



ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

Bakalářská práce

Řízení inteligentní domácnosti s One-Wire sběrnici s parazitním napájením

aneb

Globální synchronizace HW prvků připojených do sítě internet

Škola	ČVUT Praha
Studijní obor:	FEL, KME
Jméno autora:	Michael Voříšek
Vedoucí Práce:	Ing. Vladimír Janíček, Ph.D.
Školní rok:	2016 / 2017

Prohlášení autora

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Praze, dne 26. května 2017

.....

podpis

Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce, Ing. Vladimíru Janíčkoví, Ph.D., který mi pomohl s věcnou stránkou a formou práce.

Dále také děkuji všem, kteří se zúčastnili testování, měli věcné připomínky, nápady či jiným způsobem pomohli zlepšit synchronizační systém:

Ing. František Vlk

Ing. Milan Šulc

Roland Zůna

Ing. Karel Dohnal

John Vanhara

Božena Voříšková

Šimon Vejlupek

Václav Vávra

Mgr. Henrich Paľo

Martin Štoll

Stanislava Nováková

Petr Kinderman

Monika Janoščíková

Michal Škrabal

Mgr. Zuzana Štočková

Rober Neil

Jan Votroubek alias „Q“

Mgr. Libor Háček

Жорик Вартанов

a další

Zadání práce



NÁVRH ZADÁNÍ BAKALÁŘSKÉ PRÁCE ČVUT FEL, katedra mikroelektroniky

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení a jméno studenta: **Voříšek Michael**
Studijní program: **Komunikace, multimédia a elektronika** Studijní obor: **Aplikovaná elektronika**
Email: **vorismi3@fel.cvut.cz** Telefon: **+420 606 560 790**

II. ÚDAJE K ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název tématu: **Řízení inteligentní domácnosti s One-Wire sběrnici s parazitním napájením**

Název tématu anglicky: **Cloud based smart home control with One-Wire bus with parasitic supply**

Práce je zadána dle požadavků průmyslu nebo je vedena někým z průmyslu (Vedoucí, konzultant, oponent):

Ano (Ne)

Pokyny pro vypracování:

- 1) Prostudujte problematiku řízení inteligentní domácnosti.
- 2) Navrhněte koncept systému využívající cloudových služeb, skládající se z centrálního sběrného bodu (hub) a měřících senzorických jednotek (nodes).
- 3) Pro komunikaci mezi jednotkou a senzory využijte převodník ostatních síťových standardů na One-Wire sběrnici.
- 4) Návrh realizujte a ověřte parametry.

Seznam doporučené literatury:

- 1) The Internet of Things (The MIT Press Essential Knowledge series), Greengard S., ISBN-13: 978-0262527736
- 2) Internet of Things: Principles and Paradigms, Buyya R., Dastjerdi A.V., ISBN-13: 978-0128053959
- 3) Site Reliability Engineering: How Google Runs Production Systems, Beyer B., ISBN-13: 978-1491929124

III. POTVRZENÍ ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vedoucí práce

Jméno a příjmení: Ing. Vladimír Janiček Ph.D.

Email: janicev@fel.cvut.cz

Pracoviště: katedra mikroelektroniky

Telefon:

Garant za katedru (*)

(*) Uvádí se pouze v těch případech, kdy příslušná oborová katedra tento údaj požaduje. Více informací na <http://cyber.felk.cvut.cz/study/student-projects/guarantor>.

Datum zadání práce:

Podpis vedoucího práce

Podpis garanta oboru (*)

Podpis vedoucího katedry

(*) Uvádí se pouze v těch případech, kdy příslušná oborová katedra tento údaj požaduje

Anotace v češtině

Cílem mé práce „HEULO HS“ bylo vytvořit systém, který by zajistil spolehlivou real-time synchronizaci reálných věcí s internetem. Hlavní konkurenční výhodou je plná kompatibilita s NATem, real-time odezva a předpokládaná prodejní cena.

Klíčová slova v češtině

- Vzdálené ovládání
- Real-time synchronizace
- Cloud
- Elektronika
- Řízení a regulace

Annotation in English

The goal of the submitted thesis „HEULO HS“ was to create system which could guarantee real-time synchronization real things with the internet. The main advantage is full compatibility with NAT, real-time responsivity and expected retail price.

Keywords in English

- Remote control
- Real-time synchronization
- Cloud
- Electronics
- Control and regulation

Obsah práce

1. Úvod	8
1.1 Motivace	9
1.2 Rozdělení projektu.....	9
2. HEULO HW	11
2.1 Úvod	11
2.2 Hlavní jednotka.....	12
2.3 Teplotní senzory	14
2.4 Senzor vlhkosti	14
2.5 Senzor tlaku	17
2.6 CO ₂ senzor.....	17
2.7 One-Wire, řešení, jak propojit senzory jedním drátem.....	17
2.8 One-Wire, vlastní implementace a fyzikální limity	18
2.9 Sumarizace	19
3. HEULO SW – klient.....	21
3.1 Úvod	21
3.2 Real-time a stabilita	21
3.3 Demo aplikace.....	22
3.4 Sumarizace	23
4. HEULO SW – server.....	24
4.1 Úvod	24
4.2 Požadavky	25
4.3 Příprava serveru a optimální ukládání databázových dat	25
4.4 Optimální plánovač diskových úloh.....	31
4.5 Sumarizace	32
5. Závěr.....	33
Systémové požadavky - klient	34
Doporučené systémové požadavky - server	34
Seznam vývojového software	35
Seznam použitých zkratk	36
Přílohy	37
Seznam odborných zdrojů.....	38

1. Úvod

Cílem mé práce „HEULO HS“ je systém, který zajistí spolehlivou real-time synchronizaci reálných věcí s internetem. Hlavní konkurenční výhodou je plná kompatibilita s NATem, real-time odezva a předpokládaná prodejní cena.

Práce popisuje zařízení, které umožňuje v reálném čase ovládat cokoli, kdekoli, resp. v místech s dostupnou konektivitou.

O podobné řešení se na trhu snaží více firem, pomíjí však fakt, že 99% potenciálních kupců potřebuje zařízení zapojit za NAT. Nejznámější firmy z oboru se přitom snaží získat ten zbytek - 1%! Tento fakt vyplývá z toho, že pro podporu NATu je potřeba „nějakého prostředníka“ dostupného z internetu, tedy nějaký server/cloud hostovaný v datacentru, ideálně s garantovaným SLA.

Server samozřejmě vyžaduje trvalou péči – administrativní, ale i finanční. To je, dle mého názoru, hlavním důvodem, proč se firmy na trhu snaží bojovat o to 1% zákazníků. Projekt prezentovaný v této práci je zaměřený přesně na opačnou klientelu, tedy na „zbylých“ zákazníků 99%, kteří potřebují ovládat zařízení za běžnými přípojkami bez veřejné IP či s IP veřejnou, avšak nestatickou.

V systému „HEULO HS“ požadavky klientů putují nejdříve na server a ty jsou následně přenášeny po předem vytvořených cestách – socketech – ze serveru do zařízení. Předem vytvořených proto, že server na zařízení opět nemá přímý přístup, jakákoliv komunikace musí být striktně inicializována na požadavek klientského zařízení.

Implementace výše popsaného je komplexní a náročná úloha. Běžné PC si řekne o cca 100W, nějaké „zelenější“ o třeba jen 30W, což pořád ale není při provozu 24x7 nezanedbatelné, při průměrné 50W spotřebě a ceně €0.15 (4.2 Kč) za kWh samotná spotřeba vychází na cca €65 (1800 Kč) za zařízení za jeden jediný rok. Můj projekt tvoří proto i hardwarové endpointy (koncové body), které umožňují garantovat nejvyšší dosažitelnou spolehlivost s optimalizovanou spotřebou zhruba 2.5 W, tedy řádově více jak 10x menší než příklad s běžným PC uvedený výše. Vezmeme-li jen rozdíl v ceně oproti konkurenci, systém se do dvou let zaplatí a následně již jen šetří.

1.1 Motivace

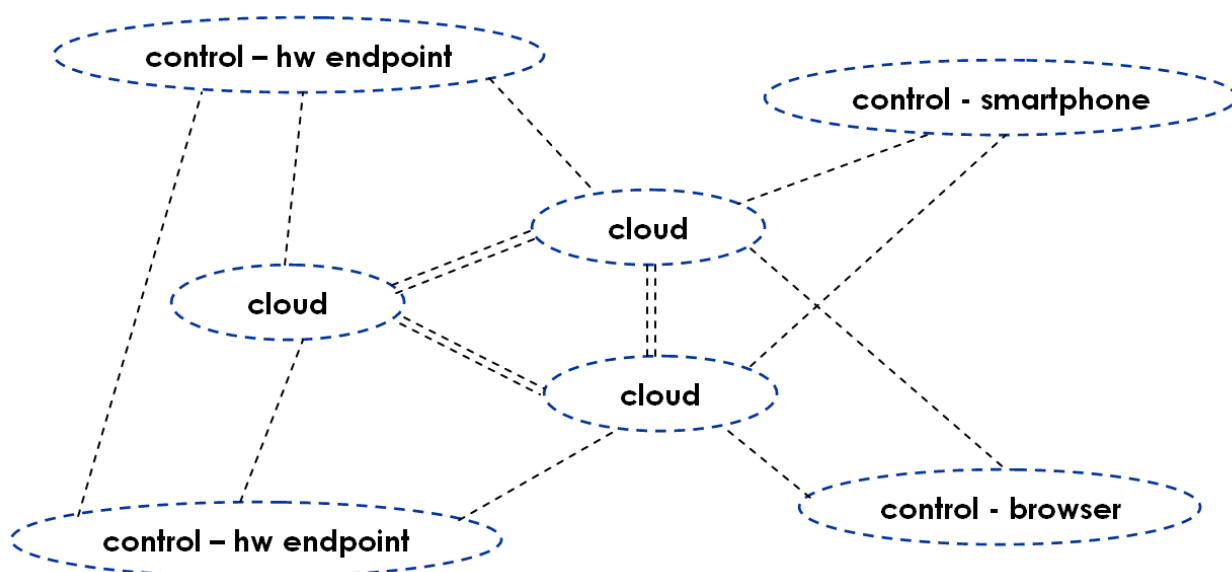
Podporu NATu, real-time synchronizaci, cenu pod €20 a jednoduchou integraci (zapojení + API rozhraní dostupné z Internetu) nikdo komerčně nenabízí. Proto jsem se rozhodl pro vlastní řešení.

1.2 Rozdělení projektu

Práce řeší ovládání od A to Z, mapa systému níže ukazuje 3 základní bloky systému (obr. 1) a jejich propojení.

- HW endpointy (vlastní hardware, který data snímá či na ně naopak reaguje)
- Cloud (serverová infrastruktura, kde si zákazník pronajímá možnost připojit svoje zařízení)
- Kontrolní rozhraní (webový interface či API)

Synchronizační software představuje jakýsi tmel mezi jednotlivými prvky, který zajišťuje jejich propojení s důrazem na UX a orientací na zákazníka.



Obr. 1 - schéma systému, P2P propojení mezi jednotlivými hw endpointy je v přípravě – aktuálně hw endpointy při výpadku konektivity přestanou reagovat, při přímém propojení by však mohly nezávisle fungovat se základními funkcemi (funkce termostat, funkce plné otevření rolet apod.)

Vzhledem k rozsahu práce jsou některé části v práci pouze nastíněny a rozsáhlejší podklady (schéma, software, data z měření, ...) pouze přiloženy. Vlastní HW je popsán v kapitole 2. (HEULO HW). Úzké hrdlo systému je cloud, v kapitole 4. (HEULO SW - server) jsou podrobněji vysvětleny nároky na serverový storage, včetně detailního měření různých typů disků, RAID řadičů,

vliv jejich nastavení a optimální metody připojení do virtuálních strojů. Zvlášť v menším nasazení je virtualizace z důvodu ekonomické výhodnosti a pohodlnější správy nezbytná.

2. HEULO HW

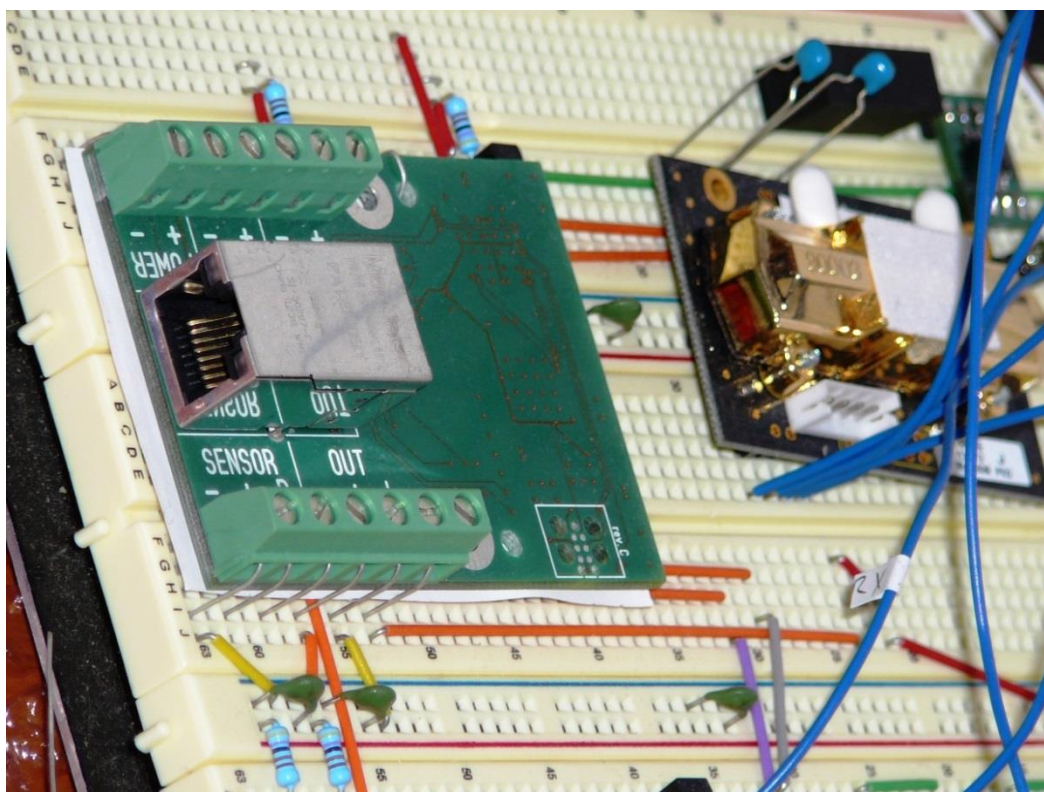
2.1 Úvod

Pro ovládání hardwaru je potřeba bezesporu senzorových/reakčních prvků, které jsou kompatibilní s řídicím systémem – v praxi se dnes vyrábějí převážně pouze systémy s lokálními sběrnici typu I2C, RS-485, CanBus apod. HEULO systém se oproti těmto systémům odlišuje už jen typem sběrnice a komunikační architekturou. Používá stále více se prosazující sběrnici ethernet připojenou do internetu. Namísto P2P architektury každý prvek komunikuje vždy přes nejbližší cloud node. Přímá P2P komunikace však může eliminovat výpadky při nedostupnosti konektivity do Internetu.

Vzhledem k náročnosti vývoje P2P řešení se může konektivita v následujících desetiletích stát 1. přípojkou, u které bude běžné i v domácnostech mít přípojku záložní.

HW prvky systému HEULO se dělí na main jednotky, které jsou připojené do ethernetu a fungují autonomně, a na periferní jednotky, což jsou senzory či jiné periferie připojené k main jednotkám. Teoreticky by periferie mohly být také zapojeny napřímo do ethernetu, ale ekonomicky výhodné to není, ethernet není zcela nejjednodušší na řízení a šifrování komunikace skrz Internet (prozatím) rozhodně procesor za \$1 nezvládne. Další náklady by byly spojené s konektivitou, extra zátěží serveru apod.

2.2 Hlavní jednotka



Obr. 2 - foto z vývoje hlavní jednotky, v pozadí senzor CO₂

Hlavní jednotka řeší přenos informace mezi vstupy/výstupy a cloudem. Důraz je kladen na spotřebu, dostupnost a near real-time (odezva v téměř reálném čase) odezvu.

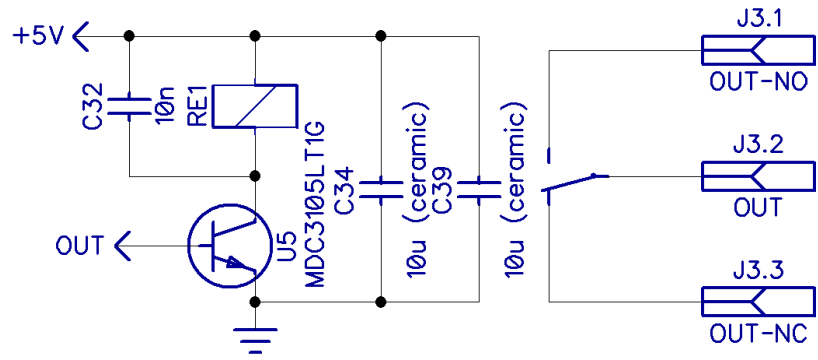
Spotřeba a dostupnost je zajištěna použitím dsPIC33 procesoru a na míru optimalizovaného návrhu HW a SW.

Near real-time odezva je dosažena pokročilými metodami komunikace, asynchronní komunikace a event driven (netestováním změn v cyklu, nýbrž asynchronním přerušením na základě akce) návrhem programu.

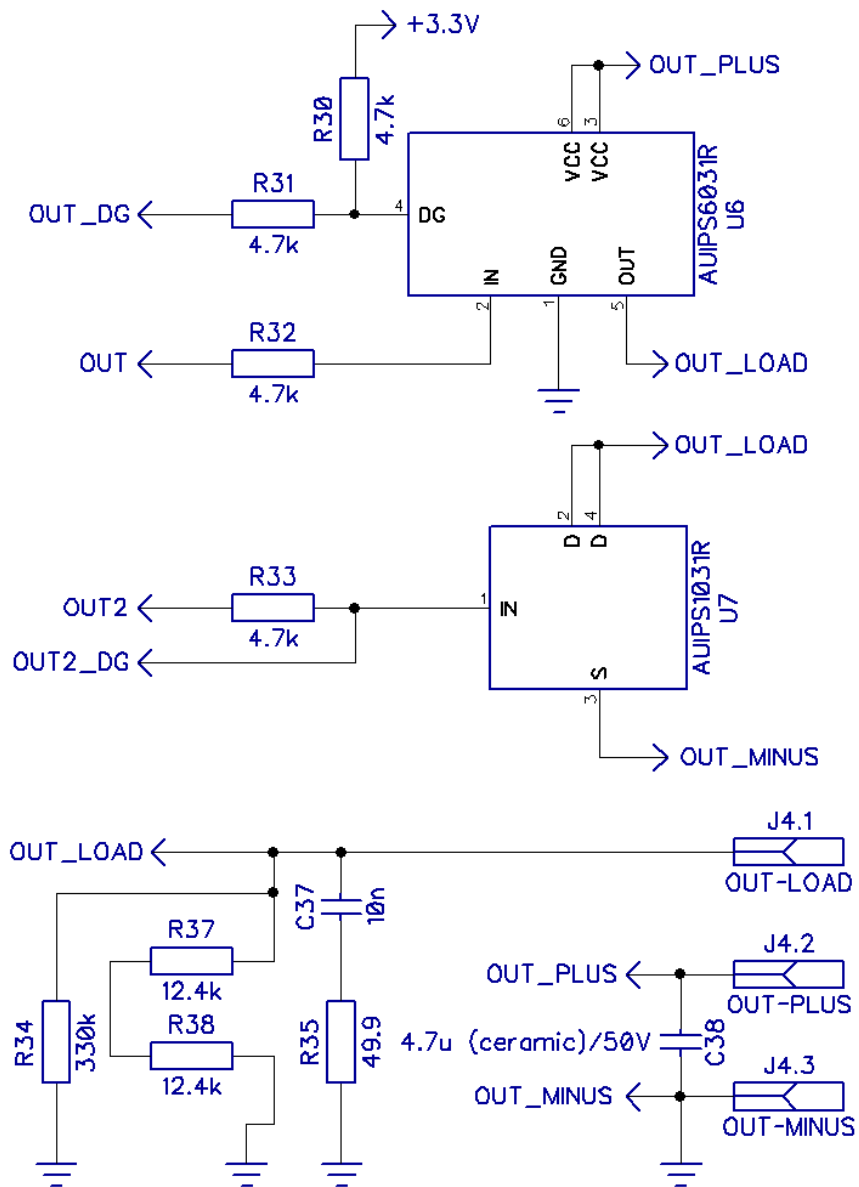
Z požadavku na dostupnost a díky cílení na integraci přes cloud byla zvolená co možná nejmenší velikost a pouze 2 digitální vstupy, 1 výstup a 1 One-Wire sběrnice ve funkci master.

Kompletní schéma je přiloženo v příloze I.

Vzhledem k běžnému požadavku spínat síťové (230 VAC) napětí a zároveň umožnit rychlé spínání (např. pro PWM) jsou výstupem práce dvě verze hlavní jednotky. Jedna s relé výstupem (230 VAC, 16A) (viz. obr. 2b) a druhá s push-pull spínačem na bázi MOSFET technologie (až 60V, 25A) (obr. 2c).

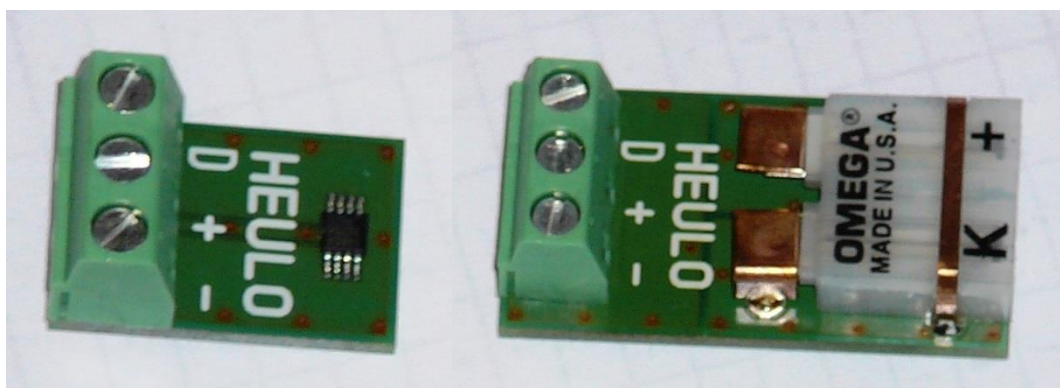


Obr. 2b – relé výstup



Obr. 2c – el. spínaný výstup

2.3 Teplotní senzory



Obr. 3 – release verze teplotního čidla a thermocouple převodníku

Tyto senzory se sestávají z integrovaného teploměru DS18B20 či přímého převodníku pro termočlánek přímo na One-Wire sběrnici.

2.4 Senzor vlhkosti



Obr. 4 – release verze vlhkostního čidla

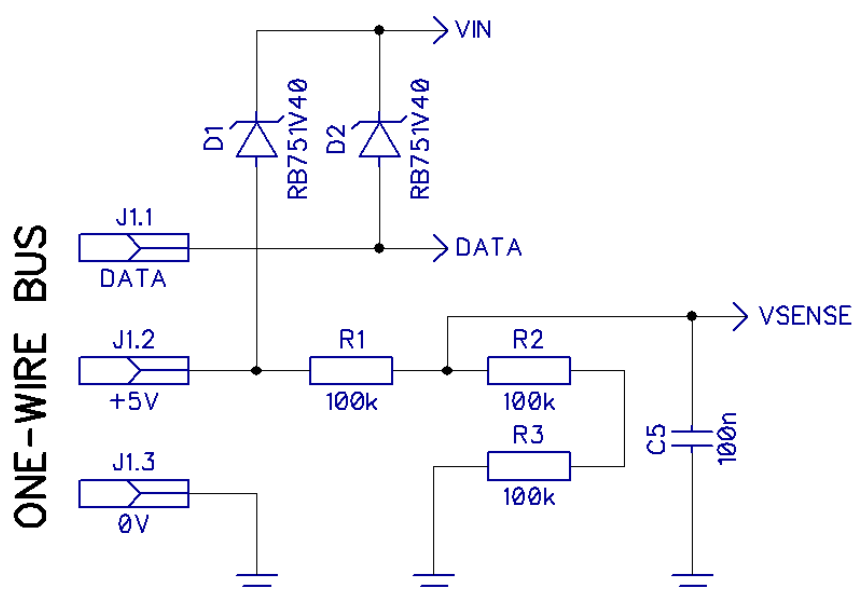
Podstatná část realizace. Na trhu neexistuje dostupný vlhkostní senzor s připojením skrz One-Wire.

V rámci této práce byl vyvinut vlastní sensorový modul (obr. 4) s procesorem na spodní straně realizující softwarový konvertor mezi sběrnici čidla (I2C) a komunikační sběrnici One-Wire.

Kritickým bodem je spotřeba (rozebírá kapitola 2.7) v případě parazitního napájení.

Kompletní schéma je přiloženo v příloze II.

Napájení je realizováno následujícím způsobem:



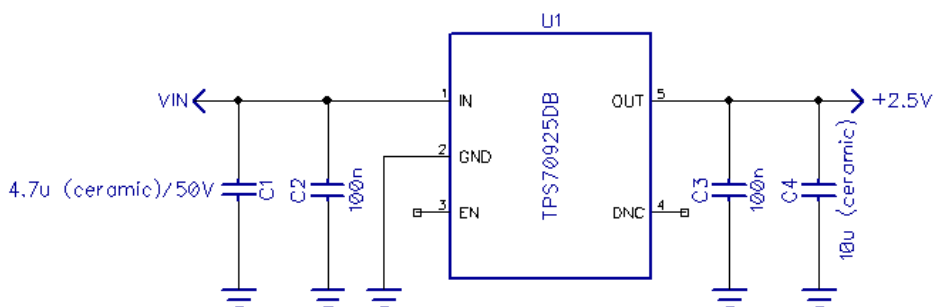
Obr. 4b – napájení

1. „Sběr energie“ (Obr. 4b) diody (schottkyho, požadavek na nejnižší úbytek) propustí proud, pokud je napětí na datovém vodiči (či napájením, je-li napájení přítomno) minus úbytek je nižší než napětí na kondenzátoru (viz. dále).

2. Akumulace energie a stabilizace (Obr. 4c) je další důležitá část. Energii pro chod procesoru a samotného senzoru potřebujeme dodávat konstantně.

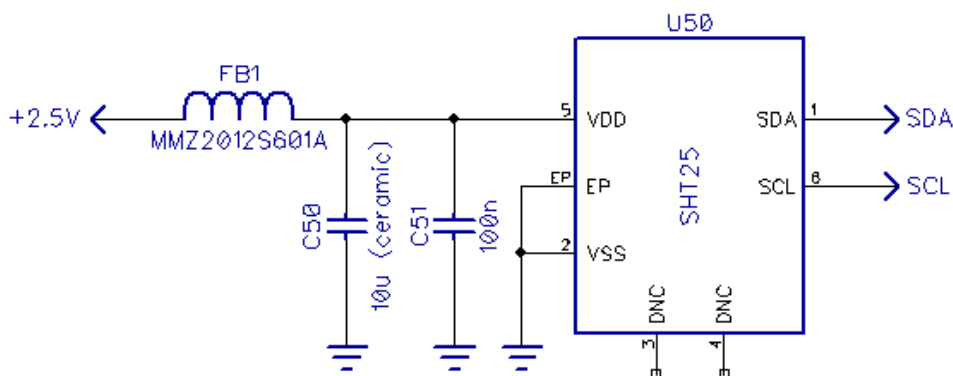
Akumulaci zajišťují kondenzátory (na vstupu vysokonapěťové, dále z ekonomických důvodů na menší napětí)

Stabilizátor musí nutně být s co možná nejnižším úbytkem (ultra LDO). Potřebujeme zajistit funkčnost stabilizace 2.5V při napájení 3.5V (tedy na vstupu stabilizátoru cca 3V z důvodu úbytku na diodách).



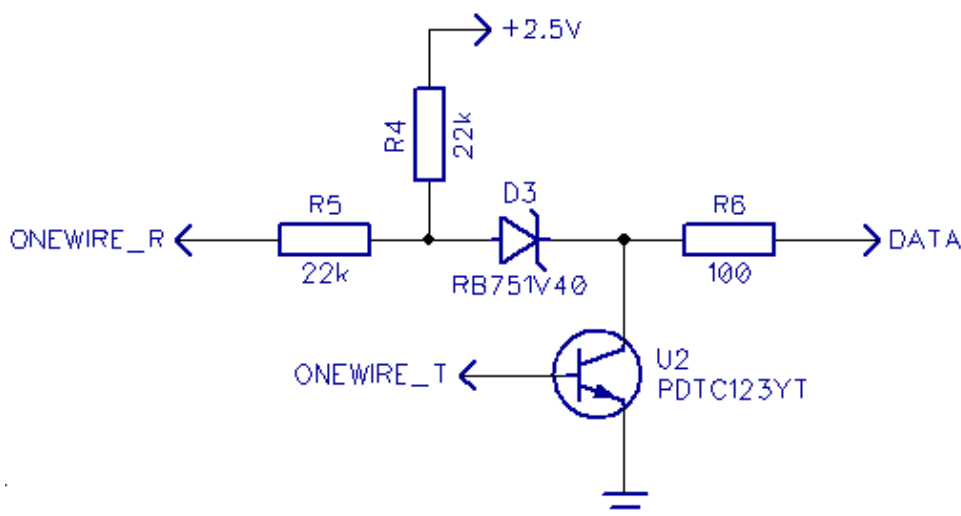
Obr. 4c – akumulace energie a napěťová stabilizace

Samotný senzor je následně napájen přes LP filtr (dolní propust) (Obr. 4d).



Obr. 4d – senzor a lokální filtrace napájení

One-Wire sběrnice je chráněné označení firmy Maxim-Integrated. Protokol včetně fyzické vrstvy je veřejně dostupný. Pro maximální kompatibilitu senzorů i s jinými systémy (např. Raspberry Pi a dostupného One-Wire driveru) bylo navrženo následující zapojení pro emulaci One-Wire slave zařízení (Obr. 4e).

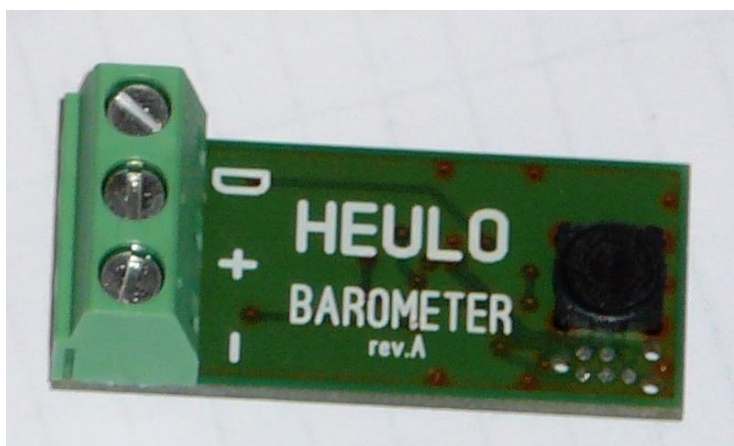


Obr. 4e – HW implementace One-Wire slave příjmu / vysílání

Shottkyho dioda je nutná pro robustní zajištění log. 0 v toleranci procesoru dané úbytkem na diodě, vedení a R6 (minimální, 11 mV, dané děličem R4 a R6).

Použitý senzor fy. Sensirion SHT25 vykazuje skvělé výsledky i pro ty nejnáročnější aplikace – skvělá přesnost, skvělá stabilita, a zde prezentovaný HW lze přímo masově vyrábět bez jakýchkoli dalších úprav.

2.5 Senzor tlaku



Obr. 5 – beta verze tlakového čidla s čidlem ST

Tlakoměrový modul se sestává ze stejného základu jako modul s vlhkoměrem.

Senzor funguje spolehlivě. Na měření předpovědi počasí plně vyhovuje. Šum však dosahuje cca 50 Pa a není tedy tímto senzorem možné přesně měřit např. výstup dronu nad zemí či nasazení v podobné aplikaci, kde je nutné měřit stabilně na 5 Pa (stačí relativně, ale opakovaně).

2.6 CO₂ senzor

Senzorový modul ve vývoji, viz. foto u hlavní jednotky (obr. 2). Testy probíhají se senzorem T6613 fy. GE, resp. Amphenol. Rozlišení v kombinaci se šumem je velmi dobré, přesnost $\pm 30\text{ppm}$ nebo 3% z rozsahu, cena pro masové rozšíření je však zřejmě vysoká – cca 2000 Kč. Uvedený senzor pracuje na principu NDIR – detekční komorou se svítí LED emitující infra záření. Část záření prochází přímo, část paralelně skrz referenční plyn – typicky dusík. Pro různou koncentraci CO₂ se potlačí odlišné různé vlnové délky a na základě toho senzor určí koncentraci. Na rozdíl od ostatních, převážně levnějších, senzorů se senzor sám kalibruje a v čase tedy tolik nedriftuje, jako senzory, které kalibraci nemají, nebo jsou kalibrovány jen při výrobě.

Zmiňovaný senzor T6613 má komunikační rozhraní UART (RS232) a pro připojení na sběrnici One-Wire je opět využit vlastní převodník.

2.7 One-Wire, řešení, jak propojit senzory jedním drátem

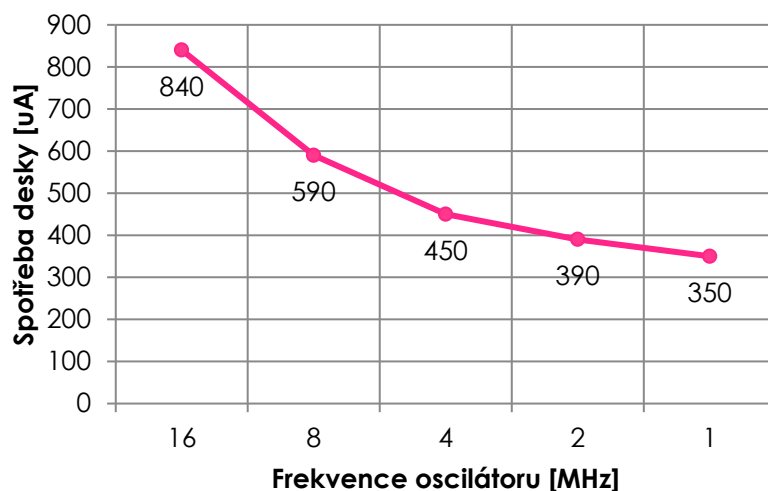
Cena za senzor výrazně ovlivňuje počet senzorů, které uživatel na m² nainstaluje, cílem je nabídnout kvalitní senzory za dostupnou cenu podstatně levněji než konkurence (podobně řídí fy. např. Elon Musk).

Každý senzor by mohl být připojený přímo přes ethernet, ale toto řešení by nebylo ekonomicky výhodné (spotřeba, cena portů ve switchi, strukturovaná kabeláž, nároky na server atp.), všechny senzory, umožňuje-li to spotřeba snímacího prvku, komunikují po jednom drátu a po tomto drátě jsou i napájené. V prezentované práci je potřeba k propojení senzorů pouze jakákoliv dvojlinka či třeba i kroucený pár z UTP kabelu.

Nespornou výhodou je velký dosah (2 km (ethernet 100m)) a nutnost pouze dvou vodičů (jeden datový vč. parazitního napájení – napájení přes datový vodič a druhý zem). Dva vodiče jsou nejen levnější, ale také podstatně jednodušší na zapojení. Zařízení jsou odolná proti přepólování, nebo proti přiložení vyššího napětí (až 12x).

2.8 One-Wire, vlastní implementace a fyzikální limity

Jelikož se po jednom vodiči přenáší i napájení, a to samozřejmě nemůže být přítomno po celou dobu komunikace, největším omezením je spotřeba na připojený senzor. Tato spotřeba definuje max. počet zařízení na sběrnici. Představená vlastní implementace se blíží fyzikálním limitům, viz. graf (obr. 6) ukazující spotřebu závislou na frekvenci uP. Pro představu, jedna LED dioda odebírá obvykle 15 mA.



Obr. 6 - Spotřeba jednoho senzorového modulu a vliv podtaktování procesoru pro převod na sběrnici One-Wire

Přínosem vlastní implementace je One-Wire slave protokolu je možnost na tuto sběrnici připojit cokoli, s nižší spotřebou a nemuset kupovat / licencovat extra čip – protokol je známý pouze proto, aby vývojář mohl implementovat One-Wire master. Při plném pochopení protokolu a dodržení časování však One-Wire slave bylo možné implementovat i bez přístupu k dokumentům, které slave dokumentují.

Podpora více zařízení, tedy adresování slave senzorů po jednom drátu, náročnost na extrémně nízkou spotřebu ještě více umocňuje. Prezentované řešení umožňuje napájet až 20 zařízení po jednom společném vodiči.

2.9 Sumarizace

Podařilo se splnit motivační požadavky. Hlavní přednosti systému oproti existujícím řešením a jejich řešení je následující:

Hlavní jednotka:

- Cenová dostupnost a velikost - počet IO omezen na nezbytné minimum, obvykle uživatel na jednom místě nepotřebuje 16 vstupů a 16 výstupů, za stejnou cenu je výhodnější použít několik separátních jednotek
- Snadné připojení senzorů prostřednictvím sběrnice One-Wire – senzory je možné připojit „dvojlinkou“ a to i několik senzorů najednou
- One-Wire master je optimalizován pro parazitní napájení slave senzorů – pull-up pro nabíjení slave senzorů energií je zesilování přímým (tvrdým) napájením v době, kdy nevysílají slave senzory
- NAT – hlavní jednotka se serverem vždy inicializuje komunikaci jako první a udržuje spojení pro server aktivní – server poté může jednotku po tomto spojení instantně oslovit
- Real-Time synchronizace – jednotka udržuje aktivní spojení se serverem a ten ji může instantně oslovit. Např. u topení odezva z důvodu samotné setrvačnosti důležitá není, avšak např. u ovládání světla uživatel okamžitou odezvu očekává.
- Snadná integrace – zařízení stačit zapojit, jakmile jednotka dostane IP od DHCP serveru navazuje spojení se serverem dostupného z Internetu, není potřeba žádná konfigurace

Senzorové jednotky

a) existuje senzor přímo na sběrnici One-Wire

- Je využito existující řešení dostupné na trhu

b) senzor na sběrnici One-Wire neexistuje

- Vyvinut vlastní převodník mezi One-Wire sběrnici a použitým čidlem (typ. I2C - zde vlhkoměr, tlakoměr - či UART (RS232) – zde CO2 senzor)
- Parazitní napájení – senzory lze napájet po společném datovém vodiči, dosaženo optimalizací celého zapojení senzorů vč. extrémních optimalizací v assembleru v softwarovém převodníku pro dodržení časování One-Wire komunikace dle specifikace a minimálního taktu procesoru, který zvyšuje spotřebu. Další optimalizace je automatické

snížení taktu v době nečinnosti. Snížením spotřeby je dosaženo vyššího možného počtu zařízení na sběrnici.

3. HEULO SW – klient

3.1 Úvod

Při vývoji byl kladen důraz na znovu použitelnost synchronizačních částí klientského kódu. Součástí práce je demo aplikace, která vytvořených synchronizačních knihoven využívá a potvrzuje funkčnost jednotlivých prvků systému.

3.2 Real-time a stabilita

Pro zajištění real-time, resp. near real-time odezev systém využívá HTTP long-pooling technologie (realizace event technologie typu Push skrz http protokol s podporou proxy).

Oproti konkurenci systém využívá více spojení pro zajištění maximální plynulosti a to rovnou přes více front serverů.

Občas se spojení rozpojí, a potom je nutné čekat na timeout, či se vyskytne chyba přenosu a spojení zpomalí (min. 1 RTT (min. čas potřebný k požadavku a odpovědi) navíc). Systému díky značkování a verzování komunikačních zpráv výpadek části komunikace nevadí. Nemusí se jen v rámci nestandardní události dočasně přerušit spojení, ale např. může i vypadnout jeden front server bez vlivu na chod systému.

Více front serverů, resp. více host jmen (názvů hostitele) je výhodné i kvůli typickému omezení browserů na max. počet souběžných spojení na host jméno (typ. 6, u starších prohlížečů 2).

Prohlížeče z důvodu bezpečnosti AJAX dotazy na jiné než aktuální host jména neumožňují. Resp. tento bezpečnostní problém popisuje problém CORS (sdílení zdrojů napříč doménami) a po implementaci správných http hlaviček v odpovědi serveru jej lze řešit.

Pro maximální kompatibilitu s IE a popisovaným CORS problémem byla implementována vlastní knihovna, která specifikace prohlížečů zohledňuje a sama se přizpůsobí. Zároveň řeší abstrakci dotazů vč. možnosti logování, zobrazení probíhajících dotazů, základních metrik (např. uběhlý čas) apod.

3.3 Demo aplikace

BINARY INPUTS



SENSORS

DS18B20 A TEMP: 25.25 [°C]	DS18B20 B TEMP: N/A [°C]	MAX31850 TEMP: 0 [°C]
SHT25 HUMIDITY: 0 [%RH] TEMP: 0 [°C]		
LPS331AP PRESSURE: 0 [kPa] TEMP: 0 [°C]		

BINARY OUTPUTS



mvorisek.version: 5.78

mvorisek.ajax.clientCode: 2 - XMLHttpRequest Level 2 (native CORS support) - modern browsers including IE10+

ID	Status	Created DateTime	Elapsed [ms]	Method	URL
6	2 - Complete	19.1.2014 11:17:1.639	10153	GET	http://root-a.heulo.draft.mvorisek.com/a/debug/speedily/comet/ajax-with-cors/backend.php?last_id=66713
7	1 - Loading	19.1.2014 11:17:11.799	0	GET	http://root-b.heulo.draft.mvorisek.com/a/debug/speedily/comet/ajax-with-cors/backend.php?last_id=66713

Obr. 7 - ukázka demo GUI

Pro testování bylo vytvořeno veřejně dostupné GUI (obr. 7) spustitelné ve všech běžných prohlížečích vč. mobilních na platformách Android a iOS. Opera Mini není a nemůže být (dané omezením Operry Mini a jejími možnostmi http long-pooling a AJAX technologií) podporována, vzhledem k tržnímu podílu, navíc klesajícimu, toto není překážkou.

Demo aplikace je rozdělená na 4 části:

- Digitální vstupy
 - zelená barva znamená log. 1,
 - červená log. 0,

- šedá – senzor není připojený (automatický přechod po 30s nedostupnosti)
- Sensory s nebinárním výstupem
 - Ukazuje se hodnota senzoru, senzor může mít více čidel (např. vlhkoměr měří vlhkost, ale zároveň teplotu)
- Digitální výstupy
 - Kliknutím na příslušné tlačítko lze změnit stav výstupu a ten se v případě připojenosti změní. Zobrazovaná barva odpovídá barvám digitálním vstupům, barva ukazuje stav na serveru, tzn. nezmění se v případě kliknutí a nedoručení akce. Uživatel tak má jistotu, že požadavek byl doručený.
- Log
 - Aplikace vypisuje všechny zadané a realizované požadavky se serverem – s využitím vlastní synchronizační knihovny, která požadavky sleduje a umožňuje je vylistovat společně s meta informacemi. Požadavky jsou ze seznamu po 60s (od vyřízení / chyby) odebírány.

3.4 Sumarizace

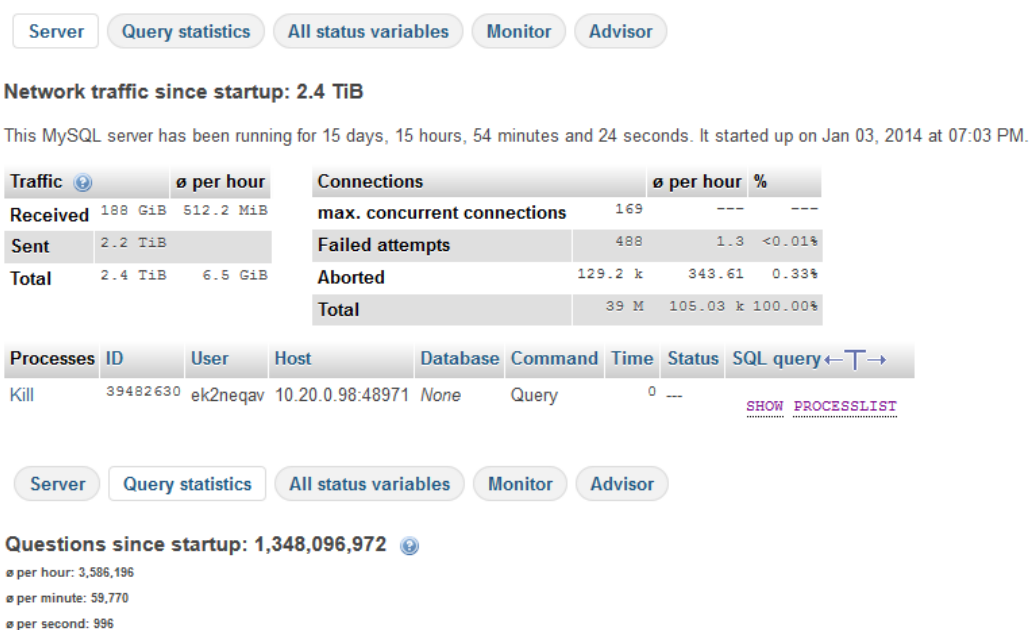
Podářilo se splnit motivační požadavky. Hlavní přednosti systému oproti existujícím řešením a jejich řešení je následující:

- Maximální kompatibilita – systém rozpozná prohlížeč a jeho verzi a přizpůsobí se
- Real-Time synchronizace – stejně jako hlavní jednotka synchronizační knihovna udržuje se serverem navázané spojení a dává serveru možnost klientovi okamžitě odeslat informaci o změně informace ze serveru / jiných senzorů. Známé pod označením Push notifikace.
- Omezení na max. počet aktivních spojení na host jméno – komunikuje s více front servery a počtem těchto serverů se omezení násobí. Takto je možné mít otevřeno desítky aktivních spojení a žádné nemusí čekat, než se vyřídí jiné požadavky. Známé pod označením CORS.

4. HEULO SW – server

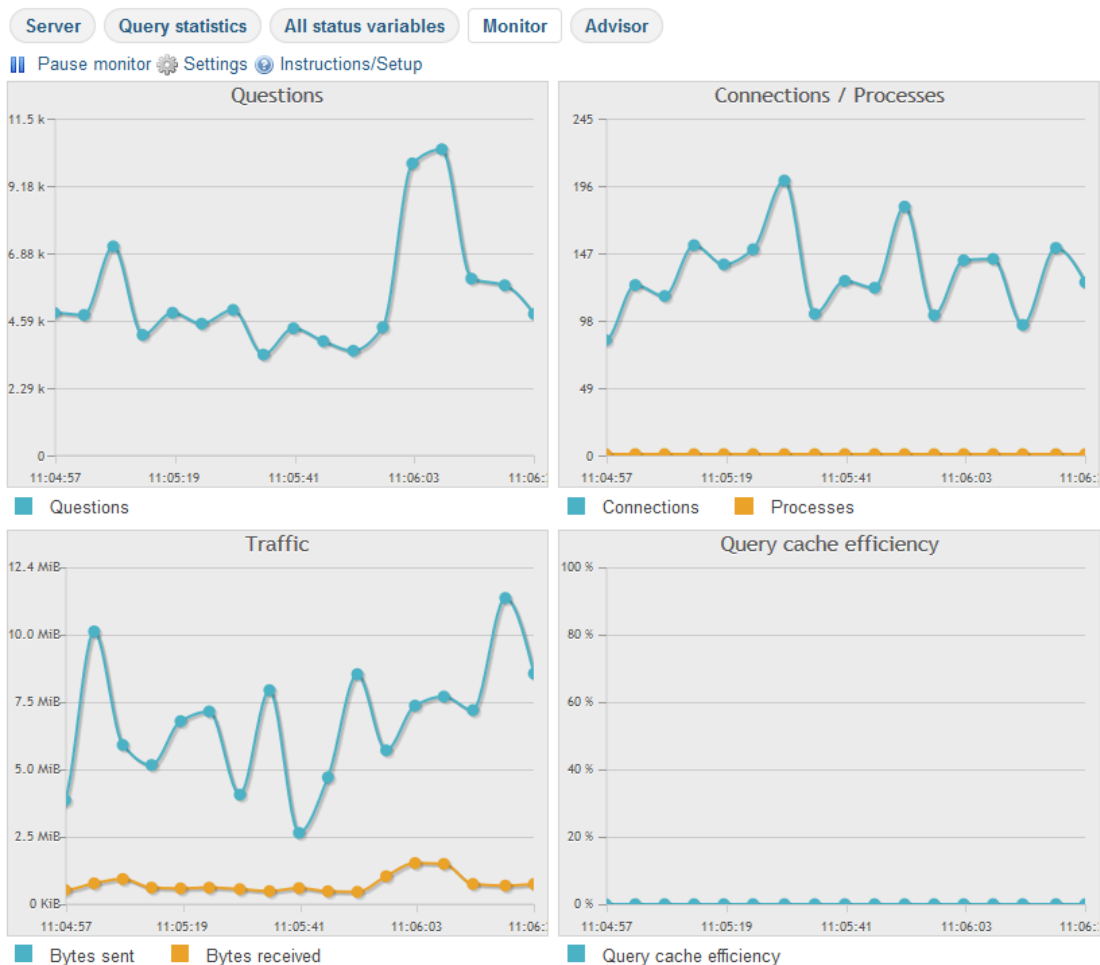
4.1 Úvod

Úzké hrdlo celého systému při velkém počtu aktivně připojených jednotek, jeden fyzický server musí být schopný obsluhovat minimálně 250 hardwarových endpointů – až šest socketů na hw endpoint, až tři sockety na jedno připojení z prohlížeče / klienta připojeného přes API. Při dvou souběžně připojených klientech a max. počtu socketů, server tedy musí v každý okamžik svého chodu udržovat aktivní spojení s $250 \cdot (6 + 2 \cdot 3) = 3000$ souběžnými připojeními! Toto číslo běžně neobsluhují ani největší webhostingy v CZ, viz. screen z databáze webhostingu WebSupport, která obsluhuje podstatnou část databázových dotazů v této spol. Webový server Apache je ve výchozí konfiguraci omezen na 100 konkurenčních spojení, MySQL je obvykle omezeno na 250 souběžných spojení. Pro stabilní chod celé synchronizační architektury je nutné výchozí hodnoty více než 20 násobně zvýšit, resp. zajistit, aby toto zvýšení bylo reálně dosahováno.



Obr. 8 - screen vyřízení databáze webhostingu WebSupport, záznam pořízen 19. ledna 2016 v 11:10.

Dle komunikace s podporou WebSupport používá výhradně SSD, 40k IOPS je oproti 200 IOPS limitu běžných 7.2k otáčkových disků značný rozdíl, avšak plošné nasazení SSD např. pro historické logy dat ze senzorů nemusí dávat smysl.



Obr. 9 - screen vytíženosti databáze webhostingu WebSupport, záznam pořízen 19. ledna 2016 v 11:10.

4.2 Požadavky

Kromě počtu spojení, z důvodu real-time aplikace, je nutné optimalizovat latenci a ještě více peak latenci. Cílem je obsluhovat požadavky do 10 až 20 milisekund, 50 milisekund je přijatelných, 100 milisekund je absolutní limit. Vyšší hodnoty jsou již pozorovatelné a výrazně zhoršují uživatelský komfort.

4.3 Příprava serveru a optimální ukládání databázových dat

Ukládat časové řady je velmi náročné na kapacitu. I jeden senzor vyprodukuje velké množství dat, proměnná denní teplota je zřejmá. U denní teploty je možné data vzorkovat beze ztráty informace, v případě automatizace je však zcela běžné, že se hodnota ze senzorů mění rychleji a záznam ukládáním např. po 30 min. možný není. Např. údaje z čidla pohybu.

Pro dostatečně kapacitní databázi byl stanoven požadavek 10TB. 10TB je na databázi velmi vysoké číslo a ekonomicky není přijatelné tuto kapacitu zajistit

použitím SSD. Proto byly zvoleny 7200 otáčkové Enterprise disky od spol. Seagate, resp. jejich řada ES.3. Zakoupeny byly 4 tyto disky.

Dalším důležitým prvkem je RAID. Opravdový RAID, ne LVM či „fake“ Rapid Storage RAID od Intelu. Po analýze trhu byl vybrán RAID od fy. LSI, RAID LSI 9260 a to vč. BBU (bateriové jednotky) pro bezpečné cachování zápisu.

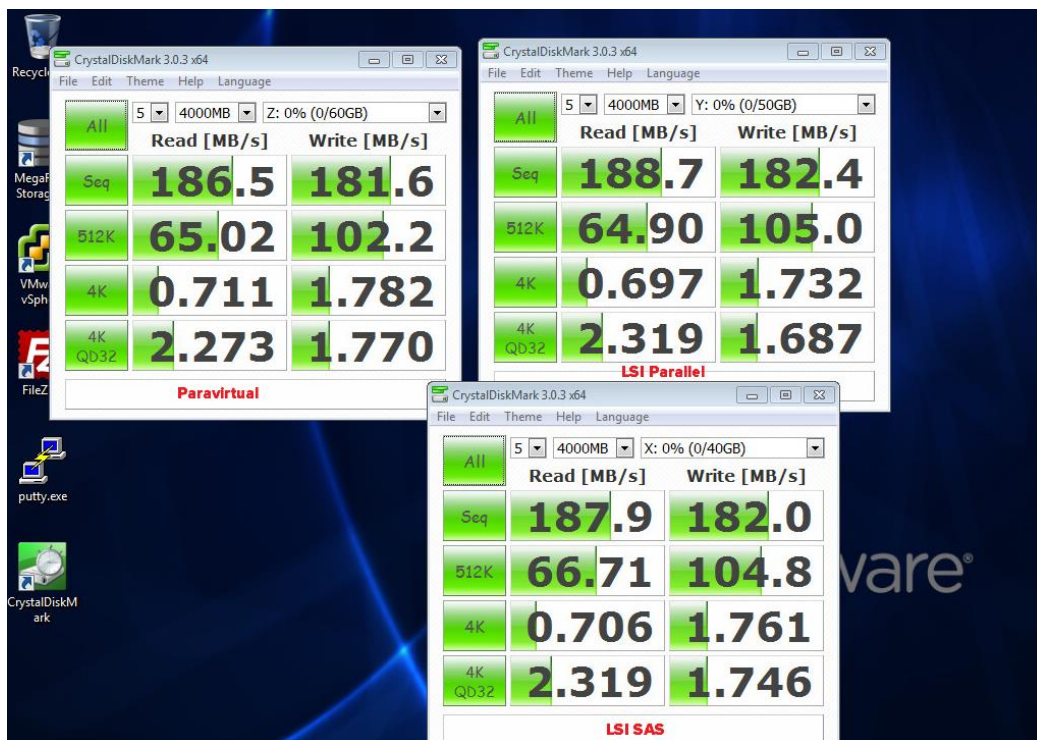
Pro srovnání a reálné hledání bottlenecků byly zakoupeny také SSD od fy. Intel řady 530 s kapacitou 240 GB. Tento typ má MLC buňky, které při intenzivním využívání velmi rychle degradují (při zápisu, částečně však i při čtení). Cena SSD s SLC buňkami je skoro 100x vyšší než cena disků magnetických se 7.2k otáčkami. Z cenových důvodů je v práci hlazen důraz na najít akceptovatelné konfigurace s točivými disky.

10k či 15k otáčkové disky představují jen malé zlepšení a jejich přínosem je spíše jen snížení latencí (na cca 1/2), více IOPS je vhodné kompenzovat vyšším počtem levnějších disků.

Experimentálně bylo zjištěno, že cache technologie typu CacheCade apod. nepřináší zásadní zlepšení. Podobně jako u rychlejších disků je vhodné systém osadit více disk a zvýšit vedlejším produktem kapacitu, než investovat do cache technologií, které mají omezenou kapacitu a data, která se často nepoužívají vůbec neakcelerují.

V rámci testů byly pro objektivnost měřeny i dva 1 TB disky WD Blue a jeden 2.5" 7.2k 320 GB disk řady Momentus 7200.4.

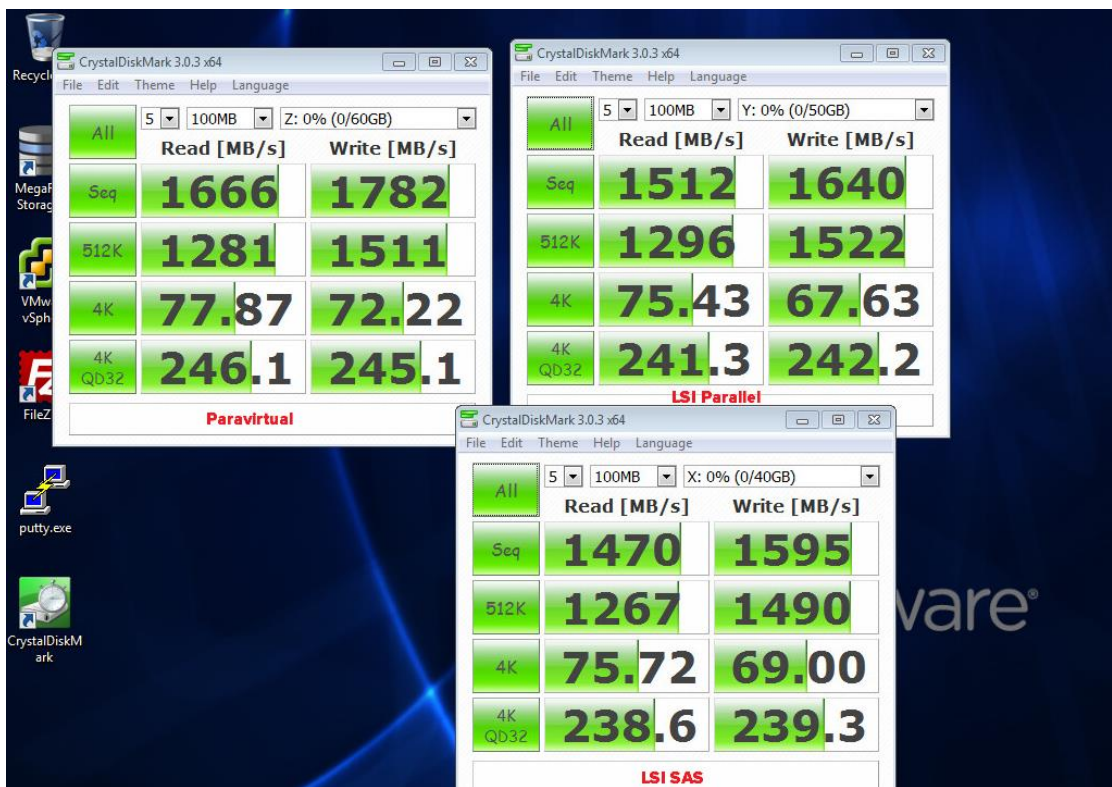
Použitý hypervizor Vmware ESXi nabízí 3 typy připojení disků: emulaci LSI SAS, emulaci LSI Parallel a Paravirtual. V rámci práce bylo provedeno měření každého typu. Paravirtual připojení se ukázalo v rámci testů jako nejvýhodnější a další měření byly prováděny s tímto připojením.



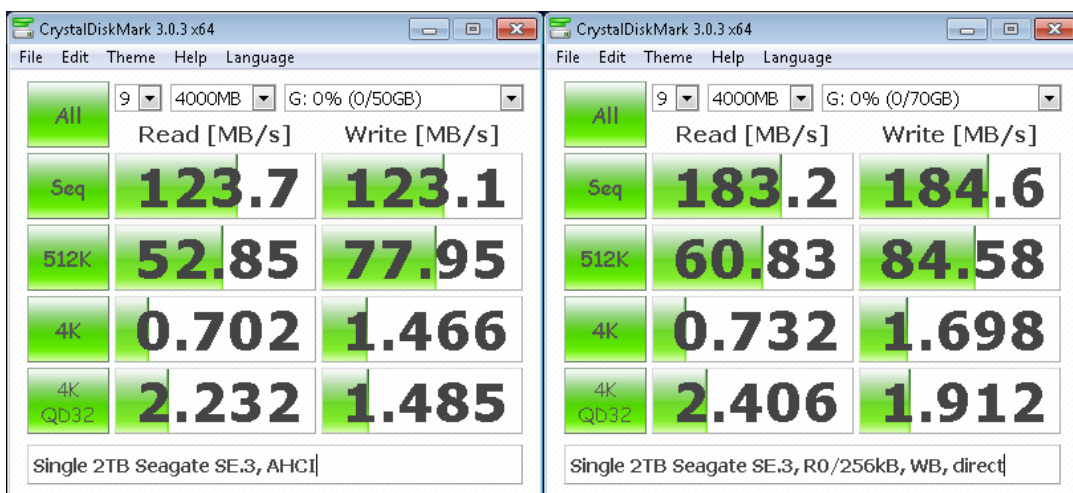
Obr. 12 - srovnání variant připojení HDD, 4 GB testovací soubor, jeden disk Seagate ES.3 2 TB připojený přes HW RAID.



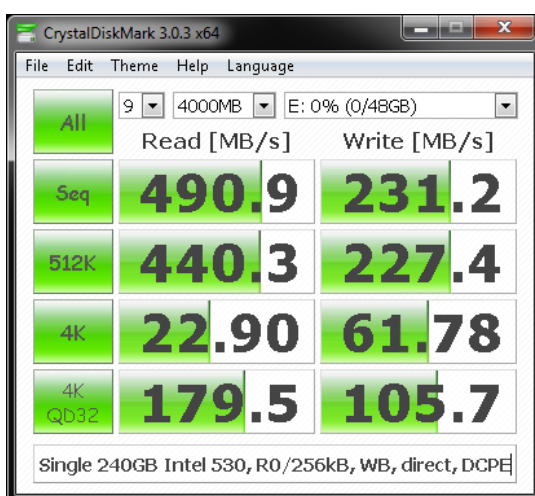
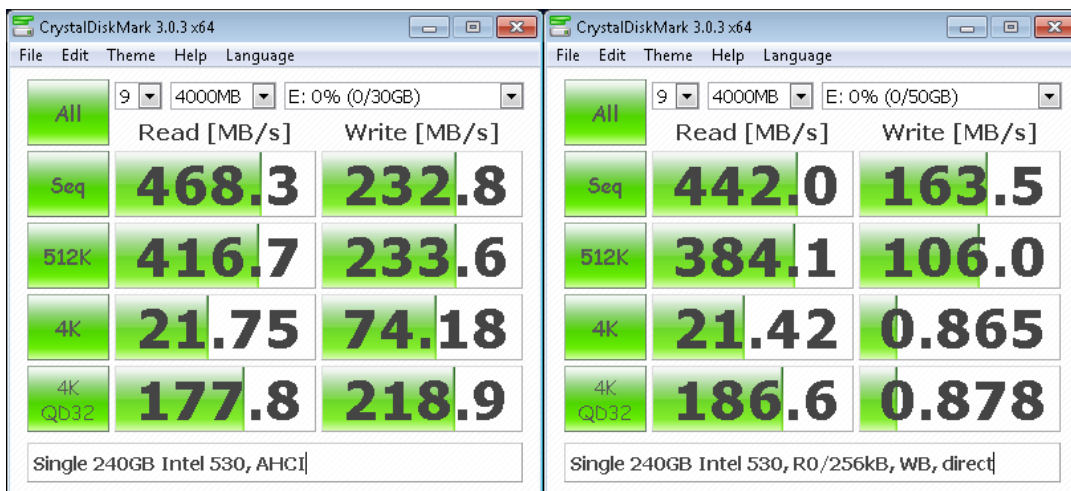
Obr. 13 - srovnání variant připojení HDD, 4 GB testovací soubor, jeden disk Seagate ES.3 2 TB připojený přes HW RAID s read-write CacheCade (tvořenou jedním netočivým diskem Intel 530 s kapacitou 240 GB).



Obr. 14 - srovnání variant připojení HDD, 100 MB testovací soubor, jeden disk Seagate ES.3 2TB připojený přes HW RAID. Použitý HW RAID s kapacitou paměti 512 MB, operace tedy probíhaly z cache se zápisem na pozadí. Test ukazuje hlavně propustnost RAIDu/VMwaru/Windows.

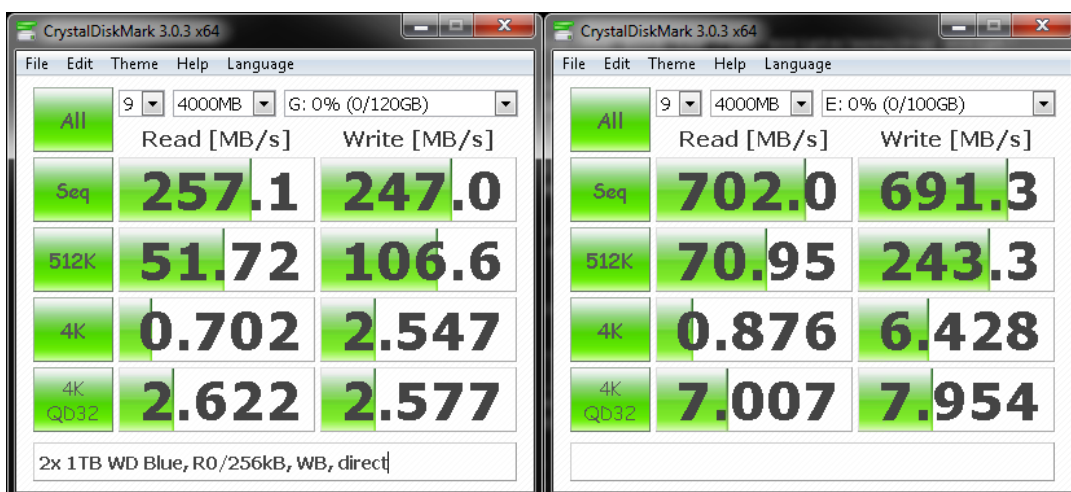


Obr. 18 - výkon 2 TB disku Seagate ES.3 (na screenu je špatně uvedeno SE.3), 4 GB testovací soubor, další popis přímo ve screenech.

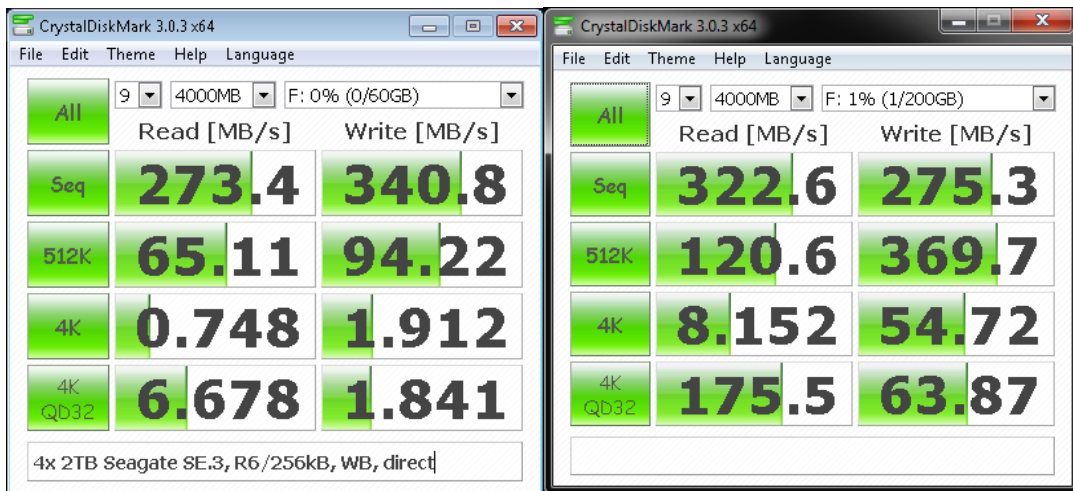


Obr. 19 - výkon 240 GB SSD Intel 530, 4 GB testovací soubor, další popis přímo ve screenech.

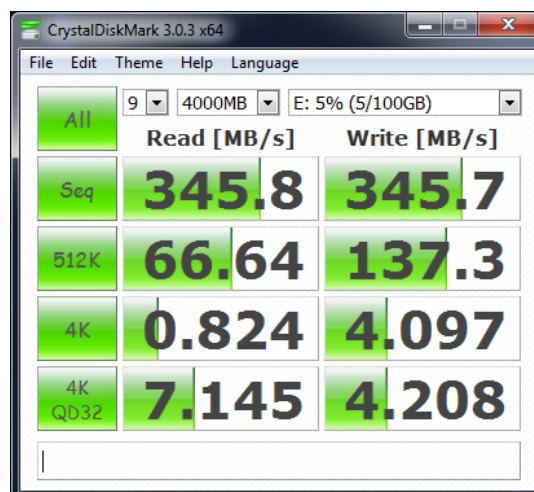
Horní pravý a dolní obrázek zobrazuje výkon při stejném nastavení RAIDu, resp. v dolním je „Disk Cache Policy“ vlastnost nastavena na „Enabled“. SSD MLC je ve skutečnosti při zápisu velmi líné a ve většině případů není vhodné, nesplňuje ACID normu.



Obr. 20 - srovnání RAID0, vlevo 2x 1 TB WD Blue, vpravo 4x 2 TB Seagate ES.3.



Obr. 21 - srovnání RAID6 se čtyřmi 2 TB Seagate ES.3 disky, screen vpravo se stejnou konfigurací plus read-write CacheCade (tvořenou jedním netočivým diskem Intel 530 s kapacitou 240 GB).



Obr. 22 - RAID10 se čtyřmi 2 TB Seagate ES.3 disky.

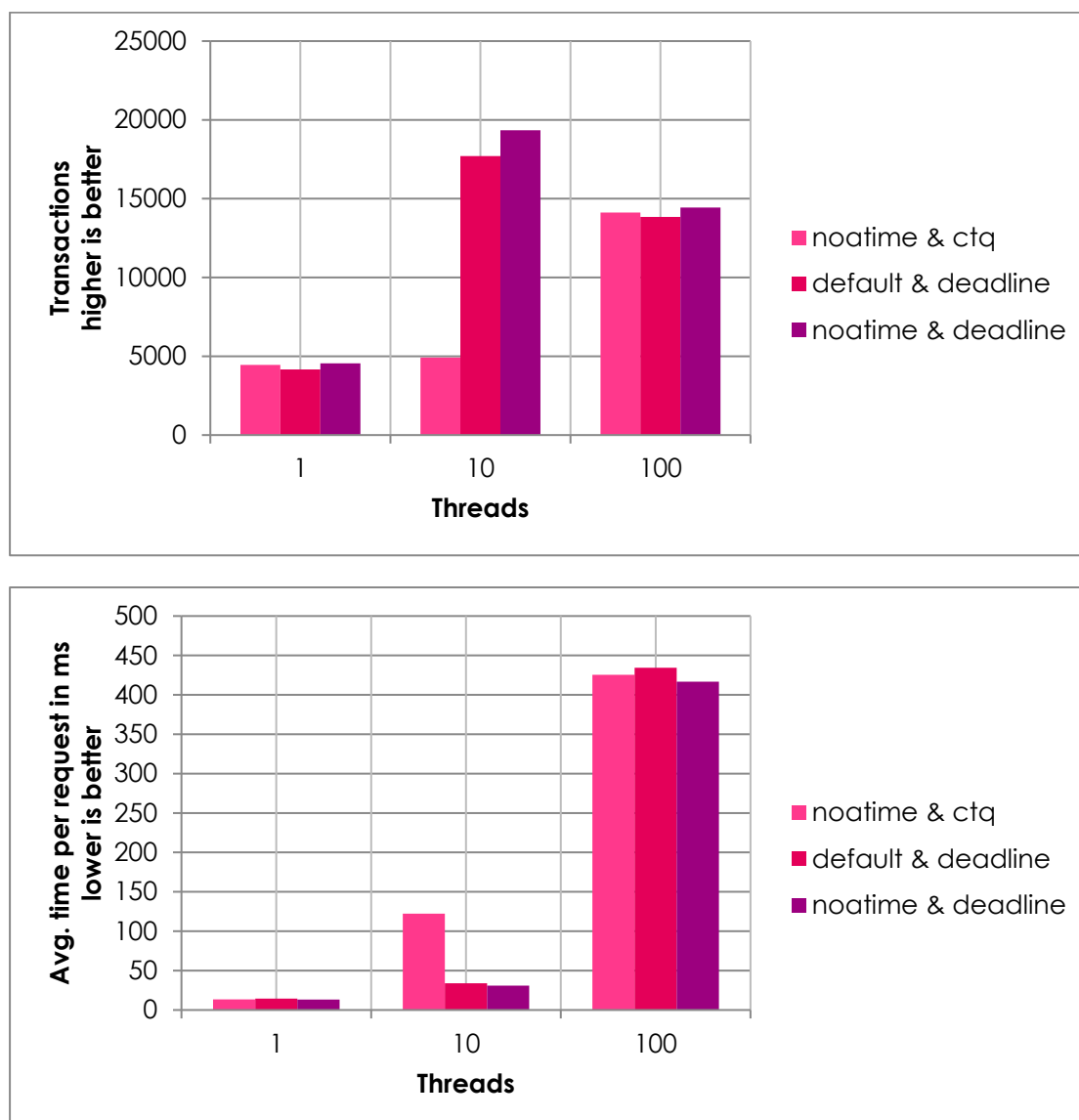
Další CrystalDiskMark a rozsáhlé Atto testy jsou přiloženy na CD.

4.4 Optimální plánovač diskových úloh

Na základě srovnání byl vybrán souborový systém ext4. Tento souborový systém je v systému Debian od verze 7 zároveň výchozí.

Důležitým faktorem je plánovač IO úloh.

Dle měření (Obr. 22b) je vhodné výchozí plánovač ctq změnit na deadline, který nabízí trochu vyšší propustnost a podstatně kratší reakční doby.



Obr. 22b – srovnání plánovačů v závislosti na počtu dotazů

Změnu plánovače je možné provést editací souboru `/etc/default/grub`, ve kterém je nutné upravit řádek `GRUB_CMDLINE_LINUX_DEFAULT`:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet elevator=deadline"
```

A změny aplikovat příkazem: `update-grub`

4.5 Sumarizace

Požadavky na výkon se podařilo splnit. Na základě testů byl vybrán optimální typ disků a RAID řadič. Testy rovněž vybraly nejlepší připojení diskových jednotek do virtuálního stroje hostujícího databázi a optimální plánovač diskových úloh.

5. Závěr

I přes řadu překážek v každé části projektu a se podařilo se splnit motivační požadavky.

Hlavní přednosti systému oproti existujícím řešením jsou:

- Cenová dostupnost a velikost
- Snadné připojení senzorů prostřednictvím sběrnice One-Wire
- Sensory je možné připojit dvěma vodiči a po datovém vodiči je i (parazitně) napájet
- Sensory, vstupy a výstupy jsou dostupné z internetu i přes NAT, není potřeba veřejné IP adresy
- Implementovaný komunikační protokol je průchozí přes proxy servery
- Synchronizace probíhá v reálném čase stylem Push notifikací
- Server a serverová architektura byla navržena pro růst systému a stabilní odezvy
- Navržený systém je plug & play a nevyžaduje specifickou konfiguraci sítě
- Sběrnici implementuje One-Wire protokol dle specifikace výrobce, existující senzory splňující tuto specifikaci lze připojit a systém snadno rozšířit
- Pro připojení senzorů, které na One-Wire sběrnici nejsou dostupné, byl vyvinut převodník, který vyniká univerzálností a umožňuje připojit jakýkoliv senzor skrz typické lokální sběrnice typu I²C, SPI nebo UART

Dosažené výsledky považuji za skvělé. Realizace hardwaru, ale i softwaru mě obohatila o cenné zkušenosti.

Za nejnáročnější část považuji vývoj převodníku, ve kterém jsem byl omezen možnou spotřebou. Při vývoji firmwaru jsem musel kritické část kódu optimalizovat v assembleru, aby zůstala splněna specifikace a systém byl kompatibilní s jinými zařízeními. V době nečinnosti se takt převodníku naopak snižuje.

Časově nejnáročnější naopak byla implementace komunikační část jak se senzory, tak serverem.

Systémové požadavky - klient

- Procesor: 1200MHz
- RAM: 256MB
- Software: webový prohlížeč s podporou AJAXu a CORS

Obecně lze k ovládání použít jakékoliv zařízení, které obsahuje prohlížeč s podporou CORS a je připojené do sítě internet rychlostí alespoň 256 kb/s.

Opera Mini podporována není, Opera Mini real-time synchronizaci či jiné asynchronní události nepodporuje. Tento prohlížeč s technologií renderování na serveru má minimální podíl na trhu a je na ústupu .

Doporučené systémové požadavky - server

- Operační systém: Debian 7 64-bit
- Procesor: 4 jádra 3.2 GHz
- RAM: DDRIII-1600 32 GB ECC
- Software: Apache 2.2, PHP 5.5, MySQL 5.5
- Ostatní:
 - Servery by měly být alespoň dva ve dvou geograficky oddělených lokalitách s přiměřenou vzdáleností od klientů
 - 4 veřejné IP / server

Seznam vývojového software

- Netbeans – vývojové IDE pro PHP
 - <https://netbeans.org/>
- MPLAB IDE X – vývojové IDE pro procesory dsPIC
 - <https://microchip.com/>

Seznam použitých zkratek

- PHP – jazyk serverové části
 - <http://php.net/>
- CORS – Cross-origin resource sharing
 - http://en.wikipedia.org/wiki/Cross-origin_resource_sharing
- MySQL – použitý databázový systém
 - <http://www.mysql.com/>
- UX – user experience
 - http://en.wikipedia.org/wiki/User_experience
- GUI – graphical user interface
 - https://en.wikipedia.org/wiki/Graphical_user_interface
- a další obvykle zkratky (HW, IP, ...)

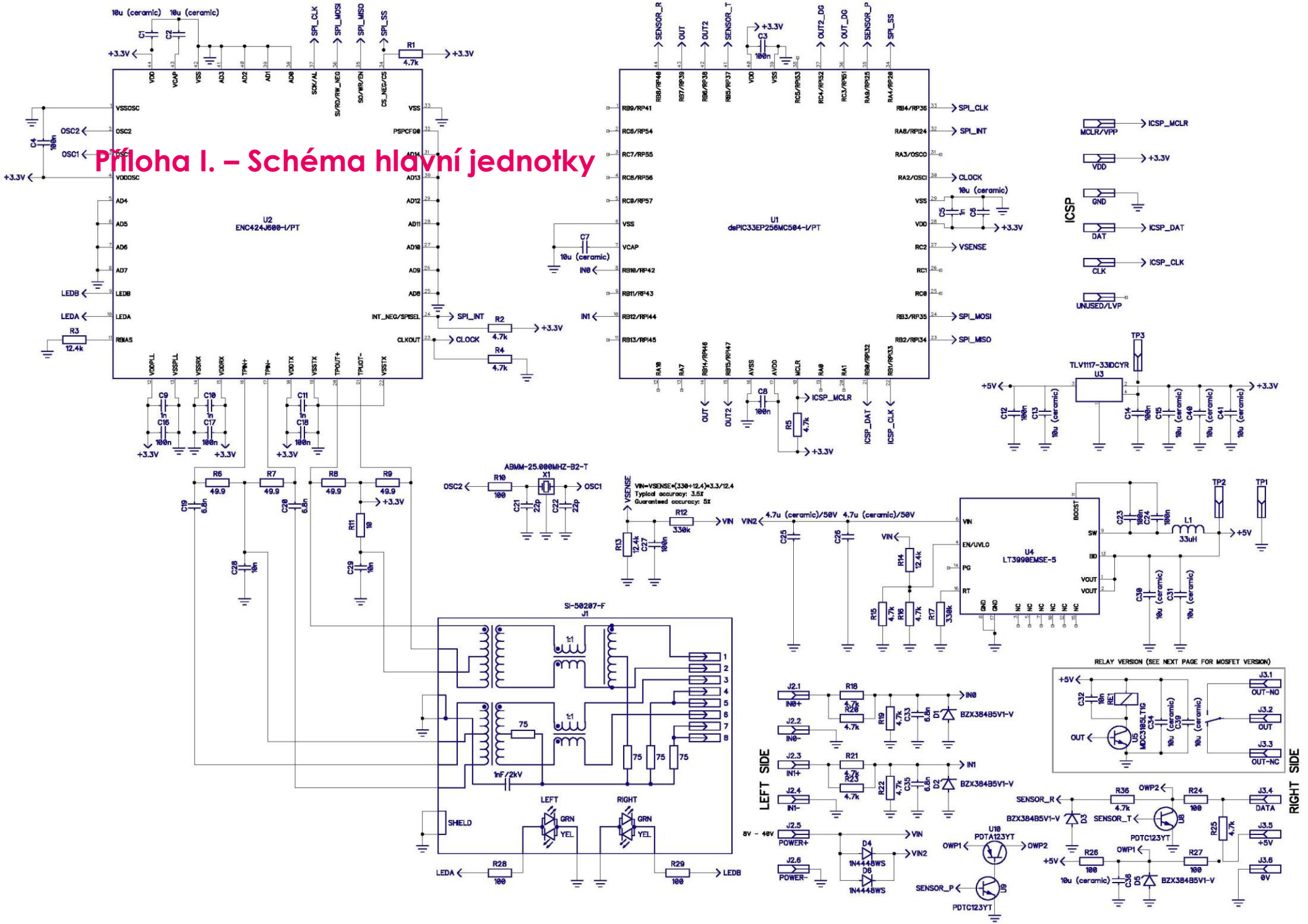
Přílohy

- Specifikace BP
 - dokument dokumentující postupy práce s hodnocením v čase
- CD
 - main.docx
 - el. podoba této práce ve formátu docx
 - main.pdf
 - el. podoba této práce ve formátu pdf
 - pictures/
 - použité obrázky v 1:1 rozlišení
 - debug/
 - surová data z ladění
 - others/
 - další podklady, které jsou součástí projektu, avšak vzhledem k požadovanému rozsahu dokumentace nejsou její součástí – např. záznamy měření, simulace, ukázky kódu apod.

Seznam odborných zdrojů

- [1] Moje ročníková práce – Synchronizační systém LAURA
 - Uložena ve škole SPŠ Ostrov
- [2] One-Wire specifikace pro implementaci One-Wire master
 - <https://www.maximintegrated.com/en/app-notes/index.mvp/id/2965>
 - <https://www.maximintegrated.com/en/app-notes/index.mvp/id/126>
 - <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>
 - <https://www.maximintegrated.com/en/app-notes/index.mvp/id/244>
 - <https://www.maximintegrated.com/en/app-notes/index.mvp/id/148>
- [3] @TODO

Príloha I. – Schéma hlavnej jednotky



Příloha II. – Schéma One-Wire vlhkoměru

