



Zadání bakalářské práce

Název:	Primární NTP server na Raspberry PI
Student:	Jan Wijnhorst
Vedoucí:	RNDr. Ing. Vladimír Smotlacha, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Implementujte primární NTP server v systému Raspberry PI. Jako externí zdroj přesného času použijte atomové hodiny typu Symetricom (Microsemi) 5071A, se kterými bude server komunikovat pomocí sériové linky a rozhraní GPIO. Server ověřte nejprve se simulátorem hodin a následně v reálném provozu. Vyhodnoťte parametry serveru prezentované programem ntpq, zejména odchylku časového údaje a frekvenční stabilitu, a porovnejte je s parametry stejného serveru bez využití atomových hodin.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Primární NTP server na Raspberry PI

Katedra počítačových systémů

Vedoucí práce: RNDr. Ing. Vladimír Smotlacha, Ph.D.

5. ledna 2022

Poděkování

Děkuji svému vedoucímu RNDr. Ing. Vladimíru Smotlachovi, Ph.D. za ochotu, trpělivost, odborné vedení a neocenitelné připomínky, děkuji technickému oddělení sdružení CESNET za poskytnutí prostor.

Prohlášení

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

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

V Praze dne 5. ledna 2022

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Jan Wijnhorst. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Wijnhorst, Jan. *Primární NTP server na Raspberry PI*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

Tato práce se zabývá tvorbou komunikační vrstvy mezi časovým serverem a atomovými hodinami Symmetricom (Microsemi) 5071A. Vytvořený systém byl instalován do uzlu sítě CESNET. Následně byla provedena měření pro vyhodnocení vlivu atomových hodin na frekvenční stabilitu a časovou odchylku časového serveru. Výsledná zpracování dat ukázala významný vzrůst ve stabilitě a pokles v časové odchylce při použití atomových hodin.

Klíčová slova Network Time Protocol (NTP) server, Raspberry PI, atomové hodiny, césiový oscilátor, sériová linka, Pulse Per Second (PPS), General-purpose input/output (GPIO)

Abstract

This thesis deals with the creation of a communication layer between time server and the Symetricom (Microsemi) 5071A atomic clock. The created system was installed in the CESNET network node. Subsequently, measurements were performed to evaluate the influence of the atomic clock on the frequency stability and time offset of the time server. The resulting data processing showed a significant increase in stability and a decrease in time deviation while using the atomic clock.

Keywords Network Time Protocol (NTP) server, Raspberry PI, Atomic clock, Cesium oscillator, Serial port, Pulse Per Second (PPS), General-purpose input/output (GPIO)

Obsah

Úvod	1
1 Cíl práce	3
1.1 Získávání dat	3
1.2 Administrace PFS	3
1.3 Fyzické vlastnosti	4
1.4 Simulátor	4
1.5 Vyhodnocení parametrů	4
2 Analýza a návrh	5
2.1 Výběr NTP služby	5
2.1.1 Referenční implementace NTP	6
2.1.2 Server <code>chrony</code>	6
2.1.3 Server <code>ntpsec</code>	6
2.2 Interpretace PPS signálu	7
2.3 Komunikace s <code>ntpsec</code>	7
2.4 Přestupná sekunda	7
3 Realizace	9
3.1 Vývoj simulátoru	9
3.1.1 Konzultace manuálu	9
3.1.1.1 Formát CLI	9
3.1.2 Proces výroby generátoru PPS	10
3.1.3 Synchronizace	11
3.2 Struktura programu	11
3.2.1 Typy dotazů SCPI	13
3.2.2 Okamžik dotazu	13
3.2.3 Inicializační fáze	14
3.3 Komponenty	15
3.3.1 Konfigurace	15

3.3.1.1	Položka <code>trust_source</code> a <code>max_local_delta</code> . . .	16
3.3.1.2	Položka <code>sync_date</code>	16
3.3.1.3	Položka <code>retires</code>	16
3.3.1.4	Položka <code>shm_id</code>	16
3.3.2	Komunikace po sériové lince	17
3.3.2.1	Formát data	17
3.3.3	Propagace času	17
3.3.4	LCD display	18
3.3.5	HTTP server	18
3.3.6	Detekce PPS signálu	20
3.3.7	Event bus	20
3.4	Konfigurace serveru <code>ntpsec</code>	20
3.4.1	Dotazy <code>ntpq</code>	20
3.4.2	Zabezpečení a firewall	21
3.5	Fyzická část	21
3.5.1	Instalace	23
4	Analýza výsledků	25
4.1	Vyhodnocení parametrů serveru	25
4.2	Měření v uživatelském prostředí	25
4.2.1	Metodika experimentu	25
4.2.2	Parametry testovaných serverů	26
4.2.3	Označení serverů	27
4.2.4	Očekávané výsledky	27
4.2.5	Prezentace dat	28
4.2.6	Odchylka času	29
4.2.7	Absolutní odchylka	29
4.2.8	Krátkodobá frekvenční stabilita	31
4.2.9	Dlouhodobá frekvenční stabilita	32
4.2.10	RMS jitter	33
4.3	Měření v laboratorním prostředí	34
	Závěr	37
	Literatura	39
	A Seznam použitých zkratk	41
	B Obsah příloženého CD	43

Seznam obrázků

3.1	Grafitový dělič napětí – vlastní tvorba	11
3.2	Časový diagram programu – vlastní tvorba	12
3.3	Webové rozhraní – vlastní tvorba	19
3.4	PFS Symetricom 5071A – vlastní tvorba	21
3.5	Návrh předního panelu – vlastní tvorba	22
3.6	Provedení předního panelu – vlastní tvorba	23
3.7	Instalace v uzlu sítě CESNET – vlastní tvorba	23
4.1	Naměřené odchylky času – vlastní tvorba	28
4.2	Očištěné odchylky času – vlastní tvorba	31
4.3	Průměry odchylek – vlastní tvorba	32
4.4	Očištěný RMS jitter – vlastní tvorba	33
4.5	Frekvenční stabilita – CESNET [1]	34
4.6	Časové odchylky – CESNET [1]	35

Seznam tabulek

3.1	Seznam konfiguračních možností – vlastní tvorba	15
4.1	Parametry NTP serverů – vlastní tvorba	26
4.2	Seznam zkratk NTP serverů – vlastní tvorba	27
4.3	Rozdíl časových odchylek – vlastní tvorba	30
4.4	Rozptyl časové odchylky – vlastní tvorba	32

Úvod

Tato bakalářská práce popisuje proces tvorby a následnou analýzu časového serveru s využitím atomových hodin Microsemi 5071A.

Prakticky každý počítačový systém vyžaduje ke svému fungování nějakou formu hodin. Pokud je systém izolován, není třeba klást příliš velký důraz na stabilitu nebo frekvenci daných hodin. Je-li ovšem daný systém připojen do sítě, je přesný čas kritický pro správné a rychlé fungování mnoha síťových protokolů.

Aby bylo možné sjednotit čas na síťových zařízeních, byly vytvořeny tzv. časové servery. Nejprve se stanoví referenční zdroj času, ze kterého je čas načítán časovým serverem a následně šířen pomocí Network Time Protocol (NTP) na jiné časové servery, případně klientská zařízení.

V dnešní době je relativně snadné vytvořit vlastní časový server, který nespolehá na fungování ostatních serverů, ale čas získává sám z nějakého referenčního zdroje. Tento referenční zdroj poskytuje informaci o přesném čase. Tento čas je poté načten časovým serverem, který jej distribuuje dále do sítě.

Způsobů, jakými může časový server načíst čas z referenčního zdroje je mnoho. Základ ale tvoří tři neoddelitelné informace. *Kdy* byl čas načten, *jaký* čas byl načten a nakonec *kolik* času uběhlo mezi jednotlivými načteními.

V laboratořích CESNET je nainstalován Primary Frequency Standard (PFS) Microsemi 5071A. Ten poskytuje frekvenční výstupy v rozsahu od 100kHz do 10MHz a Pulse Per Second (PPS) signál, jak uvádí produktová specifikace [2]. PPS signál je sekvence elektrických pulsů, kdy náběžná hrana pulsu udává začátek sekundy. Není tedy určen k síťové distribuci přesného času, ale pouze ke generování vteřinových pulzů a jiných stabilních frekvencí. K tomu, aby bylo možné využít PFS jako časový server, je obvykle zapotřebí doplňujícího systému Primary Reference Time Clock (PRTC), který za pomoci čítače PPS pulzů může poskytovat přesný čas ostatním zařízením.

Připojení PRTC k PFS a získávání přesného času tímto způsobem se tedy může zdát jako jasná volba. Primární frekvenční standard Microsemi 5071A má ale také schopnost nastavit a uchovávat čas. Tato funkcionality se využívá

především pro automatizaci, může ale být také využita pro potřeby tohoto projektu. Čas nastavený na PFS je možné získat zpět buď přečtením z předního panelu, nebo pomocí sériové linky.

Tato práce využívá schopnosti Microsemi 5071A uchovávat přesný čas a nahrazuje PRTC jednodušším a méně nákladným systémem, který v pravidelných intervalech načítá informace o čase a předává je spolu s PPS signálem NTP serveru.

V neposlední řadě je také součástí této práce měření a vyhodnocení frekvenční stability a časové odchylky z pohledu potenciálního uživatele.

Pokud budou výsledky měření dostatečné, může být výsledný produkt využíván nejen koncovými uživateli, ale může také sloužit jako jeden ze zdrojů pro ostatní časové servery.

Cíl práce

Cílem této práce je vytvořit NTP server s vysokou stabilitou a nízkou odchylkou od referenčního času. Tento NTP sever by měl poskytovat přesnější čas díky načítání dat z atomových hodin Microsemi 5071A řízených césiovým oscilátorem.

1.1 Získávání dat

K dispozici je PFS Microsemi 5071A se schopností poskytovat přesný čas, ovšem pouze přes sériovou linku. Pro dosažení hlavního cíle je nutné zajistit komunikaci mezi vývojovou platformu Raspberry PI (RPi) a PFS. Tato komunikace musí obsáhnout jak správné zpracování PPS signálu, tak načítání referenčního času.

1.2 Administrace PFS

Jedním z požadavků zadavatele bylo neztratit přístup k sériové lince, případně mít možnost PFS ovládat vzdáleně. Toho lze docílit mnoha způsoby, např.:

- přidáním virtuálního sériového portu;
- vytvořením TUI pro model klient–server;
- využitím webového rozhraní.

1.3 Fyzické vlastnosti

PFS je instalován v 19in racku v serverovém areálu sdružení CESNET. Jelikož musí být vývojová platforma RPi zapojena přímo do PFS přes RS-232, je nutné upevnit výsledný produkt adekvátním způsobem. Pouze položit vývojovou desku na atomové hodiny je v serverovém prostředí nepřijatelné.

Dalším cílem je nainstalovat diagnostický display pro možnost rychlého ověření funkčnosti a parametrů systému.

1.4 Simulátor

Vzhledem k umístění PFS a faktu, že se aktivně používá, je fyzický přístup relativně omezen. Využívat PFS při vývoji a testovat na produkčním zařízení proto nepřichází v úvahu. Je tedy třeba vytvořit prostředí, které bude schopné simulovat chování všech kritických částí PFS Microsemi 5071A.

Tyto komponenty tvoří generátor PPS pulzu a simulátor Command Line Interface (CLI) prostředí na sériové lince se sadou základních příkazů pro získávání přesného času.

Jako referenci pro tvorbu simulátoru CLI prostředí lze využít referenční manuál PFS viz. [3]. Generovat PPS pulz je možné libovolným způsobem, jsou však kladeny relativně striktní požadavky na jeho stabilitu.

1.5 Vyhodnocení parametrů

Posledním cílem této práce je vyhodnotit parametry implementovaného NTP serveru. Důležité je zachytit nejen krátkodobé chování v rozmezí několika minut, ale i dlouhodobé trendy. Při vyhodnocování není kladen důraz na parametry v laboratorním prostředí, ale na srovnání kvality z pohledu průměrného uživatele oproti jinému časovému serveru.

Analýza a návrh

Vytvořit vlastní NTP server s referenčním zdrojem času GPS je v dnešní době relativně snadné a levné, jak navrhuje D. Taylor [4]. Člověk by se tedy mohl domnívat, že stejné bude platit i pro ostatní zdroje času.

Atomové hodiny Microsemi 5071A používají pro komunikaci po sériové lince standard Standard Commands for Programmable Instruments (SCPI) viz. referenční manuál [3]. Po prostudování zdrojového kódu referenční implementace NTP, viz. [5] však zjistíme, že žádný z referenčních zdrojů času SCPI standard nepodporuje. Místo podpory komunikačních protokolů na úrovni NTP serveru je obvykle využívána překladová vrstva, např. `gpsd`. Ta podporuje mnoho exotických komunikačních protokolů pro různé zdroje času, ze kterých čas načítá a následně jej předá NTP službě v jednotném formátu.

Překladová vrstva `gpsd` vyžaduje, aby referenční zdroje času splňovaly standard NMEA-0183, případně aby pro ně byla přidána speciální podpora, viz. [6]. Ani jeden z těchto požadavků naše atomové hodiny nespĺňují a využití `gpsd` jako překladové vrstvy tedy není možné.

Vzhledem k rozmanitosti zařízení využívající SCPI pro komunikaci po sériové lince se ukázalo, že neexistuje již hotová překladová vrstva mezi PFS Microsemi 5071A a NTP službou.

2.1 Výběr NTP služby

Vytvářet vlastní NTP server je zbytečné. Existuje řada funkčních ověřených a velice dobře podporovaných NTP serverů. Ve světě Linuxu patří mezi nejpoužívanější následující:

- referenční implementace NTP;
- NTP server `chrony`;
- NTP server `ntpsec`.

Každá z těchto implementací má své výhody i nevýhody.

2.1.1 Referenční implementace NTP

Referenční implementace podporuje prakticky všechny části NTP protokolu, viz. [7]. Jedná se o ukázkou, jak může být NTP server implementován. Vzhledem k tomu, že je tento projekt vyvíjen spolu se samotnou specifikací NTP protokolu, obsahuje také mnoho zastaralých částí, které jsou udržovány pouze kvůli zpětné kompatibilitě.

Důsledkem toho je velice komplikovaný systém, který zcela zbytečně přesahuje požadavky tohoto projektu. Tato komplexita může navíc ztěžovat jakékoliv potenciální rozšiřování stávající implementace, například o další referenční zdroj času.

2.1.2 Server chrony

Jedná se o moderní implementaci NTP serveru, která s referenční implementací sdílí pouze splnění požadavků specifikace protokolu. Tento server nepodporuje celý NTP protokol, jak ale zmiňuje M. Lichvar [7], dokáže synchronizovat hodiny rychleji než referenční implementace NTP, vyžaduje méně paměti a není tolik náročný na procesor. Navíc poskytuje bezpečnostní nadstavbu NTS pro autentizaci.

Jedním z nedostatků je chybějící podpora pro NTP režim 6, který využívá diagnostický program `ntpq`. Vzhledem k tomu, že součástí zadání je „*Vyhodnoťte parametry serveru prezentované programem ntpq. . .*“, je tato implementace pro potřeby tohoto projektu nepoužitelná.

2.1.3 Server ntpsec

Tento server je bezpečnější variantou referenční implementace. Přináší jednodušší konfiguraci referenčních zdrojů času i podporu pro `ntpq`. Tento projekt je navíc rozsahem mnohem menší než referenční implementace, proto jakékoliv nutné úpravy pro podporu nové překladačové vrstvy budou mnohem méně náročné.

Tento server byl tedy zvolen pro následující vývoj tohoto projektu. Nutno podotknout, že tento projekt informuje NTP server o aktuálním čase přes Shared Memory (SHM) driver. Výsledná implementace je tedy kompatibilní se všemi zde uvedenými servery.

2.2 Interpretace PPS signálu

Linux kernel obsahuje podporu pro zpracování PPS signálu viz. [8]. Tento modul kernelu poskytuje základní interpretaci PPS pomocí přerušení. Je obsažen také v upraveném kernelu pro vývojovou desku Raspberry PI.

Jako jedna z možných alternativ je kernel modul `pps-gpio-poll`, který místo přerušení čeká v aktivní smyčce na změnu stavu GPIO. Dle autora by měl poskytnout stabilnější signál a nižší jitter [9].

2.3 Komunikace s `ntpsec`

Přidat nový referenční zdroj času do `ntpsec` je relativně snadné. Program umožňuje zpracování PPS signálu, ovšem samotný PPS signál pro určení přesného času nestačí. Jak již bylo v úvodu zmíněno, pro správné fungování potřebujeme tři údaje: kdy nastal okamžik načtení času, časový údaj, který referenční hodiny prezentují a periodu jednotlivých událostí.

Pro získání periody stačí, aby NTP server zaznamenal více časových údajů. Okamžik načtení času, známe také, jelikož se jedná o moment, kdy NTP server daný pulz zpracoval. Zbývá tedy dodat časový údaj spojený s tímto pulzem.

Časový server `ntpsec` dokáže načítat data o aktuálním čase z externích zdrojů. Využívá k tomu tzv. referenční hodiny, viz. [10]. Jedná se o sadu protokolů, přes které komunikuje s ostatními programy nebo připojenými periferiemi a načítá informace o času.

Pro potřeby tohoto projektu stačí nastavit dvoje referenční hodiny:

- `pps`, pro zpracování PPS pulzu, který bude distribuovat Linux kernel a
- `shm`, který umožňuje vzájemnou komunikaci procesů pomocí SHM.

Posledním krokem tedy zůstává ve správný okamžik načíst čas z PPS, převést jej do srozumitelné formy pro `ntpsec` a poskytnout pomocí SHM.

2.4 Přestupná sekunda

Kvůli drobným odchylkám v pravidelnosti orbitu planety Země kolem Slunce a rychlosti rotace Země kolem vlastní osy může být nutné referenční čas upravit viz. [11]. Toho je docíleno tzv. *přestupnou sekundou*, která může být několikrát za rok přidána nebo odebrána.

Dle D. Gambis [12] byly v praxi přechodné vteřiny přidávány vždy k datu buď 30.6. nebo 31.12. Informace o přidání nebo odebrání přechodné sekundy jsou publikovány organizací IERS. Následná zpracování těchto oznámení jsou provedena buď automaticky, pokud je dané zařízení připojeno k internetu, nebo je musí provést správce systému.

2. ANALÝZA A NÁVRH

Jak je uvedeno v manuálu atomových hodin Symetricom 5071A [13], jsou tyto hodiny schopné zpracovávat informaci o přestupných sekundách. Vzhledem k tomu, že není možné tento PFS připojit k internetu, musí být informace o blížící se přestupné sekundě zadána ručně.

Realizace

3.1 Vývoj simulátoru

Jako platformu pro simulátor byl zvolen starý počítač značky Lenovo, který měl port RS-232 na základové desce. Předpokladem bylo, že tato volba vývoj značně usnadní. Důsledkem tohoto rozhodnutí však vznikly mnohé komplikace, obzvláště se synchronizací simulátoru a generátoru PPS.

3.1.1 Konzultace manuálu

Simulátor byl tvořen *naslepo*. Před započítím tvorby simulátoru nebyla možnost s atomovými hodinami nijak interagovat a jako jediný podklad pro vývoj byly informace uvedené v manuálu [3]. Pozdější testování výsledné aplikace vedlo k objevení několika nesouladů atomových hodin a dodaného referenčního manuálu. Hlavní rozpor byl ve formátu CLI prostředí. Manuál v tomto ohledu nezachází do podrobností, pouze uvádí hrubý formát, kterého se většina výpisů drží jen částečně.

3.1.1.1 Formát CLI

Při připojení k sériové lince se zobrazí prompt. Ten má buď formát `scpi >` nebo `E-xxx >`, kde `xxx` je kódem poslední chyby.

Za tento prompt je možné zadávat příkazy. Ty jsou rozděleny do dvou kategorií: čtecí a ukládací.

Čtecí příkazy jsou zakončeny otazníkem a mají hierarchický formát. Například zobrazení aktuálního času je možné příkazem `:ptime?`. Zatímco hierarchicky vnořený příkaz pro získání aktuálního data je `:ptime:mjdate?`.

Příkazy určené pro ukládání dat nejsou zakončeny otazníkem a přijímají jeden nebo více argumentů. Například příkaz pro zapnutí obrazovky je `display:enable 1`.

Příkazy je také možno zkracovat. Seznam možných zkratek je uveden v manuálu, viz. [3]. Například příkaz `:ptime:mjdate?` je možné zkrátit na `:ptim:mjd?`.

Pro dostatečné otestování překladové vrstvy nemusí simulátor podporovat všechny příkazy. Byla proto vybrána pouze omezená skupina nepostradatelných funkcí:

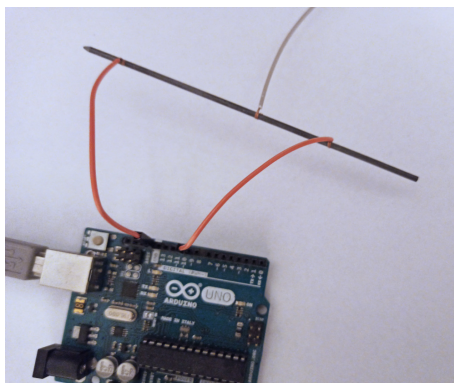
- `:ptim?` – výpis aktuálního času;
- `:ptim:mjd?` – výpis aktuálního data;
- `:ptim:leap?` – stav přechodné vteřiny;
- `:syst:err?` – vypsání chyby;
- `*CLS` – vymazání chyb.

Poslední dva příkazy umožňují vymazat chyby systému; `:syst:err?` vypíše poslední chybu a zahodí ji; `*CLS` pak vymaže všechny chyby.

3.1.2 Proces výroby generátoru PPS

Jako generátor PPS pulzů byla použita vývojová deska Arduino UNO. Stačilo pouze upravit časování v ukázkovém programu pro rozblikání diody a generátor PPS byl hotový. Problém nastal při připojení vývojové desky Arduino UNO k desce RPi, kde je vyvíjen zbytek projektu. Výstupního napětí desky Arduino UNO je 5V a v dokumentaci pro RPi se uvádí, že podporuje maximálně 3.3V, viz. [14]. Bylo tedy nutné sjednotit napětí odporovým děličem.

V době vývoje simulátoru jsem neměl přístup k základním elektronickým součástkám a musel jsem si tedy poradit jinak. K tuze ze staré tužky jsem na jeden konec připojil 5V z Arduína, na druhý konec zem a voltmetrem jsem hledal místo, kde se napětí pohybovalo kolem 3V. (Vlastní prohlášení)



Obrázek 3.1: Grafitový dělič napětí – vlastní tvorba

Tento předpotopní odporový dělič na základní testování stačil, ale postupem času byl nahrazen standardním breadboardem s klasickými součástkami.

3.1.3 Synchronizace

Jedním z problémů oddělení simulátoru CLI a generátoru PPS je synchronizace jejich vzájemných hodin. Pokud simulátor CLI nenačítá data z čítače PPS pulzů, může po chvíli nastat situace, kdy počet PPS pulzů neodpovídá času, který CLI hlásí.

Tento problém byl v první fázi testování ignorován a k řešení došlo až v posledních krocích doladování. Řešením bylo přidat do vývojové desky Arduino UNO čítač PPS pulzů, jehož hodnota se každou vteřinu vypsala na USB sériovou linku. Tento výstup pak zpracovával simulátor CLI, který běžel na stejném počítači, ke kterému bylo Arduino připojeno. K hodnotě čítače přičetl simulátor čas systému v době svého spuštění a tento výsledný čas pak poskytoval přes simulované rozhraní.

3.2 Struktura programu

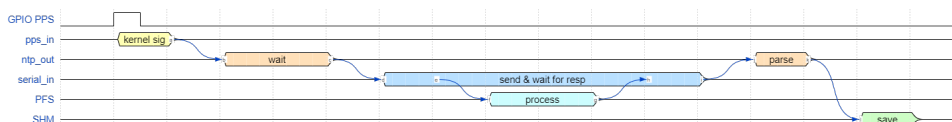
Program se skládá z několika oddělených komponent, které spolu komunikují asynchronně. Toto je seznam jednotlivých komponent s jejich krátkými názvy, které jsou použity i dále v textu:

- komunikátor po sériové lince – `serial_in`;
- publikování dat pro NTP server – `ntp_out`;
- HTTP server – `web_server`;
- ovladač LCD – `lcd_out`;
- monitor PPS signálu – `pps_in`.

3. REALIZACE

Většina těchto komponent má vlastní smyčku, ve které čeká na událost z vnějšího prostředí. Zbylé komponenty čekají na signál od jiných komponent, viz. obr. 3.2. Zde je popis časové posloupnosti běhu programu za normálních podmínek:

1. `pps_in` čeká na signál od kernelu, že přišel nový pulz.
2. V momentě, kdy je pulz zachycen, informuje `ntp_out`.
3. `ntp_out` vyhodnotí, podle délky předchozího dotazu, za jakou dobu se má dotázat `serial_in` na hodnotu času.
4. `ntp_out` pošle dotaz na čas, případně datum a případně stav přechodné vteřiny do `serial_in`.
5. `serial_in` určí jakou prioritu tento dotaz má a zařadí jej do fronty, kterou postupně zpracovává.
6. `serial_in` pošle data z dotazu po sériové lince atomovým hodinám a počká na odpověď, kterou pak pošle zpět.
7. `ntp_out` zpracuje odpověď od atomových hodin, sestaví zprávu pro NTP server a pošle mu data.
8. Cyklus je dokončen a program čeká na další PPS pulz.



Obrázek 3.2: Časový diagram programu – vlastní tvorba

Vedle tohoto hlavního cyklu je spuštěn webový server, který čeká na požadavky od klienta. V momentě, kdy požadavek přijde, pokusí se jej zpracovat. V případě, že si klient vyžádal webovou stránku, server ji načte z disku, sestaví odpověď a pošle ji klientovi zpět. Pokud klient odeslal požadavek s příkazem pro hodiny, server vyhodnotí, zda se jedná o platný příkaz. Data z příkazu pak pošle `serial_in` na zpracování, vyčká na odpověď, kterou pošle klientovi zpět.

Závěrem máme `lcd_out` komponentu, která v pravidelných intervalech sleduje stav systémových hodin a vypisuje je na display. Tato komponenta je také informována těsně před neočekávaným pádem programu a pokusí se vypsát chybovou hlášku na display. Vzhledem k asynchronnímu charakteru programu tato funkcionality nefunguje vždy zcela spolehlivě.

3.2.1 Typy dotazů SCPI

Hlavním dotazem je dotaz na aktuální čas, tedy příkaz `:ptim?`. Tento dotaz probíhá v konfigurovatelných intervalech, vždy však po PPS pulzu.

Další dotaz je pro určení data. Tento dotaz proběhne jednak při spuštění programu (v inicializační části) a také každý den při prvním dotazu po půlnoci. Toho je docíleno tak, že si program uchovává poslední načtený čas. Pokud je nově načtený čas menší než předchozí, chápeme to jako přechod z času menšího nebo rovnému 23:59:59 na čas větší nebo roven 00:00:00. Tedy nový den.

Poslední dotaz v základní smyčce je na přechodnou vteřinu. Jak uvádí D. Gambis v Bulletinu C 36 z IERS EOP PC „*Přechodné vteřiny mohou být oznámeny v UTC na konci měsíců prosince a června*“ [12]. Tento dotaz se provádí pouze spolu s dotazem na datum a pouze v měsíci červnu a prosinci.

3.2.2 Okamžik dotazu

Jednou z klíčových vlastností programu je přiřadit sekundě, která právě proběhla, správné označení. Pokud by toto načítání a označování probíhalo kdykoliv, mohla by nastat situace, kdy program načte předchozí nebo až nadcházející vteřinu. NTP server by pak tato data buď zahodil, čímž by klesla důvěryhodnost našeho zdroje, nebo by je zpracoval a jakákoliv předchozí stabilita hodin by tím byla ztracena.

Důležité tedy je vybrat správný okamžik pro načtení vteřiny. Ideální případ je, aby atomové hodiny dokončily zpracování dotazu přesně o půl vteřiny po PPS pulzu. K tomuto řešení se program snaží alespoň přiblížit.

Všechny dotazy od `ntp_out`, s výjimkou prvního, jsou zpožděny tak, aby půl vteřiny po PPS pulzu atomové hodiny zpracovávaly dotaz. Program nejprve změří round-trip prvního dotazu a délku trvání si uloží. Toto měření započítává pouze dobu trvání dotazu pro čas. Datum i přestupná sekunda jsou časovány samostatně, pouze pro případnou konečnou korekci času. Při příchozím PPS pulzu si program načte hodnotu trvání předchozího dotazu (`round trip`) a uspí vlákno na dostatečnou dobu. Tuto dobu (`wait ms`) můžeme odhadnout podle vzorce 3.1

$$\text{wait [ms]} = 500 - \frac{\text{round trip [ms]}}{2} \quad (3.1)$$

Zbylé dotazy pro datum a přechodnou vteřinu jsou časovány samostatně. Pokud celkový čas dotazů přesáhl jednu vteřinu, je k výslednému času tato vteřina přičtena. Takto se zajistí, že program poskytne správný čas, který nastal ne v okamžiku načtení hodnot, ale až v okamžiku publikování času NTP serveru. Je proto také nutné provést dotaz na čas jako první, aby při něm bylo nejmenší možné zpoždění a abychom mohli zaručit jednoznačnost načtených hodnot.

3.2.3 Inicializační fáze

Prvotní spuštění programu představuje problém, jelikož systémové hodiny nejsou seřizeny podle referenčního zdroje a časová disciplína NTP serveru ignoruje údaje rozdílnější než pět vteřin, viz. zdroj `ntpsec` [15]. Při inicializaci programu je tedy nejprve z atomových hodin načten aktuální čas a datum. Program poté zastaví NTP server, počká několik vteřin, nastaví systémový čas podle hodnot načtených z PFS a po několika dalších vteřinách NTP opět zapne. Po tomto restartu NTP serveru program spustí svou hlavní smyčku a čeká na PPS signál.

3.3 Komponenty

Celý program se skládá z několika komponent. Mimo ty, vyjmenované v sekci Struktura programu 3.2, obsahuje ještě systém načítání a zpracování konfigurace, výpisy ladění a event bus.

3.3.1 Konfigurace

Konfigurace je možná pomocí konfiguračního souboru, jehož cestu program přijímá jako první argument. Pokud konfigurační soubor není předán nebo některé položky chybí, jsou doplněny výchozími hodnotami. Dále následuje stručný popis jednotlivých možností.

Položka	Výchozí hodnota	Vysvětlení
path	/dev/ttyUSB0	Cesta k sériové lince
www_path	/var/www/atomic	Cesta ke kořenu webového serveru
pps_path	/dev/pps0	Cesta ke zdroji PPS
log_file	/var/log/atomic	Cesta k souboru výpisu záznamů
source_pool_rate	5	Perioda načítání času z PFS ve vteřinách
max_local_delta	5	Maximální rozdíl systémových hodin a načteného času
trust_source	true	Indikátor, zda má být systémový čas přepsán
sync_date	10	Perioda, jak často má být načteno datum
web_port	80	Port webového serveru
retries	2	Počet pokusů obnovení při chybě
shm_id	2	Kód SHM pro NTP server

Tabulka 3.1: Seznam konfiguračních možností – vlastní tvorba

Většina položek je jednoznačná, ovšem některé vyžadují podrobnější popis.

3.3.1.1 Položka `trust_source` a `max_local_delta`

Chování nastavení `max_local_date` závisí na položce `trust_source`. Pokud je `trust_source` nastaven na `true` a rozdíl času systému a času načteného z PFS je větší než hodnota `max_local_date`, program provede úpravu systémových hodin stejným postupem jako v inicializačním kroku. Pokud je ovšem nastaven na `false`, pak program zahlásí chybu, která se řídí podle pravidel pro `retires`.

3.3.1.2 Položka `sync_date`

Datum je automaticky synchronizováno každý den. Program ovšem může být instruován, aby datum synchronizoval častěji. Hodnota této položky udává po kolika načtení času z PFS se má načíst také datum.

3.3.1.3 Položka `retires`

Pokud při běhu programu nastane kritická chyba, je tato chyba vyhozena jako výjimka a nahlášena. V klasickém systému by toto způsobilo pád programu. Při zachycení chyby je inkrementován čítač dané chybující komponenty. Dokud čítač nepřesáhne hodnotu uvedenou v této položce, smyčka komponenty je obnovena. Pokud čítač přesáhne hodnotu, je program násilně ukončen. V případě že po restartu smyčky chybující komponenty proběhne další smyčka bez chyb, je čítač chyb nastaven zpět na nulu.

Je-li hodnota konfigurační položky `retires` nastavena na nulu, není program nikdy násilně ukončen a neustále se snaží obnovit všechny chybující komponenty.

3.3.1.4 Položka `shm_id`

Pro komunikaci s NTP serverem pomocí SHM je nutné si nejprve od Linuxového kernelu vyžádat blok ve sdílené paměti. Jednotlivé bloky jsou od sebe odlišeny unikátním klíčem. Dva programy, které spolu chtějí komunikovat pomocí SHM, musí tento klíč znát předem. Ve specifikaci `ntpsec` je hodnota tohoto klíče uvedena jako součet `0x4e545030` hex + indexu datového bloku. Tyto indexy bloků jsou poté rozlišovány při konfiguraci `ntpsec`.

Hodnota položky `shm_id` je index tohoto datového bloku. Pro správné fungování musí být jak v konfiguraci `ntpsec`, tak i v tomto programu nastavena stejně.

3.3.2 Komunikace po sériové lince

Hlavní část programu tvoří komunikační vrstva pro sériovou linku. Systém je rozdělen do dvou částí: *synchronní komunikátor* a *asynchronní rozhraní*, které dohromady tvoří model producer–consumer.

Synchronní vrstva si udržuje odkaz na aktivní spojení sériové linky, odesílá požadavky a zpracovává následné odpovědi. Všechna volání v této vrstvě jsou blokující.

Asynchronní složka přidává vrstvu abstrakce a poskytuje příkazy pro načtení času nebo odeslání surového dotazu. Umožňuje thread-safe přístup k sériové lince tak, že sestaví dotaz pro PFS, přidá jej do fronty příkazů pro zpracování a čeká na příchozí odpověď. Tato vrstva také provádí zpracování zpráv obsahující čas a datum.

3.3.2.1 Formát data

Informace o datu poskytuje PFS ve formátu Modified Julian Date (MJD). Vzhledem k tomu, že NTP server očekává čas i datum jako jedno číslo ve formátu `Unix time`, je nutné toto datum převést. To lze pomocí vzorce 3.2.

$$\text{Unix time} = (\text{MJD} - 40587) * 86400 \quad (3.2)$$

3.3.3 Propagace času

Jak referenční implementace NTP serveru, tak `ntpsec` dovedou zpracovávat čas z referenčních zdrojů pomocí SHM. Sdílená paměť je poskytována Linux kernelem pro Inter-process communication (IPC). Při konfiguraci serveru `ntpsec` je možné nastavit operační režim referenčního zdroje času.

Režim 0 je určen pro privátní přístup. Blok sdílené paměti má nastavená oprávnění 0600 a předpokládá se, že tento blok paměti využívá pouze `ntpsec` a překladová vrstva. Nevýhodou tohoto režimu je, že aby mohla překladová vrstva zapisovat data do tohoto SHM bloku, musí být spuštěna s oprávněním `root`.

Režim 1 umožňuje veřejný přístup. V tomto režimu má SHM blok oprávnění 0666. Nad daty z SHM provádí `ntpsec` kontrolu, že je od posledního zápisu nezměnil jiný program. Nevýhodou tohoto režimu je možnost pro ostatní programy ovlivnit distribuované informace o času. Jasnou výhodou ale je možnost spustit překladovou vrstvu bez oprávnění `super-uživatele`.

Tato překladová vrstva umožňuje komunikaci přes SHM pouze v režimu 1. Je tedy možné ji spustit pouze s oprávněním `standardního uživatele`.

3.3.4 LCD display

Součástí, která není příliš kritická pro samotné fungování produktu, je LCD display. Byl nainstalován hlavně pro usnadnění ladění systému. V tomto ohledu svůj účel zcela splnil.

Display zobrazuje informace o aktuálním stavu hodin, které načítá z Linux kernelu pomocí systémového volání `ntp_adjtime`. Tyto informace obsahují aktuální frekvenci, odchylku, případně informaci, že hodiny kernelu nejsou synchronizovány.

Po deseti vteřinách display přepne na informační stránku s aktuálním časem a datem. Informace pro tento panel jsou načítány ze syscall `localtime`. Po dalších deseti vteřinách se display opět přepne na panel informující o frekvenci.

Panel LCD je standardní 16×2, se kterým program komunikuje ve 4bitovém režimu. Program očekává, že jednotlivé piny z LCD jsou připojeny na následující GPIO piny. Tyto hodnoty nejsou uživatelsky konfigurovatelné. Pro jejich změnu je nutná rekompile programů.

- Register select – pin 25;
- Enable – pin 24;
- Data4 – pin 23;
- Data5 – pin 22;
- Data6 – pin 21;
- Data7 – pin 14.

3.3.5 HTTP server

Tvorba vlastního HTTP serveru může být velice komplikovaná. Pro potřeby tohoto projektu však není nutné implementovat všechny funkce HTTP protokolu. Implementace použitá v tomto projektu poskytuje pouze možnost načíst soubor z disku nebo zaslat jednoduchý příkaz na PFS.

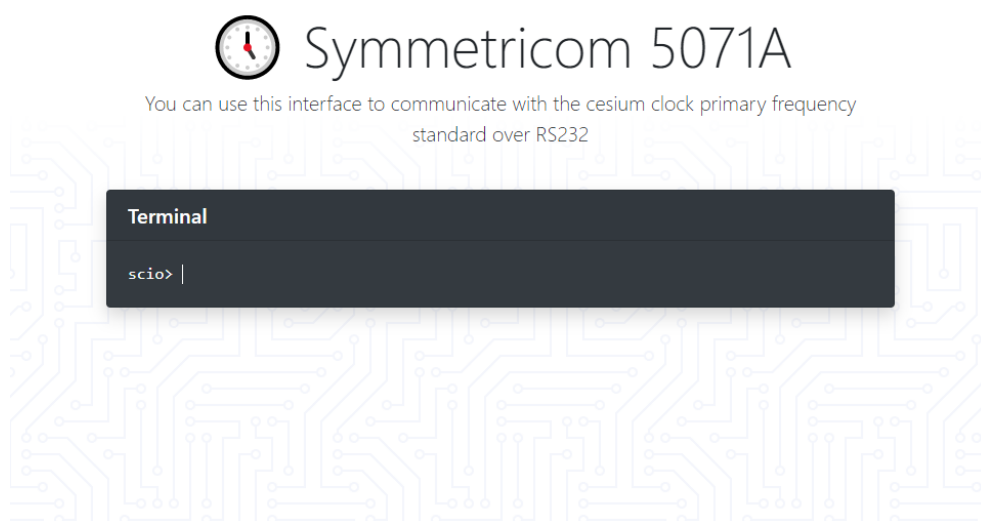
Při spuštění programu začne v základním nastavení HTTP server poslouchat buď na portu 80, nebo na jiném portu dle konfigurace `web_port`. Pokud klient naváže spojení, server zkontroluje, zda se jedná o validní HTTP dotaz.

Pro dotazy typu GET se pokusí poskytnout daný soubor. Při vyhledávání souboru zajišťuje, aby si klient nemohl vyžádat soubor mimo kořen nastavený v konfigurační možnosti `www_path`. Pokud soubor neexistuje, odešle odpověď se stavem 404. Když soubor najde, odešle ho klientovi.

Dotazy typu POST jsou rozděleny na dva koncové body. Zpráva směřovaná na `/cmd` je zpracována jako příkaz pro hodiny. Dotazy pro `/time` vrací aktuální čas. Oba koncové body vrací odpovědi ve formátu XML.

Vzhledem k cílové skupině uživatelů tohoto programu není kladen důraz na podporu celého HTTP protokolu. Tento server podporuje pouze velmi omezenou část protokolu, která je ovšem dostatečná pro zobrazení webové stránky ve většině moderních prohlížečů.

Součástí programu je i webová stránka obsahující základní funkcionalitu pro komunikaci s webovým rozhraním. Na snímku obrazovky obr. 3.3 je možno vidět její vzhled.



Obrázek 3.3: Webové rozhraní – vlastní tvorba

Jedním z cílů bylo minimalizovat množství funkcí, které musí HTTP server podporovat. Hlavní motivací bylo minimalizovat zatížení vývojové desky RPi provozem webového serveru. Dalším účelem bylo minimalizovat množství možných bezpečnostních chyb a tím i potenciálních útoků.

Veškerá funkcionalita webové stránky je spojena do jednoho souboru. Toto vedlo ke snížení množství dotazů od klienta nutných k sestavení výsledné stránky a k odstranění nutnosti odlišovat jednotlivé typy souborů na úrovni serveru.

Pro poskytnutí moderního vzhledu webové stránky a zároveň minimalizování kódu poskytovaného webovým serverem byla použita Cascading Style Sheets (CSS) knihovna Bootstrap verze 4.5 viz. [16]. Tato knihovna je distribuována mimo jiné i pomocí Content Delivery Network (CDN). Není tedy nutné do tohoto projektu přidávat jakékoliv další soubory.

3.3.6 Detekce PPS signálu

Pro určení správného okamžiku pro dotaz, zmíněného v sekci Okamžik dotazu 3.2.2, je nutné detekovat PPS pulz. Linux kernel pro operaci čekání na PPS pulz neposkytuje syscall, je tedy nutné využít surové volání `ioctl` s vlajkou `PPS_FETCH`.

Komponenta pro detekci PPS pulzu opět existuje v odděleném vlákně. Odtud informuje ostatní komponenty pomocí event bus.

3.3.7 Event bus

Páteřní komponentou celého programu je event bus. Přes něj jsou šířeny informace o všech důležitých událostech programu. Je tak zajištěna vzájemná komunikace jednotlivých vláken.

Implementující třída poskytuje možnost zaregistrovat příjem událostí (event listener) a informovat o nastalé události. Při vzniku události event bus nejprve vyhodnotí o jakou událost se jedná a poté spustí všechny připojené event listenery a předá jim argumenty z události.

Přes event bus jsou šířeny informace o PPS pulzech a o kritických událostech, které je nutné zobrazit na LCD display.

3.4 Konfigurace serveru `ntpsec`

Server přijímá data ze dvou referenčních zdrojů. Zpracovává PPS pulzy, které poskytuje Linux kernel, a pro označení jednotlivých vteřin využívá překladovou vrstvu poskytnutou tímto projektem.

```
refclock shm unit 2 refid ATOM flag4 1
refclock pps unit 0 prefer flag2 0 flag3 1 ppspath /dev/pps0
```

Konfigurace referenčních zdrojů času v `ntpsec` – vlastní tvorba

3.4.1 Dotazy `ntpq`

Program `ntpq` umožňuje odesílat dotazy na NTP server. Komunikace probíhá pomocí protokolu NTP v režimu 6 přes UDP. Obvykle je používán pro kontrolu aktuálního stavu NTP serveru a umožňuje také hlubší diagnostiku, např. referenčních zdrojů času. Správná komunikace serveru `ntpsec` s `ntpq` je tedy klíčová.

Vzhledem k tomu, že komunikace probíhá pomocí UDP, je možné NTP server zneužít jako část bot-netu pro amplification attack. Aby se tomu zabránilo, je server `ntpsec` nastaven tak, aby vyhověl pouze dotazům klientů `ntpq` spuštěných na stejném počítači. Dotazy od jiných klientů ignoruje.

Na zbylé dotazy od klientů aktualizujících si vlastní čas server reaguje `ntpsec` dle očekávání.

3.4.2 Zabezpečení a firewall

Na vývojové platformě RPi je použit firewall `nftables`. Je nastaven velmi restriktivně, jelikož server má přiřazenou veřejnou IP adresu a je tedy dostupný odkudkoliv. Umožňuje přístup pouze na port 22, kde je spuštěn SSH server pro vzdálenou správu, na port 123, který využívá NTP protokol a na port 80, na kterém je dostupný webový server. Tento přístup je dále omezen seznamem vybraných IP adres, které mají povolený přístup. Na žádné další dotazy server nereaguje.

Při první instalaci nebyl z důvodu testování firewall nastaven. Během první hodiny provozu HTTP server zaznamenal více než 100 různých pokusů o připojení. Po pozdější analýze těchto dotazů bylo zjištěno, že šlo o programy, které se pokoušely zneužít bezpečnostní chyby v různých často používaných webových serverech a CMS. Jeden z nejčastějších dotazů byl pokus o Remote Code Execution (RCE) na systémech využívající CMS WordPress. Po úspěšném nasazení firewall tyto dotazy zmizely.

3.5 Fyzická část

K tomu, aby program zajišťující komunikaci mezi PFS a NTP serverem mohl přijímat data, musí být vývojová deska Raspberry PI připojena přímo do atomových hodin. Jelikož je ale PFS instalován v 19in racku, viz. obr. 3.4, je nutné i tento projekt podobným způsobem adekvátně upevnit.



Obrázek 3.4: PFS Symmetricom 5071A – vlastní tvorba

3. REALIZACE

Základem konstrukce byla zvolena 1U záslepka do 19in racku. Jedná se o velice levné řešení, které poskytuje dostatečnou volnost při návrhu externího rozhraní. Jedním z nedostatků použití ocelové záslepky byl relativně náročný proces obrábění, který trval řádově několik dní.

Pro zajištění adekvátní komunikace a uživatelské přívětivosti musí být možno k tomuto projektu připojit následující prvky:

- napájení;
- signál PPS;
- sériovou linku;
- ethernet.

Napájení je možno buď pomocí USB portu přímo na vývojové desce, nebo externím konektorem zapojeným do GPIO. Externí konektor ovšem neposkytuje přepětovou ochranu a je tedy nutné použít dostatečně kvalitní zdroj, který vývojovou desku nepoškodí.

Signál PPS je obvykle šířen koaxiálním kabelem pro snížení možného rušení. Na PFS Symetricom 5071A jsou vývody frekvenčních generátorů termínovány konektory BNC. Pro tento projekt byl zvolen konektor BNC, který je pomocí stíněného audio-kabelu připojen přímo do GPIO rozhraní vývojové desky RPi.

Instalace *sériové linky* je možná buď pomocí GPIO, nebo USB adaptéru. Pro zapojení přes GPIO je nutné data nejprve převést z formátu RS-232 na Transistor–Transistor logic (TTL). Toho je možné docílit konvertorem, např. čipem MAX3232. Při použití USB adaptéru je realizace mnohem snazší, jelikož prakticky všechny převodníky splňují standard plug-and-play. Vzhledem k tomu, že převodník USB umožňuje velice snadnou instalaci a je zcela podporován v Linux kernelu, byl v tomto projektu využit.

Sériová linka je na PFS vyvedena do konektoru DB9. Aby byl projekt uživatelsky přívětivý, byl na záslepku také instalován konektor DB9. Jejich vzájemné propojení bylo poté provedeno standardním kříženým kabelem.

Návrh předního panelu byl proveden nejprve v Computer-aided design (CAD) software, viz. obr. 3.5 a poté přenesen do reality, viz. obr. 3.6.



Obrázek 3.5: Návrh předního panelu – vlastní tvorba

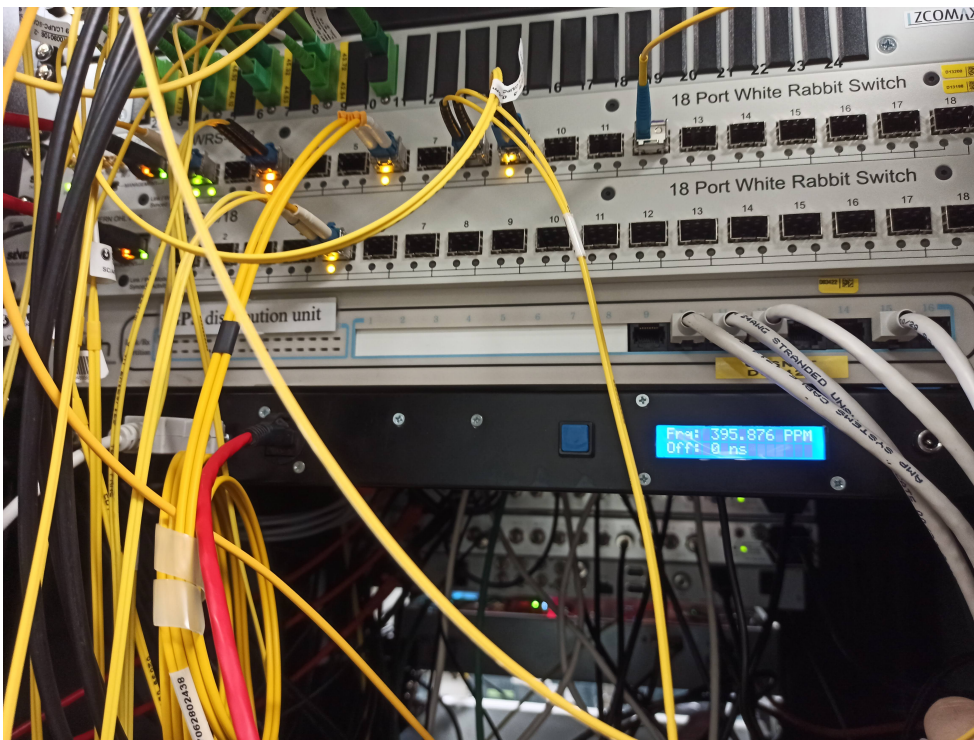


Obrázek 3.6: Provedení předního panelu – vlastní tvorba

3.5.1 Instalace

Zařízení bylo nainstalováno do uzlu sítě CESNET, nad PFS Symetricom 5071A, jak je vidět na obr. 3.7. Bylo připojeno k ethernetu a byla mu přiřazena veřejná IP adresa a záznam v DNS. Nyní je dostupné na adrese `ntp-cs.cesnet.cz`.

První spuštění odhalilo jednu chybu ve zpracování odpovědi z PFS, která pak byla následně opravena. Další chyby se zatím nevyskytly.



Obrázek 3.7: Instalace v uzlu sítě CESNET – vlastní tvorba

Analýza výsledků

4.1 Vyhodnocení parametrů serveru

Bylo provedeno dvoje měření parametrů serveru. První měření bylo provedeno v prostředí běžného uživatele. Cílem experimentu bylo ověřit, jaký vliv má použití PFS jako referenčního zdroje času.

Druhé měření bylo provedeno v prostředí laboratoří CESNET. Cílem tohoto experimentu bylo porovnat kvalitu časového serveru oproti existujícím časovým serverům.

4.2 Měření v uživatelském prostředí

Aby bylo možné vyhodnotit parametry serveru, musíme nejprve stanovit jiný referenční zdroj času, proti kterému budou hodiny ověřovány. K tomu byl využit časový server `tik.cesnet.cz`, který využívá jak vlastního cesiového oscilátoru, tak i jiných externích zdrojů času.

4.2.1 Metodika experimentu

Pro sběr dat byl spuštěn izolovaný NTP server, který své hodiny synchronizuje podle referenčního času načítaného ze serveru `tik.cesnet.cz`. S tímto časem jsou pak porovnány údaje z ostatních NTP serverů, přesněji čas z vývojové platformy RPi. Parametry z tohoto monitorovacího NTP serveru jsou zaznamenávány programem `ntpq` a ukládány pro pozdější analýzu.

Tento experiment byl spuštěn dvakrát. Nejprve při vývoji překladové vrstvy, před nainstalováním projektu do síťového uzlu CESNET, kdy hodiny překladové vrstvy nebyly nijak synchronizovány. A později po nainstalování překladové vrstvy a připojení k PFS.

Monitorující NTP server byl nainstalován v domácím prostředí a připojen k internetu přes asymetrické spojení. Poskytovatel internetového připojení nedává žádné garance na stabilitu spojení nebo minimální odezvu. Garantuje

pouze absenci Quality of service (QoS), tedy rovnoprávnost veškerého síťového provozu. Tento způsob testování byl zvolen proto, aby neověřoval časový server v laboratorním prostředí, ale aby ověřil jeho použitelnost v reálném provozu.

Program `ntpq` byl spuštěn v obou případech v pravidelných desetiminutových intervalech po dobu jednoho týdne. Celkem bylo pro každý server nashromážděno více než 1500 měření. Uložená data obsahují následující informace:

- identifikace zdroje času;
- čas načtení údajů;
- odchylku časového údaje;
- frekvenční stabilitu.

Identifikace zdroje času umožňuje rozlišit jednotlivé zdroje, tedy přesněji referenční zdroj od testovaného zdroje.

Čas načtení údajů udává čas načtení informací z monitorujícího NTP serveru programem `ntpq`.

Odchylka časového údaje určuje rozdíl času, který načetl monitorující NTP server od času, který očekával. Vzhledem k tomu, že čas monitorujícího serveru se neustále synchronizuje s referenčním časem, poskytuje tento údaj několik informací. Předně rozdíl mezi referenčním časem a časem testovaného serveru. Dále obsahuje informace o odchylce času způsobené nestabilitou síťového provozu.

Frekvenční stabilita je vypočtena jako klouzavý kvadratický průměr. Tento údaj vyhodnocuje NTP server sám. Jedná se o odchylku zbavenou trendu a drobného šumu, způsobeného nestabilitou síťového provozu. Může nám poskytnout informaci o krátkodobých i dlouhodobých změnách frekvence oscilátoru testovaného serveru.

4.2.2 Parametry testovaných serverů

V následující tabulce 4.1 jsou uvedeny základní parametry jednotlivých serverů.

Server	Operační systém	Verze NTP serveru
Vývojová platforma RPi	Linux/5.10.17-v7+	ntpd_ntpsec-1.1.3_2019-11-18T06:04:00Z
tik.cesnet.cz	Linux/4.19.0-6-amd64	ntpd_4.2.8p13@1.3847_Thu_Nov__7_18:05:13_UTC_2019_(1)
Monitorující server	Linux/5.10.88-v2	ntpd_4.2.8p13@1.3847_Thu_Nov__7_18:05:13_UTC_2019_(1)

Tabulka 4.1: Parametry NTP serverů – vlastní tvorba

Časový server `tik.cesnet.cz` obsahuje Oven-controlled Crystal Oscillator (OCXO), tedy termostatický oscilátor. Vývojová platforma RPi i monitorující server jsou pak osazeny původními, výrobcem blíže nespécifikovanými křemíkovými krystaly.

4.2.3 Označení serverů

Pro usnadnění popisů jednotlivých serverů byly zavedeny zkratky, viz. tab. 4.2, na které se bude zbytek textu odkazovat.

Zkratka	Popis serveru
<code>undis</code>	Testovaný server bez připojení k PFS
<code>ntp-cs</code>	Testovaný server s připojením k PFS
<code>tik-undis</code>	Referenční NTP server v době měření <code>undis</code>
<code>tik-cs</code>	Referenční NTP server v době měření <code>ntp-cs</code>

Tabulka 4.2: Seznam zkratk NTP serverů – vlastní tvorba

Nutno poznamenat, že mezi servery `tik-undis` a `tik-cs` je rozdíl pouze v době měření. V obou případech se jedná o stejný server `tik.cesnet.cz`.

Stejně tak tomu je i u serverů `undis` a `ntp-cs`. Opět se jedná o stejnou vývojovou desku RPi. V tomto případě je ovšem rozdíl ve zdroji přesného času. Server `undis` spoléhá na vlastní oscilátor, zatímco `ntp-cs` využívá PFS.

4.2.4 Očekávané výsledky

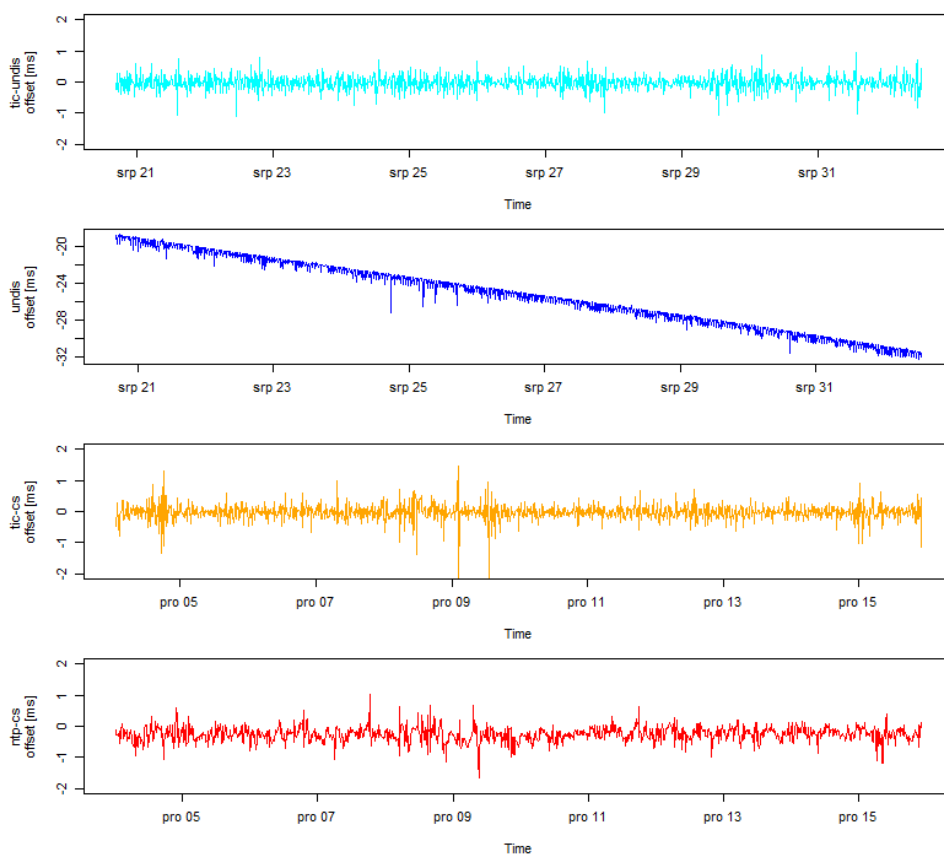
V naměřených datech je očekáván šum způsobený nestabilním oscilátorem v počítači určeném k měření. Tento typ oscilátoru však reprezentuje prostředí běžného uživatele, neboť kvalitní oscilátory se ve spotřebitelských zařízeních obvykle nevyskytují.

Další očekávaná vlastnost je rozdíl trendu časových odchylek u serveru `undis` oproti `tik-undis`. Předpokladem je konstantní rozdíl frekvence oscilátoru instalovaného na desce RPi, oproti oscilátoru v serveru `tik.cesnet.cz`. Jitter, vyskytující se u časových odchylek `undis` a `ntp-cs`, je pak očekáván řádově stejný, neboť je u vestavěného oscilátoru očekávána dostatečná krátkodobá frekvenční stabilita.

4. ANALÝZA VÝSLEDKŮ

4.2.5 Prezentace dat

Následující grafy obsahují surová data odchylek (offset) časových serverů.



Obrázek 4.1: Naměřené odchylky času – vlastní tvorba

Zde je nutno podotknout rozdíl měřítka pro server `undis` vůči ostatním měřením. Vzhledem k značnému rozdílu rozsahu hodnot mezi `undis` a `tik-undis` by se při použití stejných měřítek u serveru `tik-undis` ztratil jakýkoliv detail.

Tyto hodnoty ukazují, že mezi servery `tik-cs` a `ntp-cs` není významný rozdíl. U serveru `undis` je pak jasně vidět rozdíl frekvence oproti `tik-undis`. Tento trend se zdá být lineární, avšak rozptyl šumu se zdá být vyšší.

Vzhledem k tomu, že měření serverů `ntp-cs` a `undis` neprobíhala současně, je nutné prokázat, že hodnoty referenčního zdroje času se v obou případech statisticky významně neliší.

Byl proveden dvouvýběrový t-test s nulovou hypotézou, že střední hodnoty časových odchylek `tik-cs` a `tik-undis` jsou shodné. Na naměřených datech je p-hodnota rovna 0,48. Nulovou hypotézu se nepodařilo vyvrátit a je tedy

možno s určitou mírou rizika předpokládat, že během časového posunu měření nedošlo k výrazné změně hodnot časových odchylek. Nutno podotknout, že vzhledem k množství naměřených dat je možno na základě teorie velkých čísel prohlásit, že data se blíží normálnímu rozdělení.

4.2.6 Odchylka času

Jak je možno na obr. 4.1 vidět, všechna měření mají jistý nenulový šum. Načtená data obsahují také několik extrémních pozorování. Jasně je ale vidět vliv použití PFS. Server `undis` má šum nejen vyšší, ale vyskytuje se i rozdíl v lineárním trendu. Z toho je možno vyvodit, že frekvence hodin bez použití PFS neodpovídá frekvenci referenčního zdroje.

Pokud pro naměřená data určíme úhel, který svírá lineární trend s osou x , můžeme určit rozdíl frekvencí jednotlivých zdrojů času oproti referenčnímu serveru. Pro server `undis` byla tato frekvenční stabilita vypočtena na $-12,8$ ppm. Tato hodnota odpovídá naměřené frekvenční stabilitě v systému `ntpmon`, které je popsáno v sekci Měření v laboratorním prostředí 4.3.

Pro server `ntp-cs` byla frekvenční stabilita lepší než 10^{-11} . Takto nízká hodnota je umožněna kvalitou cesiových oscilátorů, jejichž vlastní stabilita dosahuje hodnoty 10^{-14} .

V tento moment je důležité podotknout rozdíly v použitých oscilátorech jednotlivých systémů. Vývojová platforma RPi je osazena blíže nespecifikovaným křemíkovým krystalem AEL19.2 1516, který má frekvenci 19,2 MHz. U levných oscilátorů můžeme očekávat výrobcem garantovanou přesnost pohybující se řádově v desítkách až stovkách ppm. Oproti tomu výrobce frekvenčního zdroje Microsemi 5071A garantuje frekvenční toleranci ± 5 ppt.

Jak navrhuje T. Mädel na svém blogu [17], jedním z řešení jak vyřešit problém těchto nestabilních hodin je nahradit vestavěný oscilátor. Pro publikování přesného času je samozřejmě třeba stabilní oscilátor, ovšem stejně významnou roli hraje i nastavení NTP serveru a jeho referenčních zdrojů.

4.2.7 Absolutní odchylka

Řešit absolutní odchylku mezi `undis` a `tik-undis` nedává smysl. Jejich vzájemné frekvence se liší natolik, že hodnota odchylky se výrazně změní už v průběhu jednoho dne.

Oproti tomu srovnání `ntp-cs` a `tik-cs` dává dobrý smysl. Rozdíl jejich frekvencí se zdá být statisticky nevýznamný, můžeme tedy porovnat jejich absolutní rozdíly. Tento rozdíl neudává stabilitu nebo dlouhodobou přesnost času, ale rozdíl od skutečného času, který lze posléze snadno kompenzovat.

4. ANALÝZA VÝSLEDKŮ

Porovnány byly odchylky pro každý den měření viz. tab. 4.3, celkem tedy 12 údajů.

Datum	Průměrná časová odchylka
4.12	$-26,198 \mu\text{s}$
5.12	$-25,561 \mu\text{s}$
6.12	$-24,775 \mu\text{s}$
7.12	$-25,670 \mu\text{s}$
8.12	$-23,832 \mu\text{s}$
9.12	$-25,729 \mu\text{s}$
10.12	$-20,250 \mu\text{s}$
11.12	$-19,909 \mu\text{s}$
12.12	$-26,452 \mu\text{s}$
13.12	$-25,709 \mu\text{s}$
14.12	$-26,774 \mu\text{s}$
15.12	$-25,705 \mu\text{s}$
Celé období	$-24,826 \mu\text{s}$

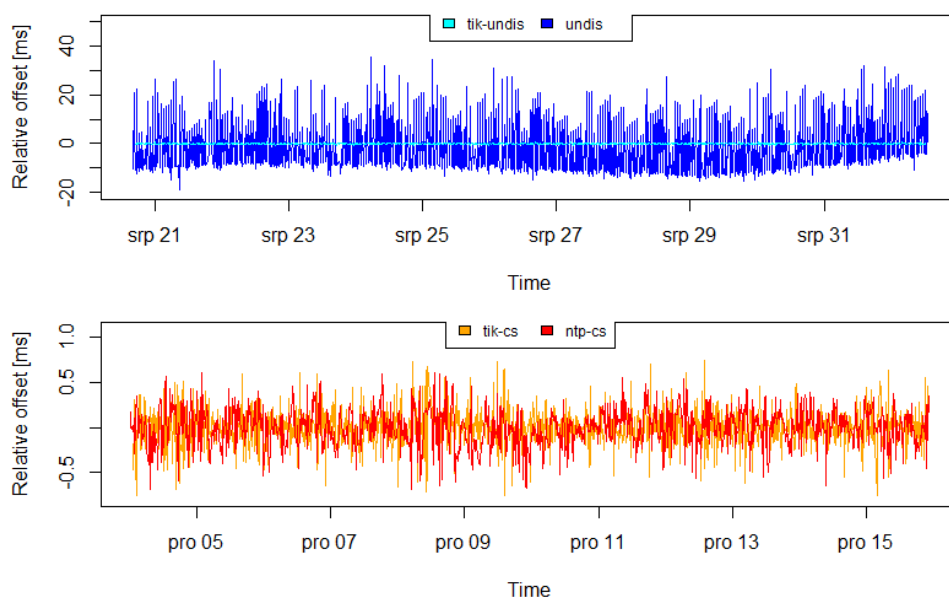
Tabulka 4.3: Rozdíl časových odchylek – vlastní tvorba

Naměřené hodnoty se pohybují kolem $-25 \mu\text{s}$. Je vidět jistý výkyv ve dnech 10.12 a 11.12. Vizualní náhled na stejný výkyv je pak v sekci Dlouhodobá frekvenční stabilita 4.2.9.

Tato absolutní odchylka časového údaje se za dobu 12 dní téměř nezměnila. Odchylka průměrů je $2,294 \mu\text{s}$.

4.2.8 Krátkodobá frekvenční stabilita

Naměřená data obsahují také několik extrémních pozorování. Jedná se o atypické hodnoty, které vznikly nespécifikovanými událostmi (např. neočekávané vytížení sítě, ...), které nejsou pro daný experiment podstatné. Naměřené odchylky byly tedy zbaveny 5 % minimálních, 5 % maximálních hodnot a lineárního trendu. Tato očištěná data jsou už vzájemně srovnatelná pro porovnání krátkodobé i dlouhodobé nestability, viz. obr. 4.2.



Obrázek 4.2: Očištěné odchylky času – vlastní tvorba

Nutno opět podotknout rozdíl měřítek pro osu y.

V očištěných datech je možno vidět jasný rozdíl mezi synchronizovaným a nesynchronizovaným zdrojem času. Z důvodu rozdílného měřítka se šum měřený na `tik-undis` prakticky ztrácí.

Naopak rozdíl mezi `ntp-cs` a `tik-cs` je téměř nerozpoznatelný.

Další studovanou metrikou jsou rozptyly jednotlivých zdrojů času. Je celkem jasné, že menší rozptyl znamená stabilnější zdroj času.

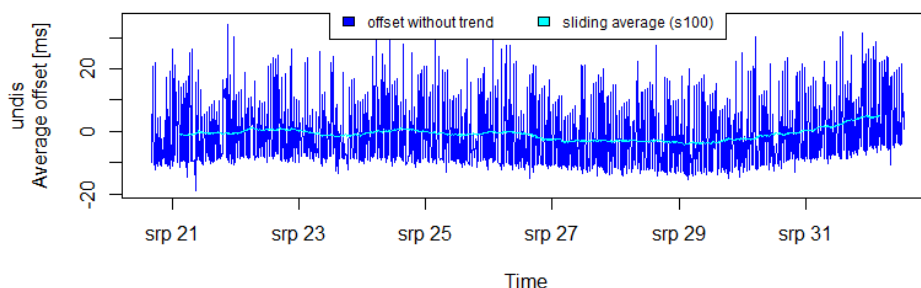
NTP server	Rozptyl časové odchylky
undis	11880 μ s
ntp-cs	43,2 μ s
tik-undis	41,1 μ s
tik-cs	44,0 μ s

Tabulka 4.4: Rozptyl časové odchylky – vlastní tvorba

Jak je možno vidět, synchronizované hodiny jsou takřka neodlišitelné od reference, zatímco hodiny bez PFS mají o 3 řády vyšší rozptyl. Tuto krátkodobou nestabilitu nemůžeme připsat za vinu síťovému provozu. Možným vysvětlením tedy zůstává nekvalitní oscilátor použitý na vývojové platformě RPi.

4.2.9 Dlouhodobá frekvenční stabilita

Jiným pohledem na stejná data je možno odhalit i dlouhodobou nestabilitu serveru `undis`. Následující graf 4.3 ukazuje naměřená data a jejich klouzavý průměr s velikostí okénka 100.



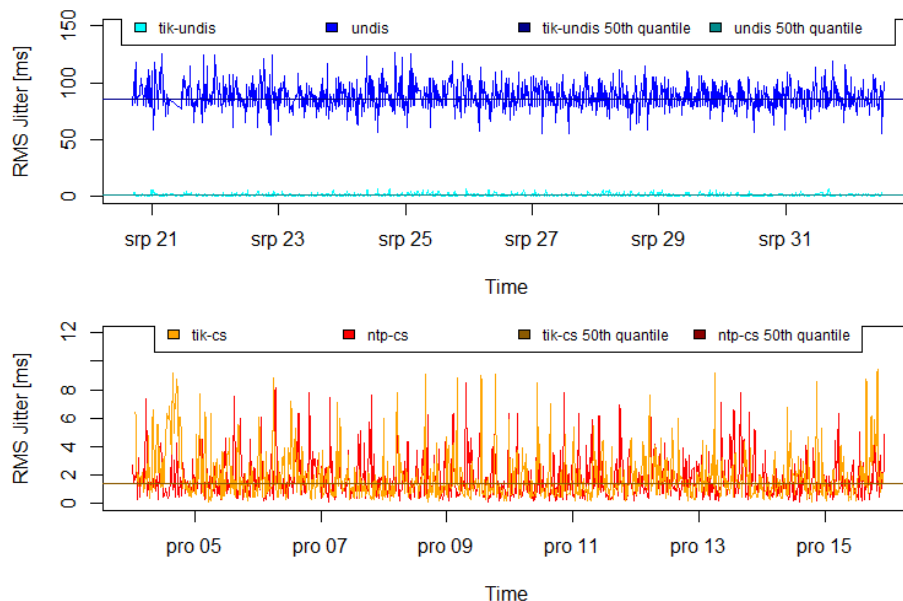
Obrázek 4.3: Průměry odchylek – vlastní tvorba

U nesynchronizovaného zdroje je vidět jasná změna trendu, který začne klesat kolem data 27. srpna a od 29. srpna zase výrazně růst. Přesněji v rozsahu 27. srpna až 29. srpna byla frekvence hodin $-7,2$ ppm a v období po 29. srpnu vystoupala na $-14,3$ ppm. Tato dlouhodobá změna frekvence může být připsána vlivům vnějšího prostředí, například teplotě systému, která v těchto experimentech nebyla zaznamenávána.

Tuto metriku nemá smysl vyhodnocovat pro `ntp-cs`, jelikož je prezentovaný čas neustále upravován podle hodnot z PFS.

4.2.10 RMS jitter

Jako poslední jsou zpracována data o Root mean square (RMS) jitter, přímo poskytovaná monitorujícím NTP serverem. Jedná se o klouzavý kvadratický průměr časových odchylek. Opět byla odebrána extrémní pozorování a to 5% maximálních a 5% minimálních hodnot, viz. 4.4.



Obrázek 4.4: Očištěný RMS jitter – vlastní tvorba

Opět nutno upozornit na rozdíl v měřítkách jednotlivých grafů. Zde je opět jasně vidět rozdíl krátkodobé nestability nesynchronizovaného zdroje. Na grafech je také zanesen 50% kvantil, který má pro server `undis` hodnotu 86,3ms a pro `ntp-cs` je roven 1,38ms. U serverů `tik-cs` a `tik-undis` jsou rovny 1,23ms a 1,31ms.

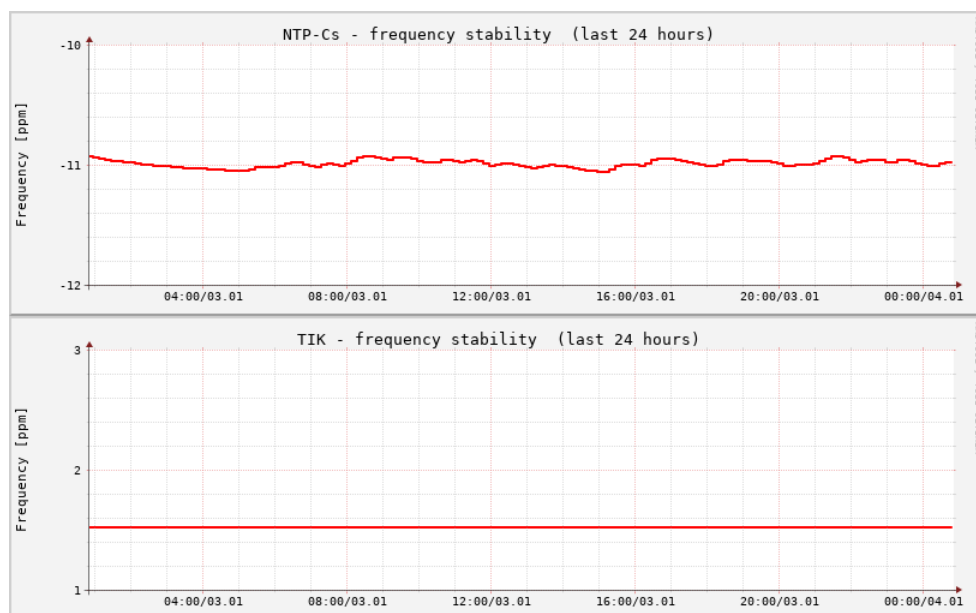
Hodnoty `undis` jsou opět o jeden až dva řády vyšší. Naproti tomu rozdíl mezi `ntp-cs` a `tik-cs` se zdá být téměř nerozlišitelný. Byl tedy proveden t-test, zda je v naměřených datech statisticky významný rozdíl nebo zda jsou `ntp-cs` a `tik-cs` z prostředí normálního uživatele neodlišitelné. Pro t-test s nulovou hypotézou, že testovaný server má stejný RMS jitter jako referenční zdroj vyšla p-hodnota 0,2225. Nulová hypotéza tedy nebyla zamítnuta a s určitým rizikem je možno prohlásit, že rozdíl naměřených dat je zanedbatelný.

4.3 Měření v laboratorním prostředí

V druhém experimentu byly měřeny parametry serveru v laboratorním prostředí. Tato část práce staví na již hotovém systému pro sběr dat z časových serverů, který vyvinul V. Smotlacha [1]. Tento systém monitoruje NTP servery provozované sdružením CESNET a je veřejně dostupný na adrese <http://ntpmon.cesnet.cz/>.

Hodnoty v tomto experimentu nejsou poměřovány proti časovému serveru `tik.cesnet.cz`, ale proti referenčnímu serveru NTP `ref`. Měření byla provedena po dobu 24 hodin.

Následující grafy 4.5 ukazují frekvenční stabilitu (frequency stability) vnitřních hodin serverů `ntp-cs.cesnet.cz` a `tik.cesnet.cz`.



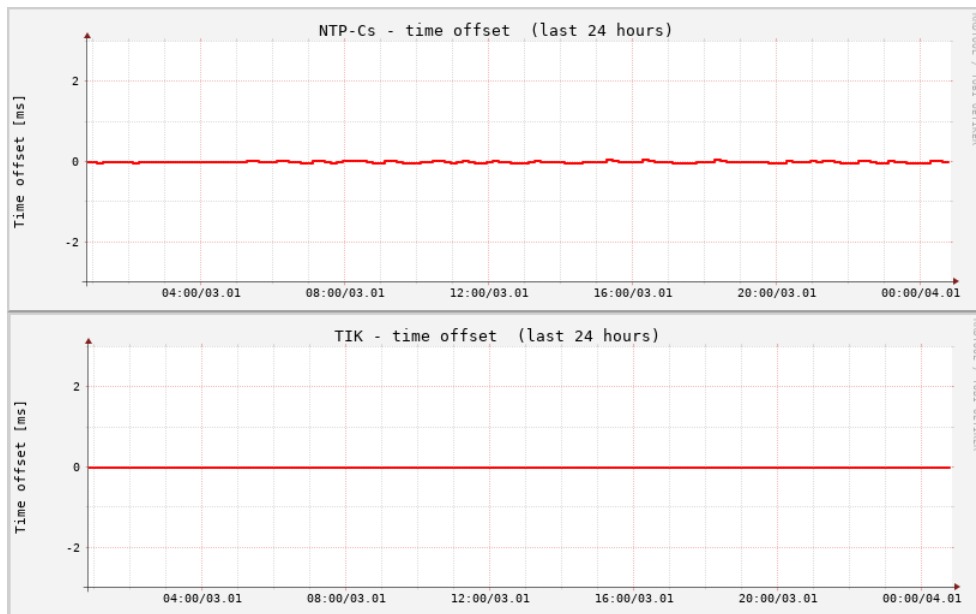
Obrázek 4.5: Frekvenční stabilita – CESNET [1]

Frekvenční stabilita udává stabilitu vnitřního oscilátoru časového serveru. Je určena jako rozdíl systémového času a času, který prezentuje NTP server. Ideální je konstantní hodnota, ke kterému se všechny servery snaží přiblížit.

Server `tik.cesnet.cz` dosahuje téměř konstantní hodnoty 1,534 ppm díky svému vysoce kvalitnímu oscilátoru OCXO. Časový server `ntp-cs.cesnet.cz` kolísá kolem -11 ppm.

Nakonec byly měřeny časové odchylky (time offset) serverů `ntp-cs.cesnet.cz` a `tik.cesnet.cz` oproti monitorujícím serveru NTP `ref`, viz. 4.6.

4.3. Měření v laboratorním prostředí



Obrázek 4.6: Časové odchylky – CESNET [1]

Časové odchylky ukazují rozdíl mezi časem načteným z NTP serveru a časem monitorujícího stroje. U serveru `ntp-cs.cesnet.cz` se vyskytuje drobný třes, který v absolutní hodnotě nepřesahuje $20 \mu\text{s}$. Ten je způsoben krátkodobou nestabilitou vestavěného oscilátoru, na který NTP server při zpracování dat spoléhá.

Pro server `tik.cesnet.cz` jsou naměřené hodnoty rovny $0 \mu\text{s}$. Jsou tedy menší, než je rozlišení škály poskytované monitorujícím serverem NTP `ref`.

Závěr

Byl implementován primární NTP server na Raspberry PI, který využíval PFS Symetricom (Microsemi) 5071A jako svůj primární zdroj času. Před implementací byl vytvořen simulátor tohoto frekvenčního generátoru, který poskytoval základní funkcionalitu, nutnou k otestování funkčnosti NTP serveru.

Tento primární NTP server využíval autorem vytvořenou překladovou vrstvu pro komunikaci s PFS. Ta poskytovala komunikaci mezi NTP serverem a PFS. Dále také umožňovala vzdálený přístup k sériové lince, přes kterou daná komunikace probíhala.

Poté byla spuštěna první fáze měření pro vyhodnocení vlivu PFS jako primárního zdroje času. Výsledek tohoto měření ukázal frekvenční stabilitu vestavěného oscilátoru pohybující se kolem -12 ppm.

Tento projekt, realizovaný na Raspberry PI byl následně nainstalován do produkčního prostředí v serverovém areálu sdružení CESNET. Tam byl připojen k frekvenčnímu generátoru Symetricom (Microsemi) 5071A a k síťové infrastruktuře.

Následně byla spuštěna druhá fáze měření, ve které NTP server využíval PFS pro synchronizaci.

Výsledky těchto měření ukázaly, že použití PFS má významný vliv na krátkodobou přesnost časových odchylek, jejichž rozptyl klesl z naměřených $11880 \mu\text{s}$ na $43,2 \mu\text{s}$. Použití PFS tedy znamená řádově významné zlepšení kvality poskytovaného času NTP serverem.

Závěrem byla vývojová platforma zapojena do monitorujícího systému pro časové servery poskytované sdružením CESNET. Tato měření ukázala negativní vliv vestavěného oscilátoru na stabilitu NTP serveru.

Následujícím možným postupem je na vývojové platformě Raspberry PI nahradit existující křemíkový krystal lepším termostatickým OCXO oscilátorem a provést měření znovu.

Systém je nyní v produkčním prostředí provozován 4 měsíce, je aktivně využíván a funguje spolehlivě. Kvalita času distribuovaného tímto systémem se blíží parametrům profesionálního časového serveru `tik.cesnet.cz`.

Literatura

- [1] Vladimír Smotlacha: *CESNET - time servers*, [online]. 2022, [cit. 2022-01-04]. Dostupné z: <http://ntpmon.cesnet.cz/index.php>
- [2] Microsemi, a wholly owned subsidiary of Microchip Technology Inc.: *5071A Primary Frequency Standard Datasheet*, [online]. 2019, [cit. 2021-11-29]. Dostupné z: https://www.microsemi.com/document-portal/doc_download/133269-5071a-datasheet
- [3] Microsemi, a wholly owned subsidiary of Microchip Technology Inc.: *5071A Primary Frequency Standard Operating and Programming Manual*. Rev. c vydání, s. 91–118, October 2006.
- [4] Taylor, D.: *The Raspberry Pi as a stratum-1 NTP server*, [online]. 2021, [cit. 2021-10-11]. Dostupné z: <https://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html>
- [5] Network Time Foundation: *The Reference Implementation of NTP*, [online]. 2016, [cit. 2021-11-08]. Dostupné z: <https://github.com/ntp-project/ntp>
- [6] Raymond, E. S.: *GPSD: Compatible Hardware*, [online]. 2021, [cit. 2021-10-25]. Dostupné z: <https://gpsd.gitlab.io/gpsd/hardware.html>
- [7] Lichvar, M.: *Comparison of NTP implementations*, [online]. 2020, [cit. 2021-09-09]. Dostupné z: <https://chrony.tuxfamily.org/comparison.html>
- [8] Giometti, R.: *The Linux driver implementer's API guide: PPS – Pulse Per Second*, [online]. 2007, [cit. 2021-11-29]. Dostupné z: <https://www.kernel.org/doc/html/v5.10/driver-api/pps.html>
- [9] Lichvar, M.: *GPIO polling PPS client for Linux*, [online]. 2018, [cit. 2021-09-09]. Dostupné z: <https://github.com/mlchvar/pps-gpio-poll>

- [10] NTP Security Project: *Reference Clock Support*, [online]. 2021, [cit. 2021-12-8]. Dostupné z: <https://docs.ntpsec.org/latest/refclock.html>
- [11] Paris: IERS EOP PC, Observatoire de Paris: *Measuring the irregularities of the Earth's rotation*, [online]. 2013, [cit. 2021-12-08]. Dostupné z: <https://www.iers.org/IERS/EN/Science/EarthRotation/EarthRotation.html>
- [12] Gambis, D.: *Bulletin C 36*, [online]. Paris: IERS EOP PC, Observatoire de Paris, 2008, [cit. 2021-11-29]. Dostupné z: <https://web.archive.org/web/20090412091616/http://tycho.usno.navy.mil/bulletinc2008.html>
- [13] Microsemi, a wholly owned subsidiary of Microchip Technology Inc.: *5071A Primary Frequency Standard Operating and Programming Manual*. Rev. c vydání, str. xiv, October 2006.
- [14] Raspberry Pi Ltd: *Raspberry Pi Documentation – Voltages*, [online]. 2021, [cit. 2021-09-15]. Dostupné z: <https://www.raspberrypi.com/documentation/computers/os.html#voltages>
- [15] NTP Security Project: *The Network Time Protocol suite – Source of refclock_shm.c*, [online]. 2019, [cit. 2021-09-15]. Dostupné z: https://github.com/ntpsec/ntpsec/blob/4490b839cd5a83544b0d0fcf833eab69434c76b5/ntpd/refclock_shm.c#L529
- [16] The Bootstrap Team: *Introduction – Bootstrap v4.5*, [online]. 2015, [cit. 2021-12-08]. Dostupné z: <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
- [17] Tobias Mädler: *Raspberry Pi GPS Locking*, [online]. 2019, [cit. 2021-12-08]. Dostupné z: <https://tbospace.de/rpigpsntp.html>

Seznam použitých zkratek

- BNC** Bayonet Neill–Concelman.
- CAD** Computer-aided design.
- CDN** Content Delivery Network.
- CLI** Command Line Interface.
- CMS** Content Management System.
- CSS** Cascading Style Sheets.
- DNS** Domain Name System.
- GPIO** General-purpose input/output.
- GPS** Global Positioning System.
- HTTP** Hypertext Transfer Protocol.
- IP** Internet Protocol.
- IPC** Inter-process communication.
- LCD** Liquid-crystal display.
- MJD** Modified Julian Date.
- NTP** Network Time Protocol.
- NTS** Network Time Security.
- OCXO** Oven-controlled Crystal Oscillator.
- PFS** Primary Frequency Standard.
- PPS** Pulse Per Second.

PRTC Primary Reference Time Clock.

QoS Quality of service.

RCE Remote Code Execution.

RMS Root mean square.

RPi Raspberry PI.

SCPI Standard Commands for Programmable Instruments.

SHM Shared Memory.

SSH Secure Shell.

TTL Transistor–Transistor logic.

TUI Text User Interface.

UDP User Datagram Protocol.

USB Universal Serial Bus.

UTC Universal Time Code.

XML Extensible Markup Language.

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	bin	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF