



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ČASOVÁ SYNCHRONIZACE MIKROKONTROLERŮ RODINY STM32

TIME SYNCHRONISATION OF STM32 MICROCONTROLLERS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Kratochvíl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Kunz, Ph.D.

BRNO 2024

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Jakub Kratochvíl

ID: 220994

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Časová synchronizace mikrokontrolerů rodiny stm32

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vyzkoušet různé způsoby časové synchronizace mikrokontrolerů s využitím pro distribuované měřicí systémy. Práci lze shrnout do těchto bodů:

1. Proveďte literární rešerši možných způsobů časové synchronizace mikrokontrolerů
2. Proveďte literární rešerši možných způsobů měření přesnosti časové synchronizace
3. Vybrané metody časové synchronizace (min 4) implementujte do stm32 a proměřte vlastnosti těchto metod synchronizace. Otestujte také vybrané rušivé vlivy.
4. Srovnajte výhody a nevýhody jednotlivých metod synchronizace.

DOPORUČENÁ LITERATURA:

TANENBAUM, Andrew S. a Maarten van STEEN. Distributed systems: principles and paradigms. Upper Saddle River, N.J.: Prentice Hall, c2002. ISBN 0-13-088893-1.

Termín zadání: 5.2.2024

Termín odevzdání: 15.5.2024

Vedoucí práce: Ing. Jan Kunz, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Tato diplomová práce se zabývá implementací metod časové synchronizace mikrokontrolerů a jejich testování. V teoretické části jsou představeny relativní a absolutní metody časové synchronizace. Následně, je představen způsob testování časové synchronizace, pomocí měření zpoždění a Allanovy odchylky. V diplomové práci jsou implementovány celkově 4 metody časové synchronizace, GPS, DCF77, RDS data a metoda využívající nadřazeného systému. Kdy jako nejlepší z nich se ukázala metoda synchronizace z nadřazeného systému. Kdy tato metoda byla schopna synchronizovat začátek sekundy na desítky ns. Jako nejlepší bezdrátová metoda se ukázala metoda s GPS, kdy její přesnost začátku sekund byla v stovkách ns.

Klíčová slova

Časová synchronizace, GPS, DCF77, RDS, STM32

Abstract

This master's thesis focuses on the implementation of methods for time synchronization of microcontrollers and their testing. The theoretical part introduces relative and absolute methods of time synchronization. Subsequently, the testing method for time synchronization is presented, using delay measurements and Allan deviation. In the thesis, a total of 4 methods of time synchronization are implemented: GPS, DCF77, RDS, and a method utilizing a superior system. Among these methods, synchronization from the superior system proved to be the most effective, achieving synchronization of the beginning of a second within tens of nanoseconds. The best wireless method has proven to be the GPS method, with its accuracy at the beginning of seconds being in the hundreds of nanoseconds.

Keywords

Time synchronization, GPS, DCF77, RDS, STM32

Bibliografická citace

KRATOCHVÍL, Jakub. Časová synchronizace mikrokontrolerů rodiny stm32 [online]. Brno, 2024 [cit. 2024-04-16]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/159973>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Jan Kunz.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Jakub Kratochvíl*

VUT ID studenta: *220994*

Typ práce: *Diplomová práce*

Akademický rok: *2023/24*

Téma závěrečné práce: *Časová synchronizace mikrokontrolerů rodiny stm32*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 15. května 2024

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jan Kunz, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 15. května 2024

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	11
ÚVOD	12
1. ČASOVÁ SYNCHRONIZACE.....	13
2. RELATIVNÍ METODY ČASOVÉ SYNCHRONIZACE	14
2.1 SYNCHRONIZACE NA ZÁKLADĚ MĚŘENÍ PROUDU PRO BLE.....	14
2.1.1 <i>Bluetooth low energy (BLE)</i>	14
2.1.2 <i>Princip časové synchronizace pro BLE</i>	14
2.1.3 <i>Měření přesnosti časové synchronizace pomocí BLE</i>	15
2.2 SYNCHRONIZACE POMOCÍ SIGNÁLU OD NADŘÍZENÉHO SYSTÉMU	16
2.2.1 <i>Přenos synchronizačního pulzu</i>	16
2.3 PTP PROTOKOL.....	17
2.3.1 <i>Princip synchronizace</i>	17
3. ABSOLUTNÍ METODY ČASOVÉ SYNCHRONIZACE.....	19
3.1 GPS.....	19
3.1.1 <i>Princip časové synchronizace pomocí GPS</i>	19
3.1.2 <i>Nepříznivé vlivy</i>	19
3.2 DCF77	20
3.2.1 <i>Časová synchronizace pomocí DCF77</i>	20
3.3 SYNCHRONIZACE POMOCÍ RDS DAT Z RADIOVÝCH STANIC	21
3.3.1 <i>Princip časové synchronizace pomocí RDS pásma</i>	21
3.4 SHRNUTÍ TEORETICKÝCH METOD.....	22
4. POUŽITÝ HARDWARE.....	24
4.1 MIKROKONTROLER NUCLEO-F303K8.....	24
4.2 MIKROKONTROLER STM32F769I-DISCO	24
4.3 ANALOG DISCOVERY 2.....	24
4.4 AGILENT 53131A	24
5. ZPŮSOBY VYHODNOCOVÁNÍ METOD	26
5.1 MĚŘENÍ ROZDÍLU V PŘIJMUTÍ SYNCHRONIZAČNÍHO SIGNÁLU	26
5.2 MĚŘENÍ STABILITY FREKVENCE VNITŘNÍHO RC OSCILÁTORU	26
5.2.1 <i>Allanův rozptyl</i>	27
6. RC OSCILÁTOR.....	29
7. SYNCHRONIZACE Z NADŘÍZENÉHO SYSTÉMU	30
7.1 ROZDÍL V PŘIJMUTÍ SYNCHRONIZAČNÍHO SIGNÁLU	30
7.1.1 <i>Nezatížený slave mikrokontroler</i>	30
7.1.2 <i>Zatížený slave mikrokontroler</i>	32
7.1.3 <i>Prodloužení synchronizačního vedení</i>	33
7.1.4 <i>Zpětná smyčka pro master mikrokontroler</i>	36
7.1.5 <i>Eliminace zpoždění na kabelu</i>	38

7.2	ZHODNOCENÍ METODY	39
8.	SYNCHRONIZACE POMOCÍ GPS.....	40
8.1	PERIODA PPS SIGNÁLU	40
8.2	POSUN SIGNÁLU MEZI PŘIJÍMAČI GPS.....	41
8.3	POSUN SIGNÁLU NA VÝSTUPECH MCU	43
8.4	ALLANOVA ODCHYLKA PŘI SYNCHRONIZACI POMOCÍ GPS.....	45
8.4.1	<i>Nastavování vnitřních hodin</i>	<i>46</i>
8.4.2	<i>PS regulátor</i>	<i>47</i>
8.4.3	<i>Měření Allanovy odchylky.....</i>	<i>47</i>
8.5	ZHODNOCENÍ METODY GPS	48
9.	SYNCHRONIZACE POMOCÍ RDS DAT Z FM RÁDÍÍ.....	49
9.1	MĚŘENÍ DOBY MEZI PO SOBĚ JDOUCÍMI PULZY	49
9.2	POSUN SIGNÁLU MEZI PŘIJÍMAČI RDS	51
9.3	POUŽITÍ GPS K MĚŘENÍ DOBY MEZI PULZY RDS	53
9.3.1	<i>Popis provedení měření</i>	<i>53</i>
9.3.2	<i>Problémy této metody.....</i>	<i>54</i>
9.3.3	<i>Výsledky měření</i>	<i>55</i>
9.4	ALLANOVA ODCHYLKA	56
9.5	ZHODNOCENÍ METODY RDS	57
10.	SYNCHRONIZACE POMOCÍ DCF77.....	58
10.1	ÚPRAVA PŘIJÍMAČŮ DCF77	58
10.2	MĚŘENÍ DOBY MEZI PO SOBĚ JDOUCÍMI PULZY	59
10.3	POSUN SIGNÁLU MEZI DVĚMA PŘIJÍMAČI DCF77	61
10.4	REGULAČNÍ OBVOD PRO SYSTÉM VYUŽÍVAJÍCÍ DCF77	64
10.5	ZHODNOCENÍ METODY DCF77	64
11.	NÁHRADNÍ MODEL.....	65
11.1	PRVNÍ VERZE NÁHRADNÍ MODEL	65
11.2	DRUHÁ VERZE NÁHRADNÍHO MODELU.....	67
11.3	TŘETÍ VERZE NÁHRADNÍHO MODELU	69
11.4	ZHODNOCENÍ.....	71
12.	SROVNÁNÍ FUNKČNOSTI IMPLEMENTOVANÝCH SYSTÉMŮ.....	72
12.1	NEREGULOVANÁ FREKVENCE.....	72
12.2	REGULACE FREKVENCE POMOCÍ GPS	73
12.3	REGULACE FREKVENCE POMOCÍ RDS.....	75
12.4	REGULACE FREKVENCE POMOCÍ DCF77	76
13.	SHRNUTÍ METOD.....	78
14.	ZÁVĚR.....	81
	LITERATURA.....	83
	SEZNAM PŘÍLOH.....	85

SEZNAM OBRÁZKŮ

2.1	Průběh komunikace přes technologii BLE [1]	14
2.2	Průběh připojování (slave zelený signál, master béžový signál) [2]	15
2.3	Rozdíl v přechodu do klidového stavu u master a slave zařízení [2]	16
2.4	Princip PTP synchronizace [4]	17
3.1	Dosah vysílače DCF77 [7]	20
3.2	Vysílaný signál z vysílače DCF77 [8]	21
5.1	Příklad výsledného grafu Allanové odchylky [15]	27
6.1	Allanova odchylka bez synchronizace	29
7.1	Schéma zapojení dané metody s měřením zpoždění synchronizace [16] [17]	31
7.2	Histogram posunu přijmutí synchronizačního pulsu, při nezatíženém slave mikrokontroleru	32
7.3	Histogram posunu přijmutí synchronizačního pulsu, při zatíženém slave mikrokontroleru matematickými úlohami	33
7.4	Průběh signálu před a po projití 700 metrovým kabelem (modrý průběh zobrazuje signál před průchodem, zelený po průchodu)	34
7.5	Posun na dlouhém kabelu (modrý průběh zobrazuje signál před průchodem, zelený po průchodu)	35
7.6	Histogram posunu přijmutí synchronizačního pulsu, při prodloužení spojení mezi mikrokontrolery	36
7.7	Schéma zapojení dané metody s měřením zpoždění synchronizace, při přivedení zpětné smyčky do hlavního mikrokontroleru [16] [17]	37
7.8	Histogram posunu přijmutí synchronizačního pulsu, při zavedení zpětné smyčky	38
8.1	GPS přijímač NEO-6M [18]	40
8.2	Histogram periody PPS signálu pro GPS	41
8.3	Schéma zapojení pro měření posunu na přijímačích GPS [21] [21]	42
8.4	Histogram posunu signálu vzniklých na přijímačích	42
8.5	Příklad vytvoření zpoždění na základě fáze hodin mikrokontroleru	44
8.6	Příklad eliminace zpoždění na základě fáze hodin mikrokontroleru	44
8.7	Histogram posunu signálu při připojení mikrokontroleru	45
8.8	Ukázka nastavení hodin v STM32	47
8.9	Allanova odchylka pro STM32 synchronizované pomocí metody využívající GPS	48
9.1	Rádiový přijímač SI4703 [22][22]	49
9.2	Histogram naměřených period mezi jednotlivých pulzy	50
9.3	Histogram naměřených period zprůměrovaných pulzů	51
9.4	Histogram naměřených posunů mezi dvěma moduly	52
9.5	Histogram zprůměrovaných posunů mezi dvěma moduly	53
9.6	Obrázek zobrazující vysvětlení rovnic 9.1 a 9.2	54
9.7	Histogram naměřených period RDS pulzů pomocí GPS	55
9.8	Histogram naměřených period RDS pulzů pomocí GPS při použití průměrování	56
9.9	Allanova odchylka pro RDS signál	57
10.1	Modul přijímače DCF signálu RCC02 CO3A [23]	58
10.2	Perioda DCF signálu neupravena	59
10.3	Perioda DCF signálu	60
10.4	Perioda DCF signálu, při využití průměrování 6 hodnot	61
10.5	Posun GPS a DCF signálu, kdy modrý signál je pro GPS, zatímco oranžový pro DCF	62
10.6	Posun pulzů na dvou modulech DCF	63
10.7	Posun pulzů na dvou modulech DCF s využitím průměrování	63
11.1	Frekvence mikrokontroleru	66

11.2	Odchylka frekvence mikrokontroleru při použití modelu verze č.1	67
11.3	Graf zachycující změnu frekvence na základě změny teploty	68
11.4	Odchylka frekvence mikrokontroleru při použití modelu verze č.2	69
11.5	Odchylka frekvence mikrokontroleru při použití modelu verze č.3	71
12.1	Změna odchylky v čase u neregulované frekvence	73
12.2	Změna odchylky v čase u regulované frekvence pomocí GPS.....	74
12.3	Přechodový děj u regulace pomocí GPS	74
12.4	Změna odchylky v čase u regulované frekvence pomocí RDS	75
12.5	Přechodový děj u regulace pomocí RDS	76
12.6	Změna odchylky v čase u regulované frekvence pomocí DCF77	77
12.7	Přechodový děj u regulace pomocí DCF77	77
13.1	Porovnání Allanových odchylek	80

SEZNAM TABULEK

3.1 Srovnání teoretických metod.....	22
7.1 Doby zpoždění mezi příjmem synchronizačními signály	39
13.1 Porovnání posunu signálu mezi jednotlivými mikrokontrolery pro jednotlivé metody	78
13.2 Porovnání period signálu pro jednotlivé metody.....	79

ÚVOD

Tato diplomová práce, se zabývá tvorbou různých metod časové synchronizace pro mikrokontrolery a jejich testování, pomocí vhodných měření. Časová synchronizace se provádí, především v systémech, které se skládá z více řídicích jednotek, ale lze se s nimi setkat i v systémech, kde existuje pouze jedna řídicí jednotka. V takovém případě, se jedná většinou o synchronizaci s globálním časem, např. UTC.

V případě, že se v systému nachází více řídicích jednotek, lze provést pouze synchronizaci času uvnitř systému, tzv. relativní synchronizaci. Při níž není snaha systém synchronizovat s globálním časem, lze pouze časově sjednotit všechny řídicí jednotky uvnitř daného systému.

S touto problematikou, se lze setkat v různých odvětvích jako je například v telekomunikaci, energetice nebo IoT. Kdy požadována přesnost synchronizace je závislá na druhu aplikace. Pokud tedy budeme uvažovat např. měření teploty v různých místnostech bytového komplexu, je patrné, že výsledná synchronizace bude stačit s přesností na jednotky sekund. Pokud však budeme chtít měřit vibrace na mostě, tak si s takto nízkou přesností nevystačíme a je tedy třeba zpřesnit výslednou synchronizaci na desítky mikrosekund.

V teoretickém úvodu jsou představeny různé metody časové synchronizace, kdy jsou tyto metody rozděleny na synchronizaci relativní a absolutní. Metody relativní, jsou metody, které jsou synchronizovány pouze vnitřně, nejsou tedy synchronizovány s globálním časem. Absolutní metody jsou naopak s globálním časem synchronizovány.

Kapitola pátá se poté zabývá teoretickými možnostmi měření časové synchronizace. Jsou zde představeny dvě metody, kdy každá z těchto metod měří jiný druh chyby, který může vzniknout při časové synchronizaci.

V kapitolách šestá až deset, jsou poté představeny jednotlivé metody, jejich implementování a také jsou představeny jejich naměřené vlastnosti.

Kapitola jedenáctá je poté věnována představení vytvořených náhradních modelů, které jsou použity, pokud dojde k výpadku přijímání signálů. Jsou v ní představeny tři verze, kdy vždy další verze vylepšuje tu předcházející v ohledech, ve kterých předcházející verze selhala.

V kapitole dvanáct, je následně ukázána funkčnost jednotlivých systému. Kdy jsou zde zobrazeny výsledné odchylky od požadované frekvence.

V závěrečné kapitole, jsou poté jednotlivé výsledky porovnány a vyhodnoceny.

1. ČASOVÁ SYNCHRONIZACE

Nejprve se pojdme podívat na to, co to vlastně je časová synchronizace mikrokontroleru a proč ji je třeba provádět. Časová synchronizace je označení pro činnost, při které se snažíme sjednotit časy jednotlivých řídicích jednotek v systému. S tímto se například můžeme setkat v hodinách, které jsou automaticky synchronizovány pomocí UTC času.

Pro provedení dokonalé synchronizace u mikrokontrolerů je třeba splnit 3 stupně synchronizace. Kdy pod pojmem dokonalé synchronizace si je třeba představit stav, kdy vnitřní hodiny mikrokontroleru budou ve všech mikrokontrolerech totožné, tak že by se dalo říct, že hodiny z jednoho mikrokontroleru jsou přivedeny přímo na hodiny druhého. Prvním stupněm je označeno přivedení synchronizačního pulzu v totožný čas, druhým stupněm se označuje sjednocení rychlosti kmitů vnitřních hodin a posledním stupněm je označeno srovnání fáze kmitů daných hodin.

Tato diplomová práce se zabývá měřením a provedením pouze prvních dvou stupňů synchronizace. Přičemž důvodem proč, nemá cenu se zabývat 3 stupněm synchronizace je nemožnost nastavení přesných frekvencí vnitřních hodin v mikrokontroleru, ale pouze jejich přiblížením. To je způsobeno minimálním krokem, který se pohybuje v rozmezí 3 až 10 kHz. Přičemž bližší vysvětlení, proč tomu tak je, lze nalézt v kapitole 8.4.1.

Jednou z důležitých otázek, při synchronizaci je, jak často je jí třeba provádět. Na to neexistuje jednoduchá odpověď, ale záleží na více faktorech, jako je požadována přesnost, zvolený stupeň synchronizace či stabilita vnitřních hodin mikrokontroleru. Požadovaná přesnost je většinou závislá na druhu aplikace, určitě bude vyžadována vyšší přesnost při měření vibrací než u měření teploty v obytném komplexu.

Při volbě stupně, je třeba dbát na volbě mikrokontroleru, neboť některé mikrokontrolery nemusí umožňovat nastavování přesné frekvence hodin. Při tomto bodu je tedy třeba brát ohled na minimální krok změny frekvence hodin. Důležitým parametrem hodin mikrokontroleru, je i přesnost vnitřního oscilátoru.

S časovou synchronizací se setkáme v různých odvětvích jako je telekomunikace, energetice, stavebním inženýrství nebo IoT. Jednoduše by se dalo říct, všude tam kde je třeba zaznamenávat časové značky měření nebo je třeba daná měření provádět v totožný moment.

Časovou synchronizaci lze rozdělit na relativní a absolutní metody. Systémy využívající absolutní časovou synchronizaci se synchronizují podle celosvětově uznávaných časových standardů jako je např. UTC. Jejich výhodou oproti relativním synchronizacím, je v tom, že jednotlivá zařízení nemusí být mezi sebou propojena, ale mohou být synchronizována samostatně na základě přijímání globálního času.

Relativní synchronizace nevyužívají ke své synchronizaci globální čas, ale synchronizují se k určité události nebo jsou synchronizována na základě relativního času, který sdílí mezi sebou. Jejich výhodou pramení z nepotřebnosti být, jakkoliv spojeni s okolním světem.

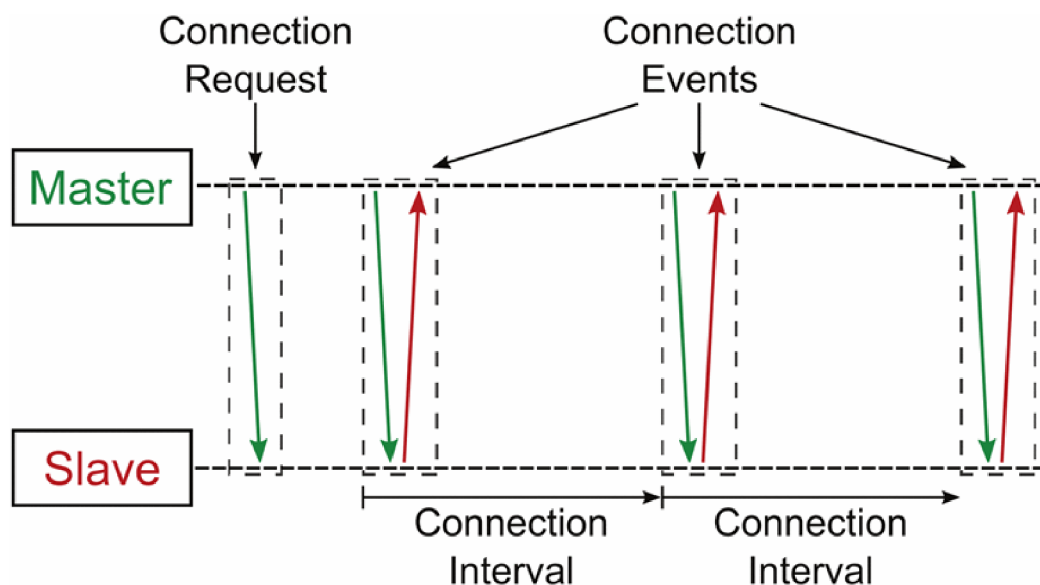
2. RELATIVNÍ METODY ČASOVÉ SYNCHRONIZACE

Jako první se podíváme na relativní metody časové synchronizace, kdy tyto metody nemusí být synchronizovány s globálním časem, ale jsou pouze synchronizovány mezi sebou.

2.1 Synchronizace na základě měření proudu pro BLE

2.1.1 Bluetooth low energy (BLE)

Jedná se o bezdrátovou komunikační technologii, která je určena pro aplikace s nízkou spotřebou. Tato technologie je založena na architektuře master-slave. Pro úsporu energie dochází k periodickému připojování a odpojování klienta od serveru. Následné připojování se provádí po uplynutí předem dohodnutého času, kdy tento interval určí master zařízení, jako jeden z parametrů, při prvotním spojení. Tento průběh je možné vidět na obrázku 2.1. [1]



Obrázek 2.1 Průběh komunikace přes technologii BLE [1]

2.1.2 Princip časové synchronizace pro BLE

Princip této metody spočívá v měření spotřeby proudu zařízení, kdy odebíraný proud se mění na základě stavu, ve kterém se systém nachází. To je možné pozorovat na obrázku 2.2, kde horní signál reprezentuje spotřebu slave zařízení a dolní signál reprezentuje spotřebu master zařízení. Jak je vidět, tak slave zařízení po přechodu z klidového stavu začne vysílat inzertní pakety na jednom z kanálů, což reprezentuje první špička v signálu. Následně poté, na daném kanále poslouchá, zda přišla odezva či nikoliv. Pokud žádný signál zpět nepříjde, tak se zařízení přesune na další kanál a pokusí se opětovně navázat

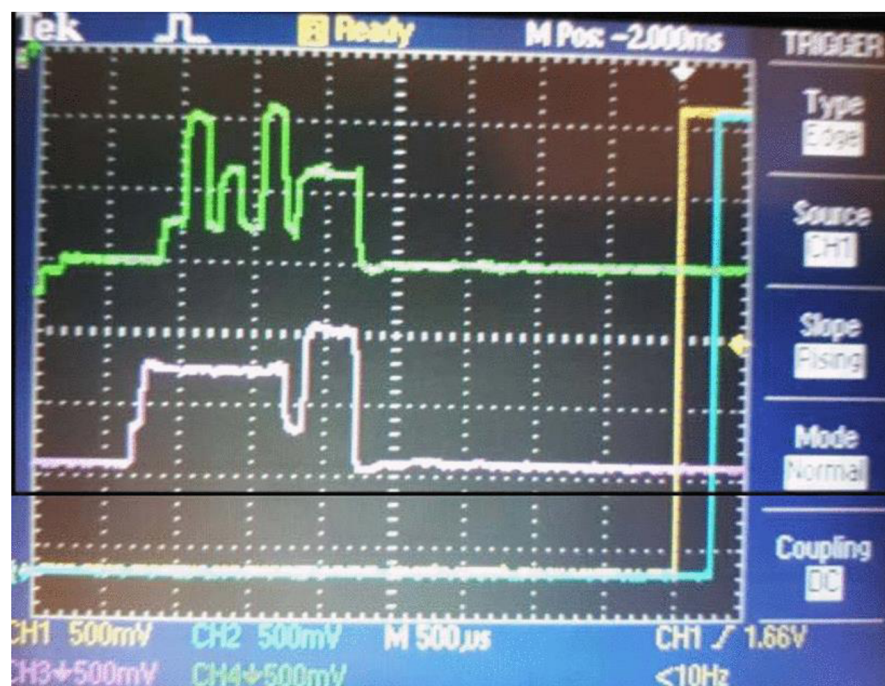
spojení na něm. [2]

Naopak master zařízení, se po probuzení z nečinnosti nastaví do režimu poslouchání a čeká na přijetí paketů ze slave zařízení. Po jejich přijetí, začne vysílat pakety s žádostí o připojení. Po odeslání toho požadavku, se obě zařízení přepnou do klidového režimu. [2]

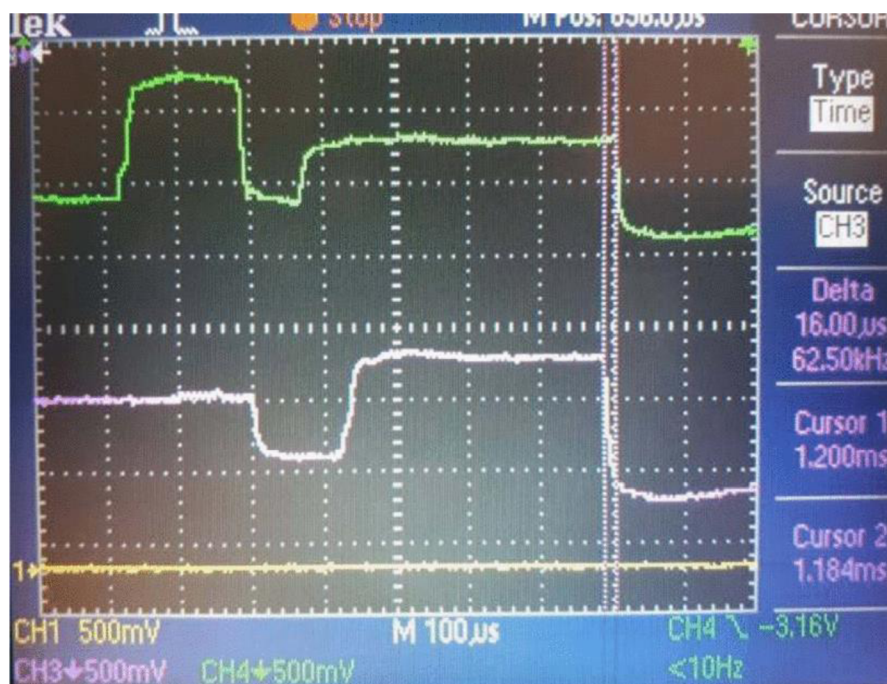
Právě tento přechod, lze využít pro časovou synchronizaci, neboť jejich časový rozdíl se pohybuje v řádu desítek μs , jak ukazuje obrázek 2.3. Podobného výsledku je možné dosáhnout i při odpojování komunikace. Zde je dobré podotknout, že tento jev není zaručen Bluetooth technologií, ale spíše bude záležet na konkrétním provedení výrobce Bluetooth zařízení. [2]

2.1.3 Měření přesnosti časové synchronizace pomocí BLE

Pro měření odebraného proudu, lze k obvodu připojit paralelně rezistor, ve funkci bočnicku, na kterém se bude zkoumat velikost napětí. Toto napětí, je poté přivedeno na mikrokontroler, který při poklesu napětí pod předem stanovenou úroveň, vyvolán synchronizační pulz. [2]



Obrázek 2.2 Průběh připojování (slave zelený signál, master bílý signál) [2]



Obrázek 2.3 Rozdíl v přechodu do klidového stavu u master a slave zařízení [2]

2.2 Synchronizace pomocí signálu od nadřízeného systému

Tato metoda, patří k nejjednodušším metodám časové synchronizace, je založena na principu posílání synchronizačních zpráv od nadřízeného systému, kdy jako nadřízený systém je zpravidla volen jeden z mikrokontrolerů. Ten poté, po uplynutí předem nadefinované časové době, vyšle synchronizační pulz.

I když přesněji řečeno, dochází k poslání pulzu po uplynutí, určitého počtu tiků hodin. Neboť hodiny v hlavním mikrokontroleru nejsou synchronizovány, čímž není zaručeno, že po uplynutí stejného počtu tiků uplyne stejně dlouhý časový úsek. Z čehož je patrné, že tato metoda vnímá jako hlavní čas, hodiny z hlavního mikrokontroleru.

2.2.1 Přenos synchronizačního pulzu

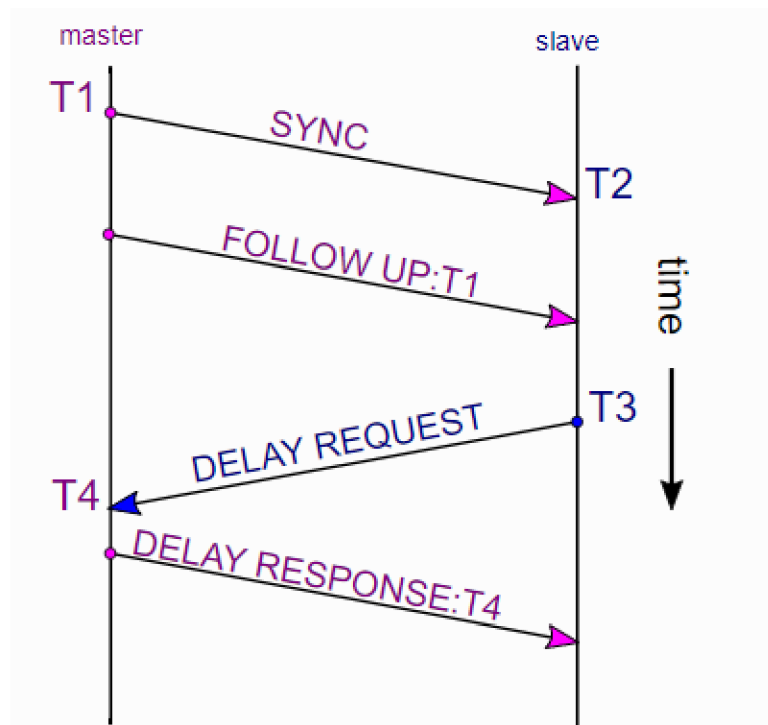
Při této metodě, je třeba propojit mezi sebou mikrokontrolery v systému. Z čehož je patrné, že na tomto propojení může docházet k zpoždění synchronizačního signálu. Kdy toto zpoždění je závislé na délce vedení a rychlosti přenosu signálu na daném vedení. Z toho tedy vyplývá, že tato metoda je závislá na výběru vedení a vzdálenosti mikrokontrolerů od sebe. Dále také tato metoda je omezena na maximální vzdálenost mikrokontrolerů od sebe, kdy tato vzdálenost je určena útlumem vedení.

Další zpoždění, které zde bude vznikat, je při vyhodnocení přijatého signálu na podřízeném mikrokontroleru. Existují však modifikace této metody, např. přidání zpětné smyčky, kdy tato metoda má za cíl odstranit či zredukovat výslednou chybu na vodiči a

chyby vznikající, při vyhodnocení signálu. Využití a potlačení zpětné smyčky je popsáno v kapitole 7.1.4.

2.3 PTP protokol

PTP protokol je ve své podstatě relativní metoda, která synchronizuje čas podle hodin hlavního zařízení. Následně až toto hlavní zařízení je synchronizováno pomocí dalších metod jako např. GPS nebo pomocí atomových hodin přes internet. Jednotlivá zařízení jsou mezi sebou propojena pomocí síťového kabelu, přes které probíhá i následná synchronizace. Z toho plyne, že pokud nebudeme synchronizovat hlavní zařízení, bude tato metoda zdánlivě připomínat předcházející metodu. Kdy jejich rozdíl je však v možnosti využití různých topologií. Zatím co předcházející metoda funguje pouze pro hvězdicovou topologii, tato metoda je schopna pracovat ve vícero topologiích. Dále tato metoda již od začátku počítá s možností dopočtu zpoždění způsobený přesunem signálu po vodiči. Tuto problematiku, řeší pouze pokročilejší modifikace předcházející metody.[3]



Obrázek 2.4 Princip PTP synchronizace [4]

2.3.1 Princip synchronizace

Časová synchronizace je zde prováděna pomocí synchronizačních zpráv posílaných přes síťový kabel. Nejprve je poslána zpráva od master zařízení s informací o prováděné časové synchronizaci. Dále je poslána zpráva s časem odeslání synchronizačního požadavku, kdy čas odeslání je označen T1. Hodnota T2 je poté čas, kdy byla tato zpráva

přijata slave zařízením. Po přijetí zprávy od mastera, je následně poslána zpráva od slave zařízení s požadavkem na posláni druhé zprávy od master zařízení, s tím že čas, kdy byl poslán požadavek, je označen jako T3 a doba kdy byla tato zpráva přijmu ta na master zařízení je označena jako T4. Tento popis je možné vidět na obrázku 2.4. [3]

Z těchto času, lze poté pomocí rovnic 2.1 a 2.2 , vypočítat zpoždění na spojení a hodnotu offsetu hodin mezi jednotlivými zařízeními. A na základě vypočítané hodnoty offsetu upravit vnitřní hodiny slave zařízení a tím provést první stupeň časové synchronizace. Tyto rovnice předpokládají stejně dlouhé zpoždění při šíření zprávy. Čehož není vždy možné dosáhnout na základě fungování ethernetu. [3]

Pro provedení druhé stupně časové synchronizace, by bylo nutné ukládat čas provedení předcházející synchronizace a současný čas. Kdy rozdílem těchto dvou časů, bychom získali uplynulý čas mezi synchronizacemi. Z čehož lze následně dopočítat, ke kolika tikům hodin mělo dojít a ke kolika doopravdy došlo. Rozdílem těchto dvou hodnot poté získáme odchylku, která by byla přivedena na vstup regulátoru.

Rovnice pro výpočet zpoždění na spojení a offsetu hodin

$$Zpozdění = ((T2 - T1) + (T4 - T3))/2 , \quad 2.1$$

$$offset = ((T2 - T1) - (T4 - T3))/2 , \quad 2.2$$

kde T1 je čas odeslání první zprávy, T2 je čas přijmutí první zprávy, T3 je čas odeslání požadavku na změření zpoždění a T4 je čas odpovědi na zpoždění.

3. ABSOLUTNÍ METODY ČASOVÉ SYNCHRONIZACE

Tyto metody, využívají pro svoji synchronizaci okolní systémy, které jim umožňují získávat informaci o globálním čase. Jejich velké výhodou oproti relativní, je v tom, že nemusí být mezi sebou propojeny. Naopak nevýhodou je nutnost, aby byly schopny přijímat zprávy od globálního systému.

3.1 GPS

System GPS je možné využívat pro určení polohy ale také ho lze využívat pro získávání přesného času, který je synchronizován s UTC. Tento čas je spolu s dalšími daty z družice, vysílán v navigačních zprávách na frekvenci 1575,42 MHz s šířkou pásma $\pm 1,023$ MHz. [5]

3.1.1 Princip časové synchronizace pomocí GPS

Pro časovou synchronizaci je nejprve potřeba přijmout navigační zprávu pomocí GPS přijímače, který je schopen na základě přijmutých dat generovat pulzy tzv. PPS signály a přeposílat NMEA zprávy. Tyto zprávy obsahují různé informace o aktuální poloze, počtu snímaných satelitů či UTC času. [5]

Právě pomocí těchto PPS pulzů, lze určit začátek další sekundy. Kdy při využití 1 PPS signálu, je při nástupné hraně pulzu vždy oznámen začátek další sekundy. Popřípadě pokud je generováno více pulzů za jednu sekundu, je nutné nalézt pulz, který se odlišný od ostatních, to může být například doba jeho trvání. [5]

Synchronizace je poté prováděna pomocí těchto pulzů, kdy při příchodu daného pulzu je oznámen začátek další sekundy. Pokud tedy budeme uvažovat pouze synchronizaci prvního stupně, při přijmutí PPS signálu je přičtena sekunda do registrů, které slouží k udržování přesného času.

Pokud budeme uvažovat druhý stupeň synchronizace, kdy dochází k úpravám frekvence vnitřních hodin, tak tyto pulzy využíváme k ohraničení doby, v které dochází k počítání tiků vnitřních hodin. Výsledný počet tiků hodin je následně použit ve zpětné vazbě k úpravě frekvence hodin.

Posunutí synchronizace na třetí stupeň by poté mohlo být provedeno srovnání naběžné hrany hodin s naběžnou hranou PPS signálu.

3.1.2 Nepříznivé vlivy

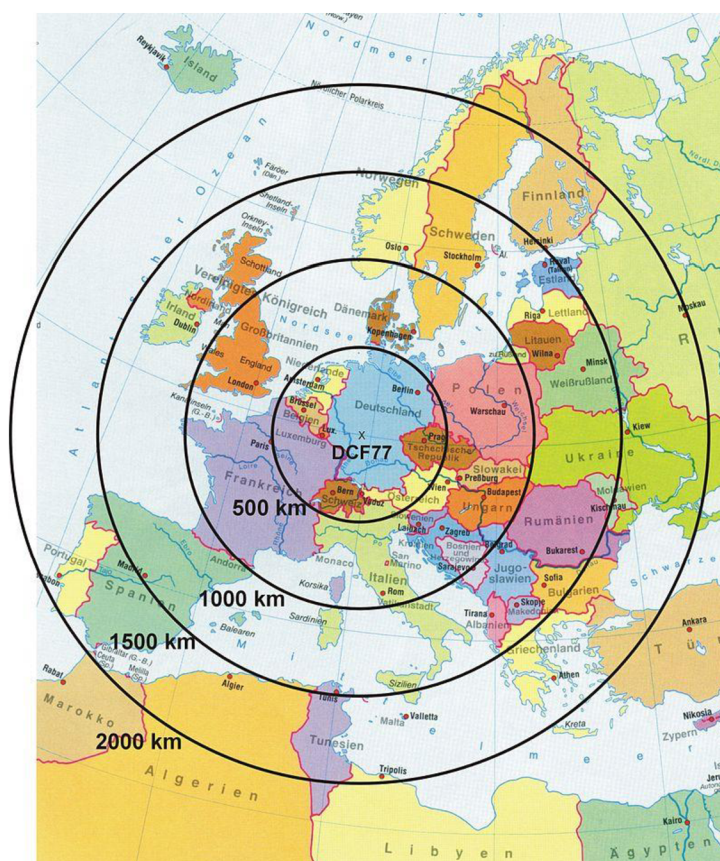
Tato metoda stojí a padá na přijatí zpráv od satelitů, pokud nejsem schopni zachytit zprávy od satelitů, stává se tato metoda nefunkční. Z toho důvodu, je třeba volit umístění a parametry přijímače takové, aby nedocházelo k výpadkům zpráv příliš často.

Jedním z největších problémů pro GPS jsou budovy, hory či hustá vegetace. Proto je třeba přijímače umístit do takových míst, kde se tyto překážky nebudou nacházet v cestě mezi satelity a přijímačem.

Dalšími vlivy, který snižují kvalitu přijímání zpráv jsou meteorologické podmínky, kdy při mohutném dešti nebo sněžení může dojít k absorpci či odrazení signálu.

3.2 DCF77

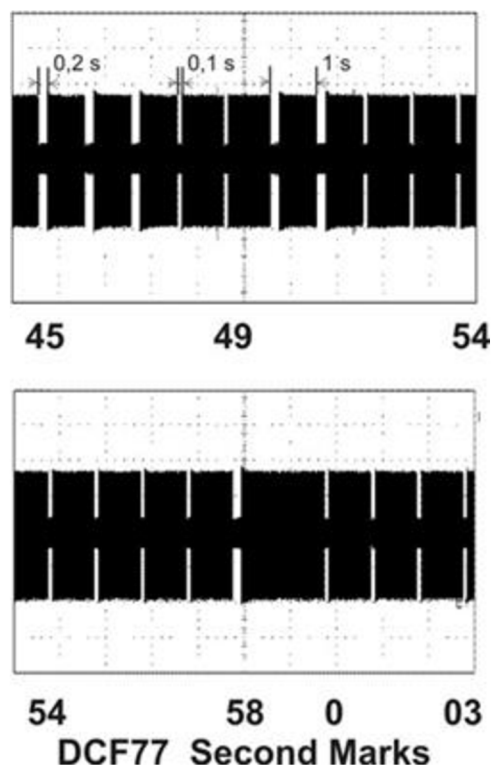
DCF77 je označení pro rádiovou stanici umístěnou ve Frankfurtském regionu (Německo) vysílající dlouhé vlny na frekvenci 77,5 kHz. Tato stanice slouží k vysílání informací o čase, přičemž její vysílání lze použít k časové synchronizaci. Maximální dosah této stanice je 2000 km, přičemž její dosah je ukázán na obrázek 3.1. [6]



Obrázek 3.1 Dosah vysílače DCF77 [7]

3.2.1 Časová synchronizace pomocí DCF77

Principiálně je tato metoda podobná metodě využívající GPS. Kdy při této metodě, již samotný vysílač posílá PPS signály, které jsou poté přijímány na anténě. Tyto PPS signály jsou prezentovány jako pokles velikosti amplitudy o 85 % nominální velikosti. Doba, po kterou je amplituda snížena je závislá na vysílané logické úrovni, kdy pro logickou 0 je to doba 0,1 sekundy a pro logickou 1 je stanovena doba 0,2 sekundy. Tyto logické úrovně jsou používány k přenosu UTC času v BCD kódu. Při poslední sekundě v každé minutě, k poklesu amplitudy nedochází, aby bylo možné jednoznačně určit začátek další minuty a tím i začátek BCD kódu. [8]



Obrázek 3.2 Vysílaný signál z vysílače DCF77 [8]

3.3 Synchronizace pomocí RDS dat z radiových stanic

RDS neboli rádiový datový systém slouží jako postranní pásmo u FM radiových stanic, k přenosu dodatečných informací ve vysílání. To může být název hrající písně, název rozhlasové stanice či přenos času a datumu. [9]

Na první pohled by se mohlo zdát, že tento čas bude možné využít k časové synchronizaci. Ovšem tento čas je pro synchronizaci nepoužitelný, neboť jeho minutová hrana má stanovenou přesnost ± 100 ms. [9]

Pro časovou synchronizaci, lze však využít faktu, že známe počet přenášených bitů, který je 104 bitů rozdělených do 4 skupin po 26 bitech a rychlost, kterou jsou tyto bity odesílány, která je 1187,5 bps. [9]

3.3.1 Princip časové synchronizace pomocí RDS pásma

Časová synchronizace je zde prováděna pomocí FM přijímače, který po dekódování všech skupin dat vyvolá impuls na svém výstupu. Přičemž ten výstup, je poté přiveden na vstup mikrokontroleru jako externí přerušení a je využit na synchronizaci stejným způsobem jako u GPS nebo DCF77 metody. Jediný rozdíl je zde v době mezi pulzy, kdy u předchozích metod tato doba byla 1 sekunda, zde je to 87,576 ms. Kdy tato doba vyplývá z rychlosti odesílání a celkového počtu bitů. [9]

3.4 Shrnutí teoretických metod

Na závěr popisu teoretických metod časové synchronizace je třeba tyto metody porovnat mezi sebou a vyzdvihnout jejich výhody a nevýhody oproti ostatním metodám. Byly představeny 3 relativní metody a 3 absolutní metody. Kdy výhoda absolutních metod je v snadné porovnání časů na základě globálního času. Toto u relativních metod neplatí, neboť u těchto metod není globální čas, ale pouze čas, který určuje hlavní zařízení. Z toho vyplývá, že tedy jedna sekunda hlavního mikrokontroleru nemusí být rovna jedné sekundě globálního času. Tento fakt ovšem v relativních systémech nevede, neboť naší snahou je srovnat hodiny podřízených mikrokontrolerů s hlavním mikrokontrolerem.

V tabulce 3.1, jsou představeny základní výhody a nevýhody jednotlivých metod. Hlavní důraz je zde kladen na maximální vzdálenost mikrokontrolerů od sebe, ovlivnění nepříznivých vlivů, při příjmu bezdrátových signálů a schopnosti synchronizace, při výpadku jednoho z mikrokontrolerů.

Tabulka 3.1 Srovnání teoretických metod

Metoda	Vlastnosti
BLE	<ul style="list-style-type: none">- Relativní metoda- Bezdrátový přenos Výhody: <ul style="list-style-type: none">- Malá spotřeba elektrické energie Nevýhody: <ul style="list-style-type: none">- Maximální vzdálenost mikrokontrolerů určena dosah BLE- Při výpadku hlavního mikrokontroleru, dojde k výpadku synchronizace
Nadřazený systém	<ul style="list-style-type: none">- Relativní metoda- Přenos pomocí vodiče Výhody: <ul style="list-style-type: none">- Metoda není ovlivněna schopností zachytit elektromagnetický signál- Není nutné pořizovat další součástky Nevýhody: <ul style="list-style-type: none">- Maximální vzdálenost mikrokontrolerů určena útlumem na vodiči- Při výpadku hlavního mikrokontroleru, dojde k výpadku synchronizace
PTP	<ul style="list-style-type: none">- Relativní metoda- Přenos pomocí síťového vodiče Výhody: <ul style="list-style-type: none">- Metoda není ovlivněna schopností zachytit elektromagnetický signál

PTP	<ul style="list-style-type: none"> - Při připojení na ethernet není určena maximální vzdálenost mikrokontrolerů <p>Nevýhody:</p> <ul style="list-style-type: none"> - Při výpadku hlavního mikrokontroleru, dojde k výpadku synchronizace - Připojení na ethernet je náhodné, což ovlivňuje přesnost synchronizace
GPS	<ul style="list-style-type: none"> - Absolutní metoda - Přenos pomocí GPS zpráv <p>Výhody:</p> <ul style="list-style-type: none"> - Při výpadku jednoho mikrokontroleru nedojde k výpadku synchronizace v celém systému. - Není určena maximální vzdálenost mikrokontrolerů <p>Nevýhody:</p> <ul style="list-style-type: none"> - Metoda je ovlivněna schopností zachytit elektromagnetický signál ze satelitu - Nutné pořídit GPS přijímač
DCF77	<ul style="list-style-type: none"> - Absolutní metoda - Přenos pomocí dlouhých vln <p>Výhody:</p> <ul style="list-style-type: none"> - Při výpadku jednoho mikrokontroleru nedojde k výpadku synchronizace v celém systému. <p>Nevýhody:</p> <ul style="list-style-type: none"> - Metoda je ovlivněna schopností zachytit elektromagnetický signál z vysílače - Maximální vzdálenost mikrokontrolerů určena dosahem vysílače
RDS	<ul style="list-style-type: none"> - Absolutní metoda - Přenos pomocí elektromagnetických vln <p>Výhody:</p> <ul style="list-style-type: none"> - Při výpadku jednoho mikrokontroleru nedojde k výpadku synchronizace v celém systému. <p>Nevýhody:</p> <ul style="list-style-type: none"> - Maximální vzdálenost mikrokontrolerů určena dosahem vysílače - Metoda je ovlivněna schopností zachytit elektromagnetický signál z vysílače

4. POUŽITÝ HARDWARE

4.1 Mikrokontroler Nucleo-F303K8

Pro vytvoření a zpracování synchronizačních signálů byly využívány mikrokontrolery Nucleo-F303K8. Tyto mikrokontrolery byly použity pouze pro měření zpoždění signálu mezi mikrokontrolery, neboť nelze měnit jejich frekvenci hodin pomocí PLL.

Tyto mikrokontrolery obsahují CPU LQFP32, které pracuje na frekvenci 8 MHz generovaných z vnitřního RC oscilátoru. Dále tyto mikrokontrolery obsahují 25 I/O, dvě komunikace USART, I2C, SPI a možnost připojení sběrnice CAN. Napájení toho mikrokontroleru se může pohybovat 3,3 až do 12 V, přičemž následná vnitřní logika pracuje s napětím 3,3 V. [10]

4.2 Mikrokontroler STM32F769I-DISCO

Pro měření přesnosti frekvence hodin, tedy druhého stupně časové synchronizace, byl použit mikrokontroler STM32F769I-DISCO. Tyto mikrokontrolery umožňují měnění frekvence hodin pomocí PLL s krokem v rozmezí 3 až 10 kHz. Uvnitř mikrokontroleru je umístěno CPU STM32F769NIH6, které je schopno pracovat na frekvenci až 216 MHz. K programování mikrokontrolerů byl využit program STM32CubeIDE ve verzi 1.13.2. [11]

4.3 Analog Discovery 2

Pro měření časového posunu synchronizačních signálů, byl využit osciloskop Analog Discovery 2, který je schopno měřit signály s rychlostí až 100 MS/s, s maximální rozlišení na 14 bitů. Maximální velikost bufferu je 16384 na kanál. Změřený signál je poté možné pomocí sériové linky přenášet do počítače. [12]

4.4 Agilent 53131A

Čítač Agilent 53131A byl použit pro měření Allanovy odchylky, kdy měřil frekvenci kmitů z výstupu mikrokontroleru. Po zjištění, že osciloskop Analog Discovery není schopno zachytit přesně zpoždění na GPS přijímačích, byl tento přístroj využit i pro měření zpoždění signálu u GPS metody.

Čítač je schopno měřit frekvence až o hodnotě 225 MHz s rozlišením 10 bitů. Při měření času je schopno rozlišení časového intervalu na 500 ps. Pro komunikaci s PC je vybaven GPIB sběrnici, která je schopna přenášet až 200 měření za sekundu. Pro čtení těchto dat byl využit program LabVIEW ve verzi 2022 Q3. [13]

Nejistota toho čítače je závislá na více parametrech, a to na naměřené době, chybě způsobené, při sledování úrovně signálu, který spouští a vypíná měření a na počtu

použitých kanálů. Kdy, pokud je využit více než jeden kanál, tak je třeba nejistotu zvětšit o hodnotu $\pm 1,5$ ns. Jak se ukázalo, při výpočtech pro všechny měření, tato přičtená nejistota je hlavní zdrojem nejistoty daného měřicího přístroje, neboť i při výpočtu nejhoršího možné scénáře, byla stále tato nejistota vyšší o 5 řádů, než nejistota vypočítána na naměřeném čase. Rovnici pro výpočet nejistoty je možné nalézt na straně 10 v datasheetu výrobce. [13][19]

5. ZPŮSOBY VYHODNOCOVÁNÍ METOD

V této kapitole se podíváme na jednotlivé metody měření. Přičemž byly zvoleny dvě metody měření, kdy první slouží k porovnání rozdílu v přijmutí synchronizačního pulzu a zatím co druhá slouží k zjištění stability vnitřního oscilátoru. Pomocí těchto metod, lze zjistit přesnost prvního a druhého stupně časové synchronizace.

5.1 Měření rozdílu v přijmutí synchronizačního signálu

Toto měření slouží k vyhodnocování časového posunu v příjmu synchronizačního signálu mezi více mikrokontrolery. Jedná se o první stupeň časové synchronizace, kdy naším cílem je dosáhnout, co nejmenšího rozdílu na příjmech synchronizačních signálů. Z tohoto měření lze následně usoudit, jak velký rozdíl je v začátku sekund mezi jednotlivými mikrokontrolery. Kdy v ideálním případě chceme dosáhnout nulového rozdílu, čímž by všechny mikrokontrolery měly začátek sekund v totožný moment.

Protože, nejsem schopni určit, kdy mikrokontroler vnitřně zpracoval příchod synchronizačního pulzu, bude do kódu mikrokontroleru, přidána instrukce, aby po zpracování synchronizačního pulzu, došlo k vystavení signalizačního pulzu na jiném výstupu. Kdy následně tyto signalizační pulzy, budou poté přivedeny do měřícího přístroje, k vyhodnocení zpoždění mezi signalizačními pulzy.

Tento způsob vede k přidání funkce do systému, která za normálního provozu nebudou součástí toho systému a tím pádem, bude vyhodnocení obsahovat chybu vzniklou na přidávaných instrukcích, které vytvářejí signalizační pulz. Zde je však potřeba podotknout, že většina metod bude tuto chybu eliminovat na základě faktu, že tuto funkci budou muset provést všechny mikrokontrolery.

Měření této metody bude probíhat na logickém analyzátoru Analog Discovery 2, popřípadě na čítači Agilent 53131A, který je schopen měřit i menší časové rozdíly. Této vlastnosti je využito např. u synchronizace pomocí GPS, která má nižší rozdíl v posunech, než je schopen logický analyzátor změřit.

5.2 Měření stability frekvence vnitřního RC oscilátoru

Pro měření přesnosti druhého stupně časové synchronizace je nutné měřit stabilitu vnitřního RC oscilátoru, neboť naším cílem, při provádění druhé stupně synchronizace, je udržet hodnotu frekvence kmitů na konstantní hodnotě. Tuto stabilitu nám však kazí drift oscilátoru, který časem posouvá hodnotu frekvence mimo nastavenou hodnotu.

Měření stability frekvence vnitřního RC oscilátoru bude prováděno, pomocí čítače, na kterém bude měřena frekvence spínání jednoho z výstupu umístěného na mikrokontroleru. Tento výstup bude spínán každou sekundu, přičemž tento čas bude určován pomocí vnitřního časovače uvnitř mikrokontroleru. Z čehož plyne, že výsledná frekvence na výstupu bude závislá na frekvenci vnitřního RC oscilátoru.

Měření těchto hodnot bylo prováděno na přístroji Agilent 53131A, který byl ovládán pomocí programovacího prostředí LabVIEW. Pro následné vyhodnocování stability frekvence vnitřního RC oscilátoru, bude využívána statistická metoda zvaná Allanův rozptyl. Její výpočet byl také prováděn v prostředí LabVIEW, kdy kód pro její výpočet byl vytvořen vedoucím práce.

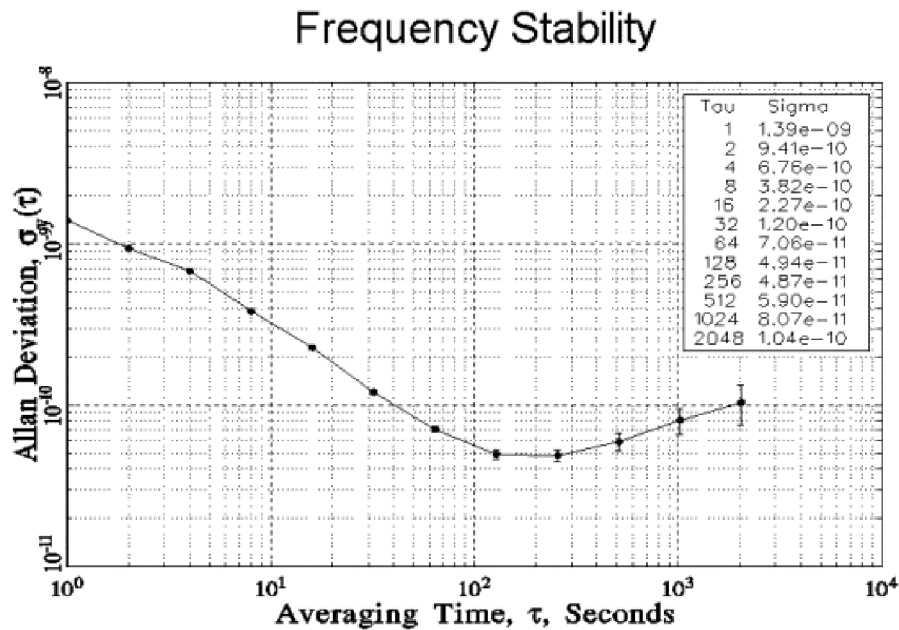
5.2.1 Allanův rozptyl

Allanův rozptyl byl vyvinut k určování míry stability frekvence v hodinách, zesilovačích či oscilátorech. Jeho hlavní přednost je v tom, že při jeho výpočtu je potlačena chyba způsobena šumem, čehož je dosaženo pomocí dvojího průměrování. Při tomto výpočtu však nedochází k potlačení nežádoucích systematických jevů, jako je stárnutí součástek nebo změna frekvence na základě teplotních změn. [14]

Allanův rozptyl se dá popsat jako průměr rozdílů mezi dvěma po sobě jdoucími shluky, kdy hodnota shluku je dána průměrem hodnot v daném shluku. Rovnice pro Allanův rozptyl zní

$$\sigma^2(T) = \frac{1}{2(K-1)} \sum_{k=1}^{K-1} (\bar{\Omega}_{k+1}(T) - \bar{\Omega}_k(T))^2, \quad 5.1$$

kde K je počet shluků a $\bar{\Omega}_k$ je průměrná hodnota k-tého shluku. [14]



Obrázek 5.1 Příklad výsledného grafu Allanové odchylky [15]

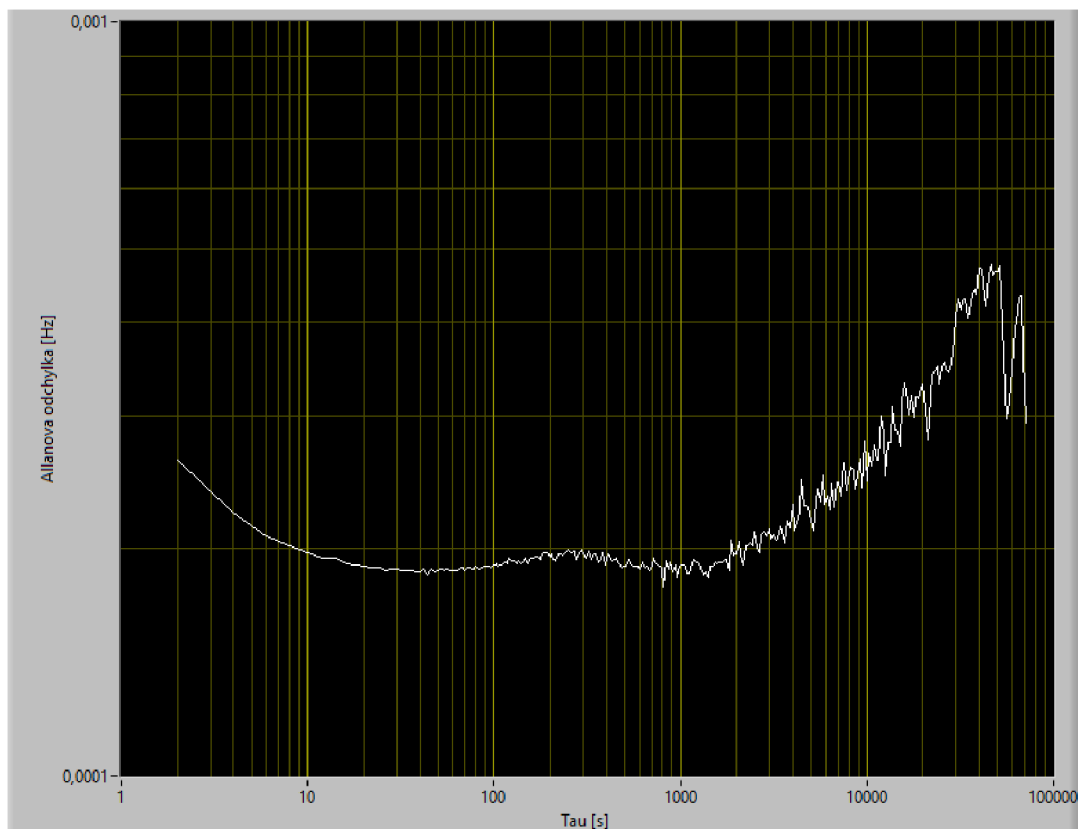
Spíše, ale než Allanův rozptyl je využívána Allanova odchylka, která je určena jako druhá odmocnina Allanova rozptylu. Důvod, proč je tato úprava prováděna je změna jednotek, neboť jednotky rozptylu jsou mocniny jednotek měřené veličiny, zatím co u odchylky jsou jednotky právě jednotky měřené veličiny. Výsledné reprezentace je poté uváděna ve formě grafu, jeho příklad je možné vidět na obrázku 5.1. Jak je vidět nejprve

dochází k poklesu odchylky, kvůli průběžnému průměrování šumu. Tak to postupuje, dokud není dosaženo minimální hodnoty odchylky, poté se nám Allanova odchylka opět začne navyšovat, neboť se začnou projevovat nežádoucí systematické jevy, jako je stárnutí součástek nebo změna frekvence na základě teplotních změn. V takovém případě, je již potřeba v takovýto moment provést synchronizaci. [14]

6. RC OSCILÁTOR

RC oscilátory umístěné v mikrokontrolerech většinou nebývají velmi přesné, ale jejich přesnost se pohybuje v řádech jednotek procent. Např. u mikrokontroleru STM32f769i-disco výrobce udává přesnost 1 %, při pokojové teplotě. Tato přesnost je však udávána jako maximální chyba. Pro získání reálné přesnosti v čase, je nutné tento RC oscilátor proměřit pomocí Allanovy odchylky, přičemž RC oscilátor nebude nijak synchronizován či řízen. Tím způsobem získáme znalost o přesnosti daného RC oscilátoru v daném mikrokontroleru.

Měření RC oscilátoru, bylo provedeno podle popisu v kapitole 5.2, kdy toto měření bylo spuštěno po dobu 4 dnů, aby došlo k projevu systematických jevů. Naměřenou Allanovu odchylku je možné pozorovat na obrázku 6.1. Jak je vidět, tak na začátku do času 10 sekund, tato odchylka klesá, tedy patrně v této části dochází k průměrování šumu. Poté zhruba do času 1000 sekund, je vidět, že RC oscilátor je stabilní a až následně po tomto čase se již začnou značně projevovat systematické chyby. Čímž se Allanova odchylka se začne zvyšovat, v tento moment je vhodné již provést synchronizaci.



Obrázek 6.1 Allanova odchylka bez synchronizace

7. SYNCHRONIZACE Z NADŘAZENÉHO SYSTÉMU

Při této metodě je jeden z mikrokontrolerů označen za master zařízení, který provádí synchronizaci, zatímco druhý mikrokontroler je označen za slave zařízení a je synchronizováno na základě synchronizace od mastera. Tato synchronizace je prováděna pomocí triggeru, který je přenášén přes měděný vodič. Tento signál by bylo možné přenášet i po jiných vodičích jako je optické vlákno či bezdrátově, čímž by se změnila i chyba způsobená na přenosu, o čemž částečně pojednává kapitola 7.1.3.

7.1 Rozdíl v přijmutí synchronizačního signálu

Nejprve je nutné určit rozdíl v přijmutí synchronizačního signálu na slave zařízení a odeslání toho signálu od master zařízení. Kdy tento rozdíl může být způsoben vícery parametry, jak je ukázáno na obrázku 7.1.

Nejprve tento rozdíl, může být způsoben na master mikrokontroleru. Kdy tento posun je způsoben na základě doby od zpracování instrukce, do doby vystavení synchronizačního signálu na výstupu mikrokontroleru. To je zobrazeno na obrázku 7.1 pod označením A. To zpoždění není použitou metodou měřeno, kvůli vysoké obtížnosti měření.

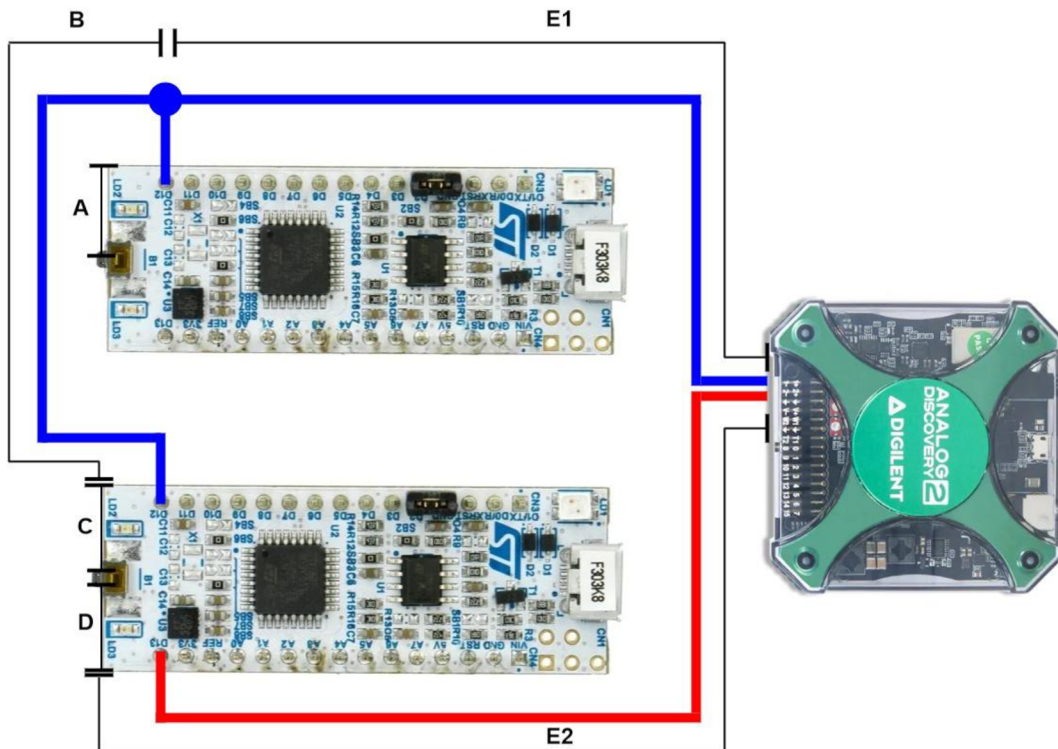
Další zpoždění (B) vzniká na vodiči. Kdy velikost toho zpoždění je závislá na délce a druhu materiálu. Jako poslední možnost, kde je možný vznik zpoždění signálu, je při zpracování přijmutí daného signálu na slave zařízení. Na obrázku 7.1 je to označeno pod písmenem C.

Další zpoždění jsou již pouze způsobena, kvůli měření předcházejících zpoždění, tedy při normálním provozu by již k těmto zpožděním nedocházelo. Jako první je doba mezi zpracováním instrukce o vystavení signálu na výstupu a jeho reálné vystavení na výstupu (D). Jako druhé zpoždění je zpoždění na vodičích vedoucích do měřícího zařízení, zde je snaha toto zpoždění eliminovat pomocí stejných vodičů od obou mikrokontrolerů (E1, E2).

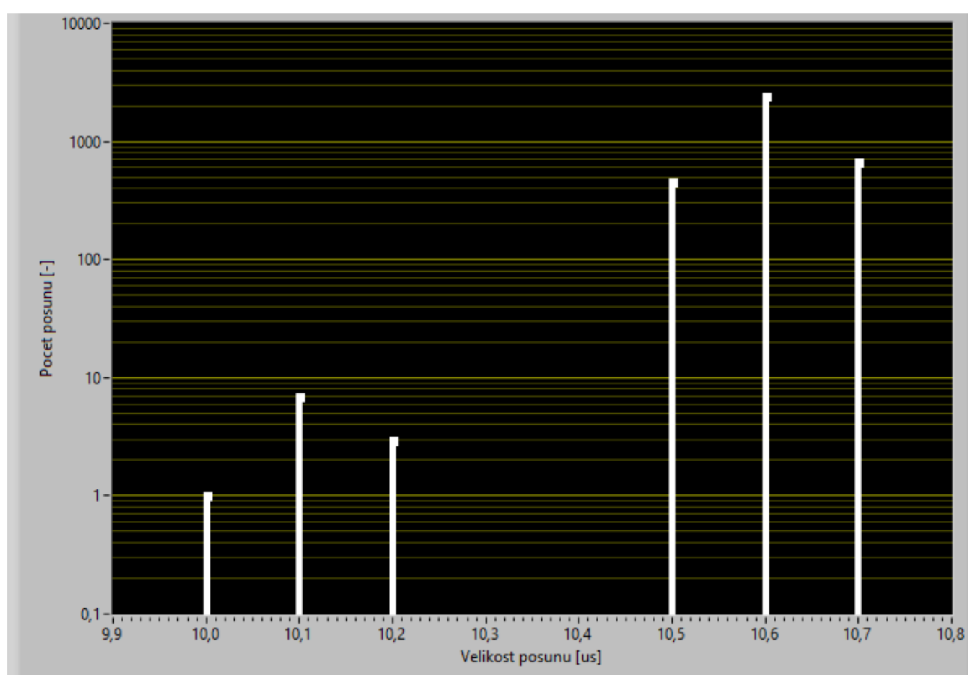
7.1.1 Nezatížený slave mikrokontroler

Jako první varianta, byl měřen posun na zařízení, při nezatíženém slave zařízení. Tedy mikrokontroler, nevykonával žádné operace, pouze čekal na vyvolání přerušení od synchronizačního pulzu, a následně tuto událost signalizoval vystavením signálního signálu na jiném výstupu. Časový posun toho signálu, byl poté porovnáván s časem přijmutí synchronizačního signálu od master mikrokontroleru. Na obrázku 7.2 je možné vidět histogram posunů, kdy největší množství z nich je v hodnotě 10,6 μs . Pokud na to použijeme statistické operace, tak zjistíme, že střední hodnota je 10,6045 μs , se směrodatnou odchylkou 0,0619 μs .

Při bližším pohledu na výsledný histogram, je možné pozorovat zajímavý výskyt pár hodnot v rozmezí 10 až 10,2 μs . Který zdánlivě, připomíná stejné rozložení hodnot jako je v rozmezí 10,5 až 10,7 μs . Pouze jsou tyto hodnoty posuny o 0,5 μs , což jsou přesně 4 kmity RC oscilátoru v mikrokontroleru. Z toho lze usoudit, že za nějakých okolností, které nebyly zjištěny, je mikrokontroler schopen zpracovat instrukci přerušování o 4 cykly rychleji. Podobný výskyt této skupiny posunů, se objevil i při testu zatíženého mikrokontroleru.



Obrázek 7.1 Schéma zapojení dané metody s měřením zpoždění synchronizace [16] [17]



Obrázek 7.2 Histogram posunu přijmutí synchronizačního pulsu, při nezatíženém slave mikrokontroleru

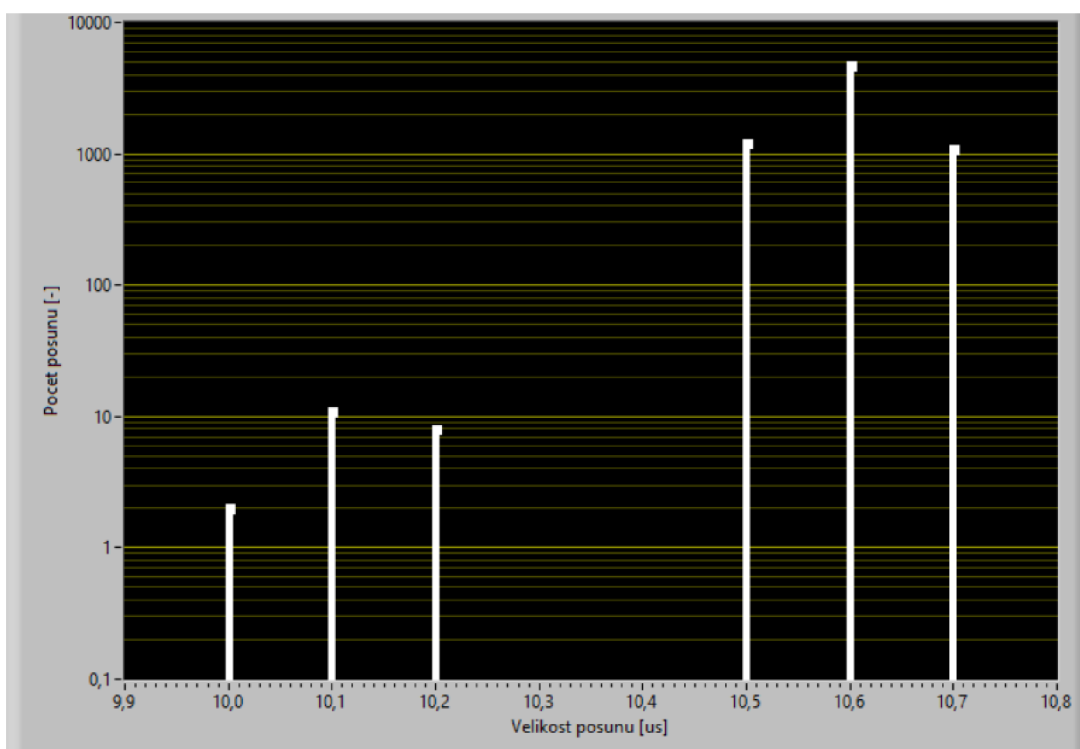
7.1.2 Zatížený slave mikrokontroler

Jako další varianta této metody, bylo zvoleno měření rozdílu v posunech signálu, při zatížení synchronizovaného mikrokontroleru. Kdy jako zatížení byly zvoleny nejprve výpočetní úlohy, které mikrokontroler neustále vypočítával. Přičemž byly použity nejrůznější matematické úlohy jako násobení, dělení či odmocniny.

Následně jako druhou možnost, bylo zvoleno zatížení mikrokontroleru pomocí komunikace I2C. Kdy z funkčních důvodů, byla nastavena vyšší priorita synchronizaci, před komunikací. Tedy pokud během komunikace, přišel požadavek na synchronizování, komunikace byla přerušena.

Obě tyto metody dosáhli podobných výsledků, jako dosáhl nezatížený mikrokontroler, kdy matematicky zatížený mikrokontroler měl, výslednou střední hodnotu 10,5967 μs a směrodatnou odchylkou 0,0626 μs . Podobně na tom byly hodnoty, při zatížení mikrokontroleru pomocí komunikace přes I2C, kdy výsledné hodnoty byly 10,5996 μs a 0,0666 μs . Tomu odpovídá i histogram na obrázku 7.3, kdy pro podobné výsledky je zde zobrazen pouze histogram zatíženého mikrokontroleru pomocí matematických úloh.

Z těchto hodnot, lze tedy usoudit, že nemá smysl řešit, zda je mikrokontroler zatížen či nezatížen, neboť střední hodnoty se od sebe liší o jednotky ns, při čemž doba mezi kmity vnitřních oscilátorů jednotlivých mikrokontrolerů, je nastavena na 125 ns, tedy frekvencí 8 MHz.



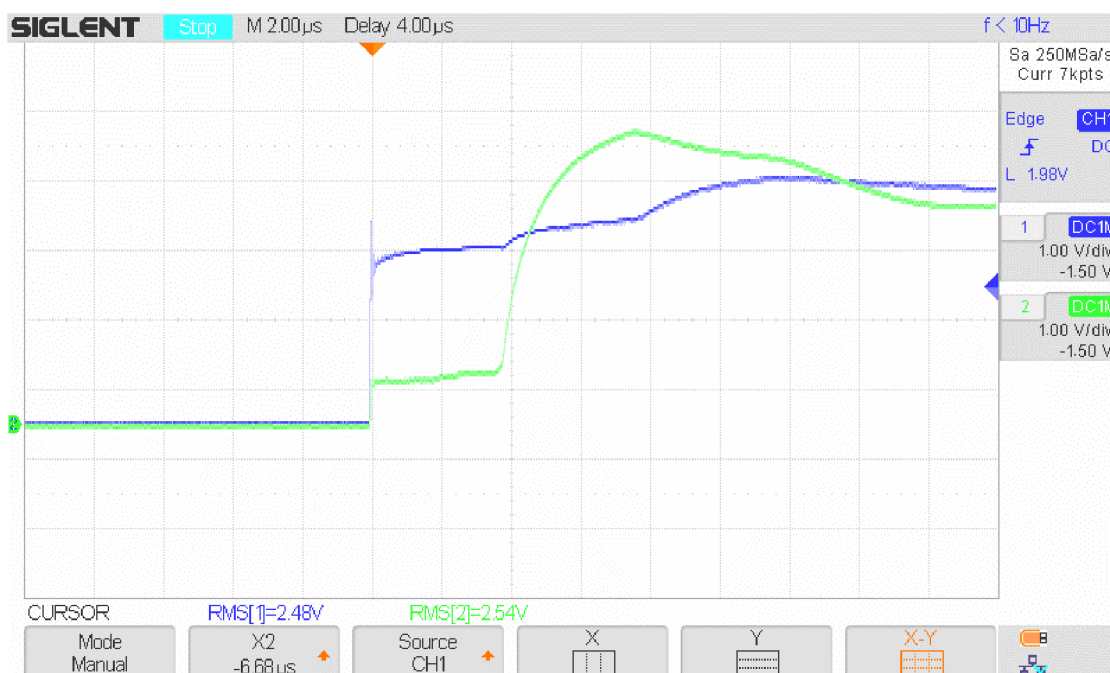
Obrázek 7.3 Histogram posunu přijmutí synchronizačního pulsu, při zatíženém slave mikrokontroleru matematickými úlohami

7.1.3 Prodloužení synchronizačního vedení

Jako další variantu pro proměření závislosti na posunu zpoždění, byla vybrána varianta s prodlouženým synchronizačním vedením. Tedy oproti předcházejícím metodám je mezi mikrokontrolery přidán 700 metrový kabel, kroucené dvojlinky. Kdy na základě poznatku z minulé kapitoly, již není nutné řešit, zda je slave mikrokontroler zatížen či nezatížen. Z toho důvodu pro toto a další měření již mikrokontroler nebude zatěžován komunikací přes I2C či matematickými úlohami.

Jako první, je třeba zjistit, jak bude vypadat výsledný signál, po průchodu prodlužovacího vedení. K tomu byl využit osciloskop SDS1102X+, na kterém byl zobrazen výsledný průběh signálu po průchodu prodlužovacím vedením a průběh signálu, před průchodem daného vedení. To je možné vidět na obrázku 7.4. Zde můžeme pozorovat, že nejprve dojde k okamžitému skoku napětí na úroveň 0,6 V a až následně dojde k zvýšení napětí na očekávanou velikost.

Tento jev lze patrně vysvětlit pomocí telegrafních rovnic, kdy lze každý vodič rozdělit na nekonečné množství malých kondenzátorů, které se při vyslání signálu nabíjí z prvotní velikosti přivedené energie. K tomuto jevu dojde okamžitě po spojení, čímž je na výstupu přivedena část napětí, kdy velikost napětí je rovna velikosti vstupní energie podělená délkou vodiče. K nárustu napětí na hodnotu vstupního napětí poté dojde v moment, co na konec kabelu připutuje šířící se vlna.



Obrázek 7.4 Průběh signálu před a po projití 700 metrovým kabelem (modrý průběh zobrazuje signál před průchodem, zelený po průchodu)

Když již máme znalost o tvaru synchronizačního signálu, je třeba zjistit, zda tento signál je synchronizovaný mikrokontroler schopen detekovat jako externí přerušení. Proto byl vytvořen kód, který počítal počet zachycených přerušení, po dobu zhruba 30 minut. Přičemž po tuto dobu nadřazený mikrokontroler, vyvolával tyto přerušení pomocí jednoho ze svých výstupů každou sekundu, zatím co na podřazeném mikrokontroleru, docházelo k zachytávání těchto synchronizačních signálů a jejich počítání.

Po uplynutí stanovené doby, bylo odesláno 1821 synchronizačních pulzů a stejné množství jich bylo zachyceno. Z toho lze usoudit, že takto vypadající signál nebyl pro mikrokontroler problém detekovat.

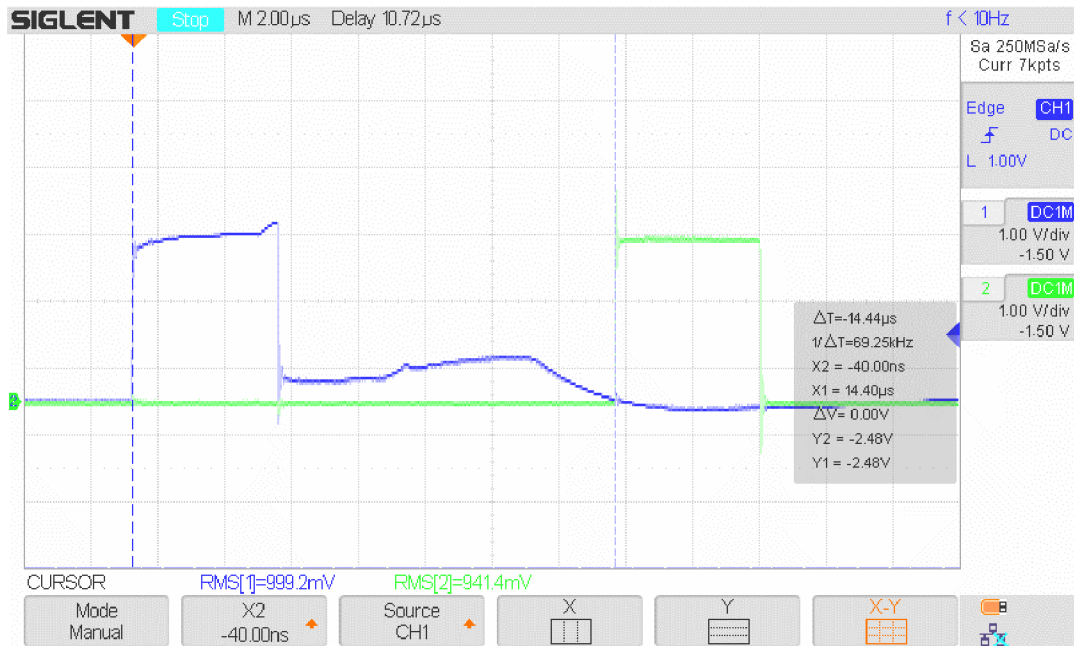
Z toho důvodu jako další krok, bylo třeba proměřit vznikající zpoždění, při propojení dvou mikrokontrolerů, přes prodlužovací vedení. Pro získání základní znalosti o tomto zpoždění, bylo nejprve toto měření provedeno pomocí osciloskopu, kdy zachycení průběh signálu je možné vidět na obrázku 7.5, kdy nejprve je vidět zachycení synchronizačního pulzu a následně po uplynutí 14,4 μs je zachycena i reakce slave mikrokontroleru, vyvoláním signalizačního pulzu pro určení doby zpoždění.

Následně na to, došlo k proměření více hodnot pomocí univerzálního čítače Agilent 53131A. Přičemž, tedy došlo k úpravě původního schématu zapojení 7.1, o přidání dlouhého kabelu do sekce označené písmenem B.

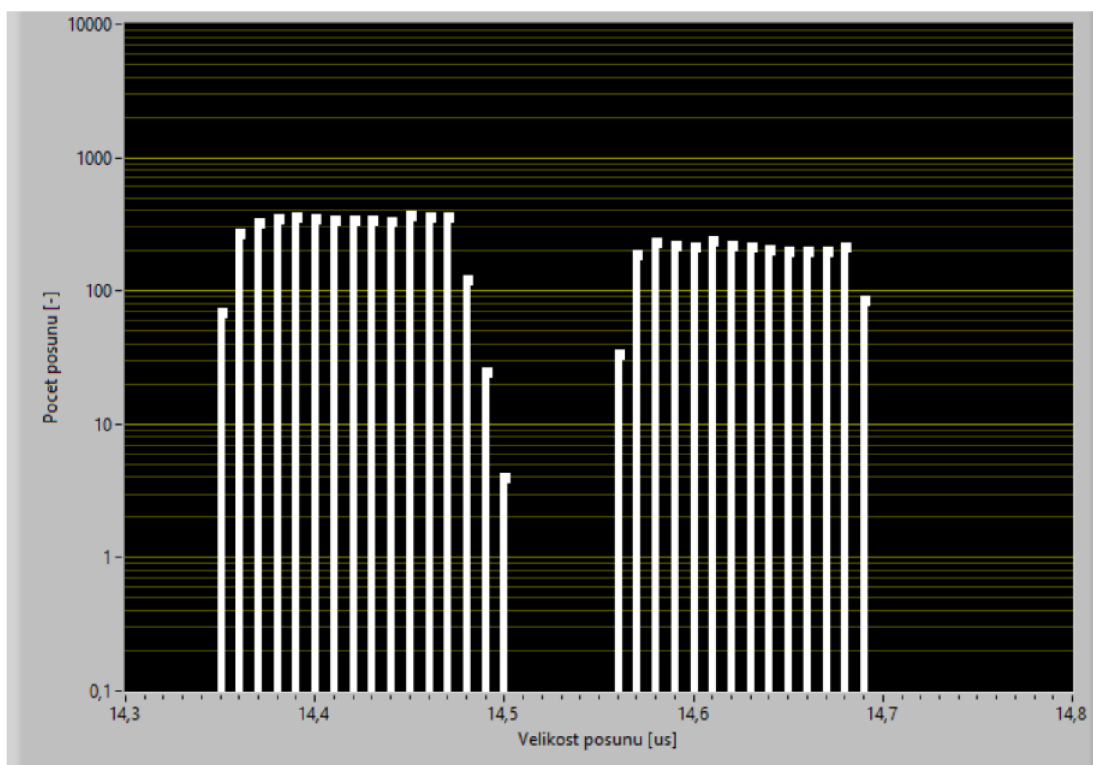
Na základě, naměřených hodnot byl poté vytvořen histogram, který je možné pozorovat na obrázku 7.6. Jak je patrné, tak časové posuny jsou rozděleny na dva rozsahy, kdy první je od 14,35 do 14,5 μs a druhý je od 14,56 do 14,69 μs . Což dává posun těchto dvou shluků zhruba o 0,2 μs . Důvod toho posunu, je neznámý, neboť nelze jednoduše

řící, že se jedná o posun způsobený frekvencí mikrokontrolerů, neboť jejich frekvence je nastavena na 8 MHz, tedy perioda je rovna 125 ns.

Střední hodnota je tedy 14,4497 μs , z čehož plyne, že přidáním prodlužovacího vedení, se zvýšila hodnota zpoždění zhruba o 3,85 μs . Pokud vezme rychlost šíření vlny v mědi, která je zhruba $2 \text{ E}+8 \text{ m/s}$ a délku vodiče v našem případě 700 m, tak je možné vypočítat čas, o který by měl být signál reálně zpožděn. V tomto případě, by mělo být zpoždění o velikosti 3,5 μs .



Obrázek 7.5 Posun na dlouhém kabelu (modrý průběh zobrazuje signál před průchodem, zelený po průchodu)



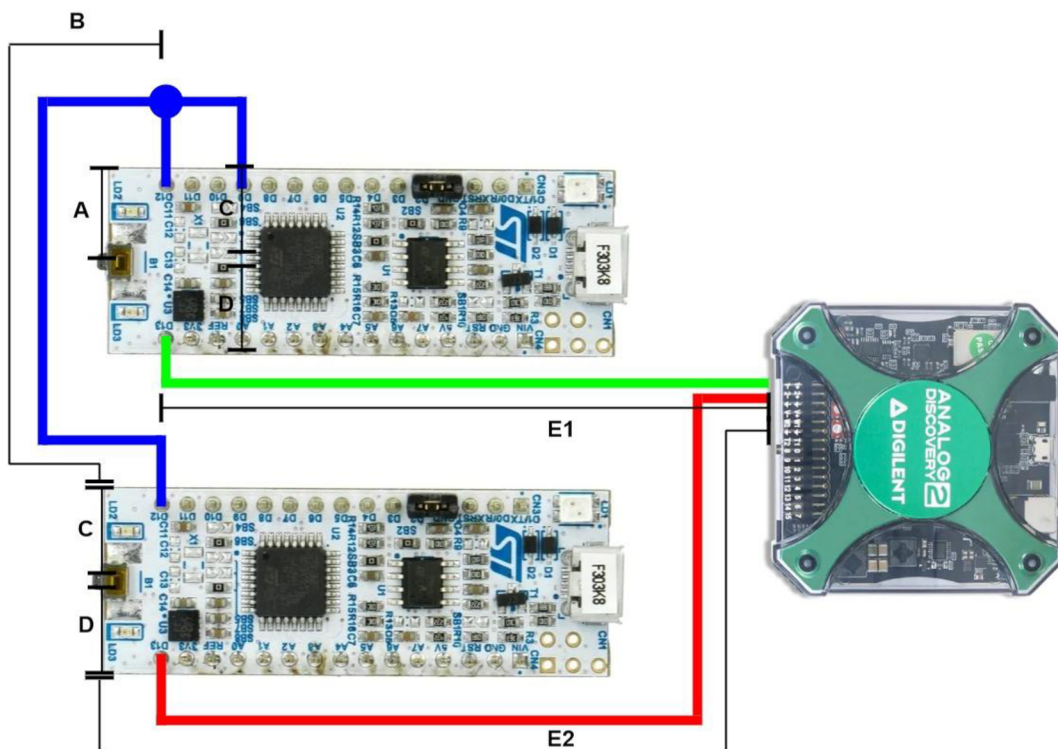
Obrázek 7.6 Histogram posunu přijmutí synchronizačního pulsu, při prodloužení spojení mezi mikrokontrolery

7.1.4 Zpětná smyčka pro master mikrokontroler

Jak bylo zmíněno v kapitole 7.1, tak se dá zpoždění, rozdělit na několik částí. Tato kapitola se blíže podívá na zpoždění označené písmeny C a D. Kdy tato zpoždění, vznikají na slave mikrokontroleru, při zpracování synchronizačního pulzu a při jeho prodloužení do měřicího systému.

Pro potlačení, těchto zpoždění je tedy potřeba přidat zpětnou smyčku do master mikrokontroleru, kdy tento mikrokontroler si bude sám sobě posílat synchronizační pulz a na základě toho, bude vystavovat i nový signalizační pulz pro měřicí systém. Způsob zapojení je zobrazen na obrázku 7.7. Touto metodou, nedojde k úplnému potlačení náhodnosti v zachycení signálu, ale při nejmenší dojde k jeho částečnému potlačení. Neboť master mikrokontroler, bude muset vykonat stejnou činnost jako synchronizovaný mikrokontroler.

Při eliminaci, zmíněných zpoždění, tedy zbývá pouze zpoždění na kabelu (B). Z toho důvodu, nemá cenu měřit, tuto metodu na krátkém vodiči ale je vhodné měřit zpoždění na prodlužovacím vedení. Výsledný histogram těchto signálů je možné vidět na obrázku 7.8. Přičemž je na něm vidět, že přidáním zpětné smyčky, opravdu dojde k snížení rozdílu časového posunu mezi dvěma mikrokontrolery.

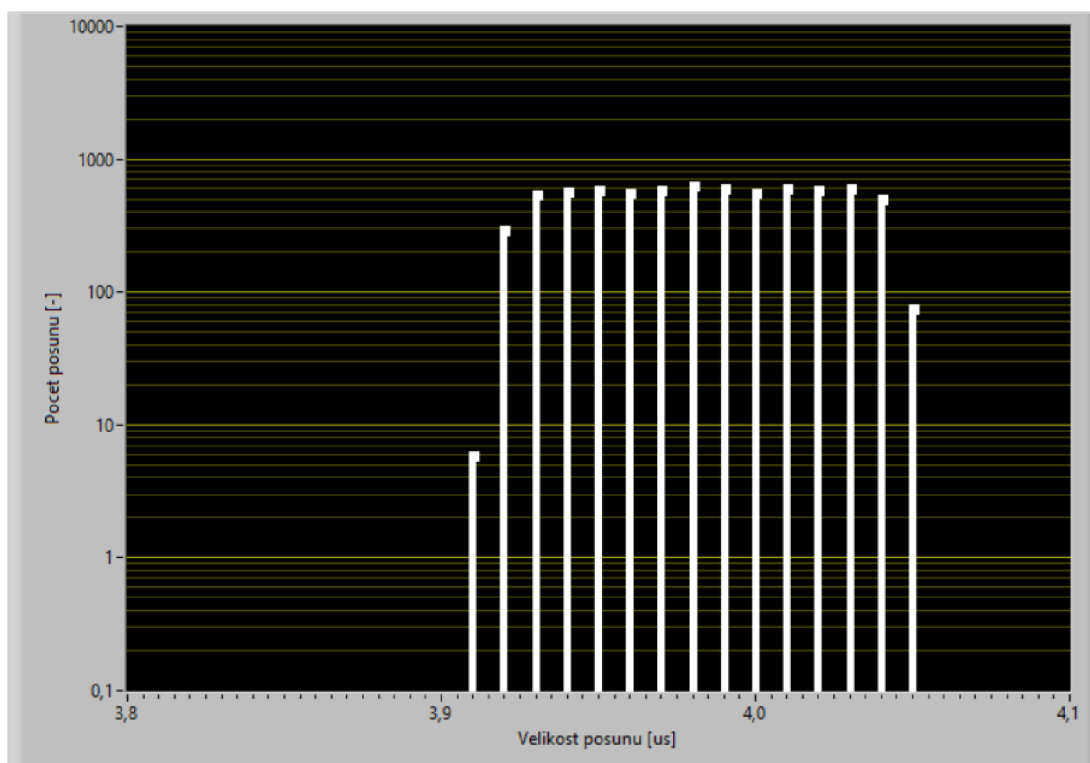


Obrázek 7.7 Schéma zapojení dané metody s měřením zpoždění synchronizace, při přivedení zpětné smyčky do hlavního mikrokontroleru [16] [17]

Pro ověření, zda se opravdu jedná pouze o součet zpoždění B a rozptylů zpoždění C a D, je třeba odečíst střední hodnotu měření bez prodlužovacího vedení od měření zpoždění s prodlužovacím vedením.

Střední hodnota se zpětnou smyčkou vyšla $3,98 \mu\text{s}$. Když toto porovnáme s rozdílem středních hodnot bez prodlužovacího vedení a s prodlužovacím vedením, tak zjistíme že rozdíl v těchto hodnotách je $3,85 \mu\text{s}$. Což značí, že výsledný zpoždění je tedy o něco vyšší, než je vypočítané. To může být způsobeno zmíněnými rozptyly zpoždění na částech označenými písmeny C a D, popřípadě neočekávaným výsledkem histogramu, z minulé kapitoly, kde bylo měřeno zpoždění s přidáním prodlužovacího vedení.

Je také třeba zmínit zpoždění A, které není měřeno, přesto je možné, že pomocí této metody dochází k jeho eliminaci. Pro jeho úplnou eliminaci, musí platit, že doba mezi vystavením signálu na výstupu hlavního mikrokontroleru a jeho zachycení na jiném vstupu hlavního mikrokontroleru, je delší než doba potřebná k dokončení instrukce vystavení signálu na výstupu. Jednodušeji řečeno, je třeba, aby při zachycení synchronizačního signálu na vstupu hlavním mikrokontroleru, již nebyl tento mikrokontroler zatížen zpracováváním instrukcí týkající se vystavení synchronizačního pulzu na výstupu.



Obrázek 7.8 Histogram posunu přijmutí synchronizačního pulsu, při zavedení zpětné smyčky

7.1.5 Eliminace zpoždění na kabelu

Jako poslední možností pro snížení časového posunu mezi mikrokontrolery je eliminace zpoždění na kabelu. Jak již bylo zmíněno dříve, tak výslednou rychlost šíření signálu ve vodiči, lze vypočítat ze znalosti rychlosti šíření vlny v daném vodiči a délkou daného vodiče. Čímž tedy získáme znalost o době zpoždění synchronizačního signálu na vodiči.

Poté lze již zvolit mezi dvěma možnostmi eliminace toho zpoždění, přičemž pokud je okolní teplota stabilní, lze toto zpoždění eliminovat softwarově, kdy hlavní mikrokontroler, by posílal synchronizační zprávu o daný čas dříve, aby příchod synchronizačního pulzu, k slave mikrokontroleru byl v požadovaný čas. Zde je potřeba pouze upravit, vyhodnocení master mikrokontroleru při příjmu signálu ze zpětné smyčky. Kdy je třeba daný čas opět, k tomuto signálu přičíst, popřípadě pozdržet vykonání reakce na toto přerušení, pomocí čekací smyčky nebo zpoždění.

Další možností by bylo hardwarové zpoždění signálu, kdy tato metoda má výhodu v automatickém přizpůsobení rychlosti signálu na základě okolní teploty. Zde by bylo nutné, zpětnou smyčku vést přes parametrově stejný vodič, popřípadě vytvořit zpětnou smyčku z kombinace kondenzátorů, rezistorů a cívek, které vytvoří požadované zpoždění na napětí.

7.2 Zhodnocení metody

Na závěr této kapitoly, je potřeba říct, že pro tuto metodu nemá smysl, měřit Allanovu odchylku pouze pro slave mikrokontroler, neboť by při dokonalé synchronizaci by byla měřena Allanova odchylka mastera mikrokontroleru. Proto pro smysluplné měření této odchylky, by bylo nutné měřit tuto odchylku na obou mikrokontrolerech najednou. Přičemž by nás nezajímala velikost odchylky ale pouze odlišnost těchto dvou Allanových odchylek od sebe.

Na základě poznatku z kapitol 7.1.1 a 7.1.2, lze říct, že výsledná synchronizace není závislá na zatížení mikrokontroleru, a tím pádem již v dalších možnostech synchronizace, nebude nutné se touto problematikou zabývat.

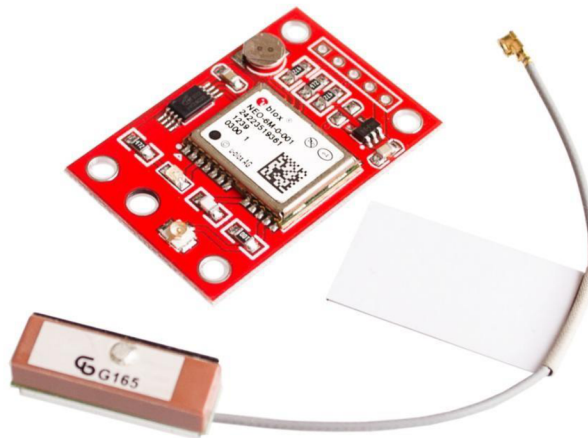
Na základě, hodnot z tabulky 7.1, je vidět, že časový posun mezi mikrokontrolery je závislý na délce vedení mezi mikrokontrolery ale i na rychlosti zareagování na přijetí externí přerušování. Přičemž, zpoždění způsobené reakcí mikrokontroleru, lze eliminovat pomocí zpětné smyčky. Zpoždění na vedení, lze eliminovat, na základě přidání zpoždění do systému, jak softwarovém, tak hardwarovém. Kdy pro určení velikosti zpoždění, lze využít matematické vzorce, využívající rychlost šíření vlny v daném vedení a délky vedení, popřípadě proměřením zpoždění pomocí přímého měření.

Tabulka 7.1 Doby zpoždění mezi příjmem synchronizačními signály

Metoda	Střední hodnota [μ s]	Směrodatná odchylka [μ s]
Bez vytížení MCU	10,6045	0,0619
I2C MCU	10,5996	0,0626
Matematika MCU	10,5967	0,0666
Prodloužený vedení	14,4497	0,1073
Zpětná smyčka	3,9830	0,0362

8. SYNCHRONIZACE POMOCÍ GPS

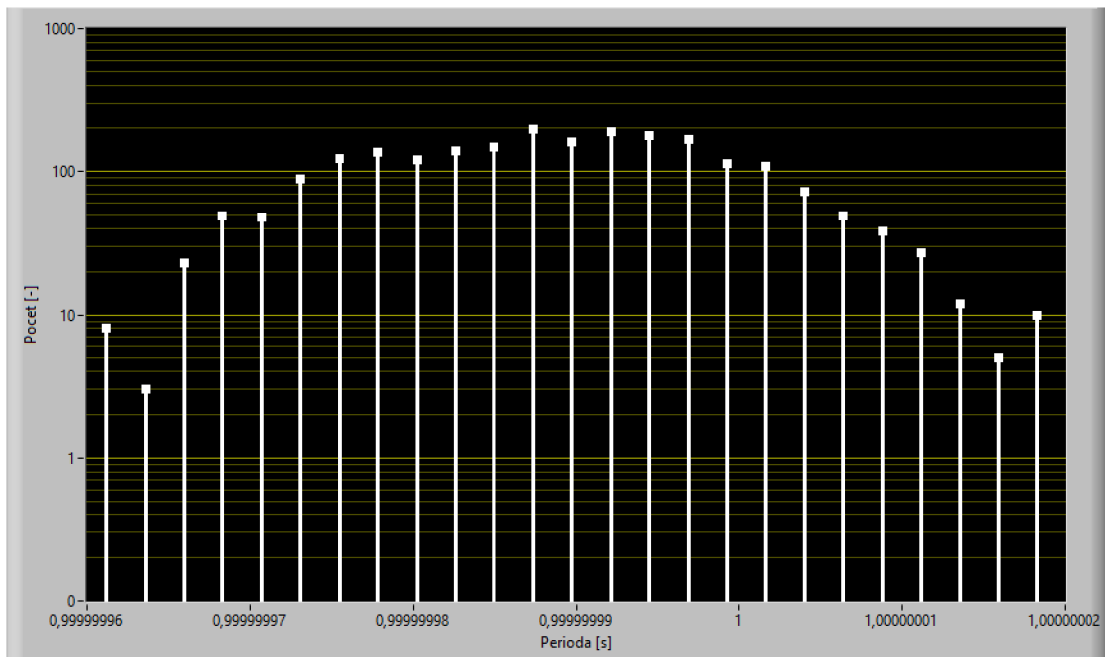
Pro příjem GPS signálu, byly pořízeny GPS přijímače NEO-6M, kdy tyto přijímače jsou schopny ze zachyceného signálu GPS, vytvářet 1 PPS signál, který byl následně využíván k synchronizaci mikrokontrolerů. Také jsou tyto přijímače schopny, posílat GPS zprávy přes UART komunikaci, kdy z těchto zpráv lze vyčíst polohu, či přesný UTC čas. Čehož, lze využít například pro zjištění, zda nedošlo k výpadku GPS příjmu na přijímači nebo pro udržování přesného času.



Obrázek 8.1 GPS přijímač NEO-6M [18]

8.1 Perioda PPS signálu

Nejprve je třeba zjistit, jak velmi přesné jsou tyto GPS přijímače a jak stabilní periodu mají na výstupu 1 PPS signálu. K proměření periody PPS signálu, byl využit Agilent 53131A, který měřil časový posun dvou po sobě jdoucích pulzů. Výsledek tohoto měření je možný pozorovat na obrázku 8.2, kde je zobrazen histogram naměřených časů. Kdy, jak je vidět chyba v periodě toho signálu se pohybuje v řádu desítek ns. Z čehož lze vyvodit, že stabilita tohoto signálu je velmi přesná. Pokud na výsledný histogram hodnot použijeme statistické operace zjistíme, že střední hodnota je rovna 999999,989 μ s, přičemž směrodatná odchylka je 0,011 μ s. Přičemž nejistota měřícího přístroje zde byla určena jako $\pm 1,5$ ns. Kdy k jejímu určení byl použit vzorec ze strany 10 v datasheetu výrobce [19].

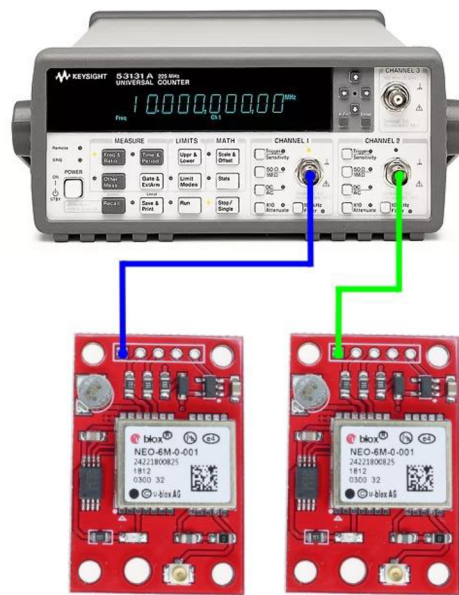


Obrázek 8.2 Histogram periody PPS signálu pro GPS

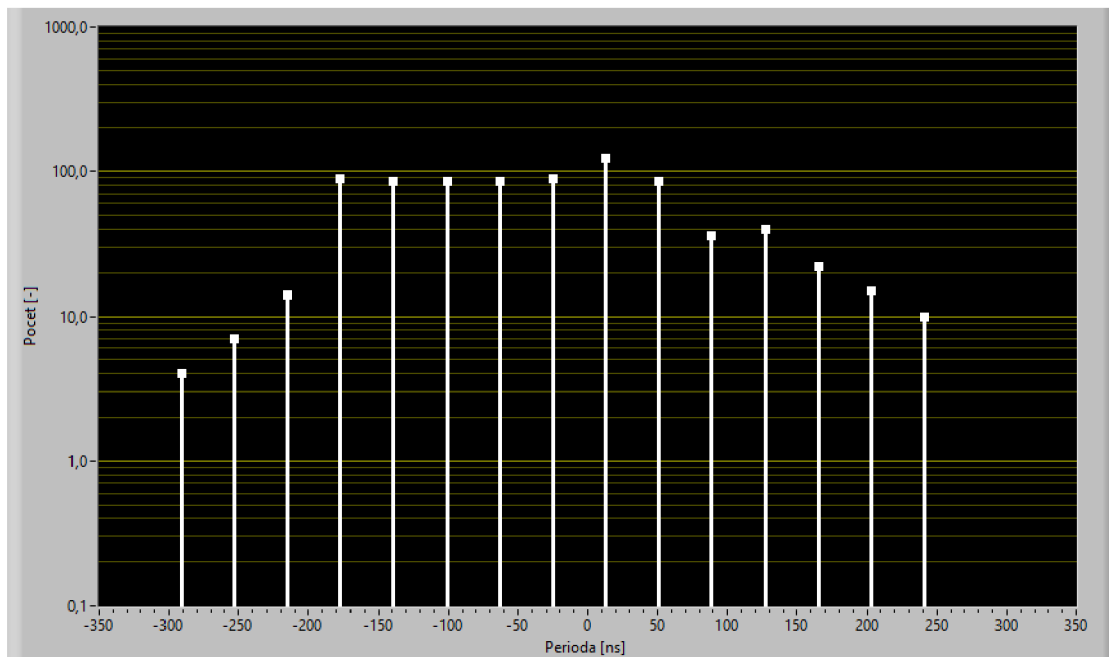
8.2 Posun signálu mezi přijímači GPS

Jako další je třeba zjistit, jak velmi přesné jsou tyto GPS přijímače a jak stabilní periodu mají na výstupu 1 PPS signálu. Proto jako další měření, bylo provedeno měření posunu signálu na výstupech těchto přijímačů.

Schéma zapojení je možné pozorovat na obrázku 8.3. Kdy opět pro zajištění minimální chyby vznikající na přívodních vodičích vedoucích do měřícího přístroje, jsou použity parametrově stejné vodiče. Výsledný histogram naměřených dat, je možné pozorovat na obrázku 8.4. Zde je dobré zmínit, že oproti předcházející metodě, kdy hlavní mikrokontroler, byl oproti synchronizovanému mikrokontroleru vždy časově napřed, zde již toto nebude neplatit. Tedy již nelze říct, že jeden z nich bude vždy opožděn oproti druhému, ale bude záviset na stabilitě přijímačů.



Obrázek 8.3 Schéma zapojení pro měření posunu na přijímačích GPS [21] [21]



Obrázek 8.4 Histogram posunu signálu vzniklých na přijímačích

Z výsledků, je patrné, že se zpoždění nyní pohybujeme ve stovkách nanosekund. Kdy střední hodnota je rovna -31,9 ns, se směrodatnou odchylkou 107,9 ns. Dále je potřeba říci, že zobrazená data byla měřena na čítači Agilent 53131A, který je schopen měřit pouze zpoždění signálu v určeném pořadí. Tedy nejdříve musí přijít signál na vstup 1 a až následně na vstup číslo 2. Proto pokud nejdříve byl přiveden signál na vstup 2 a až

následně na vstup jedna, byla naměřena hodnota blízká jedné. Neboť již druhý přijímač signalizoval příchod další sekundy. Proto od těchto hodnot byla následně odečtena střední hodnota periody signálu, kdy toto počínání může zanechat chybu do výsledných dat.

8.3 Posun signálu na výstupech MCU

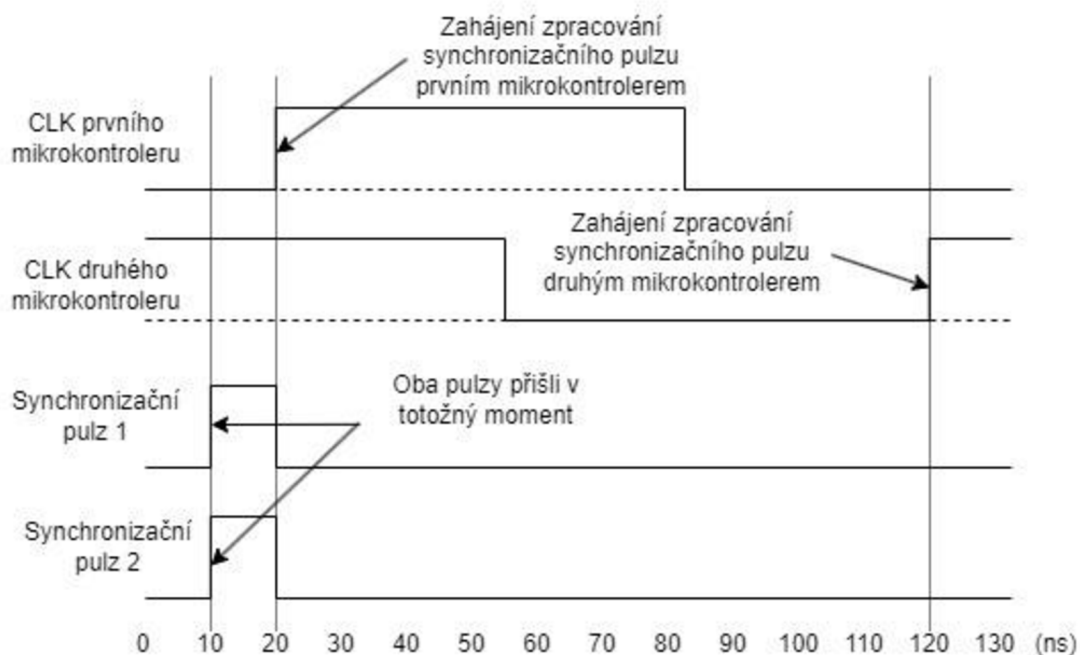
Nyní když již známe, reálnou hodnotu posunu, na přijímačích GPS, je třeba změřit i výslednou hodnotu posunu, na mikrokontrolerech. Stejně tak jak u minulé metody, jsou hodiny mikrokontrolerů nastaveny na frekvenci 8 MHz, tedy periodu 125 ns. Tedy perioda hodin je nastavena na podobnou dobu jako je chyba v posunu přijímačů. Z toho lze usoudit, že výsledná chyba posunu bude zásadně závislá na tom, v jaké chvíli bude přiveden signál na vstup mikrokontroleru.

Pokud bychom si to představili na příkladu, kdy signál z přijímačů dorazí k oběma mikrokontrolerům v stejný okamžik, ale jeden z mikrokontrolerů, provede svůj tik až za dobu 110 ns, zatímco druhý jej provede za 10 ns, od příjmu těchto signálů, tak tento druhý mikrokontroler získá náskok zmíněných 100 ns, i když byl signál přiveden na oba mikrokontrolery v totožný moment. Z toho lze usoudit, že výsledné zpoždění zde bude zásadně závislé na vnitřní frekvenci mikrokontrolerů. Tento případ je zobrazen na obrázku 8.5.

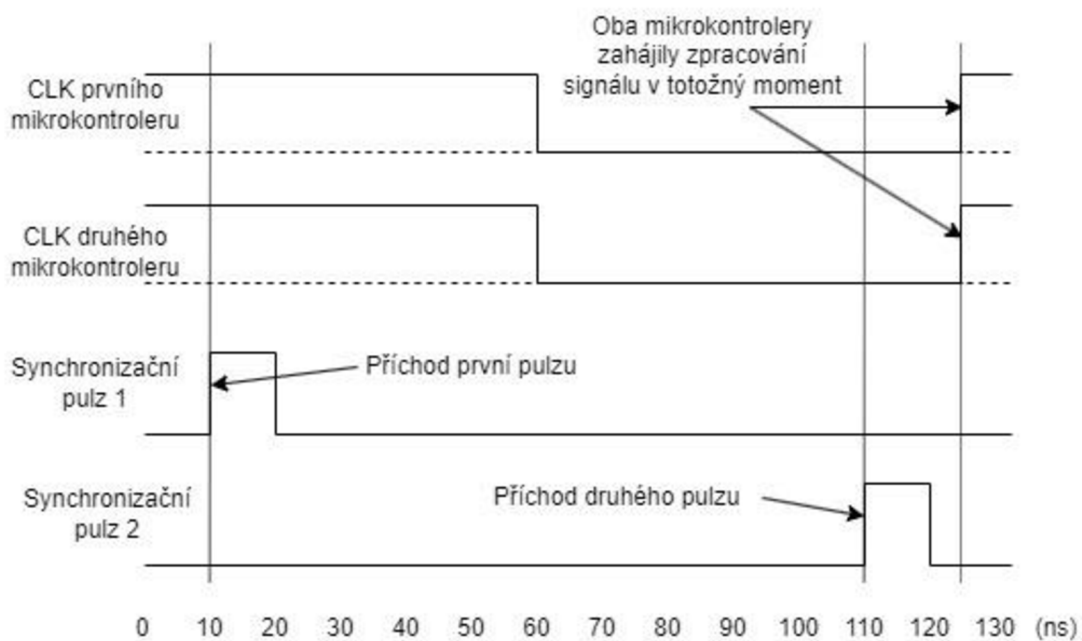
Je třeba uvažovat, že může vzniknout však i opačný případ, kdy se může tento posun zmenšit. Pokud budeme opět uvažovat příklad, kdy tentokrát signály z přijímačů nepřijdou ve stejný moment, ale naopak budou posunuty, například o těch zmíněných 100 ns. A za tu dobu nebude ani na jednom mikrokontroleru proveden hodinový pulz, tak dojde k snížení výsledné chyby posunu. V tomto případě, na rozmezí hodnot 0 až 25 ns. Při takovém scénáři, mají výhodu nižší frekvence vnitřních hodin, kdy nižší frekvence jsou tím pádem schopny, utlumovat vyšší časové posuny vzniklých na přijímačích.

Pokud budeme uvažovat pouze kladné hodnoty z minulé kapitoly, je vidět, že nemá cenu uvažovat o nižších frekvencích, než je 2 MHz, neboť nejvyšší hodnota posunu byla 500 ns. Takto probíhající eliminace zpoždění je možné vidět na obrázku 8.6.

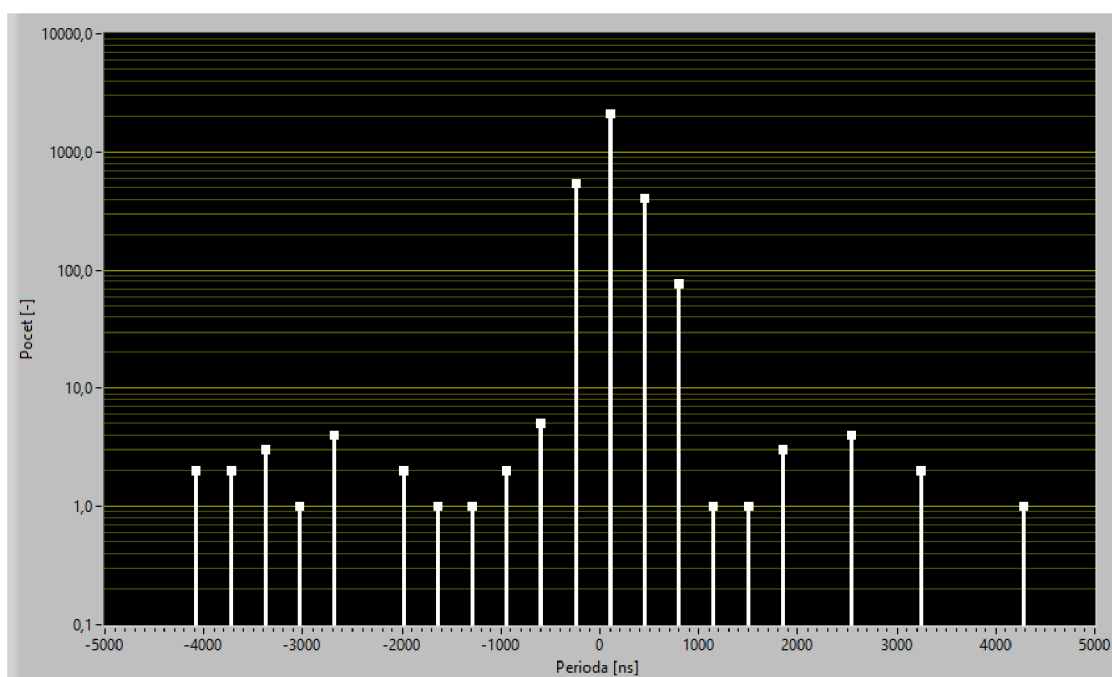
Výsledný histogram naměřených posunů na mikrokontrolerech, je možné pozorovat na obrázku 8.7. Kdy opět jako v minulé kapitole, je zde zanesena chyba s odečtem střední hodnoty od záporných dat. Jak je vidět většina posunů, je v rozmezí 0 až 1 μ s, pokud budeme uvažovat pouze neupravena data (tedy ta kladná), přičemž zhruba 2/3 jsou v rozmezí 0 až 500 ns, což odpovídá chybě čtyř tiků mikrokontroleru.



Obrázek 8.5 Příklad vytvoření zpoždění na základě fáze hodin mikrokontroleru



Obrázek 8.6 Příklad eliminace zpoždění na základě fáze hodin mikrokontroleru



Obrázek 8.7 Histogram posunu signálu při připojení mikrokontroleru

8.4 Allanova odchylka při synchronizaci pomocí GPS

Nyní, když již známe časový posun, dvou mikrokontrolerů, a známe jednu část potřebnou pro synchronizaci, kterou je sjednocení začátku každé synchronizace. Je třeba nyní vytvořit metodu, která zabrání/zpomalí rozcházení synchronizace, na základě odlišnosti vnitřních hodin. Tedy je třeba, aby došlo k sjednocení frekvence kmitů v obou mikrokontrolerech. Toho lze dosáhnout přidáním PS regulátoru, který bude řídit frekvence v mikrokontroleru za běhu, na základě zpětné vazby. Tato zpětná vazba bude v této metodě založena na základě PPS signálu, který je posílán z přijímače GPS.

Nejprve se podíváme na jednotlivé části regulační smyčky. Ta se skládá z požadované hodnoty, kdy v tomto případě je požadována hodnota rovná frekvenci hodin v mikrokontroleru. Ta byla zvolena 8 MHz, pro zachování stejné frekvence při měření posunů, tak při měření Allanovy odchylky.

Jako další je regulátor, který je vytvořen pomocí instrukcí v mikrokontroleru. Přičemž jeho výstupem je akční zásah, který je přiváděn do soustavy, čímž je v tomto případě nastavení vnitřních hodin. Více o tomto nastavení je v kapitole 8.4.1.

Poté již zde máme výslednou zpětnou vazbu, kdy jako její snímač je použit vnitřní časovač, který počítá počet tiků, které byly provedeny v době mezi přivedením dvou po sobě jdoucích synchronizačních pulzů. V našem případě, by tedy měla být v ideálním případě vrácena hodnota zvolené frekvence, tedy osm milionů.

8.4.1 Nastavování vnitřních hodin

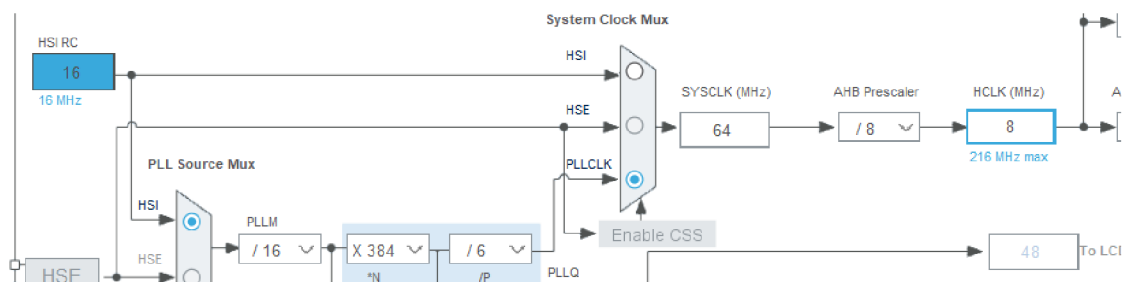
Pro možnost nastavování různých frekvencí vnitřních hodin, je umístěn v STM32 fázový závěs neboli PLL, kde je možné nastavovat různé děličky a násobičky vnitřního oscilátoru, za účelem vzniku jiných frekvencí. Jeho možnosti nastavení jsou zobrazeny na obrázku 8.8. Kdy pro změnu frekvence, lze využít násobička PLLN a děličky PLLM, PLLP a AHB. Zde je třeba podotknout, že nelze pomocí těchto násobiček a děliček vytvořit jakoukoliv frekvenci, ale lze se pomocí nich této frekvenci přiblížit. To je způsobeno nutností, volit celočíselné hodnoty, někdy i stoupajících pomocí bitového posunu, myšleno jako 1, 2, 4, 8 atd. Popřípadě, vnitřními omezeními jako je například na děličce PLLM, kdy po podělení vstupní frekvence musí být výsledná frekvence v rozmezí 0,95 až 2,1 MHz. Proto bylo třeba, pomocí Matlabu vypočítat jednotlivé možnosti nastavení jednotlivých parametrů PLL, aby došlo k získání co nejjemnějšího kroku.

Poté lze zvolit jednu ze dvou metod, kdy jedna spočívá ve volbě pouze jednoho parametru, který se bude měnit. V tomto případě, to byla násobička PLLN, která je schopna nastavovat nejvíce možných hodnot v okolí 8 MHz s nejmenším krokem. Přičemž ostatní děličky budou nastaveny pevně, čímž nám vznikne možnost snadno, dopočítávat hodnotu násobičky na základě numerického výpočtu, přímo za běhu mikrokontroleru. Další výhodou této metody je v konstantní kroku, kdy tento krok lze vypočítat jako $16\,000\,000 / \text{PLLM} / \text{PLLP} / \text{AHB}$, což v tomto případě vychází na 20,833 kHz.

Druhou možností je měnění všech parametrů PLL, kdy zde již není výhodné vypočítávat jednotlivé hodnoty frekvence pokaždé, kdy je třeba volit nové rozložení PLL, ale je třeba vytvořit look-up tabulku, kde se tyto frekvence uloží spolu s jednotlivými nastaveními pro jednotlivé komponenty PLL. Výhoda této metody je v možnosti jemnějšího měni frekvence s krokem v rozmezí 3 až 10 kHz, kdy však nejsou tyto hodnoty měněny o konstantní hodnotu. Nevýhoda této metody, je v nutnosti vytvoření dané tabulky, která zabere část paměti v mikrokontroleru.

Jak je patrné, ze zmíněných kroků, výsledná hodnota není spojitá ale digitální, proto byl do soustavy přidán integrační článek, který udržuje informaci o správné spojitě hodnotě, čímž zaniká potřeba vytvářet složitý regulátor, ale je možné ho uvažovat pouze jako diskretní regulátor nikoliv digitální.

Pro převod spojitě hodnoty na digitální, již poté stačí pouze nalézt nejbližší hodnotu k spojitě frekvenci v look-up tabulce a tu nastavit do příslušných registrů pro nastavení PLL. Další výhoda vznikající přidáním integračního článku, je v útlumu šumu na výstupu oscilátoru, kdy občas i bez ze změny parametrů na oscilátoru dochází ke změně reálné výstupní frekvence hodin. Tento jev je možné pozorovat na obrázku 12.1. V tomto případě, není třeba, aby regulátor reagoval silně na změnu přenastavení hodin, ale může reagovat mírněji, neboť jeho zásah je vždy zapsán do integračního článku.



Obrázek 8.8 Ukázka nastavení hodin v STM32

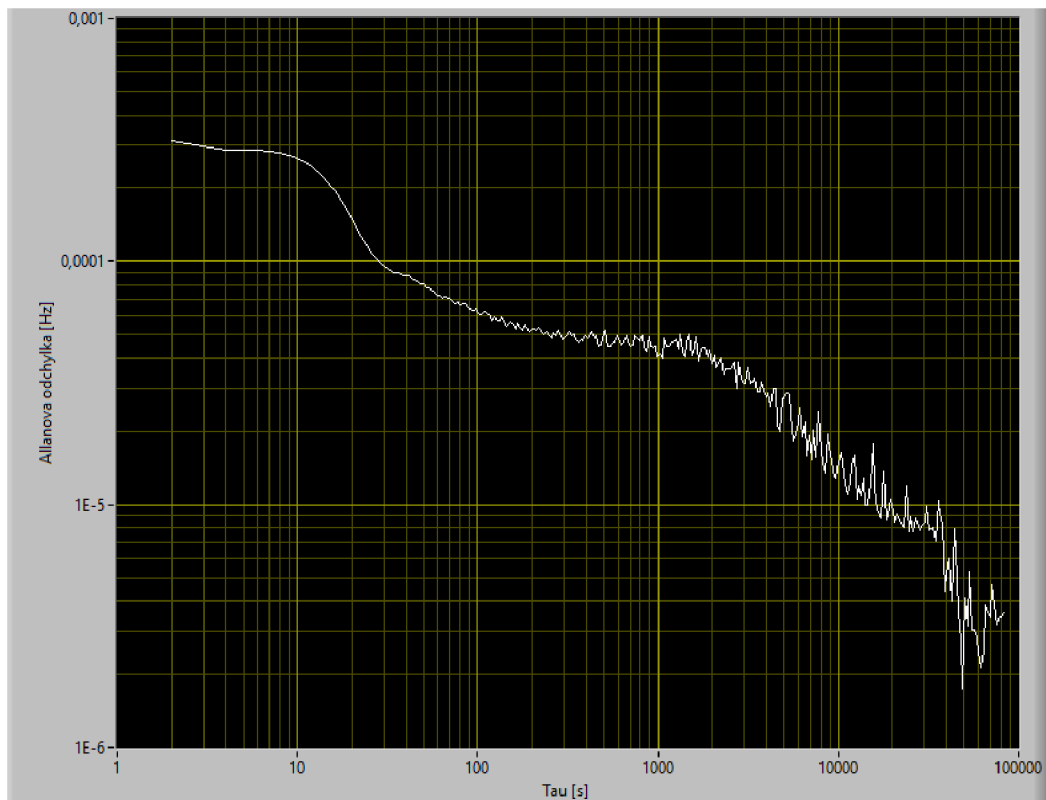
8.4.2 PS regulátor

Jak bylo zmíněno v předcházející kapitole, tak nastavení PS regulátoru, není vhodné kvůli, šumu ze zpětné vazby, nastavovat tak, aby okamžitě reagoval na vysoké změny, nějakou vysokou reakcí. Místo toho je lepší, aby pomalu postupně upravoval spojitou hodnotu v integračním článku, čímž nám vznikne takový jednoduchý filtr na rychlé změny.

Jako zpětná vazba byl využit časovač, který počítal kmity hodin mezi příchody 1 PPS signálu. Z čehož je patrné, že pokud ustane příjem GPS signálu, není možné získávat zpětnou vazbu. Proto při výpadku signálu, dojde k přepnutí regulační smyčky na náhradní model, kdy jeho popis je v kapitole 11.

8.4.3 Měření Allanovy odchylky

Po vytvoření regulační smyčky bylo nutné, proměřit její přesnost časové synchronizace pomocí Allanovy odchylky. Výsledný graf odchylek je možné pozorovat na obrázku 8.9. Jak je vidět, tak na začátku signálu je vidět val, zhruba mezi 2 až 30 sekundou, ten je patrně způsoben přechodovým dějem, při regulaci. Následně dochází k zpomalení poklesu Allanovy odchylky, a to až do času 1500 sekund, kdy opět dochází k rychlejšímu poklesu. Proč k tomu zpomalení/zrychlení došlo se nepodařilo zjistit, ale je možné že zde došlo k drobnému výpadku signálu GPS, čímž by mohl být ovlivněna výsledná podoba Allanovy odchylky. Výsledná stabilita se poté pravděpodobně zastavila na hodnotě okolo 3,5 μHz .



Obrázek 8.9 Allanova odchylka pro STM32 synchronizované pomocí metody využívající GPS

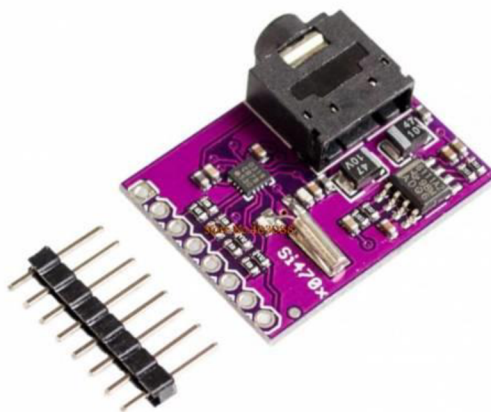
8.5 Zhodnocení metody GPS

Jak je z výsledků patrné, tak tato metoda má oproti předcházející metodě, výrazně menší časový posun mezi dvěma mikrokontrolery, kdy výsledný posun je o dva řády nižší. Jak ukázala následná metoda, s připojenými mikrokontrolery, nemá velký smysl, hledat či se snažit více tuto hodnotu snížit, neboť hlavní zdroj chyby v posunu, je na samotných mikrokontrolerech. Zde by byla možnost zvýšit jejich frekvenci kmitů, čímž by mohlo dojít k snížení dané chyby.

Nevýhoda této metody, je oproti minulé v tom, že je potřeba umístit GPS přijímač, do takového místa, aby byl schopen zachytit vysílání od satelitů. Pokud tedy, dojde k výpadku signálu, stane se tato metoda nefunkční.

9. SYNCHRONIZACE POMOCÍ RDS DAT Z FM RÁDIÍ

Pro příjem FM radia byl pořízený rádiový přijímač SI4704, který je schopen přijímat i postranní pásmo s RDS daty. Kdy při dokončení jejich zpracování, tedy přijmutí všech 104 bitů, vyvolá na jednom ze svých výstupu přerušení, které je signalizováno pomocí sestupné hrany. K tomuto přerušení by mělo docházet dle teorie každých 87,576 ms. V průběhu implementování a testování této metody byla používána FM stanice Evropa 2, která právě toto postranní pásmo používá k přenosu dodatečných informací o svém vysílání.



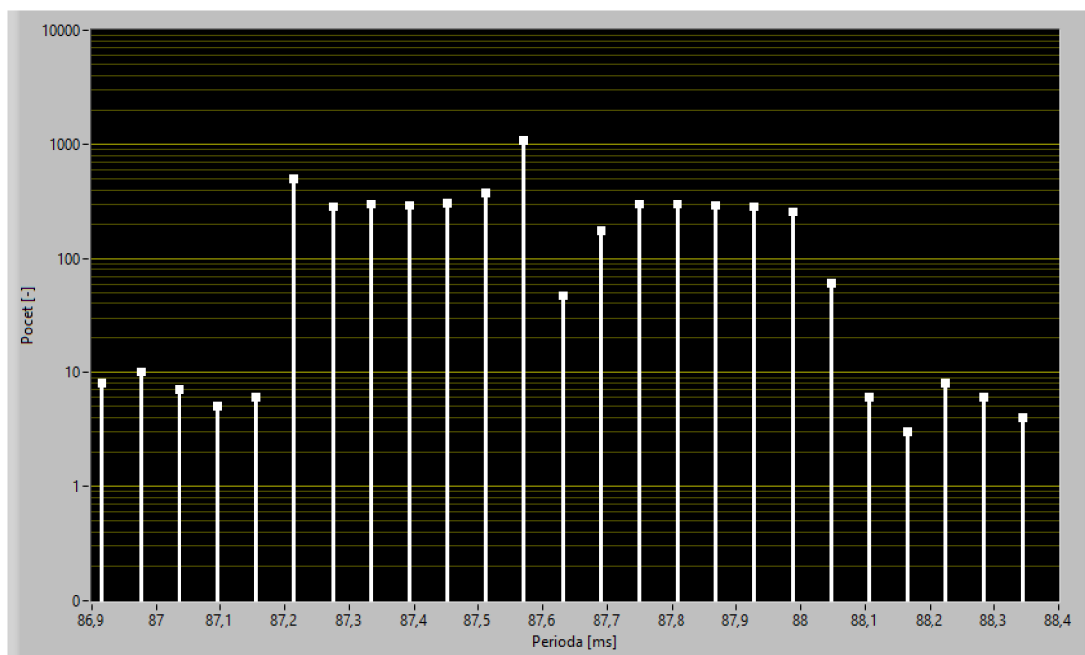
Obrázek 9.1 Rádiový přijímač SI4703 [22][22]

9.1 Měření doby mezi po sobě jdoucími pulzy

Jako první bylo třeba proměřit přesnost doby mezi po sobě jdoucími pulzy, kdy z teorie vychází, že by tato doba měla být rovna 87,576 ms. K proměření této doby byl využit čítač Agilent 53131A, který byl nastaven tak, aby měřil dobu periody sestupných hran na samotném přijímači.

Na obrázku 9.2, je možné pozorovat histogram naměřených dat kdy, jak je vidět nejvíce naměřených hodnot je v bodě, který je roven teoretické hodnotě. Dále je vidět, že i body v okolí zhruba $\pm 400 \mu\text{s}$, jsou zastoupeny v hojném počtu. Střední hodnota je rovna 87,579 ms, kdy směrodatná odchylka je 0,239 ms.

Protože by z praktické hlediska nemělo smysl provádět takto často synchronizaci mikrokontroleru, neboť by mikrokontroler nedělal nic jiného než právě synchronizaci, bylo by lepší využívat více těchto přerušení společně, k získání přesnějších výsledků. To nás tedy vede k tomu, že by bylo vhodné si ukládat průběžné doby příchodu jednotlivých pulzů, a následně je využít při příchodu určitého počtu.



Obrázek 9.2 Histogram naměřených period mezi jednotlivých pulzy

Výhodou tohoto provedení, by mělo být zmenšení rozptylu, neboť tím, že dojde ke sečení po sobě jdoucích hodnot, by mělo dojít k zprůměrování jednotlivých dob. Tuto skutečnost by měl umocnit i fakt, že rozptyl daných dob není zcela náhodný, ale je závislý na dobách předcházejících. Tato závislost se projevuje na tom, že pokud první doba byla kratší např. o hodnotu 0,2 ms, další doba by teoreticky měla být o tuto hodnotu prodloužena, pro zachování funkčnosti systému. V realitě, však nemusí dojít k úplné kompenzaci v následujícím pulzu, ale tato kompenzace bude pravděpodobně rozložena do více pulzů následujících.

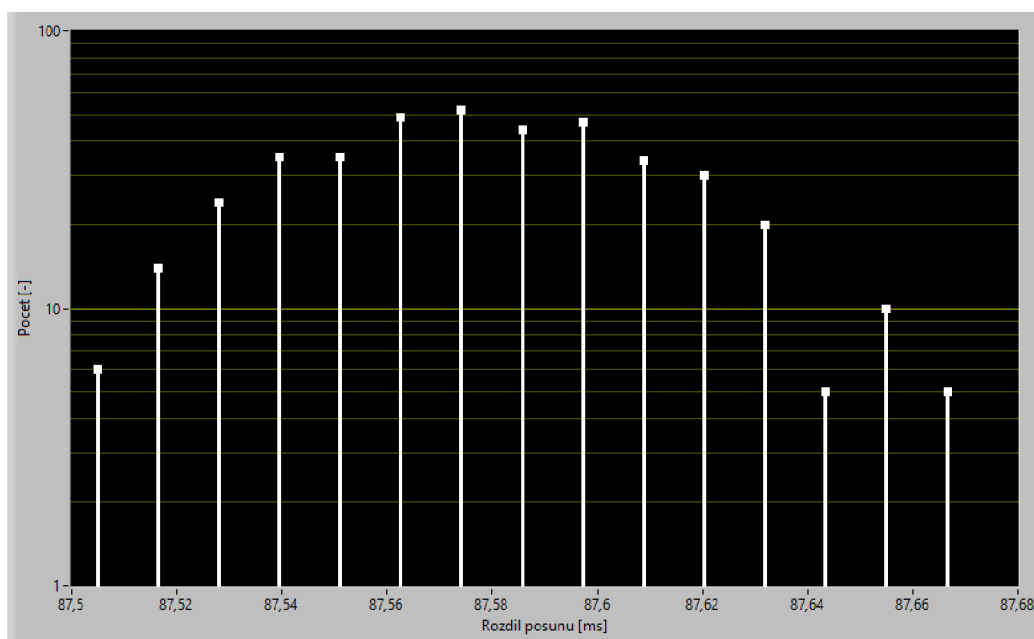
Z této úvahy, tedy vzniká otázka, po kolika pulzech je vhodné provést synchronizaci mikrokontroleru? Pokud budeme uvažovat případ, kdy první pulz je zcela náhodný, tedy jedná se o případ, kdy všechny chyby způsobené dobami předcházejících pulzů byly eliminované, tak tento první pulz přináší chybu v změřeném rozptylu z teoretické hodnoty.

Ovšem pokud budeme uvažovat, že se jedná již o X-tý pulz, který je opět čistě náhodný, tedy bez předchozí vazby, bude se opět jednat o chybu z rozptylu teoretické hodnoty. Tentokrát však již bude ovlivňovat výsledek rovný teoretické hodně násobené hodnotou X, čímž tedy chyba bude vůči celkové hodnotě, X násobně menší. Z tohoto vyplývá, že ideální hodnotou by bylo nekonečno, popřípadě velmi vysoké číslo.

Toto by se však setkalo s problémem vznikající na mikrokontroleru. Neboť by v době mezi synchronizacemi začalo docházet k projevu driftu na RC oscilátoru, čímž by se celková synchronizace začala zhoršovat. Tedy z toho hlediska by bylo ideální provádět synchronizace v co nejmenším možném časovém rozestupu.

Tedy pokud se nyní vrátíme zpět k položené otázce, po kolika pulzech je vhodné provádět synchronizaci, již víme že odpověď na tuto otázku není zcela jednoznačná a je tedy nutné volit kompromis mezi těmito dvěma podmínkami. Z toho důvodu byla nakonec zvolena hodnota 12, která má výhodu v tom, že je blízká hodnotě 1 sekundě, a tím pádem bude tato metoda snáze porovnatelná s výsledky získané metodou GPS.

Na obrázku 9.3, je tedy poté vidět histogram zprůměrovaných dat. Jak je z obrázku patrné, tak opravdu tímto průměrováním došlo k zmenšení rozsahu. Výsledná střední hodnota je 87,5790 ms, kdy směrodatná odchylka byla zmenšen na hodnotu 0,0311 ms. Což ve výsledku znamená, že zprůměrováním hodnot se zlepšila směrodatná odchylka o jeden řád.

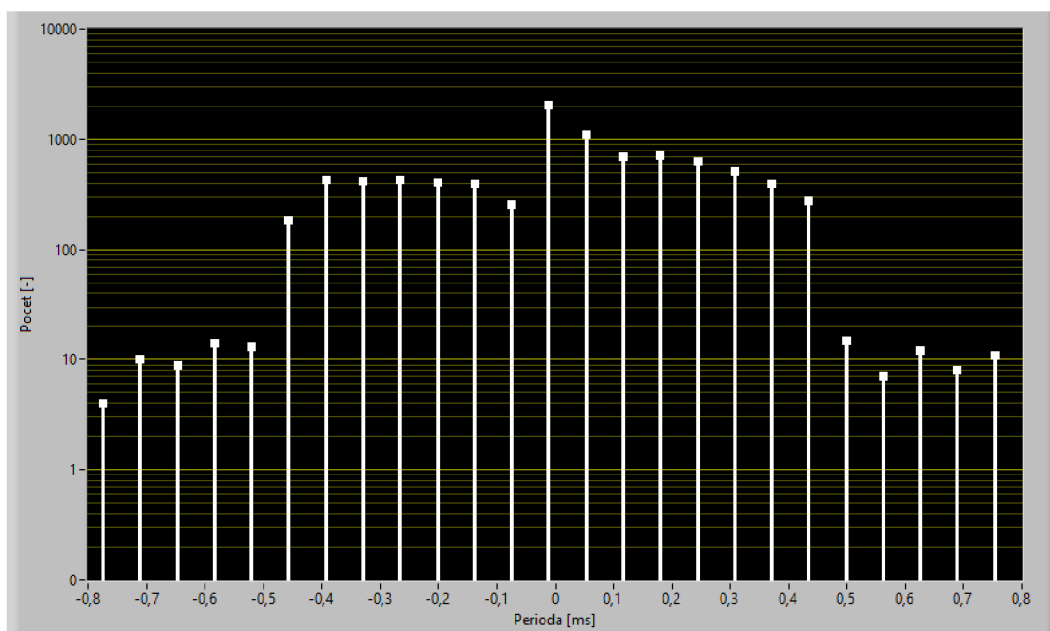


Obrázek 9.3 Histogram naměřených period zprůměrovaných pulzů

9.2 Posun signálu mezi přijímači RDS

Stejně tak jako u metody využívající GPS, je nutné proměřit i hodnoty posunů dvou RDS přijímačů ve vyvolaných pulzech. Proto stejně jako u metody GPS, byly proměřeny příchody pulzů pomocí čítače Agilent 53131A. Z tohoto důvodu v této kapitole nebude zmíněn postup měření ani případné problémy, které se objevily i u metody GPS, neboť všechny tyto vlastnosti byly popsány již v kapitole 8.2. Jediným velkým rozdílem zde je fakt, že od opačně změřených hodnot nebude odečtena 1 sekunda ale hodnota 87,576 ms.

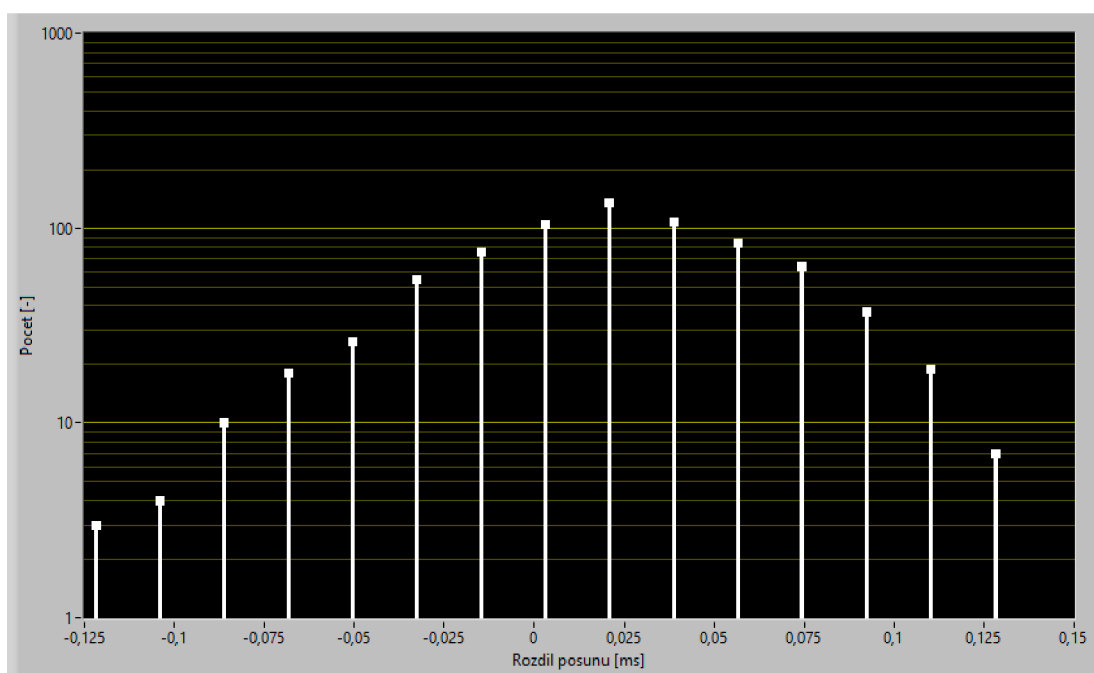
Na obrázku 9.4, je tedy možné pozorovat histogram naměřených dat kdy, jak je z obrázku patrné, nejvíce hodnot je v okolí nulového rozdílu. Z daného histogramu je také patrné, že naprostá většina hodnot se nachází v rozsahu $\pm 0,45$ ms. Střední hodnota je poté rovna 0,0214 ms se směrodatnou odchylkou 0,2272 ms.



Obrázek 9.4 Histogram naměřených posunů mezi dvěma moduly

Stejně jako v první podkapitole by bylo vhodné proměřit i posuny, při využití více hodnot, které následně budou zprůměrovány. To je možné pozorovat na obrázku 9.5 kde, jak je vidět, že došlo k výraznému zmenšení rozsahu v rozdílech posunů pulzů. Tato skutečnost má tedy pozitivní vliv na odlišnost časů v jednotlivých mikrokontrolerech. Zatím co, bez průměrování by rozdíl časů mohl pohybovat v rozmezí $\pm 0,8$ ms, nyní by se rozdíl neměl dostat přes rozsah $\pm 0,125$ ms.

Také by bylo dobré zdůraznit, že nejvíce hodnot již nebylo zaznamenáno v čase 0 ms, ale v čase 24 μ s. Čímž je i ovlivněna střední hodnota, která je rovná 21,445 μ s se směrodatnou odchylkou 45,222 μ s.



Obrázek 9.5 Histogram zprůměrovaných posunů mezi dvěma moduly

9.3 Použití GPS k měření doby mezi pulzy RDS

Protože při vytváření synchronizace pomocí DCF77, bylo zjištěno, že tento signál není možné, kvůli rušení proměřit v laboratořích v budovách fakulty FEKT, bylo nutné vymyslet jiný způsob časového měření dané metody. Jako jedním z možných způsobů se zdálo možnost využití PPS signálu přicházející z GPS, neboť perioda toho signálu je velmi přesná a blízká 1 sekundě, jak již bylo zjištěno v kapitole 8.1. Proto pro zjištění, zda by tato metoda byla využitelná, byla tato metoda použita i pro proměření RDS hodnot, kdy poté na základě srovnání obou výsledků bude možné určit, zda je tato metoda srovnatelná s čítačem Agilent 53131A.

9.3.1 Popis provedení měření

Tato metoda je založena na porovnávání měřené metody s 1 PPS signálem z GPS. K tomu to porovnání dochází v mikrokontroleru, kdy každý z příchozích signálů má přiřazený vnitřní čítač, který počítá počet tiků, který proběhl mezi příchodem dvou pulzů obou systémů. Protože se patrně nestane, že by oba signály přišly v ten samý moment, je nutné ještě u měřeného signálu, odečíst hodnotu načítaných tiků z čítače určeného pro GPS, které byly provedeny před tím, než přišel měřený signál.

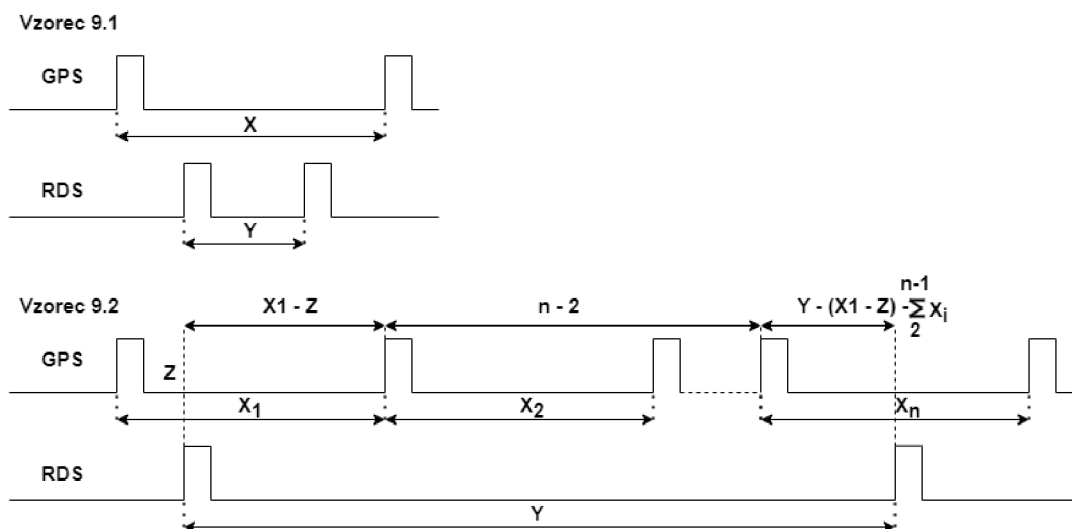
Po změření počtu tiků obou systému v době mezi pulzy, je již možné vypočítat časovou dobu měřeného signálu, kdy je čas možné vypočítat podle následujících vzorců. Vzorec 9.1 je platný pouze, pokud měřený interval nezasáhl do více intervalů GPS,

naopak vzorec 9.2 je platný, právě pokud interval měřeného signálu zasáhl do více intervalů GPS, kdy vysvětlení proč tomu tak je, je možné pozorovat na obrázku 9.6

$$t = \frac{Y}{X}, \quad 9.1$$

$$t = \frac{X_1 - Z}{X_1} + \frac{Y - (X_1 - Z) - \sum_{i=2}^{n-1} X_i}{X_n} + (n - 2), \quad 9.2$$

kdy X_1 je počet tiků čítači pro GPS, v příchodu prvního pulzu měřeného signálu, X_n je počet tiků v čítači pro GPS, v příchodu druhého pulzu měřeného signálu, Z je hodnota čítače pro GPS v momentě příchodu prvního pulzu měřeného signálu, Y je hodnota čítače pro měřený signál a n je počet použitých intervalů.



Obrázek 9.6 Obrázek zobrazující vysvětlení rovnic 9.1 a 9.2

9.3.2 Problémy této metody

První z problémů, který tato metoda má je právě ve stabilitě GPS signálu, kdy jeho rozsah hodnot je možné pozorovat v kapitole 8.1. Jak je vidět tento signál bude do výsledků přinášet chybu v řádu desítek ns, což znamená, že systémy s nižší chybou budou více ovlivněny chybou měření než svojí vlastní.

V našem případě je však vidět, že tento problém nenastane, neboť základní chyba RDS systému se pohybovala o 3 řády níž. Při měření DCF signálu, poté z dohledaných informací, bylo zjištěno, že chyba dané metody se bude pohybovat v řádek jednotek ms, tedy v porovnání s GPS v mnohem nižších řádech přesnosti. Z toho lze vyvodit, že tento problém za těchto podmínek bude možné zanedbat.

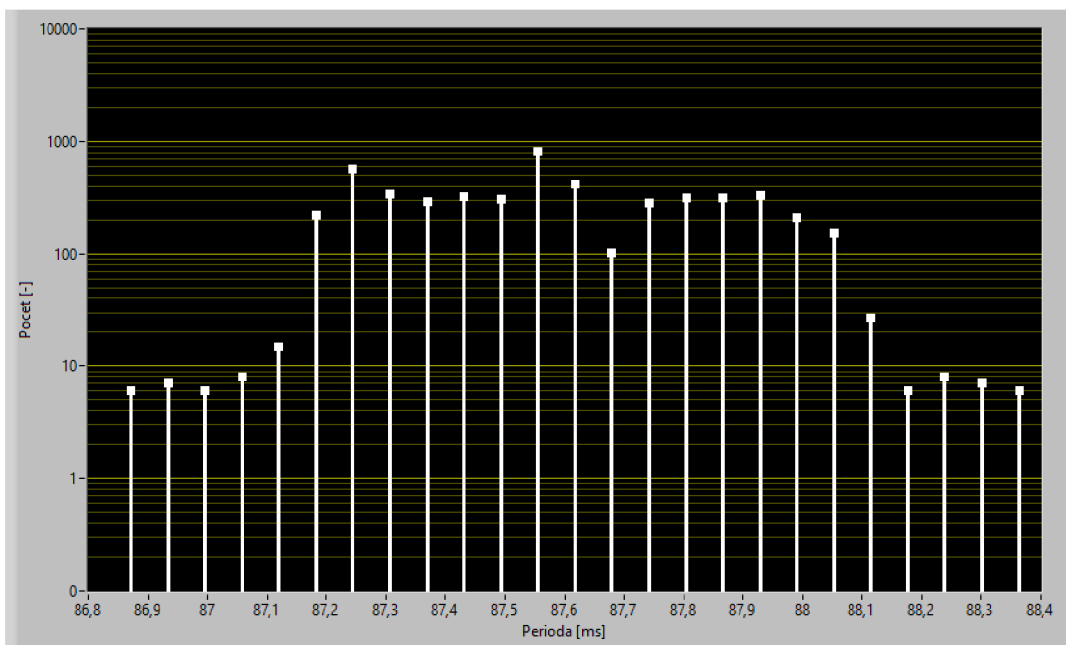
Jako další problém se může jevit problém s čtením digitálních vstupů na mikrokontroleru, kdy podobný problém již byl řešen v kapitole 8.3, při měření doby posunu GPS signálu při použití mikrokontroleru. Z toho důvodu, bylo měření prováděno,

při nastavení maximální frekvence mikrokontroleru a to při 108 MHz. Z toho lze stanovit maximální chybu rovnou 9,259 ns. Jak je vidět tato chyba je srovnatelná s chybou způsobenou GPS. Lze tedy taktéž tuto chybu zanedbat, při provádění daných měřeních.

Asi jako poslední problém, který by zde bylo možné zmínit je nutnost přijímat právě zmíněný signál z GPS, kdy na rozdíl od měřících přístrojů v laboratořích, které pracují bez nutnosti přijímat z okolí nějaké porovnávací signály, tento měřicí systém se stává nefunkčním, při ztrátě GPS signálu.

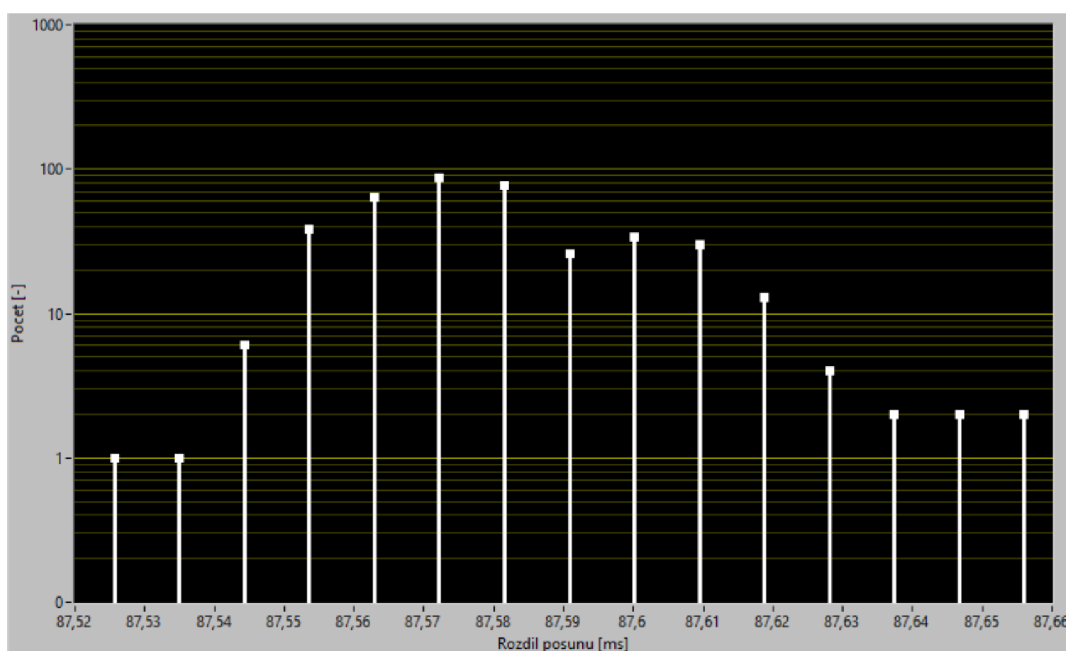
9.3.3 Výsledky měření

Nyní když, již známe chyby vznikající na použité metodě, je třeba porovnat výsledky měření. Ty jsou možné pozorovat na obrázku 9.7 kdy, jak je vidět, výsledný histogram je velmi podobný histogramu naměřený dat pomocí čítače. Stejně tomu je i v případě statistických operací, kdy střední hodnota je 87,578 ms se směrodatnou odchylkou 0,259 ms. Když toto porovnáme s výsledky měření získané pomocí čítače, zjistíme že rozdíl ve výsledcích je minimální. Střední hodnota se liší o 1 μ s a jejich směrodatná odchylka se liší o 20 μ s. Tedy z těchto výsledků, lze měření považovat za rovnocenné.



Obrázek 9.7 Histogram naměřených period RDS pulzů pomocí GPS

Stejně tomu tak je při srovnání výsledků metodou průměrování více hodnot. Výsledkem toho měření je opět histogram 9.8, kdy střední hodnota je rovna 87,5795 ms se směrodatnou odchylkou 0,0203 ms.



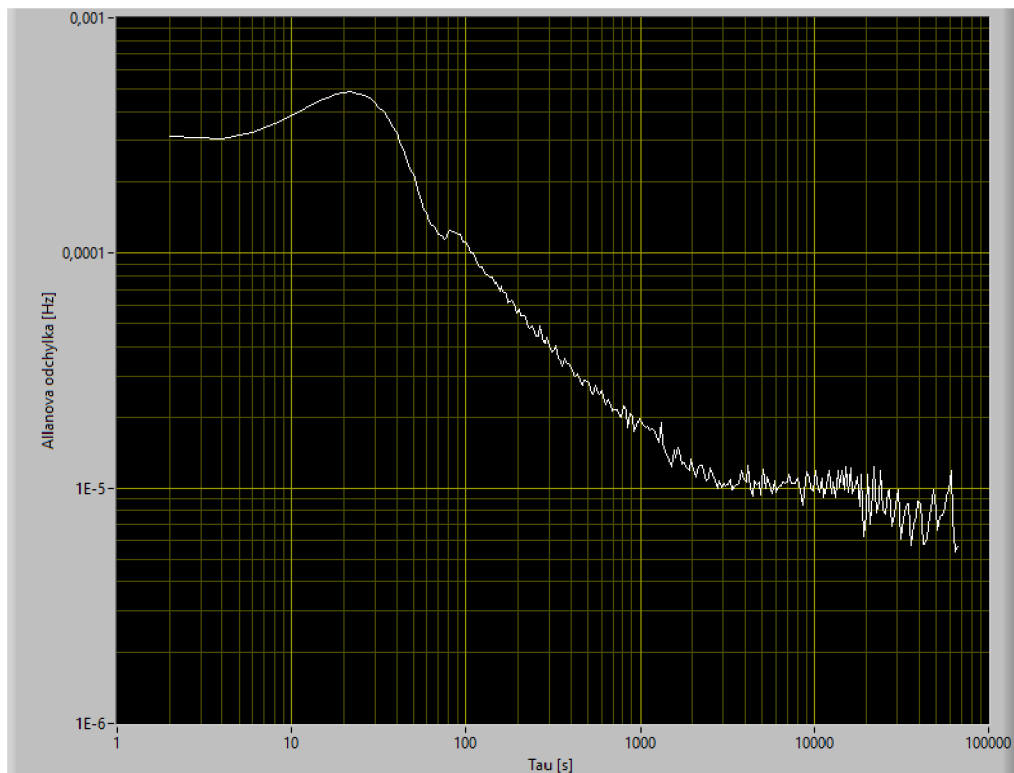
Obrázek 9.8 Histogram naměřených period RDS pulzů pomocí GPS při použití průměrování

9.4 Allanova odchylka

Stejně jako u metody GPS, i u této metody byl vytvořen regulační systém pro řízení frekvence kmitů vnitřního RC oscilátoru. Tento systém se od vytvořeného regulačního systému pro GPS v mnoha věcech neliší, přesto bylo nutné provést některé změny. Jedna ze změn již byla zmíněna v předcházejících kapitolách, a to, že výpočet odchylky vstupující do regulátoru je prováděn na základě průměrování 12 hodnot. Čímž jak bylo dokázáno v kapitole 9.1, dochází k zmenšení chyby způsobené nepřesností periody přerušování RDS signálu.

S tím je spojená i požadovaná hodnota frekvence. Neboť pokud by byla hodnota fázového závěsu upravována jednou za sekundu, byla požadovaná hodnota rovná frekvenci RC oscilátoru, tedy osmi milionům. Nyní je však hodnota upravována jednou za 1,051 s, kdy tato hodnota vychází z výpočtu doby jedné periody RDS signálu, která je rovná 87,576 ms a počtu použitých period, který je v tomto případě rovný 12. Z čehož vyplývá, že požadovaná hodnota je rovna 8407296.

Jako poslední změnu, kterou bylo třeba provést byla změna sledované hrany, kdy při použití GPS systému byla sledována náběžná hrana, zatím co při použití RDS systému je třeba sledovat sestupnou hrana.



Obrázek 9.9 Allanova odchylka pro RDS signál

Na obrázku 9.9, je možné pozorovat graf zobrazující Allanovu odchylku. Jak je vidět, tak na začátku dojde k zhoršení stability frekvence, kdy tato skutečnost je způsobena přechodovým dějem při regulaci. Následně již dochází k poklesu odchylky, kdy tato hodnota se zastaví okolo hodnoty 10 μHz .

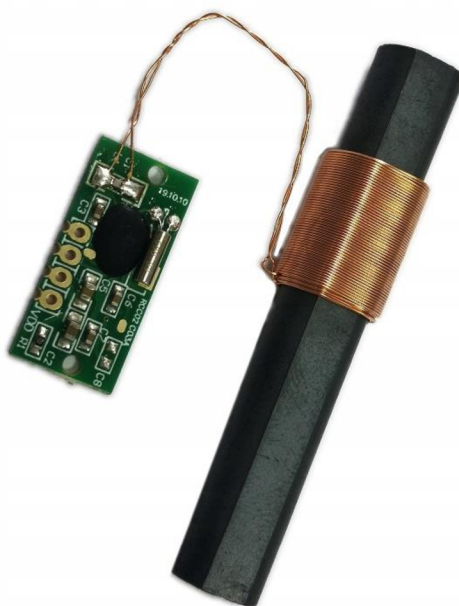
9.5 Zhodnocení metody RDS

Jak bylo v této kapitole ukázáno, tak tato metoda má svoji přesnost v řádu stovek μs . Při využití průměrování 12 hodnot tato přesnost se zpřesnilo o jeden řád, tedy desítek μs . Také při této metodě, bylo dokázáno, že lze měřící metodu využívající GPS a mikrokontroler považovat za funkční. A lze tedy tuto měřící soustavu využít pro proměření DCF77.

Výsledná Allanova odchylka ukázala, že tato metoda nedosahuje tak přesných výsledků jako GPS. Přičemž její přesnost je horší pouze o jeden řád.

10. SYNCHRONIZACE POMOCÍ DCF77

Pro příjem DCF signálu byly pořízeny přijímače RCC02 CO3A, které přicházející signál zachytí na feromagnetické antény. Tento modul, přijímá signál vysílání na frekvenci 77,5 kHz, kdy následně vyvolává pulzu na svém výstupu, při zachycení poklesu amplitudy příchozího signálu. Tento pokles je vytvářen na začátku každé sekundy vyjma poslední sekundy v každé minutě. Z toho vyplývá, že synchronizační pulz je vytvářen 59 krát za minutu s periodou jedné sekundy.



Obrázek 10.1 Modul přijímače DCF signálu RCC02 CO3A [23]

10.1 Úprava přijímačů DCF77

Protože pořízené přijímače DCF, nebyly schopny zachytit vysílaný signál. Bylo třeba zjistit, zda mají tyto přijímače správně naladěné rezonanční frekvence na hodnotě 77,5 kHz. Proto bylo nutné odpájet z nich cívku a kondenzátor, který tvoří rezonanční obvod, a tyto součástky proměřit.

Pro proměření parametrů byl následně použit LCR meter Hioki 3532, který je schopen proměřit indukčnost a kapacitu na požadované frekvenci. Kdy výsledné hodnoty indukčnosti cívky byla 1,2719 mH a kapacita kondenzátoru byla 3,152 nF, přičemž při využití Thomsonova vzorce, nám vyjde rezonanční frekvence rovna 79,487 kHz. Tato frekvence je již velmi odlišná od požadované a z toho důvodu bylo třeba vyměnit daný kondenzátor za takový, který bude mít ideálně 3,3158 nF, popřípadě za více kondenzátorů, který paralelním spojením svých kapacit tuto hodnotu splní. Součet

výsledných kapacit použitých kondenzátoru je 3,317 nF, čímž byla rezonanční obvod nastaven na frekvenci 77,485 kHz.

Thomsonův vzorec zní

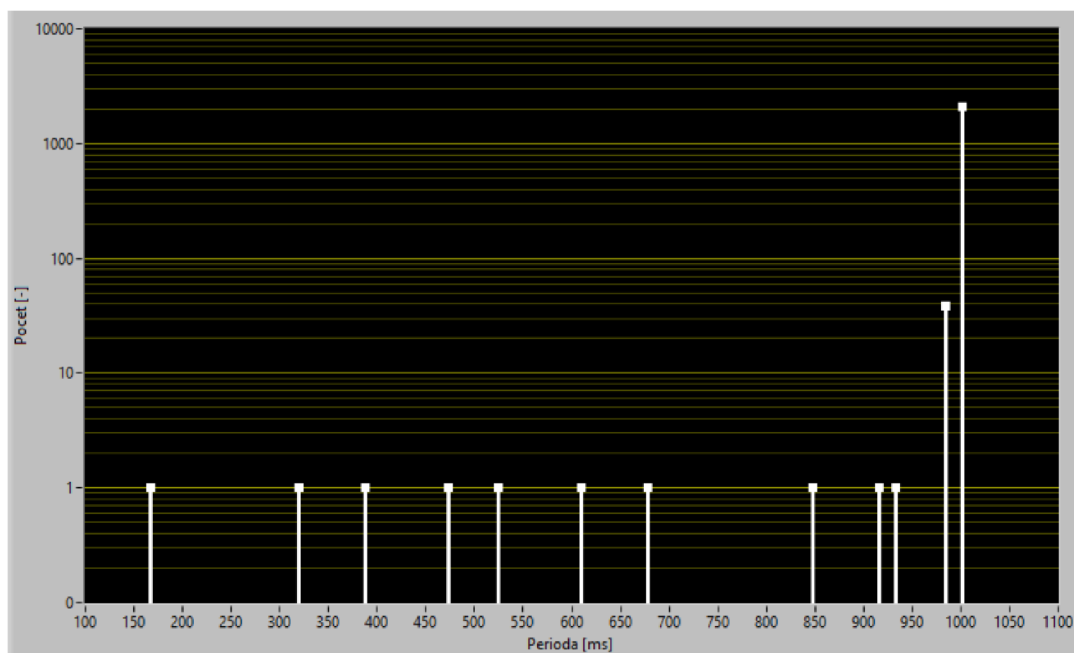
$$f = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}}, \quad 10.1$$

kdy f je výsledná rezonanční frekvence, L je indukčnost cívky a C je kapacita kondenzátoru.

10.2 Měření doby mezi po sobě jdoucími pulzy

Jak již bylo zmíněno v kapitole 9.3, frekvence používaného signálu DCF byla v budově FEK Tu rušena. Z toho důvodu bylo třeba tento signál proměřit na jiném místě, kde však nebylo možné použít čítač Agilent 53131A. Z toho důvodu byl vytvořen měřicí systém využívající mikrokontroler společně s PPS signálem vznikající na modulu GPS. Přičemž tento PPS signál byl využíván jako zdroj přesného času jedné sekundy. Více informací o tomto měření je popsáno v zmíněné kapitole 9.3.

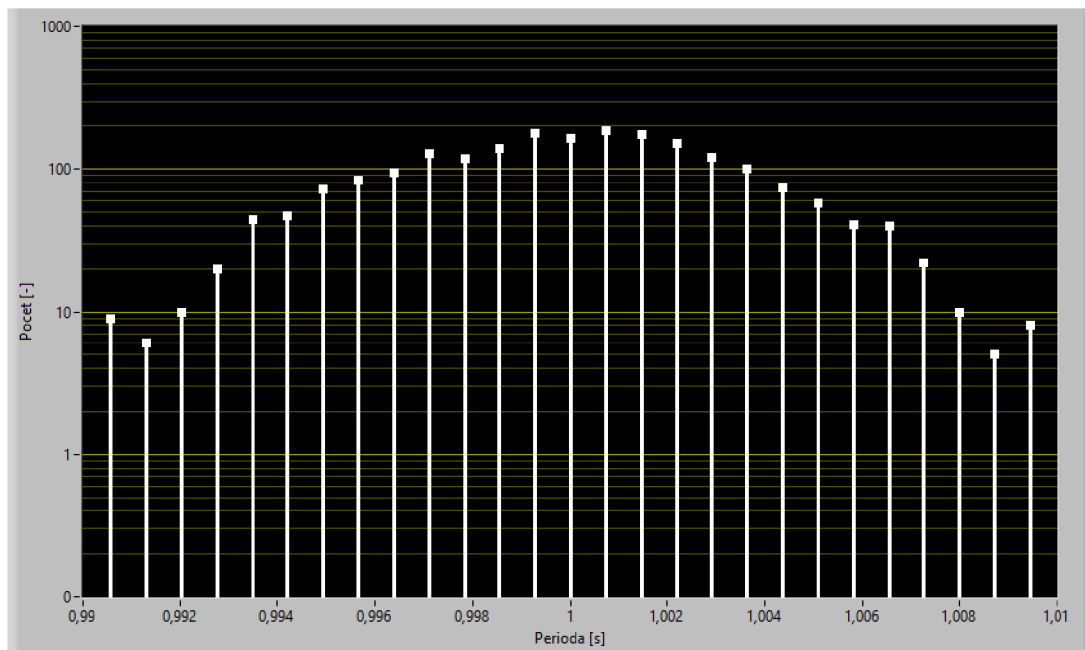
Před představení změřených výsledků by bylo patrně vhodné zmínit fakt, že tato metoda je více ovlivňována rušením než předcházející metody. Z toho důvodu, byla data z měření ručně upravována, kdy byly odstraňovány hodnoty vzniklé právě tímto rušením. Při této úpravě byly odstraňovány hodnoty lišící se od teoretické hodnoty o více než 50 ms. Tedy byly ponechány pouze hodnoty v rozmezí 0,95 až 1,05 s. Kdy tento rozsah byl zvolen na základě změřených dat. Hodnoty mimo tento rozsah se objevují pouze občasně a v zastoupení pouze jedné hodnoty, jak je možné pozorovat na obrázku 10.2.



Obrázek 10.2 Perioda DCF signálu neupravena

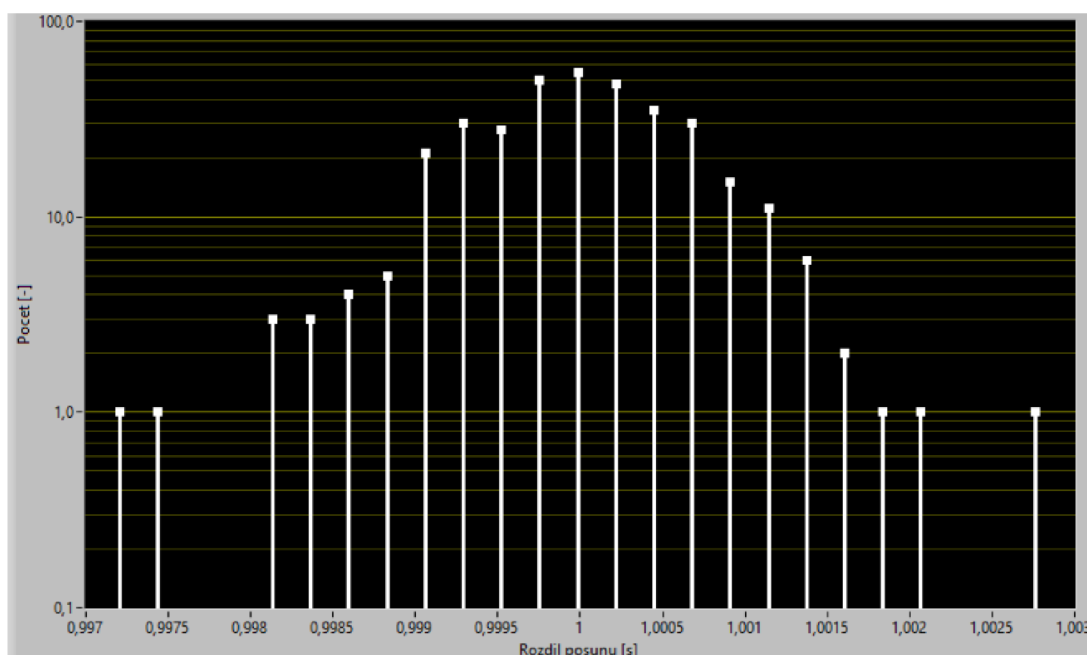
Z toho plyne, že v ponechaných hodnotách mohou být stále hodnoty změřené na základě rušení, přičemž však není možné je jednoznačně odlišit od dat, které byly změřeny na základě synchronizačních pulzů.

Na obrázku 10.3, lze tedy pozorovat upravená data z měření. Jak je vidět rozsah změřených hodnot se pohybuje v rozmezí ± 10 ms, od teoretické hodnoty. Střední hodnota je poté 999,991 ms, kdy směrodatná odchylka je 3,473 ms.



Obrázek 10.3 Perioda DCF signálu

Stejně jako u metody RDS, zde stojí za zvážení využití průměrování více hodnot, čímž by se opět měl zmenšit rozsah hodnot. To je možné pozorovat na obrázku 10.4. Přičemž je vidět, že se rozsah hodnot zmenšil na ± 3 ms v okolí teoretické hodnoty, když budeme brát i časy, které jsou v minimální zastoupení, pokud ovšem budeme uvažovat pouze hodnoty, které jsou zastoupeny častěji tak se nám rozsah zúží na $\pm 1,5$ ms z teoretické hodnoty. Výsledná střední hodnota je poté rovna 999,9909 ms se směrodatnou odchylkou 0,7064 ms.



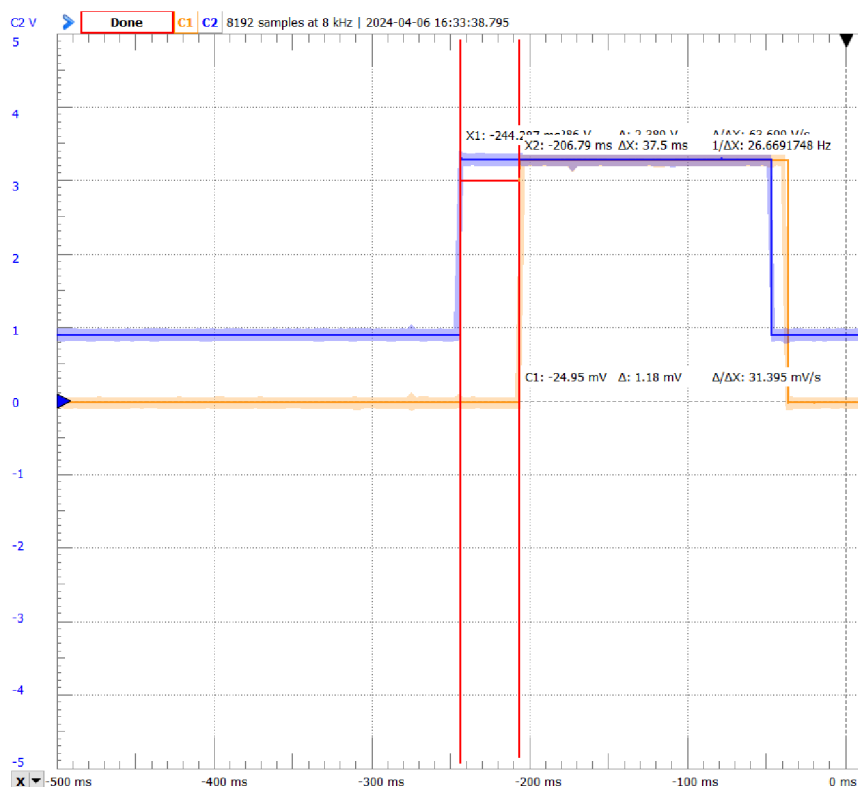
Obrázek 10.4 Perioda DCF signálu, při využití průměrování 6 hodnot

10.3 Posun signálu mezi dvěma přijímači DCF77

Stejně jako tomu bylo u GPS a RDS, je nutné proměřit i posun pulzů na dvou modelech. K tomu bude opět využit měřicí systém využívající mikrokontroler a GPS signál. Neboť by oba signály jak GPS, tak DCF, měli mít periodu 1 sekundy, je nutné si položit otázku či tyto signály budou přicházet v podobný moment nebo jsou od sebe nějak časově posuny. Tato skutečnost bude mít vliv na použitém vzorci pro výpočet doby mezi příchodem dvou pulzů.

Jak se vidět na obrázku 10.5, tak i když by tyto dva systémy měly vygenerovat pulz na začátku sekundy, tak tomu ve výsledku není. Posun v začátku jednotlivých sekund je rovný 37,5 ms, kdy jako první přichází GPS signál. Tento jev může být způsoben dobou šíření signálu od vysílačů k přijímačům.

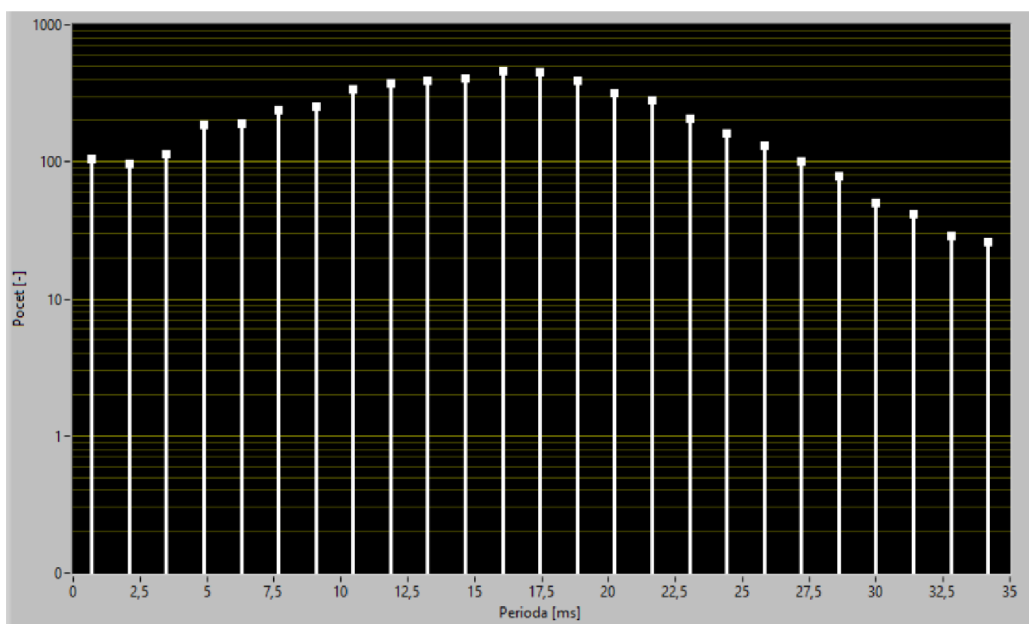
Z toho vyplývá, že pro výpočet doby mezi dvěma pulzy bude stačit využívat vzorec 9.1, který slouží k výpočtu posunu v jedné periodě GPS signálu. K určení počtu tiků mezi dvěma pulzy, bylo nutné upravit lehce připravený kód. Nyní při příchodu prvního pulzu bude spuštěn časovač, který bude počítat počet tiků do doby, než přijde druhý pulz, který následně tento časovač zastaví, a to do doby, než přijde další pulz z následující sekundy, kdy opět bude spuštěn resetovaný časovač. Tato úprava byla nutná provést z důvodu, že předcházející kód počítal s tím, že vždy když přijde další pulz je jedno měření ukončeno a ihned je i další měření spuštěno.



Obrázek 10.5 Posun GPS a DCF signálu, kdy modrý signál je pro GPS, zatímco oranžový pro DCF

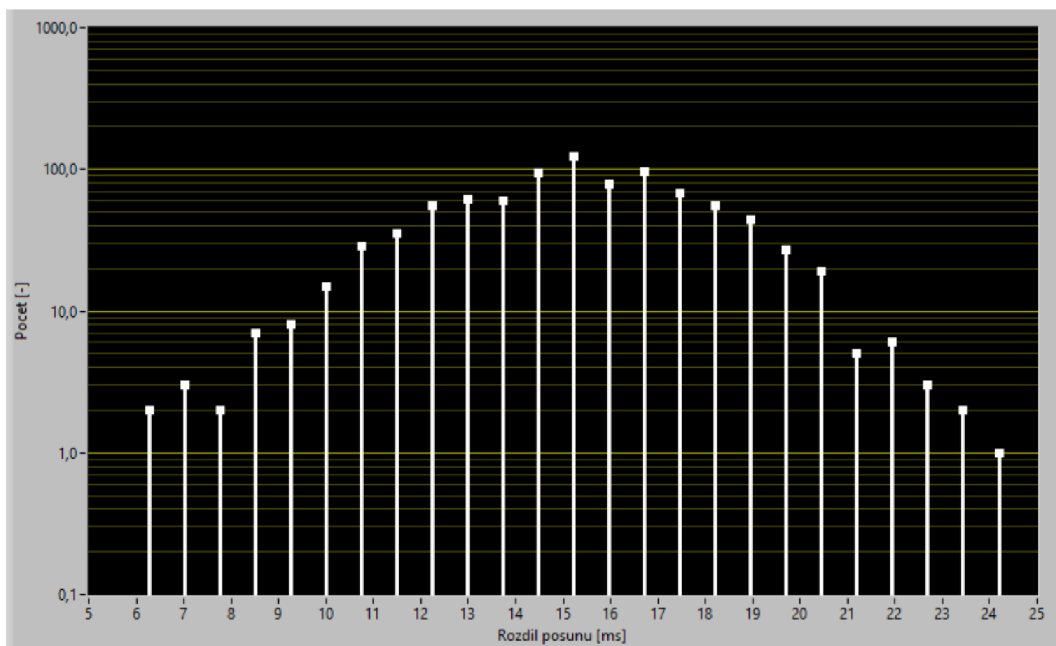
Na obrázku 10.6, je následně možné vidět naměřené hodnoty posunu pulzů mezi dvěma DCF moduly. Jak je možné z histogramu vyčíst nejvíce hodnot není v okolí 0 ms, ale v čase 16 až 18 ms. Další překvapím výsledkem jsou poté hodnoty od 20 ms a více, kdy z výsledků měření periody signálu, se zdálo, že maximální rozsah by zde neměl překročit rozsah 20 ms, přesto výsledek zde ukazuje, že rozdíl může být až dvojnásobný. Tento jev by mohl být způsoben použitím dvou různých typů DCF přijímačů.

Je důležité podotknout, že stejně jako u metody měření periody pulzů, bylo na datech provedena ruční filtrace rušení, kdy opět hodnoty vyšší než 50 ms, byly odstraněny. Také na základě použité metody nebyla měřena záporná data, čím střední hodnota je znatelně posunuta, a to do hodnoty 11,555 ms se směrodatnou odchylkou 4,572 ms.



Obrázek 10.6 Posun pulzů na dvou modulech DCF

Na obrázku 10.7, jsou následně vidět data, při využití průměrování více hodnot. Kdy, jak je vidět, touto úpravou opět došlo k snížení vzdálených hodnot od teoretických hodnot, kdy společně s tím i nejčetnější hodnoty byly posunuty více k hodnotě 0 ms. Střední hodnota je 15,303 ms se směrodatnou odchylkou 2,811 ms. Při porovnání těchto hodnot s hodnotami bez průměrování je patrné, že se střední hodnota vzdálila od teoretické hodnoty, naopak se však rozptýl tímto krokem snížil.



Obrázek 10.7 Posun pulzů na dvou modulech DCF s využitím průměrování

10.4 Regulační obvod pro systém využívající DCF77

Stejně jako tomu bylo u metod GPS a RDS, i v této metodě byla vytvořena regulační smyčka, která bude upravovat vnitřní RC oscilátor pomocí fázového závěsu. Kdy jako zpětná vazba zde bude použita hodnota čítače napočítaná za dobu mezi PPS signálem z DCF.

Podobně jako tomu bylo u metody RDS, i zde bylo použito více zprůměrovaných dat, k vytvoření zpětné vazby. Kdy jako počet použitých dat byla určena hodnota 6. Zde je však potřeba uvést, že tato metoda musí řešit problém s chybějícím jedním pulzem, který je v každé padesáté deváté sekundě. Z toho důvodu, je kontrolována hodnota, která je používána ke zpětné vazbě, a pokud dojde k tomu jevu, je tato hodnota využita s tím, že je počítána za dva pulzy. Tedy pokud se mělo jednat v pořadí o 6 pulz, je výsledný počet použitých pulzů rovný 7, na základě čehož je tedy nutné počít počet přichozích pulzů, aby byl systém stále schopný fungovat stejně, přičemž se to projeví pouze tak, že v dané skupině je průměrováno více hodnot.

Další problém, který zde bylo třeba řešit bylo objevující se rušení. Z toho důvodu byla přidána podmínka, která za správný pulz uzná pouze pulz, který je v určitém rozsahu. Tato hodnota rozsahu byla určena jako ± 400000 tiků, kdy tato hodnota odpovídá rozsahu ± 50 ms. Pokud by rušení, přišlo v tomto rozsahu již není možné touto metodou ho možné odlišit. Zároveň, je nutné tuto hodnotu zvolit dostatečně vysokou, neboť na začátku se RC oscilátor lišil i o statisíce tiků oproti požadované hodnotě.

10.5 Zhodnocení metody DCF77

Pro změření časových posunů pulzů, bylo nutné tuto metodu proměřit pomocí GPS a mikrokontroleru, kdy funkčnost této měřicí soustavy byla dokázána, při měření RDS. Z naměřených dat je pak patrné, že tato metoda je patrně nejhorší ze všech. Kdy její přesnost se pohybuje v řádu jednotek ms. Ovšem stejně jako tomu bylo u RDS, i zde při využití více hodnot pro průměrování došlo ke zlepšení výsledné přesnosti o jeden řád.

Výsledná Allanova odchylka v této metodě nemohla být změřena, kvůli nemožnosti využití čítače 53131A mimo prostory VUT a nemožnosti chytit nerušení signál v prostorách VUT. Z toho důvodu, pro porovnání přesnosti této metody při regulaci lze využít pouze změřené odchylky. Přičemž tento parametr je porovnáván v kapitole 12, spolu s ostatními metodami.

11. NÁHRADNÍ MODEL

Protože vytvořené systémy jsou schopny fungovat pouze pokud je zachytáván signál z globálních vysílačů, stávají se tyto vytvořené systémy nefunkční, pokud dojde ke ztrátě signálu. Představené systémy byly prozatím navrženy tak, aby při ztrátě signálu, zůstaly na poslední nastavené hodnotě. Takovéto řešení však není ideální, neboť takto není jisté, na jaké hodnotě zůstal systém zastaven. Navíc tato hodnota může být více vzdálená od požadované, neboť zrovna došlo k reakci regulátoru na vyšší integrační hodnotu.

Jedním z nejjednodušších řešení by bylo ukládání si historie nastavených hodnot, patrně použitého indexu od look-up tabulky, kdy při výpadku signálu by byl nastaven nejčastější použitý index. Toto řešení by mělo mít za výhodu, že použitá hodnota, by patrně nebyla příliš vzdálená od požadované hodnoty, čímž by se eliminovala možná nepříznivá chyba z minulého odstavce. Nevýhodou toho řešení by bylo úplné odstavení regulátoru, čímž by se systém stal stacionárním bez možnosti reakce na změny.

Z tohoto důvodu by bylo lepší, kdyby při výpadku signálu, bylo možné stále využívat navržený regulátor, kdy vstupní data by byla vytvářena na základě náhradního modelu. Přičemž by tento model byl vytvářen z určitého počtu předcházejících dat.

Následující představené náhradní modely, byly všechny vytvořeny a testovány pouze pro GPS, kdy však nejsou vytvářeny speciálně pro GPS, ale mohly by být přidány i do ostatních systému, pokud by byla upravena doba mezi provádění úpravy regulátoru na stejnou hodnotu jako tomu je při jejich normální fungování.

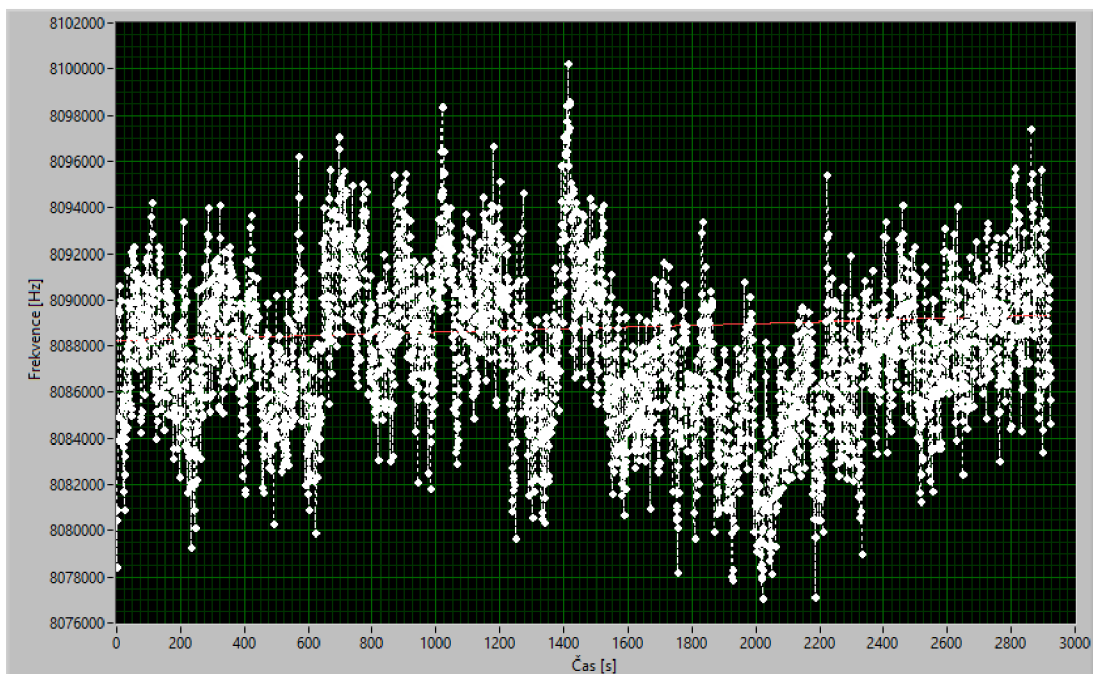
11.1 První verze náhradní model

Jak již bylo řečeno v minulé kapitole, tvoření náhradního modelu bude prováděno pokaždé kdy dojde k výpadku signálu, tedy nebude v programu uložen jeden globální model, který by byl používán, pokud dojde k výpadku signálu, ale náhradní model se bude vždy tvořit nově. Takové to řešení má nevýhodu v přidání dalších operací do programu, které bude nutné zpracovat při výpadku signálu. Naopak výhodou daného řešení je, že není nutné vytvářet globální model, který by musel být schopen fungovat v různých případech. Také model tvořený daty z blízké minulosti bude lépe popisovat aktuální situaci než ten globální.

Pro vytváření náhradní modelu budou tedy ukládány předcházející data do polí, kdy po při příchodu nových dat budou zapomínána nejstarší data. V první iteraci tvoření modelu byla ukládána pouze data o aktuální odchylce a použitém indexu z look-up tabulky. Kdy při výpadku signálu došlo k sečtení odchylek, pro stejné indexy a následnému podělení toho součtu počtem dat k danému indexu, čímž došlo k získání průměrné odchylky na daném indexu. Přičemž tyto průměrné hodnoty byly považované za náhradní model.

Původní domněnka byla, že tyto data budou platná pouze po určitou dobu, a následně bude nutné tyto hodnoty upravovat na základě uplynulého času. Z toho důvodu došlo k proměření změny hodnot v čase.

To bylo proměřeno pomocí GPS systému, kdy do programu byla přidána podmínka, která po tom, co systém byl ustálen odstavil regulátor, čímž bylo možné sledovat změnu odchylky v čase. To je možné pozorovat na obrázku 11.1 kdy, jak je vidět hodnoty se za uplynulou dobu, více méně nezměnili. Pro výpočet, jak velmi se tyto hodnoty změnily v čase byla využita metoda nejmenších čtverců, kdy byl zvolen lineární popis. V tomto případě nás zajímá pouze sklon, který vyšel 0,0384. Z čehož vyplývá, že pro změnu odchylky o jeden tik, musí uplynout 26 sekund. Proto, aby došlo k posunu hodnot o jeden index, což je něco mezi 2500 až 10000 tiky, musí uběhnout doba v rozmezí 18 až 72 hodin. Z toho důvodu, tato úprava nebyla přidána do výpočtu modelu, a tedy po uplynutí takto dlouhého úseku se může ve vytvořeném modelu daná chyba projevit.

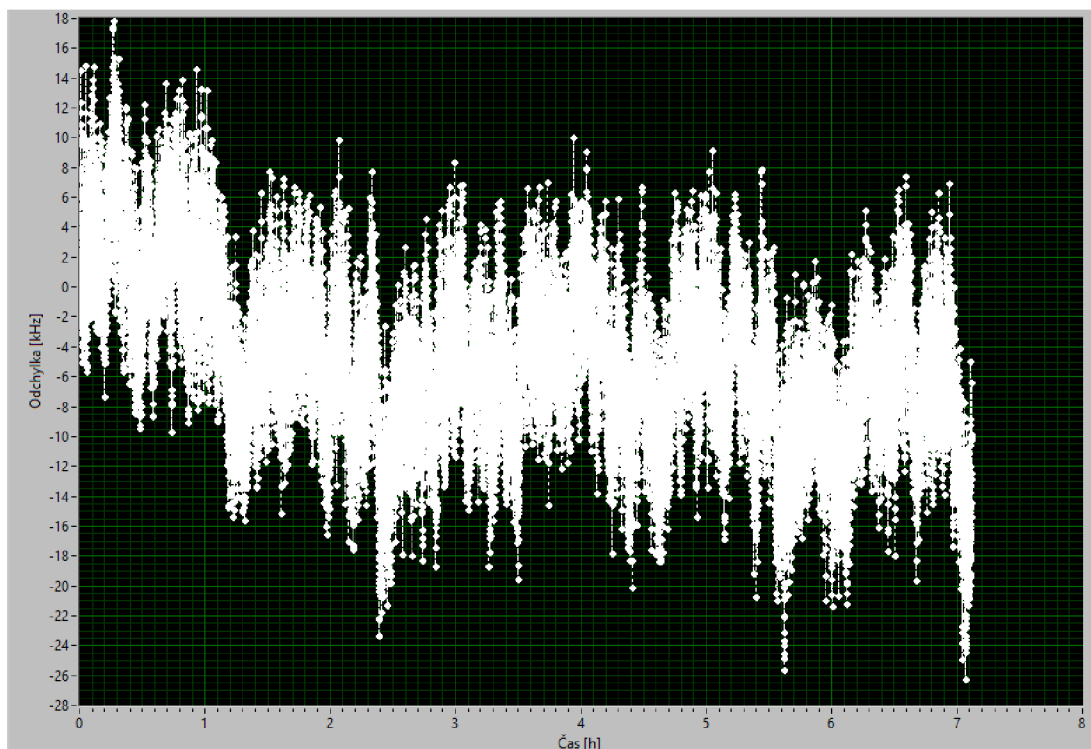


Obrázek 11.1 Frekvence mikrokontroleru

Tedy nyní když již máme vytvořený náhradní model, je třeba na základě něho upravovat vnitřní stav fázového závěsu. To je prováděno pomocí stejného regulátoru, jako je použit při normální stavu. Kdy jako odchylka vstupující do regulátoru je použita průměrná odchylka z aktuálně používaného indexu. K dané regulaci dochází po uplynutí požadovaného počtu tiků vnitřního RC oscilátoru, kdy tento požadovaný počet tiků se shoduje s požadovanou hodnotou frekvence.

Na obrázku 11.2, je možné vidět výslednou změnu odchylek, při použití pouze náhradního modelu. Kdy tyto odchylky byly vytvářeny na základě, příchozího signálu z

GPS, kdy však tyto odchylky nebyly používány jako vstup do regulátoru ale pouze jako kontrola náhradního modelu.



Obrázek 11.2 Odchylka frekvence mikrokontroleru při použití modelu verze č.1

Jak je vidět, výsledná odchylka se pohybuje okolo nuly, kdy však v určité momenty se vzdálí a následně zase přiblíží k nulové hodnotě. Proto bylo nutné zjistit, zda nějaký další parametr neovlivňuje přesnost RC oscilátoru.

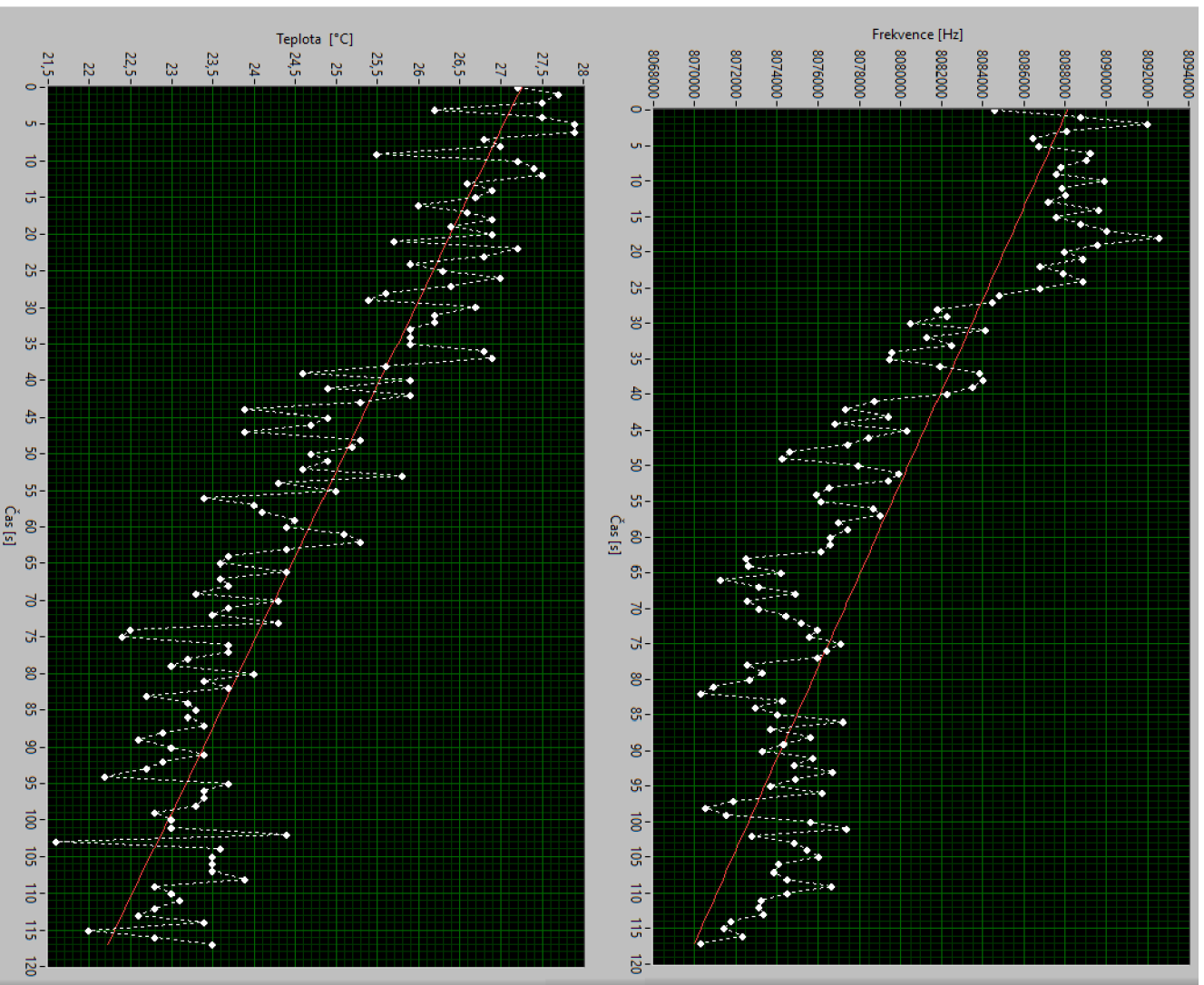
11.2 Druhá verze náhradního modelu

Nejpravděpodobnější parametr, který by takto mohl ovlivňovat RC oscilátor, je teplota. Z toho důvodu, bylo nutné měřit teplotu v okolí oscilátoru, kdy jako teploměr byl zvolen vnitřní teploměr, který je vnitřně připojen na jeden z A/D převodníků. Výhoda toho teploměru, je v umístění na desce, čímž je měřena teplota v přímém okolí RC oscilátoru. Nevýhodou je však velký výstupní šum dat. Z toho důvodu, je vždy teplota měřena třikrát, aby došlo k zprůměrování výsledné teploty.

Proto jako další parametr bylo třeba proměřit o kolik tiků se změni průměrná odchylka, při změně teploty o 1 °C. K tomu proměření došlo, pomocí stejné metody jako byl měřen vliv doby na změnu tiků, kdy však k vyčítaným výsledkům byla přidána i teplota. Naměřené výsledky je poté možné vidět na obrázku 11.3 kdy, jak je vidět, tak opravdu, při poklesu teploty dochází k snížení počtu tiků RC oscilátoru.

Stejně jako tomu bylo při určování vlivu doby na změnu tiků, i zde byla využita metoda nejmenších čtverců, kdy výsledný sklon tiků byl -154,844, zatím co u teploty byl

pokles -0,0429. Z těchto dvou sklonů, lze určit změnu za daný počet sekund, kdy za měřenou dobu došlo ke změně tiku o hodnotu 18271 tiku a u teploty o 5,06 °C. Nyní když podělíme změnu tiku změnou teploty získáme hodnotu 3600 tiku/°C, která nám říká, jak hodně se změni počet tiku za jeden °C.



Obrázek 11.3 Graf zachycující změnu frekvence na základě změny teploty

Nyní však vystává otázka, jakým způsobem by měla tato informace o teplotě zasáhnout do vytváření náhradního modelu. Jako správný přístup se ukázalo, přidání dalšího pole, do kterého byly přidávány údaje o teplotě, stejně tak jako tomu je u hodnot

odchylek. Následně poté při, výpadku dochází k výpočtu průměrné teploty na daném indexu.

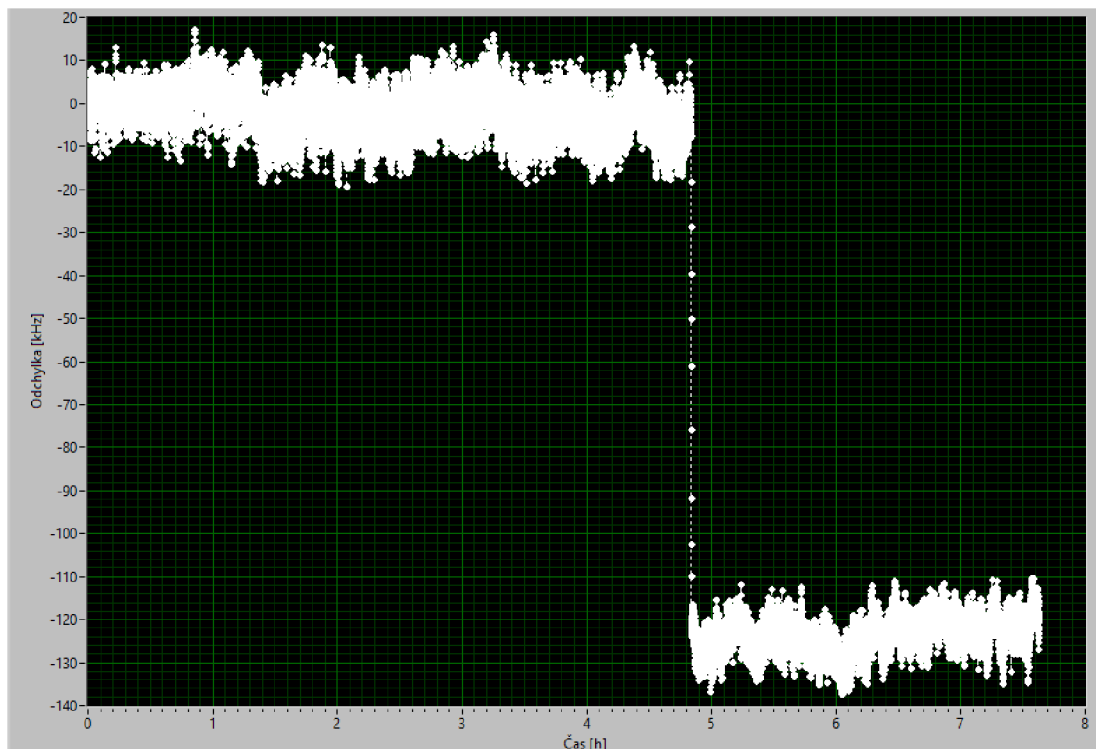
Proto je vždy před určení odchylky vstupující do regulátoru, změřena okolní teplota a následně je pomocí této teploty, průměrné teploty a průměrné odchylky na daném indexu, vypočítána odchylka vstupující do regulátoru.

Vzorec pro výpočet dané odchylky zní následovně

$$e = \bar{e} [index] + (\bar{T} [index] - T) * koeficient , \quad 11.1$$

kdy, e je výsledná odchylka vstupující do regulátoru, $\bar{e} [index]$ je průměrná hodnota odchylky na daném indexu, \bar{T} průměrná teplota na daném indexu, T je aktuální teplota a koeficient hodnota změny tiků za 1 °C.

Naměřený výsledek je možné pozorovat na obrázku 11.4 kdy, jak je vidět, v tomto případě, již zde není problém se změnou odchylky stejně jako tomu bylo u první verze. Ale objevuje se zde problém, kdy se model dostane mimo hodnoty indexu, které byly změřeny v období ukládání dat. To je možné pozorovat v čase 4,8 hodiny, kdy dojde k náhlému propadu frekvence.



Obrázek 11.4 Odchylka frekvence mikrokontroleru při použití modelu verze č.2

11.3 Třetí verze náhradního modelu

V této finální verzi, bude cílem odstranit problém vznikající v přecházející verzi s náhlým propadem frekvence. Toto bude docíleno tak, že hodnoty, které nebyly změřeny, budou

vypočítány na základě znalosti rozdílu teoretických frekvencí, vypočítaných pro look-up tabulku a změřených hodnot pro ostatní indexy.

Vypočítávání těchto indexů, je prováděno ve dvou fázích, kdy je nejprve postupováno od prvního indexu v tabulce, kdy k výpočtu odchylky aktuální indexu, dojde pouze pokud přecházející index byl změřen nebo již byl vypočítán. Pokud tomu tak není, daný index se vynechá a postupuje se dále. Z toho vyplývá, že první index, který je pomocí toho prvního průchodu vypočítán, je index, který je větší než nejmenší změřený index. Od toho indexu, jsou již pak všechny vyšší indexy vypočítány. Kdy je využita rovnice 11.2

$$\bar{e} [index] = \bar{e} [index - 1] - (F [index] - F[index - 1]), \quad 11.2$$

kde $\bar{e} [index]$ je vypočítaná průměrná odchylka, $F [index]$ je teoretická hodnota frekvence na daném indexu.

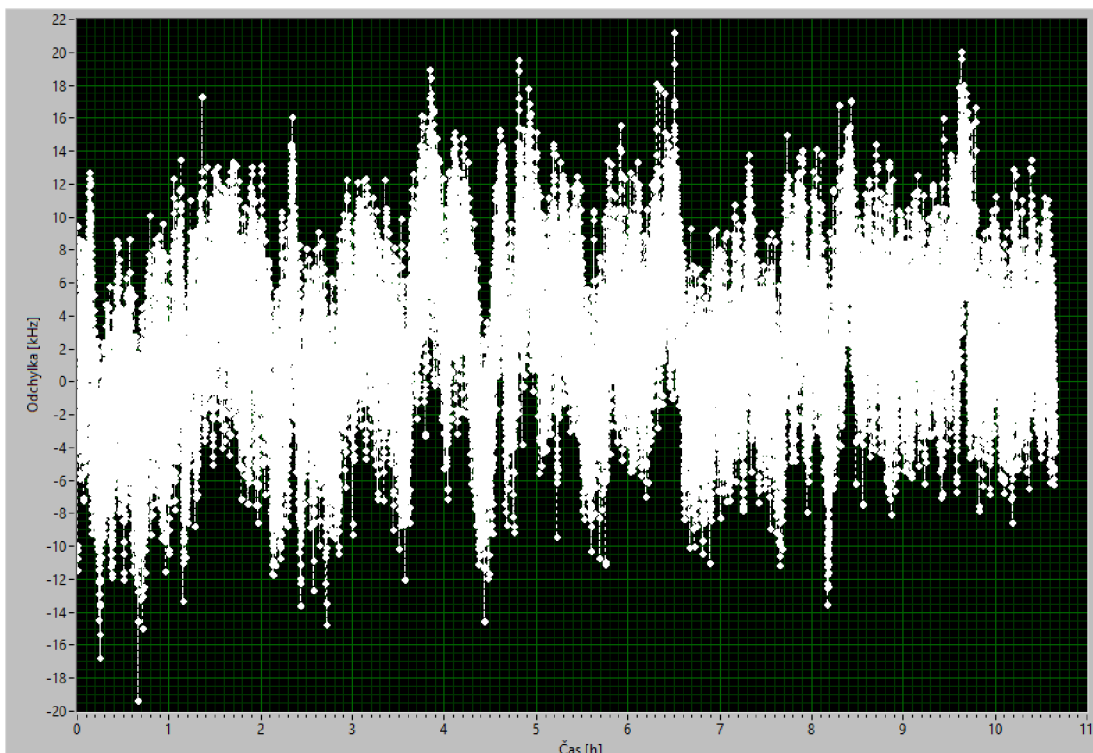
Následně dojde k druhému průchodu, který naopak jde od maximálního indexu dolu. Zde již dojde pouze k výpočtu indexů, které byly menší než nejmenší změřený index. V takovém to případě, je již použita rovnice 11.3

$$\bar{e} [index] = \bar{e} [index + 1] + (F [index + 1] - F[index]), \quad 11.3$$

kde $\bar{e} [index]$ je vypočítaná průměrná odchylka, $F [index]$ je teoretická hodnota frekvence na daném indexu.

Výsledek čehož, dojde k vypočítání průměrné odchylky ve všech indexech look-up tabulky. Jako průměrná teplota, je následně použita průměrná teplota indexu, ze kterého byla průměrná hodnota odchylky vypočítána.

Na obrázku 11.5, je poté vidět výsledná změna odchylek v čase. Kdy, jak je vidět, z počátku odchylka kmitá kolem nulové odchylky, kdy s rostoucí dobou, jak je náhradní model využíván, se již střední hodnota odchylky pomalu zvyšuje, kdy od času 8,5 hodiny je již tato střední hodnota rovna 2 kHz.



Obrázek 11.5 Odchylka frekvence mikrokontroleru při použití modelu verze č.3

11.4 Zhodnocení

Všechny představené modely, jak bylo vidět byly za určitých podmínek funkční. Pokud však tyto podmínky byly porušeny, docházelo k tomu, že náhradní modely nebyly přesné, či se stávaly nefunkčními. Z toho důvodu, byly vždy tyto modely, vylepšovány právě v těchto oblastech, aby byly schopné zvládat i situace, kde jejich předchůdci selhali. Všechny představené modely, byly testovány po příchodu 3600 hodnot, tedy po fungování 1 hodiny.

Patrně by bylo možné i nadále vylepšovat poslední náhradní model, kdy hodnota rozdílu mezi dvěma teoretickými hodnotami je v představeném řešení vždy konstantní. Toto však vždy nemusí platit, a tedy bylo by vhodné pro zpřesnění dat, které nebyly změřeny vypočítat koeficient, kterým by byl tento rozdíl upravován. Tento koeficient by měl být patrně vypočítáván, z indexů, které byly změřeny.

12. SROVNÁNÍ FUNKČNOSTI IMPLEMENTOVANÝCH SYSTÉMŮ

V této kapitole se podíváme na vytvořené metody využívající systémy GPS, RDS a DCF. Kdy hlavním cíle bude ukázat reálné fungování těchto metod. To bude prováděno na základě porovnání změřené frekvence mikrokontroleru v čase. Tyto hodnoty byly měřeny pomocí časovače uvnitř mikrokontroleru, který počítal počet tiků mezi dobami změny nastavení frekvence, tedy stejným způsobem jako je vytvářena zpětná vazba. Následně byla tato informace posílána přes UART komunikaci do počítače, kde byly tyto zprávy čteny a ukládány pomocí programu vytvořeném v LabVIEW.

Pro snazší porovnání výsledků bude od hodnot frekvence odečtena požadovaná hodnota, tedy výsledky budou porovnávány na základě odchylek změřených v čase změny regulované soustavy.

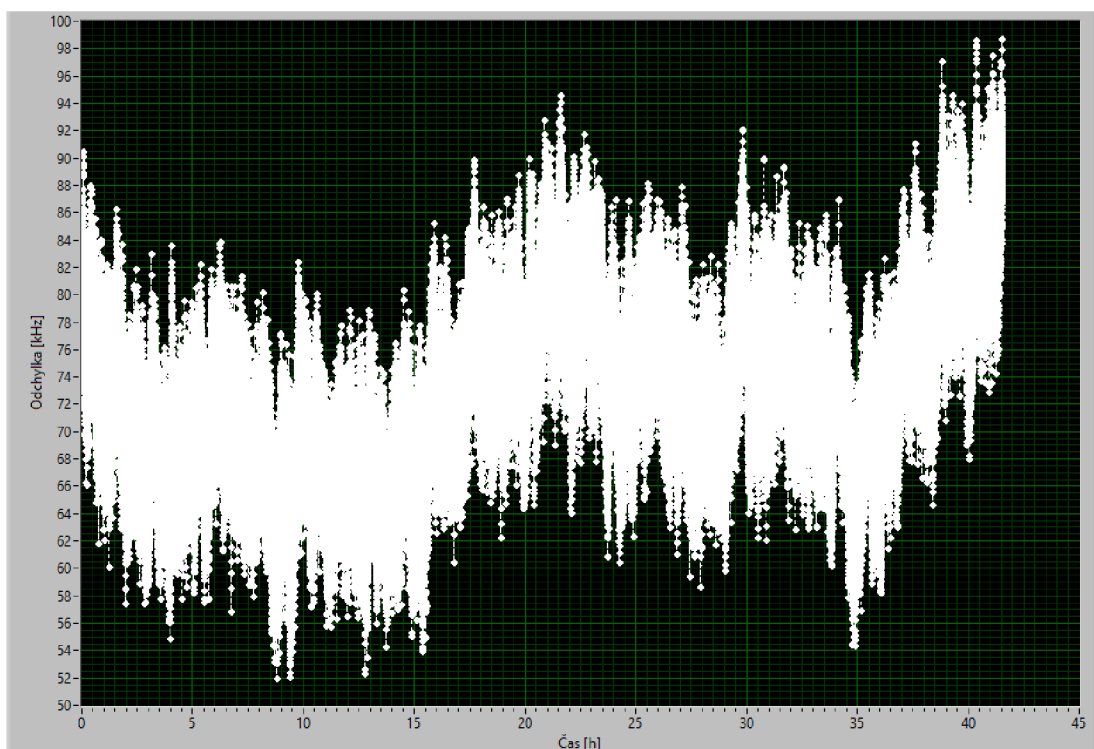
12.1 Neregulovaná frekvence

Jako první se podíváme na chování neregulované soustavy. K tomu, aby mohla být změřena frekvence vnitřních hodin, byla využita GPS. Přičemž, byl využit její výstup PPS, který signalizoval uběhnutí jedné sekundy. Z čehož vyplývá, že hodnota, kterou je třeba odečíst od změřené hodnoty je hodnota 8 MHz.

Výsledné hodnoty odchylek je možné pozorovat na obrázku 12.1 kdy, jak je vidět, tak i když jsou hodiny mikrokontroleru nastaveny na hodnotě 8 MHz, tak výsledná odchylka je posunuta o určitý offset. Ten je patrně způsoben nepřesností vnitřních hodin.

Jak je vidět tato odchylka se pohybuje v rozmezí 50 až 100 kHz, tedy pokud bychom to převedli na procentuální část z nastavené frekvence, tak se odchylka pohybuje v rozmezí 0,625 až 1,25 % z nastavené frekvence. Když toto porovnáme s údaji z datasheetu, které udávají maximální chybu 1 %, při teplotě 25 °C, popřípadě 4 % pro rozsah teplota -10 až 85 °C, je vidět, že měřený RC oscilátor je na maximální povolené hranici. [24]

Jak je dále vidět, tak samotný RC oscilátor je zatížen šumem, který je zhruba o velikosti 10 kHz, čímž budou trpět i výsledné odchylky vytvořených systému. Také je zde vidět, že výsledná odchylka je závislá na teplotě. To zde tedy není zcela přesně měřeno, ale je to možné odhadnout na základě faktu, že měření proběhlo u nezateplených oken, přičemž měření bylo spuštěno zhruba v pět hodin odpoledne, kdy venkovní teplota klesá, a naopak okolo desáté hodiny ranní, (šestnácté hodiny měření), tato teplota opět začne stoupat.

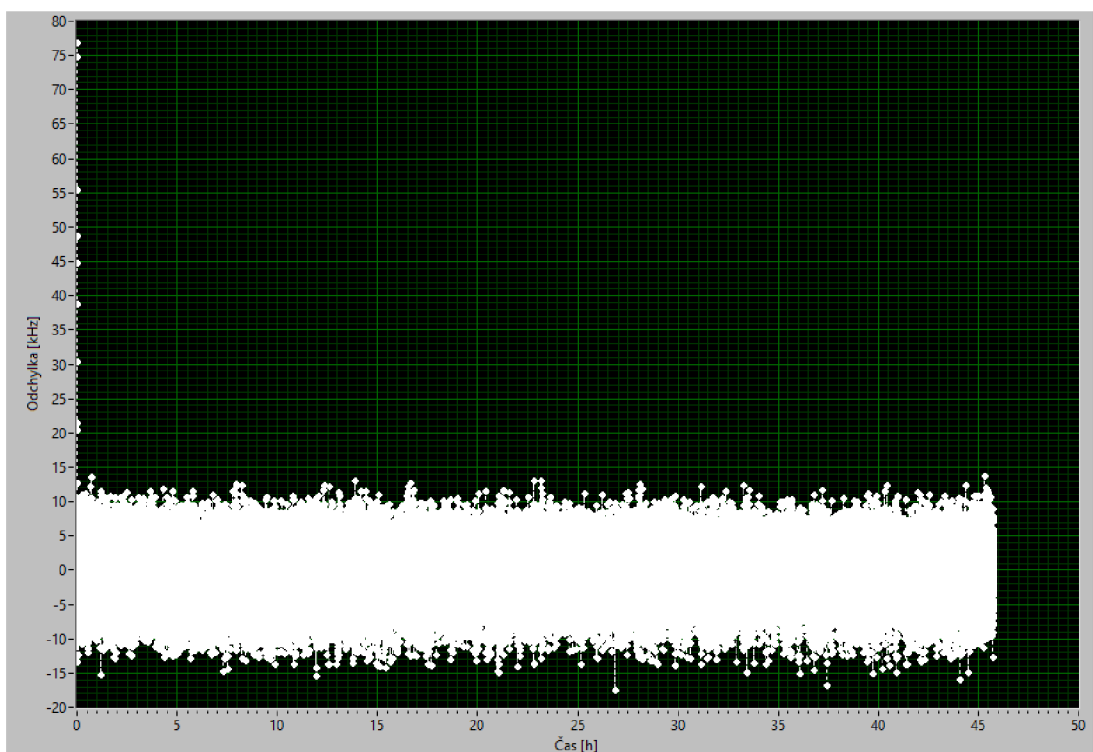


Obrázek 12.1 Změna odchylny v čase u neregulované frekvence

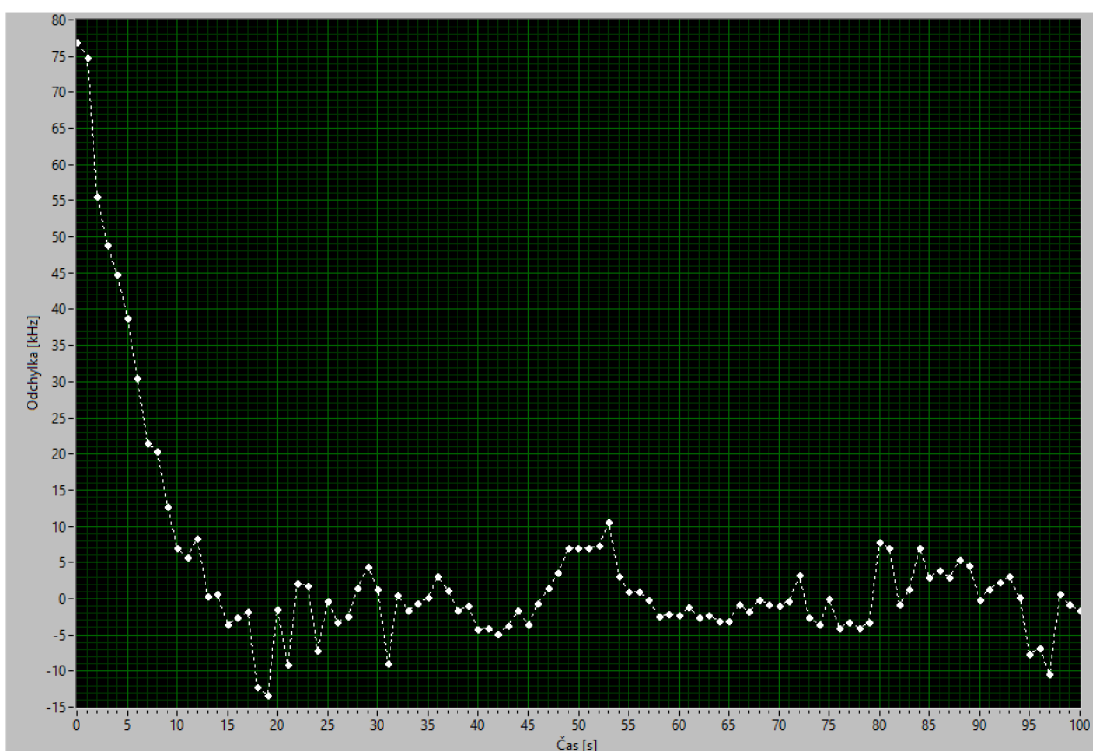
12.2 Regulace frekvence pomocí GPS

Ze změřených dat ohledně GPS, vyplývá, že by tato metoda měla být ta nejpřesnější ze zvolených metod. To dokazuje i výsledný signál změřených odchylek. Jak je vidět na obrázku 12.2, tak naprostá většina změřených odchylek se nachází v rozmezí ± 10 kHz, což jak bylo vidět u neregulované soustavy je zhruba velikost šumu RC oscilátoru, dále je třeba brát zřetel na to, že diskrétní krok je také v rozmezí 3 až 10 kHz. Z toho důvodu, předpokládám, že již není možné zlepšit dané výsledky využitím přesnějšího systému.

Dále je na začátku vidět přechodový děj, kdy je počáteční odchylka o velikosti 77 kHz postupně snižována. Pro získání lepší představy, jak vypadá tento přechodový děj, bylo vyříznuto prvních 100 vteřin regulace do obrázku 12.3. Kdy, jak je vidět, tak přechodový děj se ustálil po uplynutí zhruba 22 sekund od začátku regulace.



Obrázek 12.2 Změna odchyłky v čase u regulované frekvence pomocí GPS

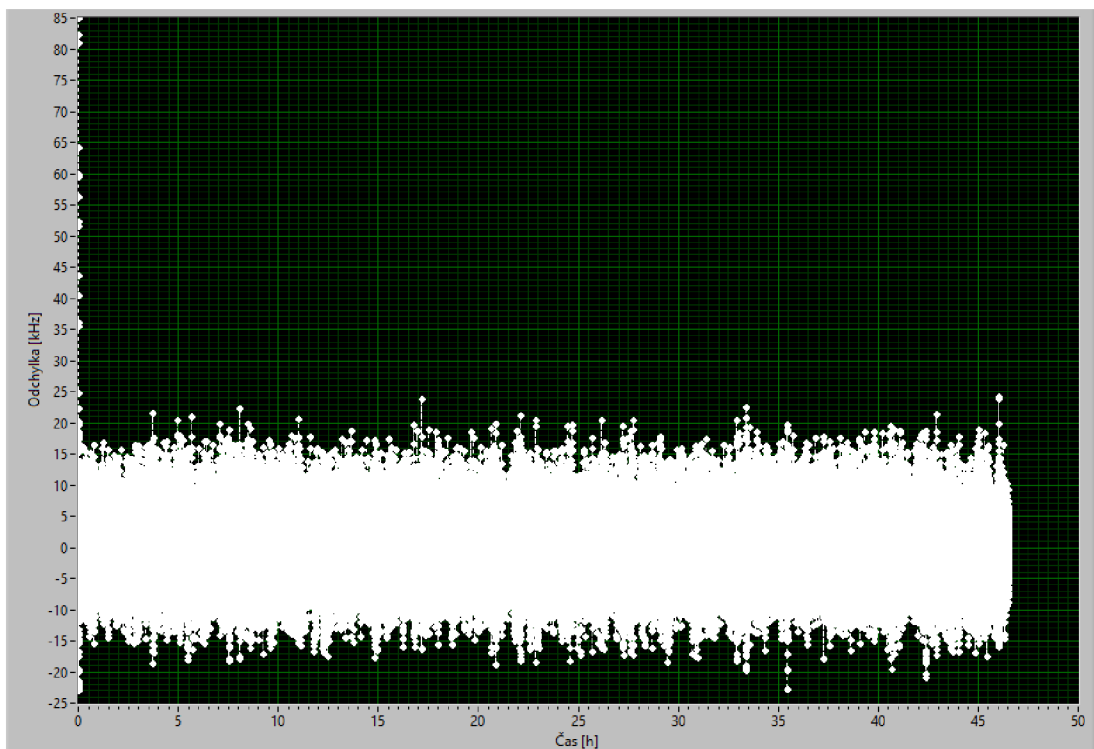


Obrázek 12.3 Přechodový děj u regulace pomocí GPS

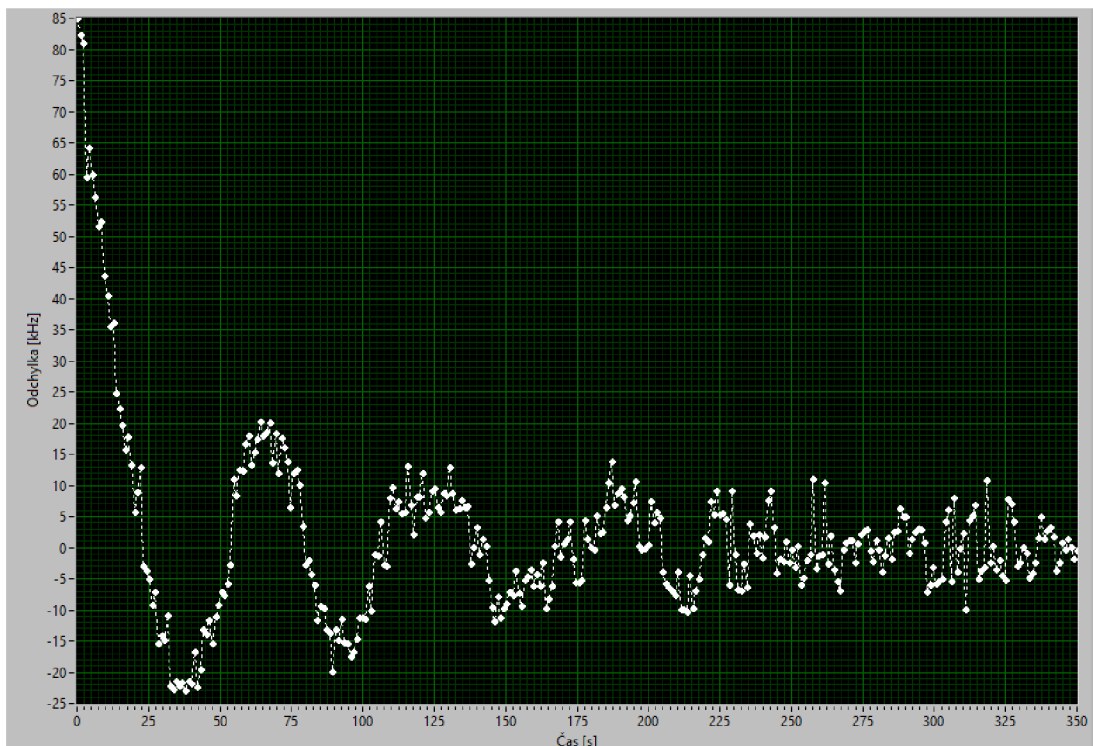
12.3 Regulace frekvence pomocí RDS

Jako další systém je představena regulace pomocí RDS. Zde je oproti, GPS upravena doba mezi regulacemi na 1,051 s. Jak je vidět na obrázku 12.4, tak tento systém již nedosahuje tak nízkého rozsahu odchylek jako GPS systém, přesto je schopen udržet odchylku v rozmezí ± 15 kHz.

Také je zajímavé pozorovat, že přesto, že je použit totožný regulátor, se u této metody změnil přechodový děj. Zatím co u GPS došlo maximální k jednomu kmitu, při této metodě došlo k sedmi kmitům, čímž se doba ustálení soustavy protáhla na 225 sekund.



Obrázek 12.4 Změna odchylky v čase u regulované frekvence pomocí RDS



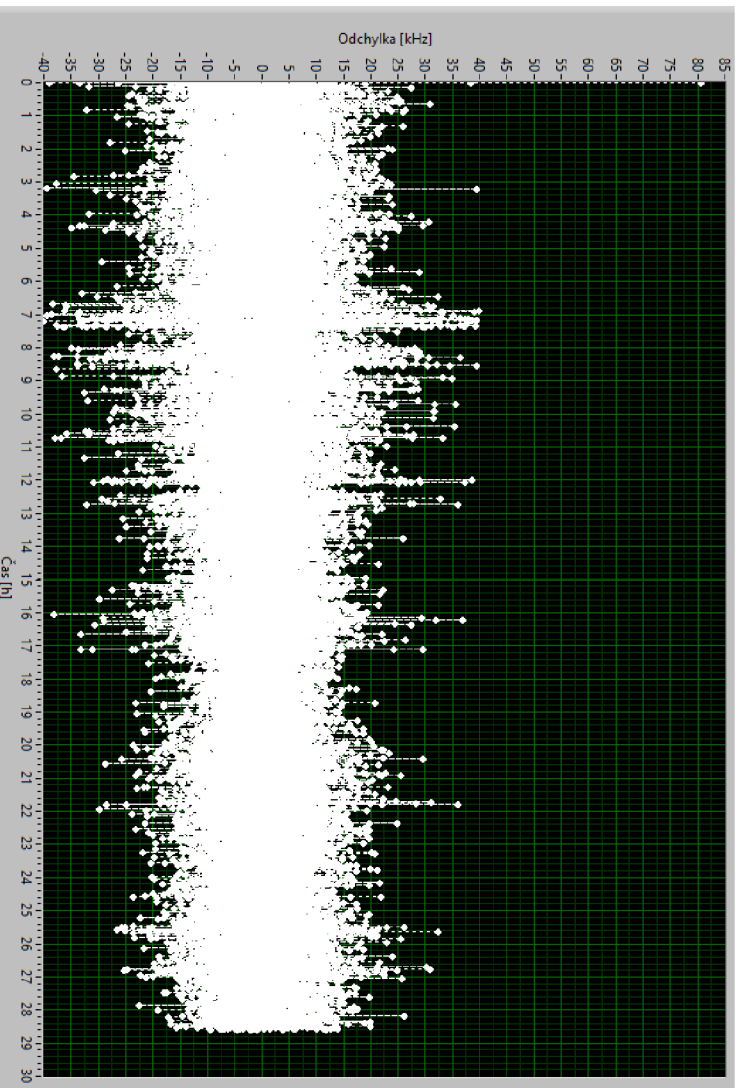
Obrázek 12.5 Přechodový děj u regulace pomocí RDS

12.4 Regulace frekvence pomocí DCF77

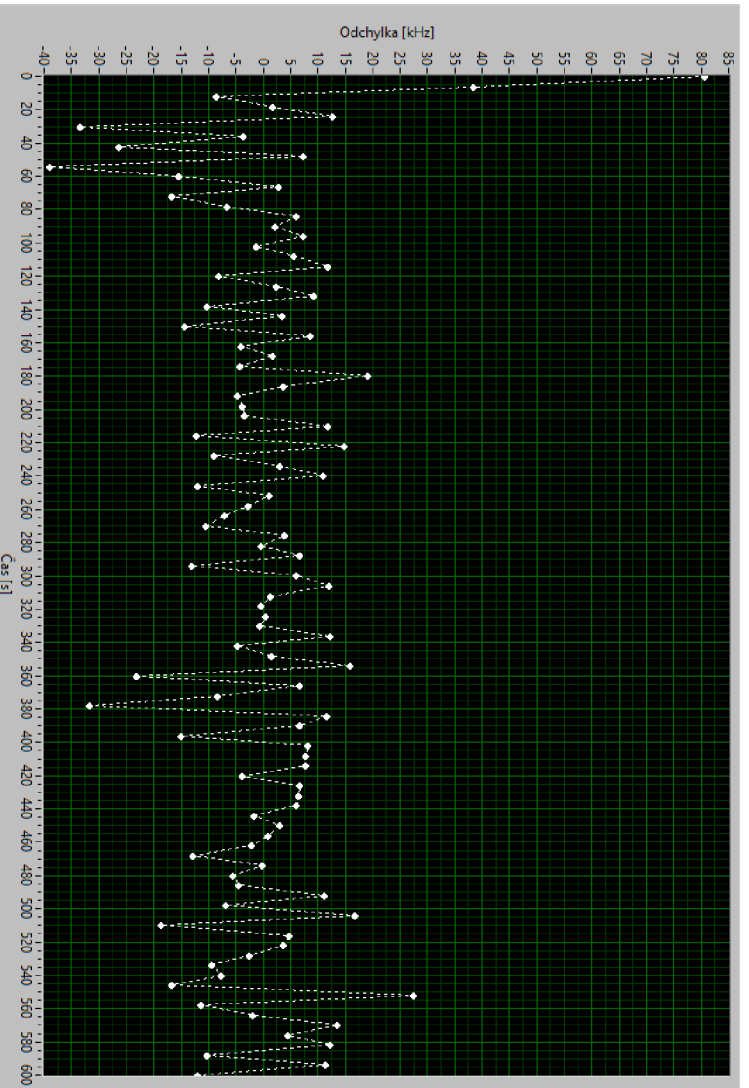
Jako poslední je zde metoda využívající DCF77. Z naměřených dat, vychází, že tato metoda by měla mít nejhorší výsledky. Kdy, jak je vidět na obrázku 12.6, tak tomu tak opravdu je. Zatím co u metody RDS, občasně docházelo k překování hranice 15 kHz, v této metodě, se rozmezí odchylek spíše blíží hodnotě ± 20 kHz.

Dále je možné vidět, že občas je odchylka odskočena na hodnotu 40 kHz, kdy nejčastěji tomu tak je mezi šestou a sedmou hodinou. Přesto, že tato informace nebyla, při měření zkoumána, domnívám se, že tyto hodnoty jsou výsledkem rušení.

Jako poslední zajímavostí je patrně přechodový děj, kdy při této metodě, bylo nutné zvětšit hodnotu anti wind-up, čímž došlo i k zrychlení přechodového děje. Přesto, že časová doba je srovnatelná s GPS, při této metodě regulátoru stačilo provést 4 změny na dosažení ustálené hodnoty.



Obrázek 12.6 Změna odchylnky v čase u regulované frekvence pomocí DCF77



Obrázek 12.7 Přechodový děj u regulace pomocí DCF77

13. SHRNUÍ METOD

V této kapitole se podíváme na porovnání jednotlivých metod. Porovnáme jejich výsledky v prvním a druhém stupni časové synchronizace.

Jako první se zaměříme na první stupeň časové synchronizace, kdy u této části je cílem, aby byly přivedeny synchronizační pulzy v totožný moment. Kdy pro tento stupeň je nejdůležitější parametr časový posun příchodu pulzů z vícero synchronizačních systémů. Výsledky tohoto parametru pro jednotlivé metody je možné pozorovat v tabulce 13.1. Kdy jsou zde uvedeny dva parametry, střední hodnota a směrodatná odchylka, přičemž oba tyto parametry bychom v ideální případě chtěli mít nulové.

Nyní je však potřeba rozhodnout, podle jakého z parametrů by bylo vhodné porovnat jednotlivé metody. Na první pohled se totiž může zdát, že důležitější parametrem je střední hodnota, která přesně říká o kolik se synchronizační pulzy budou od sebe lišit. Ovšem tento parametr lze kompenzovat softwarově uvnitř mikrokontroleru, pomocí offsetu. V takový to moment by již záleželo více na směrodatné odchylce než na střední hodnotě.

Z tohoto důvodu byla zvolena jako porovnávací parametr právě zmíněná směrodatná odchylka. Přičemž pokud se povídám na výsledky v tabulce 13.1, tak je vidět, že nejlepší metodou je metoda využívající synchronizaci z nadřazeného systému. Kdy důvodem, proč tomu tak je pramení pravděpodobně z toho, že se signál přenáší přes vodič, nikoliv vzduchem. Pokud budeme brát pouze bezdrátové přenosy tak, zde se nejlépe ukázala metoda využívající GPS. Dále v pořadí je metoda využívající RDS data, kdy její přesnost je horší zhruba o 3 řády. Nejhoršími výsledky se nakonec prezentuje DCF77, kdy její přesnost je v řádu desítek ms. Tedy opět zhruba o 3 řády horší než metoda s RDS daty.

Tabulka 13.1 Porovnání posunu signálu mezi jednotlivými mikrokontrolery pro jednotlivé metody

Metoda	Střední hodnota [μs]	směrodatná odchylka [μs]
Bez vytížení MCU	10,6045	0,0619
I2C MCU	10,5996	0,0626
Matematika MCU	10,5967	0,0666
Prodloužený vedení	14,4497	0,1073
Zpětná smyčka	3,9830	0,0362
GPS přijímače	-0,0319	0,1079
GPS přes MCU	-0,1099	0,3490
RDS přes Agilent 53131A	21,4	227,2
RDS přes Agilent 53131A, avg. 12	21,445	45,222
DCF77	11555	4572
DCF77, avg. 6	15303	2811

Jako další bod, je druhý stupeň časové synchronizace, kdy při tomto bodu dochází k regulaci frekvence mikrokontroleru, s cílem dosáhnout její stability v čase. K proměření toho parametru byla využita metoda Allanovy odchylky, která má za cíl zobrazit v grafu právě stabilitu frekvence v čase. Další parametr, který je pro porovnání této metody důležitý je perioda synchronizačního signálu, kdy hlavním parametrem je směrodatná odchylka daných period. Neboť právě směrodatná odchylka zanáší chybu do regulátoru. To může způsobit, že signál, který má chodit po 1 sekundě, je pozdržen např. o jednu milisekundu, což vede k tomu, že na čítači dojde k napočítání počtu tiků za 1,001 s, čímž je ovlivněna zpětná vazba.

Změřené výsledky period signálu je možné pozorovat v tabulce 13.2 kdy, jak je vidět, tak nejmenší směrodatná odchylka je u metody GPS, kdy chyba způsobená směrodatnou odchylkou by se pohybuje v hodnotách desítek nanosekund. Jako druhá metoda se umístila metoda využívající RDS data, kdy její směrodatná odchylka se pohybuje v řádu desítek μ s. A jako nejhorší metoda se ukázala metoda využívající signál od vysílače DCF77. Kdy její směrodatná odchylka se pohybuje v řádu jednotek milisekund, bez využití průměrování více hodnot. Při využití průměrování, se tato metoda posune o jeden řád níže.

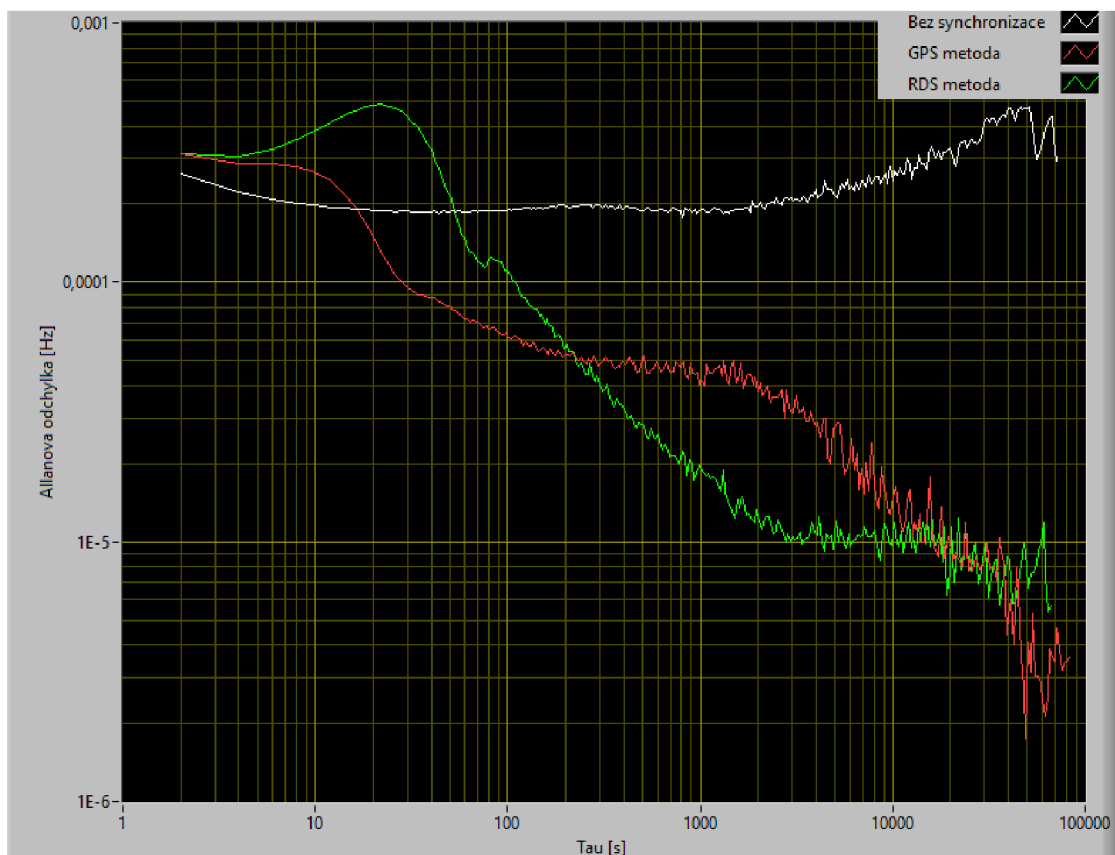
Tabulka 13.2 Porovnání period signálu pro jednotlivé metody

Metoda	Střední hodnota [ms]	směrodatná odchylka [ms]
GPS	999,999989	0,000011
RDS měřený pomocí Agilent 53131A	87,579	0,239
RDS měřený pomocí GPS	87,578	0,259
RDS měřený pomocí Agilent 53131A, avg. 12	87,5790	0,0311
RDS měřený pomocí GPS, avg.12	87,5795	0,0203
DCF77	999,991	3,473
DCF77, avg. 6	999,9909	0,7064

Nyní pokud porovnáme změřené Allanovy odchylky pro jednotlivé metody, je vidět, že výsledky korelují s výsledky směrodatných odchylek jednotlivých metod. To je možné pozorovat na obrázku 13.1. Jak je vidět, tak využitím implementovaných metod opravdu dojde k zvýšení stability a přesnosti frekvence. Pokud bychom chtěli porovnat, jednotlivé metody mezi sebou, tak patrně lepších výsledků by dosáhla metoda využívající GPS, neboť je vidět, že na konci stále dochází k poklesu signálu, zatím co u metody RDS, je již dosaženo maximální stability.

Ovšem, hlavním důvodem, proč mají tyto dvě metody tak velmi blízké výsledky bude pravděpodobně omezený diskretní krok frekvencí při regulaci. Když vezmeme nejmenší a největší diskretní krok, jejichž velikosti jsou 2 a 10 kHz a srovnáme to s požadovanou hodnotou 8 MHz, tak zjistíme, že hardwarové omezení je v rozmezí 0,025 až 0,125 % z nastavené frekvence. Z toho plyne, že při měření periody 1 s, se chyba tímto krokem

bude pohybovat v řádu stovek μs . Když toto porovnáme se směrodatnými odchylkami period jednotlivých signálu, tak je vidět, že přesnost GPS je úplně v jiných řádech, zatím co přesnost RDS dat, se pohybuje v okolí této hardwarové přesnosti.



Obrázek 13.1 Porovnání Allanových odchylek

Předcházející výsledky dokazují i výsledky z kapitoly 12, kdy v této kapitole byly ukázány, výsledné odchylky frekvence v čase, při využití regulátoru. Z těchto výsledků je možné říci, že pořadí v přesnosti metod je stejné jako tomu je u Allanovy odchylky. Kdy metoda využívající GPS, je již spíše omezena vlastnostmi RC oscilátoru než vlastní chybou. Kdy vysvětlení proč tomu tak je, je možné nalézt v kapitole 12.2.

14. ZÁVĚR

Diplomová práce je orientovaná na implementaci časové synchronizace pro mikrokontrolery a následné otestování implementovaných metod. Přičemž tato práce se zaměřila na první dva stupně časové synchronizace.

Stejně jako tomu bylo v kapitole 13, tak i zde budou výsledky u první metody časové synchronizace porovnávány na základě velikosti směrodatné odchylky. Kdy, jak ukázala tabulka 13.1, tak nejlépe z implementovaných metod si stála metoda využívající synchronizaci z nadřízeného systému. Důvod tomu, je však to, že daná metoda jako jediná využívá k přenosu drátovou komunikaci. Naopak ostatní metody využívají ke své synchronizaci bezdrátovou komunikaci. Pokud se tedy podíváme na výsledky pouze bezdrátových metod, tak si nejlépe vedla metoda využívající GPS, tato metoda měla velikost směrodatné odchylky v řádu stovek ns. To je o tři řády lepší než druhá metoda využívající RDS data z FM radií. Odchylka této metody byla 45,222 μ s. Jako nejhorší metoda se ukázala metoda využívající DCF77, kdy z byla výsledná odchylka rovna 2,811 ms.

Jako druhý bod, je zde časová synchronizace druhého stupně, kdy u toho stupně je cílem udržet stabilní frekvenci RC oscilátoru. K tomu bylo třeba vytvořit regulační smyčku, kdy jako snímač ve zpětné vazbě byl využit časovač, který počítal počet tiků v daném období.

Výsledky druhé stupně se dají rozdělit na dvě části, a to na měření periody synchronizačního signálu a měření stability RC oscilátoru, kdy k tomu byla využívána Allanova odchylka. V tabulce 13.2, jsou vidět výsledky měření period jednotlivých metod. Kdy ve výsledcích je zahrnuta střední hodnota a směrodatná odchylka. Zde stejně jako tomu bylo u první metody, nás více zajímá směrodatná odchylka. Pokud porovnáme jednotlivé metody mezi sebou tak je vidět, že nejlepší z metod je GPS metoda. Která má velikost směrodatné odchylky v řádu desítek ns. Jako druhá metoda se umístila metoda využívající RDS data, kdy její směrodatná odchylka je rovna 259 μ s, kdy u této metody byla následně využita funkce průměrování více hodnot, čímž se její odchylka zmenšila na 20,3 μ s. A jako nejhorší metoda se umístila opět metoda z DCF77. Kdy její směrodatná odchylka se pohybuje v řádu jednotek ms. Při přidání průměrování se nakonec její směrodatná odchylka také zmenšila, a to na hodnotu 0,7064 ms.

Druhá část, toho stupně je pak stabilita RC oscilátoru. Kdy ta byla měřena po dobu 4 dnů. Výsledná stabilita je poté zanesena do grafu 13.1, kde jsou zobrazeny všechny naměřené Allanovy odchylky. Zde je vidět, že dané metody opravdu zlepšují výslednou stabilitu oscilátoru, oproti neregulovanému systému. Opět je možné říci, že nejlepší metodou je GPS, kdy však rozdíl oproti DCF metodě, není nikterak velký, což je pravděpodobně způsobeno tím, že daný mikrokontroler byl hardwarově omezen v nastavování frekvencí.

Předcházející výsledky dokazují i výsledky z kapitoly 12, kdy v této kapitole byly ukázány, výsledné odchylky frekvence v čase, při využití regulátoru. Z těchto výsledků je možné říci, že pořadí v přesnosti metod je stejné jako tomu je u Allanovy odchylky. Kdy metoda využívající GPS, je již spíše omezena vlastnostmi RC oscilátoru než vlastní chybou. Kdy vysvětlení proč tomu tak je, je možné nalézt v kapitole 12.2.

Jako poslední vylepšení implementovaných metod, bylo vytvoření náhradního modelu, který bude využíván pro regulaci v případě ztráty synchronizačních pulzů. Náhradní model je vytvářen, při každé výpadku znovu z historie naměřených dat. Přičemž bylo prokázáno, že se hodnota frekvence RC oscilátoru v krátkém čase příliš nemění, z toho důvodu není doba běhu náhradního modelu využívána k změně parametrů náhradního modelu. Jinak tomu je však u teploty, kdy bylo prokázáno, že teplota mění hodnotu frekvence RC oscilátoru výrazně a to 3600 tiků za °C.

Tato diplomová práce ukázala, že využitím metod pro časovou synchronizaci, lze zlepšit vlastnosti méně kvalitních RC oscilátoru uvnitř mikrokontrolerů. Kdy pro jejich zlepšení, lze využít levné moduly pro příjem synchronizačních signálů. Dále zde byla vyzkoušena možnost vytvoření přesného měřicího systému, který by využíval pouze mikrokontroler a modul GPS. Kdy, jak bylo dokázáno v kapitole 9.3.3, tak výsledky získané pomocí daného měřicího systému jsou srovnatelné s výsledky získané pomocí čítače Agilent 53131A.

LITERATURA

- [1] C. RHEINLÄNDER, Carl a Norbert WEHN. Precise synchronization time stamp generation for Bluetooth low energy. IEEE [online]. 2016, 2016, 1-5 [cit. 2023-10-25]. Dostupné z: <https://ieeexplore.ieee.org/document/7808812>
- [2] Amirhossein YOUSEFI, Kasun SOMARATNE a F. JOHN DIAN. Analysis of time synchronization based on current measurement for Bluetooth Low Energy (BLE). IEEE [online]. 2017, 2017, 1-5 [cit. 2023-10-25]. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/8117157>
- [3] Networklessons [online]. 2013–2023 [cit. 2023-12-27]. Dostupné z: <https://networklessons.com/cisco/ccnp-encor-350-401/introduction-to-precision-time-protocol-ntp>
- [4] Dpdk [online]. [cit. 2023-12-27]. Dostupné z: https://doc.dpdk.org/guides/sample_app_ug/ptpclient.html
- [5] SEBESTA, Jiri a Pavel VYSKOCIL. Relative timing characteristics of GPS timing modules for time synchronization application. IEEE [online]. 2009, 2009, 1-4 [cit. 2023-10-25]. Dostupné z: <https://ieeexplore.ieee.org/document/5286378>
- [6] NIEMANN, Lars, Marcus VENZKE, Christian RENNEN a Volker TURAU. Clock Synchronization of TinyOS-based Sensor Networks with DCF77 [online]. 1-2 [cit. 2023-10-30]. Dostupné z: https://www.ti5.tuhh.de/events/fgsn09/proceedings/fgsn_045.pdf
- [7] DCF77. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2023-12-31]. Dostupné z: <https://en.wikipedia.org/wiki/DCF77>
- [8] PTB [online]. [cit. 2023-10-30]. Dostupné z: <https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/ag-442/dissemination-of-legal-time/DCF77/DCF77-amplitude-modulation.html>
- [9] LI, Liqun, Limin SUN, Guoliang XING, Wei HUANGFU, Ruogu ZHOU a Hongsong ZHU. ROCS: Exploiting FM Radio Data System for Clock Calibration in Sensor Networks. IEEE [online]. 2014, 2014, 1-9 [cit. 2023-10-30]. Dostupné z: <https://ieeexplore.ieee.org/document/6987310/citations#citations>
- [10] Mbed [online]. [cit. 2023-11-03]. Dostupné z: <https://os.mbed.com/platforms/ST-Nucleo-F303K8/>
- [11] Mbed [online]. [cit. 2024-04-16]. Dostupné z: <https://os.mbed.com/platforms/ST-Discovery-F769NI/>
- [12] Digilent [online]. [cit. 2023-11-03]. Dostupné z: <https://digilent.com/reference/test-and-measurement/analog-discovery-2/start>
- [13] Keysight [online]. 2000–2023 [cit. 2023-12-30]. Dostupné z: <https://www.keysight.com/us/en/assets/7018-06688/data-sheets/5967-6039.pdf>
- [14] Allan variance. In: Allan variance [online]. [cit. 2023-11-20]. Dostupné z: <https://home.engineering.iastate.edu/~shermanp/AERE432/lectures/Rate%20Gyros/Allan%20variance.pdf>

- [15] Allan variance. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2023-11-20]. Dostupné z: https://en.wikipedia.org/wiki/Allan_variance#
- [16] Botland [online]. 2023 [cit. 2023-12-30]. Dostupné z: <https://botland.cz/stm32-nucleo/4891-stm32-nucleo-f303k8-stm32f303k8-arm-cortex-m4-5904422362119.html>
- [17] Distrelec [online]. 2023 [cit. 2023-12-30]. Dostupné z: <https://www.distrelec.cz/cs/usb-osciloskop-logicky-analyzer-variabilni-napajeci-zdroj-analog-discovery-100-ms-digilent-410-321/p/30044243>
- [18] Drátek.cz [online]. [cit. 2023-12-31]. Dostupné z: https://dratek.cz/arduino/1510-gps-neo-6m-gyneo6mv2-modul-s-antenou.html?gad_source=1&gclid=CjwKCAiA1fqrBhA1EiwAMU5m_4AU5jRKVw6OUHKDx1Y5U1Ow2nkyx-EY__B9asqmCxwpPXyFS9P4KxoC6IoQAvD_BwE
- [19] Agilent 53131A Datasheet [online]. [cit. 2024-05-07]. Dostupné z: <https://www.keysight.com/us/en/assets/7018-06688/data-sheets/5967-6039.pdf>
- [20] Avmarket [online]. 2023 [cit. 2023-12-30]. Dostupné z: <https://www.avmarket.eu/modul-neo-6m-gps-6mv2-red-anteny-p-7085.html>
- [21] Keysight [online]. 2023 [cit. 2023-12-30]. Dostupné z: <https://www.keysight.com/us/en/support/53131A/225-mhz-universal-frequency-countertimer.html>
- [22] Drátek.cz [online]. [cit. 2024-04-03]. Dostupné z: https://dratek.cz/arduino/1492-fm-rds-radio-si4703-modul-tuneru-pro-avr-arm-pic.html?gad_source=1&gclid=CjwKCAjw_LOwBhBFEiwAmSEQAYyJiuFoszy-wvcMLwrluRNnNeB_aCjJGHnJPkfcCwrOXG3PzbJjjBoCpPIQAvD_BwE
- [23] Allegro.cz [online]. [cit. 2024-04-08]. Dostupné z: <https://allegro.cz/nabidka/modul-radioveho-prijimace-dcf77-radiovy-modul-s-13912354753>
- [24] STM32 [online]. [cit. 2024-04-17]. Dostupné z: <https://www.st.com/resource/en/datasheet/dm00273119.pdf>

SEZNAM PŘÍLOH

PŘÍLOHA A - LOOK-UP TABULKA	86
PŘÍLOHA B - NAMĚŘENÁ DATA A ZDROJOVÉ KÓDY	88

Příloha A - Look-UP tabulka

i	PLLM	PLLN	PLLP	ABH	Počet tiků
0	15	351	6	8	7800000
1	12	281	6	8	7805555
2	13	203	4	8	7807692
3	14	328	6	8	7809523
4	16	375	6	8	7812500
5	9	211	6	8	7814814
6	11	172	4	8	7818181
7	13	305	6	8	7820512
8	14	219	4	8	7821428
9	15	352	6	8	7822222
10	8	188	6	8	7833333
11	16	251	4	8	7843750
12	15	353	6	8	7844444
13	11	259	6	8	7848484
14	10	157	4	8	7850000
15	9	212	6	8	7851851
16	16	377	6	8	7854166
17	14	220	4	8	7857142
18	12	283	6	8	7861111
19	11	173	4	8	7863636
20	10	236	6	8	7866666
21	13	307	6	8	7871794
22	8	126	4	8	7875000
23	11	260	6	8	7878787
24	14	331	6	8	7880952
25	13	205	4	8	7884615
26	9	142	4	8	7888888
27	14	221	4	8	7892857
28	16	379	6	8	7895833
29	13	308	6	8	7897435
30	14	332	6	8	7904761
31	16	253	4	8	7906250
32	11	174	4	8	7909090
33	15	356	6	8	7911111
34	8	190	6	8	7916666
35	9	214	6	8	7925925
36	14	222	4	8	7928571
37	10	238	6	8	7933333

38	8	127	4	8	7937500
39	11	262	6	8	7939393
40	9	143	4	8	7944444
41	13	310	6	8	7948717
42	10	159	4	8	7950000
43	14	334	6	8	7952380
44	11	175	4	8	7954545
45	15	358	6	8	7955555
46	13	207	4	8	7961538
47	9	215	6	8	7962962
48	14	223	4	8	7964285
49	10	239	6	8	7966666
50	16	255	4	8	7968750
51	11	263	6	8	7969696
52	12	287	6	8	7972222
53	13	311	6	8	7974358
54	14	335	6	8	7976190
55	15	359	6	8	7977777
56	16	383	6	8	7979166
57	8	128	4	8	8000000
58	16	385	6	8	8020833
59	15	361	6	8	8022222
60	14	337	6	8	8023809
61	13	313	6	8	8025641
62	12	289	6	8	8027777
63	11	265	6	8	8030303
64	16	257	4	8	8031250
65	10	241	6	8	8033333
66	14	225	4	8	8035714
67	9	217	6	8	8037037
68	13	209	4	8	8038461
69	8	193	6	8	8041666
70	11	177	4	8	8045454
71	10	161	4	8	8050000

i – index v tabulce

PLLM – První dělička frekvence v PLL

PLLN – Násobička frekvence v PLL

PLLP – Druhá dělička frekvence v PLL

ABH – Vnitřní dělička frekvence

Počet tiků – Výsledný teoretický počet tiků za sekundu neboli frekvence RC oscilátoru

Příloha B - Naměřená data a zdrojové kódy

Příloha B – Naměřená data a zdrojové kódy jsou uložena na přiloženém disku

--- Priloha_B

--- LabVIEW

--- Uart.vi

--- mereni_periody.vi

--- NamerenaData

--- DataZKapitoly6.tdms

--- DataZKapitoly7.tdms

--- DataZKapitoly8.tdms

--- DataZKapitoly9.tdms

--- DataZKapitoly10.tdms

--- DataZKapitoly11.tdms

--- DataZKapitoly12.tdms

--- STM32

--- Allanovka_bez_synchronizace

--- Core

--- Src

--- main.c

--- DCF77_Posun

--- Core

--- Src

--- main.c

--- DCF77_Synchronizace_Allanovka

--- Core

--- Src

--- main.c

--- GPS_Synchronizace

--- Core

--- Src

--- main.c

--- RDS_Posun

--- Core

--- Src

--- main.c

--- RDS_Synchronizace

--- Core

--- Src

--- main.c