Air-

<sup>1</sup>V průběhu textu je dále věnovaná pozornost dalším dvěma způsobům prevence ztráty dat, konkrétně využíváním žurnálovacího souborového systému (viz. kapitola 3.3.2) a implementaci detekce výpadku napájení (power-fail detection) s následným bezpečným vypnutím systému (viz. kapitola 4.1.1 a kapitola 5.3).

*Drive Serial Logger Pro* stojí přibližně dvojnásobnou cenu a zařízení *Anticyclone Anti-Log Pro* v nejlevnější konfiguraci přesahuje čtyřnásobek.



Obrázek 7.1: Pohled na vnitřní uspořádání digitálního záznamníku vyvinutého pro společnost NXP Semiconductors

# Literatura

- [1] *AirDrive Serial Logger* online. Keelog. Dostupné z: <https://www.keelog.com/serial-logger/>. [cit. 2025-03-01].
- [2] *AntiLog and AntiLogPro – High performance RS232 data loggers* online. Anticyclone Systems. Dostupné z: [https://www.anticyclone-systems.co.uk/product\\_antilog.php](https://www.anticyclone-systems.co.uk/product_antilog.php). [cit. 2025-03-01].
- [3] *The Basics of GPS* online. Sparkfun. Dostupné z: <https://learn.sparkfun.com/tutorials/gps-basics/all>. [cit. 2025-03-16].
- [4] *Choosing the Right Real Time Clock Chip or Module* online. Jameco. Dostupné z: <https://www.jameco.com/Jameco/Blog/choosing-the-right-real-time-clock-chip-or-module.html>. [cit. 2025-03-16].
- [5] *Creating a New FreeRTOS Port* online. FreeRTOS. Dostupné z: <https://freertos.org/Documentation/02-Kernel/03-Supported-devices/01-FreeRTOS-porting-guide>. [cit. 2025-03-22].
- [6] *Device Monitoring Studio* online. HHD Software. Dostupné z: <https://hhdsoftware.com/device-monitoring-studio>. [cit. 2025-02-17].
- [7] *DS3231* online. Analog Devices. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf>. [cit. 2025-03-23].
- [8] *Getting Started with FRDM-MCXN947* online. NXP Semiconductors. Dostupné z: <https://www.nxp.com/document/guide/getting-started-with-frdm-mcxn947-GS-FRDM-MCXNXX>. [cit. 2025-03-07].
- [9] *Getting Started with LabVIEW* online. National Institute. Dostupné z: <https://www.sjsu.edu/people/burford.furman/docs/me120/GettingStarted.pdf>. [cit. 2025-03-11].
- [10] *GPS JAKO „DUKAZ“ PLATNOSTI TR* online. Time Theory. Dostupné z: <https://time-theory.info/gps-jako-dukaz-platnosti-tr/>. [cit. 2025-03-16].
- [11] *Interfacing RTC with Microcontroller* online. Embed Journal. Dostupné z: <https://embedjournal.com/interfacing-rtc-with-microcontroller/>. [cit. 2025-03-16].
- [12] *KL05 Sub-Family Reference Manual* online. NXP Semiconductors. Dostupné z: <https://www.nxp.com/products/KL0x>. [cit. 2025-03-07].

- [13] *MAX-7 / NEO-7 u-blox 7 GNSS modules Hardware Integration Manual* online. U-blox. Dostupné z: [https://content.u-blox.com/sites/default/files/products/documents/MAX7-NEO7\\_HardwareIntegrationManual\\_%28UBX-13003704%29.pdf](https://content.u-blox.com/sites/default/files/products/documents/MAX7-NEO7_HardwareIntegrationManual_%28UBX-13003704%29.pdf). [cit. 2025-04-23].
- [14] *MCUXpresso SDK API Reference Manual Rev. 0: The design of the little filesystem* online. NXP Semiconductors. Dostupné z: [https://mcuxpresso.nxp.com/api\\_doc/dev/1620/a00014.html](https://mcuxpresso.nxp.com/api_doc/dev/1620/a00014.html). [cit. 2025-03-11].
- [15] *MCX Nx4x Reference Manual* online. NXP Semiconductors. Dostupné z: <https://www.nxp.com/design/design-center/development-boards-and-designs/FRDM-MCXN947>. [cit. 2025-03-07].
- [16] *Media Transfer Protocol* online. USB. Dostupné z: <https://www.usb.org/document-library/media-transfer-protocol-v11-spec-and-mtp-v11-adopters-agreement>. [cit. 2025-03-11].
- [17] *Raspberry Pi Pico 2 W* online. Raspberry Pi Documentation. Dostupné z: <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html>. [cit. 2025-03-06].
- [18] *Universal Serial Bus Mass Storage Class: UFI Command Specification* online. USB. Dostupné z: <https://www.usb.org/document-library/mass-storage-ufi-command-specification-10>. [cit. 2025-03-11].
- [19] *USB Device MSC Class* online. Silicon Labs. Dostupné z: <https://docs.silabs.com/protocol-usb/1.3.0/protocol-usb-msc-scsi/#usb-device-msc-class>. [cit. 2025-03-11].
- [20] *Chapter 13 — File and Database Systems* online. Virginia Tech, 2004. Dostupné z: [https://courses.cs.vt.edu/~cs3204/spring2004/Notes/OS3e\\_13.pdf](https://courses.cs.vt.edu/~cs3204/spring2004/Notes/OS3e_13.pdf). [cit. 2025-03-08].
- [21] *Conservation Notes: Get to know a hygrothermograph* online. Art Gallery of Ontario, srpen 2012. Dostupné z: <https://ago.ca/agoinsider/conservation-notes-get-know-hygrothermograph>. [cit. 2025-02-14].
- [22] *Understanding MCU sleep modes and energy savings* online. embedded, březen 2012. Dostupné z: <https://www.embedded.com/understanding-mcu-sleep-modes-and-energy-savings/>. [cit. 2025-02-18].
- [23] *What is Structured and Unstructured Data* online. NXP Semiconductors, červenec 2020. Dostupné z: <https://www.weka.io/learn/glossary/file-storage/what-is-structured-and-unstructured-data/>. [cit. 2025-03-08].
- [24] *How Does NTP Work?* online. Kevin Sookocheff, listopad 2021. Dostupné z: <https://sookocheff.com/post/time/how-does-ntp-work/>. [cit. 2025-03-23].
- [25] *Development on Bare Metal vs. RTOS* online. SYSGO, duben 2022. Dostupné z: <https://learn.sparkfun.com/tutorials/gps-basics/all>. [cit. 2025-03-19].

- [26] *The Origin and History of Chart Recorders* online. Recorder Charts and Pens, 2023. Dostupné z: <https://recorderchartsandpens.com/blog/the-origin-and-history-of-chart-recorders/?srsltid=AfmB0ooMqIXNP8LY1II8itclRzgEKo1N0c6G0eVfD3ckKHGCo0dteTkGe>. [cit. 2025-02-14].
- [27] *Pros and Cons of Analog Chart Recorders vs. Digital* online. Recorder Charts and Pens, 2023. Dostupné z: <https://recorderchartsandpens.com/blog/pros-and-cons-of-analog-chart-recorders-vs-digital/?srsltid=AfmB0orCtQ41tB81JTSdeeAj143DQ9SWK5zJ-F3Wc8TChdN2RJS7iWsb>. [cit. 2025-04-15].
- [28] *Why We Moved from FreeRTOS to Zephyr RTOS* online. Zephyr, březen 2023. Dostupné z: <https://www.zephyrproject.org/why-we-moved-from-freertos-to-zephyr-rtos/>. [cit. 2025-03-22].
- [29] *FAT Filesystem* online. ELM, 2024. Dostupné z: [http://elm-chan.org/docs/fat\\_e.html](http://elm-chan.org/docs/fat_e.html). [cit. 2025-03-09].
- [30] *FatFs Module Application Note* online. ELM, 2024. Dostupné z: <http://elm-chan.org/fsw/ff/doc/appnote.html>. [cit. 2025-03-09].
- [31] *MQTT vs CoAP: Comparing Protocols for IoT Connectivity* online. EMQ, duben 2024. Dostupné z: <https://www.emqx.com/en/blog/mqtt-vs-coap>. [cit. 2025-02-28].
- [32] *Procps* online. GitLab, 2024. Dostupné z: [https://gitlab.com/procps-ng/procps/-/blob/master/README.md?ref\\_type=heads](https://gitlab.com/procps-ng/procps/-/blob/master/README.md?ref_type=heads). [cit. 2025-02-19].
- [33] *RTOS Fundamentals* online. FreeRTOS, březen 2025. Dostupné z: <https://www.freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals>. [cit. 2025-03-16].
- [34] *Safety Critical Real-Time OS* online. FreeRTOS, duben 2025. Dostupné z: <https://www.freertos.org/Partners/Software/SafeRTOS>. [cit. 2025-03-22].
- [35] *Security overview* online. FreeRTOS, březen 2025. Dostupné z: <https://www.freertos.org/Security/01-Security-overview>. [cit. 2025-03-22].
- [36] ASADUZZAMAN, A.; CHIDELLA, K. K. a SIBAI, F. N. A smart data logger for enhancing data communication in Wi-Fi based mobile systems. In: *SoutheastCon 2015*. 2015, s. 1–6. Dostupné z: <https://ieeexplore.ieee.org/document/7132925>.
- [37] BARRY, R. *Mastering the FreeRTOS Real Time Kernel*. Real Time Engineers Ltd., únor 2024. Dostupné z: [https://freertos.org/Documentation/02-Kernel/07-Books-and-manual/01-RTOS\\_book](https://freertos.org/Documentation/02-Kernel/07-Books-and-manual/01-RTOS_book).
- [38] BRISTOL, W. H. *Pressure Indicator and Recorder*. 1888. Dostupné z: <https://patents.google.com/patent/US389635A/en>. US Patent 389635, Bristol Manufacturing Company, 18. září 1888.

- [39] CARR, J. *Ice Cream Sandwich* online. Android Developers. Dostupné z: [https://developer.android.com/about/versions/android-4.0-highlights?utm\\_source=chatgpt.com#DeveloperApis](https://developer.android.com/about/versions/android-4.0-highlights?utm_source=chatgpt.com#DeveloperApis). [cit. 2025-04-01].
- [40] CHANDRASEKARAN, S. *Implementing Circular Buffer in C* online. Embed Journal. Dostupné z: <https://embedjournal.com/implementing-circular-buffer-embedded-c/>. [cit. 2025-02-18].
- [41] CLINTON, D. *Linux in Action*. 1. vyd. Manning, 2018. 229–251 s. ISBN 9781617294938.
- [42] GRZESIK, P. a MROZEK, D. Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks. In: KRZHIZHANOVSKAYA, V. V.; ZÁVODSZKY, G.; LEES, M. H.; DONGARRA, J. J.; SLOOT, P. M. A. et al., ed. *Computational Science – ICCS 2020*. Cham: Springer International Publishing, 2020, s. 371–383. ISBN 978-3-030-50426-7. Dostupné z: [https://link.springer.com/chapter/10.1007/978-3-030-50426-7\\_28](https://link.springer.com/chapter/10.1007/978-3-030-50426-7_28).
- [43] HARATIPOUR, N.; CHANG, S.-C.; SHIVARAMAN, S.; NEUMANN, C.; LIAO, Y.-C. et al. Hafnia-Based FeRAM: A Path Toward Ultra-High Density for Next-Generation High-Speed Embedded Memory. In: *2022 International Electron Devices Meeting (IEDM)*. 2022, s. 6.7.1–6.7.4.
- [44] HUNG, P. a VUONG, V. D. Analog Signal and Digital Signal Processing in Telecommunication System. *Journal La Multiapp* online, Leden 2021, sv. 1, s. 6–11. Dostupné z: <https://doi.org/10.37899/journallamultiapp.v1i6.277>.
- [45] JORDAN, L. *What Is FAT32 File System – A Detailed Guide* online. Recoverit, 2023. Dostupné z: <https://recoverit.wondershare.com/file-system/fat32-file-system.html>. [cit. 2025-03-09].
- [46] KOLHE, A. *What is RingBuffer?* online. Medium. Dostupné z: <https://medium.com/@kolheankita15/what-is-ringbuffer-7b6b808f33e0>. [cit. 2025-02-18].
- [47] KUMAR, D. A Study about Non-Volatile Memories. *Preprints*. Preprints, July 2016. Dostupné z: <https://doi.org/10.20944/preprints201607.0093.v1>.
- [48] KUMAR, J. a RAMESH, P. R. Low Cost Energy Efficient Smart Security System with Information Stamping for IoT Networks. In: *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*. 2018, s. 1–5. Dostupné z: <https://ieeexplore.ieee.org/document/8519875>.
- [49] KUMARI, M. a SINGH, M. P. Industry Monitoring with Data Logger in Google Sheet Using Raspberry Pi. In: SINGH, J. P.; SINGH, M. P.; SINGH, A. K.; MUKHOPADHYAY, S.; MANDAL, J. K. et al., ed. *Computational Intelligence in Communications and Business Analytics*. Cham: Springer Nature Switzerland, 2025, s. 125–133. ISBN 978-3-031-81336-8. Dostupné z: [https://link.springer.com/chapter/10.1007/978-3-031-81336-8\\_10](https://link.springer.com/chapter/10.1007/978-3-031-81336-8_10).

- [50] MAHENDRA, O.; SYAMSI, D.; RAMDAN, A. a ASTRID, M. Design and implementation of data storage system using USB flash drive in a microcontroller based data logger. In: *2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*. 2015, s. 58–62.
- [51] MAHZAN, N.; OMAR, A.; MOHAMMAD NOOR, S. a RODZI, Z. Design of data logger with multiple SD cards. Conference: *2013 IEEE Conference on Clean Energy and Technology (CEAT)*, Listopad 2013, s. 175–180. Dostupné z: [https://www.researchgate.net/publication/271547689\\_Design\\_of\\_data\\_logger\\_with\\_multiple\\_SD\\_cards](https://www.researchgate.net/publication/271547689_Design_of_data_logger_with_multiple_SD_cards).
- [52] MATESKI; STOJAN; ANASTASOVSKI a ZORAN. Digital sound recorder with ARM microcontroller and SD card. *Telecommunications Forum (TELFOR)*, 2012, s. 1741–1743. Dostupné z: <https://ieeexplore.ieee.org/document/6419564>.
- [53] MATOUŠEK, P. *Síťové aplikace a jejich architektura*. 1. vyd. VUTIUM, 2014. ISBN 978-80-214-3766-1.
- [54] MERABTANE, B. a BENABADJI, N. Design and Implementation of a Multi Connectivity Datalogger using SD Card Storage for Photovoltaic System Monitoring. *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, 2023, sv. 1, s. 1–5. Dostupné z: <https://ieeexplore.ieee.org/document/10419399>.
- [55] MIR, I.; MCEWAN, A. a PERRINS, N. *A High Performance Reconfigurable Architecture for Flash File Systems*. Dostupné z: [https://www.researchgate.net/publication/268370228\\_A\\_High\\_Performance\\_Reconfigurable\\_Architecture\\_for\\_Flash\\_File\\_Systems](https://www.researchgate.net/publication/268370228_A_High_Performance_Reconfigurable_Architecture_for_Flash_File_Systems).
- [56] MRABET, H.; BELGUTH, S.; ALHOMOUD, A. a JEMAI, A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors*, 2020, sv. 20, č. 13. ISSN 1424-8220. Dostupné z: <https://www.mdpi.com/1424-8220/20/13/3625>.
- [57] PERNÝ, J. *Zařízení pro monitorování sériové komunikace* online. Brno, 2008. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Dostupné z: <http://hdl.handle.net/11012/25413>. Vedoucí práce: Jaromír Kolouch.
- [58] PETERSON, D. A. *Program pacer* online. 1980. Dostupné z: <https://patents.google.com/patent/US4333084>. [cit. 2025-02-14].
- [59] PETRUNGARO, D. *Data Structure – Circular buffer* online. damianopetrungaro. Dostupné z: <https://www.damianopetrungaro.com/posts/data-structure-circular-buffer/>. [cit. 2025-02-18].
- [60] POROBIC, V. B.; MARCETIC, D. P. a KATIC, V. A. Data logging in the electrical drives. In: *MELECON 2008 – The 14th IEEE Mediterranean Electrotechnical Conference*. 2008, s. 490–495. Dostupné z: <https://ieeexplore.ieee.org/document/4618483>.

- [61] RANDELL, B. The History of Digital Computers. “*l’Histoire Générale des Techniques* online. edice 1. Springer Verlag, 1973, Únor 1973, sv. 3, s. 57–69. Dostupné z: <https://doi.org/https://doi.org/10.1007/978-3-642-96145-8>. [cit. 2025-02-16].
- [62] REDON, M. Strategies for Choosing the Appropriate Microcontroller when Developing Ultra Low Power Systems. *Analog Dialogue*. Analog Devices, Prosinec 2017, sv. 51, č. 4. Dostupné z: <https://www.analog.com/en/resources/analog-dialogue/articles/strategies-for-choosing-the-appropriate-microcontroller-when-developing-ultra-low-power-systems.html>.
- [63] RINALDI, J. *Local Vs. Cloud: Are You Going About Data Logging All Wrong?* online. Real Time Automation, říjen 2020. Dostupné z: <https://www.rtautomation.com/local-vs-cloud-are-you-going-about-data-logging-all-wrong/?srsltid=AfmB0oqrdjlqC47hR-VZagW2sKvRgDJkqru8xxEqag66zC36aXIztNvj>. [cit. 2025-02-20].
- [64] SHEEPARAMATTI, R.; SHEEPARAMATTI, B.; BHARAMAGOUDAR, M. a AMBALI, N. Simulink Model for Double Buffering. In: *IECON 2006 – 32nd Annual Conference on IEEE Industrial Electronics*. 2006, s. 4593–4597.
- [65] SHWETA. *Optimizing IoT Data Storage: Experience with PostgreSQL and TimescaleDB* online. Medium. Dostupné z: <https://medium.com/@paigude.shweta/optimizing-iot-data-storage-experience-with-postgresql-and-timescaledb-e4dc3dcc7ae7>. [cit. 2025-02-21].
- [66] SVEC, C. *The Architecture of Open Source Applications*. 2. vyd. Lulu.com, 2013. ISBN 978-1-105-57181-7. Dostupné z: <https://aosabook.org/en/v2/freertos.html>.
- [67] TSAI, Y.-L.; HSIEH, J. wei a KUO, T.-W. Configurable NAND Flash Translation Layer. In: *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. červenec 2006, sv. 2006, s. 8 pp. ISBN 0-7695-2553-9.
- [68] UMER, M. *Architecture and Design of the Linux Storage Stack*. Packt Publishing, 2023. 57 s. ISBN 9781837639960.
- [69] UNDERWOOD, S. *Implementing Temperature Compensated RTC* online. Texas Instruments. Dostupné z: <https://www.ti.com/lit/ml/slap107/slap107.pdf>. [cit. 2025-03-23].
- [70] WAEHNER, K. *Internet of Things (IoT) and Event Streaming at Scale with Apache Kafka and MQTT* online. Confluent, říjen 2019. Dostupné z: <https://www.confluent.io/blog/iot-with-kafka-connect-mqtt-and-rest-proxy/>. [cit. 2025-02-21].

## Příloha A

# Obsah přiloženého paměťového média

- **application/** – adresář obsahující firmware digitálního záznamníku, včetně vygenerované HTML a Doxygen dokumentace
  - **src/** – adresář obsahující zdrojové soubory firmwaru digitálního záznamníku
  - **include/** – adresář obsahující hlavičkové soubory firmwaru digitálního záznamníku
  - **doc/** – adresář obsahující LaTeX a HTML Doxygen dokumentaci firmwaru digitálního záznamníku
  - součástí projektu jsou nutné externí knihovny a moduly, které jsou potřeba k sestavení projektu, tyto celky byly převzaty z MCUXpresso SDK
- **hardware/** – adresář s projektem expanzní desky: schématy zapojení, návrhem PCB, seznamem součástek (BOM) a schématem v PDF
  - **extension\_shield/** – adresář obsahující KiCAD projekt expanzní desky.
    - \* **extension\_shield.kicad\_pro** – hlavní projektový soubor KiCadu pro návrh expanzní desky
  - **BOM.csv** – seznam komponent expanzní desky digitálního záznamníku
  - **schematik.pdf** – schématik realizované expanzní desky
- **thesis/** – adresář obsahující text technické zprávy k bakalářské práci, včetně obrázků a diagramů
  - **xdolak09-bthesis.pdf** – finální text technické zprávy k bakalářské práci
  - **xdolak09-bthesis-src.zip** – archiv se zdrojovými soubory zprávy, včetně všech použitých obrázků
- **tests/** – adresář obsahující testovací skripty, testovací soubory a výstupy ze statické analýzy kódu pomocí nástroje PC-lint Plus
- **power\_consumption/** – data z analýzy spotřeby energie, změřená pomocí Power Profiler Kit II od společnosti Nordic Semiconductors, součástí složky je také soubor **README.md** s popisem jednotlivých měření a souborů

- **powerloss\_detection/** – data z analýzy chování digitálního záznamníku při ztrátě napájecího napětí zaznamenaná pomocí Saleae Logic 16 Pro, složka rovněž obsahuje soubor **README.md** s vysvětlením struktury a významu jednotlivých souborů
- **misc/** – složka obsahující nezařazené soubory, například obrázky k **README**
- **manual.pdf** – manuál k použití digitálního záznamníku
- **video-ukazka.mp4** – video ukázka použití digitálního záznamníku ve formátu MP4
- **video-ukazka.mov** – video ukázka použití digitálního záznamníku ve formátu MOV
- **README.md** – úvodní soubor s popisem projektu, instalací a způsobem použití

Celý obsah přiloženého paměťového média je dostupný prostřednictvím cloudového úložiště Nextcloud na adrese: <https://nextcloud.fit.vutbr.cz/s/xgZG6ZdZeccKQMC>.

## Příloha B

# Projekt digitálního záznamníku

Digitální záznamník je možné dále rozšířit nebo upravit dle konkrétních požadavků. Celý projekt, včetně implementace firmwaru, návrhu expanzní desky i veškeré dokumentace, je veřejně dostupný na platformě GitHub.

- Odkaz na GitHub repozitář je <https://github.com/Doly02/nxp-mcxn947-datalogger>

## Příloha C

### Použité nástroje

- **MCUXpresso IDE** ve verzi 11.10.0, jež bylo využito jako hlavní vývojové prostředí pro implementaci firmwaru digitálního záznamníku.
- **NXP SDK FRDM-MCXN947** ve verzi 2.16.000 včetně kterého jsou ovladače pro desku **FRDM-MCXN947**, knihovna **FATFS**, operační systém reálného času **FreeRTOS** a moduly **Mass Storage, MMC a USB**.
- **MCU LinkServer**, který byl použitý pro nahrávání a ladění firmwaru.
- **nRF Connect for Desktop** od **Nordic Semiconductor** pro měření spotřeby digitálního záznamníku pomocí **Power Profiler Kit II**.
- **Logic** ve verzi 2.4.22 pro pozorování signálů digitálního záznamníku pomocí logického analyzátoru **Saleae Logic Pro 16**.
- **GitHub** pro verzování firmwaru digitálního záznamníku a technické zprávy.
- **KiCAD 8** pro návrh expanzní desky digitálního záznamníku.
- **PC-lint Plus** ve verzi 2.2 pro statickou analýzu zdrojového kódu.
- **Doxygen** pro generování HTML a LaTeX dokumentace zdrojového kódu.
- **Overleaf** pro úpravu zdrojových souborů technické zprávy.
- **Draw.io** pro tvorbu diagramů, které jsou součástí technické zprávy.
- **ChatGPT**, který byl využit pro návrhy vylepšení struktury a formulace vět v technické zprávě.