

České vysoké učení technické v Praze

Fakulta dopravní

Bc. Marcel Šulek

SYNCHRONIZOVANÉ MĚŘENÍ  
PRO DYNAMICKÉ ZKOUŠKY  
SYSTÉMŮ INTEGROVANÉ BEZPEČNOSTI

2015



**K623 .....Ústav bezpečnostních technologií a inženýrství**

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

**Bc. Marcel Šulek**

Kód studijního programu a studijní obor studenta:

**N 3710 – BI – Bezpečnost informačních a telekomunikačn. systémů**

Název tématu (česky): **Synchronizované měření pro dynamické zkoušky  
systémů integrované bezpečnosti**

Název tématu (anglicky): Synchronized Measurment for Dynamic Testing of  
Integrated Safety Systems

### **Zásady pro vypracování**

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Problematika časové synchronizace více senzorů při měření dynamických veličin
- Specifikace GPS modulu
- Specifikace senzorů kinematických veličin
- Specifikace integrační platformy
- Vyhodnocení používaných metod – řešerše
- Popis architektury softwarového řešení
- Algoritmizace vybraných problémů

Rozsah grafických prací: dle pokynů vedoucího DP

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: PRINZ, Peter, KIRCH-PRINZ Ulla. C Pocket Reference. Sebastopol, USA: O'Reilly Media, Inc, 2002. ISBN: 978-0596004361

HEROUT, Pavel. Učebnice jazyka C. Praha: KOPP, 2009. 280 s. ISBN: 978-80-7232-383-8

Vedoucí diplomové práce: **doc. Ing. Václav Jirovský, CSc.**

Datum zadání diplomové práce: **26. června 2013**

(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **31. května 2015**

- a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
- b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

  
doc. Ing. Václav Jirovský, CSc.

vedoucí

Ústavu bezpečnostních technologií a inženýrství



  
prof. Dr. Ing. Miroslav Svítek

děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

  
Bc. Marcel Šulek

jméno a podpis studenta

V Praze dne.....30. listopadu 2014

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

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

Marcel Šulek  
31. května 2015

**Abstrakt**

Diplomová práce se zabývá možnostmi použití senzorických sítí při měření kinematických veličin. Ústředním tématem je časová synchronizace jednotlivých měřících prvků. Jsou diskutovány jednotlivé synchronizační metody a jejich náročnost z hardwarového, softwarového i ekonomického pohledu.

Práce také ověřuje, zda a do jaké míry je možné jako prvky senzorické sítě využít levné, málo výkonné mikrokontroléry Libelium Waspote a Arduino.

**Klíčová slova**

Synchronizace, časová synchronizace, synchronizační puls, kinematika, zrychlení, GPS, PPS, Atmel, ATmega, Libelium, Arduino.

**Abstract**

Thesis deals with the use of sensor networks in measuring of kinematic values. The focal point is time synchronization of individual network nodes. Various synchronization methods are considered from hardware, software and economical viewpoints.

The work also verifies if, and to what extent it is possible to use inexpensive, low-powered microcontrollers as the nodes of the network.

**Keywords**

Synchronization, Time Synchronization, Synchronization Pulse, Kinematics, Acceleration, GPS, PPS, Atmel, ATmega, Libelium, Arduino.

## Obsah

1.	Problematika časové synchronizace více senzorů při měření dynamických veličin .....	3
1.1.	Scénář měření.....	3
1.2.	Topologie sítí .....	3
1.2.1.	Semi-centralizovaná síť.....	4
1.2.2.	Decentralizovaná síť .....	4
1.3.	Absolutní a relativní časová synchronizace.....	5
2.	Specifikace GPS modulu .....	6
2.1.	Datové rozhraní .....	6
2.2.	Popis jednotlivých modulů.....	7
2.2.1.	u-blox MAX-7C.....	7
2.2.2.	Telit Jupiter JN3 .....	7
2.2.3.	Topcon B110.....	8
2.3.	Doba potřebná pro výpočet pozice.....	8
2.4.	Přesnost a posun dvou PPS signálů.....	12
3.	Specifikace senzorů kinematických veličin .....	13
3.1.	InvenSense MPU-6050.....	13
3.1.1.	Napájení .....	13
3.1.2.	Komunikační sběrnice.....	14
3.1.3.	Rozsah a snímkovací frekvence měření .....	14
3.1.4.	Požadavky na šířku přenosového pásma a přesnost synchronizace integrační platformy .....	15
4.	Specifikace integrační platformy .....	16
4.1.	Mikrokontroléry Arduino Nano a Pro Mini .....	16
4.1.1.	Platforma Arduino .....	16
4.1.2.	Hardwarový popis.....	17
4.1.3.	Softwarové prostředí.....	17
4.1.4.	Architektura měřicí jednotky .....	18
4.1.5.	Možnosti uložení dat .....	18
4.1.6.	Snímkovací frekvence digitálních a analogových pinů.....	22
4.1.7.	Možnosti synchronizace .....	23
4.2.	Libelium Wasp mote.....	31
4.2.1.	Hardwarový popis.....	31
4.2.2.	Softwarové prostředí.....	32
4.2.3.	Možnosti uložení dat .....	32

4.2.4. Možnosti synchronizace .....	34
4.3.    National Instruments CompactRIO .....	35
4.3.1. Hardwarový popis.....	35
4.3.2. Softwarové prostředí.....	36
4.3.3. Možnosti uložení dat .....	36
4.3.4. Možnosti synchronizace .....	37
5.    Vyhodnocení použitých metod .....	42
5.1.    Metody ukládání nebo přenosu dat .....	42
5.2.    Synchronizační metody.....	44
5.2.1. GPS .....	44
5.2.2. Fotodioda a 433MHz rádiová spoušť.....	45
5.2.3. NRF24L01 2,4 GHz Radio .....	45
5.2.4. UART WiFi moduly.....	45
5.2.5. NI cRIO GPS .....	45
5.2.6. NI cRIO NTP .....	45
5.3.    Možné konflikty při kombinování mikrokontrolérových modulů.....	46
5.4.    Mikrokontroléry Arduino a Waspmote.....	46
5.5.    Synchronizované měření dvěma mikrokontroléry.....	47
6.    Popis architektury softwarového řešení .....	49
7.    Algoritmizace vybraných problémů .....	51
7.1.    ESP-8266 Arduino HTTP server .....	51
7.2.    GPS PPS synchronizace v prostředí Linuxu.....	51
8.    Závěr.....	53
9.    Rejstříky .....	54
9.1.    Seznam použitých zkratk .....	54

## 1. Problematika časové synchronizace více senzorů při měření dynamických veličin

S přechodem senzorů na mikroelektromechanickou architekturu začal jejich cenový vývoj kopírovat ceny ostatních mikročipových technologií. I velice přesné senzory je dnes možné zakoupit za jednotky dolarů. Díky tomu přestává být problém měřit kinematiku i menších komponent – prvky zavěšení automobilových kol nebo části deformační zóny při crashtestu.

Se vzrůstajícím počtem senzorů ale roste i složitost vedení datových a napájecích kabelů mezi nimi a sběrnými jednotkami – tradičně průmyslovými počítači. Typický sedan střední třídy jako Škoda Octavia má rozvor 2,5 m a rozchod 1,5 m. Měření zavěšení všech čtyř kol znamená i při velmi efektivní práci návrh vedení, montáž a následnou demontáž nejméně 50 metrů kabelů.

Jedním ze způsobů, jak takové zapojení zjednodušit, je použití více sběrných jednotek. Pokud je však požadováno vyhodnocení jednotlivých měření nikoliv samostatně, ale jako součást systému, tak musí být všechna měření opatřena stejnou časovou osou, se stejným počátečním časem  $t_0$ . Prvním tématem této práce tedy budou právě metody, jak získat – a udržet – na všech zařízeních stejný čas.

Nicméně ani použití více sběrných PC nemusí být optimální řešení. Obvyklý průmyslový počítač je z fyzikálního hlediska poměrně velké a těžké zařízení s nezanedbatelnou spotřebou. Rovněž z ekonomického pohledu je každá jednotka investicí v řádu desítek, někdy i stovek tisíc korun. V některých situacích by přitom bylo možné použít i mnohem jednodušší mikrokontroléry s pořizovací cenou ve stovkách či tisících korun a dostatečně nízkou spotřebou na to, aby mohly být provozovány na baterie.

Na druhou stranu je i u nich celá řada obtíží: kapacita operační paměti se pohybuje v jednotkách kilobajtů, úložného prostoru bývá pod megabajt a takt jediného procesorového jádra se počítá na desítky megahertzů místo dnes obvyklých gigahertzů.

Přitom měření jediné osy digitálním akcelerometrem generuje datový tok 2 kBps a k tomu ještě nejméně jeden kB přidá časová značka. Druhým tématem této práce tedy bude zjištění maximálních limitů mikrokontrolérů a určení postupů a periférií, jak se jim co nejvíce přiblížit.

### 1.1. Scénář měření

V práci budu uvažovat následující scénář měření: jeden až pět automobilů, ve kterých je instalován libovolný počet sběrných jednotek libovolného typu. Začátek i konec měření bude v jednom místě, přičemž zde bude PC určené ke sběru naměřených dat. Preferovaná technologie spojení s měřícím systémem je bezdrátová WiFi síť standardu IEEE 802.11. PC lze považovat za zdroj přesného času, je možné k němu instalovat synchronizační rádiový či optický maják.

V průběhu měření se auta budou vzájemně pohybovat. Nedá se tedy uvažovat se spolehlivým optickým či dokonce fyzickým spojením. Trvalé rádiové spojení je teoreticky možné, avšak nedoporučované z důvodu náročnosti na spotřebu elektrické energie a komplikace s instalací antén.

Pro kinematické senzory platí následující výkonnostní kritérium: cílem je změřit kinematiku zavěšení kol automobilu, pohybujícího se rychlostmi do 70 m/s (250 km/h), s koly o obvodu 2 m, pohybující se po povrchu s nerovnostmi vzdálenými od 0,2 m. Velikost impulsů lineárního zrychlení  $\pm 15$  g.

Požadovaná přesnost synchronizace byla zadána  $\pm 2$  ms mezi libovolnými dvěma zařízeními.

### 1.2. Topologie sítí

Vzhledem k odlišným výkonnostním charakteristikám průmyslových PC a jednočipových senzorů je nutné počítat s odlišnými topologiemi zapojení senzorů a sběrných zařízení. V této práci je budu označovat *semi-centralizovaná* a *decentralizovaná* topologie.



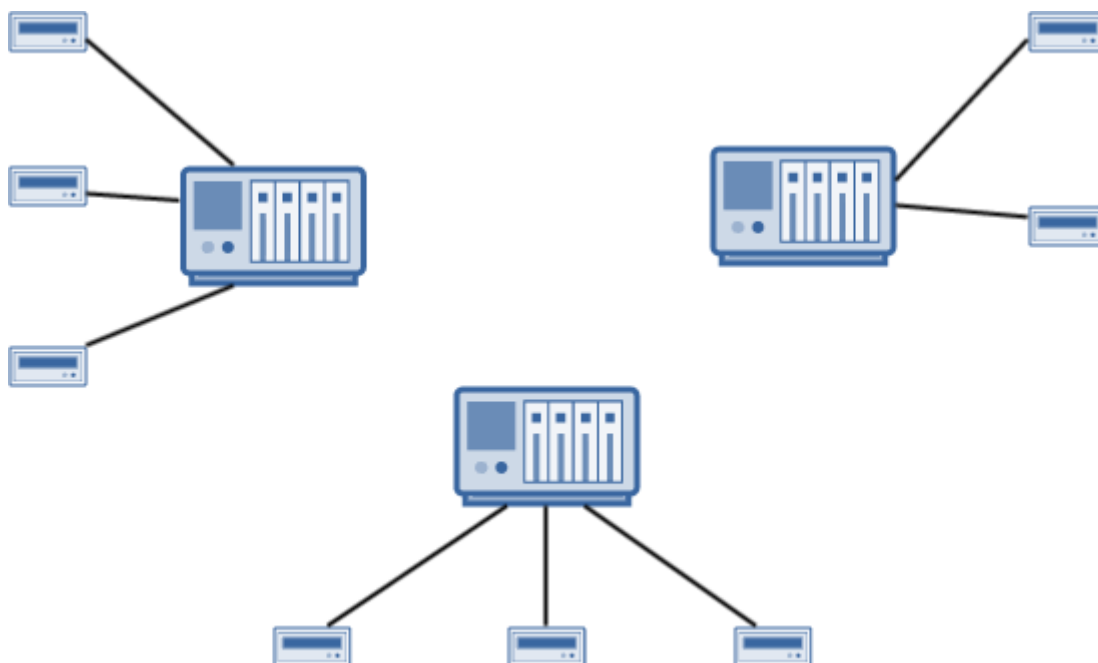
### 1.2.1. Semi-centralizovaná síť

Tímto termínem budu označovat síť senzorů spojených v  $n$  nezávislých hvězdicových topologiích. Středem každé hvězdice je průmyslové PC, v této práci model National Instruments CompactRIO (dále v textu označovaný cRIO). K němu jsou přes sériovou sběrnici SPI připojeny jednotlivé senzory.

Časy měření všech senzorů v jednom systému lze vztáhnout k vnitřnímu času cRIO. Pro účely synchronizace tak dostačuje, když se sjednotí časy pouze centrálních jednotek. To znamená mnohem méně prvků, než v případě decentralizované sítě.

Detailní popis platformy cRIO včetně technologických možností synchronizace bude následovat v kapitole 4. SPECIFIKACE INTEGRAČNÍ PLATFORMY, část 4.3. NATIONAL INSTRUMENTS COMPACTRIO.

Při praktickém měření se obvykle používá jedno cRIO na jeden automobil. V této práci se budu zaměřovat na poměrně nízký počet měřících systémů, typicky do pěti automobilů.



Obrázek 1 – Semi-centralizovaná síť. Několik centrálních jednotek, k nimž je připojeno více senzorů.

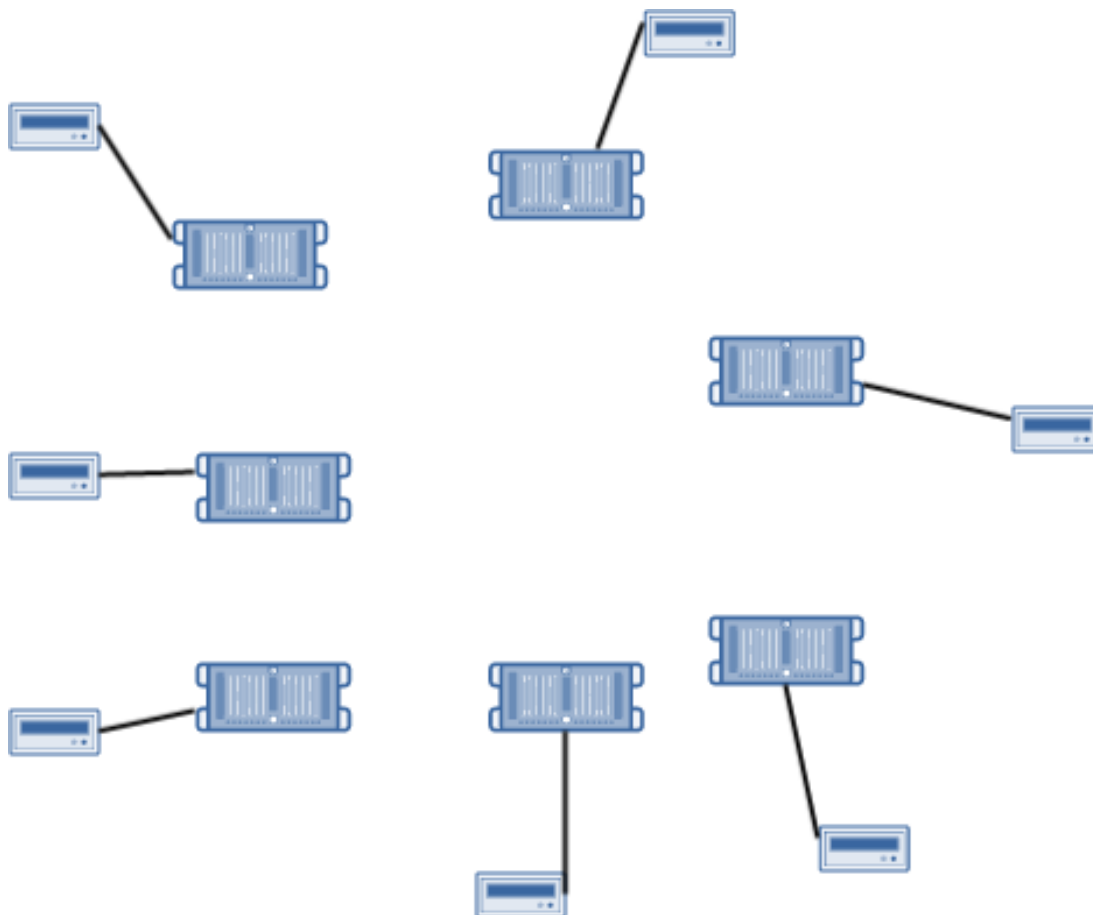
### 1.2.2. Decentralizovaná síť

Decentralizovanou variantu bude představovat síť mnoha jednoduchých systémů mikrokontrolér – senzor – úložné zařízení (buď paměťové médium, nebo rádiová přenosová technika). Tato topologie je na synchronizaci složitější, neboť je třeba dostat synchronizační puls ke každému jednomu mikrokontroléru – tedy prakticky ke každému jednomu senzoru.

Pro měření byly vybrány dva mikrokontroléry:

- 1) Atmel ATmega328 - prodáváný včetně obslužných čipů (regulátor napětí, oscilátor, USB interface a další) pod obchodním označením Arduino, konkrétně jeho varianty Nano a Pro
- 2) Atmel ATmega1281 - nabízený jako komplexní ekosystém senzorických a komunikačních desek pod obchodním označením Libelium Waspmote.

Oba mikrokontroléry budou detailně popsány v částech 4.1 MIKROKONTROLÉRY ARDUINO NANO A PRO MINI, resp. 4.2 LIBELIUM WASPMOTE.



Obrázek 2 – Decentralizovaná síť. K mikrokontrolérům je připojen jediný senzor, mikrokontrolerů je více, než v případě semi-centralizované sítě.

### 1.3. Absolutní a relativní časová synchronizace

Synchronizaci hodin můžeme v prvním přiblížení rozdělit na absolutní a relativní. Absolutní je myšlena synchronizace k všeobecně uznávaným hodinám, jako například UTC nebo GPS čas. Její výhodou je její standardizace a tedy porovnatelnost jakýchkoliv dvou měření pořízených libovolnou technologií a v libovolné vzájemné vzdálenosti (s výjimkou relativistických podmínek, které však vzhledem ke scénáři měření není nutné kompenzovat). Naopak nevýhodou je složitost získání globálního synchronizačního signálu.

Relativní synchronizací je myšlen stav, kdy hodiny zařízení nejdou *přesně*, nýbrž *stejně špatně*. Jedná se tedy o synchronizaci k jednomu času, který však neodpovídá svým počátkem a/nebo časovou jednotkou žádnému standardu. Hodiny tak mohou být posunuté nebo se „předbíhat“. Takováto měření nejsou porovnatelná s jinými, ale na druhou stranu je podstatně jednodušší získat zdroj takového relativního hodinového signálu. *Nějaký* oscilátor má totiž k dispozici každý mikroprocesor.

## 2. Specifikace GPS modulu

GPS (nebo obecněji GNSS) přijímače lze kromě zjištění polohy využít i jako zdroj velice přesného absolutního času. Umožňuje to samotný princip všech GNSS systémů, kdy satelity nepřetržitě vysílají přesný čas a přijímač zjišťuje svoji polohu výpočtem posunu signálů, přičemž vedlejším produktem je i aktuální čas. Tento čas může být použit k synchronizaci libovolného počtu zařízení v libovolné vzdálenosti od sebe a bez nutnosti jakékoliv komunikace mezi těmito zařízeními.

Nevýhodami tohoto přístupu je zejména nutnost poměrně komplexních programů zpracovávajících data z přijímačů a dále i nutnost mít poměrně drahý GNS přijímač na každém jednom zařízení. Za cenu dalšího zvýšení komplexity synchronizačních algoritmů však je možné navrhnout i hybridní systém, kdy je v rámci skupiny zařízení jedno z nich určeno k synchronizaci k absolutnímu GPS času a ostatní se synchronizují na jeho hodiny nějakým jednodušším a levnějším řešením (elektrickým, rádiovým nebo světelným pulsem).

*GPS satelity nevysílají čas v UTC, nýbrž tzv. GPS čas. Jedná se o čas jejich interních atomových hodin a neobsahuje tedy přestupné sekundy přidávané k UTC přibližně každé dva roky. Většina přijímačů ale nabízí automatický převod do UTC.*

Pro účely této práce budu pracovat se třemi GPS moduly: u-blox MAX-7C, Telit Jupiter JN3 a Topcon B110. Přestože se jedná o cenou i výkonem velmi rozdílné přístroje, tak pro účely získání přesného času mají většinu vlastností společných. Jejich detailní popis tedy bude až v závěru kapitoly – v části 2.2. POPIS JEDNOTLIVÝCH MODULŮ. Nyní se budu věnovat sdíleným rysům.

### 2.1. Datové rozhraní

Primárně vstupně-výstupním rozhraním obou zařízení je dvouvodičová sériová linka nastavitelného baud rate. Výstupními daty jsou zprávy standardu NMEA 0183<sup>1</sup>, přičemž je možné nastavit, které typy zpráv se budou či nebudou vysílat. Je tak možné šetřit přenosové pásmo například vynecháním zpráv o příjmu satelitů GSA.

Kromě zpráv standardu NMEA nabízí jednotky u-blox a Topcon i další zprávy proprietárních formátů. Pro účely časové synchronizace jsou důležité věty s časovým razítkem hrany na pinu externího přerušení (EVT, resp. EVENT0 pin).

Data se na GPS přijímači negenerují postupně, ale v „pulsech“ odpovídajících nastavené frekvenci pozičních updatů. V každém cyklu se po vypočítání polohy pošlou na rozhraní všechna data a jsou přenášena v limitech zvolené baud rate. Po vyslání posledního bitu zprávy se na lince neobjeví žádná data až do dalšího pozičního updatu. Nestane se tedy například, že by se jako první vyslala zpráva o viditelných satelitech GSV (která neobsahuje žádné výpočetně náročné informace), následovala vysílací pauza a po vypočtení polohy se vyslala zpráva o fixu GGA. Všechna data se odešlou najednou až po spočítání pozičního fixu.

Kvůli čekání na poziční fix tedy nelze použít samotné datové rozhraní jako zdroj přesného času. Přestože zpráva GGA obsahuje dostatečně přesný čas získání fixu, tak se z definice jedná o zprávu *v tento čas byl přijímač na těchto souřadnicích*, nikoliv *právě teď je přijímač na těchto souřadnicích*. Zároveň není výpočet polohy časově konstantní, ale variuje s měnící se konstelací – plynule s měnící se elevací, skokově s výstupem či sestupem satelitů za horizont.

K získání přesného času tak jsou na obou přijímačích dedikované digitální piny – výstupní PPS a vstupní EVT (resp. EVENT0).

PPS je digitální výstup, na kterém GPS přijímač generuje v pravidelných intervalech hrany, tedy přepnutí z jedné logické úrovně na druhou. Na zařízeních u-blox a Topcon je volitelná frekvence

---

<sup>1</sup> NMEA data. Gpsinformation.org [online]. 2008 [cit. 2015-05-20]. Dostupné z: <http://www.gpsinformation.org/dale/nmea.htm>

časových pulsů, pracovní cyklus (jak dlouho bude trvat změněná logická úroveň), smysl změny (vzestupná či sestupná hrana) a referenční čas (GPS nebo UTC).

Druhou možností jak z přijímačů získat přesný čas pin EVT (EVENT0 na Topcon B110). Jedná se o protiklad PPS pinu, je to vstupní pin, na kterém přijímač čeká na změnu logické úrovně. Pokud ji v daném navigačním cyklu zaznamená, přidá k výstupní dávce dat proprietární zprávu (UBX-TIM-TM2, resp. XA) obsahující nanosekundově přesné časy vzestupné a sestupné hrany. Pokud v navigačním cyklu dojde k více vzestupným/sestupným hranám, je zaznamenána pouze doba poslední hrany daného typu.

## 2.2. Popis jednotlivých modulů

### 2.2.1. u-blox MAX-7C

Jedná se o mid-level GPS přijímač, schopný zpracovat data z GPS nebo GLONASS (nikoliv však oba zároveň) a SBAS diferenciální GPS. Maximální frekvence pozičních updatů je 10 Hz. Vstupně-výstupní rozhraní lze volit ze dvou UART linek, SPI, USB a I<sup>2</sup>C rozhraní. Výstupní data jsou ve standardním formátu zpráv NMEA, případně proprietárním formátu UBX. Nepodporuje RTK výstup kompatibilní s DGPS systémy jako Czepon. Pro časovou synchronizaci nabízí digitální PPS a EVT pin s 3,3V logikou, obojí dle datasheetu s přesností do 1 μs a 5 V tolerantní.

Použitý čip je integrován na breakout desce, dostupné piny jsou napájecí VCC, GND, UART1 TX a RX, I<sup>2</sup>C SDA a SCL, PPS a EVT. Anténa je v této variantě neoddělitelná. Na desce je integrována dobíjecí 3V baterie, která po cca 7 dní udržuje napětí volatilní paměti, do které se ukládají přijaté navigační zprávy (efemeridy, časové korekce apod.). To umožňuje rychlejší, tzv. warm start, v řádu sekund.

Pro urychlení cold startu, tedy situace, kdy je nutné přijmout celou navigační zprávu, zařízení podporuje Asistovanou GPS, tedy stažení navigační zprávy out-of-band (nejčastěji z internetu) a následný upload na přijímač přes UART rozhraní. Zdroj platných navigačních zpráv provozuje samotný u-blox jako službu AssistNow, komunikační protokol je definován v u-blox Application Note<sup>2</sup>, celý postup zprovoznění je pak popsán v dokumentu Getting u-blox MAX-7C GPS to work with Assisted A-GPS<sup>3</sup>.

Nastavení přijímače se provádí zasláním binárních zpráv přes UART rozhraní. Sestavení zpráv je možné podle specifikace<sup>4</sup> nebo vygenerováním v grafickém programu u-center. Jejich integrita je ověřena dvěma 1B kontrolními součty. Příjem zpráv je indikován UBX ACK zprávami.

Další parametry je možné najít v datasheetu<sup>5</sup>. Maloobchodní cena se pohybuje kolem \$15.

### 2.2.2. Telit Jupiter JN3

Podobně jako u-blox je i Telit JN3 přijímač cílený do středního segmentu, nabízí tedy více možností konfigurace než mobilní SoC řešení. Na druhou stranu ale neumožňuje použít pokročilé DGPS technologie jako RTK. Maximální frekvence pozičních updatů je 5 Hz, čip nabízí jedno UART a jedno I<sup>2</sup>C

---

<sup>2</sup> U-BLOX AG. Implementing AssistNow® Online Client for u-blox GPS Receivers: Application Note [online]. C. 2006, 15 s. [cit. 2015-05-15]. Dostupné také z: [http://people.openmoko.org/matt\\_hsu/ImplementationAssistNowServerAndClient\(GPS.G4-SW-05017-C\).pdf](http://people.openmoko.org/matt_hsu/ImplementationAssistNowServerAndClient(GPS.G4-SW-05017-C).pdf)

<sup>3</sup> Getting u-blox MAX-7C GPS to work with Assisted A-GPS. Github: Gist [online]. 2015 [cit. 2015-05-15]. Dostupné z: <https://gist.github.com/veproza/55ec6eaa612781ac29e7>

<sup>4</sup> U-BLOX AG. U-blox 7 Receiver Description Including Protocol Specification [online]. V14. 2013, 209 s. [cit. 2015-05-15]. Dostupné také z: [https://www.u-blox.com/images/downloads/Product\\_Docs/u-blox7-V14\\_ReceiverDescriptionProtocolSpec\\_Public\\_%28GPS.G7-SW-12001%29.pdf](https://www.u-blox.com/images/downloads/Product_Docs/u-blox7-V14_ReceiverDescriptionProtocolSpec_Public_%28GPS.G7-SW-12001%29.pdf)

<sup>5</sup> U-BLOX AG. MAX-7 Data Sheet: u-blox 7 GNSS modules [online]. R05. 2014, 24 s. [cit. 2015-05-15]. Dostupné také z: [https://www.u-blox.com/images/downloads/Product\\_Docs/MAX-7\\_DataSheet\\_%28GPS.G7-HW-12012%29.pdf](https://www.u-blox.com/images/downloads/Product_Docs/MAX-7_DataSheet_%28GPS.G7-HW-12012%29.pdf)

rozhraní pro data a jeden 1PPS pin s 3,3V logickou úrovní. Formát dat je standardní NMEA nebo proprietární SiRF.

Čip je dodáváný k systému Libelium na speciální desce, která je opatřena dvěma čtyřpinovými segmenty, pasujícími na patici na WaspMote. Na segmentech jsou vždy okrajové piny nezapojeny. Vnitřní piny jsou na jedné straně napájecí 3,3 V VCC a GND, na druhé UART RX a TX. Na desce jsou dále „testovací body“ A – D. Na „A“ je vyveden 1PPS pin, na „B“ je I<sup>2</sup>C SDA, na „C“ SCL a na „D“ vstup z případné baterie k napájení volatilní paměti s navigační zprávou.

Pin 1PPS standardně není napájený, jeho funkce je navázána na posílání NMEA věty ZDA (datum a čas), kterou je třeba zapnout v ovládacím programu. K přijímači se dodává placený konfigurační program SiRFLive, požadované nastavení je však možné provést i ve zdarma dostupném SiRFDemo. Nastavení se provádí přes UART sběrnici, při použití SiRFDemo nebo SiRFLive je tedy nutné GPS připojit přes USB-UART převodník k PC. Alternativně je možné nastavovací zprávu zkonstruovat dle specifikace SiRF protokolu<sup>6</sup> a zaslat na přijímač přímo z WaspMote, tuto variantu jsem však nezkoušel.

Pokud je požadováno, aby GPS nadále fungovala s WaspMote, tak je při nastavování NMEA zpráv třeba věnovat pozornost zachování baud rate 4800. Defaultně je v softwaru zvolena hodnota 9600.

Ostatní parametry lze dohledat v hardwarové příručce<sup>7</sup> a uživatelském manuálu<sup>8</sup>.

### 2.2.3. Topcon B110

Přijímač Topcon je určený do zařízení cRIO, kde se bude používat primárně k sledování polohy vozidla. Jedná se o profesionální GNSS zařízení, schopné pracovat se systémy GPS, GLONASS a Galileo. Umožňuje využívat všechny systémy zároveň a díky příjmu více satelitů zpřesňovat vypočtenou polohu. Poskytuje neobvykle vysoký 100Hz update rate. Nabízí dvě RS 232, šest UART, USB a I<sup>2</sup>C rozhraní a možnost přímého připojení SD karty. Bohužel nenabízí sběrnici SPI. Komunikace probíhá ve standardech NMEA a RTK a proprietárním formátu GRIL<sup>9</sup>. Pro časovou synchronizaci nabízí PPS a EVENT0 pin.

Nastavení se provádí přes libovolné sériové rozhraní. Konfigurační příkazy jsou v čitelném ASCII formátu a řídí se specifikací GRIL.

Další parametry jsou popsány v datasheetu<sup>10</sup>.

### 2.3. Doba potřebná pro výpočet pozice

Pokud by byla doba výpočtu pozice *přibližně konstantní*, bylo by možné významně zjednodušit architekturu systému vynecháním měření přes PPS pin. Za tímto účelem jsem provedl měření zpoždění prvního bitu na rozhraní UART za detekci hrany na PPS pinu. Měření proběhlo na přijímači u-blox, který byl připojený na single-board PC Raspberry Pi 2 (dále v textu zkracováno RPi). Nastavená frekvence pozičního updatu byla 1 Hz, stejně jako frekvence PPS pinu.

Zpoždění bylo měřeno na interních hodinách RPi. Z tohoto důvodu jsem prvně určil jejich přesnost měřením intervalů mezi vzestupnými hranami PPS pinu, které by teoreticky měly být vždy 1 s.

<sup>6</sup> SiRF Technology Inc. 2008. SiRF Binary Protocol Reference Manual. Revision 2.4. 128 s. [cit. 2015-05-04]. Dostupné také z: [http://usglobalsat.com/downloads/SiRF\\_Binary\\_Protocol.pdf](http://usglobalsat.com/downloads/SiRF_Binary_Protocol.pdf)

<sup>7</sup> TELIT. 2012. JN3 Hardware User Guide. Rev.0 – 2012-01-20. 31 s.

<sup>8</sup> TELIT. 2012. JN3 EVK User Manual. Rev.0 – 2012-01-20. 34 s

<sup>9</sup> TOPCON POSITIONING SYSTEMS. *GRIL Reference Manual*. 2009

<sup>10</sup> TOPCON POSITIONING SYSTEMS. B110 GNSS Receiver Board: Integrator's Guide [online]. Rev. A. 2013, 130 s. [cit. 2015-05-15].



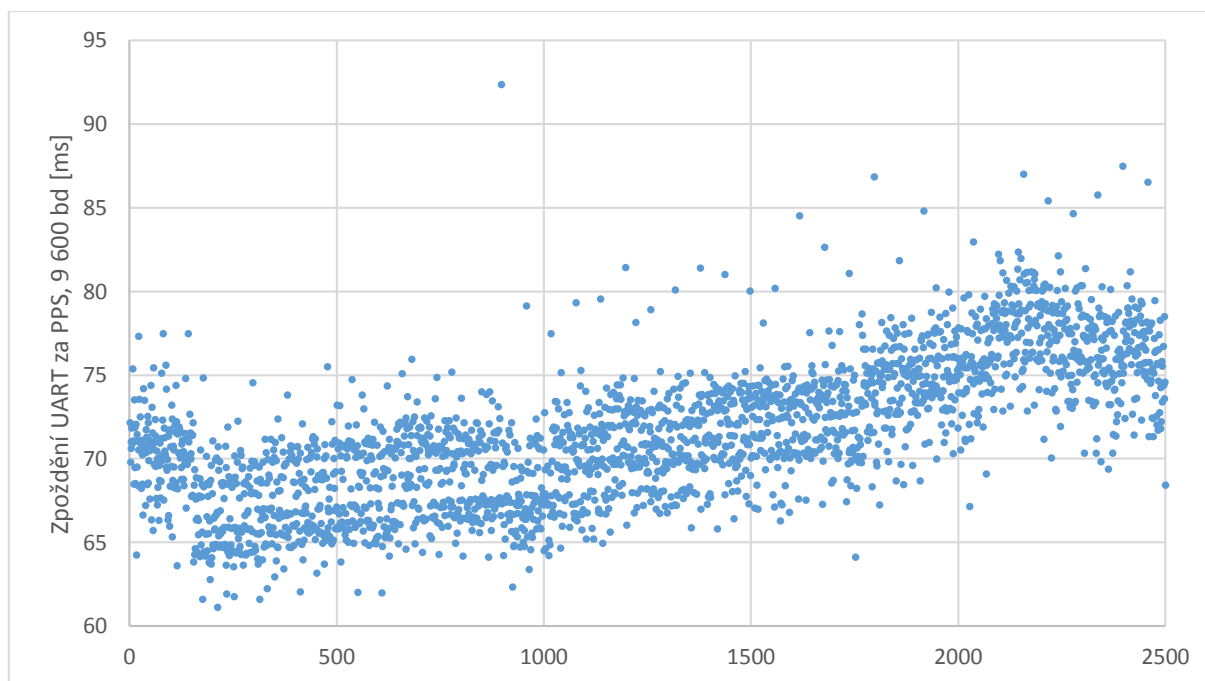
Obrázek 3 – Měření vteřinové odchylky vnitřních hodin RPi. Zobrazeno je 1543 hodnot, měření probíhalo přibližně 26 minut.

	100%	99,99%	99,9%	99%	95%	90%	50%	COUNT
MAX	6,0	5,8	5,0	4,1	2,9	2,1	-1,0	1543
MIN	-7,2	-7,2	-7,0	-5,0	-4,1	-3,1		
DIFF	13,1	13,0	12,0	9,1	6,9	5,2		

Tabulka 1 – Rozložení hodnot po vybraných percentilech (hodnoty v  $\mu\text{s}$ ).

Nejvyšší naměřené zpoždění bylo 6  $\mu\text{s}$ , předcházely se nejvíce o 7  $\mu\text{s}$  – rozdíl mezi nimi je 13  $\mu\text{s}$ . Měření doby výpočtu pozice tedy lze očekávat přesné na desetiny milisekund.

Měření zpoždění dat za PPS pulsem proběhlo za dvou nastavení baud rate - standardních 9600 (měření 1) a maximálních 115 200 (měření 2). Měření 1 bylo zahájeno poté, co přijímač získal fix, a trvalo 41 minut. Následovala několikahodinová pauza (při které ale byl přijímač v provozu) a následně měření 2 v délce necelých 90 minut. Při obou měřeních měl přijímač výhled na přibližně polovinu nebeského horizontu a byl po celou dobu statický. Naměřené variace jsou tedy pouze v důsledku změn konstelace. Nebyly použity funkce, které by dále ztížily výpočet pozice, jako například zpřesnění polohy pomocí SBAS (v Evropě systém EGNOS).



Obrázek 4 – Měření zpoždění prvního bitu na sběrnici UART od detekce hrany na PPS pinu. Graf je podseknutý (začíná na 0,06 s) pro větší názornost. Zobrazený úsek je 41 minut, 39 sekund (2500 měření).

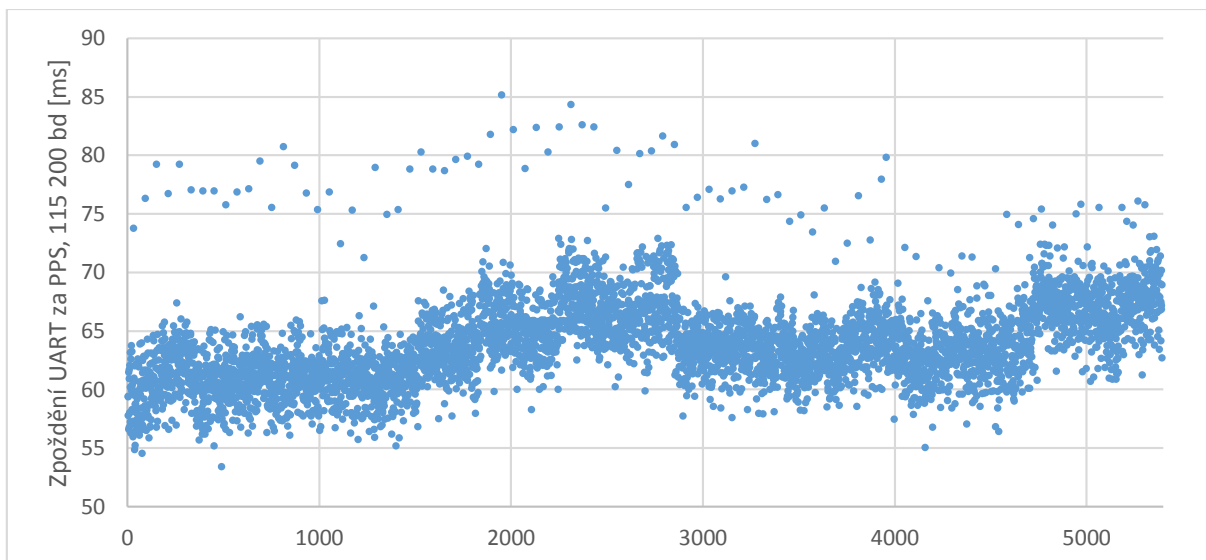
	100%	99,99%	99,9%	99%	95%	90%	50%	COUNT
MAX	92,3	91,1	86,9	81,1	78,7	77,4	71,2	2500
MIN	61,1	61,2	61,7	63,9	65,3	66,2		
DIFF	31,2	29,9	25,2	17,2	13,4	11,2		

Tabulka 2 – Rozptyl hodnot po vybraných percentilech (hodnoty v ms).

Na průběhu měření je možné si všimnout následujícího:

- Výpočet polohy trvá přibližně 60 – 90 ms. To je 60 – 90 % časového rozpočtu daného maximální obnovovací frekvencí 10 Hz (intervalem 100 ms).
- Doba mezi dvěma i po sobě následujícími měřeními variuje o přibližně 3 ms.
- Každých 60 sekund má výpočet přibližně 5 ms zpoždění.
- Skokově (zde kolem měření 133) může dojít ke změně „střední hodnoty“ o více než 5ms.
- Střední doba výpočtu se v čase plynule posouvá až o 10 ms (tedy 10 % maximální obnovovací frekvence).
- Rozdíl mezi nejrychlejším a nejpomalejším výpočtem je 31,2 ms. Neuvažujeme-li výrazný extrém v bode 897, který mohl být způsoben chybou měření, pak je rozdíl 26,4 ms.

K podobným závěrům lze dojít i na základě druhého měření na vyšší baud rate. Připomínám, že metodika měření měřila první bit na sběrnici, nikoliv přijetí celé sady navigačních zpráv.



Obrázek 5 – Měření zpoždění prvního bitu na sběrnici UART při 115 200 Bd. Měření probíhalo s několikahodinovým odstupem za předchozím měřením. Graf je podsekutý pro větší názornost. Zobrazený úsek je 89 minut, 52 sekund (5392 měření).

	100%	99,99%	99,9%	99%	95%	90%	50%	COUNT
MAX	92,3	91,1	86,9	81,1	78,7	77,4	71,2	2500
MIN	61,1	61,2	61,7	63,9	65,3	66,2		
DIFF	31,2	29,9	25,2	17,2	13,4	11,2		

Tabulka 3 – Rozložení hodnot po vybraných percentilech (hodnoty v ms).

- Zpoždění mezi jednotlivými měřeními variuje opět o přibližně 3 ms.
- Každou minutu se opět vyskytují maxima o velikosti 5 až 12 ms.
- Je vidět několik skokových změn střední hodnoty zpoždění o přibližně 5 ms.
- Rozdíl mezi nejrychlejším a nejpomalejším výsledkem byl 31,8 ms.
- Rozdíl mezi nejvyšším a nejnižším krátkodobým (z 30 měření) mediánem je přibližně 9 ms.

Každominutové maximum je pravděpodobně specifikum přijímače a je softwarově filtrovatelné. Variaci mezi jednotlivými měřeními lze odstranit vícenásobným měřením a následným použitím vhodné statistické funkce.

Protože hodnoty vykazují přibližně normální rozložení, je takovou funkcí medián. Za kritickou hodnotu pro určení přesnosti UART-only přístupu tedy považuji **rozdíl krátkodobých mediánů**. Ten v případě GPS u-blox dosahoval přibližně **10 ms ± 1ms**. U přijímače Topcon B110, který je teoreticky 10 × rychlejší (100Hz frekvence pozičních updatů), lze tedy očekávat rozptyl pouhé 1 ms.

Na druhou stranu je možné, že řádově rychlejší zařízení bude mít menší výkonovou rezervu. Je také nutné předpokládat, že případné využití DGPS schopností přijímače vnese do synchronizace další zdroj zpoždění. Vzhledem k nemožnosti otestovat reálný přístroj nezbyvá než konečný rozptyl kvalifikovaně odhadnout na **3 ms ± 1ms**.

Měření byla provedena za účelem zjištění realizovatelnosti varianty zapojení GPS modulů, která vědomě nevyužívá jejich plného potenciálu, zato je jednodušší na implementaci. Metodika měření je proto také jednodušší – přijímač byl v průběhu celého měření v klidu, ale měl velmi neoptimální výhled na oblohu. Neanalyzoval jsem počet satelitů, ze kterých se poloha počítala a ani průběh jednotlivých odstupů signálu a šumu.

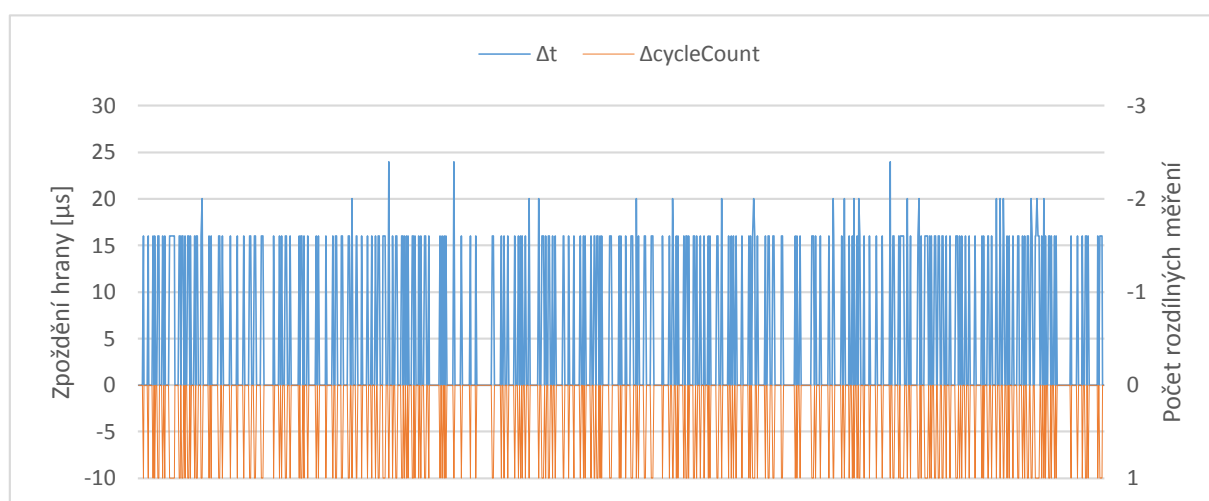
Surová naměřená data jsou k dispozici v souboru [data/gps-delays.xlsx](#). Zdrojový kód použitého skriptu je v souboru [sources/gps-delay-meter.py](#).



## 2.4. Přesnost a posun dvou PPS signálů

K ověření přesnosti a případného posunu signálů ze dvou zdrojů jsem zvolil přijímače u-blox MAX-7C a Telit Jupiter JN3. Výstup z PPS pinů obou jednotek byl přiveden na dva IO piny mikrokontroléru Arduino Nano. Na něm byl spuštěn program detekující hrany na obou pinech a počítající časový rozdíl mezi dvěma vzestupnými hranami. Zároveň byla kontrolní smyčka napsána tak, aby kromě časového rozdílu reportovala i konkrétní počet měření, o kolik dříve detekovala hranu na jednom přístroji oproti druhému.

Před započítáním měření byly oba GPS přijímače připojeny k napájení a ponechány přibližně půl hodiny, aby získaly fix. Vzhledem k omezenému výhledu na nebeský horizont (měření probíhalo na parapetu okna) ztratil v průběhu měření přijímač u-blox několikrát fix, což se projevilo tím, že přestal vysílat PPS pulsy. Tyto výsledky byly rozpoznány díky stejnému časovému razítku posledního fixu a byly z měření odfiltrovány. Z celkových 2 964 měření bylo takto odstraněno 91 měření - zbylo tedy 2 874 vzorků.



Obrázek 6 – Prvních 1 000 měření vzájemného posunu PPS signálů přijímačů u-blox a Telit.

	TOTAL	$\Delta t$				$\Delta cycleCount$	
		0	16	20	24	0	1
COUNT	2873	2002	811	53	7	2002	871
PERCENT	100,0%	69,7%	28,2%	1,8%	0,2%	69,7%	30,3%

Tabulka 4 – Procentuální zastoupení naměřených hodnot.

Nenulové zpoždění bylo naměřeno v 871 případech (30,3 %), nejčastěji 16  $\mu s$ . Při všech naměřených zpožděních byla neshoda detekována pouze v jediném cyklu a v následujícím již byla vzestupná hrana i na druhém kanálu.

Pro ověření bylo následně provedeno měření, kdy byl na oba piny přiveden stejný signál a i při něm docházelo v 30 % případů k posunu o jedno měření. Detekovaná zpoždění 16  $\mu s$  – 24  $\mu s$  tak lze uznat jako chybu metody.

Z provedeného měření tedy můžeme tvrdit, že posun PPS signálů je menší nebo roven 24  $\mu s$  a že nemá kumulativní charakter. Při jejich použití je limitujícím faktorem přesnost mikrokontroléru ATmega328.

Naměřená data jsou k dispozici v souboru [data/gps-pps-slip.xlsx](#). Zdrojový kód měřícího skriptu v [sources/gps-timer.ino](#).

### 3. Specifikace senzorů kinematických veličin

Parametry senzorů lze v prvním přiblížení rozdělit na kvalitativní a kvantitativní. Kvalitativní – zejména přesnost a úroveň šumu – jsou sice zásadní pro samotné měření, ale z pohledu sběrných jednotek nepředstavují faktor. Počítači vcelku nezáleží na tom, zda mu po sběrnici přišlo číslo přesné, či plné šumu. Naopak potřebujeme vědět, jak budou ta čísla velká a jak často budou přicházet – tedy velikost jedné datové věty a snímkovací frekvenci senzoru. Vynásobením těchto hodnot poté dostaneme datový tok, který je nutné průběžně zpracovávat.

Datovou větu i snímkovací frekvenci na akcelerometrech určují ADC (analog-digital, signál-číslo) převodníky, ať již jsou umístěny přímo na čipu (v případě digitálních senzorů) nebo jsou využity ADC sběrné jednotky (při použití analogových senzorů). Nejčastějšími rozlišeními ADC jsou 10 a 16 bitů, přičemž 16b ADC jsou použity v systémech cRIO, Arduino i Waspnote. Běžně dostupné maximum pak je 32 bitů, vzhledem k typickým hodnotám šumu však jejich použití nemá praktický význam. Snímkovací frekvence ADC se pohybují v rozmezí 1 – 100 kHz<sup>11</sup>, frekvence digitálních akcelerometrů pak běžně končí kolem 4 kHz<sup>12</sup>.

V případě digitálních senzorů je pak ještě třeba rozlišovat komunikační sběrnice, přičemž nejčastěji se využívají sériové I<sup>2</sup>C nebo SPI. SPI podporuje systém cRIO i čipy Atmel, sběrnici I<sup>2</sup>C pak standardně pouze Atmel. V cRIO je možné ji implementovat v FPGA nebo zakoupit dedikovaný I<sup>2</sup>C modul.

Pro účely této práce budu vycházet ze zadání měření automobilu pohybujícího se rychlostí 70 m/s po povrchu s nerovnostmi každých 0,2 m, od kterých dostává rázy do  $\pm 16$  g. Základní frekvence kmitání nápravy tedy bude 350 Hz. Abychom ji změřili, tak je nutné mít snímkovací frekvenci nejméně  $2,5 \times$  větší<sup>13</sup>, tedy 875 Hz. V případě vyšších harmonických frekvencí je výpočet analogický.

Parametry měření základní frekvence splňuje například v úvodu zmiňovaný akcelerometr InvenSense MPU-6050. V následující části jej podrobně představím a po zbytek práce bude sloužit jako ukázkový senzor pro systémy Arduino.

#### 3.1. InvenSense MPU-6050

MPU-6050 je MEMS integrované zařízení sdružující tříosý akcelerometr, tříosý gyroskop a digitální pohybový procesor (Digital Motion Processor, dále v textu DMP). Dodává se jako SMD čip o hranách 4×4 mm a výšce 0,9 mm. V této práci je použita varianta na breakout boardu s piny napájení, primární a pomocné I<sup>2</sup>C sběrnice, přerušení a nastavení posledního bitu I<sup>2</sup>C adresy. Cena této desky se pohybuje kolem \$2. K sehnání je i dražší varianta, která na jedné desce navíc integruje i tříosý magnetometr (kompas) a tlakoměr (výškoměr). Její cena se pohybuje kolem \$7.

##### 3.1.1. Napájení

Modul je napájený 3,3 V. Se stejnou hodnotou pracuje i jeho logika, která je i 5V tolerantní. Pracovní proud se pohybuje od 500  $\mu$ A při provozu pouze akcelerometru, přes 3,6 mA při zapnutém pouze gyroskopu až po maximální 3,9 mA při zapnutém akcelerometru, gyroskopu a DMP (všechny hodnoty platné při maximální snímkovací frekvenci zařízení). Klidová spotřeba samotného čipu bez zapnutých senzorů činí 5  $\mu$ A.

<sup>11</sup> Analog-to-Digital Converters. Linear Technology [online]. 2015 [cit. 2015-05-15]. Dostupné z: [http://www.linear.com/products/Analog-to-Digital\\_Converters\\_\(ADC\)](http://www.linear.com/products/Analog-to-Digital_Converters_(ADC))

<sup>12</sup> MEMS Accelerometers. Analog Devices [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://www.analog.com/en/products/mems/mems-accelerometers.html>

<sup>13</sup> Sampling Rate: Sound and Vibration. National Instruments: Manuals [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://zone.ni.com/reference/en-XX/help/373398C-01/svaconcepts/svsamplingrate/>

### 3.1.2. Komunikační sběrnice

Čip komunikuje po sběrnici I<sup>2</sup>C. I<sup>2</sup>C je 400kbps multi-master sériová dvouvodičová linka, s jedním datovým a jedním časovým vodičem. Zařízení mohou být v režimu *master* nebo *slave*, přičemž specifikace umožňuje zařízení změnit svoji roli bez odpojení. Sběrnice je half-duplexní – v jednu chvíli může vysílat jen jedno zařízení (a přitom řídit jak data, tak čas). Specifikace umožňuje *clock stretching*, tedy snížení frekvence časování (a tím i snížení šířky pásma), pokud vysílací zařízení ještě nemá k dispozici další bit. Časový vodič tak nahrazuje i vodič *Ready to Send* např. ze standardu RS232. Adresace jednotlivých prvků je řešena unikátní 7bitovou adresou každého připojeného zařízení.

MPU-6050 má dvě I<sup>2</sup>C sběrnice – hlavní a pomocnou. Na hlavní sběrnici funguje vždy jako *slave* s adresou x3B nebo, pokud je na pinu AD0 nastavená logická 1, x3C. Pin AD0 tedy umožňuje současný běh dvou zařízení MPU-6050 na jedné sběrnici. Pomocná sběrnice, na které funguje vždy jako *master*, umožňuje připojit další senzory a jejich výstupy hardwarově zpracovávat a integrovat na DMP procesoru (tato sběrnice se využívá na zmíněné variantě s magnetometrem a tlakoměrem).

K samotným datům se přistupuje registrovým způsobem. *Slave* zařízení drží mapu 128 adres, v každé je 8 b (1 B) dat. *Master* zařízení vyšle zprávu, ze kterého registru chce číst a z kolika následujících registrů chce data. *Slave* na tuto zprávu odpoví požadovanými daty. V případě MPU-6050 se po odeslání dat z registru tento nastaví na x00 a v tomto stavu zůstane, dokud nejsou k dispozici nová data. Všechny výstupy (všechny osy akcelerometru i gyroskopu) mají rozsah 2B (v big-endianovém pořadí) – zabírají tedy dvě adresy registrů. Adresy pro výstupy akcelerometru jsou x3B, x3D a x3F pro osy X, Y a Z, pro gyroskop pak x43, x45 a x47 ve stejném pořadí. Na adrese x41 je k dispozici výstup teplotního čidla, které slouží pro interní teplotní kompenzaci.

Výstupům senzorů připojených k pomocné I<sup>2</sup>C sběrnici jsou přiřazeny adresy od x49 po x60. Na adrese x6B je zapisovatelné nastavení řízení spotřeby. Nejjednodušší nastavení – hodnota x00 – zapne všechny senzory na maximální výkon. Detailnější rozbor je k dispozici v [mapě registrů](#)<sup>14</sup>.

Na adrese příslušné dané veličině je vždy nejnovější naměřená hodnota (případně zmíněna x00 jako indikace, že novější není k dispozici), což by při požadavku na získání každé naměřené hodnoty vyžadovalo pravidelné dotazování s periodou snímkovací frekvence. To by u komplexnějších non-realtime systémů mohl být problém, proto pohybový procesor nabízí konfigurovatelný FIFO buffer o velikosti 1024 B. Novou hodnotu v něm signalizuje nastavením logické 1 na pinu přerušení.

### 3.1.3. Rozsah a snímkovací frekvence měření

#### Akcelerometr

Akcelerometr měří zrychlení v metrech za sekundu přepočtených na povrchové gravitační zrychlení ( $g$ ,  $1g = 9,81 \text{ m/s}^2$ ). Výstupní hodnota je 16bitová se znaménkem v big-endianovém uspořádání. Rozsah lze volit ze čtyř nastavení:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  a  $\pm 16g$ . Nejmenší měřitelná jednotka tedy je  $1/16384g$ ,  $61 \mu g$ . Udávaná kalibrační přesnost je  $\pm 3 \%$  hodnoty. Snímkovací frekvence je nastavitelná od 4 Hz po 1 kHz. Celou specifikaci je možné najít v datasheetu.<sup>15</sup>

#### Gyroskop

Gyroskop měří úhlovou rychlost ve stupních za sekundu ( $^\circ/\text{s}$ ), výstupní hodnota je 16bitová se znaménkem v big-endianovém uspořádání. Rozsah lze zvolit ze čtyř nastavení:  $\pm 250 \text{ }^\circ/\text{s}$ ,  $\pm 500 \text{ }^\circ/\text{s}$ ,  $\pm 1000 \text{ }^\circ/\text{s}$  a  $\pm 2000 \text{ }^\circ/\text{s}$ . Nejvyšší dosažitelná přesnost tedy je  $1/131 \text{ }^\circ/\text{s}$ . Celkový udávaný RMS šum dosahuje  $0,05 \text{ }^\circ/\text{s}$ . Snímkovací frekvence je nastavitelná od 4 Hz po 8 KHz.

<sup>14</sup> INVENSENSE. MPU-6500 Register Map and Descriptions. Rev. 2.1. 16. 9. 2013, 47 s. [cit. 2015-05-05]. Dostupné z: <http://www.invensense.com/mems/gyro/documents/RM-MPU-6500A-00.pdf>

<sup>15</sup> INVENSENSE. MPU-6000/MPU-6050 Product Specification. Rev. 3.4. 19. 8. 2013, 52 s. [cit. 2015-05-05]. Dostupné z: <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>

#### 3.1.4. Požadavky na šířku přenosového pásma a přesnost synchronizace integrační platformy

Akcelerometr má snímkovací frekvenci 1 kHz a každá osa představuje 16bitový payload. Jedna osa měření tedy při využití maximálního potenciálu senzoru vyžaduje šířku pásma **16 000bps** (15,625 Kbps). Tuto šířku pásma (a schopnost zpracování takového přítoku dat) je nutné dodržet na rozhraní senzor – mikrokontrolér. Na rozhraní mikrokontrolér – úložné zařízení je nutné připočítat ještě overhead v podobě časové značky. Zároveň – pokud je vyžadována přesná synchronizace – je nutno dosáhnout odchylky hodin jednotlivých mikrokontrolérů menší než  $0,5 \times (1/f)$  - tedy **500  $\mu$ s**.

## 4. Specifikace integrační platformy

Za integrační platformu pro účely této práce považuji síť zařízení s následujícími zodpovědnostmi:

- sběr dat z připojených senzorů
- uložení naměřených dat a jejich časové souslednosti tak, aby se daly po skončení měření dále zpracovat
- synchronizace hodin jednotlivých zařízení

Dále je třeba brát v úvahu jejich použití – měření parametrů několika nezávislých automobilů na testovací trati s trváním jednotlivých měření do 30 minut. Z toho plyne požadavek na relativní energetickou nenáročnost. Všechny prvky by měly být schopné po dobu měření pracovat z baterie, případně z palubní elektrické sítě – jejich příkon by tedy neměl přesáhnout jednotky wattů.

Dalším rysem je vzdálenost aut, resp. jednotlivých zařízení – budou v relativní blízkosti (jednotky metrů) pouze při začátku a konci měření. V jeho průběhu se mohou vzdálit až na jednotky kilometrů. To prakticky znamená, že nelze počítat s trvalým spojením všech zařízení, protože konstrukce spolehlivého a všesměrového bezdrátového spojení na takové vzdálenosti by byla příliš složitá a nákladná. Tento bod implikuje, že buď budou hodiny jednotlivých zařízení dostatečně přesné, aby udržely synchronizaci v požadované toleranci po celou dobu měření, nebo bude použita synchronizační metoda založená na nějakém třetím, nezávislém a dostatečně výkonném vysílači synchronizačních pulsů (např. systémy GNSS).

V této kapitole detailně popíšu relevantní vlastnosti tří platform: Arduino – levné open hardware platformy založené na čipu Atmel ATmega328, Libelium Waspmote – vycházející z Arduina a využívající výkonnější ATmega1281 a National Instruments CompactRIO – enterprise řešení postavené na dvoujádrovém 667MHz ARM Cortex-A9 a Xilinx 7 FPGA.

### 4.1. Mikrokontroléry Arduino Nano a Pro Mini

Obě desky, postavené na open hardware platformě Arduino, jsou téměř totožné. Liší se prakticky pouze přítomností USB portu na verzi Nano. Následující odstavce se tedy budou vztahovat na obě varianty, pouze v samotném závěru sekce popíšu jejich rozdíly.

#### 4.1.1. Platforma Arduino

Samotné Arduino je italská firma spravující Arduino Bootloader (běžové prostředí kódu na ATmega). Navrhuje, vyrábí a prodává většinu Arduino desek, udržuje vývojové prostředí Arduino IDE a spravuje vývojářskou komunitu. Není to však výrobce komponent v tradičním slova smyslu, protože všechny Arduino produkty jsou poskytovány pod otevřenou licencí – open hardware i open software. Kdokoliv tedy může přijít a stáhnout si návrhy plošných spojů a osazených komponent, upravit je a vyrobit svoji vlastní vylepšenou „Arduino“ desku (nebo je naopak neupravit a vyrobit vlastní klon).

Tato otevřenost dala za vznik dvěma proudům. V prvním mezi sebou začala soupeřit řada zejména čínských výrobců komponent, a protože většinou soupeří toliko cenou, dají se dnes Arduino kompatibilní desky sehnat za o málo více, než výrobní náklady – použité Nano se běžně prodává za ceny okolo \$2,70, levnější Mikro za \$2,10.

Druhým proudem jsou specializované, od „mateřského“ Arduina odvozené desky (jedním z příkladů je LIBELIUM WASPMOTE, popsané v další části této práce). Využívají Arduino Bootloader a upravené Arduino IDE a přímo na desce integrují moduly ke snazšímu splnění různých úkolů – komunikační, sensorické nebo například spravující nabíjení baterie. Ve většině parametrů zůstávají Arduino kompatibilní a závěry této práce tedy mohou být aplikovány i na tyto odvozené platformy.

#### 4.1.2. Hardwarový popis

##### *Atmel ATmega328*

Jádrem desek Arduino je mikrokontrolér ATmega328. Jedná se o osmibitový RISC procesor o taktu až 20 MHz (většina desek však používá oscilátory na úspornějších 16MHz), nabízí 32 kB vnitřní flash (programové) paměti (z ní 2 kB zabírá bootloader), 1 kB EEPROM (dostupná při runtime) a 2 kB SRAM (operační) paměti. Přímě na čipu je dále řada rozhraní zmíněných v části ROZHRANÍ. Další specifikace lze najít v datasheetu <sup>16</sup>.

##### *Napájení*

Specifikované napětí desky Arduino Nano je 5 V. Stejná je i hodnota napětí na digitálních pinech při logické 1. Proud vyžadovaný pro běh samotné desky je při napájení přes 5V pin přibližně 18 mA (pro běh na baterie nabízí ATmega režimy snížené spotřeby, ve kterých se dá dostat na podmiliampérové proudy). Pro případ, kdy není k dispozici regulované 5V napětí, je na desce integrovaný regulátor schopný zpracovat napětí 6 – 20 V (doporučováno je 7 – 12 V bez dodatečného chlazení). Jedná se však o poměrně neefektivní zařízení, je tedy nutné počítat s nárůstem spotřeby o přibližně 10 mA.

Verze Nano může být napájena přímo z USB 5V vodiče. Použitá verze Pro Mini je spotřebou i regulátorem napětí totožná, její napětí a logická úroveň je však 3,3 V.

##### *Rozhraní*

Jedním z důvodů, proč byl pro platformu Arduino zvolen ATmega328, je jeho on-chip podpora řady rozhraní. Nabízí 14 digitálních I/O pinů za běhu rekonfigurovatelných na výstup, vstup nebo vstup s pull-up 20kΩ rezistorem. Piny 0 a 1 mohou dále fungovat jako hardwarová UART linka s maximální přenosovou rychlostí 2 000 000 Bd (ostatní piny lze omezeně použít jako software UART). Piny 2 a 3 mohou být použity jako generátor přerušování při detekci hrany, piny 3, 5, 6, 9, 10 a 11 nabízí 8bitové PWM a piny 10, 11, 12 a 13 fungují i jako SPI sběrnice. Na pinu 13 je připojená jedna LED. Maximální proud na každém pinu je 40 mA.

Čip nabízí i šest analogových vstupních pinů s 10bitovým rozlišením, 9 600 Hz snímkovací frekvencí a za běhu volitelným referenčním napětím 5V, 1,1V nebo externím napětím přivedeným na pin AREF (v rozsahu 0 – 5V). Analogové piny 4 a 5 jsou dále použity pro sběrnici I<sup>2</sup>C jako SDA a SCL.

##### *Rozdíl mezi Nano a Pro Mini*

Jediný rozdíl mezi dvěma použitými deskami je – kromě velikosti – přítomnost FTDI USB-to-UART převodníku na větším Nano. Zatímco Nano lze z PC programovat přes na desce integrovaný USB konektor, ovládání Pro Mini vyžaduje externí USB UART převodník. Tento by navíc měl mít vyveden kromě běžných RX a TX pinů ještě pin DTR (Data Terminal Ready). Arduino totiž „poslouchá“ programovací příkazy pouze první vteřinu po zapnutí, poté je sériová linka předána spuštěnému kódu. Pro nahrání nového programu je tak třeba desku restartovat a krátce nato na ni začít posílat data. Toto časování zajišťuje programovací prostředí Arduino a reset se realizuje právě přes DTR pin. Sice je možné resetování provést manuálně, vyžaduje to však vcelku přesné načasování stisku tlačítka pro upload v programovacím prostředí a tlačítka pro reset na desce. V praxi se tedy spíše vyplatí si připlatit přibližně 10 Kč za složitější převodník.

Použitý model Pro Mini je ještě odlišný svojí 3,3V logickou úrovní, existují však i 5V verze.

#### 4.1.3. Softwarové prostředí

Spustitelné programy se programují v C++, případně v jazyce Arduino, což je v zásadě zjednodušená nástavba C++. Jako vývojové prostředí (IDE) se dá použít Arduino IDE, jedná se však o velmi základní prostředí nenabízející téměř nic kromě editace textu, zvýrazňování syntaxe a automatického

<sup>16</sup> ATMEL. ATmega 48PA, 88PA, 168PA, 328PA datasheet. Rev. 8161D–AVR–10/09. 2009, 448 s. [cit. 2015-05-14]. Dostupné z: <http://www.atmel.com/Images/doc8161.pdf>



odsazování kódu. Nenabízí pokročilé funkce jako napovídání jmen či parametrů funkcí, statickou typovou kontrolu před kompilací nebo pomocníky pro refaktoring. Jeho největší výhodou je snadný přístup ke kompilaci a uploadu programu a integrovaný monitor sériové konzole.

Pro větší projekty je možné a doporučené použít specializovaná IDE jako Eclipse nebo Visual Studio, pro něž jsou nabízeny Arduino pluginy. V případě Visual Studia je k dispozici i placený development kit, který nabízí mimo jiné on-chip debugging, tedy možnost například vzdáleně krokovat nebo sledovat hodnoty proměnných.

#### 4.1.4. Architektura měřící jednotky

Na rozdíl od platformy CompactRIO je Arduino mnohem menší, s minimální spotřebou, zato však mnohem méně výkonné. Jak bude dokázáno později v části 4.1.5. MOŽNOSTI ULOŽENÍ DAT, Arduino je schopné spolehlivě zpracovat datový tok kolem 10 kBps. V případě použití výkonných senzorů s kilohertzovou a vyšší vzorkovací frekvencí je tedy ideální tvořit uzavřené trojice senzor-arduino-úložné zařízení (např. čtečka SD karet).

To s sebou nese výhodu v podobě kompaktnosti jednotlivých „balíčků“. Na jednom místě je třeba pouze kinematického senzoru, desky Arduino, čtečky SD karty, synchronizačního modulu a baterie. Rozměry takovéto soustavy nepřesáhnou jednotky centimetrů na šířku a délku a jednotky milimetrů na hloubku. Navíc je v případě dlouhodobějšího používání této koncepce možné navrhnout a vyrobit vlastní desku integrující některé nebo všechny komponenty. Zcela zde pak odpadá nutnost složitého vedení kabelů od centrální jednotky ke všem periferním senzorům.

Na druhou stranu tato decentralizace přináší i komplikace – je nutné synchronizovat více jednotek, tedy dostat synchronizační puls a potvrzení jeho přijetí k více přijímačům. V porovnání s cRIO je zde také komplikovanější stažení naměřených dat ze všech senzorů do jednoho centrálního úložiště k vyhodnocení. Oběma problémy se zabývám v následujících podkapitolách.

#### 4.1.5. Možnosti uložení dat

Na rozdíl od systému cRIO, který může být vybaven několikagigabajtovým pevným diskem, má čip ATmega328 k dispozici pouhý 1 kB paměti EEPROM. Při datovém toku minimálně 4kBps je tedy možné na samotné zařízení uložit maximálně čtvrt vteřiny dat, takže k dosažení použitelných dob měření je nutné použití periférií. Vzhledem ke scénáři měření jsou k dispozici dvě možnosti: uložení na SD kartu přímo na zařízení nebo odeslání dat po rádiových vlnách.

Hlavní omezující faktor při výběru metody na uložení/odeslání dat bude latence. ATmega má celkově k dispozici 2 kB SRAM, z nich však významně uberou potřebné knihovny a na samotnou práci je bezpečné uvažovat přibližně 512 B. S šestikilobajtovým datovým tokem je tedy maximální přípustná latence, kdy se nebudou ztrácet data,  $512 \text{ B} / 6144 \text{ B/s} = 0,083 \text{ s} = \mathbf{83 \text{ ms}}$ .

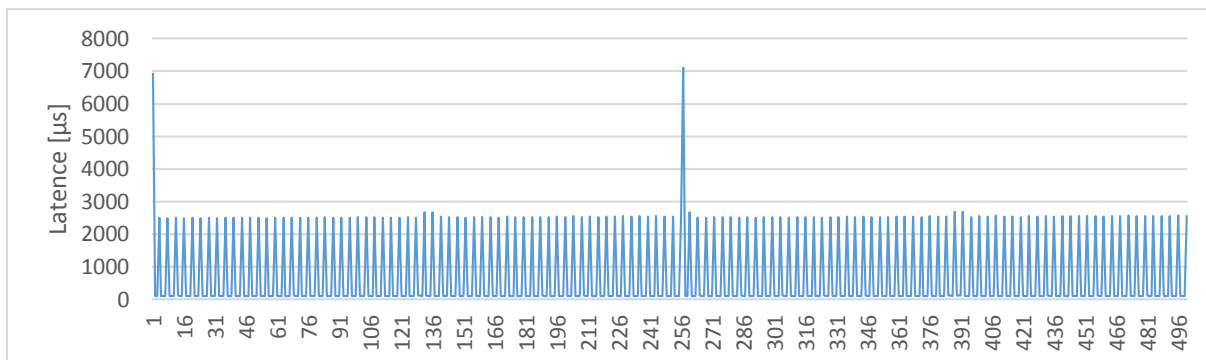
##### *SD karty Sandisk Extreme a LTLM*

SD karty jsou levné, dostupné a vyráběné v řadě rychlostních i kapacitních variant. Významnou výhodou je, že jako komunikační sběrnici využívají SPI, kterou čip ATmega328 nativně nabízí. Díky tomu je k připojení SD karty potřeba pouze pasivního adaptéru na její pouzdro (ať už v plné nebo micro velikosti). Knihovna SdFat<sup>17</sup> je poté schopná pracovat s SD i SDHC kartami až do kapacity 32 GB (nad tuto kapacitu je používán zatím nepodporovaný standard SDXC), přičemž vyžaduje systém souborů FAT16 nebo FAT32.

Rychlost SD karet se běžně udává ve třídách indikujících minimální garantované rychlosti zápisu. Ta je však použitelná pro výkonné zařízení s dostatkem operační paměti, které mohou využít ukládání dat ve velkých dávkách. Pro mikrokontroléry s jednotkami kB RAM je však mnohem zásadnější latence malých zápisů, protože na rozdíl od „velkých“ počítačů nemají možnost vyhlazovat datový tok velkými buffery.

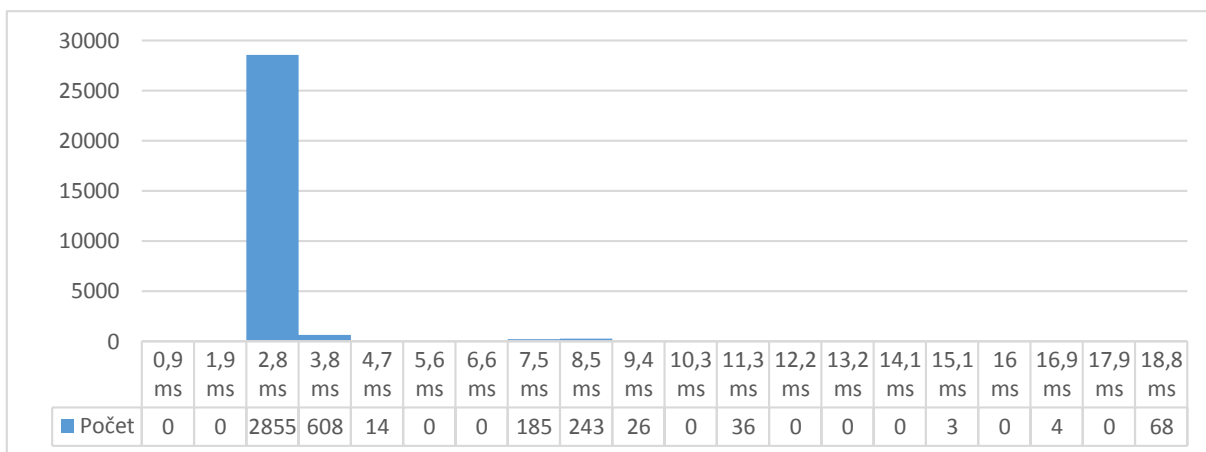
<sup>17</sup> Sdfatlib. Github [online]. 2015 [cit. 2015-05-13]. Dostupné z: <https://github.com/jbeynon/sdfatlib>

Z tohoto důvodu jsem provedl měření latence zápisu 100B „packetu“ na dvou kartách odlišných výkonnostních tříd: levné Class 10 LTLM 32 GB a téměř dvakrát dražší Class 10 + UHS-I Sandisk Extreme 32 GB. Benchmark běžel přímo na hardwaru Arduino a sestával ze zápisu 5MB dat po 100B dávkách, v každém běhu tedy 50 000 měření. Pro každou kartu byly provedeny tři běhy, mezi kterými karta nebyla formátována. Filesystém byl použit FAT32.

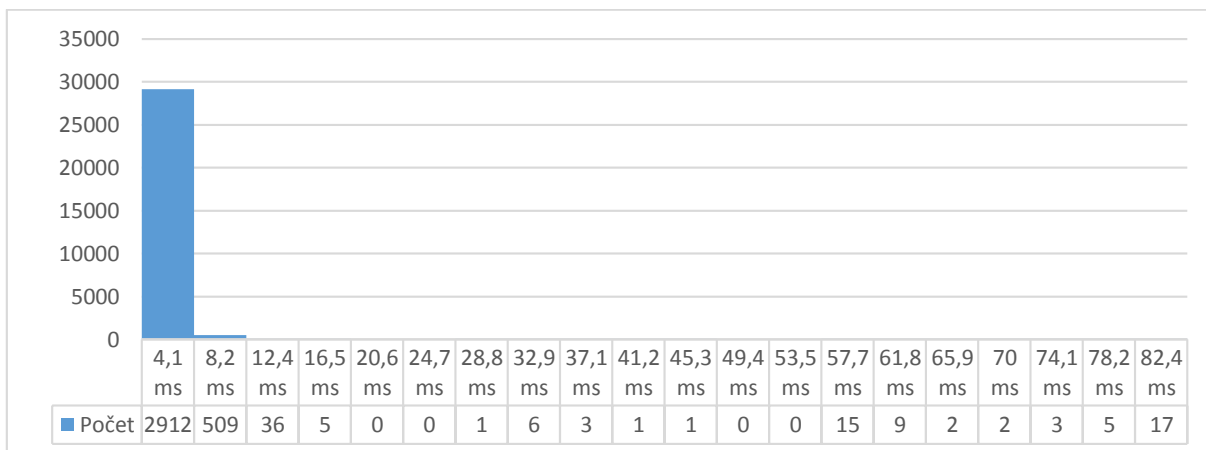


Obrázek 7 – Latence sekvenčního zápisu 128B paketů na SD kartu Sandisk Extreme. Zobrazeno prvních 500 zápisů. Je použita logaritmická škála.

V naměřených datech je vidět, že vždy 3 zápisy proběhnou extrémně rychle (s latencí 90 µs) a následující čtvrtý zápis trvá řádově déle (na zobrazeném výběru 2 500 – 7 000 µs). Z toho lze soudit, že knihovna SdFat používá interní 512B buffer, který na kartu pošle vždy po jeho zaplnění. Pro analýzu latencí tedy byly u obou karet vyfiltrovány zápisy s latencí pod 150 µs.



Obrázek 8 – Rozložení hodnot latence karty LTLM.



Obrázek 9 – Rozložení hodnot latence karty Sandisk Extreme.



Nejvyšší naměřená latence pro kartu LTLM byla 18 816  $\mu$ s, pro *teoreticky* rychlejší kartu Sandisk pak 82 368  $\mu$ s. Nejmenší dosažený datový tok tak byl v případě LTLM 512 B / 0,019 s = 26 kBps, Sandisk 6,1 kBps.

Pro ověření platnosti teze o bufferovaném zápisu každých 512 B bylo další měření nastaveno jako zápis 20 MB dat po 512B paketech. Bylo však nutné použít jednodušší verzi benchmarku, která vypisovala pouze nejvyšší, nejnižší a průměrnou latenci. Logiku, která by vypisovala hodnoty jednotlivých měření, nebylo možné implementovat tak, aby se zároveň vešla do zbývajících RAM a neovlivňovala samotné měření.

Při použití 512B paketu byly latence karty LTLM 28 000  $\mu$ s, 29 344  $\mu$ s a 31 040  $\mu$ s, z čehož vychází nejnižší propustnost 16 kBps. Nižší výkon je možné vysvětlit „těsnější“ zápisovou smyčkou, kdy měla karta méně času na management bloků mezi jednotlivými zápisy. U karty Sandisk bylo zpoždění oproti 128B relativně menší; nejvyšší naměřené latence byly 76 648, 81 824 a 103 120  $\mu$ s, vedoucí k nejmenší propustnosti 4,85 kBps.

Při požadavku na uložení 100 % dat tedy lze určit maximální datový tok pro kartu LTLM nad 10 kBps, čili zápis všech tří os 16bitového akcelerometru včetně několika bajtů metadat při snímkovací frekvenci 1000 Hz. Karta Sandisk pravděpodobně spolehlivě uloží jednu osu zrychlení a bajt metadat.

Tyto hodnoty jsou však nejčernější scénář při požadavku zachování 100 % naměřených dat. Jak je vidět v histogramech na Obrázek 8 – Rozložení hodnot latence karty LTLM a Obrázek 9 – Rozložení hodnot latence karty Sandisk Extreme, drtivá většina zápisů proběhla mnohem rychleji než zmiňovaná maxima. Na úrovni 99,7 percentilu již dosahuje latence LTLM 11 080  $\mu$ s (46 kBps), Sandisk 8 756  $\mu$ s (55 kBps). Nejběžnějších latencí kolem 3 000  $\mu$ s a datového toku 150 kBps obě karty dosahují v 97,75 % měřeních. Pokud tedy aplikace nevyžaduje naprostou spolehlivost, poskytují SD karty poměrně široké přenosové pásmo.

*Při prvních benchmarcích karty LTLM dosahovala maximální latence stabilně hodnot 250 ms a i v prototypu měřícího programu docházelo z důvodu plného bufferu k častému zahazování dat. Proto byla koupena druhá – rychlejší karta Sandisk. Bohužel při těchto měřeních prováděných přibližně o měsíc později se již tento „výkon“ nepodařilo replikovat. Na vině mohl být vadný sektor, který karta v průběhu času vyřadila z používání.*

Všechna naměřená data jsou v souboru [sd-bench.xlsx](#). Zdrojový kód použitý k měření 512B paketu je v [sd-bench-simple.ino](#), měření s reportingem latencí jednotlivých zápisů [sources/sd-bench-individual.ino](#). Oba vycházejí z benchmarku zveřejněném v knihovně SdFat.

#### ESP 8266 WiFi transceiver

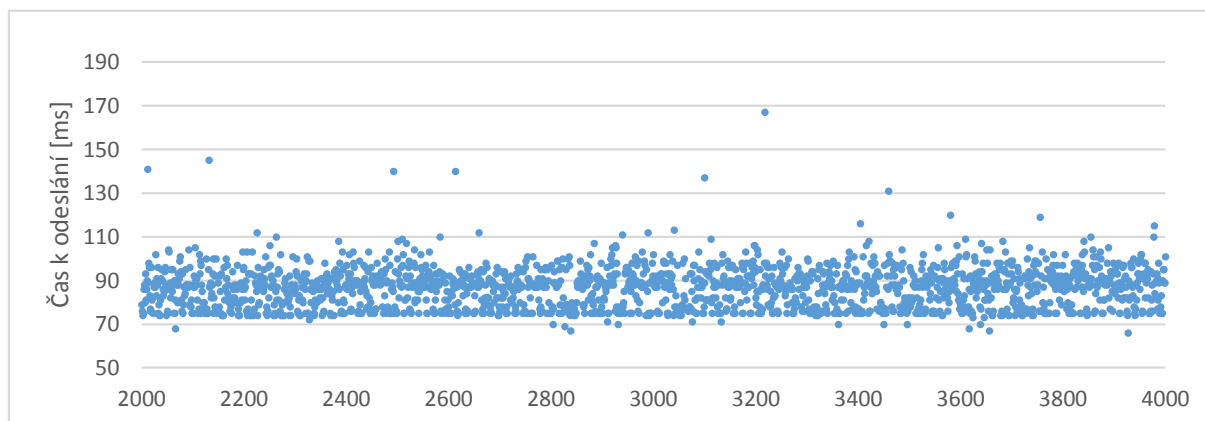
S příchodem trendu Internet of Things se začala vyrábět řada WiFi modulů s integrovaným TCP/IP stackem, komunikujících vysokoúrovňovými příkazy po UART sběrnici. Jedním z nich je čip ESP 8266, který podporuje standardy 802.11b/g/n, včetně šifrování WEP, WPA a WPA2. Na TCP protokolu se umí chovat jako klient i server. V obou režimech umožňuje držet až 4 otevřená spojení. Maloobchodní cena čipu je podle verze \$3 – \$10.

Modul je napájen 3,3 V. Digitální I/O piny používají stejnou logickou úroveň a jsou udávány jako 5 V tolerantní. Udávaná špičková spotřeba činí 215 mA – vyžaduje tedy externí napájení (limit 3,3V pinu na deskách Arduino je 50 mA).

Data se z modulu odesílají po uvození příkazem AT+CIPSEND=<id spojení>,<množství dat v B>. Po potvrzení znakem > modul čeká na příjem zadaného množství bajtů. Jejich úspěšné odeslání potvrzuje zprávou SEND OK. Teprve poté je připraven na další cyklus uvozený AT+CIPSEND. Případný retransmit zařizuje modul interně, SEND OK poté přijde později.

Měření latence probíhalo v bezdrátové síti 802.11n šifrované WPA2, poskytované access pointem Mikrotik RB951G. Podmínky příjmu signálu byly průměrné. AP se nacházelo dvě místnosti od ESP 8266,

udávaná síla signálu byla okolo -66 dBm. Každé měření sestávalo z odeslání datového paketu o velikosti 512 B na TCP server umístěný v pevné lokální gigabitové síti (jediným směrovačem na cestě byl Mikrotik).



Obrázek 10 – Čas od zaslání požadavku k odeslání (AT+CIPSEND) po obdržení potvrzení (SEND OK). Je zobrazen výběr 2 000 po sobě jdoucích měření.

	100%	99,99%	99,9%	99%	95%	90%	50%	COUNT
TIME	714	423	202	116	102	98	87	8807
SPEED	0,7	1,2	2,5	4,4	5,0	5,2	5,9	

Tabulka 5 – Percentilové rozložení doby odeslání 512B dat (hodnoty v ms) a odpovídajícího datového toku (kBps)

Na měření je vidět řada výrazných extrémů způsobených retransmity – největší z nich 714 ms. Při požadavku na 100 % spolehlivost tak lze očekávat datový tok pouhých 0,7 kB. Průměrná doba odeslání byla 87 ms, průměrný datový tok 5,9 kBps.

Vzhledem k požadovanému minimálnímu datovému toku 6 kBps tedy není tento WiFi modul pro tuto aplikaci možné použít. Nicméně pro jiné aplikace s nižším požadovaným datovým tokem a/nebo požadavkem na realtime stream dat ze senzoru se jedná o zajímavou volbu zejména díky snadnosti použití, 0% paket lossu (za cenu retransmitů) a nízké ceně.

Naměřené hodnoty jsou v souboru [data/wifi-time-to-send.xlsx](#). Zdrojové kódy ovládacího skriptu WiFi modulu jsou v [sources/wifi-sender.js](#), implementace TCP serveru [sources/wifi-server.js](#).

#### NRF24L01 ISM 2,4 GHz radio transceiver

V nelicencovaném ISM pásmu 2,4 GHz pracují kromě WiFi i méně výkonné RF transceivery, typicky používané například k bezdrátovým klávesnicím. Jedním z nejpopulárnějších modulů pro Arduino je Nordic Semiconductor NRF24L01<sup>18</sup>, 2,4 GHz rádio s teoretickou šířkou pásma až 2 Mbps. Komunikační rozhraní je sběrnice SPI.

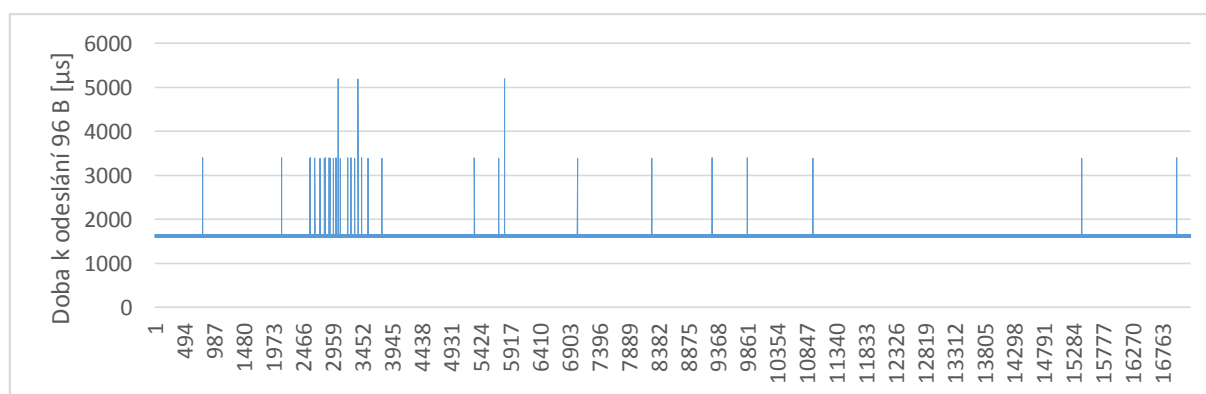
Modul se vyrábí v několika variantách, které se liší zejména anténou. Nejlevnější ji mají integrovanou, vyleptanou na tištěném spoji. Dražší verze nabízí SMA nebo IPX konektory na externí antény. Ceny se pohybují od cca \$1 po \$5.

<sup>18</sup> NORDIC SEMICONDUCTOR. 2008. NRF24L01+ Single Chip 2,4GHz Transceiver: Preliminary Product Specification. V1.0. 75 s. [cit. 2015-05-14]. Dostupné také z: <https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus Preliminary Product Specification v1.0.pdf>

Softwarové ovládání modulů zajišťuje knihovna tmrh20 NRF24L01<sup>19</sup>. Tato umí provozovat vysílač v režimu unicast, multicast nebo broadcast, volitelně se zajištěním spolehlivosti spojení (na softwarové úrovni). Software pracuje se třemi 32B buffery, data odesílá po jejich naplnění.

Při předběžném testování vykazovaly moduly značný packet loss (přes 50 %), jehož příčinou se ukázaly být proudové špičky, které nedokázal vykrýt zdroj na desce Arduino. Z toho důvodu byl ke zdroji napětí připojen 4,7μF kondenzátor, který snížil ztrátovost na cca 6 % a konečně 47μF kondenzátor, který packet loss na krátké vzdálenosti prakticky eliminoval.

Zkouška propustnosti probíhala při vzdálenosti přijímačů 5 m se zachováním přímé viditelnosti (uvažovaný scénář je připevnění obou jednotek na karoserii automobilu). Jeden datový paket byl 92 B (3 × 32 B). V softwaru bylo zapnuto zajištění spolehlivosti spojení opakovanými retransmity. Použitá varianta modulů byla vybavena pouze integrovanou anténou.



Obrázek 11 – Doba k odeslání 96 B paketu po 2,4 GHz rádiu NRF24L01.

	TOTAL	1588 – 1656	3388 – 3400	5196
SPEED		58	28	18,5
COUNT	17237	17203	30	4
PERCENT	100,00%	99,80%	0,17%	0,02%

Tabulka 6 – Procentuální zastoupení skupin naměřených dob odeslání.

Paket byl ve většině případů (99,8 %) odeslán napoprvé za 1 656 μs, představující datový tok 58 kbps. V 0,17 % případech došlo k jednomu retransmitu (3 400 μs, 28 kbps) a konečně v 0,02 % ke dvěma retransmitům trvajícím 5 196 μs a představujícím nejnižší dosažený datový tok 18,5 kbps.

Z uvedeného plyne, že při podmínce přímé viditelnosti mezi jednotkami je možné tyto použít k přenosu dat ze všech os akcelerometru v reálném čase. Jako přijímač lze uvažovat další desku Arduino, ze které by PC četlo přes UART-USB převodník. Vzhledem k použité SPI sběrnici se také nabízí integrace do systému CompactRIO, který by pak sloužil jako jedno úložiště pro všechna data.

V případě složitějších podmínek pro příjem signálu je také možné zvážit použití varianty s externí SMA nebo IPX anténou.

Surová naměřená data jsou v souboru [data/radio-time-to-send.xlsx](#). Implementace odesílacího programu je v [sources/radio-data-sender.ino](#), přijímací program [sources/radio-data-receiver.ino](#).

#### 4.1.6. Snímkovací frekvence digitálních a analogových pinů

Digitální I/O operace na čipu ATmega328 trvají jeden instrukční cyklus, tedy 1/16 μs. Nicméně abstrakce digitalWrite() v prostředí Arduino zajišťující např. přenositelnost zdrojového kódu na jiné mikrokontroléry trvá řádově déle.

<sup>19</sup> Optimized High Speed NRF24L01+ Driver Class Documentation. 2015. C79CCC7. [cit. 2015-05-14]. Dostupné také z: <https://tmrh20.github.io/RF24/index.html>

Jak bylo naměřeno v kapitole 2.

SPECIFIKACE GPS modulu, části 2.4. PŘESNOST A POSUN DVOU PPS SIGNÁLŮ, jeden cyklus detekce logické úrovně trvá 16 – 24  $\mu$ s. Ve stejném experimentu se ukázalo, že i při přivedení stejného signálu na dva piny obsluhované stejným registrem (piny jsou rozděleny do dvou skupin: 0 – 7 a 8 – 13) dojde v přibližně 30 % případů k jednocyklovému zpoždění detekce hrany na jednom pinu (v provedeném měření byl vždy pin 3 za pinem 2, nikdy naopak).

Na základě těchto měření lze usuzovat přesnost detekce logické úrovně na digitálních pinech -0 – +24  $\mu$ s.

Rychlost snímání analogových pinů je omezena pracovní frekvencí ADC. Ta je softwarově nastavitelná pomocí frekvenční děličky, jejíž standardní hodnota je 128, což vede na frekvenci 125 kHz (16 MHz / 128). Jedna konverze trvá 13 pracovních cyklů, z čehož vyplývá standardní snímkový frekvence 9 600 Hz, a tedy chyba měření -0 – +104  $\mu$ s. Tuto chybu lze snížit použitím vyšší pracovní frekvence, avšak pouze za cenu sníženého rozlišení měřeného napětí.

#### 4.1.7. Možnosti synchronizace

Čip ATmega328 nabízí 14 digitálních a 8 analogových (ADC) pinů. Na některých z nich pak jsou připojeny ještě sběrnice UART, I<sup>2</sup>C a SPI. Je možné volit z celé řady vysílačů synchronizačních pulsů – od primitivní fotodiody, až po zpracování broadcastovaného UDP paketu. V této části představím vybrané metody, změřím jejich přesnost a popíši výhody a nevýhody.

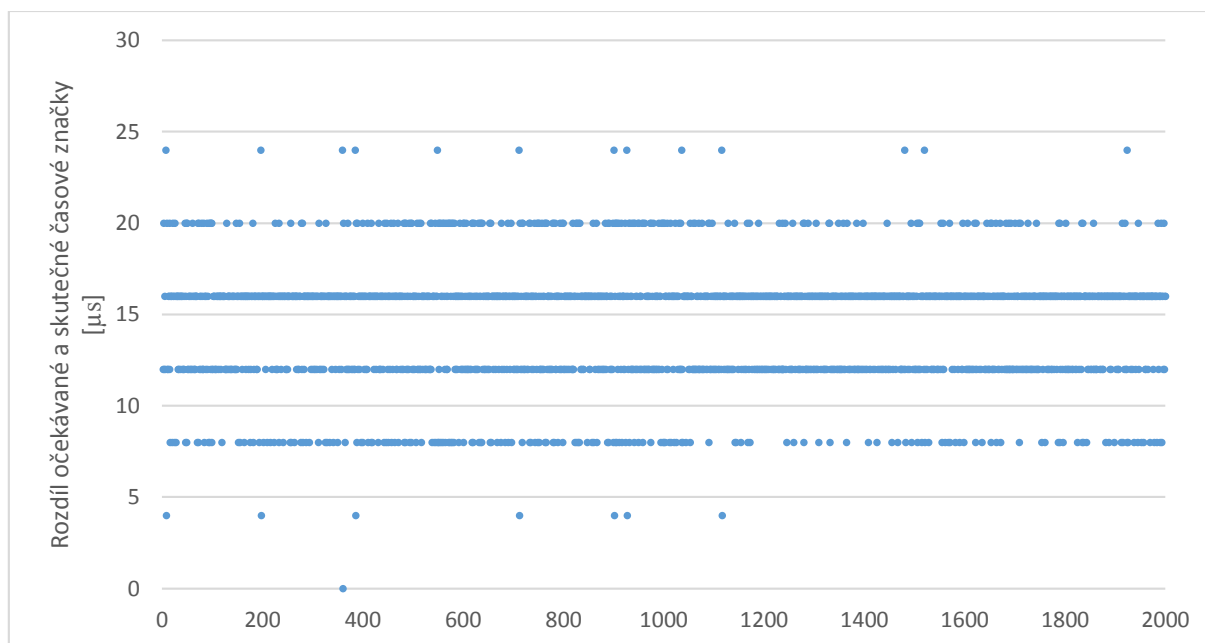
##### *Vnitřní hodiny*

Ještě než se dostaneme k jednotlivým způsobům synchronizace, je třeba změřit přesnost interních hodin. Čip ATmega328 je vybaven počítadlem hodin reálného času. Jeho výstup je dostupný přes funkce `millis()`, vracející milisekundy od zapnutí desky a zejména `micros()`, vracející mikrosekundy. Jeho přesnost je závislá na řídicím oscilátoru, integrovaném na desce.

Právě přesnost těchto hodin určuje potřebný interval mezi jednotlivými synchronizačními pulsy. Vzhledem k nepraktičnosti vysílání synchronizačních pulsů v průběhu samotného měření považuji za optimální takové řešení, které dosáhne požadované přesnosti měření pouze se dvěma synchronizačními pulsy – jedním na začátku a druhým na konci měření.

Jako časový etalon v tomto měření sloužily vteřinové PPS pulsy z GPS u-blox MAX-7C. Software na desce Arduino kontroloval, zda na vstupním pinu došlo k přepnutí z logické 0 na logickou 1 (vzestupná hrana) a v tomto cyklu uložil aktuální mikrosekundový čas `micros()`, který následně poslal po UART lince do řídicího PC. Ideální výstup by tedy měla být řada hodnot zvyšující se vždy o 1 000 000 ( $\mu$ s, tedy 1 s). Měření probíhalo po dobu téměř 37 minut (2200 sekund) a za stálé teploty.

Po naměření hodnot jsem analýzu začal výpočtem každovteřinové odchylky. Pokud tedy byl výstup 904 748, následující by teoreticky měl být 1 904 748. Rozdíl mezi teoretickou a reálnou hodnotou každého měření ukazuje následující graf.



Obrázek 12 – Rozdíl očekávané a skutečné mikrosekundové časové značky

	TOTAL	0	4	8	12	16	20	24
COUNT	2199	1	7	206	595	1164	211	15
PERCENT	100,0	0,0	0,3	9,4	27,1	52,9	9,6	0,7
	%	%	%	%	%	%	%	%

Tabulka 7 – Procentuální rozložení naměřených hodnot.

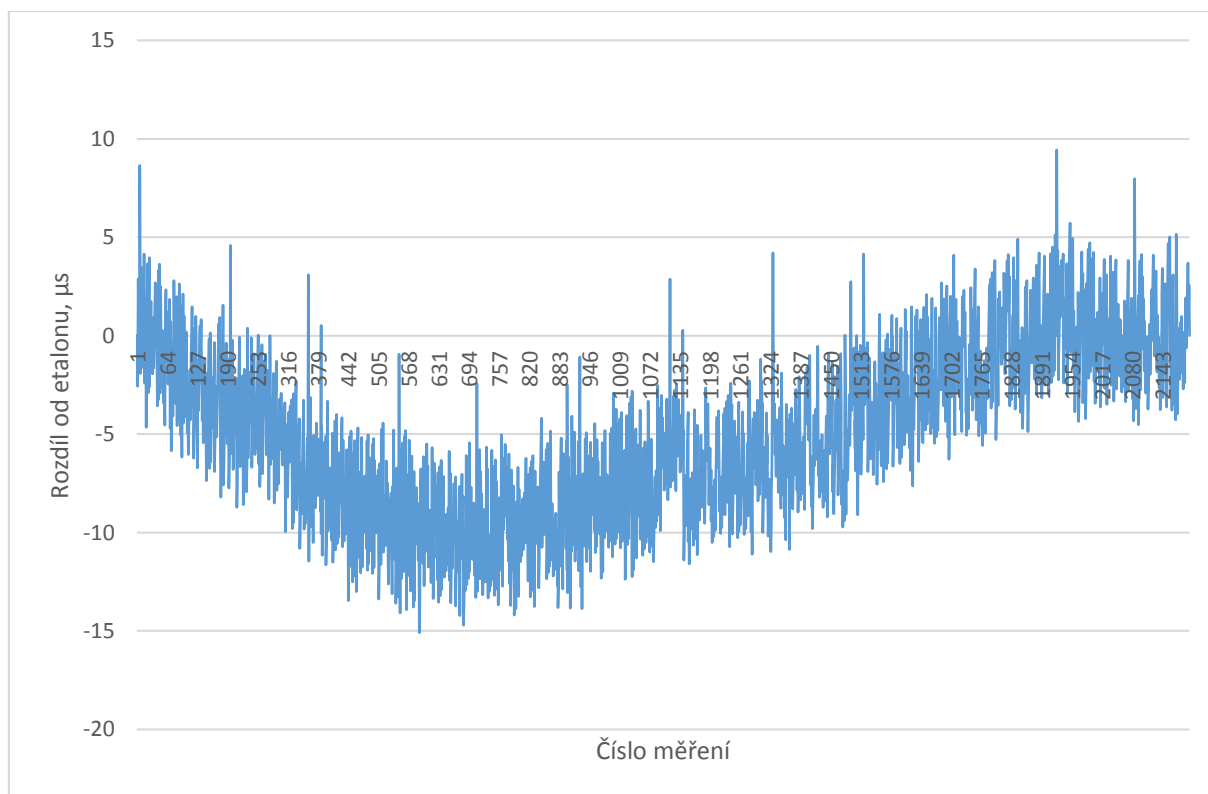
Z grafu je vidět a analýza to potvrzuje, že mezivteřinová chyba variuje nejčastěji mezi 16  $\mu$ s a 12  $\mu$ s, s občasným přeskočením na 8  $\mu$ s a 20  $\mu$ s. Poměrně symetrické rozložení dává naději na dostatečnou přesnost i při zmíněných dvou synchronizačních pulsech.

*Pozn.: přestože `micros()` vrací čas v mikrosekundách, rozlišení (nejmenší měřitelná jednotka) interních hodin je 4  $\mu$ s, proto jsou hodnoty diskrétní a násobky 4.*

Z naměřených dat není problém nasimulovat dva synchronizační pulsy – na začátku a na konci měření. V každém měření mějme k dispozici *skutečný* timestamp, jehož zdrojem jsou interní hodiny. Dále mějme k dispozici dva timestamps v prvním a posledním bodě měření, jejichž zdrojem je synchronizační senzor (v tomto případě PPS pin GPS připojený na přerušení generující pin) – tyto budu označovat *očekávané*. Pro zjednodušení v tomto případě nebudu počítat s absolutním časem, ale jako nulu obou timestampů budu považovat první měření.

Vzhledem k omezení na pouze dva synchronizační pulsy lze vypočítat *drift* vnitřních hodin – tedy poměr, o kolik jsou rychlejší než etalon. Ten v našem případě počítání času od nuly získáme prostým vydělením posledního *skutečného* timestampu *očekávaným*. Korekci naměřených hodnot provedeme vydělením *očekávaného* timestampu *driftem* a dostaneme *korigovaný* timestamp.

Vypočítané *korigované* timestamps jsem porovnal s *očekávanými* (protože byla měření generována PPS GPS, byly vždy inkrementované o 1 000 000) a dostal následující průběh odchylky *korigovaného* času od etalonu:



Obrázek 13 – odchylka korigovaného času od etalonu v průběhu půlhodinového měření. K dispozici byly pouze dva synchronizační pulsy, na začátku a konci měření.

Nejvyšší kladná odchylka dosahuje 9,44  $\mu\text{s}$ , nejvyšší záporná -15,06  $\mu\text{s}$ . Rozdíl extrémů tedy činí 24,52  $\mu\text{s}$ . I s velkým bezpečnostním faktorem tedy lze očekávat, že v průběhu půlhodinového měření lze s dvěma synchronizačními pulsy dosáhnout přesnosti  $\pm 50 \mu\text{s}$ .

Vzhledem k relativně nízkému driftu interních hodin – přibližně 16 ppm (parts per million) – budu ve všech následujících měřeních používat hodnoty interních hodin. Ve většině měření by korekce driftu nebyla větší, než chyba způsobená 4  $\mu\text{s}$  rozlišením funkce `micros()`.

Všechna naměřená data jsou v souboru [data/internal-clock-accuracy.xlsx](#). Použitý zdrojový kód je v [sources/internal-clock-timer.ino](#).

#### GPS PPS signál

Aplikačně nejjednodušší metoda spočívající v detekci hrany na jednom z digitálních vstupů (s výhodou lze použít piny 2 a 3, které lze nastavit jako zdroje přerušení). Přesnosti této metody se podrobně věnuji v kapitole

#### SPECIFIKACE GPS modulu.

Informaci o absolutním čase – tedy v jaký UTC čas PPS hrana vlastně nastala – je možné získat z NMEA zpráv ZDA, GGA, RMC, GLL a dalších <sup>1</sup>. Výběrem nejúspornější zprávy lze ušetřit na datovém toku použité komunikační sběrnice i procesorovém čase mikrokontroleru při zpracování dat.

Sběrnici lze u většiny modulů volit ze všech sběrnic implementovaných ATmega328 – UART, I<sup>2</sup>C i SPI. Díky malému datovému toku lze navíc použít i softwarové (*bit banded*) emulace na DIO pinech.

Nevýhodou použití GPS je zejména vysoká cena přijímačů (v porovnání s ostatními zde diskutovanými řešeními). Je také třeba počítat s vhodným umístěním antény.

#### *LED a fotodiody*

Elektronicky velmi jednoduchá metoda sestávající z vysílače synchronizačních pulsů v podobě LED diody a přijímací fotodiody citlivé na stejnou vlnovou délku. Výhodami jsou zejména nízká cena, rozměry, energetické nároky, bezdrátovost a snadno pochopitelné řízení adresace – tedy unicast, multicast či broadcast – pomocí směrování LED a stínění fotodiod.

Na druhou stranu se jedná o systém citlivý na znečištění, při umístění na spodní stranu automobilu a při deštivém počasí může být méně spolehlivý. Není možné softwarově měnit adresaci jednotlivých přijímačů. Vyžaduje přímou viditelnost, v případě měření odrazem se snižuje odstup signálu a šumu. Další nevýhodou je zarušenost přenosového spektra, kdy pohybující se části automobilu mohou svým stíněním okolního světla generovat falešný signál.

Měření přesnosti probíhalo v systému dvou desek Arduino, kdy jedna sloužila jako vysílací a druhá jako přijímací. Vzhledem k proudovému omezení digitálních I/O pinů na 40 mA a požadavku případné škálovatelnosti byla vysílací dioda napájena přes transistorový obvod. Přijímací fotodiody byla připojena k analogovému pinu na druhém Arduino.

Měřenou veličinou byla velikost a hlavně rozptyl zpoždění od vyslání signálu (nastavení logické 1 na výstupním pinu vysílače) po detekci stoupající hrany na přijímači. Z tohoto důvodu byl výstupní signálový pin přiveden také na jeden vstupní pin přijímače. Zpoždění se pak měřilo jako rozdíl detekce hrany na digitálním (referenčním) a analogovém vstupu.

Použitá LED dioda byla infračervená Kingbright L-53F3C, 20mW, 940nm, rise time není výrobcem udáváný <sup>20</sup>. Tranzistor ST 2N9304 s udávaným delay time 35 ns, rise time 35 ns <sup>21</sup>. Fotodiody Vishay BP104, citlivá mezi 780 nm – 1 100 nm, udávaný rise time 100 ns, delay time není specifikován <sup>22</sup>.

---

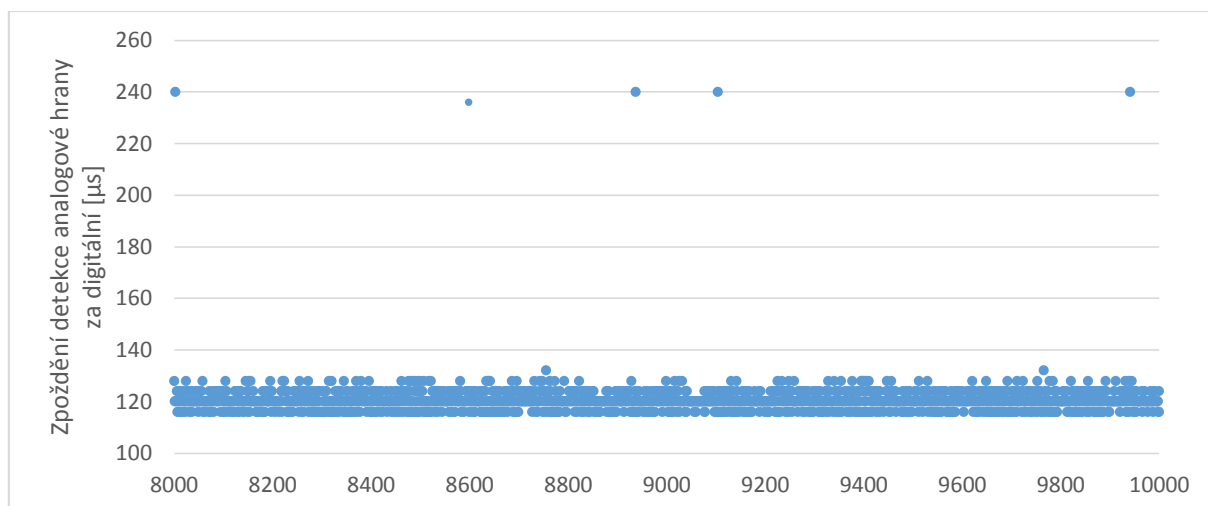
<sup>20</sup> KINGBRIGHT ELECTRONIC CO., LTD. 2001. L-53 Series Datasheet [online]. V.1. 5 s. [cit. 2015-05-14].

Dostupné také z: <http://www.ges.cz/sheets/l/l53f.pdf>

<sup>21</sup> STMICROELECTRONICS. 2003. 2N3904 Small Signal NPN Transistor [online]. 6 s. [cit. 2015-05-14]. Dostupné také z: <http://pdf.datasheetcatalog.com/datasheet/stmicroelectronics/8850.pdf>

<sup>22</sup> VISHAY SEMICONDUCTORS. 2011. BP104: Silicon PIN Photodiode [online]. Rev. 1.6. 5 s. [cit. 2015-05-14]. Dostupné také z: <http://www.vishay.com/docs/81500/81500.pdf>





Obrázek 14 – Průběh zpoždění detekce analogové vzestupné hrany za digitální vzestupnou hranou. Zobrazeno je vybraných 2 000 po sobě následujících měření. Graf ke podsekutý (začíná na 110  $\mu$ s).

	TOTAL	116	120	124	128	132	136	236	240
COUNT	12286	2280	6857	2623	485	30	2	2	7
PERCENT	100,00%	18,56%	55,81%	21,35%	3,95%	0,24%	0,02%	0,02%	0,06%

Tabulka 8 – Procentuální rozložení naměřených hodnot.

Z měření je patrné úzké hrdlo systému: ADC převodník o pracovní frekvenci 9 600 Hz, tedy délkou cyklu 104  $\mu$ s. Vzhledem ke zpožděním systému se tedy analogová vzestupná hrana nestihla zaznamenat ve stejném cyklu jako digitální, což vedlo ke zpoždění 116 – 136  $\mu$ s (99,93 % měření). Výjimečně (0,07 %) se však vzestupná hrana nezaregistrovala ani v druhém cyklu, což vedlo ke zpoždění 236 – 240  $\mu$ s.

Přesnost synchronizace pomocí fotodiody tedy lze předpokládat jako dvojnásobek pracovního cyklu ADC převodníku, tj.  $\pm 104$   $\mu$ s. V případě požadavku na vyšší přesnost je pak možné pracovat s frekvenční děličkou řídící pracovní cyklus ADC za cenu snížení rozlišení měření.

Naměřená data jsou v souboru [data/led-delays.xlsx](#). Zdrojový kód generátoru signálu je v [sources/analog-source.ino](#), měření latence na přijímači je v [sources/analog-latency.ino](#).

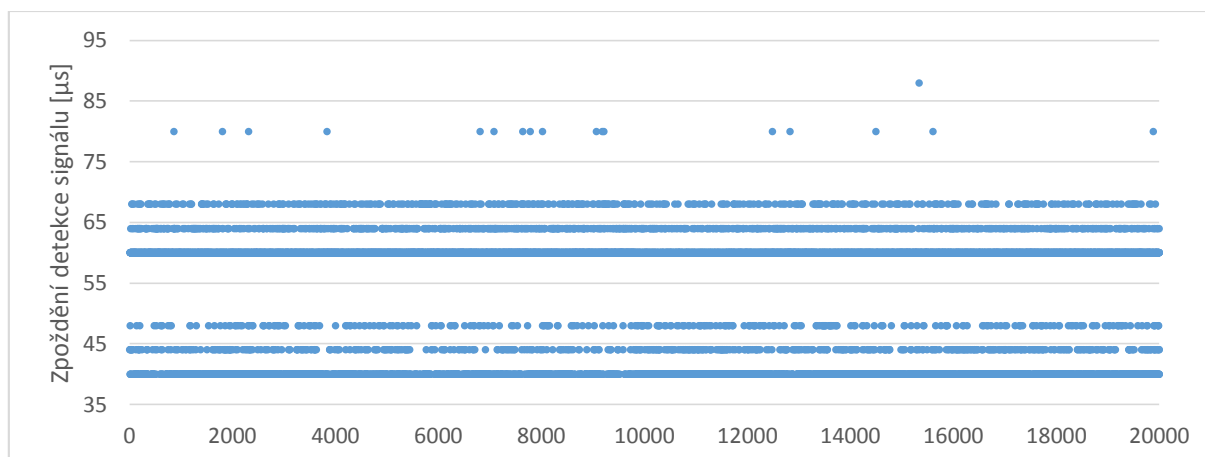
#### 433MHz rádiová spoušť

Levná rádiová spoušť s označením FS1000a (datasheet bohužel není k dispozici) pracuje v ISM pásmu 433 MHz a využívá amplitudovou modulaci. Veškerý interface sestává z jednoho datového pinu na vysílači a druhého na přijímači. Vysílač je možné napájet 5 – 12 V, přičemž v celém rozsahu využívá vysílací proud 14,5 mA. Vyšší napětí tedy znamená vyšší výkon i dosah. Digitální vstup vysílače pracuje s 3,3 V logickou úrovní a je 5 V tolerantní. Adresace přijímačů je výhradně broadcastová.

Přijímač vyžaduje striktně 5V napájení, digitální výstup používá logickou úroveň 3,3 V a je opačný než vstup (tedy při poslání logické 1 bude na výstupu log. 0). Výstup postrádá jakékoliv zpracování a při přechodu mezi stavy dochází k období neurčitosti, kdy se na výstupu rychle střídá logická 0 a 1.

Výhodou řešení je zejména jednoduché zapojení – nabízí se připojit data pin vysílače přímo na PPS pin GPS, čímž velmi levně „rozvedeme“ přesný synchronizační puls na všechny senzory. Nevýhodami pak jsou nutnost softwarově ošetřit přechodové stavy a jednosměrnost komunikace, kdy není možné implementovat in-band potvrzení o příjmu signálu.

Měření probíhalo podobně jako měření fotodiody. Měřeným parametrem je zpoždění detekce sestupné hrany signálu za vzestupnou hranou referenčního pinu. Generovaný signál byl pulsně modulovaný s délkou cyklu 100 ms a 50 % střidou. Měření probíhalo přibližně půl hodiny.



Obrázek 15 – Průběh zpoždění sestupné hrany přenášeného signálu za vzestupnou hranou referenčního signálu. Graf je podseknutý (začíná na 40 μs) pro větší názornost.

	TOTAL	40	44	48	60	64	68	80	88
COUNT	20596	3875	578	305	14458	931	431	17	1
PERCENT	100,0%	18,8%	2,8%	1,5%	70,2%	4,5%	2,1%	0,1%	0,0%
		23,1%			76,8%			0,1%	

Tabulka 9 – Procentuální zastoupení jednotlivých hodnot a skupin.

Díky použití rychlejšího digitálního interface zde došlo k podstatně menšímu rozptylu než v případě fotodiody měřené přes ADC. Nejnížší hodnota zpoždění byla 40 μs, nejvyšší 88 μs – celkový rozdíl tedy činí 48 μs.

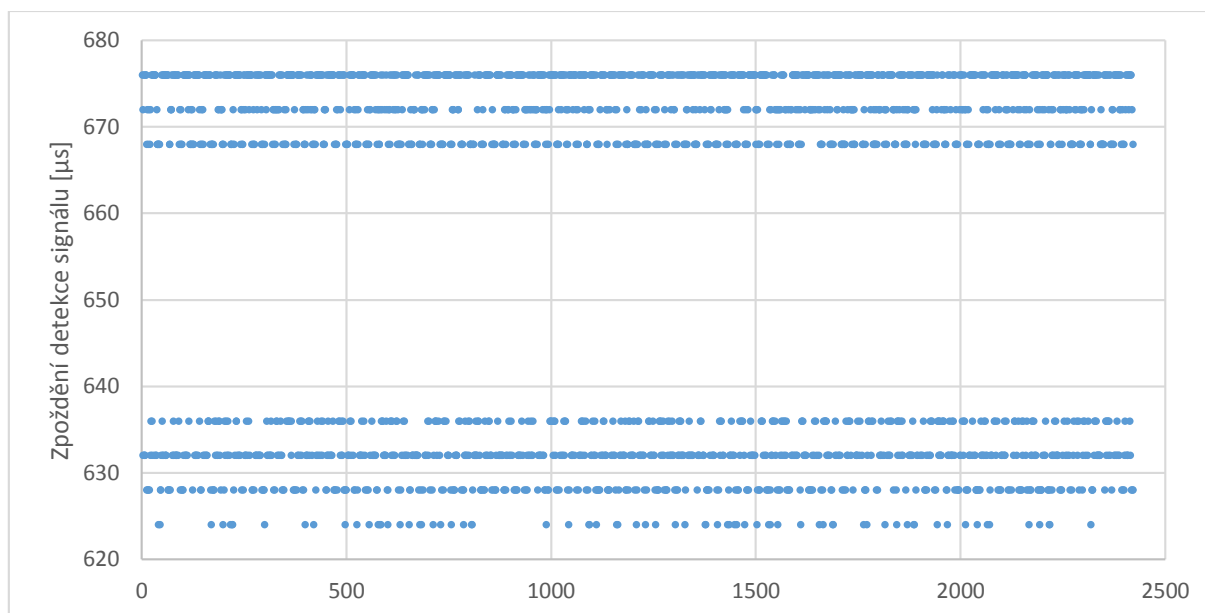
Všechna naměřená data jsou v [data/rf433-delays.xlsx](#), zdrojový kód vysílače [sources/rf433-tx.ino](#), přijímače [sources/rf433-rx.ino](#).

#### NRF24L01 ISM 2,4 GHz radio transceiver

Nordic Semiconductor NRF24L01 popsany v předchozí sekci (NRF24L01 ISM 2,4 GHz RADIO TRANSCEIVER) je možné díky jeho duplexnímu návrhu použít i jako synchronizační prvek. Zejména v případě, kdy by uložení dat bylo řešeno právě 2,4GHz datalinkem, by takové řešení představovalo výrazné zjednodušení architektury, neboť by se ušetřila jedna periferie – a s ní komplexita zapojení (event. návrhu tištěného spoje), softwaru a napájení.

Další výhodou NRF24L01 je možnost volby mezi unicastem (s potvrzením příjmu přímo v ovladačích), multicastem a broadcastem (s potvrzením příjmu implementovatelným v ovládacím softwaru). Nejvýznamnější nevýhoda se projeví, pokud bude zvoleno ukládání dat na SD kartu, neboť v tomto případě bude muset karta sdílet SPI sběrnici s rádiem. To by vedlo k nežádoucí komplexitě ovládacího software a snížení výkonu obou zařízení.

Měření probíhalo podobně jako v předchozích případech. Před příkazem k odeslání dat byla referenční linka zvednuta na logickou 1. Po provedení příkazu byla ještě 1 ms držena vysoko a poté přepnuta zpět na logickou 0. Měřená veličina je zpoždění prvního úspěšného volání `radio.available()` (tedy kdy je zařízení připraveno zaslat/přijmout data) za hranou referenčního signálu. Odesílal se 2B datový paket každých 100 ms po dobu čtyř minut.



Obrázek 16 – Průběh zpoždění příjmu dat za počátkem jejich vysílání.

	TOTAL	624	628	632	636	668	672	676
COUNT	2421	74	262	472	239	380	285	709
PERCENT	100,0%	3,1%	10,8%	19,5%	9,9%	15,7%	11,8%	29,3%
		43,2%				56,8%		

Tabulka 10 – Procentuální zastoupení jednotlivých hodnot a skupin.

Nejnižší naměřené zpoždění bylo 624  $\mu\text{s}$ , nejvyšší 676  $\mu\text{s}$  – rozdíl extrémů tedy je 52  $\mu\text{s}$ . Z grafu na Obrázek 16 je však vidět, že se hodnoty shlukovaly ve dvou pásmech, a to 624  $\mu\text{s}$  – 636  $\mu\text{s}$  (rozdíl 12  $\mu\text{s}$ ) a 668  $\mu\text{s}$  – 676  $\mu\text{s}$  (9  $\mu\text{s}$ ). Šířka dělícího pásma 32  $\mu\text{s}$  by při vložení časové značky do odesílaného paketu měla dostačovat k zpětnému určení, jaké zpoždění bylo zaznamenáno. Tím by se nejistota snížila na pouhých 12  $\mu\text{s}$ .

Při použití NRF24L01 k přenosu dat je možné takto vysoký determinismus využít ke zjednodušení synchronizační logiky. Pokud by odeslaný paket, stejně jako každý datový bod v něm, obsahoval časovou značku, bylo by možné uvažovat o zpětné synchronizaci naměřených hodnot – tedy systému, kdy by přesný čas byl pouze na přijímači. Z doby příjmu by se poté dopočítaly přesné časy jednotlivých měření. Výhodou je, že přijímač může agregovat data z více vysílačů, přičemž vyžaduje jediný zdroj přesného času.

*Tento scénář vyžaduje, aby NRF24L01 byl na SPI samotný. Pokud by měla být SPI sdílená např. se čtečkou SD karet, má čip NRF24L01 vyvedený IRQ pin, který slouží k indikaci, že byla přijata data. Je možné nastavit synchronizaci i proti tomuto pinu, rozptyl jeho zpoždění je obdobný, jako při přijímání zpracovaných dat. Na druhou stranu se v průběhu měření neaktivoval zdaleka vždy, když byla vysílána data – jeho „packet loss“ činil kolem 40 %.*

Všechna naměřená data jsou v souboru [data/radio-delays.xlsx](#). Zdrojový kód vysílače je v [sources/radio-delays-sender.ino](#), přijímače v [sources/radio-delays-receiver.ino](#).

#### ESP 8266 WiFi transceiver

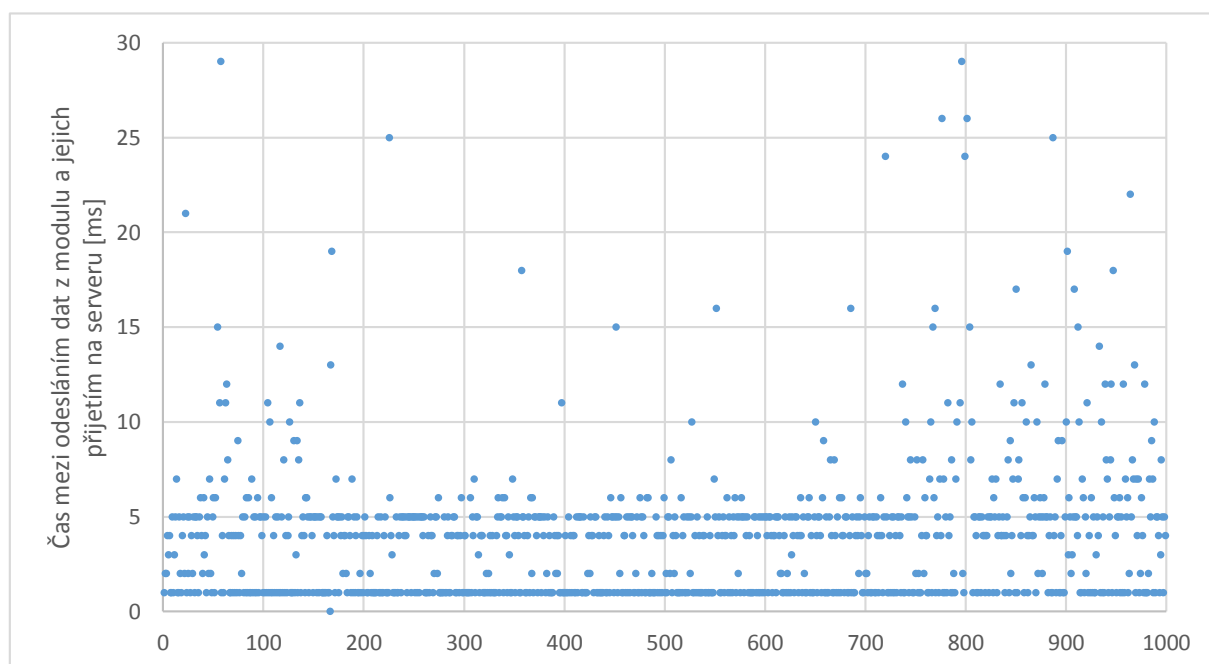
Hardware i rozhraní WiFi modulu ESP 8266 byly popsány v sekci věnující se datovým přenosům (ESP 8266 WiFi TRANSCEIVER). V této části se budu zabývat jen jeho použitím v roli synchronizačního prvku. Vzhledem k tomu, že scénář měření očekává na začátku a konci měření přítomnost serveru

s přesným časem dostupného přes WiFi, je výhodou tohoto modulu, že nevyžaduje k synchronizaci žádný hardware navíc. Dále díky tomu, že komunikuje přes UART neblokuje SPI, které používá SD karta.

Nevýhodami pak jsou zejména jeho vysoká spotřeba vyžadující vlastní měnič na 3,3 V a – pokud nechceme používat další synchronizační hardware uvnitř vozidla – nemožnost spolehlivě přijímat synchronizační pulsy v průběhu měření.

Měření latencí probíhalo stejným způsobem, jako při měření jeho datové propustnosti – tedy za průměrných příjmových podmínek (-66 dBm) a s jediným minimálně vytíženým směrovačem mezi WiFi sítí a serverem na pevné síti.

Vzhledem k schopnosti modulu pracovat nad stavovým protokolem TCP je možné využít několika synchronizačních signálů – sestavení spojení, odeslání dat, přijetí dat a rozpad spojení. Měření tedy sestávalo z 1 001 cyklu těchto operací. Mezi jednotlivé fáze bylo vloženo 100 ms zpoždění, aby zařízení mohlo vyprázdnit případné buffery. Při jednom cyklu se spojení nepovedlo navázat – tento byl z výsledků vyřazen.



Obrázek 17 – Rozložení času mezi odesláním dat z ESP 8266 a jejich přijetím na serveru. Osa Y je kvůli dostatečnému rozlišení oříznuta, je zobrazeno 98,8 % výsledků.

		100%	99,9%	99%	95%	90%	50%	COUNT
Conn	MAX	12024,0	4542,5	1377,6	25,1	21,0	15,0	1000
	MIN	7,0	7,0	8,0	8,0	10,0		
	DIFF	12017,0	4535,5	1369,6	17,1	11,0		
TX	MAX	22272,0	19485,8	67,0	11,0	7,0	4,0	
	MIN	0,0	1,0	1,0	1,0	1,0		
	DIFF	22272,0	19484,8	66,0	10,0	6,0		
RX	MAX	769,0	681,1	121,0	116,0	68,0	22,0	
	MIN	5,0	5,0	5,0	6,0	6,0		
	DIFF	764,0	676,1	116,0	110,0	62,0		
Disconn	MAX	18619,0	8736,9	1010,4	12,0	9,0	6,0	
	MIN	2,0	2,0	2,0	2,0	2,0		
	DIFF	18617,0	8734,9	1008,4	10,0	7,0		

Tabulka 11 – Vybrané percentily doby sestavení spojení (Conn), odeslání dat (TX), příjmu dat (RX) a rozpadu spojení (Disconn). Časy jsou v milisekundách.

Z analýzy naměřených časů vyplynulo, že k nejmenšímu rozptylu zpoždění dochází mezi odesláním dat z modulu a jejich přijetím na serveru - v 82,3 % měření bylo mezi 1 – 5 ms (další percentily: 98,6 % 29 ms, 99% 55 ms). Podobným kandidátem je i rozpad spojení - 80,7 % je v rozmezí 2 – 7 ms (98,6 % 28 ms, 99% 1 010 ms).

Je vidět, že složitější přenosová vrstva vede k podstatně vyššímu rozptylu, který se i při použití opakovaných měření nedostane pod 5 ms. ESP 8266 je tedy použitelná jen pro aplikace s nižší vyžadovanou přesností než v této aplikaci zadaná.

*Všechna naměřená data jsou v souboru <data/wifi-delays.xlsx>. Zdrojový kód měřícího skriptu je v <sources/wifi-delays.js>.*

## 4.2. Libelium Waspnote

Platforma Libelium byla vytvořena s cílem usnadnit budování levných senzorických sítí sestávajících z nezávislých nodů Waspnote a schopných pracovat měsíce z běžné Li-Ion baterie. Vznikla odštěpením z platformy Arduino, díky čemuž je kód mezi oběma ekosystémy do určité míry přenositelný.

Na rozdíl od platformy Arduino, která vznikla jako otevřený ekosystém k rychlému prototypování prakticky čehokoliv, má Libelium jasněji daný účel: sbírat data o prostředí a rádiovou sítí je odeslat do sběrné jednotky. K tomu cíli nabízí řadu periférií na vlastních interface deskách, které stačí „zacvaknout“ do příslušné patice na Waspnote. Ke všem jsou výrobcem dodávány odladěné knihovny a jejich použití vyžaduje – ve srovnání s Arduinem – jen minimum programování, ladění a pájení scházejících kondenzátorů.

### 4.2.1. Hardwarový popis

#### *Atmel ATmega1281*

Mikrokontrolér desky Waspnote je vybavenější verzí Atmelu používaného v systému Arduino. Stejně jako ATmega328 se jedná o 8bitový RISC procesor s výkonem 1 MIPS na MHz a nízkou spotřebou. Oproti němu však má čtyřnásobnou kapacitu volatilní i stálé paměti, konkrétně 8 kB SRAM (oproti 2 kB ATmega328), 4 kB EEPROM (1 kB) a 128 kB flash paměti (32 kB). Na desce Waspnote je taktován na 14 MHz. Výkonem a přidruženými snímkovacími frekvencemi je tedy srovnatelný s Arduinem (16 MHz).

#### *Napájení*

Waspnote je navržený na snímání dat bez závislosti na další infrastruktuře. Hlavními elektrickými vstupy tedy jsou konektor na 4,2V baterii a konektor na 6 – 12V solární panel. USB slouží hlavně jako alternativa solárního panelu k nabíjení baterií, pro běžný provoz má velmi omezený maximální odběr. Provoz desky bez připojené baterie je možný pouze bez periférií. Připojení i energeticky relativně nenáročného GPS modulu vede k nestabilitě.

Určení desky k provozu na baterie odpovídají i udávané hodnoty spotřeby: 15 mA při běžném zatížení, 55 µA v úsporném režimu (ukončitelném časovačem nebo přerušením na I/O pinu) a 0,7 µA v režimu hibernace (ukončitelném alarmem hodin reálného času).

#### *Rozhraní*

Waspnote zpřístupňuje ve třech patičkách s 2,54mm roztečí následující rozhraní:

- 8 digitálních pinů s 3,3V logikou, z toho jeden s 8b pulsní modulací
- 7 analogových pinů s 10b rozlišením a nenastavitelným 3,3V referenčním napětím
- 2 páry RX a TX UART sběrnice; UART0 je sdílená s USB a komunikačními moduly (WiFi, Bluetooth, Zigbee a další), UART1 s GPS modulem
- SDA a SCL sériového I<sup>2</sup>C rozhraní (sdílené s on-board RTC a akcelerometrem)

- MOSI, MISO a SCK sériového SPI rozhraní (sdílené s on-board microSD kartou)

Konkrétní pinout je k dispozici v hardwarovém manuálu <sup>23</sup>.

#### *Integrované moduly*

Přímo na desce Waspote je k dispozici akcelerometr a přesné hodiny reálného času. Akcelerometr STMicroelectronics LIS331DLH (v datasheetu chybně uvedený jako LIS331LDH) je I<sup>2</sup>C tříosý MEMS senzor s rozsahem  $\pm 2 \text{ g} - \pm 8 \text{ g}$  a maximální snímkovací frekvencí 1000 Hz. V prostředí Waspote je nejvyšší dosažitelný sample rate 800 Hz.

Použitý RTC čip je teplotně kompenzovaný Maxim DS3231SN s udávanou maximální odchylkou  $\pm 2$  ppm (0,16 s / den, 1 minuta / rok). Jeho použití v aplikaci synchronizovaného měření je bohužel omezeno absencí implementace funkce `micros()` nebo jejího ekvivalentu vracející čas s mikrosekundovou přesností <sup>24</sup>.

*Zdrojový kód použitý ke zjištění nejvyššího dosažitelného sample rate akcelerometru je v souboru [sources/waspote-accelerometer-samplerate.pde](https://github.com/libelium/waspote-accelerometer-samplerate.pde).*

#### 4.2.2. Softwarové prostředí

Libelium vzniklo odštěpením z platformy Arduino. Softwarové prostředí tomu odpovídá – programuje se ve stejném upraveném C++ a rozhraní dodávaného IDE se od Arduina liší ponejvíce výraznějším, červeným barevným schématem. Platí zde tedy většina toho, co bylo napsáno v části 4.1.3. SOFTWARE PROSTŘEDÍ v prostředí Arduino.

Na rozdíl od univerzální, k perifériím „agnostické“ platformy Arduino zde výrobce dodává s IDE oficiální knihovny ke všem svým senzorům a komunikačním modulům, stejně jako ukázkový kód k jejich ovládání. Jejich API je orientováno více na jednoduchost používání než na využití plného potenciálu modulů. Příkladem může být GPS knihovna: polohová data se získají jednoduchým příkazem `GPS.getPosition()`, čip však není možné nijak konfigurovat. Přestože použitý Telit JN3 umožňuje 5Hz frekvenci pozičních updatů, při použití dodávané knihovny je uživatel omezen na výchozí 1 Hz.

Kompatibilitě programů mezi systémy Libelium a Arduino částečně brání zastaralost prvně zmíněného. Libelium se oddělilo ve verzi ARDUINO 0002 vydané v roce 2005. Funkce a úpravy novějších verzí Arduina byly do Libelia portovány, patrně však pouze pokud nevyžadovaly výrazné zásahy do existujících knihoven. Takto například knihovna `HardwareSerial` (zprostředkující komunikaci po UART sběrnicích) stále používá synchronní API a knihovna `Wire` (pro I<sup>2</sup>C) používá metody `send()` a `receive()` místo novějších `write()` a `read()` (obě změny byly implementovány v ARDUINO 0100, vydané 30. 11. 2011 <sup>25</sup>).

#### 4.2.3. Možnosti uložení dat

Waspote nabízí všechna rozhraní, jaká byla použita k měření na platformě Arduino. Všechny metody zmíněné v části 4.1.5. MOŽNOSTI ULOŽENÍ DAT **CHYBA! NENALEZEN ZDROJ ODKAZŮ.** tedy lze použít i zde. Je tu však několik rozdílů, čtyřnásobná kapacita RAM například umožňuje používat větší buffery. Starší verze knihovny `SdLib` na druhou stranu podporuje pouze souborový systém Fat16. Jak se tyto změny projeví na dosažitelné šířce pásma změřím v následujících kapitolách.

Z měření na Waspote jsem vynechal 2,4 GHz rádio NRF24L01 ISM 2,4 GHz radio transceiver **CHYBA! NENALEZEN ZDROJ ODKAZŮ.** Kvůli nedostupnosti funkce `micros()` nemá význam měřit

<sup>23</sup> LIBELIUM COMUNICACIONES DISTRIBUIDAS S.L. Waspote Technical Guide [online]. V5.6. 2014, 167 s. [cit. 2015-05-16]. Dostupné také z:

[http://www.libelium.com/downloads/documentation/waspote\\_technical\\_guide.pdf](http://www.libelium.com/downloads/documentation/waspote_technical_guide.pdf)

<sup>24</sup> Libelium Forum: RTC Micro Seconds [online]. 2015. [cit. 2015-05-16]. Dostupné také z:

<https://www.libelium.com/forum/viewtopic.php?f=16>

<sup>25</sup> Arduino Release Notes [online]. 2015 [cit. 2015-05-16]. Dostupné také z:

<http://www.arduino.cc/en/Main/ReleaseNotes>

rozdíl výkonu při použití ATmega1281 – případné odchylky by byly zcela zastíněny nižším rozlišením měření.

#### SD karty Sandisk Extreme a LTL

Na rozdíl od použitých desek platformy Arduino má Waspote adaptér na SD karty přímo na desce. Použitý kontrolér ATmega1281 však má – stejně jako ATmega328 – jedinou SPI sběrnici, jedná se tedy pouze o „hardwarové“ zjednodušení. Při provozu více SPI zařízení je na softwarové úrovni stále nutné řešit vybrání správného přijímače přes CS pin.

Nejvýznamnější změnou oproti Arduino je použití starší verze SdLib, která je omezená na souborový systém Fat16. Z toho vyplývá maximální velikost datové partition 2 GB (jde pouze o omezení souborového systému, není problém použít 32GB SDHC kartu). Na druhou stranu je díky větší RAM možné použít několikakilobajtové buffery a tím ukládat data s poměrně nižší režii.

Benchmark probíhal podobně jako v případě systému Arduino: sekvenčním zápisem paketů dat o velikosti 1024, 2048 a 4096 B, přičemž se měřila latence každého zápisu. Naměřené hodnoty se vypsalý najednou každých 128 zápisů, aby výpis nedával kartě „odpočinout“ mezi zápisy.

		100%	99,99%	99,9%	99%	95%	50%	COUNT
1024	LATENCY	167,0	138,7	104,7	47,0	41,0	35,0	8320
	SPEED	6,1	7,4	9,8	21,8	25,0	29,3	
2048	LATENCY	155,0	153,1	80,4	63,0	58,0	52,0	4864
	SPEED	13,2	13,4	25,5	32,5	35,3	39,4	
4096	LATENCY	159,0	131,1	70,0	58,0	58,0	52,0	4992
	SPEED	25,8	31,3	58,5	70,6	70,6	78,8	

Tabulka 12 – Percentilové rozložení latencí (v ms) a rychlostí (v kBps) při zápisu paketů různé velikosti na kartu Sandisk.

Při měření doby ukládání na kartě Sandisk se se zvyšující velikostí datového paketu zvyšovala doba mediánové latence (35ms u 1024 B, 51 ms u 2048 B a 52 ms u 4096 B). Nicméně maximální latence zůstávala prakticky stejná 155 – 167 ms. Při použití 2kB bufferu (s 6 kB RAM zbývajících pro zbytek programu) tedy lze u karty Sandisk počítat s nejnižším stabilním datovým tokem 12 kBps, u 4kB bufferu pak s dvojnásobným 24 kBps.

U levnější karty LTLM byly výkony méně předvídatelné. Mediánové latence byly 85, 74 a 74 ms, maximální 329, 163 a 434 ms (pro 1024, 2048 a 4096 B). Vzhledem k nevysvětlitelnosti takových hodnot bylo provedeno ještě jedno měření, při kterém karta zcela přestala reagovat – ať už v čtečce ve Waspote nebo v PC. Bohužel se v termínu odevzdání této práce nestihla vyřídit reklamáce a test nové karty. Dané hodnoty je tedy nutné brát s rezervou.

Naměřená data jsou v souboru [data/waspote-sd-bench.xlsx](#), zdrojový kód v [sources/waspote-sd-bench.pde](#).

#### Microchip WiFly RN171 WiFi transceiver

WiFly RN171<sup>26</sup> je - podobně jako ESP 8266 - WiFi modul komunikující po UART sběrnici. Podporuje šifrované síť WPA2 a abstrahuje TCP a UDP transportní protokoly. Na rozdíl od ESP 8266 ale má vnitřní 1,5 kB buffer a data odesílá vždy až po naplnění 1440 B MTU. Také jeho API je přímočařejší, nevyžaduje „handshake“ s přáním odesílat data a délkou zasílaného paketu. Obojí by mohlo vést k vyšší efektivitě a tudíž vyšším přenosovým rychlostem než u ESP 8266.

Měření probíhalo stejně jako v sekci ESP 8266 WiFi TRANSCIVER, tedy v průměrných podmínkách příjmu, na WPA2 šifrovaném spojení a proti serveru na pevné síti za jediným směrovačem. Při každém

<sup>26</sup> MICROCHIP TECHNOLOGY INC. RN171 Datasheet [online]. 2014, 30 s. [cit. 2015-05-17]. Dostupné také z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70005171A.pdf>

měření bylo navázáno TCP spojení a v něm odesláno 128 paketů o délce 2048 B. Po jejich odeslání se spojení rozvázalo, vypsaly se naměřené hodnoty a další měření začalo od navázání spojení.

Měřenými veličinami byla v prostředí Wasmote doba odeslání dat (tedy doba mezi zavoláním metody `wifi.send()` a jejím vrácením). Na serveru pak doba mezi prvním a posledním paketem v rámci spojení a množství přijatých dat. Do rychlosti nebyly započítány „servisní“ úkony navázání a rozpadu TCP spojení. Scénář předpokládá, že tyto se v aplikaci provedou jen jednou.



	100%	99%	95%	90%	50%	COUNT
SPEED	1,69	2,61	4,03	5,56	10,45	194

Tabulka 13 – Percentilové rozložení rychlostí přijímaných dat na serveru. Hodnoty jsou v kBps.

Měření v aplikaci Wasmote reportovalo čas k odeslání 2kB stabilně jako 180 ms – tedy šířku pásma 11 kBps. Měření na serveru pak reportovalo 1,7 – 11,1 kBps, s průměrem 9,1 kBps. Všechna spojení měla stejnou dobu trvání kolem 23 s. Z toho vyplývá – a analýza přijatých dat potvrzuje – že WiFly v prostředí Wasmote neřeší ztrátu dat a spojení tak není spolehlivé. Ztrátu dat navíc nelze v prostředí Wasmote detekovat. Metoda `wiFi.send()` je implementována velmi naivně<sup>27</sup> a nenabízí žádnou zpětnou vazbu o úspěšnosti svého provedení.

Čip WiFly dle API dokumentace<sup>28</sup> nabízí CTS a jeho doplnění do dodávané knihovny je relativně jednoduché. Na druhou stranu dostupná šířka pásma je natolik malá, že by modul pro měřicí aplikaci nešlo použít, ani kdyby spojení bylo spolehlivé.

Naměřená data jsou v souboru [data/wasmote-wifi.xlsx](#). Zdrojový kód odesílací aplikace je v [sources/wasmote-wifi-sender.pde](#), server v [sources/wasmote-wifi-server.ls](#).

#### 4.2.4. Možnosti synchronizace

Vzhledem k podobné hardwarové výbavě nabízí Wasmote stejné MOŽNOSTI SYNCHRONIZACE, jako desky Arduino. S ohledem na sub-milisekundové přesnosti naměřené v předchozí kapitole nemá význam měření opakovat. Variace způsobené změnou platformy by byly zakryty omezením Wasmote na milisekundové rozlišení.

Na druhou stranu jsem ale na platformě Arduino neměl možnost otestovat WiFi transceiver WiFly RN171, protože jsem ho měl k dispozici pouze na Wasmote patici, jejíž pinout není v žádném dostupném manuálu<sup>23, 29</sup>.

##### *Microchip WiFly RN171 WiFi transceiver*

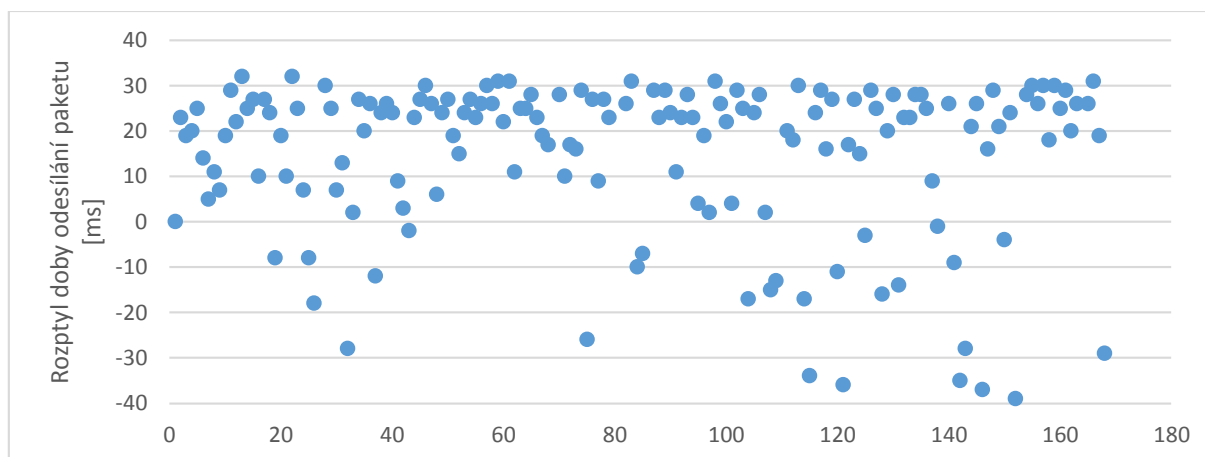
Pro účely přenosu dat se WiFly modul ukázal jako spíše nevhodný kvůli malé šířce pásma a ze strany standardní knihovny neřešenému řízení toku dat. Synchronizace však vyžaduje přenos pouze malého objemu dat a umožňuje mezi jednotlivými zprávami dělat i několikasekundové mezery, které plně dostačují na vyprázdnění veškeré vyrovnávací paměti. U předchozího modulu ESP 8266 pak docházelo k příliš velkým výkyvům v trvání přenosu. WiFly je ale výkonem i cenou zcela odlišný čip, proto jsem provedl separátní měření jeho rozptylu doby přenosu.

Kvůli zmíněné nedostupnosti pinoutu patice nebylo možné dodržet stejný postup jako u ESP 8266, které jsem ovládal z PC, a tím získal přesnou časovou kontrolu nad časováním jednotlivých příkazů. Zvolil jsem tedy odlišnou metodu, kdy jsem do vysílané zprávy zakódoval čas těsně předcházející jejímu vyslání a na serveru zaznamenal čas přijetí zprávy. V ideálních podmínkách by tedy měl být rozdíl obou časů konstantní. Měření probíhalo na stejné síťové konfiguraci jako všechna předcházející měření WiFi modulů.

<sup>27</sup> LIBELIUM COMUNICACIONES DISTRIBUIDAS S.L. Wasmote API Source Code [online]. V015. 2015 [cit. 2015-05-17]. Dostupné také z: <https://github.com/Libelum/wasmoteapi/blob/a70c91900290faec4bfe90dd2f7769bd97555a2f/libraries/WIFI/WaspWiFi.cpp#L3652>

<sup>28</sup> MICROCHIP TECHNOLOGY INC. WiFly Command Reference, Advanced Features and Applications User's Guide [online]. 2014, 120 s. [cit. 2015-05-17]. Dostupné také z: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002230A.pdf>

<sup>29</sup> LIBELIUM COMUNICACIONES DISTRIBUIDAS S.L. WiFi Module Networking Guide [online]. V5.6. 2014, 54 s. [cit. 2015-05-17]. Dostupné také z: [http://www.libelum.com/downloads/documentation/wifi\\_networking\\_guide.pdf](http://www.libelum.com/downloads/documentation/wifi_networking_guide.pdf)



Obrázek 18 – Rozptyl doby odeslání paketu. První časová značka na Waspnote i serveru byla nastavena na 0, další měření jsou vztažena k ní. Nejmenší naměřená hodnota byla -77 ms, pro větší přehlednost jsou zobrazeny pouze hodnoty nad -40 ms (95 % měření).

	100%	99%	95%	90%	50%	COUNT
MAX	32	31	30	29	22	168
MIN	-77	-58	-42	-27		
DIFF	109	89	72	56		

Tabulka 14 – Percentilové rozložení doby odeslání paketu.

Rozdíl extrémů dosáhl 109 ms, polovina zpoždění pak byla v intervalu 22 – 32 ms. Pro účely časové synchronizace tak nelze ani v případě opakovaného měření počítat s vyšší přesností než desítky milisekund.

Naměřená data jsou v souboru [data/waspnote-wifi.xlsx](#). Zdrojové kódy vycházejí z měření šířky pásma a jsou pro Waspnote v [sources/waspnote-wifi-delay-server.ls](#), server v [sources/waspnote-wifi-delay-server.ls](#).

### 4.3. National Instruments CompactRIO

Zatímco dosud představené desky postavené na mikrokontrolérech Atmel jsou určeny jako „vykrývací“ jednotky s jediným připojeným senzorem, systém CompactRIO (Reconfigurable Input / Output, dále v textu zkracováno na „cRIO“) je rodina výkonných počítačů s možností připojení – a zpracování dat z – desítek senzorů.

Systém CompactRIO není přímo předmětem této práce, zadání však požaduje, aby data naměřená mikrokontroléry byla časově synchronizovatelná s hodnotami z cRIO a aby bylo možné spouštět a ukončovat měření na všech zařízeních z jednoho místa. Z tohoto důvodu v této kapitole stručně popíšu relevantní části systému cRIO.

#### 4.3.1. Hardwarový popis

Základními prvky všech modelů cRIO jsou výkonný procesor, programovatelné hradlové pole a vstupně-výstupní moduly. Jednotlivé modely a jejich cena pak ovlivňují výkon procesorů (od 233 MHz single-core po 1,33 GHz dual-core), FPGA (85 – 326 tisíc logických buněk) i velikost šasi a tím počet připojitelných I/O modulů.

Moduly jsou k cRIO připojeny přes 15pinový D-SUB konektor, kterým je vedené napájení (5 V) i komunikace (3,3 V logická úroveň, 5 V tolerantní). Komunikační rozhraní doporučuje NI jedno ze dvou: sériové SPI (s jedním „trigger“ pinem, sloužícím jako zdroj přerušení) nebo paralelní s osmi digitálními vstupně/výstupními (DIO) piny, jejichž chování lze uživatelsky definovat z programovacího prostředí.

Všechny moduly musí podporovat tzv. ID mód, pokud je na D-SUB pinu 1 logická 0. V tomto módu musí modul umožnit přístup k 16B EEPROM obsahující data jako ID výrobce, modelu či sériového čísla modulu. Přístup k EEPROM je vždy přes SPI, i pokud „provozní“ mód pracuje s DIO rozhraním.

Detaily konstrukce a zapojení I/O modulů jsou popsány v manuálu vývoje modulů <sup>30</sup>.

Šířku pásma a obnovovací frekvenci v závislosti na počtu připojených I/O modulů řeší NI White Paper Scan Engine Performance Benchmarks <sup>31</sup>. V uživatelsky jednodušším Scan Mode je maximální obnovovací frekvence 1 000 Hz, přičemž je dosažitelná s až 16 paralelními kanály. Použitím pokročilejšího FPGA módu je pak možné dosáhnout frekvence 7 100 Hz při přístupu na jediný kanál a 3 500 Hz u 16 kanálů.

S okolním světem CompactRIO kontroléry komunikují přes rozhraní USB, Ethernet (10/100BASE-TX) a RS-232. Scénář měření na Fakultě dopravní pak počítá s připojením konfigurovatelných WiFi směrovačů, tzv. access pointů. V tomto případě ale budou konfigurovány jako klient a pouze přemostovat bezdrátovou síť do drátové.

#### 4.3.2. Softwarové prostředí

Výrobcem doporučený postup pro programování systémů CompactRIO je z prostředí LabView. Jedná se o vizuální programovací jazyk, aplikace se programují přesouváním a propojováním jednotlivých modulů. Cílem LabView je, aby cRIO mohli nastavovat a prototypovat přímo vědci nebo technici s doménovou znalostí měřené problematiky, ale bez znalosti konkrétního programovacího jazyka.

Z LabView je možné ovládat všechny režimy systému cRIO – jednoduchý Scan mode, výkonný FPGA mode i jejich kombinaci Hybrid mode. Je možné ovládat prioritu jednotlivých procesů včetně managementu práce v reálném čase. Jednoduchá aplikace, která sbírá data ze senzorů a přes vyrovnávací paměť je ukládá na disk, je dokumentována na webových stránkách NI <sup>32</sup>.

Při snímání dat je možné každé naměřené hodnotě přiřadit časovou značku s mikrosekundovým rozlišením. Pár hodnota – časová značka se poté bude do datového souboru ukládat společně.

LabView také umožňuje nastavit, aby vybrané proměnné byly čitelné a/nebo zapisovatelné přes HTTP REST rozhraní <sup>33</sup>. Kromě standardní implementace HTTP metod nabízí také kompatibilní režim, kdy se využívá metoda GET a požadovaná akce se definuje URL parametrem. Pomocí metody Set HTTP Header <sup>34</sup> je možné nastavit hlavičky, například Access-Control-Allow-Origin a umožnit tak ovládání spuštěného programu z běžného webového prohlížeče.

#### 4.3.3. Možnosti uložení dat

Data mohou být ukládána buď do vnitřní nevolatilní flash paměti (podle modelu 128 MB – 4 GB) nebo pomocí rozšiřujícího modulu NI 9802 na dvě SD karty (omezeno Fat16 souborovým systémem,

---

<sup>30</sup> NATIONAL INSTRUMENTS CORPORATION. 2009. CompactRIO Module Development Kit User Manual: NI cRIO-9951. [cit. 2015-05-18].

<sup>31</sup> NATIONAL INSTRUMENTS CORPORATION. 2010. Scan Engine Performance Benchmarks [online]. [cit. 2015-05-18]. Dostupné také z: <http://www.ni.com/white-paper/7792/en/>

<sup>32</sup> Getting Started with CompactRIO - Logging Data to Disk. National Instruments: Tutorials [online]. 2011 [cit. 2015-05-19]. Dostupné z: <http://www.ni.com/tutorial/11198/en/>

<sup>33</sup> INTERNET ENGINEERING TASK FORCE. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content [online]. 2014 [cit. 2015-05-19]. Dostupné také z: <https://tools.ietf.org/html/rfc7231#section-4.3>

<sup>34</sup> Configuring HTTP Headers, Streaming, and Buffering. National Instruments: Tutorials [online]. 2012 [cit. 2015-05-19]. Dostupné z: [http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/ws\\_configuring\\_headers](http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/ws_configuring_headers)

tedy s maximální kapacitou 2×2 GB). Vnitřní paměť má udávanou šířku pásma až 2,2 MBps <sup>35</sup>, NI 9802 pak 2 MBps <sup>36</sup>.

Další možností je využití USB zařízení podporující standard Mass Storage s udávanou šířkou pásma 700 kBps <sup>35</sup> (rychlost specifikovaná na straně cRIO, připojené USB zařízení musí být minimálně stejně rychlé).

Data uložená ve vnitřní paměti, v SD modulu nebo na USB jsou dostupná přes webový interface a po skončení měření stažitelná přes standardní HTTP protokol.

Ethernetový port je také možné použít k reálnému streamování dat po TCP protokolu na dostupný server. Specifikovaná rychlost je 1MBps <sup>35</sup>.

#### 4.3.4. Možnosti synchronizace

##### *GPS PPS signál*

Vzhledem k centralizované architektuře systému a z toho plynoucího nízkého počtu jednotek, jež je třeba synchronizovat, se jako nejvhodnější synchronizační metoda jeví využití GPS PPS pinu. Poskytuje přesný absolutní čas a s ohledem na finanční náročnost cRIO systémů je většina přijímačů cenově dostupná <sup>37</sup>.

Způsob zapojení a dosažitelné přesnosti z hlediska cRIO systému jsou popsány v NI White Paper 8 Ways to Use a Digital Module in a NI CompactRIO System <sup>38</sup>. Princip fungování PPS pinu a přesnost časového signálu z nezávislých GPS přijímačů popisují v této práci v kapitole 2.

---

<sup>35</sup> NATIONAL INSTRUMENTS CORPORATION. Data Storage Options for CompactRIO [online]. 2008 [cit. 2015-05-18]. Dostupné také z: <http://www.ni.com/white-paper/6884/en/>

<sup>36</sup> NATIONAL INSTRUMENTS CORPORATION. 2009. NI 9802 OPERATING INSTRUCTIONS AND SPECIFICATIONS [online]. 20 s. [cit. 2015-05-18]. Dostupné také z: <http://www.ni.com/pdf/manuals/374181b.pdf>

<sup>37</sup> CompactRIO Advisor. National Instruments [online]. 2015 [cit. 2015-05-20]. Dostupné z: <http://ohm.ni.com/advisors/crio/pages/common/intro.xhtml>

<sup>38</sup> 8 Ways to Use a Digital Module in a NI CompactRIO System. National Instruments: White Papers [online]. 2013 [cit. 2015-05-20]. Dostupné z: <http://www.ni.com/white-paper/14549/en/#toc6>

### NTP

V případě, že by měření umožňovalo nižší (milisekundovou) přesnost, se nabízí časová synchronizace přes síťový časový protokol NTP <sup>39</sup>. Její výhodou je, že vyžaduje pouze časový server a síťové spojení s ním, vše ostatní je otázkou jednoduché softwarové konfigurace <sup>40,41</sup>. Vzhledem k tomu, že samotný scénář měření vyžaduje přítomnost serveru, na který se po síti nahrávají naměřená data, se jedná o pravděpodobně nejefektivnější řešení z hlediska poměru nákladů a výkonu.

Na druhou stranu je třeba brát v úvahu, že NTP není navržen s cílem dosáhnout milisekundové přesnosti. Síťový standard Ethernet je definován jako nerealtimový, doba přenosu paketu může být při každém pokusu jiná – ovlivněna nepředvídatelným síťovým provozem. Přestože NTP implementuje metody na potlačení vlivu latence a jitteru, nemůže je zcela potlačit.

Realtimovost je možné do velké míry nahradit použitím málo vytížené sítě, obvykle se za prakticky realtimeovou považuje síť využitá na méně než 1 %. Takto nevytíženou místní síť je možné při měření zajistit. Lokální časový server pak může distribuovat relativní čas, všichni klienti pak budou mít hodiny *stejně špatně*.

V případě požadavku na absolutní čas by bylo třeba zajistit jeho zdroj. Opět se nabízí GPS, například použitím serveru nabízejícího digitální vstupní piny (například Raspberry Pi nebo Arduino Yún). Pokud by dostačovala nižší přesnost, je též možné použít vzdálený NTP server s absolutním časem.

Aby bylo možné odhadnout přesnost časové synchronizace při použití NTP, provedl jsem několik měření rozptylu jím vracených hodnot. Protože cRIO nenabízí potřebná API (nelze na programovou žádost provést synchronizaci k NTP serveru), použil jsem v roli klienta PC Raspberry Pi 2. Jako časový etalon sloužil PPS signál GPS čipu u-blox MAX-7C.

Měření probíhala proti lokálnímu i vzdálenému časovému serveru, s drátovým i bezdrátovým připojením. Lokální server představoval směrovač Mikrotik RB951G pluginem rozšířený o NTP funkcionalitu. Vzdálený server byl použit tik.cesnet.cz, protože měl z několika vybraných nejstabilnější výsledky (kandidáty byly ntp.nic.cz, tak.cesnet.cz, 0.cz.pool.ntp.org, 3.cz.pool.ntp.org a time.windows.com).

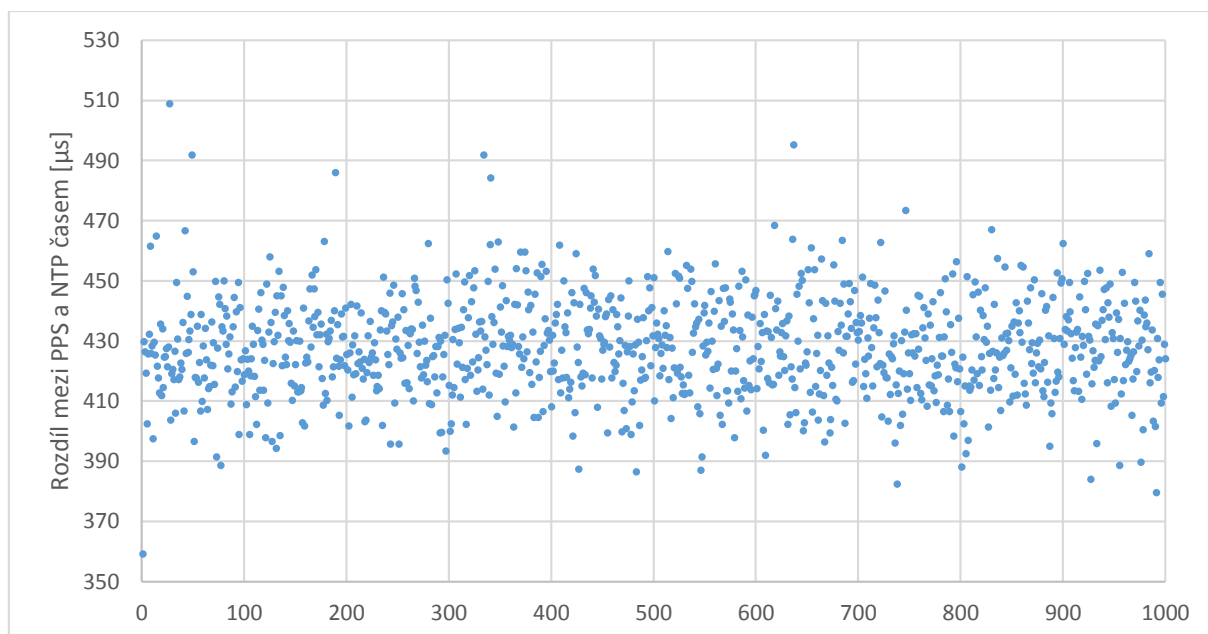
### Lokální server, drátové připojení

Nejlepší scénář měření, kdy byl mezi klientem a serverem jediný síťový segment s prakticky nulovým vytížením.

<sup>39</sup> INTERNET ENGINEERING TASK FORCE. Network Time Protocol Version 4: Protocol and Algorithms Specification [online]. 2010 [cit. 2015-05-20]. Dostupné také z: <https://tools.ietf.org/html/rfc5905>

<sup>40</sup> How Do I Configure My CompactRIO Real-Time Controllers to Synchronize to SNTP Servers? National Instruments: Knowledge Base [online]. 2007 [cit. 2015-05-20]. Dostupné z: <http://digital.ni.com/public.nsf/allkb/F2B057C72B537EA2862572D100646D43>

<sup>41</sup> How Do I Configure My Host Computer as an NTP Time Server? National Instruments: Knowledge Base [online]. 2009 [cit. 2015-05-20]. Dostupné z: <http://digital.ni.com/public.nsf/allkb/EA90C9FF24D9A041862575EE004ED415>



Obrázek 19 – Rozdíl mezi GPS časem a časem lokálního NTP serveru. Měření probíhalo každou vteřinu, 1 000 samplů tedy odpovídá přibližně 17 minutám. Data jsou opraveny o drift hodin lokálního serveru.

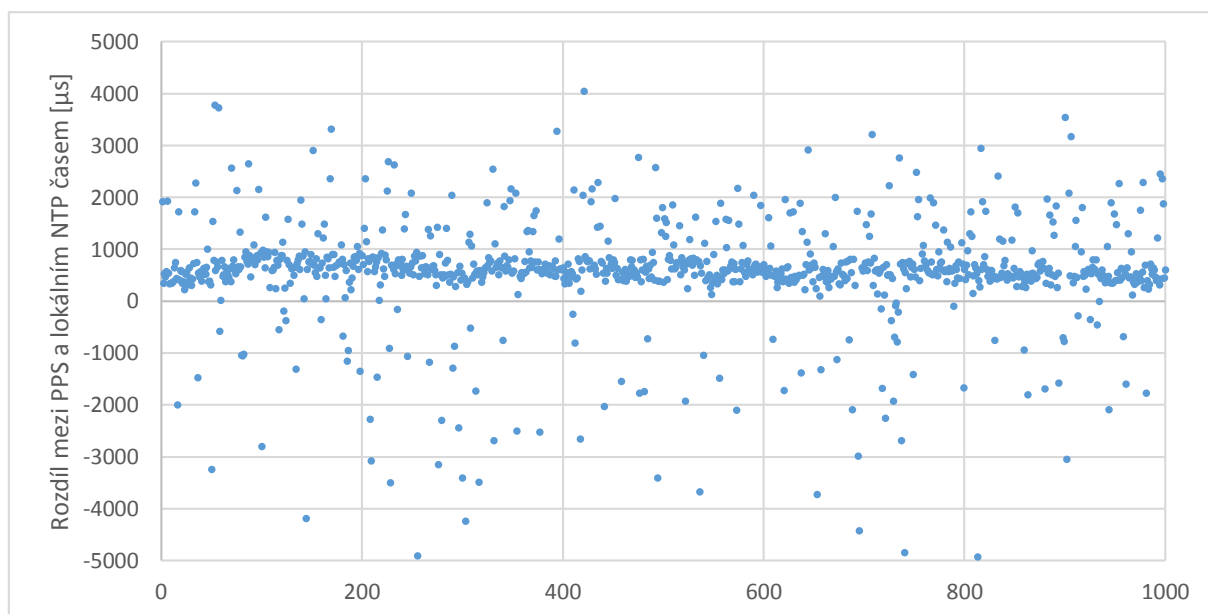
	100%	99,99%	99,9%	99%	95%	90%	80%	50%	COUNT
MAX	509	507	495	465	453	448	440	427	1000
MIN	359	361	380	390	402	408	415		
DIFF	150	146	116	75	51	40	26		

Tabulka 15 – Percentilové rozložení naměřených hodnot. Údaje jsou v  $\mu s$ .

Z měření je vidět, že na lokální síti lze dosáhnout velmi přesného relativního času s přesností přes  $\pm 0,15$  ms.

#### Lokální server, bezdrátové připojení

Bezdrátové připojení na lokální server bylo realizováno pomocí WiFi, jedná se tedy o simulaci synchronizace aut na začátku měření. Použitý WiFi modul na RPi byl MediaTek MT7601. Měření probíhalo v průměrných podmínkách příjmu. NTP server byl umístěn přímo na WiFi směrovači.



Obrázek 20 – Rozdíl mezi GPS a NTP časem při použití WiFi. Nevýrazný drift je způsoben nepřesnými hodinami serveru.

	100%	99,99%	99,9%	99%	95%	90%	80%	50%	COUNT
MAX	23144	22720	18897	3544	2000	1543	873	599	1000
MIN	-20957	-20546	-16848	-8334	-2652	-760	368		
DIFF	44101	43265	35745	11878	4653	2303	506		

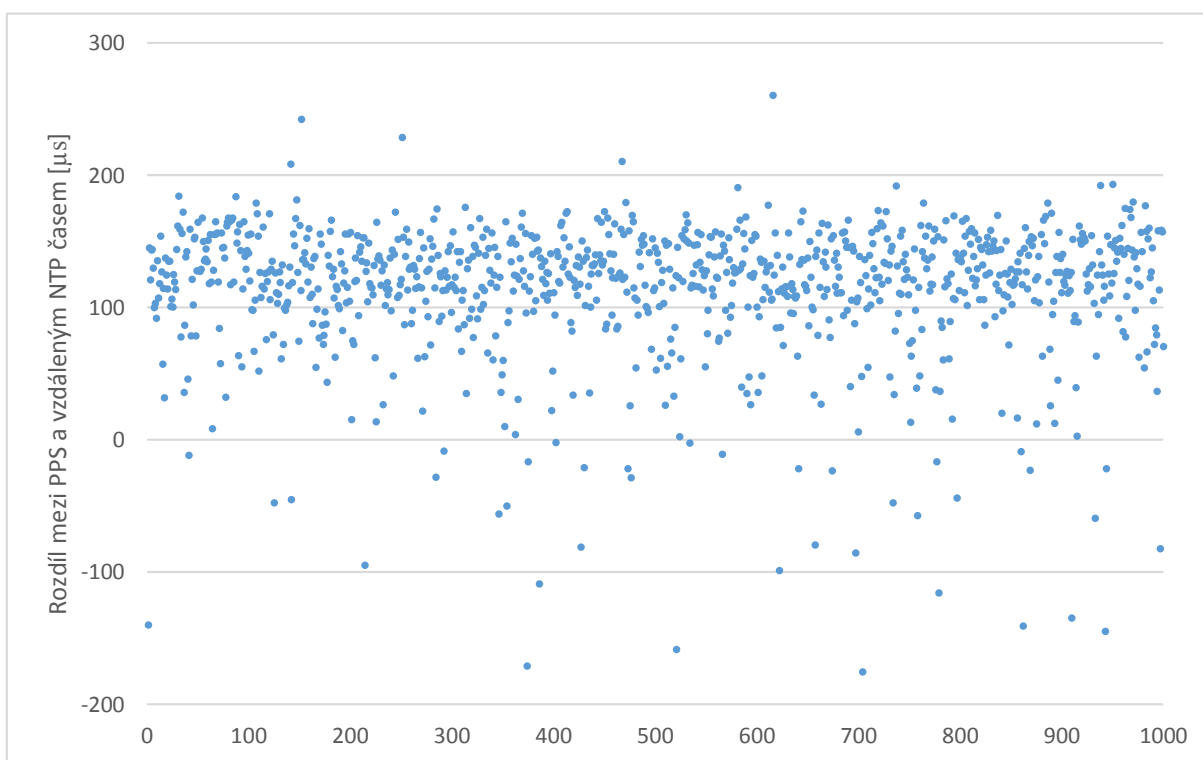
Tabulka 16 – Percentilové rozložení rozdílů mezi GPS a NTP časy.

Maximální naměřený rozptyl byl 44 ms, na grafu je však vidět výrazný pruh o šířce 0,5 ms, kam spadla většina měření. Při použití několika samplů a jejich následném statistickém vyhodnocení by tedy mělo být možné dosáhnout požadované přesnosti vyšší než  $\pm 2$  ms.

Na druhou stranu prostředím cRIO takovou práci s NTP neumožňuje. Řešením by mohly být timeservery schopné jednoduchého skriptování, umístěné na ethernetovou část sítě (mezi cRIO a jeho WiFi směrovač). Je však otázkou, zda by takové řešení nezkomplikovalo celý systém měření nad únosnou mez.

#### Vzdálený server, drátové připojení

V dalším měření byl použit vzdálený server s absolutním UTC časem. Měření bylo provedeno proti serveru tik.cesnet.cz na velmi kvalitní lince třídy FTTx (optické vlákno končící dva síťové segmenty od klienta). Jedná se tedy o nejlepší dosažitelný scénář při použití vzdáleného serveru.



Obrázek 21 – Rozdíl mezi GPS časem a časem získaným ze vzdáleného NTP serveru. Graf je kvůli dostatečnému rozlišení osy Y oříznutý, není zobrazeno 9 hodnot (0,9 %).

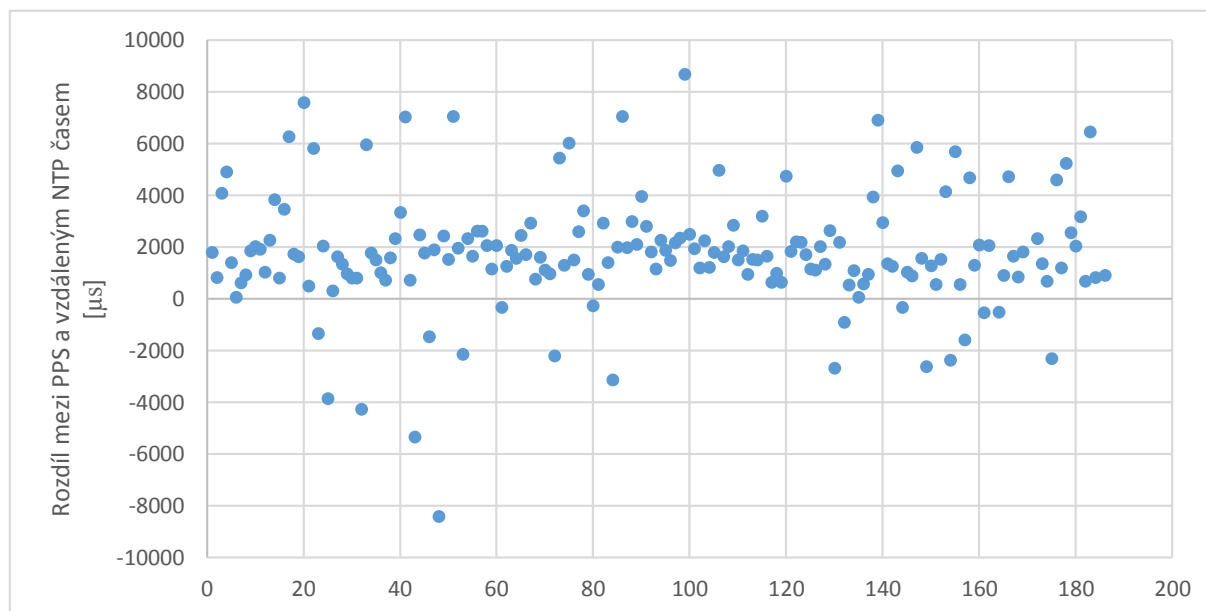
	100%	99,99%	99,9%	99%	95%	90%	50%	COUNT
MAX	543	536	475	192	168	160	125	1000
MIN	-915	-892	-686	-140	12	55		
DIFF	1457	1428	1161	332	155	105		

Tabulka 17 – Percentilové rozložení naměřených odchylek od GPS etalonu. Hodnoty jsou v  $\mu$ s.

Maximální naměřený rozptyl byl 1 457  $\mu\text{s}$ , což je stále dostačující pro zkoumanou aplikaci. Při opakovaném měření a použití vhodné statistické metody je limitujícím faktorem přesnost NTP serveru. Za ideálních podmínek se dá očekávat přesnost 0,1 – 0,2 ms.

#### Vzdálený server, bezdrátové připojení

Poslední scénář byl opět měřený proti serveru tik.cesnet.cz, připojení k němu ale bylo realizováno přes WiFi a mobilní 4G připojení (telefon sloužil jako access point). Cílem bylo zjistit, jak přesný absolutní čas je možné získat „v poli“, tedy bez pevného připojení a bez specializovaných zařízení.



Obrázek 22 – Rozdíl mezi GPS časem a časem získaným ze vzdáleného NTP serveru přes mobilní 4G připojení. Kvůli zvýšení rozlišení osy Y je graf oříznutý, není zobrazeno 6 (3 %) hodnot.

	100%	99%	90%	80%	75%	50%	COUNT
MAX	20490	15676	5512	3111	2603	1687	188
MIN	-8395	-4392	-279	813	948		
DIFF	28885	20068	5791	2299	1655		

Tabulka 18 – Percentilové rozložení naměřených hodnot

Dle očekávání je tato varianta nejméně přesná, celkový rozptyl hodnot byl téměř 30 ms. Opět je však patrný užší pás mezi 0 – 4 ms, který obsahuje 93 (49 %) hodnot. Při použití dostatečného počtu vzorků tedy lze očekávat přesnost do 4 ms.

Při tomto měření se projevila značná nekompatibilita mezi Linuxovou správou bezdrátových sítí, WiFi modulem MediaTek MT7601 a telefonem Nokia Lumia 925. Bylo proto naměřeno méně vzorků, než v ostatních měřeních.

Všechna naměřená data, včetně porovnání různých NTP serverů, jsou v souboru [data/ntp-accuracy.xlsx](#). Zdrojový kód měřicího skriptu je v [sources/ntp-accuracy.py](#).



## 5. Vyhodnocení použitých metod

V minulé kapitole jsem představil a naměřil řadu metod použitelných buď k ukládání dat a komunikaci, nebo k časové synchronizaci. Na následujících stranách shrnuji výkonnostní a cenové parametry a diskutuji možnosti kombinace jednotlivých modulů.

Problematika vzájemné kompatibility modulů se týká především systémů Arduino a Waspote. Vzhledem k požadavku měření nejen bezdrátově synchronizovat, ale též ovládat jeho spuštění a ukončení, je u některých metod kombinace více periférií nevyhnutelná.

Systém CompactRIO má proti Atmelovským mikrokontrolérům mnohem vyšší výkon a robustnější architekturu. Není tak třeba brát ohledy na počet periférií připojených k jedné sběrnici. Budu se tak zaměřovat hlavně na zhodnocení kompatibility jednotlivých metod s mikrokontroléry, tedy jak složité by bylo ovládat nebo synchronizovat cRIO i mikrokontroléry jednou metodou.

V předposlední části této kapitoly stručně zhodnotím rozdíly mezi mikrokontroléry Arduino a Waspote. Nakonec uvádím výsledky synchronizace skutečného měření zrychlení na dvou nezávislých mikrokontrolérech.

### 5.1. Metody ukládání nebo přenosu dat

Požadavky na ukládání plynou z datového toku, který generuje kinematický senzor. V uvažované aplikaci se počítá s 1kHz snímkovací frekvencí a 2B rozlišením každé měřené osy. Ke každému měření je pak nutné přidat časovou značku a případně příznak, zda v době měření byl detekován signál na synchronizačním pinu. Tyto metadata představují další 2B ke každému měření.

Celkově tedy požadovaný datový tok vychází u jedné osy 4 kBps, u dvou 6 kBps a u tří 8 kBps. Požadavky lze zmírnit také použitím nižší vzorkovací frekvence, dojde ale ke 2,5 násobnému snížení maximální rozpoznatelné oscilační frekvence<sup>13</sup>.

	Metoda	Interface	Rychlost [kBps]				Cena
			100%	99,99%	99,9%	99%	
Arduino	SD karta	SPI	6,2	6,2	8,7	73,4	650 Kč
	NRF24L01 2,4GHz radio	SPI	18,5	18,5	58	58	30 – 80 Kč
	ESP 8266 WiFi	UART	0,7	1,2	2,5	4,4	70 – 200 Kč
Waspnote	SD karta	SPI	25,8	31,3	58,5	70,6	620 Kč
	WiFly RN171	UART	1,69	1,7	1,8	2,61	1 640 Kč <sup>42</sup>
cRIO	Interní flash		2252				
	USB Mass Storage	USB	700				
	SD karta	SPI	2048				34 000 Kč <sup>36</sup>
	Ethernet	Ethernet	1024				

Tabulka 19 – Shrnutí výkonů uvažovaných metod ukládání nebo přenosu dat podle požadované spolehlivosti.

Z měřených metod pro mikrokontroléry vychází nejlépe 2,4 GHz transceiver Nordic Semiconductor NRF24L01. I při požadavku na absolutní spolehlivost je schopný s rezervou přenést data ze všech tří os akcelerometru. Vzhledem k velmi nízkému rozptylu doby odeslání jej lze využít i jako zdroj synchronizačního signálu. A díky half-duplexovému charakteru komunikace jej lze použít i k ovládání měřících zařízení. Jediné zařízení tak zastane role všech tří potřebných periférií.

Jeho největší nevýhodou je nízký dosah a z toho plynoucí nutnost mít zpracovávací jednotku (například notebook) na palubě každého auta. Na druhou stranu se tím ale získá možnost reálného vyhodnocení dat.

Další možností je použití SD karty. Její největší výhodou je nezávislost na dalších zařízeních. Snímací jednotka se senzorem, zdrojem času a SD kartou nevyžaduje žádné externí vybavení a je schopna pracovat zcela samostatně. Při provozu na baterie je také plusem její minimální spotřeba.

Nevýhodou je zejména náročnost stáhnutí dat do centrálního úložiště. Jediná praktická možnost je kartu fyzicky vyjmout ze zařízení, přesunout k serveru a data ručně zkopírovat.

UART WiFi moduly se pro přenos dat neukázaly jako vhodné, spolehlivě lze očekávat šířku pásma menší než 2 kbps. Zajímavé je však jejich použití v roli modulu řídicího započítání a ukončení měření. Lze na nich snadno postavit jednoduchý HTTP server s podobným REST API jako cRIO. Navíc pracují na – při měření obvykle nevyužitě – sběrnici UART, takže „nepřekáží“ ostatním modulům.

U systému CompactRIO je možné bez významnějších omezení vybírat ze všech dostupných metod. Možné faktory jsou zejména omezená kapacita interní paměti a – při použití velkého množství náročných senzorů – nízká šířka pásma při použití USB Mass Storage zařízení. Při použití čtečky SD karet je třeba počítat s její vyšší cenou.

<sup>42</sup> LIBELIUM COMUNICACIONES DISTRIBUIDAS S.L. Product Catalogue 2015. 2015, 65 s. [cit. 2015-05-24].

## 5.2. Synchronizační metody

Zadáním požadovaná přesnost měření byla  $\pm 2$  ms, přičemž je akceptovatelné použití relativního (všechny hodiny jdou *stejně špatně*) i absolutního (UTC) času. Z měřených metod obsahují vlastní zdroj absolutního času pouze GPS, a prakticky také NTP servery. Jejich hodnoty rozptylu tedy odpovídají celkové chybě měření absolutního času. U ostatních zařízení je nutné kromě chyby metody počítat i s chybou zdroje času.

	Metoda	Interface	Rozptyl [ $\mu$ s]				Cena
			100%	99,99%	99,9%	99%	
Arduino	GPS	DIO	24	24	24	20	350 Kč
	Fotodioda	ADC	124	124	20	12	30 Kč
	433MHz Radio	DIO	44	40	28	28	30 Kč
	NRF24L01	SPI	52	52	52	52	30 – 80 Kč
	ESP 8266 WiFi	UART	764000		676100	116000	70 – 200 Kč
WM	WiFly RN171	UART	109000			89000	1 640 Kč
CRIO	GPS	DIO	1				350 Kč
	GPS NI 9467	SPI	0,2				16 090 Kč <sup>43</sup>
	NTP	Ethernet	1457	1428	1161	332	
	NTP	WiFi	44101	43265	35745	11878	
	NTP	4G	28885			20068	

Tabulka 20 – Shrnutí uvažovaných metod synchronizace. Některé percentily nejsou uvedeny kvůli nízkému počtu vzorků. Hodnota maximálního rozptylu NTP po 4G síti je ovlivněna velmi krátkou dobou měření.

Ze shrnutí je vidět, že při použití vhodné metody lze i s nevýkonnými mikrokontroléry v ceně pár stokorun lze dosahovat přesnosti v řádu desítek mikrosekund, o dva řády pod požadovanými 2 milisekundami.

### 5.2.1. GPS

Nejvýhodnější metodou pro synchronizaci mikrokontrolérů se, očekávatelně, stala GPS. Při jejím použití je limitujícím faktorem přesnost snímání digitálních vstupně-výstupních (DIO) pinů. Hlavními výhodami jejího použití jsou poskytování absolutního času, snadnost integrace (zapojení jediného pinu) a samostatnost (vyžaduje pouze napájení).

Nevýhodou GPS je její vyšší cena (v porovnání s ostatními metodami) a náročnost na umístění s ohledem na dostatečnou kvalitu příjmu satelitních signálů.

<sup>43</sup> NI 9467: GPS Timestamping and Synchronization Module. National Instruments [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/210834>

### 5.2.2. Fotodioda a 433MHz rádiová spoušť

Fotodioda a 433MHz rádiová spoušť představují jednoduché metody, jak synchronizační puls bezdrátově broadcastovat do více zařízení. Jejich nejvýhodnější použití je právě v kombinaci s GPS – jediný přijímač se může umístit do místa s dobrým příjmem a jejich prostřednictvím vysílat všem měřicím zařízením v autě.

Výhodou fotodiody je její nenáročnost na zdroj napájení, může být napájena přímo PPS pinem GPS. Nevýhodou je kromě nutnosti vedení napájecího a zemnicího kabelu její náchylnost na mechanické znečištění.

Rádiová spoušť tyto nevýhody nemá, vyžaduje ale napájení 5 – 12 V. Vzhledem k velkému rozsahu povoleného napětí je však možné k napájení použít řadu zdrojů, například neregulovaný vstup z autobaterie.

### 5.2.3. NRF24L01 2,4 GHz Radio

Modul NRF24L01 je k synchronizaci nejvýhodnější použít, pokud již je využíván na přenos dat. Umožní tak konstrukci velmi kompaktního měřicího zařízení, sestávajícího pouze z trojice mikrokontrolér – senzor – NRF24L01.

V případě použití pouze jako synchronizační signál je nutné počítat zejména s tím, že se na přijímači bude dělit o SPI sběrnici s SD kartou, čímž klesne dostupná šířka pásma.

Konstrukce vysílače pak bude ztížena tím, že na rozdíl od ostatních metod si nevystačí se zdrojem signálu v podobě jednoduché linky se střídající se logickou úrovní. Vyžaduje ovládající mikrokontrolér nebo jiné zařízení s SPI sběrnici.

### 5.2.4. UART WiFi moduly

Oba testované moduly se neukázaly jako vhodné pro přijímání zpětných pulsů. U čipu WiFly bylo možné dosáhnout rozptylu 109 ms, u ESP 8266 pak 764 ms. Obě hodnoty jsou řádově vzdálené požadovaným 4 ms.

### 5.2.5. NI cRIO GPS

Při synchronizaci systému CompactRIO pomocí GPS je možné vybírat ze dvou přístupů. Lze použít levný GPS modul a dovést jeho PPS signál na DIO kanál cRIO modulu, například NI 9402<sup>44</sup>. V takovém případě bude přesnost odpovídat přesnosti PPS pinu, která je u levných GPS přijímačů obvykle udávána 1  $\mu$ s<sup>5,8</sup>.

Na jeden PPS pin lze připojit více zařízení, je tak možné ze stejného modulu obsluhovat cRIO i například 433MHz rádiovou spoušť pro mikrokontroléry.

Druhou možností je pořídit dedikovaný modul NI 9467. Jeho udávaná přesnost je  $\pm 100$  ns<sup>43</sup>. Modul však nenabízí externí PPS pin, jedná se o jednoúčelové zařízení pro časovou synchronizaci s cRIO.

### 5.2.6. NI cRIO NTP

Možnost synchronizace s NTP servery je v prostředí cRIO omezená tím, že nelze zvolit počet requestů, z nichž by se pak vybrala střední hodnota. Jedinou fyzickou vrstvou, která je schopná konzistentně zajistit dostatečnou přesnost, je kabelový Ethernet. Bezdrátové technologie mají při jediném samplu příliš velkou nejistotu.

Zapojovat a následně rozpojovat kabely před každým měřením by bylo možné, ale zvláště při použití více jednotek cRIO poněkud nepraktické. Dále existuje možnost umístit PC schopné vícenásobného smplování na Ethernetovou síť cRIO a provozovat lokální NTP server na něm. Jedná se však o řešení komplikovanější, dražší a méně přesné, než implementace GPS PPS synchronizace.

<sup>44</sup> NI 9402: LVTTTL, Bidirectional Digital I/O, 4 Ch Module. National Instruments [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/208810>

NTP synchronizace se tak nejlépe uplatní u jiných aplikací, kterým bude stačit přesnost synchronizace po WiFi, tedy  $\pm 25$  ms. Je přitom nutné počítat s další nepřesností při získávání absolutního času, při použití 4G mobilní sítě  $\pm 2$  ms.

Při použití relativního času by bylo nutné vyřešit, jak jej dostat i na mikrokontroléry Arduino či Wasmote, zejména pokud server nemá vlastní digitální IO piny nebo SPI sběrnici. V dané přesnosti sice dostačuje použití libovolného z mikrokontrolérů v roli vysílače ovládaného přes USB-UART převodník, opět to ale komplikuje synchronizační systém.

### 5.3. Možné konflikty při kombinování mikrokontrolérových modulů

Všechna měření byla provedena tak, aby sběrnice hostící synchronizační, resp. data ukládající modul již neobsluhovala žádné další zařízení. Při sdílení kapacity sběrnice více periferiemi je nutné počítat se snížením přesnosti detekce synchronizačních pulsů, zmenšením dostupné šířky pásma a také s vyšší komplexitou ovládacího software.

Z toho důvodu je vhodné volit periferie tak, aby kinematický senzor, generátor časových pulsů a úložné zařízení používaly každé jinou sběrnici. ATmega328 (Arduino) i ATmega1281 (Wasmote) jich nabízí dostatek: UART (2× na ATmega1281), I<sup>2</sup>C, SPI, digitální a analogové piny.

Konflikt nejspíše vznikne na SPI sběrnici. Používají ji SD karty, 2,4GHz transceiver NRF24L01, některé modely kinematických senzorů i některé modely GPS čipů. Zejména použití SD karty prakticky vylučuje připojení jakéhokoliv dalšího zařízení, protože po celou dobu zápisu dat na kartu je nutné ji sběrnici rezervovat. Tyto zápisy mohou trvat až 80 ms a po tuto dobu by docházelo ke ztrátě dat z ostatních periférií.

Sběrnice UART neimplementuje žádné řízení přístupu, je možné k ní připojit jediné zařízení. S jejím obsazením je nutné počítat zejména při požadavku na zaznamenávání polohových zpráv z GPS a současného použití některého z WiFi čipů (případně jiných, v této práci netestovaných komunikačních modulů jako ZigBee, Bluetooth nebo GSM). Wasmote, používající čip ATmega1281 s dvojicí UART rozhraní zde umožňuje vyšší flexibilitu.

I<sup>2</sup>C rozhraní kromě senzorů často používají moduly hodin reálného času. V této práci jsem žádné netestoval, protože dostupné měly příliš nízkou přesnost 2 – 3 ppm<sup>45</sup>, ale při volnějším požadavcích na přesnost měření se jedná o zajímavou, snadno integrovatelnou synchronizační metodu.

### 5.4. Mikrokontroléry Arduino a Wasmote

Systém Arduino se při měřeních ukázal jako velmi levná a přitom dostačující platforma. Přes 16MHz procesor a 2kB operační paměť zvládal čist měřená data požadovanou 1kHz snímkovací frekvencí, snímat synchronizační linku a vše ukládat na paměťovou kartu.

Největší komplikací jeho použití je, že při použití běžně prodávané desky je třeba celou sestavu „poskládat“. Na rozdíl od systému Wasmote není na PCB integrována SD karta, akcelerometr ani patice na připojení dalších senzorů. Na druhou stranu díky otevřené licenci je možné vzít návrh jedné z vyráběných variant, rozšířit jej o požadované periferie a takto upravenou desku nechat vyrobit.

Deska Wasmote se i přes konfiguraci silnější procesor ukázala jako méně vhodná. Použití komplikuje zejména použití zastaralého firmware, ve kterém se hůře implementuje asynchronní a do jisté míry vícevláknová aplikace (současné hlídání synchronizačního signálu, snímání dat ze senzoru a ukládání na SD kartu).

V případné aplikaci s nižšími požadavky na přesnost a snímkovací frekvenci lze naopak využít kompaktnosti systému, kdy je na desce již integrován relativně přesný akcelerometr a slot na SD kartu. Na rozdíl od Arduina je zde také komplexněji řešený provoz na baterie: modul má dedikovaný konektor, ze kterého je možné baterie i přímo dobíjet.

<sup>45</sup> MAXIM INTEGRATED. DS3231: Extremely Accurate I2C-Integrated RTC/TCXO/Crystal [online]. Rev. 10. 2015, 20 s. [cit. 2015-05-25]. Dostupné také z: <http://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

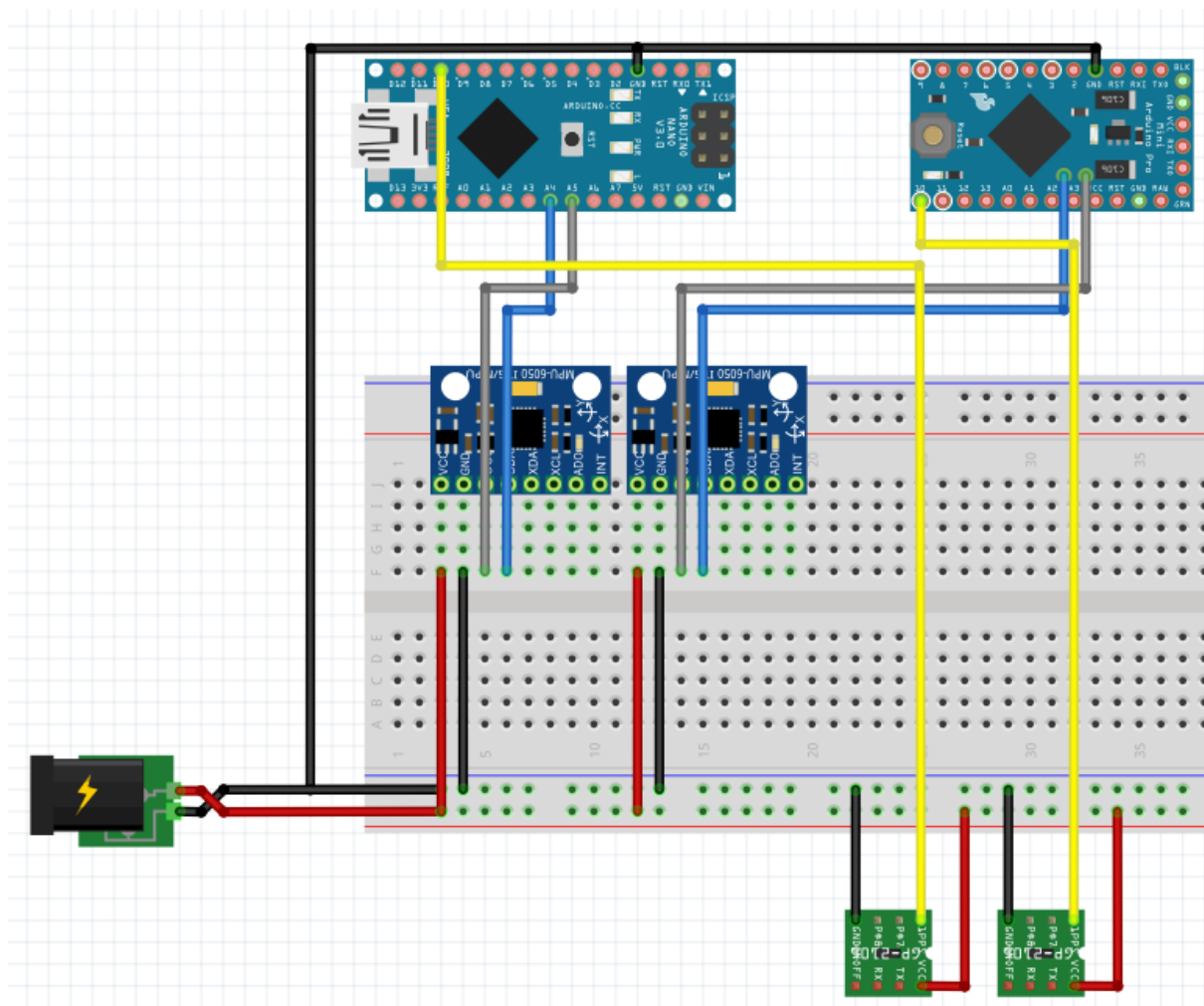
### 5.5. Synchronizované měření dvěma mikrokontroléry

Cílem měření bylo vyhodnotit, zda teoreticky nejvýhodnější metoda synchronizace pomocí PPS signálu GPS přijímače bude fungovat i v podmínkách reálného měření. Provedl jsem tedy experiment, jehož cílem bylo zjistit maximální časovou odchylku dvou mikrokontrolérů v průběhu půlhodinového měření.

Měřenou veličinou bylo zrychlení měřené dvěma akcelerometry MPU-6050, které byly připojeny do stejného nepájivého kontaktního pole. S kontaktním polem jsem pak manipuloval tak, aby oba akcelerometry měřily stejné impulsy – poklepání, krátký volný pád apod. Ideálním výstupem tak budou dvě překrývající se časové řady.

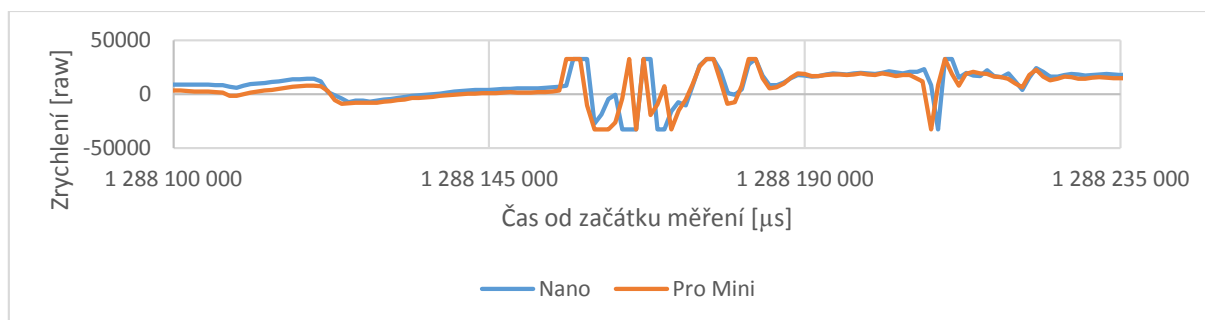
Hodnoty z akcelerometrů sbíraly mikrokontroléry Arduino Nano a Pro Mini. Ty data ukládaly do dvou 64B bufferů. Po zaplnění prvního bufferu se tento odeslal a jako zapisovací se začal používat druhý, po jeho zaplnění se role opět obrátily. Data byly v tomto experimentu odesílány po UART sběrnici do PC (cílem měření bylo ověřit výkon synchronizace, nikoliv ukládání dat).

Oba mikrokontroléry byly synchronizovány vteřinovým signálem z PPS pinu GPS přijímačů u-blox MAX-7C (Arduino Nano) a Telit JN3 (Pro Mini). Zapojení je ilustrováno na následujícím obrázku.



Obrázek 23 – Zapojení synchronizačního experimentu. Není zakreslené USB připojení a napájení Arduino Nano (vlevo) a UART-to-USB připojení a napájení Arduino Pro Mini (vpravo).

Zrychlení bylo sbíráno z osy Z (kolmé na podložku) v 1kHz snímkovací frekvenci, 2B rozlišení a rozsahu  $\pm 2$  g. Ke každému měření byly přiřazeny 2 B metadat: časová značka (počet mikrosekund od spuštění s přetečením každých 10 ms, 6 b) a hodnota vstupu digitálního pinu 10 (1 b). První bit prvního bajtu metadat zůstal nevyužitý a byl vždy 0.



Obrázek 24 – Naměřené zrychlení obou akcelerometrů v čase mezi 21:28,100 a 21:28,235 (zobrazený úsek 135 ms). Na ose Y je bitová hodnota zrychlení (32 768 = 2 g). Nižší hodnoty zrychlení akcelerometru Pro Mini jsou způsobeny mírně vyšším úhlem jeho uchycení na podložku.

Z posunu extrémů je patrná vzájemná odchylka 1 ms. Ta je pravděpodobně způsobena metodou, kdy se nezapisuje čas detekce náběžné hrany, ale pouze stav PPS pinu při každém měření. Pokud se tak dvě téměř současná měření dostanou na opačné strany PPS hrany, dojde k chybně vypočtenému zpoždění 1 ms.

I přes tuto chybu měření s dostatečnou rezervou splňují požadavek na přesnost  $\pm 2$  ms.

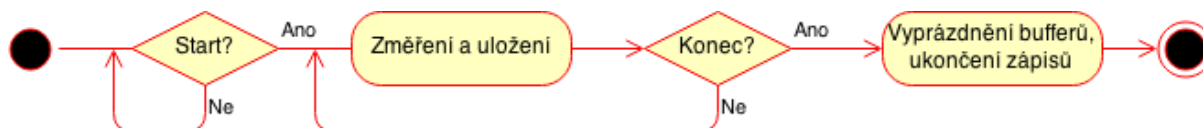
V případě požadavku na vyšší přesnost je možné změnit datový formát tak, aby se zapisoval mikrosekundový čas detekce hrany na PPS pinu. Jeden z metadatových bitů by pak indikoval, zda se jedná o záznam hodnoty zrychlení nebo času.

Použitý zdrojový kód na obou Arduinech je v souboru <sources/arduino-accelerometer-sync.ino>. Naměřená binární data jsou v souborech <data/gps-accel-data1.bin> a <data/gps-accel-data2.bin>. Skript, který data synchronizoval podle přijatých pulsů a převedl je do formátu TSV je v <sources/arduino-data-syncer.ls>.



## 6. Popis architektury softwarového řešení

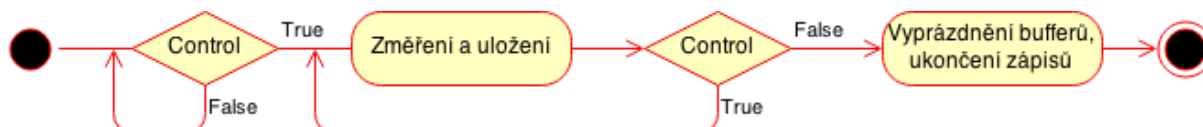
Na mikrokontrolérech i na systému cRIO je běh programu v prvním přiblížení identický: po spuštění čeká na pokyn k začátku měření, následně se spustí měřicí smyčka, která běží, dokud není externím povelům zastavena. Po zastavení dojde k ukončení zápisů a uvolnění paměti. Následuje ukončení běhu programu, případně návrat na začátek.



Obrázek 25 – Diagram obecného běhu programu.

Zdrojový kód měřicí a ukládací smyčky je již v této práci použitý na několika místech. Samotné měření z akcelerometru například v části Libelium Wasp mote: INTEGROVANÉ MODULY, ukládání dat je popsáno u každé testované metody. Celá smyčka včetně bufferingu byla implementována pro měření synchronizace v minulé kapitole 5.5. SYNCHRONIZOVANÉ MĚŘENÍ DVĚMA MIKROKONTROLÉRY. Pro systém CompactRIO je postup detailně popsán na webových stránkách NI <sup>46</sup>.

Dosud neprobírané byly právě metody ovládání běhu měření. Z pohledu uživatele by bylo praktické, aby se všechny připojené měřicí jednotky – CompactRIO i mikrokontroléry – ovládaly z jediného místa a bezdrátově. Vzhledem k požadavkům stačí přenést jediný *control* bit, který uložíme do lokální proměnné. Běh programu pak bude následující:



Obrázek 26 – Diagram běhu programu při použití jediné jednobitové řídicí proměnné.

Systém CompactRIO umožňuje nastavit proměnné čitelné a/nebo zapisovatelné přes HTTP protokol <sup>47</sup>. Endpoint proměnné je pak ve formátu `http://crio.local/namespace/variable/`. K nastavování, čtení, přepisování a mazání hodnot proměnných se používají standardní HTTP metody POST, GET, PUT a DELETE. Alternativně je možné vše obsloužit GET metodou a akci volit v URL, například `http://crio.local/namespace/variable/update?variable=newValue` <sup>48</sup>.

Při použití http hlavičky `Access-Control-Allow-Origin: *` <sup>34</sup> pak je možné cRIO ovládat aplikací běžící v libovolném webovém prohlížeči, včetně prohlížečů v mobilních telefonech. Jediným omezením je, aby byl PC či mobilní telefon připojený do stejné sítě, jako cRIO. V případě potřeby (a s příslušným řízením přístupu) touto sítí může být Internet.

Stejný způsob ovládání je možné použít i na mikrokontrolérech pomocí WiFi modulů ESP-8266 nebo WiFly RN171. V takovém případě je sice nutné napsat vlastní implementaci HTTP serveru, při použití GET-ové (non-REST) varianty přístupu k proměnným se však jedná o relativně jednoduchý algoritmus. Podrobněji jej rozebírám v následující kapitole 7. ALGORITMIZACE VYBRANÝCH PROBLÉMŮ.

Při důrazu na jednoduchost řešení, případně při požadavku na nízkou spotřebu je možné jako ovládání použít synchronizační signál. Měření se spustí při detekci první hrany na synchronizačním pinu

<sup>46</sup> Logging Data to Disk: Getting Started with CompactRIO. National Instruments: Tutorials [online]. 2011 [cit. 2015-05-27]. Dostupné z: <http://www.ni.com/tutorial/11198/en/>

<sup>47</sup> Building and Accessing a LabVIEW Web Service Application. National Instruments: Tutorials [online]. 2010 [cit. 2015-05-27]. Dostupné z: [http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/build\\_web\\_service/](http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/build_web_service/)

<sup>48</sup> LabVIEW Web Services - The RESTful CRUD. Chris Larson [online]. 2014 [cit. 2015-05-27]. Dostupné z: <http://chrislarson.me/blog/labview-web-services-restful-crud.html>



a ukončí se po dosažení zvoleného timeoutu od detekce poslední hrany. Timeout slouží nejen k překonání období mezi synchronizačními pulsy (1 s při použití GPS PPS pinu), ale též v případě momentálního výpadku signálu.

Nevýhodou takového řešení je nutnost před a po každém měření fyzicky připojit a odpojit synchronizační pin. Navíc při dlouhodobějším výpadku synchronizačního signálu dojde k vypnutí měření a tím úplné ztrátě měřených dat, na rozdíl od předchozí metody, kdy dojde jen k degradaci časové přesnosti.

## 7. Algoritmizace vybraných problémů

V této kapitole představím dva pomocné programy, které jsem pro účely této práce vytvořil, ale v žádném představeném měření přímo nepoužil.

Prvním programem je implementace jednoduchého HTTP 1.0 serveru pro WiFi modul ESP-8266. v prostředí Arduino. Umožňuje ovládání Arduina přes libovolný webový prohlížeč a je přes něj možné stáhnout libovolný soubor umístěný na připojené paměťové kartě.

Druhý skript řeší synchronizaci RTC hodin Linuxového systému podle GPS PPS signálu. Je implementovaný v jazyce Python a primárně určený na single-board PC Raspberry Pi 2, je však možné jej portovat na libovolný systém s GPIO piny. Dosahovaná přesnost synchronizace je  $\pm 0,1$  ms a je tak určen primárně jako zdroj absolutního času pro NTP server.

### 7.1. ESP-8266 Arduino HTTP server

HTTP server jsem psal s cílem maximální jednoduchosti a nenáročnosti na systémové prostředky. Jeho primárním určením je dálkové ovládání běhu měření (zapnutí a vypnutí) kompatibilní se systémy CompactRIO, v průběhu měření by měl spotřebovávat co nejméně procesorového času.

Server implementuje základní nastavení modulu ESP-8266. Při spuštění skriptu jej resetuje (zapsáním logické 0 na nastavený reset pin) a počká, až se modul zapne. Poté zapne multiplexování připojení (tedy že modul může udržovat až 4 aktivní spojení, což je vyžadováno pro funkci TCP serveru) a inicializuje server na portu 80.

Řešení skutečného paralelního přístupu však není pro danou aplikaci nutné, a tak je multiplexování připojení softwarově potlačeno. Při pokusu o připojení druhého a dalšího spojení je nové spojení okamžitě uzavřeno. To je nutné zejména pro přímý přístup prohlížečem na webovou stránku, protože většina prohlížečů posílá paralelně s požadavkem na danou URL i požadavek na ikonu serveru /favicon.ico.

Kvůli úspoře výkonu server neparsuje celou hlavičku, ale pouze identifikuje požadovanou URL – začíná po mezeře za klíčovým slovem GET a končí další mezerou. Za ní následuje verze HTTP protokolu, která je programem zahozena, stejně jako všechny ostatní hlavičky.

Server podporuje následující URL požadavky:

- / HTML výpis všech souborů na paměťové kartě, uspořádaný v struktuře `ul > li > a`<sup>49</sup>. Odkaz obsahuje adresu, ze které je soubor možný v prohlížeči stáhnout.
- /f/{název souboru} Adresa stažení libovolného souboru z SD karty. Je nutné počítat s omezenou šířkou pásma a z toho plynoucí dobou stahování.
- /status Příkaz vracející vždy stavový kód HTTP 200 OK. Slouží k detekci přítomnosti a provozuschopnosti zařízení.
- /webservice/control/ Vrací aktuální stav řídicí proměnné control (její význam popisují v předchozí kapitole 6. POPIS ARCHITEKTURY SOFTWAREVÉHO ŘEŠENÍ).
- /webservice/control/update?control={1|0} Nastavuje řídicí proměnnou na novou hodnotu.

Na ostatní požadavky vrací stavový kód nenalezeného objektu HTTP 404.

Implementace serveru je umístěná v přílohách, v souboru [sources/esp8266.ino](https://github.com/JanKucera/esp8266.ino).

### 7.2. GPS PPS synchronizace v prostředí Linuxu

V prostředí operačního systému Linux existuje řada utilit, které synchronizují čas k GPS času. Vesměs jej však nastavují pouze podle NMEA zpráv přijatých po sériové konzoli. Měření ale ukazují, že taková metoda je zatížena větší chybou, než synchronizace z NTP serverů na stabilním internetovém

---

<sup>49</sup> WORLD WIDE WEB CONSORTIUM. HTML5: A vocabulary and associated APIs for HTML and XHTML [online]. 2014 [cit. 2015-05-28]. Dostupné také z: <http://www.w3.org/TR/html5/>

připojení (2.3. DOBA POTŘEBNÁ PRO VÝPOČET POZICE a 4.3.4. MOŽNOSTI SYNCHRONIZACE, NTP, VZDÁLENÝ SERVER, DRÁTOVÉ PŘIPOJENÍ).

Z toho důvodu jsem napsal skript, který jako synchronizační signál využívá PPS pin GPS přijímače. I při jeho použití ale byla detekována chyba vnitřních hodin přibližně 8 ms, způsobená dobou vykonání systémového volání k nastavení času. V druhé fázi se tedy skript snaží iterativně eliminovat i tuto chybu.

Bohužel chyba způsobená dobou nastavování času není konstantní a s každým voláním variuje v řádu stovek mikrosekund. Skript se tedy snaží nastavovat čas tak dlouho, dokud se mu to nepovede s dostačující přesností. Z několika měření lze konstatovat, že je možné dosáhnout přesnosti  $\pm 100 \mu\text{s}$  do 10 sekund od spuštění ( $\pm 100 \mu\text{s}$  odpovídá hodnotě 99,9 % přesnosti NTP synchronizace po lokálním Ethernetu).

Implementace synchronizačního skriptu je v souboru [sources/rpi-rtc-pps-sync.py](#).

## 8. Závěr

V této práci jsem měl cíl zjistit, zda je pro měření kinematických veličin možné využít síť levných a málo výkonných mikrokontrolérů, synchronizovaných v čase. Požadovaná snímkovací frekvence byla 875 Hz a požadovaná přesnost synchronizace  $\pm 2$  ms.

Testované mikrokontroléry byly Atmel ATmega328 na deskách Arduino Nano a Pro Micro a Atmel ATmega1281 na desce Libelium Waspote. Jimi naměřené hodnoty pak musely být kompatibilní s hodnotami naměřenými systémem National Instruments CompactRIO.

Jako nejvýznamnější úzká hrdla mikrokontrolérů jsem identifikoval přesnost vnitřních hodin, přesnost čtení synchronizačních pulsů a propustnost zařízení k ukládání, případně přenosu naměřených dat. Vybral jsem tedy řadu různých způsobů generování i čtení hodinových signálů a metod, jak data dostat na paměťovou kartu nebo je přenést na další zařízení.

Na základě změřených jejich výkonu a přesnosti jsem poté vyloučil ty, které nesplňují výše zmíněné zadání. U vyhovujících metod jsem posoudil náročnost jejich implementace – hardwarově (požadavky na napájení či umístění antény), softwarově (složitost obslužného software) i finančně.

Za optimální sestavu považuji mikrokontrolér ATmega328, synchronizovaný hodinovým signálem z GPS přijímače a ukládající data na dostatečně rychlou SD kartu. Za méně než 1 000 Kč je možné sestavit měřicí zařízení, které sbírá data z libovolného I<sup>2</sup>C senzoru, časově je určuje s přesností  $\pm 30$   $\mu$ s a spolehlivě ukládá rychlostí 20 kBps.

Nejedná se o nejlevnější řešení, cenu zejména GPS přijímače ale vyvažuje jeho implementační jednoduchost, vysoká přesnost a použití univerzálního času UTC. Díky tomu je možné data přidat k libovolnému stejně časovanému datasetu.

Jednoduché mikrokontroléry rozhodně nemohou nahradit systémy jako CompactRIO, mají méně sběrnic s menší propustností, jsou obtížněji rekonfigurovatelné a nabízejí mnohem méně pohodlné ovládací prostředí.

Při měření se však někdy vyskytnou zájmová místa, která jsou vzdálená od ostatních měřených bodů a kde mnohdy postačuje jediný senzor. Po zvážení náročnosti vedení napájecích a datových kabelů se taková místa často obětují ve prospěch jednodušší montáže a demontáže experimentu. A právě na vykrývání takových dosud slepých bodů jsou kompaktní, baterií napájené mikrokontroléry zajímavou a dostatečně výkonnou volbou.

## 9. Rejstříky

### 9.1. Seznam použitých zkratk

- ADC, 13, 23, 26, 27, 44  
Analog-Digital Converter, převodník signál-  
číslo
- API, 32, 33, 34, 38, 43  
Application Programming Interface,  
programovací rozhraní aplikace
- Bd, 11, 17  
Baud, přenosová rychlost
- cRIO, 4, 8, 13, 18, 35, 36, 37, 38, 39, 42, 43, 44,  
45, 49  
National Instruments CompactRIO
- CS, 32  
Chip Select (pin na SPI sběrnici)
- CTS, 34  
Clea To Send (pin na RS-232 sběrnici)
- DGPS, 7  
Differential GPS
- DIO, 25, 35, 36, 44, 45  
Digital Input Output, Digitální vstupně-  
výstupní pin
- DMP, 13  
Digital Motion Processor, procesor  
pohybových dat
- EEPROM, 17, 18, 31, 36  
Electrically Erasable Programmable Read-  
Only Memory, elektricky mazatelná  
nevolatilní paměť
- EGNOS, 9  
European Geostationary Navigation  
Overlay Service, evropský DGPS systém
- EVT, 6  
Event pin, signál externího přerušení
- FIFO, 14  
First In First Out, první dovnitř, první ven
- FPGA, 13, 16, 35, 36  
Field Programmable Gate Array,  
programovatelné hradlové pole
- GLONASS, 7  
Globalnaja navigacionnaja sputnikovaja  
sistěma, ruský GNSS
- GNSS, 6  
Global Navigation Satellite System, globální  
družicový navigační systém
- GPS, 5, 6, 7, 8, 11, 12, 23, 24, 25, 27, 31, 32, 37,  
38, 39, 40, 41, 44, 45, 46, 47, 50, 51, 52, 53  
Global Positioning System, americký GNSS
- I/O, 17, 20, 22, 26, 31, 35, 36, 45  
Input-Output, Vstupně-výstupní
- I<sup>2</sup>C, 7, 8, 13, 14, 17, 23, 25, 31, 32, 46  
Inter-Integrated Circuit, dvou vodičová  
sériová sběrnice
- MEMS, 13, 32  
Micro-Electro-Mechanical Systems
- MTU, 33  
Maximum transmission unit, maximální  
přenosová jednotka v jednom datagramu
- NI, 35, 36, 37, 44, 45, 49  
National Instruments
- NMEA, 6  
National Marine Electronics Association
- NTP, 37, 38, 39, 40, 41, 44, 45, 46, 51, 52  
Network Time Protocol, síťový časový  
protokol
- PCB, 46  
Printed Circuit Board, deska plošných spojů
- ppm, 25  
Parts Per Million, dílů na jeden milion
- PPS, 6, 7, 8, 9, 10, 12, 23, 24, 25, 27, 37, 38, 45,  
47, 50, 51, 52  
Pulse Per Second, hodinový signál
- RPi, 8  
Raspberry Pi
- RX, 7  
Receiver, přijímací vodič, opak TX
- SBAS, 7, 9  
Satellite-based augmentation systems,  
satelitní DGPS systémy
- SCL, 7  
Serial Clock, časový vodič I<sup>2</sup>C sběrnice
- SDA, 7  
Serial Data, datový vodič I<sup>2</sup>C sběrnice
- SMD, 13, 21  
Surface Mount Device, součástky pro  
povrchovou montáž
- SoC, 7  
System on a Chip, na jednom čipu  
integrované CPU, GPU, paměť a další  
komponenty

SPI, 4, 7, 8, 13, 17, 18, 21, 22, 23, 28, 29, 31, 32, 35, 36  
Serial Peripheral Interface, čtyřvodičová sériová sběrnice  
TCP, 20, 21, 29, 33, 37, 51  
Transmission Control Protocol, protokol transportní vrstvy IP sítě  
TX, 7  
Transmitter, vysílací vodič, opak RX  
UART, 7, 8, 10, 11, 17, 20, 22, 23, 25, 29, 31, 32, 33, 43, 44, 45, 46, 47  
Universal Asynchronous Receiver and Transmitter, dvouvodičová duplexní sériová sběrnice

UBX, 7  
u-blox protokol  
UDP, 23, 33  
User Datagram Protocol, protokol transportní vrstvy IP sítě  
UTC, 5, 37  
Universal Coordinated Time, koordinovaný světový čas  
WPA2, 20, 33  
Wi-Fi Protected Access, šifrovací protokol bezdrátových sítí